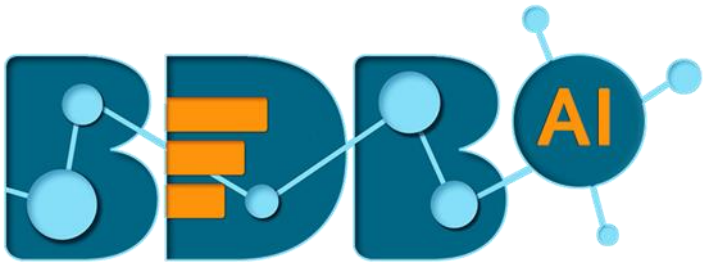


User Guide

Data Science Workbench R-6.5

Contents

1. About This Guide

1.1. Document History

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

Product Version	Date (Release Date)	Description
Predictive Workbench 1.0	June 9th, 2015	First Release of the document
Predictive Workbench 2.0	Feb 18th, 2016	Updated document
Predictive Workbench 2.0	May 31st, 2016	Modified document
Predictive Workbench 2.5	November 9th, 2016	Updated document
Predictive Workbench 2.5.1	January 3rd, 2017	Updated document
Predictive Workbench 2.5.3	March 16th, 2017	Updated document
Predictive Workbench 3.0	August 31st, 2017	Updated document
Predictive Workbench 3.0	November 22nd, 2017	Modified document
Predictive Workbench 3.2	January 25th, 2018	Updated document
Predictive Workbench 3.5	April 15th, 2018	Updated document
Predictive Workbench 3.6	August 20th, 2018	Updated document
Predictive Workbench 3.7	October 10th, 2018	Updated document
Predictive Workbench 3.8	December 1st, 2018	Updated document
Predictive Workbench 4.0	December 31st, 2018	Updated document
Predictive Workbench 4.2	March 25th, 2019	Updated document
Predictive Workbench 4.3	April 24th, 2019	Updated document
Predictive Workbench 4.4	June 7th, 2019	Updated document
Data Science Workbench 4.5	August 5th, 2019	Updated document
Data Science Workbench 4.6	November 15th, 2019	Updated document
Data Science Workbench 5.0	February 17th, 2020	Updated document
Data Science Workbench 6.5	September 30th, 2021	Updated document

Note:

a. The Predictive Workbench plugin is renamed as Data Science Workbench from R-4.5 onwards.
b. The Spark ML and PySpaces are experimental workspaces so the detailed description of those workspaces is not included in the current document.

1.2. Overview

This guide covers steps to:

- Access the BDB Data Science Workbench
- Server requirements and configuration details for the BDB Data Science Workbench
- Designer Part of the BDB Data Science Workbench
- Result or Analysis Part (Visualizing the analyzed data) of the BDB Data Science Workbench
- Creation and use of various Data Science Models

1.3. Target Audience

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

2. Introducing BDB Data Science Workbench

2.1. Introduction

BDB Data Science Workbench provides the required environment for its users to create AI and ML models to empower their business insights. These Models can be used to envision the future outcomes of business processes based on past data. It is a user-friendly tool that shields users from mathematical complexity and offers an interactive graphical interface to provide a smooth, intuitive experience. It enables the users to discover hidden patterns in their data by Applying various statistical algorithms provided by the popular R statistical language, Spark ML, Python, and Deep Learning using Neural Network.

2.2. Supported Web Browsers

The BDB Platform is a web browser-based application. The users can run the BDB Platform and its various plugins on the below-given versions of the browsers:

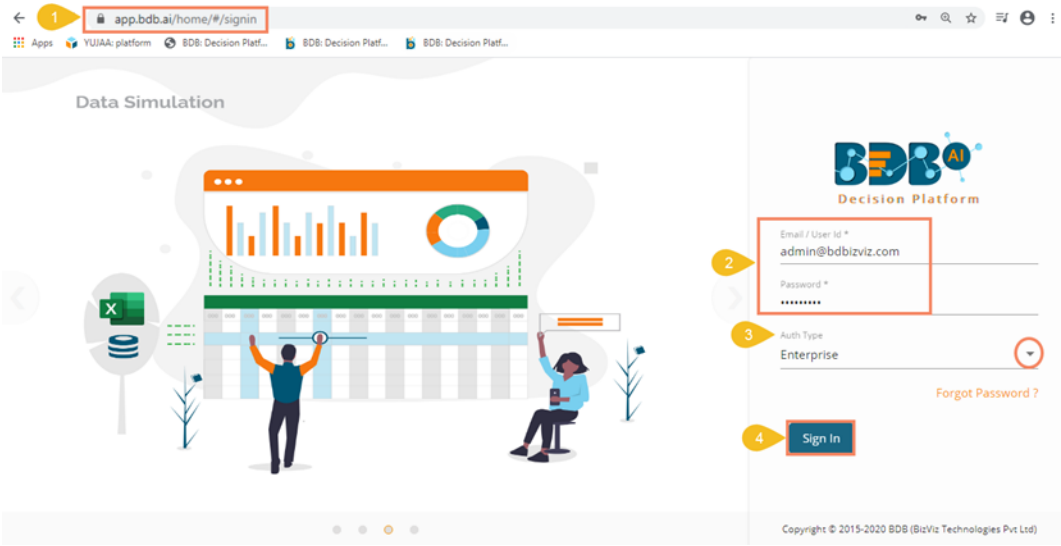
Mozilla Firefox/ Firefox ESR	**Latest Version**
Microsoft Edge	Latest Version
Apple Safari	10
Google Chrome	Latest Version (recommended web browser)

3. Getting Started with the Data Science Workbench

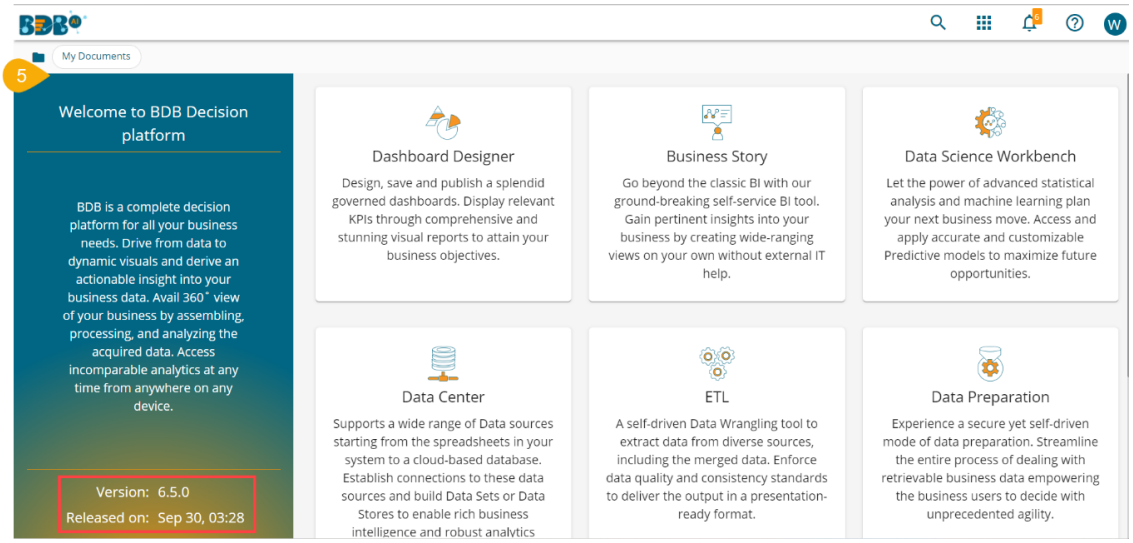
3.1. Accessing Data Science Workbench

This section explains how to access the BDB Platform and the variety of plugins that it offers:

i) Open BDB Enterprise Platform Link: https://app.bdb.ai
ii) Enter your credentials.
iii) Select an Auth Type from the drop-down menu.
iv) Click the '**Sign In**' option.



v) BDB Platform homepage opens.

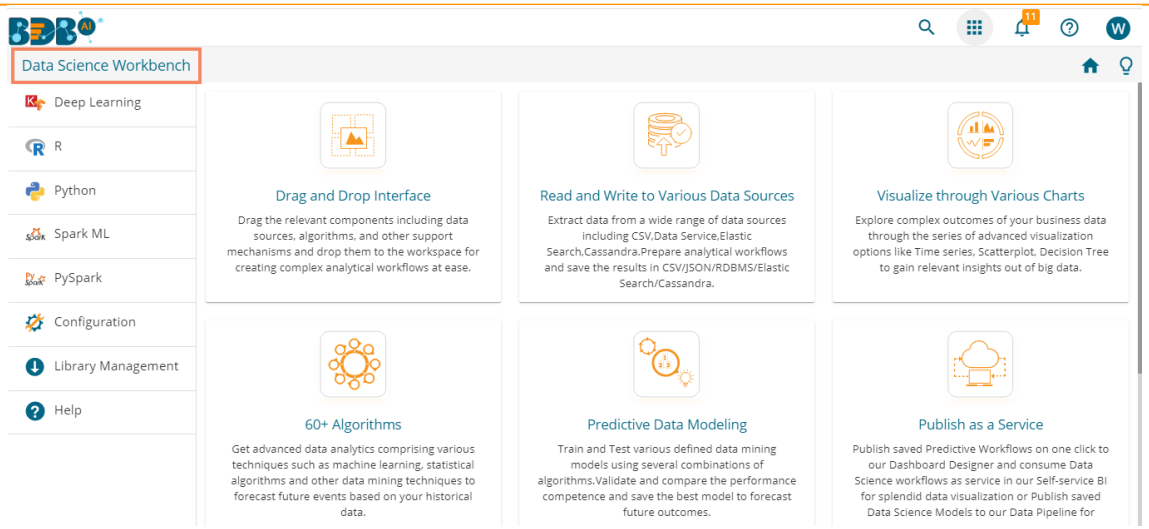


Note:

a. The above screen opens only for those newly created users who have not yet created any document using the BDB Platform.
b. If the user has created some documents previously, then the Platform homepage opens, displaying the '**My Documents**' page by default.

vi) Click the '**Apps**' icon.
vii) All the available plugin applications get displayed.
viii) Select the '**Data Science Workbench**' plugin.

ix) The Data Science Workbench homepage opens.
x) The major Data Science Workspaces get listed on this page.



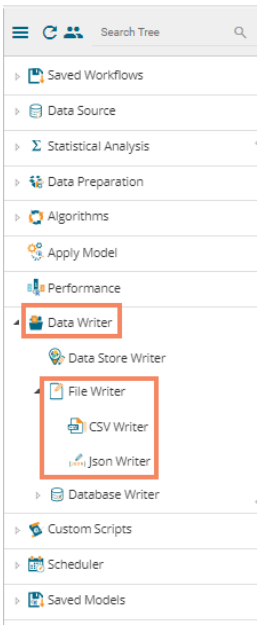
This document aims to describe all the significant components and the related workflows in detail.

4. Overview of the Data Science Workspace(s)

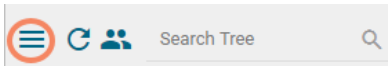
This section describes all the options and icons provided on the landing page of the different Data Science Workspaces. The landing page of any selected Data Science Workflow contains the following Menus:

4.1. Tree-node Menu

The Tree-node menu has all the available component connectors to run a data science execution. The components are provided in the hierarchical order via a tree structure menu. All the main categories are included as tree nodes, and sub-categories are committed as petals to the respective tree nodes.
E.g. The following image displays the R Workspace landing page where '**Data Writer**' is the main category to which '**File Writer**' is committed as a subcategory and '**CSV Writer,**' and '**JSON Writer**' are displayed at the second level of the hierarchy.

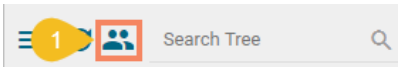


c. Click the '**Menu**' option Next to the '**Search**' box to collapse the tree structure menu from the homepage.

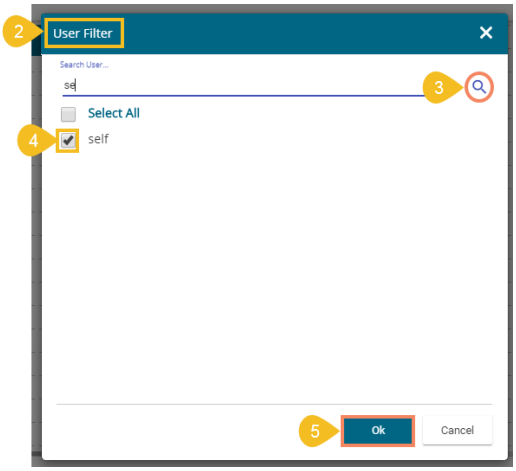


d. The User Filter functionality is provided to restrict the display of the Workspace list to the other user of the same space.

1) Click the '**User Filter**' icon.

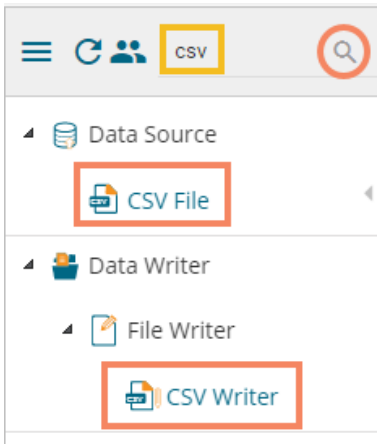


2) The User Filter window appears.
3) Search for the specific user.
4) Select the user(s) by a checkmark in the given box.
5) Click the '**Ok**' option.



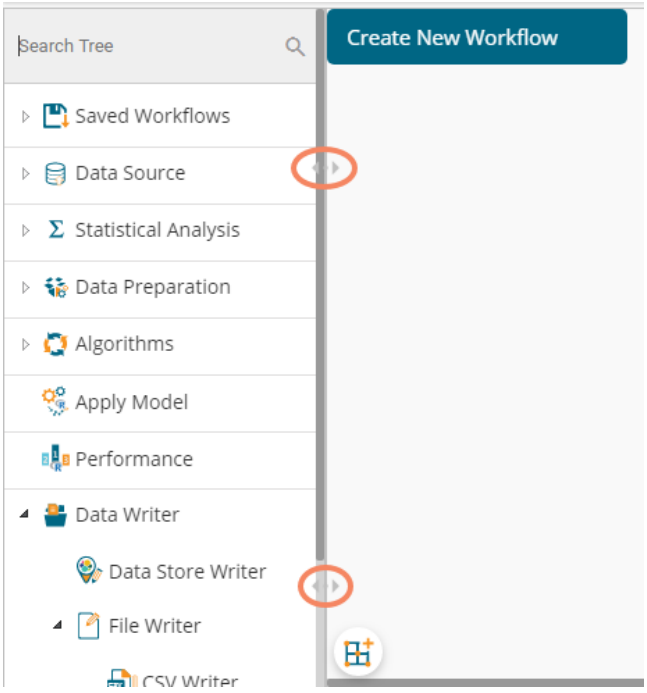
6) The Workflow saved by the user gets displayed only to the selected user(s).

e. Click the '**Search**' icon to search across the entire tree-node menu.



f. Click on the  icon to show or hide the gridlines on the workspace.

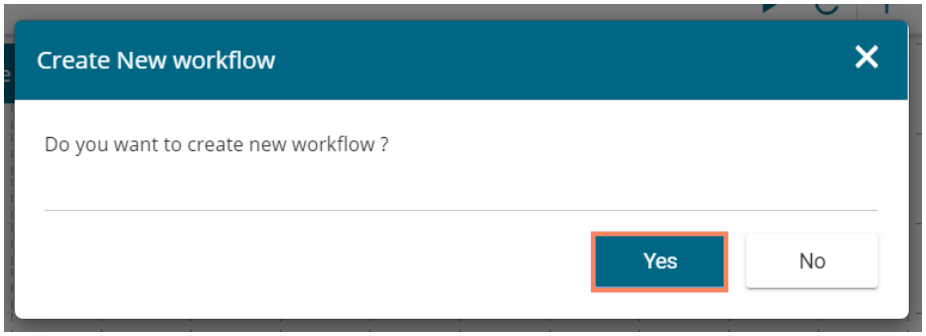
g. The user can use these scrolling icons to increase or decrease horizontal space for the Tree Menu.



Note: 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.

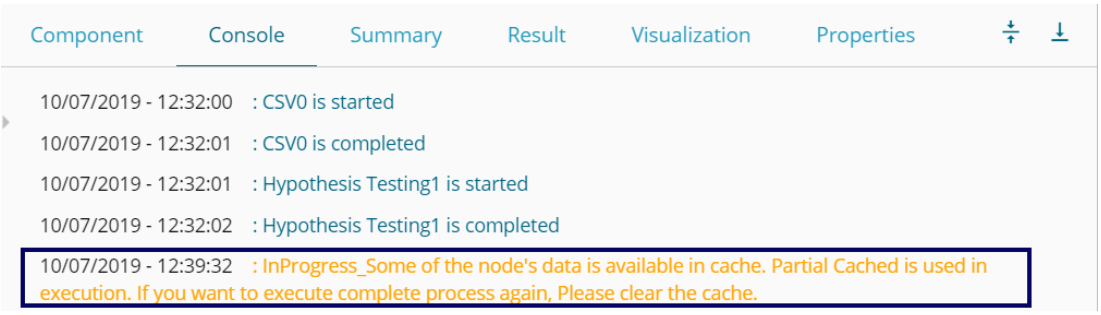
4.2. Header Menu-Options

1. **Run:** Click the '**Run**' icon to run the process and display the Result set view. This option can be applied to the data source, algorithms, and data preparation components.
2. **Refresh**: The '**Refresh**' icon is provided the clear the cache memory and runs the component/ workflow.
3. **Create New Workflow:** Click the '**Create New Workflow**' icon to clean the workspace removing the current component connectors.
   The **'Create New Workflow'** dialog box opens. Click the '**Yes**' option to clean the workspace.



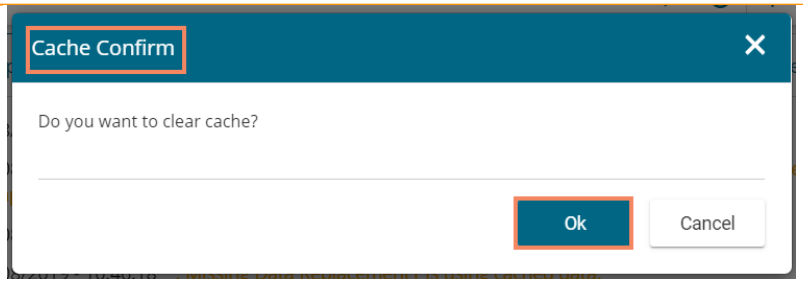
4. **Clear Cache:**
   a. After using the '**Run**' option by default, the data gets cached in the server for the Next 10 minutes. For the latest Results, users need to rerun the workflow.
   b. The user needs to click the '**Clear Cache**' option to remove the cached data before running the workflow (again).
   c. If the user changes any component parameter which is to be applied to fetch the Result then, the '**Clear Cache**' icon needs to be clicked.

   If you get a message to clear cache to execute your process as shown in the following image:

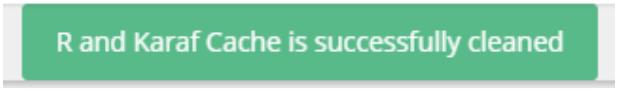


   Please follow the below-given steps to Clear Cache:

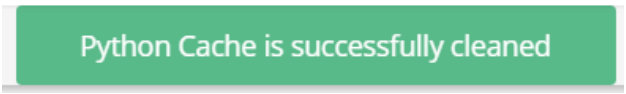
   i) Click the '**Clear Cache**' icon from the header menu.
   ii) A message appears to confirm.
   iii) Click the '**Ok**' option.



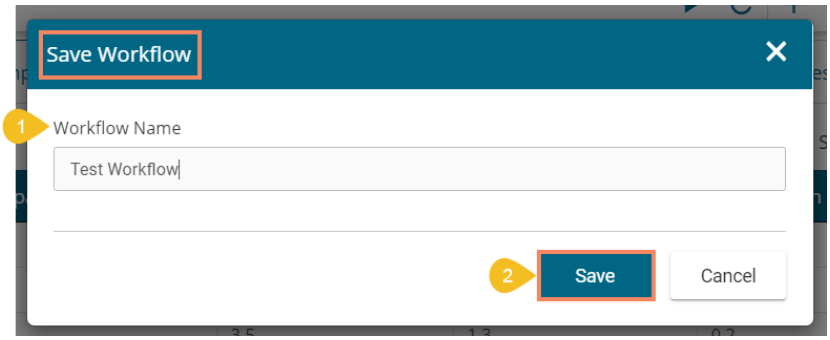
iv) A message appears to confirm that the workspace-specific cache is cleaned. The below message appears for the R Workspace:



The following message appears for the Deep Learning and Python Workspaces.

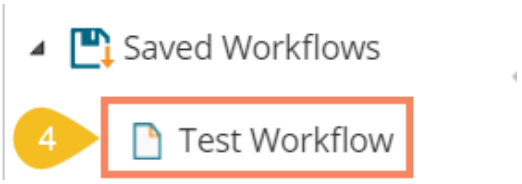


5. **Save:** Use the '**Save**' icon to save a created predictive workflow.
   i) Create a workflow by connecting various configured components.
   ii) Click the '**Save**' icon from the landing page header menu.
   iii) A new window appears to confirm the action.
      a. Provide a Workflow Name.
 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



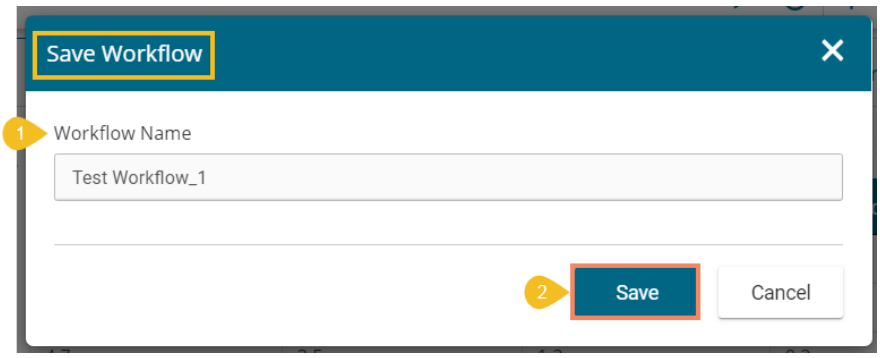
iv) A success message appears.

v) The selected workflow gets saved to the list of **Saved Workflows**.

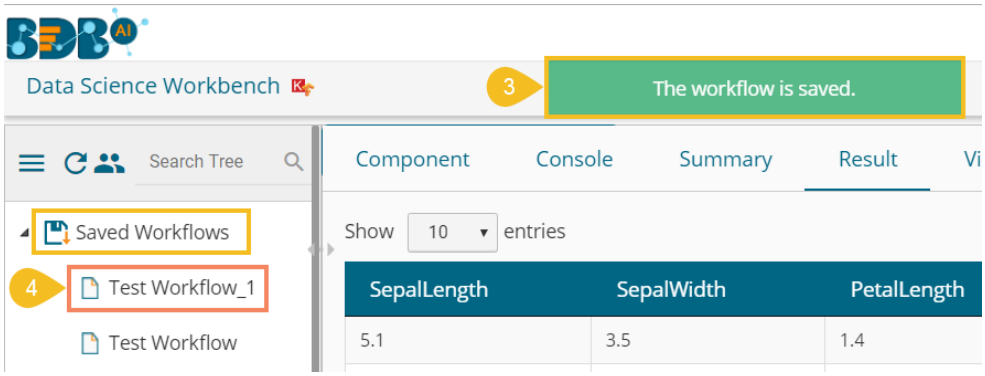


6. **Save As:** Click the '**Save As**' icon to copy a data science workflow with the desired name.

   i) Create a workflow by connecting various configured components.
   ii) Click the '**Save As**' icon.
   iii) A new window appears to confirm the task.
      a. The Workflow Name contains the suffix '_1' by default (If wished, users can also modify the name of workflow manually).
      b. Click the '**Save**' option.



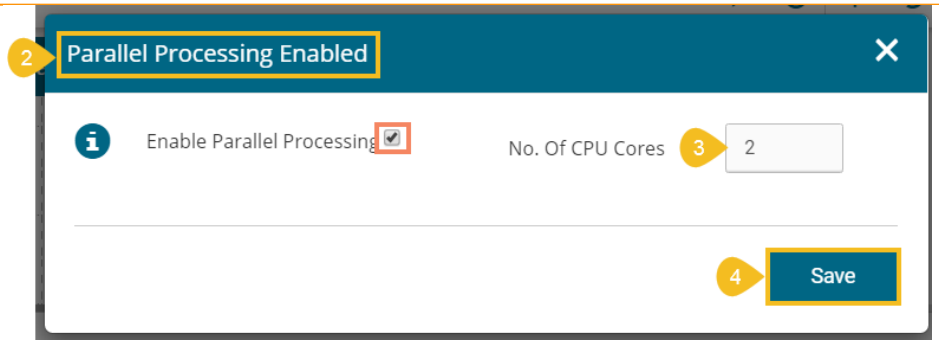
   iv) A success message appears.
   v) The selected workflow gets saved by the new name in the '**Saved Workflows**' list.



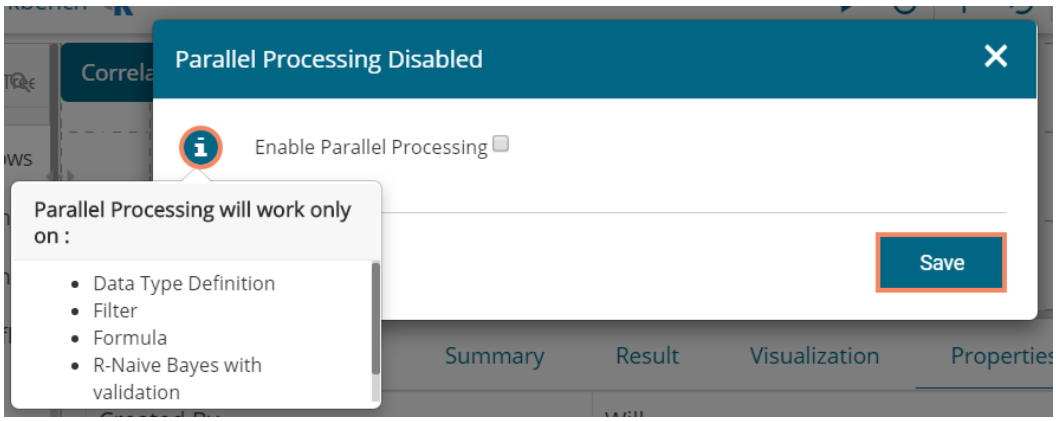
7. **Parallel Processing:** The user can enable or disable parallel processing by clicking the '**Parallel Processing**' icon on the R landing page header. **This option is only available for the R Workspace**.

   a. Click the '**Parallel Processing**' icon.

   b. The '**Parallel Processing Enabled**' dialog box opens with a checkmark in the given box.
   c. Provide No. of CPU Cores in the given space.
   d. Click the '**Save**' option.

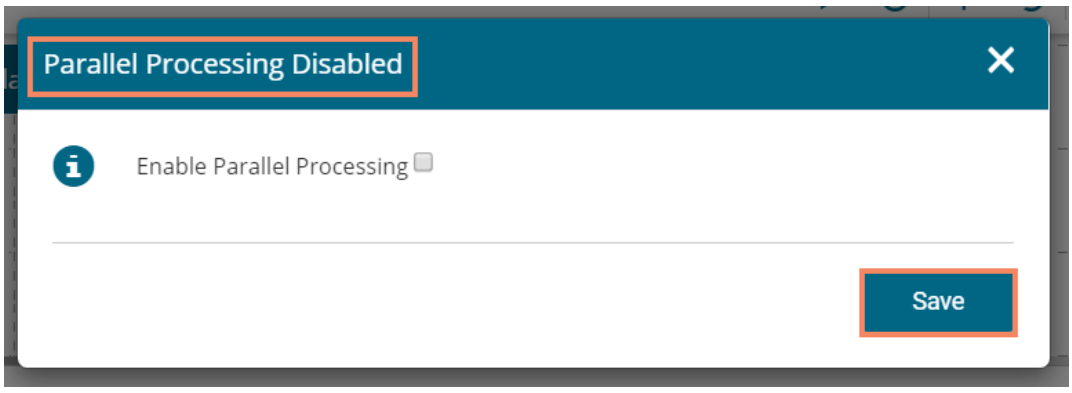


e. The parallel processing gets enabled for the R Workspace.
f. Click the '**Information**' icon to get information about Parallel Processing.



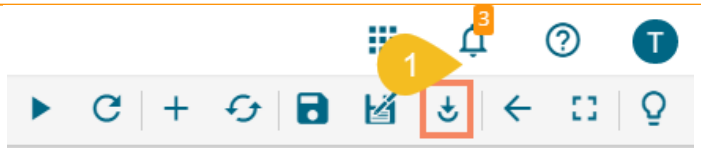
The Parallel Processing works only on Data Type Definition, Filter, Formula data Preparation components and R-Naïve Bayes (with Validation) algorithm.

Note: The user gets the Parallel Processing Disabled screen as given below:

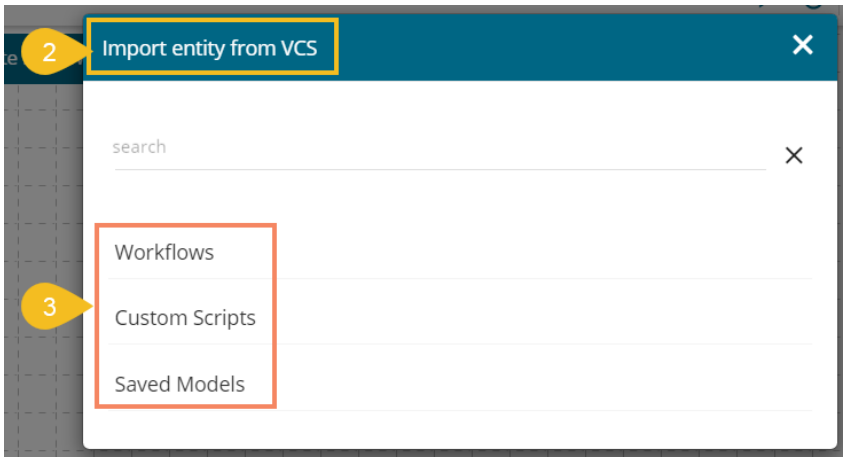


8. **Version Control Panel:** The user gets a dialog box to import Workflows, Custom Scripts, and Saved Models from Version Control Service (VCS) by clicking on the '**Version Control Panel**' icon. This icon is available only for the Python Workspace.

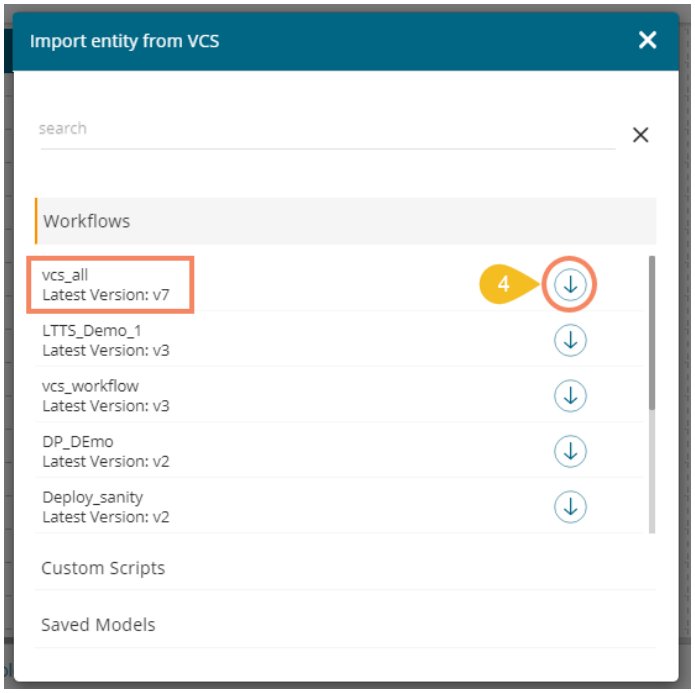
a. Click the '**Version Control Panel**' icon.



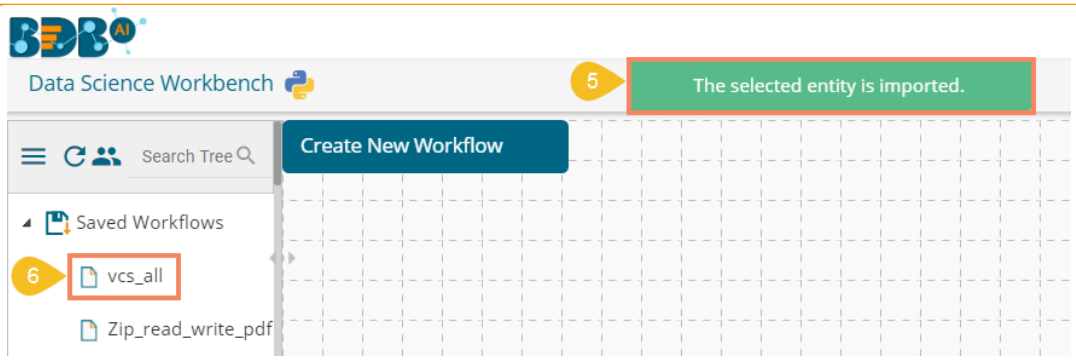
b. A dialog box opens displaying Workflows, Custom Scripts, and Saved Models categories to be imported.



c. Select a Workflow/ Custom Script/ Saved Model and click the import icon.



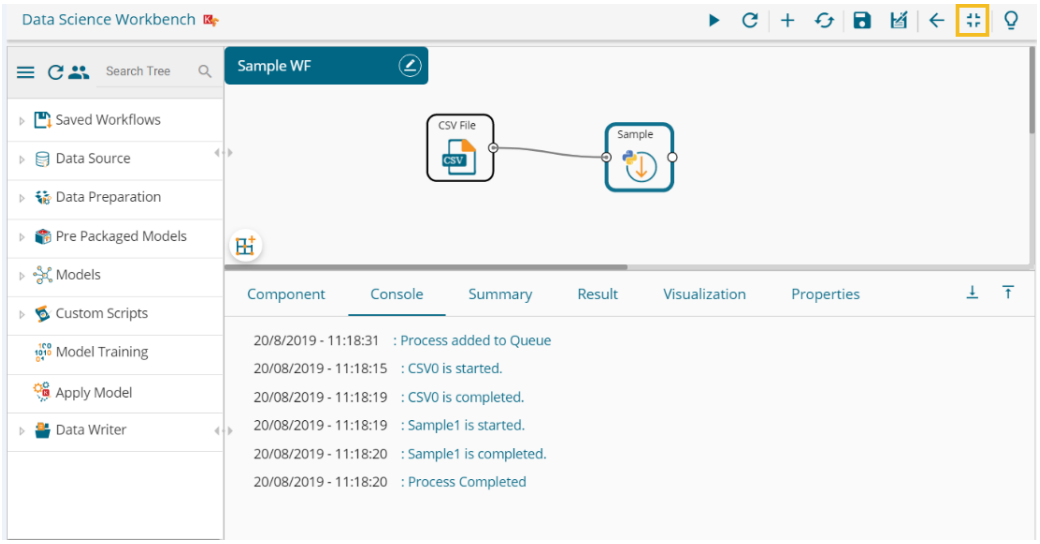
d. A success message appears.
e. The selected entity (workflow/custom script/saved model) gets imported under the specific section.



Note: If the user tries to import an existing workflow, script, or model then a warning message appears, and the selected entity does not get imported.

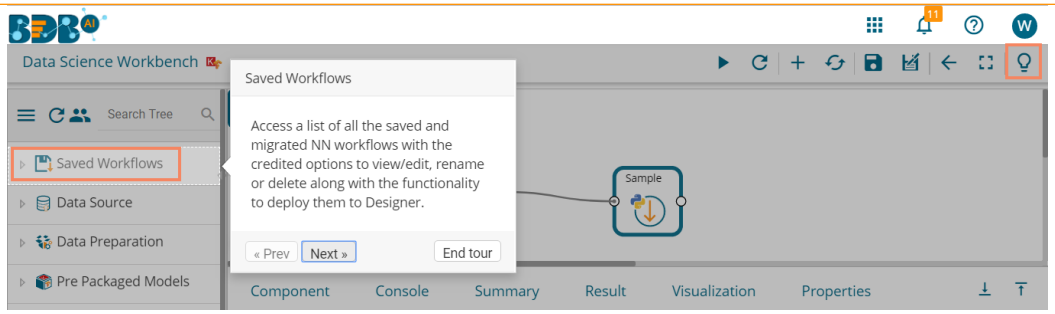
9. **Back:** Click the **'Back'** icon to return on the Data Science homepage from any specific workspace.

10. **Full Screen/ Full-Screen Exit:** Click the '**Full Screen**' icon to display the selected Workspace on the full screen. The '**Full-Screen Exit**' icon appears to exit the full-screen view.



Note: The user can also use the '**Esc**' key to close the full-screen view.

11. Start Tour: Click the '**Start Tour**' icon to begin the auto-guided tour for the selected workspace.



Note:

a. Click the '**Next**' option to proceed in the guided tour of the selected workspace.
b. Click the '**Prev**' option to go back to the guided tour of the selected workspace.
c. Click the '**End tour**' option to end up the guided tour.

4.3. Tabbed Menu Strip - Options

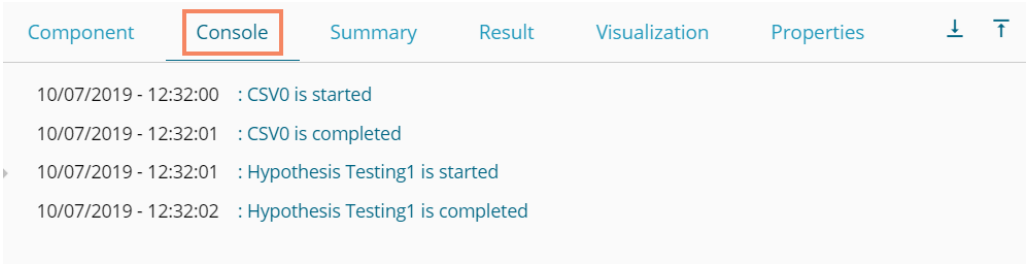
1. **Component:** The **'Component'** tab displays the required configuration fields for the dragged elements onto the workspace.

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

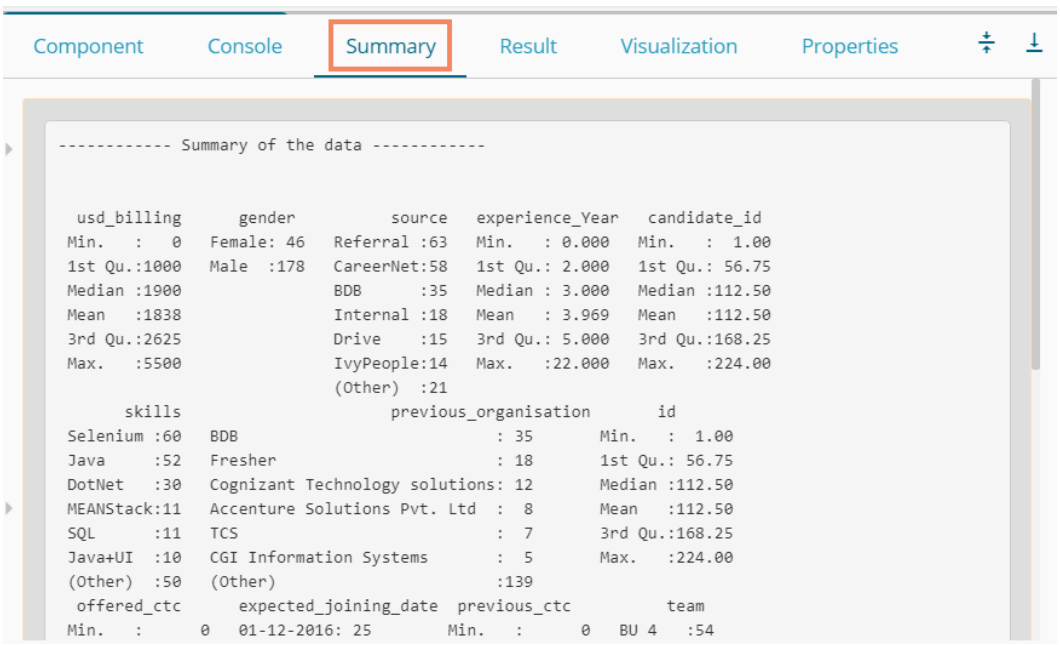
E.g., If the dragged data source is a CSV file, then the component tab displays General and Properties fields, while for a Cassandra Reader as a data source, the component tabs display General, Properties, and Column Selection.

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

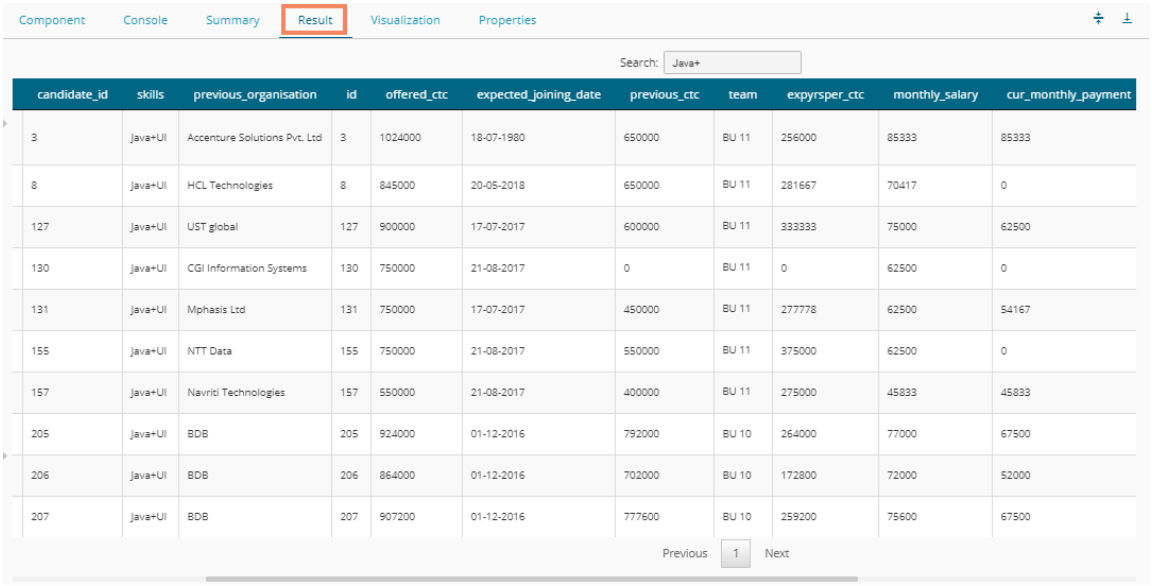
i) Click the **'Console'** option.
ii) The workflow process records (starting and ending time) get displayed:



3. **Summary:** Click the **'Summary'** tab to display the 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.

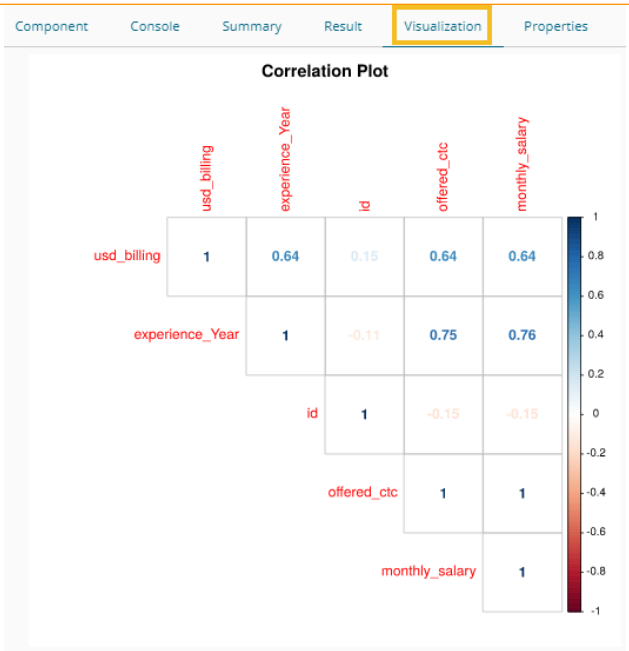


Note:

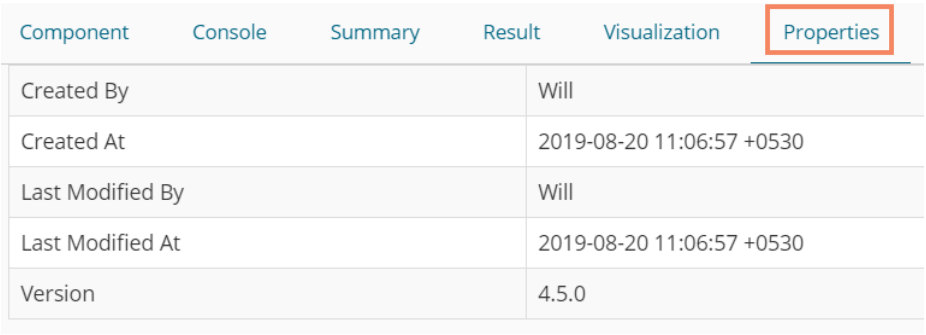
a. The '**Result**' tab gets displayed for the given data only after data is configured and the '**Run**' option has been selected. Up to 50000 cells can be displayed in the Result view.

b. The user can search for specific data using the 'Result' tab.

5. **Visualization:** Click the **'Visualization'** tab to display a graphical representation of the Result data. E.g., The following image displays a Correlation in the chosen data via the '**Correlation Plot**' chart.

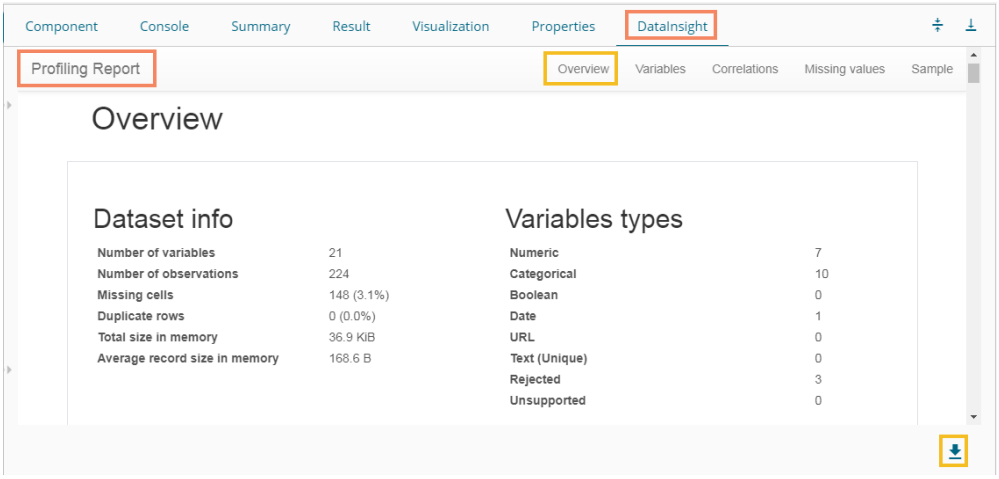


6. **Properties:** Click the '**Properties**' tab to display properties for the current workflow on the Workspace.

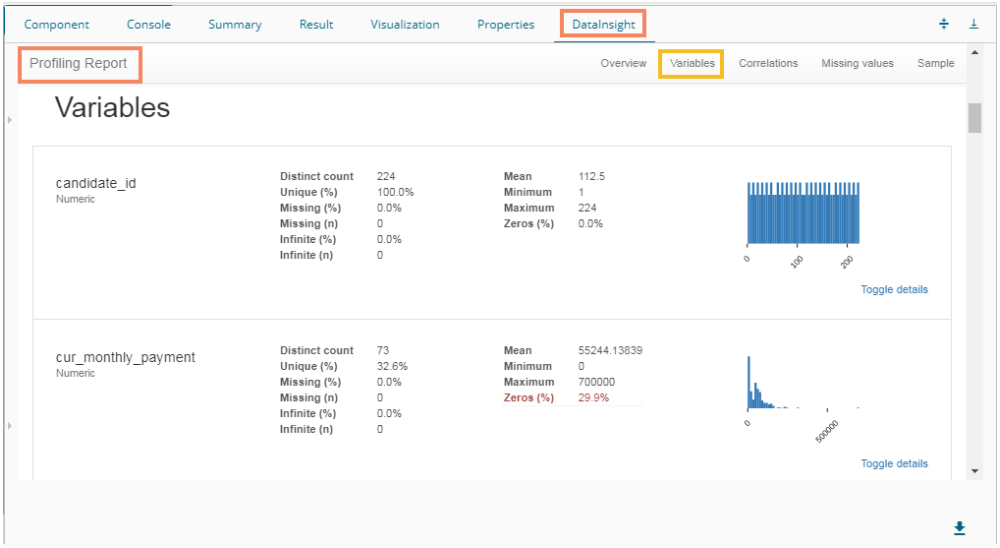


7. **Data Insight**: Click the '**DataInsight**' tab from the Python workspace to display a detailed profiling report for the uploaded/processed data. The report opens displaying Overview, Variables, Correlations, Missing Values, and Sample sections.

   Overview: It displays an overview of the uploaded dataset.

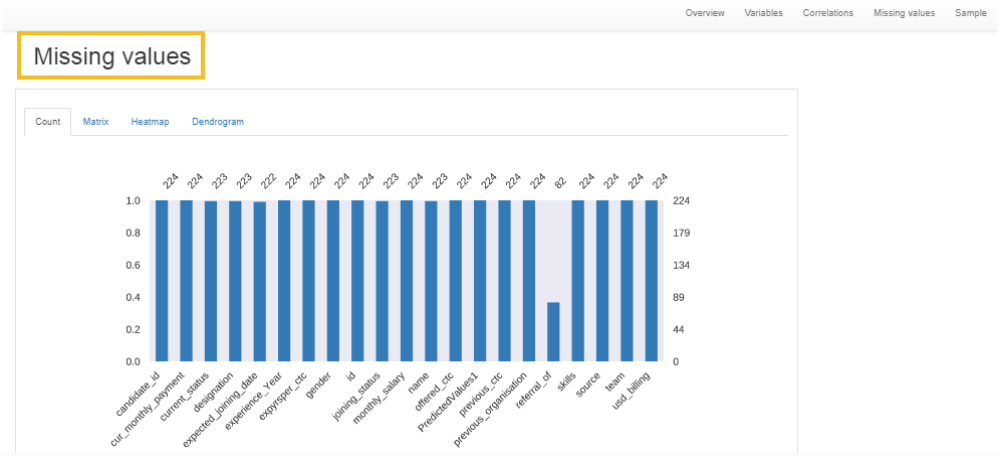


Variables: All the variables get displayed via a column chart.

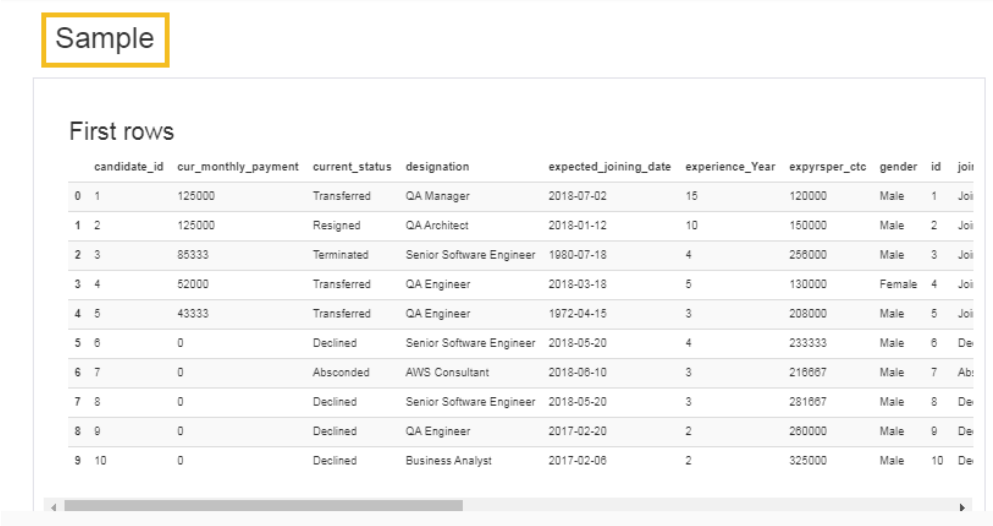


Correlations: It displays the correlational values via the Correlation Plot chart.

Missing Values: It displays the missing values through the column chart.



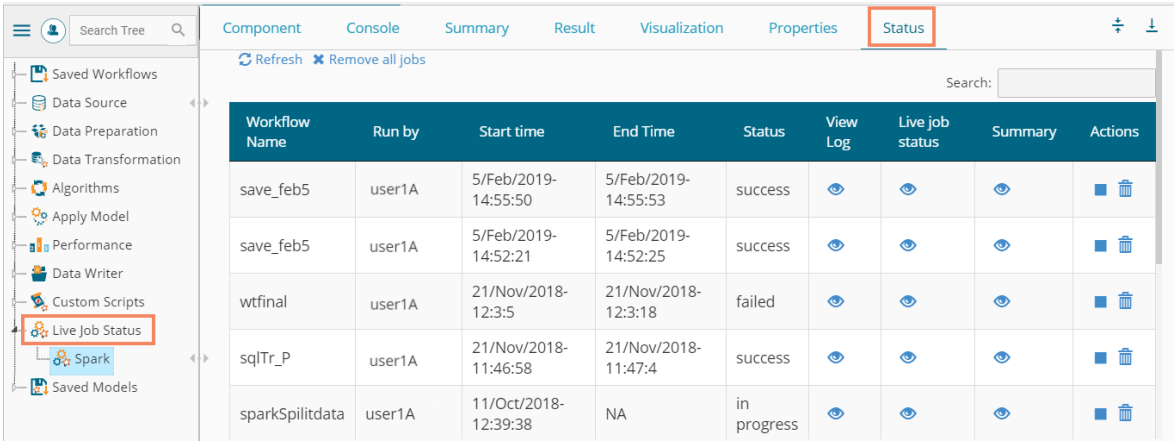
Sample: It displays data from the First 10 and Last 10 rows as a sample.



	candidate_id	cur_monthly_payment	current_status	designation	expected_joining_date	experience_Year	expyrsper_ctc	gender	id	joi
0	1	125000	Transferred	QA Manager	2018-07-02	15	120000	Male	1	Joi
1	2	125000	Resigned	QA Architect	2018-01-12	10	150000	Male	2	Joi
2	3	85333	Terminated	Senior Software Engineer	1980-07-18	4	256000	Male	3	Joi
3	4	52000	Transferred	QA Engineer	2018-03-18	5	130000	Female	4	Joi
4	5	43333	Transferred	QA Engineer	1972-04-15	3	208000	Male	5	Joi
5	6	0	Declined	Senior Software Engineer	2018-05-20	4	233333	Male	6	De
6	7	0	Absconded	AWS Consultant	2018-06-10	3	216667	Male	7	Ab
7	8	0	Declined	Senior Software Engineer	2018-05-20	3	281667	Male	8	De
8	9	0	Declined	QA Engineer	2017-02-20	2	260000	Male	9	De
9	10	0	Declined	Business Analyst	2017-02-06	2	325000	Male	10	De

Note: The '**Download Report**' icon gets provided to download the entire DataInsight report.

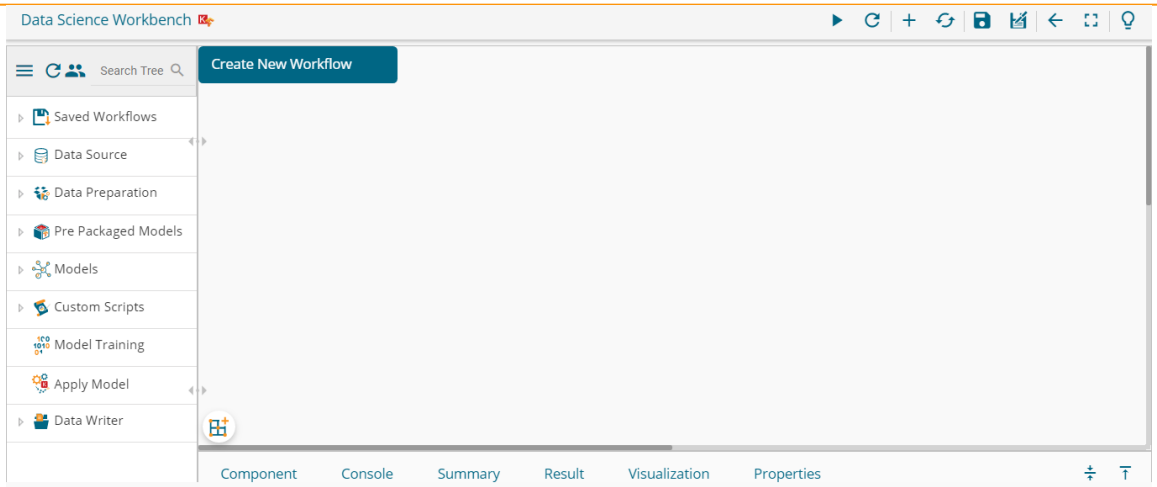
8. **Status:** Click the '**Status**' tab to view the live job status of a running Spark job.



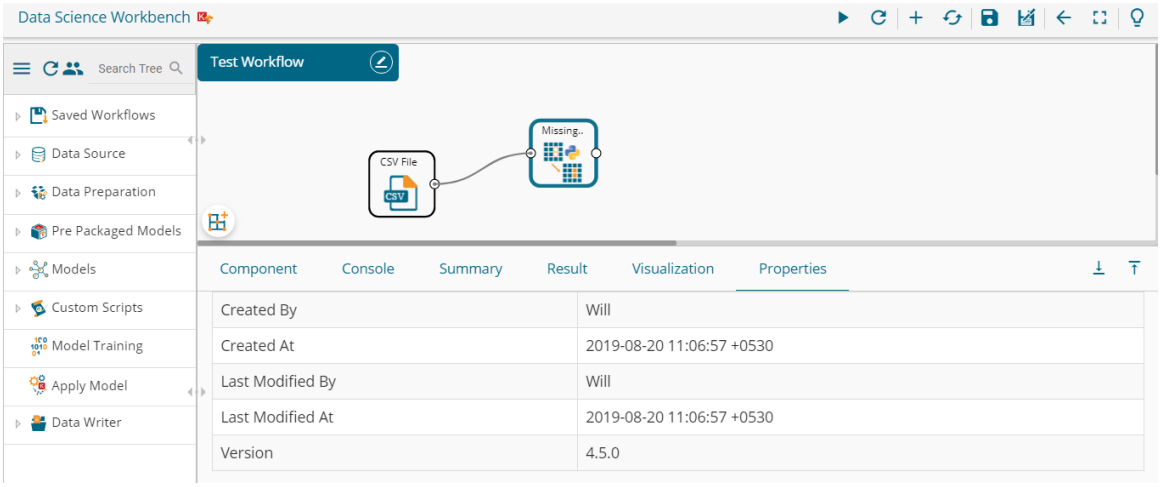
Workflow Name	Run by	Start time	End Time	Status	View Log	Live job status	Summary	Actions
save_feb5	user1A	5/Feb/2019-14:55:50	5/Feb/2019-14:55:53	success				
save_feb5	user1A	5/Feb/2019-14:52:21	5/Feb/2019-14:52:25	success				
wtfinal	user1A	21/Nov/2018-12:3:5	21/Nov/2018-12:3:18	failed				
sqlTr_P	user1A	21/Nov/2018-11:46:58	21/Nov/2018-11:47:4	success				
sparkSpilitdata	user1A	11/Oct/2018-12:39:38	NA	in progress				

Note: The Status tab appears when the user needs to check the live job status of a running job inside the Spark Workspace. The '**Status**' tab does not appear for other workspaces.

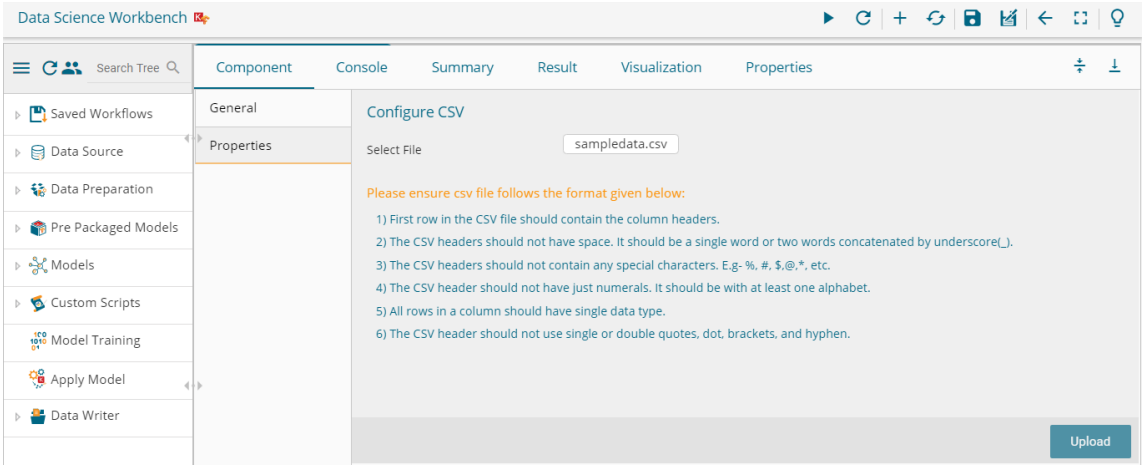
9. **Center-Top-Bottom icons:** These icons have been provided on the tabbed Menu Strip to customize the workspace and view space as per the user requirement.
   The Default view of the Data Science Workspace canvass is as shown below:



a. Click the '**Center**' icon to get equal space for the workspace and process view space.



b. Click the '**Top**' icon to maximize view space and minimize the workspace on the Predictive landing page.



Note: Click the '**Bottom**' icon to maximize the workspace as displayed in the default view of the Predictive Workspace landing page.

5. Data Sources

Acquiring data from a data source is the initial step to move ahead in the Data Science Workbench. The '**Data Source'** tree-node offers the following types of data source connectors:

a. CSV File
b. Data Service
c. Cassandra Reader
d. Data Store Reader
e. Zip File
f. SFTP Reader
g. HDFS Reader

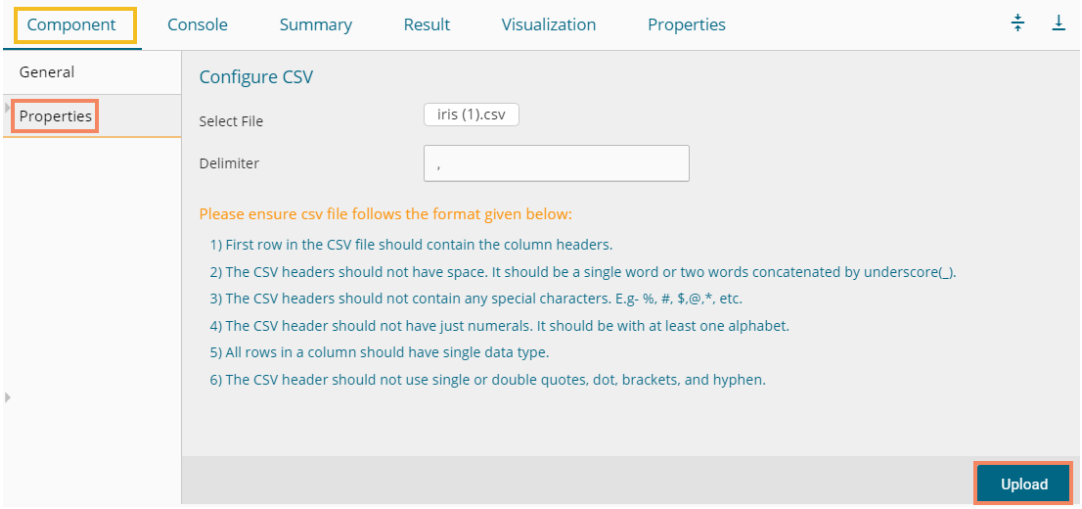
The present section aims at describing the steps to get data from all the above-mentioned data sources.

Note: The Data Source list may differ based on the Workspace. The configuration steps as displayed below for a specific data source connector remains the same across the workspaces.

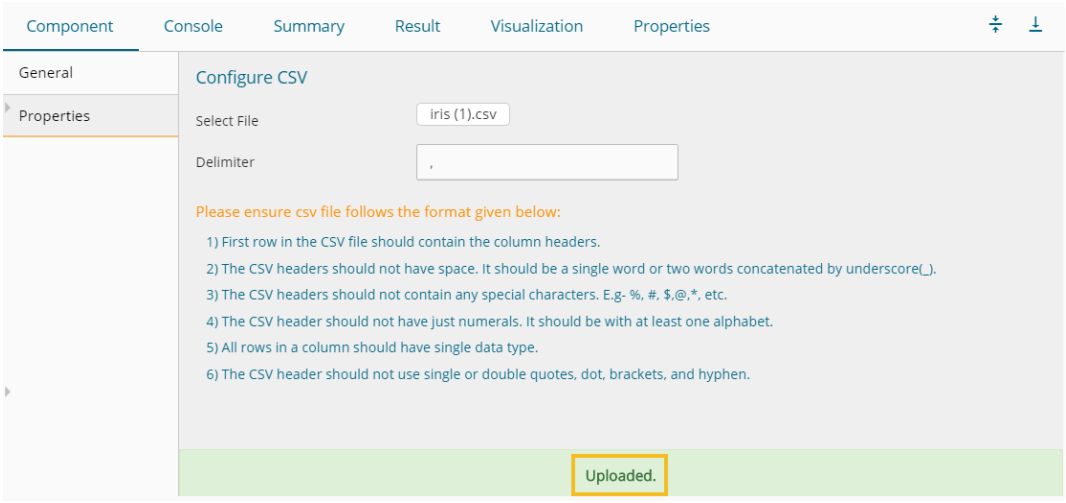
5.1. CSV File

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

iii) Configure the following fields for a data source:
   a. **Select File**: Browse a CSV file.
   b. **Delimiter:** Mention the delimiter used in the CSV file (it is a comma).
iv) Click the **'Upload'** option.



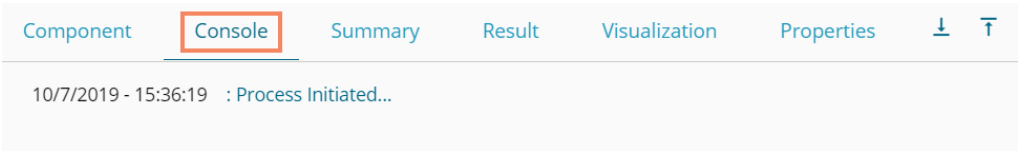
v) The user should get a success message, as highlighted in the image given below:



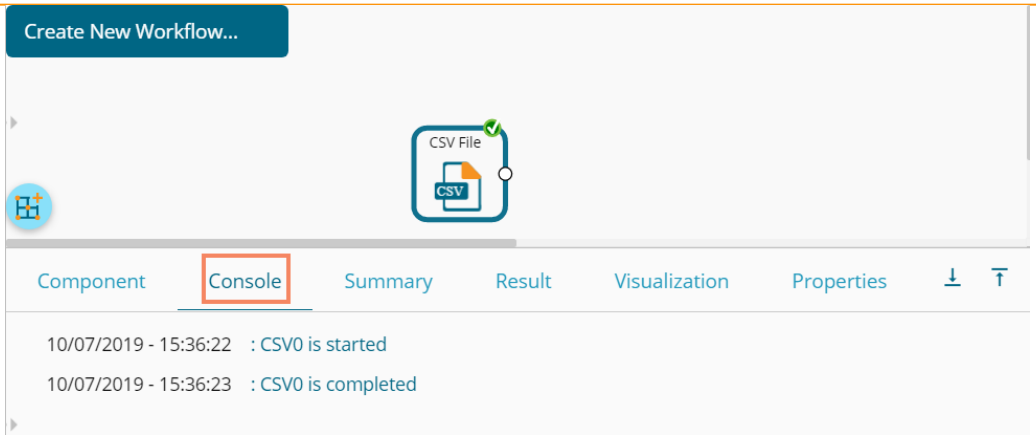
vi) Click the '**Run**' or '**Refresh**' icon.

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

a. It first displays that the process has been initiated.



b. The completion of the process is marked with a green checkmark on the dragged component.

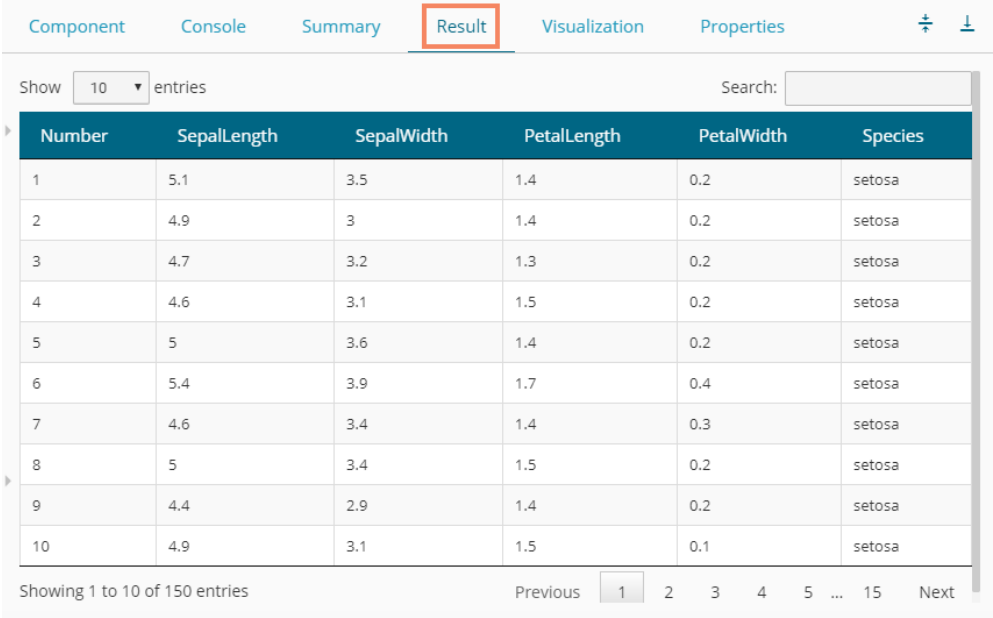


viii) After the Console process gets completed, the uploaded data appears under the 'Result' tab.

ix) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.



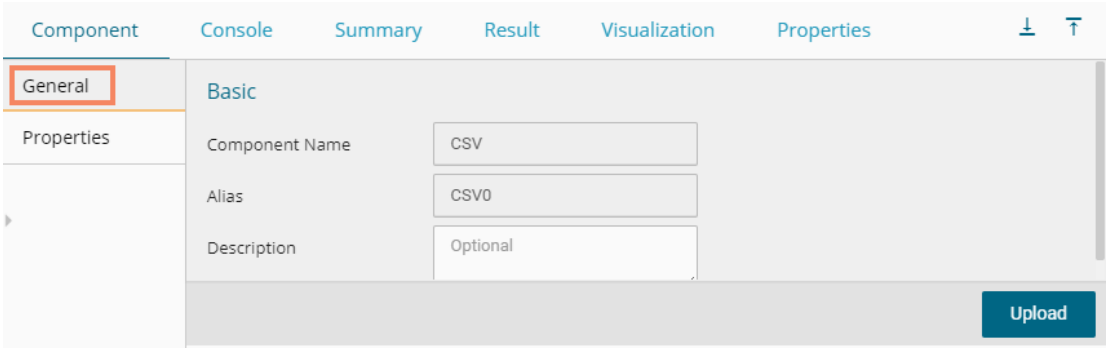
- **Rules to be followed while uploading a CSV File**
  1. The first row provided in the CSV file should contain the column headers.
  2. The second row of the CSV file should contain the data under all the headers without any 'null' or 'NA.'
  3. CSV headers should not have space. It should be a single 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, dots, 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 are the '.csv' and '.tsv'

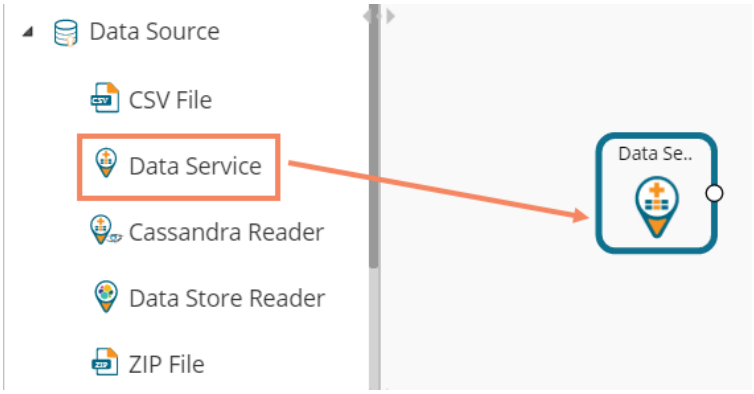
b. All the supported data sources get the '**General**' tab to configure the following information for any tree-node component:

i. Component Name: A predefined name of the component is displayed in this field
ii. Alias: A predefined component name appears with the number to provide a record of its sequence in the workflow.
iii. Description (it is an optional field)
(E.g. the following image displays the '**General**' tab for a CSV data source.)
Click the 'Upload' option after providing the required details



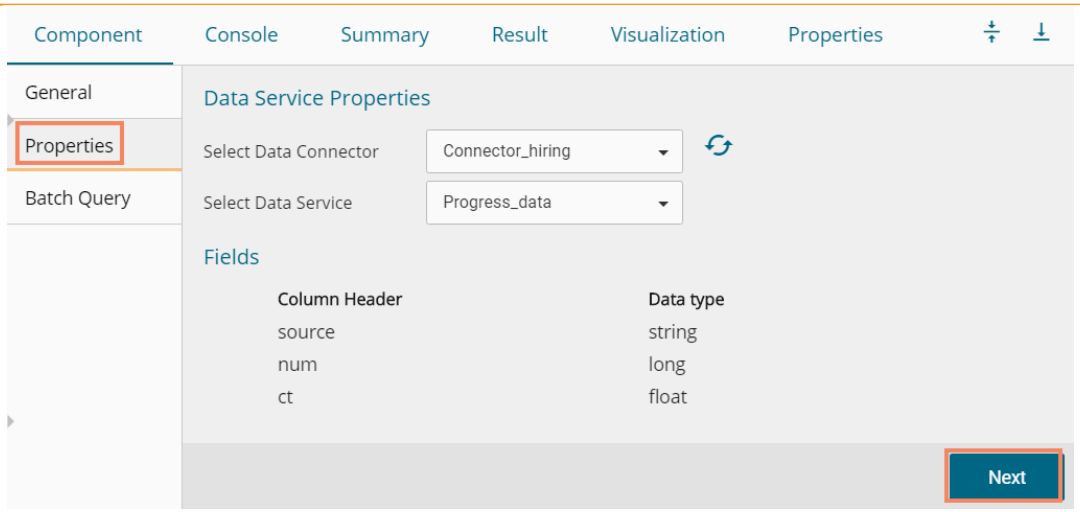
5.2. Data Service

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

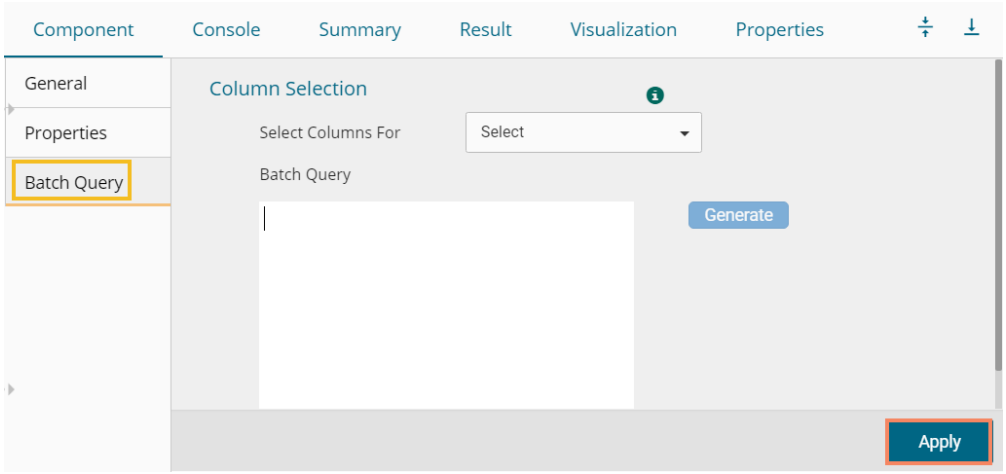


iii) The **'Properties'** fields open for the Data Service data source connector under the '**Components'** tab.
iv) Configure the **'Data Service Properties':**

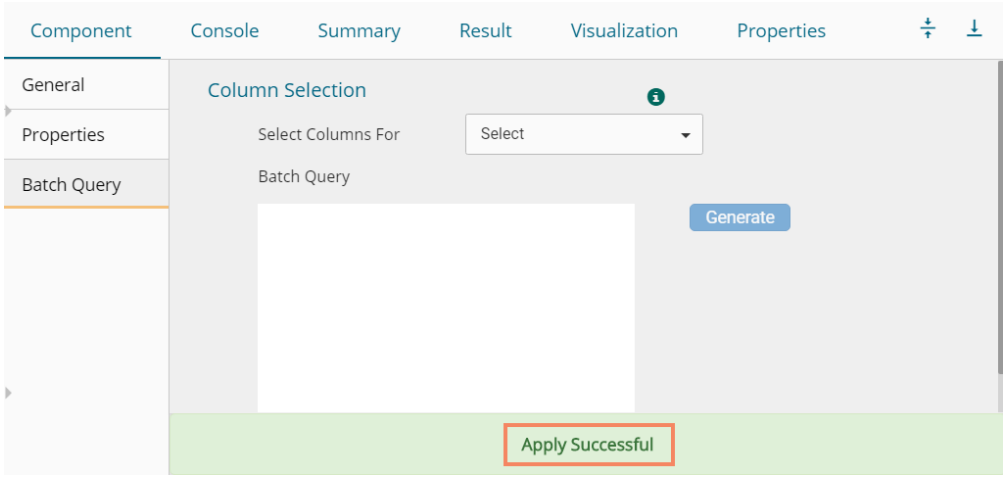
a. **Select Data Connector**: Select a data source from the drop-down menu
b. **Select Data Service**: Select a query service from the drop-down menu
c. **Fields**:
The following tables get displayed:
i. Column Header
ii. Data Type
d. Click the '**Next**' option.



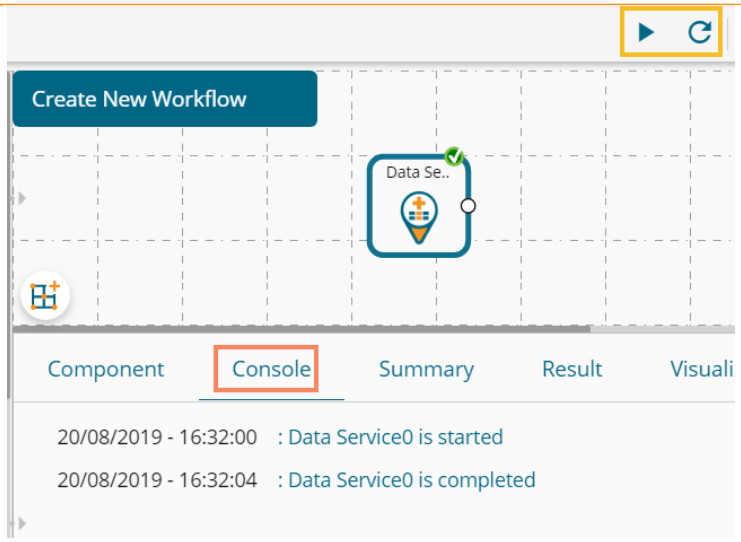
e. The Batch Query tab gets displayed.
f. Click the 'Apply' option from the Batch Query page.



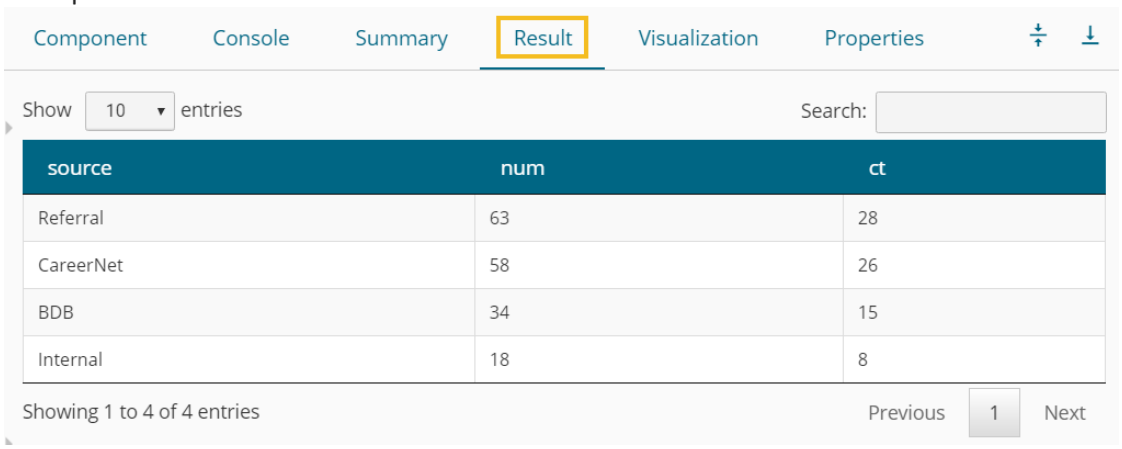
g. A success message appears if the Apply is successful.



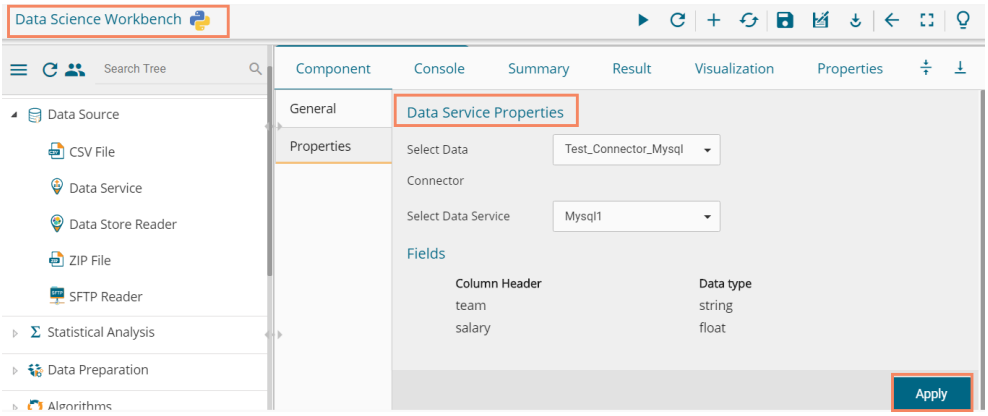
h. Click the '**Run**' or '**Refresh**' icon to start the Console process.
i. The completion of the Console process gets marked with the Green checkmark on the data source connector.



j. The uploaded data from the data service data source can be seen under the 'Result' tab.



Note: The Batch Query tab appears only for the R Workspace. The Properties tab for the Data Service connector in the Python Workspace appears with the Apply option if the Data Service does not contain any filter.



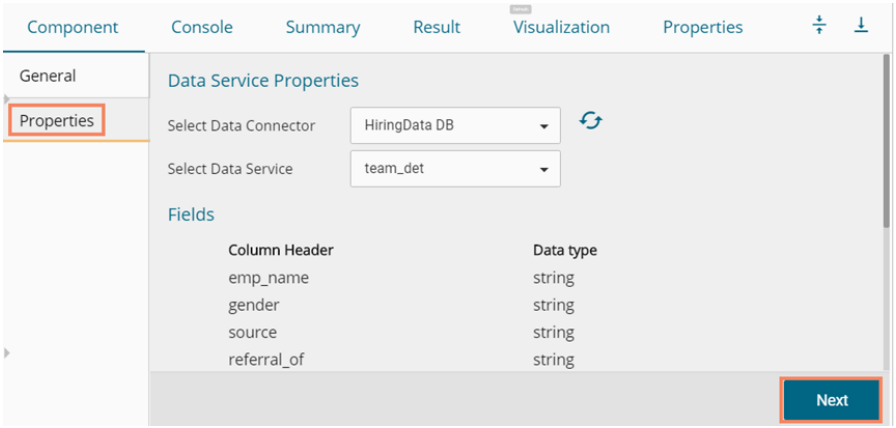
5.2.1. Data Service with Conditions (Filters)

The Conditions tab appears for the Data Service that has filters. The Data Science Workbench supports Text, LOV, and Batch Query control types to configure the Conditions tab. The section aims to explain them in detail.

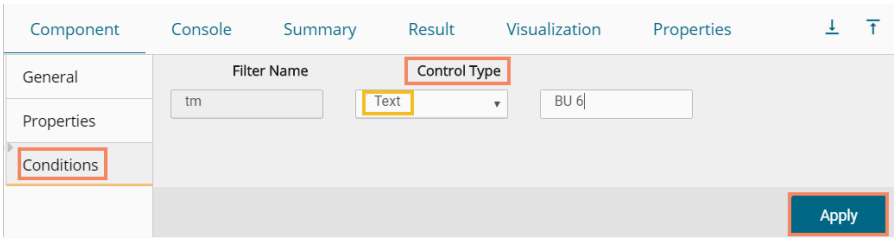
5.2.1.1. Text Control Type

The filter value needs to be configured manually for the '**Text**' control type option. The user can enter multiple filter values separated by a comma.

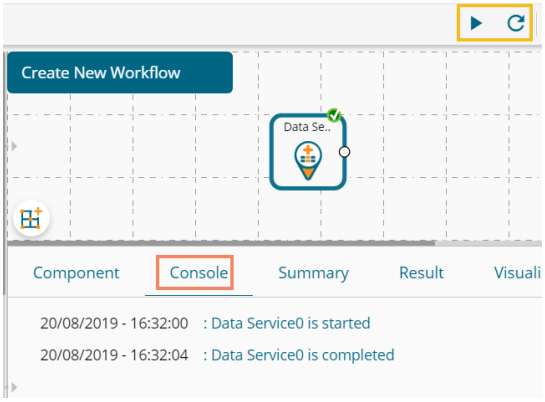
i) Drag a Data Service data source connector to the workspace canvas.
ii) Choose a Data Connector and Data Service using the '**Properties**' tab.
iii) Click the '**Next**' option.



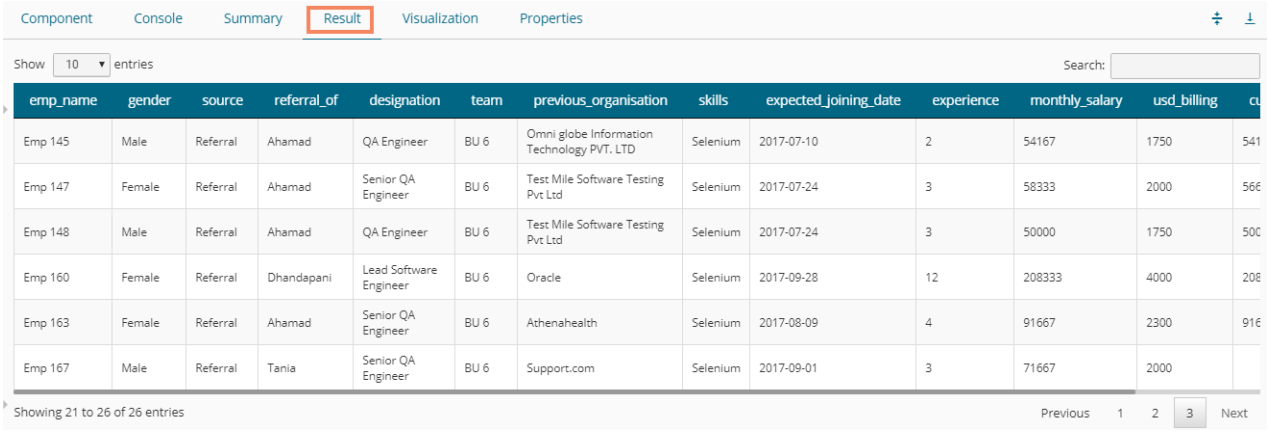
iv) The Conditions tab opens.
v) Select the '**Text**' as a Control Type option.
vi) Manually enter the Filter value.
vii) Click the 'Apply' option.



viii) Click the '**Run**' or '**Refresh**' icon to begin the Console process.
ix) The completion of the process gets marked by a green checkmark on the top of the component.

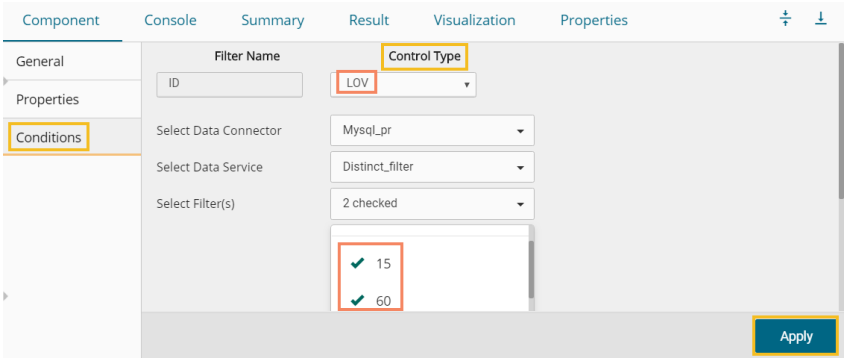


x) The filtered data for team BU 6 gets uploaded from the selected data service.

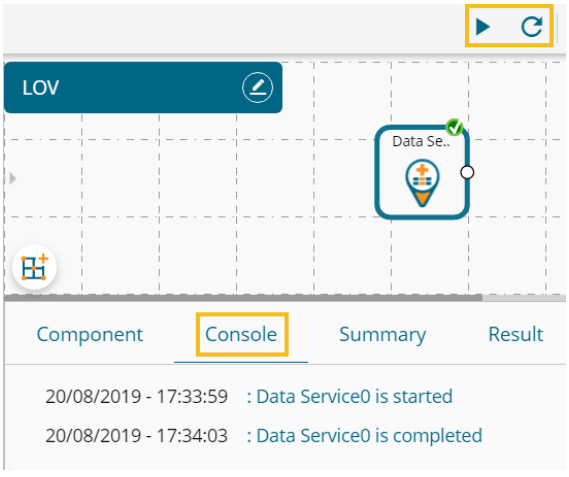


5.2.1.2. LOV Control Type

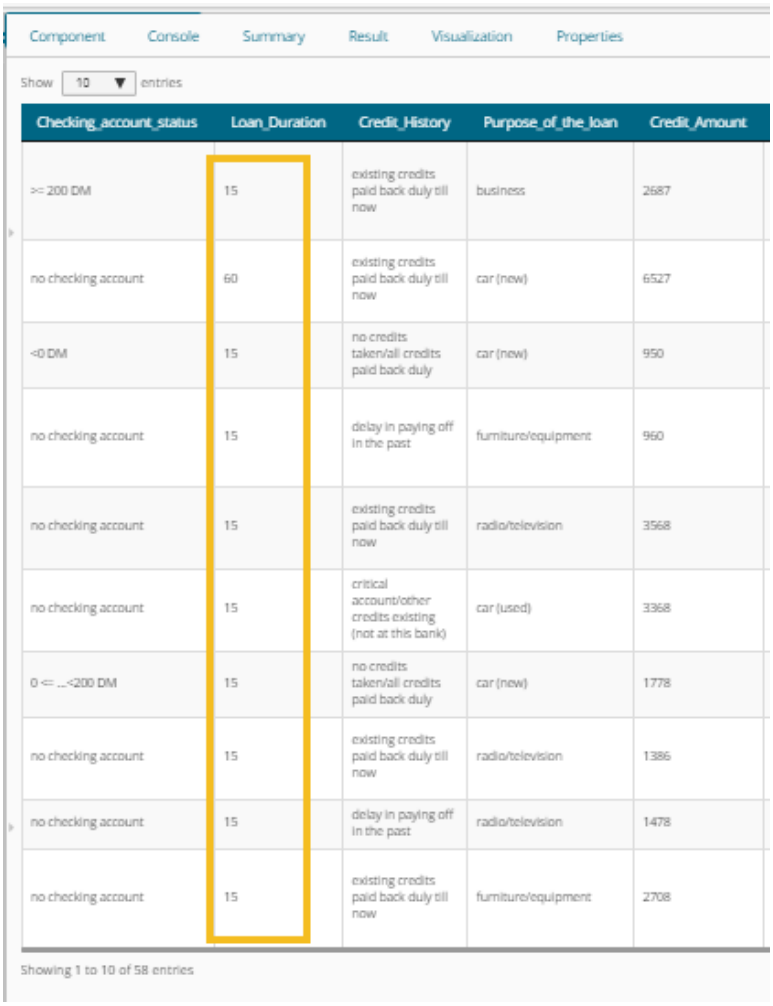
i) The Conditions tab opens.
ii) Select '**LOV**' as a Control Type option.
iii) Select another Data Connector and Data Service from the lookup.
iv) Select filter value
v) Click the 'Apply' option.



vi) Run the component to get data.



vii) The filtered data for the provided values gets uploaded.

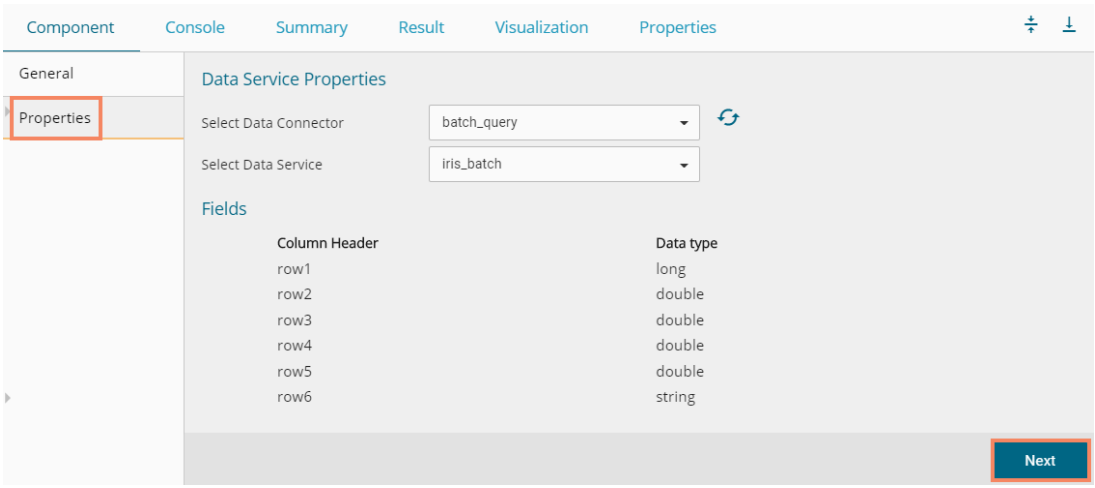


5.2.1.3. Batch Query

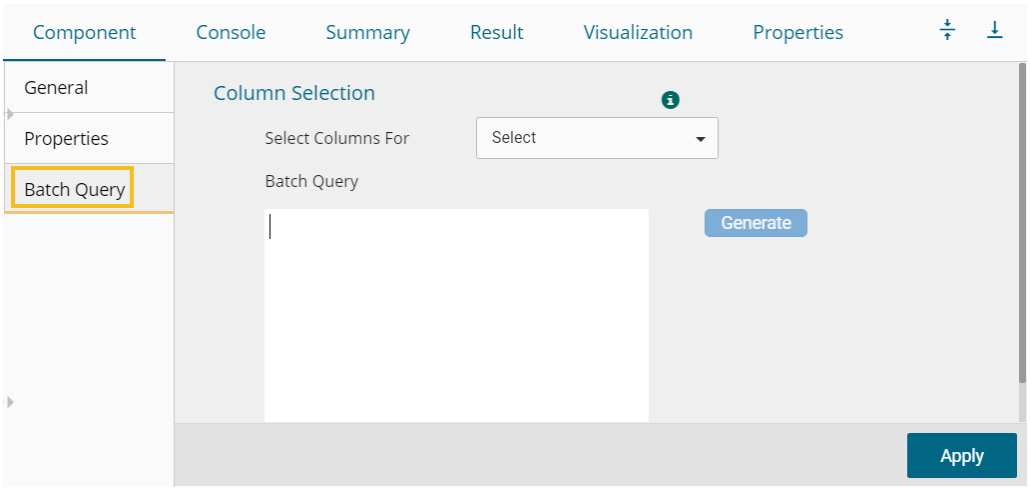
i) Drag and drop a Data Service connector to the Workspace canvas.

ii) Configure the Properties tab.

iii) Click the **'Next'** option.

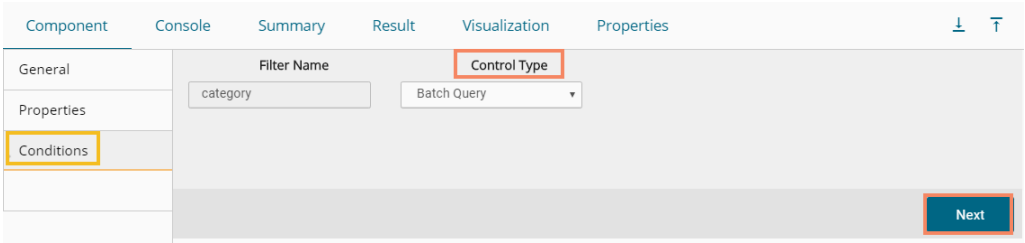


iv) If the selected data does not contain a filter, then while clicking the '**Next**' option, the Batch Query tab appears.



Or

The '**Conditions**' tab opens (if the selected data service contains filter values). Select the '**Batch Query**' option as the Control Type. Select the '**Next**' option.

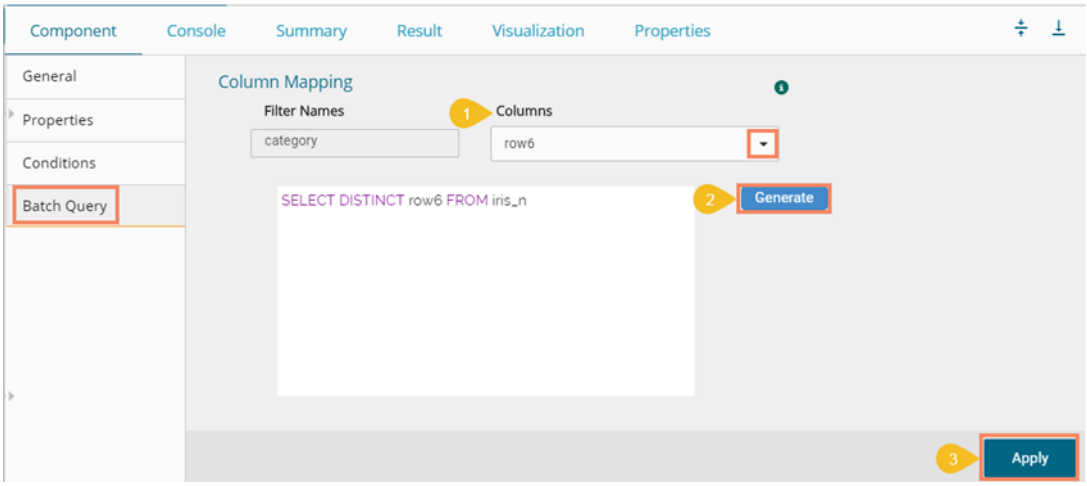


The '**Batch Query**' tab appears.

v) Select a column using the '**Columns**' drop-down menu.

vi) Click the '**Generate**' option to generate a batch query.

vii) Click the 'Apply' option after configuring the '**Conditions**' tab.

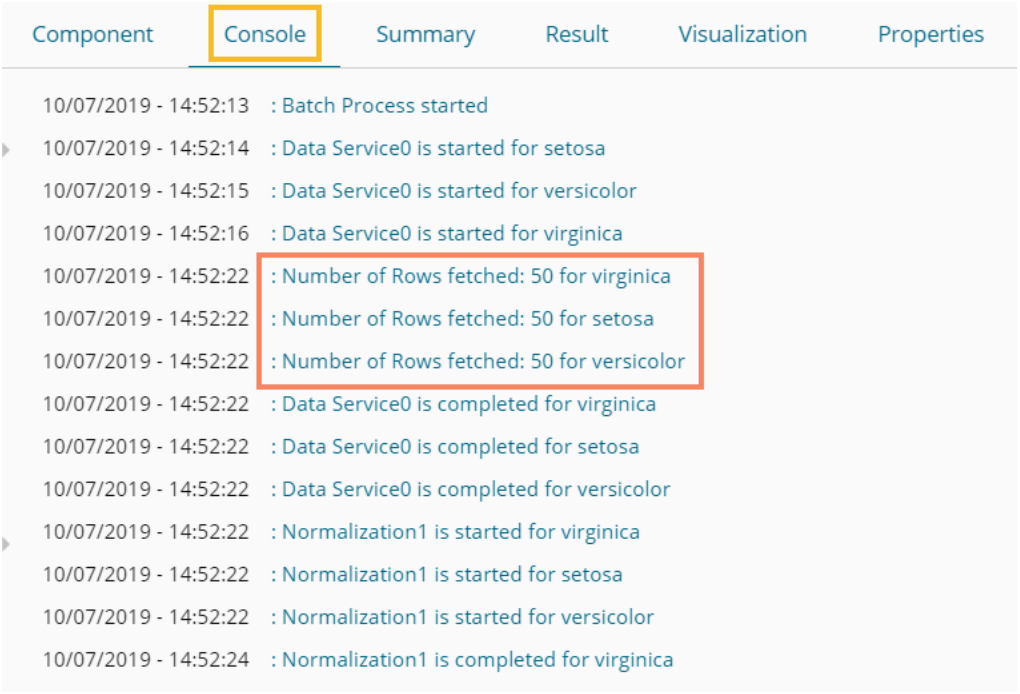


viii) The '**Apply Successful**' message appears.

ix) To see batch-wise completion of the process under the 'Console' tab connect the Data Service component to a data writer. E.g., the following image displays the normalization and internal data writer connectors connected to the Data Service component.

x) Configure the component and run the workflow.

xi) Open the batch-wise completion of the process that can be seen under the 'Console' tab.



Note:

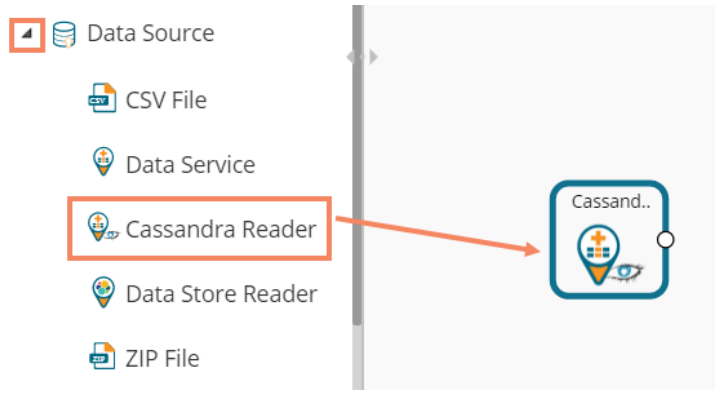
a. The Result tab displays no data in the case of the Batch Query option in the R workspace.
b. The Batch Query option is available only for the **R** and **Python** Workspaces.
c. The user can develop a data service via the Data Management module of the BDB Platform.
d. **'Fields'** option under the '**Properties**' tab appears only after selecting the appropriate query service.
e. LOV service provided under the '**Conditions**' tab can contain only one column, in case of more than one column, a warning message appears.
f. The user can configure the following information for a data service data source via the '**General**' tab:
    i. Alias Name
    ii. Description (it is an optional field)

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

5.3. Cassandra Reader

The Cassandra Reader data source connector is provided for R and Spark ML workspaces.

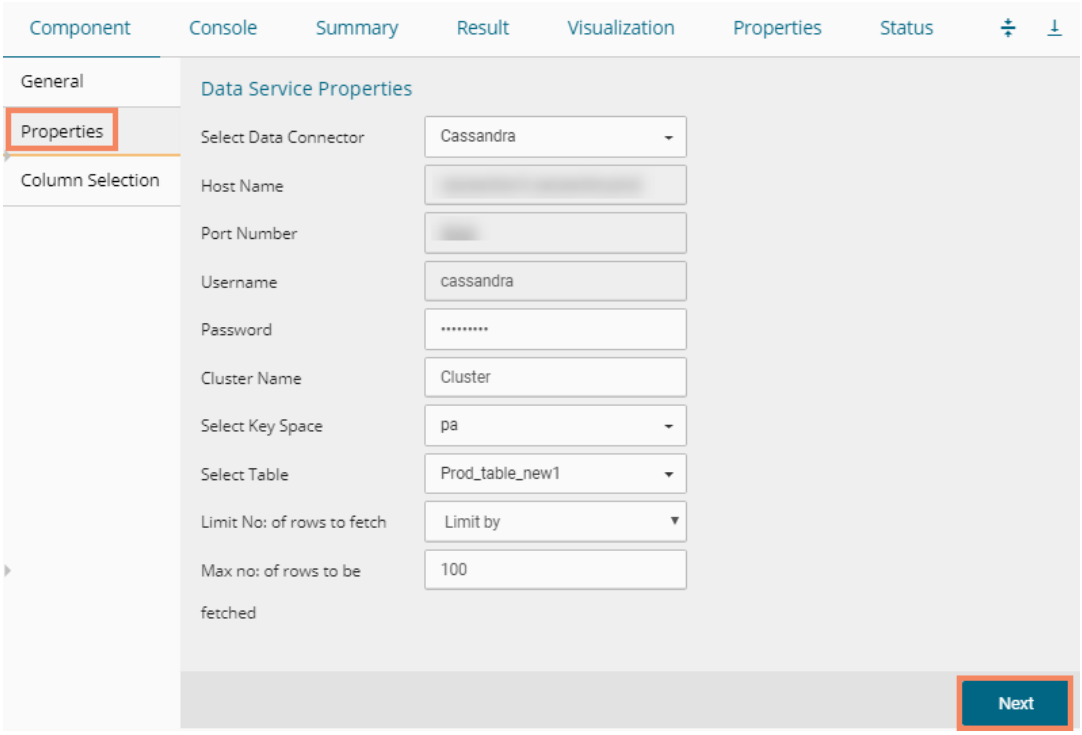
i) Select and drag the '**Cassandra Reader**' component onto the workspace.
ii) Click on the dragged '**Cassandra Reader**' component.



iii) Users will be redirected to the '**Properties**' tab of the component.
iv) Configure the required properties:
  a. Select Data Connector: Select a data connector using the drop-down menu
  b. Host Name: Data connector specific hostname will be displayed
  c. Port Number: Port number will be displayed
  d. User Name: Username gets displayed
  e. Password: Enter the password
  f. Cluster Name: Enter a cluster name
  g. Select Key Space: Select a keyspace from the drop-down menu
  h. Select Table: Select a table from the drop-down menu
  i. Limit No. of the row to fetch: Select an option using the drop-down menu. Two options are provided, as shown below:
    1. Select all Rows
    2. Limit By

j. Max. No. of Rows to be fetched: Enter a number to decide maximum fetched rows. (This option appears only if the '**Limit By**' option has been selected using the '**Limit by Row**' field. The default value for this field is 1000).

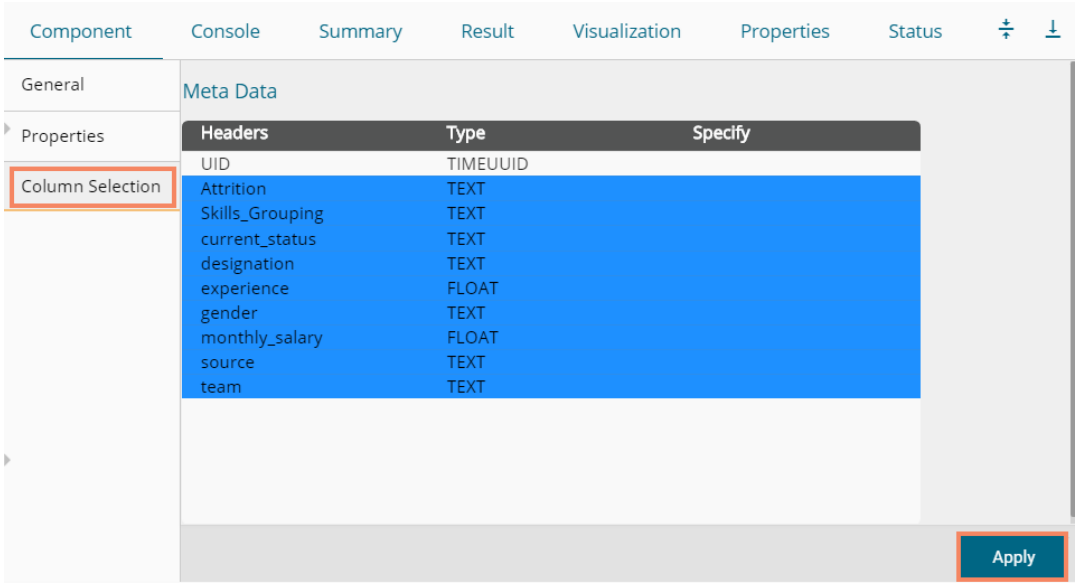
v) Click the '**Next**' option.



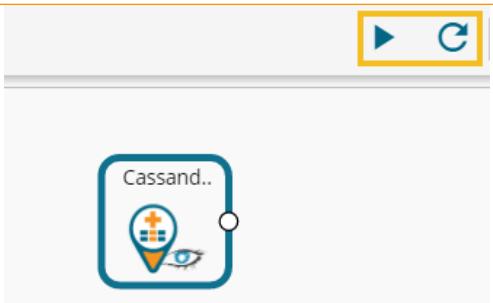
vi) Users will be redirected to the '**Column Selection**' tab.

vii) Select the required columns from the list.

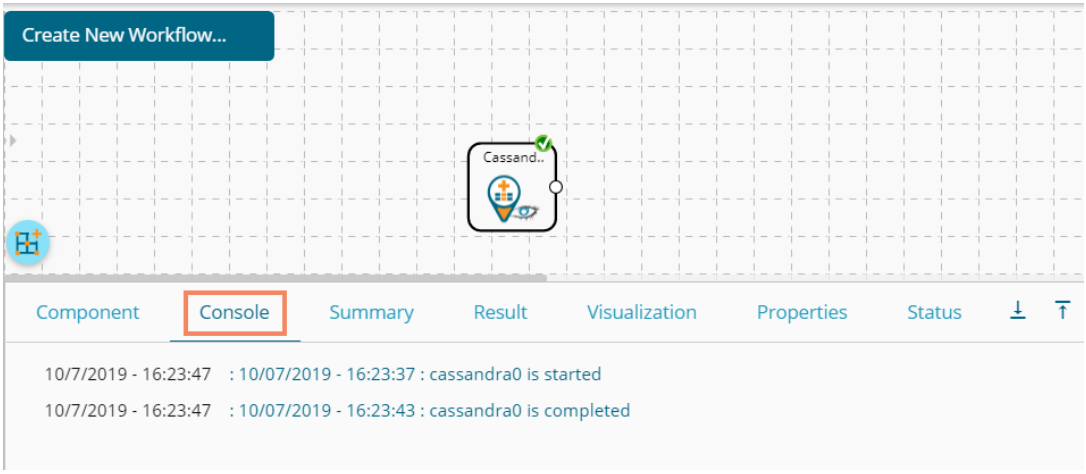
viii) Click the 'Apply' option.



ix) Run the component process for fetching data clearing the Cache.



x) The 'Console' tab opens to display the progress of the process. The completion of the Console process is marked through the green checkmark on the top of the component.

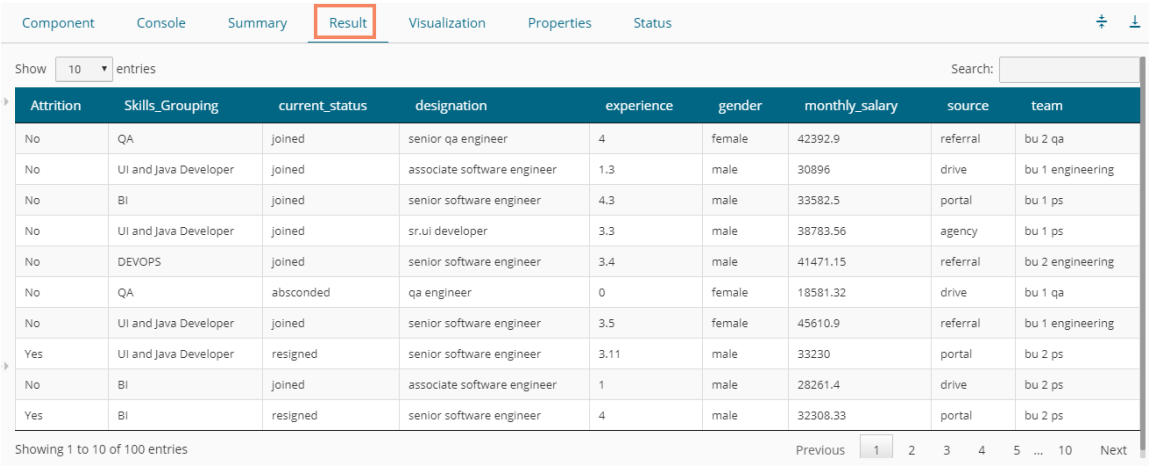


xi) After the Console process gets completed, users can view the Result data using the 'Result' tab.

xii) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.

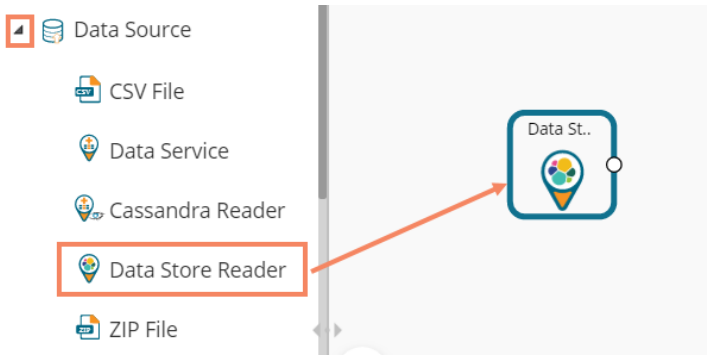


Attrition	Skills_Grouping	current_status	designation	experience	gender	monthly_salary	source	team
No	QA	joined	senior qa engineer	4	female	42392.9	referral	bu 2 qa
No	UI and Java Developer	joined	associate software engineer	1.3	male	30896	drive	bu 1 engineering
No	BI	joined	senior software engineer	4.3	male	33582.5	portal	bu 1 ps
No	UI and Java Developer	joined	sr.ui developer	3.3	male	38783.56	agency	bu 1 ps
No	DEVOPS	joined	senior software engineer	3.4	male	41471.15	referral	bu 2 engineering
No	QA	absconded	qa engineer	0	female	18581.32	drive	bu 1 qa
No	UI and Java Developer	joined	senior software engineer	3.5	female	45610.9	referral	bu 1 engineering
Yes	UI and Java Developer	resigned	senior software engineer	3.11	male	33230	portal	bu 2 ps
No	BI	joined	associate software engineer	1	male	28261.4	drive	bu 2 ps
Yes	BI	resigned	senior software engineer	4	male	32308.33	portal	bu 2 ps

5.4. Data Store Reader

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

ii) Click the '**Data Store Reader**' component.



iii) Users will be redirected to the '**Properties**' tab of the component.

iv) Configure the required properties:

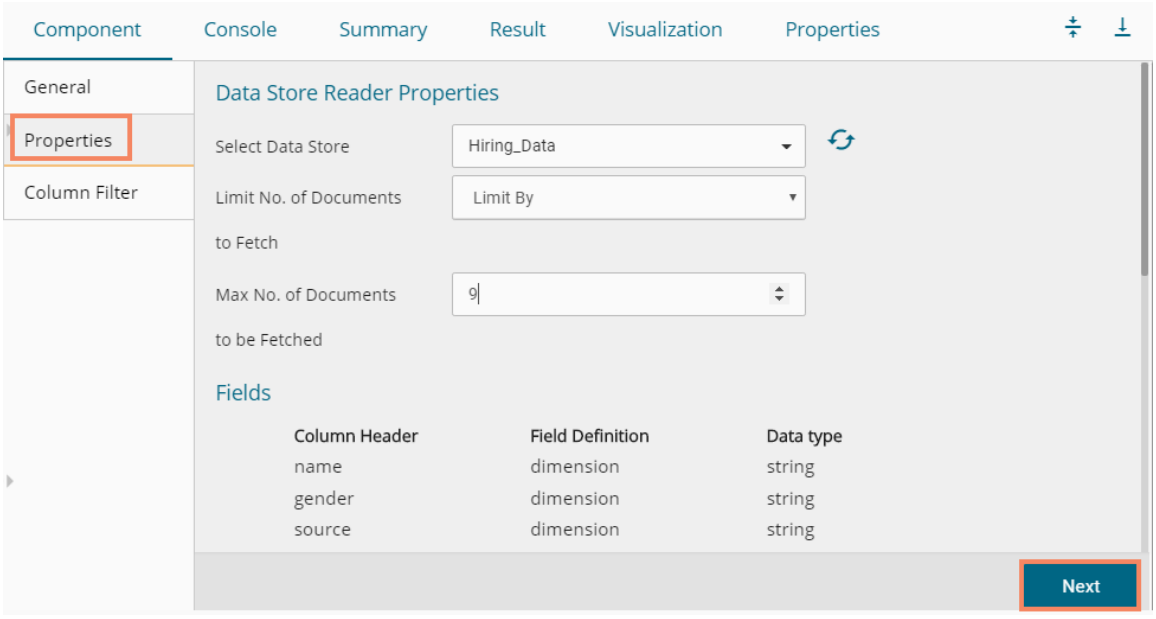
a. Select Data Store: Select a data store using the drop-down menu.

b. Limit No. of Documents to Fetch: Select an option using the drop-down menu. Two options are provided, as shown below:

1. Fetch all Documents
2. Limit By

c. Max. No. of Documents to be Fetched: Enter a number to decide maximum fetched documents (This option appears when the 'Limit By' option has been selected using the 'Limit No. of Documents to Fetch' field. Users can select any positive integer value).

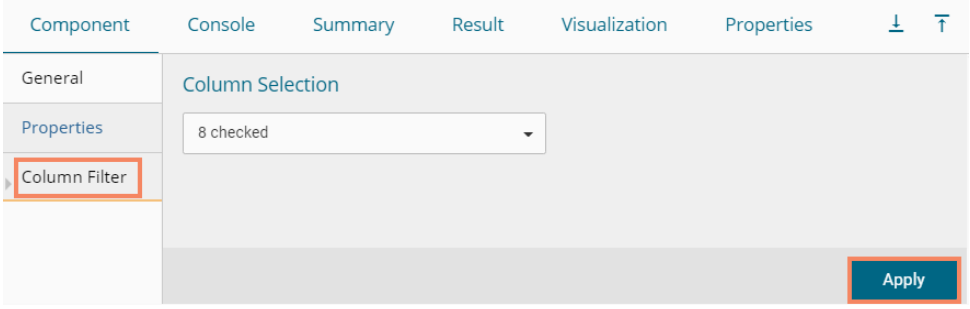
v) Click the '**Next**' option.



vi) The '**Column Filter**' tab opens.

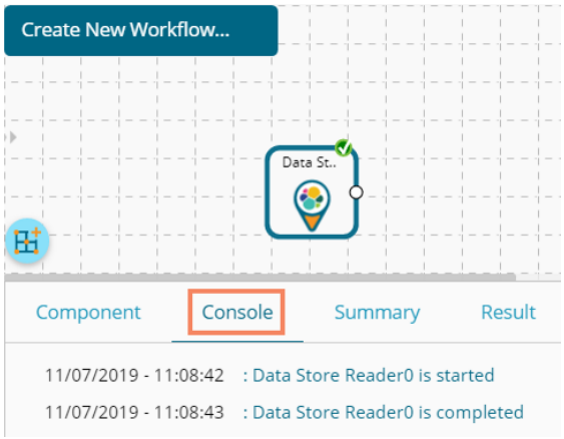
vii) Select the required columns from the drop-down list.

viii) Click the 'Apply' option.



ix) Run the component by clearing the previous cache to get Data.

x) The 'Console' tab opens to display the progress of the process. The completion of the Console Process is marked by a green checkmark on the top of the dragged Datastore component.

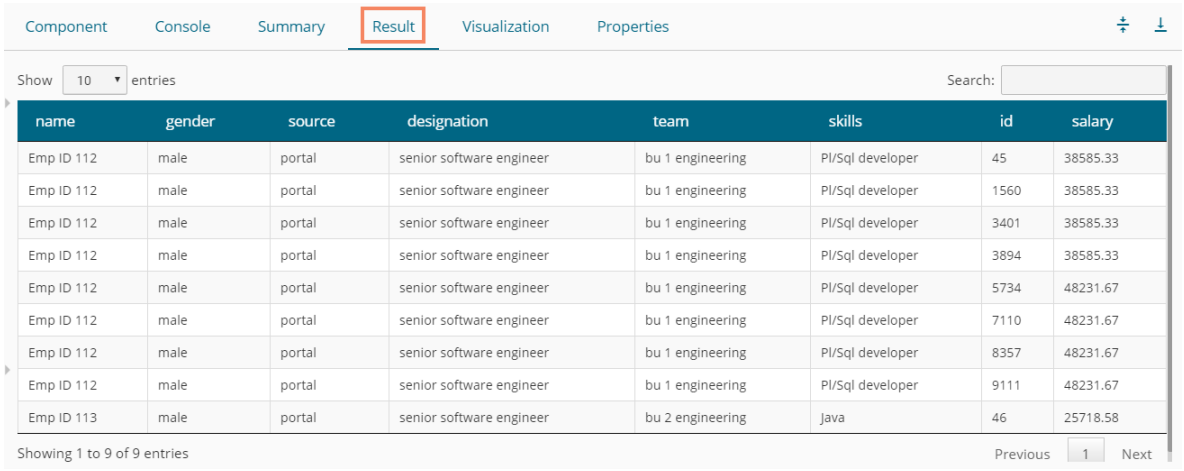


xi) The user can view the Result data using the 'Result' tab.

xii) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.

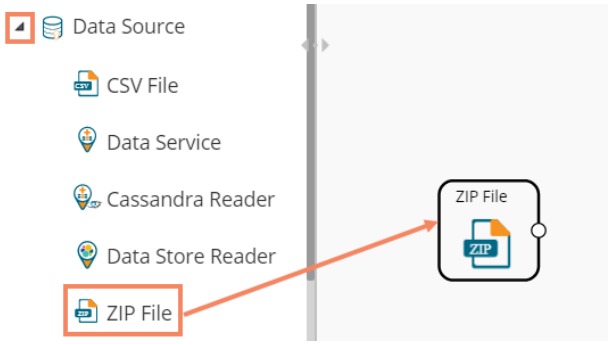


Note: Empty values present in any row of the numeric column get replaced with zero (0) while reading data from a data store reader.

5.5. Zip File

i) Select and drag the **'Zip File'** component onto the workspace.

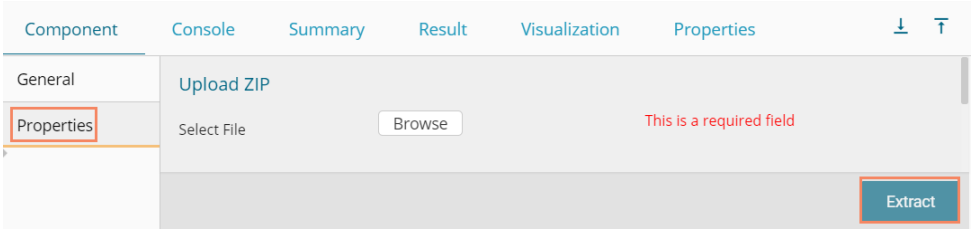
ii) Click the **'ZIP File'** component.



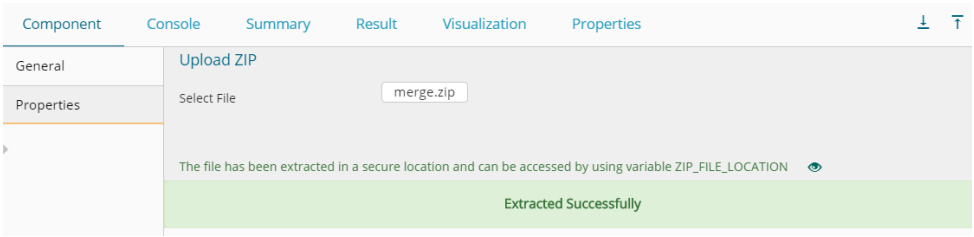
iii) The Properties tab opens for the Zip File.

iv) Browse a Zip file.

v) Click the **'Extract'** option.

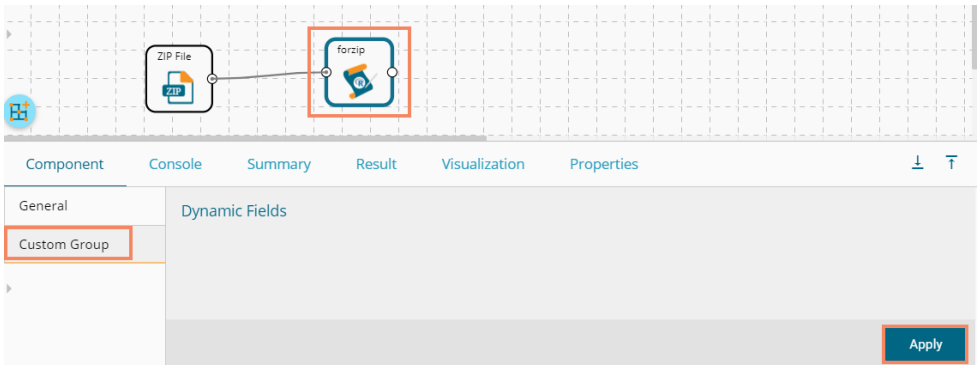


vi) After extracting data, the following message appears.



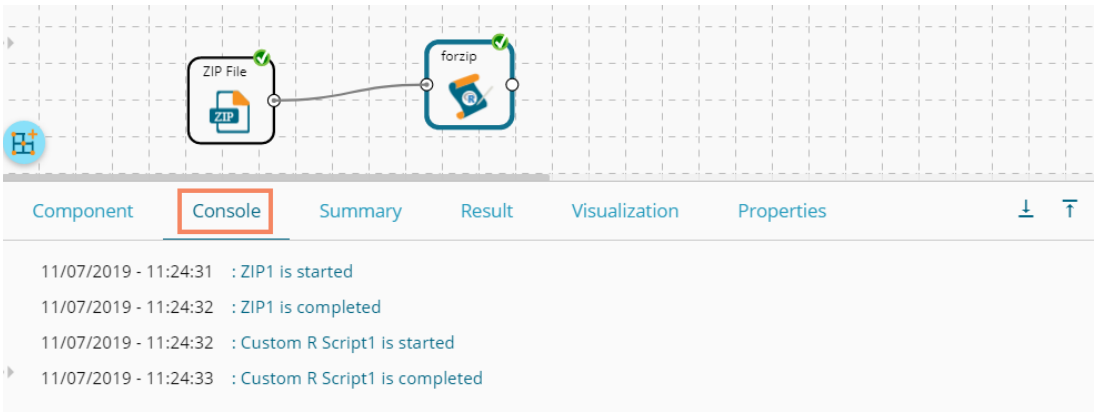
vii) Connect the dragged ZIP file to a script component to read the extracted data from the ZIP file.

viii) Click the 'Apply' option for the Custom Group tab of the script component.

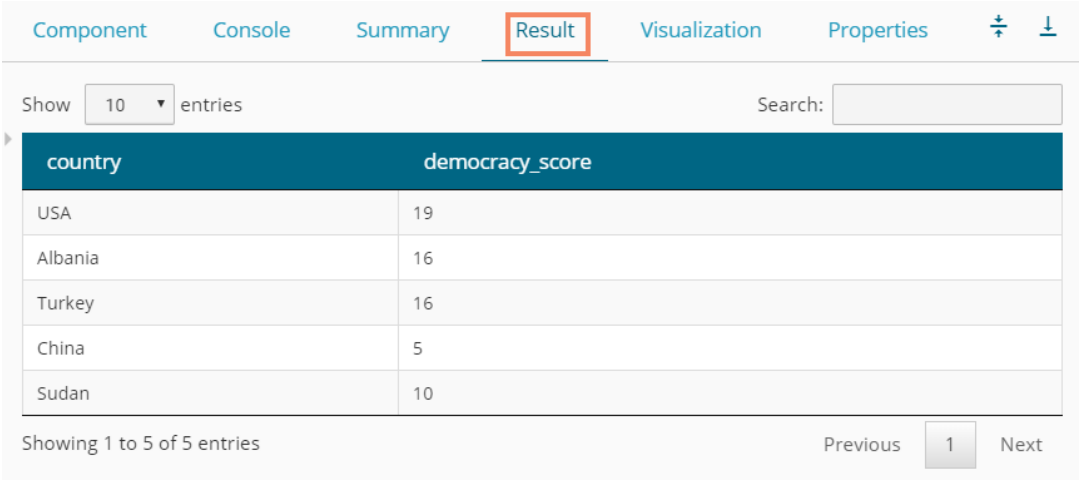


ix) After getting the '**Apply Successful**' message, click the '**Run**' or '**Refresh**' icon to get started with the process.

x) The progress of the process appears under the 'Console' tab, and the completion of the process gets marked by the green checkmarks on the top of the dragged components.



xi) After the successful completion of the Console process, open the 'Result' tab to view the Result data.



Note:

a. In the R workspace, the ZIP file can have files with the following extensions- .csv, .xlsx, and .json
b. The ZIP file will have the following properties:
   1) Extensions supported for ZIP will be ".zip",".tar",".rar",".7z", "tar.gz".
   2) The ZIP file data source should only get connected to the Custom Scripts. If connected to any other component, an error should occur, saying, "Cannot be connected. Connect to Custom Scripts".
   3) After uploading a ZIP file, the contents of the ZIP file get shown in UI after decompressing it.
   4) Within the script, the files in the zip can be accessed with the drive location.

5.6. SFTP Reader

The SFTP reader is provided to handle enormous data for the Python Workspace. The SFTP reader can read data from any file extension using a relevant Script.

i) Select and drag the '**SFTP Reader'** component onto the workspace.

ii) Click the **'SFTP Reader'** component.

iii) The Properties tab opens for the SFTP data source connector.

iv) Configure the required details:

a. Host address

b. Port number

c. Username

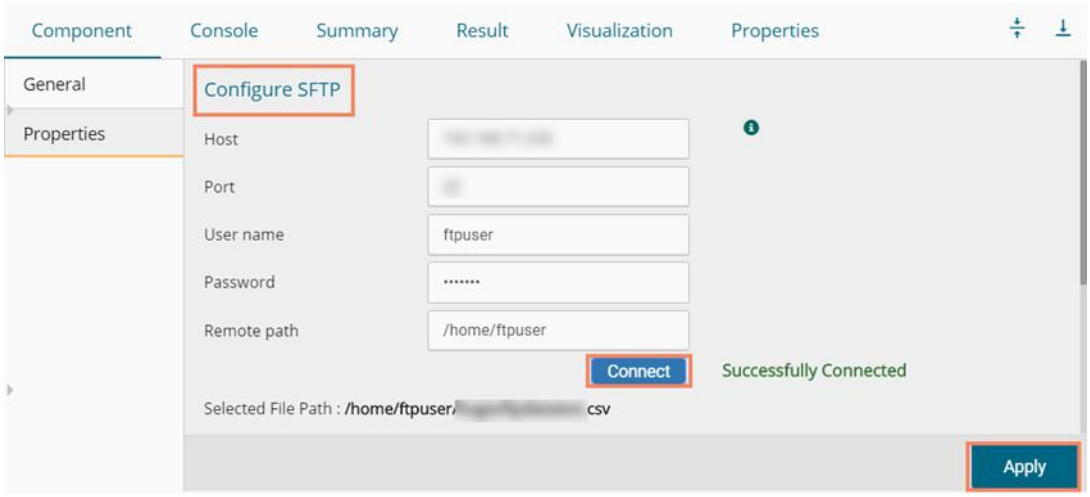
d. Password

e. Remote Path

f. Click the '**Connect**' option. It should return a notification that it is successfully connected.

g. The user can select a file with a double click from the available options. The selected file path gets mentioned.

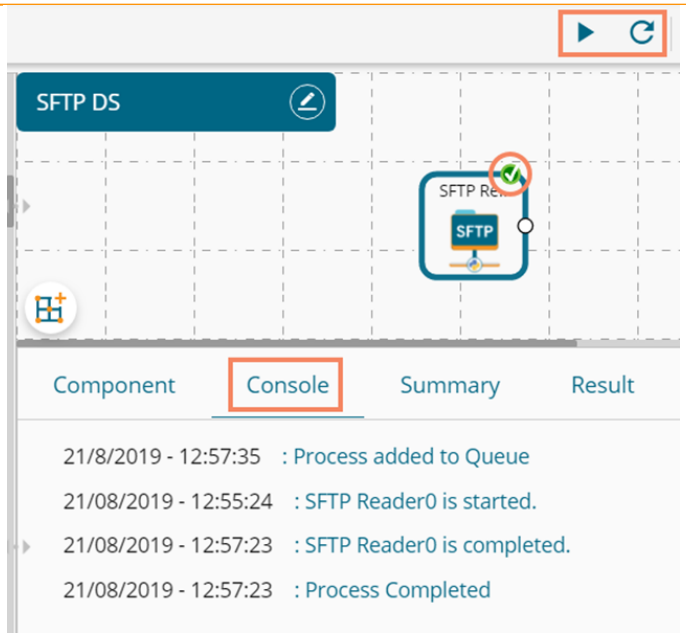
v) Click the 'Apply' option.



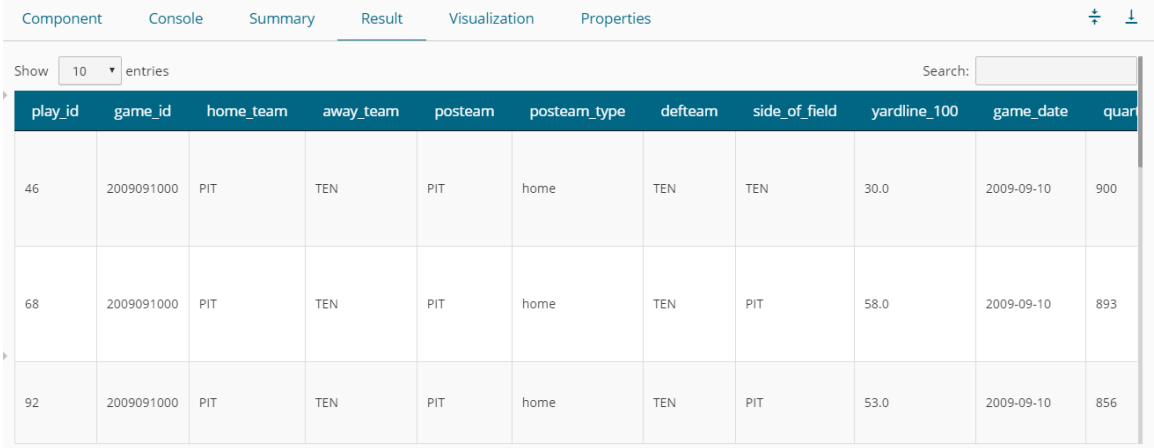
vi) A success message should appear, stating that the data source has been applied.

vii) Run the component process to get data.

viii) Completion of the 'Console' process gets marked by a green checkmark on the top of the dragged SFTP reader component.



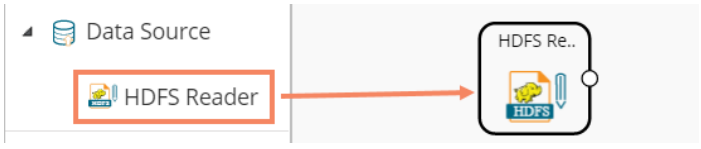
ix) The fetched data appears under the 'Result' tab.



5.7. HDFS Reader

The HDFS Reader is provided for the PySpark workspace. The HDFS reader loads distributed data (in batches) and supports only CSV extension.

i) Select and drag the **'HDFS Reader'** component onto the workspace.
ii) Click the **'HDFS Reader'** component.



iii) The Properties tab opens for the HDFS data source connector.

iv) Configure the required fields.

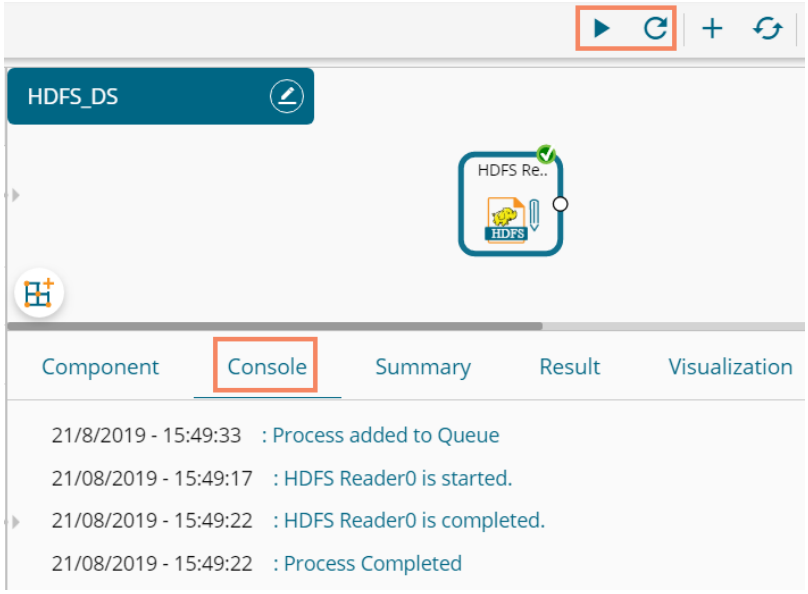
v) Click the '**Connect**' option. It should return a success message that is successfully connected.

vi) The user can select a file with a double click from the available options. The selected file path gets mentioned.

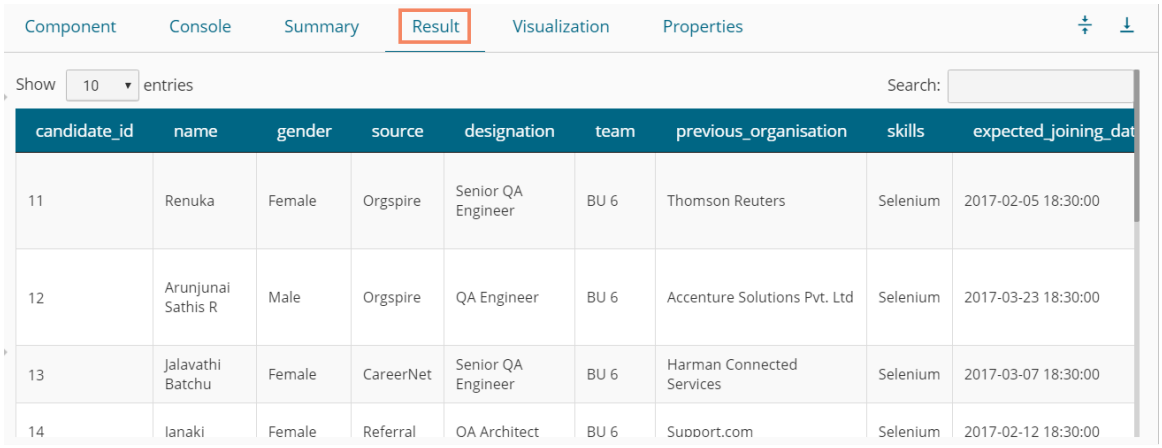
vii) Click the '**Apply**' option.

viii) Run the workflow after getting the success message.

ix) The progress of the process appears under the '**Console**' tab. The completion of the process gets marked by a green checkmark on the top of the dragged HDFS reader component.



x) The fetched data appears under the 'Result' tab.

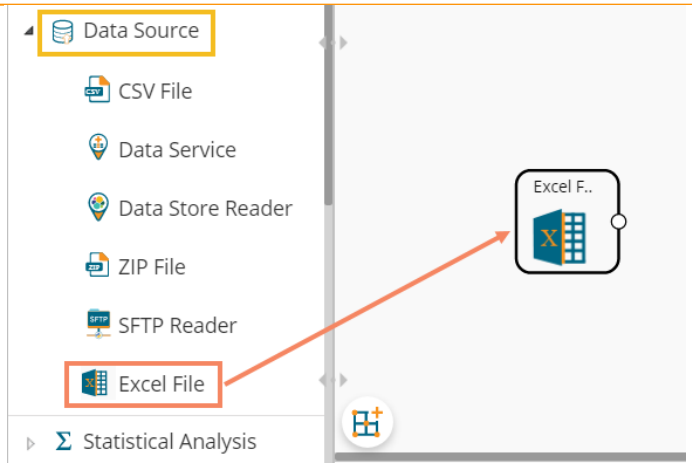


5.8. Excel File

The Excel File reader is provided in the **Python Workspace** to handle minutes to large data from your spreadsheets and make it analytics-ready.

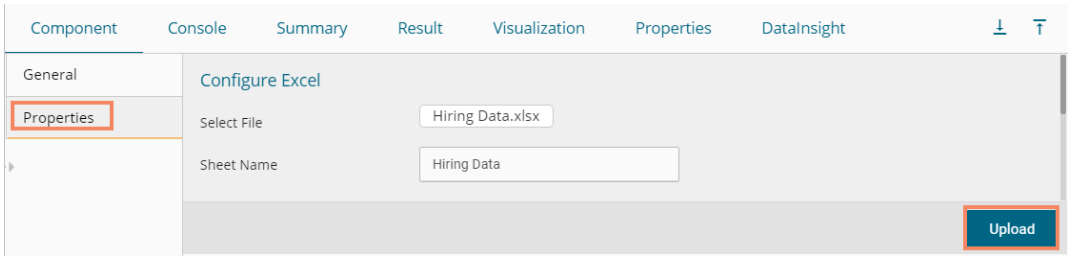
i) Select and drag the **'CSV File'** component onto the workspace.

ii) Click the **'Excel File'** component.

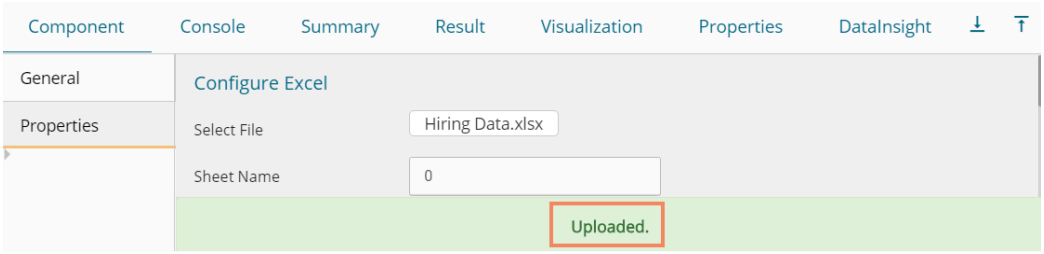


iii) Configure the following fields for a data source:
   a. **Select File**: Browse an Excel file.
   b. **Sheet Name:** Provide the sheet name.

iv) Click the **'Upload'** option.

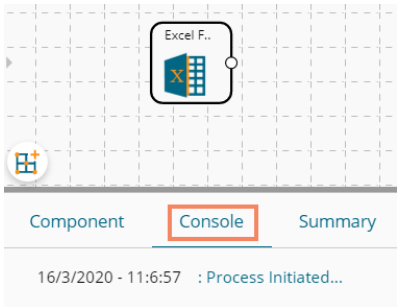


v) The user should get a success message, as highlighted in the image given below:

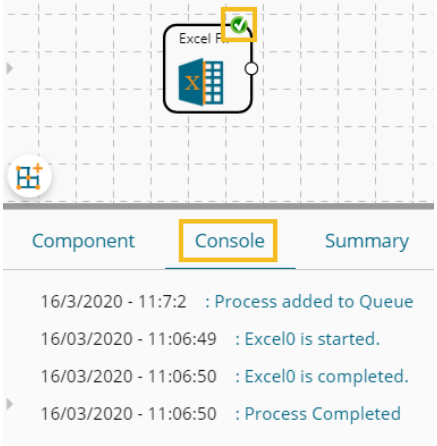


vi) Click the '**Run**' ▶ or '**Refresh**' ↻ icon.

vii) The users will be redirected to the '**Console**' tab to display the progress of the process.
   a. It first displays that the process has been initiated.



b. The completion of the process is marked with a green checkmark on the dragged component.

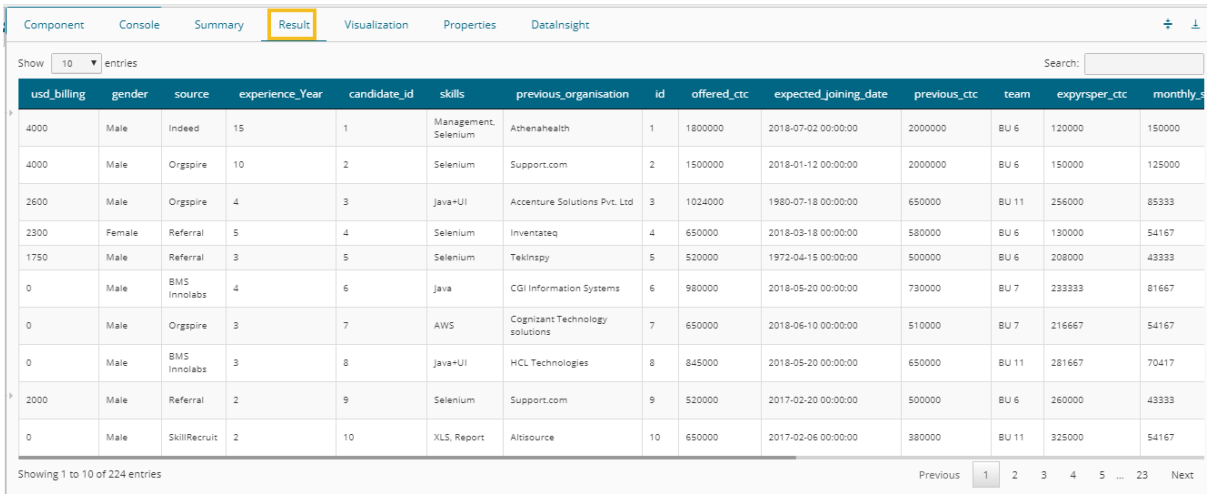


viii) After the Console process gets completed, the uploaded data appears under the 'Result' tab.

ix) Follow the below-given steps to display the Result view:

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

b. Click the '**Result**' tab.



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

9. The first row provided in the Excel file should contain the column headers.
10. The second row of the Excel file should contain the data under all the headers without any 'null' or 'NA.'
11. Excel headers should not have space. It should be a single word or two words concatenated by an underscore (_).
12. Excel headers should not contain any special characters. E.g. - %, #, $, @,*, etc.
13. Excel headers should not contain single or double quotes, dots, brackets, and high-fen.
14. Excel headers should not contain merely numbers. Numerals should be used with at least one alphabet.
15. Excel header should not exceed 50 characters.
16. All rows in a column should have the same data type.

Note:

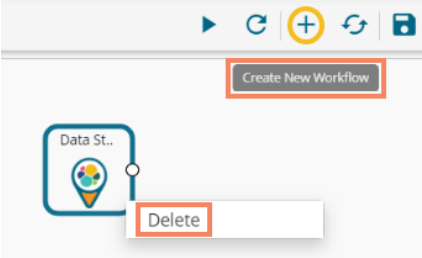
a. The Excel File component supports the .xlsx file type.

5.9. Removing a Data Source from the Workspace

i) Right-click on the data source connector (in the workspace).
ii) A context menu appears.
iii) Click the **'Delete'** option.
iv) The selected Data Source component gets removed from the workspace.

OR

Click the **'Create New Workflow'** icon to remove the connector(s) from the workspace.



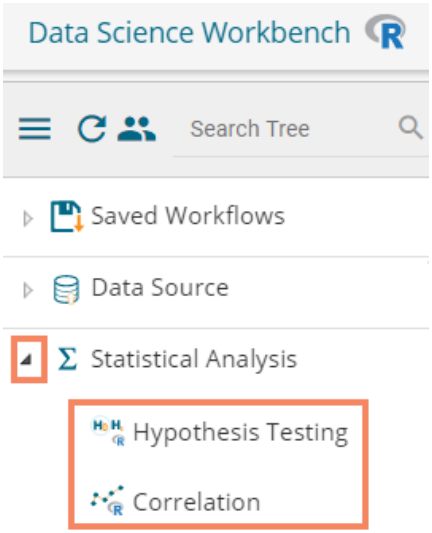
Note: The same set of steps applies to remove all types of data source connectors.

6. Statistical Analysis

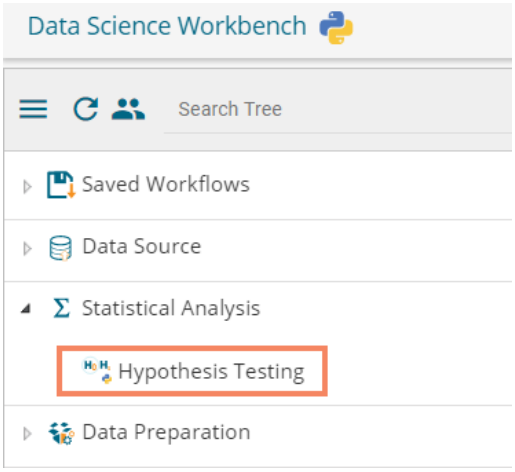
Statistical inference makes propositions about a population, using data drawn from the population with some form of sampling. Given a hypothesis about a population, for which the user wishes to draw inferences, statistical inference consists of two things, first selecting a statistical model of the process that generates the data and second deducing propositions from the model.

