



User Guide

Predictive Analysis R-3.2



Contents

- 1. About This Guide 6
 - 1.1. Document History 6
 - 1.2. Overview 6
 - 1.3. Target Audience..... 6
- 2. Introducing BizViz Predictive Analysis Tool 6
 - 2.1. Introduction to the BizViz Predictive Analysis..... 6
 - 2.2. Prerequisites..... 6
 - 2.2.1. Pre-requisites for Predictive Analysis..... 6
 - 2.2.2. R Server Requirements..... 7
 - 2.2.3. Predictive Spark Application Deployment Details 7
- 3. Getting Started with the BDB Predictive Analysis..... 22
 - 3.1. Forgot Password Option..... 23
- 4. Predictive Analysis Home Page 25
 - 4.1. Tree-node Menu 25
 - 4.2. Header Menu-Options 26
 - 4.3. Tabbed Menu Strip - Options 28
- 5. Getting Data from a Data Source 32
 - 5.1. Getting Data from a CSV File 33
 - 5.2. Getting Data from a Data Service 35
 - 5.3. Getting Data from a Cassandra Reader..... 38
 - 5.4. Removing a Data Source from the Workspace..... 40
- 6. Data Preparation 40
 - 6.1. Data Type Definition 40
 - 6.2. Filter 43
 - 6.3. Missing Value Replacement 45
 - 6.4. Formula 47
 - 6.5. Normalization..... 51
 - 6.5.1. Min-Max Normalization 51
 - 6.5.2. Zero-Score..... 53
 - 6.5.3. Decimal-Scaling 54
 - 6.6. Sample..... 55
 - 6.6.1. Sampling Methods 62
 - 6.6.2. Steps to Apply a Sampling Method 62

6.6.3.	Result View for the Available Sampling Methods.....	63
6.7.	R Split Data	66
6.8.	Spark Split Data.....	69
6.9.	Spark Filter.....	74
6.10.	Spark Data Type Definition.....	76
7.	Data Transformation	78
7.1.	String Indexer	78
7.2.	Spark R Formula.....	80
7.3.	Spark PCA	82
7.4.	Spark Chi Square.....	83
7.5.	Spark Index to String	85
7.6.	Spark SQL Transformer	87
7.7.	Spark Group By	88
8.	Algorithms.....	90
8.1.	Clustering	92
8.1.1.	R-K Means.....	92
8.1.2.	Spark-K- Means	95
8.1.3.	Spark K-Means Connected to the Pipeline Components	98
8.2.	Forecasting.....	100
8.2.1.	Triple Exponential Smoothing.....	100
8.2.2.	Single Exponential Smoothing	105
8.2.3.	Double Exponential Smoothing	109
8.2.4.	R-Auto ARIMA.....	113
8.2.5.	R- Auto Forecasting	117
8.2.6.	Result View with 'Trend' Output Mode:.....	122
8.3.	Association	135
8.3.1.	Market Basket Analysis	135
8.4.	Regression Analysis.....	139
8.4.1.	R-Linear Regression.....	139
8.4.2.	R-Multiple Linear Regression.....	142
8.4.3.	R-Logistic Regression	144
8.5.	Outliers.....	153
8.5.1.	Interquartile Range	153
8.6.	Classification.....	156
8.6.1.	R-CNR Tree.....	156

8.6.2.	R-Naive Bayes	167
8.6.3.	Spark-Naive Bayes.....	171
8.6.4.	Spark Decision Tree.....	175
8.6.5.	Spark Random Forest	183
8.7.	Correlation.....	190
8.7.1.	R- Correlation.....	190
8.8.	Recommendation Engine	192
8.8.1.	Spark ALS	192
9.	Apply Model.....	196
9.1.	Spark Apply Model.....	196
9.2.	R Apply Model	199
10.	Performance.....	203
10.1.	Spark Performance.....	203
10.1.1.	Steps to Connect a Spark Performance Component (to a Model).....	203
10.2.	R Performance	208
10.2.1.	Steps to Connect a R Performance component (to a model).....	208
11.	Data Writer(s).....	212
11.1.	File Writer.....	213
11.1.1.	CSV Writer	213
11.1.2.	JSON Writer	213
11.2.	Database Writer.....	214
11.2.1.	Internal Data Writer	214
11.2.2.	Cassandra Writer.....	219
12.	Custom R Script	224
12.1.	Creating a New R Script.....	224
12.2.	Saved R-Scripts.....	227
12.2.1.	Viewing a Saved R Script.....	227
12.2.2.	Editing a Saved R Script	227
12.2.3.	Sharing a Saved R Script	228
12.2.4.	Deleting a Saved R Script.....	229
12.2.5.	Connecting Saved R Script with a Data Source	229
13.	Custom Scala Script	231
13.1.	Creating a New Scala Script	231
13.2.	Saved Scala Scripts	235
13.2.1.	Viewing a Saved Scala Script.....	235

13.2.2.	Editing a Saved Scala Script	235
13.2.3.	Sharing a Saved Scala Script.....	235
13.2.4.	Deleting a Saved Scala Script	236
13.2.5.	Connecting Saved Scala Script with a Data Source	237
14.	Scheduler	239
14.1.	New Schedule	247
14.1.1.	Configuring General Tab.....	247
14.1.2.	Configuring Data Source	248
14.1.3.	Configuring a Data Writer.....	250
14.1.4.	Scheduling a New job	251
14.1.5.	Notification	254
14.2.	Status.....	256
15.	Live Job Status	257
16.	Saved Workflows.....	260
16.1.	Opening a Workflow	260
16.2.	Deleting a Workflow	261
16.3.	Delete Connection for a Workflow	262
16.4.	Renaming a Workflow	262
16.5.	Sharing a Workflow.....	263
16.6.	Deploying a Workflow.....	264
17.	Saved Spark Models.....	267
17.1.	Saving a Spark Model	267
17.2.	Reading a Spark Model.....	268
17.3.	Renaming a Spark Model.....	271
17.4.	Deleting a Spark Model	271
17.5.	Sharing a Spark Model	272
18.	Saved R Models	273
18.1.	Saving an R Model.....	273
18.2.	Reading an R Model	274
18.3.	Renaming an R Model	276
18.4.	Deleting an R Model.....	277
19.	Signing Out	278

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
BizViz Predictive Analysis 1.0	June 9 th , 2015	First Release of the document
BizViz Predictive Analysis 2.0	Feb 18 th , 2016	Updated document
BizViz Predictive Analysis 2.0	May 31 st , 2016	Minor Changes and Editing of the document
BizViz Predictive Analysis 2.5	November 9 th , 2016	Updated document
BizViz Predictive Analysis 2.5.1	January 3 rd , 2017	Updated document
BizViz Predictive Analysis 2.5.3	March 16 th , 2017	Updated document
BizViz Predictive Analysis 3.0	August 31 st , 2017	Updated document
BizViz Predictive Analysis 3.0	November 22 nd , 2017	Modification and Editing of the document
BizViz Predictive Analysis 3.2	January 25 th , 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 is aimed at business professionals, data analysts, data scientists, and statisticians who use BizViz Predictive Analysis tool to conduct various experimentations with data as in a Data Science Lab.

2. Introducing BizViz Predictive Analysis Tool

2.1. Introduction to the BizViz Predictive Analysis

BizViz 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 the 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 and Spark ML.

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 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 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																
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	<input type="text"/>	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	<input type="text"/>	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	<input type="text"/>	History	N/A					
application_1476353597736_0002	hdfs	smb-analytics-17	SPARK	default	Thu Oct 13 15:53:04 +0550 2016	N/A	RUNNING	UNDEFINED	<input type="text"/>	ApplicationMaster	0					
application_1476353597736_0001	hdfs	org.apache.spark.sql.hive.thriftserver.HiveThriftServer2	SPARK	default	Thu Oct 13	N/A	RUNNING	UNDEFINED	<input type="text"/>	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 Django framework. The overall server and it is 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
 - For Centos 7.0
 - \$ sudo yum - y update
 - \$ sudo yum - y install yum-utils
 - \$ sudo yum - y groupinstall development
 - For Ubuntu
 - \$ sudo apt-get upgrade
- Step 2- Installing Python 3.6
 - 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
 - For Ubuntu
 - \$ sudo apt-get update
 - \$ sudo apt-get install python3.6
 - \$ wget https://bootstrap.pypa.io/get-pip.py
 - \$ python3 get-pip.py
 - To check Python 3.6 in System,
 - \$ python 3.6 -V
- Step 3- Creating Virtual Environment
 - \$ cd <path-to-virtual-environment-directory-to-create>
 - eg: \$ cd ~/
 - \$ mkdir <VIRTUAL_ENVIRONMENT_DIRECTORY_NAME>
 - eg: \$ mkdir venv
 - \$ 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)
 - \$python3.6mvenv<PATH_TO_VIRTUAL_ENVIRONMENT_DIRECTORY> --without-pip
 - eg: \$ python3.6 -m venv /home/bizviz/venv --without-pip
 - \$ cd <VIRTUAL_ENVIRONMENT_DIRECTORY>
 - \$ source bin/activate # Activating Environment
 - \$ wget https://bootstrap.pypa.io/get-pip.py # Obtaining pip File
 - \$ python get-pip.py # Installing pip

Note: In case still you are facing problems with 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 will happen in the activated virtual environment.

To Activate Virtual Environment,

- \$ cd <VIRTUAL_ENVIRONMENT_DIRECTORY>

- \$ source bin/activate
 - eg: \$ cd /venv
 - 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,

- Django Server Port - 8000s
- 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 will be 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 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-connector	2.1.6	\$ pip install mysql-connector==2.1.6
2.	PyMsSql	2.1.3	<ul style="list-style-type: none"> • In Centos 7.0 <ul style="list-style-type: none"> ○ \$ sudo yum install freetds-devel ○ \$ pip install pymssql==2.1.3 • In Ubuntu <ul style="list-style-type: none"> ○ \$ 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 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 will 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 directory where our virtual environment has been setup
- PA_Python is directory in which we will create,
 - <CACHE_DIR> named: 'CacheData'
 - <SAVED_MODEL_DIR> named: 'SavedPythonModels'
 - <VALIDATION_DATA_DIR>: 'ValidationData'
 - <CELERY_DIR>: 'celery'
- bizviz3.2 is git-cloned directory
- python-predictive is our project bundle

Directory Structure of Cloned Project will look something like as shown in the image; the images show the sub-directories and files present inside the **python-predictive** folder,

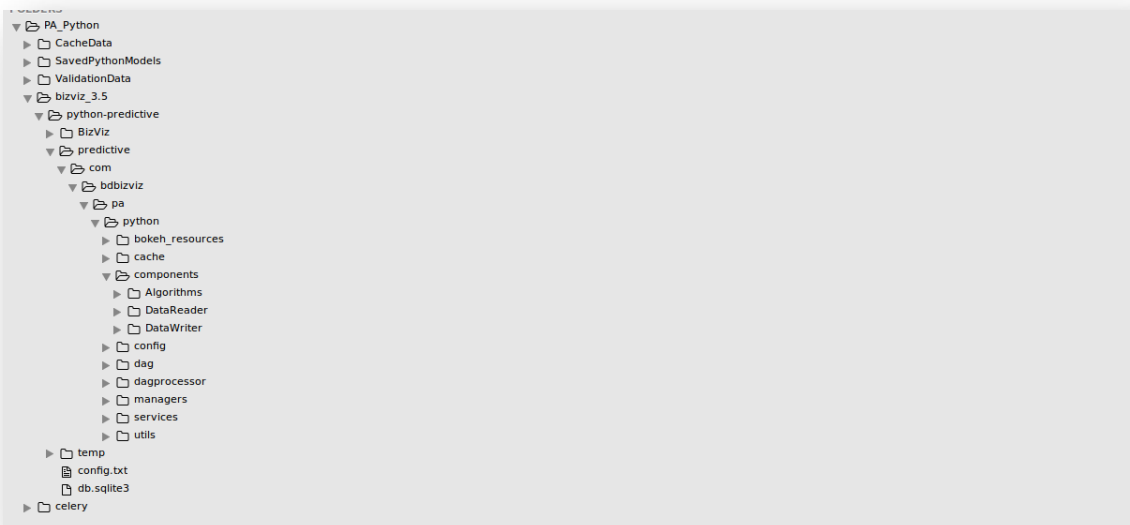


Fig. Directory Structure of Python Predictive

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
 - 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>
 - These details will 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

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

These above configurations are as per 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 will 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 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
 - `$ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>`
 - `$ source bin/activate`
 - `$ cd <path-to-python-predictive>`
 - `$ celery -A BizViz worker -l info -c 2`
 - # It will start Celery worker on BizViz app with Concurrency value = 2
- Open another Terminal, then execute below commands
 - `$ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>`
 - `$ source bin/activate`
 - `$ cd <path-to-python-predictive>`
 - `$ celery -A BizViz beat -l info`
 - # It will start Celery beat Scheduler on BizViz app
- Open another Terminal, then execute below commands
 - `$ cd <path-to-VIRTUAL_ENVIRONMENT_DIRECTORY>`
 - `$ source bin/activate`
 - `$ cd <path-to-python-predictive>`
 - `$ 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 package is missing.

2.2.4.5. Creating Django & Celery Services

Creating services will be very useful to work with Django Server, Celery Workers, Celery Scheduler. In this section, we will create Linux OS based services that we will 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/systemd/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** & **ExecReload** also
- 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.
# - %l 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%l.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
 - sudo systemctl start service name
- To Stop any Service
 - sudo systemctl stop service name
- To know the status of any service
 - 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 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 following commands:

- \$ ps -aux | grep java

It will show you a list of Java processes running on the system, then find the Tomcat process and 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 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.bizvizardscheduler
- com.bdbizviz.bizvizardschedulerhistory
- 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.engine

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.bizvizzescheduler
- com.bdbizviz.bizvizzeschedulerhistory
- 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.

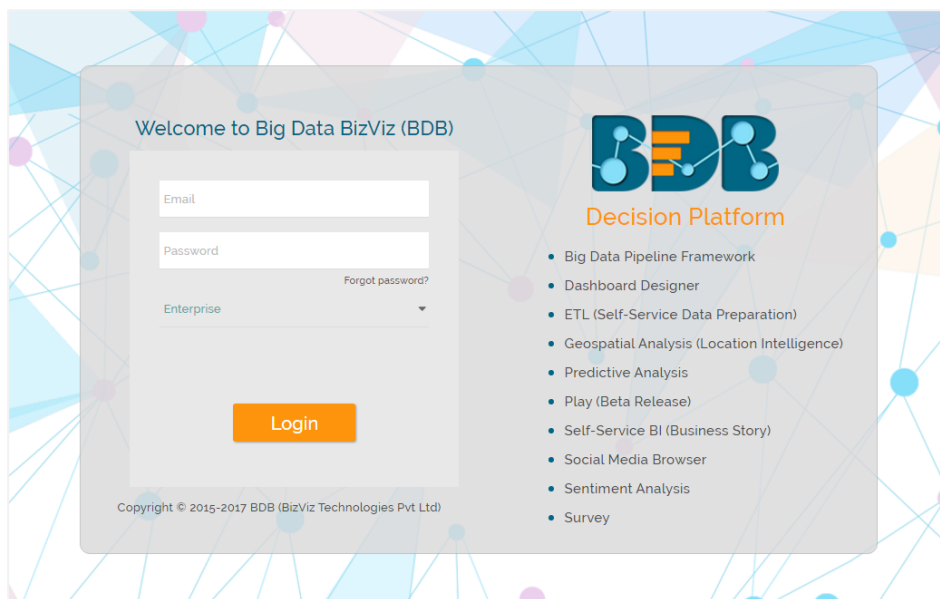
Use the following URL to check whether Karaf is started or not:

<http://<IP>:<Port>/cxf>

3. Getting Started with the BDB Predictive Analysis


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

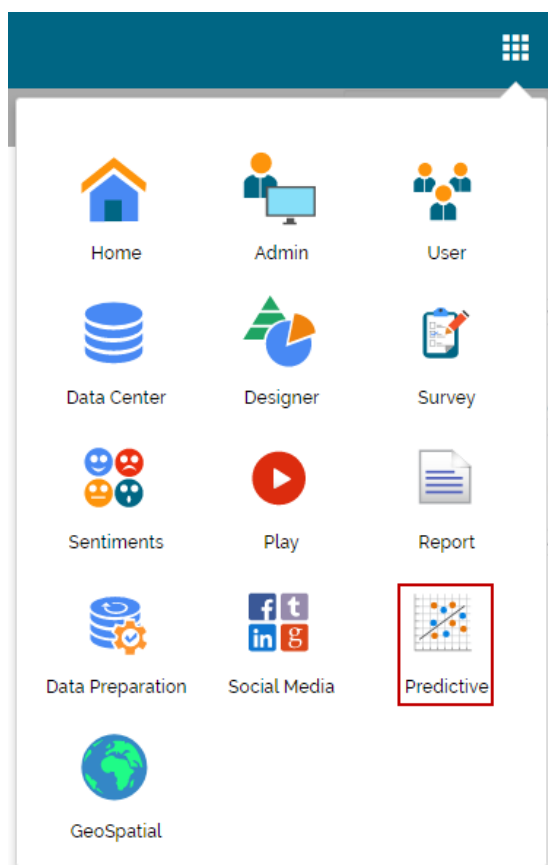
- i) Open BizViz Enterprise Platform Link: <http://apps.bdbizviz.com/app/>
- ii) Enter your credentials to Login.
- iii) Click 'LOGIN'



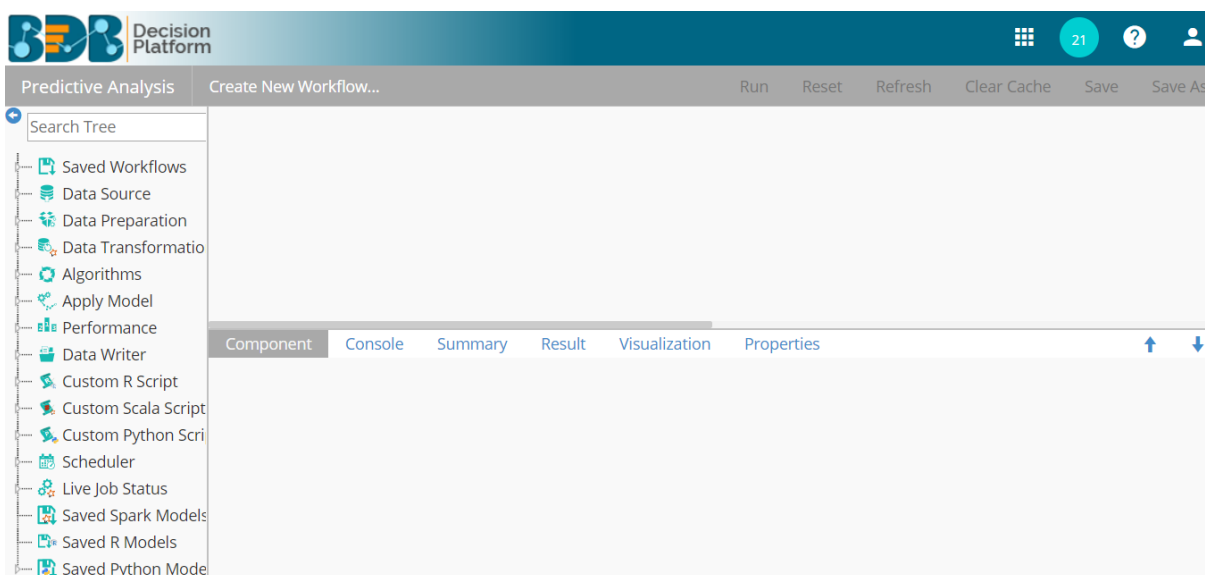
- iv) Users will be redirected to the BizViz Platform home page.



- v) Click the 'Apps'  icon to display all the plugin applications.
- vi) Select 'Predictive Analysis' from the Apps menu.



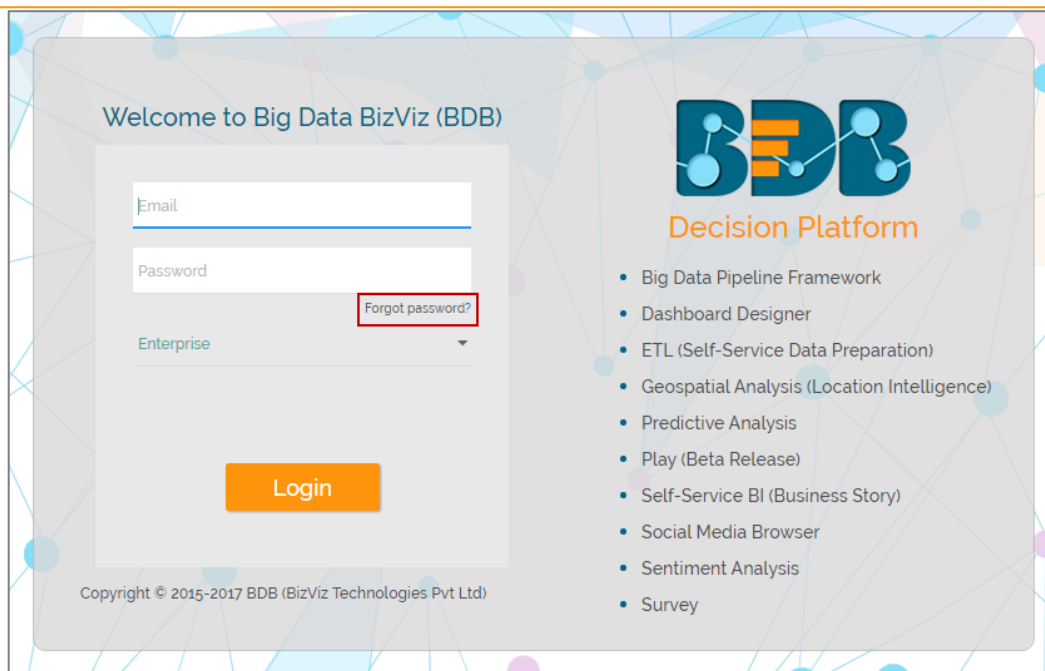
vii) Users will be directed to the Predictive Analysis home page.



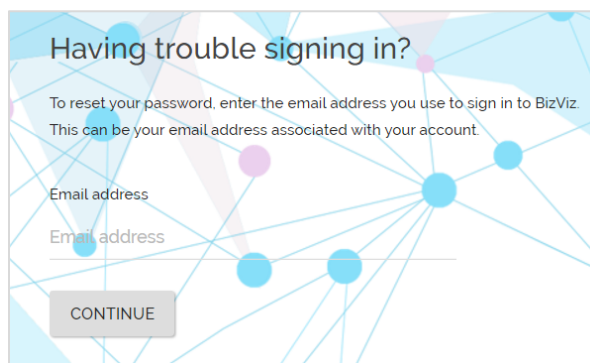
3.1. Forgot Password Option

Users are provided with a choice to change the password.

- i) Navigate to the Login page.
- ii) Click 'Forgot Your Password?' option.



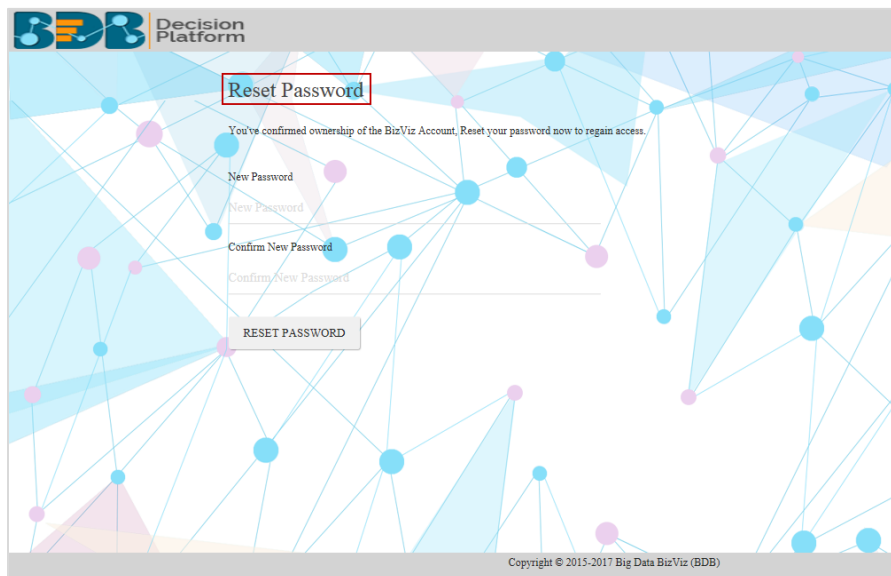
- iii) Users will be redirected to a new window.
- iv) Provide the email id that is registered with BDB to send the reset password link.
- v) Click 'Continue.'



- vi) Users will be directed to select a space and continue.



- vii) A reset password link will be sent through email.
- viii) Click on the link.
- ix) Users will be redirected to the 'Reset Password' page to set a new password.
 - a. Set a new password.
 - b. Confirm the newly set password.
 - c. Click 'RESET PASSWORD.'



- x) The password will be successfully reset.

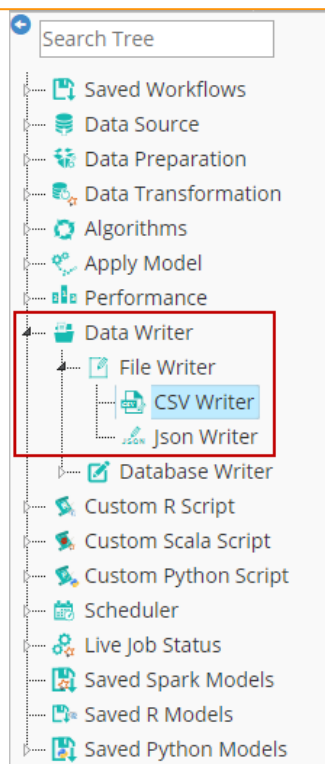
4. Predictive Analysis Home Page

This section describes all the options and icons provided on the Predictive Analysis home page. The Predictive Analysis home page can be described in the following Menu:


4.1. Tree-node Menu

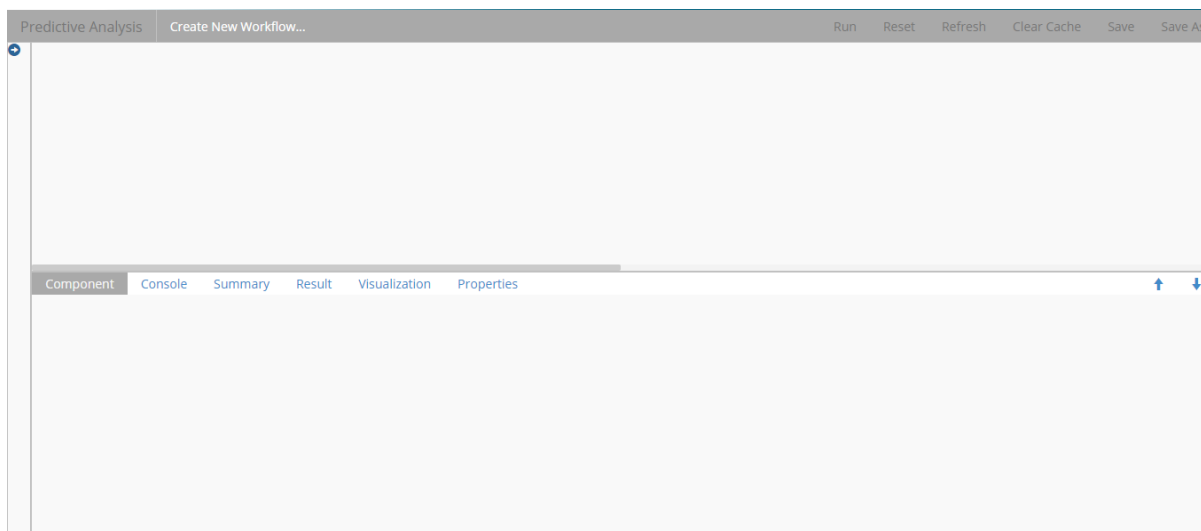
The Tree-node menu has all the available component connectors to run a predictive execution. The components will be provided in the hierarchical order via a tree structure menu. All the main categories are included as tree-nodes and sub-categories are committed as petals to the respective tree-nodes.

E.g. '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 'Arrow'  next to the 'Search' box to collapse the tree structure menu from the homepage.



- c. 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

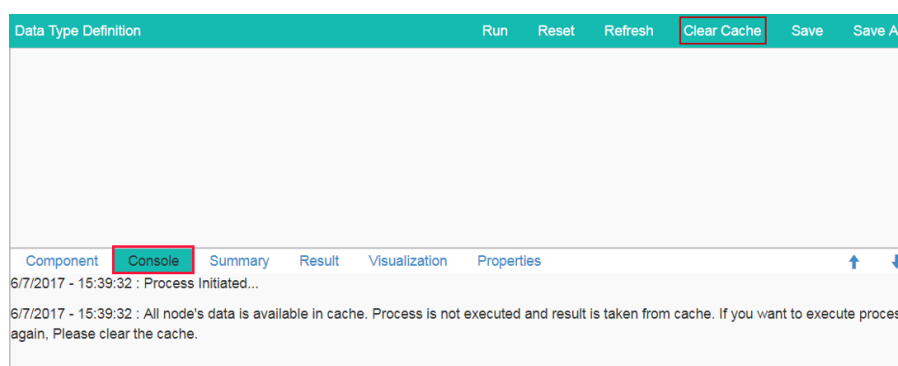
1. Run: Click 'Run' option to run the process and display the result set

view. This option can be applied to the data source, algorithms, and data preparation components.

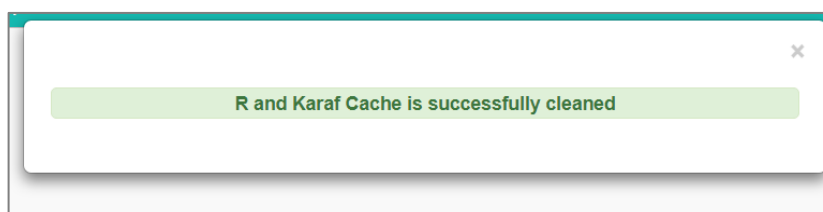
2. **Reset:** The **'Reset'** option to clean the workspace removing the current component connectors.
3. **Refresh:** The **'Refresh'** option is provided on the menu row to fetch fresh data when adding a new component to the **Spark workflow**.
4. **Clear Cache:**
 - a. After using the **'Run'** option, by default data will be cached in the server for the next 10 minutes. For 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:

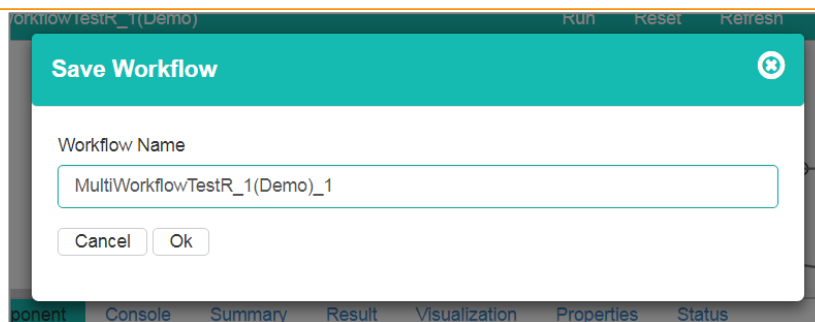
- i) Click **'Clear Cache'** option from the header menu.
- ii) A message will pop-up.
- iii) Click **'Ok.'**



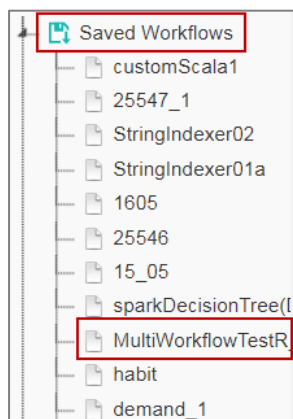
- iv) Another message will pop-up to confirm that the cache data has been cleared.



5. **Save:** Click the **'Save'** option to save the created predictive workflow.
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 pop-up window will appear for confirmation.
 - iv) Click **'Ok.'**

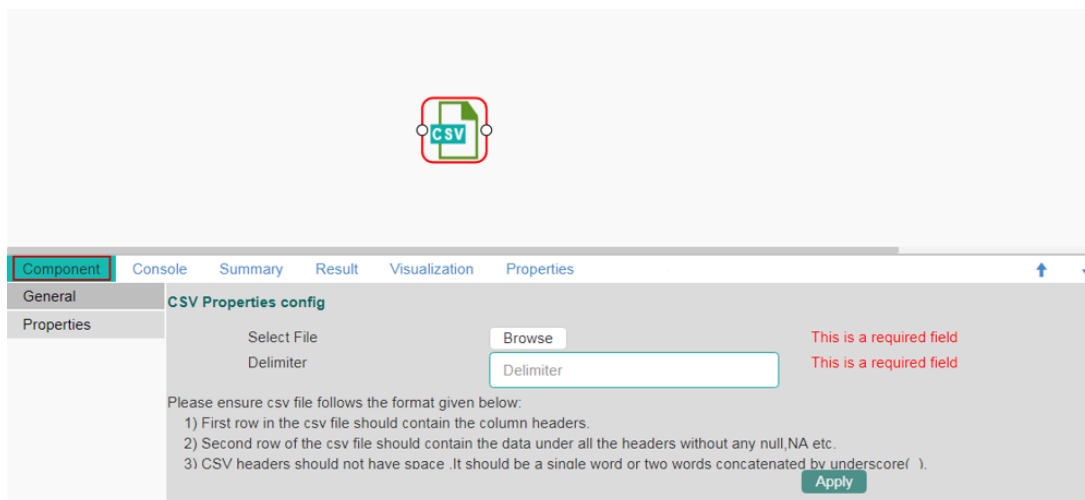


v) The workflow will be saved by the provided name in the ‘Saved Workflows’ list.



4.3. Tabbed Menu Strip - Options

1. **Component:** The ‘Component’ tab displays required configuration fields for the dragged elements onto the workspace.

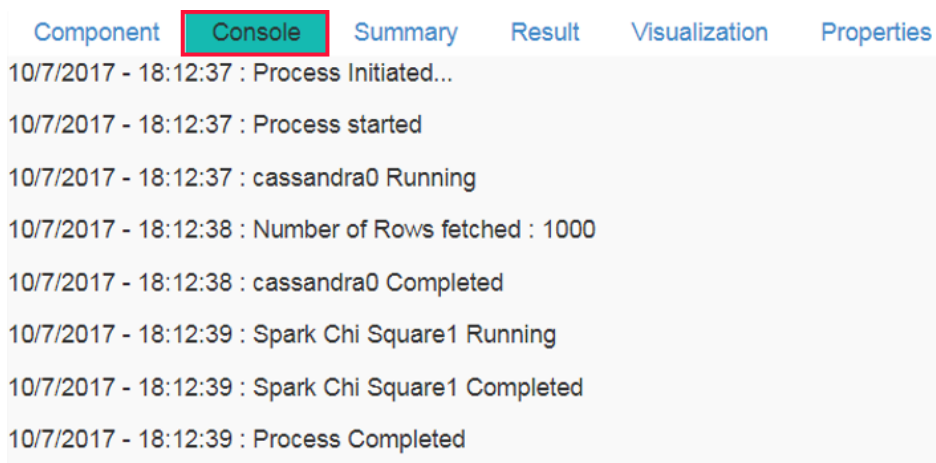


Note: The component tab may display various sub-tabs as per the selected components onto the workspace.

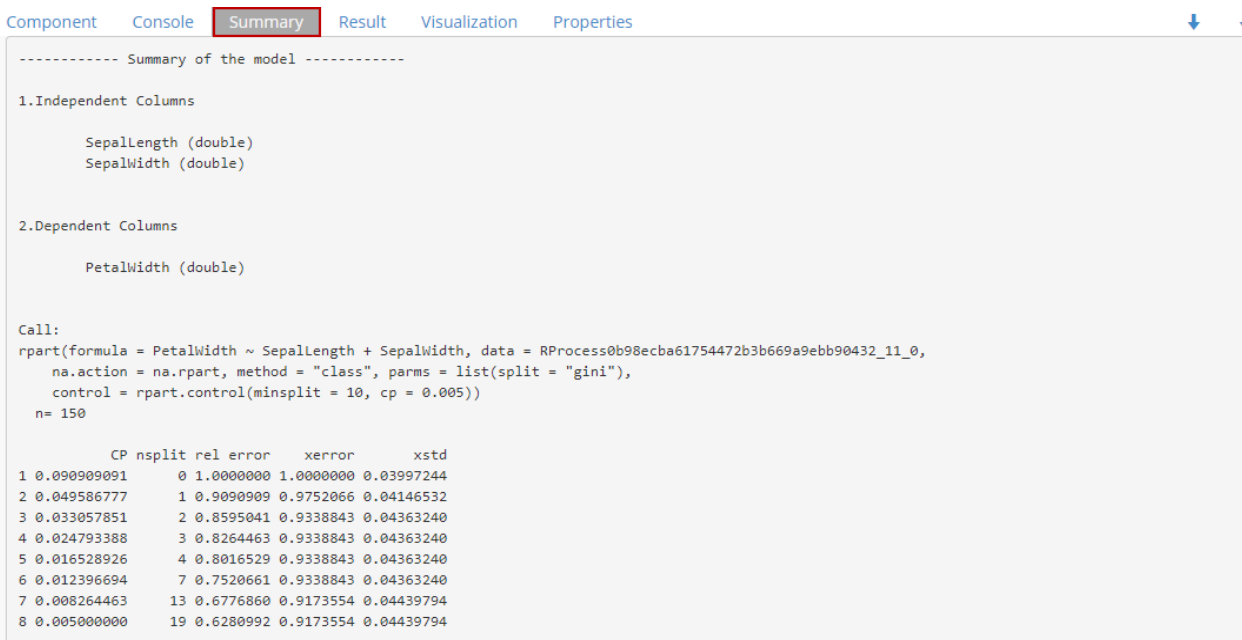
E.g., If the dragged data source is a CSV file, then the component tab will display General and Properties fields while for the Cassandra Reader as a data source, the component tab will display General, Properties, and Column Selection.

2. **Console:** The ‘Console’ tab displays 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 and Spark Process (starting and ending time)



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



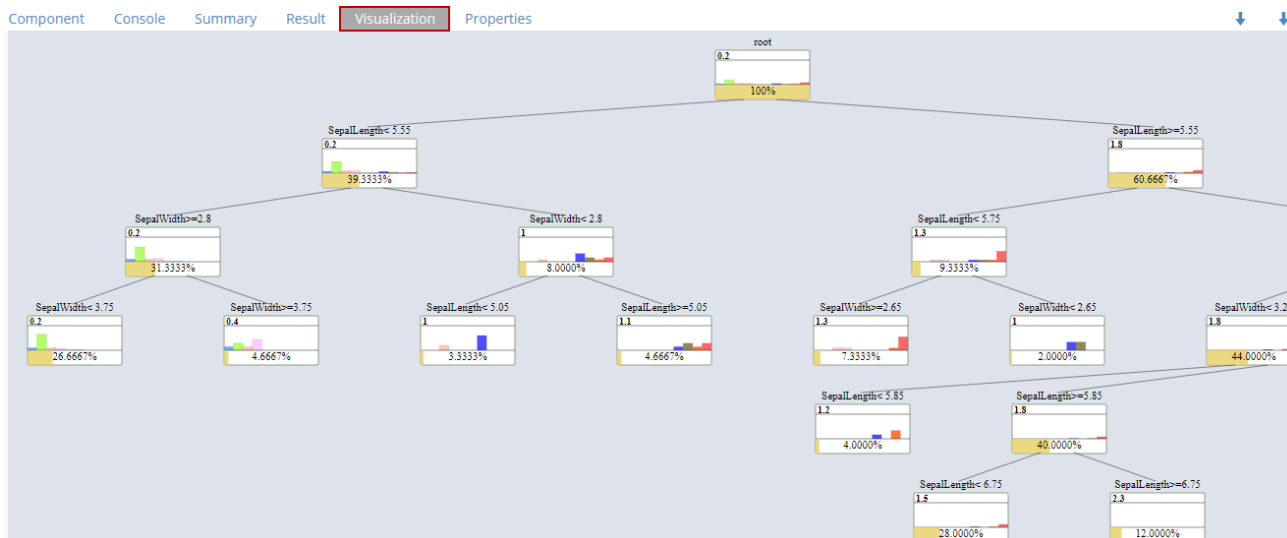
4. **Result:** Click the ‘Result’ tab to display a result list view based on the selected execution.

SepalLength	SepalWidth	PetalLength	PetalWidth	Species	ClusterNumber	idd	PredictedValues1	Probability1
5.1	3.5	1.4	0.2	setosa	2	1	0.2	0.65
4.9	3	1.4	0.2	setosa	5	2	0.2	0.65
4.7	3.2	1.3	0.2	setosa	5	3	0.2	0.65
4.6	3.1	1.5	0.2	setosa	5	4	0.2	0.65
5	3.6	1.4	0.2	setosa	2	5	0.2	0.65
5.4	3.9	1.7	0.4	setosa	2	6	0.4	0.4285714
4.6	3.4	1.4	0.3	setosa	5	7	0.2	0.65
5	3.4	1.5	0.2	setosa	2	8	0.2	0.65
4.4	2.9	1.4	0.2	setosa	5	9	0.2	0.65
4.9	3.1	1.5	0.1	setosa	5	10	0.2	0.65

Showing 1 to 10 of 150 entries

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.

Component	Console	Summary	Result	Visualization	Properties
Created By					admin
Created At					2018-01-09 11:25:05 +0530
Last Modified By					Raghvendra
Last Modified At					2018-01-12 17:52:07 +0530
Version					3.0.1

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

Workflow Name	Run by	Start time	End Time	Status	View Log	Live job status	Summary	Actions
untitled	Ranjit Krishnan	30/June/2017-11:12:46	NA	in progress				
untitled	Ranjit Krishnan	30/June/2017-10:59:15	30/June/2017-10:59:19	failed				
25546	Ranjit Krishnan	27/June/2017-12:24:12	NA	in progress				
Cassandraris	Ranjit Krishnan	26/June/2017-20:9:50	26/June/2017-20:14:46	failed				
untitled	Ranjit Krishnan	8/May/2017-17:2:32	8/May/2017-16:59:31	failed				
untitled	Ranjit Krishnan	24/Apr/2017-15:42:49	NA	in progress				
saveFilter	Ranjit Krishnan	8/Mar/2017-11:56:7	8/Mar/2017-11:56:28	success				
testnaive	Ranjit Krishnan	28/Feb/2017-18:6:18	28/Feb/2017-18:9:50	success				
untitled	Ranjit Krishnan	13/Feb/2017-12:25:12	NA	in progress				
kmean	Ranjit Krishnan	10/Feb/2017-15:57:40	10/Feb/2017-16:0:25	success				

Showing 1 to 10 of 14 entries

Previous 1 2 Next

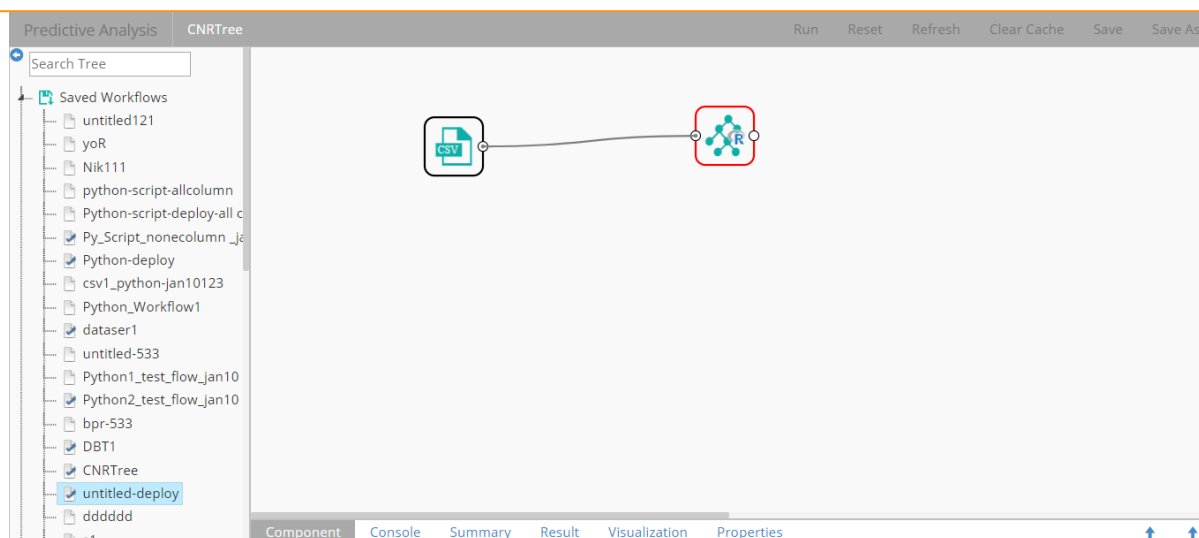
Note: The Status tab will appear when users need to check the live job status of a running job.


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.

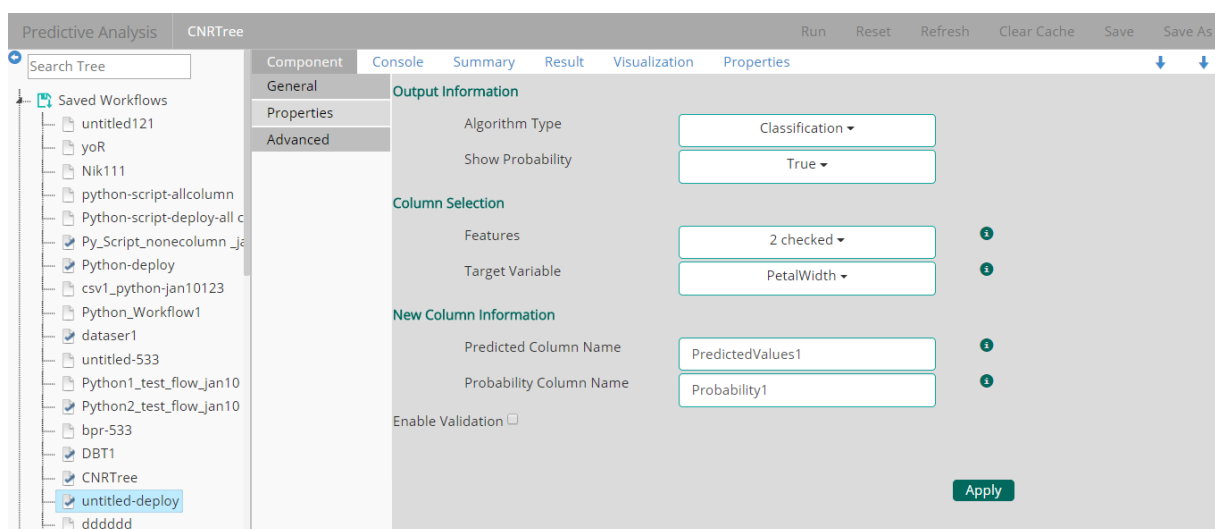
The Predictive homepage default view is as displayed below:

The screenshot shows the Predictive Analysis interface. On the left is a 'Saved Workflows' tree with 'untitled-deploy' selected. The main workspace displays a workflow diagram with a 'CSV' node connected to a 'R' (Random Forest) node. Below the workspace is a configuration panel with tabs for 'Component', 'Console', 'Summary', 'Result', 'Visualization', and 'Properties'. The 'Properties' tab is active, showing 'Output Information' with 'Algorithm Type' set to 'Classification' and 'Show Probability' set to 'True'. The 'Column Selection' section shows 'Features' set to '2 checked'. An 'Apply' button is visible at the bottom right of the configuration panel.

a. Click the downward sign to minimize view space and maximize workspace on the Predictive Analysis home page.



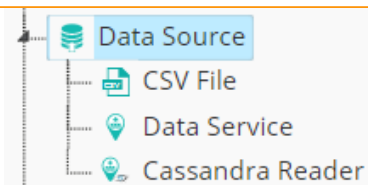
- b. Click the upward sign  to maximize view space and minimize workspace on the Predictive Analysis home page.



5. Getting Data from a Data Source

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

- a. CSV File
- b. Query Service
- c. Cassandra Reader

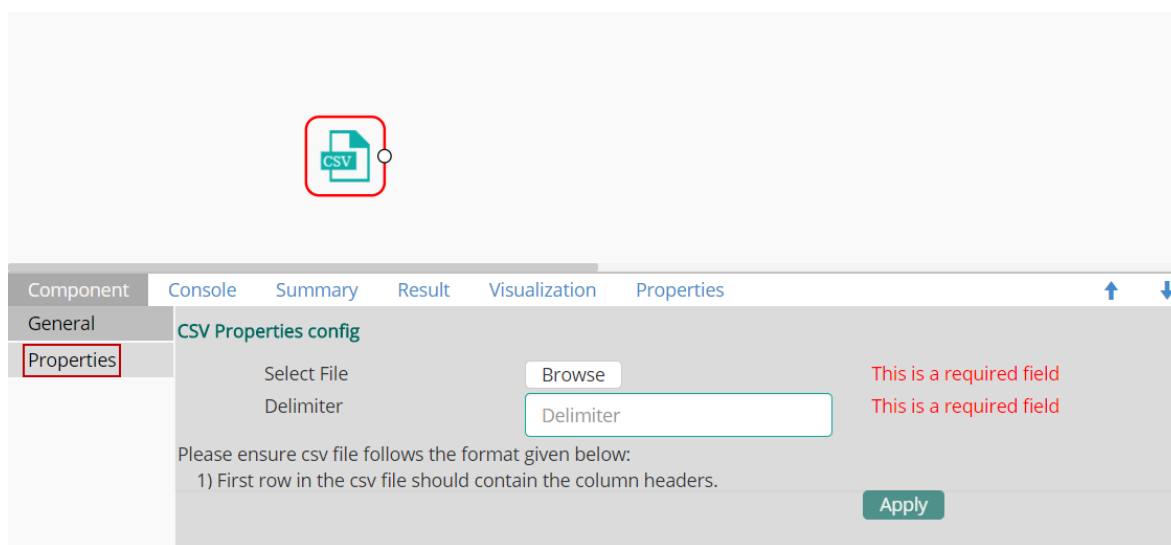


5.1. Getting Data from a CSV File

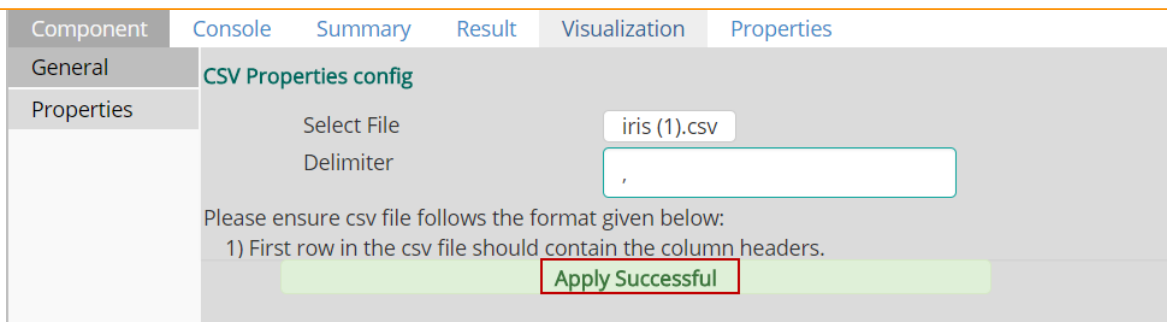
- i) Select and drag 'CSV File' component onto the workspace.
- ii) Click the 'CSV File' component.



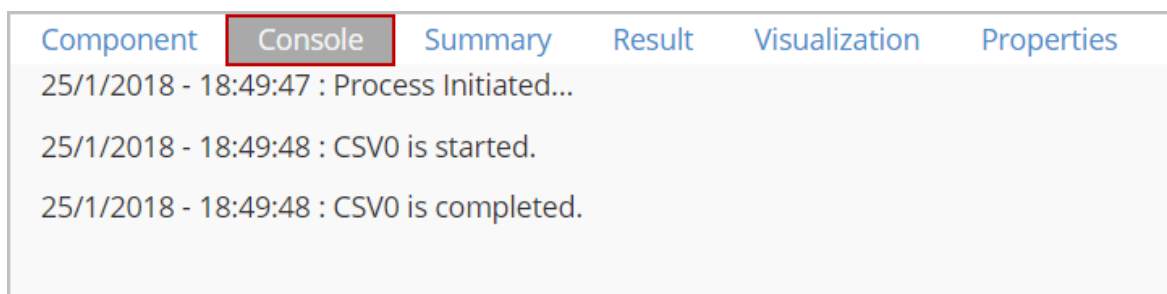
- iii) Configure the following 'CSV Properties Configuration' fields:
 - a. **Select File:** Browse a CSV file
 - b. **Delimiter:** Mention the delimiter used in the CSV file
- iv) Click 'Apply.'



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



- vi) Click 'Run'
- vii) Users will be redirected to the 'Console' tab



- viii) 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.

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

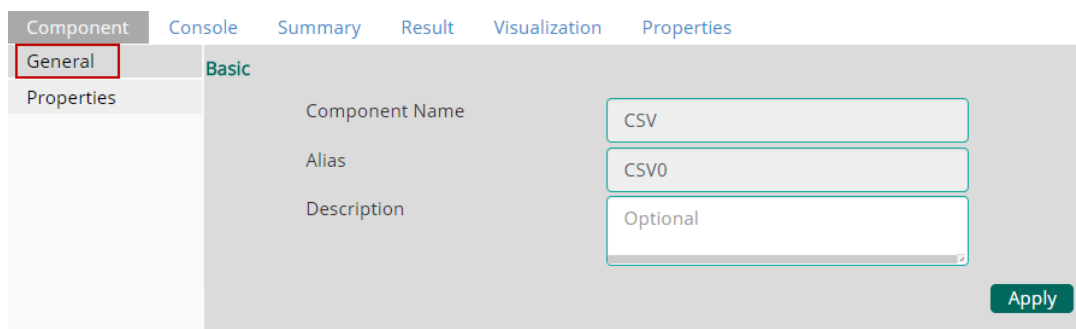
● **Rules to be followed while uploading a CSV File**

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.

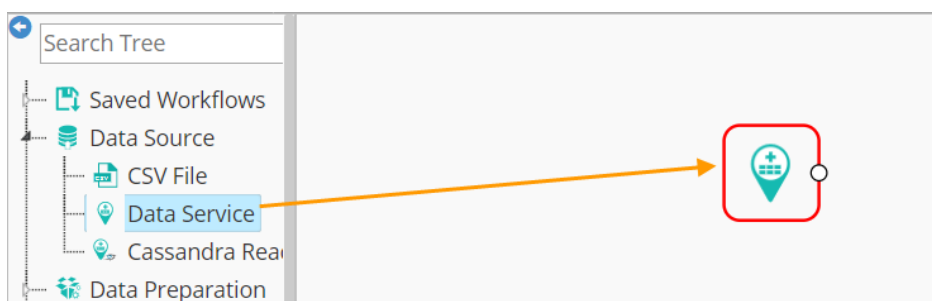
Note:

- 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. Alias Name
 - ii. Description (it is an optional field)
(E.g. the following image displays ‘General’ tab for a CSV data source.)



5.2. Getting Data from a Data Service

- i) Select and drag ‘Data Service’ connector onto the workspace.
- ii) Click the ‘Data Service’ connector.



- 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’

Component	Console	Summary	Result	Visualization	Properties
General	Data Service Properties				
Properties	Select Data Connector		<input type="text" value="pred"/>		
	Select Data Service		<input type="text" value="iris_new"/>		
	Fields				
	Column Header				Data type
	sepallength				double
	sepalwidth				double
	petallength				double
	petalwidth				double
	species				string
					<input type="button" value="Next"/>

- 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	<input type="text" value="val"/>		<input type="text" value="Text"/>		
Conditions					
					<input type="button" value="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	<input type="text" value="val"/>		<input type="text" value="LOV"/>		
Conditions	Select Data Connector		<input type="text" value="Select"/>		
	Select Data Service		<input type="text" value="Select"/>		
					<input type="button" value="Apply"/>

- viii) Click 'Apply'.
- ix) Click 'Run.'
- x) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

29/1/2018 - 12:41:19 : Process Initiated...

29/1/2018 - 12:41:19 : Data Service0 is started.

29/1/2018 - 12:41:20 : Data Service0 is completed.

- xi) 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:

sepalwidth	sepalwidth	petalwidth	petalwidth	species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
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**

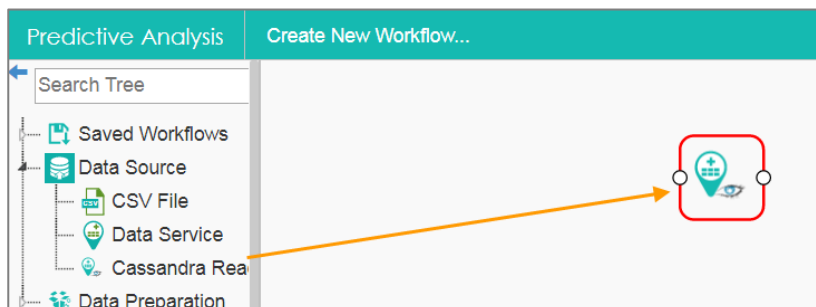
1. Data service header should not have space. It should be a single word or two words concatenated by an underscore (_).
2. Data service header should not contain any special characters. E.g. - %, #, \$, @, *, etc.
3. Data service header should not contain single or double quotes, dot, brackets, and 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:

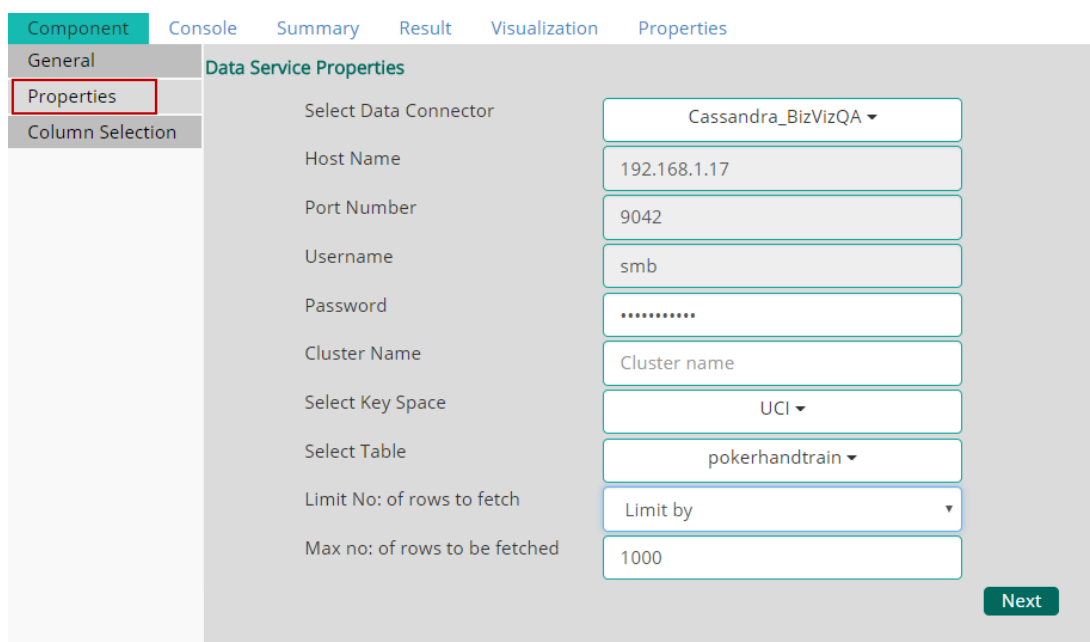
- a. Users can develop a data service via the Data Management module of the BizViz Platform.
- b. 'Fields' option under 'Properties' tab will appear only after selecting the appropriate query service.
- c. LOV service provided under 'Conditions' tab can contain only one column, in case of more than one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

5.3. Getting Data from a Cassandra Reader

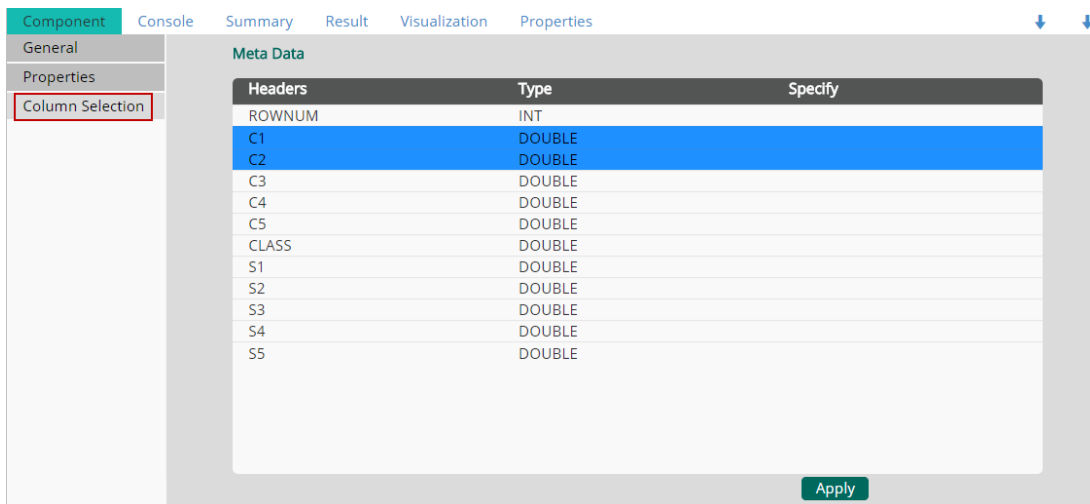
- i) Select and drag 'Cassandra Reader' connector onto the workspace.
- ii) Click on the 'Cassandra Reader' connector.



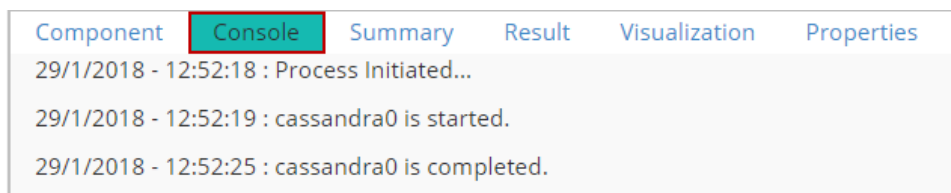
- iii) Users will be redirected to the 'Properties' tab.
- iv) Configure the required properties:
 - a. Select Data Connector: Select a data connector using the drop-down menu
 - b. Host Name: Data connector specific hostname will be displayed
 - c. Port Number: Port number will be displayed
 - d. User 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 by Row: Select an option using the drop-down menu. Two options will be provided as shown below:
 1. Select all Rows
 2. Limit By
 - b. 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.'



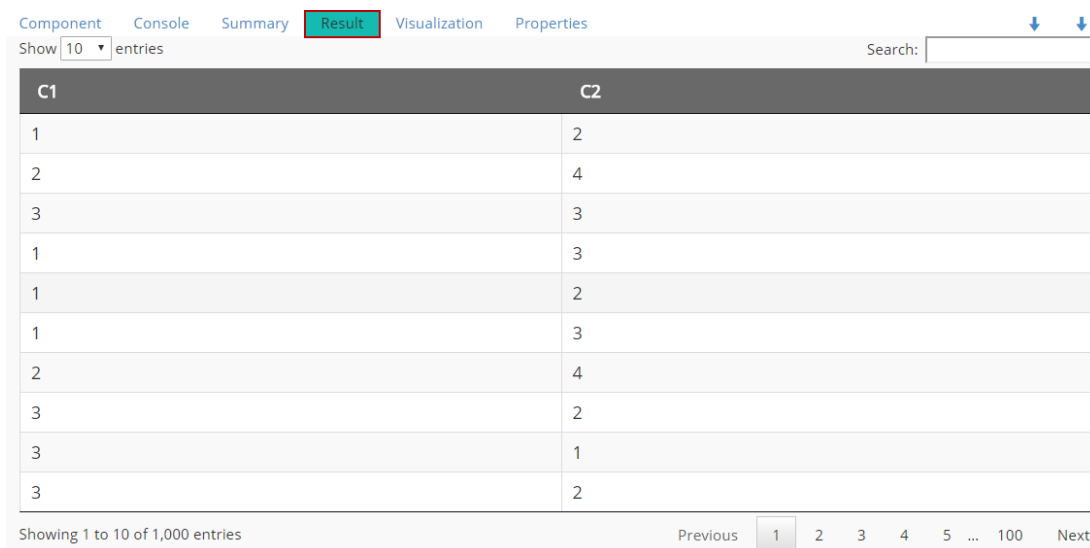
- vi) Users will be redirected to the ‘Column Selection’ tab.
- vii) Select the required columns from the list.
- viii) Click ‘Apply’.



