User Guide

Predictive Workbench R-3.6

Contents

1. About This Guide

1.1. Document History

The following table gives an overview of the most recent document updates:

Product Version	Date (Release date)	Description
BDB Predictive Workbench 1.0	June 9th, 2015	First Release of the document
BDB Predictive Workbench 2.0	Feb 18th, 2016	Updated document
BDB Predictive Workbench 2.0	May 31st, 2016	Minor Changes and Editing of the document
BDB Predictive Workbench 2.5	November 9th, 2016	Updated document
BDB Predictive Workbench2.5.1	January 3rd, 2017	Updated document
BDB Predictive Workbench 2.5.3	March 16th, 2017	Updated document
BDB Predictive Workbench 3.0	August 31st, 2017	Updated document
BDB Predictive Workbench 3.0	November 22nd, 2017	Modification and Editing of the document
BDB Predictive Workbench 3.2	January 25th, 2018	Updated document
BDB Predictive Workbench 3.5	April 15th, 2018	Updated document
BDB Predictive Workbench 3.6	August 20th, 2018	Updated document

1.2. Overview

This guide covers steps to:

- Access the BDB Predictive Analysis
- Server Requirements and Deployment Details for the BDB Predictive Analysis
- Designer Part of the BDB Predictive Analysis
- Result or Analysis Part of the BDB Predictive Analysis

1.3. Target Audience

This guide aims at business professionals, data analysts, data scientists, and statisticians who use BDB Predictive Workbench tool to conduct various experimentations with data as in a Data Science Lab.

2. Introducing BDB Predictive Analysis Tool

2.1. Introduction to the BizViz Predictive Analysis

BDB Predictive Analysis is a statistical analysis tool that empowers its users by providing predictive models. These Predictive Models can be used to envision the future outcomes of business processes based on past data. It is a user-friendly tool that shields users from the mathematical complexity and offers an interactive graphical interface to provide a smooth, intuitive experience. It enables the users to discover hidden insights and relationships in their data by applying various statistical algorithms provided by the popular R statistical language, Spark ML, and Python.

2.2. Prerequisites

2.2.1. Pre-requisites for Predictive Analysis

1. Predictive Analysis is a web-based service so, the only requirement is a browser.
2. Predictive Analysis can be viewed only in desktops (mobile and tablet views are not supported).
3. R server and Predictive Spark App Settings should be configured from the Administration module.
4. The user should be provided with all the necessary permissions to access and use the Predictive Analysis plugin from the User Management module of the BizViz Platform.
5. The user should be permitted to access Data Management module from the BizViz Platform to use query service and Cassandra reader and writer for Predictive Analysis.
6. Limit of data connectors rows needs to be configured via the Administration module.

2.2.2. R Server Requirements

1. R server should be deployed publically.
2. Port should be open.
3. R server should be configured in the Administration page of the BizViz platform.
4. Following packages should be installed on the R Server for predefined algorithms:
   - stringr
   - forecast
   - arules
   - arulesViz
   - rpart
   - e1071
5. In the case of Custom R Script, script-specific packages should be installed on the R Server.

2.2.3. Predictive Spark Application Deployment Details

1. Spark, Hadoop, Cassandra should be running in Cluster. For this application, Cluster should have free resources (Min 3 Core, 2 GB RAM in each executor according to application property).
2. Create a file with name spark_pa.properties in spark's configuration folder (cd $SPARK_HOME/conf) and provide the following properties:

   - spark.master   <Spark master url:port> #Mandatory
   - spark.app.name   Spark Predictive Application #Mandatory
   - spark.scheduler.mode   FAIR
   - spark.eventLog.enabled   true
   - spark.eventLog.dir   <log dir>
   - spark.serializer   org.apache.spark.serializer.KryoSerializer
   - spark.extraListeners
     org.apache.spark.ui.jobs.JobProgressListener,org.apache.spark.PASparkListener #Mandatory (Custom listener for the PA app)

3. **Port Configuration:** Any port series is fine provided they are exposed via the firewall. This is for the nodes within the Spark cluster.
   - spark.ui.port   5003
   - spark.history.ui.port   20080
   - spark.driver.port   20081
   - spark.executor.port   20082
   - spark.fileserver.port   20083
   - spark.broadcast.port   20084
   - spark.replClassServer.port   20085
   - spark.blockManager.port   20086

4. **Cassandra Configuration**
   - spark.cassandra.input.split.size_in_mb   16
   - spark.cassandra.input.fetch.size_in_rows   1000

5. **Spark PA Configuration**
   - spark.pa.fs.default.name   <HDFS host URL:port>*hdfs://localhost:8020* #Mandatory
   - spark.pa.process.queue.size   10 #Mandatory Default is 10. Queue size for PA app.
   - spark.pa.process.pool.size   10 #Mandatory Default is 10. Pool size for PA app.
   - spark.pa.cache.size   100 #Mandatory Default is 100. Cache size for PA app.
   - spark.pa.cache.timeout_sec   600 #Mandatory Default is 600 sec. Cache timeout for PA app
   - spark.pa.hdfs.model.dir   *hdfs://hostname:port/directory name* #Mandatory hdfs storage location for the models *hdfs://localhost:8020/pa/model*
   - spark.pa.hdfs.tmp.dir   hdfs://hostname:port/director name_#Mandatory *hdfs://localhost:8020/pa/tmp*
   - spark.pa.model.timeout_sec   86400 #Mandatory Default is 86400 (1 day). Time interval for deleting temporary model/s from the temporary hdfs location.

spark-pa.properties

6. **Copy shade jar of the pa_spark bundle in "spark/jars/" folder**
   - Com.bdbizviz.pa.spark-shade-2.2.0.jar
7. **Create a Script file named "start-pa.sh" in Spark's sbin folder to start the application**

If you need to execute in Kerberos mode, you need to generate the key tab file.

Script Contents in Kerberos Mode:

```
#!/usr/bin/env bash


dir="$(cd "`dirname "$0"`"/..; pwd)"

nohup $dir/bin/spark-submit --keytab $dir/conf/hdfs.keytab \
--principal hdfs/<principlename> \
--executor-memory 3G  --executor-cores 4 --num-executors 1 \
--verbose --properties-file $dir/conf/spark-pa.properties \
--driver-class-path $dir/jars/com.bdbizviz.pa.spark-shade
2.2.0.jar \
--class com.bdbizviz.pa.spark.executor.Executor --master yarn
deploy-mode client \
jars/com.bdbizviz.pa.spark-shade-2.2.0.jar 18786 >>
```

$dir/logs/spark-pa.log 2>&1&

please note that 18786 is a jetty port and can be changed to suite your needs

Script Contents in Normal Mode:

```
#!/usr/bin/env bash

dir="$(cd "`dirname "$0"`"/..; pwd)"

nohup $dir/bin/spark-submit \
--executor-memory 3G  --executor-cores 4 --num-executors 1 \
--verbose --properties-file $dir/conf/spark-pa.properties \
--driver-class-path $dir/jars/com.bdbizviz.pa.spark-shade
   2.2.0.jar \
--class com.bdbizviz.pa.spark.executor.Executor --master yarn
deploy-mode client \
jars/com.bdbizviz.pa.spark-shade-2.2.0.jar 18786 >>
$dir/logs/spark-pa.log 2>&1&
```

Note: 18786 is a jetty port and can be changed to suit your needs.

start-pa.txt

Save this file as a shell script (.sh)

8. Start Application with this command- **sbin/start-pa.sh**
9. Confirm the Spark PA Application is running on YARN:

Cluster Metrics																
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
5	0	3	2	8	22 GB	25 GB	0 B	8	20	0	5	0	0	0	0

Scheduler Metrics			
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:5120, vCores:4>

Show 20 entries   Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
application_1476353597736_0005	hdfs	Spark Predictive Application	SPARK	default	Tue Oct 18 14:52:02 +0550 2016	N/A	RUNNING	UNDEFINED		ApplicationMaster	0
application_1476353597736_0004	hdfs	Spark Predictive Application	SPARK	default	Mon Oct 17 17:13:15 +0550 2016	Tue Oct 18 14:49:23 +0550 2016	FINISHED	SUCCEEDED		History	N/A
application_1476353597736_0003	hdfs	Spark Predictive Application	SPARK	default	Thu Oct 13 16:11:09 +0550 2016	Mon Oct 17 17:11:56 +0550 2016	FINISHED	SUCCEEDED		History	N/A
application_1476353597736_0002	hdfs	smb-analytics-17	SPARK	default	Thu Oct 13 15:53:04 +0550 2016	N/A	RUNNING	UNDEFINED		ApplicationMaster	0
application_1476353597736_0001	hdfs	org.apache.spark.sql.hive.thriftserver.HiveThriftServer2	SPARK	default	Thu Oct 13	N/A	RUNNING	UNDEFINED		ApplicationMaster	0

Note: Confirm that application has sufficient resources by the highlighted columns such as "Cores" and "Memory per Nodes."

2.2.4. Predictive Python Application Deployment Details
The Predictive Python Server is mainly built upon the Django framework. The overall server and all necessary components run in a virtual environment that keeps it in a separate virtual space regarding processing.

2.2.4.1. Setup Virtual Environment
Please follow the below instructions to set up Virtual Environment:

- Step 1- Updating the Linux System
  i) For Centos 7.0
     - $ sudo yum – y update
     - $ sudo yum – y install yum-utils
     - $ sudo yum – y groupinstall development
  ii) For Ubuntu
     - $ sudo apt-get upgrade
- Step 2- Installing Python 3.6
  i) For Centos 7.0
     - $ sudo yum –y install https://centos7.iuscommunity.org/ius-releas.rpm
     - $ sudo yum –y install python36u
     - $ sudo yum –y install python36u-pip
  ii) For Ubuntu
     - $ sudo apt-get update
     - $ sudo apt-get install python3.6
     - $ wget https://bootstrap.pypa.io/get-pip.py
     - $ python3 get-pip.py
  iii) To check Python 3.6 in System,
     - $ python 3.6 –V
- Step 3- Creating Virtual Environment
  i) $ cd <path-to-virtual-environment-directory-to-create>
     - eg: $ cd ~/

  ii) $ mkdir <VIRTUAL_ENVIRONMENT_DIRECTORY_NAME>
     - eg: $ mkdir venv

  iii) $ virtualenv –system-site-packages –python=/usr/bin/python3.6 <VIRTUAL_ENVIRONMENT_DIRECTORY_NAME>
     - eg: $ virtualenv –system-site-packages –python=/usr/bin/python3.6 venv
- In case if users find errors while installing the above Commands, follow the below instructions, (at this point we are assuming that users have successfully installed **python3.6** into their machines)
  i) $python3.6mvenv<**PATH_TO_VIRTUAL_ENVIRONMENT_DIRECTORY**> --without-pip
     - eg: $ python3.6 –m venv /home/bizviz/venv --without-pip

  ii) $ cd <VIRTUAL_ENVIRONMENT_DIRECTORY>
  iii) $ source bin/activate   # Activating Environment
  iv) $ wget https://bootstrap.pypa.io/get-pip.py   # Obtaining pip File
  v) $ python get-pip.py   # Installing pip

  Note: In case still you are facing problems with the above installation

- Follow the link ->https://snakeycode.wordpress.com/2017/11/18/working in-python-3-6-in-ubuntu-14-04/
- Alternatively, please google as per your system configuration. Virtual Environment is set on System. The further installation takes place in the activated virtual environment. To Activate Virtual Environment,

- $ cd <VIRTUAL_ENVIRONMENT_DIRECTORY>
- $ source bin/activate
  i) eg: $ cd /venv
  ii) eg: $ source bin/activate

2.2.4.2. Prerequisites for Predictive Analysis Python

1. **Ports**
   Make sure Ports needed for PA are accessible from the machine that has BizViz environment. List of ports is given below,

   a. Django Server Port – 8000s
   b. RabbitMQ Server Port - 5672

2. **Karaf Directory for Storing Temporary Data Files**
   The temp folder should have Read/Write/Delete permission since temporary data files get stored and deleted inside this directory by PA application.

3. **Dependencies** for Python Server
   Below are details of dependencies which are required for Predictive Python Server to operate correctly.

   Note: Please activate the virtual environment before dependency installation.

Django Server related Packages			
Sr. No.	Package Name	Version	Installation Step(s)
1.	Django	1.10	$ pip install django==1.10
2.	Djangorestframework	-	$ pip install djangorestframework
3.	Channels	-	$ pip install channels
4.	asgi-rabbitmq	Latest	$ pip install asgi_rabbitmq
5.	Celery	Latest	$ pip install celery
6.	rabbitmq-server	Latest	$ sudo apt-get install rabbitmq-server
7.	python3-tk	Latest	$ sudo apt-get install python3-tk
8.	python3.6-dev	Latest	$ sudo apt-get install python3.6-dev

Table 3.1: Dependency Package Installation Details

Scientific & Chart Plotting Packages			
Sr. No.	Package Name	Version	Installation Step(s)
1.	Numpy	1.13.1	$ pip install numpy==1.13.1
2.	Scipy	0.19.1	$ pip install scipy==0.19.1
3.	Scikit-learn	0.19.0	$ pip install scikit-learn==0.19.0
4.	Pandas	0.21.0	$ pip install pandas==0.21.0
5.	Matplotlib	2.0.2	$ pip install matplotlib==2.0.2
6.	Bokeh	0.12.4	$ pip install bokeh==0.12.4

7.	Bokeh node packages	-	Follow this link -> https://bokeh.pydata.org/en/latest/docs/dev_guide/setup.html#node-packages
			 $ pip install npm
			 $ pip install nodejs

8.	Paramiko	2.4.0	$ pip install paramiko==2.4.0
9.	Schema	0.6.6	$ pip install schema==0.6.6
10.	Elasticsearch	5.5.1	$ pip install elasticsearch==5.5.1
11.	Termcolor	Latest	$ pip install termcolor
Database Connector Packages			
Sr. No.	Package Name	Version	Installation Step(s)
1.	MySql-	2.1.6	$ pip install mysql-connector==2.1.6

	connector		

2.	PyMsSql	2.1.3	• In Centos 7.0
			  ○ $ sudo yum install freetds-devel
			  ○ $ pip install pymssql==2.1.3
			• In Ubuntu
			  ○ $ sudo apt-get install freetds-dev
			$ pip install pymssql==2.1.3

3.	cx_Oracle	6.0.2	$ pip install cx_Oracle==6.0.2 **Note**: And Install instaclient by oracle using this instruction => https://oracle.github.io/odpi/doc/installation.html#linux

Note: The version number depicted in Table 3.1 is initial version values which we have followed at the time of the development server, for better experience latest version can be installed. Please check for package document before installing.

2.2.4.3. Setting –up Predictive Python Project

As for now, we have collected the required packages along with our Virtual Environment & Django server setup. In this step, we obtain the project bundle from the git-lab repository and will migrate to the current system.

Note: Please ensure that you have installed 'Git' in your system before proceeding.

Follow the below steps to acquire the project,

- $ git clone URL   # here URL correspond to git-lab repo for cloning
- $ cd <path-to-PROJECT_DIR>   # place PROJECT_DIR into

  VIRTUAL_ENVIRONMENT_DIRECTORY

We have collected the bundle from the repo. For better convenience, please make the directory structure as given below,
~ /<VIRTUAL_ENVIRONMENT_DIRECTORY> /PA_Python /bizviz3.5 /python-predictive

Explanation,

- <VIRTUAL_ENVIRONMENT_DIRECTORY> is the directory where our virtual environment has been set up
- PA_Python is a directory in which we create,
  i)   <CACHE_DIR> named: **'CacheData'**
  ii) <SAVED_MODEL_DIR> named: **'SavedPythonModels'**
  iii) <VALIDATION_DATA_DIR>: **'ValidationData'**
  iv) <CELERY_DIR>: **'celery'**
- bizviz3.5 is a git-cloned directory
- python-predictive is our project bundle

Directory Structure of Cloned Project looks something like as shown in the image; the images show the sub-directories and files present inside the **python-predictive** folder,
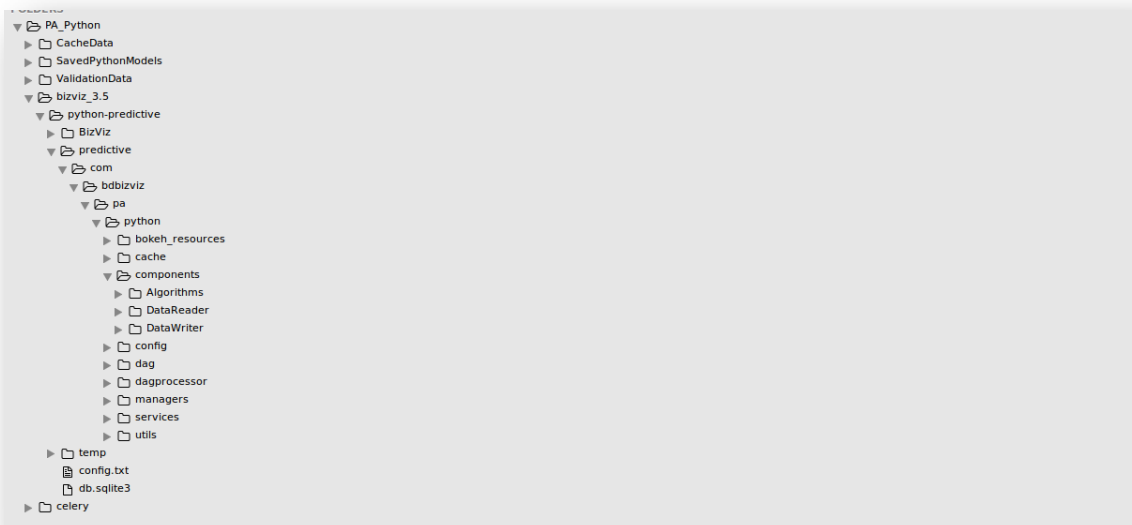
Fig. Directory Structure of Python Predictive

Note: Please provide correct details in,

- *python-predictive/config.txt*

    of <PY_IP> which is the python interpreter inside the Virtual Environment,

    <BASE_DIR> which is the path till 'PA_Python'directory,

    eg. BASE_DIR = /home/bizviz/Desktop/PA_Python/

    and <SERVER_IP_ADDR>

Note: When you have done your RabbitMQ configurations, please update the RabbitMQ details also in the *config.txt*

- *python-predictive/predictive/com/bizviz/pa/python/config/properties.py* file
    i) All required details needed to setup Django Server and Project is already given in *config.txt* file above. In *properties.py*, you can give the System Username & Password and the path to <CACHE_DIR>
    ii) These details can be used when you are using a distributed Django Servers Environment, (i.e., Distributed Celery Workers on different-different Machines)
- $ cd <path-to-python-predictive>
- $ python manage.py migrate   # To migrate server onto current system settings
    Now, we will setup RabbitMQ Configuration. Follow the below steps,

- $ rabbitmqctl add_user USERNAME PASSWORD
    i) eg: $ rabbitmqctl add_user pa_python password123
- $ rabbitmqctl set_user_tags USERNAME TAG
    i) eg: $ rabbitmqctl set_user_tags pa_python administrator
- $ rabbitmqctl add_vhost VIRTUAL_HOST_NAME
    i) eg: $ rabbitmqctl add_vhost django_app
- $ rabbitmqctl set_permissions –p VIRTUAL_HOST_NAME USERNAME CONFIG
    i) eg: $ rabbitmqctl set_permissions –p django_app pa_python ".*"".*"".*"

These above configurations are as per the initial project configuration. You can give configuration according to your wish.

Note**:** Please update the same RabbitMQ details in the **python-predictive/config.txt** file.

For more details on RabbitMQ configuration please visit –

- https://www.rabbitmq.com/rabbitmqctl.8.html
- https://www.rabbitmq.com/configure.html

At last, we create a superuser, so with these credentials in Base64 encoded (Basic Auth), we can access the views of Django server.
- $ cd <path-to-python-predictive>
- $ python manage.py createsuperuser
- Then enter your preferred credentials. Moreover, set up the same in Predictive Settings for Python Server Setting in Admin Module on BizViz Platform

2.2.4.4. Starting the Django Server
- Open a Terminal, then execute below commands
  i) $ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>
  ii) $ source bin/activate
  iii) $ cd <path-to-python-predictive>
  iv) $ celery–A BizViz worker –l info –c 2
     - # It starts Celery worker on BDB app with Concurrency value = 2

- Open another Terminal, then execute below commands
  i) $ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>
  ii) $ source bin/activate
  iii) $ cd <path-to-python-predictive>
  iv) $ celery–A BizViz beat –l info
     - # It starts Celery beat Scheduler on BDB app

- Open another Terminal, then execute below commands
  i) $ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>
  ii) $ source bin/activate
  iii) $ cd <path-to-python-predictive>
  iv) $ python manage.py runserver IP:PORT
     - E.g., $ python manage.py runserver 192.168.1.9:8000

  Note: If running it shows error like "ModuleNotFoundError" then that means any python the package is missing.

2.2.4.5. Creating Django & Celery Services
Create services to work with Django Server, Celery Workers, Celery Scheduler. In this section, we create Linux OS based services that we can use to start/stop the Django server and Celery Workers. We can also use these services to know the current status of Django Server and Celery Workers.
- At first, create django.service, celery.service & celerybeat.service in '/etc/system/system/'

Note: Please take care of **User, Group, Working-Directory** and paths inside commands while configuring.

Please edit the above-created files as given below,

 # django.service

[Unit]

Description=Django Service

After=network.target

[Service]

Type=simple

User=ubuntu

Group=ubuntu

Restart=on-failure

WorkingDirectory=/home/ubuntu/venv/PA_Python/bizviz_3.5/python-predictive

ExecStart=/bin/sh -c '/home/ubuntu/venv/bin/python manage.py runserver --noreload 172.31.42.225:8000'

[Install]

WantedBy=multi-user.target

 # celerybeat.service

[Unit]

Description=Celery Beat Scheduler

After=network.target

[Service]

Type=simple

User=ubuntu

Group=ubuntu

WorkingDirectory=/home/ubuntu/venv/PA_Python/bizviz_3.5/python-predictive

**ExecStart=/bin/sh -c '/home/ubuntu/venv/bin/celery -A BizViz beat **

 **--pidfile=/home/ubuntu/venv/PA_Python/celery/beat.pid **

 --logfile=/home/ubuntu/venv/PA_Python/celery/beat.log   --loglevel=INFO'

[Install]

WantedBy=multi-user.target

 # celery.service

[Unit]

Description=Celery Service

After=network.target

[Service]

Type=forking

User=ubuntu

Group=ubuntu

EnvironmentFile=-/etc/conf.d/celery

WorkingDirectory=/home/ubuntu/venv/PA_Python/bizviz_3.5/python-predictive

**ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} **

  **-A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} **

 --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

 **ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} **

 --pidfile=${CELERYD_PID_FILE}'

 **ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} **

 **-A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} **

 --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

 [Install]

 WantedBy=multi-user.target

Note:

a. Please provide details for below variables as per your system in these service files,

- User – Your System's Username
- Group – Groups' Name which can access this service
- WorkingDirectory – The path of python-predictive Directory present in your system
- Please check the command directory and Server Address in '**ExecStart**,' '**ExecStop**,' and '**ExecReload**'

b. For celery.service, we need one more file that will be used for its worker's environment as it will contain all the required data for celery worker to start and work accordingly.

c. Create 'celery' in '/etc/conf.d/' and write in the file as given below,

```
# celery

# Name of nodes to start

# here we have a single node

CELERYD_NODES="CeleryNode"

# or we could have three nodes:

#CELERYD_NODES="w1 w2 w3"


# Absolute or relative path to the 'celery' command:

CELERY_BIN="/home/ubuntu/venv/bin/celery"

#CELERY_BIN="/virtualenvs/def/bin/celery"


# App instance to use

# comment out this line if you do not use an app

CELERY_APP="BizViz"

# or fully qualified:

#CELERY_APP="proj.tasks:app"


# How to call manage.py

CELERYD_MULTI="multi"


# Extra command-line arguments to the worker

CELERYD_OPTS="--concurrency=8"


# - %n will be replaced by the first part of the node name.

# - %I will be replaced with the current child process index

#   and is significant when using the prefork pool to avoid race conditions.

CELERYD_PID_FILE="/home/ubuntu/venv/PA_Python/celery/%n.pid"

CELERYD_LOG_FILE="/home/ubuntu/venv/PA_Python/celery/%n%I.log"

CELERYD_LOG_LEVEL="INFO"
```

Note: Please check for path details as per your system in celery file. Now run '**sudo systemctl daemon-reload.**' Till now all systemd files are created.

- To Start any service
    i) sudo systemctl start service name

- To Stop any Service
  - i) sudo systemctl stop service name

- To know the status of any service
  - i) sudo systemctl status service name

The service name will be given as you have created above. E.g., 'django.service,' 'celery.service,' and 'celerybeat.service.'

Note: Either run the Django Server and Celery workers using the commands (that is stated in Point No. 5 'Start-up the Django Server') or these services. We recommend our users to use the service method.

2.2.4.6. Stopping Karaf
open Karaf console using these commands:
- $ cd /<path to karaf>/karaf/bin/
- $ sudo ./karaf start

Once you see Karaf console, list all Karaf instances.

- instance: list

After listing instances, connect to all instances one by one and deploy respective bundles. Users need to uninstall the bundles if they are already deployed. Use the following steps for the same:
Once child instance console is open list the existing bundles using **list** command. It will show you all the bundles.

To uninstall bundle, you can use uninstall command.

- uninstall <bundle Id>   # To Uninstall Single Bundle
- uninstall <bundle Id start - bundle Id end>   # To Uninstall Multiple Bundles

Logout from the current instance using the '**Logout**' command. Users need to follow the same procedure for all other nodes.

2.2.4.7. Stopping Tomcat
Stop Tomcat, if already running.

- $ cd /home/tomcat
- $ ./bin/catalina stop

Note: Try the following URLs on your browser to check whether Tomcat is running or not

http://<IP>:<Port>/BizVizEP/services

http://<IP>:<Port>/app/

After stopping tomcat, clean work directory and existing war files.

- $ sudo rm -rf <path to tomcat>/work/Catalina/localhost/*
- $ sudo rm -rf <path to tomcat>/webapps/BizVizEP.war
- $ sudo rm -rf <path to tomcat>/webapps/BizVizEP/
- $ sudo rm -rf <path to tomcat>/webapps/app.war
- $ sudo rm -rf <path to tomcat>/webapps/app/

After cleaning tomcat kill the Java process running for Tomcat by using the following commands:

- $ ps -aux | grep java

It will show you a list of Java processes running on the system, then find the Tomcat process moreover, kill it using the 'kill' command.

- $ kill <Process Id

2.2.4.8. Starting Tomcat

Now copy the UI and BizVizEP war files inside "webapp" folder (/apache tomcat7/webapps) of tomcat and start tomcat and see the URL and check whether Tomcat has begun or not.

- $ cd /home/tomcat/
- $ ./bin/catalina start

You can also see the logs of tomcat.

- $ tail -f <path to tomcat>/logs/catalina.out

Note: You can put either put UI and BizVizEP war files in the same Tomcat (using one Tomcat for both) or two separate Tomcats.

2.2.4.9. Starting Karaf

After Tomcat, you need to start Karaf instance nodes, for that start Karaf and deploy the respective bundles in each instance using the following steps:

- instance:list   # It will list all instances of Karaf.
- instance: start instance_name   # It will start "instance_name" instance of Karaf
- instance: connect instance_name   # It will connect with "instance_name" of Karaf

Install all required bundles by using the following command once users see the karaf console:

- bundle:install -s file:/<Path to the folder containing .jar file>/<name of jar file>.jar

Users need to run these commands for each bundle. Users can log out from the current instance after deploying it. Users need to trail the above steps for each instance.

The list of bundles required for each instance of PA is given below:
Node: - Main Node

- com.bdbizviz.rs.base
- com.bdbizviz.audittrail
- com.bdbizviz.bizvizcassandranativeconnector
- com.bdbizviz.bizvizelasticsearch
- com.bdbizviz.bizvizfileconnector
- com.bdbizviz.bizvizmssqlconnector
- com.bdbizviz.bizvizmysqlconnector
- com.bdbizviz.bizvizoracleconnector

- com.bdbizviz.bizvizscheduler
- com.bdbizviz.bizvizschedulerhistory
- com.bdbizviz.bizvizsettings
- com.bdbizviz.camel.context
- com.bdbizviz.camel.websocket
- com.bdbizviz.csvWriter
- com.bdbizviz.datamanagement
- com.bdbizviz.datamanagementbase
- com.bdbizviz.dataservice.cassandranative
- com.bdbizviz.dataservice.mssql
- com.bdbizviz.dataservice.mysql
- com.bdbizviz.dataservice.oracle
- com.bdbizviz.datatypedefinition
- com.bdbizviz.filebase
- com.bdbizviz.fileupload
- com.bdbizviz.filter
- com.bdbizviz.formula
- com.bdbizviz.jdbcwriter
- com.bdbizviz.jsonwriter
- com.bdbizviz.mailservice
- com.bdbizviz.normalization
- com.bdbizviz.osgi.session
- com.bdbizviz.pa
- com.bdbizviz.pa.audittrail
- com.bdbizviz.pa.cassandra.native
- com.bdbizviz.pa.router
- com.bdbizviz.pa.wrapper.datapreparation
- com.bdbizviz.pa.wrapper.datareaderprocess
- com.bdbizviz.pa.wrapper.datawriter
- com.bdbizviz.predictivebase
- com.bdbizviz.rs.bizvizapi
- com.bdbizviz.rs.bizvizplugin
- com.bdbizviz.rs.dbase
- com.bdbizviz.rs.services
- com.bdbizviz.sample
- com.bdbizviz.thirdpartyauth
- com.bizviz.pa.rcache.cleaner
- com.bizviz.pa.rengine

Node: - PA Scheduler Node

- com.bdbizviz.rs.base
- com.bdbizviz.filebase
- com.bdbizviz.predictivebase
- com.bdbizviz.rs.dbase
- com.bdbizviz.datamanagementbase
- com.bdbizviz.rs.bizvizplugin
- com.bdbizviz.rs.bizvizapi
- com.bdbizviz.rs.services
- com.bdbizviz.camel.context

- com.bdbizviz.bizvizreportingservice
- com.bizviz.pa.rengine
- com.bizviz.pa.rcache.cleaner
- com.bdbizviz.pa
- com.bdbizviz.fileupload
- com.bdbizviz.filter
- com.bdbizviz.datatypedefinition
- com.bdbizviz.formula
- com.bdbizviz.jdbcwriter
- com.bdbizviz.sample
- com.bdbizviz.normalization
- com.bdbizviz.pa.router
- com.bdbizviz.pa.wrapper.datapreparation
- com.bdbizviz.pa.wrapper.datareaderprocess
- com.bdbizviz.pa.wrapper.datawriter
- com.bdbizviz.pdfbuilder
- com.bdbizviz.pa.cassandra.native
- com.bdbizviz.mail.service
- com.bdbizviz.pa.scheduler.manager
- com.bdbizviz.bizvizelasticsearch
- com.bdbizviz.bizvizmonitor.node
- com.bdbizviz.bizvizscheduler
- com.bdbizviz.bizvizschedulerhistory
- com.bdbizviz.datamanagement
- com.bdbizviz.bizvizmysqlconnector
- com.bdbizviz.dataservice.mysql
- com.bdbizviz.dataservice.mysql
- com.bdbizviz.dataservice.mssql
- com.bdbizviz.dataservice.oracle

Node: - ActiveMQ Node

This node does not need any bundle, only features are required by this node which is already installed in prebuilt Karaf for BizViz.
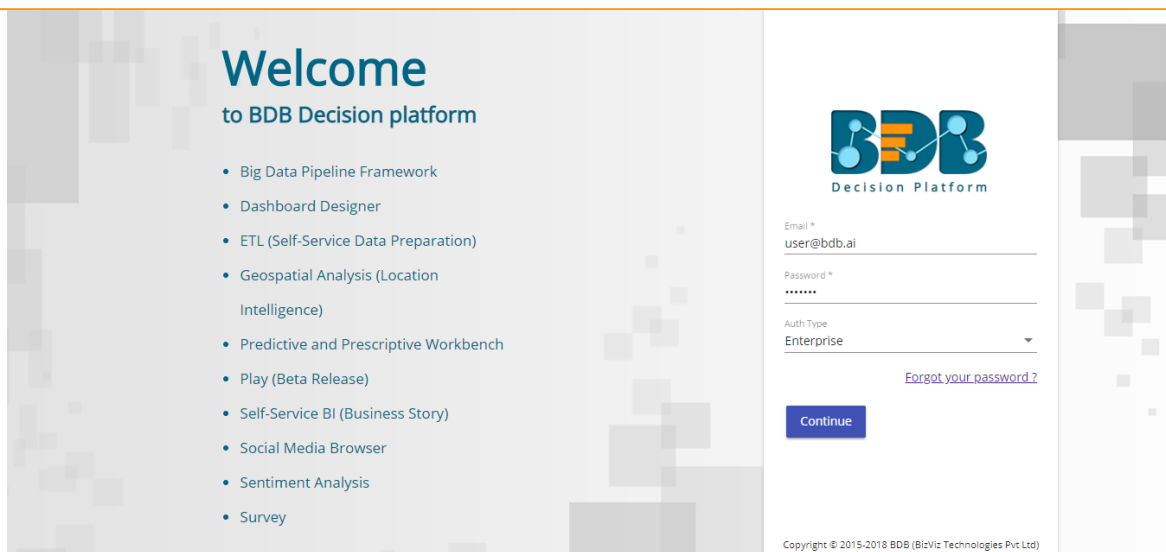
Use the following URL to check whether Karaf is started or not:
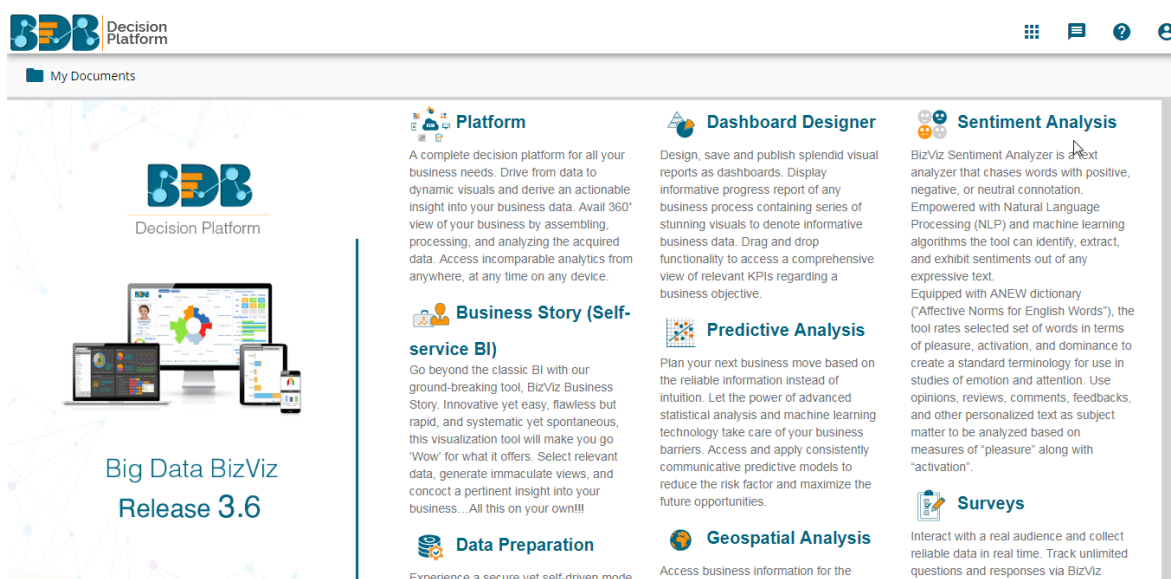
http://<IP>:<Port>/cxf

3. Getting Started with the BDB Predictive Workbench

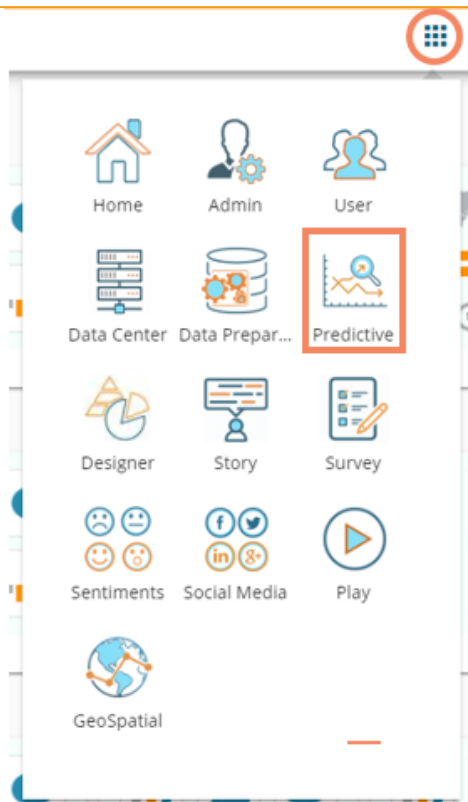BizViz Predictive analysis is a plugin application provided by BizViz Platform.

i)   Open BDB Enterprise Platform Link: https://app.bdb.ai
ii)   Enter your credentials to log in to the platform.
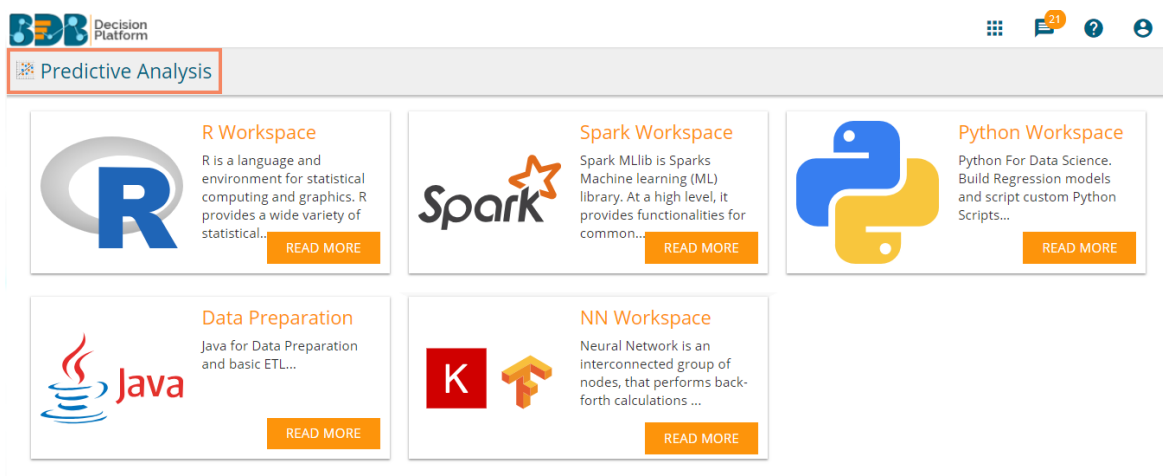iii) Click '**Continue**'

iv) BDB Platform homepage opens.

v) Click the '**Apps**' ⠿ menu icon.
vi) Select '**Predictive**' plugin from the Apps menu.
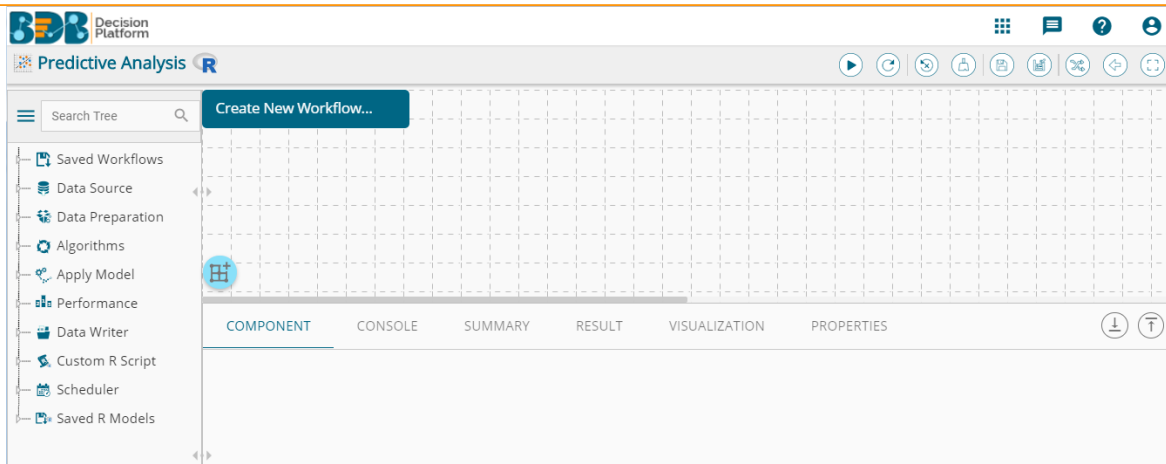
vii) Users will be redirected to the following page to select a workspace:

viii) Click on a Workspace to access the workspace-specific landing page

ix) The following is the landing page displayed for the R Workspace:

3.1.1. Forgot Password Option

Users are provided with a choice to change the password on the Login page of the platform.

    i)   Navigate to the Login page
    ii)   Click '**Forgot your password?**' option

    iii)   Users get redirected to a new window
    iv)   Provide the email id that is registered with BDB to send the reset password link
    v)   Click the '**Continue**' option

Having trouble signing in?
To reset your password, enter the email address you use to sign in to BizViz. This can be your email address associated with your account.
Email *

Sign in

Continue

vi) Users may be redirected to select a space in case of multiple spaces under one server link; they need to choose a space and click the 'Continue' option once again. Otherwise, a message will pop-up to notify that the password reset link has been sent to the registered email.

Password reset Link has been sent to your mail.

vii) Click the link from your registered email
viii) Users get redirected to the 'Reset Password' page to set a new password
ix) Set a new password
x) Confirm the newly set password
xi) Click the '**Continue**' option

Reset Password
You've confirmed ownership of the BizViz Account, Reset your password now to regain access.
New Password *

Confirm New Password *

Continue

xii) The password is successfully reset for the selected BDB account

Note: The 'Force Login' functionality has been introduced to control the number of active sessions up to three. Users can access only 3 sessions at a time when try to access 4th session a warning message displays to inform that the user has consumed the permitted sessions and a click on the 'Force Login' would kill all those active sessions.
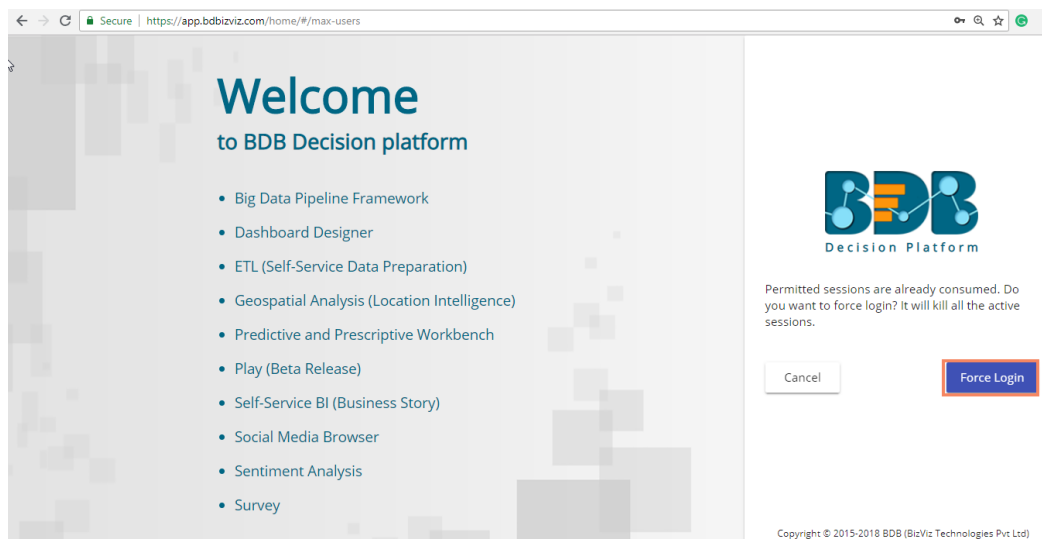

4. Overview of the Predictive Workspace(s)

This section describes all the options and icons provided on the landing page of the different Predictive Workspaces. The landing page of any selected Predictive Workflow can be described in the following Menus:

4.1. Tree-node Menu

The Tree-node menu has all the available component connectors to run a predictive execution. The components will be provided in the hierarchical order via a tree structure menu. All the main categories are included as tree-nodes and sub-categories are committed as petals to the respective tree-nodes.

E.g. The following image displays the R Workspace landing page where 'Data Writer' is the main category to which 'File Writer' is committed as a subcategory and 'CSV Writer' is displayed at the second level of the hierarchy.
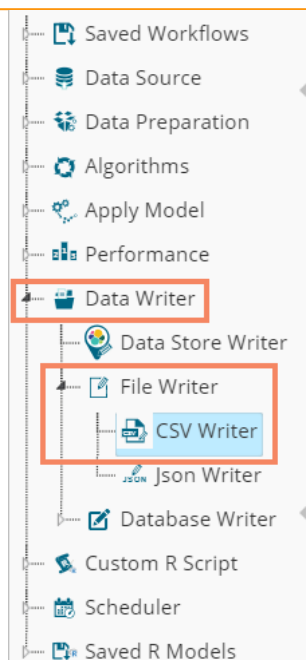
Note:
   a. The '**Search**' option has been provided for the entire tree structure menu.
   b. Click the '**Menu**' ☰ option next to the '**Search**' box to collapse the tree structure menu from the homepage.
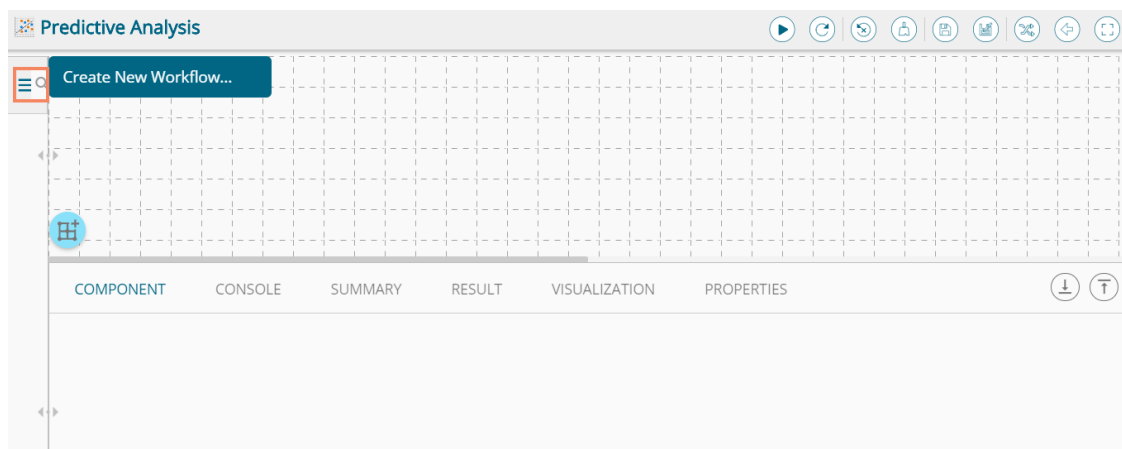
   c. Users are provided with an icon ⊞ to show or hide the grid lines on the workspace
   d. Users can use the scrolling icons to increase or decrease horizontal space for the Tree Menu

e. This document is created focusing on each petal of the tree structure menu. All the available major and minor categories are described at length to understand a Predictive process.
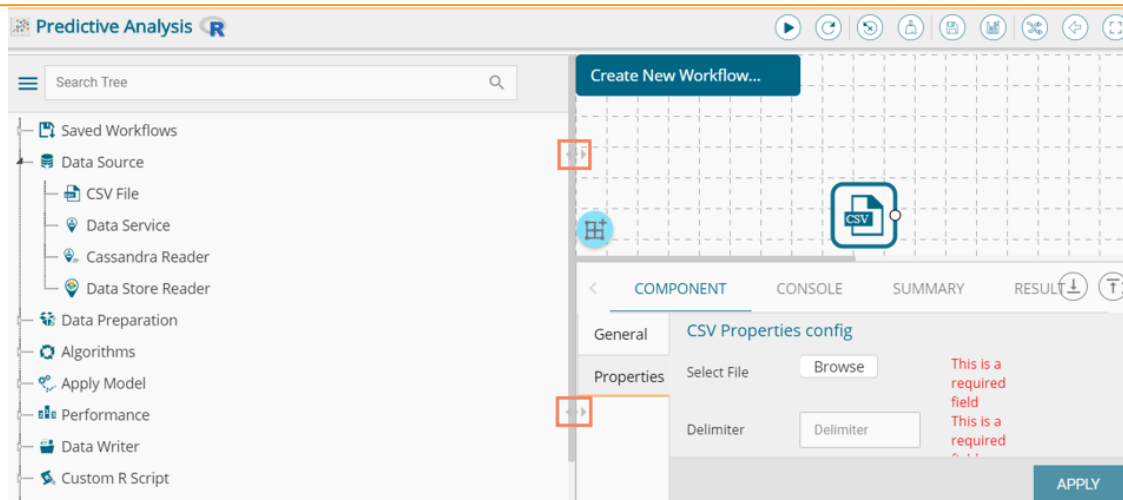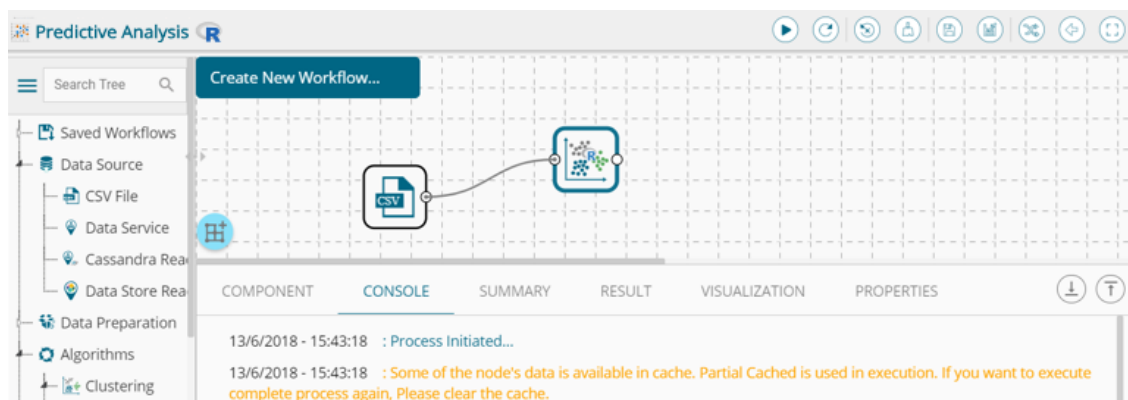
4.2. Header Menu-Options

1. **Run:** Click '**Run**' ▶ option to run the process and display the result set view. This option can be applied to the data source, algorithms, and data preparation components.

2. Refresh: The '**Refresh**' ↻ option is provided on the clear the cache memory, and it will run the component/ workflow.

3. **Reset:** Click the '**Reset**' ⊗ option to clean the workspace removing the current component connectors.

4. **Clear Cache:**
   a. After using the '**Run**' option, by default data will be cached in the server for the next 10 minutes. For the latest results, users need to rerun the workflow.
   b. Users need to click the '**Clear Cache**' 🔒 option to remove the cached data before running the workflow (again).
   c. If users change any component parameter which is to be applied to fetch the result then, '**Clear Cache**' option must be clicked.

   If you get a message to clear cache to execute your process, follow the below given steps:

   i) Click '**Clear Cache**' 🔒 option from the header menu
   ii) A message appears to confirm

iii) Click '**OK**'
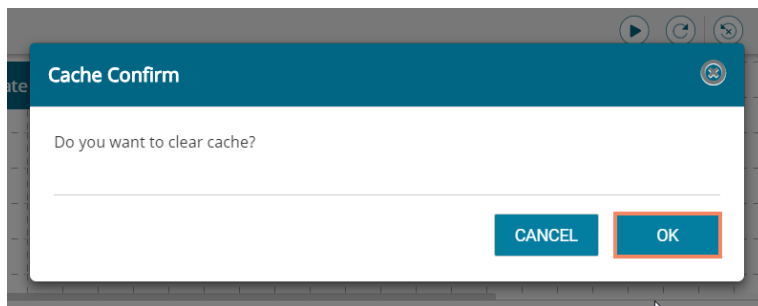
iv) Another message will pop-up to confirm that the cache data has been cleared.

R and Karaf Cache is successfully cleaned
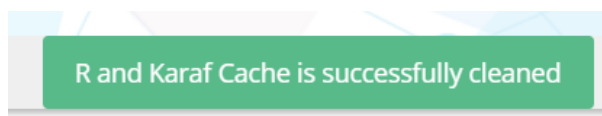
5. **Save:** Click the '**Save**' option to save the created predictive workflow.)

   i) Create a workflow by connecting various configured components.
   ii) Click '**Save**' icon from the landing page header menu
   iii) A new window appears to confirm the action
        a. Provide a Workflow Name
        b. Click '**SAVE**'

iv) A success message appears

workflow saved successfully
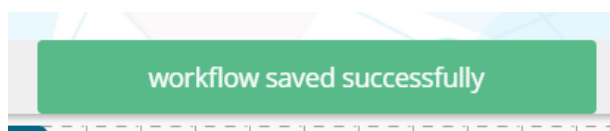
v) The selected workflow will be saved and added to the list of '**Saved Workflows**'
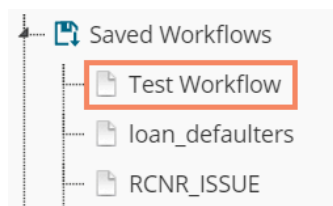
6. **Save As:** Click the '**Save As**' option to copy a predictive workflow with the desired name.

i) Create a workflow by connecting various configured components.
ii) Click '**Save As.**'
iii) A new window appears to confirm the task
    a. The Workflow Name will have the suffix '_1' by default (If wished, users can also modify the name of workflow manually)
    b. Click '**SAVE**'
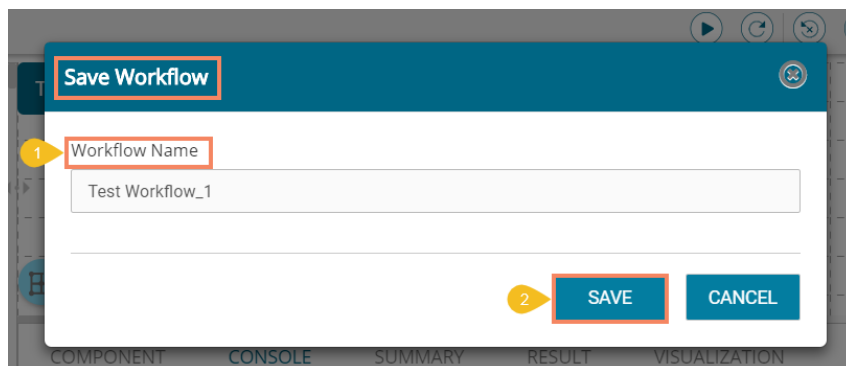
iv) A success message appears
v) The selected workflow gets saved by the new name in the '**Saved Workflows**' list
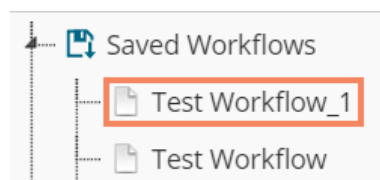
7. **Parallel Processing:** Users can enable parallel processing by using '**Parallel Processing**' ⚒ icon on the R landing page header. This option is only available for the R Workspace.
    a. Enable Parallel Processing option by a checkmark in the given box
    b. Provide No. of CPU Cores in the given space
    c. Click '**SAVE**'
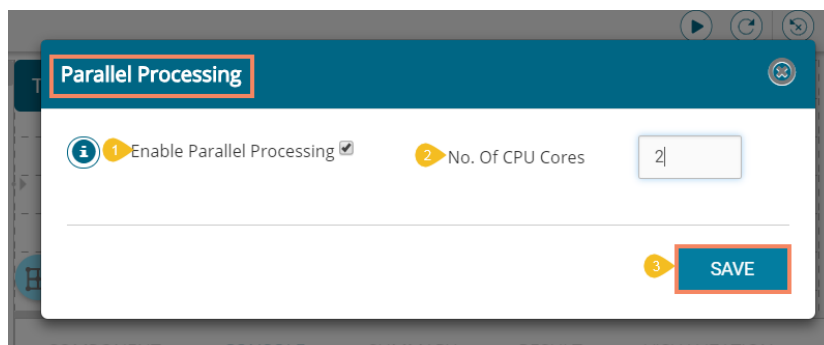
    d. The parallel processing will be enabled for the R Workspace

8. **Back:** Click the '**Back**' ⬅ icon to return on the Predictive landing page from any specific workspace.

9. **Full Screen/ Full-Screen Exit:** Click the '**Full Screen**' ⛶ icon to display the predictive landing page in the full screen.

After clicking once the same icon appears as '**Full-Screen Exit**' ⊕ and clicking it users can close the full-screen view of the predictive landing page (Users can also use '**Esc**' key to close the full-screen view)

4.3. Tabbed Menu Strip – Options

1. **Component:** The '**COMPONENT**' tab displays the required configuration fields for the dragged elements onto the workspace.
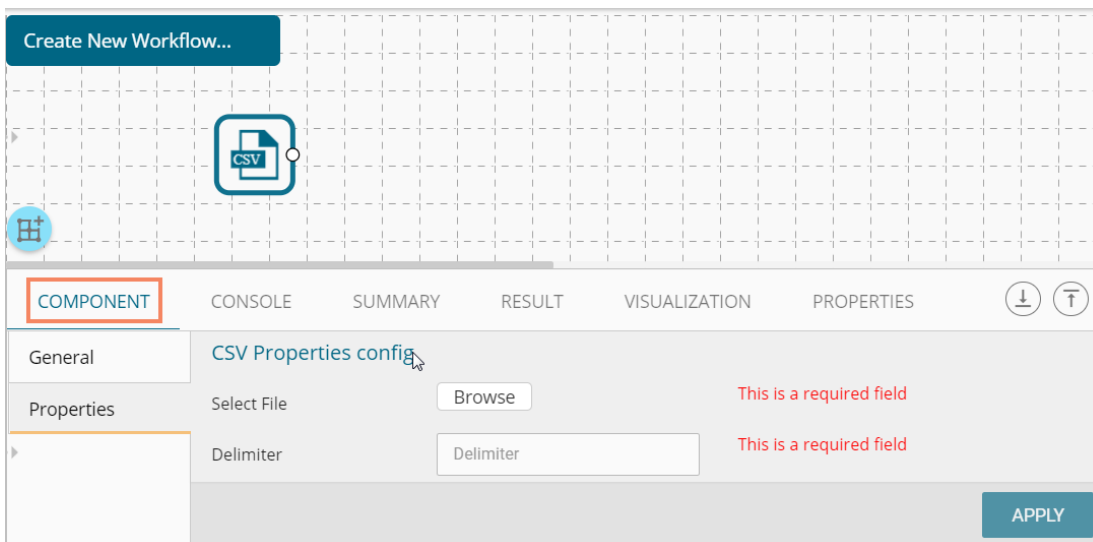
Note: The component tab may display various sub-tabs as per the selected components onto the workspace.

E.g., If the dragged data source is a CSV file, then the component tab will display General and Properties fields while for a Cassandra Reader as a data source, the component tab will display General, Properties, and Column Selection.

2. **Console:** The '**CONSOLE**' tab displays the date and time for the entire process.

    i)   Click on '**CONSOLE**' option.
    ii) The below-mentioned records will be displayed:

a. Process
b. Data Reader Process (starting and ending time)
c. R, Spark, and Python Process (starting and ending time)

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | |

13/6/2018 - 18:6:49 : Process Initiated...

13/6/2018 - 18:6:50 : dt24 started.

13/6/2018 - 18:6:50 : dt24completed.

13/6/2018 - 18:6:50 : CSV0 is started.

13/6/2018 - 18:6:52 : CSV0 is completed.

13/6/2018 - 18:6:52 : R Apply Model1 is started.

13/6/2018 - 18:6:53 : R Apply Model1 is completed.

13/6/2018 - 18:6:53 : R Apply Model4 is started.

13/6/2018 - 18:6:53 : R Apply Model4 is completed.

3. **Summary:** Click the **'SUMMARY'** tab to display the R and Spark Server overview of the process.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | |

```
************ Summary of All Stages ************
~~~~~~~~~~~~~ Summary of stage 1 ~~~~~~~~~~~~~~
------------ Summary of the model ------------

Columns used in the algorithm

        SepalLength     (double)
        SepalWidth      (double)
        PetalLength     (double)
        PetalWidth      (double)

K-means clustering with 5 clusters of sizes 27, 28, 28, 45, 22

Cluster means:
  SepalLength SepalWidth PetalLength PetalWidth
1    7.014815   3.096296    5.918519  2.1555556
2    5.242857   3.667857    1.500000  0.2821429
3    5.532143   2.635714    3.960714  1.2285714
4    6.264444   2.884444    4.886667  1.6666667
5    4.704545   3.122727    1.413636  0.2000000

Clustering vector:
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
   2   5   5   5   2   2   5   2   5   5   2   5   5   5   2   2   2   2   2   2
  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
   2   2   5   2   5   5   2   2   2   5   5   2   2   2   5   5   2   2   5   2
```

4. **Result:** Click the **'RESULT'** tab to display a result list view based on the selected execution.

SepalLength	SepalWidth	PetalLength	PetalWidth	Species	ClusterNumber2
5.1	3.5	1.4	0.2	setosa	2
4.9	3	1.4	0.2	setosa	5
4.7	3.2	1.3	0.2	setosa	5
4.6	3.1	1.5	0.2	setosa	5
5	3.6	1.4	0.2	setosa	2
5.4	3.9	1.7	0.4	setosa	2
4.6	3.4	1.4	0.3	setosa	5
5	3.4	1.5	0.2	setosa	2
4.4	2.9	1.4	0.2	setosa	5
4.9	3.1	1.5	0.1	setosa	5

Showing 1 to 10 of 150 entries   Previous   1   2   3   4   5   …   15   Next

Note: The '**Result**' tab will be displayed for the given data only after data is configured and the '**Run**' option has been selected. Up to 50000 cells can be displayed in the Result view.

5. **Visualization:** Click the '**VISUALIZATION**' tab to display a graphical representation of the result data.
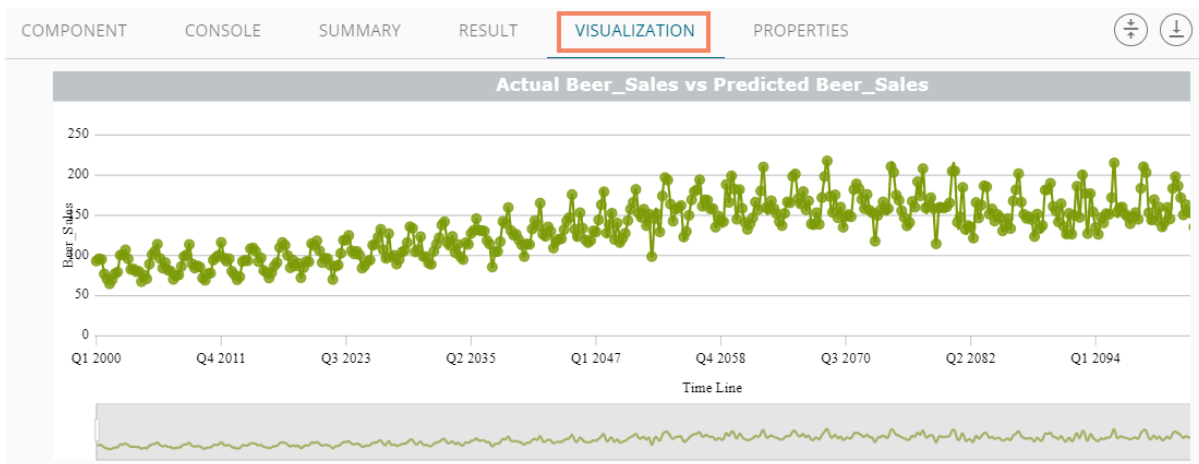
6. **Properties:** Click the '**PROPERTIES**' tab to display properties for the current workflow on the Workspace.

Created By	paadmin
Created At	2018-05-24 19:15:23 +0530
Last Modified By	paadmin
Last Modified At	2018-05-24 19:15:23 +0530
Version	3.5.0

7. **Status:** Click the '**STATUS**' tab to view the live job status of a running Spark job.

Note: The Status tab will appear when users need to check the live job status of a running job inside the Spark Workspace.

8. **Minimize Maximize Button:** The '**Minimize/Maximize**' buttons have been provided to the tabbed menu strip to customize the workspace and view space as per the user requirement.
The Predictive landing page default view displays the workspace canvass in the maximized space as shown below:
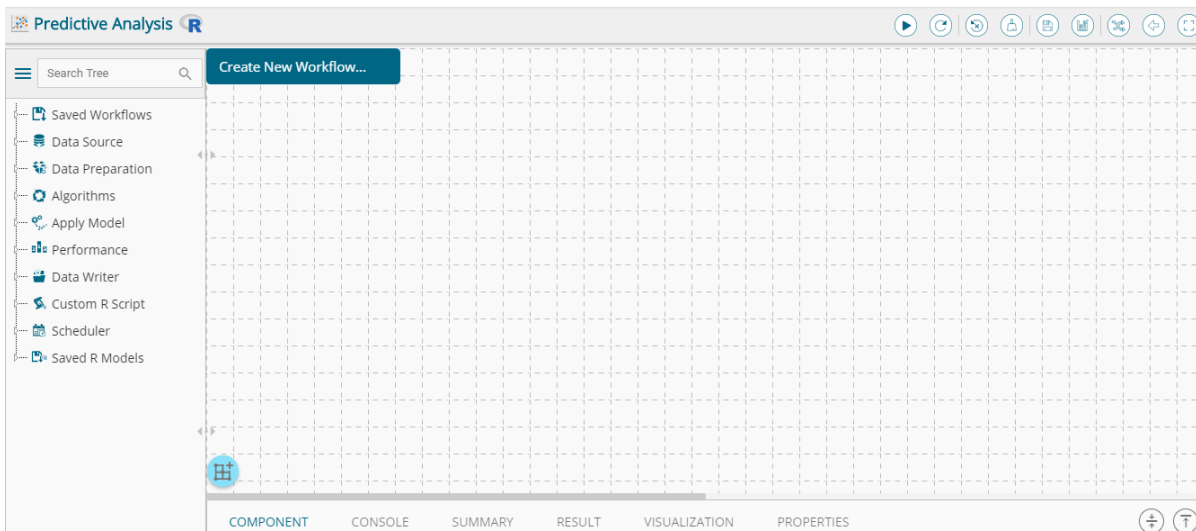
a. Click the '**Center**' icon to get equal space for the workspace and process view space on the Predictive landing page.
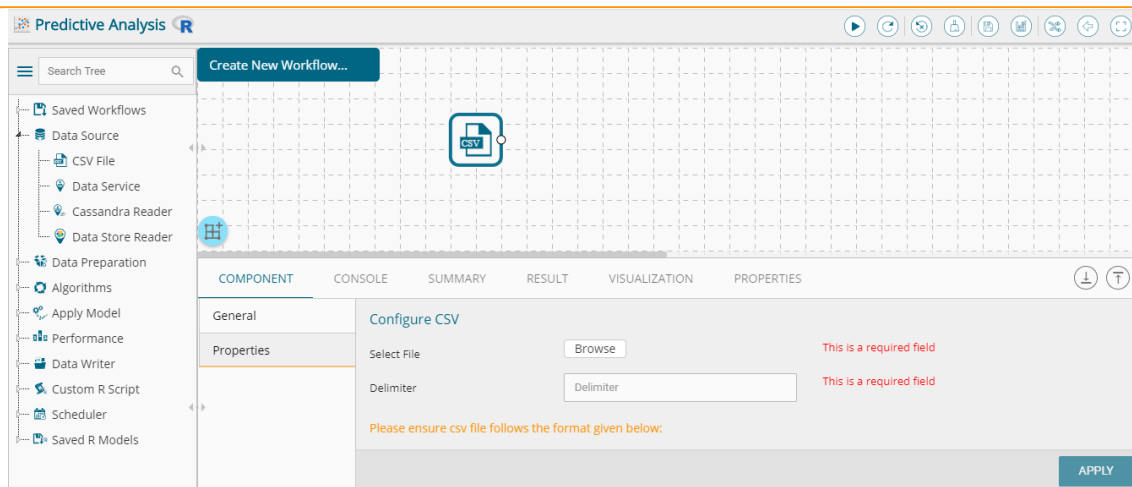
b. Click the '**Top**' ⊤ icon to maximize view space and minimize the workspace on the Predictive landing page.
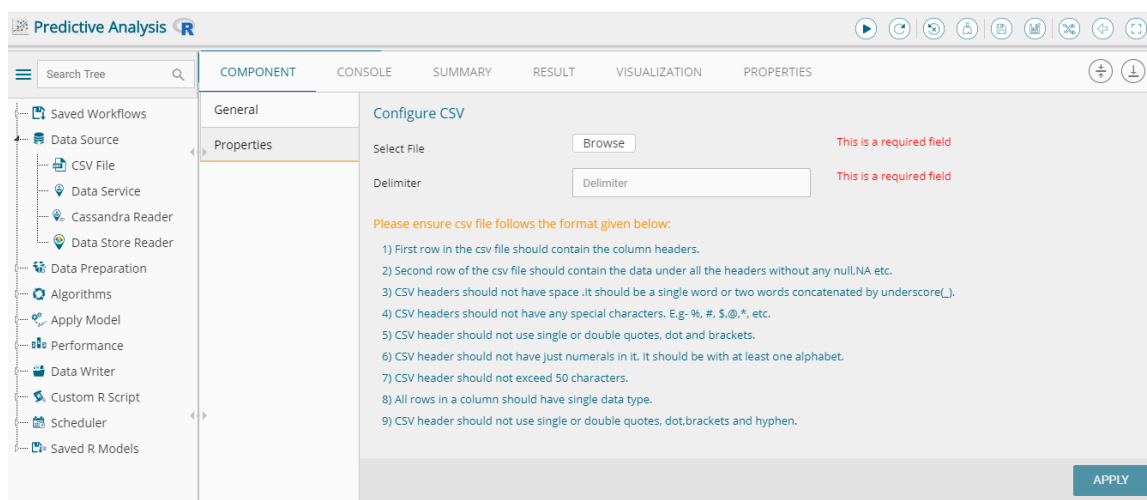
5. R Workspace

This section of the document describes all the components required to build an R workflow under the Predictive environment.

Users can select the R Workspace from the Predictive landing page to access the R Environment under the Predictive Workbench.

R Workspace

R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical...

READ MORE

Spark Workspace

Spark MLlib is Sparks Machine learning (ML) library. At a high level, it provides functionalities for common...

READ MORE

Python Workspace

Python For Data Science. Build Regression models and script custom Python Scripts...

READ MORE

Data Preparation

Java for Data Preparation and basic ETL...

READ MORE

NN Workspace

Neural Network is an interconnected group of nodes, that performs back-forth calculations ...

READ MORE

Users will be redirected to the following page by selecting the R Workspace:

5.1. Data Source

Acquiring data from a data source is the initial step in Predictive Analysis. The '**Data Source**' tree node offers three types of data connectors:

a. CSV File
b. Data Service
c. Cassandra Reader
d. Data Store Reader

5.1.1. Getting Data from a CSV File

i)   Select and drag '**CSV File**' component onto the workspace.
ii) Click the '**CSV File**' component.

iii) Configure the following **'CSV Properties Configuration'** fields:

    a. **Select File**: Browse a CSV file

    b. **Delimiter:** Mention the delimiter used in the CSV file

iv) Click **'APPLY'**

v) Users should get the **'Apply Successful'** message as displayed in the following image:

vi) Click the **'Run'** ▶ icon or click **'Refresh'** ↻ icon to run the workflow by clearing the previous cache

vii) Users will be redirected to the **'CONSOLE'** tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix) Follow the below given steps to display the result view:

    a. Click the dragged data source component on the workspace.

    b. Click the '**RESULT**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

Show 10 ▼ entries   Search: [____]

RowID	SLength	SWidth	PLength	PWidth
1	5.1	3.5	1.4	0.2
2	4.9	3	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

- **Rules to be followed while uploading a CSV File**
  1. The first row provided in the CSV file should contain the column headers.
  2. The second row of the CSV file should contain the data under all the headers without any 'null' or 'NA.'
  3. CSV headers should not have space. It should be a single word or two words concatenated by an underscore (_).
  4. CSV headers should not contain any special characters. E.g. - %, #, $, @,*, etc.
  5. CSV headers should not contain single or double quotes, dot, brackets, and high-fen.
  6. CSV headers should not contain merely numbers. Numerals should be used with at least one alphabet.
  7. CSV header should not exceed 50 characters.
  8. All rows in a column should have the same data type.

  Note:
      a. The supported file types will be .csv, .tsv
      b. '**General**' tab is provided to configure the following information for any tree-node component:

          i. Component Name: The predefined name of the component is displayed in this field
          ii. Alias Name:
          iii. Description (it is an optional field)
          (E.g. the following image displays '**General**' tab for a CSV data source.)

5.1.2. Getting Data from a Data Service

    i)   Select and drag '**Data Service**' component onto the workspace.

    ii)   Click the '**Data Service**' component.

    iii)   Users will be redirected to the '**Properties**' fields provided under '**Components**' tab on the Tabbed Menu Strip.

    iv)   Configure the '**Data Service Properties**':

        a. **Select Data Connector**: Select a data source from the drop-down menu

        b. **Select Data Service**: Select a query service from the drop-down menu

        c. **Fields**:

            The following tables will be displayed:

           i. Column Header

           ii. Data Type

    v)   Click '**NEXT**' (The '**NEXT**' option will appear only for the data service that has filters, otherwise the '**APPLY**' option will be displayed)

vi) Users will be redirected to the '**Conditions**' tab. (If the selected data service contains the filter values).

vii) Configure the following information:

    a. **Filter Type:** Available filter(s) in the data service will be displayed in this space.

    b. **Control Type:** Users are provided with the following options to pass the filter values under this option:

       • **Text:** By selecting this option users can manually enter multiple filter values separated by comma

       • **LOV:** By selecting this filter value option users will be directed to **choose** another Data Connector and Data Service available in the space

viii) Click '**APPLY**'

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ⟳ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

COMPONENT | CONSOLE | SUMMARY

13/4/2018 - 11:43:15   : Process Initiated...

13/4/2018 - 11:43:16   : Data Service0 is started.

13/4/2018 - 11:43:17   : Data Service0 is completed.

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:
   a. Click the dragged data source component on the workspace
   b. Click the '**RESULT**' tab

id	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.1	3.6	1.4	0.2	setosa
6	5.1	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

- **Rules to be Followed while Creating a Data Service**
  1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
  2. Data service header should not contain any special characters. E.g. - %, #, $, @,*, etc.
  3. Data service header should not contain single or double quotes, dot, brackets, and high-fen.
  4. Data service header should not contain merely numbers. Numerals should be used with at least one alphabet.
  5. Data service header should not exceed 50 characters.

Note:
   a. Users can develop a data service via the Data Management module of the BizViz Platform.
   b. '**Fields**' option under '**Properties**' tab will appear only after selecting the appropriate query service.
   c. LOV service provided under the '**Conditions**' tab can contain only one column, in case of more than one column, a warning message will appear.

d. Users can configure the following information for a data service data source via '**General**' tab:
i. Alias Name
ii. Description (it is an optional field)

5.1.3. Getting Data from a Cassandra Reader

i) Select and drag '**Cassandra Reader**' connector onto the workspace.
ii) Click on the '**Cassandra Reader**' connector.

iii) Users will be redirected to the '**Properties**' tab of the component.
iv) Configure the required properties:
a. Select Data Connector: Select a data connector using the drop-down menu
b. Host Name: Data connector specific hostname will be displayed
c. Port Number: Port number will be displayed
d. User Name: Username will be displayed
e. Password: Enter the password
f. Cluster Name: Enter a cluster name
g. Select Key Space: Select a keyspace from the drop-down menu
h. Select Table: Select a table from the drop-down menu
i. Limit No. of row to fetch: Select an option using the drop-down menu. Two options will be provided as shown below:
1. Select all Rows
2. Limit By
j. Max. No. of Rows to be fetched: Enter a number to decide maximum fetched rows. (This option will appear only if '**Limit By**' option has been selected using the '**Limit by Row**' field. The Default value for this field is 1000).
v) Click '**NEXT**'

vi) Users will be redirected to the '**Column Selection**' tab.

vii) Select the required columns from the list.

viii) Click '**APPLY**'.

ix) Click the '**Run**' ⊳ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:
    a. Click the dragged data source component on the workspace.
    b. Click the '**Result**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
6	1.7	0.4	5.4	3.9	0
80	3.5	1	5.7	2.6	1
75	4.3	1.3	6.4	2.9	1
57	4.7	1.6	6.3	3.3	1
113	5.5	2.1	6.8	3	1
67	4.5	1.5	5.6	3	1
118	6.7	2.2	7.7	3.8	1
82	3.7	1	5.5	2.4	1
120	5	1.5	6	2.2	1
112	5.3	1.9	6.4	2.7	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note: The Apache Spark workflows require a '**Cassandra Reader**' as a data source. The Cassandra Reader can also be used as a data source for the R Workflows.

5.1.4. Getting Data from a Data Store Reader

i) Select and drag '**Data Store Reader**' component onto the workspace
ii) Click on the '**Data Store Reader**' component

iii) Users will be redirected to the '**Properties**' tab of the component
iv) Configure the required properties:
    a. Select Data Store: Select a data store using the drop-down menu
    b. Limit No. of Documents to Fetch: Select an option using the drop-down menu. Two options will be provided as shown below:
        1. Fetch all Documents
        2. Limit By
    c. Max. No. of Documents to be Fetched: Enter a number to decide maximum fetched documents (This option will appear only if 'Limit By' option has been selected using the 'Limit No. of Documents to Fetch' field. Users can select any positive integer value).
v) Click '**NEXT**'

vi) Users will be redirected to the '**Conditions**' tab
vii) Select the required columns from the drop-down list
viii) Click '**APPLY**'

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
xii) Follow the below given steps to display the result view:
    a. Click the dragged data source component on the workspace
    b. Click the '**RESULT**' tab

Show 10 ▾ entries   Search:

SepalLength	SepalWidth	category
1	-0.32251082	SepalLength
-0.32251082	1	SepalWidth

Showing 1 to 2 of 2 entries   Previous   1   Next

Note: Empty values present in any row of the numeric column gets replaced with zero (0) while reading data from a data store reader.

5.1.5. Removing a Data Source from the Workspace

i)   Right-click on the data source connector (in the workspace)
ii)   A context menu appears
iii)   Click the **'Delete'** option

Reset

Delete
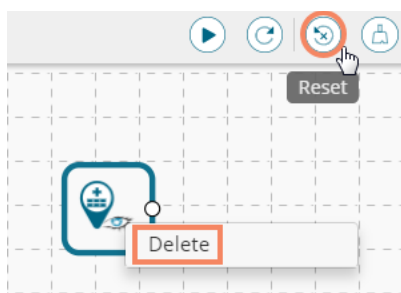
iv)   The selected Data Source component will be removed from the workspace
                        OR
Click on the **'Reset'** icon to remove the connector(s) from the workspace

Note: The same set of steps can be followed to remove any data source type in the given tree-node menu.

5.2. Data Preparation
Components provided under the **Data Preparation** tree-node help in preparing the raw data from the data source and make it suitable for analysis. They organize data to gain accurate result out of it.

5.2.1. Data Type Definition
The Data Type Definition option can be used to change the name, data type of the data source column. This component helps users to prepare data and make it suitable for further analysis.

i)   Navigate to the Predictive home page
ii)   Click '**Data Preparation**' tree-node
iii)   A context menu opens

iv) Drag '**Data Type Definition**' component and connect it to a configured data source onto the workspace.

v) Click the '**Data Type Definition**' component (in the workspace).

vi) Users will be redirected to the '**Properties**' tab.

vii) Configure the following 'Data Type Mapping' details:

a. **Column Name:** Select a column name which you want to change
b. **Alias Name:** Enter an alias name for the required source column
c. **Primary Data Type:** Select a primary data type column that you want to change
d. **Date Format:** Select a date format that you want to display (Date format is optional for date Data Type)

e. '**Add**' option ➕: Click on this button to add one more row of the '**Data Type Mapping**' fields

viii) Click '**APPLY**'.

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ⟳ icon to run the workflow by clearing the previous

cache

x)   Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

COMPONENT   CONSOLE   SUMMARY   RESULT

13/4/2018 - 17:47:14   : Process Initiated...

13/4/2018 - 17:47:15   : CSV1 is started.

13/4/2018 - 17:47:15   : CSV1 is completed.

13/4/2018 - 17:47:15   : Data Type Definition0 is started.

13/4/2018 - 17:47:16   : Data Type Definition0 is completed.

xi)   After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii)   Follow the below given steps to display the result view:

       a. Click the dragged Data Type Definition component in the workspace.

       b. Click the '**RESULT**' tab.

xiii)   Users can see the given column names on the selected columns in the '**RESULT**' data.

Number	SL	SepalWidth	PL	PetalWidth	Species
1	5.1	3.5	1	0.2	setosa
2	4.9	3	1	0.2	setosa
3	4.7	3.2	1	0.2	setosa
4	4.6	3.1	1	0.2	setosa
5	5	3.6	1	0.2	setosa
6	5.4	3.9	1	0.4	setosa
7	4.6	3.4	1	0.3	setosa
8	5	3.4	1	0.2	setosa
9	4.4	2.9	1	0.2	setosa
10	4.9	3.1	1	0.1	setosa

Showing 1 to 10 of 150 entries   Previous   1   2   3   4   5   ...   15   Next

5.2.2. Filter

This option is used to filter the data by column or row.

i)   Select and Drag '**Filter**' component onto the workspace.

ii)   Connect the '**Filter**' component to a configured data source component.
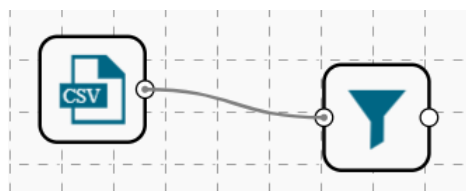
iii)   Configure the filter component as described below:

Column Filter

i) Select a column from the **'Selected Columns'** context menu.
ii) Click **'APPLY'** to configure the data.
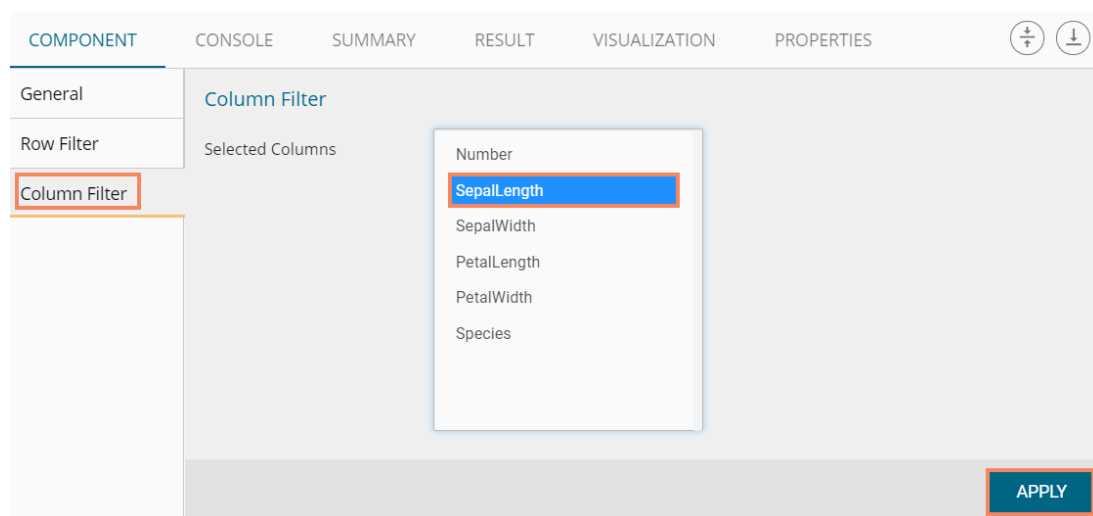
iii) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
iv) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process
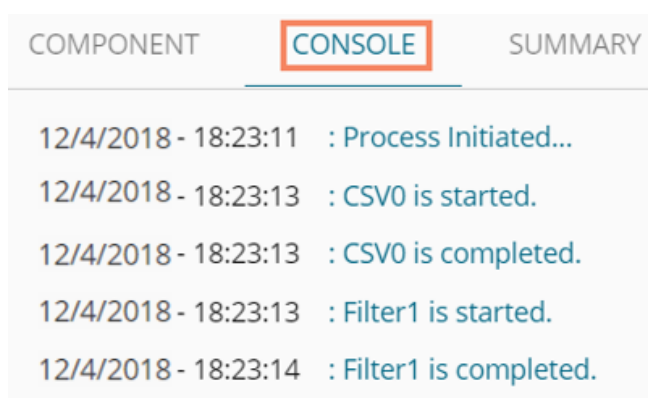
v) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
vi) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component in the workspace
   b. Click the '**RESULT**' tab
vii) The filtered data will be displayed via the '**RESULT**' tab

Show 10 ▼ entries   Search: [   ]

SepalLength
5.1
4.9
4.7
4.6
5
5.4
4.6
5
4.4
4.9

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Row Filter

i)   Drag and connect the **'Filter'** component onto the workspace
ii) Connect the **'Filter'** component to a configured data source
iii) Click the **'Filter'** component
iv) The **'Column Filter'** tab will be displayed (by default)
v)   Select a column using the context menu
vi) Select **'Row Filter'** tab from the **'Component'** menu list
vii) Configure the required fields:

   a. Double click on the components from **Columns, Operators,** and **Functions** in the sequence as shown in the image below
   b. A formula will be entered in the given box (E.g., in this case, the entered formula is [Number]>SELECT(2))
   c. Click '**APPLY**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

① Row Filter

Column Filter

Row Filter

[Number]>SELECT(2)

② Columns   **④ Functions**   **③ Operators**

Number

Functions:
MIN
AVERAGE
SUM
Data Manipulation functions
REPLACE
BLANK
SELECT
Conditional functions
IFELSECONDITION

Operators:
Equal to
Not Equal to
Greater than
Greater than or equal to
Less than
Less than or equal to
Multiply
Divide

⑤ APPLY

viii) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
ix) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

x) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xi) Follow the below given steps to display the result view:
   a. Click the dragged data preparation component on the workspace
   b. Click the '**RESULT**' tab

xii) The filtered data as per the applied formula will be displayed via the '**RESULT**' tab

Note:

a. The expression should retain Boolean output.
b. Users can not use Data manipulation functions.

5.2.3. Missing Value Replacement

Users can replace the missing data in the specified variable with the determined value. Users will be provided with a list of options that can be considered for replacement.

i) Drag a data source on the workspace, configure it, run it, and check the data using '**RESULT**' tab. (in this case, the selected input data is displayed in the following image)

Show 10 entries   Search:

SepalLength	SepalWidth	PetalLength	PetalWidth	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.5	1.4	0.2	setosa
4.7	3.5	1.3	0.2	setosa
4.6	3.5	1.5	0.2	setosa
	3.6	1.4	0.2	
	3.9	1.7	0.4	
	3.4	1.4	0.3	
	3.4	1.5	0.2	setosa
	2.9	1.4	0.2	setosa
	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

ii)   Select and drag '**Missing Value Replacement**' component onto the workspace.

iii) Connect the '**Missing Value Replacement**' component to a configured data source.

iv) Use the Right-click on the '**Missing Value Replacement**' component to configure.
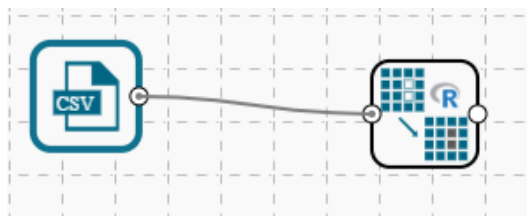
v)   Choose the replacement value by configuring the following fields:

  a. **Column Name**: Select a column using the drop-down that contains some missing values.

  b. **Replacement Options**: Select a replacement option using the drop-down menu. The following replacement options are provided under this field:

    1. Mean

    2. Median

    3. Mode

    4. Maximum

    5. Minimum

    6. Remove Entire Row

    7. Remove Entire Column

    8. Custom Replacement

vi) Click '**APPLY**'

vii) Click the '**Run**' ▶ icon or click '**Refresh**' ⟳ icon to run the workflow by clearing the previous cache

viii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

COMPONENT	CONSOLE	SUMMARY	RESULT

12/4/2018 - 19:15:17   : Process Initiated...

12/4/2018 - 19:15:18   : CSV0 is started.

12/4/2018 - 19:15:18   : CSV0 is completed.

12/4/2018 - 19:15:18   : Missing Data Replacement1 is started.

12/4/2018 - 19:15:19   : Missing Data Replacement1 is completed.

ix) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

x) Follow the below given steps to display the result view:
   a. Click the dragged data preparation component on the workspace
   b. Click the '**RESULT**' tab

xi) The missing values in the selected column will be substituted with the chosen replacement option (E.g., 7.9 is the Maximum value for the Sepal Length column)

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 entries   Search:

SepalLength	SepalWidth	PetalLength	PetalWidth	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.5	1.4	0.2	setosa
4.7	3.5	1.3	0.2	setosa
4.6	3.5	1.5	0.2	setosa
7.9	3.6	1.4	0.2	
7.9	3.9	1.7	0.4	
7.9	3.4	1.4	0.3	
7.9	3.4	1.5	0.2	setosa
7.9	2.9	1.4	0.2	setosa
7.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

5.2.4. Formula

Users can create a calculated column using '**Formula.**' A formula can be formed by using available columns, functions, and operators.

i) Select and drag '**Formula**' component onto the workspace
ii) Connect the '**Formula**' component to a configured data source
iii) Click on the '**Formula**' component
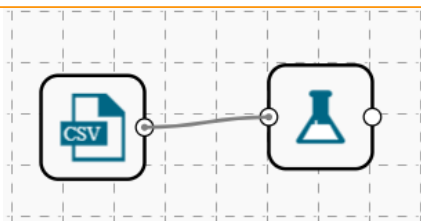
iv) Configure the required component fields to apply a formula:
   a. **'Columns,'** **'Functions,'** and **'Operators':** Double click on these lists will enter a formula in the given box.
   b. **Formula Name:** Enter a formula name in the given field.
   c. Click **'APPLY'** to configure the formula.

v) Click the **'Run'** ▶ icon or click **'Refresh'** ↻ icon to run the workflow by clearing the previous cache
vi) Users will be redirected to the **'CONSOLE'** tab to display the progress of the process

vii) After the Console process gets completed, users can view the result data using the **'RESULT'** tab
viii) Follow the below given steps to display the result view:
   a. Click the dragged data preparation component on the workspace
   b. Click the **'RESULT'** tab
ix) A new Formula column is added to the result data

Show 10 entries   Search:

| Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	FormulaColumn
1	5.1	3.5	1.4	0.2	setosa	1.45714285714286
2	4.9	3	1.4	0.2	setosa	1.63333333333333
3	4.7	3.2	1.3	0.2	setosa	1.46875
4	4.6	3.1	1.5	0.2	setosa	1.48387096774194
5	5	3.6	1.4	0.2	setosa	1.38888888888889
6	5.4	3.9	1.7	0.4	setosa	1.38461538461538
7	4.6	3.4	1.4	0.3	setosa	1.35294117647059
8	5	3.4	1.5	0.2	setosa	1.47058823529412
9	4.4	2.9	1.4	0.2	setosa	1.51724137931034
10	4.9	3.1	1.5	0.1	setosa	1.58064516129032

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

5.2.5. Normalization

This component controls the relevant data. It attempts to convert the available data from a larger Range to a smaller range. It can be done over numerical columns.

5.2.5.1. Min-Max Normalization

It implements a linear transformation of the original data values and sets a new range for all the data values to fit in. The user can fix the New Maximum and New Minimum Value for the data from the new field. Consequently, each value "v" from the original interval will be mapped into value "new_v" following the below-given formula:

$$new\_v = \frac{v - min_x}{max_x - min_x} \cdot (new\_max_x - new\_min_x) + new\_min_x$$

i) Select and drag **'Normalization'** component onto the Workspace.
ii) Connect the **'Normalization'** component to a configured data source.
iii) Click the **'Normalization'** component.

iv) Configure the following component fields:

Properties
a. **Column Selection**
   i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)
b. **Behavior**
   i. **Normalization Type:** Select **'Min-Max'** normalization type from the drop-down menu
   ii. **New Maximum:** Set a new maximum value (Default value for this field is 1)
   iii. **New Minimum:** Set a new minimum value (Default value for New Minimum field is 0)

v)   Click 'APPLY'

vi)   Click the 'Run' ▶ icon or click 'Refresh' ⟳ icon to run the workflow by clearing the previous cache

vii)   Users will be redirected to the 'CONSOLE' tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the 'RESULT' tab

ix)   Follow the below given steps to display the result view:
   a.   Click the dragged Formula component in the workspace.
   b.   Click the 'RESULT' tab.

Show [10 ▼] entries   Search: [   ]

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	22.2222222222222	3.5	1.4	0.2	setosa
2	16.6666666666667	3	1.4	0.2	setosa
3	11.1111111111111	3.2	1.3	0.2	setosa
4	8.33333333333333	3.1	1.5	0.2	setosa
5	19.4444444444444	3.6	1.4	0.2	setosa
6	30.5555555555556	3.9	1.7	0.4	setosa
7	8.33333333333333	3.4	1.4	0.3	setosa
8	19.4444444444444	3.4	1.5	0.2	setosa
9	2.77777777777779	2.9	1.4	0.2	setosa
10	16.6666666666667	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous   1   2   3   4   5   ...   15   Next

5.2.5.2. Zero-Score

This normalization also is known as **'Zero Mean Normalization'** is calculated on the **'mean'** and **'standard deviation'** for each attribute. It determines whether a specific value is above or below average. It also signifies the exact proportion of the variance from the fixed limit of aver3age. After applying **'Zero-Score'** normalization, each feature will have a mean value of zero (0). The unit of each value will be the number of (estimated) standard deviations away from the (estimated) mean. Zero score normalization may be sensitive to small values of 'σ_x' new value the **'new_v'** can be found by using the following expression:

$$new\_v = \frac{v - \mu_x}{\sigma_x}$$

i)   Select and drag **'Normalization'** component onto the Workspace
ii)   Connect the **'Normalization'** component to a configured data source
iii)   Click the **'Normalization'** Component
iv)   Configure the required component fields:

    Properties
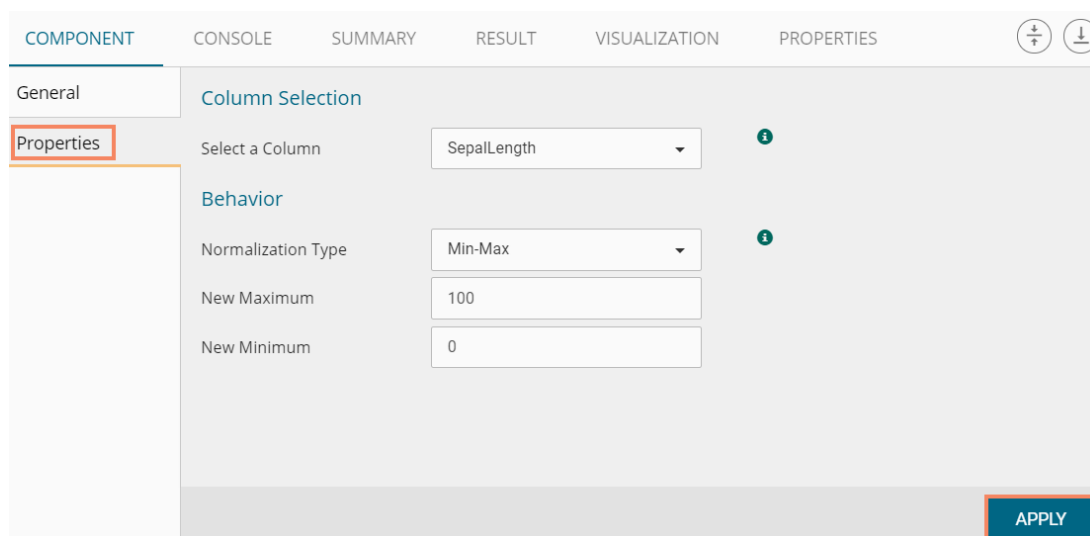   a. **Column Selection**
        i.   **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)
    b. **Behavior**
        i.   **Normalization Type:** Select **'Zero-Score'** normalization type from the drop-down menu
v)   Click **'APPLY'** to configure the fields.

General

Properties

Column Selection

Select a Column   [SepalLength ▼]

Behavior

Normalization Type   [, Zero-Score ▼]

APPLY

vi) Click the 'Run' ▶ icon or click 'Refresh' ↻ icon to run the workflow by clearing the previous cache

vii) Users will be redirected to the 'CONSOLE' tab to display the progress of the process

COMPONENT	CONSOLE	SUMMARY	RESULT

12/4/2018 - 15:18:4   : Process Initiated...

12/4/2018 - 15:18:5   : CSV0 is started.

12/4/2018 - 15:18:5   : CSV0 is completed.

12/4/2018 - 15:18:6   : Normalization1 is started.

12/4/2018 - 15:18:7   : Normalization1 is completed.

viii) After the Console process gets completed, users can view the result data using the 'RESULT' tab

ix) Follow the below given steps to display the result view:
  a. Click the dragged algorithm component in the workspace.
  b. Click the 'RESULT' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 ▼ entries   Search: [   ]

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	-0.897673879196766	3.5	1.4	0.2	setosa
2	-1.13920048346495	3	1.4	0.2	setosa
3	-1.38072708773314	3.2	1.3	0.2	setosa
4	-1.50149038986724	3.1	1.5	0.2	setosa
5	-1.01843718133086	3.6	1.4	0.2	setosa
6	-0.535383972794483	3.9	1.7	0.4	setosa
7	-1.50149038986724	3.4	1.4	0.3	setosa
8	-1.01843718133086	3.4	1.5	0.2	setosa
9	-1.74301699413542	2.9	1.4	0.2	setosa
10	-1.13920048346495	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

5.2.5.3. Decimal-Scaling

The decimal point of the value of each element is moved in accord with its maximum absolute value. A modified value 'new_v' can be obtained using the following formula:

$$new\_v = \frac{v}{10^c}$$

Note: In the decimal-scaling expression 'c' is the smallest integer so that max(new_v) < 1.

i) Select and drag 'Normalization' component onto the Workspace.
ii) Connect the 'Normalization' component to a configured data source.
iii) Click the 'Normalization' Component.
iv) Configure the required component fields:

Properties
  a. **Column Selection**
      i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected).
  b. **Behavior**
      i. **Normalization Type:** Select '**Decimal Scaling**' normalization type from the drop-down menu.
  v) Click '**Apply**' to configure the fields:

  vi) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
  vii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

  viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
  ix) Follow the below given steps to display the result view:
      a. Click the dragged data preparation component on the workspace
      b. Click the '**RESULT**' tab

Show 10 ▼ entries   Search: [   ]

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	0.51	3.5	1.4	0.2	setosa
2	0.49	3	1.4	0.2	setosa
3	0.47	3.2	1.3	0.2	setosa
4	0.46	3.1	1.5	0.2	setosa
5	0.5	3.6	1.4	0.2	setosa
6	0.54	3.9	1.7	0.4	setosa
7	0.46	3.4	1.4	0.3	setosa
8	0.5	3.4	1.5	0.2	setosa
9	0.44	2.9	1.4	0.2	setosa
10	0.49	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

Note:

   a. Normalization displays columns containing only numerical data.

   b. **'New Maximum Value'** must be greater than **'New Minimum Value.**

5.2.6. Sample

This component can be used to select a subsection of data from a large dataset. The sample component supports the following sample types:

5.2.6.1. Sampling Methods

1. **First N:** It will select the first N records from the data source. E.g., If the chosen value for "N" is 10, then it will select the first ten records from the data.

2. **Last N:** It will select the last N records from the data source. E.g., If the chosen value for "N" is 5, then it will select the last five records from the data.

3. **Every Nth:** It will select every Nth record from the data source, wherein "N" indicates an interval. E.g., If N=3, then 3rd, 6th, and 9th records will be selected from the data.

4. **Simple Random:** It will select records randomly as per the value of "N" or percentage mentioned for "N" from the data source. E.g., If the selected value for "N" is four then, it will select randomly any four records from the data source. If the selected value for "N" is 4% then, it will select 4% of records from the data source.

5. **Systematic Random:** It will select data based on the bucket size. E.g., If the chosen value for the bucket is two then, it will select 1st, 3rd, 5th records or 2nd, 4th, 6th records from the data source.

5.2.6.2. Steps to Apply a Sampling Method

i)   Select and drag **'Sample'** component onto the workspace

ii)   Connect the **'Sample'** component to a configured data source

iii) Click the **'Sample'** component

iv) Configure the required component fields:
Properties
a. Sampling Information
  i. **Sampling Type**: Select an option from the drop-down menu
  ii. **Limit Rows by** Select an option from the drop-down menu. This field will offer two options as described below:
    1. **Numbers of Rows**: By selecting this option, it will display a new field '**Number of Rows.**'
    2. **Percentage of Rows**: By selecting this option, it will display the new field '**Percentage of Rows.**'
b. Sample Size Limit
  i. **Maximum Rows**: The maximum number of rows that can be viewed in the 'RESULT' tab (It is an optional field)
v) Click '**APPLY**'

vi) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
vii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
ix) While accessing the '**RESULT**' tab, Users will be displayed as a result view based on the selected Sampling Type

5.2.6.3. Result View for the Available Sampling Methods

1. **First N (Where 'N' is 1 number of row)**

2. **Last N ('N' is 5% and maximum rows are 6)**

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
136	7.7	3	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6	3	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica

Showing 1 to 7 of 7 entries   Previous   1   Next

3. **Every Nth (Interval is 3, and the maximum rows are 7)**

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

General

Properties

Sampling Information

Sampling Type   Every Nth ▼

Step Size   3

Sample Size Limit

Maximum Rows   7

APPLY

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
7	4.6	3.4	1.4	0.3	setosa
10	4.9	3.1	1.5	0.1	setosa
13	4.8	3	1.4	0.1	setosa
16	5.7	4.4	1.5	0.4	setosa
19	5.7	3.8	1.7	0.3	setosa

Showing 1 to 7 of 7 entries   Previous   1   Next

4. **Simple Random (the 'Number of Rows' are 3). The randomly selected any three rows will be displayed.**

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Sampling Information

Sampling Type | Simple Random

Limit Rows by | Number of Rows

Number of Rows | 4

Sample Size Limit

Maximum Rows | 10

APPLY

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 entries

Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
65	5.6	2.9	3.6	1.3	versicolor
72	6.1	2.8	4	1.3	versicolor
96	5.7	3	4.2	1.2	versicolor
109	6.7	2.5	5.8	1.8	virginica

Showing 1 to 10 of 10 entries

Previous | 1 | Next

5. **Systematic Random (Bucket Size is 3).**

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Sampling Information

Sampling Type | Systematic Random

Bucket Size | 3

Sample Size Limit

Maximum Rows | 10

APPLY

Show 10 ▼ entries   Search:

| Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
2	4.9	3	1.4	0.2	setosa
5	5	3.6	1.4	0.2	setosa
8	5	3.4	1.5	0.2	setosa
11	5.4	3.7	1.5	0.2	setosa
14	4.3	3	1.1	0.1	setosa
17	5.4	3.9	1.3	0.4	setosa
20	5.1	3.8	1.5	0.3	setosa
23	4.6	3.6	1	0.2	setosa
26	5	3	1.6	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa

Showing 1 to 10 of 10 entries   Previous   1   Next

5.2.7. R Split Data

The R Split Data component is used to split a dataset into training and testing per percentage and method. Once the most suitable model is decided from the trained data, users can pass test data to validate the model.

R Split Data appears as a leaf node under the Data Preparation Tree node.

The R Split Data consists of two connector nodes: Upper node for the **training data set** and a lower node for the **testing data set**.
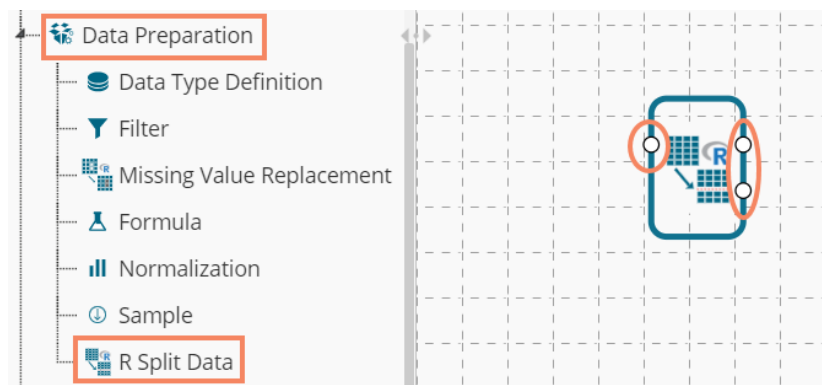
i)   Select the '**R Split Data**' component and connect it with a valid data source
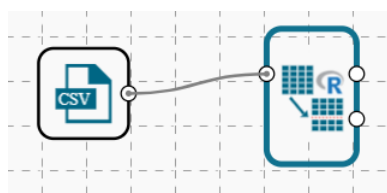
ii)   Click the '**R Split Data**' component in the workspace
iii)   Users will be directed to the Properties fields provided under the '**Components**' tab
iv)   Users can choose the size of the first partition:
    a. Relative (train): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and sum of train and test data should be 1)
    b. Relative (test): Enter a value to decide the ratio of train data out of the dataset (Type:

Decimal, Range: 0-1 and sum of train and test data should be 1)
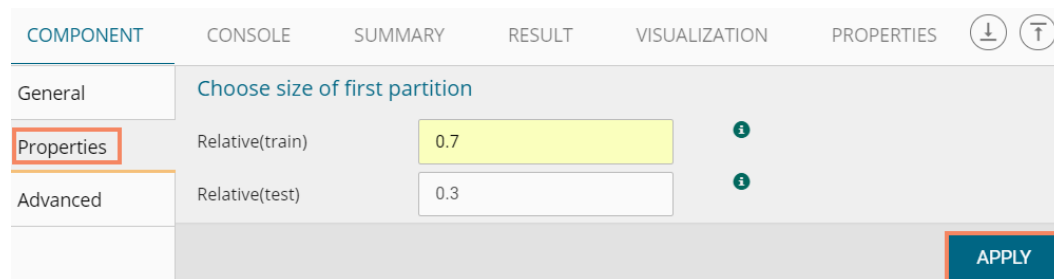
v) Users can configure the sampling type using the Advanced fields
   a. Sampling Type: Select any one option from the drop-down menu
      i. Linear Sampling
      ii. Shuffled Sampling
      iii. Stratified Sampling

vi) Click '**APPLY**'
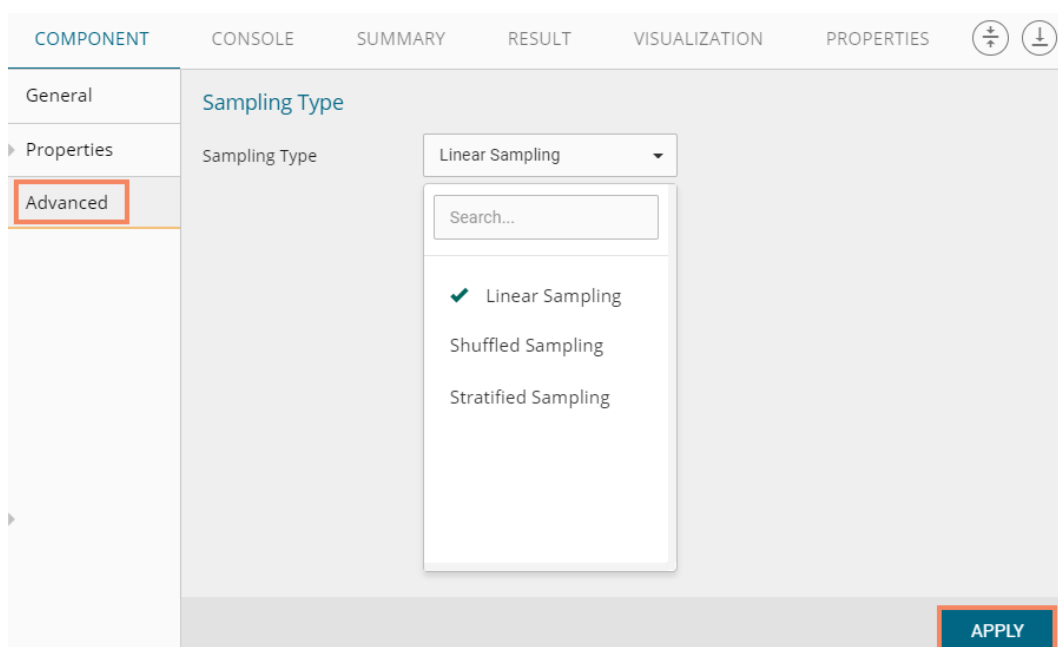
vii) Run the workflow
viii) Users will be directed to the '**CONSOLE**' tab
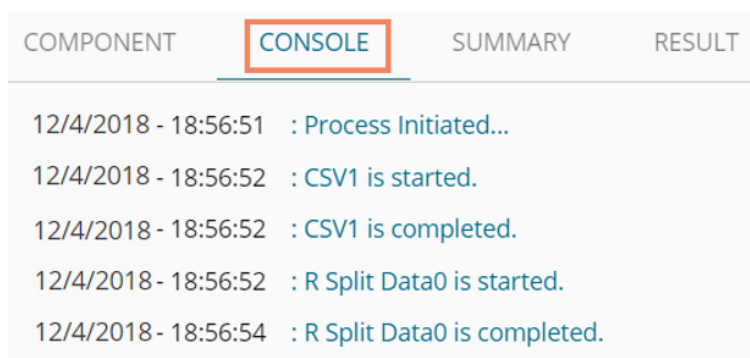
ix) Follow the below given steps to display the result view:

a. Click the dragged algorithm component in the workspace.
b. Click the '**RESULT**' tab.

The Result tab will have two data sets separated by a sub-tab. As shown in the below-given images:

i. Select the '**Split 1**' tab to see one set of data (the training dataset)

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | |

Split 1   Split 2

Show 10 ▾ entries   Search: |

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 105 entries   Previous 1 2 3 4 5 … 11 Next

ii. Select the '**Split 2**' tab to see another set of data (the testing dataset)

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | |

Split 1   **Split 2**

Show 10 ▾ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
106	7.6	3	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2	virginica
112	6.4	2.7	5.3	1.9	virginica
113	6.8	3	5.5	2.1	virginica
114	5.7	2.5	5	2	virginica
115	5.8	2.8	5.1	2.4	virginica

Showing 1 to 10 of 45 entries   Previous 1 2 3 4 5 Next

Note: Current document covers steps to deal with a CSV File dataset for all the R Data Preparation components. The similar steps can be followed for a Data Service data set.
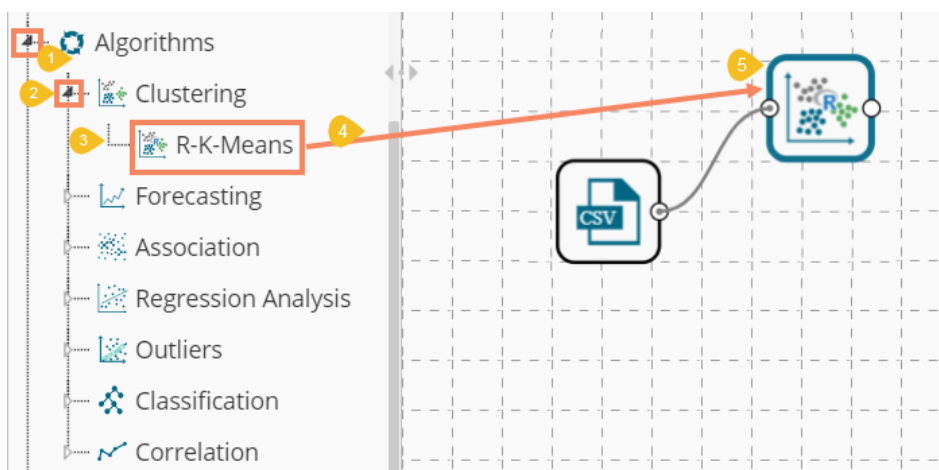
5.3. Algorithms

Algorithms are a statistical set of rules that help users analyze vast quantities of numerical data and extract appropriate information out of it. BDB Predictive Analysis allows users to apply more than one algorithm to manage the enormous amount of data.

Step by Step Process to Apply an Algorithm:

i) Click the '**Algorithms**' tree-node on the Predictive Analysis home page.

ii) Click the Algorithm Category tree-node ┊ to display the available algorithm subcategories.

iii) Select and drag an algorithm component onto the workspace.

iv) Connect the algorithm component to a configured data source.

v) Click on the algorithm component.

vi) Configure the following '**COMPONENT**' fields for the dragged algorithm component.

vii) Click '**APPLY**' to save the information.

viii) Run the workflow

ix) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

x) After the Console process gets completed, users can view result data using the '**RESULT**' tab
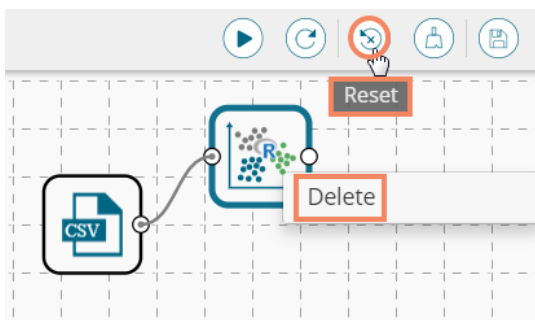
a. Click the algorithm component on the workspace
b. Click the '**RESULT**' tab

xi) The newly created Cluster Column will be added to the displayed result dataset

xii) Click the '**VISUALIZATION**' tab to see a graphical representation of the result data.

xiii) Click '**Delete**' or '**Reset**' option to remove the selected algorithm component from the workspace.

Note:

    a. Users can follow the steps mentioned above to configure all the available R- algorithms.

    b. Users can configure alias name for the algorithm component via the '**General**' tab.

    c. Basic configuration for all the algorithms is done through the '**Properties**' tab. Users are required to configure this tab while applying an algorithm component manually.

    d. Users can avail of all the default values under the '**Advanced**' tab. Users can manually set the '**Advanced**' tab or modify the default values, only if the advanced level configuration is required.

    e. After execution, users can click on the respective component to get data. Pipeline component will not have any result set; the only summary will be available. Users need to connect the pipeline components with an '**Apply Model**' component and test data set to view the result.

5.3.1. Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).

5.3.1.1. R-K Means

K- means clustering is one of the most commonly used clustering methods. It clusters data points into a predefined number of clusters. It first assembles observations into 'K' groups, wherein 'K' is an input parameter. The algorithm then assigns each observation to a cluster based on the proximity of the observation.

Applying R-K Means to a Data Source

Users will be redirected to the '**Component**' tabs when applying the '**R-K Means**' algorithm component to a configured data source.

i) Drag the R-K Means to the Workspace and connect it to a configured Data Source.

ii) The Component tabs will be displayed on the Viewspace.

iii) Configure the following fields in the '**Properties**' tab:

    a. **Output Information**

        i. **Number of Clusters**: Enter number of groups for clustering. The default value for this field is 5. Range should be between 1 and the total number of clusters.

    b. **Column Selection**

        i. **Feature:** Select the input columns with which you want to perform the Analysis

    c. **New Column Information**

        i. **Cluster Name:** Enter a name for the new column displaying cluster number

- **Rules for Naming a New Column**
  1. Do not use space in the name of a new column. It should be a single word, or two words should be connected by an underscore (_). E.g., SampleData or Sample_Data.
  2. Do not use any special symbol alone or with any character as the name of a new column. Eg. %, #, $, @,* or Sample# are not acceptable.
  3. Do not use single or double quotes, dot, and brackets to name a new column.
  4. Do not use numbers alone to name a new column. Numbers can be used with at least one character of the alphabet, and the name should not begin with a numeral.
  5. Name given to a new column should not exceed 50 characters.

  Note: Users can access a list of rules for naming a new column by clicking the information
  icon 🛈 provided next to the **'New Column Information'** tab.

iv) Click the **'Advanced'** tab **(**if required**)**
   a. Configure the required **'Behavior'** fields:
      i. **Maximum Iterations**: Enter the number of iterations allowed for discovering clusters. (The default value for this field is 100).
     ii. **Number of Initial Centroids**: Enter the number of random initial centroid sets for clustering (The default value for this field is 1).
    iii. **Algorithm type**: Select an algorithm type from the drop-down menu
     iv. **Initial Cluster Center Seed**: Enter a number indicating initial cluster center seed (The default value for this field is 10).

v) Click **'APPLY'**

vi) Run the workflow after getting the success message

vii) Users get redirected to the **'CONSOLE'** tab describing the progress of the process.

COMPONENT	CONSOLE	SUMMARY	RESULT

12/4/2018 - 13:20:20 : Process Initiated...

12/4/2018 - 13:20:21 : CSV0 is started.

12/4/2018 - 13:20:21 : CSV0 is completed.

12/4/2018 - 13:20:22 : R-K-Means1 is started.

12/4/2018 - 13:20:23 : R-K-Means1 is completed.

viii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the **'RESULT'** tab

ix) A new column **'Cluster Number'** will be displayed in the result view

COMPONENT	CONSOLE	SUMMARY	**RESULT**	VISUALIZATION	PROPERTIES

Show 10 ▼ entries   Search:

RowID	SLength	SWidth	PLength	PWidth	ClusterNumber1
1	5.1	3.5	1.4	0.2	5
2	4.9	3	1.4	0.2	5
3	4.7	3.2	1.3	0.2	5
4	4.6	3.1	1.5	0.2	5
5	5	3.6	1.4	0.2	5
6	5.4	3.9	1.7	0.4	5
7	4.6	3.4	1.4	0.3	5
8	5	3.4	1.5	0.2	5
9	4.4	2.9	1.4	0.2	5
10	4.9	3.1	1.5	0.1	5

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

x) Click the **'VISUALIZATION'** tab.

xi) The result data will be displayed via the Scatter Plot Matrix Chart.

5.3.2. Forecasting

Forecasting is a method that used extensively in time series analysis to predict a response variable, such as monthly profits, stock performance, or unemployment figures, for a specified period. Forecasts are based on patterns in existing data. For example, a warehouse manager can create a model of how much product to order for the next three months based on the previous 12 months of orders.

All the sub-categories of the Forecasting Algorithms provide two Output modes (to be set from the Properties tab):

1. Forecasting
2. Trend

The document describes all the available Forecasting algorithms as per the selected Output mode.
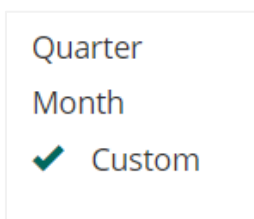
5.3.2.1. Triple Exponential Smoothing

Triple exponential smoothing considers seasonal changes as well as trends (all of which are trends). Seasonality is defined to be the tendency of time-series data to exhibit behavior that repeats itself every L period, much like any harmonic function. The term season is used to represent the period before behavior begins to repeat itself. There are different types of seasonality: 'multiplicative' and 'additive' in nature, much like addition and multiplication are fundamental operations in mathematics.

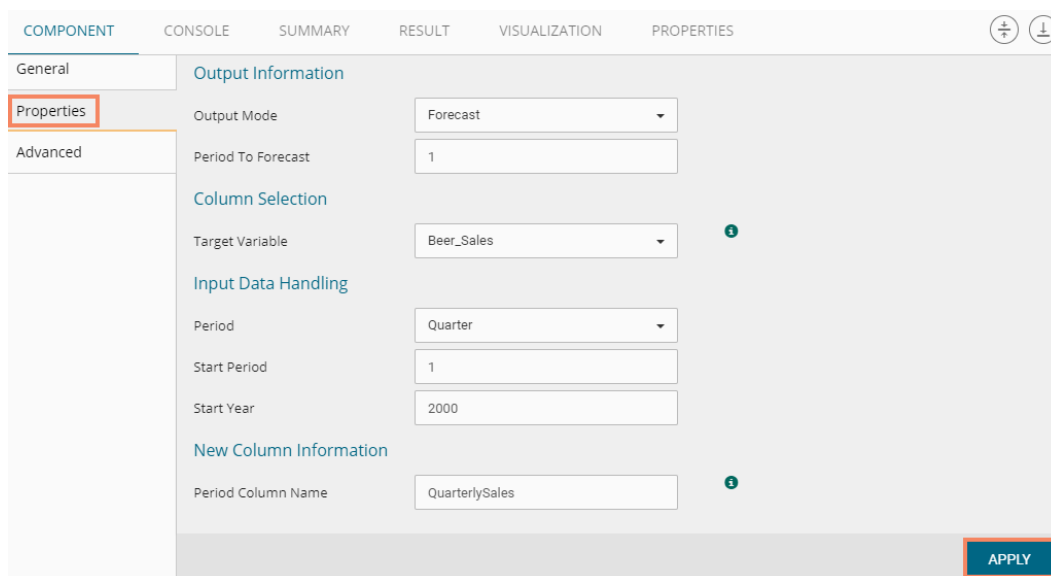i) Drag the Triple Exponential Smoothing component to the workspace and connect to a configured data source.
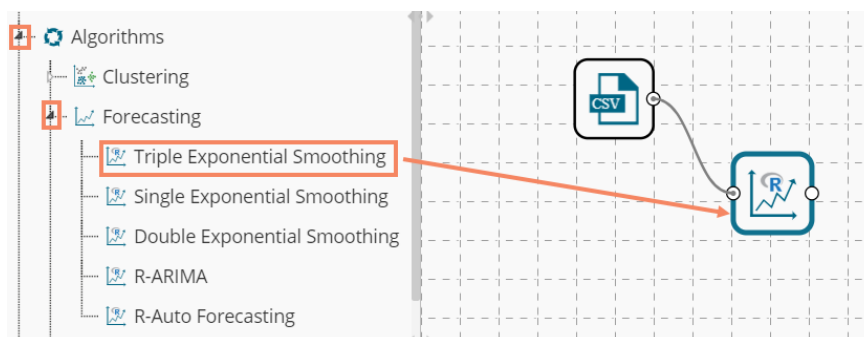
ii) Configure the following fields in the '**Properties**' tab:
   a. **Output Information**
      i. **Output Mode**: Select a mode in which you want to display output data. Users will get two options for this field.
         1. **Trend**: Selecting this option will display source data along with predicted values for the given data set.
         2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.
      ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected '**Output Mode**' option is '**Forecast.**'
   b. **Column Selection**
      i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)
   c. **Input Data Handling**
      i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu

      ii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field
      iii. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

   d. **New Column Information**
      i. **Period Column Name:** Enter a name for the column containing a period value. (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure, if required:
    a. Configure the following '**Behavior**' fields:
        i. **Alpha:** Enter a valid double value in the given field for smoothing observations (Alpha Range: 0<alpha<=1)
        ii. **Beta:** Enter a valid double value in the given field for finding trend parameters (Beta Range: 0-1)
        iii. **Gamma:** Enter a valid double value in the given field for finding a seasonal trend parameter (Gamma Range: 0-1)
        iv. **Seasonal:** Select a smoothing algorithm type from the drop-down list (Holtwinter's Exponential Smoothing algorithm)
        v. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.

    b. Configure the following '**Initial Values**' information:
        i. **Level:** Enter the initial value for the level. It is an optional field.
        ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.
        iii. **Season:** Enter initial values for finding seasonal parameters. It depends on the selected column. It is an optional field.
        iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.
        v. **Confidence**: Enter Confidence level for prediction intervals. It accepts only 0-99 and comma separated value. According to the number of comma-separated values new low and high range columns will be added to the result dataset. (the default value for this field is 95)
        vi. **Show Range:** Select an option using the drop-down menu.
           1. True: By selecting this option '**Lower Range**' and '**Upper Range**' will be displayed in the Result and Visualization of the dataset.
           2. False: By selecting this option, Ranges will not be shown in the dataset
iv) Click '**APPLY**'

Properties	**Initial Values**	
Advanced	Level	Optional
	Trend	Optional
	Season	Optional
	Optimizer Inputs	Optional
	Confidence	95
	Show Range	False ▼

APPLY

v)   Run the workflow after getting the success message

vi)   Users get directed to the '**CONSOLE**' tab displaying the ongoing process

COMPONENT   **CONSOLE**   SUMMARY   RESULT   VISUALIZATION

12/4/2018 - 18:56:11   : Process Initiated...

12/4/2018 - 18:56:11   : CSV0 is started.

12/4/2018 - 18:56:11   : CSV0 is completed.

12/4/2018 - 18:56:12   : R-Triple Exponential Smoothing1 is started.

12/4/2018 - 18:56:13   : R-Triple Exponential Smoothing1 is completed.

vii)   Follow the below-given steps to display the result view:
a. Click the dragged algorithm component onto the workspace.
b. Click the '**RESULT**' tab (In this case, the selected output mode is '**Forecasting**')
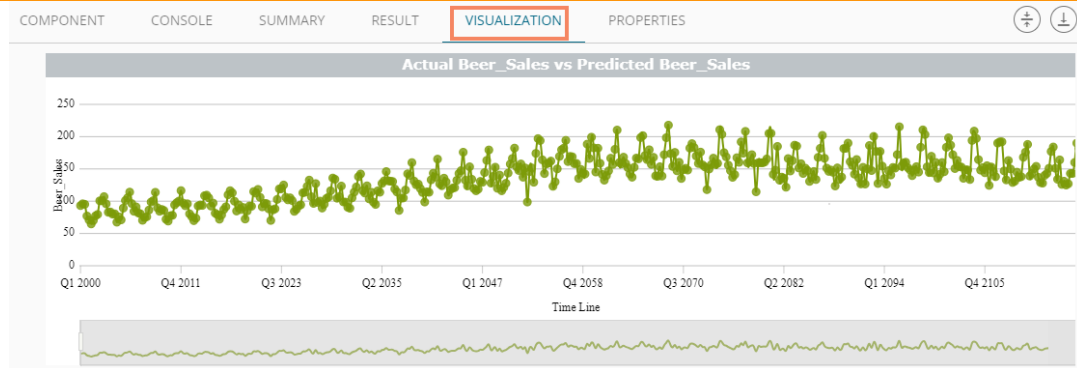
COMPONENT   CONSOLE   SUMMARY   **RESULT**   VISUALIZATION   PROPERTIES

Show 10 ▾ entries   Search:

Year	Month	Beer_Sales	QuarterlySales
1965	January	93.2	Q1 2000
1965	February	96	Q2 2000
1965	March	95.2	Q3 2000
1965	April	77.1	Q4 2000
1965	May	70.9	Q1 2001
1965	June	64.8	Q2 2001
1965	July	70.1	Q3 2001
1965	August	77.3	Q4 2001
1965	September	79.5	Q1 2002
1965	October	100.6	Q2 2002

Showing 1 to 10 of 469 entries   Previous 1 2 3 4 5 ... 47 Next

viii)   Click the '**VISUALIZATION**' tab.

ix)   The result data will be displayed via the Time Line Chart.

Actual Beer_Sales vs Predicted Beer_Sales

x)   Click the '**SUMMARY**' tab to view the model summary.