The R workspace provides two Statistical Analysis options as described below:

1) Hypothesis Testing
2) Correlation



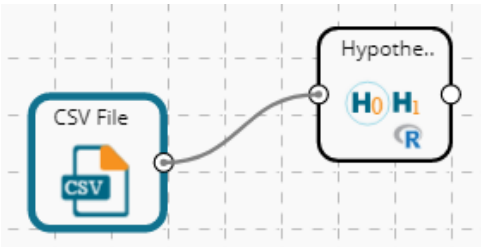
The Python Workspace provides Hypothesis Testing as a Statistical Analytics option.



6.1. Hypothesis Testing

A statistical hypothesis test is a method of statistical inference. Commonly, two statistical data sets are compared, or a data set obtained by sampling is compared against a synthetic data set from an idealized model. A hypothesis is proposed for the statistical relationship between the two data sets, and this is compared as an alternative to an idealized null hypothesis that proposes no relationship between two data sets.

i) Drag the Hypothesis Testing component to the workspace and connect it to a configured data source.



ii) Click the Hypothesis Testing component to open the configuration fields.

iii) The user needs to configure various properties fields based on the Hypothesis Testing component. The following are some possibilities of the various Properties fields when a specific **method** has been selected to perform Statistical Analysis:

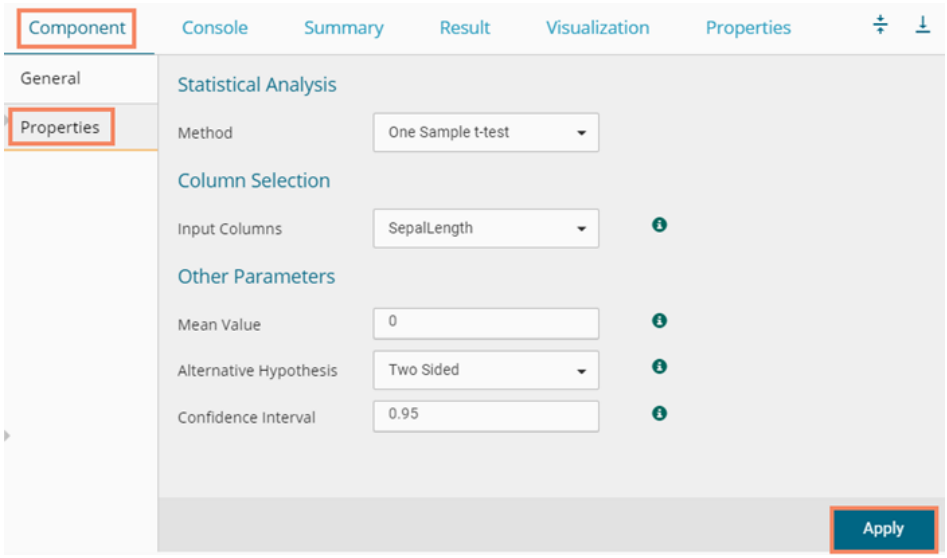
a. One Sample t-test: The one-sample t-test compares the mean of sample data to a known value. For example, one may want to know how sample means get compared to the population means. For this, one should run a one-sample t-test.

i. **Statistical Analysis**:
   1. **Method**: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'One Sample t-test')

ii. Column Selection:
   1. **Input Columns**: Select any one column from the drop-down menu (it lists only Numeric Column)

iii. **Other Parameters**
   1. **Mean Value**: Pass any integer/ decimal value. The default value for this field is 0.
   2. **Alternative Hypothesis**: select any one option from the drop-down menu (provided choices for this field are- Two-Sided, Greater, Lesser)

3. **Confidence Interval**: the textbox takes a single number between 0 and 1 (the default value for this field is 0.95)

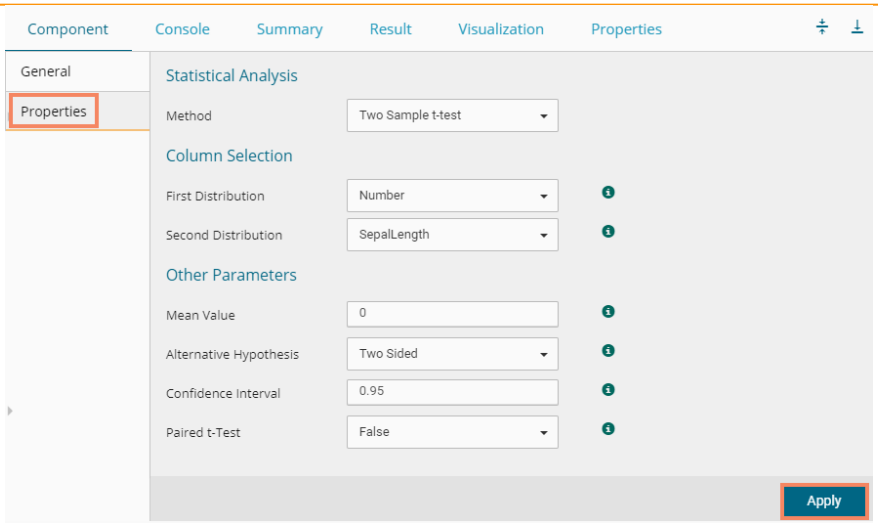


b. Two Sample t-test: A two-sample t-test is used to test the difference between two population means. A typical application is to determine whether the means are equal.

i. **Statistical Analysis**:
   1. **Method**: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'Two Sample t-test')

ii. **Column Selection**:
   1. First Distribution: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)
   2. Second Distribution: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)

iii. **Other Parameters**
   1. Mean Value: Pass any integer/ decimal value. The default value for this field is 0.
   2. Alternative Hypothesis: select any one option from the drop-down menu (provided choices for this field are- Two-Sided, Greater, Lesser)
   3. Confidence Interval: the textbox takes a single number between 0 and 1 (the default value for this field is 0.95)
   4. Paired t-Test: It has two values: True and False (The default value is False)



c. Chi-Square Test:

A Chi-Square Test is used to determine whether there is a significant association between the two variables.

i. Statistical Analysis

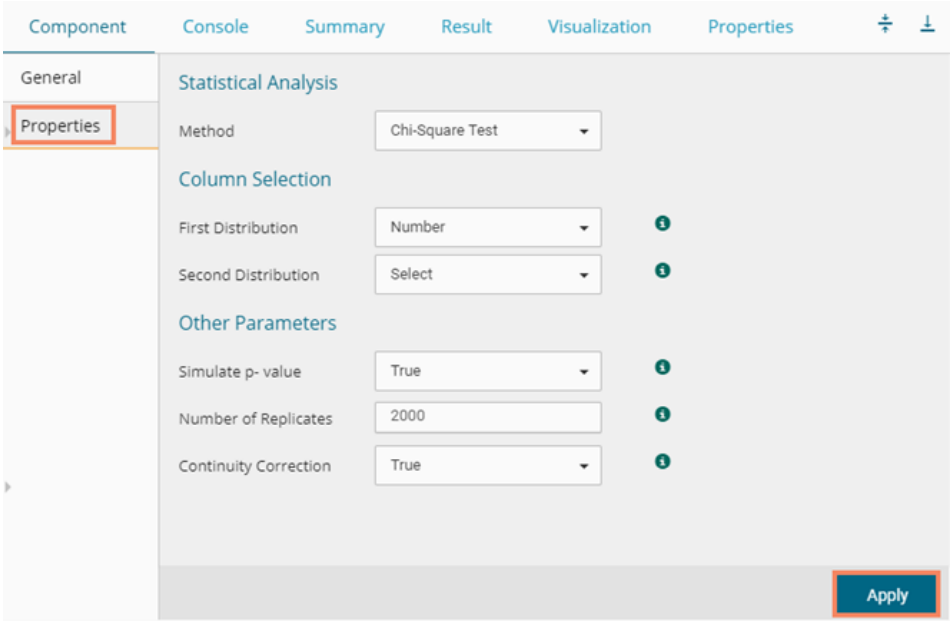
1. Method: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'Chi-Square Test')

ii. Column Selection:

1. First Distribution: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)
2. Second Distribution: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)

iii. Other Parameters

1. Simulate p-Value: It has two values: True and False (The default value for this field is True)
2. Number of Replicates: It takes positive integers (The default value for this field is 2000)
3. Continuity Correction: It has two values- True and False (The default value for this field is True)



d. **One-Way ANOVA**: There are many situations where the user may want to compare the mean between multiple groups. The ANOVA test can tell if the groups have similar performances. One-way ANOVA takes one target variable and one independent variable at a time.

i. **Statistical Analysis**

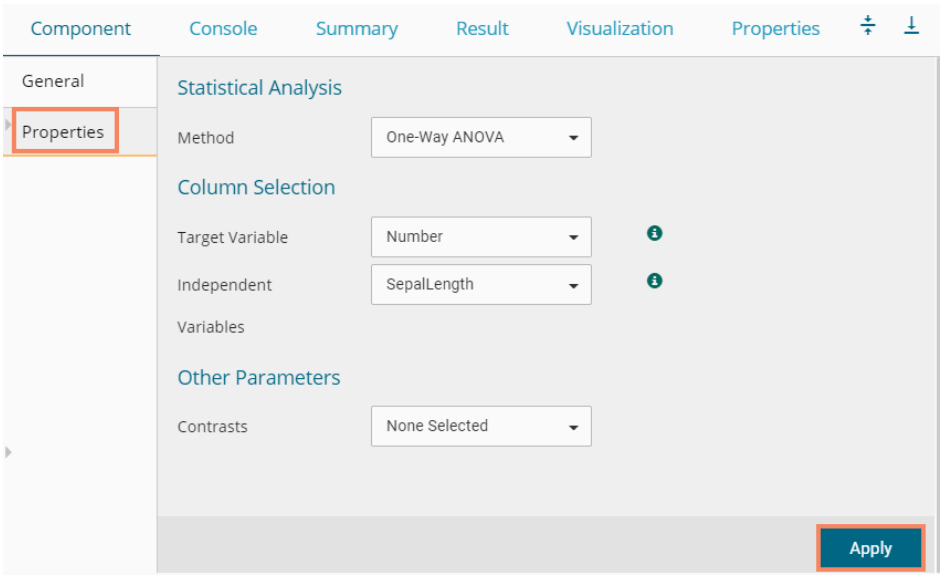
1. **Method**: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'One Way ANOVA')

ii. **Column Selection:**

1. **Target Variable**: Select any one column from the drop-down menu (it lists only Numeric Columns)
2. **Independent Variables**: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)

iii. **Other Parameters**

1. Contrasts: Select an option from the given choices to display a list of contrast items that can be used for some variables in the model. (the provided options are contr. treatment, contr. poly, contr. sum, contr. Helmert)



e. **Two-Way ANOVA**: There are many situations where the user might want to compare the mean between multiple groups. The ANOVA test can tell if the groups have similar performances. Two-way ANOVA takes one target variable and multiple independent columns at a time.

i. **Statistical Analysis**

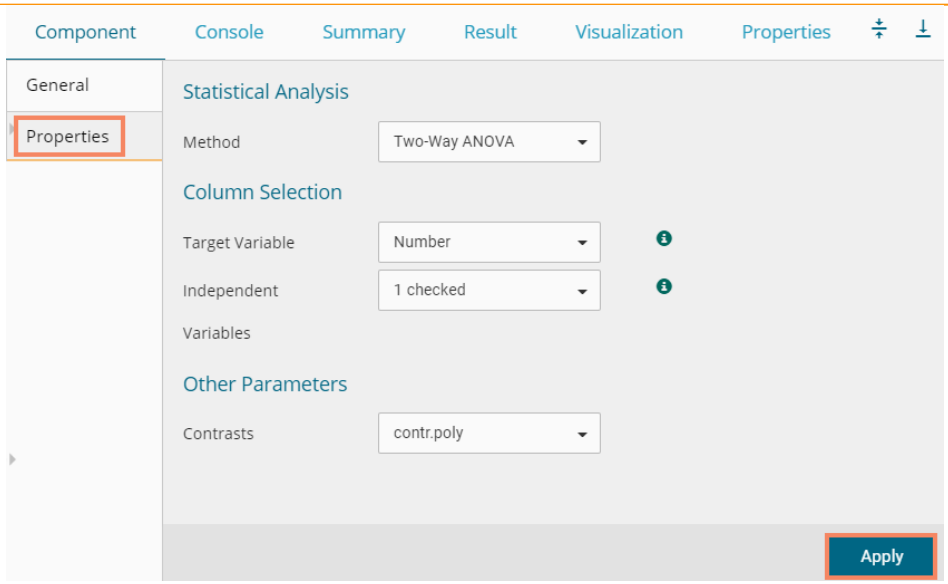
1. **Method**: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'Two-Way ANOVA')

ii. **Column Selection:**

1. **Target Variable**: Select any one column from the drop-down menu (it lists only Numeric Columns)
2. **Independent Variables**: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)

iii. **Other Parameters**

1. **Contrasts:** Select an option from the given choices to display a list of contrast items that can be used for some variables in the model. (the provided options are contr. treatment, contr. poly, contr. sum, contr. Helmert)



f. Z-test: Z-test is a statistical test where normal distribution is applied and is used for dealing with problems relating to large samples when n ≥ 30.

i. **Statistical Analysis**:

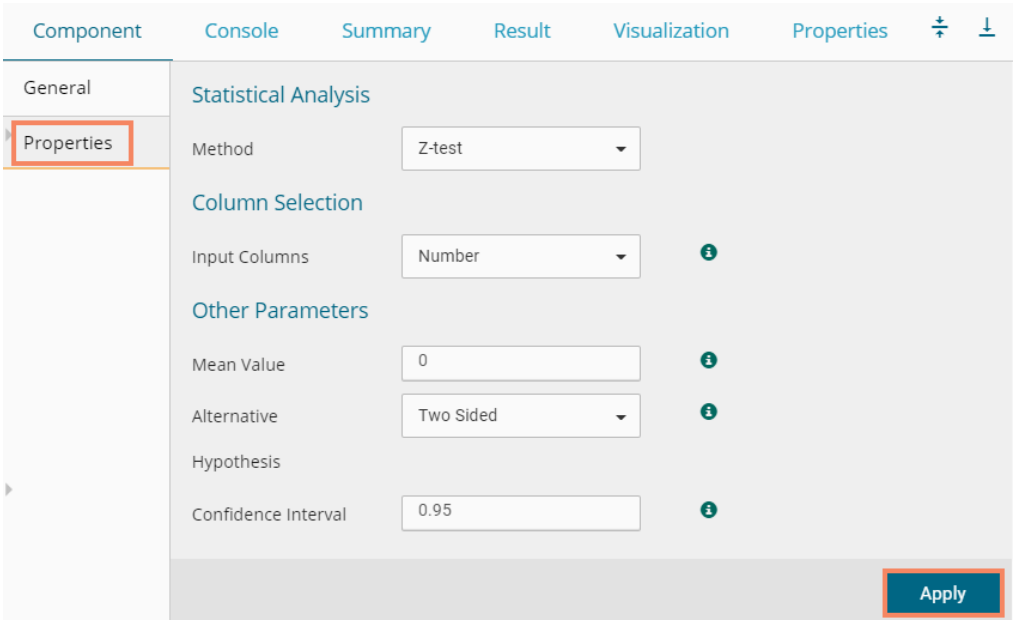
1. **Method**: Select an option from the drop-down menu. Other properties fields get displayed based on the selection of the Method option. (In this case, the selected method is 'Z-test')

ii. Column Selection:

1. **Input Columns**: Select any one column from the drop-down menu (it lists only Numeric Column)

iii. Other Parameters

1. **Mean Value**: Pass any integer/ decimal value. The default value for this field is 0.
2. **Alternative Hypothesis**: select any one option from the drop-down menu (provided choices for this field are- Two-Sided, Greater, Lesser)
3. **Confidence Interval**: The textbox takes a single number between 0 and 1 (the default value for this field is 0.95)



g. **Fisher Test**: Fisher's exact test is a statistical test used to determine if there are non-random associations between two categorical variables.

i. **Statistical Analysis**

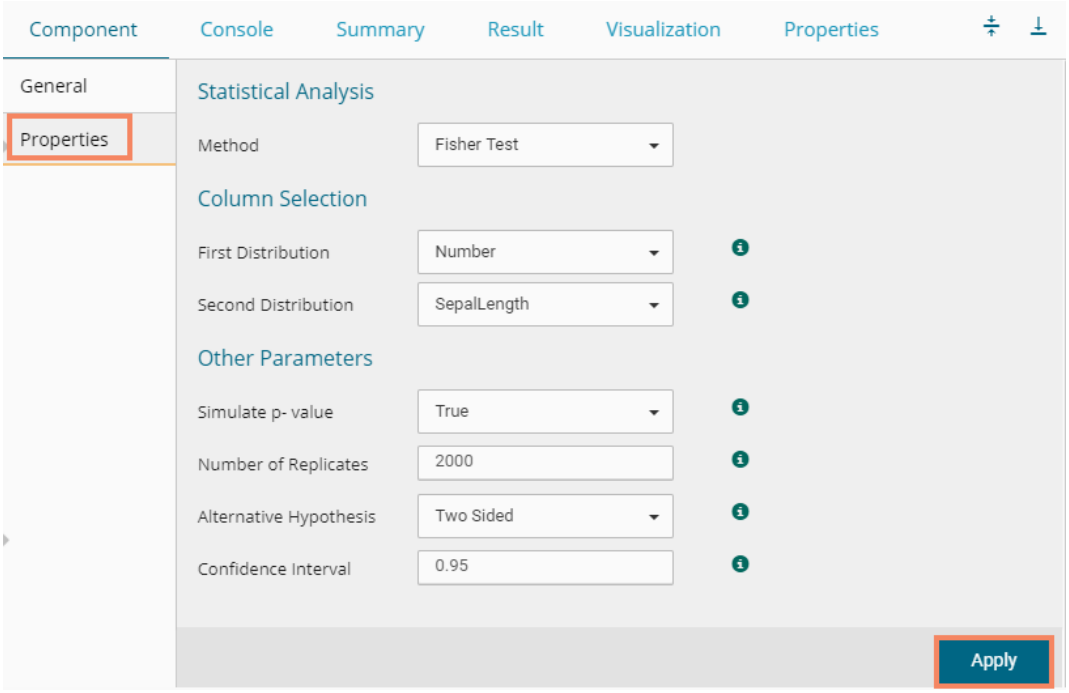
1. **Method**: Select an option from the drop-down menu. (In this case, the selected method is '**Fisher Test'**)

ii. **Column Selection:**

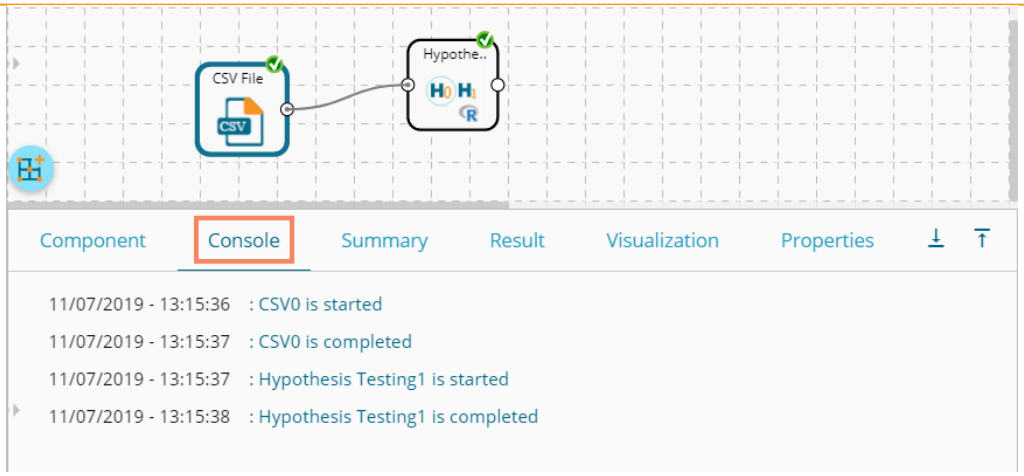
1. **First Distribution**: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)
2. **Second Distribution**: Select any one column from the drop-down menu (it lists only Numeric and Factor Columns)

iii. **Other Parameters**

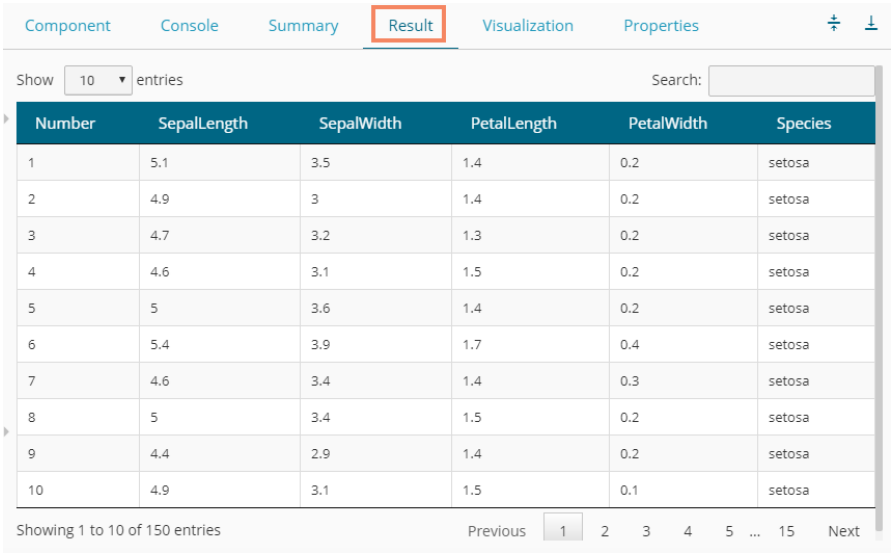
1. **Simulate p-Value**: It has two values: True and False (The default value for this field is True)
2. **Number of Replicates**: It takes positive integers (The default value for this field is 2000)
3. **Alternative Hypothesis**: select any one option from the drop-down menu (provided choices for this field are- Two-Sided, Greater, Lesser)
4. **Confidence Interval**: The textbox takes a single number between 0 and 1 (the default value for this field is 0.95)



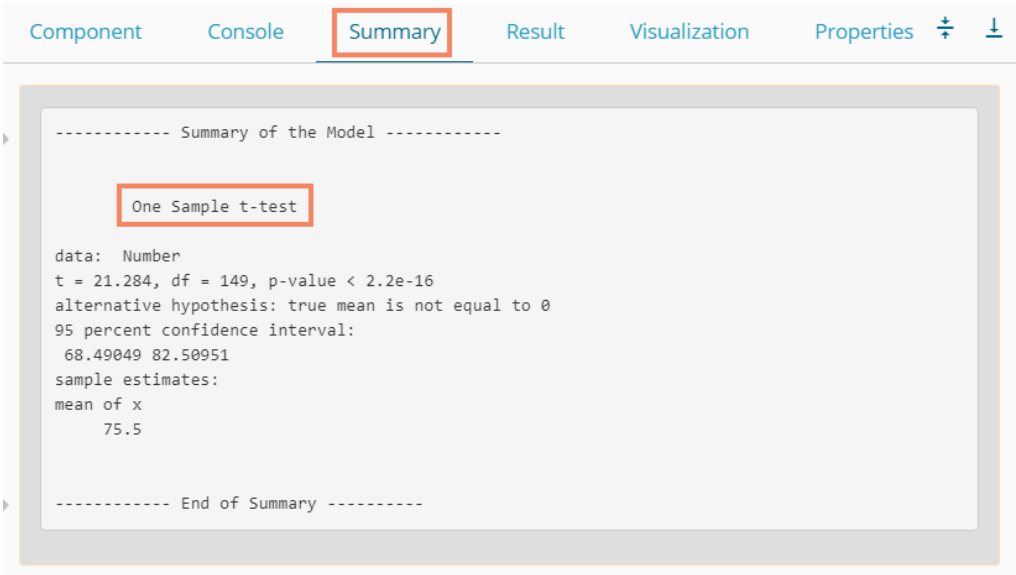
iv) After a successful configuration, runs the workflow.
v) The 'Console' tab opens, displaying the progress of the process.
vi) The success of the process gets indicated through the green marks on the components.



vii) Click the '**Result**' tab to see the Result view of the data.



viii) Click the '**Summary'** tab to see the summary of the Hypothesis Test.

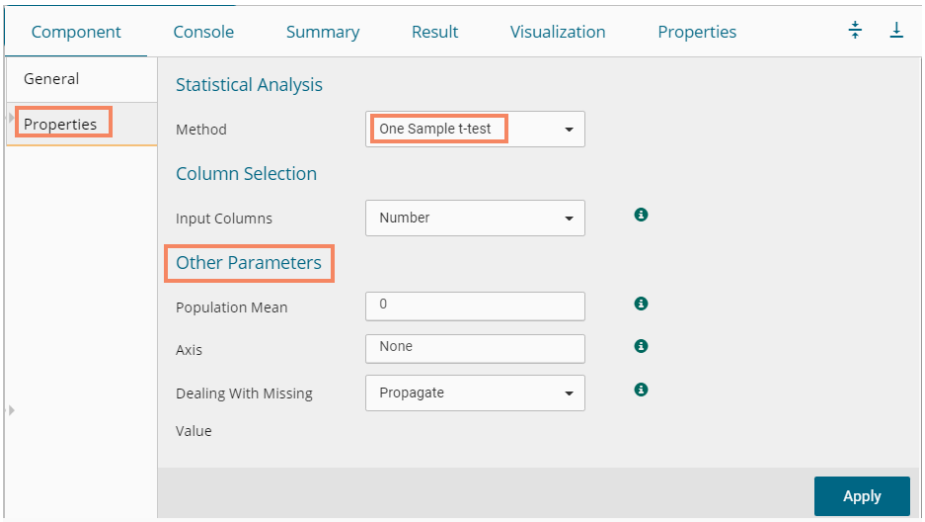


Note:

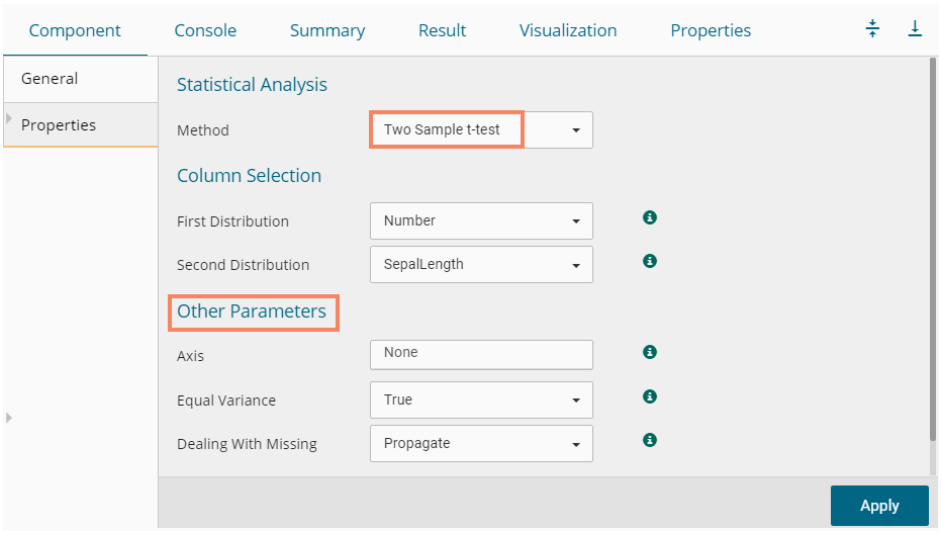
a. Other properties fields get displayed based on the selection of the '**Method**' option.

b. The Hypothesis Testing provided under the Python Workspace contains the same steps of configuration, but the Other Parameters fields vary as per the selected testing method. Please find all the Other Parameters variations provided below based on a specific testing method. Click the '**Information**' icon to get the details of these fields.

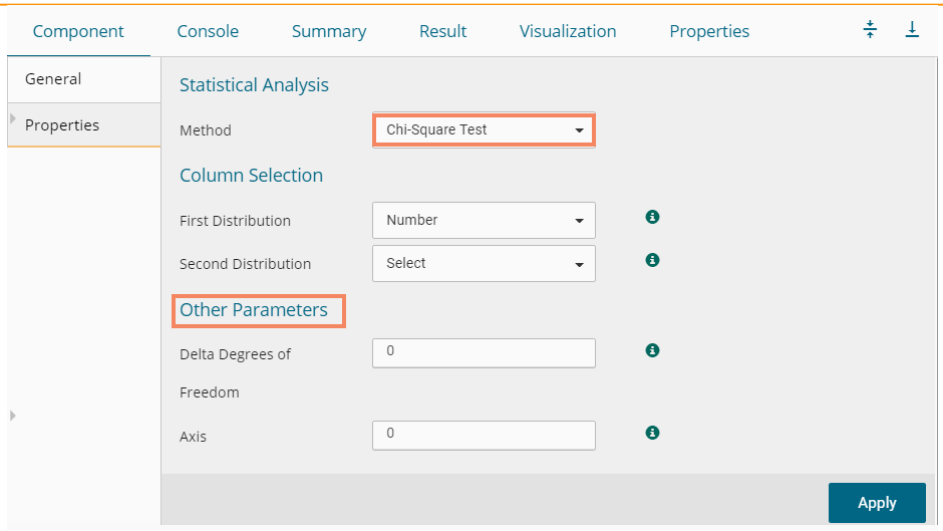
i. One Sample t-test



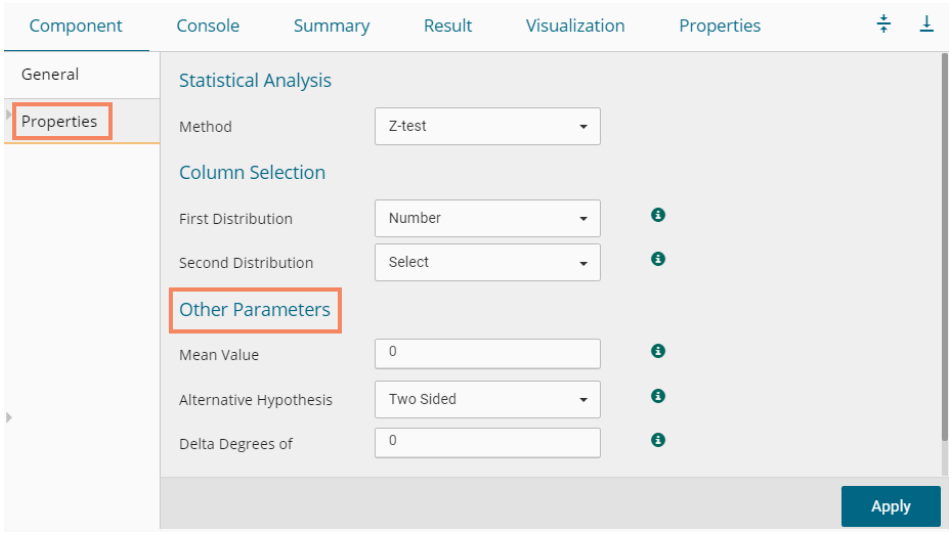
ii. Two Sample t-test



iii. Chi-Square Test



iv. Z-test

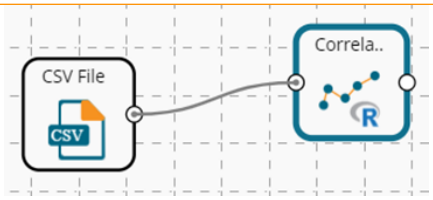


There are no Other Parameters fields provided for the methods One-Way ANOVA and Two-Way ANOVA.

6.2. Correlation

Correlation is a statistical inference method that measures the degree to which two or more variables fluctuate together. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases.

i) Drag the Correlation component to the workspace and connect it to a configured data source.
ii) Click the Correlation component to open the configuration fields.



iii) Configure the following properties fields for the Correlation component:

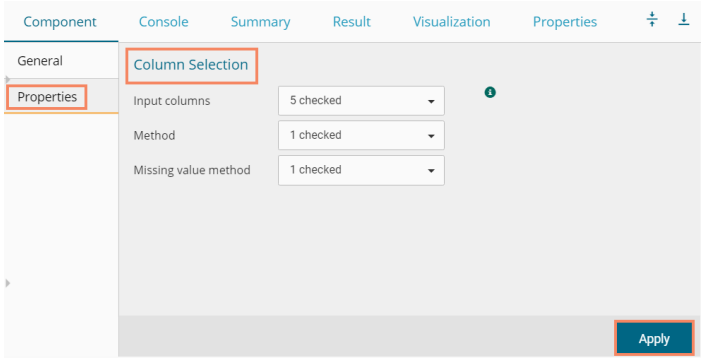
a. Input Columns: Select any two columns using the drop-down menu

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:

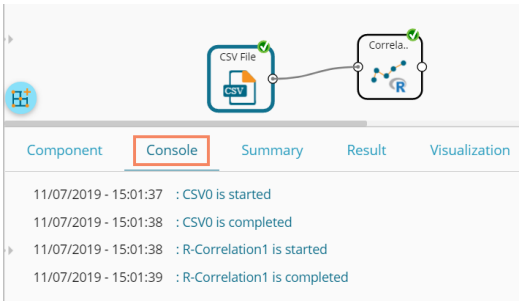
i. Everything
ii. All. obs
iii. Complete. obs
iv. Na.or. complete
v. Pairwise.complete.obs



a. Click the '**Apply'** option.

iv) Run the workflow.

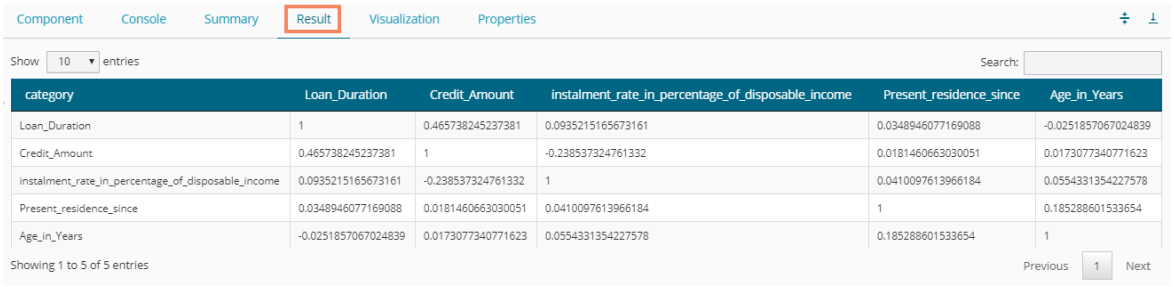
v) The progress of the process gets displayed in the '**Console**' tab.



vi) Follow the below-given steps to display the Result view:

a. Click the dragged correlation component onto the workspace.

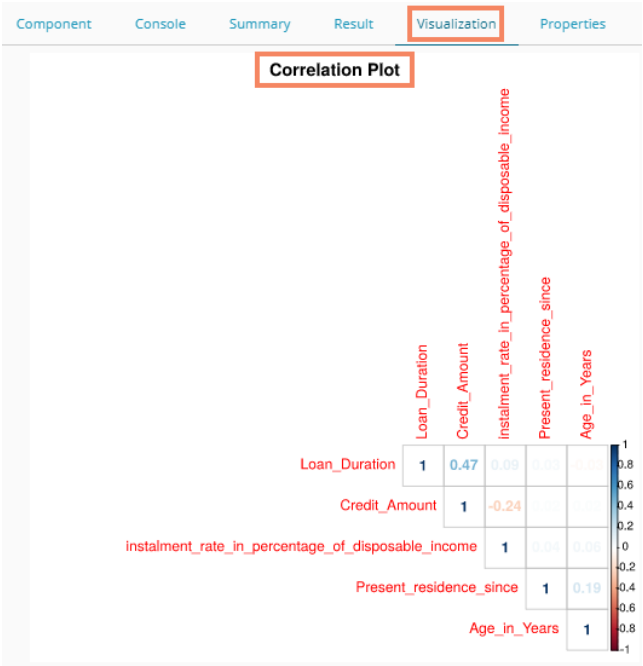
b. Click the '**Result**' tab.



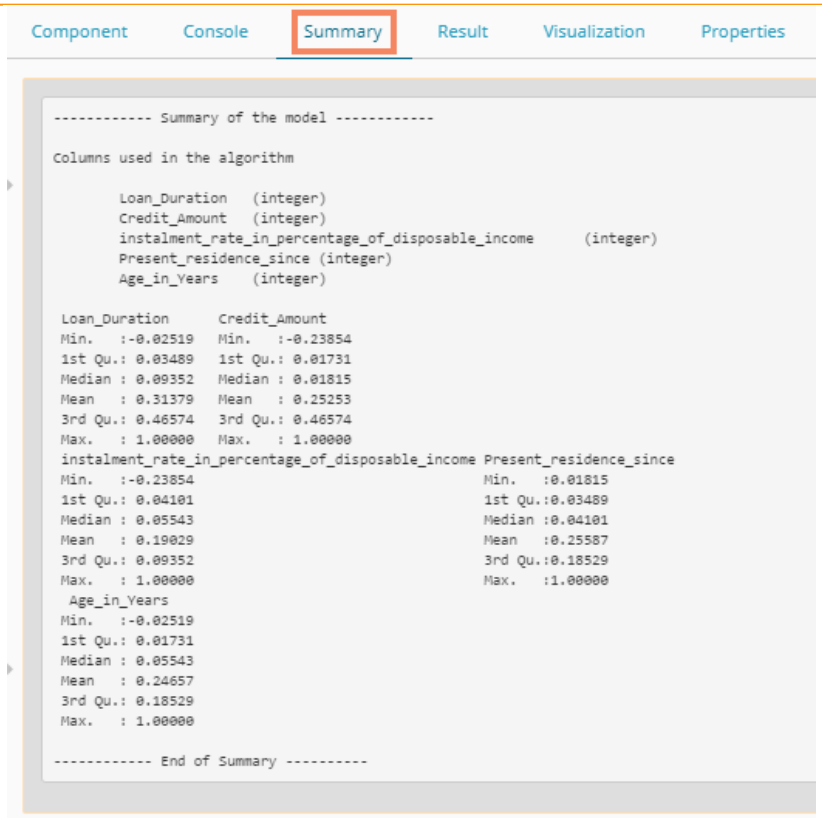
Note: The selected dataset has more columns than displayed in the below-given Result view.

vii) Click the '**Visualization'** tab.

viii) The probable values of the selected columns get displayed via the Correlation Plot.



ix) Click the '**Summary'** tab to view the model summary.



Note: The displayed Result, Visualization, and Summary tabs are based on the selection of the Kendall method. The user may have a slight variation based on another selection.

7. Data Preparation

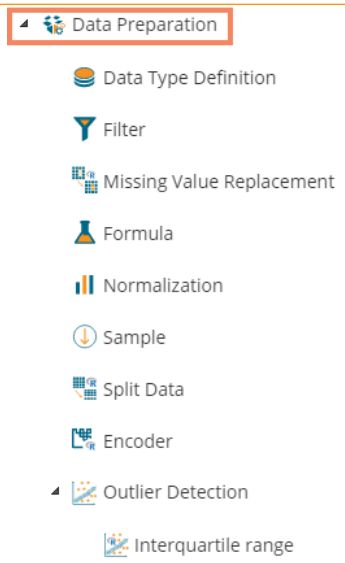
Components provided under the **Data Preparation** tree-node help in preparing the raw data from the data source and make it suitable for analysis. They organize data to gain accurate Results out of it. The list of the Data Preparation components may vary based on the different Workspace, but the configuration steps remain the same. This section aims at listing all the available Data Preparation components collectively.

Note: The Data Preparation list may vary based on various Data Science Workspaces, but the configuration process remains the same for all.

7.1. Data Type Definition

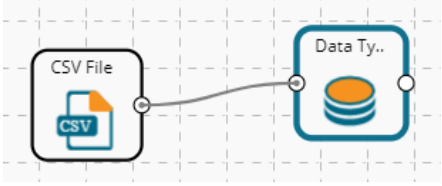
The Data Type Definition option can be used to change the name, data type of the data source column. This component helps users to prepare data and make it suitable for further analysis.

i) Navigate to the landing page of any Data Science Workspace.
ii) Click the '**Data Preparation**' tree node.
iii) Various data preparation options get displayed (The below-given list displays the Data Preparation options provided under the R Workspace since it includes all the available Data Preparation components).



iv) Drag the '**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).

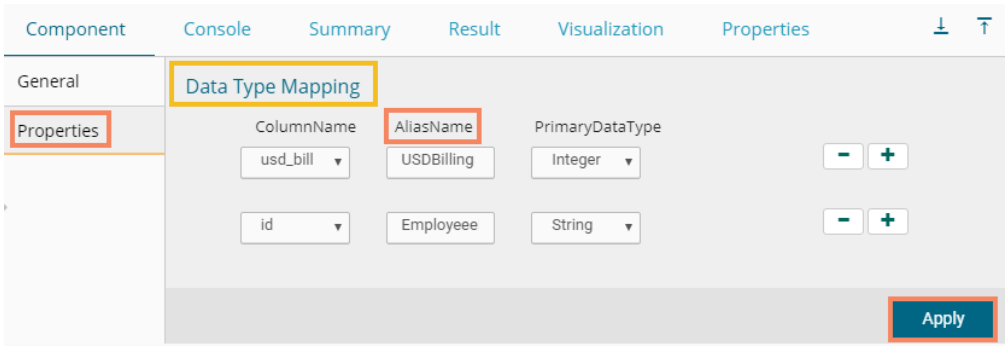


vi) The '**Properties**' tab opens.

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

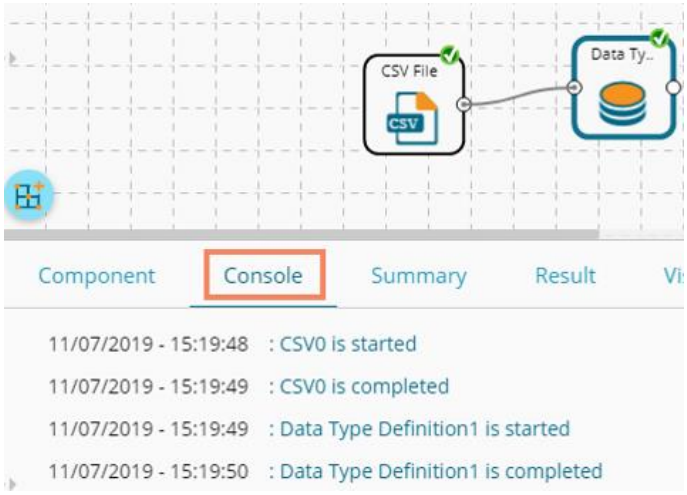
a. **Column Name:** Select a column name that you want to change
b. **Alias Name:** Enter an alias name for the required source column
c. **Primary Data Type:** Select a primary data type column that you want to change
d. **Date Format:** Select a date format that you want to display (the Date format is optional for date Data Type)
e. '**Add**' **option** **:** Click on this icon to add one more row of the **'Data Type Mapping'** fields

viii) Click the '**Apply**' option.



ix) **Run** the workflow by clearing the previous Cache.

x) Open the 'Console' tab to see the progress of the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.



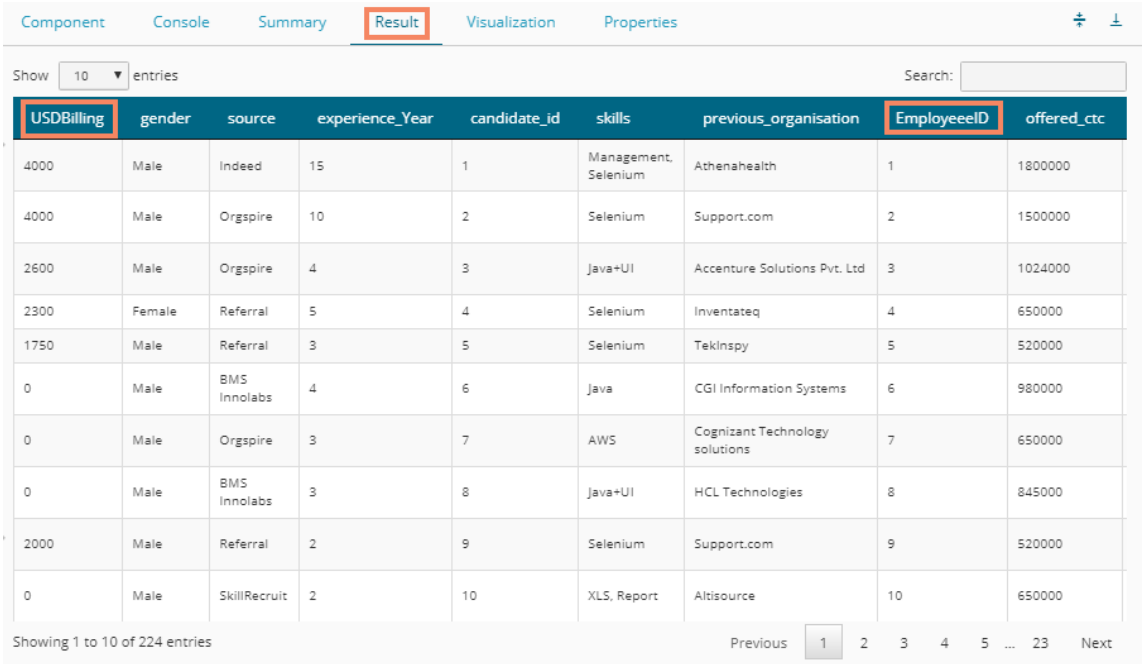
xi) After the Console process gets completed, users can view the Result data using the 'Result' tab

xii) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.

xiii) The user can see the given column names on the selected columns in the displayed **Result** data.

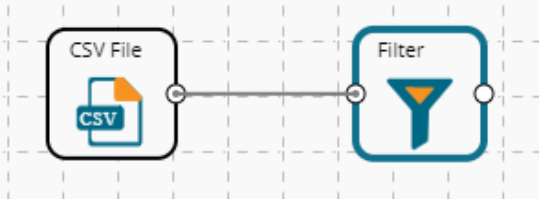


7.2. Filter

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

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

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

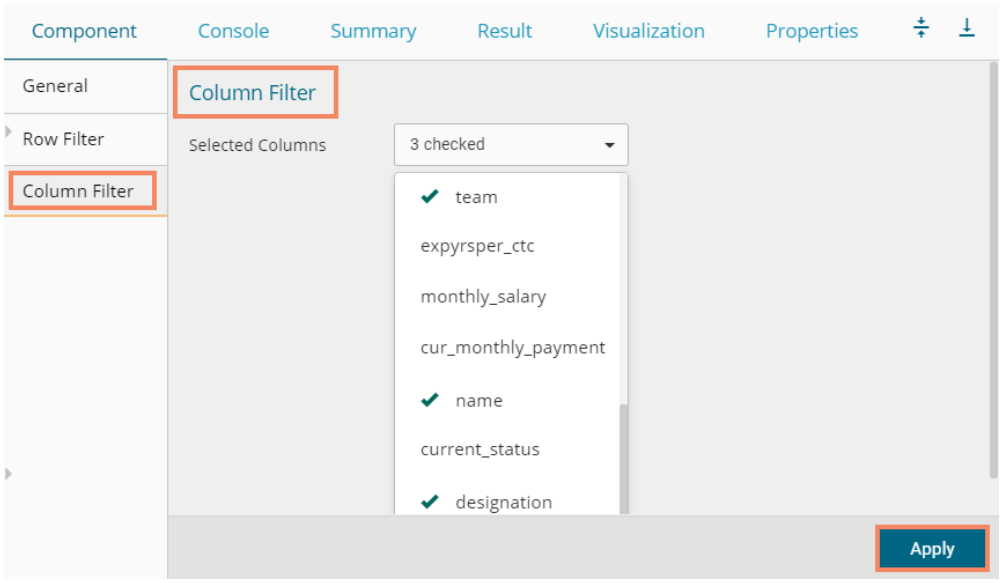


iii) Configure the filter component as described below:

7.2.1. Column Filter

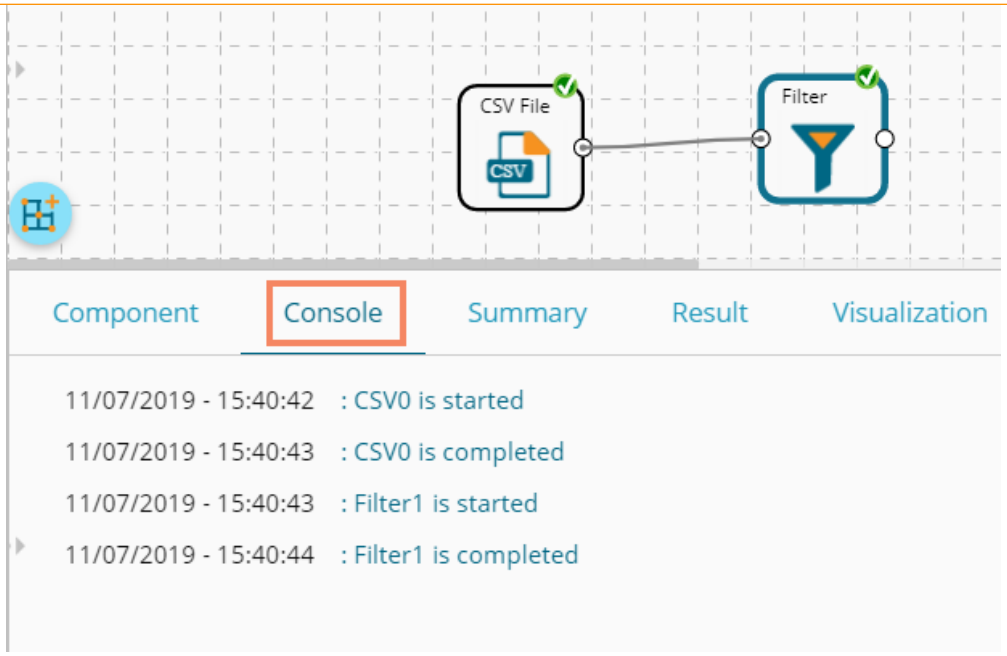
i) Select a column from the **'Selected Columns'** context menu.

ii) Click the **'Apply'** option to configure the data.



iii) Run the workflow by clearing the previous cache.

iv) The 'Console' tab opens to display the progress of the process. The completion of the Console process gets marked by green checkmarks on the top of the dragged components.



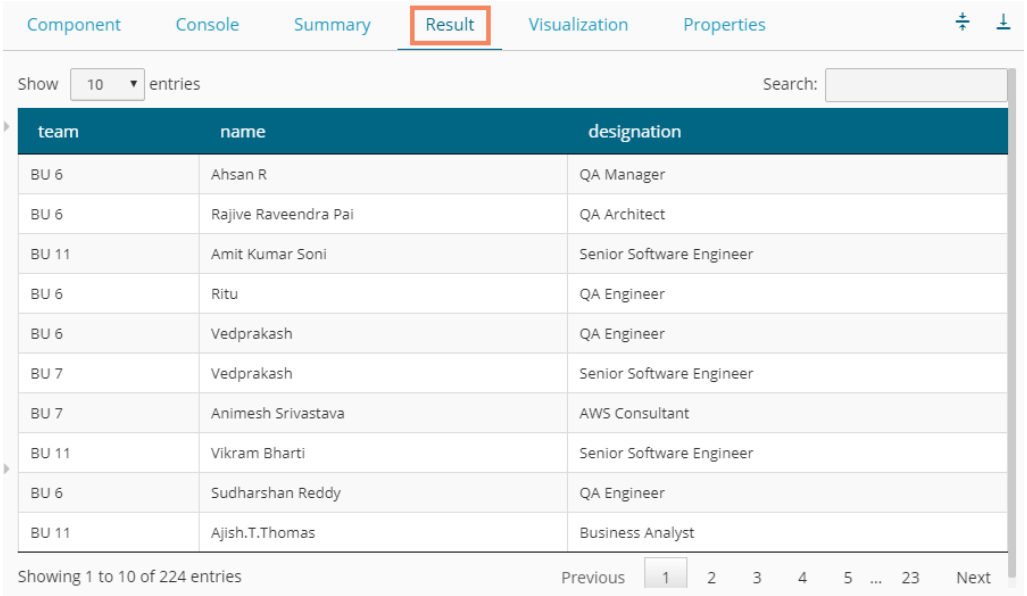
v) After the Console process gets completed, users can view the Result data using the 'Result' tab.

vi) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component in the workspace.

b. Click the 'Result' tab.

vii) The filtered data gets displayed via the 'Result' tab.



7.2.2. Row Filter

i) Drag the Filter Component to the workspace and connect it to a configured data source.

ii) Click the **'Filter'** component.

iii) The '**Column Filter'** tab gets displayed (by default).

iv) Select a column using the context menu.

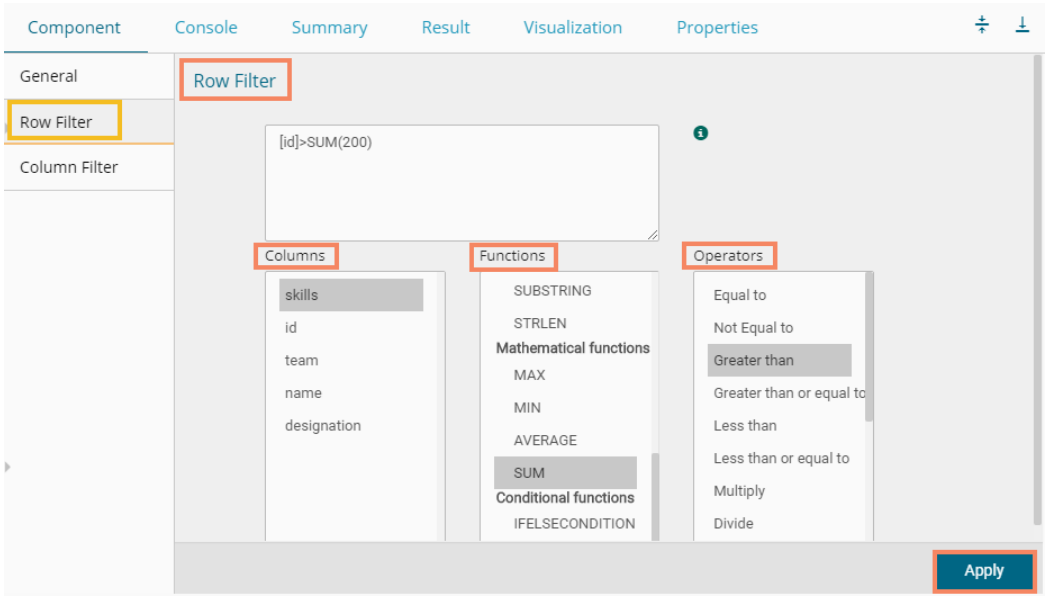
v) Select the '**Row Filter**' tab from the '**Component**' menu list.

vi) Configure the required fields:

a. Double click on the components from **Columns, Operators,** and **Functions** in the sequence as shown in the image below

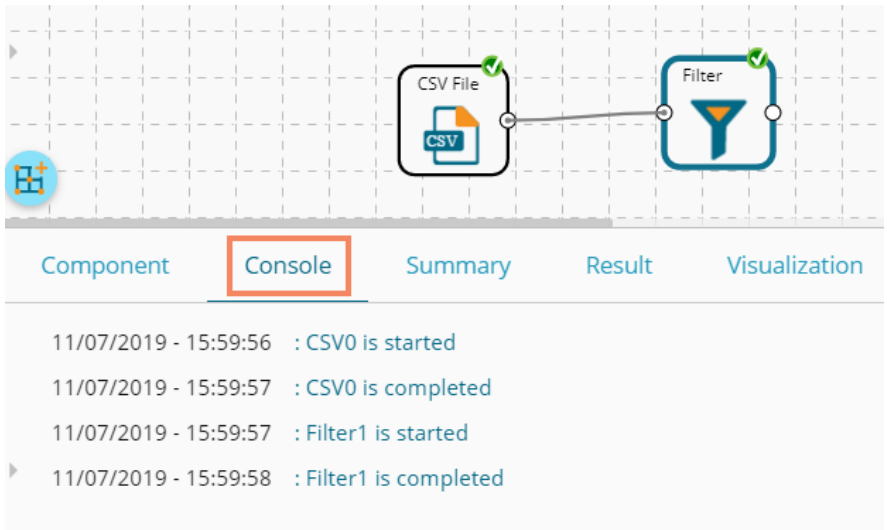
b. A formula gets entered in the given box (E.g., in this case, the entered formula is [id]>SELECT(200)

c. Click the 'Apply' option.



vii) Run the workflow by clearing the previous cache.

viii) The 'Console' tab opens to display the progress of the process. The completion of the Console process is marked by the green tick marks on the top of the dragged components.



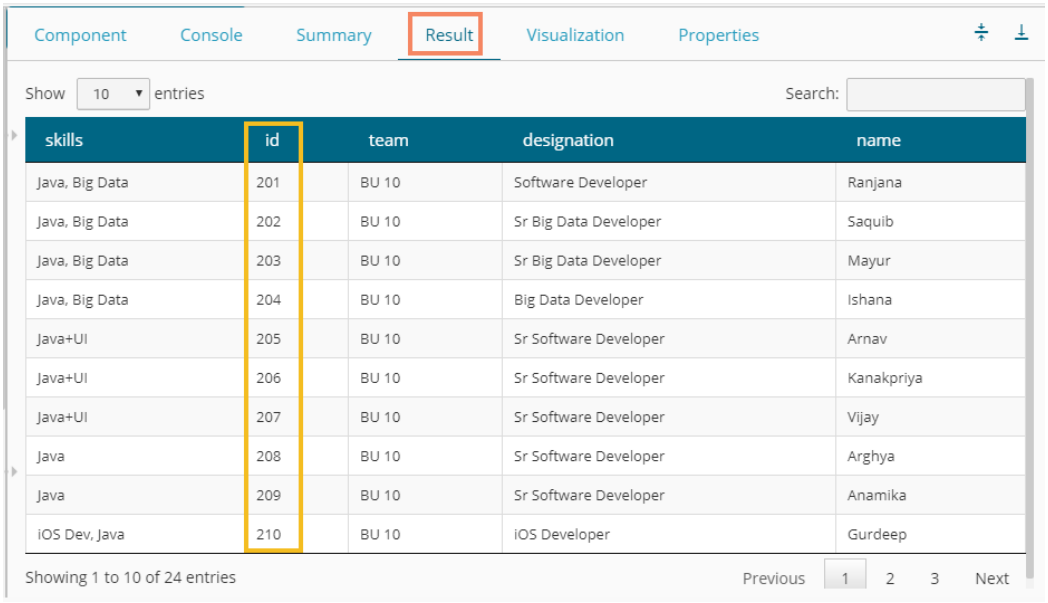
ix) After the Console process gets completed, users can view the Result data using the 'Result' tab

x) Follow the below-given steps to display the Result view:

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

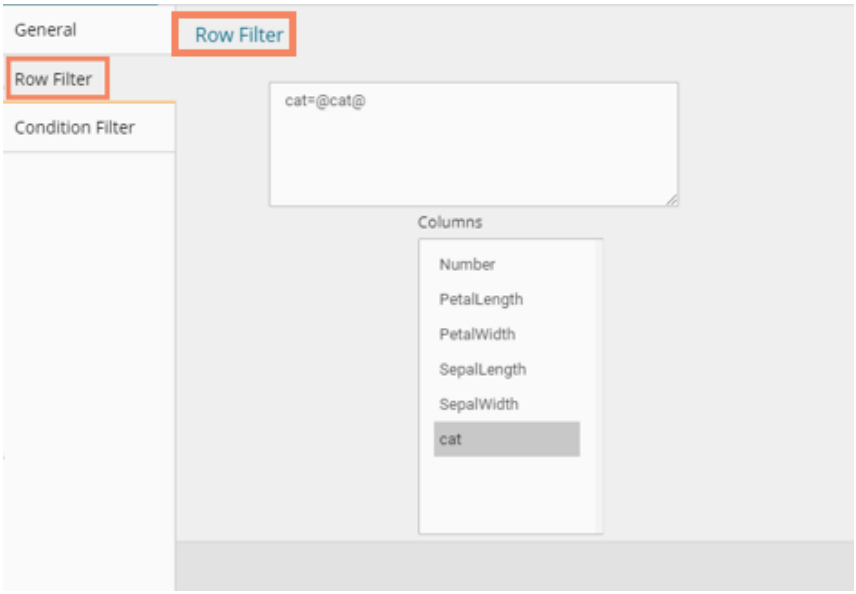
b. Click the 'Result' tab.

xi) The filtered data, as per the applied formula, gets displayed under the '**Result**' tab.



Note:

a. The expression should retain Boolean output.
b. Users can not use Data manipulation functions.
c. The Row Filter functionality provided under the Spark workspace takes the specific column name in between the @ symbols.
E.g., @cat@ as displayed below.

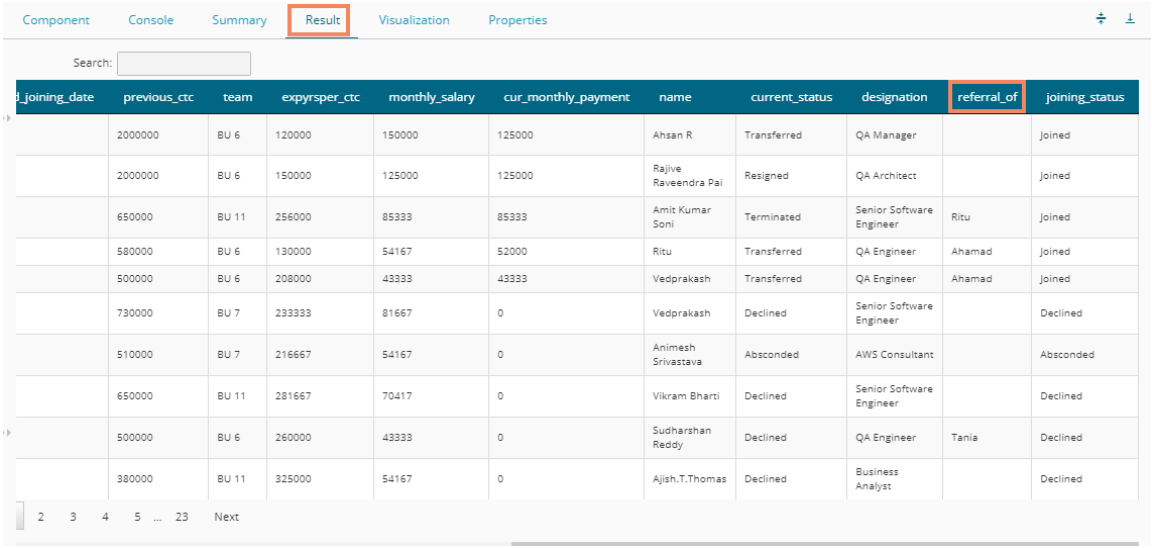


7.3. Missing Value Replacement

Users can replace the missing data in the specified variable with the determined value. The user is

provided with a list of options that can be considered for replacement.

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



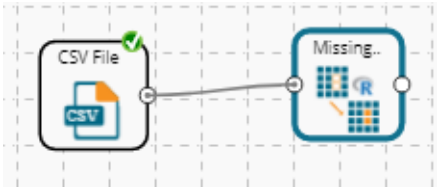
d_joining_date	previous_ctc	team	expyrsper_ctc	monthly_salary	cur_monthly_payment	name	current_status	designation	referral_of	joining_status
	2000000	BU 6	120000	150000	125000	Ahsan R	Transferred	QA Manager		Joined
	2000000	BU 6	150000	125000	125000	Rajive Raveendra Pai	Resigned	QA Architect		Joined
	650000	BU 11	256000	85333	85333	Amit Kumar Soni	Terminated	Senior Software Engineer	Ritu	Joined
	580000	BU 6	130000	54167	52000	Ritu	Transferred	QA Engineer	Ahamad	Joined
	500000	BU 6	208000	43333	43333	Vedprakash	Transferred	QA Engineer	Ahamad	Joined
	730000	BU 7	233333	81667	0	Vedprakash	Declined	Senior Software Engineer		Declined
	510000	BU 7	216667	54167	0	Animesh Srivastava	Absconded	AWS Consultant		Absconded
	650000	BU 11	281667	70417	0	Vikram Bharti	Declined	Senior Software Engineer		Declined
	500000	BU 6	260000	43333	0	Sudharshan Reddy	Declined	QA Engineer	Tania	Declined
	380000	BU 11	325000	54167	0	Ajish.T.Thomas	Declined	Business Analyst		Declined

2 3 4 5 ... 23 Next

ii) Select and drag the '**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.



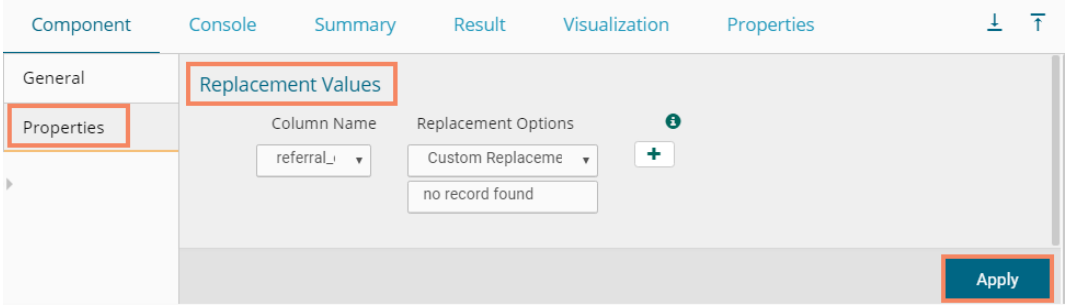
v) Choose the replacement value by configuring the following fields:

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:

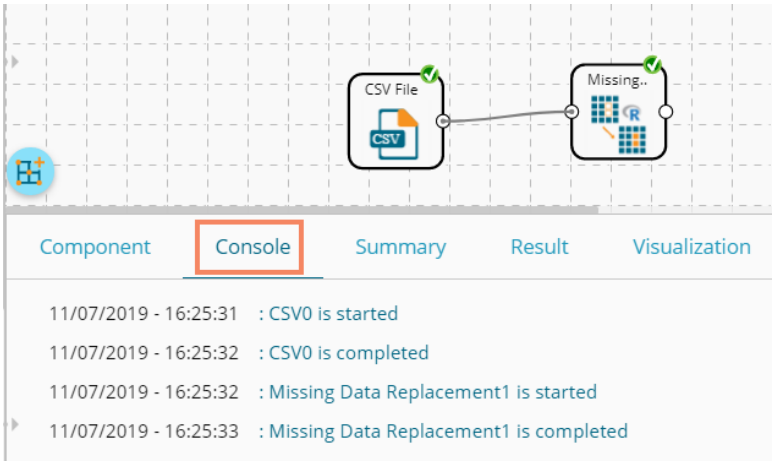
1. Mean
2. Median
3. Mode
4. Maximum
5. Minimum
6. Remove Entire Row
7. Remove Entire Column
8. Custom Replacement

vi) Click the 'Apply' option.



vii) Run the workflow by clearing the previous cache.

viii) The user can be redirected to the 'Console' tab to display the progress of the process.



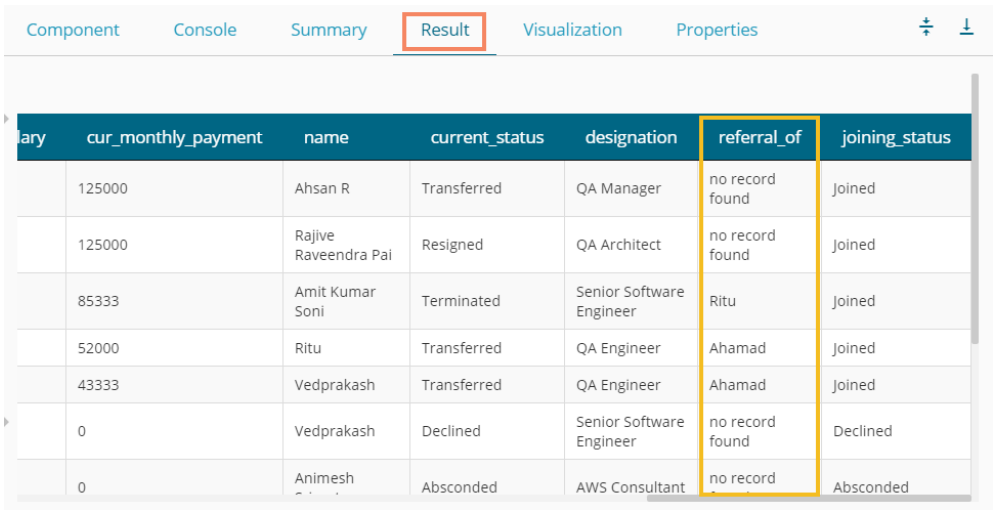
ix) After the Console process gets completed, the user can view the Result data using the 'Result' tab.

x) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.

xi) The missing values in the selected column get replaced with the selected custom replacement value.

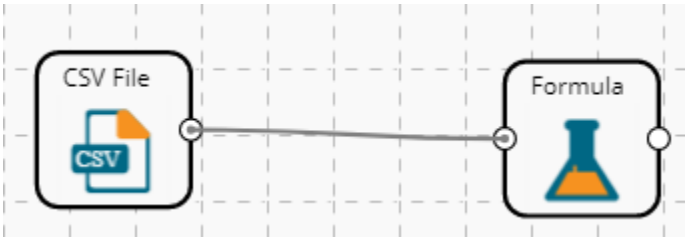


lary	cur_monthly_payment	name	current_status	designation	referral_of	joining_status
	125000	Ahsan R	Transferred	QA Manager	no record found	Joined
	125000	Rajive Raveendra Pai	Resigned	QA Architect	no record found	Joined
	85333	Amit Kumar Soni	Terminated	Senior Software Engineer	Ritu	Joined
	52000	Ritu	Transferred	QA Engineer	Ahamad	Joined
	43333	Vedprakash	Transferred	QA Engineer	Ahamad	Joined
	0	Vedprakash	Declined	Senior Software Engineer	no record found	Declined
	0	Animesh	Absconded	AWS Consultant	no record	Absconded

7.4. Formula

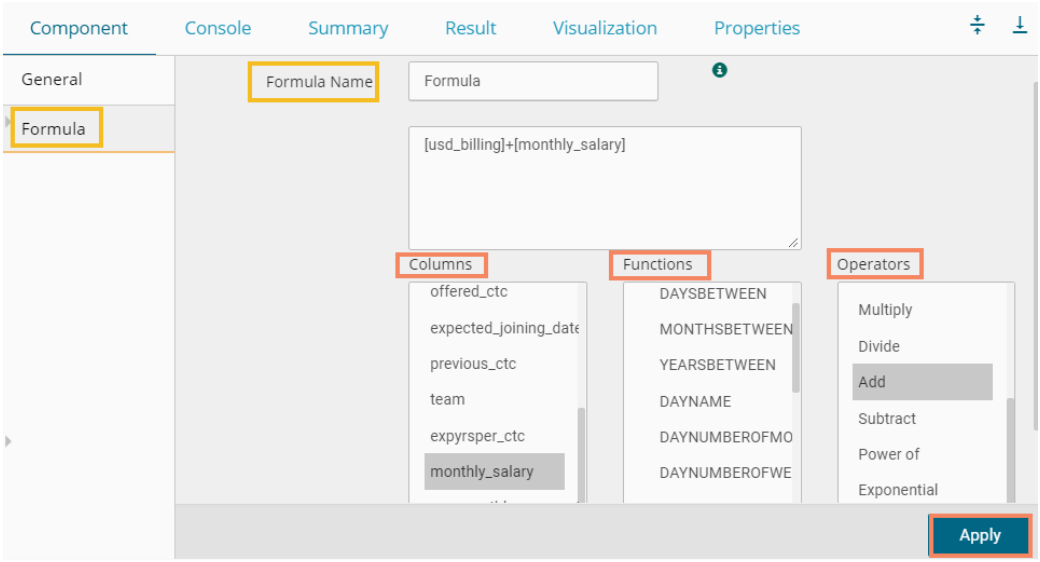
The user can create a calculated column using **'Formula.'** A formula can be formed by using available columns, functions, and operators.

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

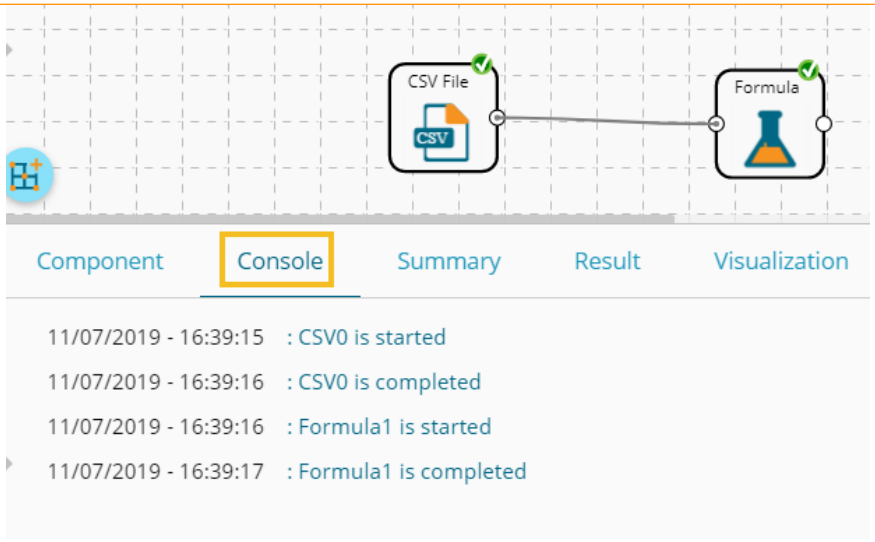


iv) Configure the required component fields to Apply a formula:

a. '**Columns,'** '**Functions,'** and **'Operators':** Double click on these lists 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) Run the workflow by clearing the previous cache.
vi) The 'Console' tab opens displaying the progress of the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.



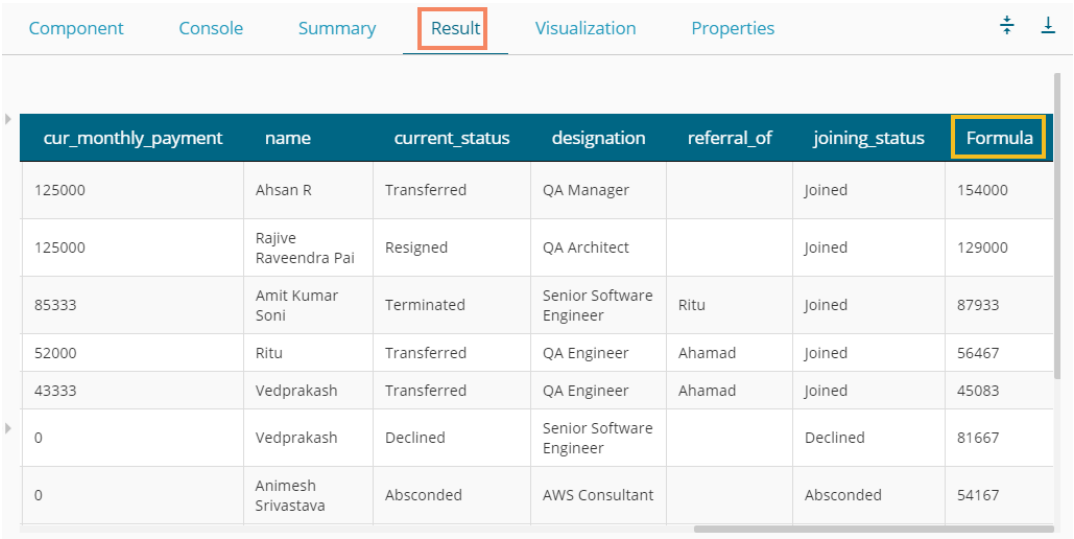
vii) After the Console process gets completed, the user can view the Result data using the 'Result' tab.

viii) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.

ix) A new column containing the data based on the inserted formula gets added to the Result data. (E.g., the '**Formula**' column as displayed below.)



cur_monthly_payment	name	current_status	designation	referral_of	joining_status	Formula
125000	Ahsan R	Transferred	QA Manager		Joined	154000
125000	Rajive Raveendra Pai	Resigned	QA Architect		Joined	129000
85333	Amit Kumar Soni	Terminated	Senior Software Engineer	Ritu	Joined	87933
52000	Ritu	Transferred	QA Engineer	Ahamad	Joined	56467
43333	Vedprakash	Transferred	QA Engineer	Ahamad	Joined	45083
0	Vedprakash	Declined	Senior Software Engineer		Declined	81667
0	Animesh Srivastava	Absconded	AWS Consultant		Absconded	54167

7.5. Normalization

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

7.5.1. Min-Max Normalization

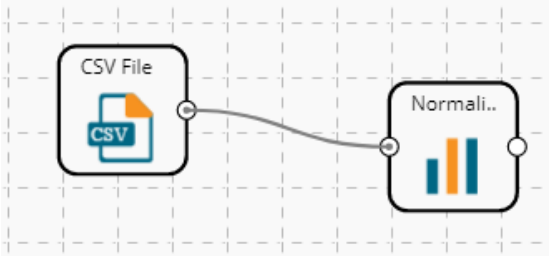
It implements a linear transformation of the original data values and sets a new range for all the data values to fit in. The user can fix the New Maximum and New Minimum Value for the data from the new field. Consequently, each value "v" from the original interval gets mapped into value "new_v" following 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 **the '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 gets selected)

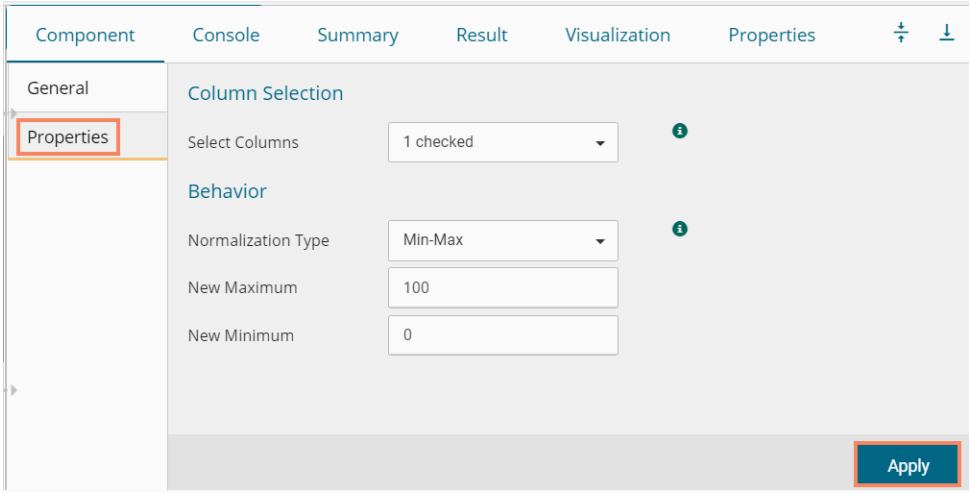
b. **Behavior**

   i. **Normalization Type:** Select '**Min-Max**' normalization type from the drop-down menu

   ii. **New Maximum:** Set a new maximum value (the Default value for this field is 1)

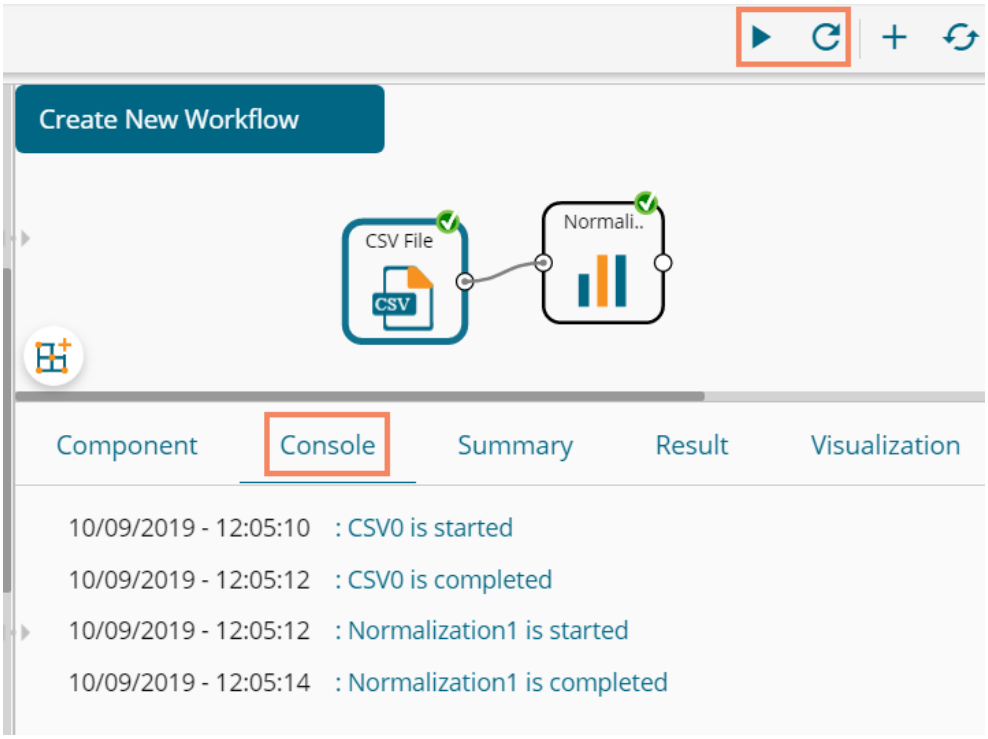
   iii. **New Minimum:** Set a new minimum value (the Default value for the New Minimum field is 0)

v) Click the 'Apply' option.



vi) Run the workflow by clearing the previous cache.

vii) The ‘Console’ tab opens displaying the progress of the process. The completion of the Console process gets marked by the green checkmarks.

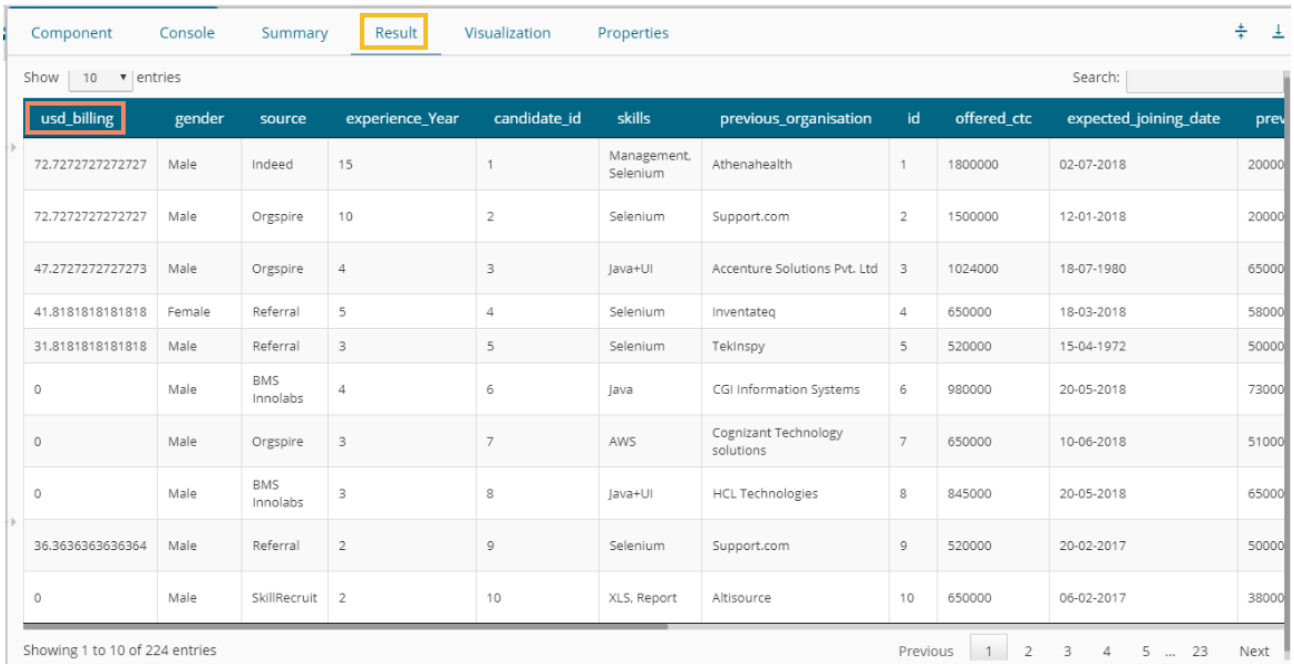


viii) After the Console process gets completed, the user can view the Result data using the ‘Result’ tab.

ix) Follow the below-given steps to display the Result view:

a. Click the dragged Formula component in the workspace.

b. Click the ‘Result’ tab.



7.5.2. Zero-Score

This normalization is known as **Zero Mean Normalization,** which is calculated on the **mean** and **standard deviation** for each attribute. It determines whether a specific value is above or below average. It also signifies the exact proportion of the variance from the fixed limit of aver3age. After Applying **'Zero-Score'** normalization, each feature has a mean value of zero (0). The unit of each value is 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:

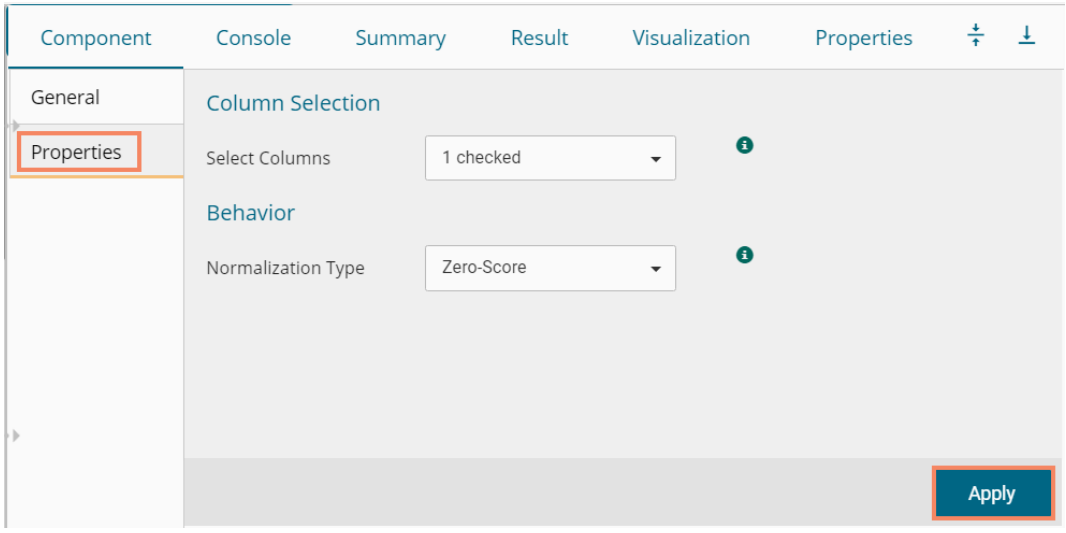
$$new\_v = \frac{v - \mu_x}{\sigma_x}$$

i) Select and drag the **'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:

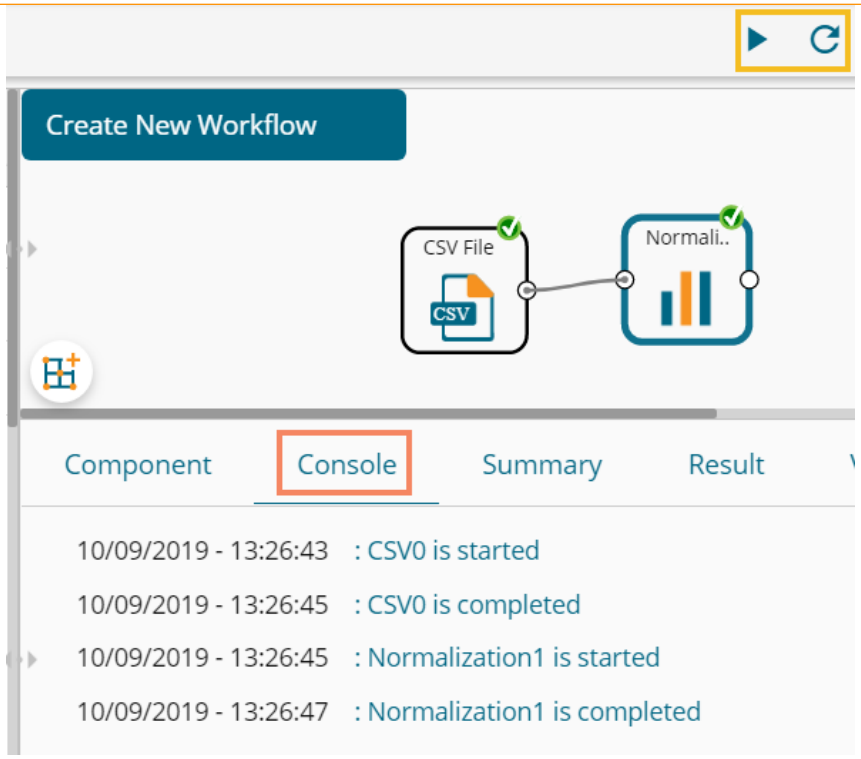
Properties

a. **Column Selection**
   i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column gets selected)
b. **Behavior**
   i. **Normalization Type:** Select '**Zero-Score**' normalization type from the drop-down menu

v) Click the **'Apply'** option.



vi) Run the workflow by clearing the previous cache.
vii) The user gets redirected to the 'Console' tab to display the progress of the process. The completion of the Console process is marked by the green checkmarks on the top of the dragged components.

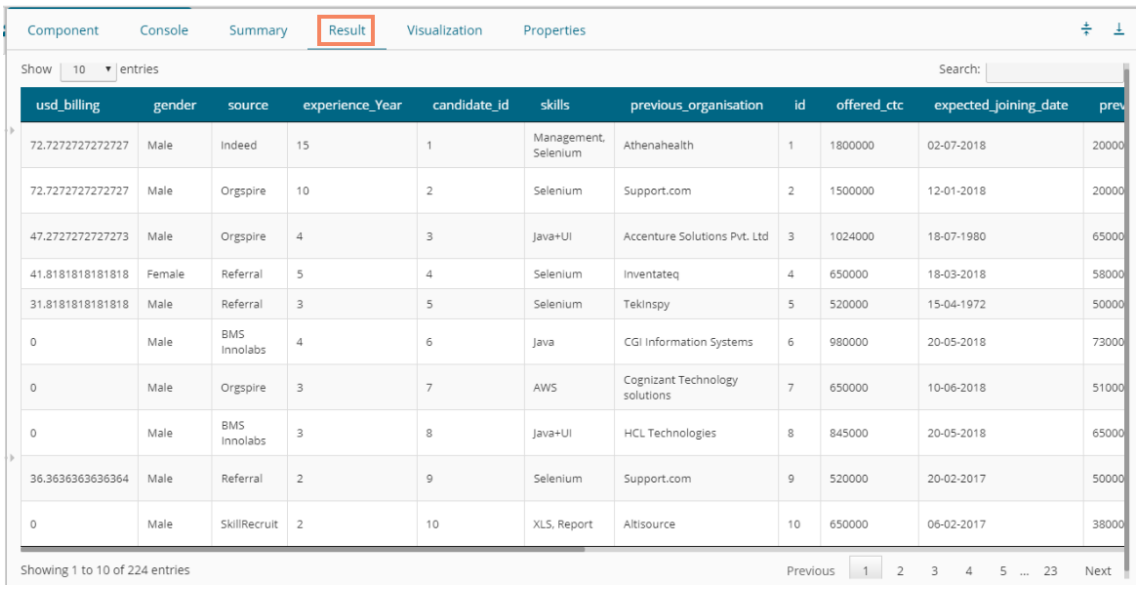


viii) After the Console process gets completed, the user can view the Result data using the 'Result' tab.

ix) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component in the workspace.

b. Click the 'Result' tab.



7.5.3. Decimal-Scaling

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

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

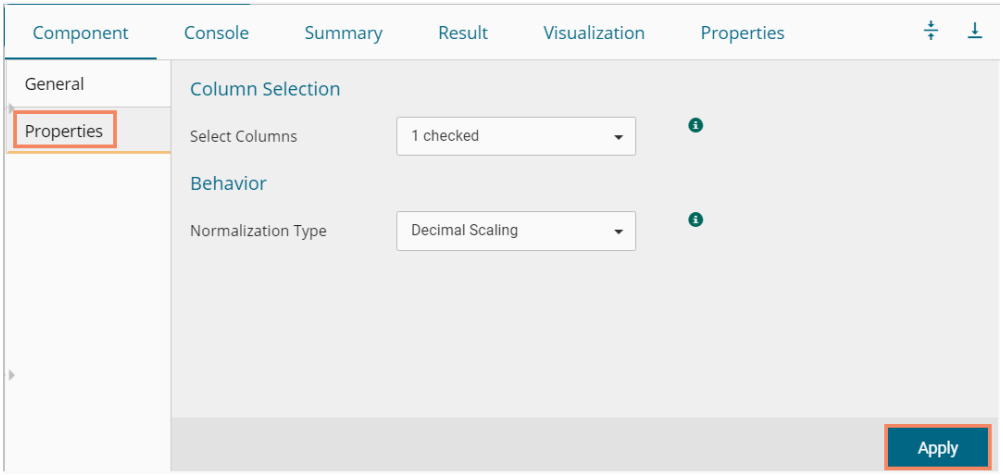
Note: In the decimal-scaling expression, '**c**' is the smallest integer so that max(new_v) < 1.

i) Select and drag **the '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:

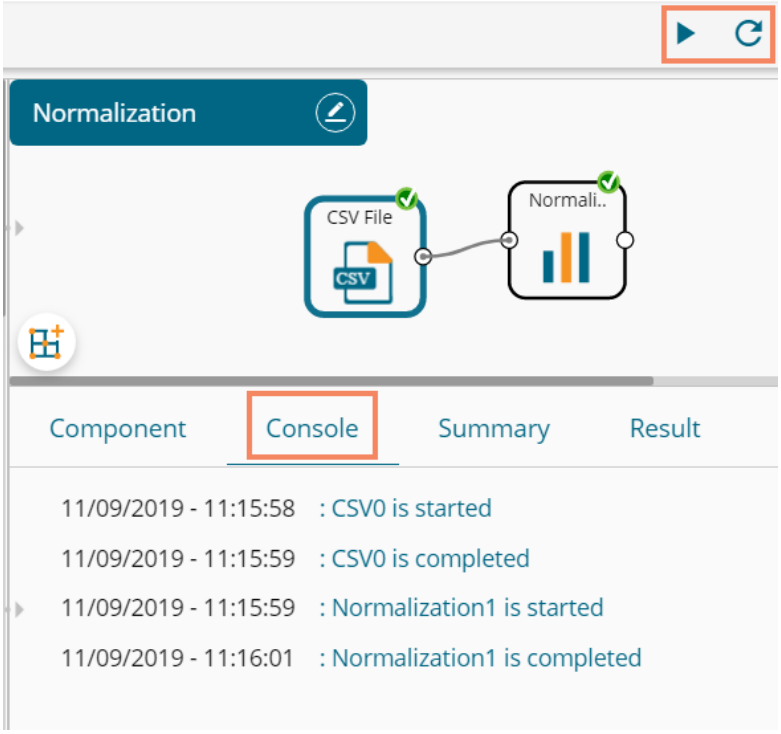
Properties

a. **Column Selection**
   i. **Select a Column**: Select a column using the drop-down menu (Only the numerical column gets selected).
b. **Behavior**
   i. **Normalization Type:** Select '**Decimal Scaling**' normalization type from the drop-down menu.

v) Click **'Apply'** to configure the fields:



vi) Run the workflow by clearing the previous cache.
vii) The 'Console' tab opens displaying the progress of the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.

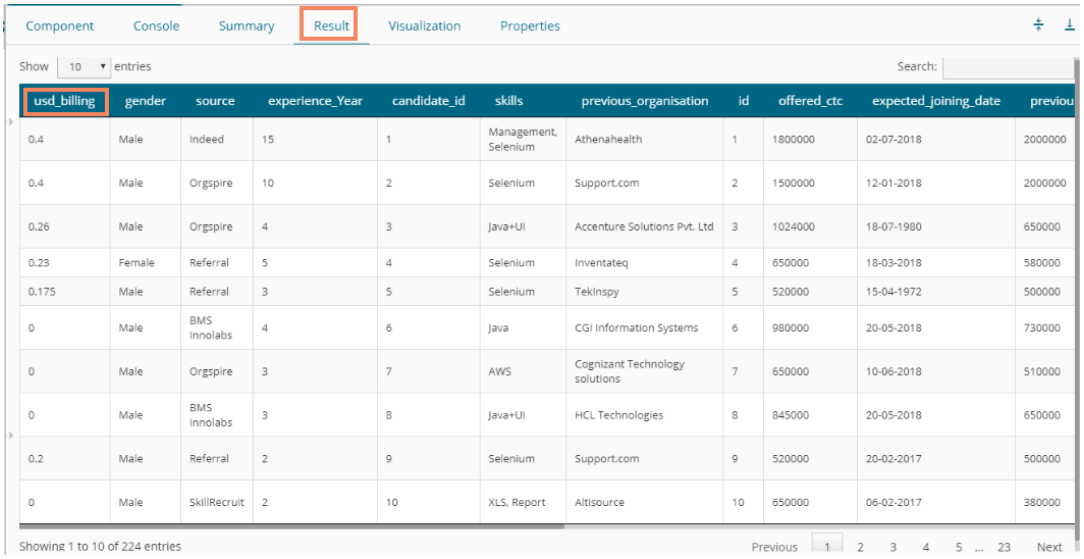


viii) After the Console process gets completed, users can view the Result data using the 'Result' tab.

ix) Follow the below-given steps to display the Result view:

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

b. Click the 'Result' tab.



Note:

a. Normalization displays columns containing only numerical data.

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

7.6. Sample

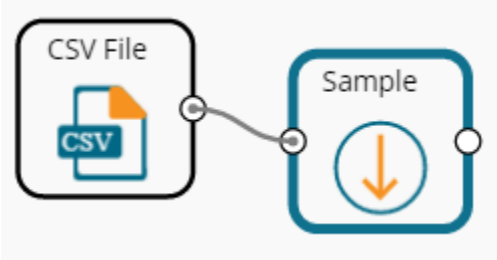
This component can be used to select a subsection of data from a large dataset. The sample component supports the following sample types:

7.6.1. Sampling Methods

1. **First N:** It selects the first N records from the data source. E.g., If the chosen value for "N" is 10, then it will select the first ten records from the data.
2. **Last N:** It selects the last N records from the data source. E.g., If the chosen value for "N" is 5, then it will select the last five records from the data.
3. **Every Nth:** It selects every Nth record from the data source, wherein "N" indicates an interval. E.g., If N=3, then the 3rd, 6th, and 9th records get selected from the data.
4. **Simple Random:** It selects records randomly as per the value of "N" or the percentage mentioned for "N" from the data source. E.g., If the selected value for "N" is four then, it selects randomly any four records from the data source. If the selected value for "N" is 4% then, it selects 4% of records from the data source.
5. **Systematic Random:** It selects data based on the bucket size. E.g., If the chosen value for the bucket is two then, it selects 1st, 3rd, 5th records or 2nd, 4th, 6threcords from the data source.

7.6.2. Steps to Apply a Sampling Method

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



iv) Configure the required component fields:

Properties

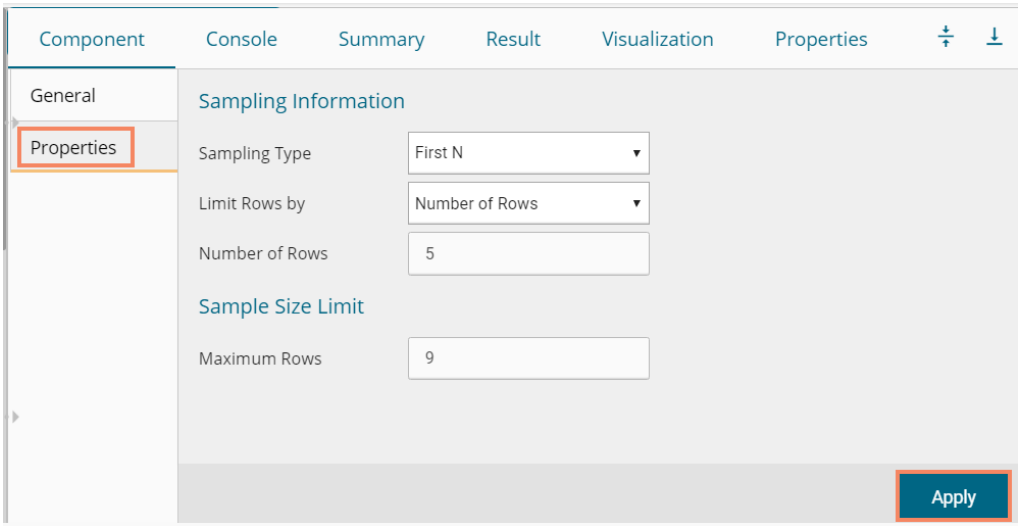
a. Sampling Information

i. **Sampling Type**: Select an option from the drop-down menu
ii. **Limit Rows by** Select an option from the drop-down menu. This field will offer two options, as described below:
1. **Numbers of Rows**: By selecting this option, it will display a new field '**Number of Rows.**'
2. **Percentage of Rows**: By selecting this option, it will display the new field '**Percentage of Rows.**'

b. Sample Size Limit

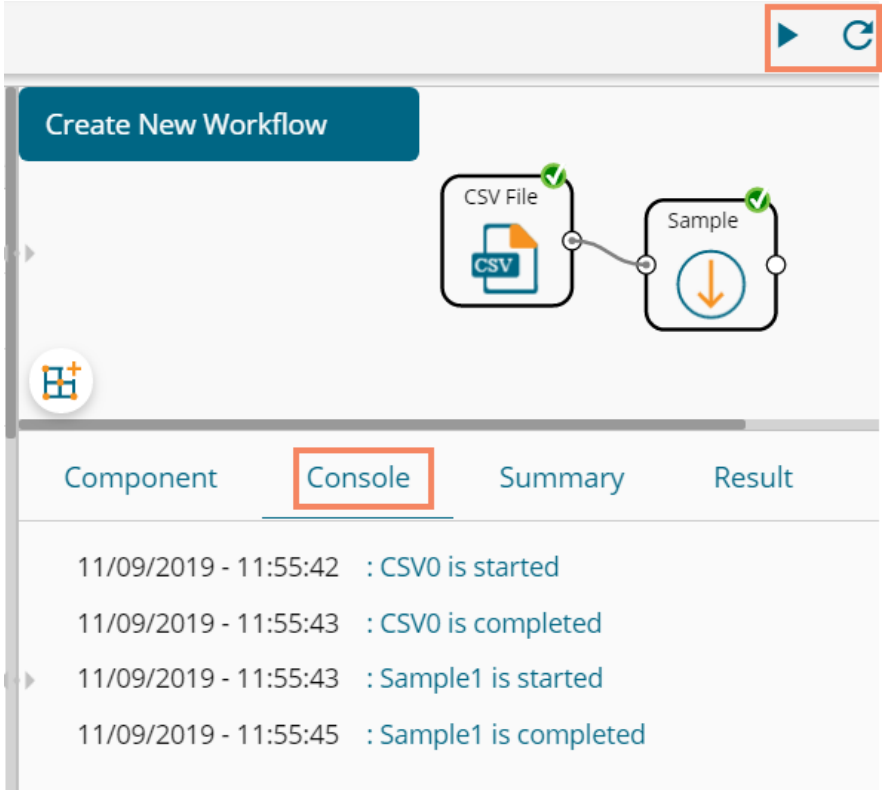
i. **Maximum Rows**: The maximum number of rows that can be viewed in the 'Result' tab (It is an optional field)

v) Click the 'Apply' option.



vi) Run the workflow by clearing the previous cache.

vii) The 'Console' tab opens displaying the progress of the process. The completion of the process gets marked by the green checkmarks on the top of the dragged components.

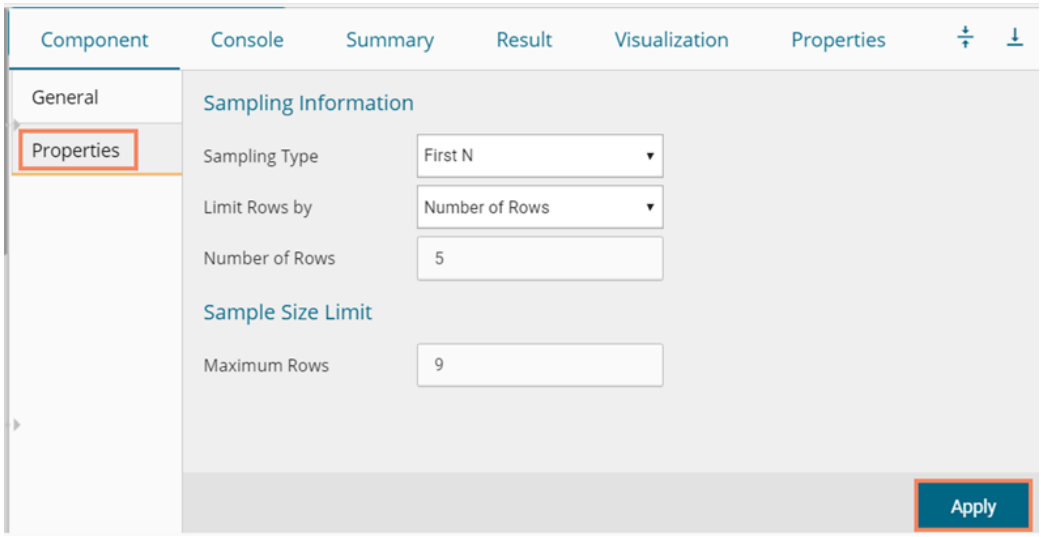


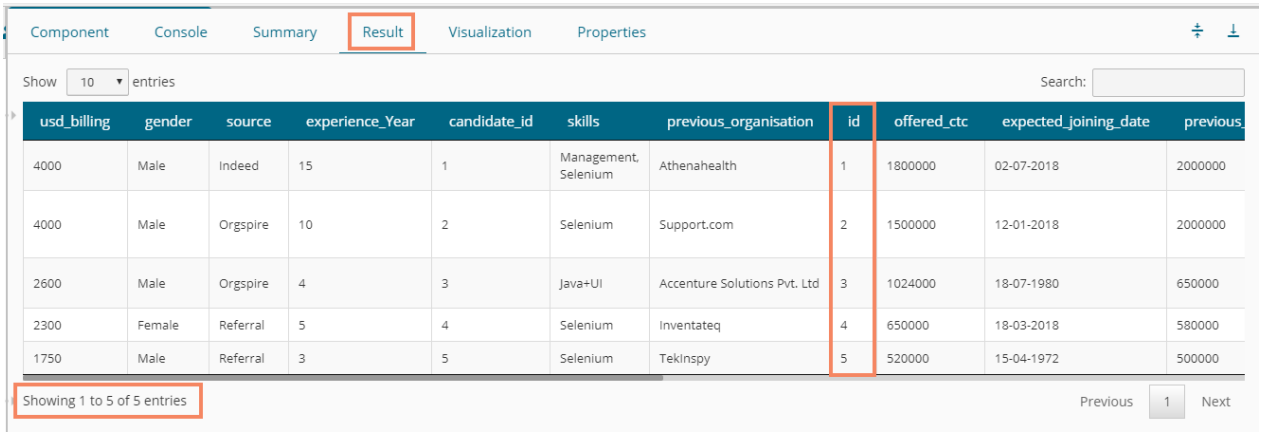
viii) After the Console process gets completed, open the 'Result' tab to view Result data.

ix) While accessing the 'Result' tab, the user gets the Result view based on the selected Sampling Type.

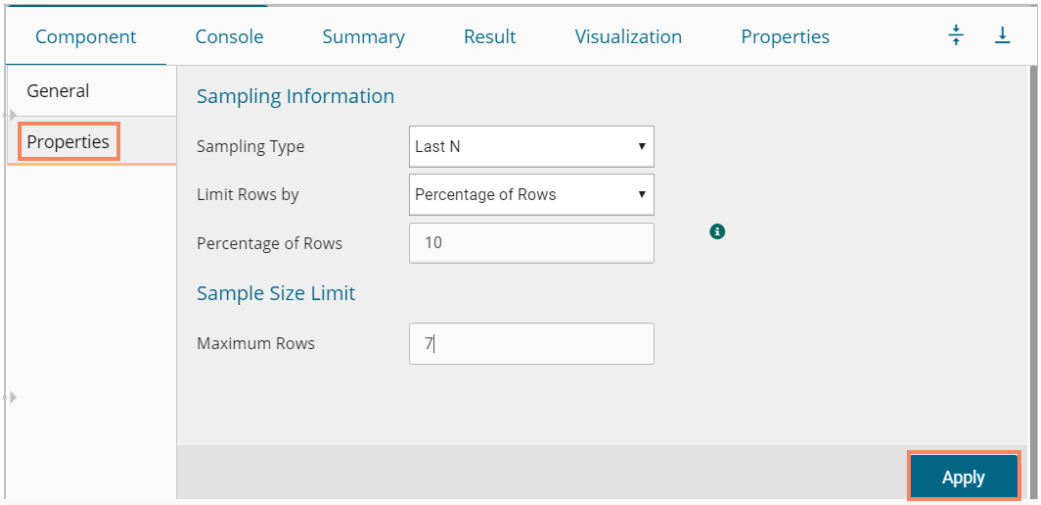
7.6.3. Result View for the Available Sampling Methods

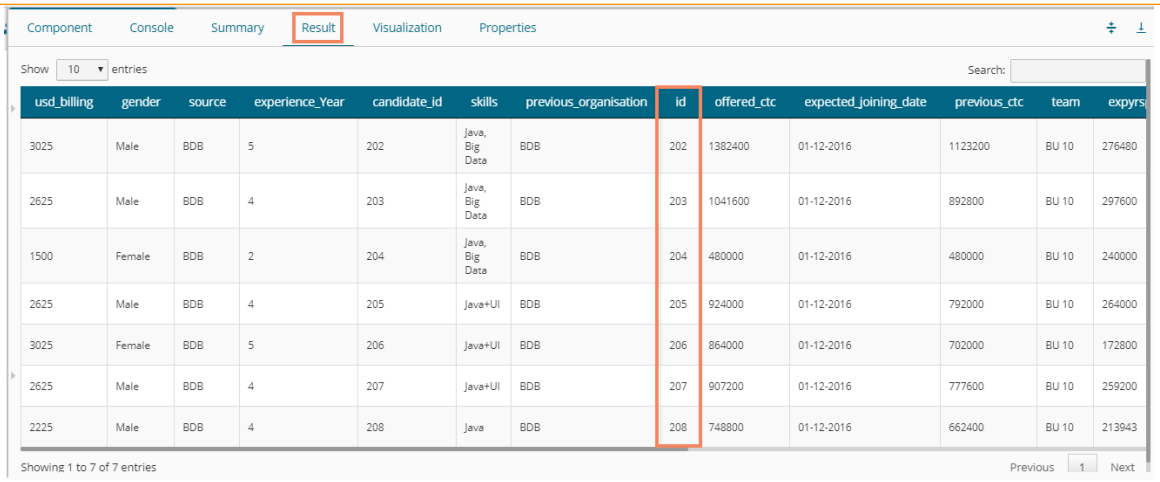
1. **First N (Where 'N' is 1 number of the row)**



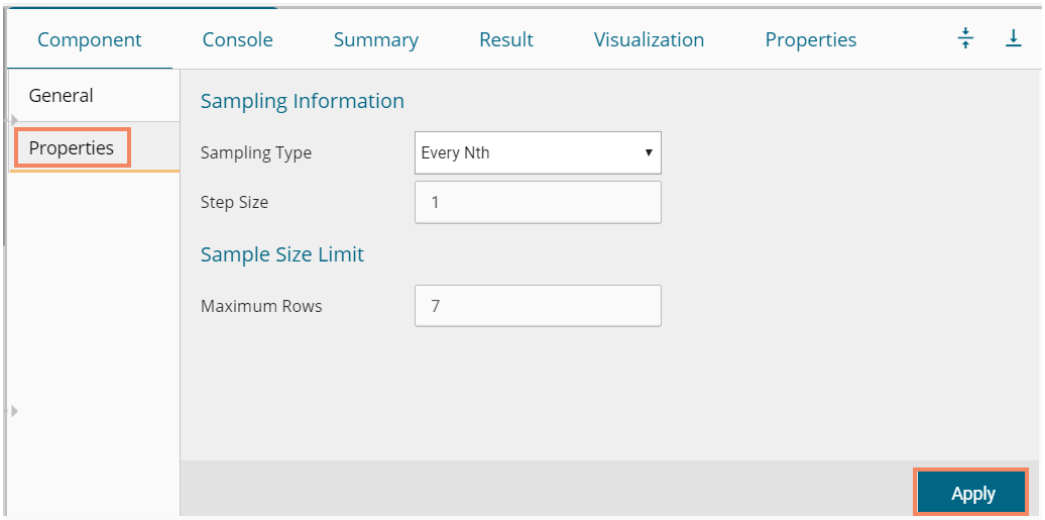


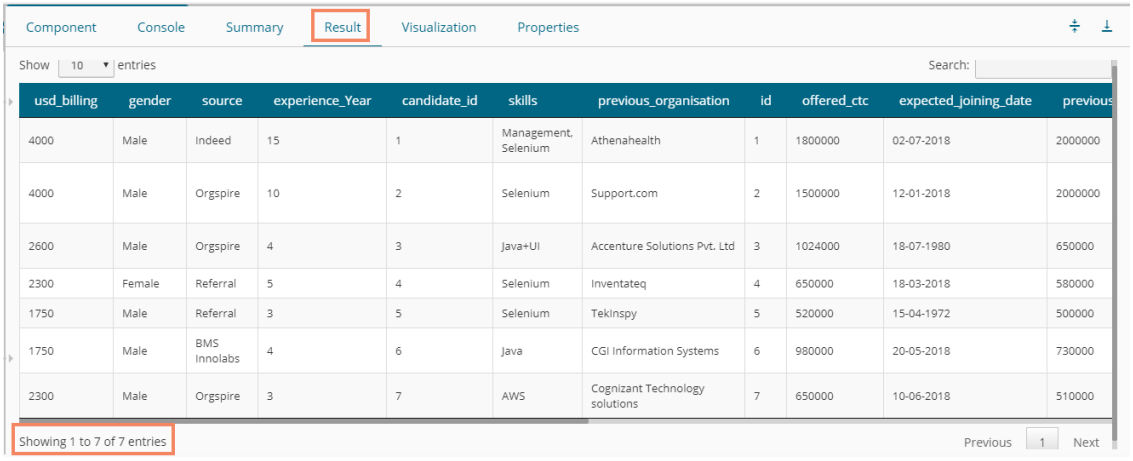
2. **Last N ('N' is 10% and maximum rows are 7)**



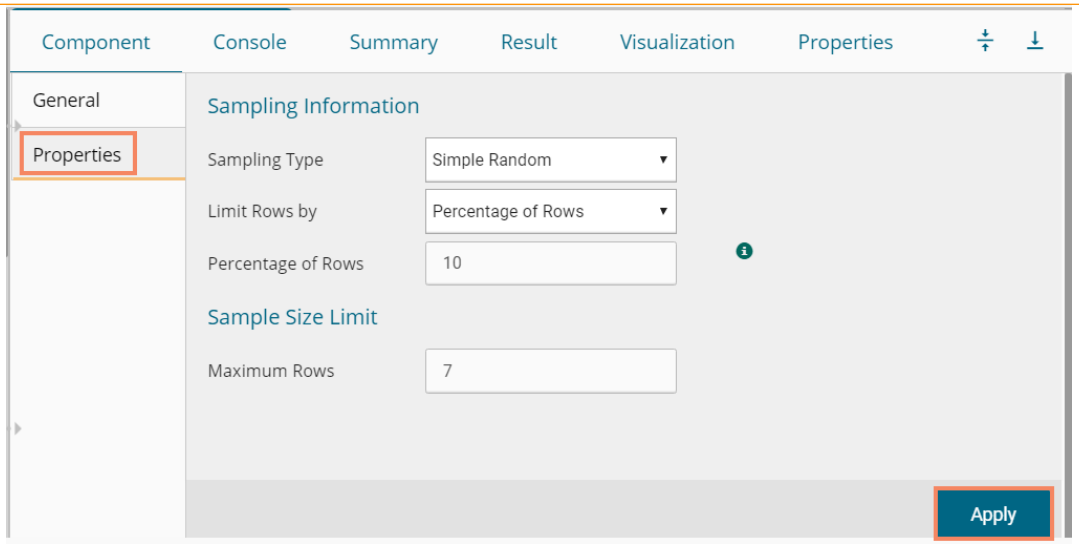


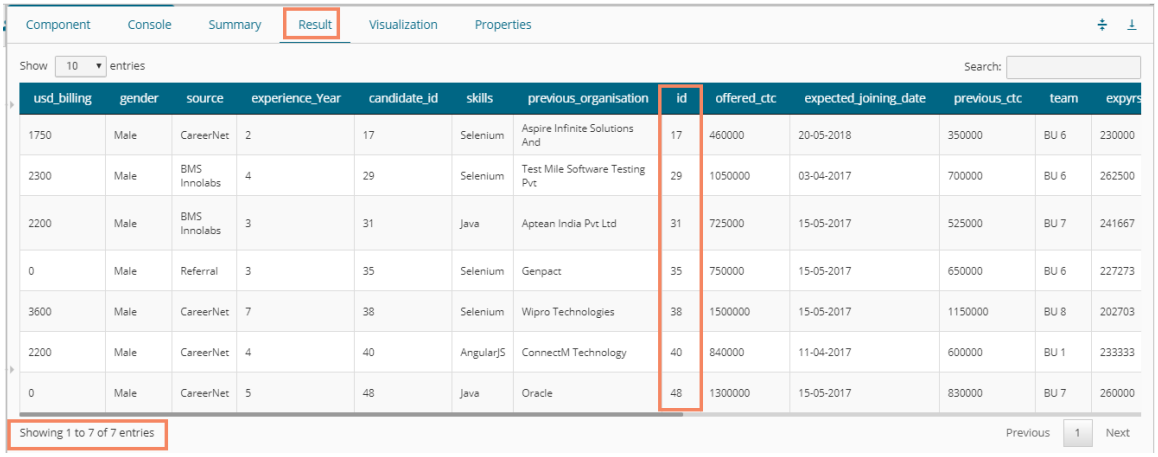
3. **Every Nth (Interval is 3, and the maximum rows are 7)**



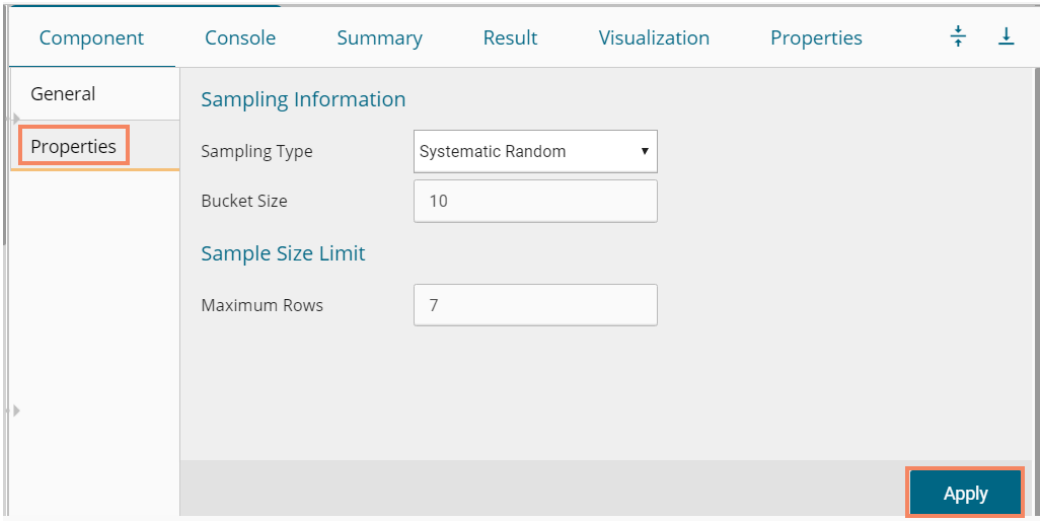


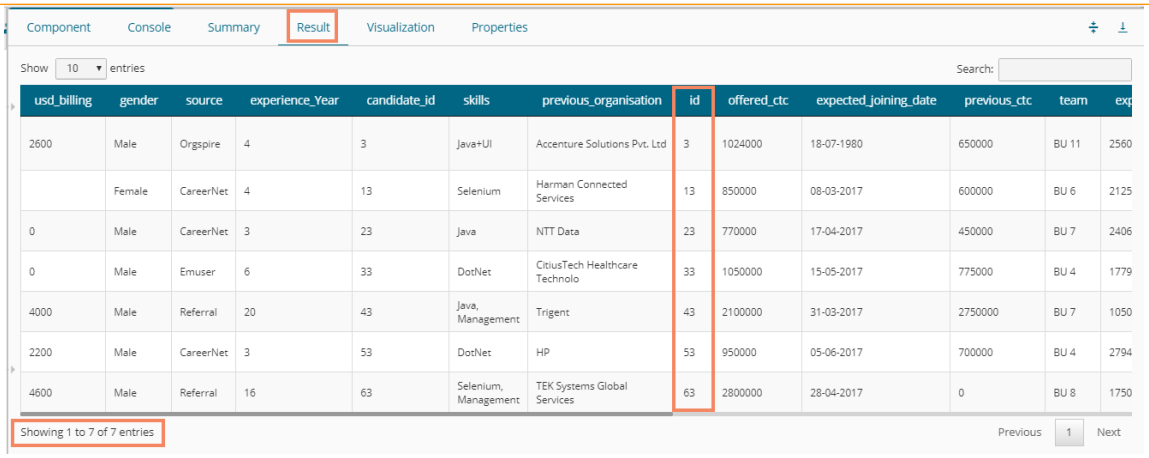
4. **Simple Random (the 'Maximum Rows' are 7). The randomly selected seven rows will be displayed.**





5. Systematic Random (Bucket Size is 10).



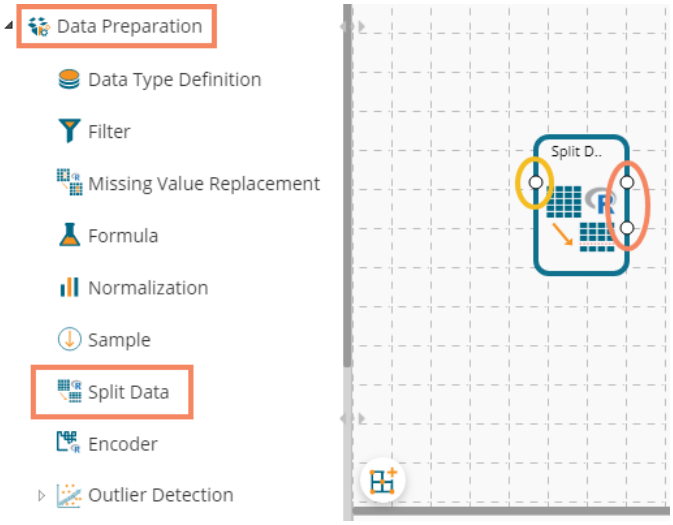


7.7. Split Data

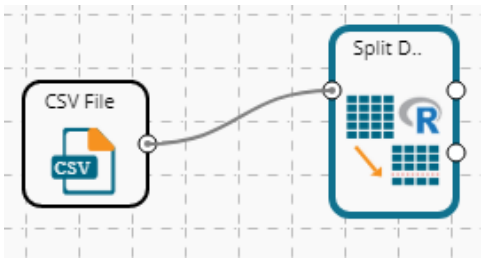
The Split Data component is used to split a dataset into training and testing per percentage and method. Once the most suitable model is decided from the trained data, users can pass test data to validate the model.

Split Data appears as a leaf node under the Data Preparation Tree node (the current description displays the Split Data component provided under the R Workspace).

The Split Data consists of two connector nodes: An upper node for the **training data set** and a lower node for the **testing data set**.



i) Select the '**Split Data**' component and connect it with a valid data source.



ii) Click the '**Split Data**' component in the workspace.

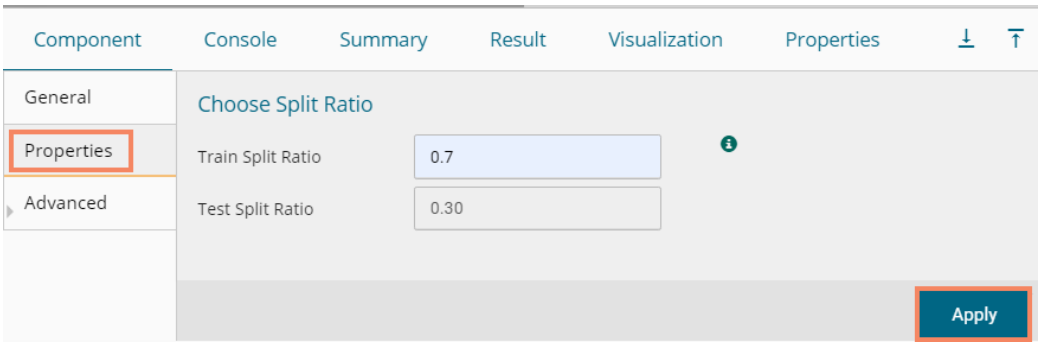
iii) The user gets directed to the Properties fields provided under the '**Components**' tab

iv) The user can choose the size of the first partition:

a. Relative (train): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and the sum of train and test data should be 1)

b. Relative (test): Enter a value to decide the ratio of train data out of the dataset (Type: Decimal, Range: 0-1 and the sum of train and test data should be 1)

Note: If the user does not want to configure the Advanced tab then the 'Apply' option provided for the '**Properties**' tab must be clicked, otherwise click the 'Apply' option provided for the Advanced tab.



v) The user can configure the sampling type using the Advanced fields if needed.

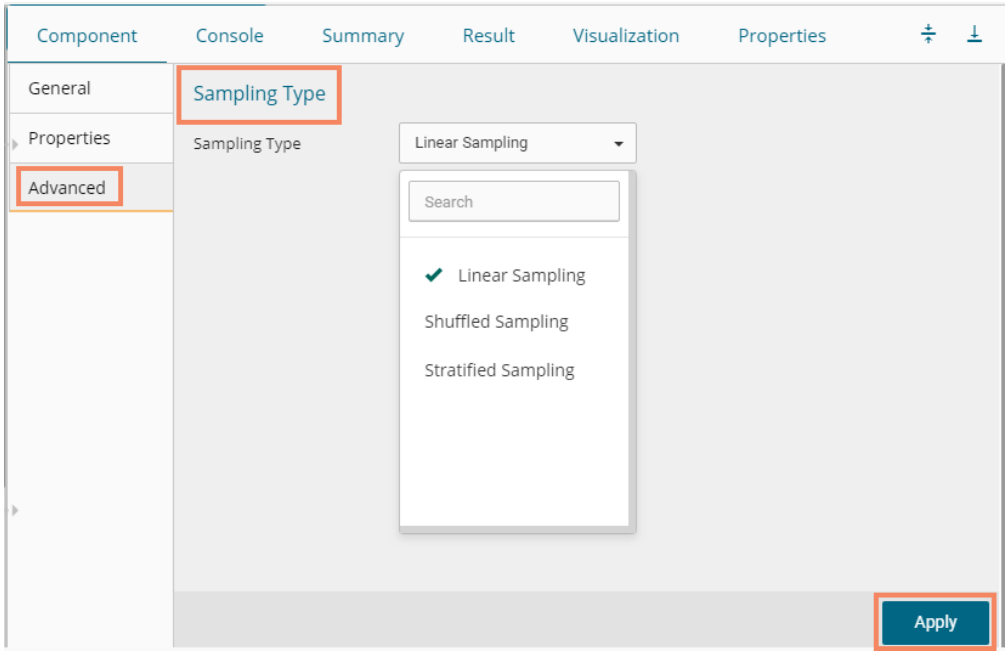
a. Sampling Type: Select any one option from the drop-down menu

i. Linear Sampling

ii. Shuffled Sampling

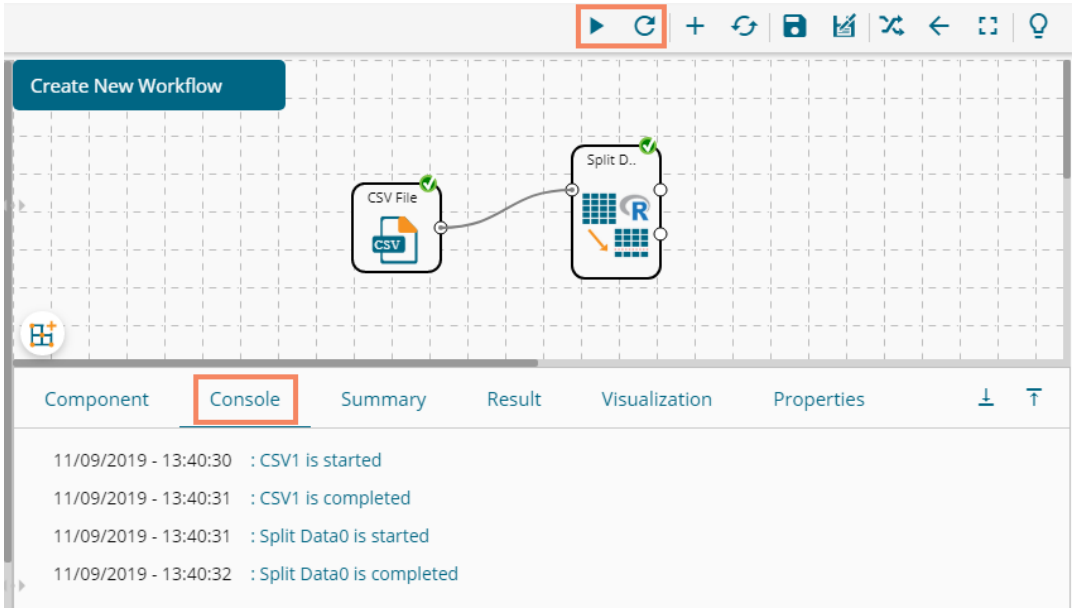
iii. Stratified Sampling

vi) Click the 'Apply' option.



vii) Run the workflow after clearing the cache.

viii) The 'Console' tab opens displaying the progress of the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.



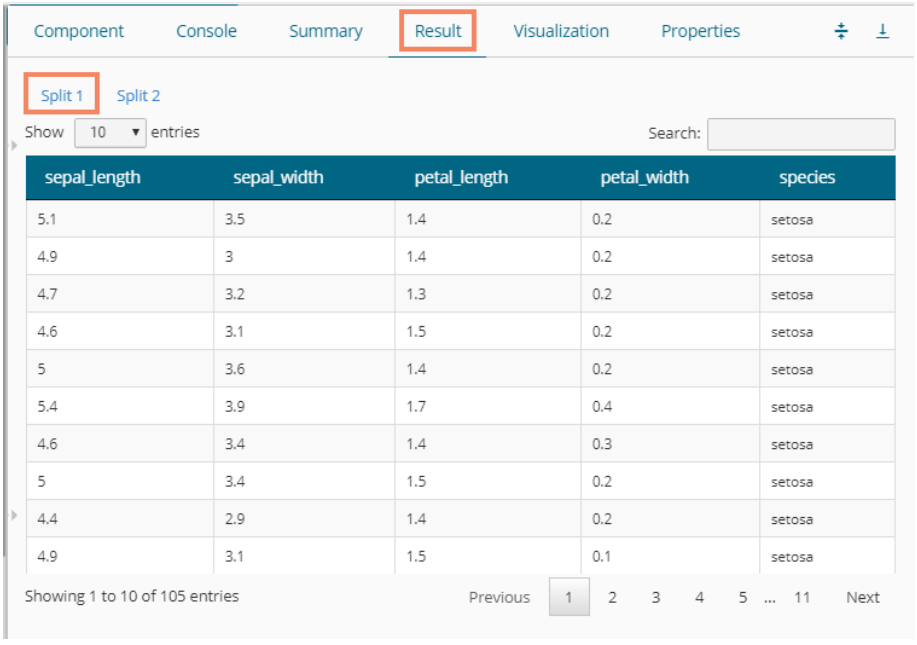
ix) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component in the workspace.

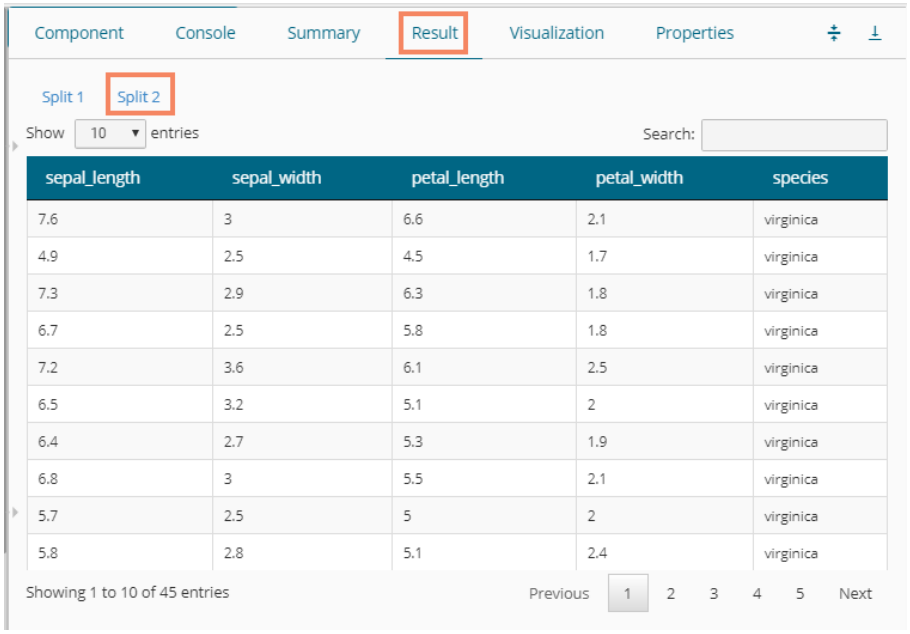
b. Click the 'Result' tab.

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

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



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



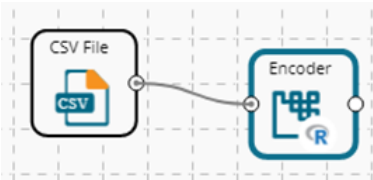
Note:

a. The current document covers steps to deal with a CSV File dataset for all the R Data Preparation components. Similar steps can be followed for a Data Service data set.
b. The Data Preparation list may vary based on different workspaces, but the configuration process remains the same. All the unique Data Preparation components are explained under this section.

7.8. Encoder

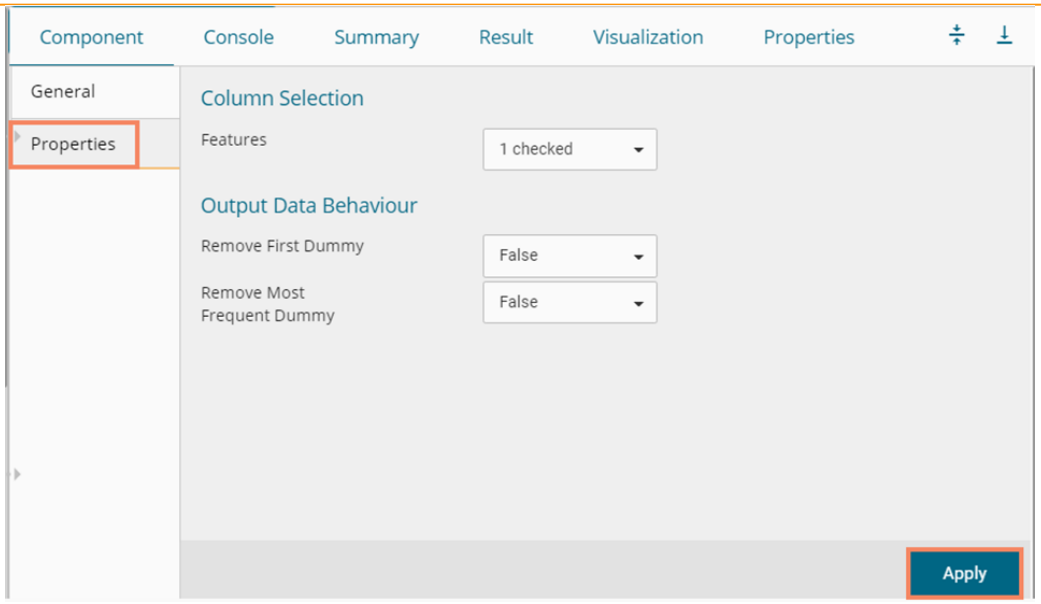
Encoding operation determines the existence of a string value in a selected column within each row in a worksheet. It converts categorical values in a worksheet to numeric values (only zero and one) required by machine learning algorithms.

i) Drag the Encoder component and connect it with a configured data source.



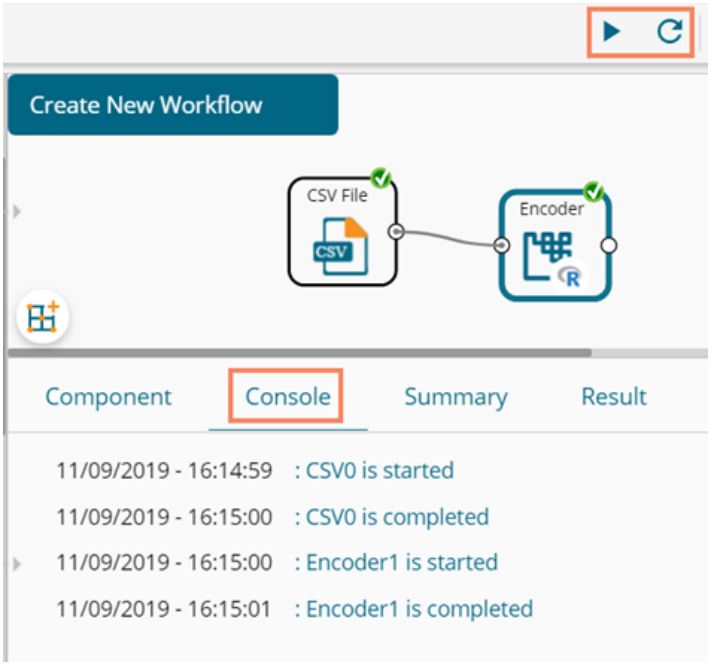
ii) Click the Encoder component to configure the Properties tab:

a. Column Selection
   i. Feature: Select a column using the drop-down option. All the string value columns get listed.
b. Output Data Behaviour
   i. Remove First Dummy: Select an option from the drop-down menu (out of True/False)
   ii. Remove Most Frequent Dummy: Select an option from the drop-down menu (out of True/False)
   iii. Click the 'Apply' option.



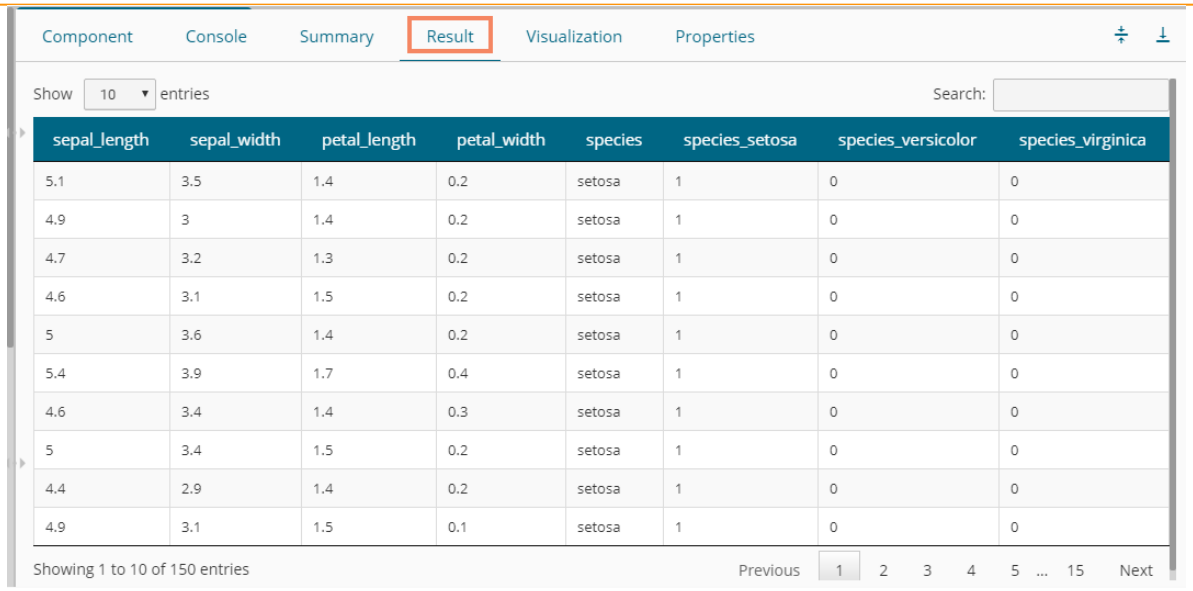
iii) Run the workflow.

iv) The Console tab opens displaying the process. The completion of the Console process gets marked by the green marks on the top of the dragged components.



v) Open the Result tab to see the processed data.

a. Click the Encoder component.
b. Click the 'Result' tab to open the Result view.
(The data of the selected column gets displayed by the 0 and 1 numbers)

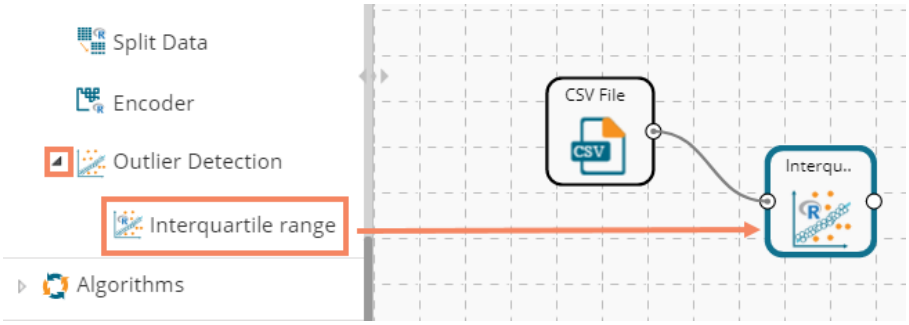


7.9. Outlier Detection

This component is used to discover patterns in the 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.

7.9.1. Interquartile Range

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



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

a. **Output Information**

i. **Output Mode**: Select a mode of display for output data.

1. **Show Outlier**: Select this option to add a Boolean column to the input data identifying whether the Resultant value is an outlier.
2. **Remove Outlier**: Select this option to remove outlying values from the input data.

b. **Column Selection**

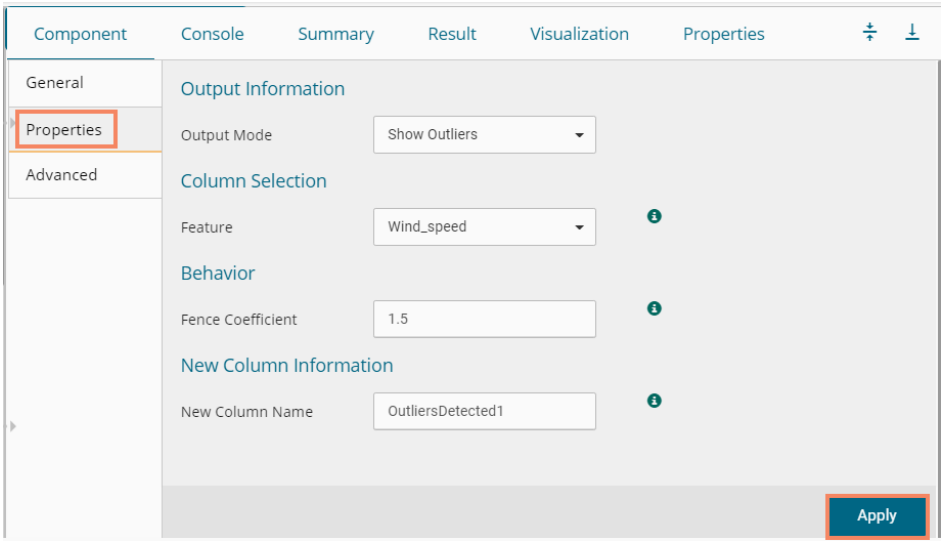
i. **Feature:** Select an input column that can be used to perform the analysis.

c. **Behavior**

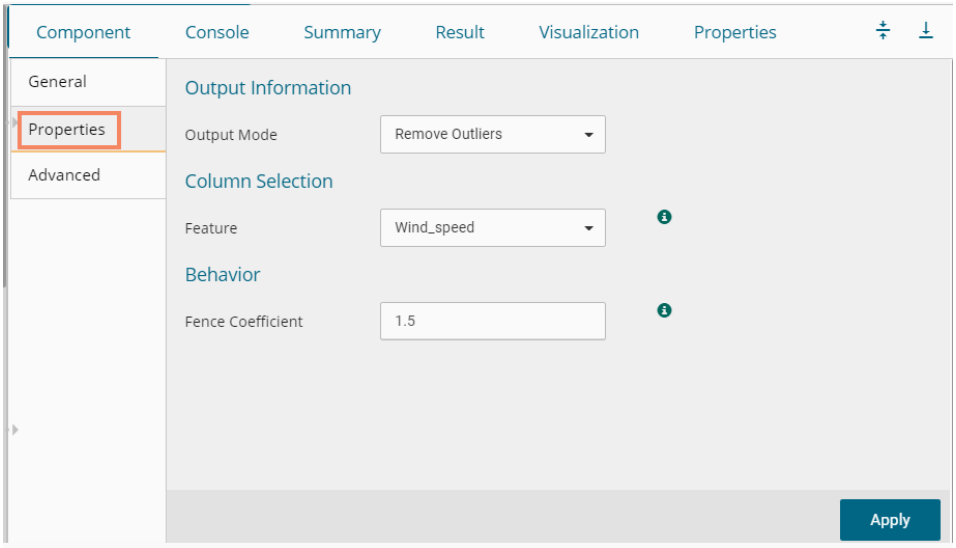
i. **Fence Coefficient:** Enter the permissible deviation limit for values from the Interquartile Range (The default value for this field is 1.5)

d. New Column Information

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



Properties fields with the '**Remove Outliers**' option selected to display Output Information.



Note: If the user does not need to configure the '**Advanced'** tab, then the 'Apply' option must be clicked from the Properties tab.

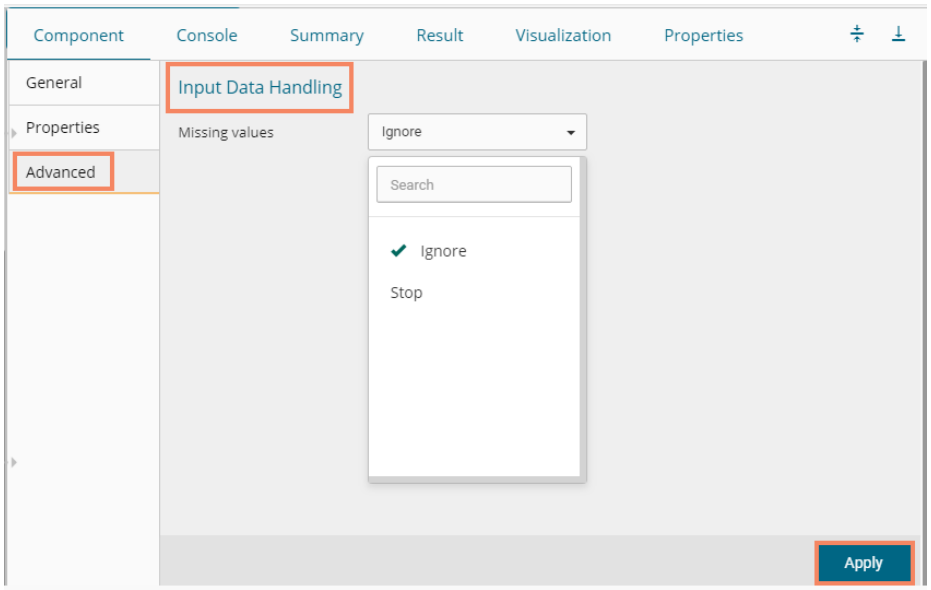
iii) Click the '**Advanced**' tab and configure if required:

a. Input Data Handling

i. **Missing Values**: Select a method to deal with missing values from the drop-down menu.

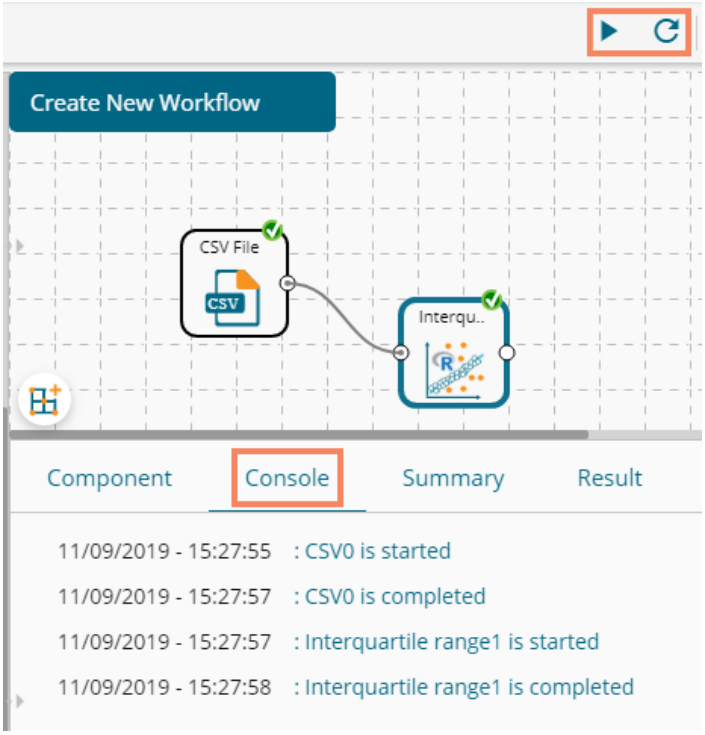
1. **Ignore**: Select this option to skip the records containing missing values in the columns.
2. **Stop**: Select this option to stop the application of the algorithm if a value is missing in any column.

iv) Click the **'Apply'** option.



v) Run the workflow after clearing the cache.

vi) The 'Console' tab opens, displaying the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.