- ix) Click ‘Run.’
- x) Users will be redirected to the ‘Console’ tab.



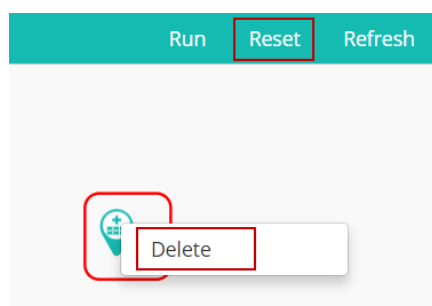
- xi) Follow the below given steps to display the result view:
 - a. Click the dragged data source component on the workspace.
 - b. Click the ‘Result’ tab.



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.

5.4. Removing a Data Source from the Workspace

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



- iv) The selected Data Source connector will be removed from the workspace.
OR
 Click on the **'Reset'** option to remove the connector(s) from the workspace.

Note: The same set of steps can be followed to remove a Data Service and Cassandra Reader data sourced from the workspace.

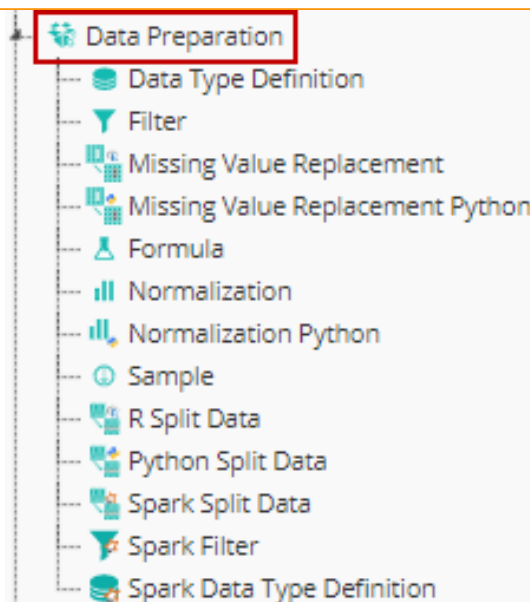
6. Data Preparation

Components provided under **'Data Preparation'** 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.

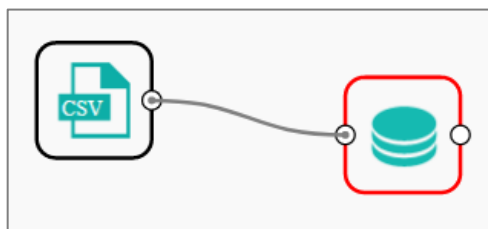
6.1. Data Type Definition

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 will open.



- 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'.

Component Console Summary Result Visualization Properties

General Data Type Mapping

Properties

ColumnName	AliasName	PrimaryDataType	DateFormat
SepalLen; ▾	sl	Double ▾	- +
SepalWid ▾	sw	Double ▾	- +
PetalLen; ▾	pl	Double ▾	- +

Apply

- ix) Click 'Run.'
- x) Users will be directed to the 'Console' tab.

Component Console Summary Result Visualization Properties

29/1/2018 - 13:14:32 : Process Initiated...

29/1/2018 - 13:14:32 : CSV0 is started.

29/1/2018 - 13:14:32 : CSV0 is completed.

29/1/2018 - 13:14:32 : Data Type Definition1 is started.

29/1/2018 - 13:14:32 : Data Type Definition1 is completed.

- xi) 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.
 Users can see the given column names on the selected columns in the 'Result' data.

Component Console Summary Result Visualization Properties

Show 10 entries Search: []

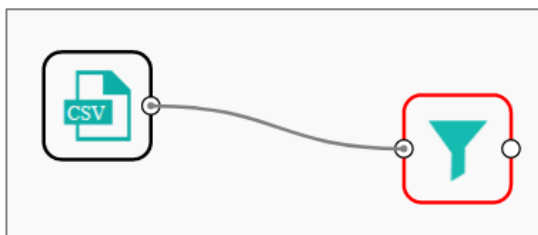
Number	sl	sw	pl	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

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

6.2. Filter

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

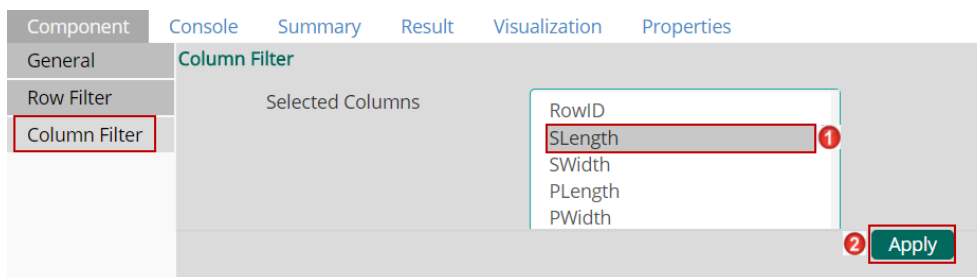
- i) Select and Drag 'Filter' component onto the workspace.
- ii) Connect the 'Filter' component to a configured data source component.



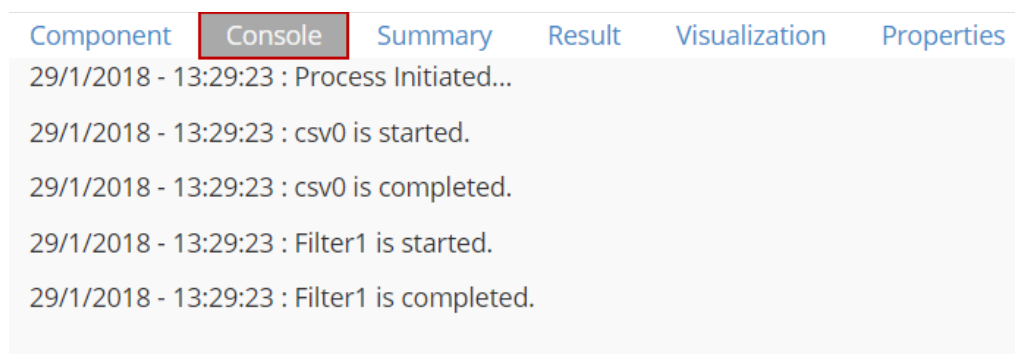
- iii) Configure the filter component as described below:

6.3. Column Filter

- i) Select a column from the 'Selected Columns' context menu.
- ii) Click 'Apply' to configure the data.



- iii) Click 'Run'
- iv) Users will be redirected to the 'Console' tab.



- v) Follow the below given steps to display the result view:
 - a. Click the dragged algorithm component in the workspace.
 - b. Click the 'Result' tab.
- vi) The filtered data will be displayed via the 'Result' tab.

Component	Console	Summary	Result	Visualization	Properties
Show	10	entries	Search:		
SLength					
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**, **Functions**, and **Operators** list menus
 - b. A formula will be entered in the given box
 - c. Click **'Apply'**.

Component	Console	Summary	Result	Visualization	Properties
General	Column Filter				
Row Filter	Selected Columns				
Column Filter	<div style="border: 1px solid gray; padding: 5px; width: fit-content;"> Number SepalLength SepalWidth PetalLength PetalWidth Species </div>				
Apply					

- viii) Click **'Run'**
- ix) Users will be redirected to the **'Console'** tab.

Component **Console** Summary Result Visualization Properties

29/1/2018 - 13:48:17 : Process Initiated...

29/1/2018 - 13:48:18 : CSV0 is started.

29/1/2018 - 13:48:18 : CSV0 is completed.

29/1/2018 - 13:48:18 : Filter1 is started.

29/1/2018 - 13:48:18 : Filter1 is completed.

- 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 filtered data will be displayed via the 'Result' tab.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
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
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3	1.4	0.1	setosa
14	4.3	3	1.1	0.1	setosa
15	5.8	4	1.2	0.2	setosa

Showing 1 to 10 of 145 entries Previous 1 2 3 4 5 ... 15 Next

Note:

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

6.4. 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)

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
	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'

Component Console Summary Result Visualization Properties

General Replacement Values

Properties

Column Name Replacement Options

SepalLength Maximum - +

Species Custom Replacement - +

Species

Apply

- vii) Click 'Run'
- viii) Users will be redirected to the 'Console' tab.

Component Console Summary Result Visualization Properties

30/1/2018 - 13:43:11 : Process Initiated...

30/1/2018 - 13:43:11 : CSV1 is started.

30/1/2018 - 13:43:12 : CSV1 is completed.

30/1/2018 - 13:43:12 : Missing Data Replacement0 is started.

30/1/2018 - 13:43:12 : Missing Data Replacement0 is completed.

- 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.

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	Species
7.9	3.9	1.7	0.4	Species
7.9	3.4	1.4	0.3	Species
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

6.5. 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 'Result' tab. (in this case, the selected input data is displayed in the following image)

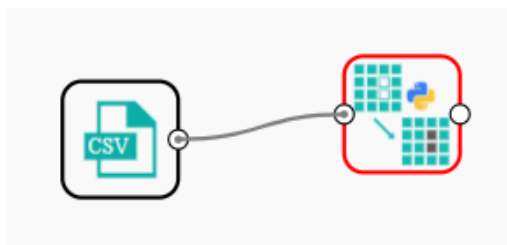
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth
1	5.1	3.5	1.4	0.2
	4.9		1.4	0.2
3	4.7	3.2		0.2
4	4.6	3.1	1.5	0.2
5	5	3.6	1.4	
6	5.4	NaN	NaN	0.4
	4.6	3.4	1.4	0.3
8	5	3.4	1.5	0.2
9	4.4			
10	4.9	3.1	1.5	0.1

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

- ii) Select and drag '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:
- Column Name:** Select a column using the drop-down that contains some missing values.
 - Replacement Options:** Select a replacement option using the drop-down menu. The following replacement options are provided under this field:
 - Mean
 - Median
 - Mode
 - Maximum
 - Minimum
 - Remove Entire Row
 - Remove Entire Column
 - Custom Replacement
 - Missing Value:** Users can get two options under this field.
 - NaN
 - Custom
- v) Click 'Apply'

Component Console Summary Result Visualization Properties

General Replacement Values

Properties

Column Name	Replacement Options	Missing Value	-	+
Number	Most Frequent	NaN	-	+
SepalLength	Mean	Custom	-	+
		5.1		

Apply

- vi) Click 'Run'
- vii) Users will be redirected to the 'Console' tab.

Component Console Summary Result Visualization Properties

```

2/4/2018 - 15:56:15 : Process Initiated...
2/4/2018 - 15:56:17 : CSV0 is started.
2/4/2018 - 15:56:17 : CSV0 is completed.
2/4/2018 - 15:56:17 : Python-Missing Data Replacement1 is started.
2/4/2018 - 15:56:17 : Python-Missing Data Replacement1 is completed.
  
```

- 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:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth
1	5.4	3.7	1.5	0.2
12	4.8	3.4	NaN	0.2
13	4.8	3	1.4	0.1
14	4.3	3		
15	5.8	4	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4
1	5.890780141843971		1.4	0.3
1	5.7	3.8	NaN	0.3
20	5.890780141843971	3.8	1.5	0.3

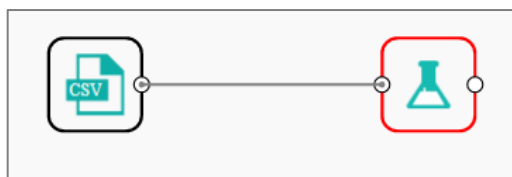
Showing 11 to 20 of 150 entries Previous 1 2 3 4 5 ... 15 Next

6.6. Formula

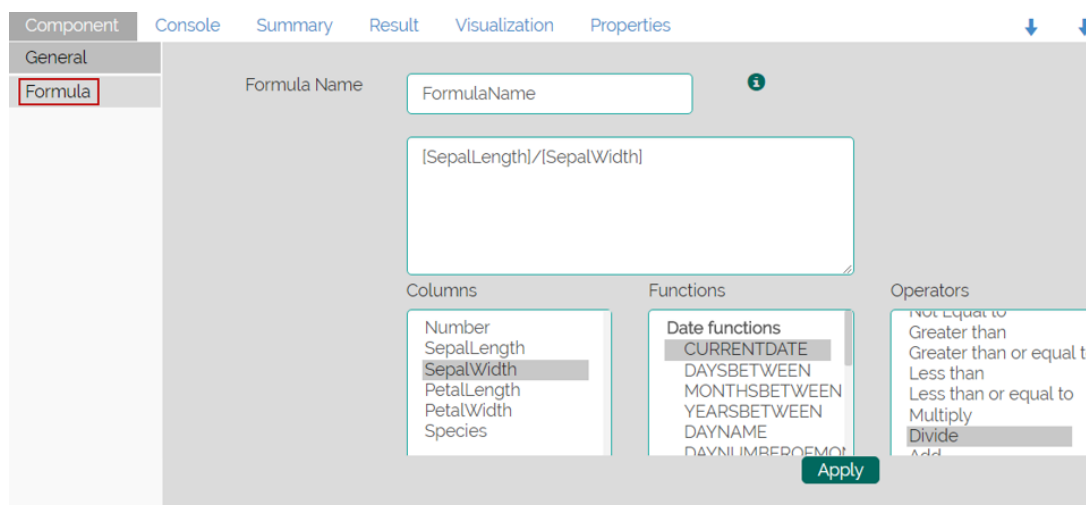
Users can create a calculated column using 'Formula.' A formula can be formed by using available columns, functions, and operators.

- i) Select and drag 'Formula' component onto the workspace.

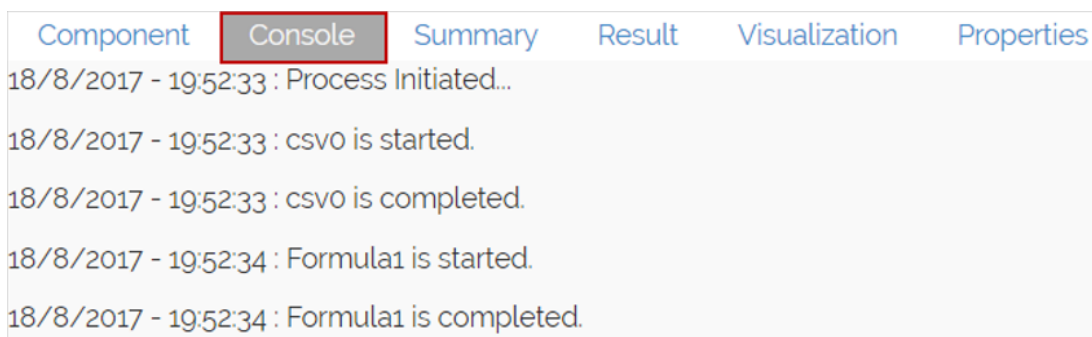
- ii) Connect the 'Formula' component to a configured data source.
- iii) Click on the 'Formula' component.



- 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 'Run.'
- vi) Users will be redirected to the 'Console' tab.



- vii) 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:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	FormulaName
1	5.1	3.5	1.4	0.2	setosa	1457
2	4.9	3	1.4	0.2	setosa	1633
3	4.7	3.2	1.3	0.2	setosa	1469
4	4.6	3.1	1.5	0.2	setosa	1484
5	5	3.6	1.4	0.2	setosa	1389
6	5.4	3.9	1.7	0.4	setosa	1385
7	4.6	3.4	1.4	0.3	setosa	1353
8	5	3.4	1.5	0.2	setosa	1471
9	4.4	2.9	1.4	0.2	setosa	1517
10	4.9	3.1	1.5	0.1	setosa	1581

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

6.7. 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.

6.7.1. Min-Max Normalization

It implements a linear transformation on the original data values and sets a new range for all the data values to fit in. The user can fix 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 'Run.'
- vii) Users will be directed to 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.

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
141	6.67	3.1	5.6	2.4	virginica
142	7.22	3.1	5.1	2.3	virginica
143	4.17	2.7	5.1	1.9	virginica
144	6.94	3.2	5.9	2.3	virginica
145	6.67	3.3	5.7	2.5	virginica
146	6.67	3	5.2	2.3	virginica
147	5.56	2.5	5	1.9	virginica
148	6.11	3	5.2	2	virginica
149	5.28	3.4	5.4	2.3	virginica
150	4.44	3	5.1	1.8	virginica

6.7.2. 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 average. 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.

- vi) Click ‘Run’
- vii) Users will be directed to the ‘Console’ tab.

```

Component Console Summary Result Visualization Properties
21/8/2017 - 11:54:6 : Process Initiated...
21/8/2017 - 11:54:6 : csv0 is started.
21/8/2017 - 11:54:6 : csv0 is completed.
21/8/2017 - 11:54:6 : Normalization1 is started.
21/8/2017 - 11:54:6 : Normalization1 is completed.
  
```

- 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	-0.9	3.5	14	0.2	setosa
2	-1.14	3	14	0.2	setosa
3	-1.38	3.2	13	0.2	setosa
4	-1.5	3.1	15	0.2	setosa
5	-1.02	3.6	14	0.2	setosa
6	-0.54	3.9	17	0.4	setosa
7	-1.5	3.4	14	0.3	setosa
8	-1.02	3.4	15	0.2	setosa
9	-1.74	2.9	14	0.2	setosa
10	-1.14	3.1	15	0.1	setosa

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

6.7.3. 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:

Component Console Summary **Result** Visualization Properties

General **Column Selection**

Select a Column

Behavior

Normalization Type

- vi) Click 'Run.'
- vii) Users will be directed to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
21/8/2017 - 11:35:40 : Process Initiated...
21/8/2017 - 11:35:40 : csv0 is started.
21/8/2017 - 11:35:40 : csv0 is completed.
21/8/2017 - 11:35:40 : Normalization1 is started.
21/8/2017 - 11:35:40 : Normalization1 is completed.

```

- ix) Follow the below given steps to display the result view:
- Click the dragged data preparation component on the workspace.
 - Click the 'Result' tab.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	0.51	3.5	14	0.2	setosa
2	0.49	3	14	0.2	setosa
3	0.47	3.2	13	0.2	setosa
4	0.46	3.1	15	0.2	setosa
5	0.5	3.6	14	0.2	setosa
6	0.54	3.9	17	0.4	setosa
7	0.46	3.4	14	0.3	setosa
8	0.5	3.4	15	0.2	setosa
9	0.44	2.9	14	0.2	setosa
10	0.49	3.1	15	0.1	setosa

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

Note:

- Normalization displays columns containing only numerical data.
- 'New Maximum Value' must be greater than 'New Minimum Value.'

6.8. Normalization Python

Normalization components transform data from more extensive range to 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

6.8.1. 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,

$$X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))$$

$$X_scaled = X_std * (max - min) + min$$

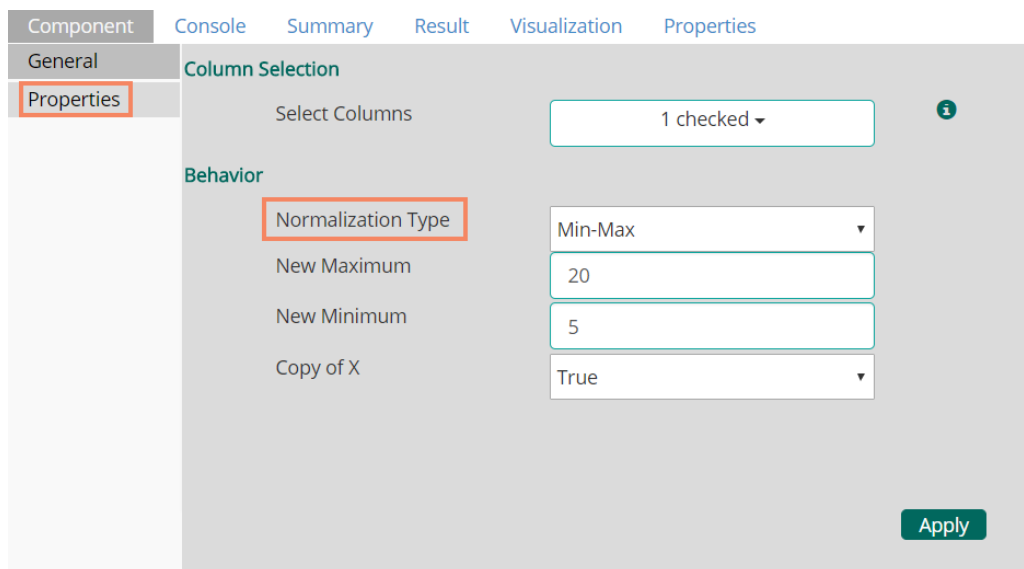
Where min, max= feature_range

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'**
- v) Click **'Apply'**.



- vi) Click **'Run.'**
- vii) Users will be directed to the **'Console'** tab.

Component **Console** Summary Result Visualization Properties

2/4/2018 - 16:31:59 : Process Initiated...

2/4/2018 - 16:32:1 : CSV0 is started.

2/4/2018 - 16:32:1 : CSV0 is completed.

2/4/2018 - 16:32:1 : Python-Normalization1 is started.

2/4/2018 - 16:32:1 : Python-Normalization1 is completed.

- 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

6.8.2. Maximum Absolute Scaler

Minimum 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'
- iii) Click 'Apply'.

Component Console Summary Result Visualization Properties

General Column Selection

Properties

Select Columns 1 checked

Behavior

Normalization Type Maximum Absolute Scaler

Copy of X True

Apply

- iv) Click 'Run.'
- v) Users will be directed to the 'Console' tab.

Component Console Summary Result Visualization Properties

2/4/2018 - 16:31:59 : Process Initiated...

2/4/2018 - 16:32:1 : CSV0 is started.

2/4/2018 - 16:32:1 : CSV0 is completed.

2/4/2018 - 16:32:1 : Python-Normalization1 is started.

2/4/2018 - 16:32:1 : 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	0.6456	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

6.8.3. 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 matrix 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

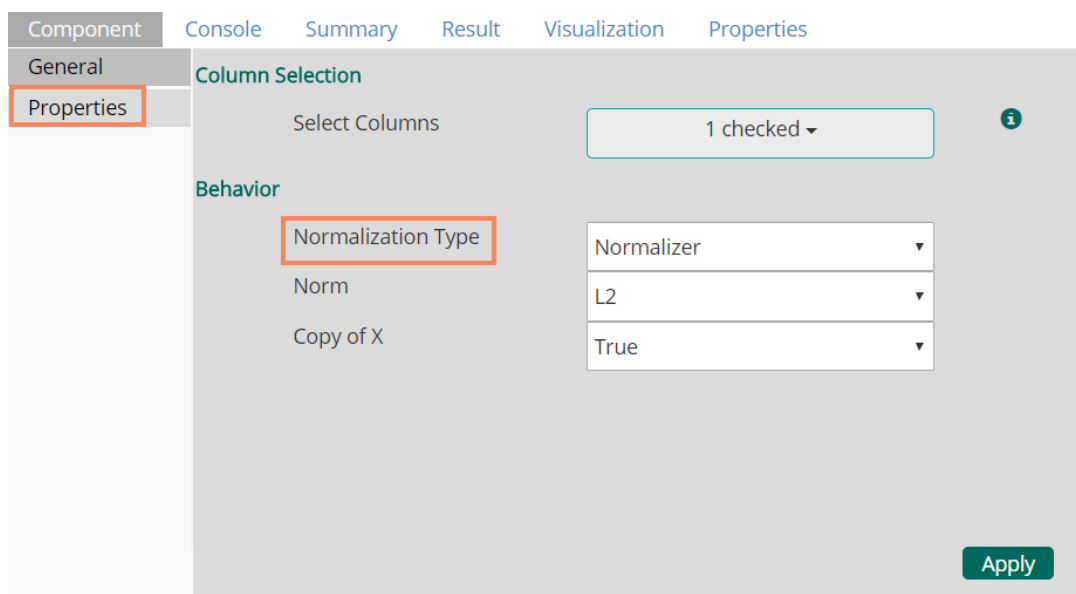
a. **Column Selection**

- 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'

- iii) Click 'Apply'.



- iv) Click 'Run.'
- v) Users will be directed to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

```
2/4/2018 - 16:31:59 : Process Initiated...
2/4/2018 - 16:32:1 : CSV0 is started.
2/4/2018 - 16:32:1 : CSV0 is completed.
2/4/2018 - 16:32:1 : Python-Normalization1 is started.
2/4/2018 - 16:32:1 : 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	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

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

6.8.4. 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 normally 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

- ii. **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'
- iii. **With Std. Dev:** Select an option from the drop-down menu out of 'True' or 'False'
- iv. **Copy of X:** Select an option from the drop-down menu out of 'True' or 'False'

iii) Click 'Apply'.

iv) Click 'Run.'

v) Users will be directed to the 'Console' 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

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
141	1.038	3.1	5.6	2.4	virginica
142	1.2803	3.1	5.1	2.3	virginica
143	-0.0525	2.7	5.1	1.9	virginica
144	1.1592	3.2	5.9	2.3	virginica
145	1.038	3.3	5.7	2.5	virginica
146	1.038	3	5.2	2.3	virginica
147	0.5533	2.5	5	1.9	virginica
148	0.7957	3	5.2	2	virginica
149	0.4322	3.4	5.4	2.3	virginica
150	0.0687	3	5.1	1.8	virginica

Showing 141 to 150 of 150 entries Previous 1 ... 11 12 13 14 15 Next

6.9. Sample

This component can be used to select a subsection of data from a large dataset.

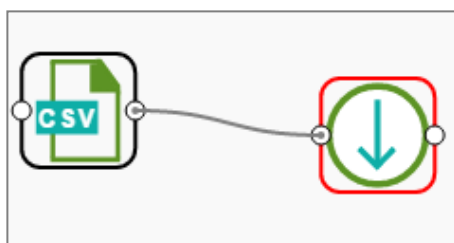
The sample component supports the following sample types:

6.9.1. Sampling Methods

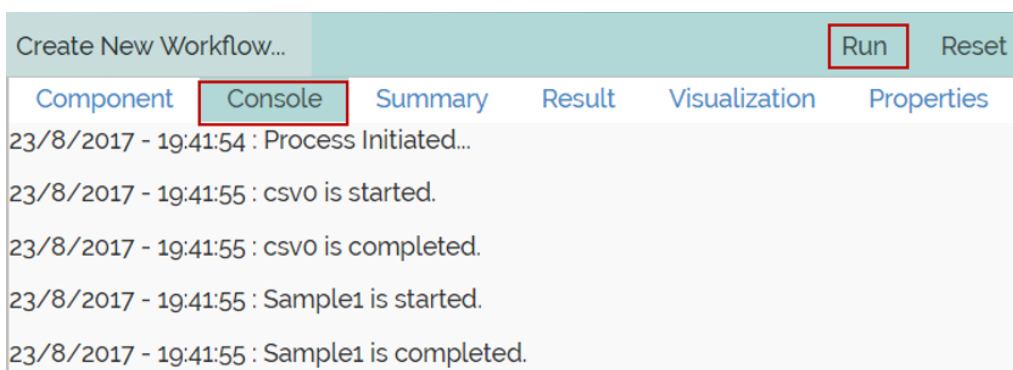
1. **First N:** It will select first N records from the data source. E.g., If the chosen value for “N” is 10, then it will select first 10 records from the data.
2. **Last N:** It will select last N records from the data source. E.g., If the chosen value for “N” is 5, then it will select last 5 records 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 4 then, it will select randomly any 4 records from the data source. If the selected value for “N” is 4% then, it will select 4% 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 2 then, it will select 1st, 3rd, 5th records or 2nd, 4th, 6th records from the data source.

6.9.2. 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 new field 'Percentage of Rows.'
 - b. **Sample Size Limit**
 - iii. **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 'Run.'
- vii) Users will be redirected to the 'Console' tab.



- viii) While accessing the 'Result' tab, Users will be displayed a result view based on the selected Sampling Type.

6.9.3. Result View for the Available Sampling Methods

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

Component Console Summary Result Visualization Properties

General **Sampling Information**

Properties

Sampling Type: First N

Limit Rows by: Number of Rows

Number of Rows: 1

Sample Size Limit

Maximum Rows: optional

Apply

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname
8	8	307	130	3504	12	70	1	chevrolet chevelle malibu

Showing 1 to 1 of 1 entries Previous 1 Next

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

Component Console Summary Result Visualization Properties

General **Sampling Information**

Properties

Sampling Type: First N

Limit Rows by: Percentage of Rows

Percentage of Rows: 5

Sample Size Limit

Maximum Rows: 6

Apply

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname
27	4	151	90	2950	17.3	82	1	chevrolet camaro
27	4	140	86	2790	15.6	82	1	ford mustang gl
44	4	97	52	2130	24.6	82	2	vw pickup
32	4	135	84	2295	11.6	82	1	dodge rampage
28	4	120	79	2625	18.6	82	1	ford ranger
31	4	119	82	2720	19.4	82	1	chevy s-10

Showing 1 to 6 of 6 entries Previous 1 Next

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

Component Console Summary **Result** Visualization Properties

General **Sampling Information**

Properties

Sampling Type: Every Nth

Step Size: 3

Sample Size Limit

Maximum Rows: 7

Apply

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname
18	8	318	150	3436	11	70	1	plymouth satellite
15	8	429	198	4341	10	70	1	ford galaxie 500
14	8	455	225	4425	10	70	1	pontiac catalina
14	8	340	160	3609	8	70	1	plymouth 'cuda 340
24	4	113	95	2372	15	70	3	toyota corona mark ii
21	6	200	85	2587	16	70	1	ford maverick
25	4	110	87	2672	17.5	70	2	peugeot 504

Showing 1 to 7 of 7 entries Previous 1 Next

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

Component Console Summary **Result** Visualization Properties

General **Sampling Information**

Properties

Sampling Type: Simple Random

Limit Rows by: Number of Rows

Number of Rows: 3

Sample Size Limit

Maximum Rows: optional

Apply

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname
32	4	71	65	1836	21	74	3	toyota corolla 1200
17.7	6	231	165	3445	13.4	78	1	buick regal sport coupe (turbo)
32	4	135	84	2295	11.6	82	1	dodge rampage

Showing 1 to 3 of 3 entries Previous 1 Next

5. Systematic Random (Bucket Size is 3).

Component Console Summary **Result** Visualization Properties

General **Sampling Information**

Properties

Sampling Type: Systematic Random

Bucket Size: 3

Sample Size Limit

Maximum Rows: optional

Apply

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname
32	4	71	65	1836	21	74	3	toyota corolla 1200
17.7	6	231	165	3445	13.4	78	1	buick regal sport coupe (turbo)
32	4	135	84	2295	11.6	82	1	dodge rampage

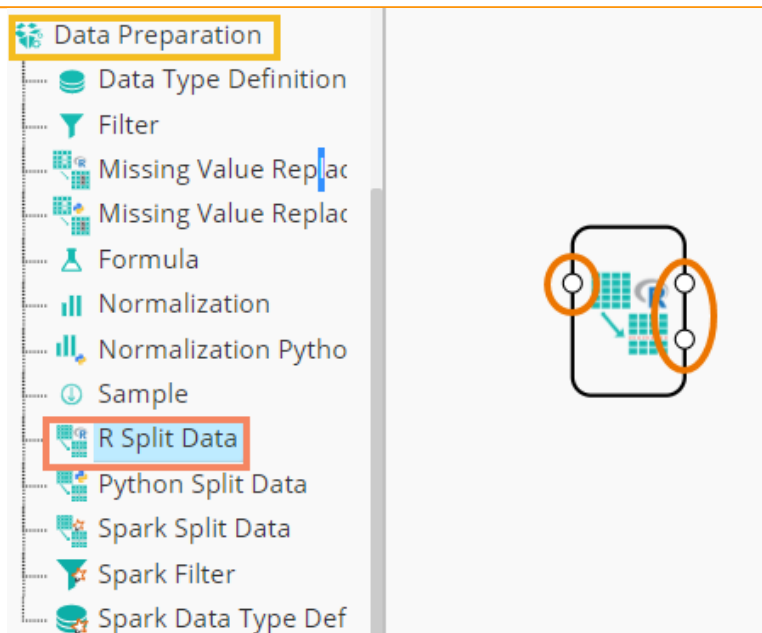
Showing 1 to 3 of 3 entries Previous 1 Next

6.10. R Split Data

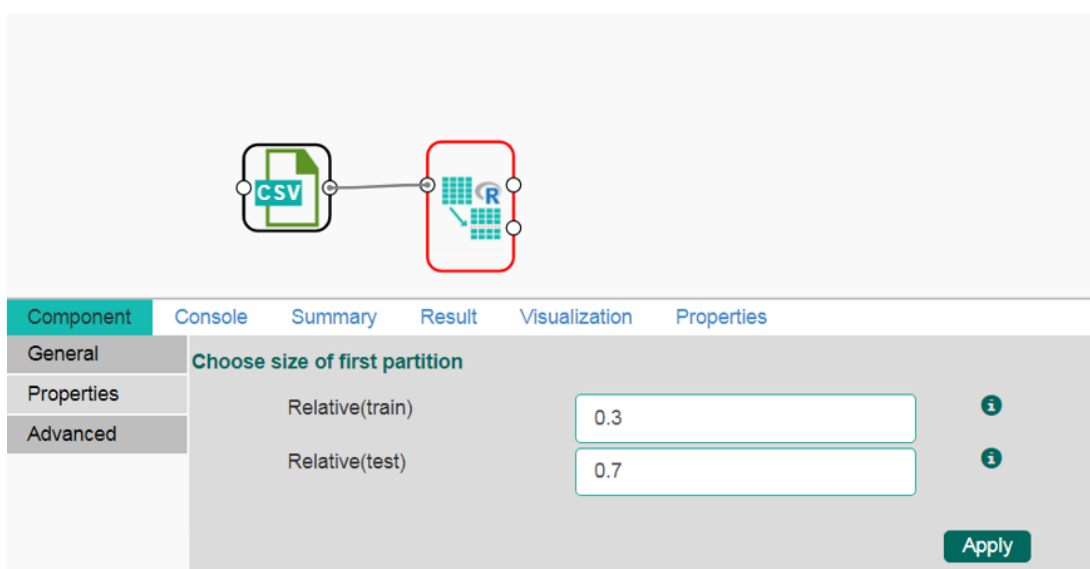
The R Split Data component is used to split a dataset into training and testing per percentage and method. Once the most suitable model is decided from the trained data, users can pass test data to validate the model.

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 lower node for the **testing data set**.



- i) Select the 'R Split Data' component and connect it with a valid data source (in this case, select Cassandra reader).
- 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) 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) Click 'Apply'



- vi) Click 'Run'
- vii) Users will be directed to the 'Console' tab.

R Split Data **Run** Reset

Component **Console** Summary Result Visualization Properties

```

7/7/2017 - 17:49:8 : Process Initiated...
7/7/2017 - 17:49:14 : csv0 is started.
7/7/2017 - 17:49:14 : csv0 is completed.
7/7/2017 - 17:49:14 : R Split Data1 is started.
7/7/2017 - 17:49:15 : R Split Data1 is completed.
    
```

- 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.

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:

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

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

Split 1 **Split 2**

Show 10 entries Search:

SepalLength	SepalWidth	PetalLength	PetalWidth	Species
4.8	3	1.4	0.3	setosa
5.1	3.8	1.6	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	3.7	1.5	0.2	setosa
5	3.3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor

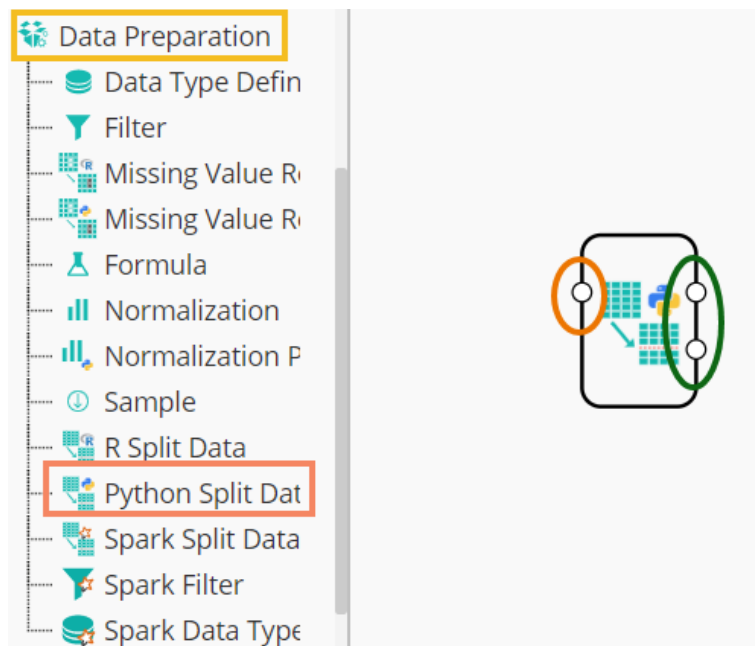
Showing 1 to 10 of 105 entries Previous **1** 2 3 4 5 ... 11 Next

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.

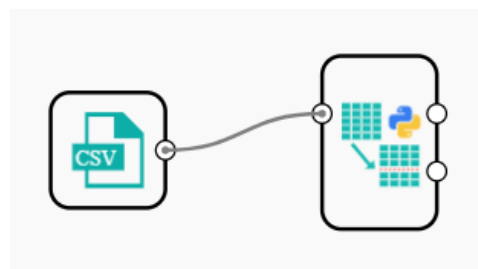
6.11. Python Split Data

Python Split Data component is used to split data into training and testing datasets. Once the user finds 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 '**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).
- v) Click '**Apply**'

Component Console Summary Result Visualization Properties

General Choose size of first partition

Properties

Advanced

Relative(train) 0.7

Relative(test) 0.3

Apply

- ix) Click 'Run'
- x) Users will be directed to the 'Console' tab.

Component Console Summary Result Visualization Properties

2/4/2018 - 17:33:20 : Process Initiated...

2/4/2018 - 17:33:22 : CSV1 is started.

2/4/2018 - 17:33:22 : CSV1 is completed.

2/4/2018 - 17:33:22 : Python Split Data1 is started.

2/4/2018 - 17:33:22 : Python Split Data1 is completed.

- xi) Follow the below given steps to display the result view:
 - c. Click the dragged algorithm component in the workspace.
 - d. 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:

RowID	SLength	SWidth	PLength	PWidth
42	4.5	2.3	1.3	0.3
29	5.2	3.4	1.4	0.2
2	4.9	3	1.4	0.2
48	4.6	3.2	1.4	0.2
116	6.4	3.2	5.3	2.3
85	5.4	3	4.5	1.5
36	5	3.2	1.2	0.2
112	6.4	2.7	5.3	1.9
119	7.7	2.6	6.9	2.3
88	6.3	2.3	4.4	1.3

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:

PLength	PWidth	SWidth	RowID	SLength
5.9	2.3	3.2	144	6.8
6.4	2	3.8	132	7.9
1.4	0.3	3	46	4.8
4.1	1.3	2.8	100	5.7
4.6	1.4	3	92	6.1
5.1	2.4	2.8	115	5.8
3.9	1.4	2.7	60	5.2
1.3	0.4	3.9	17	5.4
4.9	1.5	2.5	73	6.3
5.1	2	3.2	111	6.5

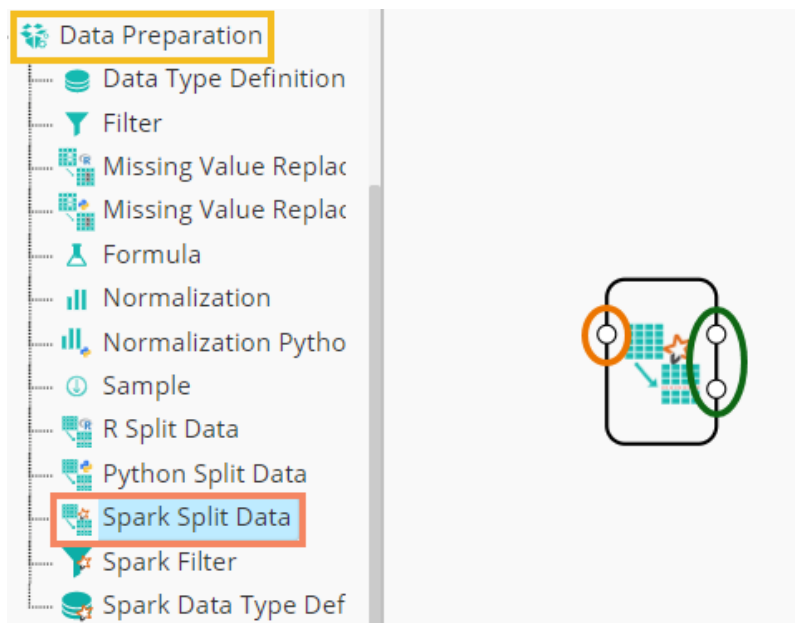
Showing 1 to 10 of 45 entries Previous 1 2 3 4 5 Next

6.12. 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**.

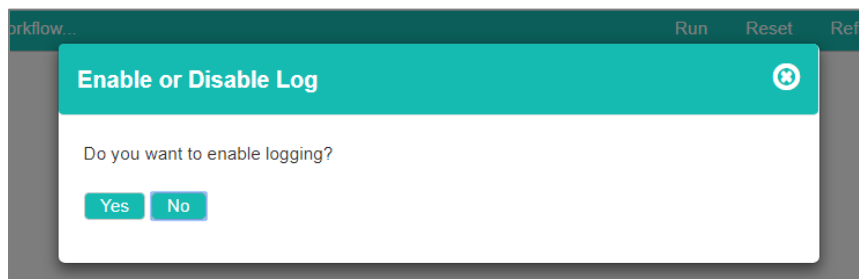


- i) Select the '**Spark Split Data**' component and connect it to a valid data source (in this case, select Cassandra reader).
- ii) Click the '**Spark Split Data**' component in the workspace.
- iii) Users will be directed to the Properties fields provided under the '**Components**' tab
- iv) Configure 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'

The screenshot shows a web interface with a sidebar on the left containing 'Component', 'Console', 'Summary', 'Result', 'Visualization', and 'Properties'. The 'Properties' tab is selected. Under the 'Properties' section, there is a sub-section titled 'Choose size of first partition'. It contains three input fields: 'Relative(train)' with the value '0.3', 'Relative(test)' with the value '0.7', and 'Seeds' with the value '1'. Each input field has a small information icon to its right. At the bottom right of this section is a green 'Apply' button.

- vi) Click 'Run'
- vii) A message will pop-up to confirm whether users want to enable logging.
- viii) Click 'No'



- ix) Users will be directed to the 'Console' tab.

Spark Split Data Run Reset

Component **Console** Summary Result Visualization Properties

```

7/7/2017 - 19:17:17 : Process Initiated...
7/7/2017 - 19:17:19 : Number of Rows fetched : 1000
7/7/2017 - 19:17:19 : cassandra0 Completed
7/7/2017 - 19:17:19 : Spark Split Data1 Running
7/7/2017 - 19:17:19 : Spark Split Data1 Completed
7/7/2017 - 19:17:19 : 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

Split 1 Split 2

Show 10 entries Search:

ROWNUM	C1	C2	C3	C4	C5	CLASS	S1	S2	S3	S4	S5
12	2	2	4	2	4	0	6	1	13	4	9
32	4	2	3	2	3	0	2	7	13	11	10
53	4	3	4	3	2	0	2	4	7	9	1
91	2	3	3	1	3	1	1	12	8	2	1
120	4	2	3	2	1	1	2	7	13	12	7
196	2	2	4	2	2	1	4	7	4	6	8
309	4	1	1	3	3	1	1	13	7	13	11
363	1	2	2	3	4	1	4	9	5	5	7
516	4	2	4	2	2	0	6	3	13	10	7
547	2	2	4	1	2	2	13	3	13	3	1

Showing 1 to 10 of 330 entries Previous 1 2 3 4 5 ... 33 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:

ROWNUM	C1	C2	C3	C4	C5	CLASS	S1	S2	S3	S4	S5
24	2	3	4	1	4	1	1	13	6	9	1
77	3	4	2	1	3	1	2	10	6	12	10
90	1	3	4	1	2	0	9	3	4	10	1
110	3	1	3	4	2	0	12	1	13	11	7
128	1	1	1	4	4	0	4	3	12	8	13
136	3	4	3	2	1	0	6	2	10	7	8
164	1	1	4	2	4	1	4	6	13	13	3
239	3	3	1	2	2	1	1	10	4	10	3
371	2	4	4	4	2	0	10	6	11	5	12
393	4	3	3	2	2	1	1	9	2	11	9

Showing 1 to 10 of 670 entries Previous 1 2 3 4 5 ... 67 Next

6.13. 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) Click ‘Run’ and check ‘Result’ for the data source.

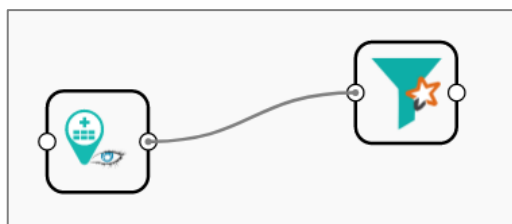
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

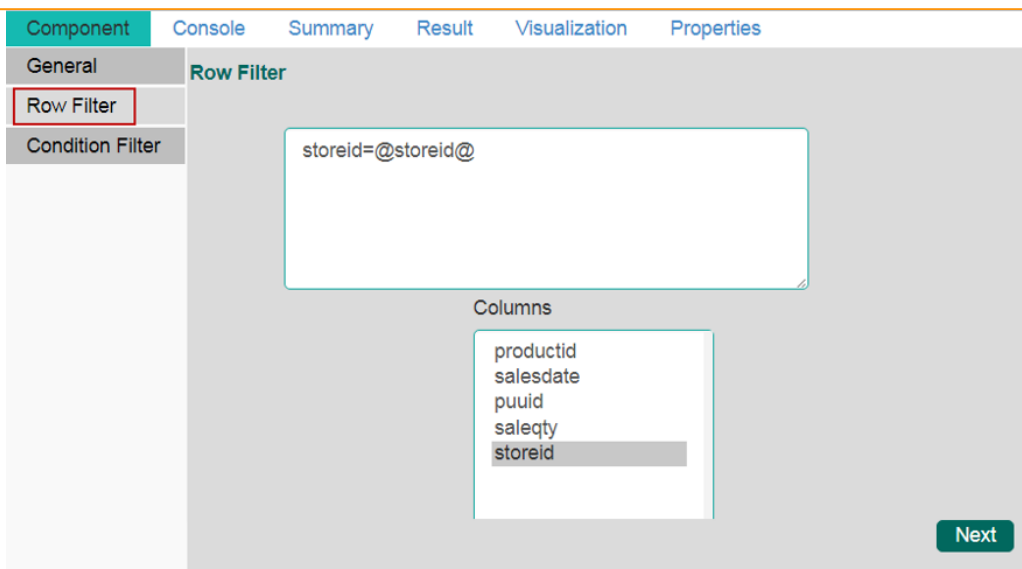
productid	salesdate	puuid	saleqty	storeid
colgate	1483209000000	32991ec1-e6ba-11e6-9917-7f22c2f70eae	111	100
colgate	1483209000000	7db92527-e6ba-11e6-9917-7f22c2f70eae	109	100
colgate	1483209000000	7db94c2c-e6ba-11e6-9917-7f22c2f70eae	101	200
colgate	1483209000000	7db94c4f-e6ba-11e6-9917-7f22c2f70eae	102	200
colgate	1485887400000	32991eb5-e6ba-11e6-9917-7f22c2f70eae	102	200
colgate	1485887400000	7db92517-e6ba-11e6-9917-7f22c2f70eae	101	200
colgate	1485887400000	7db92532-e6ba-11e6-9917-7f22c2f70eae	105	200
colgate	1485887400000	7db94c52-e6ba-11e6-9917-7f22c2f70eae	107	200
colgate	1488306600000	32991eb6-e6ba-11e6-9917-7f22c2f70eae	112	100
colgate	1488306600000	32991ebf-e6ba-11e6-9917-7f22c2f70eae	111	200

Showing 1 to 10 of 200 entries Previous 1 2 3 4 5 ... 20 Next

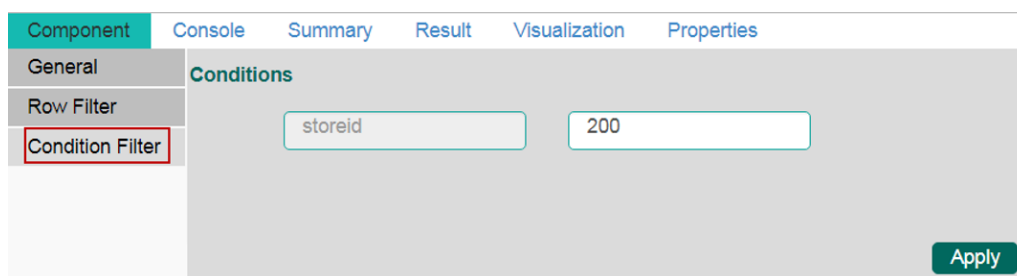
- iii) Drag the ‘Spark Filter’ component onto the workspace.
- iv) Connect it to the configured data source.



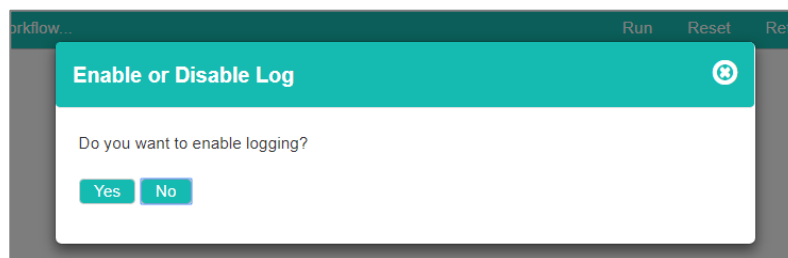
- v) Right-click on the Spark Filter component.
- vi) Provide condition for the ‘Row Filter’
- vii) Click ‘Next’



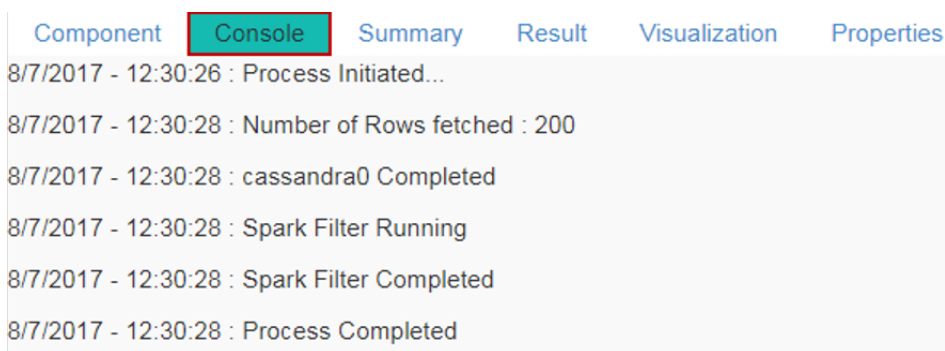
- viii) Users will be directed to configure a condition for the 'Column Filter'
- ix) Click 'Apply' after configuration.



- x) Click 'Run'
- xi) A message will pop-up to confirm whether users want to enable logging
- xii) Click 'No'



- xiii) Users will be directed to the 'Console' tab.



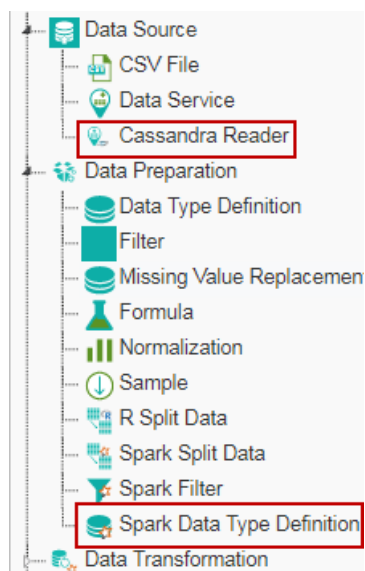
- xiv) Follow the below given steps to display the result view:
 - a. Click the dragged algorithm component onto the workspace.
 - b. Click the 'Result' tab.
- xv) The filtered result data will be displayed.

productid	salesdate	puuid	saleqty	storeid
colgate	1483209000000	7db94c2c-e6ba-11e6-9917-7f22c2f70eae	101	200
colgate	1483209000000	7db94c4f-e6ba-11e6-9917-7f22c2f70eae	102	200
colgate	1485887400000	32991eb5-e6ba-11e6-9917-7f22c2f70eae	102	200
colgate	1485887400000	7db92517-e6ba-11e6-9917-7f22c2f70eae	101	200
colgate	1485887400000	7db92532-e6ba-11e6-9917-7f22c2f70eae	105	200
colgate	1485887400000	7db94c52-e6ba-11e6-9917-7f22c2f70eae	107	200
colgate	1488306600000	32991ebf-e6ba-11e6-9917-7f22c2f70eae	111	200
colgate	1488306600000	329945bf-e6ba-11e6-9917-7f22c2f70eae	107	200
colgate	1488306600000	329945dc-e6ba-11e6-9917-7f22c2f70eae	104	200
colgate	1488306600000	7db94c29-e6ba-11e6-9917-7f22c2f70eae	103	200

6.14. Spark Data Type Definition

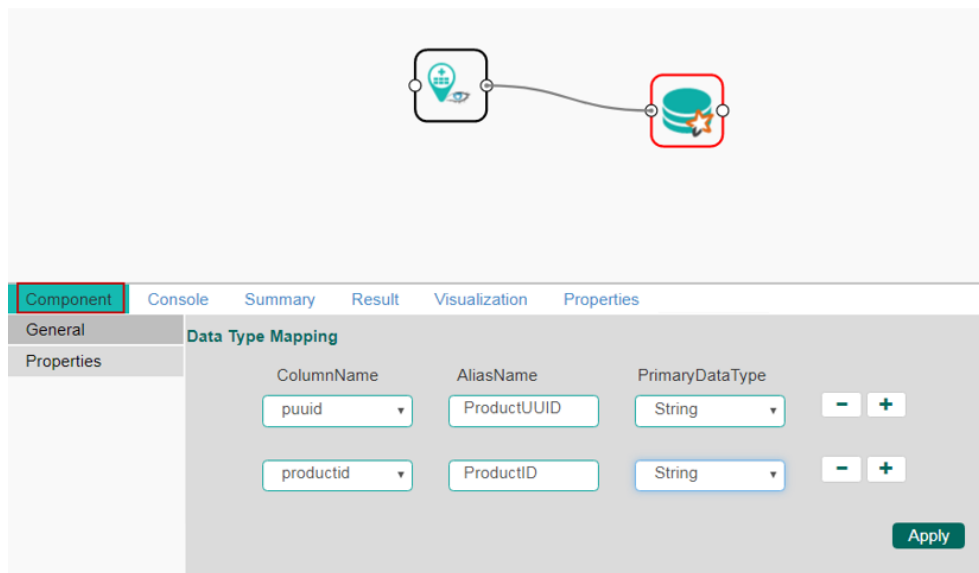
This component can be used to typecast data into another form. Users can change the data type of a column or change the alias name of the column using this component. Spark Data Type definition will appear as a leaf node under the Data Preparation tree node.

- 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).

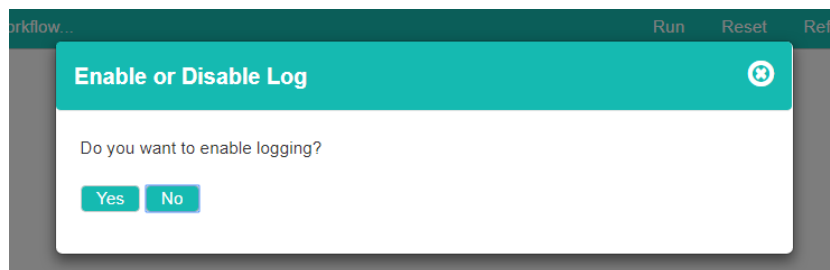


- ii) Configure the Properties fields for the Spark Data Type Definition component.
- iii) Configure the following 'Data Type Transformation' details:
 - a. **Column Name:** Select a column name which you want to change
 - b. **Alias Name:** Enter an alias name for the required source column
 - c. **Primary 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

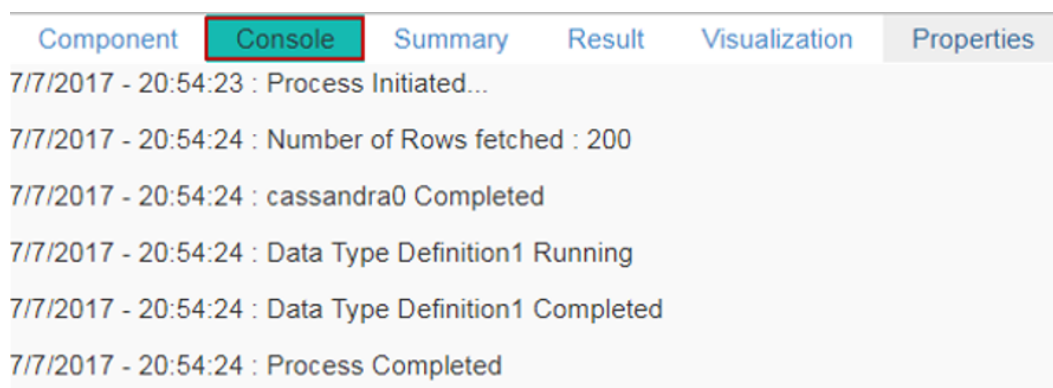
- transformed.
- iv) Click **'Apply'**



- v) Click **'Run'**
- vi) A message will pop-up to confirm whether users want to enable logging.
- vii) Click **'No'**



- viii) Users will be directed to the **'Console'** tab.



- ix) Follow the below given steps to display the result view:
- Click the data preparation component onto the workspace.
 - Click the **'Result'** tab.

Component	Console	Summary	Result	Visualization	Properties
Show 10 entries					
Search: <input type="text"/>					
salesdate	saleqty	storeid	ProductUUID		
1483209000000	111	100	32991ec1-e6ba-11e6-9917-7f22c2f70eae		
1483209000000	109	100	7db92527-e6ba-11e6-9917-7f22c2f70eae		
1483209000000	101	200	7db94c2c-e6ba-11e6-9917-7f22c2f70eae		
1483209000000	102	200	7db94c4f-e6ba-11e6-9917-7f22c2f70eae		
1485887400000	102	200	32991eb5-e6ba-11e6-9917-7f22c2f70eae		
1485887400000	101	200	7db92517-e6ba-11e6-9917-7f22c2f70eae		
1485887400000	105	200	7db92532-e6ba-11e6-9917-7f22c2f70eae		
1485887400000	107	200	7db94c52-e6ba-11e6-9917-7f22c2f70eae		
1488306600000	112	100	32991eb6-e6ba-11e6-9917-7f22c2f70eae		
1488306600000	111	200	32991ebf-e6ba-11e6-9917-7f22c2f70eae		
Showing 1 to 10 of 200 entries					
Previous 1 2 3 4 5 ... 20 Next					

Note:

- Users cannot typecast the advanced column types (E.g., map, list, UDT), UUID, and timestamp.
- Spark Data Type Definition supports only Integer, Double, and String data types.
- Users need to click the Spark component and then click the 'Result' tab to display result view for any Spark Component.
- Spark Data Preparation components supports only Cassandra reader.

7. 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:

- The Data Transformation components can be connected to only those Data Preparation components that have 'Spark' prefix in their names.
- A 'Data Preparation' component cannot be added in between the 'Data Transformation' and 'Apply Model' components in a workflow.
- All the 'Data Transformation' components are pipeline components. Results can be viewed only after connecting them to an 'Apply Model' component.
- End of the pipeline component should be an 'Apply Model' component.
- A model can be saved from the context menu of an 'Apply Model' component.

7.1. 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.

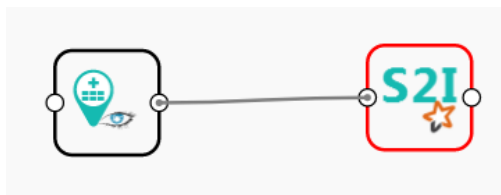
The Spark String Indexer will come as a leaf node under Data Preparation. Component consist of one node for input data and another for output data.

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 label column from previous component headers. After

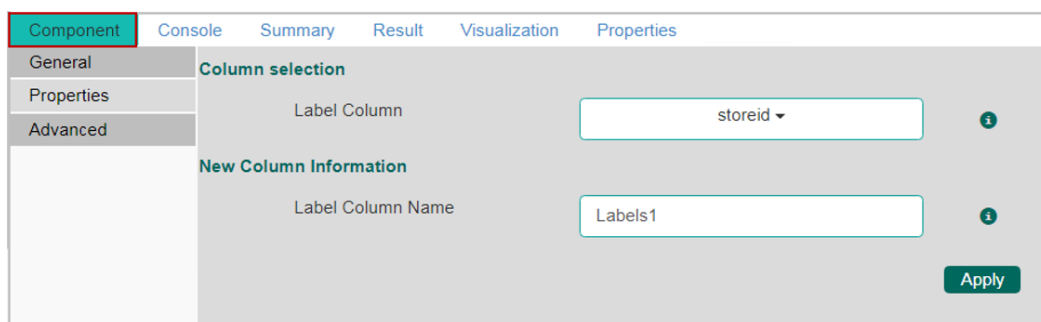
choosing a label, column user can change column header of the newly indexed column which is Label by default.

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.

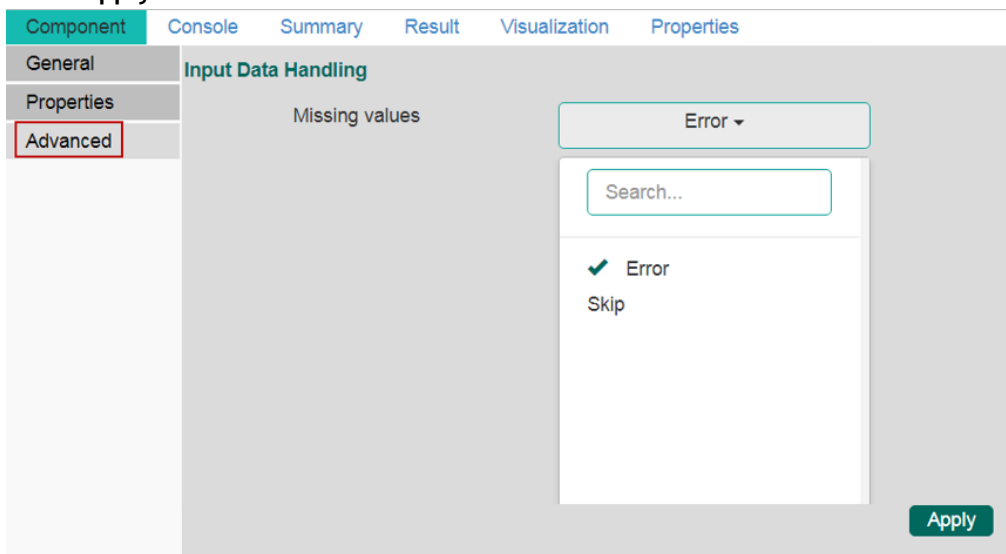
- i) Users need to select the String Indexer component and connect it with a configured data source.



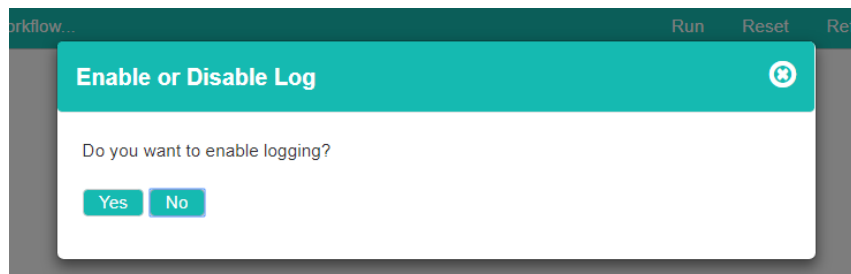
- ii) Configure the required component fields for the String Indexer.
 - a. The Properties tab for Spark Indexer contains an option to select 'Label Column' from previous component headers on which a new column was created.
 - b. Users can rename the created label column using the 'Label Column Name'



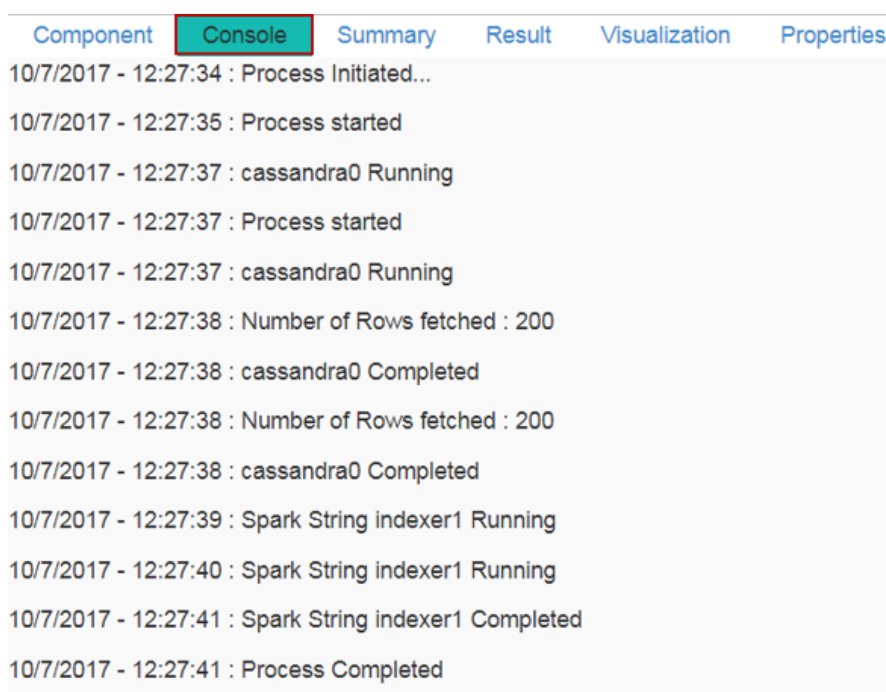
- c. The String Indexer, when applied on one dataset, will handle unseen labels using either of the methods provided under the 'Advanced' tab:
 - d. Users are provided with two options in the 'Advanced' tab to manage the unseen labels.
 - i. Error: The unseen labels will be thrown as an exception. (by default)
 - ii. Skip: The rows containing the unobserved labels will be skipped.
- iii) Click 'Apply'



- iv) Click 'Run'
- v) A message will pop-up to confirm whether users want to enable logging.
- vi) Click 'No'



- vii) Users will be directed to the 'Console' tab.



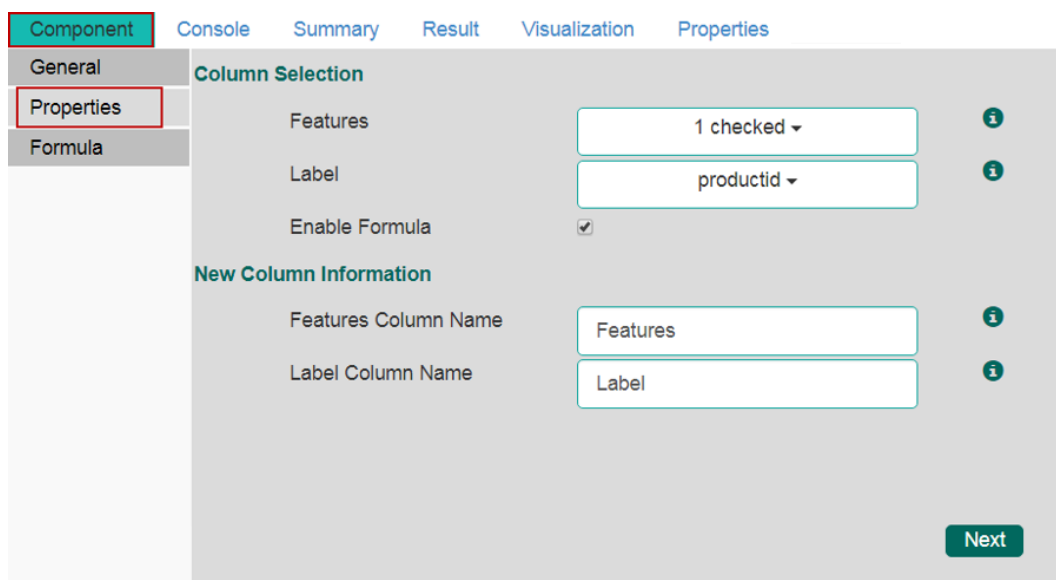
7.2. Spark R Formula

The Spark R Formula can be used to produce a vector column of features and a double column of labels.

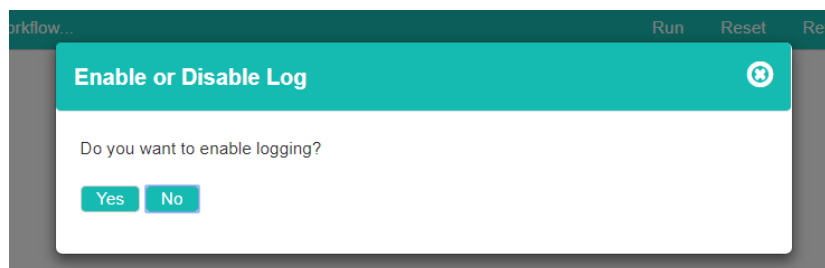
The Spark R Formula is a feature selector for the BDB Predictive Analysis which can be used to transform string columns to numerical columns. After selecting desired features and labels from previous columns.

- 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')
 - 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) Click 'Run'
- viii) A message will pop-up to confirm whether users want to enable logging.
- ix) Click 'No'



x) Users will be directed to the 'Console' tab.

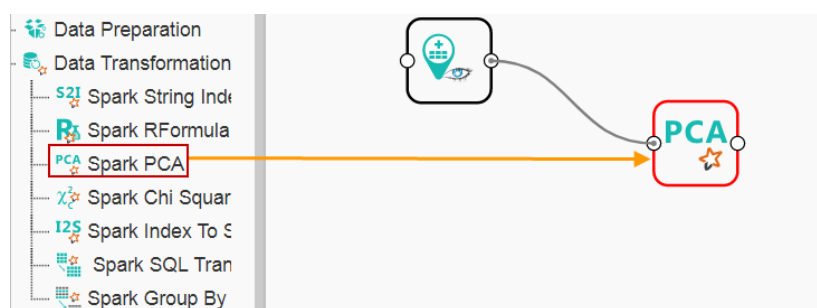
Component	Console	Summary	Result	Visualization	Properties
10/7/2017 - 15:13:16 : Process Initiated...					
10/7/2017 - 15:13:17 : Process started					
10/7/2017 - 15:13:17 : cassandra0 Running					
10/7/2017 - 15:13:19 : Number of Rows fetched : 200					
10/7/2017 - 15:13:19 : cassandra0 Completed					
10/7/2017 - 15:13:19 : Spark RFormula1 Running					
10/7/2017 - 15:13:19 : Spark RFormula1 Completed					
10/7/2017 - 15:13:19 : Process Completed					

7.3. 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.

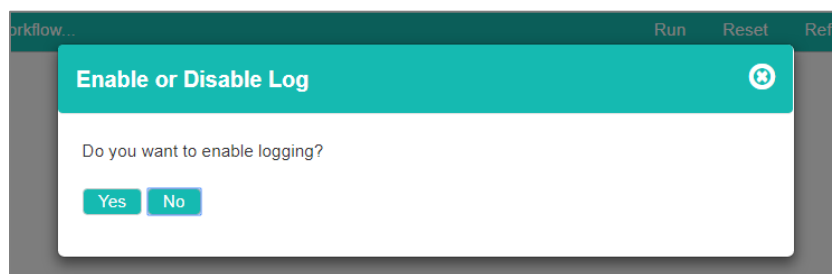
- i) Users need to select the Spark PCA component and connect it to a configured data source.



- 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'

Component	Console	Summary	Result	Visualization	Properties
General					
Properties	<p>Input Column</p> <p>Features: 1 checked ▾ i</p> <p>K Value: 1 i</p> <p>Output Column</p> <p>Predicted Column Name: OutputCol i</p> <p style="text-align: right;">Apply</p>				

- iv) Click 'Run'
- v) A message will pop-up to confirm whether users want to enable logging.
- vi) Click 'No'



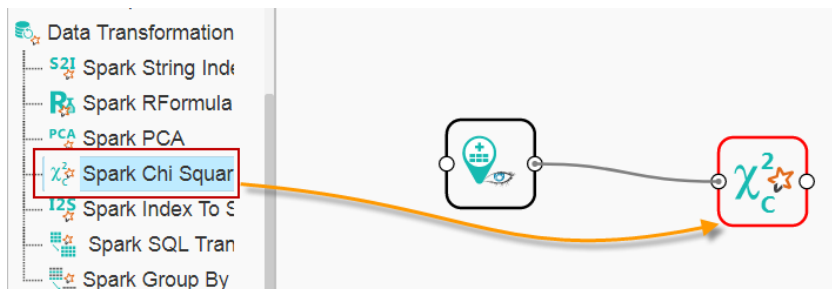
- vii) Users will be directed to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
<pre> 10/7/2017 - 18:2:10 : Process started 10/7/2017 - 18:2:10 : cassandra0 Running 10/7/2017 - 18:2:11 : Number of Rows fetched : 1000 10/7/2017 - 18:2:11 : cassandra0 Completed 10/7/2017 - 18:2:11 : Spark PCA1 Running 10/7/2017 - 18:2:12 : Spark PCA1 Completed 10/7/2017 - 18:2:12 : Process Completed </pre>					

7.4. Spark Chi-Square

In probability theory and statistics, the chi-squared distribution (also chi-square or χ^2 -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.



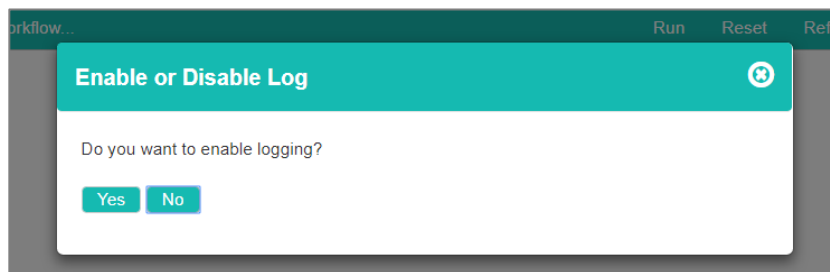
- ii) Configure the following component fields for the Spark Chi-Square:
 - a. Input Column
 - i. Features: Select the required elements from the drop-down menu.
 - ii. K Value: Enter the number of principal components.
 - b. Output Column
 - i. Predicted Column Name: Enter column header for the predicted column.
- iii) Click 'Apply'

The screenshot shows the configuration interface for the Spark Chi-Square component. The 'Properties' tab is selected in the left sidebar. The configuration fields are as follows:

- Input Column**
 - Feature: 1 checked (dropdown menu)
 - Label: C1 (dropdown menu)
- Column Selection**
 - Selector Type: Num of Top Features (dropdown menu)
 - Number of top feature: 50 (text input)
- Output Column**
 - Predicted Column Name: OutputCol (text input)

An 'Apply' button is located at the bottom right of the configuration area.

- iv) Click 'Run'
- v) A message will pop-up to confirm whether users want to enable logging.
- vi) Click 'No'



vii) Users will be directed to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
10/7/2017 - 18:12:37 : Process Initiated...
10/7/2017 - 18:12:37 : Process started
10/7/2017 - 18:12:37 : cassandra0 Running
10/7/2017 - 18:12:38 : Number of Rows fetched : 1000
10/7/2017 - 18:12:38 : cassandra0 Completed
10/7/2017 - 18:12:39 : Spark Chi Square1 Running
10/7/2017 - 18:12:39 : Spark Chi Square1 Completed
10/7/2017 - 18:12:39 : Process Completed