```
------------ Summary of the model ------------
Columns used in the algorithm
              Beer_Sales      (double)


 Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = tso, alpha = as.numeric(0.3), beta = as.numeric(0.1),    gamma = as.numeric(0.1), seasonal
= c("additive"), start.periods = as.numeric(2),    s.start = c(), optim.start = c())

Smoothing parameters:
 alpha: 0.3
 beta : 0.1
 gamma: 0.1

Coefficients:
      [,1]
a  160.221
b    1.757
s1  -4.298
s2  -1.413
s3  12.655
s4  10.583


------------ End of Summary ----------
```
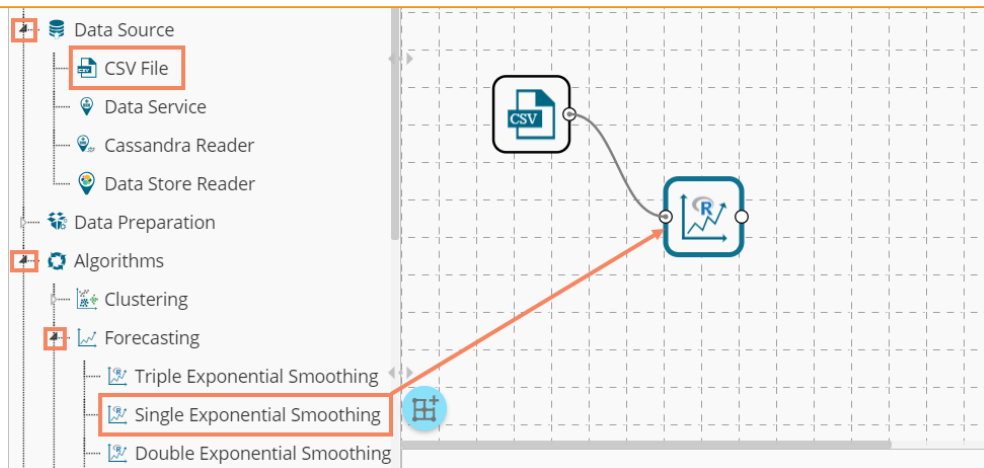
5.3.2.2.   Single Exponential Smoothing

The Single Exponential Smoothing is the simplest of all the smoothing methods also known as Simple Exponential Smoothing. This method is suitable for forecasting data with no trend or seasonal pattern.

i)   Drag the Single Exponential Smoothing component to the workspace and connect to a configured data source.

ii) Configure the '**Properties**' tab.
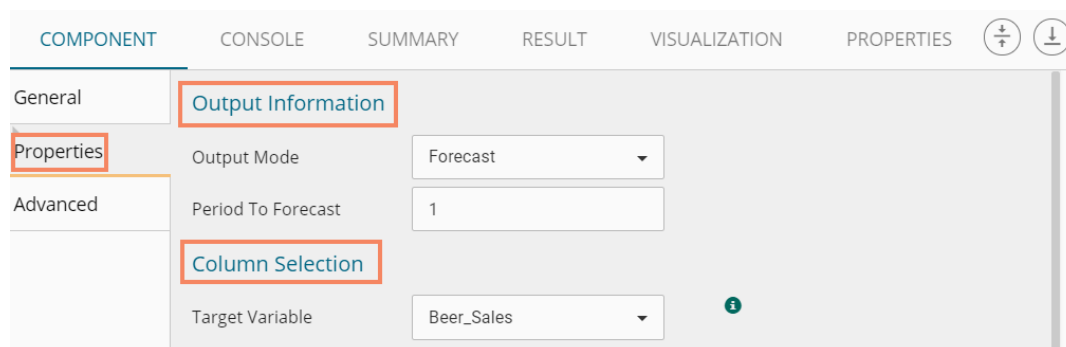 a. **Output Information**
    i. **Output Mode**: Select a mode in which you want to display output data
      1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column '**Predicted Values**' will be added in the result view when '**Trend**' output mode has been selected.
      2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.
    ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected '**Output Mode**' option is '**Forecast.**'
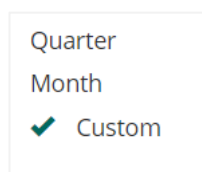 b. **Column Selection**
    i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (the first option gets selected by default. Only numerical columns are accepted)
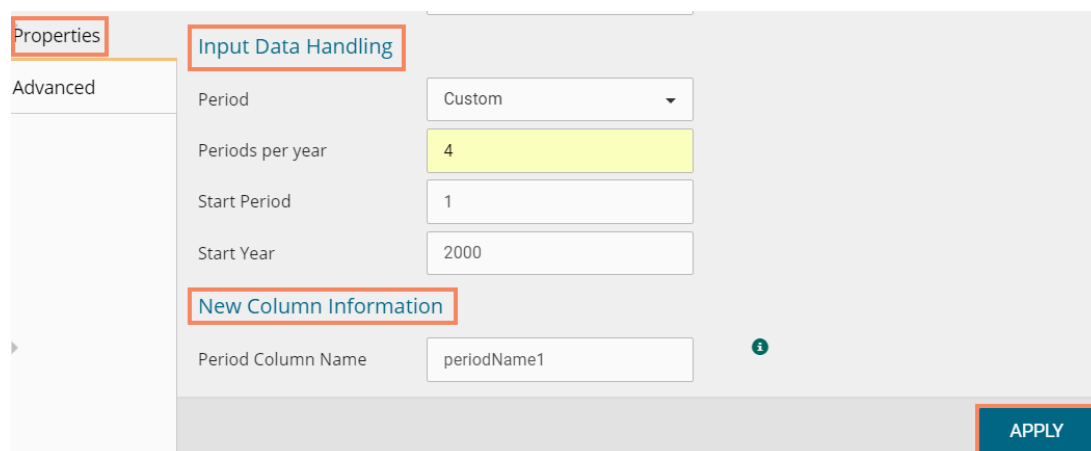

  c. **Input Data Handling**
    i. **Period:** Select period of forecasting by choosing any one option from the drop-down menu

  ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'

  iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field

  iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

 d. **New Column Information**

  i. **Period Column Name:** Enter a name for the column containing a period value. (This field will be predefined, but users can change the value if needed).

Note: The '**Period Per Year**' field will display only when the selected value for the '**Period**' field is '**Custom.**'

 iii) Click the '**Advanced**' tab and configure if required.

  a. Configure the following '**Behavior**' fields:

   i. **Alpha:** Enter a valid double value in the given field for smoothing observations. Alpha Range: 0<alpha<=1.

   ii. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.

  b. Configure the following '**Initial Values**' information:

   i. **Level**: Enter the initial value for the level. It is an optional field.

   ii. **Confidence**: Enter Confidence level for prediction intervals. It accepts only 0-99 and comma separated value. According to the number of comma-separated values new low and high range columns will be added to the result dataset. (the default value for this field is 95)

   iii. **Show Range:** Select an option using the drop-down menu.

    1. True: By selecting this option '**Lower Range**' and '**Upper Range**' will be displayed in the Result and Visualization of the dataset.

    2. False: By selecting this option, Ranges will not be shown in the dataset.

 iv) Click '**APPLY**'

v) Run the workflow after getting the success message
vi) Users get directed to the '**CONSOLE**' tab displaying the ongoing process

vii) Follow the below-given steps to display the result view:
    a. Click the dragged algorithm component onto the workspace
    b. Click the '**RESULT**' tab
viii) Predicted values will be appended to the target column in the result data (In the case, the selected output mode is '**Forecasting.**'
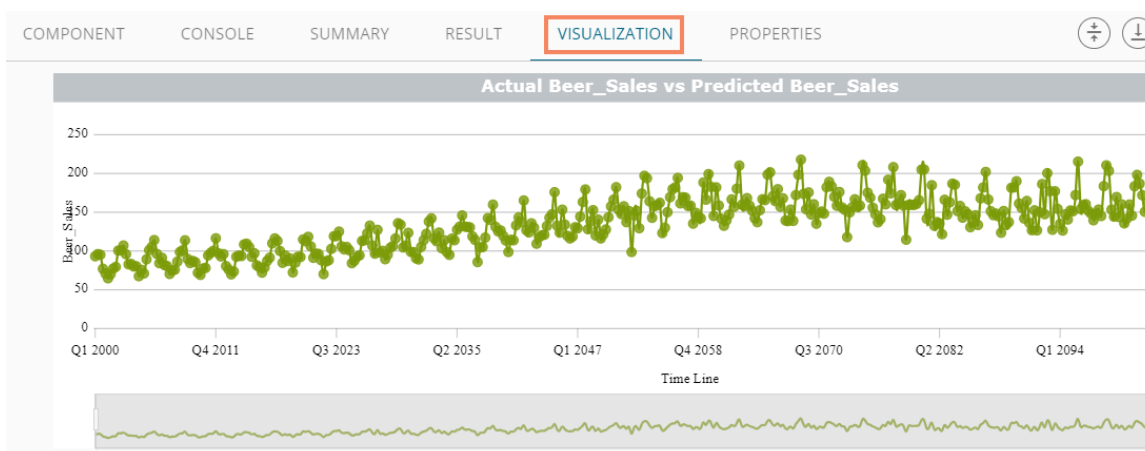
ix)   Click the '**VISUALIZATION**' tab

x)   The result data will be displayed via the Time Line Chart

xi)   Click the '**SUMMARY**' tab to view the model summary

5.3.2.3. Double Exponential Smoothing

Single Exponential smoothing method cannot perform well when there is a trend in the data. In such circumstances, several methods were devised under the name Double Exponential Smoothing or Second-order Exponential Smoothing which is the recursive application of an exponential filter twice. Therefore it was termed Double Exponential Smoothing. The basic idea behind double exponential smoothing is to introduce a term to consider the possibility of a series exhibiting some form of the trend. This slope component is itself updated via exponential smoothing.

i)   Drag the Double Exponential Smoothing component to the workspace and connect to a configured data source

ii) Configure the '**Properties**' tab

a. **Output Information**
   i. **Output Mode**: Select a mode in which you want to display output data
      1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column '**Predicted Values**' will be added in the result view when '**Trend**' output mode has been selected.
      2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.
   ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected '**Output Mode**' option is '**Forecast.**'

b. **Column Selection**
   i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.
   ii. **Start Period:** Enter a value between 1 and the value specified for the selected option for **'Period'** field
   iii. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

d. **New Column Information**
   i. Period Column Name: Enter a name for the column containing period value (This field will be predefined, but users can change the value if needed)

Input Data Handling

Period	Month
Start Period	1
Start Year	2000

New Column Information

Period Column Name	Months

APPLY

iii) Click the '**Advanced**' tab and configure if required
   a. Configure the following '**Behavior**' fields:
      i. **Alpha:** Enter a valid double value in the given field for smoothing observations (Alpha Range: 0<alpha<=1)
      ii. **Beta:** Enter a valid double value in the given field for smoothing observations (Beta Range: 0-1)
      iii. **No. of Periodic Observation:** Enter the number of periods observations required to start the calculation (The default value for this field is 2)
   b. Configure the following '**Initial Values**' information:
      i. **Level:** Enter the initial value for the level (It is an optional field)
      ii. **Trend:** Enter the initial value for finding trend parameters (It is an optional field)
      iii. **Optimizer Inputs:** Enter the initial values given for alpha and beta required for the optimizer (it is an optional field)
      iv. **Confidence**: Enter Confidence level for prediction intervals. It accepts only 0-99 and comma-separated value. According to the number of commas separated values new low and high range columns will be added to the result dataset (the default value for this field is 95).
      v. **Show Range:** Select an option using the drop-down men**u**
         1. True: By selecting this option '**Lower Range**' and '**Upper Range**' will be displayed in the Result and Visualization of the dataset
         2. False: By selecting this option, Ranges will not be shown in the dataset
iv) Click '**APPLY**'

v) Run the workflow after getting the success message
vi) Users get directed to the '**CONSOLE**' tab displaying the ongoing process

COMPONENT | **CONSOLE** | SUMMARY | RESULT | VISUALIZATION

12/4/2018 - 18:54:58 : Process Initiated...

12/4/2018 - 18:54:59 : CSV0 is started.

12/4/2018 - 18:55:0 : CSV0 is completed.

12/4/2018 - 18:55:0 : R-Double Exponential Smoothing1 is started.

12/4/2018 - 18:55:0 : R-Double Exponential Smoothing1 is completed.

vii) Follow the below-given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab
viii) Predicted values will be appended to the target column in the result data (The selected output mode is '**Forecasting**')

COMPONENT | CONSOLE | SUMMARY | **RESULT** | VISUALIZATION | PROPERTIES

Show 10 ▼ entries   Search: _____

Year	Month	Beer_Sales	Months	Lower_Range_95_11	Upper_Range_95_11
2003	May	131	May 2038		
2003	June	125	Jun 2038		
2003	July	127	Jul 2038		
2003	August	143	Aug 2038		
2003	September	143	Sep 2038		
2003	October	160	Oct 2038		
2003	November	190	Nov 2038		
2003	December	182	Dec 2038		
		167.2	Jan 2039	126.4	208.1
		169.2	Feb 2039	126.1	212.2

Showing 461 to 470 of 480 entries   Previous 1 ... 44 45 46 47 48 Next

ix) Click the '**VISUALIZATION**' tab.
x) The result data will be displayed via the Time Line chart.
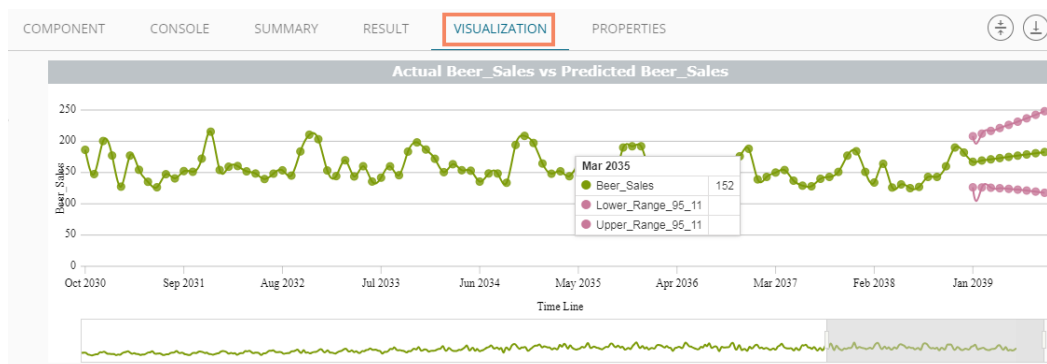
xi) Click the '**SUMMARY**' tab to view the model summary.

```
------------ Summary of the model ------------
Columns used in the algorithm
            Beer_Sales    (double)


  Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = tso, alpha = as.numeric(0.3), beta = as.numeric(0.1),    gamma = FALSE, start.periods = as.numeric(2), optim.start = c(0,
0.1, 0.2))

Smoothing parameters:
 alpha: 0.3
 beta : 0.1
 gamma: FALSE

Coefficients:
     [,1]
a 165.251
b   1.954


------------ End of Summary ----------
```
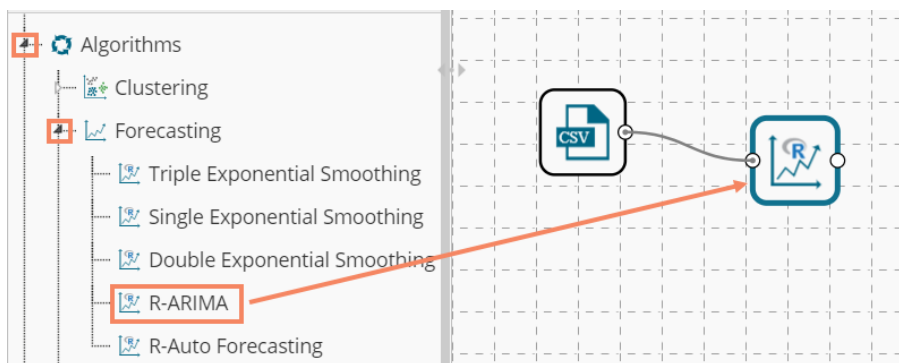
5.3.2.4.   R-ARIMA

R- ARIMA returns best ARIMA model according to either AIC, AICc or BIC value. The function searches for a possible model within the order constraints provided.

i)   Drag the R-ARIMA component to the workspace and connect to a configured data source.

ii)   Configure the '**Properties**' tab.
   a. Output Information
      i. **Output Mode**: Select a mode in which you want to display output data
         1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column '**Predicted Values**' will be added in the result view when '**Trend**' output mode has been selected.
         2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.
      ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected '**Output Mode**' option is '**Forecast.**'
   b. Column Selection
      i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.

   ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'
   iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field
   iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

d. **New Column Information**

   i. **Period Column Name:** Enter a name for the column containing period value (This field will be predefined, but users can change the value if needed).

iii) Enable Manual Arima option by putting a checkmark in the given box
iv) The '**NEXT**' option will be added to the page
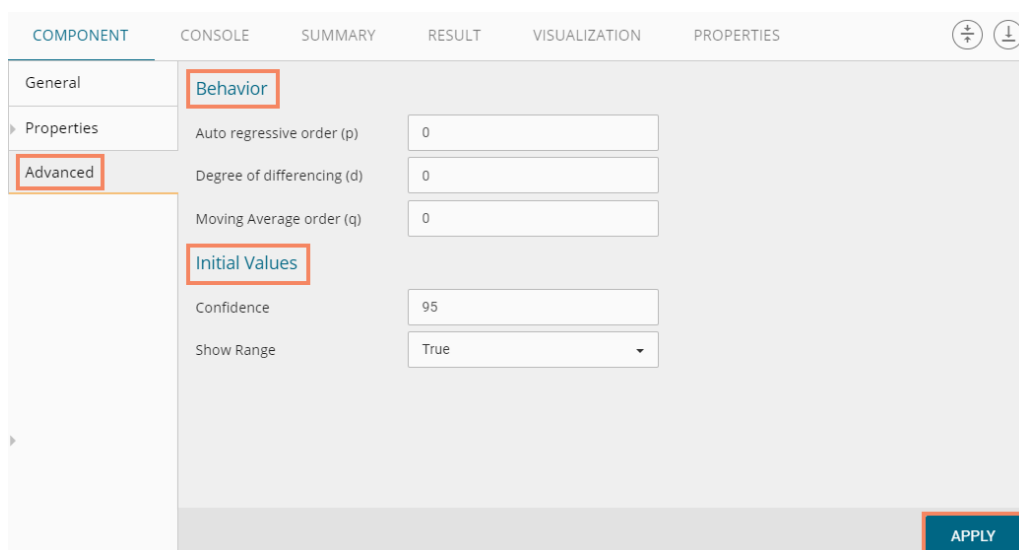
v) Click the '**Advanced**' tab and configure if required
   a. Configure the following '**Behavior**' fields:
      i. **Autoregressive order(p):** It is a mandatory field; only integer values are accepted. The default value for this field is 0.

      ii. **Degree of differencing(d):** It is a mandatory field; only integer values are accepted. The default value for this field is 0.

     iii. **Moving Average Order(q):** It is a mandatory field; only integer values are accepted. The default value for this field is 0.

  b. Configure the following '**Initial Values**' information:

      i. **Confidence**: Enter Confidence level for prediction intervals. It accepts only 0-99 and comma separated value. According to the number of commas separated values new low and high range columns will be added to the result dataset. (the default value for this field is 95)

      ii. **Show Range:** Select an option using the drop-down menu**.**

         1. **True**: By selecting this option '**Lower Range**' and '**Upper Range**' will be displayed in the Result and Visualization of the dataset.

         2. **False**: By selecting this option, Ranges will not be shown in the dataset.

vi)   Click '**APPLY**'

vii)   Run the workflow after getting the success message

viii)   Users get directed to the '**CONSOLE**' tab displaying the progress of the process

ix)   Follow the below given steps to display the result view:

  a. Click the dragged algorithm component onto the workspace

  b. Click the '**RESULT**' tab

x)   Predicted values will be appended to the target column in the result data (The selected output mode is '**Forecasting**')

Year	Month	Beer_Sales	QuarterlySales	Lower_Range_95_12	Upper_Range_95_12
2003	May	131	Q1 2115		
2003	June	125	Q2 2115		
2003	July	127	Q3 2115		
2003	August	143	Q4 2115		
2003	September	143	Q1 2116		
2003	October	160	Q2 2116		
2003	November	190	Q3 2116		
2003	December	182	Q4 2116		
		136.4	Q1 2117	69.82	202.9
		136.4	Q2 2117	69.82	202.9

Showing 461 to 470 of 476 entries

Previous 1 ... 44 45 46 47 48 Next

xi) Click the '**VISUALIZATION**' tab.

xii) The result data will be displayed via the Time Line chart.
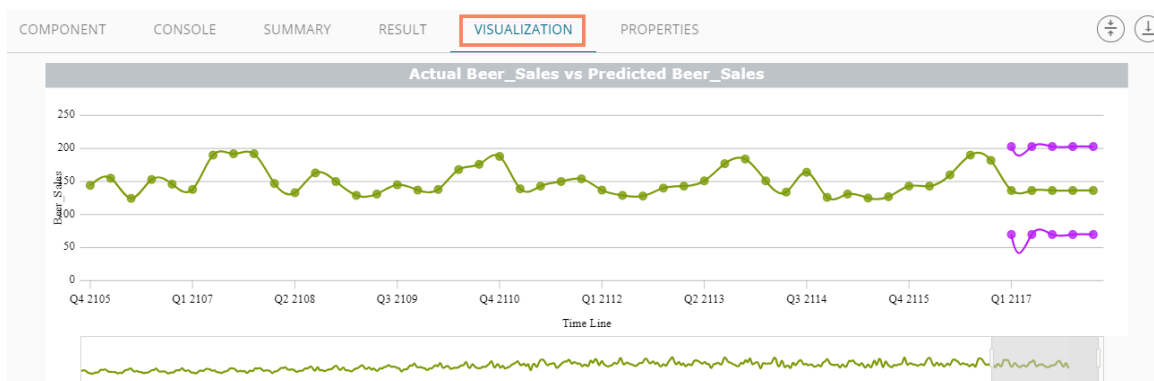
xiii) Click the '**SUMMARY**' tab to view the model summary.

```
------------ Summary of the model ------------
Columns used in the algorithm
            Beer_Sales      (double)



Call:
arima(x = tso, order = c(0, 0, 0))

Coefficients:
        intercept
        136.3637
s.e.      1.5695

sigma^2 estimated as 1153:  log likelihood = -2313.76,  aic = 4631.52

------------ End of Summary ----------
```

Note: When 'Manual ARIMA' option is not disabled for the R-ARIMA algorithm, the 'Advanced' tab will not display Behavior fields. The following images display respectively the 'Advanced,' 'Result' and 'Visualization' tabs for the same dataset when manual ARIMA option has been disabled.

Advanced Tab

Result Tab

Year	Month	Beer_Sales	periodName1	Lower_Range_95_12	Upper_Range_95_12
2003	May	131	Q1 2115		
2003	June	125	Q2 2115		
2003	July	127	Q3 2115		
2003	August	143	Q4 2115		
2003	September	143	Q1 2116		
2003	October	160	Q2 2116		
2003	November	190	Q3 2116		
2003	December	182	Q4 2116		
		162.5	Q1 2117	133.48	191.4
		138.0	Q2 2117	106.19	169.8

Show 10 entries   Search:

Showing 461 to 470 of 476 entries   Previous 1 ... 44 45 46 **47** 48 Next

Visualization Tab

Actual Beer_Sales vs Predicted Beer_Sales
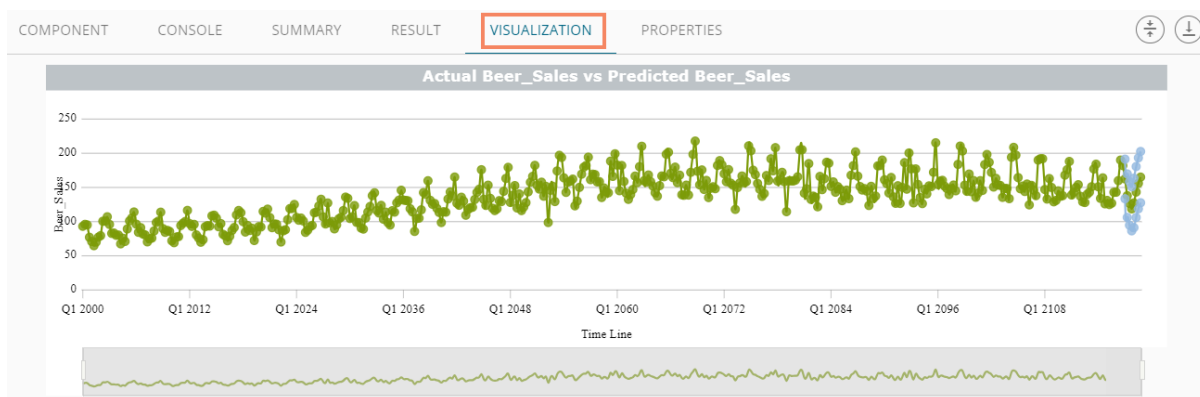
5.3.2.5. R- Auto Forecasting
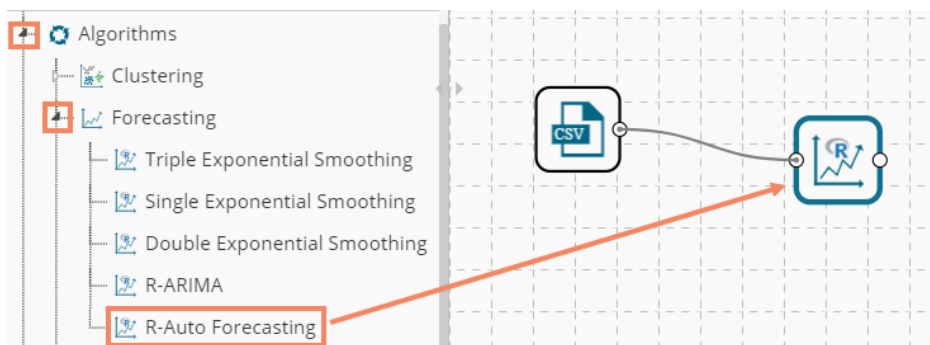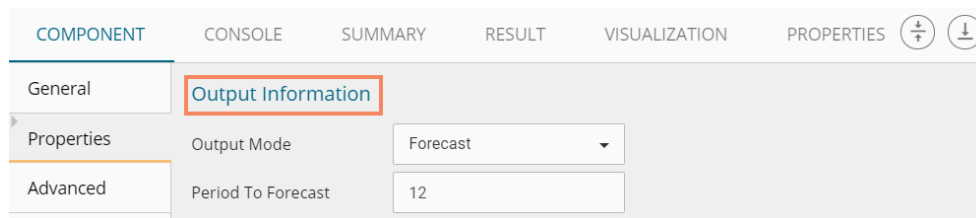The user can run the algorithm by adjusting smoothing parameters and other initial state variables to find the best AIC value.
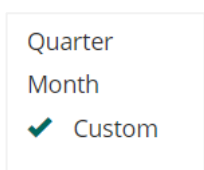
i) Drag the R-Auto Forecasting component to the workspace and connect to a configured data source.

ii) Configure the '**Properties**' tab.
   a. **Output Information**
      i. **Output Mode**: Select a mode in which you want to display output data
         1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column **'Predicted Values' will be added in the result view when** '**Trend**' output mode has been selected.
         2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.
      ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected '**Output Mode**' option is '**Forecast.**'
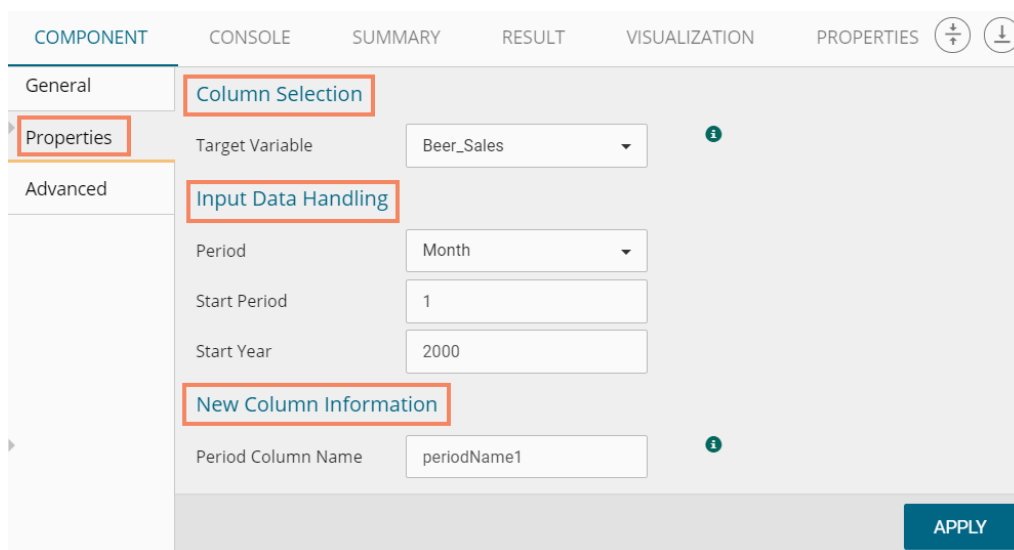

   b. **Column Selection**
      i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

   c. **Input Data Handling**
      i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu

ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'

iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field

iv. **Start Year:** Enter four digit value for selecting a year from which you want the data entries to be considered (E.g., 2000)

d. New Column Information

i. **Period Column Name:** Enter a name for the column containing period value (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure if required:

a. Configure the following '**Behavior**' fields:
i. **Seasonal:** Select a smoothing algorithm type from the drop-down menu (Holtwinter's Exponential Smoothing algorithm)
ii. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.

b. Configure the following '**Initial Values**' fields:
i. **Level:** Enter the initial value for the level (It is an optional field)
ii. **Trend:** Enter the initial value for finding trend parameters (It is an optional field)
iii. **Season:** Enter initial values for finding seasonal parameters. It will depend on the selected column. It is an optional field.
iv. **Optimizer Inputs:** Enter the initial values given for alpha and beta required for the optimizer (It is an optional field).
v. **Confidence**: Enter Confidence level for prediction intervals. It accepts only 0-99 and comma-separated value. According to the number of comma-separated values new low and high range columns will be added to the result dataset (the default value for this field is 95).
vi. **Show Range:** Select an option using the drop-down menu.
1. **True**: By selecting this option '**Lower Range**' and '**Upper Range**' will be displayed in the Result and Visualization of the dataset.
2. **False**: By selecting this option, Ranges will not be shown in the dataset.

iv) Click '**APPLY**'

v) Run the workflow after getting the success message

vi) Users get redirected to the '**CONSOLE**' tab displaying the progress of the process

vii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab

viii) Predicted values will be appended to the target column in the result data (The selected output mode is '**Forecasting**')

Year	Month	Beer_Sales	periodName1	Lower_Range_95_13	Upper_Range_95_13
1965	January	93.2	Jan 2000		
1965	February	96	Feb 2000		
1965	March	95.2	Mar 2000		
1965	April	77.1	Apr 2000		
1965	May	70.9	May 2000		
1965	June	64.8	Jun 2000		
1965	July	70.1	Jul 2000		
1965	August	77.3	Aug 2000		
1965	September	79.5	Sep 2000		
1965	October	100.6	Oct 2000		

Showing 1 to 10 of 480 entries   Previous 1 2 3 4 5 ... 48 Next

ix) Click the '**VISUALIZATION**' tab

x) The result data will be displayed via the time series chart

xi) Click the '**SUMMARY**' tab to view the model summary

5.3.2.6. Forecasting Algorithms with 'Trend' Output Mode:

A new column '**Predicted Values**' will be added to the result view when '**Trend**' is selected as an output mode.

1. **Triple Exponential Smoothing**

i) Drag the Forecasting algorithm to the workspace and connect it with the configured data source.

ii) Configure the '**Properties**' tab for the Forecasting Algorithm component keeping '**Trend**' as the '**Output Mode.**'

  a. **Output Information**
     i. **Output Mode**: Select a mode in which you want to display output data
        1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column displaying the predicted values will be added in the result view when '**Trend**' output mode has been selected.
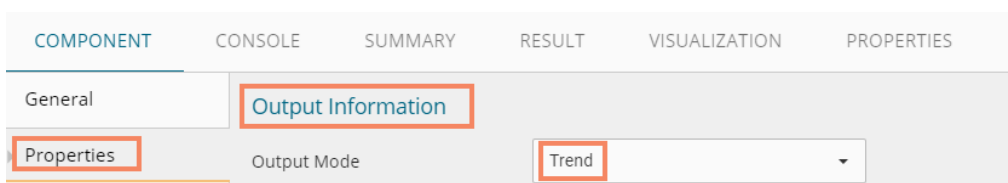
COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES
General		Output Information			
Properties		Output Mode	Trend		

  b. **Column Selection**
     i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

  c. **Input Data Handling**
     i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.
     ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'
     iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for **'Period'** field
     iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

  d. **New Column Information**
     i. **Predicted Column Name:** Enter a name for the column containing predicted values (This field will be predefined and displayed only if the selected 'Output Mode' is '**Trend**').
     ii. **Period Column Name:** Enter a name for the column containing a period value. (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure
   a. Configure the following '**Behavior**' fields:
      i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)
      ii. **Beta:** Enter a valid double value in the given field for finding trend parameters. (Beta Range: 0-1.)
      iii. **Gamma:** Enter a valid double value in the given field for finding seasonal trend parameters. (Gamma Range: 0-1.)
      iv. **Seasonal:** Select a smoothing algorithm type from the drop-down list (Holtwinter's Exponential Smoothing algorithm)
      v. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.
   b. Configure the following '**Initial Values**' information:
      i. **Level:** Enter the initial value for the level. It is an optional field.
      ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.
      iii. **Season:** Enter initial values for finding seasonal parameters. It will depend on the selected column. It is an optional field.
      iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.
iv) Click '**APPLY**'

v) Run the workflow and open the '**RESULT**' tab after the console process gets completed
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab
   c. A new column '**Predicted Values**' will be added in the result view when 'Trend' output mode has been selected.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 ▼ entries   Search:

Year	Month	Beer_Sales	periodName1	PredictedValues1
1965	January	93.2		
1965	February	96		
1965	March	95.2		
1965	April	77.1		
1965	May	70.9	Q1 2001	85.22
1965	June	64.8	Q2 2001	71.75
1965	July	70.1	Q3 2001	76.84
1965	August	77.3	Q4 2001	56.81
1965	September	79.5	Q1 2002	56.81
1965	October	100.6	Q2 2002	55.85

Showing 1 to 10 of 468 entries   Previous 1 2 3 4 5 ... 47 Next

vi) Click the '**VISUALIZATION**' tab.
vii) The result data will be displayed via the Time Line Chart

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Actual Beer_Sales vs Predicted Beer_Sales

viii) Click the '**SUMMARY**' tab to view the model summary

```
------------ Summary of the model ------------
Columns used in the algorithm
            Beer_Sales      (double)


 Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = tso, alpha = as.numeric(0.3), beta = as.numeric(0.1),    gamma = as.numeric(0.1), seasonal = c("additiv
e"), start.periods = as.numeric(2),    s.start = c(), optim.start = c())

Smoothing parameters:
 alpha: 0.3
 beta : 0.1
 gamma: 0.1

Coefficients:
      [,1]
a  160.221
b    1.757
s1  -4.298
s2  -1.413
s3  12.655
s4  10.583


------------ End of Summary ----------
```
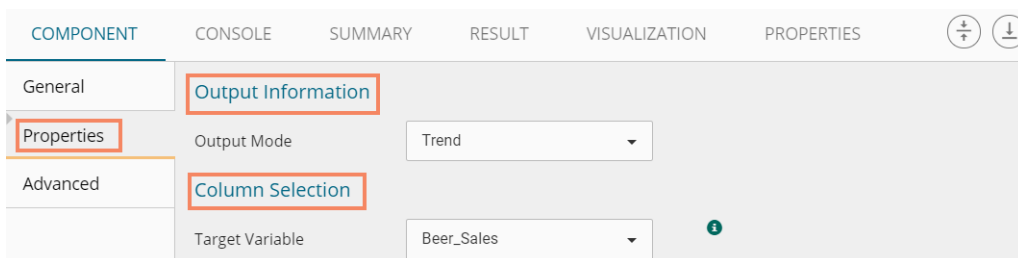
Note:

a. **'Properties'** and **'General'** sections remain the same for all the Forecasting sub-algorithms.

b. The **'Advanced'** tab displays different fields as per the Forecasting sub-types. Hence, **'Advanced'** fields for all the sub-types are explained over here. Predicted values will be appended to the target column in the result view for all the **'Forecasting'** algorithms.

2. Single Exponential Smoothing

i)   Configure the following 'Properties' fields with **'Trend'** the selected **'Output Mode'** option.

ii) Configure the following fields in the **'Properties'** tab:

a. **Output Information**
   i. **Output Mode**: Select a mode in which you want to display output data
      1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column displaying the predicted values will be added in the result view when **'Trend'** output mode has been selected.

b. **Column Selection**
   i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)
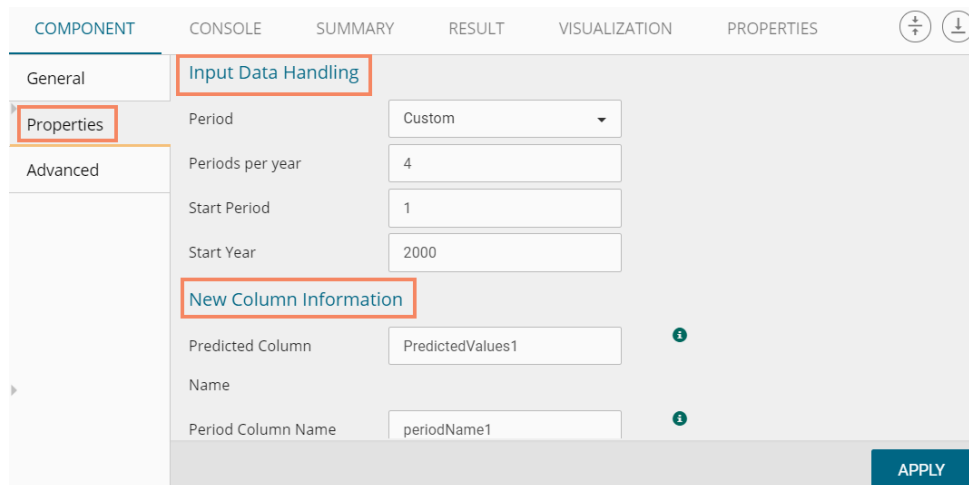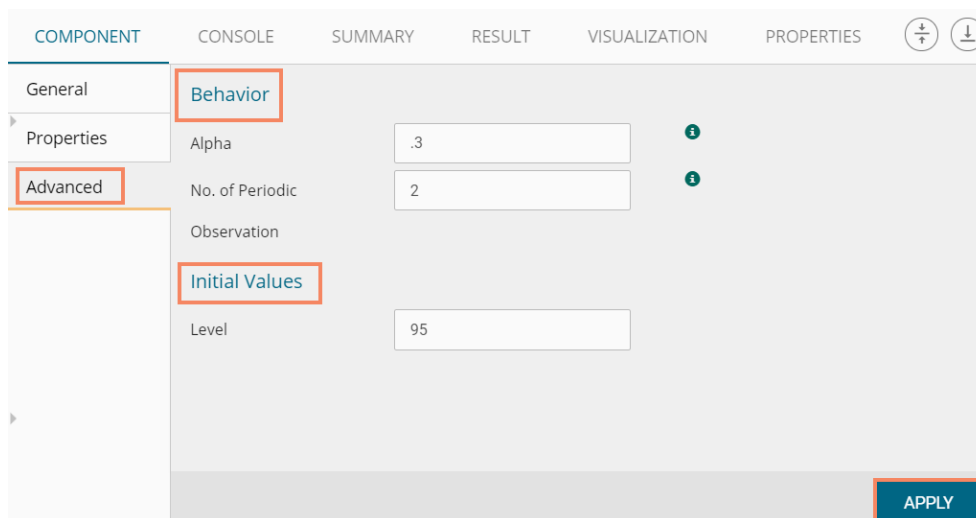

c. **Input Data Handling**
   i.   **Period:** Select period of forecasting by choosing any one option from the drop-down menu.
   ii. **Period Per Year**: This field appears only when the selected **'Period'** option is **'Custom.'**
   iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for **'Period'** field

    iv. **Start Year:** Enter four digit value for selecting a year from which you want the data entries to be considered (E.g., 2000)

  d. **New Column Information**

    i. **Predicted Column Name:** Enter a name for the column containing predicted values (This field will be predefined and displayed if the selected Output Mode is '**Trend**').

    iii. **Period Column Name:** Enter a name for the column containing a period value. (This field will be predefined, but users can change the value if needed).

iii) Configure the required '**Advanced**' fields:

  a. Configure the following '**Behavior**' fields:

    i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)

    ii. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.

  b. Configure the following '**Initial Values**' information**:**

    i. **Level:** Enter the initial value for the level. It is an optional field.
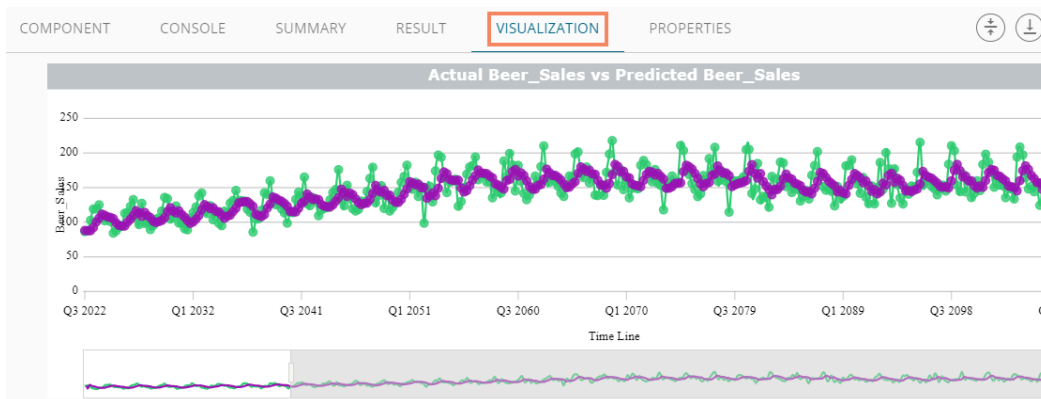
iv) Click '**APPLY**'

v) Run the workflow and open the 'RESULT' tab after the console process gets completed

  a. Click the dragged algorithm component from the workspace and then click

b. Click the 'RESULT' tab.

Year	Month	Beer_Sales	periodName1	PredictedValues1
1965	January	93.2		
1965	February	96	Q2 2000	95
1965	March	95.2	Q3 2000	95.3
1965	April	77.1	Q4 2000	95.27
1965	May	70.9	Q1 2001	89.82
1965	June	64.8	Q2 2001	84.14
1965	July	70.1	Q3 2001	78.34
1965	August	77.3	Q4 2001	75.87
1965	September	79.5	Q1 2002	76.3
1965	October	100.6	Q2 2002	77.26

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 entries   Search:

Showing 1 to 10 of 468 entries   Previous 1 2 3 4 5 ... 47 Next

vi) Click the '**VISUALIZATION**' tab.

vii) The result data will be displayed via the Time Series Chart.

viii) Click the '**SUMMARY**' tab to view the model summary
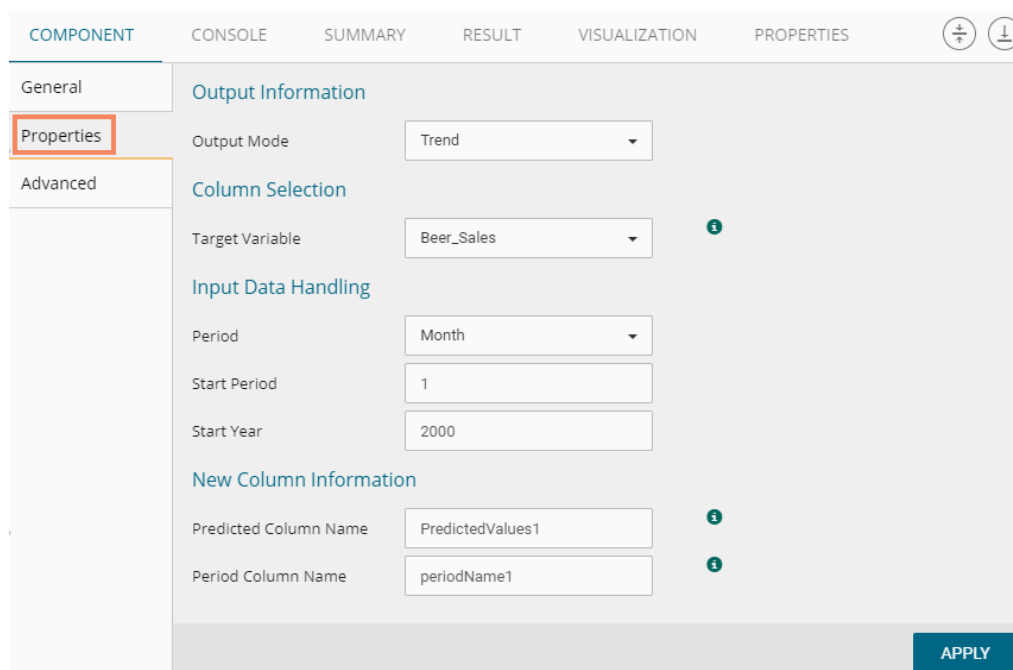

3. Double Exponential Smoothing

i) Select '**Trend**' option from the '**Output Mode**' drop-down menu.

ii) Configure the following fields in the 'Properties' tab:

a. **Output Information**
   i. **Output Mode**: Select a mode in which you want to display output data
      1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column displaying the predicted values will be added in the result view when 'Trend' output mode has been selected.
b. **Column Selection**
   i. **Target Variable**: Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)
c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down Menu.
   ii. **Start Period:** Enter a value between 1 and the value specified for the selected option for **'Period'** field
   iii. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)
d. **New Column Information**
   i. **Predicted Column Name:** Enter a name for the column containing predicted values (This field will be predefined and displayed if the selected Output Mode is '**Trend**').
   iv. **Period Column Name:** Enter a name for the column containing a period value. (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure
   a. Configure the following '**Behavior'** fields:
      i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)
      ii. **Beta:** Enter a valid double value in the given field for finding trend parameters. (Beta Range: 0-1.)
      iii. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.
   b. Configure the following '**Initial Values**' information**:**
      i. **Level:** Enter the initial value for the level. It is an optional field.

      ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.

      iii. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.

iv) Click '**APPLY**'

v) Run the workflow and open the '**RESULT**' tab after the console process gets completed
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.

vi) Click the '**VISUALIZATION**' tab.
vii) The result data will be displayed via the Time Line Chart.

Actual Beer_Sales vs Predicted Beer_Sales

4. R-Auto ARIMA

i) Select '**Trend**' option from the '**Output Mode**' drop-down menu.

ii) Configure the following fields in the '**Properties**' tab:

    a. **Output Information**

        i. **Output Mode**: Select a mode in which you want to display output data

            1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column '**Predicted Values**' will be added in the result view when '**Trend**' output mode has been selected.

            2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.

    b. **Column Selection**

        i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

    c. **Input Data Handling**

        i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.

        ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'

        iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field

        iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

    d. **New Column Information**

        i. **Predicted Column Name:** Enter a name for the column containing predicted values (This field will be predefined and displayed if the selected Output Mode is '**Trend**')

        v. **Period Column Name:** Enter a name for the column containing period value (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure
   a. Configure the following '**Behavior**' fields:
      i. **Alpha:** Enter a valid double value in the given field for smoothing observations (Alpha Range: 0<alpha<=1)
      ii. **Beta:** Enter a valid double value in the given field for finding trend parameters (Beta Range: 0-1)
      iii. **Gamma:** Enter a valid double value in the given field for finding a seasonal trend parameter (Gamma Range: 0-1)
      iv. **Seasonal:** Select a smoothing algorithm type from the drop-down list (Holtwinter's Exponential Smoothing algorithm)
      v. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation (The default value for this field is 2)
   b. Configure the following '**Initial Values**' information**:**
      i. **Level:** Enter the initial value for the level. It is an optional field.
      ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.
      iii. **Season:** Enter initial values for finding seasonal parameters. It will depend on the selected column. It is an optional field.
      iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.
iv) Click '**APPLY**'

v) Run the workflow and open the '**RESULT**' tab after the console process gets completed
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab
   c. A new column displaying the predicted values will be added to the result view

The following is the 'RESULT' tab display when 'Manual Arima' is Enabled
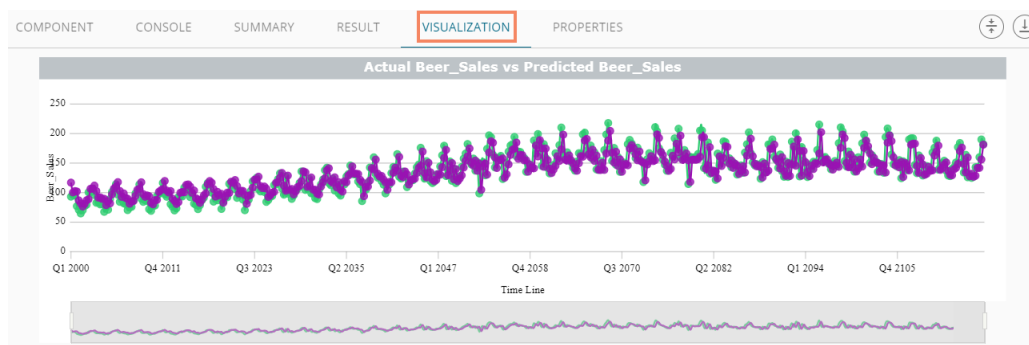

vi) Click the '**VISUALIZATION**' tab.
vii) The result data will be displayed via the Time Series Chart.


The following are the 'RESULT' and 'VISUALIZATION' tabs for the selected dataset when 'Manual Arima' is Disabled

Show 10 ▼ entries   Search:

Year	Month	Beer_Sales	periodName1	PredictedValues1
1965	January	93.2	Q1 2000	93.11
1965	February	96	Q2 2000	94.24
1965	March	95.2	Q3 2000	95.78
1965	April	77.1	Q4 2000	89.12
1965	May	70.9	Q1 2001	75.51
1965	June	64.8	Q2 2001	71.14
1965	July	70.1	Q3 2001	70.19
1965	August	77.3	Q4 2001	81.28
1965	September	79.5	Q1 2002	84.43
1965	October	100.6	Q2 2002	88.77

Showing 1 to 10 of 468 entries   Previous 1 2 3 4 5 ... 47 Next

Actual Beer_Sales vs Predicted Beer_Sales
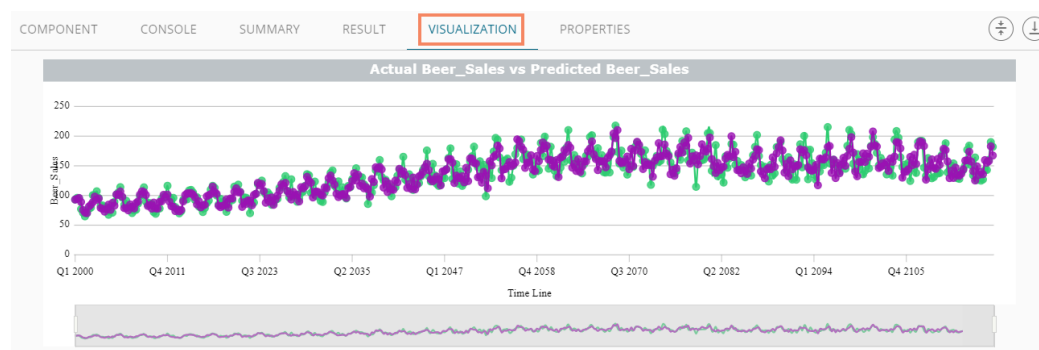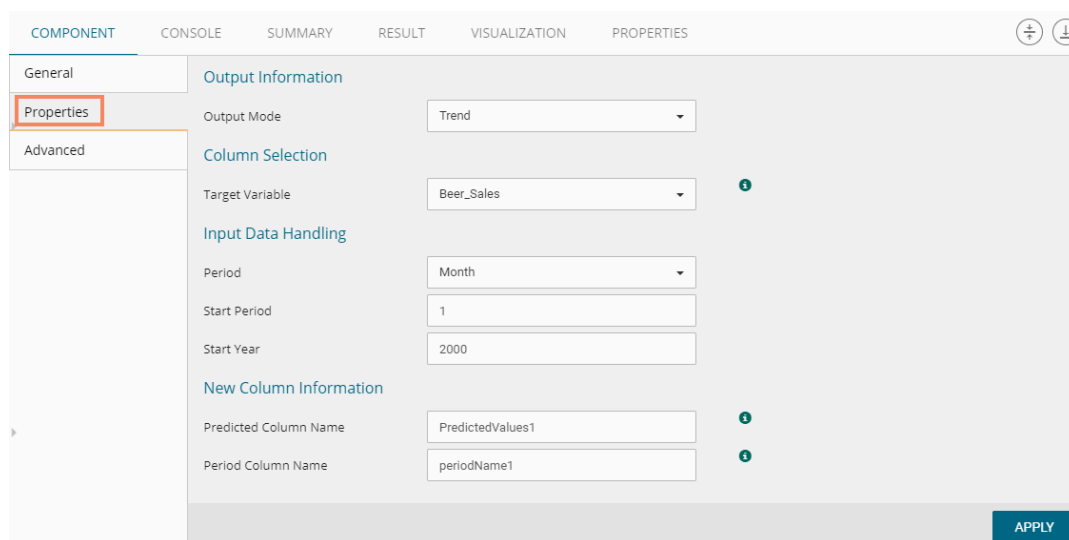
5. R-Auto Forecasting

i) Select '**Trend**' option from the '**Output Mode**' drop-down menu.

ii) Configure the following fields in the '**Properties**' tab:

  a. **Output Information**

    i. **Output Mode**: Select a mode in which you want to display output data

      1. **Trend**: Selecting this option will display source data along with predicted values for the given data set. A new column '**Predicted Values**' will be added in the result view when '**Trend**' output mode has been selected.

      2. **Forecast**: Selecting this option will display forecasted values for the given period. Results will be appended to the target column when '**Forecast**' output mode has been selected.

  b. **Column Selection**

    i. **Target Variable:** Select the target variable for which you want to apply forecasting analysis (First selected option gets selected by default. Only numerical columns are accepted.)

  c. **Input Data Handling**

    i. **Period:** Select period of forecasting by choosing any one option from the drop-down menu.

    ii. **Period Per Year**: This field appears only when the selected '**Period**' option is '**Custom.**'

    iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for '**Period**' field

iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter four digit value for selecting a year (E.g., 2000)

d. **New Column Information**

i. **Predicted Column Name:** Enter a name for the column containing predicted values (This field will be predefined and displayed only if the selected Output Mode is 'Trend').

ii. **Period Column Name:** Enter a name for the column containing period value (This field will be predefined, but users can change the value if needed).

iii) Click the '**Advanced**' tab and configure

a. Configure the following '**Behavior**' fields:

i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)

ii. **Beta:** Enter a valid double value in the given field for finding trend parameters. (Beta Range: 0-1.)

iii. **Gamma:** Enter a valid double value in the given field for finding seasonal trend parameters. (Gamma Range: 0-1.)

iv. **Seasonal:** Select a smoothing algorithm type from the drop-down list (Holtwinter's Exponential Smoothing algorithm)

v. **No. of Periodic Observation:** Enter the number of periodic observations required to start the calculation. The default value for this field is 2.

b. Configure the following '**Initial Values**' information:

i. **Level:** Enter the initial value for the level. It is an optional field.

ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.

iii. **Season:** Enter initial values for finding seasonal parameters. It will depend on the selected column. It is an optional field.

iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.

iv) Click '**APPLY**'

viii) Run the workflow and open the '**RESULT**' tab after the console process gets completed
  a. Click the dragged algorithm component onto the workspace.
  b. Click the '**RESULT**' tab.
  c. A new column with the **predicted values** will be added to the result data.

Year	Month	Beer_Sales	periodName1	PredictedValues1
1965	November	100.7		
1965	December	107.1		
1966	January	95.9	Jan 2001	95.38
1966	February	82.8	Feb 2001	82.47
1966	March	83.3	Mar 2001	82.98
1966	April	80	Apr 2001	79.4
1966	May	80.4	May 2001	79.77
1966	June	67.5	Jun 2001	66.58
1966	July	75.7	Jul 2001	70.15
1966	August	71.1	Aug 2001	78.37

Showing 11 to 20 of 468 entries   Previous 1 2 3 4 5 … 47 Next

v) Click the '**VISUALIZATION**' tab.
vi) The result data will be displayed via the time series chart.

Actual Beer_Sales vs Predicted Beer_Sales
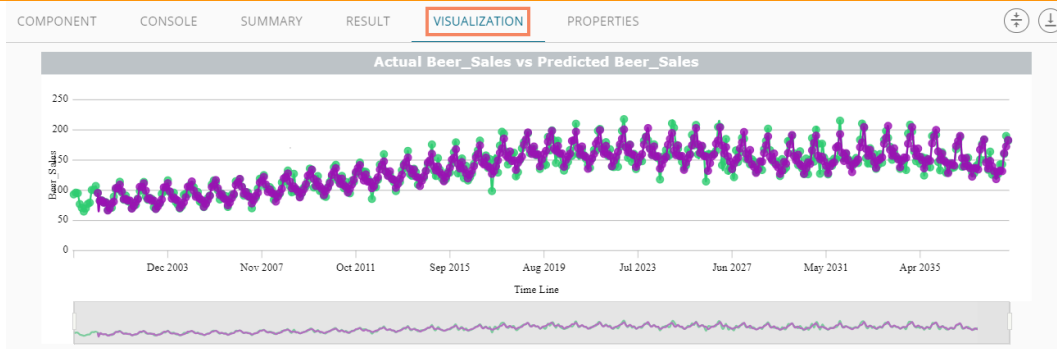
Note: Users can click the '**SUMMARY**' tab to view the model summary for the Forecasting models with 'Trend' as the output mode.

5.3.3. Association

This algorithm generates association rules discovering the recurrent patterns in large transactional data sets. It tries to understand the future trends of customers based on their previous purchases and assists the vendors to associate items or services together.
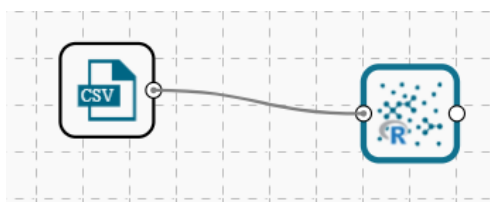
5.3.3.1. Market Basket Analysis

i) Drag the Market Basket Analysis component to the workspace and connect it with a configured data source.

ii) Configure the following fields in the '**Properties**' tab:
   a. Output Information
      i. **Output Mode**: Select a mode of display for output data
         1. Selecting '**Rules**' will display rules for the selected dataset
         2. Selecting '**Transaction**' will display the transaction IDs for the selected dataset
   b. Input Data Information
      i. **Input Data Format**: Select an input data format out of the following choices via the drop-down menu:
         1. **Tabular**
         2. **Transactions**
            As per the selected '**Input Data Format**,' the result view will be of 2 types.
      ii. **Item Columns:** Select the item columns on which you want to apply association rules/analysis. Choose at least one option from the drop-down menu. This field displays numerical and strings columns. It cannot display date columns.
      iii. **Transaction Id Column:** Select the column containing Transaction Ids to which you can apply the algorithm. (This field will be added when the selected 'Input Data Information' will be 'Transactions')

      Note: '**Transaction Id Column**' field appears only when the '**Transactions**' option has been selected from the '**Input Data Format**' drop-down menu.
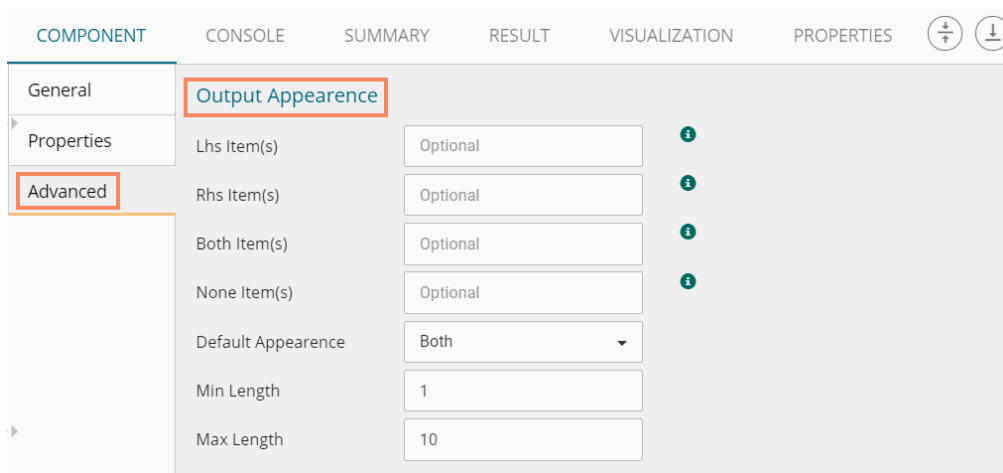
   c. Behavior

i. **Support:** Enter a value for the minimum support of an item. The default value for this field is 0.1

ii. **Confidence**: Select a value for the minimum confidence of the association (The default value for this field is 0.8)

Properties fields with '**Transactions**' as '**Input Data Information**'

iii) Click the '**Advanced**' tab and configure if required:

    a. Output Appearance

      i. **Lhs Item(s):** Enter item tags separated by a comma which should display on the left-hand side of rules or item sets

      ii. **Rhs Item(s):** Enter item tags separated by a comma which should display on the right-hand side of rules or item sets

      iii. **Both Item(s):** Enter item tags separated by a comma which should display on both sides of rules or item sets
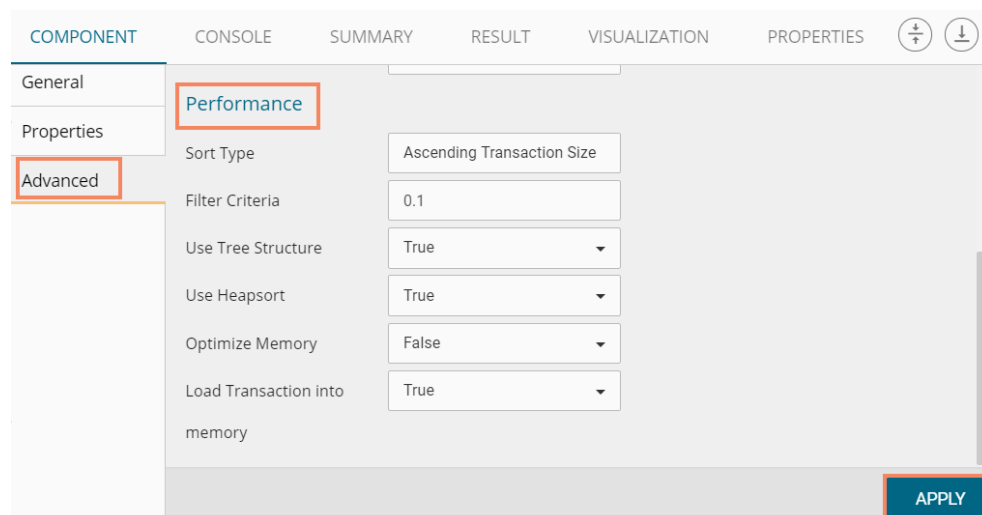
iv. **None Item(s):** Enter item tags separated by a comma which need not display in the rules or item sets

v. **Default Appearance:** Select the default appearance of the items out of the above-given choices using a drop-down menu

vi. **Min Length:** Set a minimum length value. The default value for this field is 1.

vii. **Max Length:** Set a maximum length value. The default value for this field is 10.

b. **Performance**
   i. **Sort Type:** Select a sort type using the drop-down menu for sorting items based on their frequency.
   ii. **Filter Criteria:** Enter an indicating numerical value for filtering unused items from transactions. The default value for this field is 0.1.
   iii. **Use Tree Structure:** Selecting **'True'** option from the drop-down menu will organize transaction as a prefix tree.
   iv. **Use Heapsort:** Selecting **'True'** option from the drop-down menu will use heapsort against quicksort for sorting transaction.
   v. **Optimize Memory:** Selecting **'True'** option from the drop-down menu will minimize memory usage instead of maximizing speed.
   vi. **Load Transaction into Memory:** Selecting **'True'** from the drop-down menu will load transactions into memory.

iv) Click **'Apply'**

v) Run the workflow after getting a success message.

vi) Users get directed to the '**Console**' tab displaying the progress of the process

COMPONENT	CONSOLE	SUMMARY

13/4/2018 - 16:44:38 : Process Initiated...

13/4/2018 - 16:44:39 : CSV0 is started.

13/4/2018 - 16:44:39 : CSV0 is completed.

13/4/2018 - 16:44:39 : R-Apriori1 is started.

13/4/2018 - 16:49:44 : R-Apriori1 is completed.

vii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.

viii) Result view will be of 2 types:
   a. '**Rules**' will be displayed as a first column in the result data (When the selected '**Output Mode**' option is '**Rules**').

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

Show 10 ▼ entries   Search:

Rules	Support	Confidence	Lift
{Affluence=Low} => {MetroPolitan=Yes}	0.12	1	1.66666666666667
{Affluence=Low} => {SKYBox=Sky+HD 2TB}	0.12	1	1.51515151515152
{Affluence=Very Low} => {MetroPolitan=No}	0.1	0.833333333333333	2.08333333333333
{Affluence=Mid Low} => {MetroPolitan=Yes}	0.12	0.857142857142857	1.42857142857143
{Affluence=Mid Low} => {SKYBox=Sky+HD 2TB}	0.12	0.857142857142857	1.2987012987013
{Demographiclifestyle=Liberal Opinion} => {HouseholdComposition=Men only HH}	0.12	0.857142857142857	2.52100840336134
{Demographiclifestyle=Liberal Opinion} => {MetroPolitan=Yes}	0.12	0.857142857142857	1.42857142857143
{Demographiclifestyle=Liberal Opinion} => {SKYBox=Sky+HD 2TB}	0.12	0.857142857142857	1.2987012987013
{Affluence=Mid} => {MetroPolitan=No}	0.12	0.857142857142857	2.14285714285714
{Demographiclifestyle=Terraced Melting Pot} => {HouseholdComposition=Men only HH}	0.14	0.875	2.57352941176471

Showing 1 to 10 of 85 entries   Previous 1 2 3 4 5 ... 9 Next

   b. '**Transaction_Id**' will be displayed as the second column in the result data (When the selected '**Output Mode**' option is '**Transaction**').

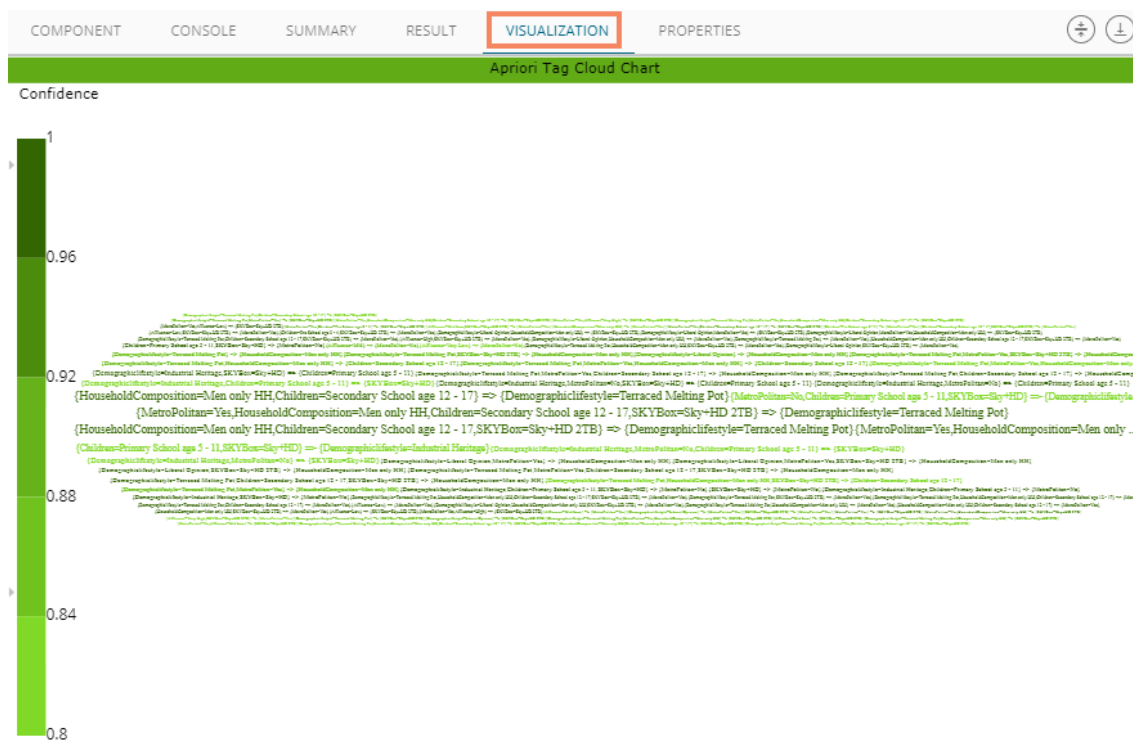   The matching rules for the selected items will be displayed through the '**Matching_Rules**' column.

	COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	

Show 10 entries   Search:

Items	Transaction_Id	Matching_Rules
1	396	103
2	434	
3	486	1455
4	576	1392
5	664	1176
6	700	382

Showing 1 to 6 of 6 entries   Previous   1   Next

ix)   Click the '**VISUALIZATION**' tab.
x)   The result data will be displayed via the Word Cloud chart.
   a.   Result View for the '**Rules**' output mode.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Apriori Tag Cloud Chart

Confidence

b. Result view when 'Transactions' is the output mode.

{Bubble Gum,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets}{Chicken,Detergent} => {Lightbulb}{Bubble Gum,Chocolate,Cole Slaw,Magazine,Strawberry} => {Yams}
{Chocolate,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets}{Chicken,Waffles} => {TV Dinner}{Beets,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}
{Bubble Gum,Cole Slaw,Detergent,Shrimp,Toothpaste} => {Beets}{TV Dinner,Waffles} => {Lightbulb}{Lightbulb,Waffles} => {TV Dinner}{TV Dinner} => {Water}
{Detergent,Waffles} => {Water}{Beets,Bubble Gum,Chocolate,Detergent,Shrimp} => {Toothpaste}{Chicken,TV Dinner} => {Lightbulb}{Chicken,Lightbulb} => {TV Dinner}
{Waffles,Water} => {Lightbulb}{Chicken,Detergent} => {Water}{Beets,Bubble Gum,Chocolate,Cole Slaw,Shrimp,Strawberry} => {Toothpaste}{Chocolate,Cole Slaw,Detergent,Shrimp,Toothpaste} => {Be..
{TV Dinner,Waffles} => {Water}{Detergent,Water} => {Lightbulb}{Detergent,Lightbulb} => {Water}{Bubble Gum,Chocolate,Shrimp,Strawberry,Toothpaste} => {Beets}
{Beets,Bubble Gum,Cole Slaw,Detergent,Strawberry} => {Toothpaste}{Beets,Bubble Gum,Chocolate,Cole Slaw,Shrimp} => {Toothpaste}{Waffles,Water} => {TV Dinner}
{Bubble Gum,Cole Slaw,Detergent,Strawberry,Toothpaste} => {Beets}{TV Dinner,Water} => {Lightbulb}{Lightbulb,Water} => {TV Dinner}{Lightbulb,TV Dinner} => {Water}
{Bubble Gum,Cole Slaw,Detergent,Shrimp,Strawberry} => {Beets}{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent} => {Toothpaste}{Chocolate,Cole Slaw,Detergent,Strawberry,Toothpaste} => {Beet..
{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp} => {Beets}{Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Beets}{Bubble Gum,Chocolate,Detergent,Shrimp,Strawberry} => {Beets}
{Bubble Gum,Chocolate,Cole Slaw,Magazine,Strawberry,Waffles} => {Yams}{Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Detergent,Shrimp,Strawberry} => ..
{Bubble Gum,Cole Slaw,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets}{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}{Beets} => {Toothpaste}{Toothpaste} => {Beets}
{Bubble Gum,Chocolate,Cole Slaw,Shrimp,Strawberry,Toothpaste} => {Beets}{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp} => {Toothpaste}{Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry,Too..
{Lightbulb} => {TV Dinner}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Strawberry,Toothpaste} => {Beets}{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent,Strawberry} => {Toothpaste}
{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets} {Water} => {TV Dinner}{Water} => {Lightbulb}{Lightbulb} => {Water}{TV Dinner} => {Lightbulb}{Water} => {Chicken}
{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}
{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Beets}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,To..
{Beets,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets}{Beets,Bubble Gum,Chocolate,Detergent,Shrimp,Strawb..
{Bubble Gum,Chocolate,Cole Slaw,Detergent,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Cole Slaw,Shrimp,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp} => {T..
{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry} => {Bee..
{Bubble Gum,Chocolate,Cole Slaw,Strawberry,Toothpaste} => {Beets}{Beets,Bubble Gum,Chocolate,Cole Slaw,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Cole Slaw,Detergent,Toothpaste} => {..
{Beets,Chocolate,Cole Slaw,Detergent,Strawberry} => {Toothpaste}{Bubble Gum,Chocolate,Detergent,Strawberry,Toothpaste} => {Tooth..
{Chicken,Water} => {TV Dinner}{Chicken,TV Dinner} => {Water}{Bubble Gum,Chocolate,Cole Slaw,Shrimp,Toothpaste} => {Beets}{Detergent,Water} => {TV Dinner}
{Detergent,TV Dinner} => {Water}{Chocolate,Cole Slaw,Shrimp,Strawberry,Toothpaste} => {Beets}{Chicken,Water} => {Lightbulb}{Chicken,Lightbulb} => {Water}
{Beets,Chocolate,Cole Slaw,Shrimp,Strawberry} => {Toothpaste}{Lightbulb,Waffles} => {Water}{Beets,Bubble Gum,Chocolate,Shrimp,Strawberry} => {Toothpaste}
{Bubble Gum,Cole Slaw,Shrimp,Strawberry,Toothpaste} => {Beets}{Chicken,Waffles} => {Water}{Beets,Chocolate,Cole Slaw,Detergent,Shrimp} => {Toothpaste}
{Bubble Gum,Chocolate,Detergent,Shrimp,Toothpaste} => {Beets}{Detergent,TV Dinner} => {Lightbulb}{Detergent,Lightbulb} => {TV Dinner}
{Beets,Bubble Gum,Cole Slaw,Detergent,Shrimp} => {Toothpaste}{Chicken,Detergent} => {TV Dinner}{Beets,Chocolate,Detergent,Shrimp,Strawberry} => {Toothpaste}
{Cole Slaw,Detergent,Shrimp,Strawberry,Toothpaste} => {Beets}{Detergent,Waffles} => {TV Dinner}{Beets,Bubble Gum,Detergent,Shrimp,Strawberry} => {Toothpaste}
{Bubble Gum,Chocolate,Cole Slaw,Strawberry,Waffles} => {Yams}{Chicken,Waffles} => {Lightbulb}{Chocolate,Cole Slaw,Magazine,Strawberry,Waffles} => {Yams}
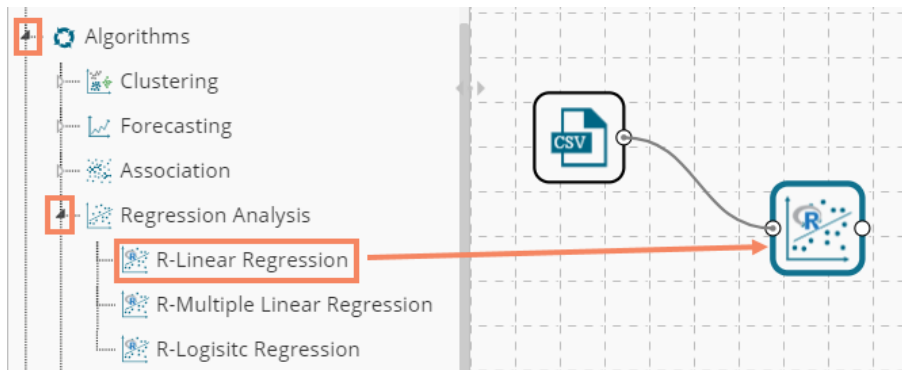
5.3.4. Regression Analysis

This algorithm is used to determine how an individual variable influences another variable using an exponential function. It finds a trend in the dataset applying univariate regression analysis.

There are three subtypes provided under **'Regression Analysis':**

5.3.4.1. R-Linear Regression

i) Drag the R-linear Regression component to the workspace and connect it with a configured data source.

ii) Configure the following fields in the '**Properties**' tab:
   a. Column Selection
      i. **Dependent Column**: Select the target column on which the regression analysis gets applied
      ii. **Independent Column**: Select the required input columns against which the regression analysis will be applied to the target column
   b. New Column Information
      i. **Predicted Column Name**: Enter a name for the new column containing the predicted values
   c. Model Tuning
      i. Enable Validation: Use a checkmark to enable validation tab
      ii. XG Boosting: Use a checkmark in the box to enable XG Boosting

Scenario-1- when Validation and XG Boosting are enabled

Scenario-2- when Validation and XG Boosting are disabled

Scenario-3- when Validation is enabled, but XG Boosting is disabled

iii) **Click the 'Validation' tab and configure it:**
   a. **Model Selection (when XG Boosting is enabled)**
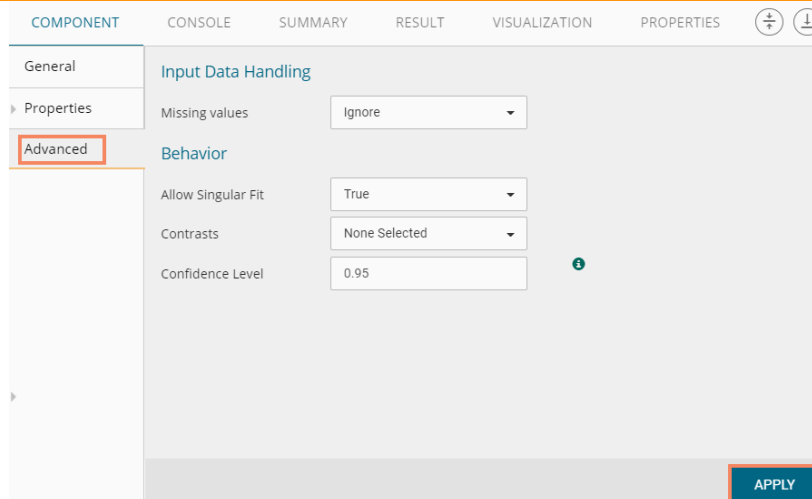   i. **Number of folds:** Enter a number deciding the creation of folds in a model

Validation tab when XG Boosting is disabled

a. **Model Selection**
   i. **Model Selection Method:** Select a Model Method using the drop-down menu
   ii. **Number of folds:** Enter a number deciding the creation of folds in a model

iv) Click the '**Advanced**' tab and configure if required:
   Advanced tab when XG Boosting and Validation are disabled

a. **Input Data Handling**
   i. **Missing Values**: Select a method to deal with missing values from the drop-down menu
      1. **Ignore**: Select this option to skip the records containing missing values from the dependent and independent columns.
      2. **Keep**: Select this option to retain the records containing missing values while performing the calculation.
      3. **Stop**: Select this option to stop the algorithm application if a value is missing in any column.

b. **Behavior**
   i. **Allow Singular Fit**: Select an option for providing value to the Boolean Column
      1. **True:** Select this option to ignore aliased coefficients from the coefficient covariance matrix.
      2. **False:** Select this option to show an error in a model containing aliased coefficients
   ii. **Contrasts**: Select this option to display a list of contrast items that can be used for some variables in the model.
   iii. **Confidence Level**: Enter a value specifying accuracy (Confidence Level) of predictions for the algorithm. This field takes 0.95 as the default value.

Advanced Tab when XG Boosting is disabled, but Validation is enabled

c. **Intercept Parameter**
   i. Intercept Value: Enter an intercept value
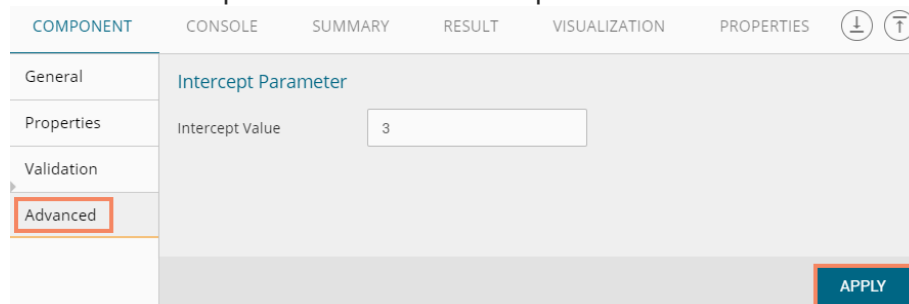
Advanced Tab when XG Boosting and Validation is enabled or XG Boosting is enabled, but Validation is disabled

a. **Boosting Parameter**
   i. No. of Iterations: Enter number of iterations

v) Click '**APPLY**'

Note: Model containing aliased coefficients signifies that the square matrix x*x is singular.

vi) Run the workflow
vii) Users will be redirected to the '**CONSOLE**' tab.

viii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.
      i. A new column '**Predicted Values1**' gets added to the result data displaying the predicted values.
      Result when Validation and XG Boosting are disabled

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	5.74445883693983
2	4.9	3	1.4	0.2	setosa	5.85613936750478
3	4.7	3.2	1.3	0.2	setosa	5.8114671552788
4	4.6	3.1	1.5	0.2	setosa	5.83380326139179
5	5	3.6	1.4	0.2	setosa	5.72212273082684
6	5.4	3.9	1.7	0.4	setosa	5.65511441248787
7	4.6	3.4	1.4	0.3	setosa	5.76679494305282
8	5	3.4	1.5	0.2	setosa	5.76679494305282
9	4.4	2.9	1.4	0.2	setosa	5.87847547361777
10	4.9	3.1	1.5	0.1	setosa	5.83380326139179

Showing 1 to 10 of 150 entries   Previous   1   2   3   4   5   ...   15   Next

Result when XG Boosting enabled, and Validation enabled or disabled

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	3.86565351486206
2	4.9	3	1.4	0.2	setosa	4.03112602233887
3	4.7	3.2	1.3	0.2	setosa	4.03112602233887
4	4.6	3.1	1.5	0.2	setosa	4.03112602233887
5	5	3.6	1.4	0.2	setosa	3.86565351486206
6	5.4	3.9	1.7	0.4	setosa	3.86565351486206
7	4.6	3.4	1.4	0.3	setosa	3.86565351486206
8	5	3.4	1.5	0.2	setosa	3.86565351486206
9	4.4	2.9	1.4	0.2	setosa	4.03112602233887
10	4.9	3.1	1.5	0.1	setosa	4.03112602233887

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

ix)   Click the '**VISUALIZATION**' tab.

x)   The result data gets displayed via the Scatter Plot with Regression line chart.
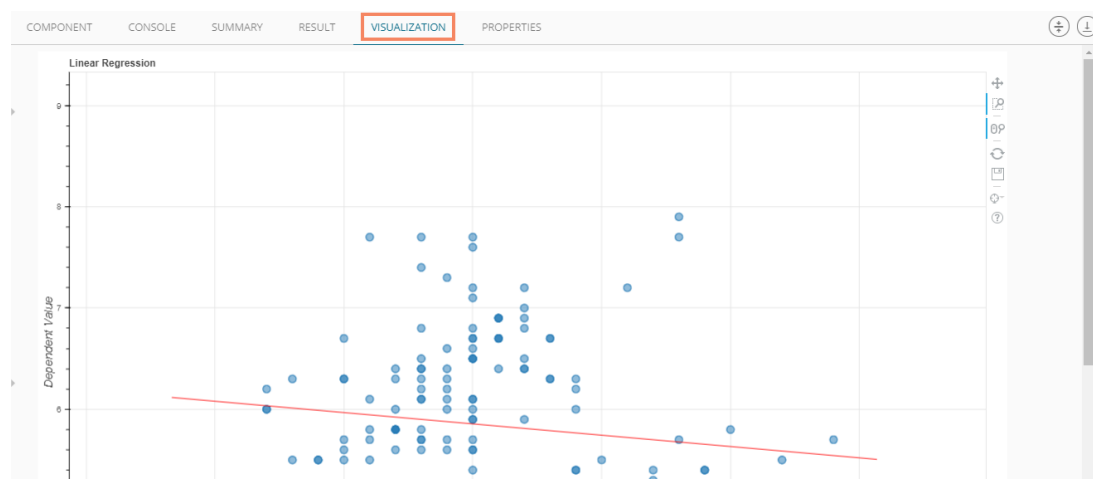
Note: '**Behavior**' fields provided under '**Advanced**' section differs as per the algorithm sub-type. '**Input Data Handling**' remains the same for all the provided Regression types. Hence, only the '**Advanced**' tab is explained below for the remaining R sub-algorithms provided under '**Regression.**'

5.3.4.2. R-Multiple Linear Regression

i)   Drag the R-Multiple Linear Regression component to the workspace and connect it with a configured data source
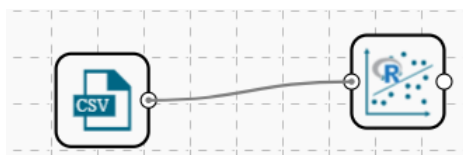
ii)   Configure the '**Properties**' tab
    a. Column Selection

      i. **Dependent Column**: Select the target column on which the regression analysis gets applied

      ii. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column

b. **New Column Information**

      i. **Predicted Column Name**: Enter a name for the new column containing the predicted values

c. **Model Tuning**

      i. Enable Validation: Use a checkmark to enable validation tab

      ii. XG Boosting: Use a checkmark in the box to enable XG Boosting

Scenario 1: When Validation is enabled, and XG Boosting is disabled

Scenario 2: When Validation and XG Boosting are enabled

Scenario 3: When Validation is disabled, but XG Boosting is enabled

Scenario 4: When Validation and XG Boosting are disabled

iii) **Validation**
  a. **Model Selection (When XG Boosting is disabled)**
    i. **Model Selection Method:** Select a model selection method using the drop-down menu
    ii. **Number of folds:** Enter a value for the number of folds

Validation when XG Boosting is enabled

  i. **Number of folds:** Enter a value for the number of folds

iv) Click the '**Advanced**' tab and configure if required:
When Validation and XG Boosting are disabled
a. Input Data Handling
    i. **Missing Values**: Select a method to deal with missing values (via the drop-down menu).
        1. **Ignore**: Select this option to skip the records containing missing values from the dependent and independent columns.
        2. **Keep**: Select this option to retain the records containing missing values while performing the calculation.
        3. **Stop**: Select this option to stop the algorithm application if a value is missing in any column.

b. Behavior
    i. **Confidence Level**: Enter a value specifying accuracy (confidence level) of Predictions for the algorithm. This field takes 0.95 as the default value.

When Validation is enabled and XG Boosting disabled

a. Intercept Parameter
    i. **Intercept Value:** Enter an intercept value

When XG Boosting is enabled with either Validation is enabled or disabled

a. **Boosting Parameter**
    i. **No. of Iterations:** Enter number suggesting no. of iterations

v) Click '**APPLY**'
vi) Run the workflow
vii) Users will be redirected to the '**CONSOLE**' tab.

viii) Follow the below-given steps to display the result view:
    a. Click the dragged algorithm component onto the workspace.
    b. Click the '**RESULT**' tab.
ix) A new column is added to the result data.
    a. Result when XG Boosting is disabled

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	5.05687661229313
2	4.9	3	1.4	0.2	setosa	4.73646963139815
3	4.7	3.2	1.3	0.2	setosa	4.79026561122786
4	4.6	3.1	1.5	0.2	setosa	4.86784805813776
5	5	3.6	1.4	0.2	setosa	5.11270992950984
6	5.4	3.9	1.7	0.4	setosa	5.42179124865001
7	4.6	3.4	1.4	0.3	setosa	4.93396846048268
8	5	3.4	1.5	0.2	setosa	5.05105863638273
9	4.4	2.9	1.4	0.2	setosa	4.65903420261356
10	4.9	3.1	1.5	0.1	setosa	4.90350163954186

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

b. Result when XG Boosting is enabled, and Validation is enabled or disabled (No visualization is available for this result data)

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	3.50660634040833
2	4.9	3	1.4	0.2	setosa	3.50660634040833
3	4.7	3.2	1.3	0.2	setosa	3.50660634040833
4	4.6	3.1	1.5	0.2	setosa	3.50660634040833
5	5	3.6	1.4	0.2	setosa	3.50660634040833
6	5.4	3.9	1.7	0.4	setosa	3.50660634040833
7	4.6	3.4	1.4	0.3	setosa	3.50660634040833
8	5	3.4	1.5	0.2	setosa	3.50660634040833
9	4.4	2.9	1.4	0.2	setosa	3.50660634040833
10	4.9	3.1	1.5	0.1	setosa	3.50660634040833

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

x)   Click the 'VISUALIZATION' tab.
xi)   The Scatterplot with Regression Line Chart appears to display the result data.
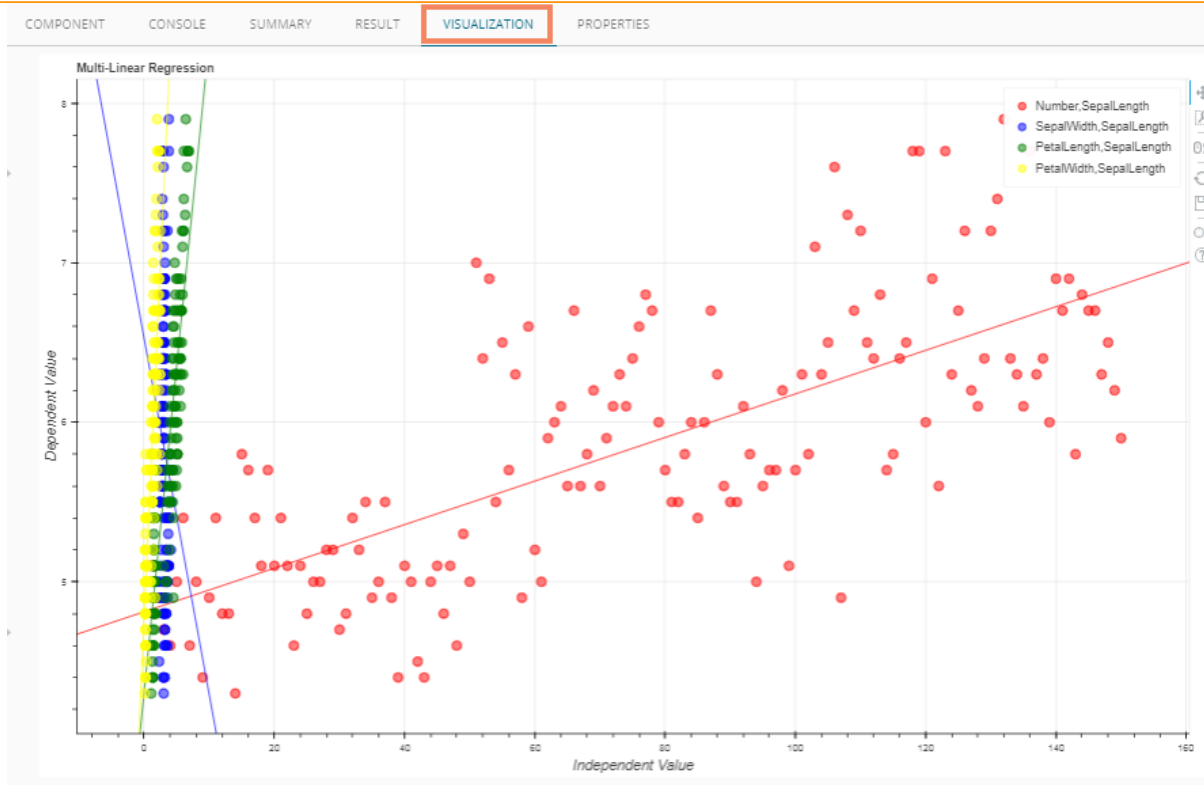
5.3.4.3. R-Logistic Regression

i) Drag the R-Logistic Regression component to the workspace and connect it with a configure data source.

ii) Configure the **'Properties'** tab.
   a. Column Selection
      i. **Dependent Column**: Select the target column on which the regression analysis gets applied
      ii. **Independent Column**: Select the required input columns against which the regression analysis to the target column gets applied
   b. New Column Information
      i. **Predicted Column Name**: Enter a name for the new column containing the predicted values
   c. Model Tuning
      i. Enable Validation: Use a checkmark to enable validation tab
      ii. XG Boosting: Use a checkmark in the box to enable XG Boosting
   Scenario 1: XG Boosting and Validation are disabled

Scenario 2: When Validation is enabled, and XG Boosting is disabled

Scenario 3: When Validation is disabled, and XG Boosting is enabled

Scenario 4: When Validation and XG Boosting are enabled

iii) **Validation Tab**
Validation tab when XG Boosting is disabled

    a. Model Selection
       i. **Model Selection Method:** Select a model selection method from the drop-down menu
       ii. **Number of folds:** Enter a value for the number of folds

Validation tab when XG Boosting is enabled

    b. Model Selection
       i. **Number of folds:** Enter a value for the number of folds
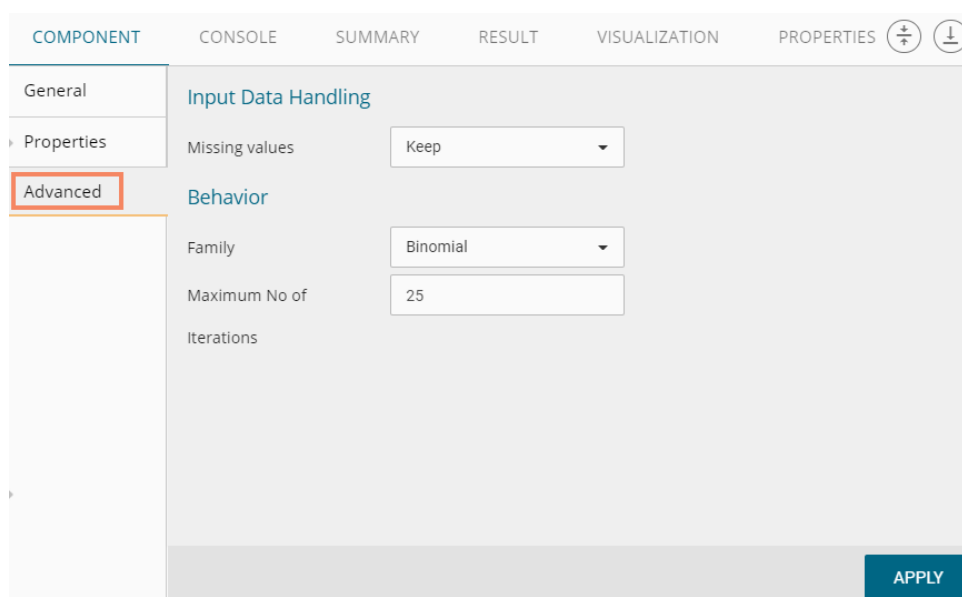
iv) Click the '**Advanced**' tab and configure if required:
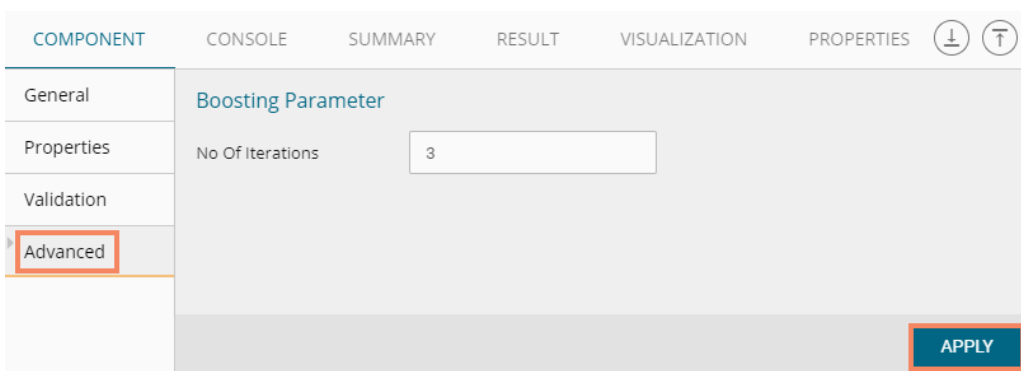**Advanced Tab when Validation and XG Boosting are disabled**
    a. Input Data Handling
       i. **Missing Values**
          1. **Ignore**: Selecting this option will skip the records containing missing values in the columns

2. **Keep:** Select this option to retain the records containing missing values while performing the calculation
3. **Stop:** Select this option to **stop (not allow)** the records containing missing values while performing the calculation

b. **Behavior**
   i. **Family:** Select an option from the drop-down list
      1. Binomial
      2. Poisson
      3. Gaussian
      4. Gamma
      5. Quasi
      6. Quasi-Poisson
      7. Quasibinomial
   ii. **Maximum No. of Iterations:** Enter a valid integer value allowed to calculate the algorithm coefficient. The default values for this field is 25.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Advanced

Input Data Handling

Missing values   [Keep ▾]

Behavior

Family   [Binomial ▾]

Maximum No of   [25]
Iterations

APPLY

Advanced Tab with Validation enabled and XG Boosting disabled

a. **Input Data Handling**
   i. **Missing Values:**
      1. **Ignore:** Select this option to skip the records containing missing values in the columns
      2. **Keep:** Select this option to retain the records containing missing values while performing the calculation
      3. **Stop:** Select this option to **stop (not allow)** the records containing missing values while performing the calculation
b. **Behavior**
   i. **Contrast:** Select an option from the following list
      1. None Selected
      2. Contr.treatment
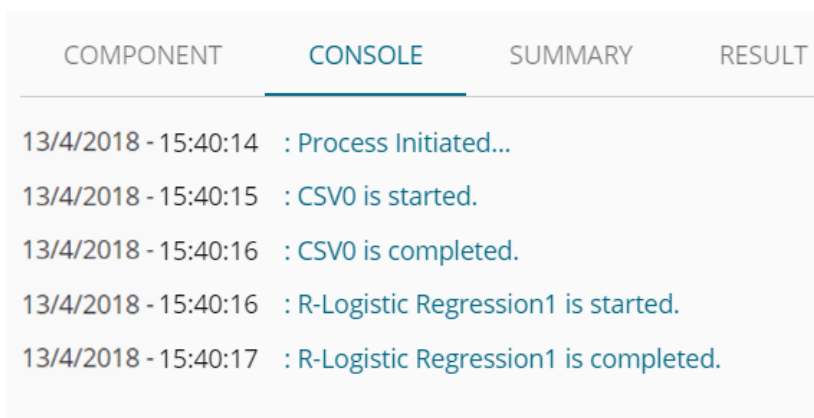   3. Contr.poly
      4. Contr.sum
      5. Contr.helmert
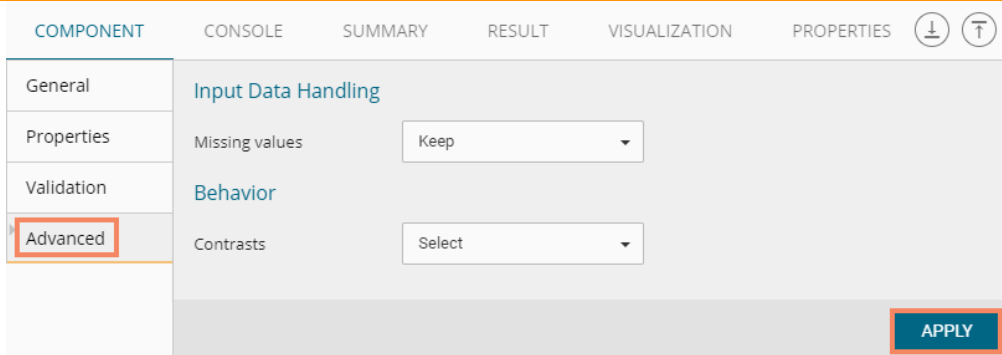
Advanced tab when XG Boosting is enabled and Validation is enabled or disabled

a. **Boosting Parameter**
   i. **No. of Iterations:** Enter a number suggesting no. of Iterations

v) Click '**APPLY**'
vi) Run the workflow
vii) Users will be redirected to the '**CONSOLE**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT

13/4/2018 - 15:40:14 : Process Initiated...

13/4/2018 - 15:40:15 : CSV0 is started.

13/4/2018 - 15:40:16 : CSV0 is completed.

13/4/2018 - 15:40:16 : R-Logistic Regression1 is started.

13/4/2018 - 15:40:17 : R-Logistic Regression1 is completed.

viii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab
ix) A new column is inserted into the result Data.

Result when XG Boosting is disabled

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

competitorname	chocolate	fruity	caramel	peanutyalmondy	nougat	crispedricewafer	hard	bar	pluribus	sugarpercent	pricepercent	winpercent	PredictedValues1
100 Grand	1	0	1	0	0	1	0	1	0	0.73199999	0.86000001	66.971725	0.999999999997099
3 Musketeers	1	0	0	0	1	0	0	1	0	0.60399997	0.51099998	67.602936	0.999999999997099
One dime	0	0	0	0	0	0	0	0	0	0.011	0.116	32.261086	2.90070146547081e-12
One quarter	0	0	0	0	0	0	0	0	0	0.011	0.51099998	46.116505	2.90070146546389e-12
Air Heads	0	1	0	0	0	0	0	0	0	0.90600002	0.51099998	52.341465	2.90070146546978e-12
Almond Joy	1	0	0	1	0	0	0	1	0	0.465	0.76700002	50.347546	0.999999999997099
Baby Ruth	1	0	1	1	1	0	0	1	0	0.60399997	0.76700002	56.914547	0.999999999997099
Boston Baked Beans	0	0	0	1	0	0	0	0	1	0.31299999	0.51099998	23.417824	2.90070146546935e-12
Candy Corn	0	0	0	0	0	0	0	0	1	0.90600002	0.32499999	38.010963	2.90070146546818e-12
Caramel Apple Pops	0	1	1	0	0	0	0	0	0	0.60399997	0.32499999	34.517681	2.90070146546964e-12

Showing 1 to 10 of 85 entries   Previous 1 2 3 4 5 ... 9 Next

Result when XG Boosting is enabled

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

competitorname	chocolate	fruity	caramel	peanutyalmondy	nougat	crispedricewafer	hard	bar	pluribus	sugarpercent	pricepercent	winpercent	PredictedValues1
100 Grand	1	0	1	0	0	1	0	1	0	0.73199999	0.86000001	66.971725	0.787244617938995
3 Musketeers	1	0	0	0	1	0	0	1	0	0.60399997	0.51099998	67.602936	0.787244617938995
One dime	0	0	0	0	0	0	0	0	0	0.011	0.116	32.261086	0.284415751695633
One quarter	0	0	0	0	0	0	0	0	0	0.011	0.51099998	46.116505	0.461076647043228
Air Heads	0	1	0	0	0	0	0	0	0	0.90600002	0.51099998	52.341465	0.222202509641647
Almond Joy	1	0	0	1	0	0	0	1	0	0.465	0.76700002	50.347546	0.787244617938995
Baby Ruth	1	0	1	1	1	0	0	1	0	0.60399997	0.76700002	56.914547	0.787244617938995
Boston Baked Beans	0	0	0	1	0	0	0	0	1	0.31299999	0.51099998	23.417824	0.284415751695633
Candy Corn	0	0	0	0	0	0	0	0	1	0.90600002	0.32499999	38.010963	0.529607653617859
Caramel Apple Pops	0	1	1	0	0	0	0	0	0	0.60399997	0.32499999	34.517681	0.222202509641647

Showing 1 to 10 of 85 entries   Previous 1 2 3 4 5 ... 9 Next

x)   Click the '**VISUALIZATION**' tab.
xi)   The result data displayed via the chart displaying the Scatter Plot with a Regression Line.
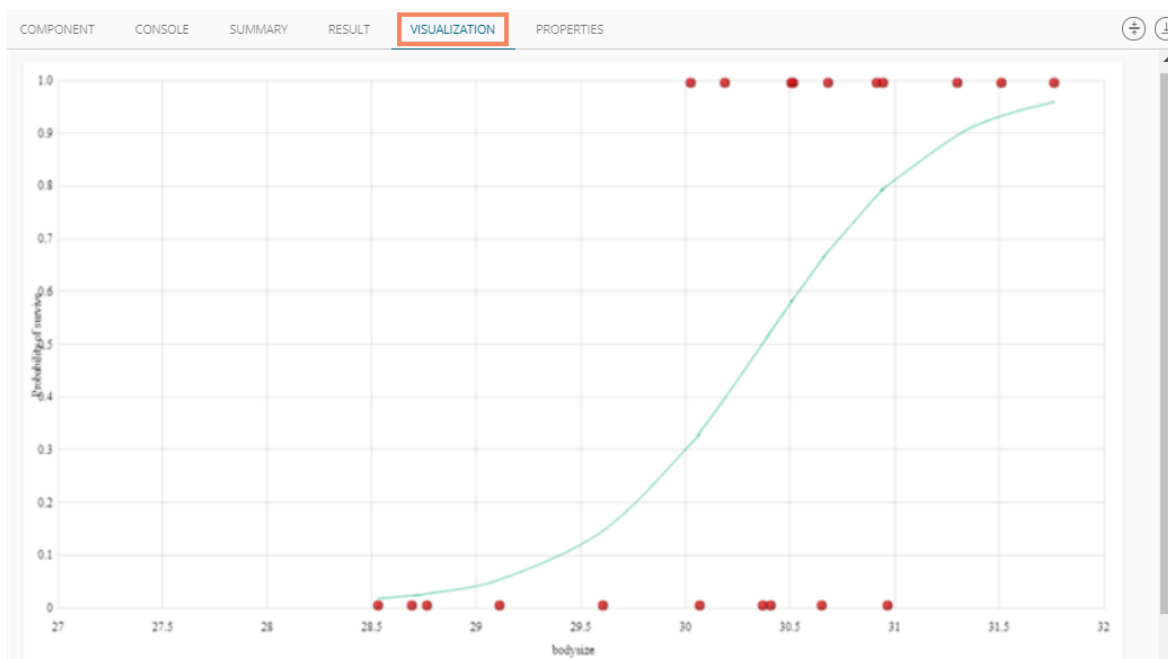
Note: No visualization is available for the models in which XG Boosting is enabled.
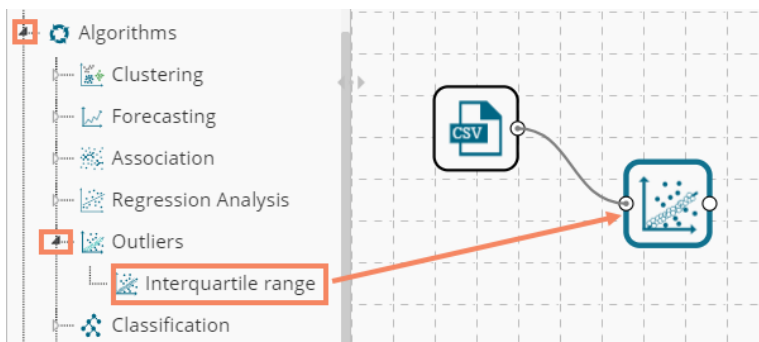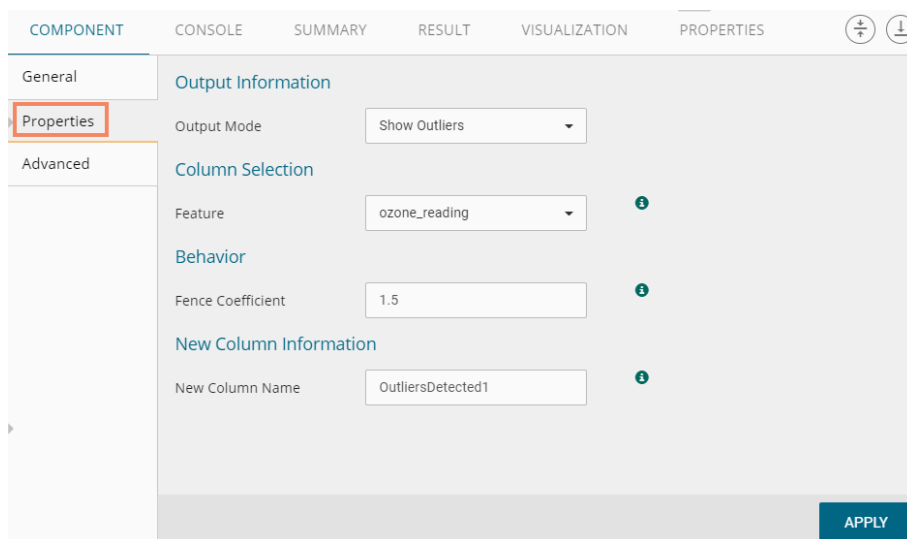
5.3.5. Outliers

This algorithm is used to discover patterns in data set that do not follow the expected behavior. It lists the outlying values based on the statistical distribution between the first and third quartiles. Interquartile Range has been provided as a sub-algorithm type.

5.3.5.1. Interquartile Range

i) Drag the Interquartile Range component to the workspace and connect it to a configured data source.

ii) Configure the following fields in the **'Properties'** tab:

a. **Output Information**
   i. **Output Mode**: Select a mode of display for output data.
      1. **Show Outlier**: Select this option to add a Boolean column to the input data identifying whether the resultant value is an outlier.
      2. **Remove Outlier**: Select this option to remove outlying values from the input data.

b. **Column Selection**
   i. **Feature:** Select an input column that can be used to perform the analysis.

c. **Behavior**
   i. **Fence Coefficient:** Enter the permissible deviation limit for values from the Interquartile Range (The default value for this field is 1.5)

d. **New Column Information**
   i. **New Column Name:** Enter a name for the new column containing the predicted values   (This column appears only when **'Show Outliers'** is selected as an **Output Mode).**

Properties fields with the 'Remove Outliers' option selected to display Output Information
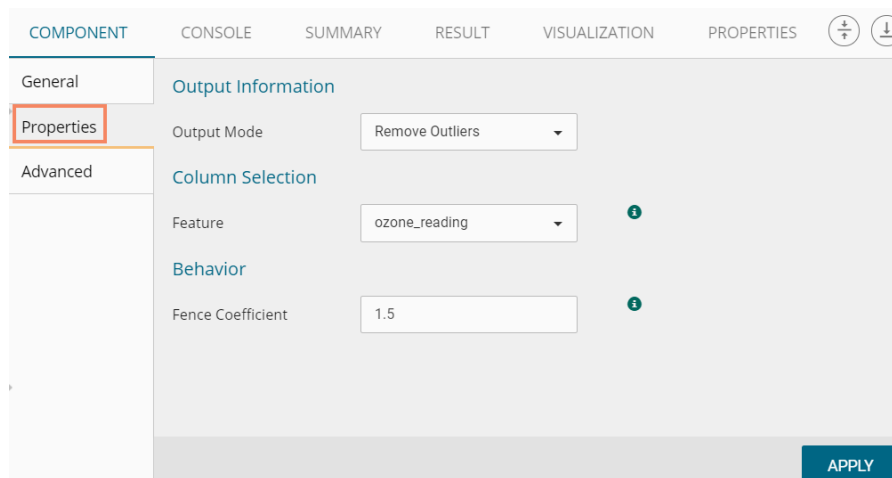
iii) Click the '**Advanced**' tab and configure if required:
a. **Input Data Handling**
   i. **Missing Values**: Select a method to deal with missing values from the drop-down menu.
      1. **Ignore**: Select this option to skip the records containing missing values in the columns.
      2. **Stop**: Select this option to stop the application of the algorithm if a value is missing in any column.
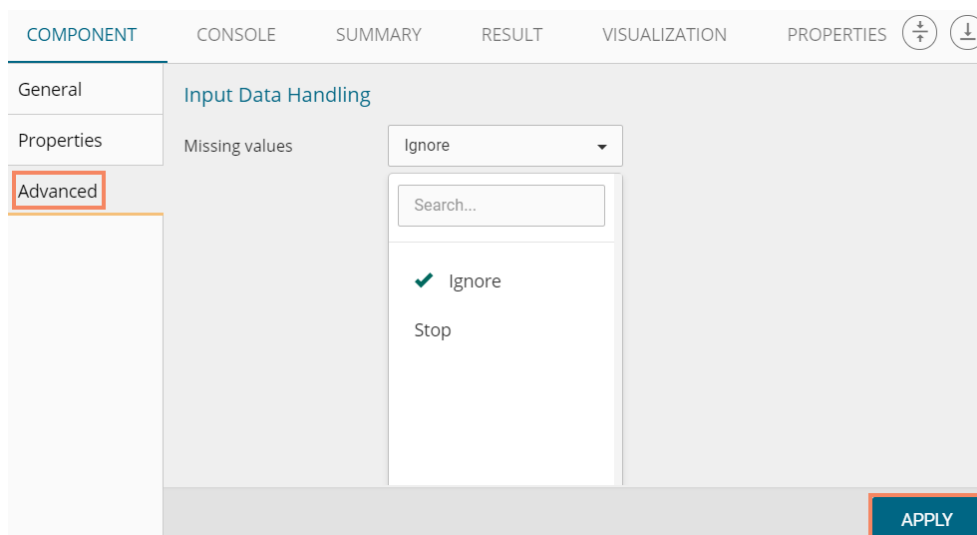
iv) Click '**APPLY**'
v) Run the workflow
vi) Users will be redirected to the '**CONSOLE**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT |

13/4/2018 - 18:48:15   : Process Initiated...

13/4/2018 - 18:48:18   : CSV0 is started.

13/4/2018 - 18:48:19   : CSV0 is completed.

13/4/2018 - 18:48:19   : Interquartile range1 is started.

13/4/2018 - 18:48:19   : Interquartile range1 is completed.

vii)   Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.
   A new column '**OutliersDetected1**' displays in the result data (If '**Show Outliers**' option
   has been selected).

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Search:

k	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	Inversion_base_height	Pressure_gradient	Inversion_temperature	Visibility	OutliersDetected1
	4.1	5860	0	25	60	61.52	5000	-38	63.5	140	FALSE
	10.99	5900	0	24	62	62.6	5000	-36	60.08	150	FALSE
	5.91	5850	5	41	65	59.54	2014	-20	69.98	200	FALSE
	8.3	5780	3	50	66	59.72	436	1	70.34	4	FALSE
	14.17	5790	0	76	66		830	3	66.02	40	FALSE
	17.61	5780	2	82	63		1112	-8	66.38	30	FALSE
	11.89	5770	2	81	62	60.62	1210	-17	67.82	30	FALSE
	9.09	5750	2	85	60	59.72	501	-22	70.88	2	FALSE
	7.01	5780	5	76	63	60.44	875	-15	68.9	0	FALSE
	13.9	5790	5	66	60		1601	7	62.06	30	FALSE

Previous   1 ... 32   33   34 ... 37   Next

viii)   Click the '**VISUALIZATION**' tab.
ix)   The result data is displayed via the Box Plot chart.

OR
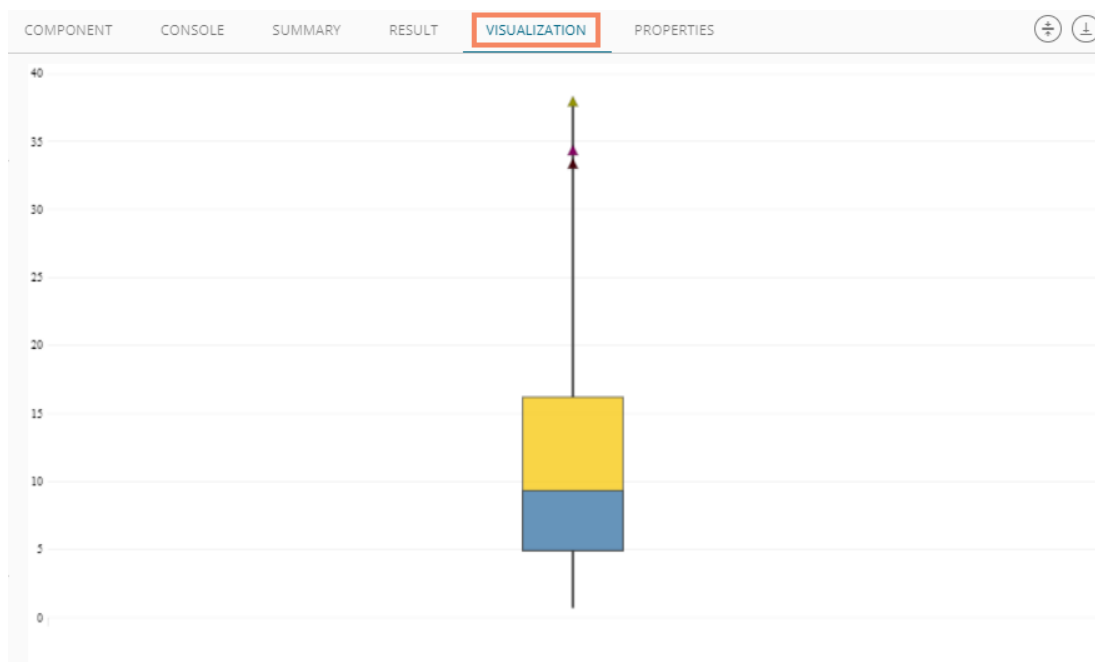
Outliers column is removed from the result data (If '**Remove Outliers**' option has been selected).

	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	Inversion_base_height	Pressure_gradient	Inversion_temperature	Visibility
	3.01	5480	8	20			5000	-15	30.56	200
	3.2	5660	6		38			-14		300
	2.7	5710	4	28	40		2693	-25	47.66	250
	5.18	5700	3	37	45		590	-24	55.04	100
	5.34	5760	3	51	54	45.32	1450	25	57.02	60
	5.77	5720	4	69	35	49.64	1568	15	53.78	60
	3.69	5790	6	19	45	46.4	2631	-33	54.14	100
	3.89	5790	3	25	55	52.7	554	-28	64.76	250
	5.76	5700	3	73	41	48.02	2083	23	52.52	120
	6.94	5700	3	59	44		2654	-2	48.38	120

Previous 1 2 3 4 5 … 36 Next

Click the '**VISUALIZATION**' to see the result data via the Box Plot chart.

5.3.6. Classification

This algorithm categorizes a new observation by a trained set of data that contains observations from the known category. It compares each new observation to previous observations using means of similarity or distance.

5.3.6.1. R-CNR Tree

The R-CNR Tree can be configured using two algorithm types from the '**Properties**' tab. Check out the below given description of the configuration details:

5.3.6.1.1. Classification as Algorithm Type

i) Drag the R-CNR Tree component to the workspace and connect it with a configured data source.
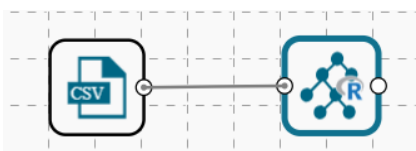
ii) Configure the '**Properties**' tab:

a. Output Information

i. **Algorithm Type:** Select an algorithm type from the drop-down menu.

    1. **Classification**: Select this option if users want to pass dependent column as the categorical values.

    2. **Regression:** Select this option if users want to pass dependent column as numerical values.

ii. **Show Probability**: Select an option from the drop-down menu to create a new column for indicating the chance factor involved in the probability.

    1. **True:** Select this option to display a new column in the output data with probability values.

    2. **False:** Select this option to display any probability value in the output data.

b. Column Selection

i. **Features**: Select input columns from the drop-down list to which the target column needs to compare performing the analysis.

ii. **Target Variable**: Select the target column for which the analysis is performed.

c. New Column Information

i. **Predicted Column Name**: Enter a name for the new column containing the predicted values.

ii. **Probability Column Name**: Enter a name for the new column containing the probability values.

d. Model Tuning

i. **Enable Validation:** Enable validation as a model tuning option by a check mark in the given box.

ii. **XG Boosting:** Enable validation as a model tuning option by a check mark in the given box.

Properties Tab when Model Tunning is not Enabled

Properties Tab when Validation is Enabled as Model Tuning

Properties Tab when XG Boosting is Enabled as Model Tuning

Note: The '**Show Probability**' field appears only if, '**Classification**' option is selected via the '**Algorithm Type**' drop-down menu.

iii)   Click the '**Advanced**' tab and configure if required:

- **Advanced Tab when both the Model Tuning options are Disabled**

a. **Input Data Handling**
   i. **Missing Values:** Select a method to deal with missing values from the drop-down list.
      1. **Rpart:** Select this option to get the estimated missing values for the dependent column based on the independent columns.
      2. **Ignore:** Select this option to skip the records containing missing values in the columns.
      3. **Keep:** Select this option to retain the records containing missing values while performing the calculation.
      4. **Stop:** Select this option to stop the algorithm application if a value is missing in any column.

b. **Tree Pruning**
   i. **Minimum Split:** It indicates a minimum number of observations within a single node for a split to be attempted. The default value for this field is 10.
   ii. **Complexity Parameter:** This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is purned off performing cross-validation, hence the program does not pursue it. The default value for this field is 0.05.
   iii. **Maximum Depth:** It sets the maximum depth of any node of the final tree keeping the depth count for root node 0. It is an optional field (It is recommended to set Maximum Depth value less than 30 rpart for 32 bit-machines.)

c. **Behavior**
   i. **Split Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'**. (This field appears only when the selected algorithm type is **'Classification'**).
   The splitting index can be:
      1. **Gini:** Select this option to measure inequality among values of randomly chosen elements from a set.
      2. **Information:** Select this option to get information about the variables used in the algorithm.
   ii. **Cross-Validation:** It indicates the number of cross-validations that were performed to check the accuracy of the analysis method.
   iii. **Prior Probability:** It is an optional field. This field is dependent on the preceding data values mentioned in the selected dataset. (This field appears when the selected algorithm type is **'Classification'**).

d. **Surrogate Information**
   i. **Use Surrogate:** Select one option from the drop-down menu.
      1. **Display Only:** Select this option to display only the observation, but not split it further.
      2. **Use Surrogate:** Select this option to search surrogate value for the missing values to split the observation. Two fields are displayed:
         a. **Surrogate Style:** Select a style using the drop-down menu.
         b. **Maximum Surrogate:** Set the maximum surrogate value.
      3. **Stop if missing:** Select this option to choose an action based on the nature of majority observations. If values are missed for all the observations, then it will stop splitting further.

- **Advanced Tab when 'Validation' is enabled:**

  a. **Tree Pruning**:
     i. **Complexity Parameter**: This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is purned off performing cross-validation, hence the programme does not pursue it. The default value for this field is 0.05.

iv) Click the '**Validation**' tab and configure the required fields
   a. **Model Selection Method**: Select a method using the drop-down menu. Users need to configure the other fields based on the selected model method.
      i. **Cross-Validation**
         Users need to configure the '**Number of folds**'if the selected model method is '**Cross Validation**'.

ii. **Bootstrap**
Users need to configure the '**Number of resamples**' (Default value for this field is 5), if the selected model method is '**Bootstrap.**'

iii. **Repeated Cross-Validation**
Users need to configure the '**Number of repeats**' and '**Number of folds**' if the selected method is '**Repeated Cross Validation.**'

iv. **Leave One Out Cross Validation**
Users do not get any other field to configure if the selected model method is '**Leave one out cross validation**'.

- **Advanced Tab when 'XG Boosting' is enabled**

  a. **Boosting Parameter**
     i. Number of Iterations: Enter a number suggesting the Number of Iterations
     ii. Number of Classes: Enter a number suggesting the Number of Classes

v) Click **'APPLY'** (After configuring the required Properties, Advanced and/or Validation fields as per your selection of the model)

vi) **Run** the workflow after getting the success message

vii) **Users** will be redirected to the **'CONSOLE'** tab

viii) Follow the below given steps to display the result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the **'RESULT'** tab.

i. Result view when both the Model Tuning options are disabled.

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1	Probability1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	0.6312139
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	0.6312139
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	0.6312139
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	0.6312139
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	0.6312139
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	0.6312139
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	I	0.6312139
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M	0.4319018
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	0.6312139
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	M	0.4319018

Showing 1 to 10 of 4,177 entries   Previous 1 2 3 4 5 ... 418 Next

ii. Result view when **'Validation'** is enabled.

Show 10 entries   Search:

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1	Probability1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	["0.1531792","0.63121387","0.2156069"]
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	["0.1531792","0.63121387","0.2156069"]
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	["0.1531792","0.63121387","0.2156069"]
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	["0.1531792","0.63121387","0.2156069"]
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	["0.1531792","0.63121387","0.2156069"]
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	["0.1531792","0.63121387","0.2156069"]
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	I	["0.1531792","0.63121387","0.2156069"]
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M	["0.3411043","0.22699387","0.4319018"]
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	["0.1531792","0.63121387","0.2156069"]
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	M	["0.3411043","0.22699387","0.4319018"]

Showing 1 to 10 of 4,177 entries   Previous 1 2 3 4 5 ... 418 Next

iii.   Result view when '**XG Boosting**' is enabled.

Show 10 entries   Search:

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	M
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	F
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F

Showing 1 to 10 of 4,177 entries   Previous 1 2 3 4 5 ... 418 Next

Note: The Probability column displays data in the Array format when Validation is enabled.

ix)   Click the '**VISUALIZATION**' tab.
x)   The result data gets displayed via the tree chart.
       a.   Visualization when no Model Tuning option is enabled
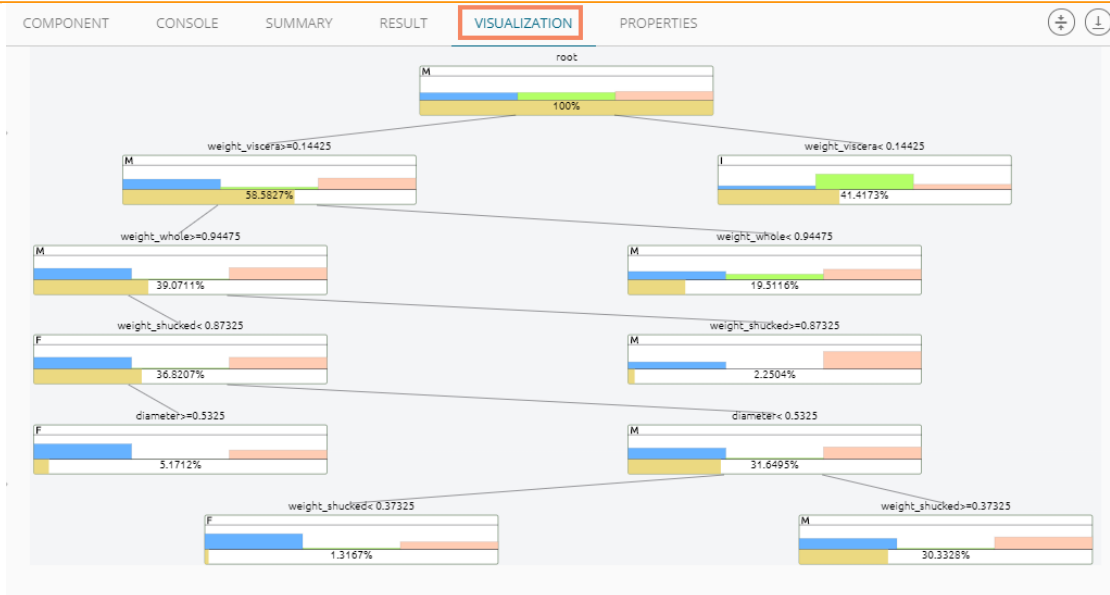
b. Visualization when Validation is enabled
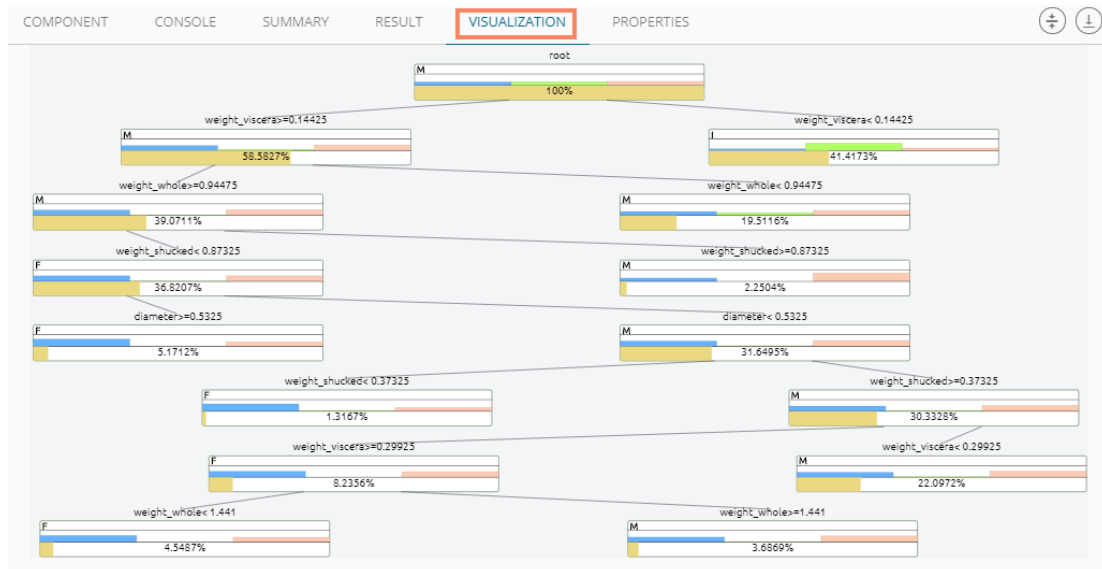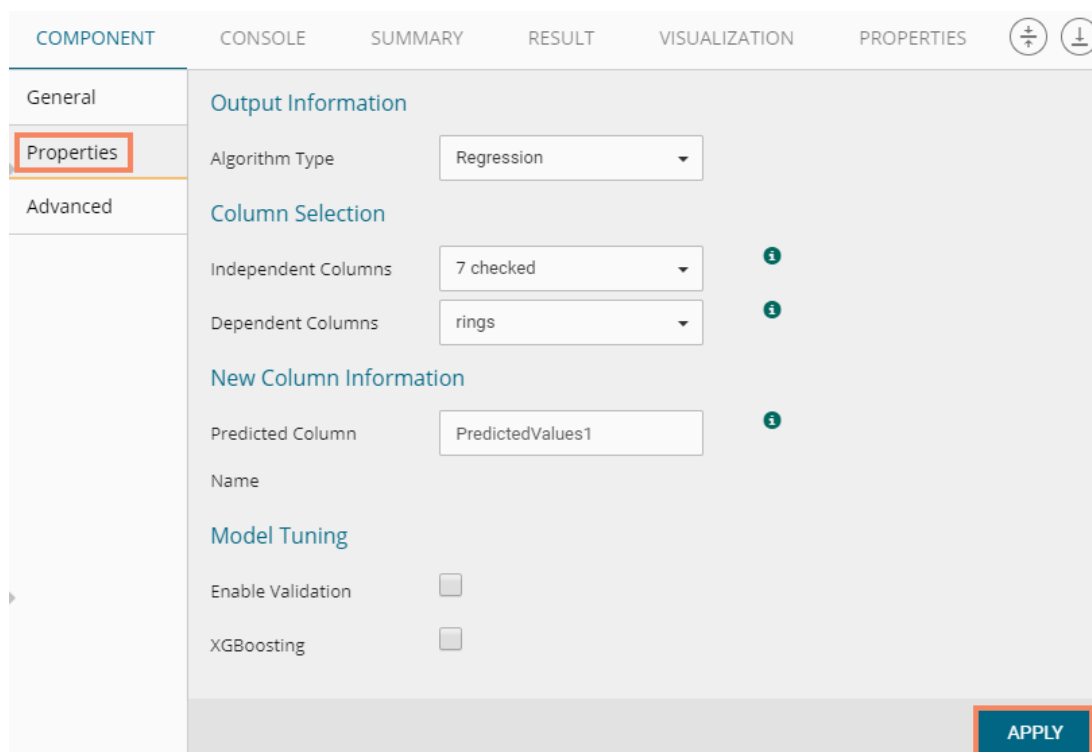
5.3.6.1.2. Regression as Algorithm Type

i) Drag the R-CNR Tree component to the workspace and connect it to a configured data source.
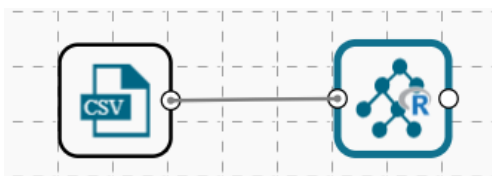
ii) Configure the following fields in the **'Properties'** tab:

    a. **Output Information**

        i. **Algorithm Type:** Select an algorithm type from the drop-down menu.

            1. **Classification**: Select this option if users want to pass dependent column as the categorical values.

2. **Regression:** Select this option if users want to pass dependent column as numerical values.

   b. **Column Selection**
      i. **Features**: Select input columns from the drop-down list to which the target **the** column can be compared to performing the analysis.
      ii. **Target Variable**: Select the target column for which the analysis is performed.

   c. **New Column Information**
      i. **Predicted Column Name**: Enter a name for the new column containing the predicted values.
      ii. **Probability Column Name:** Enter a name for the new column containing the probability values.

   d. **Enable Validation:** Enable validation by a check mark in the given box.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

General

Properties

Advanced

Output Information

Algorithm Type   Regression ▾

Column Selection

Independent Columns   7 checked ▾   ❶

Dependent Columns   rings ▾   ❶

New Column Information

Predicted Column   PredictedValues1   ❶
Name

Model Tuning

Enable Validation ☐

XGBoosting ☐

APPLY

iii)   Click the '**Advanced**' tab and configure if required:

• Advanced Tab when both the Model Tuning options are disabled:

   a. **Input Data Handling**
      i. **Missing Values:** Select a method to deal with missing values from the drop-down list.
         1. **Rpart:** Select this option to estimate the missing values for the dependent column based on the independent columns.
         2. **Ignore:** Select this option to skip the records containing missing values in the columns.
         3. **Keep:** Select this option to retain the records containing missing values while performing the calculation.
         4. **Stop:** Select this option to stop the algorithm application if a value is missing in any column.

   b. **Tree Pruning**
      i. **Minimum Split:** It indicates a minimum number of observations within a single node for a split to be attempted. The default value for this field is 10.
      ii. **Complexity Parameter:** This parameter is primarily used to save computing time

by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is purned off performing cross-validation, hence the program does not pursue it. The default value for this field is 0.05.

    iii. **Maximum Depth:** It sets the maximum depth of any node of the final tree keeping the depth count for root node 0. It is an optional field (It is recommended to set Maximum Depth value less than 30 rpart for 32 bit-machines.)

  c. **Behavior**

    i. **Split Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'** tab. (This field appears only when the selected algorithm type is **'Classification'**).
The splitting index can be:

      1. **Gini:** Select this option to measure inequality among values of randomly chosen elements from a set.

      2. **Information:** Select this option to get information about the variables used in the algorithm.

    ii. **Cross-Validation:** It indicates the number of cross-validations that were performed to check the accuracy of the analysis method.

    iii. **Prior Probability:** It is an optional field. This field is dependent on the preceding data values mentioned in the selected dataset. (This field appears when the selected algorithm type is **'Classification'**).

  d. **Surrogate Information**

    i. **Use Surrogate:** Select one option from the drop-down menu.

      1. **Display Only:** Select this option to only display the observation, but not split it further.

      2. **Use Surrogate:** Select this option to search surrogate value for the missing values to split the observation. Two fields are displayed:

        a. **Surrogate Style:** Select a style using the drop-down menu.

        b. **Maximum Surrogate:** Set the maximum surrogate value.

      3. **Stop if missing:** Select this option to choose an action based on the nature of majority observations. If values are missed for all the observations, then it stops splitting further.

- **Advanced Tab when 'Validation' is enabled:**

a. **Tree Pruning**:
   i. Complexity Parameter: This parameter is primarily used to save the computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is purned off performing cross-validation, hence the programme does not pursue it. The default value for this field is 0.05.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	⬇ Ⓣ
General	**Tree Pruning**					
Properties	Complexity Parameter	.005				
Advanced						
Validation						**APPLY**

iv) Click the '**Validation**' tab and configure the required fields.
   a. **Model Selection Method**: Select a method using the drop-down menu. Users need to configure the other fields based on the model selection method.
      i. **Cross-Validation**
         Users need to configure the '**Number of folds**' if the selected model method is '**Cross Validation**'.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	⬇ Ⓣ
General	**Model Selection**					
Properties	Model Selection Method	Cross validation ▼				
Advanced	Number of folds	3				
Validation						**APPLY**

ii. **Bootstrap**
    Users need to configure the '**Number of resamples**' (Default value for this field is 5) if the selected model method is '**Bootstrap**'.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	⬇ Ⓣ
General	**Model Selection**					
Properties	Model Selection Method	Bootstrap ▼				
Advanced	Number of resamples	5				
Validation						**APPLY**

iii. **Repeated Cross-Validation**
     Users need to configure the '**Number of repeats**' and '**Number of folds**' if the selected method is '**Repeated Cross Validation**'.

iv. Leave One Out Cross Validation

Users do not get any other field to configure if the selected model method is '**Leave one out cross validation**'.

• Advanced Tab when XG Boosting is Enabled

b.   Boosting Parameter
   i.   Number of Iterations: Enter a number suggesting the Number of Iterations
   ii. Number of Classes: Enter a number indicating the Number of Classes

v)   Click '**APPLY'.**
vi)   Run the workflow after getting the success message.
vii) Users will be redirected to the '**CONSOLE**' tab.

COMPONENT	**CONSOLE**	SUMMARY

30/7/2018 - 12:59:53 : Process Initiated...

30/7/2018 - 12:59:54 : CSV0 is started.

30/7/2018 - 12:59:55 : CSV0 is completed.

30/7/2018 - 12:59:55 : R-CNR Tree1 is started.

30/7/2018 - 12:59:56 : R-CNR Tree1 is completed.

viii) Follow the below given steps to display the result view:
  a. Click the dragged algorithm component onto the workspace.
  b. Click the '**RESULT**' tab.
      i. Result View when both the Model Tuning options are disabled.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1	Probability1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	0.6312139
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	0.6312139
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	0.6312139
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	0.6312139
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	0.6312139
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	0.6312139
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	I	0.6312139
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M	0.4319018
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	0.6312139
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	M	0.4319018

Showing 1 to 10 of 1,000 entries   Previous 1 2 3 4 5 ... 100 Next

ii. Result view when '**Validation**' is enabled.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1	Probability1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	["0.1531792","0.63121387","0.2156069"]
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	["0.1531792","0.63121387","0.2156069"]
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	["0.1531792","0.63121387","0.2156069"]
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	["0.1531792","0.63121387","0.2156069"]
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	["0.1531792","0.63121387","0.2156069"]
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	["0.1531792","0.63121387","0.2156069"]
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	I	["0.1531792","0.63121387","0.2156069"]
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M	["0.3411043","0.22699387","0.4319018"]
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	["0.1531792","0.63121387","0.2156069"]
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	M	["0.3411043","0.22699387","0.4319018"]

Showing 1 to 10 of 1,000 entries   Previous 1 2 3 4 5 ... 100 Next

iii. Result view when '**XG Boosting**' is enabled.

Show 10 ▼ entries   Search:

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	M
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	F
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F

Showing 1 to 10 of 4,177 entries   Previous   1   2   3   4   5 … 418   Next

Note: The Probability column is displayed in the Array format while enabling the 'Validation' option.

ix)   Click the '**VISUALIZATION**' tab.
x)   The result data will be displayed via the tree chart.
   a. Visualization when no Model Tuning option is Enabled
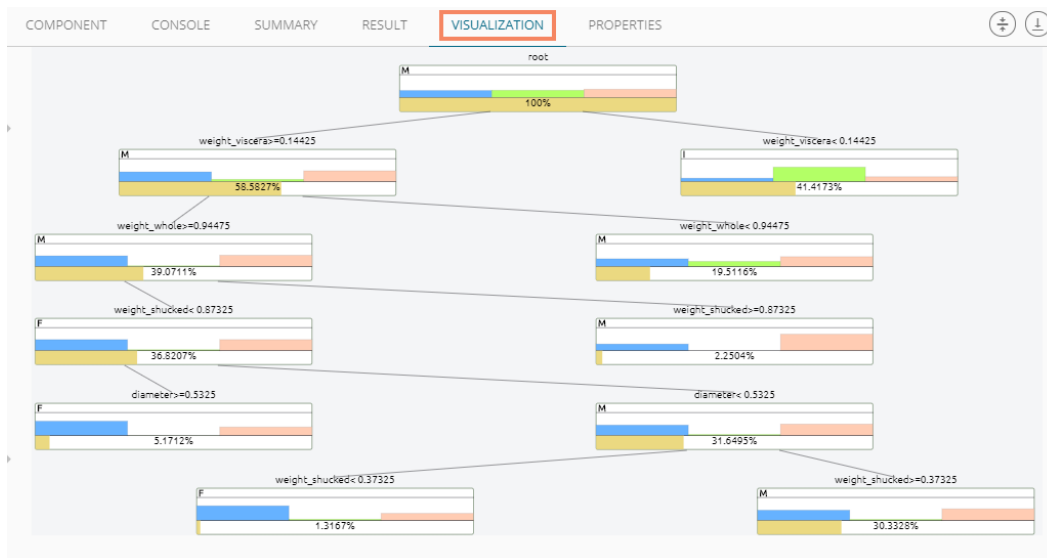
   b. Visualization when Validation is enabled
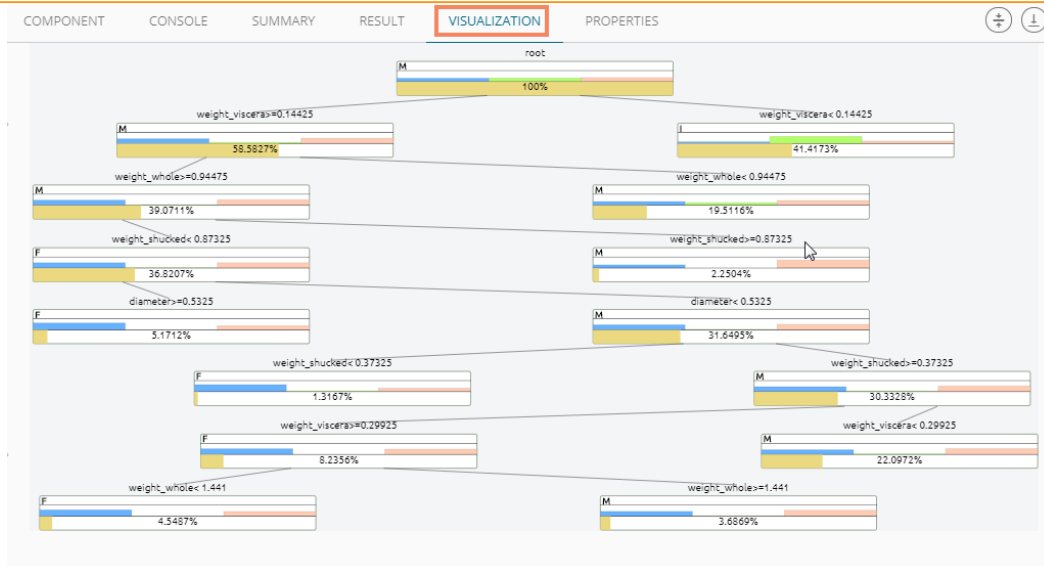
5.3.6.2. R-Naive Bayes
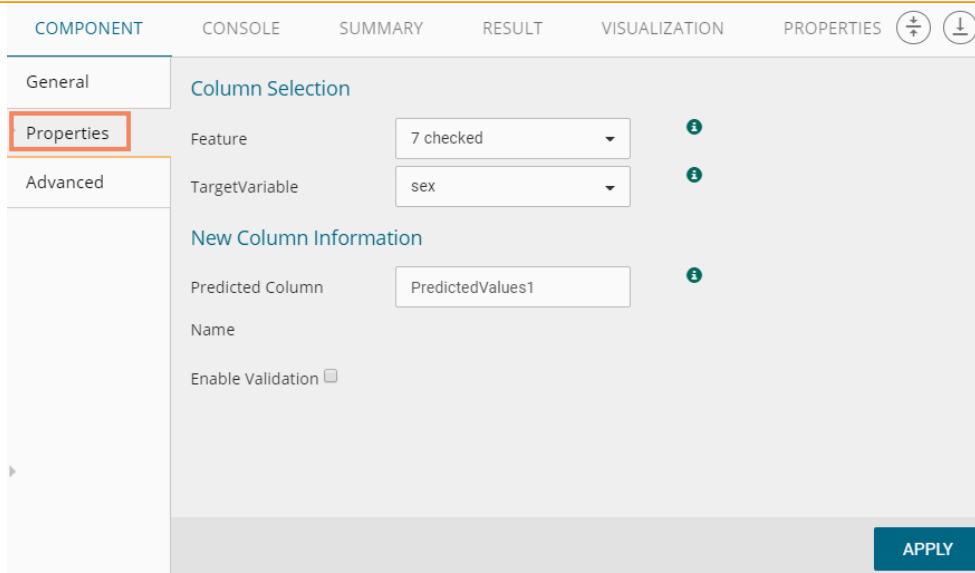
Naive Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature. For example, a fruit may be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

R Naive Bayes is as a leaf node under Classification algorithms under the Algorithm tree node. The component consists of one node for reading data from a data source and another one for giving the result.
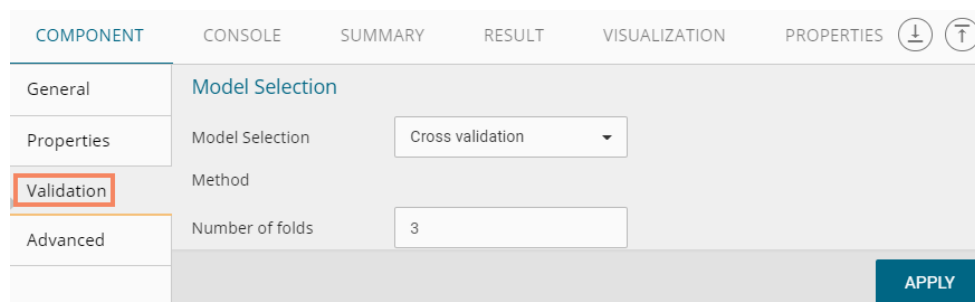
i) Drag the R-Naive Bayes component to the workspace and connect it with a configured data source.
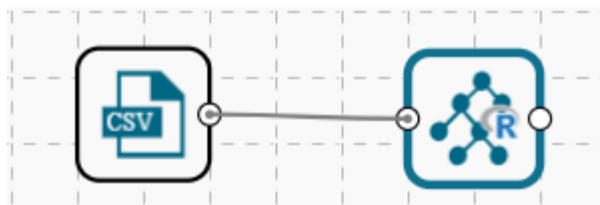
ii) Configure the following fields in the '**Properties**' tab:
   a. **Column Selection**
      i. **Feature**: Select input columns from the drop-down menu to which the target variable can be compared performing the analysis.
      ii. **Target Variable**: Select the target column for which the analysis is Performed.
   b. **New Column Information**
      i. **Predicted Column Name**: Enter a name for the new column containing the predicted values.
   c. **Enable Validation:** Enable validation by a checkmark in the given box.

iii) Click the '**Validation**' tab and configure it, if it has been enabled from the Properties tab
   a. Model Selection
      i. Model Selection Method: Select a modeling method using the drop-down menu.
         1. Cross-Validation
         2. BootStrap
         3. Repeated Cross-Validation
         4. Leave One Out Cross Validation
      ii. Number of folds: Enter a numerical value for the number of folds.

iv) Click the '**Advanced**' tab and configure if required.

- **Advanced Tab when 'Validation' is Disabled:**

a. **Input Data Handling**
   i. **Missing Values:** Select a method to deal with missing values from the drop-down menu.
      1. **Ignore**: Selecting this option will skip the records containing missing values in the columns.
      2. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.
   ii. **Laplace Smoothing:** Enter the smoothing constant for smoothing observations. Smoothing constant must be a double value greater than 0. Entering 0 will disable Laplace smoothing.
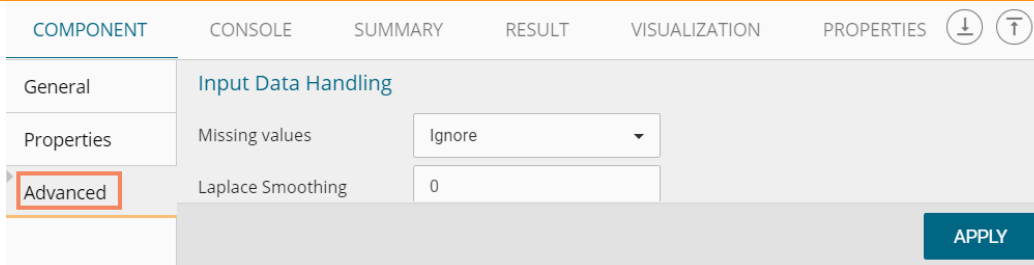
Advanced Tab when 'Validation' is Enabled:

a. Input Data Handling

 i.   **Laplace Smoothing:** Enter the smoothing constant for smoothing observations. Smoothing constant must be a double value greater than 0. Entering 0 disables Laplace smoothing.

 ii. **Kernel:** Select an option using the drop-down menu.
  1. **True**
  2. **False**

 iii. **Band Width:** Enter a bandwidth value (Default value for this field is 0.1).
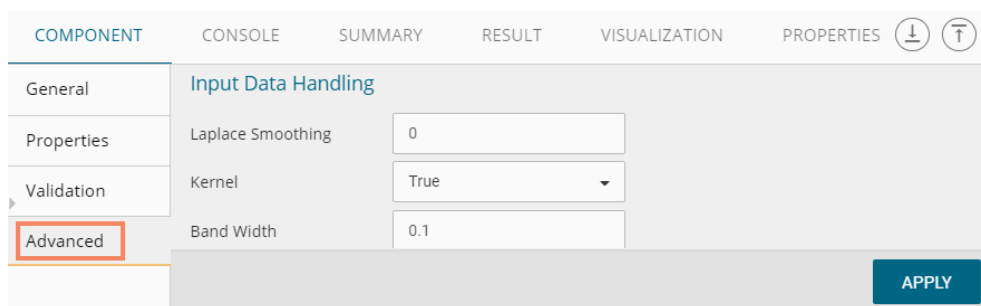
v)   Click the **'Apply'** option
vi) Run the workflow and after getting the success message
vii) Users will be redirected to the **'CONSOLE'** tab.
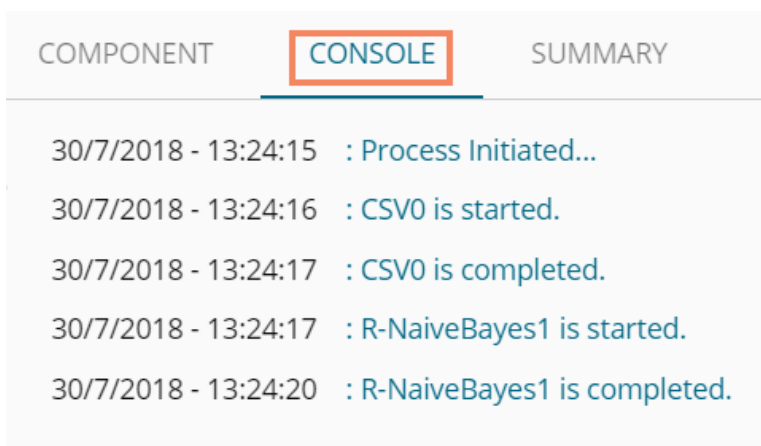
viii) Follow the below given steps to display the result view:
 a. Click the dragged algorithm component onto the workspace.
 b. Click the **'Result'** tab.
  i. **Result View when Validation was disabled**

| | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | |

Show 10 ▾ entries   Search: [   ]

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	M
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F

Showing 1 to 10 of 1,000 entries   Previous 1 2 3 4 5 … 100 Next

ii. Result View when Validation was Enabled

| | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | |

Show 10 ▾ entries   Search: [   ]

sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues1
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	F
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	F
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F

Showing 1 to 10 of 1,000 entries   Previous 1 2 3 4 5 … 100 Next

ix)   Click the 'SUMMARY' tab to see the detailed Model Summary

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | |