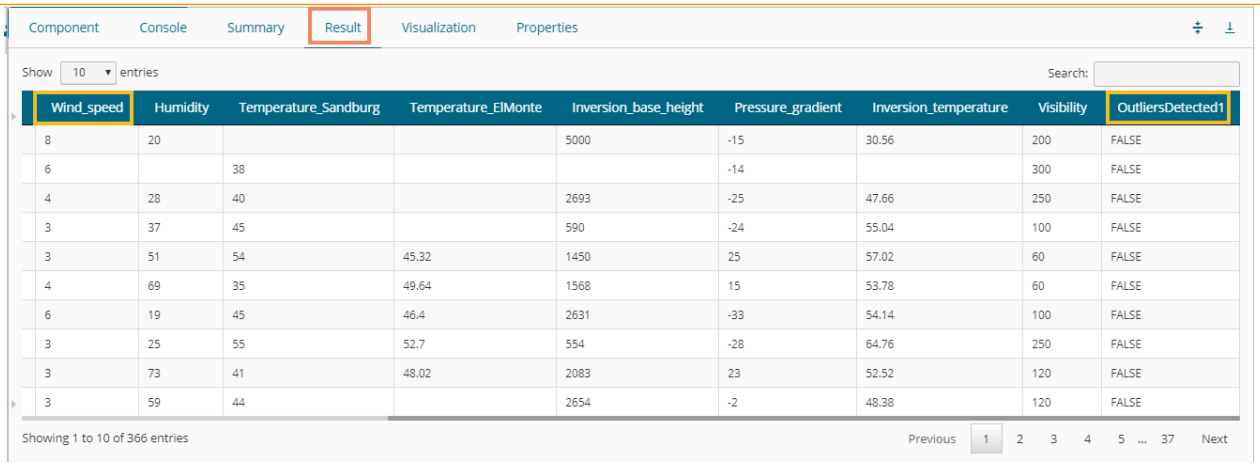


vii) Follow the below-given steps to display the Result view:

a. Click the dragged Outlier component.

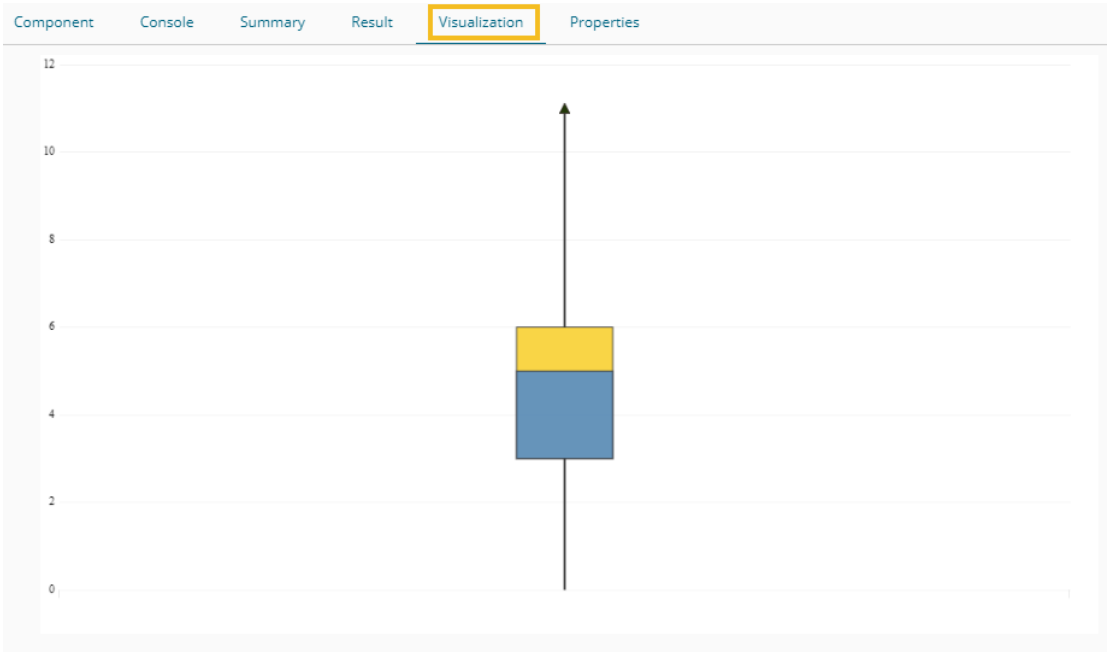
b. Click the 'Result' tab.

A new column '**OutliersDetected1'** displays in the Result data (If '**Show Outliers**' option has been selected).



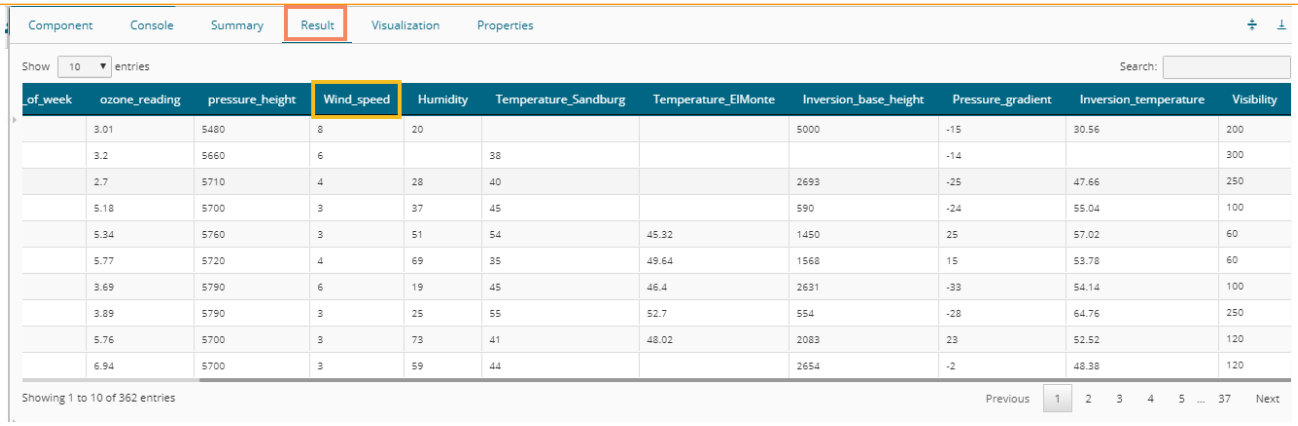
viii) Click the 'Visualization' tab.

ix) The Result data is displayed via the Box Plot chart.



OR

The outliers column is removed from the Result data (If the '**Remove Outliers**' option has been selected).



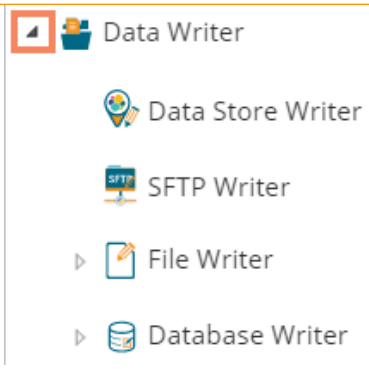
_of_week	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	Inversion_base_height	Pressure_gradient	Inversion_temperature	Visibility
	3.01	5480	8	20			5000	-15	30.56	200
	3.2	5660	6		38			-14		300
	2.7	5710	4	28	40		2693	-25	47.66	250
	5.18	5700	3	37	45		590	-24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

Click the 'Visualization' to see the Result data via the Box Plot chart.

8. Data Writers

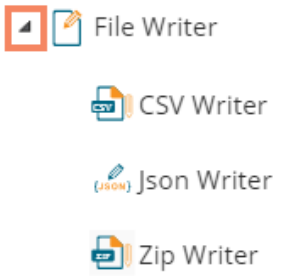
Data Writers are provided to store the Results of the Data Science Workspace in flat files or databases for further in-depth analysis. The Data Science Workspace contains the following types of Data Writers across the various Workspaces.

1. Data Store Writer
2. SFTP Writer (only available for the Python Workspace at present)
3. File Writer
4. Database Writer



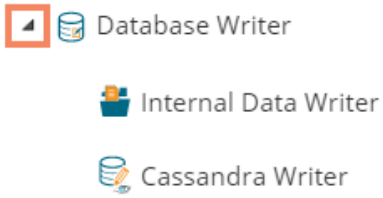
The File writer has the following categories:

1. CSV Writer
2. JSON Writer
3. Zip Writer (only for the Python Workspace)



The Database Writer has the following categories:

1. Internal Data Writer
2. Cassandra Writer

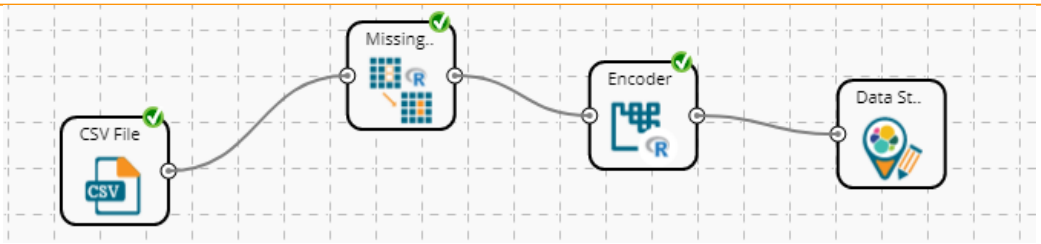


Find the step by step description for each data writer given below:

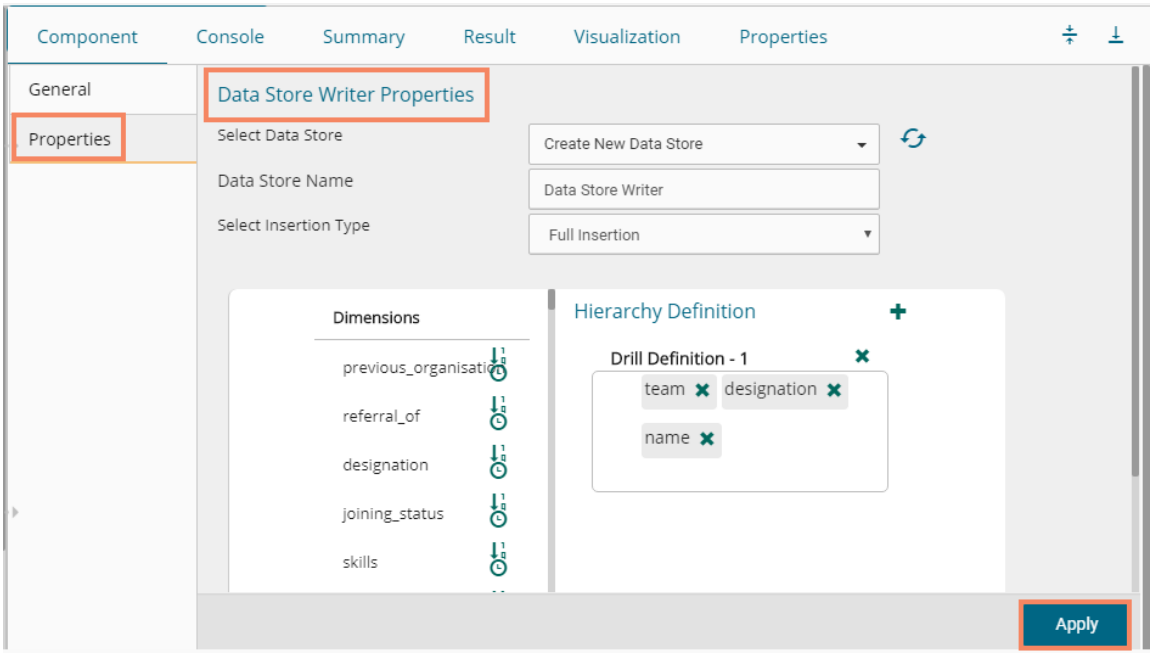
8.1. Data Store Writer

The elastic Search Writer component is listed under the Data Writer Tree node. The Data Store Writer allows the user to write the processed data onto the Elastic Search server, which makes it more distributed.

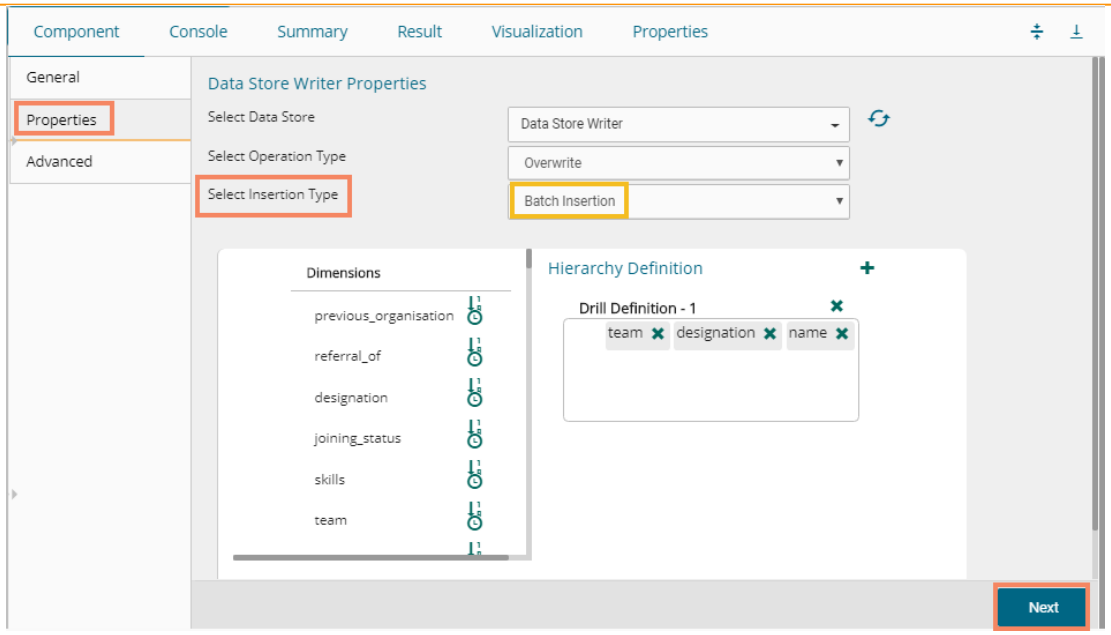
i) Drag the Data Store Writer component to the workspace and connect it with a configured data source or any valid combination of a data source with other given components. (In this case, there is a combination of CSV file with a Missing Value Replacement and Encoder components to bring the input data to the Data Store writer)



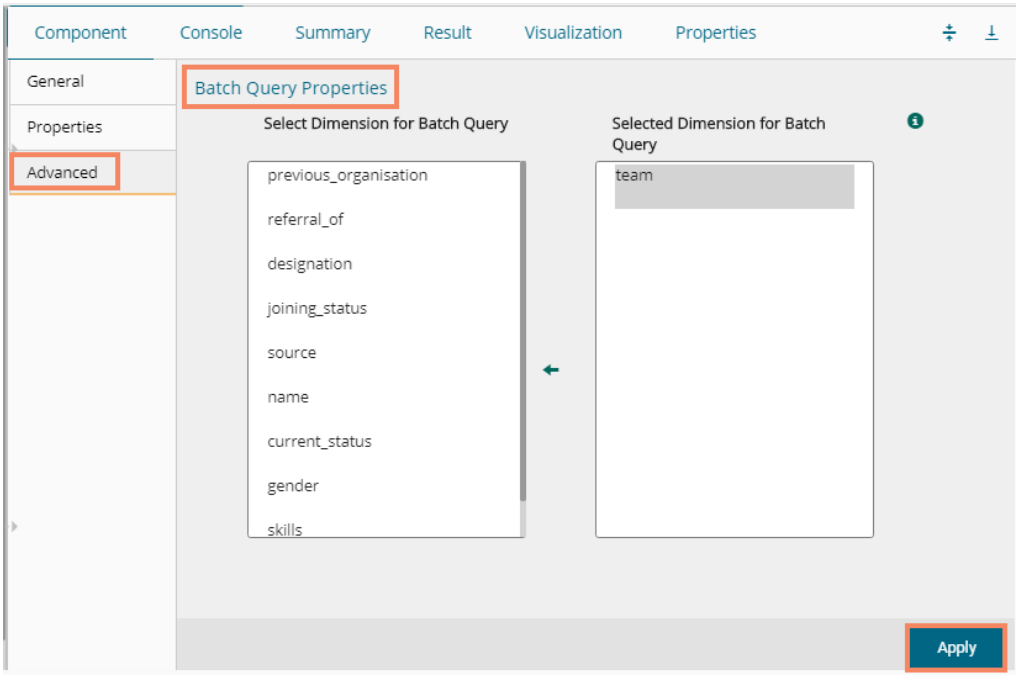
ii) Click on the connected Data Store Writer component.
iii) The component tab for the data writer opens.
iv) Configure the required component properties.
   i. **Select Data Store**: Select a datastore from the drop-down menu or select the '**Create New Data Store**' option from the drop-down menu
   ii. Select Operation Type: This field appears by choosing an existing Data Store. Select an option from the drop-down menu (Overwrite/Append/Upsert).
   OR
   Data Store Name: This field appears by choosing the 'Create New Data Store' option. The user can define a name for the data store.
   iii. **Select Insertion Type**: Select an insertion type from the drop-down menu (Full Insertion/Batch Insertion)
   iv. The user gets all the Dimensions, Measures, and Time fields from the selected data source.
   v. They can define hierarchy by dragging the required Dimensions using the '**Drill Definition**' box.
v) Click the 'Apply' option.



Note: If the selected insertion type is '**Batch Insertion**,' the Properties configuration displays the '**Next**' option.



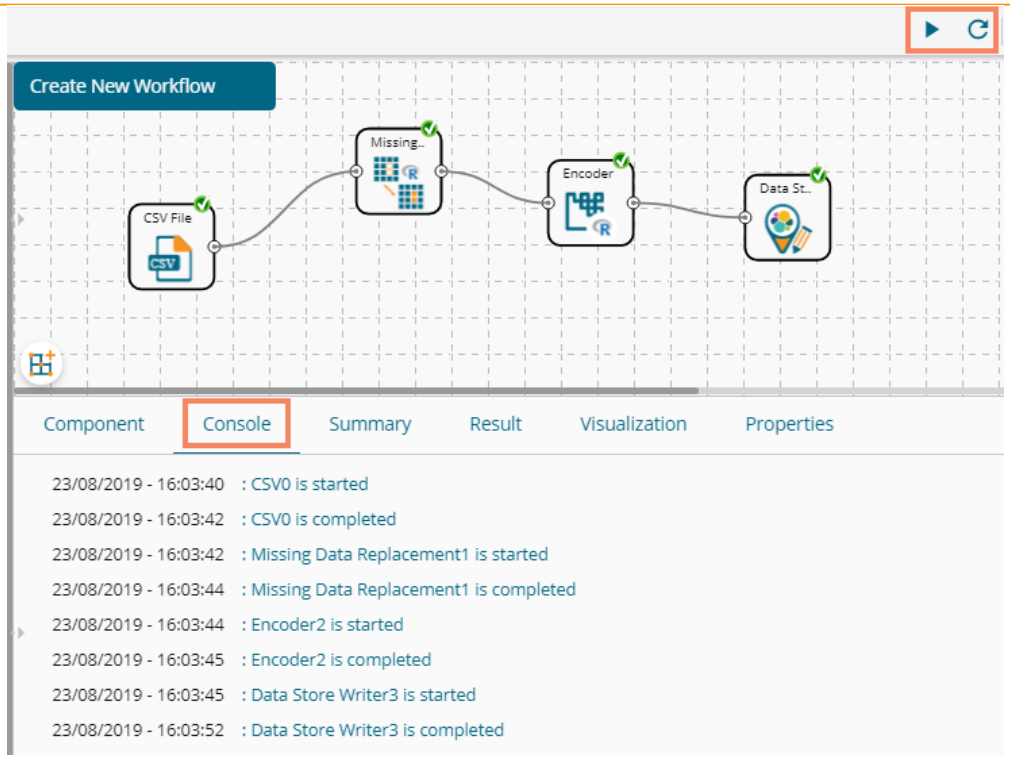
The user gets redirected to the '**Advanced**' fields to configure the Batch Query Properties. Select and then click the 'Apply' option as displayed in the following image:



The user can move only one dimension at a time from the list of '**Select Dimension for Batch Query**' value for the batch query.

vi) Run the workflow after getting the success message.

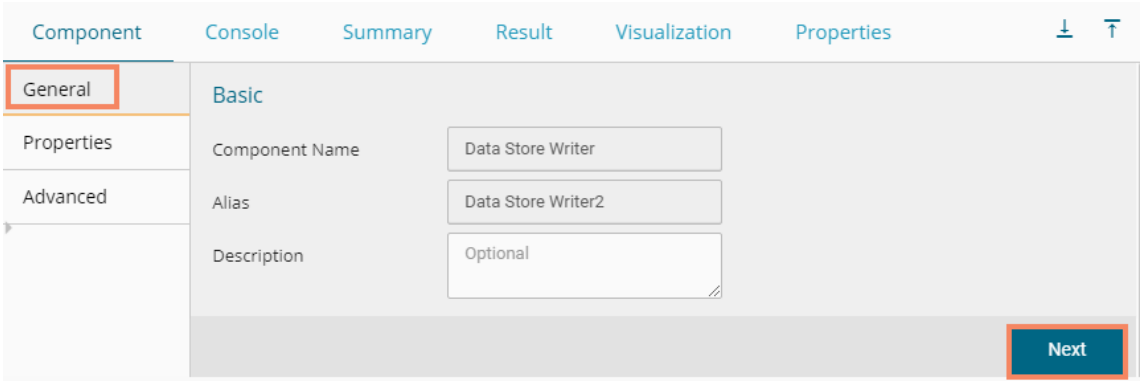
vii) Users will get the process status under the 'Console' tab. The completion of the process is marked with green checkmarks on the components.



viii) The data will be saved in the desired format to the selected Data Store Writer after the Console process gets completed.

Note:

a. The user also gets the '**General**' fields for the Data Store Writer component, but they need not configure it.

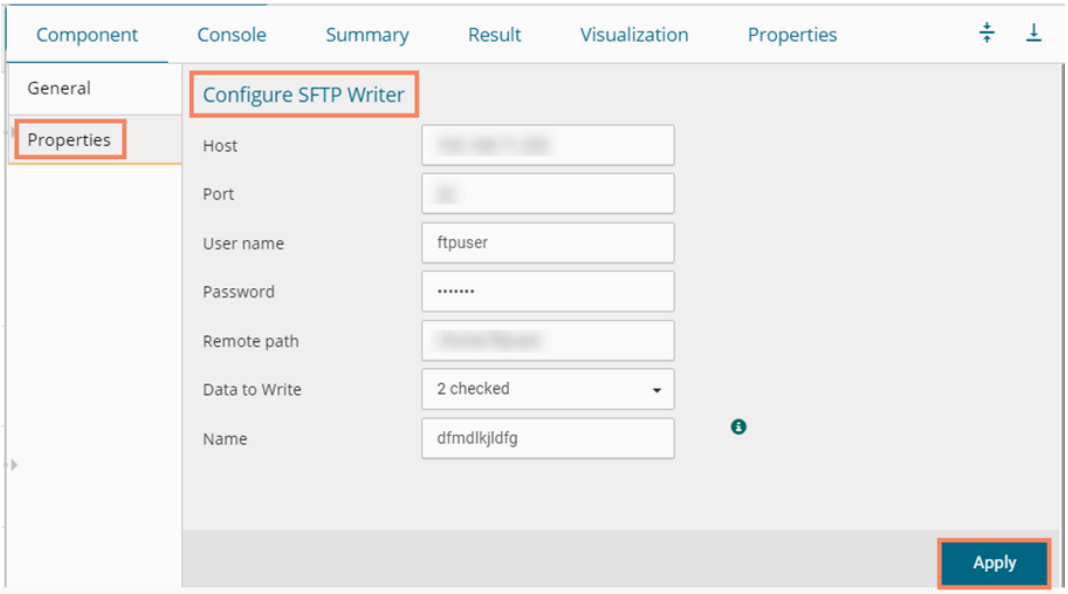


8.2. SFTP Writer

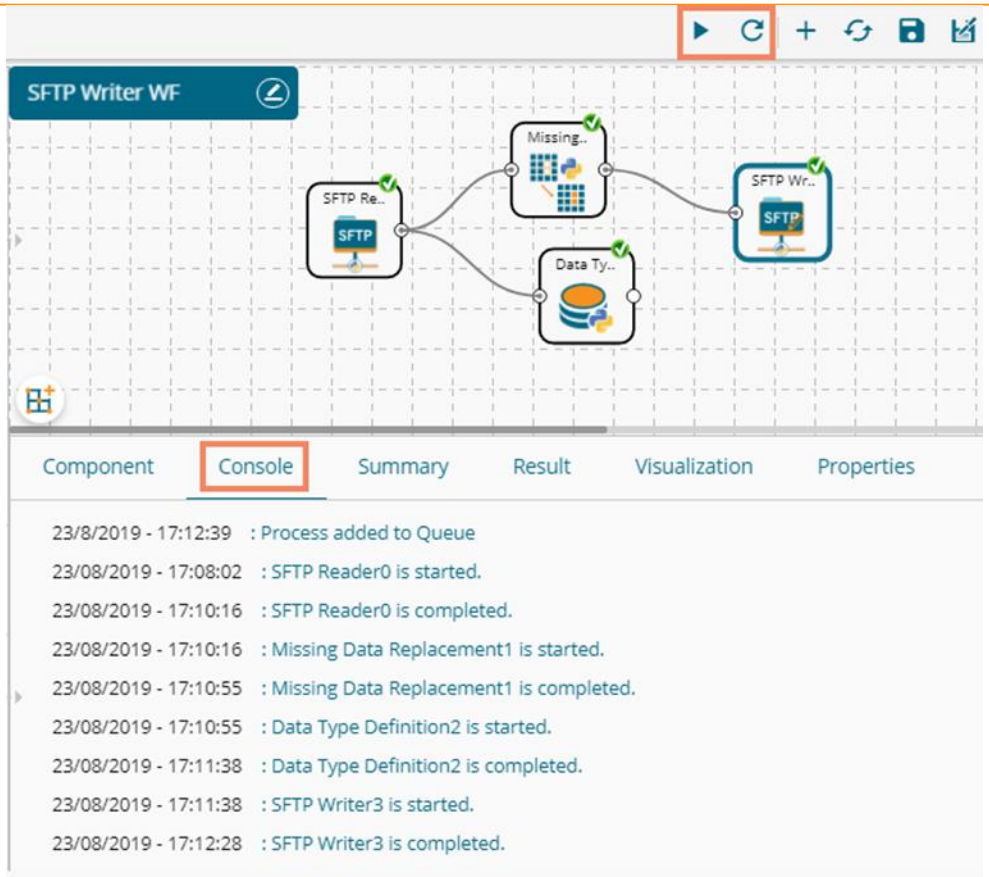
The SFTP Writer is available under the Python Workspace to write the processed data securely.

i) Drag and drop the SFTP writer to the workspace and connect it to the configured combination of the data source and other relevant components to create a workflow.

ii) Click on the writer to get the configuration fields.
iii) Fill in the required details to configure the properties of the SFTP Writer.
   a. Host address
   b. Port Number
   c. Username
   d. Password
   e. Remote path
   f. Data to Write
   g. Name
iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.
vi) The stepwise completion of the process gets displayed in the 'Console' tab. The completion of the Console process is marked by the green checkmarks on the top of the components.



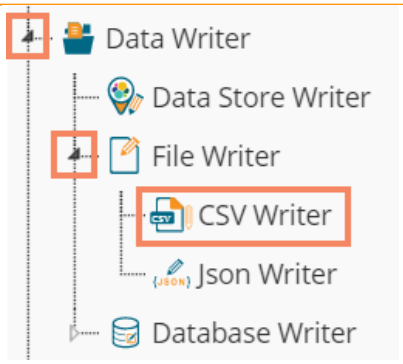
vii) The processed data gets written at the configured SFTP file/location through the SFTP writer.

8.3. File Writer

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

8.3.1. CSV Writer

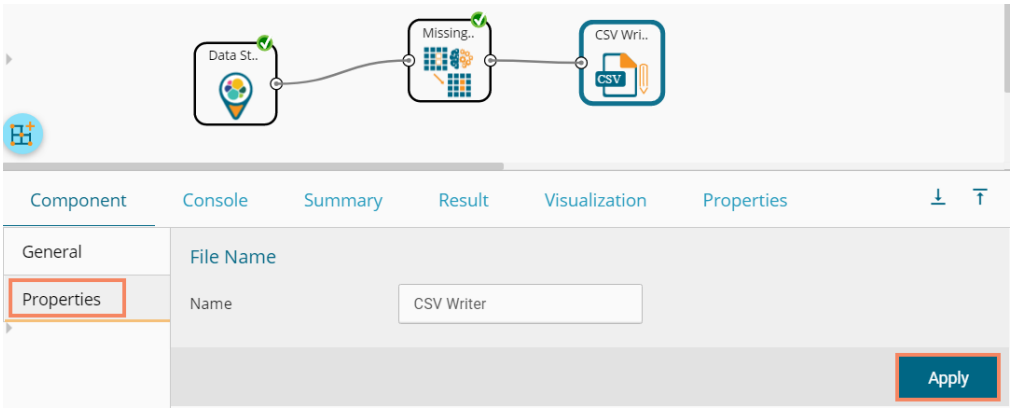
i) Drag and drop the **CSV Writer** component and connect it to a configured workflow to get the input.



ii) Click on the CSV Writer component to access component properties.

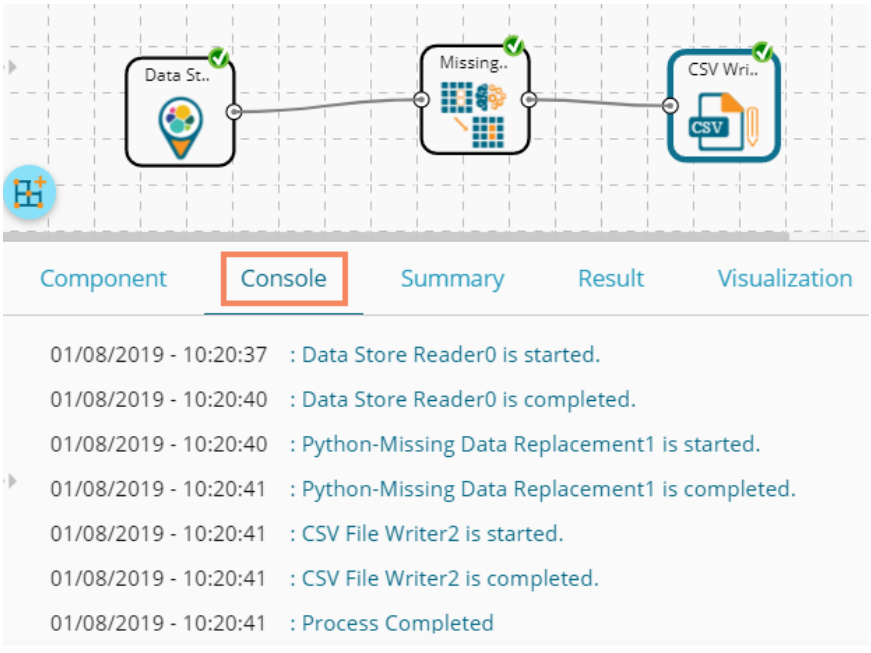
iii) Enter the **'File Name'** in the displayed field**.**

iv) Click the **'Apply'** option.

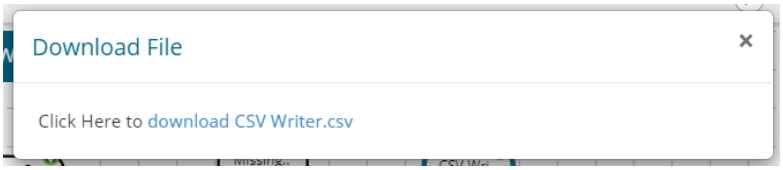


i) Run the workflow after getting the success message.

v) The process status gets displayed under the 'Console' tab, and green checkmarks get displayed a the top of the dragged components indicating completion of the process.



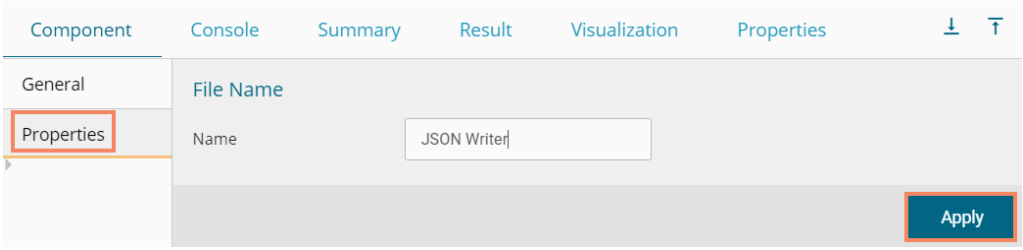
vi) The data gets written in the CSV File.
vii) Click the '**CSV Writer**' component.
viii) A pop-up message appears with a link to download the CSV file.



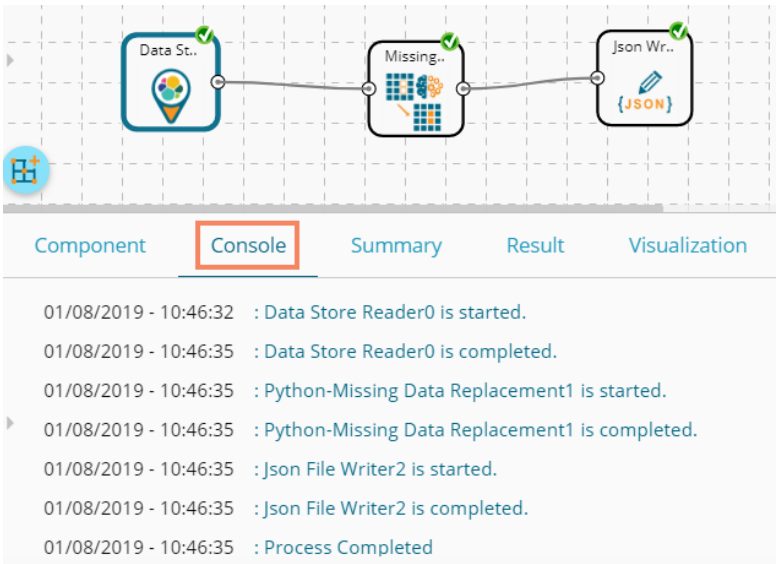
ix) Click the link to download the CSV file.

8.3.2. JSON Writer

ii) Drag and drop the **'JsonWriter'**component to the workspace and connect it to the configured workflow to get input.
iii) Click on the **'JsonWriter'** component to access component properties.
iv) Enter **'File Name'** in the displayed space.
v) Click the **'Apply'** option**.**

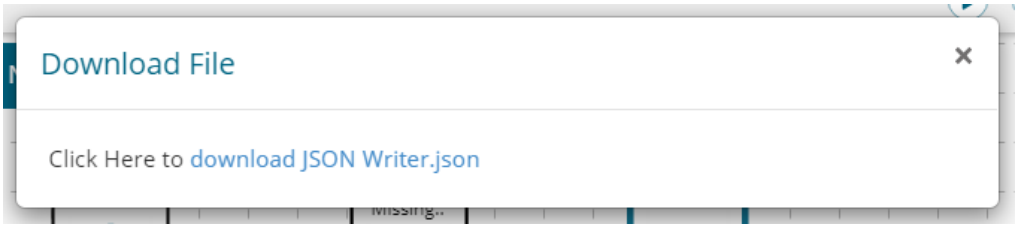


vi) Run the workflow after getting the success message.
vii) The process status gets displayed under the 'Console' tab, and the completion of the process gets marked by green checkmarks on the dragged components.



viii) A pop-up message appears with a link to download the **JSON** file.

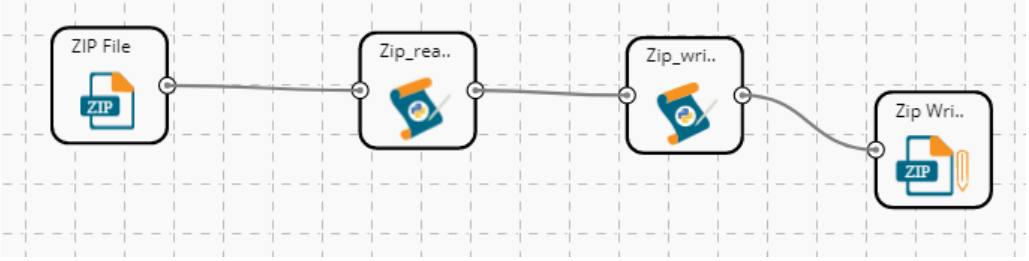
ix) Click the link to download the JSON file.



8.3.3. ZIP Writer

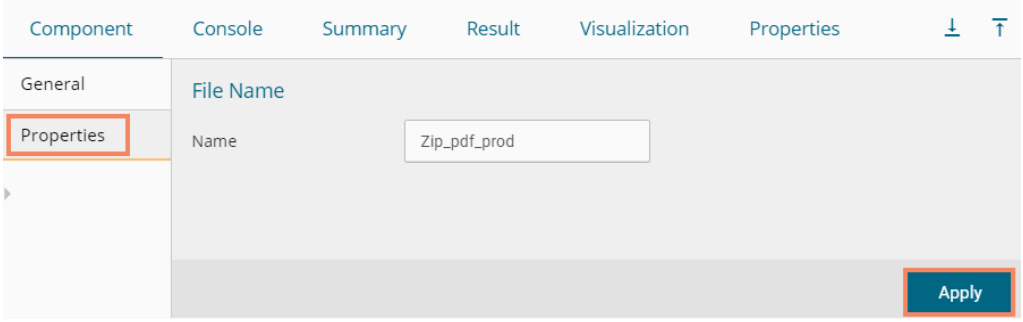
This data writer helps the user to write the processed data into a Zip file.

i) Drag the '**Zip Writer**' from the Data Writer tree-node and connect it to a configured data source and other relevant components to create a workflow.
(The Zip writer requires a relevant script to write the data in the specified Zip file.)



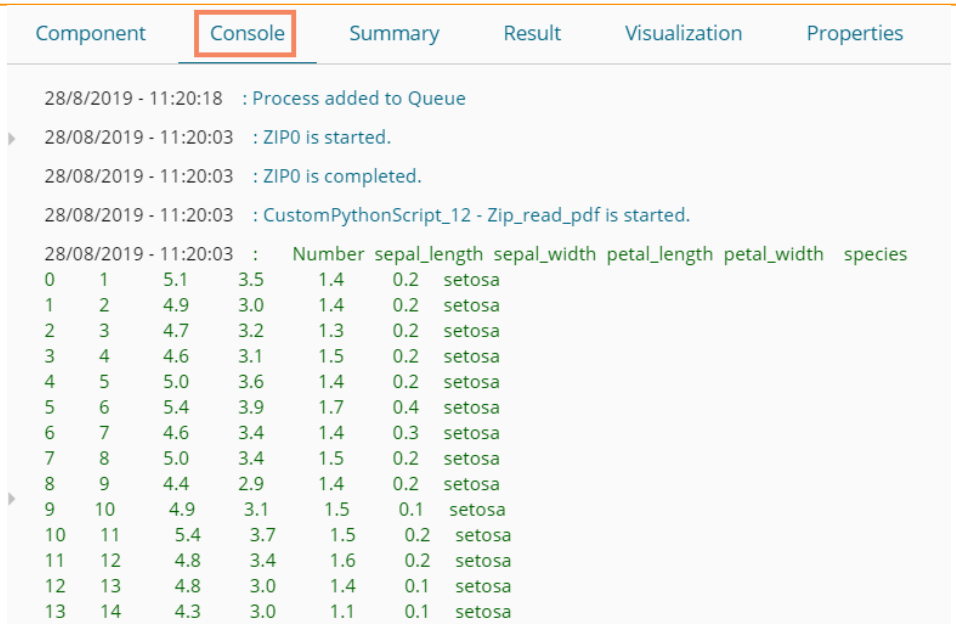
ii) Click the dragged Zip Writer component to get the configuration fields.

iii) Provide the file name to configure the Zip Writer properties.

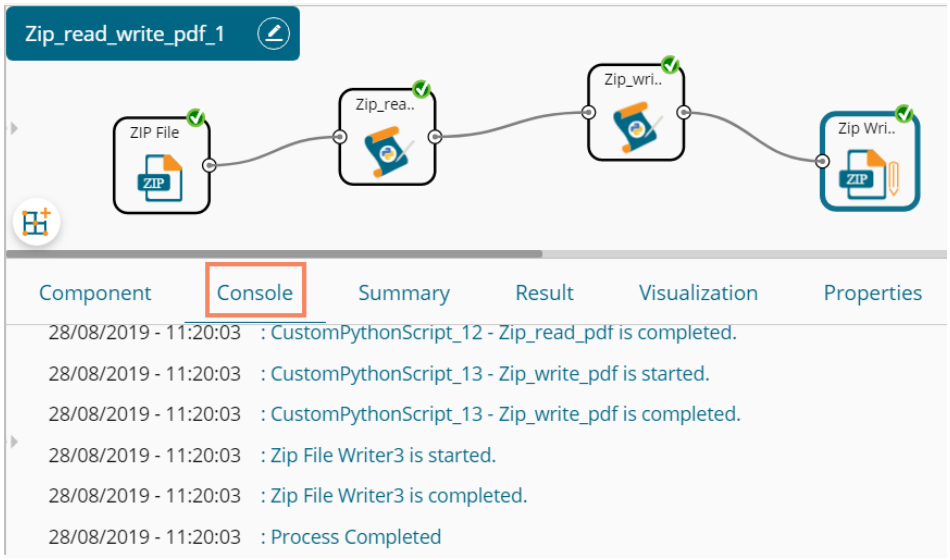


iv) Run the workflow after getting a success message.

v) The Console process displays the process step by step.



vi) The completion of the success process gets indicated by a green checkmark on the top of all components in the selected workflows.

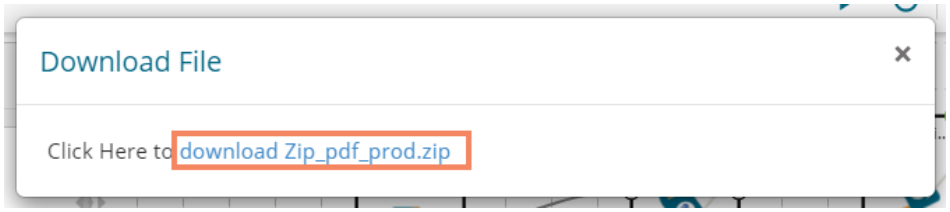


vii) The processed data gets written in a Zip file through the Zip Writer.

viii) After the process gets completed, click on the Zip writer component from the workflow.

ix) The '**Download File**' dialog box appears to download the Zip file.

x) The user can download the Zip file by clicking on the link mentioned in the dialog box.



8.4. Database Writer

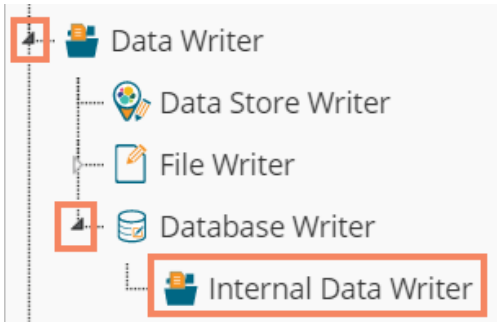
8.4.1. Internal Data Writer

The user can store data in databases like MySQL, MSSQL, and Oracle by Internal Data writer.

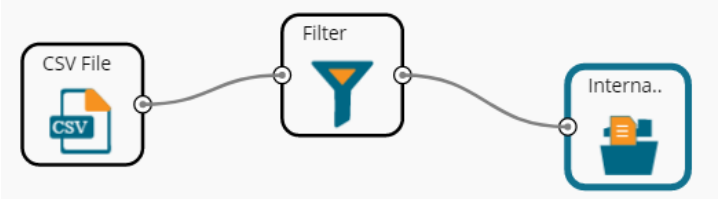
i) Click the **'Data Writer'** tree node option.

ii) Select the **'Database Writer'** option.

iii) Select and drag the **'Internal Data Writer'** component to the workspace.



iv) Drag and Connect the '**Internal Data Writer**' component to a configured data source and other related components to create a workflow.



v) Click the '**Internal Data Writer'** component to access the Component properties. The user gets different '**Properties**' fields based on the selected table operation as described below:

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

i. **Data Connector Name**: All the available data connectors in particular user id get listed. Select a data connector from the drop-down menu.

ii. **Type**: This field is 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 the '**Create New Table**' option from the list

vii. **Table Operation**: Select an option from the drop-down menu

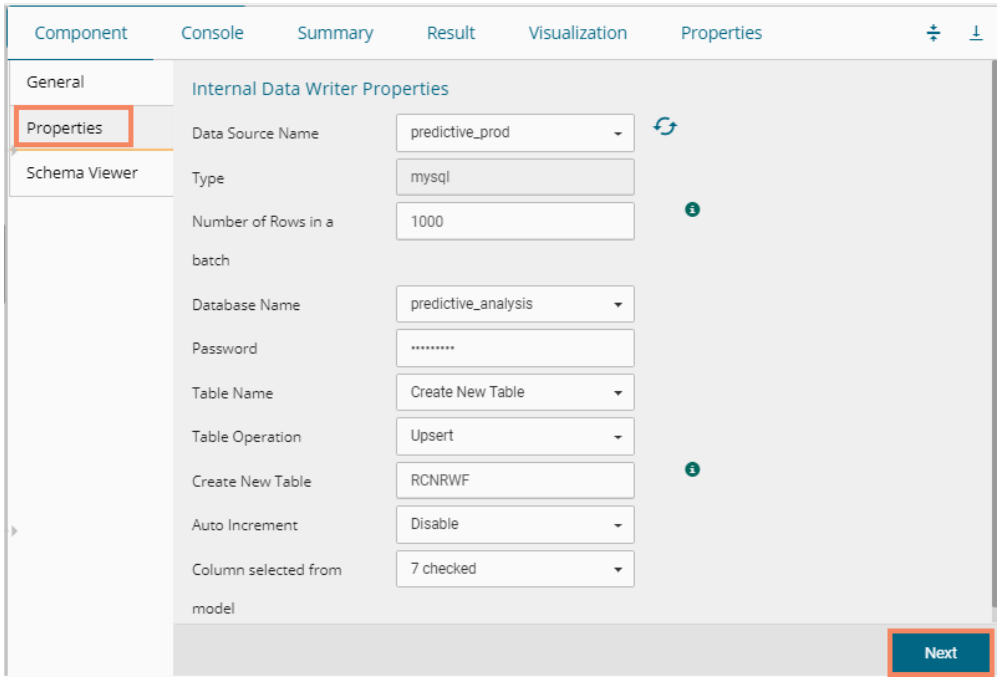
viii. **Create a New Table:** It is an optional field. It appears when the user selects the **'Create New Table'** option from the **'Table Name'** drop-down menu.

ix. **Auto Increment:** Select an option to enable or disable the auto-increment. By enabling this option, a new column gets added to the dataset, and the same column gets selected as the primary key by default.

x. **Auto Increment Label:** Enter a name for the auto-increment label

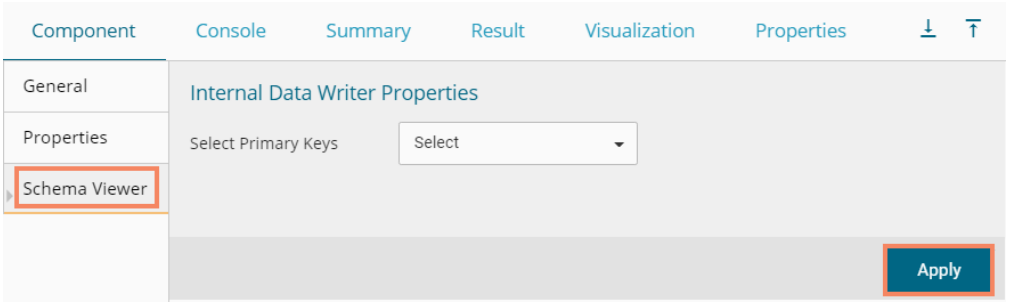
xi. **Column Selected from a model**: Select columns that are needed to be written into the selected database.

vi) Click the '**Next**' option.



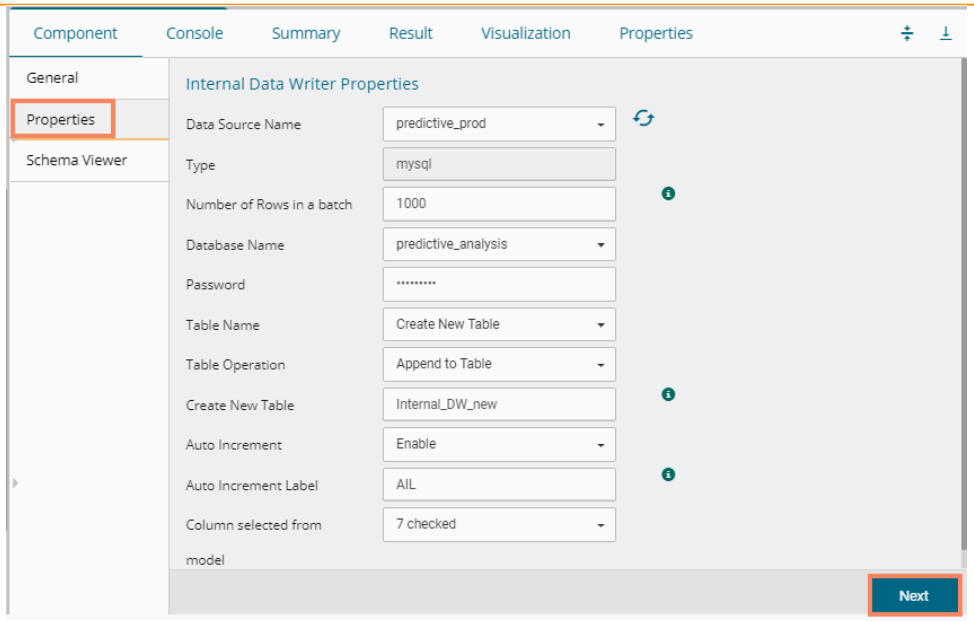
vii) The user gets the '**Schema Viewer**' tab to select the primary keys.

viii) Click the 'Apply' option.

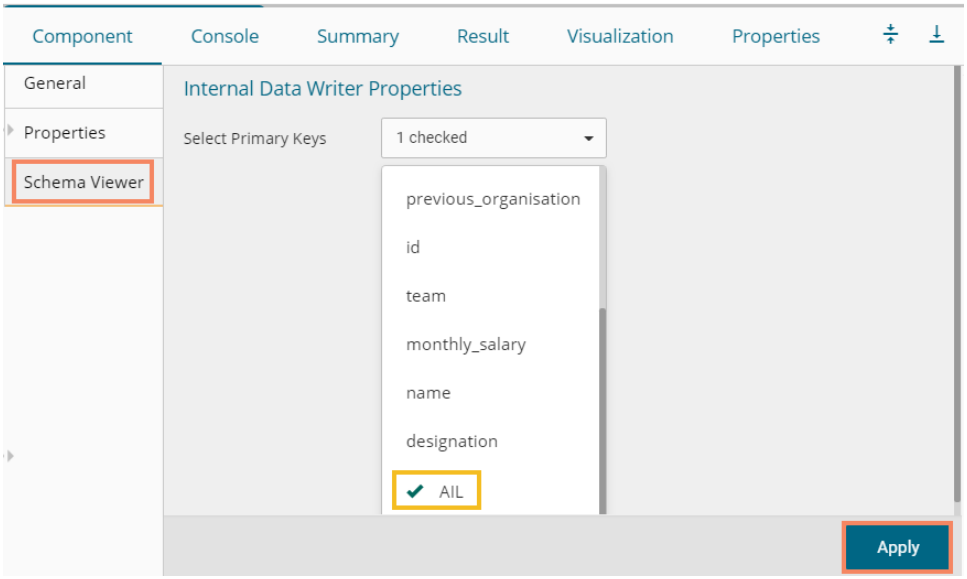


Note: The selected Auto Increment Label appears as the selected Primary Keys by default, if the '**Auto Increment**' option is enabled.

1. Enable the '**Auto Increment**' option from the '**Properties**' tab.
2. Click the '**Next**' option.



3. The Schema Viewer tab opens.
4. The configured Auto Increment Label gets selected as a Primary Key by default.
5. Click the 'Apply' option to save the configuration.

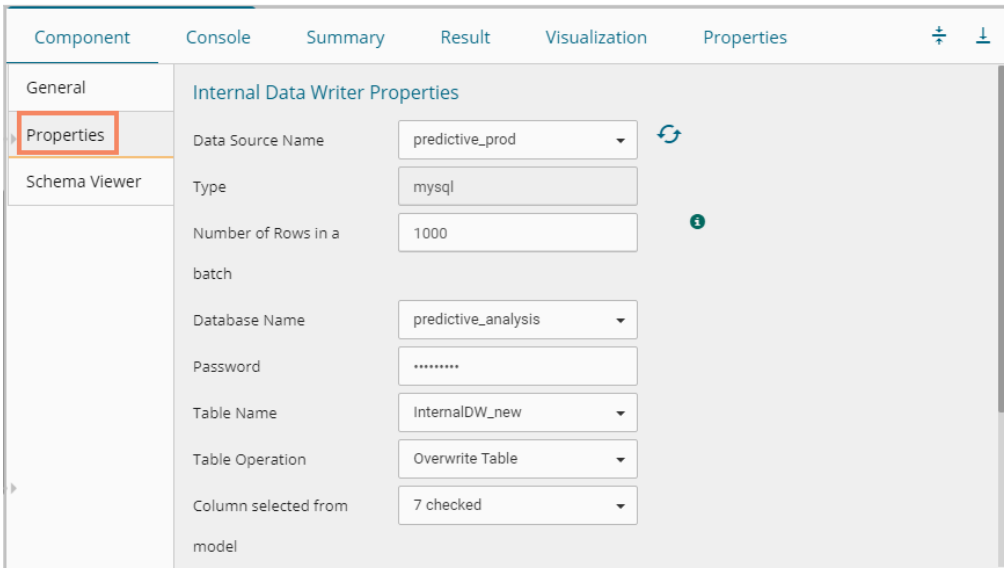


b. Selecting an Existing Table as the 'Table Operation':

i. **Data Connector Name**: Select a data connector from the drop-down menu
ii. **Type**: Displays a type based on the selected data connector
iii. **Number of Rows in a batch**: Enter a number to limit the entries of rows for one batch
iv. **Database Name**: Select a database name from the drop-down menu
v. **Password:** Enter the database password
vi. **Table Name**: Select an existing table name from the drop-down menu
vii. **Table Operation**: Select an option using the drop-down menu. The following are the provided choices:

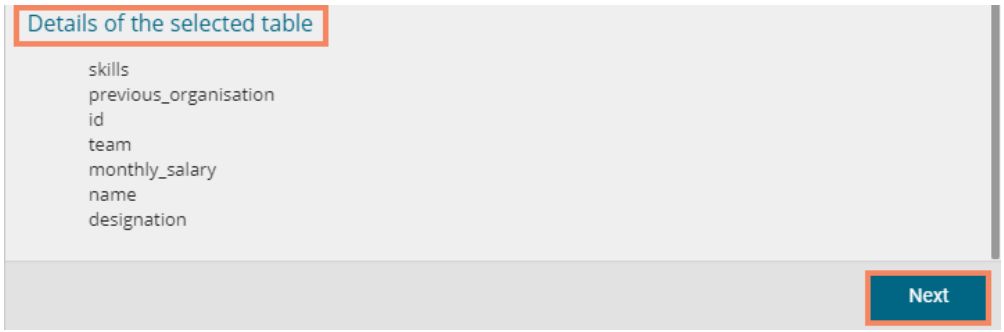
1. Append Table
2. Overwrite Table

viii. **Column Selected from a model**: Select columns that are needed to be written into the selected database.



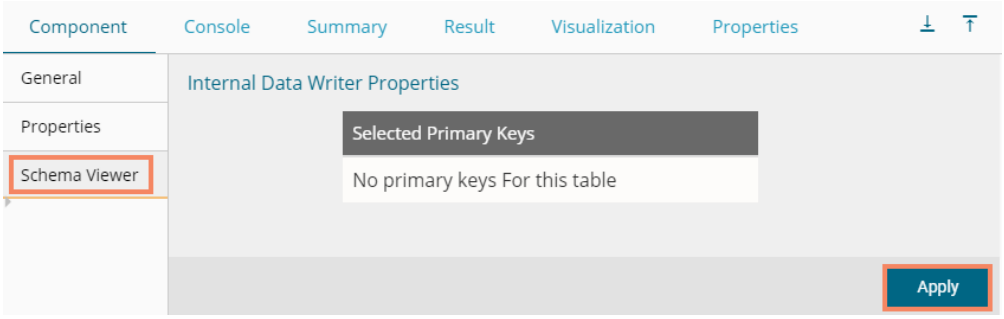
ix. **Details of the Selected table**: Displays column headers from the selected table.

ix) Click the '**Next**' option.



The internal data writer can extract new or changed records while loading data from the MySQL database. The Schema View tab has been added to the internal database writer to extract data using the delta data load type.

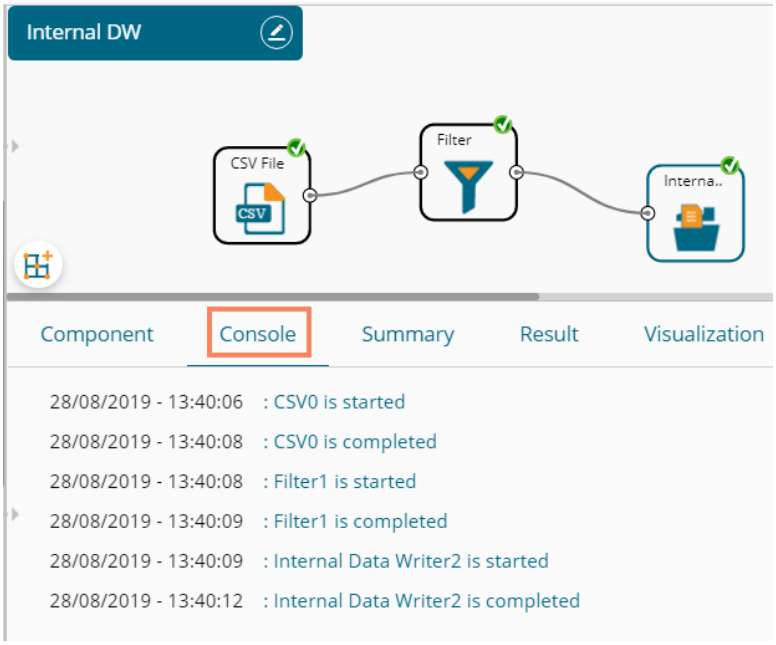
x) The Schema Viewer tab opens displaying the selected Primary Keys (in this case, no Primary Keys is selected).



xi) Click the 'Apply' option.

xii) Run the Workflow after getting the success message.

xiii) The progress of the process gets displayed in the 'Console' tab, and the completion of the process gets marked by the green tick marks on the dragged components.



xiv) The processed data gets saved in the selected database.

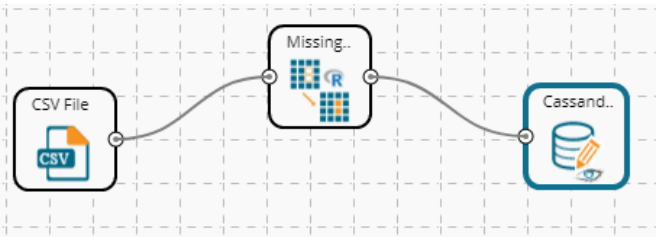
8.4.2. Cassandra Writer

Cassandra Writer can be used to store the data science executions.

i) Open the '**Database Writer**' tree node.

ii) Select and drag the '**Cassandra Writer**' component to the workspace.

iii) Connect the Cassandra Writer to a configured data source or relevant components to create a Workflow.



iv) Click the '**Cassandra Writer**' component to access it.

v) Configure the following Properties details:

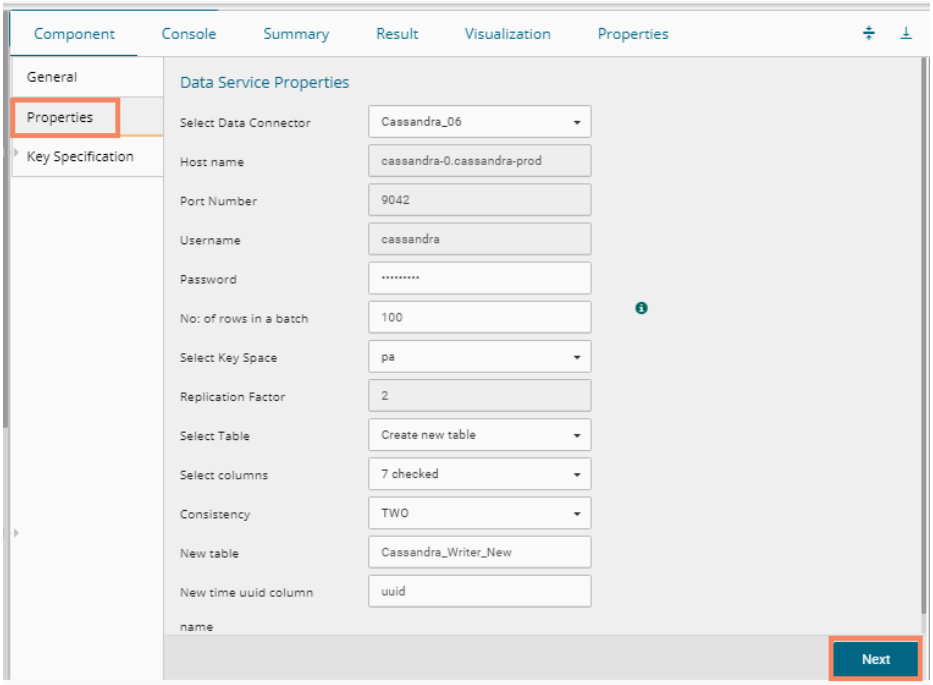
a. Selecting Create New Table as a Table option

i. **Select Data Connector**: Select a data connector using the drop-down menu

ii. **Host Name**: Based on the chosen data connector a hostname gets displayed (the user cannot edit this field)

iii. **Port Name**: The server port number gets displayed (the user cannot edit this field)

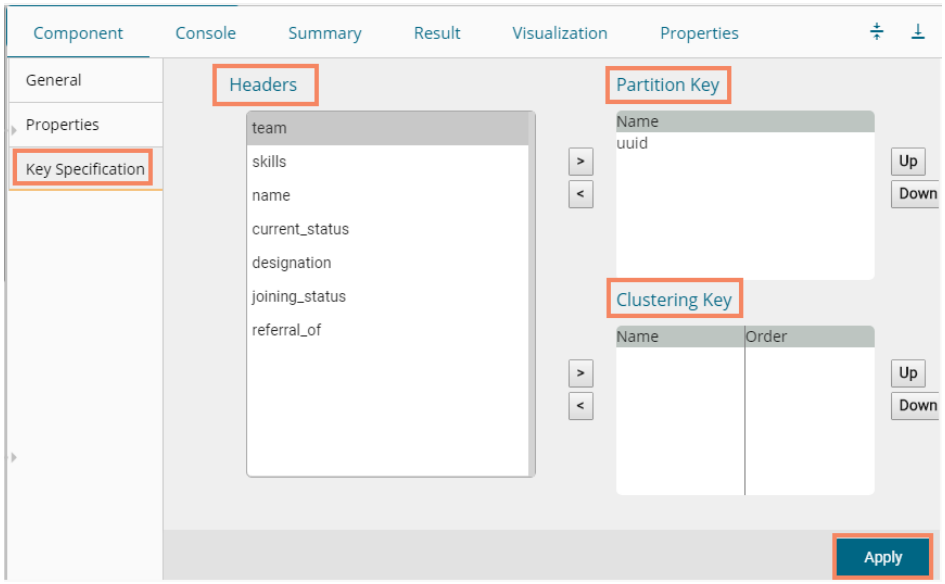
iv. **Username**: The username of the selected connection appears by default. (the user cannot edit this field)
v. **Password:** the database password
vi. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
vii. **Select Key Space**: Select a keyspace using the drop-down menu
viii. **Replication Factor**: The replication factor mentioned in the selected '**Key Space**' get displayed (the user cannot edit this field)
ix. **Select Table**: Select the '**Create a New Table'** option from the drop-down list
x. **Select Columns**: Select the columns that you want to write
xi. **Consistency**: Select an option from the drop-down list
xii. **New Table**: Provide a name for the newly created table
xiii. **New time UUID column name**: Enter a UUID column name
xiv. Click the **'Next'** option**.**



vi) The '**Key Specification**' tab opens.
vii) Configure the following information:
a. **Headers**: All the columns from the data set get listed.
b. **Partition Key (Name)**: The Partition Key determines which node stores the data. It is responsible for data distribution across the nodes.
- The UUID Column name gets displayed under the '**Partition Key**' window.
- The user 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 the '**Up**' or '**Down**' options.

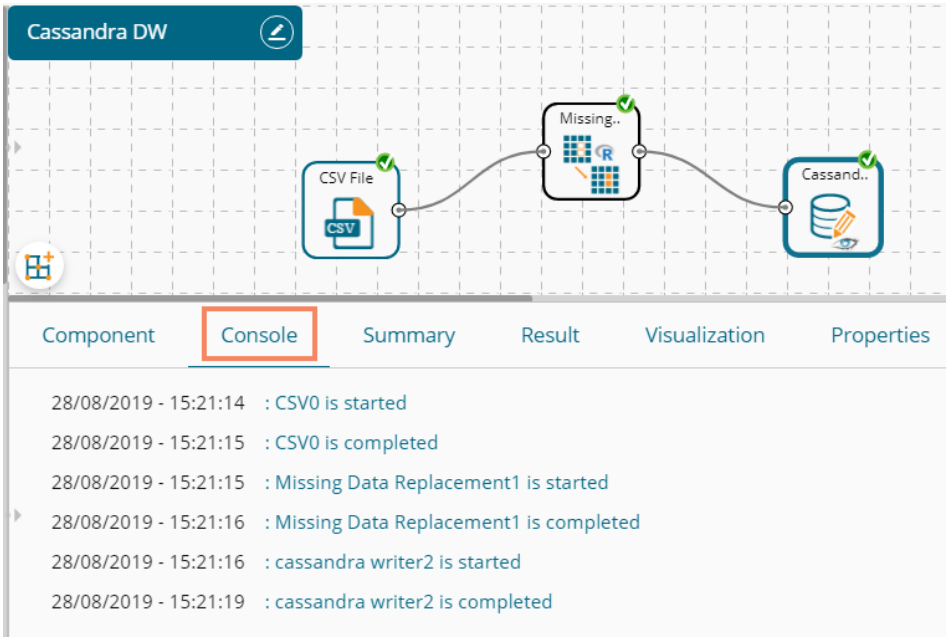
c. **Clustering Key**: The Clustering Key is a storage engine process that sorts data within the partition. It determines per-partition clustering.
- The items listed under the Clustering Key box can be arranged by using the '**Up**' or '**Down**' options.
- The user can select any column from '**Headers**'(Select Column) to the '**Clustering Key**' space.

viii) Click the 'Apply' option.



ix) Run the workflow after getting the success message.

x) The step-by-step process gets displayed under the Console tab. The completion of the process gets marked by the green checkmarks.



Note: The user gets some defined consistency levels while defining the KeySpace, which can be overridden based on the selected replica nodes. The user gets the following options for the Consistency field:

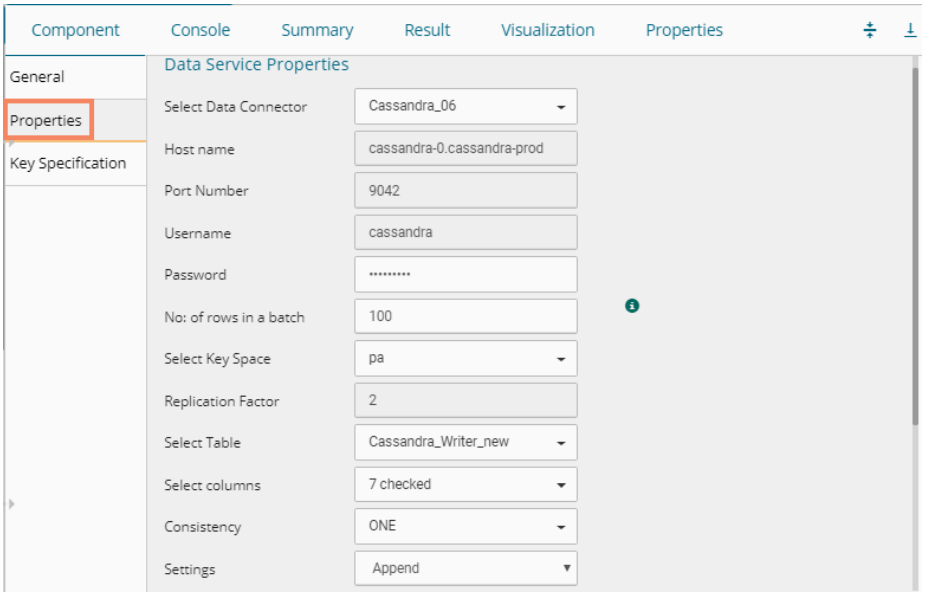
- One
- Two
- Three
- Quarum

or

b. Selecting an Existing Table as Table Operation

Configure the following **Properties** details:

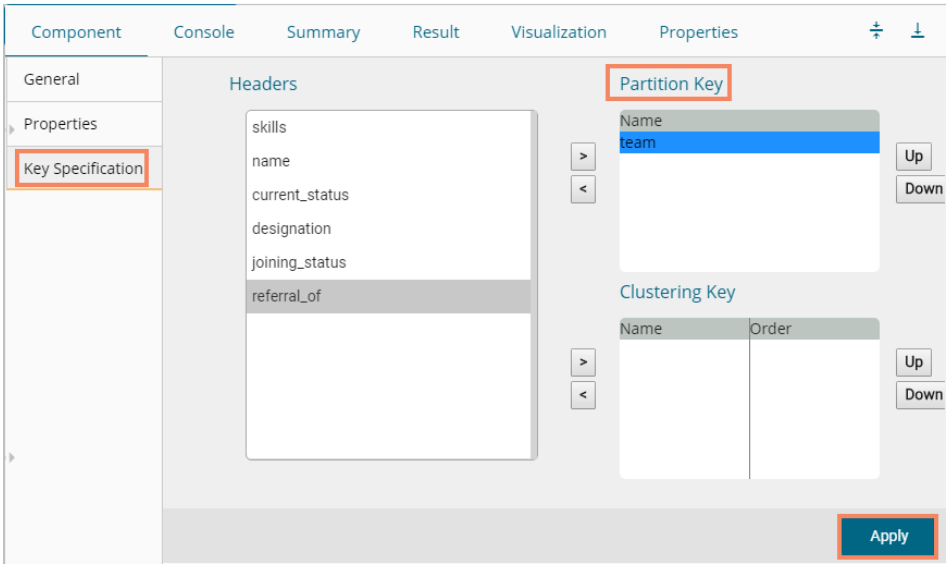
i. **Select Data Connector**: Select a data connector from the drop-down menu
ii. **Host Name**: Enter database server details (from where the user wants to fetch data)
iii. **Port Name**: The server port number
iv. **Username**: Username of the selected connection appears by default (Users cannot edit this field)
v. **Password**: the database password
vi. **No. of rows in a batch:** Enter a number to limit the entries of rows for one batch
vii. **Select Key Space**: Select a keyspace using the drop-down menu
viii. **Replication Factor**: Replication factor in the selected '**Key Space**' gets 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 are provided:
1. Append Table (to select an existing table the selected settings option should be Append)
2. Overwrite Table



xiii. The list of column headers existing in the table gets displayed once the user selects an existing table.
xiv. Click the 'Apply' option.

OR

Configure the Key Specification settings and click the 'Apply' option.

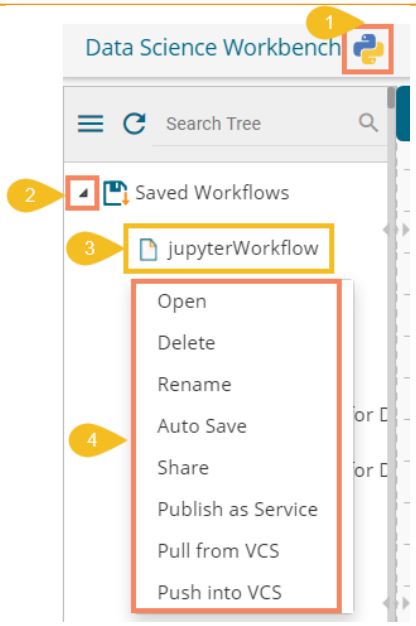


xi) Run the workflow after getting the success message.
xii) The process status gets displayed under the '**Console**' tab. The completion of the Console process gets marked by the green checkmarks on the dragged components.
xiii) The data gets saved in the selected Cassandra Writer.

9. Saved Workflows

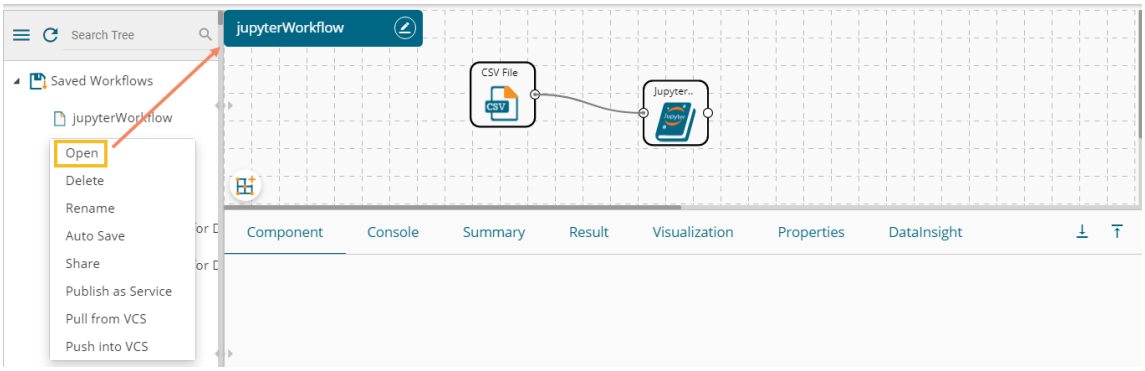
The user can save a workflow by clicking the '**Save**' icon provided on the workspace menu row. All the saved Workflows the selected Workspace gets listed under the '**Saved Workflow**' tree node. This section explains various options assigned to a saved workflow.

i) Navigate to any Data Science Workspace (in this case, the Python Workspace has been selected).
ii) Click the '**Saved Workflow**' tree node.
iii) Select a saved workflow from the list and use a right-click on it.
iv) A context menu opens with various options (As shown below):



9.1. Opening a Workflow

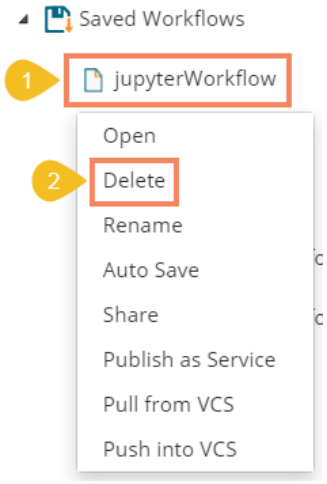
i) Select a workflow from the list of **Saved Workflows** and use a right-click on it.
ii) Select the '**Open**' option from the context menu.
iii) The selected workflow gets displayed in the right pane of the screen.



Note: The workflow name gets displayed on the left side of the workspace menu row while opening a workflow.

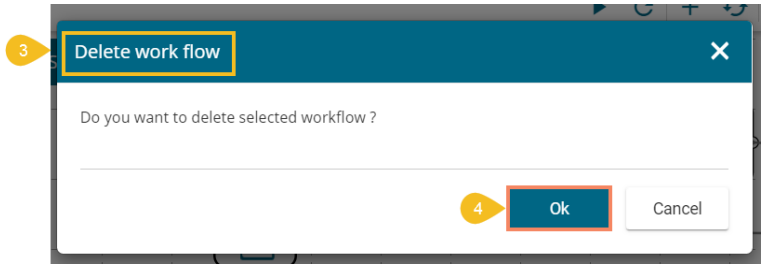
9.2. Deleting a Workflow

i) Select a workflow from the list of **Saved Workflows** and use a right-click on it.
ii) Select the '**Delete**' option from the context menu.



iii) A dialog box appears to confirm the deletion.

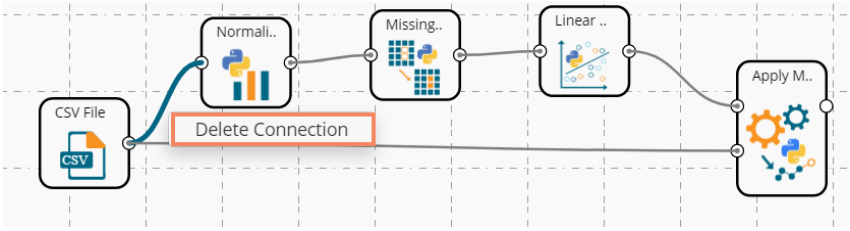
iv) Click the '**Ok**' option.



v) The selected workflow gets removed from the list.

9.2.1. Delete Connection in a Workflow

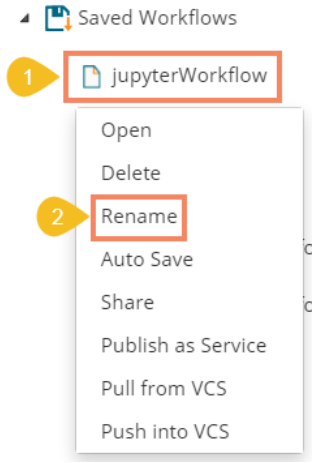
A Right-click on the inter-node connection displays the '**Delete Connection**' option in the workflow. Click the '**Delete Connection**' option to delete a connection.



9.3. Renaming a Workflow

i) Select a workflow from the list of **Saved Workflows** and use a right-click on it.

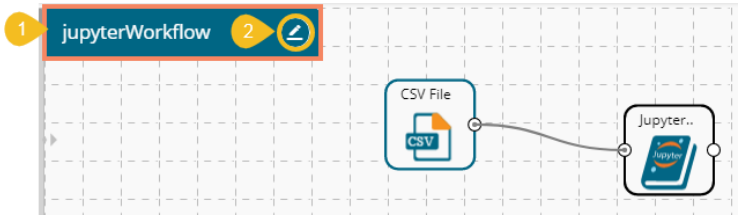
ii) Select the '**Rename**' option from the context menu.



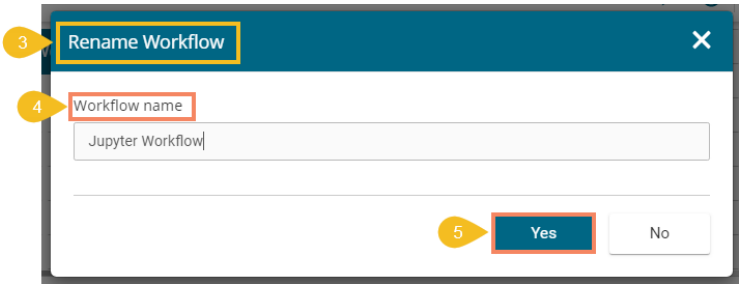
or

Open a Saved Workflow.

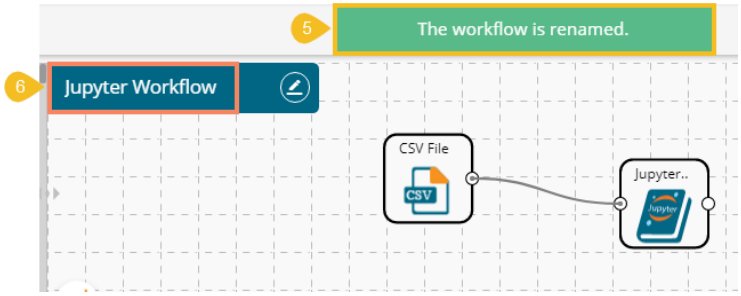
Click the '**Rename**' icon provided next to the workflow name.



iii) The Rename Workflow window opens.
iv) Enter a new/modified name for the workflow.
v) Click the '**Yes**' option.



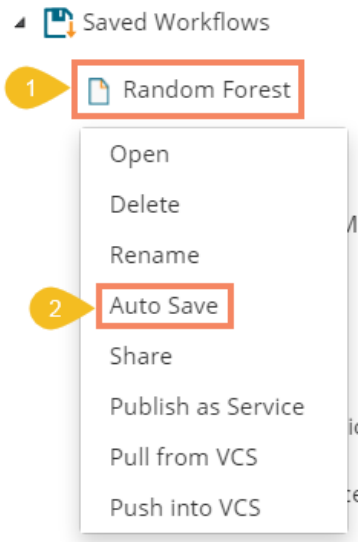
vi) A success message appears.
vii) The workflow gets renamed.



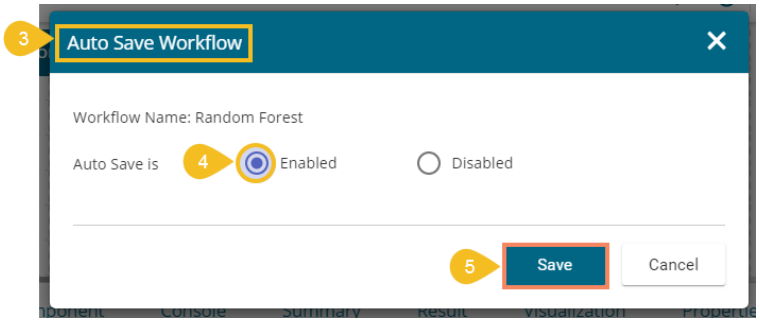
9.4. Auto-Save

The workflow gets auto-saved by enabling this option for a saved workflow.

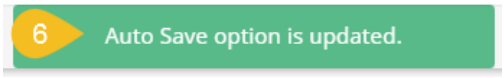
i) Select a workflow from the list of Saved Workflows and use right-click on it.
ii) Select the '**Auto Save'** option from the context menu.



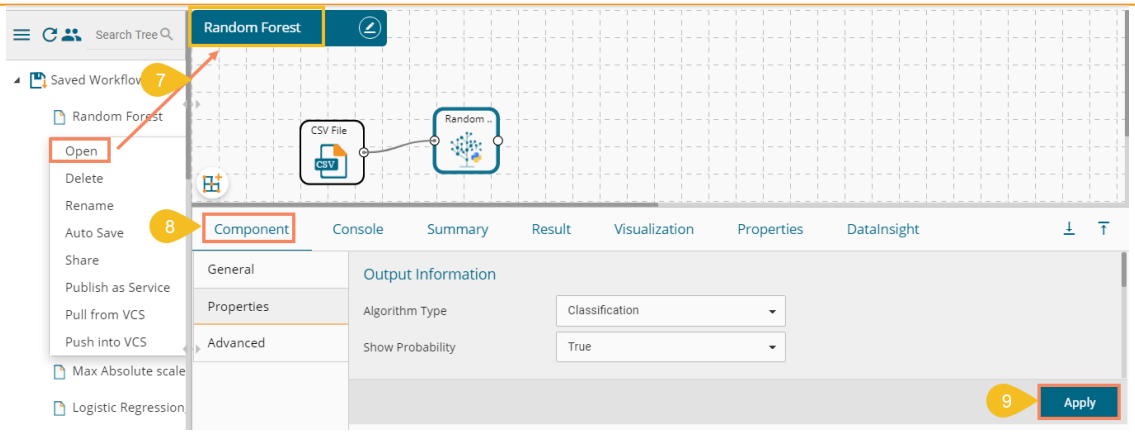
iii) The '**Auto Save Workflow'** window opens.
iv) Select the '**Enabled**' option by using the checkbox.
v) Click the '**Save**' option.



vi) A message appears to inform the user that the Auto-Save option is updated.

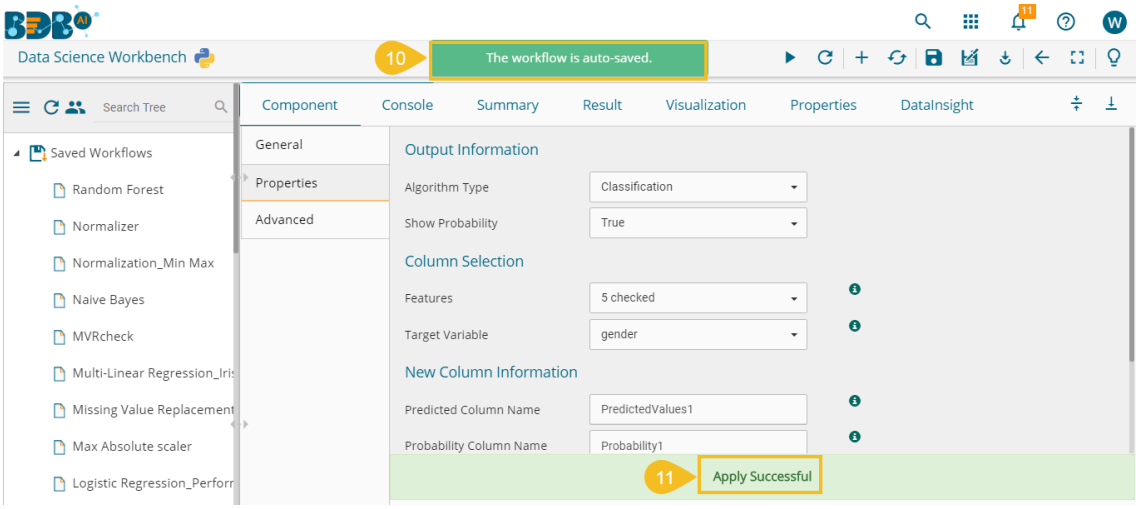


vii) Open the Workflow.
viii) Edit some Component information.
ix) Click the '**Apply**' option.



x) A message confirms that the edited information has been applied.

xi) Another message on the top appears to inform the user that the Workflow is auto-saved.



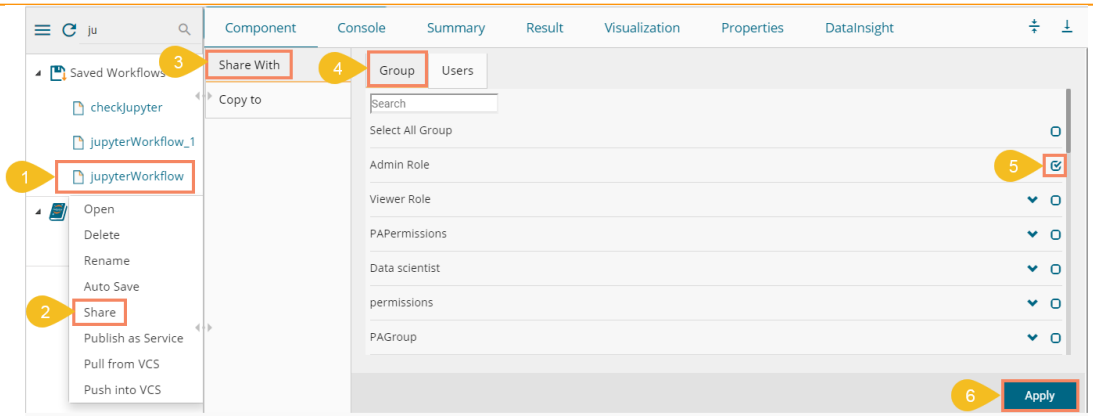
9.5. Sharing a Workflow

The user can share a saved workflow with other users and groups through this option.

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 the file get transferred to all the users with whom the file has been shared.

   i) Select a workflow from the list of **Saved Workflows** and use right-click on it.
   ii) Select the '**Share**' option from the context menu.
   iii) The '**Share With**' option gets displayed (by default)
   iv) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group get listed. You can exclude the users by not selecting them from the group.
      b. The users can also get excluded by not selecting a username from the list when the '**Users**' option has been selected.
   v) Select a specific group or user from the list by putting a checkmark in the given box.
   vi) Click the '**Apply**' option.

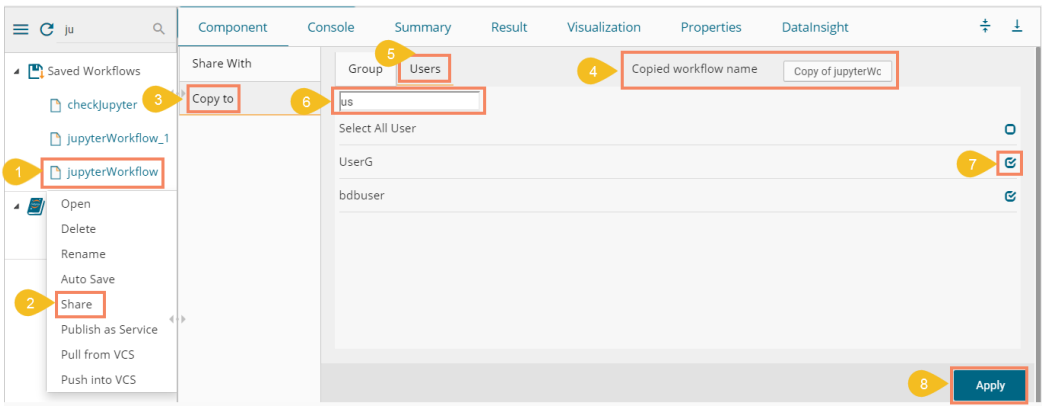


vii) A success message appears.

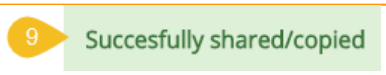
viii) The selected workflow gets shared with the chosen user(s)/group(s).

2. **Copy To**: This option creates a copy and shares the copy with the selected users and user groups. Any change to the original file after sharing does not display for the users that received the shared file via the '**Copy To**' method.

   i) Select a workflow from the list of **Saved Workflows** and use right-click on it.
   ii) Select the '**Share**' option from the context menu.
   iii) Select the '**Copy To**' option.
   iv) The Workflow name gets displayed with the '**copy of**' prefix.
   v) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group get listed. The users can be excluded by not selecting them from the group.
      b. The user can also get excluded by not selecting a username from the list when the 'Users' option has been selected.
   vi) Use search space to search for a specific user.
   vii) Select a specific group or user from the list by putting a checkmark in the given box.
   viii) Click the '**Apply**' option.



ix) A success message appears.

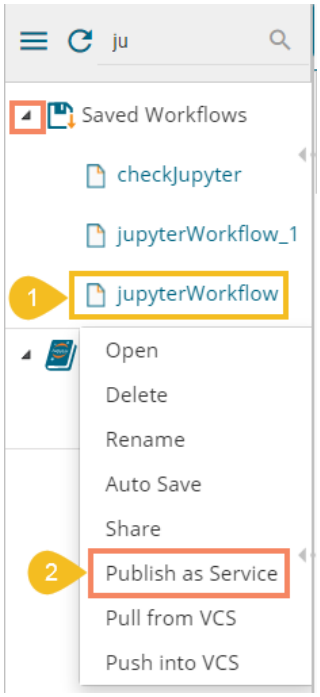


x) The copied workflow gets shared with the chosen users/groups.

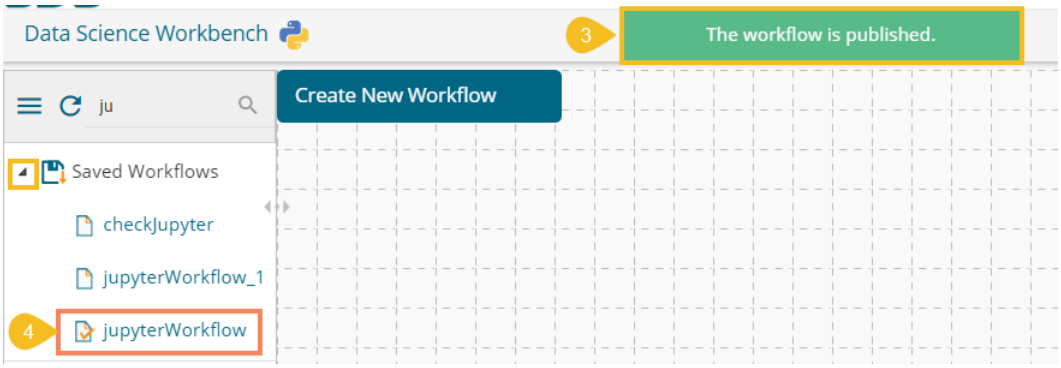
9.6. Publish a Workflow as Service

The Data Science workflow can be deployed to the BDB Dashboard Designer as a service.

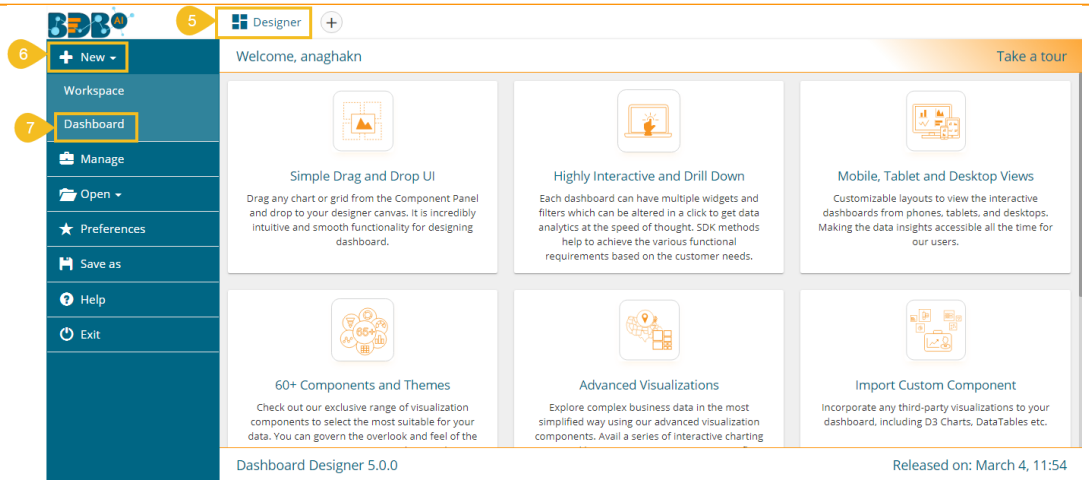
i) Select a Workflow from the list of **Saved Workflows** and use a right-click on it.
ii) Select the '**Publish as Service**' option from the context menu.



iii) A success message appears to assure that the workflow has been published.
iv) The published workflows get marked by a checkmark (as displayed below).

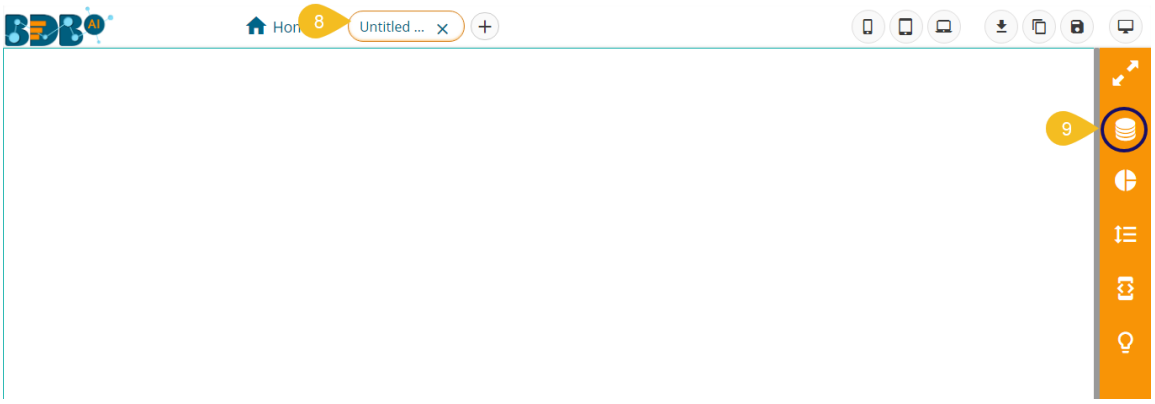


v) Navigate to the Dashboard Designer homepage.
vi) Click the '**New**' option.
vii) Click the '**Dashboard**' option.



viii) The Dashboard canvas opens.

ix) Click the '**Data Connectors**' icon to display all the available data connectors.



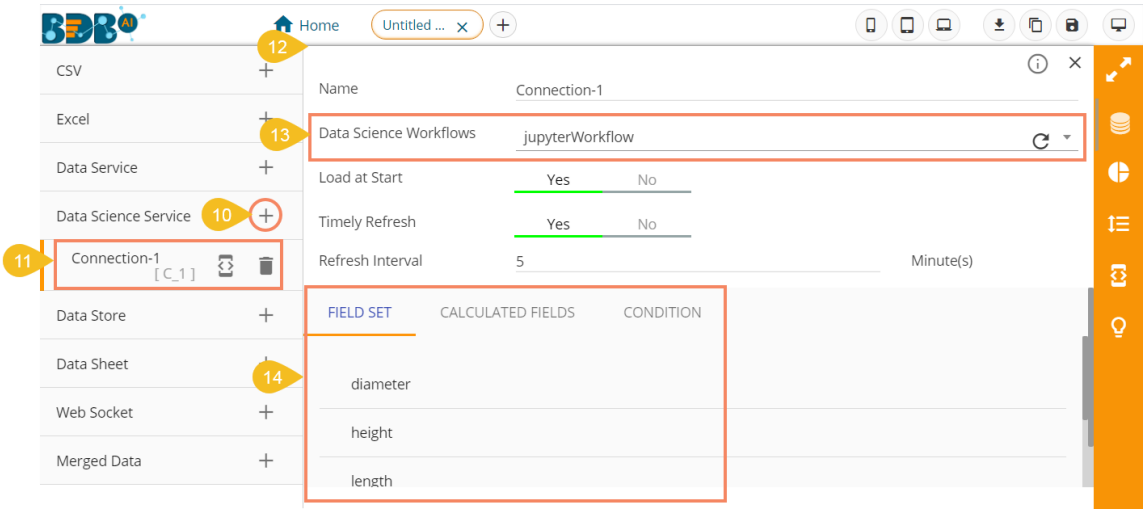
x) Click the '**Create New Connection**' option provided Next to the '**Data Science Models'** option on the Data Connector page.

xi) A new connection gets created and added below.

xii) The connection-specific details get displayed on the right.

xiii) Select the deployed Data Science workflow as a data source via the drop-down menu.

xiv) After selecting the Data Science Workflow the FIELD SET tab displays the available fields.



xv) Once the data connection is established the selected predictive workflow can be used as a data source to the Dashboard Designer.

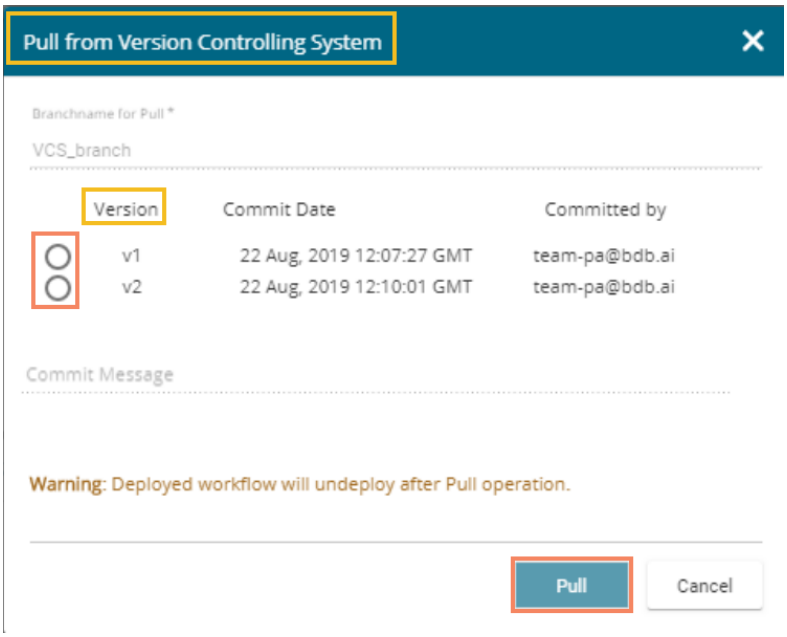
Note:

b. If a deployed Predictive Workflow has a summary, it can be viewed using the Dashboard Designer tool.
c. If the model included in the selected saved NN Workflow contains NumPy script, then after the successful deployment of that workflow still users cannot create a dashboard based on it.
d. The dashboards created based on the deployed Python workflows also support Bokeh charts.

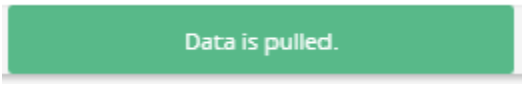
9.7. Pull from VCS

The option helps to pull the workflow from the Version Controlling Service.

i) Select a workflow from the Saved Workflow list.
ii) Click the '**Pull from VCS**' option.
iii) A window opens like below:
   a) The branch name for pull comes pre-written.
   b) The details of the existing version get displayed from where the user can select the desired version using the radio button.
   c) Click the '**Pull**' option.



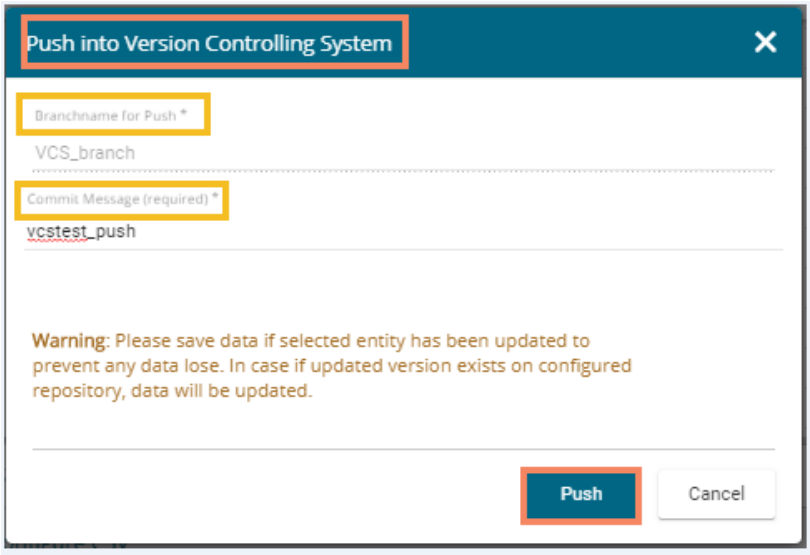
d) A success message appears to indicate that the selected entity has been pulled from the VCS.



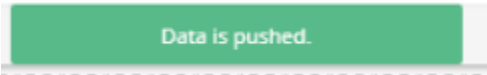
9.8. Push into VCS

The option helps to push the workflow into the Version Controlling Service.

i) Select a workflow from the Saved Workflow list.
ii) Click the '**Pull from VCS**' option.
iii) A window opens like below:
   a) The branch name for push comes pre-written.
   b) Provide Commit message (it is mandatory)
   c) Click the '**Push**' option.



d) A success message appears to indicate that the selected entity has been pushed into the VCS.



Note: At present, the **Pull from VCS** and **Push into VCS** options are available only for the Python workflows.

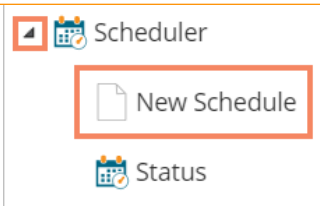
10. Scheduler

The Scheduler component helps to schedule the Data Science workflows as per the requirement.

10.1. New Schedule

This section explains the steps to schedule a new job. Scheduling a new job is a continuous step by step process as described below:

i) Navigate to the Predictive homepage.
ii) Click the '**Scheduler**' tree node.
iii) Two options get displayed:
   a. New Scheduler
   b. Status
iv) Select the '**New Schedule**' option from the menu.

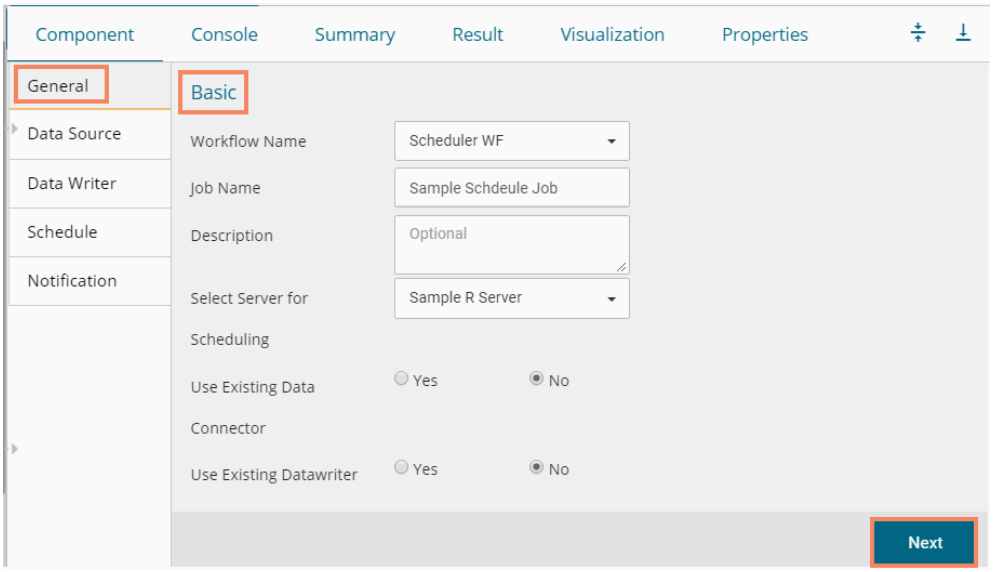


v) The '**General**' tab opens.

10.1.1. Configuring General Tab

i) The '**General**' tab opens (by default) by clicking the New Schedule.
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.

   (Only Data service and Data Store data connectors can be allowed to use an existing data connector option.)
   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 the '**Next**' option.



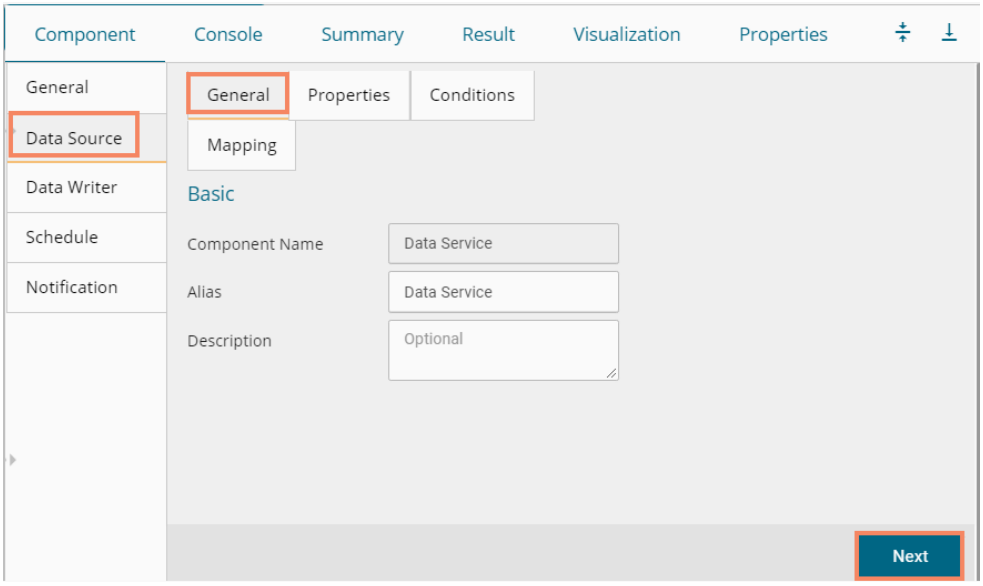
iv) The '**Data Source**' tab opens.

10.1.2. Configuring Data Source

Provide the required information to configure a data source:
i) The '**General**' fields to configure the data source appears by default.
ii) The user can fill in the required fields:

a. Component Name: A default name provided for the component.
b. Alias Name: The user can enter a name for the component.
c. Description: Users can describe the component (optional).

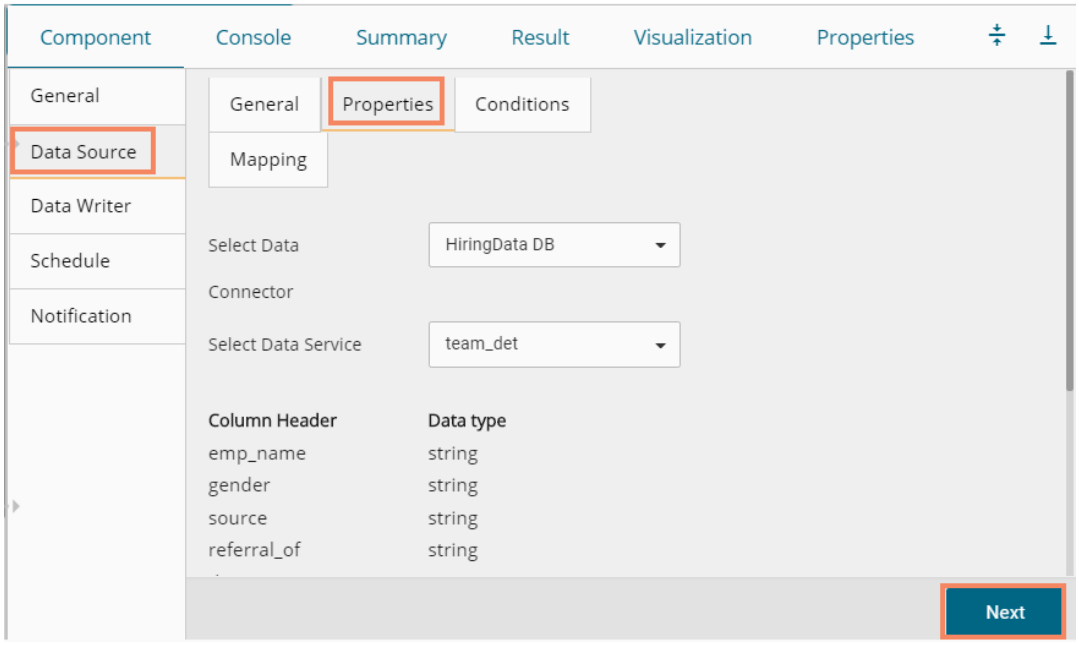
iii) Click the '**Next**' option.



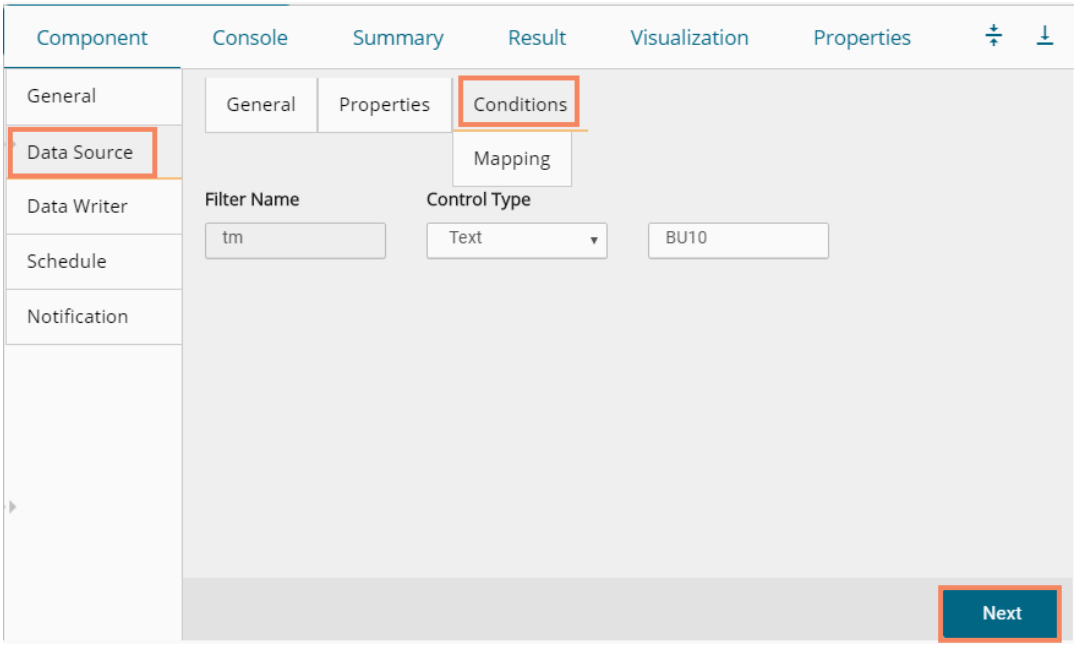
iv) The user gets redirected to the '**Properties**' fields.

v) Configure the following fields (to configure a new data source):
a. **Select Data Connector**: Select a data connector from the drop-down menu
b. **Select Data Service**: Select a data service from the drop-down menu
c. Based on the selected data service the below-given columns get displayed
i. Column Header
ii. Data Type

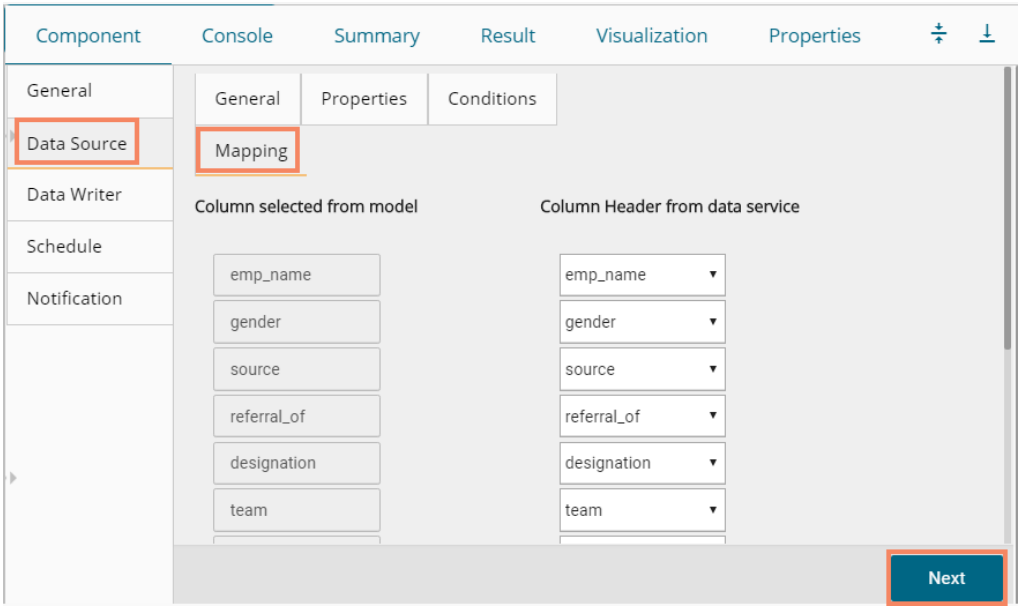
vi) Click the '**Next**' option.



vii) The '**Conditions**' tab opens (If conditions are available, else the user gets redirected to the 'Mapping' page).
viii) Configure the required '**Conditions**' fields.
ix) Click the '**Next**' option.



x) The user gets redirected to the '**Mapping**' tab.
xi) Configure the column header information from the data service that is used for the selected model columns.
xii) Click the '**Next**' option.

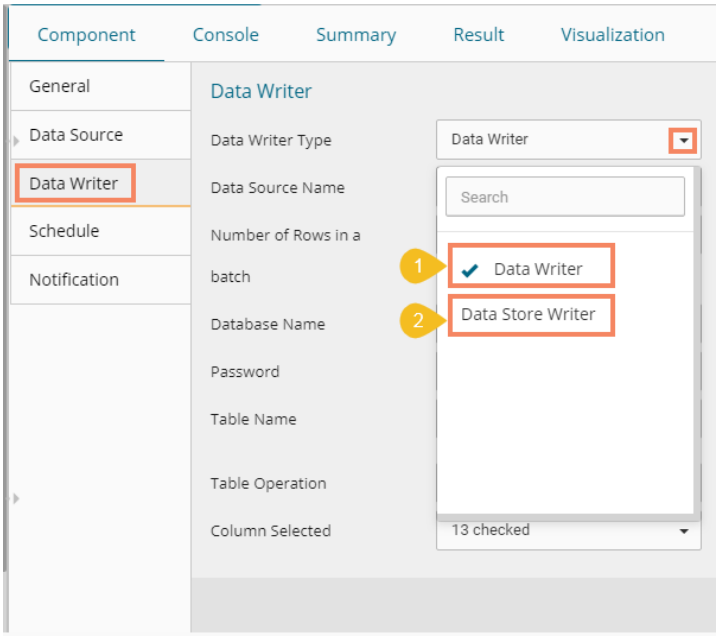


xiii) The '**Data Writer**' tab opens.

Note: The user can skip this step if the existing data connector is used. The user needs to configure the data source.

10.1.3. Configuring a Data Writer

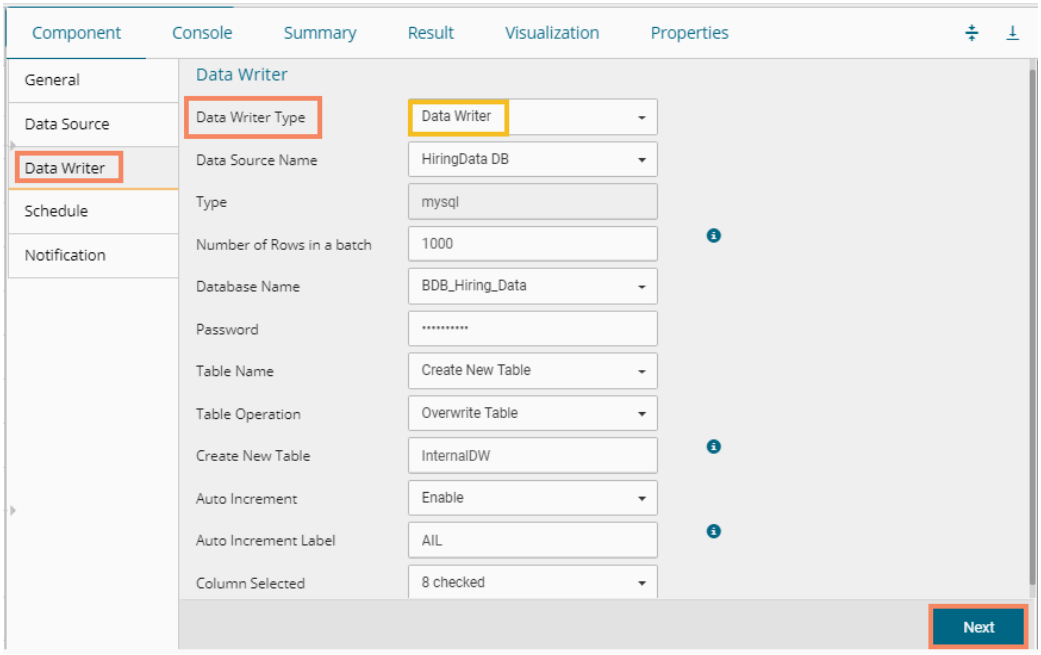
The Data Writer fields are reliant on the selected data writer types. The scheduler is provided with two kinds of data writers: 1. Data Writer, and 2. Data Store Writer.



1. Data Writer

i) Fill in the required details to configure a database writer.

ii) Click the '**Next**' option.

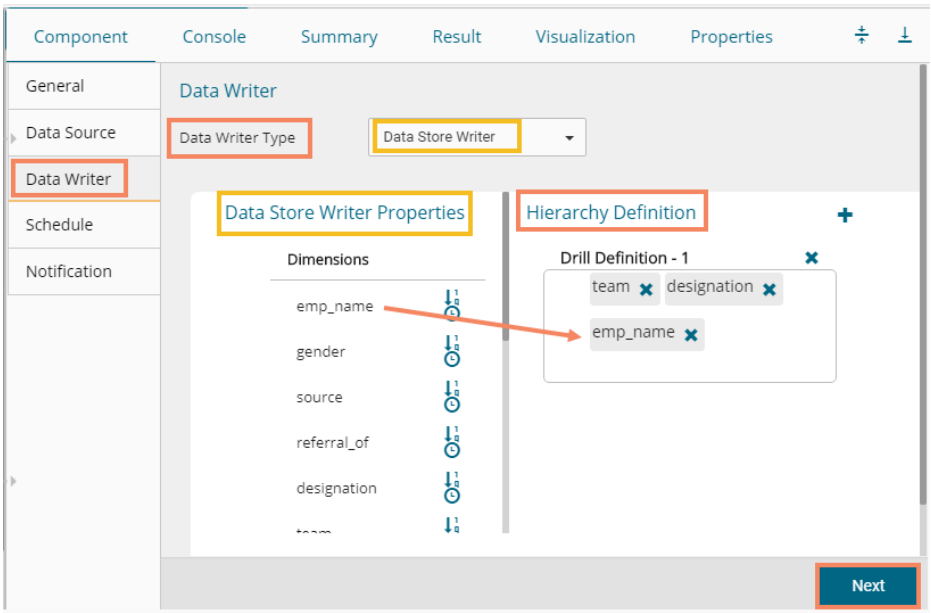


iii) The '**Schedule**' tab opens.

2. Data Store Writer

Users can directly use the predictive workflows to create Business Stories if the workflows are written using the Elastic Search Writer.

i) Select '**Data Store Writer**' as a Data Writer Type to schedule a Predictive workflow.
ii) The Data Store Writer Properties appears.
iii) Drag and drop the required dimensions to define a hierarchical drill.
iv) Click the '**Next**' option.



v) The '**Schedule**' tab opens.

Note: The user can skip this step if the existing data writer has been marked to use.

10.1.4. Scheduling a New job

The user can select a time to schedule a new job using this section. The refresh interval option appears as per the selected scheduling time.

i) **Start Date**: Select a start date and time for the scheduled job (It should be higher than **the Current System Date and Time)**
ii) **Select a Job Refresh Interval option**:
E.g., When the selected time range is '**Hourly**,' the selected interval option can be as described below:
Every_hour: Selecting this option refreshes the scheduled job after every selected interval.
OR
At: Selecting this option refreshes the scheduled job at the selected hour.

iii) **Start Time:** Select a start time higher than the current system time.
iv) **End Date**: Select an end date and time for the scheduled job (It should be higher than the Start date and the Current System Date and Time).
v) **Run Now**: Select this option to run the scheduled job on Applying.
vi) Click the '**Next**' option.
vii) The '**Notification**' tab opens.

10.1.4.1. Job Refresh Intervals Details

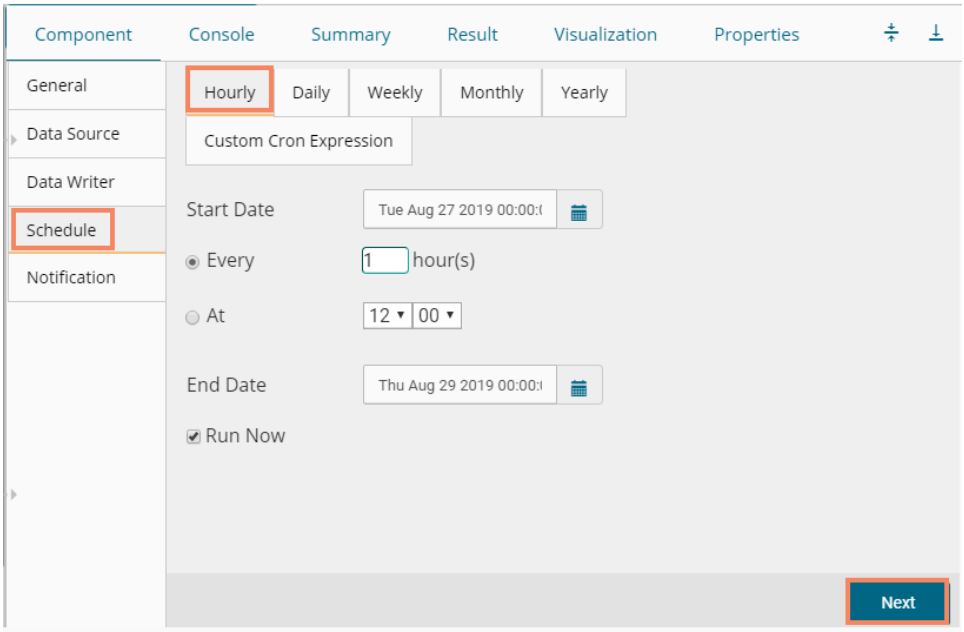
- **Hourly**: By selecting this option, the user can schedule the job on an hourly basis.

1. Select a specific hour by using the below-given options:

Every_hour: Selecting this option refreshes the scheduled job after the selected hourly interval.

OR

At: Selecting this option refreshes the scheduled job at the selected hour.



- **Daily**: By selecting this option, the user can schedule the job daily.

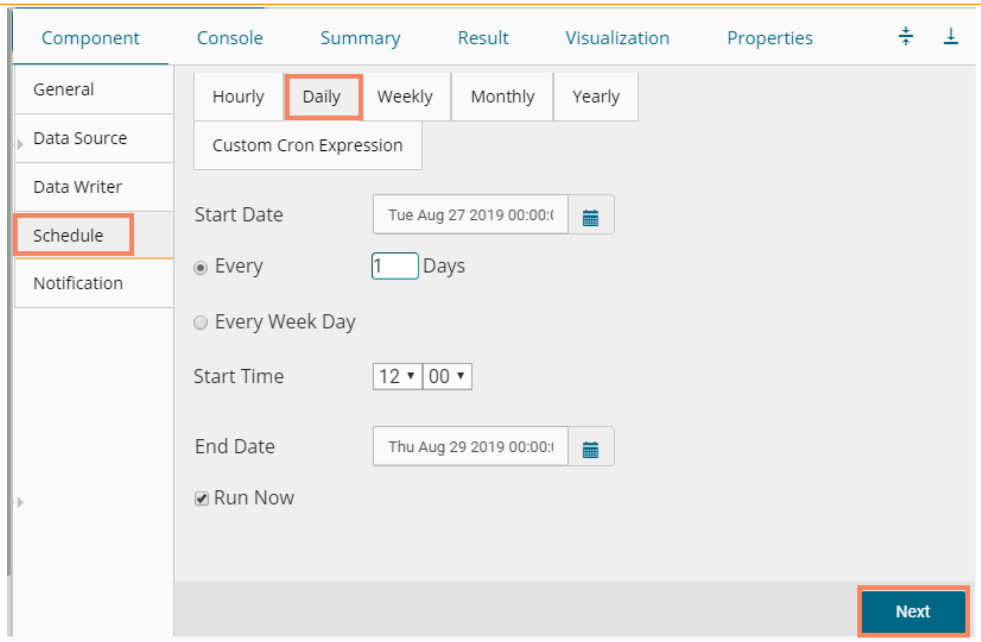
1. Select a specific day by using the below-given options:

Every_ Days: the scheduled job gets refreshed after every selected number of days. E.g., if 2 is selected then; the scheduled job gets refreshed every alternate day at the set time.

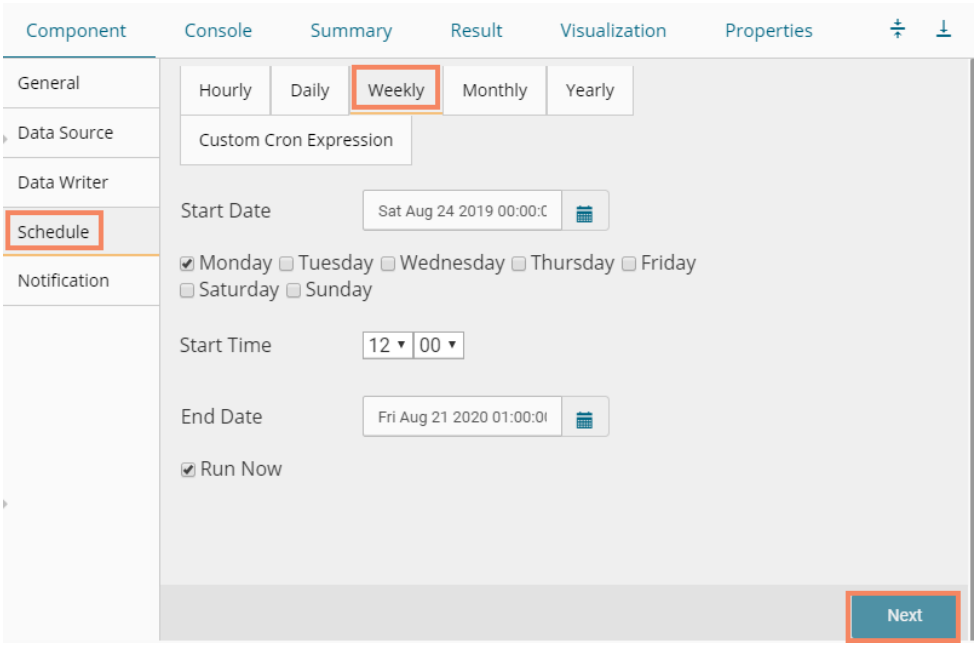
OR

Every Week Day: the scheduled job gets refreshed daily till the end date.

1. Select the Start time.



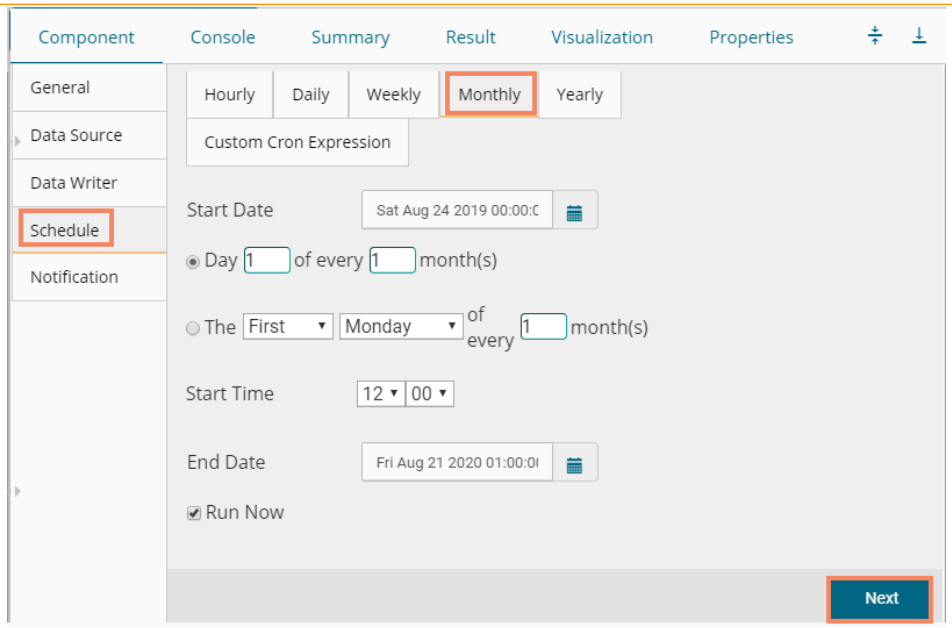
- **Weekly**: By selecting this option, the user can schedule the job every week. Select a day **or** days of the week when the scheduled job can be refreshed.



- **Monthly**: By selecting this option, users can schedule the job every month. This time the range can be used to set schedule refresh for more than a month.
  Select a specific day of the month by using the below-given options:
  E.g., Set monthly refresh interval (E.g., the first day of every month)

  OR

  Set a specific day after the desired monthly interval (the first Monday of the every month)



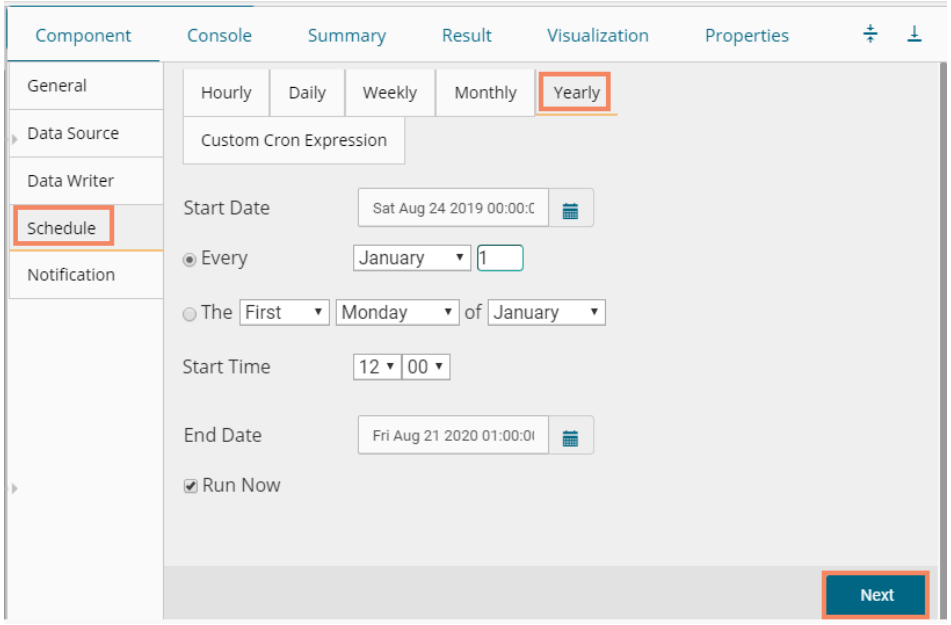
- **Yearly**: By selecting this option, users can schedule the job every year. This time range is provided for jobs that run for more than one year.

Select a specific day of the month by using the below-given options:

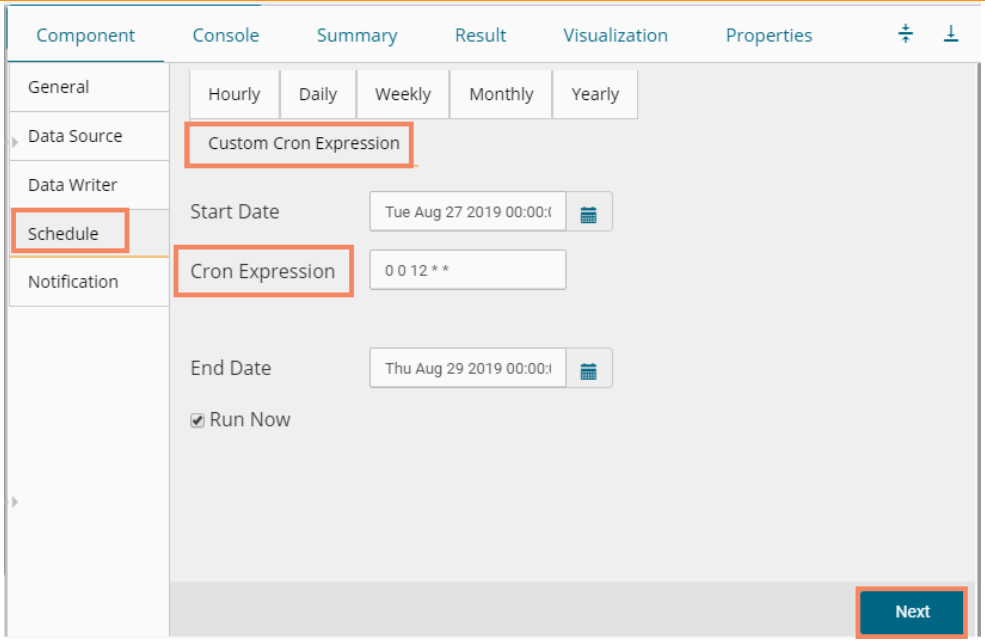
Set a date for any month (E.g., The 1st January of every year till it approaches the end date)

Or

Select a day of any month (E.g. The 1st Monday of January every year until it approaches the end date)



- **Custom Cron Expression:** The user can schedule a more flexible and customizable schedule runs by using the 'Custom Cron Expression' option. The scheduled workflow can be more specific with the custom cron expression that supports timing up to minutes and seconds. Users need to enter a valid Cron Expression in the given field.



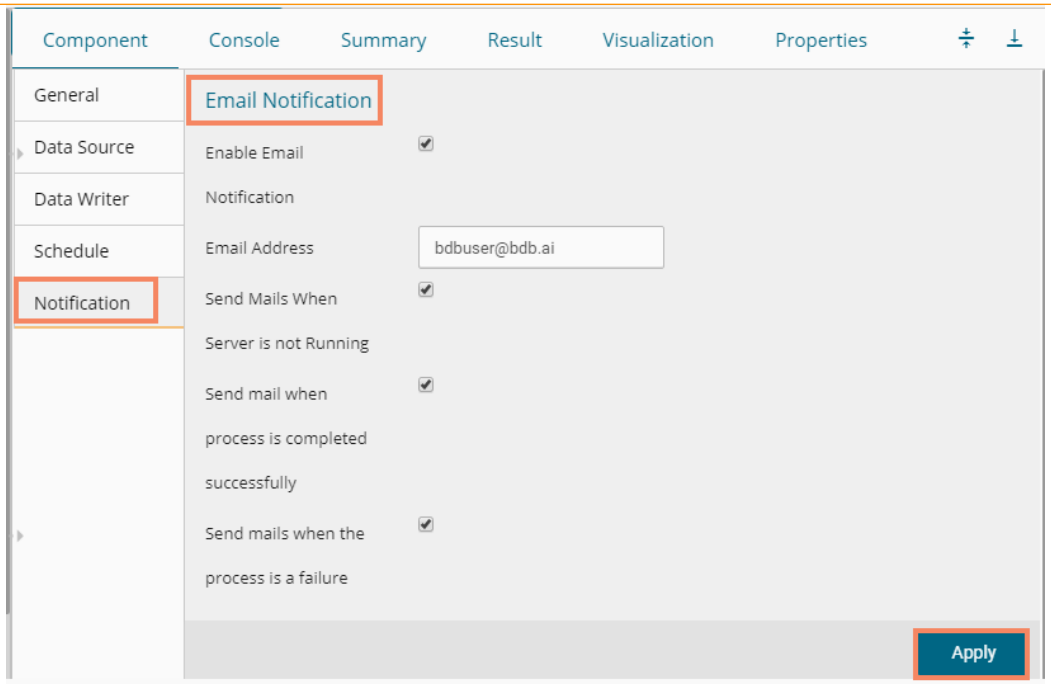
Note: By selecting the '**Use Existing Data Connector**' and '**Use Existing Data Writer**' options the '**Schedule**' tab gets displayed immediately after the '**General**' tab.

10.1.5. Notification

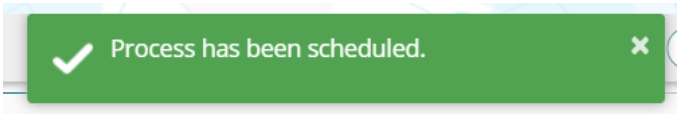
The '**Notification**' tab opens to configure the email settings to get a notification.

i) Configure the below-given fields:
   a. **Enable Email Notification**: Use a checkmark in the box to enable email
   b. **Email Address**: Enable this option by using checkmarks in the box
   c. **Send Mail when Server is not running**: Users can checkmark in the box to enable this option. By enabling this option, the user gets an email when the R server is not running.
   d. **Send Mail when Process is Completed Successfully**: Users can put a checkmark in the box to enable this option. By enabling this option, the user gets mail after the process is completed.
   e. **Send Mail when the Process is a Failure**: Users can checkmark in the box to enable this option. By enabling this option, the user gets an email when the process fails.

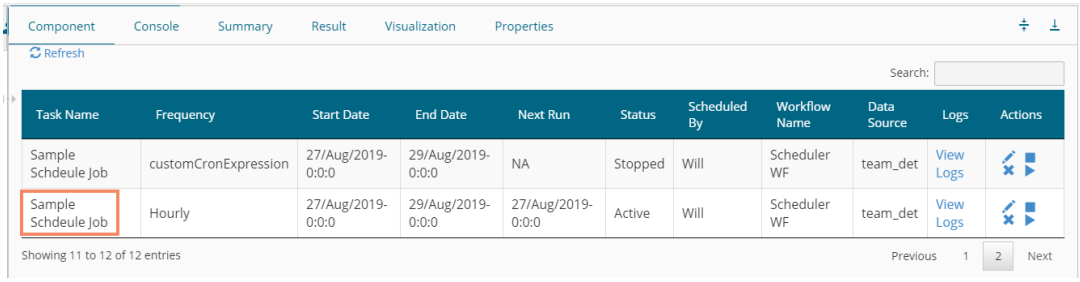
ii) Click the '**Apply**' option.



iii) A success message appears.



iv) The scheduled job/ process gets added to a list provided under the '**Status**' tab.



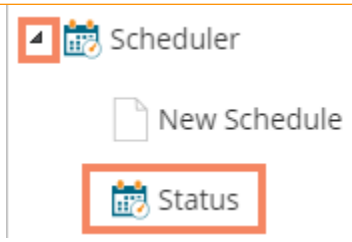
Note:

a. The PDF summary gets sent through email for the scheduled workflows.
b. Multiple email addresses can be entered into a comma-separated value.
c. At present, Spark Workflows are not supported by Scheduler.

10.2. Status

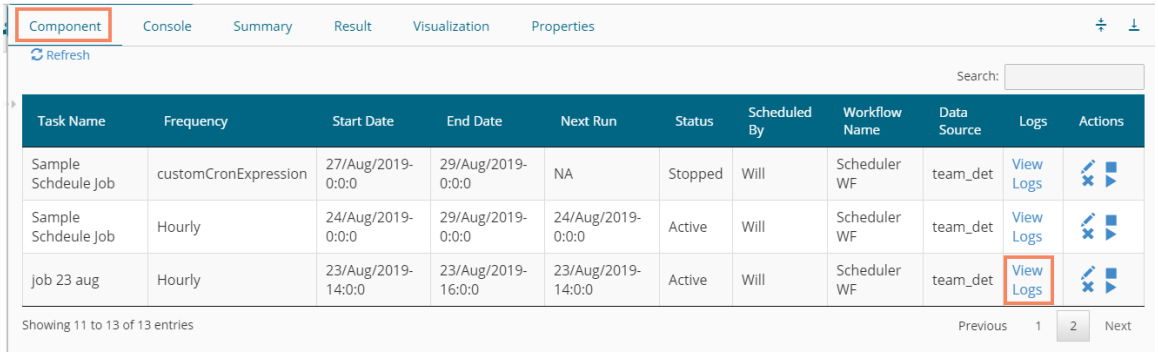
This section displays detailed information for all the scheduled jobs.

i) Click the '**Scheduler**' tree node.
ii) Select the '**Status**' option.



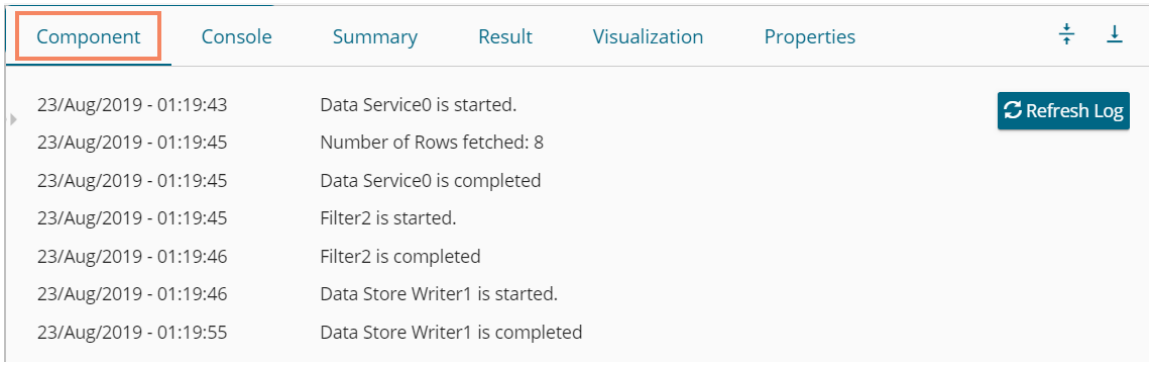
iii) The Component tab opens with a list of all the scheduled jobs.

iv) Click the '**View Logs**' icon.



v) The logs of the selected workflow get displayed under the '**Component**' tab.

vi) Click the '**Refresh Log**' option to refresh the logs.



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. The '**Edit**' option allows the user to update/ edit all the tabs for the selected job.

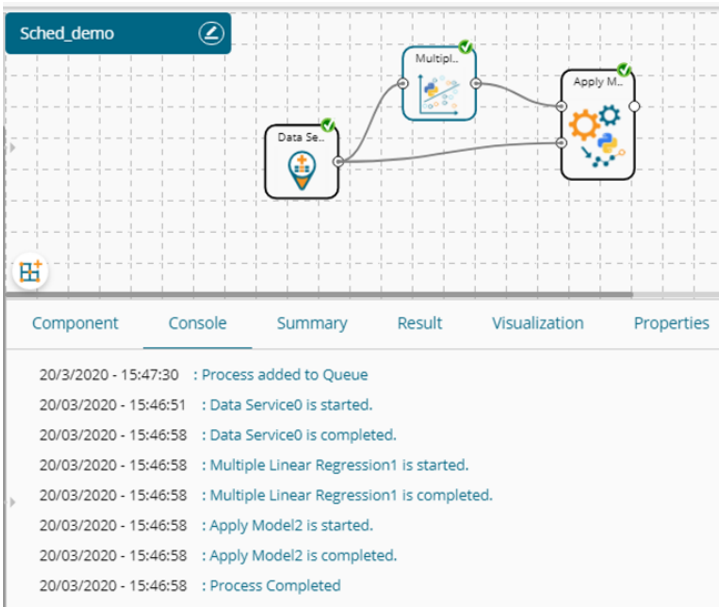
b. The user can click the '**Start**' button to restart the scheduler for a scheduled job until it reaches the end date.

c. The user can enable '**Edit**' and '**Remove**' actions only after stopping the Scheduled job.

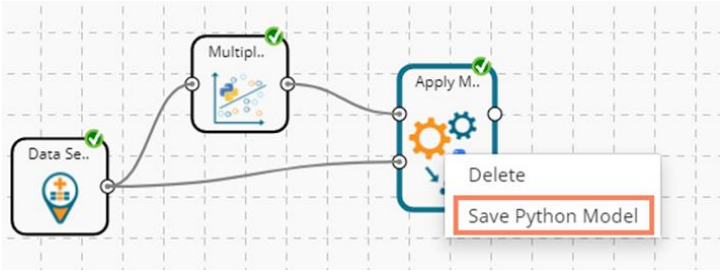
10.2.1. Model Retraining in Scheduler

The users can monitor the model retraining steps through the scheduler.

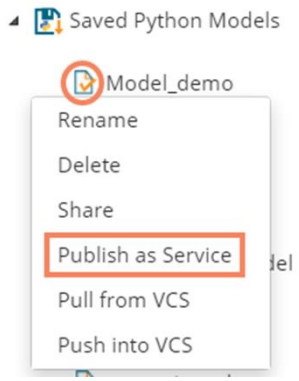
i) Create a Workflow or select a workflow with an Apply Model component.
ii) Run the workflow.



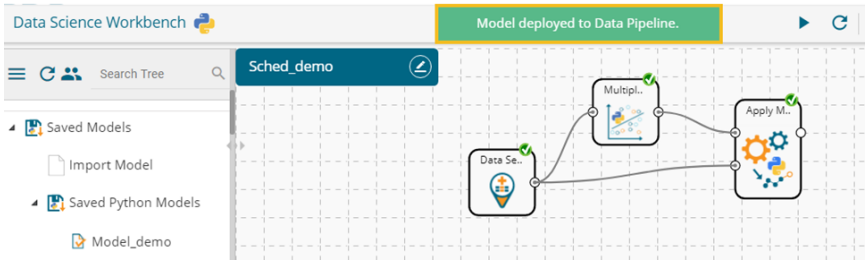
iii) Save the Model.



iv) Navigate to the Saved model.
v) Select the '**Publish as Service**' option to deploy the model to the Data Pipeline.

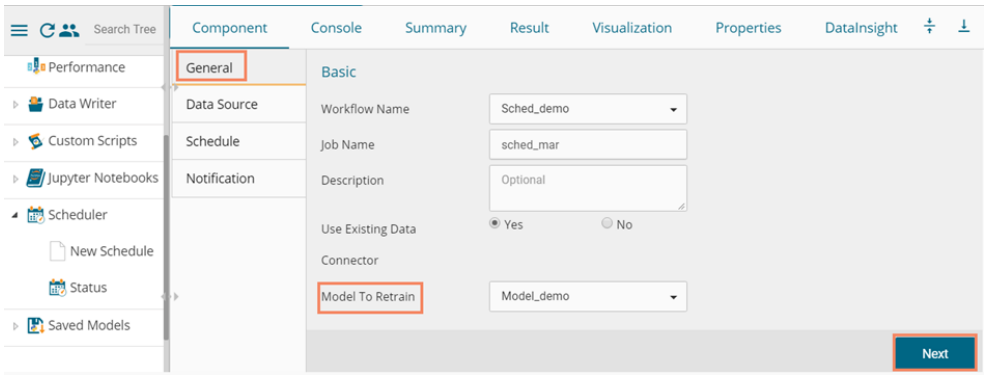


vi) A success message appears to confirm the deployment.

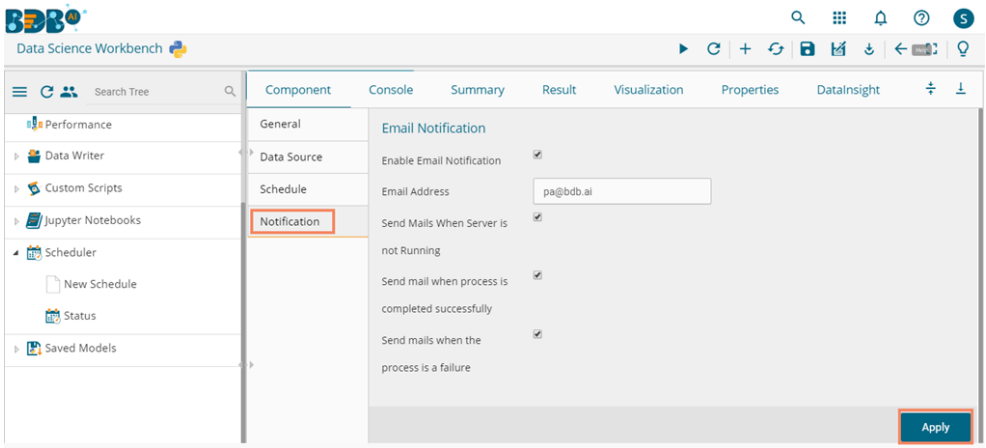


vii) Navigate to the Scheduler and select the same workflow using the '**General**' tab.