```

7.5. 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 label column from previous component headers. After choosing a label, column user can change 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 String Indexer component with the data source and configure it. (Ref. section 7.1)

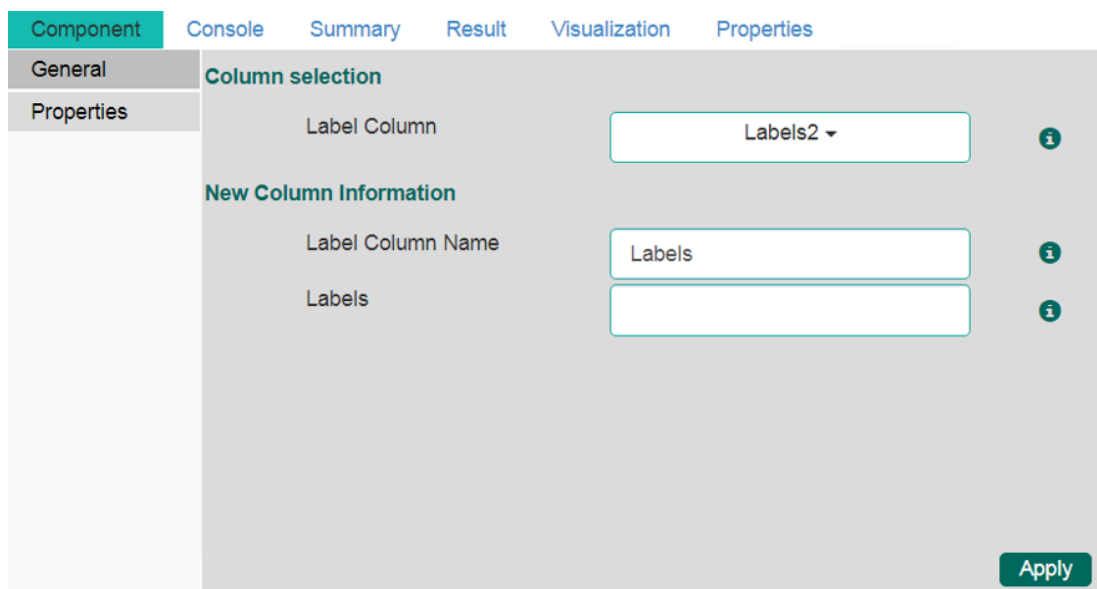
The first screenshot shows the configuration for the 'Column selection' section. The 'Label Column' is set to 'PetalLength' and the 'Label Column Name' is set to 'Labels2'. An 'Apply' button is visible at the bottom right.

The second screenshot shows the configuration for the 'Input Data Handling' section. The 'Missing values' are set to 'Error'. An 'Apply' button is visible at the bottom right.

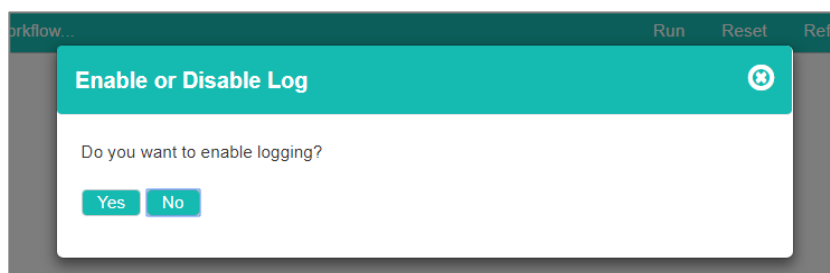
- iii) Connect the Spark Index to String component with the Spark String Indexer component on the workspace.



- 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:
- v) Click ‘Apply’



- vi) Click ‘Run’
- vii) A message will pop-up to confirm whether users want to enable logging.
- viii) Click ‘No’



- ix) Users will be directed to the ‘Console’ tab.

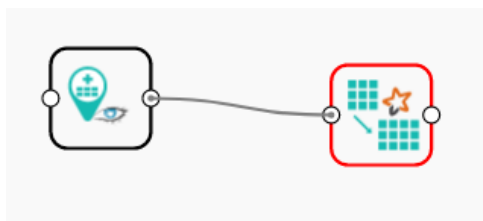
Component	Console	Summary	Result	Visualization	Properties
11/7/2017 - 15:22:57 : Process Initiated...					
11/7/2017 - 15:22:59 : Number of Rows fetched : 68					
11/7/2017 - 15:22:59 : cassandra0 Completed					
11/7/2017 - 15:22:59 : Spark String indexer2 Running					
11/7/2017 - 15:22:59 : Spark String indexer2 Completed					
11/7/2017 - 15:22:59 : Spark Index To String1 Running					
11/7/2017 - 15:22:59 : Spark Index To String1 Completed					
11/7/2017 - 15:22:59 : Process Completed					

Note: Users need to first connect the data source with the ‘String Indexer’ component, and then the combination can be connected to the ‘Index to String’ component.

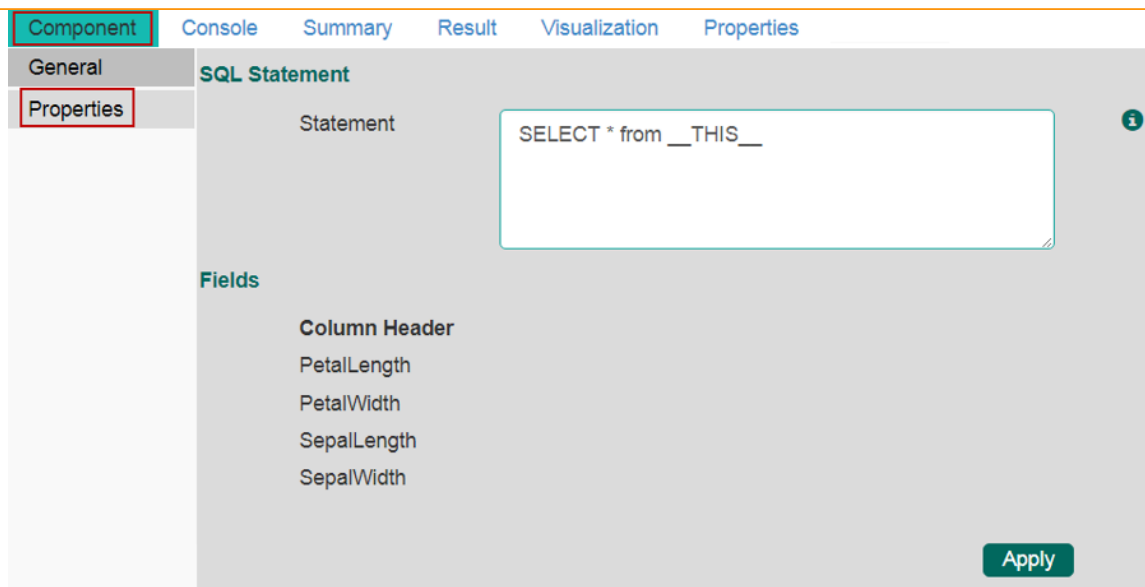
7.6. Spark SQL Transformer

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.

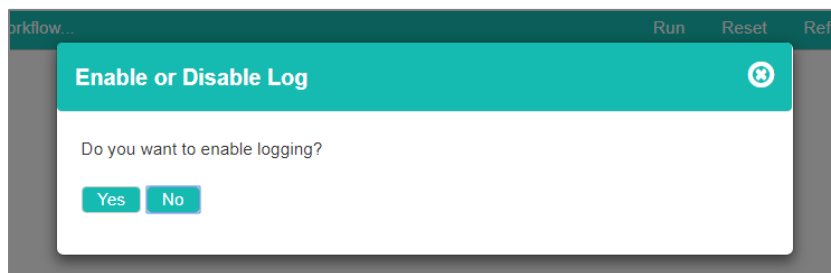
- i) Select the Spark SQL Transformer component and connect it to a configured data source.



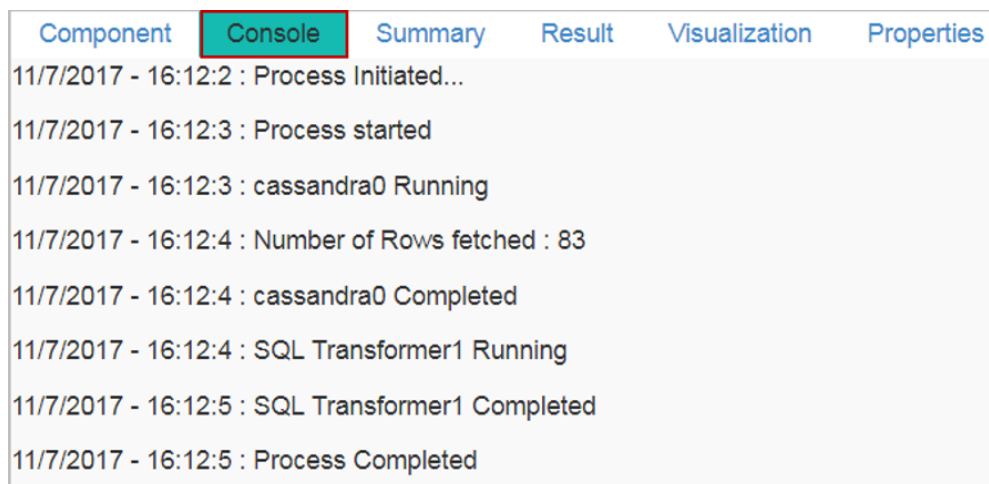
- ii) Configure the required component fields for the Spark SQL Transformer.
 - a. SQL Statement: Provide an SQL statement.
 - b. Fields: All the available fields under the selected data source will be listed.
- iii) Click ‘Apply’.



- iv) Click 'Run'
- v) A message will pop-up to confirm whether users want to enable logging.
- vi) Click 'No'



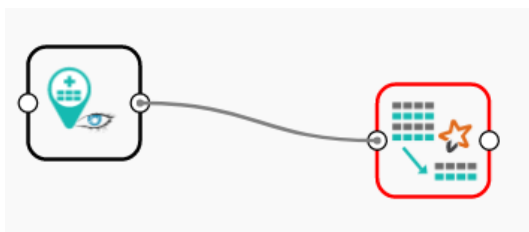
- vii) Users will be directed to the 'Console' tab.



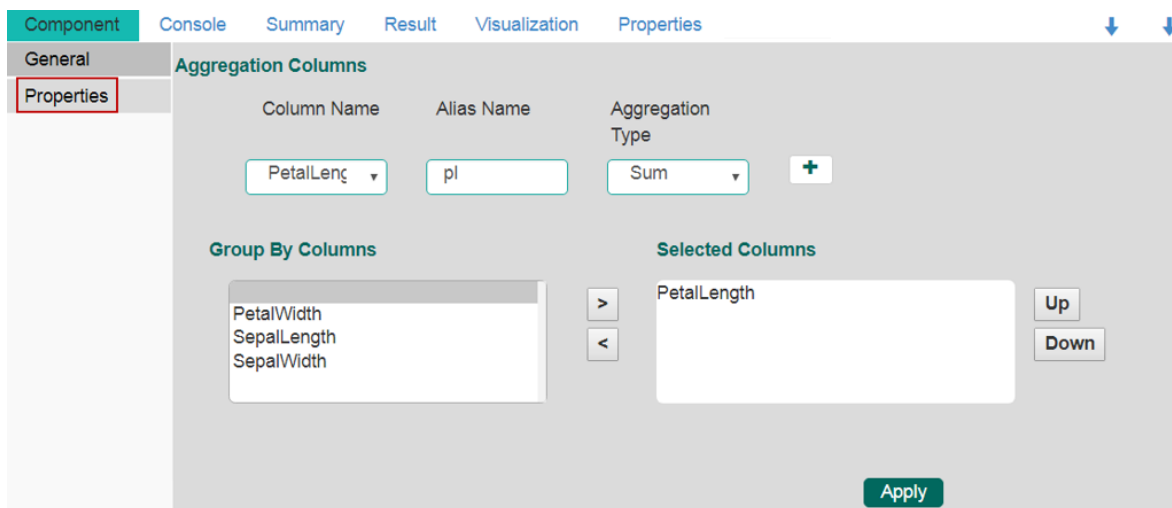
7.7. Spark Group By

Spark Group By is a transformation operation. Users can apply 'Spark Group By' transformation on 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.

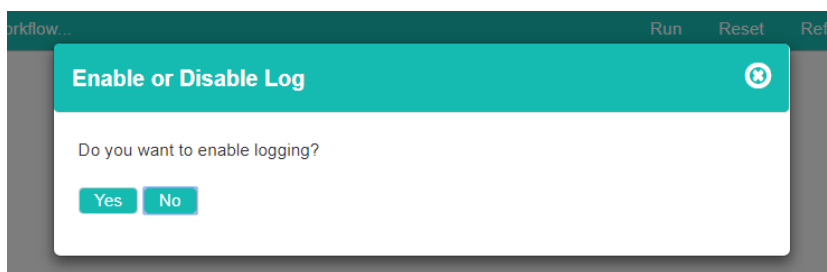
- i) Select the Spark Group By component and connect it to a configured data source.



- 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) Click 'Run'
- v) A message will pop-up to confirm whether users want to enable logging.
- vi) Click 'No'




- vii) Users will be directed to the 'Console' tab.

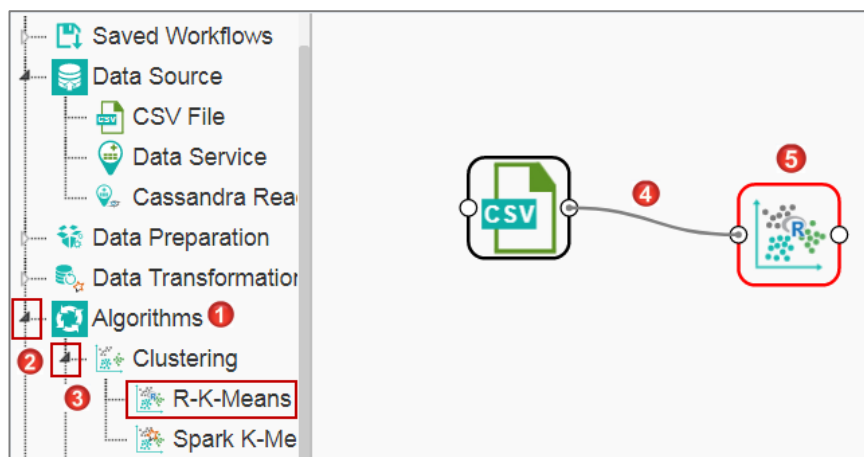
Component	Console	Summary	Result	Visualization	Properties
11/7/2017 - 16:28:47 : Process Initiated...					
11/7/2017 - 16:28:49 : Number of Rows fetched : 67					
11/7/2017 - 16:28:49 : cassandra0 Completed					
11/7/2017 - 16:28:49 : Spark Group By Running					
11/7/2017 - 16:28:49 : Spark Group By Completed					
11/7/2017 - 16:28:49 : Process Completed					

8. Algorithms

Algorithms are a statistical set of rules that help the user analyze vast quantities of numerical data and extract appropriate information out of it. BDB Predictive Analysis allows the user 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 'Components' fields for the dragged algorithm component.
- vii) Click 'Apply' to save the information.

Component Console Summary Result Visualization Properties

General **Output Information**

Number Of Clusters ⓘ

Column Selection

Features ⓘ

New Column Information

Cluster Name ⓘ

Apply

- viii) Click 'Run.'
- ix) Users will be directed to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

12/7/2017 - 13:12:0 : Process Initiated...

12/7/2017 - 13:12:1 : csv0 is started.

12/7/2017 - 13:12:1 : csv0 is completed.

12/7/2017 - 13:12:1 : R-K-Means1 is started.

12/7/2017 - 13:12:1 : R-K-Means1 is completed.

- x) Click the algorithm component on the workspace and click the 'Result' tab.
- xi) The resulting view will be displayed.

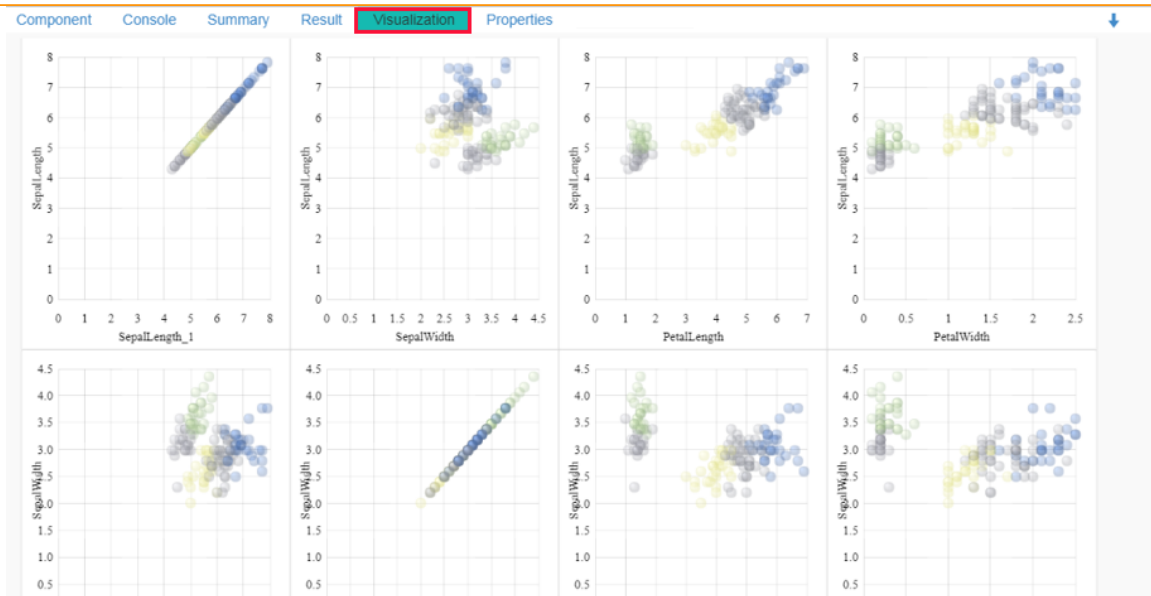
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

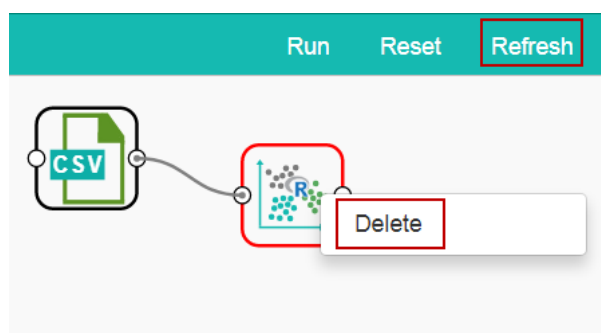
SepalLength	SepalWidth	PetalLength	PetalWidth	Species	ClusterNumber1
5.1	3.5	1.4	0.2	setosa	2
4.9	3	1.4	0.2	setosa	5
4.7	3.2	1.3	0.2	setosa	5
4.6	3.1	1.5	0.2	setosa	5
5	3.6	1.4	0.2	setosa	2
5.4	3.9	1.7	0.4	setosa	2
4.6	3.4	1.4	0.3	setosa	5
5	3.4	1.5	0.2	setosa	2
4.4	2.9	1.4	0.2	setosa	5
4.9	3.1	1.5	0.1	setosa	5

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

- 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:

- Users can follow the steps mentioned above to configure all the available R- algorithms.
- Users can configure alias name for the algorithm component via the ‘General’ tab.
- 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.
- Users can avail all the default values under ‘Advanced’ tab. Users can manually set the ‘Advanced’ tab, only if the advanced level configuration is required.
- 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.

8.1. 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).

8.1.1. 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.
 - 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).

Component	Console	Summary	Result	Visualization	Properties
General	Behavior				
Properties					
Advanced	<p>Maximum Iterations <input type="text" value="100"/></p> <p>Number of initial centroids <input type="text" value="1"/></p> <p>Algorithm Type <input type="text" value="1 checked ▾"/></p> <p>Initial Cluster Center Seed <input type="text" value="10"/></p> <p style="text-align: right;">Apply</p>				

- v) Click **'Apply'**
- vi) Click **'Run'**
- vii) Users will be redirected to the **'Console'** tab.

Component	Console	Summary	Result	Visualization	Properties
12/4/2018 - 19:38:45 : Process Initiated...					
12/4/2018 - 19:38:47 : CSV0 is started.					
12/4/2018 - 19:38:48 : CSV0 is completed.					
12/4/2018 - 19:38:48 : R-K-Means1 is started.					
12/4/2018 - 19:38:48 : 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	2
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	2
6	5.4	3.9	1.7	0.4	2
7	4.6	3.4	1.4	0.3	5
8	5	3.4	1.5	0.2	2
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.



8.1.2. 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.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Number Of Clusters		<input type="text" value="5"/>	<i>i</i>	
Advanced	Column Selection				
	Features		<input type="text" value="5 checked"/>	<i>i</i>	
	New Column Information				
	Cluster Name		<input type="text" value="ClusterNumber"/>	<i>i</i>	
<input type="button" value="Apply"/>					

- 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).

Component Console Summary Result Visualization Properties

General Behavior

Properties

Advanced

Maximum Iterations

Initialization Mode

Initialization Steps

Convergence tolerance

Initial Cluster Center Seed

Apply

- iv) Click 'Apply'
- v) Click 'Run' to run the execution.
- vi) Users will be directed to the 'Console' tab. A message will pop-up to confirm, whether users want to enable logging or no.
- vii) Click 'No'
- viii) Users will be directed to the 'Console' tab.

Component Console Summary Result Visualization Properties

12/7/2017 - 15:10:1 : Process Initiated...

12/7/2017 - 15:10:3 : Process started

12/7/2017 - 15:10:3 : Spark-K-Means1 Running

12/7/2017 - 15:10:4 : Spark-K-Means1 Completed

12/7/2017 - 15:10:4 : Process 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.
- x) A new column 'ClusterNumber' will be added to the displayed result data.

Component Console Summary Result Visualization Properties

Show 10 entries Search:

uide	Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	Species	featuresCol1	ClusterNumber
6367d610-2f09-11e7-be1c-d9db04d902f5	1.5	0.2	5.2	3.5	setosa	{"values": [1.5, 0.2, 5.2, 3.5]}	4
636c1bd0-2f09-11e7-be1c-d9db04d902f5	4.8	1.8	5.9	3.2	versicolor	{"values": [4.8, 1.8, 5.9, 3.2]}	0
636f2910-2f09-11e7-be1c-d9db04d902f5	6	2.5	6.3	3.3	virginica	{"values": [6.2, 5.6, 3.3, 3.3]}	3
6370fdd0-2f09-11e7-be1c-d9db04d902f5	6.9	2.3	7.7	2.6	virginica	{"values": [6.9, 2.3, 7.7, 2.6]}	3
63664770-2f09-11e7-be1c-d9db04d902f5	1.4	0.1	4.8	3	setosa	{"values": [1.4, 0.1, 4.8, 3]}	1
636a4710-2f09-11e7-be1c-d9db04d902f5	4.9	1.5	6.9	3.1	versicolor	{"values": [4.9, 1.5, 6.9, 3.1]}	0
63689960-2f09-11e7-be1c-d9db04d902f5	1.2	0.2	5	3.2	setosa	{"values": [1.2, 0.2, 5, 3.2]}	1
6368c070-2f09-11e7-be1c-d9db04d902f5	1.3	0.2	5.5	3.5	setosa	{"values": [1.3, 0.2, 5.5, 3.5]}	4
6365da40-2f09-11e7-be1c-d9db04d902f5	1.4	0.2	4.4	2.9	setosa	{"values": [1.4, 0.2, 4.4, 2.9]}	1
637088a0-2f09-11e7-be1c-d9db04d902f5	5	2	5.7	2.5	virginica	{"values": [5, 2, 5.7, 2.5]}	0

Showing 1 to 10 of 113 entries Previous 1 2 3 4 5 ... 12 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.

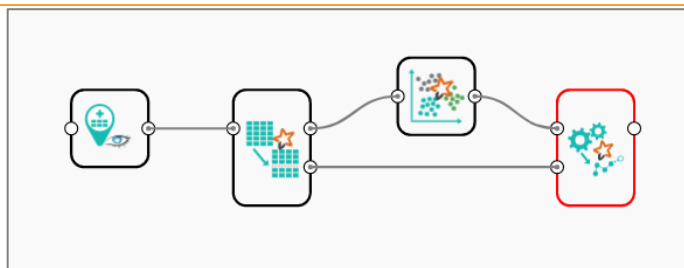
```

Component Console Summary Result Visualization Properties
----- Summary of the model -----
Columns used in the algorithm:
Age_in_Years (integer)
Credit_Amount (integer)
Loan_Duration (integer)
Number_of_existing_credits_at_this_bank (integer)
Number_of_people_maintenance (integer)
Present_residence_since (integer)
instalment_rate_in_percentage_of_disposable_income (integer)
Cluster Centers =
[34.85795454545455,4124.926136363636,25.335227272727273,1.4261363636363638,1.1931818181818181,2.9204545454545454,2.6363636363636362],
[36.80487804878049,12576.463414634147,40.36585365853659,1.3902439024390245,1.1219512195121952,2.975609756097561,2.3902439024390243],
[36.05361930294906,1165.9436997319035,13.324396782841823,1.4128686327077749,1.1528150134048258,2.798927613941019,3.3029490616621984],
[36.983870967741936,7243.096774193548,33.12096774193548,1.5,1.185483870967742,2.846774193548387,2.5161290322580645],
[34.50349650349651,2435.646853146853,19.972027972027973,1.3496503496503496,1.1258741258741258,2.839160839160839,3.0314685314685317]
Within Set Sum of Squared Errors = 4.726457049160745E8
----- End of Summary -----

```

8.1.2.1. 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 Click 'Run' option.
- iii) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

```

17/7/2017 - 20:11:44 : Process Initiated...
17/7/2017 - 20:11:49 : Number of Rows fetched : 43
17/7/2017 - 20:11:49 : cassandra0 Completed
17/7/2017 - 20:11:49 : Spark Split Data1 Running
17/7/2017 - 20:11:49 : Spark Split Data1 Completed
17/7/2017 - 20:11:49 : Spark-K-Means2 Running
17/7/2017 - 20:11:51 : Spark-K-Means2 Completed
17/7/2017 - 20:11:51 : Spark Apply Model3 Running
17/7/2017 - 20:11:52 : Spark Apply Model3 Completed
17/7/2017 - 20:11:52 : 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.

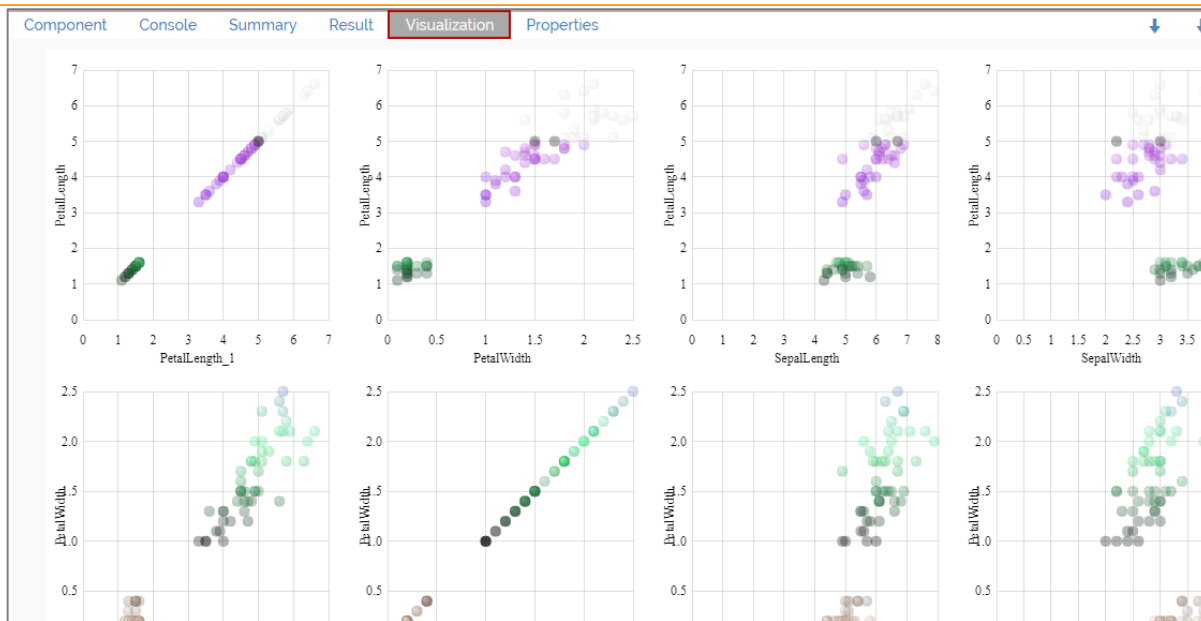
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	featuresCol2	ClusterNumber
3.3	1	5	2.3	versicolor	{"values": [3.3, 1.5, 2.3]}	2
1.4	0.2	5	3.3	setosa	{"values": [1.4, 0.2, 5, 3.3]}	0
5	2	5.7	2.5	virginica	{"values": [5, 2, 5.7, 2.5]}	1
5.1	2.4	5.8	2.8	virginica	{"values": [5.1, 2.4, 5.8, 2.8]}	1
6.1	2.5	7.2	3.6	virginica	{"values": [6.1, 2.5, 7.2, 3.6]}	3
1.5	0.2	4.6	3.1	setosa	{"values": [1.5, 0.2, 4.6, 3.1]}	0
1.4	0.2	5	3.6	setosa	{"values": [1.4, 0.2, 5, 3.6]}	0
4.9	1.8	6.1	3	virginica	{"values": [4.9, 1.8, 6.1, 3]}	1
5.9	2.3	6.8	3.2	virginica	{"values": [5.9, 2.3, 6.8, 3.2]}	3
1.3	0.3	4.5	2.3	setosa	{"values": [1.3, 0.3, 4.5, 2.3]}	0

Showing 1 to 10 of 11 entries Previous 1 2 Next

- v) Click the 'Visualization' tab to see the result data via the Scatter Plot Matrix chart.



8.2. Forecasting

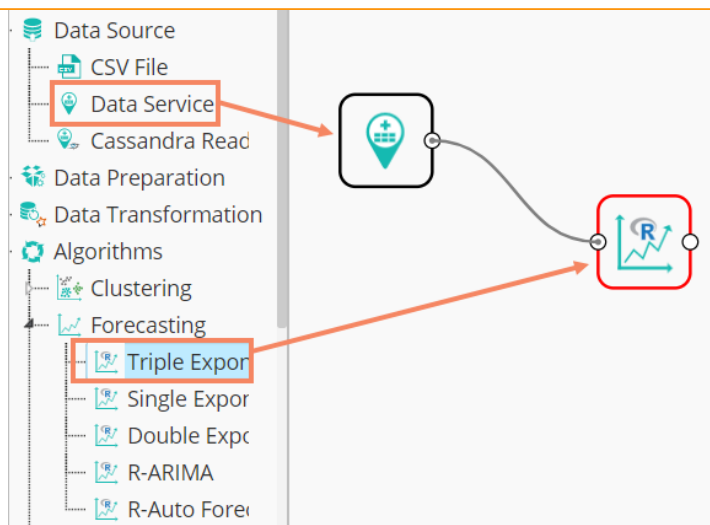
Forecasting is a method that used extensively in time series analysis to predict a response variable, such as monthly profits, stock performance, or unemployment figures, for a specified period. Forecasts are based on patterns in existing data. For example, a warehouse manager can model how much product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

Forecasting algorithms will come under the tree node of Algorithms. The following sub-categories of algorithms are included under-forecasting:

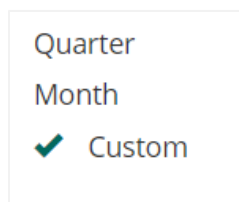
8.2.1. Triple Exponential Smoothing

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 periods, 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 basic 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
 - 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 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. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).

- ii. **Period Values:** Enter a name for the column containing period Value (This field will be Predefined, but users can change the value if needed). In this case, the selected Period option is ‘Custom’ hence, ‘Period Values’ field is displayed under the ‘New Column Information’.

Note:

- a. ‘New Column Information’ varies as per the selected ‘Period’ option from the ‘Input Data Handling.’ It displays the below-mentioned column names for the Period Value columns based on the selected ‘Period’ option from the ‘Input Data Handling’ section.

Selected ‘Period’ option	Displayed Period Value field under ‘New Column Information.’
Quarter	Quarter Values
Month	Month Values
Custom	Period Values

- b. The ‘Period Per Year’ field under the ‘Input Data Handling’ section is displayed only when ‘Custom’ is selected as an option for the ‘Period’ field.
- 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: $0 < \alpha \leq 1$.)
 - 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 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.
- 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.

Component	Console	Summary	Result	Visualization	Properties
General	Behavior				
Properties	Alpha <input type="text" value=".3"/>				
Advanced	Beta <input type="text" value=".1"/>				
	Gamma <input type="text" value=".1"/>				
	Seasonal <input type="text" value="Additive"/>				
	No: of Periodic Observation <input type="text" value="2"/>				
	Initial Values				
	Level <input type="text" value="Optional"/>				
	Trend <input type="text" value="Optional"/>				
	Season <input type="text" value="Optional"/>				
	Optimizer Inputs <input type="text" value="Optional"/>				
	Confidence <input type="text" value="95"/>				
	Show Range <input type="text" value="False"/>				
Apply					

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be directed to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
	12/4/2018 - 15:48:41 : Process Initiated...				
	12/4/2018 - 15:48:44 : CSV0 is started.				
	12/4/2018 - 15:48:44 : CSV0 is completed.				
	12/4/2018 - 15:48:44 : R-Triple Exponential Smoothing1 is started.				
	12/4/2018 - 15:48:47 : 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').

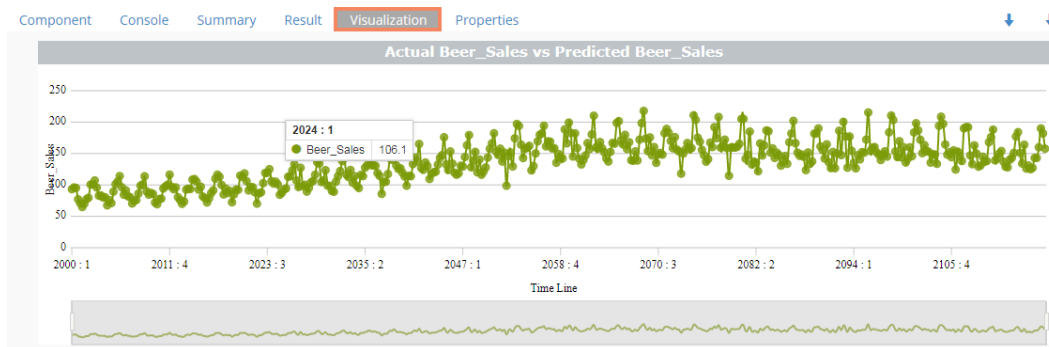
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Period1
1965	January	93.2	2000	1
1965	February	96	2000	2
1965	March	95.2	2000	3
1965	April	77.1	2000	4
1965	May	70.9	2001	1
1965	June	64.8	2001	2
1965	July	70.1	2001	3
1965	August	77.3	2001	4
1965	September	79.5	2002	1
1965	October	100.6	2002	2

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 Series Chart.



- x) Click the 'Summary' tab to view the model summary.


```

Component  Console  Summary  Result  Visualization  Properties
----- Summary of the model -----
Column used in the algorithm :
  Beer_Sales      (double)

fitted      Length Class  Mode
-----
x           468    ts     numeric
alpha       1     -none- numeric
beta        1     -none- numeric
gamma       1     -none- numeric
coefficients 6     -none- numeric
seasonal    1     -none- character
SSE         1     -none- numeric
call        7     -none- call

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

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

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

Coefficients:
      [,1]
a 160.221268
b  1.756644
s1 -4.297796
s2 -1.413476
s3 12.655244
s4 10.583199
----- End of Summary -----

```

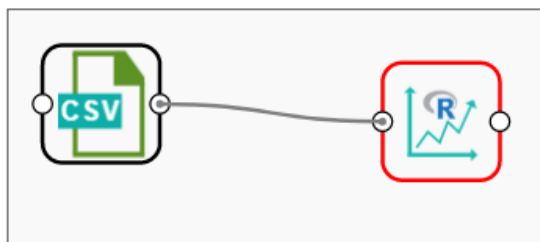
Note:

- 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.
- c. Predicted values will be appended to the target column in the result view for all the 'Forecasting' algorithms.

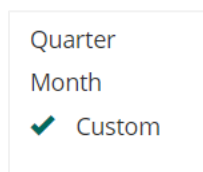
8.2.2. Single Exponential Smoothing

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 (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. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - ii. **Period Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is '**Custom**' hence, '**Period Values**' field is displayed under the '**New Column Information**'.

Note:

- a. The '**Period Values**' field is dependent upon the '**Period**' field.
- b. The '**Period Per Year**' field will display only when the selected value for the '**Period**' field is '**Custom.**'

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode	Forecast ▾			
Advanced	Period To Forecast	1			
	Column Selection				
	Target Variable	Beer_Sales ▾			i
	Input Data Handling				
	Period	Custom ▾			
	Periods per year	4			
	Start Period	1			
	Start Year	2000			
	New Column Information				
	Year Values	Year1			i
	Period Values	Period1			i
		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: $0 < \alpha \leq 1$.
 - 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'

Component	Console	Summary	Result	Visualization	Properties
General	Behavior				
Properties	Alpha	.3			i
Advanced	No: of Periodic Observation	2			i
	Initial Values				
	Level	Optional			
	Confidence	95			
	Show Range	True ▾			
		Apply			

- v) Click 'Run'
- vi) Users will be directed to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

12/4/2018 - 18:42:16 : Process Initiated...

12/4/2018 - 18:42:17 : Data Service0 is started.

12/4/2018 - 18:42:18 : Data Service0 is completed.

12/4/2018 - 18:42:18 : R-Single Exponential Smoothing1 is started.

12/4/2018 - 18:42:20 : R-Single Exponential Smoothing1 is completed.

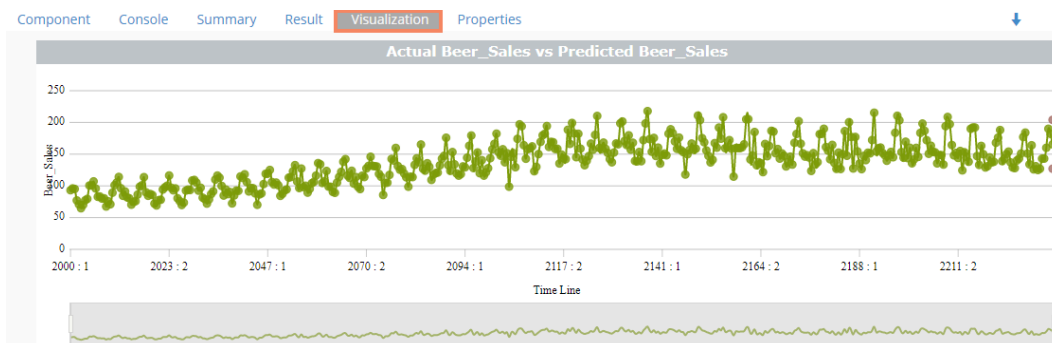
- vii) Follow the below-given steps to display the result view:
 - a. Click the dragged algorithm component onto the workspace.
 - b. Click the 'Result' tab.
- viii) Predicted values will be appended to the target column in the result data (In this case, the selected output mode is 'Forecasting').

Component Console Summary **Result** Visualization Properties

Year	Month	Beer_Sales	Year1	Period1	Lower_Range_95_12	Upper_Range_95_12
2003	May	131	2230	1		
2003	June	125	2230	2		
2003	July	127	2231	1		
2003	August	143	2231	2		
2003	September	143	2232	1		
2003	October	160	2232	2		
2003	November	190	2233	1		
2003	December	182	2233	2		
		165.533	2234	1	127.048	204.017

Showing 461 to 469 of 469 entries Previous 1 ... 43 44 45 46 47 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.

```

Component  Console  Summary  Result  Visualization  Properties

----- Summary of the model -----

Column used in the algorithm :
      Beer_Sales      (double)

      Length Class  Mode
fitted      934   mts  numeric
x           468   ts   numeric
alpha        1  -none- numeric
beta         1  -none- logical
gamma        1  -none- logical
coefficients 1  -none- numeric
seasonal     1  -none- character
SSE          1  -none- numeric
call         6  -none- call

The Model representation
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = tso, alpha = 0.3, beta = FALSE, gamma = FALSE, start.periods = 2)

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

Coefficients:
      [,1]
a 165.5328

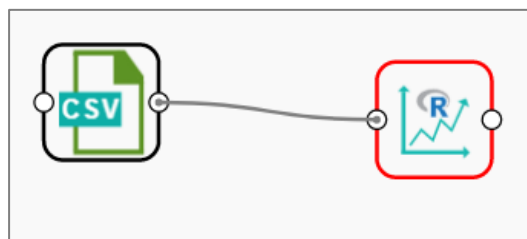
----- End of Summary -----

```

8.2.3. 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 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. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - ii. **Month Values:** Enter a name for the column containing Month Value (This field will be Predefined, but users can change the value if needed). In this case, the selected Period option is '**Month**' hence, '**Month Values**' field is displayed under the '**New Column Information**'.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode	Forecast ▾			
Advanced	Period To Forecast	1			
	Column Selection				
	Target Variable	Beer_Sales ▾ ⓘ			
	Input Data Handling				
	Period	Month ▾			
	Start Period	1			
	Start Year	2000			
	New Column Information				
	Year Values	Year1 ⓘ			
	Month Values	Month1 ⓘ			
	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: $0 < \alpha \leq 1$
 - 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 comma 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 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) Click ‘Run’

vi) Users will be directed to 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.

viii) Predicted values will be appended to the target column in the result data.

(The selected output mode is 'Forecasting').

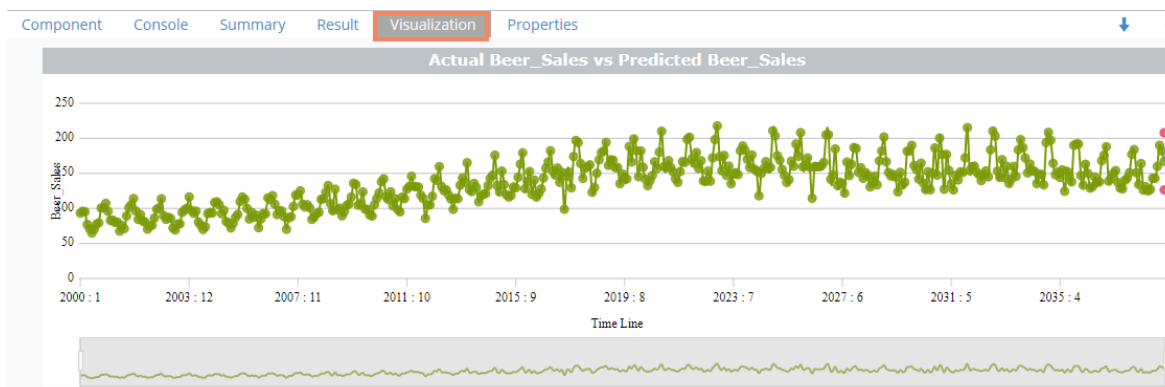
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	Lower_Range_95_13	Upper_Range_95_13
2003	May	131	2038	5		
2003	June	125	2038	6		
2003	July	127	2038	7		
2003	August	143	2038	8		
2003	September	143	2038	9		
2003	October	160	2038	10		
2003	November	190	2038	11		
2003	December	182	2038	12		
		167.205	2039	1	126.358	208.053

Showing 461 to 469 of 469 entries Previous 1 ... 43 44 45 46 47 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.


```

Component  Console  Summary  Result  Visualization  Properties

----- Summary of the model -----

Column used in the algorithm :
    Beer_Sales    (double)

      Length Class  Mode
fitted    1398  mts   numeric
x          468   ts   numeric
alpha         1 -none- numeric
beta          1 -none- numeric
gamma         1 -none- logical
coefficients  2 -none- numeric
seasonal      1 -none- character
SSE           1 -none- numeric
call         6 -none- call

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

Call:
HoltWinters(x = tso, alpha = 0.3, beta = 0.1, gamma = FALSE, start.periods = 2)

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

Coefficients:
      [,1]
a 165.251008
b  1.954222

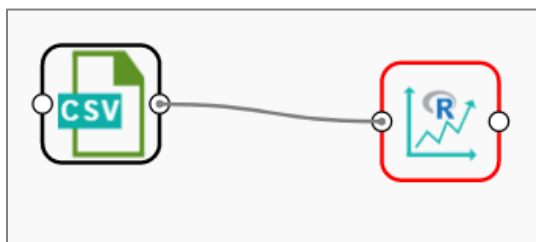
----- End of Summary -----

```

8.2.4. 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 period of forecasting by choosing any one option from the drop-down menu.

Quarter

Month

✓ Custom

- 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. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - ii. **Month Values:** Enter a name for the column containing Month Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is 'Month' hence, 'Month Values' field is displayed under the 'New Column Information'.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode	Forecast ▾			
Advanced	Period To Forecast	1			
	Column Selection				
	Target Variable	Beer_Sales ▾ ⓘ			
	Input Data Handling				
	Period	Month ▾			
	Start Period	1			
	Start Year	2000			
	New Column Information				
	Year Values	Year1 ⓘ			
	Month Values	Month1 ⓘ			
	Manual Arima	<input checked="" type="checkbox"/>			
		<input type="button" value="Apply"/> <input type="button" value="Next"/>			

- iii) Click the 'Advanced' tab and configure if required
 - a. Configure the following 'Behavior' fields:

- i. **Auto regressive 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 comma 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.

xii) Click ‘Apply’

The screenshot shows a configuration window with a sidebar on the left containing tabs: Component, General, Properties, and Advanced (highlighted with a red box). The main area has tabs: Console, Summary, Result, Visualization, and Properties. Under the 'Advanced' tab, there are two sections: 'Behavior' and 'Initial Values'. The 'Behavior' section contains three input fields: 'Auto regressive order (p)' with value 0, 'Degree of differencing (d)' with value 0, and 'Moving Average order (q)' with value 0. The 'Initial Values' section contains two fields: 'Confidence' with value 95, and 'Show Range' with a dropdown menu set to 'True'. An 'Apply' button is located at the bottom right of the configuration area.

- iv) Click ‘Run’
- v) Users will be directed to the ‘Console’ tab.

The screenshot shows the 'Console' tab selected in the sidebar. The main area displays a log of events:

```

12/4/2018 - 21:3:12 : Process Initiated...
12/4/2018 - 21:3:14 : CSV0 is started.
12/4/2018 - 21:3:15 : CSV0 is completed.
12/4/2018 - 21:3:15 : R-Arima1 is started.
12/4/2018 - 21:3:17 : R-Arima1 is completed.
  
```

- 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) Predicted values will be appended to the target column in the result data. (The selected output mode is ‘Forecasting’).

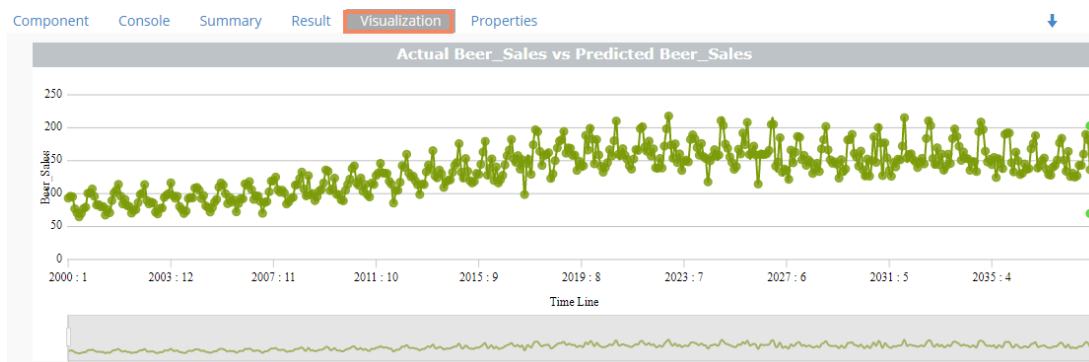
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	Lower_Range_95_14	Upper_Range_95_14
2003	May	131	2038	5		
2003	June	125	2038	6		
2003	July	127	2038	7		
2003	August	143	2038	8		
2003	September	143	2038	9		
2003	October	160	2038	10		
2003	November	190	2038	11		
2003	December	182	2038	12		
		136.364	2039	1	69.816	202.911

Showing 461 to 469 of 469 entries Previous 1 ... 43 44 45 46 47 Next

- viii) Click the 'Visualization' tab.
- ix) The result data will be displayed via the time series chart.



- x) Click the 'Summary' tab to view the model summary.

Component Console **Summary** Result Visualization Properties

```

----- Summary of the model -----

Column 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

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.354107e-14 33.95339 27.82144 -7.333014 23.30006 1.887145
              ACF1
Training set 0.8314011

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

```

Note: The 'R-ARIMA' algorithm does not contain 'Advanced' tab when 'Manual ARIMA' option is disabled. The following images display the 'Result' and 'Visualization' tabs for the same dataset when manual ARIMA option has been disabled.

Result Tab

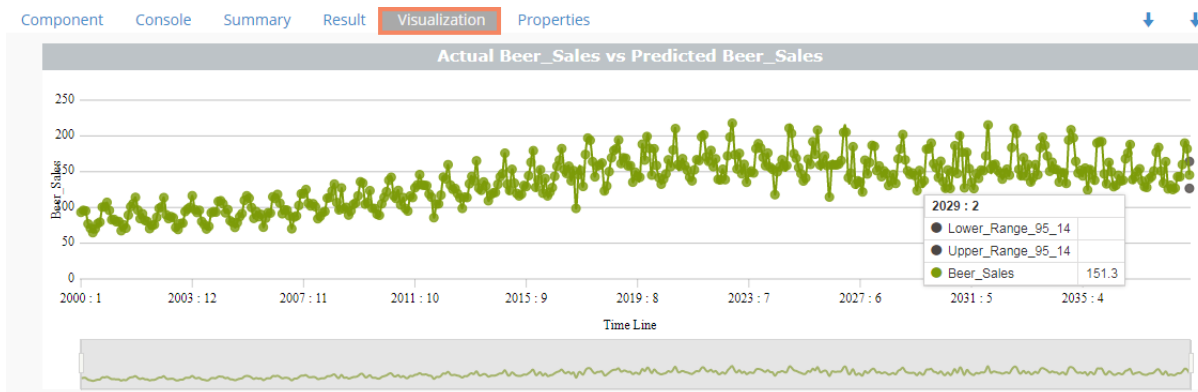
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	Lower_Range_95_14	Upper_Range_95_14
2003	May	131	2038	5		
2003	June	125	2038	6		
2003	July	127	2038	7		
2003	August	143	2038	8		
2003	September	143	2038	9		
2003	October	160	2038	10		
2003	November	190	2038	11		
2003	December	182	2038	12		
		145.469	2039	1	126.69	164.248

Showing 461 to 469 of 469 entries Previous 1 ... 43 44 45 46 47 Next

Visualization Tab

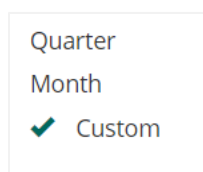


8.2.5. R- Auto Forecasting

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 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**').
 - ii. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - iii. **Month Values:** Enter a name for the column containing Month Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period the option is '**Month**' hence, '**Month Values**' field is displayed under the '**New Column Information**'.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode	Forecast ▾			
Advanced	Period To Forecast	1			
	Column Selection				
	Target Variable	Beer_Sales ▾ ⓘ			
	Input Data Handling				
	Period	Month ▾			
	Start Period	1			
	Start Year	2000			
	New Column Information				
	Year Values	Year1 ⓘ			
	Month Values	Month1 ⓘ			
		Apply			

- 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 (Holtwinters’ 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.

Component	Console	Summary	Result	Visualization	Properties
General	Behavior				
Properties	Seasonal <input type="text" value="Additive"/>				
Advanced	No: of Periodic Observation <input type="text" value="2"/>				
	Initial Values				
	Level <input type="text" value="Optional"/>				
	Trend <input type="text" value="Optional"/>				
	Season <input type="text" value="Optional"/>				
	Optimizer Inputs <input type="text" value="Optional"/>				
	Confidence <input type="text" value="95"/>				
	Show Range <input type="text" value="True"/>				
	<input type="button" value="Apply"/>				

- iv) Click **'Apply'**
- v) Click **'Run'**
- vi) Users will be redirected to the **'Console'** tab.

Component	Console	Summary	Result	Visualization	Properties
	13/4/2018 - 12:26:2 : Process Initiated...				
	13/4/2018 - 12:26:4 : CSV0 is started.				
	13/4/2018 - 12:26:5 : CSV0 is completed.				
	13/4/2018 - 12:26:5 : R-Auto Forecasting1 is started.				
	13/4/2018 - 12:26:7 : R-Auto Forecasting1 is completed.				

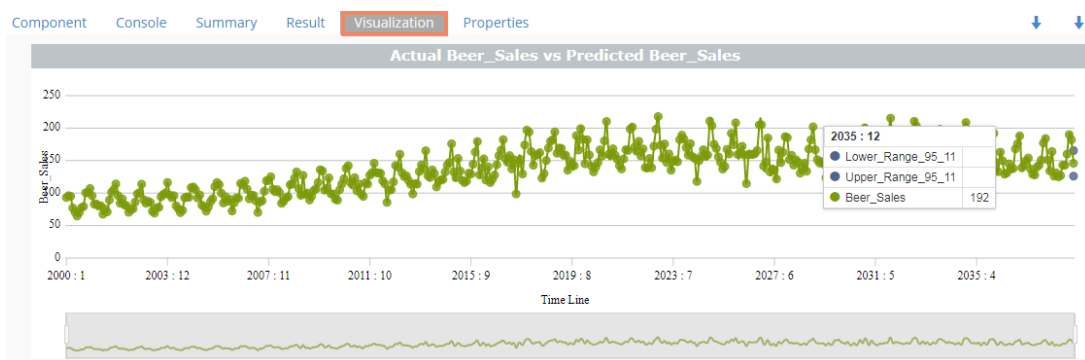
- vii) Follow the below given steps to display the result view:
 - a. Click the dragged algorithm component onto the workspace.
 - b. Click the **'Result'** tab.
- viii) Predicted values will be appended to the target column in the result data. (The selected output mode is **'Forecasting'**).

Year	Month	Beer_Sales	Year1	Month1	Lower_Range_95_11	Upper_Range_95_11
2003	May	131	2038	5		
2003	June	125	2038	6		
2003	July	127	2038	7		
2003	August	143	2038	8		
2003	September	143	2038	9		
2003	October	160	2038	10		
2003	November	190	2038	11		
2003	December	182	2038	12		
		145.779	2039	1	126.087	165.471

Showing 461 to 469 of 469 entries

Previous 1 ... 43 44 45 46 47 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.