```
------------ Summary of the model ------------


1.Independent Columns


length  (double)
        diameter        (double)
        height  (double)
        weight_whole    (double)
        weight_shucked  (double)
        weight_viscera  (double)
        weight_shell    (double)




2.Dependent Column used in the algorithm :


sex     (string)
```
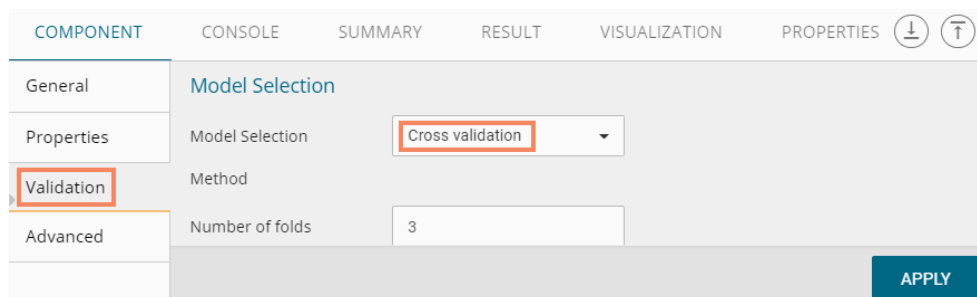
Note:

a. The '**VISUALIZATION**' tab does not display any graphical representation for the R Naive Bayes results in data.

b. The '**Validation**' tab provides multiple options under the '**Model Selection Method**' drop-down menu.

   All the available Model Selection Methods are described below:

   i. **Cross-Validation**
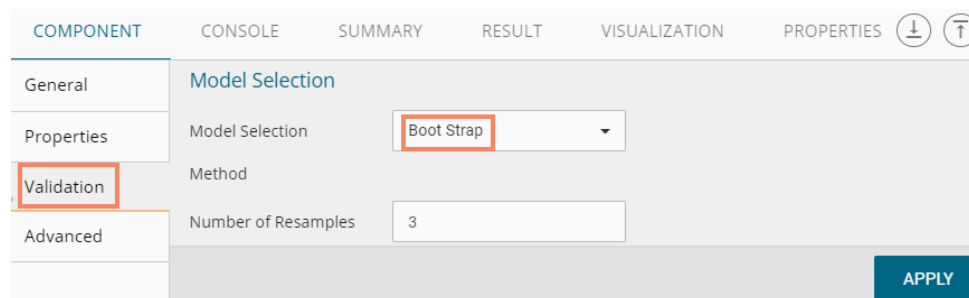
   Users need to configure the '**Number of folds**'if '**Cross Validation**' is the model selection
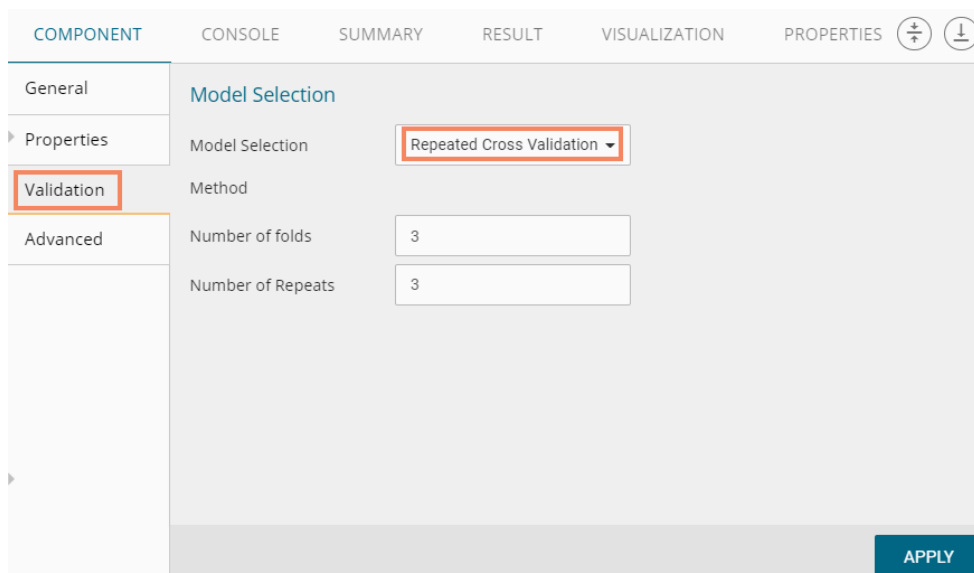
   ii. **Bootstrap**

   Users need to configure the '**Number of resamples**' if '**Bootstrap**' is the model selection method

   iii. **Repeated Cross-Validation**

   Users need to configure the '**Number of repeats**' and '**Number of folds**'if the selected method is '**Repeated Cross Validation**.'

    iv.   **Leave One Out Cross Validation**

Users do not get any other field to configure if the selected model method is **'Leave one out cross validation'**

5.3.7. Correlation

The Correlation algorithm provides a method for clustering a set of objects into the optimal number of clusters without specifying the number in advance.

5.3.7.1. R- Correlation

i)   Drag the R-Correlation component to the workspace and connect to a configured data source.
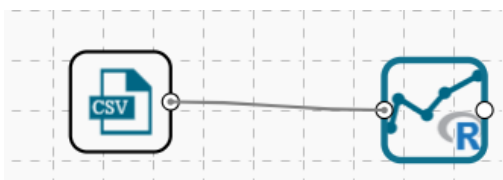
ii)   Configure the following fields in the **'Properties'** tab:

    a. **Input Columns:** Select any two columns using the drop-down menu

    b. **Method:** Select a method using the drop-down menu. The available methods are:

        i.   Pearson

        ii. Kendall

        iii. Spearman

    c. **Missing Value Method:** Select the required option using the drop-down menu.The available methods to apply the Missing Value are:

        i.   Everything

        ii. All.obs

        iii. Complete.obs

        iv. Na.or. complete

        v.   Pairwise.complete.obs

iii)   Click '**APPLY**'

iv) Run the workflow
v) Users will be redirected to the '**CONSOLE**' tab

vi) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.
vii) Columns displaying '**Eruption**' and '**Waiting**' probable values gets added to the result data.
   Note: The selected dataset has more columns then displayed in the below given result view.

category	age	test_time	motor_UPDRS	total_UPDRS	Jitter_Absolute	Jitter_RAP	Jitter_PPQ5	Jitter_DDP
age	1	0.0198838435361529	0.273664760443451	0.310289928642946	0.0356913404516575	0.0102549882693341	0.0131993668204403	0.0102578355
test_time	0.0198838435361529	1	0.06791826408574	0.0752626604217251	-0.0113648116570903	-0.0288878317410302	-0.0232899082521126	-0.028875982
motor_UPDRS	0.273664760443451	0.06791826408574	1	0.947231314131496	0.050903280466618	0.0726835303937712	0.0762908727395432	0.0726979194
total_UPDRS	0.310289928642946	0.0752626604217251	0.947231314131496	1	0.0669267342935041	0.064015417055308	0.0633517753115959	0.0640274572
Jitter_Absolute	0.0356913404516575	-0.0113648116570903	0.050903280466618	0.0669267342935041	1	0.844626279907459	0.790537650669139	0.8446303547
Jitter_RAP	0.0102549882693341	-0.0288878317410302	0.0726835303937712	0.064015417055308	0.844626279907459	1	0.947195933695748	0.9999996211
Jitter_PPQ5	0.0131993668204403	-0.0232899082521126	0.0762908727395432	0.0633517753115959	0.790537650669139	0.947195933695748	1	0.9472025633
Jitter_DDP	0.0102578355360288	-0.028875982725496	0.0726979194936288	0.0640274572105285	0.844630354740171	0.999999621128701	0.947202563388296	1
Shimmer	0.101553855701336	-0.0338701798079251	0.102348700363377	0.0921409137348206	0.649046375246799	0.68172901329222	0.7327474778762011	0.6817337641
Shimmer_dB	0.111129663999778	-0.0309624120719725	0.110075997050723	0.0987897305289653	0.65587068086138	0.685550536141321	0.734590791517138	0.6855561312

Showing 1 to 10 of 20 entries   Previous 1 2 Next

viii) Click the '**VISUALIZATION**' tab.
ix) The probable values of the selected columns will be displayed via the Correlation Plot.

Correlation Plot
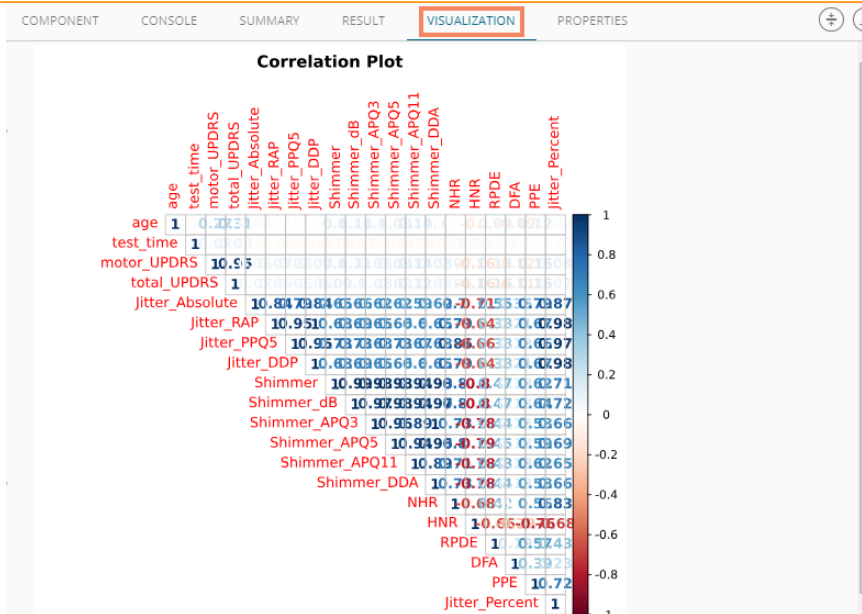
x) Click the '**SUMMARY**' tab to view the model summary
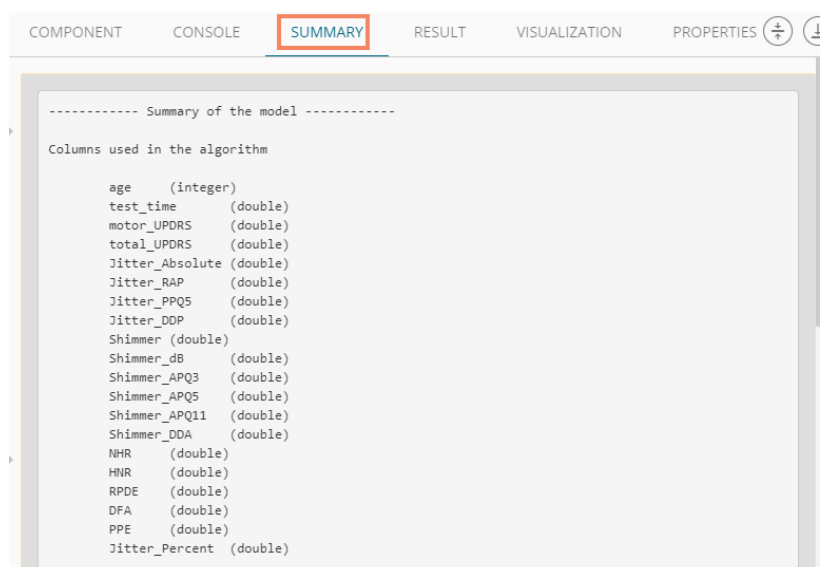
5.4. Apply Model

5.4.1. R Apply Model

This component is provided to generate predictions based on R trained classification model. Users can view predicted column value and probability of each label class by using the classification model.

Users can create a model via the following ways:
- Generate a model using an algorithm
- Generate a model using the saved models

The R Apply Model consists of 2 input nodes and 1 output node.
- **Input Nodes**
  - o Upper node – Model/Training data
  - o Lower node – Testing data
- **Output Node**

o Node – Result data

i) Click the '**Apply Model**' tree-node to access the '**R Apply Model**' leaf-node will be displayed
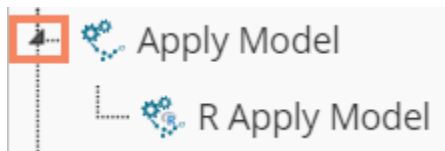
ii) Drag the R Apply Model component onto the workspace and connect it with a valid combination of Data source and algorithm (Configure the data source and algorithm components. In this case, the used algorithm is R CNR Tree.)

iii) Click '**R Apply Model**' component.
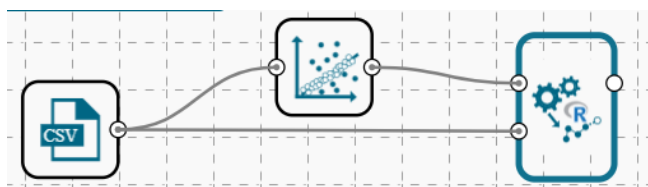
iv) Basic component details will be displayed

a. Component Name: It displays the predefined name of the component

b. Alias Name: It displays a predefined name that suggests even the component's position in the workflow

v) Click '**APPLY**'
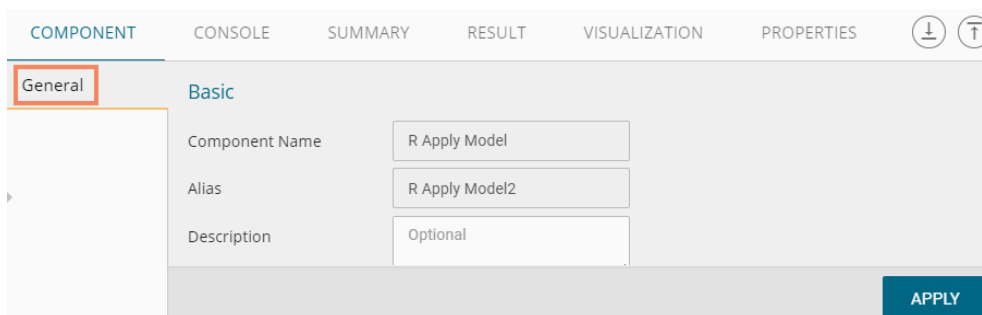
Note: Number given to the Apply Model signifies its place in the workflow, E.g., R Apply Model2 in the below given image suggests that it is in the third position in the workflow.

vi) Run the workflow

vii) Users will be redirected to the '**CONSOLE**' tab.

COMPONENT	CONSOLE	SUMMARY

13/4/2018 - 18:40:29   : Process Initiated...

13/4/2018 - 18:40:32   : CSV0 is started.

13/4/2018 - 18:40:32   : CSV0 is completed.

13/4/2018 - 18:40:32   : Interquartile range1 is started.

13/4/2018 - 18:40:32   : Interquartile range1 is completed.

13/4/2018 - 18:40:32   : R Apply Model2 is started.

13/4/2018 - 18:40:33   : R Apply Model2 is completed.

viii)   Follow the below given steps to display the result view:
   a.   Click the dragged R Apply Model component on the workspace.
   b.   Click the '**RESULT**' tab.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

Month	Day_of_month	Day_of_week	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	I
1	1	4	3.01	5480	8	20			50
1	2	5	3.2	5660	6		38		
1	3	6	2.7	5710	4	28	40		26
1	4	7	5.18	5700	3	37	45		59
1	5	1	5.34	5760	3	51	54	45.32	14
1	6	2	5.77	5720	4	69	35	49.64	15
1	7	3	3.69	5790	6	19	45	46.4	26
1	8	4	3.89	5790	3	25	55	52.7	55
1	9	5	5.76	5700	3	73	41	48.02	20
1	10	6	6.94	5700	3	59	44		26

Showing 1 to 10 of 358 entries   Previous 1 2 3 4 5 ... 36 Next

ix)   Click the '**SUMMARY**' tab to view the model summary.

```
************* Summary of All Stages *************
~~~~~~~~~~~~~ Summary of stage 1 ~~~~~~~~~~~~~
------------ Summary of the model ------------
Columns used in the algorithm
              ozone_reading   (double)

Inter Quartile Outlier Detection Summary
  ---------------------------------------------------------------
  Quartile Information:
          First Quartile found at row  90  with a value  4.94
          Third Quartile found at row  271  with a value  16.22
  For a fence coefficent of  1.5
          Lower fence value :  -11.98
          Upper fence value :  33.14
  Total Number of Outliers detected :  8

Data set Summary
  ---------------------------------------------------------------
  1.Median :  9.35
  2.Standerd Deviation :  7.91386523072765
  3.Number of values considered :  366

------------ End of Summary ----------
~~~~~~~~~~~~ End of stage 1 summary ~~~~~~~~~~~~
*************** End of Summary ****************
```

Note:
   a. The result dataset of the model can be written to a database using a Data Writer.
   b. Column header and data type of feature column for both the saved model and testing data should match. If column headers and data types do not match, an alert message will be displayed.
   c. It is not mandatory for the testing data set to contain a label column.

5.5. Performance

Users can evaluate model performance through a list of parameters using the performance component. Users can use the R Performance components only for the classification algorithms.

5.5.1. R Performance

The R Performance component is provided as a leaf-node under the Performance tree-node. It contains 3 input nodes that can be used to compare up to 3 models. Each node has a static name like model_0, model_1, and model_2. Based on the connection to the node model summary can be viewed with respective names.

R Performance components can be of the following formats:

1. Binary Classification: Used when the label has two classes
2. Multi Classification: Used when the label has 3 or more beta values

In the case of multiple models, all the model statistics will come in the summary of performance (up to 3 models can be compared).

Steps to Connect an R Performance component (to a model)

i)   Drag the R Performance component to the workspace and connect to a valid workflow (In this example, a workflow created with the R Naïve Bayes algorithm has been used)
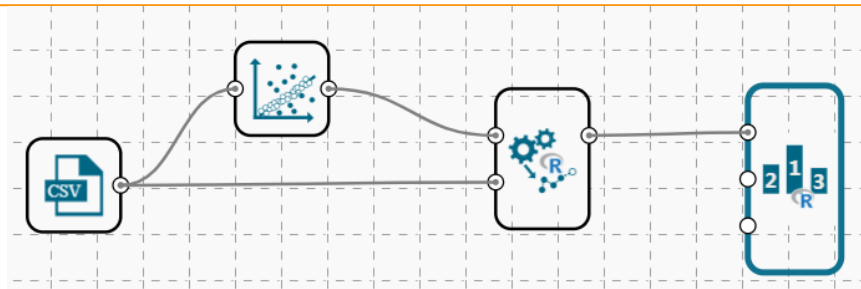
ii) Configure the '**Properties**' tab
   a. **Performance Type**: Select an option using the drop-down menu.
      i. Binary Classification: To be used when the label has two classes.
      ii. Multiclass Classification (Default option): To be used when the label has 3 or more beta values.
iii) Click '**APPLY**'

Users will get different outcomes based on the selected Performance types as described below:

- **Multi Classification Metrics**
   1. Navigate to the '**Properties**' tab of the R Performance component.
   2. Select '**Multi-Classification Metrics**' Performance type via the drop-down menu
   3. Click '**APPLY**'

   4. Run the workflow
   5. Users will be redirected to the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY

13/4/2018 - 19:6:4 : Process Initiated...

13/4/2018 - 19:6:5 : CSV0 is started.

13/4/2018 - 19:6:6 : CSV0 is completed.

13/4/2018 - 19:6:6 : R-NaiveBayes1 is started.

13/4/2018 - 19:6:31 : R-NaiveBayes1 is completed.

13/4/2018 - 19:6:31 : R Apply Model2 is started.

13/4/2018 - 19:6:42 : R Apply Model2 is completed.

13/4/2018 - 19:6:42 : R-Performance3 is started.

13/4/2018 - 19:6:43 : R-Performance3 is completed.

6. Users can view the summary by clicking the '**SUMMARY**' tab (First click the performance component and then click on the '**SUMMARY**' tab).

The following details will be displayed by clicking on the '**SUMMARY**' tab:

a. Confusion Metrix and Statistics
   i. Displays Confusion Matrix of each model
   ii. The column consists of Actual labels and row consist of Predicted labels

b. Overall Statistics
   i. Overall statistics of each model can be viewed in a tabular format
   ii. Each model will be rows and following statistics columns
      1. Accuracy
      2. 95% CI
      3. No Information Rate
      4. P – value
      5. Kappa
      6. Mcnemar's Test P-Value

c. Statistics by Class
   i. Label-wise the following statistics can be shown:
      1. Sensitivity
      2. Specificity
      3. Pos Pred Value
      4. Neg Pred Value
      5. Prevalence
      6. Detection Rate
      7. Detection Prevalence
      8. Balanced Accuracy

```
------------ Summary of Model Comparision ------------

------------  Performance of first model  ------------

Confusion Matrix and Statistics


        I    F    M
 I  1097  271  401
 F   106  789  553
 M   139  247  574


Overall Statistics

               Accuracy : 0.5889
                 95% CI : (0.5738, 0.6039)
    No Information Rate : 0.3658
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3877
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: I Class: F Class: M
Sensitivity            0.8174   0.6037   0.3757
Specificity            0.7630   0.7704   0.8543
Pos Pred Value         0.6201   0.5449   0.5979
Neg Pred Value         0.8983   0.8102   0.7035
Prevalence             0.3213   0.3129   0.3658
Detection Rate         0.2626   0.1889   0.1374
Detection Prevalence   0.4235   0.3467   0.2298
Balanced Accuracy      0.7902   0.6870   0.6150