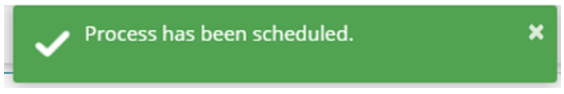
viii) Select the saved model from the '**Model to Retrain**' drop-down menu.



ix) Configure the required steps to schedule the workflow.

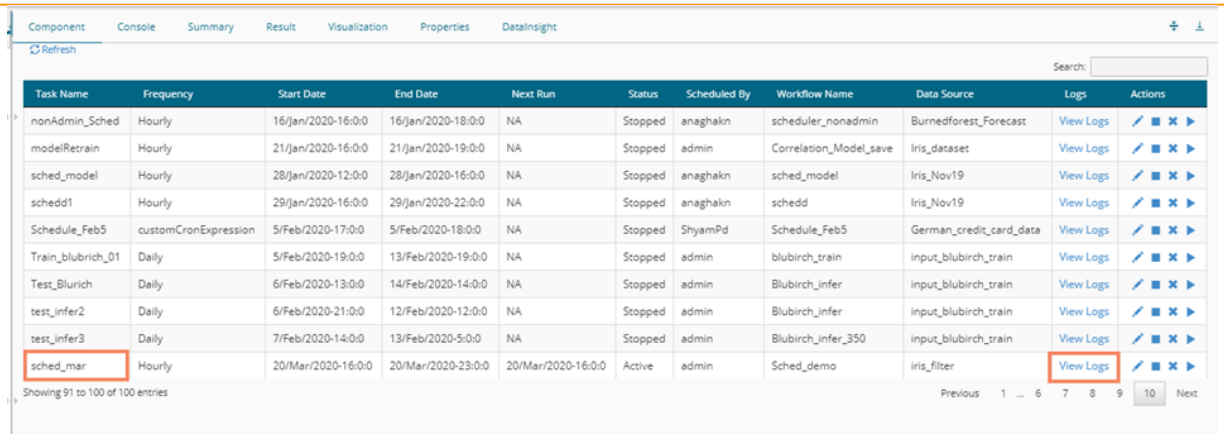


x) A message appears to inform that the process has been scheduled.

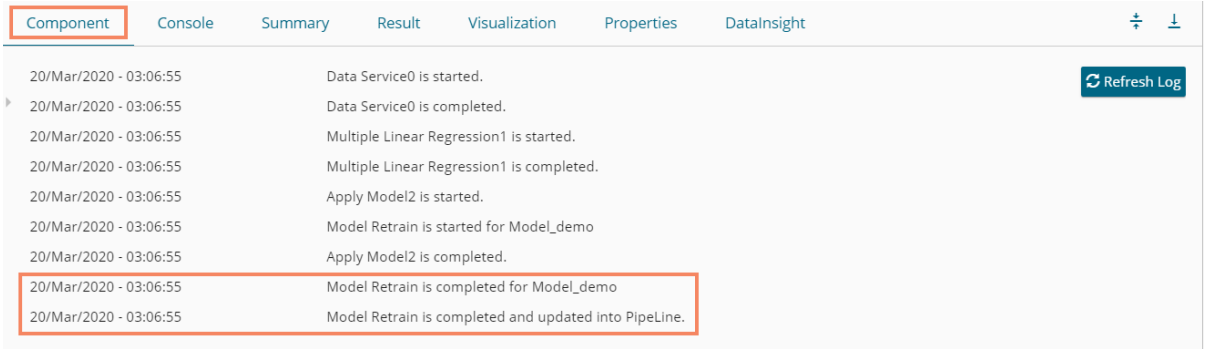


xi) The users get redirected to the Status option displaying all the scheduled processes.

xii) Click the '**View Log**' option.



xiii) The stepwise logs get displayed confirming the model retaining and upload to the Data Pipeline.



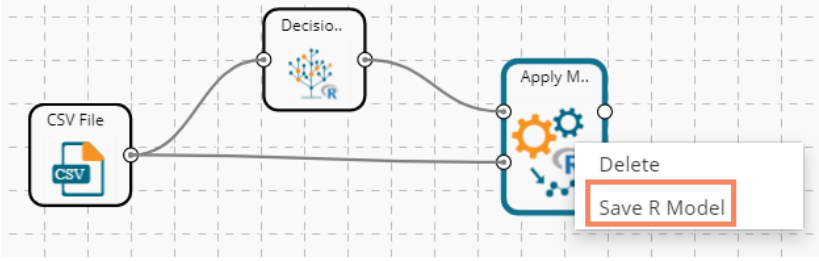
11. Saved Models

The user can save a trained model through the Apply Model component. The user can either split the dataset into training and testing, create a model with training data, and Apply the testing data. Another approach is to save the model and Apply the model over a new test data set.

The user can save a model after successful execution. The saved R models get listed under the '**Saved R Model**' tree node. Users can select a saved R model from the list and use it to create a new workflow.

11.1.1. Saving a Trained Model

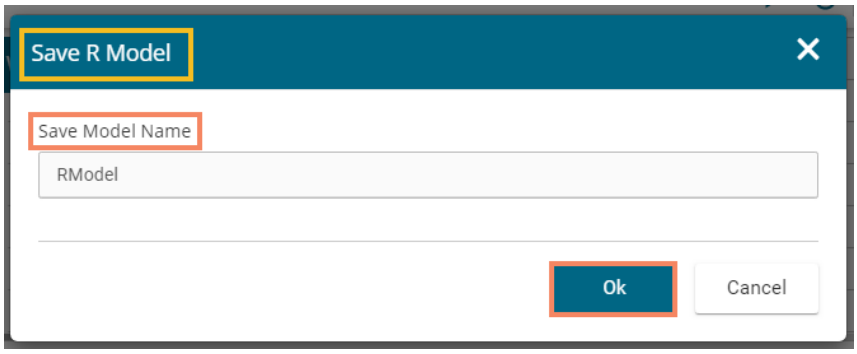
i) Create a Workflow with Apply Model or Open a saved workflow that contains an Apply Model.
ii) Use right-click on the '**Apply Model**' component.
iii) A context menu opens.
iv) Select the '**Save R Model**' option (The 'Save' option for Python and Spark, which gets displayed as '**Save Python Model**' and '**Save Spark Model**' based on the selected workbench).



v) The **Save R Model** window opens (The heading of the Save R Model gets changed as '**Save Python Model**' and '**Save Spark Model**' based on the selected workbench).

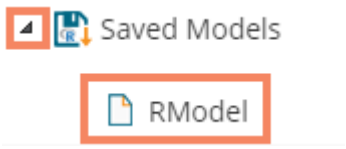
vi) Enter the model name by which you wish to save the model.

vii) Click the '**Ok**' option.

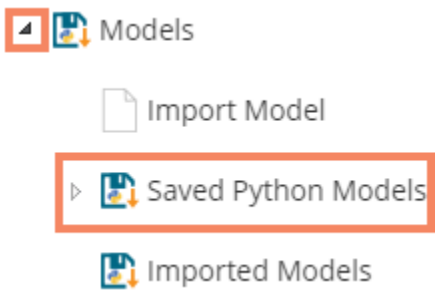


Note: The 'Save Model Name' is a mandatory field. The user cannot give in-between space for two words. The first character of the model name should be an alphabet and must be mentioned in a capital case.

viii) The selected model gets saved in the **'Saved Models'** list.



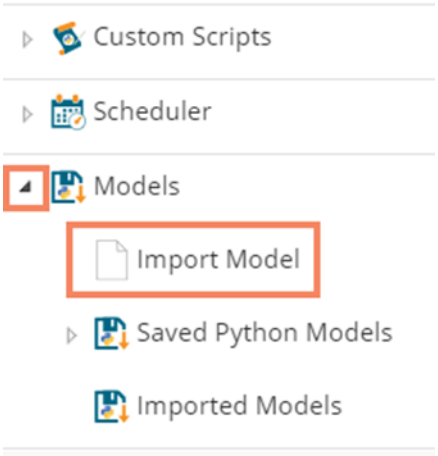
Note: The heading for Python Workbench is '**Models**' as it includes Imported Models together with **Saved Python Models**.



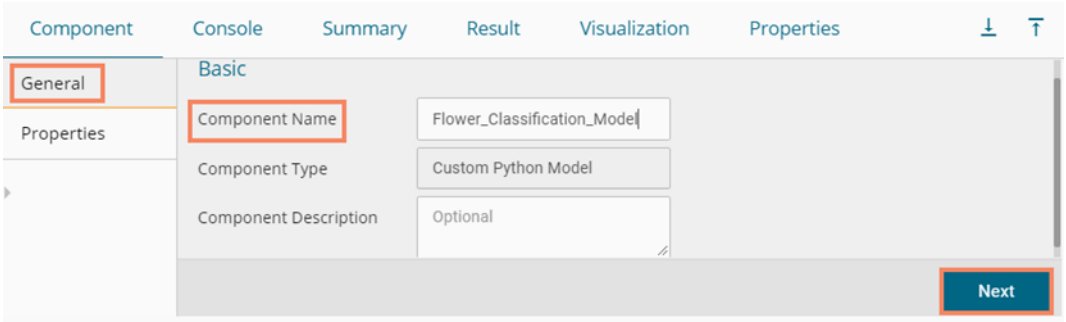
11.1.2. Importing a Model

This component lets a user import any localized model in the Python workbench to use it directly in the BDB platform.

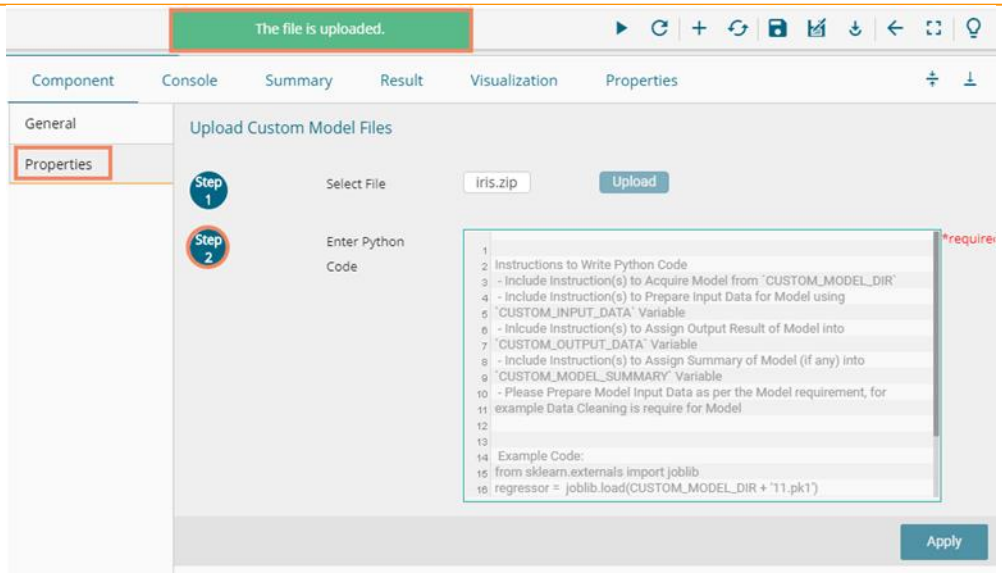
xiv) Navigate to the Models tree-node from the Python Workspace.
xv) Click the '**Import Model'** component tree node.



xvi) The General tab for the Import model opens.
xvii) The user can edit the Component Name.
xviii) Click the '**Next'** option.

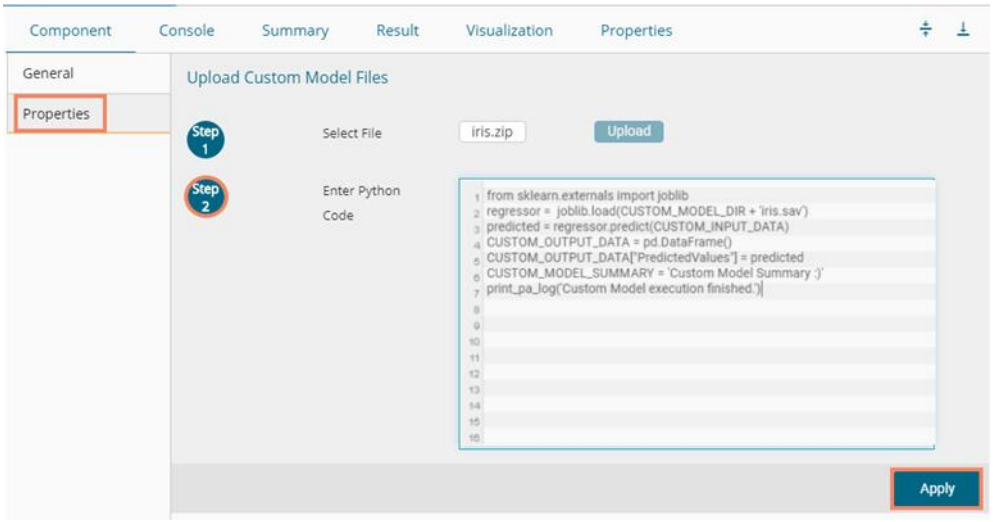


xix) The Properties tab opens.
xx) Upload the model file.
   a. Select a file from the system.
   b. Click the 'Upload' option.
   c. A success message appears to convey that the selected file has been uploaded.

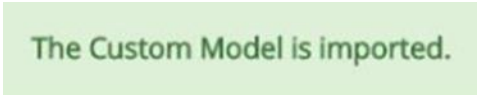


xxi) Enter a Python Code (Script)

xxii) Click the '**Apply**' option.

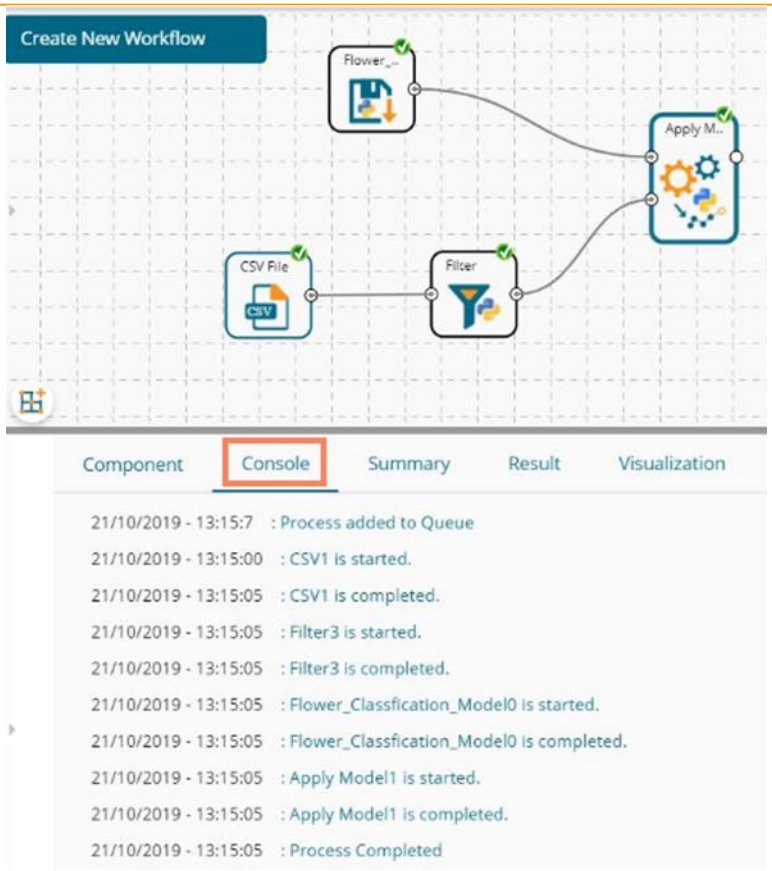


xxiii) A success message appears to convey that the custom model has been imported.

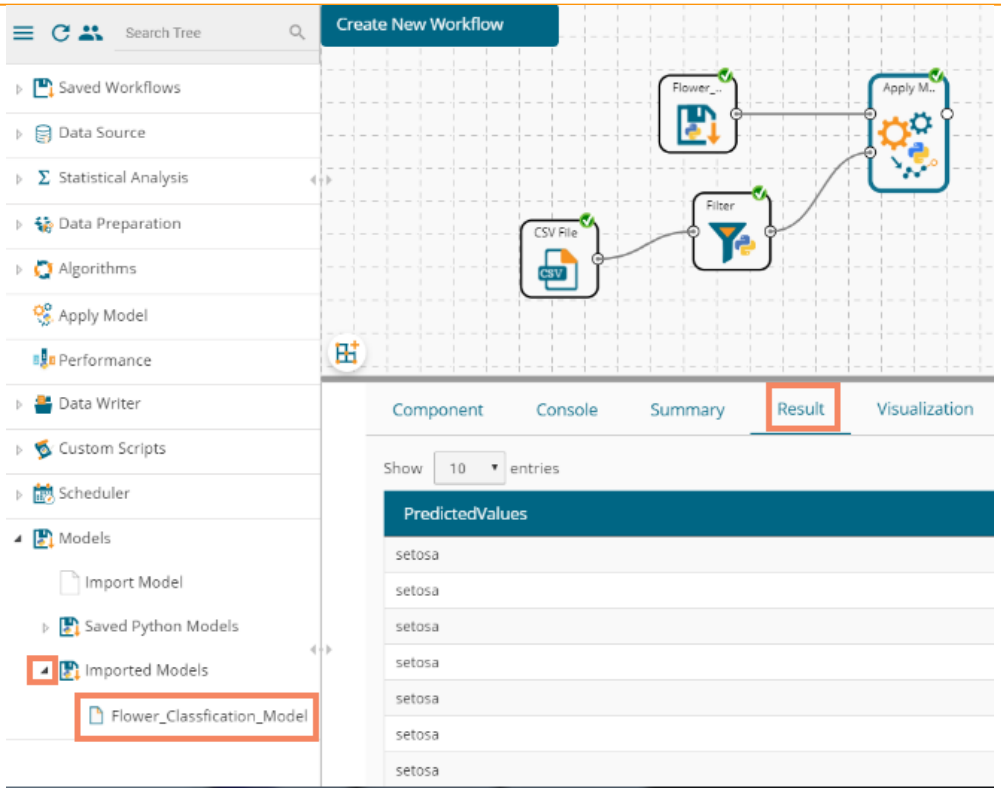


xxiv) Click the Run or Refresh icon to run the model.

xxv) Stepwise completion of the process can be seen under the 'Console' tab. The green checkmarks at the top of the dragged components mark the completion of the console process.



xxvi) Click the '**Result**' tab to get the processed data (To open the Result tab first click on the Apply Model component, then click the 'Result' tab).

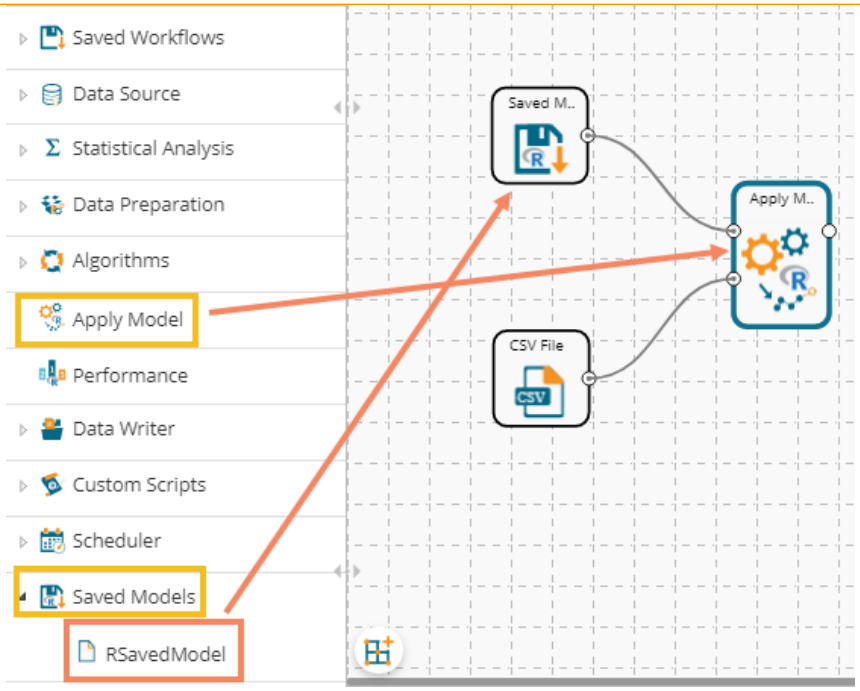


11.1.3. Reading a Saved Model

The user can drag a saved model to the workspace and reuse the model for test data. A saved model can be connected to only Apply Model and new test data source to create a workflow.

i) Select and drag a saved R model component onto the workspace.

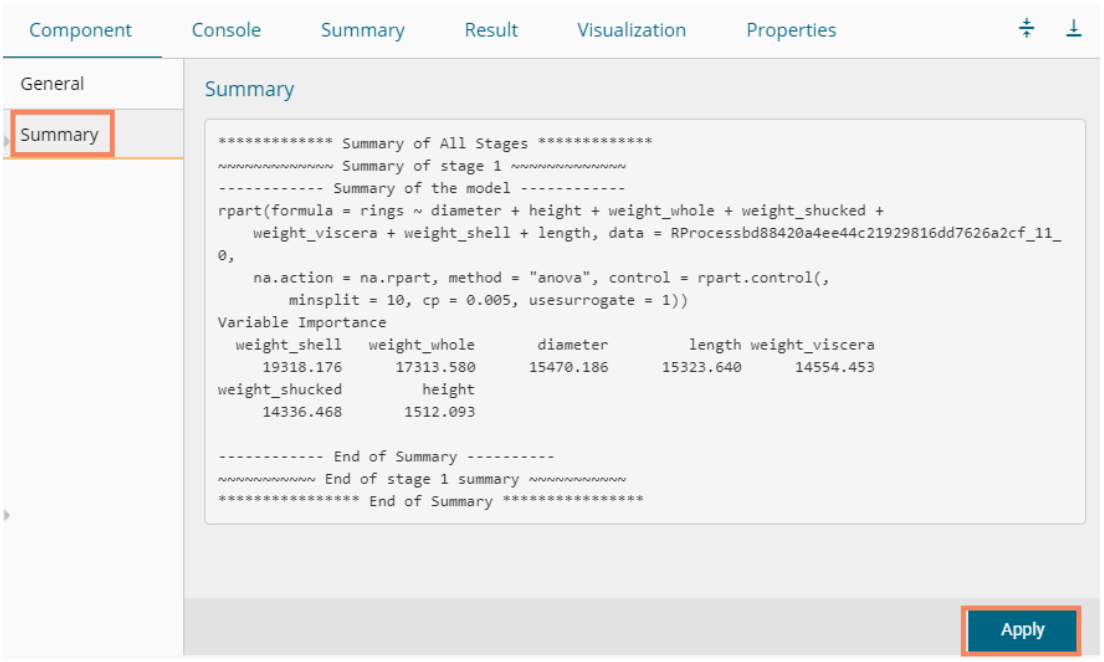
ii) Connect the dragged saved model component and a configured data source to an Apply Model component. Pass the Saved model data in the training node and data source's data in the testing node of the dragged Apply Model component (As shown in the following image).



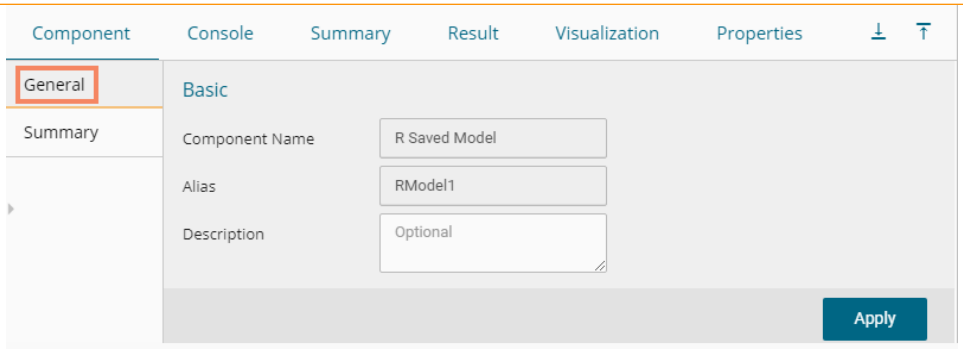
iii) Click on the dragged Saved Model component.

a. The '**Summary**' tab opens by default displaying the model summary.

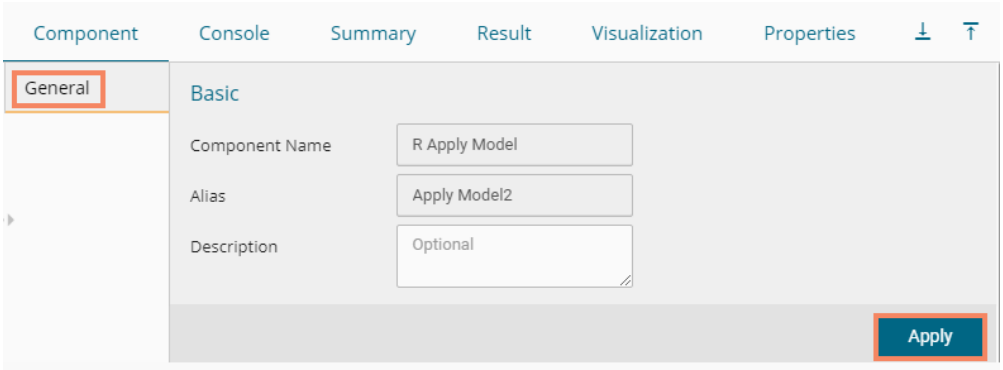
i. Click the '**Apply**' option for the saved model component.



b. Click the '**General**' tab to display the Basic information of the concerned Saved Model component.

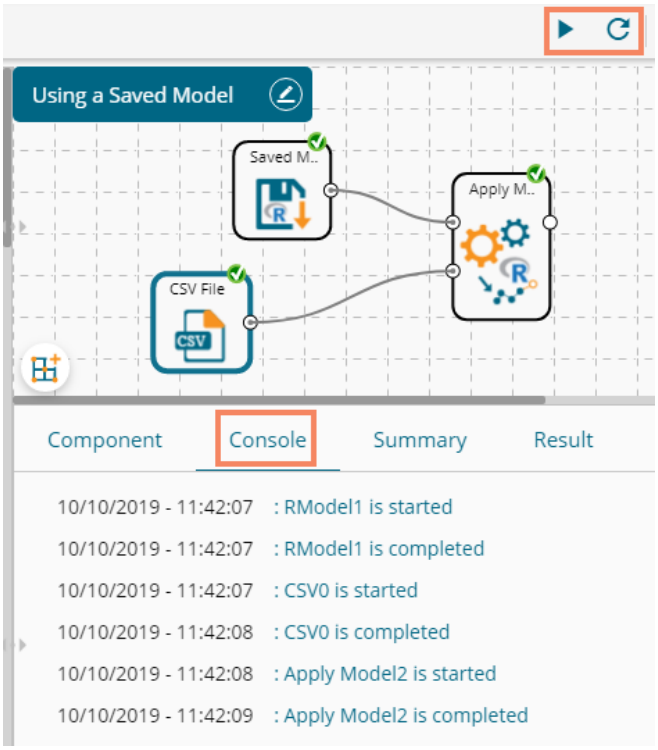


iv) Click the '**Apply**' option provided in the Apply Model component.

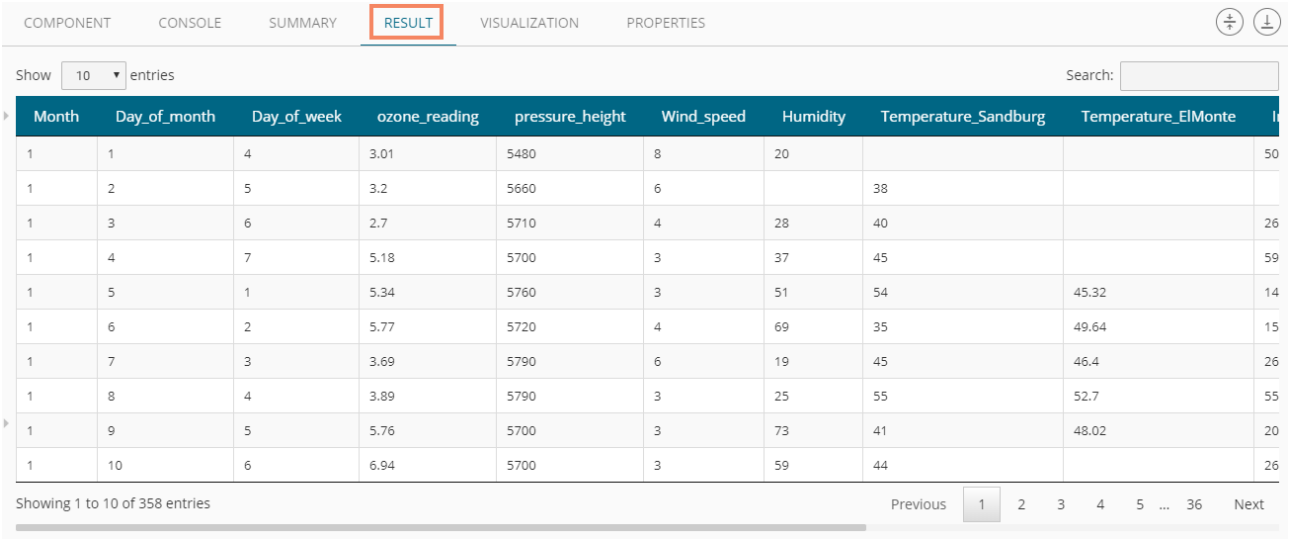


v) Run the workflow after getting the success message.

vi) The '**Console**' tab opens displaying the progress of the process.



vii) After the process gets completed under the Console tab, click the '**Result**' tab to see the processed data.



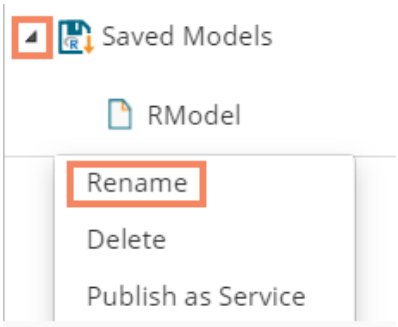
Month	Day_of_month	Day_of_week	ozone_reading	pressure_height	Wind_speed	Humidity	Temperature_Sandburg	Temperature_ElMonte	I
1	1	4	3.01	5480	8	20			50
1	2	5	3.2	5660	6		38		
1	3	6	2.7	5710	4	28	40		26
1	4	7	5.18	5700	3	37	45		59
1	5	1	5.34	5760	3	51	54	45.32	14
1	6	2	5.77	5720	4	69	35	49.64	15
1	7	3	3.69	5790	6	19	45	46.4	26
1	8	4	3.89	5790	3	25	55	52.7	55
1	9	5	5.76	5700	3	73	41	48.02	20
1	10	6	6.94	5700	3	59	44		26

Note:

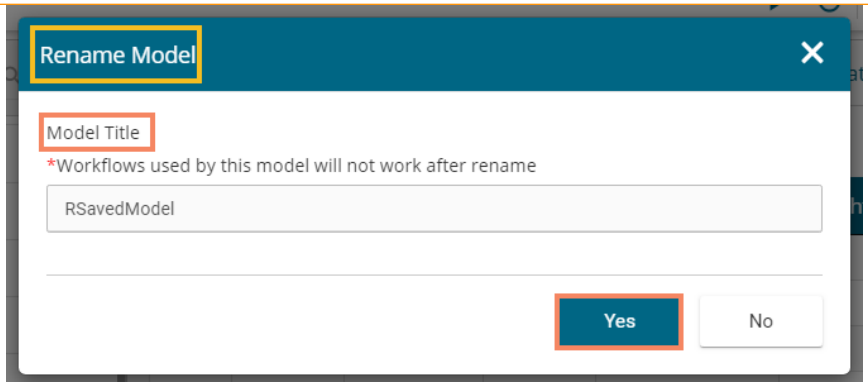
a. A mandatory condition to run the workflow with a '**Saved R Model**' component is that the column headers and data type of the test data source should match with the selected saved model. Otherwise, an error notification of validation failure appears while running the workflow.
b. The user can connect a data writer to the '**Apply Model**' component in a workflow containing a saved model.

11.1.3.1. Renaming a Saved Model

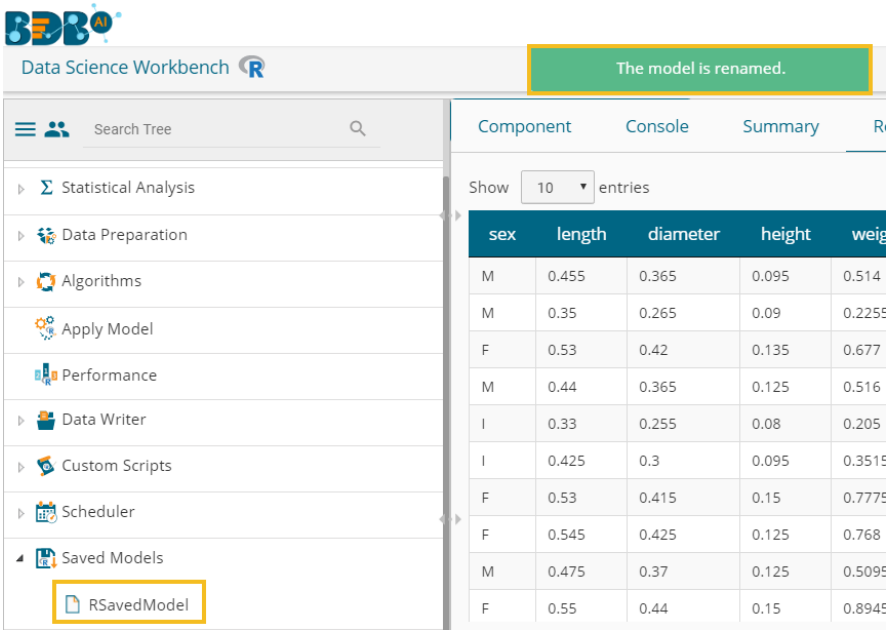
i) Select a model from the '**Saved Models**' list.
ii) Use a right-click on the selected saved model component.
iii) A context menu opens.
iv) Select the '**Rename**' option.



v) A pop-up window appears to rename the model.
vi) Enter a new **'Model Title'** or modify the existing model title in the given field (if desired)
vii) Click the **'Yes'** option**.**

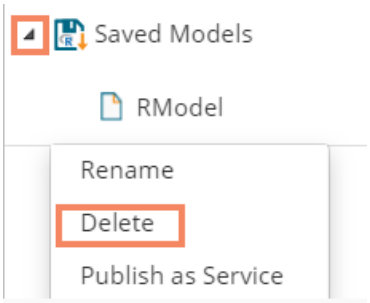


viii) The selected Saved Model gets renamed. A success message appears to notify for the same action.



11.1.3.2. Deleting a Model

i) Select a model from the '**Saved Models**' list.
ii) Right-click on the selected model.
iii) A context menu opens.
iv) Select the **'Delete'** option.



v) A new window opens, asking for confirmation for the deletion.
vi) Click the '**Ok**' option.
vii) The selected saved model gets removed from the **Saved Models** list.

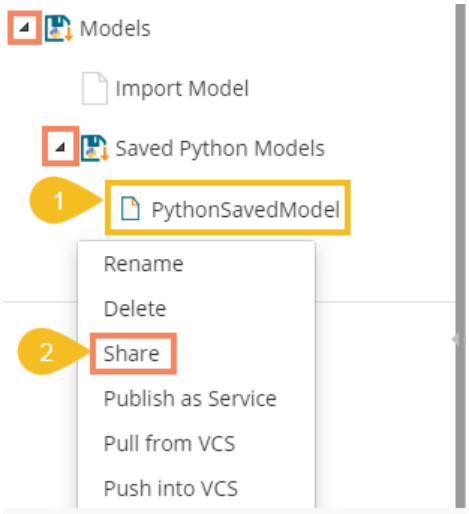
Note: After renaming or deleting a Saved R Model, workflows used by the same model don't work.

11.1.3.3. Sharing a Python Model

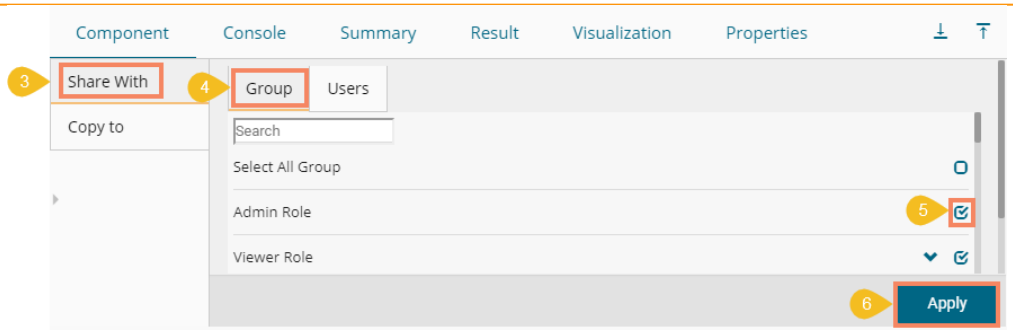
The user can share a saved model with other users or user groups. There are two options to share a selected model:

1. **Share With**: This option allows the user to share a file with the selected users or user groups. Any changes made to the file are 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** (In this case, a Python saved model is selected from the Python Workspace).
   ii) Select the '**Share**' option from the context menu.



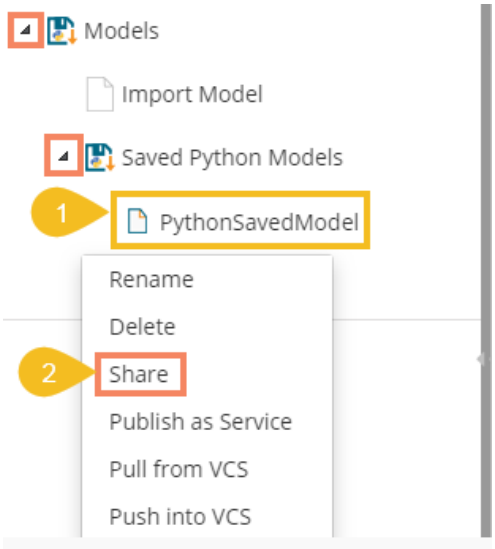
   iii) The '**Share With**' option gets displayed by default.
   iv) Select either the '**Group**' or '**Users**' option.
      a. By selecting a group, all group members inside the group get listed. Users can be excluded by not selecting them from the group.
      b. Users can be excluded by not selecting a username from the list when the '**User**' option has been selected.
   v) Select a specific group or user from the list by using checkmarks in the box.
   vi) Click the '**Apply**' option.



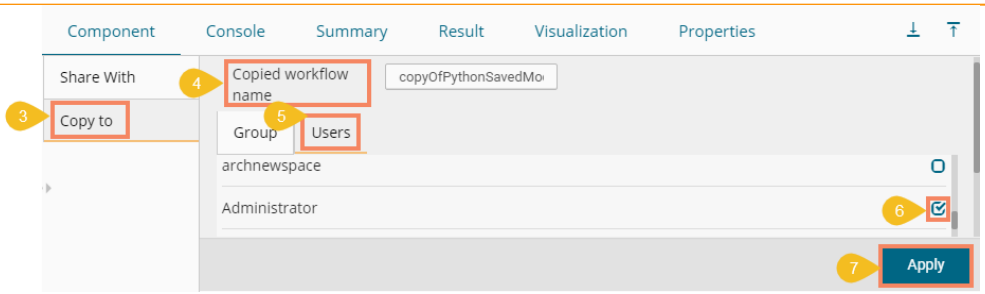
vii) The saved model gets shared with the selected group of users.

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) Use right-click on a model from the list of the **Saved Models** (In this case, a Python saved model is selected from the Python Workspace).
   ii) Select the '**Share**' option from the context menu.



   iii) Select the '**Copy To**' option.
   iv) The copied model name gets displayed.
   v) Select either the '**Group**' or '**Users**' option with a click.
      a. By selecting a group, all group members inside the group get listed. Users can be excluded by not selecting them from the group.
      b. Users can be excluded by not selecting a username from the list when the '**Users**' option has been selected.
   vi) Select a specific group or user from the list by using checkmarks in the box.
   vii) Click the '**Apply**' option.

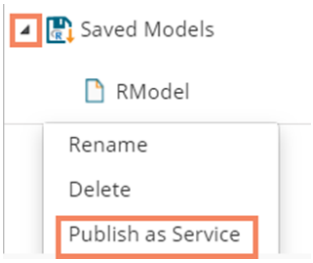


viii) A copy of the model gets shared with the selected user or group.

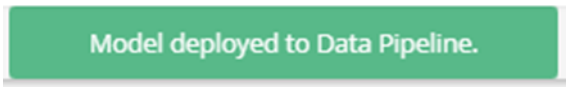
11.1.3.4. Publishing a Saved Model as Service

The user can publish the saved Data Science models to the Data Pipeline module using this option. The user can access the published Data Science model using the ML model runner component to use them in a pipeline workflow.

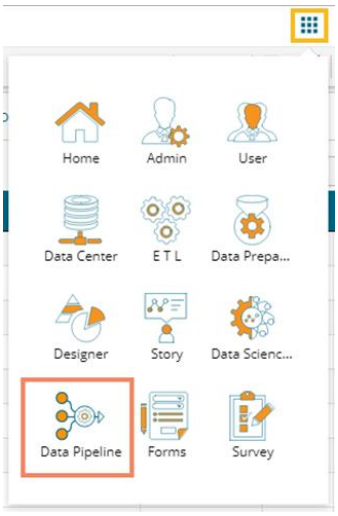
i) Select a model from the Saved Models list.
ii) Open the context menu provided for the selected saved model.
iii) Select the '**Publish as Service**' option for the selected model.



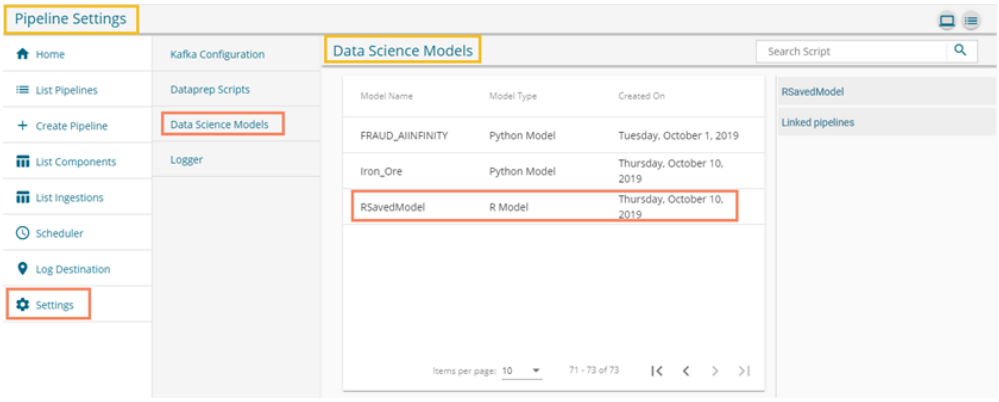
iv) A success message appears to notify the user that the selected saved model is deployed to the Data Pipeline plugin.



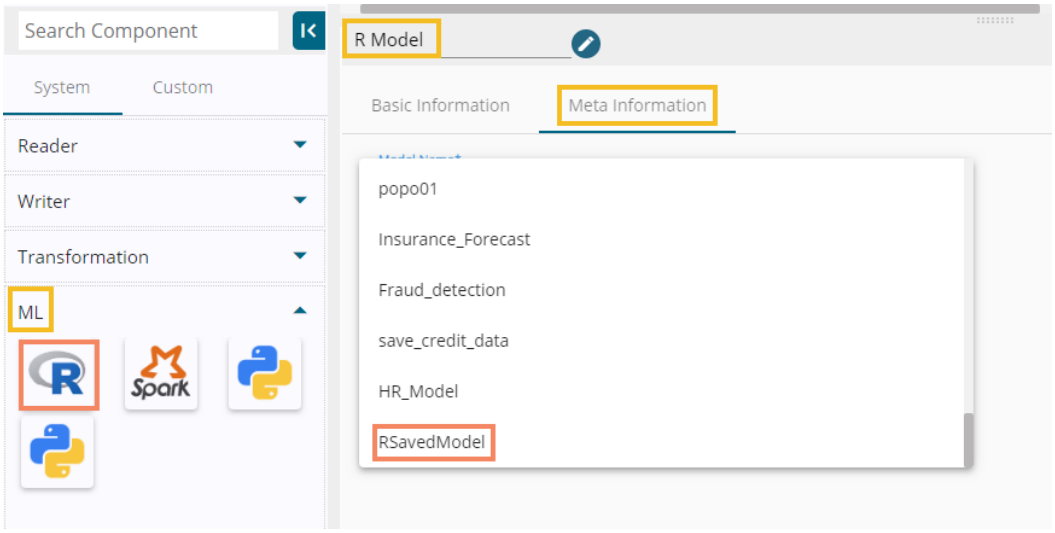
v) Navigate to the Data Pipeline plugin using the '**Apps**' menu.



vi) Open the '**Settings**' page.
vii) The published saved model gets added to the Data Science Models list.



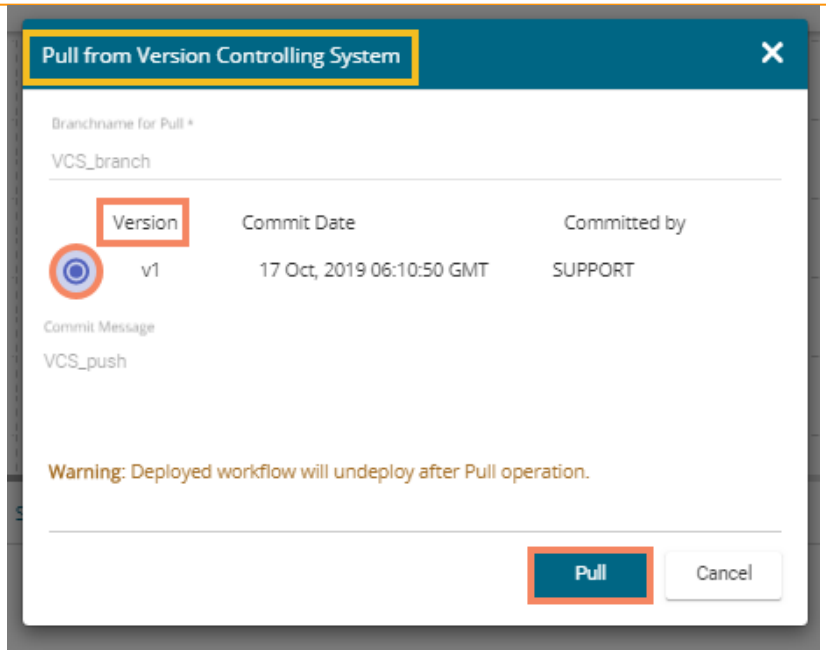
viii) Access the R Model runner component from the Component Pallet.
ix) Open the Meta Information tab and scroll down the provided drop-down list.
x) The published saved model from the Data science Workbench appears in the drop-down list.



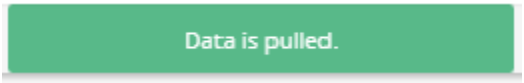
11.1.3.5. Pull from VCS

The option helps to pull models from the Version Controlling Service.

i) Select a model from the Saved Workflow list.
ii) Click the '**Pull from VCS**' option.
iii) A window opens like below:
   a) The branch name for pull comes pre-written.
   b) The details of the existing version get displayed from where the user can select the desired version using the radio button.
   c) Click the '**Pull**' option.



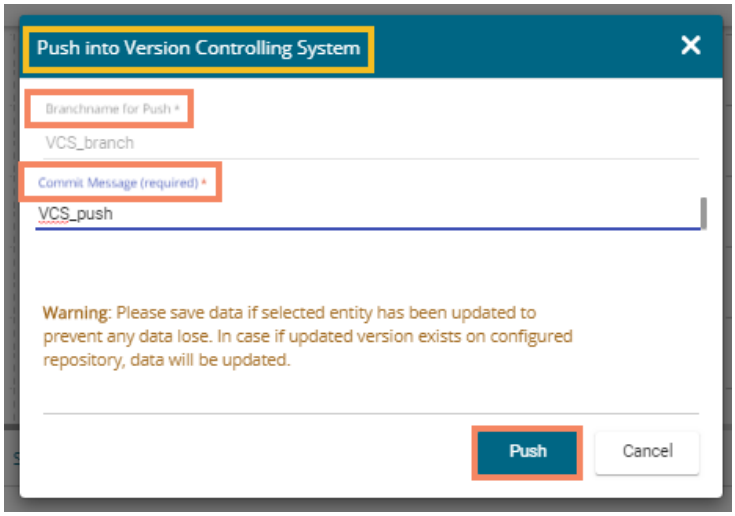
d) A success message appears to indicate that the selected entity has been pulled from the VCS.



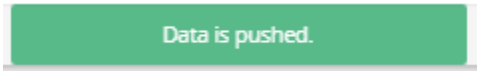
11.1.3.6. Push into VCS

The option helps to push the workflow into the Version Controlling Service.

i) Select a workflow from the Saved Workflow list.
ii) Click the '**Pull from VCS**' option.
iii) A window opens like below:
   a) The branch name for push comes pre-written.
   b) Provide Commit message (it is mandatory)
   c) Click the '**Push**' option.



d) A success message appears to indicate that the selected entity has been pushed into the VCS.

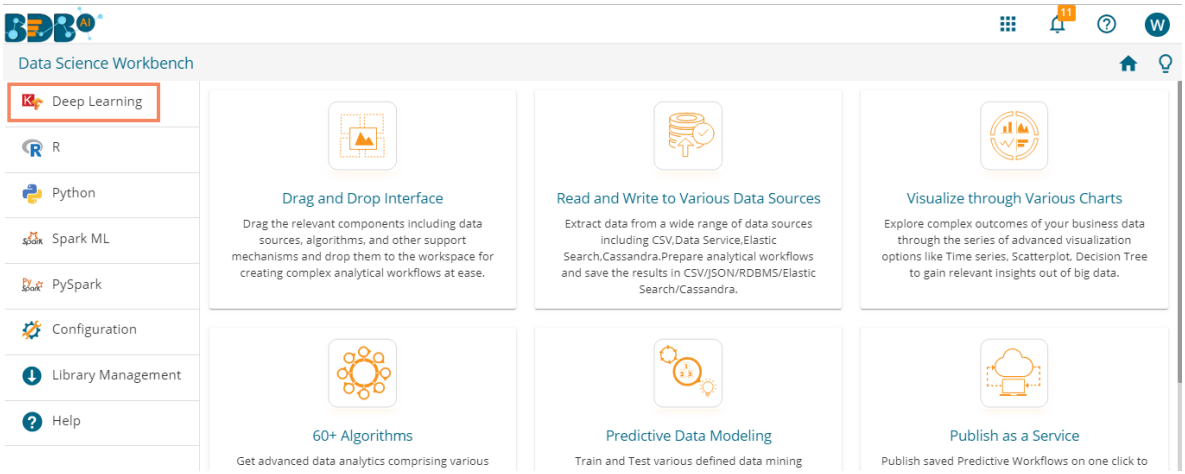


Note:

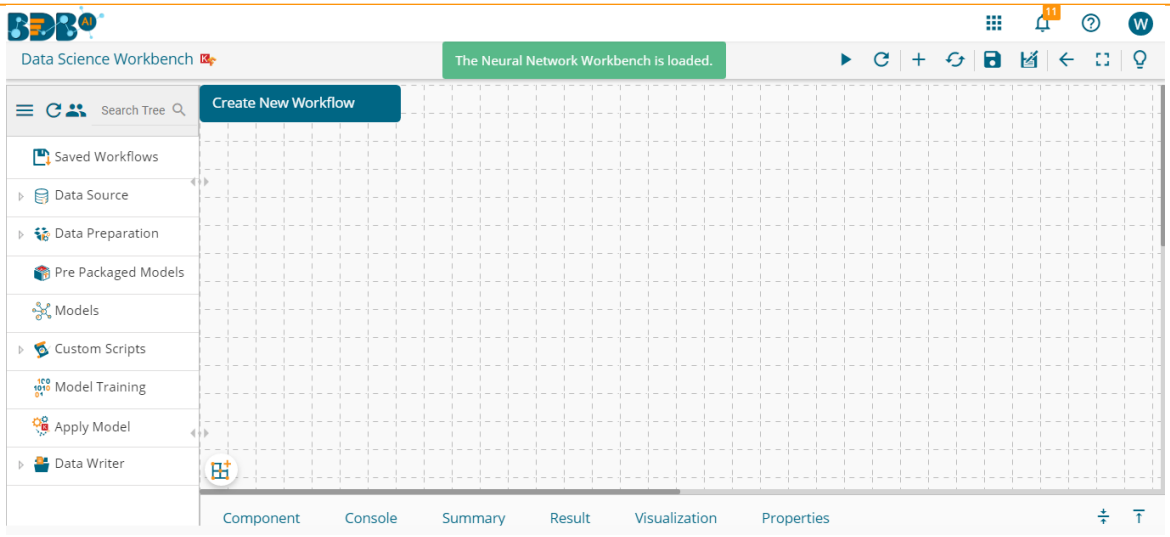
a. At present, the **Pull from VCS** and **Push into VCS** options are available only for the Python Workspace.
b. Data Science Models can get deployed multiple times to the Data Pipeline module and get marked to identify the deployed models.

12. Deep Learning Workspace

The user can select the Deep Learning Workspace from the Data Science landing page to access the Neural Network Environment under the Data Science Workbench.



The user gets redirected to the following screen by selecting the NN Workspace:



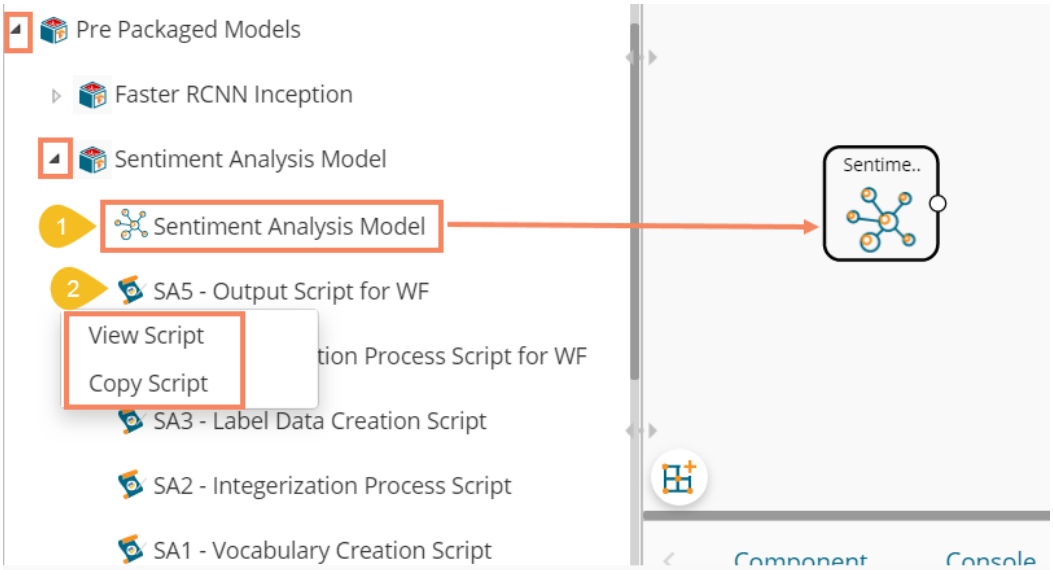
Note:

a. Neural Network Space is applicable only for Python Environment.
b. Keras (as a High-level API) is supported by the Tensorflow Backend.
c. Tensorboard is attached for the Live Visual Tracking of Model during Training.
d. Model Creation using Python Script is supported.
e. A pre-trained Model of Sentiment Analysis is Provided along with its feature scripts.

The Component Tree-node menu displays various components with their sub-components to be used in the NN workspace as per requirement.

12.1. Pre-Packaged Models

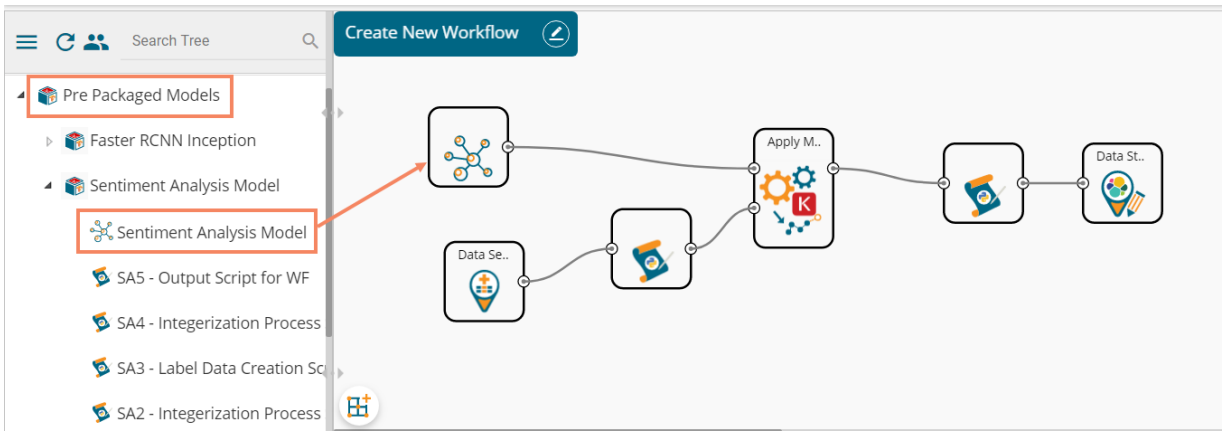
The component tree-node provided on the NN Workspace contains one node as Pre-Packaged Models which contains the Pre-trained Sentiment Analysis Model and its feature scripts.



- The user can use the Pre-trained Model in a Workflow.

- These Scripts can be used directly in Workbench Area using drag-n-drop Functionality.
- The user can Copy the Script, Modify the Code, and then use them as per their need.
- The user must use the 'NN Apply Model' that applies the selected NN-Model over input data to get predicted Results.
- Along with these Pre-trained Models and Scripts, you get support files for training this model (these can be viewed in the 'Supporting Files' tabs of View Model). These supporting files users can access using the **SHARED_PATH** variable in the scripts.

Note: The featured scripts are provided with a Pre-trained Sentiment Analysis Model. If the users wish to modify the scripts OR refer to these scripts for other user-defined models, then they must be modified as per their requirements and need to avoid error(s) & incorrect calculation.
The following image displays a workflow created by using a pre-trained model:



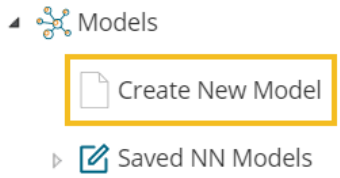
12.2. Working with Deep Learning Workspace

This section explains the general steps for Training a Neural Network Model. The entire process can be described in the below-mentioned parts:

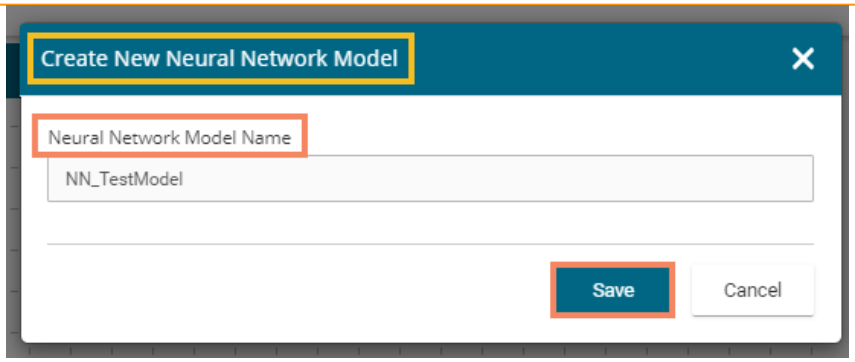
12.2.1. Creating a New Model

The user needs to start the process from the creation of a new model.

i) Click on the '**Create New Model**' option from the Models tree node.



ii) A Dialog Box opens.
iii) Provide a name for the Deep Learning model.
iv) Click the '**Save**' option.

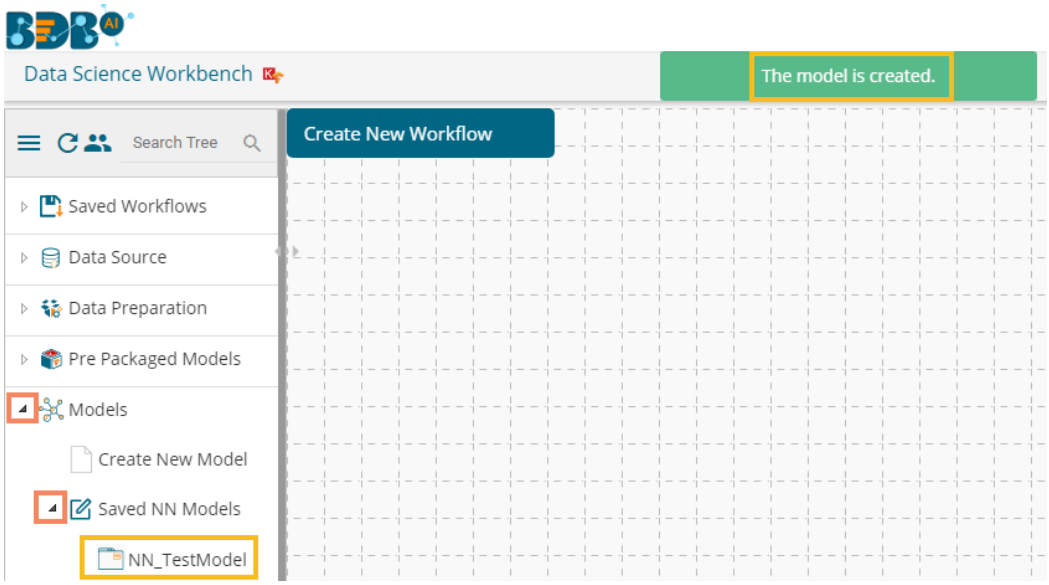


Note:

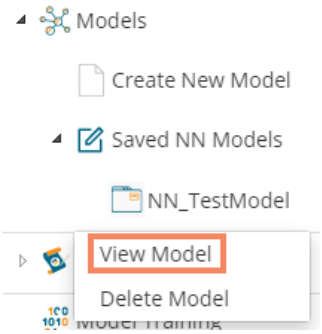
a. The user can use a maximum of 20 characters to provide a name for the newly created Model
b. No other Special Character(s) except Underscore (_) is allowed
c. Model Name cannot begin with Space/Numeric Digit or Underscore
d. Model Name should be unique

v) A success message appears to assure that the new model has been created.

vi) The new model gets listed under the '**Saved NN Models**' tree node.

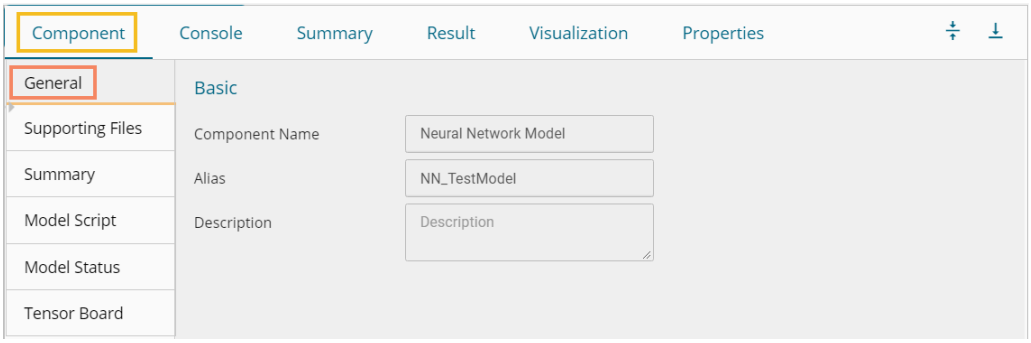


vii) Use a right-click on the model and select the '**View Model**' option.



viii) The component details open for the selected model, as shown in the following image:
The user can view only the General tab displaying the Basic information about the newly created NN Model.

a. General: The Basic Details regarding the NN model are displayed in this tab.



Note: The remaining tabs do not display any information until the model gets trained.

12.2.2. Data Preprocessing

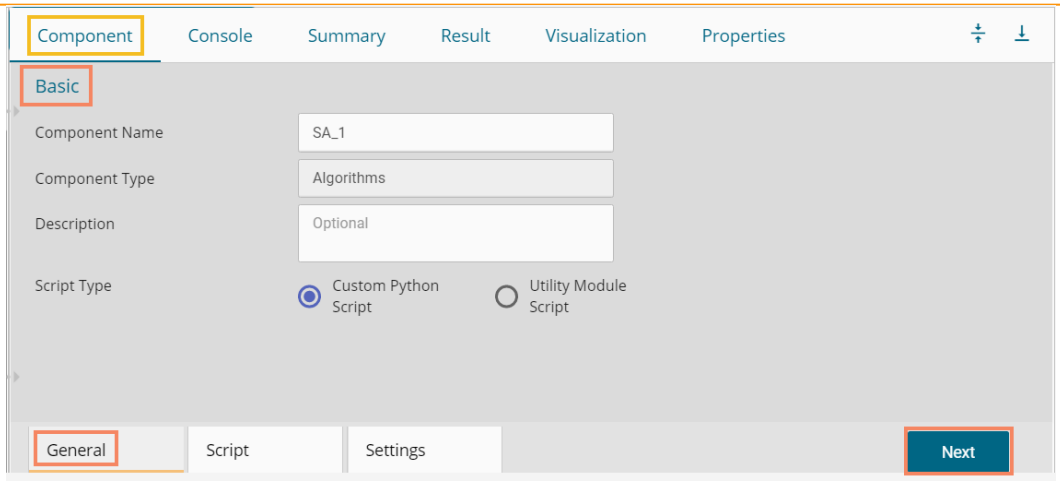
12.2.2.1. Creating a NumPy Script

This section describes data preprocessing from creating NumPy files to have the required data in a binary format that a Model Script can use for training or prediction purposes.
In this section, the user must pre-process the data that is required for a model to get trained; we call this process '**Data Preprocessing**' or NumPy-fication.

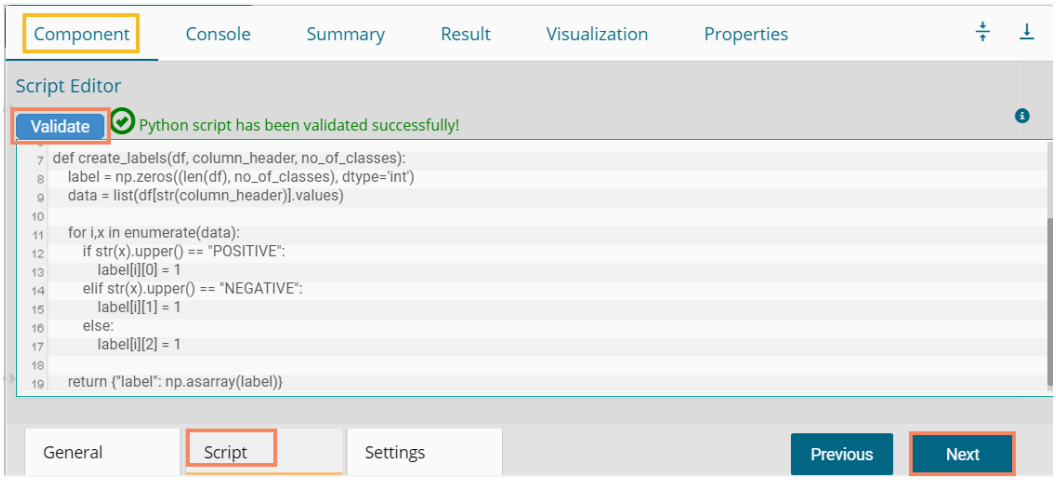
Here, the user creates NumPy files; these files have the information of data in a binary format that can be fed into the model during/after training.

Use the '**Custom Script**' tree-node to create a new script inside the NN Workspace. The workflow for creating a new script is like the Python Workspace. The user can also choose an option to create a Utility Module Script.

i) Select the '**Create New Script**' option using the '**Custom Python Script**' tree node.
ii) The Component tab displaying the General tab opens.
iii) Provide the Basic component information:
   1. Provide a Component Name.
   2. The Component Type comes pre-filled.
   3. Provide relevant Descriptions about the component.
   4. Select a script type by using the radio button.
   5. Click the '**Next**' option.



iv) The '**Script**' tab opens.
v) Insert script syntax in the Script Editor space.
vi) Click the '**Validate**' option. It should get the success message to move ahead.
vii) Click the '**Next**' option.

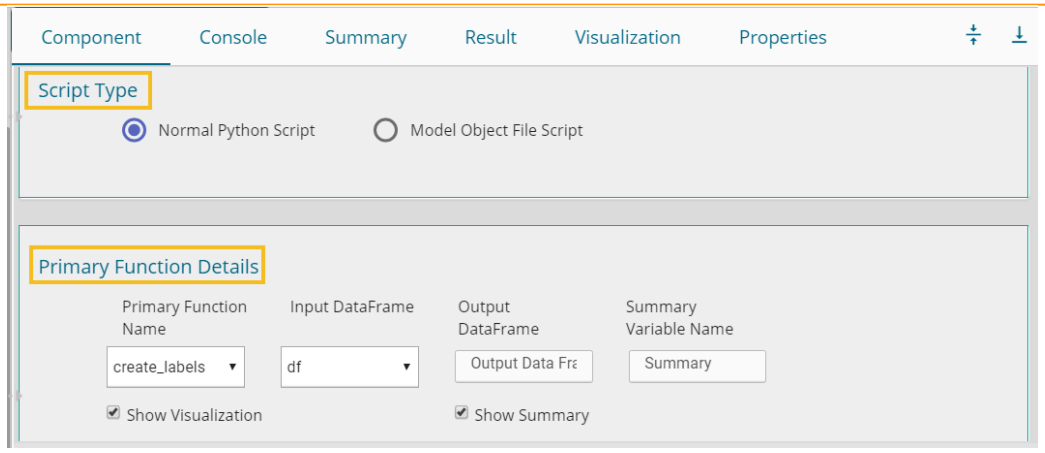


viii) The '**Settings**' tab opens.
ix) Select a Script Type using the checkbox.

1. **Normal Python Script**

   If the selected script type is **Normal Python Script**, then the Primary Function Details gets displayed immediately after the Script Type to be configured:

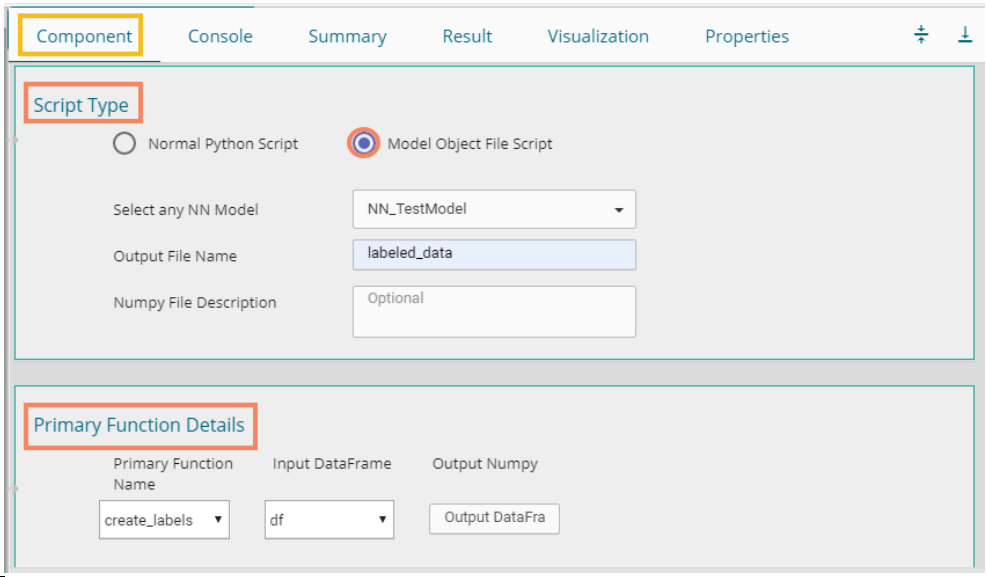
   ii. Select a Primary Function Name from the drop-down list
   iii. Select an Input Data Frame option from the drop-down list
   iv. Provide a name for the Output Data Frame
   v. Provide the Summary Variable Name (if the View Summary option is enabled)
   vi. Enable '**Show Visualization**' and '**Show Summary**' options by enabling them in the boxes.



2. **Model Object File Script**

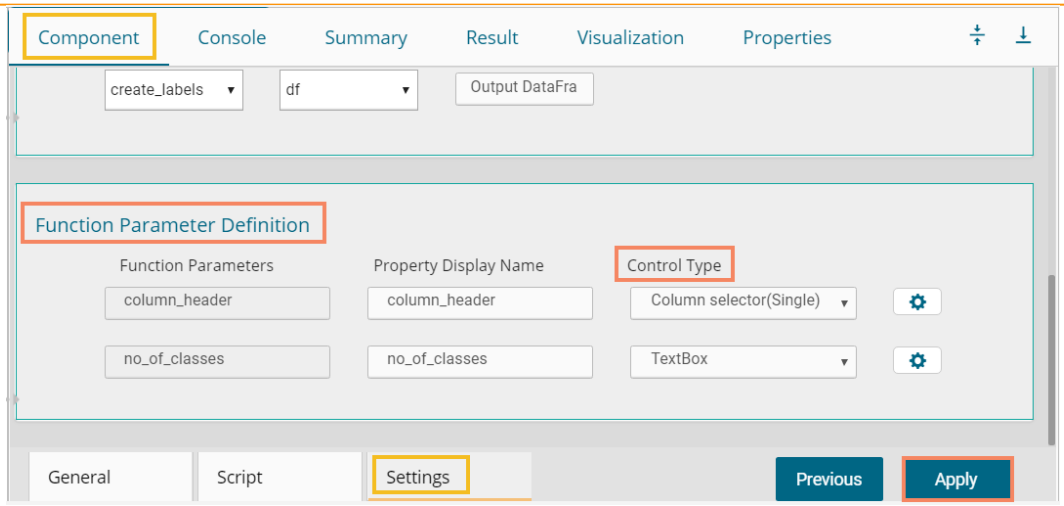
   If it is a Model Object File Script (i.e., NumPy File Creation), then the user needs to provide the following details to configure the Primary Function details:

   i. Select any one NN Model using the drop-down list, which can be associated with an Output NumPy Filename.
   ii. The Output File Name appears in the given box.
   iii. Describe the NumPy File.
   iv. Configure the Primary function details
      a) Select a Primary Function Name using the drop-down list
      b) Select an Input Data Frame using the drop-down list
      c) Provide an Output name for the NumPy

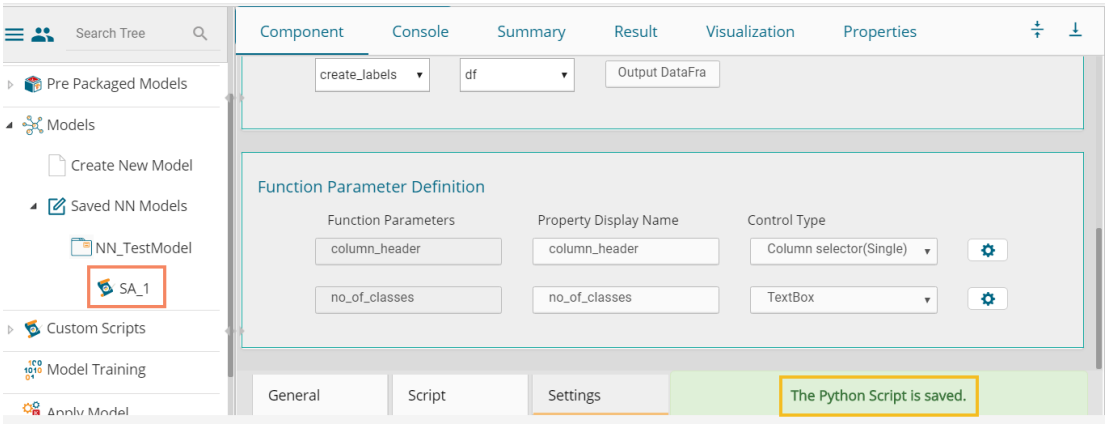


Note: The user needs to create a model object File Script to get it listed along with the model.

x) Configure the Function Parameters by providing relevant Property Display Name and defining the Control Type:
xi) Click the 'Apply' option.



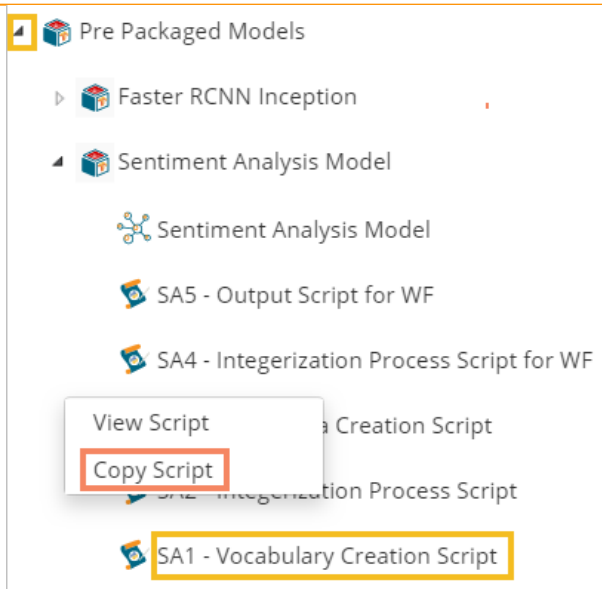
xii) A Success message appears to confirm the creation of a Python script.
xiii) The newly created NumPy script gets added to the model folder.



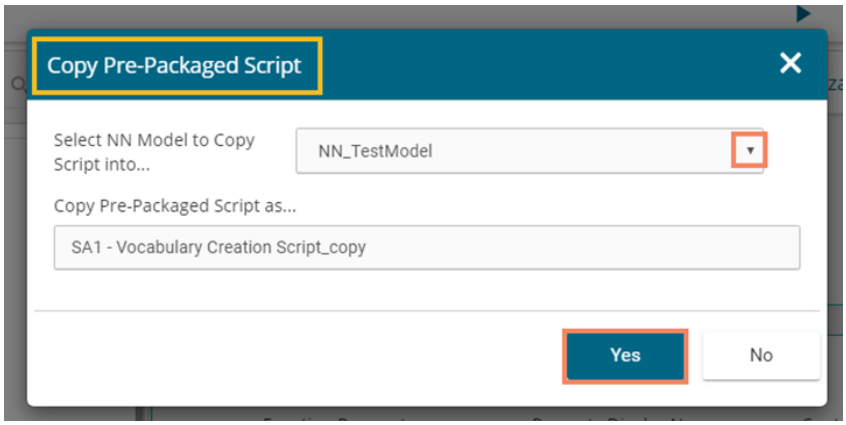
12.2.2.2. Copying a Pre-Packaged Script

The user can copy the existing scripts and use them if he wants to use the pre-packaged script instead of creating a new NumPy Script.

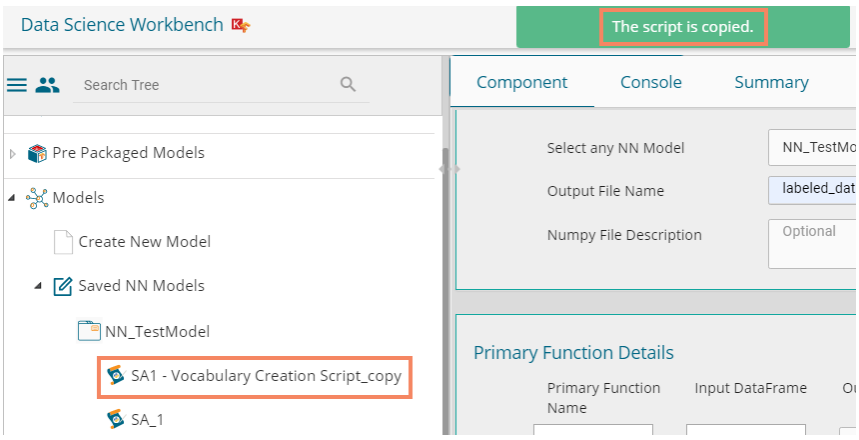
i) Navigate to the pre-packaged Scripts option.
ii) Select a pre-packaged script and click on it avail of the options.
iii) Click the '**Copy Script**' option.



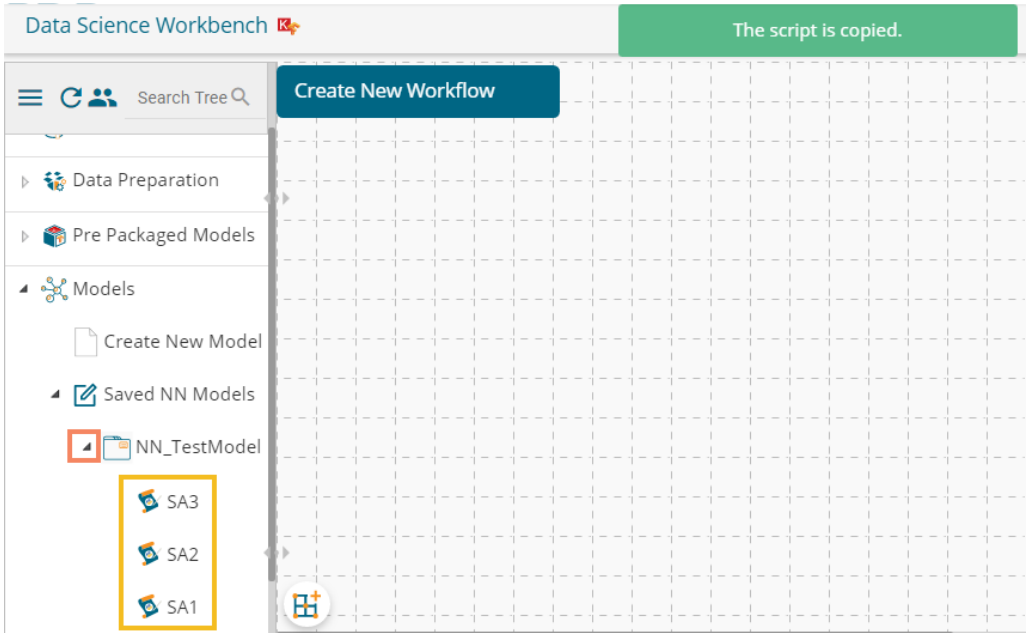
iv) The Copy Pre-Packaged Script dialog box opens.
v) Select the NN Model to Copy the script using the drop-down option.
vi) Provide a name that you wish to display for the copied pre-package script.
vii) Click the '**Yes**' option.



viii) A success message appears.
ix) The copied script gets listed below the model.



x) The copied script for the 'NN_TestModel' is as displayed below:



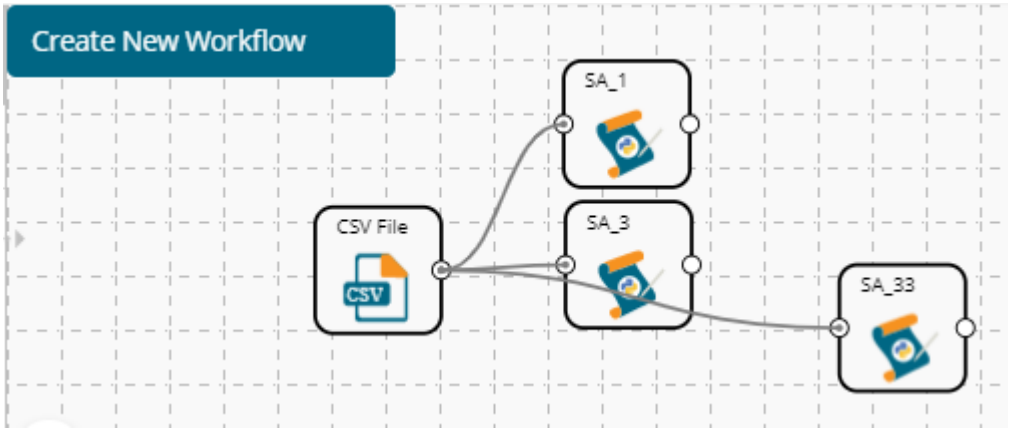
Note:

a. Output for NumPy Script must be a NumPy array. The created NumPy script can be used with any Data-Source, and as the workflow gets completed, the NumPy file gets created and stored for future use with the selected NN Model.
b. To access a NumPy file from the selected model use, FAKE_PATH+ '/<filename>.'
c. To access the shared NumPy file from the Pre-packaged models provided use, SHARED_PATH+ '/<filename>.'
d. The user can also add multiple files/scripts and click the 'Apply' option to enable them for the saved model.

12.2.3. Running the NumPy Script(s)

The user needs to run the script(s) created or copied to the selected model.

i) Connect the script component(s) to a data source.

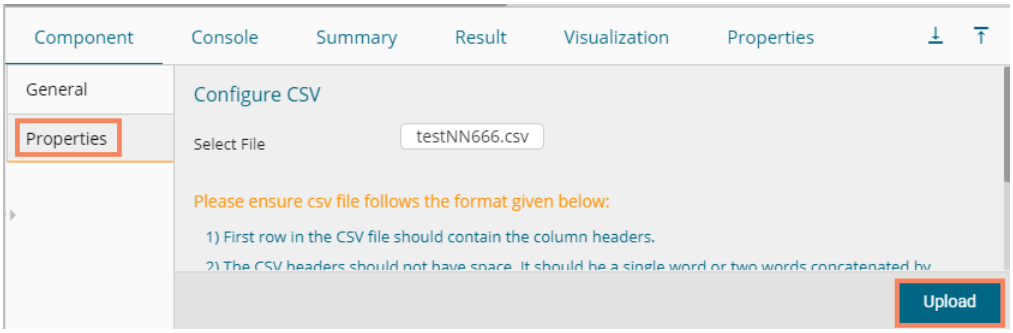


ii) Configure the required fields (The fields for all the script components and data source should be configured)

a) Data Source

i. Browse a file

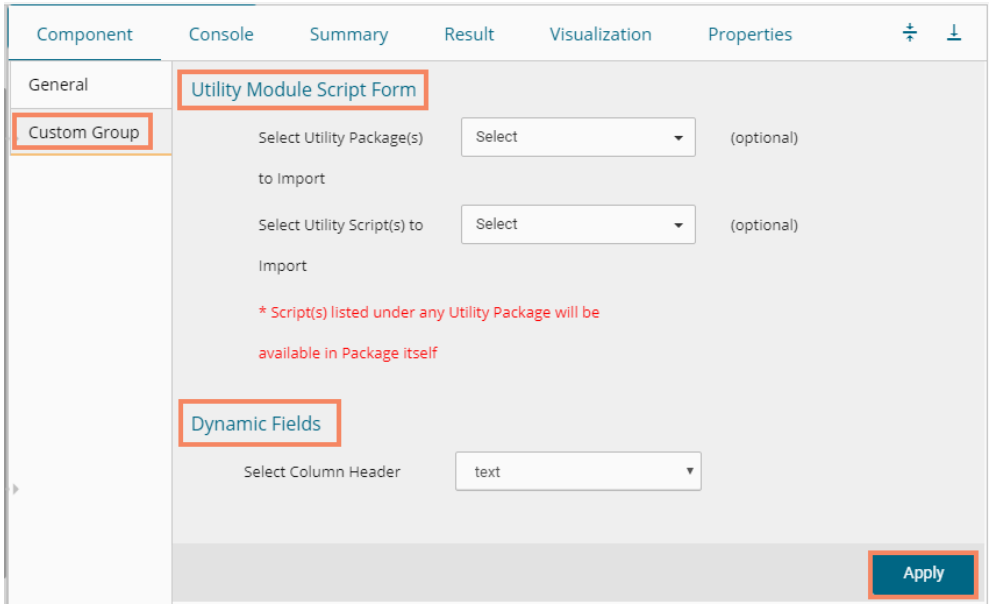
ii. Click the '**Upload**' option.



b) Script 1 (SA_1)

i. Configure the Custom Group options

1. Utility Module Script Form
   a. Select Utility (Package(s) to import (optional)
   b. Select Utility Script(s) to import (optional)
2. Dynamic Fields
   a. Select Column Header from the drop-down.
3. Click the 'Apply' option.

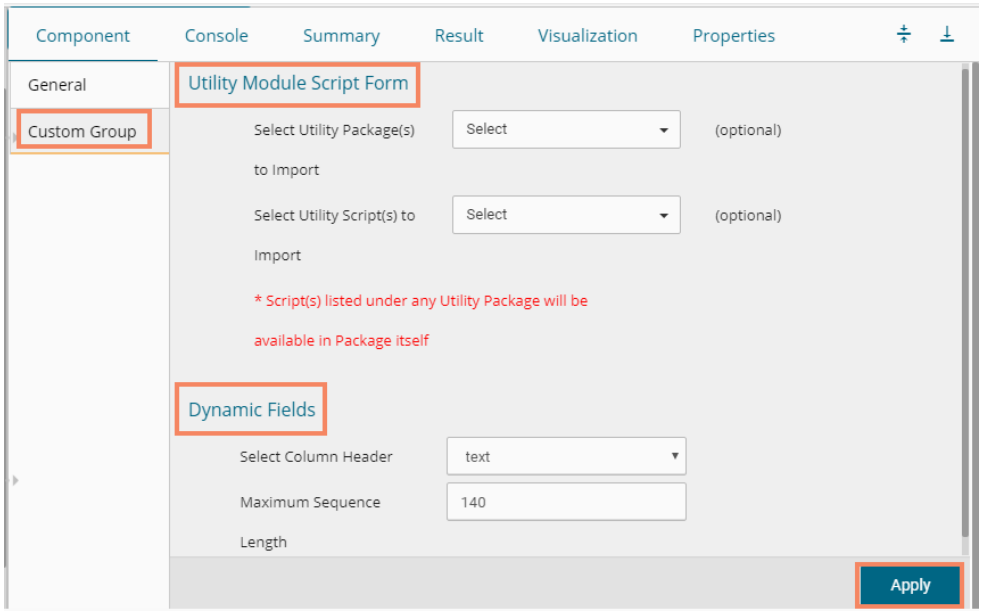


c) Script 2 (SA_2)

i. Configure the Custom Group options

1. Utility Module Script Form
   a. Select Utility (Package(s) to import (optional)
   b. Select Utility Script(s) to import (optional)
2. Dynamic Fields
   a. Select Column Header from the drop-down.
   b. Provide Maximum Sequence Length

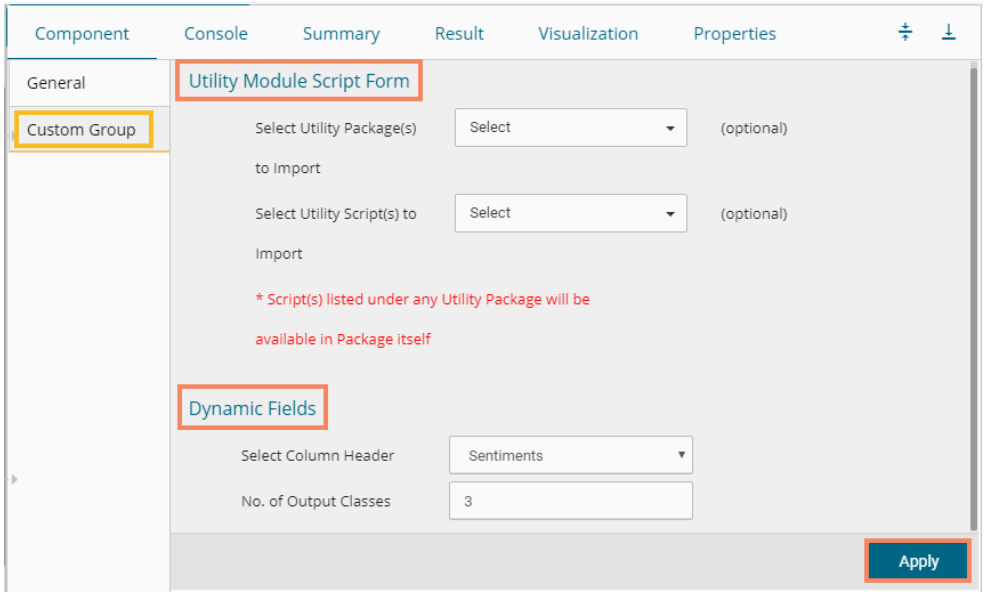
3. Click the ‘Apply’ option.



d) Script 3 (SA_3)

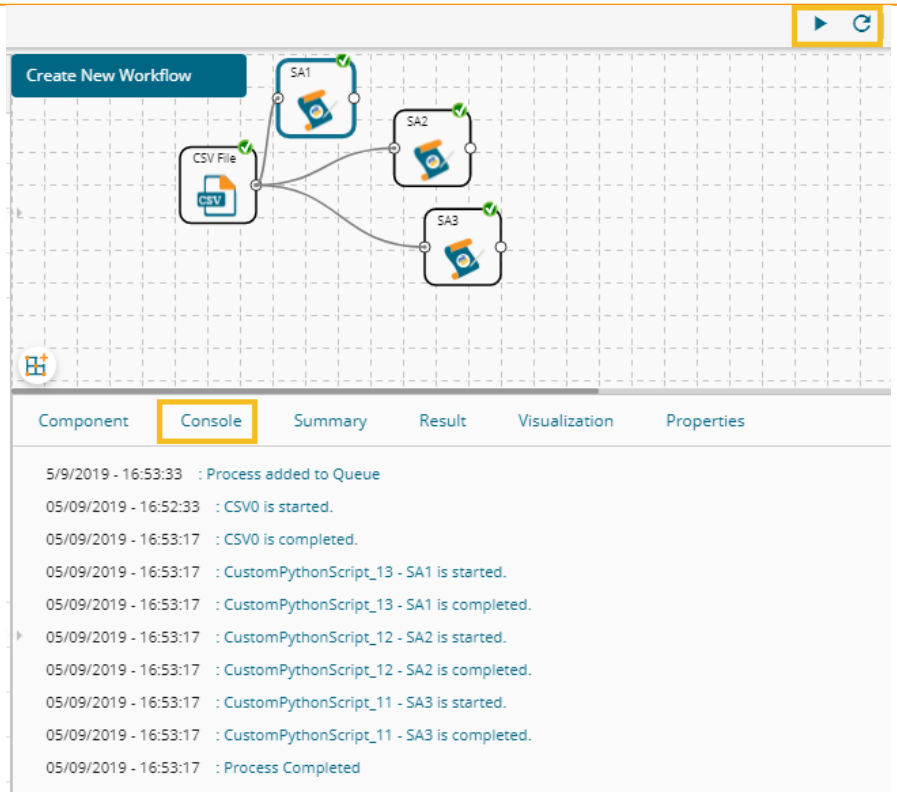
i. Configure the Custom Group options

1. Utility Module Script Form
   a. Select Utility (Package(s) to import (optional)
   b. Select Utility Script(s) to import (optional)
2. Dynamic Fields
   a. Select Column Header from the drop-down.
   b. No. of Output Classes.
3. Click the ‘Apply’ option.

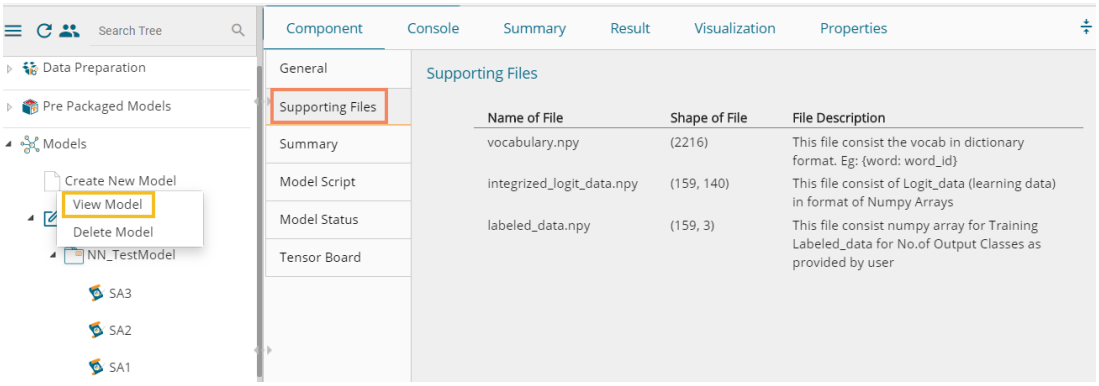


iii) Run the workflow.

iv) The completion of the process is marked with green checkmarks on the top of the dragged components.



v) The script files are listed under the Supporting File tab for the selected model.



12.2.4. Model Training

This part of the document describes the steps involved in the model training. The entire process of the Model, training involves '**Model Structure**' and '**Model Training**' sections. The user can create a Neural Network Model structure based on his/her problem statement. The user gets three options to form a structure for the selected model:

i) User Interface
ii) Script Editor
iii) Model Configuration

This section describes steps to create a Keras Model Structure using the preprocessed file details. The created model can then be used for training purposes.

i) Click the '**Model Training**' tree node.

ii) Configure the Model Selection fields provided under the '**General**' tab:

a. Select the NN Model for Training: All Created Neural Network Models list here. The user needs to select a Model for which it needs the training.

b. Re-train Model (if exist): Opt for this option if the selected model is already created and required to re-train the existing model

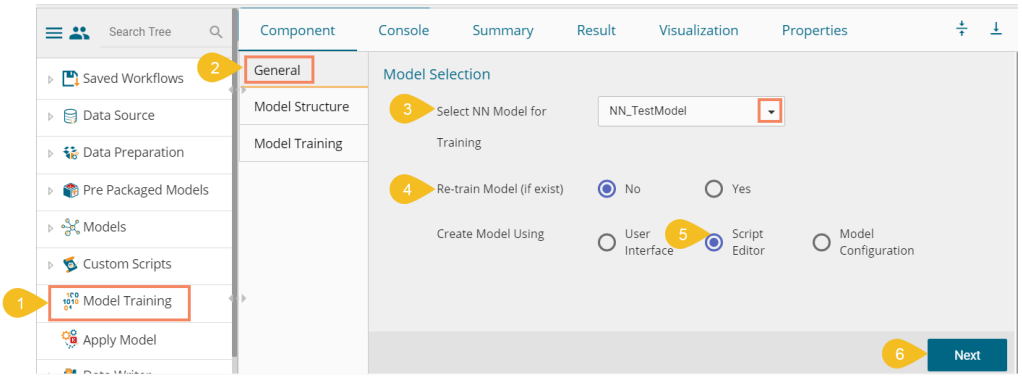
c. Create Model Using: Select a medium through which the model structure can be created

   i. User Interface

   ii. Script Editor

   iii. Model Configuration

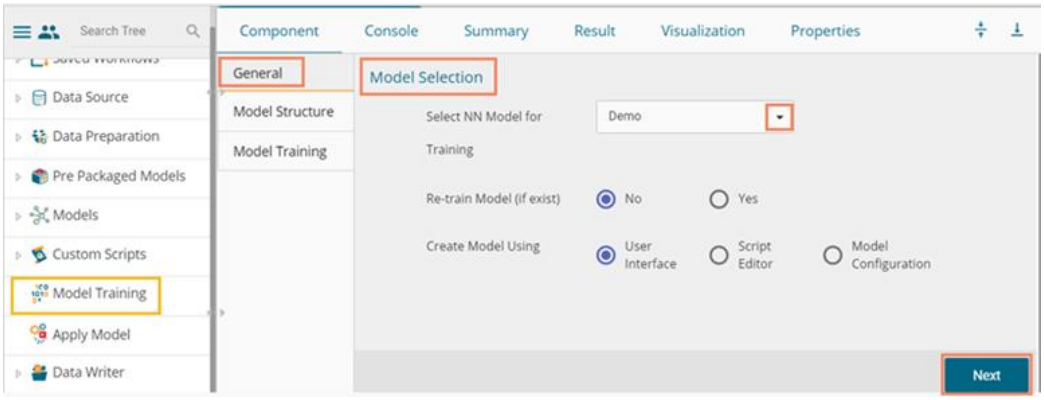
d. Click the '**Next**' option to proceed.



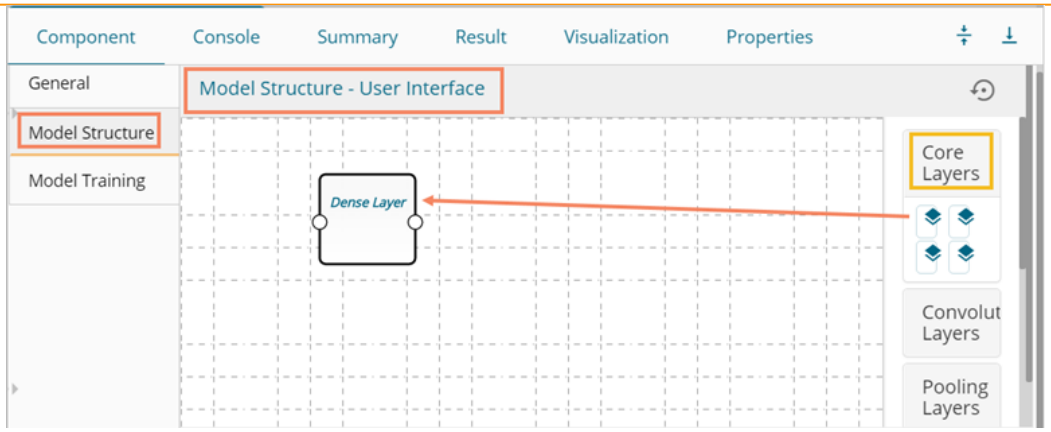
12.2.4.1. Create Model using User Interface

i) Select the '**User Interface**' as a model creation option.

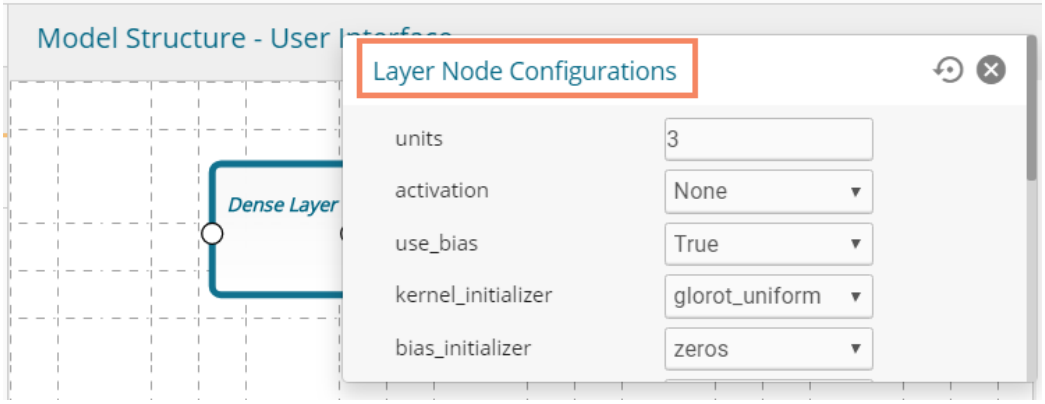
ii) Click the '**Next**' option.



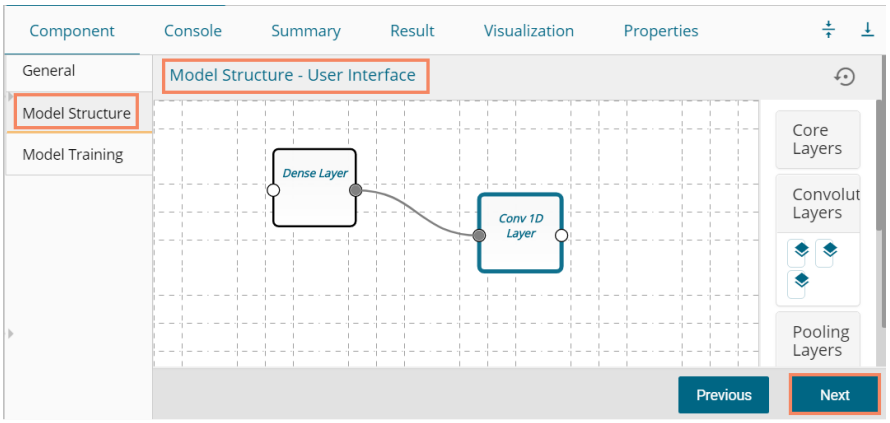
iii) The user gets another page to create the model by drag and drop of the various layers.



iv) The user needs to configure each of the dragged layers.



v) Click the '**Next**' option to proceed.



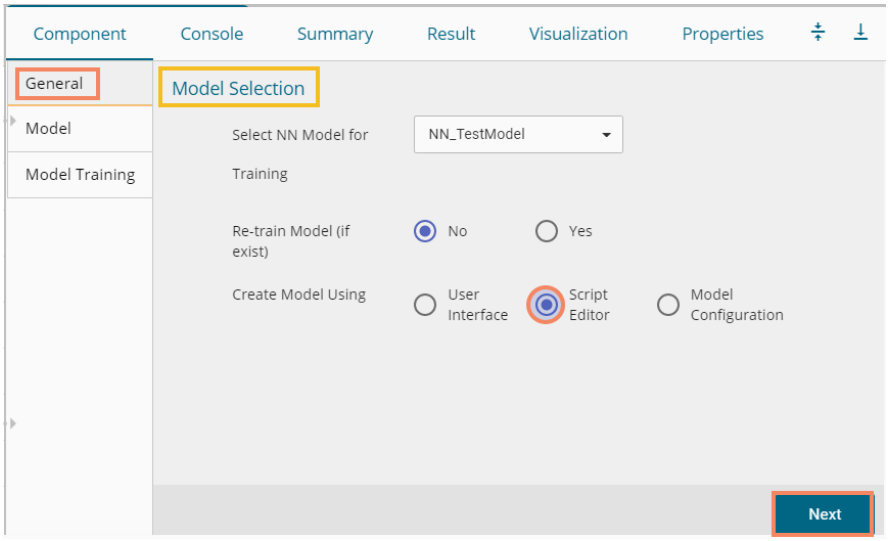
vi) If users have chosen the '**User Interface**' option to create a model, then a script for the dragged components display on this page. However, the users need to edit the script using the Script Editor to proceed further in the creation of a model.

vii) Validate the script. A success message should appear after script validation.

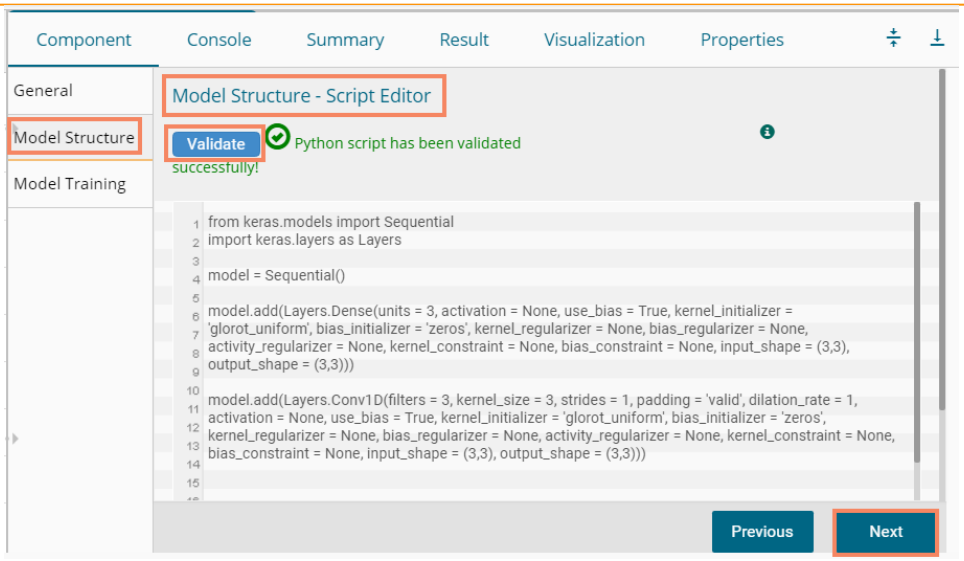
viii) Click the '**Next**' option to open the '**Model Training**' tab.

12.2.4.2. Create Model using Script Editor

i) Select the '**Script Editor**' as a model creation option.
ii) Click the '**Next**' option.



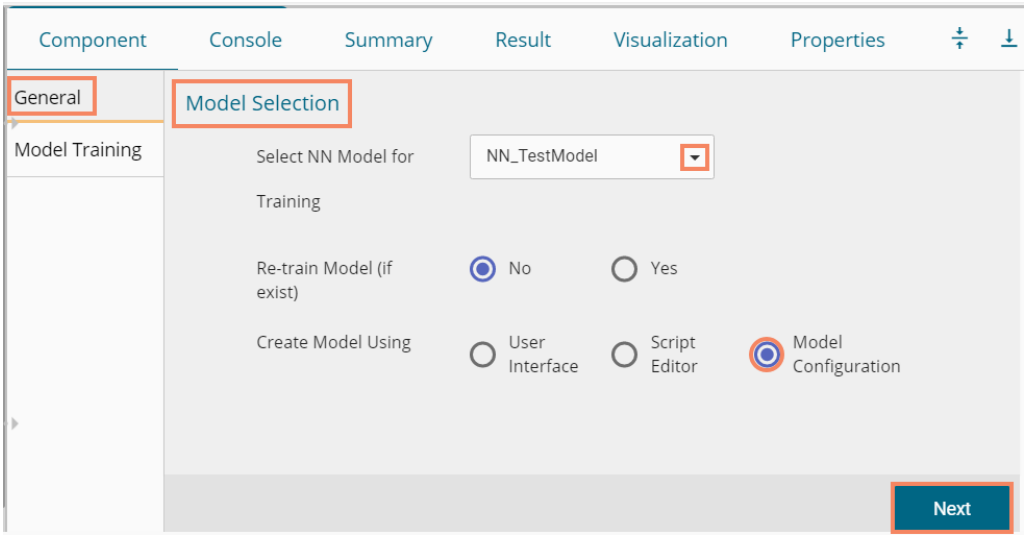
iii) The '**Model Structure**' tab opens displaying the **Script Editor**.
iv) Provide a relevant python script.
v) Validate the script. The success message should appear after the script validation.
vi) Click the '**Next**' option to open the Model training tab.



12.2.4.3. Create Model using Model Configuration

This option can be used for the object deduction models only.

i) Select the '**Model Configuration**' as the Model Creation option.
ii) Click the '**Next**' option.



iii) Selecting the Model Configuration option redirects the user to the '**Model Training**' page with the Model Configuration fields displayed below
iv) Configure the required fields.
   a) Model Reference for Training: Select a reference model that can be used to refer to the inputs (At present, it displays the RCNN Inception model only).
   b) Number of Classes: Provide the value (number) of distinct classes present in your training data.
   c) Number of Steps: Define the number of steps required for training.
   d) Initial Learning Rate: Provide the value of the learning rate to start the model training (it should be in 0.00 to 1.00 where 0 and 1 are included).
   e) Momentum Optimizer Value: Provide value for the optimization function (it should be in 0.00 to 1.00 where 0 and 1 are included).

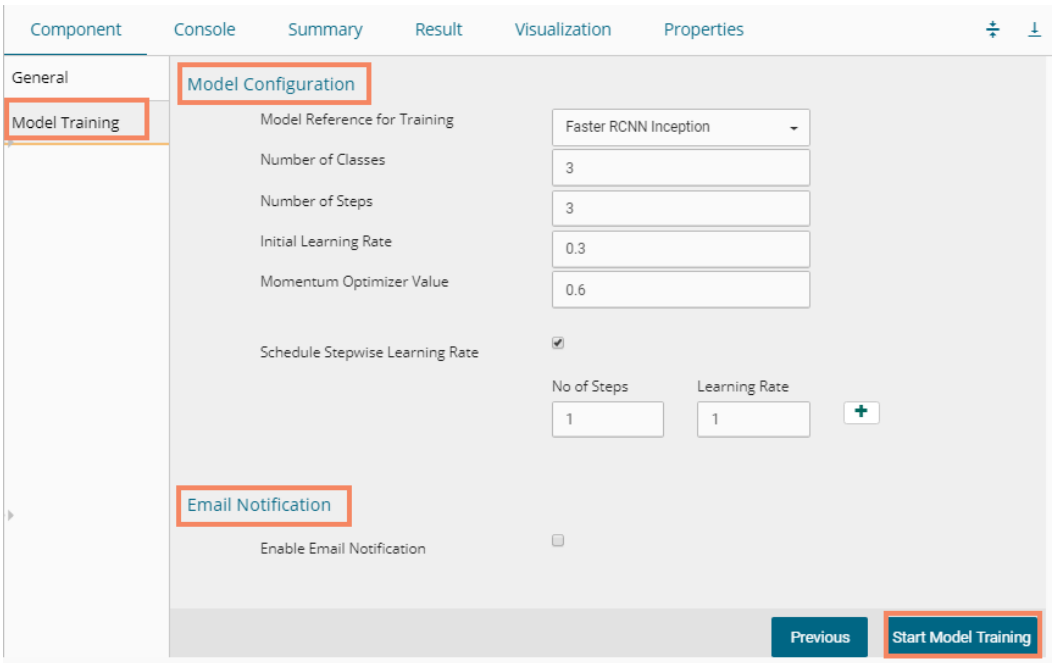
f) Schedule Stepwise Learning Rate: enable this option if you wish to schedule the stepwise learning rate.

g) After enabling the Schedule Stepwise Learning Rate, the user gets to configure the following options

1. No of Steps
2. Learning Rate

v) Enable the '**Email Notification**' option and provide the required information for the same.

vi) Click the '**Start Model Training**' option to begin with the model training.



Note:

a. The '**Model Structure**' tab does not appear if the selected option for creating the model is **Model Configuration**.

b. If the selected model is already undergoing training, it throws an error message.

12.2.4.4. Model Training Tab

This section describes steps to select and interpret the variable files to proceed with the model training.

The user can interpret Logit File as independent variables data, which is preprocessed already, and Label File as target (or labeled) data. The selected model learns using the Label File data over the Logit File data and builds up weights internally, which can be used for prediction using the trained model.

i) Navigate to the Model Training tab using the Model Training tree-node.

ii) Configure the required fields to Train Model:

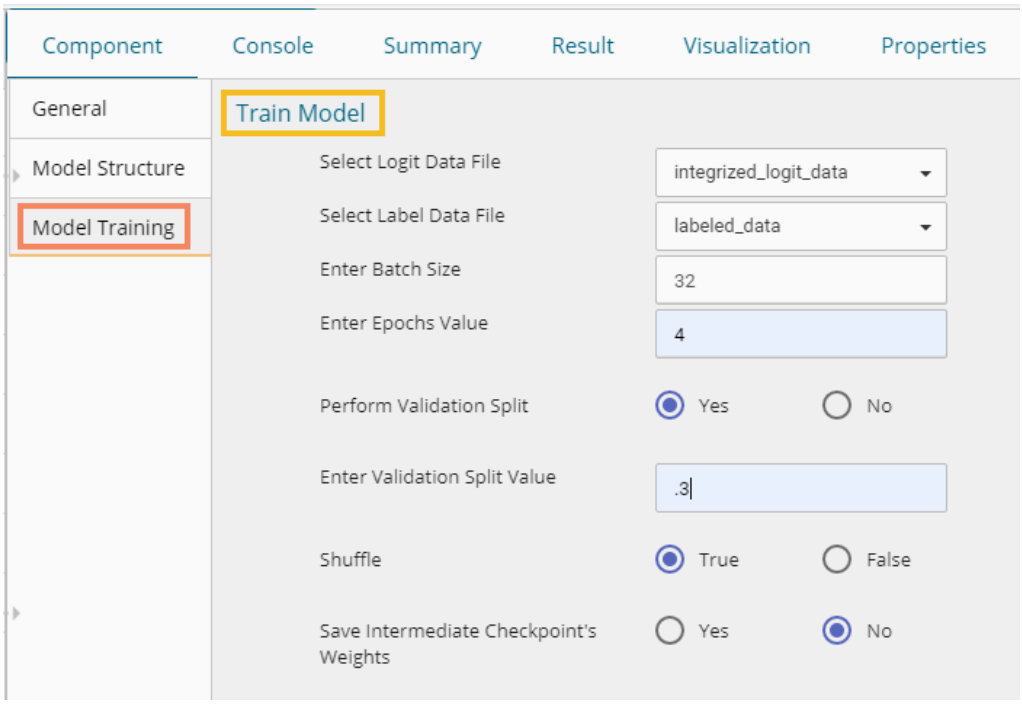
a. Select Logit Data File: Select the file with logit data using the drop-down option.

b. Select Label Data File: Select the file with labeled data using the drop-down option.

c. Enter Batch Size: Enter a value for batch size

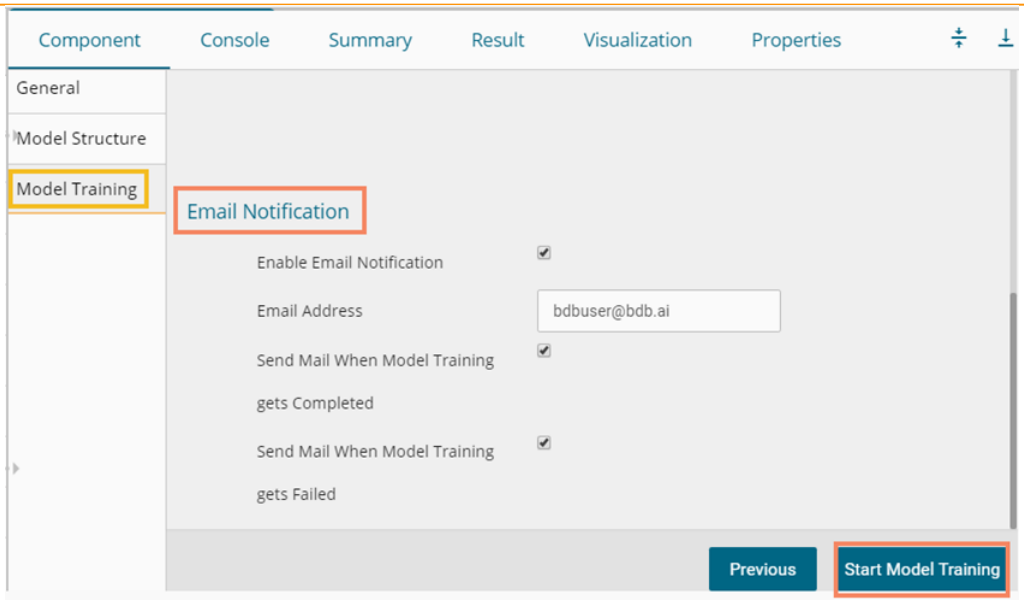
d. Enter Epochs Value: Enter Epochs Value (the suggested value for this field is 4)

e. Perform Validation Split: Select an option out of **Yes/No**
f. Enter Validation Split Value: Enter a value indicating the validation split (the suggested value for this field is .3)
g. Shuffle: Select an option out of **True/False**.
h. Save Intermediate Checkpoint's Weights: Select an option out of **Yes/No**.



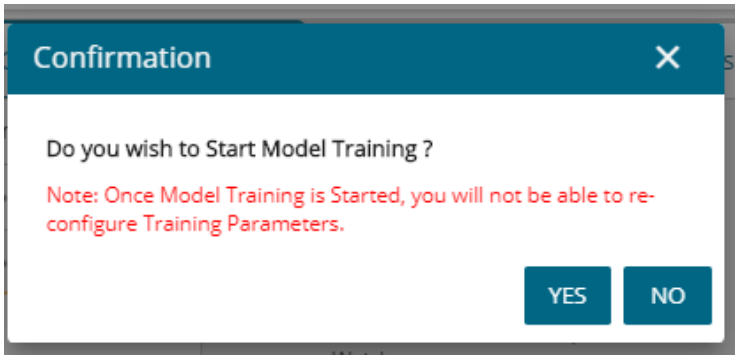
iii) Configure the following fields to send Email Notification for success or failure of the model training.
   a. Enable Email Notification: Enable the box to get email notifications.
   b. Email Address: Provide a valid email address where the notification can be sent.
   c. Send Mail when Model Training gets Completed: Enable this option if you wish to get notified when the Model training gets completed.
   d. Send Mail when Model Training gets failed: Enable this option if you wish to get notified when the Model training gets failed.

iv) Click the '**Start Model Training**' option to begin the training.



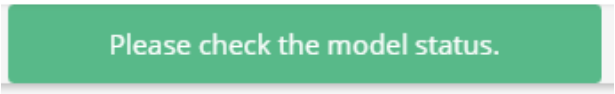
v) A dialog box appears to confirm the action of Model Training.

vi) Click the '**YES**' option to confirm the model training.



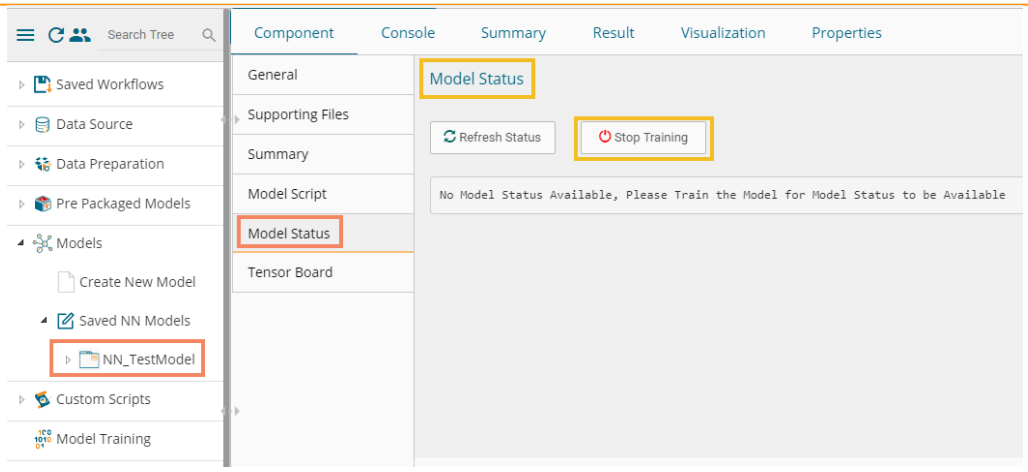
vii) A notification message appears asking the user to check the model status.

viii) Once the model is trained successfully, the user can use the model for prediction purposes.



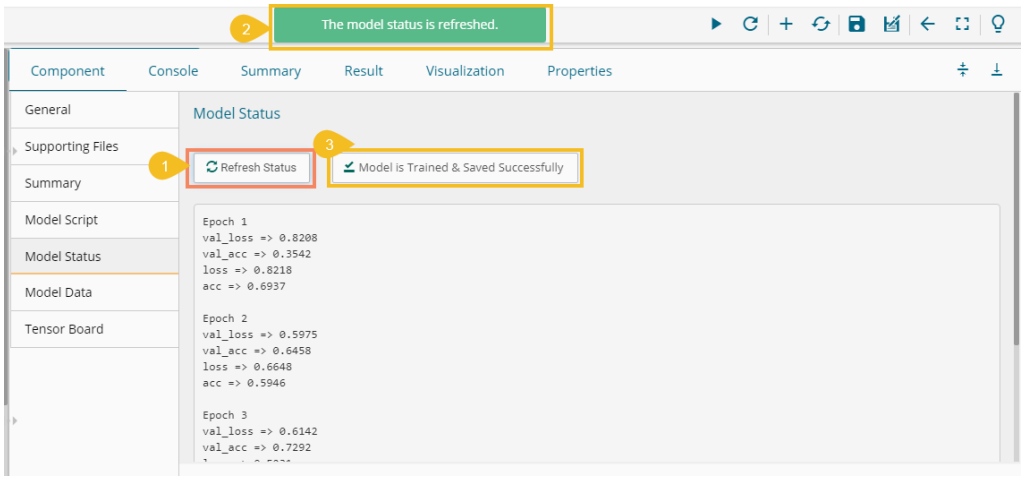
ix) Navigate to the Model Status tab given for the created model folder under the Saved Model tree node.

x) The '**Stop Training**' option appears for the model that is undergoing training.

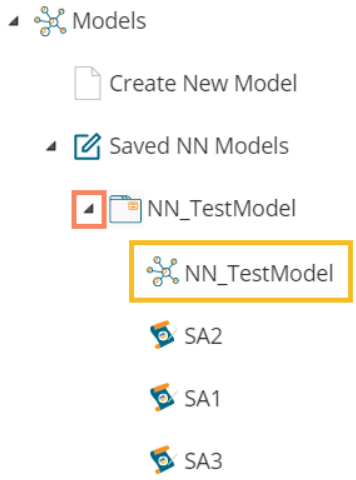


xi) Click the '**Refresh Status**' option if the user needs to refresh the model status.

xii) The '**Model is Trained & Saved Successfully**' message appears for the model once the training gets completed.

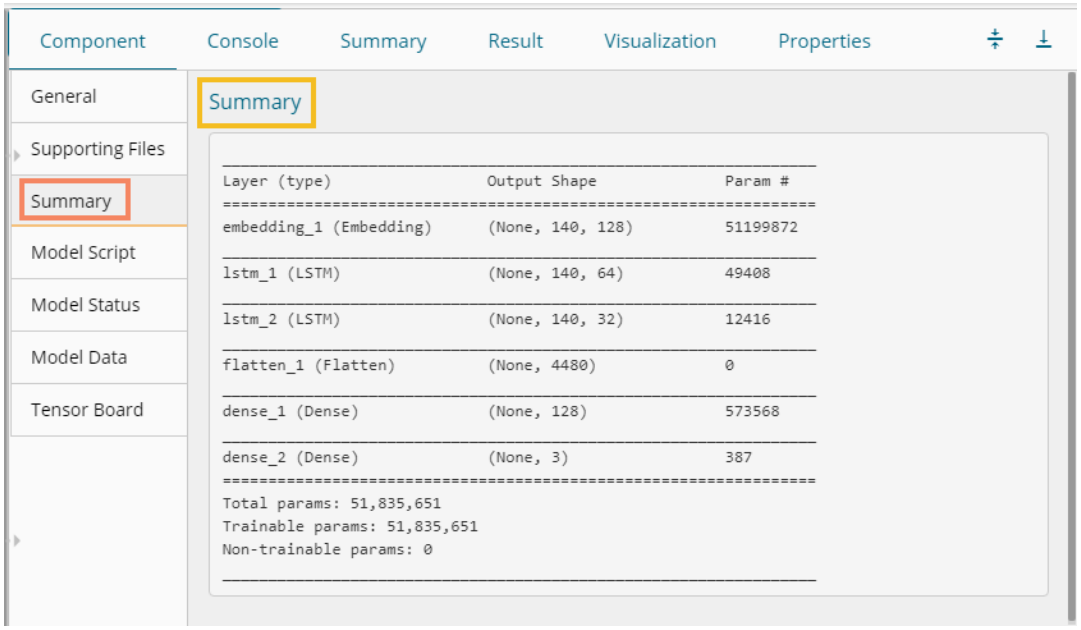


xiii) After successful completion of the model training, the trained NN model gets added to the created model folder containing the same folder name.

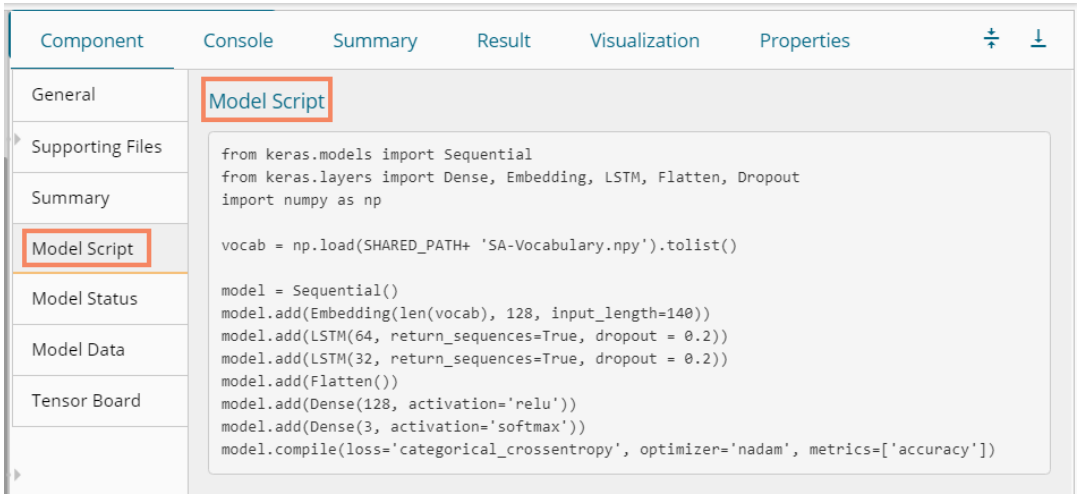


Note:

a. The selected Logit and Label data files should not be the same.
b. Users can provide details of Batch Size, Epochs, Validation Split as per the model requirement.
c. Users can track the status of the Model for each epoch, including visual tracking using Tensorboard when the model is undergoing the training process.
d. Users can stop the model training in between during the period when the model training process is going on.
e. Users cannot process a Neural Network Model for Model Training if it is already in between the training process.
f. The user must provide specific parameter values for Model Training purposes.
g. Since training a model is a time-consuming task, the user can set the Model for training and provide email details to get a notification when the training gets finished or if an error occurs.
h. Click the 'Summary' tab to view the model summary using the 'View Model' option provided for the selected NN Models. The Summary appears for the trained model.



i. Click the '**Model Script**' tab to view the Model script using the '**View Model**' option provided for the Saved NN Models. The Model Script appears for the trained model.



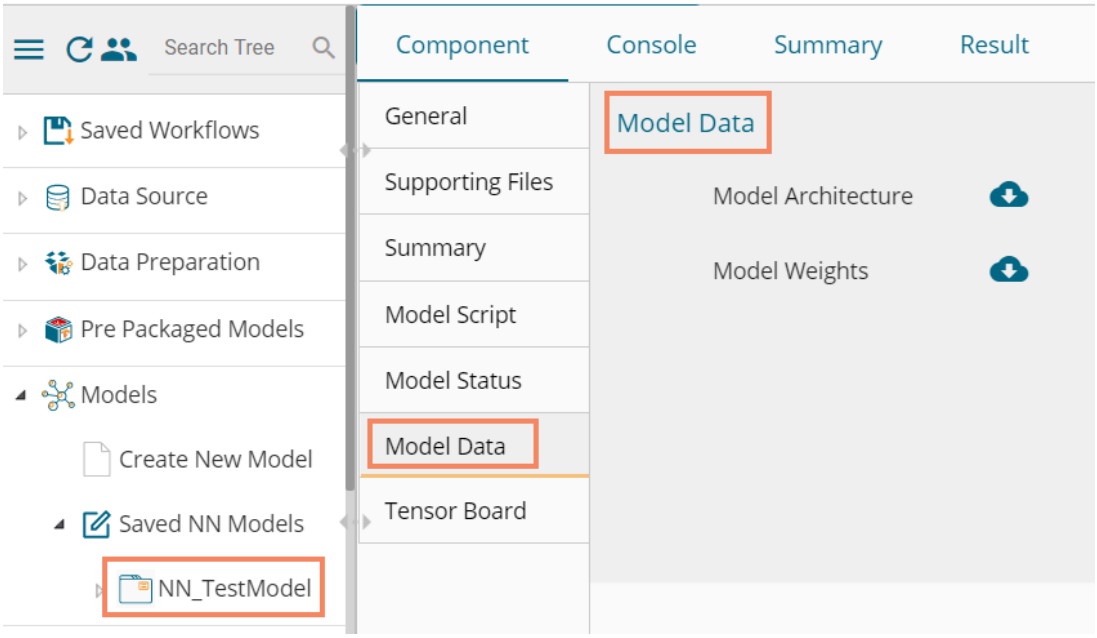
j. Please note that the above given '**Model Training**' fields display only when the model creation option is either **User Interface** or **Script Editor**. The Model Training tab displays different fields when the model creation option is '**Model Configuration**' (which has been already explained within section 10.2.4.3)

12.2.5. Model Data

The user can see the Model Data tab with the Model Architecture and Model Weights options, both provided with the download option.

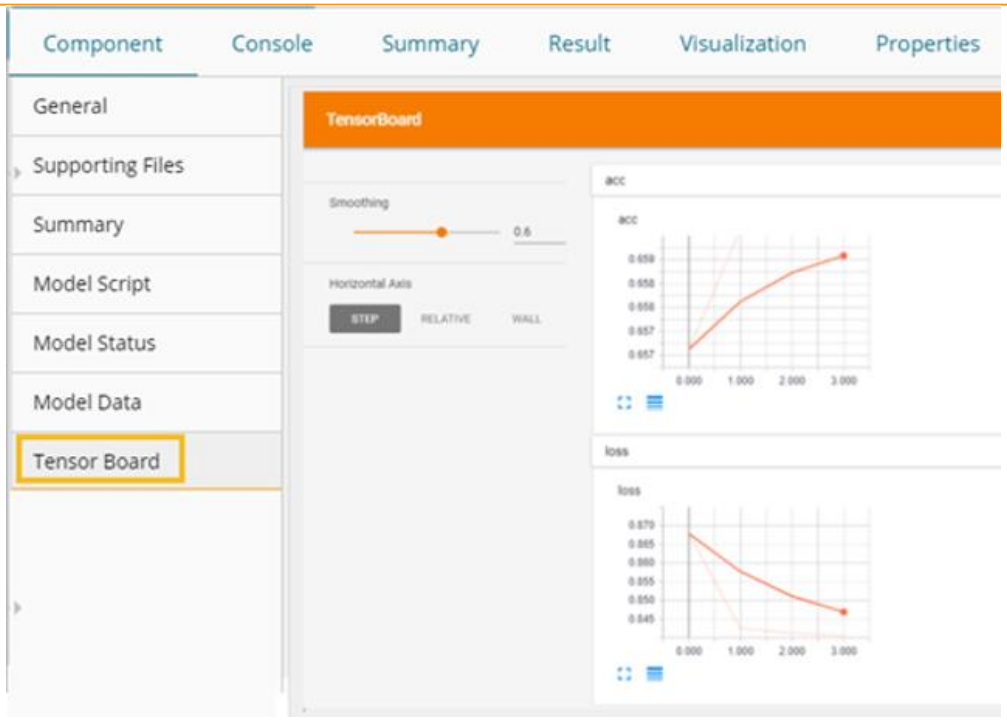
Model Architecture: It is metadata for the selected model. It contains the details of layers and the configured parameters. The architecture file gets downloaded in the JSON format as it is a simplified way to recreate the model from JSON with Keras API.

Model Weight: The Model Weight option consists of Resultant assigned weights for each layer present in the model architecture during training and/or after the training is completed. The model weights file gets downloaded in the .h5/HDF5 as it is suitable to store multiple data types and extensive data. It can be loaded over a model using the Keras API.



12.2.6. Tensor Board

This tab displays live Tensor Board Visualization for the selected model (if enabled). The below image displays a sample visual for the reference of the user.



12.3. Apply Model

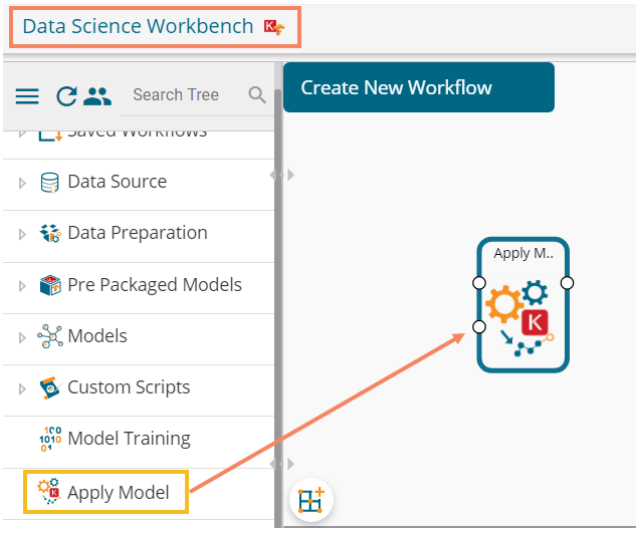
This component is provided to generate predictions based on NN trained model. The user can view predicted column value for each label class.

Users can create an NN Apply Model in the following ways:

- Generate a model by pre-processing the selected data and training the model based on the created structure.
- Generate a new NN Apply Model using the saved NN model

The Apply Model within the Deep Learning Workspace 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

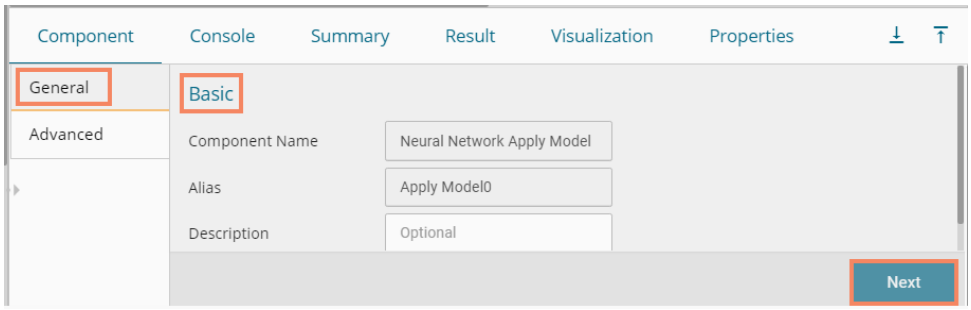


i) Drag the Apply Model component onto the workspace and connect it with a valid combination of data source and other components to create a valid workflow.

ii) Click the dragged '**Apply Model**' component.

iii) Basic component details get displayed.

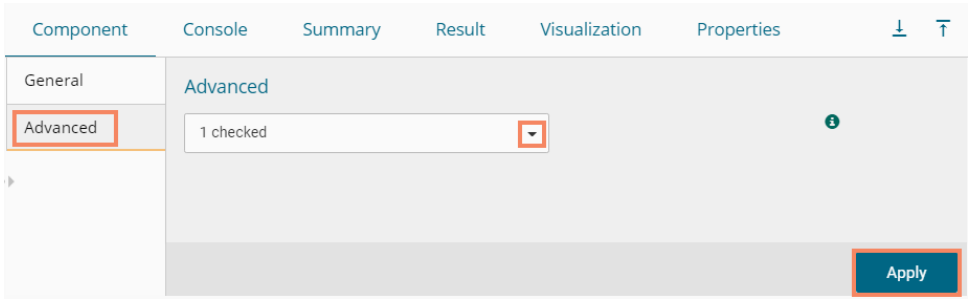
iv) Click the '**Next**' option to move ahead.



v) The Advanced tab opens.

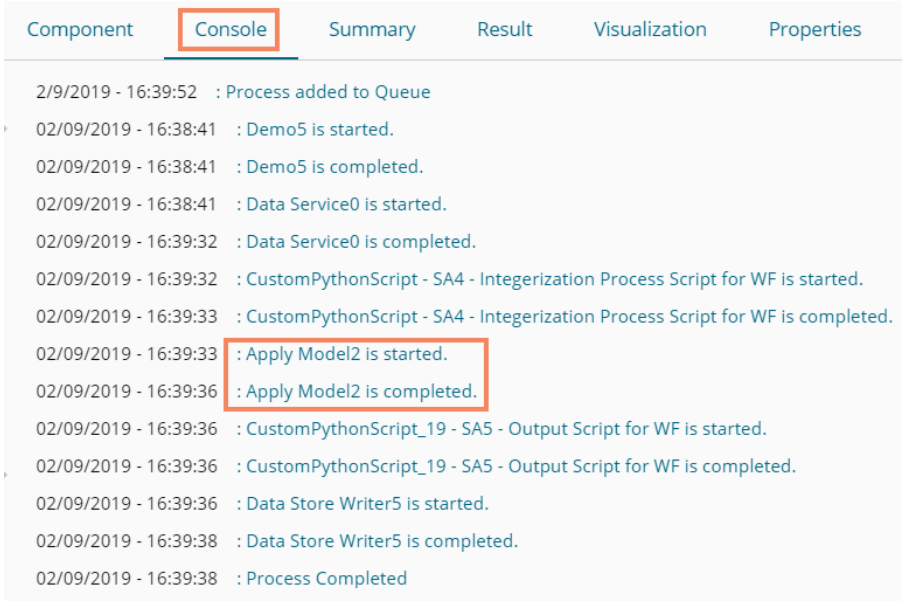
vi) Select the required columns from the drop-down menu.

vii) Click the 'Apply' option.

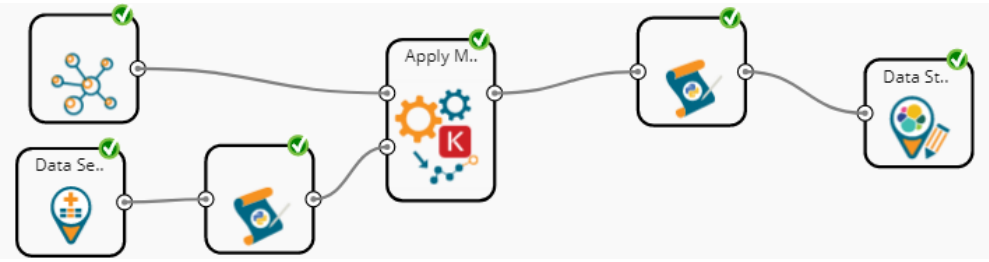


viii) Run the workflow after getting the success message.

ix) The process status gets displayed under the '**Console**' tab.



x) The completion of the Console process gets marked with green checkmarks on the top of the dragged components.

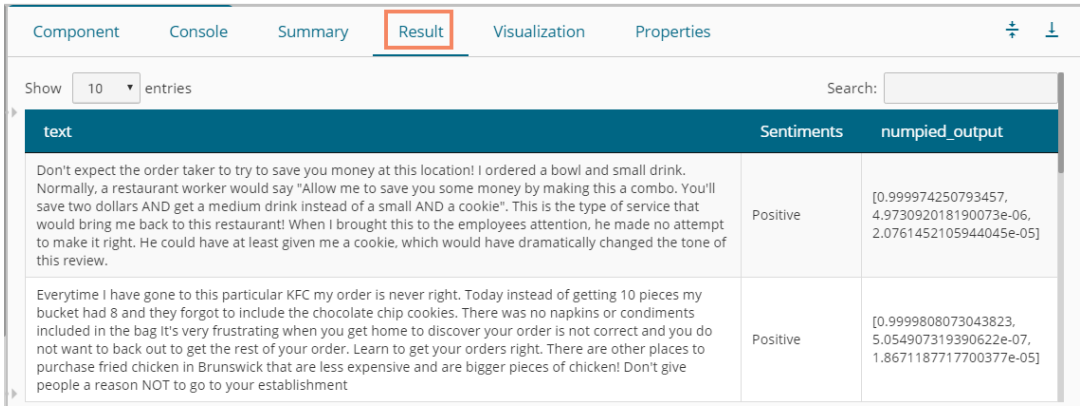


xi) Follow the below-given steps to display the Result view:

a. Click the dragged Apply Model component on the workspace.

b. Click the 'Result' tab.

xii) The columns displaying numpied_output probability get added to the Result view. The Apply Model displays the Result in the array format.



Note:

a. The user can connect the Apply model output to a related Python script to convert the predicted output from the array format to the predicted class Output.

b. The Result data set of the model can be written to a database using a Data Writer.

c. The Column header and data type of feature column should match for the saved model and testing data. If column headers and data types do not match, an alert message will be displayed.

d. It is not mandatory for the testing data set to contain a label column.

e. The user can view the model summary by clicking on the '**Summary**' tab.

12.4. Prediction using Trained Models

Users can use the Saved NN Model in a workflow as displayed below for the prediction purpose:

i) Select and drag a Data Source onto the workspace.

ii) Using Custom Python Script Component, create a script that can pre-process the data and transform the input Data Source data into a consumable form by the Apply Model component.

iii) Drag a trained Neural Network Model and configure it.

iv) Drag and Apply Model component. The Apply Model provided for the Deep Learning workspace is the same as the Apply Model component provided for the other workspaces; the only difference is in this, the user needs to select the Column Headers on which the Model predicts the values.

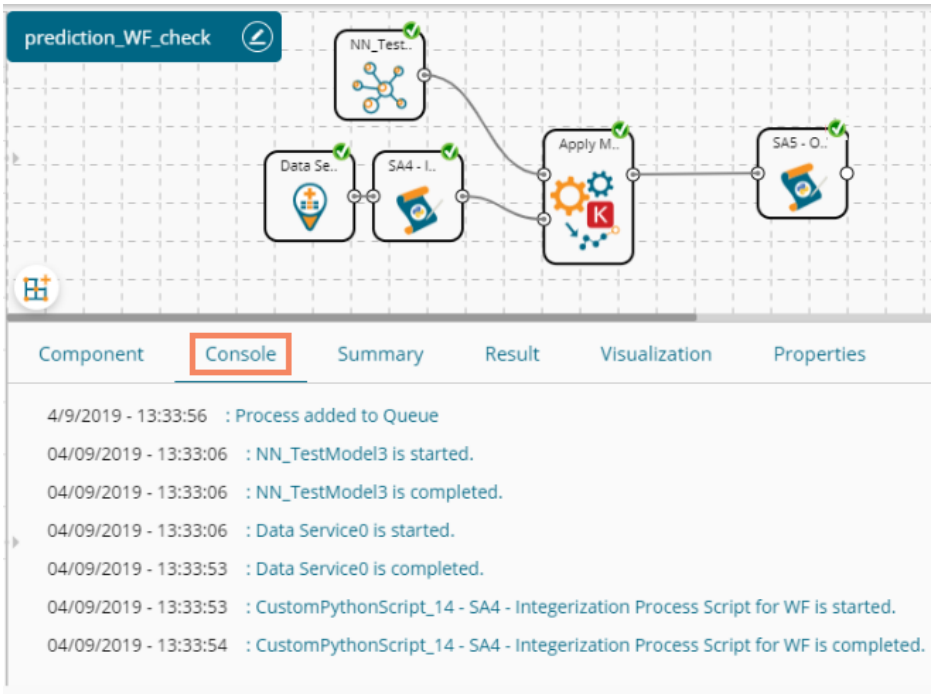
v) After NN Apply Model, put a Custom Python Script to reverse the transform implemented by the previous script component turns the predicted values into the Predicted class Output.

vi) The predicted output can be written to a Data Writer (in this case, it is the Data Store writer)

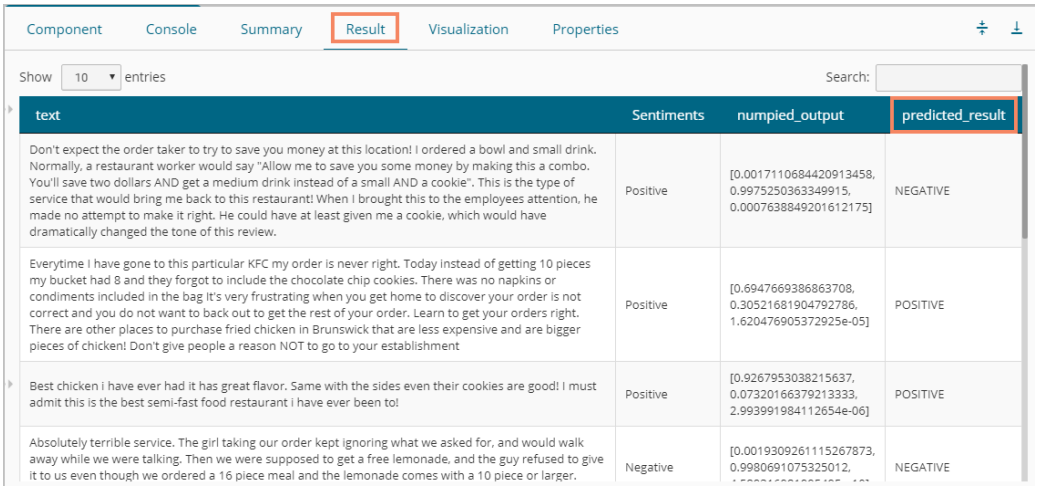
vii) Run the workflow by clearing the previous cache.

viii) The steps of the Console process get displayed under the 'Console' tab.

ix) The completion of the Console process is marked by the green checkmarks on the top of the dragged components.



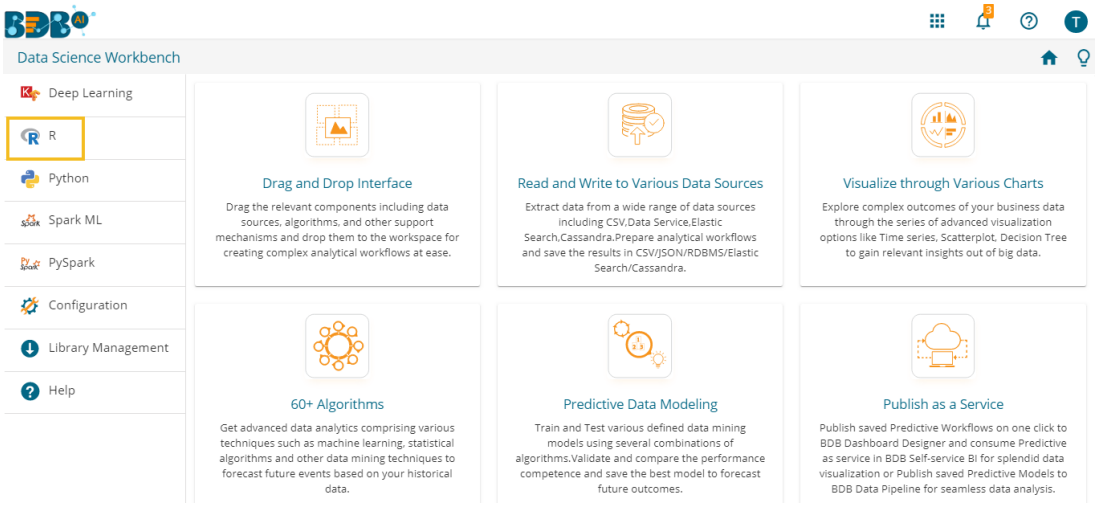
x) The processed data appears under the 'Result' tab.