```

Component Console Summary Result Visualization Properties
----- Summary of the model -----

Column used in the algorithm :
  Beer_Sales      (double)

      Length Class Mode
fitted    1824 mts  numeric
x          468 ts   numeric
alpha      1  -none- numeric
beta       1  -none- numeric
gamma      1  -none- numeric
coefficients 14 -none- numeric
seasonal   1  -none- character
SSE        1  -none- numeric
call       7  -none- call

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

Call:
HoltWinters(x = tso, alpha = NULL, beta = NULL, gamma = NULL, seasonal = c("additive"), start.periods = 2)

Smoothing parameters:
alpha: 0.07500866
beta : 0.06694109
gamma: 0.1423688

```

8.2.6. Result View 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) 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 time 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 if the selected Output Mode is ‘Trend’).
 - ii. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - iii. **Month Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is ‘Month’ hence, ‘Month Values’ field is displayed under the ‘New Column Information’.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode: Trend				
Advanced	Column Selection				
	Target Variable: Beer_Sales				
	Input Data Handling				
	Period: Month				
	Start Period: 1				
	Start Year: 2000				
	New Column Information				
	Predicted Column Name: PredictedValues1				
	Year Values: Year1				
	Month Values: Month1				
	Apply				

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

Component Console Summary Result Visualization Properties

General Behavior

Properties

Advanced

Alpha

Beta

Gamma

Seasonal

No: of Periodic Observation

Initial Values

Level

Trend

Season

Optimizer Inputs

Apply

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component Console Summary Result Visualization Properties

12/4/2018 - 20:19:35 : Process Initiated...

12/4/2018 - 20:19:37 : CSV0 is started.

12/4/2018 - 20:19:37 : CSV0 is completed.

12/4/2018 - 20:19:37 : R-Triple Exponential Smoothing1 is started.

12/4/2018 - 20:19:40 : 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.

Component Console Summary Result Visualization Properties

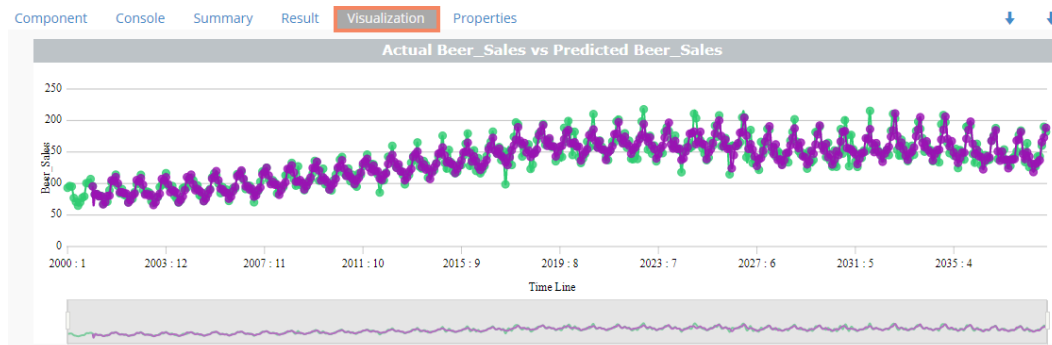
Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	PredictedValues1
1966	September	89.3	2001	9	80.753
1966	October	101.1	2001	10	105.115
1966	November	105.2	2001	11	103.696
1966	December	114.1	2001	12	110.286
1967	January	96.3	2002	1	100.283
1967	February	84.4	2002	2	86.233
1967	March	91.2	2002	3	86.217
1967	April	81.9	2002	4	84.336
1967	May	80.5	2002	5	84.056
1967	June	70.4	2002	6	69.782

Showing 21 to 30 of 468 entries Previous 1 2 3 4 5 ... 47 Next

viii) Click the 'Visualization' tab.

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



2. Single 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 '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 time 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 if the selected Output Mode is 'Trend').
- ii. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
- iii. **Month Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is 'Month' hence, 'Month Values' field is displayed under the 'New Column Information'.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode <input type="text" value="Trend"/>				
Advanced	Column Selection				
	Target Variable <input type="text" value="Beer_Sales"/>				
	Input Data Handling				
	Period <input type="text" value="Month"/>				
	Start Period <input type="text" value="1"/>				
	Start Year <input type="text" value="2000"/>				
	New Column Information				
	Predicted Column Name <input type="text" value="PredictedValues1"/>				
	Year Values <input type="text" value="Year1"/>				
	Month Values <input type="text" value="Month1"/>				
	<input type="button" value="Apply"/>				

- 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: $0 < \alpha \leq 1$.)
 - 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.

Component	Console	Summary	Result	Visualization	Properties
General	Behavior				
Properties	Alpha <input type="text" value=".3"/>				
Advanced	No. of Periodic Observation <input type="text" value="2"/>				
	Initial Values				
	Level <input type="text" value="Optional"/>				
	<input type="button" value="Apply"/>				

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
	12/4/2018 - 20:41:3 : Process Initiated...				
	12/4/2018 - 20:41:4 : CSV0 is started.				
	12/4/2018 - 20:41:5 : CSV0 is completed.				
	12/4/2018 - 20:41:5 : R-Single Exponential Smoothing1 is started.				
	12/4/2018 - 20:41:7 : R-Single 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.

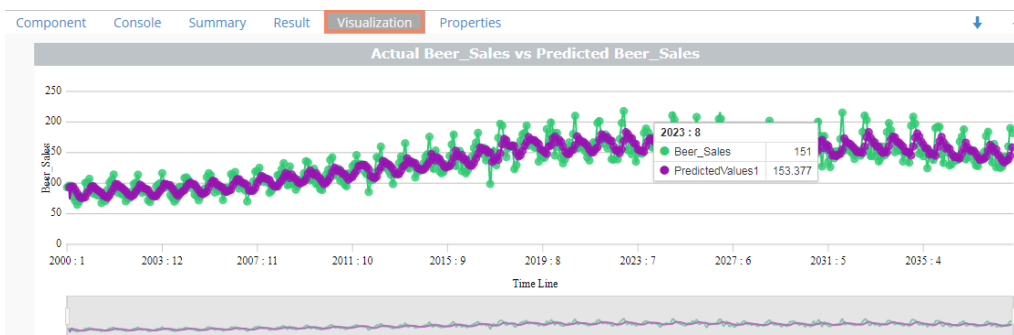
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	PredictedValues1
1965	January	93.2	2000	1	
1965	February	96	2000	2	93.2
1965	March	95.2	2000	3	94.04
1965	April	77.1	2000	4	94.388
1965	May	70.9	2000	5	89.202
1965	June	64.8	2000	6	83.711
1965	July	70.1	2000	7	78.038
1965	August	77.3	2000	8	75.656
1965	September	79.5	2000	9	76.15
1965	October	100.6	2000	10	77.155

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

- viii) Click the 'Visualization' tab.
- ix) The result data will be displayed via the Time Series Chart.



- x) Click the 'Summary' tab to view the model summary.

Component Console **Summary** Result Visualization Properties

```

----- Summary of the model -----
Column used in the algorithm :
  Beer_Sales      (double)

fitted      Length Class Mode
x            934   mts  numeric
alpha       468   ts   numeric
beta         1    -none- numeric
gamma        1    -none- logical
coefficients 1    -none- numeric
seasonal     1    -none- character
SSE          1    -none- numeric
call        6    -none- call

The Model representation
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = tso, alpha = 0.3, beta = FALSE, gamma = FALSE, start.periods = 2)

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

Coefficients:
      [,1]
a 165.5328

----- End of 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 '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. **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').
 - ii. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
 - iii. **Month Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is 'Month' hence, 'Month Values' field is displayed under the 'New Column Information'.

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode		Trend ▾		
Advanced	Column Selection		Target Variable: Beer_Sales ▾ ⓘ		
	Input Data Handling		Period: Month ▾		
	Start Period		1		
	Start Year		2000		
	New Column Information		Predicted Column Name: PredictedValues1 ⓘ		
	Year Values		Year1 ⓘ		
	Month Values		Month1 ⓘ		
Apply					

- 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: $0 < \alpha \leq 1$.)
 - 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'**

Component	Console	Summary	Result	Visualization	Properties																					
General	Behavior																									
Properties																										
Advanced	<table> <tr> <td>Alpha</td> <td><input type="text" value=".3"/></td> <td></td> </tr> <tr> <td>Beta</td> <td><input type="text" value=".1"/></td> <td></td> </tr> <tr> <td>No: of Periodic Observation</td> <td><input type="text" value="2"/></td> <td></td> </tr> <tr> <td colspan="3">Initial Values</td> </tr> <tr> <td>Level</td> <td><input type="text" value="Optional"/></td> <td></td> </tr> <tr> <td>Trend</td> <td><input type="text" value="Optional"/></td> <td></td> </tr> <tr> <td>Optimizer Inputs</td> <td><input type="text" value="Optional"/></td> <td></td> </tr> </table>					Alpha	<input type="text" value=".3"/>		Beta	<input type="text" value=".1"/>		No: of Periodic Observation	<input type="text" value="2"/>		Initial Values			Level	<input type="text" value="Optional"/>		Trend	<input type="text" value="Optional"/>		Optimizer Inputs	<input type="text" value="Optional"/>	
Alpha	<input type="text" value=".3"/>																									
Beta	<input type="text" value=".1"/>																									
No: of Periodic Observation	<input type="text" value="2"/>																									
Initial Values																										
Level	<input type="text" value="Optional"/>																									
Trend	<input type="text" value="Optional"/>																									
Optimizer Inputs	<input type="text" value="Optional"/>																									
					Apply																					

- v) Click **'Run'**
- vi) Users will be redirected to the **'Console'** tab.

Component	Console	Summary	Result	Visualization	Properties
12/4/2018 - 20:49:31 : Process Initiated...					
12/4/2018 - 20:49:32 : CSV0 is started.					
12/4/2018 - 20:49:33 : CSV0 is completed.					
12/4/2018 - 20:49:33 : R-Double Exponential Smoothing1 is started.					
12/4/2018 - 20:49:35 : 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.

Component Console Summary **Result** Visualization Properties

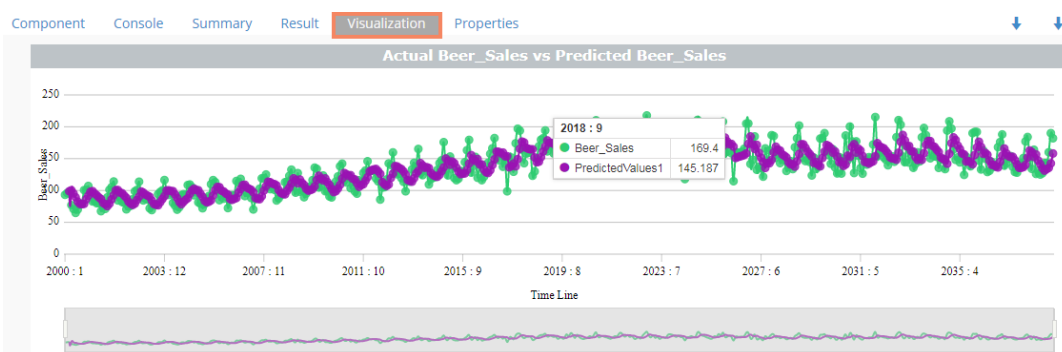
Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	PredictedValues1
1965	January	93.2	2000	1	
1965	February	96	2000	2	
1965	March	95.2	2000	3	98.8
1965	April	77.1	2000	4	100.412
1965	May	70.9	2000	5	95.411
1965	June	64.8	2000	6	89.315
1965	July	70.1	2000	7	82.482
1965	August	77.3	2000	8	78.918
1965	September	79.5	2000	9	78.534
1965	October	100.6	2000	10	78.955

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

viii) Click the 'Visualization' tab.

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



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

- 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.
- 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 if the selected Output Mode is ‘Trend’).
- ii. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
- iii. **Month Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is ‘Month’ hence, ‘Month Values’ field is displayed under the ‘New Column Information’.

- 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: $0 < \alpha \leq 1$.)
 - 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 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) Click ‘Run’
- vi) Users will be redirected to the ‘Console’ tab.

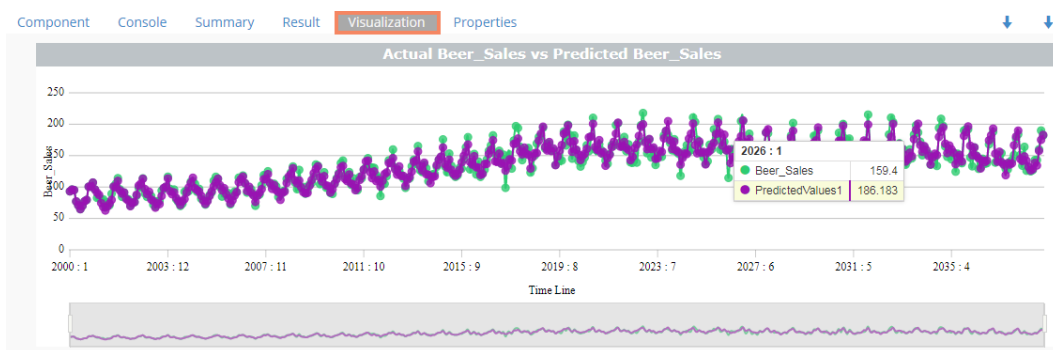
Component	Console	Summary	Result	Visualization	Properties	Status
18/7/2017 - 18:30:25 : Process Initiated...						
18/7/2017 - 18:30:26 : csv0 is started.						
18/7/2017 - 18:30:26 : csv0 is completed.						
18/7/2017 - 18:30:26 : R-Auto Arima1 is started.						
18/7/2017 - 18:30:30 : R-Auto Arima1 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.

Year	Month	Beer_Sales	Year1	Month1	PredictedValues1
1965	January	93.2	2000	1	93.146
1965	February	96	2000	2	95.974
1965	March	95.2	2000	3	95.184
1965	April	77.1	2000	4	77.104
1965	May	70.9	2000	5	70.909
1965	June	64.8	2000	6	64.813
1965	July	70.1	2000	7	70.106
1965	August	77.3	2000	8	77.298
1965	September	79.5	2000	9	79.497
1965	October	100.6	2000	10	100.577

Showing 1 to 10 of 468 entries

- viii) Click the 'Visualization' tab.
- ix) The result data will be displayed via the Time Series Chart.



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. **Year Values:** Enter a name for the column containing year value. (This field will be predefined, but users can change the value if needed).
- iii. **Month Values:** Enter a name for the column containing period Value (This field will be predefined, but users can change the value if needed). In this case, the selected Period option is 'Month' hence, 'Month Values' field is displayed under the 'New Column Information'.

Component	Console	Summary	Result	Visualization	Properties
General	Output information				
Properties	Output Mode		Trend ▾		
Advanced	Column Selection				
	Target Variable		Beer_Sales ▾		i
	Input Data Handling				
	Period		Month ▾		
	Start Period		1		
	Start Year		2000		
	New Column Information				
	Predicted Column Name		PredictedValues1		i
	Year Values		Year1		i
	Month Values		Month1		i
Apply					

- 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: $0 < \alpha \leq 1$.)
 - 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 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.

The screenshot shows a configuration panel with a sidebar on the left containing 'Component', 'General', 'Properties', and 'Advanced' (highlighted with a red box). The main area is titled 'Behavior' and contains two sections: 'Behavior' and 'Initial Values'. Under 'Behavior', there is a dropdown menu for 'Seasonal' set to 'Additive' and a text input for 'No. of Periodic Observation' set to '2'. Under 'Initial Values', there are four text input fields for 'Level', 'Trend', 'Season', and 'Optimizer Inputs', all set to 'Optional'. An 'Apply' button is located at the bottom right.

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

The screenshot shows the 'Console' tab with the following log entries:

```

13/4/2018 - 12:4:59 : Process Initiated...
13/4/2018 - 12:5:0 : CSV0 is started.
13/4/2018 - 12:5:1 : CSV0 is completed.
13/4/2018 - 12:5:1 : R-Auto Forecasting1 is started.
13/4/2018 - 12:5:3 : R-Auto Forecasting1 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 'predicted values' will be added to the result data.

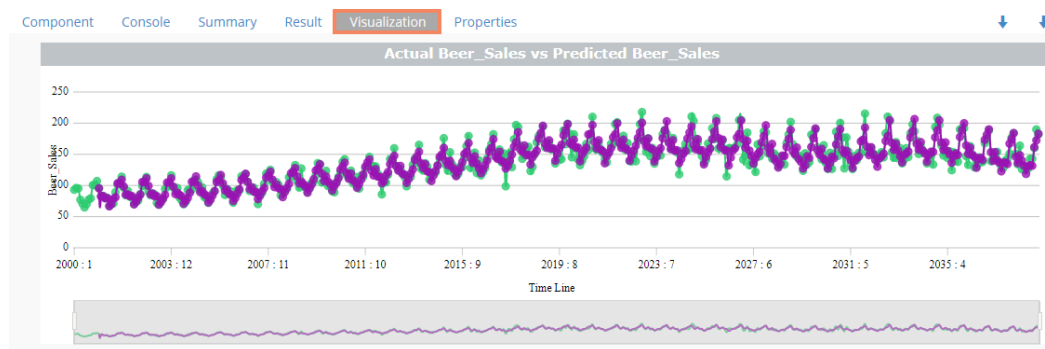
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Year	Month	Beer_Sales	Year1	Month1	PredictedValues1
1965	November	100.7	2000	11	
1965	December	107.1	2000	12	
1966	January	95.9	2001	1	95.382
1966	February	82.8	2001	2	82.474
1966	March	83.3	2001	3	82.978
1966	April	80	2001	4	79.404
1966	May	80.4	2001	5	79.775
1966	June	67.5	2001	6	66.58
1966	July	75.7	2001	7	70.145
1966	August	71.1	2001	8	78.369

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

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the time series chart.



Note: Users can click the 'Summary' tab to view the model summary for the Forecasting models with 'Trend' as the output mode.

8.3. Association

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

8.3.1. 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 string columns. It cannot display date columns.
- iii. **Transaction Id Column:** Select the column containing Transaction Ids to which you can apply the algorithm.

Note: 'Transaction Id Column' field appears only when '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).

- 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 the 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 default appearance of the items out of the above-given choices using a drop-down menu

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

Component	Console	Summary	Result	Visualization	Properties
General	Output Appearance				
Properties					
Advanced					
	Lhs Item(s)	<input type="text" value="Optional"/>			
	Rhs Item(s)	<input type="text" value="Optional"/>			
	Both Item(s)	<input type="text" value="Optional"/>			
	None Item(s)	<input type="text" value="Optional"/>			
	Default Appearance	<input type="text" value="Both"/>			
	Min Length	<input type="text" value="1"/>			
	Max Length	<input type="text" value="10"/>			
	Performance				
	Sort Type	<input type="text" value="Ascending Transaction Size"/>			
	Filter Criteria	<input type="text" value="0.1"/>			
	Use Tree Structure	<input type="text" value="True"/>			
	Use Heapsort	<input type="text" value="True"/>			
	Optimize Memory	<input type="text" value="False"/>			
	Load Transaction into memory	<input type="text" value="True"/>			
					<input type="button" value="Apply"/>

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be directed to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
18/7/2017 - 19:12:14 : Process Initiated...
18/7/2017 - 19:12:15 : csv0 is started.
18/7/2017 - 19:12:15 : csv0 is completed.
18/7/2017 - 19:12:15 : R-Apriori1 is started.
18/7/2017 - 19:12:19 : 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	1.0	1.667
{Affluence=Low} => {SKYBox=Sky+HD 2TB}	0.12	1.0	1.515
{Affluence=Very Low} => {MetroPolitan=No}	0.1	0.833	2.083
{Affluence=Mid Low} => {MetroPolitan=Yes}	0.12	0.857	1.429
{Affluence=Mid Low} => {SKYBox=Sky+HD 2TB}	0.12	0.857	1.299
{Demographiclifestyle=Liberal Opinion} => {HouseholdComposition=Men only HH}	0.12	0.857	2.521
{Demographiclifestyle=Liberal Opinion} => {MetroPolitan=Yes}	0.12	0.857	1.429
{Demographiclifestyle=Liberal Opinion} => {SKYBox=Sky+HD 2TB}	0.12	0.857	1.299
{Affluence=Mid} => {MetroPolitan=No}	0.12	0.857	2.143
{Demographiclifestyle=Terraced Melting Pot} => {HouseholdComposition=Men only HH}	0.14	0.875	2.574

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
{Chicken,Magazine,Oranges}	396	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
{Waffles}	434	
{Beets,Bubble Gum,Chocolate,Cole Slaw,Detergent,Shrimp,Strawberry,Toothpaste}	486	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
{Bubble Gum,Chocolate,Cole Slaw,Magazine,Strawberry,Waffles,Yams}	576	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
{Chicken,Detergent,Lightbulb,TV Dinner,Waffles,Water}	664	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,
{Chocolate,Cole Slaw,Oranges,Shrimp}	700	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,

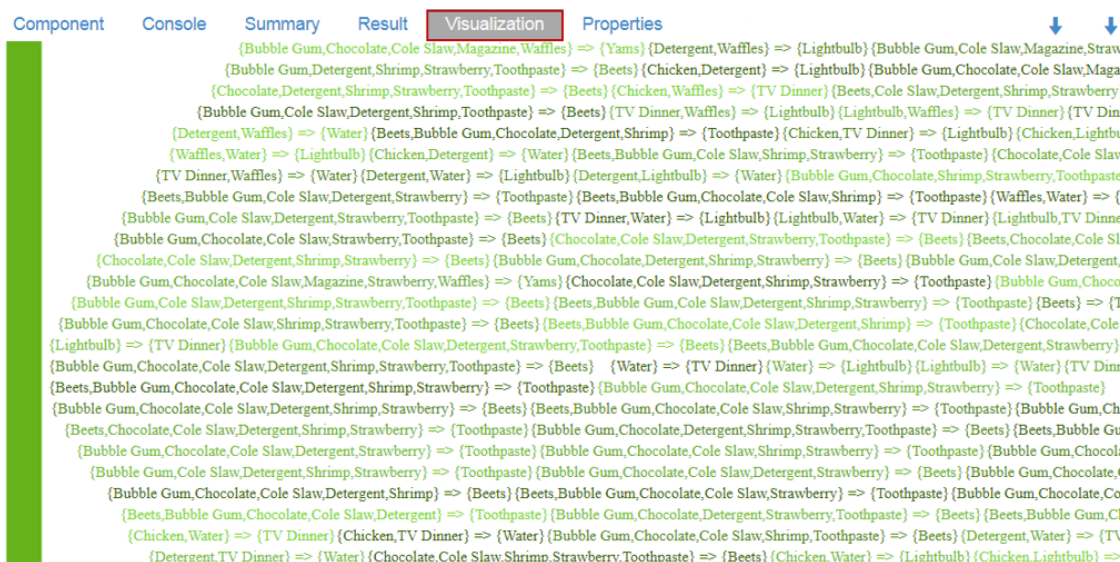
Showing 1 to 6 of 6 entries Previous 1 Next

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the word tag chart.

a. Result View for the 'Rules' output mode.



b. Result View for the 'Transaction' output mode.



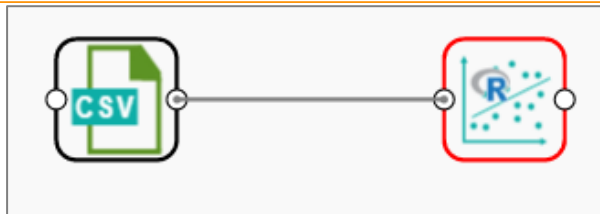
8.4. Regression Analysis

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':

8.4.1. 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 will be applied
 - ii. **Independent Column:** Select the required input columns against which the regression the 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.

Component	Console	Summary	Result	Visualization	Properties
General	Column selection				
Properties	Dependent Column	mpg ▾		<i>i</i>	
	Independent Column	cylinders ▾		<i>i</i>	
Advanced	New Column Information				
	Predicted Column Name	PredictedValues1		<i>i</i>	
Apply					

- 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:** Selecting this option will skip the records containing missing values from the dependent and independent columns.
 2. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.
 3. **Stop:** Selecting this option will stop application of the algorithm 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:** Selecting this option will ignore aliased coefficients from the coefficient covariance matrix.
 2. **False:** Selecting this option will show an error in a model containing aliased coefficients
 - ii. **Contrasts:** Selecting this option will 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 will take 0.95 as the default value.

Component Console Summary Result Visualization Properties

General **Input Data Handling**

Properties

Advanced

Missing values

Behavior

Allow Singular Fit

Contrasts

Confidence Level

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

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

19/7/2017 - 12:19:38 : Process Initiated...

19/7/2017 - 12:19:38 : csv0 is started.

19/7/2017 - 12:19:38 : csv0 is completed.

19/7/2017 - 12:19:38 : R-Linear Regression1 is started.

19/7/2017 - 12:19:39 : R-Linear 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.
 - i. A new column 'Predicted Values1' will be added to the result data displaying the predicted values.

Component Console Summary **Result** Visualization Properties

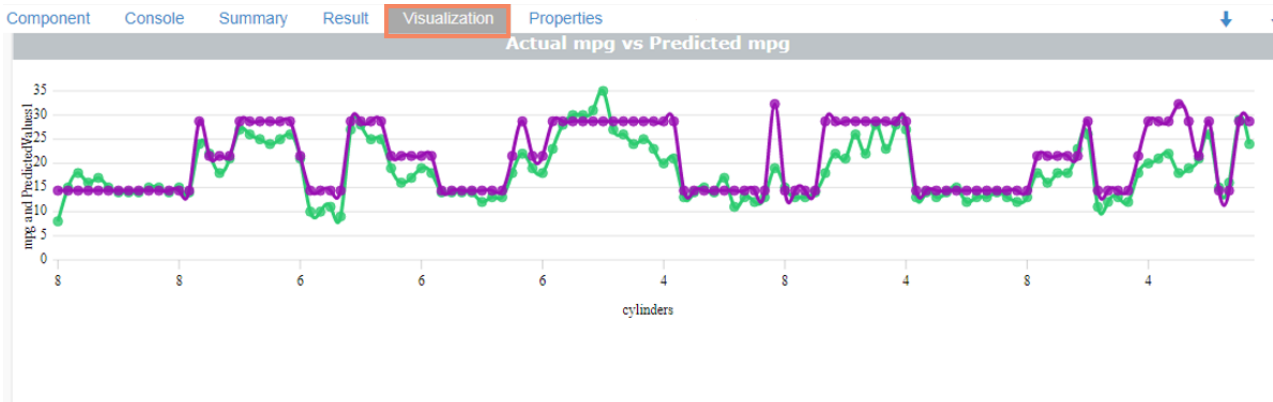
Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname	PredictedValues1
8	8	307	130	3504	12	70	1	chevrolet chevelle malibu	14.3646987928089
15	8	350	165	3693	11.5	70	1	buick skylark 320	14.3646987928089
18	8	318	150	3436	11	70	1	plymouth satellite	14.3646987928089
16	8	304	150	3433	12	70	1	amc rebel sst	14.3646987928089
17	8	302	140	3449	10.5	70	1	ford torino	14.3646987928089
15	8	429	198	4341	10	70	1	ford galaxie 500	14.3646987928089
14	8	454	220	4354	9	70	1	chevrolet impala	14.3646987928089
14	8	440	215	4312	8.5	70	1	plymouth fury iii	14.3646987928089
14	8	455	225	4425	10	70	1	pontiac catalina	14.3646987928089
15	8	390	190	3850	8.5	70	1	amc ambassador dpl	14.3646987928089

Showing 1 to 10 of 398 entries Previous 1 2 3 4 5 ... 40 Next

- viii) Click the 'Visualization' tab.

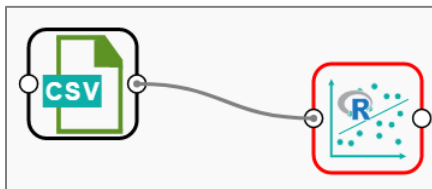
ix) The result data will be displayed via the Time Series Chart.



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 ‘Advanced’ tab is explained below for the remaining R sub-algorithms provided under ‘Regression.’

8.4.2. 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.
- 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. **Ignore:** Selecting this option will skip the records containing missing values from the dependent and independent columns.
 2. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.
 3. **Stop:** Selecting this option will stop application of the algorithm 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 will take 0.95 as the default value.

Component Console Summary Result Visualization Properties

General **Input Data Handling**

Properties

Advanced

Missing values

Behavior

Confidence Level ⓘ

Apply

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

19/7/2017 - 12:52:55 : Process Initiated...

19/7/2017 - 12:53:8 : csv0 is started.

19/7/2017 - 12:53:8 : csv0 is completed.

19/7/2017 - 12:53:8 : R-Multiple Linear Regression1 is started.

19/7/2017 - 12:53:8 : R-Multiple Linear 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.

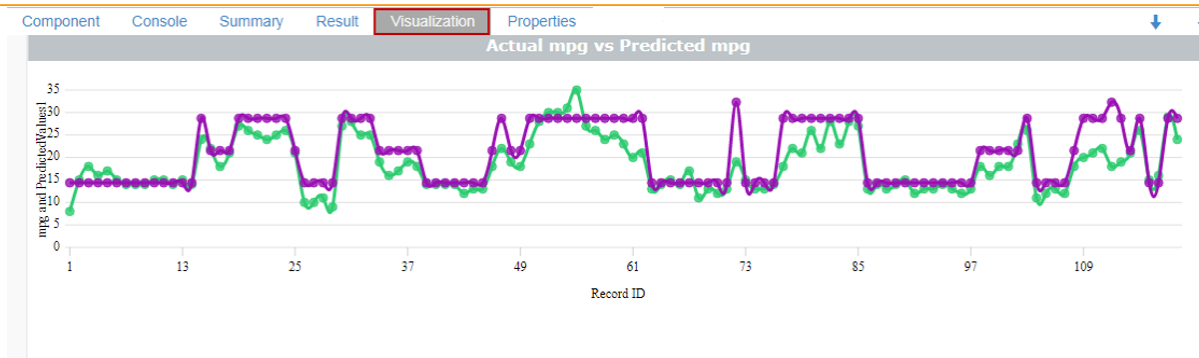
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	carname	PredictedValues1
8	8	307	130	3504	12	70	1	chevrolet chevelle malibu	14.3646987928089
15	8	350	165	3693	11.5	70	1	buick skylark 320	14.3646987928089
18	8	318	150	3436	11	70	1	plymouth satellite	14.3646987928089
16	8	304	150	3433	12	70	1	amc rebel sst	14.3646987928089
17	8	302	140	3449	10.5	70	1	ford torino	14.3646987928089
15	8	429	198	4341	10	70	1	ford galaxie 500	14.3646987928089
14	8	454	220	4354	9	70	1	chevrolet impala	14.3646987928089
14	8	440	215	4312	8.5	70	1	plymouth fury iii	14.3646987928089
14	8	455	225	4425	10	70	1	pontiac catalina	14.3646987928089
15	8	390	190	3850	8.5	70	1	amc ambassador dpl	14.3646987928089

Showing 1 to 10 of 398 entries Previous 1 2 3 4 5 ... 40 Next

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the Time Series Chart.



8.4.3. 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.
- iii) Click the 'Advanced' tab and configure if required:
 - a. 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	Input Data Handling				
Properties	Missing values		Keep ▾		
Advanced	Behavior				
	Family		Binomial ▾		
	Maximum No Of Iterations		25		
Apply					

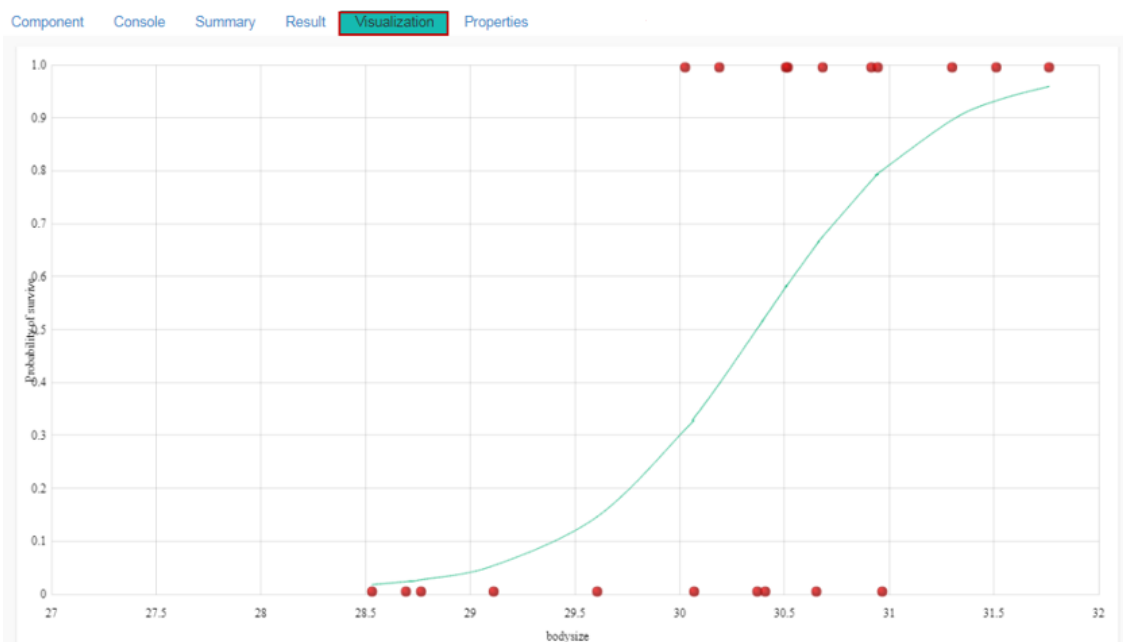
- iv) Click 'Apply'
- v) Click on 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
19/7/2017 - 13:22:35 : Process Initiated...					
19/7/2017 - 13:22:36 : csv0 is started.					
19/7/2017 - 13:22:36 : csv0 is completed.					
19/7/2017 - 13:22:36 : R-Logistic Regression1 is started.					
19/7/2017 - 13:22:36 : R-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.

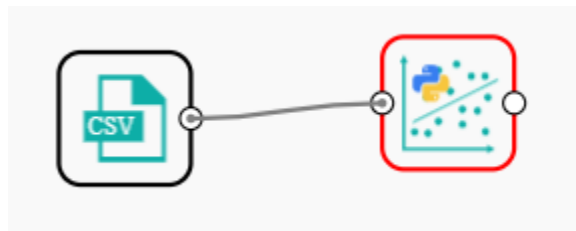
Component	Console	Summary	Result	Visualization	Properties
Showing 1 to 10 of 20 entries					
bodysize	survive	PredictedValues1			
27.1435042073117	0	0.0131319069986111			
27.3698819529919	0	0.0190925848887748			
27.4713319662842	0	0.0225603512636812			
27.9561274911923	0	0.0495342106666383			
28.6486888249175	0	0.142975991879034			
29.2359919584729	1	0.309144256222506			
29.2962754702829	0	0.331181305548463			
29.4650506154866	1	0.396682907099961			
29.7186205793043	0	0.501672626463761			
29.7719375103457	0	0.524047107310898			
Showing 1 to 10 of 20 entries					

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the chart displaying Scatter Plot with Regression Line.



8.4.4. Python Linear Regression

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

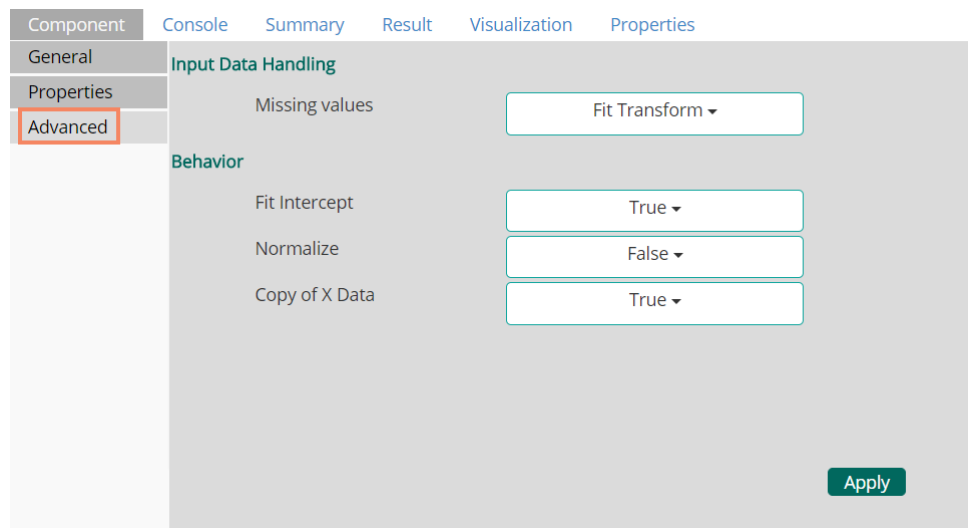


- ii) Configure the following fields in the 'Properties' tab:
 - a. **Column Selection**
 - i. **Dependent Column:** Select the target column on which the regression analysis will be applied
 - ii. **Independent Column:** Select the required input columns against which the regression analysis will be applied to the target column
 - b. **New Column Information**
 - i. **Predicted Column Name:** Enter a name for the new column containing the predicted values.

Component	Console	Summary	Result	Visualization	Properties
General	Column selection				
Properties	Dependent Column	SLength ▾		i	
Advanced	Independent Column	PLength ▾		i	
	New Column Information				
	Predicted Column Name	PredictedValues1		i	
Apply					

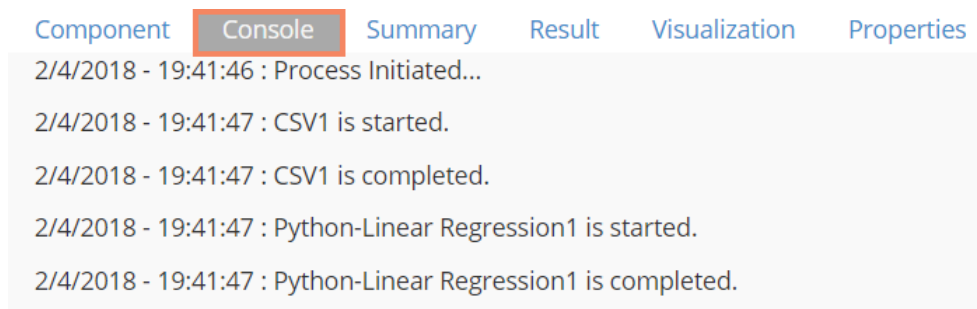
- iii) Click the 'Advanced' tab and configure if required:
 - a. **Input Data Handling**
 - i. **Missing Values:** Select a method to deal with missing values from the drop-down menu
 1. **Fit Transform:** Selecting this option two actions will be performed on the data, Fit and Transform.
 2. **Stop:** Selecting this option will stop application of the algorithm 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



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

- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to 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.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

RowID	SLength	SWidth	PLength	PWidth	PredictedValues1
1	5.1	3.5	1.4	0.2	4.88
2	4.9	3	1.4	0.2	4.88
3	4.7	3.2	1.3	0.2	4.84
4	4.6	3.1	1.5	0.2	4.92
5	5	3.6	1.4	0.2	4.88
6	5.4	3.9	1.7	0.4	5
7	4.6	3.4	1.4	0.3	4.88
8	5	3.4	1.5	0.2	4.92
9	4.4	2.9	1.4	0.2	4.88
10	4.9	3.1	1.5	0.1	4.92

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

- viii) Click the 'Visualization' tab.
- ix) The result data will be displayed via the Scatterplot with Regression Line chart.



8.4.5. 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:

Component	Console	Summary	Result	Visualization	Properties
General	Column selection				
Properties	Dependent Column	SLength ▾		i	
Advanced	Independent Column	2 checked ▾		i	
	New Column Information				
	Predicted Column Name	PredictedValues1		i	
					Apply

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 application of the algorithm 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'

Component Console Summary Result Visualization Properties

General **Input Data Handling**

Properties

Advanced

Missing values

Behavior

Fit Intercept

Normalize

Copy of X Data

- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

2/4/2018 - 19:53:19 : Process Initiated...

2/4/2018 - 19:53:21 : CSV1 is started.

2/4/2018 - 19:53:21 : CSV1 is completed.

2/4/2018 - 19:53:21 : Python-Multiple Linear Regression1 is started.

2/4/2018 - 19:53:21 : Python-Multiple Linear 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.

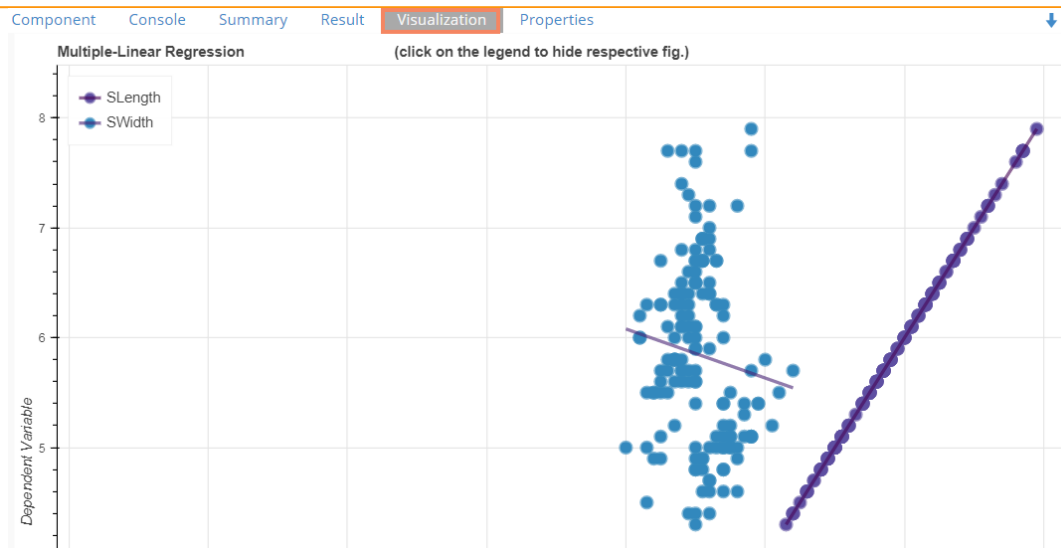
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

RowID	SLength	SWidth	PLength	PWidth	PredictedValues1
1	5.1	3.5	1.4	0.2	5.1
2	4.9	3	1.4	0.2	4.9
3	4.7	3.2	1.3	0.2	4.7
4	4.6	3.1	1.5	0.2	4.6
5	5	3.6	1.4	0.2	5
6	5.4	3.9	1.7	0.4	5.4
7	4.6	3.4	1.4	0.3	4.6
8	5	3.4	1.5	0.2	5
9	4.4	2.9	1.4	0.2	4.4
10	4.9	3.1	1.5	0.1	4.9

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

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the Scatterplot with Regression Line Chart.



8.4.6. 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', selecting 'True' will calculate 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'

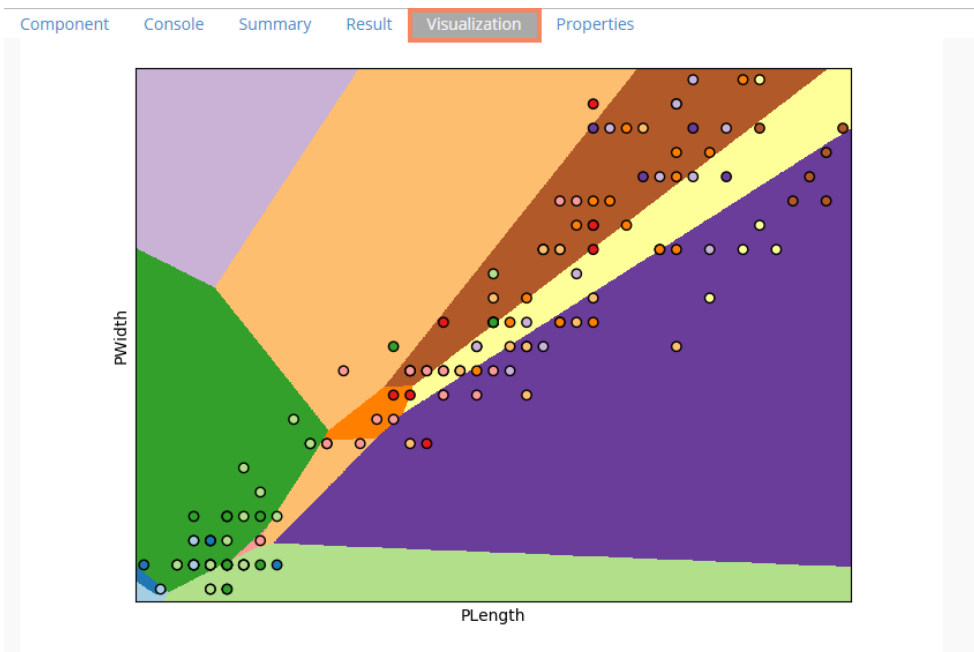
- v) Click 'Run'
- vi) Users will be redirected to the 'Console' tab.

- 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.

viii) A new column will be added to the result data.

RowID	SLength	SWidth	PLength	PWidth	PredictedValues1
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	4.8
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	4.8
9	4.4	2.9	1.4	0.2	5
10	4.9	3.1	1.5	0.1	4.8

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the Logistic Regression Classifier Chart.

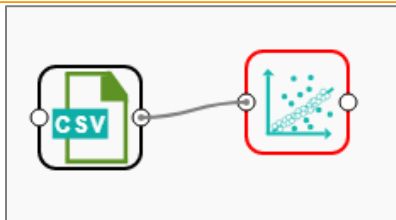


8.5. Outliers

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.

8.5.1. Interquartile Range

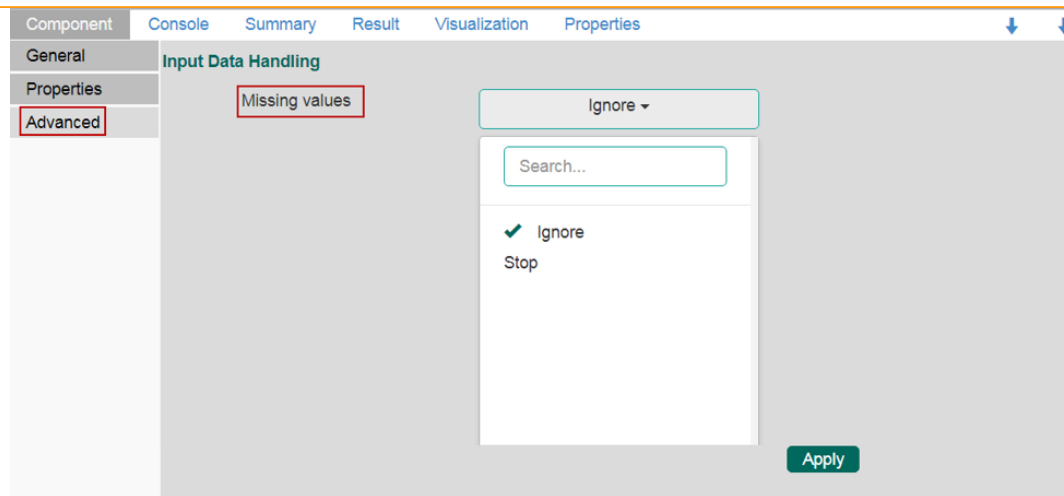
- i) Drag the Interquartile Range component to the workspace and connect it to a configured data source.



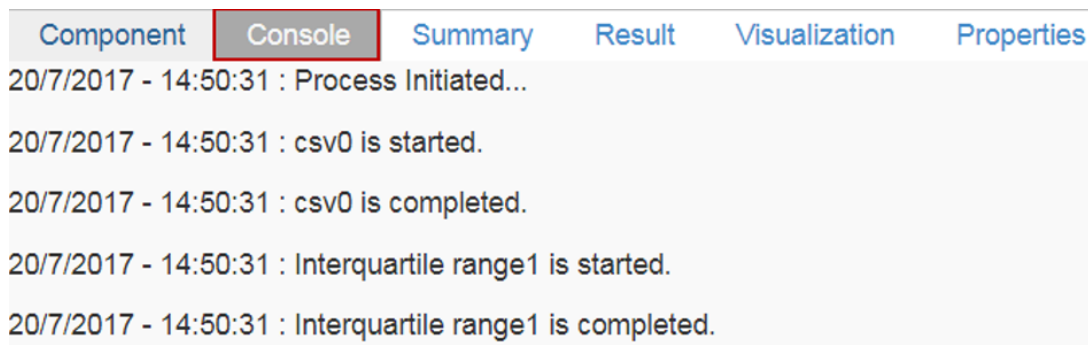
- ii) Configure the following fields in the ‘Properties’ tab:
 - a. **Output Information**
 - i. **Output Mode:** Select a mode of display for output data.
 1. **Show Outlier:** Selecting this option will add a Boolean column to the input data identifying whether the resultant value is an outlier.
 2. **Remove Outlier:** Selecting this option will 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**).

Component	Console	Summary	Result	Visualization	Properties
General	Output Information				
Properties	Output Mode	Show Outliers ▾			
Advanced	Column Selection				
	Feature	rivers ▾			<i>i</i>
	Behavior				
	Fence Coefficient	1.5			<i>i</i>
	New Column Information				
	New Column Name	OutliersDetected1			<i>i</i>
Apply					

- 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:** Selecting this option will skip the records containing missing values in the columns.
 2. **Stop:** Selecting this option will stop application of the algorithm if a value is missing in any column.



- iv) Click 'Apply'
- v) Click 'Run'
- vi) Users will be redirected to 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.
- viii) 'OutliersDetected' column will be displayed in the result data (If 'Show Outliers' option has been selected).

rivers	OutliersDetected1
735	FALSE
320	FALSE
325	FALSE
392	FALSE
524	FALSE
450	FALSE
1459	TRUE
135	FALSE
465	FALSE
600	FALSE

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the box plot chart.

OR

Outliers column will not be displayed in the result data (If 'Remove Outliers' option has been selected).

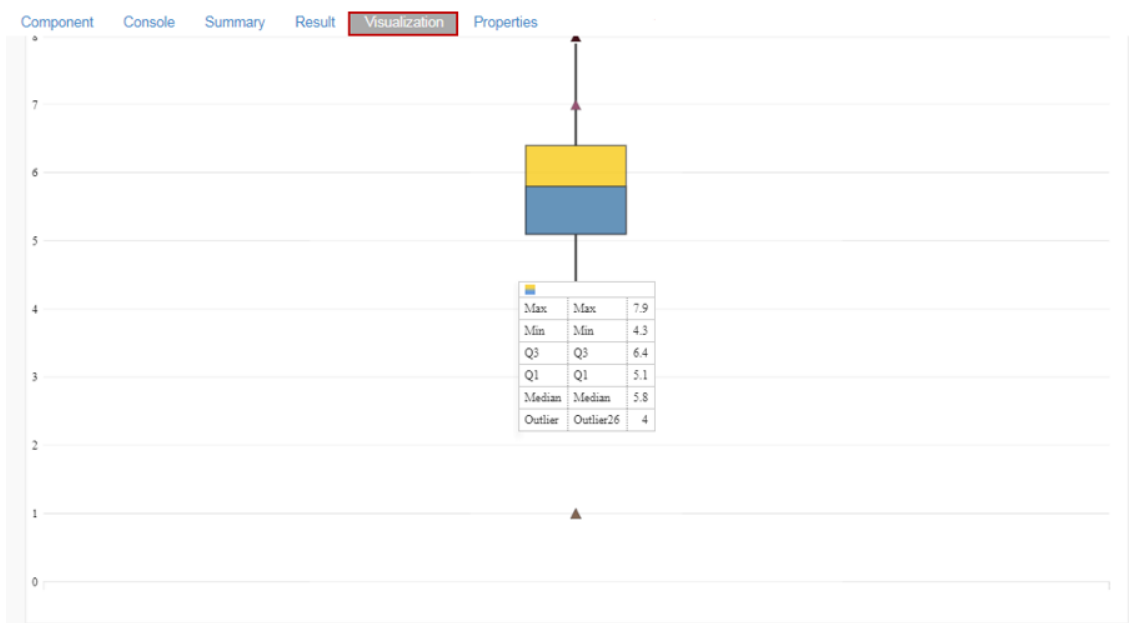
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

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

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

Click the 'Visualization' to see the result data via the box plot chart.



8.6. Classification

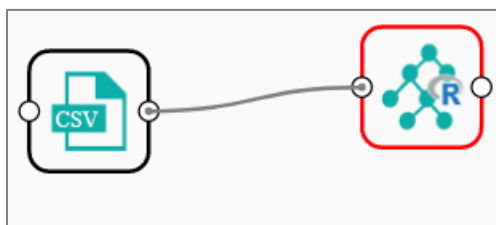
This algorithm categorizes a new observation by a trained set of data that contains observations from the known category. It compares each new observation to previous observations using means of similarity or distance.

8.6.1. 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:

8.6.1.1. 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 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.
 - 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:** Selecting this option will display a new column in the output data with probability values.
 - 2. **False:** Selecting this option will not 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 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	Algorithm Type				
Properties	Classification ▾				
Advanced	Show Probability				
Validation	True ▾				
Column Selection					
	Features	2 checked ▾			<i>i</i>
	Target Variable	SepalLength ▾			<i>i</i>
New Column Information					
	Predicted Column Name	PredictedValues1			<i>i</i>
	Probability Column Name	Probability1			<i>i</i>
	Enable Validation	<input checked="" type="checkbox"/>			
Apply					

Note: The ‘Show Probability’ field will appear 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 ‘Validation’ is disabled

a. Input Data Handling

- i. **Missing Values:** Select a method to deal with missing values from the drop-down list.
 1. **Rpart:** Selecting this option will try to estimate the missing values for the dependent column based on the independent columns.
 2. **Ignore:** Selecting this option will skip the records containing missing values in the columns.
 3. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.
 4. **Stop:** Selecting this option will stop application of the algorithm 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 the computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complexity parameter is pruned off performing cross-validation, hence the program will 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 prior 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:** Selecting this option will only display the observation, but not split it further.
 2. **Use Surrogate:** Selecting this option will search surrogate value for the missing values in order to split the observation. Two fields will be displayed:
 - a. **Surrogate Style:** Select a style using the drop-down menu.
 - b. **Maximum Surrogate:** Set the maximum surrogate value.

3. **Stop if missing:** Selecting this option will 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 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 will 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 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 <input type="text" value="Cross validation"/>				
Advanced	Number of folds <input type="text" value="3"/>				
Validation					
<input type="button" value="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 <input type="text" value="Bootstrap"/>				
Advanced	Number of resamples <input type="text" value="5"/>				
Validation					
<input type="button" value="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.’

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method <input type="text" value="Repeated cross validation"/>				
Advanced	Number of repeats <input type="text" value="5"/>				
Validation	Number of folds <input type="text" value="3"/>				
<input type="button" value="Apply"/>					

iv. **Leave One Out Cross Validation**

Users will not get any other field to configure if the selected model method is ‘Leave one out cross validation’.

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method <input type="text" value="Leave one out cross validation"/>				
Advanced					
Validation					
<input type="button" value="Apply"/>					

- v) Click ‘Apply’
- vi) Click ‘Run’
- vii) Users will be redirected to the ‘Console’ tab.


```

Component Console Summary Result Visualization Properties
20/7/2017 - 17:15:27 : Process Initiated...
20/7/2017 - 17:15:27 : csv0 is started.
20/7/2017 - 17:15:27 : csv0 is completed.
20/7/2017 - 17:15:27 : R-CNR Tree1 is started.
20/7/2017 - 17:15:28 : 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 'Validation' is disabled.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1	Probability1
5.1	3.5	1.4	0.2	setosa	5	0.3529412
4.9	3	1.4	0.2	setosa	4.4	0.3333333
4.7	3.2	1.3	0.2	setosa	4.7	0.4
4.6	3.1	1.5	0.2	setosa	4.9	0.375
5	3.6	1.4	0.2	setosa	5	0.3529412
5.4	3.9	1.7	0.4	setosa	5.4	0.3333333
4.6	3.4	1.4	0.3	setosa	5	0.3529412
5	3.4	1.5	0.2	setosa	5	0.3529412
4.4	2.9	1.4	0.2	setosa	4.4	0.3333333
4.9	3.1	1.5	0.1	setosa	4.9	0.375

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

- ii. Result view when 'Validation' is enabled.

Component Console Summary **Result** Visualization Properties

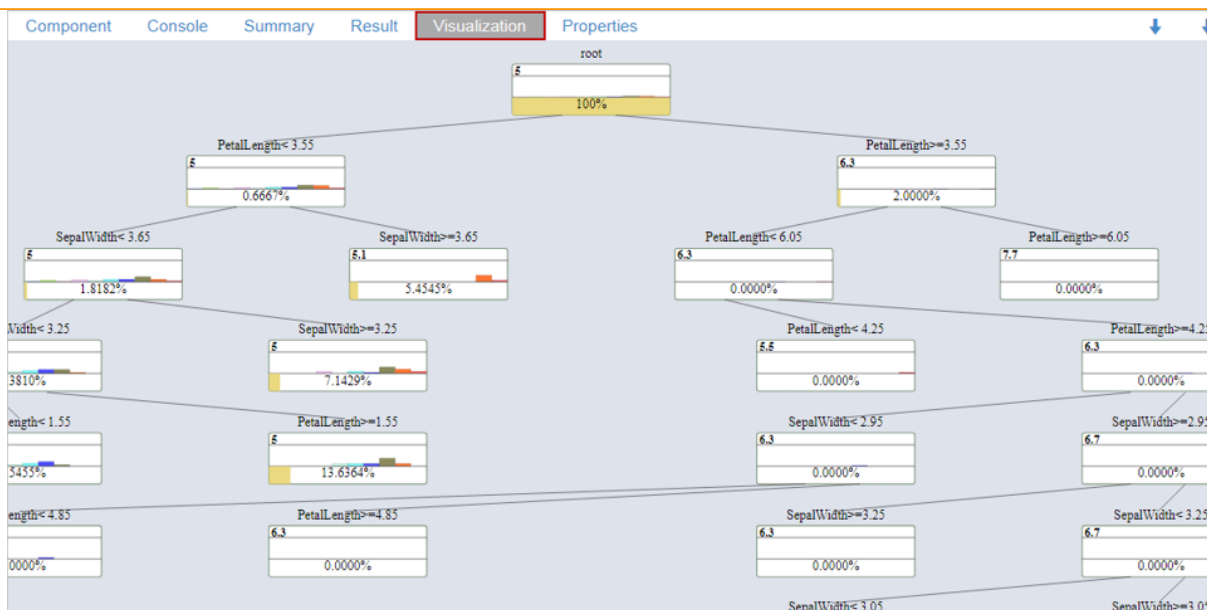
Show 10 entries Search:

SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1	Probability1
5.1	3.5	1.4	0.2	setosa	5	["0.0000000","0.0000000","0.0000000"]
4.9	3	1.4	0.2	setosa	4.4	["0.07142857","0.2142857","0.07142857"]
4.7	3.2	1.3	0.2	setosa	4.4	["0.07142857","0.2142857","0.07142857"]
4.6	3.1	1.5	0.2	setosa	4.4	["0.07142857","0.2142857","0.07142857"]
5	3.6	1.4	0.2	setosa	5	["0.0000000","0.0000000","0.0000000"]
5.4	3.9	1.7	0.4	setosa	5.1	["0.0000000","0.0000000","0.0000000"]
4.6	3.4	1.4	0.3	setosa	5	["0.0000000","0.0000000","0.0000000"]
5	3.4	1.5	0.2	setosa	5	["0.0000000","0.0000000","0.0000000"]
4.4	2.9	1.4	0.2	setosa	4.4	["0.07142857","0.2142857","0.07142857"]
4.9	3.1	1.5	0.1	setosa	4.4	["0.07142857","0.2142857","0.07142857"]

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

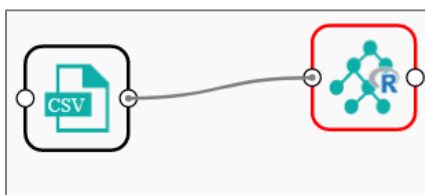
Note: The Probability column will be displayed in the Array format when Validation is enabled.

- ix) Click the 'Visualization' tab.
- x) The result data will be displayed via the tree chart.



8.6.1.2. 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	Output Information				
Properties	Algorithm Type		Regression ▾		
Advanced	Column Selection				
	Independent Columns		3 checked ▾		<i>i</i>
	Dependent Columns		PetalWidth ▾		<i>i</i>
	New Column Information				
	Predicted Column Name		PredictedValues1		<i>i</i>
	Enable Validation <input type="checkbox"/>				
					Apply

iii) 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 list.
 1. **Rpart:** Selecting this option will try to estimate the missing values for the dependent column based on the independent columns.
 2. **Ignore:** Selecting this option will skip the records containing missing values in the columns.
 3. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.
 4. **Stop:** Selecting this option will stop application of the algorithm 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 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 pruned off performing cross-validation, hence the program will 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:** Selecting this option will only display the observation, but not split it further.
 - 2. **Use Surrogate:** Selecting this option will search surrogate value for the missing values to split the observation. Two fields will be displayed:
 - a. **Surrogate Style:** Select a style using the drop-down menu.
 - b. **Maximum Surrogate:** Set the maximum surrogate value.
 - 3. **Stop if missing:** Selecting this option will choose an action based on the nature of majority observations. If values are missed for all the observations, then it will stop splitting further.