------------ End ----------
```

- **Binary Classification Metrics**
  1. Navigate to the '**Properties**' tab of the R Performance component
  2. Select '**Binary Classification Metrics**' Performance type via the drop-down menu

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	⊥	T̄
General		R-Performance					
Properties		Performance Type	Binary Classification Metrics ▾				
▸							APPLY

  3. Click '**APPLY**'
  4. Run the workflow
  5. Users will be redirected to the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY |

13/4/2018 - 19:6:4   : Process Initiated...

13/4/2018 - 19:6:5   : CSV0 is started.

13/4/2018 - 19:6:6   : CSV0 is completed.

13/4/2018 - 19:6:6   : R-NaiveBayes1 is started.

13/4/2018 - 19:6:31   : R-NaiveBayes1 is completed.

13/4/2018 - 19:6:31   : R Apply Model2 is started.

13/4/2018 - 19:6:42   : R Apply Model2 is completed.

13/4/2018 - 19:6:42   : R-Performance3 is started.

13/4/2018 - 19:6:43   : R-Performance3 is completed.

6.   Click the '**VISUALIZATION**' tab to see the graphical representation of the result data.
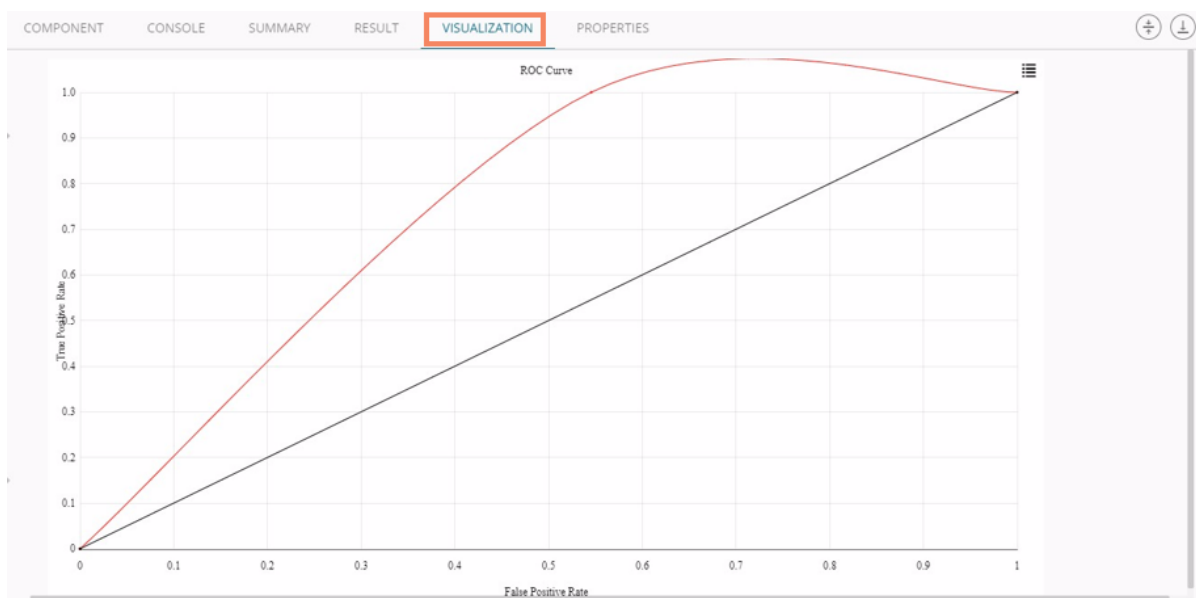
Note:
a. In the case of multiple models, all the model statistics will be displayed in the summary tab of the performance component (up to 3 models can be compared).
b. No data will be displayed under the '**RESULT**' tab for R-Performance (Binary Classification).

5.6. Data Writer(s)

Data Writers are provided to store the results of the predictive analysis in flat files or databases for further in-depth analysis.

5.6.1. Data Store Writer

Elastic Search Writer component is listed under the Data Writer Tree node. The Data Store Writer allows users to write the processed data onto the Elastic Search server which makes it more distributed.

i)   Drag the Data Store Writer component to the workspace and connect it with a configured data source or any valid combination of a data source with other given components

ii) Click on the connected Data Store Writer component
iii) The component tab for the data writer will open
iv) Configure the required component properties
    i. Select Data Store: Select a data store from the drop-down menu
    ii. Select Operation Type: Select an option from the drop-down menu
    iii. Users will get all the Dimensions, Measures, and Time fields from the selected data source
    iv. They can define hierarchy by dragging the required Dimensions into the Drill Definition box
v) Click '**NEXT**'

vi) Users will be redirected to the Advanced fields to configure the Batch Query Properties
vii) Select a dimension for the batch query
viii) Click '**APPLY**'

Batch Query Properties

Select Dimension for Batch Query

- Credit_History
- Purpose_of_the_loan
- Savings_account_or_bonds
- Property_Type
- Other_instalment_plans
- Housing_Type
- Job
- Telephone
- Foreign_Worker

Selected Dimension for Batch Query

- Present_employment_since

APPLY

ix) After getting the success message run the workflow

x) Users will get the process status under the '**CONSOLE**' tab

COMPONENT   CONSOLE   SUMMARY   RESULT

19/7/2018 - 17:34:23   : Process Initiated...

19/7/2018 - 17:34:24   : CSV0 is started.

19/7/2018 - 17:34:24   : CSV0 is completed.

19/7/2018 - 17:34:25   : R-K-Means1 is started.

19/7/2018 - 17:34:25   : R-K-Means1 is completed.

19/7/2018 - 17:34:25   : Data Store Writer2 is started.

19/7/2018 - 17:34:37   : Data Store Writer2 is completed.

xi) The data will be saved in the desired format to the selected Data Store Writer after the console process gets completed.

Note:

a. Users also get '**General**' fields for the Data Store Writer component, but they need not configure it.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

General   Basic

Properties   Component Name   Data Store Writer

Advanced   Alias   Data Store Writer2

             Description   Optional

NEXT

b. Users can also create a new data store using the 'Create New Data Store' option from the 'Select Data Store' drop-down menu. Users can give a name to the newly created data store by using the 'Data Store Name' field.

c. Users can move only one-dimension at a time from the list of '**Select Dimension for Batch Query**' value for the batch query.

5.6.2. File Writer

Users can write output data to flat files like CSV, TEXT, and DAT files using the File Writer.

5.6.2.1. CSV Writer

i)   Click '**TreeNode**' provided next to the '**Data Writer**' option.
ii) Select '**File Writer**' option.
iii) Select and drag '**CSV Writer**' component to the workspace.

iv) Connect the '**CSV Writer**' to a configured data source or a valid workflow
v)   Click on CSV Writer component to access component properties.
vi) Enter '**File Name**' in the displayed field**.**
vii) Click '**APPLY**'

viii)   After getting the success message run the workflow

ix)   Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY |

13/4/2018 - 19:0:16   : Process Initiated...

13/4/2018 - 19:0:17   : CSV0 is started.

13/4/2018 - 19:0:18   : CSV0 is completed.

13/4/2018 - 19:0:18   : R-NaiveBayes1 is started.

13/4/2018 - 19:0:43   : R-NaiveBayes1 is completed.

13/4/2018 - 19:0:43   : CSV File Writer2 is started.

13/4/2018 - 19:0:43   : CSV File Writer2 is completed.

x)   The data will be written in the CSV File

xi)   Click the '**CSV Writer**' component

xii)   A pop-up message will appear with a link to download the CSV file

Download File   ×

Click Here to download Sample CSV File.csv
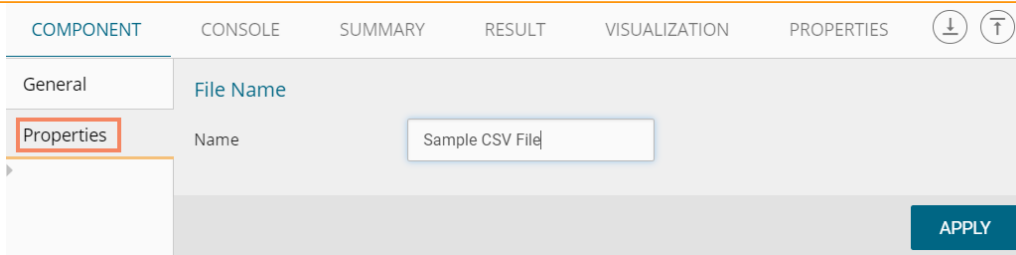
xiii)   Click the link to download the CSV file.

5.6.2.2.   JSON Writer

i)   Click on '**TreeNode**' provided next to the '**Data Writer**' option.

ii)   Select '**File Writer**' option.

iii)   Select and drag '**JsonWriter**'component to the workspace.

iv)   Connect the **'JsonWriter'** to a configured data source.
v)   Click on **'JsonWriter'** component to access component properties.
vi)   Enter **'File Name'** in the displayed field.
vii) Click **'APPLY'**

viii) After getting the success message run the workflow
ix)   Users will get the process status under the **'CONSOLE'** tab

x)   A Pop-up message will appear with a link to download the **JSON** file.
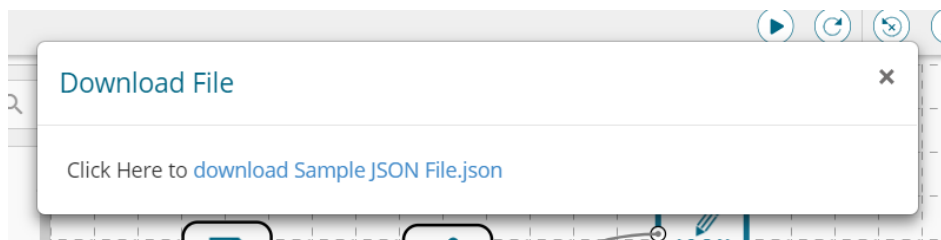
xi)   Click the link to download the JSON file.

5.6.3. Database Writer

5.6.3.1. Internal Data Writer

This data writer will store the data in databases like MySQL, MSSQL, and Oracle.

i) Click **'TreeNode'** provided next to the **'Data Writer'** option.
ii) Select **'Database Writer'** option.
iii) Select and drag **'Internal Data Writer'** component to the workspace.

iv) Drag and Connect the '**Internal Data Writer**' component to a configured data source onto the workspace.
v) Click '**Internal Data Writer**' component to access the Component properties

Users will have different '**Properties**' fields based on the selected table operation as described below:

a. **Selecting the 'Create a New Table' as Table Operation**:

   i. **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.

   ii. **Type**: This field will be preselected based on the selected data Connector.

   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch

   iv. **Database Name**: Select a database name from the drop-down menu

   v. **Password:** Enter the database password

   vi. **Table Name**: Select '**Create New Table**' option from the list

   vii. **Table Operation**: Select an option from the drop-down menu
1. Append to Table
2. Overwrite Table
3. Upsert

   viii. **Create New Table:** It is an optional field. It appears when the user selects **'Create New Table'** option from the **'Table Name'** drop-down menu.

   ix. **Auto Increment:** Select an option to enable or disable the auto increment. By enabling this option, a new column will be added to the dataset, and the same column will be selected as the primary key by default.

   x. **Auto Increment Label:** Enter a name for the auto-increment label

   xi. **Column Selected from the model**: Select columns that are needed to be written into the selected database.

vi) Click '**NEXT**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Schema Viewer

Internal Data Writer Properties

Data Source Name	predictive_prod ▾
Type	mysql
Number of Rows in a batch	1000
Database Name	predictive_analysis ▾
Password	•••••••••
Table Name	Create New Table ▾
Table Operation	Upsert ▾
Create New Table	RNaiveBayes
Auto Increment	Enable ▾
Auto Increment Label	AIL
Column selected from model	10 checked ▾

NEXT

vii)   Users will be redirected to the **'Schema Viewer'** option

    a. Select Primary Keys: Select primary key(s) using the drop=down menu

viii)   Click '**APPLY**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Schema Viewer

Internal Data Writer Properties

Select Primary Keys	1 checked ▾

APPLY

xii)   After getting the success message run the workflow

xiii)   Users will get the process status under the '**CONSOLE**' tab
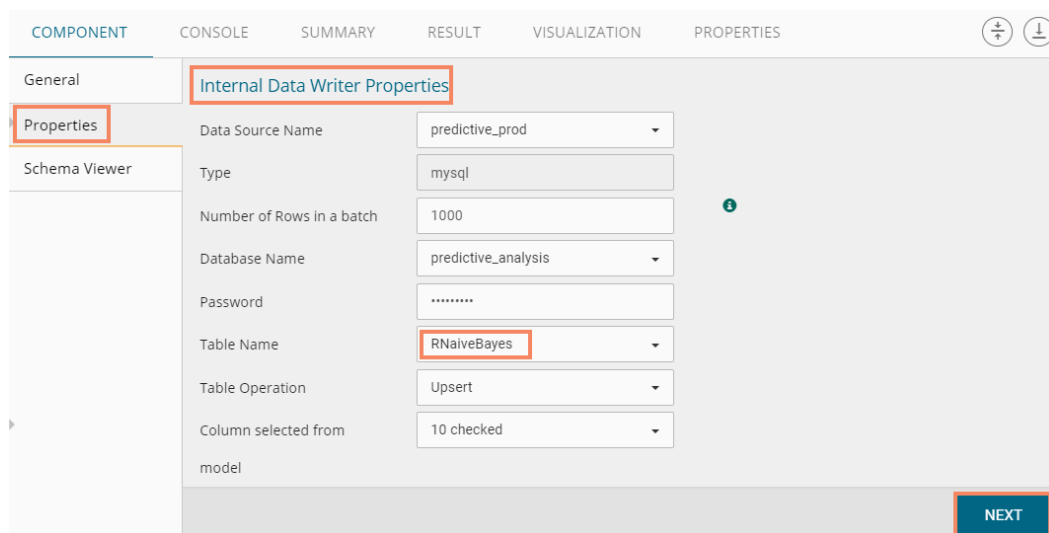
ix) The selected data will be written to the internal data writer successfully

b. Selected Table Operation is an Existing Table:
    i. **Data Connector Name**: Select a data connector from the drop-down menu
    ii. **Type**: Displays a type based on the data connector chosen
    iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
    iv. **Database Name**: Select a database name from the drop-down menu
    v. **Password:** Enter the database password
    vi. **Table Name**: Select an existing table name from the drop-down menu
    vii. **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:
        1. Append Table
        2. Overwrite Table
        3. Upsert Table
    viii. **Column Selected from model**: Select columns that are needed to be written into the selected database.

    ix. **Details of the Selected table:** Displays column headers from the selected table.
x) Click '**NEXT**'

Details of the selected table

AIL
sex
length
diameter
height
weight_whole
weight_shucked
weight_viscera
weight_shell
rings
PredictedValues1

NEXT

xi) Users will be redirected to the '**Schema Viewer**' page.
xii) Click '**APPLY**'
xiii) After getting the success message run the workflow
xiv) Users will get the process status under the '**CONSOLE**' tab
xv) The data will be saved in the selected database at the end of the process

Note:

a. Users will not be able to see the '**Result**' tab for the Internal Data Writer.
b. Auto Increment Column(delta load) supports only for MySQL. Users can configure the Auto-Increment Column only while using the 'Create New Table' option as a Table Name.
c. By selecting an auto-increment column by default, it will be selected as the primary key. If users want to use another column as a primary key other than the Auto Increment Column, then it has to be configured using the '**Schema Viewer**' tab.
d. If users do not mention the primary key for the 'Upsert' table operation, it will act as '**Append.**'

5.6.3.2. Cassandra Writer

Cassandra Writer can be used to store the predictive executions.
a. **Selecting 'Create a New Table' as Table Operation**

i) Click '**TreeNode**' provided next to the '**Data Writer**' option
ii) Select '**Database Writer**'
iii) Select and drag '**Cassandra Writer**' component to the workspace

iv) Connect the '**Cassandra Writer**' to a configured data source
v) Click the '**Cassandra Writer**' component to access it

vi) Configure the following **Properties** details:
   a. **Select Data Connector**: Select a data connector using the drop-down menu
   b. **Host Name**: Based on the chosen data connector a hostname will be displayed (Users cannot edit this field)
   c. **Port Name**: The server port number will be displayed (Users cannot edit this field)
   d. **Username**: Username of the selected connection appears by default. (Users cannot edit this field)
   e. **Password: the database password**
   f. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
   g. **Select Key Space**: Select a keyspace using the drop-down menu
   h. **Replication Factor**: The replication factor mentioned in the selected '**Key Space**' will be displayed (Users cannot edit this field)
   i. **Select Table**: Select 'Create a New Table table from the drop-down menu
   j. **Select Columns**: Select the columns that you want to write
   k. **Consistency**: Select an option from the drop-down menu
   l. **New Table**: Provide a name for the newly created table
   m. **New time uuid column name**: Enter a UUID column name
vii) Click '**Next**'

viii) Users will be redirected to the '**Key Specification**' tab.
ix) Configure the following information:
   a. **Headers**: All the columns from the data set will be listed.
   b. **Partition Key (Name)**: The Partition Key determines which node stores the data. It is responsible for data distribution across the nodes.
      • The UUID Column name will be displayed under the '**Partition Key**' window.
      • Users can select and move any column from '**Header**' (Select Column) to '**Partition Key**' space.

- The sequence of the columns listed under Partition Key can be arranged by using '**Up**' or '**Down**' options.
  c. **Clustering Key**: The Clustering Key is a storage engine process that sorts data within the partition. It determines per-partition clustering.
    - The items listed under the Clustering Key box can be arranged by using '**Up**' or '**Down**' options.
    - Users can select any column from '**Headers**'(Select Column) to '**Clustering Key**' space.

  x) Click '**APPLY**'

  xi) After getting the success message run the workflow
  xii) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

13/4/2018 - 19:39:3 : Process Initiated...

13/4/2018 - 19:39:5 : Data Store Reader0 is started.

13/4/2018 - 19:39:7 : Data Store Reader0 is completed.

13/4/2018 - 19:39:7 : R Split Data2 is started.

13/4/2018 - 19:39:7 : R Split Data2 is completed.

13/4/2018 - 19:39:7 : R-CNR Tree2 is started.

13/4/2018 - 19:39:7 : R-CNR Tree2 is completed.

13/4/2018 - 19:39:7 : R Apply Model3 is started.

13/4/2018 - 19:39:7 : R Apply Model3 is completed.

13/4/2018 - 19:39:7 : R-Performance4 is started.

13/4/2018 - 19:39:7 : R-Performance4 is completed.

13/4/2018 - 19:39:7 : cassandra writer5 is started.

13/4/2018 - 19:39:10 : cassandra writer5 is completed.

Note: Users will be provided with some defined consistency level while designing the KeySpace which can be overridden based on the selected replica nodes. Users are provided with the following consistency options:

- One
- Two
- Three
- Quorum

or

b. **Selecting an Existing Table as Table Operation**
   i) Connect the '**Cassandra Writer**' to a configured data source.
   ii) Click the '**Cassandra Writer**' component to access it.
   iii) Configure the following **Properties** details
      i. **Select Data Connector**: Select a data connector from the drop-down menu
      ii. **Host Name**: Enter database server details (from where the user wants to fetch data)
      iii. **Port Name**: The server port number
      iv. **Username**: Username of the selected connection appears by default (Users cannot edit this field)
      v. **Password**: the database password
      vi. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
      vii. **Select Key Space**: Select a keyspace using the drop-down menu
      viii. **Replication Factor**: Replication factor in the selected '**Key Space**' will be displayed (Users cannot edit this field)
      ix. **Select Table**: Select a table from the drop-down menu
      x. **Choose Columns**: Select columns from the drop-down menu that users want to be written in the data writer.
      xi. **Consistency**: Select an option using the drop-down menu
        a. ONE
        b. TWO
        c. THREE

d. QUORUM

xii. **Settings**: Select an option using the drop-down menu. The following choices will be provided:
1. Append Table
2. Overwrite Table

xiii. The list of column headers existing in the table will be displayed once users select a table.

iv) Click '**APPLY**'

v) After getting the success message run the workflow
vi) Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | **CONSOLE** | SUMMARY | RESULT |

13/4/2018 - 19:39:3 : Process Initiated...

13/4/2018 - 19:39:5 : Data Store Reader0 is started.

13/4/2018 - 19:39:7 : Data Store Reader0 is completed.

13/4/2018 - 19:39:7 : R Split Data2 is started.

13/4/2018 - 19:39:7 : R Split Data2 is completed.

13/4/2018 - 19:39:7 : R-CNR Tree2 is started.

13/4/2018 - 19:39:7 : R-CNR Tree2 is completed.

13/4/2018 - 19:39:7 : R Apply Model3 is started.

13/4/2018 - 19:39:7 : R Apply Model3 is completed.

13/4/2018 - 19:39:7 : R-Performance4 is started.

13/4/2018 - 19:39:7 : R-Performance4 is completed.

13/4/2018 - 19:39:7 : cassandra writer5 is started.

13/4/2018 - 19:39:10 : cassandra writer5 is completed.

vii) The data will be saved in the selected Cassandra Writer

5.7. Custom R Script

Users can create and add customized algorithm components by using the '**Custom R-Script**' component. The created scripts will be stored in the '**Saved Scripts'** option.

5.7.1. Creating a New R Script

i) Click '**Custom R Script**' tree-node on the Predictive Analysis home page.
ii) Click '**Create New Script.'**
iii) Users will be directed to the '**Component**' tab.
iv) Configure the following fields in the '**General**' tab:
   a. **Basic**
      i. **Component Name:** Enter a name or title that you wish to give a created R script.
      ii. **Component Type:** Default Component type will be displayed in this field.
      iii. **Description:** Describe the Component (It is an optional field).
v) Click '**NEXT**'

vi) Users will be directed to the '**Script**' tab.
vii) Provide the following information as required:
   a. **Script Editor**
      i. Paste an R-script in the given space on the '**Script Editor**' page.
      ii. Click the '**Validate**' option.
      iii. Use '**Primary Function Details**' to embed the customized R-script into the function.
      iv. Set the function details as shown below:
         1. **Primary Function Name:** Select the name of the created function from the drop-down menu.
         2. **Input Data Frame:** Select a dataset (that has been used above) from a drop-down menu.
         3. **Output Data Frame:** Enter a choice to which the data will be passed.
         4. **Model Variable Name:** Enter the output model variable (This field will appear only when the model summary has been enabled).
      v. If you need a visualization chart for the ensuring data, tick the '**Show Visualization**' checkbox.
      vi. If you need to show the summary, tick the '**Show Summary**' checkbox.
viii) Click '**NEXT**'
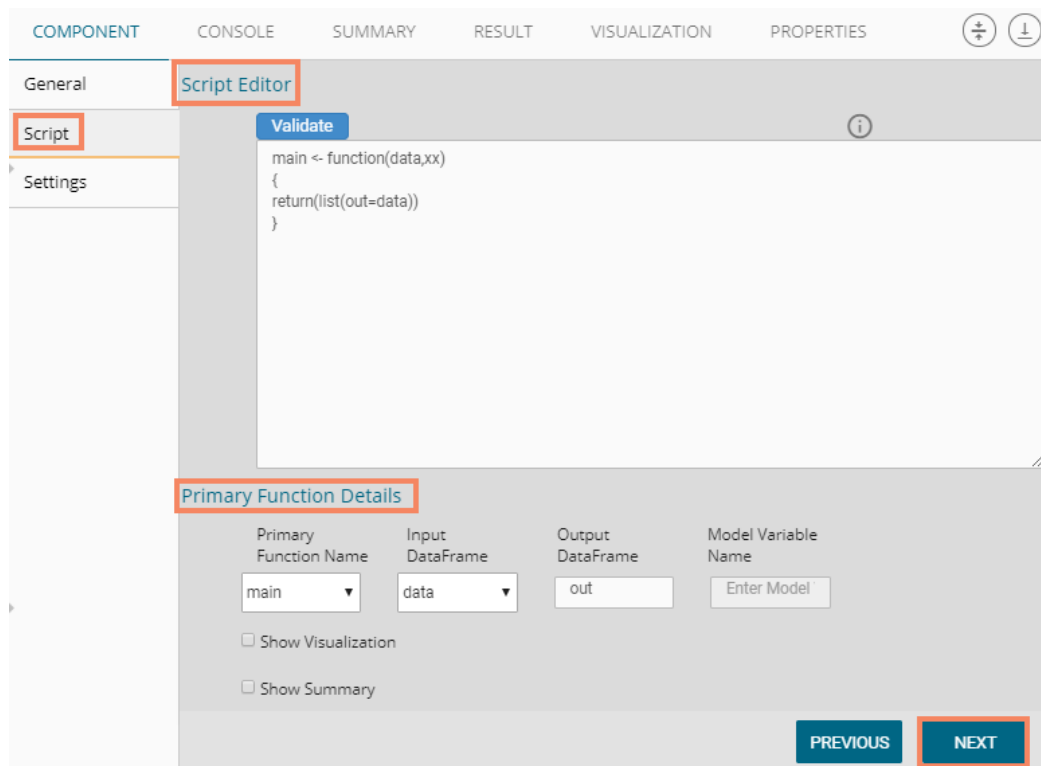
ix) Users will be directed to the '**Settings**' tab.
x) Configure the following fields:
   a. **Output Table Definition**
      This option will configure a number of output columns, column headers, data types.
      i. **Consider all columns from the previous component:** To display all columns of the prior component.
      ii. **Consider None:** To display no column from the previous component.
      iii. **Data Type:** Select a data type for the newly created column using the drop-down list.
      iv. **New Predicted Column Name**: Enter an appropriate name for the new predicted column.

v. ![minus button]: To remove the added row containing **'Data Type'** and **'New Predicted Column Name.'**

vi. ![plus button]: To add a new row containing **'Data Type'** and **'New Predicted Column Name.'**

b. **Property View Definition**

i. **Function Parameters**: Actual names of parameters configured in the script.

ii. **Property Display Name**: Parameter name to be displayed while configuring saved R script as a component.

iii. **Control Type:** User can select out of the following options:
1. Text box,
2. Drop-down menu,
3. Column Selector (single),
4. Column Selector (multiple).

iv. **Settings option** ![gear icon]: To set display for mandatory fields and validate data type for input column. This field is associated with function parameters.

xi) Click **'APPLY'**
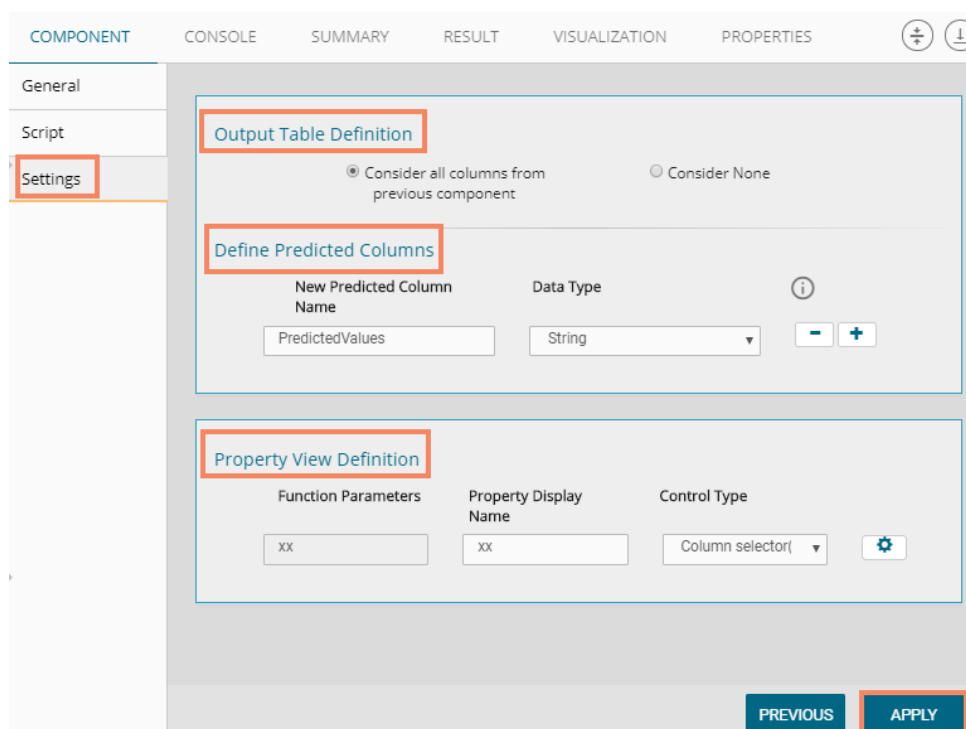
xii) A message will appear to confirm that the newly created R script has been saved.

Rscript saved successfully

xiii) The newly created R Script will be saved in the **'Saved Scripts'** list for the R scripts.
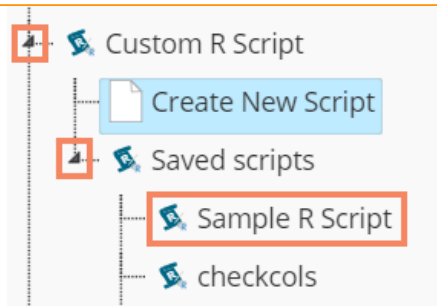
Guidelines for Writing an R- Script

1. R- script needs to be written inside a valid R function. i.e., The entire code body should be inside the curly braces of the function.
2. The R-script should have at least one main function. Multiple functions are acceptable, and one function can call another function, but it should be written above the calling function body. (If called function is an outer function) alternatively, above the calling statement (if called function is an inner function).
3. Any extra packages that are required to run your R script must be installed on the R-server, and it should be loaded using library ('library_name') statement, before calling the associated function in your script.
4. The R-script should return data in the form of a list only, containing the data frame and model (if used).
5. In the return statement, only a data frame can be assigned to the variable 'out.' This data frame supports all structures like list, string, vector, matrix, table.
6. If **'Show Visualization'** field is marked as **'yes'** during the creation of component, then there should be a plot created in the R-script, and if **'Show Summary'** field is marked as **'yes'** then the structures list should have the **'model'** variable.
7. Empty cells, (NULL), (null), NULL, null, /N, NA, N/A are considered as unwanted values and replaced by "NaN" in case of double, long, short, float, byte, integer, and "NA" in case of boolean, string, so instead of using these values in R code use "NaN" or "NA" according to data type of input data.

Note:

   a. Click the '**Information**' button 🛈 to get the list mentioned above of rules for R-script.
   b. '**Model Variable Name**' can be enabled only after selecting '**Show Summary**' option.
   c. Select '**Show Summary**' and '**Show Visualization**' option only if, the R-script carries both the items.
   d. All the supported date data types are listed in date formats in data type definition, all other date formats are considered as string data type.
   e. Mssql data types are considered as string data type.
   f. If the input and output components have a different structure, it will not subset or row bind with "Consider All" option, Users must change to "Consider None" and give different column names for the output to make it run successfully.

5.8. Scheduler
Scheduler helps to schedule the Predictive Workflow as per the requirement.

5.8.1. New Schedule
This section explains the steps to schedule a new job. Scheduling a new job is a continuous step by step process as described below:

   i) Navigate to the Predictive home page.
   ii) Click the '**Scheduler**' tree node.
   iii) Two options will be displayed:
       a. New Scheduler

b. Status

iv) Select '**New Schedule**' from the menu
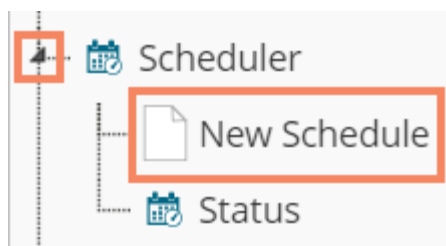
v) Users will be redirected to the '**General**' tab.

5.8.1.1. Configuring General Tab

i) A '**General**' tab will open (by default).
ii) Fill in the required information:
   a. **Model Name**: Select a model name using the drop-down menu.
   b. **Job Name**: Enter a job name.
   c. **Description**: Describe the job (optional field).
   d. **Use Existing Data Connector**: Use radio buttons to select an option.
      i. Select '**Yes**' to use an existing data connector.
      ii. Select '**No**' for not using an existing data connector.
   e. **Use Existing Datawriter**: Use radio buttons to select an option.
      i. Select '**Yes**' to use an existing data writer.
      ii. Select '**No**' for not using an existing data writer.
iii) Click '**NEXT**'

iv) Users will be redirected to the '**Data Source**' tab.

5.8.1.2. Configuring Data Source

Provide the required information to configure a data source:

i) '**General**' fields will be displayed by default.

ii) Users can fill in the required fields:

    a. Component Name: A default name provided for the component.

    b. Alias Name: User can enter a name for the component.

    c. Description: Users can describe the component (optional).

iii) Click '**NEXT**'

iv) Users will be redirected to the '**Properties**' fields.

v) Configure the following fields (to configure a new data source):

    a. **Select Data Connector**: Select a data connector from the drop-down menu

    b. **Select Data Service**: Select a data service from the drop-down menu

    c. Based on the selected data service the below-given columns will be displayed

        i. Column Header

        ii. Data Type

vi) Click '**NEXT**'

vii) Users will be redirected to the '**Conditions**' tab (If conditions are available, else users will be redirected to the 'Mapping' page)

viii) Configure the required '**Conditions**' fields

ix) Click '**NEXT**'

x) Users will be redirected to the '**Mapping**' tab.

xi) Configure the column header information from the data service that will be used for the selected model columns.

xii) Click '**NEXT**'

xiii) Users will be redirected to the '**Data Writer**' tab.

Note: The '**Data Source**' tab will be enabled, only if users select '**No**' for 'Use Existing Data Connector' option while configuring the '**General**' tab for a new schedule.

5.8.1.3. Configuring a Data Writer

The Data Writer fields are reliant on the selected data writer types. The scheduler is provided with two kinds of data writers: 1. Data Writer and 2. Elastic Search Writer.

1. Data Writer

    i)   Fill in the required details to configure a data writer

    ii)   Click '**NEXT**'

iii) Users will be redirected to the '**Schedule**' tab.

2. Data Store Writer

Users can directly use the predictive workflows to create Business Stories if the workflows are written using the Elastic Search Writer.

i) Select '**Data Store Writer**' as a Data Writer Type to schedule a Predictive workflow.

ii) Users will be directed to create Hierarchy Definition.

iii) Drag and drop the required dimensions to define hierarchical drill.

iv) Click '**NEXT**'
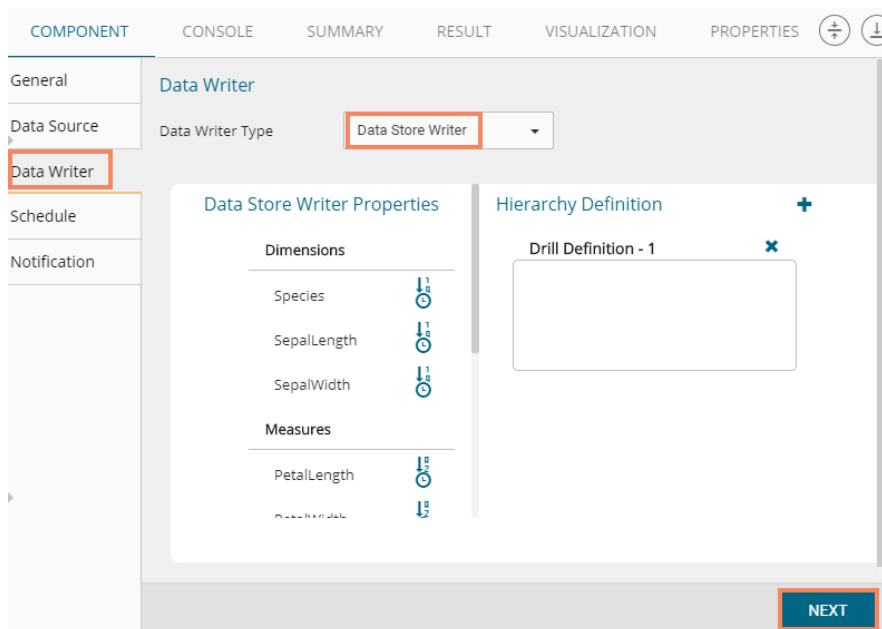
v) Users will be redirected to the '**Schedule**' tab.

Note: The '**Data Writer**' tab will be enabled, only if users select '**No**' for '**Use Existing Data Writer**' while configuring the '**General**' tab for a new schedule.

5.8.1.4. Scheduling a New job

Users can select a time to schedule a new job using this section. As per the selected scheduling time, refresh interval option will be provided.

i) **Start Date**: Select a start date and time for the scheduled job (It should be higher than **the Current System Date and Time)**

ii) **Select a Job Refresh Interval option**:
E.g., When the selected time range is '**Hourly**,' the selected interval option can be as described below:
Every_hour: Selecting this option will refresh the scheduled job after every selected interval.

OR

At: Selecting this option will refresh the scheduled job at the selected hour.

iii) **Start Time**: Select a start time higher than the current system time.

iv) **End Date**: Select an end date and time for the scheduled job. (It should be higher than the Start date and the Current System Date and Time)

v) **Run Now**: Select this option to run the scheduled job on applying.

vi) Click '**NEXT**'

vii) Users will be redirected to the '**Notification**' tab.

5.8.1.5. Job Refresh Intervals Details

- **Hourly:** By selecting this option users can schedule the job on an hourly basis.
  1. Select a specific hour by using the below-given options:

     Every_hour: Selecting this option will refresh the scheduled job after the selected hourly interval.

     OR

     At: Selecting this option will refresh the scheduled job at the selected hour.

- **Daily:** By selecting this option users can schedule the job on a daily basis.
  1. Select a specific day by using the below-given options:
     Every_ Days: the scheduled job will be refreshed after every selected number of days. E.g., if 2 is selected then, the scheduled job will be refreshed every alternate day at the set time.

     OR

     Every Week Day: the scheduled job will be refreshed daily till the end date.
  2. Select the Start time.

- **Weekly**: By selecting this option users can schedule the job on a weekly basis. Select a day **or** days of the week when the scheduled job can be refreshed.

- **Monthly:** By selecting this option users can schedule the job on a monthly basis. This time **the** range can be used to set schedule refresh for more than a month.
  Select a specific day of the month by using the below given options:
       E.g., Set monthly refresh interval (E.g., the first day of every month)
              OR
       Set a specific day after the desired monthly interval (the first Monday of the every month)

- **Yearly**: By selecting this option users can schedule the job on a yearly basis. This time range is provided for jobs that run more than one year.

  Select a specific day of the month by using the below-given options:

  Set a date for any month (E.g. The 1st January of every year till it approaches the end date)

  Or

  Select a day of any month (E.g. The 1st Monday of January every year till it approaches the end date)

- **Custom Cron Expression:** Users can schedule more flexible and customizable schedule runs by using the 'Custom Cron Expression' option. The scheduled workflow can be more specific with the custom cron expression that supports timing up to minutes and seconds. Users need to enter a valid Cron Expression in the given field.

Note: By selecting the '**Use Existing Data Connector**' and '**Use Existing Data Writer**' options '**Schedule**' tab will be displayed immediately after the '**General**' tab.

5.8.1.6. Notification

After selecting a schedule and clicking 'NEXT' users will be redirected to the 'Notification' section

i) Configure the below-given fields:

a. **Enable Email Notification**: Use a check mark in the box to enable email

b. **Email Address**: Enable this option by check marking the box

c. **Send Mail when Server is not running**: Users can check mark in the box to enable this option. By enabling this option, users will get an email when R server is not running.

d. **Send Mail when Process is Completed Successfully**: Users can check mark in the box to enable this option. By enabling this option user will get mail after the process is completed.

e. **Send Mail when the Process is a Failure**: Users can check mark in the box to enable this option. By enabling this option user will get an email when the process fails.

ii) Click '**APPLY**'

iii) A success message will pop-up to assure that the job/process has been scheduled

Process has been scheduled. ✕

iv) The scheduled job/ process will be added to a list provided under the '**Status**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | | | |

🗘 Refresh

Search: [_____]

Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
job_sanityCheck	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏ ✖ ▶
wf_sanityTest	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		Workflow_Save	iris_new	View Logs	✏ ✖ ▶
jobcheckIssue	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏ ✖ ▶
jobCheckJOBBBB	Hourly	14/Feb/2018-22:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏ ✖ ▶
Scheduler Job	Yearly	8/Apr/2018-1:0:0	28/Apr/2019-0:0:0	1/Apr/2019-12:0:0	Active		Scheduler_Workflow	iris_Filter	View Logs	✏ ✖ ▶

Showing 81 to 85 of 85 entries   Previous   1 … 5   6   7   8   **9**   Next

Note:

a. The PDF summary will be sent through email for the scheduled workflows.
b. Multiple email addresses can be entered in coma separated value.
c. At present, Spark Workflows are not supported by Scheduler.

5.8.2. Status

This section will display detailed information for all the scheduled jobs.

i) Click the '**Scheduler**' tree node
ii) Select '**Status**'

iii) Users will be redirected to the Component tab
iv) A list containing all the scheduled jobs will be displayed

↻ Refresh

Search:

Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
job check sch	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		chck_sch_1	iris	View Logs	✏ ■ ✖ ▶
job sch	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		sch_check	iris	View Logs	✏ ■ ✖ ▶
job for sch333	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		sch_check111	teadata	View Logs	✏ ■ ✖ ▶
sch	Hourly	3/Jan/2018-14:0:0	3/Jan/2018-16:0:0	NA	Stopped		CreditCard_Scoring	German_data	View Logs	✏ ■ ✖ ▶
sch	Hourly	3/Jan/2018-15:0:0	3/Jan/2018-16:0:0	NA	Stopped		samplech	iris	View Logs	✏ ■ ✖ ▶
bs_ccc	Hourly	19/Jan/2018-21:0:0	19/Jan/2018-22:0:0	NA	Stopped		check_BS_CNR	iris	View Logs	✏ ■ ✖ ▶
job_sch_mails	Hourly	29/Jan/2018-16:0:0	29/Jan/2018-17:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
check_R sch	Hourly	29/Jan/2018-17:0:0	29/Jan/2018-18:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
job_sch_auto	Hourly	29/Jan/2018-18:0:0	29/Jan/2018-19:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
jobbbb	Hourly	29/Jan/2018-18:0:0	29/Jan/2018-19:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶

Showing 1 to 10 of 85 entries   Previous 1 2 3 4 5 … 9 Next

a. Click '**View Logs**' to see the logs of the selected workflow under the '**Component**' tab

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

06/Apr/2018 - 05:00:50	DataReaderProcess is started.
06/Apr/2018 - 05:00:53	Number of Rows fetched : 150
06/Apr/2018 - 05:00:53	DataReaderProcess is completed.
06/Apr/2018 - 05:00:53	R-CNR Tree1 is started.
06/Apr/2018 - 05:00:54	R-CNR Tree1 is completed.
06/Apr/2018 - 05:00:54	Data Store Writer is started.
06/Apr/2018 - 05:00:55	Data Store Writer is completed.

Related Actions for a Scheduled Job:

Options	Name	Description
✏	Edit	To edit/update the scheduled job details
■	Stop	To stop the scheduled job
✖	Remove	To remove the scheduled job from the list
▶	Start	To start the scheduled job

Note:

a. '**Edit**' option will allow the user to update/ edit all the tabs for the selected job.
b. Users can click the '**Start**' button to restart the scheduler for a scheduled job until it reaches the end date.
c. Users can enable '**Edit**' and '**Remove**' actions only after stopping the Scheduled job.

5.8.3.   Saved R-Scripts
This section describes options that can be applied to a saved R Script.

5.8.3.1.   Viewing a Saved R Script
i)   Select an R Script from the list of '**Saved R-Script**'

ii) Right-click on the selected R Script.
iii) A context menu will open.
iv) Select '**View**'
v) Users will be redirected to the '**Component**' tab of the selected saved R Script.
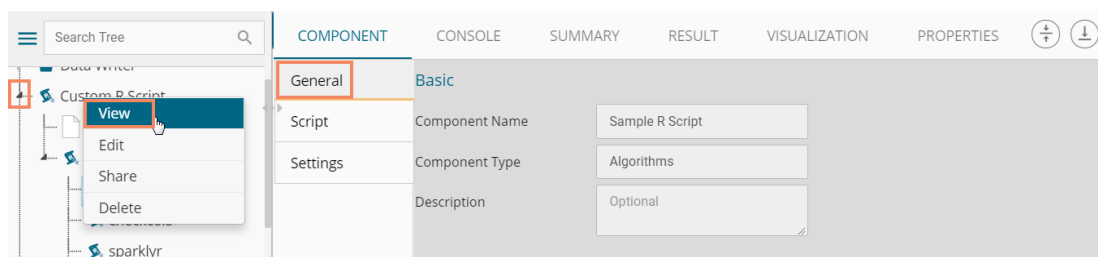
5.8.3.2. Editing a Saved R Script

i) Select an R Script from the list of '**Saved R-Script**'
ii) Right-click on the selected R Script.
iii) A context menu will open
iv) Select '**Edit**'
v) Users will be redirected to the '**Component**' tab
vi) Users can edit the required fields provided under **General**, **Script**, and **Settings** tabs
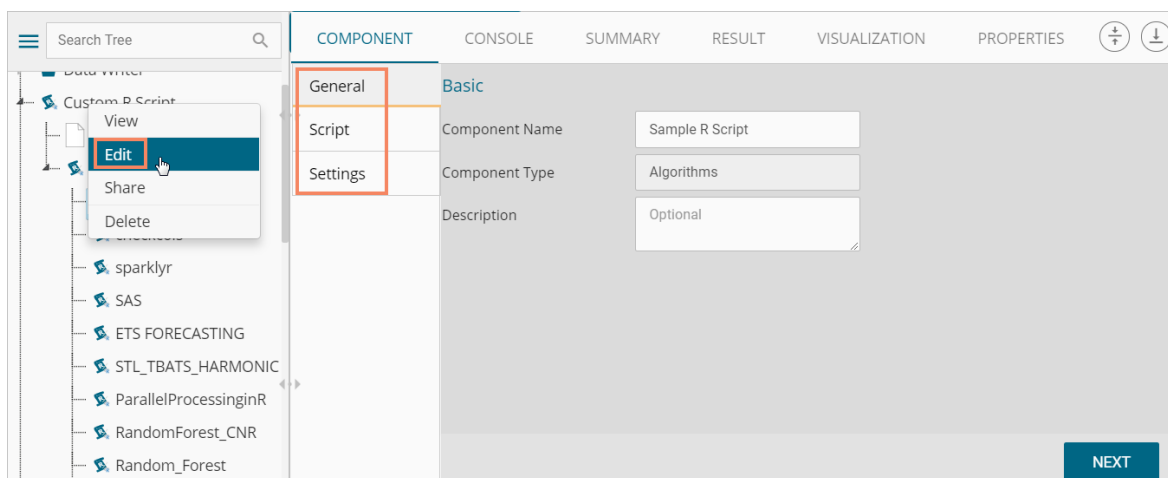
5.8.3.3. Sharing a Saved R Script

This feature gives users the ability to share a custom R script with other users and groups.
The following options are available to share a custom R script:

1. **Share With:** This option allows the user to share a custom R script with selected users or user groups. Any changes made to the custom R script will be transferred to all the users with whom the custom R script has been shared.

   i) Right-click on a saved R script from the list of '**Saved Scripts**'
   ii) Select '**Share Custom R Script**' from the context menu.
   iii) The '**Share With**' option will be displayed (by default)
   iv) Select either '**Group**' or '**Users**'
       a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
       b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.

v) Select a specific user or group from the list by check marking the box.
vi) Click '**APPLY**'
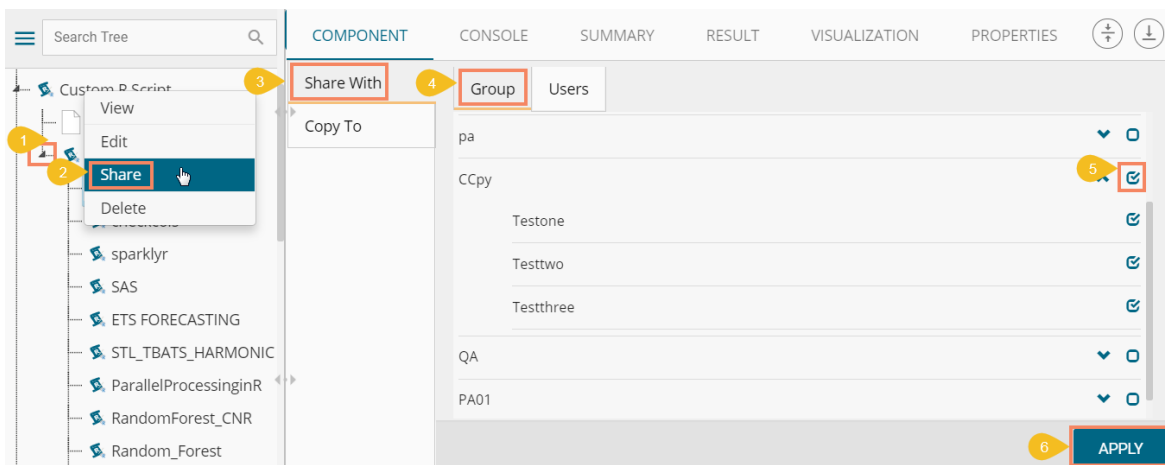
vii) The selected saved R script will be shared with the chosen user(s)/group(s).

2. **Copy To**: This option creates a copy and shares the copy of the custom R script with the selected users and user groups. Any changes to the original custom R script after sharing will not show up for the users that received the shared file via the '**Copy To**' option.

i) Right-click on a saved R script from the list of '**Saved Scripts**'
ii) Select '**Share Custom R Script**' from the context menu.
iii) Select '**Copy To**' option.
iv) The copied custom R script name will be displayed in a box.
v) Select either the '**Group**' or '**Users**' tab.
   a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
   b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
vi) Select a specific group or user from the list by check marking the box.
vii) Click '**APPLY**'
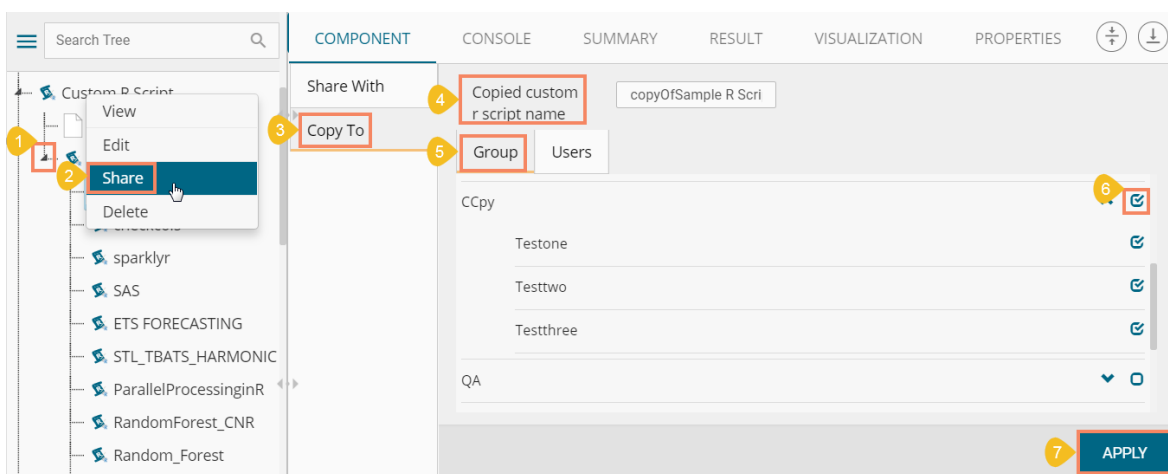
viii) The selected saved R script will be copied to the selected user(s)/group(s).

5.8.3.4. Deleting a Saved R Script
i) Select an R Script from the list of '**Saved R-Script**'

ii) Right-click on the selected R Script.
iii) A context menu will open.
iv) Select '**Delete**.
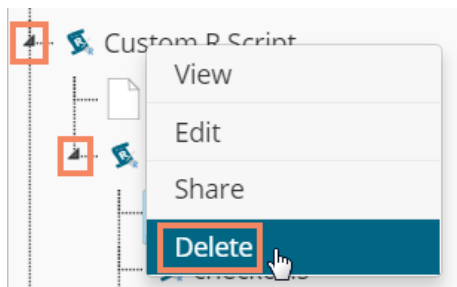
v) A pop-up window will appear to assure the deletion.
vi) Click '**OK**'
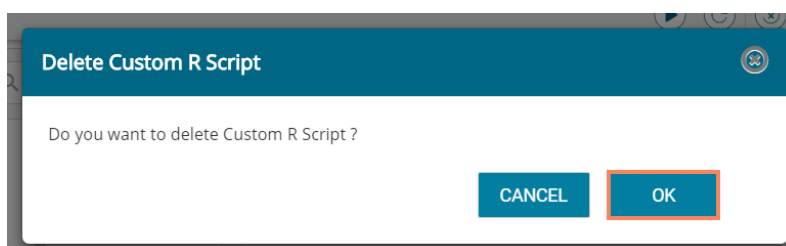
vii) The selected R-Script will be deleted.

5.8.3.5. Connecting Saved R Script with a Data Source

i) Click the '**Custom R Script**' tree node
ii) Select and drag a saved R-script to the workspace
iii) Connect the R-Script to a configured data source component
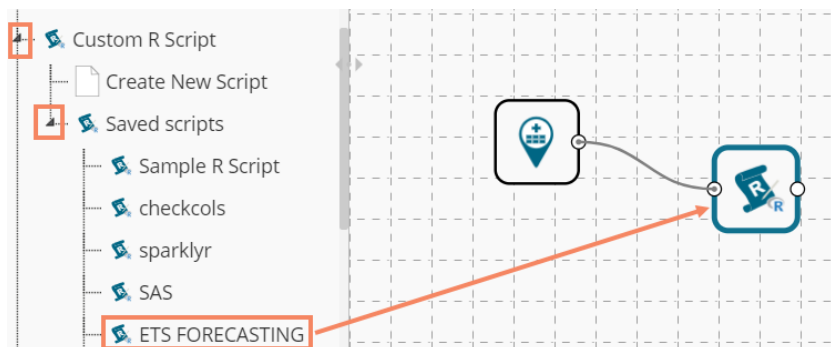
iv) Click the '**R Script**' component
v) Configure the required component fields
vi) Click '**APPLY**'

vii) After getting the success message run the workflow

viii) Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY |

10/7/2018 - 12:29:55   : Process Initiated...

10/7/2018 - 12:29:56   : Data Service0 is started.

10/7/2018 - 12:30:52   : Data Service0 is completed.

10/7/2018 - 12:30:52   : Custom R Script1 is started.

10/7/2018 - 12:30:58   : Custom R Script1 is completed.

ix) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 entries   Search:

PRODUCTS_NAME	PRODUCTS_ID	YEAR	MONTH	QUANTITIES	periodname
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	1	547	Jan 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	2	1002	Feb 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	3	1537	Mar 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	4	476	Apr 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	5	903	May 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	6	666	Jun 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	7	1140	Jul 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	8	823	Aug 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	9	889	Sep 2015
PhytoBright Whitening Day Lotion with SPF 20***	TS14007923	2015	10	797	Oct 2015

Showing 1 to 10 of 49 entries   Previous 1 2 3 4 5 Next

x) Click the '**VISUALIZATION**' tab

xi) Users will get a visual representation of the result data

Forecasts from ETS(M,N,N)
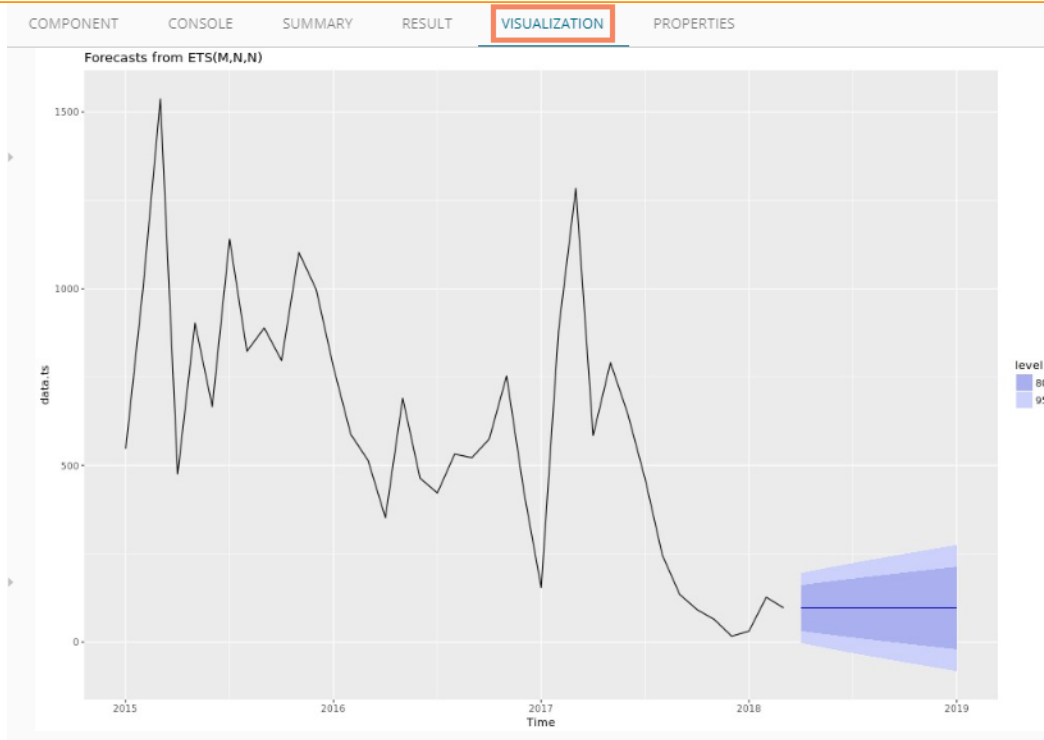
Note:

a. The above-given process is displayed for a CSV data source. A similar set of steps can be followed for other data source types.
b. A new tree-node 'Pre-Defined Scripts' is provided under the 'Custom R Script' tree-node with a list of predefined scripts focusing on various business verticles to facilitate the users.
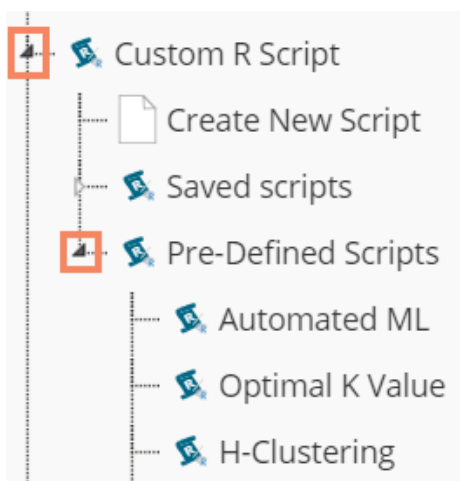
5.9. Saved Workflows

Users can save a workflow by clicking the '**Save**' button provided on the workspace menu row. All the Save workflows will be displayed under the '**Saved Workflow**' tree node. This section explains various options
assigned to a saved workflow.

i)   Click '**Saved Workflow**' tree-node to display a list containing all the saved workflows
ii)   Select a workflow from the list and user right-click to open the context menu
iii)   A context menu will open with various options (As shown below):
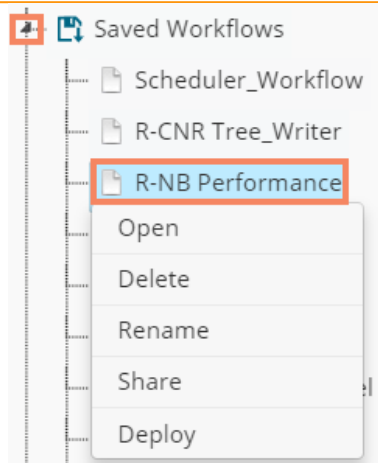
5.9.1. Opening a Workflow

i)   Right-click on a workflow from the list of **'Saved Workflows'**
ii) Select **'Open'** from the context menu
iii) The selected workflow will be displayed in the right pane of the screen
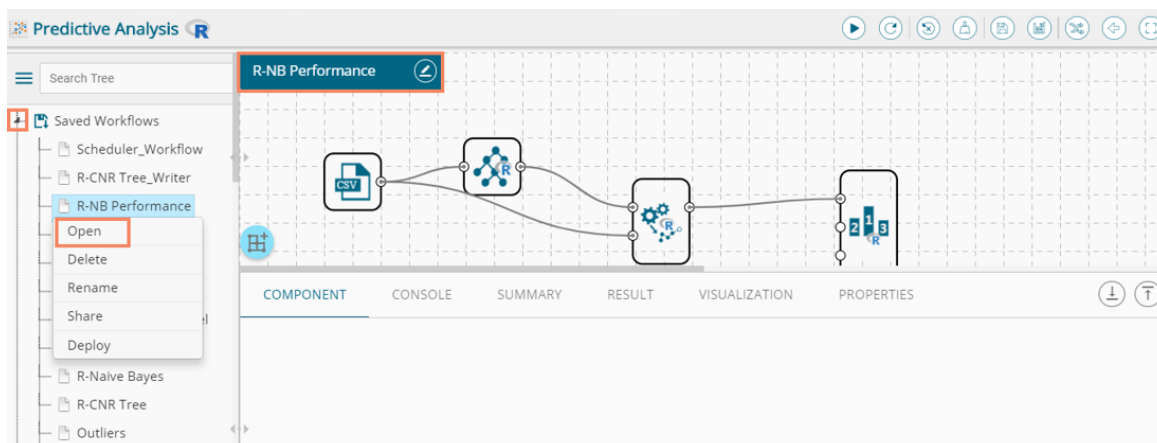
Note: The workflow name will be displayed on the left side of the workspace menu row while opening a workflow.
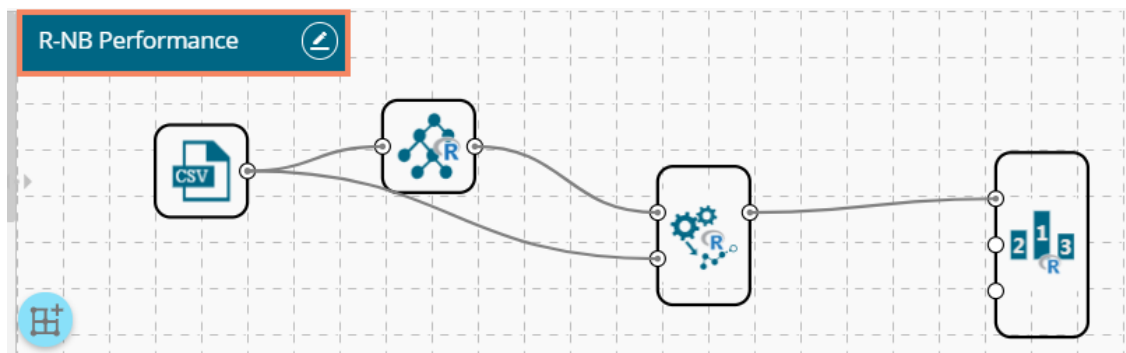
5.9.2. Deleting a Workflow

i)   Right-click on a workflow from the list of **'Saved Workflows'**
ii) Select **'Delete'** from the context menu
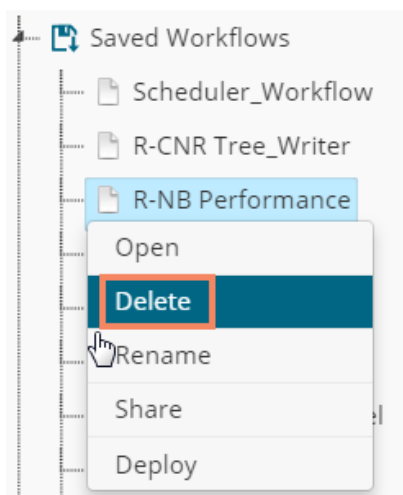
iii) A message window will pop-up to confirm the deletion
iv) Click '**OK**'
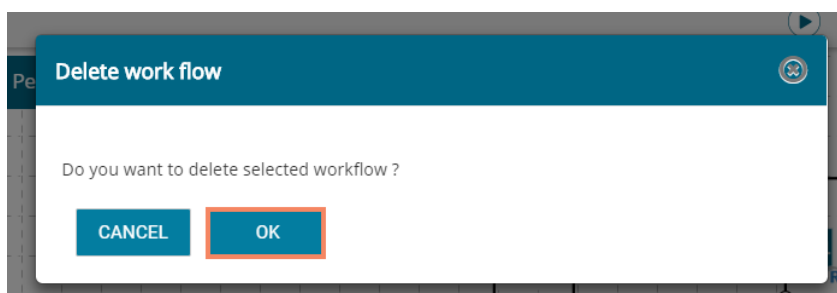
v) The selected workflow will be removed from the list

5.9.3. Delete Connection in a Workflow

A Right click on the inter-node connection will display the '**Delete Connection**' option in a workflow. Click the '**Delete Connection**' option to delete a connection.
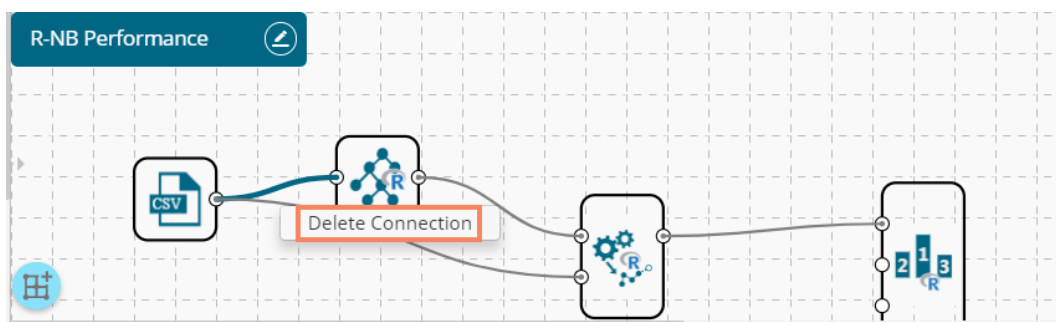
5.9.4. Renaming a Workflow

i) Press a right click on a workflow from the list of '**Saved Workflows**'
ii) Select '**Rename**' from the context menu
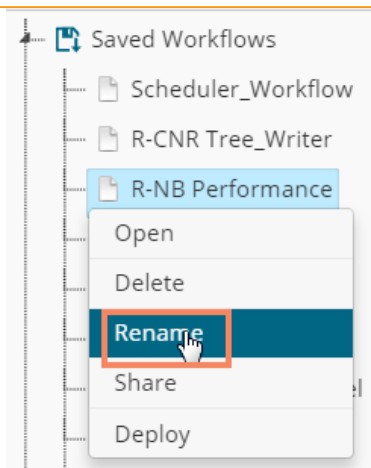
iii) A pop-up window will appear
iv) Enter a new/modified name for the workflow
v) Click '**YES**'
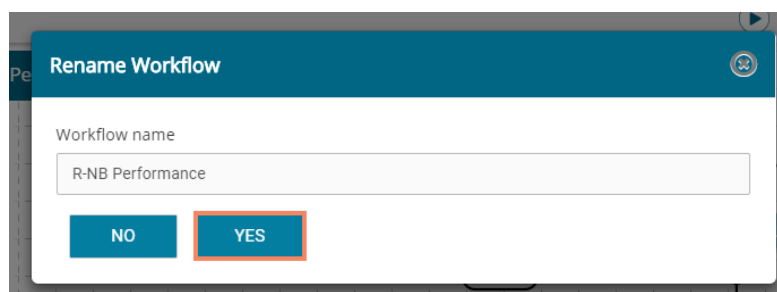
vi) The selected workflow will be renamed

5.9.5. Sharing a Workflow

This feature gives users the ability to share saved workflows with other users and groups.
The following options are available to share a selected workflow:

1. **Share With**: This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.

   i)   Press a right click on a workflow from the list of '**Saved Workflows**'
   ii) Select '**Share**' from the context menu
   iii) The '**Share With**' option will be displayed (by default)
   iv) Select either '**Group**' or '**Users**'
       a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
       b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
   v)   Select a specific group or user from the list by check marking the box
   vi) Click '**APPLY**

vii)   The selected workflow will be shared with the chosen user(s)/group(s)

2.   **Copy To**: This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the '**Copy To**' method.

   i)   Press a right click on a workflow from the list of '**Saved Workflows**'
   ii)   Select '**Share**' from the context menu
   iii)   Select '**Copy To**'
   iv)   The copied workflow name will be displayed
   v)   Select either '**Group**' or '**Users**'
       a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
       b. Users can be excluded by not selecting a username from the list when '**User**' option Has been selected
   vi)   Select a specific group or user from the list by check marking the box
   vii) Click '**APPLY**'

viii)   The selected workflow will be copied to the chosen users/groups

5.9.6.   Deploying a Workflow
The Predictive Workflows can be deployed to the BizViz Dashboard Designer.

i)   Press a right click on a Workflow from the list of 'Saved Workflows'
ii) Select 'Deploy Workflow' from the context menu
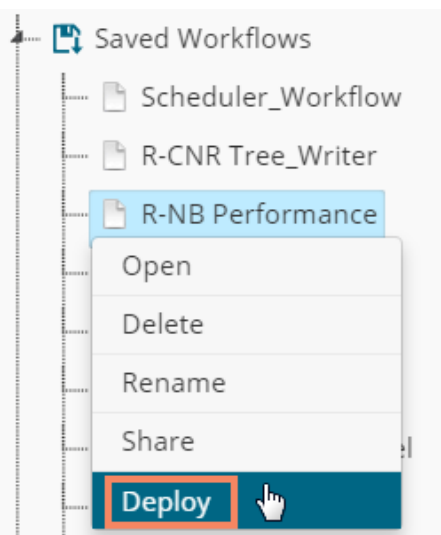
iii) Users will be redirected to select an Apply Model component from the workflow
iv) Select an Apply Model component and click the 'YES' option
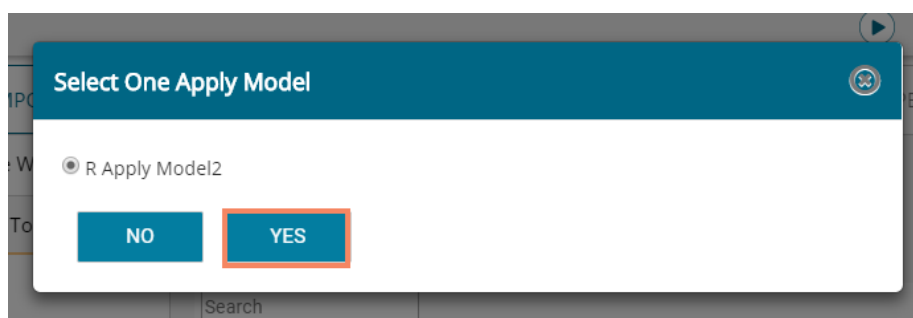
v)   A success message will pop-up to assure that the workflow has been published successfully
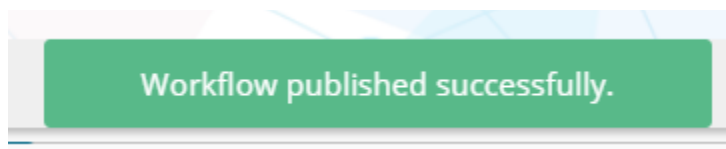
vi) A checkmark will be added to the selected workflow name
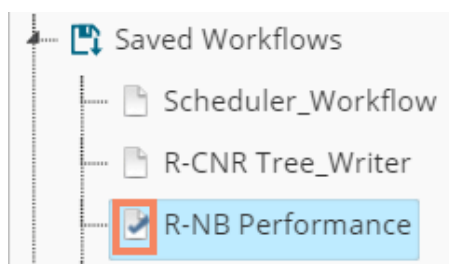
vii) Navigate to the Dashboard Designer home page
viii) Click 'New'
ix)   Click 'Dashboard'
      Or
      Click on the 'Add Dashboard' option
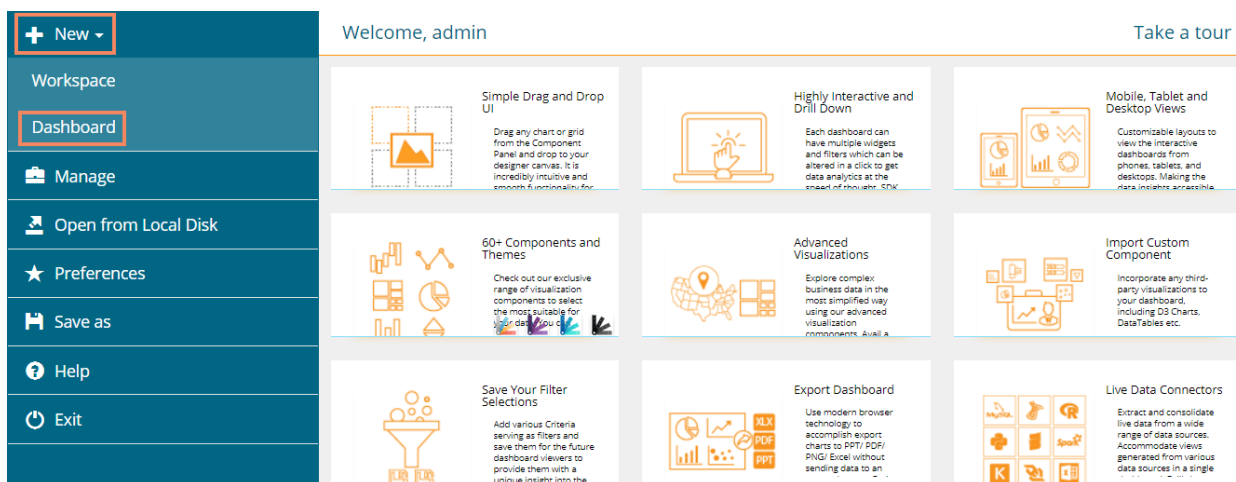
x) Users will be directed to the Dashboard canvas

xi) Click the '**Data Source**' icon 🗄 to display all the available data sources

xii) Click the '**Create New Connection**' option ➕ provided next to the '**Predictive Service**' data source

xiii) A new connection will be created and added below
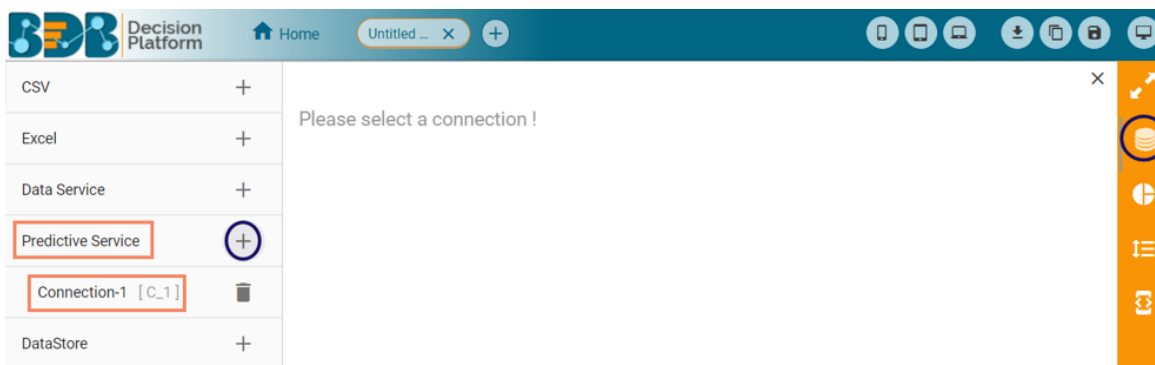
xiv) Click on the connection to display the connection specific details

xv) Select the deployed Predictive workflow as a data source via the drop-down menu
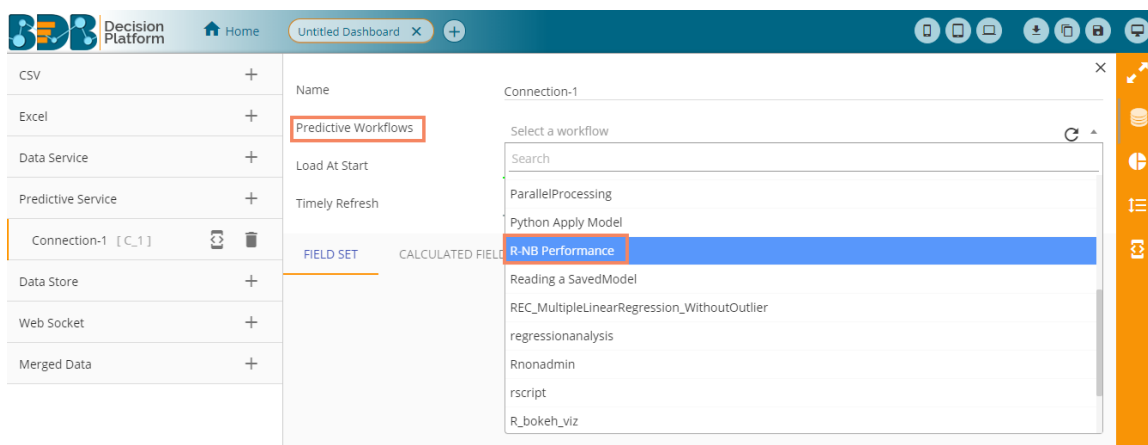
xvi) Configure the other subsequent details:
- a. Load At Start: Enable this option to get the updated data
- b. Timely Refresh: Enable this option to refresh data
- c. Refresh Interval: Select the time interval to refresh the data

Name	Connection-1
Predictive Workflows	R-NB Performance
Load At Start	Yes No
Timely Refresh	Yes No
Refresh Interval	5 Minute(s)

FIELD SET   CALCULATED FIELDS   CONDITION

diameter

height

length

PredictedValues1

rings

d. Once the data connection is established the selected predictive workflow can be used as a connection to the Dashboard Designer for fetching data

Recommendations

▪ **R Workflows**: The result set located before a data writer component within a deployed R workflow will be considered as a data set by the Dashboard Designer.

Note: If a deployed Predictive Workflow has a summary, it can be viewed using the Dashboard Designer tool.

5.10. Saved R Models

R Apply Model is a component used to generate predictions based on the trained classification or regression model. The user can either split the dataset into training and testing, create a model with training data and apply the testing data. Another approach is to save the model and apply the model over new test data set.

Users can save an R model after successful execution. The saved R models will be listed under the '**Saved R Model**' tree node. Users can select a saved R model from the list and use to create a new workflow.

R Apply Model will come as a leaf node under Apply model tree node. The R Apply Model Component consists of two nodes for reading data from the data source and another one for giving the result.

5.10.1. Saving an R Model

i) Open an R workflow
ii) Connect '**Apply Model**' component with the workflow (as shown below)
iii) Right-click on the '**Apply Model**' component
iv) A context menu will open
v) Select '**Save Model**'
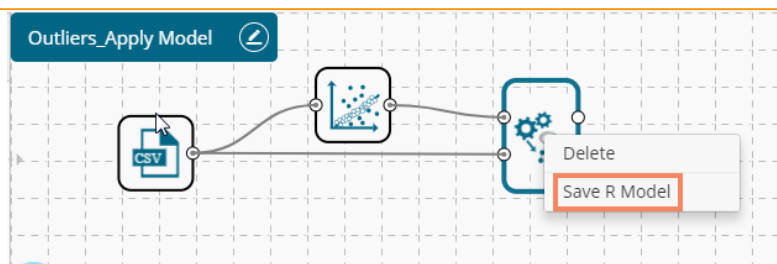
vi)   A new window will pop-up

vii)   Enter a name for the model that you wish to save

viii) Click '**OK**'
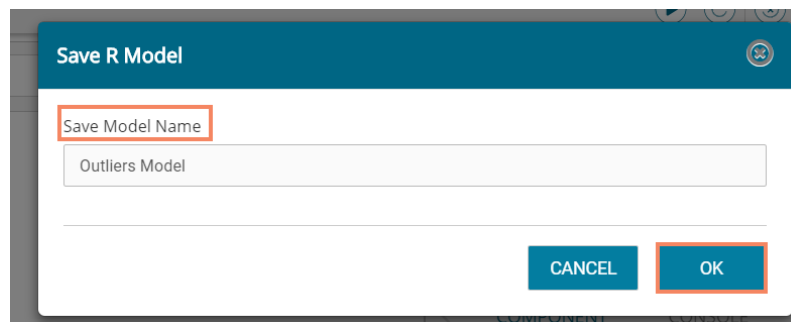
ix)   The created Predictive Model will be saved to the **'Saved R Models'** list
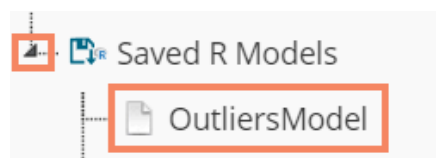
5.10.2. Reading an R Model

Users can drag a saved model to the workspace and reuse the model for test data. A saved R model can be connected to only Apply Model and new test data source.

i)   Select and drag a saved R model component onto the workspace.

ii)   Connect the dragged model with a configured data source and an Apply Model component (As shown in the following image).
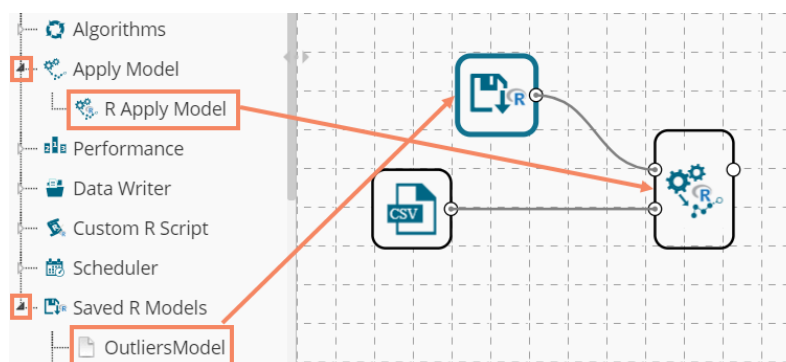
iii)   Click on the dragged Saved Model component.

iv)   Users will be able to view the following '**Component**' tabs:

a. General
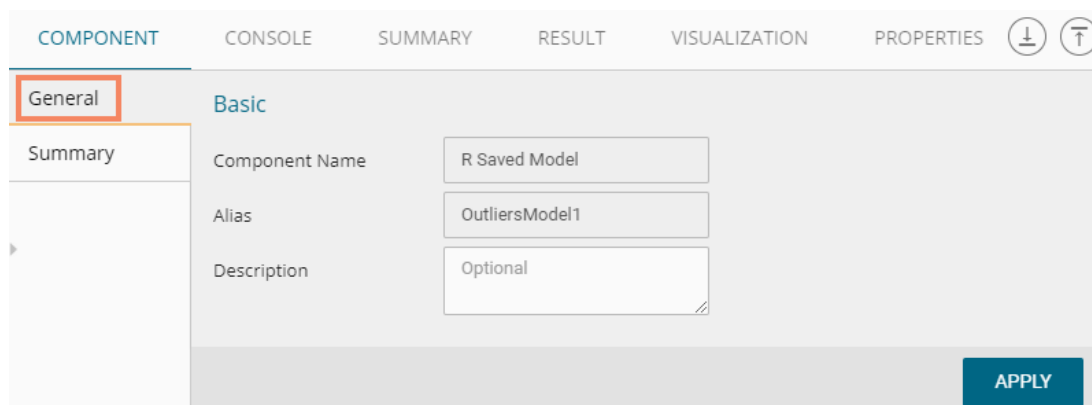
b. Click '**SUMMARY**' tab to display the model summary

v) Click '**APPLY**' using the Apply Model component.
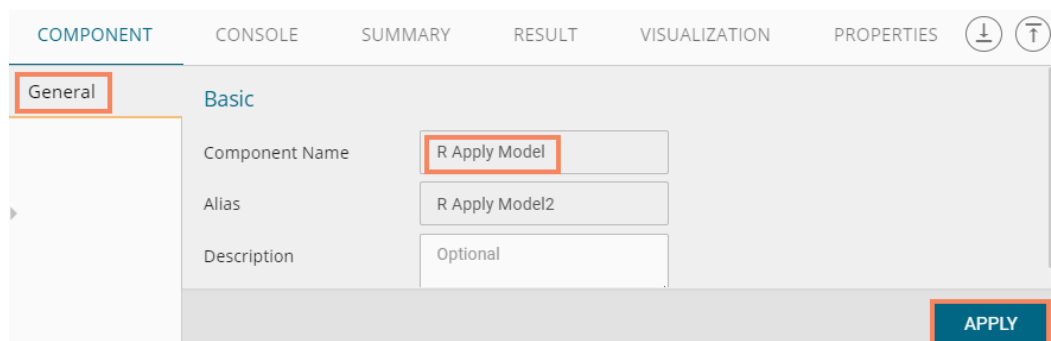
vi) After getting the success message run the workflow

vii) Users will get the process status under the '**CONSOLE**' tab

	CONSOLE	SUMMARY	RESULT
COMPONENT			

13/4/2018 - 19:28:12   : Process Initiated...

13/4/2018 - 19:28:13   : OutliersModel1 started.

13/4/2018 - 19:28:13   : OutliersModel1completed.

13/4/2018 - 19:28:13   : CSV0 is started.

13/4/2018 - 19:28:14   : CSV0 is completed.

13/4/2018 - 19:28:14   : R Apply Model2 is started.

13/4/2018 - 19:28:14   : R Apply Model2 is completed.

viii)   After the process gets completed under the Console tab, click the '**RESULT**' tab to see the result view of data.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

Month	Day_of_month	Day_of_week	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	I
1	1	4	3.01	5480	8	20			50
1	2	5	3.2	5660	6		38		
1	3	6	2.7	5710	4	28	40		26
1	4	7	5.18	5700	3	37	45		59
1	5	1	5.34	5760	3	51	54	45.32	14
1	6	2	5.77	5720	4	69	35	49.64	15
1	7	3	3.69	5790	6	19	45	46.4	26
1	8	4	3.89	5790	3	25	55	52.7	55
1	9	5	5.76	5700	3	73	41	48.02	20
1	10	6	6.94	5700	3	59	44		26

Showing 1 to 10 of 358 entries   Previous   1   2   3   4   5   ...   36   Next

Note:

a. A mandatory condition to run the workflow with a '**Saved R Model**' component is that column headers and data type of the test data source should match with the selected saved model. Users will encounter an error if validation fails while running the workflow.

b. Users can connect a data writer to the '**Apply Model**' component in a workflow containing a saved model.

5.10.2.1.   Renaming an R Model

i)   Select a model from the '**Saved R Models**' list

ii)   Right-click on the selected model

iii)   A context menu will open

iv)   Select '**Rename**'
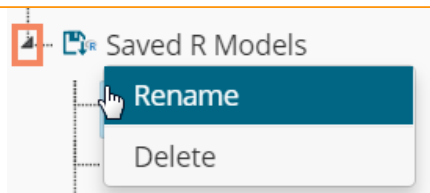
v)   A pop-up window will appear to rename the model

vi) Enter a new **'Model Title'** or modify the existing model title in the given field (if desired)

vii) Click **'YES'**
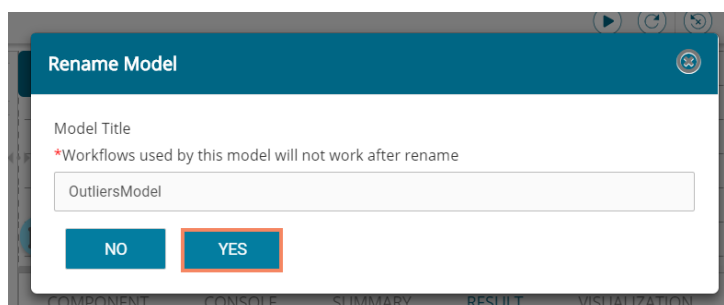
viii) The selected R Predictive Model will be renamed

Note: Workflows used by this model will not work after users rename the model.

5.10.2.2.   Deleting an R Model

i)   Select a model from the '**Saved R Models'** list

ii)   Right-click on the selected model

iii) A context menu will open

iv)   Select **'Delete'** from the menu
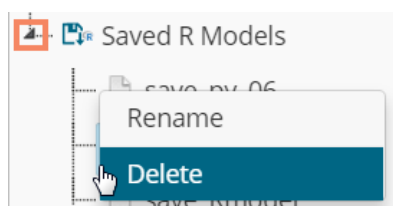
v)   A pop-up window will appear to confirm the deletion

vi)   Click **'OK'**
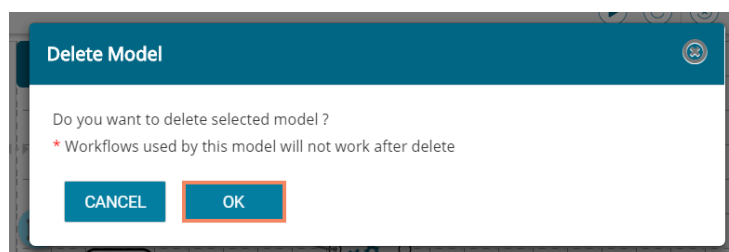
vii) The selected predictive model will be deleted and removed from the list of **'Saved R Models.'**

Note: After renaming or deleting a Saved R Model, workflows used by the same model don't work.

6. Spark Workspace

Users can select the Spark Workspace from the Predictive landing page to access the Spark Environment under the Predictive Workbench.
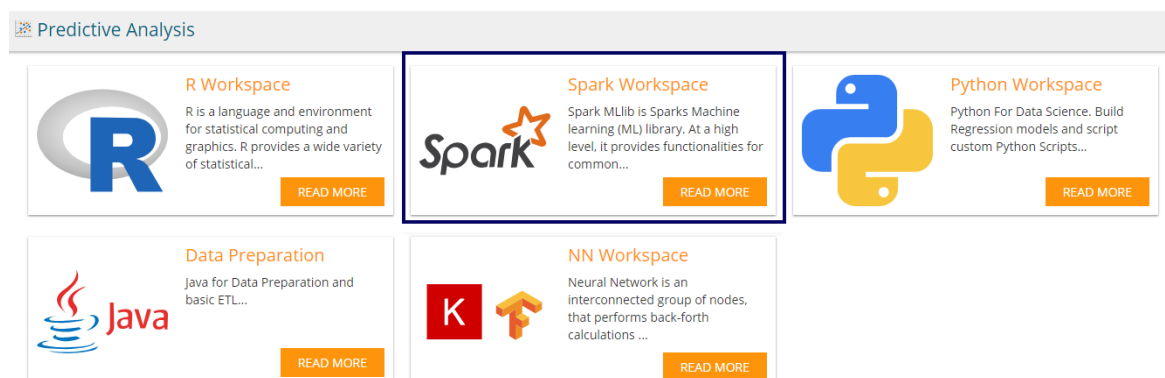
Users will be redirected to the following page by selecting the Spark Workspace:
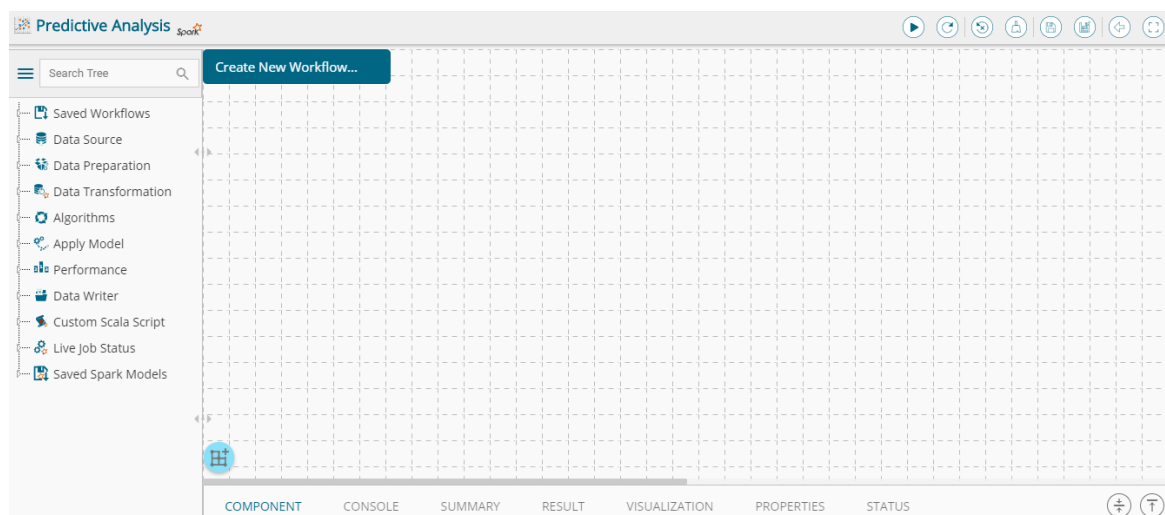
6.1. Data Source

6.1.1. Getting Data from a Data Service

i) Select and drag '**Data Service**' component onto the workspace.

ii) Click the '**Data Service**' component.

iii) Users will be redirected to the '**Properties**' fields provided under '**Components**' tab on the

Tabbed Menu Strip.

iv) Configure the **'Data Service Properties':**
  a. **Select Data Connector**: Select a data source from the drop-down menu
  b. **Select Data Service**: Select a query service from the drop-down menu
  c. **Fields:**
     The following tables will be displayed:
      i. Column Header
      ii. Data Type

v) Click **'NEXT'** (The **'NEXT'** option will appear only for the data service that has filters, otherwise the **'APPLY'** option will be displayed)

vi) Users will be redirected to the **'Conditions'** tab. (If the selected data service contains the filter values).

vii) Configure the following information:
  a. **Filter Type:** Available filter(s) in the data service will be displayed in this space.
  b. **Control Type:** Users are provided with the following options to pass the filter values under this option:
     • **Text:** By selecting this option users can manually enter multiple filter values separated by comma

     • **LOV:** By selecting this filter value option users will be directed to **choose** another Data Connector and Data Service available in the space

viii) Click **'APPLY'**

ix) Click the **'Run'** ▶ icon or click **'Refresh'** ↻ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the **'CONSOLE'** tab to display the progress of the process

xi) After the Console process gets completed, users can view the result data using the **'RESULT'** tab

xii) Follow the below given steps to display the result view:
   a. Click the dragged data source component on the workspace
   b. Click the **'RESULT'** tab

- **Rules to be Followed while Creating a Data Service**

1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
2. Data service header should not contain any special characters. E.g. - %, #, $, @,*, etc.
3. Data service header should not contain single or double quotes, dot, brackets, and high-fen.
4. Data service header should not contain merely numbers. Numerals should be used with at least one alphabet.
5. Data service header should not exceed 50 characters.

Note:
   a. Users can develop a data service via the Data Management module of the BizViz Platform.
   b. **'Fields'** option under **'Properties'** tab will appear only after selecting the appropriate query service.
   c. LOV service provided under the **'Conditions'** tab can contain only one column, in case of more than one column, a warning message will appear.
   d. Users can configure the following information for a data service data source via **'General'** tab:
      i. Alias Name
      ii. Description (it is an optional field)

6.1.2. Getting Data from a Cassandra Reader

i) Select and drag '**Cassandra Reader'** connector onto the workspace.
ii) Click on the '**Cassandra Reader'** connector.
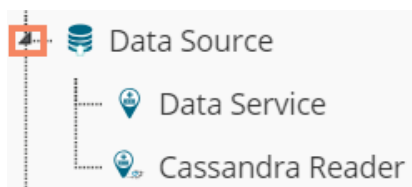
iii) Users will be redirected to the '**Properties'** tab of the component.
iv) Configure the required properties:
   a. Select Data Connector: Select a data connector using the drop-down menu
   b. Host Name: Data connector specific hostname will be displayed
   c. Port Number: Port number will be displayed
   d. User Name: Displays the username
   e. Password: Enter the password
   f. Cluster Name: Enter a cluster name
   g. Select Key Space: Select a keyspace from the drop-down menu
   h. Select Table: Select a table from the drop-down menu
   i. Limit No. of row to fetch: Select an option using the drop-down menu. By clicking the 'Limit No. of row to fetch' the following options appear:
      1. Select all Rows
      2. Limit By
   j. Max. No. of Rows to be fetched: Enter a number to decide maximum fetched rows. (This option appears only if '**Limit By'** option has been selected using the '**Limit by Row'** field. The Default value for this field is 1000).
v) Click '**NEXT'**

vi) Users get redirected to the 'Column Selection' tab.

vii) Select the required columns from the list.

viii) Click 'APPLY'

ix) Click the 'Run' ▶ icon or click 'Refresh' ↻ icon to run the workflow by clearing the Previous cache

x) Users will be redirected to the 'CONSOLE' tab to display the progress of the process

xi) After the Console process gets completed, users can view the result data using the 'RESULT.' tab

xii) Follow the below given steps to display the result view:

　　a. Click the dragged data source component on the workspace

b. Click the '**RESULT**' tab

| | | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | | |

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
6	1.7	0.4	5.4	3.9	0
80	3.5	1	5.7	2.6	1
75	4.3	1.3	6.4	2.9	1
57	4.7	1.6	6.3	3.3	1
113	5.5	2.1	6.8	3	1
67	4.5	1.5	5.6	3	1
118	6.7	2.2	7.7	3.8	1
82	3.7	1	5.5	2.4	1
120	5	1.5	6	2.2	1
112	5.3	1.9	6.4	2.7	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note: The Apache Spark workflows require a '**Cassandra Reader**' as a data source. The Cassandra Reader can also be used as a data source for the R Workflows.

6.2. Data Preparation

6.2.1. Spark Split Data

The Spark Split Data component is used to split a dataset into training and testing datasets. Once the most suitable model is decided from the trained data, users can pass test data to that model.

Spark Split Data appears as a leaf node under the Data Preparation Tree node.

The Spark Split Data consists of two connector nodes: Upper node for the **training dataset** and lower node for the **testing data set**.
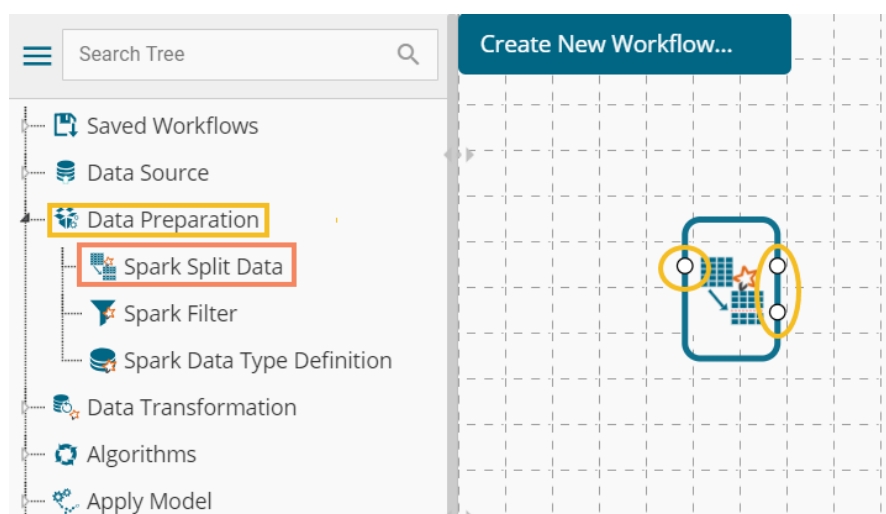
i)   Select the '**Spark Split Data**' component and connect it to a valid data source (in this case, select Cassandra reader)
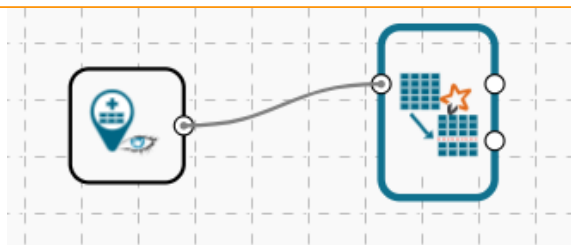
ii) Click the '**Spark Split Data**' component in the workspace

iii) Users will be directed to the Properties fields provided under the '**Components**' tab

iv) Configure the following Properties:

　　a. Relative (Train): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and sum of train and test should be 1).

　　b. Relative (Test): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and sum of train and test should be 1).

　　c. Seeds: Enter a numerical value. Default Value: 10. It is an optional field. Set the seed of Spark's random number generator, which is useful for creating simulations or random objects that can be reproduced. The random numbers are the same, and they would continue to be the same irrespective of how far in the sequence the users go. Use the seed function when running simulations to ensure all results, figures are reproducible.

v) Click '**APPLY**'

vi) After getting the success message run the workflow

vii) A message will pop-up to confirm whether users want to enable logging

viii) Click '**NO**'
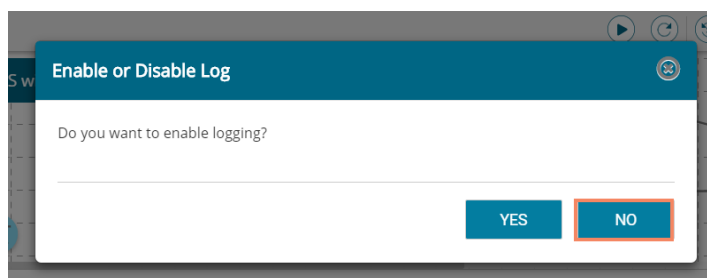
ix) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

14/4/2018 - 20:21:51   : Process Initiated...

14/4/2018 - 20:21:54   : Number of Rows fetched : 150

14/4/2018 - 20:21:54   : cassandra0 Completed

14/4/2018 - 20:21:54   : Spark Split Data1 Running

14/4/2018 - 20:21:54   : Spark Split Data1 Completed

14/4/2018 - 20:21:54   : Process Completed

x) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab
xi) The Result tab will contain two datasets separated by a sub-tab. As shown in the below-given images:
   a. Select the '**Split 1**' tab to see one set of data (the training dataset)

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

Split 1   Split 2

Show 10 ▾ entries   Search: [   ]

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
59	4.6	1.3	6.6	2.9	1
83	3.9	1.2	5.8	2.7	1
7	1.4	0.3	4.6	3.4	0
145	5.7	2.5	6.7	3.3	1
6	1.7	0.4	5.4	3.9	0
57	4.7	1.6	6.3	3.3	1
16	1.5	0.4	5.7	4.4	0
44	1.6	0.6	5	3.5	0
62	4.2	1.5	5.9	3	1
56	4.5	1.3	5.7	2.8	1

Showing 1 to 10 of 45 entries   Previous 1 2 3 4 5 Next

   b. Select the '**Split 2**' tab to see another set of data (the testing dataset)

| | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS | |

Split 1

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
111	5.1	2	6.5	3.2	1
42	1.3	0.3	4.5	2.3	0
93	4	1.2	5.8	2.6	1
106	6.6	2.1	7.6	3	1
114	5	2	5.7	2.5	1
128	4.9	1.8	6.1	3	1
135	5.6	1.4	6.1	2.6	1
75	4.3	1.3	6.4	2.9	1
80	3.5	1	5.7	2.6	1
5	1.4	0.2	5	3.6	0

Showing 1 to 10 of 105 entries   Previous 1 2 3 4 5 … 11 Next

6.2.2. Spark Filter

The Spark Filter has been added as a leaf node to the Data Preparation tree-node. Users can provide a filter condition appended by "@" to filter out data. Users should make sure that the given condition will return only true or false.

i)   Drag and configure the data source (in this case, select Cassandra reader)

ii)   Run the data source and check result data by clicking the '**RESULT**' tab

| | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS | |

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
6	1.7	0.4	5.4	3.9	0
80	3.5	1	5.7	2.6	1
75	4.3	1.3	6.4	2.9	1
57	4.7	1.6	6.3	3.3	1
113	5.5	2.1	6.8	3	1
67	4.5	1.5	5.6	3	1
118	6.7	2.2	7.7	3.8	1
82	3.7	1	5.5	2.4	1
120	5	1.5	6	2.2	1
112	5.3	1.9	6.4	2.7	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

iii) Drag the '**Spark Filter**' component onto the workspace

iv) Connect it to the configured data
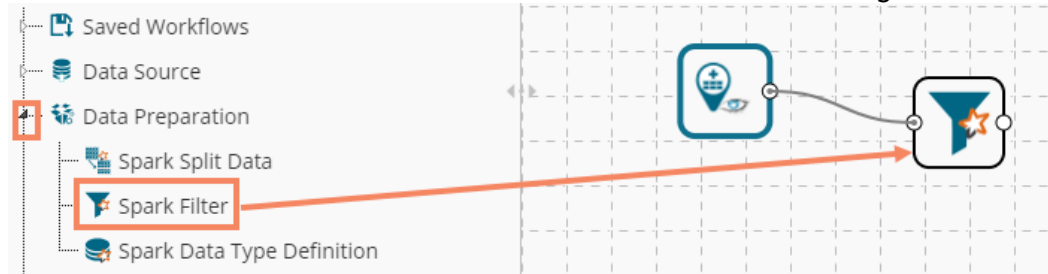
v) Right-click on the Spark Filter component
vi) Provide condition for the '**Row Filter**'
vii) Click '**NEXT**'
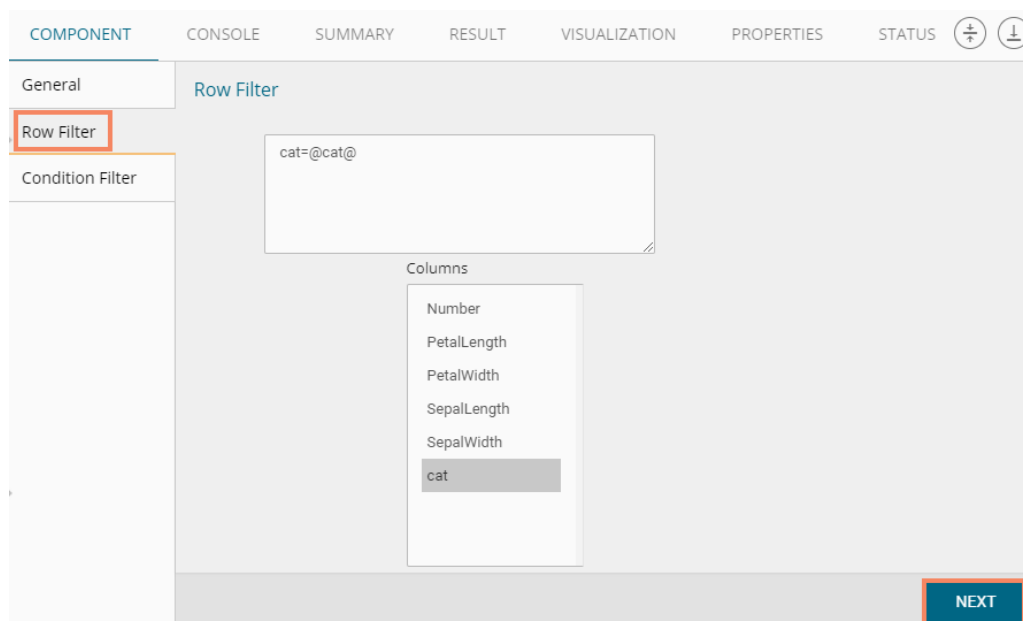
viii) Users will be directed to configure a condition for the '**Column Filter**'
ix) Click '**APPLY**'
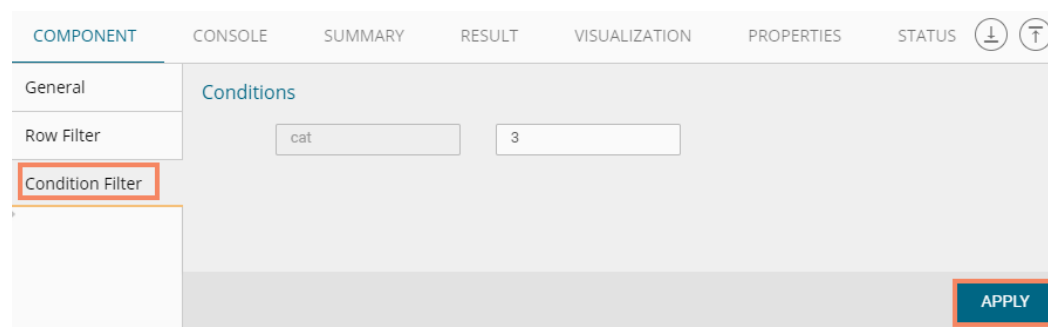
x) After getting the success message run the workflow
xi) A message will pop-up to confirm whether users want to enable logging
xii) Click '**No**'
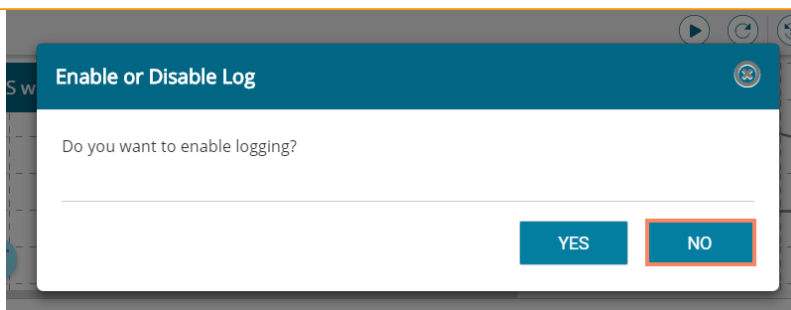
Enable or Disable Log

Do you want to enable logging?

[YES] [NO]

xiii) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY

14/4/2018 - 20:44:11 : Process Initiated...

14/4/2018 - 20:44:14 : Number of Rows fetched : 150

14/4/2018 - 20:44:14 : cassandra0 Completed

14/4/2018 - 20:44:15 : Spark Filter1 Running

14/4/2018 - 20:44:15 : Spark Filter1 Completed

14/4/2018 - 20:44:15 : Process Completed

xiv) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.
xv) The filtered result data will be displayed.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

Show 10 entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
46	1.4	0.3	4.8	3	0
14	1.1	0.1	4.3	3	0
31	1.6	0.2	4.8	3.1	0
10	1.5	0.1	4.9	3.1	0
29	1.4	0.2	5.2	3.4	0
45	1.9	0.4	5.1	3.8	0
39	1.3	0.2	4.4	3	0
4	1.5	0.2	4.6	3.1	0
25	1.9	0.2	4.8	3.4	0
47	1.6	0.2	5.1	3.8	0

Showing 1 to 10 of 50 entries   Previous 1 2 3 4 5 Next

6.2.3. Spark Data Type Definition

This component can be used to typecast data into another form. Users can change the data type of a column or change the alias name of the column using this component. Spark Data Type definition will appear as a leaf node under the Data Preparation tree node.

i)   Select the '**Spark Data Type Definition**' component and connect it with a valid data source (in this case, select Cassandra Reader as the data source)
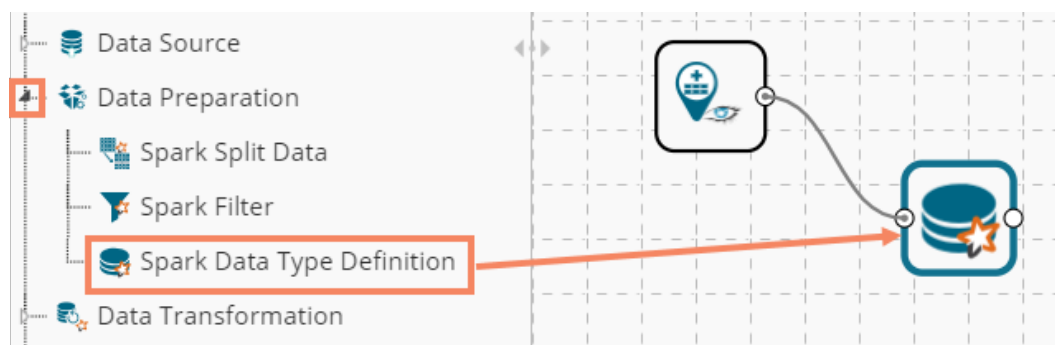
ii)   Configure the Properties fields for the Spark Data Type Definition component
iii) Configure the following '**Data Type Transformation**' details:
      a. **Column Name:** Select a column name which you want to change
      b. **Alias Name:** Enter an alias name for the required source column
      c. **Primary Data Type:** Select a primary data type column that you want to change
      d. '**Add' option ⁺:** Click on this button to add more columns to be transformed
iv)   Click '**APPLY**'
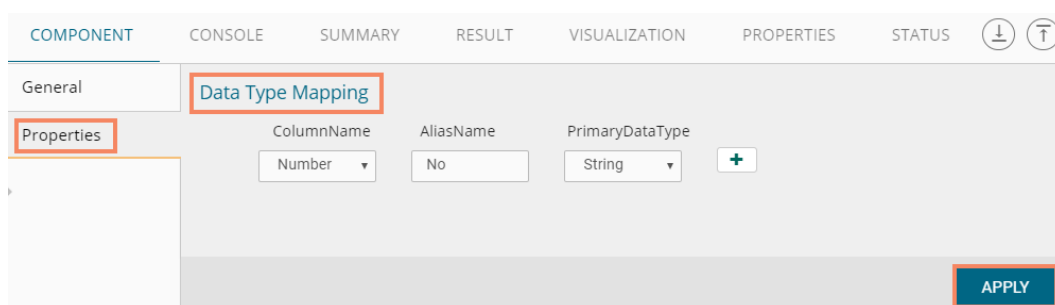
v)   After getting the success message run the workflow
      a. A message will pop-up to confirm whether users want to enable logging
      b. Click '**NO**'
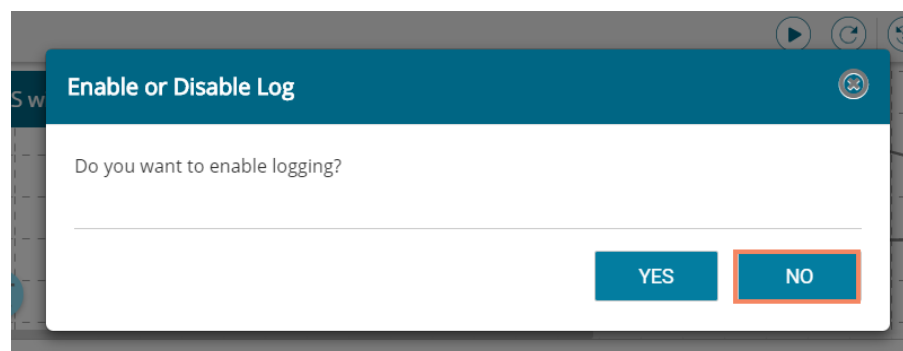
vi)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

14/4/2018 - 21:39:12   : Process Initiated...

14/4/2018 - 21:39:15   : Number of Rows fetched : 150

14/4/2018 - 21:39:15   : cassandra0 Completed

14/4/2018 - 21:39:15   : Spark Data Type Definition1 Running

14/4/2018 - 21:39:15   : Spark Data Type Definition1 Completed

14/4/2018 - 21:39:15   : Process Completed

vii)   Follow the below given steps to display the result view:
   a.   Click the data preparation component onto the workspace.
   b.   Click the '**RESULT**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

Show [10 ▼] entries   Search: [   ]

| PetalLength	PetalWidth	SepalLength	SepalWidth	cat	No
4.7	1.4	7	3.2	1	51
1.4	0.3	4.8	3	0	46
1.1	0.1	4.3	3	0	14
1.6	0.2	4.8	3.1	0	31
3.8	1.1	5.5	2.4	1	81
4	1.3	5.5	2.5	1	90
4.7	1.2	6.1	2.8	1	74
1.5	0.1	4.9	3.1	0	10
1.4	0.2	5.2	3.4	0	29
4.6	1.5	6.5	2.8	1	55

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

Note:
   a. Users cannot typecast the advanced column types (E.g., map, list, UDT), UUID, and timestamp.
   b. Spark Data Type Definition supports only Integer, Double, and String data types.
   c. Users need to click the Spark component and then click the 'Result' tab to display the result view for any Spark Component.
   d. Spark Data Preparation components support only Cassandra reader.

6.3.   Data Transformation

The Data Transformation components are pipeline components. Users need to connect an Apply Model component with these elements to complete workflow and get the results.

Standard Rules for all the Data Transformation Components:

a. The Data Transformation components can be connected to only those Data Preparation components that have '**Spark**' prefix in their names.
b. A '**Data Preparation**' component cannot be added in between the '**Data Transformation**' and '**Apply Model**' components in a workflow.
c. All the '**Data Transformation**' components are pipeline components. Results can be viewed only after connecting them to an '**Apply Model**' component.
d. End of the pipeline component should be an '**Apply Model**' component.

e. A model can be saved from the context menu of an '**Apply Model**' component.

6.3.1. String Indexer

Spark String Indexer converts a string column of labels to a column of label indices. The indices are in [0, numLabels), ordered by label frequencies, so the most common label gets index 0. If the input column is numeric, users can cast it to string and index the string values.

The Spark String Indexer comes as a leaf node under the Data Preparation tree-node.
The component consists of one node for input data and another for output data.

The BDB Predictive Analysis uses the Spark String Indexer to convert string label column to numerical column so that it can be applied to a specific algorithm which requires numerical column as label column. It consists of an option to select a label column from previous component headers. After choosing a label, column user can change the column header of the newly indexed column which is Label by default.

Users must set the input column of the component to this string-indexed column name when pipeline components such as Estimator or Transformer make use of this string-indexed label.

i)   Users need to select the String Indexer component and connect it with a configured data source
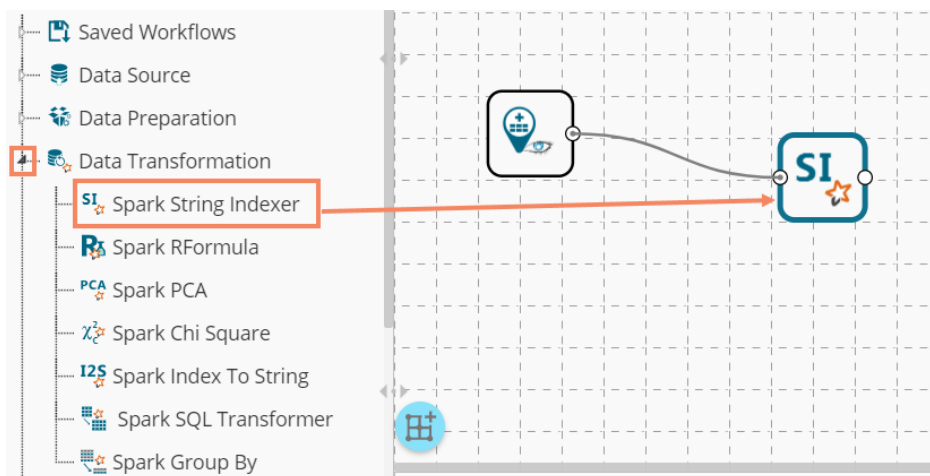
ii) Configure the required component fields for the String Indexer
a.   The Properties tab for Spark Indexer contains an option to select '**Label Column**' from previous component headers on which a new column was created
b.   Users can rename the created label column using the '**Label Column Name**'
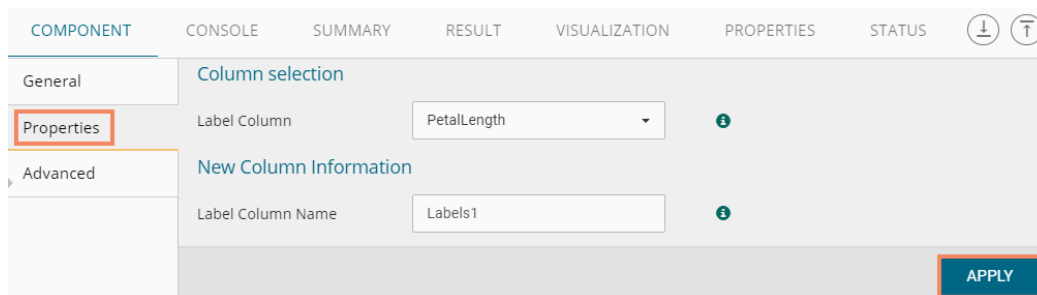
c.   The String Indexer, when applied on one dataset, will handle unseen labels using either of the methods provided under the '**Advanced**' tab:
d.   Users are provided with two options in the '**Advanced**' tab to manage the unseen labels

i.   Error: The unseen labels will be thrown as an exception (by default)
ii.   Skip: The rows containing the unobserved labels will be skipped

iii)   Click '**APPLY**'
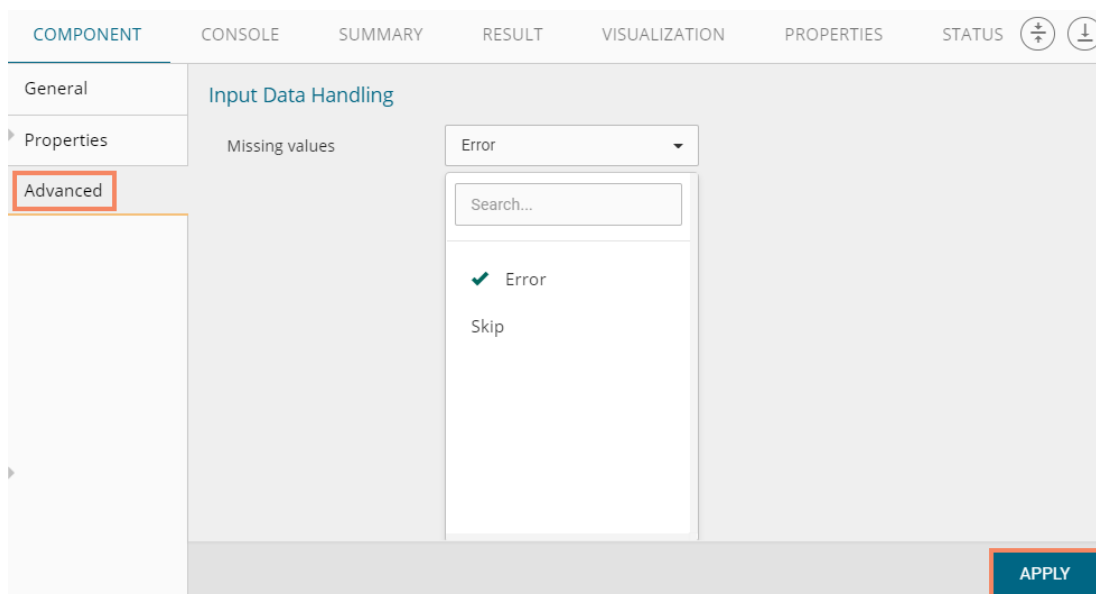
iv)   After getting the success message run the workflow
v)   A message will pop-up to confirm whether users want to enable logging
vi)   Click '**NO**'
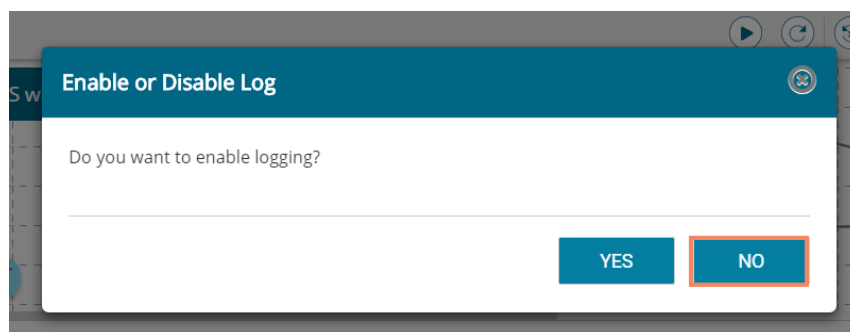
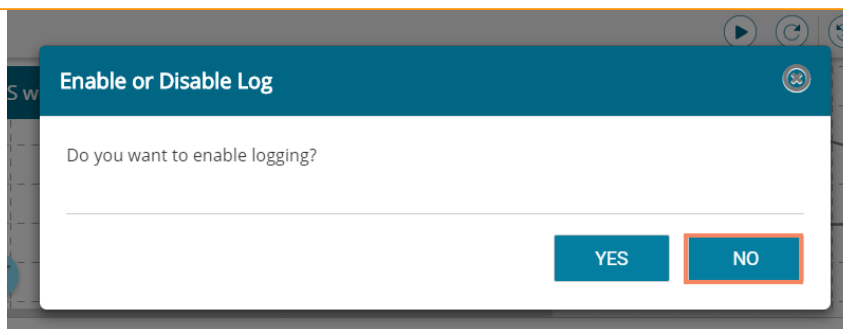vii)   Users will get the process status under the '**CONSOLE**' tab
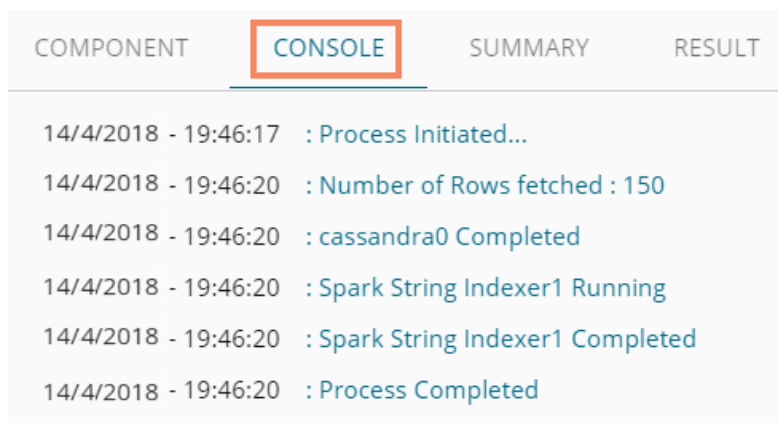
6.3.2. Spark R Formula

The Spark R Formula can be used to produce a vector column of features and a double column of labels. The Spark R Formula is a feature selector for the BDB Predictive Analysis which can be used to transform string columns to numerical columns. After selecting the desired features and labels from previous columns.

i)   Users need to select the Spark R Formula component and connect it to a configured data source.
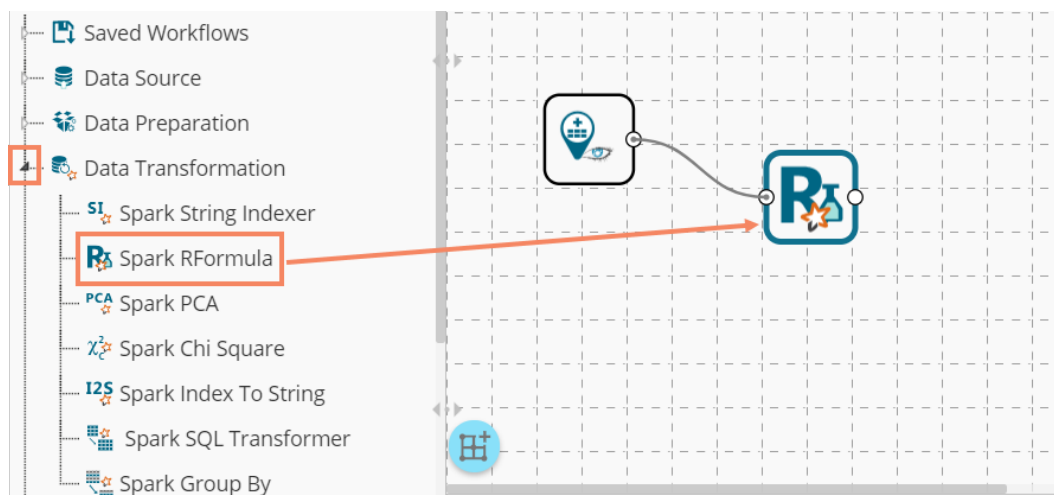
ii) Select the Spark R Formula and configure the following fields under the component tab:
a.   **Column Selection**: Select the desired Features and Labels from the column headers provided under the Properties tab
b.   **Enable Formula**: Enable this option to get a formula. (By enabling formula, the '**Apply**' option will change to 'Next' and the '**Formula**' option will be listed under the 'COMPONENT' tabs)