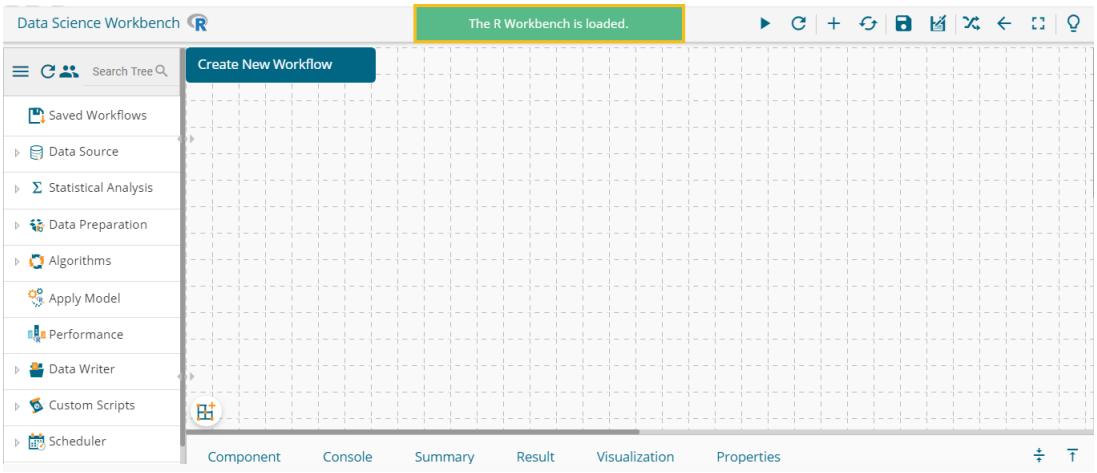
13. R Workspace

This section of the document describes the R environment by focusing on the Statistical Analysis, Data Preparation connectors, Algorithms, Apply Model, Performance, and Custom R script components to build an R workflow under the Data Science environment.

The user can select the R Workspace from the Predictive landing page to access the R Environment under the BDB Data Science Workbench.



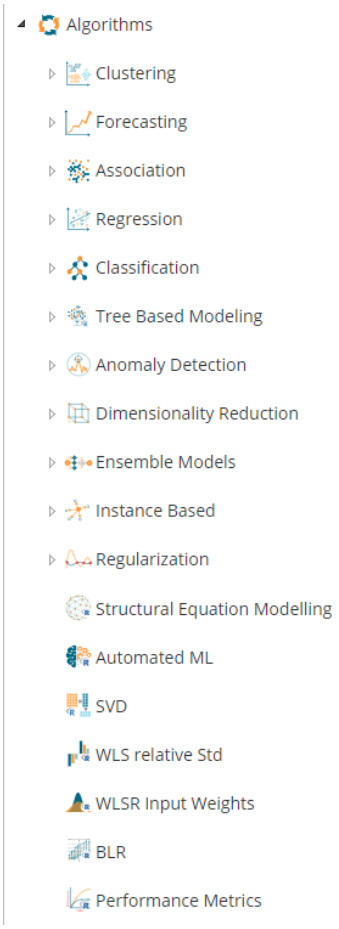
The user gets redirected to the following page by selecting the R Workspace:



13.1. Algorithms

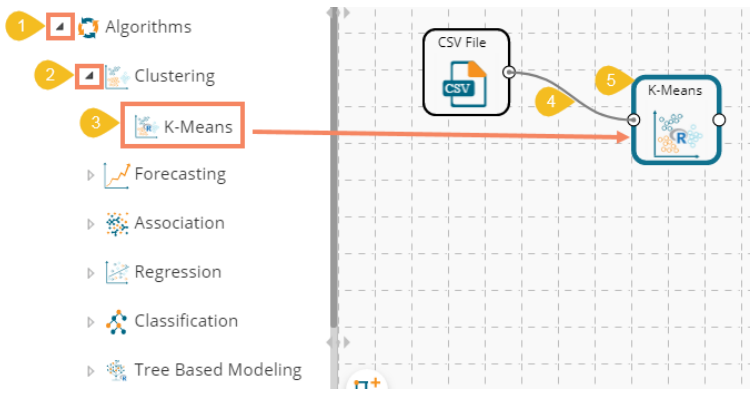
Algorithms are a statistical set of rules that help users analyze vast quantities of numerical data and extract appropriate information out of it. The Data Science Workbench allows users to apply more than one algorithm to manage the enormous amount of data.

The following algorithms are available in the R Workspace:

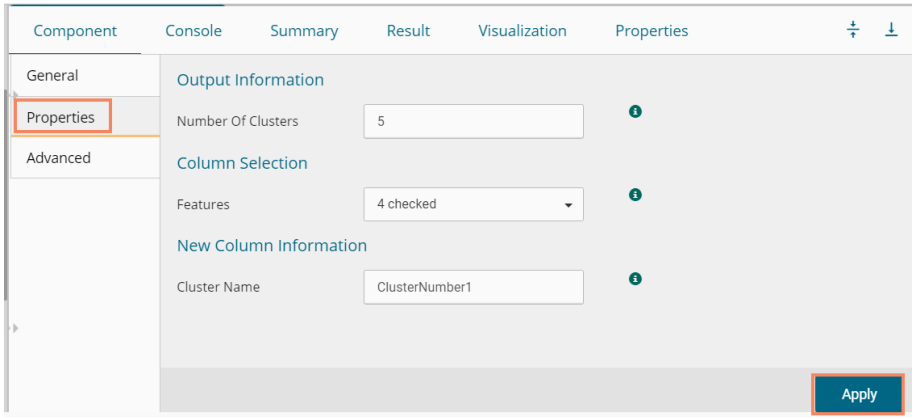


Step by Step Process to configure and connect an Algorithm in the R Workspace:

i) Click the '**Algorithms**' tree-node on the Predictive Analysis home page.
ii) Click the Algorithm Category tree-node to display the available algorithm subcategories.
iii) Select and drag an algorithm component onto the workspace.
iv) Connect the algorithm component to a configured data source.
v) Click on the algorithm component.

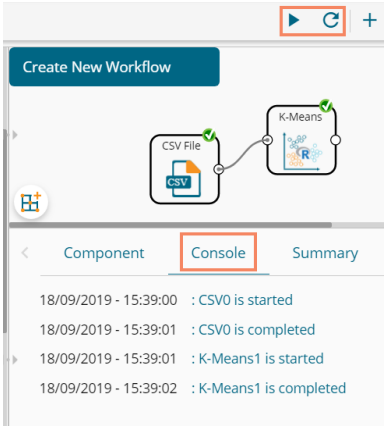


vi) Configure the following '**Component**' fields for the dragged algorithm component.
vii) Click the **'Apply'** option to save the information.



viii) Run the workflow.

ix) The '**Console**' tab opens displaying the step-by-step completion of the process. The green marks on the top of the dragged components mark the completion of the Console process.

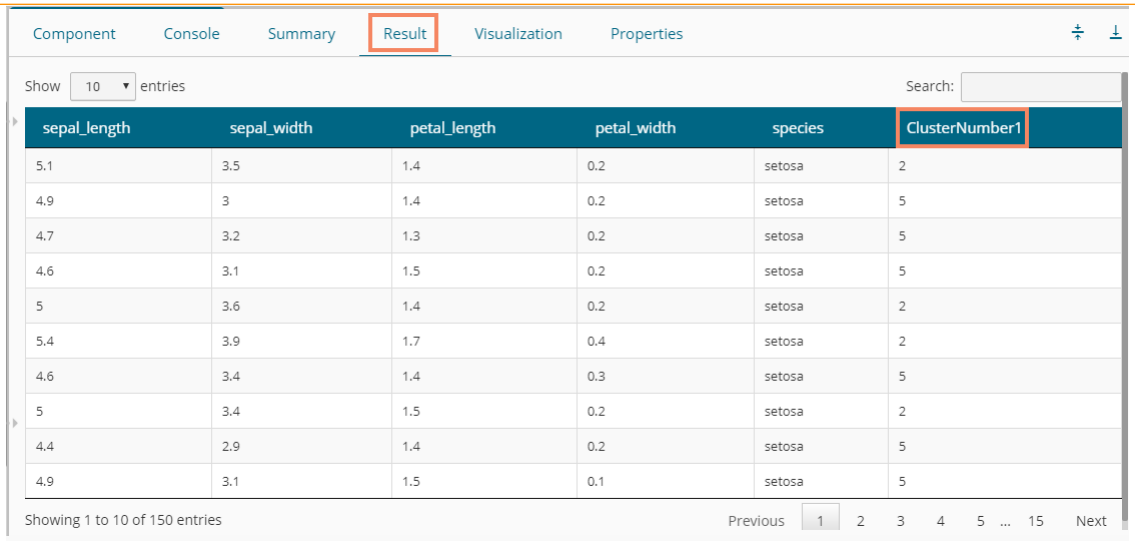


x) After the Console process gets completed, the user can view Result data using the 'Result' tab.

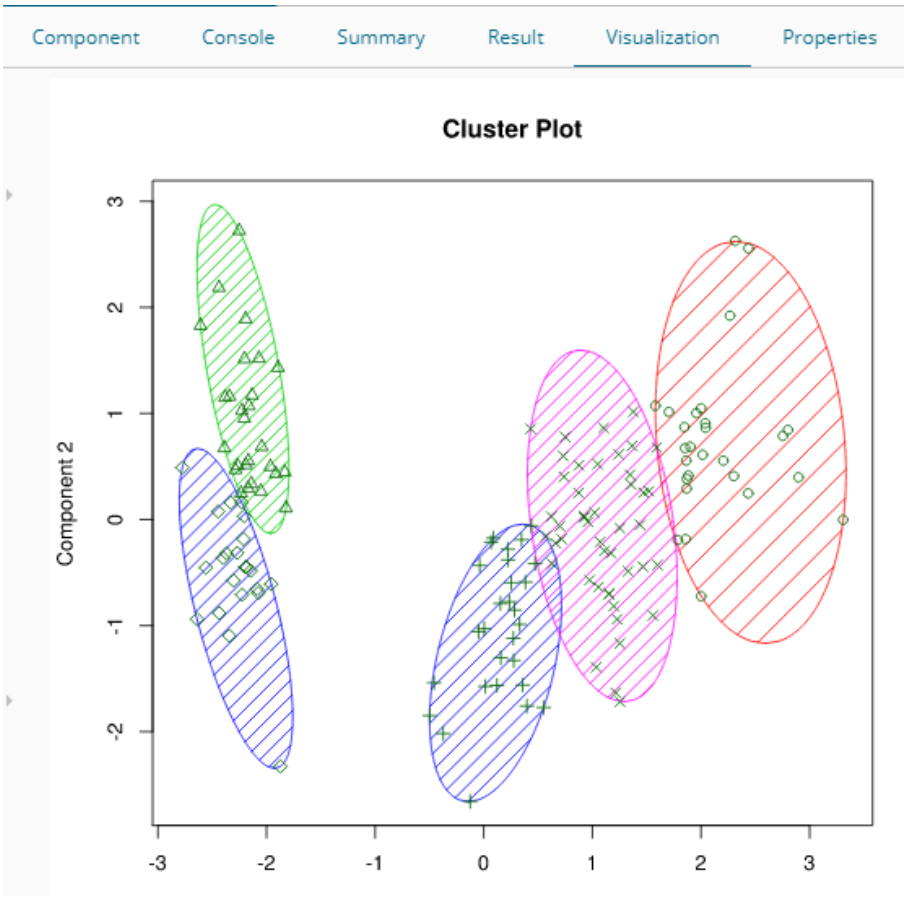
a. Click the algorithm component on the workspace.

b. Click the 'Result' tab.

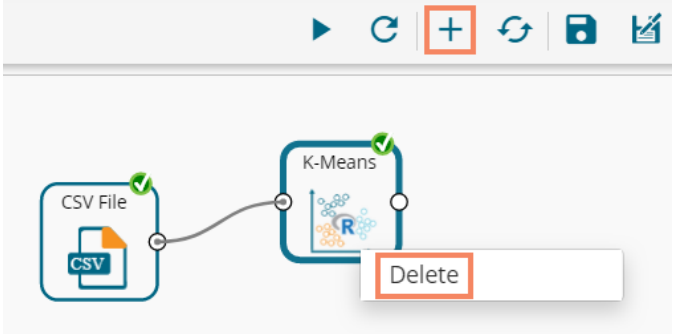
xi) The newly created Cluster Column gets added to the displayed Result dataset.



xii) Click the 'Visualization' tab to see a graphical representation of the Result data.



xiii) Click the **'Delete'** option or the icon for the **'Create New Workflow'** option to remove the selected algorithm component from the workspace.



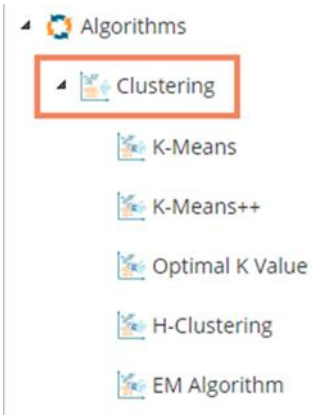
Note:

a. The user can follow the steps mentioned above to configure all the available R- algorithms.
b. The user can configure the alias name for the algorithm component via the '**General**' tab.
c. The basic configuration for all the algorithms is done through the '**Properties**' tab. The user is required to configure this tab while Applying an algorithm component manually.
d. The user can avail of all the default values under the '**Advanced**' tab. The user can manually set the '**Advanced**' tab or modify the default values, only if the advanced level configuration is required.
e. After execution, The user can click on the respective component to get data. The pipeline component does not have any Result set; it has only the summary to see. Users need to connect the pipeline components with an '**Apply Model**' component and test dataset to view the Result.

13.1.1. 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).

The Clustering tree node contains the below-given algorithms:



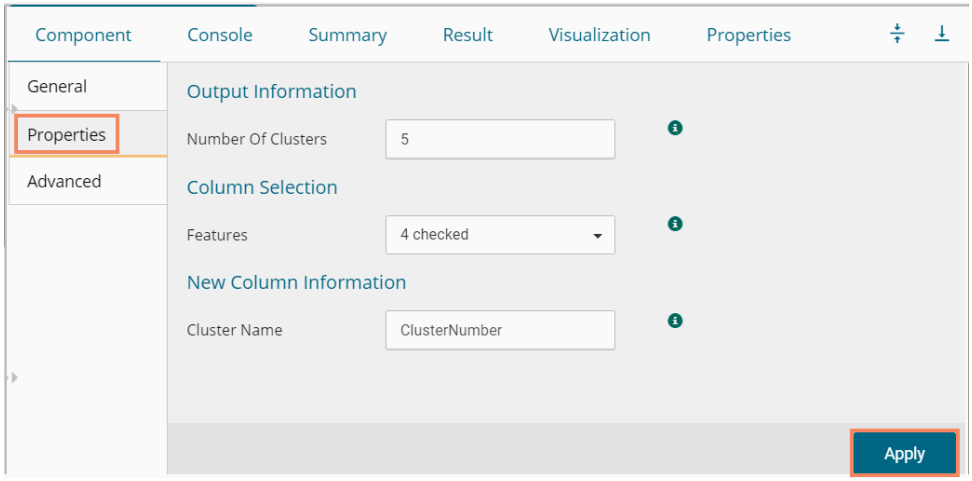
13.1.1.1. K-Means

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 get displayed on the View space.
iii) Configure the following fields in the '**Properties**' tab:
   a. Output Information
      i. **Number of Clusters**: Enter the number of groups for clustering. The default value for this field is 5. The range should be between 1 and the total number of clusters.
   b. Column Selection
      i. **Features:** 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**.**



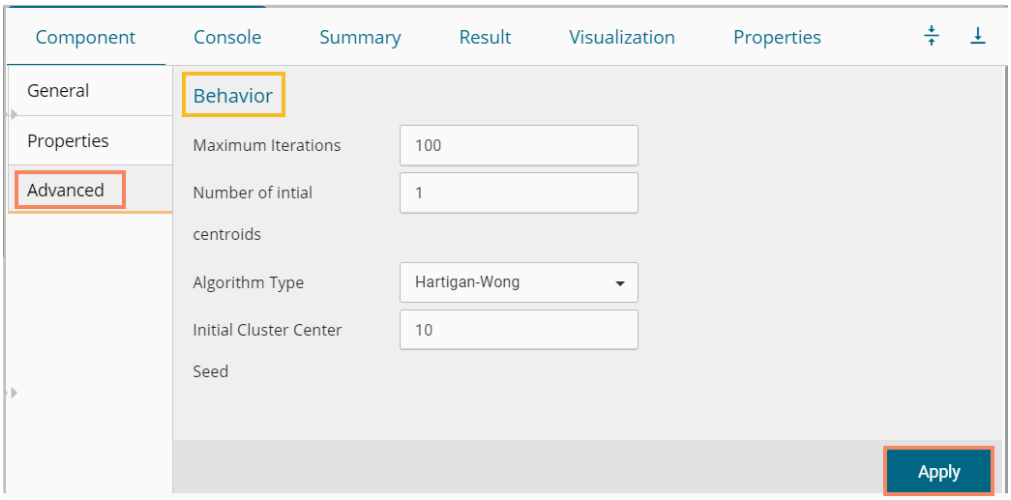
- **Rules for Naming a New Column**
   1. Do not use space in the name of a new column. It should be a single word, or two words should be connected by an underscore (_). E.g., SampleData or Sample_Data.
   2. Do not use any special symbol alone or with any character as the name of a new column. Eg.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

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

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

iv. **Initial Cluster Center Seed**: Enter a number indicating the initial cluster center seed (The default value for this field is 10).

v) Click the **'Apply'** option**.**



vi) Run the workflow after getting the success message.

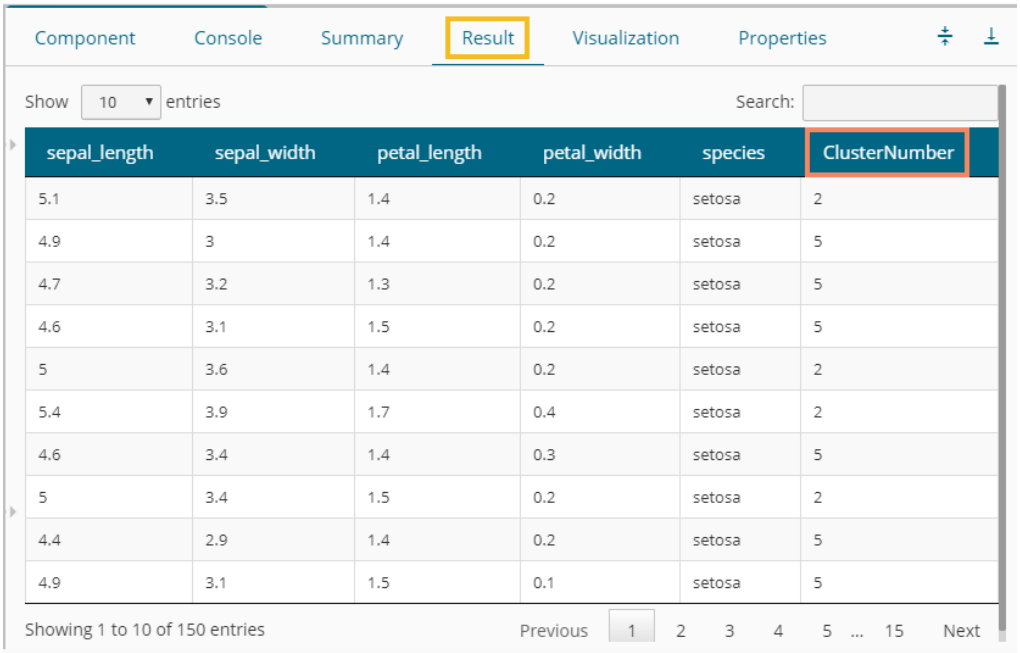
vii) The '**Console**' tab opens describing the progress of the process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged component.

viii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

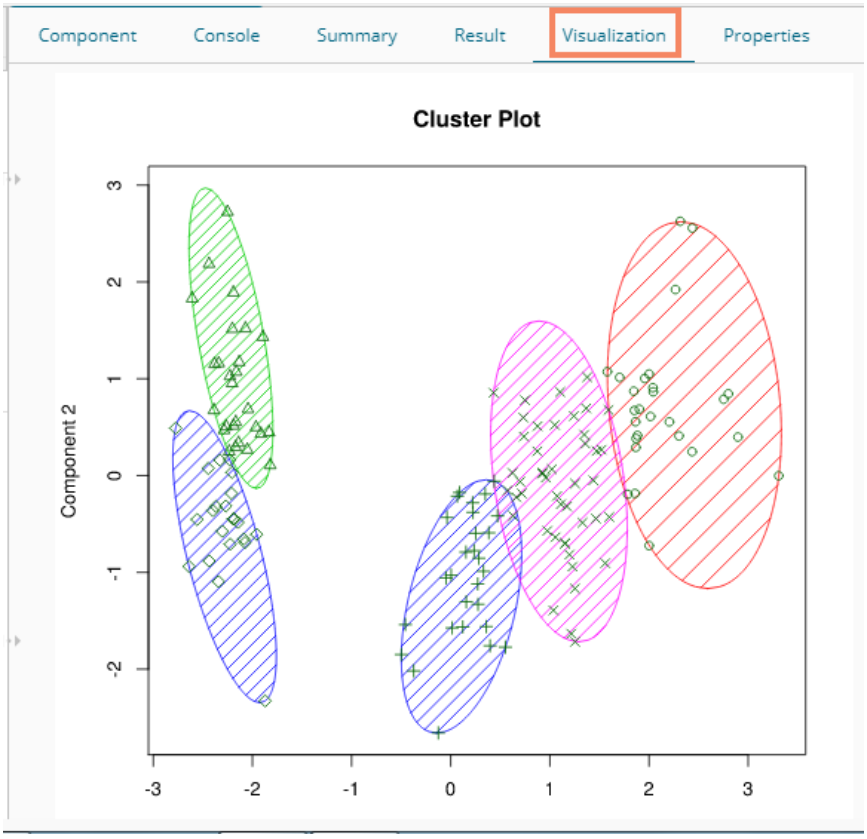
b. Click the 'Result' tab.

ix) A new column '**Cluster Number**' gets displayed in the Result view.



x) Click the 'Visualization' tab.

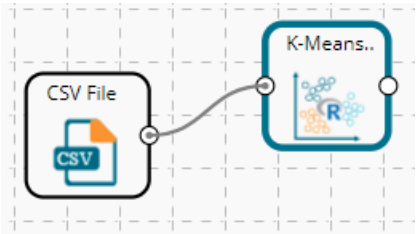
xi) The Result data gets displayed via the Cluster Plot Chart.



13.1.1.2. K-Means ++

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

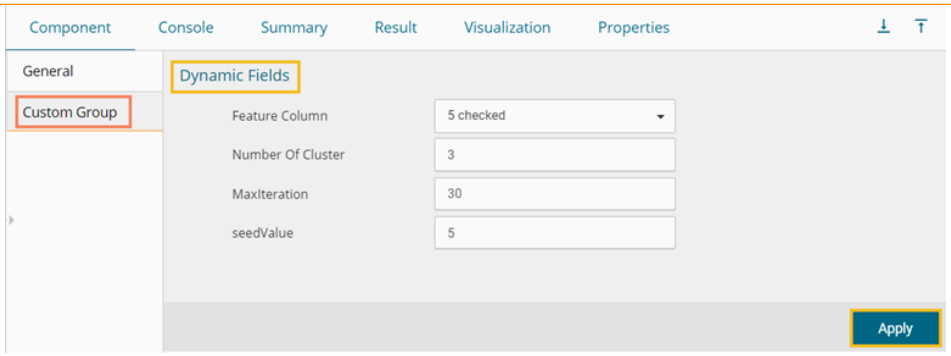
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

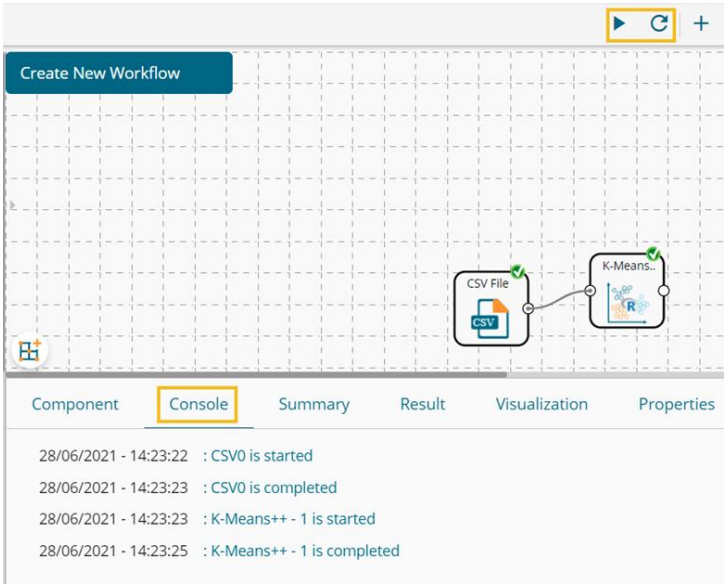
a. Feature Column: Select the feature column from the input data set.
b. Number of Clusters: Set the number of clusters to be created (The default value for this field is 5).
c. Max-Iteration: Enter the number of iterations allowed for discovering clusters. (The default value for this field is 20).
d. Seed Value: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. Set the initial seed value (the default seed value for this field is 10).

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

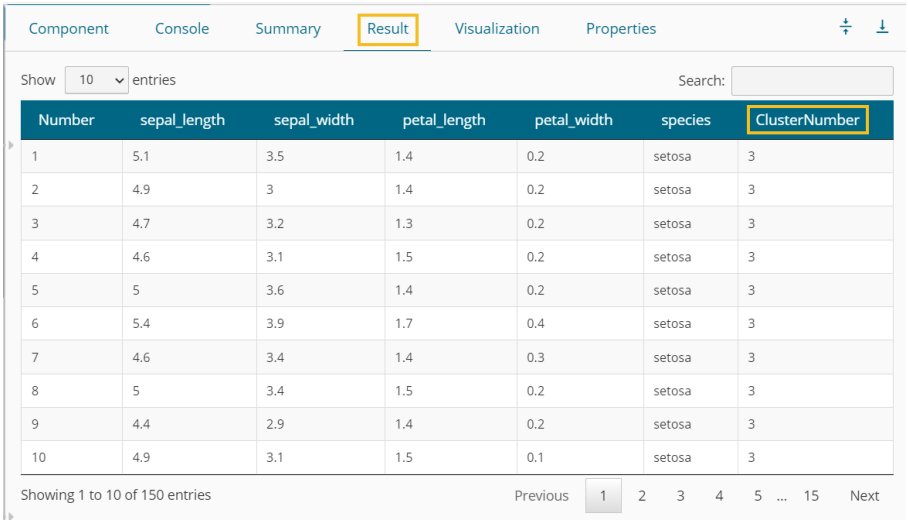
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



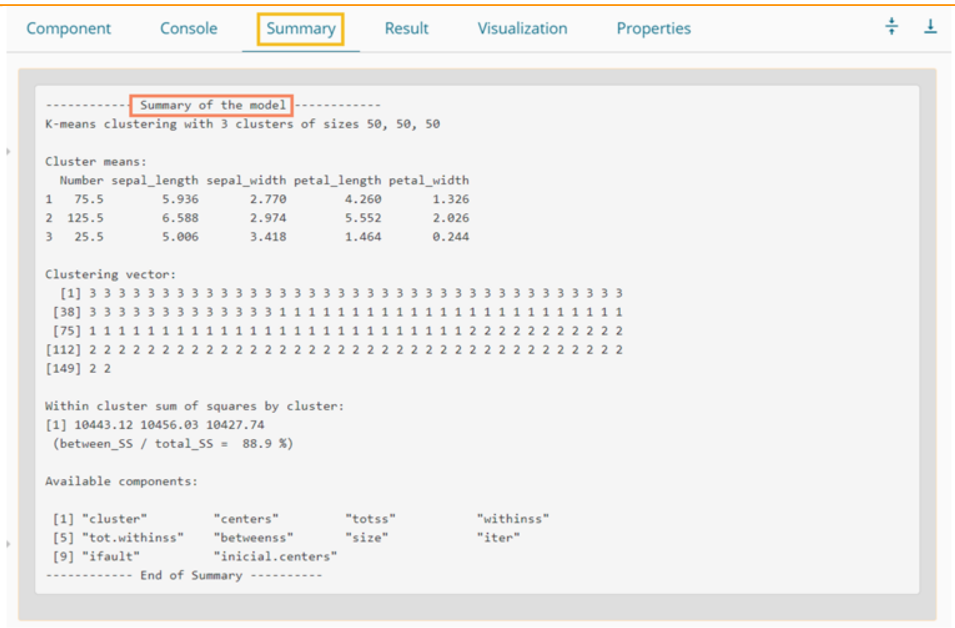
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

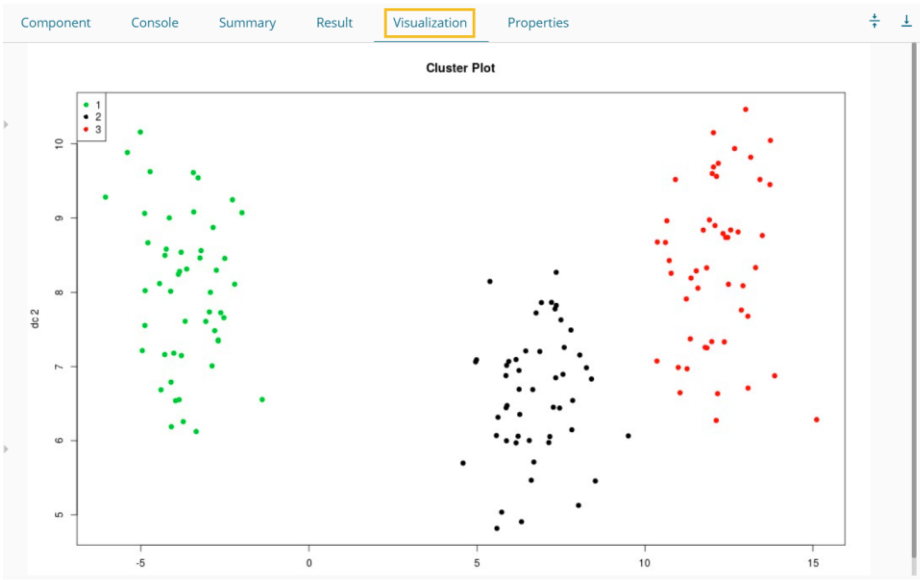


viii) Click the '**Summary**' tab to open the model summary.




```
----------- Summary of the model ------------
K-means clustering with 3 clusters of sizes 50, 50, 50

Cluster means:
  Number sepal_length sepal_width petal_length petal_width
1   75.5        5.936       2.770        4.260       1.326
2  125.5        6.588       2.974        5.552       2.026
3   25.5        5.006       3.418        1.464       0.244

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
[112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[149] 2 2

Within cluster sum of squares by cluster:
[1] 10443.12 10456.03 10427.74
 (between_SS / total_SS =  88.9 %)

Available components:

 [1] "cluster"      "centers"      "totss"        "withinss"
 [5] "tot.withinss" "betweenss"    "size"         "iter"
 [9] "ifault"       "inicial.centers"
------------ End of Summary ----------
```

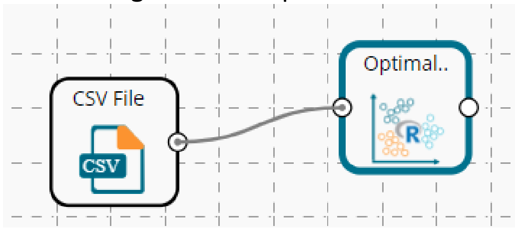
ix) Click the '**Visualization**' tab to see a scatter plot chart representing the processed data.



13.1.1.3. Optimal K Value

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

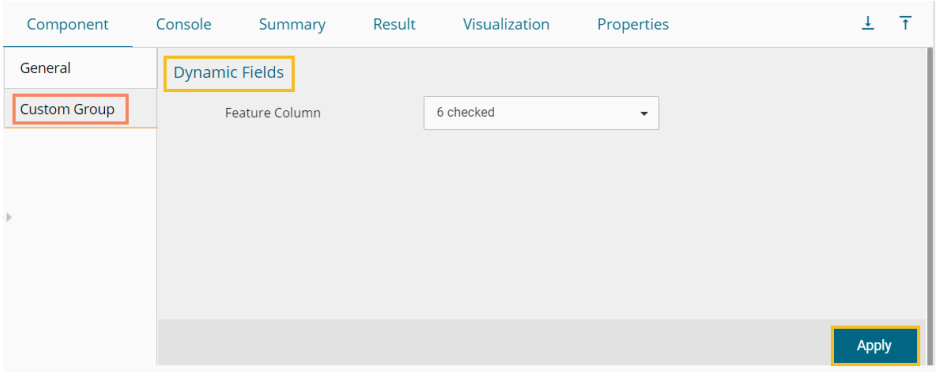
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

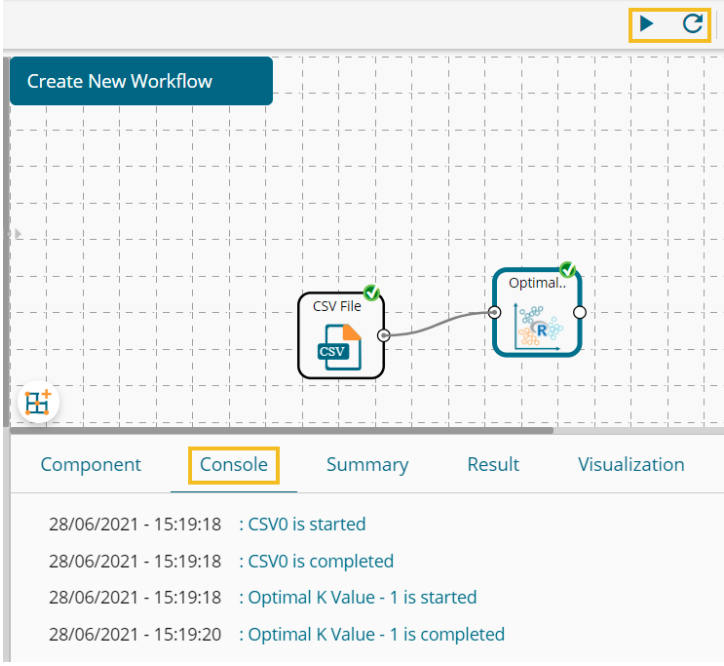
a. Feature Column: Select the feature column from the input data set

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

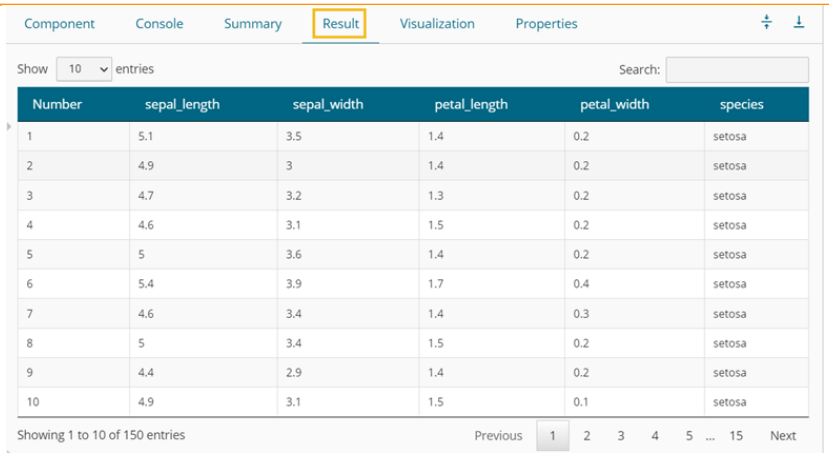
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



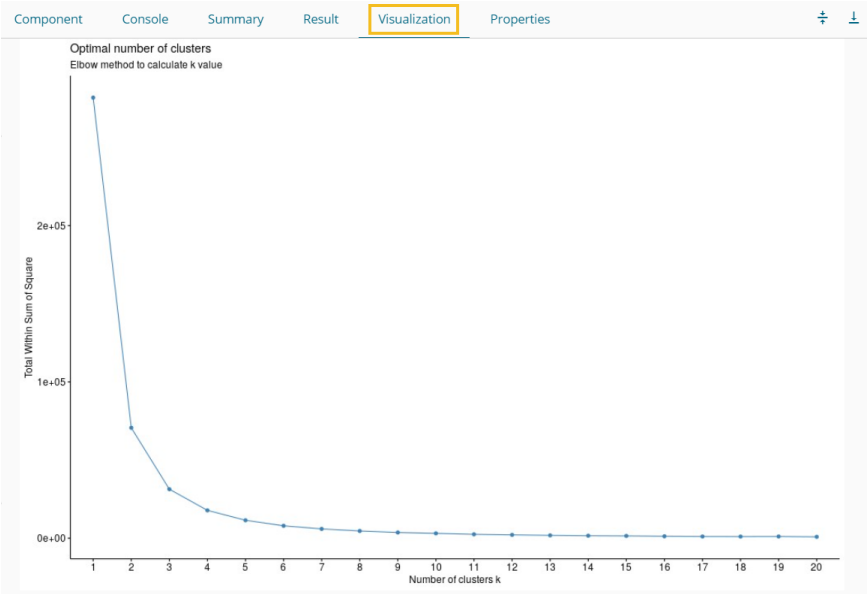
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



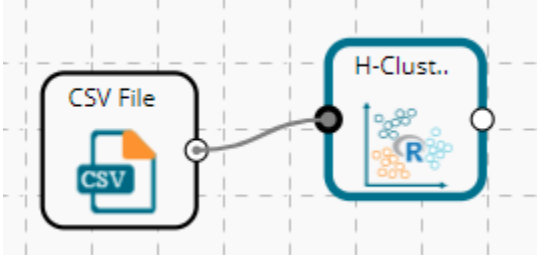
viii) Click the '**Summary**' tab to open the model summary.



13.1.1.4. H-Clustering

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

ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

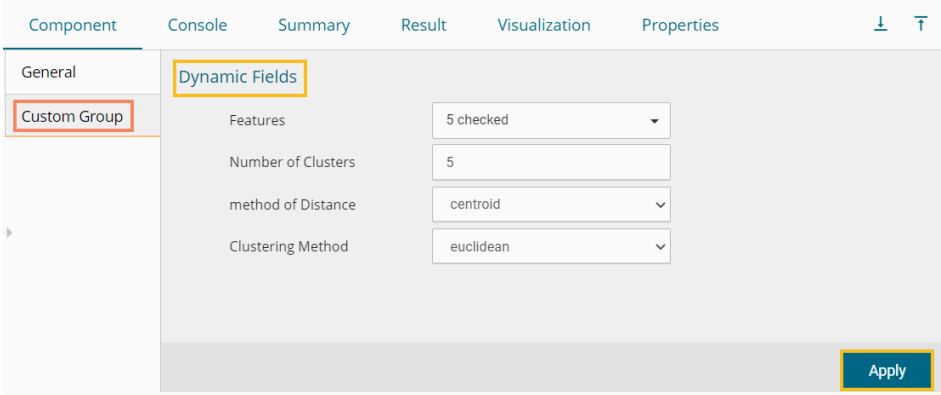
a. Features: Select the feature columns from the input data set

b. Number of Clusters: Set the number of clusters to be created (The default value for this field is 5).

c. Method of Distance: Select a method of distance out of the given choices that include centroid, median, mcquitty, average, single, complete, ward.D2, ward.D

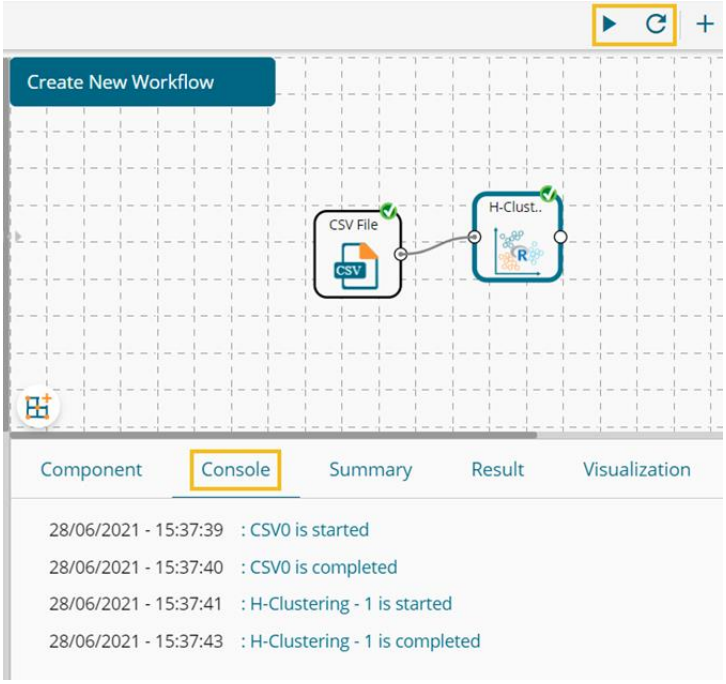
d. Clustering Method: Select a clustering method out of the given choices that include euclidean, manhattan, maximum, binary, canberra, minkowski

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

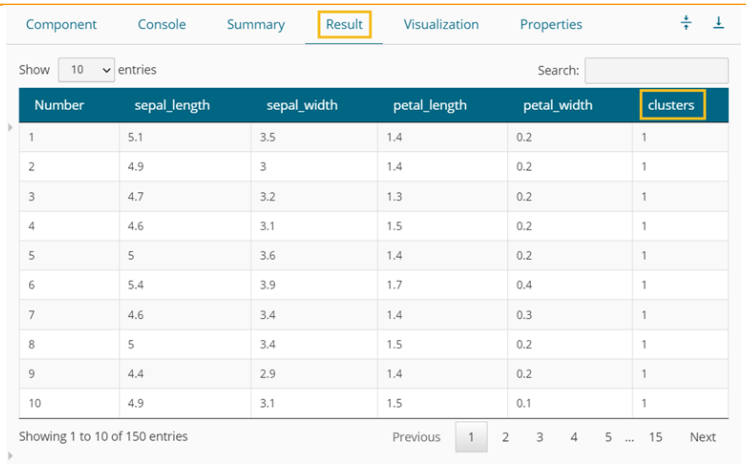
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



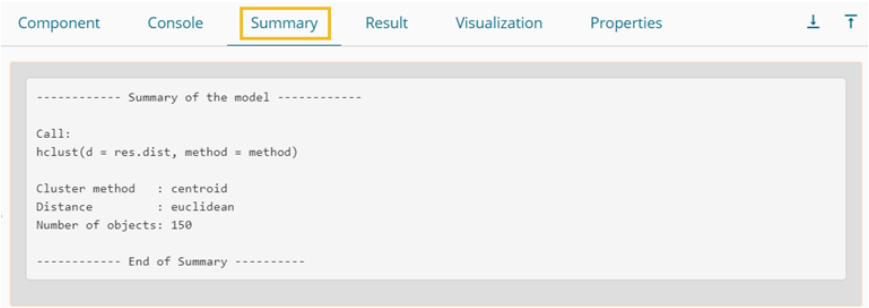
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

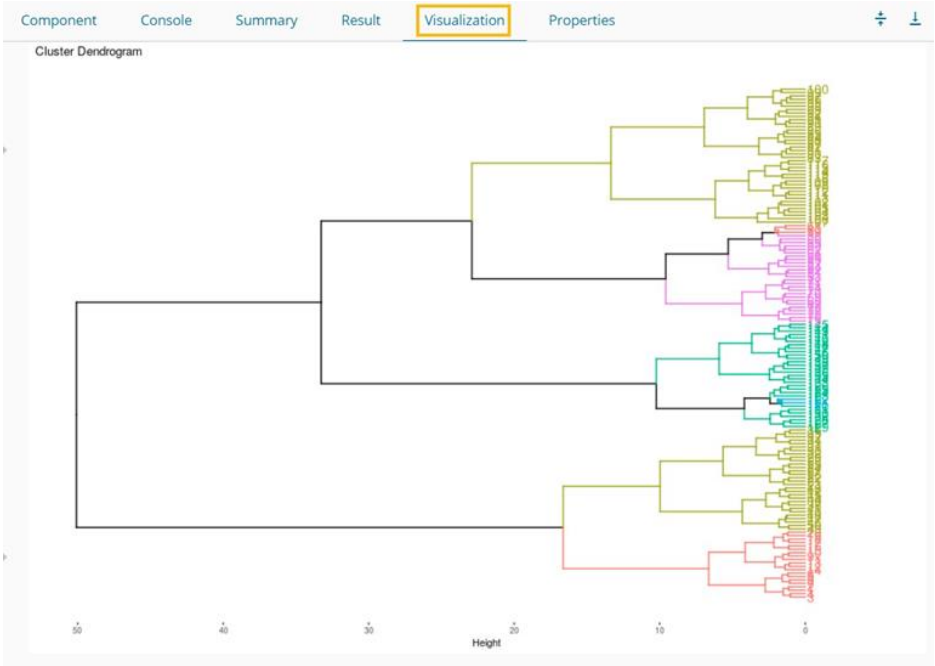
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



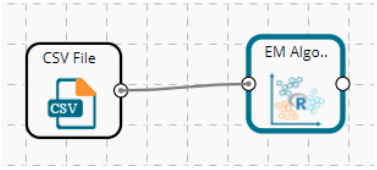
ix) Click the '**Visualization**' tab to see the processed data represented through a Cluster Dendrogram chart.



13.1.1.5. EM Algorithm

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

ii) Click the algorithm component.



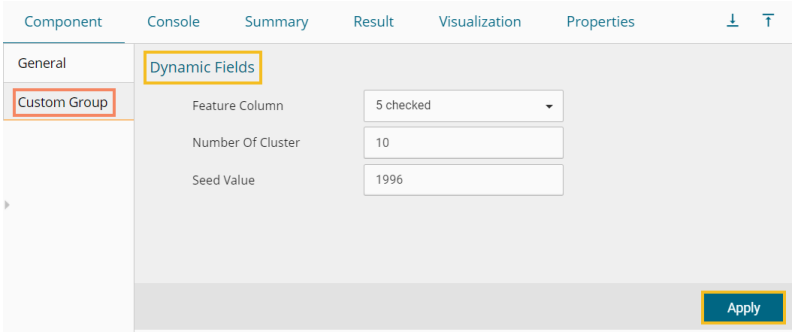
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. Feature Column: Select the feature columns from the input data set

b. Number of Cluster: Set the number of clusters to be created (The default value for this field is 5).

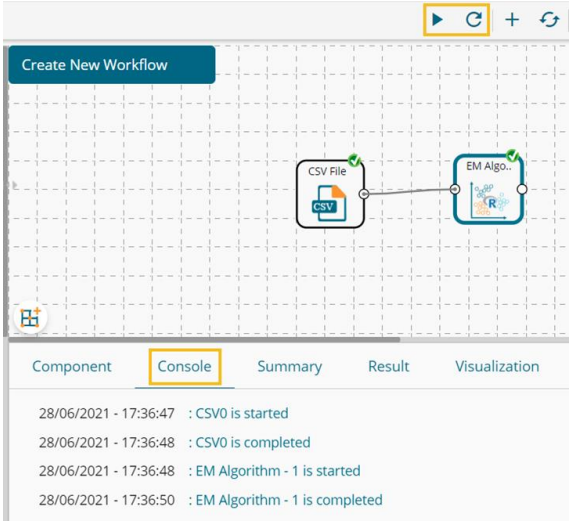
c. Seed Value: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. Set the initial seed value (the default seed value for this field is 10).

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

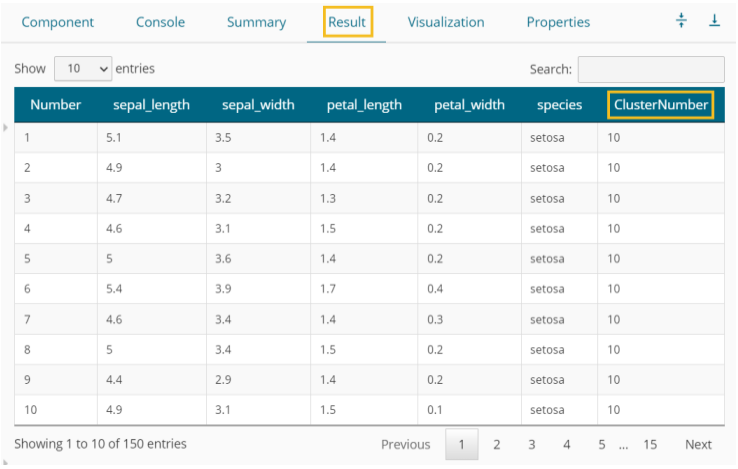
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:

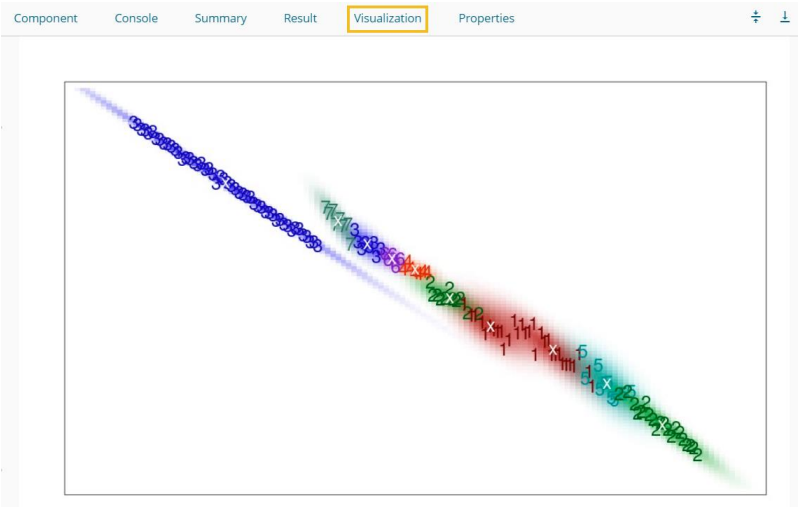
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.

ix) Click the '**Visualization**' tab to see the processed data through a chart.



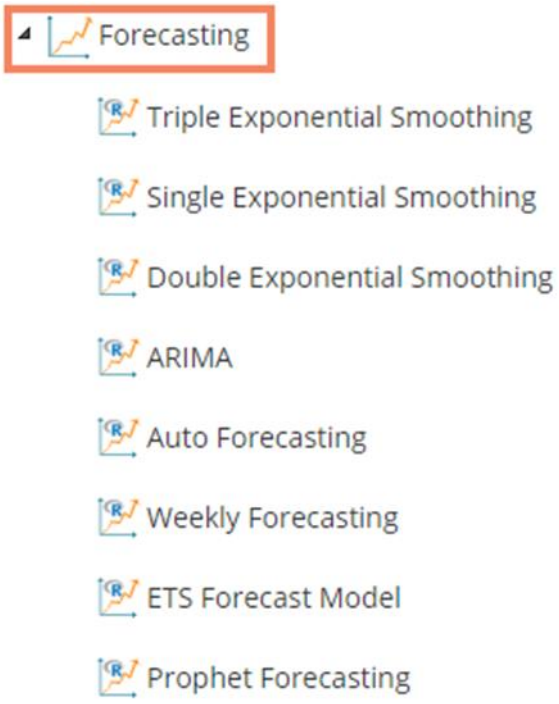
13.1.2. Forecasting

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

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

1. Forecasting

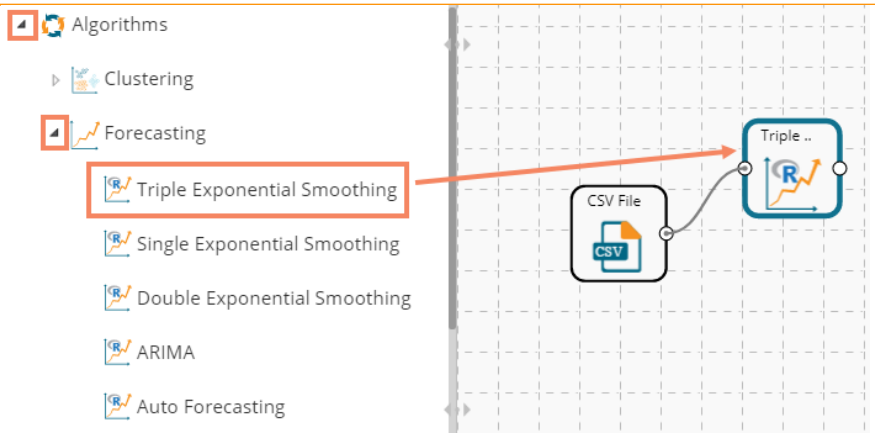
2. Trend
   The document describes all the available Forecasting algorithms considering both the output modes as possibilities.



13.1.2.1. Triple Exponential Smoothing

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

i) Drag the Triple Exponential Smoothing component to the workspace and connect it 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. The user gets two options for this field.

1. **Trend**: Selecting this option displays source data along with predicted values for the given data set.
2. **Forecast**: Selecting this option displays forecasted values for the given period. The forecasted values get appended to the target column when the '**Forecast**' output mode has been selected.

ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected **'Output Mode'** option is **'Forecast.'**

b. **Column Selection**

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

c. **Input Data Handling**

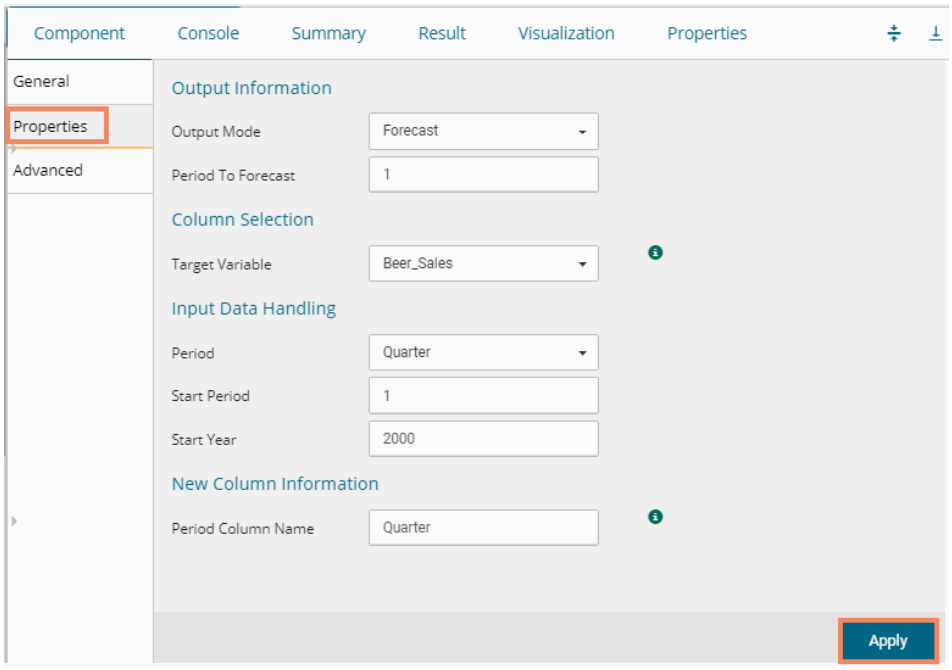
i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.

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

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

d. **New Column Information**

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



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

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

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

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

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

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

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

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

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

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

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

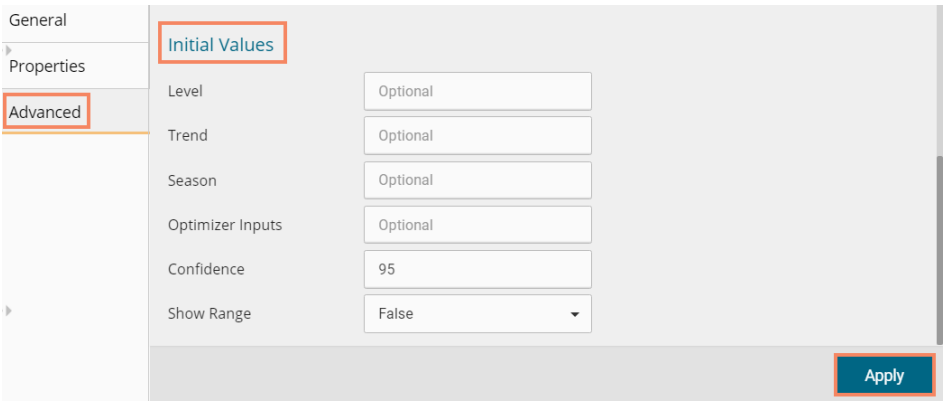
iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.

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

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

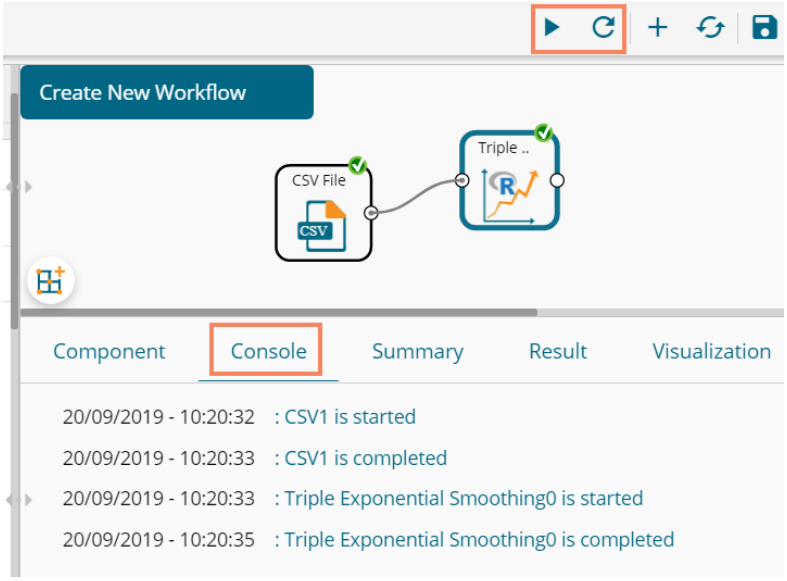
1. True: By selecting this option, **Lower Range** and **Upper Range** get displayed in the Result and Visualization of the dataset.
2. False: By selecting this option, Ranges do not get displayed in the dataset

iv) Click the **'Apply'** option.



v) Run the workflow after getting the success message.

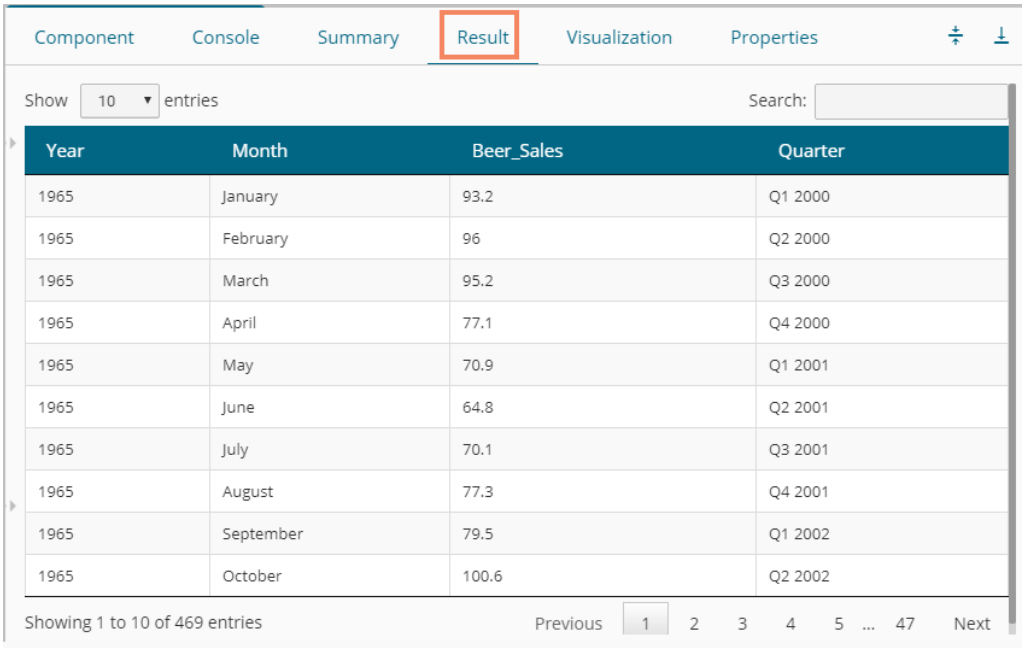
vi) The user gets directed to the 'Console' tab displaying the ongoing process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged component.



vii) Follow the below-given steps to display the Result view:

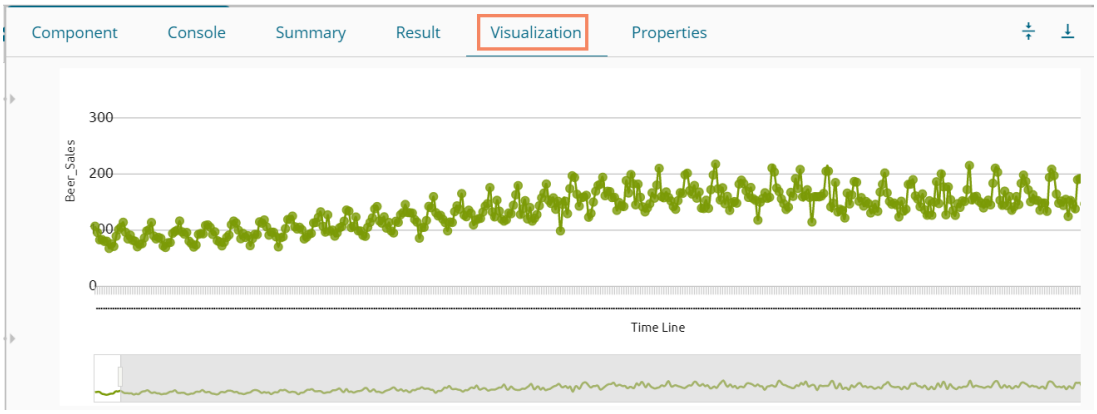
a. Click the dragged algorithm component onto the workspace.

b. Click the 'Result' tab (In this case, the selected output mode is '**Forecasting**')

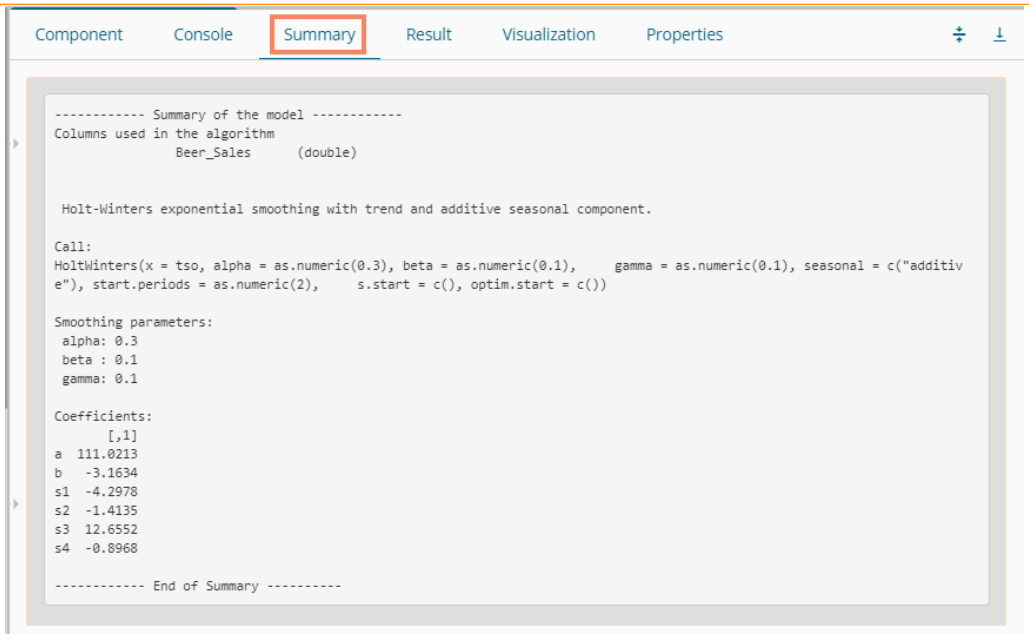


viii) Click the 'Visualization' tab.

ix) The Result data will be displayed via the TimeLine Chart.



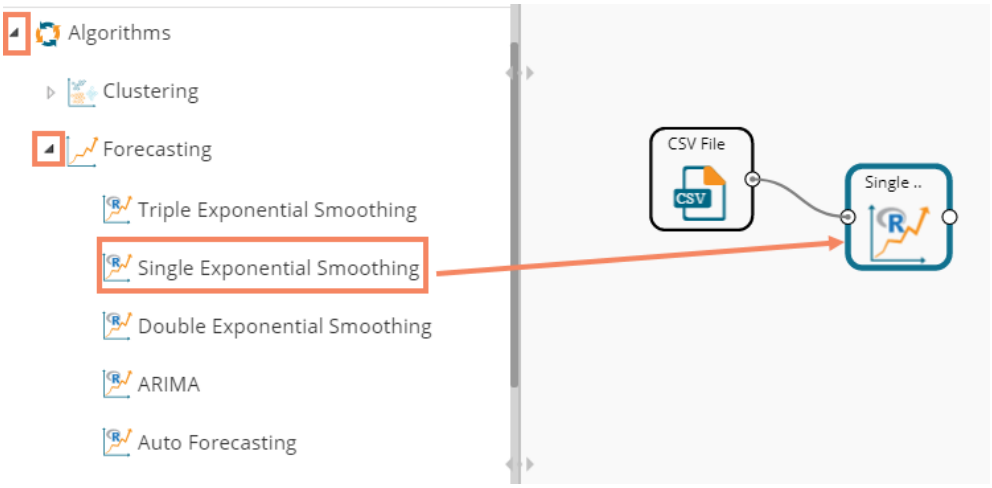
x) Click the '**Summary**' tab to view the model summary.



13.1.2.2. 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 it 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 displays source data along with predicted values for the given data set. A new column '**Predicted Values**' gets added in the Result view when the '**Trend**' output mode has been selected.
2. **Forecast**: Selecting this option displays forecasted values for the given period. The forecasted values get appended to the target column when the '**Forecast**' output mode has been selected.

ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected **'Output Mode'** option is **'Forecast.'**

b. **Column Selection**

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

c. **Input Data Handling**

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

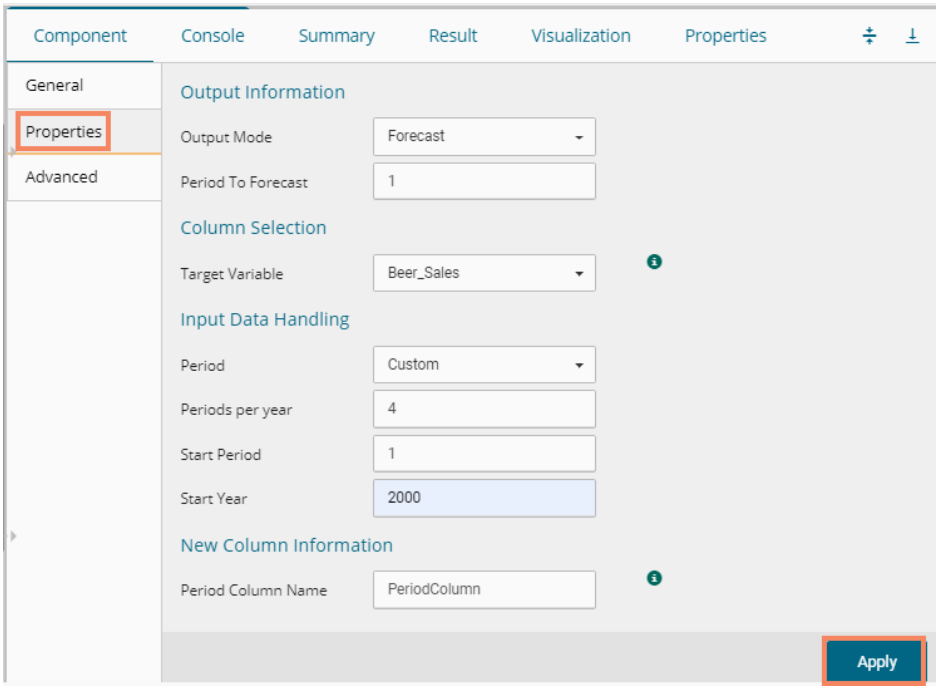
ii. **Period Per Year**: This field appears only when the selected '**Period**' option is **'Custom.'**

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

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

d. **New Column Information**

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



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

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

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

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

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

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

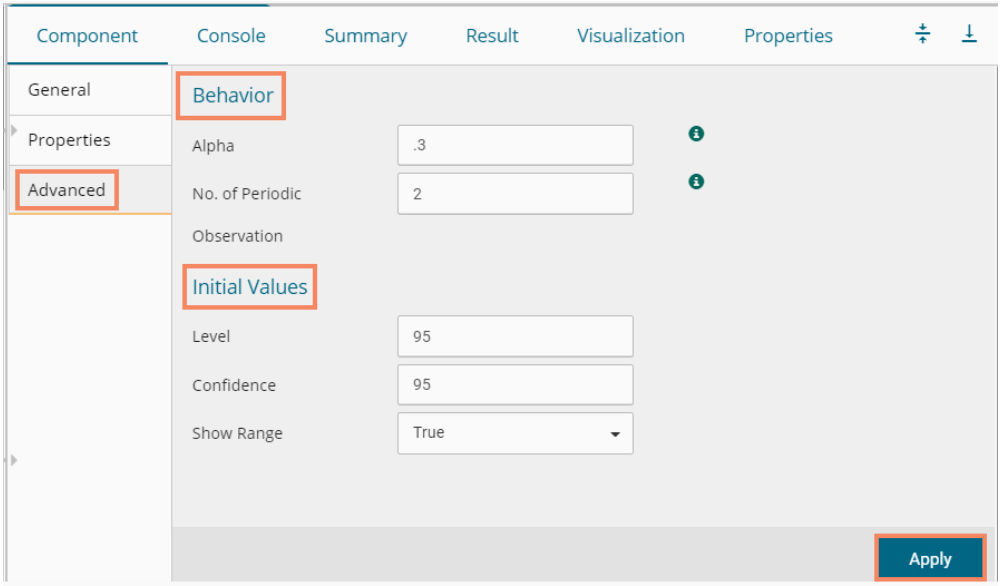
i. **Level**: Enter the initial value for the level. It is an optional field.

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

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

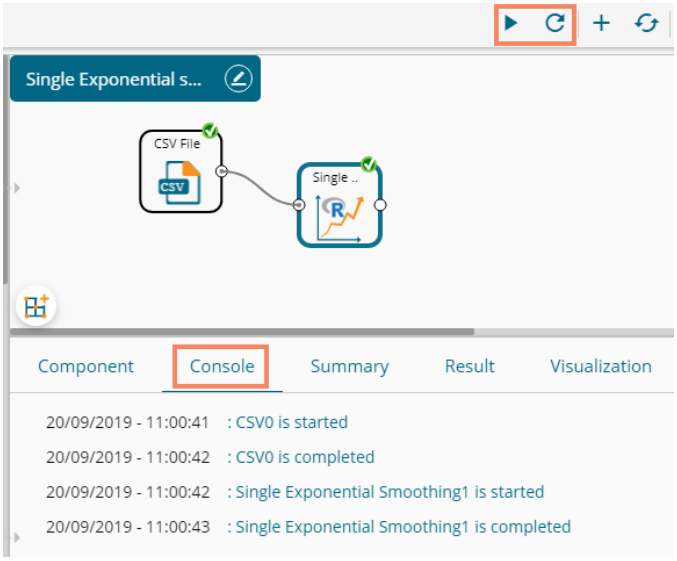
1. True: By selecting this option, **Lower Range** and **Upper Range** get displayed in the Result and Visualization of the dataset.
2. False: By selecting this option, Ranges do not get shown in the dataset.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

vi) The 'Console' tab opens, displaying the ongoing process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged components.

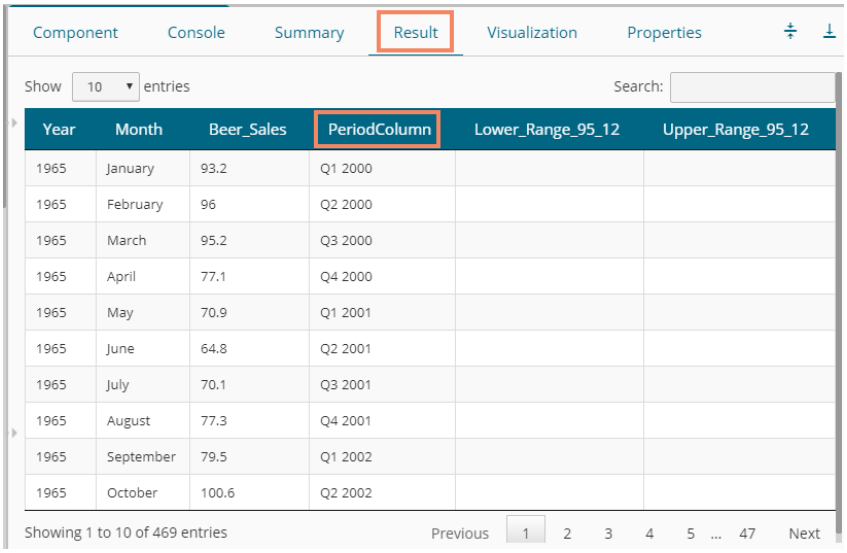


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace

b. Click the 'Result' tab.

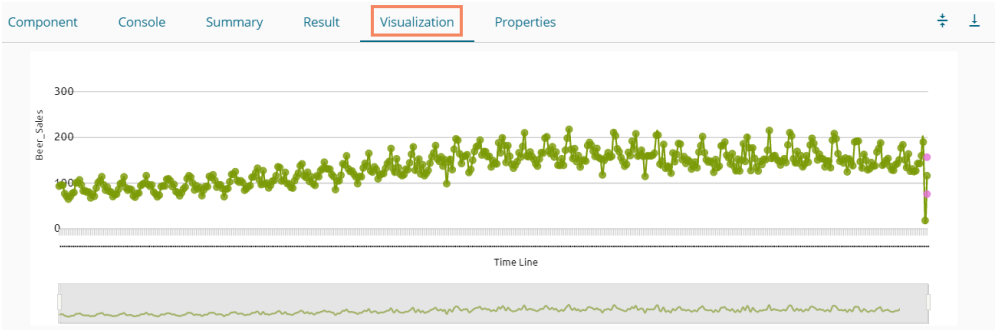
viii) Predicted values get appended to the target variable column when the selected output mode is Forecasting. The Lower Range and Upper Range columns display when the '**Show Range**' field is marked '**True**' from the **Advanced** tab.



Year	Month	Beer_Sales	PeriodColumn	Lower_Range_95_12	Upper_Range_95_12
1965	January	93.2	Q1 2000		
1965	February	96	Q2 2000		
1965	March	95.2	Q3 2000		
1965	April	77.1	Q4 2000		
1965	May	70.9	Q1 2001		
1965	June	64.8	Q2 2001		
1965	July	70.1	Q3 2001		
1965	August	77.3	Q4 2001		
1965	September	79.5	Q1 2002		
1965	October	100.6	Q2 2002		

ix) Click the 'Visualization' tab.

x) The Result data gets displayed via the **TimeLine** Chart.

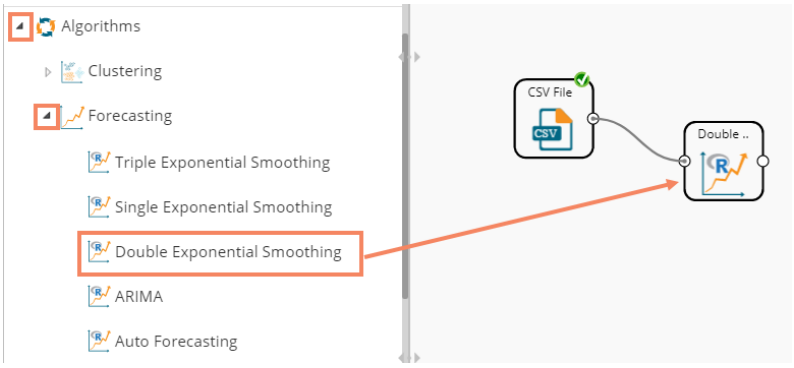


xi) Click the **'Summary'** tab to view the model summary.

13.1.2.3. Double Exponential Smoothing

The 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 it 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 displays source data along with predicted values for the given data set. A new column '**Predicted Values**' gets added in the Result view when the '**Trend**' output mode has been selected.
2. **Forecast**: Selecting this option displays forecasted values for the given period. The forecasted values get appended to the target column when the '**Forecast**' output mode has been selected.

ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected **'Output Mode'** option is **'Forecast.'**

b. **Column Selection**

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

c. **Input Data Handling**

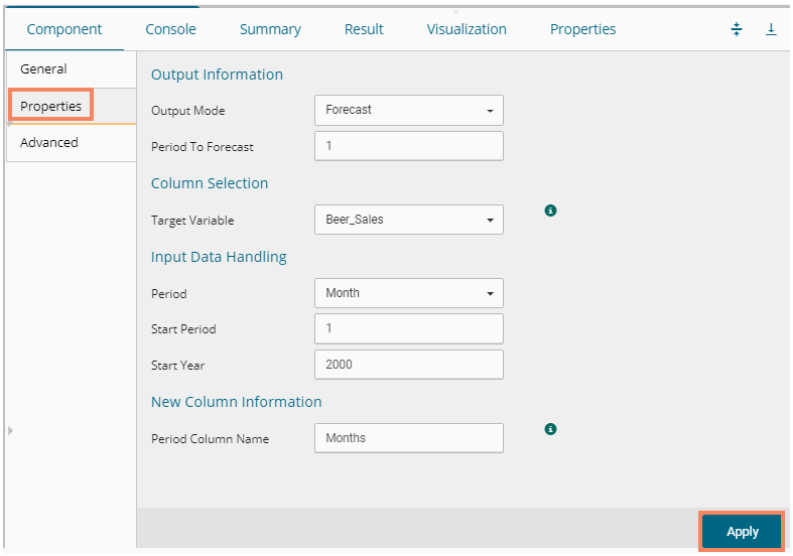
i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.

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

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

d. **New Column Information**

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



Note: The user can click the 'Apply' option from the Properties tab if the configuration of the Advanced tab is not required.

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

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

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

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

iii. **No. of Periodic Observation:** Enter the number of periods observations required to start the calculation (The default value for this field is 2)

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

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

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

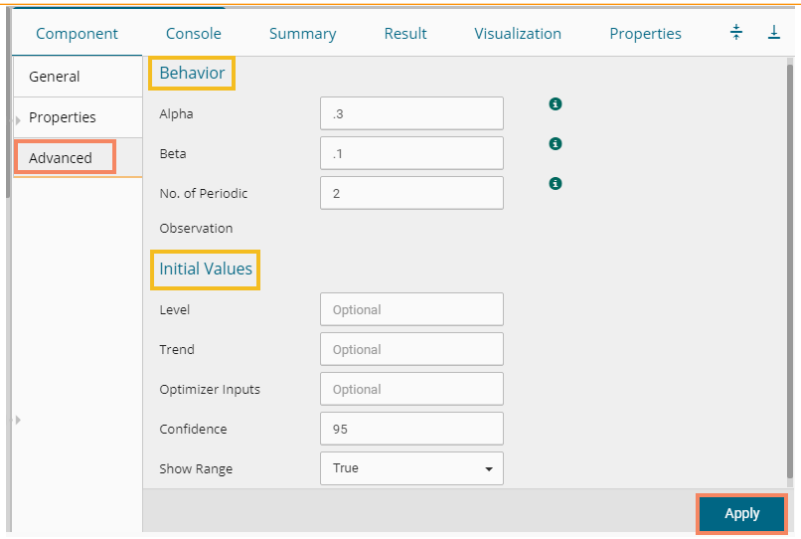
iii. **Optimizer Inputs:** Enter the initial values given for alpha and beta required for the optimizer (it is an optional field)

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

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

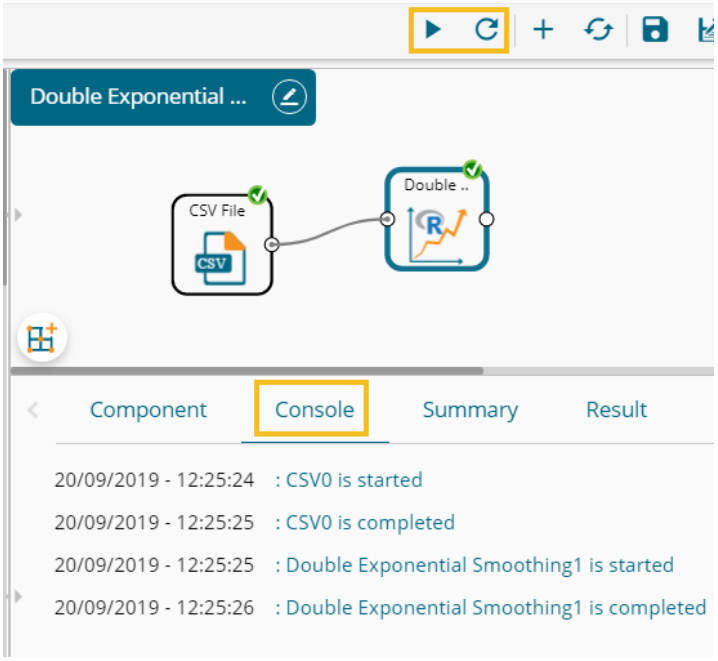
1. True: By selecting this option '**Lower Range**' and '**Upper Range**' get displayed in the Result and Visualization of the dataset
2. False: By selecting this option, Ranges do not get shown in the dataset

iv) Click the **'Apply'** option.



v) Run the workflow after getting the success message.

vi) The 'Console' tab opens, displaying the ongoing process. The completion f the Console process gets marked by the green checkmarks on the top of the dragged components.

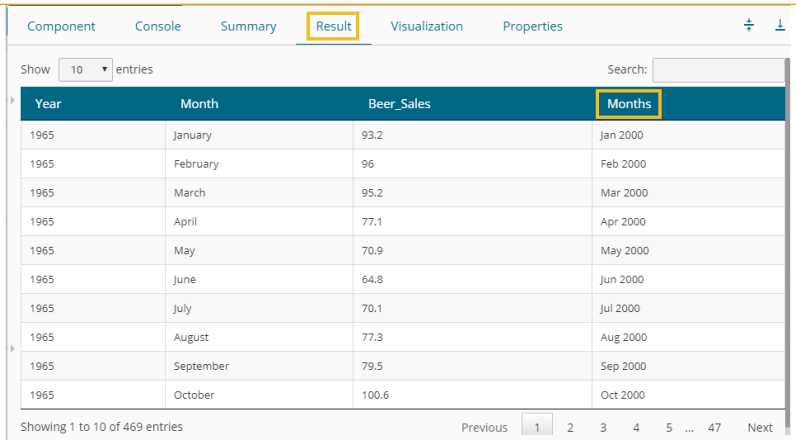


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

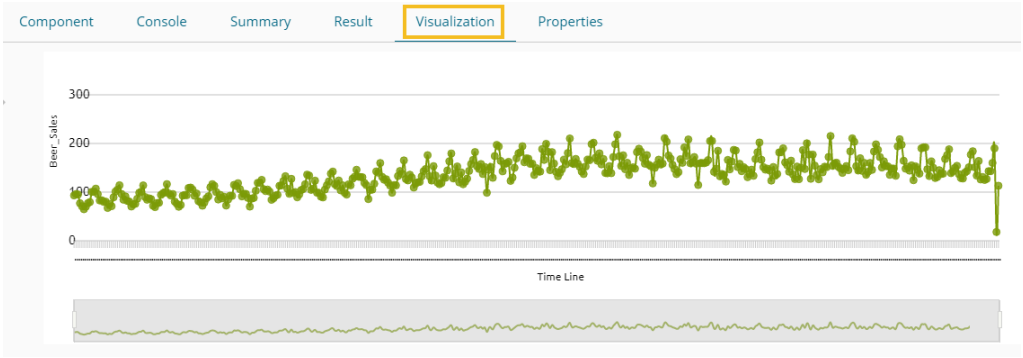
b. Click the 'Result' tab.

The Predicted values get appended to the target column in the Result data if the selected output mode is **Forecasting**.

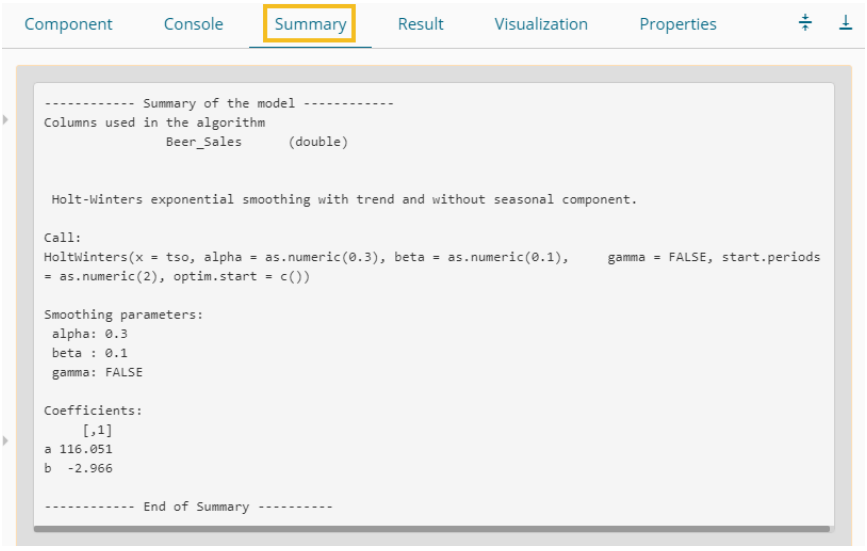


viii) Click the 'Visualization' tab.

ix) The Result data will be displayed via the TimeLine chart.



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

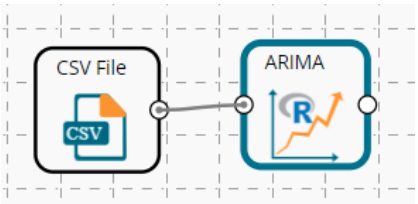


13.1.2.4. ARIMA

ARIMA returns the 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 ARIMA component to the workspace and connect it to a configured data source.

ii) Click on the algorithm component to get the configuration fields.



iii) Configure the '**Properties**' tab.

a. Output Information

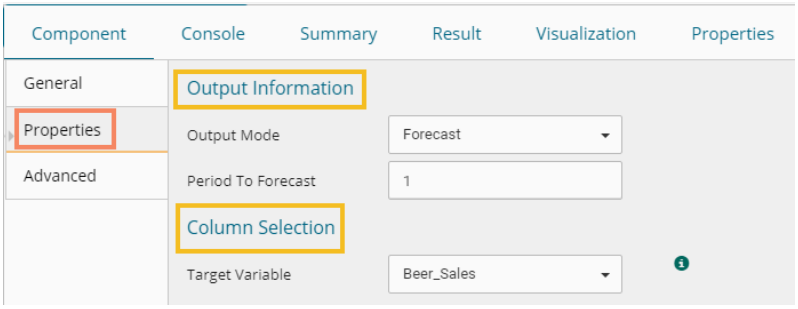
i. **Output Mode**: Select a mode in which you want to display output data

1. **Trend**: Selecting this option displays source data along with predicted values for the given data set. A new column '**Predicted Values**' gets added in the Result view when the '**Trend**' output mode has been selected.
2. **Forecast**: Selecting this option displays forecasted values for the given period. The forecasted values get appended to the target column when the '**Forecast**' output mode has been selected.

ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected **'Output Mode'** option is **'Forecast.'**

b. Column Selection

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



c. Input Data Handling

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

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

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

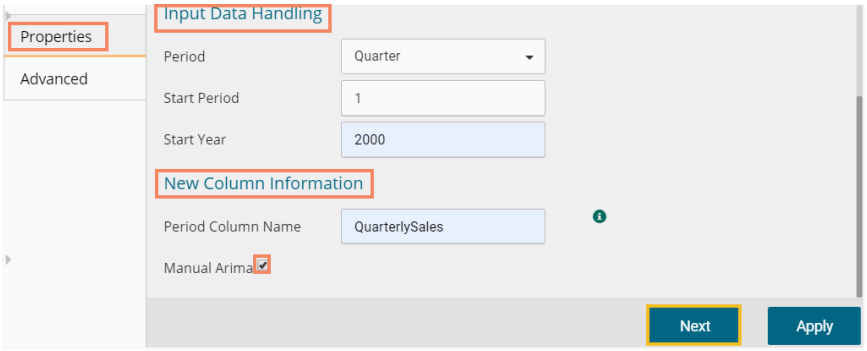
iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter **a** four-digit value for selecting a year (E.g., 2000)

d. New Column Information

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

iv) Enable the Manual Arima option by putting a checkmark in the given box.

v) The '**Next**' option appears on the page.



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

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

i. **Autoregressive order(p):** It is a mandatory field; only integer values are accepted. The default value for this field is 0.

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

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

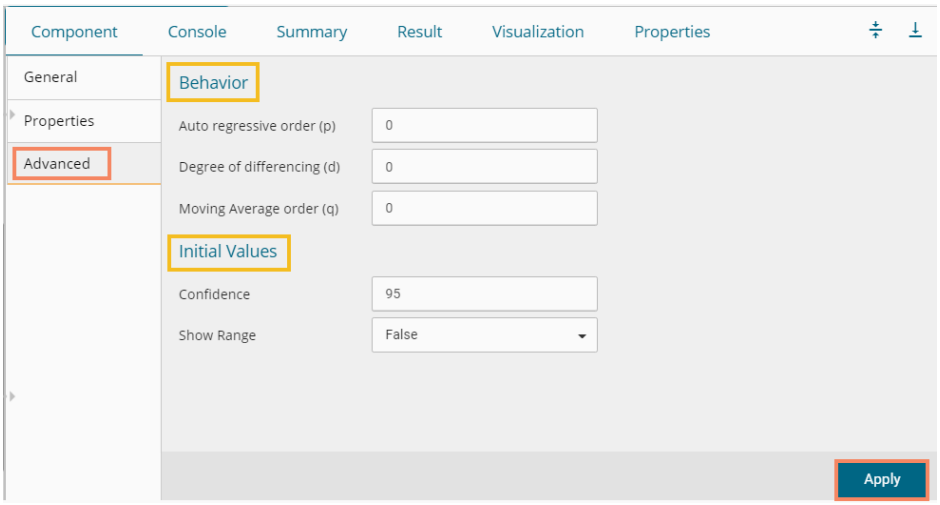
b. Configure the following '**Initial Values**' information:

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

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

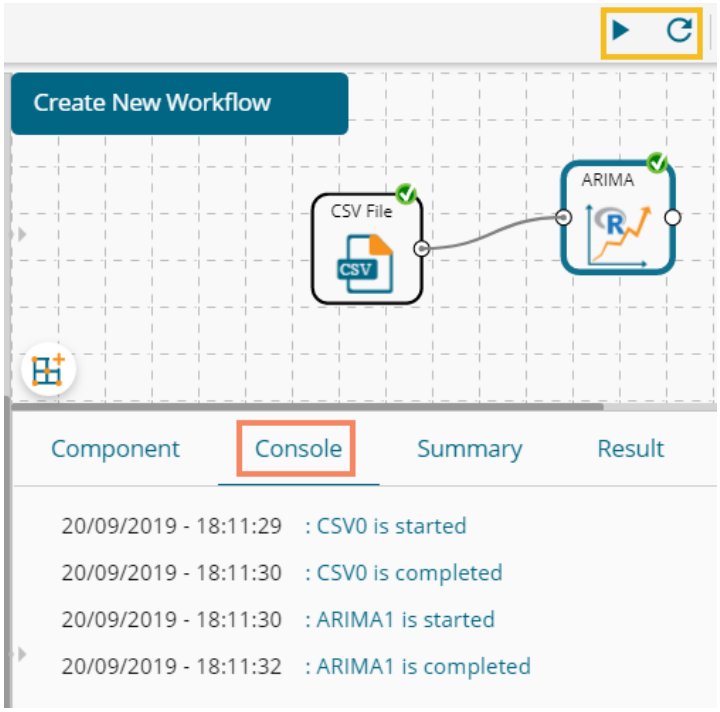
1. **True**: By selecting this option, **Lower Range** and **Upper Range** get displayed in the Result and Visualization of the dataset.
2. **False**: By selecting this option, Ranges do not get shown in the dataset.

vii) Click the **'Apply'** option**.**



viii) Run the workflow after getting the success message.

ix) The '**Console**' tab opens displaying the progress of the process. The completion of the Console process gets marked by the green marks on the top of the dragged components.

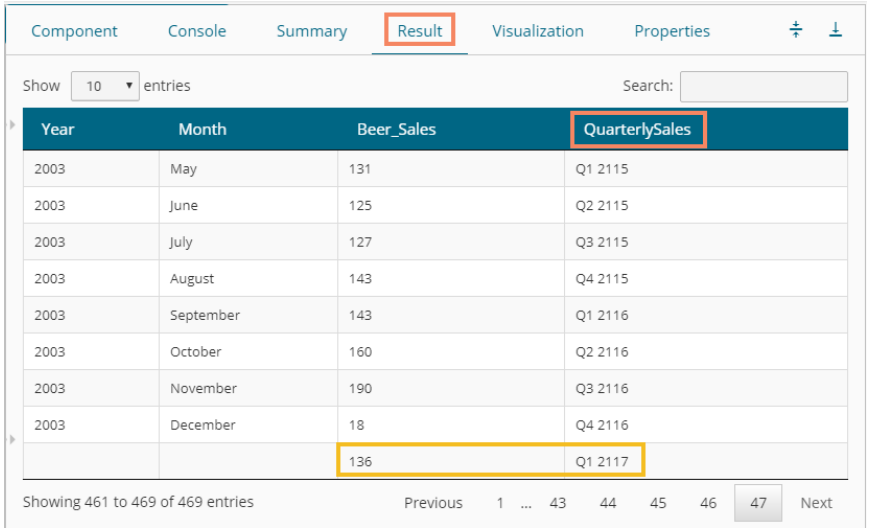


x) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace

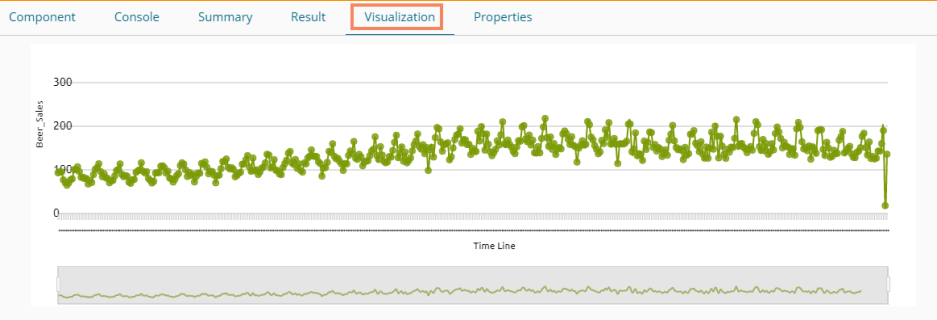
b. Click the '**Result**' tab.

xi) Predicted values get appended to the target column in the Result data (The selected output mode is '**Forecasting**')

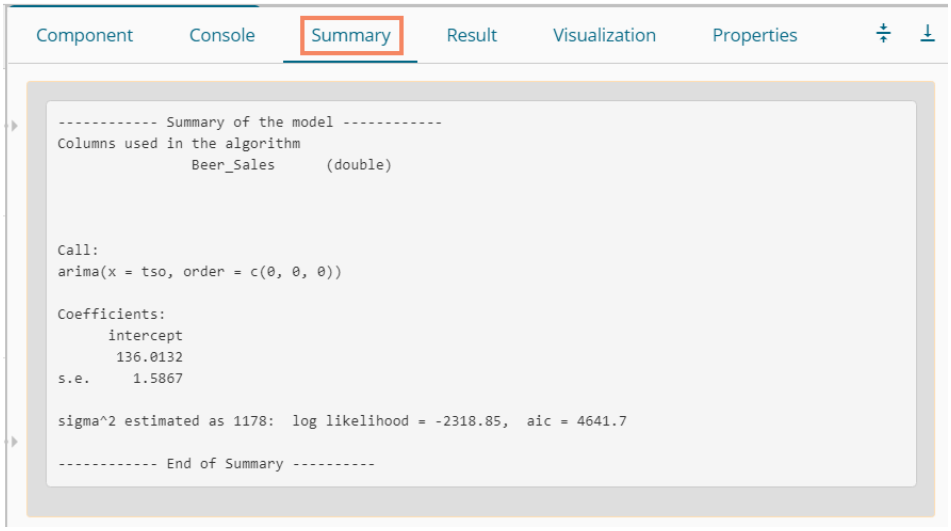


xii) Click the '**Visualization**' tab.

xiii) The Result data will be displayed via the TimeLine chart.

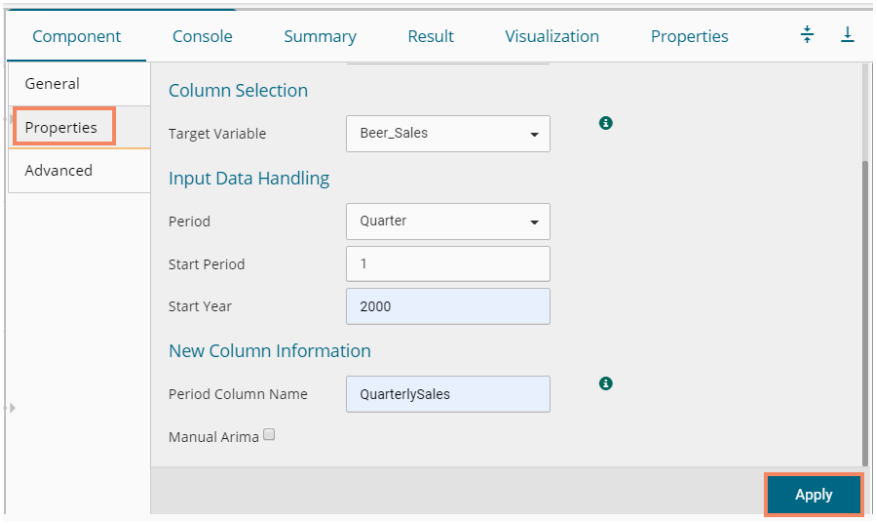


xiv) Click the '**Summary**' tab to view the model summary.

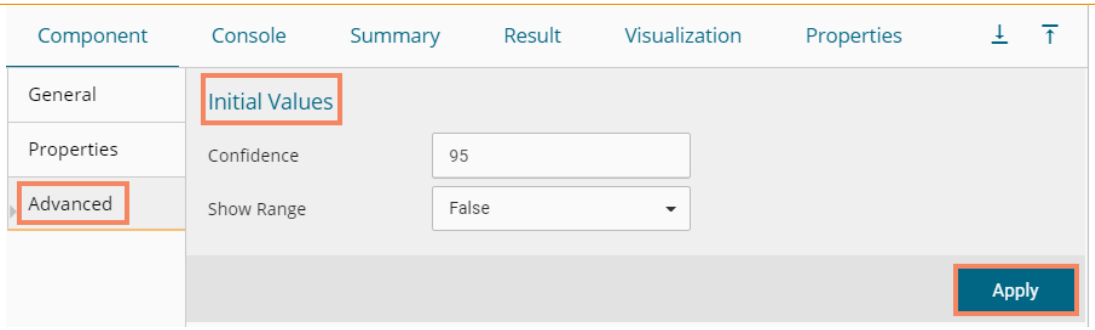


Note: When the '**Manual Arima**' option is not enabled for the R-ARIMA algorithm, the '**Advanced**' tab does not display Behavior fields. The following images display, respectively, the '**Advanced**,' 'Result,' and '**Visualization'** tabs for the same dataset when the manual ARIMA option has been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

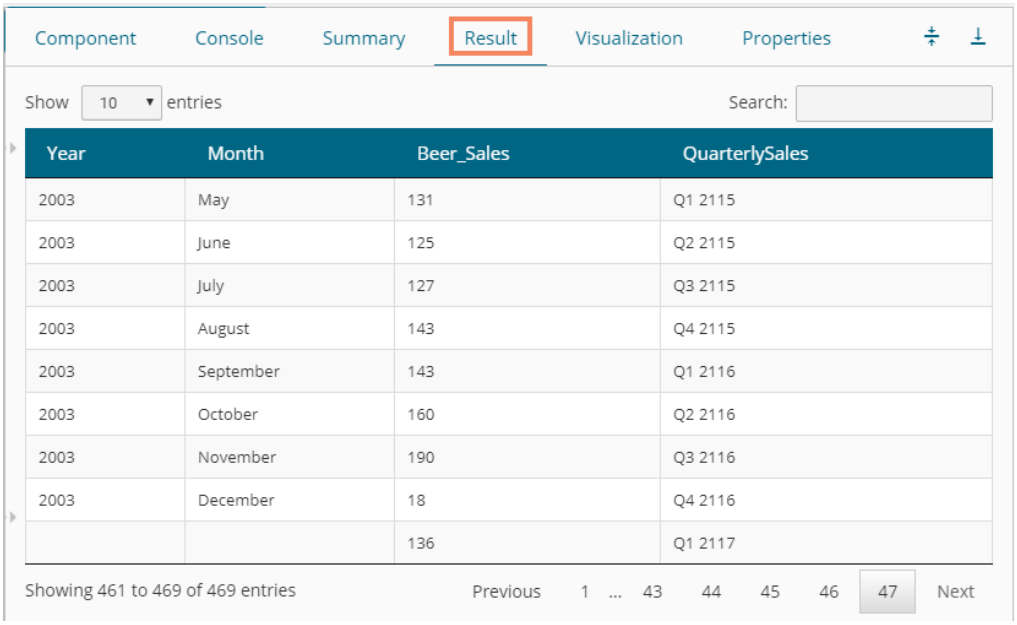
Properties Tab



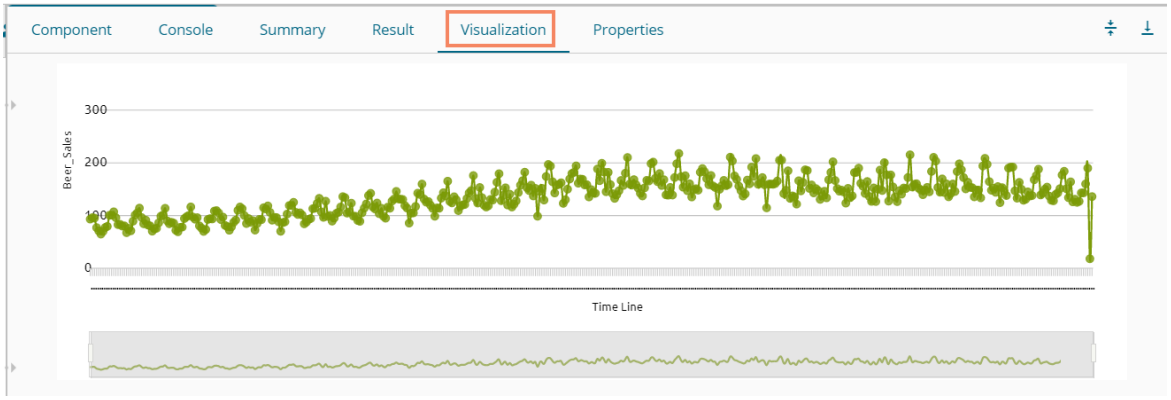
Advanced Tab



Result Tab



Visualization Tab



13.1.2.5. Auto Forecasting

The user can run the algorithm by adjusting smoothing parameters and other initial states variables to find the best AIC value.

i) Drag the R-Auto Forecasting component to the workspace and connect it 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 displays source data along with predicted values for the given data set. A new column **'Predicted** Values' gets added in the Result view when the '**Trend**' output mode has been selected.
   2. **Forecast**: Selecting this option displays forecasted values for the given period. Result values get appended to the target column when '**Forecast**' output mode has been selected.

ii. **Period to Forecast**: Enter a period to forecast. This field appears only when the selected **'Output Mode'** option is **'Forecast.'**

b. Column Selection

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

c. Input Data Handling

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

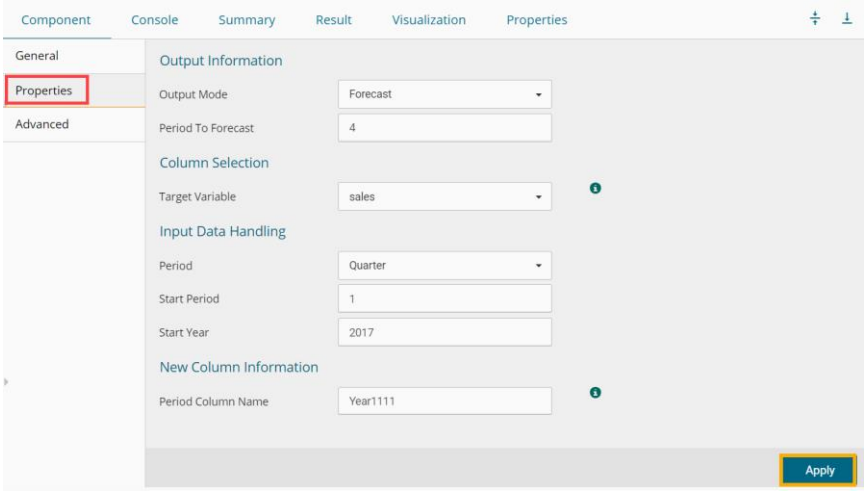
ii. **Period Per Year**: This field appears only when the selected '**Period**' option is **'Custom.'**

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

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

d. New Column Information

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



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

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

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

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

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

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

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

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

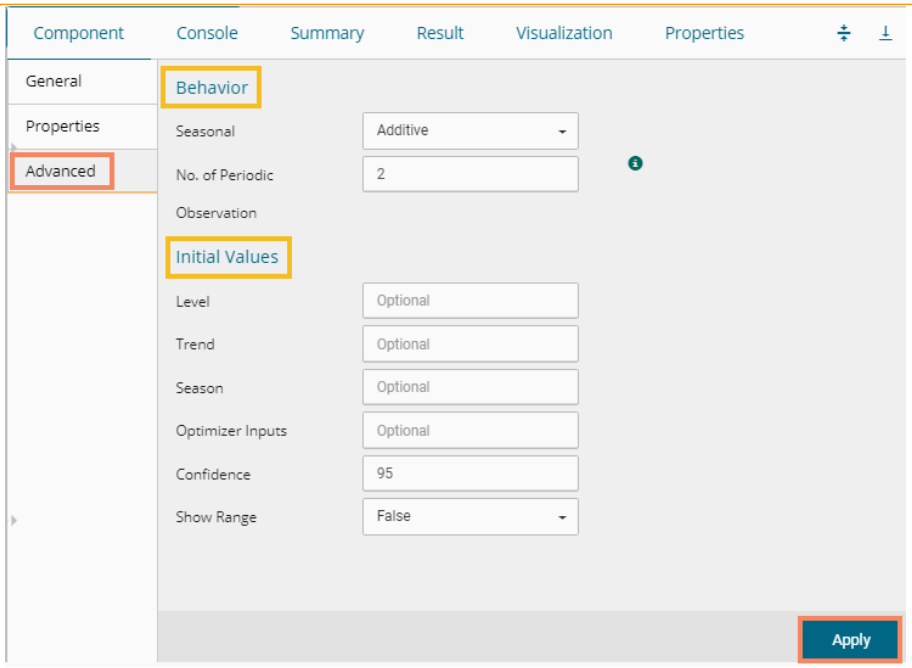
iv. **Optimizer Inputs:** Enter the initial values given for alpha and beta required for the optimizer (It is an optional field).

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 get added to the Result dataset (the default value for this field is 95).

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

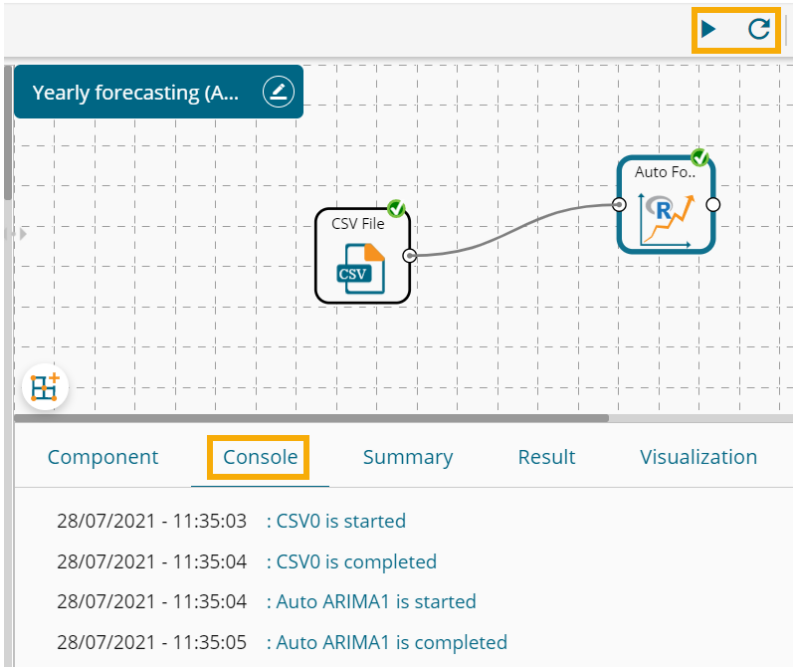
1. **True**: By selecting this option, '**Lower Range**' and '**Upper Range**' get displayed in the Result and Visualization of the dataset.
2. **False**: By selecting this option, Ranges do not get displayed in the dataset.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

vi) The '**Console**' tab opens displaying the progress of the process. The completion of the Console process gets marked with green checkmarks on the top of the dragged components.

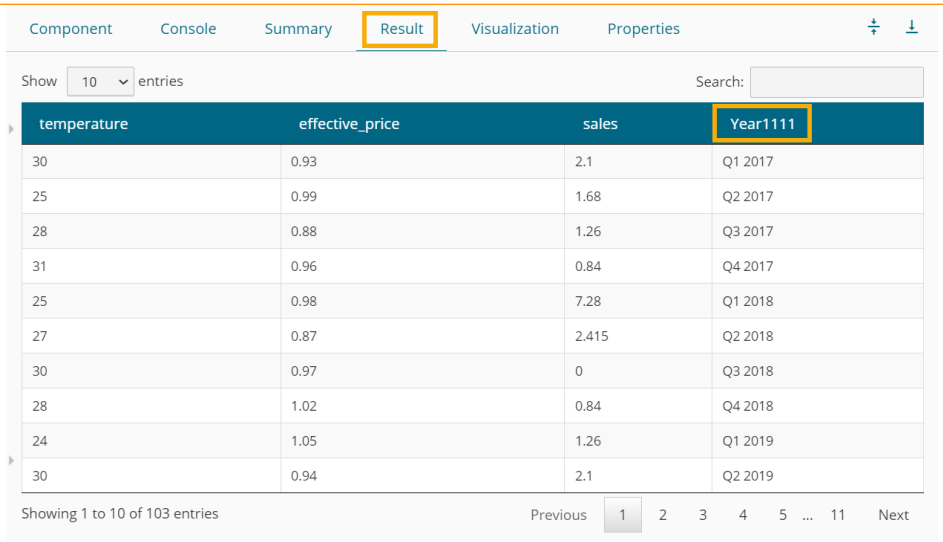


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

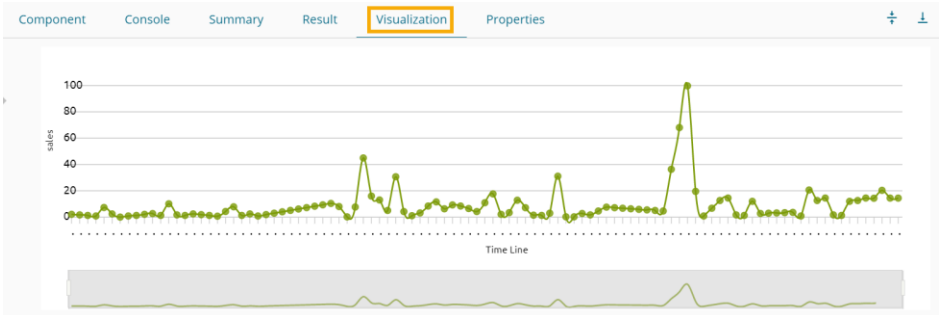
b. Click the '**Result**' tab.

viii) Predicted values get appended to the target column in the Result data (The selected output mode is '**Forecasting**').

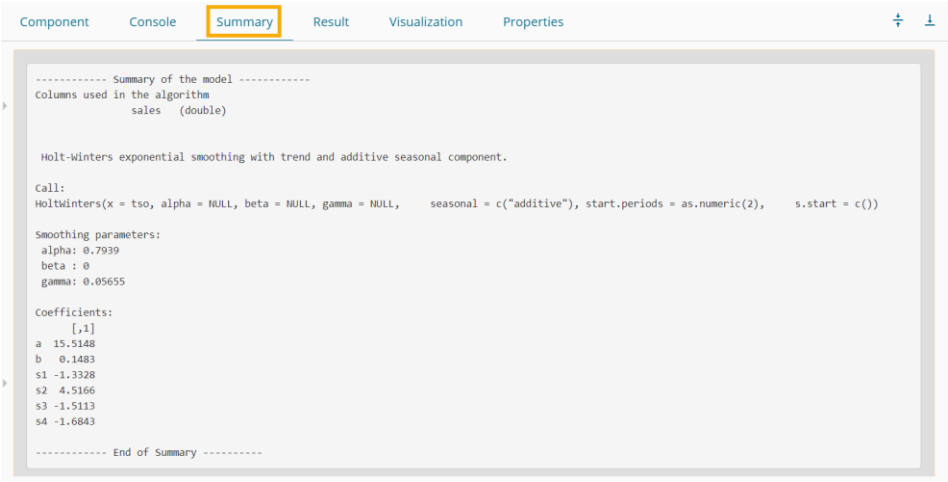


ix) Click the 'Visualization' tab.

x) The processed data will be displayed via the TimeLine chart.



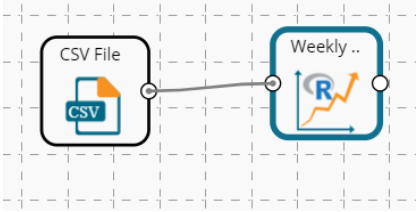
xi) Click the '**Summary**' tab to view the model summary.



13.1.2.6. Weekly Forecasting

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

ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Frequency: Provide a number to indicate the frequency for the data to get processed.

v) Start Year: Enter a year from which you want the data entries to be considered. Enter a four-digit value for selecting a year (E.g., 2017)

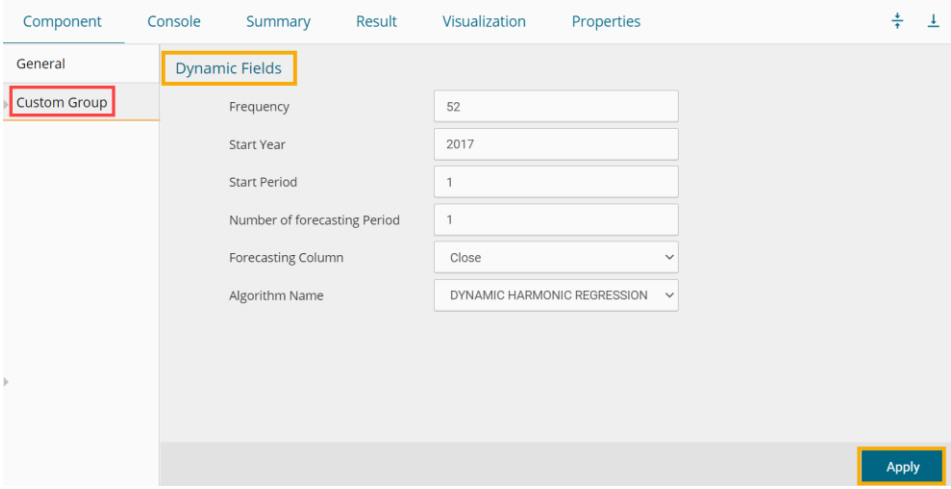
vi) Start Period: Enter a value between 1 and the value specified for the selected option for the **'Period'** field.

vii) Number of forecasting Period: Provide the number of the forecasting period.

viii) Forecasting Column: Select a column from the input dataset.

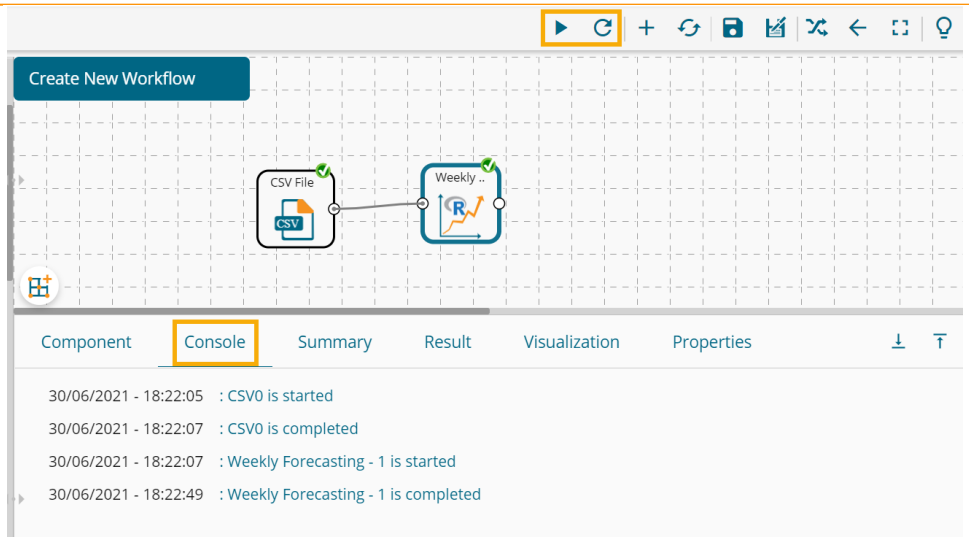
ix) Algorithm Name: Select an option from the given choices that include Dynamic Harmonic Regression, STL, and TBATS.

x) Click the '**Apply**' option.



xi) Run the workflow after getting the success message.

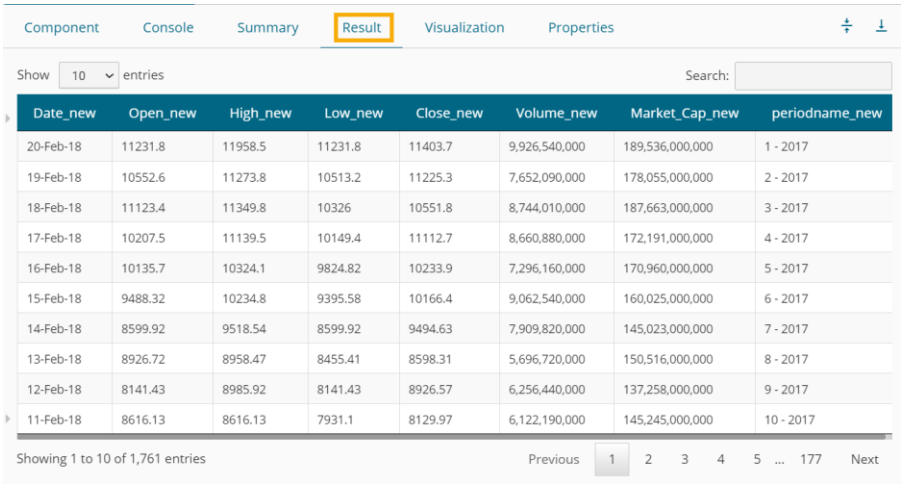
xii) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



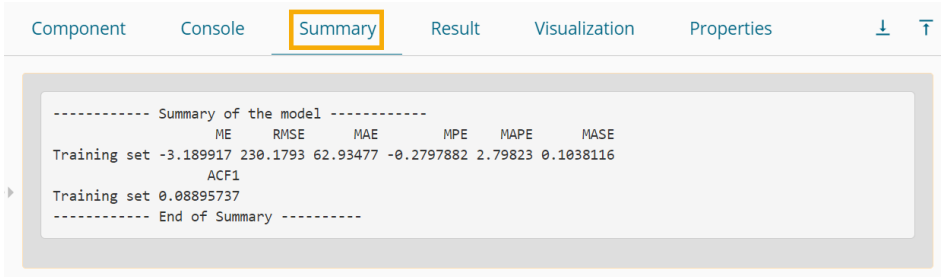
xiii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

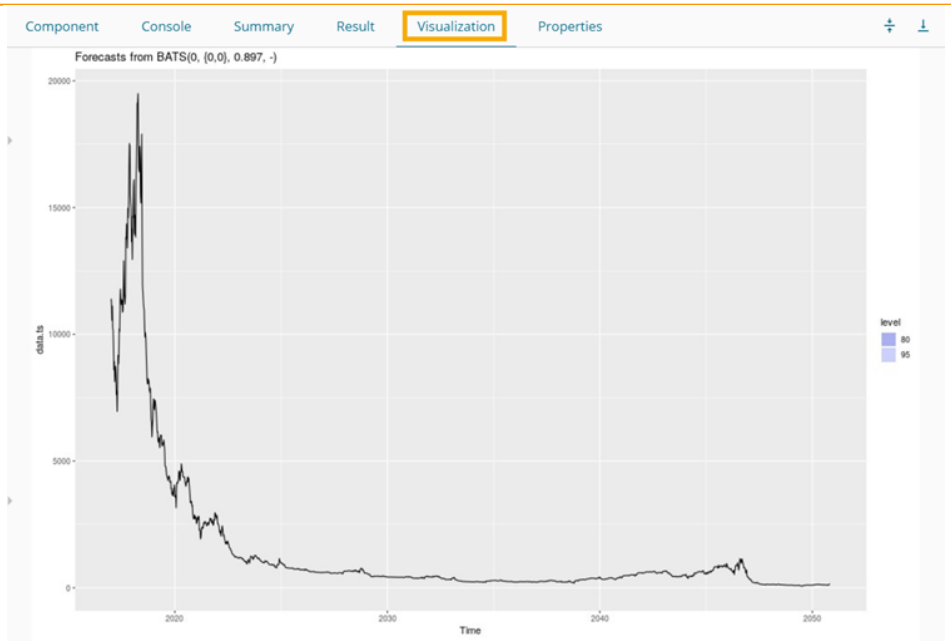
b. Click the '**Result**' tab.



xiv) Click the '**Summary**' tab to open the model summary.



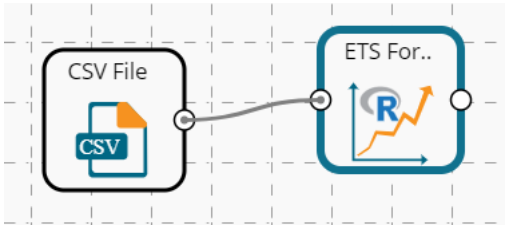
xv) Click the '**Visualization**' tab to see the processed data through a chart.



13.1.2.7. ETS Forecasting Model

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

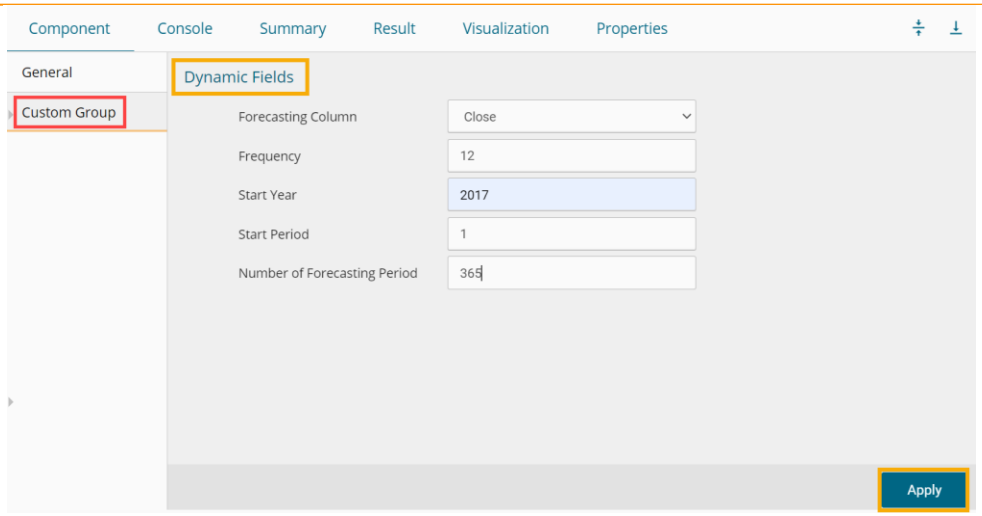
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

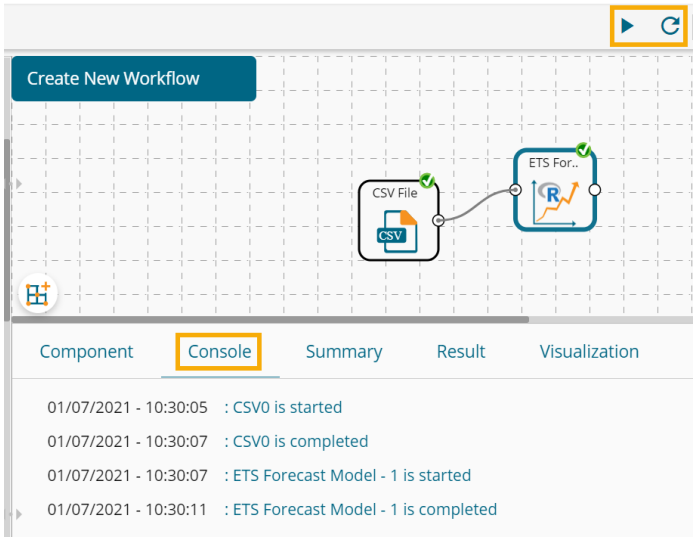
- **a.** Forecasting Column: Select a column from the input dataset.
- **b.** Frequency: Provide a number to indicate the frequency for the data to get process.
- **c.** Start Year: Enter a year from which you want the data entries to be considered. Enter a four-digit value for selecting a year (E.g., 2017)
- **d.** Start Period: Enter a value between 1 and the value specified for the selected option for the **'Period'** field.
- **e.** Number of Forecasting Period: Provide the number of forecasting periods.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

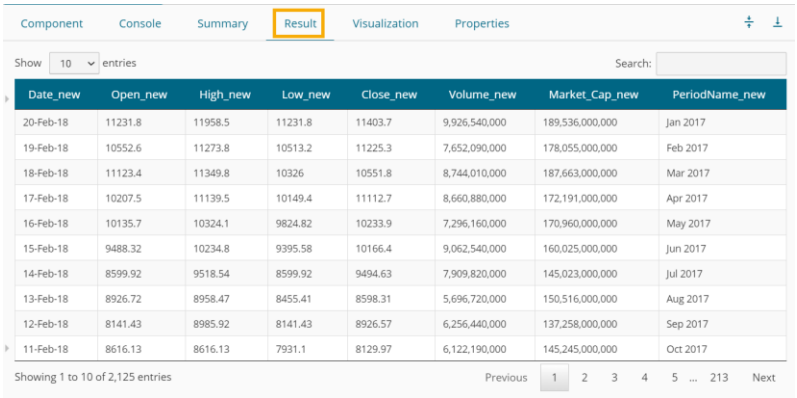
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



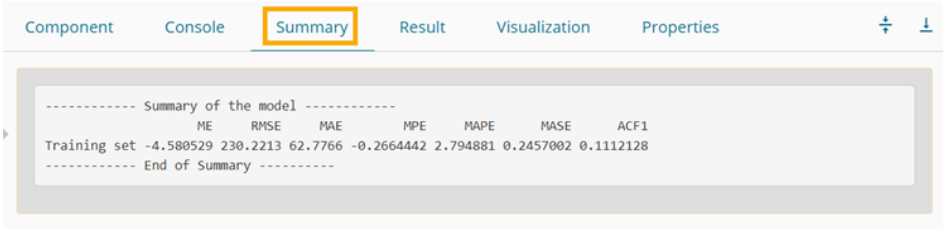
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

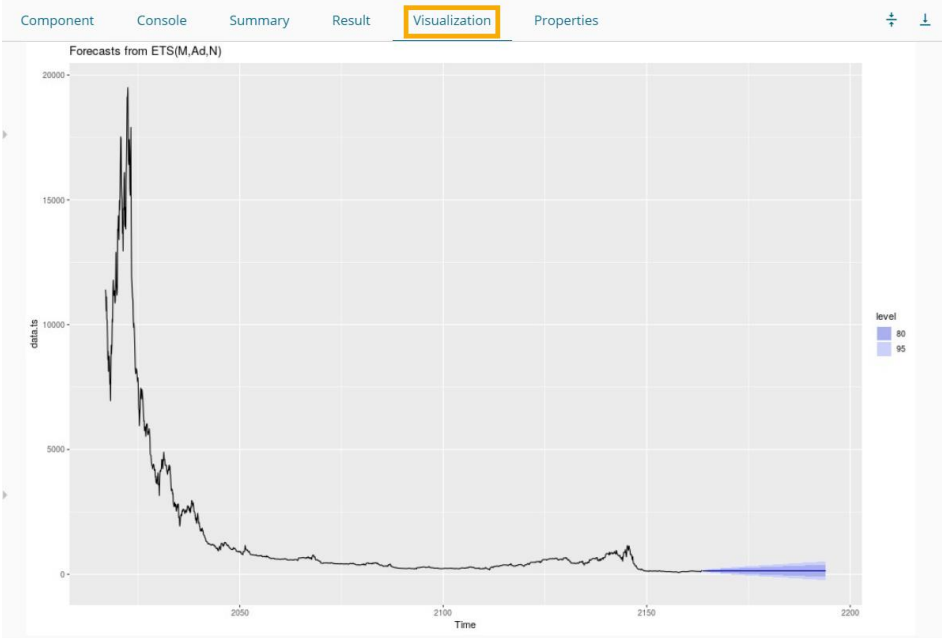
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



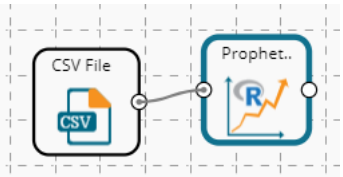
ix) Click the '**Visualization**' tab to see the processed data through a chart.



13.1.2.8. Prophet Forecasting

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

ii) Click the algorithm component.



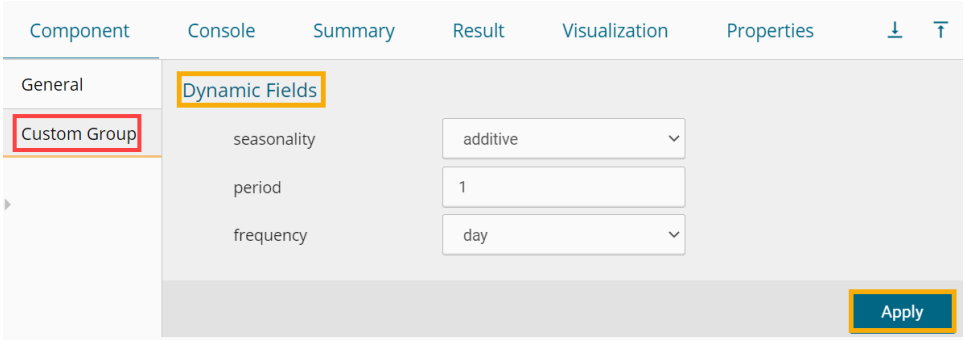
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. Seasonality: Select an option from the given choices that include additive and multiplicative options.

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

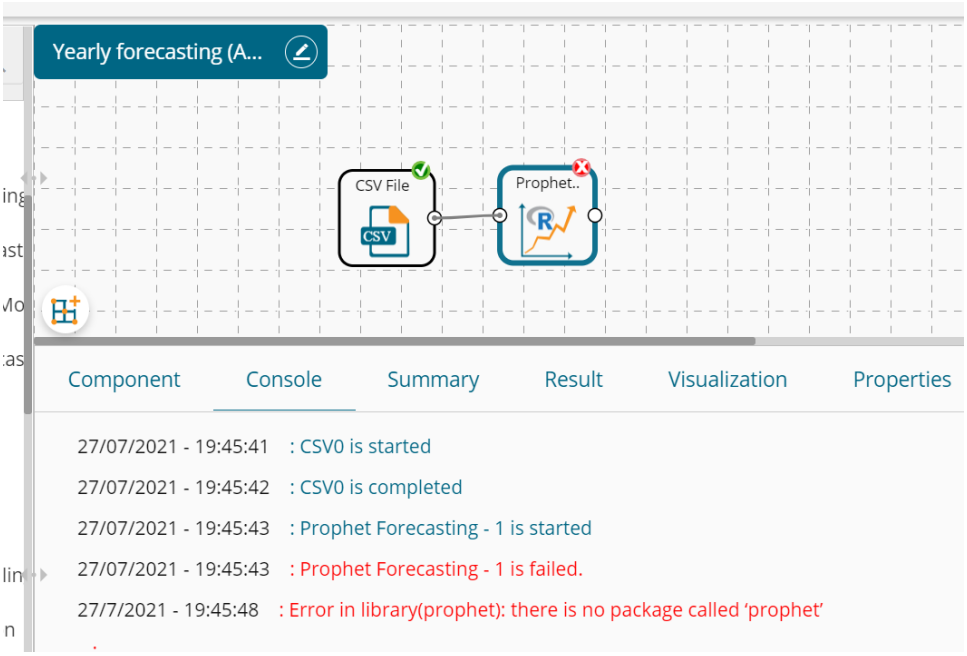
c. Frequency: Choose an option to set the frequency for running the data out of the given choices that include day, month, or yearly.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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) Click the '**Summary**' tab to open the model summary.

13.1.2.9. Forecasting Algorithms with 'Trend' Output Mode:

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

1. Triple Exponential Smoothing

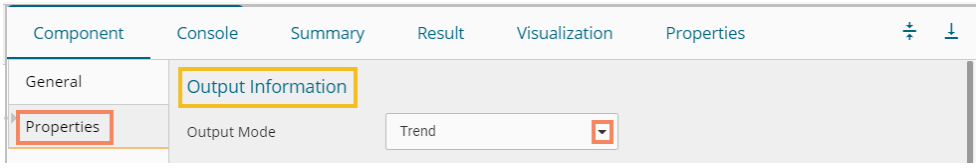
i) Drag the Forecasting algorithm to the workspace and connect it with the configured data source.

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

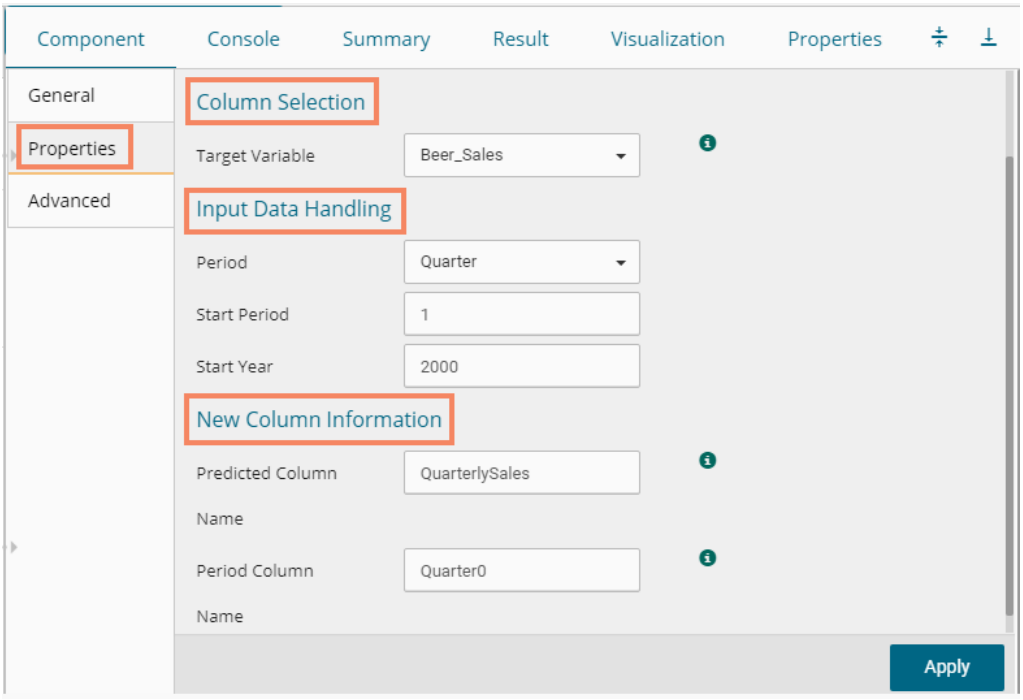
a. **Output Information**

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

1. **Trend**: Selecting this option displays source data along with predicted values for the given data set. A new column displaying the predicted values gets added in the Result view when the '**Trend**' output mode has been selected.



b. **Column Selection**
   i. **Target Variable:** Select the target variable for which you want to Apply forecasting analysis (The first selected option gets selected by default. Only numerical columns are accepted.)
c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.
   ii. **Period Per Year**: This field appears only when the selected '**Period**' option is **'Custom.'**
   iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for the **'Period'** field
   iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter **a** 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 is predefined. It gets displayed if the selected '**Output Mode**' is '**Trend**').
   ii. **Period Column Name:** Enter a name for the column containing a period value. (This field is predefined, but users can change the value if needed).

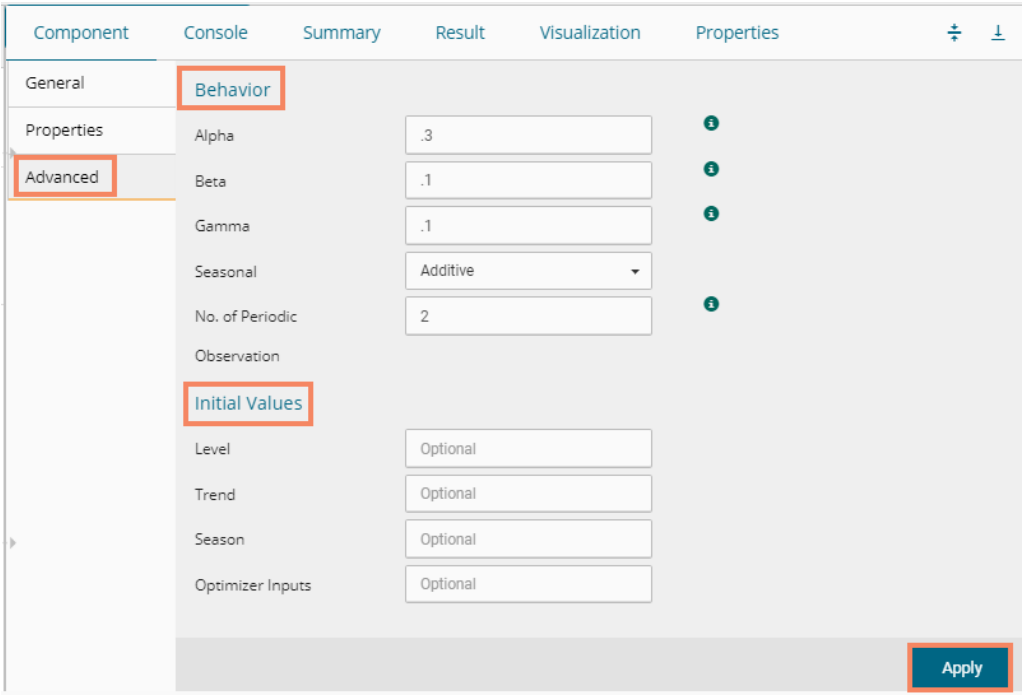


iii) Click the '**Advanced**' tab and configure it.
   a. Configure the following '**Behavior'** fields:

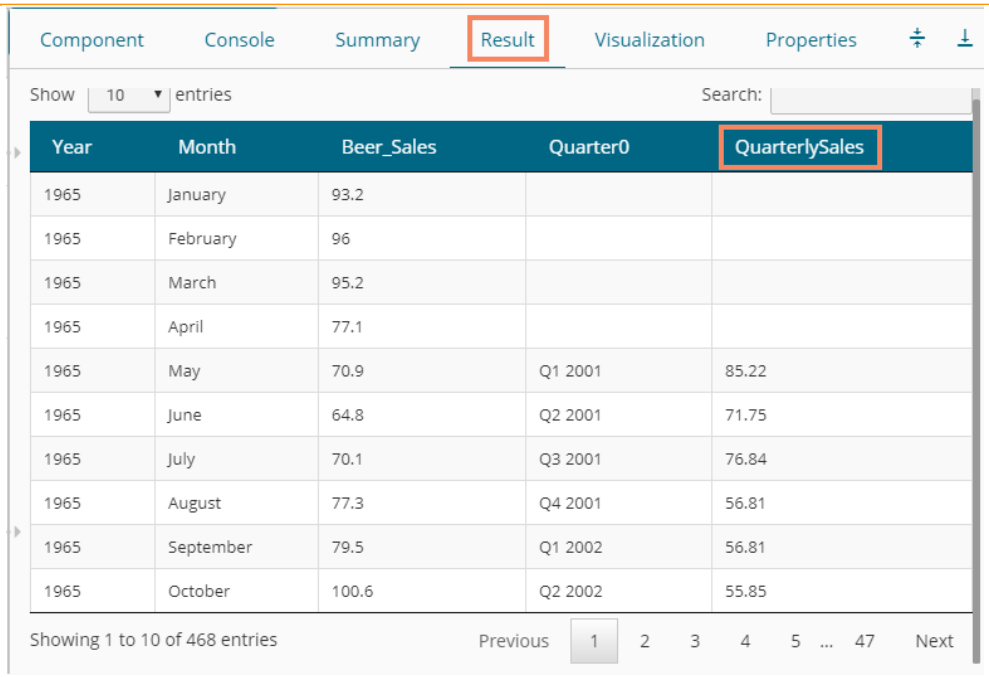
i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)
ii. **Beta:** Enter a valid double value in the given field for finding trend parameters. (Beta Range: 0-1.)
iii. **Gamma:** Enter a valid double value in the given field for finding seasonal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

b. Configure the following '**Initial Values**' information**:**
i. **Level:** Enter the initial value for the level. It is an optional field.
ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.
iii. **Season:** Enter initial values for finding seasonal parameters. It depends on the selected column. It is an optional field.
iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.

iv) Click the **'Apply'** option.

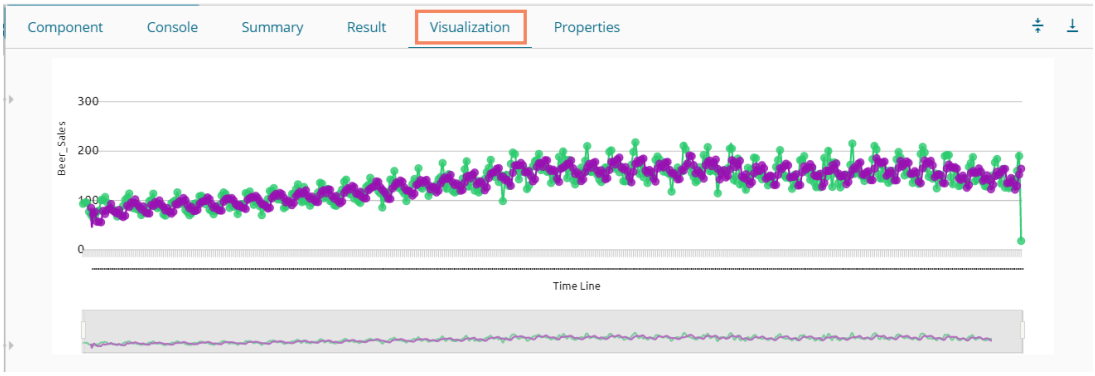


v) Run the workflow and open the '**Result**' tab after the Console process gets completed
a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.
In this case, the QuarterlySales column displays the predicted values in the Result tab.

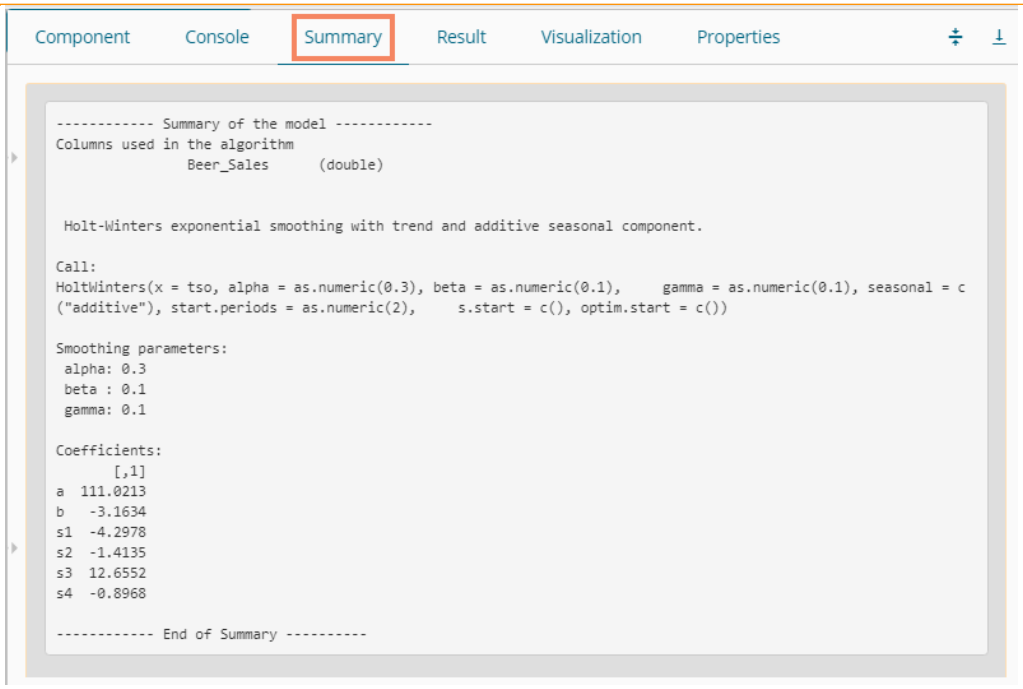


vi) Click the '**Visualization**' tab.

vii) The Result data gets displayed via the TimeLine Chart.



viii) Click the '**Summary**' tab to view the model summary.



Note:

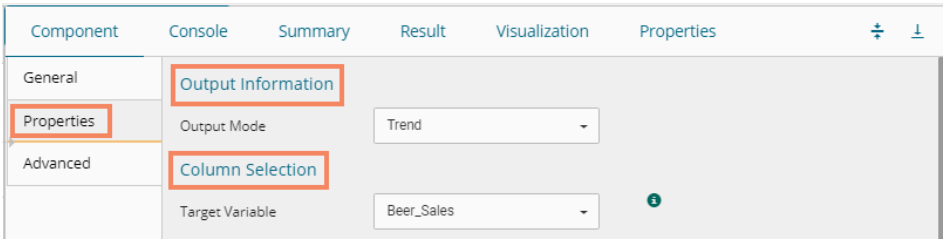
a. **'Properties'** and **'General'** sections remain the same for all the Forecasting sub-algorithms.
b. The **'Advanced'** tab displays different fields as per the Forecasting sub-types. Hence, '**Advanced**' fields for all the sub-types are explained over here. Predicted values get appended to the target column in the Result view for all the '**Forecasting**' algorithms.

1. Single Exponential Smoothing

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

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

b. **Column Selection**
   i. **Target Variable:** Select the target variable for which you want to Apply forecasting analysis (The 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.'

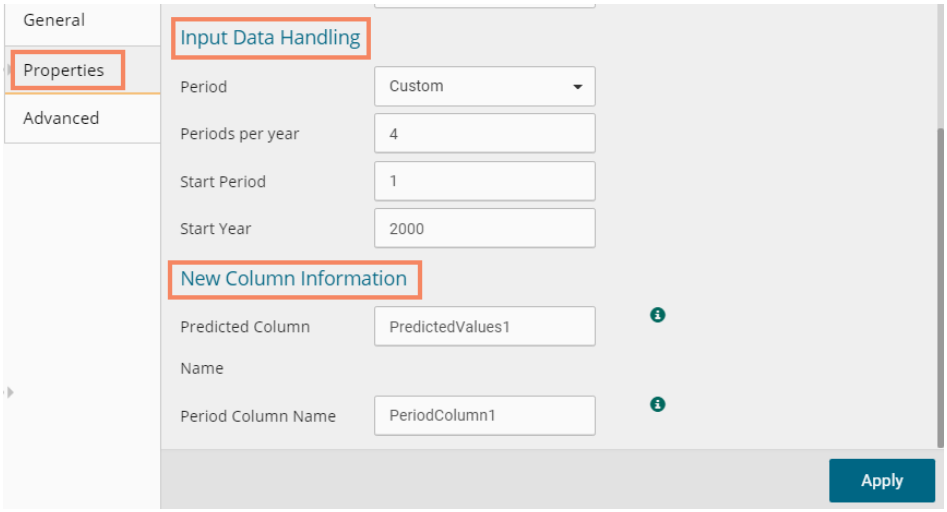
iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for **'Period'** field

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

d. **New Column Information**

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

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



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

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

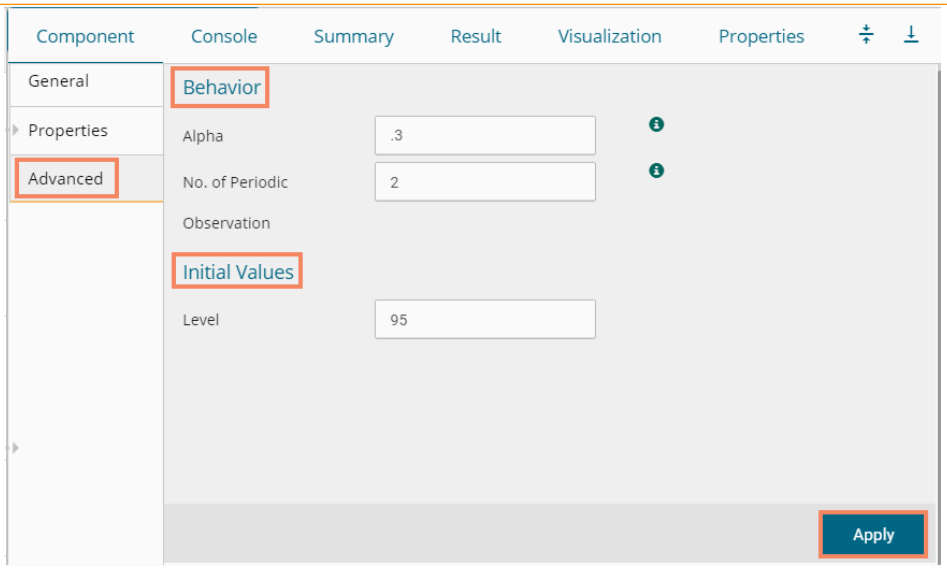
i. **Alpha:** Enter a valid double value in the given field for smoothing observations. (Alpha Range: 0<alpha<=1.)