The screenshot shows a software interface with a sidebar on the left containing tabs: 'Component', 'General', 'Properties', and 'Advanced' (which is highlighted with a red box). The main area displays the 'Properties' configuration for a component. The tabs at the top are 'Console', 'Summary', 'Result', 'Visualization', and 'Properties'. The configuration is organized into sections: 'Input Data Handling' with a 'Missing values' field set to 'Rpart'; 'Tree Pruning' with 'Minimum Split' set to '10' and 'Maximum Depth' set to 'Optional'; 'Behavior' with 'Cross Validation' set to 'Optional'; and 'Surrogate Information' with a 'Use Surrogate' field set to '--select--'. An 'Apply' button is located at the bottom right of the configuration area.

• **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 pruned off performing cross-validation, hence the programme will not pursue it. The default value for this field is 0.05.

This screenshot shows the same software interface as above, but with the 'Validation' tab selected in the sidebar (highlighted with a red box). The main configuration area now shows the 'Tree Pruning' section with a 'Complexity Parameter' field set to '.005'. The 'Apply' button remains at the bottom right.

- 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 <input type="text" value="Cross validation"/>				
Advanced	Number of folds <input type="text" value="3"/>				
Validation					
<input type="button" value="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 <input type="text" value="Bootstrap"/>				
Advanced	Number of resamples <input type="text" value="5"/>				
Validation					
<input type="button" value="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’.

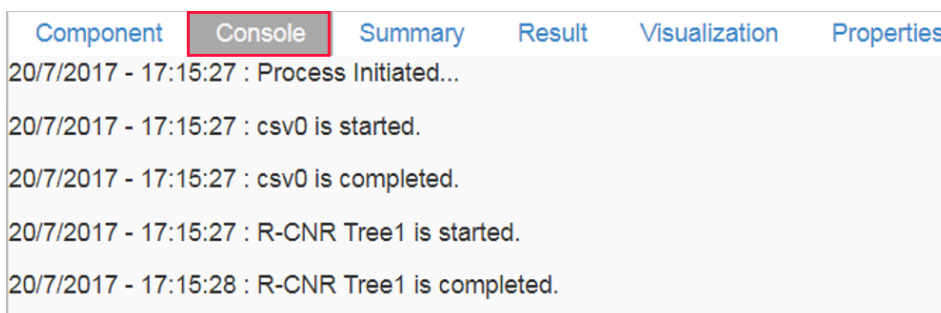
Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method <input type="text" value="Repeated cross validation"/>				
Advanced	Number of repeats <input type="text" value="5"/>				
Validation	Number of folds <input type="text" value="3"/>				
<input type="button" value="Apply"/>					

- iv. **Leave One Out Cross Validation**
Users will not get any other field to configure if the selected model method is ‘Leave one out cross validation’.

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method <input type="text" value="Leave one out cross validation"/>				
Advanced					
Validation					
<input type="button" value="Apply"/>					

- v) Click ‘Apply’
- vi) Click ‘Run’

vii) Users will be redirected to the 'Console' tab.



- viii) Follow the below given steps to display the result view:
- Click the dragged algorithm component onto the workspace.
 - Click the 'Result' tab.
 - Result View when 'Validation' is disabled.

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	0.246000
2	4.9	3	1.4	0.2	setosa	0.246000
3	4.7	3.2	1.3	0.2	setosa	0.246000
4	4.6	3.1	1.5	0.2	setosa	0.246000
5	5	3.6	1.4	0.2	setosa	0.246000
6	5.4	3.9	1.7	0.4	setosa	0.246000
7	4.6	3.4	1.4	0.3	setosa	0.246000
8	5	3.4	1.5	0.2	setosa	0.246000
9	4.4	2.9	1.4	0.2	setosa	0.246000
10	4.9	3.1	1.5	0.1	setosa	0.246000

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

ii. Result view when 'Validation' is enabled.

Component Console Summary Result Visualization Properties

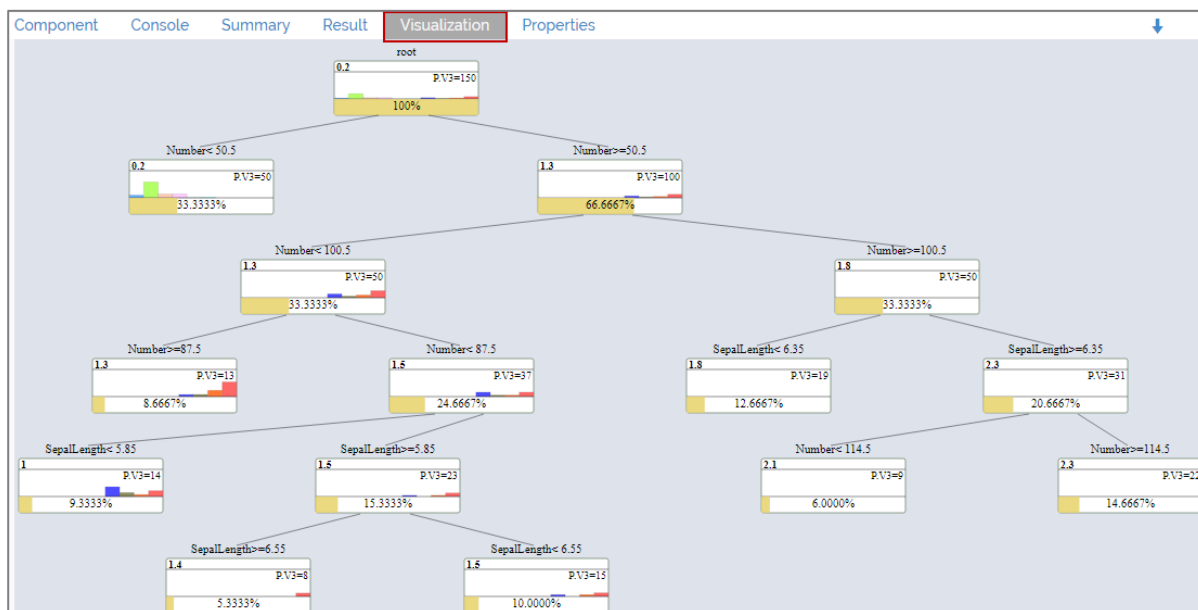
Show 10 entries Search: []

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

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

Note: The Probability column will be displayed in the Array format when Validation is enabled.

- Click the 'Visualization' tab.
- The result data will be displayed via the tree chart.

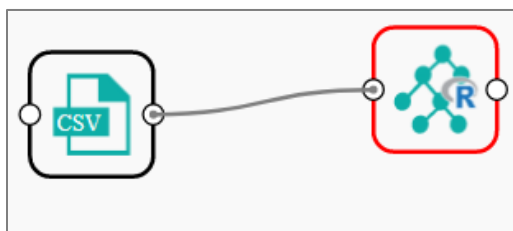


8.6.2. R-Naive Bayes

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 Naïve Bayes is as a leaf node under Classification algorithms under the Algorithm tree node. The component consists of one node for reading data from 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. **Validation:** Enable validation by a check mark in the given box.

Component	Console	Summary	Result	Visualization	Properties
General	Column Selection				
Properties	Feature	3 checked ▾		i	
Validation	TargetVariable	Species ▾		i	
Advanced	New Column Information				
	Predicted Column Name	PredictedValues1		i	
	Enable Validation	<input checked="" type="checkbox"/>			
Apply					

iii) Click the ‘**Validation**’ tab and configure it.

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.

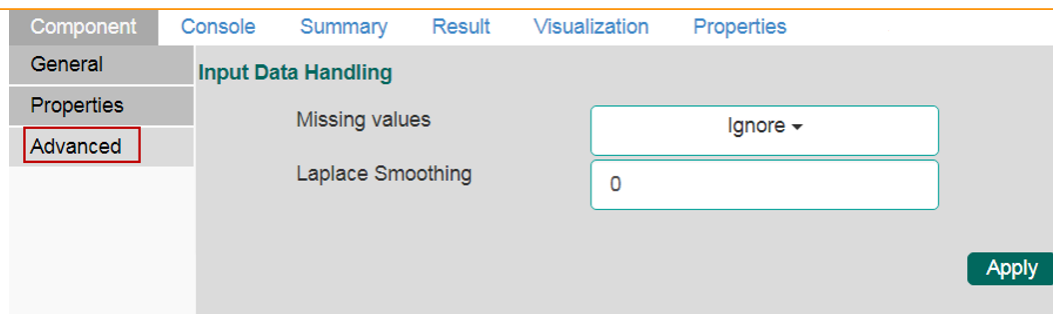
Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method	Cross validation ▾			
Validation	Number of folds	3			
Advanced					
Apply					

iv) Click the ‘**Advanced**’ tab and configure if required.

• **Advanced Tab when ‘Validation’ is Disabled:**

a. **Input Data Handling**

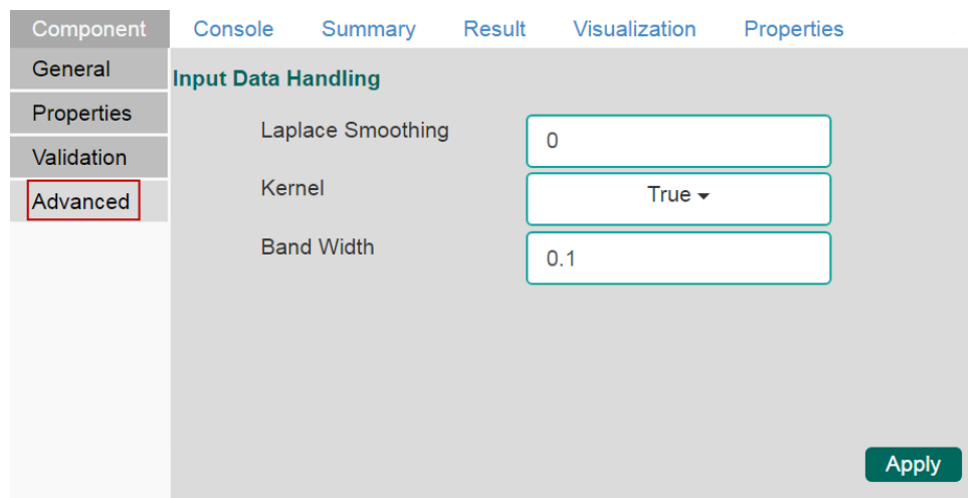
- i. **Missing Values:** Select a method to deal with missing values from the drop-down menu.
 1. **Ignore:** Selecting this option will skip the records containing missing values in the columns.
 2. **Keep:** Selecting this option will retain the records containing missing values while performing the 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.



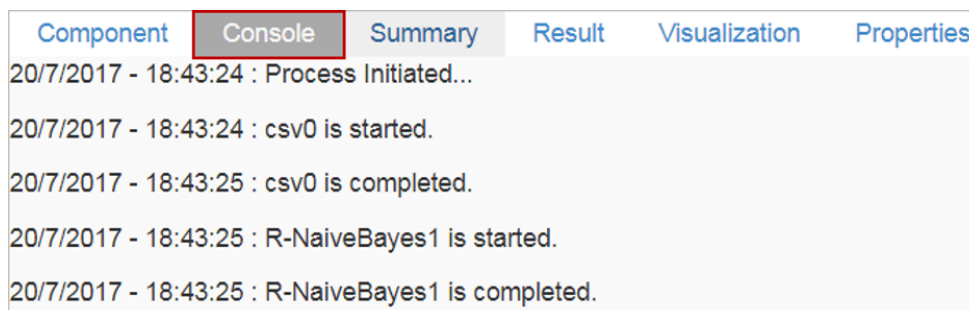
• **Advanced Tab when ‘Validation’ is Enabled:**

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 will disable 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 ‘**Apply**’
- vi) Click ‘**Run**’
- 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.

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	setosa
2	4.9	3	1.4	0.2	setosa	setosa
3	4.7	3.2	1.3	0.2	setosa	setosa
4	4.6	3.1	1.5	0.2	setosa	setosa
5	5	3.6	1.4	0.2	setosa	setosa
6	5.4	3.9	1.7	0.4	setosa	setosa
7	4.6	3.4	1.4	0.3	setosa	setosa
8	5	3.4	1.5	0.2	setosa	setosa
9	4.4	2.9	1.4	0.2	setosa	setosa
10	4.9	3.1	1.5	0.1	setosa	setosa

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

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 Model Selection Methods are described below:

i. **Cross-Validation**

Users need to configure the 'Number of folds' if 'Cross Validation' is the model selection

Component Console Summary **Result** Visualization Properties

General **Model Selection**

Properties

Advanced

Validation

Model Selection Method

Number of folds

Apply

ii. **Bootstrap**

Users need to configure the 'Number of resamples' if 'Bootstrap' is the model selection method

Component Console Summary **Result** Visualization Properties

General **Model Selection**

Properties

Advanced

Validation

Model Selection Method

Number of resamples

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'.

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method		Repeated cross validation ▾		
Advanced	Number of repeats		5		
Validation	Number of folds		3		
					Apply

iv. **Leave One Out Cross Validation**

Users will not get any other field to configure if the selected model method is 'Leave one out cross validation'.

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method		Leave one out cross validation ▾		
Advanced					
Validation					
					Apply

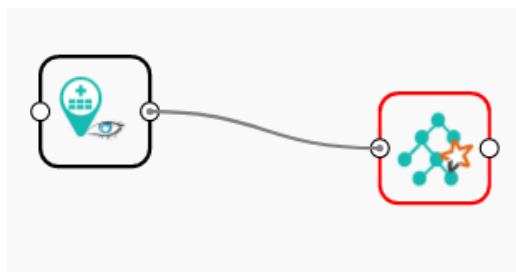
8.6.3. Spark-Naive Bayes

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.

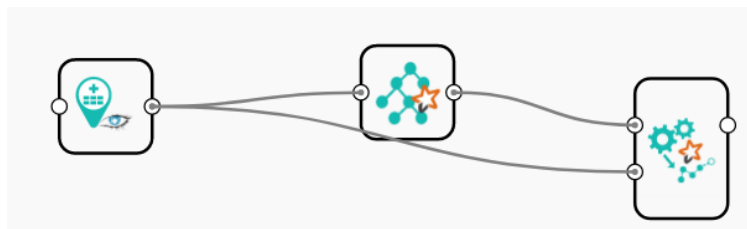
Spark Naive Bayes consists of two types of model selection methods:

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 source 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).

- **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 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.

- **Additive Smoothing:** Enter values between 0 and 1 where 1.0 is the default value.

- **Advanced Tab when ‘Validation’ is Enabled**

- iv) Click ‘Next’ (By enabling ‘Validation’ the ‘Apply’ option changes into ‘Next’).

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 data sets 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 separate training and test datasets. e.g., with k=3 folds, Cross Validator will generate 3 (training, test) data set 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.

- v) 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.

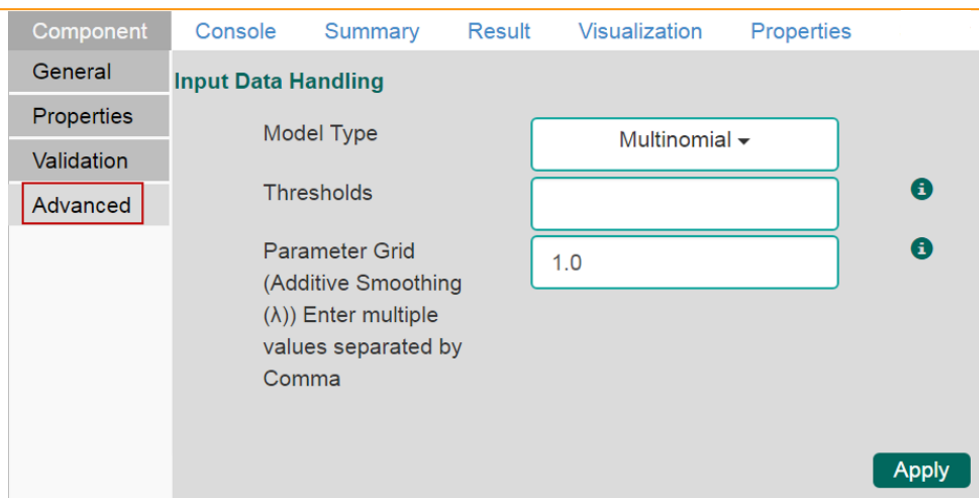
Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method				
Validation	Train validation ▾				
Advanced	Evaluator				
	Multi Class Classification ▾				
	Train Ratio				
	0.75				
					Apply

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)

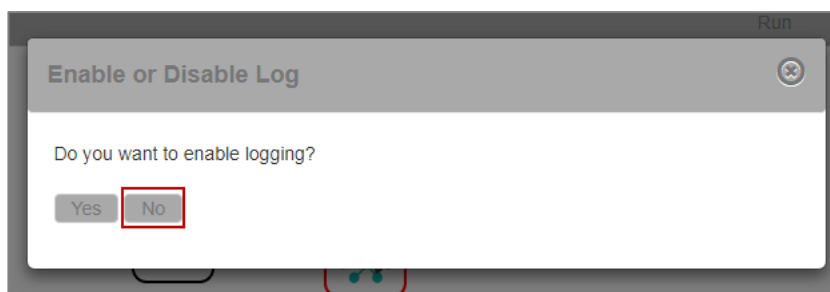
Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method				
Validation	Cross validation ▾				
Advanced	Evaluator				
	Multi Class Classification ▾				
	Number of folds				
	3				
					Apply

- vi) 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. Number of values entered as **the** threshold should be 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.
- vii) Click **'Apply'**

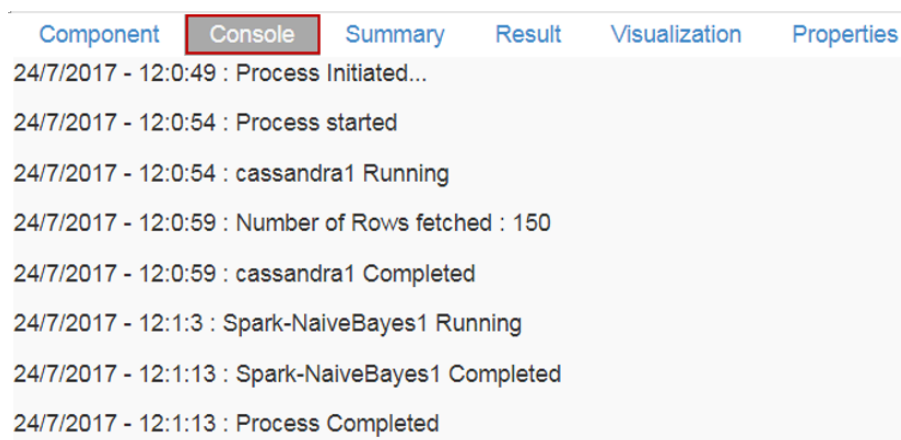


Note: If validation is enabled, users can enter multiple commas separated values in the Parameter Grid in the Advanced tab and they will be taken as paraMapS.

- viii) Configure the 'Apply Model' component and click 'Apply'
- ix) Click 'Run'
- x) A message will pop-up to confirm whether users want to enable logging.
- xi) Click 'No'



- xii) Users will be directed to the 'Console' tab.



- xiii) Follow the below given steps to display the result view:
 - a. Click the dragged algorithm component onto the workspace.
 - b. Click the 'Result' tab.

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	featuresCol1	rawPrediction1	probability1	binarycolumn	prediction1
4.9	1.8	6.3	2.7	virginica	['values': [4.9]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	0	0
1.7	0.2	5.4	3.4	setosa	['values': [1.7]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	1	0
1.4	0.2	5.1	3.5	setosa	['values': [1.4]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	1	0
1.5	0.4	5.7	4.4	setosa	['values': [1.5]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	1	0
1.9	0.2	4.8	3.4	setosa	['values': [1.9]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	1	0
4.7	1.5	6.7	3.1	versicolor	['values': [4.7]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	0	0
6.3	1.8	7.3	2.9	virginica	['values': [6.3]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	0	0
1.1	0.1	4.3	3	setosa	['values': [1.1]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	1	0
4.3	1.3	6.2	2.9	versicolor	['values': [4.3]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	0	0
5.1	1.9	5.8	2.7	virginica	['values': [5.1]]	['values': [-0.4087600040050168, -1.0920548881219507]]	['values': [0.6644736842105263, 0.3355263157894736]]	0	0

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.

Component Console **Summary** Result Visualization Properties

```

----- Summary of the model -----
Columns used as Feature:
Features
Label Column = Label
Number of Classes = 2
Number of Features = 30
Model Types = multinomial
Smoothing value = 1.0
----- End of Summary -----

```

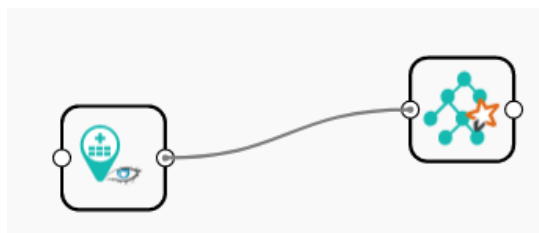
8.6.4. Spark Decision Tree

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.

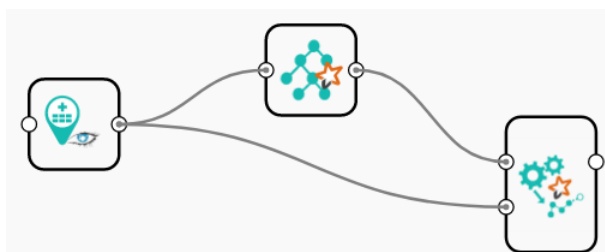
BizViz Predictive Analysis provides Spark Decision Tree under the Classification algorithm in the tree-node menu.

8.6.4.1. 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) Connect the Spark Decision Tree basic workflow with a configured ‘Spark Apply Model’ component to get the result view and evaluate the model performance.



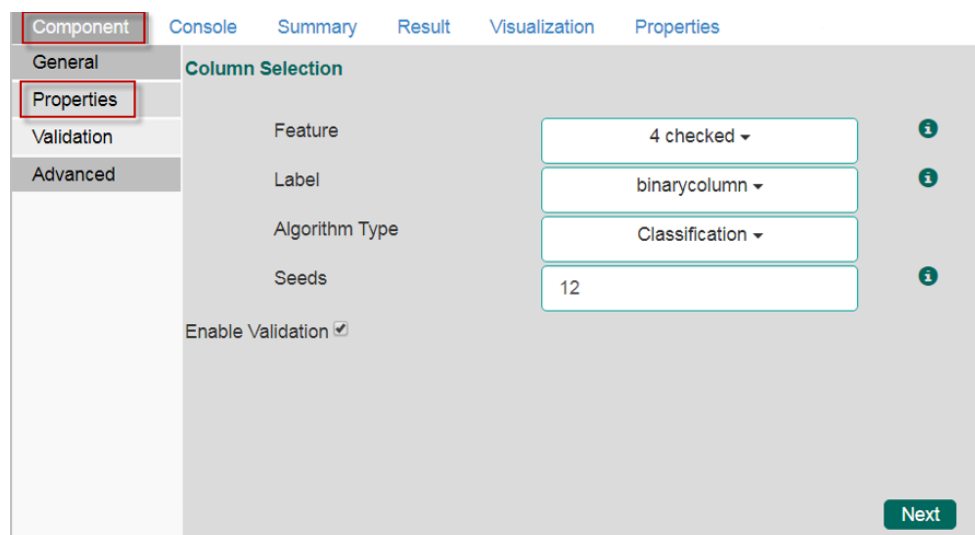
- iii) Configure the required fields for the algorithm component:

- **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).

- iv) 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 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.
- v) Click 'Next' (The 'Apply' option turns into 'Next' when 'Validation' is enabled).

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties					
Validation	Model Selection Method		Train validation ▾		
Advanced	Evaluator		Multi Class Classification ▾		
	Train Ratio		0.75		
Next					

- **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 ≥ 2 and \geq 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**

vi) Click 'Apply'

The screenshot shows the 'Properties' tab with the 'Advanced' sub-tab selected. Under the 'Column Selection' section, the 'Impurity' field is a dropdown menu currently set to 'gini'. Other fields include 'Maximum Depth' (5), 'Maximum Bins' (32), 'Minimum Instances Per Node' (1), and 'Minimum Info Gain' (0.0). An 'Apply' button is located at the bottom right.

vii) Configure the component tab for the 'Apply Model' and click 'Apply' option.

The screenshot shows the 'Properties' tab with the 'Basic' sub-tab selected. The 'Component Name' is 'Spark Apply Model', the 'Alias' is 'Spark Apply Model2', and the 'Description' is 'Optional'. An 'Apply' button is located at the bottom right.

viii) Click 'Run'

ix) A message will pop-up to confirm whether users want to enable logging.

x) Click 'No'

The screenshot shows a dialog box titled 'Enable or Disable Log' with the question 'Do you want to enable logging?'. There are two buttons: 'Yes' and 'No'. The 'No' button is highlighted with a red box.

Note: The 'Advanced' tab fields remain the same if 'Validation' is disabled.

Component Console Summary Result Visualization Properties

General Column Selection

Properties

Advanced

Maximum Depth

Maximum Bins

Minimum Instances Per Node

Minimum Info Gain

Thresholds

Impurity

Apply

xi) Users will be directed to the 'Console' tab.

Component Console Summary Result Visualization Properties

17/11/2017 - 18:30:7 : Process Initiated...

17/11/2017 - 18:30:9 : Number of Rows fetched : 150

17/11/2017 - 18:30:9 : cassandra1 Completed

17/11/2017 - 18:30:9 : Spark-Decision-Tree0 Running

17/11/2017 - 18:30:11 : Spark-Decision-Tree0 Completed

17/11/2017 - 18:30:11 : Spark Apply Model2 Running

17/11/2017 - 18:30:11 : Spark Apply Model2 Completed

17/11/2017 - 18:30:11 : Process Completed

xii) Follow the below given steps to display the result view:
 a. Click the 'Apply Model' component onto the workspace.
 b. Click the 'Result' tab.

Component Console Summary Result Visualization Properties

Show 10 entries

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	dfFeaturesCol0	rawPrediction0	probability0	binarycolumn	prediction0
4.9	1.8	6.3	2.7	virginica	{"values": [4.9, 1.8, 6.3, 2.7]}	{"values": [100, 0]}	{"values": [1, 0]}	0	0
1.7	0.2	5.4	3.4	setosa	{"values": [1.7, 0.2, 5.4, 3.4]}	{"values": [0, 50]}	{"values": [0, 1]}	1	1
1.4	0.2	5.1	3.5	setosa	{"values": [1.4, 0.2, 5.1, 3.5]}	{"values": [0, 50]}	{"values": [0, 1]}	1	1
1.5	0.4	5.7	4.4	setosa	{"values": [1.5, 0.4, 5.7, 4.4]}	{"values": [0, 50]}	{"values": [0, 1]}	1	1
1.9	0.2	4.8	3.4	setosa	{"values": [1.9, 0.2, 4.8, 3.4]}	{"values": [0, 50]}	{"values": [0, 1]}	1	1
4.7	1.5	6.7	3.1	versicolor	{"values": [4.7, 1.5, 6.7, 3.1]}	{"values": [100, 0]}	{"values": [1, 0]}	0	0
6.3	1.8	7.3	2.9	virginica	{"values": [6.3, 1.8, 7.3, 2.9]}	{"values": [100, 0]}	{"values": [1, 0]}	0	0
1.1	0.1	4.3	3	setosa	{"values": [1.1, 0.1, 4.3, 3]}	{"values": [0, 50]}	{"values": [0, 1]}	1	1
4.3	1.3	6.2	2.9	versicolor	{"values": [4.3, 1.3, 6.2, 2.9]}	{"values": [100, 0]}	{"values": [1, 0]}	0	0
5.1	1.9	5.8	2.7	virginica	{"values": [5.1, 1.9, 5.8, 2.7]}	{"values": [100, 0]}	{"values": [1, 0]}	0	0

Showing 1 to 10 of 150 entries

Previous 1 2 3 4 5 ... 15 Next

8.6.4.2. Regression as Algorithm Type

i) If the selected algorithm type is 'Regression' (from the 'Properties' tab)

Component	Console	Summary	Result	Visualization	Properties
General	Column Selection				
Properties	Feature	4 checked ▾		i	
Validation	Label	binarycolumn ▾		i	
Advanced	Algorithm Type	Regression ▾		i	
	Seeds	12		i	
	Enable Validation <input checked="" type="checkbox"/>				
					Next

- 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:
 4. **Multi-Class Classification** - If the dataset has multiple classes in the label column
 5. **Binary Class Classification**- if the data set has two classes in label Column
 6. **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).

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method	Cross validation ▾			
Validation	Evaluator	Regression Class Classification ▾			
Advanced	Number of folds	3			
					Next

- **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.)
 - i. **Maximum Bins:** Maximum number of bins for discretizing continuous features. (The value must be ≥ 2 and \geq number of categories for any categorical feature. (Type integer only. Default value 32.)

- ii. **Minimum Instances Per Node:** Minimum number of instances each child must have after the split is referred 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)
- iii. **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'

The screenshot shows a configuration window with tabs: Component, Console, Summary, Result, Visualization, and Properties. The 'Advanced' tab is selected and highlighted with a red box. Under the 'Column Selection' section, the following settings are visible:

Maximum Depth	5	<i>i</i>
Maximum Bins	32	<i>i</i>
Minimum Instances Per Node	1	<i>i</i>
Minimum Info Gain	0.0	<i>i</i>

An 'Apply' button is located at the bottom right of the configuration area.

v) Configure the component tab for the 'Apply Model' and click 'Apply'

The screenshot shows the same configuration window with the 'Basic' tab selected and highlighted with a red box. The following settings are visible:

Component Name	Spark Apply Model
Alias	Spark Apply Model2
Description	Optional

An 'Apply' button is located at the bottom right of the configuration area.

vi) Click 'Run'

vii) A message will pop-up to confirm whether users want to enable logging.

viii) Click 'No'

The screenshot shows a dialog box titled 'Enable or Disable Log' with a close button in the top right corner. The text inside the dialog asks: 'Do you want to enable logging?'. Below the text are two buttons: 'Yes' and 'No'. The 'No' button is highlighted with a red box.

ix) Users will be directed to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
20/11/2017 - 12:56:11 : Process Initiated...
20/11/2017 - 12:56:12 : Process started
20/11/2017 - 12:56:12 : cassandra1 Running
20/11/2017 - 12:56:17 : Number of Rows fetched : 150
20/11/2017 - 12:56:17 : cassandra1 Completed
20/11/2017 - 12:56:17 : Spark-Decision-Tree0 Running
20/11/2017 - 12:56:20 : Spark-Decision-Tree0 Completed
20/11/2017 - 12:56:20 : Spark Apply Model2 Running
20/11/2017 - 12:56:21 : Spark Apply Model2 Completed
20/11/2017 - 12:56:21 : 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

Show 10 entries Search:

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	dfFeaturesColo	binarycolumn	predictiono
4.9	1.8	6.3	2.7	virginica	["values":[4.9,1.8,6.3,2.7]]	0	0
1.7	0.2	5.4	3.4	setosa	["values":[1.7,0.2,5.4,3.4]]	1	1
1.4	0.2	5.1	3.5	setosa	["values":[1.4,0.2,5.1,3.5]]	1	1
1.5	0.4	5.7	4.4	setosa	["values":[1.5,0.4,5.7,4.4]]	1	1
1.9	0.2	4.8	3.4	setosa	["values":[1.9,0.2,4.8,3.4]]	1	1
4.7	1.5	6.7	3.1	versicolor	["values":[4.7,1.5,6.7,3.1]]	0	0
6.3	1.8	7.3	2.9	virginica	["values":[6.3,1.8,7.3,2.9]]	0	0
1.1	0.1	4.3	3	setosa	["values":[1.1,0.1,4.3,3]]	1	1
4.3	1.3	6.2	2.9	versicolor	["values":[4.3,1.3,6.2,2.9]]	0	0
5.1	1.9	5.8	2.7	virginica	["values":[5.1,1.9,5.8,2.7]]	0	0

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 only if the 'Apply Model' component contains summary to show.

```

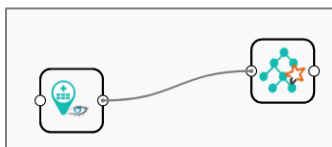
Component Console Summary Result Visualization Properties
----- Summary of the model -----
Columns used as Feature:
Features
Label Column = Label
Number of Classes = 2
Number of Features = 30
Model Types = multinomial
Smoothing value = 1.0
----- End of Summary -----
  
```

8.6.5. Spark Random Forest

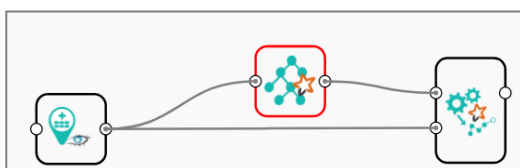
The Random Forest is a top performer tree ensemble algorithm for classification and regression tasks. The algorithm builds multiple decision trees based on different subsets of the features in the data. Outcomes are then predicted by running observations through all the trees and averaging the individual predictions.

8.6.5.1. 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'.

A screenshot of the configuration interface for the Spark Random Forest component. The 'Component' tab is selected in the top navigation bar. On the left, a sidebar shows 'Properties' selected. The main area is titled 'Column Selection' and contains the following configuration options:

- Feature:** A dropdown menu showing '4 checked'.
- Label:** A dropdown menu showing 'binarycolumn'.
- Algorithm Type:** A dropdown menu showing 'Classification'.
- Seeds:** A text input field containing the value '12'.
- Enable Validation:** A checkbox that is checked.

Information icons (i) are present next to the Feature, Label, and Seeds fields. A 'Next' button is located at the bottom right of the configuration area.

- **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:
 7. **Multi-Class Classification** - If the dataset has multiple classes in the label column
 8. **Binary Class Classification**- if the data set has two classes in label Column
 9. **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).

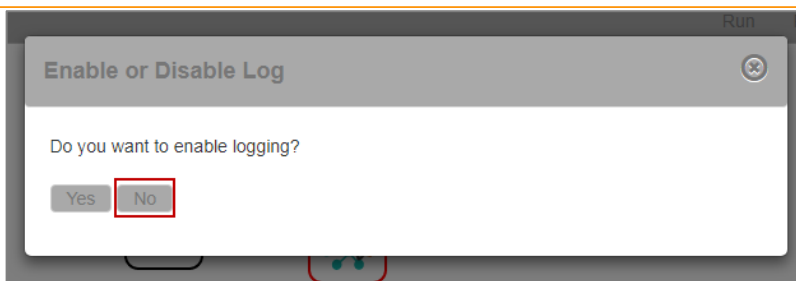
Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties					
Validation	Model Selection Method		Train validation ▾		
	Evaluator		Multi Class Classification ▾		
Advanced	Train Ratio		0.75		
Next					

- **Advanced**
 - b. **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 \geq 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 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)

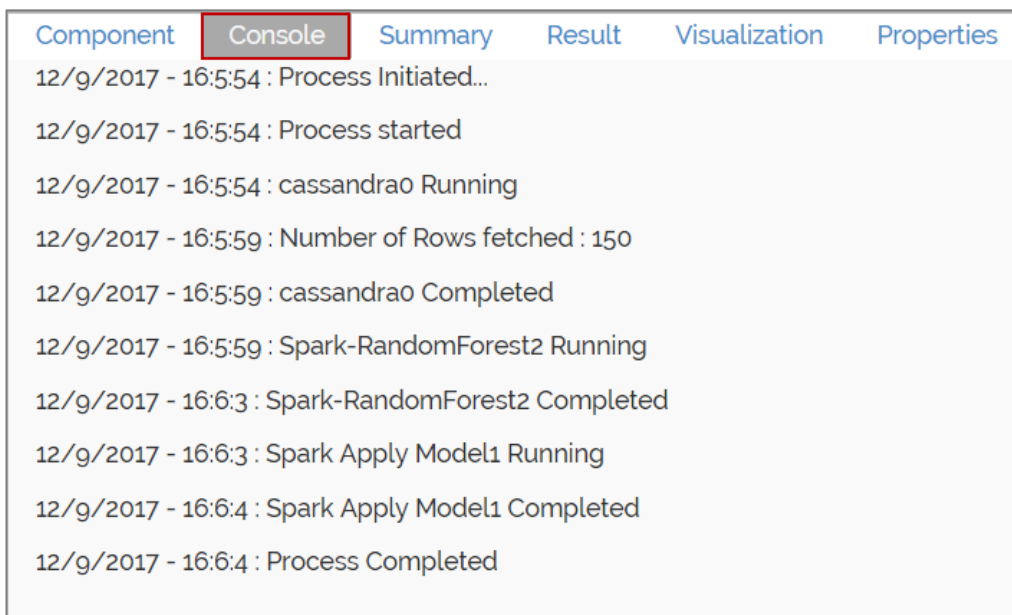
- v. **Minimum Info Gain:** Enter min. info. Gain for a split to be considered at a tree-node. (Type double only. Default value 0.0)
 - vi. **Number of Trees:** Enter number of trees to train (≥ 1).
 - vii. **Thresholds:** Thresholds in multiclass classification to adjust the probability of predicting 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) Click 'Run'
- ix) A message will pop-up to confirm whether users want to enable logging.
- x) Click 'No'



xi) Users will be directed to the 'Console' tab.



- xii) Follow the below given steps to display the result view:
- Click the dragged algorithm component onto the workspace.
 - Click the 'Result' tab.

Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	Species	rfFeaturesCol1	rawPrediction1	probability1	binarycolumn	prediction1
4.9	1.8	6.3	2.7	virginica	{"values": [4.9, 1.8, 6.3, 2.7]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
1.7	0.2	5.4	3.4	setosa	{"values": [1.7, 0.2, 5.4, 3.4]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
1.4	0.2	5.1	3.5	setosa	{"values": [1.4, 0.2, 5.1, 3.5]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
1.5	0.4	5.7	4.4	setosa	{"values": [1.5, 0.4, 5.7, 4.4]}	{"values": [1, 19]}	{"values": [0.05, 0.95]}	1	1
1.9	0.2	4.8	3.4	setosa	{"values": [1.9, 0.2, 4.8, 3.4]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
4.7	1.5	6.7	3.1	versicolor	{"values": [4.7, 1.5, 6.7, 3.1]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
6.3	1.8	7.3	2.9	virginica	{"values": [6.3, 1.8, 7.3, 2.9]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
1.1	0.1	4.3	3	setosa	{"values": [1.1, 0.1, 4.3, 3]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
4.3	1.3	6.2	2.9	versicolor	{"values": [4.3, 1.3, 6.2, 2.9]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
5.1	1.9	5.8	2.7	virginica	{"values": [5.1, 1.9, 5.8, 2.7]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0

Note: There is no change in the advanced tab or result when 'Validation' is disabled for Spark Random Forest with classification algorithm type.

8.6.5.2. Regression as Algorithm Type

- If the selected algorithm type is 'Regression' (from the 'Properties' tab)

Component	Console	Summary	Result	Visualization	Properties
General	Column Selection				
Properties	Feature	4 checked ▾		i	
Validation	Label	binarycolumn ▾		i	
Advanced	Algorithm Type	Regression ▾		i	
	Seeds	12		i	
	Enable Validation	<input checked="" type="checkbox"/>			
Next					

- **Validation**

- Model Selection Method:** Select any one validation method using the drop-down menu:

- Train Validation
- Cross-Validation

- Evaluator:** Select any one option using the drop-down menu to define evaluator. Evaluator consist of three types:

- Multi-Class Classification** - If the data set has multiple classes in the label column
- Binary Class Classification**- If the data set has two classes in label Column
- Regression Class Classification**-If the 'Label' column is continuous.

- Train Ratio:** This field will be displayed if train validation has been selected by using the 'Model Selection Method' field.

ii) Click 'Next'

Component	Console	Summary	Result	Visualization	Properties
General	Model Selection				
Properties	Model Selection Method	Train validation ▾			
Validation	Evaluator	Multi Class Classification ▾			
Advanced	Train Ratio	0.75			
Next					

- **Advanced**

a. Column Selection

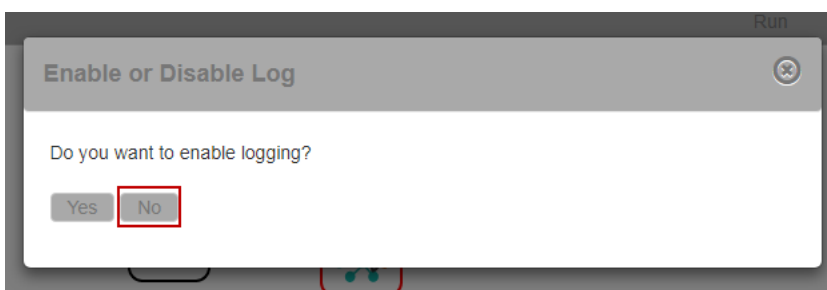
- i. **Feature Subset Strategy:** Select an option from the drop-down menu. The number of features to consider for splits at each tree-node (Supported options: auto, all, n, one-third, sqrt, log2).
 - ii. **Maximum Depth:** Maximum depth of the tree. (≥ 0) E.g., depth 0 means 1 leaf node; depth 1 means 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 \geq 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 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 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’

Component	Console	Summary	Result	Visualization	Properties
General	Column Selection				
Properties	Feature Subset Strategy	<input type="text" value="auto"/>			
Validation	Maximum Depth	<input type="text" value="5"/>			
Advanced	Maximum Bins	<input type="text" value="32"/>			
	minimum Instances Per Node	<input type="text" value="1"/>			
	Minimum Info Gain	<input type="text" value=".5"/>			
	Number of Trees	<input type="text" value="20"/>			
	Impurity	<input type="text" value="gini"/>			
	Sub Sampling rate	<input type="text" value="1"/>			
					<input type="button" value="Apply"/>

- iv) Configure the ‘Apply Model’ component and click ‘Apply’

Component	Console	Summary	Result	Visualization	Properties
General	Basic				
	Component Name	<input type="text" value="Spark Apply Model"/>			
	Alias	<input type="text" value="Spark Apply Model2"/>			
	Description	<input type="text" value="Optional"/>			
					<input type="button" value="Apply"/>

- v) A message pops-up to assure successful apply.
- vi) Click 'Run'
- vii) A message will pop-up to confirm whether users want to enable logging.
- viii) Click 'No'



- ix) Users will be directed to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
12/9/2017 - 16:5:54 : Process Initiated...					
12/9/2017 - 16:5:54 : Process started					
12/9/2017 - 16:5:54 : cassandra0 Running					
12/9/2017 - 16:5:59 : Number of Rows fetched : 150					
12/9/2017 - 16:5:59 : cassandra0 Completed					
12/9/2017 - 16:5:59 : Spark-RandomForest2 Running					
12/9/2017 - 16:6:3 : Spark-RandomForest2 Completed					
12/9/2017 - 16:6:3 : Spark Apply Model1 Running					
12/9/2017 - 16:6:4 : Spark Apply Model1 Completed					
12/9/2017 - 16:6:4 : 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

Show 10 entries Search:

Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	Species	rfFeaturesCol1	binarycolumn	prediction1
4.9	1.8	6.3	2.7	virginica	["values": [4.9, 1.8, 6.3, 2.7]]	0	0
1.7	0.2	5.4	3.4	setosa	["values": [1.7, 0.2, 5.4, 3.4]]	1	1
1.4	0.2	5.1	3.5	setosa	["values": [1.4, 0.2, 5.1, 3.5]]	1	1
1.5	0.4	5.7	4.4	setosa	["values": [1.5, 0.4, 5.7, 4.4]]	1	0.95
1.9	0.2	4.8	3.4	setosa	["values": [1.9, 0.2, 4.8, 3.4]]	1	1
4.7	1.5	6.7	3.1	versicolor	["values": [4.7, 1.5, 6.7, 3.1]]	0	0
6.3	1.8	7.3	2.9	virginica	["values": [6.3, 1.8, 7.3, 2.9]]	0	0
1.1	0.1	4.3	3	setosa	["values": [1.1, 0.1, 4.3, 3]]	1	1
4.3	1.3	6.2	2.9	versicolor	["values": [4.3, 1.3, 6.2, 2.9]]	0	0
5.1	1.9	5.8	2.7	virginica	["values": [5.1, 1.9, 5.8, 2.7]]	0	0

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.

Component Console **Summary** Result Visualization Properties

```

----- Summary of the model -----
Columns used as Feature:
Features
Label Column = Label
Number of Classes = 2
Number of Features = 30
Model Types = multinomial
Smoothing value = 1.0
----- End of Summary -----

```

8.7. 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.

8.7.1. 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'

General	Column Selection	
Properties	Input columns	2 checked ▾
	Method	1 checked ▾
	Missing value method	1 checked ▾
		Apply

- iv) Click 'Run'
- v) Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

24/7/2017 - 16:26:11 : Process Initiated...

24/7/2017 - 16:26:12 : csv0 is started.

24/7/2017 - 16:26:12 : csv0 is completed.

24/7/2017 - 16:26:12 : R-Correlation1 is started.

24/7/2017 - 16:26:14 : R-Correlation1 is completed.

- 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 will be added to the result data.

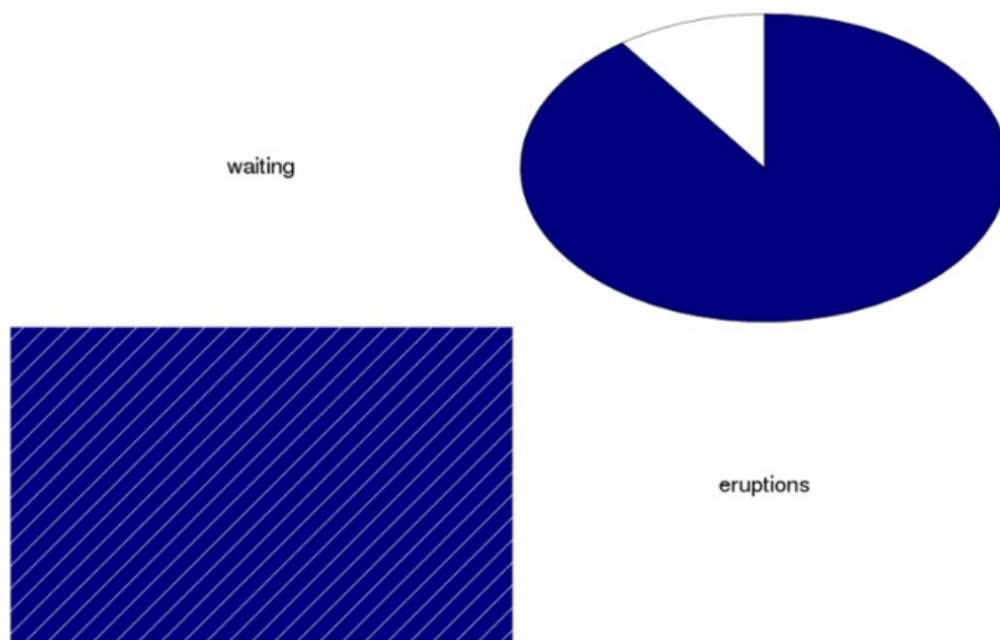
Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

category	eruptions	waiting
eruptions	1	0.900811168321813
waiting	0.900811168321813	1

Showing 1 to 2 of 2 entries Previous 1 Next

- viii) Click the 'Visualization' tab.
- ix) The probable values of the selected columns will be displayed via the Correlogram Chart.



8.8. Recommendation Engine

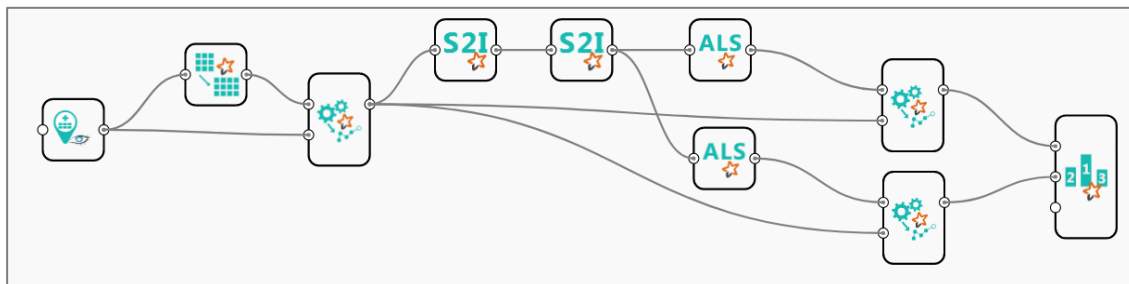
The Recommendation Engine algorithm helps to build a prediction model. The algorithm will consider the known user-item association as training data. The Training data is then used to predict the unknown set of data at Test data.

8.8.1. Spark ALS

The Spark ALS (Alternating Least Squares) can be used to make basic 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:

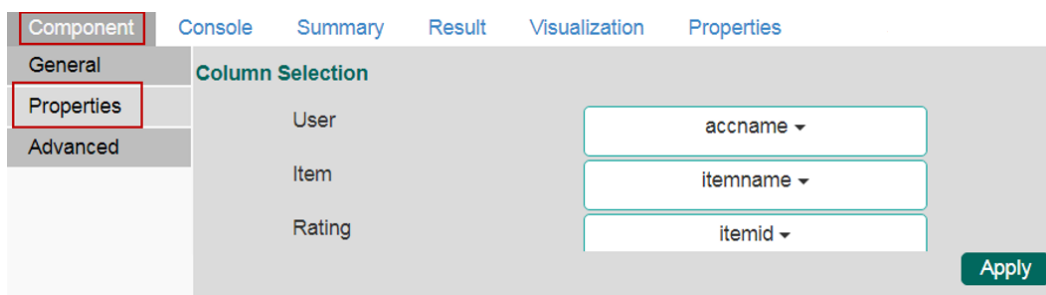


- ii) Configure the following fields in the 'Properties' tab:

- a. **Column Selection**

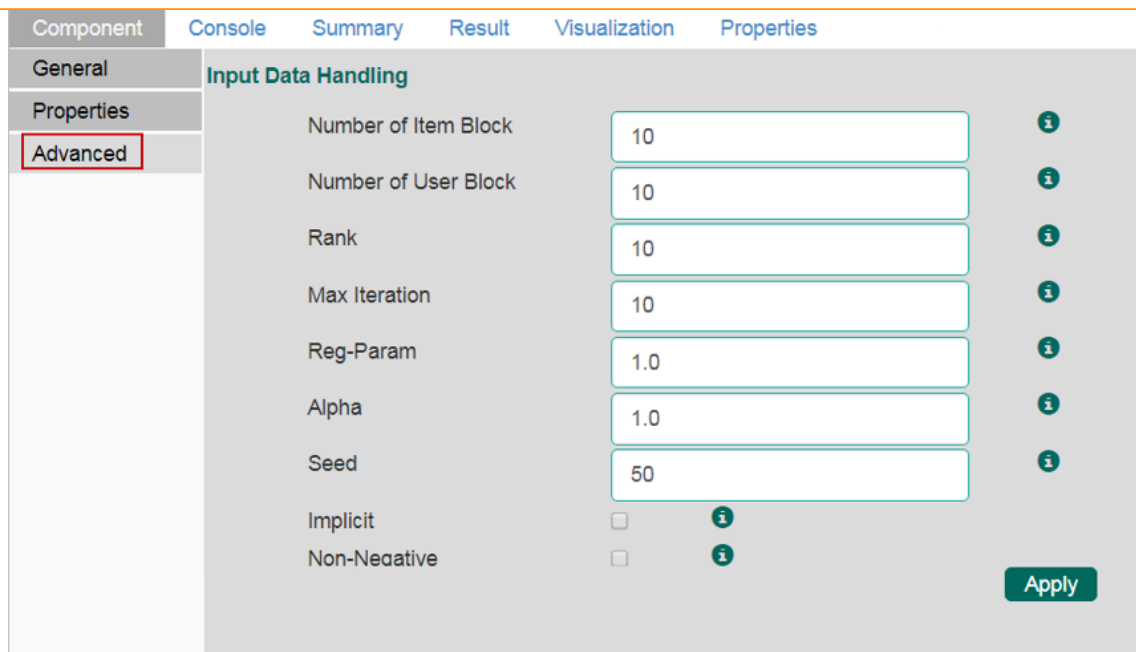
- i. **User:** Select a user column from the drop-down menu.

- ii. **Item:** Select an item column from the drop-down menu.
- iii. **Rating:** Select a rating column from the drop-down menu.
- iii) Click **'Apply'** (If you do not require to configure **'Advanced'** tab. Else, configure the **'Advanced'** tab).

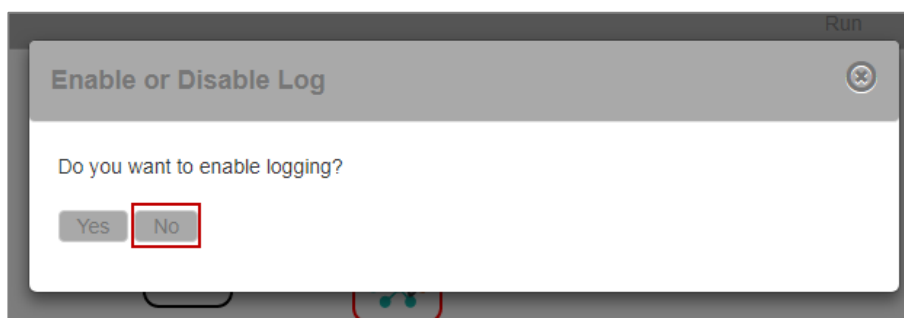


- iv) Configure the required **'Advanced'** information:
 - a. **Input Data Handling**
 - i. **Number of Item Block:** Items will be partitioned as per the entered the number of item block to parallelize computation (default value is 10).
 - ii. **Number of User Block:** Users will be partitioned as per the entered number of user block to parallelize computation (default value is 10).
 - iii. **Rank:** This refers to the number of factors in ALS model, that is the number of hidden features in our low-rank approximation matrices.

Generally, the greater 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 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:** Select **'Non-Negative'** to use nonnegative constraints for least squares (Default value is **'False'**).
- v) Click **'Apply'**



- vi) Configure all the required components to create a workflow and Click 'Run' option.
- vii) A message will pop-up to confirm whether users want to enable logging.
- viii) Click 'No'



- ix) Users will be directed to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
25/7/2017 - 14:31:0 : Process Initiated...					
25/7/2017 - 14:31:0 : Process started					
25/7/2017 - 14:31:0 : cassandra0 Running					
25/7/2017 - 14:31:5 : Number of Rows fetched : 14861					
25/7/2017 - 14:31:5 : cassandra0 Completed					
25/7/2017 - 14:31:5 : SQL Transformer1 Running					
25/7/2017 - 14:31:5 : SQL Transformer1 Completed					
25/7/2017 - 14:31:5 : Spark Apply Model2 Running					
25/7/2017 - 14:31:9 : Spark Apply Model2 Completed					
25/7/2017 - 14:31:9 : Spark String indexer3 Running					
25/7/2017 - 14:31:9 : Spark String indexer3 Completed					
25/7/2017 - 14:31:9 : Spark String indexer4 Running					
25/7/2017 - 14:31:9 : Spark String indexer4 Completed					
25/7/2017 - 14:31:9 : Spark-ALS5 Running					
25/7/2017 - 14:31:11 : Spark-ALS5 Completed					
25/7/2017 - 14:31:11 : Spark-ALS8 Running					
25/7/2017 - 14:31:14 : Spark-ALS8 Completed					
25/7/2017 - 14:31:14 : Spark Apply Model6 Running					
25/7/2017 - 14:31:14 : Spark Apply Model6 Completed					
25/7/2017 - 14:31:14 : Spark Apply Model9 Running					
25/7/2017 - 14:31:15 : Spark Apply Model9 Completed					
25/7/2017 - 14:31:15 : Spark-Performance7 Running					
25/7/2017 - 14:31:15 : Spark-Performance7 Completed					
25/7/2017 - 14:31:15 : Process Completed					

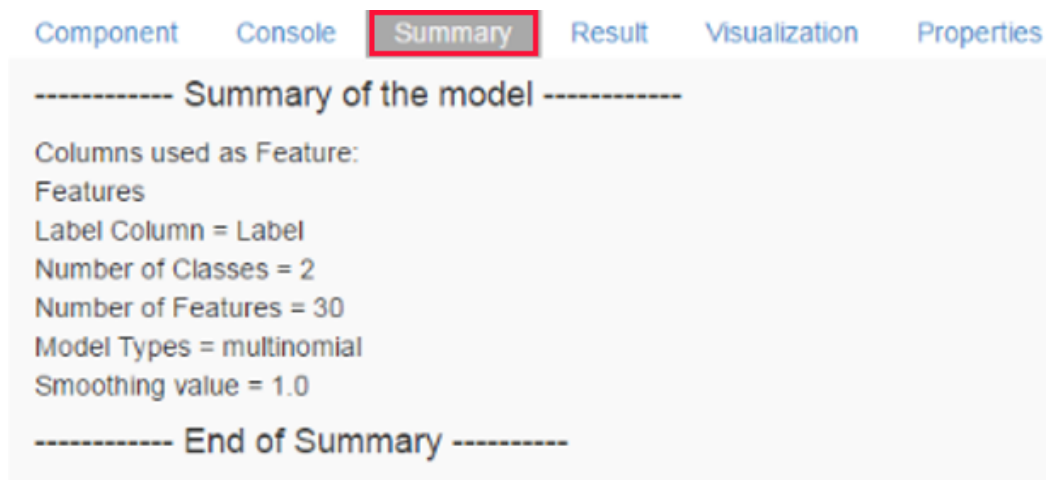
- x) Follow the below given steps to display the result view:
 - a. Click the dragged algorithm component onto the workspace.
 - b. Click the 'Result' tab.
- xi) A new column will be added to the 'Result' view.

Component	Console	Summary	Result	Visualization	Properties
Show 10 entries					
accname	itemname	rating	user	item	prediction5
william whitner	Smoothie - Very Berry 12oz	3	46	14	2.0926957
Monique Mills	Smoothie - Very Berry 12oz	1	233	14	0.3255447
Sanaa Montoya	Smoothie - Very Berry 12oz	1	282	14	0.76831234
Shele Lieberman	Smoothie - Very Berry 12oz	1	1030	14	0.20940614
Courtney Noce	Smoothie - Very Berry 12oz	1	6	14	1.5831974
Rafat Bello	Smoothie - Very Berry 12oz	3	104	14	0.6935842
Kevin Ponton	Smoothie - Very Berry 12oz	1	379	14	1.2290695
Liz Peck	Smoothie - Very Berry 12oz	2	81	14	0.5231458
Ayanna HillGill	Smoothie - Very Berry 12oz	2	445	14	1.6731519
Diana Ortiz	Smoothie - Very Berry 12oz	2	513	14	0.59379417
Showing 1 to 10 of 2,796 entries					
Previous 1 2 3 4 5 ... 280 Next					

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.



9. Apply Model

9.1. 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.

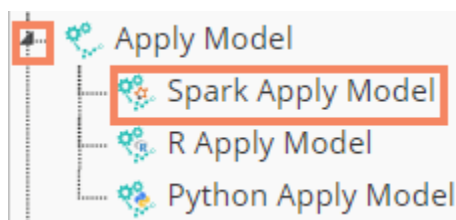
Users can create a model via the following ways:

- Generate a model using an algorithm
- Generate a model using the saved models

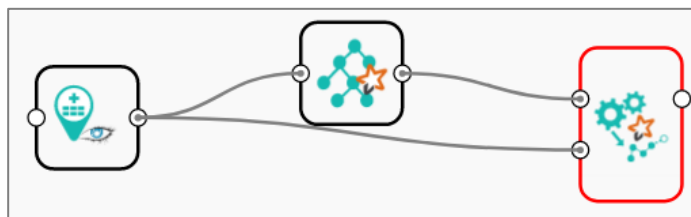
The Spark Apply Model consists of 2 input nodes and 1 output node.

- **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.



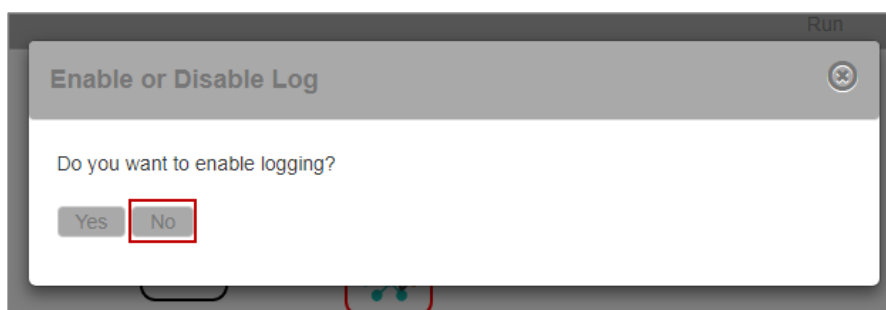
- iii) Drag the Spark Apply Model component onto the workspace and connect it with a valid combination of Data source and algorithm (Configure the data source and algorithm components. In this case, the used algorithm is Spark Decision Tree)
- iv) Click ‘**Spark Apply Model**’ component.



- v) Basic component details will be displayed.
- vi) Click 'Apply'

Component	Console	Summary	Result	Visualization	Properties
General	Basic				
	Component Name	<input type="text" value="Spark Apply Model"/>			
	Alias	<input type="text" value="Spark Apply Model2"/>			
	Description	<input type="text" value="Optional"/>			
					<input type="button" value="Apply"/>

- vii) Click 'Run'
- viii) A message will pop-up to confirm whether users want to enable logging.
- ix) Click 'No'



- x) Users will be redirected to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
25/7/2017 - 16:23:56	: Process Initiated...				
25/7/2017 - 16:23:56	: Process started				
25/7/2017 - 16:23:56	: cassandra1 Running				
25/7/2017 - 16:24:1	: Number of Rows fetched : 150				
25/7/2017 - 16:24:1	: cassandra1 Completed				
25/7/2017 - 16:24:1	: Spark-Decission-Tree0 Running				
25/7/2017 - 16:24:2	: Spark-Decission-Tree0 Completed				
25/7/2017 - 16:24:2	: Spark Apply Model2 Running				
25/7/2017 - 16:24:2	: Spark Apply Model2 Completed				
25/7/2017 - 16:24:2	: Process Completed				

- xi) Follow the below given steps to display the result view:
- Click the dragged Spark Apply Model component on the workspace.
 - Click the 'Result' tab.

PetalLength	PetalWidth	SepalLength	SepalWidth	Species	rfFeaturesCol1	rawPrediction1	probability1	binarycolumn	prediction1
4.9	1.8	6.3	2.7	virginica	{"values": [4.9, 1.8, 6.3, 2.7]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
1.7	0.2	5.4	3.4	setosa	{"values": [1.7, 0.2, 5.4, 3.4]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
1.4	0.2	5.1	3.5	setosa	{"values": [1.4, 0.2, 5.1, 3.5]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
1.5	0.4	5.7	4.4	setosa	{"values": [1.5, 0.4, 5.7, 4.4]}	{"values": [1, 19]}	{"values": [0.05, 0.95]}	1	1
1.9	0.2	4.8	3.4	setosa	{"values": [1.9, 0.2, 4.8, 3.4]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
4.7	1.5	6.7	3.1	versicolor	{"values": [4.7, 1.5, 6.7, 3.1]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
6.3	1.8	7.3	2.9	virginica	{"values": [6.3, 1.8, 7.3, 2.9]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
1.1	0.1	4.3	3	setosa	{"values": [1.1, 0.1, 4.3, 3]}	{"values": [0, 20]}	{"values": [0, 1]}	1	1
4.3	1.3	6.2	2.9	versicolor	{"values": [4.3, 1.3, 6.2, 2.9]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0
5.1	1.9	5.8	2.7	virginica	{"values": [5.1, 1.9, 5.8, 2.7]}	{"values": [20, 0]}	{"values": [1, 0]}	0	0

- xii) Click the 'Properties' tab to view the properties details (This Properties tab display workflow properties).