c.   **New Column Information**: Provide names for the newly created Feature and Label columns
iii) Click '**NEXT**'
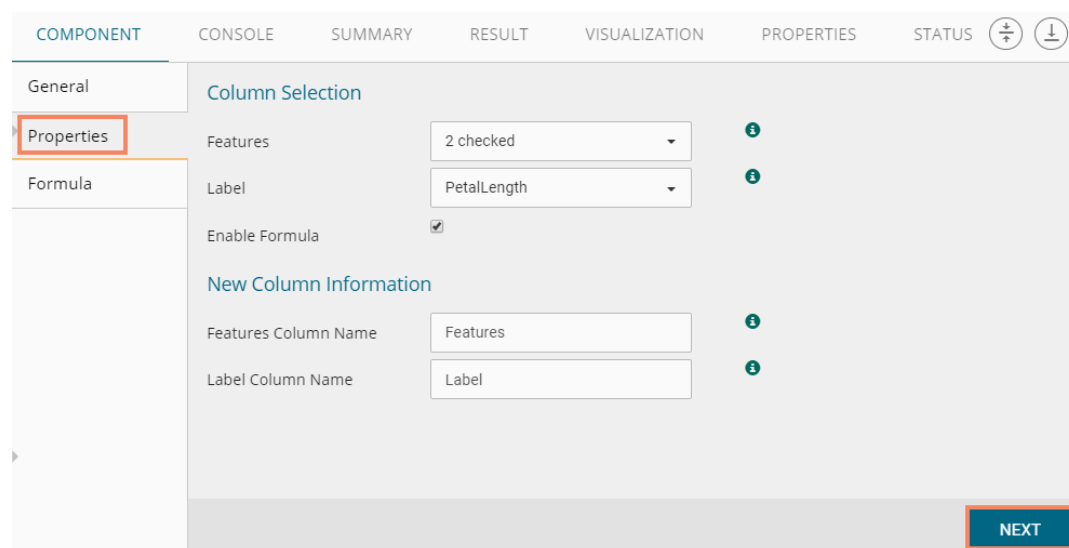
iv) Users will be directed to the next page to enter a formula
v)   Enter a formula in the given box by double clicks on the available values
vi) Click '**APPLY**'
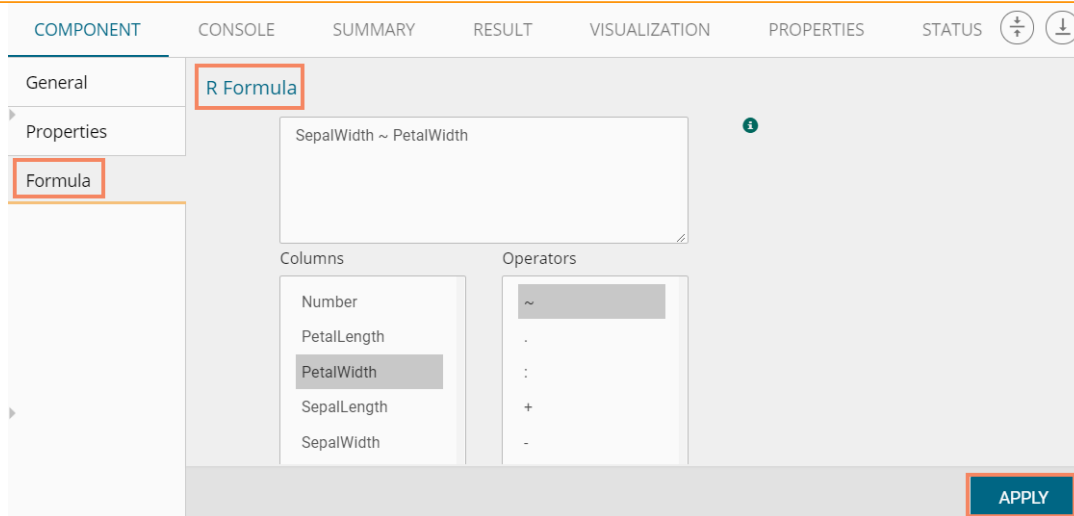
vii) After getting the success message run the workflow
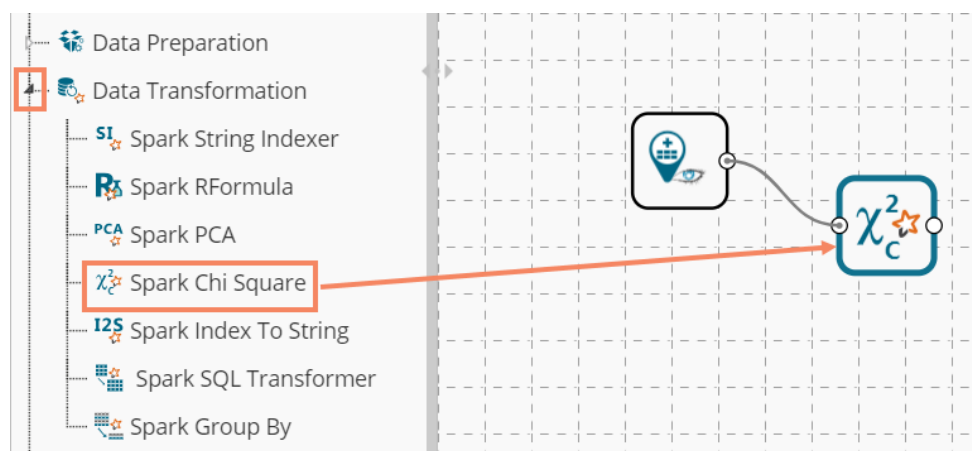   a. A message will pop-up to confirm whether users want to enable logging
   b. Click '**NO**'
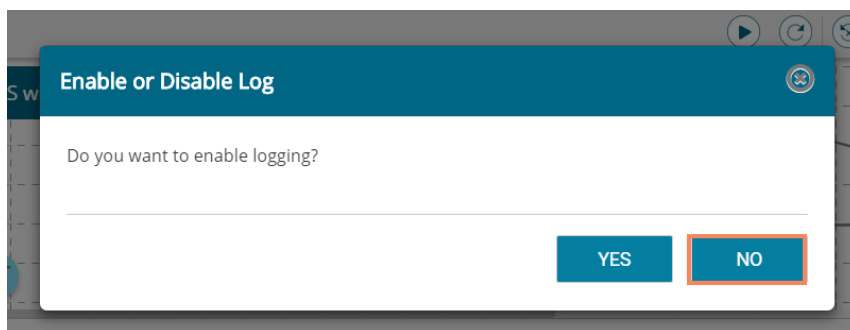
viii) Users will get the process status under the '**CONSOLE**' tab
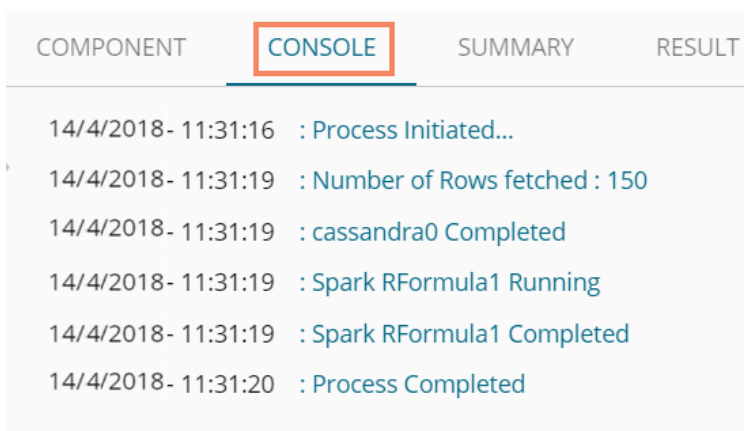
6.3.3. Spark PCA
The Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of correlated variables into a set of values of linearly uncorrelated variables called principal components (PCs). A PCA class trains a model to project vectors to a low-dimensional space using PCA.

The PCA transformation is defined in such a way that the first principal component has the most significant variance (it accounts for as much of the variability in the data as possible), and each

following component, in turn, has the highest difference possible under the constraint that it is orthogonal to the other components. The resulting vectors will be uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

i) Users need to select the Spark PCA component and connect it to a configured data source
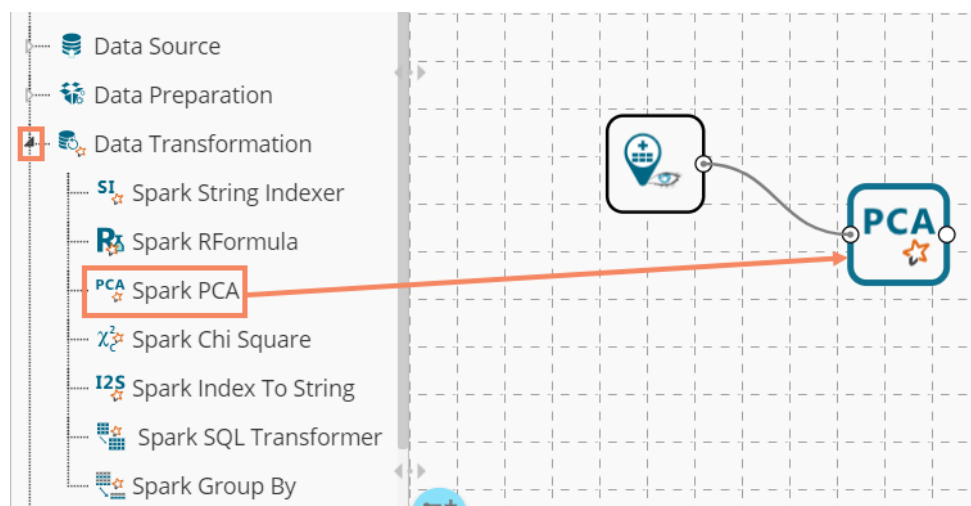
ii) Configure the following component fields for the Spark PCA:
   a. Input Column
      i. Features: Select the required elements from the drop-down menu
      ii. K Value: Enter the number of principal components
   b. Output Column
      i. Predicted Column Name: Enter column header for the predicted column
iii) Click '**APPLY**'

iv) After getting the success message run the workflow
   **a.** A message will pop-up to confirm whether users want to enable logging
   b. Click '**NO**'

Enable or Disable Log

Do you want to enable logging?

YES   NO

v)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY

14/4/2018 - 12:38:5 : Process Initiated...

14/4/2018 - 12:38:8 : Number of Rows fetched : 150

14/4/2018 - 12:38:8 : cassandra0 Completed

14/4/2018 - 12:38:8 : Spark PCA1 Running

14/4/2018 - 12:38:8 : Spark PCA1 Completed

14/4/2018 - 12:38:8 : Process Completed

6.3.4. Spark Chi-Square

In probability theory and statistics, the chi-squared distribution (also chi-square or x2-distribution) with K degrees of freedom is the distribution of a sum of the squares of k independent standard random variables. It is a unique case of the gamma distribution and is one of the most widely used probability distributions in inferential statistics. E. g. in hypothesis testing or the construction of confidence intervals. When it is being distinguished from the more general noncentral chi-squared distribution, this distribution is sometimes called the central chi-squared distribution.

i)   Users need to select the Spark Chi-Square component and connect it to a configured data source.

ii) Configure the following component fields for the Spark Chi-Square:
   a. Input Column
      i. Features: Select the required elements from the drop-down menu.
      ii. K Value: Enter the number of principal components.
   b. Output Column
      i. Predicted Column Name: Enter the column header for the predicted column.
iii) Click '**APPLY**'.
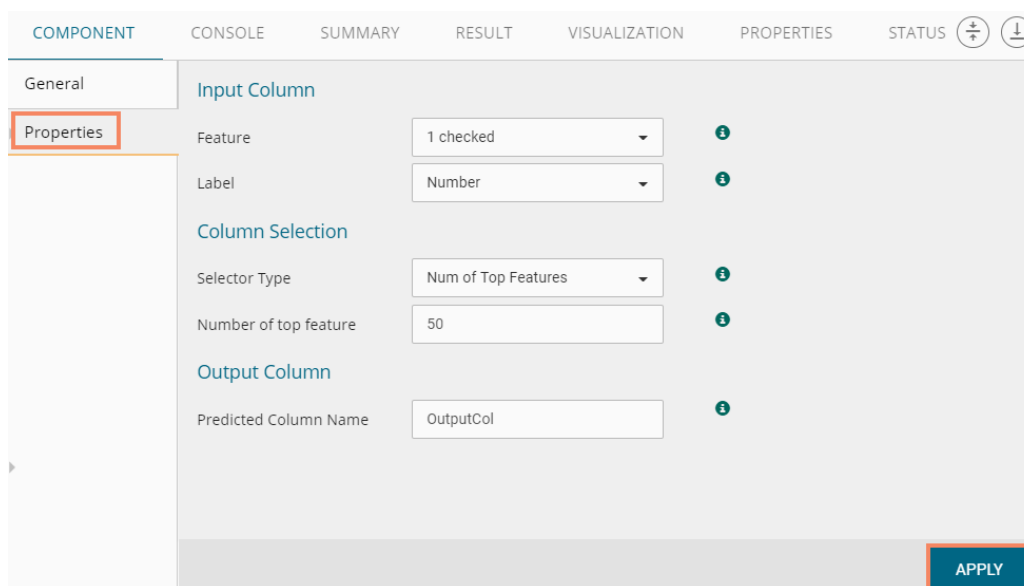
iv) After getting the success message run the workflow.
   a. A message will pop-up to confirm whether users want to enable logging.
   b. Click '**NO**'.
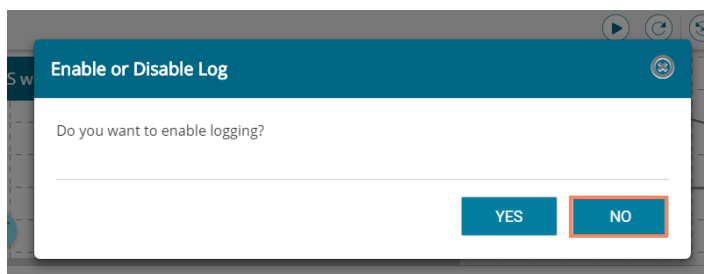
v) Users will get the process status under the '**CONSOLE**' tab.
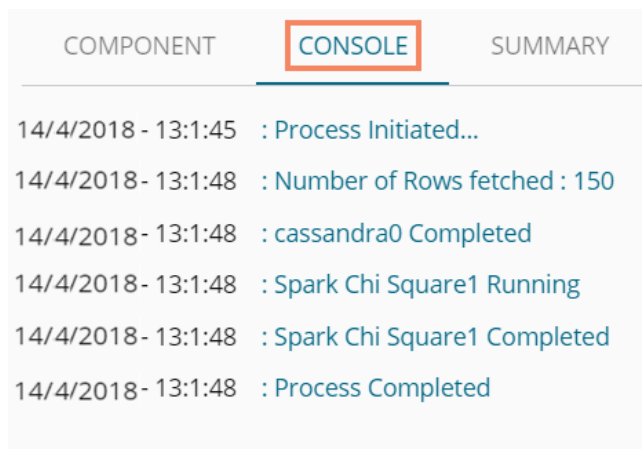
6.3.5. Spark Index to String

The Spark Index to String component can be used to convert index label column into String column so that it can be applied to specific algorithms that require index column as the Label Column. This component consists of an option to select a label column from previous component headers. After choosing a label, column user can change the column header of the newly Stringed column which will be called '**Label**' by default.

i) Users need to select and drag a configured data source on the workspace
ii) Connect the Spark Index to String component with the data source
iii) Connect a Spark Apply Model to the configured data source and Spark Index to String components
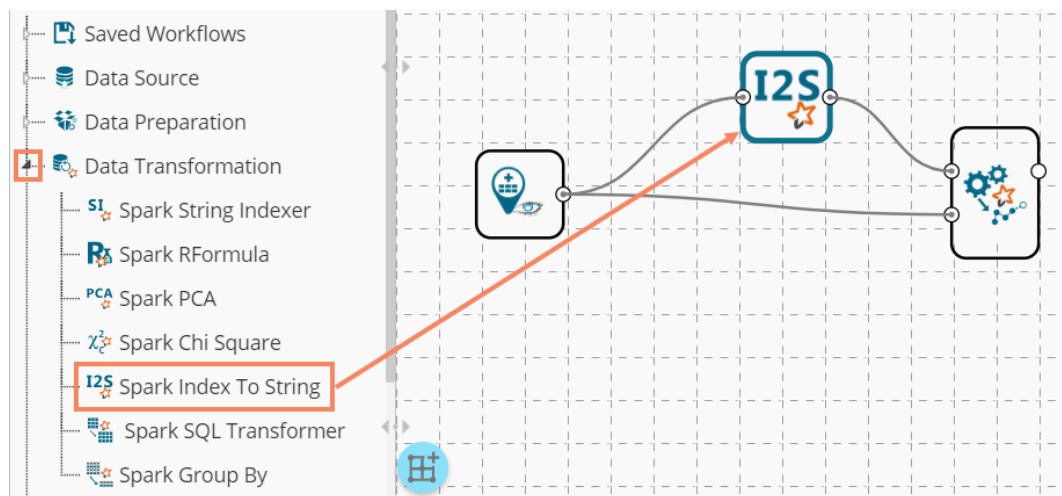
iv) Configure the following component fields for the '**Spark Index to String**' component:
   a. Column **Selection**
      i. Label Column: Select a column using the drop-down menu. Make sure that you select the same column that was selected while configuring the String Indexer component (In this case, it is 'PetalLength')
   b. **New Column Information**
      i. Label Column Name: By default, the column name appears as '**Labels**' user can change the column heard/name using this field.
      ii. Labels: Enter the labels separated by a comma
v) Click '**APPLY**'.

vi) Configure the '**Apply Model**' component.

vii) After getting the success message run the workflow.
   a. A message will pop-up to confirm whether users want to enable logging.
   b. Click '**No**'.

viii) Users will get the process status under the '**CONSOLE**' tab.

ix) Users can view the result with the Label column by clicking on the 'Spark Apply Model' component and then opening the '**RESULT**' tab.

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	Labels2
51	4.7	1.4	7	3.2	1	label2
46	1.4	0.3	4.8	3	0	label1
14	1.1	0.1	4.3	3	0	label1
31	1.6	0.2	4.8	3.1	0	label1
81	3.8	1.1	5.5	2.4	1	label2
90	4	1.3	5.5	2.5	1	label2
74	4.7	1.2	6.1	2.8	1	label2
10	1.5	0.1	4.9	3.1	0	label1
29	1.4	0.2	5.2	3.4	0	label1
55	4.6	1.5	6.5	2.8	1	label2

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

Note: Users can also use this component in a workflow where first the '**String Indexer**' component has been connected to the data source, and then the combination can be connected to the '**Index to String**' component as displayed below:
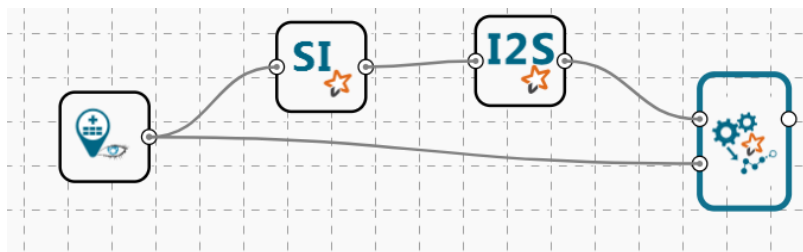
Users can configure all the components and get a result for the 'Spark Apply Model.'

6.3.6. Spark SQL Transformer

Spark SQL Transformer implements the transformations which are defined by an SQL statement. Currently, we only support SQL syntax. E.g., "SELECT ... FROM __THIS__ ..." where "__THIS__" stands for the underlying table of the input data set. The select clause specifies the fields, constants, and expressions to display in the output. Any clause supported by Spark SQL can be used. Users can also use Spark SQL built-in function and UDFs.

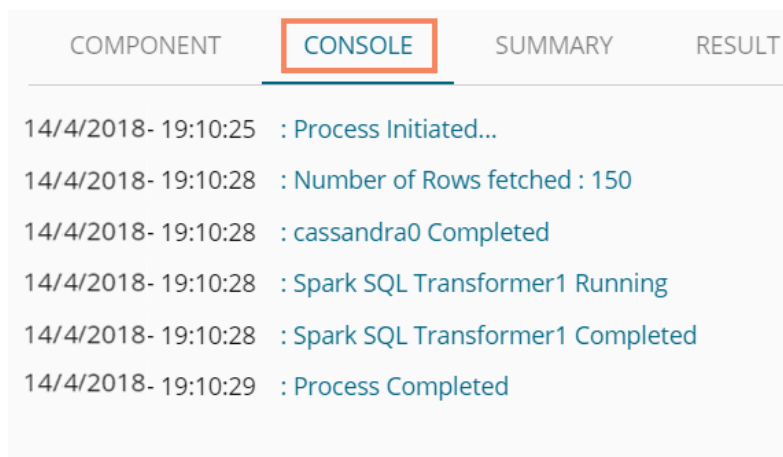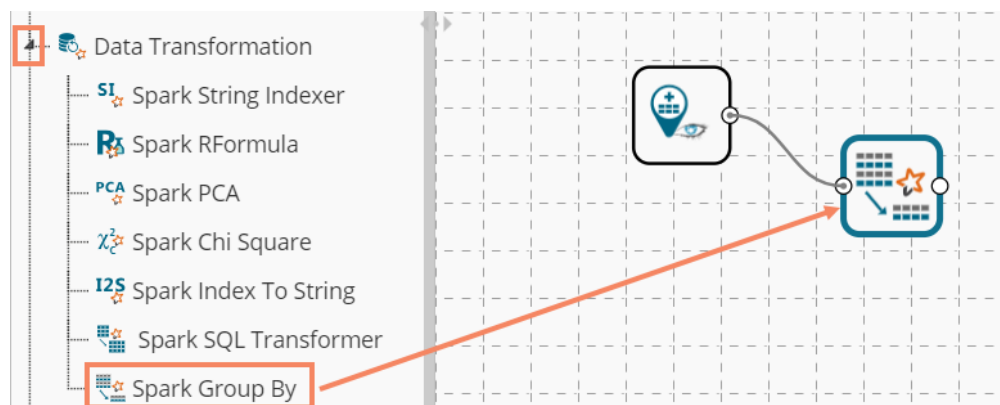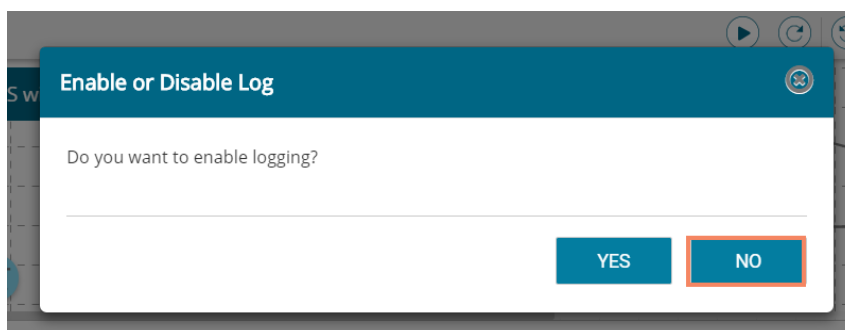i)   Select the Spark SQL Transformer component and connect it to a configured data source.

ii)   Configure the required component fields for the Spark SQL Transformer.
        a. SQL Statement: Provide an SQL statement.
        b. Fields: All the available fields under the selected data source will be listed.
iii)   Click '**APPLY**'
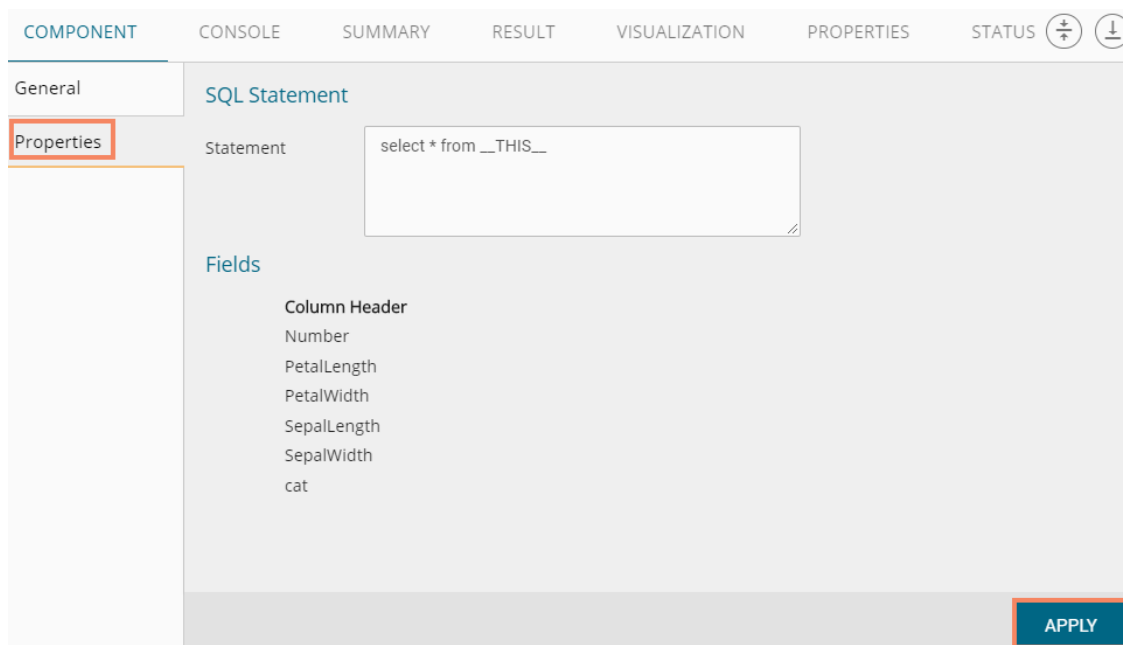
iv)   After getting the success message run the workflow
        a. A message will pop-up to confirm whether users want to enable logging
        b. Click '**NO**'
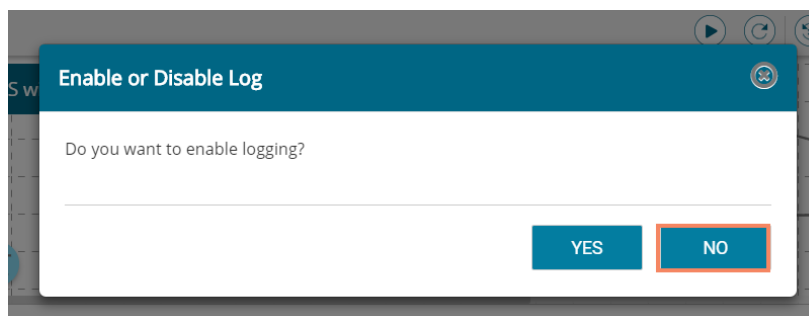
v)   Users will get the process status under the '**CONSOLE**' tab

6.3.7. Spark Group By

Spark Group By is a transformation operation. Users can apply '**Spark Group By**' transformation to the data frame of the last node output. The on top of which aggregation is done can be added to the output with the alias name.

i) Select the Spark Group By component and connect it to a configured data source

ii) Configure the required component fields for the Spark SQL Transformer
   a. **Aggregation Columns**
      i. Column Name: Select a Column from the drop-down menu
      ii. Alias Name: Enter an alias name for the selected column
      iii. Aggregation Type: Select an aggregation type from the drop-down menu
      iv. Click '**Add**' ✚ icon to add a new series to configure aggregation column
   b. Select the required column from the '**Group By Columns**' and move it to the '**Selected Columns**'
   c. Use '**Up**' and '**Down**' to change the order of the selected columns
iii) Click '**Apply**'
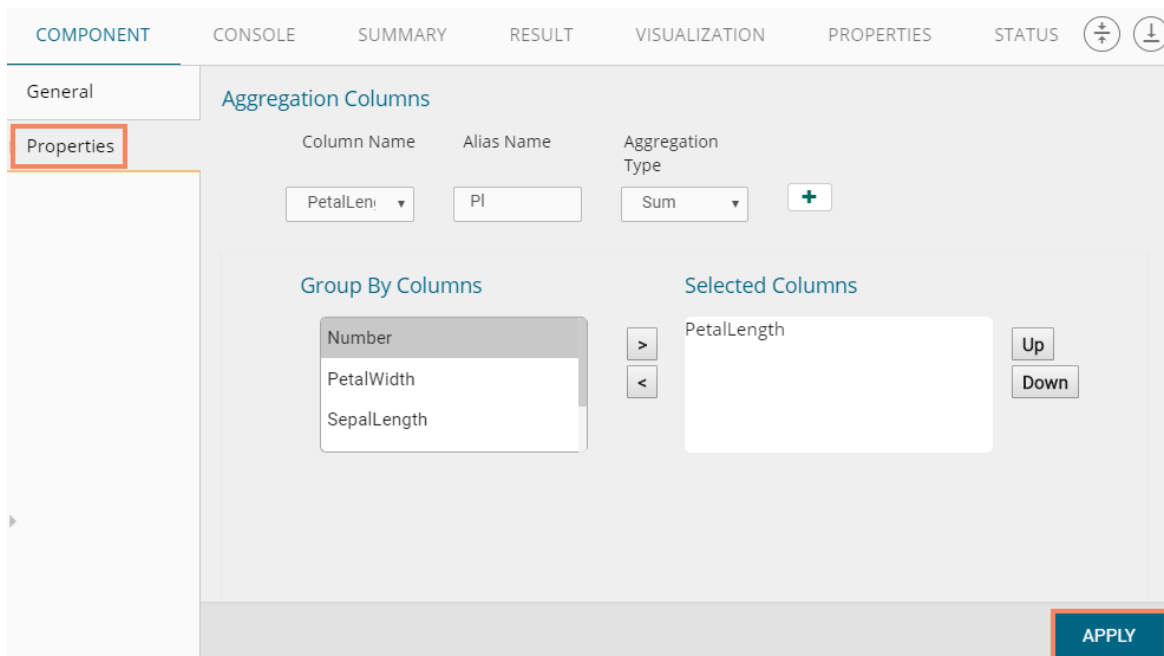
iv) After getting the success message run the workflow
   a. A message will pop-up to confirm whether users want to enable logging
   b. Click '**NO**'

Enable or Disable Log

Do you want to enable logging?

YES   NO

v)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT   CONSOLE   SUMMARY

14/4/2018 - 19:20:57   : Process Initiated...

14/4/2018 - 19:21:0   : Number of Rows fetched : 150

14/4/2018 - 19:21:0   : cassandra0 Completed

14/4/2018 - 19:21:0   : Spark Group By1 Running

14/4/2018 - 19:21:0   : Spark Group By1 Completed

14/4/2018 - 19:21:0   : Process Completed
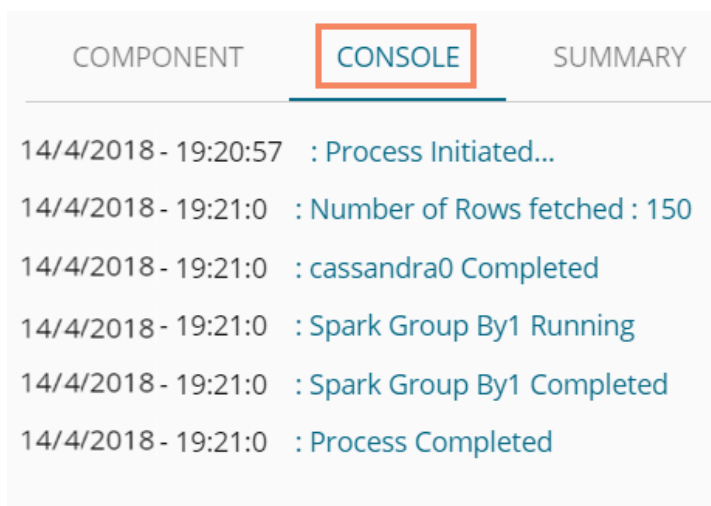
6.4.   Algorithms

6.4.1.   Clustering

6.4.1.1.   Spark-K- Means

The Spark K-Means algorithm is provided as an option under the clustering algorithm category. The spark.ml implementation includes a parallelized variant of the k-means++ method called k-means||.

Applying Spark-K-Means to a Data Source

i)   Drag the Spark-K-Means to the workspace and connect to a configured data source.
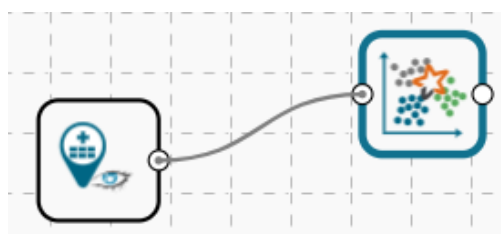
ii)   Configure the following fields in the '**Properties**' tab:
   a. **Output Information**
      i. **Number of Clusters**: Enter number of groups for clustering. The default value for this field is 5. Range should be between one and a total number of clusters.
   b. **Column Selections**

i. **Feature:** Select the input columns with which you want to perform the Analysis.
c. **New Column Information**
   i. **Cluster Name:** Enter a name for the new column displaying cluster number.


iii) Select the '**Advanced**' tab.
   a. Configure the following '**Behavior**' fields:
      i. **Maximum Iterations**: Enter the number of iterations allowed for discovering clusters (The default value for this field is 20).
      ii. **Initialization Mode**: Select any one option at the beginning of the algorithm out of '**Random**' or '**k-means||**' (default)
      iii. **Initialization Steps**: Set number for the initialization mode as random (The default value for this field is 5)
      iv. **Convergence Tolerance**: Set tolerance level to include clusters in exponential form (the default value for this field is 1.0e-4)
      v. **Initial Cluster Center Seed**: Enter a number indicating initial cluster center seed (The default value for this field is 10)
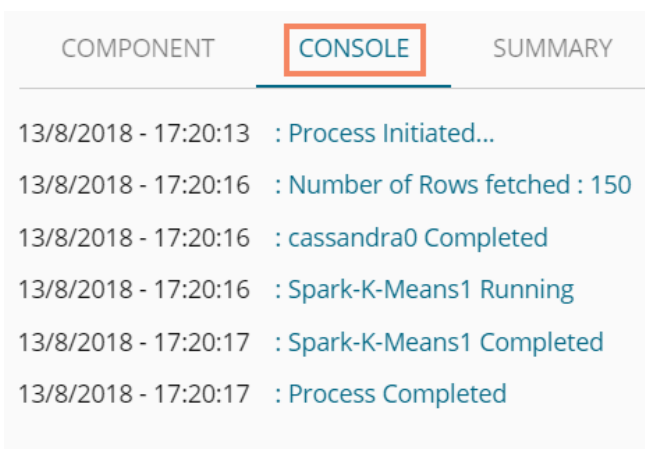iv) Click '**APPLY**'

v) After getting the success message run the workflow
vi) A message will pop-up to confirm, whether users want to enable logging or no

vii)   Click '**NO**'
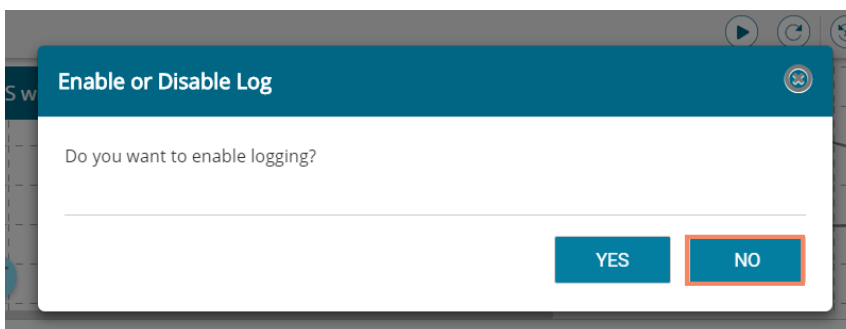
viii)   Users will get the process status under the '**CONSOLE**' tab

ix)   Follow the below given steps to display the result view:
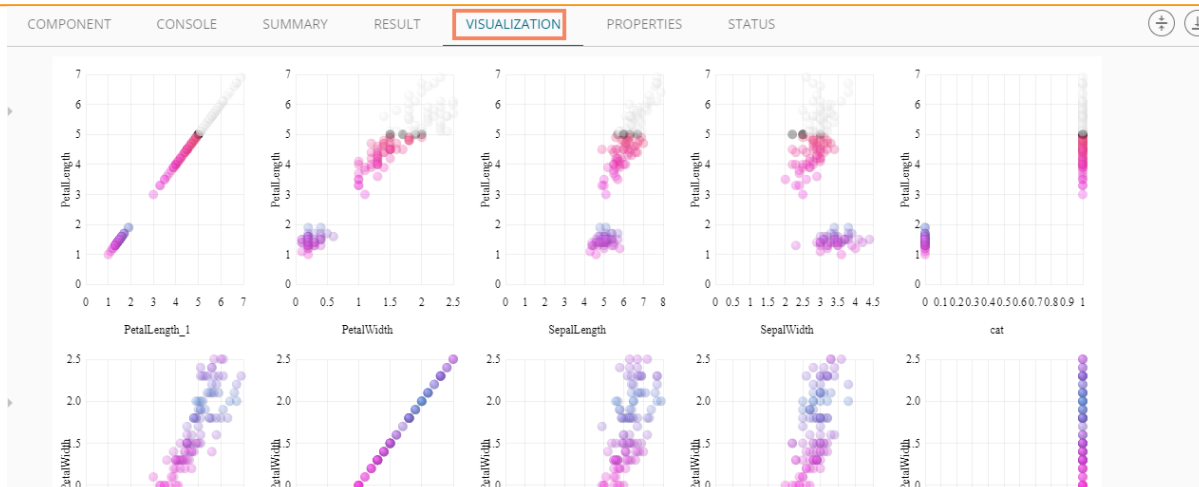   **a.** Click the dragged algorithm component onto the workspace
    b. Click the '**RESULT**' tab
x)   A new column '**ClusterNumber**' will be added to the displayed result data

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	featuresCol1	ClusterNumber
51	4.7	1.4	7	3.2	1	{"values":[4.7,1.4,7,3.2,1]}	3
46	1.4	0.3	4.8	3	0	{"values":[1.4,0.3,4.8,3,0]}	0
14	1.1	0.1	4.3	3	0	{"values":[1.1,0.1,4.3,3,0]}	0
31	1.6	0.2	4.8	3.1	0	{"values":[1.6,0.2,4.8,3.1,0]}	0
81	3.8	1.1	5.5	2.4	1	{"values":[3.8,1.1,5.5,2.4,1]}	4
90	4	1.3	5.5	2.5	1	{"values":[4,1.3,5.5,2.5,1]}	4
74	4.7	1.2	6.1	2.8	1	{"values":[4.7,1.2,6.1,2.8,1]}	3
10	1.5	0.1	4.9	3.1	0	{"values":[1.5,0.1,4.9,3.1,0]}	0
29	1.4	0.2	5.2	3.4	0	{"values":[1.4,0.2,5.2,3.4,0]}	0
55	4.6	1.5	6.5	2.8	1	{"values":[4.6,1.5,6.5,2.8,1]}	3

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

xi)   Click the '**VISUALIZATION**' tab
xii)   The result data will be displayed via the Scatter Plot Matrix Chart

Note: Users can click the 'SUMMARY' tab to display a summary of the model. E.g. The following image is a sample to demonstrate how summary can be shown for the Spark-K-Means algorithm component.

------------ Summary of the model ------------

Columns used in the algorithm:
PetalLength (double)
PetalWidth (double)
SepalLength (double)
SepalWidth (double)
cat (double)
Cluster Centers = [1.4333333333333331,0.23030303030303031,4.818181818181818,3.2363636363636363,0.0],
[5.846875,2.1312499999999996,6.91249999999999,3.0999999999999996,1.0],
[1.5176470588235293,0.2764705882352942,5.370588235294117,3.8,0.0],
[4.807317073170733,1.6219512195121952,6.236585365853658,2.858536585365854,1.0],
[3.94074074074074,1.2185185185185183,5.529629629629628,2.622222222222222,1.0]
Within Set Sum of Squared Errors = 50.1640823994975

------------ End of Summary ----------

6.4.1.2. Spark K-Means Connected to the Pipeline Components

i) Connect a combination of the data source and Spark K-Means algorithm component to a pipeline component as shown in the following image:
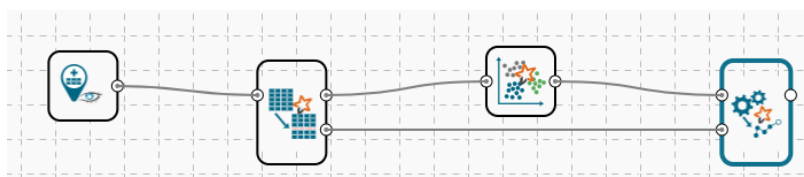

ii) Configure the required component fields and run the workflow
iii) Users will get the process status under the 'CONSOLE' tab

COMPONENT   **CONSOLE**   SUMMARY

14/4/2018- 8:0:35   : Process Initiated...

14/4/2018- 8:0:37   : Process started

14/4/2018- 8:0:37   : cassandra0 Running

14/4/2018- 8:0:41   : Number of Rows fetched : 150

14/4/2018- 8:0:41   : cassandra0 Completed

14/4/2018- 8:0:41   : Spark Split Data2 Running

14/4/2018- 8:0:41   : Spark Split Data2 Completed

14/4/2018- 8:0:41   : Spark-K-Means1 Running

14/4/2018- 8:0:43   : Spark-K-Means1 Completed

14/4/2018- 8:0:43   : Spark Apply Model3 Running

14/4/2018- 8:0:43   : Spark Apply Model3 Completed

14/4/2018- 8:0:43   : Process Completed

iv)   Follow the below given steps to display the result view:
      a. Click the data preparation component onto the workspace
      b. Click the '**RESULT**' tab

COMPONENT   CONSOLE   SUMMARY   **RESULT**   VISUALIZATION   PROPERTIES   STATUS

Show [10 ▾] entries   Search: [ ]

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	featuresCol1	ClusterNumber
31	1.6	0.2	4.8	3.1	0	{"values":[1.6,0.2,4.8,3.1,0]}	1
10	1.5	0.1	4.9	3.1	0	{"values":[1.5,0.1,4.9,3.1,0]}	1
29	1.4	0.2	5.2	3.4	0	{"values":[1.4,0.2,5.2,3.4,0]}	1
81	3.8	1.1	5.5	2.4	1	{"values":[3.8,1.1,5.5,2.4,1]}	3
79	4.5	1.5	6	2.9	1	{"values":[4.5,1.5,6,2.9,1]}	0
76	4.4	1.4	6.6	3	1	{"values":[4.4,1.4,6.6,3,1]}	0
96	4.2	1.2	5.7	3	1	{"values":[4.2,1.2,5.7,3,1]}	3
91	4.4	1.2	5.5	2.6	1	{"values":[4.4,1.2,5.5,2.6,1]}	3
143	5.1	1.9	5.8	2.7	1	{"values":[5.1,1.9,5.8,2.7,1]}	0
18	1.4	0.3	5.1	3.5	0	{"values":[1.4,0.3,5.1,3.5,0]}	1

Showing 1 to 10 of 45 entries   Previous   1   2   3   4   5   Next

v)   Click the '**VISUALIZATION**' tab
vi)   The result data will be displayed via the Scatter Plot Matrix Chart
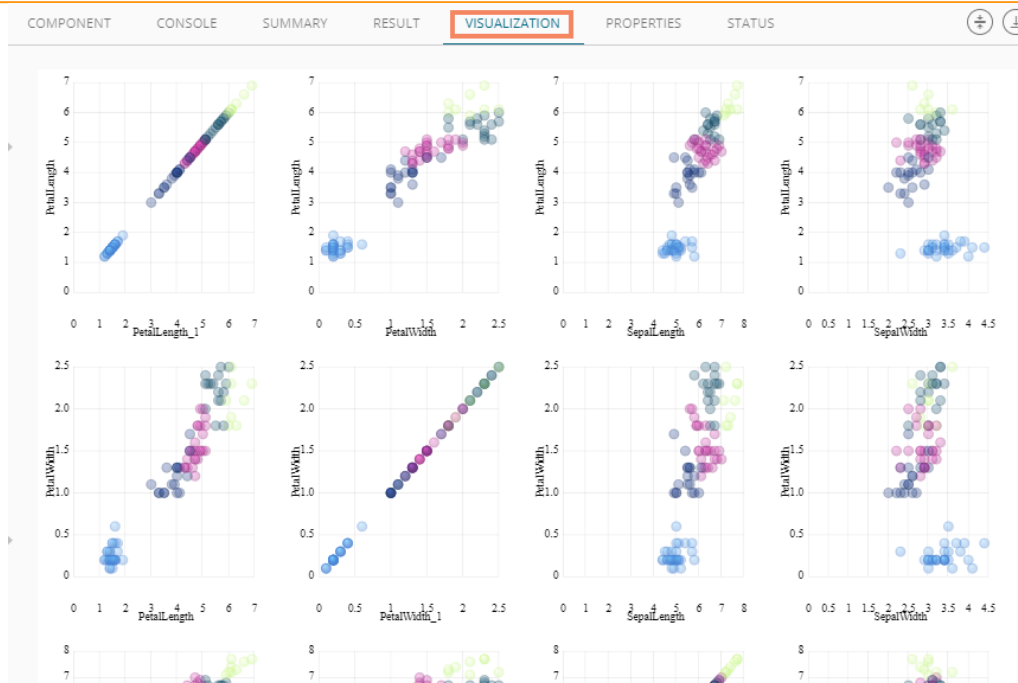
6.4.2. Classification

6.4.2.1. Spark-Naive Bayes

The Naive Bayes is a simple multiclass classification algorithm with an assumption of independence between every pair of features. This algorithm can be trained to be very efficient. The user can set a threshold for each class. The algorithm will then classify values as per the set thresholds.
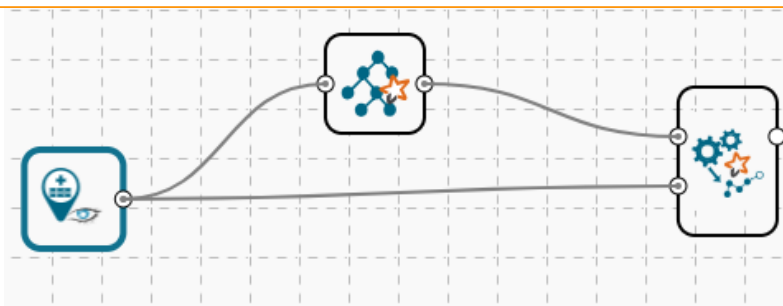
Spark Naive Bayes consists of two types of model selection methods:
1. Multinomial- If the data set is numerical
2. Bernoulli- If the dataset contains 0 and 1

i)   Drag the Spark Naive Bayes component to the workspace and connect it with a configured data source
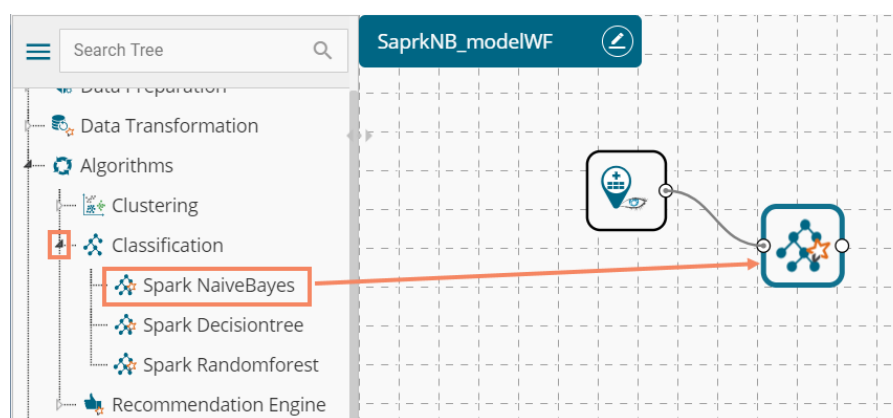
ii)   Connect and configure the Spark Apply Model component to the combination of a data sources and Spark Naive Bayes component (to display the results)

iii) Configure the following fields in the '**Properties**' tab:
   a. **Feature**: Select column(s) from the drop-down menu
   b. **Label**: Select column(s) from the drop-down menu
   c. **Enable Validation**: Put a check mark in the box to enable the validation (It is an optional field)

By enabling '**Validation**' via the '**Properties**' tab, Users will be redirected to the '**Validation**' tab.

There are two types of validation methods:
   a. **Train Validation** – Train validation begins by splitting a data set into two parts, as training and testing datasets as per the training ratio. It also iterates through paramMapS. For each combination of parameters, the algorithm will iterate over it and select based on the evaluation metric.
   b. **Cross-Validation** – Cross validation begins by splitting the data set into a set of folds which are used as a separate training and test datasets,e.g., with k=3 folds, Cross Validator will generate 3 (training, testing) dataset pairs, each of which uses 2/3 of the data for training and 1/3 for testing. It also iterates through paramMapS. The algorithm will iterate over each combination of parameters and folds to decide the best model using an average of the k folds.

iv) Configure the following '**Validation**' information:
   a. **Model Selection Method**: Select any one validation method using the drop-down menu:
       i. Train Validation
       ii. Cross-Validation
   b. **Evaluator**: Select any one option using the drop-down menu to define evaluator. Evaluator consist of two types:
       i. Multi-Class Classification – If the data set has multiple classes in the label column
       ii. Binary Class Classification- if the data set has two classes in the label column
   c. **Train Ratio**: This field will be displayed if train validation has been selected by using the '**Model Selection Method**' field

OR

If '**Cross Validation**' is enabled, users will be provided with a field '**Number of folds**' from the input data to be taken as training data for the cross-validation. (Spark Naive Bayes supports only string data when cross-validation is selected)

- **Advanced Tab when 'Validation' is Disabled**

  a. **Input Data Handling**

     i. **Model Type**: Select an option from the drop-down list. The Spark Naive Bayes consists of two types of model selection methods:
        1. **Multinomial**- If the data set is numerical
        2. **Bernoulli**- If the dataset contains 0 and 1
     ii. **Thresholds**: Enter multiple values separated by a comma. Many values entered as threshold should be the same as that of many classes in labels. Sum of values must be equal to 1. Enter at least two commas separated values in this field.
     iii. **Additive Smoothening**: Enter values between 0 and 1 where 1.0 is the default value.
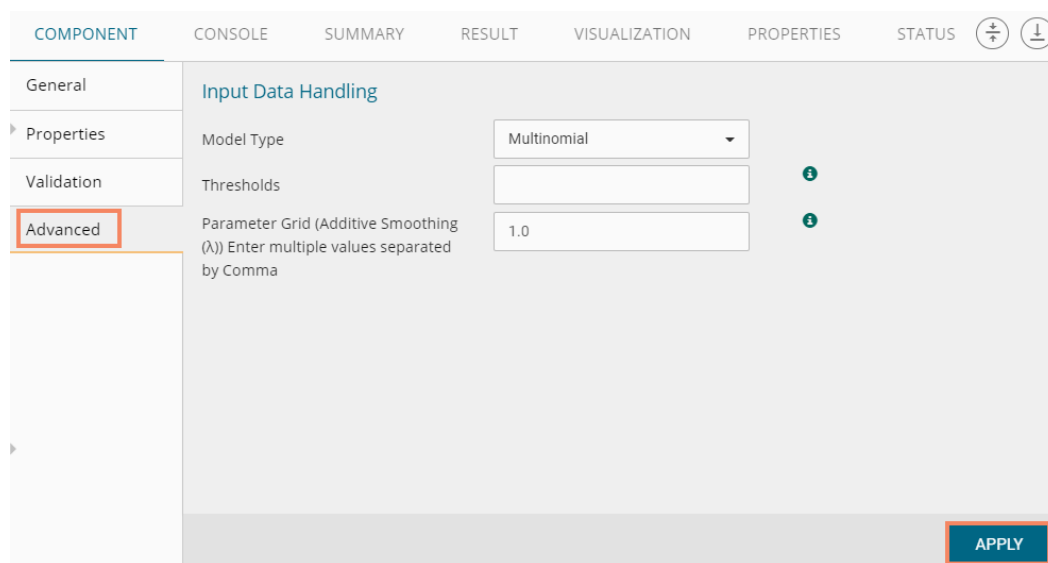

- **Advanced Tab when 'Validation' is Enabled**

  i) Click '**Next**' (By enabling '**Validation**' the '**Apply**' option changes into '**Next**')
  ii) Configure the following '**Advanced**' information:
      a. **Model Type**: Select an option from the drop-down list.
   The Spark Naive Bayes consists of two types of model selection methods:
         i. **Multinomial**- If the data set is numerical

ii. **Bernoulli**- If the dataset contains 0 and 1

b. **Thresholds**: Enter multiple values separated by a comma. The number of values entered as **the** threshold should be the same as that of many classes in labels. Sum of values must be equal to 1. Enter at least two commas separated values in this field.

c. **Parameter Grid**: Enter a valid double value between 0 and 1 (1 included). Users can enter single, or comma separated valid double value.

iii) Click '**APPLY**'

Note: If validation is enabled, users can enter multiple commas separated values in the Parameter Grid in the Advanced tab and they will be taken as paraMapS.

iv) Configure the '**Apply Model**' component and click '**APPLY**' option

v) After getting the success message run the workflow

a. A message will pop-up to confirm whether users want to enable logging

b. Click '**NO**'
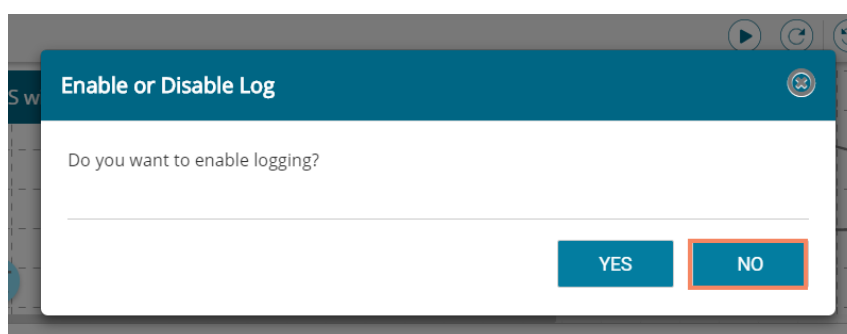
vi) Users will get the process status under the '**CONSOLE**' tab

14/4/2018- 20:22:45   : Process Initiated...

14/4/2018- 20:22:48   : Process started

14/4/2018- 20:22:48   : cassandra0 Running

14/4/2018- 20:22:49   : Number of Rows fetched : 150

14/4/2018- 20:22:49   : cassandra0 Completed

14/4/2018- 20:22:49   : Spark-NaiveBayes1 Running

14/4/2018- 20:22:49   : Spark-NaiveBayes1 Completed

14/4/2018- 20:22:49   : Spark Apply Model2 Running

14/4/2018- 20:22:49   : Spark Apply Model2 Completed

14/4/2018- 20:22:49   : Process Completed

    vii)   Follow the below given steps to display the result view:
           a. Click the dragged Apply Model component onto the workspace
           b. Click the '**RESULT**' tab

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES   STATUS

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	featuresCol1	rawPrediction1	probability1	cat	prediction1
51	4.7	1.4	7	3.2	{"values": [51,4.7,1.4,7,3.2]}	{"values": [-61.30794547293571,-60.9005634001106]}	{"values": [0.39954001678595313,0.6004599832140468]}	1	1
46	1.4	0.3	4.8	3	{"values": [46,1.4,0.3,4.8,3]}	{"values": [-38.945848552489046,-37.91339302669247]}	{"values": [0.26260832629712605,0.737391673702874]}	0	1
14	1.1	0.1	4.3	3	{"values": [14,1.1,0.1,4.3,3]}	{"values": [-25.250351967460713,-29.99783010844071]}	{"values": [0.991401042319541,0.008598957680455972]}	0	0
31	1.6	0.2	4.8	3.1	{"values": [31,1.6,0.2,4.8,3.1]}	{"values": [-34.27326986741876,-36.3081058929029]}	{"values": [0.8844063929834641,0.11559360701653579]}	0	0
81	3.8	1.1	5.5	2.4	{"values": [81,3.8,1.1,5.5,2.4]}	{"values": [-62.265670841364695,-53.81456824129352]}	{"values": [0.00021361905587770789,0.9997863809441222]}	1	1
90	4	1.3	5.5	2.5	{"values": [90,4,1.3,5.5,2.5]}	{"values": [-67.1480040132693,-56.97430031529983]}	{"values": [0.00003815926937339714,0.9999618407306265]}	1	1
74	4.7	1.2	6.1	2.8	{"values": [74,4.7,1.2,6.1,2.8]}	{"values": [-65.37475592956791,-59.29643294217229]}	{"values": [0.0022867758439047434,0.9977132241560953]}	1	1
10	1.5	0.1	4.9	3.1	{"values": [10,1.5,0.1,4.9,3.1]}	{"values": [-26.586943935028177,-32.802301371043754]}	{"values": [0.9980054841574933,0.0019945158425065676]}	0	0
29	1.4	0.2	5.2	3.4	{"values": [29,1.4,0.2,5.2,3.4]}	{"values": [-34.44919550644582,-37.661314914716534]}	{"values": [0.9612878134066031,0.0387121865933396924]}	0	0
55	4.6	1.5	6.5	2.8	{"values": [55,4.6,1.5,6.5,2.8]}	{"values": [-60.91090279982933,-58.6512196488879]}	{"values": [0.09451748265350766,0.9054825173464923]}	1	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note:
    a. Users can get a graphical display of their result data by first clicking the Algorithm component and then clicking the '**Apply Model**' component

b. Users can click the '**SUMMARY**' tab to view the model summary after connecting to a Spark Apply Model component. The Summary will be displayed if the '**Apply Model**' component contains a summary to show.

6.4.2.2. Spark Decision Tree

Decision Trees and their ensembles are popular methods for the machine learning tasks such as Classification and Regression. Decision trees are widely used since they are easy to interpret and do not require feature scaling. They can handle categorical features and extend to the multiclass classification setting. The Decision tree is an acquisitive algorithm that performs a recursive binary partitioning of the feature space and capture non-linearities and feature interactions. The tree predicts the same label for each bottom-most (leaf) partition. Each partition is chosen avidly by selecting the best split from a set of possible splits, to maximize the information gain at a tree node.

BizViz Predictive Analysis provides Spark Decision Tree under the Classification algorithm in the tree-node menu.

6.4.2.2.1. Classification as the Algorithm Type

i)   Drag the Spark Decision Tree component to the workspace and connect to a configured data source to create a basic workflow.
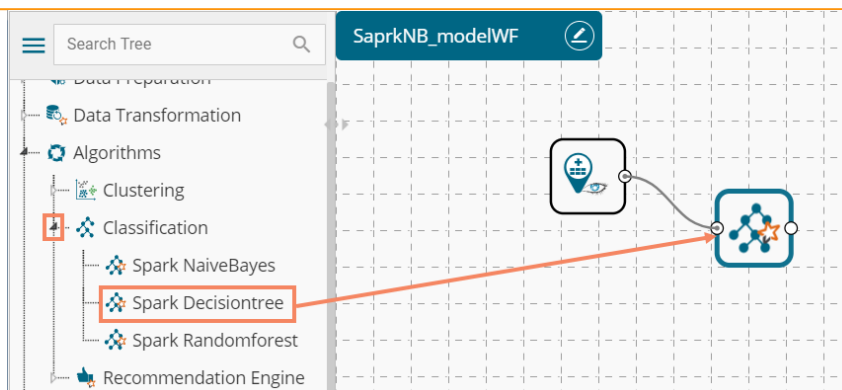
ii) Configure the required fields for the algorithm component:
- **Properties**
  - **a. Column Selection**
    - i. **Feature:** Select column(s) from the drop-down menu
    - ii. **Label:** Select column(s) from the drop-down menu
    - iii. **Algorithm Type:** Select an algorithm type from the drop-down menu
      1. **Classification**: Select this option if users want to pass dependent column as the categorical values (Default option).
      2. **Regression:** Select this option if users want to pass dependent column as numerical values.
    - iv. **Seeds:** Enter a numerical value to randomize the data.
    - v. **Enable Validation:** Put a check mark in the box to enable the validation (It is an optional field).

iii) Click **'NEXT'** (The **'APPLY'** option turns into **'NEXT'** if **'Validation'** has been enabled)
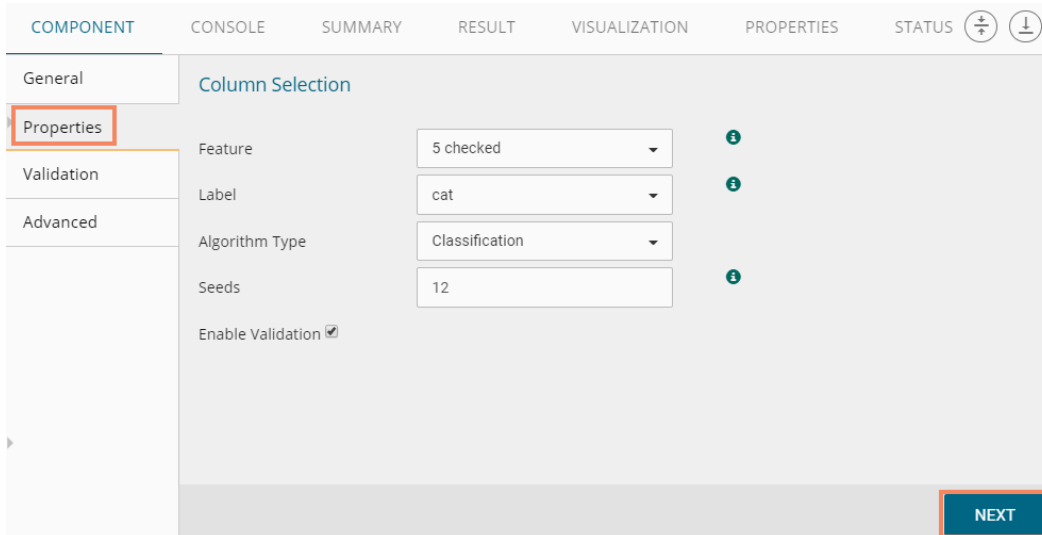
- **Validation**
  - **a. Model Selection**
    - i. **Model Selection Method**: Select any one validation method using the drop-down menu:
      1. **Train Validation**: By selecting this method, the **'Train Ratio'** field will be displayed to configure.
      2. **Cross-Validation**: By selecting this method, the **'Number of folds'** field will be displayed to configure.
    - ii. **Evaluator**: Select any one option using the drop-down menu to define **the** evaluator
      Evaluator consist of three types:

1. **Multi-Class Classification** – If the dataset has multiple classes in the label column
2. **Binary Class Classification**- if the data set has two classes in label Column
3. **Regression Class Classification**-if the '**Label**' column is continuous.
iii. **Train Ratio**: This field will be displayed if train validation has been selected via the '**Model Selection Method**' field.

iv) Click '**NEXT**' (The '**APPLY**' option turns into '**NEXT**' when '**Validation**' is enabled).

- **Advanced**

  a. **Column Selection**
  i. **Maximum Depth**: Maximum depth of the tree. (>= 0) E.g., depth 0 means one leaf node; depth 1 means 1 internal node + 2 leaf nodes. (Type integer only. Default value 5.)
  ii. **Maximum Bins:** Maximum number of bins for discretizing continuous features. (The value must be >=2 and >=number of categories for any categorical feature. (Type integer only. Default value 32.)
  iii. **Minimum Instances Per Node:** Minimum number of instances each child must have after the split. If a split causes the left or right child to have fewer than Min. Instances Per Node, the split will be discarded as invalid (The value should be >=1). (Type integer only. Default value 1.)
  iv. **Minimum Info Gain:** Enter Minimum Info Gain for a split to be considered at a tree-node (Type double only. Default value 0.0).
  v. **Thresholds:** Thresholds in multiclass classification to adjust the probability of predicting each class. The array must have a length equal to the number of classes, with values >=0. This class with the largest value p/t is predicted, where '**p**' is the optional probability of that class and '**t**' is the class' threshold. (Type: Comma separated double value. Thresholds will be displayed only in case of the Classification algorithm type.)
  vi. **Impurity:** Select an option from the drop-down menu. The '*impurity*' field is a measure of the homogeneity of the labels at the node. The current implementation of the algorithm provides two impurity measures for classification:
  1. **Gini**
  2. **Entropy**

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES   STATUS

General

Properties

Validation

Advanced

Column Selection

Maximum Depth   5

Maximum Bins   32

Minimum Instances Per   1
Node

Minimum Info Gain   0.0

Thresholds

Impurity   gini
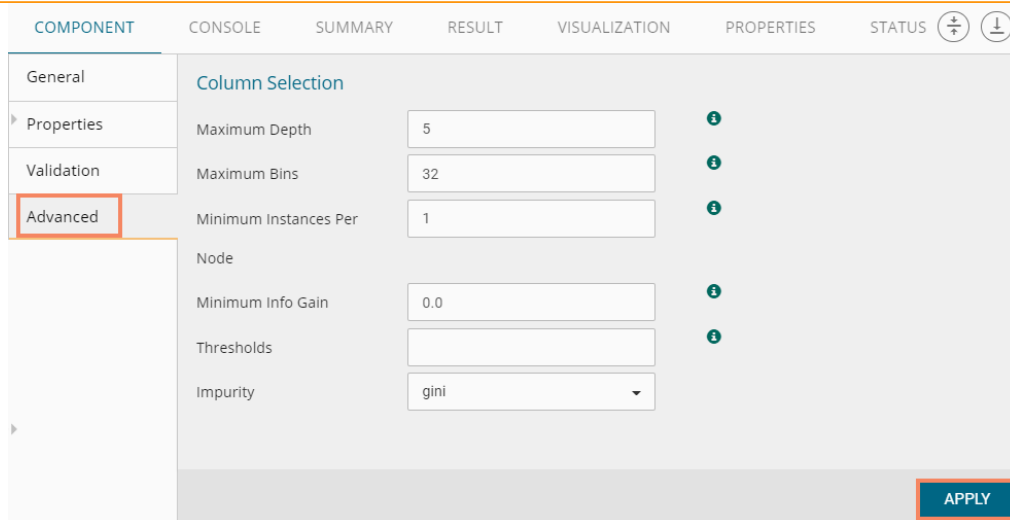
APPLY

v)   Connect the '**Spark Apply Model**' component to the workflow and configure it using the '**APPLY**' button
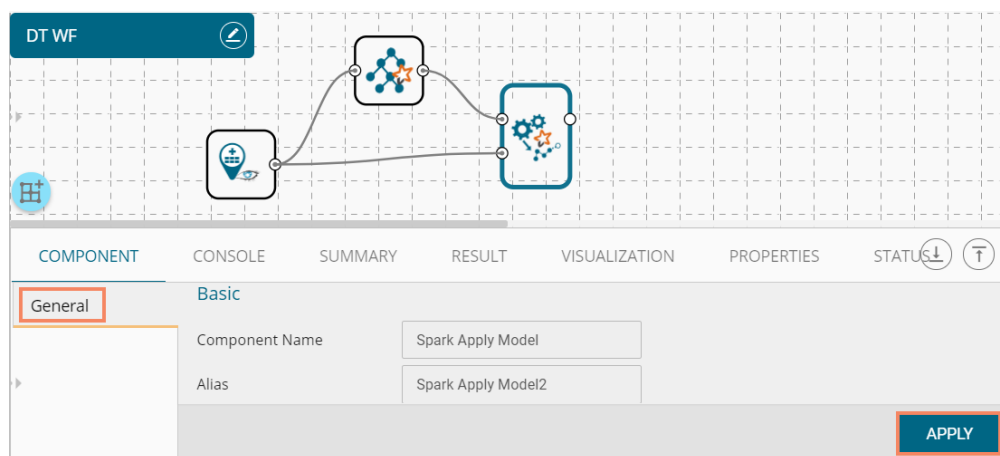
DT WF

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES   STATUS

General

Basic

Component Name   Spark Apply Model

Alias   Spark Apply Model2

APPLY

vi)   After getting the success message run the workflow
 a.   A message will pop-up to confirm whether users want to enable logging
 b.   Click '**NO**'

Enable or Disable Log

Do you want to enable logging?

YES   NO
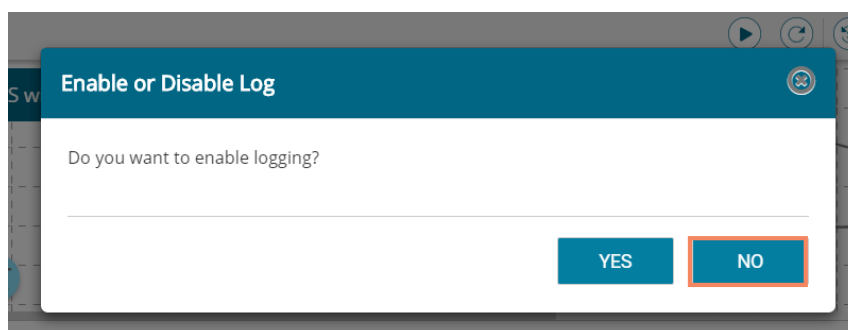
Note: The '**Advanced**' tab fields remain the same if '**Validation**' is disabled.

vii)   Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT |

14/4/2018 - 16:32:39   : Process Initiated...

14/4/2018 - 16:32:42   : Number of Rows fetched : 150

14/4/2018 - 16:32:42   : cassandra0 Completed

14/4/2018 - 16:32:42   : Spark-Decision-Tree1 Running

14/4/2018 - 16:32:43   : Spark-Decision-Tree1 Completed

14/4/2018 - 16:32:43   : Spark Apply Model2 Running

14/4/2018 - 16:32:43   : Spark Apply Model2 Completed

14/4/2018 - 16:32:43   : Process Completed

viii)   Users need to connect the 'Apply Model' component to the workflow and rerun it to view the result data.
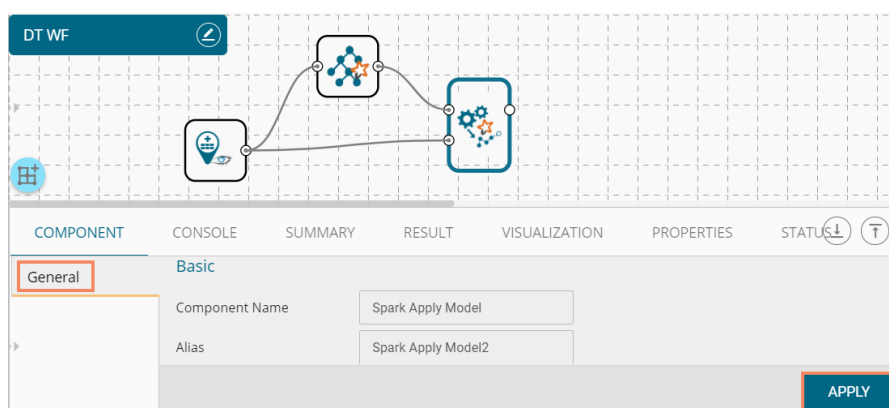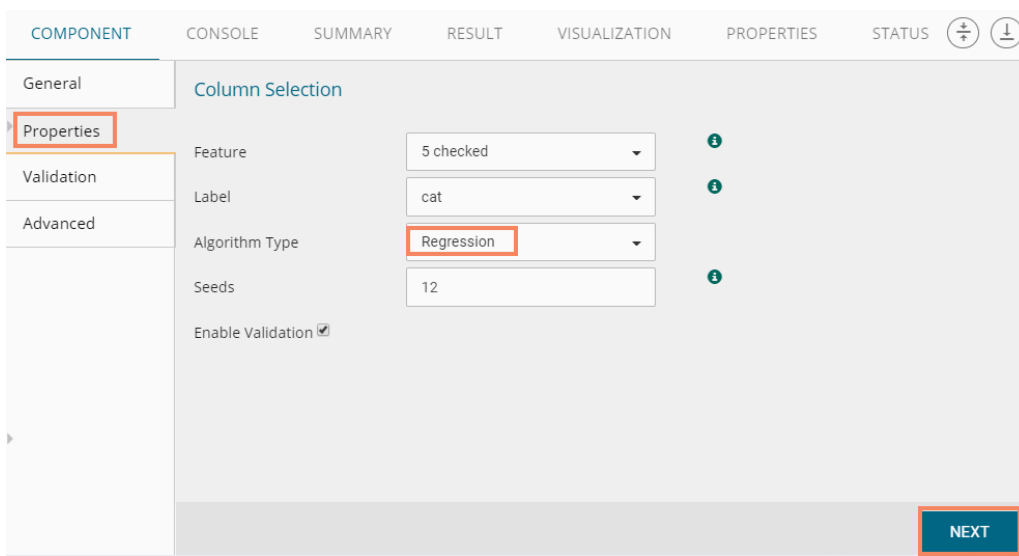
ix)   Follow the below given steps to display the result view:
   a.   Click the '**Spark Apply Model**' component onto the workspace.
   b.   Click the '**RESULT**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

Show 10 entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	dfFeaturesCol1	rawPrediction1	probability1	cat	prediction1
83	3.9	1.2	5.8	2.7	{"values":[83,3.9,1.2,5.8,2.7]}	{"values":[0,100]}	{"values":[0,1]}	1	1
111	5.1	2	6.5	3.2	{"values":[111,5.1,2,6.5,3.2]}	{"values":[0,100]}	{"values":[0,1]}	1	1
59	4.6	1.3	6.6	2.9	{"values":[59,4.6,1.3,6.6,2.9]}	{"values":[0,100]}	{"values":[0,1]}	1	1
114	5	2	5.7	2.5	{"values":[114,5,2,5.7,2.5]}	{"values":[0,100]}	{"values":[0,1]}	1	1
106	6.6	2.1	7.6	3	{"values":[106,6.6,2.1,7.6,3]}	{"values":[0,100]}	{"values":[0,1]}	1	1
7	1.4	0.3	4.6	3.4	{"values":[7,1.4,0.3,4.6,3.4]}	{"values":[50,0]}	{"values":[1,0]}	0	0
128	4.9	1.8	6.1	3	{"values":[128,4.9,1.8,6.1,3]}	{"values":[0,100]}	{"values":[0,1]}	1	1
93	4	1.2	5.8	2.6	{"values":[93,4,1.2,5.8,2.6]}	{"values":[0,100]}	{"values":[0,1]}	1	1
135	5.6	1.4	6.1	2.6	{"values":[135,5.6,1.4,6.1,2.6]}	{"values":[0,100]}	{"values":[0,1]}	1	1
145	5.7	2.5	6.7	3.3	{"values":[145,5.7,2.5,6.7,3.3]}	{"values":[0,100]}	{"values":[0,1]}	1	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

6.4.2.2.2. Regression as Algorithm Type

i) If the selected algorithm type is '**Regression**' (from the '**Properties**' tab)

ii) Users need to configure the following information:
- **Validation** (If validation is enabled)
  - a. **Model Selection**
    - i. **Model Selection Method**: Select any one validation method using the drop-down menu:
      1. **Train Validation**: By selecting this method, the '**Train Ratio**' field will be displayed to configure.
      2. **Cross-Validation**: By selecting this method, the '**Number of folds**' field will be displayed to configure.
    - ii. **Evaluator**: Select any one option using the drop-down menu to define evaluator. Evaluator consist of three types:
      1. **Multi-Class Classification** – If the dataset has multiple classes in the label column
      2. **Binary Class Classification**- if the data set has two classes in label Column
      3. **Regression Class Classification**-if the '**Label**' column is continuous.
    - iii. **Number of folds**: This field will be displayed if cross-validation has been selected via the '**Model Selection Method**' field

iii) Click '**NEXT**' (The '**Apply**' option turns into '**Next**' when '**Validation**' is enabled).

- **Advanced**
  - a. **Column Selection**
    - i. **Maximum Depth**: Maximum depth of the tree. (>= 0) E.g., depth 0 means 1 leaf node; depth 1 means 1 internal node + 2 leaf nodes. (Type integer only. Default value 5.)

ii. **Maximum Bins:** Maximum number of bins for discretizing continuous features. (The value must be of integer type only, it should be >=2 and >=number of categories for any categorical feature. The default value is 32.)

iii. **Minimum Instances Per Node:** Minimum number of instances each child must have after the split is referred to as Minimum Instances Per Node. The split will be discarded as invalid if it causes the left or right child to have fewer than minimum instances per node. (The value should be >=1, the default value for the field is 1, only integer value should be allowed)

iv. **Minimum Info Gain:** Enter Minimum Info Gain for a split to be considered at a tree-node (Type double only. Default value 0.0)

iv) Click '**APPLY**'
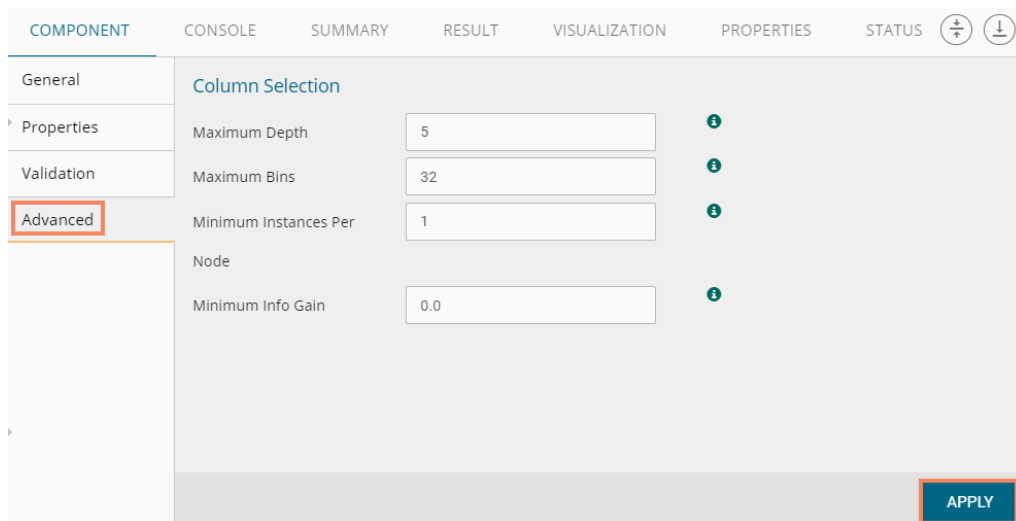
v) Configure the Spark Apply Model component by clicking the '**APPLY**' option

vi) After getting the success message run the workflow

  a. A message will pop-up to confirm whether users want to enable logging.

  b. Click '**NO**'

vii) Users will get the process status under the '**CONSOLE**' tab

14/4/2018 - 16:32:39 : Process Initiated...

14/4/2018 - 16:32:42 : Number of Rows fetched : 150

14/4/2018 - 16:32:42 : cassandra0 Completed

14/4/2018 - 16:32:42 : Spark-Decision-Tree1 Running

14/4/2018 - 16:32:43 : Spark-Decision-Tree1 Completed

14/4/2018 - 16:32:43 : Spark Apply Model2 Running

14/4/2018 - 16:32:43 : Spark Apply Model2 Completed

14/4/2018 - 16:32:43 : Process Completed

viii) Follow the below given steps to display the result view:
a. Click the dragged algorithm component onto the workspace.
b. Click the '**RESULT**' tab.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES   STATUS

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	dfFeaturesCol1	rawPrediction1	probability1	cat	prediction1
83	3.9	1.2	5.8	2.7	{"values":[83,3.9,1.2,5.8,2.7]}	{"values":[0,100]}	{"values":[0,1]}	1	1
111	5.1	2	6.5	3.2	{"values":[111,5.1,2,6.5,3.2]}	{"values":[0,100]}	{"values":[0,1]}	1	1
59	4.6	1.3	6.6	2.9	{"values":[59,4.6,1.3,6.6,2.9]}	{"values":[0,100]}	{"values":[0,1]}	1	1
114	5	2	5.7	2.5	{"values":[114,5,2,5.7,2.5]}	{"values":[0,100]}	{"values":[0,1]}	1	1
106	6.6	2.1	7.6	3	{"values":[106,6.6,2.1,7.6,3]}	{"values":[0,100]}	{"values":[0,1]}	1	1
7	1.4	0.3	4.6	3.4	{"values":[7,1.4,0.3,4.6,3.4]}	{"values":[50,0]}	{"values":[1,0]}	0	0
128	4.9	1.8	6.1	3	{"values":[128,4.9,1.8,6.1,3]}	{"values":[0,100]}	{"values":[0,1]}	1	1
93	4	1.2	5.8	2.6	{"values":[93,4,1.2,5.8,2.6]}	{"values":[0,100]}	{"values":[0,1]}	1	1
135	5.6	1.4	6.1	2.6	{"values":[135,5.6,1.4,6.1,2.6]}	{"values":[0,100]}	{"values":[0,1]}	1	1
145	5.7	2.5	6.7	3.3	{"values":[145,5.7,2.5,6.7,3.3]}	{"values":[0,100]}	{"values":[0,1]}	1	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

6.4.2.3. Spark Random Forest
The Random Forest is a top performer tree ensemble algorithm for classification and regression tasks. The algorithm builds multiple decision trees based on different subsets of the features in the data. Outcomes are then predicted by running observations through all the trees and averaging the individual predictions.

6.4.2.4. Classification as the Algorithm Type
i) Drag the Spark Random Forest component to the workspace and connect to a configured data source.
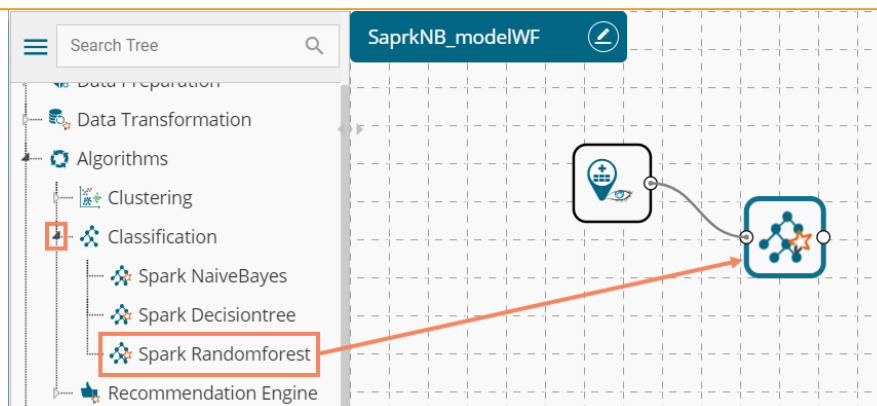
ii) Connect the Spark Random Forest basic workflow with a configured '**Spark Apply Model**' and '**Spark Performance**' component to get and the result view.
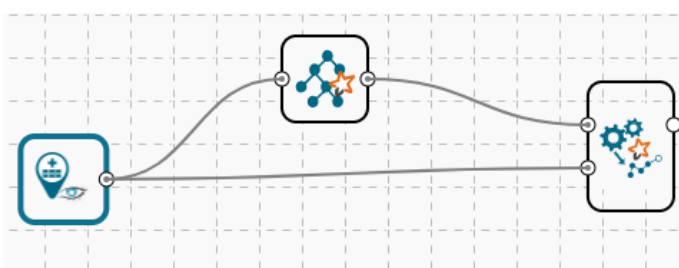
iii) Configure the required information:
- **Properties**
  a. **Column Selection**
     i. **Feature:** Select feature columns from the drop-down menu.
     ii. **Label:** Select a binary column as a label from the drop-down menu.
     iii. **Algorithm Type:** Select an algorithm type from the drop-down menu.
        1. **Classification:** Select this option if users want to pass dependent column as the categorical values (Default option)
        2. **Regression:** Select this option if users want to pass dependent column as numerical values.
     iv. **Seeds:** Enter numerical value to randomize data (Only integer value).
     v. **Enable Validation:** Enable validation by check marking the box.

iv) Click '**NEXT**'

- **Validation (if 'Validation' is enabled)**
  - a. **Model Selection**
    - i. **Model Selection Method**: Select any one validation method using the drop-down menu:
      1. **Train Validation**: By selecting this method, the '**Train Ratio**' field will be displayed to configure.
      2. **Cross-Validation**: By selecting this method, the '**Number of folds'** field will be displayed to configure**.**
    - ii. **Evaluator**: Select any one option using the drop-down menu to define evaluator. Evaluator consist of three types:
      1. **Multi-Class Classification** – If the dataset has multiple classes in the label column
      2. **Binary Class Classification**- if the data set has two classes in label Column
      3. **Regression Class Classification**-if the '**Label**' the column is continuous
    - iii. **Train Ratio**: This field will be displayed if train validation has been selected via the '**Model Selection Method**' field.
- v) Click '**NEXT**' (The '**Apply**' option turns into '**NEXT**' when '**Validation**' is enabled).

- **Advanced**
  - a. **Column Selection**
    - i. **Feature Subset Strategy:** Select an option from the drop-down menu**.** The number of features to consider for splits at each tree-node (Supported options: auto, all, n, one-third, sqrt, log2).
    - ii. **Maximum Depth**: Maximum depth of the tree. (>= 0) E.g. depth 0 means 1 leaf node; depth 1 means 1 internal node + 2 leaf nodes. (Type integer only. Default value 5.)
    - iii. **Maximum Bins:** Maximum number of bins for discretizing continuous features. (The value must be >=2 and >=number of categories for any categorical feature. (Type integer only. Default value 32.)
    - iv. **Minimum Instances Per Node:** Minimum number of instances each child must have after the split is referred to as Minimum Instances Per Node. The split will be discarded as invalid if it causes the left or right child to have fewer than minimum instances per node. (The value should be >=1, the default value for the field is 1, only integer value should be allowed)
    - v. **Minimum Info Gain:** Enter min. Info. Gain for a split to be considered at a tree-node. (Type double only. Default value 0.0)
    - vi. **Number of Trees**: Enter the number of trees to train (>=1).
    - vii. **Thresholds:** Thresholds in multiclass classification to adjust the probability of predicting each class. The array must have a length equal to the number of classes, with values >=0. This class with the largest value p/t is predicted, where '**p**' is the optional probability of that class and '**t**' is the class' threshold. (Type: Comma separate double value. Thresholds will be displayed only in case of the Classification algorithm type.)

viii. **Impurity:** Select an option from the drop-down menu. The **'impurity'** field is a measure of the homogeneity of the labels at the node. The current implementation of the algorithm gives two impurity measures for classification.

    1. Gini

    2. Entropy

  ix. **Sub Sampling Rate**: Set sub sampling rate (Default value is 1).

vi)   Click '**APPLY**'
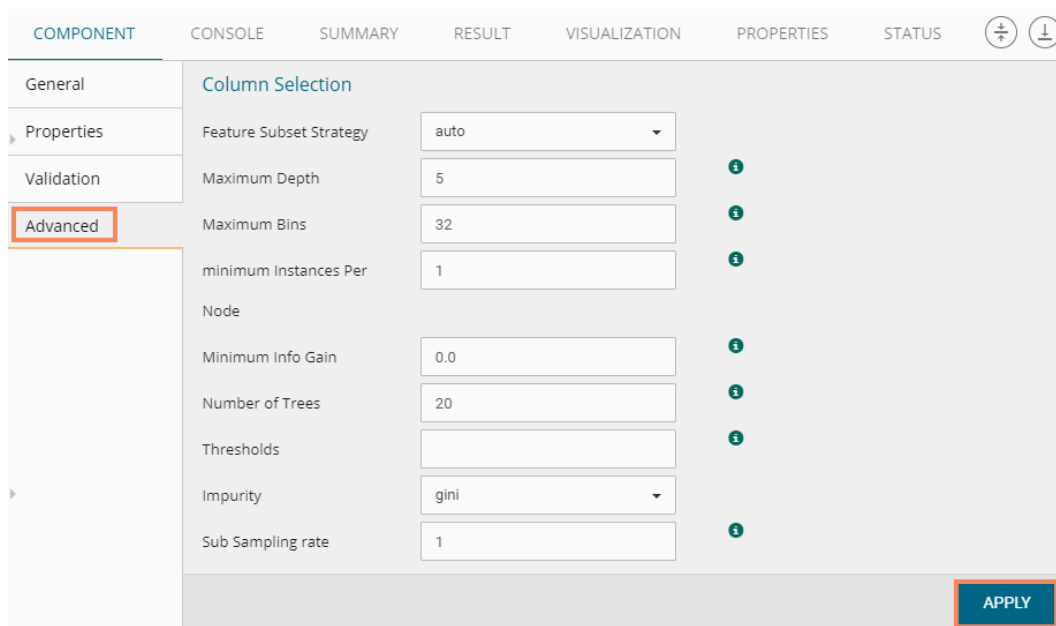
vii)   Configure the component tab for the **'Apply Model'** component and click '**APPLY**'
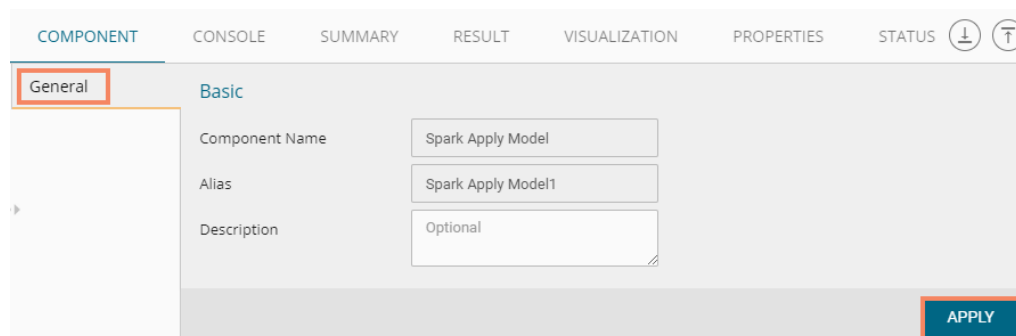
viii)   After getting success message run the workflow

    a. A message will pop-up to confirm whether users want to enable logging

    b. Click '**NO**'
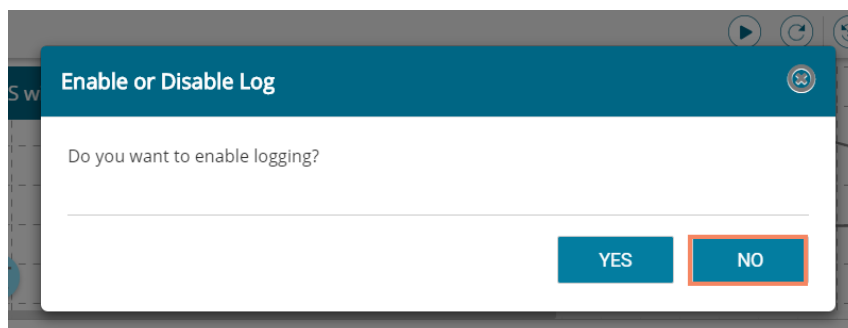
ix) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	**CONSOLE**	SUMMARY

14/4/2018 - 18:19:7 : Process Initiated...

14/4/2018 - 18:19:9 : Process started

14/4/2018 - 18:19:9 : cassandra0 Running

14/4/2018 - 18:19:10 : Number of Rows fetched : 150

14/4/2018 - 18:19:10 : cassandra0 Completed

14/4/2018 - 18:19:10 : Spark-RandomForest2 Running

14/4/2018 - 18:19:11 : Spark-RandomForest2 Completed

14/4/2018 - 18:19:11 : Spark Apply Model1 Running

14/4/2018 - 18:19:11 : Spark Apply Model1 Completed

14/4/2018 - 18:19:11 : Process Completed

x) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.

COMPONENT CONSOLE SUMMARY **RESULT** VISUALIZATION PROPERTIES STATUS

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	rfFeaturesCol2	rawPrediction2	probability2	cat	prediction2
51	4.7	1.4	7	3.2	{"values":[51,4.7,1.4,7,3.2]}	{"values":[2,18]}	{"values":[0.1,0.9]}	1	1
46	1.4	0.3	4.8	3	{"values":[46,1.4,0.3,4.8,3]}	{"values":[20,0]}	{"values":[1,0]}	0	0
14	1.1	0.1	4.3	3	{"values":[14,1.1,0.1,4.3,3]}	{"values":[20,0]}	{"values":[1,0]}	0	0
31	1.6	0.2	4.8	3.1	{"values":[31,1.6,0.2,4.8,3.1]}	{"values":[20,0]}	{"values":[1,0]}	0	0
81	3.8	1.1	5.5	2.4	{"values":[81,3.8,1.1,5.5,2.4]}	{"values":[0,20]}	{"values":[0,1]}	1	1
90	4	1.3	5.5	2.5	{"values":[90,4,1.3,5.5,2.5]}	{"values":[0,20]}	{"values":[0,1]}	1	1
74	4.7	1.2	6.1	2.8	{"values":[74,4.7,1.2,6.1,2.8]}	{"values":[0,20]}	{"values":[0,1]}	1	1
10	1.5	0.1	4.9	3.1	{"values":[10,1.5,0.1,4.9,3.1]}	{"values":[20,0]}	{"values":[1,0]}	0	0
29	1.4	0.2	5.2	3.4	{"values":[29,1.4,0.2,5.2,3.4]}	{"values":[20,0]}	{"values":[1,0]}	0	0
55	4.6	1.5	6.5	2.8	{"values":[55,4.6,1.5,6.5,2.8]}	{"values":[0,20]}	{"values":[0,1]}	1	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note: There is no change in the advanced tab or result when '**Validation**' is disabled for Spark Random Forest with a classification algorithm type.

6.4.2.5. Regression as Algorithm Type

i) If the selected algorithm type is '**Regression**' (from the '**Properties**' tab)

- **Validation**
  a. **Model Selection Method**: Select any one validation method using the drop-down menu:
     i. Train Validation
     ii. Cross-Validation
  b. **Evaluator**: Select any one option using the drop-down menu to define evaluator. Evaluator consist of three types:
     i. **Multi-Class Classification** – If the data set has multiple classes in the label column
     ii. **Binary Class Classification**- If the data set has two classes in label Column
     iii. **Regression Class Classification**-If the '**Label**' column is continuous
  c. **Train Ratio**: This field will be displayed if train validation has been selected by using the '**Model Selection Method**' field.

ii) Click '**NEXT**'

- **Advanced**
  a. **Column Selection**
     i. **Feature Subset Strategy:** Select an option from the drop-down menu**.** The number of features to consider for splits at each tree-node (Supported options: auto, all, n, one-third, sqrt, log2).
     ii. **Maximum Depth**: Maximum depth of the tree. (>= 0) E.g., depth 0 means 1 leaf node; depth 1 means 1 internal node + 2 leaf nodes. (Type integer only. Default value 5.)
     iii. **Maximum Bins:** Maximum number of bins for discretizing continuous features. (The value must be >=2 and >=number of categories for any categorical feature. (Type integer only. Default value 32.)
     iv. **Minimum Instances Per Node:** Minimum number of instances each child must have after the split is referred to as Minimum Instances Per Node. The split will be

discarded as invalid if it causes the left or right child to have fewer than minimum instances per node. (The value should be >=1, the default value for the field is 1, only integer value should be allowed)

    v. **Minimum Info Gain:** Enter Minimum Info Gain for a split to be considered at a tree-node. (Type double only. Default value 0.0)

    vi. **Number of Trees**: Enter the number of trees to train (>=1).

    vii. **Impurity:** Select an option from the drop-down menu. The '**impurity**' field is a measure of the homogeneity of the labels at the node. The current implementation of the algorithm provides two impurity measures for classification.

        1. Gini

        2. Entropy

    viii. **Sub Sampling Rate**: Set sub sampling rate (Default value is 1).

iii) Click '**APPLY**'

iv) Configure the '**Apply Model**' component and click '**APPLY**' option

v) After getting success message run the workflow

    a. A message will pop-up to confirm whether users want to enable logging

    b. Click '**NO**'

Enable or Disable Log

Do you want to enable logging?

YES   NO
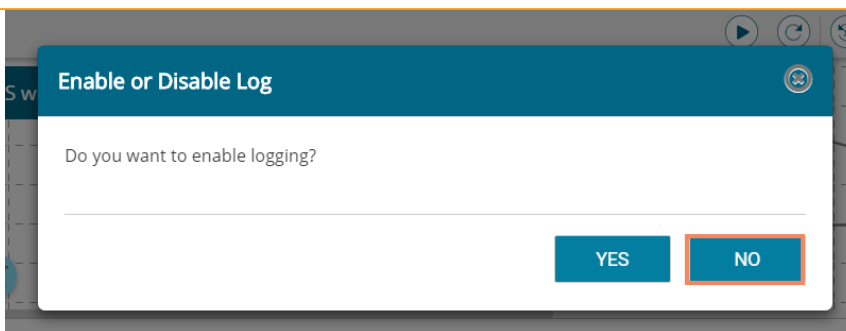
vi) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY

14/4/2018 - 19:50:1   : Process Initiated...

14/4/2018 - 19:50:4   : Number of Rows fetched : 150

14/4/2018 - 19:50:4   : cassandra0 Completed

14/4/2018 - 19:50:4   : Spark-RandomForest1 Running

14/4/2018 - 19:50:5   : Spark-RandomForest1 Completed

14/4/2018 - 19:50:5   : Spark Apply Model2 Running

14/4/2018 - 19:50:5   : Spark Apply Model2 Completed

14/4/2018 - 19:50:5   : Process Completed

vii) Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace
   b. Click the '**RESULT**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

Show 10 entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	rfFeaturesCol1	cat	prediction1
83	3.9	1.2	5.8	2.7	{"values":[3.9,1.2,5.8,2.7,83]}	1	1
111	5.1	2	6.5	3.2	{"values":[5.1,2,6.5,3.2,111]}	1	1
59	4.6	1.3	6.6	2.9	{"values":[4.6,1.3,6.6,2.9,59]}	1	1
114	5	2	5.7	2.5	{"values":[5,2,5.7,2.5,114]}	1	1
106	6.6	2.1	7.6	3	{"values":[6.6,2.1,7.6,3,106]}	1	1
7	1.4	0.3	4.6	3.4	{"values":[1.4,0.3,4.6,3.4,7]}	0	0
128	4.9	1.8	6.1	3	{"values":[4.9,1.8,6.1,3,128]}	1	1
93	4	1.2	5.8	2.6	{"values":[4,1.2,5.8,2.6,93]}	1	1
135	5.6	1.4	6.1	2.6	{"values":[5.6,1.4,6.1,2.6,135]}	1	1
145	5.7	2.5	6.7	3.3	{"values":[5.7,2.5,6.7,3.3,145]}	1	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note: Users can click the '**SUMMARY**' tab to view the model summary after connecting to a Spark Apply Model component. The Summary will be displayed if the '**Apply Model**' component contains summary to show.

6.4.3. Recommendation Engine

The Recommendation Engine algorithm helps to build a prediction model. The algorithm will consider the known user-item association as training data. The Training data is then used to predict the unknown set of data on Test data.

6.4.3.1. Spark ALS

The Spark ALS (Alternating Least Squares) can be used to make a primary recommendation. This feature uses the collaborative filtering techniques by filling in the missing entries of a user-item association matrix. Spark currently supports model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries.

Users can use this component as in spark pipeline and predict what people might like and to uncover relationships between items to aid in the discovery process.

i) Drag the Spark ALS component to the workspace and connect to a configured data source and other required pipeline components as shown below:
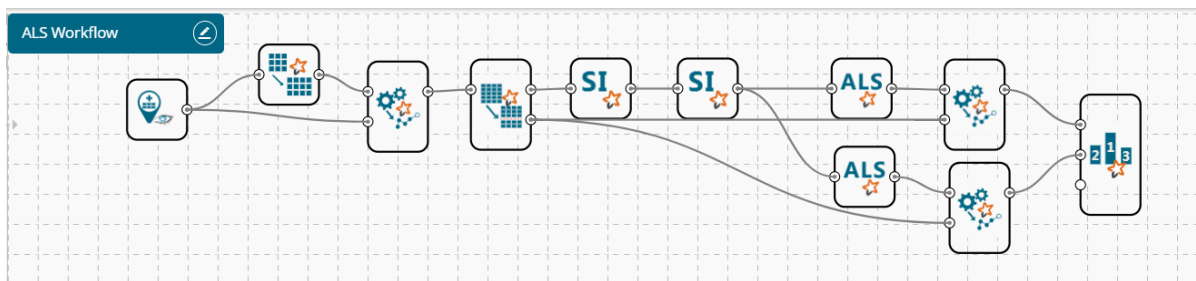
Configure the following fields in the '**Properties**' tab:

    a. **Column Selection**
        i. **User:** Select a user column from the drop-down menu.
        ii. **Item:** Select an item column from the drop-down menu.
        iii. **Rating:** Select a rating column from the drop-down menu.

ii) Click '**Apply**' (If you do not require to configure '**Advanced**' tab. Else, configure the '**Advanced**' tab).
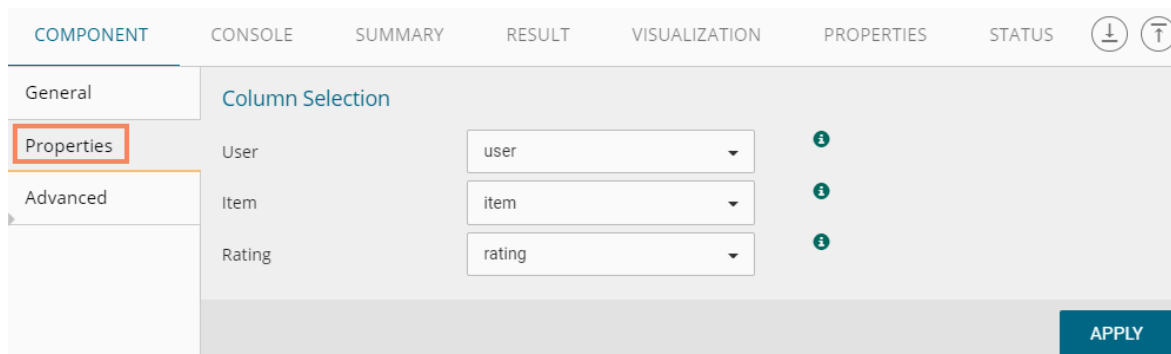
iii) Configure the required '**Advanced**' information:
    a. **Input Data Handling**

    i. **Number of Item Block:** Items will be partitioned as per the entered the number of item block to parallelize computation (default value is 10).

    ii. **Number of User Block:** Users will be partitioned as per the entered number of user block to parallelize computation (default value is 10).

    iii. **Rank:** This refers to the number of factors in the ALS model, that is the number of hidden

features in our low-rank approximation matrices.
Generally, the higher the number of factors, the better, but this has a direct impact on memory usage, both for computation and to store models for serving, particularly for a large number of users or items. Hence, this is often a trade-off in real-world use cases. A rank in the range of 10 to 200 is usually reasonable (default value is 10).

    iv. **Max Iteration:** This refers to the number of iterations to run. Each iteration in ALS is guaranteed to decrease the reconstruction error of the rating matrix. ALS models will converge to a reasonably good solution after relatively few iterations. Users do not require to run for too many iterations in most cases (Default value is 10)

    v. **Reg. Param:** This parameter controls the regularization and overfitting of the ALS model.
The regularization value is dependent on the size, nature, and sparsity of the underlying data. The '**Reg. Param**' should be tuned using the sample test data and cross-validation approach.

    vi. **Alpha:** Alpha is a parameter applicable to the implicit feedback **a** variant of ALS that governs the baseline confidence in preference observations (Default value is 1.0).

    vii. **Seed**: to replicate the randomization of data

    viii. **Implicit:** ImplicitPrefs specifies whether to use the explicit feedback ALS variant or one adapted for implicit feedback data (Default value is '**false**' which means to use explicit feedback).

    ix. **Non-Negative:** Enable '**Non-Negative**'with a checkmark to use non-negative constraints for least squares (Default value is '**False**')

iv)   Click '**APPLY**'

v)   After getting a successful message run the workflow
    a. A message will pop-up to confirm whether users want to enable logging
    b. Click '**No**'

Enable or Disable Log

Do you want to enable logging?

YES　　NO
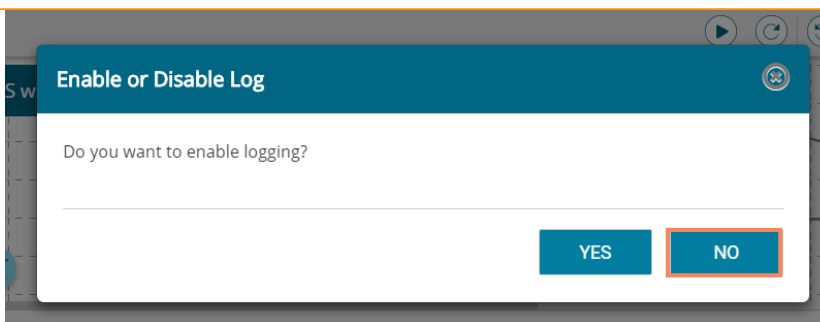
vi)　　Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT |

14/4/2018- 13:43:34　: Process Initiated...

14/4/2018 13:43:38　: Number of Rows fetched : 14861

14/4/2018- 13:43:38　: cassandra0 Completed

14/4/2018- 13:43:38　: Spark SQL Transformer1 Running

14/4/2018- 13:43:38　: Spark SQL Transformer1 Completed

14/4/2018- 13:43:38　: Spark Apply Model2 Running

14/4/2018- 13:43:39　: Spark Apply Model2 Completed

14/4/2018- 13:43:39　: Spark Split Data3 Running

14/4/2018- 13:43:39　: Spark Split Data3 Completed

14/4/2018- 13:43:39　: Spark String Indexer4 Running

14/4/2018- 13:43:39　: Spark String Indexer4 Completed

14/4/2018- 13:43:39　: Spark String Indexer5 Running

14/4/2018- 13:43:39　: Spark String Indexer5 Completed

14/4/2018- 13:43:39　: Spark-ALS6 Running

14/4/2018- 13:43:41　: Spark-ALS6 Completed

14/4/2018 - 13:43:41　: Spark-ALS7 Running

14/4/2018 - 13:43:44　: Spark-ALS7 Completed

14/4/2018 - 13:43:44　: Spark Apply Model8 Running

14/4/2018- 13:43:45　: Spark Apply Model8 Completed

14/4/2018- 13:43:45　: Spark Apply Model9 Running

14/4/2018- 13:43:45　: Spark Apply Model9 Completed

14/4/2018- 13:43:45　: Spark-Performance10 Running

14/4/2018 - 13:43:46　: Spark-Performance10 Completed

14/4/2018 - 13:43:46　: Process Completed

vii)　　Follow the below given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.
viii)　　A new column will be added to the '**RESULT**' view.

Show 10 entries   Search:

accname	itemname	user	item	rating	prediction7
	Juice - Variety of 100% All Natural	1015	14	1	0.12025179
	Juice - Variety of 100% All Natural	1069	14	1	1.6354712
	Juice - Variety of 100% All Natural	299	14	1	0.33671662
	Juice - Variety of 100% All Natural	579	14	1	1.0582461
	Juice - Variety of 100% All Natural	28	14	3	1.8499624
	Juice - Variety of 100% All Natural	330	14	1	0.8815267
	Juice - Variety of 100% All Natural	362	14	1	1.0642278
	Juice - Variety of 100% All Natural	110	14	1	0.52995366
	Juice - Variety of 100% All Natural	1039	14	1	0.096204184
	Milk - Organic 1%	399	18	1	1.7953756

Note:
  a. Users need to connect the ALS component with a Spark Apply model to get the result view.

  b. Users can click the '**SUMMARY**' tab to view the model summary after connecting to a Spark Apply Model component. The Summary will be displayed if the '**Apply Model**' component contains summary to show.

6.5. Apply Model

6.5.1. Spark Apply Model

This element is provided to generate predictions based on a Spark trained classification model. Users can view predicted column value and probability of each label class by using the classification model.

Users can create a model via the following ways:
- Generate a model using an algorithm
- Generate a model using the saved models

The Spark Apply Model consists of 2 input nodes and 1 output node.
- **Input Nodes**
  - Upper node – Model/Training data
  - Lower node – Testing data
- **Output Node**
  - Node – Result data

i) Click the '**Apply Model**' tree-node.
ii) The '**Spark Apply Model**' leaf-node will be displayed.
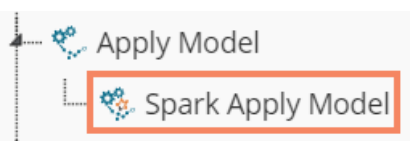
iii) Drag the Spark Apply Model component onto the workspace and connect it with a valid combination of Data source and algorithm (Configure the data source and algorithm components. In this case, the used algorithm is Spark Decision Tree)
iv) Click the '**Spark Apply Model**' component.
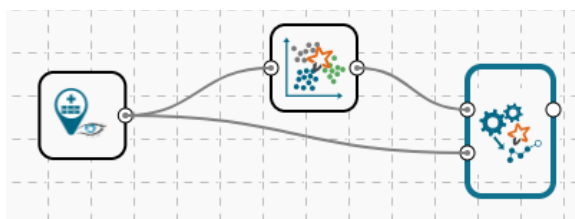
v) Displays the 'Basic' details of the selected component

vi) Click '**APPLY**'

vii) After getting a success message run the workflow
 a. A message will pop-up to confirm whether users want to enable logging
 b. Click '**NO**'

viii) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY

13/8/2018 - 17:37:0   : Process Initiated...

13/8/2018 - 17:37:2   : Process started

13/8/2018 - 17:37:2   : cassandra0 Running

13/8/2018 - 17:37:3   : Number of Rows fetched : 150

13/8/2018 - 17:37:3   : cassandra0 Completed

13/8/2018 - 17:37:3   : Spark-K-Means1 Running

13/8/2018 - 17:37:4   : Spark-K-Means1 Completed

13/8/2018 - 17:37:4   : Spark Apply Model2 Running

13/8/2018 - 17:37:4   : Spark Apply Model2 Completed

13/8/2018 - 17:37:4   : Process Completed

ix) Follow the below given steps to display the result view:
   a. Click the dragged Spark Apply Model component on the workspace.
   b. Click the '**RESULT**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS | | |

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	featuresCol1	ClusterNumber
51	4.7	1.4	7	3.2	1	{"values":[4.7,1.4,7,3.2,1]}	3
46	1.4	0.3	4.8	3	0	{"values":[1.4,0.3,4.8,3,0]}	0
14	1.1	0.1	4.3	3	0	{"values":[1.1,0.1,4.3,3,0]}	0
31	1.6	0.2	4.8	3.1	0	{"values":[1.6,0.2,4.8,3.1,0]}	0
81	3.8	1.1	5.5	2.4	1	{"values":[3.8,1.1,5.5,2.4,1]}	4
90	4	1.3	5.5	2.5	1	{"values":[4,1.3,5.5,2.5,1]}	4
74	4.7	1.2	6.1	2.8	1	{"values":[4.7,1.2,6.1,2.8,1]}	3
10	1.5	0.1	4.9	3.1	0	{"values":[1.5,0.1,4.9,3.1,0]}	0
29	1.4	0.2	5.2	3.4	0	{"values":[1.4,0.2,5.2,3.4,0]}	0
55	4.6	1.5	6.5	2.8	1	{"values":[4.6,1.5,6.5,2.8,1]}	3

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

x) Click the '**PROPERTIES**' tab to view the properties details (This Properties tab display workflow properties).

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS | | |

Created By	
Created At	2018-04-09 14:36:23 +0530
Last Modified By	
Last Modified At	2018-04-13 15:40:35 +0530
Version	3.5.

Note:
   a. The result data set of the model can be written to a database using the Cassandra Writer.
   b. Column header and data type of feature column for both the saved model and testing data should match. If column headers and data types do not match, an alert message will be displayed.
   c. It is not mandatory for the testing dataset to contain a label column.

6.6. Performance

6.6.1. Spark Performance

The Spark Performance component is provided as a leaf-node under the Performance tree-node. It contains 3 input nodes that can be used to compare up to 3 models. Each node has a static name like model_0, model_1, and model_2. Based on the connection to the node model summary can be viewed with respective names.

Spark Performance components can be of the following formats:

1. Binary Classification Metrics: Used when the label has two classes
2. Multi Classification Metrics: Used when the label has 3 or more beta values
3. Regression Evaluator Metrics: Used when the algorithm is of regression type

In the case of multiple models, all the model statistics will come in the summary of performance (up to 3 models can be compared).

Steps to Connect a Spark Performance Component (to a Model)
i) Drag a Spark Performance component to the workspace and connect to a valid workflow (In this example, a workflow created with the Spark Decision Tree algorithm has been used)
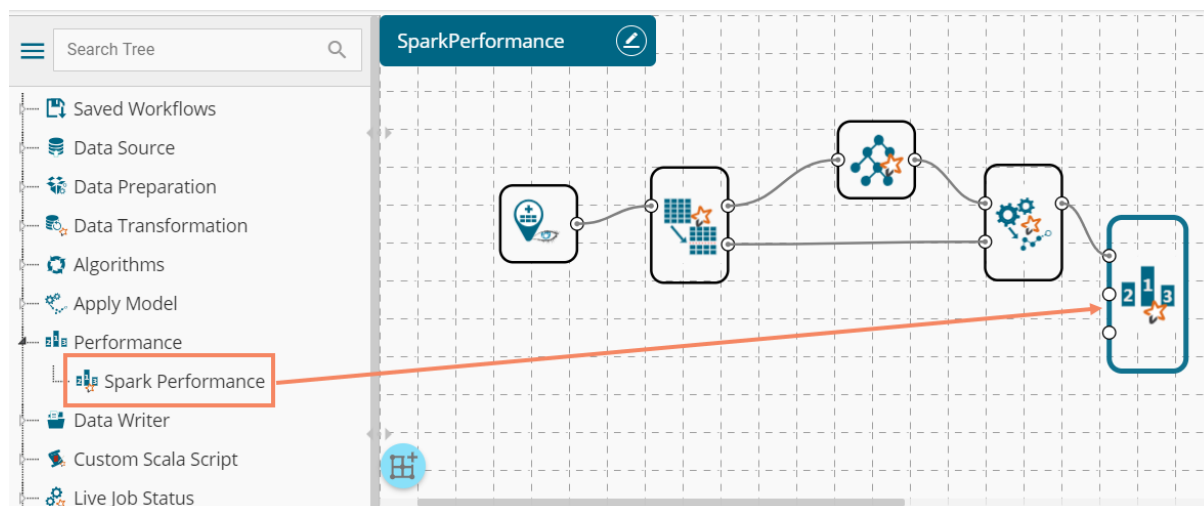
ii) Configure the '**Properties**' tab
   a. **Performance Type**: Select an option out of
      i. Binary Classification Metrics
      ii. Multiclass Classification Metrics (Default option)
      iii. Regression Evaluator Metrics
   b. **Beta Value**: Enter a numerical value
iii) Click '**APPLY**'
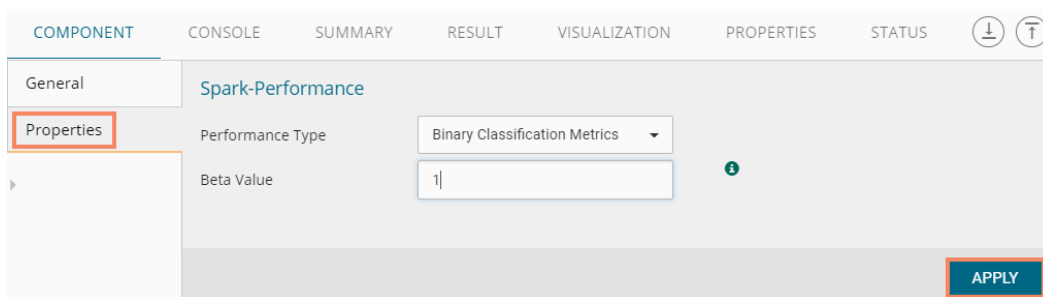
Users will get different outcomes based on the selected Performance types as described below:

- **Multi Classification Metrics**

1. Navigate to the '**Properties**' tab of the Spark Performance component.
2. Select '**Multi Classification Metrics**' Performance type via the drop-down menu
3. Click '**APPLY**'
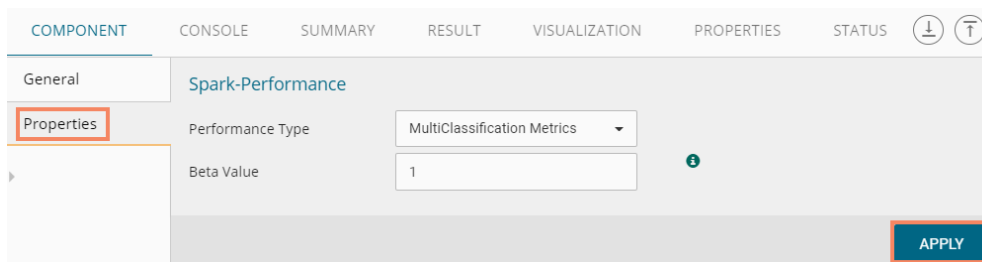
4. After getting success message run the workflow
5. A message will pop-up to confirm whether users want to enable logging
6. Click '**NO**'
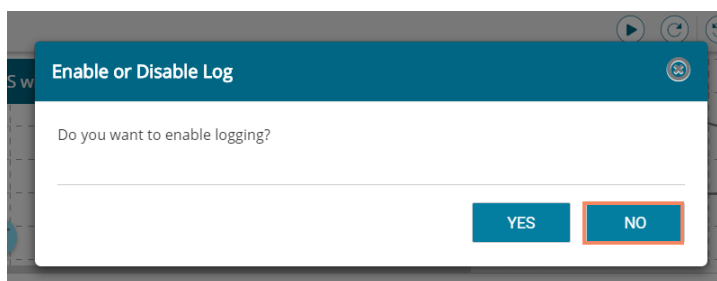
7. Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

14/4/2018 - 14:38:34   : Process Initiated...

14/4/2018 - 14:38:37   : Process started

14/4/2018 - 14:38:37   : cassandra3 Running

14/4/2018 - 14:38:38   : Number of Rows fetched : 150

14/4/2018 - 14:38:38   : cassandra3 Completed

14/4/2018 - 14:38:38   : Spark Split Data0 Running

14/4/2018 - 14:38:38   : Spark Split Data0 Completed

14/4/2018 - 14:38:38   : Spark-NaiveBayes4 Running

14/4/2018 - 14:38:38   : Spark-NaiveBayes4 Completed

14/4/2018 - 14:38:38   : Spark Apply Model1 Running

14/4/2018 - 14:38:38   : Spark Apply Model1 Completed

14/4/2018 - 14:38:38   : Spark-Performance2 Running

14/4/2018 - 14:38:39   : Spark-Performance2 Completed

14/4/2018 - 14:38:39   : Process Completed

8. After the console process gets completed, users can click on the '**SUMMARY**' tab to view Summary of Multiclass Metrics.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

------------ Summary of MultiClass Metrics ------------

Model Name	Accuracy	Weighted Precision	Weighted Recall	Weighted FMeasure	Weighted FMeasure(beta 1.0)	Weighted True Positive Rate	Weighted False Positive Rate
Model 0	1.0	1.0	1.0	1.0	1.0	1.0	0.0

-------- Label Wise Model - 0 ----------

Labels	Precision	Recall	FMeasure	FMeasure(beta 1.0)	TruePositiveRate	FalsePositiveRate
0.0	1.0	1.0	1.0	1.0	1.0	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.0

---- Confusion Matrix (Model - 0)----

	Predict_0.0	Predict_1.0
Actual_0.0	7.0	0.0
Actual_1.0	0.0	23.0

------------ End of Summary ----------

- **Binary Classification Metrics**
    1. Navigate to the '**Properties**' tab of the Spark Performance component
    2. Select '**Binary Classification Metrics**' Performance type via the drop-down menu

    3. Click '**APPLY**'
    4. Run the workflow
    5. A message will pop-up to confirm whether users want to enable logging
    6. Click '**NO**'
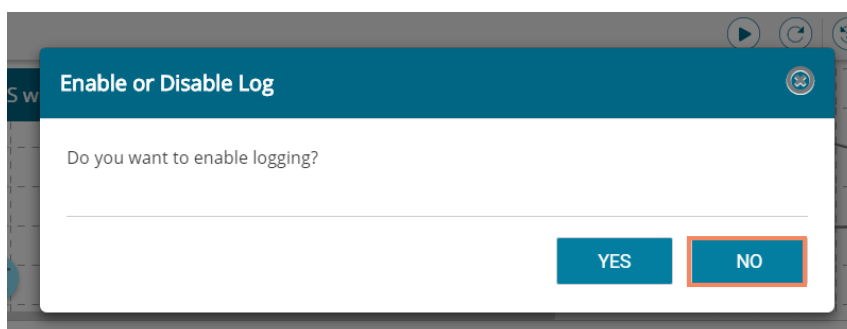
    7. Users will get the process status under the '**CONSOLE**' tab
    8. Users can follow the below given steps to display the result view if the selected performance type is Binary:
        a. Click the dragged performance component on the workspace
        b. Click the '**RESULT**' tab

Model_0

Show 10 ▼ entries   Search: [   ]

falsepositiverate	fMeasure	precision	recall	threshold	fMeasure -beta 1.0
1	0.8461538461538461	0.7586206896551724	0.9565217391304348	-81.44666707663345	0.8461538461538461
1	0.6956521739130435	0.6956521739130435	0.6956521739130435	-74.37026561670204	0.6956521739130435
1	0.06451612903225806	0.125	0.043478260869565216	-51.004805587328576	0.06451612903225806
0.14285714285714285	0	0	0	-32.7685861180848	0
1	0.723404255319149	0.7083333333333334	0.7391304347826086	-75.74011458960186	0.723404255319149
1	0.5365853658536586	0.6111111111111112	0.4782608695652174	-67.24078806597247	0.5365853658536586
0.2857142857142857	0	0	0	-33.091593407986586	0
1	0.5714285714285715	0.631578947368421	0.5217391304347826	-68.91038666853086	0.5714285714285715
1	0.3783783783783784	0.5	0.30434782608695654	-60.54850822615485	0.3783783783783784
1	0.5	0.5882352941176471	0.43478260869565216	-63.21879338526145	0.5

Showing 1 to 10 of 30 entries   Previous 1 2 3 Next

9. Click the '**VISUALIZATION**' tab.
10. The resulting view will be presented via the PR Curve or ROC Curve.
    a. **Result data displayed via the PR Curve**

PR CURVE   ROC CURVE

Precision-Recall curve
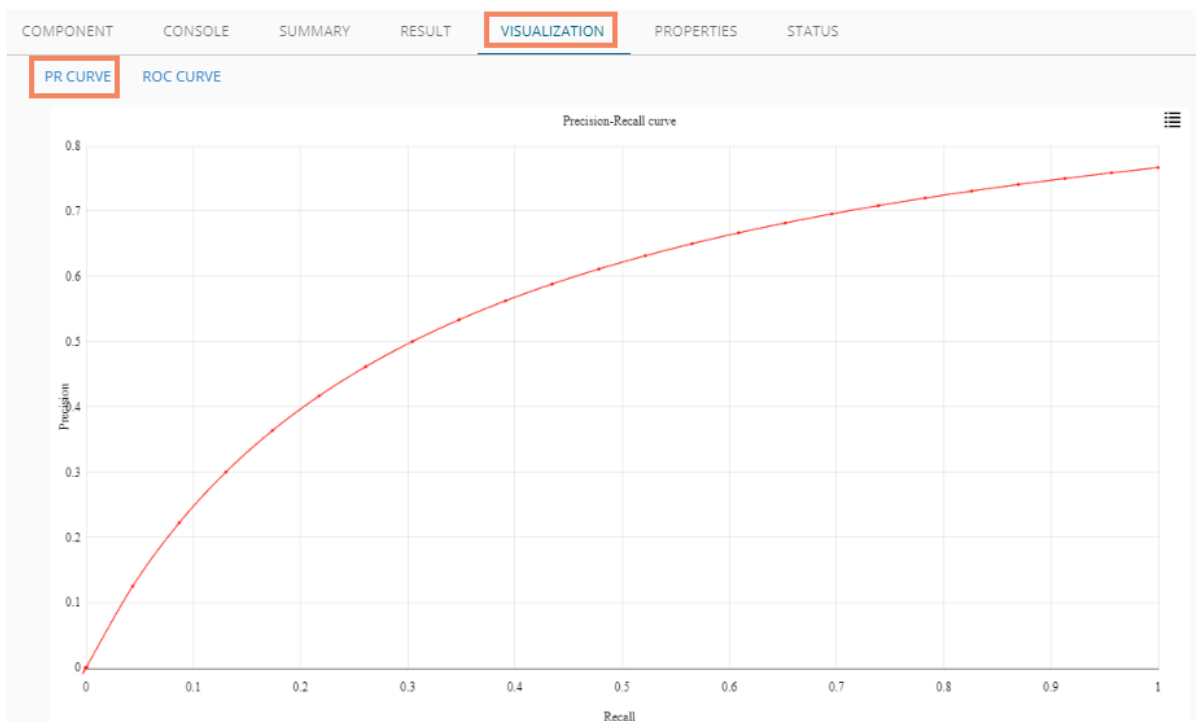
    b. **Result data displayed via the ROC Curve**

PR CURVE   ROC CURVE

ROC Curve
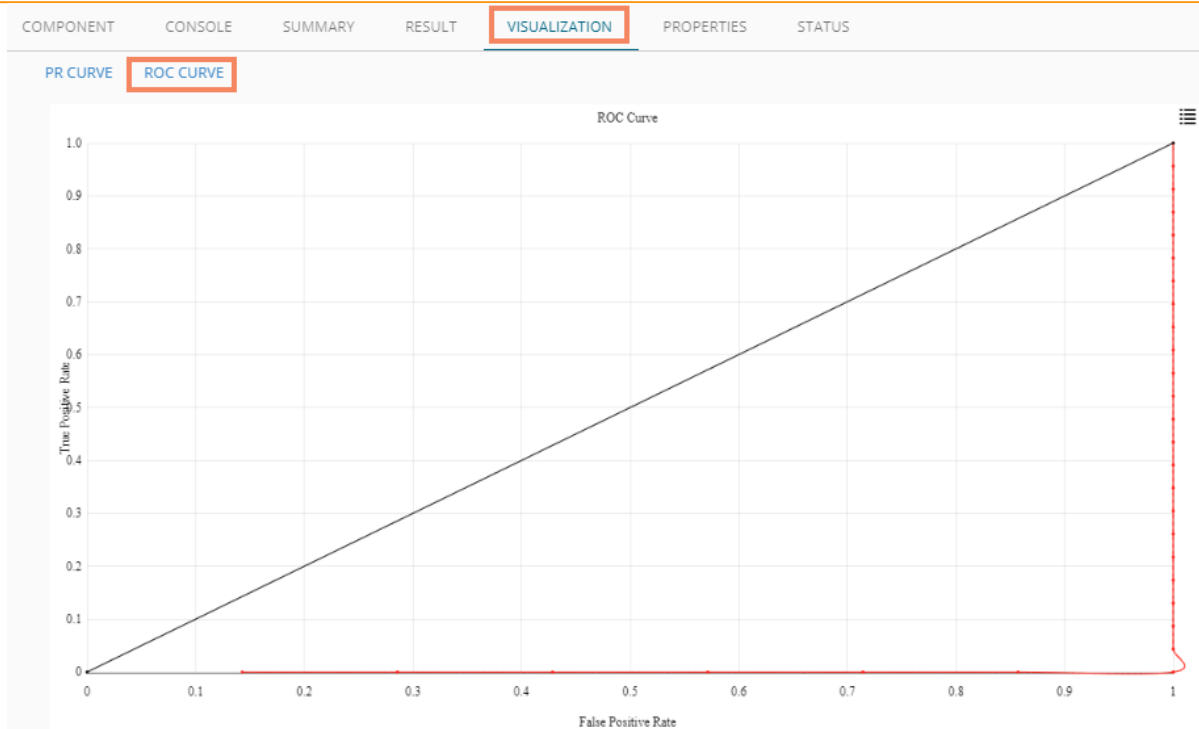
- **Regression Evaluator Metrics**

  The '**Beta Value**' field will not appear on the '**Regression Evaluator Metrics**' Performance type
  1. Navigate to the '**Properties**' tab of the Spark Performance component
  2. Select '**Regression Evaluator Metrics**' Performance type via the drop-down menu
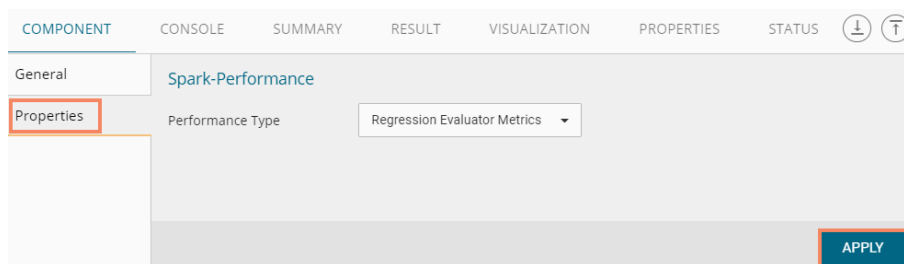
  3. Click '**APPLY**'
  4. After getting success message run the workflow
     a. A message will pop-up to confirm whether users want to enable logging
     b. Click '**NO**'
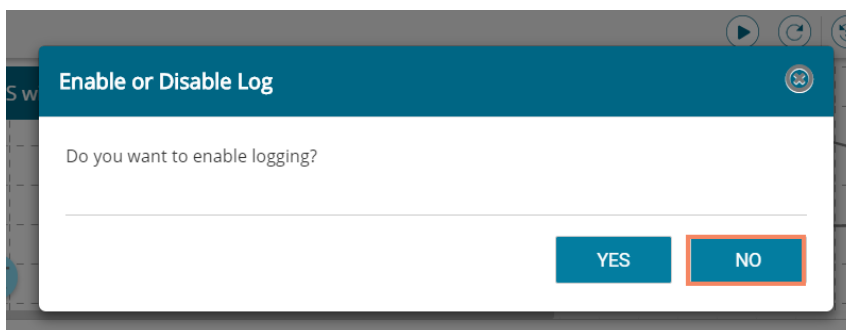
  5. Users will get the process status under the '**CONSOLE**' tab
  6. View summary by following the steps given below:
     a. Click the performance component onto the workspace
     b. Click the '**SUMMARY**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

------------ Summary of the Regression Evaluator Metrics ------------

Model Name	Mean Squared Error (MSE)	Root MSE (RMSE)	Mean Absolute Error (MAE)	Coefficient of Determination (R2)
0	0.0	0.0	0.0	1.0

------------ End of Summary ----------

6.7. Data Writer

6.7.1. Database Writer

6.7.1.1. Internal Data Writer
This data writer will store the data in databases like MySQL, MSSQL, and Oracle.

i) Click **'TreeNode'** provided next to the **'Data Writer'** option
ii) Select **'Database Writer'** option
iii) Select and drag **'Internal Data Writer'** component to the workspace
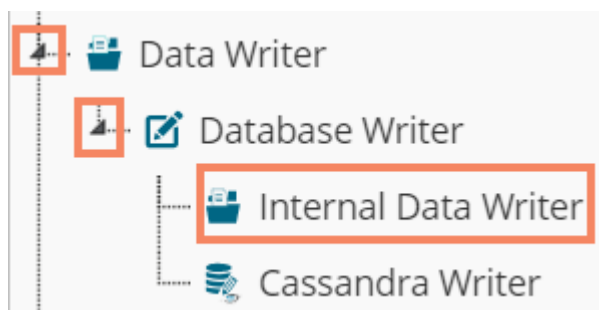
iv) Drag and Connect the '**Internal Data Writer**' component to a configured data source onto the workspace
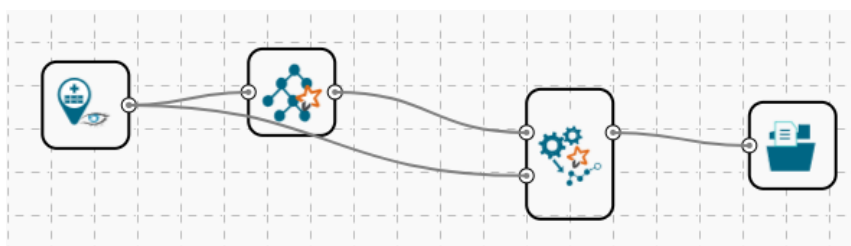
v) Click '**Internal Data Writer**' component to access the Component properties

Users will have different '**Properties**' fields based on the selected table operation as described below:

a. Selecting the 'Create a New Table' as Table Operation:
  i. **Data Source Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.

ii. **Type**: This field will be preselected based on the selected data Connector
iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
iv. **Database Name**: Select a database name from the drop-down menu
v. **Password:** Enter the database password
vi. **Table Name**: Select '**Create New Table**' option from the list
vii. **Table Operation**: Select an option from the drop-down menu
1. Append to Table
2. Overwrite Table
viii. **Create New Table:** It is an optional field. It appears when the user selects '**Create New Table**' option from the '**Table Name**' drop-down menu.
ix. **Auto Increment:** Select an option to enable or disable the auto increment. By enabling this option, a new column will be added to the dataset, and the same column will be selected as the primary key by default.
x. **Auto Increment Label:** Enter a name for the auto increment label
xi. **Column Selected from the model**: Select columns that are needed to be written into the
   selected database
vi) Click '**NEXT**'

vii) Users will be redirected to the '**Schema Viewer**' option
   a. Select Primary Keys: Select primary key(s) using the drop=down menu
viii) Click '**APPLY**'

b. **Selecting an Existing Table as the 'Table Operation':**

  i. **Data Connector Name**: Select a data connector from the drop-down menu
  ii. **Type**: Displays a type based on the selected data connector
  iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
  iv. **Database Name**: Select a database name from the drop-down menu
  v. **Password:** Enter the database password
  vi. **Table Name**: Select an existing table name from the drop-down menu
  vii. **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:
    1. Append Table
    2. Overwrite Table
  viii. **Column Selected from the model**: Select columns that are needed to be written into the
   selected database.

  ix. **Details of the Selected table**: Displays column headers from the selected table.
ix) Click '**NEXT**'

x) Users will be redirected to the '**Schema Viewer**' page.
xi) Click **'APPLY'**

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

General

Properties

Schema Viewer

Internal Data Writer Properties

| Selected Primary Keys |
No primary keys For this table

APPLY
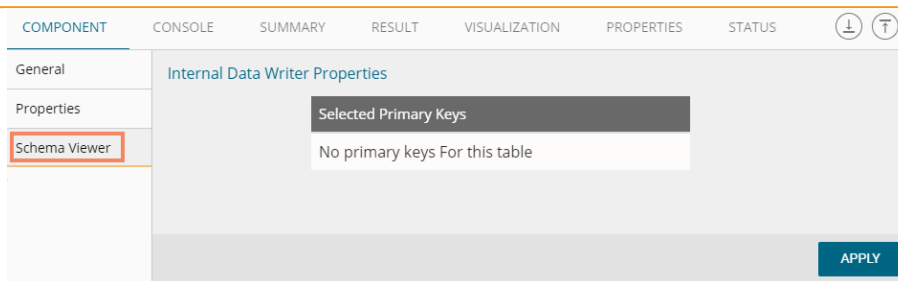
xii)   After getting the success message run the workflow

    a. Users will be asked to enable or disable log

    b. Click '**NO**'

Enable or Disable Log

Do you want to enable logging?

YES   NO

xiii)   Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT |

12/4/2018 - 12:4:20 : Process Initiated...

12/4/2018 - 12:4:34 : Number of Rows fetched : 150

12/4/2018 - 12:4:34 : Data Service0 Completed

12/4/2018 - 12:4:34 : Spark-NaiveBayes1 Running

12/4/2018 - 12:4:35 : Spark-NaiveBayes1 Completed

12/4/2018 - 12:4:35 : Spark Apply Model2 Running

12/4/2018 - 12:4:35 : Spark Apply Model2 Completed

12/4/2018 - 12:4:35 : DataWriter Process Running

12/4/2018 - 12:4:35 : DataWriter Process Completed

12/4/2018 - 12:4:35 : Process Completed

xiv)   The data will be saved in the selected database at the end of the process

Note:

    a. Users will not be able to see the '**Result**' tab for the Internal Data Writer.

    b. Auto Increment Column(delta load) supports only for MySQL. Users can configure the Auto Increment Column only while using the 'Create New Table' option as a Table Name.

    c. By selecting an auto increment column by default, it will be selected as the primary key. If users want to use another column as a primary key other than the Auto Increment Column, then it has to be configured using the '**Schema Viewer**' tab.

    d. If users do not mention primary key for the 'Upsert' table operation, it will act as the '**Append**' operation

6.7.1.2. Cassandra Writers

Cassandra Writer can be used to store the predictive executions.

i) Click '**TreeNode**' provided next to the '**Data Writer**' option
ii) Select '**Database Writer**'
iii) Select and drag '**Cassandra Writer**' component to the workspace
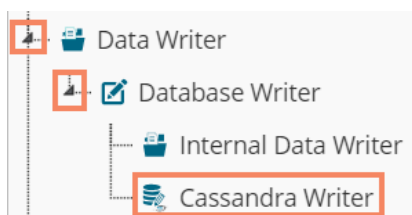
iv) Connect the '**Cassandra Writer**' to a configured data source or a workflow
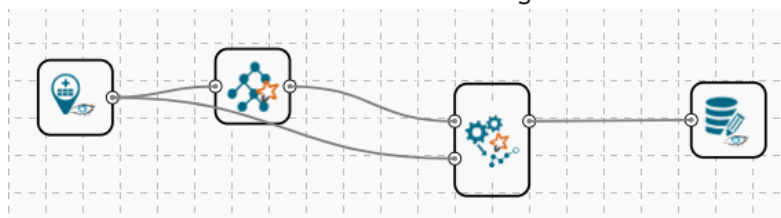
v) Click the '**Cassandra Writer**' component to access it
vi) Configure the following Properties details:
   a. **Selecting Create New Table as Table option**
      i. **Select Data Connector**: Select a data connector using the drop-down menu
      ii. **Host Name**: Based on the chosen data connector a hostname will be displayed (Users cannot edit this field)
      iii. **Port Name**: The server port number will be displayed (Users cannot edit this field)
      iv. **Username**: Username of the selected connection appears by default. (Users cannot edit this field)
      v. **Password: the database password**
      vi. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
      vii. **Select Key Space**: Select a keyspace using the drop-down menu
      viii. **Replication Factor**: The replication factor mentioned in the selected '**Key Space**' will be displayed (Users cannot edit this field)