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

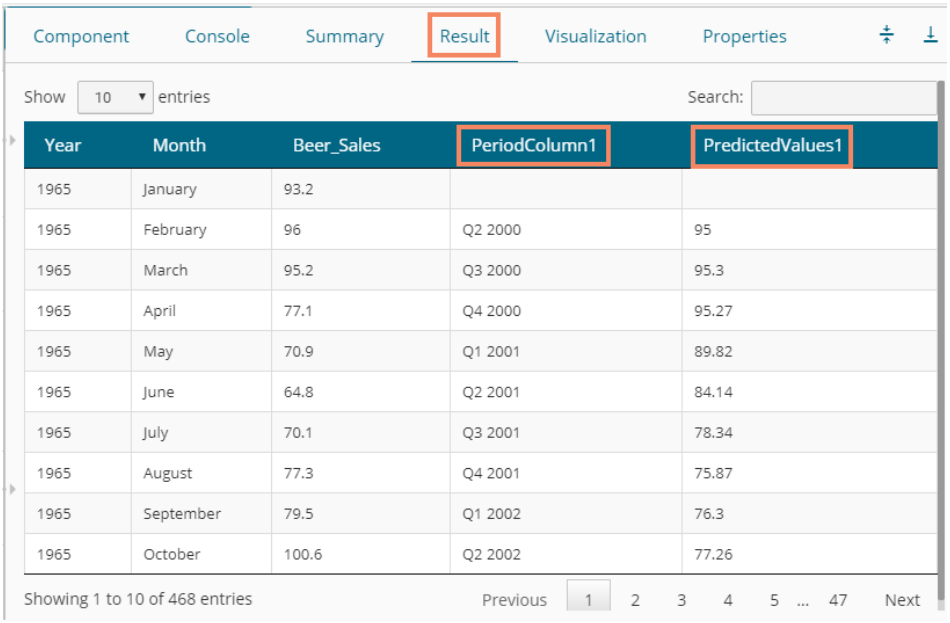
b. Configure the following '**Initial Values**' information**:**

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

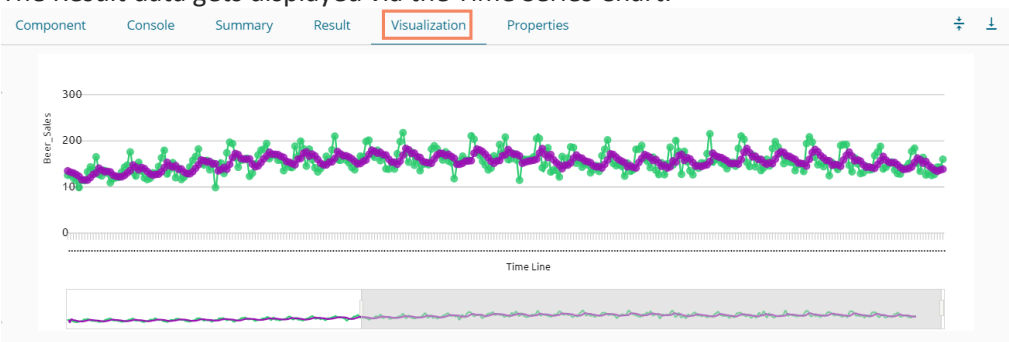
iii) Click the '**Apply**' option.



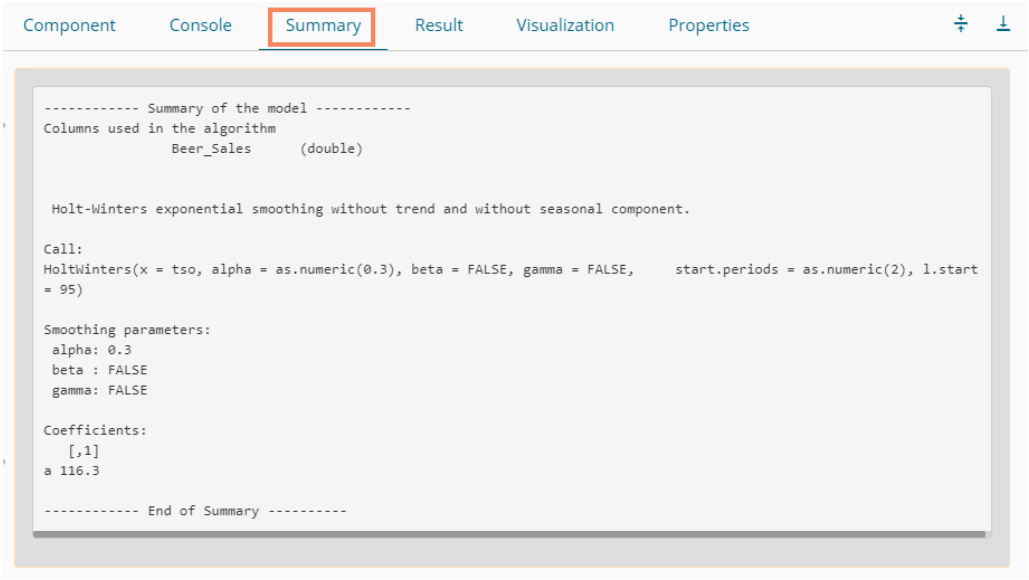
iv) Run the workflow and open the 'Result' tab after the Console process gets completed
a. Click the dragged algorithm component from the workspace and then click
b. Click the '**Result**' tab.



v) Click the '**Visualization**' tab.
vi) The Result data gets displayed via the Time Series Chart.



vii) Click the 'Summary' tab to view the model summary.



2. Double Exponential Smoothing

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

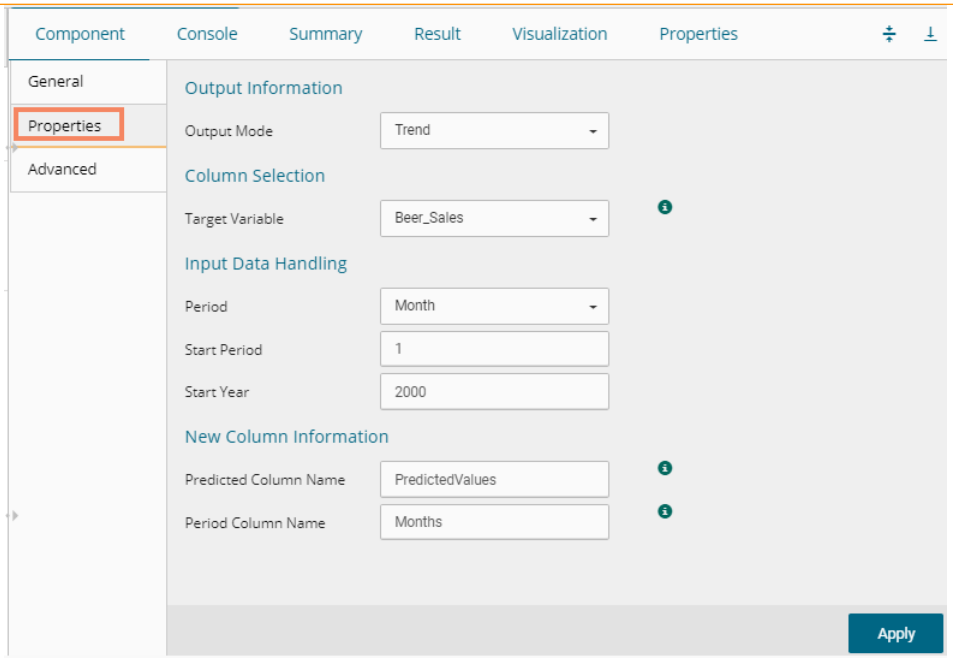
ii) Configure the following fields in the 'Properties' tab:

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

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

c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.
   ii. **Start Period:** Enter a value between 1 and the value specified for the selected option for the **'Period'** field
   iii. **Start Year:** Enter a year from which you want the data entries to be considered. Enter **a** 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 is predefined and displayed if the selected Output Mode is '**Trend**').
   iv. **Period Column Name:** Enter a name for the column containing a period value. (This field is predefined, but users can change the value if needed).



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

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

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

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

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

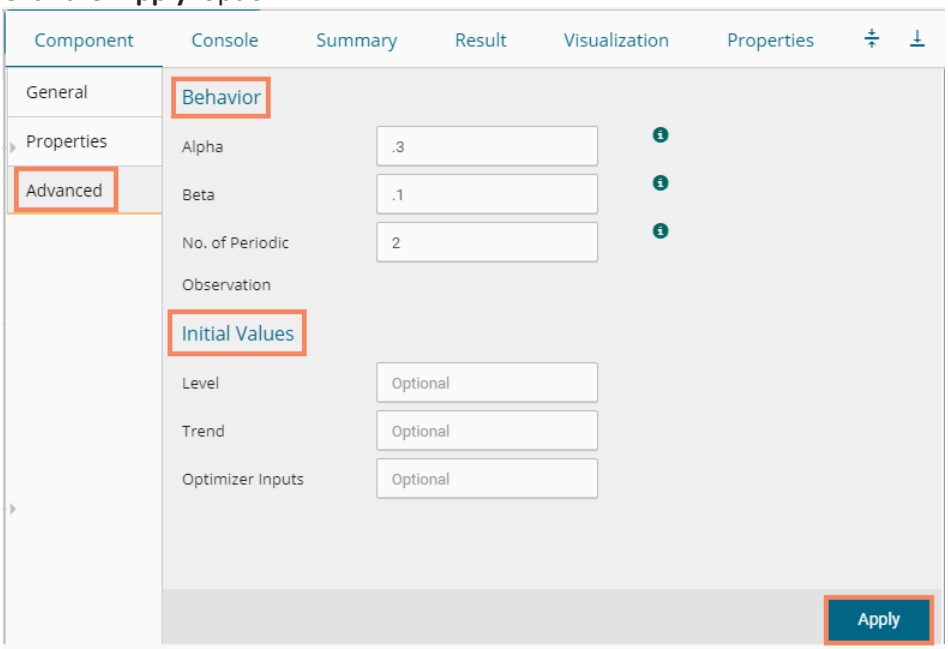
b. Configure the following '**Initial Values**' information**:**

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

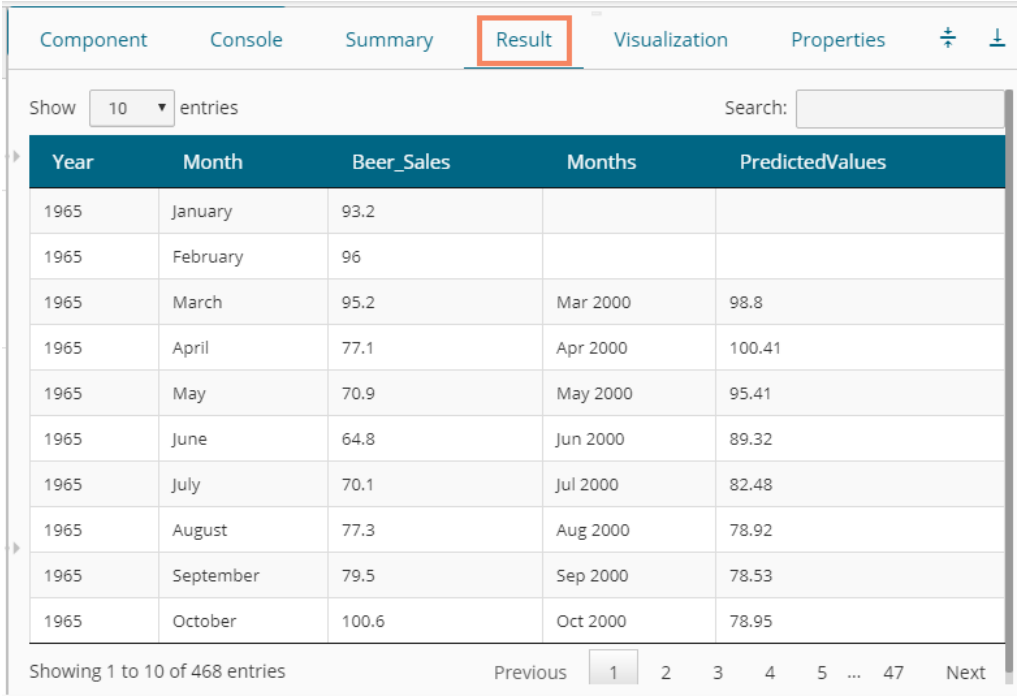
ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.

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

iv) Click the '**Apply**' option.

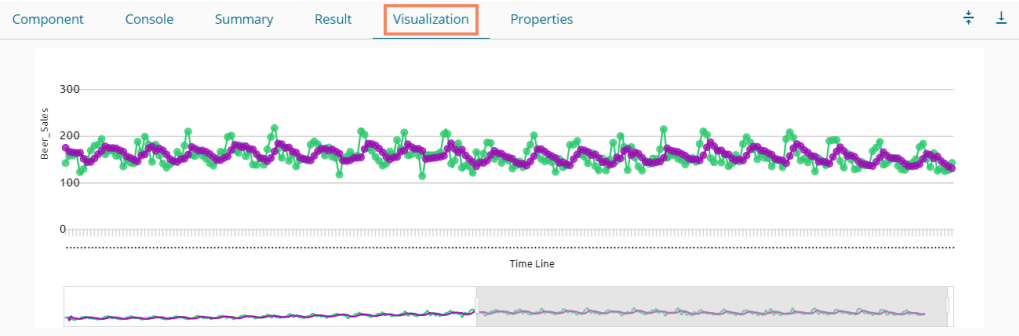


v) Run the workflow and open the 'Result' tab after the Console process gets completed
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



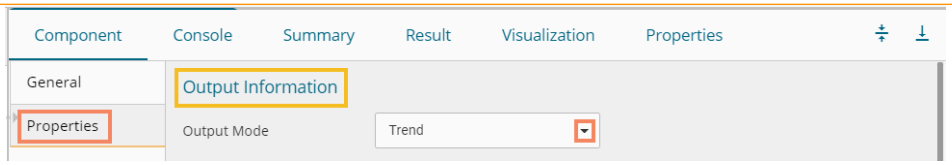
Year	Month	Beer_Sales	Months	PredictedValues
1965	January	93.2		
1965	February	96		
1965	March	95.2	Mar 2000	98.8
1965	April	77.1	Apr 2000	100.41
1965	May	70.9	May 2000	95.41
1965	June	64.8	Jun 2000	89.32
1965	July	70.1	Jul 2000	82.48
1965	August	77.3	Aug 2000	78.92
1965	September	79.5	Sep 2000	78.53
1965	October	100.6	Oct 2000	78.95

vi) Click the '**Visualization**' tab.
vii) The Result data gets displayed via the TimeLine Chart.

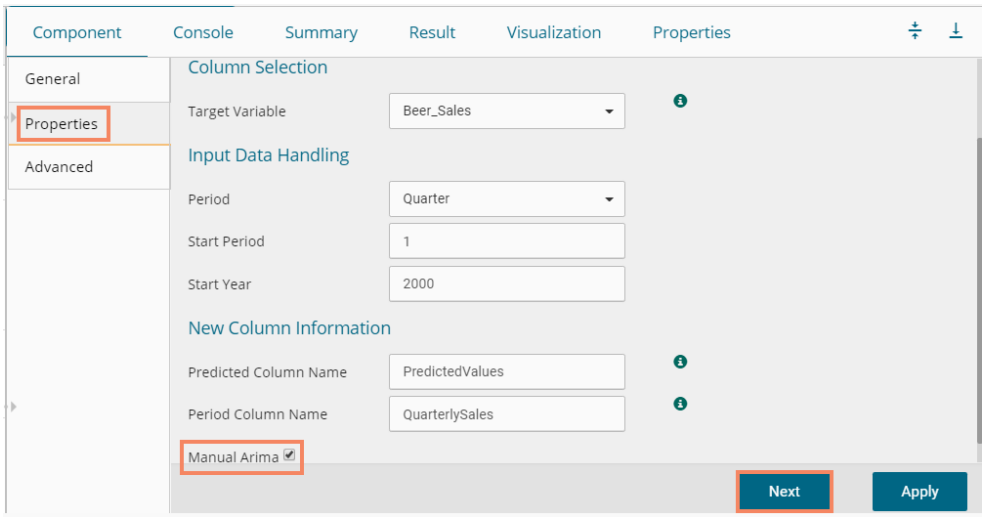


3. R-ARIMA

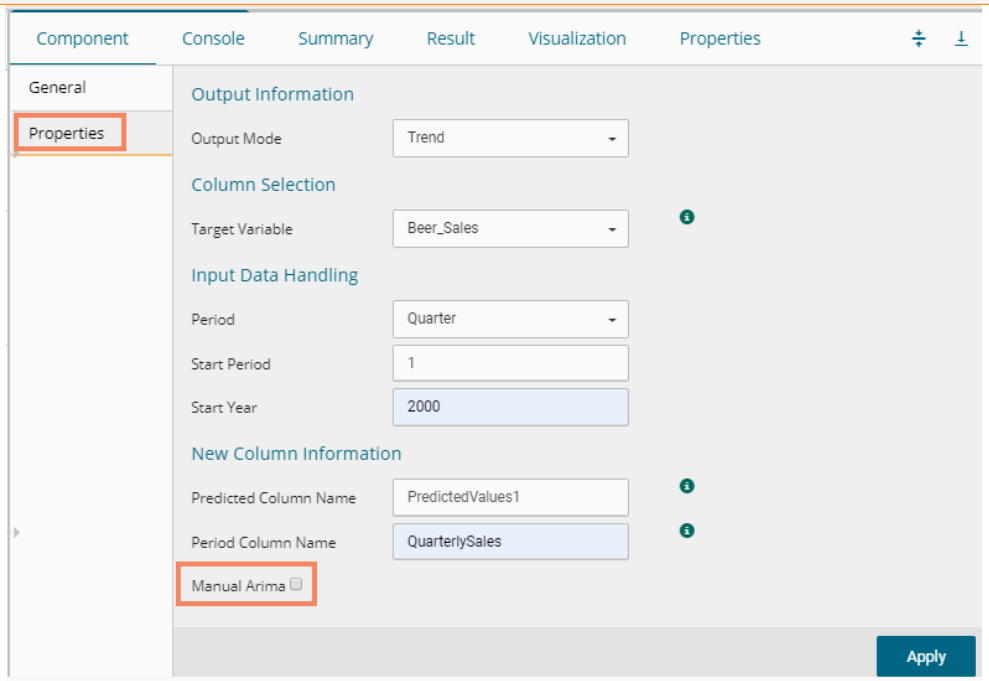
i) Select the '**Trend**' option from the '**Output Mode**' drop-down menu.
ii) Configure the following fields in the '**Properties**' tab:
   a. **Output Information**
      i. **Output Mode**: Select a mode in which you want to display output data
         1. **Trend**: Selecting this option displays source data along with predicted values for the given data set. A new column '**Predicted Values**' gets added in the Result view when the '**Trend**' output mode has been selected.
         2. **Forecast**: Selecting this option displays forecasted values for the given period. The Result values are 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 (The first selected option gets selected by default. Only numerical columns are accepted.)
c. **Input Data Handling**
   i. **Period:** Select a period of forecasting by choosing any one option from the drop-down menu.
   ii. **Period Per Year**: This field appears only when the selected '**Period**' option is **'Custom.'**
   iii. **Start Period:** Enter a value between 1 and the value specified for the selected option for the **'Period'** field
   iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter **a** 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 is predefined and displayed if the selected Output Mode is '**Trend**')
   ii. **Period Column Name:** Enter a name for the column containing the period value (This field will be predefined, but users can change the value if needed).
   iii. **Manual Arima**: Enable this option to get Behaviour fields in the Advanced tab. If the Manual Arima option is enabled, then the '**Next**' option appears on the Properties configuration page, and the user can click it to configure the Advanced fields.



Properties tab with Manual Arima option Disabled



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

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

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

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

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

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

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

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

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

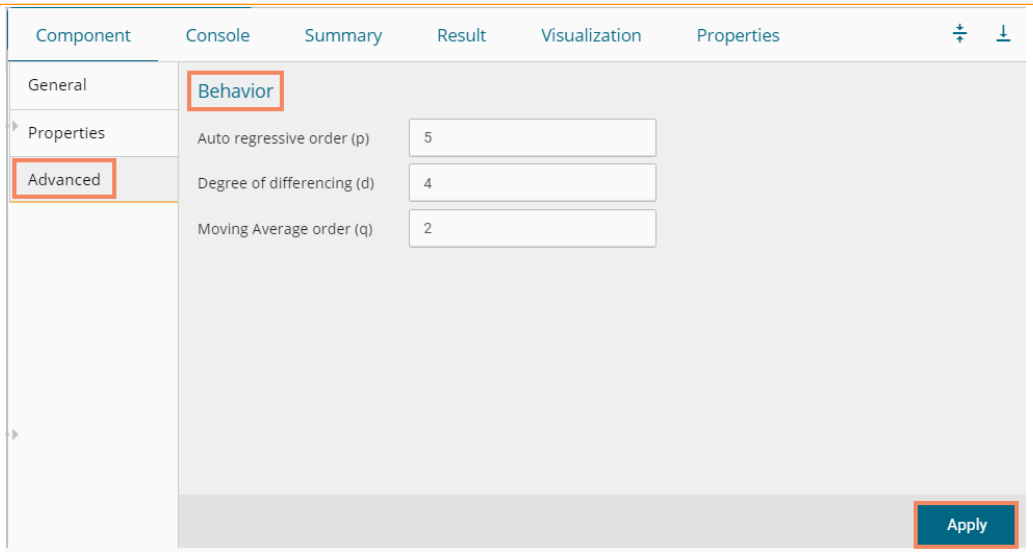
ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.

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

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

iv) Click the '**Apply'** option**.**

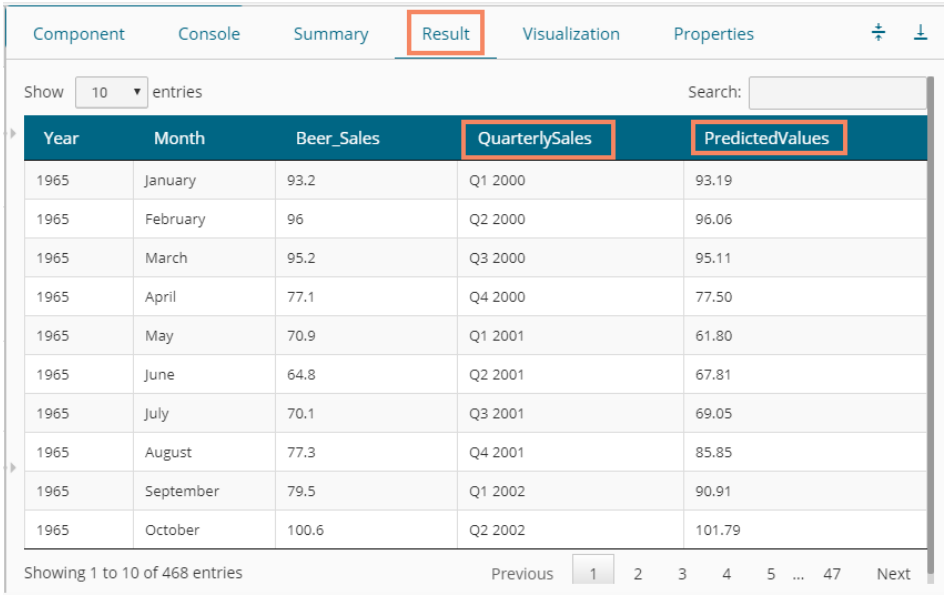
Advanced Tab when Manual Arima is enabled



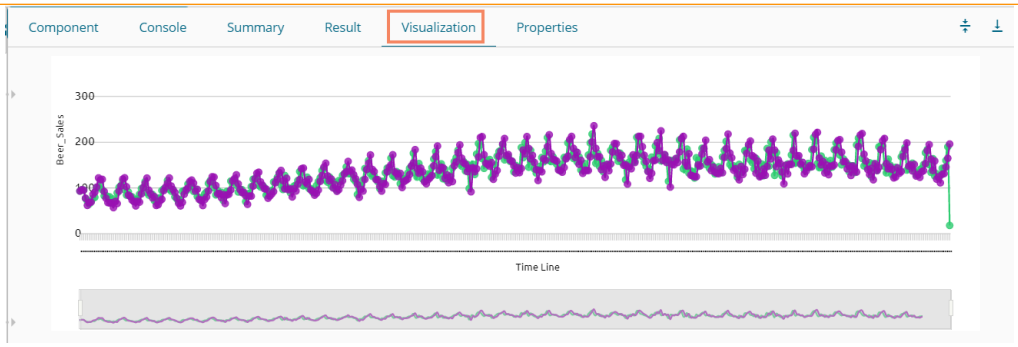
Note: The Advanced tab does not appear if the Manual Arima option is disabled.

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

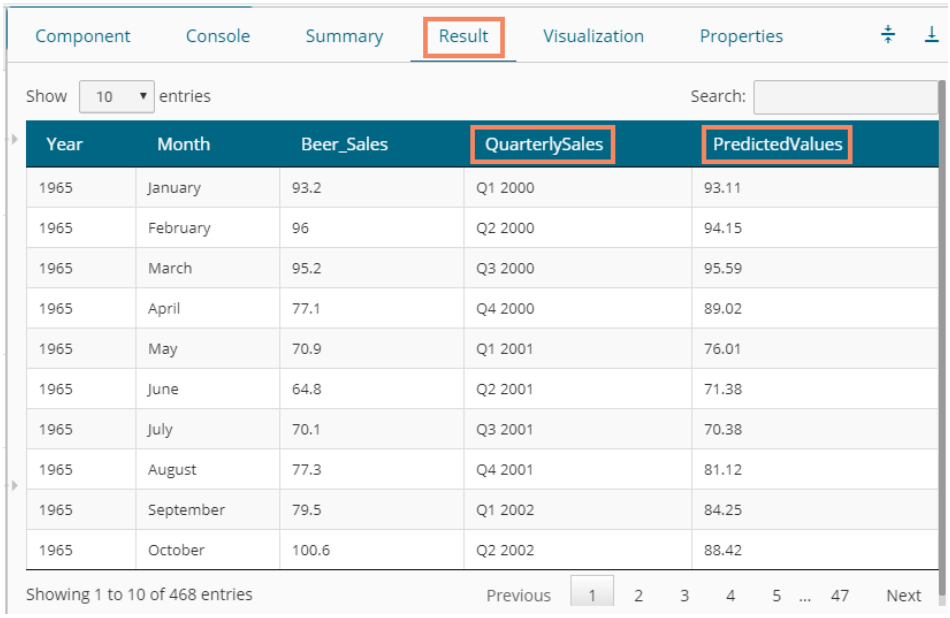
The following is the 'Result' tab display when 'Manual Arima' is Enabled

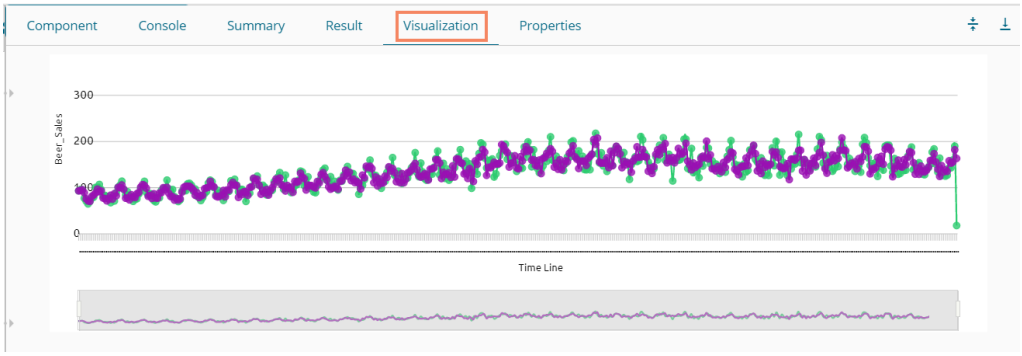


vi) Click the '**Visualization**' tab.
vii) The Result data gets displayed via the TimeLine Chart.



The following are the 'Result' and 'Visualization' tabs for the selected dataset when 'Manual Arima' is Disabled.





4. R-Auto Forecasting

i) Select the '**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

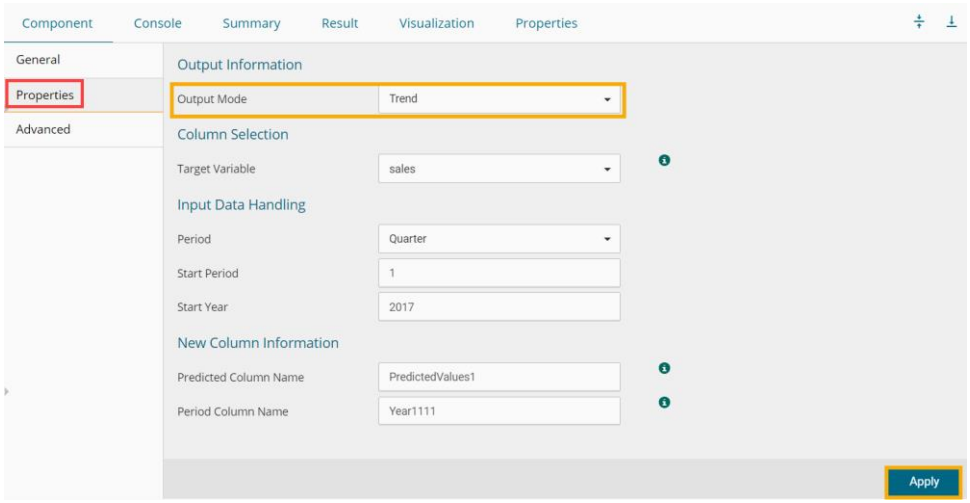
1. **Trend**: Selecting this option displays source data along with predicted values for the given data set. A new column '**Predicted Values**' gets added in the Result view when the '**Trend**' output mode has been selected.

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

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

c. **Input Data Handling**
   i. **Period:** Select the 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 the **'Period'** field
   iv. **Start Year:** Enter a year from which you want the data entries to be considered. Enter **a** 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 is predefined and displayed only if the selected Output Mode is '**Trend**').
   ii. **Period Column Name:** Enter a name for the column containing the period value (This field will be predefined, but users can change the value if needed).



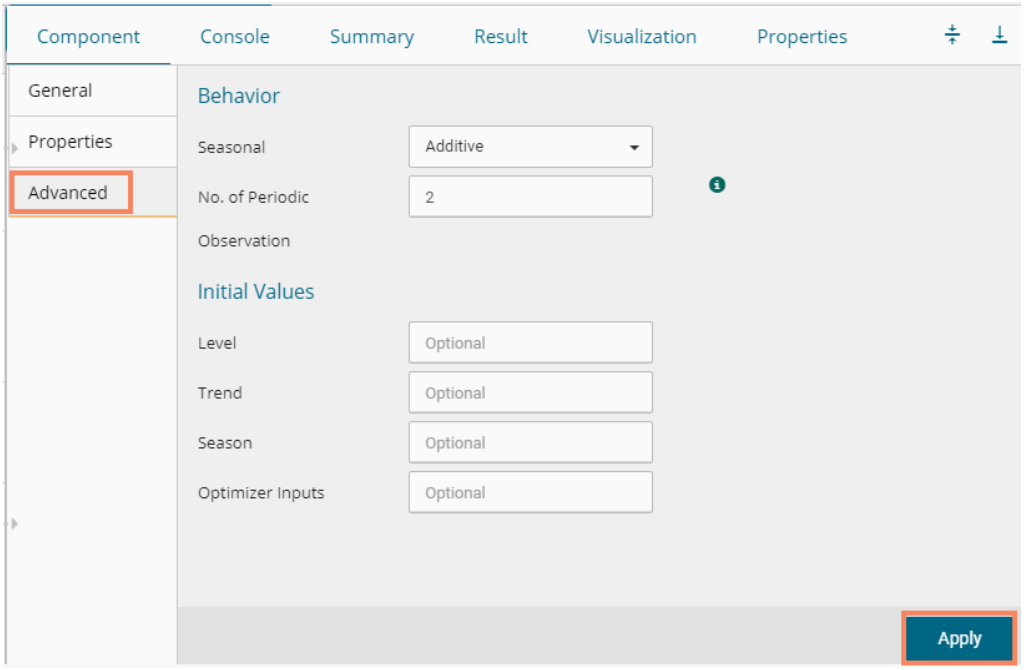
iii) Click the '**Advanced**' tab and configure

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

b. Configure the following '**Initial Values**' information**:**
   i. **Level:** Enter the initial value for the level. It is an optional field.

ii. **Trend:** Enter the initial value for finding trend parameters. It is an optional field.
iii. **Season:** Enter initial values for finding seasonal parameters. It depends on the selected column. It is an optional field.
iv. **Optimizer Inputs:** Enter the initial values given for alpha, beta, gamma required for the optimizer. It is an optional field.

iv) Click the '**Apply**' option.

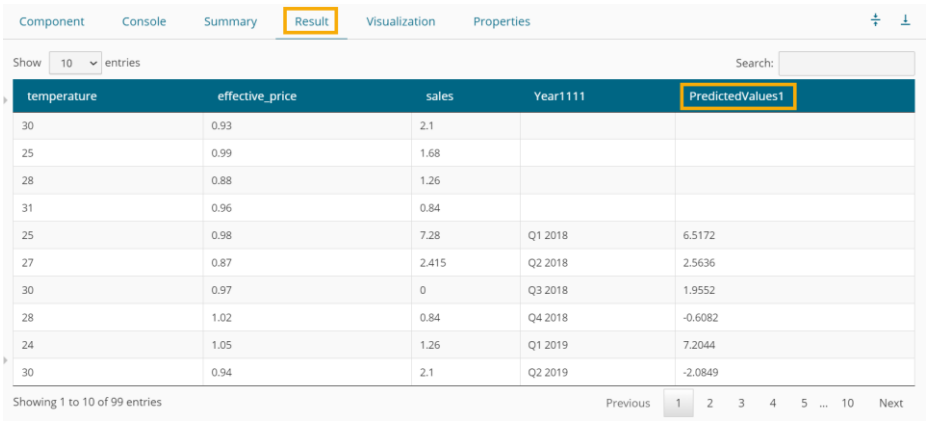


viii) Run the workflow and open the 'Result' tab after the Console process gets completed

a. Click the dragged algorithm component onto the workspace.

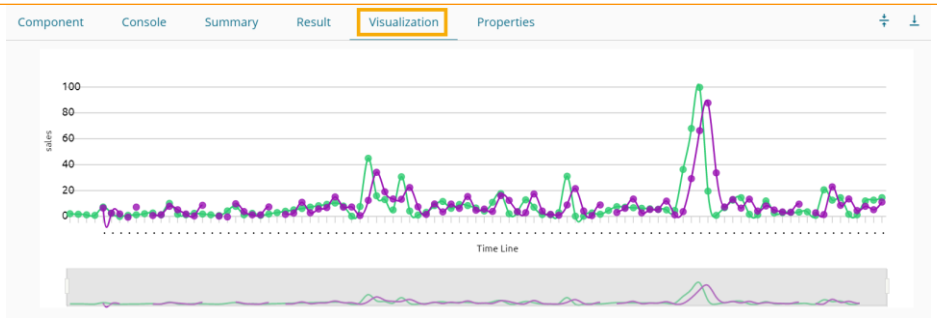
b. Click the '**Result**' tab.

c. A new column with the **predicted values** gets added to the Result data.

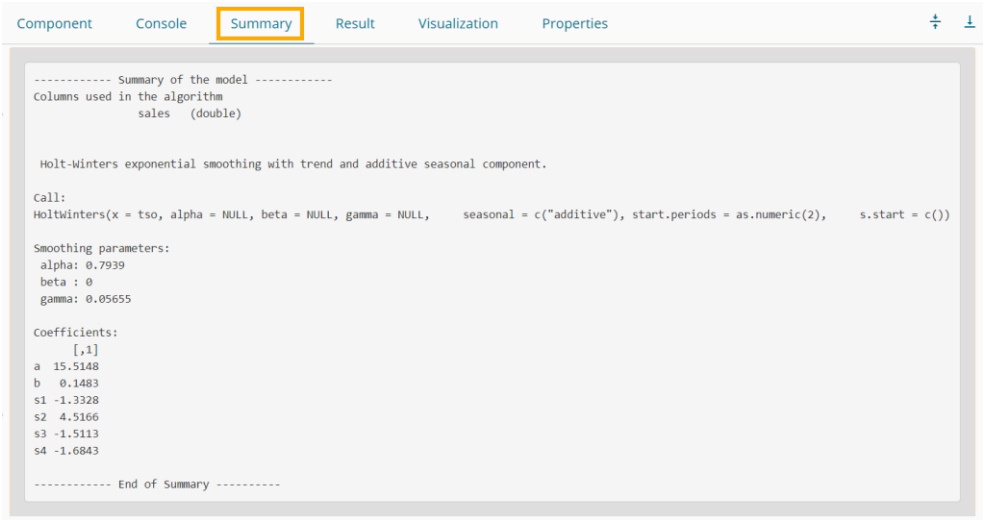


v) Click the '**Visualization**' tab.

vi) The Result data gets displayed via the TimeLine chart.



Note: Click the '**Summary**' tab to view the model summary for the Forecasting models with '**Trend**' as the output mode.

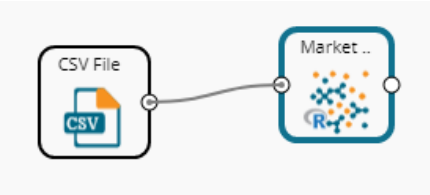


13.1.3. Association

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

13.1.3.1. 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 the **'Rules'** option displays rules for the selected dataset.
2. Selecting the **'Transaction'** option displays 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**,' two types of the result view appear.

ii. **Item Columns:** Select the item columns on which you want to Apply association rules/analysis. Choose at least one option from the drop-down menu. This field displays numerical and strings columns. It cannot display Date columns.

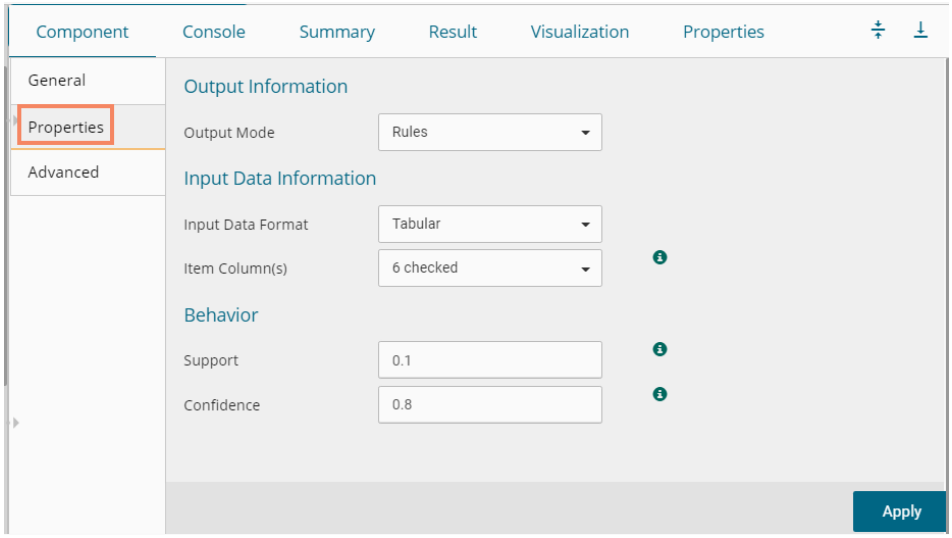
iii. **Transaction Id Column:** Select the column containing Transaction Ids to which you can apply the algorithm. (This field gets added when the selected 'Input Data Information' is 'Transactions')

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

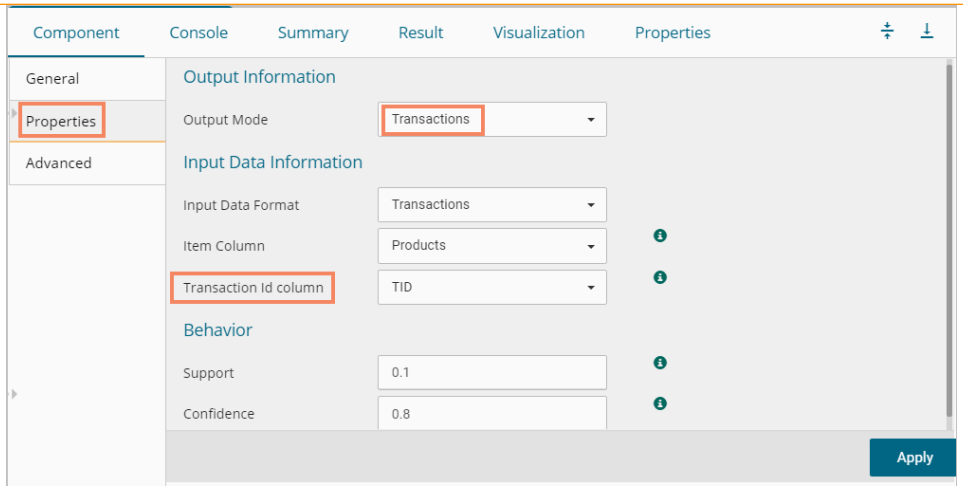
c. Behavior

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

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



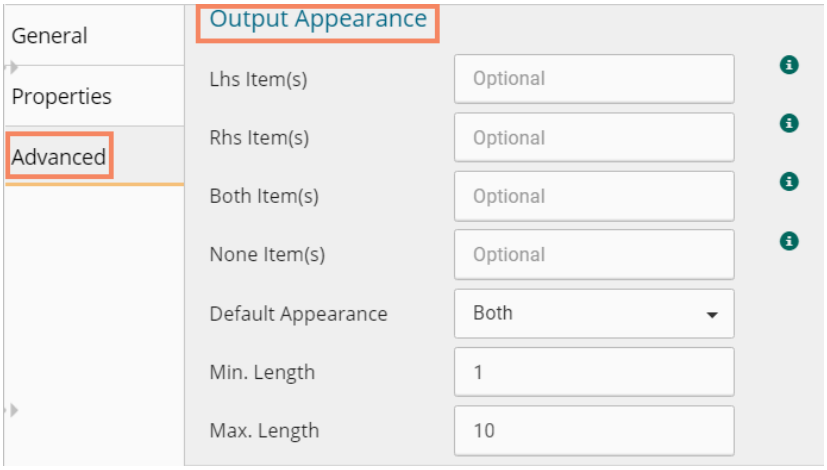
Properties fields with '**Transactions**' as '**Input Data Information**'



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

a. Output Appearance

i. **Lhs Item(s):** Enter item tags separated by a comma which should display on the left-hand side of rules or item sets
ii. **Rhs Item(s):** Enter item tags separated by a comma which should display on the right-hand side of rules or item sets
iii. **Both Item(s):** Enter item tags separated by a comma which should display on both sides of rules or item sets
iv. **None Item(s):** Enter item tags separated by a comma which need not display in the rules or item sets
v. **Default Appearance**: Select the default appearance of the items out of the above-given choices using a drop-down menu
vi. **Min Length:** Set a minimum length value. The default value for this field is 1.
vii. **Max Length:** Set a maximum length value. The default value for this field is 10.



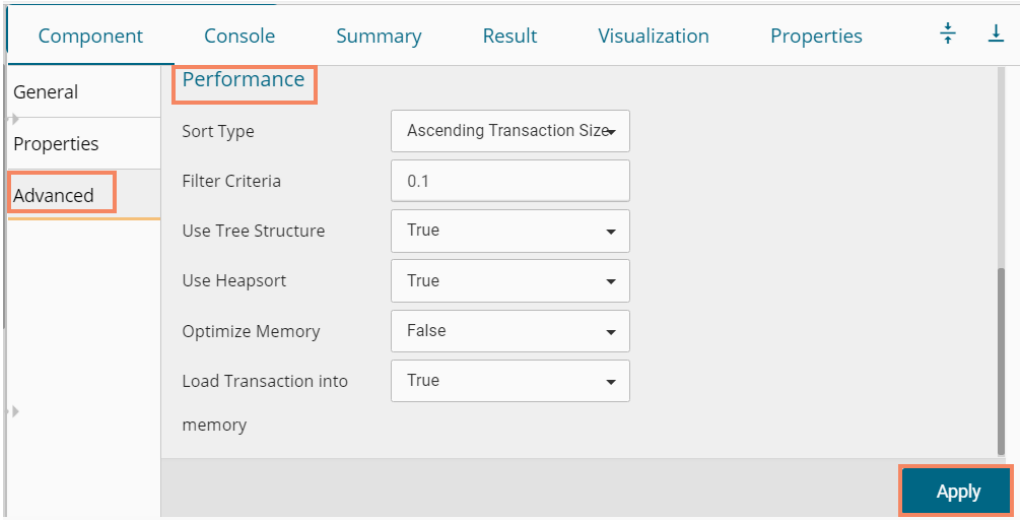
b. Performance

i. **Sort Type**: Select a sort type using the drop-down menu for sorting items based on their frequency.
ii. **Filter Criteria**: Enter an indicating numerical value for filtering unused items from Transactions. The default value for this field is 0.1.
iii. **Use Tree Structure**: Selecting **the' True'** option from the drop-down menu organizes the transaction as a prefix tree.

iv. **Use Heapsort**: Selecting the **'True'** option from the drop-down menu uses heapsort against quicksort for sorting transactions.

v. **Optimize Memory**: Selecting **the' True'** option from the drop-down menu minimizes memory usage instead of maximizing speed.

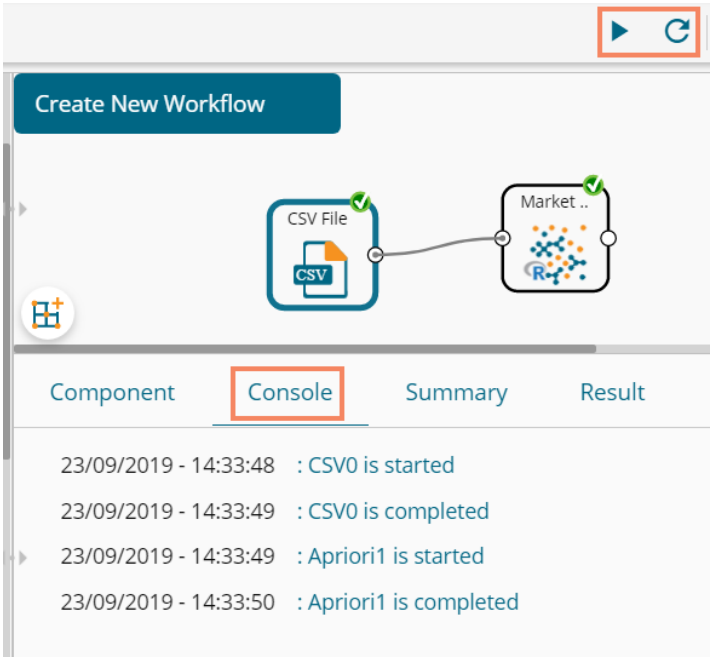
vi. **Load Transaction into Memory**: Selecting **'True'** from the drop-down menu loads transactions into memory.



iv) Click the **'Apply'** option**.**

v) Run the workflow after getting a success message.

vi) The user gets directed to the '**Console**' tab displaying the progress of the process.



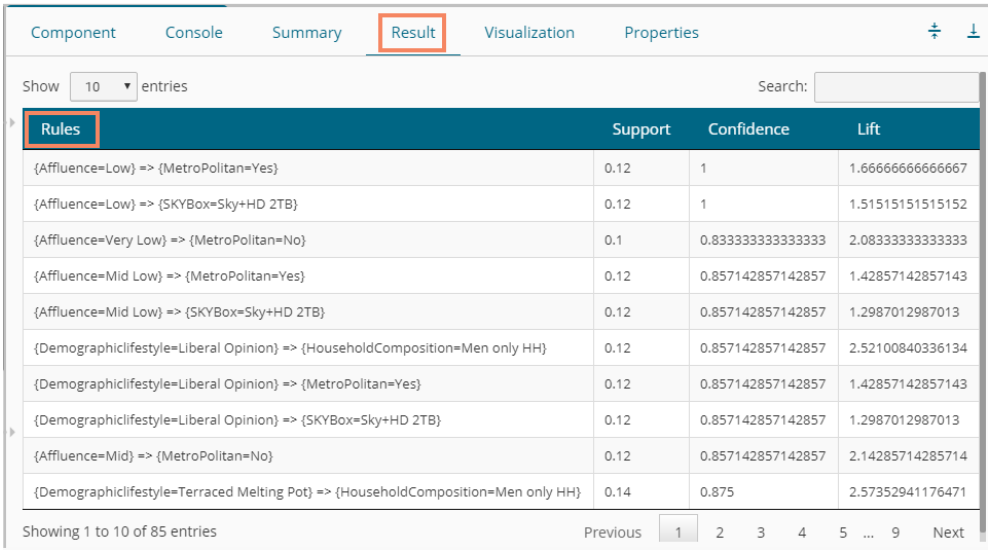
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

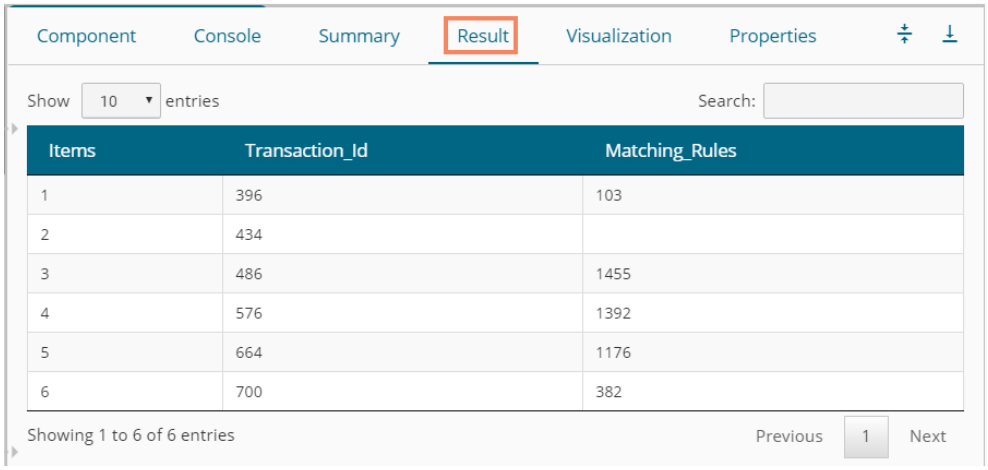
viii) Two types of Result view gets displayed:

a. **'Rules'** gets displayed as a first column in the Result data (When the selected '**Output Mode**' option is '**Rules**').



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

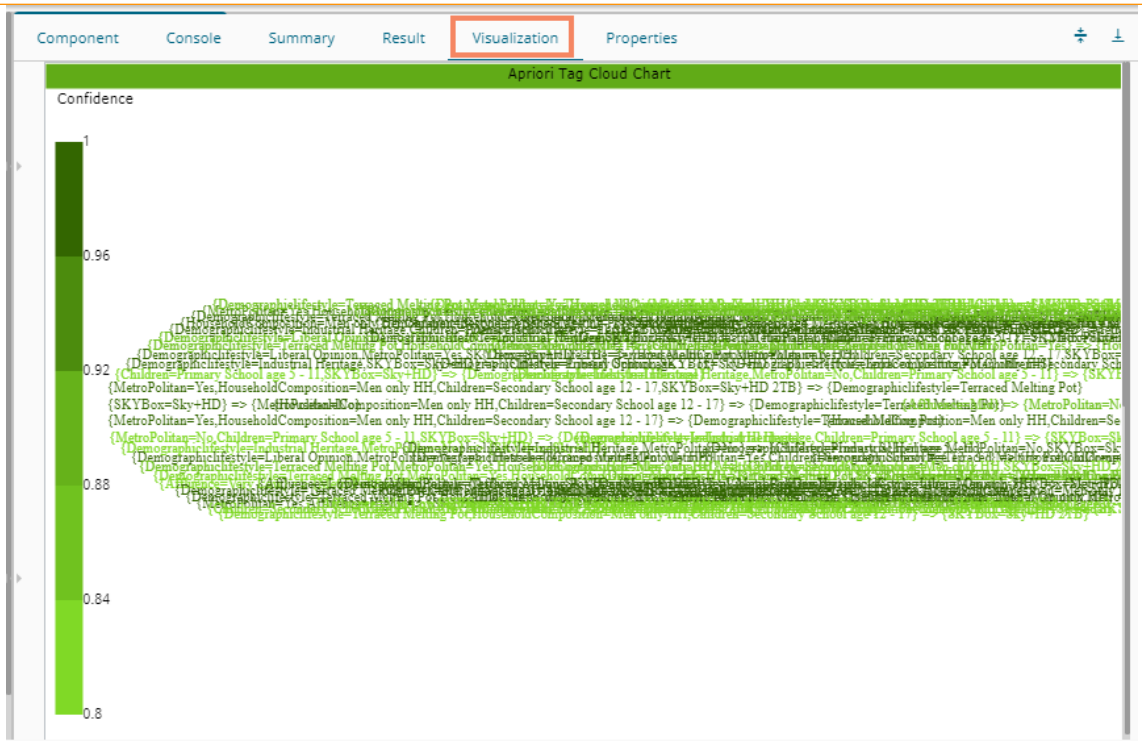
The matching rules for the selected items get displayed through the **'Matching_Rules'** column.



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

x) The Result data will be displayed via the Apriori Tag Cloud chart.

a. The Visualization tab for the '**Rules**' output mode



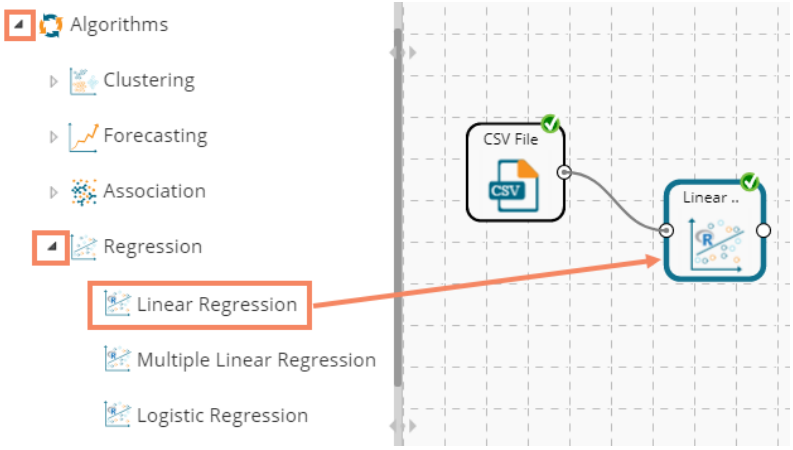
b. Visualization tab for the '**Transactions**' output mode

13.1.4. 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. The Regression Analysis tree node has the following algorithms:

13.1.4.1. Linear Regression

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



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

a. Column Selection

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

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

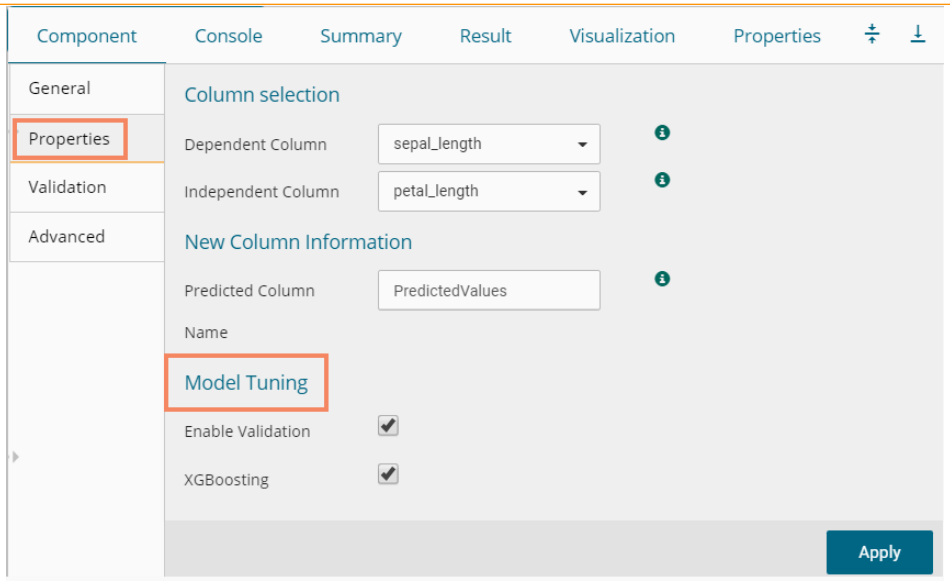
b. New Column Information

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

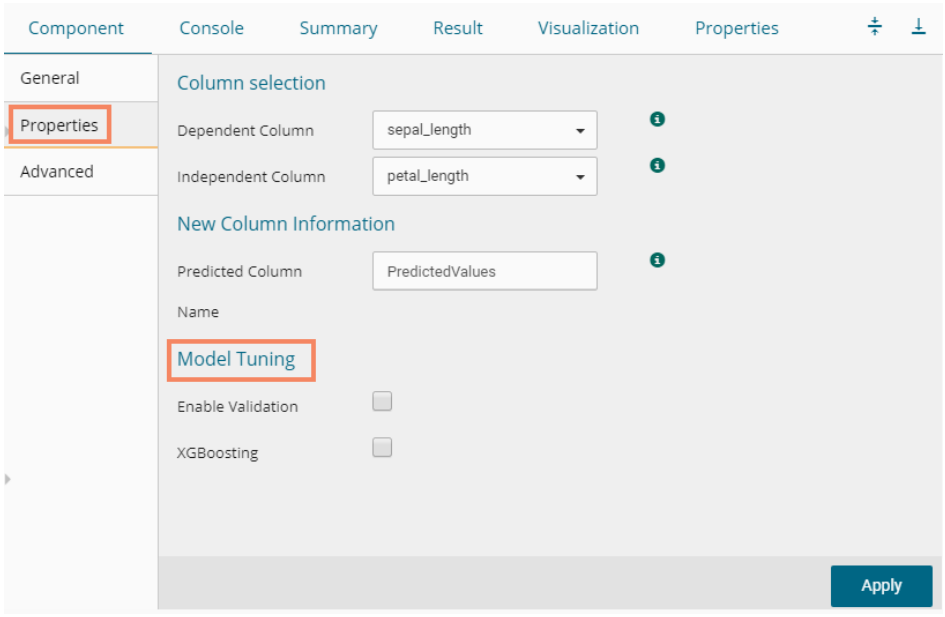
c. Model Tuning

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

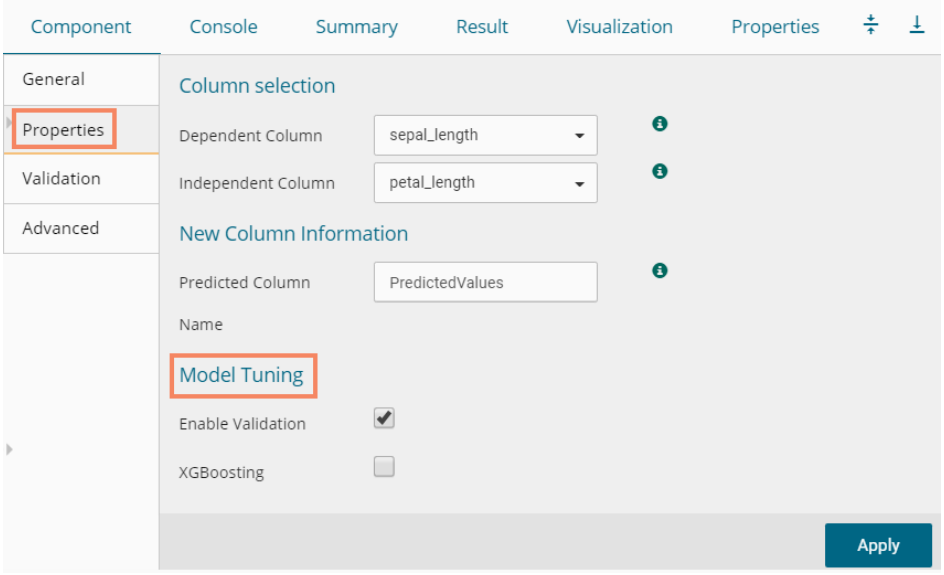
ii. XG Boosting: Use a checkmark in the box to enable XG Boosting
Scenario-1- Validation and XG Boosting are enabled



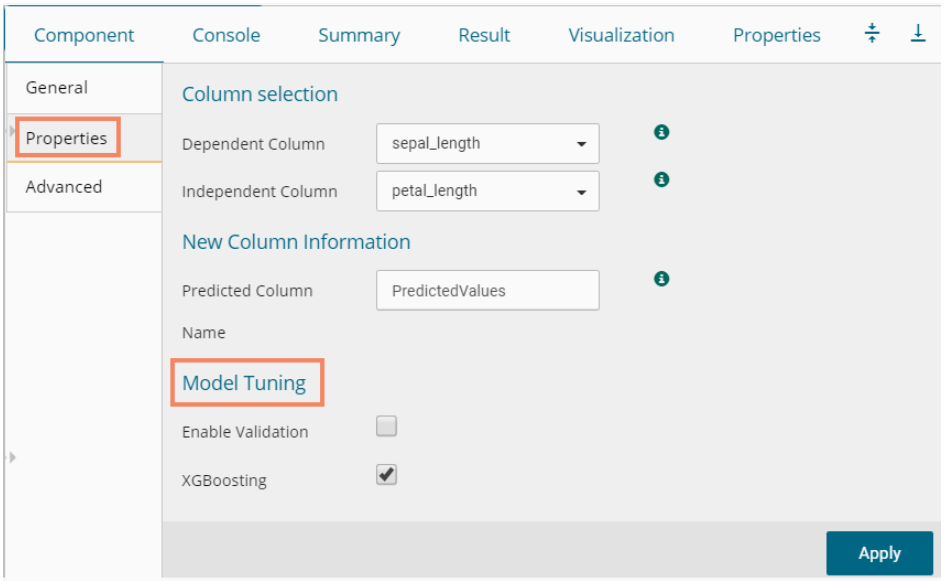
Scenario-2- Validation and XG Boosting are disabled



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



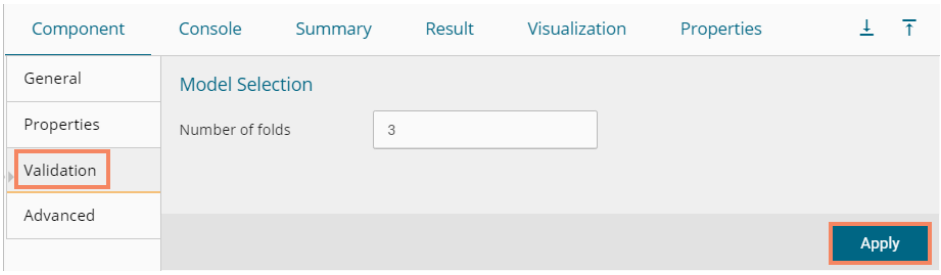
Scenario-4- Validation is disabled, but XG Boosting is enabled



iii) **Click the 'Validation' tab and configure it:**

a. Model Selection (when XG Boosting is enabled)

i. **Number of folds:** Enter a number deciding the creation of folds in a model.

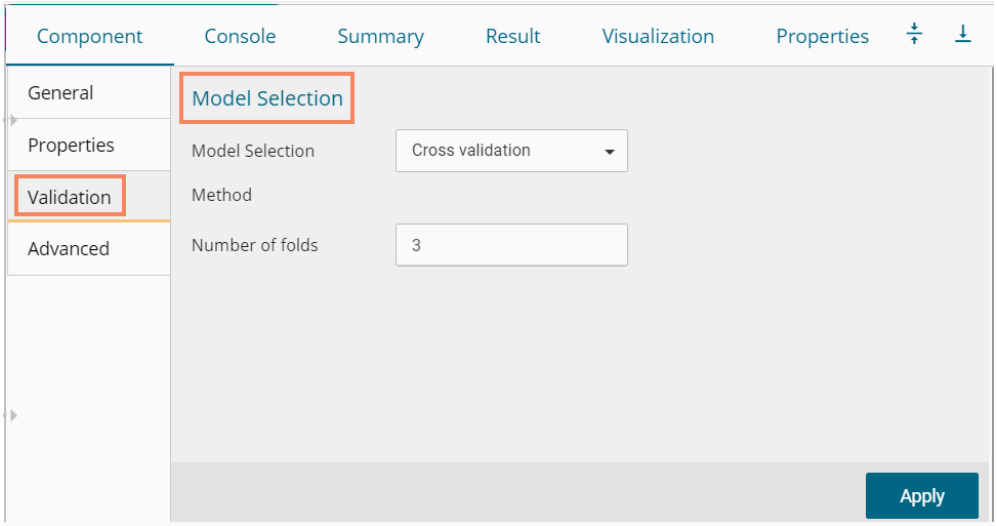


Validation tab when XG Boosting is disabled

a. **Model Selection**

i. **Model Selection Method:** Select a Model Method using the drop-down menu

ii. **Number of folds:** Enter a number deciding the creation of folds in a model



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

Advanced tab when XG Boosting and Validation are disabled

a. **Input Data Handling**

i. **Missing Values**: Select a method to deal with missing values from the drop-down menu

1. **Ignore**: Select this option to skip the records containing missing values from the dependent and independent columns.
2. **Keep**: Select this option to retain the records containing missing values while performing the calculation.
3. **Stop**: Select this option to stop the algorithm application if a value is missing in any column.

b. **Behavior**

i. **Allow Singular Fit**: Select an option for providing value to the Boolean Column

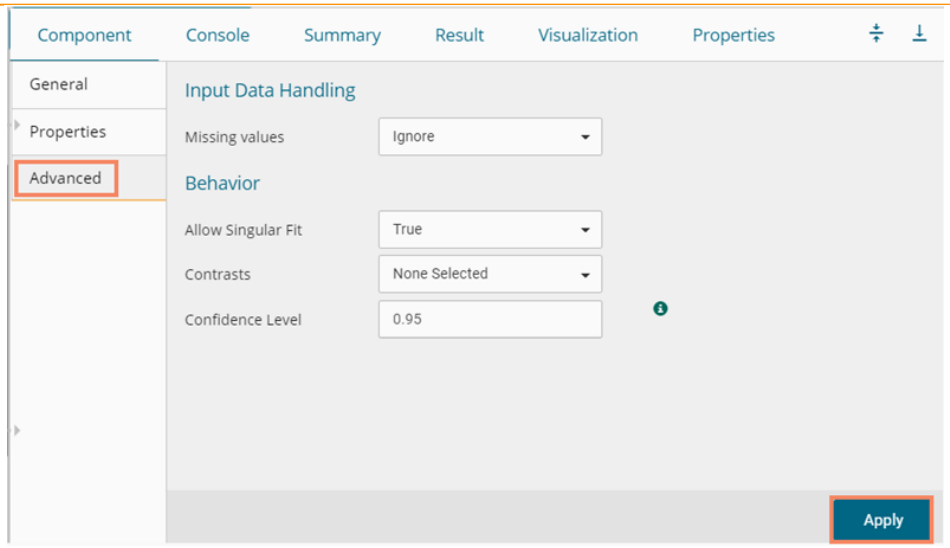
1. **True:** Select this option to ignore aliased coefficients from the coefficient covariance matrix.
2. **False:** Select this option to show an error in a model containing aliased coefficients

ii. **Contrasts**: Select this option to display a list of contrast items that can be used for some variables in the model. The available options are:

1. **Contr. Treatment**
2. **Contr.poly**
3. **Contr.sum**
4. **Contr.helmert**

iii. **Confidence Level**: Enter a value specifying accuracy (Confidence Level) of predictions for the algorithm. This field takes 0.95 as the default value.

iv. Click the '**Apply**' option.

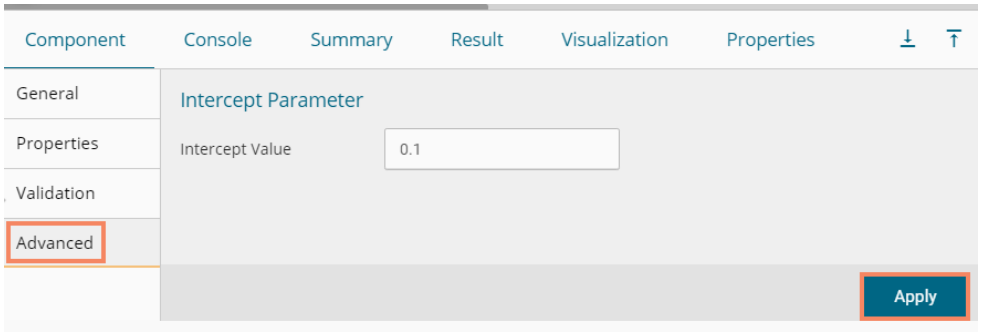


Advanced Tab when XG Boosting is disabled, but Validation is enabled

a. **Intercept Parameter**

i. Intercept Value: Enter an intercept value

ii. Click the '**Apply**' option.

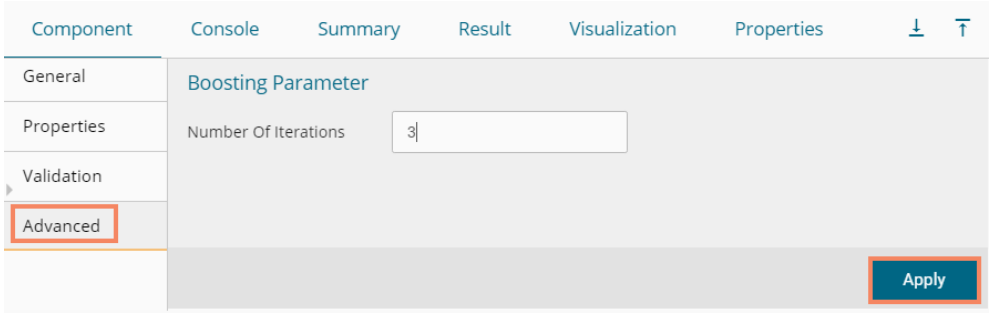


Advanced Tab when XG Boosting and Validation is enabled or XG Boosting is enabled, but Validation is disabled

a. **Boosting Parameter**

i. Number of Iterations: Enter the number of iterations.

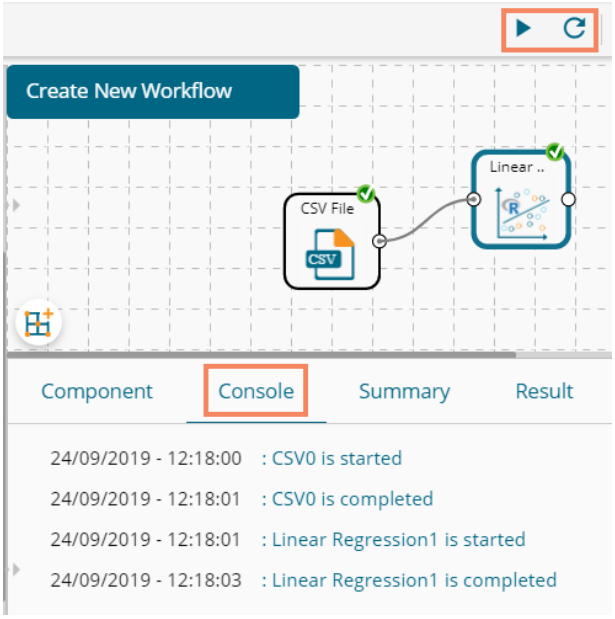
ii. Click the **'Apply'** option.



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

v) Run the workflow after getting the success message.

vi) The '**Console**' tab opens, displaying the process. The completion of the console process gets marked by the green checkmarks at the top of the dragged components.



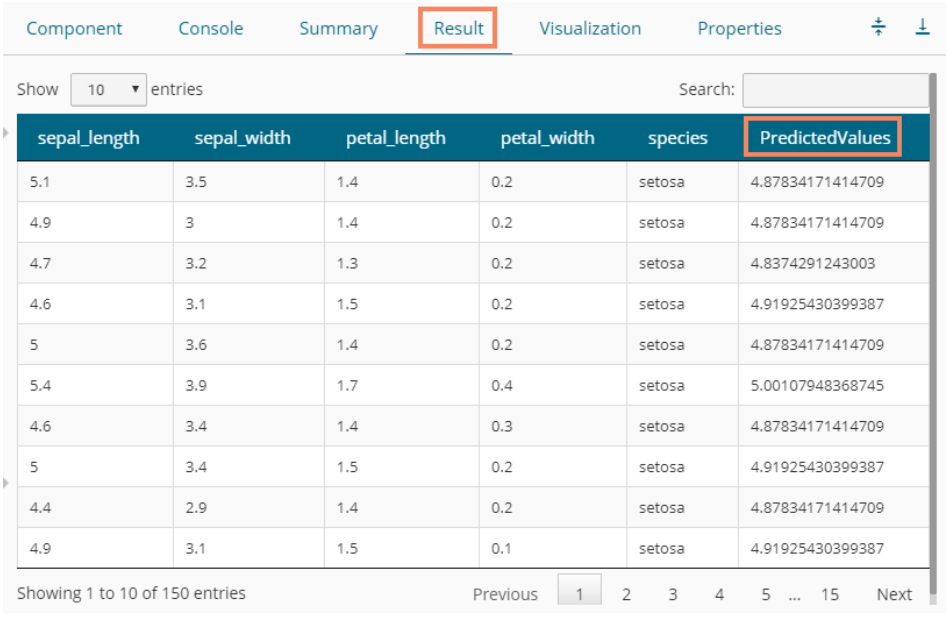
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

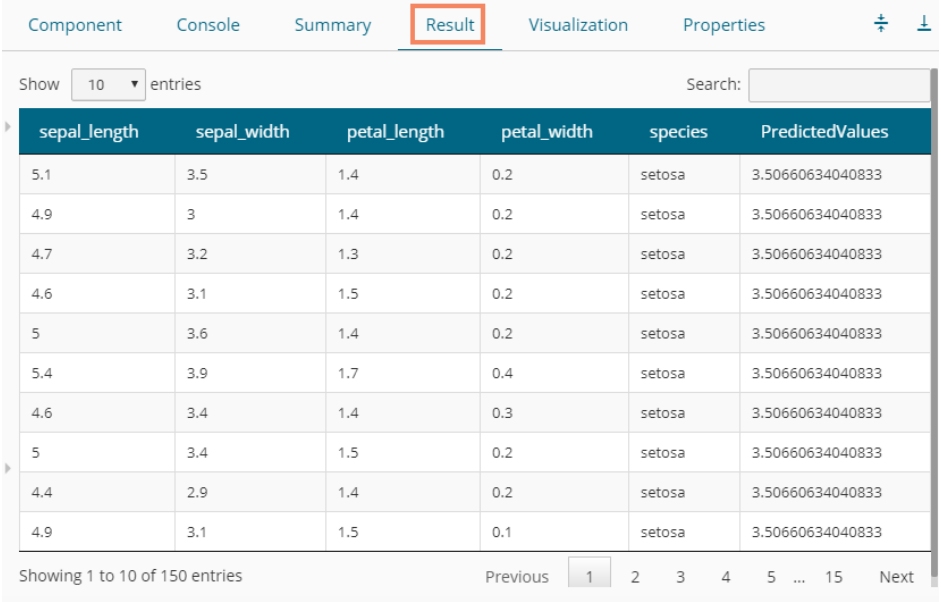
b. Click the '**Result**' tab.

i. A new column '**Predicted Values1**' gets added to the Result data displaying the predicted values.

Result when Validation and XG Boosting are disabled.

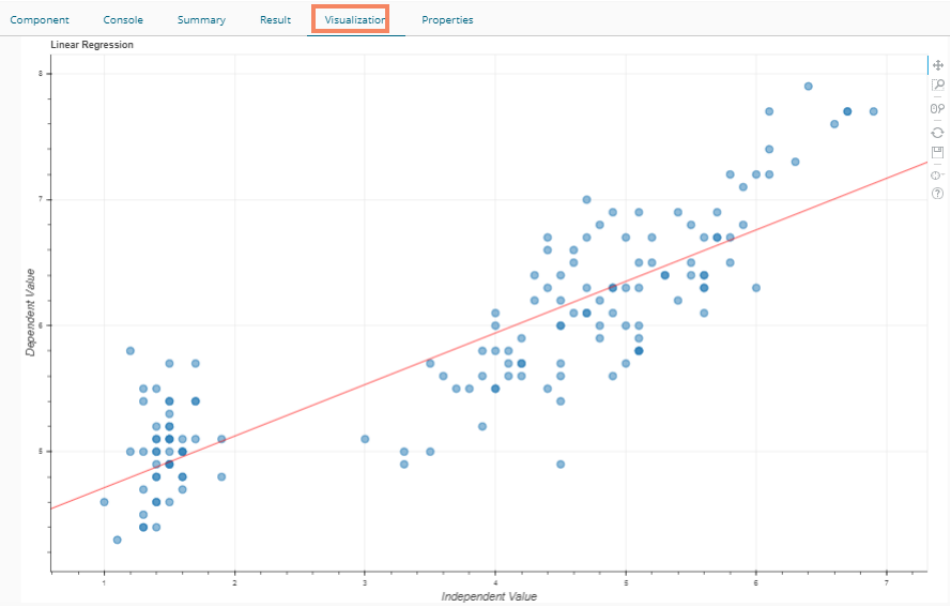


Result when XG Boosting enabled, and Validation enabled or disabled (No visualization is available for this situation).



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

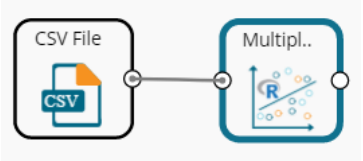
ix) The Result data gets displayed via the Scatter Plot with Regression line chart.



Note: '**Behavior**' fields provided under the '**Advanced**' section differs as per the algorithm sub-type. '**Input Data Handling**' remains the same for all the provided Regression types. Hence, only the '**Advanced**' tab is explained below for the remaining R sub-algorithms provided under '**Regression.**'

13.1.4.2. Multiple Linear Regression

i) Drag the Multiple Linear Regression component to the workspace and connect it with a configured data source.



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

a. Column Selection

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

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

b. New Column Information

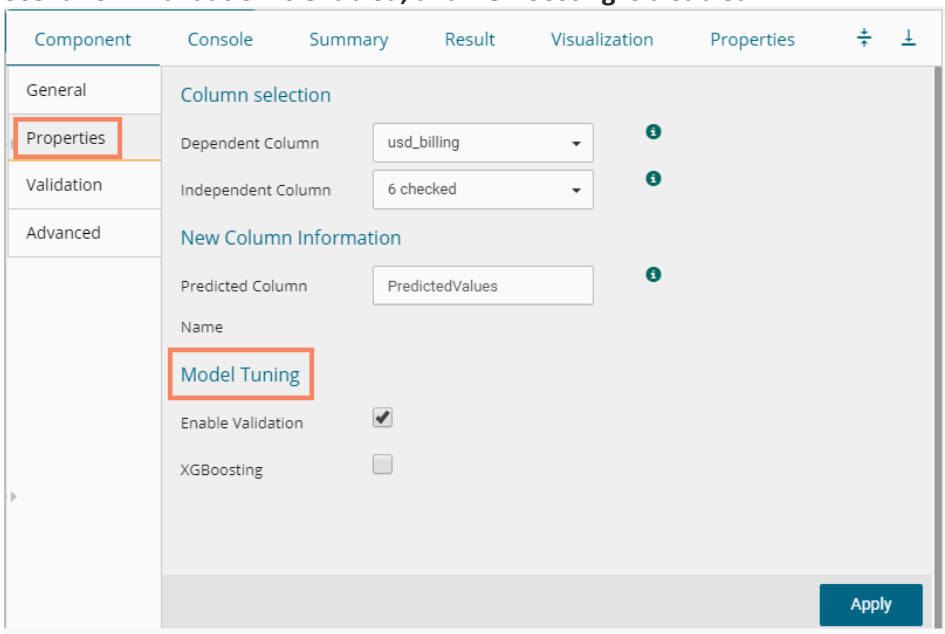
i. **Predicted Column Name**: Enter a name for the new column containing the predicted values

c. Model Tuning

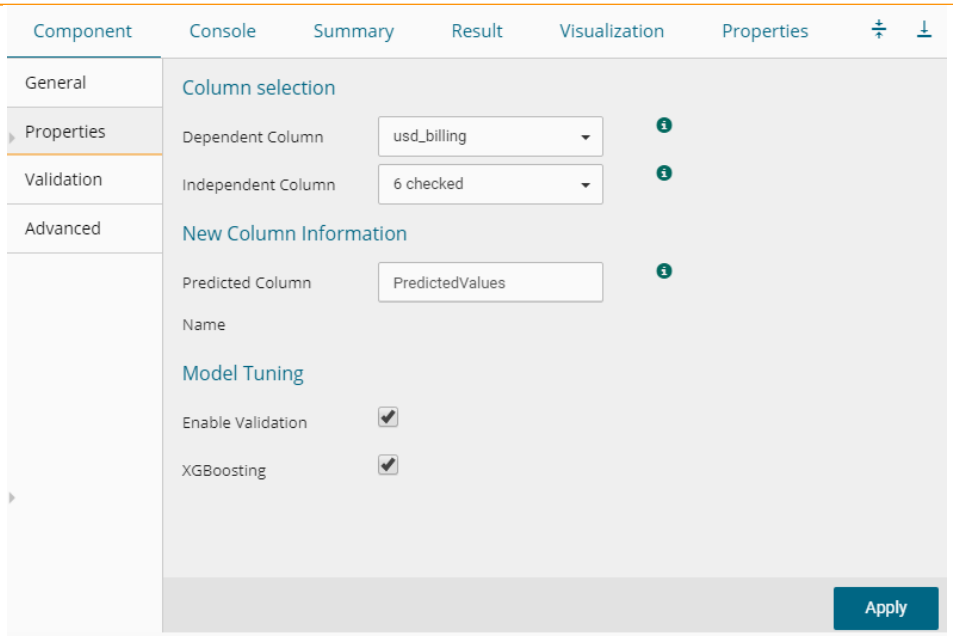
i. Enable Validation: Use a checkmark to enable the validation tab

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

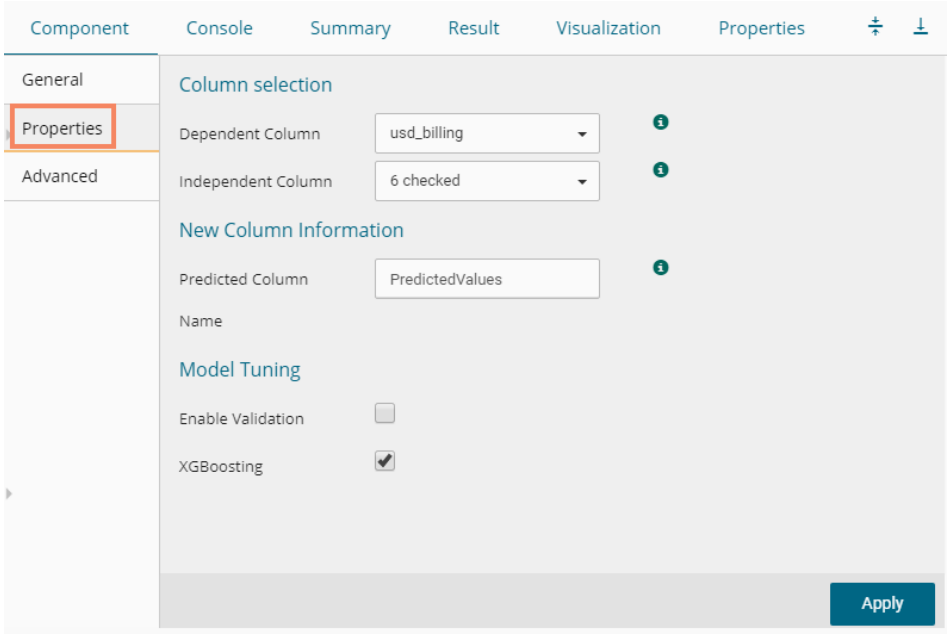
Scenario 1: Validation is enabled, and XG Boosting is disabled



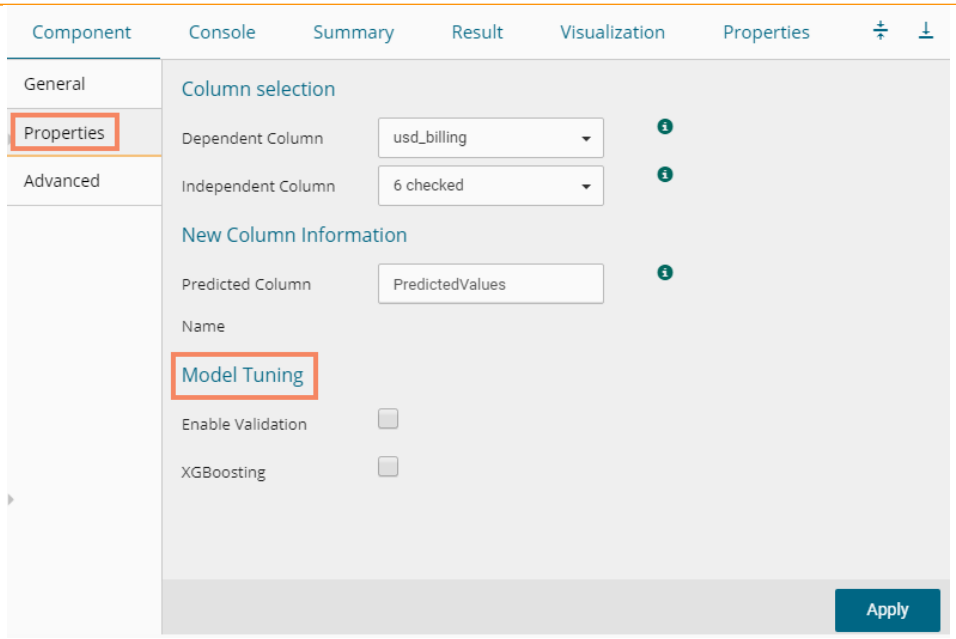
Scenario 2: Validation and XG Boosting are enabled



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



Scenario 4: When Validation and XG Boosting are disabled.

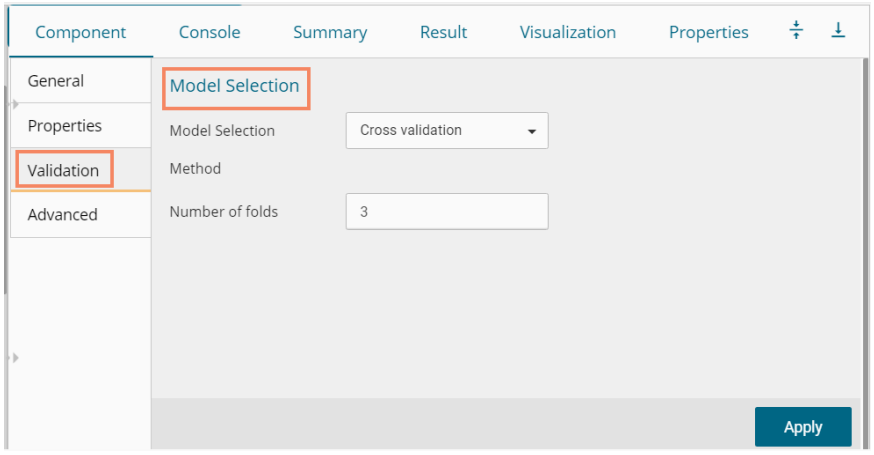


iii) **Validation**

a. Validation Model Selection when XG Boosting is disabled

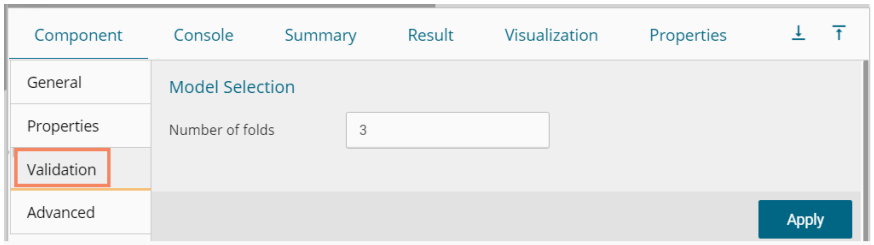
i. **Model Selection Method:** Select a model selection method using the drop-down menu.

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



b. Validation Model Selection when XG Boosting is enabled

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



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

<u>When Validation and XG Boosting are disabled</u>

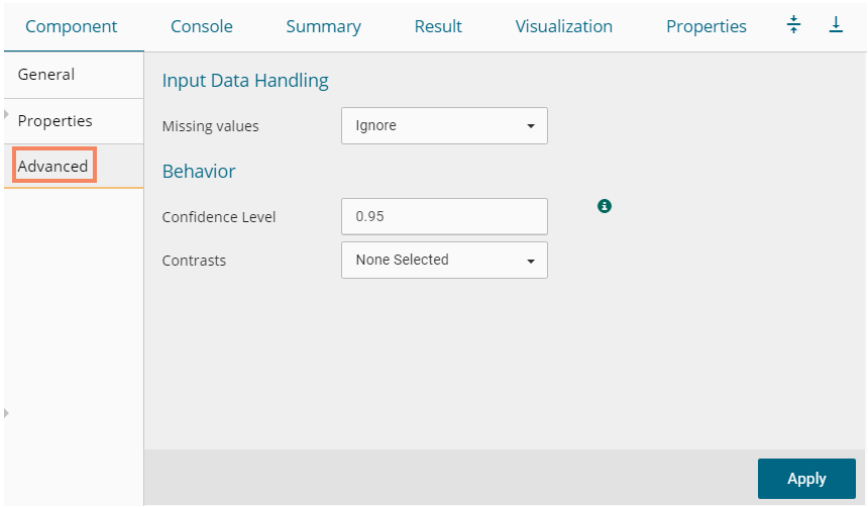
a. Input Data Handling

i. **Missing Values**: Select a method to deal with missing values (via the drop-down menu).

1. **Ignore**: Select this option to skip the records containing missing values from the dependent and independent columns.
2. **Keep**: Select this option to retain the records containing missing values while performing the calculation.
3. **Stop**: Select this option to stop the algorithm application if a value is missing in any column.

b. Behavior

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

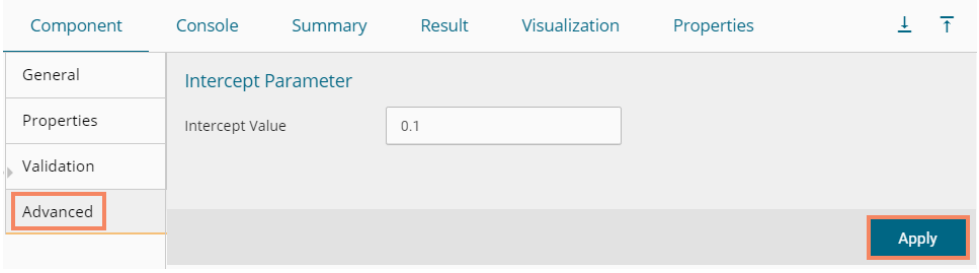


<u>When Validation is enabled and XG Boosting disabled</u>

a. Intercept Parameter

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

ii. Click the **'Apply'** option**.**

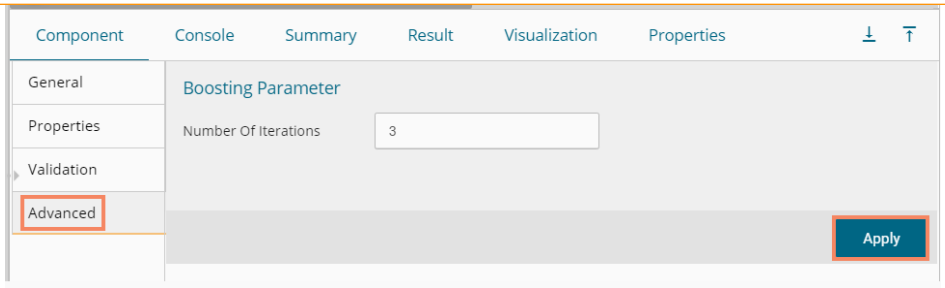


<u>When XG Boosting is enabled with either Validation is enabled or disabled</u>

a. Boosting Parameter

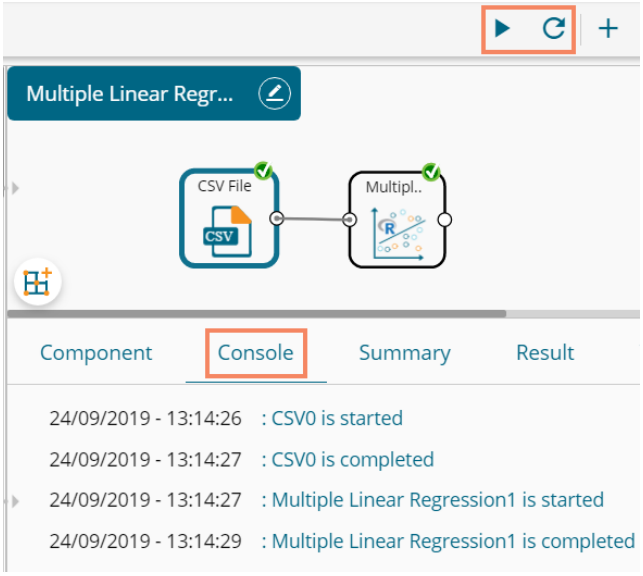
i. **No. of Iterations:** Enter number suggesting no. of iterations.

ii. Click the **'Apply'** option.



v) Run the workflow after getting the success message.

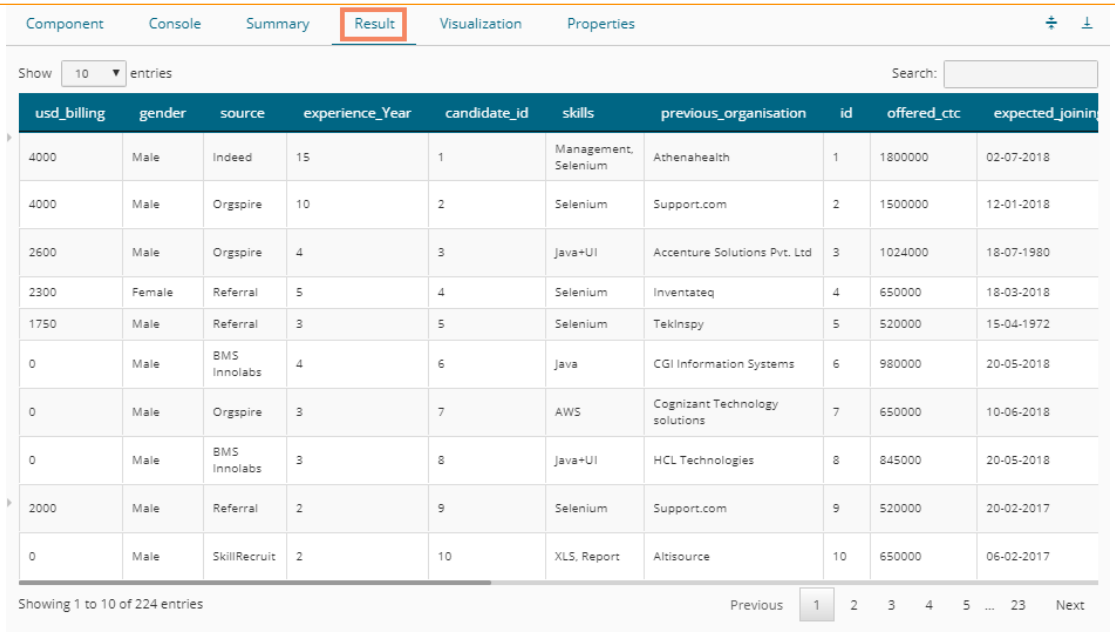
vi) The '**Console**' tab opens displaying the steps of the process. The completion of the console process is marked by the green checkmarks on the top of the dragged components.



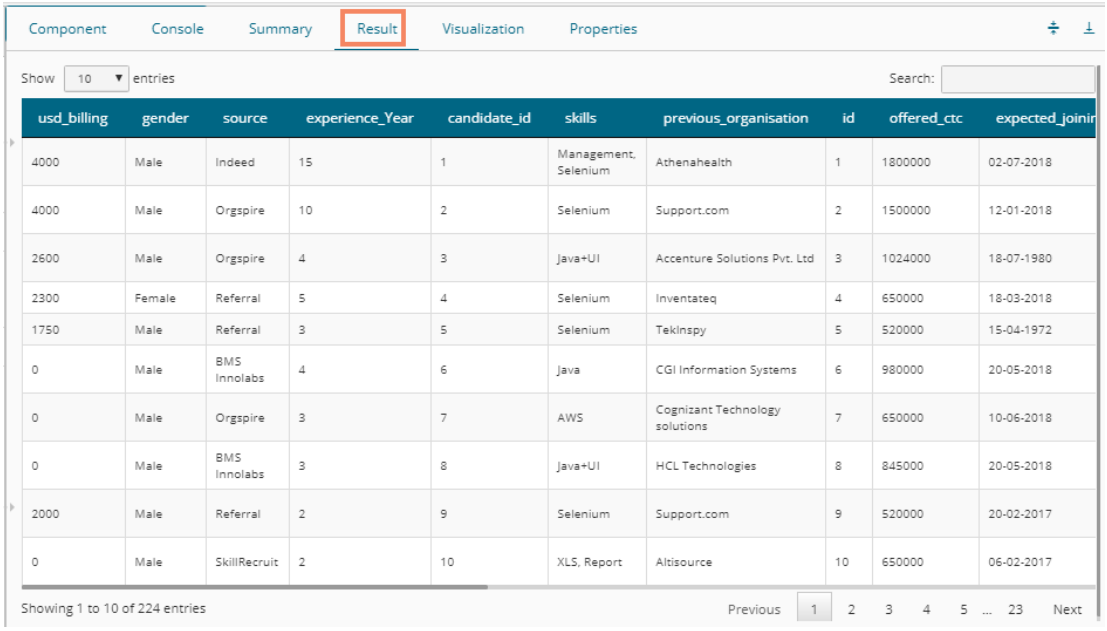
vii) The processed data gets displayed under the '**Result**' tab (a new column gets added to the result data) with the following possibilities:

viii) A new column is added to the Result data.

a. Result when XG Boosting is disabled.



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

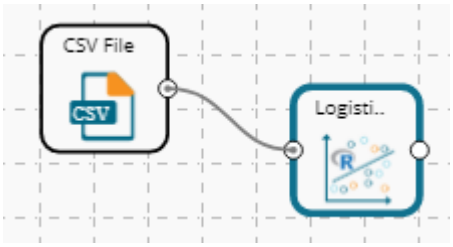


ix) Click the '**Visualization**' tab.

x) The Scatterplot with Regression Line Chart appears to display the Result data when the XG Boosting is disabled.

13.1.4.3. Logistic Regression

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



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

a. Column Selection

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

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

b. New Column Information

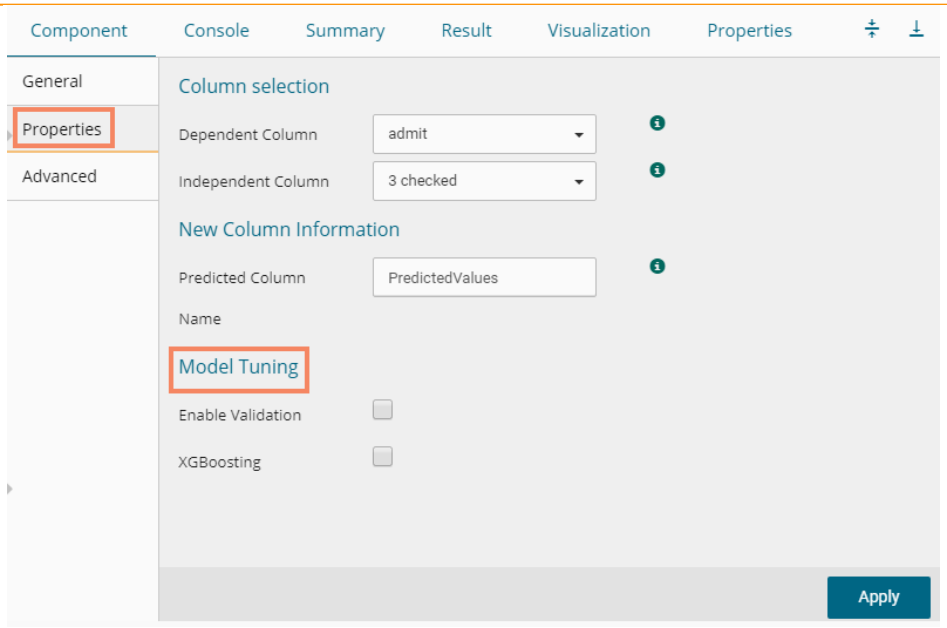
i. **Predicted Column Name**: Enter a name for the new column containing the predicted values

c. Model Tuning

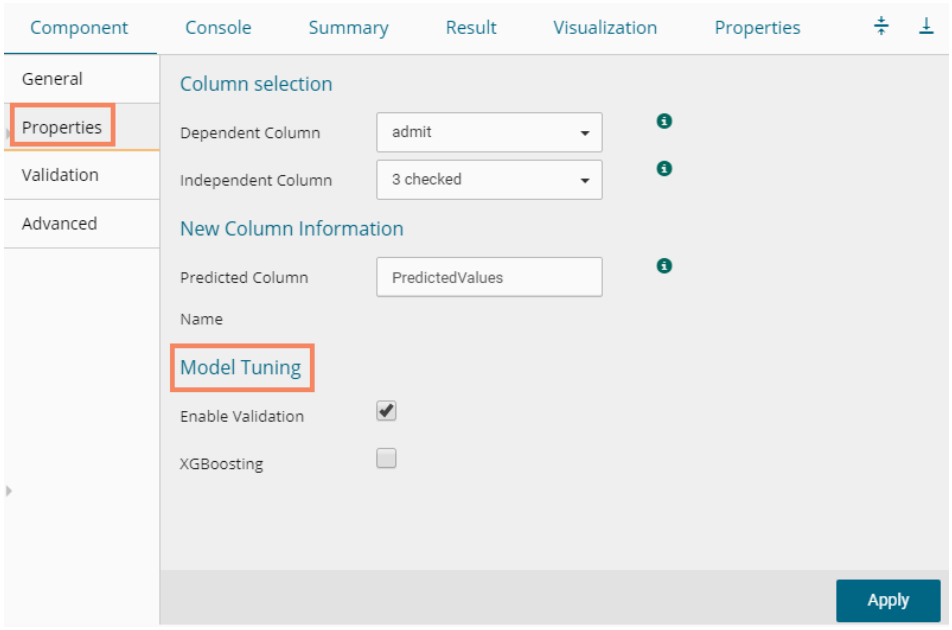
i. Enable Validation: Use a checkmark to enable the validation tab

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

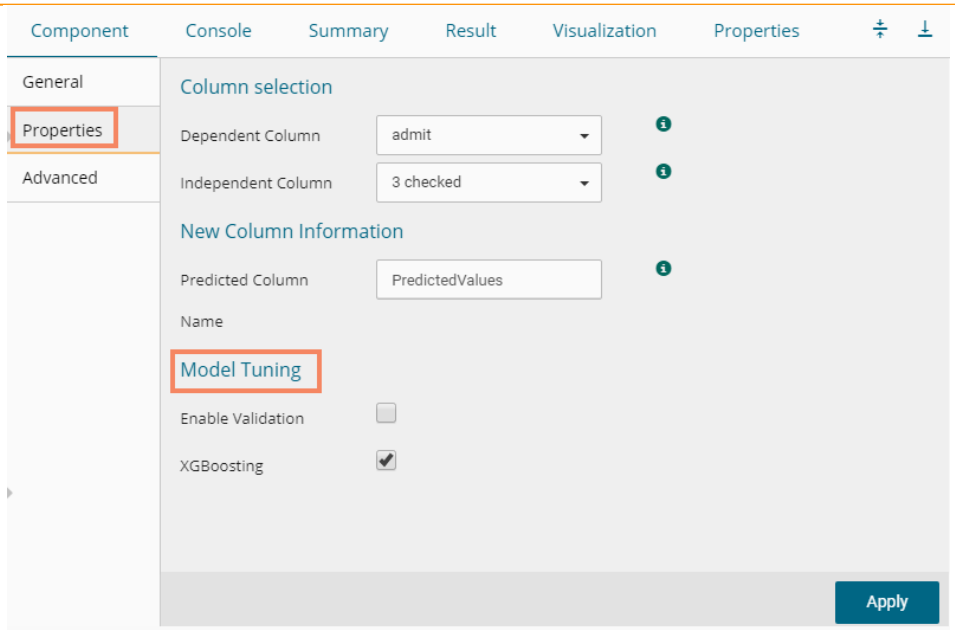
Scenario 1: XG Boosting and Validation are disabled.



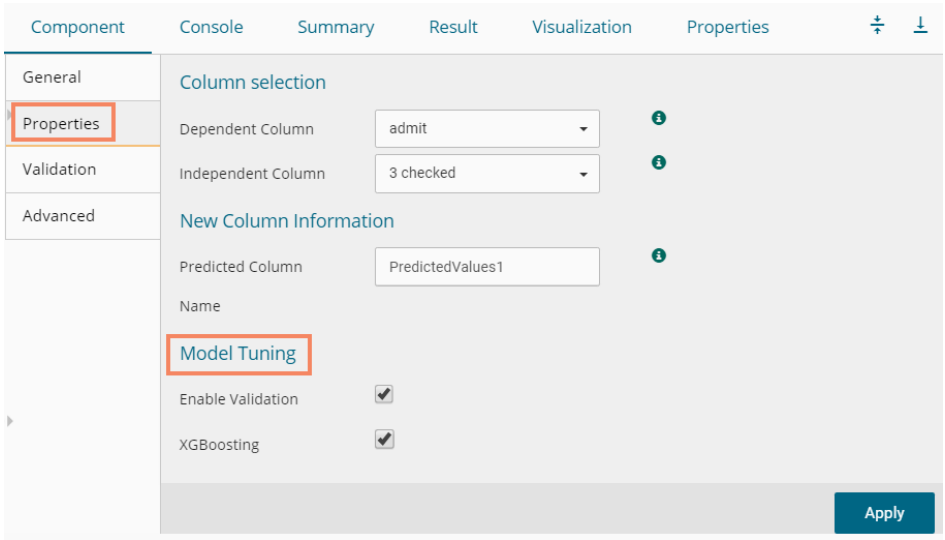
Scenario 2: When Validation is enabled, and XG Boosting is disabled.



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



Scenario 4: Validation and XG Boosting are enabled



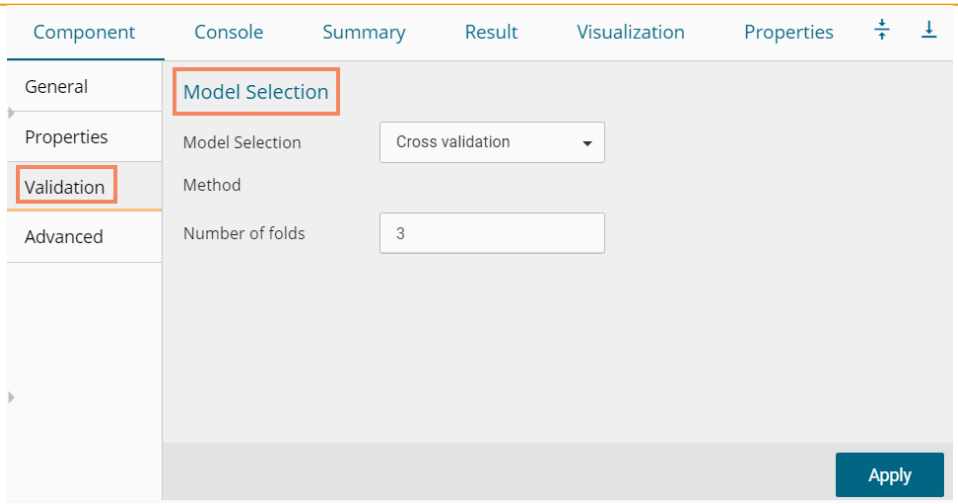
iii) **Validation Tab**

a. Validation tab when XG Boosting is disabled.

Model Selection

i. **Model Selection Method:** Select a model selection method from the drop-down menu.

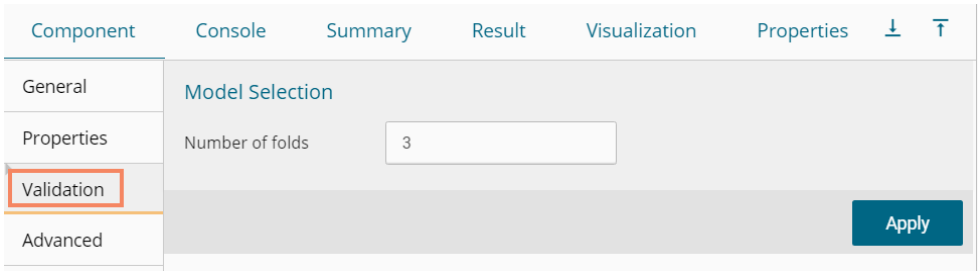
ii. **Number of folds:** Enter a value for the number of folds.



b. Validation tab when XG Boosting is enabled

Model Selection

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



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

Advanced Tab when Validation and XG Boosting are disabled

a. Input Data Handling

i. **Missing Values**

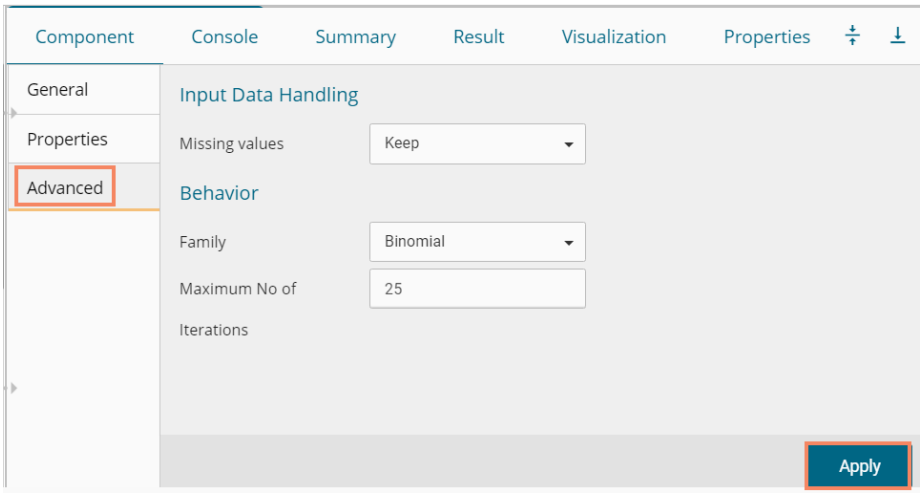
1. **Ignore**: Selecting this option will skip the records containing missing values in the columns
2. **Keep:** Select this option to retain the records containing missing values while performing the calculation
3. **Stop:** Select this option to stop (not allow) the records containing missing values while performing the calculation

b. Behavior

i. **Family:** Select an option from the drop-down list

1. Binomial
2. Poisson
3. Gaussian
4. Gamma
5. Quasi
6. Quasi-Poisson
7. Quasibinomial

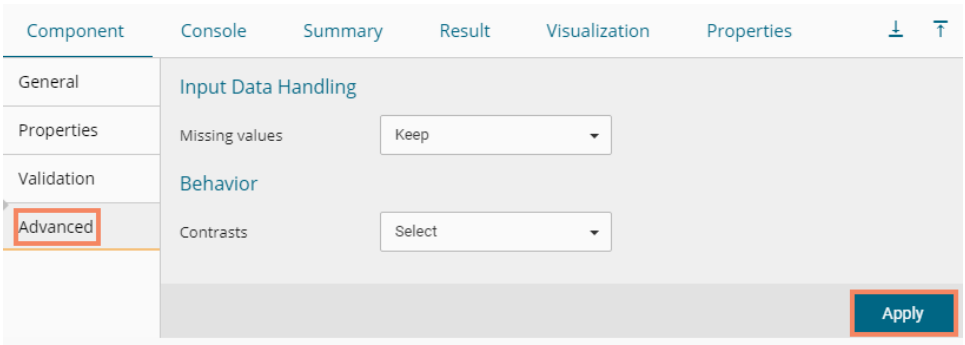
ii. **Maximum No. of Iterations:** Enter a valid integer value allowed to calculate the algorithm coefficient. The default value for this field is 25.



Advanced Tab with Validation enabled and XG Boosting disabled

a. **Input Data Handling**
   i. **Missing Values:**
      1. **Ignore**: Select this option to skip the records containing missing values in the columns
      2. **Keep:** Select this option to retain the records containing missing values while performing the calculation
      3. **Stop:** Select this option to stop (not allow) the records containing missing values while performing the calculation

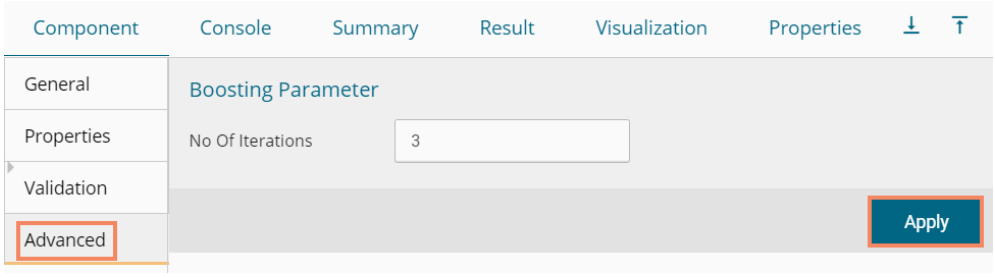
b. **Behavior**
   i. **Contrast:** Select an option from the following list
      1. None Selected
      2. Contr.treatment
      3. Contr.poly
      4. Contr.sum
      5. Contr.helmert



Advanced tab when XG Boosting is enabled and Validation is enabled or disabled

a. Boosting Parameter

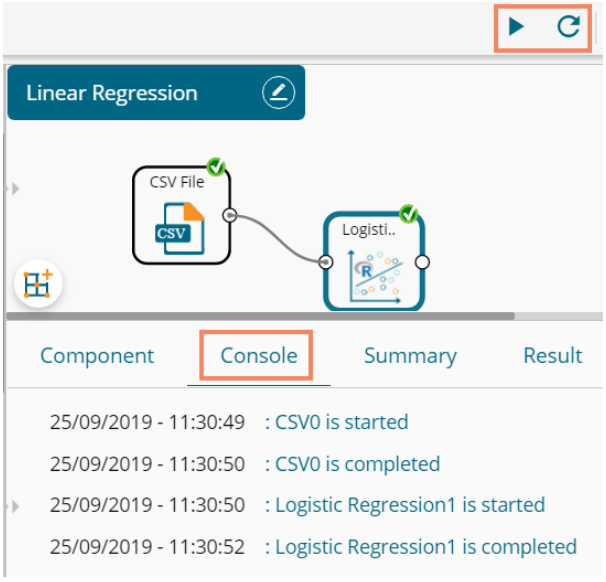
i. **No. of Iterations:** Enter a number suggesting no. of Iterations



v) Click the '**Apply'** option.

vi) Run the workflow.

vii) The '**Console**' tab opens, displaying the stepwise process. The completion of the console process gets marked by the green checkmarks at the top of the dragged components.



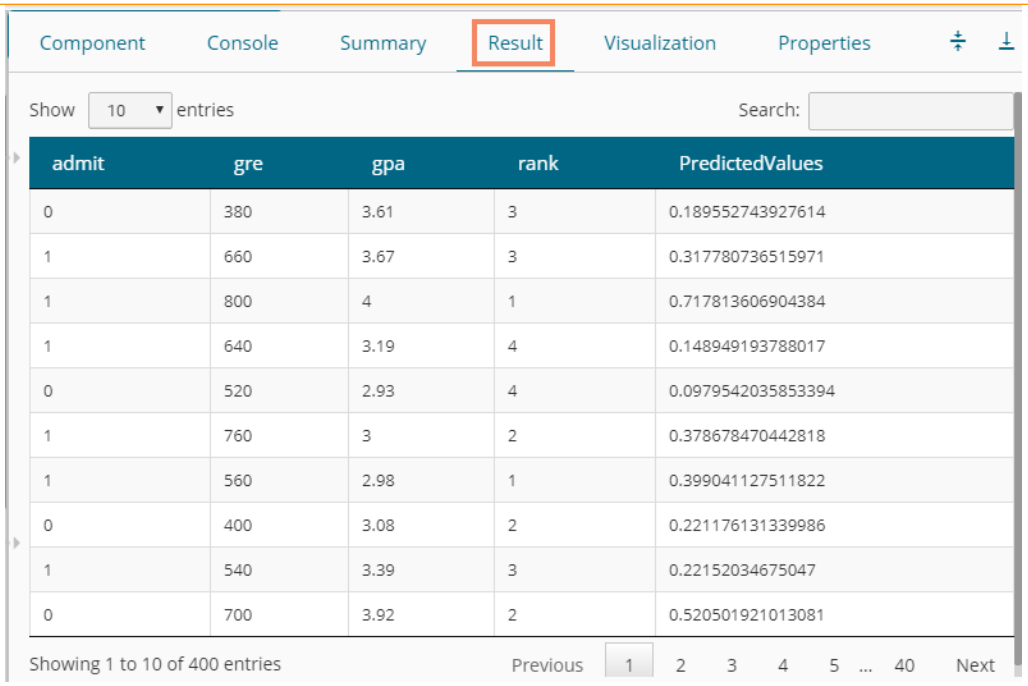
viii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

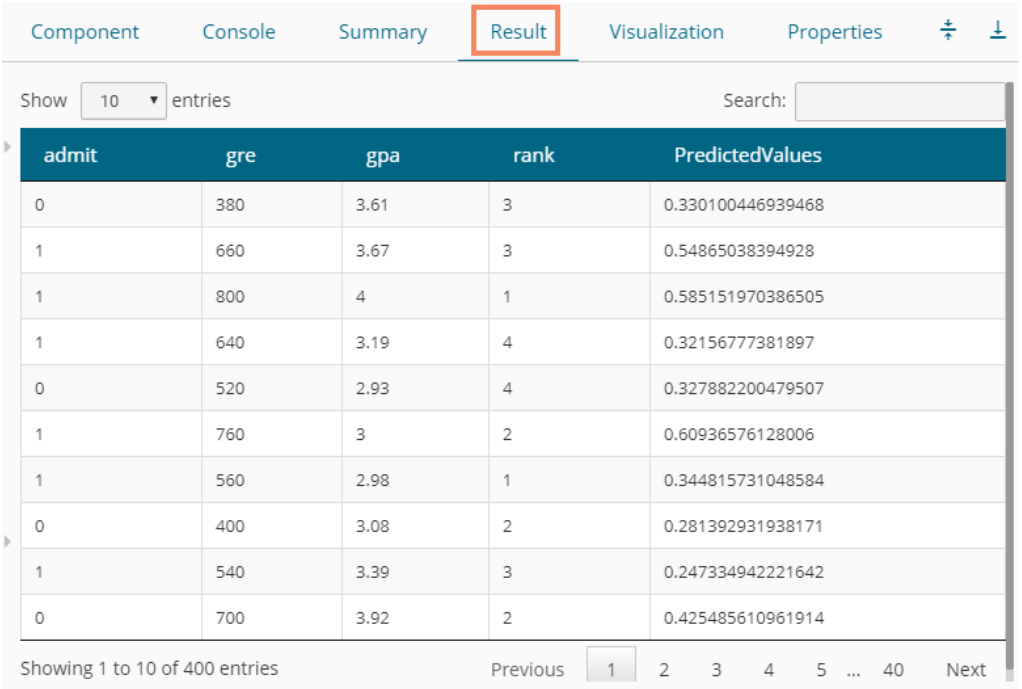
b. Click the '**Result**' tab.

ix) A new column is inserted into the Result Data.

Result when XG Boosting is disabled

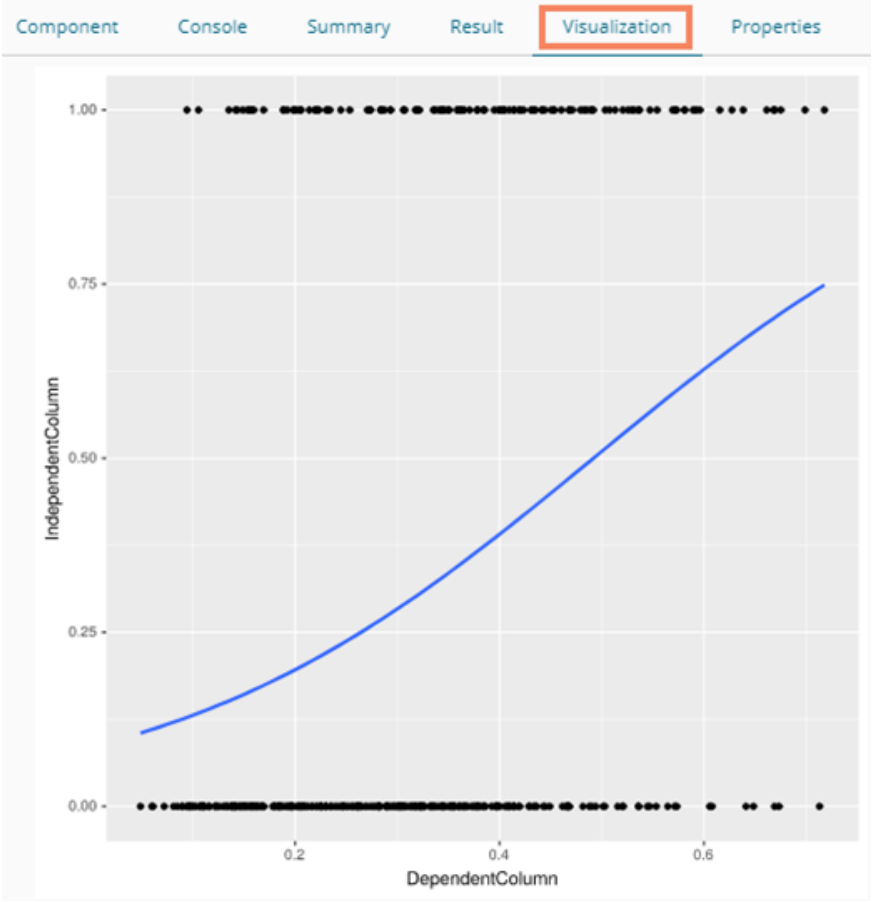


Result when XG Boosting is enabled



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

xi) The Result data gets displayed via the chart displaying the Scatter Plot with a Regression Line.

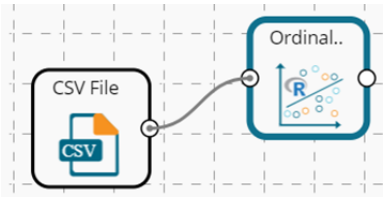


Note: No Visualization is available for the models in which XG Boosting is enabled.

13.1.4.1. Ordinal Regression

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

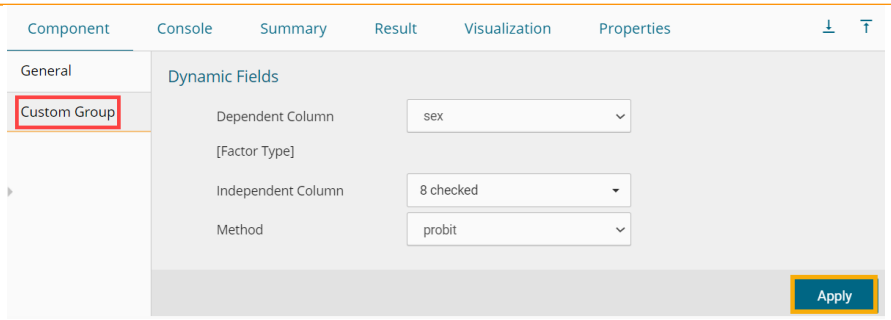
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

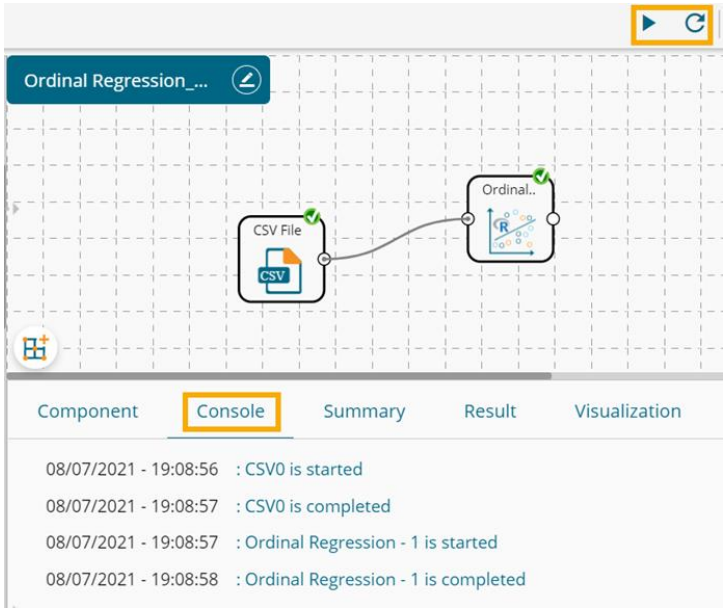
- **a.** **Dependent Column**: Select the target column on which the regression analysis gets applied.
- **b.** **Independent Column**: Select the required input columns against which the analysis gets applied to the target column.
- **c.** **Method**: Select a method from the given choices that include Probit, loglog, and cloglog

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

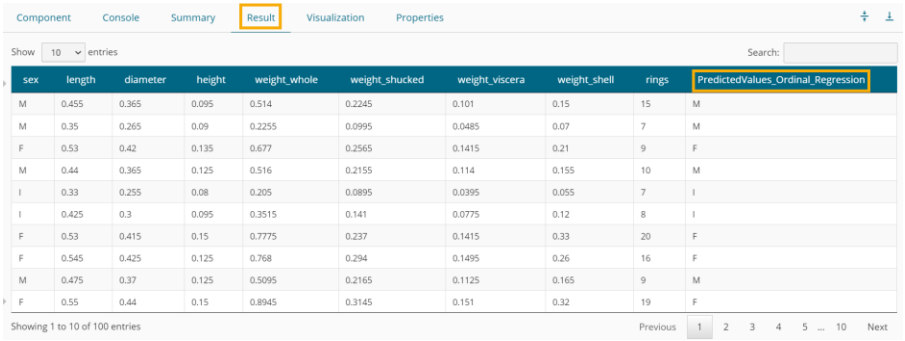
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



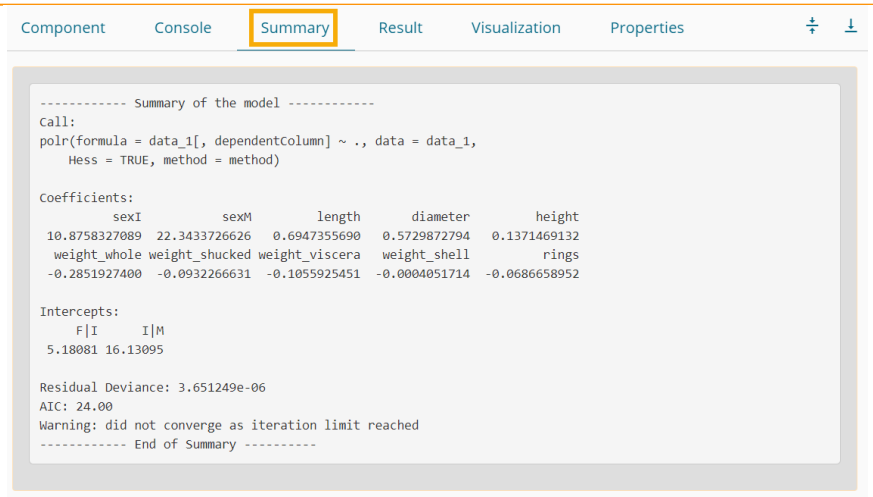
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



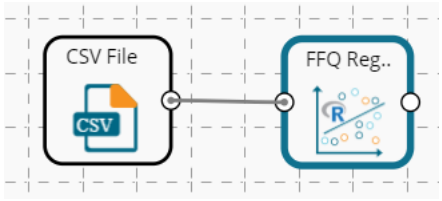
viii) Click the '**Summary**' tab to open the model summary.



13.1.4.2. FFQ Regression

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

ii) Click the algorithm component.



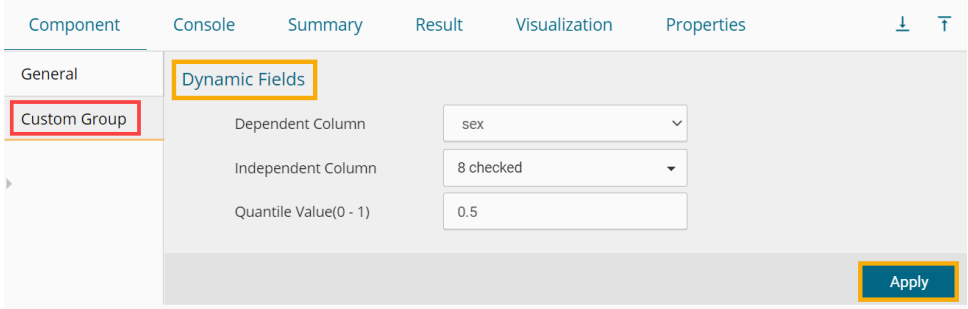
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

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

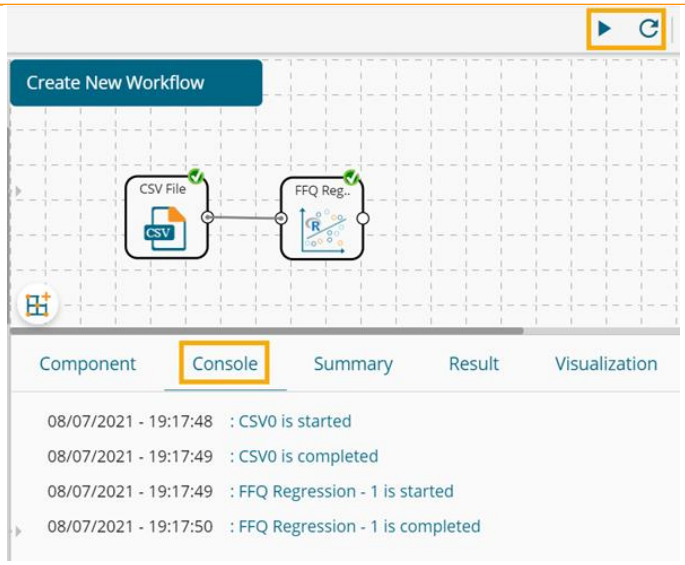
c. **Quantile Value (0-1)**: Select a number between 0 to 1 to configure the Quantile Value.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

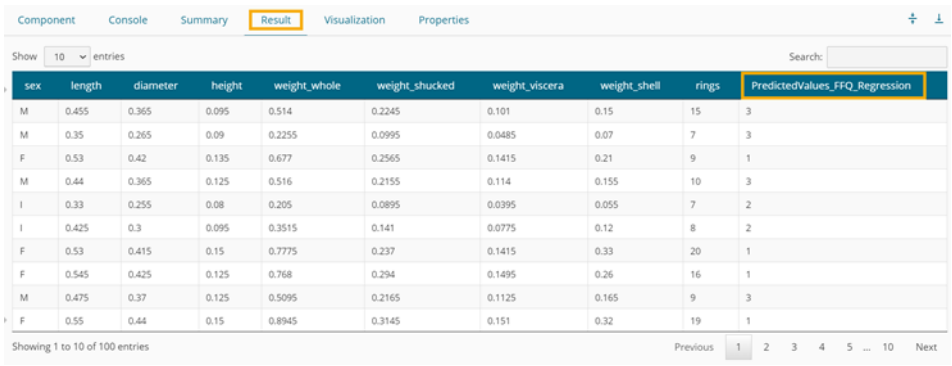
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



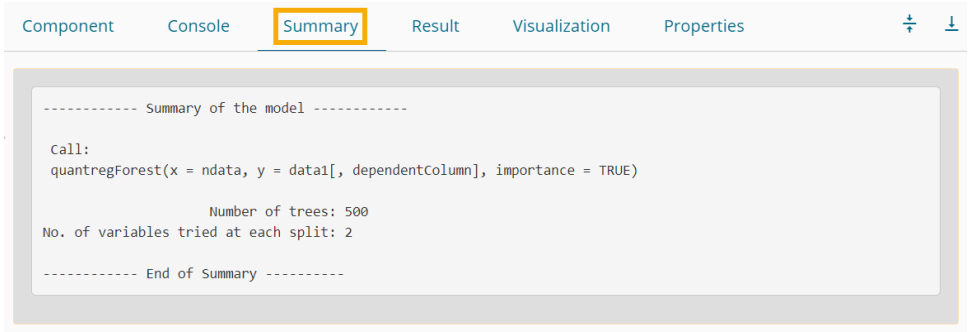
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

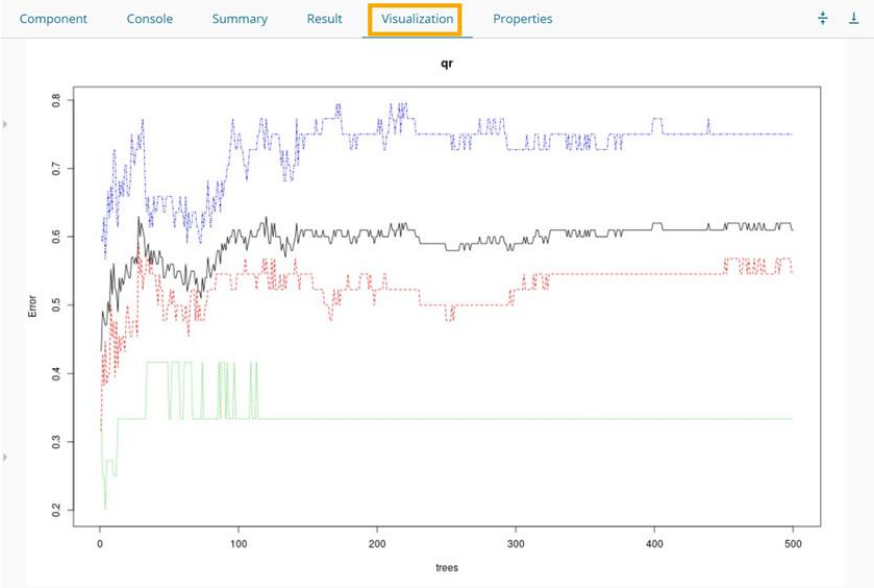
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



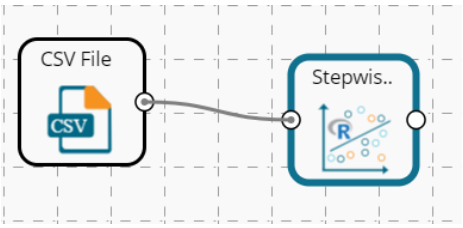
ix) Click the '**Visualization**' tab to see the processed data through a chart.



13.1.4.3. Stepwise Regression

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

ii) Click the algorithm component.

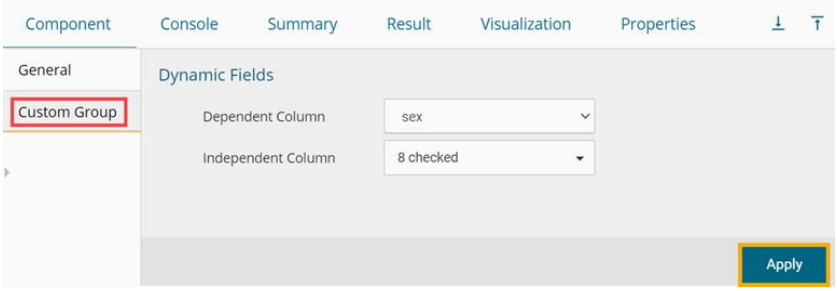


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

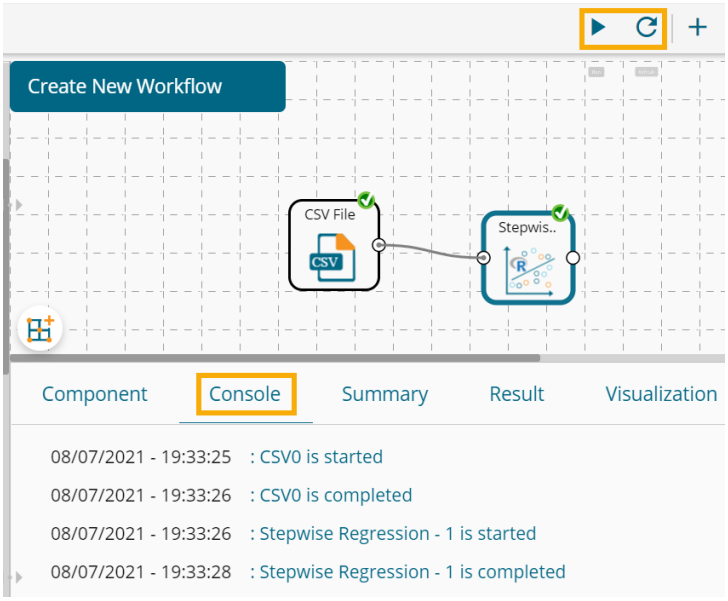
b. **Independent Column**: Select the required input columns against which the analysis gets applied to the target column.

iv) Click the '**Apply**' option.



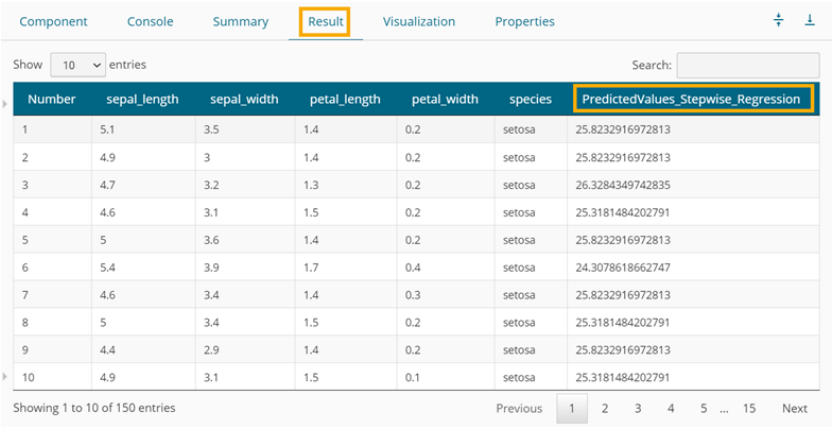
v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

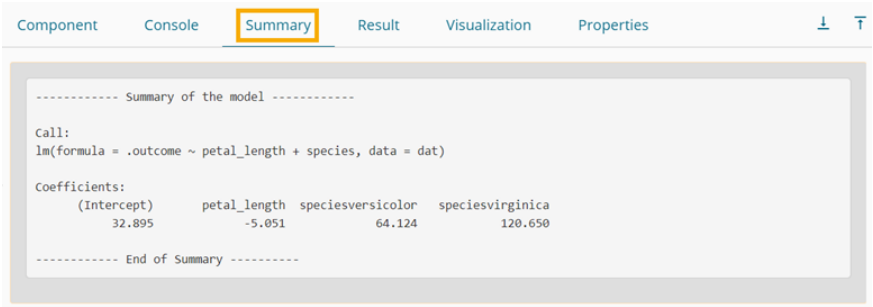


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



13.1.5. Classification

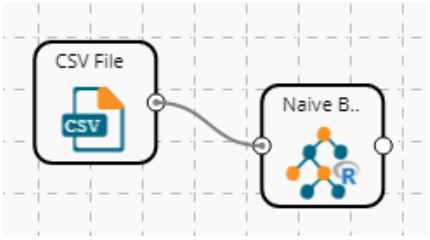
This algorithm categorizes a new observation by a trained set of data that contains observations from the known category. It compares each new observation to previous observations using means of similarity or distance.

13.1.5.1. 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**.

Naive Bayes is a leaf node under Classification algorithms under the Algorithm tree node. The component consists of one node for reading data from a data source and another one for giving the Result.

i) Drag the 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 to performing the analysis.

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

b. Output Information

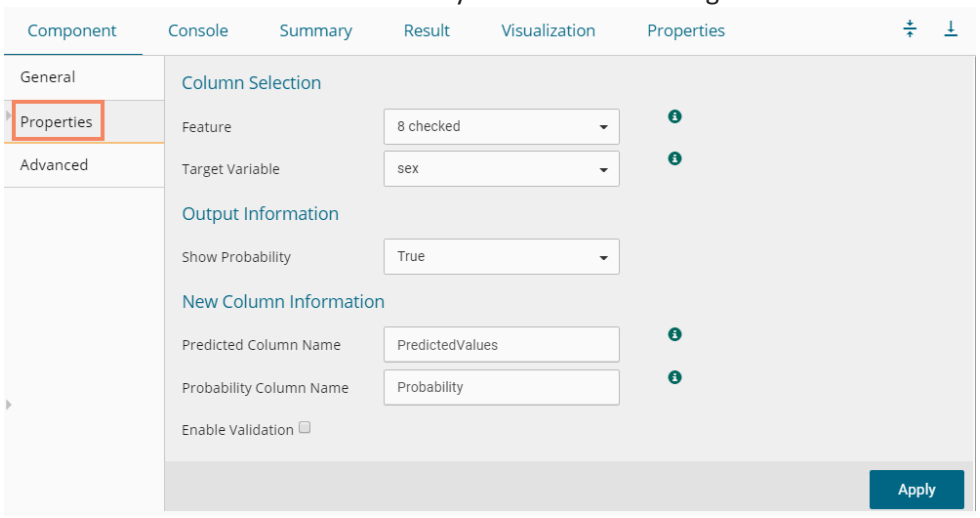
i. **Show Probability:** Select an option out of True or False (Selecting the 'True' option displays the Probability Column Name field under the 'New Column Information' section).

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 checkmark in the given box.



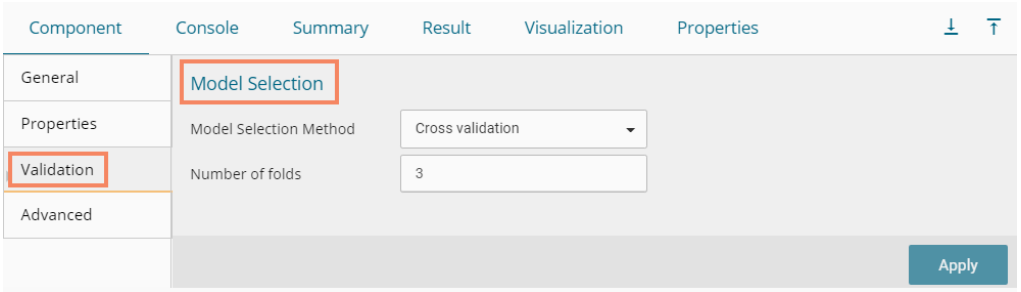
iii) Click the '**Validation**' tab and configure it, if it has been enabled from the Properties tab

a. Model Selection

i. Model Selection Method: Select a modeling method using the drop-down menu.

1. Cross-Validation
2. BootStrap
3. Repeated Cross-Validation
4. Leave One Out Cross-Validation

ii. The number of folds: Enter a numerical value for the number of folds.



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

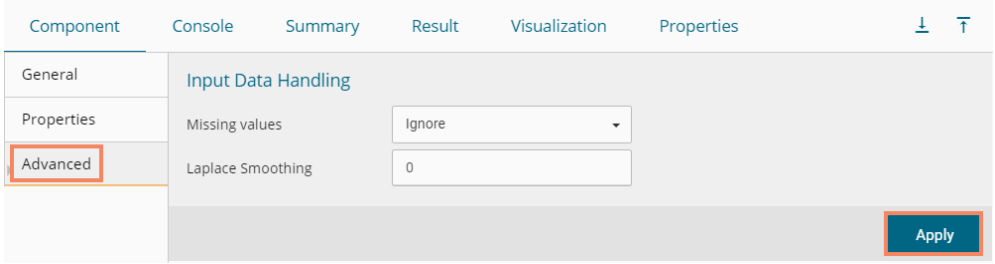
- **Advanced Tab when 'Validation' is Disabled:**

a. Input Data Handling

i. **Missing Values:** Select a method to deal with missing values from the drop-down menu.

1. **Ignore**: Selecting this option will skip the records containing missing values in the columns.
2. **Keep:** Selecting this option will retain the records containing missing values while performing the calculation.

ii. **Laplace Smoothing:** Enter the smoothing constant for smoothing observations. The smoothing constant must be a double value greater than 0. Entering 0 will disable Laplace smoothing.



- **Advanced Tab when 'Validation' is Enabled:**

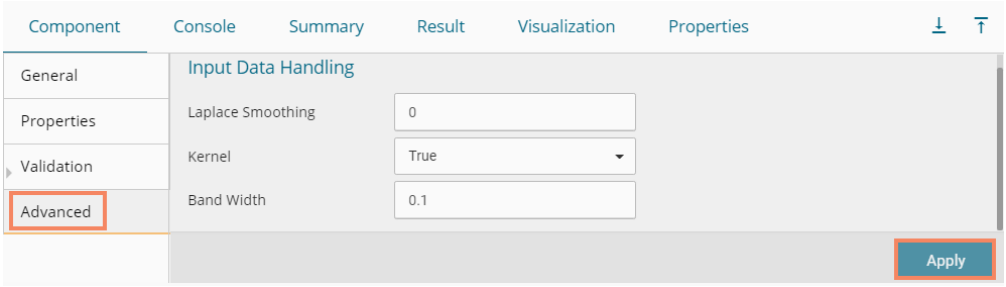
a. Input Data Handling

i. **Laplace Smoothing:** Enter the smoothing constant for smoothing observations. The smoothing constant must be a double value greater than 0. Entering 0 disables Laplace smoothing.

ii. **Kernel:** Select an option using the drop-down menu.
   1. **True**
   2. **False**

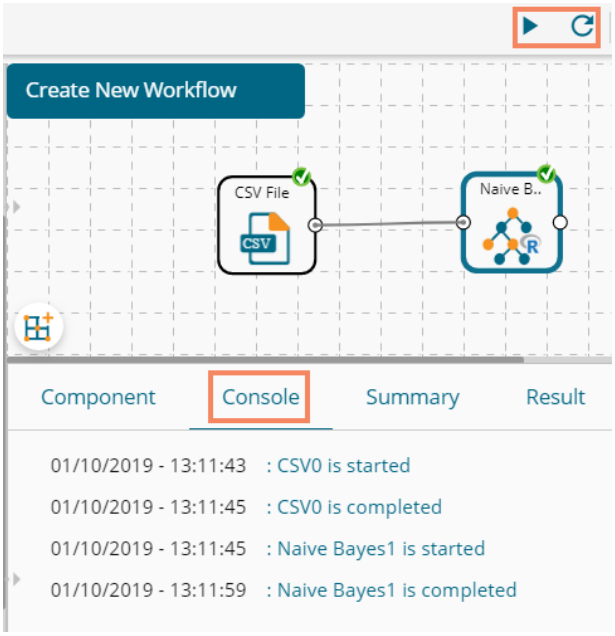
iii. **Band Width:** Enter a bandwidth value (the Default value for this field is 0.1).

iv. Click the **'Apply'** option.



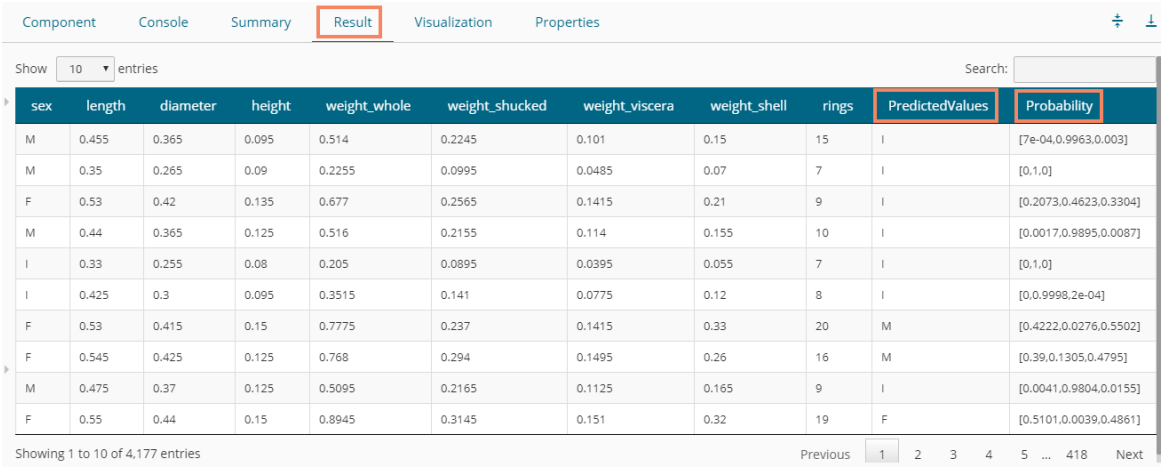
v) Run the workflow and after getting the success message.

vi) The '**Console**' tab opens displaying the steps of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Click the '**Result**' tab to display the dataset in the result view.