Component	Console	Summary	Result	Visualization	Properties
Created By					
Created At					2017-07-18 17:16:46 +0530
Last Modified By					
Last Modified At					2017-07-24 13:15:59 +0530
Version					3.0.0

Note:

- The result data set of the model can be written to a database using the Cassandra Writer.
- Column header and data type of feature column for both 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.

9.2. 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.

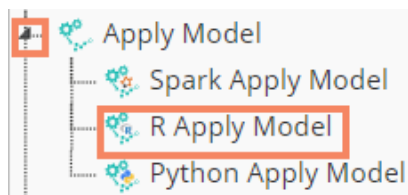
Users can create a model via the following ways:

- Generate a model using an algorithm
- Generate a model using the saved models

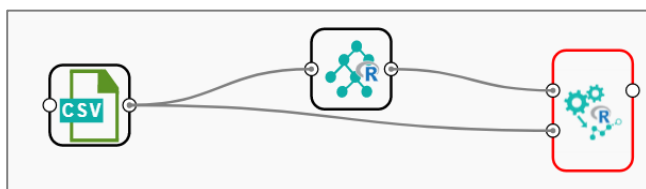
The R Apply Model consists of 2 input nodes and 1 output node.

- **Input Nodes**
 - Upper node - Model/Training data
 - Lower node - Testing data
- **Output Node**
 - Node - Result data

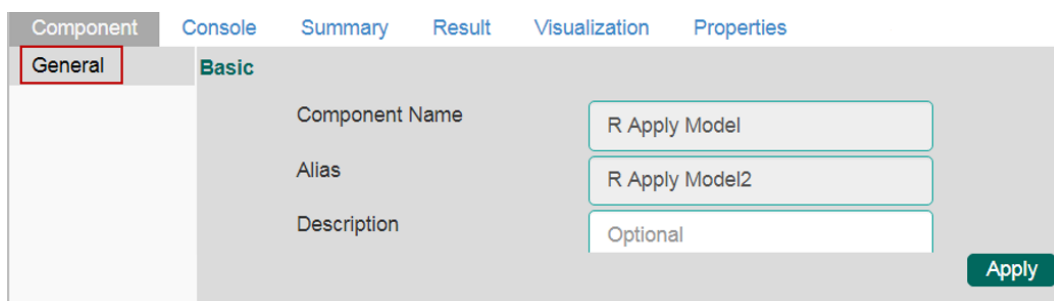
- Click the 'Apply Model' tree-node.
- The 'R Apply Model' leaf-node will be displayed.



- 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.)
- Click 'R Apply Model' component.



- Basic component details will be displayed.
- Click 'Apply'



- Click 'Run'
- Users will be redirected to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
25/7/2017 - 17:7:16 : Process Initiated...
25/7/2017 - 17:7:16 : csv0 is started.
25/7/2017 - 17:7:16 : csv0 is completed.
25/7/2017 - 17:7:17 : R-CNR Tree1 is started.
25/7/2017 - 17:7:17 : R-CNR Tree1 is completed.
25/7/2017 - 17:7:17 : R Apply Model2 is started.
25/7/2017 - 17:7:17 : R Apply Model2 is completed.
  
```

- ix) 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.
- x) The columns displaying Predicted values and probability will be added to the result view.

SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1	Probability1
5.1	3.5	1.4	0.2	setosa	setosa	1
4.9	3	1.4	0.2	setosa	setosa	1
4.7	3.2	1.3	0.2	setosa	setosa	1
4.6	3.1	1.5	0.2	setosa	setosa	1
5	3.6	1.4	0.2	setosa	setosa	1
5.4	3.9	1.7	0.4	setosa	setosa	1
4.6	3.4	1.4	0.3	setosa	setosa	1
5	3.4	1.5	0.2	setosa	setosa	1
4.4	2.9	1.4	0.2	setosa	setosa	1
4.9	3.1	1.5	0.1	setosa	setosa	1

- xi) Click the 'Summary' tab to view the model summary.

```

Component Console Summary Result Visualization Properties Status
----- Summary of All Stages -----
----- Summary of stage 1 -----
n= 150

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
2) PetalLength< 2.45 50 0 setosa (1.00000000 0.00000000 0.00000000) *
3) PetalLength>=2.45 100 50 versicolor (0.00000000 0.50000000 0.50000000)
6) PetalWidth< 1.75 54 5 versicolor (0.00000000 0.90740741 0.09259259)
12) PetalLength< 4.95 48 1 versicolor (0.00000000 0.97916667 0.02083333) *
13) PetalLength>=4.95 6 2 virginica (0.00000000 0.33333333 0.66666667) *
7) PetalWidth>=1.75 46 1 virginica (0.00000000 0.02173913 0.97826087) *

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

Note:

- a. The result data set of the model can be written to a database using a Data Writer.
- b. Column header and data type of feature column for both 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.

9.3. 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

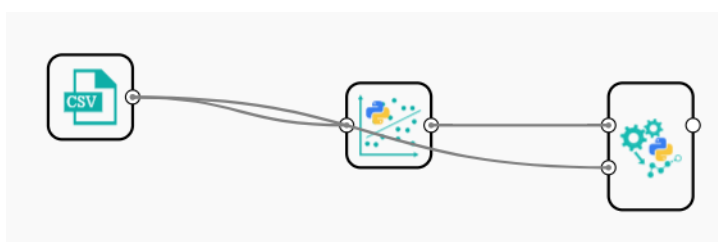
The R Apply Model consists of 2 input nodes and 1 output node.

- **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 '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 'Python Apply Model' component.



- v) Basic component details will be displayed.
- vi) Click 'Apply'

Component Console Summary Result Visualization Properties

General Basic

Component Name

Alias

Description

Apply

- vii) Click 'Run'
- viii) Users will be redirected to the 'Console' tab.

Component Console Summary Result Visualization Properties

2/4/2018 - 20:10:26 : Process Initiated...

2/4/2018 - 20:10:29 : CSV1 is started.

2/4/2018 - 20:10:29 : CSV1 is completed.

2/4/2018 - 20:10:29 : Python-Logistic Regression1 is started.

2/4/2018 - 20:10:32 : Python-Logistic Regression1 is completed.

2/4/2018 - 20:10:32 : Python Apply Model2 is started.

2/4/2018 - 20:10:32 : Python Apply Model2 is completed.

- ix) Follow the below given steps to display the result view:
 - c. Click the dragged Python Apply Model component on the workspace.
 - d. 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:

RowID	SLength	SWidth	PLength	PWidth	PredictedValues1
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

- 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:
LogisticRegression(class_weight=None, dual=True, random_state=None, tol=0.0001, max_iter=100, solver=liblinear, warm_start=False, n_jobs
=1, C=1.0, multi_class=ovr, verbose=0, penalty=l2, intercept_scaling=1.0, fit_intercept=True)

Coefficients:

      (Intercept)  PWidth  PLength
              -1.878   -0.021   -0.204
----- 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. Column header and data type of feature column for both 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.

10. Performance

Users can evaluate model performance through a list of parameters. The performance component can be attached to classification or regression algorithms.

10.1. 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 connection to the node model summary can be viewed with respective names.

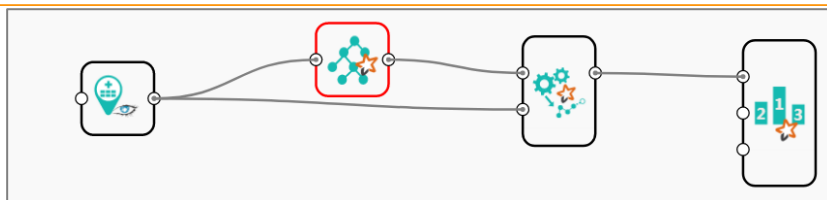
Spark Performance components can be of the following formats:

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'.

Component	Console	Summary	Result	Visualization	Properties
General	Spark-Performance				
Properties	Performance Type: MultiClassification Metrics ▾ Beta Value: 1				
	<input type="button" value="Apply"/>				

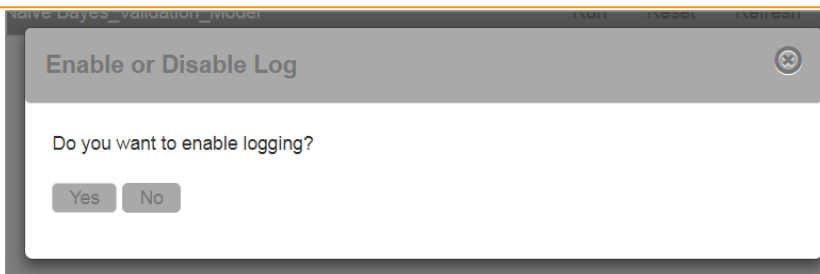
Users will get different outcomes based on the selected Performance types as described below:

- **Multi Classification Metrics**

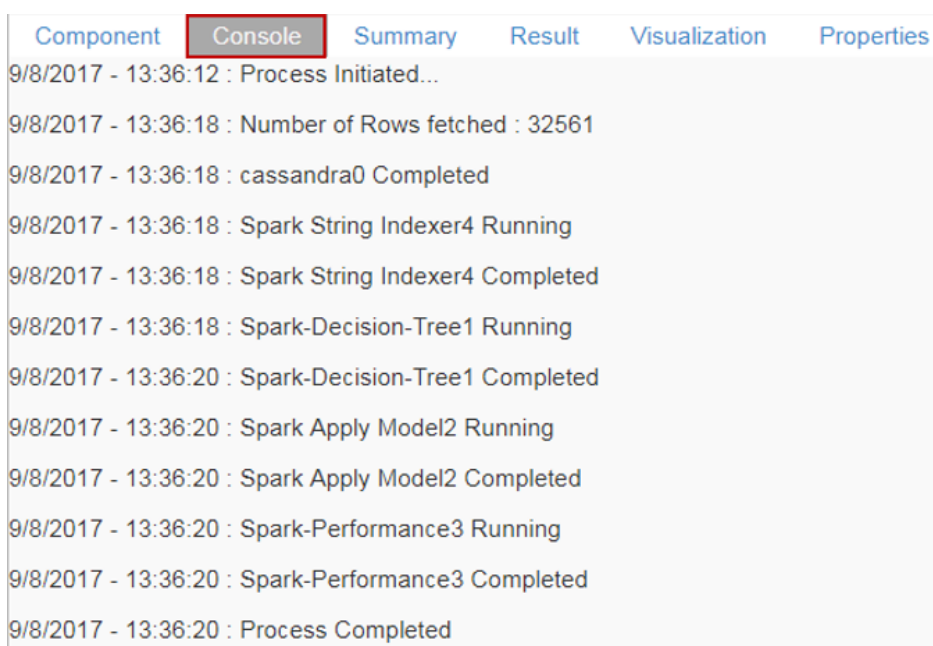
1. Navigate to the 'Properties' tab of the Spark Performance component.
2. Select 'Multi Classification Metrics' Performance type via the drop-down menu

Component	Console	Summary	Result	Visualization	Properties
General	Spark-Performance				
Properties	Performance Type: MultiClassification Metrics ▾ Beta Value: 0.6				
	<input type="button" value="Apply"/>				

3. Click 'Apply'.
4. Click 'Run'.
5. A message will pop-up to confirm whether users want to enable logging.
6. Click 'No'.



7. Users will be redirected to the 'Console' tab.



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

Summary of MultiClass Metrics

Model Name	Accuracy	Weighted Precision	Weighted Recall	Weighted FMeasure	Weighted FMeasure(beta 0.6)	Weighted True Positive Rate	Weighted False Positive Rate
Model 0	0.8233776603912656	0.8117325916023782	0.8233776603912657	0.8075883986455182	0.8063464941537403	0.8233776603912657	0.43674439111422253

----- Label Wise Model - 0 -----

Labels	Precision	Recall	FMeasure	FMeasure(beta 0.6)	TruePositiveRate	FalsePositiveRate
0.0	0.8422401039223469	0.9442152103559871	0.8903171666698454	0.8670268380485194	0.9442152103559871	0.557581941078944
1.0	0.715552805280528	0.442418058921056	0.5467727953345417	0.6150420558067257	0.442418058921056	0.05578478964401295

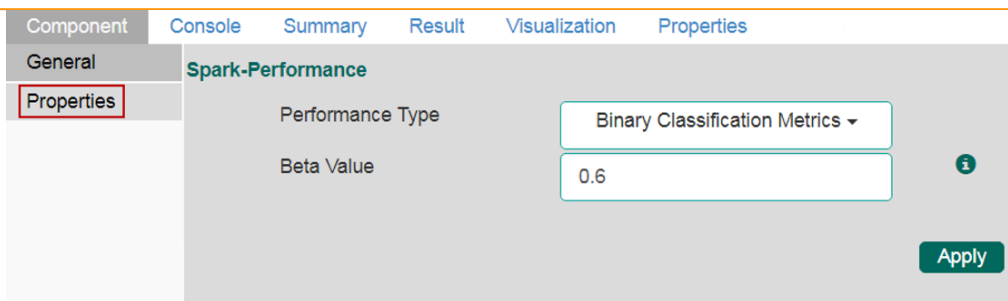
---- Confusion Matrix (Model - 0)----

	Predict 0.0	Predict 1.0
Actual 0.0	23341.0	1379.0
Actual 1.0	4372.0	3469.0

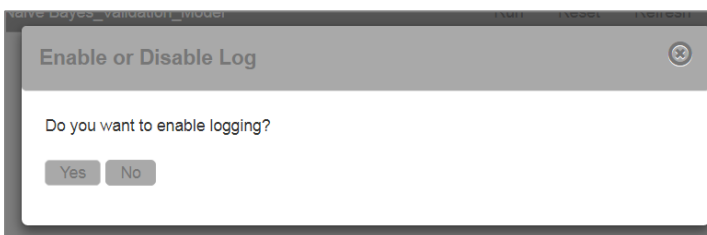
----- End of Summary -----

- **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



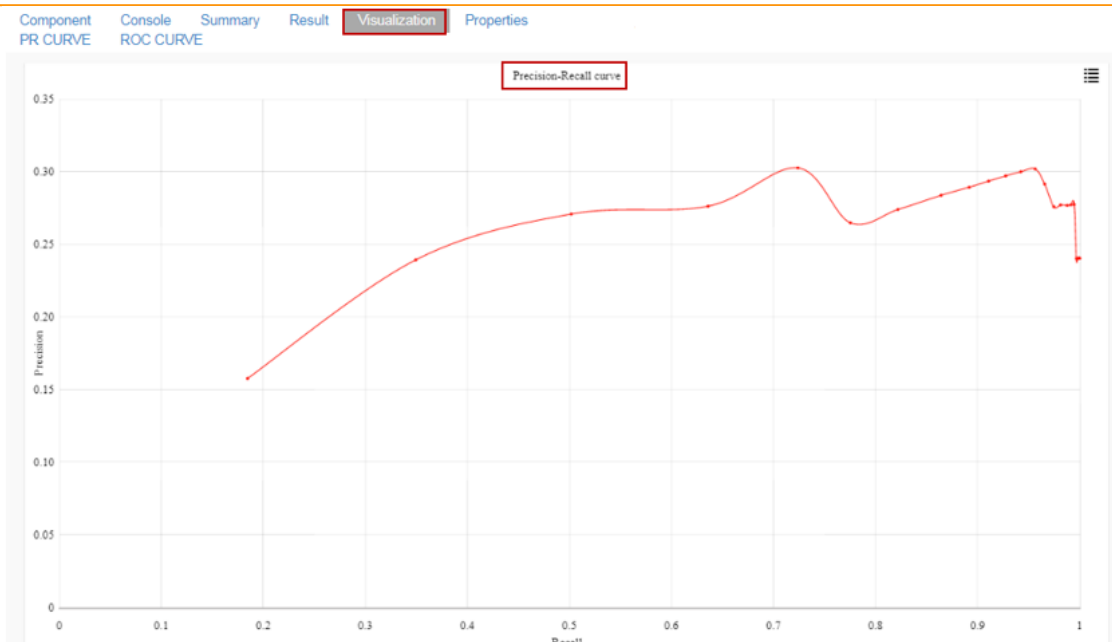
3. Click 'Apply'.
4. Click 'Run'.
5. A message will pop-up to confirm whether users want to enable logging.
6. Click 'No'.



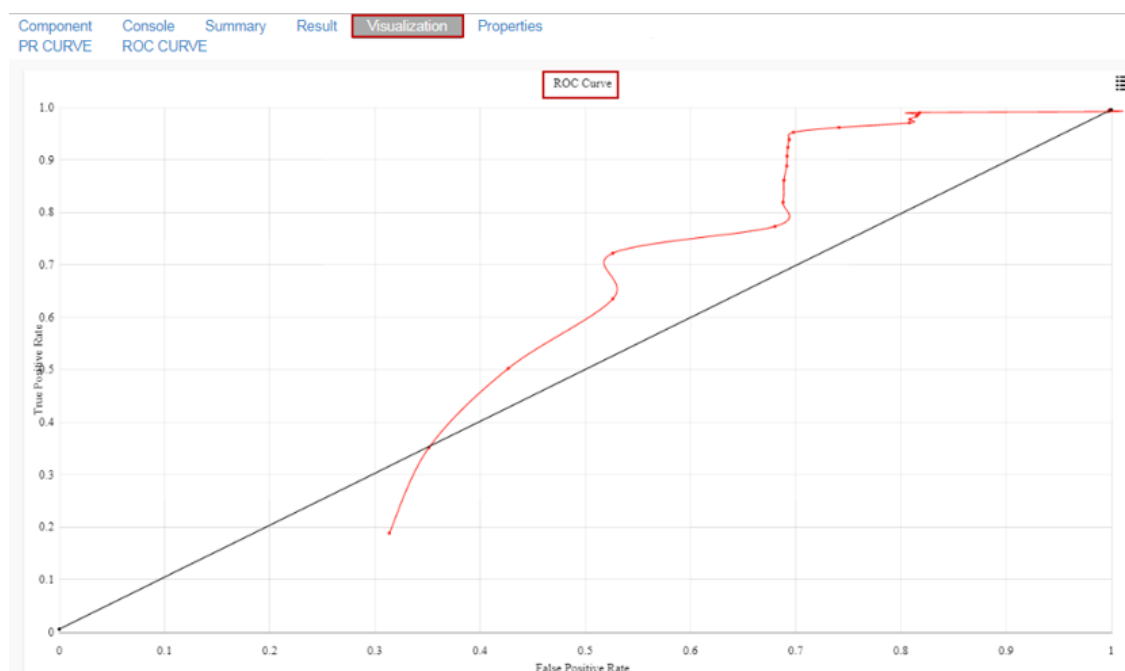
7. Users will be redirected to the 'Console' tab.
8. Users can follow the below given steps to display the result view if the selected performance type is Binary:
 - a. Click the dragged performance component on the workspace.
 - b. Click the 'Result' tab.

Component	Console	Summary	Result	Visualization	Properties
Model_0					
Show 10 entries					
falsepositiverate	fMeasure	precision	recall	threshold	fMeasure -beta 0.6
0.9978559870550162	0.38817958697969607	0.2409219596258001	0.998469582961357	13	0.30146698241954134
0.5262540453074434	0.3859730608453321	0.27707696582383995	0.6358882795561791	1051	0.3257295693676937
0.9991504854368932	0.38835095713330525	0.24096496619545174	1	1	0.30155338553891153
0.8149676375404531	0.43354882041810094	0.27771403986806253	0.987884198444076	51	0.34298037816346283
0.818042071197411	0.43494506720205234	0.27832696905892007	0.9946435403647494	25	0.34388284696807386
0.7413025889967637	0.4489166839562524	0.2924164028110279	0.9658206861369724	73	0.3586005384258634
0.6919093851132686	0.44516470844890454	0.2945640518023592	0.9108532074990435	149	0.3588313613183611
0.6915857605177993	0.43802067021609775	0.2902984764830421	0.8918505292692258	216	0.3533949470417394
0.4268203883495146	0.35248801899046	0.2716415849786	0.5018492539216937	1195	0.30918448230838025
0.351415857605178	0.28440938343367245	0.23978297015839678	0.3494452238234919	1292	0.2615061587002167
Showing 1 to 10 of 25 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



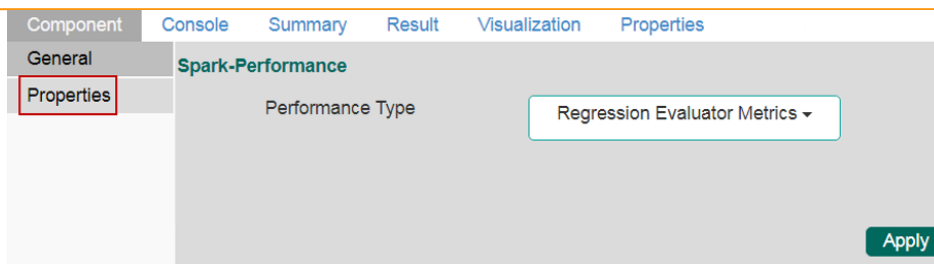
b. Result data displayed via the ROC Curve



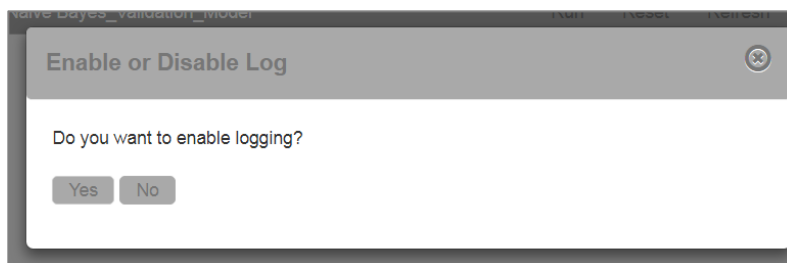
- **Regression Evaluator Metrics**

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. Click 'Run'.
5. A message will pop-up to confirm whether users want to enable logging.
6. Click 'No'.



7. Users will be redirected to the 'Console' tab.
8. View summary by following the steps given below:
 - a. Click the performance component onto the workspace
 - b. Click the 'Summary' tab.

Model Name	Mean Squared Error (MSE)	Root MSE (RMSE)	Mean Absolute Error (MAE)	Coefficient of Determination (R2)
0	0.17662233960873436	0.42026460665720394	0.17662233960873439	0.03390200316236669

10.2. 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 connection to the node model summary can be viewed with respective names.

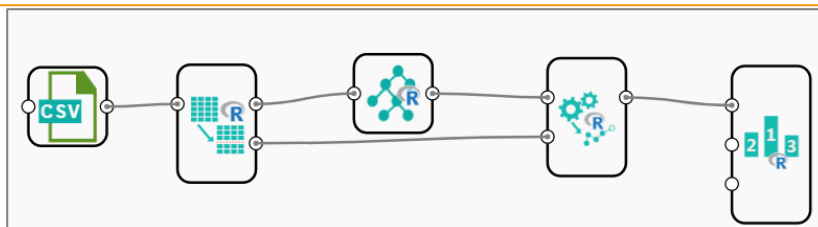
R Performance components can be of the following formats:

1. Binary Classification: Used when the label has two classes
2. Multi Classification: Used when the label has 3 or more beta values

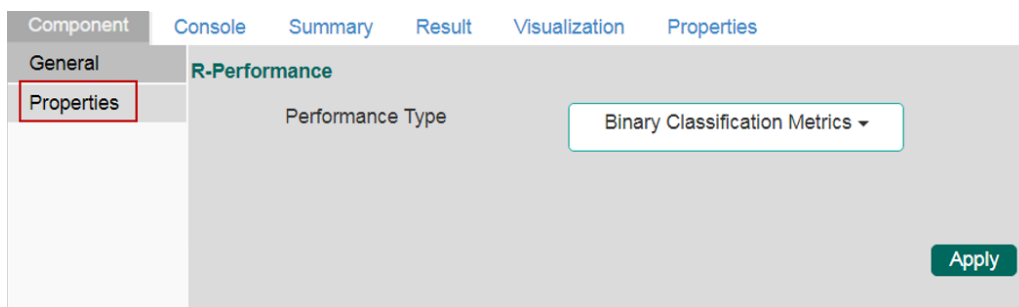
In the case of multiple models, all the model statistics will come in the summary of performance (up to 3 models can be compared).

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 CNR Tree has been used.)



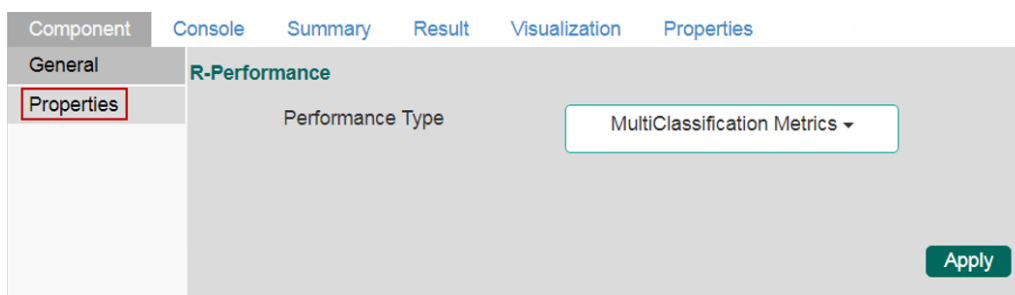
- ii) Configure the ‘Properties’ tab.
 - a. **Performance Type:** Select an option using the drop-down menu.
 - i. Binary Classification: To be used when the label has two classes.
 - ii. Multiclass Classification (Default option): To be used when the label has 3 or more beta values.
- iii) Click ‘Apply’



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. Click ‘Run’
5. Users will be redirected to the ‘Console’ tab.

Component	Console	Summary	Result	Visualization	Properties
	9/8/2017 - 14:13:30 : Process Initiated...				
	9/8/2017 - 14:13:31 : csv1 is started.				
	9/8/2017 - 14:13:31 : csv1 is completed.				
	9/8/2017 - 14:13:31 : R Split Data0 is started.				
	9/8/2017 - 14:13:34 : R Split Data0 is completed.				
	9/8/2017 - 14:13:34 : R-CNR Tree2 is started.				
	9/8/2017 - 14:13:36 : R-CNR Tree2 is completed.				
	9/8/2017 - 14:13:36 : R Apply Model3 is started.				
	9/8/2017 - 14:13:36 : R Apply Model3 is completed.				
	9/8/2017 - 14:13:36 : R-Performance4 is started.				
	9/8/2017 - 14:13:37 : R-Performance4 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 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

Component Console **Summary** Result Visualization Properties

-----Summary of Model Comparison -----
 ----- Performance of first model -----

Confusion Matrix and Statistics

	setosa	versicolor	virginica
setosa	11	0	0
versicolor	0	4	0
virginica	0	1	14

Overall Statistics

Accuracy : 0.9667
 95% CI : (0.8278, 0.9992)
 No Information Rate : 0.4667
 P-Value [Acc > NIR] : 4.148e-09

Kappa : 0.9454
 McNemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.8000	1.0000
Specificity	1.0000	1.0000	0.9375
Pos Pred Value	1.0000	1.0000	0.9333
Neg Pred Value	1.0000	0.9615	1.0000
Prevalence	0.3667	0.1667	0.4667
Detection Rate	0.3667	0.1333	0.4667
Detection Prevalence	0.3667	0.1333	0.5000
Balanced Accuracy	1.0000	0.9000	0.9688

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

- **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

General **R-Performance**

Properties

Performance Type

Apply

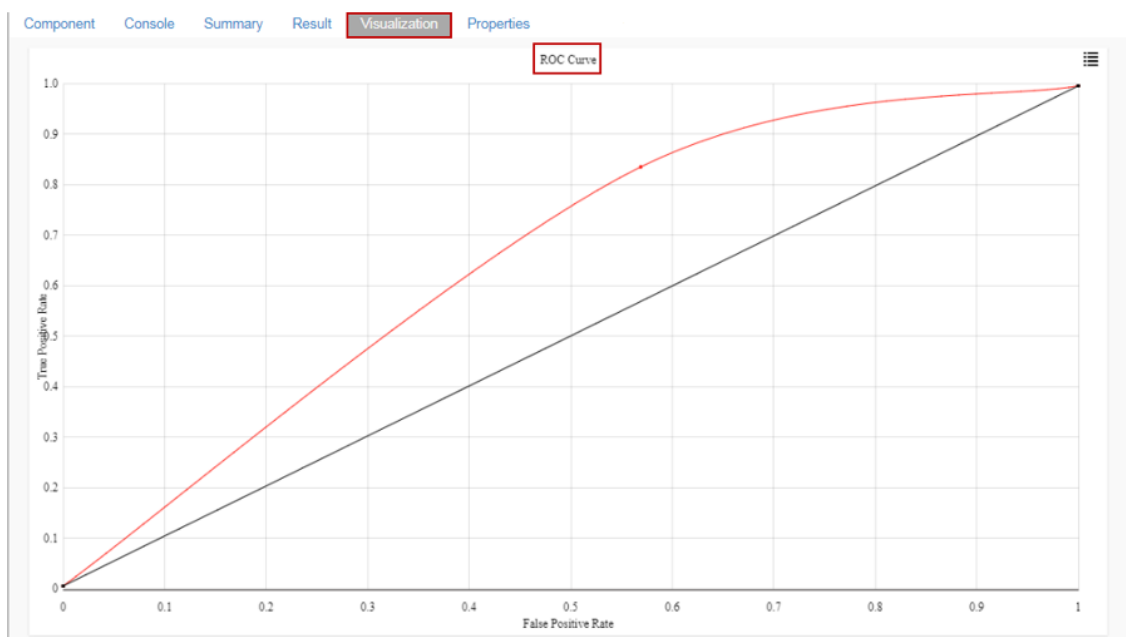
3. Click 'Apply'
4. Click 'Run'
5. Users will be redirected to the 'Console' tab.

Component **Console** Summary Result Visualization Properties

```

9/8/2017 - 15:10:34 : Process Initiated...
9/8/2017 - 15:10:35 : csv4 is started.
9/8/2017 - 15:10:35 : csv4 is completed.
9/8/2017 - 15:10:35 : R Split Data0 is started.
9/8/2017 - 15:10:37 : R Split Data0 is completed.
9/8/2017 - 15:10:37 : R-CNR Tree2 is started.
9/8/2017 - 15:10:39 : R-CNR Tree2 is completed.
9/8/2017 - 15:10:39 : R Apply Model1 is started.
9/8/2017 - 15:10:39 : R Apply Model1 is completed.
9/8/2017 - 15:10:39 : R-Performance3 is started.
9/8/2017 - 15:10:39 : R-Performance3 is completed.
  
```

6. Click the 'Visualization' tab to see the graphical representation of the result data.



Note:

- a. In case of the multiple models, all the model statistics will be displayed in the summary tab of performance component (up to 3 models can be compared).
- b. No data will be displayed under the 'Result' tab for R-Performance (Binary Classification).

11. Data Writer(s)

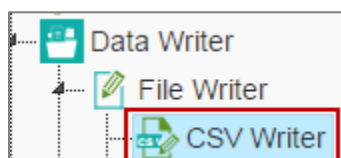
Data Writers are provided to store the results of the predictive analysis in flat files or databases for further in-depth analysis.

11.1. File Writer

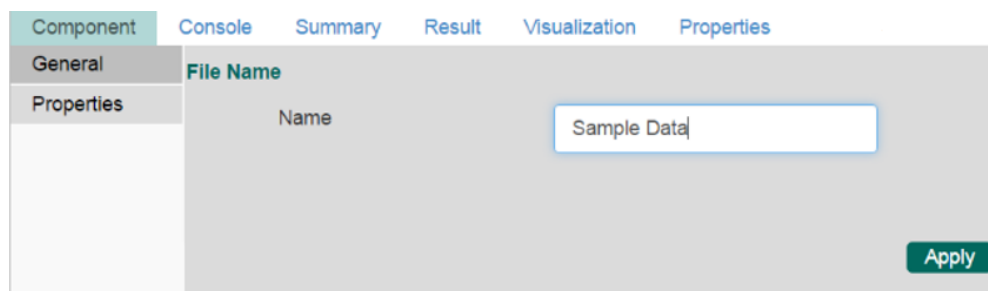
Users can write output data to flat files like CSV, TEXT, and DAT files using the File Writer.

11.1.1. CSV Writer

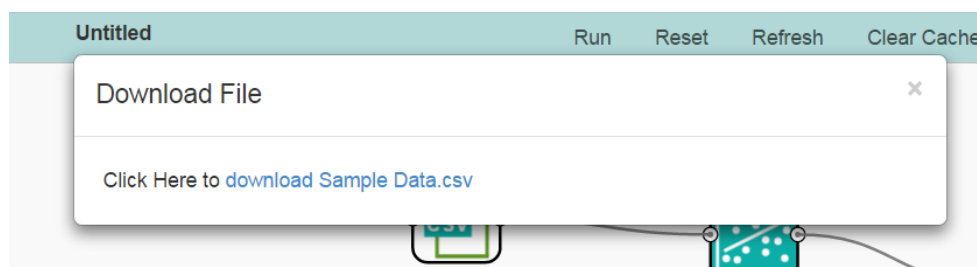
- i) Click 'TreeNode' provided next to the 'Data Writer' option.
- ii) Select 'File Writer' option.
- iii) Select and drag 'CSV Writer' component to the workspace.



- iv) Connect the 'CSV Writer' to a configured data source.
- v) Click on CSV Writer component to access component properties.
- vi) Enter 'File Name' in the displayed field.
- vii) Click 'Apply'



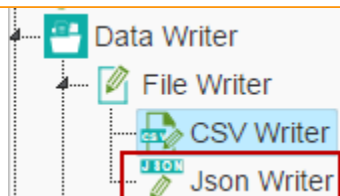
- viii) Click 'Run'
- ix) A pop-up message will appear with a link to download the CSV file.



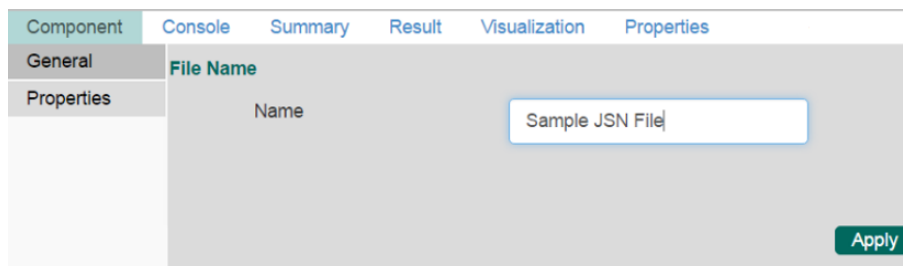
11.1.2. Click the link to download the CSV file.

11.1.3. JSON Writer

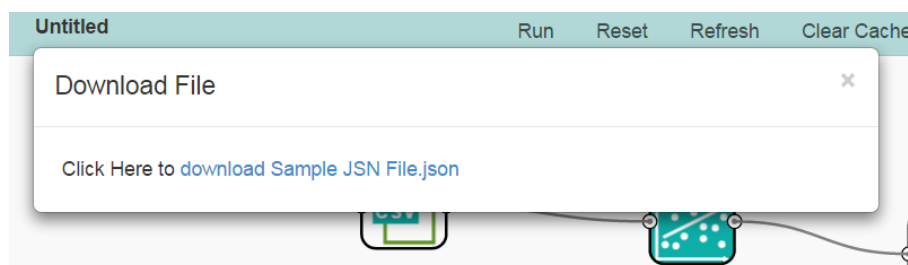
- i) Click on 'TreeNode' provided next to the 'Data Writer' option.
- ii) Select 'File Writer' option.
- iii) Select and drag 'JsonWriter' component to the workspace.



- 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) Click on 'Run' option.
- ix) A Pop-up message will appear with a link to download the JSON file.



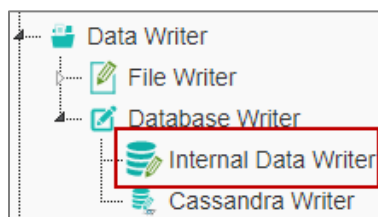
- x) Click the link to download the JSON file.

11.2. Database Writer

11.2.1. Internal Data Writer

This data writer will store the data into 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.

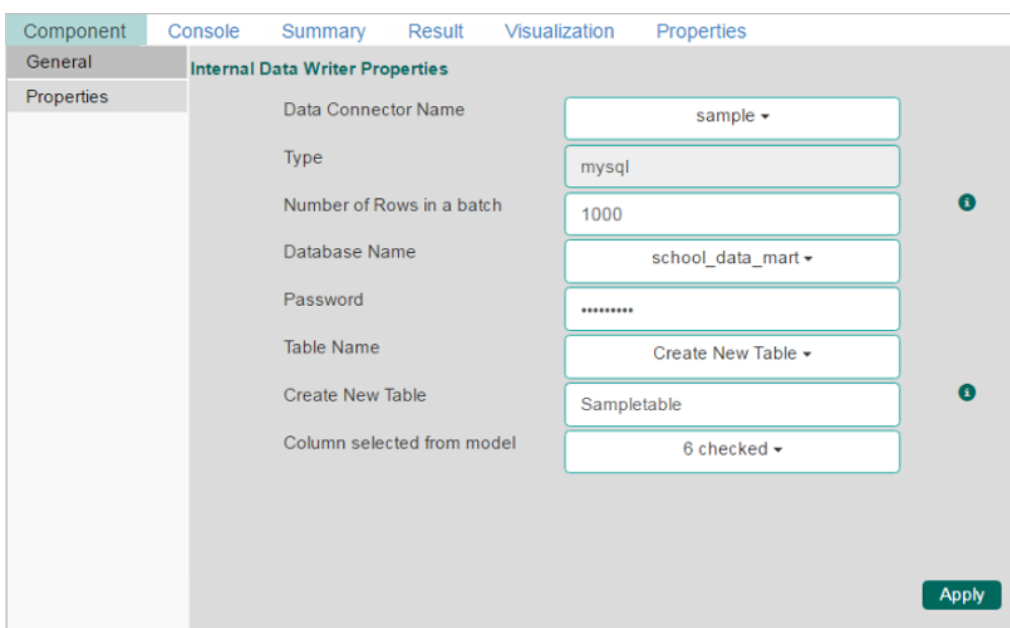


- iv) Drag and Connect the 'Internal Data Writer' component to a configured data source onto the workspace.
- v) Click 'Internal Data Writer' component to access the Component properties

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:

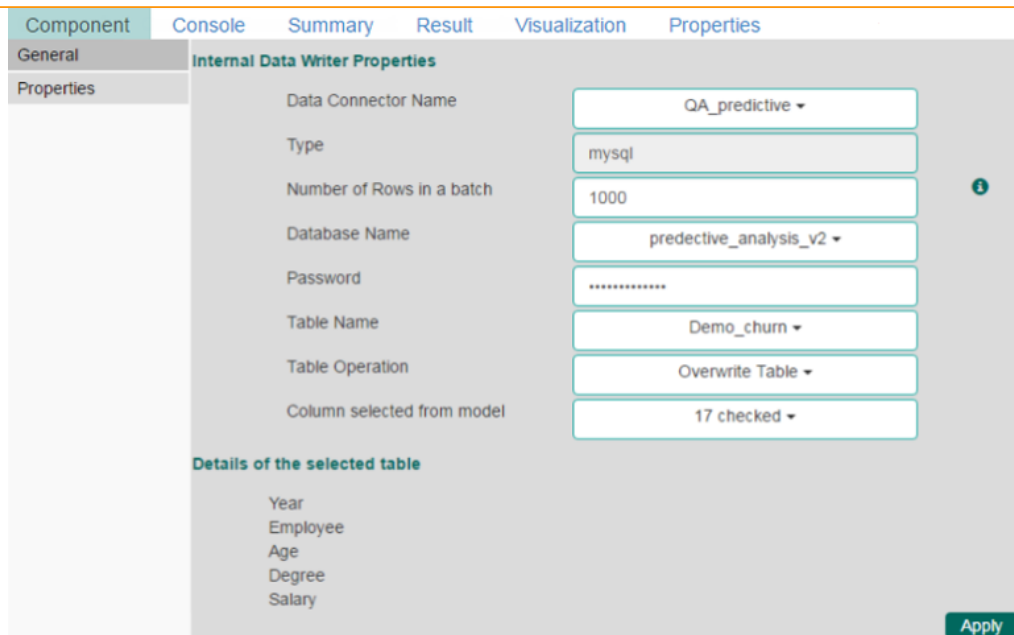
- i. **Data Connector Name:** All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.
- ii. **Type:** This field will be preselected based on the selected data Connector.
- iii. **Number of Rows in a batch:** Enter a number to limit the 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. **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.
- viii. **Column Selected from model:** Select columns that are needed to be written into the selected database.



Component	Console	Summary	Result	Visualization	Properties
General	Internal Data Writer Properties				
Properties	<p>Data Connector Name: sample</p> <p>Type: mysql</p> <p>Number of Rows in a batch: 1000</p> <p>Database Name: school_data_mart</p> <p>Password:</p> <p>Table Name: Create New Table</p> <p>Create New Table: Sampletable</p> <p>Column selected from model: 6 checked</p> <p style="text-align: right;">Apply</p>				

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 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.

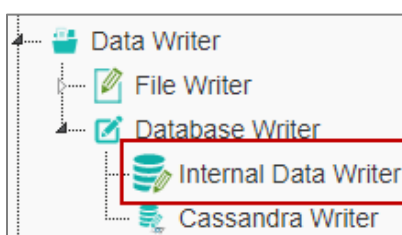


- vi) Click 'Apply'
- vii) Click 'Run'
- viii) Users will be directed to the 'Console' tab.
- ix) The data will be saved in the selected database.

11.2.1.1. 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 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.



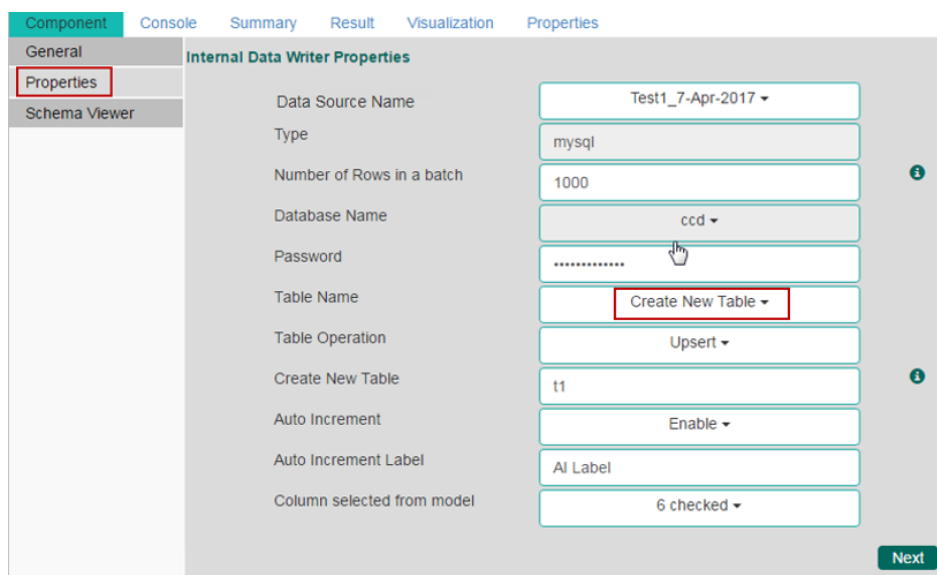
- iv) Connect the 'Internal Data Writer' component to a configured data source.
- v) Click the 'Internal Data Writer' component.
- vi) Users will be directed to the components tab.

Users will have different properties fields based on the selected table choice as described below:

a. Selecting 'Create a New Table' as Table Operation:

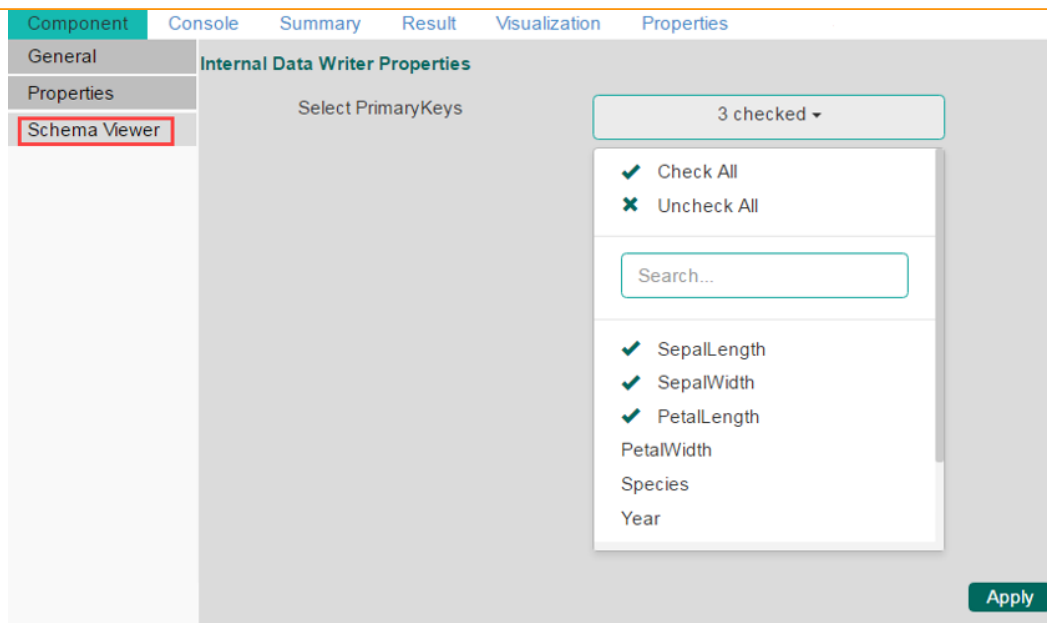
- i. **Data Connector Name:** All the available data connectors in particular user id will be listed. Select a data connector from the drop-down menu.
- ii. **Type:** This field will be preselected based on the selected data Connector.
- iii. **Number of Rows in a batch:** Enter a number to limit the 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 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’



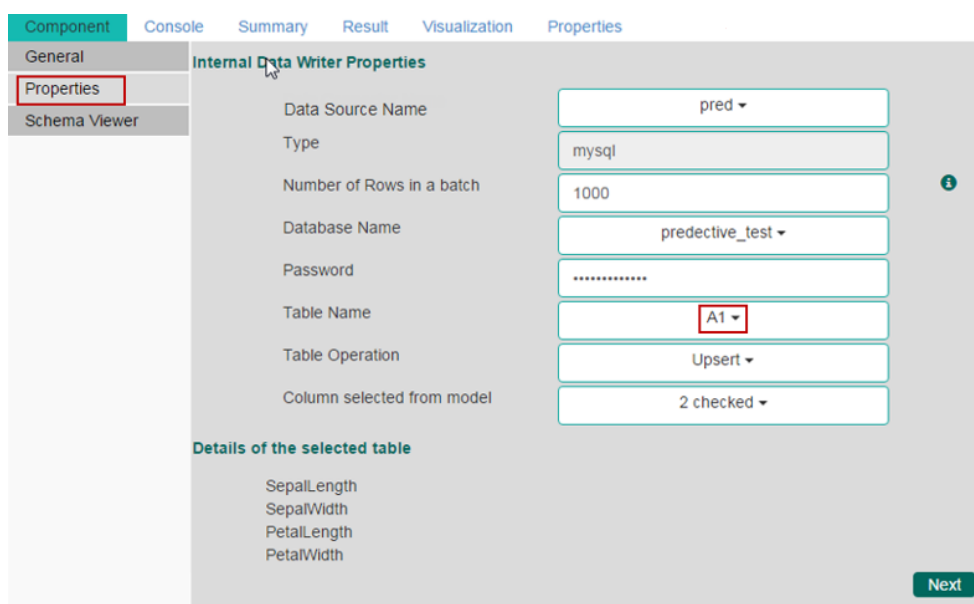
Note: The Schema Viewer tab will be displayed only after configuring the ‘Table Name’ field.

- vii) Users will be directed to the ‘Schema Viewer’ tab.
- viii) Define Primary keys by using the ‘Select Primary Keys’ field.
- ix) Click ‘Apply’

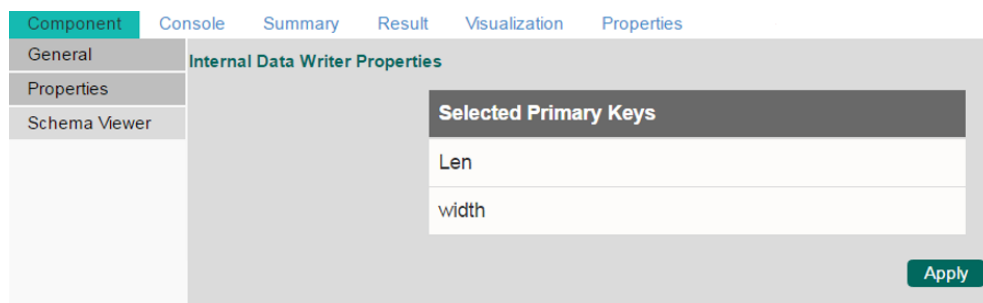


b. Selecting an Existing Table as Table Operation:

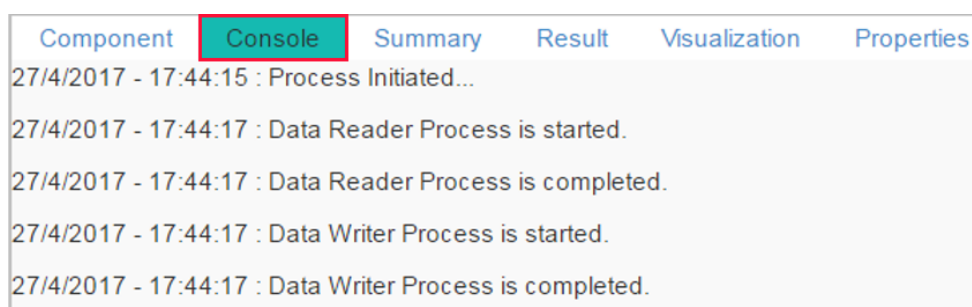
- i. **Data Connector Name:** Select a data connector from the drop-down menu
 - ii. **Type:** Displays a type based on the selected data connector
 - iii. **Number of Rows in a batch:** Enter a number to limit the entries of rows for one batch
 - iv. **Database Name:** Select a database name from the 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'



- xi) Users will be directed to the 'Schema Viewer' tab.
- xii) The defined/selected primary keys will be displayed.
- xiii) Click 'Apply'



- xiv) Click 'Run'
- xv) Users will be directed to the console tab.



- xvi) Users will be directed to the result tab.

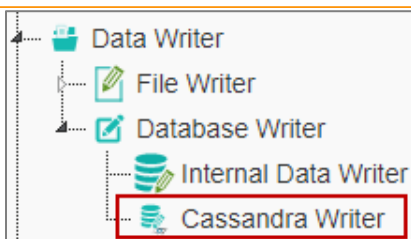
SepalLength	SepalWidth	PetalLength	PetalWidth	Species	ClusterNumber
5.1	3.5	1.4	0.2	setosa	5
4.9	3	1.4	0.2	setosa	5
4.7	3.2	1.3	0.2	setosa	3
4.6	3.1	1.5	0.2	setosa	3
5	3.6	1.4	0.2	setosa	5
5.4	3.9	1.7	0.4	setosa	1
4.6	3.4	1.4	0.3	setosa	3
5	3.4	1.5	0.2	setosa	5
4.4	2.9	1.4	0.2	setosa	3
4.9	3.1	1.5	0.1	setosa	5

Note: The Result tab appears when the data source is connected with an algorithm component. The data will be saved in the selected data source.

11.2.2. Cassandra Writer

Cassandra Writer can be used to store 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.
- v) Click the 'Cassandra Writer' component to access it.

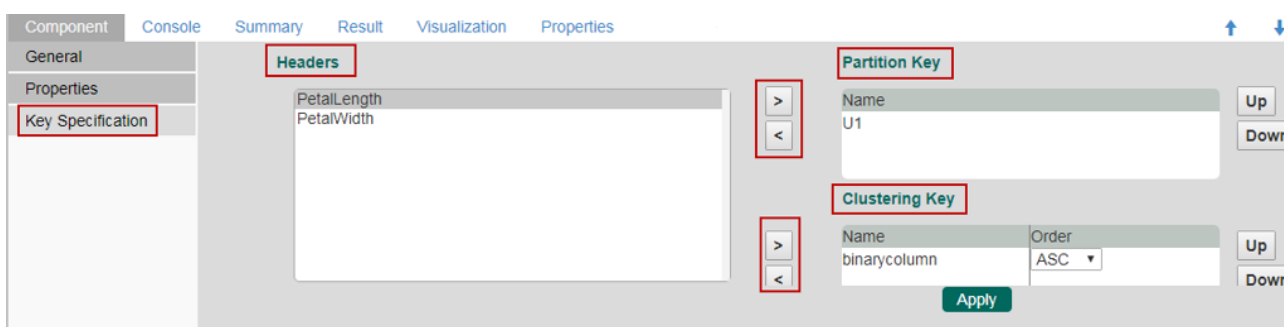
Properties:

a. Selecting 'Create a New Table' as Table Operation

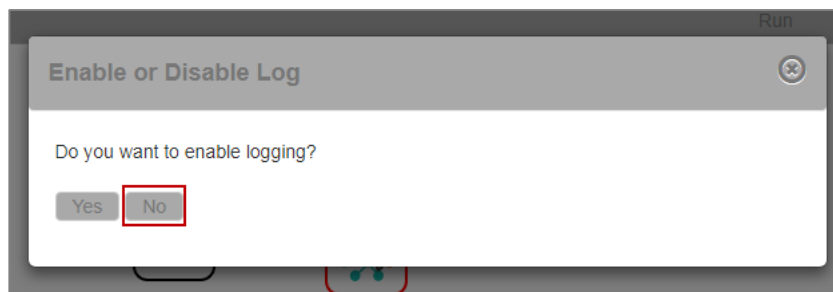
- i. **Select Data Connector:** Select a data connector using the drop-down menu
 - ii. **Host Name:** Based on the chosen data connector a hostname will be displayed (Users cannot edit this field).
 - iii. **Port Name:** The server port number will be displayed (Users cannot edit this field).
 - iv. **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.
- vi) Click 'Next'.

- vii) Users will be redirected to the 'Key Specification' tab.

- viii) Configure the following information:
 - i. **Headers:** All the columns from the data set will be listed.
 - ii. **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.
 - iii. **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 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.



- ix) Click ‘Apply’
- x) Click ‘Run’
- xi) A message will pop-up to confirm whether users want to enable logging.
- xii) Click ‘No’



- xiii) Users will be redirected to the ‘Console’ tab.

Component	Console	Summary	Result	Visualization	Properties
	28/7/2017 - 13:59:32 : Process Initiated...				
	28/7/2017 - 13:59:32 : Process started				
	28/7/2017 - 13:59:32 : cassandra1 Running				
	28/7/2017 - 13:59:38 : Number of Rows fetched : 150				
	28/7/2017 - 13:59:38 : cassandra1 Completed				
	28/7/2017 - 13:59:38 : Spark-Decission-Tree0 Running				
	28/7/2017 - 13:59:39 : Spark-Decission-Tree0 Completed				
	28/7/2017 - 13:59:39 : cassandra writer2 Running				
	28/7/2017 - 13:59:55 : cassandra writer2 Completed				
	28/7/2017 - 13:59:55 : Process 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
- Quarum

Or

- i) Connect the 'Cassandra Writer' to a configured data source.
- ii) Click the 'Cassandra Writer' component to access it.
- iii) **Properties Tab**

b. Selecting an Existing Table as Table Operation

- 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) Click 'Apply'

Component Console Summary Result Visualization Properties

General Data Service Properties

Properties

Key Specification

Select Data Connector: cassandraqa ▾

Host name: 192.168.1.17

Port Number: 9042

Username: smb

Password:

No: of rows in a batch: 100

Select Key Space: UCI ▾

Replication Factor: 3

Select Table: pok1 ▾

Select columns: Select ▾

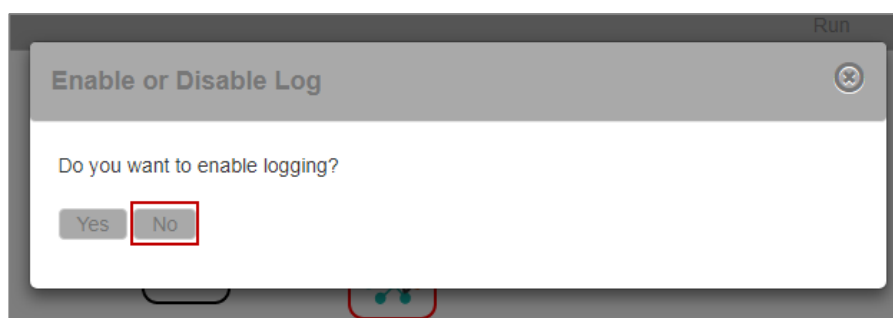
Consistency: 1 checked ▾

Settings: Overwrite ▾

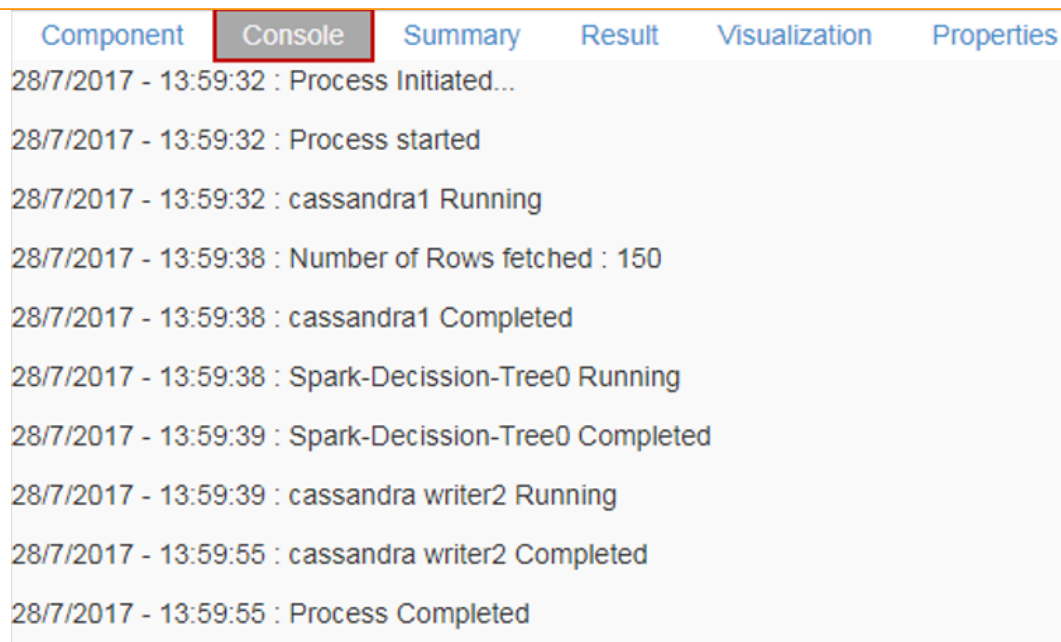
Headers	Type
u1	TIMEUUID
acceleration	INT
carname	TEXT
cylinders	INT
displacement	INT
horsepower	INT
model_year	INT
mpg	INT
origin	INT
weight	INT

Apply

- v) Click 'Run'
- vi) A message will pop-up to confirm whether users want to enable logging.
- vii) Click 'No'



- viii) Users will be redirected to the 'Console' tab.



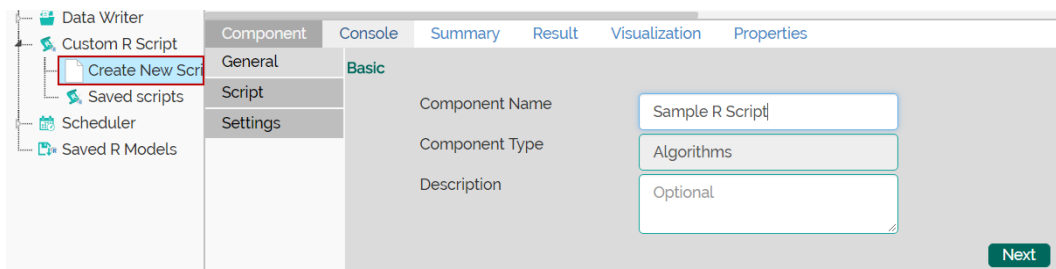
- ix) The data will be saved in the selected Cassandra Writer.

12. 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.

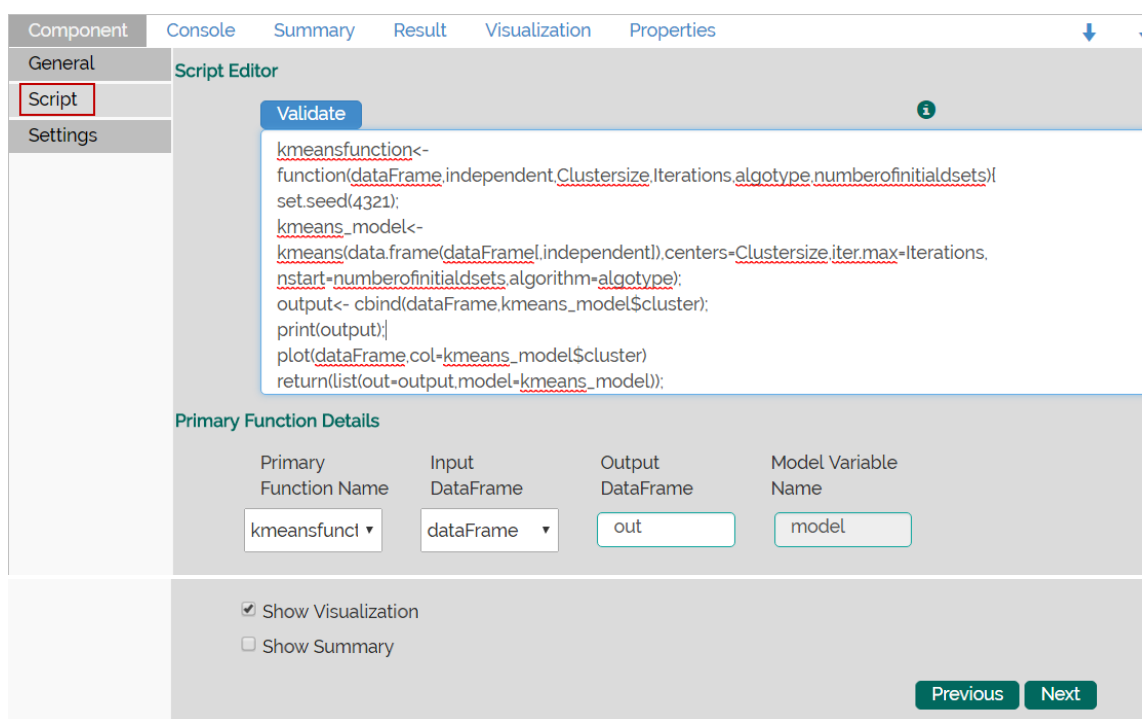
12.1. Creating a New R Script



- 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 the R-script in the given space under ‘Script Editor’.