      ix. **Select Table**: Select 'Create a New Table table from the drop-down menu
      x. **Select Columns**: Select the columns that you want to write
      xi. **Consistency**: Select an option from the drop-down menu
      xii. **New Table**: Provide a name for the newly created table
      xiii. **New time uuid column name**: Enter a UUID column name
vii) Click '**NEXT**'

viii) Users will be redirected to the '**Key Specification**' tab.

ix) Configure the following information:

    a. **Headers**: All the columns from the data set will be listed.

    b. **Partition Key (Name)**: The Partition Key determines which node stores the data. It is responsible for data distribution across the nodes.
- The UUID Column name will be displayed under the '**Partition Key**' window.
- Users can select and move any column from '**Header**' (Select Column) to '**Partition Key**' space.
- The sequence of the columns listed under Partition Key can be arranged by using '**Up**' or '**Down**' options.

    c. **Clustering Key**: The Clustering Key is a storage engine process that sorts data within the partition. It determines per-partition clustering.
- The items listed under the Clustering Key box can be arranged by using '**Up**' or '**Down**' options.
- Users can select any column from '**Headers**'(Select Column) to '**Clustering Key**' space.

x) Click '**APPLY**'

xi) After getting success message run the workflow
   a. A message will pop-up to confirm whether users want to enable logging
   b. Click '**NO**'

xii) Users will be redirected to the '**CONSOLE**' tab
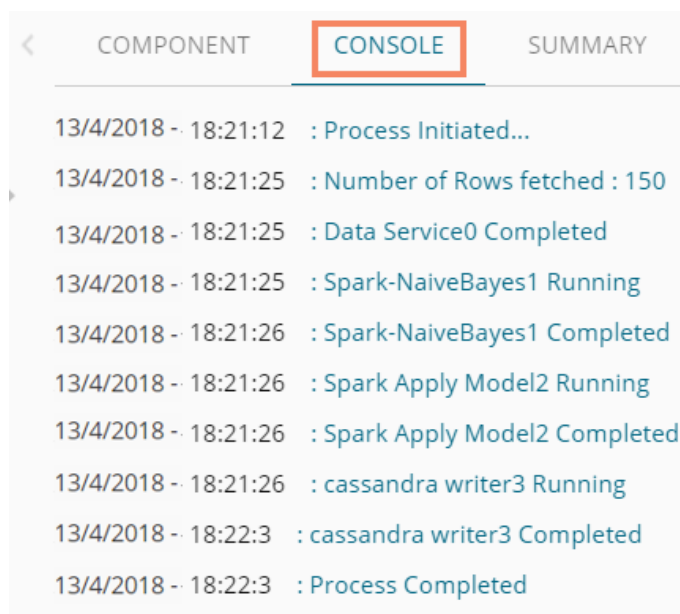
Note: Users will be provided with some defined consistency level while designing the KeySpace which can be overridden based on the selected replica nodes. Users are provided with the following consistency options:

- One
- Two
- Three
- Quarum

or

b. Selecting an Existing Table as Table Operation

i) Connect the '**Cassandra Writer**' to a configured data source.
ii) Click the '**Cassandra Writer**' component to access it.
iii) Configure the following **Properties** details
    i. **Select Data Connector**: Select a data connector from the drop-down menu
    ii. **Host Name**: Enter database server details (from where the user wants to fetch data)
    iii. **Port Name**: The server port number
    iv. **Username**: Username of the selected connection appears by default (Users cannot edit this field)
    v. **Password**: the database password
    vi. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
    vii. **Select Key Space**: Select a keyspace using the drop-down menu
    viii. **Replication Factor**: Replication factor in the selected '**Key Space**' will be displayed (Users cannot edit this field)
    ix. **Select Table**: Select a table from the drop-down menu
    x. **Choose Columns**: Select columns from the drop-down menu that users want to be written in the data writer.
    xi. **Consistency**: Select an option using the drop-down menu
    xii. **Settings**: Select an option using the drop-down menu
      The following choices will be provided:
      1. Append Table
      2. Overwrite Table

xiii. The list of column headers existing in the table will be displayed once users select a table.

iv) Configure the Partition Key and Clustering Key using the '**Key Specification**' option
v) Click '**APPLY**'

vi) After getting success message run the Workflow
   a. A message will pop-up to confirm whether users want to enable logging
   b. Click '**NO**'

vii) Users will get the process status under the '**CONSOLE**' tab

13/4/2018 - 18:21:12 : Process Initiated...

13/4/2018 - 18:21:25 : Number of Rows fetched : 150

13/4/2018 - 18:21:25 : Data Service0 Completed

13/4/2018 - 18:21:25 : Spark-NaiveBayes1 Running

13/4/2018 - 18:21:26 : Spark-NaiveBayes1 Completed

13/4/2018 - 18:21:26 : Spark Apply Model2 Running

13/4/2018 - 18:21:26 : Spark Apply Model2 Completed

13/4/2018 - 18:21:26 : cassandra writer3 Running

13/4/2018 - 18:22:3 : cassandra writer3 Completed

13/4/2018 - 18:22:3 : Process Completed
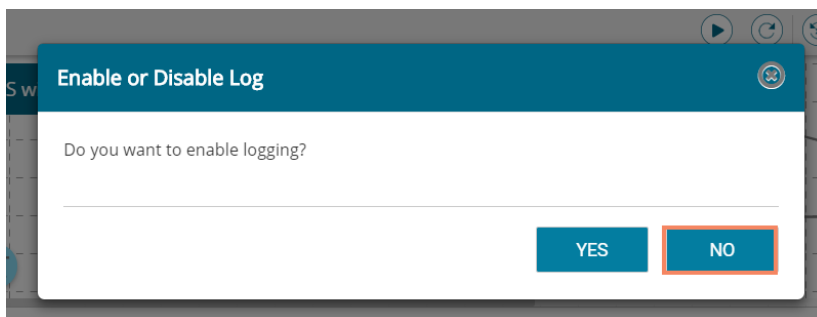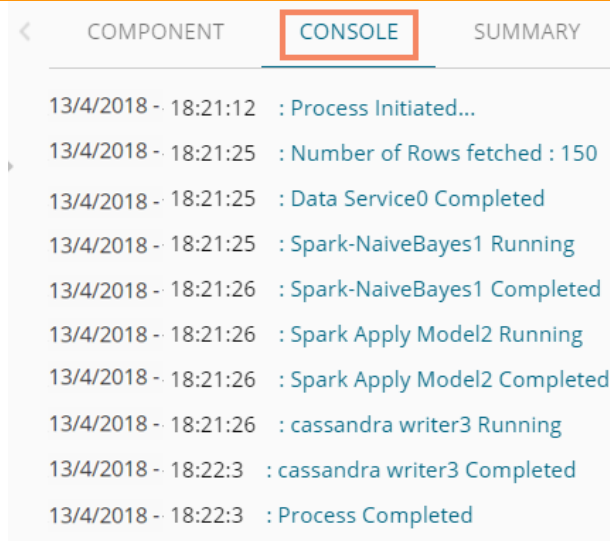
viii) The data will be saved in the selected Cassandra Writer

6.8. Custom Scala Script

Users can create and add customized algorithm components using the '**Custom Scala Script**' component. The created scripts will be stored in the '**Saved Scripts**' module provided for the Scala Scripts. The '**Custom Scala Script**' component will run only on Spark.

6.8.1. Creating a New Scala Script

i) Click '**Custom Scala Script**' tree-node on the Predictive Analysis home page.
ii) Click '**Create New Script**' option
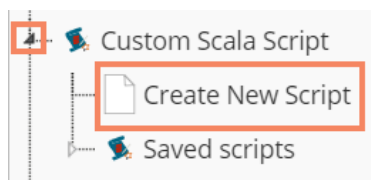
iii) Users will be directed to the '**COMPONENT**' tab
iv) Configure the following fields in the '**General**' tab:
   a. **Basic**
      i. **Component Name:** Enter a name or title that you wish to give a saved Scala Script.
      ii. **Component Type:** Default Component type will be displayed in this field.
      iii. **Description:** Describe the Component (It is an optional field).
v) Click '**NEXT**'
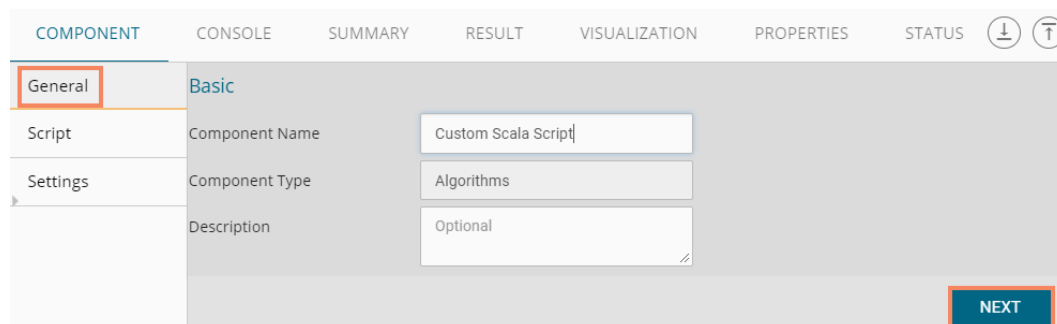
vi) Users will be directed to the **'Script'** tab
vii) Provide the following information:
  a. Script Editor
     i.   Write the Scala script in the given space
     ii. Click the **'Validate'** option
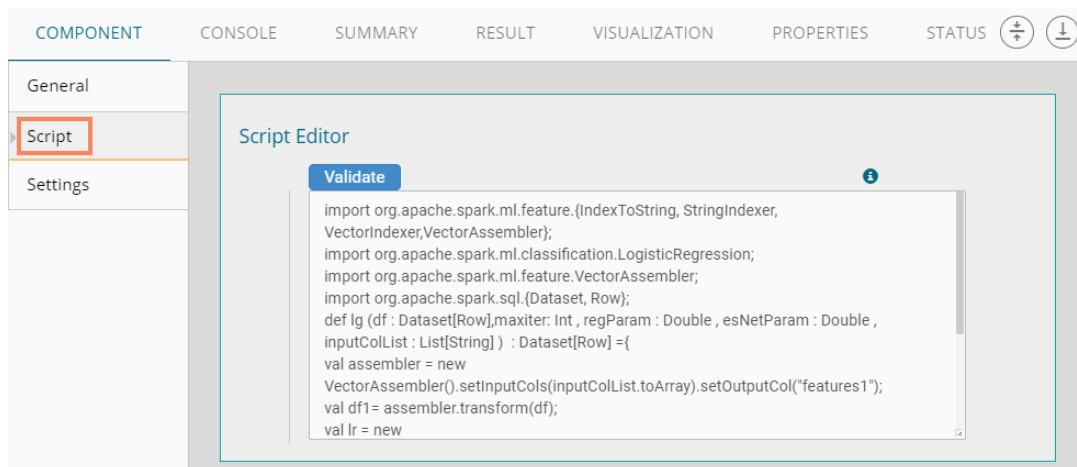
  iii. Configure the required **'Primary Function Details'** to embed the customized Scala script into a function.
     1. **Primary Function Name:** Select a name for the created function from the drop-down menu.
     2. **Input Data Frame:** Select a dataset (that has been used above) from a drop-down menu.
viii) Click **'NEXT'** (Users can click **'Previous'** if wish to open the previous page)

ix) Users will be directed to the **'Settings'** tab**.**
x) Configure the following fields:
  a. Output Table Definition
    This option will configure a number of output columns, column headers, data types. Select any one out of the following options:
     i.   **Consider all columns from the previous component:** To display all columns from the previous component.
     ii. **Consider None:** To display no column from the previous component.
  b. Define Output Columns

i. **Output Column Name**: Enter an appropriate name for the new predicted column.

ii. [−] : To remove the added row containing '**Data Type**' and '**New Predicted Column Name.**'

iii. [+] : To add a new row containing '**Data Type**' and '**New Predicted Column Name.**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

General

▸ Script

Settings

Output Table Definition

○ Consider all columns from previous component   ● Consider None

Define Output Columns

Output Column Name   ⓘ

[PredictedValues]   [−] [+]

c. **Property View Definition**

 i. **Function Parameters**: Actual names of parameters configured in the script.

 ii. **Property Display Name**: Parameter name to be displayed while configuring saved Scala script as a component.

 iii. **Control Type:** User can select out of the following options:
 1. Text box,
 2. Drop-down menu,
 3. Column Selector (single),
 4. Column Selector (multiple).

 iv. **Settings option** [⚙] : To set display for mandatory fields and validate **the** data type for the input column. This field is associated with function parameters.

xi) Click '**APPLY**'

Property View Definition

Function Parameters	Property Display Name	Control Type	
maxiter:Int	maxiter:Int	TextBox ▾	⚙
esNetParam:Double	esNetParam:Double	TextBox ▾	⚙
inputColList:List[Strin	inputColList:List[Strin	TextBox ▾	⚙
regParam:Double	regParam:Double	Column selector(N ▾	⚙

[PREVIOUS]   [APPLY]
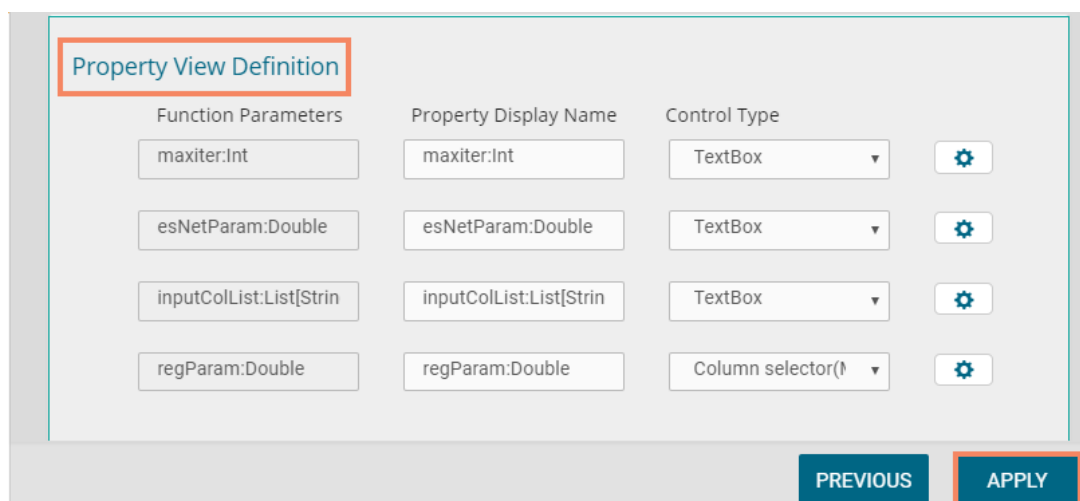
xii) A message will pop-up to notify that the newly created Scala script has been saved successfully

xiii) The newly created Scala script will be added to the '**Saved Scripts**' list
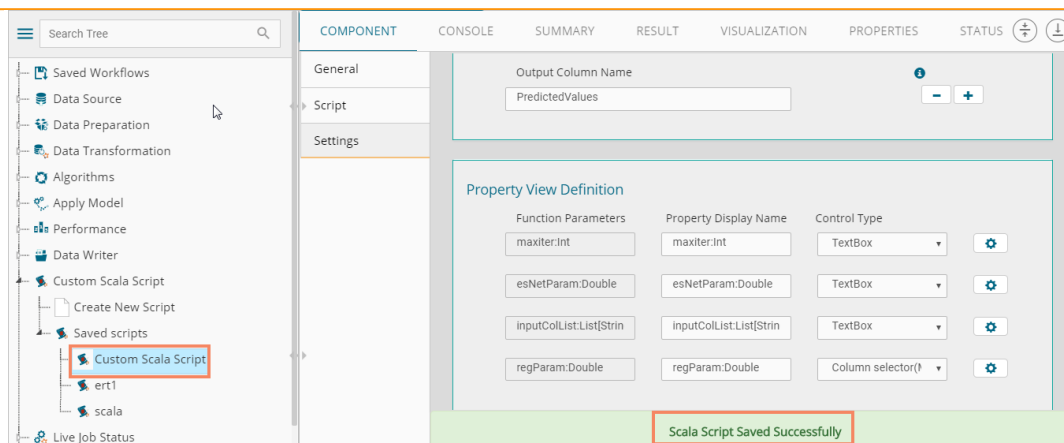
Guidelines for Writing a Scala Script

1. The First argument of the function should be a data frame.
2. The Scala script needs to be written inside a valid Scala function. E.g., the entire code body should be inside the curly braces of the function.
3. The Scala script should have at least one main function. Multiple functions are acceptable, and one function can call another function, but it should be written above the calling function body (if the called function is an outer function) or above the calling statement (if the called function is an inner function).
4. All the packages used in function need to import explicitly before writing function. # import org.apache.spark.sql. {Dataset, Row}.
5. The Scala script should return data in the form of a data set only and should define while writing function.
6. The column names should remain the same while creating new columns in the Output Table Definition.
7. If users need to define column selector (Multiple), then by definition ': List[String]' should be used and body of the function should be in 'to Array.'
8. If users need to define column selector (Single), then 'String' has to be used in the definition.

Note:

a. Click the '**Information**' button 🛈 to get the rules to write a Scala script.
b. All the supported date data types are listed in date formats in data type definition, all other date formats are considered as string data type.
c. Mssql data types are considered as string data type.

6.8.2. Saved Scala Scripts

6.8.2.1. Viewing a Saved Scala Script

i) Select a Scala Script from the '**Saved Scripts**' list.
ii) Right-click on the selected Scala Script.
iii) A context menu will open.
iv) Select the '**View**' option.
v) Users will be redirected to the '**Component**' tab.
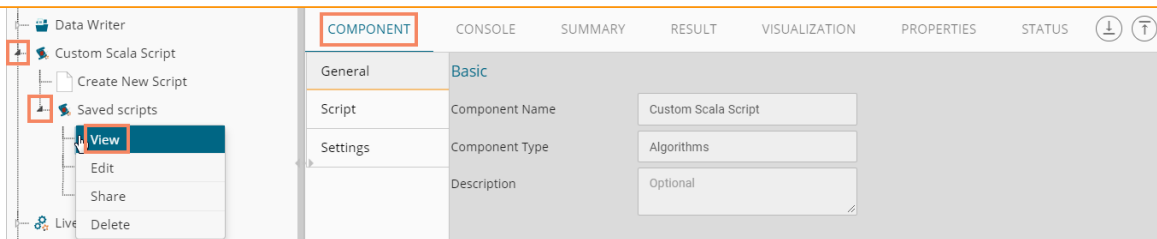
6.8.2.2. Editing a Saved Scala Script

i) Select a Scala Script from the list of '**Saved Scripts**' list
ii) Right-click on the selected Scala Script
iii) A context menu will open
iv) Select '**Edit**'
v) Users will be redirected to the '**Component**' tab
vi) Users can edit the required fields provided under **General**, **Script**, and **Settings** tab
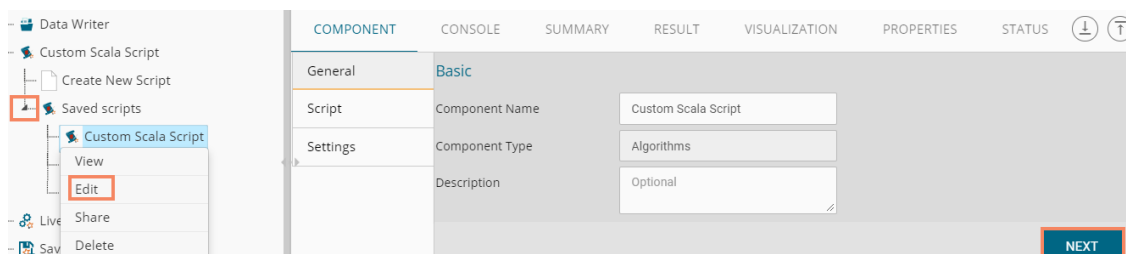
6.8.2.3. Sharing a Saved Scala Script

This feature gives users the ability to share a custom Scala script with other users and groups.

The following options are available to share a custom R script:

1. **Share With**: This option allows the user to share a custom Scala script with selected users or user groups. Any changes made to the custom Scala script will be transferred to all the users with whom the custom Scala script has been shared.

   i) Select a Scala script from the list of '**Saved Scripts**' tree-node
   ii) Right-click on the selected Scala script and select 'Share' option from the context menu
   iii) The '**Share With**' option will be displayed (by default)
   iv) Select either '**Group**' or '**Users**'
   a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
   b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
   v) Select a specific user or group from the list by check marking the box
   vi) Click '**APPLY**'

vii)   The selected Scala script will be shared with the chosen user(s)/group(s).

2.   **Copy To**: This option creates a copy and shares the copy of the custom Scala script with the selected users and user groups. Any changes to the original custom Scala script after sharing will not show up for the users that received the shared file via the '**Copy To**' option.

i)   Select a Scala script from the list of '**Saved Scripts**' tree-node
ii)   Right-click on the selected Scala script
iii)   Select '**Share**' from the context menu
iv)   Select '**Copy To**' option
v)   The copied custom Scala script name will be displayed in a box
vi)   Select either the '**Group**' or '**Users**' tab
    a.   By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
    b.   Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
vii)   Select a specific group or user from the list by check marking the box
viii)   Click '**APPLY**'
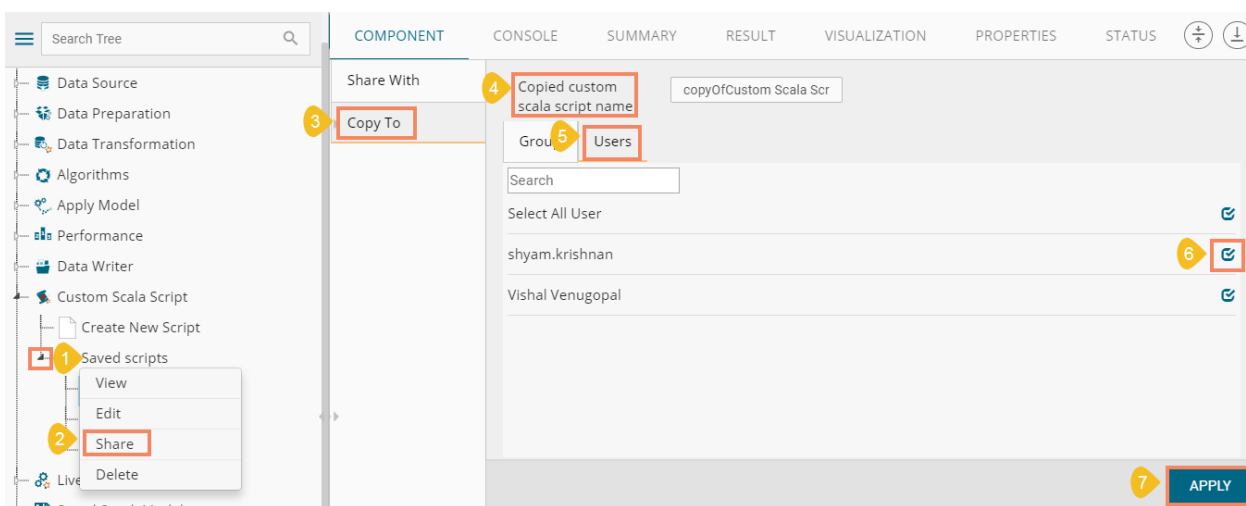
ix) The copied Scala script will be shared with the selected user(s)/group(s).

6.8.2.4.   Deleting a Saved Scala Script

i)   Select a Scala Script from the '**Saved Scripts**' list
ii)   Right-click on the selected Scala Script
iii)   A context menu will open
iv)   Select '**Delete**' option
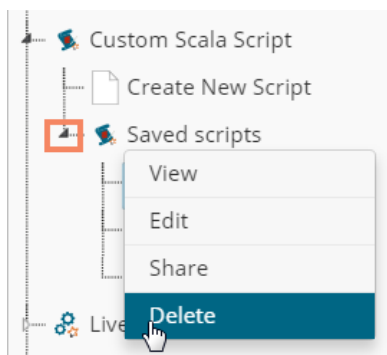
v)   A pop-up window will appear to assure the deletion
vi)   Click '**OK**'
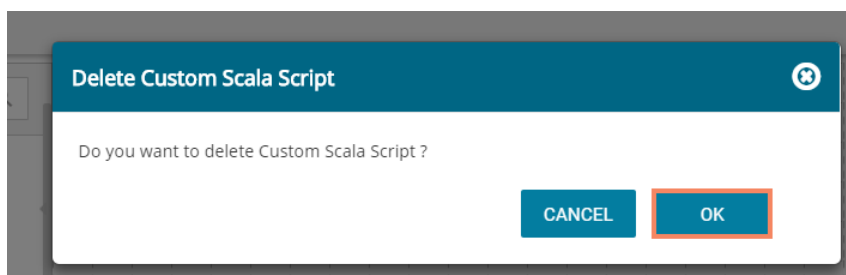
vii) The selected Scala Script will be deleted

6.8.2.5. Connecting Saved Scala Script with a Data Source

i)   Click the '**Custom Scala Script**' tree node.
ii)   Select and drag a saved Scala script to the workspace.
iii) Connect the Scala Script to a configured data source (Here, the used workflow has String Indexer and Spark Apply Model components connected with the Scala script component).
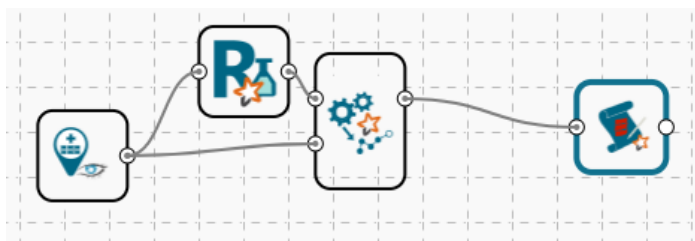
iv)   Click the dragged '**Scala Script**' component
v)   Configure the required fields in the '**Custom Group**' tab
vi)   Click '**APPLY**'
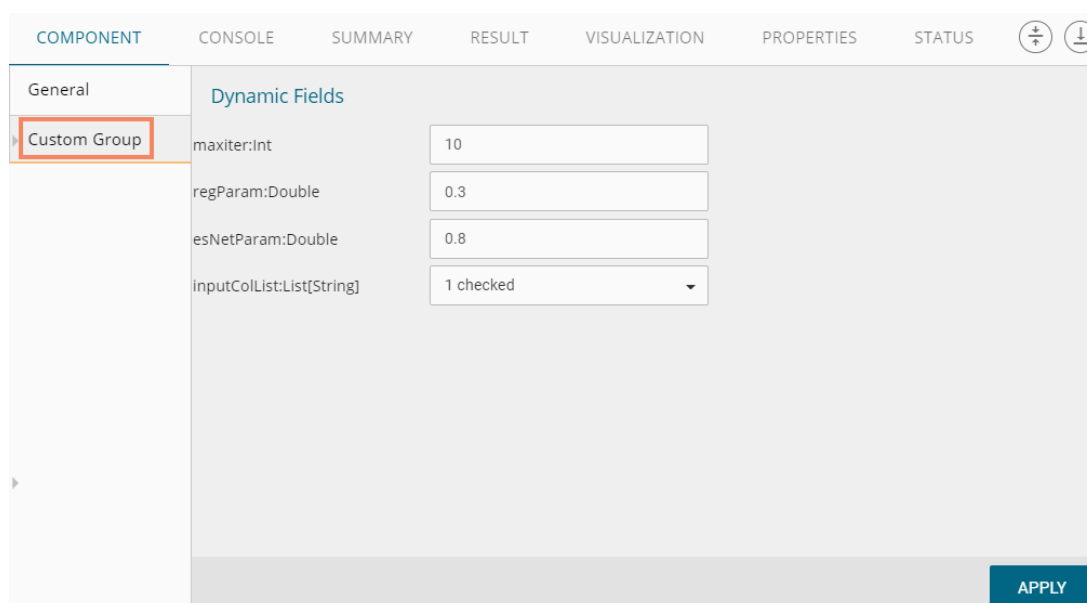
vii) After getting the success message run the workflow

a. A message will pop-up to confirm whether users want to enable logging
b. Select '**NO**'
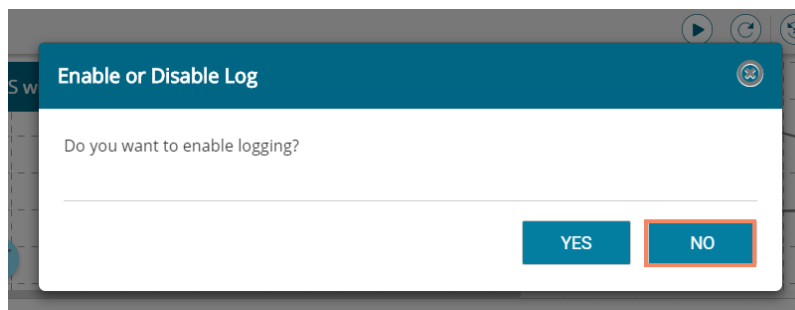
viii) Users will get the process status under the '**CONSOLE**' tab

12/4/2018 - 19:7:0	: Process Initiated...
12/4/2018 - 19:7:2	: Process started
12/4/2018 - 19:7:2	: Data Service0 Running
12/4/2018 - 19:7:13	: Number of Rows fetched : 150
12/4/2018 - 19:7:13	: Data Service0 Completed
12/4/2018 - 19:7:13	: Spark RFormula1 Running
12/4/2018 - 19:7:13	: Spark RFormula1 Completed
12/4/2018 - 19:7:13	: Spark Apply Model2 Running
12/4/2018 - 19:7:14	: Spark Apply Model2 Completed
12/4/2018 - 19:7:14	: ert1 Running
12/4/2018 - 19:7:16	: ert1 Completed
12/4/2018 - 19:7:16	: Process Completed

ix) Follow the below given steps to display the result view:
a. Click the dragged Spark Apply Model component on the workspace
b. Click the '**RESULT**' tab

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	label	features	prediction
36	1.2	0.2	5	3.2	0	0	{"values":[1.2,0.2]}	0
129	5.6	2.1	6.4	2.8	1	1	{"values":[5.6,2.1]}	1
89	4.1	1.3	5.6	3	1	1	{"values":[4.1,1.3]}	1
4	1.5	0.2	4.6	3.1	0	0	{"values":[1.5,0.2]}	0
61	3.5	1	5	2	1	1	{"values":[3.5,1]}	1
25	1.9	0.2	4.8	3.4	0	0	{"values":[1.9,0.2]}	0
47	1.6	0.2	5.1	3.8	0	0	{"values":[1.6,0.2]}	0
76	4.4	1.4	6.6	3	1	1	{"values":[4.4,1.4]}	1
87	4.7	1.5	6.7	3.1	1	1	{"values":[4.7,1.5]}	1
101	6	2.5	6.3	3.3	1	1	{"values":[6,2.5]}	1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

6.9. Live Job Status

Users can monitor spark processes using the '**Live job Status**' feature. The '**Live Job Status**' option will be a new tree node on the existing tree structure, and Spark will be a leaf node to the new tree node. Users need to enable logging to view the log in live job status in Spark after running a workflow.

i) Create a workflow in Spark
ii) Configure it and after getting success message run the workflow
iii) A window will pop-up asking confirmation to enable or disable log.
iv) Click '**YES**' to enable logging. (Selecting '**No**' will not display the log in the live job status.)
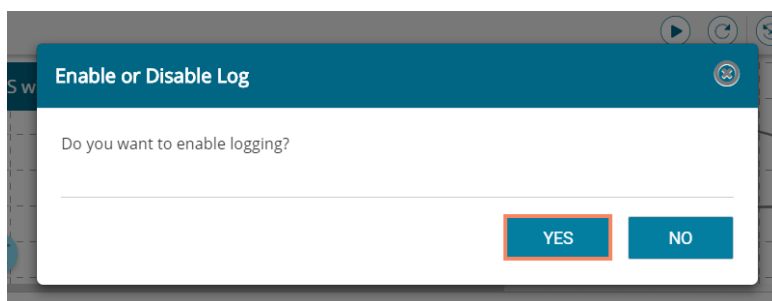
v) Click the '**Live Job Status**' tree node from the tree structure menu
vi) Click the '**Spark**' leaf node
vii) Users will be redirected to the '**STATUS**' tab

a. **View Log**: log of the completed workflow can be viewed under the '**CONSOLE**' tab by clicking the '**View Log**' icon 👁.

b. **Live Job Status:** If the workflow execution is still in progress, users can view live action by clicking the '**Live Job Status**' icon 👁. Live jobs will be displayed under the '**CONSOLE**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT

17/8/2017 – 11:46:44 : Job Id-442 : 220 tasks completed out of 295 with 0 failed task

17/8/2017 – 11:46:44 : Job Id-442 : 220 tasks completed out of 295 with 0 failed task

17/8/2017 – 11:46:44 : Job Id-443 : 0 task completed out of 285 with 0 failed task

17/8/2017 – 11:46:44 : Job Id-443 : 10 tasks completed out of 285 with 0 failed task

17/8/2017 – 11:46:44 : Job Id-443 : 10 tasks completed out of 285 with 0 failed task

17/8/2017 – 11:46:44 : Spark-ALS5 Completed

17/8/2017 – 11:46:45 : Spark-ALS8 Running

17/8/2017 – 11:46:45 : Job Id-444 : 0 task completed out of 63 with 0 failed task

17/8/2017 – 11:46:45 : Job Id-444 : 24 tasks completed out of 63 with 0 failed task

17/8/2017 – 11:46:45 : Job Id-444 : 36 tasks completed out of 63 with 0 failed task

c. **Summary:** Click the '**Summary**' icon 👁 to view a consolidated summary of all the components in a workflow. It will be displayed under the '**SUMMARY**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

----------- Summary of the model -----------

Impurity = gini
maxBins = 32
maxDepth = 5
labelCol = binarycolumn
featuresCol = dfFeaturesCol1
seed = 12
minInfoGain = 0.0
minInstancePerNode = 1

----------- End of Summary ----------

d. **Actions**
   i. **Stop:** Users can stop an ongoing execution at any time by clicking on the stop button. The status of the process will change to '**Cancelled**' if the execution has been stopped.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

🔄 Refresh ✖ Remove all jobs   Search: _____

Workflow Name	Run by	Start time	End Time	Status	View Log	Live job status	Summary	Actions
untitled		8/Aug/2018-17:11:46	8/Aug/2018-17:11:48	success	👁	👁	👁	■ 🗑
untitled		1/Aug/2018-12:54:31	1/Aug/2018-12:54:34	success	👁	👁	👁	■ 🗑
untitled		9/July/2018-14:56:35	9/July/2018-14:56:38	cancelled	👁	👁	👁	■ 🗑
wtfinal		21/Mar/2018-15:56:9	NA	in progress	👁	👁	👁	■ 🗑
wtfinal		21/Mar/2018-15:53:55	NA	in progress	👁	👁	👁	■ 🗑

Showing 11 to 15 of 15 entries   Previous 1 2 Next

   ii. **Delete:** Click the '**Delete**' icon to remove an execution.

Workflow Name	Run by	Start time	End Time	Status	View Log	Live job status	Summary	Actions
untitled		8/Aug/2018-17:11:46	8/Aug/2018-17:11:48	success	👁	👁	👁	■ 🗑
untitled		1/Aug/2018-12:54:31	1/Aug/2018-12:54:34	success	👁	👁	👁	■ 🗑
untitled		9/July/2018-14:56:35	9/July/2018-14:56:38	failed	👁	👁	👁	■ 🗑
wtfinal		21/Mar/2018-15:56:9	NA	in progress	👁	👁	👁	■ 🗑
wtfinal		21/Mar/2018-15:53:55	NA	in progress	👁	👁	👁	■ 🗑

Showing 11 to 15 of 15 entries   Previous   1   2   Next

The selected workflow will be removed from the '**Live Job Status**' table and a message will be displayed to convey the same.
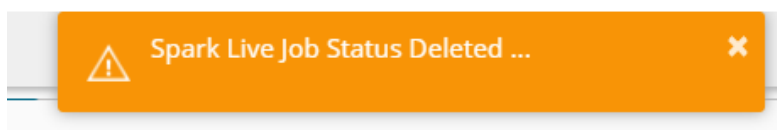
Note:

a. Click the '**Refresh**' option ⟳ Refresh to refresh the table for viewing a live job.
b. Click the '**Remove all jobs**' option ✖ Remove all jobs to delete all the jobs from the table.

6.10. Saved Workflows

Users can save a workflow by clicking the '**Save**' button provided on the workspace menu row. All the saved workflows will be displayed under the '**Saved Workflow**' tree node. This section explains various options assigned to a saved workflow.

i)   Navigate to the Predictive home page
ii) Click '**Saved Workflow**' tree-node
iii) A list of all the saved workflows will be displayed
iv) Right, click on a workflow from the list of '**Saved Workflows**'
v)   A context menu will open with various options (As shown below):
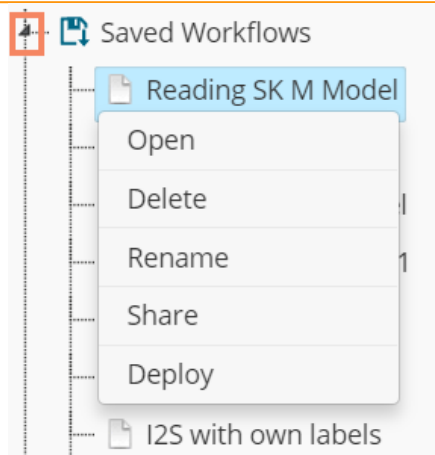
6.10.1. Opening a Workflow

    i)   Right-click on a workflow from the list of '**Saved Workflows**'

    ii)   Select '**Open**' from the context menu

    iii)   The selected workflow will be displayed in the right pane of the screen
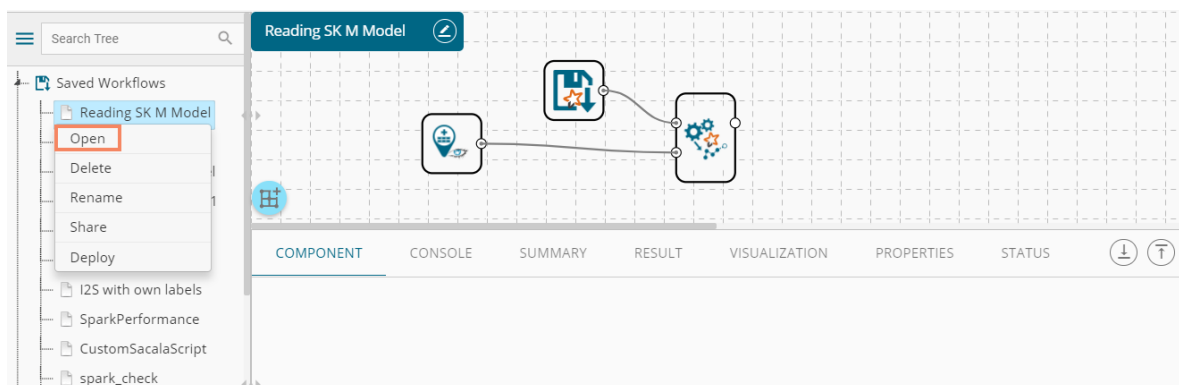
Note: The workflow name will be displayed on the left side of the workspace menu row while opening a workflow.
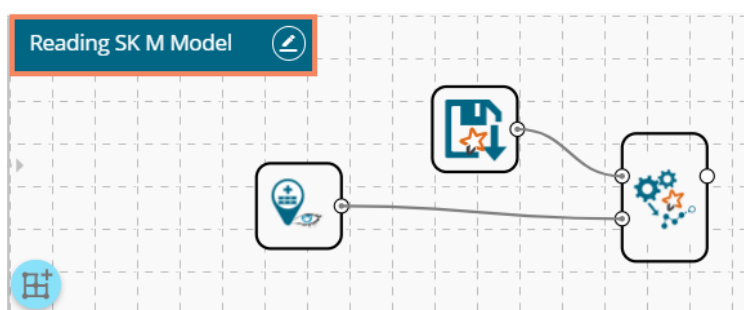
6.10.2. Deleting a Workflow

    i)   Right-click on a workflow from the list of '**Saved Workflows**'

    ii)   Select '**Delete**' from the context menu
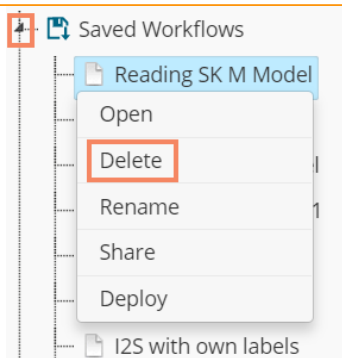
iii) A message window will pop-up to confirm the deletion
iv) Click '**OK**'
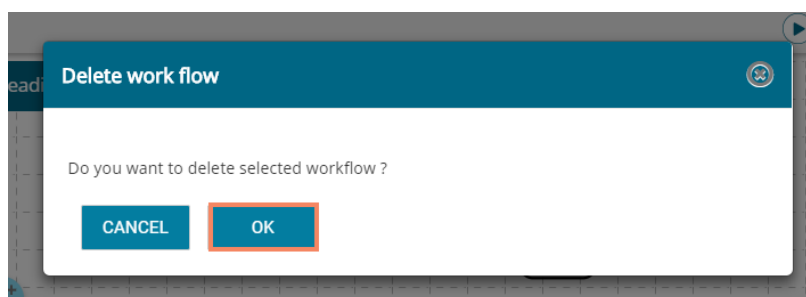
v) The selected workflow will be removed from the list

6.10.3. Delete Connection in a Workflow

A Right click on the inter-node connection will display the '**Delete Connection**' option in a workflow. Click the '**Delete Connection**' option to delete a connection.
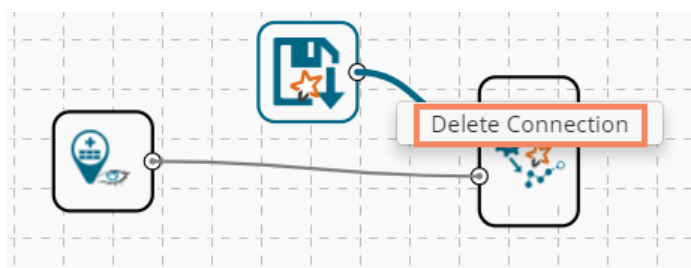
6.10.4. Renaming a Workflow

i) Press a right click on a workflow from the list of '**Saved Workflows**'
ii) Select '**Rename**' from the context menu
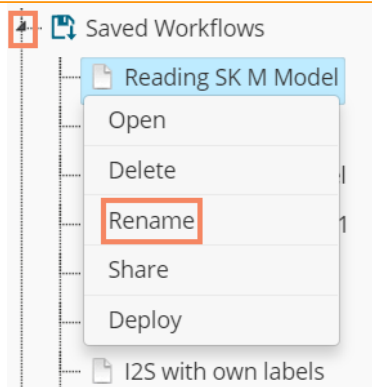
iii) A pop-up window will appear
iv) Enter a new/modified name for the workflow
v) Click 'YES'
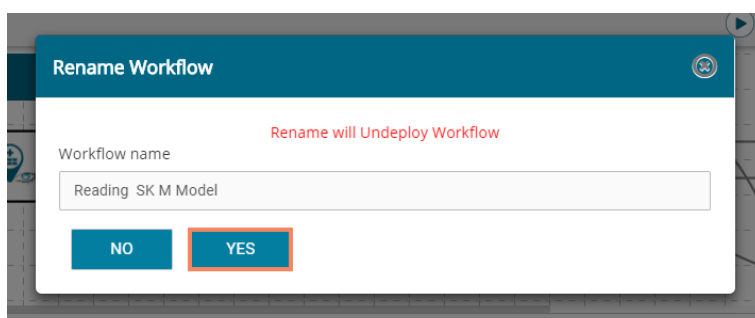
vi) The selected workflow will be renamed
Note: Renaming a deployed workflow will undeploy the workflow.

6.10.5. Sharing a Workflow
This feature gives users the ability to share saved workflows with other users and groups.

The following options are available to share a selected workflow:

3. **Share With**: This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.

   i) Press a right click on a workflow from the list of '**Saved Workflows**'
   ii) Select '**Share Workflow**' from the context menu
   iii) The '**Share With**' option will be displayed (by default)
   iv) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
      b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
   v) Select a specific group or user from the list by check marking the box
   vi) Click '**APPLY**'

vii) The selected workflow will be shared with the chosen user(s)/group(s)

4. **Copy To**: This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the '**Copy To**' method.

   i) Press a right click on a workflow from the list of '**Saved Workflows**'
   ii) Select '**Share Workflow**' from the context menu
   iii) Select '**Copy To**'
   iv) The copied workflow name will be displayed
   v) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
      b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected
   vi) Select a specific group or user from the list by check marking the box
   vii) Click '**APPLY**'

viii) The copied workflow will be shared with the chosen users/groups

6.10.6. Deploying a Workflow
The Predictive Workflows can be deployed to the BizViz Dashboard Designer.

i) Press a right click on a Workflow from the list of **'Saved Workflows'**
ii) Select **'Deploy'** from the context menu

iii) A success message will pop-up to assure that the workflow has been published

Workflow published successfully.
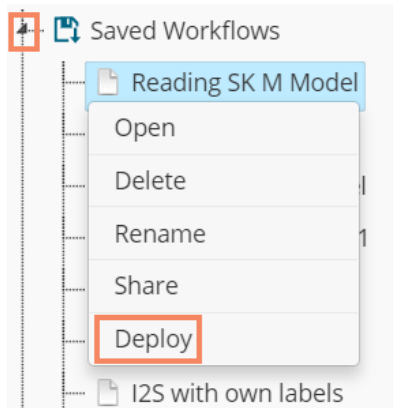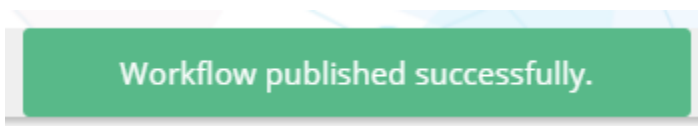
iv) The deployed workflows will be marked with a checkmark

v) Navigate to the Dashboard Designer home page
vi) Click **'New'**
vii) Click **'Dashboard'**

viii) Users will be directed to the Dashboard canvas
ix) Click the **'Data Source'** icon 🗄 to display all the available data sources
x) Click the **'Create New Connection'** option ＋ provided next to the **'Predictive Service'** data source

xi)   A new connection will be created and added below

xii)   Click on the connection to display the connection specific details
xiii) Select the deployed Predictive workflow as a data source via the drop-down menu

xiv)   Configure the other subsequent details:
       a. Load At Start: Enable this option to get the updated data
       b. Timely Refresh: Enable this option to refresh data
       c. Refresh Interval: Select the time interval to refresh the data

Name	Connection-1
Predictive Workflows	Reading SK M Model
Load At Start	Yes No
Timely Refresh	Yes No
Refresh Interval	5 Minute(s)

FIELD SET   CALCULATED FIELDS   CONDITION

cat

ClusterNumber

featuresCol1

Number

d. Once the data connection is established the selected predictive workflow can be used as a connection to the Dashboard Designer for fetching data

Recommendations

- **Spark Workflows:**
  - The result set from the '**Apply Model**' component within a deployed Spark workflow will be considered as a data set by the Dashboard Designer (a result set after the 'Apply Model' component will not be considered).
  - A Spark workflow must contain one Apply model, read model (Saved Model component), and Spark filter (optional) component to deploy the workflow.

  Note:

a. Users will be redirected to select an Apply Model component from the workflow
   Users will be asked to select an Apply Model component when the selected workflow contains two or more apply model components.
   i. Users need to select an Apply Model component
   ii. Click 'Yes'
b. If a deployed Predictive Workflow has a summary, it can be viewed using the Dashboard Designer tool.
c. Users can view the result of each component in a spark workflow, provided the component is not a pipeline component.
   i) Select a component from the spark workflow after the execution is completed
   ii) Click the '**Result**' tab
   iii) The result data of the selected component will be displayed

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	Label1	I2S_col
1.6	0.4	5	3.4	setosa	1	setosa
4.1	1	5.8	2.7	versicolor	2	versicolor
5.4	2.1	6.9	3.1	virginica	0	virginica

d. Users can stop an ongoing Spark workflow execution by clicking the '**Stop**' button on the progress bar.

6.11. Saved Spark Models

A model is a reusable component created by training an algorithm using historical data and saving the instance. The '**Saved Spark Models**' tree-node contains a list of all the saved predictive models.

6.11.1. Saving a Spark Model

i)   Open a spark workflow
ii)   Connect '**Apply Model**' component with the workflow (as shown below)
iii)   Right-click on the '**Apply Model**' component
iv)   A context menu will open
v)   Select '**Save Model**'
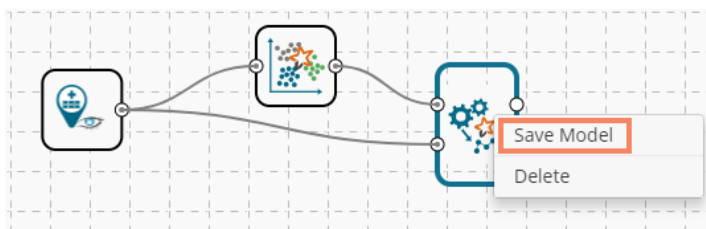
vi)   A pop-up window will appear
vii)   Enter a name for the model that you wish to save
viii)   Click '**OK**'
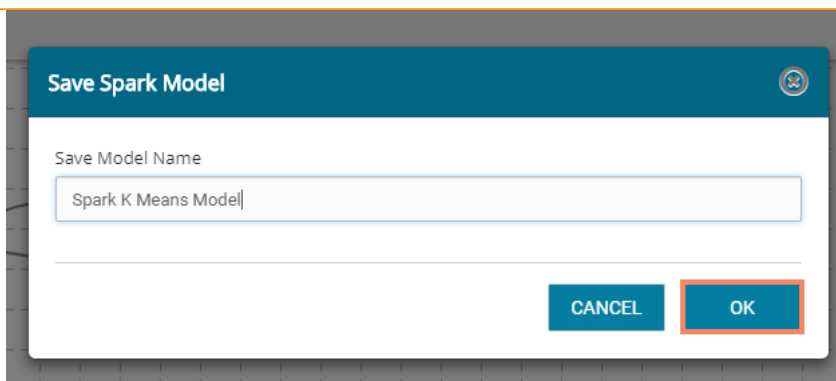
ix) A new message pops-up to confirm the action

Spark model saved successfully!
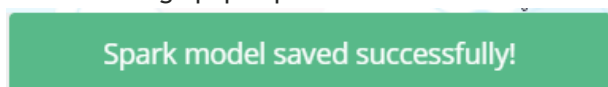
x) The created Predictive Model will be saved to the **'Saved Spark Models'** list

Saved Spark Models

Spark K Means Model

6.11.2. Reading a Spark Model

Users can drag a saved model to the workspace and reuse the model for test data. A saved model can be connected to only Apply Model and new test data source.

i) Select and drag a saved model onto the workspace

ii) Connect the saved model with a configured data source and an Apply Model component (As shown in the following image)
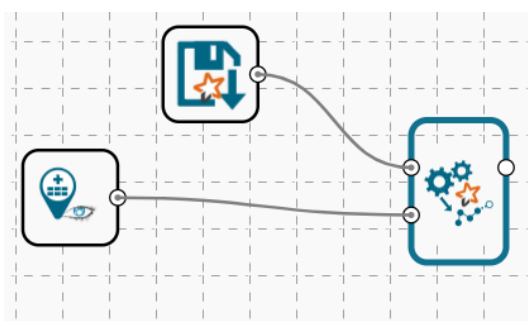
iii) Click on the dragged Saved Model component

iv) Users will be redirected to the component tab containing the following options:

a. The basic information of the saved model will be displayed by the 'General' section

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

Basic

General		
Summary	Component Name	Spark Saved Model
	Alias	Spark K Means Model1
	Description	Optional
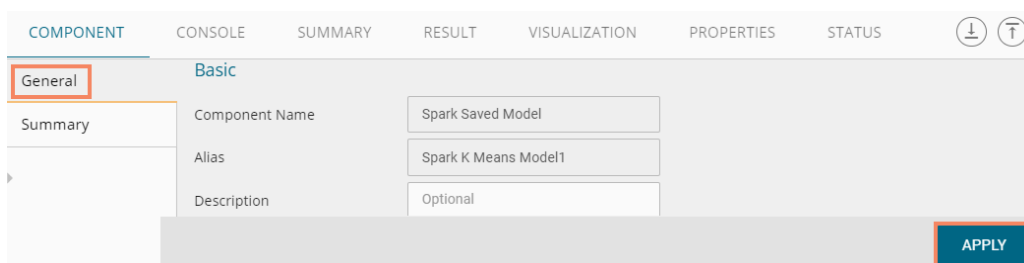
APPLY

b. Summary option displaying the summary of the model
c. Click 'APPLY'

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

General

Summary

Summary

------------ Summary of the model ------------

Columns used in the algorithm:
PetalLength (double)
PetalWidth (double)
SepalLength (double)
SepalWidth (double)
cat (double)
Cluster Centers = [1.4333333333333333,0.23030303030303034,4.818181818181818,3.2363636363636363,0.0],
[5.846875,2.13125,6.9125,3.0999999999999996,1.0],
[1.5176470588235293,0.2764705882352942,5.370588235294117,3.8,0.0],
[4.807317073170733,1.6219512195121948,6.236585365853657,2.858536585365854,1.0],
[3.9407407407407407,1.2185185185185186,5.52962962962963,2.622222222222222,1.0]
Within Set Sum of Squared Errors = 50.16408239949591

------------ End of Summary ----------

APPLY

d. Configure the '**Apply Model**' component by clicking the '**APPLY**' option

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | STATUS |

General

Basic

Component Name	Spark Apply Model
Alias	Spark Apply Model2
Description	Optional

APPLY

v)   After getting success message run the workflow
vi) Users will be redirected to the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY |

13/4/2018 - 16:27:56 : Process Initiated...

13/4/2018 - 16:27:58 : Process started

13/4/2018 - 16:27:58 : cassandra0 Running

13/4/2018 - 16:27:58 : Number of Rows fetched : 150

13/4/2018 - 16:27:58 : cassandra0 Completed

13/4/2018 - 16:27:59 : Spark-K-Means1 Running

13/4/2018 - 16:27:59 : Spark-K-Means1 Completed

13/4/2018 - 16:27:59 : Spark Apply Model2 Running

13/4/2018 - 16:27:59 : Spark Apply Model2 Completed

13/4/2018 - 16:27:59 : Process Completed

vii) Follow the below given steps to display Result.
      a. Click Apply model component.
      b. Click the '**RESULT**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

Show [10 ▾] entries   Search: [   ]

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat	featuresCol1	ClusterNumber
51	4.7	1.4	7	3.2	1	{"values":[4.7,1.4,7,3.2,1]}	3
46	1.4	0.3	4.8	3	0	{"values":[1.4,0.3,4.8,3,0]}	0
14	1.1	0.1	4.3	3	0	{"values":[1.1,0.1,4.3,3,0]}	0
31	1.6	0.2	4.8	3.1	0	{"values":[1.6,0.2,4.8,3.1,0]}	0
81	3.8	1.1	5.5	2.4	1	{"values":[3.8,1.1,5.5,2.4,1]}	4
90	4	1.3	5.5	2.5	1	{"values":[4,1.3,5.5,2.5,1]}	4
74	4.7	1.2	6.1	2.8	1	{"values":[4.7,1.2,6.1,2.8,1]}	3
10	1.5	0.1	4.9	3.1	0	{"values":[1.5,0.1,4.9,3.1,0]}	0
29	1.4	0.2	5.2	3.4	0	{"values":[1.4,0.2,5.2,3.4,0]}	0
55	4.6	1.5	6.5	2.8	1	{"values":[4.6,1.5,6.5,2.8,1]}	3

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

viii) Click the '**PROPERTIES**' tab to display the model properties.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	STATUS

Created By	
Created At	2018-04-09 14:36:23 +0530
Last Modified By	
Last Modified At	2018-04-13 15:40:35 +0530
Version	3.5.

Note:

a. Column headers and data type of the test data source must match the selected saved model to run the workflow with a 'Saved Model' component.
b. Users will encounter an error if validation fails while running the workflow.
c. Users can connect a data writer to the '**Apply Model**' component in a workflow that contains a saved model.
d. Currently, only Spark trained Workflows can be saved to the '**Saved Models**' tree-node.

6.11.2.1. Renaming a Spark Model

i)   Select a model from the '**Saved Models**' list
ii)   Right-click on the selected model
iii) A context menu will open
iv)   Select '**Rename**' from the menu
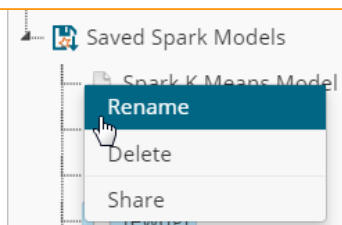
v) A pop-up window will appear to rename the model
vi) Enter a new **'Model Title'** or modify the existing model title in the given field (if desired)
vii) Click **'YES'**
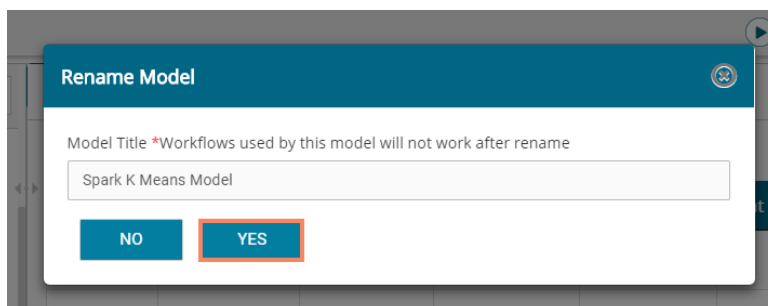
viii) The selected Spark Predictive Model will be renamed
Note: Workflows used by the model that has been renamed will not work after rename action is performed.

6.11.2.2. Deleting a Spark Model

i) Select a model from the '**Saved Models**' list
ii) Right-click on the selected model
iii) A context menu will open
iv) Select **'Delete'**
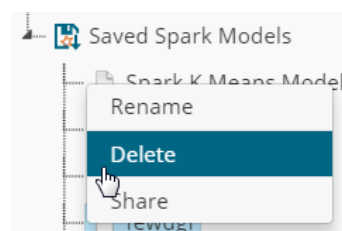
v) A pop-up window will appear to confirm the deletion
vi) Click **'OK'**
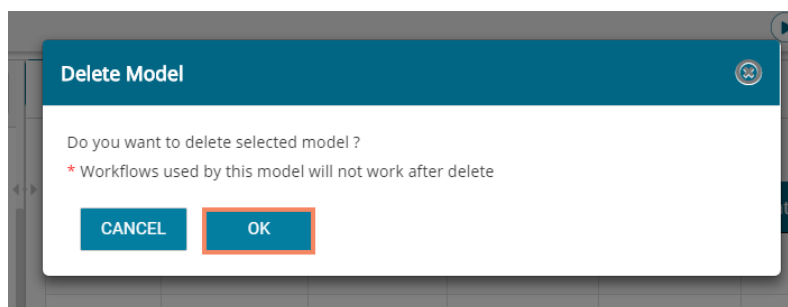
vii) The selected predictive model will be deleted and removed from the list of **'Saved Spark Models'**

Note: The workflows used by this model will not work after the model is deleted.

6.11.2.3. Sharing a Spark Model

Users can share a saved model with other users or user groups. There are two options to share a selected model:

1. **Share With:** This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.
    i)   Right, click on a model from the list of **'Saved Models'**
    ii) Select **'Share Model'** from the context menu
    iii) The **'Share With'** option will be displayed (by default)
    iv) Select either **'Group'** or **'Users'** option
         a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
         b. Users can be excluded by not selecting a username from the list when **'User'** option has been selected
    v)   Select a specific group or user from the list by check marking the box
    vi) Click **'APPLY'**

   vii) The saved Spark model will be shared with the selected group(s)/user(s)

2. **Copy To:** This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the **'Copy To'** method.
    i)   Use right-click on workflow from the list of **'Saved Models'**
    ii) Select **'Share Model'** from the context menu
    iii) Select **'Copy To'** option
    iv) The copied model name will be displayed
    v)   Select either **'Group'** or **'Users'** option with a click
         a.   By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
         b.   Users can be excluded by not selecting a username from the list when **'User'** option has been selected

    vi) Select a specific group or user from the list by check marking the box
    vii) Click **'APPLY'**
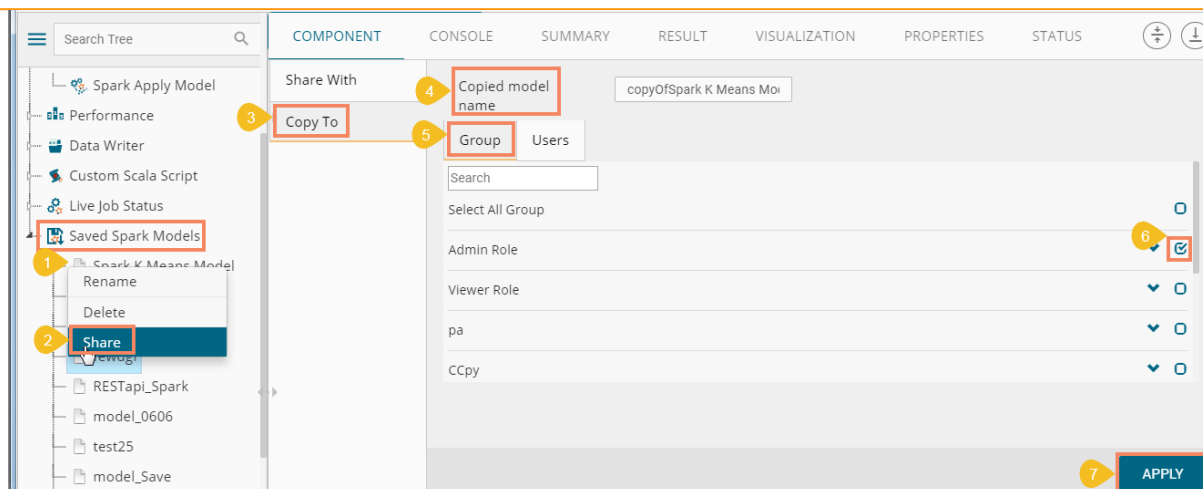
viii)   A copy of the model will be shared with the selected user or group

7. Python Workspace

Users can select the Python Workspace from the Predictive landing page to access the Python Environment under the Predictive Workbench.
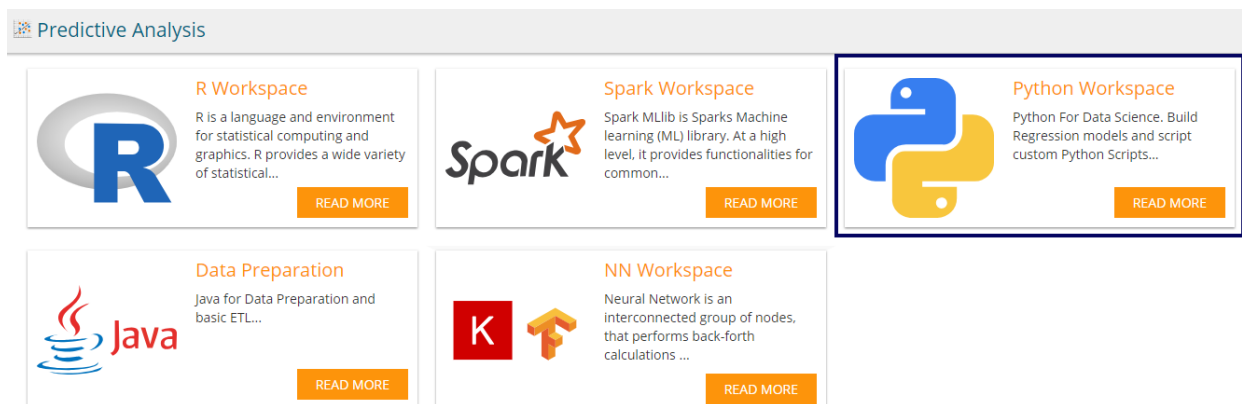
Users will be redirected to the following screen by selecting the Python Workspace:
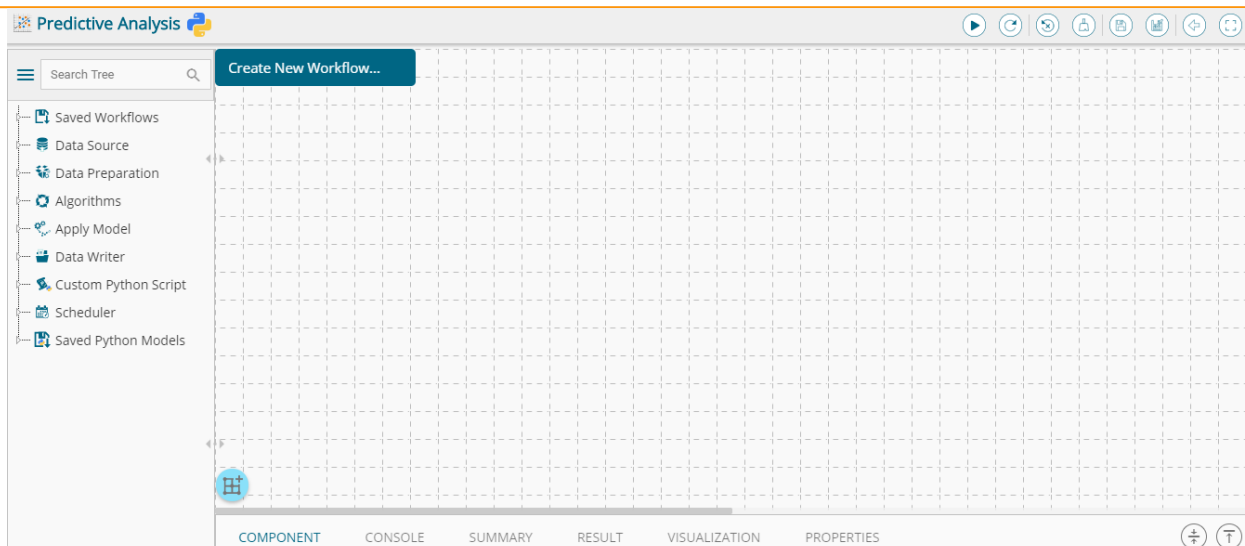
7.1. Getting Data from a Data Source

Acquiring data from a data source is the initial step in Predictive Analysis. The '**Data Source**' tree node offers three types of data connectors:

   a. CSV File
   b. Data Service
   c. Data Store Reader

7.1.1. Getting Data from a CSV File

   i)   Select and drag '**CSV File**' component onto the workspace
   ii)   Click the '**CSV File**' component

   iii)   Configure the following '**CSV Properties Configuration**' fields:
   a. **Select File**: Browse a CSV file
   b. **Delimiter:** Mention the delimiter used in the CSV file
   iv)   Click '**APPLY**'

v) Users should get the 'Apply Successful' message as displayed in the following image:

vi) Click the '**Run**' ⓟ icon or click '**Refresh**' ⓒ icon to run the workflow by clearing the previous cache

vii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix) Follow the below given steps to display the result view:
 a. Click the dragged data source component on the workspace
 b. Click the '**RESULT**' tab

RowID	SLength	SWidth	PLength	PWidth
1	5.1	3.5	1.4	0.2
2	4.9	3	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

Showing 1 to 10 of 150 entries

- **Rules to be followed while uploading a CSV File**
  1. The first row provided in the CSV file should contain the column headers.
  2. The second row of the CSV file should contain the data under all the headers without any 'null' or 'NA.'
  3. CSV headers should not have space. It should be a single word or two words concatenated by an underscore (_).
  4. CSV headers should not contain any special characters. E.g. - %, #, $, @,*, etc.
  5. CSV headers should not contain single or double quotes, dot, brackets, and high-fen.
  6. CSV headers should not contain merely numbers. Numerals should be used with at least one alphabet.
  7. CSV header should not exceed 50 characters.
  8. All rows in a column should have the same data type.

  Note:
  a. The supported file types will be .csv, .tsv
  b. '**General**' tab is provided to configure the following information for any tree-node component:
     i. Component Name: The predefined name of the component is displayed in this field
     ii. Alias Name:
     iii. Description (it is an optional field)
        (E.g. the following image displays '**General**' tab for a CSV data source.)

7.1.2. Getting Data from a Data Service

i) Select and drag '**Data Service**' component onto the workspace.

ii) Click the '**Data Service**' component.

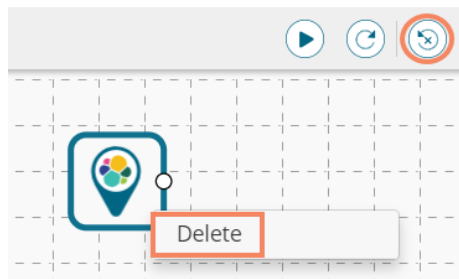iii) Users will be redirected to the '**Properties**' fields provided under '**Components**' tab on the Tabbed Menu Strip.

iv) Configure the '**Data Service Properties**':

    a. **Select Data Connector**: Select a data source from the drop-down menu

    b. **Select Data Service**: Select a query service from the drop-down menu

    c. **Fields**:

        The following tables will be displayed:

        i. Column Header

        ii. Data Type

v) Click '**NEXT**' (The '**NEXT**' option will appear only for the data service that has filters, otherwise the '**APPLY**' option will be displayed)

vi) Users will be redirected to the '**Conditions**' tab. (If the selected data service contains the filter values).

vii) Configure the following information:

    a. **Filter Type:** Available filter(s) in the data service will be displayed in this space.

    b. **Control Type:** Users are provided with the following options to pass the filter values under this option:

      • **Text:** By selecting this option users can manually enter multiple filter values separated by comma

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

	Filter Name	Control Type	
General			
Properties	val1	Text ▾	Sepal Length
Conditions			

APPLY

- **LOV:** By selecting this filter value option users will be directed to **choose** another Data Connector and Data Service available in the space

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

	Filter Name	Control Type
General		
Properties	val1	LOV ▾
Conditions	Select Data Connector	Select ▾
	Select Data Service	Select ▾

APPLY

viii)   Click **'APPLY'**

ix)   Click the '**Run**' ▶ icon or click '**Refresh**' ⟳ icon to run the workflow by clearing the previous cache

x)   Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

COMPONENT	CONSOLE	SUMMARY

13/4/2018 - 11:43:15   : Process Initiated...

13/4/2018 - 11:43:16   : Data Service0 is started.

13/4/2018 - 11:43:17   : Data Service0 is completed.

xi)   After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii)   Follow the below given steps to display the result view:
  a. Click the dragged data source component on the workspace
  b. Click the '**RESULT**' tab

| | COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | |

id	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.1	3.6	1.4	0.2	setosa
6	5.1	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Show 10 entries   Search:

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

- **Rules to be Followed while Creating a Data Service**
  1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
  2. Data service header should not contain any special characters. E.g. - %, #, $, @,*, etc.
  3. Data service header should not contain single or double quotes, dot, brackets, and high-fen.
  4. Data service header should not contain merely numbers. Numerals should be used with at least one alphabet.
  5. Data service header should not exceed 50 characters.

  Note:
     a. Users can develop a data service via the Data Management module of the BizViz Platform.
     b. **The 'Fields'** option under the '**Properties**' tab will appear only after selecting the appropriate query service.
     c. LOV service provided under the '**Conditions**' tab can contain only one column, in case of more than one column, a warning message will appear.
     d. Users can configure the following information for a data service data source via the '**General**' tab:
        i. Alias Name
        ii. Description (it is an optional field)

7.1.3. Getting Data from a Data Store Reader

i)   Select and drag '**Data Store Reader**' component onto the workspace
ii)   Click on the '**Data Store Reader**' component

iii)   Users will be redirected to the '**Properties**' tab of the component

iv) Configure the required properties:
   a. Select Data Store: Select a data store using the drop-down menu
   b. Limit No. of Documents to Fetch: Select an option using the drop-down menu. Two options will be provided as shown below:
      1. Fetch all Documents
      2. Limit By
   c. Max. No. of Documents to be Fetched: Enter a number to decide maximum fetched documents (This option will appear only if 'Limit By' option has been selected using the 'Limit No. of Documents to Fetch' field. Users can select any positive integer value).
v) Click '**NEXT**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

General	**Data Store Reader Properties**		
Properties	Select Data Store	iris	▼
Conditions	Limit No. of Documents to Fetch	Limit By	▼
	Max No. of Documents to be Fetched	10	

Fields

Column Header	Field Definition	Data type
SepalLength	measure	double
SepalWidth	measure	double
category	dimension	string

NEXT

vi) Users will be redirected to the '**Conditions**' tab
vii) Select the required columns from the drop-down list
viii) Click '**APPLY**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

General	**Column Selection**	
Properties	3 checked	▼
Conditions		

APPLY

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache
x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

COMPONENT	**CONSOLE**	SUMMARY	RESULT

13/4/2018 - 11:22:28   : Process Initiated...

13/4/2018 - 11:22:29   : Data Store Reader0 is started.

13/4/2018 - 11:22:30   : Data Store Reader0 is completed.

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:
    a. Click the dragged data source component on the workspace
    b. Click the '**RESULT**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES	

Show [10 ▾] entries   Search: [ ]

SepalLength	SepalWidth	category
1	-0.32251082	SepalLength
-0.32251082	1	SepalWidth

Showing 1 to 2 of 2 entries   Previous | 1 | Next
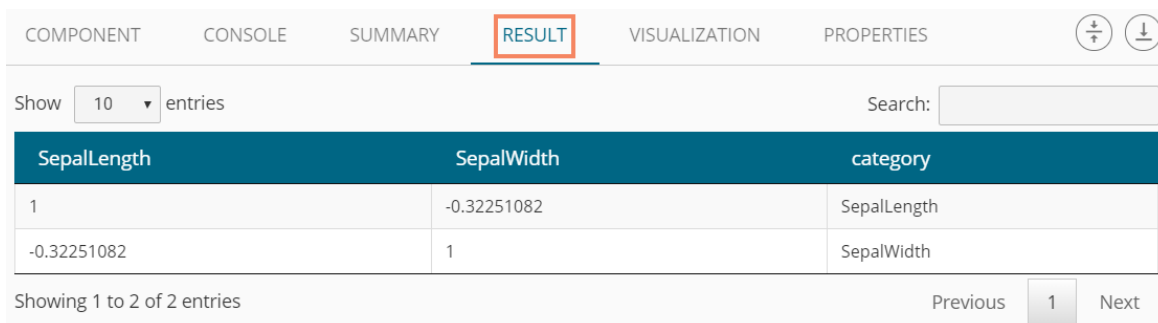
7.1.4. Removing a Data Source from the Workspace

i) Right-click on the data source connector (in the workspace)
ii) A context menu appears
iii) Click the '**Delete**' option
iv) The selected Data Source component will be removed from the workspace

OR

Click on the '**Reset**' icon to remove the connector(s) from the workspace

Note: The same set of steps can be followed to remove any data source type in the given tree-node menu.

7.2. Data Preparation

7.2.1. Missing Value Replacement Python

Users can replace the missing data in the specified variable with the determined value using the Missing Value Replacement Python component as well. Users will be provided with a list of options that can be considered for replacement.

i) Drag a data source on the workspace, configure it, run it, and check the data using the '**Result**' tab (in this case, the selected input data is displayed in the following image).

ii) Select and drag the '**Missing Value Replacement Python**' component onto the workspace.

iii) Connect the '**Missing Value Replacement Python**' component to a configured data source and use the Right-click to configure it.
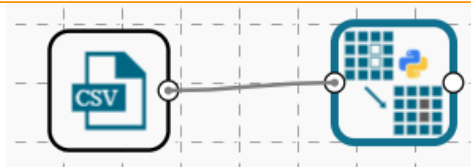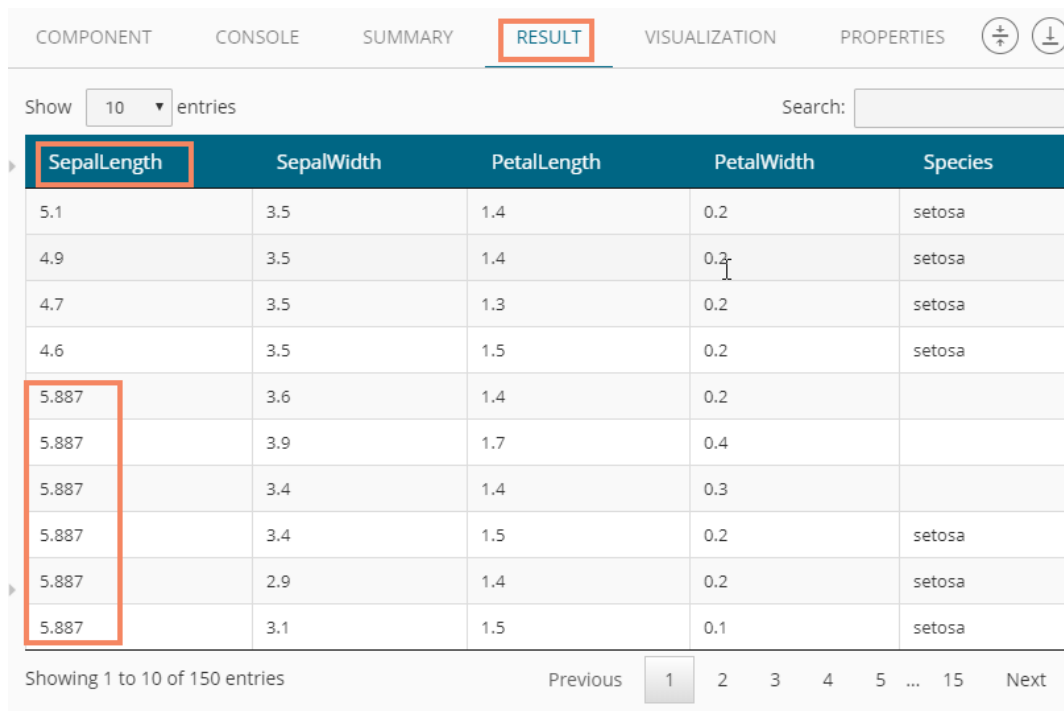
iv) Choose the replacement value by configuring the following fields:
   a. **Column Name**: Select a column using the drop-down that contains some missing values.
   b. **Replacement Options**: Select a replacement option using the drop-down menu.
      The following replacement options are provided under this field:

   1. Mean
   2. Median
   3. Mode
   4. Maximum
   5. Minimum
   6. Remove Entire Row
   7. Remove Entire Column
   8. Custom Replacement

   c. **Missing Value:** Users can get two options in this field-
      1. NaN
      2. Custom

v) Click the '**APPLY**' option.

vi) Run the workflow after getting success message.
vii) Users will get the process status under the '**CONSOLE**' tab.

viii) Follow the below given steps to display the result view:
    a. Click the dragged data preparation component on the workspace
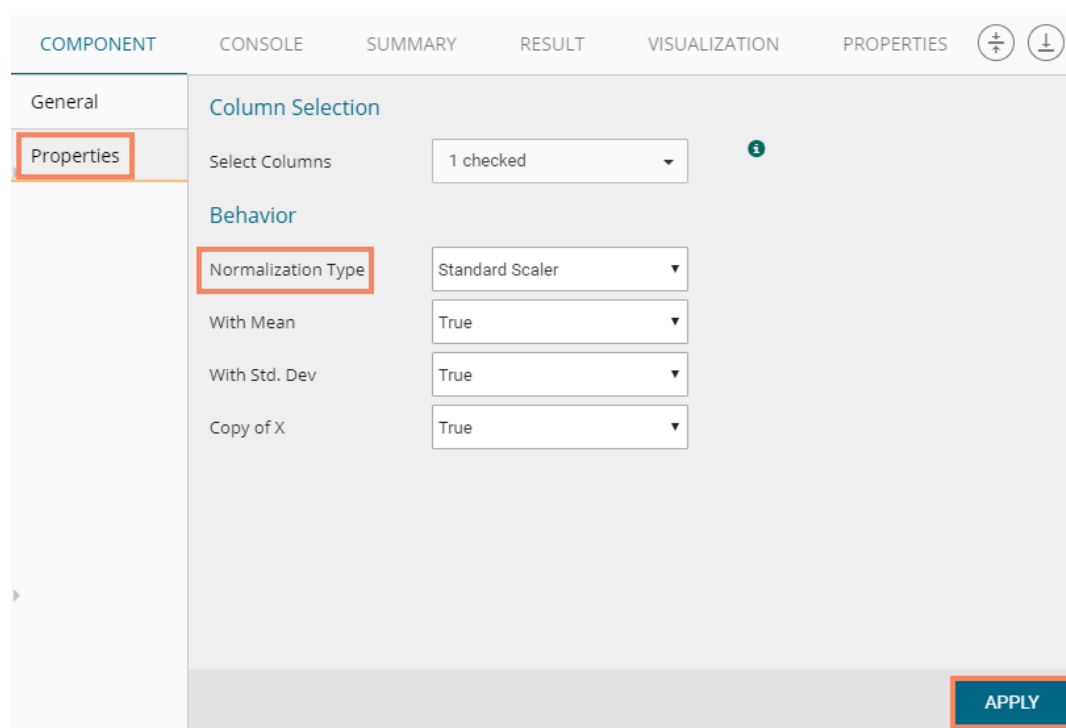   b. Click the '**Result**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 ▼ entries   Search:
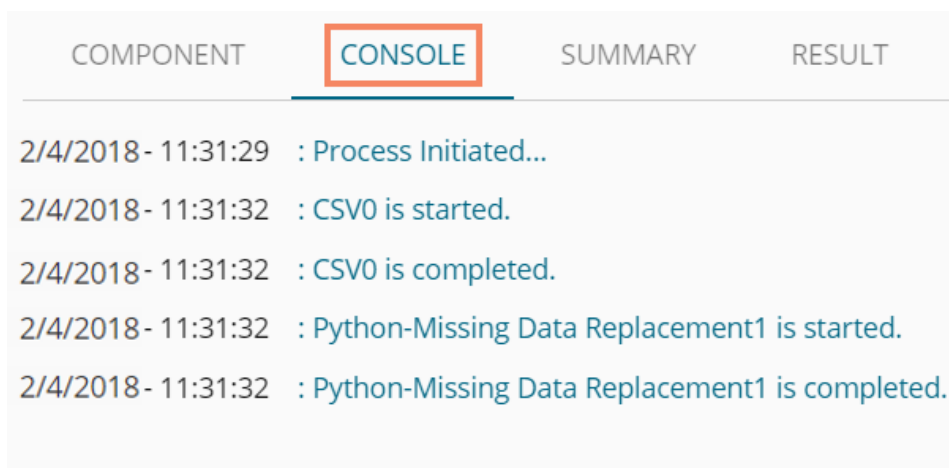
SepalLength	SepalWidth	PetalLength	PetalWidth	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.5	1.4	0.2	setosa
4.7	3.5	1.3	0.2	setosa
4.6	3.5	1.5	0.2	setosa
5.887	3.6	1.4	0.2	
5.887	3.9	1.7	0.4	
5.887	3.4	1.4	0.3	
5.887	3.4	1.5	0.2	setosa
5.887	2.9	1.4	0.2	setosa
5.887	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

7.2.2. Normalization Python

Normalization components transform data from more extensive range to a smaller range. Normalization can be done over numerical columns. The Python Normalization component supports following normalization methods which can be selected using the Normalization Type field provided under 'Properties' tab.

- Min-Max Scaling
- Maximum Absolute Scaler
- Normalizer
- Standard Scaler

7.2.2.1. Min-Max Normalization

Transform features by scaling each element by a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, i.e., between zero and one.
The transformation is given by,
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
Where min, max= feature_range

It is often used as an alternative to zero mean.

i)   Select and drag '**Normalization**' component onto the Workspace.
ii)   Connect the '**Normalization**' component to a configured data source.
iii) Click the '**Normalization**' component.

iv) Configure the following component fields:
     Properties
     a. **Column Selection**
          i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected).
     b. **Behavior**
          i. **Normalization Type:** Select '**Min-Max**' normalization type from the drop-down menu.
         ii. **New Maximum:** Set a new maximum value (Default value for this field is 1).
        iii. **New Minimum:** Set a new minimum value (Default value for New Minimum field is 0).
         iv. **Copy of X:** Select an option from the drop-down menu out of '**True**' or '**False**' options.
v)   Click the '**APPLY**' option.
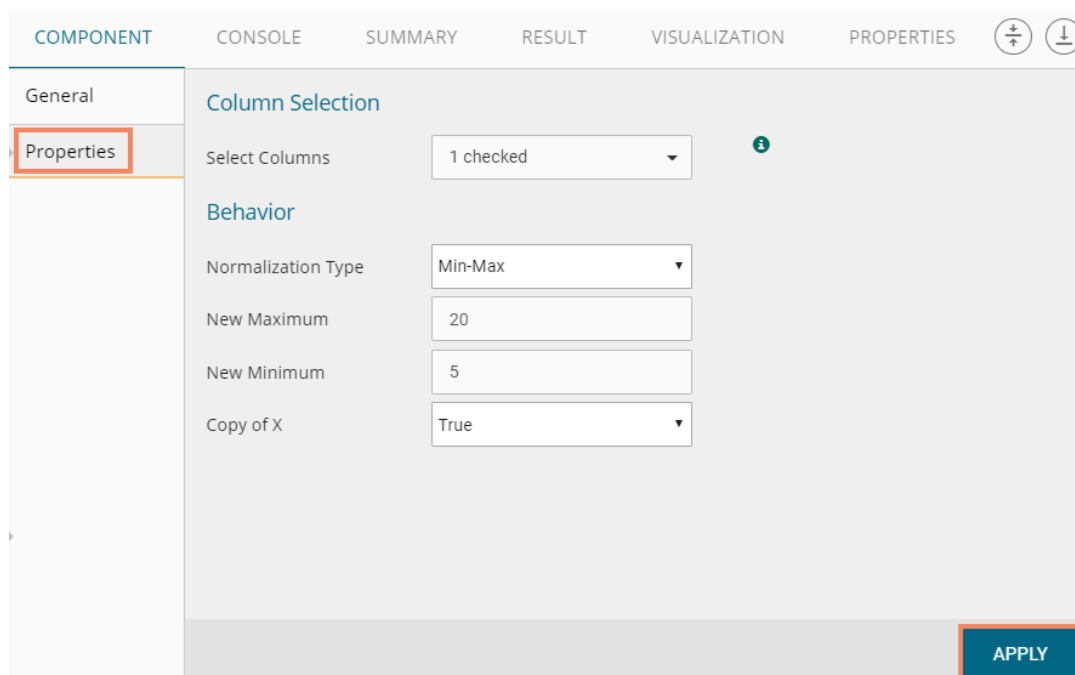
vi) Run the workflow after getting success message.
vii) Users will get the process status under the '**CONSOLE**' tab.

viii)   Follow the below given steps to display the result view:
   a.   Click the dragged algorithm component in the workspace.
   b.   Click the '**RESULT**' tab.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show [10 ▼] entries   Search: [   ]
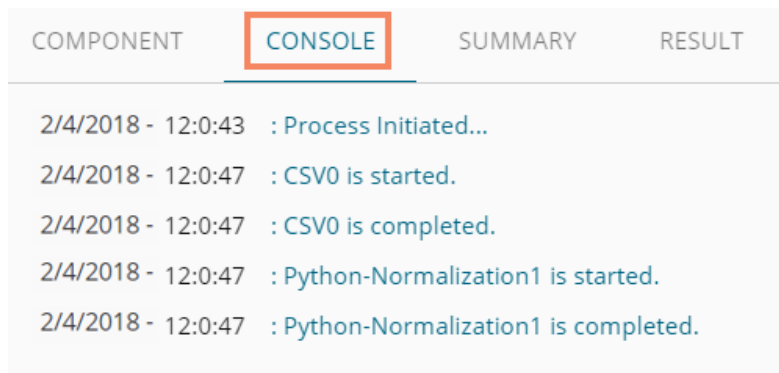
Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	8.3333	3.5	1.4	0.2	setosa
2	7.5	3	1.4	0.2	setosa
3	6.6667	3.2	1.3	0.2	setosa
4	6.25	3.1	1.5	0.2	setosa
5	7.9167	3.6	1.4	0.2	setosa
6	9.5833	3.9	1.7	0.4	setosa
7	6.25	3.4	1.4	0.3	setosa
8	7.9167	3.4	1.5	0.2	setosa
9	5.4167	2.9	1.4	0.2	setosa
10	7.5	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

7.2.2.2. Maximum Absolute Scaler

Maximum Absolute Scaler scales each feature by its maximum absolute value. This estimator scales and translates each feature individually such that the maximum absolute value of each feature in the training set will be 1.0. It does not shift/center the data and thus does not destroy any sparsity.

This scaler can be applied to sparse CSR or CSC matrix.

i)   Drag and connect a data source and Normalization Python components onto the workspace.
ii)   Configure the following component fields:
   Properties
   a. **Column Selection**
      i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)
   b. **Behavior**
      i. **Normalization Type:** Select '**Maximum Absolute Scaler**' normalization type from the drop-down menu
      ii. **Copy of X:** Select an option from the drop-down menu out of '**True**' or '**False**' options
iii)   Click '**APPLY**'.

iv)   Run the workflow after getting success message.

v)   Users will get the process status under the '**CONSOLE**' tab.

vi)   Follow the below given steps to display the result view:

a.   Click the dragged algorithm component on the workspace.
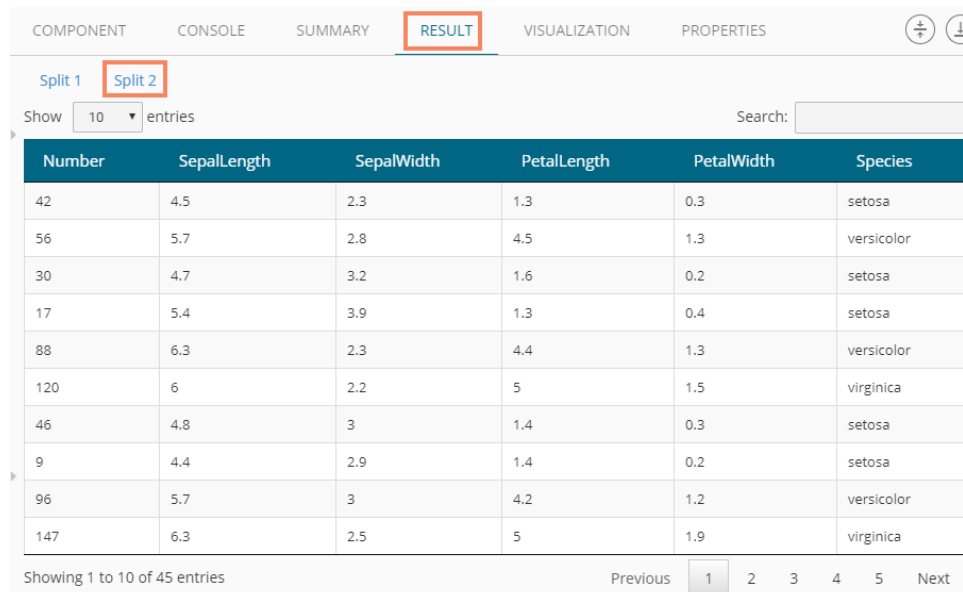
b.   Click the '**RESULT**' tab.

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	0.6456	3.5	1.4	0.2	setosa
2	0.6203	3	1.4	0.2	setosa
3	0.5949	3.2	1.3	0.2	setosa
4	0.5823	3.1	1.5	0.2	setosa
5	0.6329	3.6	1.4	0.2	setosa
6	0.6835	3.9	1.7	0.4	setosa
7	0.5823	3.4	1.4	0.3	setosa
8	0.6329	3.4	1.5	0.2	setosa
9	0.557	2.9	1.4	0.2	setosa
10	0.6203	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries

7.2.2.3. Normalizer

Normalizer: Normalize samples individually to unit norm. Each sample (i.e., each row of the data matrix) with at least one non-zero component is rescaled independently of other examples so that its norm (l1 or l2) equals one.

This transformation can work both with dense NumPy arrays and SciPy. Sparse matrix (use CSR format if you want to avoid the burden of a copy/ conversation).

Scaling inputs to unit norms is a common operation for text classification or clustering. For instance, the dot-product of two L2-normalized TF-IDF in the cosine similarity of the vectors and is the base similarity matric for the vector Space model commonly used by the Information Retrieval community.

- L1
- L2
- Max

This norm is used to normalize each non-zero sample.

i) Drag and connect a data source and Normalization Python components onto the workspace
ii) Configure the following component fields:

Properties
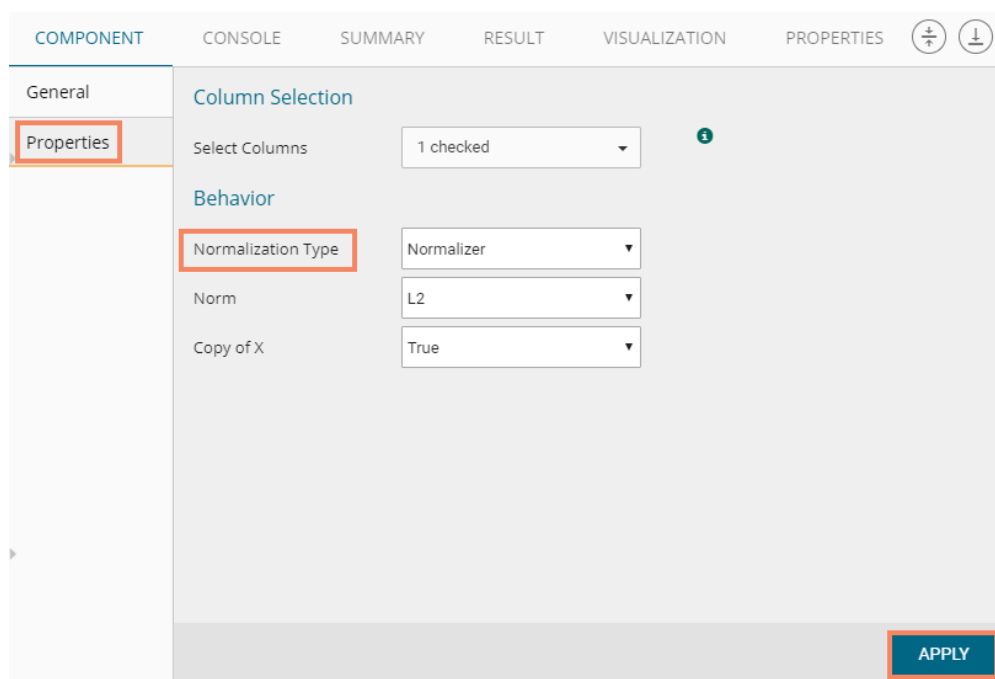
a. **Column Selection**
   i. **Select Columns**: Select a column using the drop-down menu (Only the numerical column will be selected)

b. **Behavior**
   i. **Normalization Type:** Select '**Maximum Absolute Scaler**' normalization type from the drop-down menu
   ii. **Norm: Select a norm option from the drop-down menu**
      1. **L1**
      2. **L2**
      3. **Max**
   iii. **Copy of X:** Select an option from the drop-down menu out of '**True**' or '**False**' options

iii) Click '**APPLY**'

iv)   After getting the success message run the workflow

v)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

2/4/2018 - 12:0:43   : Process Initiated...

2/4/2018 - 12:0:47   : CSV0 is started.

2/4/2018 - 12:0:47   : CSV0 is completed.

2/4/2018 - 12:0:47   : Python-Normalization1 is started.

2/4/2018 - 12:0:47   : Python-Normalization1 is completed.

vi)   Follow the below given steps to display the result view:

a.   Click the dragged algorithm component in the workspace

b.   Click the '**RESULT**' tab

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	1	3.5	1.4	0.2	setosa
2	1	3	1.4	0.2	setosa
3	1	3.2	1.3	0.2	setosa
4	1	3.1	1.5	0.2	setosa
5	1	3.6	1.4	0.2	setosa
6	1	3.9	1.7	0.4	setosa
7	1	3.4	1.4	0.3	setosa
8	1	3.4	1.5	0.2	setosa
9	1	2.9	1.4	0.2	setosa
10	1	3.1	1.5	0.1	setosa

Show 10 entries   Search:

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

7.2.2.4. Standard Scaler

This Normalization Type standardizes feature by removing the mean and scaling of unit variance. Centering and scaling happen independently on each element by computing the relevant statistics on the samples in the training set. Mean, and standard deviation are then stored to be used on later data using the transform method.
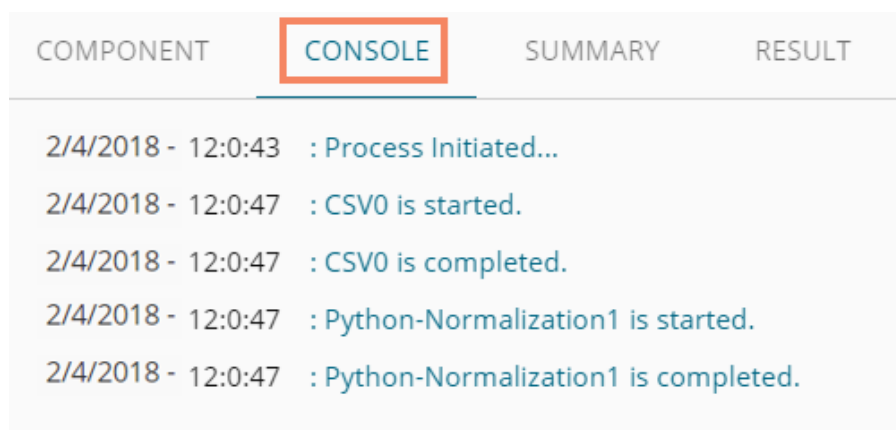
Standardization of a dataset is a common requirement for many machine learning estimators: they might misbehave if the individual feature does not more or less look like standard customarily distributed data (e.g., Gaussian with 0 mean and unit variance).

i) Drag and connect a data source and Normalization Python components onto the workspace
ii) Configure the following component fields:
   Properties
   a. **Column Selection**
      i. **Select Columns**: Select a column using the drop-down menu (Only the numerical column will be selected)
   b. **Behavior**
      i. **Normalization Type:** Select '**Maximum Absolute Scaler**' normalization type from the drop-down menu
      ii. **With Mean:** Select an option from the drop-down menu out of 'True' or 'False' options
      iii. **With Std. Dev:** Select an option from the drop-down menu out of 'True' or 'False' options
      iv. **Copy of X:** Select an option from the drop-down menu out of '**True**' or '**False**' options
iii) Click '**APPLY**'.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

Column Selection

Select Columns   1 checked   ▾   ⓘ

Behavior

Normalization Type   Standard Scaler ▾

With Mean   True ▾

With Std. Dev   True ▾

Copy of X   True ▾

APPLY

iv) After getting the success message run the workflow
v) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

2/4/2018 - 12:0:43 : Process Initiated...

2/4/2018 - 12:0:47 : CSV0 is started.

2/4/2018 - 12:0:47 : CSV0 is completed.

2/4/2018 - 12:0:47 : Python-Normalization1 is started.

2/4/2018 - 12:0:47 : Python-Normalization1 is completed.

vi)   Follow the below given steps to display the result view:
   a. Click the dragged algorithm component in the workspace
   b. Click the '**RESULT**' tab

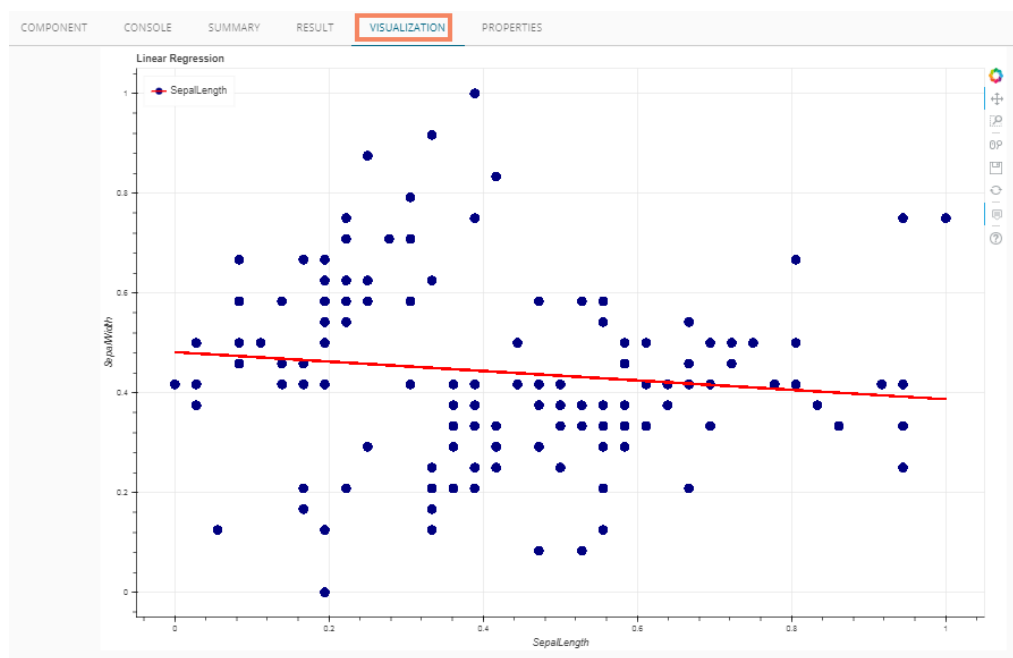| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	-0.9007	3.5	1.4	0.2	setosa
2	-1.143	3	1.4	0.2	setosa
3	-1.3854	3.2	1.3	0.2	setosa
4	-1.5065	3.1	1.5	0.2	setosa
5	-1.0218	3.6	1.4	0.2	setosa
6	-0.5372	3.9	1.7	0.4	setosa
7	-1.5065	3.4	1.4	0.3	setosa
8	-1.0218	3.4	1.5	0.2	setosa
9	-1.7489	2.9	1.4	0.2	setosa
10	-1.143	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

7.2.3. Python Split Data

Python Split Data component is used to split data into training and testing datasets. Once users find the best model from the trained data, he can pass test data to validate the model. Python Split Data will come as a leaf node under the Data Preparation tree node.

Python Split Data component consists of two connector nodes: Upper node for the **training dataset** and lower node for the **testing data set**.
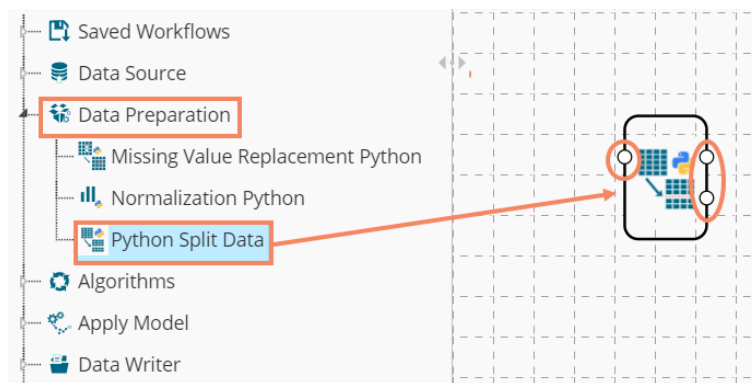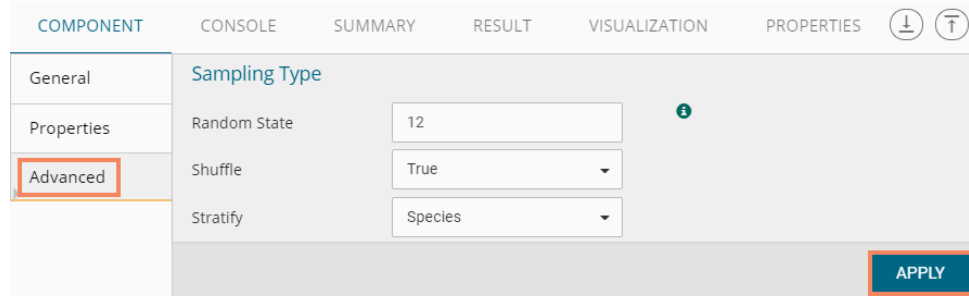
i)   Select the '**Python Split Data**' component and connect it with a valid data source (in this case, select Cassandra reader).
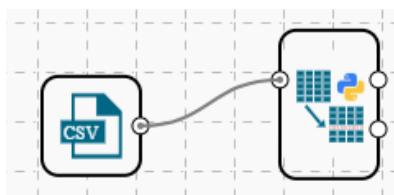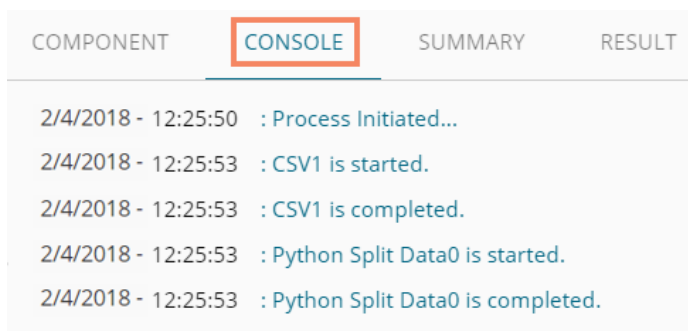
ii) Click the '**Python Split Data**' component in the workspace.

iii) Users will be directed to the Properties fields provided under the '**COMPONENT**' tab.

iv) Configure the following Properties:

    a. Relative (Train): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and sum of train and test should be 1).

    b. Relative (Test): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and sum of train and test should be 1).

v) Users can configure Sampling Type using the 'Advanced' fields

    a. Random State: Enter any positive integer value to configure this field

    b. Shuffle: Select an option using the drop-down menu

      i. True

      ii. False

    c. Stratify: Select an option from the drop-down menu

vi) Click '**APPLY**'
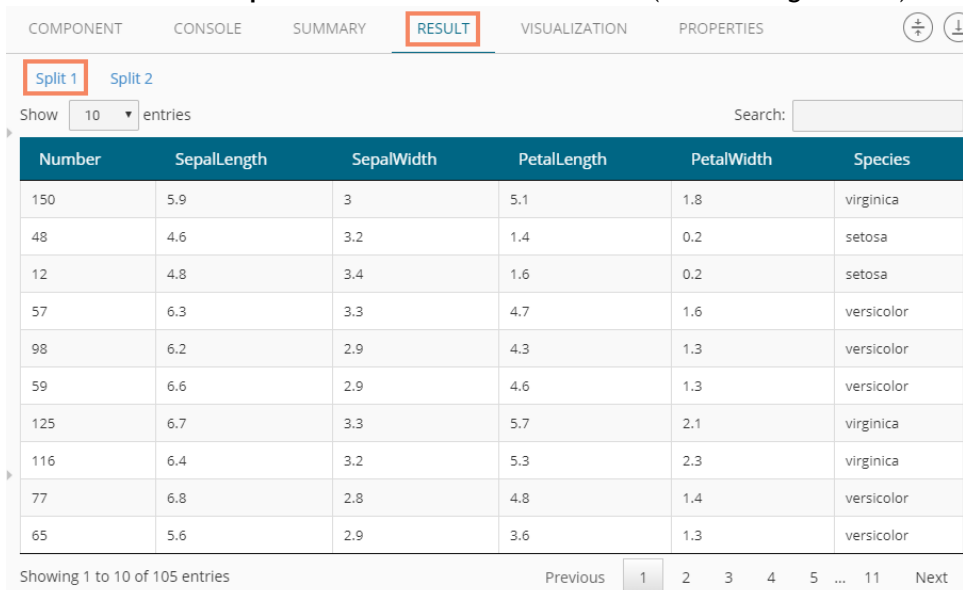
vii) After getting the success message run the workflow

viii) Users will get the process status under the '**CONSOLE**' tab

ix) Follow the below given steps to display the result view:

    a. Click the dragged algorithm component on the workspace.

    b. Click the '**RESULT**' tab.

The Result tab will have two data sets separated by a sub-tab. As shown in the below-given images:

a. Select the '**Split 1**' tab to see one set of data (the training dataset).

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

Split 1 | Split 2

Show 10 ▾ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
150	5.9	3	5.1	1.8	virginica
48	4.6	3.2	1.4	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
57	6.3	3.3	4.7	1.6	versicolor
98	6.2	2.9	4.3	1.3	versicolor
59	6.6	2.9	4.6	1.3	versicolor
125	6.7	3.3	5.7	2.1	virginica
116	6.4	3.2	5.3	2.3	virginica
77	6.8	2.8	4.8	1.4	versicolor
65	5.6	2.9	3.6	1.3	versicolor

Showing 1 to 10 of 105 entries   Previous 1 2 3 4 5 ... 11 Next

b. Select the '**Split 2**' tab to see another set of data (the testing dataset).

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES		

Split 1 | Split 2

Show 10 ▾ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
42	4.5	2.3	1.3	0.3	setosa
56	5.7	2.8	4.5	1.3	versicolor
30	4.7	3.2	1.6	0.2	setosa
17	5.4	3.9	1.3	0.4	setosa
88	6.3	2.3	4.4	1.3	versicolor
120	6	2.2	5	1.5	virginica
46	4.8	3	1.4	0.3	setosa
9	4.4	2.9	1.4	0.2	setosa
96	5.7	3	4.2	1.2	versicolor
147	6.3	2.5	5	1.9	virginica

Showing 1 to 10 of 45 entries   Previous 1 2 3 4 5 Next

7.3. Algorithms

7.3.1. Regression Analysis

This algorithm is used to determine how an individual variable influences another variable using an exponential function. It finds a trend in the dataset applying univariate regression analysis.

There are three subtypes provided under '**Regression Analysis':**

7.3.1.1. Python Linear Regression

i) Drag the Python linear Regression component to the workspace and connect it to a

configured data source.
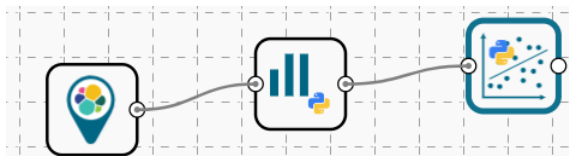
ii) Configure the following fields in the '**Properties**' tab:
   a. **Column Selection**
       i. **Dependent Column**: Select the target column on which the regression analysis will be applied
       ii. **Independent Column**: Select the required input columns against which the regression analysis will be applied to the target column
   b. **New Column Information**
       i. **Predicted Column Name**: Enter a name for the new column containing the predicted values.

iii) Click the '**Advanced**' tab and configure if required:
   a. **Input Data Handling**
       i. **Missing Values**: Select a method to deal with missing values from the drop-down menu
           1. **Fit Transform**: Selecting this option two actions will be performed on the data, Fit and Transform.
           2. **Stop**: Selecting this option will stop the algorithm application if a value is missing in any column.
   b. **Behavior**
       i. **Fit Intercept**: This option is used to select whether to calculate intercept for the selected model or not
           1. **True:** By selecting this option intercept will be calculated (It is the default selection)
           2. **False:** By selecting this option intercept will not be calculated
       ii. **Normalize**: This option is used to select whether to normalize the feature column or not

           1. True: If Normalize option is '**True**' the feature column will be it normalizes the feature column
           2. False: If Normalize option is '**False**,' the feature column will not be normalized (It is the default option)
       iii. **Copy of X Data**: This option is used to whether copy the feature column or not

1. True: If 'Copy of X Data' is 'True' then feature column will be copied (It is the default option)
2. False: If 'Copy of X Data' is 'False' then feature column will not be copied

iv)　Click '**APPLY**'

Note: Model containing aliased coefficients signifies that the square matrix x*x is singular.

v)　After getting the success message run the workflow
vi)　Users will get the process status under the '**CONSOLE**' tab
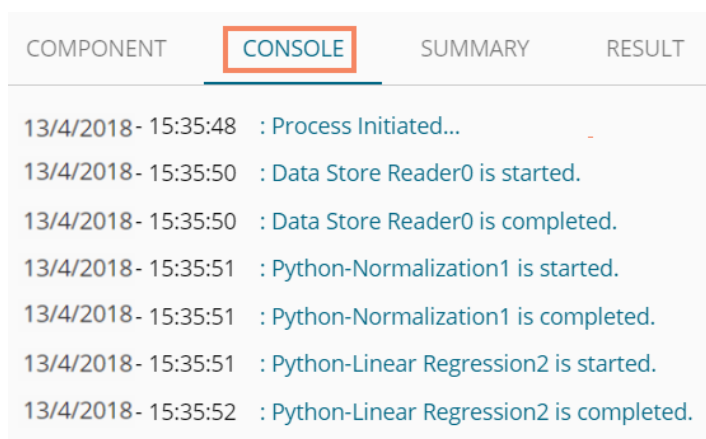
vii)　Follow the below given steps to display the result view:
a.　Click the dragged algorithm component onto the workspace.
b.　Click the '**RESULT**' tab.
i.　A new column '**Predicted Values1**' will be added to the result data displaying the predicted values.

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues2
0.12	0.56	0.38	0.78	0.71	virginica	0.43
0.12	0.92	0.42	0.95	0.83	virginica	0.39
0.12	0.81	0.67	0.86	1	virginica	0.41
0.12	0.61	0.5	0.69	0.79	virginica	0.42
0.12	0.56	0.29	0.66	0.71	virginica	0.43
0.12	0.58	0.33	0.78	0.83	virginica	0.43
0.12	0.81	0.42	0.81	0.62	virginica	0.41
0.12	0.72	0.46	0.75	0.83	virginica	0.41
0.13	0.69	0.5	0.83	0.92	virginica	0.42
0.13	0.22	0.62	0.07	0.04	setosa	0.46

Showing 1 to 10 of 7,142 entries   Previous 1 2 3 4 5 ... 715 Next

viii) Click the '**VISUALIZATION**' tab.

ix) The result data will be displayed via the Scatterplot with Regression line chart.

7.3.1.2. Python Multiple Linear Regression

i) Drag the R-Multiple Linear Regression component to the workspace and connect it with a configured data source.

ii) Configure the '**Properties**' tab as displayed below:

iii) Click the '**Advanced**' tab and configure if required:
   a. **Input Data Handling**
      i. **Missing Values**: Select a method to deal with missing values from the drop-down menu
         1. **Fit Transform**: Selecting this option two actions will be performed on the data, Fit and Transform.
         2. **Stop**: Selecting this option will stop the algorithm application if a value is missing in any column.
   b. **Behavior**
      i. **Fit Intercept**: This option is used to select whether to calculate intercept for the selected model or not
         1. **True:** By selecting this option intercept will be calculated (It is the default selection)
         2. **False:** By selecting this option intercept will not be calculated
      ii. **Normalize**: This option is used to select whether to normalize the feature column or not
         1. True: If Normalize option is '**True**,' it normalizes the feature column
         2. False: If Normalize option is '**False**,' the feature column will not be normalized (It is the default option)
      iii. **Copy of X Data**: This option is used to whether copy the feature column or not
         1. True: If 'Copy of X Data' is 'True' then feature column will be copied (It is the default option)
         2. False: If 'Copy of X Data' is '**False**' then feature column will not be copied
iv) Click '**APPLY**'
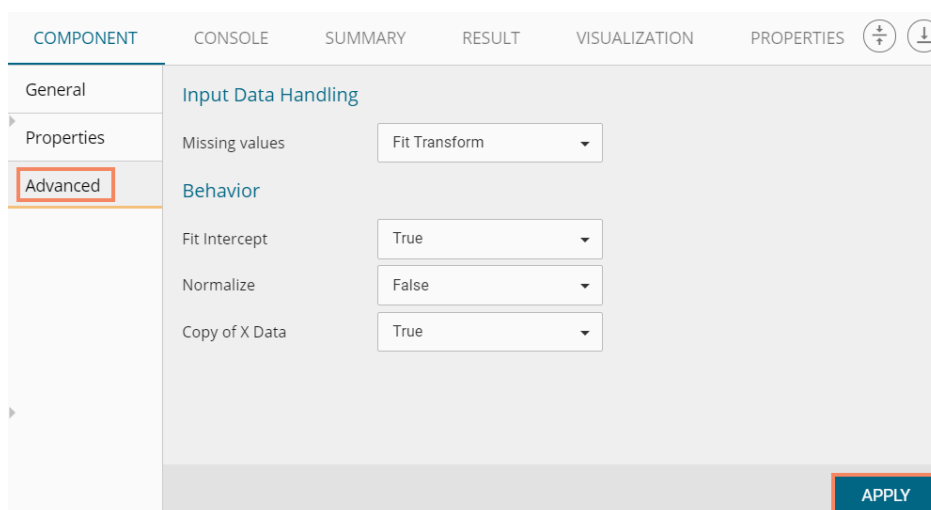
i)   After getting the success message run the workflow
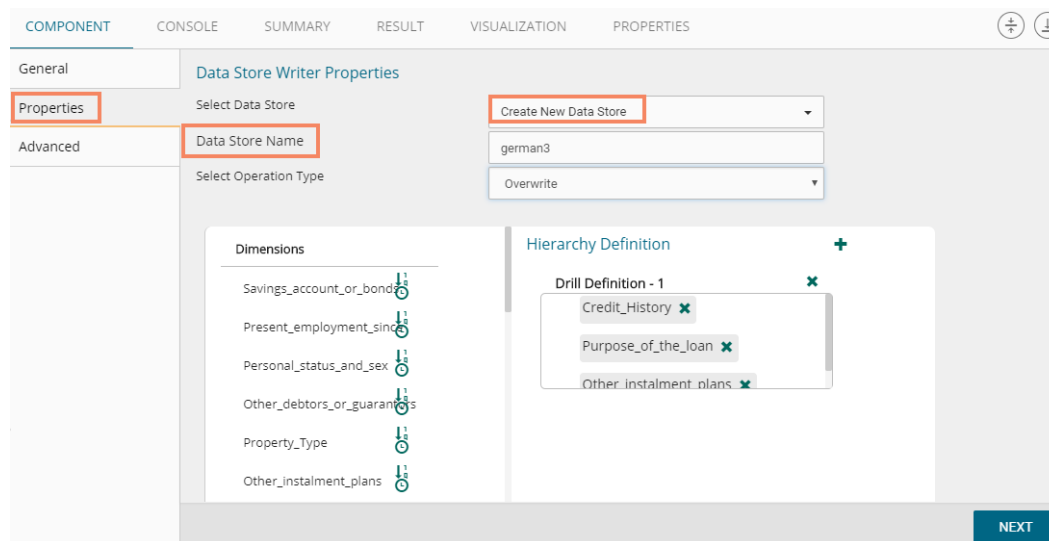ii)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION

13/4/2018 - 19:36:49   : Process Initiated...

13/4/2018 - 19:36:51   : CSV0 is started.

13/4/2018 - 19:36:52   : CSV0 is completed.

13/4/2018 - 19:36:52   : Python-Multiple Linear Regression1 is started.

13/4/2018 - 19:36:52   : Python-Multiple Linear Regression1 is completed.

v)   Follow the below-given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.
vi)   A new column will be added to the result data.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show [10 ▼] entries   Search: [   ]

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	55.62
2	4.9	3	1.4	0.2	setosa	49.9
3	4.7	3.2	1.3	0.2	setosa	44.18
4	4.6	3.1	1.5	0.2	setosa	41.32
5	5	3.6	1.4	0.2	setosa	52.76
6	5.4	3.9	1.7	0.4	setosa	64.2
7	4.6	3.4	1.4	0.3	setosa	41.32
8	5	3.4	1.5	0.2	setosa	52.76
9	4.4	2.9	1.4	0.2	setosa	35.6
10	4.9	3.1	1.5	0.1	setosa	49.9

Showing 1 to 10 of 154 entries   Previous 1 2 3 4 5 ... 16 Next

vii)   Click the '**VISUALIZATION**' tab.
viii) The result data will be displayed via the Scatterplot Chart with Regression line.
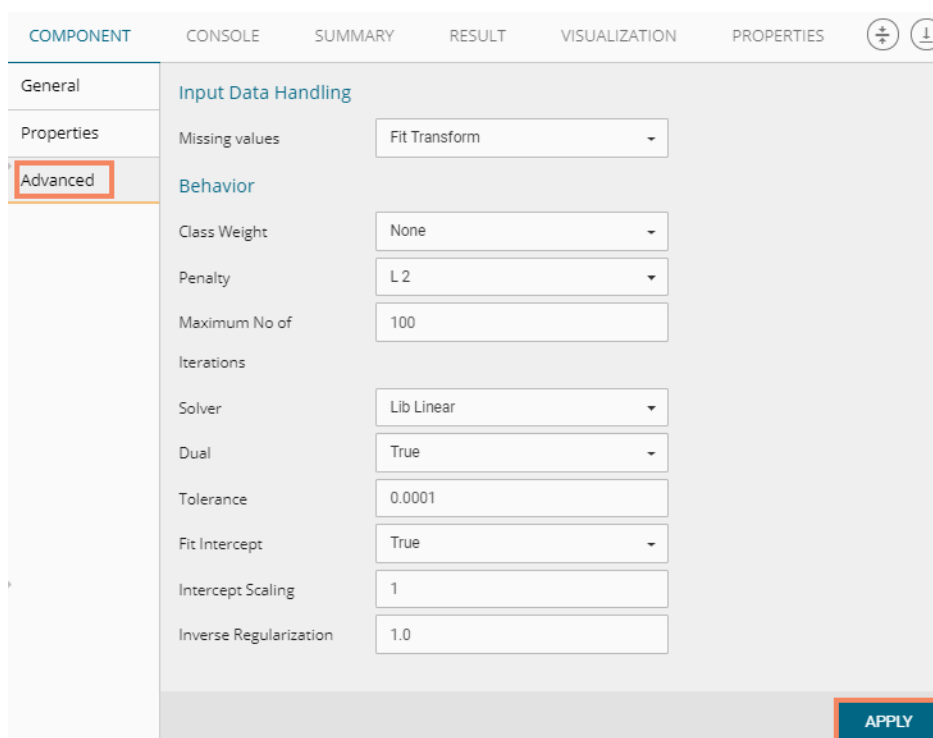
Multiple-Linear Regression   (click on the legend to hide respective fig.)
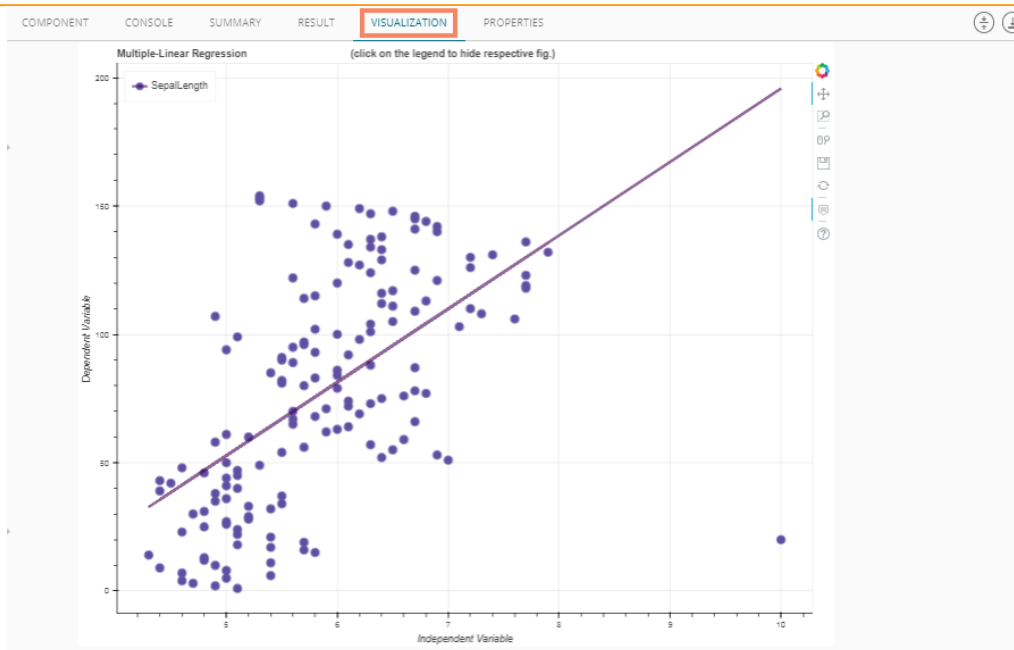
7.3.1.3. Python Logistic Regression

i)   Drag the R-Multiple Linear Regression component to the workspace and connect it with a configured data source.
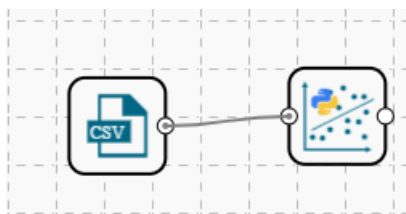
ii)   Configure the '**Properties**' tab as displayed below:

iii)   Click the '**Advanced**' tab and configure if required:

a.   **Input Data Handling**

i. **Missing Values**: Select a method to deal with missing values (via the drop-down menu)
1. **Fit Transform**: Selecting this option will consider the records containing missing values from the independent columns
2. **Stop**: Selecting this option will stop application of the algorithm if a value is missing in any column

b. **Behavior:** The fields provided under this section are used to improve model accuracy
i. Weight: This field can have either 'None' or 'Balanced' as value. The default value for this field is 'None.'
ii. Class Penalty: This field can have value either 'L1' or 'L2'. The default value for this field is 'L2'.
iii. Maximum No. of Iterations: Enter a valid integer value allowed to calculate the algorithm coefficient. The default values for this field is 100.
iv. Solver: The following options will be listed for this field
1. Newton-CG,
2. Lib- Linear (It is the default value for this field)
3. LBFGS
4. SAG
v. Dual: It can have Boolean value (The default value for this field is 'False')
vi. Tolerance: It can have double type value (The default value for this field is 0.0001)
vii. Fit Intercept: It has two options 'True' and 'False.' By selecting 'True' it calculates the intercept for the selected model (The default value for this field is 'True')
viii. Intercept Scaling: It can have double type value (The default value for this field is 1.0)
ix. Inverse Regularization: This field can only take value in double type (The default value for this field is 1.0)

iv) Click '**APPLY**'

v) After getting the success message run the workflow
vi) Users will get the process status under the '**CONSOLE**' tab

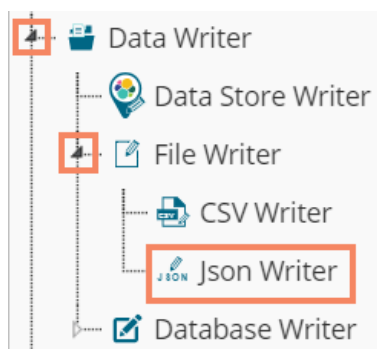| COMPONENT	CONSOLE	SUMMARY	RESULT

13/4/2018 - 19:56:5   : Process Initiated...

13/4/2018 - 19:56:8   : CSV0 is started.

13/4/2018 - 19:56:8   : CSV0 is completed.

13/4/2018 - 19:56:8   : Python-Logistic Regression1 is started.

13/4/2018 - 19:56:17   : Python-Logistic Regression1 is completed.

vii) Follow the below-given steps to display the result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**RESULT**' tab.

viii) A new column will be added to the result data.

admit	gre	gpa	rank	PredictedValues1
0	380	3.61	3	0
1	660	3.67	3	0
1	800	4	1	0
1	640	3.19	4	0
0	520	2.93	4	0
1	760	3	2	0
1	560	2.98	1	0
0	400	3.08	2	0
1	540	3.39	3	0
0	700	3.92	2	0

Showing 1 to 10 of 400 entries   Previous 1 2 3 4 5 ... 40 Next

ix) Click the '**VISUALIZATION**' tab.

x) The result data will be displayed via the Logistic Regression Classifier Chart.
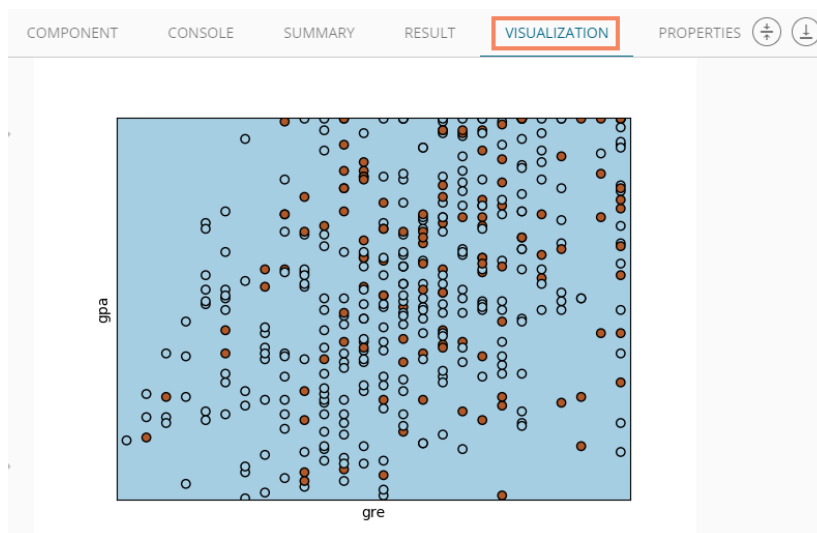
7.4. Apply Model

7.4.1. Python Apply Model

This component is provided to generate predictions based on Python trained model. Users can View predicted column value for each label class.

Users can create a model via the following ways:

- Generate a model using an algorithm
- Generate a model using the saved models

The Python Apply Model consists of 2 input nodes and 1 output node.

- **Input Nodes**
  - o Upper node – Model/Training data
  - o Lower node – Testing data
- **Output Node**
  - o Node – Result data

i)   Click the '**Apply Model**' tree-node

ii)   The '**Python Apply Model**' leaf-node will be displayed
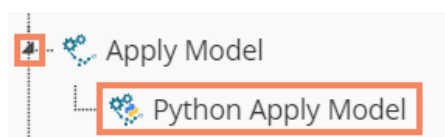
iii)   Drag the Python Apply Model component onto the workspace and connect it with a valid combination of Data source and algorithm (Configure the data source and algorithm components. In this case, the used algorithm is Python Logistic Regression)

iv)   Click the '**Python Apply Model**' component
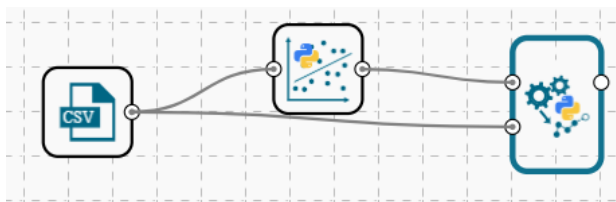
v)   Displays the basic details for the selected component
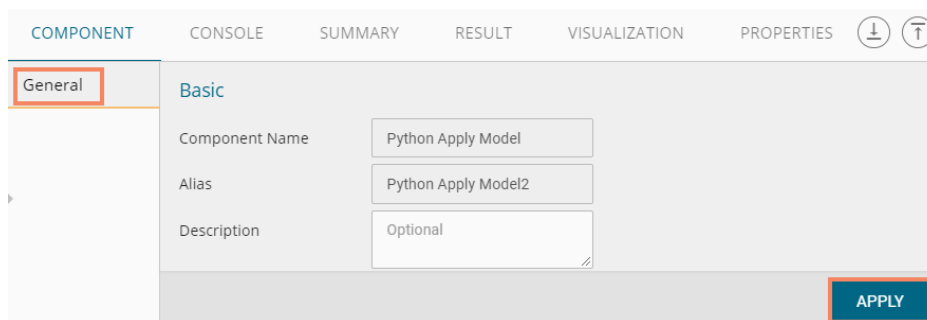
vi)   Click '**APPLY**'

vii)   After getting the success message run the workflow

viii)   Users will get the process status under the '**CONSOLE**' tab

COMPONENT	**CONSOLE**	SUMMARY	RESULT

2/4/2018 - 12:47:45 : Process Initiated...

2/4/2018 - 12:47:49 : CSV0 is started.

2/4/2018 - 12:47:49 : CSV0 is completed.

2/4/2018 - 12:47:49 : Python-Linear Regression1 is started.

2/4/2018 - 12:47:49 : Python-Linear Regression1 is completed.

2/4/2018 - 12:47:49 : Python Apply Model2 is started.

2/4/2018 - 12:47:49 : Python Apply Model2 is completed.

ix) Follow the below given steps to display the result view:
   a. Click the dragged Python Apply Model component on the workspace
   b. Click the '**RESULT**' tab

x) The columns displaying Predicted values and probability will be added to the result view

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	55.62119753165504
2	4.9	3	1.4	0.2	setosa	49.90076977816
3	4.7	3.2	1.3	0.2	setosa	44.18032838608934
4	4.6	3.1	1.5	0.2	setosa	41.320114509341835
5	5	3.6	1.4	0.2	setosa	52.760983654907506
6	5.4	3.9	1.7	0.4	setosa	64.2018528004732
7	4.6	3.4	1.4	0.3	setosa	41.320114509341835
8	5	3.4	1.5	0.2	setosa	52.760983654907506
9	4.4	2.9	1.4	0.2	setosa	35.599686755846776
10	4.9	3.1	1.5	0.1	setosa	49.90076977816

Showing 1 to 10 of 154 entries   Previous 1 2 3 4 5 ... 16 Next

xi) Click the '**SUMMARY**' tab to view the model summary

COMPONENT	CONSOLE	**SUMMARY**	RESULT	VISUALIZATION	PROPERTIES