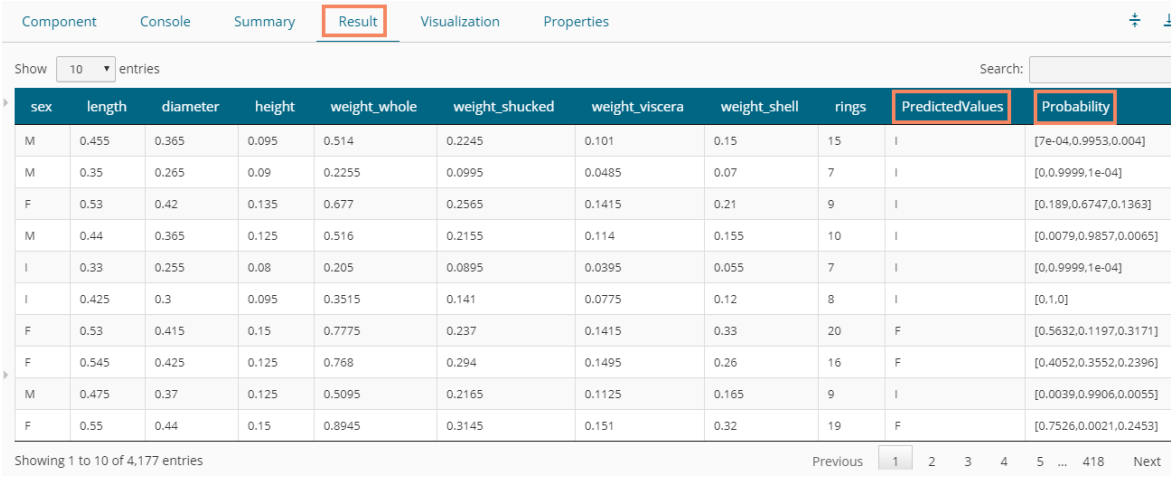
i. **Result View when Validation is disabled.**



sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues	Probability
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	[7e-04,0.9963,0.003]
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	[0,1,0]
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	[0.2073,0.4623,0.3304]
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	[0.0017,0.9895,0.0087]
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	[0,1,0]
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	[0,0.9998,2e-04]
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	M	[0.4222,0.0276,0.5502]
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	M	[0.39,0.1305,0.4795]
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	[0.0041,0.9804,0.0155]
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F	[0.5101,0.0039,0.4861]

Showing 1 to 10 of 4,177 entries Previous 1 2 3 4 5 ... 418 Next

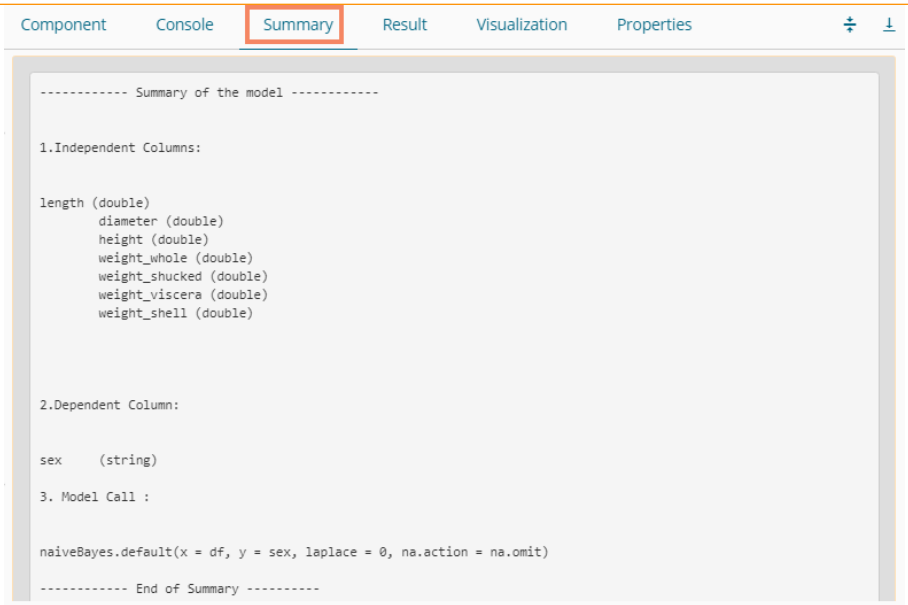
ii. **Result View when Validation is Enabled.**



sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues	Probability
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	I	[7e-04,0.9953,0.004]
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	I	[0,0.9999,1e-04]
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	I	[0.189,0.6747,0.1363]
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	I	[0.0079,0.9857,0.0065]
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	I	[0,0.9999,1e-04]
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	I	[0,1,0]
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	F	[0.5632,0.1197,0.3171]
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	F	[0.4052,0.3552,0.2396]
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	I	[0.0039,0.9906,0.0055]
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	F	[0.7526,0.0021,0.2453]

Showing 1 to 10 of 4,177 entries Previous 1 2 3 4 5 ... 418 Next

viii) Click the **'Summary'** tab to see the detailed Model Summary.



Note:

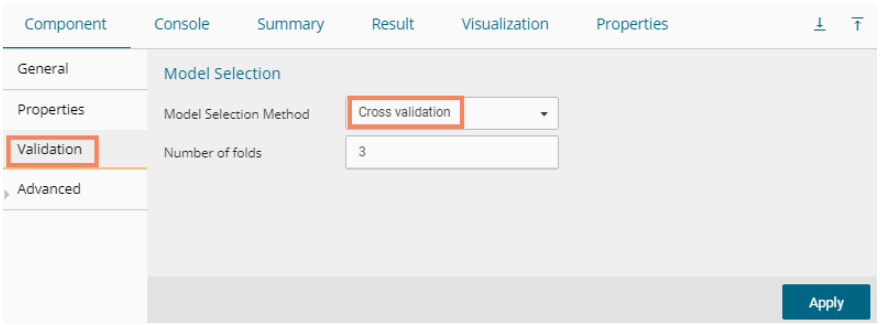
a. The '**Visualization**' tab does not display any graphical representation for the Naive Bayes Results in data.

b. The '**Validation**' tab provides multiple options under the '**Model Selection Method**' drop-down menu.

All the available Model Selection Methods are described below:

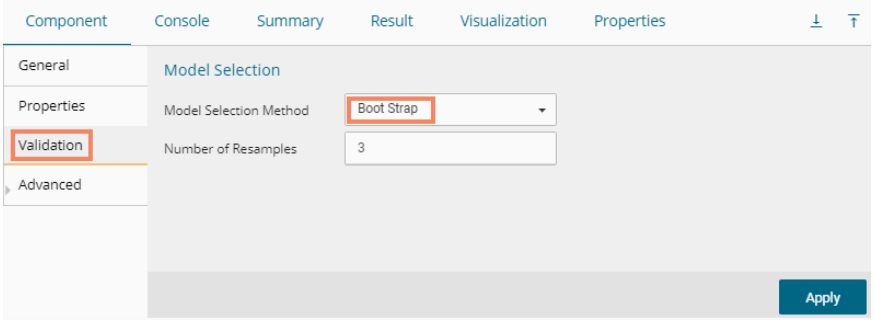
i. Cross-Validation

The user needs to configure the '**Number of folds**' if **Cross-Validation** is selected as the Model Selection Method.



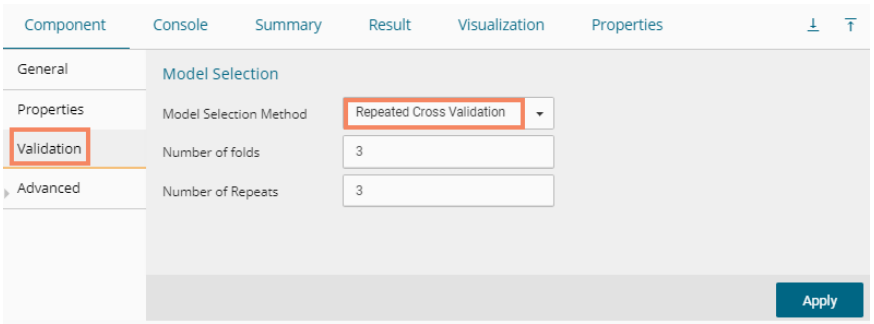
ii. Bootstrap

The user needs to configure the '**Number of resamples**' if '**Bootstrap**' is selected as the Model Selection Method.



iii. Repeated Cross-Validation

The user needs to configure the **Number of repeats,** and **the Number of folds** fields if the selected modeling method is **Repeated Cross-Validation.**



iv. Leave One Out Cross-Validation

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

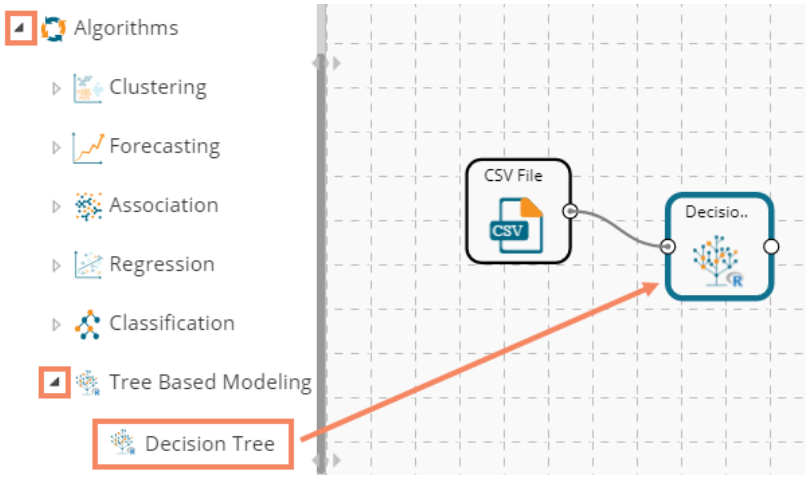
13.1.6. Tree-Based Modeling

The Tree-Based Modeling Decision Tree can be configured using two algorithm types from the '**Properties**' tab.

Check out the below-given description of the configuration details:

13.1.6.1. Classification as Algorithm Type for Decision Tree

i) Drag the Decision Tree component to the workspace and connect it with a configured data source.



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

a. Output Information

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

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

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

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

b. Column Selection

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

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

c. New Column Information

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

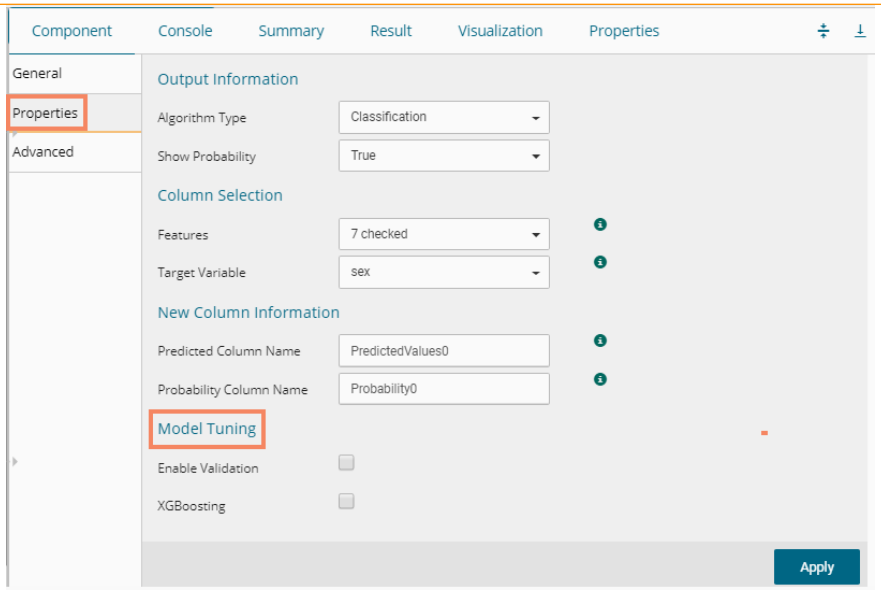
ii. **Probability Column Name:** Enter a name for the new column containing the probability values.

d. Model Tuning

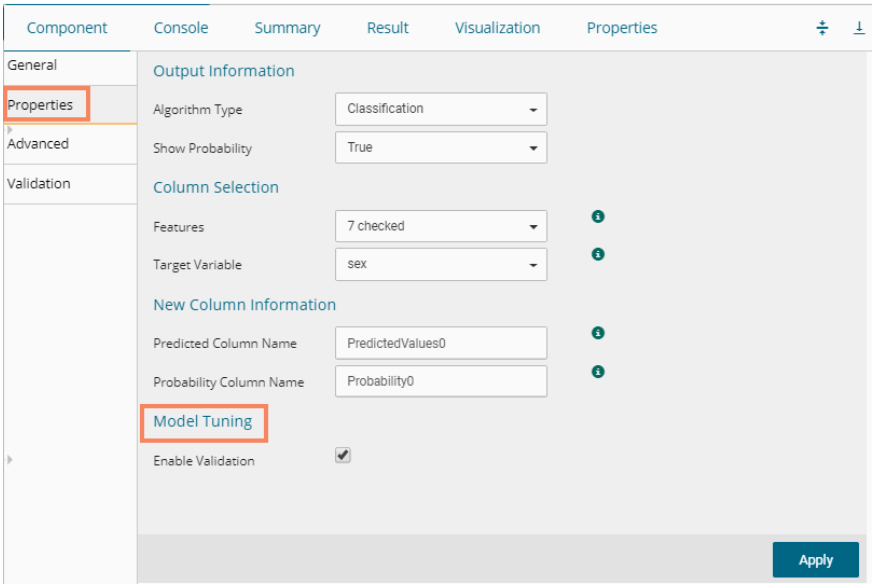
i. **Enable Validation:** Enable validation as a model tuning option by a checkmark in the given box.

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

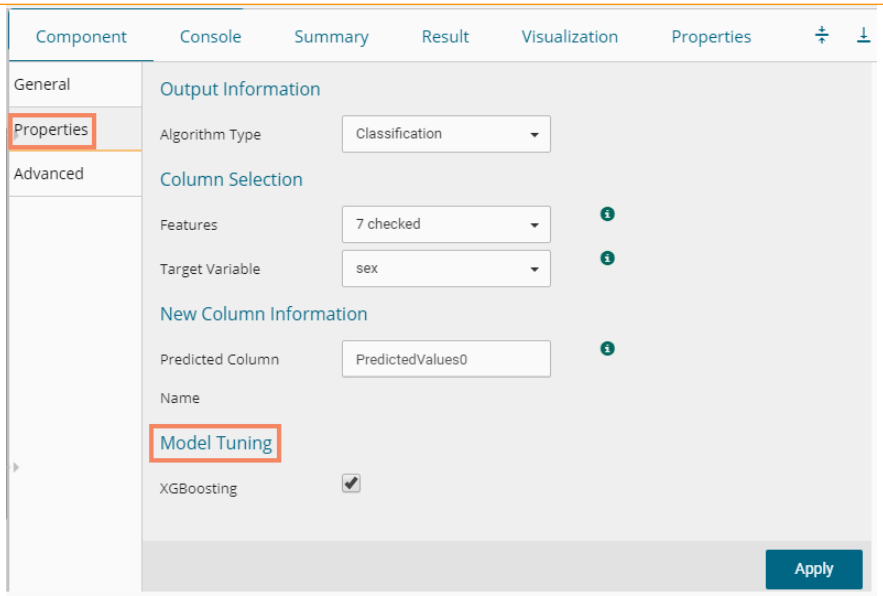
Properties Tab when Model Tunning is not Enabled



Properties Tab when Validation is Enabled as Model Tuning



Properties Tab when XG Boosting is Enabled as Model Tuning



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

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

- **Advanced Tab when both the Model Tuning options are Disabled**

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

  b. **Tree Pruning**
     i. **Minimum Split:** It indicates a minimum number of observations within a single node for a split to be attempted. The default value for this field is 10.
     ii. **Complexity Parameter:** This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is pruned off performing cross-validation, hence the program does not pursue it. The default value for this field is 0.05.
     iii. **Maximum Depth:** It sets the maximum depth of any node of the final tree keeping the depth count for root node 0. It is an optional field (It is recommended to set Maximum Depth value less than 30 rpart for 32 bit-machines.)

  c. **Behavior**

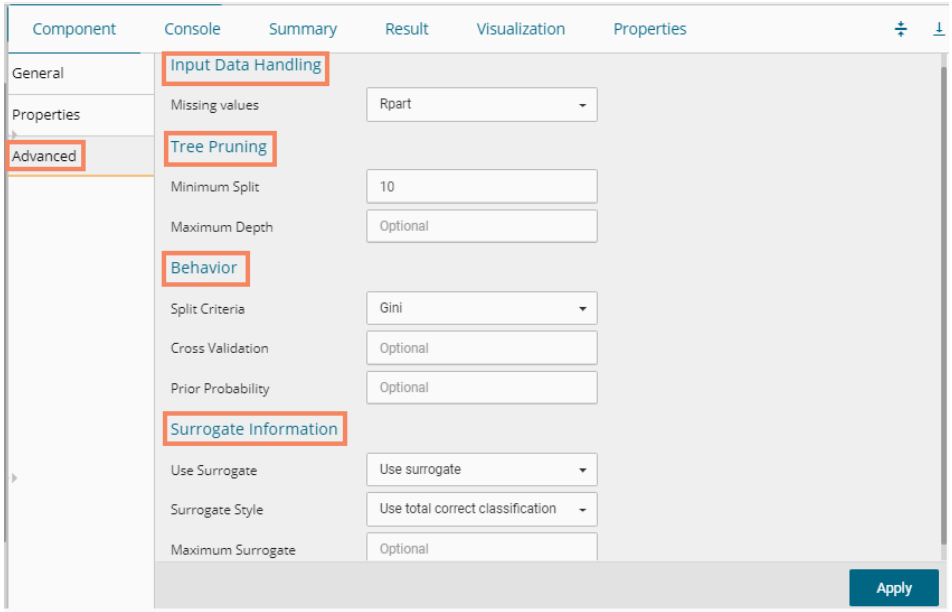
i. **Split Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'** tab. (This field appears only when the selected algorithm type is '**Classification**').
The splitting index can be:
1. **Gini:** Select this option to measure inequality among values of randomly chosen elements from a set.
2. **Information:** Select this option to get information about the variables used in the algorithm.

ii. **Cross-Validation:** It indicates the number of cross-validations that were performed to check the accuracy of the analysis method.

iii. **Prior Probability:** It is an optional field. This field is dependent on the other data values mentioned in the selected dataset. (This field appears when the selected algorithm type is '**Classification**').

d. Surrogate Information

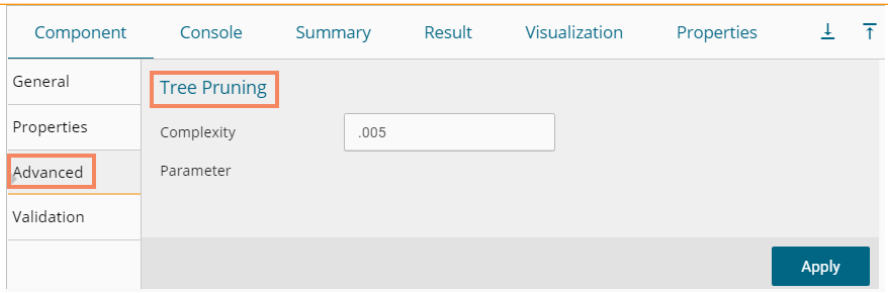
i. **Use Surrogate:** Select one option from the drop-down menu.
1. **Display Only:** Select this option to display only the observation, but not split it further.
2. **Use Surrogate:** Select this option to search surrogate values for the missing values to split the observation. Two fields are displayed:
   a. Surrogate Style: Select a style using the drop-down menu.
   b. Maximum Surrogate: Set the maximum surrogate value.
3. **Stop if missing:** Select this option to choose an action based on the nature of the majority observations. If values are missed for all the observations, then they will stop splitting further.



- **Advanced Tab when 'Validation' is enabled:**

a. Tree Pruning:

i. **Complexity Parameter**: This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is pruned off performing cross-validation, hence the program does not pursue it. The default value for this field is 0.05.

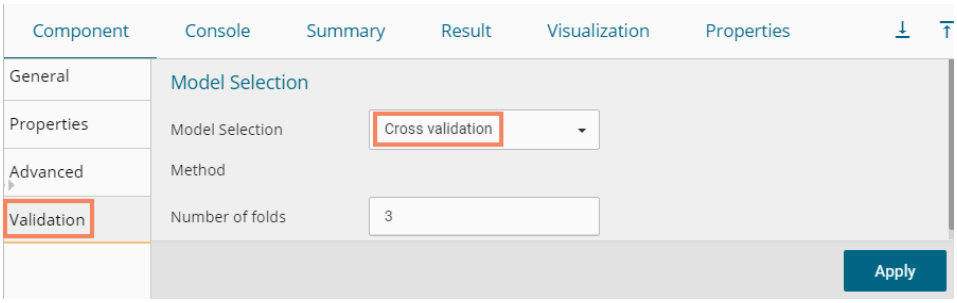


iv) Click the '**Validation**' tab and configure the required fields

a. Model Selection Method: Select a method using the drop-down menu. Users need to configure the other fields based on the selected model method.

i. Cross-Validation

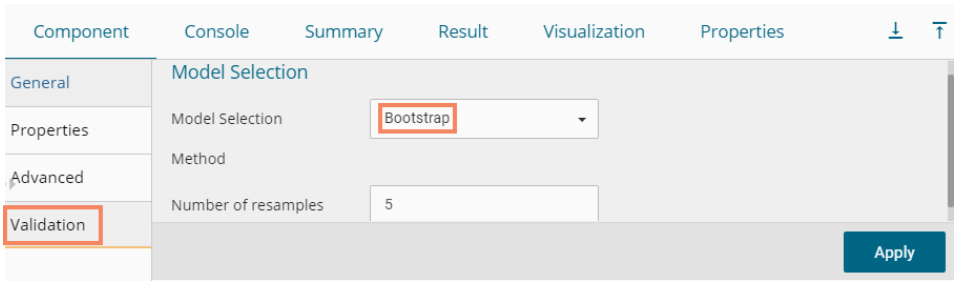
The user needs to configure the '**Number of folds**' if the selected model method is **Cross Validation**.



ii. Bootstrap

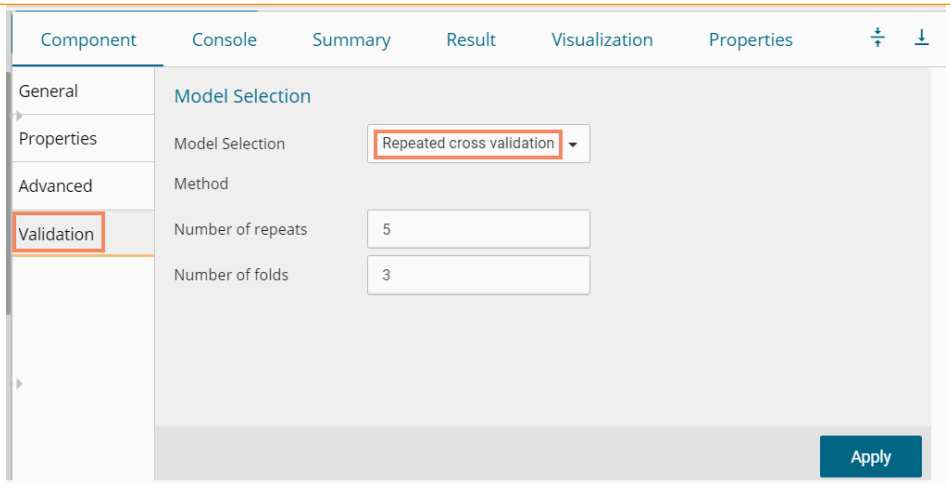
The user needs to configure the '**Number of resamples**' (the Default value for this field is

5) if the selected model method is '**Bootstrap.**'



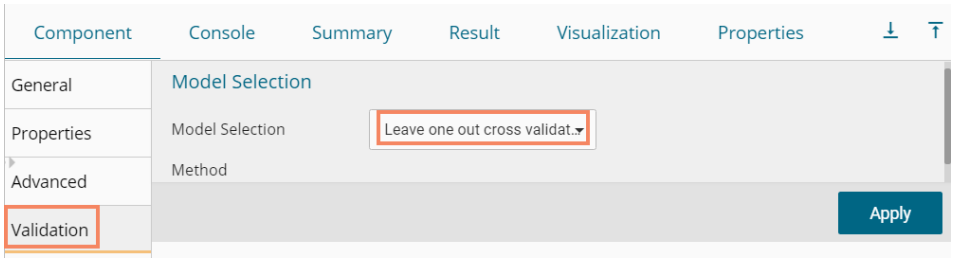
iii. Repeated Cross-Validation

The user needs to configure the '**Number of repeats**' and '**Number of folds**' if the selected method is '**Repeated Cross-Validation.**'



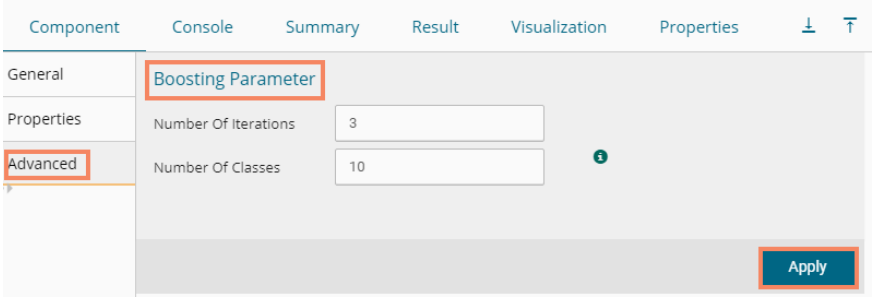
iv. Leave One Out Cross-Validation

The user does not get any other field to configure if the selected model method is **Leave one out cross-validation**.



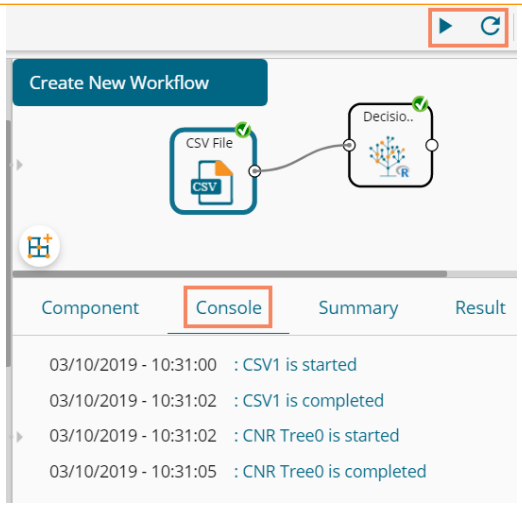
- **Advanced Tab when 'XG Boosting' is enabled**
  - **a. Boosting Parameter**
    - i. Number of Iterations: Enter a number suggesting the Number of Iterations
    - ii. Number of Classes: Enter a number suggesting the Number of Classes

v) Click the '**Apply**' option after configuring the required Properties, Advanced, and/or Validation fields as per your selection of the model.



vi) Run the workflow after getting the success message.

vii) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

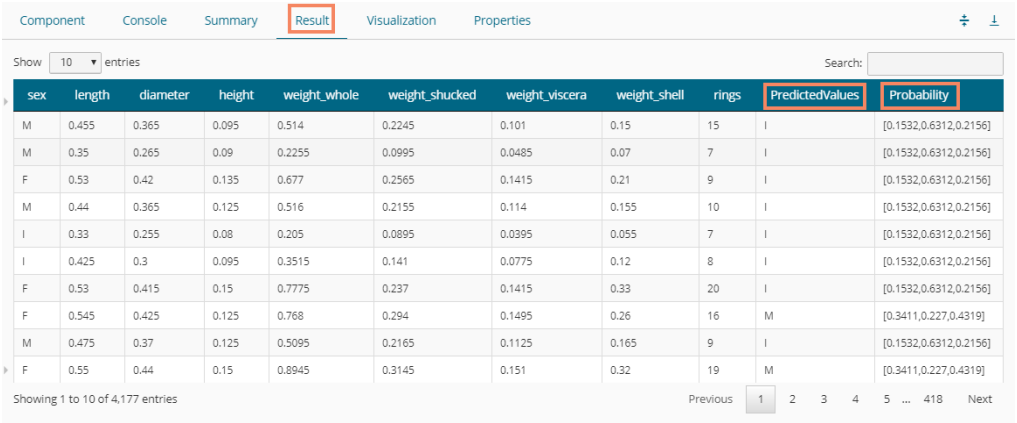


viii) Follow the below-given steps to display the Result view:

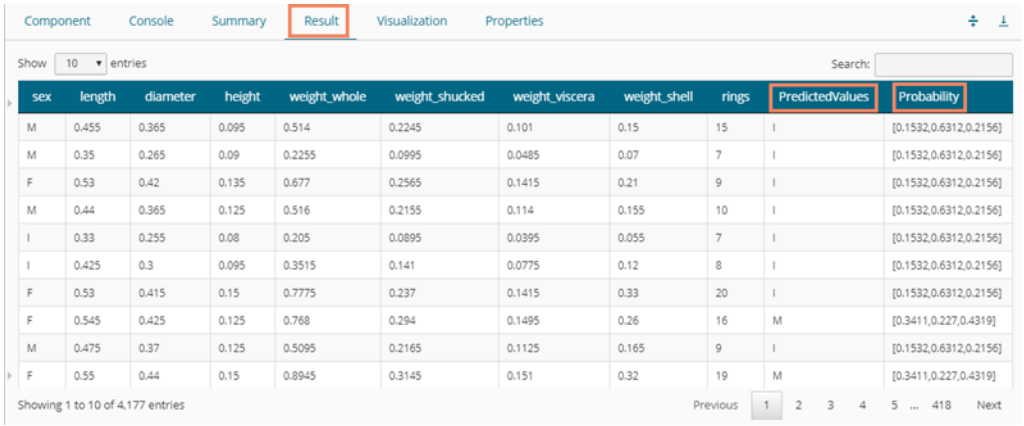
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

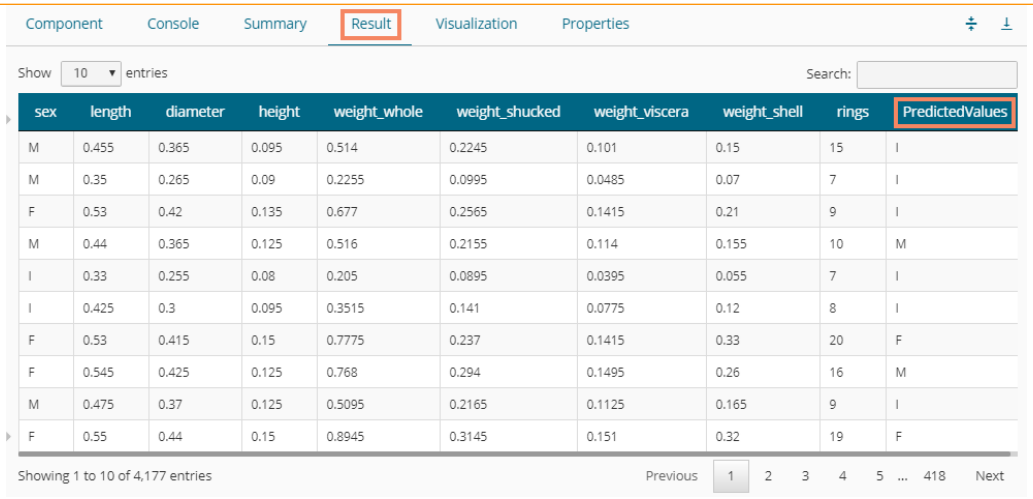
i. Result view when both the Model Tuning options are disabled



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



iii. Result view when '**XG Boosting**' is enabled

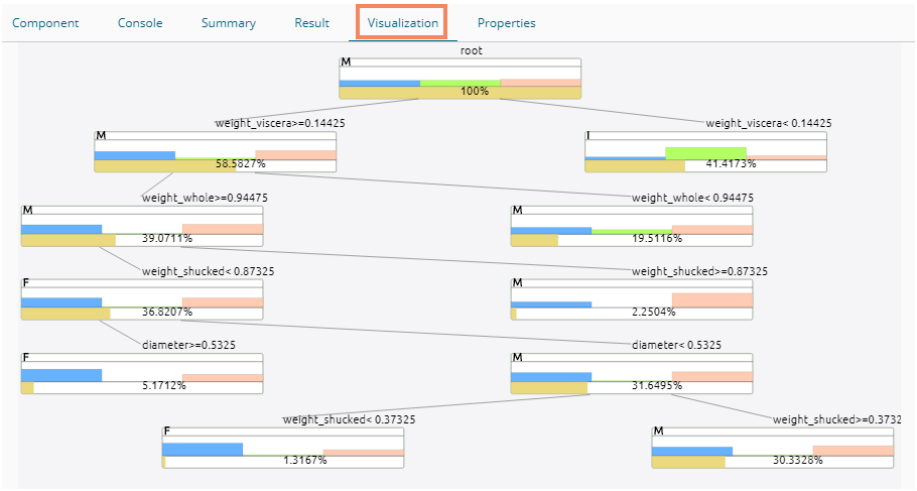


Note: The Probability column displays data in the Array format when Validation is enabled.

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

x) The Result data gets displayed via the tree chart.

a. Visualization tab when no Model Tuning option is enabled



b. Visualization tab when Validation is enabled

13.1.6.2. Regression as Algorithm Type for Decision Tree

i) Drag the Decision 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 the dependent column as the categorical values.
2. **Regression:** Select this option if users want to pass the dependent column as numerical values.

b. Column Selection

i. **Features**: Select input columns from the drop-down list to which the target 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

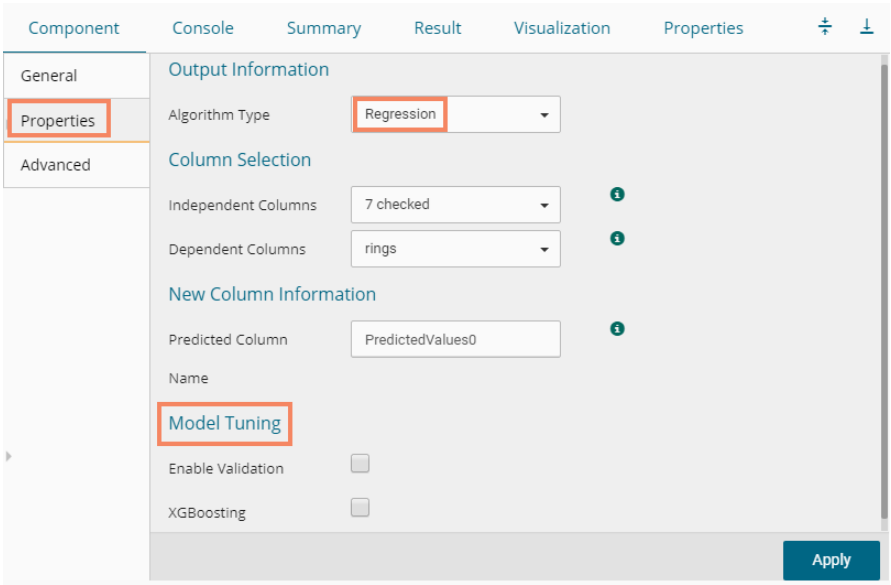
i. **Predicted Column Name**: Enter a name for the new column containing the predicted values.

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

d. Model Tuning

i. **Enable Validation:** Enable validation by a checkmark in the given box.

ii. **XG Boosting:** Enable XG Boosting with a checkmark in the given box.



Note: Other possible scenarios to configure the Properties tab can be when either of the Model Tuning option is enabled.

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

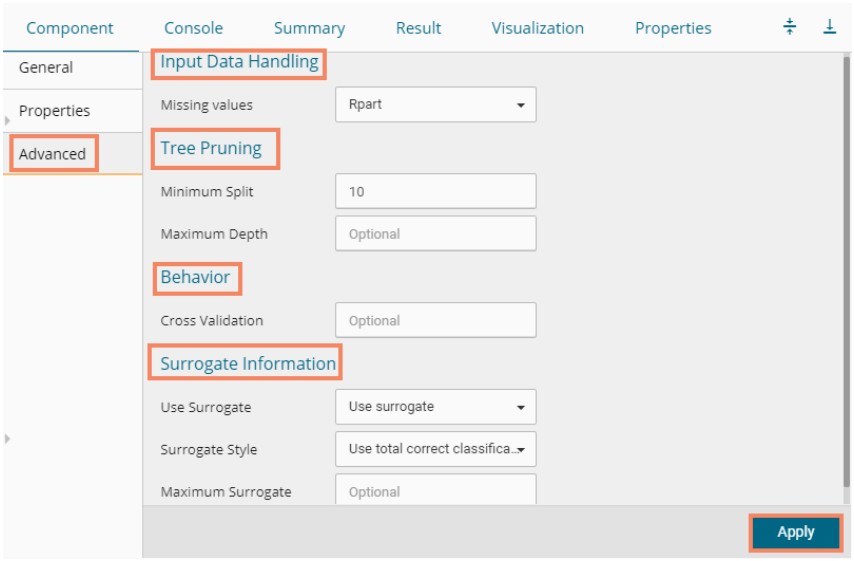
- **Advanced Tab when both the Model Tuning options are disabled:**

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

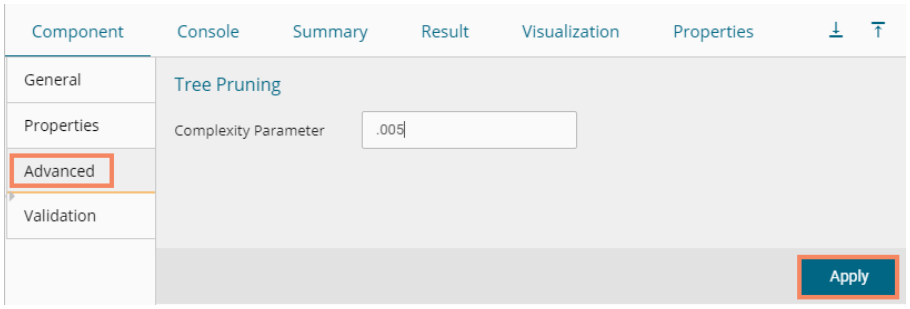
b. **Tree Pruning**
   i. **Minimum Split:** It indicates a minimum number of observations within a single node for a split to be attempted. The default value for this field is 10.
   ii. **Complexity Parameter:** This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is pruned off performing cross validation, hence the program does not pursue it. The default value for this field is 0.05.
   iii. **Maximum Depth:** It sets the maximum depth of any node of the final tree keeping the depth count for root node 0. It is an optional field (It is recommended to set Maximum Depth value less than 30 rpart for 32 bit-machines.)

c. **Behavior**
   i. **Split Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'** tab. (This field appears 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 other data values mentioned in the selected dataset. (This field appears when the selected algorithm type is '**Classification**').

d. **Surrogate Information**
   i. **Use Surrogate:** Select one option from the drop-down menu.
      1. **Display Only:** Select this option to display only the observation, but not split it further.
      2. **Use Surrogate:** Select this option to search surrogate values for the missing values to split the observation. Two fields are displayed:
         a. **Surrogate Style:** Select a style using the drop-down menu.
         b. **Maximum Surrogate:** Set the maximum surrogate value.
      3. **Stop if missing:** Select this option to choose an action based on the nature of the majority observations. If values are missed for all the observations, then they will stop splitting further.

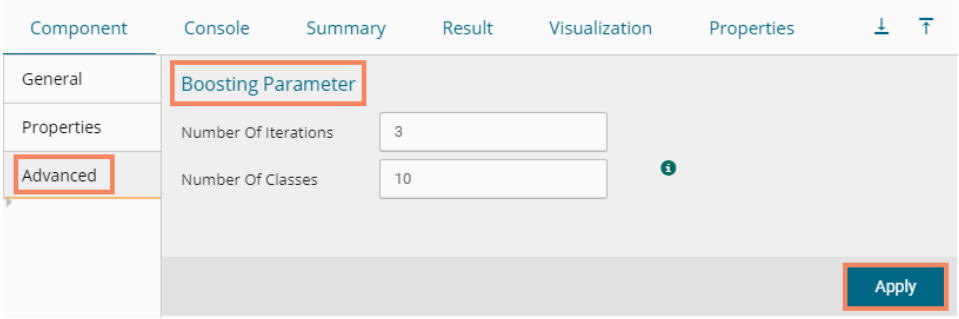


- **Advanced Tab when 'Validation' is enabled:**
  - **a. Tree Pruning**:
    - i. Complexity Parameter: This parameter is primarily used to save computing time by pruning off splits that are not worthwhile. Any split which does not improve the fit by a factor of the complex parameter is pruned off performing cross-validation, hence the program does not pursue it. The default value for this field is 0.05.



iv) Click the '**Validation**' tab and configure the required fields. The user can refer to the description provided under section 12.2.6.1 to configure the Validation tab.

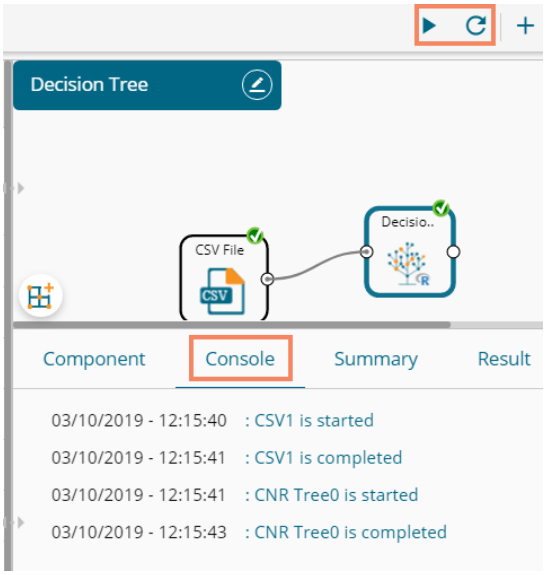
- **Advanced Tab when XG Boosting is Enabled**
  - **a. Boosting Parameter**
    - i. Number of Iterations: Enter a number suggesting the Number of Iterations
    - ii. Number of Classes: Enter a number indicating the Number of Classes



v) Click the **'Apply'** option**.**

vi) Run the workflow after getting the success message.

vii) The '**Console**' tab opens.

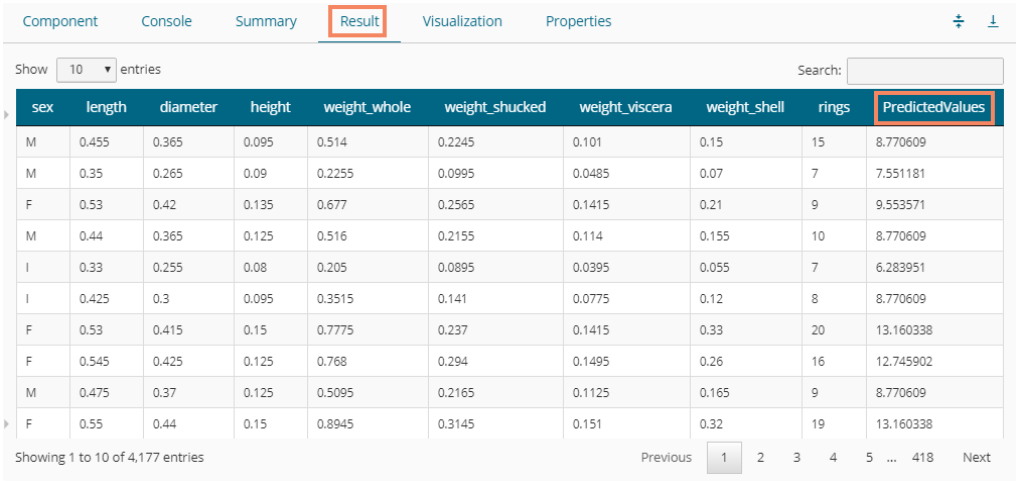


viii) Follow the below-given steps to display the Result view:

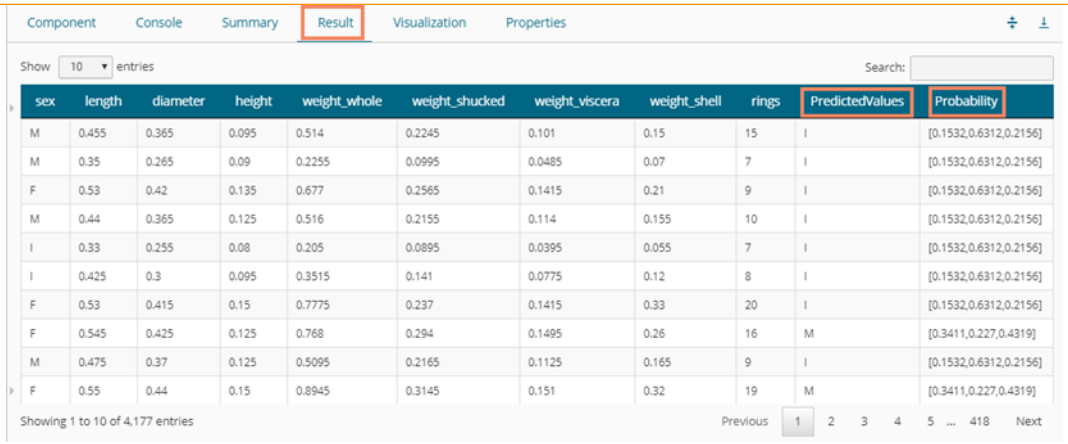
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

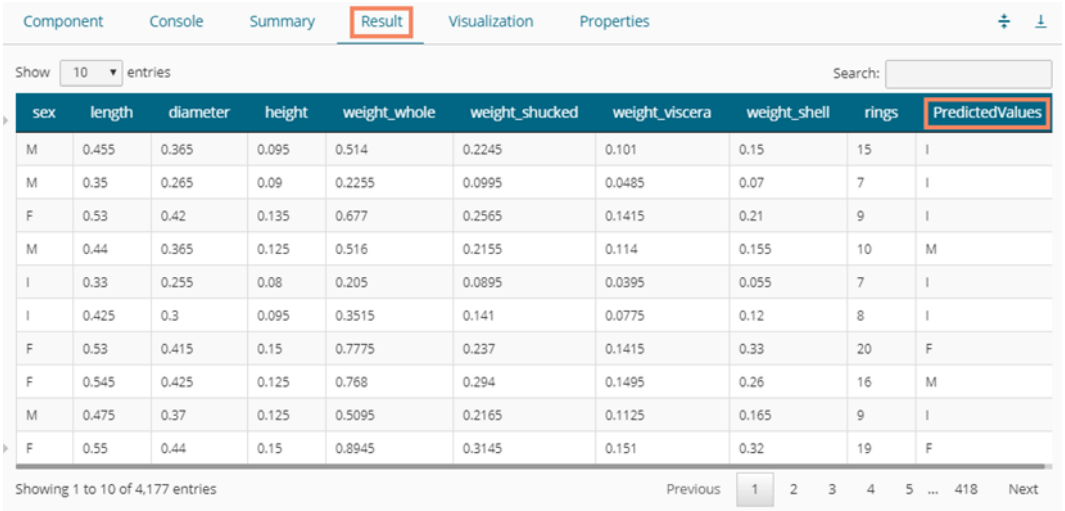
i. The Result tab when both the Model Tuning options are disabled



ii. The Result tab when the '**Validation**' option is enabled



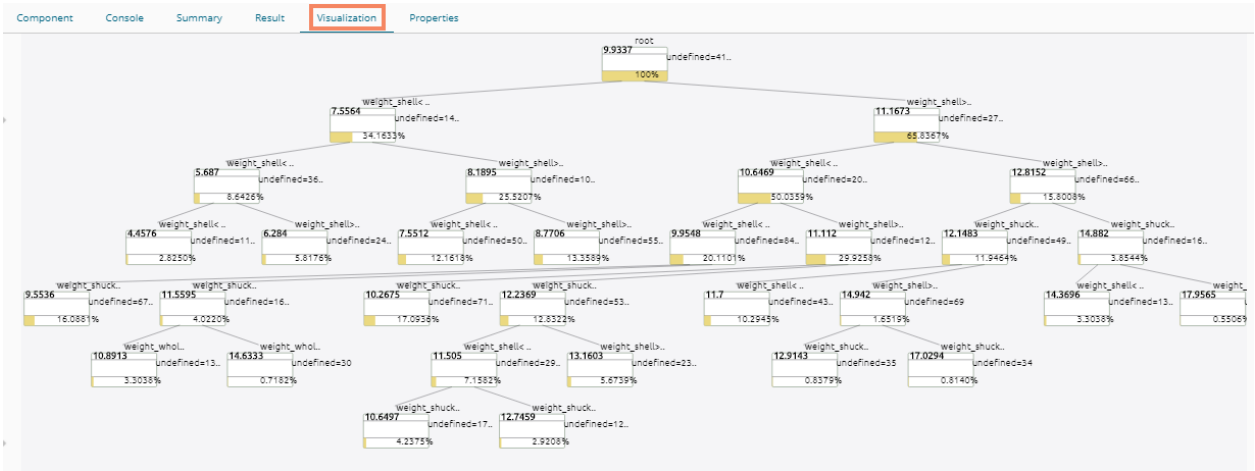
iii. Result view when '**XG Boosting**' is enabled



Note: The Probability column is displayed in the Array format while enabling the 'Validation' option.

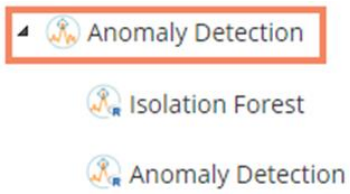
ix) Click the '**Visualization**' tab.

x) The Result data will be displayed via the tree chart.
(The following visualization displays processed data when no Model Tuning option is enabled.)



13.1.7. Anomaly Detection

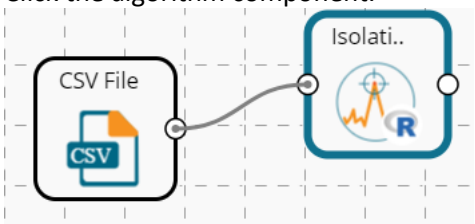
The Anomaly Detection algorithm has the following algorithms:



13.1.7.1. Isolation Forest

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

ii) Click the algorithm component.

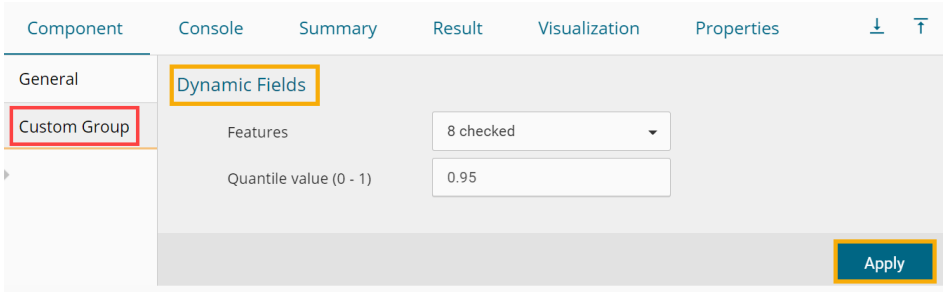


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. **Features**: Select the input columns with which you want to perform the Analysis

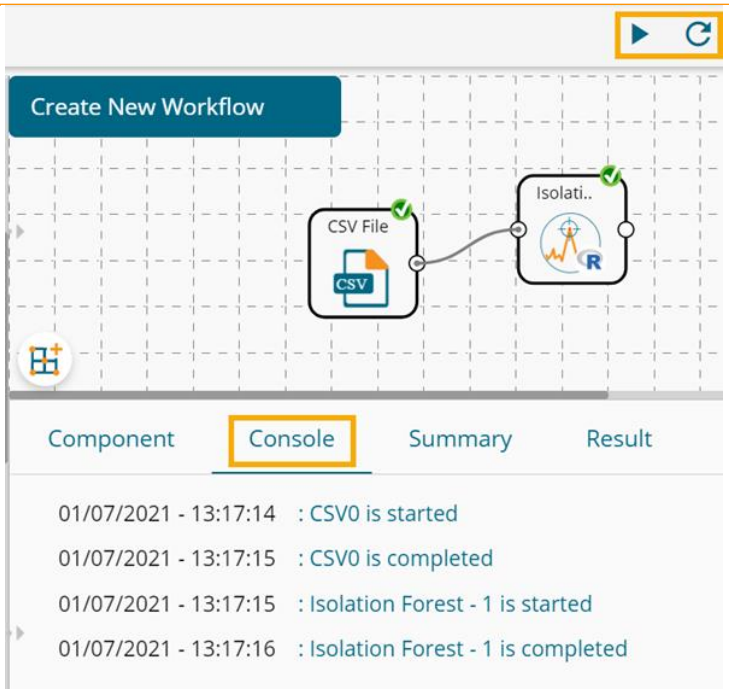
b. **Quantile Value (0-1)**: Select a number between 0 to 1 to configure the Quantile Value.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

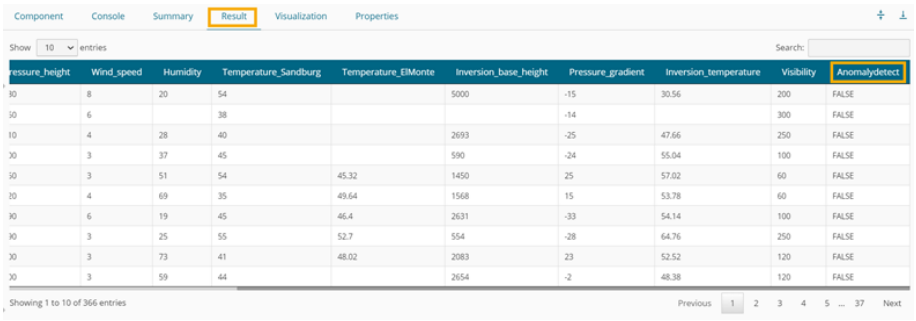
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



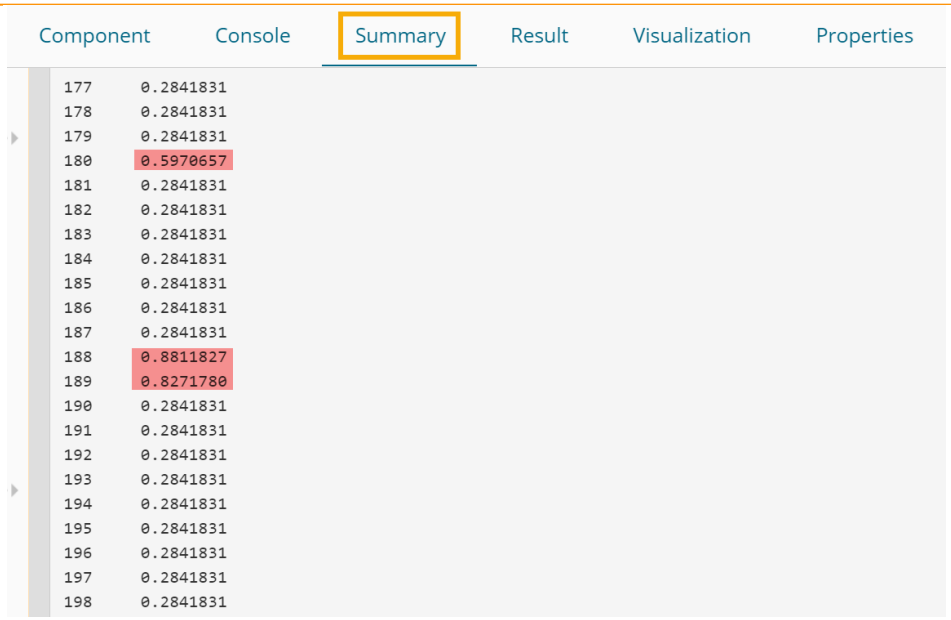
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



13.1.7.2. Anomaly Detection

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

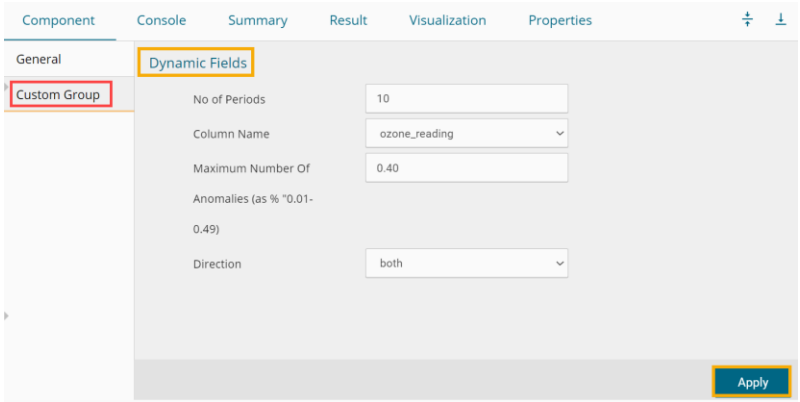
ii) Click the algorithm component.

iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Provide the following details to configure the **Dynamic Fields:**

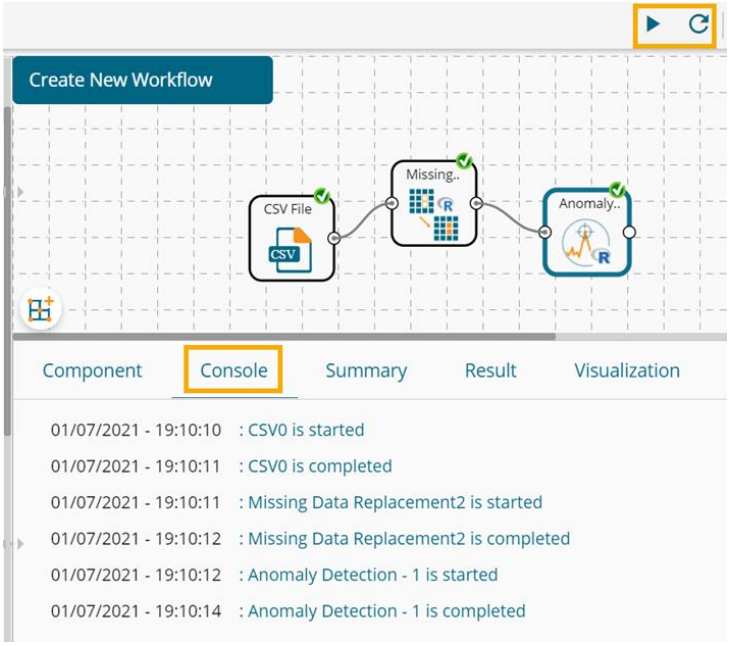
a. **No. of Periods**
b. **Column Name**
c. **Maximum Number of Anomalies (as % "0.01-0.49)**
d. **Direction**

v) Click the '**Apply**' option.



vi) Run the workflow after getting the success message.

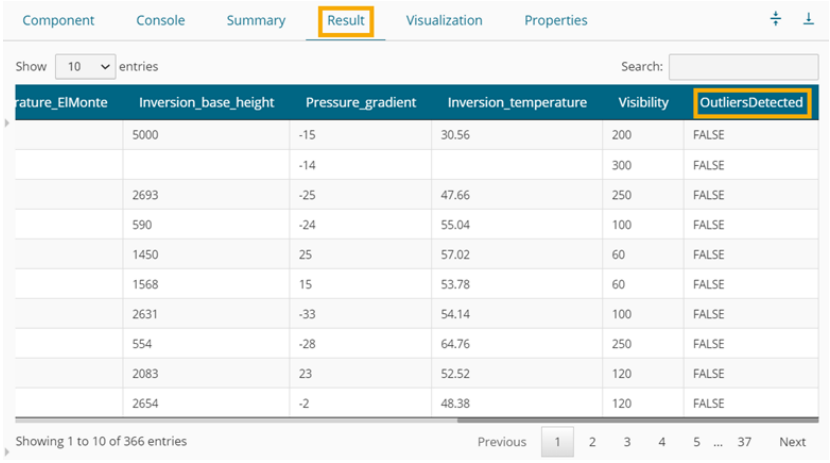
vii) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



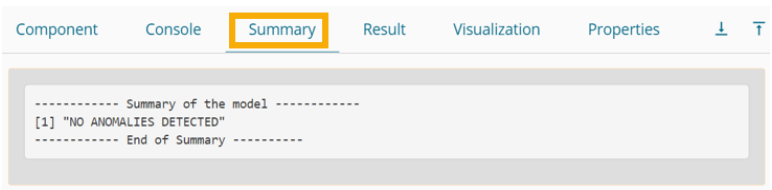
viii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

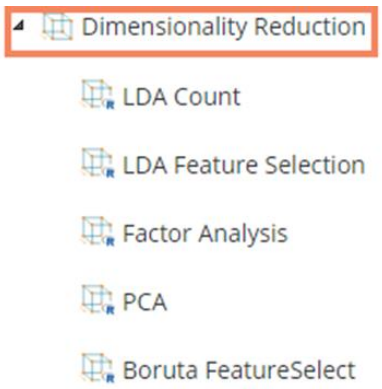


ix) Click the '**Summary**' tab to open the model summary.



13.1.8. Dimensionality Reduction

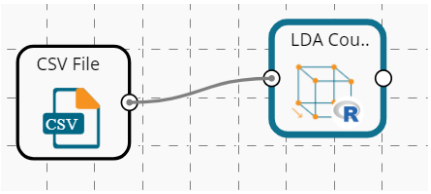
The Dimensionality Reduction algorithm has the following algorithms:



13.1.8.1. LDA Count

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

ii) Click the algorithm component.

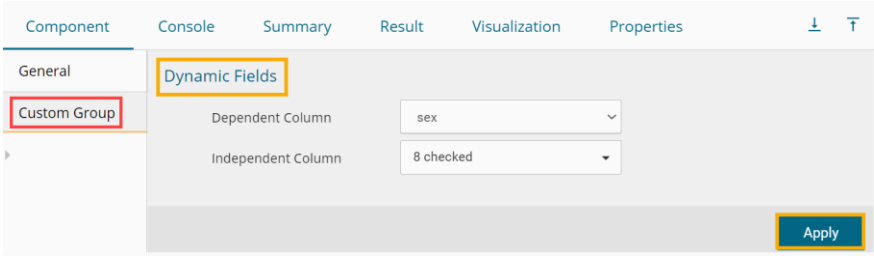


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

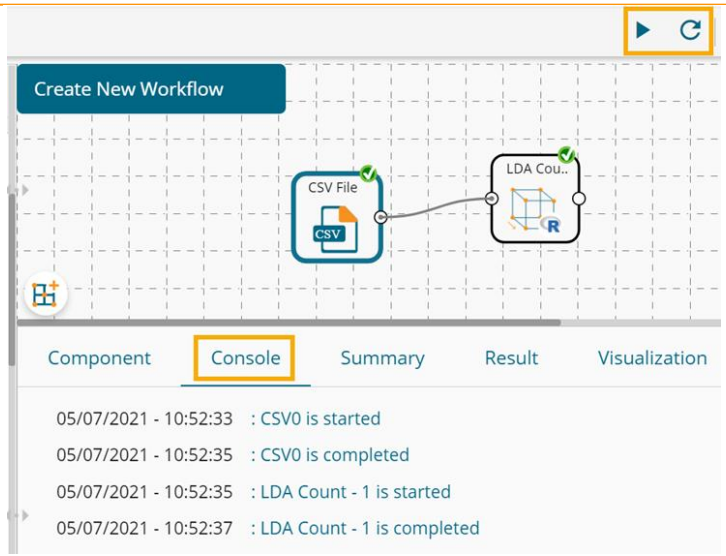
b. **Independent Column**: Select the required input columns against which the analysis gets applied to the target column.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

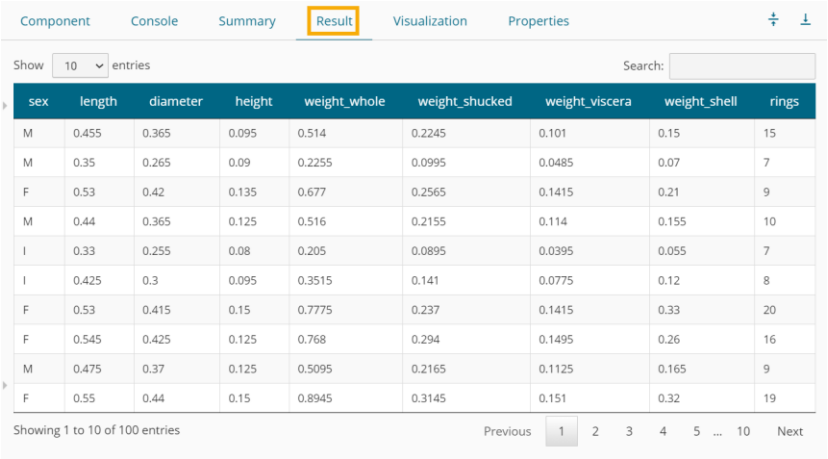
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



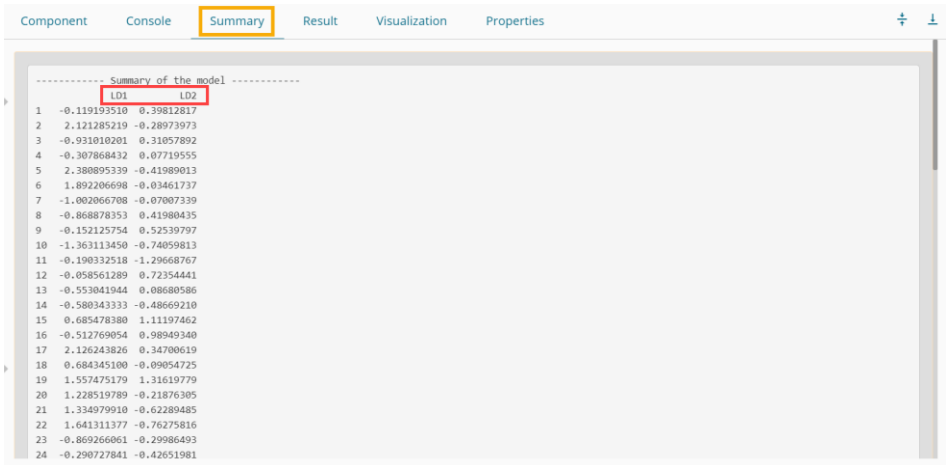
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



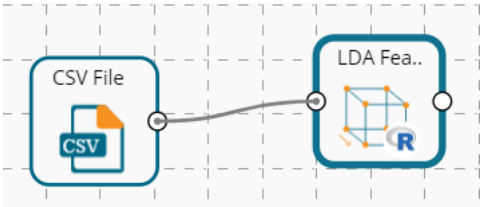
viii) Click the '**Summary**' tab to open the model summary.



13.1.8.2. LDA Feature Selection

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

ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

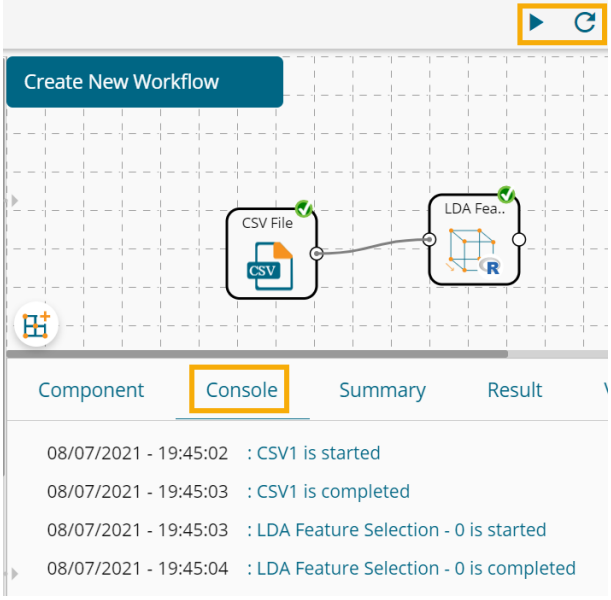
a. **Dependent Column**: Select the target column on which the regression analysis gets applied.

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

iv) Click the '**Apply**' option.

v) Run the workflow after getting the success message.

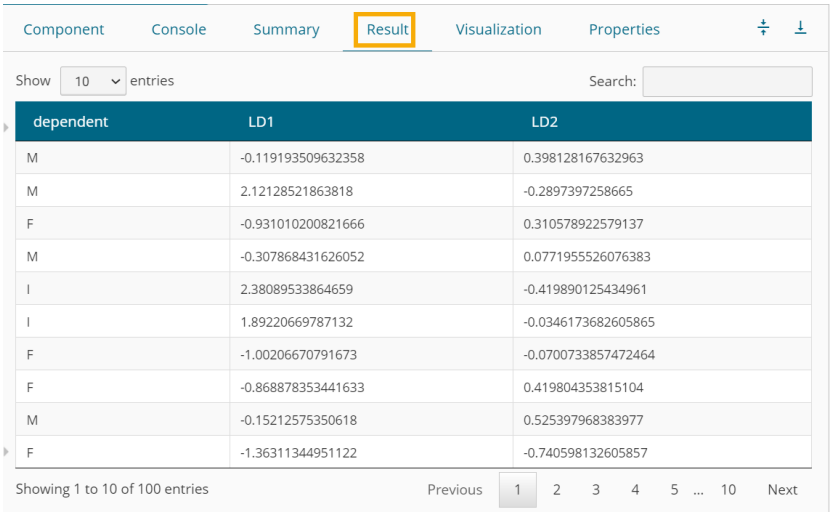
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



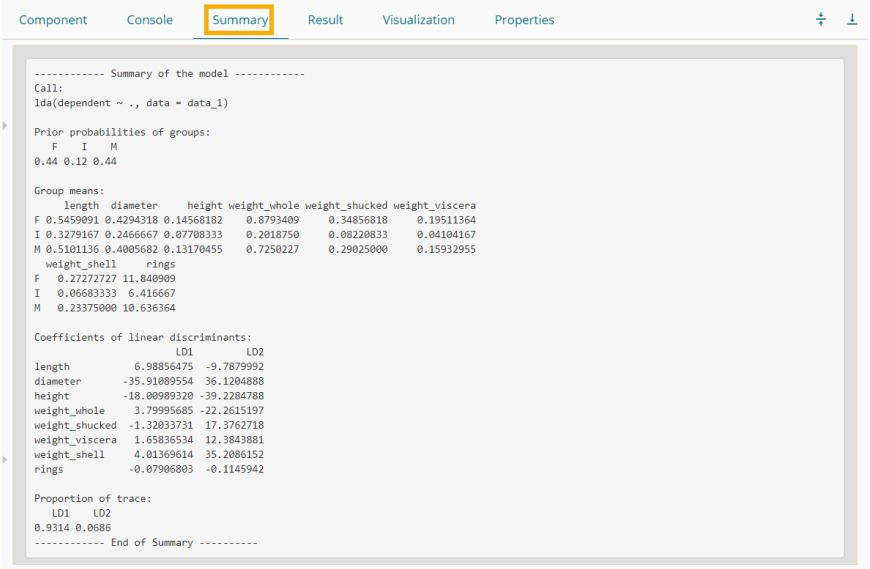
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



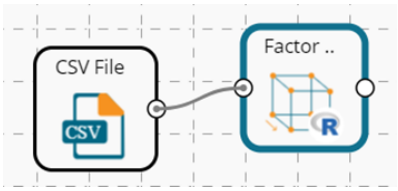
viii) Click the '**Summary**' tab to open the model summary.



13.1.8.3. Factor Analysis

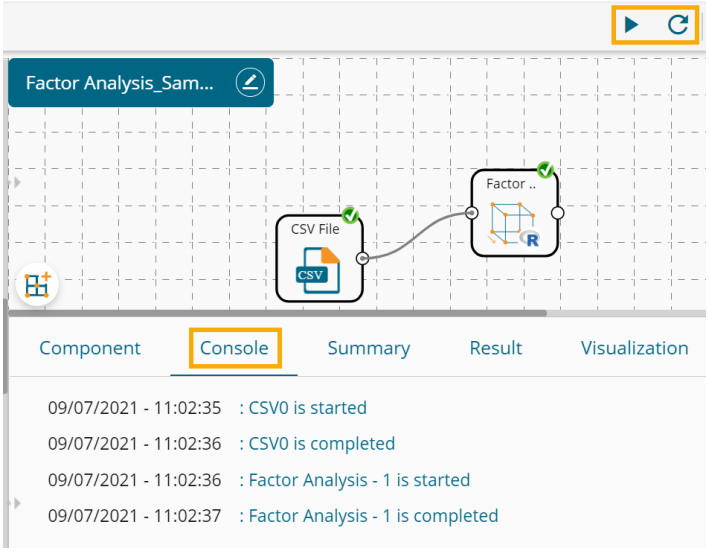
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component.

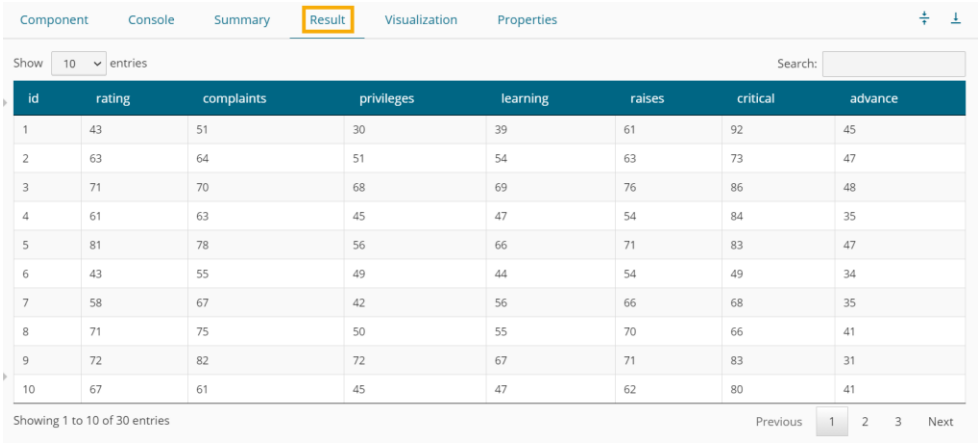


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
iv) Provide the value of Factors and Variables.
v) Click the '**Apply**' option.

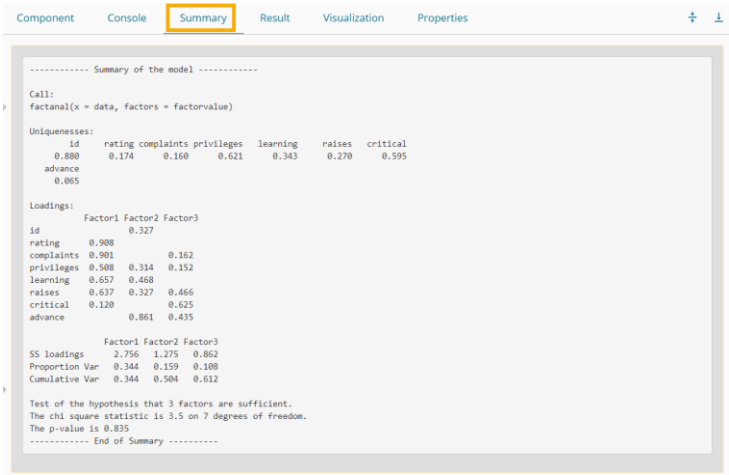
vi) Run the workflow after getting the success message.
vii) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.



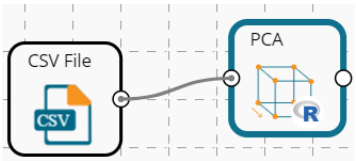
ix) Click the '**Summary**' tab to open the model summary.



13.1.8.4. PCA

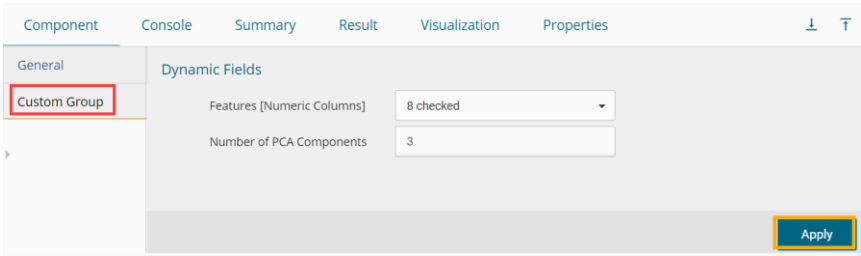
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component.



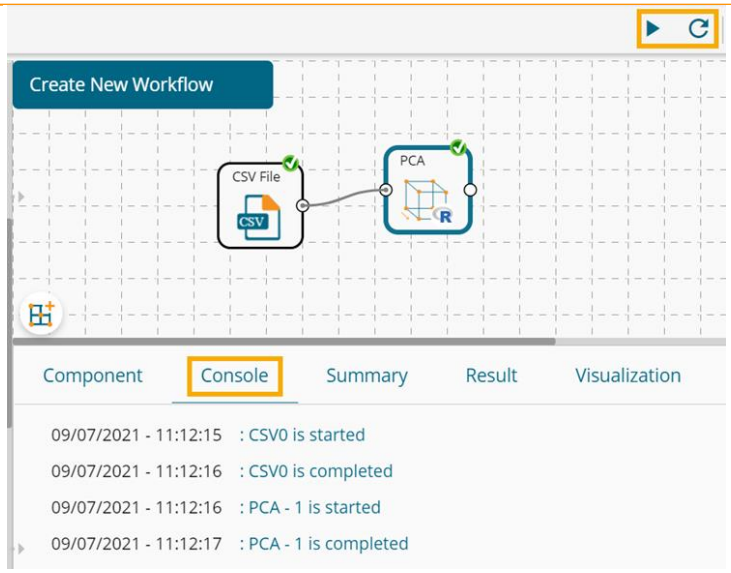
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component as given below:
   a. Features [Numeric Columns]
   b. Number of PCA Components

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

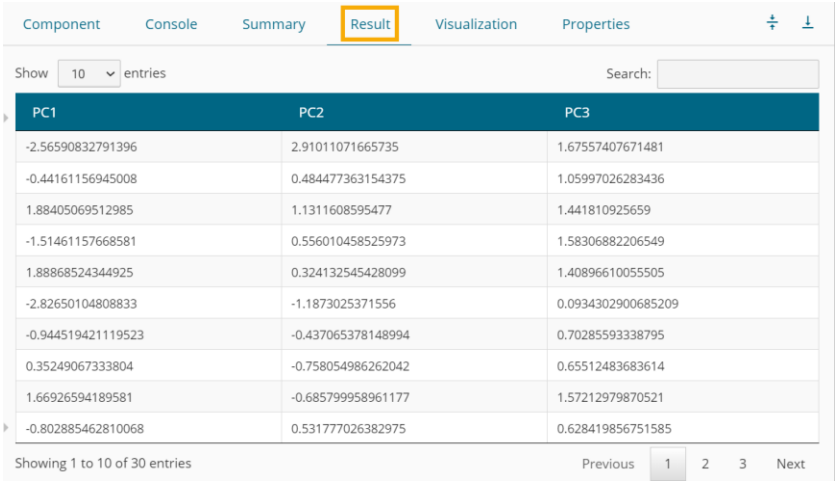
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



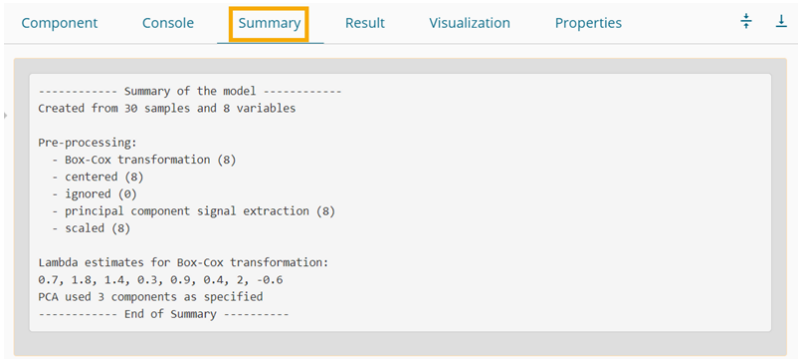
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



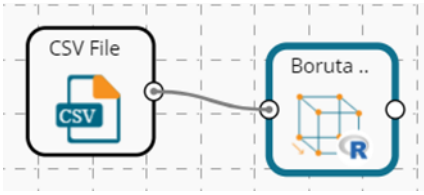
viii) Click the '**Summary**' tab to open the model summary.



13.1.8.5. Boruta Feature Select

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

ii) Click the algorithm component.

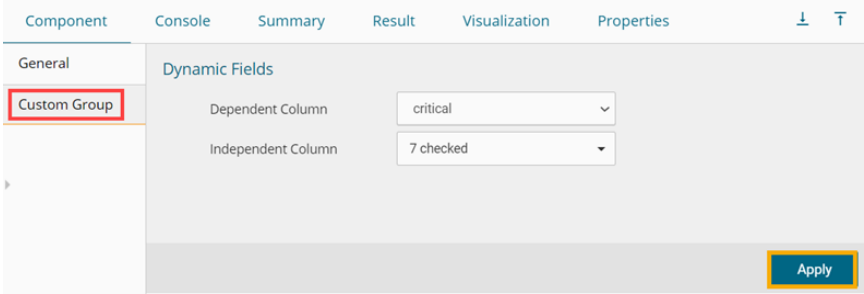


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

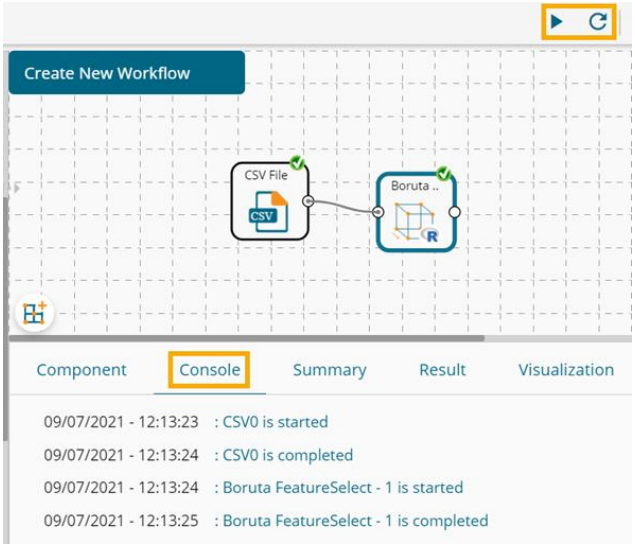
b. **Independent Column**: Select the required input columns against which the analysis gets applied to the target column.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

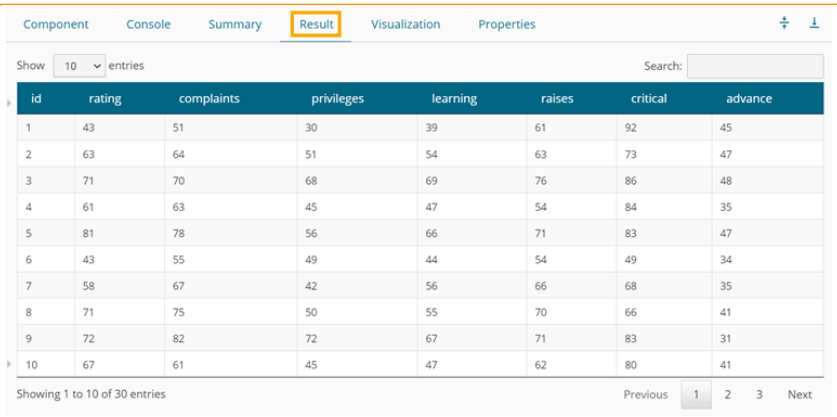
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:

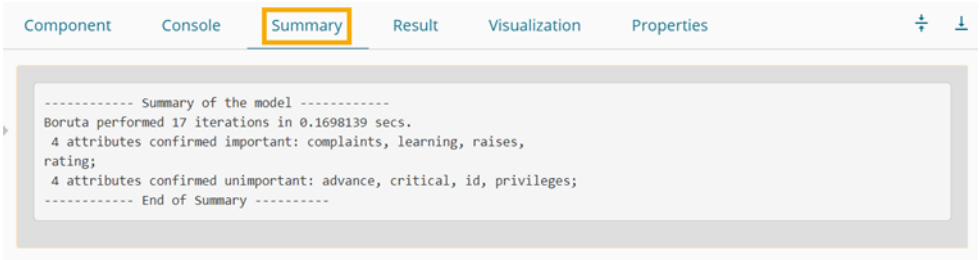
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

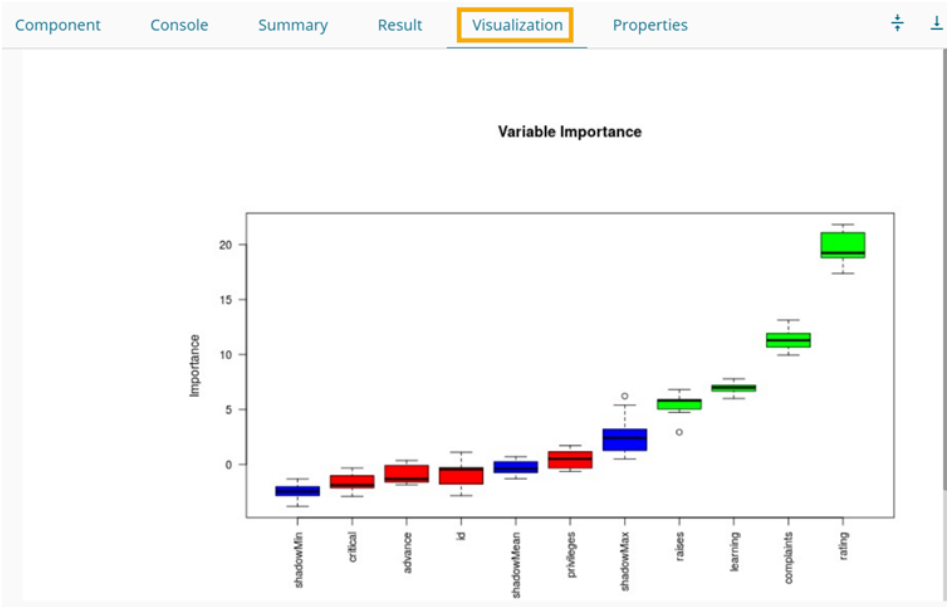


id	rating	complaints	privileges	learning	raises	critical	advance
1	43	51	30	39	61	92	45
2	63	64	51	54	63	73	47
3	71	70	68	69	76	86	48
4	61	63	45	47	54	84	35
5	81	78	56	66	71	83	47
6	43	55	49	44	54	49	34
7	58	67	42	56	66	68	35
8	71	75	50	55	70	66	41
9	72	82	72	67	71	83	31
10	67	61	45	47	62	80	41

viii) Click the '**Summary**' tab to open the model summary.

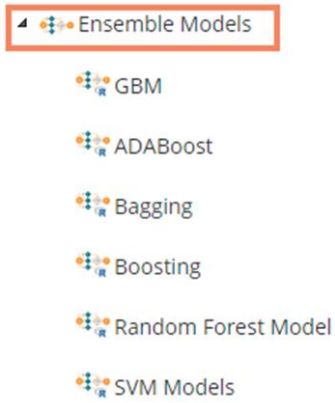


ix) Click the '**Visualization**' tab to see the processed data through a chart having a candlestick pattern.



13.1.9. Ensemble Models

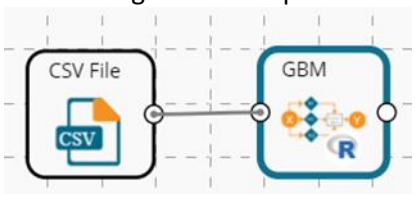
The Ensemble Models tree node contains the following algorithms:



13.1.9.1. GBM

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

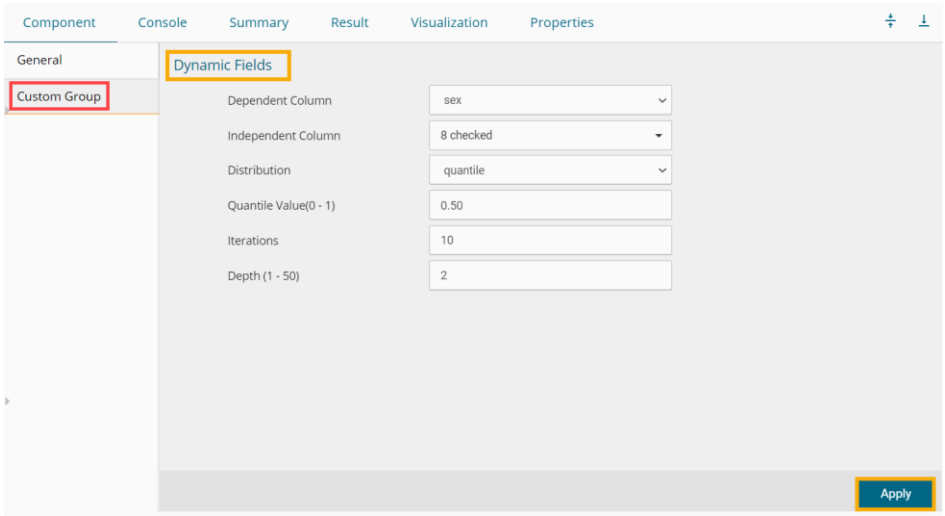
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component as given below:

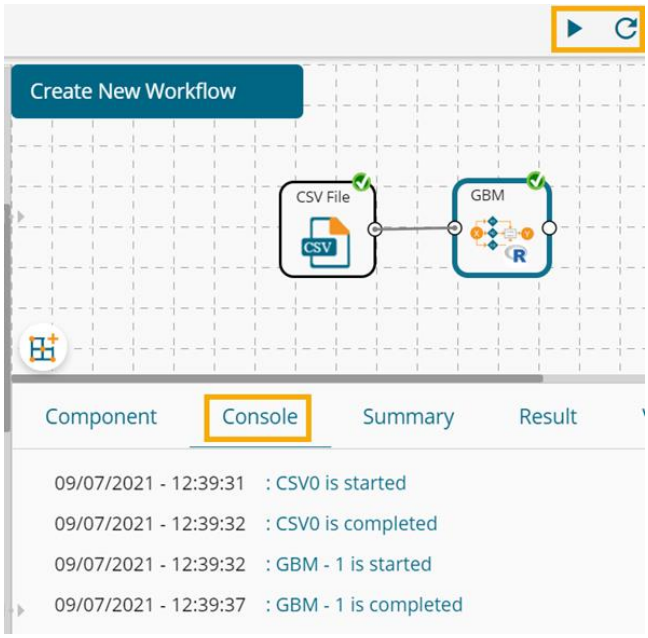
a. Dependent Column
b. Independent Column
c. Distribution
d. Quantile Value (0-1)
e. Iterations
f. Depth (1-50)

iv) Click the '**Apply**' option.



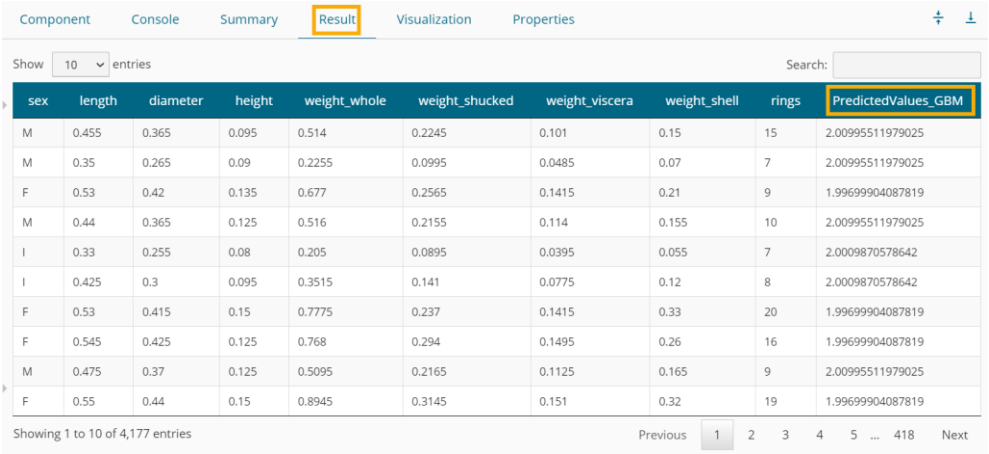
v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

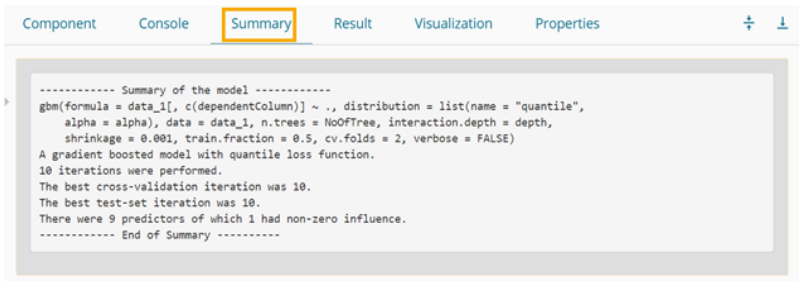


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.



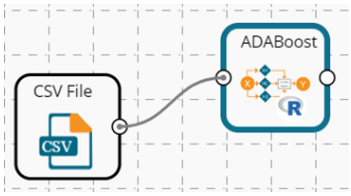
viii) Click the '**Summary**' tab to open the model summary.



13.1.9.2. ADABoost

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

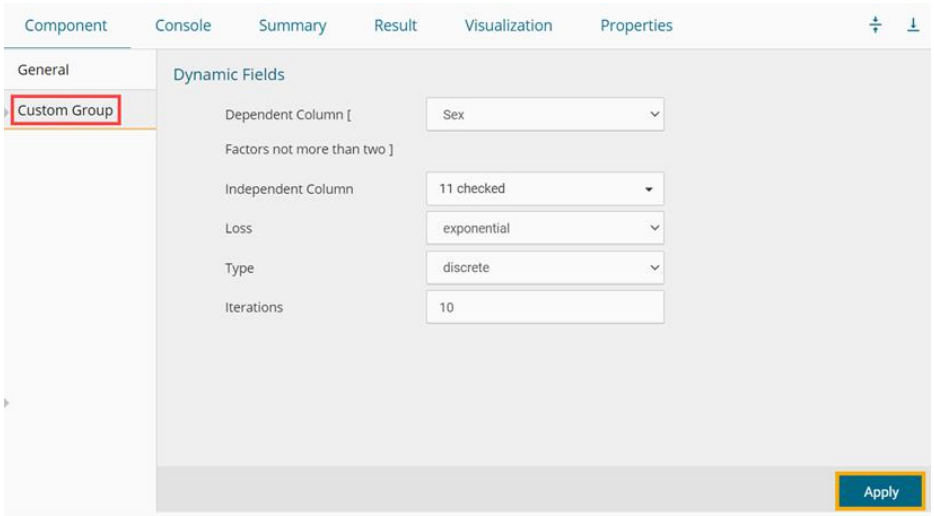
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component as given below:

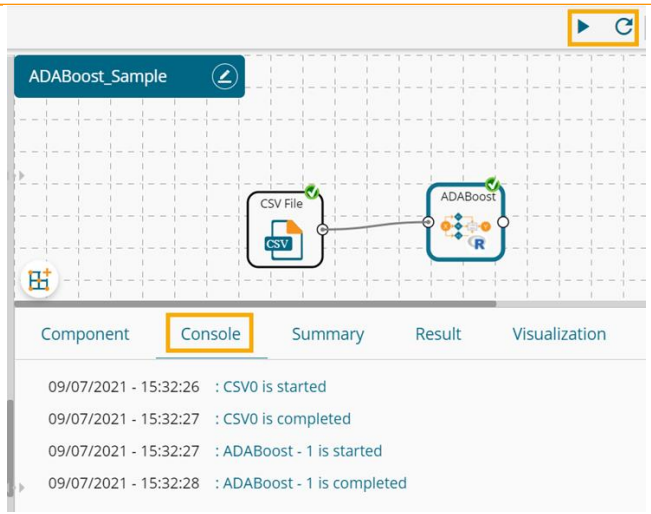
a. Dependent Column [Factors not more than two]
b. Independent Column
c. Loss
d. Type (The provided choice for this field is discrete, real, and gentle}
e. Iterations

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

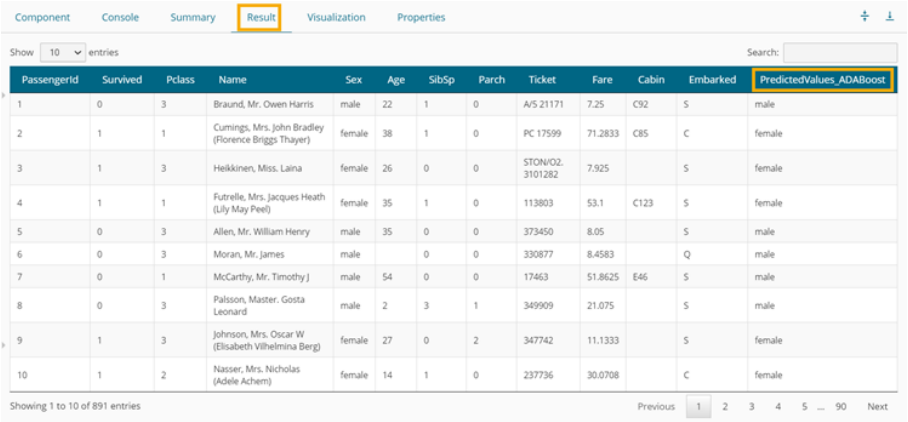
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



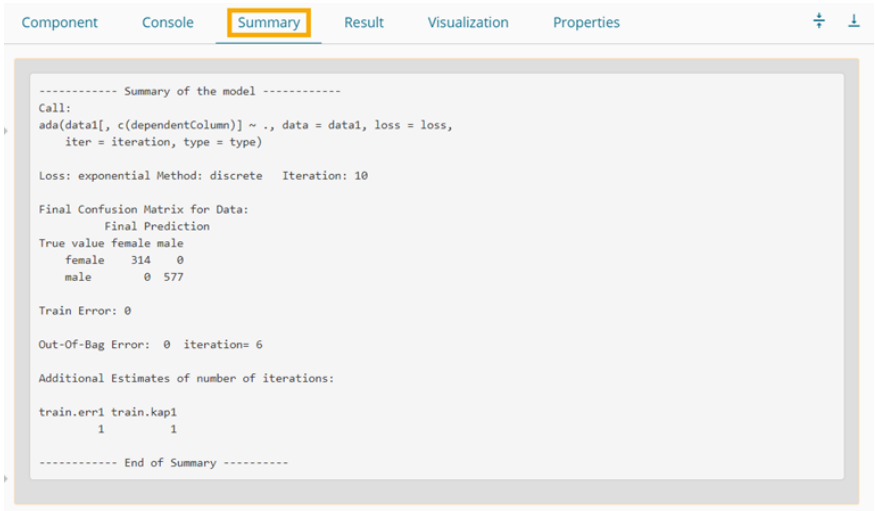
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



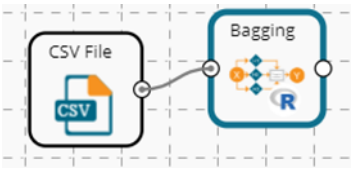
viii) Click the '**Summary**' tab to open the model summary.



13.1.9.3. Bagging

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

ii) Click the algorithm component.



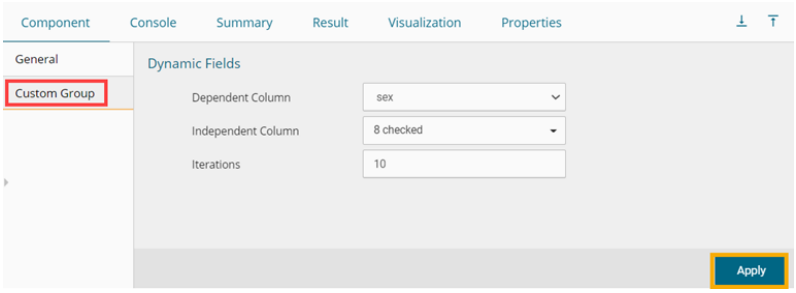
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component as given below:

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

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

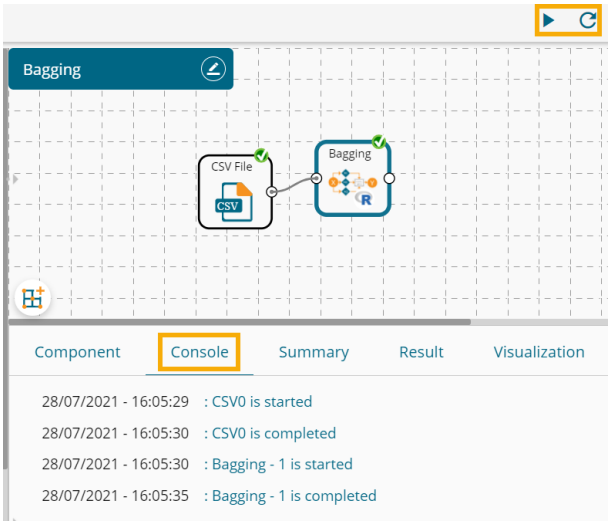
c. **Iterations**: The number of passes over the training data (aka epochs)

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

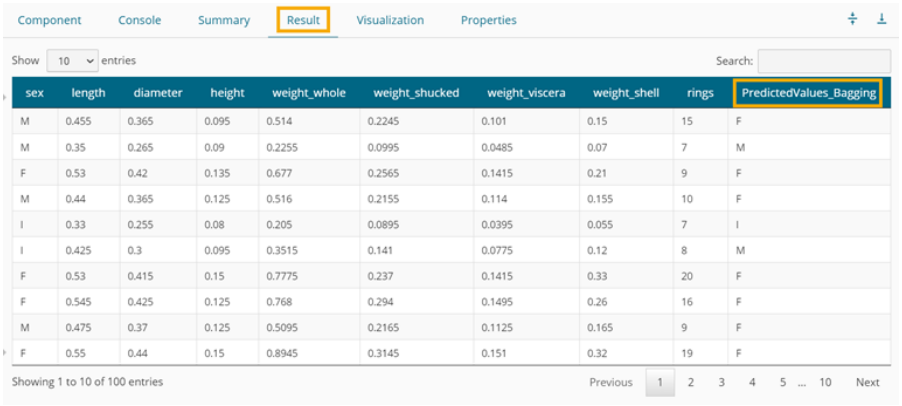
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



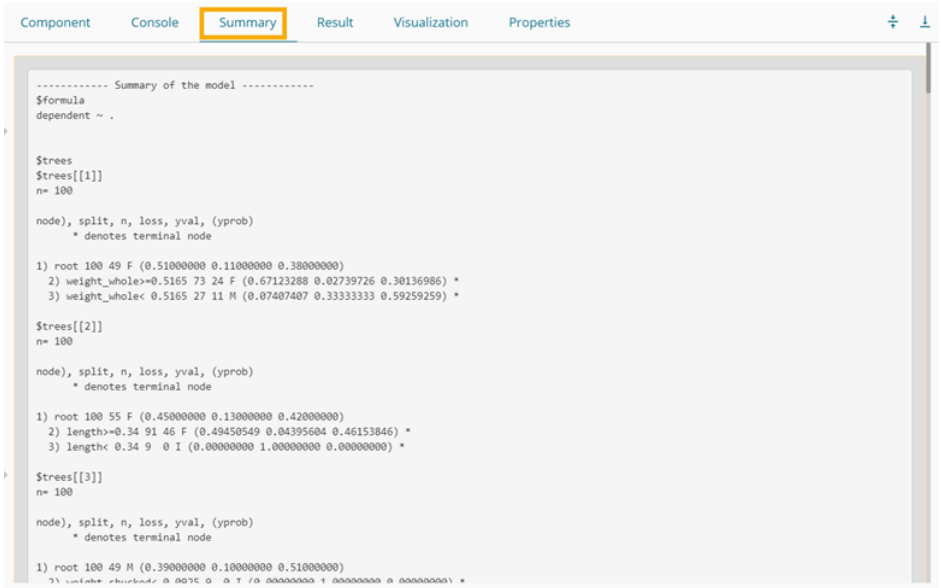
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

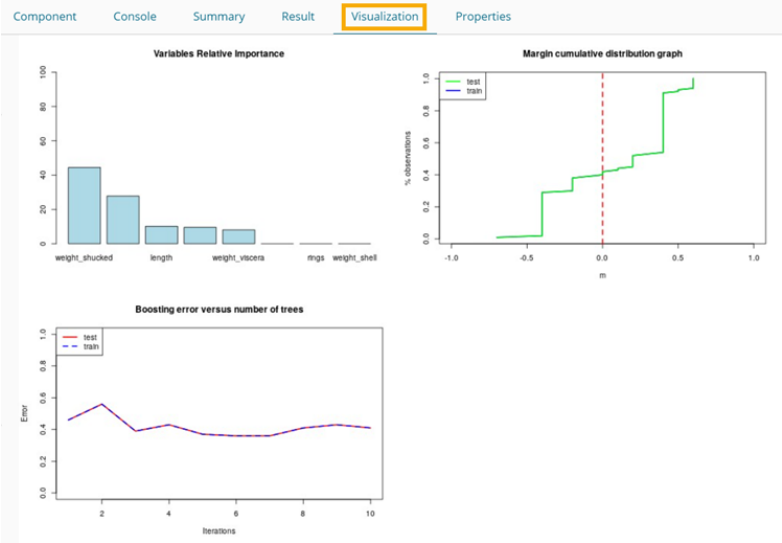
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



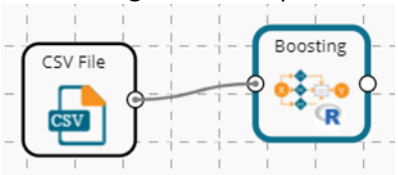
ix) Click the '**Visualization**' tab to open the processed data through a chart.



13.1.9.4. Boosting

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

ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

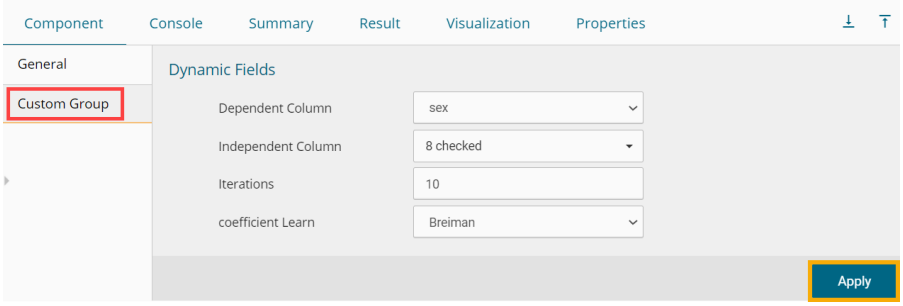
a. **Dependent Column**: Select the target column on which the regression analysis gets applied.

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

c. **Iterations**: The number of passes over the training data (aka epochs)

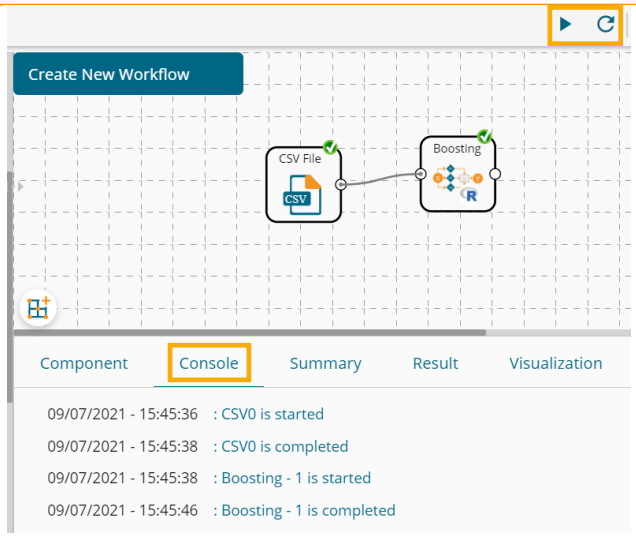
d. **Coefficient Learn**

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

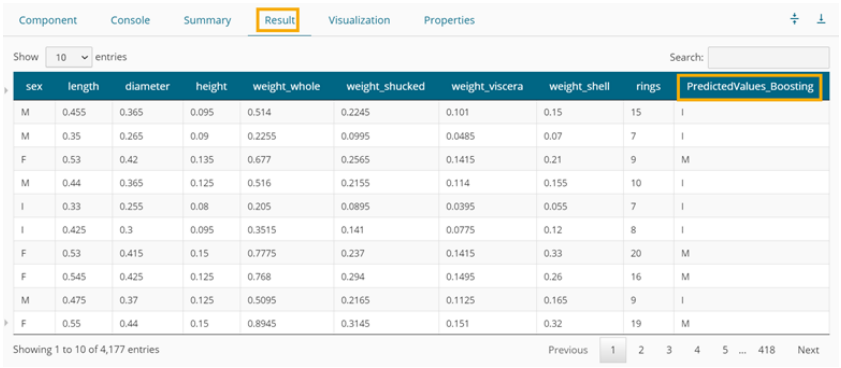
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



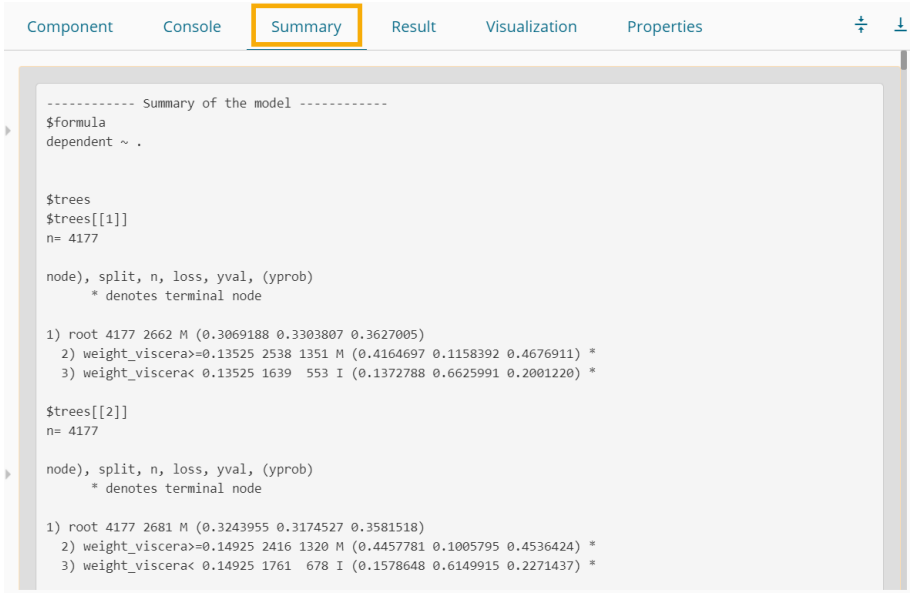
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

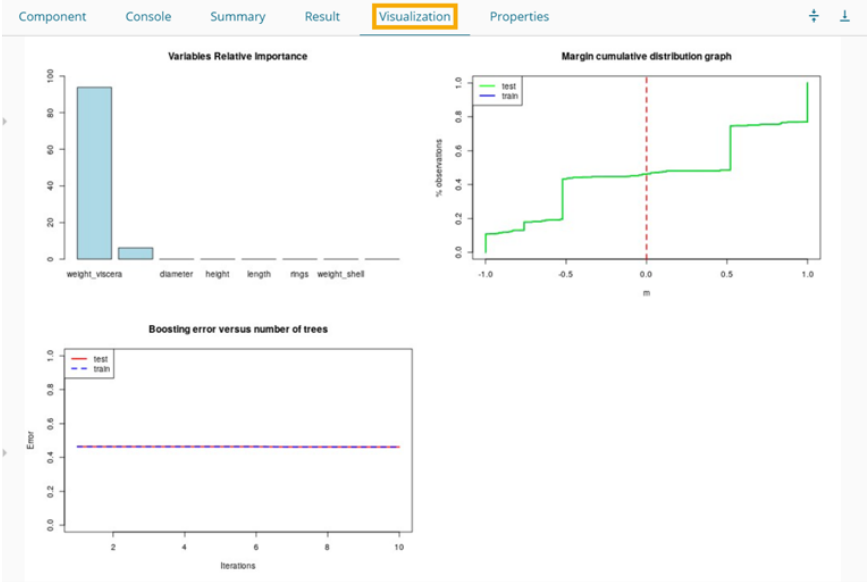
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



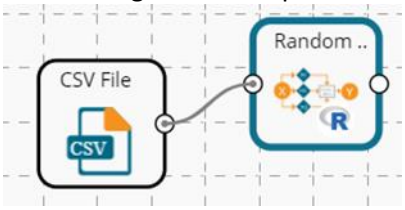
ix) Click the '**Visualization**' tab to see the result data through a chart.



13.1.9.5. Random Forest Model

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

ii) Click the algorithm component.



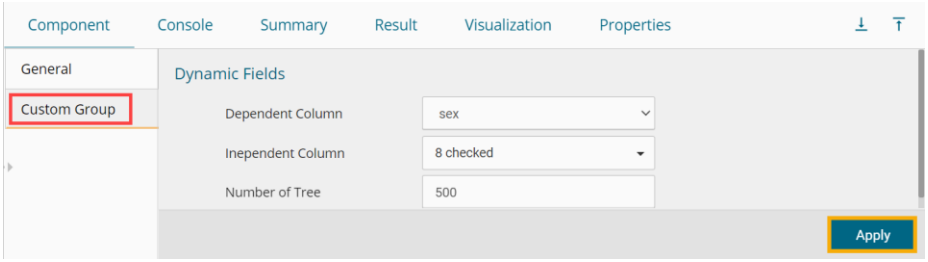
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

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

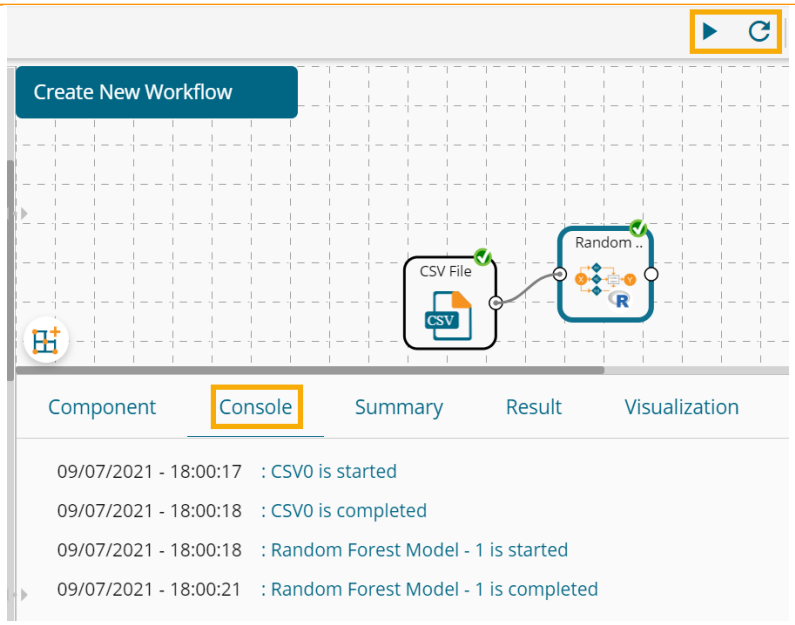
c. **Number of Tree**: Provide a digit to set the number of trees.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

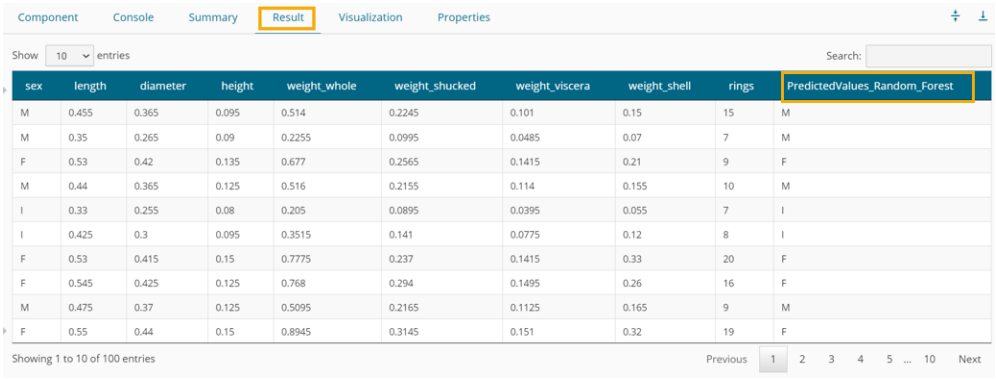
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



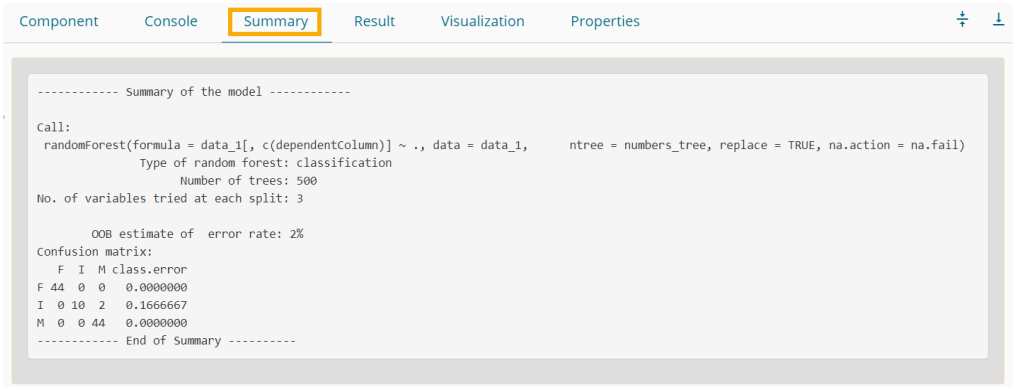
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

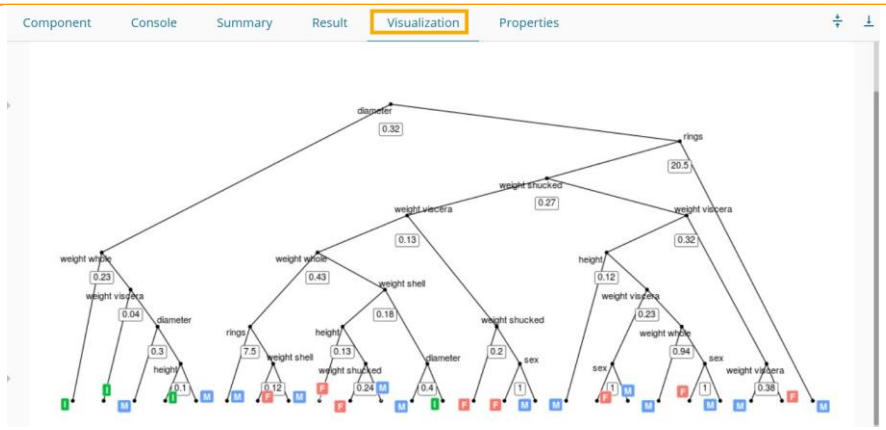
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



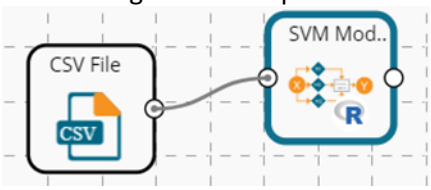
ix) Click the '**Visualization**' tab to see the result data through a chart.



13.1.9.6. SVM Models

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

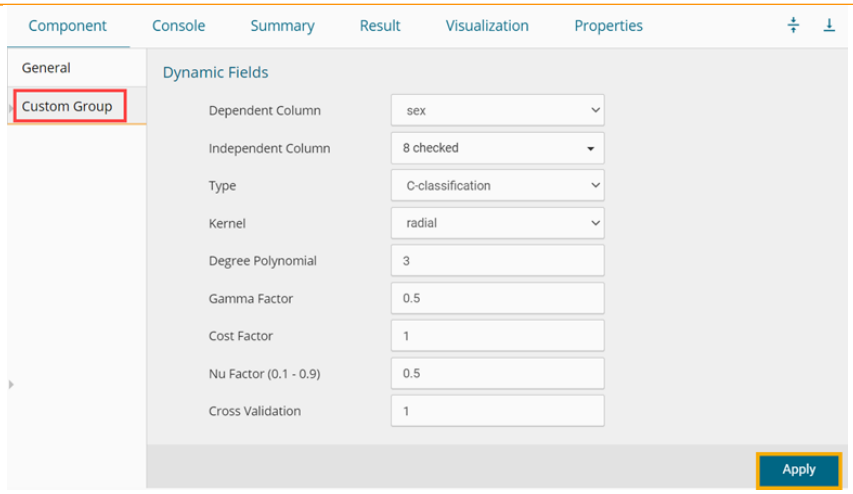
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component as given below:

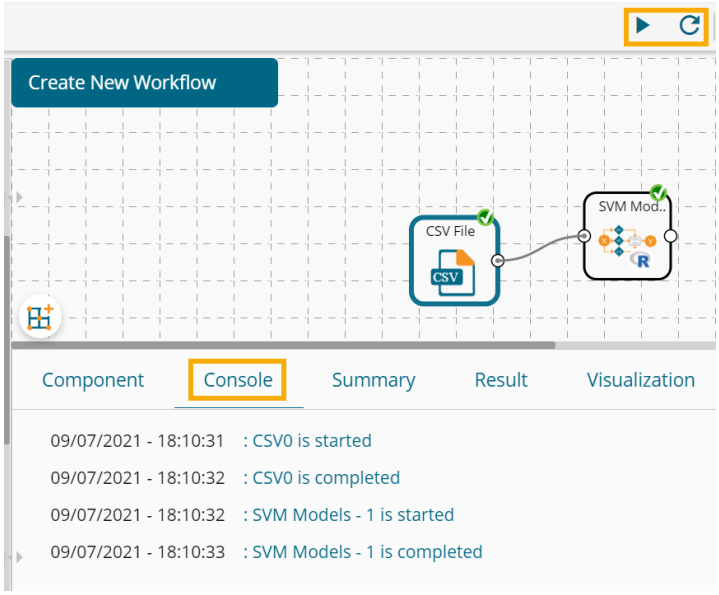
a. Dependent Column
b. Independent Column
c. Type: This field contains the following options:
   i. C-classification
   ii. nu-classification
   iii. eps-regression
   iv. nu-regression
d. Kernel: This field contains the following options:
   i. Linear
   ii. polynomial
   iii. radial
   iv. sigmoid
e. Degree Polynomial
f. Gamma Factor
g. Cost Factor
h. Nu Factor (0.1-0.9)
i. Cross Validation

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

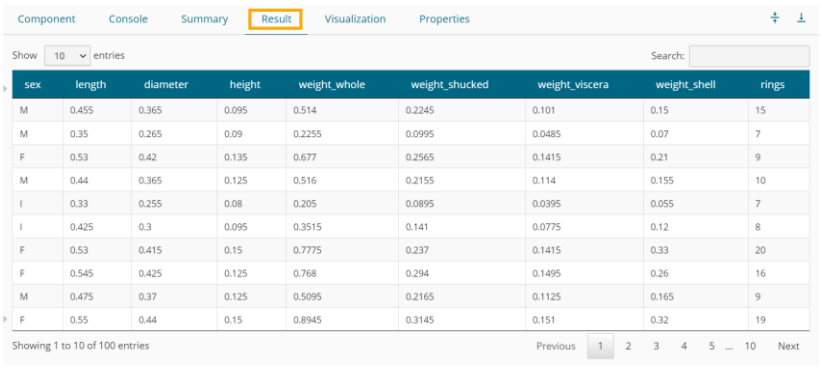
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



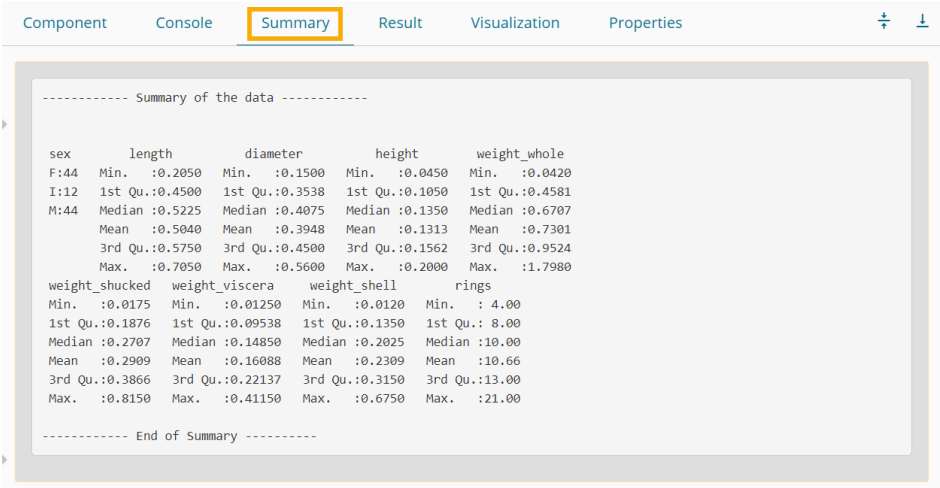
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



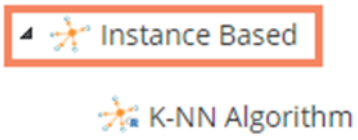
viii) Click the '**Summary**' tab to open the model summary.



13.1.10. Instance Based

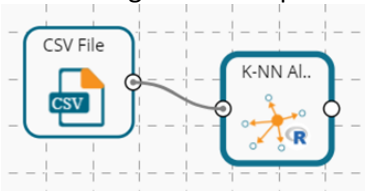
The Instance-based algorithms also known as memory-based algorithms compare new problem instances with the instances already seen in training and have been stored in memory. These algorithms are sometimes referred to as 'Lazy' computation gets postponed until a new instance is observed.

The Data Science Workbench provides the K-Nearest Neighbor under the R Workspace as displayed in the below-given image:



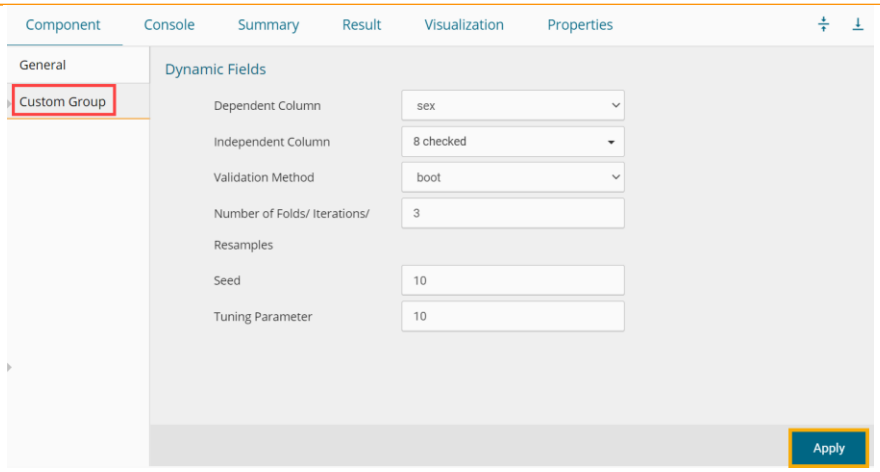
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component.



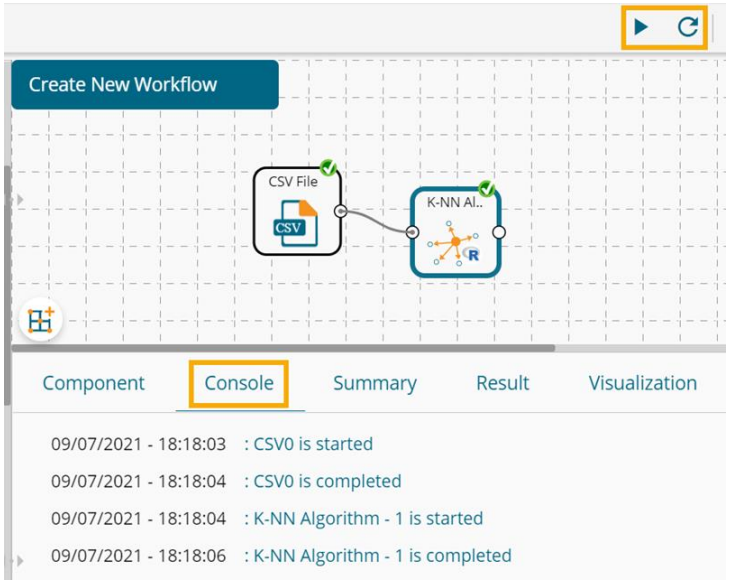
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

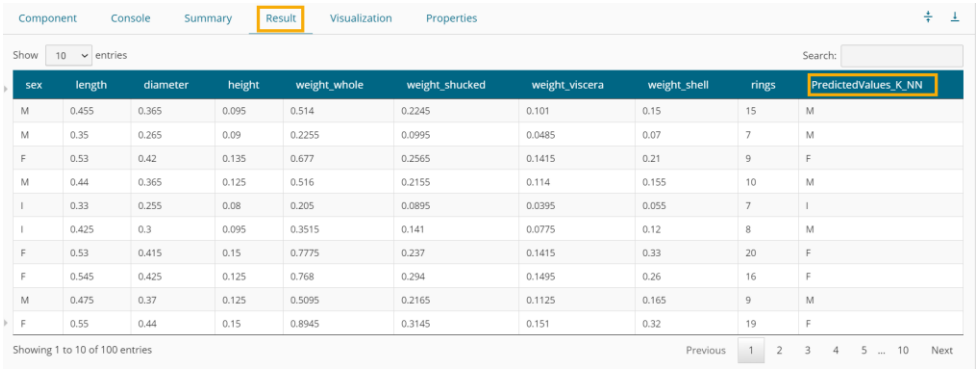
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



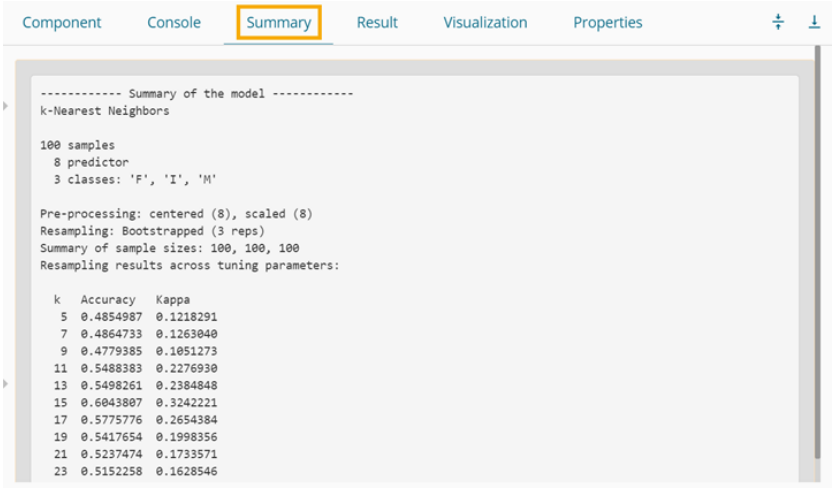
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

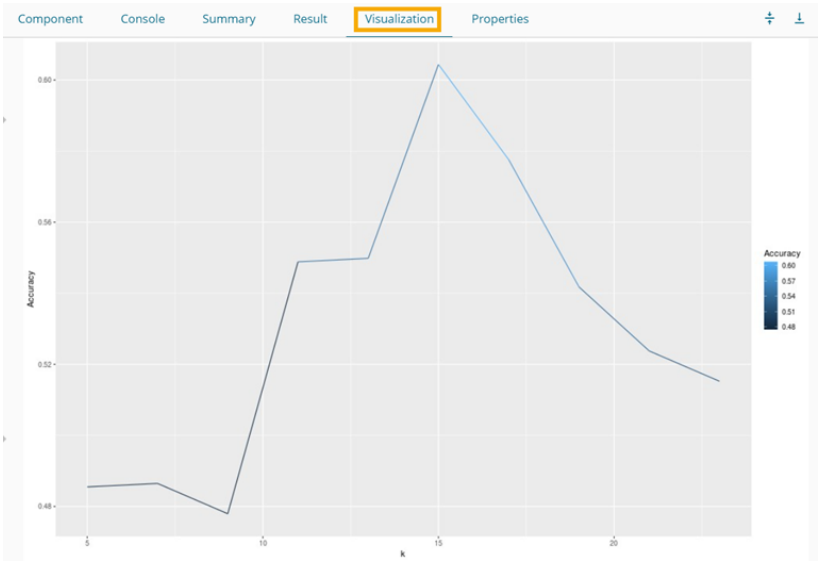
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.

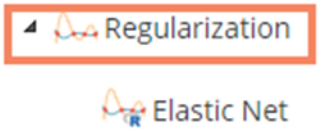


ix) Click the '**Visualization**' tab to open the result data through a chart.



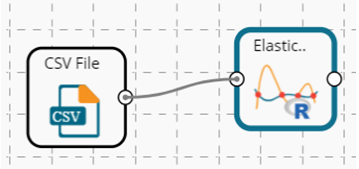
13.1.11. Regularization

The Regularization tree node contains the Elastic Net algorithm as shown in the following image:



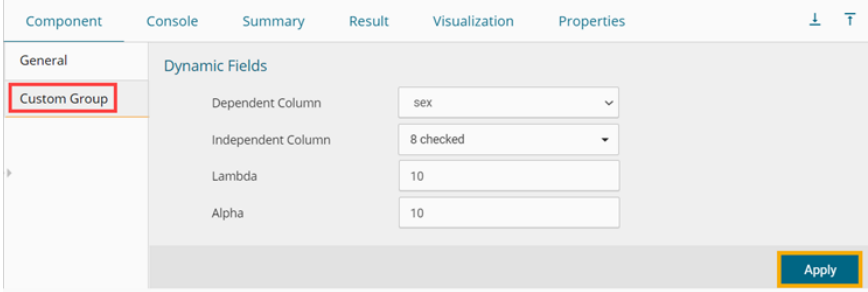
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component.



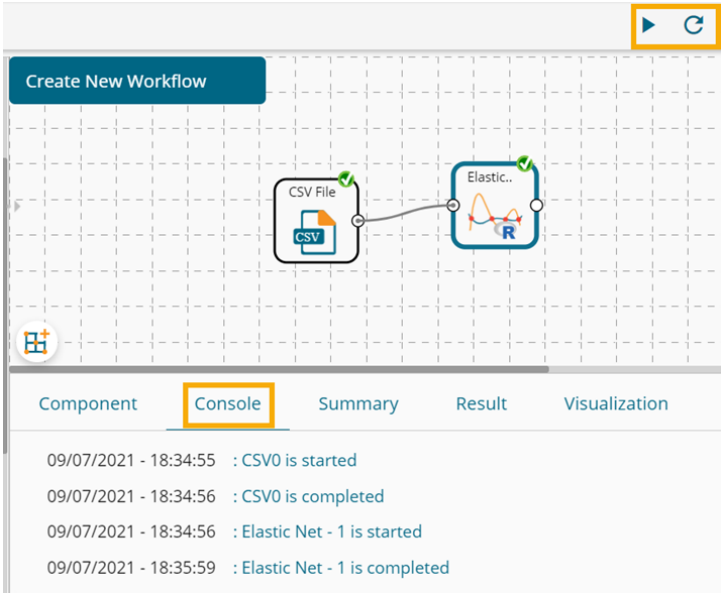
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a. Dependent Column
   b. Independent Column
   c. Lambda
   d. Alpha

iv) Click the '**Apply**' option.

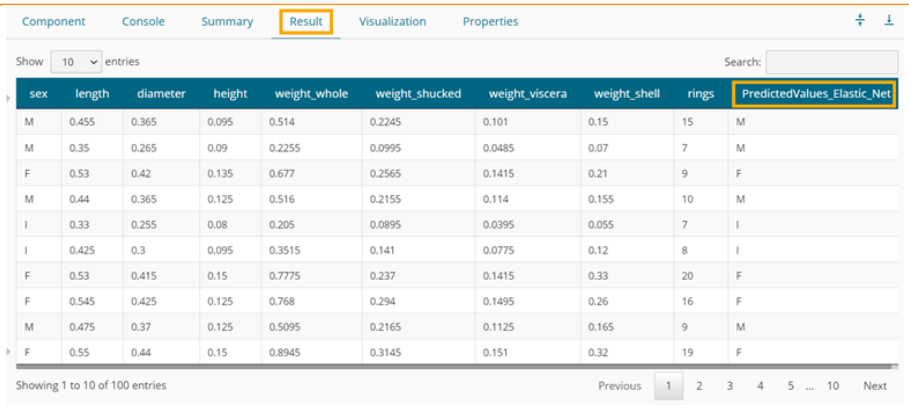


v) Run the workflow after getting the success message.

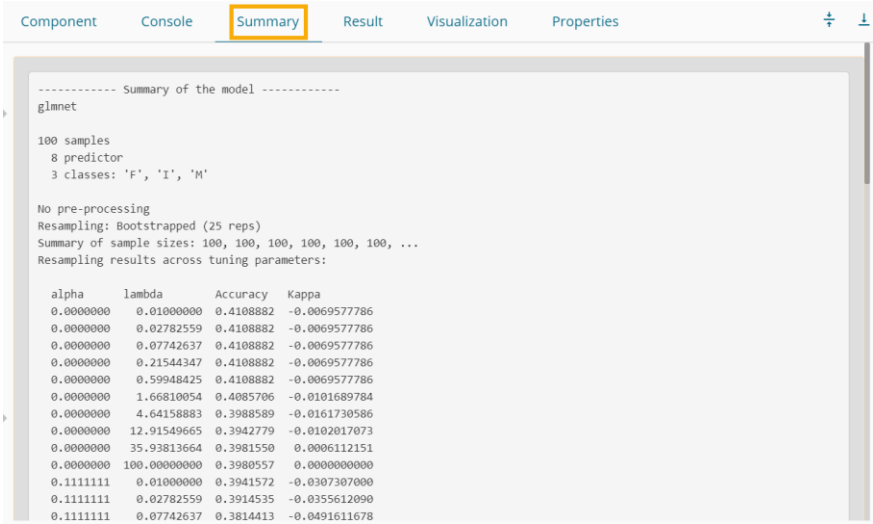
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



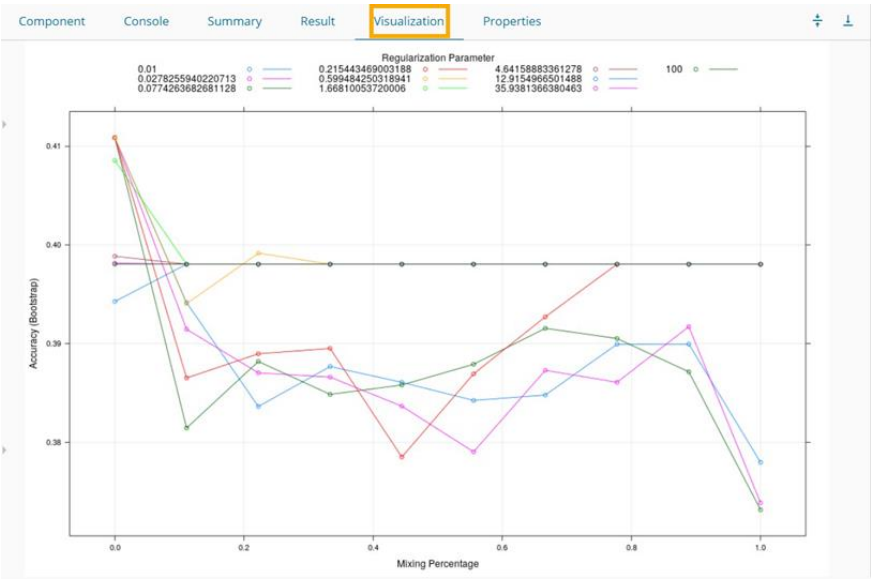
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



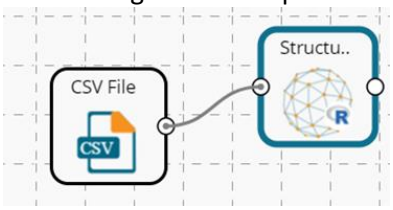
ix) Click the '**Visualization**' tab to see the result data through a chart.



13.1.12. Structural Equation Modelling

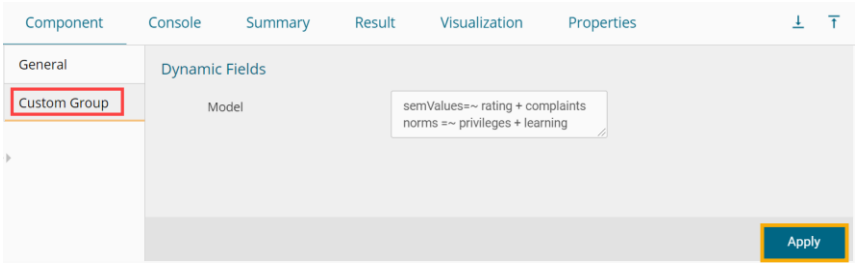
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component.



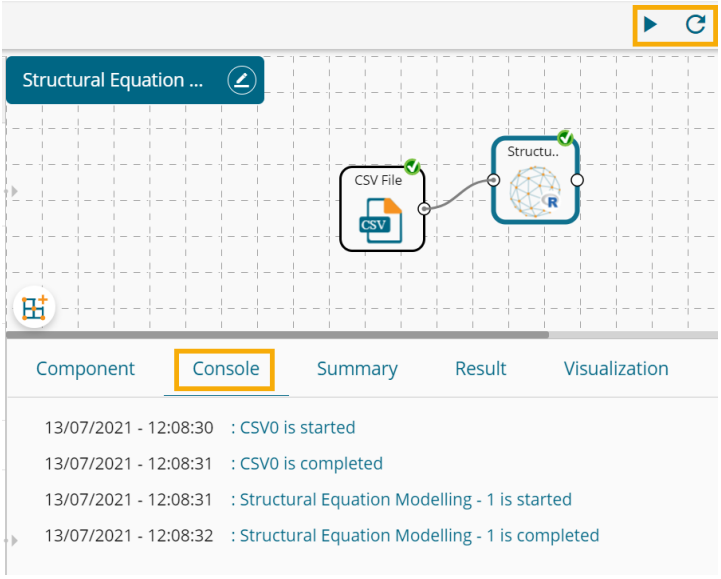
iii) Configure the model field provided under the '**Dynamic Fields**' of the dragged algorithm component.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

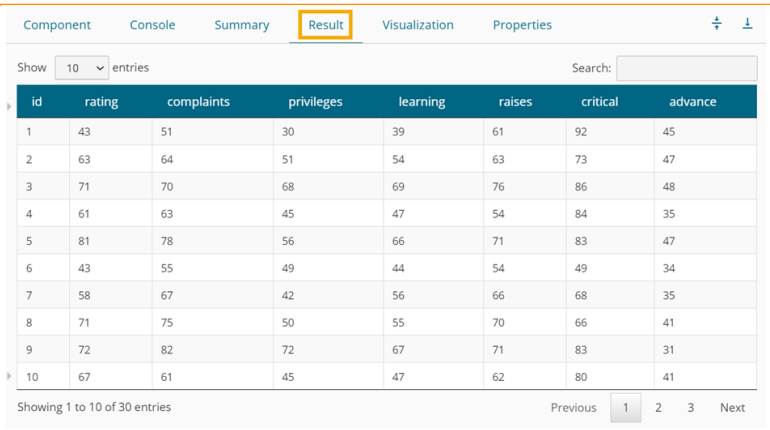
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.



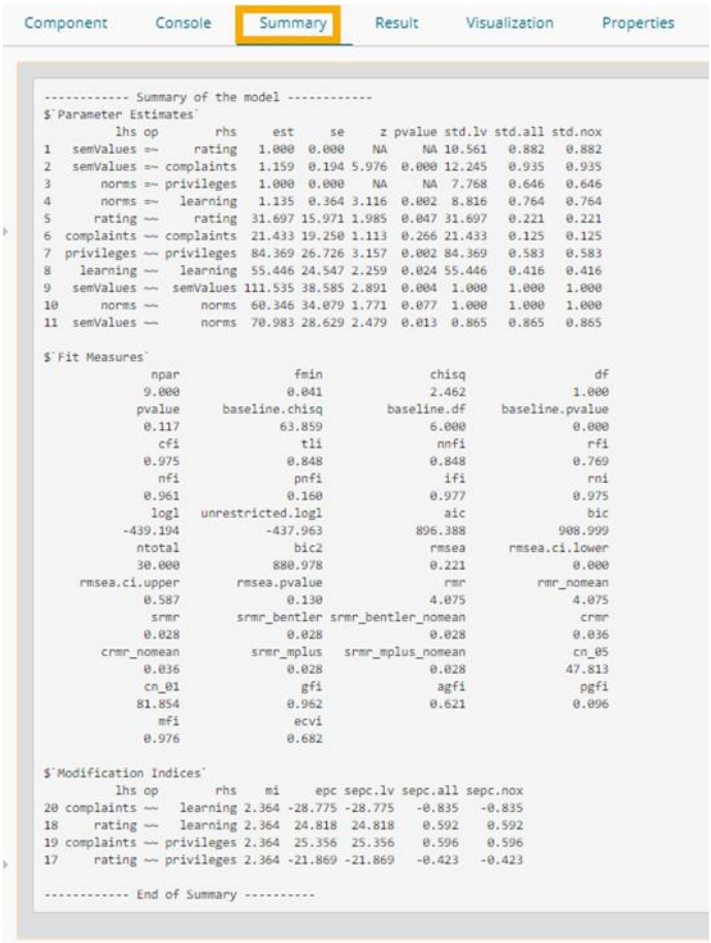
Component | Console | Summary | Result | Visualization | Properties

Show 10 entries Search:

id	rating	complaints	privileges	learning	raises	critical	advance
1	43	51	30	39	61	92	45
2	63	64	51	54	63	73	47
3	71	70	68	69	76	86	48
4	61	63	45	47	54	84	35
5	81	78	56	66	71	83	47
6	43	55	49	44	54	49	34
7	58	67	42	56	66	68	35
8	71	75	50	55	70	66	41
9	72	82	72	67	71	83	31
10	67	61	45	47	62	80	41

Showing 1 to 10 of 30 entries Previous 1 2 3 Next

viii) Click the '**Summary**' tab to open the model summary.