- 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 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’

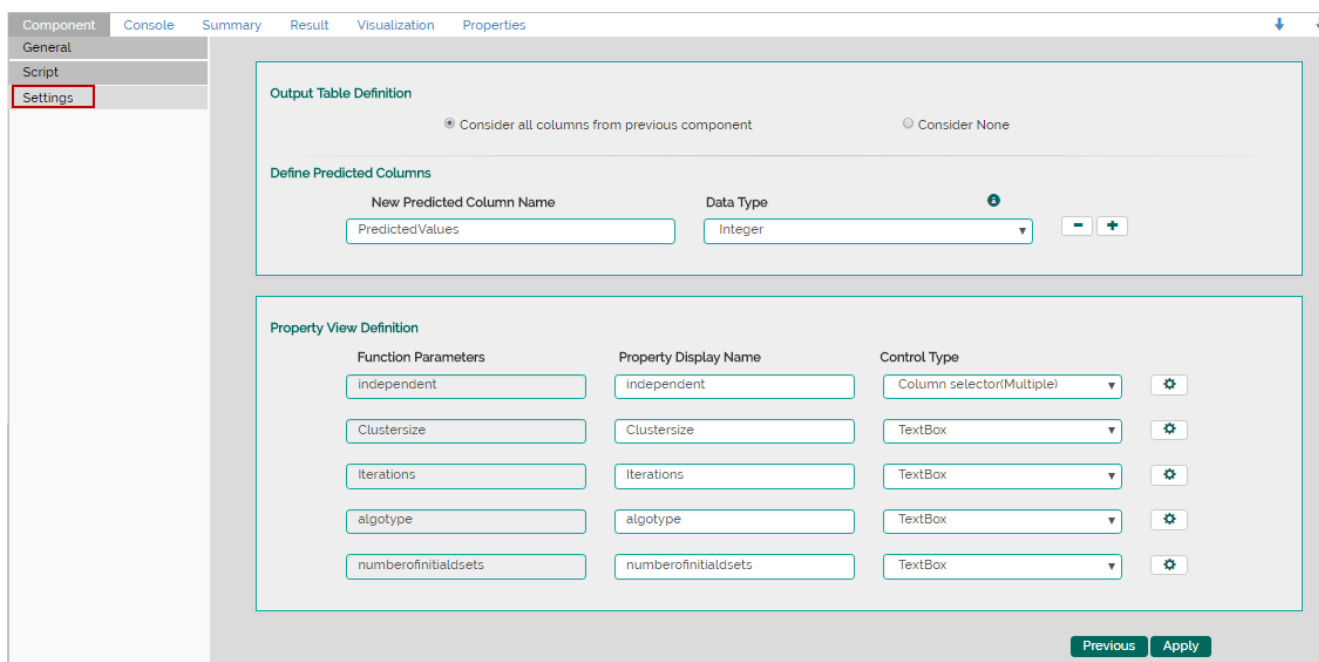


- ix) Users will be directed to the ‘Settings’ tab.
- x) Configure the following fields:
 - a. **Output Table Definition**
 - This option will configure a number of output columns, column headers, data types.
 - i. **Consider all columns from the previous component:** To display all columns from the previous 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. : To remove the added row containing ‘Data Type’ and ‘New Predicted Column Name’.
 - vi. : To add a new row containing ‘Data Type’ and ‘New Predicted Column Name’.

b. Property View Definition

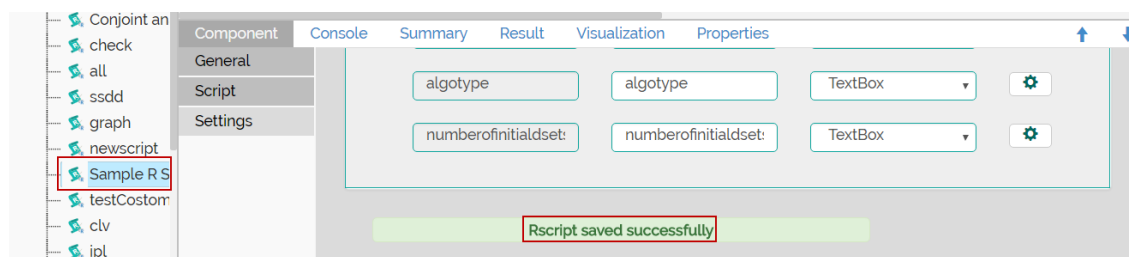
- i. **Function Parameters:** Actual names of parameters configured in the script.
- ii. **Property Display Name:** Parameter name to be displayed while configuring saved R script as a component.
- iii. **Control Type:** User can select out of the following options:
 1. Text box,
 2. Drop-down menu,
 3. Column Selector (single),
 4. Column Selector (multiple).
- iv. **Settings option** : To set display for mandatory fields and validate data type for input column. This field is associated with function parameters.

xi) Click 'Apply'



Function Parameters	Property Display Name	Control Type
independent	independent	Column selector(Multiple)
Clustersize	Clustersize	TextBox
Iterations	Iterations	TextBox
algotype	algotype	TextBox
numberofinitialdsets	numberofinitialdsets	TextBox

xii) The newly created R Script will be saved in the 'Saved Scripts' list for the R scripts.



Function Parameters	Property Display Name	Control Type
algotype	algotype	TextBox
numberofinitialdset:	numberofinitialdset:	TextBox


Rscript saved successfully

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) or 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:

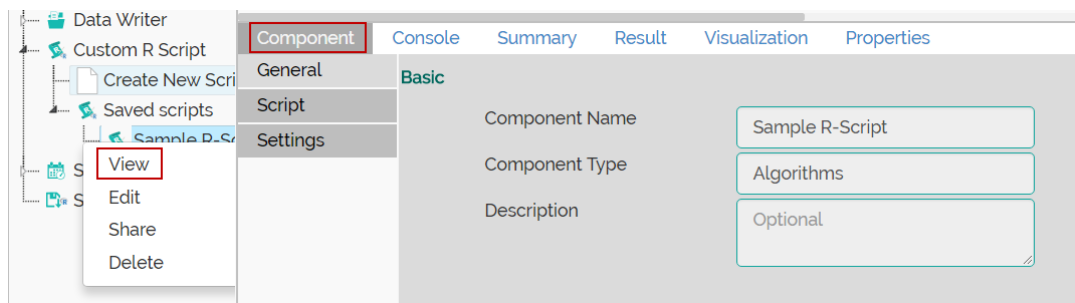
- a. Click the 'Information' button  to get the above-mentioned list of rules for R-script.
- b. 'Model Variable Name' can be enabled only after selecting 'Show Summary' option.
- c. Select 'Show Summary' and 'Show Visualization' option only if, the R-script carries both the items.
- d. All the supported 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.

12.2. Saved R-Scripts

This section describes options that can be applied to a saved R Script.

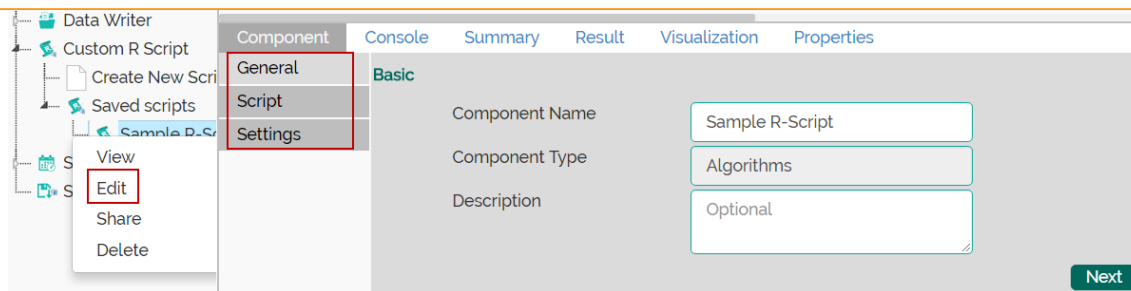
12.2.1. Viewing a Saved R Script

- 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.



12.2.2. Editing a Saved R Script

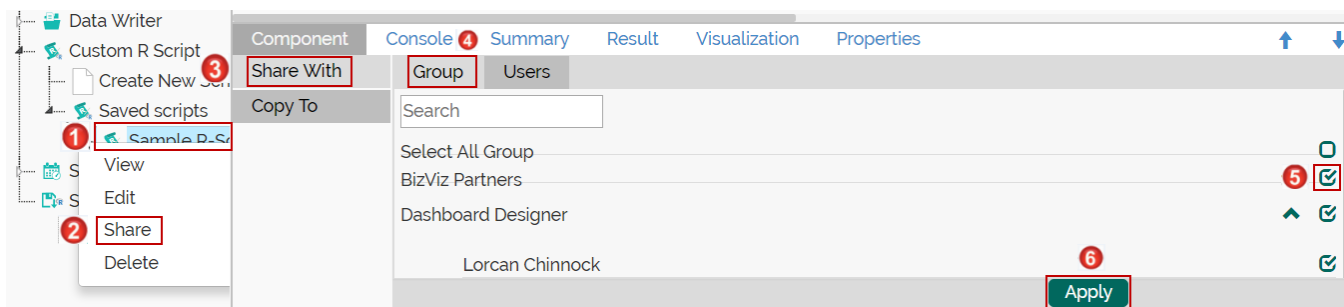
- i) Select an R Script from the list of 'Saved R-Script'.
- ii) Right-click on the selected R Script.
- iii) A context menu will open.
- iv) Select 'Edit'
- v) Users will be redirected to the 'Component' tab
- vi) Users can edit the required fields provided under **General**, **Script**, and **Settings** tabs.



12.2.3. 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:

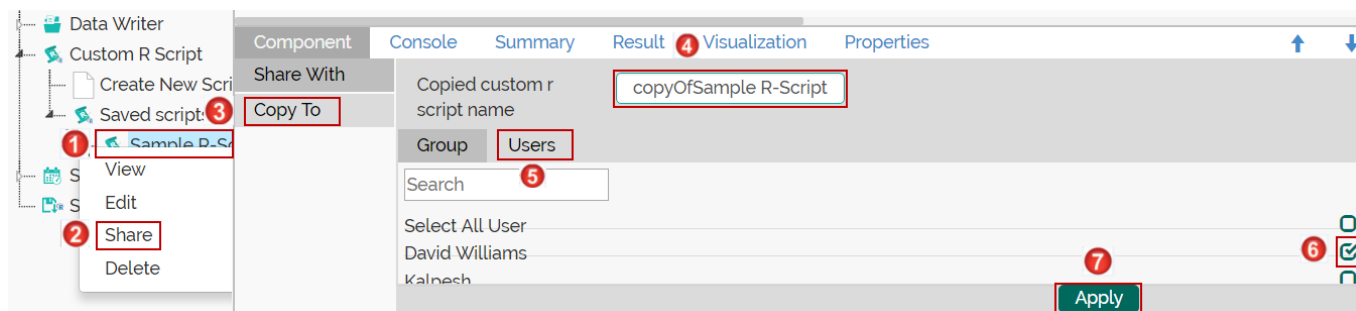
1. **Share With:** This option allows the user to share a custom R script with selected users or user groups. Any changes made to the custom R script will be transferred to all the users with whom the custom R script has been shared.
 - i) Right-click on a saved R script from the list of **'Saved Scripts'**.
 - 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.

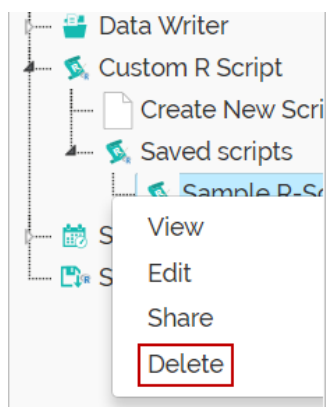
vii) Click 'Apply'



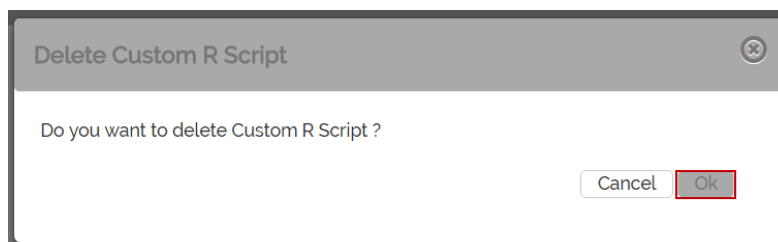
viii) The copied saved R script will be shared with the selected user(s)/group(s).

12.2.4. 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'.



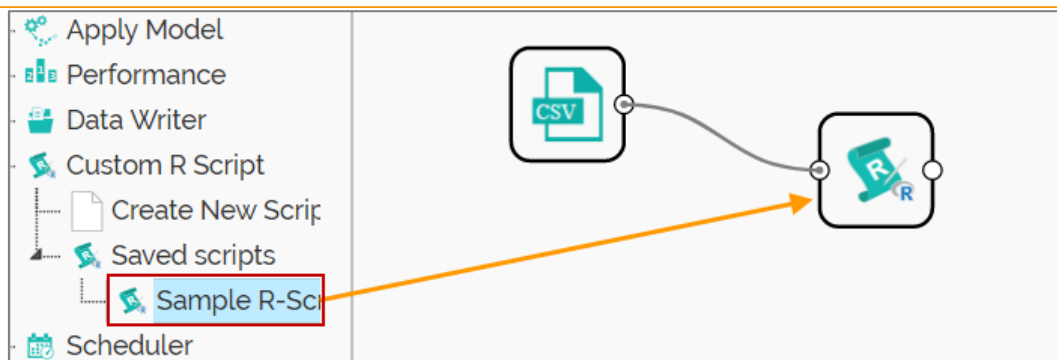
- v) A pop-up window will appear to assure the deletion.
- vi) Click 'Ok'



vii) The selected R-Script will be deleted.

12.2.5. 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'

Component	Console	Summary	Result	Visualization	Properties
General	Dynamic Fields				
Custom Group	independent			6 checked ▾	
	Clustersize			5	
	Iterations			100	
	algotype			Lloyd	
	numberofinitialdsets			1	
					Apply

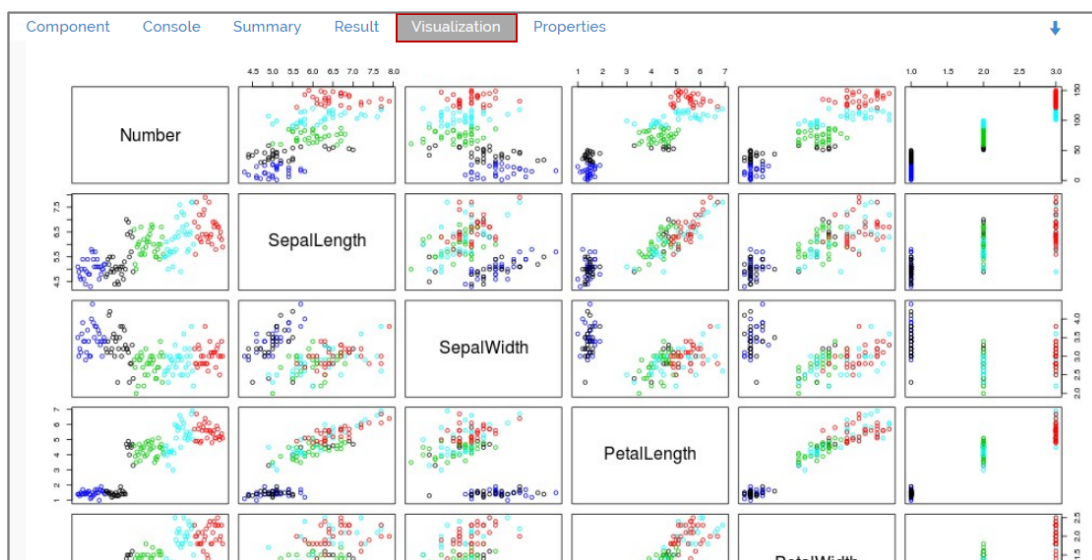
- vii) Click 'Run'
- viii) Users will be directed to the 'Console' tab.

Component	Console	Summary	Result	Visualization	Properties
	22/11/2017 - 11:56:34 : Process Initiated...				
	22/11/2017 - 11:56:39 : csv0 is started.				
	22/11/2017 - 11:56:39 : csv0 is completed.				
	22/11/2017 - 11:56:50 : Custom R Script1 is started.				
	22/11/2017 - 11:56:50 : 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.

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues
1	5.1	3.5	1.4	0.2	setosa	4.0
2	4.9	3	1.4	0.2	setosa	4.0
3	4.7	3.2	1.3	0.2	setosa	4.0
4	4.6	3.1	1.5	0.2	setosa	4.0
5	5	3.6	1.4	0.2	setosa	4.0
6	5.4	3.9	1.7	0.4	setosa	4.0
7	4.6	3.4	1.4	0.3	setosa	4.0
8	5	3.4	1.5	0.2	setosa	4.0
9	4.4	2.9	1.4	0.2	setosa	4.0
10	4.9	3.1	1.5	0.1	setosa	4.0

- x) Click the 'Visualization' tab.
- xi) The result data will be displayed through graphics.



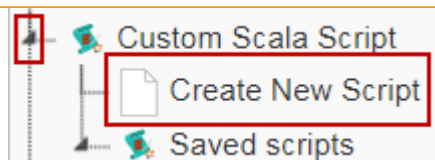
Note: The above-given process is displayed for a CSV data source. A similar set of steps can be followed for other data source types.

13. Custom Scala Script

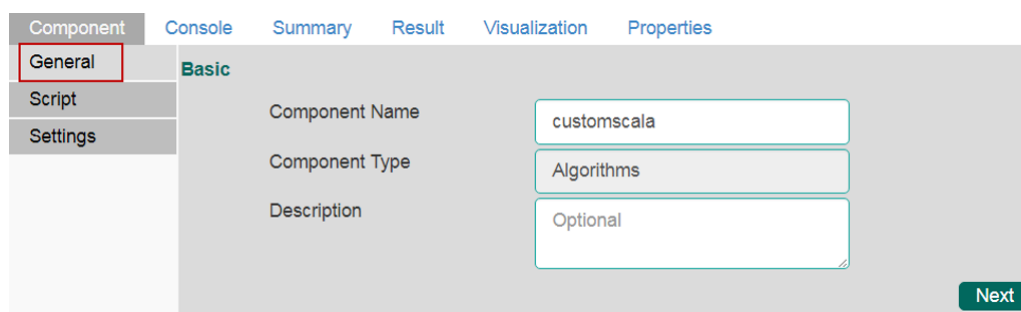
Users can create and add customized algorithm components using the 'Custom Scala Script' component. The created scripts will be stored in the 'Saved Scripts' module provided for the Scala Scripts. The 'Custom Scala Script' component will run only on Spark.

13.1. Creating a New Scala Script

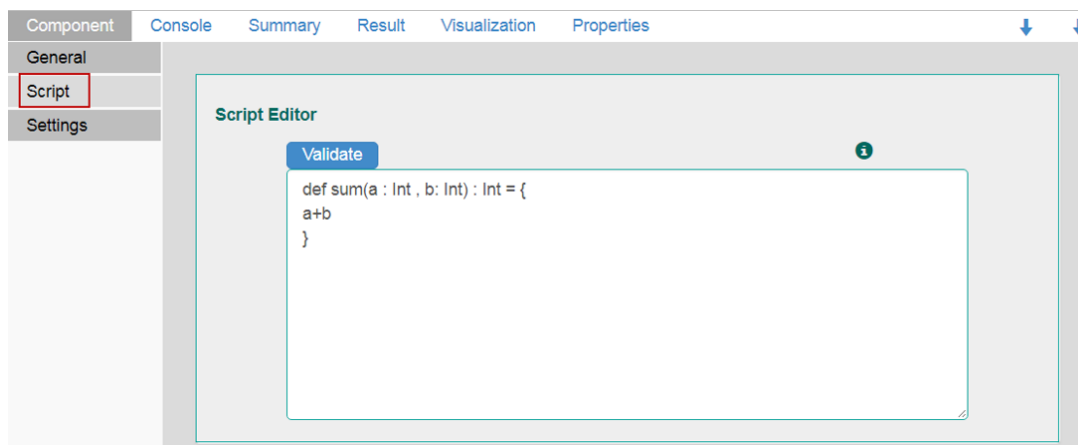
- i) Click 'Custom Scala Script' tree-node on the Predictive Analysis home page.
- ii) Click 'Create New Script'.



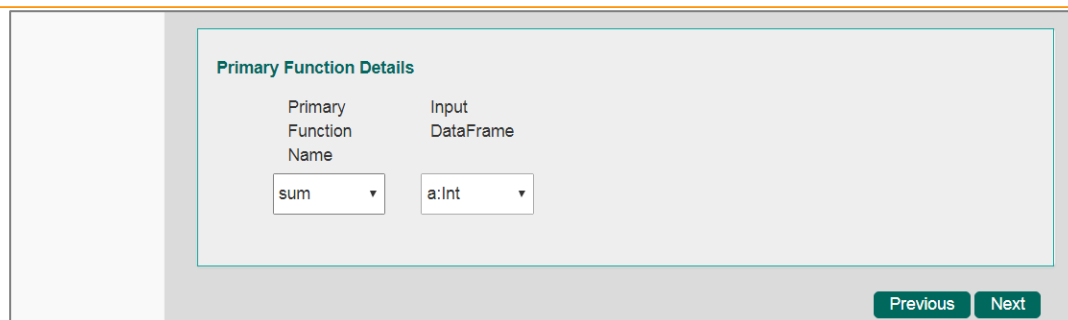
- iii) Users will be directed to the 'Component' tab.
- iv) Configure the following fields in the 'General' tab:
 - a. Basic
 - i. **Component Name:** Enter a name or title that you wish to give a saved Scala Script.
 - ii. **Component Type:** Default Component type will be displayed in this field.
 - 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 under 'Script Editor'.
 - ii. Click the 'Validate' option.





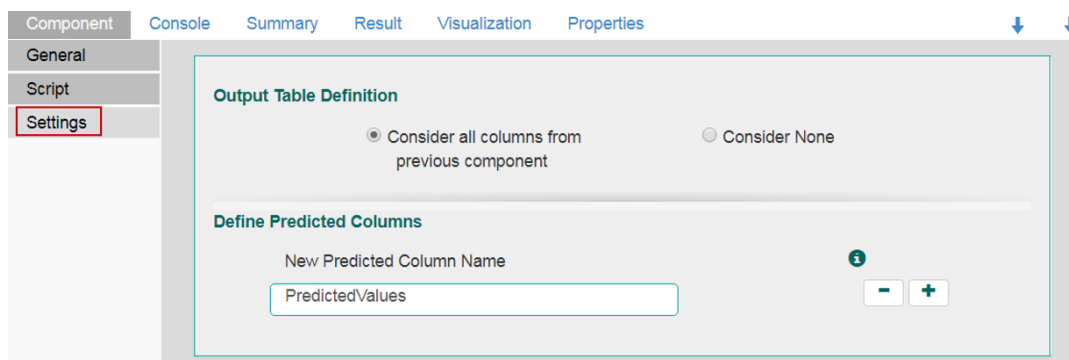
- iii. Configure the required 'Primary Function Details' to embed the customized Scala script into a function.
 - 1. **Primary Function Name:** Select name of the created function from the drop-down menu.
 - 2. **Input Data Frame:** Select a dataset (that has been used above) from a drop-down menu.
- viii) Click 'Next' (Users can click 'Previous' if wish to open the previous page)




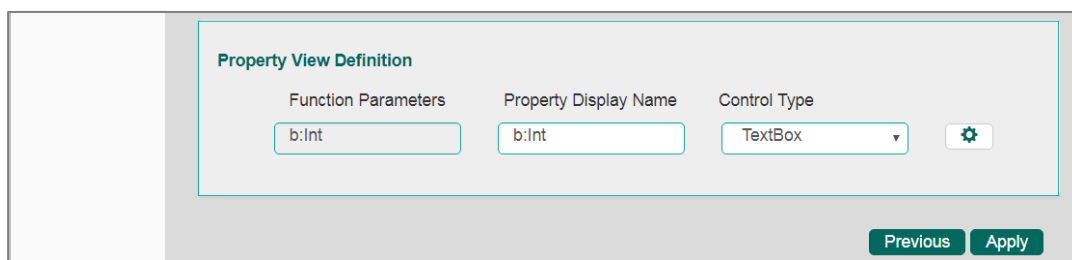
- ix) Users will be directed to the 'Settings' tab.
- x) Configure the following fields:
 - a. **Output Table Definition**

This option will configure a number of output columns, column headers, data types. Select any one out of the following options:

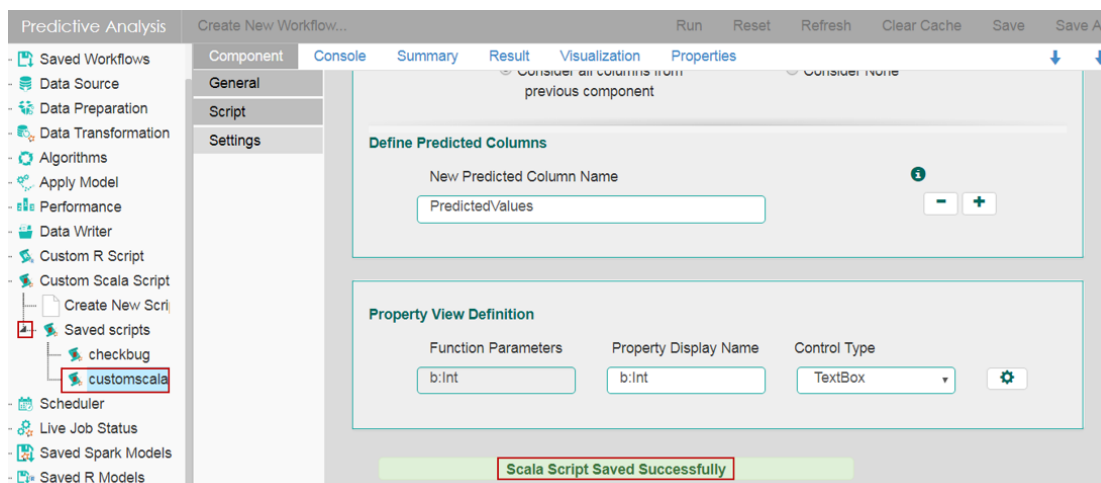
 - i. **Consider all columns from the previous component:** To display all columns from the previous component.
 - ii. **Consider None:** To display no column from the previous component.
 - b. **Define Predicted Columns**
 - i. **New Predicted Column Name:** Enter an appropriate name for the new predicted 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 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 input column. This field is associated with function parameters.
- xi) Click '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 saved in 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 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 above-mentioned 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.

13.2. Saved Scala Scripts

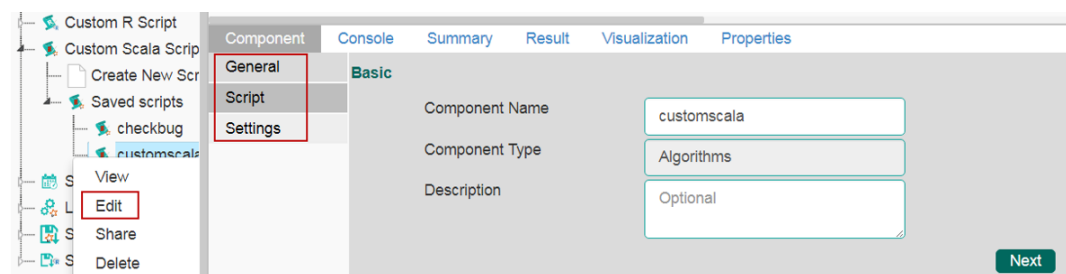
13.2.1. Viewing a Saved Scala Script

- 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 'View' option.
- v) Users will be redirected to the 'Component' tab.



13.2.2. 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.

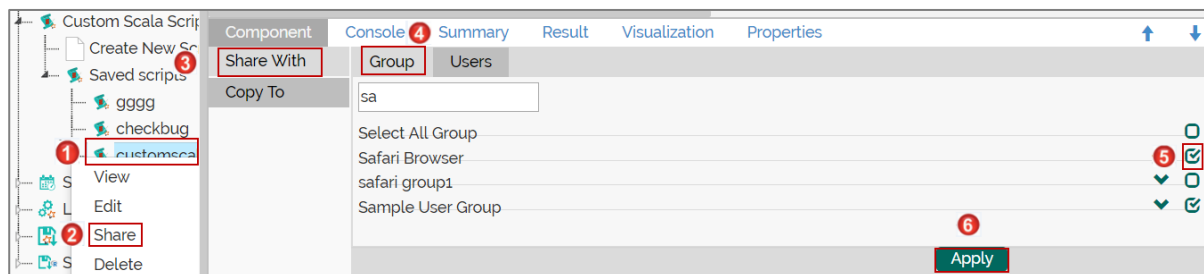


13.2.3. Sharing a Saved Scala Script

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'.
 - ii) Right-click on the selected Scala script.
 - iii) Select 'Share' from the context menu.
 - iv) The 'Share With' option will be displayed (by default).
 - v) Select either 'Group' or 'Users'.

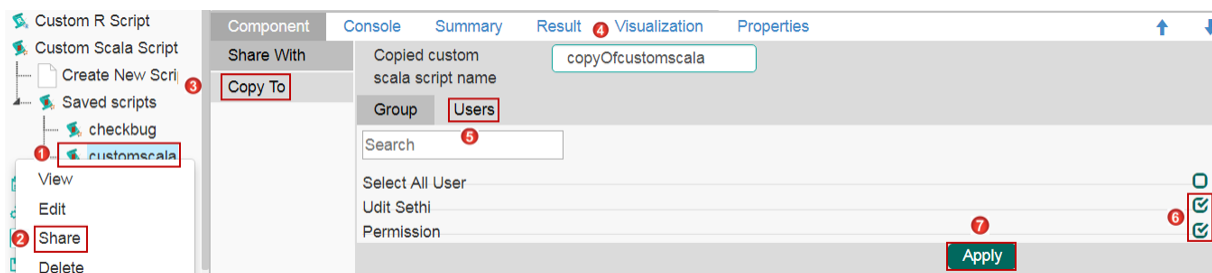
- a. By selecting a group all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
 - b. Users can be excluded by not selecting a username from the list when 'User' option has been selected.
- vi) Select a specific user or group from the list by check marking the box.
 - vii) Click 'Apply'



viii) The selected 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.

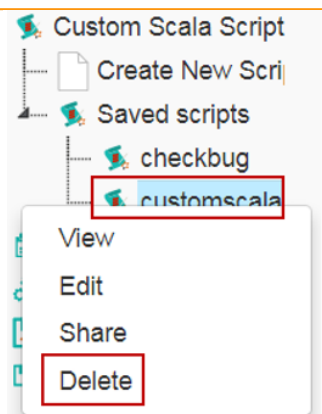
- i) Select a Scala script from the list of 'Saved Scripts'.
- ii) Right-click on the selected Scala script.
- iii) Select 'Share' from the context menu.
- iv) Select 'Copy To' option.
- v) The copied custom Scala script name will be displayed in a box.
- vi) Select either the 'Group' or 'Users' tab.
 - 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'



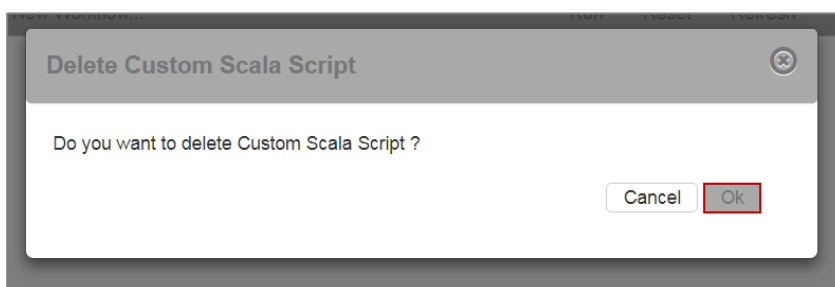
ix) The copied Scala script will be shared with the selected user(s)/group(s).

13.2.4. 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.



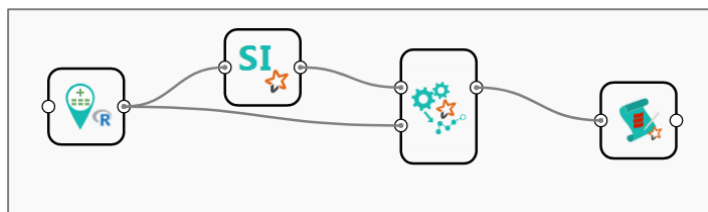
- v) A pop-up window will appear to assure the deletion.
- vi) Click 'Ok'



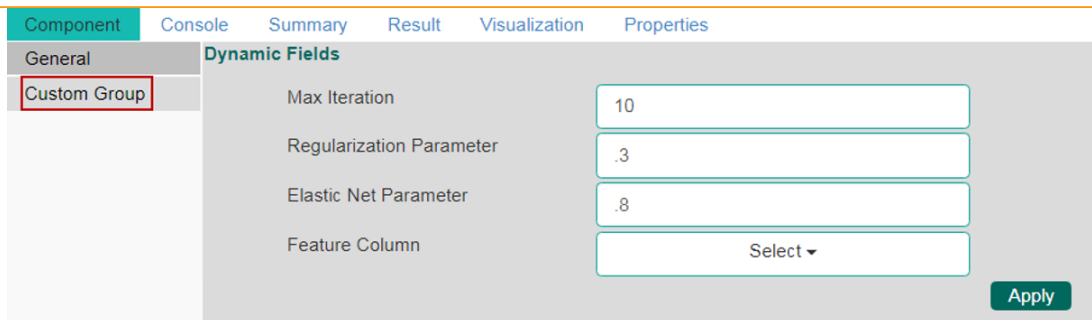
- vii) The selected Scala Script will be deleted.

13.2.5. Connecting Saved Scala Script with a Data Source

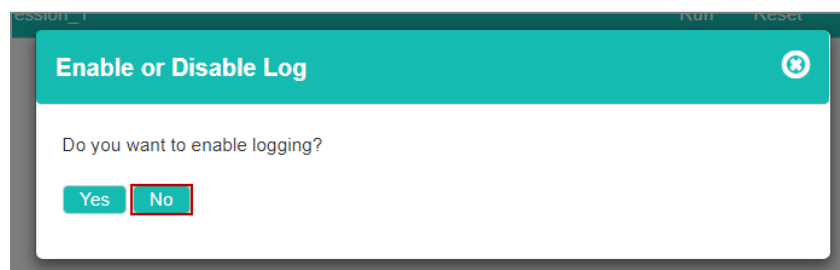
- i) Click the 'Custom Scala Script' tree node.
- ii) Select and drag a saved Scala script to the workspace.
- iii) Connect the Scala Script to a configured data source (Here, the used workflow has String Indexer and Spark Apply Model components connected with the Scala script component).



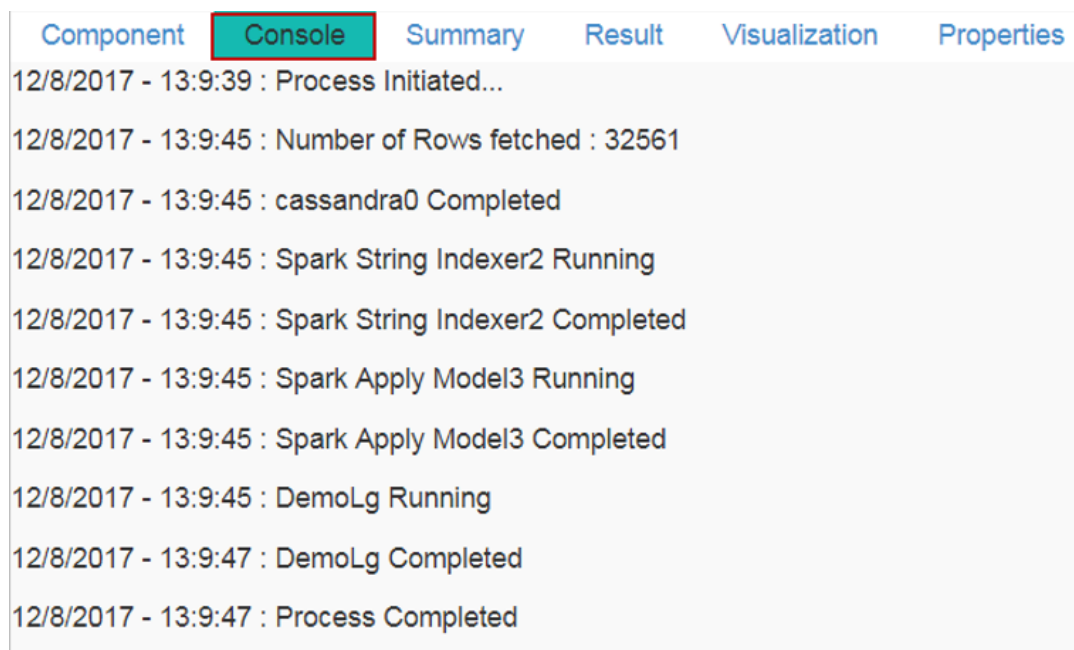
- iv) Click the dragged 'Scala Script' component.
- v) Configure the required fields in the 'Custom Group' tab.
- vi) Click 'Apply'



- vii) Click **'Run'**
- viii) A message will pop-up to confirm whether users want to enable logging.
- ix) Select **'No'**



- x) Users will be redirected to the **'Console'** tab.



- xi) 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.

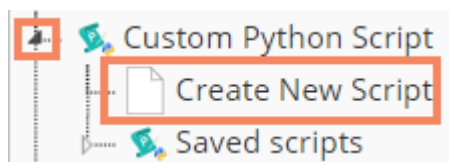
age	capital_gain	capital_loss	education_num	income	label	features
48	0	0	8	<=50K	0	{"values": [48]}
44	0	0	14	>50K	1	{"values": [44]}
46	0	0	14	<=50K	0	{"values": [46]}
40	0	0	10	>50K	1	{"values": [40]}
49	0	0	9	<=50K	0	{"values": [49]}
39	0	0	9	<=50K	0	{"values": [39]}
31	0	0	4	<=50K	0	{"values": [31]}
17	0	0	7	<=50K	0	{"values": [17]}
51	5178	0	12	>50K	1	{"values": [51]}
57	0	0	9	<=50K	0	{"values": [57]}

14. 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.

14.1. Creating a New Python Script

- i) Click ‘Custom Python Script’ tree-node on the Predictive Analysis home page.
- ii) Click ‘Create New Script’.

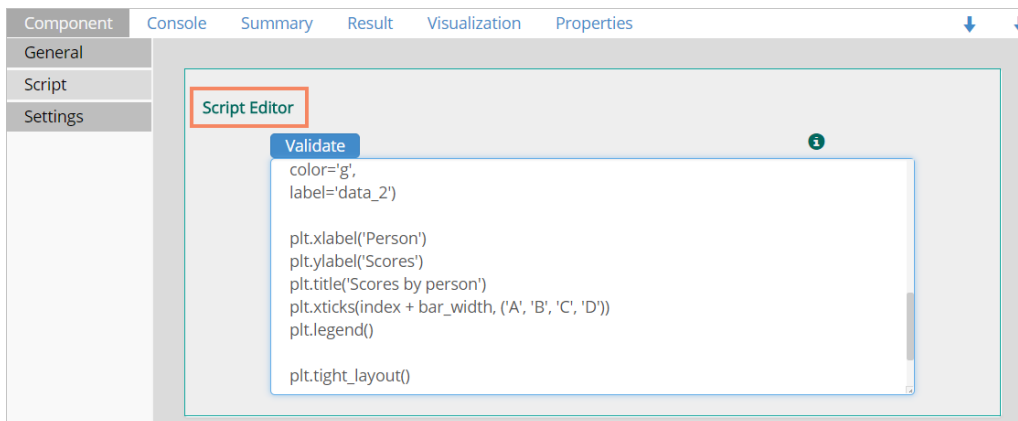


- iii) Users will be directed to the ‘Component’ tab.
- iv) Configure the following fields in the ‘General’ tab:
 - b. Basic
 - i. **Component Name:** Enter a name or title that you wish to give a saved Python Script.
 - ii. **Component Type:** Default Component type will be displayed in this field.
 - iii. **Description:** Describe the Component (It is an optional field).
- v) Click ‘Next’

Component	Console	Summary	Result	Visualization	Properties
General	Basic				
Script	Component Name				
Settings	Component Type				
	Description				
					Next

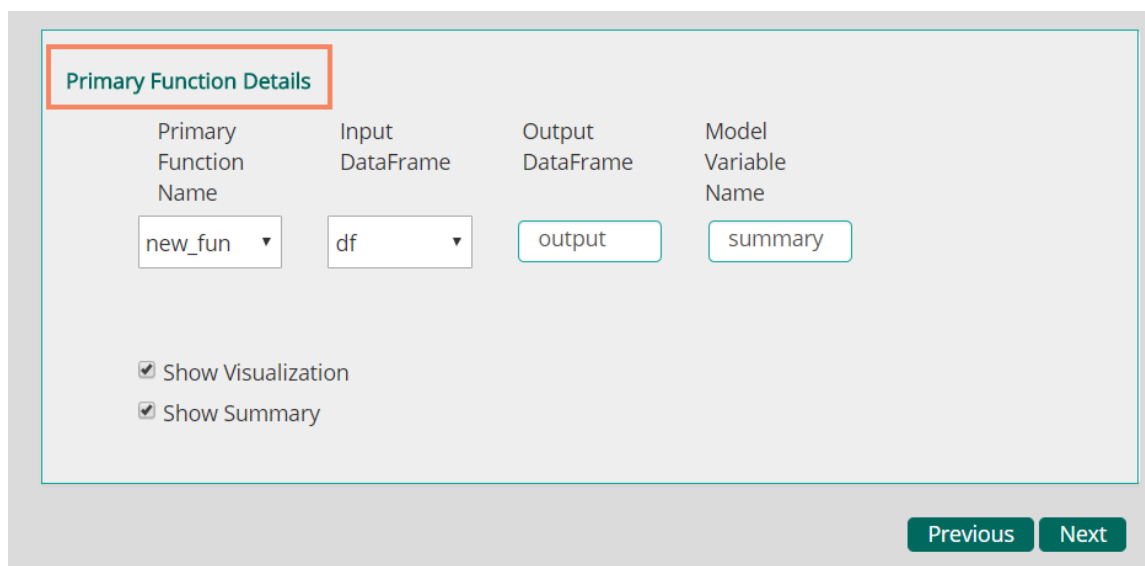
- vi) Users will be directed to the ‘Script’ tab.
- vii) Provide the following information:

- a. **Script Editor**
 - i. Write the python script in the given space under ‘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 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.



viii) Click ‘Next’ (Users can click ‘Previous’ if wish to open the previous page)

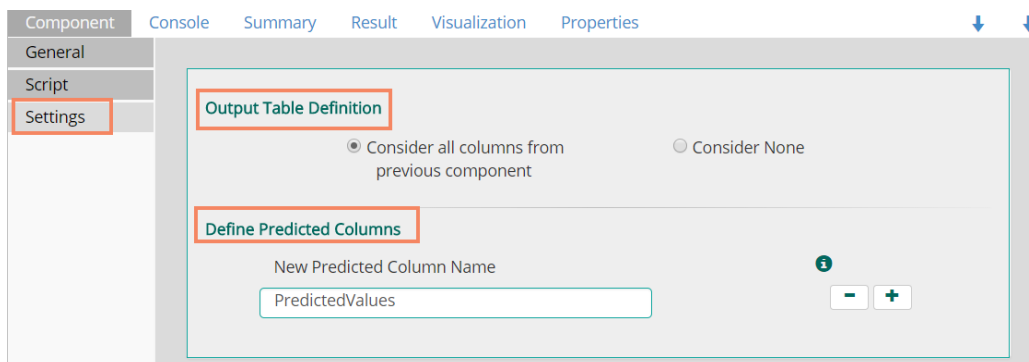


- ix) Users will be directed to the ‘Settings’ tab.
- x) Configure the following fields:
 - a. **Output Table Definition**


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 Predicted Columns**

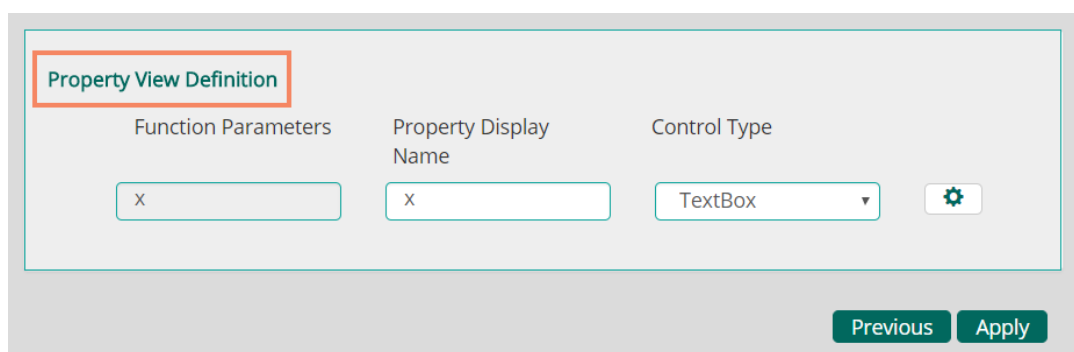
- i. **New Predicted Column Name:** Enter an appropriate name for the new predicted column.
- ii. : To remove the added row containing 'Data Type' and 'New Predicted Column Name'.
- iii. : To add a new row containing 'Data Type' and 'New Predicted Column Name'.



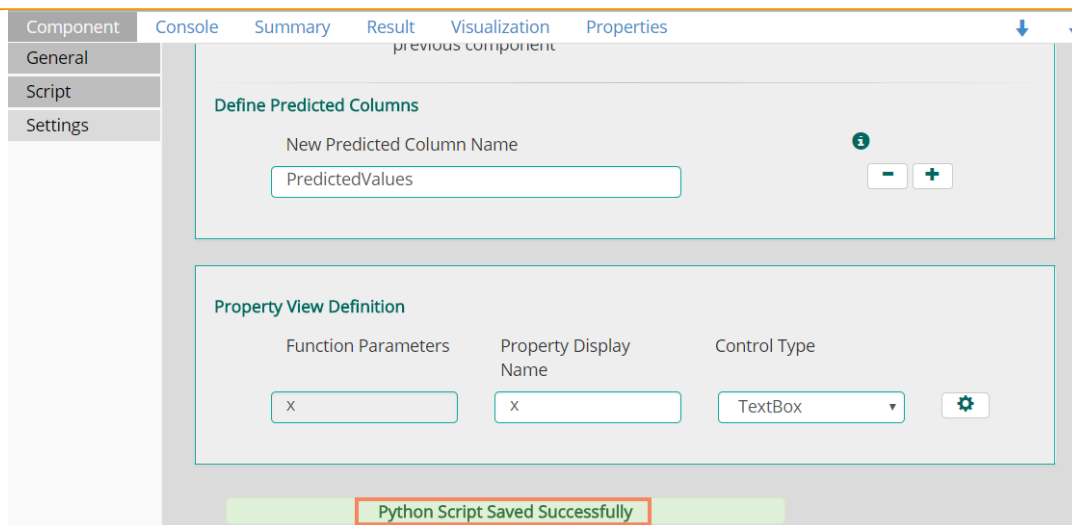
c. Property View Definition

- i. **Function Parameters:** Actual names of parameters configured in the script.
- ii. **Property Display Name:** Parameter name to be displayed while configuring 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 input column. This field is associated with function parameters.

xi) Click 'Apply'



xii) A message will pop-up to notify that the newly created Python script has been saved successfully.




xiii) 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 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. 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 clearly distinguish itself as a continuation line.
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 docstrings/comments to 72).
7. Do not use "type" as the function argument. As it is 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 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.

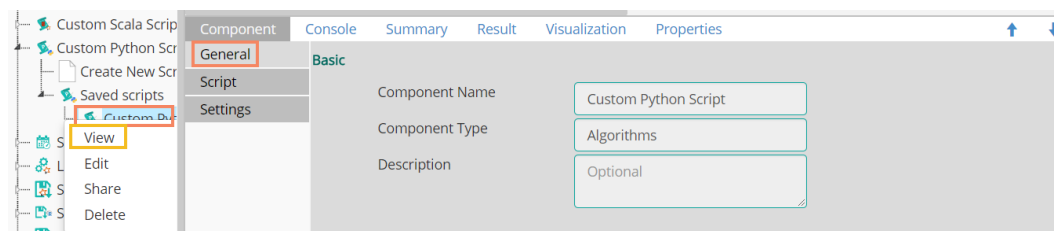
Note:

- a. Click the ‘Information’ button  to get the above-mentioned 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.

14.2. Saved Python Scripts

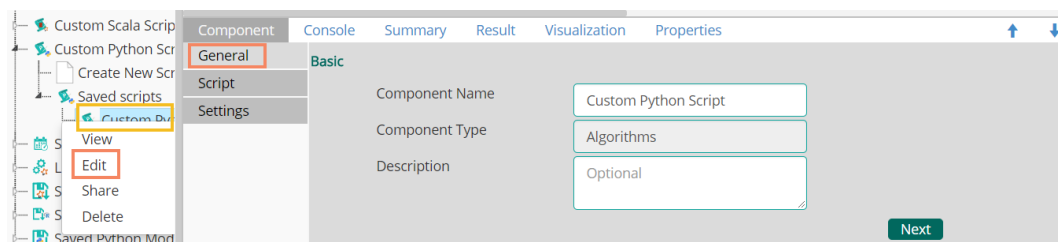
14.2.1. Viewing a Saved Python Script

- 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 ‘**View**’ option.
- v) Users will be redirected to the ‘**Component**’ tab.



14.2.2. Editing a Saved Python Script

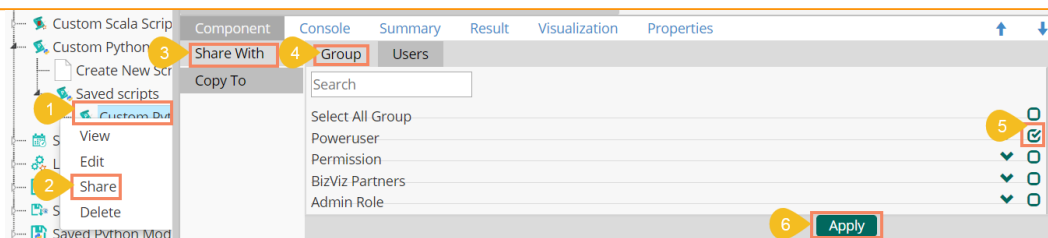
- i) Select a Scala Script from the list of ‘**Saved Scripts**’ list.
- ii) Right-click on the selected Python Script.
- iii) A context menu will open.
- iv) Select ‘**Edit**’.
- v) Users will be redirected to the ‘**Component**’ tab.
- vi) Users can edit the required fields provided under **General**, **Script**, and **Settings** tabs.



14.2.3. 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 ‘**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**’

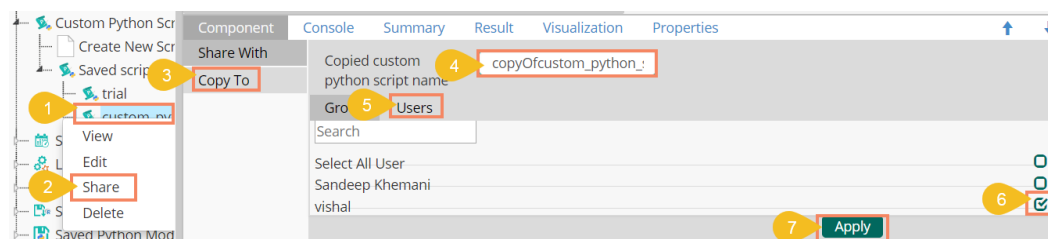


viii) The selected Python script will be shared with the chosen user(s)/group(s).

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 ‘Group’ option has been selected.
 - b. Users can be excluded by not selecting a username from the list when ‘Users’ option has been selected.

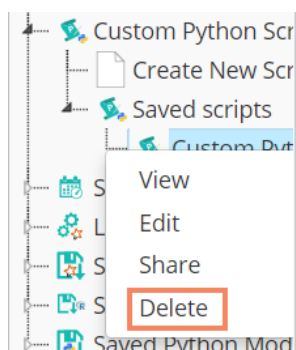
vii) Select a specific user or group from the list by check marking the box.

viii) Click ‘Apply’

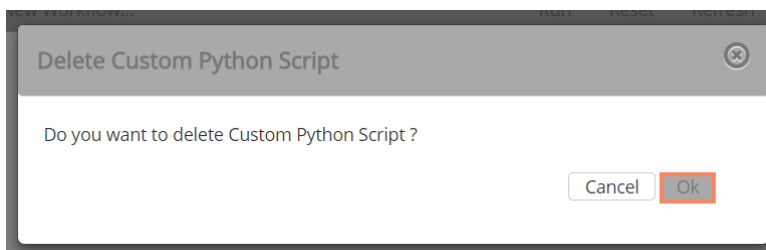


14.2.4. Deleting a Saved Python Script

- 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 ‘Delete’ option.



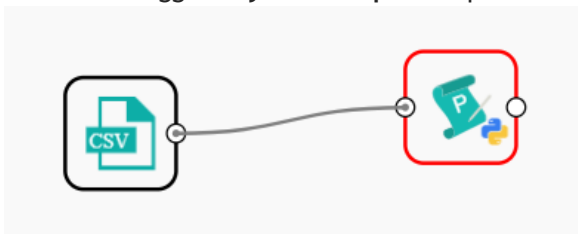
- v) A pop-up window will appear to assure the deletion.
- vi) Click 'Ok'



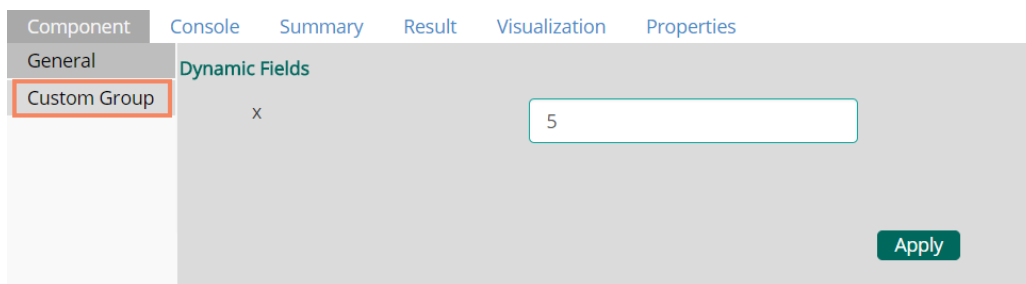
- vii) The selected Scala Script will be deleted.

14.2.5. Connecting Saved Python Script with a Data Source

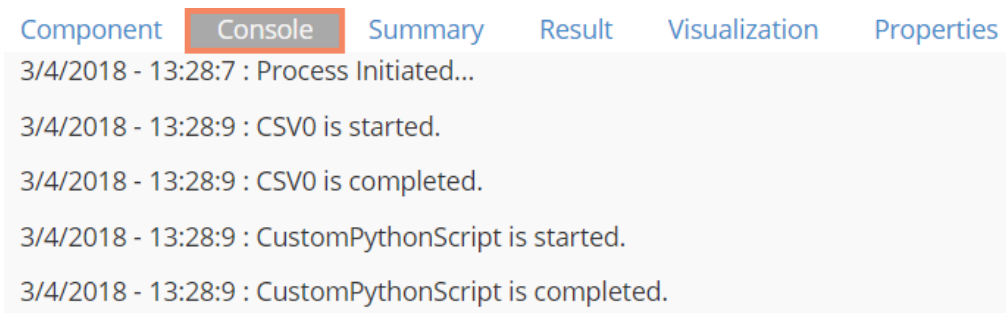
- i) Click the 'Custom Python Script' tree node.
- ii) Select and drag a saved Python script to the workspace.
- iii) Connect the Python Script to a configured data source.
- iv) Click the dragged 'Python Script' component.



- v) Configure the required fields in the 'Custom Group' tab.
- vi) Click 'Apply'



- vii) Click 'Run' or 'Refresh' to run the process.
- viii) Users will be redirected to the 'Console' tab.



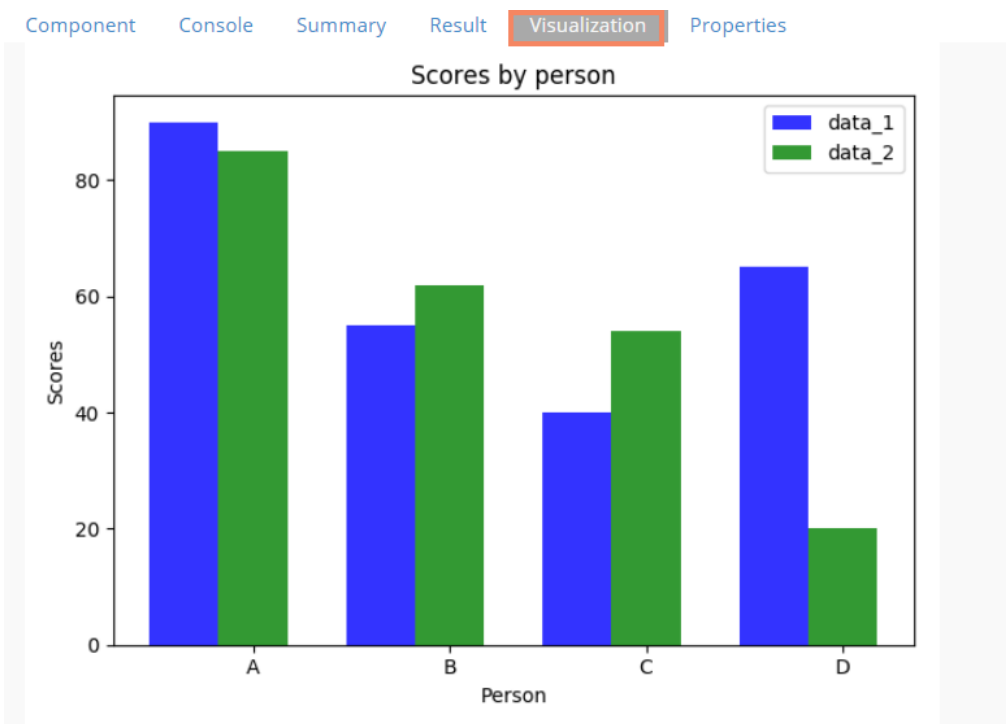
- ix) Follow the below given steps to display the result view:
 - a. Click the dragged Python component on the workspace.
 - b. Click the 'Result' tab.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

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

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



- x) Click the 'Visualization' tab to display the result data through a column chart.

xi) Click 'Summary' tab to view summary of the process.

	Component	Console	Summary	Result	Visualization	Properties
		Number	PetalLength	PetalWidth	SepalLength	SepalWidth
count	150.000000	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	3.758000	1.199333	5.843333	3.057333	
std	43.445368	1.765298	0.762238	0.828066	0.435866	
min	1.000000	1.000000	0.100000	4.300000	2.000000	
25%	38.250000	1.600000	0.300000	5.100000	2.800000	
50%	75.500000	4.350000	1.300000	5.800000	3.000000	
75%	112.750000	5.100000	1.800000	6.400000	3.300000	
max	150.000000	6.900000	2.500000	7.900000	4.400000	

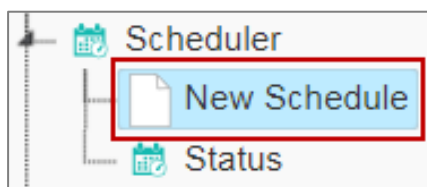
15. Scheduler

Scheduler helps to schedule the Predictive Workflow as per the requirement.

15.1. New Schedule

This section explains steps to schedule a new job. Scheduling 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.

15.1.1. 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.

15.1.2. 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'

- 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.

15.1.3. 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. Elastic Search 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.

15.1.4. 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 greater than the **Current System Date and Time**)
- ii) **Select a Job Refresh Interval option:**
E.g. When 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 greater than the current system time.
- iv) **End Date:** Select an end date and time for the scheduled job. (It should be greater 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.

15.1.4.1. 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.

The screenshot shows a configuration window with a sidebar on the left containing tabs: Component, Console, Summary, Result, Visualization, and Properties. The 'Console' tab is active, and within it, the 'Schedule' sub-tab is selected. The main area shows scheduling options: 'Hourly' is selected under the refresh interval options, with a dropdown menu showing '1' hour(s). Other options include 'At 12:00' and 'Run Now' (checked). The 'Start Date' is set to 'Wed Apr 04 2018 18:00:0' and the 'End Date' is 'Thu Apr 05 2018 06:00:0'. A 'Next' button is located at the bottom right.

- **Daily:** By selecting this option users can schedule the job on 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 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)

Component	Console	Summary	Result	Visualization	Properties
General	Hourly	Daily	Weekly	Monthly	Yearly
Data Source	Custom Cron Expression				
Data Writer					
Schedule	Start Date <input type="text" value="Wed Apr 04 2018 18:00:0"/>				
Notification	<input checked="" type="radio"/> Day <input type="text" value="1"/> of every <input type="text" value="1"/> month(s) <input type="radio"/> The First <input type="text" value="1"/> Monday <input type="text" value="1"/> of every <input type="text" value="1"/> month(s)				
	Start Time <input type="text" value="12"/> <input type="text" value="00"/>				
	End Date <input type="text" value="Thu Aug 30 2018 02:00:0"/>				
	<input checked="" type="checkbox"/> Run Now				
	<input type="button" value="Next"/>				

- **Yearly:** By selecting this option users can schedule the job on a yearly basis. This time range is provided for jobs running 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)

Component	Console	Summary	Result	Visualization	Properties
General	Hourly	Daily	Weekly	Monthly	Yearly
Data Source	Custom Cron Expression				
Data Writer					
Schedule	Start Date <input type="text" value="Wed Apr 04 2018 18:00:0"/>				
Notification	<input checked="" type="radio"/> Every <input type="text" value="January"/> <input type="text" value="1"/> <input type="radio"/> The First <input type="text" value="1"/> Monday <input type="text" value="1"/> of <input type="text" value="January"/>				
	Start Time <input type="text" value="12"/> <input type="text" value="00"/>				
	End Date <input type="text" value="Wed Jan 02 2019 01:00:0"/>				
	<input checked="" type="checkbox"/> Run Now				
	<input type="button" value="Next"/>				

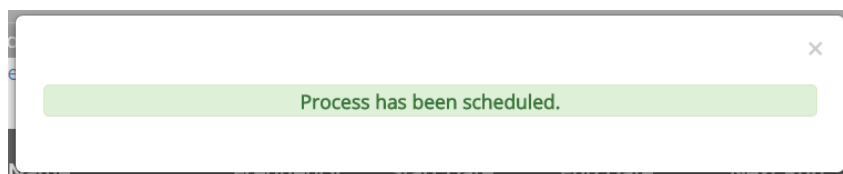
Note: By selecting the ‘Use Existing Data Connector’ and ‘Use Existing Data Writer’ options ‘Schedule’ tab will be displayed immediately after the ‘General’ tab.

15.1.5. 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 R 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 successfully 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.



- iv) The scheduled job/ process will be added to a list provided under the **'Status'** tab.

Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
importantvar	Hourly	8/Mar/2018-15:0:0	8/Mar/2018-16:0:0	NA	Stopped	Prakash	REC_VariableImportance_Attrition	Hiring_Data	View Logs	
salaryvsexp	Hourly	15/Mar/2018-15:0:0	15/Mar/2018-16:0:0	NA	Stopped	Prakash	sampleBS	salaryvsexp	View Logs	
scheduler_job	Yearly	4/Apr/2018-18:0:0	2/Jan/2019-1:0:0	1/Jan/2019-12:0:0	Active	Will	Scheduler_workflow	iris	View Logs	
PAInBS	Hourly	21/Feb/2018-20:0:0	21/Feb/2018-22:0:0	NA	Stopped	Prakash	forBS	Hiring	View Logs	

Note:

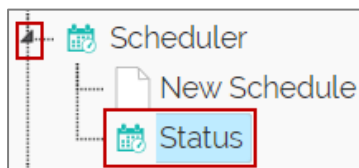
- a. The PDF summary will be sent through email for the scheduled workflows.
- b. Multiple email addresses can be entered in coma separated value.

c. At present, Spark Workflows are not supported by Scheduler.

15.2. 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.




Task Name	Frequency	Start Date	End Date	Next Run	Status	Scheduled By	Workflow Name	Data Source	Logs	Actions
importantvar	Hourly	8/Mar/2018-15:0:0	8/Mar/2018-16:0:0	NA	Stopped	Prakash	REC_Variableimportance_Attrition	Hiring_Data	View Logs	
salaryvexp	Hourly	15/Mar/2018-15:0:0	15/Mar/2018-16:0:0	NA	Stopped	Prakash	sampleBS	salaryvexp	View Logs	
scheduler_job	Yearly	4/Apr/2018-18:0:0	2/Jan/2019-1:0:0	NA	Stopped	Will	Scheduler_workflow	iris	View Logs	
PAinBS	Hourly	21/Feb/2018-20:0:0	21/Feb/2018-22:0:0	NA	Stopped	Prakash	forBS	Hiring	View Logs	
OptimizedCLV	Hourly	22/Feb/2018-17:0:0	22/Feb/2018-18:0:0	NA	Stopped	Avin Jain	OptimizedCLV_1	CLV_dataset	View Logs	
HIRING_PA_DATASTORE	Hourly	24/Feb/2018-14:0:0	25/Feb/2018-17:0:0	NA	Stopped	Prakash	REC_Variableimportance_Attrition	Hiring_Data	View Logs	

- a. Click 'View Logs' to see the logs of the selected workflow under the 'Component' tab.