```
--------- Summary of the Stage No. 1 Starts ---------

------------ Summary of the model ------------

1.Independent Columns

--------------------------------------------

Call:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

Coefficients:

        (Intercept) SepalLength
            -90.250     28.602

------------ End of Summary ----------

--------- Summary of the Stage No. 1 Ends ---------
```

Note:
- a. The result data set of the model can be written to a database using a Data Writer.
- b. The Column header and data type of feature column both should match for the saved model and testing data. If column headers and data types do not match, an alert message will be displayed.
- c. It is not mandatory for the testing data set to contain a label column.

7.5. Data Writer

Data Writers are provided to store the results of the predictive analysis in flat files or databases for further in-depth analysis.

7.5.1. Data Store Writer

Elastic Search Writer component is listed under the Data Writer Tree node. The Data Store Writer allows users to write the processed data onto the Elastic Search server which makes it more distributed.

i) Drag the Data Store Writer component to the workspace and connect it with a configured data source or any valid combination of a data source with other given components

ii) Click on the connected Data Store Writer component
iii) The component tab for the data writer will open
iv) Configure the required component properties
    i. Select Data Store: Select a data store from the drop-down menu
    ii. Select Operation Type: Select an option from the drop-down menu
    iii. Users will get all the Dimensions, Measures, and Time fields from the selected data source
    iv. They can define hierarchy by dragging the required Dimensions using the 'Drill Definition' box
v) Click '**NEXT**'

vi)   Users will be redirected to the Advanced fields to configure the Batch Query Properties

vii)   Select a dimension for the batch query

viii) Click '**APPLY**'

ix)   After getting the success message run the workflow

x)   Users will get the process status under the '**CONSOLE**' tab

xi)   The data will be saved in the desired format to the selected Data Store Writer after the console process gets completed.

Note:

a.   Users also get '**General**' fields for the Data Store Writer component, but they need not configure it.
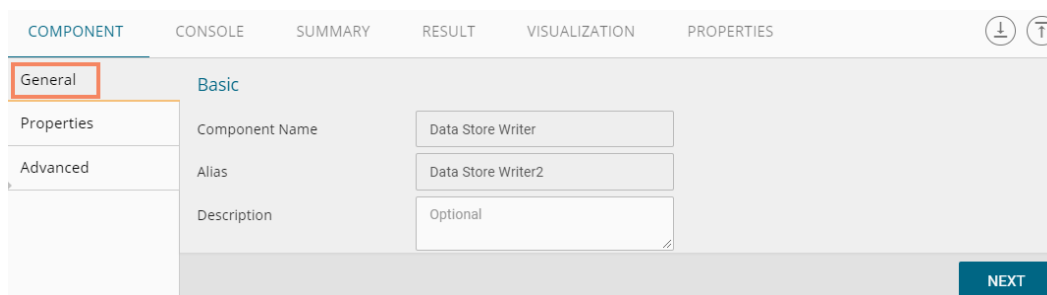
b. Users can also create a new data store using the 'Create New Data Store' option from the 'Select Data Store' drop-down menu. Users can give a name to the newly created data store by using the 'Data Store Name' field.

c. Users can move only one-dimension at a time from the list of '**Select Dimension for Batch Query**' value for the batch query.

7.5.2. File Writer

Users can write output data to flat files like CSV, TEXT, and DAT files using the File Writer.

7.5.2.1. CSV Writer

i)   Click '**TreeNode**' provided next to the '**Data Writer**' option.
ii)   Select '**File Writer**' option.
iii) Select and drag '**CSV Writer**' component to the workspace.

iv)   Connect the '**CSV Writer**' to a configured data source or a valid workflow
v)   Click on CSV Writer component to access component properties.
vi)   Enter '**File Name**' in the displayed field**.**
vii) Click '**APPLY**'

viii) After getting the success message run the workflow

ix) Users will get the process status under the '**CONSOLE**' tab

<	COMPONENT	CONSOLE	SUMMARY	RESULT

25/4/2018 - 11:1:11 : Process Initiated...

25/4/2018 - 11:1:13 : Data Service0 is started.

25/4/2018 - 11:1:13 : Data Service0 is completed.

25/4/2018 - 11:1:13 : Python-Linear Regression1 is started.

25/4/2018 - 11:1:13 : Python-Linear Regression1 is completed.

25/4/2018 - 11:1:13 : CSV File Writer2 is started.

25/4/2018 - 11:1:13 : CSV File Writer2 is completed.

x) The data will be written in the CSV File

xi) Click the '**CSV Writer**' component

xii) A pop-up message will appear with a link to download the CSV file

Download File   ×

Click Here to download Sample CSV File.csv
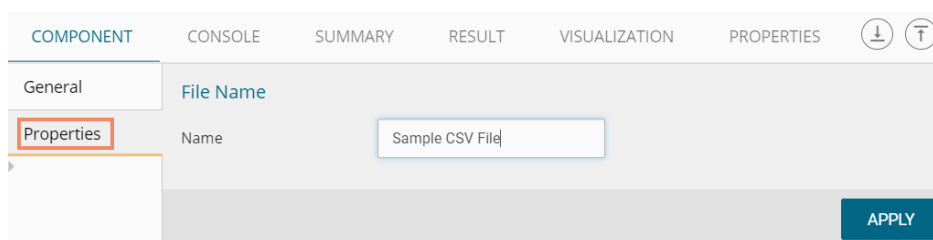
xiii) Click the link to download the CSV file.

7.5.2.2. JSON Writer

i) Click on '**TreeNode**' provided next to the '**Data Writer**' option.

ii) Select '**File Writer**' option.

iii) Select and drag '**JsonWriter**'component to the workspace.

Data Writer
— Data Store Writer
— File Writer
— CSV Writer
— Json Writer
— Database Writer

iv) Connect the '**JsonWriter**' to a configured data source or valid workflow

v) Click on '**JsonWriter**' component to access component properties.

vi) Enter '**File Name**' in the displayed field.

vii) Click '**APPLY**'
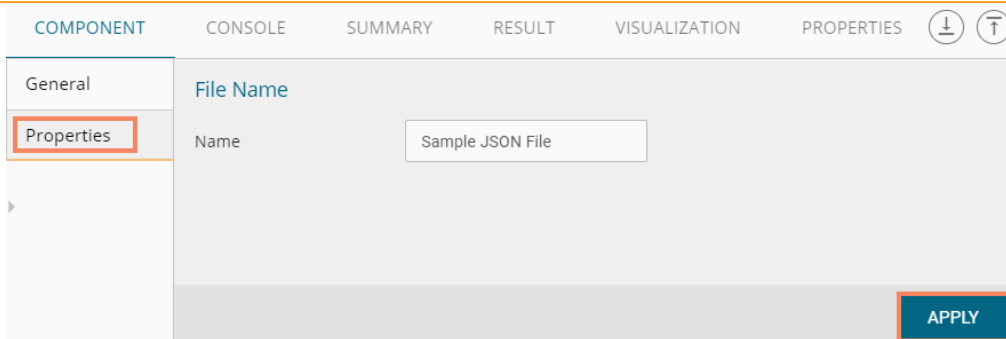
viii)   After getting the success message run the workflow

ix)   Users will get the process status under the '**CONSOLE**' tab.

x)   A Pop-up message will appear with a link to download the **JSON** file.
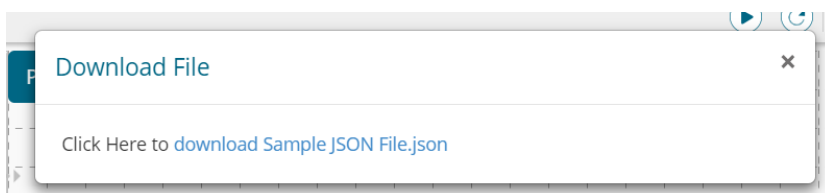
xi)   Click the link to download the JSON file.

7.5.3. Database Writer

7.5.3.1. Internal Data Writer

This data writer will store the data in databases like MySQL, MSSQL, and Oracle.

i)   Click '**TreeNode**' provided next to the '**Data Writer**' option.

ii)   Select '**Database Writer**' option.

iii)   Select and drag '**Internal Data Writer**' component to the workspace.
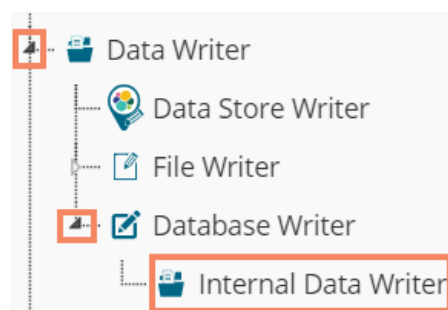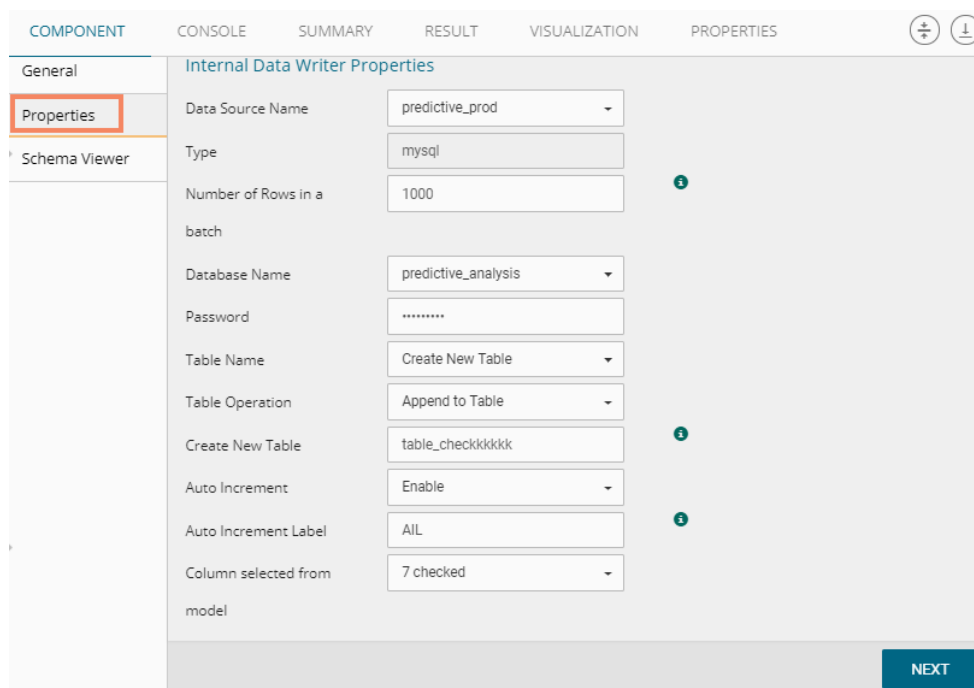
iv)   Drag and Connect the '**Internal Data Writer**' component to a configured data source or workflow onto the workspace.
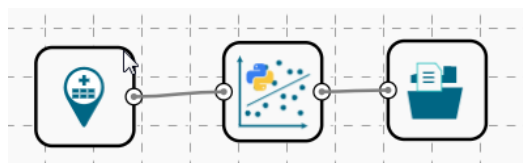
v)   Click '**Internal Data Writer**' component to access the Component properties

Users will have different '**Properties**' fields based on the selected table operation as described below:
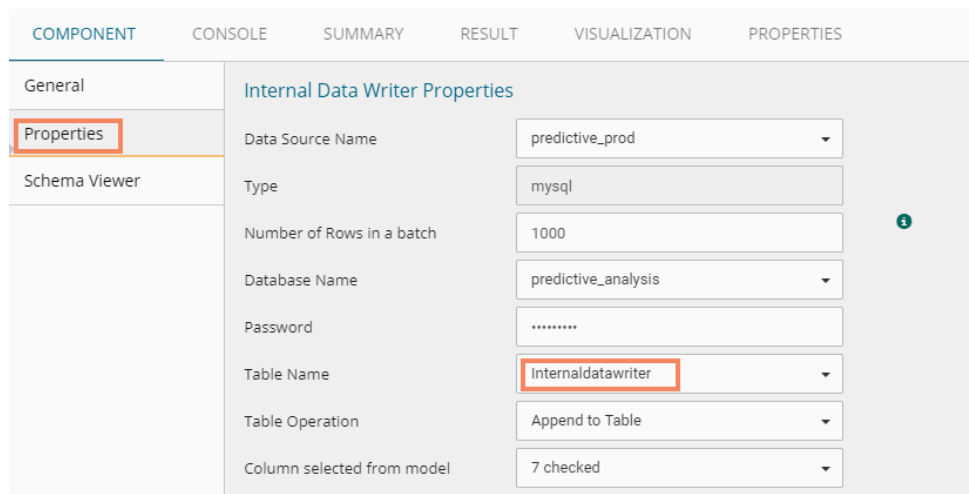
a. Selecting the 'Create a New Table' option as the 'Table Operation':

   i. **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.
   ii. **Type**: This field will be preselected based on the selected data Connector.
   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
   iv. **Database Name**: Select a database name from the drop-down menu
   v. **Password:** Enter the database password
   vi. **Table Name**: Select '**Create New Table**' option from the list
   vii. **Table Operation**: Select an option from the drop-down menu
   viii. **Create New Table:** It is an optional field. It appears when the user selects '**Create New Table**' option from the '**Table Name**' drop-down menu.
   ix. **Auto Increment:** Select an option to enable or disable the auto increment. By enabling this option, a new column will be added to the dataset, and the same column will be selected as the primary key by default.
   x. **Auto Increment Label:** Enter a name for the auto increment label
   xi. **Column Selected from model**: Select columns that are needed to be written into the selected database.

vi) Click '**NEXT**'

b. Selecting an Existing Table as the 'Table Operation':

   i. **Data Connector Name**: Select a data connector from the drop-down menu
   ii. **Type**: Displays a type based on the selected data connector
   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
   iv. **Database Name**: Select a database name from the drop-down menu
   v. **Password:** Enter the database password
   vi. **Table Name**: Select an existing table name from the drop-down menu

    vii.   **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:
1. Append Table
2. Overwrite Table

   viii.   **Column Selected from model**: Select columns that are needed to be written into the selected database.

   ix.   **Details of the Selected table**: Displays column headers from the selected table.

vii)   Click '**NEXT**'

viii)   Run the Workflow

ix)   Users will be directed to the '**Console**' tab to check the progress of the process

x)   The data will be saved in the selected database

7.5.3.2. Delta Load

The internal data writer can extract only new or changed records while loading data from the MySQL database. The Schema View has been added to the internal database writer to extract data using the delta data load type.
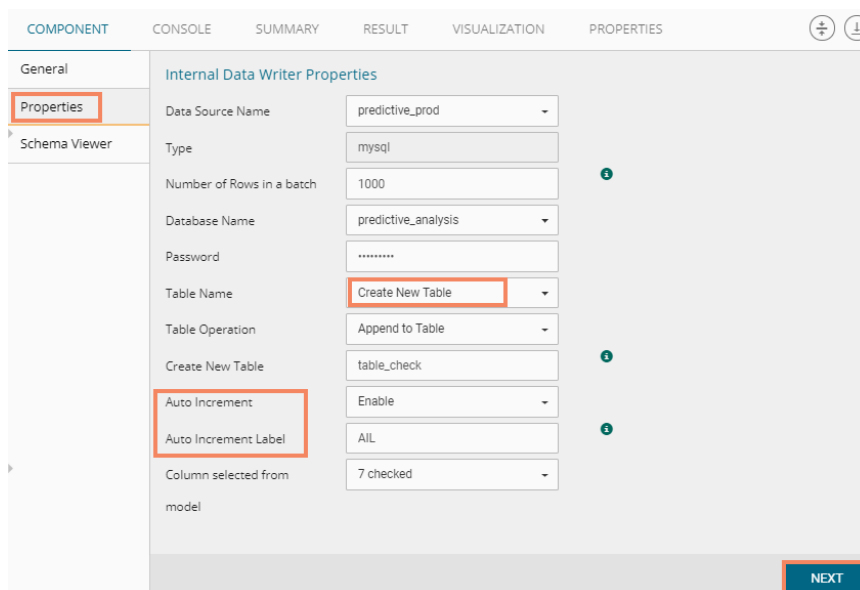
i)   Click **the tree-node** provided next to the '**Data Writer**' option.

ii)   Select '**Database Writer**' option.

iii)   Select and drag '**Internal Data Write**r' component to the workspace.

iv)   Connect the '**Internal Data Writer**' component to a configured data source

v)   Click the '**Internal Data Writer**' component

vi)   Users will be directed to the Properties of the Data Writer component

Users will have different properties fields based on the selected table choice as described below:

a. Selecting 'Create a New Table' as Table Operation:

i.   **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.
ii.   **Type**: This field will be preselected based on the selected data Connector
iii.   **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
iv.   **Database Name**: Select a database name from the drop-down menu
v.   **Password:** Enter the database password.
vi.   **Table Name**: Select '**Create New Table**' option from the list.
vii.   **Table Operation**: Select an option using the drop-down menu.
The following choices are provided:
1.   **Append**: Rows can be appended to the table
2.   **Overwrite**: Delete the existing information and write the new data.
3.   **Upsert**: Insert rows to table if they do not exist or update them if they do.
viii.   **Create New Table:** Enter the table name using this field (This field appears when the user selects '**Create New Table**' option using the '**Table Name**' field).
ix.   **Auto Increment**: User can enable or disable '**Auto Increment**' by selecting any one out of '**Enable**' or '**Disable**' options.
x.   **Auto Increment Label**: Enter a label for the autoincrement column (This field will be displayed only if, the user has enabled '**Auto Increment**' option).
xi.   **Column Selected from the model**: Select columns from the model that is to be written into the selected database.
xii.   Click '**NEXT**'

Note: The Schema Viewer tab will be displayed only after configuring the '**Table Name**' field.

vii)   Users will be directed to the '**Schema Viewer**' tab.
viii)   Define Primary keys by using the '**Select Primary Keys**' field.
ix)   Click '**APPLY**'

b. **Selecting an Existing Table as the 'Table Operation':**

  i. **Data Connector Name**: Select a data connector from the drop-down menu

  ii. **Type**: Displays a type based on the selected data connector

  iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch

  iv. **Database Name**: Select a database name from the drop-down menu

  v. **Password:** Enter the database password

  vi. **Table Name**: Select an existing table name from the drop-down menu

  vii. **Table Operation**: Select an option using the drop-down menu. The following choices are provided:

    1. **Append**: Rows can be appended to the table
    2. **Overwrite**: Delete the existing information and write the new data.
    3. **Upsert**: Insert rows to table if they do not exist or update them if they do

  viii. **Column Selected from the model**: Select columns that are to be written into the selected database.

  ix. **Details of the Selected table**: Displays column headers from the selected table.

  x) Click '**NEXT**'

Details of the selected table

auto
price
variety
kind
aroma
PredictedValues1

NEXT

xi) Users will be directed to the '**Schema Viewer**' tab.
xii) The defined/selected primary keys will be displayed.
xiii) Click '**APPLY**'

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

General
Properties
Schema Viewer

Internal Data Writer Properties

Selected Primary Keys
AIL

APPLY
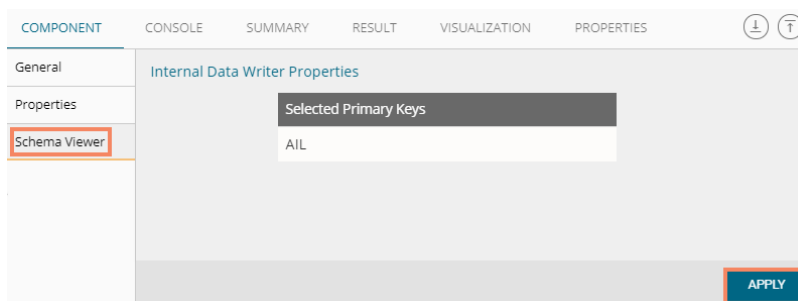
xiv) After getting the success message run the workflow
xv) Users will get the process status under the '**CONSOLE**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT |

25/4/2018 - 19:23:58 : Process Initiated...

25/4/2018 - 19:24:1 : Data Service0 is started.

25/4/2018 - 19:24:1 : Data Service0 is completed.

25/4/2018 - 19:24:1 : Python-Linear Regression1 is started.

25/4/2018 - 19:24:1 : Python-Linear Regression1 is completed.

25/4/2018 - 19:24:1 : Internal Data Writer2 is started.

25/4/2018 - 19:24:1 : Internal Data Writer2 is completed.

xvi) Users will be directed to the '**RESULT**' tab

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 entries   Search:
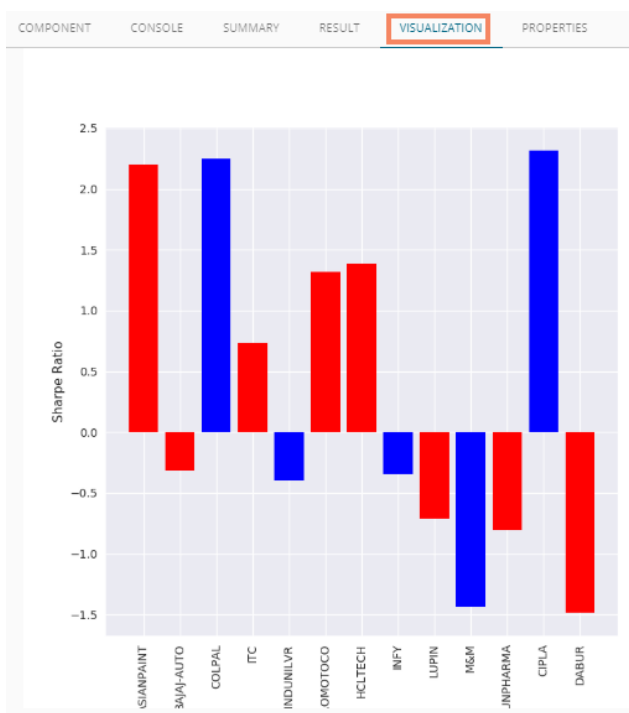
Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	48
2	4.9	3	1.4	0.2	setosa	40
3	4.7	3.2	1.3	0.2	setosa	33
4	4.6	3.1	1.5	0.2	setosa	29
5	5	3.6	1.4	0.2	setosa	44
6	5.4	3.9	1.7	0.4	setosa	59
7	4.6	3.4	1.4	0.3	setosa	29
8	5	3.4	1.5	0.2	setosa	44
9	4.4	2.9	1.4	0.2	setosa	21
10	4.9	3.1	1.5	0.1	setosa	40

Showing 1 to 10 of 450 entries   Previous 1 2 3 4 5 ... 45 Next

Note: The Result data appears based on the input data source. Users can even use the Data Preparation components and algorithms in a workflow before saving the data in a data writer.

7.6. Custom Python Script

Users can create and add customized algorithm components using the '**Custom Python Script**' component. The created scripts will be stored in the '**Saved Scripts**' module provided for the Python Scripts.

7.6.1. Creating a New Python Script

ii) Click '**Custom Python Script**' tree-node on the Predictive Analysis home page.

iii) Click '**Create New Script**' option
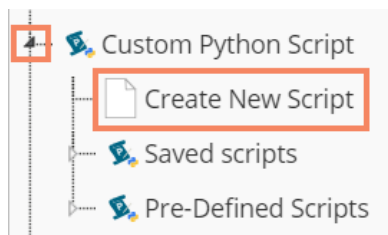
iv) Users will be directed to the '**Component**' tab.

v) Configure the following fields in the '**General**' tab:

    a. Basic

        i.   **Component Name:** Enter a name or title that you wish to give a saved Python Script.

        ii.   **Component Type:** Default Component type will be displayed in this field.

        iii. **Description:** Describe the Component (It is an optional field).

vi) Click '**NEXT**'
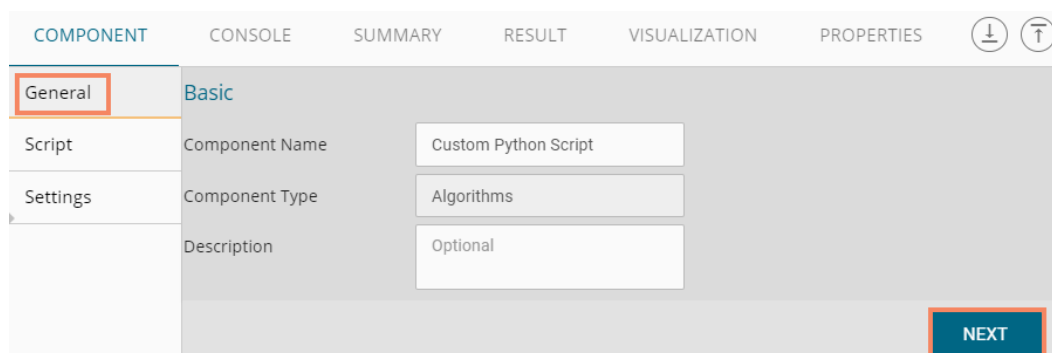
vii) Users will be directed to the '**Script**' tab.

viii) Provide the following information:

    a. Script Editor

        i.   Write the python script in the given space under the '**Script Editor.**'

        ii. Click the '**Validate**' option
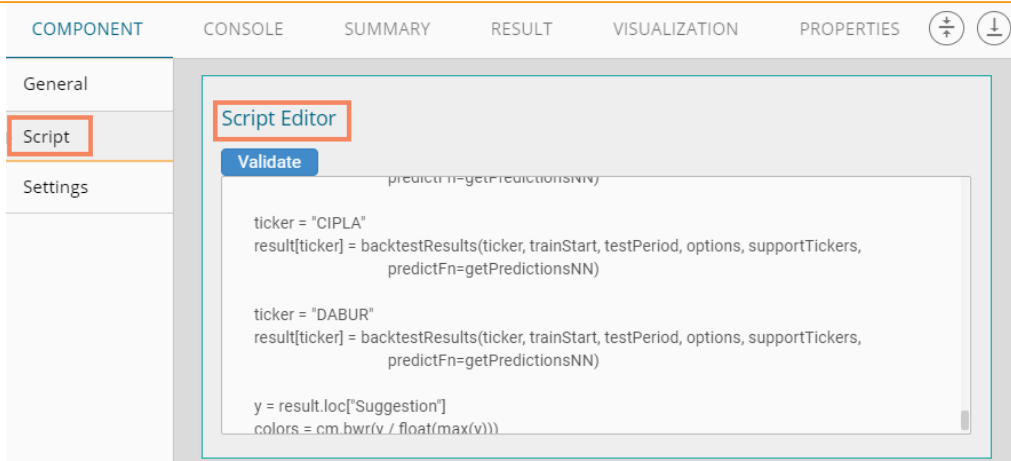
b. Configure the required 'Primary Function Details' to embed the customized Python script into a function.
    i. **Primary Function Name:** Select the name of the created function from the drop-down menu.
    ii. **Input Data Frame:** Select a dataset (that has been used above) from a drop-down menu.
ix) Click **'NEXT'** (Users can click **'Previous'** if wish to open the previous page)
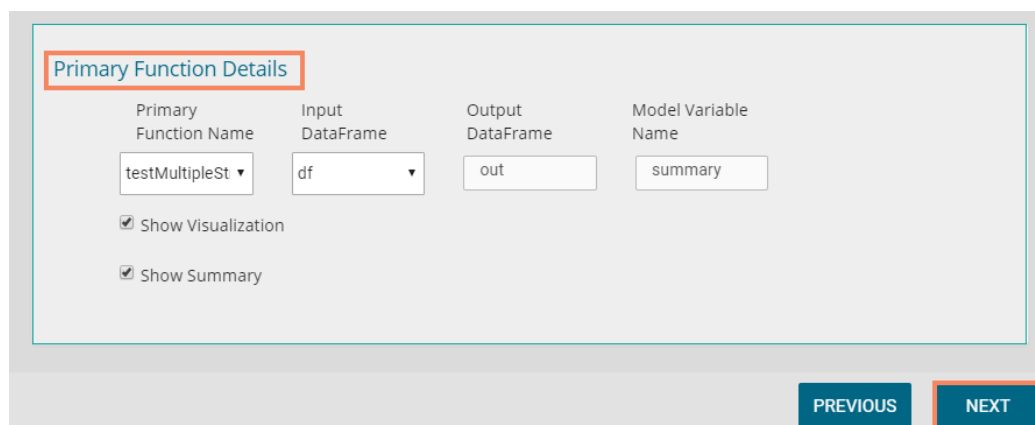
x) Users will be directed to the **'Settings'** tab.
xi) Configure the following fields:
    a. **Output Table Definition**
       This option will configure a number of output columns, column headers, data types. Select any one out of the following options:
       i. **Consider all columns from the previous component:** To display all columns from the previous component
       ii. **Consider None:** To display no column from the previous component
    b. **Define Output Columns**
       i. **Output Column Name**: Enter an appropriate name for the new predicted column
       ii. [**–**]: To remove the added row containing **'Data Type'** and **'New Predicted Column Name.'**
       iii. [**+**]: To add a new row containing **'Data Type'** and **'New Predicted Column Name.'**

c. **Property View Definition**
   i. **Function Parameters**: Actual names of parameters configured in the script.
   ii. **Property Display Name**: Parameter name to be displayed while configuring the saved script as a component.
   iii. **Control Type:** User can select out of the following options:
      1. Text box,
      2. Drop-down menu,
      3. Column Selector (single),
      4. Column Selector (multiple).

   iv. **Settings option** [⚙]: To set display for mandatory fields and validate **the** data type for the input column. This field is associated with function parameters.

xii) Click '**APPLY**'
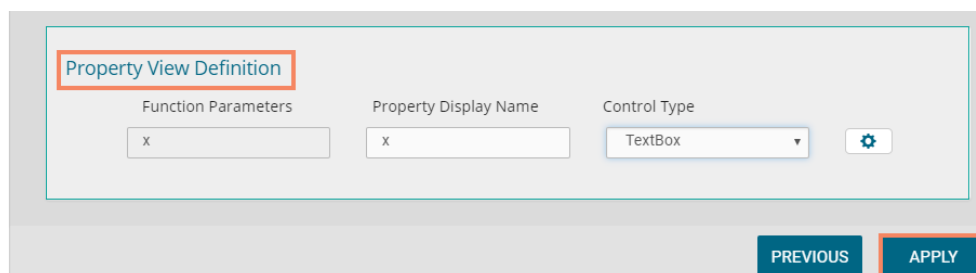
xiii) A message will pop-up to notify that the newly created Python script has been saved successfully.
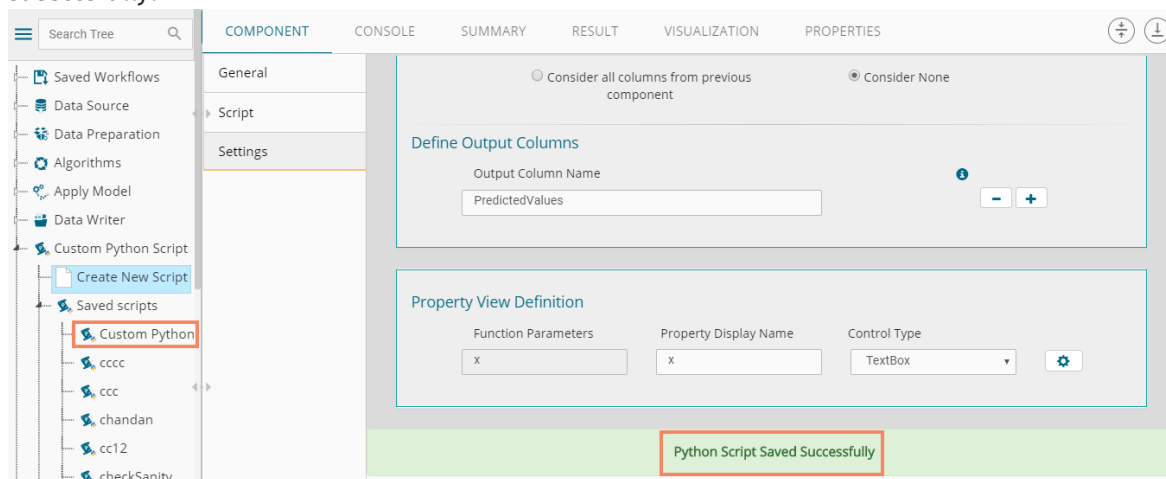
xiv) The newly created Python Script will be saved in the '**Saved Scripts**' list provided for the Custom Python Script.

Guidelines for Writing a Python Script

1. The First argument of the function should be a data frame.
2. The Python script needs to be written inside a valid Python function. E.g., the entire code body should be inside the proper indentation of the function (Use 4 spaces per indentation level.)
3. The Python script should have at least one primary function. Multiple functions are acceptable, and one function can call another function, but it should be written above the calling function body (if the called function is an outer function) or above the calling statement (if the called function is an inner function).
4. Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets, and braces, or using a hanging indent. When using a hanging indent, the following should be considered; there should be no arguments on the first line, and further indentation should be used to distinguish itself as a continuation line clearly.
5. Spaces are the preferred indentation method.
6. Limit all lines to a maximum of 79 characters. The Python standard library is conservative and requires limiting lines to 79 characters (and doctrines/comments to 72).
7. Do not use "type" as the function argument, as it is a predefined keyword.
8. In Python, single-quoted strings and double-quoted strings are the same.
9. All the packages used in function need to import explicitly before writing function.
10. The Python script should return data in the form of a data Frame only and should define while writing function.
11. The column names should remain the same while creating new columns in the Output Table Definition.
12. If users need to define column selector (Multiple), then in definition ': List[String]' should be used and body of the function should be in '.to Array'.
13. If users need to define column selector (Single), then 'String' must be used in the definition.

Note:

a. Click the '**Information**' button 🛈 to get the rules to write a Python script.
b. All the supported date data types are listed in date formats in data type definition, all other date formats are considered as string data type.
c. Mssql data types are considered as string data type.

7.6.2. Saved Python Scripts

7.6.2.1. Viewing a Saved Python Script

i)   Select a Scala Script from the '**Saved Scripts**' list.
ii) Right-click on the selected Python Script.
iii) A context menu will open.
iv) Select the '**View**' option.
v)   Users will be redirected to the '**Component**' tab.
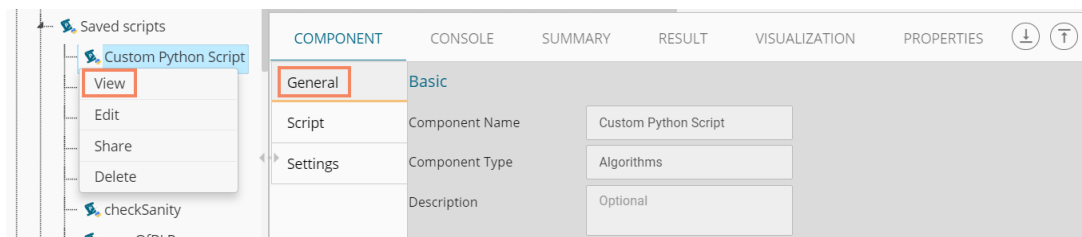
7.6.2.2. Editing a Saved Python Script

i)   Select a Scala Script from the list of '**Saved Scripts**' list.
ii) Right-click on the selected Python Script.

iii) A context menu will open.
iv) Select '**Edit**'
v) Users will be redirected to the '**Component**' tab
vi) Users can edit the required fields provided under **General**, **Script**, and **Settings** tabs
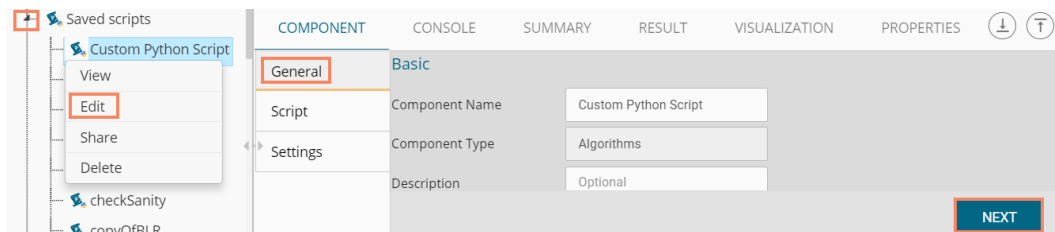
7.6.2.3. Sharing a Saved Python Script

This feature gives users the ability to share a custom Python script with other users and groups. The following options are available to share a custom R script:

1. **Share With**: This option allows the user to share a custom Python script with selected users or user groups. Any changes made to the custom Python script will be transferred to all the users with whom the custom Python script has been shared.

   i) Select a Python script from the list of '**Saved Scripts**'
   ii) Right-click on the selected Python script
   iii) Select '**Share**' from the context menu
   iv) The '**Share With**' option will be displayed (by default)
   v) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group when the '**Group**' option has been selected.
      b. Users can be excluded by not selecting a username from the list when '**User**' option has been selected.
   vi) Select a specific user or group from the list by check marking the box.
   vii) Click '**APPLY**'
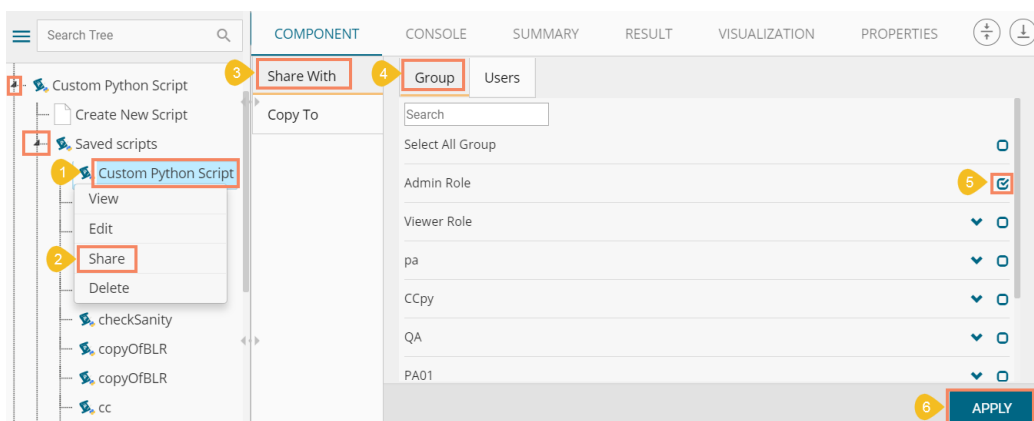
viii) The selected Python script will be shared with the chosen user(s)/group(s).

2. **Copy To**: This option creates a copy and shares the copy of the custom Scala script with the selected users and user groups. Any changes to the original custom Scala script after sharing will not show up for the users that received the shared file via the '**Copy To**' option.

i)   Select a Python script from the list of '**Saved Scripts**'.
ii)   Right-click on the selected Python script.
iii) Select '**Share**' from the context menu.
iv)   Select '**Copy To**' option.
v)   The copied custom Python script name will be displayed in a box.
vi)   Select either the '**Group**' or '**Users**' tab.
    a.   By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group when the '**Group**' option has been selected.
    b.   Users can be excluded by not selecting a username from the list when the '**Users**' option has been selected

vii) Select a specific user or group from the list by check marking the box.
viii) Click '**APPLY**'
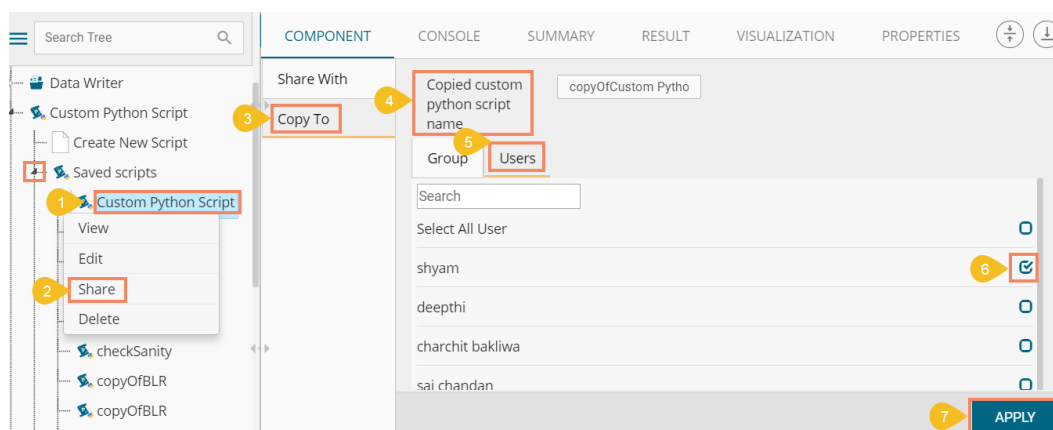
7.6.2.4. Deleting a Saved Python Script

i)   Select a Python Script from the '**Saved Scripts**' list.
ii)   Right-click on the selected Scala Script.
iii) A context menu will open.
iv)   Select the '**Delete**' option.
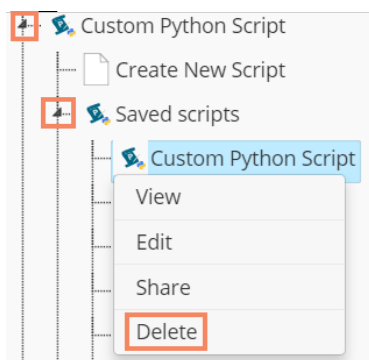
v)   A pop-up window will appear to assure the deletion.
vi)   Click '**OK**'

Delete Custom Python Script ⊗

Do you want to delete Custom Python Script ?

CANCEL   OK
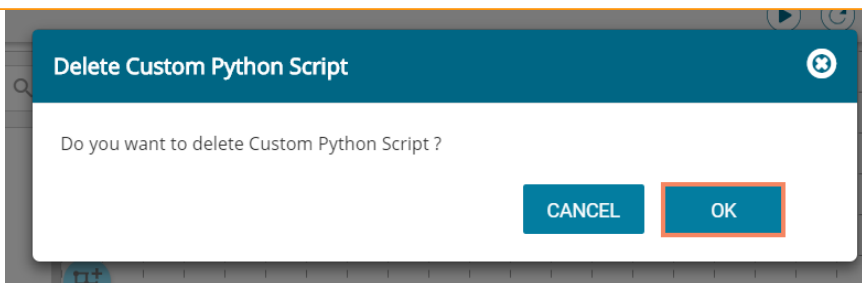
vii)   The selected Scala Script will be deleted.

7.6.2.5. Connecting Saved Python Script with a Data Source

    i)   Click the **'Custom Python Script'** tree node.
    ii)   Select and drag a saved Python script to the workspace.
    iii)   Connect the Python Script to a configured data source.
    iv)   Click the dragged **'Python Script'** component.
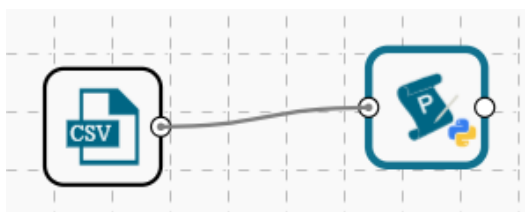
    v)   Configure the required fields in the **'Custom Group'** tab.
    vi)   Click **'APPLY'**

    vii)   After getting the success message run the workflow
    viii) Users will get the process status under the **'CONSOLE'** tab

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

6/9/2018 - 16:22:7   : Process Initiated...

6/9/2018 - 16:22:12   : CSV0 is started.

6/9/2018 - 16:22:12   : CSV0 is completed.

6/9/2018 - 16:22:12   : CustomPythonScript - quantML is started.

6/9/2018 - 16:24:33   : CustomPythonScript - quantML is completed.
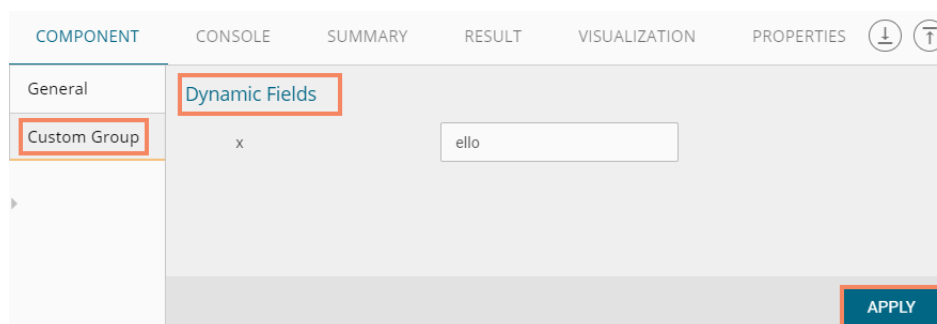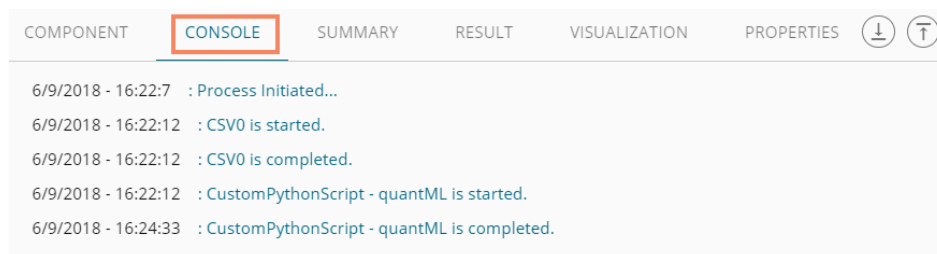
    ix)   Follow the below given steps to display the result view:
        a. Click the dragged Python component on the workspace.
        b. Click the **'RESULT'** tab.

Show [10 ▼] entries   Search: [   ]

Category	Sharpe	Mean	Risk	Skew	%up	%Down	Suggestion
ASIANPAINT	2.2030408166105375	0.14000661722622762	0.22014896192869232	-0.06900642087301212	0.75	0.25	3
BAJAJ-AUTO	-0.3177065940151844	-0.013857152174100246	0.15109092518619893	0.11717177808347531	0.5	0.5	3
COLPAL	2.251838714300893	0.07889388828628727	0.12136590604885886	0.9535998577259107	0.75	0.25	-3
ITC	0.7331135544309868	0.06519084746374554	0.30803920978740906	1.473192027990805	0.5	0.5	3
HINDUNILVR	-0.4002884334177015	-0.011890271063565994	0.10289856952410058	-0.09109831006676725	0.5	0.5	-3
HEROMOTOCO	1.3202203304714948	0.05652638362336265	0.14831852857292047	0.03267872250176619	0.6666666666666666	0.3333333333333333	3
HCLTECH	1.3869160530891287	0.03971886370384778	0.0992058456612971	-0.4683947882728144	0.6666666666666666	0.25	3
INFY	-0.3437118922664428	-0.01835622747553245	0.1850033085167015	0.5903718468849175	0.4166666666666667	0.5833333333333334	-3
LUPIN	-0.7128405424741218	-0.037619918477645675	0.18281679084561048	-0.1086621290968751	0.4166666666666667	0.5833333333333334	3
M&M	-1.4382216587471626	-0.06983137833970447	0.1681959029212423	0.32982346399266066	0.3333333333333333	0.6666666666666666	-3

Showing 1 to 10 of 13 entries   Previous   1   2   Next

x)   Click the '**VISUALIZATION**' tab to display the result data through a column chart.

xi)   Click '**SUMMARY**' tab to view a summary of the process.

Note: A new tree-node 'Pre-Defined Scripts' has been added under the 'Custom Python Script' tree-node with the list of predefined python scripts on various business verticles to facilitate the users.
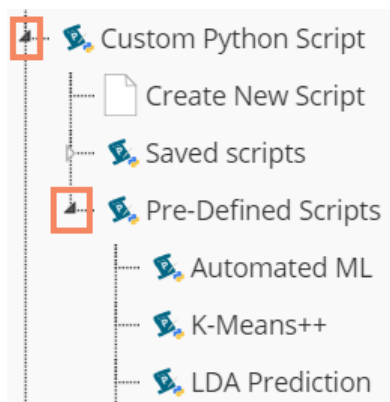
7.7. Scheduler

Scheduler helps to schedule the Predictive Workflow as per the requirement.

7.7.1. New Schedule

This section explains the steps to schedule a new job. Scheduling a new job is a continuous step by step process as described below:
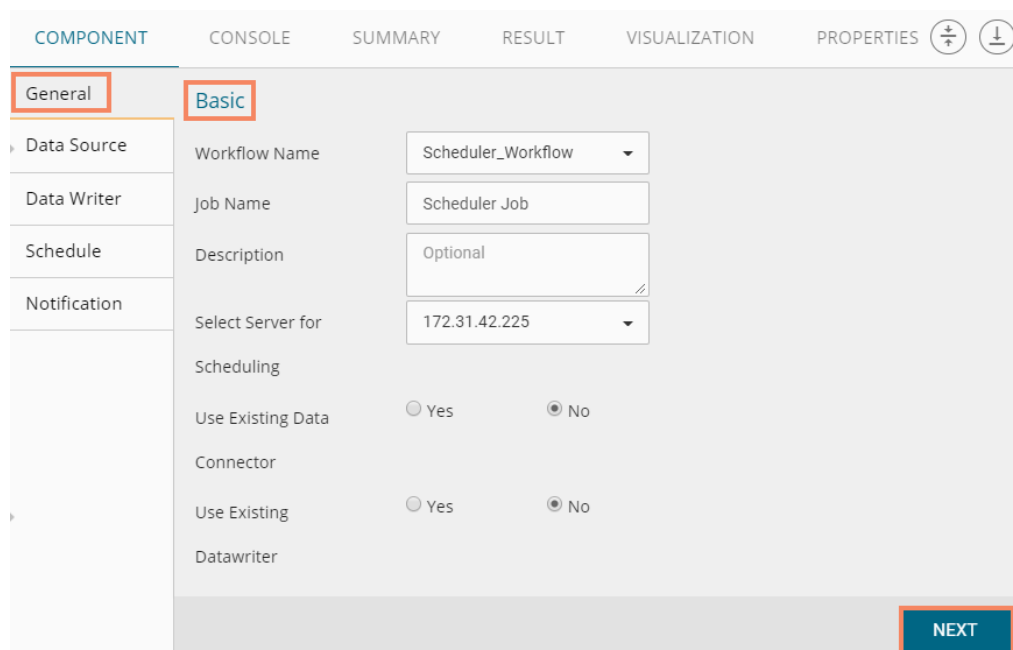
i) Navigate to the Predictive home page.
ii) Click the '**Scheduler**' tree node.
iii) Two options will be displayed:
   a. New Scheduler
   b. Status
iv) Select '**New Schedule**' from the menu.
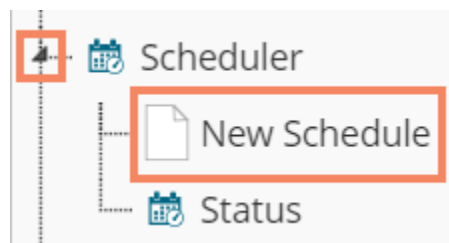
v) Users will be redirected to the '**General**' tab.

7.7.1.1. Configuring General Tab

i) A '**General**' tab will open (by default).
ii) Fill in the required information:
   a. **Model Name**: Select a model name using the drop-down menu.
   b. **Job Name**: Enter a job name.
   c. **Description**: Describe the job (optional field).
   d. **Use Existing Data Connector**: Use radio buttons to select an option.
      i. Select '**Yes**' to use an existing data connector.
      ii. Select '**No**' for not using an existing data connector.
   e. **Use Existing Datawriter**: Use radio buttons to select an option.
      i. Select '**Yes**' to use an existing data writer.
      ii. Select '**No**' for not using an existing data writer.
iii) Click '**NEXT**'

iv) Users will be redirected to the '**Data Source**' tab.

7.7.1.2. Configuring Data Source

Provide the required information to configure a data source:

i) '**General**' fields will be displayed by default.

ii) Users can fill in the required fields:

    a. Component Name: A default name provided for the component

    b. Alias Name: User can enter a name for the component

    c. Description: Users can describe the component (optional)

iii) Click '**NEXT**'

iv) Users will be redirected to the '**Properties**' fields.

v) Configure the following fields (to configure a new data source):
- a. **Select Data Connector**: Select a data connector from the drop-down menu
- b. **Select Data Service**: Select a data service from the drop-down menu
- c. Based on the selected data service the below-given columns will be displayed
  - i. Column Header
  - ii. Data Type

vi) Click '**NEXT**'

vii) Users will be redirected to the '**Conditions**' tab. (If conditions are available, else the data source configuration will end at the previous step.)

viii) Configure the required '**Conditions**' fields.

ix) Click '**NEXT**'

x) Users will be redirected to the '**Mapping**' tab.

xi) Configure the column header information from the data service that will be used for the selected model columns.

xii) Click '**NEXT**'

xiii) Users will be redirected to the '**Data Writer**' tab.

Note: The '**Data Source**' tab will be enabled, only if users select '**No**' for 'Use Existing Data Connector' option while configuring the '**General**' tab for a new schedule.

7.7.1.3. Configuring a Data Writer

The Data Writer fields are reliant on the selected data writer types. The scheduler is provided with two kinds of data writers: 1. Data Writer and 2. Elastic Search Writer.

1. Data Writer

i) Fill in the required details to configure a data writer

ii) Click '**NEXT**'

iii) Users will be redirected to the '**Schedule**' tab.

2. Data Store Writer

Users can directly use the predictive workflows to create Business Stories if the workflows are written using the Elastic Search Writer.

i) Select '**Elastic Search Writer**' as a Data Writer Type to schedule a Predictive workflow.
ii) Users will be directed to create Hierarchy Definition.
iii) Drag and drop the required dimensions to define hierarchical drill.
iv) Click '**NEXT**'

v) Users will be redirected to the '**Schedule**' tab.

Note: The ‘**Data Writer**’ tab will be enabled, only if users select ‘**No**’ for ‘**Use Existing Data Writer**’ while configuring the ‘**General**’ tab for a new schedule.

7.7.1.4. Scheduling a New job

Users can select a time to schedule a new job using this section. As per the selected scheduling time, refresh interval option will be provided.

7.7.1.4.1. Job Refresh Intervals Details

- **Hourly:** By selecting this option users can schedule the job on an hourly basis.
  1. Select a specific hour by using the below-given options:

     Every_hour: Selecting this option will refresh the scheduled job after the selected hourly interval.

     <div align="center">OR</div>

     At: Selecting this option will refresh the scheduled job at the selected hour.

- **Daily:** By selecting this option users can schedule the job on a daily basis.
  1. Select a specific day by using the below-given options:
     Every_ Days: the scheduled job will be refreshed after every selected number of days. E.g., if two is selected then, the scheduled job will be refreshed every alternate day at the set time.

     <div align="center">OR</div>

     Every Week Day: the scheduled job will be refreshed daily till the end date.
  2. Select the Start time.

- **Weekly**: By selecting this option users can schedule the job on a weekly basis.
  Select a day **or** days of the week when the scheduled job can be refreshed.

- **Monthly:** By selecting this option users can schedule the job on a monthly basis. This time **the** range can be used to set schedule refresh for more than a month.
  Select a specific day of the month by using the below given options:
  E.g., Set monthly refresh interval (E.g., the first day of every month)
  OR
  Set a specific day after the desired monthly interval (the first Monday of the every month)

- **Yearly**: By selecting this option users can schedule the job on a yearly basis. This time range is provided for jobs that run more than one year.

  Select a specific day of the month by using the below-given options:

  Set a date for any month (E.g., The 1st January of every year until it approaches the end date)

  Or

  Select a day of any month (E.g. The 1st Monday of January every year till it approaches the end date)

- **Custom Cron Expression:** Users can schedule more flexible and customizable schedule run by using the 'Custom Cron Expression' option. The scheduled workflow can be more specific with the custom cron expression that supports timing up to minutes and seconds. USers need to

  enter a valid Cron Expression in the given field.

Note:

a. By selecting the '**Use Existing Data Connector**' and '**Use Existing Data Writer**' options '**Schedule**' tab will be displayed immediately after the '**General**' tab.
b. Click '**NEXT**' after configuring the desired scheduling time to move on.

7.7.1.5. Notification

i) Configure the below-given fields:

a. **Enable Email Notification**: Use a check mark in the box to enable email

b. **Email Address**: Enable this option by check marking the box

c. **Send Mail when Server is not running**: Users can check mark in the box to enable this option. By enabling this option, users will get an email when the Python server is not running.

d. **Send Mail when Process is Completed Successfully**: Users can check mark in the box to enable this option. By enabling this option user will get mail after the process completed.

e. **Send Mail when the Process is a Failure**: Users can check mark in the box to enable this option. By enabling this option user will get an email when the process fails.

ii) Click '**APPLY**' to save the details

iii)   A success message will pop-up to assure that the job/process has been scheduled.

Process has been scheduled. ✕

iv)   The scheduled job/ process will be added to a list provided under the '**Status**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES					

🔄 Refresh

Search: _____

Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
job_sanityCheck	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏️ ✖️ ▶️
wf_sanityTest	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		Workflow_Save	iris_new	View Logs	✏️ ✖️ ▶️
jobcheckIssue	Hourly	14/Feb/2018-21:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏️ ✖️ ▶️
jobCheckJOBBBB	Hourly	14/Feb/2018-22:0:0	14/Feb/2018-23:0:0	NA	Stopped		WF_checkk	iris_new	View Logs	✏️ ✖️ ▶️
Scheduler Job	Yearly	8/Apr/2018-1:0:0	28/Apr/2019-0:0:0	1/Apr/2019-12:0:0	Active		Scheduler_Workflow	iris_Filter	View Logs	✏️ ✖️ ▶️

Showing 81 to 85 of 85 entries

Previous   1 ... 5   6   7   8   9   Next

Note:

a. The PDF summary will be sent through email for the scheduled workflows.
b. Multiple email addresses can be entered in coma separated value.
c. At present, Spark Workflows are not supported by Scheduler.

7.7.2. Status

This section will display detailed information for all the scheduled jobs.

i)   Click the '**Scheduler**' tree node.
ii)   Select '**Status**'

iii)   Users will be redirected to the Component tab.
iv)   A list containing all the scheduled jobs will be displayed.

Refresh

Search:

Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
job check sch	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		chck_sch_1	iris	View Logs	✏ ■ ✖ ▶
job sch	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		sch_check	iris	View Logs	✏ ■ ✖ ▶
job for sch333	Hourly	21/Dec/2017-20:0:0	21/Dec/2017-21:0:0	NA	Stopped		sch_check111	teadata	View Logs	✏ ■ ✖ ▶
sch	Hourly	3/Jan/2018-14:0:0	3/Jan/2018-16:0:0	NA	Stopped		CreditCard_Scoring	German_data	View Logs	✏ ■ ✖ ▶
sch	Hourly	3/Jan/2018-15:0:0	3/Jan/2018-16:0:0	NA	Stopped		samplech	iris	View Logs	✏ ■ ✖ ▶
bs_ccc	Hourly	19/Jan/2018-21:0:0	19/Jan/2018-22:0:0	NA	Stopped		check_BS_CNR	iris	View Logs	✏ ■ ✖ ▶
job_sch_mails	Hourly	29/Jan/2018-16:0:0	29/Jan/2018-17:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
check_R sch	Hourly	29/Jan/2018-17:0:0	29/Jan/2018-18:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
job_sch_auto	Hourly	29/Jan/2018-18:0:0	29/Jan/2018-19:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶
jobbbb	Hourly	29/Jan/2018-18:0:0	29/Jan/2018-19:0:0	NA	Stopped		R_sch_check	iris	View Logs	✏ ■ ✖ ▶

Showing 1 to 10 of 85 entries   Previous   1   2   3   4   5   …   9   Next

a. Click '**View Logs**' to see the logs of the selected workflow under the '**COMPONENT**' tab.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

06/Apr/2018 - 07:07:58	Data Service0 is started.
06/Apr/2018 - 07:07:58	Data Service0 is completed.
06/Apr/2018 - 07:07:58	Python-Linear Regression1 is started.
06/Apr/2018 - 07:07:58	Python-Linear Regression1 is completed.
06/Apr/2018 - 07:07:58	Internal Data Writer is started.
06/Apr/2018 - 07:07:58	Internal Data Writer is completed.

Related Actions for a Scheduled Job:

Options	Name	Description
✏	Edit	To edit/update the scheduled job details
■	Stop	To stop the scheduled job
✖	Remove	To remove the scheduled job from the list
▶	Start	To start the scheduled job

Note:

a. '**Edit**' option will allow the user to update/ edit all the tabs for the selected job.
b. Users can click the '**Start**' button to restart the scheduler for a scheduled job until it reaches the end date.
c. Users can enable '**Edit**' and '**Remove**' actions only after stopping the Scheduled job.

7.8. Saved Workflows

Users can save a workflow by clicking the '**Save**' button provided on the workspace menu row. All the saved workflows will be displayed under the '**Saved Workflow**' tree node. This section explains various options assigned to a saved workflow.

i) Navigate to the Predictive home page
ii) Click '**Saved Workflow**' tree-node

iii) A list of all the saved workflows will be displayed
iv) Right, click on a workflow from the list of '**Saved Workflows**'
v) A context menu will open with various options (As shown below):s
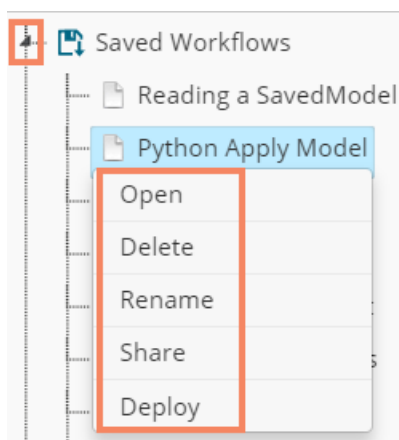
7.8.1. Opening a Workflow

i) Right-click on a workflow from the list of '**Saved Workflows**'
ii) Select '**Open**' from the context menu
iii) The selected workflow will be displayed in the right pane of the screen
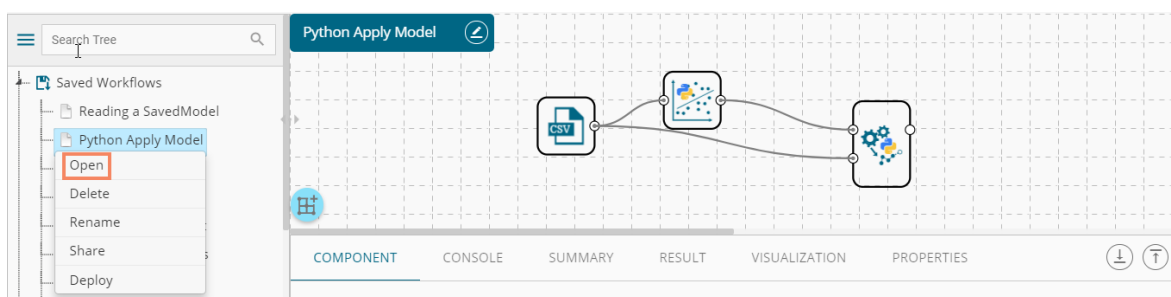
Note: The workflow name will be displayed on the left side of the workspace menu row while opening a workflow.
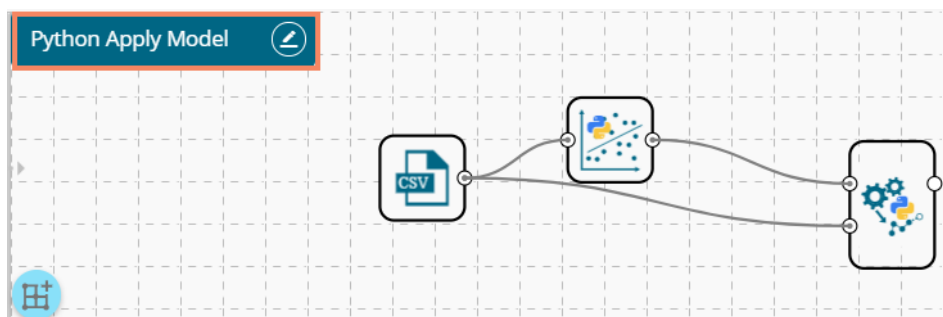
7.8.2. Deleting a Workflow

i) Right-click on a workflow from the list of '**Saved Workflows**'
ii) Select '**Delete**' from the context menu
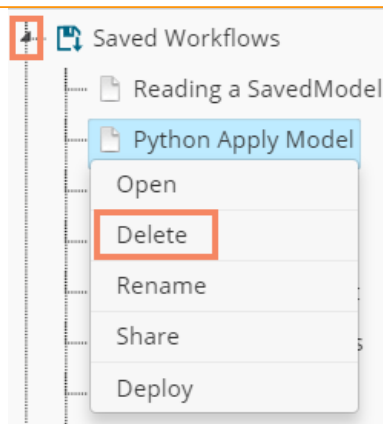
iii)   A message window will pop-up to confirm the deletion

iv)   Click '**OK**'
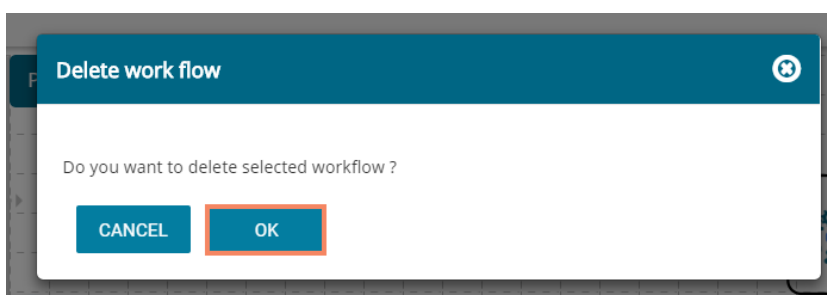
v)   The selected workflow will be removed from the list

7.8.2.1. Delete Connection in a Workflow

 A Right click on the inter-node connection will display the '**Delete Connection**' option in a workflow. Click the '**Delete Connection**' option to delete a connection.
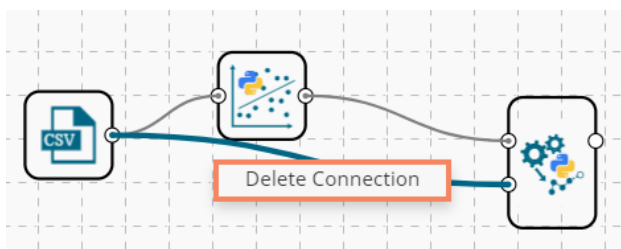
7.8.3. Renaming a Workflow

i)   Press a right click on workflow from the list of '**Saved Workflows**'

ii)   Select '**Rename**' from the context menu
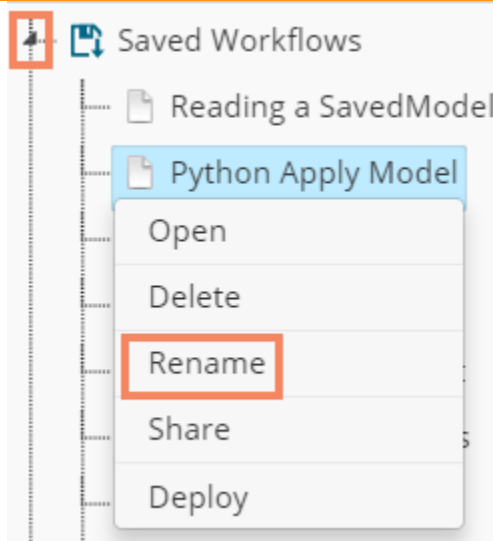
iii) A pop-up window will appear
iv) Enter a new/modified name for the workflow
v) Click '**YES**'
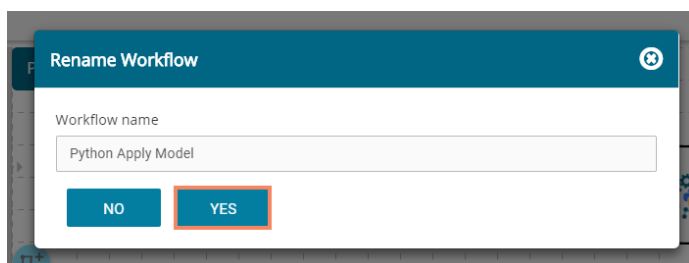
vi) The selected workflow will be renamed

7.8.4. Sharing a Workflow
This feature gives users the ability to share saved workflows with other users and groups.

The following options are available to share a selected workflow:

1. **Share With**: This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.

    i) Press a right click on workflow from the list of '**Saved Workflows**'
    ii) Select '**Share Workflow**' from the context menu
    iii) The '**Share With**' option will be displayed (by default)
    iv) Select either '**Group**' or '**Users**'
        a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
        b. Users can be excluded by not selecting a username from the list when the '**User**' option has
            been selected.
    v) Select a specific group or user from the list by check marking the box
    vi) Click '**APPLY**'

vii) The selected workflow will be shared with the chosen user(s)/group(s)

2. **Copy To**: This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the '**Copy To**' method.

   i) Press a right click on workflow from the list of '**Saved Workflows**'
   ii) Select '**Share Workflow**' from the context menu
   iii) Select '**Copy To**'
   iv) The copied workflow name will be displayed
   v) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
      b. Users can be excluded by not selecting a username from the list when the '**User**' option has been selected
   vi) Select a specific group or user from the list by check marking the box
   vii) Click '**APPLY**'

viii) The copied workflow will be shared with the chosen users/groups

7.8.5. Deploying a Workflow
The Predictive Workflows can be deployed to the BizViz Dashboard Designer.

   i) Press a right click on a Workflow from the list of '**Saved Workflows**'
   ii) Select '**Deploy**' from the context menu

iii) A success message will pop-up to assure that the workflow has been published

Workflow published successfully.

iv) The published workflows will be marked by a checkmark in the list of the 'Saved Workflows'

v) Navigate to the Dashboard Designer home page
vi) Click '**New**'
vii) Click '**Dashboard**'

viii) Users will be directed to the Dashboard canvas

ix) Click the '**Data Source**' icon 🗄 to display all the available data sources

x) Click the '**Create New Connection**' option ➕ provided next to the '**Predictive Service**' data source

xi) A new connection will be created and added below

xii) Click on the connection to display the connection specific details
xiii) Select the deployed Predictive workflow as a data source via the drop-down menu
xiv) Configure the other subsequent details:
  a. Load At Start: Enable this option to get the updated data
  b. Timely Refresh: Enable this option to refresh data
  c. Refresh Interval: Select the time interval to refresh the data

  d. Once the data connection is established the selected predictive workflow can be used as a data source to the Dashboard Designer

Note:
  a. If a deployed Predictive Workflow has a summary, it can be viewed using the Dashboard Designer tool.
  b. Dashboards created based on the deployed Python workflows also support Bokeh charts.

7.9. Saved Python Models

7.9.1. Saving a Python Model

i) Open a Python workflow
ii) Connect '**Apply Model**' component with the workflow (as shown below)
iii) Right-click on the '**Apply Model**' component
iv) A context menu will open
v) Select '**Save Python Model**'
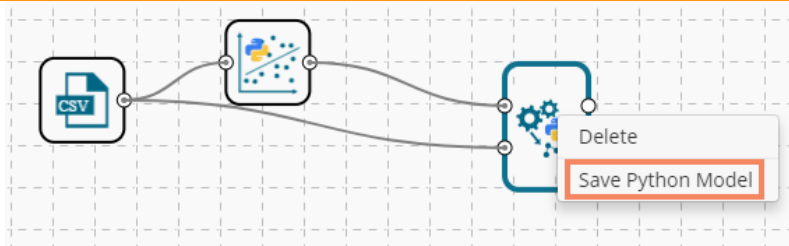
vi) A new window will pop-up

vii) Enter a name for the model that you wish to save

viii) Click '**OK**'
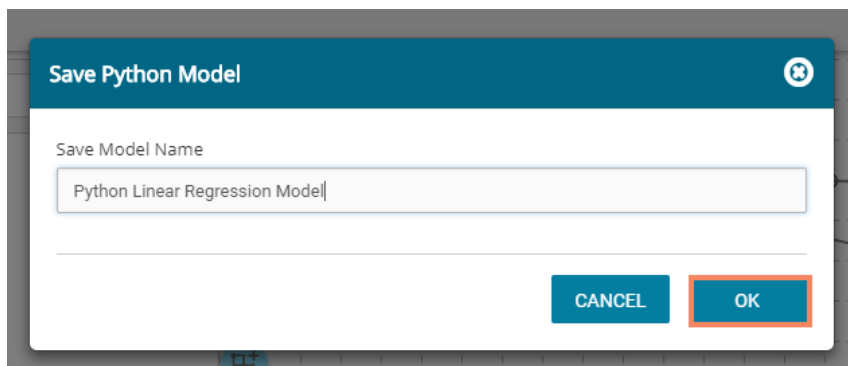
ix) A success message will pop-up at the top

x) The newly created Predictive Model will be saved to the '**Saved Python Models'** list
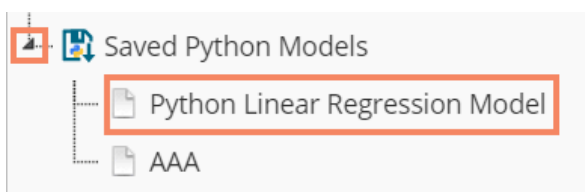
7.9.2. Reading a Python Model

Users can drag a saved model to the workspace and reuse the model for test data. A saved R model can be connected to only Apply Model and new test data source.

i) Select and drag a saved Python saved model component onto the workspace

ii) Connect the dragged model with a configured data source and an Apply Model component (As shown in the following image)
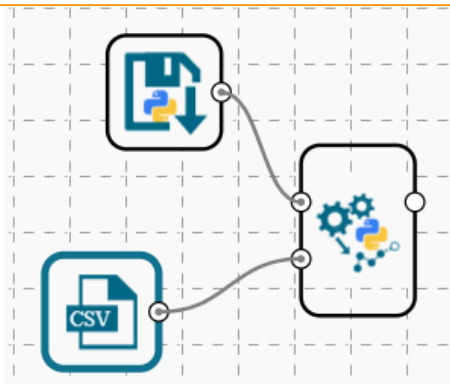
iii) Click on the dragged Saved Model component

iv) Users will be able to view the following 'Component' tabs:
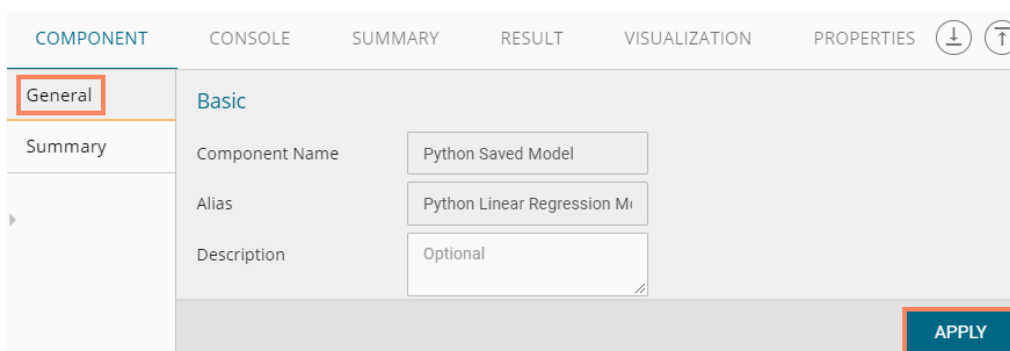
   a. General

b. Click 'Summary' tab to display the model summary

c. Click 'APPLY'

v) Configure the Apply Model component

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Basic

Component Name	Python Apply Model
Alias	Python Apply Model2
Description	Optional

APPLY

vi) After getting the success message run the workflow

vii) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

2/4/2018 - 13:9:40 : Process Initiated...

2/4/2018 - 13:9:42 : CSV1 is started.

2/4/2018 - 13:9:42 : CSV1 is completed.

2/4/2018 - 13:9:42 : Python Linear Regression Model0 is started.

2/4/2018 - 13:9:42 : Python Linear Regression Model0 is completed.

2/4/2018 - 13:9:42 : Python Apply Model2 is started.

2/4/2018 - 13:9:42 : Python Apply Model2 is completed.

viii) After the process gets completed under the Console tab, click the '**RESULT**' tab to see the result view of data

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1
1	5.1	3.5	1.4	0.2	setosa	55.62119753165504
2	4.9	3	1.4	0.2	setosa	49.90076977816
3	4.7	3.2	1.3	0.2	setosa	44.18032838608934
4	4.6	3.1	1.5	0.2	setosa	41.320114509341835
5	5	3.6	1.4	0.2	setosa	52.760983654907506
6	5.4	3.9	1.7	0.4	setosa	64.2018528004732
7	4.6	3.4	1.4	0.3	setosa	41.320114509341835
8	5	3.4	1.5	0.2	setosa	52.760983654907506
9	4.4	2.9	1.4	0.2	setosa	35.599686755846776
10	4.9	3.1	1.5	0.1	setosa	49.90076977816

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

Note:

    a. A mandatory condition to run the workflow with a '**Saved Python Model**' component is that column headers and data type of the test data source should match with the selected saved model. Users will encounter an error if validation fails while running the workflow.

    b. Users can connect a data writer to the '**Apply Model**' component in a workflow containing a saved model.

7.9.2.1. Renaming a Python Model

    i)   Select a model from the '**Saved Python Models**' list

    ii)   Right-click on the selected model

    iii)   A context menu will open

    iv)   Select '**Rename**'
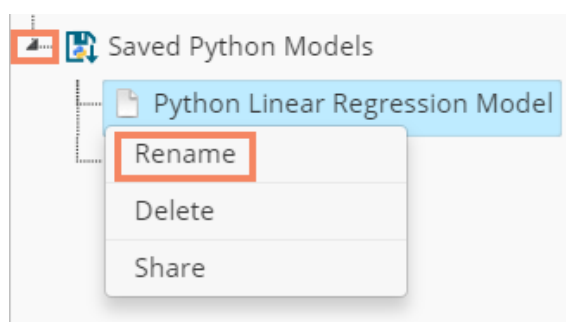
    v)   A pop-up window will appear to rename the model

    vi)   Enter a new '**Model Title**' or modify the existing model title in the given field (if desired)

    vii)   Click '**YES**'
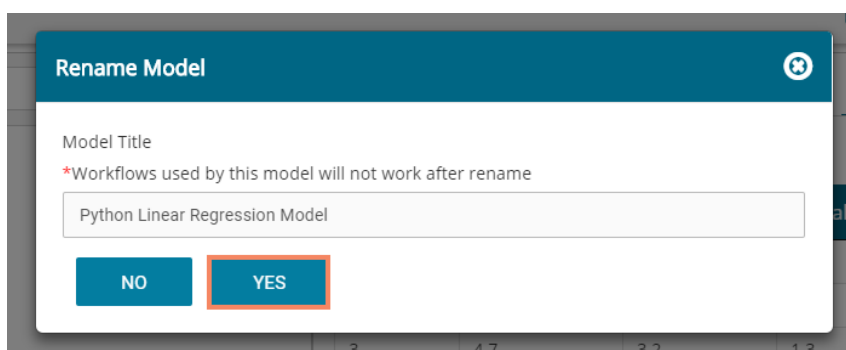
    viii)   The selected Python saved model will be renamed

7.9.2.2. Deleting a Python Model

    i)   Select a model from the '**Saved Python Models**' list

    ii)   Right-click on the selected model

    iii)   A context menu will open

    iv)   Select '**Delete**' from the menu
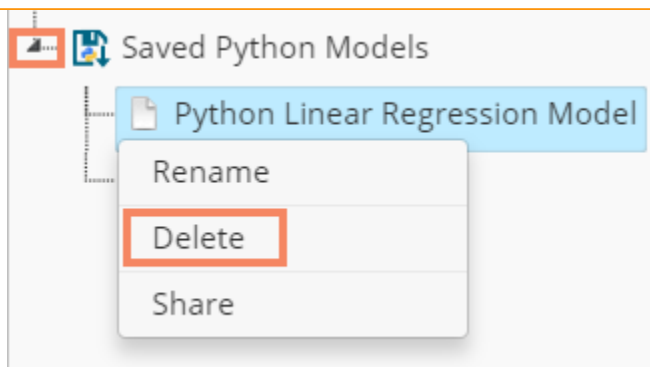
v) A pop-up window will appear to confirm the deletion
vi) Click **'OK'**
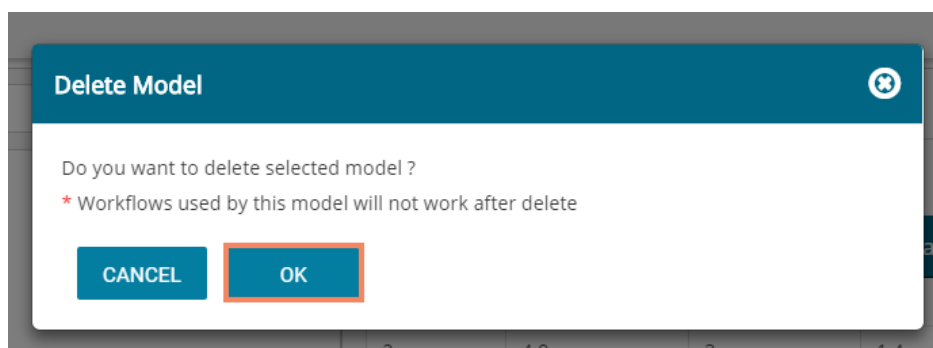
vii) The selected predictive model will be deleted and removed from the **'Saved Python Models'** list

Note: After renaming or deleting a Saved R Model, workflows used by the same model will not work.

7.9.2.3. Sharing a Python Model

Users can share a saved model with other users or user groups. There are two options to share a selected model:

1. **Share With:** This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.

i) Use right-click on a model from the list of **'Saved Models.'**
ii) Select **'Share Model'** from the context menu.
iii) The **'Share With'** option will be displayed (by default).
iv) Select either **'Group'** or **'Users'** option.
   a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
   b. Users can be excluded by not selecting a username from the list when the **'User'** option has been selected.
v) Select a specific group or user from the list by check marking the box.
vi) Click **'APPLY'**
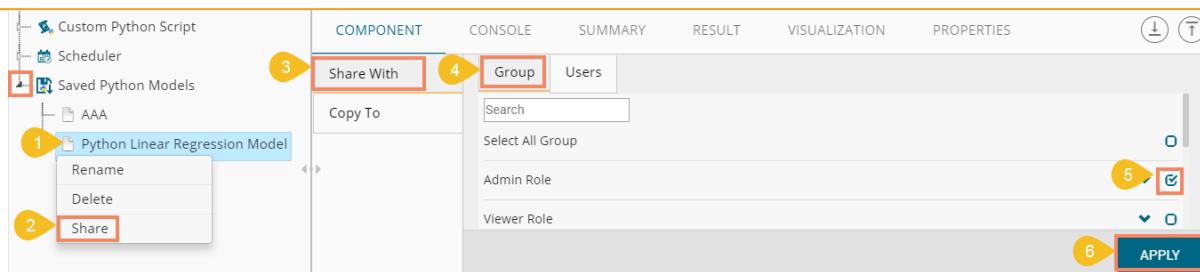
vii) The saved Spark model will be shared with the selected group or users.

2. **Copy To:** This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the '**Copy To**' method.

i) Right, click on workflow from the list of '**Saved Models**'
ii) Select '**Share Model**' from the context menu
iii) Select '**Copy To**' option
iv) The copied model name will be displayed
v) Select either '**Group**' or '**Users**' option with a click
    a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group

    b. Users can be excluded by not selecting a username from the list when   the '**Users**' option has been selected
vi) Select a specific group or user from the list by check marking the box
vii) Click '**APPLY**'
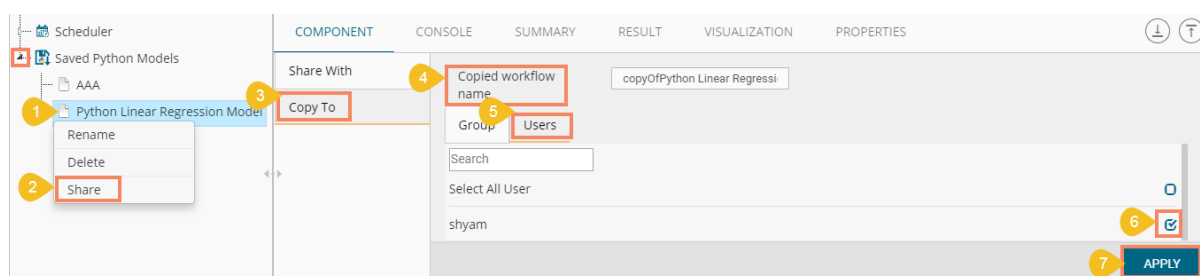
viii) A copy of the model will be shared with the selected user or group

8. JAVA Data Preparation

Users can select the Data Preparation Workspace from the landing page of the Predictive Workbench.
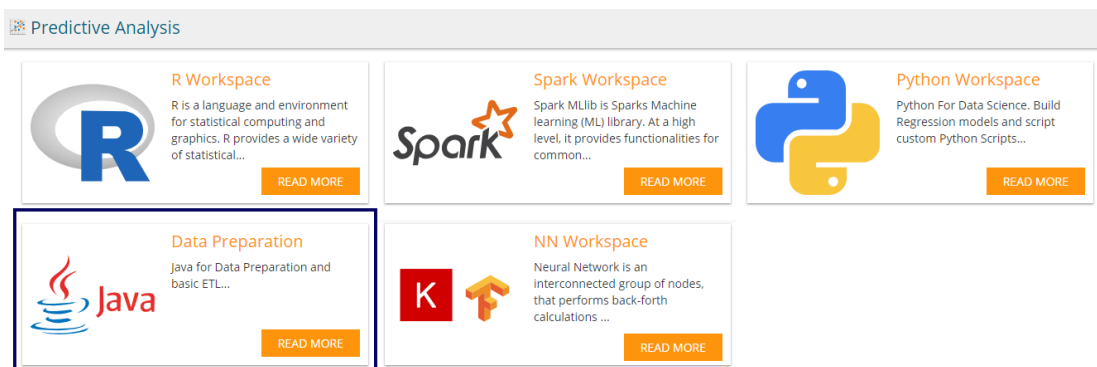
Users will be redirected to the following screen by clicking the Data Preparation Workspace:
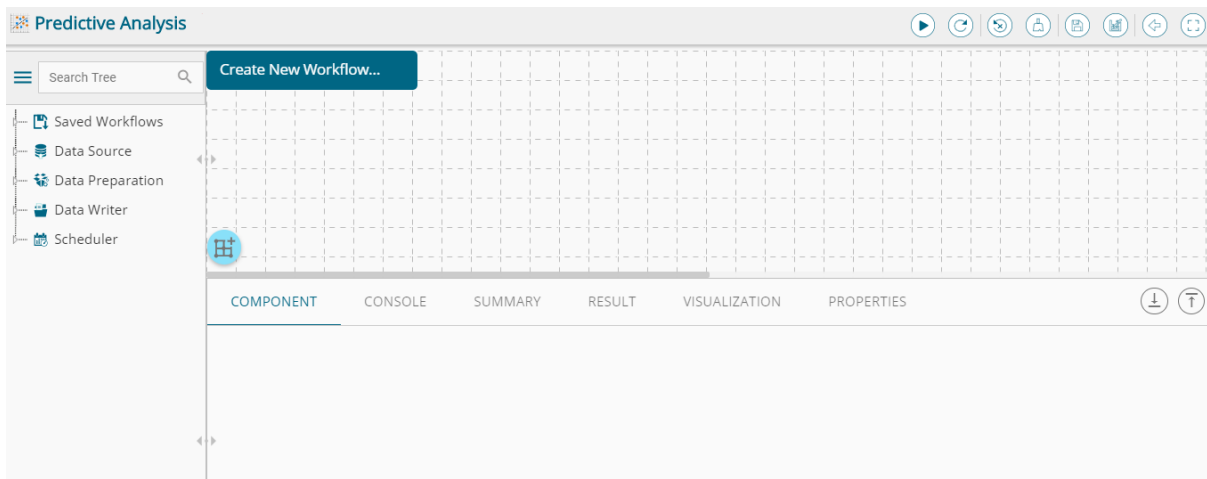
8.1. Getting Data from a Data Source

Acquiring data from a data source is the initial step in Predictive Analysis. The '**Data Source**' tree node offers three types of data connectors:

a. CSV File
b. Data Service
c. Cassandra Reader

8.1.1. Getting Data from a CSV File

i) Select and drag '**CSV File**' component onto the workspace.
ii) Click the '**CSV File**' component.

iii) Configure the following '**CSV Properties Configuration**' fields:

a. **Select File**: Browse a CSV file
b. **Delimiter:** Mention the delimiter used in the CSV file

iv) Click **'APPLY'**

v) Users should get the 'Apply Successful' message as displayed in the following image:

vi) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache

vii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix) Follow the below given steps to display the result view:
   a. Click the dragged data source component on the workspace.
   b. Click the '**RESULT**' tab.

Show 10 ▼ entries   Search:

RowID	SLength	SWidth	PLength	PWidth
1	5.1	3.5	1.4	0.2
2	4.9	3	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

- **Rules to be followed while uploading a CSV File**
  1. The first row provided in the CSV file should contain the column headers.
  2. The second row of the CSV file should contain the data under all the headers without any 'null' or 'NA.'
  3. CSV headers should not have space. It should be a single word or two words concatenated by an underscore (_).
  4. CSV headers should not contain any special characters. E.g. - %, #, $, @,*, etc.
  5. CSV headers should not contain single or double quotes, dot, brackets, and high-fen.
  6. CSV headers should not contain merely numbers. Numerals should be used with at least one alphabet.
  7. CSV header should not exceed 50 characters.
  8. All rows in a column should have the same data type.

  Note:
  a. The supported file types will be .csv,.tsv
  b. '**General**' tab is provided to configure the following information for any tree-node component:

      i. Component Name: The predefined name of the component is displayed in this field
      ii. Alias Name:
      iii. Description (it is an optional field)
          (E.g. the following image displays '**General**' tab for a CSV data source.)

General

Properties

Basic

Component Name   CSV

Alias   CSV0

Description   Optional

APPLY

8.1.2. Getting Data from a Data Service

i)   Select and drag '**Data Service**' component onto the workspace.

ii) Click the '**Data Service**' component.

iii) Users will be redirected to the '**Properties**' fields provided under '**Components**' tab on the Tabbed Menu Strip.

iv) Configure the '**Data Service Properties**':

   a. **Select Data Connector**: Select a data source from the drop-down menu

   b. **Select Data Service**: Select a query service from the drop-down menu

   c. **Fields**:

      The following tables will be displayed:

       i. Column Header

       ii. Data Type

v)   Click '**NEXT**' (The '**NEXT**' option will appear only for the data service that has filters, otherwise the '**APPLY**' option will be displayed)

vi) Users will be redirected to the '**Conditions**' tab. (If the selected data service contains the filter values).

vii) Configure the following information:

   a. **Filter Type:** Available filter(s) in the data service will be displayed in this space.

   b. **Control Type:** Users are provided with the following options to pass the filter values under this option:

- **Text:** By selecting this option users can manually enter multiple filter values separated by comma

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | ⊥ ⊤ |

General	Filter Name	Control Type	
Properties	val1	Text ▼	Sepal Length
Conditions			
			APPLY

- **LOV:** By selecting this filter value option users will be directed to **choose** another Data Connector and Data Service available in the space

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES | |

General	Filter Name	Control Type	
Properties	val1	LOV ▼	
Conditions	Select Data Connector	Select ▼	
	Select Data Service	Select ▼	
			APPLY

viii) Click '**APPLY**'

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

19/6/2018 - 11:43:15   : Process Initiated…

19/6/2018 - 11:43:16   : Data Service0 is started.

19/6/2018 - 11:43:17   : Data Service0 is completed.

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:

   a. Click the dragged data source component on the workspace

   b. Click the '**RESULT**' tab

Show 10 ▼ entries   Search:

id	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.1	3.6	1.4	0.2	setosa
6	5.1	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

- **Rules to be Followed while Creating a Data Service**
  1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
  2. Data service header should not contain any special characters. E.g. - %, #, $, @,*, etc.
  3. Data service header should not contain single or double quotes, dot, brackets, and high-fen.
  4. Data service header should not contain merely numbers. Numerals should be used with at least one alphabet.
  5. Data service header should not exceed 50 characters.

Note:
  a. Users can develop a data service via the Data Management module of the BizViz Platform.
  b. **'Fields'** option under **'Properties'** tab will appear only after selecting the appropriate query service.
  c. LOV service provided under the **'Conditions'** tab can contain only one column, in case of more than one column, a warning message will appear.
  d. Users can configure the following information for a data service data source via **'General'** tab:
      i. Alias Name
      ii. Description (it is an optional field)

8.1.3. Getting Data from a Cassandra Reader
  i) Select and drag **'Cassandra Reader'** connector onto the workspace.
  ii) Click on the **'Cassandra Reader'** connector.

iii) Users will be redirected to the '**Properties**' tab of the component.
iv) Configure the required properties:
   a. Select Data Connector: Select a data connector using the drop-down menu
   b. Host Name: Data connector specific hostname will be displayed
   c. Port Number: Port number will be displayed
   d. User Name: Username will be displayed
   e. Password: Enter the password
   f. Cluster Name: Enter a cluster name
   g. Select Key Space: Select a keyspace from the drop-down menu
   h. Select Table: Select a table from the drop-down menu
   i. Limit No. of row to fetch: Select an option using the drop-down menu. Two options will be provided as shown below:
      1. Select all Rows
      2. Limit By
   j. Max. No. of Rows to be fetched: Enter a number to decide maximum fetched rows. (This option will appear only if '**Limit By**' option has been selected using the '**Limit by Row**' field. The Default value for this field is 1000).
v) Click '**NEXT**'

vi) Users will be redirected to the '**Column Selection**' tab
vii) Select the required columns from the list
viii) Click '**APPLY**'

ix) Click the 'Run' ▶ icon or click 'Refresh' ↻ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the 'CONSOLE' tab to display the progress of the process

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION |

19/6/2018 - 12:25:16   : Process Initiated...

19/6/2018 - 12:25:17   : cassandra0 is started.

19/6/2018 - 12:26:31   : cassandra0 is completed.

xi) After the Console process gets completed, users can view the result data using the 'RESULT' tab

xii) Follow the below given steps to display the result view:

a. Click the dragged data source component on the workspace

b. Click the 'Result' tab

Show 10 ▼ entries   Search:

Number	PetalLength	PetalWidth	SepalLength	SepalWidth	cat
6	1.7	0.4	5.4	3.9	0
80	3.5	1	5.7	2.6	1
75	4.3	1.3	6.4	2.9	1
57	4.7	1.6	6.3	3.3	1
113	5.5	2.1	6.8	3	1
67	4.5	1.5	5.6	3	1
118	6.7	2.2	7.7	3.8	1
82	3.7	1	5.5	2.4	1
120	5	1.5	6	2.2	1
112	5.3	1.9	6.4	2.7	1

Showing 1 to 10 of 150 entries   Previous   1   2   3   4   5   ...   15   Next

Note: The Apache Spark workflows require a 'Cassandra Reader' as a data source. The Cassandra Reader can also be used as a data source for the R Workflows.

8.1.4. Removing a Data Source from the Workspace

i) Right-click on the data source connector (in the workspace)
ii) A context menu appears
iii) Click the **'Delete'** option

iv) The selected Data Source component will be removed from the workspace

<div align="center">OR</div>
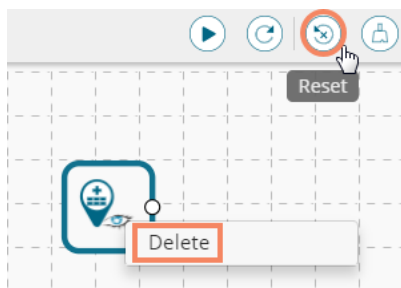
Click on the **'Reset'** icon to remove the connector(s) from the workspace

Note: The same set of steps can be followed to remove any data source type in the given tree-node menu.

8.2. Data Preparation

Components provided under the **Data Preparation** tree-node help in preparing the raw data from the data source and make it suitable for analysis. They organize data to gain accurate result out of it.

8.2.1. Data Type Definition

The Data Type Definition option can be used to change the name, data type of the data source column. This component helps users to prepare data and make it suitable for further analysis.

i) Navigate to the Predictive homepage
ii) Click **'Data Preparation'** tree-node
iii) A context menu opens

iv) Drag **'Data Type Definition'** component and connect it to a configured data source onto the workspace.
v) Click the **'Data Type Definition'** component (in the workspace).

vi)   Users will be redirected to the '**Properties**' tab.

vii)   Configure the following 'Data Type Mapping' details:

      a. **Column Name:** Select a column name which you want to change

      b. **Alias Name:** Enter an alias name for the required source column

      c. **Primary Data Type:** Select a primary data type column that you want to change

      d. **Date Format:** Select a date format that you want to display (Date format is optional for date Data Type)

      e. **'Add' option** : Click on this button to add one more row of the '**Data Type Mapping**' fields

viii)   Click '**APPLY**'.

ix)   After getting the success message run the workflow

x)   Users will get the process status under the '**CONSOLE**' tab

xi)   After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii)   Follow the below given steps to display the result view:

      a. Click the dragged Data Type Definition component in the workspace.

      b. Click the '**RESULT**' tab.

xiii)   Users can see the given column names on the selected columns in the '**RESULT**' data.

8.2.2. Filter
This option is used to filter the data by column or row.

Column Filter
i)   Select and Drag '**Filter**' component onto the workspace
ii)   Connect the '**Filter**' component to a configured data source component

iii)   Configure the filter component as described below:
    a. Select a column from the '**Selected Columns**' context menu
iv)   Click '**APPLY**' to configure the data

v) After getting the success message run the workflow
vi) Users will get the process status under the '**CONSOLE**' tab

vii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
viii) Follow the below given steps to display the result view:
  a. Click the dragged algorithm component in the workspace
  b. Click the '**RESULT**' tab
ix) The filtered data will be displayed via the '**RESULT**' tab

Row Filter

i) Drag and connect the '**Filter**' component onto the workspace
ii) Connect the '**Filter**' component to a configured data source
iii) Click the '**Filter**' component
iv) The '**Column Filter**' tab will be displayed (by default)
v) Select a column using the context menu
vi) Select '**Row Filter**' tab from the '**Component**' menu list
vii) Configure the required fields:
  a. Double click on the components from **Columns, Operators,** and **Functions** in the sequence as shown in the image below
  b. A formula will be entered in the given box (E.g., in this case, the entered formula is [Number]>SELECT(2))
viii) Click '**APPLY**'
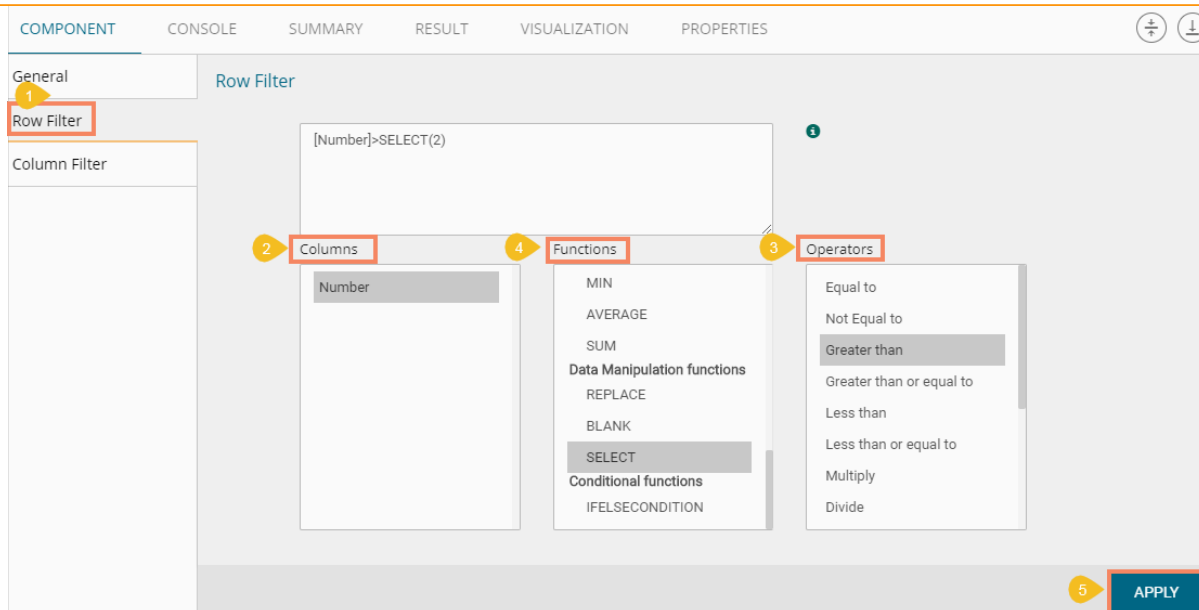
ix) After getting the success message run the workflow

x) Users will get the process status under the '**CONSOLE**' tab
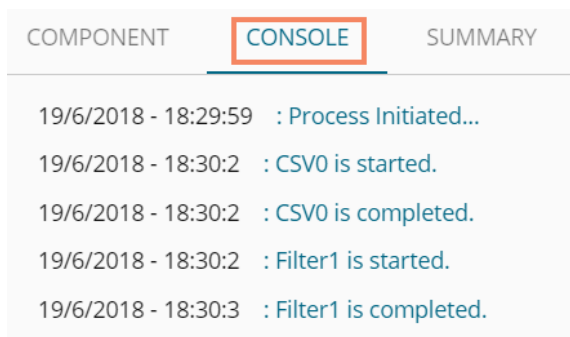
xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:

    a. Click the dragged data preparation component on the workspace

    b. Click the '**RESULT**' tab

xiii) The filtered data as per the applied formula will be displayed via the '**RESULT**' tab
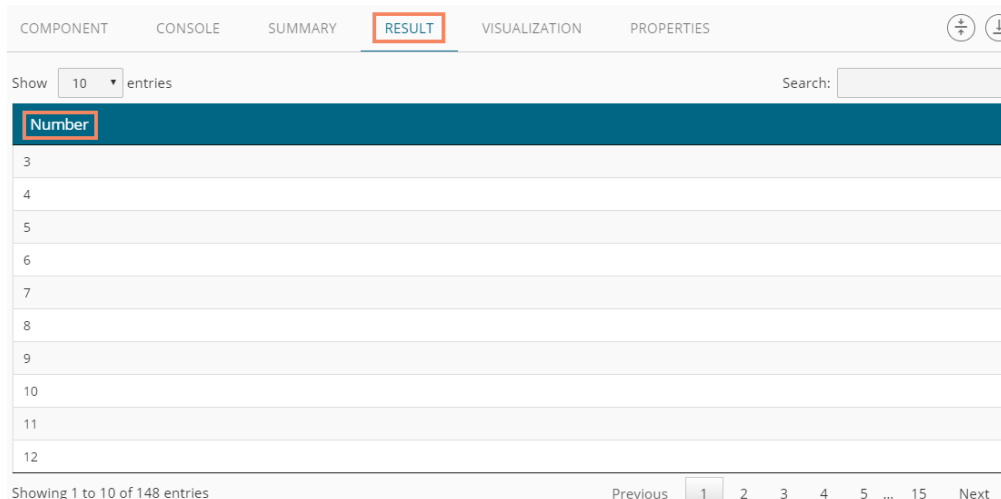
Note:

    a. The expression should retain Boolean output.
    b. Users can not use Data manipulation functions.

8.2.3. Formula

Users can create a calculated column using **'Formula.'** A formula can be formed by using available columns, functions, and operators.

i)   Select and drag **'Formula'** component onto the workspace
ii)   Connect the **'Formula'** component to a configured data source
iii)   Click on the **'Formula'** component
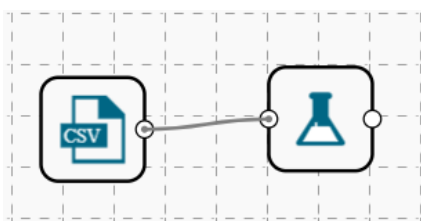
iv)   Configure the required component fields to apply a formula:
    a. **'Columns,'** **'Functions,'** and **'Operators':** Double click on these lists will enter a formula in the given box
    b. **Formula Name:** Enter a formula name in the given field
    c. Click **'APPLY'** to configure the formula

v)   After getting the success message run the workflow
vi)   Users will get the process status under the **'CONSOLE'** tab

COMPONENT	CONSOLE	SUMMARY	RESULT

20/6/2018 - 14:57:9 : Process Initiated...

20/6/2018 - 14:57:11 : CSV0 is started.

20/6/2018 - 14:57:11 : CSV0 is completed.

20/6/2018 - 14:57:11 : Formula1 is started.

20/6/2018 - 14:57:12 : Formula1 is completed.

vii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

viii) Follow the below given steps to display the result view:

    a. Click the dragged data preparation component on the workspace

    b. Click the '**RESULT**' tab

ix) A new Formula column is added to the result data

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	FormulaColumn
1	5.1	3.5	1.4	0.2	setosa	1.45714285714286
2	4.9	3	1.4	0.2	setosa	1.63333333333333
3	4.7	3.2	1.3	0.2	setosa	1.46875
4	4.6	3.1	1.5	0.2	setosa	1.48387096774194
5	5	3.6	1.4	0.2	setosa	1.38888888888889
6	5.4	3.9	1.7	0.4	setosa	1.38461538461538
7	4.6	3.4	1.4	0.3	setosa	1.35294117647059
8	5	3.4	1.5	0.2	setosa	1.47058823529412
9	4.4	2.9	1.4	0.2	setosa	1.51724137931034
10	4.9	3.1	1.5	0.1	setosa	1.58064516129032

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

8.2.4. Normalization

This component controls the relevant data. It attempts to convert the available data from a larger Range to a smaller range. It can be done over numerical columns.

8.2.4.1. Min-Max Normalization

It implements a linear transformation of the original data values and sets a new range for all the data values to fit in. The user can fix the New Maximum and New Minimum Value for the data from the new field. Consequently, each value "v" from the original interval will be mapped into value "new_v" following the below-given formula:

$$new\_v = \frac{v - min_x}{max_x - min_x} \cdot (new\_max_x - new\_min_x) + new\_min_x$$

i) Select and drag '**Normalization**' component onto the Workspace.

ii) Connect the '**Normalization**' component to a configured data source.

iii) Click the '**Normalization**' component.

iv)   Configure the following component fields:

Properties
a. **Column Selection**
   i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)
b. **Behavior**
   i. **Normalization Type:** Select '**Min-Max**' normalization type from the drop-down menu
   ii. **New Maximum:** Set a new maximum value (Default value for this field is 1)
   iii. **New Minimum:** Set a new minimum value (Default value for New Minimum field is 0)
v)   Click '**APPLY**'

vi)   After getting the success message run the workflow
vii)   Users will get the process status under the '**CONSOLE**' tab

viii)   After the Console process gets completed, users can view the result data using the 'RESULT' tab
ix)   Follow the below given steps to display the result view:
   a.   Click the dragged Formula component in the workspace.
   b.   Click the '**RESULT**' tab.

Show 10 ▾ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	22.2222222222222	3.5	1.4	0.2	setosa
2	16.6666666666667	3	1.4	0.2	setosa
3	11.1111111111111	3.2	1.3	0.2	setosa
4	8.33333333333333	3.1	1.5	0.2	setosa
5	19.4444444444444	3.6	1.4	0.2	setosa
6	30.5555555555556	3.9	1.7	0.4	setosa
7	8.33333333333333	3.4	1.4	0.3	setosa
8	19.4444444444444	3.4	1.5	0.2	setosa
9	2.77777777777779	2.9	1.4	0.2	setosa
10	16.6666666666667	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

8.2.4.2. Zero-Score

This normalization also is known as **'Zero Mean Normalization'** is calculated on the **'mean'** and **'standard deviation'** for each attribute. It determines whether a specific value is above or below average. It also signifies the exact proportion of the variance from the fixed limit of aver3age. After applying **'Zero-Score'** normalization, each feature will have a mean value of zero (0). The unit of each value will be the number of (estimated) standard deviations away from the (estimated) mean. Zero score normalization may be sensitive to small values of 'σ_x' new value the **'new_v'** can be found by using the following expression:

$$new\_v = \frac{v - \mu_x}{\sigma_x}$$

i)   Select and drag **'Normalization'** component onto the Workspace
ii)   Connect the **'Normalization'** component to a configured data source
iii) Click the **'Normalization'** Component
iv)   Configure the required component fields:

      Properties
      a. **Column Selection**
         i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)
      b. **Behavior**
         i. **Normalization Type:** Select **'Zero-Score'** normalization type from the drop-down menu
v)   Click **'APPLY'**

General

Properties

Column Selection

Select a Column   SepalLength   ▾   ⓘ

Behavior

Normalization Type   , Zero-Score   ▾   ⓘ

APPLY

vi) After getting the success message run the workflow

vii) Users will get the process status under the '**CONSOLE**' tab

COMPONENT	CONSOLE	SUMMARY	RESULT

20/6/2018 - 15:18:4   : Process Initiated...

20/6/2018 - 15:18:5   : CSV0 is started.

20/6/2018 - 15:18:5   : CSV0 is completed.

20/6/2018 - 15:18:6   : Normalization1 is started.

20/6/2018 - 15:18:7   : Normalization1 is completed.

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix) Follow the below given steps to display the result view:

    a. Click the dragged algorithm component in the workspace.

    b. Click the '**RESULT**' tab.

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

Show [10 ▾] entries   Search: [ ]

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	-0.897673879196766	3.5	1.4	0.2	setosa
2	-1.13920048346495	3	1.4	0.2	setosa
3	-1.38072708773314	3.2	1.3	0.2	setosa
4	-1.50149038986724	3.1	1.5	0.2	setosa
5	-1.01843718133086	3.6	1.4	0.2	setosa
6	-0.535383972794483	3.9	1.7	0.4	setosa
7	-1.50149038986724	3.4	1.4	0.3	setosa
8	-1.01843718133086	3.4	1.5	0.2	setosa
9	-1.74301699413542	2.9	1.4	0.2	setosa
10	-1.13920048346495	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 ... 15 Next

8.2.4.3. Decimal-Scaling

The decimal point of the value of each element is moved in accord with its maximum absolute value. A modified value '**new_v**' can be obtained using the following formula:

$$new\_v = \frac{v}{10^c}$$

Note: In the decimal-scaling expression '**c**' is the smallest integer so that max(new_v) < 1.

i) Select and drag '**Normalization**' component onto the Workspace.

ii) Connect the '**Normalization**' component to a configured data source.

iii) Click the '**Normalization**' Component.

iv) Configure the required component fields:

Properties

a. **Column Selection**

         i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column will be selected)

    b. **Behavior**

         i. **Normalization Type:** Select '**Decimal Scaling**' normalization type from the drop-down menu.

v)   Click '**APPLY'** to configure the fields:

vi)   After getting the success message run the workflow

vii)   Users will get the process status under the '**CONSOLE**' tab

viii)   After the Console process gets completed, users can view the result data using the 'RESULT' tab

ix)   Follow the below given steps to display the result view:

    a. Click the dragged data preparation component on the workspace

    b. Click the '**RESULT**' tab

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	0.51	3.5	1.4	0.2	setosa
2	0.49	3	1.4	0.2	setosa
3	0.47	3.2	1.3	0.2	setosa
4	0.46	3.1	1.5	0.2	setosa
5	0.5	3.6	1.4	0.2	setosa
6	0.54	3.9	1.7	0.4	setosa
7	0.46	3.4	1.4	0.3	setosa
8	0.5	3.4	1.5	0.2	setosa
9	0.44	2.9	1.4	0.2	setosa
10	0.49	3.1	1.5	0.1	setosa

Showing 1 to 10 of 150 entries   Previous 1 2 3 4 5 … 15 Next

Note:

a. Normalization displays columns containing only numerical data.

b. **'New Maximum Value'** must be greater than **'New Minimum Value.**

8.2.5. Sample

This component can be used to select a subsection of data from a large dataset. The sample component supports the following sample types:

8.2.5.1. Sampling Methods

1. **First N:** It will select the first N records from the data source. E.g., If the chosen value for "N" is 10, then it will select the first ten records from the data.

2. **Last N:** It will select the last N records from the data source. E.g., If the chosen value for "N" is 5, then it will select the last five records from the data.

3. **Every Nth:** It will select every Nth record from the data source, wherein "N" indicates an interval. E.g., If N=3, then 3rd, 6th, and 9th records will be selected from the data.

4. **Simple Random:** It will select records randomly as per the value of "N" or percentage mentioned for "N" from the data source. E.g., If the selected value for "N" is four then, it will select randomly any four records from the data source. If the selected value for "N" is 4% then, it will select 4% of records from the data source.

5. **Systematic Random:** It will select data based on the bucket size. E.g., If the chosen value for the bucket is two then, it will select 1st, 3rd, 5th records or 2nd, 4th, 6^{th}records from the data source.

8.2.5.2. Steps to Apply a Sampling Method

i) Select and drag **'Sample'** component onto the workspace
ii) Connect the **'Sample'** component to a configured data source
iii) Click the **'Sample'** component

iv) Configure the required component fields:
Properties
a. **Sampling Information**
   i. **Sampling Type**: Select an option from the drop-down menu
   ii. **Limit Rows by** Select an option from the drop-down menu. This field will offer two options as described below:
      1. **Numbers of Rows**: By selecting this option, it will display a new field '**Number of Rows.**'
      2. **Percentage of Rows**: By selecting this option, it will display the new field '**Percentage of Rows.**'
b. **Sample Size Limit**
   i. **Maximum Rows**: The maximum number of rows that can be viewed in the 'RESULT' tab (It is an optional field)

v) Click '**APPLY**'

vi)   Run the workflow after getting the success message

vii)   Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