Component	Console	Summary	Result	Visualization	Properties
16/Aug/2017 - 08:40:0	DataReaderProcess is started.				
16/Aug/2017 - 08:40:2	Number of Rows fetched : 50				
16/Aug/2017 - 08:40:2	DataReaderProcess is completed.				
16/Aug/2017 - 08:40:2	Data Type Definition1 is started.				
16/Aug/2017 - 08:40:2	Data Type Definition1 is completed.				
16/Aug/2017 - 08:40:2	DataWriterProcess is started.				
16/Aug/2017 - 08:40:5	DataWriterProcess 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

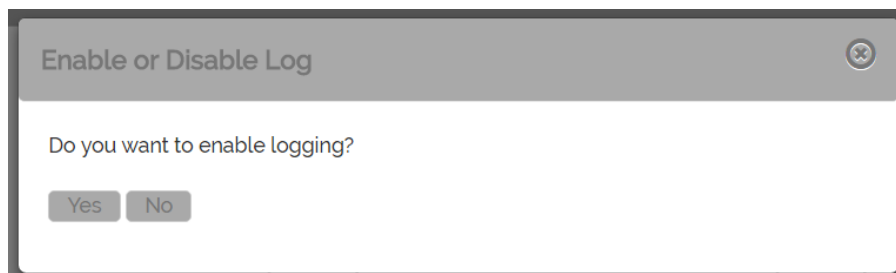
Note:

- a. 'Edit' option will allow the user to update/ edit all the tabs for the selected job.
- b. Users can click '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.

16. 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) Click 'Run'
- 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.
- vi) Click the 'Spark' leaf node.
- vii) Users will be redirected to the 'Status' tab.

The screenshot shows the 'Predictive Analysis' interface. The 'Status' tab is selected, displaying a table of workflows. The table has columns for Workflow Name, Run by, Start time, End Time, Status, View Log, Live job status, Summary, and Actions. Two workflows are listed: 'wtfinal' and 'checksparkpa', both with a status of 'success'. The 'View Log' icon (an eye) is visible for each workflow. The 'Live Job Status' icon (an eye with a red dot) is also visible in the left sidebar.

Workflow Name	Run by	Start time	End Time	Status	View Log	Live job status	Summary	Actions
wtfinal	Ranjit Krishnan	14/Aug/2017-13:21:9	14/Aug/2017-13:21:29	success				
checksparkpa	Ranjit Krishnan	14/Aug/2017-13:17:5	14/Aug/2017-13:17:13	success				

- a. **View Log:** log of the completed workflow can be viewed under the 'Console' tab by clicking the 'View Log' icon

The screenshot shows the 'Console' tab selected, displaying the execution log for a workflow. The log entries are as follows:

```

14/8/2017 - 13:17:5 : Process started
14/8/2017 - 13:17:6 : cassandrao Running
14/8/2017 - 13:17:10 : cassandrao Completed
14/8/2017 - 13:17:11 : Spark-Decision-Tree1 Running
14/8/2017 - 13:17:12 : Spark-Decision-Tree1 Completed
14/8/2017 - 13:17:12 : Spark Apply Model2 Running
14/8/2017 - 13:17:13 : Spark Apply Model2 Completed
14/8/2017 - 13:17:13 : Execution completed
  
```

- 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.

The screenshot shows the 'Console' tab selected, displaying the live job status for a workflow. The log entries are as follows:

```

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
wtfinal	Ranjit Krishnan	17/Aug/2017-17:0:8	17/Aug/2017-17:0:12	cancelled				
wtfinal	Ranjit Krishnan	17/Aug/2017-16:50:20	NA	in progress				

Showing 1 to 2 of 2 entries Previous 1 Next

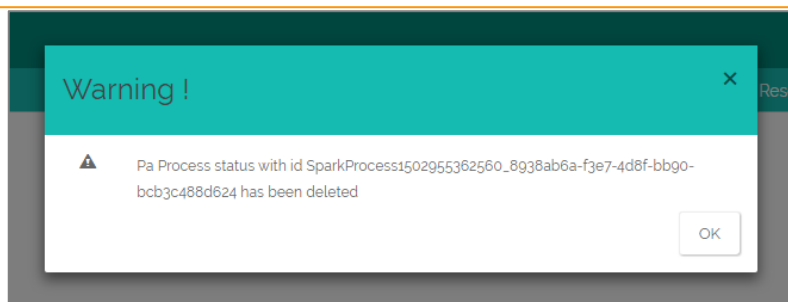
- ii. **Delete:** Click the 'Delete' icon to remove an execution.

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
checksparkpa	Ranjit Krishnan	17/Aug/2017-13:6:3	NA	in progress				
untitled	Ranjit Krishnan	17/Aug/2017-13:0:7	17/Aug/2017-13:0:15	success				
DT_check	Ranjit Krishnan	17/Aug/2017-12:51:48	17/Aug/2017-12:51:55	failed				
untitled	Ranjit Krishnan	17/Aug/2017-12:51:10	17/Aug/2017-12:51:17	success				
sparkser	Ranjit Krishnan	17/Aug/2017-12:51:3	17/Aug/2017-12:51:14	failed				
untitled	Ranjit Krishnan	17/Aug/2017-12:50:49	17/Aug/2017-12:50:56	failed				
wtfinal	Ranjit Krishnan	17/Aug/2017-12:47:58	17/Aug/2017-12:48:23	success				

The selected workflow will be removed from the 'Live Job Status' table and a warning message will be displayed to convey the same.



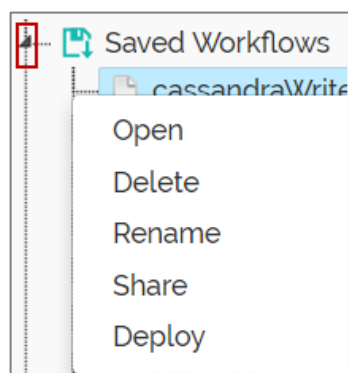
Note:

- 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.

17. Saved Workflows

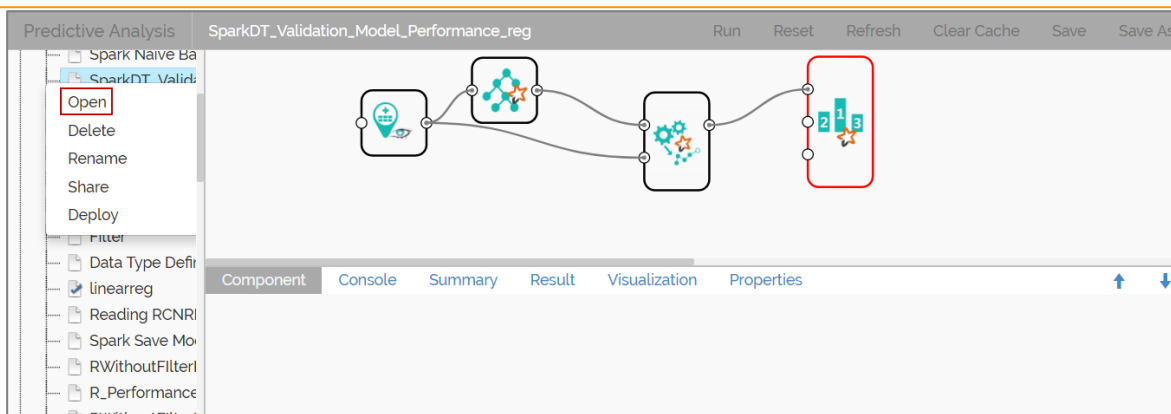
Users can save a workflow by clicking the 'Save' button provided on the workspace menu row. All the saved workflows will be displayed under the 'Saved Workflow' tree node. This section explains various options assigned to a saved workflow.

- 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):

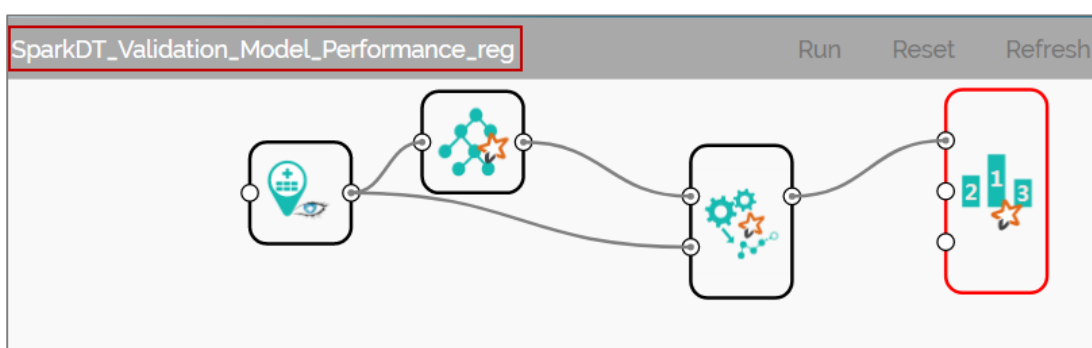


17.1. Opening a Workflow

- 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.

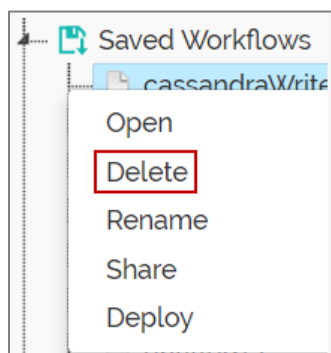


Note: The workflow name will be displayed on the left side of the workspace menu row while opening a workflow.

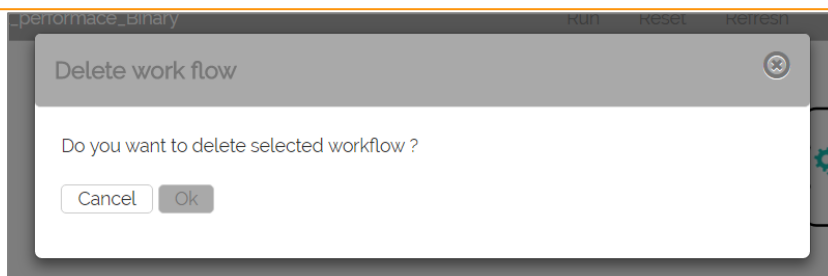


17.2. Deleting a Workflow

- i) Right-click on a workflow from the list of 'Saved Workflows'
- ii) Select 'Delete' from the context menu.



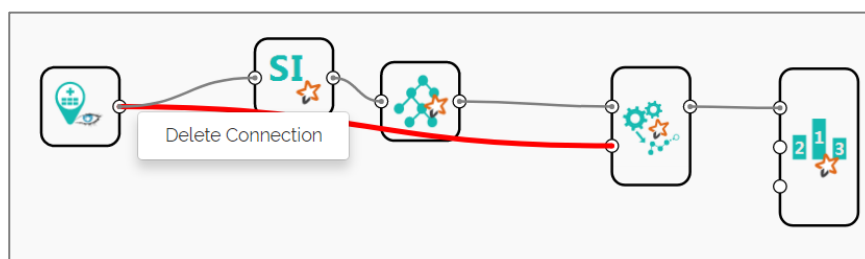
- iii) A message window will pop-up to confirm the deletion.
- iv) Click 'OK'



v) The selected workflow will be removed from the list.

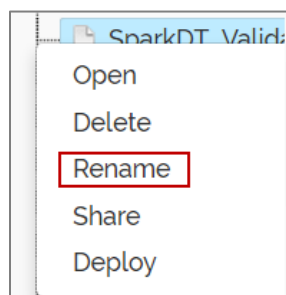
17.2.1. Delete Connection to a Workflow

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.

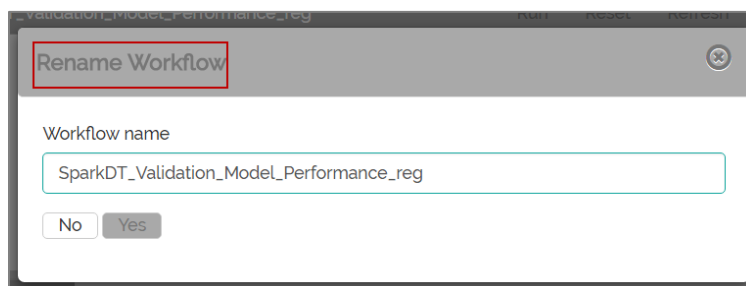


17.3. Renaming a Workflow

- 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'



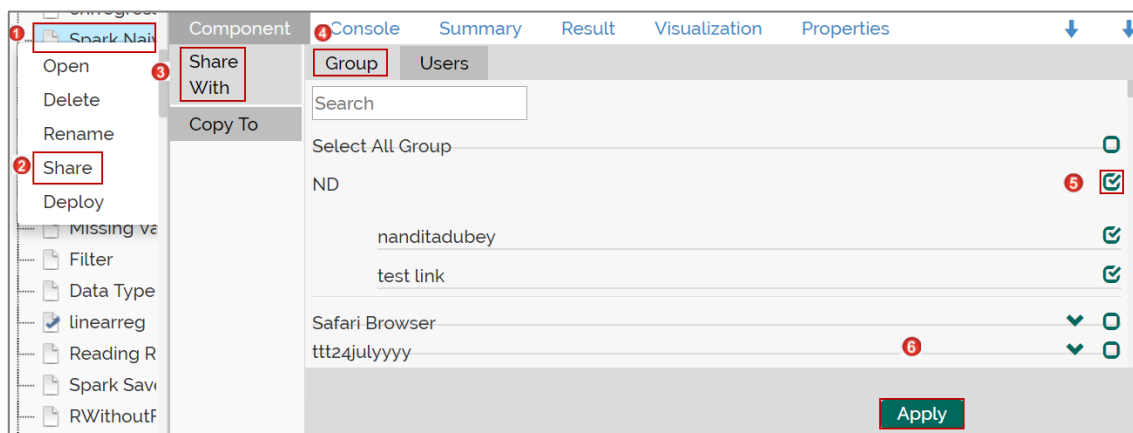
vi) The selected workflow will be renamed.

17.4. Sharing a Workflow

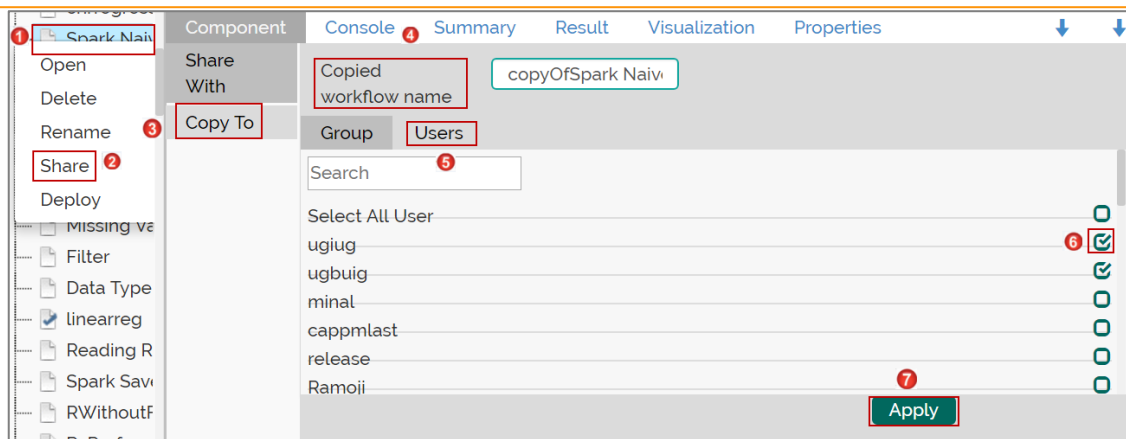
This feature gives users the ability to share saved workflows with other users and groups.

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 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).
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 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'**

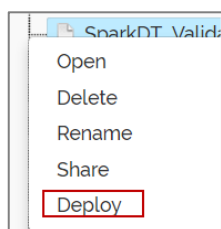


viii) The copied workflow will be shared with the chosen users/groups.

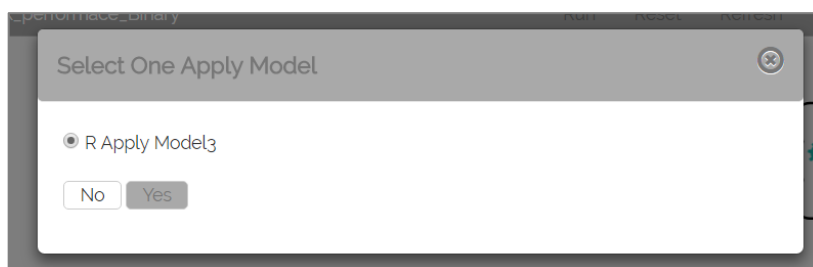
17.5. Deploying a Workflow

The Predictive Workflows can be deployed to the BizViz Dashboard Designer.

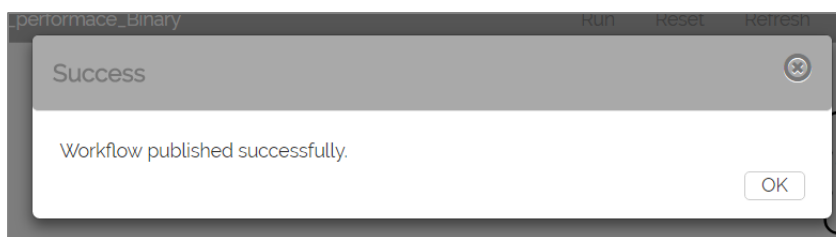
- i) Press a right click on a Workflow from the list of 'Saved Workflows'
- ii) Select 'Deploy Workflow' from the context menu.



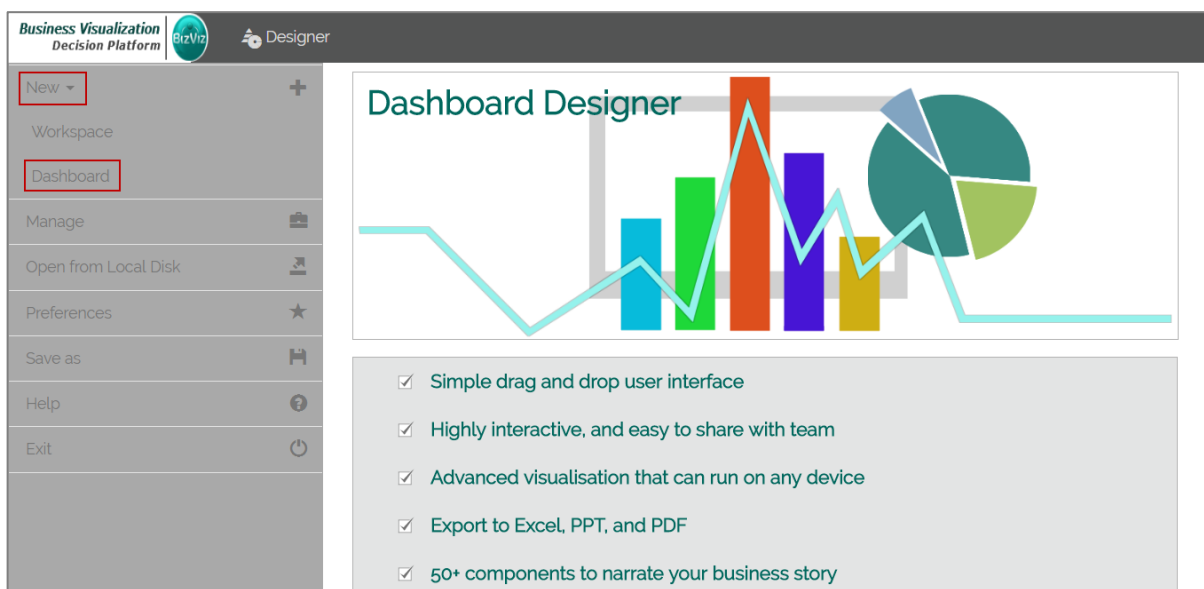
- iii) Users will be redirected to select an Apply Model component from the workflow.
- iv) Select an Apply Model component and click 'Yes'





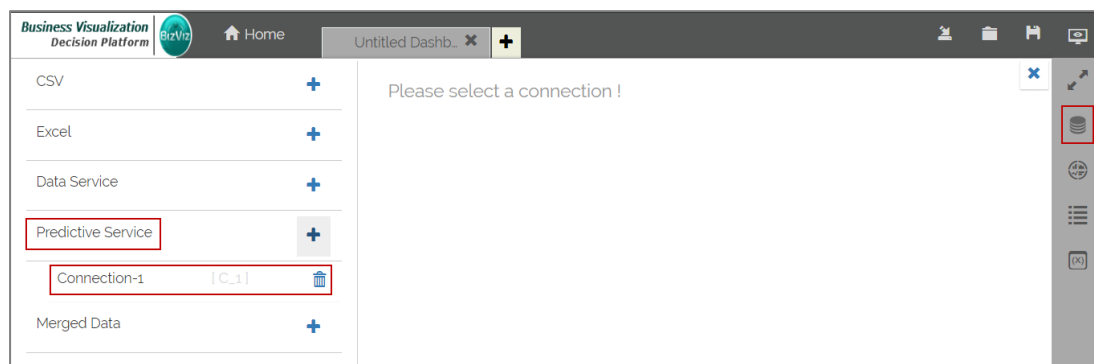
- v) A success message will pop-up to assure that the workflow has been published.



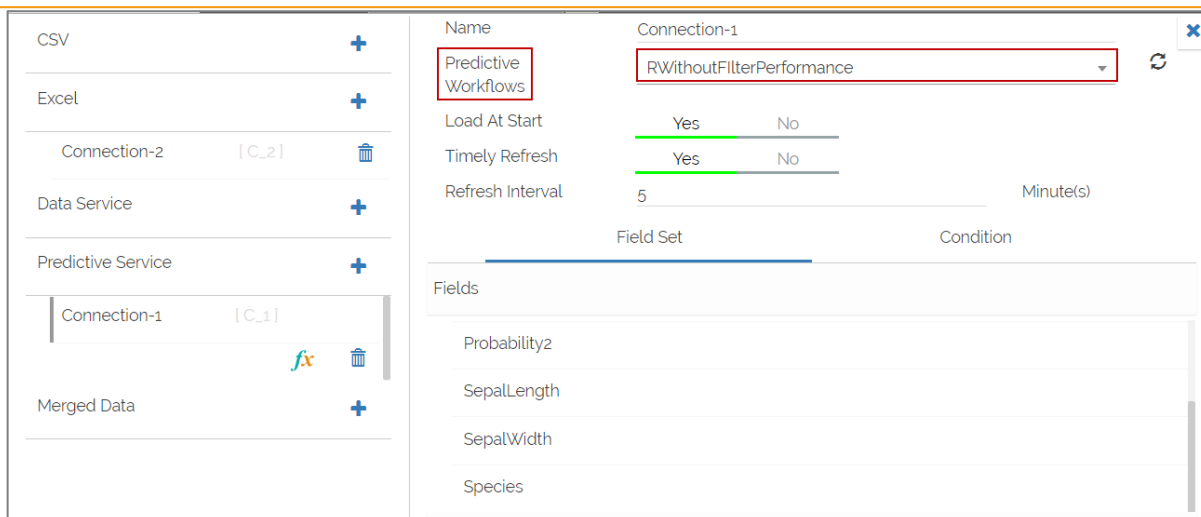
- vi) Navigate to the Dashboard Designer home page.
- vii) Click 'New'
- viii) Click 'Dashboard'



- ix) Users will be directed to the Dashboard canvas.
- x) Click the 'Data Source' icon  to display all the available data sources.
- xi) Click the 'Create New Connection' option  provided next to the 'Predictive Service' data source.
- xii) A new connection will be created and added below.



- xiii) Click on the connection to display the connection specific details.
- xiv) Select the deployed Predictive workflow as a data source via the drop-down menu.
- xv) 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.

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.
- **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.
- **Python Workflows:**
 - The result set from the ‘**Apply Model**’ component within a deployed Python 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 Python workflow must contain one Apply model, read model (Saved Model component), and Spark filter (optional) component to deploy the workflow.

Note:

- a. If a deployed Predictive Workflow has summary, it can be viewed using the Dashboard Designer tool.
- b. Users can view the result of each component in the spark workflow.
 - 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.

ClusterNumber	PetalLength	PetalWidth	SepalLength	SepalWidth
1	5.8	2.2	6.5	3
3	4.6	1.3	6.6	2.9
3	4.7	1.2	6.1	2.8
3	5.1	1.9	5.8	2.7
5	1.5	0.2	5	3.4

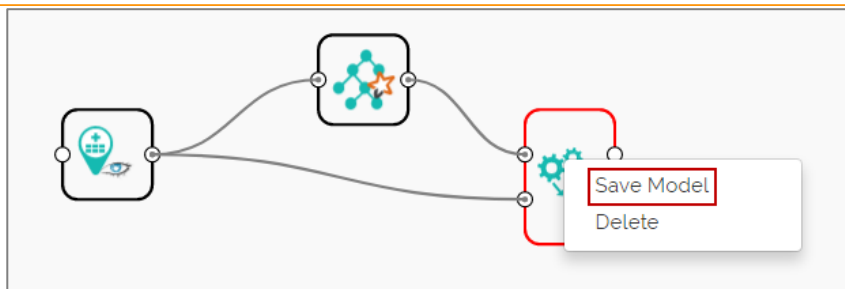
- c. Users can stop an ongoing Spark workflow execution by clicking the ‘Stop’ button on the progress bar.

18. Saved Spark Models

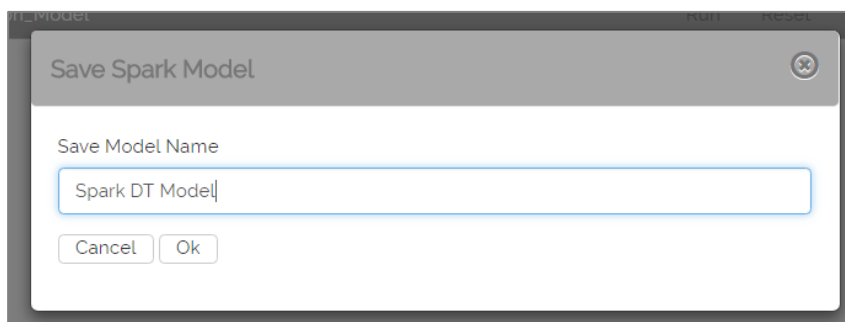
A model is a reusable component created by training an algorithm using historical data and saving the instance. The ‘Saved Spark Models’ tree-node contains a list of all the saved predictive models.

18.1. Saving a Spark Model

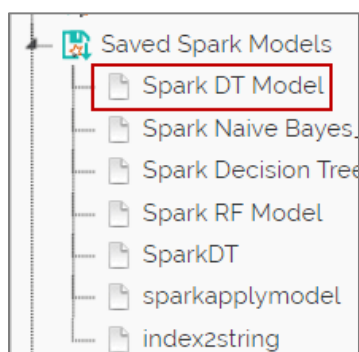
- i) Open a spark workflow.
- ii) Connect ‘Apply Model’ component with the workflow (as shown below).
- iii) Right-click on the ‘Apply Model’ component.
- iv) A context menu will open.
- v) Select ‘Save Model’



- vi) A pop-up window will appear.
- vii) Enter a name for the model that you wish to save.
- viii) Click 'OK'



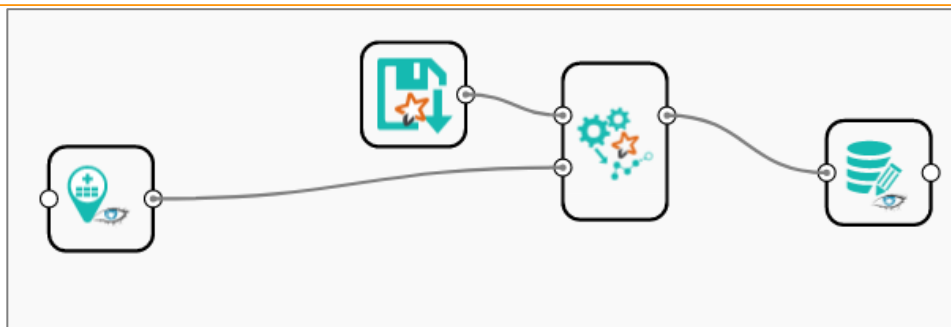
- ix) The created Predictive Model will be saved to the 'Saved Spark Models' list.



18.2. Reading a Spark Model

Users can drag a saved model to the workspace and reuse the model for a 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
- v) Configure the following fields in 'General':

Component	Console	Summary	Result	Visualization	Properties
General	<p>Basic</p> <p>Component Name: Spark Saved Model</p> <p>Alias: Spark DT Model3</p> <p>Description: Optional</p> <p style="text-align: right;">Apply</p>				
Summary					

- vi) Click the 'Summary' tab.

Component	Console	Summary	Result	Visualization	Properties
General	<p>Summary</p> <p>----- Summary of the model -----</p> <p>Impurity = gini maxBins = 32 maxDepth = 5 labelCol = binarycolumn featuresCol = dfFeaturesCol1</p> <p style="text-align: right;">Apply</p>				
Summary					

- vii) Click 'Run'
- viii) Users will be redirected to the 'Console' tab.

```

Component Console Summary Result Visualization Properties
17/8/2017 - 19:29:26 : Process Initiated...
17/8/2017 - 19:29:26 : SparkDT2 Completed
17/8/2017 - 19:29:26 : cassandra0 Running
17/8/2017 - 19:29:32 : Number of Rows fetched : 150
17/8/2017 - 19:29:32 : cassandra0 Completed
17/8/2017 - 19:29:32 : Spark Apply Model1 Running
17/8/2017 - 19:29:32 : Spark Apply Model1 Completed
17/8/2017 - 19:29:32 : Process Completed
  
```

- ix) Follow the below given steps to display Result.
 - a. Click Apply model component.
 - b. Click the 'Result' tab.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	dfFeaturesCol1	rawPrediction1	probability1	binarycolumn	prediction1
4.9	1.8	6.3	2.7	["values":["4.9,1.8,6.3,2.7]]	["values":["100.0]]	["values":["1.0]]	0	0
1.7	0.2	5.4	3.4	["values":["1.7,0.2,5.4,3.4]]	["values":["0.50]]	["values":["0.1]]	1	1
1.4	0.2	5.1	3.5	["values":["1.4,0.2,5.1,3.5]]	["values":["0.50]]	["values":["0.1]]	1	1
1.5	0.4	5.7	4.4	["values":["1.5,0.4,5.7,4.4]]	["values":["0.50]]	["values":["0.1]]	1	1
1.9	0.2	4.8	3.4	["values":["1.9,0.2,4.8,3.4]]	["values":["0.50]]	["values":["0.1]]	1	1
4.7	1.5	6.7	3.1	["values":["4.7,1.5,6.7,3.1]]	["values":["100.0]]	["values":["1.0]]	0	0
6.3	1.8	7.3	2.9	["values":["6.3,1.8,7.3,2.9]]	["values":["100.0]]	["values":["1.0]]	0	0
1.1	0.1	4.3	3	["values":["1.1,0.1,4.3,3]]	["values":["0.50]]	["values":["0.1]]	1	1
4.3	1.3	6.2	2.9	["values":["4.3,1.3,6.2,2.9]]	["values":["100.0]]	["values":["1.0]]	0	0
5.1	1.9	5.8	2.7	["values":["5.1,1.9,5.8,2.7]]	["values":["100.0]]	["values":["1.0]]	0	0

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

- x) Click the 'Properties' tab to display the model properties.

Component Console Summary Result Visualization **Properties**

Created By	
Created At	2017-07-27 15:23:44 +0530
Last Modified By	
Last Modified At	2017-07-27 15:23:44 +0530
Version	3.0.0

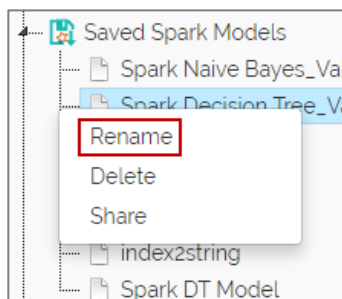
Note:

- a. To run the workflow with a 'Saved Model' component, it is mandatory 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 that contains a saved model.

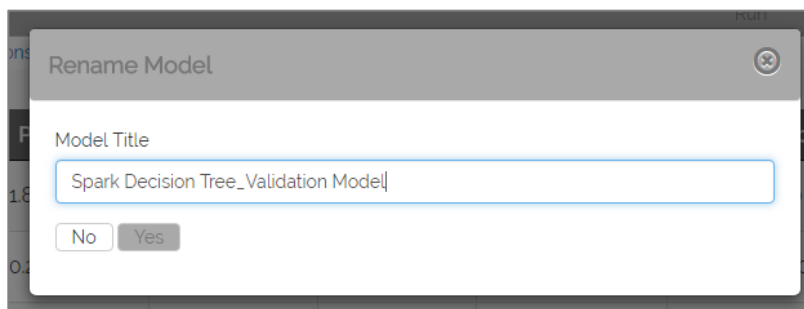
- c. Currently, only Spark trained Workflows can be saved to the 'Saved Models' tree-node.

18.2.1. 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.



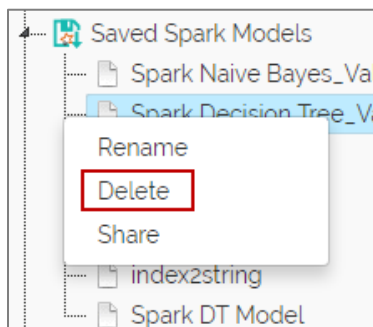
- v) A pop-up window will appear to rename the model.
- vi) Enter a new 'Model Title' or modify the existing model title in the given field (if desired).
- vii) Click 'Yes'



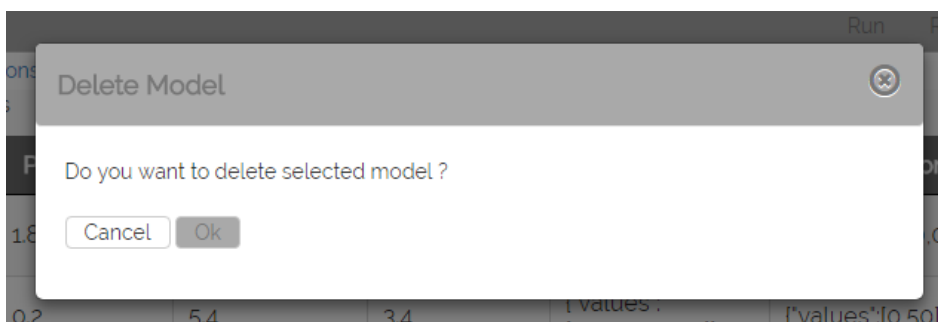
- viii) The selected Spark Predictive Model will be renamed.

18.2.2. 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'



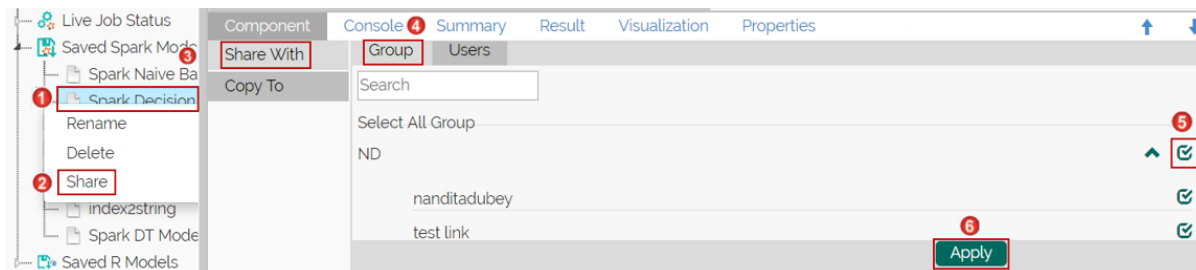
- vii) The selected predictive model will be deleted and removed from the list of 'Saved Spark Models'

18.2.3. 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'

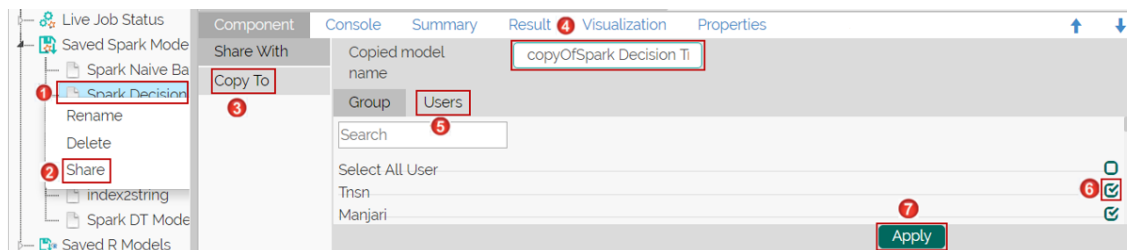


- vii) The saved Spark model will be shared with the selected group or users.

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 a 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.

19. Saved R Models

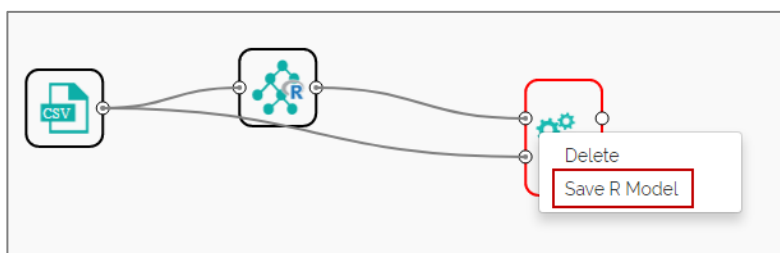
R Apply Model is a component used to generate predictions based on trained classification or regression model. The user can either split the dataset into training and testing, create a model with training data and apply the testing data. Another approach is to save the model and apply model over new test data set.

Users can save an R model after a 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.

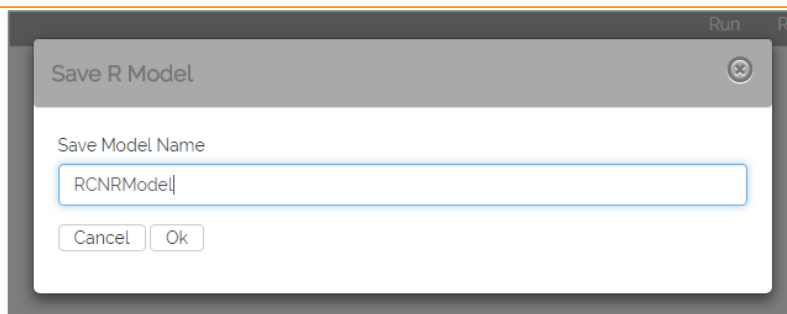
R Apply Model will come as a leaf node under Apply model tree node. The R Apply Model Component consists of two nodes for reading data from data source and another one for giving the result.

19.1. Saving an R Model

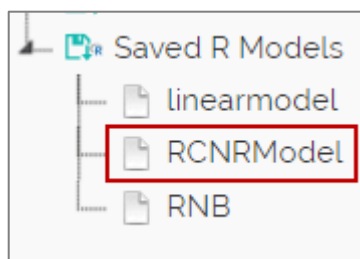
- i) Open an R workflow.
- ii) Connect **'Apply Model'** component with the workflow (as shown below).
- iii) Right-click on the **'Apply Model'** component.
- iv) A context menu will open.
- v) Select **'Save Model'**



- vi) A new window will pop-up.
- vii) Enter a name for the model that you wish to save.
- viii) Click **'Ok'**



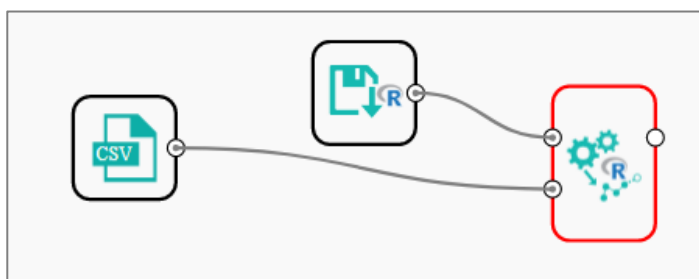
- ix) The created Predictive Model will be saved to the 'Saved R Models' list.



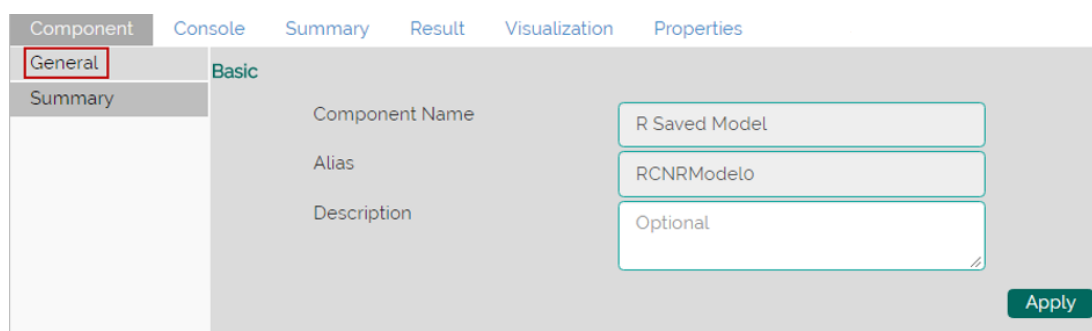
19.2. Reading an R Model

Users can drag a saved model to the workspace and reuse the model for a 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.

Component Console Summary Result Visualization Properties

General Summary

----- Summary of All Stages -----
 ----- Summary of stage 1 -----

CART

150 samples
 2 predictors
 3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
 Resampling: Cross-Validated (3 fold)
 Summary of sample sizes: 100, 101, 99
 Resampling results:

Accuracy Kappa
 0.939984 0.9100243

Tuning parameter 'cp' was held constant at a value of 0.005

Apply

v) Click **'Apply'** using the Apply Model component.

Component Console Summary Result Visualization Properties

General Basic

Component Name R Apply Model

Alias R Apply Model2

Description Optional

Apply Successful

vi) Click **'Run'**

vii) Users will be redirected to the **'Console'** tab.

```

Component Console Summary Result Visualization Properties
18/8/2017 - 12:22:44 : Process Initiated...
18/8/2017 - 12:22:44 : RCNRModelo started.
18/8/2017 - 12:22:44 : RCNRModelocompleted.
18/8/2017 - 12:22:44 : csv1 is started.
18/8/2017 - 12:22:44 : RCNRModelo started.
18/8/2017 - 12:22:44 : RCNRModelocompleted.
18/8/2017 - 12:22:44 : csv1 is started.
18/8/2017 - 12:22:44 : csv1 is completed.
18/8/2017 - 12:22:44 : csv1 is completed.
18/8/2017 - 12:22:44 : R Apply Model2 is started.
18/8/2017 - 12:22:44 : R Apply Model2 is started.
18/8/2017 - 12:22:46 : R Apply Model2 is completed.

```

viii) After the process gets completed under the Console tab, click the ‘Result’ tab to see result view of data.

Component Console Summary **Result** Visualization Properties

Show 10 entries Search:

Number	SepalLength	SepalWidth	PetalLength	PetalWidth	Species	PredictedValues1	Probability1
1	5.1	3.5	14	0.2	setosa	setosa	1.0000000
2	4.9	3	14	0.2	setosa	setosa	1.0000000
3	4.7	3.2	13	0.2	setosa	setosa	1.0000000
4	4.6	3.1	15	0.2	setosa	setosa	1.0000000
5	5	3.6	14	0.2	setosa	setosa	1.0000000
6	5.4	3.9	17	0.4	setosa	setosa	1.0000000
7	4.6	3.4	14	0.3	setosa	setosa	1.0000000
8	5	3.4	15	0.2	setosa	setosa	1.0000000
9	4.4	2.9	14	0.2	setosa	setosa	1.0000000
10	4.9	3.1	15	0.1	setosa	setosa	1.0000000

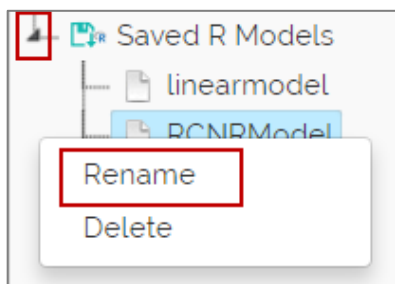
Showing 1 to 10 of 150 entries Previous 1 2 3 4 5 ... 15 Next

Note:

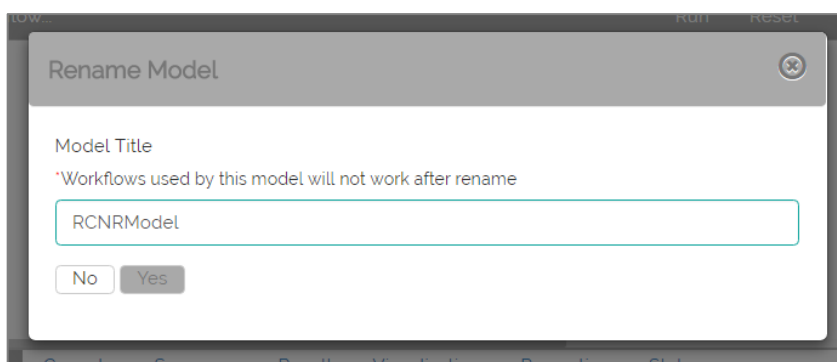
- A mandatory condition to run the workflow with a ‘Saved R Model’ component is that column headers and data type of the test data source should match with the selected saved model. Users will encounter an error if validation fails while running the workflow.
- Users can connect a data writer to the ‘Apply Model’ component in a workflow containing a saved model.

19.2.1. Renaming an R Model

- Select a model from the ‘Saved R Models’ list.
- Right-click on the selected model.
- A context menu will open.
- Select ‘Rename’.



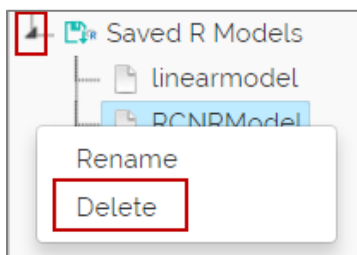
- v) A pop-up window will appear to rename the model.
- vi) Enter a new **'Model Title'** or modify the existing model title in the given field (if desired).
- vii) Click **'Yes'**



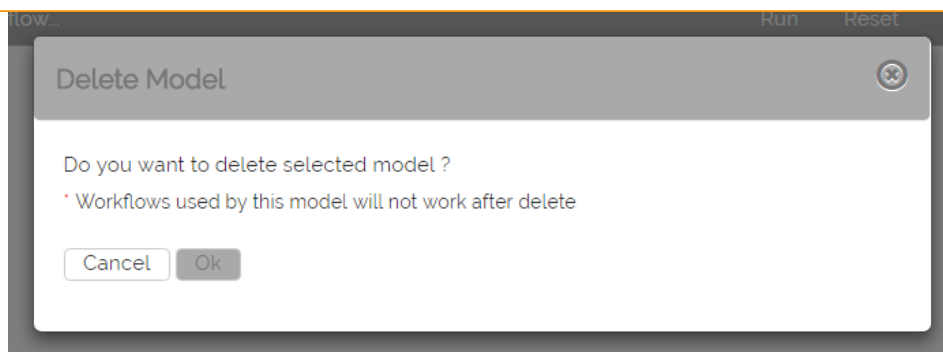
- viii) The selected R Predictive Model will be renamed.

19.2.2. 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'**



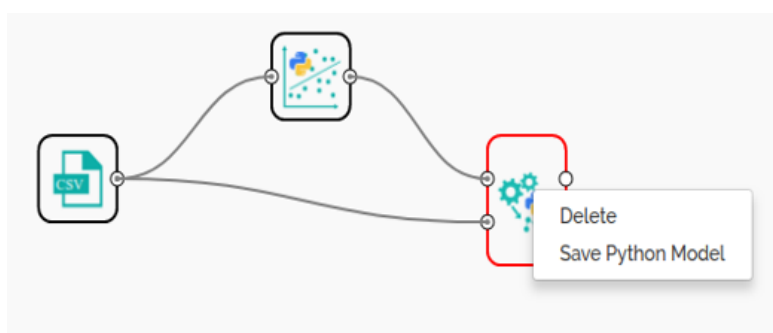
vii) The selected predictive model will be deleted and removed from the list of **'Saved R Models.'**

Note: After renaming or deleting a Saved R Model, workflows used by the same model will not work.

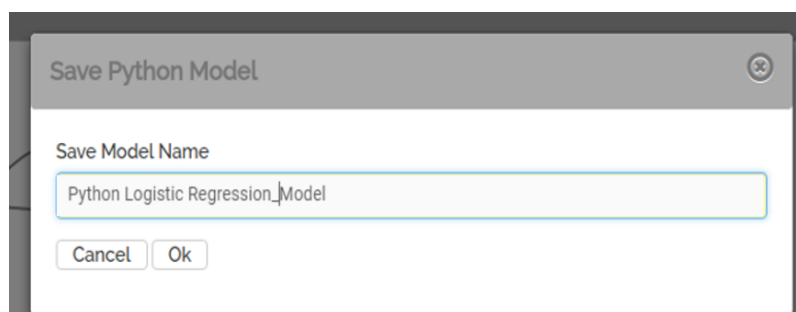
20. Saved Python Models

20.1. Saving a Python 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'**



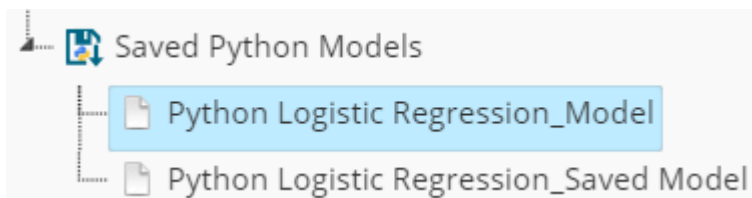
- vi) A new window will pop-up.
- vii) Enter a name for the model that you wish to save.
- viii) Click **'OK'**



- ix) A success message will pop-up at the top.

Python model saved successfully!

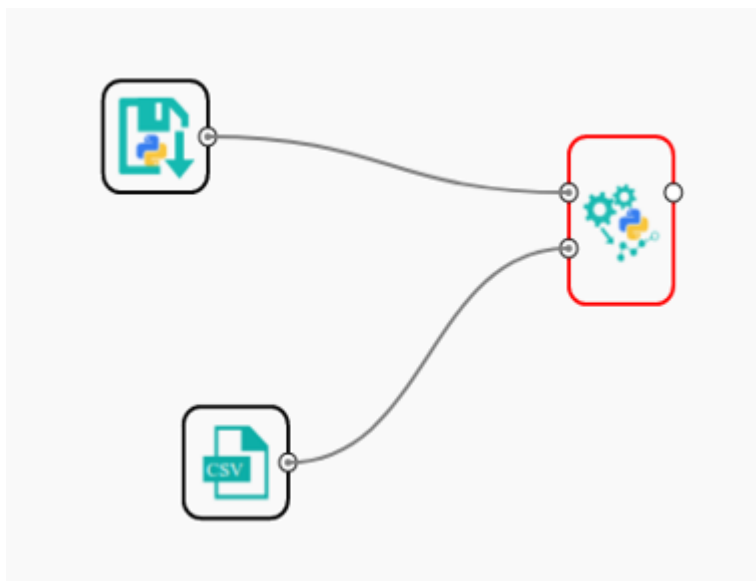
- x) The newly created Predictive Model will be saved to the 'Saved Python Models' list.



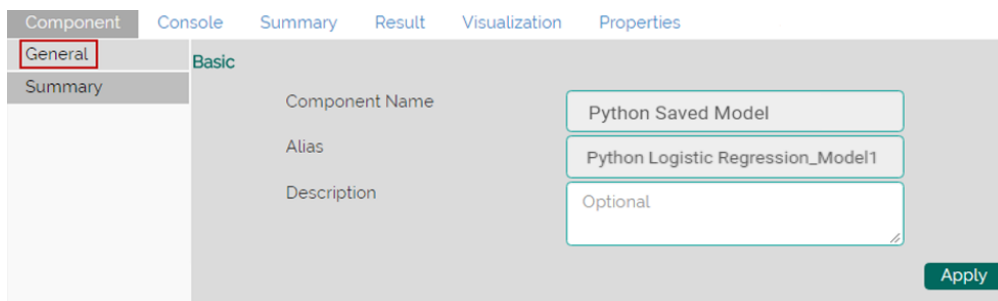
20.2. Reading a Python Model

Users can drag a saved model to the workspace and reuse the model for a test data. A saved R model can be connected to only Apply Model and new test data source.

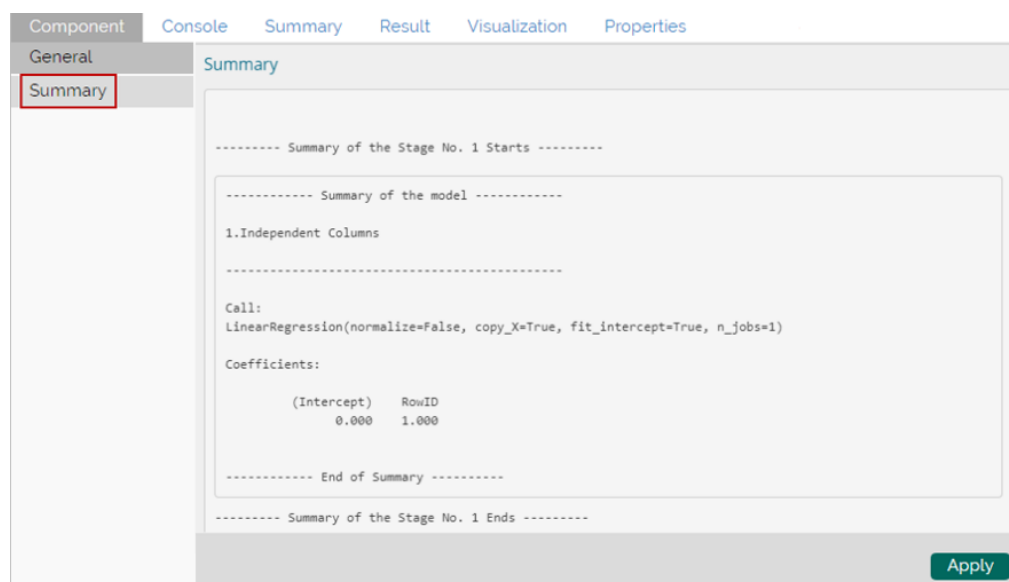
- i) Select and drag a saved Python saved model component onto the workspace.
- ii) Connect the dragged model with a configured data source and an Apply Model component (As shown in the following image).



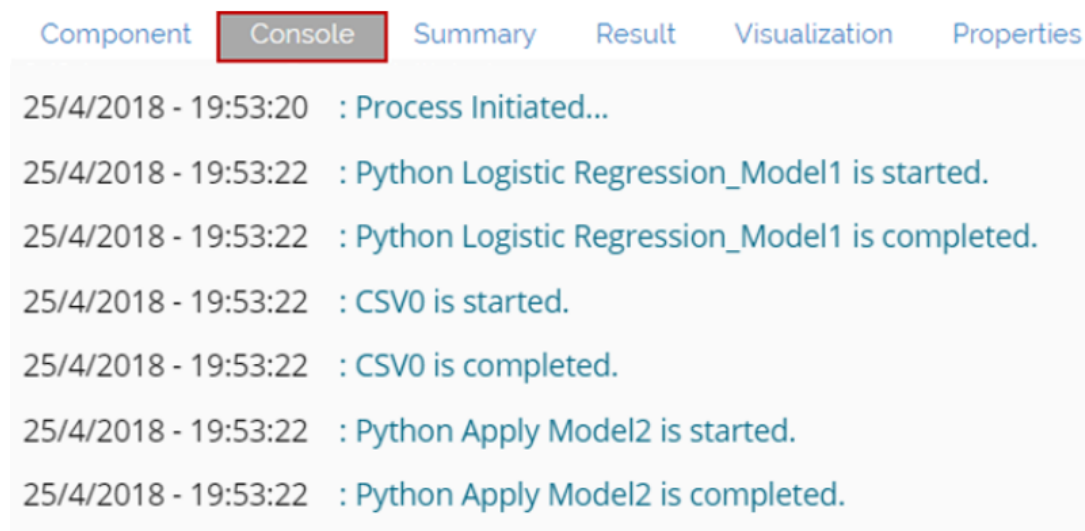
- iii) Click on the dragged Saved Model component.
- iv) Users will be able to view the following 'Component' tabs:
 - a. General



- b. Click **'Summary'** tab to display the model summary.
- c. Click **'Apply'**.



- v) Click **'Run'**
- vi) Users will be redirected to the **'Console'** tab.



- vii) After the process gets completed under the Console tab, click the **'Result'** tab to see result view of data.

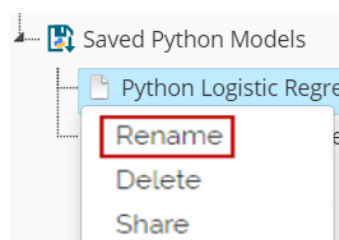
RowID	SLength	SWidth	PLength	PWidth	PredictedValues1
1	5.1	3.5	1.4	0.2	1.0000000000000028
2	4.9	3	1.4	0.2	2.00000000000000275
3	4.7	3.2	1.3	0.2	3.00000000000000275
4	4.6	3.1	1.5	0.2	4.0000000000000027
5	5	3.6	1.4	0.2	5.0000000000000027
6	5.4	3.9	1.7	0.4	6.0000000000000027
7	4.6	3.4	1.4	0.3	7.0000000000000026
8	5	3.4	1.5	0.2	8.0000000000000025
9	4.4	2.9	1.4	0.2	9.0000000000000025
10	4.9	3.1	1.5	0.1	10.0000000000000025

Note:

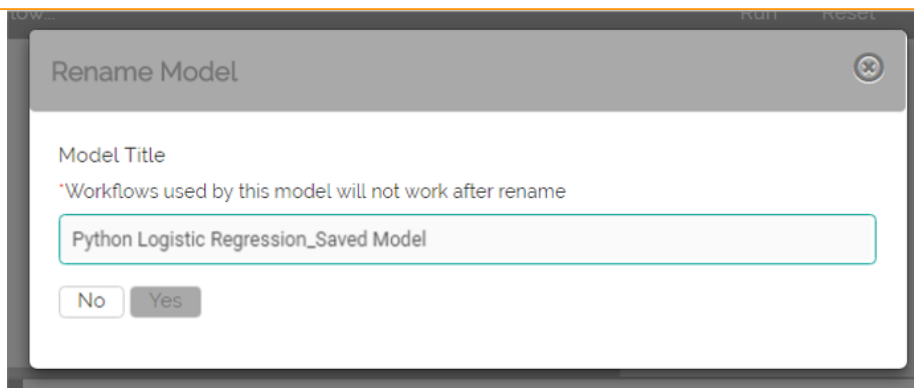
- a. A mandatory condition to run the workflow with a ‘**Saved Python Model**’ component is that column headers and data type of the test data source should match with the selected saved model. Users will encounter an error if validation fails while running the workflow.
- b. Users can connect a data writer to the ‘**Apply Model**’ component in a workflow containing a saved model.

20.2.1. 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**’.



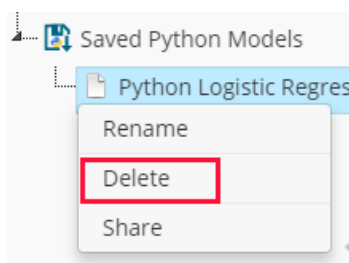
- v) A pop-up window will appear to rename the model.
- vi) Enter a new ‘**Model Title**’ or modify the existing model title in the given field (if desired).
- vii) Click ‘**Yes**’



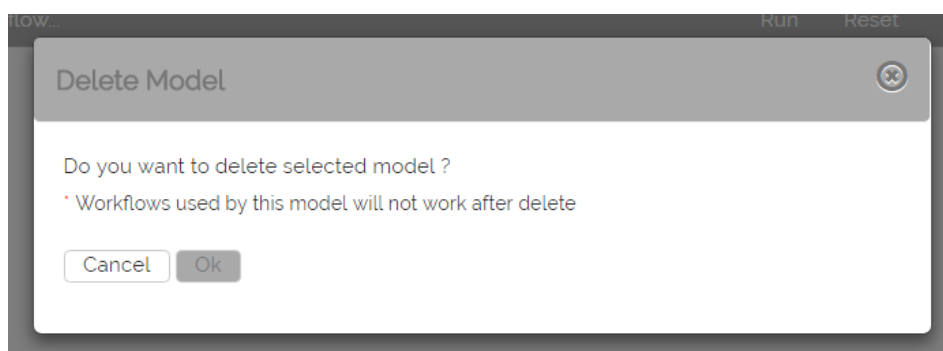
viii) The selected Python saved model will be renamed.

20.2.2. 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 list of 'Saved Python Models.'

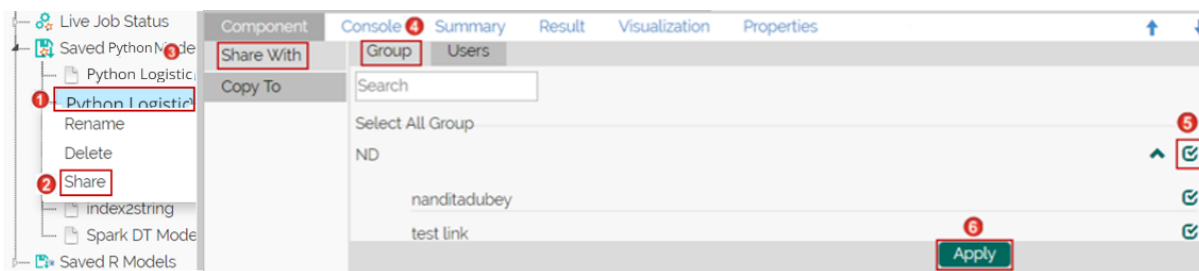
Note: After renaming or deleting a Saved R Model, workflows used by the same model will not work.

20.2.3. 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.

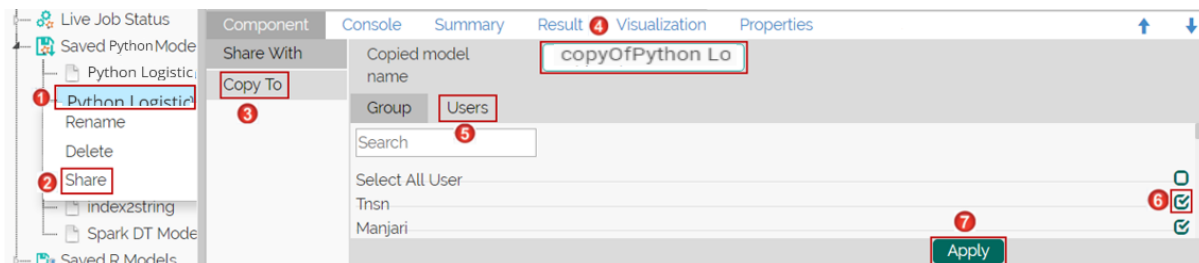
- i) Use right-click on a model from the list of ‘**Saved Models.**’
- ii) Select ‘**Share Model**’ from the context menu.
- iii) The ‘**Share With**’ option will be displayed (by default).
- iv) Select either ‘**Group**’ or ‘**Users**’ option.
 - c. By selecting a group all group members inside the group will be listed. Users can be excluded by not selecting them from the group.
 - d. 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 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 a 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

21. Signing Out

Follow the below given steps to log out from the BizViz Platform.

- i) Click 'User' icon  on the Platform home page.
- ii) A menu appears with the logged in user details.
- iii) Click 'Sign Out' option from the menu.
- iv) Users will be successfully logged out from the **BizViz Platform**.

Note: Clicking on 'Sign Out' will redirect the user back to the 'Login' page of the BizViz platform.