viii)   After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix)   While accessing the '**Result**' tab, Users will be displayed as a result view based on the selected Sampling Type

8.2.5.3. Result View for the Available Sampling Methods

    1. **First N (Where 'N' is 1 number of row)**

2. Last N ('N' is 5% and maximum rows are 6)

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
136	7.7	3	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6	3	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica

Showing 1 to 7 of 7 entries   Previous 1 Next

3. **Every Nth (Interval is 3, and the maximum rows are 7)**

COMPONENT   CONSOLE   SUMMARY   RESULT   VISUALIZATION   PROPERTIES

General

Properties

Sampling Information

Sampling Type   Every Nth   ▼

Step Size   3

Sample Size Limit

Maximum Rows   7

APPLY

COMPONENT   CONSOLE   SUMMARY   **RESULT**   VISUALIZATION   PROPERTIES

Show 10 ▼ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
7	4.6	3.4	1.4	0.3	setosa
10	4.9	3.1	1.5	0.1	setosa
13	4.8	3	1.4	0.1	setosa
16	5.7	4.4	1.5	0.4	setosa
19	5.7	3.8	1.7	0.3	setosa

Showing 1 to 7 of 7 entries   Previous 1 Next

4. **Simple Random (the 'Number of Rows' are 3). The randomly selected any three rows will be displayed.**

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

General

Properties

Sampling Information

Sampling Type	Simple Random ▾
Limit Rows by	Number of Rows ▾
Number of Rows	4

Sample Size Limit

| Maximum Rows | 10 |

APPLY

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Show 10 ▾ entries   Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
65	5.6	2.9	3.6	1.3	versicolor
72	6.1	2.8	4	1.3	versicolor
96	5.7	3	4.2	1.2	versicolor
109	6.7	2.5	5.8	1.8	virginica

Showing 1 to 10 of 10 entries   Previous 1 Next

5. **Systematic Random (Bucket Size is 3).**

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

General

Properties

Sampling Information

| Sampling Type | Systematic Random ▾ |
| Bucket Size | 3 |

Sample Size Limit

| Maximum Rows | 10 |

APPLY

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
2	4.9	3	1.4	0.2	setosa
5	5	3.6	1.4	0.2	setosa
8	5	3.4	1.5	0.2	setosa
11	5.4	3.7	1.5	0.2	setosa
14	4.3	3	1.1	0.1	setosa
17	5.4	3.9	1.3	0.4	setosa
20	5.1	3.8	1.5	0.3	setosa
23	4.6	3.6	1	0.2	setosa
26	5	3	1.6	0.2	setosa
29	5.2	3.4	1.4	0.2	setosa

Showing 1 to 10 of 10 entries

Data Writers are provided to store the results of the predictive analysis in flat files or databases for further in-depth analysis.

8.3. Data Writers

8.3.1. File Writer

Users can write output data to flat files like CSV, TEXT, and DAT files using the File Writer.

8.3.1.1. CSV Writer

i) Click **'TreeNode'** provided next to the **'Data Writer'** option
ii) Select **'File Writer'** option
iii) Select and drag **'CSV Writer'** component to the workspace
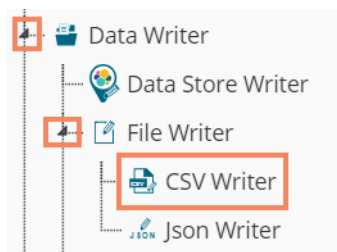
iv) Connect the **'CSV Writer'** to a configured data source or a valid workflow
v) Click on CSV Writer component to access component properties.
vi) Enter **'File Name'** in the displayed field.
vii) Click **'APPLY'**
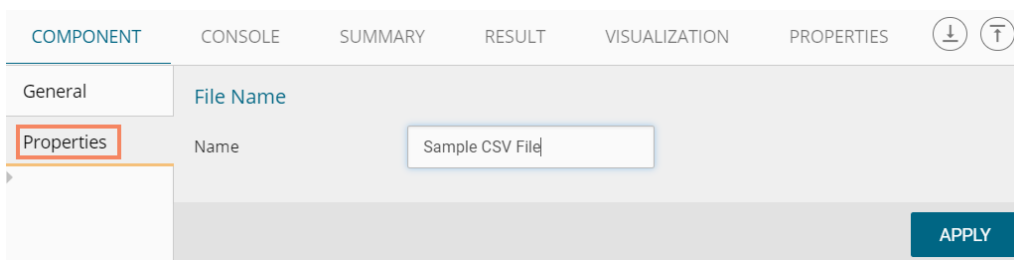
viii) After getting the success message run the workflow
ix) Users will get the process status under the **'CONSOLE'** tab

| COMPONENT | CONSOLE | SUMMARY | RESULT |

13/4/2018 - 17:46:7   : Process Initiated...

13/4/2018 - 17:46:8   : Data Service0 is completed.

13/4/2018 - 17:46:8   : Filter1 is started.

13/4/2018 - 17:46:9   : Filter1 is completed.

13/4/2018 - 17:46:9   : CSV File Writer2 is started.

13/4/2018 - 17:46:9   : CSV File Writer2 is completed.

x)   The data will be written in the CSV File
xi)   Click the '**CSV Writer**' component
xii)   A pop-up message will appear with a link to download the CSV file

Download File   ✕

Click Here to download Sample CSV File.csv
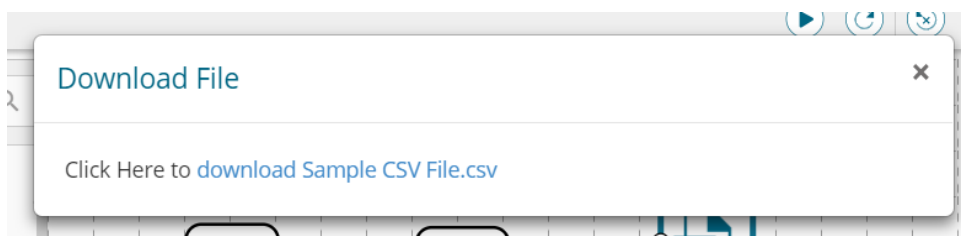
xiii)   Click the link to download the CSV file.

8.3.1.2. JSON Writer

i)   Click on '**TreeNode**' provided next to the '**Data Writer**' option.
ii)   Select '**File Writer**' option.
iii)   Select and drag '**JsonWriter**'component to the workspace.
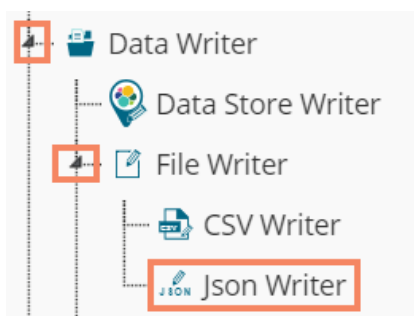
iv)   Connect the '**JsonWriter**' to a configured data source.
v)   Click on '**JsonWriter**' component to access component properties.
vi)   Enter '**File Name**' in the displayed field.
vii)   Click '**APPLY**'
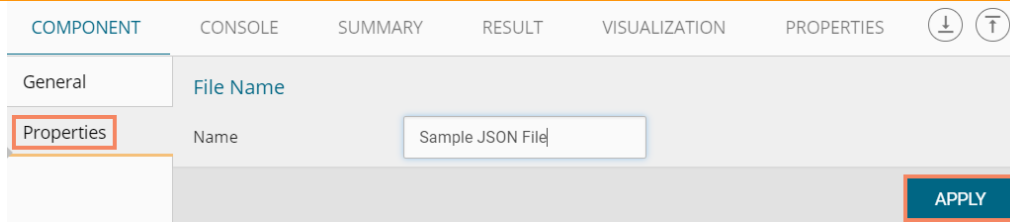
viii) Run the workflow and see the ongoing process under the '**CONSOLE**' tab
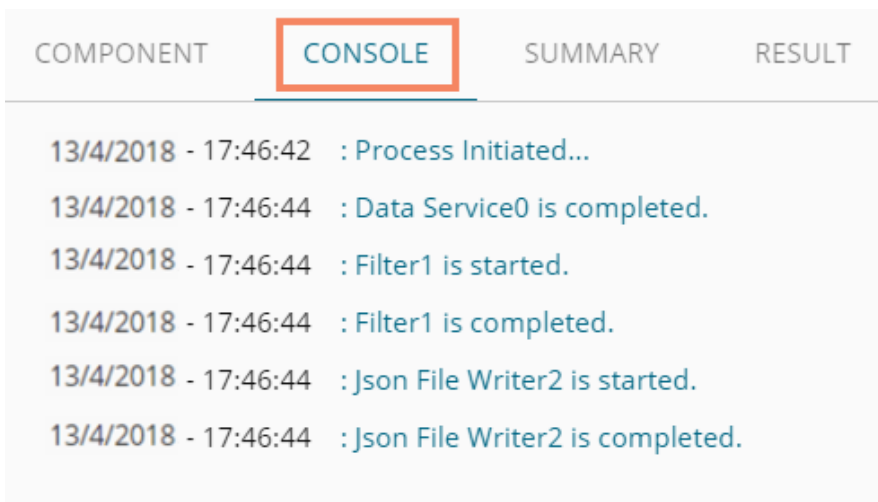
ix) After successful completion of the console process, a Pop-up message will appear with a link to download the **JSON** file.
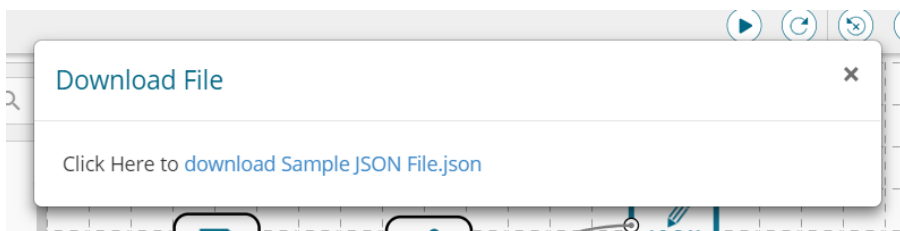
x) Click the link to download the JSON file.

8.3.2. Database Writer

8.3.2.1. Internal Data Writer

This data writer will store the data in databases like MySQL, MSSQL, and Oracle.

i) Click '**TreeNode**' provided next to the '**Data Writer**' option
ii) Select '**Database Writer**' option
iii) Select and drag '**Internal Data Writer**' component to the workspace

iv) Drag and Connect the '**Internal Data Writer**' component to a configured data source onto the workspace.

v) Click '**Internal Data Writer**' component to access the Component properties

Users will have different '**Properties**' fields based on the selected table operation as described below:

a. Selecting the 'Create a New Table' as Table Operation:

i. **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.

ii. **Type**: This field will be preselected based on the selected data Connector

iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch

iv. **Database Name**: Select a database name from the drop-down menu

v. **Password:** Enter the database password

vi. **Table Name**: Select '**Create New Table**' option from the list

vii. **Table Operation**: Select an option from the drop-down menu
1. Append to Table
2. Overwrite Table
3. Upsert

viii. **Create New Table:** It is an optional field. It appears when the user selects '**Create New Table**' option from the '**Table Name**' drop-down menu

ix. **Auto Increment:** Select an option to enable or disable the auto increment. By enabling this option, a new column will be added to the dataset, and the same column will be selected as the primary key by default

x. **Auto Increment Label:** Enter a name for the auto increment label

xi. **Column Selected from model**: Select columns that are needed to be written into the selected database

vi) Click '**NEXT**'

vii)   Users will be redirected to the '**Schema Viewer**' option
         a. Select Primary Keys: Select primary key(s) using the drop-down menu
viii)   Click '**APPLY**'

ix)   Run the workflow after getting the success message
x)   Users will be redirected to the '**CONSOLE**' tab
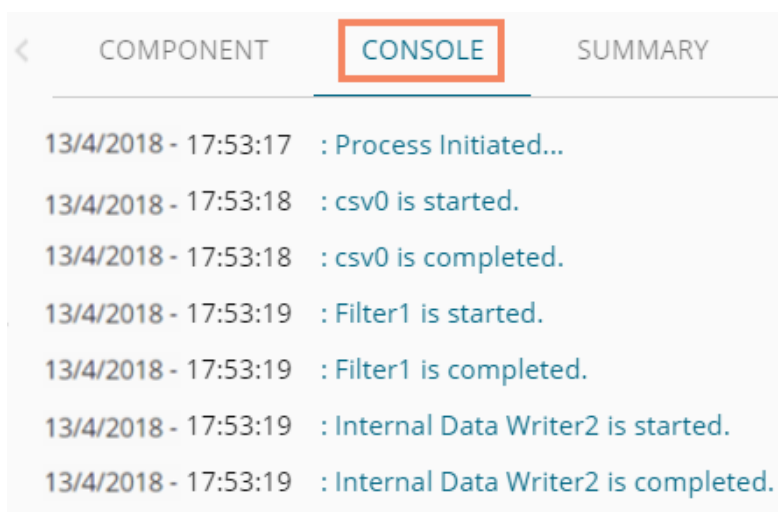
xi)   The selected data will be written to the internal data writer successfully

b. Selecting an Existing Table as Table Operation:
   i.   **Data Connector Name**: Select a data connector from the drop-down menu
   ii. **Type**: Displays a type based on the data connector chosen
   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
   iv. **Database Name**: Select a database name from the drop-down menu
   v.   **Password:** Enter the database password
   vi. **Table Name**: Select an existing table name from the drop-down menu
   vii. **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:
        1.   Append to Table
        2.   Overwrite Table
        3.   Upsert Table
   viii. **Column Selected from model**: Select columns that are needed to be written into the selected database
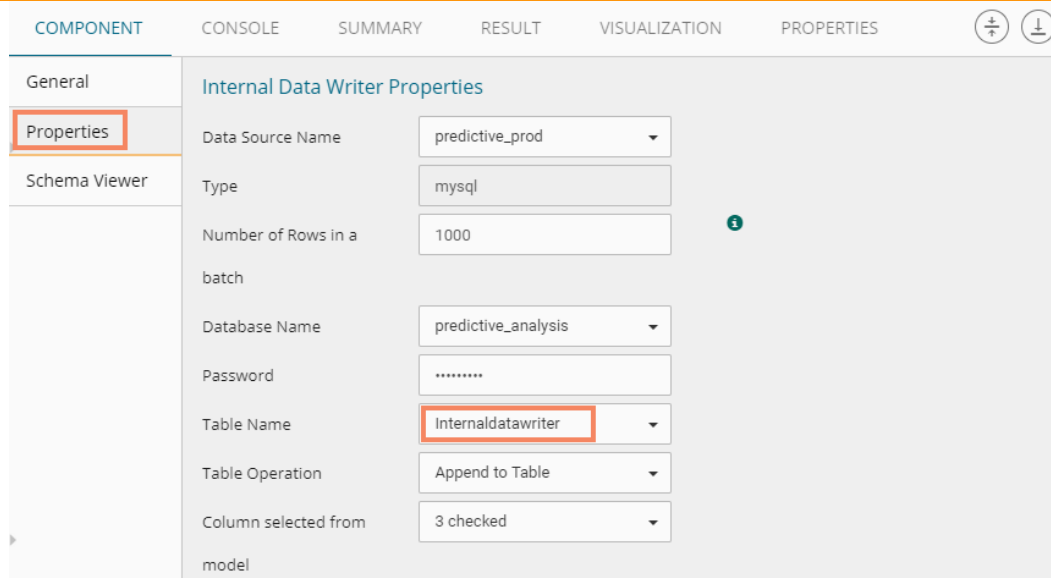
        ix. **Details of the Selected table**: Displays column headers from the selected table.

xii)   Click '**NEXT**'
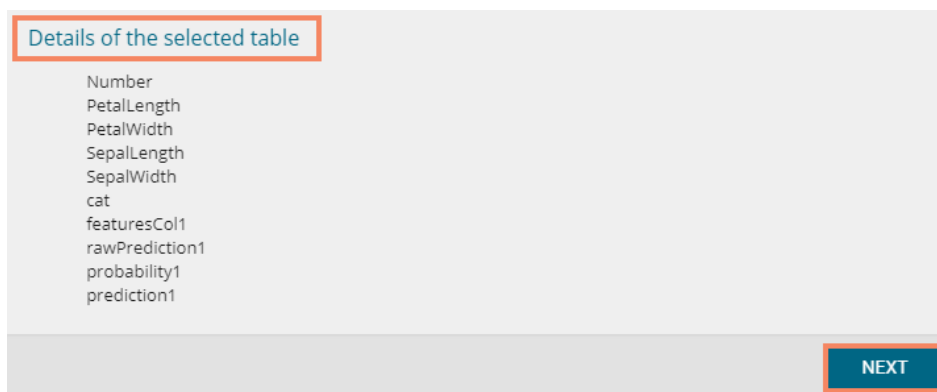
xiii)   Users will be redirected to the '**Schema Viewer**' page

xiv)   It will display the selected primary keys

xv)   Click '**APPLY**'
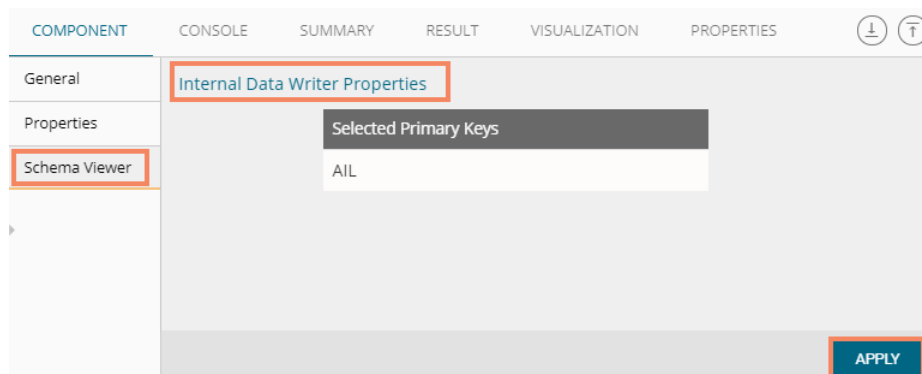
xvi)   Run the workflow after getting a success message

xvii)   Users will be directed to the '**CONSOLE**' tab displaying the ongoing process

| COMPONENT | CONSOLE | SUMMARY |

13/4/2018 - 17:53:17   : Process Initiated...

13/4/2018 - 17:53:18   : csv0 is started.

13/4/2018 - 17:53:18   : csv0 is completed.

13/4/2018 - 17:53:19   : Filter1 is started.

13/4/2018 - 17:53:19   : Filter1 is completed.

13/4/2018 - 17:53:19   : Internal Data Writer2 is started.

13/4/2018 - 17:53:19   : Internal Data Writer2 is completed.

xviii)   The data will be saved in the selected database at the end of the process

Note:

a. Users will not be able to see the '**Result**' tab for the Internal Data Writer.
b. Auto Increment Column(delta load) supports only for MySQL. Users can configure the Auto-Increment Column only while using the 'Create New Table' option as a Table Name.
c. By selecting an auto increment column by default, it will be selected as the primary key. If users want to use another column as a primary key other than the Auto-Increment Column, then it has to be configured using the '**Schema Viewer**' tab.
d. If users do not mention primary key for the 'Upsert' table operation, it will act as the '**Append**'operation

8.3.2.2. Cassandra Writer

Cassandra Writer can be used to store the predictive executions.

a. Selecting 'Create a New Table' as Table Operation

i)   Click '**TreeNode**' provided next to the '**Data Writer**' option
ii)   Select '**Database Writer**'
iii)   Select and drag '**Cassandra Writer**' component to the workspace

iv)   Connect the '**Cassandra Writer**' to a configured data source
v)   Click the '**Cassandra Writer**' component to access it
vi)   Configure the following **Properties** details:
   a. **Select Data Connector**: Select a data connector using the drop-down menu
   b. **Host Name**: Based on the chosen data connector a hostname will be displayed (Users cannot edit this field)
   c. **Port Name**: The server port number will be displayed (Users cannot edit this field)

d. **Username**: Username of the selected connection appears by default. (Users cannot edit this field)
e. **Password: the database password**
f. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
g. **Select Key Space**: Select a keyspace using the drop-down menu
h. **Replication Factor**: The replication factor mentioned in the selected '**Key Space**' will be displayed (Users cannot edit this field)
i. **Select Table**: Select 'Create a New Table table from the drop-down menu
j. **Select Columns**: Select the columns that you want to write
k. **Consistency**: Select an option from the drop-down menu
l. **New Table**: Provide a name for the newly created table
m. **New time uuid column name**: Enter a UUID column name

vii) Click '**NEXT**'

viii) Users will be redirected to the '**Key Specification**' tab.
ix) Configure the following information:
   a. **Headers**: All the columns from the data set will be listed.
   b. **Partition Key (Name):** The Partition Key determines which node stores the data. It is responsible for data distribution across the nodes.
      • The UUID Column name will be displayed under the '**Partition Key**' window.
      • Users can select and move any column from '**Header**' (Select Column) to '**Partition Key**' space.
      • The sequence of the columns listed under Partition Key can be arranged by using '**Up**' or '**Down**' options.
   c. **Clustering Key**: The Clustering Key is a storage engine process that sorts data within the partition. It determines per-partition clustering.
      • The items listed under the Clustering Key box can be arranged by using '**Up**' or '**Down**' options.

- Users can select any column from **'Headers'** (Select Column) to **'Clustering Key'** space.
x) Click **'APPLY'**

xi) **Run** the workflow after getting a success message
xii) Users will be redirected to the **'CONSOLE'** tab

Note: Users will be provided with some defined consistency level while designing the KeySpace which can be overridden based on the selected replica nodes. Users are provided with the following consistency options:

- One
- Two
- Three
- Quorum

or

b. **Selecting an Existing Table as Table Operation**

i) Connect the **'Cassandra Writer'** to a configured data source.

ii) Click the '**Cassandra Writer**' component to access it.

iii) Configure the following **Properties** details

    i.   **Select Data Connector**: Select a data connector from the drop-down menu

    ii.   **Host Name**: Enter database server details (from where the user wants to fetch data)

    iii.   **Port Name**: The server port number

    iv.   **Username**: Username of the selected connection appears by default (Users cannot edit this field)

    v.   **Password**: the database password

    vi.   **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch

    vii.   **Select Key Space**: Select a keyspace using the drop-down menu

    viii.   **Replication Factor**: Replication factor in the selected '**Key Space**' will be displayed (Users cannot edit this field)

    ix.   **Select Table**: Select a table from the drop-down menu

    x.   **Choose Columns**: Select columns from the drop-down menu that users want to be written in the data writer.

    xi.   **Consistency**: Select an option using the drop-down menu

        a. ONE

        b. TWO

        c. THREE

        d. QUORUM

    xii.   **Settings**: Select an option using the drop-down menu

        The following choices will be provided:

        a. Append Table

        b. Overwrite Table

    xiii.   The list of column headers existing in the table will be displayed once users select a table.

iv) Click '**APPLY**'

Headers	Type
uu	TIMEUUID
Number	INT
PetalLength	DOUBLE
PetalWidth	DOUBLE
SepalLength	DOUBLE
SepalWidth	DOUBLE
cat	DOUBLE

APPLY

v) After getting the success message run the workflow

vi) Users will get the process status under the '**CONSOLE**' tab

vii) The data will be saved in the selected Cassandra Writer

8.4. Scheduler

Scheduler helps to schedule the Predictive Workflow as per the requirement.

8.4.1. New Schedule

This section explains the steps to schedule a new job. Scheduling a new job is a continuous step by step process as described below:

i) Navigate to the Predictive home page
ii) Click the '**Scheduler**' tree node
iii) Two options will be displayed:
   a. New Scheduler
   b. Status
iv) Select '**New Schedule**' from the menu
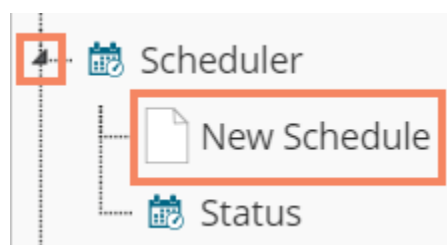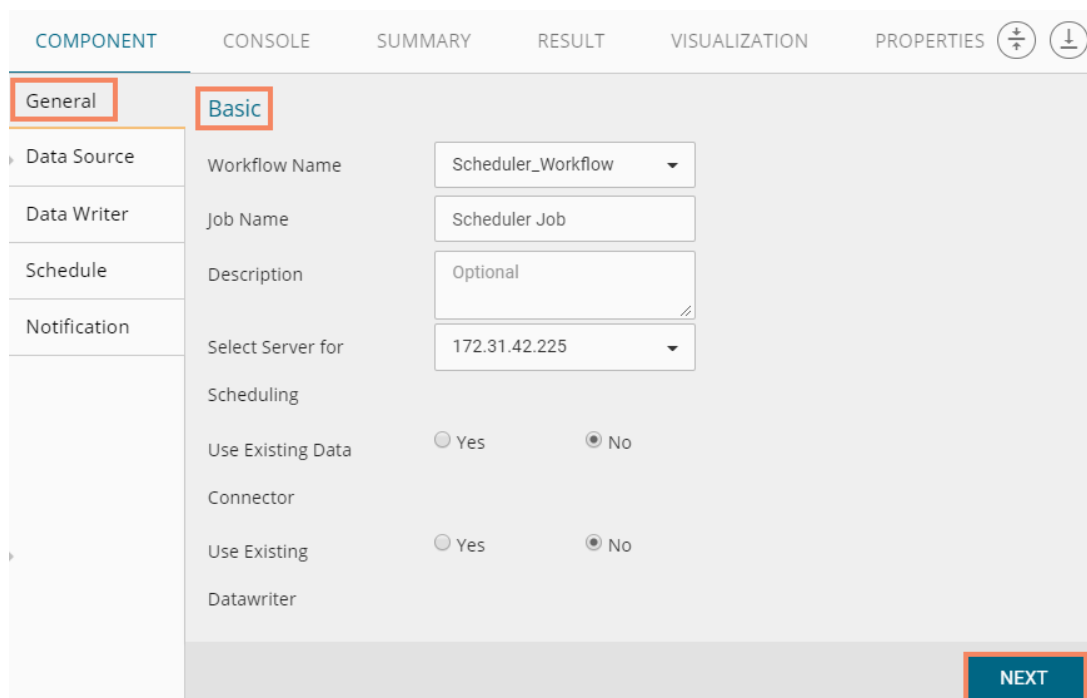
v) Users will be redirected to the '**General**' tab

8.4.1.1. Configuring General Tab

i) A '**General**' tab will open (by default).
ii) Fill in the required information:

a. **Model Name**: Select a model name using the drop-down menu
b. **Job Name**: Enter a job name
c. **Description**: Describe the job (optional field)
d. **Use Existing Data Connector**: Use radio buttons to select an option
   i. Select '**Yes**' to use an existing data connector.
   ii. Select '**No**' for not using an existing data connector.
e. **Use Existing Datawriter**: Use radio buttons to select an option.
   i. Select '**Yes**' to use an existing data writer.
   ii. Select '**No**' for not using an existing data writer.
iii) Click '**NEXT**'

iv) Users will be redirected to the '**Data Source**' tab.

8.4.1.2. Configuring Data Source

Provide the required information to configure a data source:
i) '**General**' fields will be displayed by default.
ii) Users can fill in the required fields:
   a. Component Name: A default name provided for the component
   b. Alias Name: User can enter a name for the component
   c. Description: Users can describe the component (optional)
iii) Click '**NEXT**'

iv) Users will be redirected to the '**Properties**' fields.
v) Configure the following fields (to configure a new data source):
    a. **Select Data Connector**: Select a data connector from the drop-down menu
    b. **Select Data Service**: Select a data service from the drop-down menu
    c. Based on the selected data service the below-given columns will be displayed
        i. Column Header
        ii. Data Type
vi) Click '**NEXT**'

vii) Users will be redirected to the '**Conditions**' tab. (If conditions are available, else the data source configuration will end at the previous step.)
viii) Configure the required '**Conditions**' fields.
ix) Click '**NEXT**'

x) Users will be redirected to the '**Mapping**' tab

xi) Configure the column header information from the data service that will be used for the selected model columns

xii) Click '**NEXT**'

xiii) Users will be redirected to the '**Data Writer**' tab.

Note: The '**Data Source**' tab will be enabled, only if users select '**No**' for 'Use Existing Data Connector' option while configuring the '**General**' tab for a new schedule.

8.4.1.3. Configuring a Data Writer

The Data Writer fields are reliant on the selected data writer types. The schedule is provided with two kinds of data writers: 1. Data Writer and 2. Elastic Search Writer.

1. Data Writer

i) Fill in the required details to configure a data writer
ii) Click '**NEXT**'

iii) Users will be redirected to select the 'Primary Keys'

iv) Users will be redirected to the '**Schedule**' tab.

2. **Data Store Writer**

Users can directly use the predictive workflows to create Business Stories if the workflows are written using the Elastic Search Writer.
   i) Select '**Elastic Search Writer**' as a Data Writer Type to schedule a Predictive workflow.
   ii) Users will be directed to create Hierarchy Definition.
   iii) Drag and drop the required dimensions to define hierarchical drill.
   iv) Click '**NEXT**'

v) Users will be redirected to the '**Schedule**' tab.

Note: The '**Data Writer**' tab will be enabled, only if users select '**No**' for '**Use Existing Data Writer**' while configuring the '**General**' tab for a new schedule.

8.4.1.4. Scheduling a New job

Users can select a time to schedule a new job using this section. As per the selected scheduling time, refresh interval option will be provided.

8.4.1.4.1. Job Refresh Intervals Details

- **Hourly:** By selecting this option users can schedule the job on an hourly basis.
  1. Select a specific hour by using the below-given options:

     Every_hour: Selecting this option will refresh the scheduled job after the selected hourly interval.

                          OR
     At: Selecting this option will refresh the scheduled job at the selected hour.

- **Daily**: By selecting this option users can schedule the job on a daily basis.
    1. Select a specific day by using the below-given options:
        Every_ Days: the scheduled job will be refreshed after every selected number of days. E.g., if two is selected then, the scheduled job will be refreshed every alternate day at the set time.

                                    OR

        Every Week Day: the scheduled job will be refreshed daily till the end date.
    2. Select the Start time.

- **Weekly**: By selecting this option users can schedule the job on a weekly basis.
        Select a day **or** days of the week when the scheduled job can be refreshed.

- **Monthly:** By selecting this option users can schedule the job on a monthly basis. This time **the** range can be used to set schedule refresh for more than a month.
  Select a specific day of the month by using the below given options:
  - E.g., Set monthly refresh interval (E.g., the first day of every month)
    - **OR**
    - Set a specific day after the desired monthly interval (the first Monday of the every month)

- **Yearly**: By selecting this option users can schedule the job on a yearly basis. This time range is provided for jobs running for more than one year.

  Select a specific day of the month by using the below-given options:

  Set a date for any month (E.g., The 1st January of every year until it approaches the end date)

  Or

Select a day of any month (E.g. The 1st Monday of January every year till it contacts the end date)

- **Custom Cron Expression:** Users can schedule more flexible and customizable schedule runs by using the 'Custom Cron Expression' option. The scheduled workflow can be more specific with the custom cron expression that supports timing to minutes and seconds. USers need to enter a valid Cron Expression in the given field.

Note:

a. By selecting the '**Use Existing Data Connector**' and '**Use Existing Data Writer**' options '**Schedule**' tab will be displayed immediately after the '**General**' tab.
b. Click '**NEXT**' after configuring the desired scheduling time to move on.

8.4.1.5. Notification

i) Configure the below-given fields:

    a. **Enable Email Notification**: Use a check mark in the box to enable email

    b. **Email Address**: Enable this option by check marking the box

    c. **Send Mail when Server is not running**: Users can check mark in the box to enable this option. By enabling this option, users will get an email when the server is not running.

    d. **Send Mail when Process is Completed Successfully**: Users can check mark in the box to enable this option. By enabling this option, the users get mail after the process is completed.

    e. **Send Mail when the Process is a Failure**: Users can check mark in the box to enable this option. Users will get an email when the process fails if this option is enabled.

ii) Click '**APPLY**' to save the details

iii) A success message will pop-up to assure that the job/process has been scheduled.

iv) The scheduled job/ process will be added to a list provided under the '**Status**' tab

Note:

a.   The PDF summary will be sent through email for the scheduled workflows.
b.   Multiple email addresses can be entered in coma separated value.
c.   At present, Spark Workflows are not supported by Scheduler.

8.4.2. Status

This section will display detailed information for all the scheduled jobs.

i)   Click the '**Scheduler**' tree node.
ii) Select '**Status**'

iii) Users will be redirected to the Component tab.
iv)   A list containing all the scheduled jobs will be displayed.

a.   Click '**View Logs**' to see the logs of the selected workflow under the '**Component**' tab.

Related Actions for a Scheduled Job:

Options	Name	Description

✏️	Edit	To edit/update the scheduled job details
⬛	Stop	To stop the scheduled job
✖️	Remove	To remove the scheduled job from the list
▶️	Start	To start the scheduled job

Note:

a. '**Edit**' option will allow the user to update/ edit all the tabs for the selected job.
b. Users can click the '**Start**' button to restart the scheduler for a scheduled job until it reaches the end date.
c. Users can enable '**Edit**' and '**Remove**' actions only after stopping the Scheduled job.

9. Neural Network Workspace

Users can select the NN Workspace from the Predictive landing page to access the Neural Network Environment under the Predictive Workbench.

Users will be redirected to the following screen by selecting the NN Workspace:

Note:

    a. Neural Network Space is applicable only for Python Environment
    b. Keras (as High-level API) is supported with Tensorflow Backend
    c. Tensorboard is attached for Live Visual Tracking of Model during Training
    d. Model Creation using Python Script is supported
    e. Pre-trained Model of Sentiment Analysis is Provided along with its feature scripts

The Component Tree-node menu displays various components with their sub-components to be used in the NN workspace as per requirement.

9.1. Data Source

Acquiring data from a data source is the initial step in Predictive Analysis. The '**Data Source**' tree node offers three types of data connectors:

a. CSV File
b. Data Service
c. Data Store Reader

9.1.1. Getting Data from a CSV File

    i)   Select and drag '**CSV File**' component onto the workspace.
    ii)   Click the '**CSV File**' component.

    iii)   Configure the following '**CSV Properties Configuration**' fields:
        a. **Select File**: Browse a CSV file
        b. **Delimiter:** Mention the delimiter used in the CSV file
    iv)   Click '**APPLY**'

v) Users should get the '**Apply Successful**' message as displayed in the following image:

vi) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache

vii) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

viii) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

ix) Follow the below given steps to display the result view:

a. Click the dragged data source component on the workspace.

b. Click the '**RESULT**' tab.

The current dataset contains 'text' and 'Sentiments' as displayed in the following image:

- **Rules to be followed while uploading a CSV File**
    1. The first row provided in the CSV file should contain the column headers.
    2. The second row of the CSV file should contain the data under all the headers without any 'null' or 'NA.'
    3. CSV headers should not have space. It should be a single word or two words concatenated by an underscore (_).
    4. CSV header should not contain any special characters. E.g. - %, #, $, @,*, etc.
    5. CSV header should not contain single or double quotes, dot, brackets, and high-fen.
    6. CSV header should not contain merely numbers. Numerals should be used with at least one alphabet.
    7. CSV header should not exceed 50 characters.
    8. All rows in a column should have the same data type.

    Note:
    a. The supported file types will be .csv, .tsv
    b. '**General**' tab is provided to configure the following information for any tree-node component:
        i. Component Name: The predefined name of the component is displayed in this field
        ii. Alias Name: This is user-defined name for the data source
        iii. Description (it is an optional field.)
        E.g., the following image displays '**General**' tab for a CSV data source.

9.1.2. Getting Data from a Data Service

i) Select and drag **'Data Service'** component onto the workspace.

ii) Click the **'Data Service'** component.
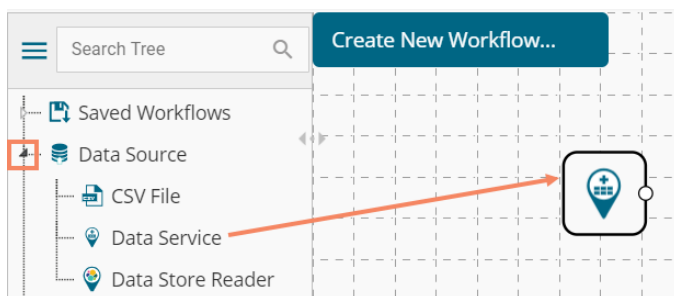
iii) Users will be redirected to the **'Properties'** fields provided under the **'Components'** tab on the Tabbed Menu Strip.

iv) Configure the **'Data Service Properties'**:

    a. **Select Data Connector**: Select a data source from the drop-down menu

    b. **Select Data Service**: Select a query service from the drop-down menu

    c. **Fields**:

       The following tables will be displayed:

        i. Column Header

        ii. Data Type

v) Click the **'APPLY'** option

vi) Click the **'APPLY'** option

vii) Click the **'Run'** ▶ icon or click **'Refresh'** ↻ icon to run the workflow by clearing the previous cache

viii) A green checkmark appears on the data source component, and users get redirected to the **'CONSOLE'** tab to display the progress of the process

ix) After the Console process gets completed, users can view the result data using the '**RESULT**' tab
x) Follow the below given steps to display the result view:
   a. Click the dragged data source component on the workspace
   b. Click the '**RESULT**' tab

- **Rules to be Followed while Creating a Data Service**
  1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
  2. Data service header should not contain any special characters. E.g. - %, #, $, @,*, etc.
  3. Data service header should not contain single or double quotes, dot, brackets, and high-fen.
  4. Data service header should not contain merely numbers. Numerals should be used with at least one alphabet.
  5. Data service header should not exceed 50 characters.

  Note:
  a. Users can develop a data service via the Data Management module of the BDB Platform.
  b. '**Fields**' option under '**Properties**' tab will appear only after selecting the appropriate query service.
  c. LOV service provided under the '**Conditions**' tab can contain only one column, in case of more than one column, a warning message will appear.
  d. Users can configure the following information for a data service data source via the '**General**' tab:

        i.   Alias Name

        ii.   Description (it is an optional field)

   e. **Users** will get the **'NEXT'** option on the Properties page only for the data service that has filters otherwise, the **'APPLY'** option will be displayed). Users have to configure the filter conditions for the data service containing filters.

9.1.3. Getting Data from a Data Store Reader

  i)   Select and drag **'Data Store Reader'** component onto the workspace

  ii) Click on the **'Data Store Reader'** component

  iii) Users will be redirected to the **'Properties'** tab of the component

  iv) Configure the required properties:

   a. Select Data Store: Select a data store using the drop-down menu

   b. Limit No. of Documents to Fetch: Select an option using the drop-down menu. Two options will be provided as shown below:

     1. Fetch all Documents

     2. Limit By

   c. Max. No. of Documents to be Fetched: Enter a number to decide maximum fetched documents (This option appears only if 'Limit By' option has been selected using the 'Limit No. of Documents to Fetch' field. Users can select any positive integer value).

  v)   Click '**NEXT**'

  vi) Users will be redirected to the **'Column Filter'** tab

  vii) Select the required columns from the drop-down list
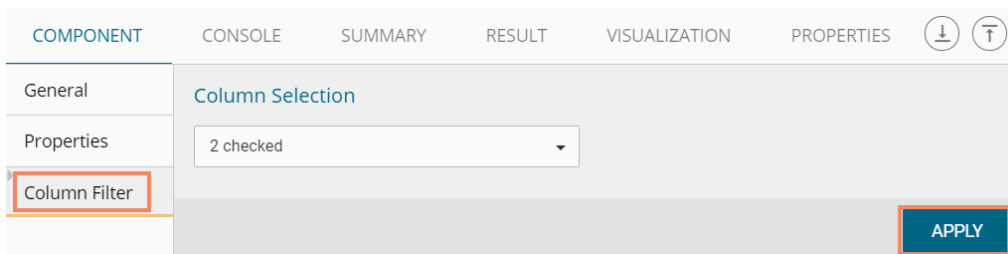
viii) Click '**APPLY**'

ix) Click the '**Run**' ▶ icon or click '**Refresh**' ↻ icon to run the workflow by clearing the previous cache

x) Users will be redirected to the '**CONSOLE**' tab to display the progress of the process

xi) After the Console process gets completed, users can view the result data using the '**RESULT**' tab

xii) Follow the below given steps to display the result view:
   a. Click the dragged data source component on the workspace
   b. Click the '**RESULT**' tab

SepalLength	SepalWidth	category
1	-0.32251082	SepalLength
-0.32251082	1	SepalWidth

Showing 1 to 2 of 2 entries   Previous   1   Next
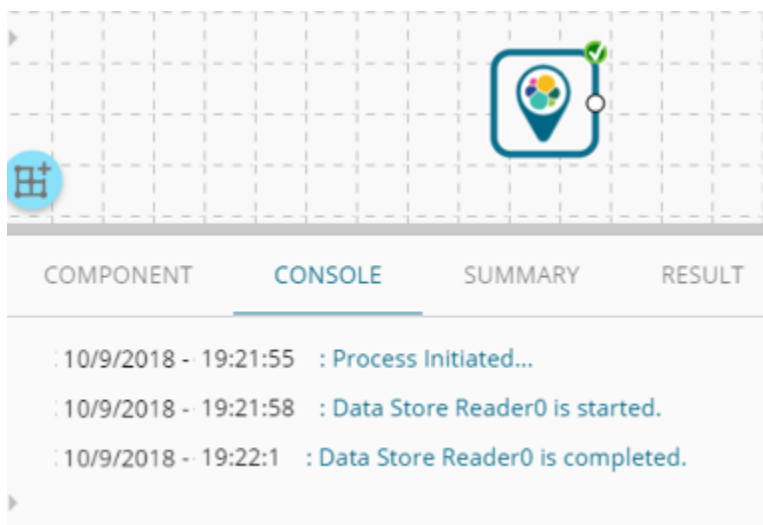
Note: Empty values present in any row of the numeric column gets replaced with zero (0) while reading data from a data store reader.

9.1.4. Removing a Data Source from the Workspace

i)   Right-click on the data source connector (in the workspace)

ii) A context menu displays '**Delete**' option

iii) Click the **'Delete'** option
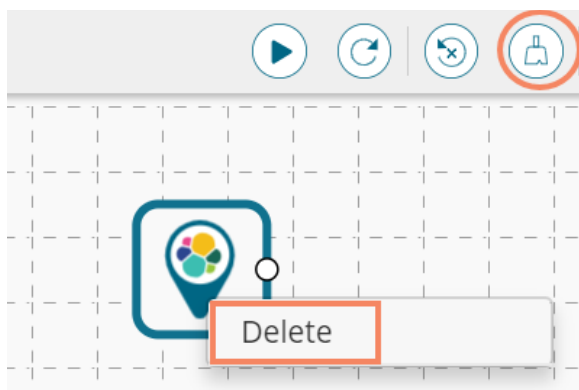
iv) The selected Data Source component will be removed from the workspace

OR

Click on the **'Reset'** icon to remove the connector(s) from the workspace

Note: The same set of steps can be followed to remove any data source type in the given tree-node menu.

9.2. Pre-Packaged Models

The component tree-node provided on the NN Workspace contains one node as Pre-Packaged Models which contains the Pre-trained Sentiment Analysis Model and it's feature scripts.
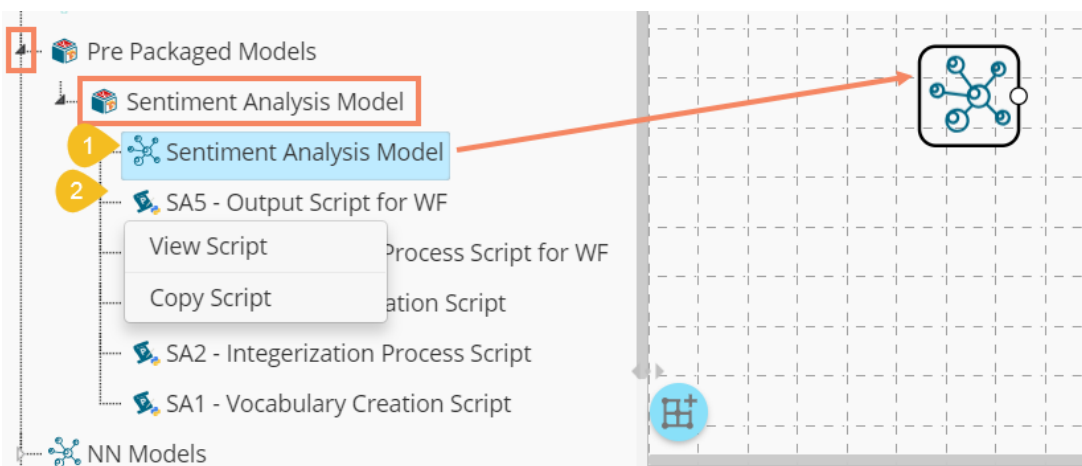
o Users can use the Pre-trained Model in a Workflow.
o These Scripts can be used directly in Workbench Area using drag-n-drop Functionality.
o The user can Copy the Script, Modify the Code and then use them as per their need.
o The user must use 'NN Apply Model' that applies the selected NN-Model over input data to get predicted results.
o Along with these Pre-trained Model and Scripts, you get support files for training this model (these can be viewed in 'Supporting Files' tabs of View Model). These supporting files user can access using **SHARED_PATH** variable in the scripts.

Note: The featured scripts are provided with Pre-trained Sentiment Analysis Model. If the users wish to modify the scripts OR refer these scripts for other user-defined models, then it must be modified as per their requirements and need to avoid error(s) & incorrect calculation.

The following image displays a workflow created by using a pre-trained model:
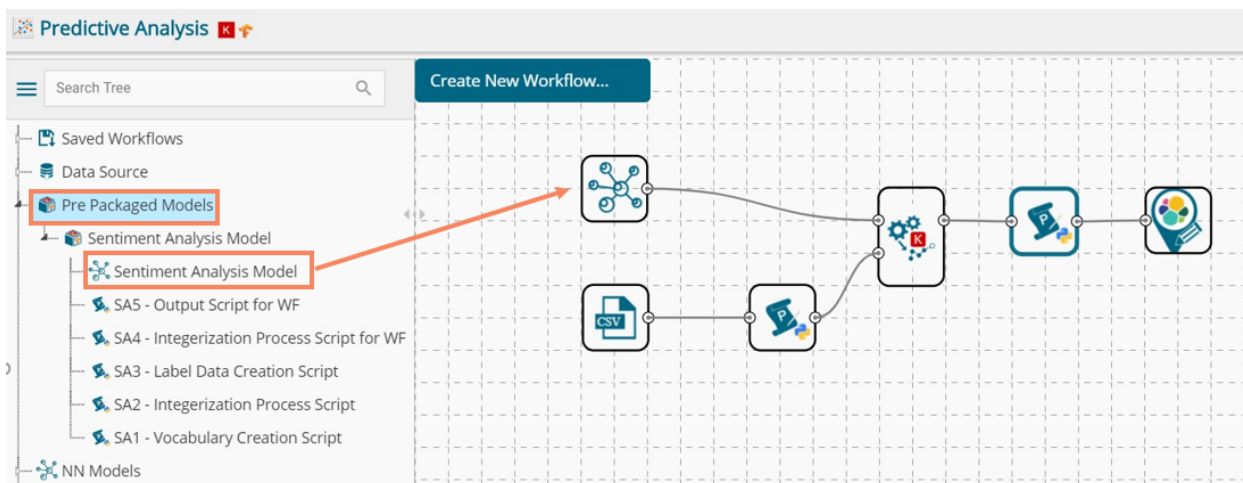
9.3. Working with Neural Network Space

This section explains the general steps for Training a Neural Network Model. The entire process can be described into the below-mentioned parts:

9.3.1. Data Preprocessing

This section describes data preprocessing from creating NumPy files to have the required data in a binary format that a Model Script can use for training or prediction purpose.

In this section the user must pre-process the data that is required for a model to get trained, we call this process 'Data Preprocessing' or NumPy-fication.

Here, the user creates NumPy files; these files have the information of data in a binary format that can be fed into the model during/after training.

Step 1 – Creating a New Model

i)   Click on the 'Create New Model' option from the NN Models tree-node
ii)   A Dialog Box opens in which the user can enter details for a New Model Name
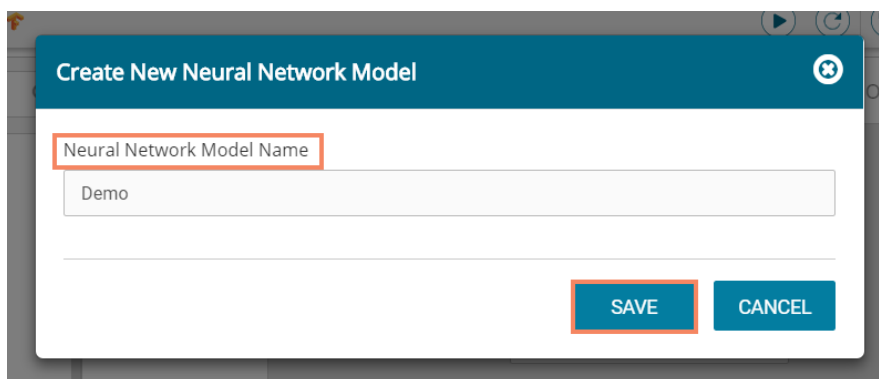
Note:

a. The user can use the maximum 20 characters to provide a name for the newly created Model
b. No other Special Character(s) except Underscore (_) is allowed
c. Model Name cannot begin with Space/Numeric Digit or Underscore
d. Model Name should be unique

iii) The newly created model lists under the '**Saved NN Models**' heading.
iv) Use a right-click on the model and select the '**View Model**' option to see the component details as shown in the following image:
    Users can view the following model properties of the selected Saved NN Model:

   a. General: The Basic Details regarding NN model is displayed in this tab
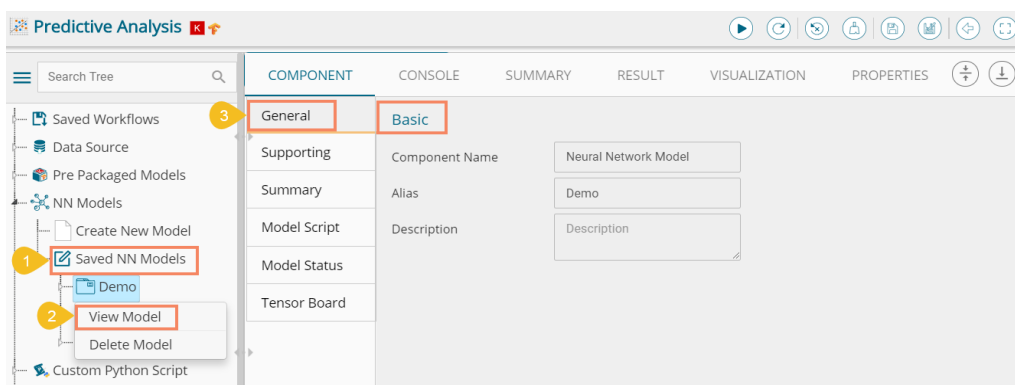
   b. Supporting Files: Details of NumPy Files used during the Model Preparation Process are displayed
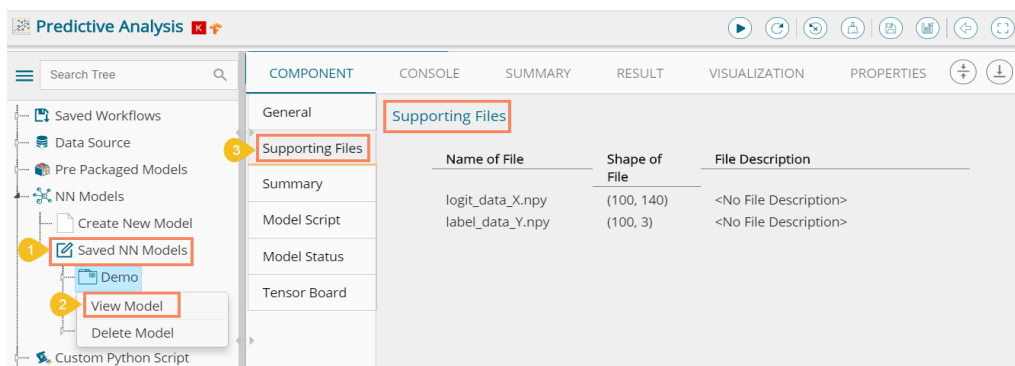
   c. Summary: The users can see the recent Keras Model Summary once the Model Training gets started
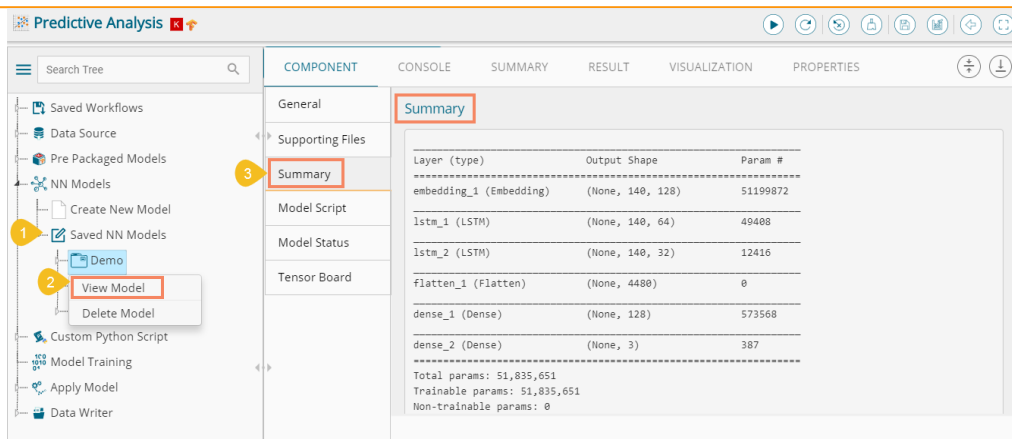
d. Model Script: The users can see the Model Structure Script
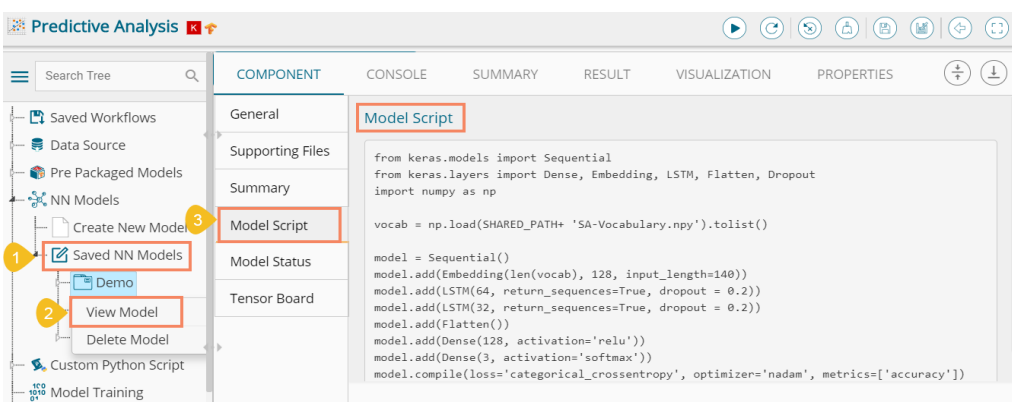
e. Model Status: The live status of a model is displayed under this section. Users can use 'Refresh Status' & 'Stop Model Training' from this section.
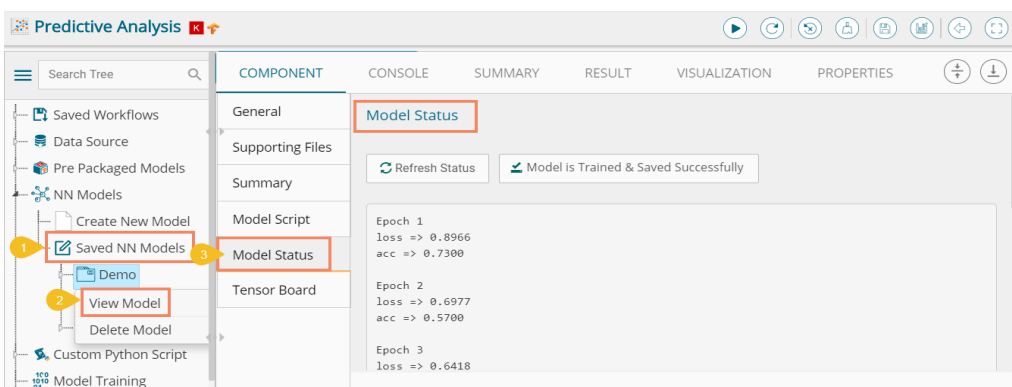
f. Tensor Board: If enabled, the live Tensorboard Visualization can be seen in this section.
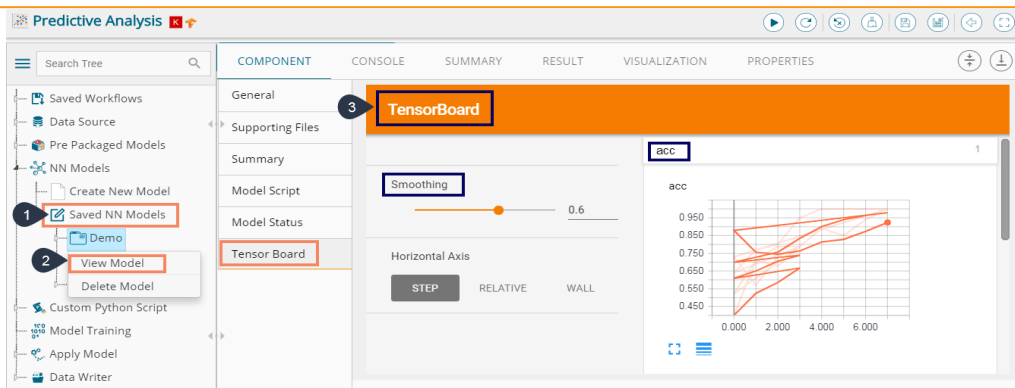
Step 2 - Creating the Preprocessing Files

Use the 'Custom Python Script' tree-node to create a new Python script inside the NN Workspace. Create a New Python Script (working with Python Script is same as it is used in the Python Workspace), the only difference in Neural Network Workspace is, Custom Python Script supports the creation of NumPy Files as well.

i)   Click the '**Custom Python Script**' tree-node
ii)   Select the 'Create New Script' node
iii) The Component tab for a new Python script opens asking for the below given configurations:
      a. General
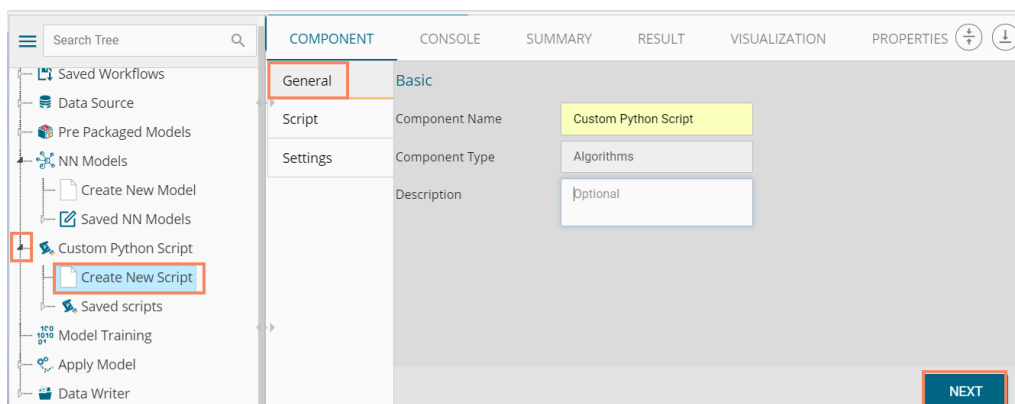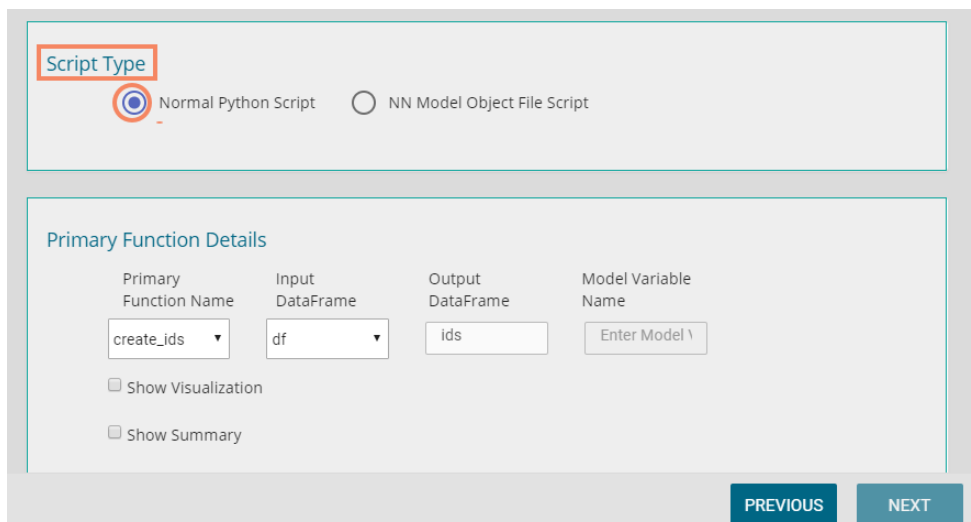
      b. Script
          i. Script Editor: Insert the script syntax inside the given space
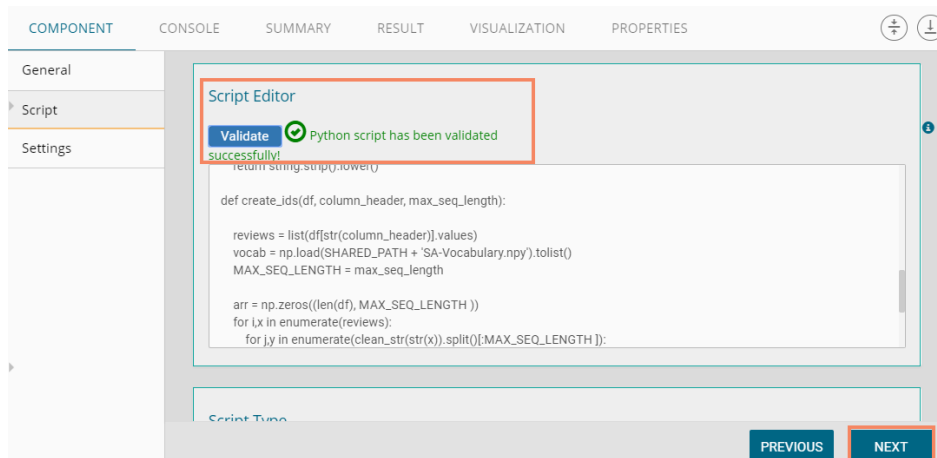
ii. Script Type: Select a script type using the checkbox
   1. **Normal Python Script**
      If the selected script type is 'Normal Python Script', then the users need to provide
         the Primary Function Details as displayed below:
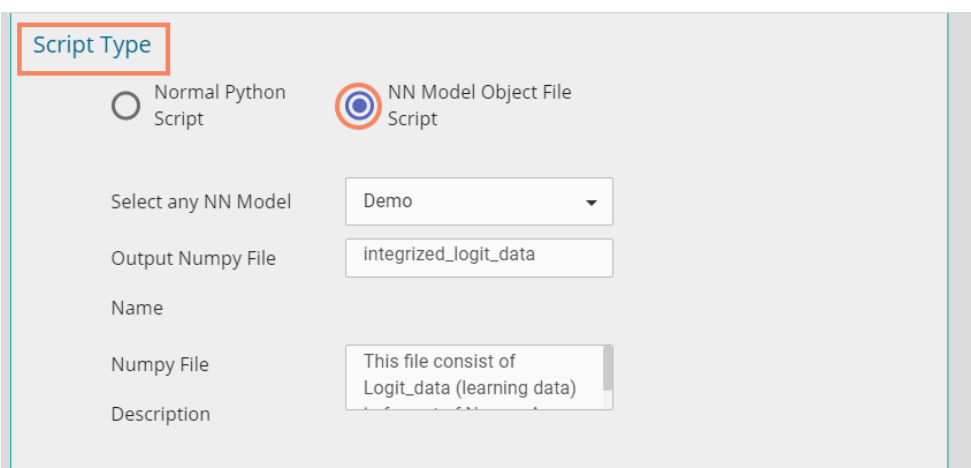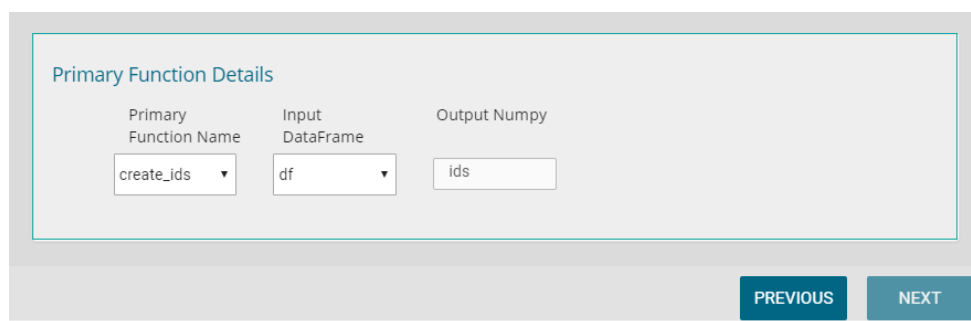
   2. **NN Model Object File**
      If it is an NN Model Object File (i.e., NumPy File Creation), then the user needs to provide the NN Model name which will be associated with an Output NumPy Filename and NumPy File Description.

iii. Click the '**Validate**' option given under the '**Script Editor**' section to validate the inserted script
iv. The '**NEXT**' option gets enabled after the successful validation of the script.

v. Click the '**NEXT**' option
c. Settings
   i. Define the Property View by using the '**Settings**' tab.
   ii. Click the '**APPLY**' option
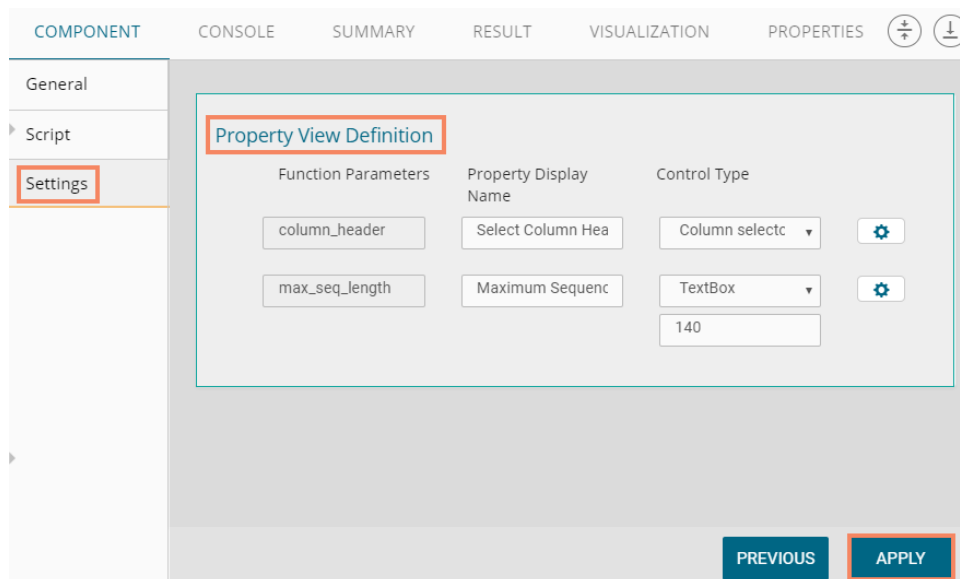
iii. A Success message appears to confirm that the Python script has been created.
iv. Users can also add multiple files and click the '**APPLY**' option to enable it for the saved model.
v. The newly created NumPy file gets stored for the future use with the selected NN Model.
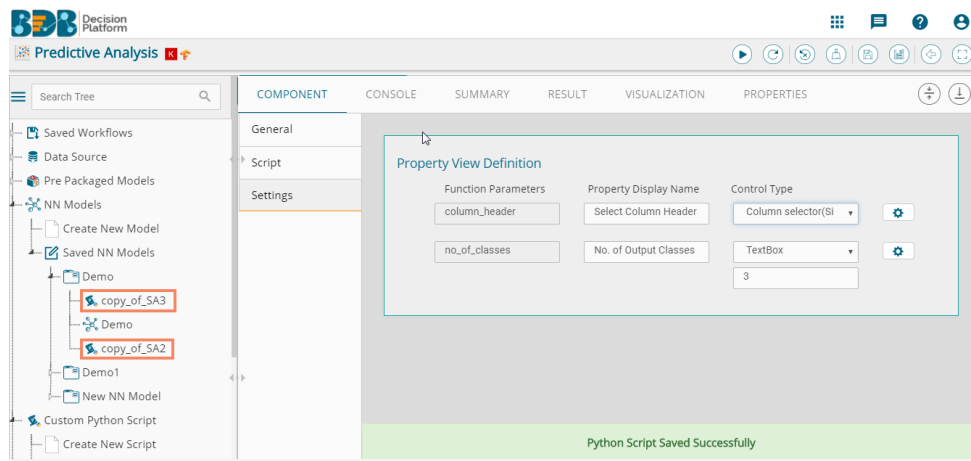
Note:

a. Output for NumPy Script must be a NumPy array. The created NumPy script can be used with any Data-Source data, and as the workflow gets completed, the NumPy file will be created and stored for the future use with the selected NN Model.
b. To access a NumPy file from the selected model use, FAKE_PATH+ '/<filename>'
c. To access the shared NumPy file from the Pre-packaged models provided use, SHARED_PATH+ '/<filename>'

   As displayed in the below given image a NumPy file is created for the '**New Model.**' This file can be used further for Model Training purpose.

As the below given image displays a NumPy file is created for the 'New Model.' This file can be used further for Model Training purpose.

9.3.2. Model Structure Creation

The user can create a Neural Network Model structure based on his/her problem statement. As of now, the user can form a structure using Script Editor provided in the 'Model Training' part using Keras API and Tensorflow as backend. We have UI support also for ease in the Model Creation.

This section describes steps to create a Keras Model Structure using the preprocessed files details. The created model can then be used for training purpose.

i)   Click the '**Model Training**' tree-node.

ii)   Configure the Model Selection fields provided under the '**General**' tab:

    a.   Select the NN Model for Training: All Created Neural Network Models list here. The user needs to select a Model for which it needs the training.

    b.   Re-train Model (if exist): Opt for this option if the selected model is already created and required to re-train the existing model

    c.   Create Model Using: Select a medium through which the model structure can be created

       i.   Script Editor

       ii.   User Interface
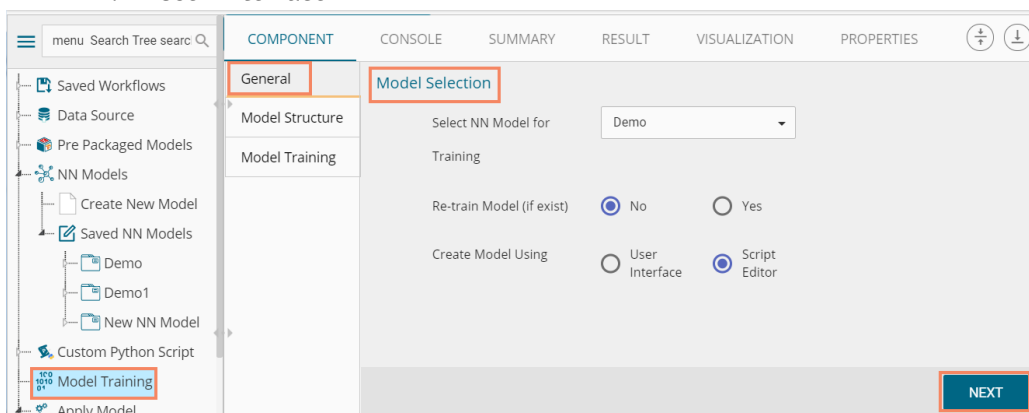
Note:

    a.   Currently, we support only the Script Editor as the Model Creation Medium.

    b.   If the selected model is already undergoing training, it will throw an error message.

iii)   After the selection of initial configuration, continue to design the Model Structure as displayed below:

    a.   Insert Python script in the space given under the Model Structure Script Editor

    b.   Click the '**Validate**' option to validate the script

    c.   Click the '**NEXT**' option to select variable files (known as Logits and Labels) for Keras Model to fit
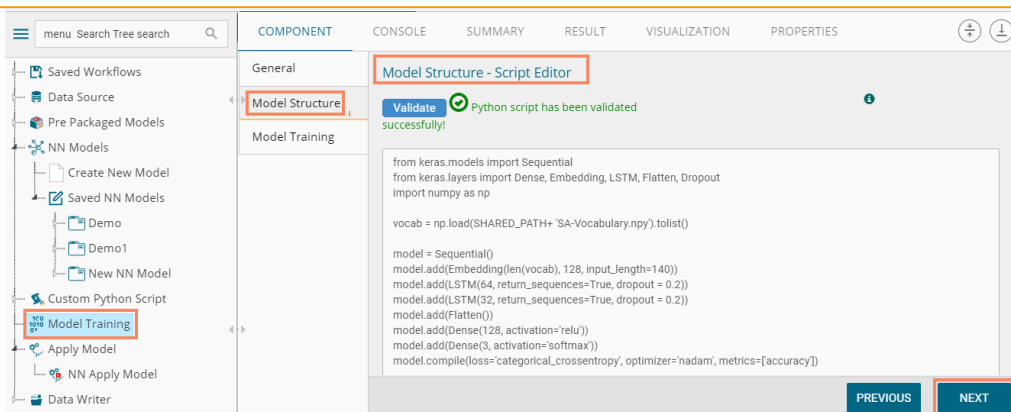
9.3.3. Model Training

This section describes steps to select and interpret the variable files.

Users can interpret Logit File as independent variables data which is preprocessed already, and Label File as target (or labeled) data. The selected model learns using the Label File data over the Logit File data and builds up weights internally which will be used for prediction using the trained model.

i) Navigate to the Model Training tab using the Model Training tree-node.
ii) Configure the required fields to Train Model:
    a. Select Logit Data File
    b. Select Label Data File
    c. Enter Batch Size
    d. Enter Epochs Value
    e. Perform Validation Split
    f. Enter Validation Split Value
    g. Shuffle
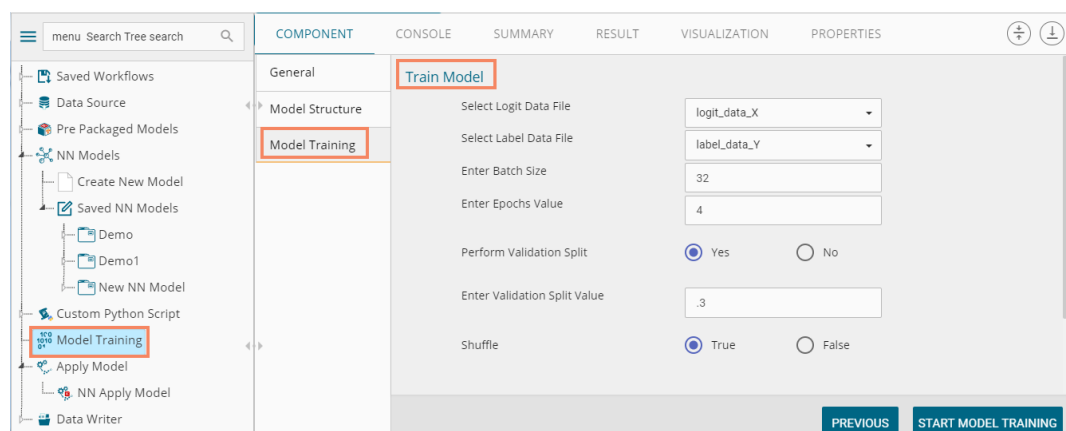
iii) Configure the following fields to send Email Notification for success or failure of the model training
    a. Enable Email Notification
    b. Email Address
    c. Send Mail when Model Training gets Completed
    d. Send Mail when Model Training gets failed
iv) Click the '**START MODEL TRAINING**' option to start the training
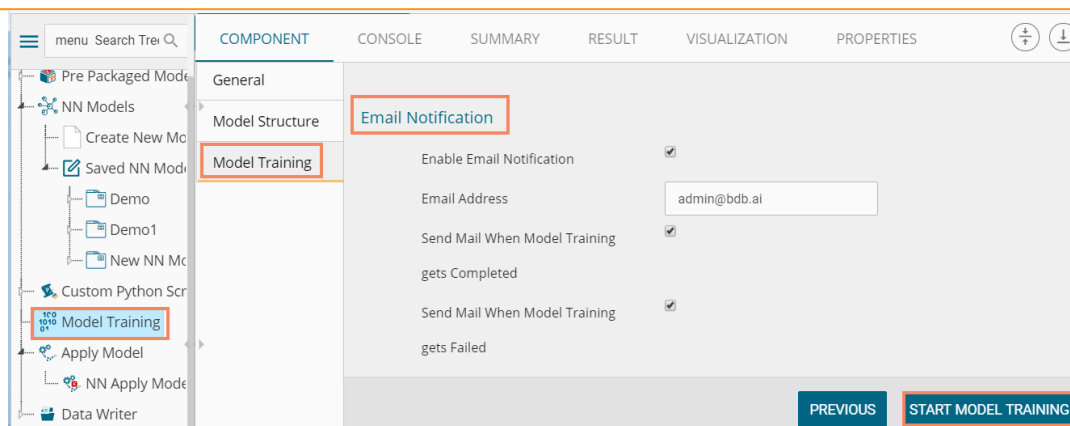
Note:

    a. The selected Logit and Label data files should not be the same.
    b. Users can provide details of Batch Size, Epochs, Validation Split as per the model requirement.
    c. After applying the model training, User can View the model status using the 'View Model' option in the context menu of the Saved NN Model.
    d. Click the '**Model Script**' tab to view the Model script using the '**View Model**' option provided for the Saved NN Models.

    e. The user must provide specific parameter values for Model Training purpose
    f. Users can track the status of the Model for each epoch including the visual tracking using Tensorboard when the model is undergoing the training process.
    g. Users can stop the model training in between during period when the model training process is going on.
    h. Users cannot process a Neural Network Model for Model Training if it is already in between the training process.
    i. Since training a model is a time-consuming task, the user can set the Model for training and provide email details to get a notification when the training gets finished or if an error occurs.

Once the model is trained successfully, users can use the model for prediction purpose.

9.4. Apply Model

9.4.1. NN Apply Model

This component is provided to generate predictions based on NN trained model. Users can view predicted column value for each label class.
Users can create an NN Apply Model via the following ways:
- Generate a model by pre-processing the selected data and training the model based on the created structure.
- Generate a new NN Apply Model using the saved NN model

The NN Apply Model consists of 2 input nodes and 1 output node.
- **Input Nodes**
  - Upper node – Model/Training data
  - Lower node – Testing data
- **Output Node**
  - Node – Result data
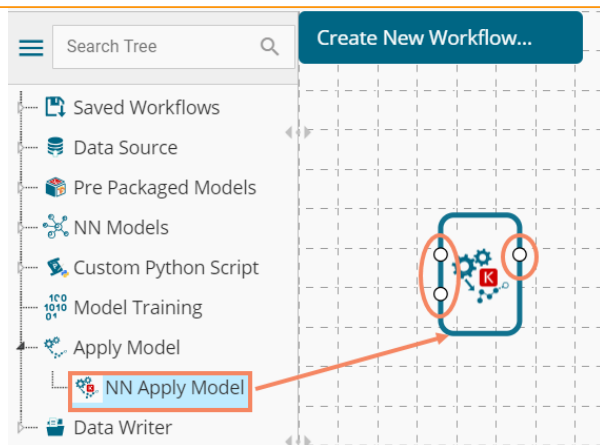
i) Click the '**Apply Model**' tree-node
ii) The '**NN Apply Model**' leaf-node will be displayed
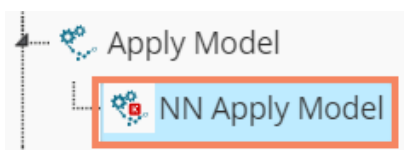
iii) Drag the NN Apply Model component onto the workspace and connect it with a valid combination of Data source
iv) Click the '**NN Apply Model**' component
v) Basic component details will be displayed
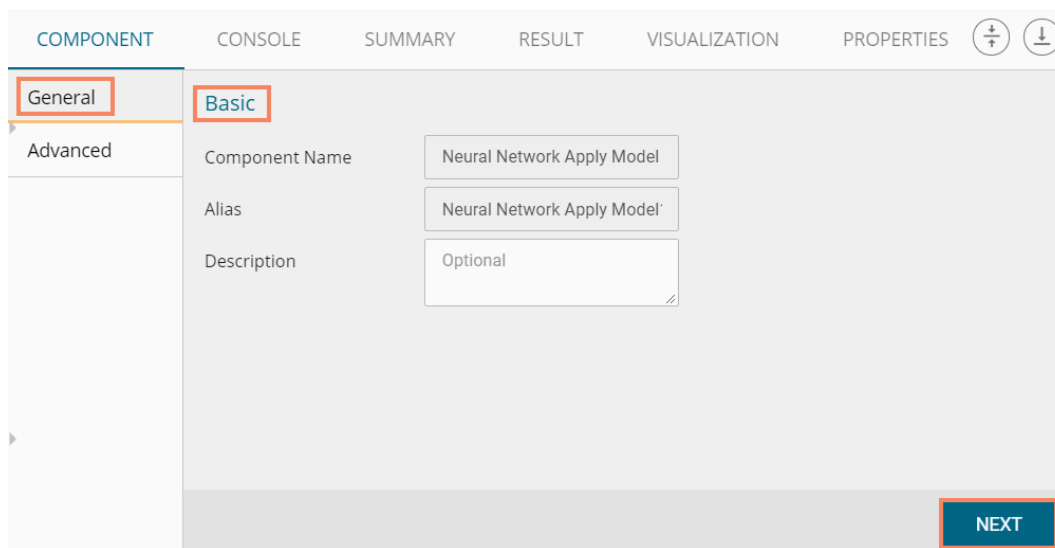
vi) Configure the Advanced tab by selecting an option from the drop-down menu
vii) Click the '**APPLY**' option
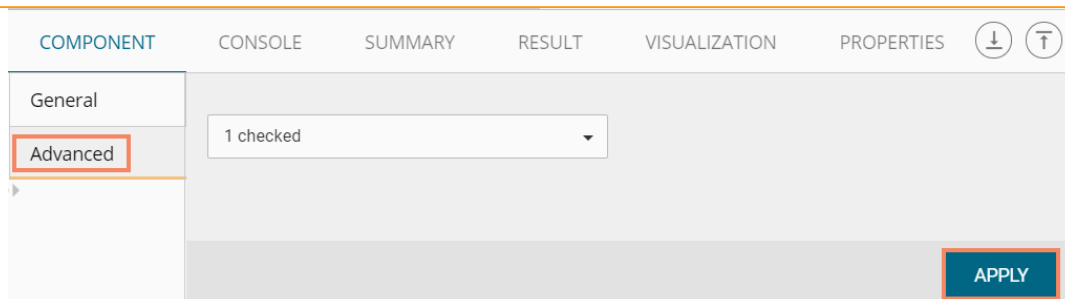
viii) After getting the success message run the workflow

ix) Users will get the process status under the '**CONSOLE**' tab
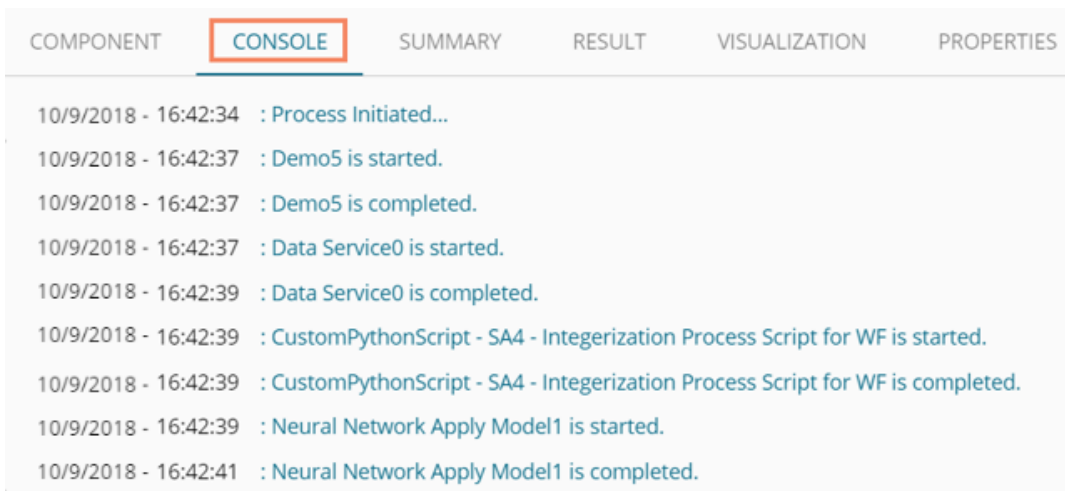
x) Follow the below given steps to display the result view:
   a. Click the dragged NN Apply Model component on the workspace
   b. Click the '**RESULT**' tab

xi) The columns displaying numpied_output probability will be added to the result view
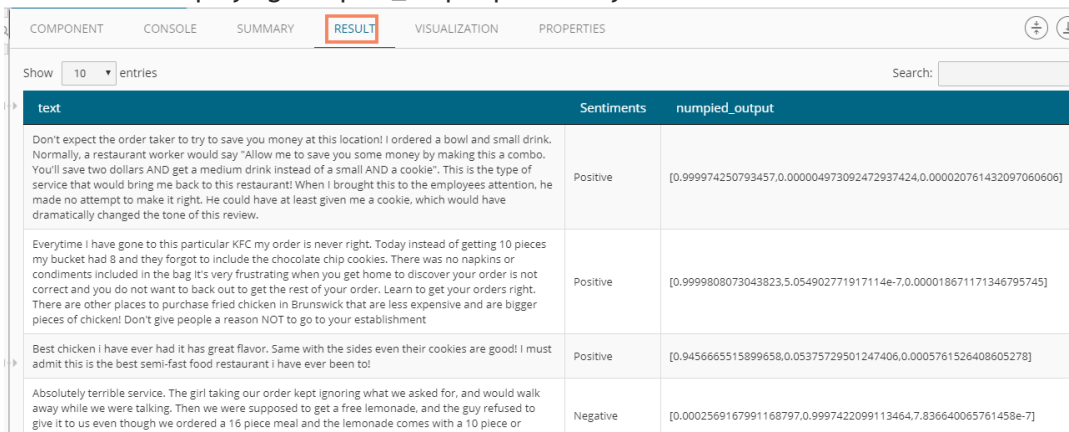
xii) Click the '**SUMMARY**' tab to view the model summary
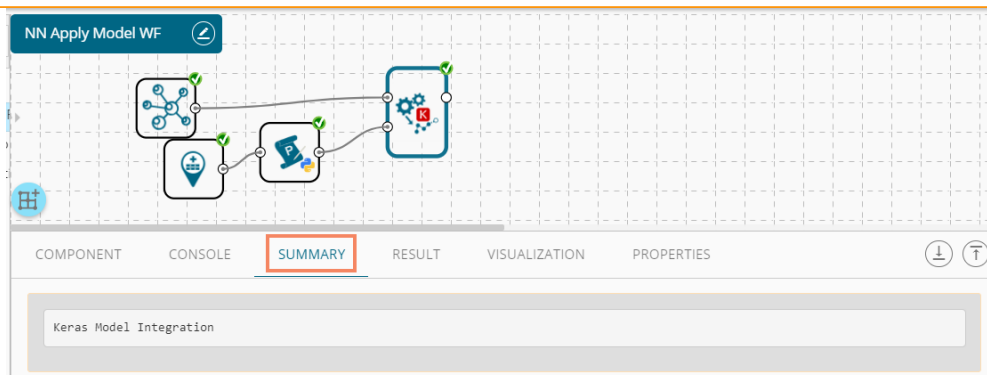
Note:
    a.   The result data set of the model can be written to a database using a Data Writer.
    b.   The Column header and data type of feature column both should match for the saved model and testing data. If column headers and data types do not match, an alert message will be displayed.
    c.   It is not mandatory for the testing data set to contain a label column.

9.5. Data Writer

Data Writers are provided to store the results of the predictive analysis in flat files or databases for further in-depth analysis.

9.5.1. Data Store Writer

Elastic Search Writer component is listed under the Data Writer Tree node. The Data Store Writer allows users to write the processed data onto the Elastic Search server which makes it more distributed.

i)   Drag the Data Store Writer component to the workspace and connect it with a configured data source or any valid combination of a data source with other given components
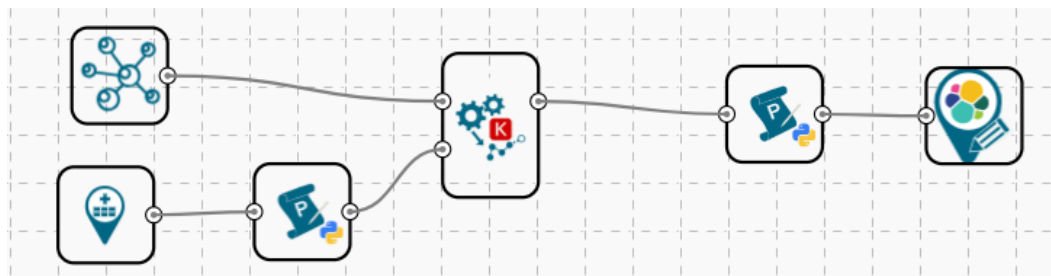
ii)   Click on the connected Data Store Writer component
iii)   Configure the required component properties
      i.   Select Data Store: Select a data store from the drop-down menu
      ii.   Select Operation Type: Select an option from the drop-down menu
      iii.   Users will get all the Dimensions, Measures, and Time fields from the selected data source
      iv.   They can define hierarchy by dragging the required Dimensions using the 'Drill Definition' box
iv)   Click '**NEXT**'
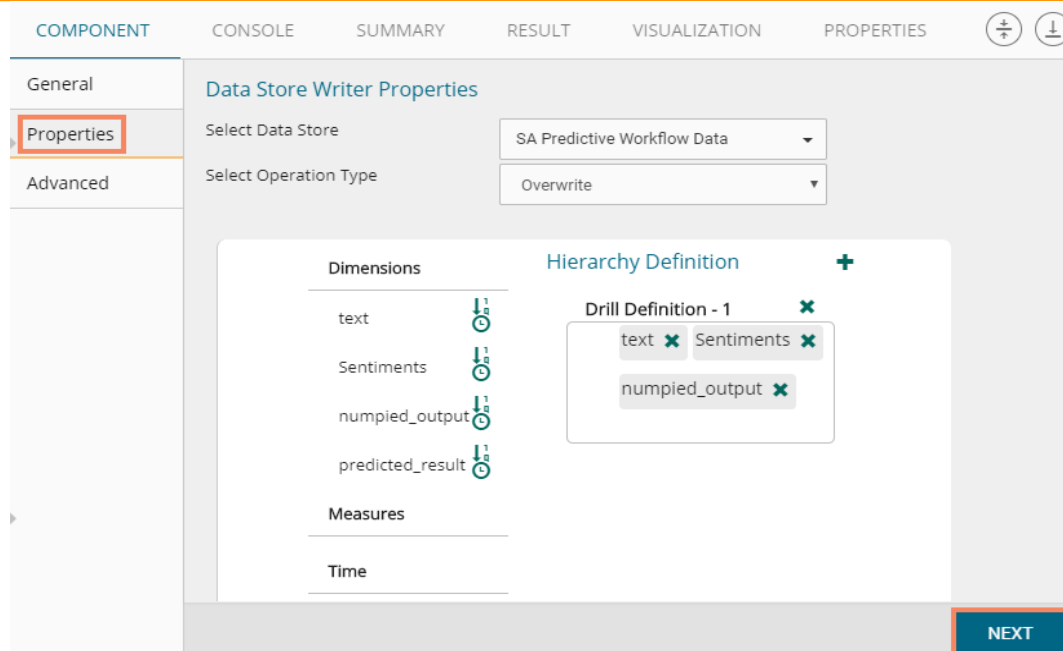
v) Users will be redirected to the Advanced fields to configure the Batch Query Properties
vi) Select a dimension for the batch query
vii) Click '**APPLY**'

viii) After getting the success message run the workflow
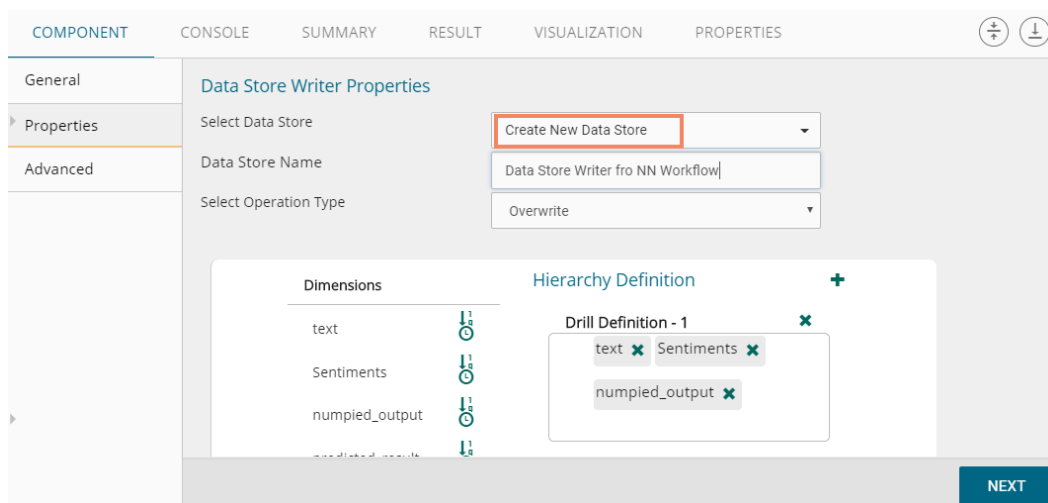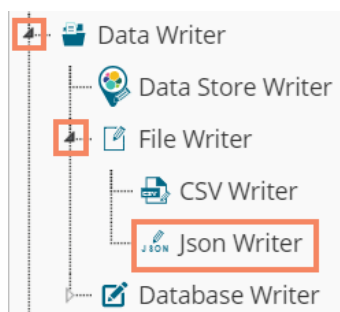ix) Users will get the process status under the '**CONSOLE**' tab

x) The data will be saved in the desired format to the selected Data Store Writer after the console process gets completed.

Note:

a. Users also get '**General**' fields for the Data Store Writer component, but they need not configure it.
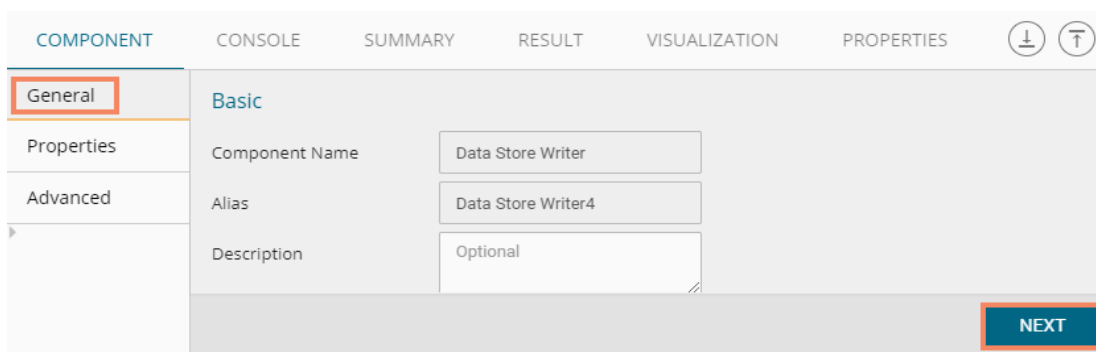
b. Users can also create a new data store using the 'Create New Data Store' option from the 'Select Data Store' drop-down menu. Users can give a name to the newly created data store by using the 'Data Store Name' field.

c. Users can move only one-dimension at a time from the list of '**Select Dimension for Batch Query'** value for the batch query.

9.5.2. File Writer

Users can write output data to flat files like CSV, TEXT, and DAT files using the File Writer.
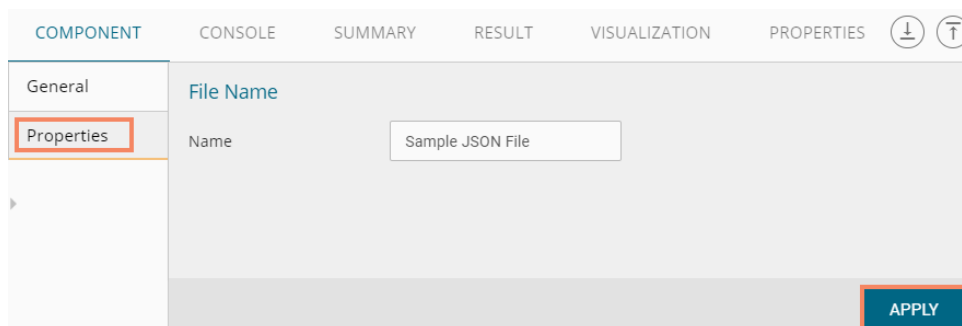
9.5.2.1. CSV Writer

i)   Click '**TreeNode'** provided next to the '**Data Writer'** option.
ii)   Select '**File Writer'** option.
iii) Select and drag '**CSV Writer'** component to the workspace.

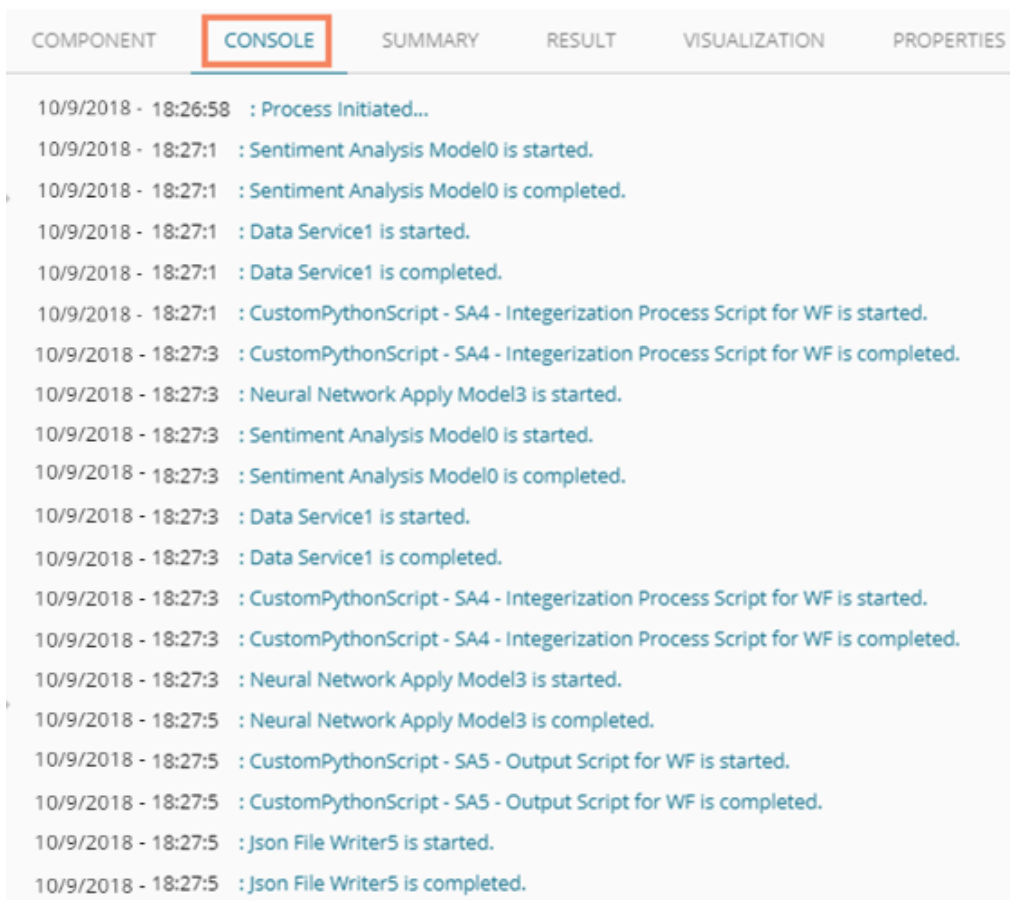iv)   Connect the '**CSV Writer'** to a configured data source or a valid workflow
v)   Click on CSV Writer component to access component properties.
vi)   Enter '**File Name'** in the displayed field**.**
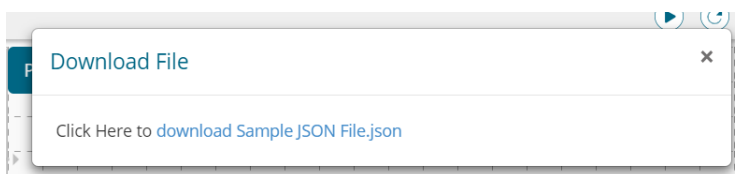vii) Click '**APPLY'**

viii) After getting the success message run the workflow
ix)   Users will get the process status under the '**CONSOLE'** tab

x) The data will be written in the CSV File
xi) Click the '**CSV Writer**' component
xii) A pop-up message will appear with a link to download the CSV file

Download File   ✕

Click Here to download Sample CSV File.csv
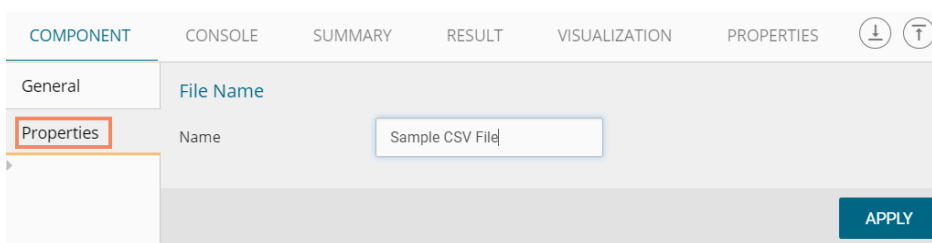
xiii) Click the link to download the CSV file.

9.5.2.2. JSON Writer

i) Click on '**TreeNode**' provided next to the '**Data Writer**' option.
ii) Select '**File Writer**' option.
iii) Select and drag '**JsonWriter**'component to the workspace.

iv) Connect the '**JsonWriter**' to a configured data source or valid workflow

v) Click on '**JsonWriter**' component to access component properties.
vi) Enter '**File Name**' in the displayed field.
vii) Click '**APPLY**'

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

General

Properties

File Name

Name   Sample JSON File

APPLY

viii) After getting the success message run the workflow

ix)   Users will get the process status under the '**CONSOLE**' tab.

COMPONENT	CONSOLE	SUMMARY	RESULT	VISUALIZATION	PROPERTIES

10/9/2018 - 18:26:58   : Process Initiated...
10/9/2018 - 18:27:1   : Sentiment Analysis Model0 is started.
10/9/2018 - 18:27:1   : Sentiment Analysis Model0 is completed.
10/9/2018 - 18:27:1   : Data Service1 is started.
10/9/2018 - 18:27:1   : Data Service1 is completed.
10/9/2018 - 18:27:1   : CustomPythonScript - SA4 - Integerization Process Script for WF is started.
10/9/2018 - 18:27:3   : CustomPythonScript - SA4 - Integerization Process Script for WF is completed.
10/9/2018 - 18:27:3   : Neural Network Apply Model3 is started.
10/9/2018 - 18:27:3   : Sentiment Analysis Model0 is started.
10/9/2018 - 18:27:3   : Sentiment Analysis Model0 is completed.
10/9/2018 - 18:27:3   : Data Service1 is started.
10/9/2018 - 18:27:3   : Data Service1 is completed.
10/9/2018 - 18:27:3   : CustomPythonScript - SA4 - Integerization Process Script for WF is started.
10/9/2018 - 18:27:3   : CustomPythonScript - SA4 - Integerization Process Script for WF is completed.
10/9/2018 - 18:27:3   : Neural Network Apply Model3 is started.
10/9/2018 - 18:27:5   : Neural Network Apply Model3 is completed.
10/9/2018 - 18:27:5   : CustomPythonScript - SA5 - Output Script for WF is started.
10/9/2018 - 18:27:5   : CustomPythonScript - SA5 - Output Script for WF is completed.
10/9/2018 - 18:27:5   : Json File Writer5 is started.
10/9/2018 - 18:27:5   : Json File Writer5 is completed.

x)   A Pop-up message will appear with a link to download the **JSON** file.

Download File   ×

Click Here to download Sample JSON File.json

xi)   Click the link to download the JSON file.

9.5.3. Database Writer

9.5.3.1. Internal Data Writer
This data writer will store the data in databases like MySQL, MSSQL, and Oracle.

i)   Click '**TreeNode**' provided next to the '**Data Writer**' option.
ii)   Select '**Database Writer**' option.
iii)   Select and drag '**Internal Data Writer**' component to the workspace.
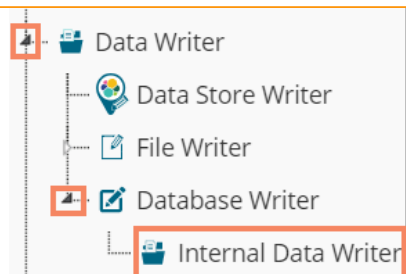
iv) Drag and Connect the '**Internal Data Writer**' component to a configured data source or workflow onto the workspace.

v) Click '**Internal Data Writer'** component to access the Component properties
Users will have different '**Properties**' fields based on the selected table operation as described below:

a. **Selecting the 'Create a New Table' as the 'Table Operation':**

   i. **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.

   ii. **Type**: This field will be preselected based on the selected data Connector.

   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch

   iv. **Database Name**: Select a database name from the drop-down menu

   v. **Password:** Enter the database password

   vi. **Table Name**: Select '**Create New Table**' option from the list

   vii. **Table Operation**: Select an option from the drop-down menu

   viii. **Create New Table:** It is an optional field. It appears when the user selects '**Create New Table**' option from the '**Table Name**' drop-down menu.

   ix. **Auto Increment:** Select an option to enable or disable the auto increment. By enabling this option, a new column will be added to the dataset, and the same column will be selected as the primary key by default.

   x. **Auto Increment Label:** Enter a name for the auto increment label

   xi. **Column Selected from model**: Select columns that are needed to be written into the selected database.

   xii. Click '**NEXT**'

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

Internal Data Writer Properties

Data Source Name	pred
Type	mysql
Number of Rows in a batch	1000
Database Name	predictive_analysis
Password	·········
Table Name	Create New Table
Table Operation	Overwrite Table
Create New Table	NNTB3
Auto Increment	Disable
Column selected from model	4 checked

NEXT

xiii. Users get directed to the '**Schema Viewer**' tab.

xiv. Click the '**APPLY**' option.

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

General
Properties
Schema Viewer

Internal Data Writer Properties

Select Primary Keys	Select

APPLY

b. Selecting an Existing Table as the 'Table Operation':

　i. **Data Source Name**: Select a data connector from the drop-down menu

　ii. **Type**: Displays a type based on the selected data connector

　iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch

　iv. **Database Name**: Select a database name from the drop-down menu

　v. **Password:** Enter the database password

　vi. **Table Name**: Select an existing table name from the drop-down menu

　vii. **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:

　　1. Append Table

　　2. Overwrite Table

　viii. **Column Selected from model**: Select columns that are needed to be written into the selected database
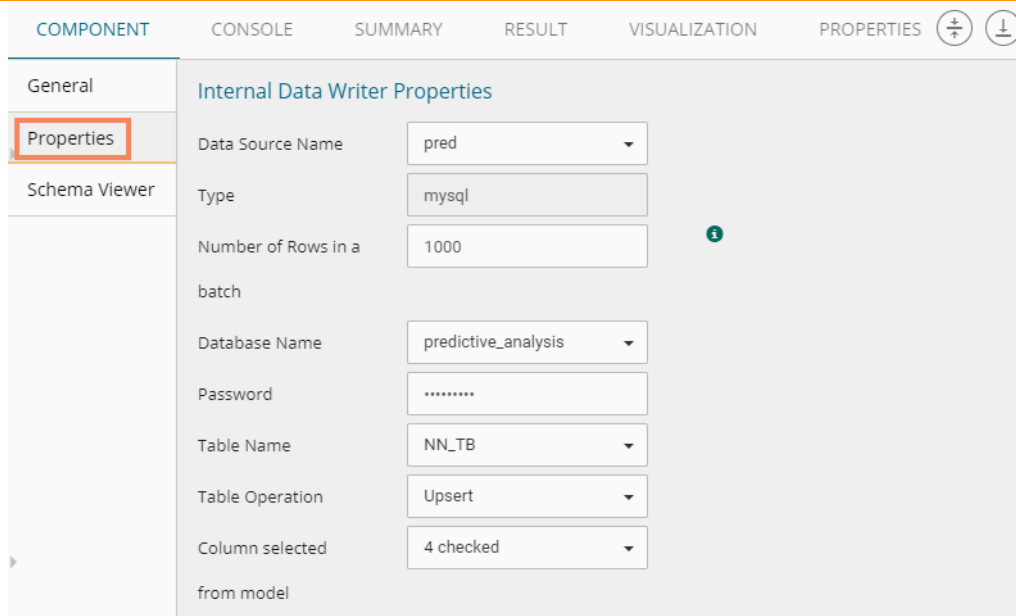
ix. **Details of the Selected table**: Displays column headers from the selected table.

vi)   Click the '**NEXT**' option
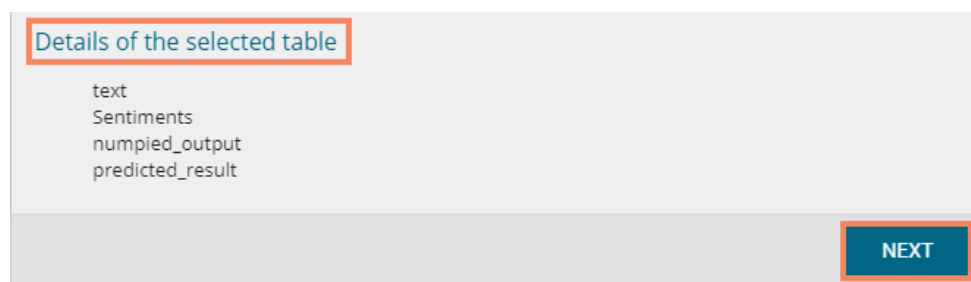
vii)   Users get directed to the Schema Viewer tab displaying the selected Primary Keys

viii)   Click the '**APPLY**' option
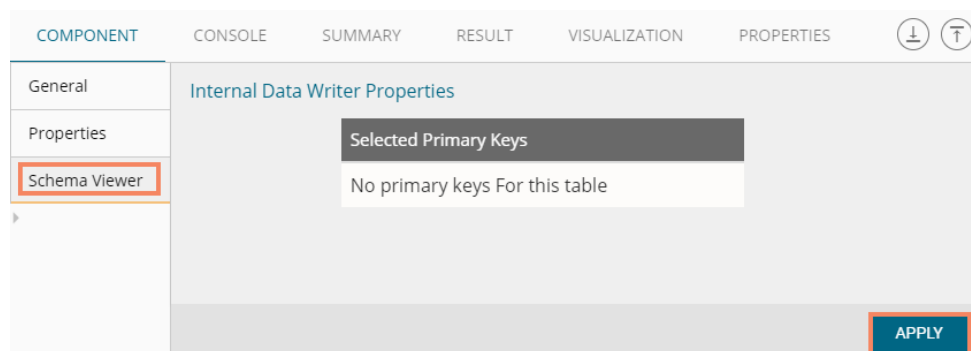
ix)   Run the Workflow after getting the success message

x)   Users will be directed to the '**Console**' tab to check the progress of the process

| COMPONENT | CONSOLE | SUMMARY | RESULT | VISUALIZATION | PROPERTIES |

10/9/2018 - 16:52:50 : Process Initiated...

10/9/2018 - 16:52:53 : Data Service0 is started.

10/9/2018 - 16:52:53 : Data Service0 is completed.

10/9/2018 - 16:52:54 : CustomPythonScript - SA4 - Integerization Process Script for WF is started.

10/9/2018 - 16:52:55 : CustomPythonScript - SA4 - Integerization Process Script for WF is completed.

10/9/2018 - 16:52:55 : NNmodddd1 is started.

10/9/2018 - 16:52:55 : NNmodddd1 is completed.

10/9/2018 - 16:52:55 : Neural Network Apply Model2 is started.

10/9/2018 - 16:52:56 : Data Service0 is started.

10/9/2018 - 16:52:56 : Data Service0 is completed.

10/9/2018 - 16:52:56 : CustomPythonScript - SA4 - Integerization Process Script for WF is started.

10/9/2018 - 16:52:56 : CustomPythonScript - SA4 - Integerization Process Script for WF is completed.

10/9/2018 - 16:52:56 : NNmodddd1 is started.

10/9/2018 - 16:52:56 : NNmodddd1 is completed.

10/9/2018 - 16:52:56 : Neural Network Apply Model2 is started.

10/9/2018 - 16:52:56 : Neural Network Apply Model2 is completed.

10/9/2018 - 16:52:56 : CustomPythonScript - n5 is started.

10/9/2018 - 16:52:57 : CustomPythonScript - n5 is completed.

10/9/2018 - 16:52:57 : Internal Data Writer5 is started.

10/9/2018 - 16:52:57 : Internal Data Writer5 is completed.

xi)   Users get directed to the 'RESULT' tab; the data will be saved in the selected database

text	Sentiments	numpied_output	predicted_result
Dont expect the order taker to try to save you money at this location! I ordered a bowl and small drink. Normally, a restaurant worker would say "Allow me to save you some money by making this a combo. Youll save two dollars AND get a medium drink instead of a small AND a cookie". This is the type of service that would bring me back to this restaurant! When I brought this to the employees attention, he made no attempt to make it right. He could have at least given me a cookie, which would have dramatically changed the tone of this review.	Positive	[1.0, 1.6056735802072453e-11, 4.813574192019665e-11]	POSITIVE
Everytime I have gone to this particular KFC my order is never right. Today instead of getting 10 pieces my bucket had 8 and they forgot to include the chocolate chip cookies. There was no napkins or condiments included in the bag Its very frustrating when you get home to discover your order is not correct and you do not want to back out to get the rest of your order. Learn to get your orders right. There are other places to purchase fried chicken in Brunswick that are less expensive and are bigger pieces of chicken! Dont give people a reason NOT to go to your establishment	Positive	[0.9999998807907104, 1.586170128575759e-08, 6.464643576009621e-08]	POSITIVE
Best chicken i have ever had it has great flavor. Same with the sides even their cookies are good! I must admit this is the best semi-fast food restaurant i have ever been to!	Positive	[1.0, 6.822673936229817e-12, 8.032920303328167e-14]	POSITIVE
Absolutely terrible service. The girl taking our order kept ignoring what we asked for, and would walk away while we were talking. Then we were supposed to get a free lemonade, and the guy refused to give it to us even though we ordered a 16 piece meal and the lemonade comes with a 10 piece or larger. Never going back.	Negative	[3.6708328821077885e-08, 1.0, 4.5597787767803766e-09]	NEGATIVE
Food is good but every single time I stop there to order pot pies, they always only seem to have 1 left....Frustrating!!!	Negative	[5.779090770374751e-07, 0.9999994039535522,	NEGATIVE

9.5.3.1.1.   Delta Load

The internal data writer can extract only new or changed records while loading data from the MySQL database. The Schema View has been added to the internal database writer to extract data using the delta data load type.
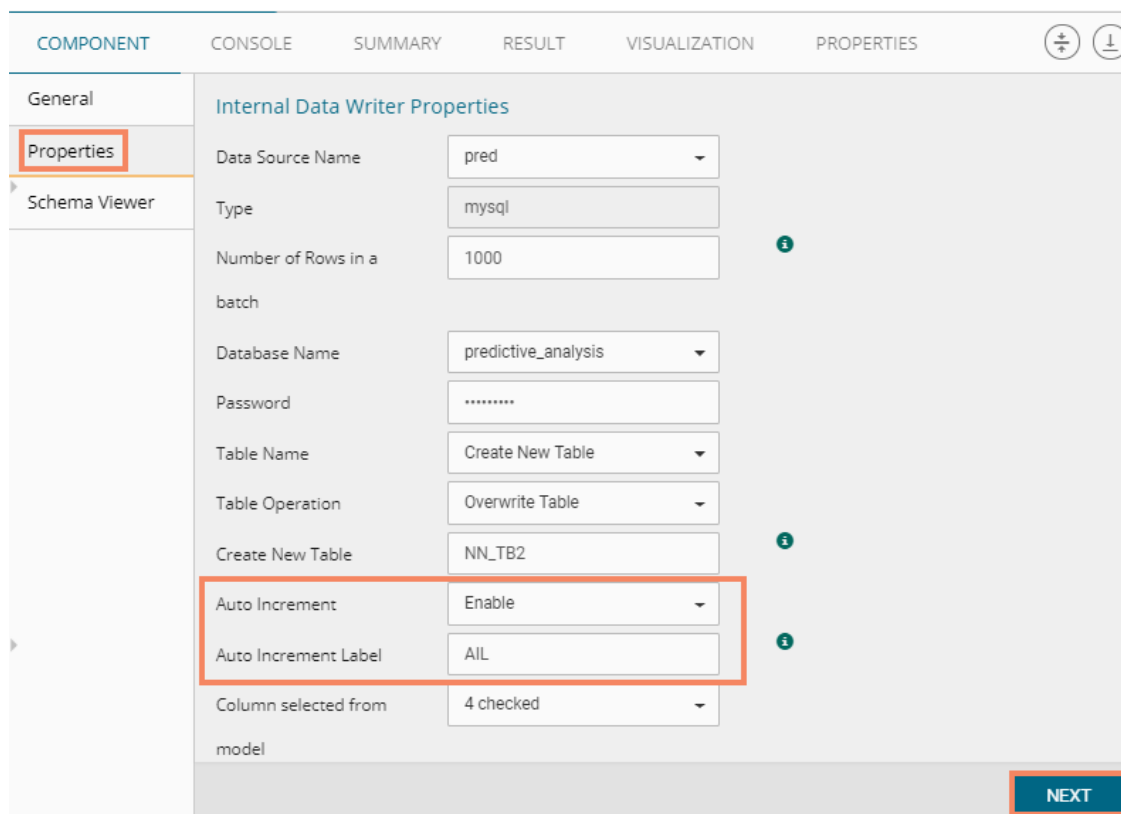
i)   Click '**TreeNode**' provided next to the '**Data Writer**' option.

ii)   Select '**Database Writer**' option.

iii)   Select and drag '**Internal Data Writer**' component to the workspace.
iv)   Connect the '**Internal Data Writer**' component to a configured data source
v)   Click the '**Internal Data Writer**' component
vi)   Users will be directed to the Properties of the Data Writer component

Users will have different properties fields based on the selected table choice as described below:

a. Selecting 'Create a New Table' as Table Operation:

i.   **Data Connector Name**: All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.
ii.   **Type**: This field will be preselected based on the selected data Connector
iii.   **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
iv.   **Database Name**: Select a database name from the drop-down menu
v.   **Password:** Enter the database password.
vi.   **Table Name**: Select '**Create New Table**' option from the list.
vii.   **Table Operation**: Select an option using the drop-down menu.
       The following choices are provided:
       1.   **Append**: Rows can be appended to the table
       2.   **Overwrite**: Delete the existing information and write the new data.
       3.   **Upsert**: Insert rows to table if they do not exist or update them if they do.
viii.   **Create New Table:** Enter the table name using this field (This field appears when the user selects '**Create New Table**' option using the '**Table Name**' field).
ix.   **Auto Increment**: User can enable or disable '**Auto Increment**' by selecting any one out of '**Enable**' or '**Disable**' options.
x.   **Auto Increment Label**: Enter a label for the autoincrement column (This field will be displayed only if, the user has enabled '**Auto Increment**' option).
xi.   **Column Selected from model**: Select columns from the model that is to be written into the selected database.
xii.   Click the '**NEXT**' option.

Note: The Schema Viewer tab will be displayed only after configuring the 'Table Name' field.

vii)   Users will be directed to the 'Schema Viewer' tab.
viii) Define Primary keys by using the 'Select Primary Keys' field. If Auto Increment is enabled, then the Auto Increment Label gets selected by default as the Primary Key.
ix)   Click the 'APPLY' option.
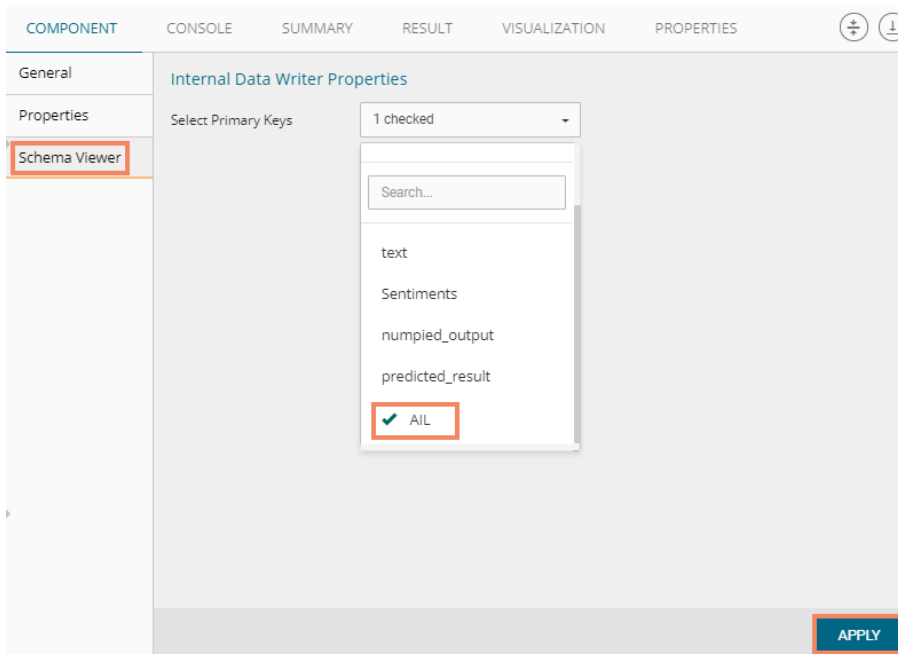
b. **Selecting an Existing Table as the 'Table Operation':**
   i.   **Data Connector Name**: Select a data connector from the drop-down menu
   ii. **Type**: Displays a type based on the selected data connector
   iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
   iv. **Database Name**: Select a database name from the drop-down menu
   v.   **Password:** Enter the database password
   vi. **Table Name**: Select an existing table name from the drop-down menu
   vii. **Table Operation**: Select an option using the drop-down menu. The following choices are provided:
        4. **Append**: Rows can be appended to the table
        5. **Overwrite**: Delete the existing information and write the new data.
        6. **Upsert**: Insert rows to table if they do not exist or update them if they do
   viii. **Column Selected from the model**: Select columns that are to be written into the selected database.

ix. **Details of the Selected table**: Displays column headers from the selected table.
x. Click the '**NEXT**' option.

x) Users will be directed to the '**Schema Viewer**' tab.
xi) The defined/selected primary keys will be displayed.
xii) Click '**APPLY**'.

xiii) Run the workflow after getting the success message run the workflow.
xiv) Users get the process status under the '**CONSOLE**' tab.

COMPONENT	**CONSOLE**	SUMMARY	RESULT	VISUALIZATION

'10/9/2018 - 12:43:54 : Process Initiated...

'10/9/2018 - 12:43:57 : Data Service1 is started.

'10/9/2018 - 12:43:58 : Data Service1 is completed.

'10/9/2018 - 12:43:58 : CustomPythonScript - IntegerizationScript is started.

'10/9/2018 - 12:43:58 : CustomPythonScript - IntegerizationScript is completed.

'10/9/2018 - 12:43:58 : Sentiment Analysis Model4 is started.

'10/9/2018 - 12:43:58 : Sentiment Analysis Model4 is completed.

'10/9/2018 - 12:43:58 : Neural Network Apply Model2 is started.

'10/9/2018 - 12:43:58 : Neural Network Apply Model2 is completed.

'10/9/2018 - 12:43:58 : CustomPythonScript - OutputScript is started.

'10/9/2018 - 12:43:59 : CustomPythonScript - OutputScript is completed.

'10/9/2018 - 12:43:59 : Internal Data Writer5 is started.

'10/9/2018 - 12:43:59 : Internal Data Writer5 is completed.

xv)   Users get directed to the 'RESULT' tab.

text	Sentiments	numpied_output	predicted_result	AIl
Dont expect the order taker to try to save you money at this location! I ordered a bowl and small drink. Normally, a restaurant worker would say "Allow me to save you some money by making this a combo. Youll save two dollars AND get a medium drink instead of a small AND a cookie". This is the type of service that would bring me back to this restaurant! When I brought this to the employees attention, he made no attempt to make it right. He could have at least given me a cookie, which would have dramatically changed the tone of this review.	Positive	[1.0, 1.6056735802072453e-11, 4.813574192019665e-11]	POSITIVE	1
Everytime I have gone to this particular KFC my order is never right. Today instead of getting 10 pieces my bucket had 8 and they forgot to include the chocolate chip cookies. There was no napkins or condiments included in the bag Its very frustrating when you get home to discover your order is not correct and you do not want to back out to get the rest of your order. Learn to get your orders right. There are other places to purchase fried chicken in Brunswick that are less expensive and are bigger pieces of chicken! Dont give people a reason NOT to go to your establishment	Positive	[0.9999998807907104, 1.586170128575759e-08, 6.464643576009621e-08]	POSITIVE	2
Best chicken i have ever had it has great flavor. Same with the sides even their cookies are good! I must admit this is the best semi-fast food restaurant i have ever been to!	Positive	[1.0, 6.822673936229817e-12, 8.032920303328167e-14]	POSITIVE	3

Note: The Result data appears based on the input data source. Users can even use the Data Preparation components and algorithms in a workflow before saving the data in a data writer.

9.6. Prediction using Trained Models

Users can use the Saved NN Model in a workflow as displayed below for the prediction purpose:

i)   Select and drag a Data Source for data reading purpose onto the workspace

ii)   Using Custom Python Script Component, create a script that can pre-process the data and will transform the input Data Source data into a consumable form by the Neural Network Model

iii) Trained Neural Network Model

iv)   NN Apply Model, it is the same as Normal Apply Model only difference is in this user needs to select the Column Headers on which the Model will predict the values

v)   After NN Apply Model, one Custom Python Script that will reverse the transform implemented by the previous script component turns the predicted values into the Predicted class Output.

vi)   The predicted output can be written to a Data Writer (in this case, it is the Data Store writer)

vii) Run the workflow

viii) The green check marks suggest that the workflow has been run successfully

ix) Users will get the process status under the '**CONSOLE**' tab

x) Click the '**RESULT**' tab to view the data result

9.7. Saved Workflows

Users can save a workflow by clicking the '**Save**' button provided on the workspace menu row. All the saved workflows will be displayed under the '**Saved Workflow**' tree node. This section explains various options assigned to a saved workflow.

i)   Navigate to the Predictive home page
ii)   Click '**Saved Workflow**' tree-node
iii)   A list of all the saved workflows will be displayed
iv)   Use the right-click on workflow from the list of '**Saved Workflows**'
v)   A context menu will open with various options (As shown below):
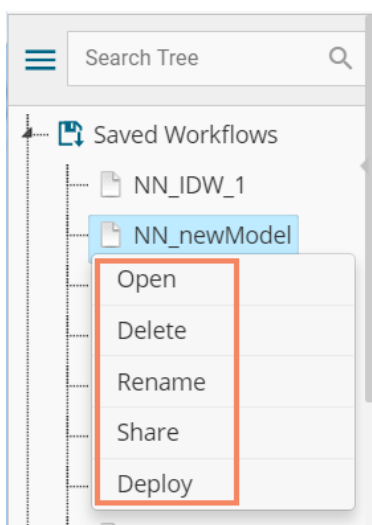
9.7.1. Opening a Workflow

i)   Right-click on a workflow from the list of '**Saved Workflows**'
ii)   Select '**Open**' from the context menu
iii)   The selected workflow will be displayed in the right pane of the screen
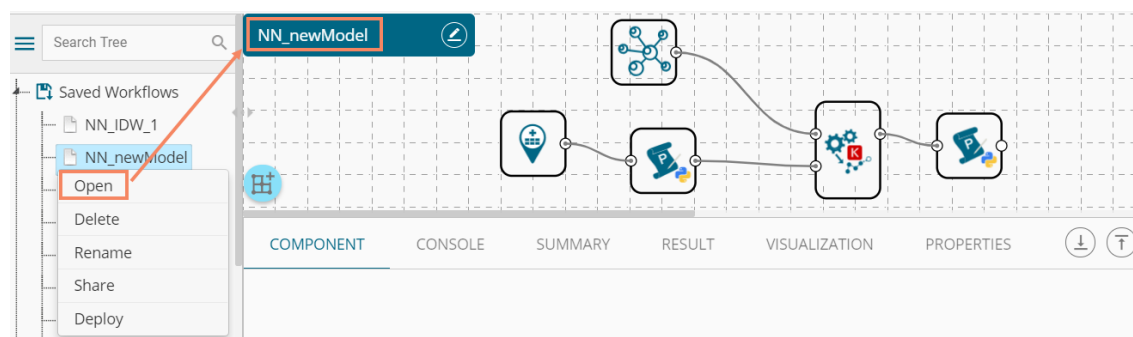
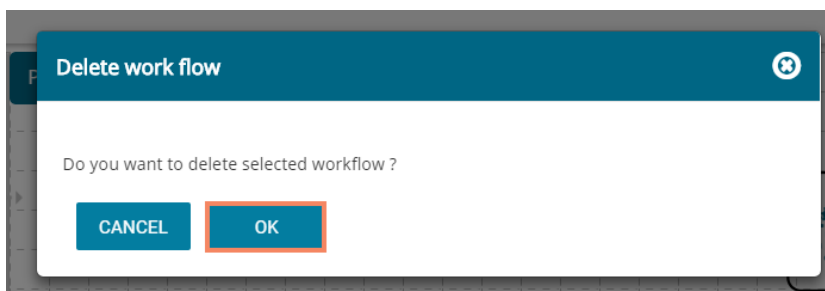**Note:** The workflow name will be displayed on the left side of the workspace menu row while opening a workflow.
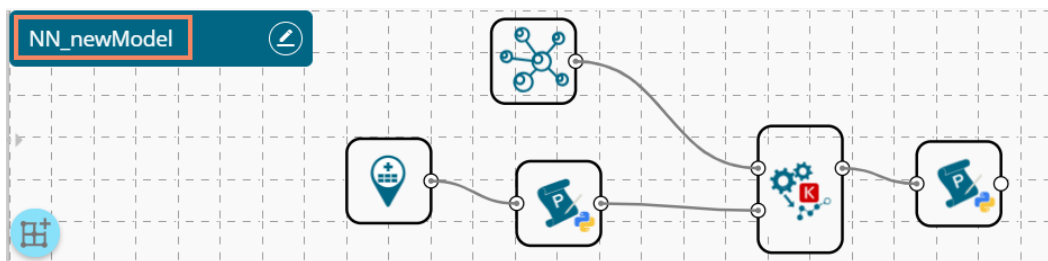
9.7.2. Deleting a Workflow

i)   Right-click on a workflow from the list of '**Saved Workflows**'
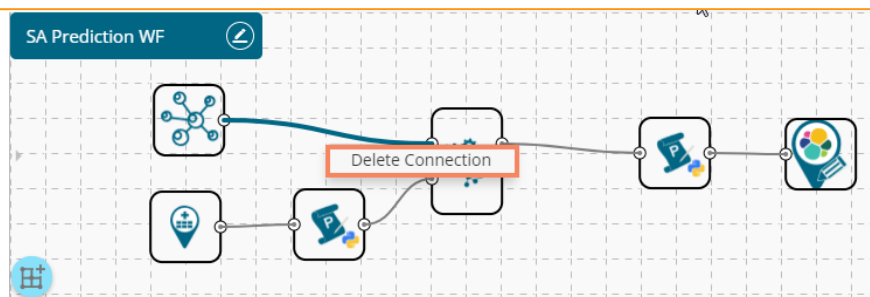
ii) Select '**Delete**' from the context menu
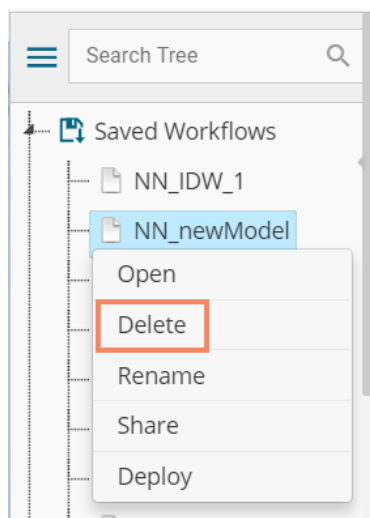
iii) A message window will pop-up to confirm the deletion

iv) Click '**OK**'
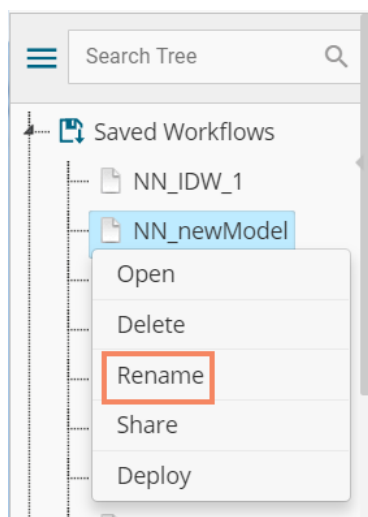

v)   The selected workflow will be removed from the list

9.7.2.1. Delete Connection in a Workflow

A Right click on the inter-node connection will display the '**Delete Connection**' option in a workflow. Click the '**Delete Connection**' option to delete a connection.
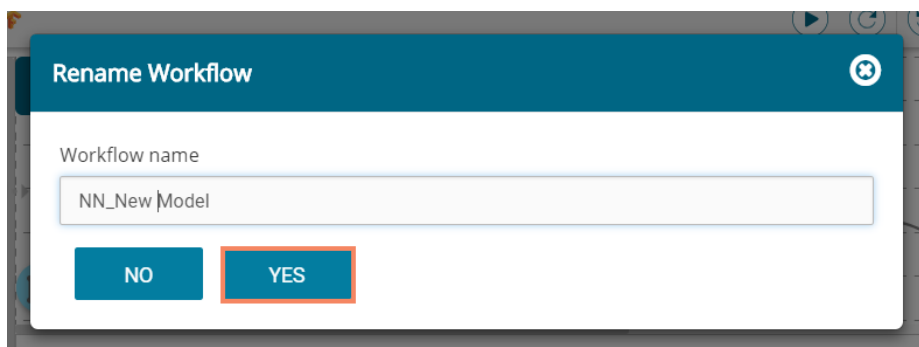
9.7.3. Renaming a Workflow
i) Press a right click on workflow from the list of 'Saved Workflows'
ii) Select the 'Rename' option from the context menu

iii) A pop-up window will appear
iv) Enter a new/modified name for the workflow
v) Click 'YES'

vi) The selected workflow will be renamed

9.7.4. Sharing a Workflow
This feature gives users the ability to share saved workflows with other users and groups.

The following options are available to share a selected workflow:

1. **Share With**: This option allows the user to share a file with the selected users or user groups. Any changes made to file will be transferred to all the users with whom the file has been shared.

   i) Press a right click on workflow from the list of '**Saved Workflows**'
   ii) Select '**Share Workflow**' from the context menu
   iii) The '**Share With**' option will be displayed (by default)
   iv) Select either '**Group**' or '**Users**'
       a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
       b. Users can be excluded by not selecting a username from the list when the '**User**' option has
          been selected.
   v) Select a specific group or user from the list by check marking the box
   vi) Click the '**APPLY**' option
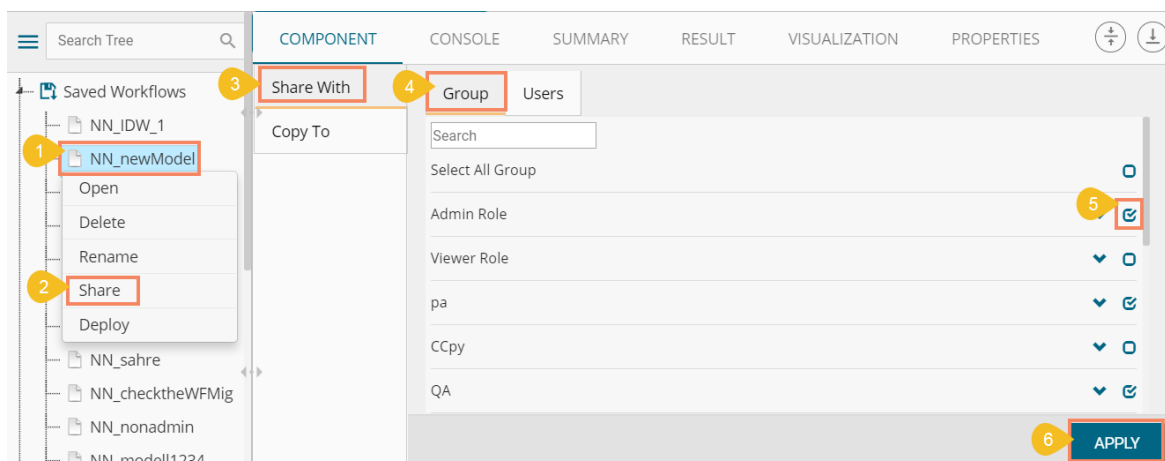
   vii) The selected workflow will be shared with the chosen user(s)/group(s)

2. **Copy To**: This option creates a copy and shares the copy with the selected users and user groups. Any changes to the original file after sharing will not show up for the users that received the shared file via the '**Copy To**' method.

   i) Press the right-click on workflow from the list of '**Saved Workflows**'
   ii) Select '**Share Workflow**' from the context menu
   iii) Select '**Copy To**'
   iv) The copied workflow name will be displayed
   v) Select either '**Group**' or '**Users**'
       a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group
       b. Users can be excluded by not selecting a username from the list when the '**User**' option has been selected
   vi) Select a specific group or user from the list by check marking the box
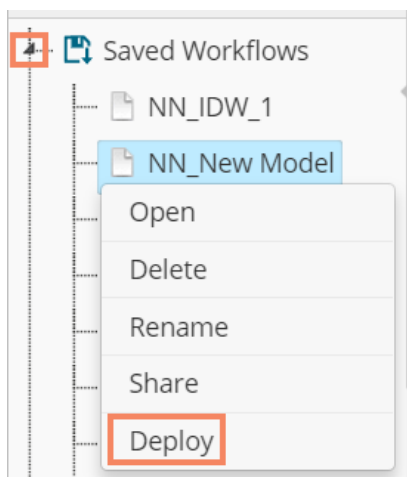   vii) Click '**APPLY**'
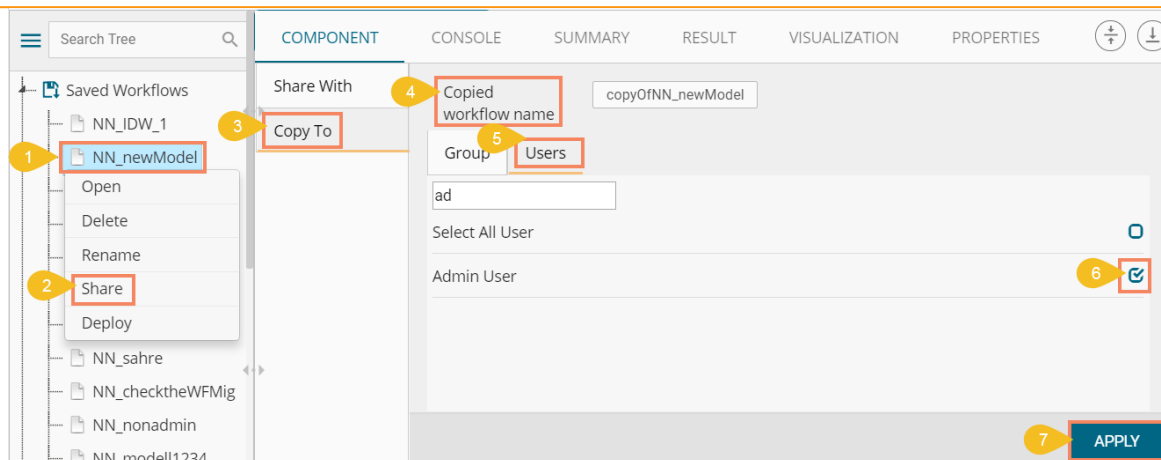
viii) The copied workflow will be shared with the chosen users/groups

9.7.5. Deploying a Workflow
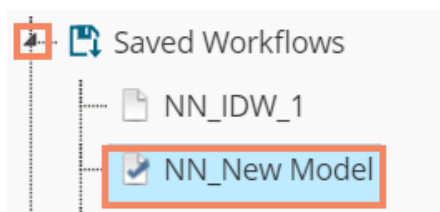The Predictive Workflows can be deployed to the BDB Dashboard Designer.

i) Press the right-click on a Workflow from the list of 'Saved Workflows'
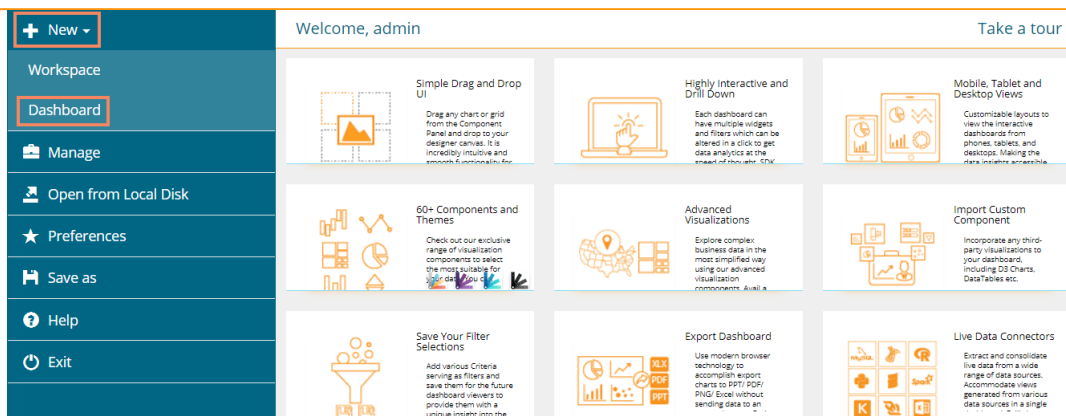ii) Select 'Deploy' from the context menu

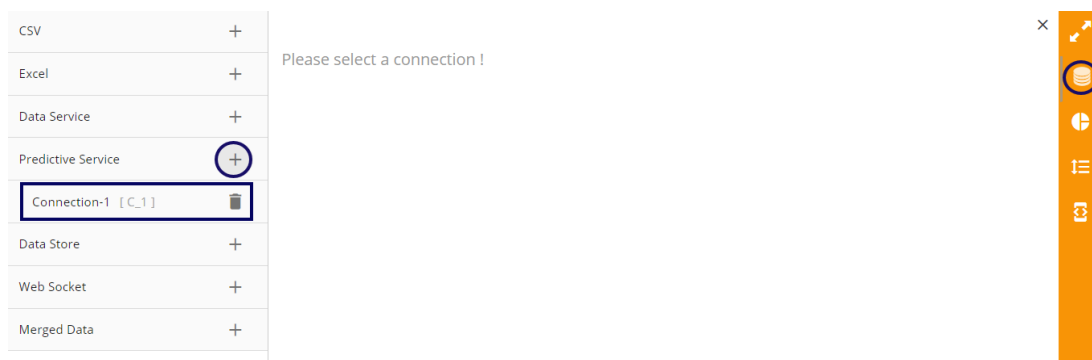iii) A success message will pop-up to assure that the workflow has been published

iv) The published workflows will be marked by a checkmark in the list of the 'Saved Workflows'
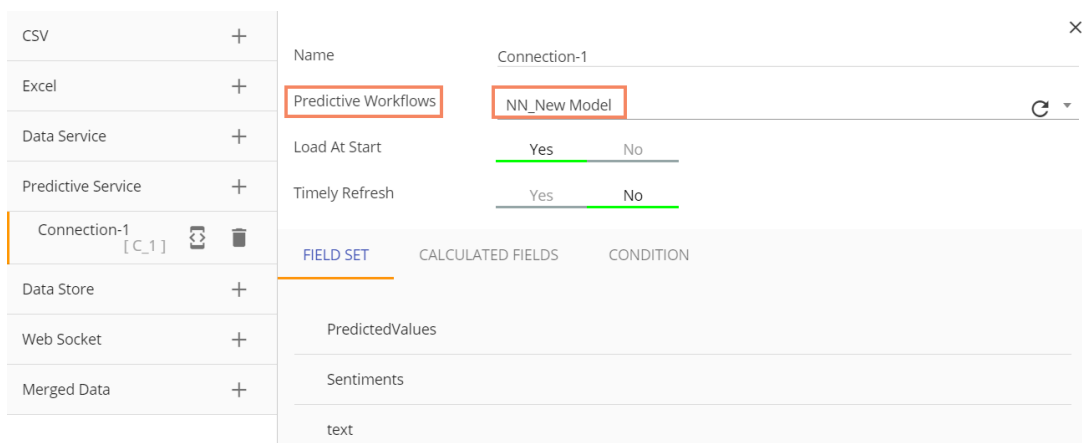

v) Navigate to the Dashboard Designer homepage
vi) Click 'New'
vii) Click 'Dashboard'

viii) Users will be directed to the Dashboard canvas

ix) Click the '**Data Source**' icon 🗄 to display all the available data sources

x) Click the '**Create New Connection**' option ➕ provided next to the '**Predictive Service**' data source

xi) A new connection will be created and added below

xii) Click on the connection to display the connection specific details

xiii) Select the deployed Predictive workflow as a data source via the drop-down menu

xiv) Configure the other subsequent details:

    a. Load At Start: Enable this option to get the updated data

    b. Timely Refresh: Enable this option to refresh data

    c. Refresh Interval: Select the time interval to refresh the data

    d. Once the data connection is established the selected predictive workflow can be used as a data source to the Dashboard Designer
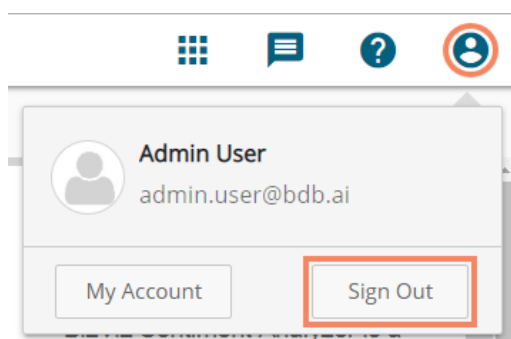
Note:

a. If a deployed Predictive Workflow has a summary, it can be viewed using the Dashboard Designer tool.
b. If the model included in the selected saved NN Workflow contains NumPy script, then after successful deployment of that workflow still users cannot create a dashboard based on it.

10. Signing Out

Users can log out from the BDB Predictive Workspace at any time they want to close it. Users can follow the below given steps to log out from the BizViz Platform.

i) Click the '**User**' icon on the Platform homepage.
ii) A menu appears with the logged in user details (User's name and email id).
iii) Click the '**Sign Out**' option.

iv) Users successfully log out from the **BizViz Platform**.

Note: Clicking on '**Sign Out**' will redirect the user back to the '**Login**' page of the BDB Platform.