Component | Console | Summary | Result | Visualization | Properties

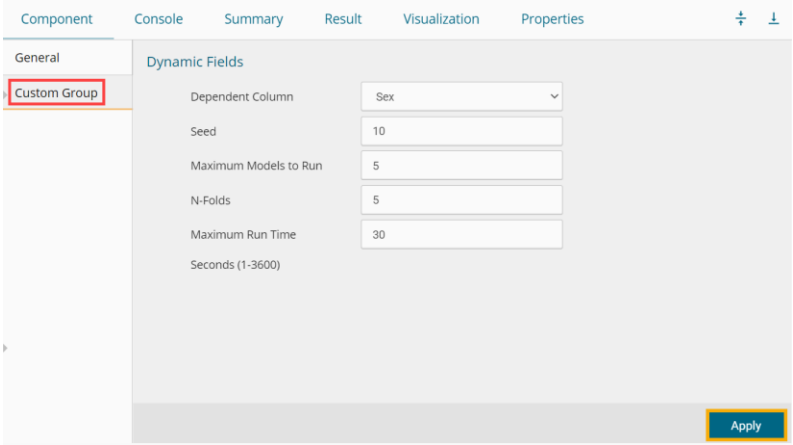
```
------------ Summary of the model ------------
$`Parameter Estimates`
          lhs op       rhs     est     se     z pvalue std.lv std.all std.nox
1   semValues =~     rating   1.000  0.000    NA     NA 10.561   0.882   0.882
2   semValues =~ complaints   1.159  0.194 5.976  0.000 12.245   0.935   0.935
3       norms =~ privileges   1.000  0.000    NA     NA  7.768   0.646   0.646
4       norms =~   learning   1.135  0.364 3.116  0.002  8.816   0.764   0.764
5      rating ~~     rating  31.697 15.971 1.985  0.047 31.697   0.221   0.221
6  complaints ~~ complaints  21.433 19.250 1.113  0.266 21.433   0.125   0.125
7  privileges ~~ privileges  84.369 26.726 3.157  0.002 84.369   0.583   0.583
8    learning ~~   learning  55.446 24.547 2.259  0.024 55.446   0.416   0.416
9   semValues ~~  semValues 111.535 38.585 2.891  0.004  1.000   1.000   1.000
10      norms ~~      norms  60.346 34.079 1.771  0.077  1.000   1.000   1.000
11  semValues ~~      norms  70.983 28.629 2.479  0.013  0.865   0.865   0.865

$`Fit Measures`
               npar                fmin               chisq                  df
              9.000               0.041               2.462               1.000
             pvalue      baseline.chisq         baseline.df     baseline.pvalue
              0.117              63.859               6.000               0.000
                cfi                 tli                nnfi                 rfi
              0.975               0.848               0.848               0.769
                nfi                pnfi                 ifi                 rni
              0.961               0.160               0.977               0.975
               logl   unrestricted.logl                 aic                 bic
           -439.194            -437.963             896.388             908.999
             ntotal                bic2               rmsea       rmsea.ci.lower
             30.000             880.978               0.221               0.000
     rmsea.ci.upper        rmsea.pvalue                 rmr        rmr_nomean
              0.587               0.130               4.075               4.075
               srmr       srmr_bentler srmr_bentler_nomean                crmr
              0.028               0.028               0.028               0.036
       crmr_nomean          srmr_mplus   srmr_mplus_nomean                cn_05
              0.036               0.028               0.028              47.813
              cn_01                 gfi                agfi                pgfi
             81.854               0.962               0.621               0.096
                mfi                ecvi
              0.976               0.682

$`Modification Indices`
          lhs op       rhs    mi     epc sepc.lv sepc.all sepc.nox
20 complaints ~~   learning 2.364 -28.775 -28.775   -0.835   -0.835
18     rating ~~   learning 2.364  24.818  24.818    0.592    0.592
19 complaints ~~ privileges 2.364  25.356  25.356    0.596    0.596
17     rating ~~ privileges 2.364 -21.869 -21.869   -0.423   -0.423

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

13.1.13. Automated ML

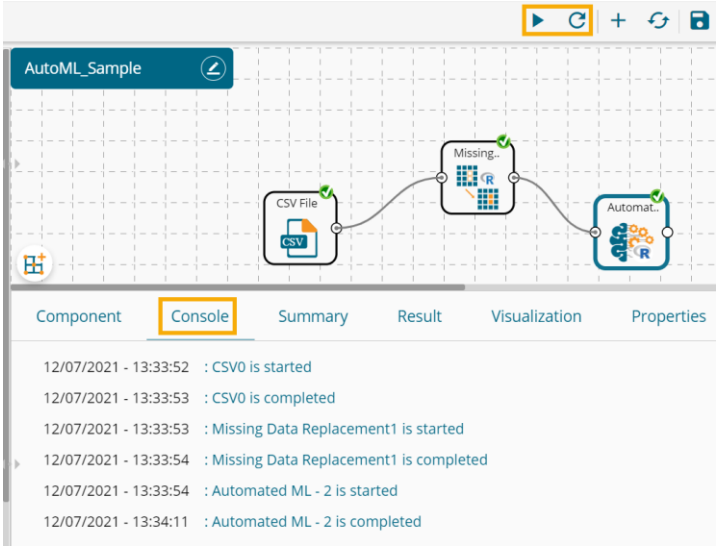
i) Drag the algorithm component to the workspace and connect it with a configured data source (In this case the workflow contains a Missing Value Replacement component to prepare the selected input data).

ii) Click the algorithm component.

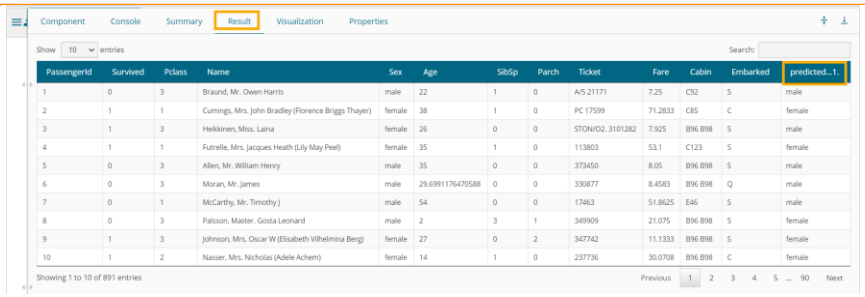
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
iv) Click the '**Apply**' option.



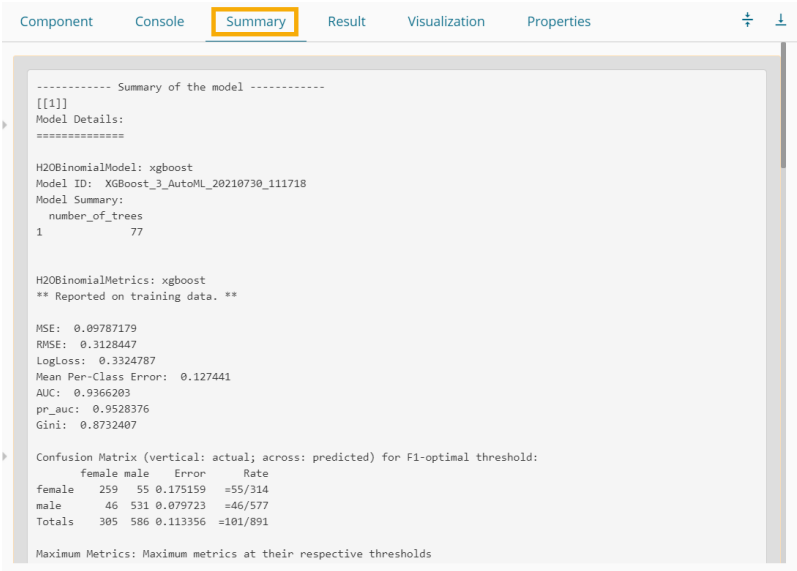
v) Run the workflow after getting the success message.
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.



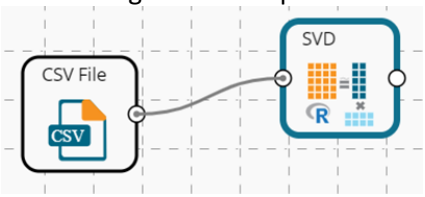
viii) Click the '**Summary**' tab to open the model summary.



13.1.14. SVD

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

ii) Click the algorithm component.

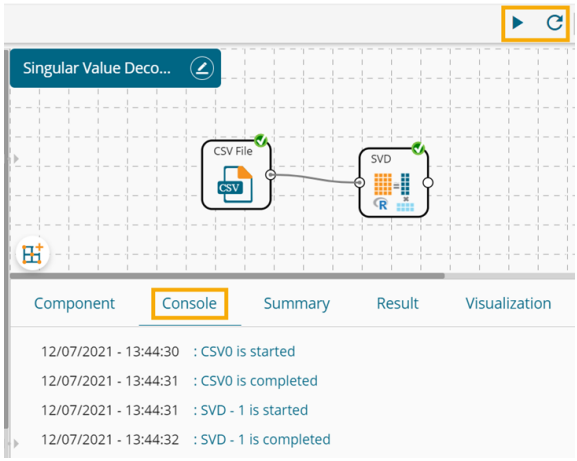


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Click the '**Apply**' option.

v) Run the workflow after getting the success message.

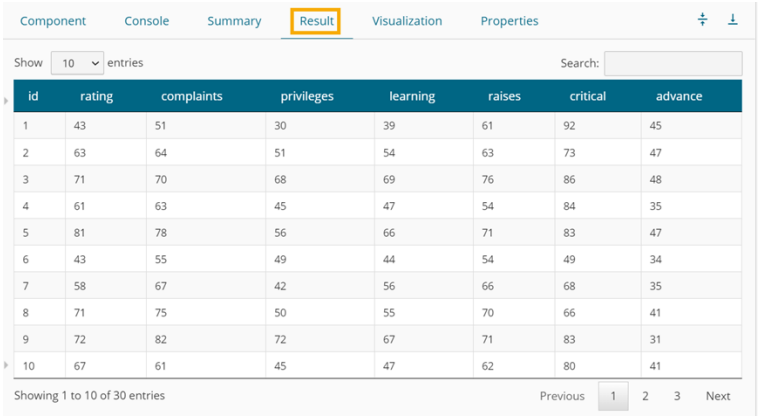
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



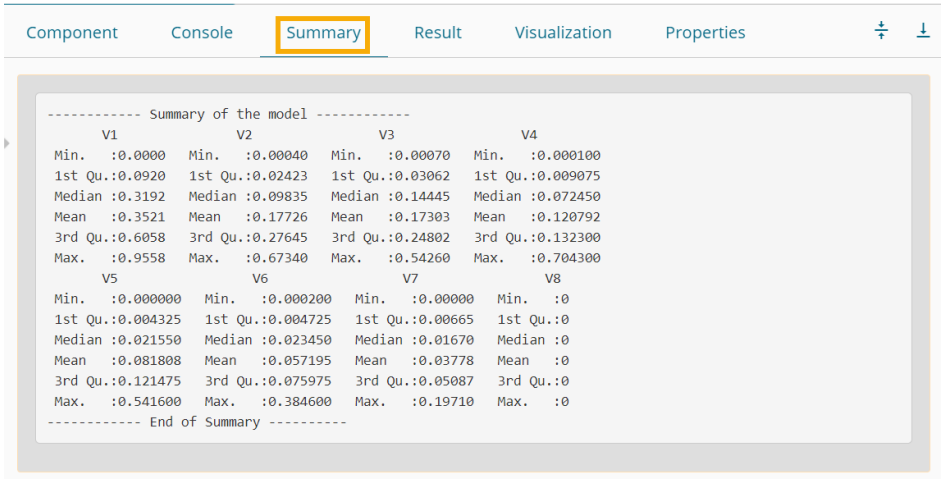
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



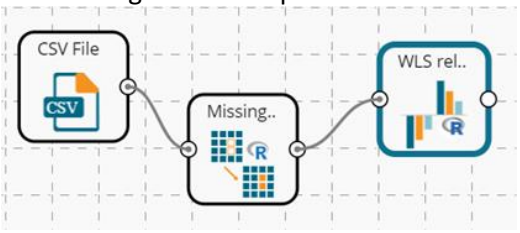
viii) Click the '**Summary**' tab to open the model summary.



13.1.15. WLS Relative Std

i) Drag the algorithm component to the workspace and connect it with a configured data source (In this case the Missing Value Replacement component is connected to prepare the input data).

ii) Click the algorithm component.

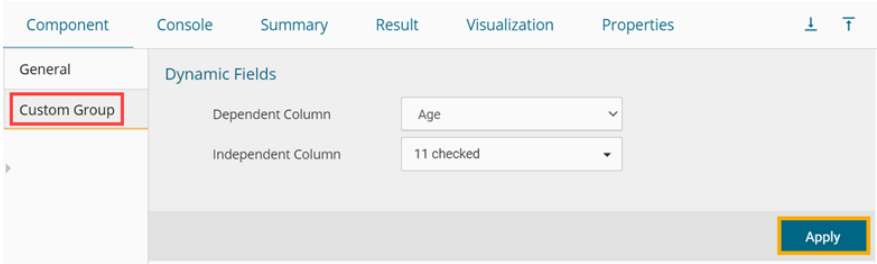


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

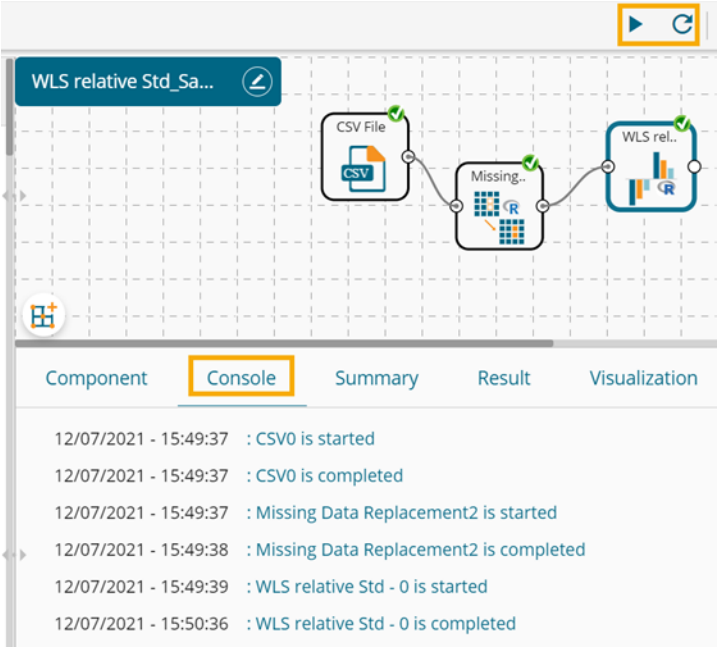
b. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.

iv) Click the '**Apply**' option.



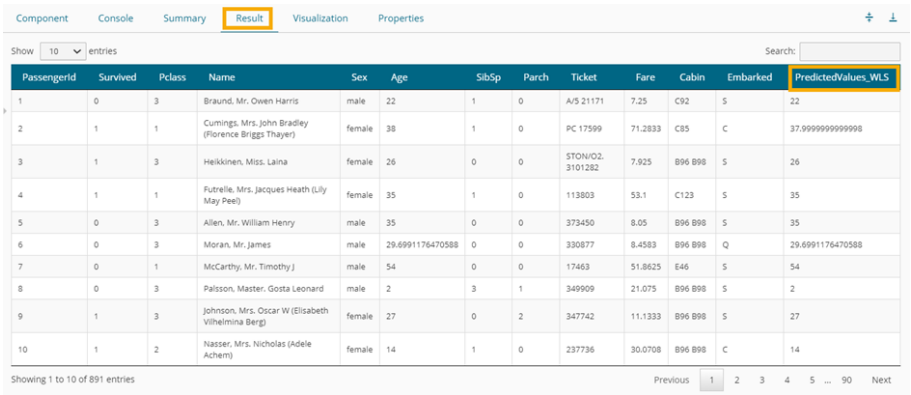
v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

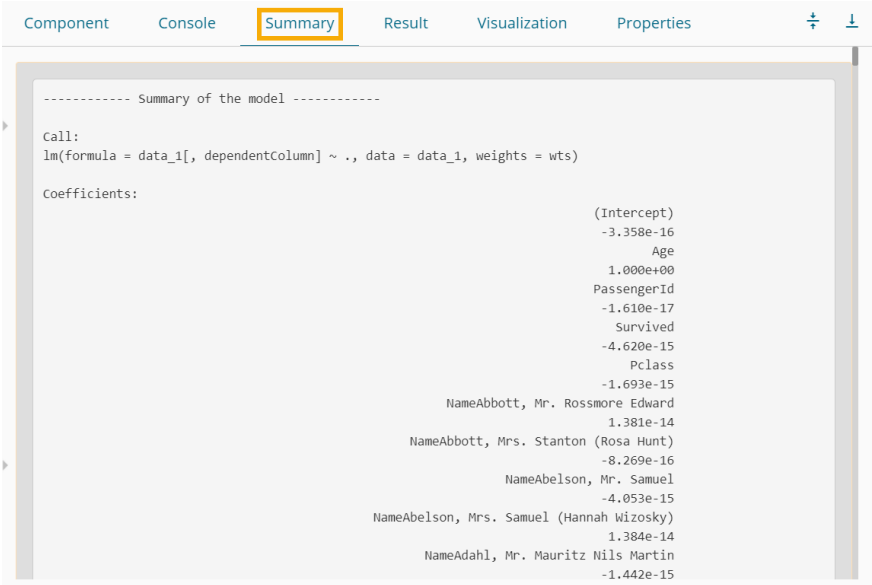


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.

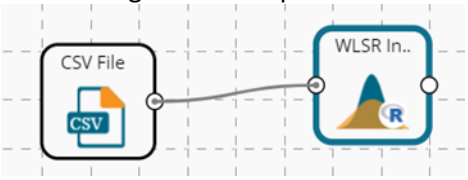


viii) Click the '**Summary**' tab to open the model summary.



13.1.16. WLSR Input Weights

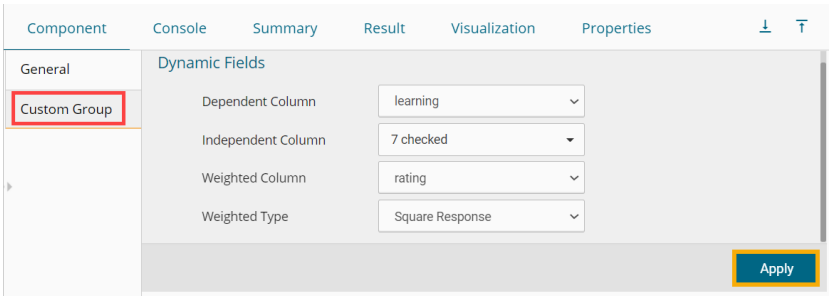
i) Drag the algorithm component to the workspace and connect it with a configured data source.
ii) Click the algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a. **Dependent Column**: Select the target column on which the regression analysis gets applied.
   b. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.

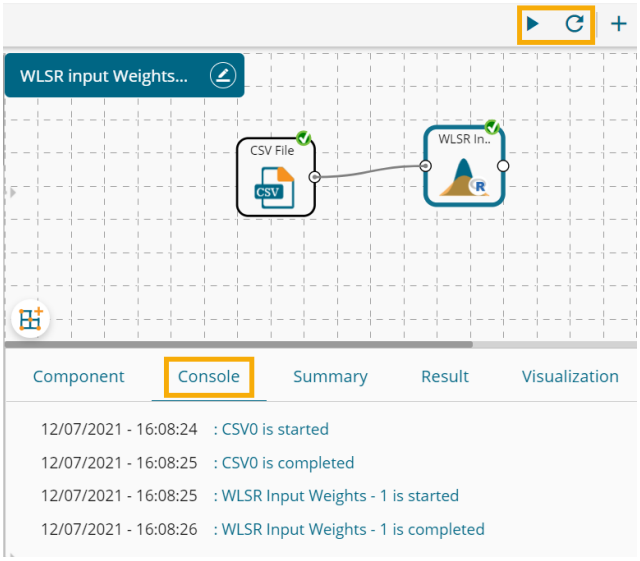
c. **Weighted Column:** Select a column from the input dataset.
d. **Weighted Type:** Select a Weighted Type out of the given options that are Square Response and Normal Response.

iv) Click the '**Apply**' option.



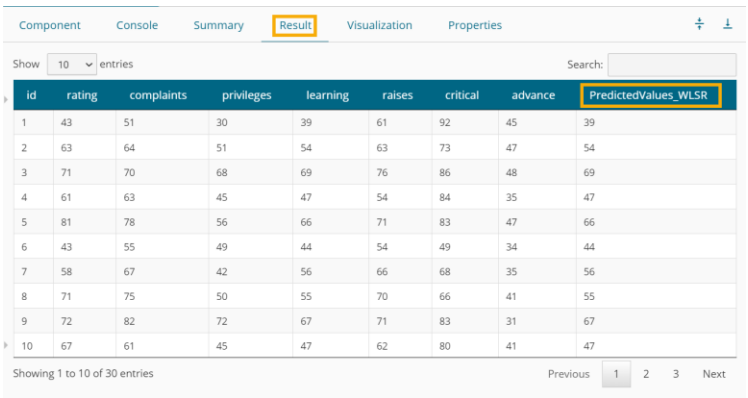
v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

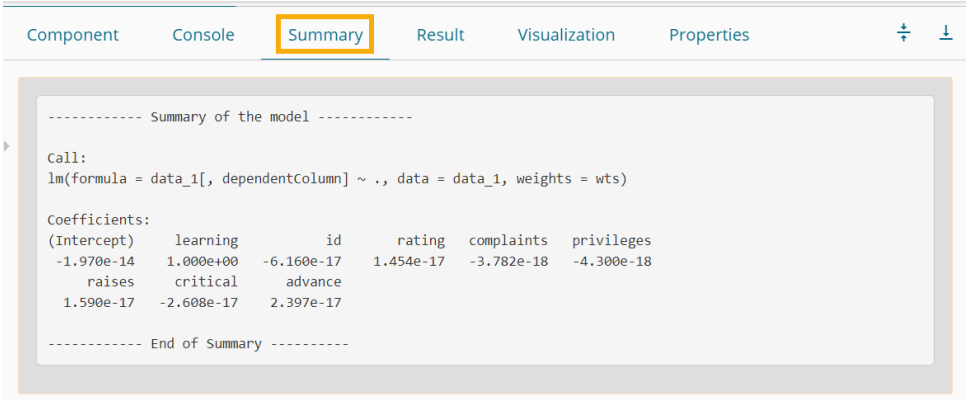


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.



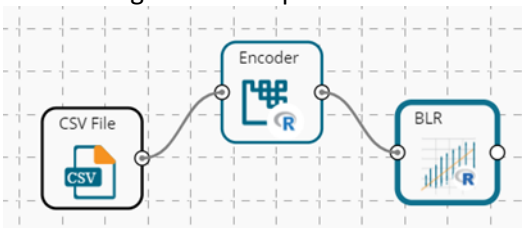
viii) Click the '**Summary**' tab to open the model summary.



13.1.17. BLR

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

ii) Click the algorithm component.

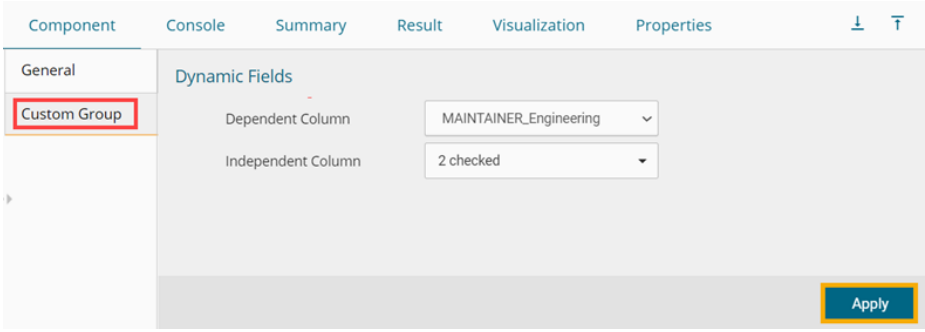


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

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

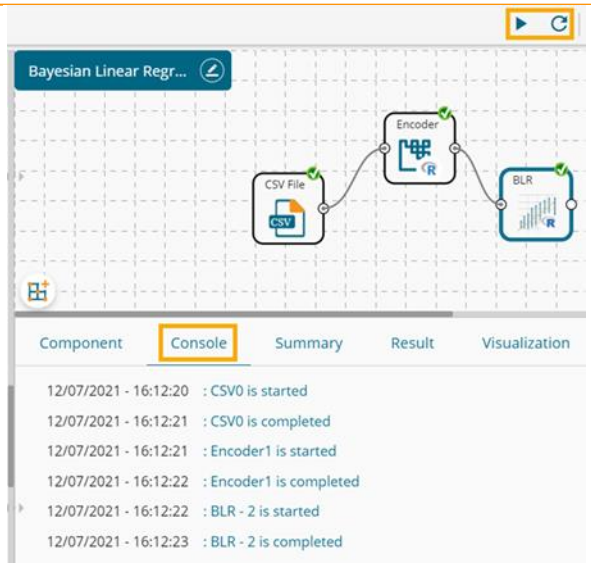
b. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

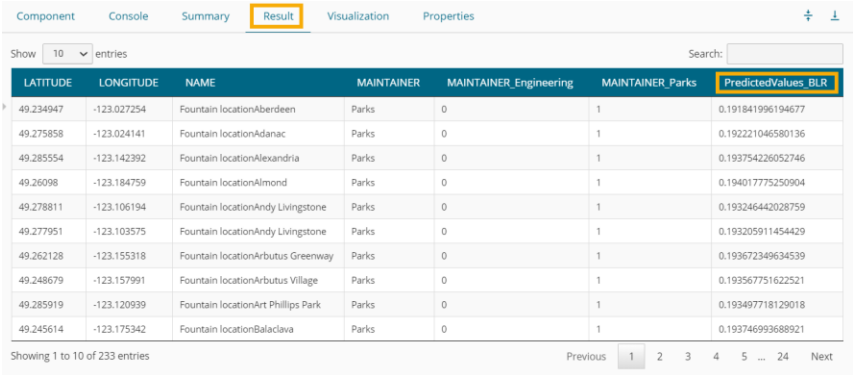
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.

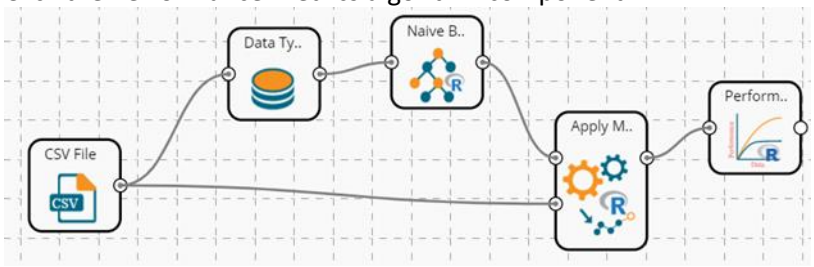


Note: No Summary is available for this component node.

13.1.18. Performance Metrics

i) Drag the Performance metrics component and connect it to the workflow containing an Apply Model component as displayed in the image given below.

ii) Click the Performance Metrics algorithm component.



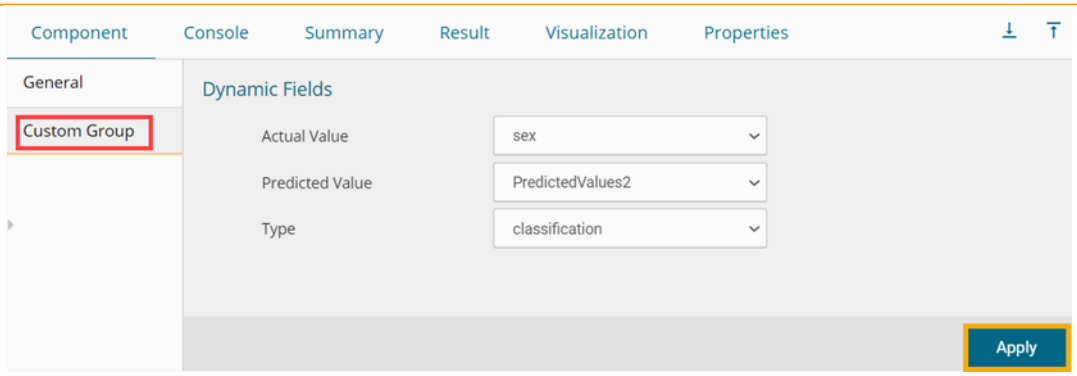
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. Actual Value: Select a column from the input data to find the actual value.

b. Predicted Value: Select a column from the input data to get the predicted value.

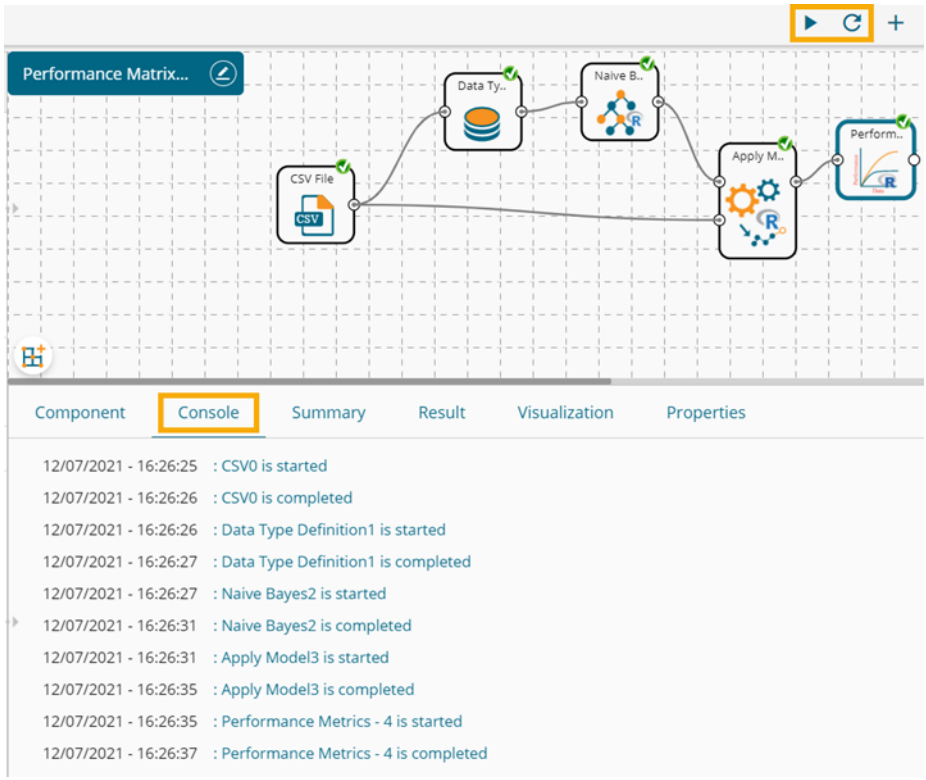
c. Type: Select a type out of Classification or Regression options.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

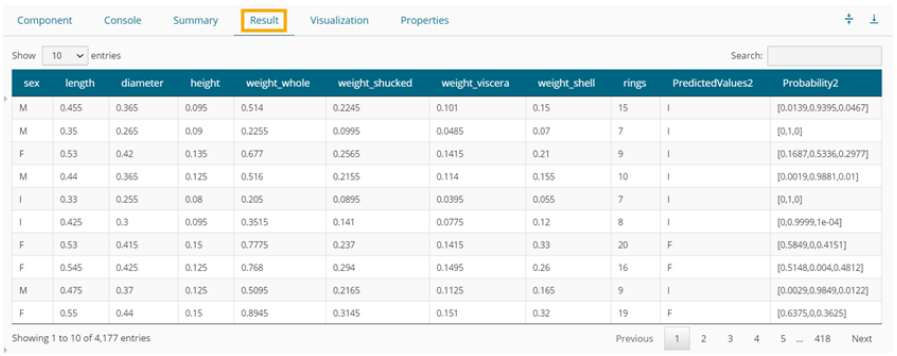
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



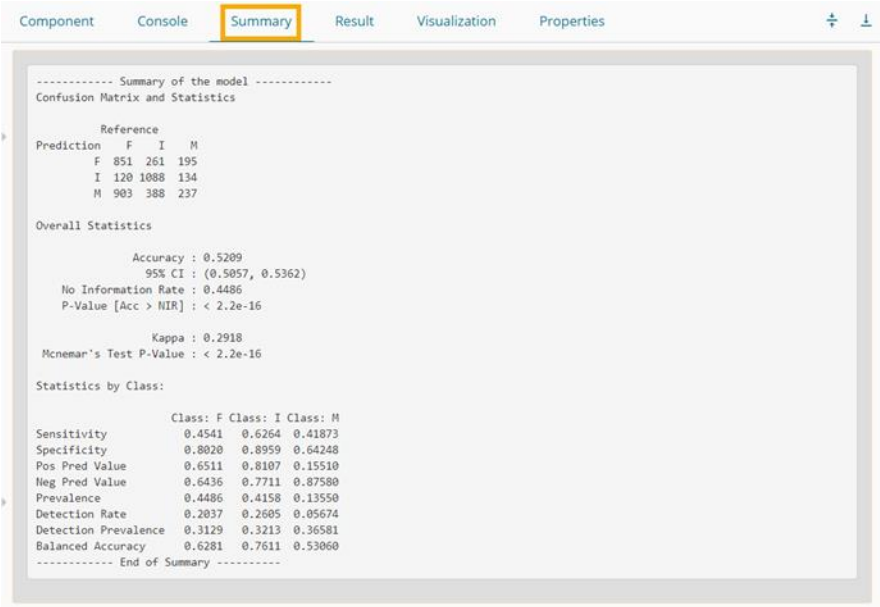
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



13.2. Apply Model

This component is provided to generate predictions based on the trained model. The user can view the predicted column value and probability of each label class by using the Apply Model component.

The user can create a model in the following ways:

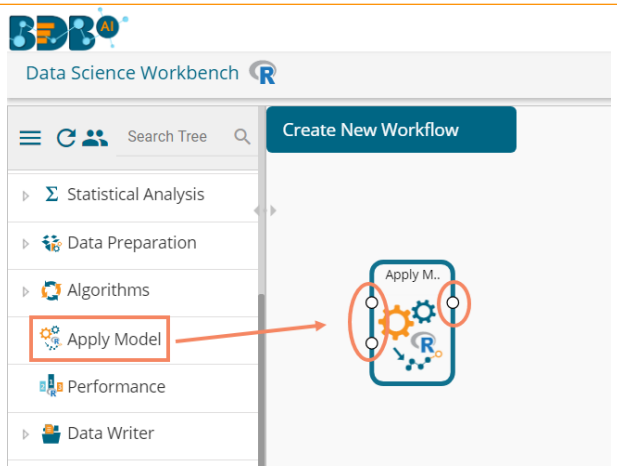
- Generate a model using an algorithm
- Generate a model using the saved models

The 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

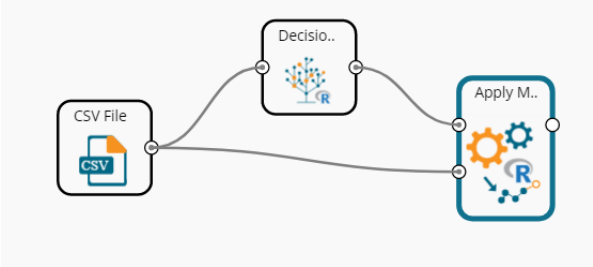
The Apply Model component provided under R, Python, and Spark can be configured using the same set of steps within the respected Workbenches, so this component is only described for R.

i) Drag the '**Apply Model**' component to the workspace.
ii) The Apply Model has two input components and one output component.



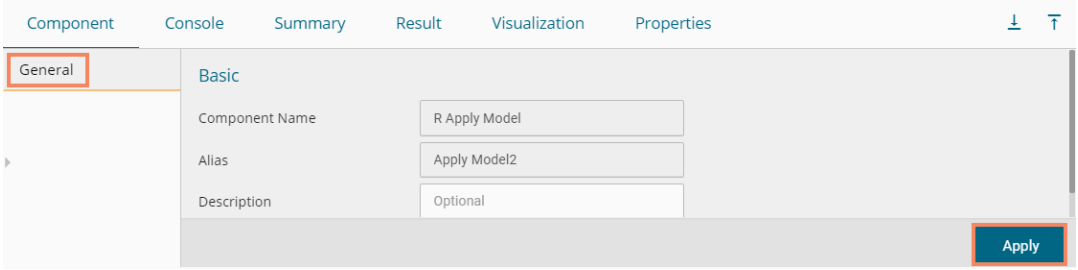
iii) Connect the Apply Model component with a valid combination of Data source and algorithm (Configure the data source and algorithm components. In this case, the used algorithm is Decision Tree.)

iv) Click the '**Apply Model**' component.



v) Basic component details get displayed.
   a. Component Name: It displays the predefined name of the component
   b. Alias Name: It displays a predefined name that suggests the component's position in the workflow

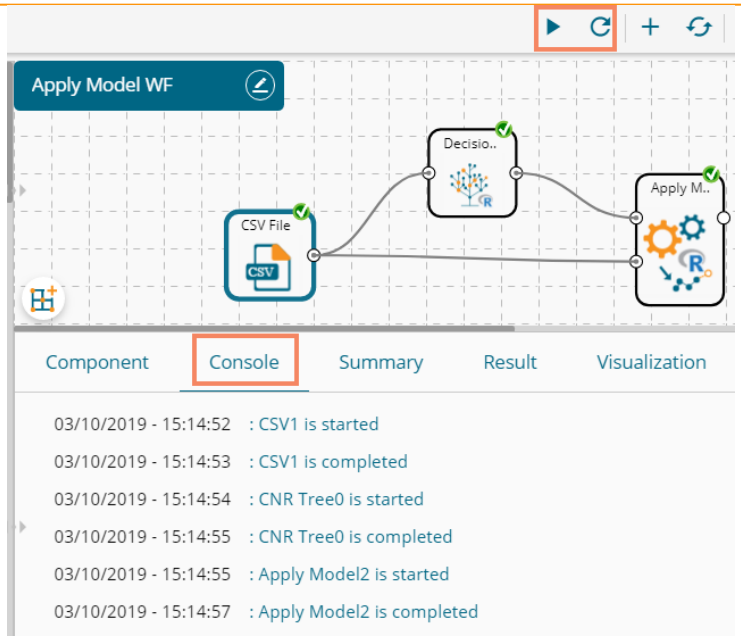
vi) Click the '**Apply**' option.



Note: The number given to the Apply Model signifies its place in the workflow. E.g., R Apply Model2 in the below-given image suggests that it is in the third position in the workflow.

vii) Run the workflow.

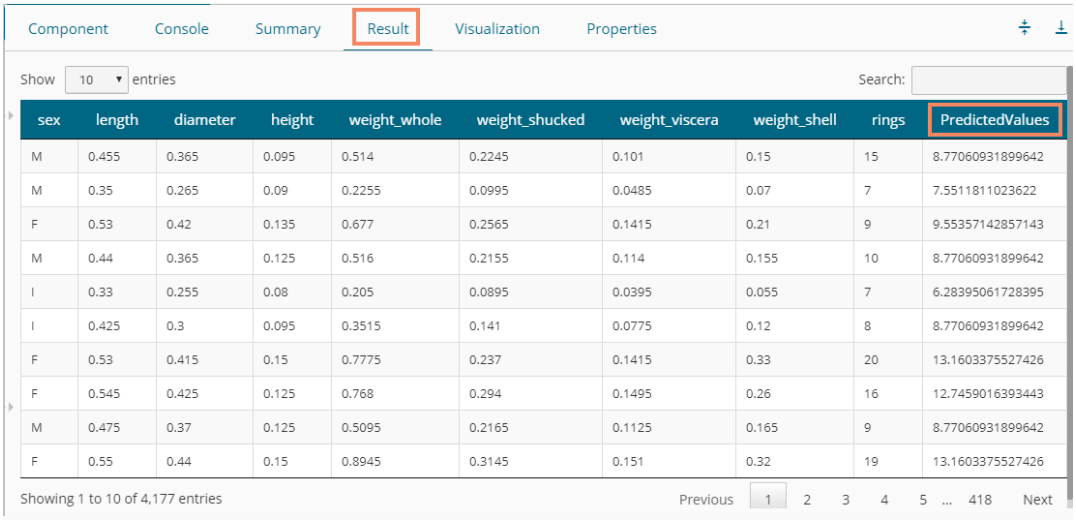
viii) The '**Console**' tab opens displaying the progress of the process. Completion of the console process gets marked by the green checkmarks on the top of the dragged components.



ix) Follow the below-given steps to display the Result view:

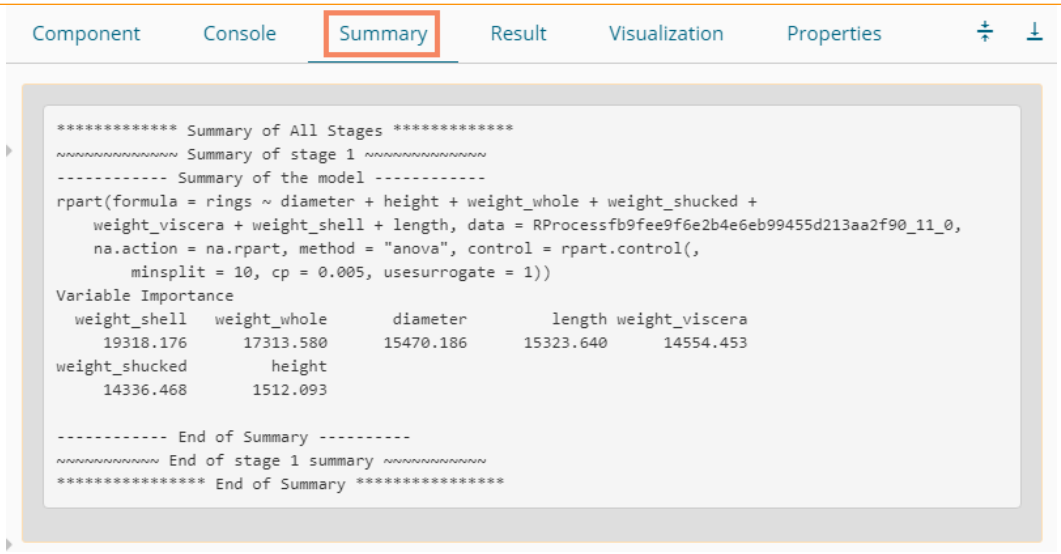
a. Click the dragged R Apply Model component on the workspace.

b. Click the '**Result**' tab.



sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	8.77060931899642
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	7.5511811023622
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	9.55357142857143
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	8.77060931899642
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	6.28395061728395
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8	8.77060931899642
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20	13.1603375527426
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16	12.7459016393443
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9	8.77060931899642
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19	13.1603375527426

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



Note:

a. The Result dataset of the model can be written to a database using a Data Writer.
b. Column header and data type of feature column for both the saved model and testing data should match. If column headers and data types do not match, an alert message gets displayed.
c. It is not mandatory for the testing data set to contain a label column.

13.3. Performance

The user can evaluate model performance through a list of parameters using the performance component. The user can use the R Performance components only for the classification algorithms.

The Performance component is provided as a leaf node under the Performance tree node. It contains 3 input nodes that can be used to compare up to 3 models. Each node has a static name like model_0, model_1, and model_2. Based on the connection to the node model, the summary can be viewed with respective names.

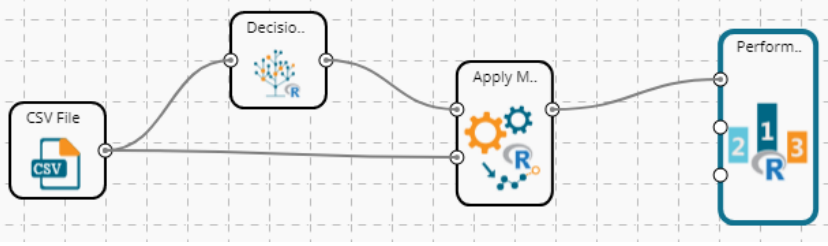
The performance component can be of the following formats:

1. Binary Classification: Used when the label has two classes
2. Multi Classification: Used when the label has 3 or more beta values
3. Regression Metrics: Used when the regression algorithm is used in the workflow

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

Steps to Connect a Performance component (to a model)

i) Drag the Performance component to the workspace and connect to a valid workflow (In this example, a workflow created with the Decision Tree algorithm has been used).



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

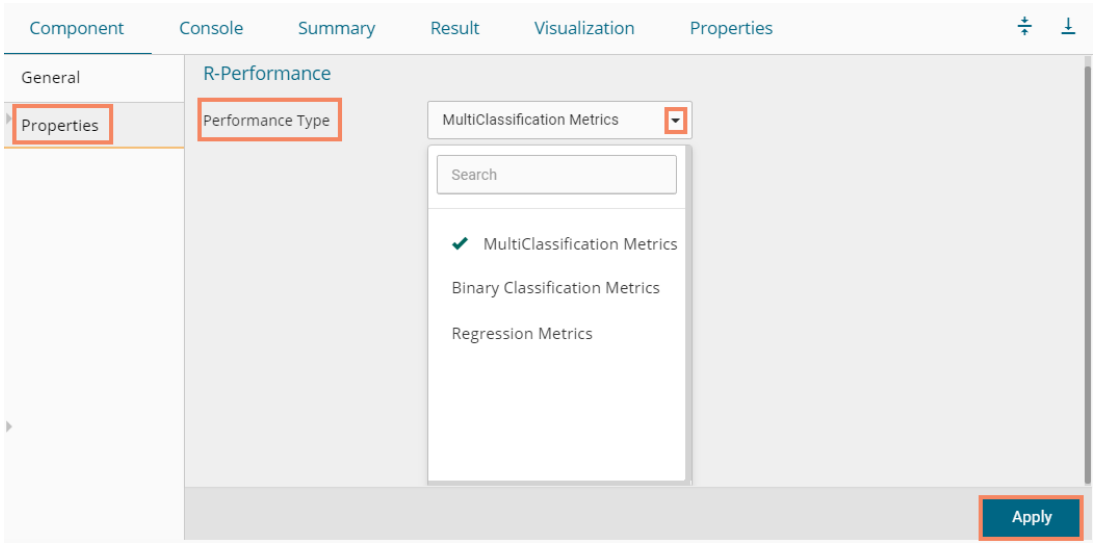
a. **Performance Type**: Select an option using the drop-down menu.

i. Binary Classification: Use this option when the label has two classes.

ii. MultiClassification Metrics(Default option): Use this option when the label has 3 or more beta values.

iii. Regression Metrics: Use this option when the Apply model in the workflow is trained using the Regression Algorithm.

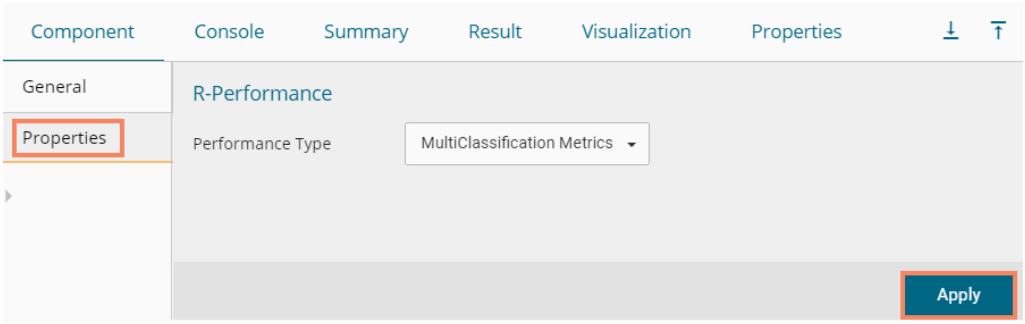
iii) Click the '**Apply**' option.



The user gets different outcomes based on the selected Performance types as described below:

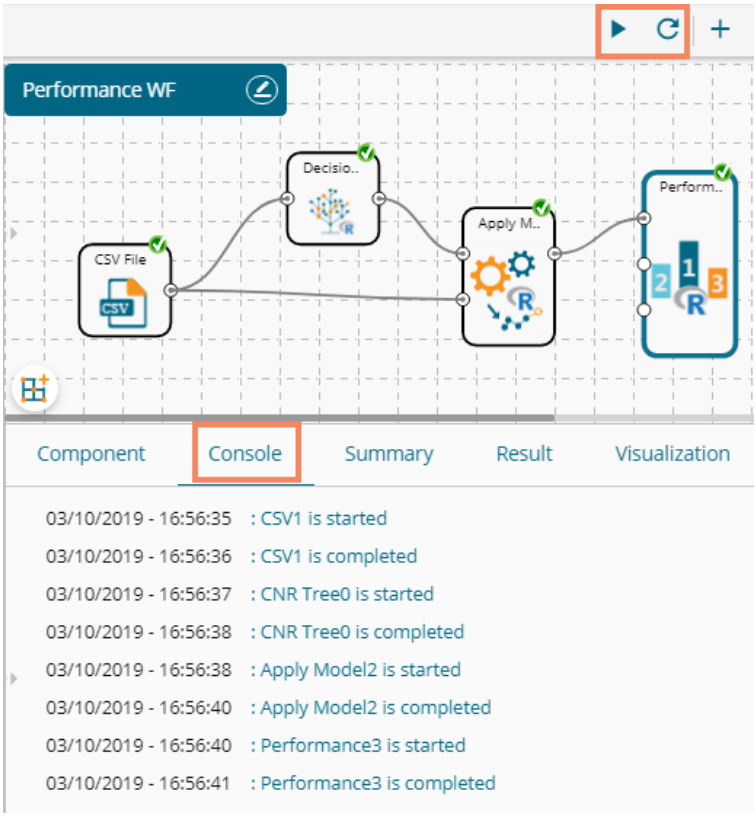
- **Multi Classification Metrics**

1. Navigate to the '**Properties**' tab of the R-Performance component.
2. Select the '**Multi-Classification Metrics**' Performance type via the drop-down list.
3. Click the '**Apply**' option.



4. Run the workflow.

5. The '**Console**' tab opens, displaying steps of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



6. The user can view the summary by clicking the '**Summary**' tab (First click the performance component and then click on the '**Summary**' tab).

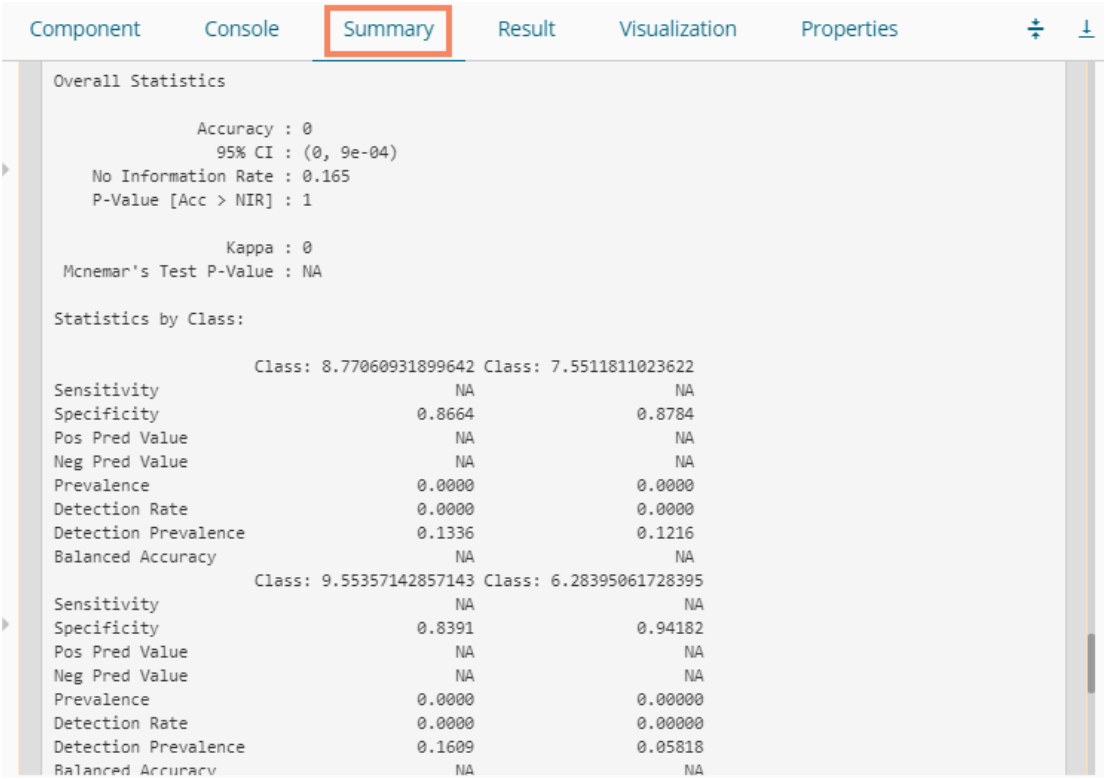
   The following details get displayed by clicking on the '**Summary**' tab:

   a. Confusion Metrix and Statistics
      i. The Confusion Matrix of each model gets displayed.
      ii. The column consists of Actual labels and the row consists of Predicted labels.

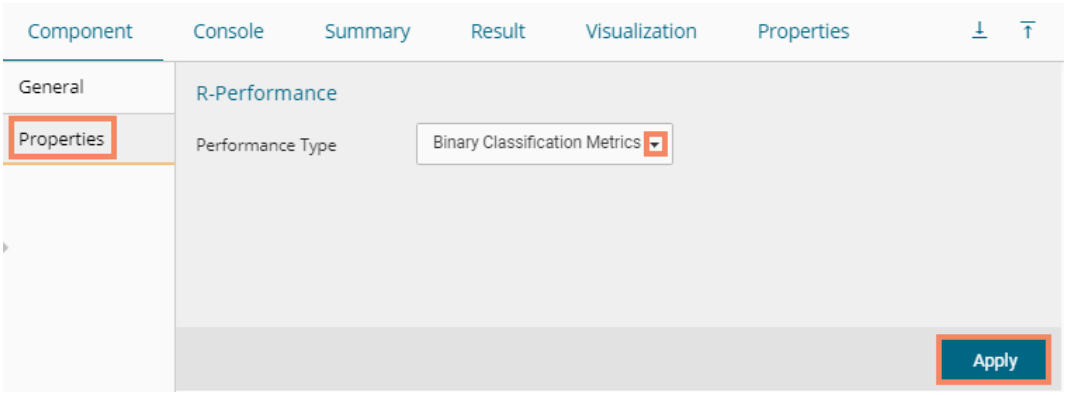
   b. Overall Statistics
      i. Overall statistics of each model can be viewed in a tabular format
      ii. Each model displays the following statistics columns
         1. Accuracy
         2. 95% CI
         3. No Information Rate
         4. P-value
         5. Kappa
         6. Mcnemar's Test P-Value

   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

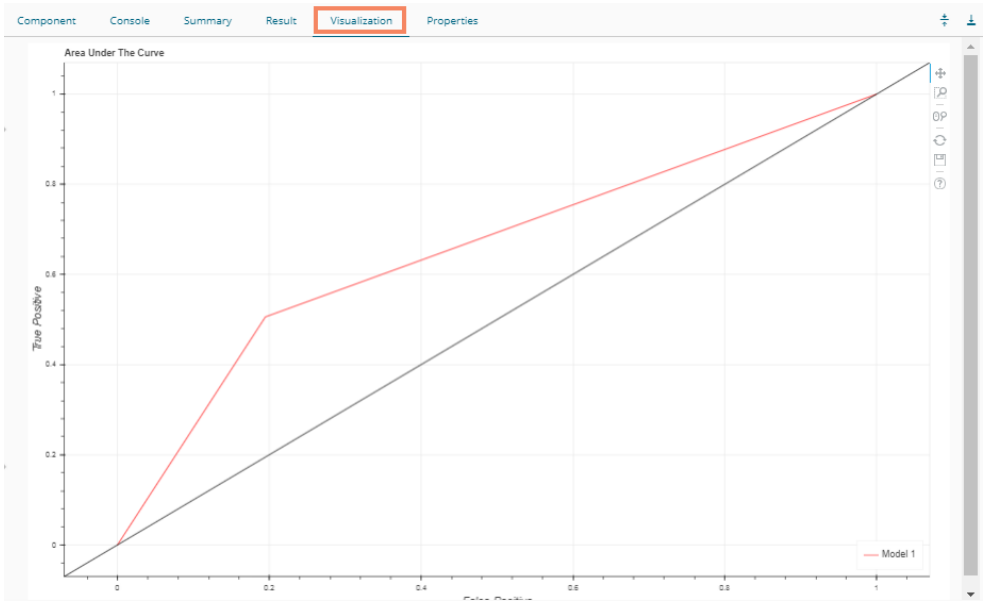


- **Binary Classification Metrics**
  1. Navigate to the '**Properties**' tab of the R-Performance component.
  2. Select the '**Binary Classification Metrics**' Performance type via the drop-down menu. (Select columns with binary attributes from the dataset).
  3. Click the '**Apply**' option.

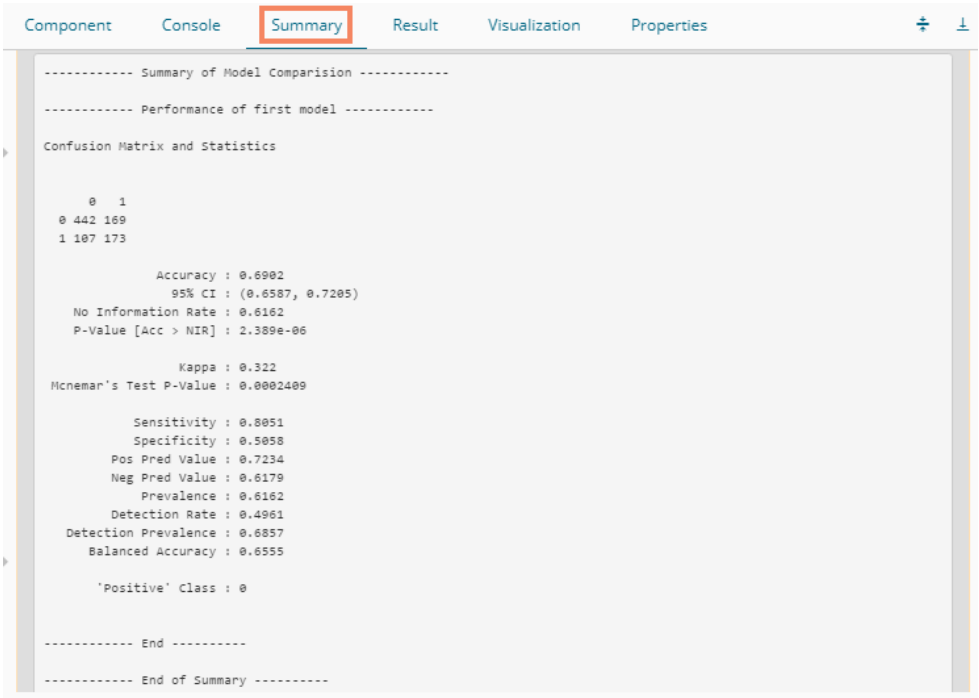


4. Run the workflow.
5. The '**Console**' tab opens, displaying the steps of the process, and the completion gets marked by the green checkmarks on the top of the dragged components.

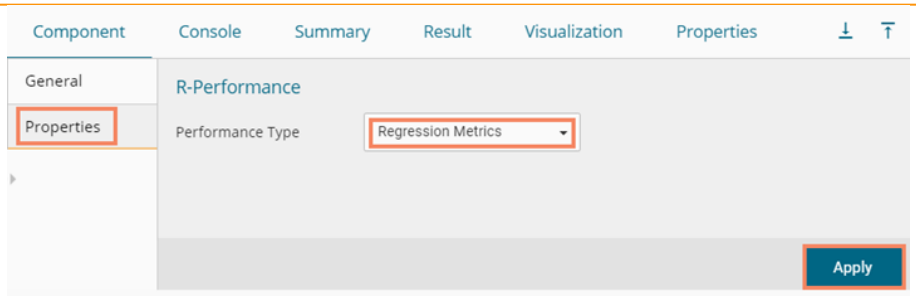
6. Click the '**Visualization**' tab to see the graphical representation of the process data (No data displays under the '**Result**' tab for the Binary Classification Metrics).



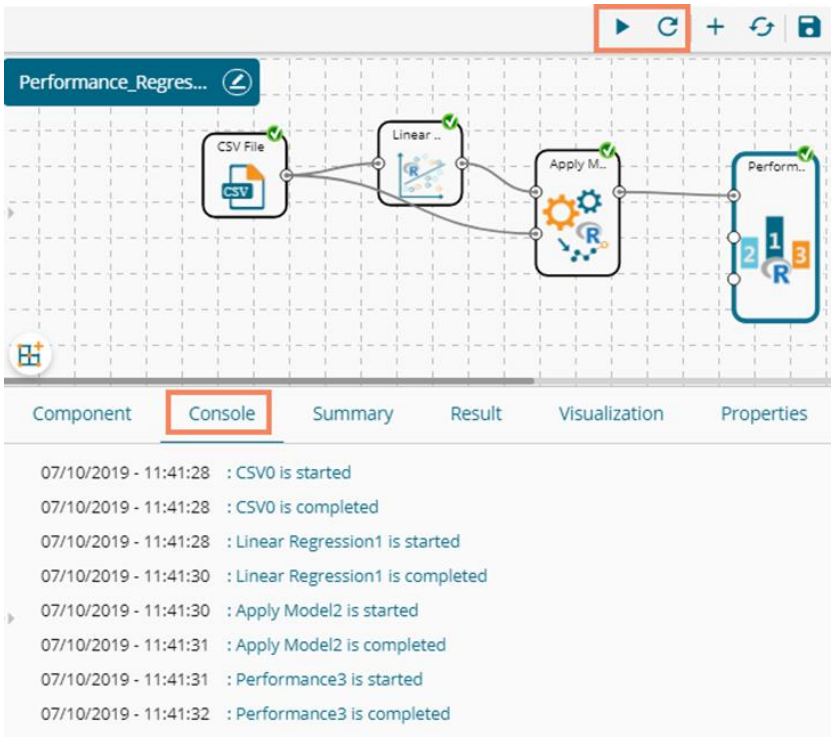
7. Click the '**Summary**' tab to see the model comparison summary.



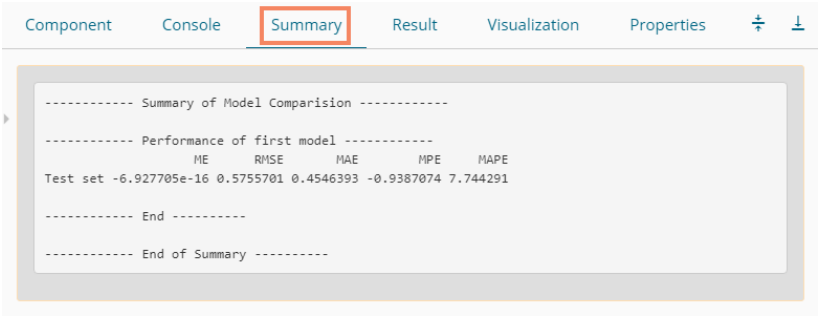
- **Regression Metrics**
  1. Navigate to the '**Properties**' tab of the R-performance component.
  2. Select the '**Regression Metrics**' Performance Type via the drop-down menu. (Make sure that the workflow chosen for Performance check has Regression Algorithm).
  3. Click the '**Apply**' option.



4. Run the workflow.
5. The console tab gets displayed with steps of the process completion. The process completion is also suggested through the green marks on the top of the dragged components.



6. Click the '**Summary**' tab to view the model comparison summary.



Note:

a. In the case of multiple models, all the model statistics get displayed in the summary tab of the performance component (up to 3 models can be compared).
b. The '**Result**' tab for Binary Classification Performance (Binary Classification)

13.4. Custom Scripts (R Scripts)

The user can create and add customized R algorithm components by using the '**Custom Scripts**' component. The created scripts get stored in the '**Saved Scripts'** option.

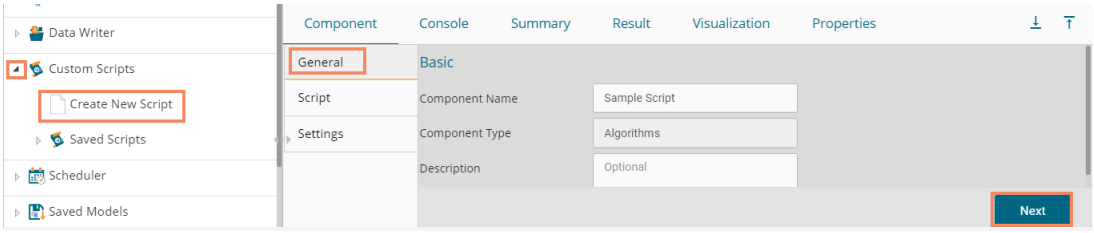
13.4.1. Creating a New Script

i) Click the '**Custom Scripts'** tree node from the tree menu.
ii) Click the **'Create New Script'** component.
iii) The '**General**' tab opens, displaying the Basic information for the script component.

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 gets displayed in this field.
iii. **Description:** Describe the Component (It is an optional field).

iv) Click the **'Next'** option.

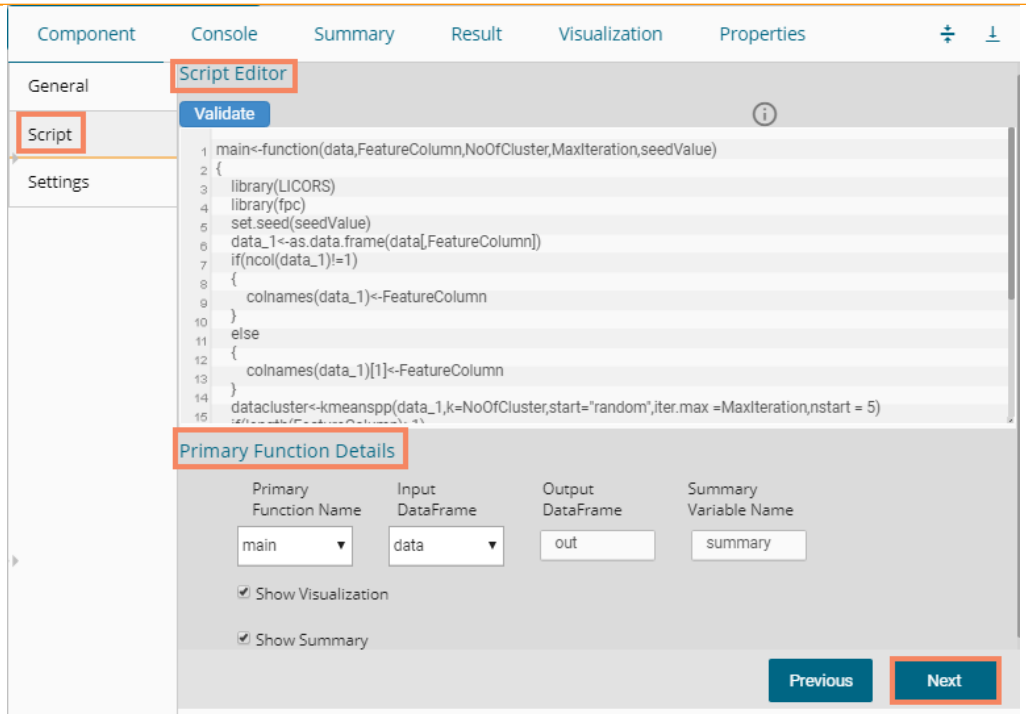


v) The '**Script**' tab opens.
vi) Provide the following information as required:

a. Script Editor

i. Provide a relevant script in the given space on the **'Script Editor'** page.
ii. Click the '**Validate'** option.
iii. Configure the **'Primary Function Details'** to embed the customized script into the function.
   1. **Primary Function Name:** Select the name of the created functio**n** 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 gets passed.
   4. **Model Variable Name:** Enter the output model variable (This field appears only when the model summary has been enabled).
iv. If you need a Visualization chart for ensuring data, tick the '**Show Visualization**' checkbox.
v. If you need to show the summary, tick the **'Show Summary'** checkbox.

vii) Click the '**Next'** option.



viii) The '**Settings**' tab opens.

ix) Configure the following fields:

a. Output Table Definition

The Output TableDefinition option helps to configure some output columns, column headers, and data types.

i. **Consider all columns from the previous component:** To display all columns of the prior component.

ii. **Consider None:** To display no column from the previous component.

iii. **Data Type:** Select a data type for the newly created column using the drop-down list.

iv. **New Predicted Column Name**: Enter an appropriate name for the new predicted column.

v. : 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.

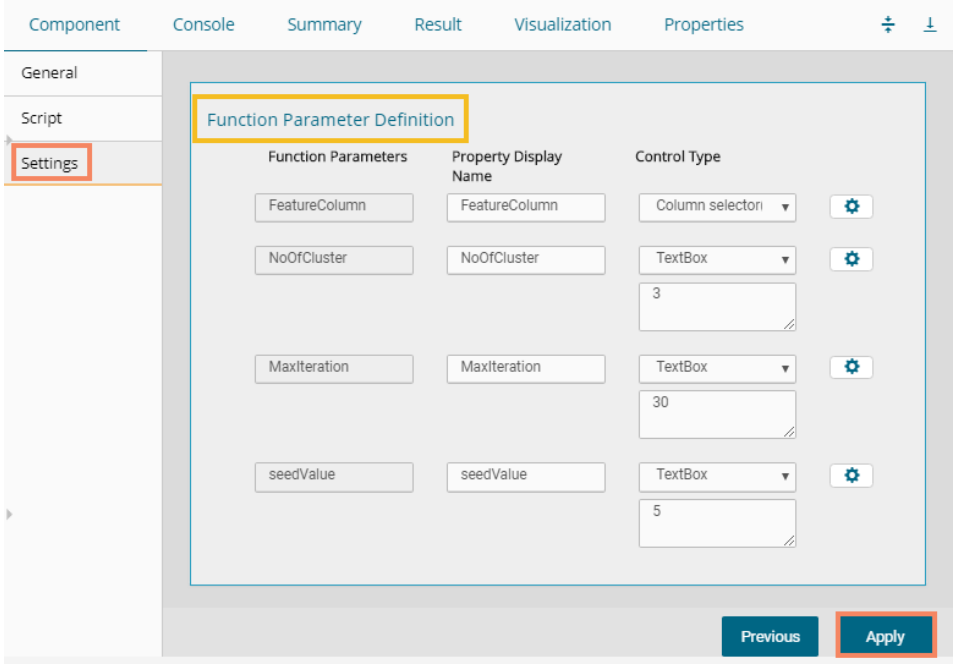
ii. **Property Display Name**: Parameter name to be displayed while configuring saved R script as a component.

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)

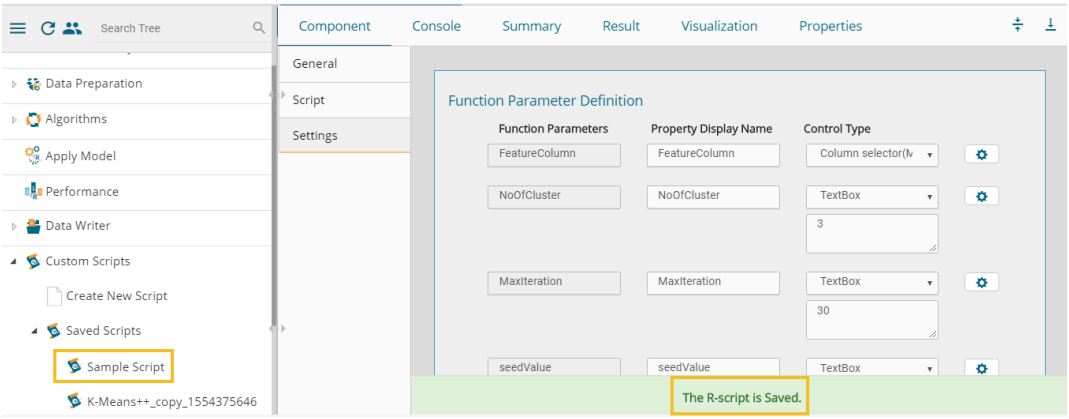
iv. **Settings option**  **:** To set the display for mandatory fields and validate data type for the input column. This field is associated with function parameters.

x) Click the '**Apply**' option.



xi) A success message appears to confirm the creation of the new script.

xii) The newly created script gets saved under the **'Saved Scripts'** options.



Guidelines for Writing an R- Script

1. R- script needs to be written inside a valid R function. i.e., The entire code body should be inside the curly braces of the function.
2. The R-script should have at least one main function. Multiple functions are acceptable, and one function can call another function, but it should be written above the calling function body. (If called function is an outer function) alternatively, above the calling statement (if called function is an inner function).
3. Any extra packages that are required to run your R script must be installed on the R-server, and it should be loaded using the 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 the **'Show Visualization'** field is marked as **'yes'** during the creation of the component, then there should be a plot created in the R-script, and if the **'Show Summary'** field is marked, 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 the data type of input data.

Note:

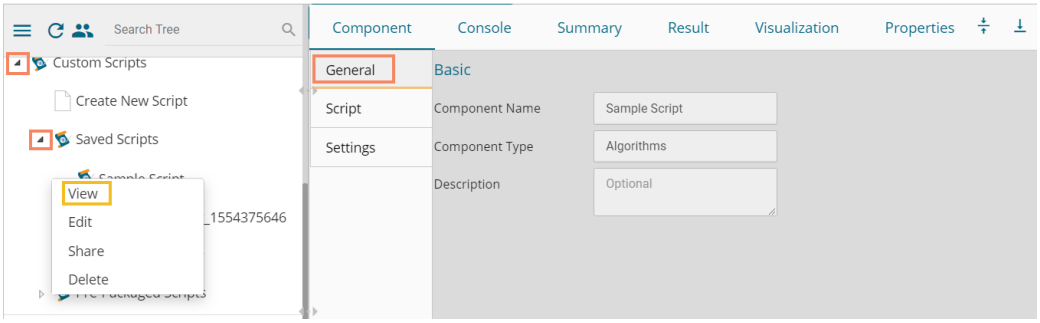
a. Click the ‘**Information**’ button to get the list mentioned above of rules for R-script.
b. ‘**Model Variable Name**’ can be enabled only after selecting the ‘**Show Summary**’ option.
c. Select the **‘Show Summary’** and **‘Show Visualization’** options only if the R-script carries both items.
d. All the supported date data types are listed in date formats in the data type definition; all other date formats are considered as a string data type.
e. Mssql data types are considered string data types.
f. If the input and output components have a different structure, it will not subset or row bind with the “Consider All” option, Users must change to “Consider None” and give different column names for the output to make it run successfully.

13.4.2. Saved Scripts

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

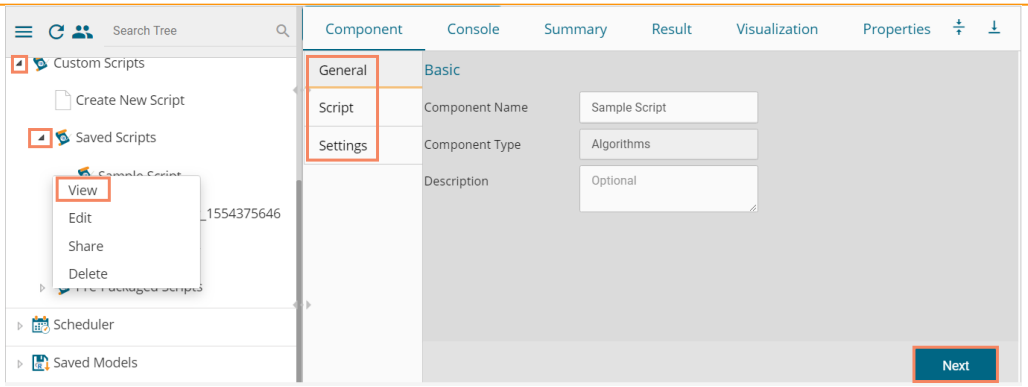
13.4.2.1. Viewing a Saved R Script

i) Select a saved Script from the list of ‘**Saved Scripts**’
ii) Open the context menu by using the right-click.
iii) Select the ‘**View**’ option.
iv) The ‘**General**’ tab opens for the selected saved script.



13.4.2.2. Editing a Saved R Script

i) Select a saved Script from the list of ‘**Saved Scripts**’
ii) Open the context menu of the selected script by using the right-click.
iii) Select the ‘**Edit**’ option.
iv) The General tab opens, displaying the Basic component information.
v) The user can edit the required fields provided under the displayed script component tabs (**General**, **Script**, and **Settings** tabs).



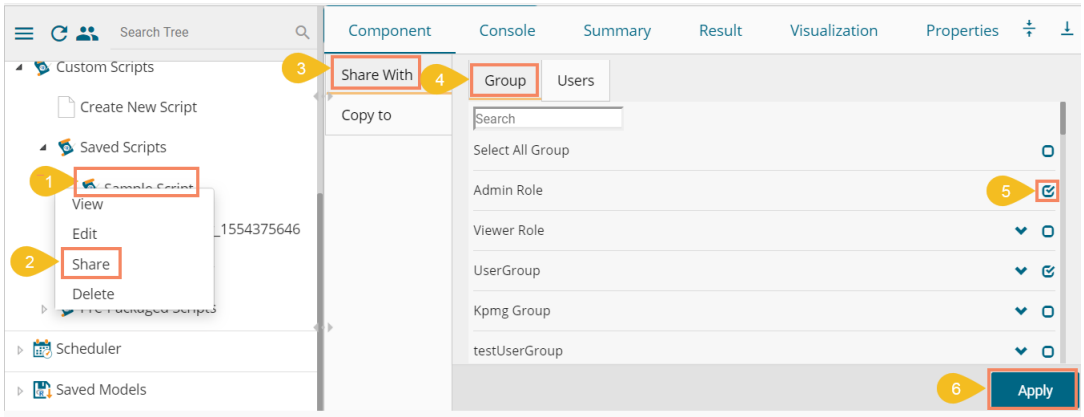
Note: The '**Next**' and '**Apply**' options get displayed for the various tabs of the selected script component.

13.4.2.3. 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 script with the selected users or user groups. Any changes made to the script get transferred to all the users with whom it has been shared.

   i) Right-click on a saved script from the list of '**Saved Scripts**'
   ii) Select the '**Share**' option from the context menu.
   iii) The '**Share With**' option gets displayed (by default)
   iv) Select either '**Group**' or '**Users**'
      a. By selecting a group, all group members inside the group get listed. The users can be excluded by not selecting them from the group.
      b. Users can be excluded by not selecting a username from the list when the '**User**' option has been selected.
   v) Select a specific user or group from the list by using a checkmark in the box.
   vi) Click the '**Apply**' option.

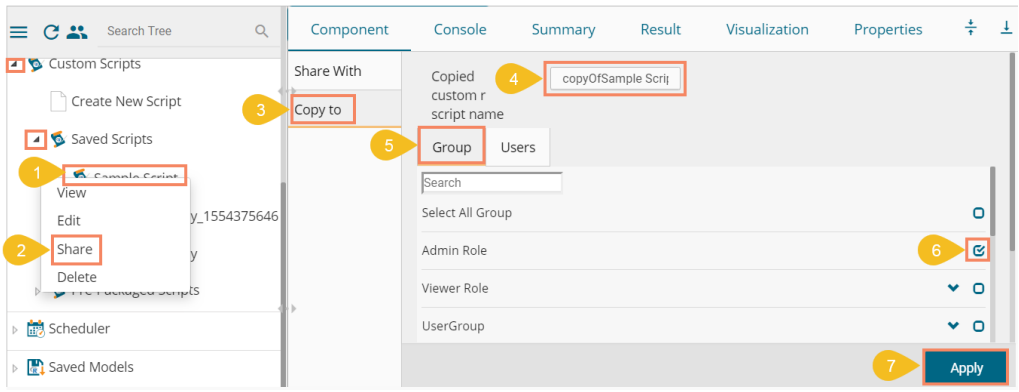


   vii) The selected saved R script gets shared with the chosen user(s)/group(s).

2. **Copy To**: This option creates a copy and shares a 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.

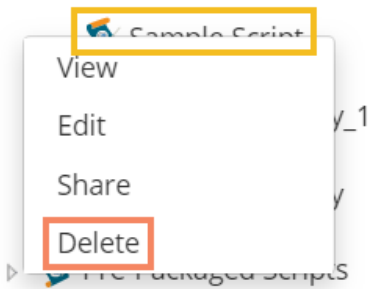
i) Use right-click on a saved R script from the '**Saved Scripts**' list.
ii) Select the '**Share**' option from the context menu.
iii) Select the '**Copy to**' option for sharing the script.
iv) The copied custom R script name gets displayed in a box.
v) Select either the '**Group**' or '**Users**' tab.
   a. By selecting a group, all group members inside the group get listed. Users can be excluded by not selecting them from the group.
   b. Users can be excluded by not selecting a username from the list when the '**Users**' option has been selected.
vi) Select a specific group or user from the list by using a checkmark in the box.
vii) Click the '**Apply**' option.



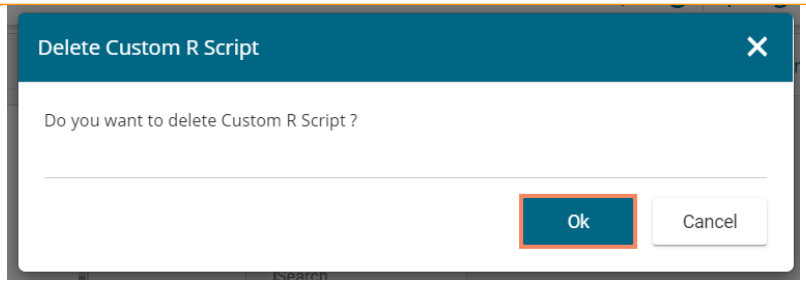
viii) The selected saved R script gets copied to the selected user(s)/group(s).

13.4.2.4. Deleting a Saved R Script

i) Select a Script from the list of '**Saved R-Script**'
ii) Right-click on the selected R Script.
iii) A context menu will open.
iv) Select the '**Delete'** option.



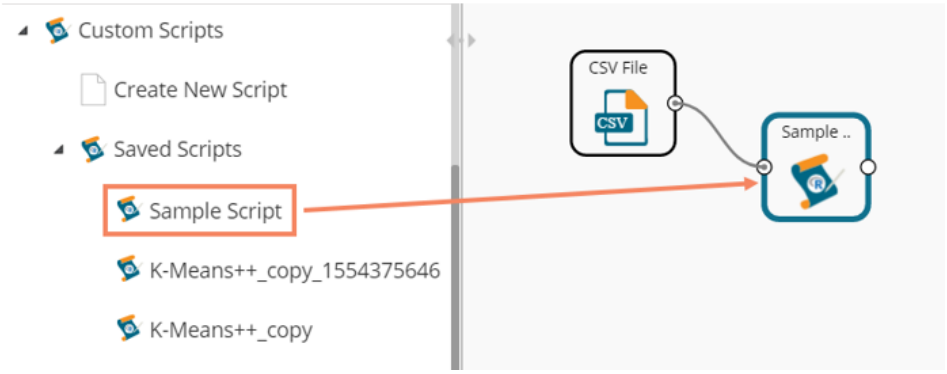
v) A pop-up window appears to assure the deletion.
vi) Click the '**Ok**' option.



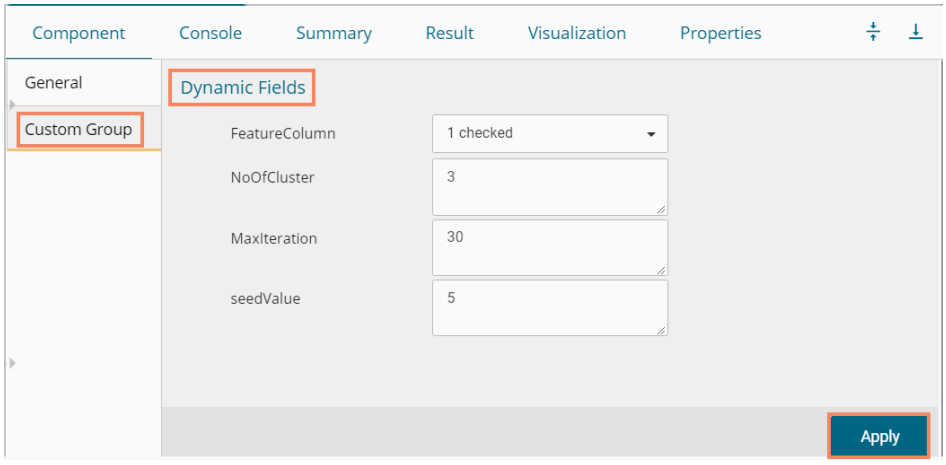
vii) The selected R-Script gets deleted.

13.4.2.5. Connecting Saved R Script with a Data Source

i) Click the **'Custom Script'** tree node.
ii) Select and drag a saved R-script to the workspace.
iii) Connect the Script component to a configured data source.
iv) Click the dragged script component to get the configuration fields.

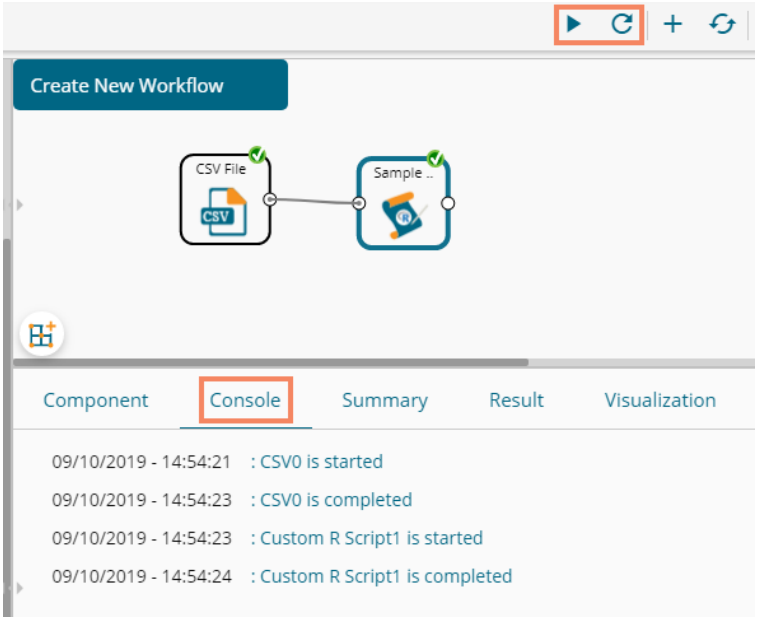


v) Configure the Dynamic Fields.
vi) Click the '**Apply**' option.



vii) Run the workflow after getting the success message.

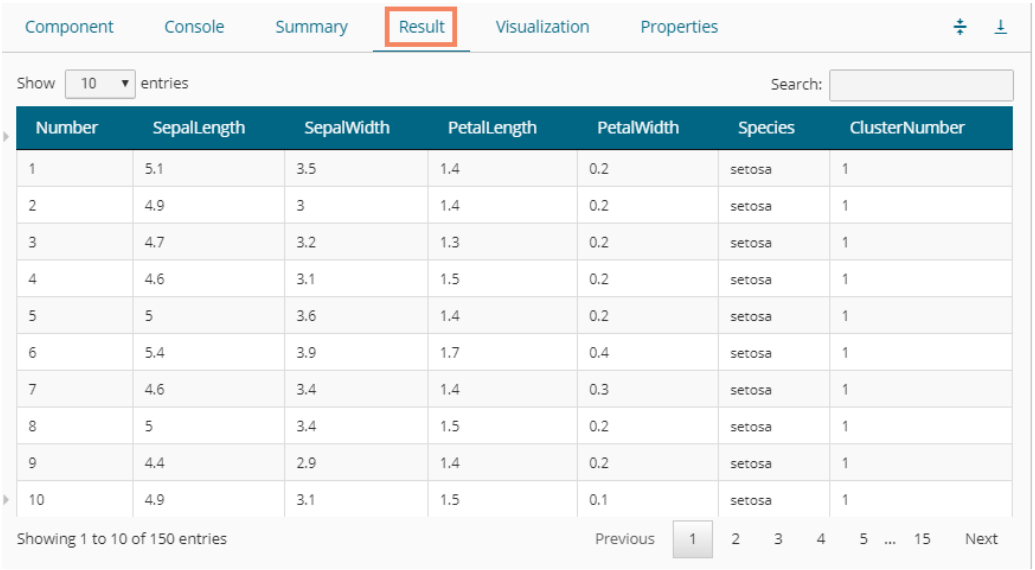
viii) The console tab appears displaying steps of the process. The completion of the console process gets marked by green marks at the top of the dragged components.



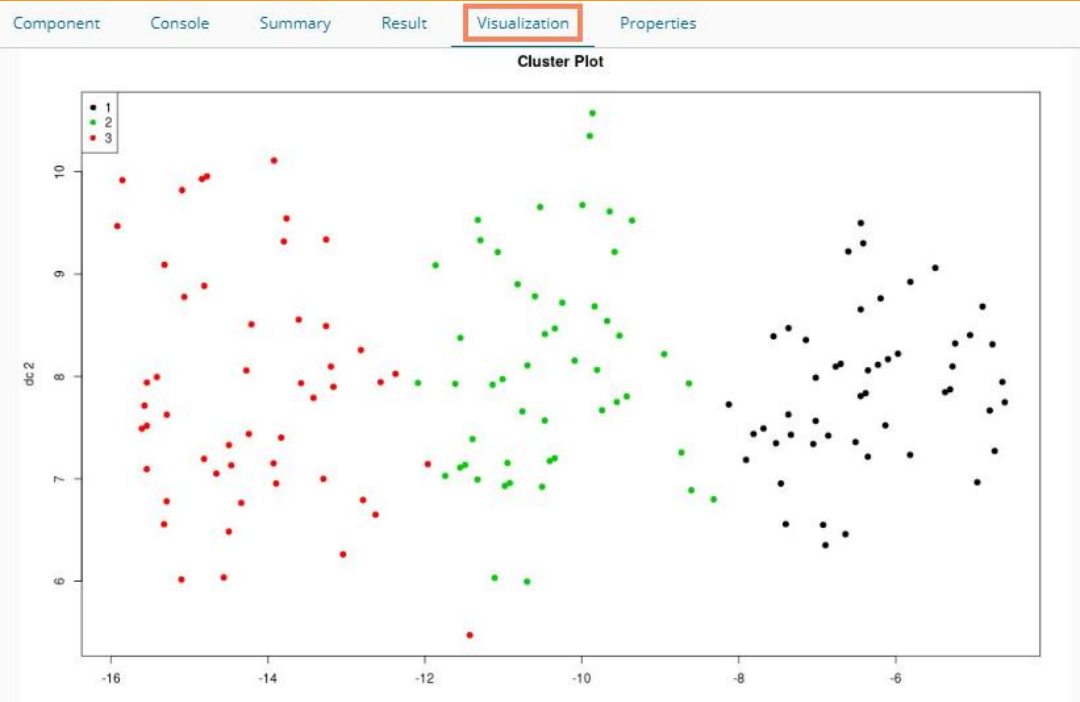
ix) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



x) Click the **'Visualization'** tab to see the result data presented through the Cluster Plot chart.

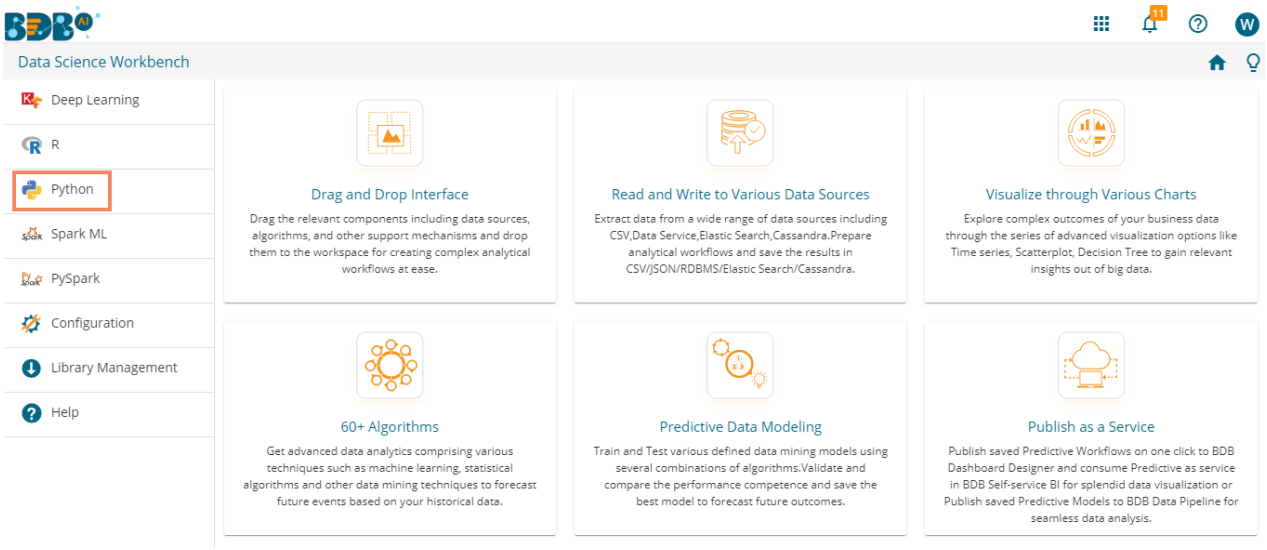


Note:

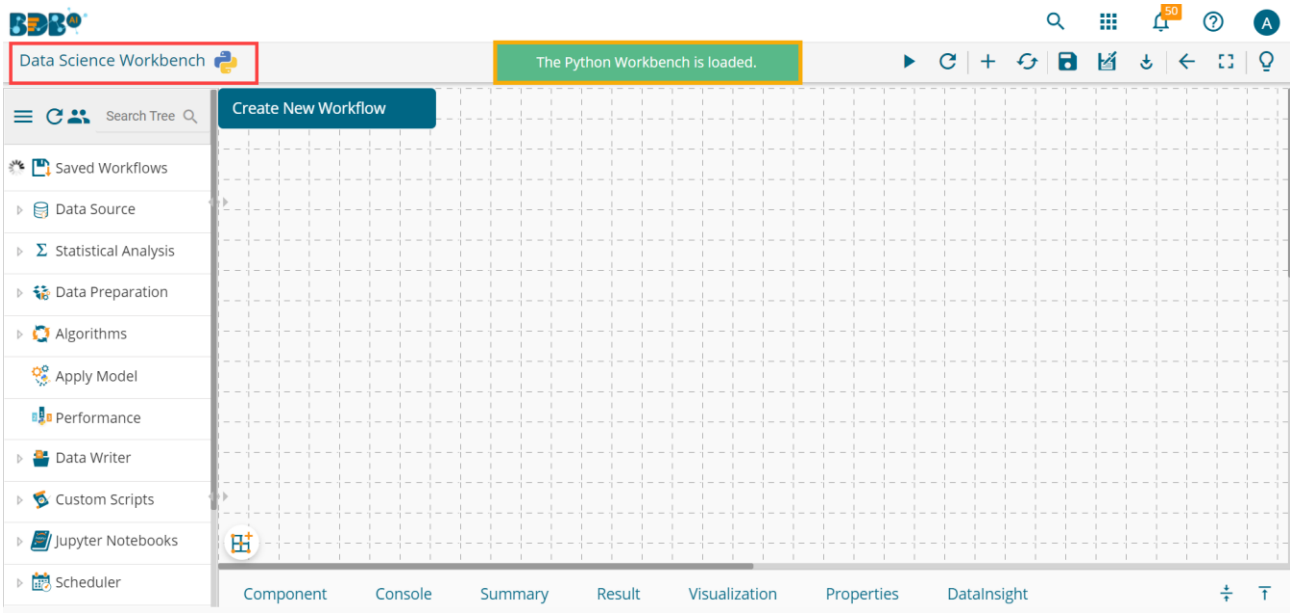
a. The above-given process is displayed for a CSV data source. A similar set of steps can be followed for other data source types.

14. Python Workspace

The user can select the Python Workspace from the Predictive landing page to access the Python Environment under the Data Science Workbench.



The following screen opens loading the Python Workbench



14.1. Algorithms

14.1.1. Forecasting

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

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

1. Forecasting
2. Trend

The document describes all the available Forecasting algorithms considering both the output modes as possibilities.

14.1.1.1. SARIMAX

Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors Variables X (SARIMAX) is an extension of SARIMA (Seasonal ARIMA) and ARIMA model that explicitly supports univariate time series data with a seasonal component along with the inclusion of exogenous variables X.

It adds three new hyperparameters to specify the autoregression (AR), differencing (I), and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

i) Drag the SARIMAX component to the workspace and connect it to a configured data source.
ii) Click on the dragged SARIMAX component to get the component properties fields.

iii) The user gets **Properties** fields based on the selected **Output Mode (Forecast/ Trend)**

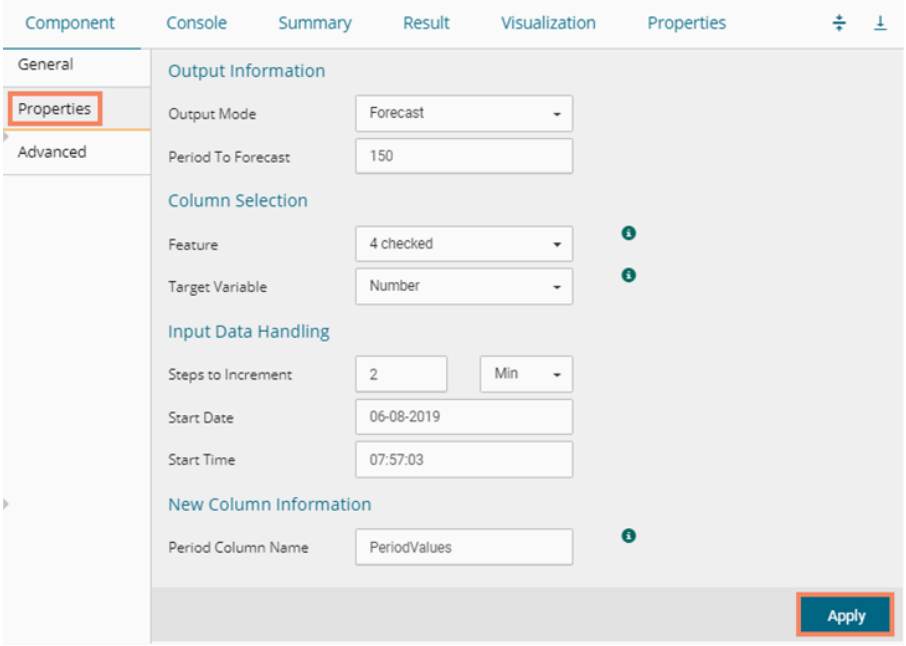
- **Properties with Forecast Output Mode**

a. **Output Information**
   i. **Output Mode: Select a mode in which you want to display output data. The user** gets two options for this field.
      1. **Trend**: Selecting this option displays source data along with predicted values for the given data set.
      2. **Forecast: Selecting this option displays forecasted values for the given period.** Results data gets 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.'**

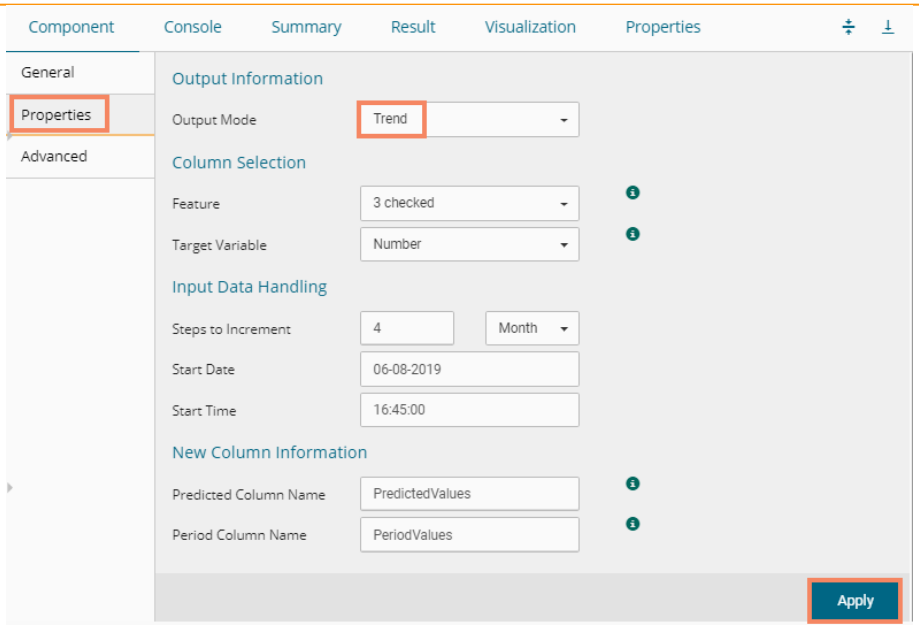
b. **Column Selection**
   i. **Feature**: Select the feature columns using the drop-down menu.
   ii. **Target Variable:** Select the target variable for which you want to Apply forecasting analysis (The first selected option gets selected by default. Only numerical columns are accepted.)

c. **Input Data Handling**
   i. **Steps to Increment:** Provide a number to decide increment and a period by choosing any one option from the drop-down menu.
   ii. **Start Date:** Select a start date using the calendar or set it manually.
   iii. **Start Time:** Enter the definite time to start the process in the hh/mm/ss format

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



- **Properties with Trend Output Mode**
  a. **Output Information**
    i. **Output Mode: Select a mode in which you want to display output data. The user gets** two options for this field.
      1. **Trend**: Selecting this option displays source data along with predicted values for the given dataset.
      2. **Forecast:** Selecting this option displays forecasted values for the given period. Results get appended to the target column when the '**Forecast**' output mode has been selected.
    ii. **Period to Forecast**: Enter a period to forecast. This field appears only when '**Forecast**' is selected as an **'Output Mode'** option**.**
  b. **Column Selection**
    i. **Feature**: Select the feature columns using the drop-down menu.
    ii. **Target Variable:** Select the target variable for which you want to Apply forecasting analysis (The first selected option gets selected by default. Only numerical columns are accepted.)
  c. **Input Data Handling**
    i. **Steps to Increment:** Provide a number to decide increment and a period by choosing any one option from the drop-down menu.
    ii. **Start Date:** Select a start date using the calendar or set it manually.
    iii. **Start Time:** Enter the definite time to start the process in the hh/mm/ss format
  d. **New Column Information**
    i. **Predicted Column Name**: Enter a name for the column containing the Predicted Values (The title for this field comes pre-defined, but the users can change the value if needed).
    ii. **Period Column Name:** Enter a name for the column containing a period value (This field comes predefined, but users can change the value if needed).

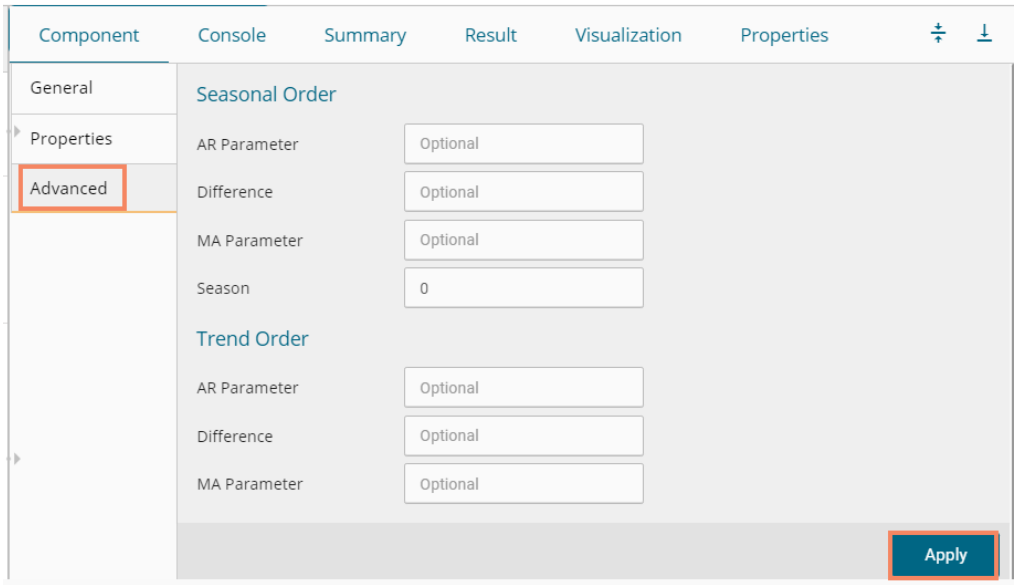


iv) Click the '**Advanced'** tab and configure, if required:

a. Configure the following '**Seasonal Order'** fields
   i. AR Parameter
   ii. Difference
   iii. MA Parameter
   iv. Season

b. Configure the following '**Trend Order**' information
   i. AR Parameter
   ii. Difference
   iii. MA Parameter

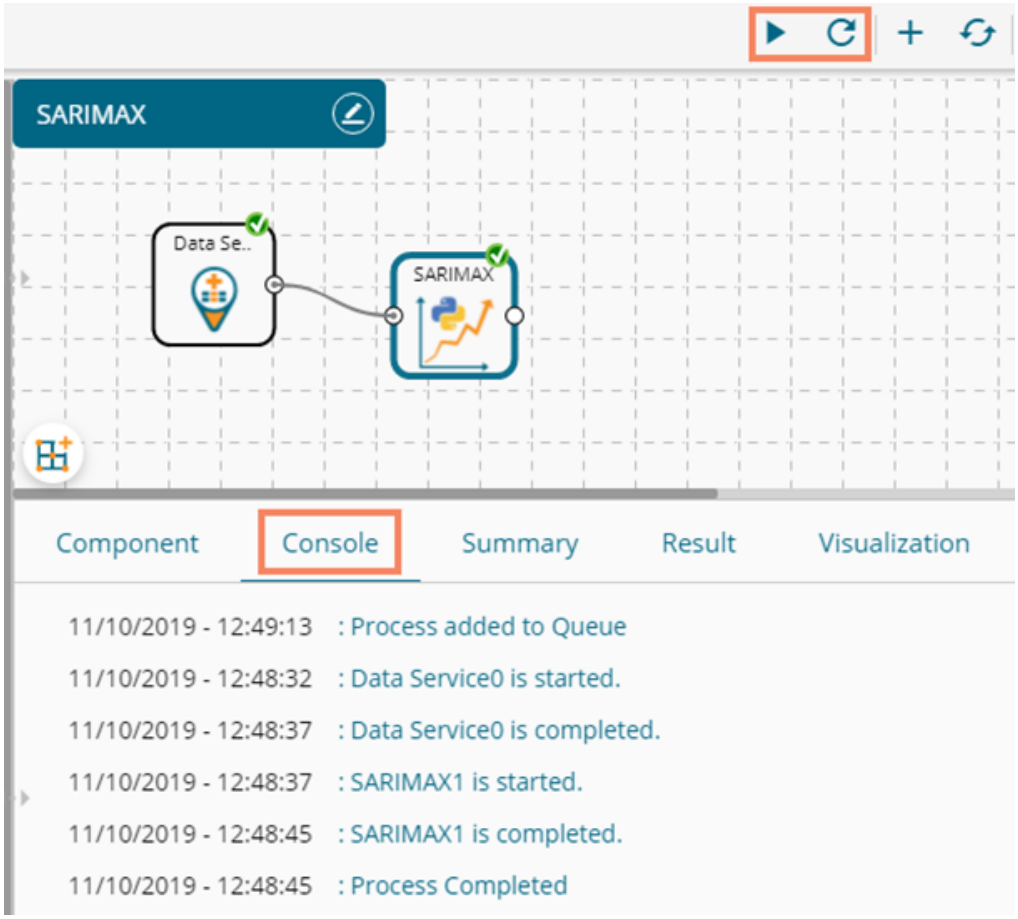
v) Click the **'Apply'** option.



Note: The '**Advanced**' tab remains the same for any output mode.

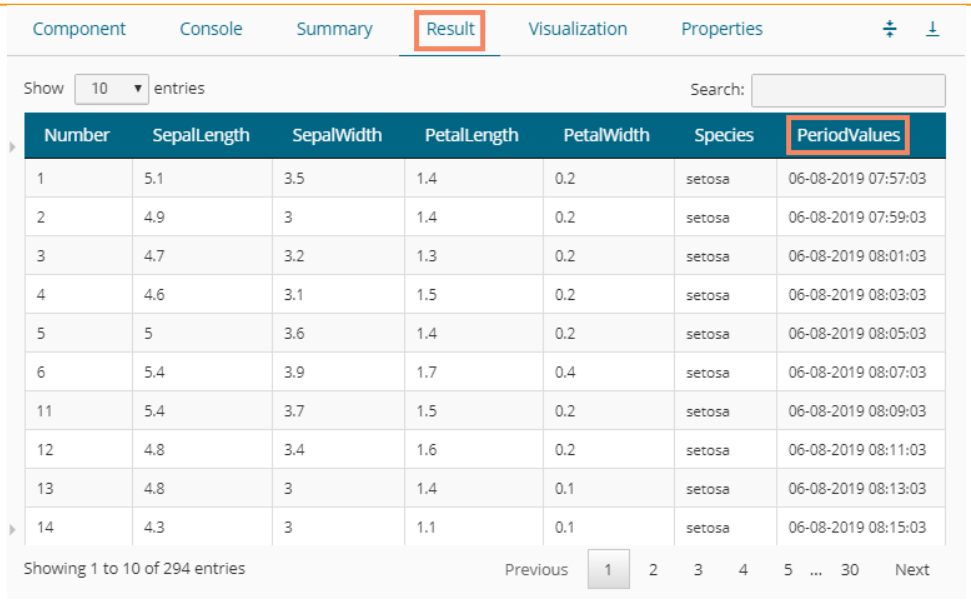
vi) Run the workflow after getting the success message.

vii) The user gets directed to the '**Console**' tab displaying the ongoing process. The completion of the Console process gets marked by the green checkmarks on the top of the dragged component.

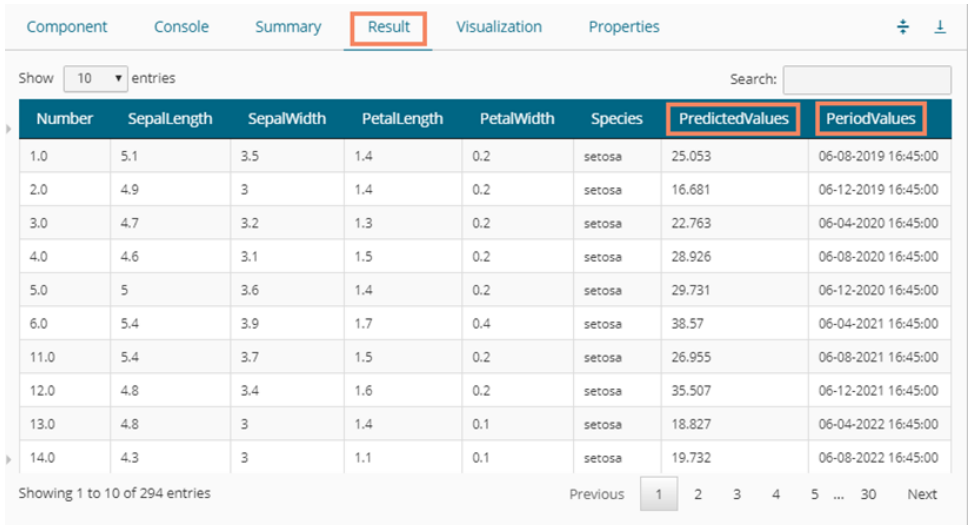


viii) View the processed data by clicking the dragged SARIMAX component and then clicking the '**Result**' tab.

a) Result tab with '**Forecast**' as the output mode

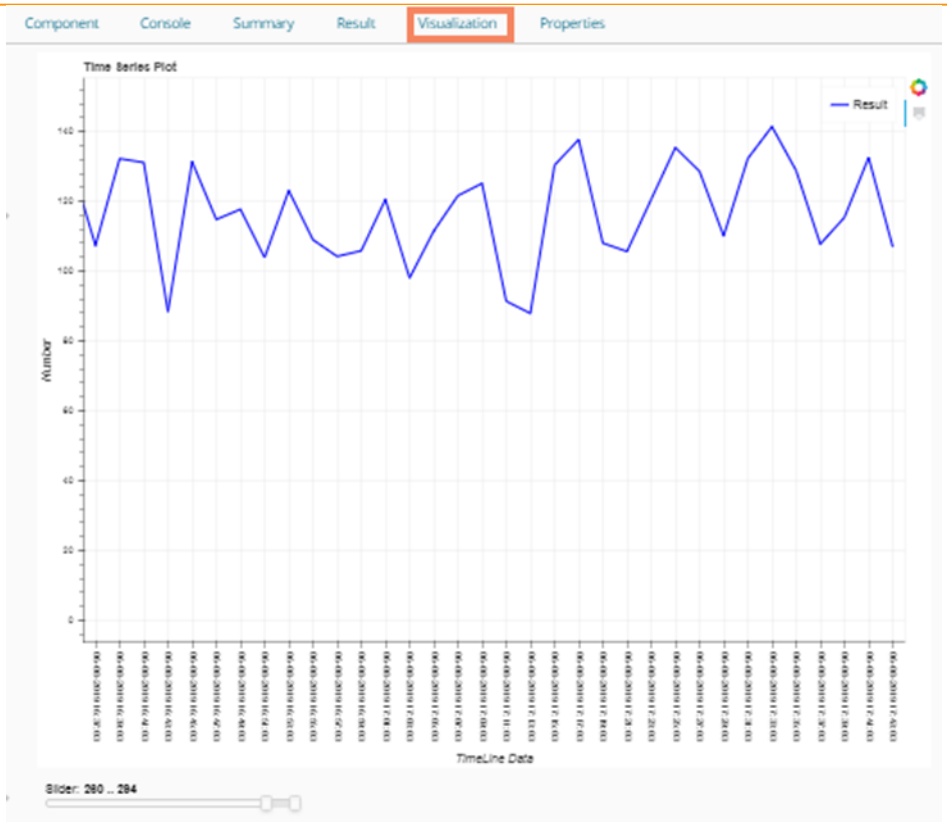


b) Result tab with '**Trend**' as the output mode

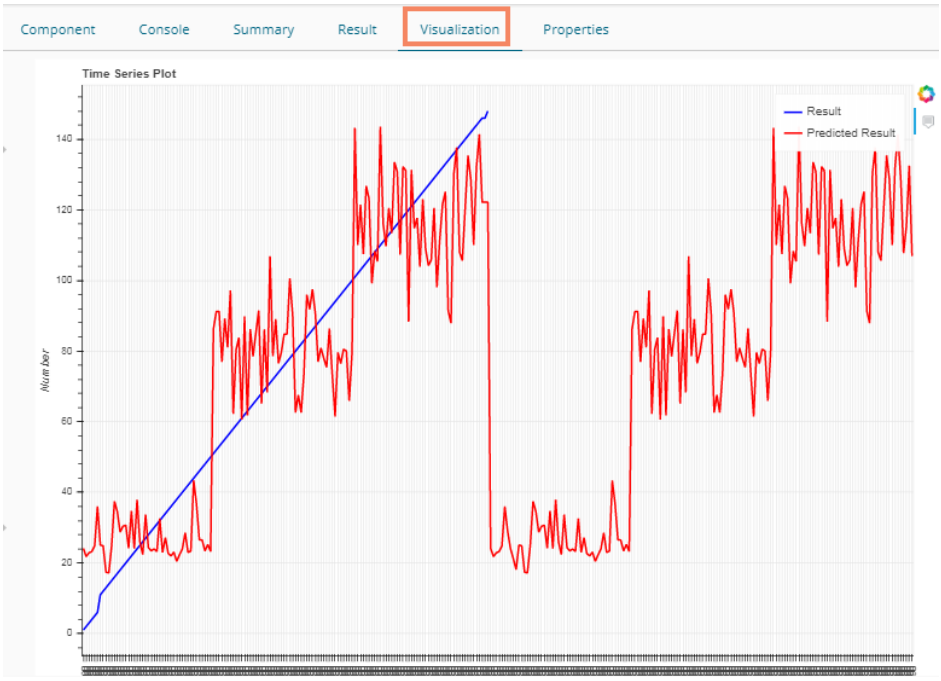


ix) Click the '**Visualization**' tab to open the graphical representation of the processed data through a time series chart.

a) Visualization of the processed data with '**Forecast**' as output mode.



b) Visualization of the processed data with Trend as output mode

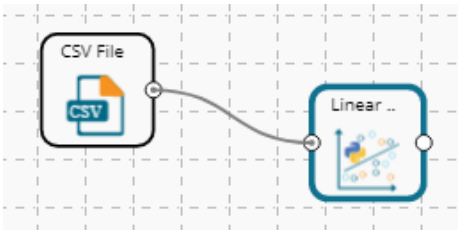


14.1.2. Regression

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.

14.1.2.1. Linear Regression

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



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

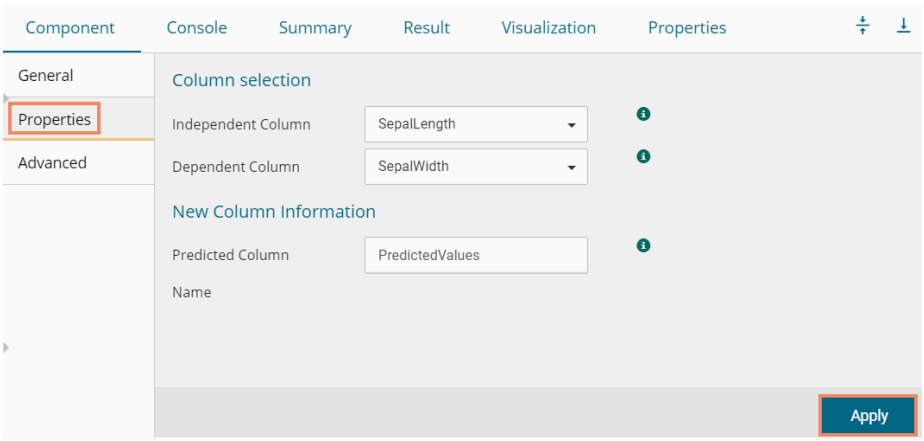
a. Column Selection

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

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

b. New Column Information

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



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

a. Behavior

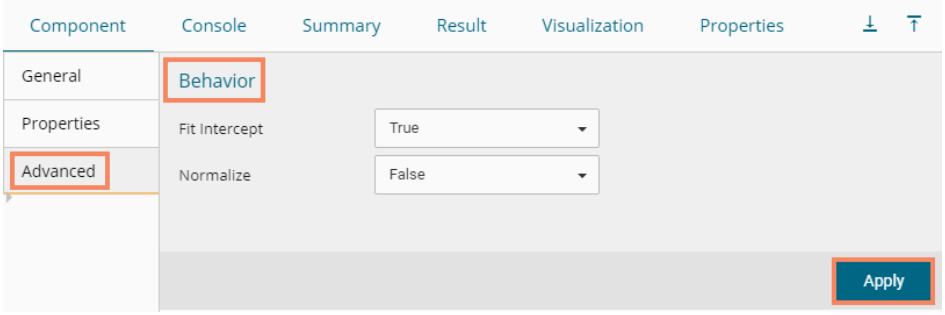
i. **Fit Intercept**: This option is used to select whether to calculate the intercept for the selected model or not

1. **True:** By selecting this option intercept gets calculated (It is the default selection)
2. **False:** By selecting this option intercept does not get calculated

ii. **Normalize**: This option is used to select whether to normalize the feature column or not

1. **True**: If the selected Normalize option is **True**, the feature column gets the selected normalization option.
2. False: If the Normalize option is False, the feature column cannot be normalized (It is the default option).

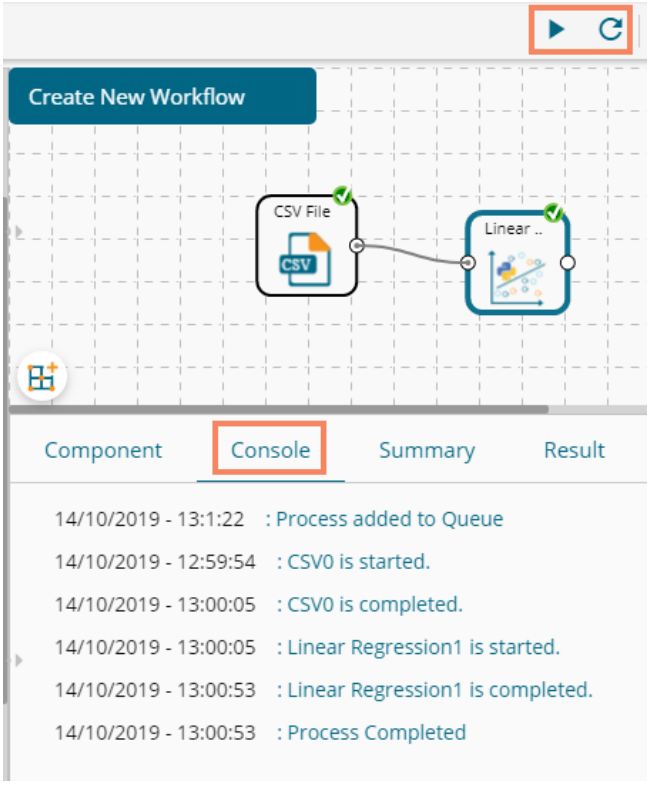
iv) Click the '**Apply**' option.



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

v) Run the workflow after getting the success message.

vi) The user gets the process status under the '**Console**' tab.

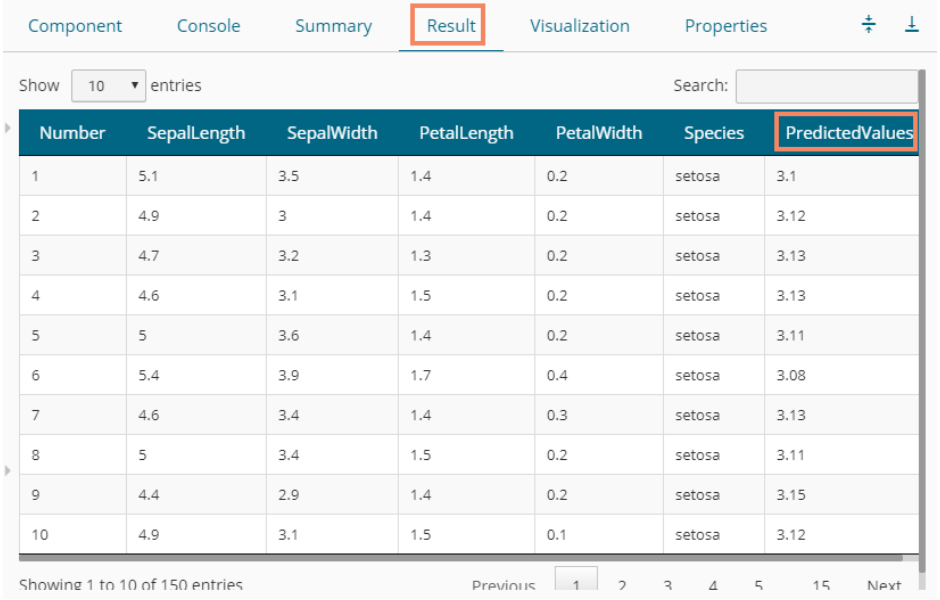


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

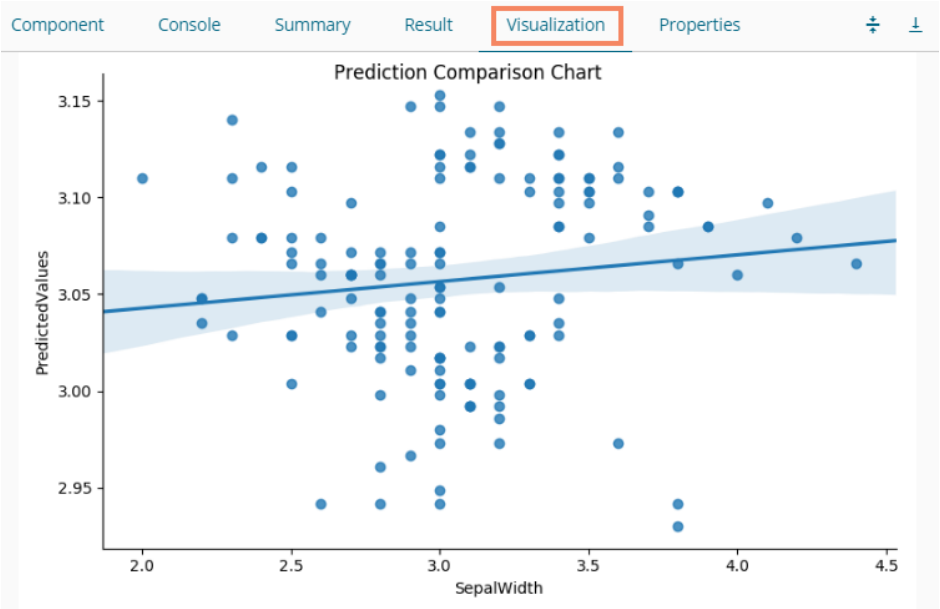
b. Click the '**Result**' tab.

i. A new column displaying the predicted values gets added to the result view.



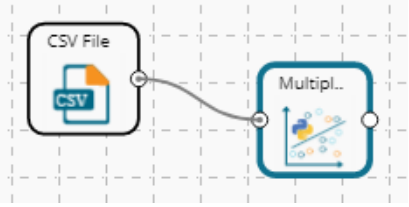
viii) Click the '**Visualization**' tab.

ix) The processed data gets displayed via the Prediction Comparison chart with a Regression line.

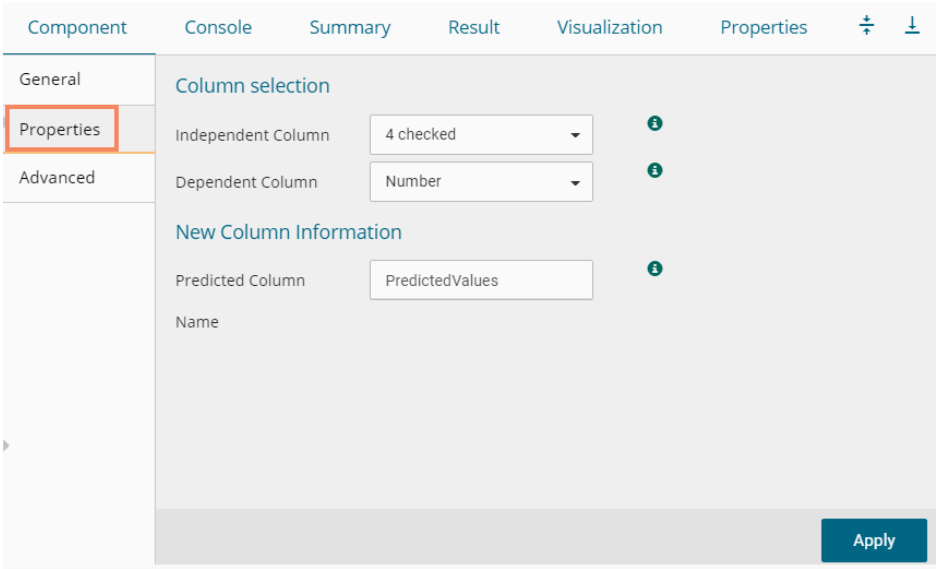


14.1.2.2. Multiple Linear Regression

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



ii) Configure the '**Properties**' tab as displayed below:



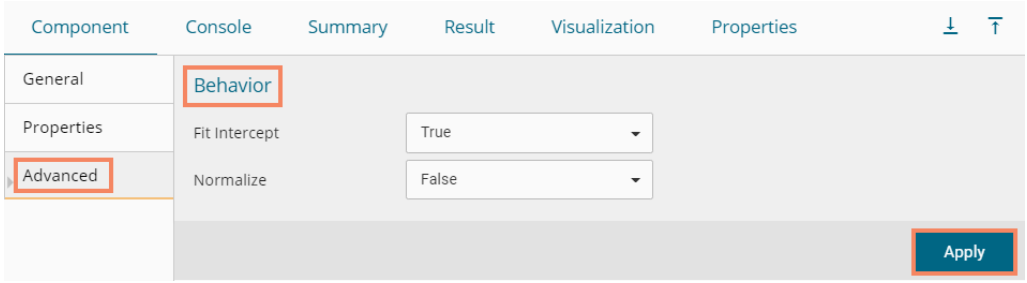
iii) Click the '**Advanced**' tab and configure if required:

a. Behavior

i. **Fit Intercept**: This option is used to select whether to calculate the intercept for the selected model or not
   1. **True:** By selecting this option intercept gets calculated (It is the default selection)
   2. **False:** By selecting this option intercept gets calculated

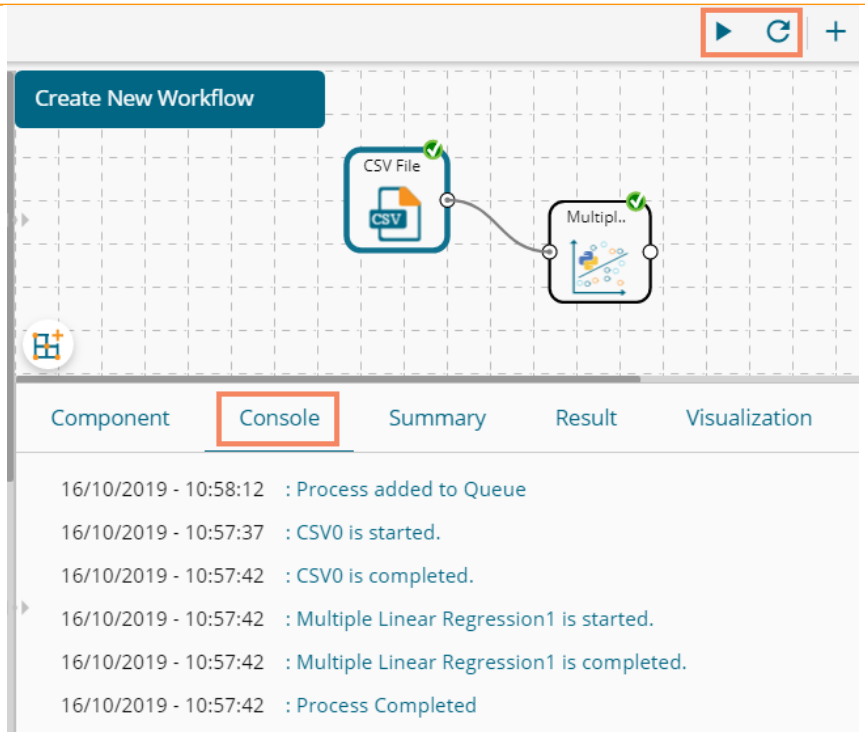
ii. **Normalize**: This option is used to select whether to normalize the feature column or not
   1. True: If Normalize option is '**True**,' it normalizes the feature column
   2. False: If Normalize option is '**False**,' the feature column does not take the normalization value (It is the default option)

iv) Click the **'Apply'** option**.**



v) Run the workflow after getting the success message.

vi) The '**Console**' tab opens displaying the progress of the process. The completed console process gets marked by the green checkmarks on the top of the dragged components.

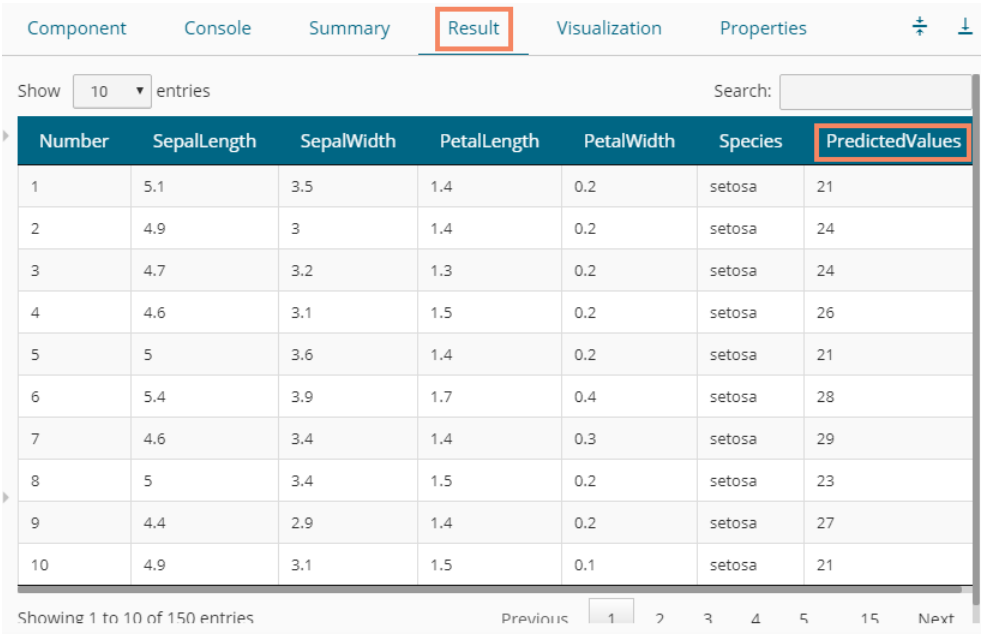


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

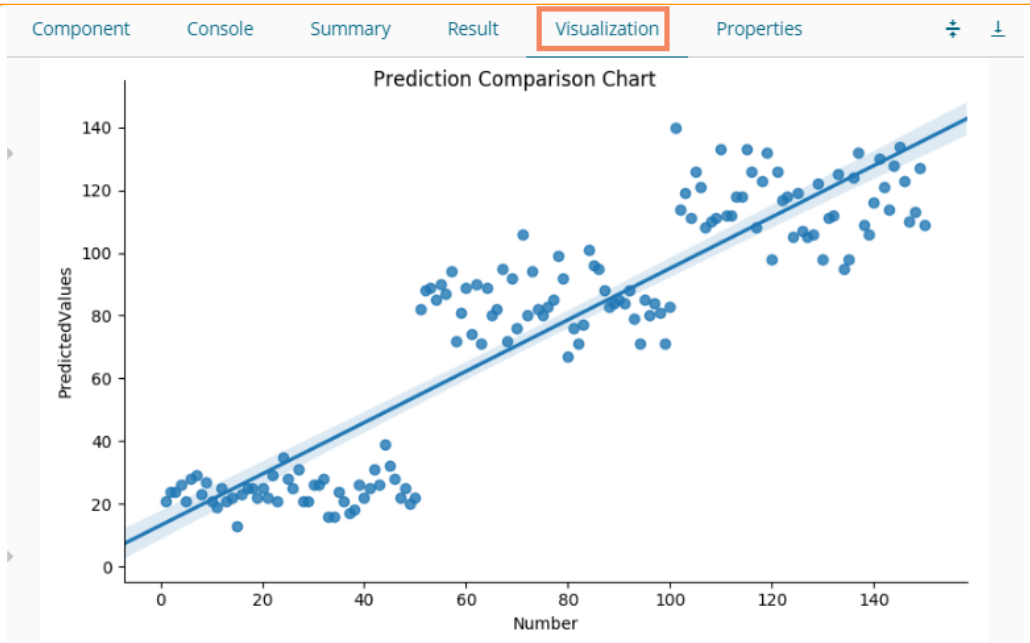
b. Click the '**Result**' tab.

i. A new column containing the Predicted Values gets added to the Result data.



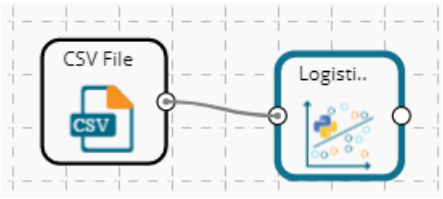
viii) Click the '**Visualization**' tab.

ix) The Result data gets displayed via the Prediction Comparison Chart with a Regression line.

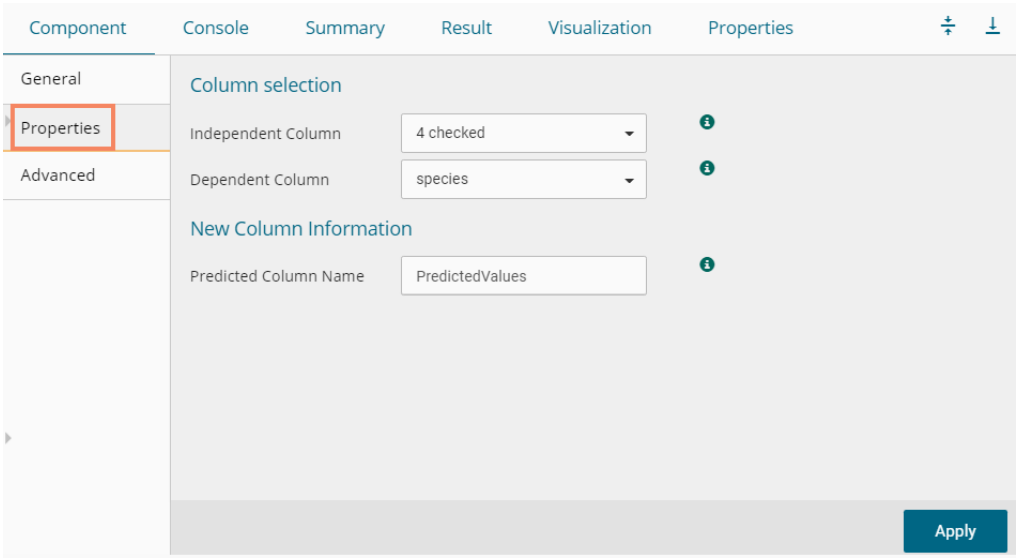


14.1.2.3. Logistic Regression

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 stops the 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 a value of 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 value for this field is 100.

iv. Solver: The following options get 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 a Boolean value (The default value for this field is 'False')

vi. Tolerance: It can have a double type value (The default value for this field is 0.0001)

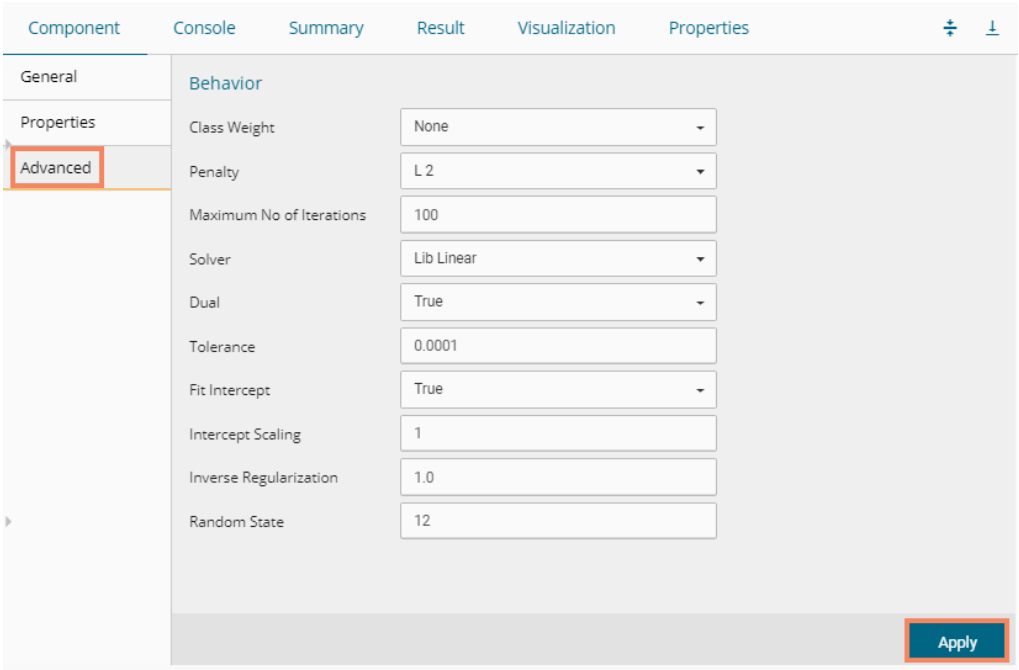
vii. Fit Intercept: It has two options 'True' and 'False.' By selecting 'True,' it calculates the intercept for the selected model (The default value for this field is 'True')

viii. Intercept Scaling: It can have a 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)

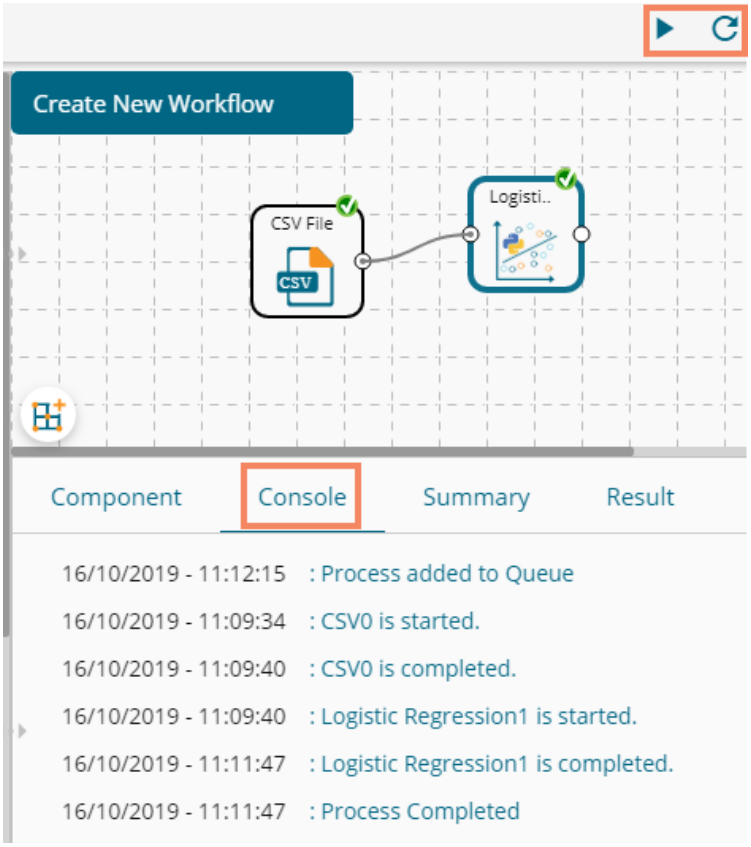
x. Random State: This field can only take integer values (The default value for this field is 12)

iv) Click the **'Apply'** option.



v) Run the workflow after getting the success message.

vi) The '**Console**' tab opens, displaying steps of the ongoing process.

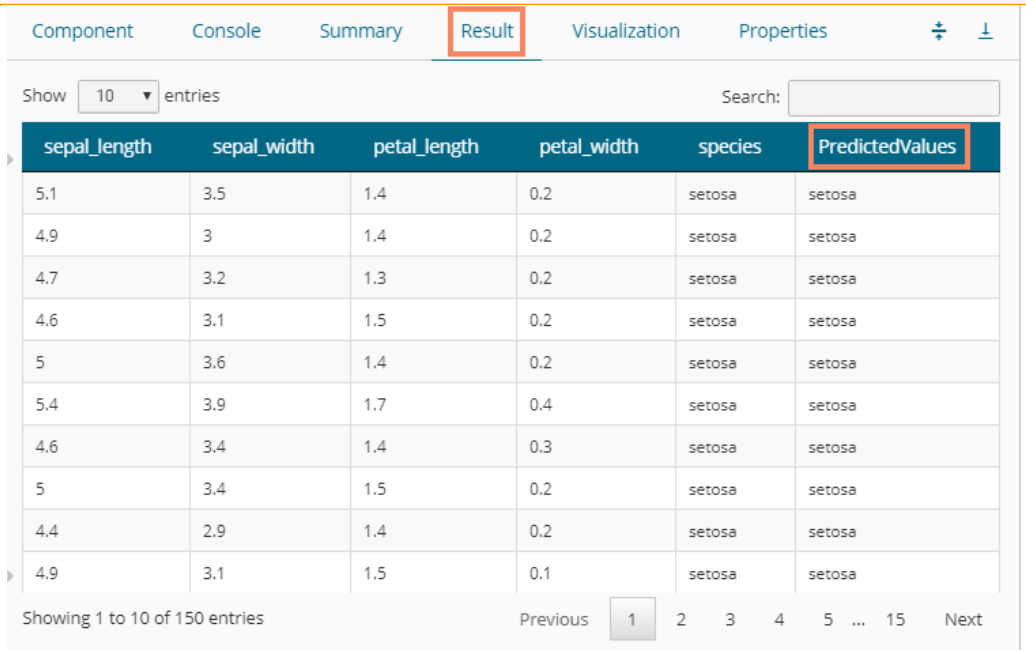


vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

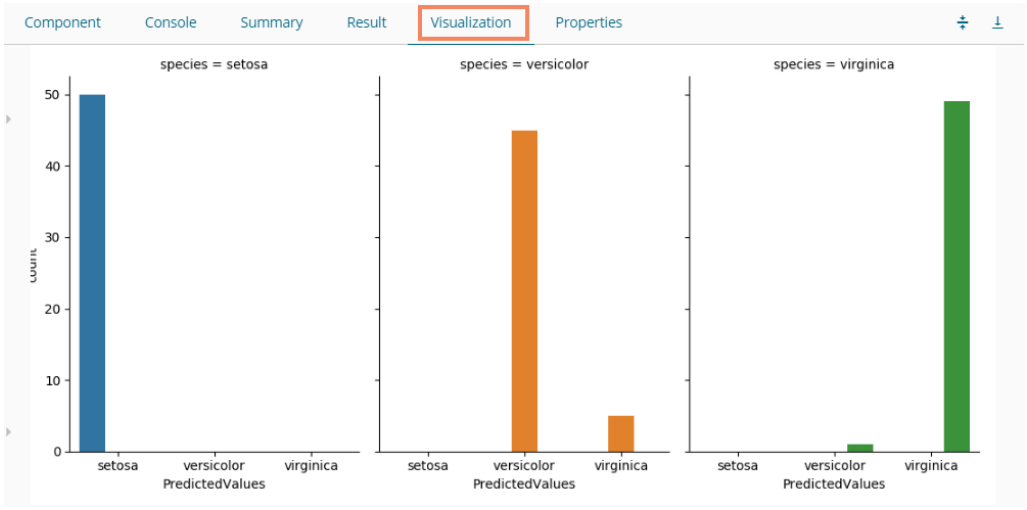
b. Click the '**Result**' tab.

viii) A new column containing Predicted values gets added to the Result data.

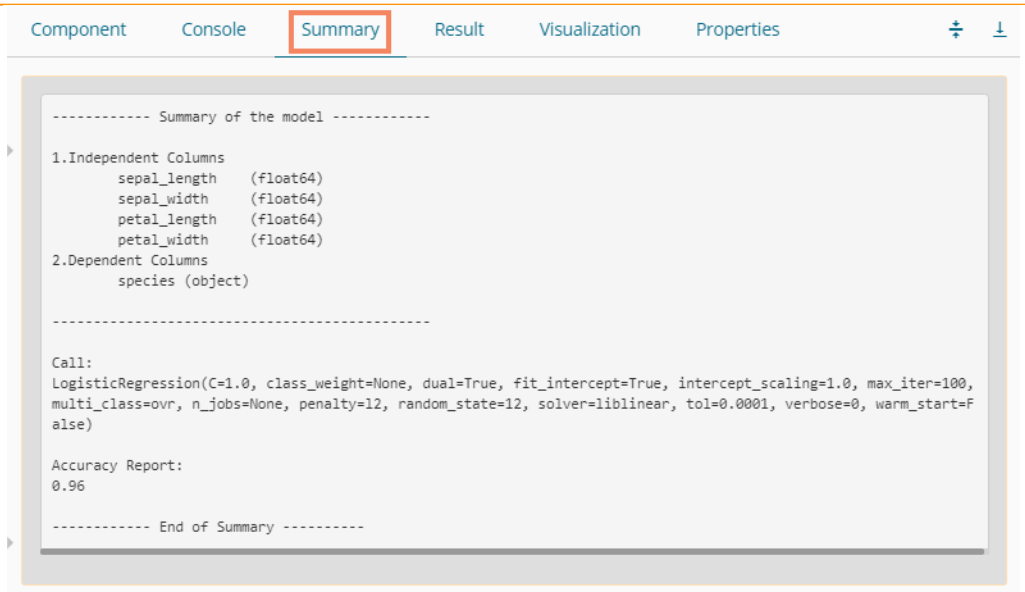


ix) Click the '**Visualization**' tab.

x) The processed data gets displayed via the Comparative Column chart.

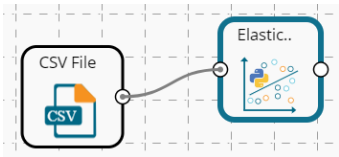


xi) Click the '**Summary**' tab to view the model summary.



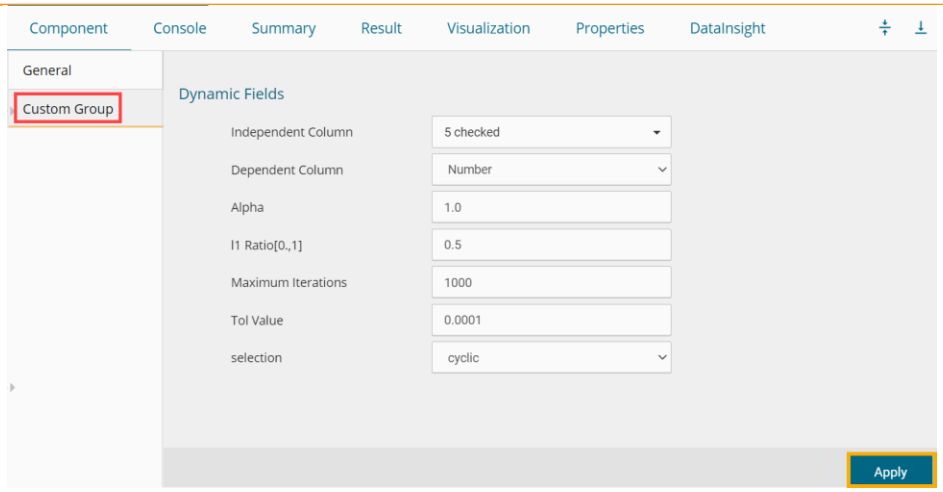
14.1.2.4. Elastic Net

i) Drag the Elastic Net component to the workspace and connect it to a configured data source.



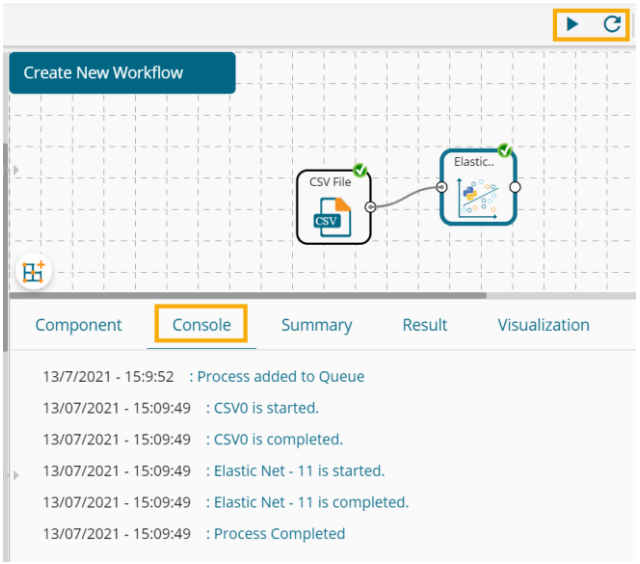
ii) Configure the following fields in the '**Custom Group**' tab:
   a. **Dynamic Fields**
      i. **Dependent Column**: Select the target column on which the regression analysis gets applied
      ii. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.
      iii. **Alpha:** Provide value for the alpha field
      iv. **l1 Ration [0..1]:** Provide a number to decide the ratio (It should be between 0 to 1)
      v. **Maximum Iterations:** Provide the maximum number of iterations.
      vi. **Tol Values:** Provide the Tol values.
      vii. **Selection:** Select one option out of the '**Random**' or '**Cyclic**' options.

iii) Click the '**Apply**' option.

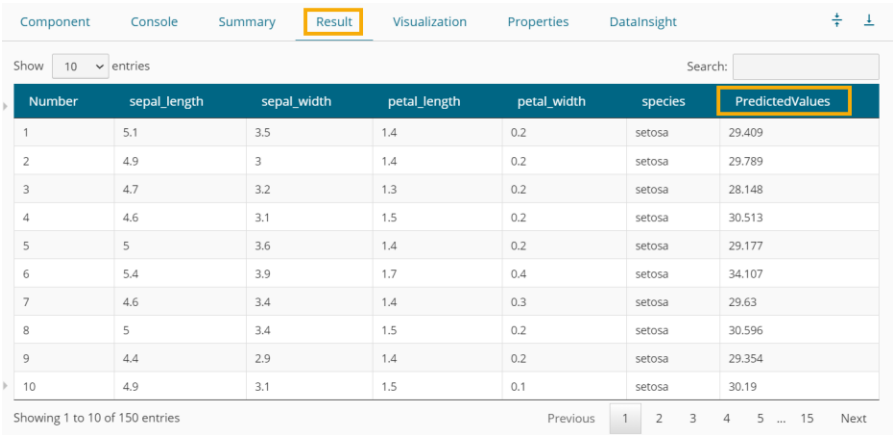


Note: The model containing aliased coefficients signifies that the square matrix x*x is singular.

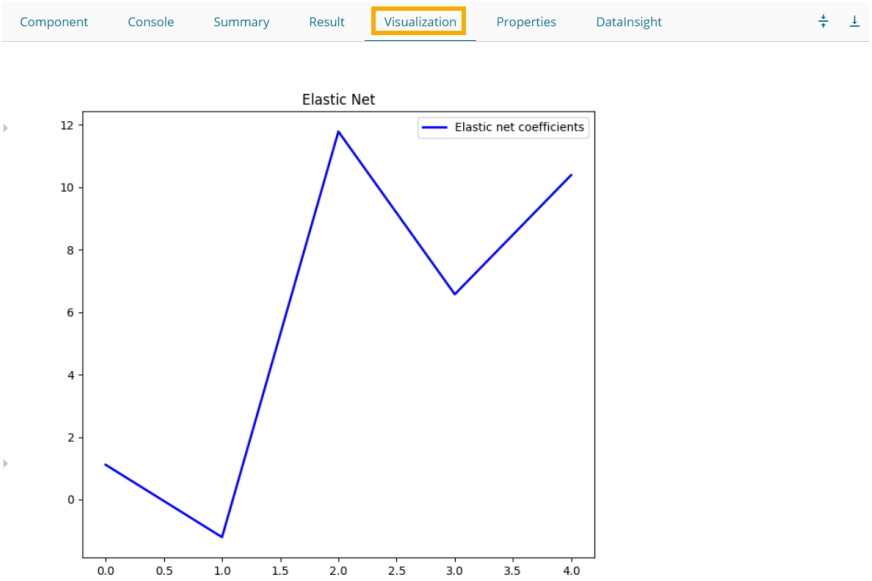
iv) Run the workflow after getting the success message.
v) The user gets the process status under the '**Console**' tab.



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



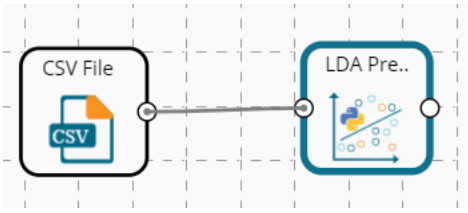
vii) A new column displaying the predicted values gets added to the result view.
viii) Click the '**Visualization**' tab.



ix) The processed data gets displayed via the Prediction Comparison chart with a Regression line.
x) Click the '**Summary**' tab to display the summary.

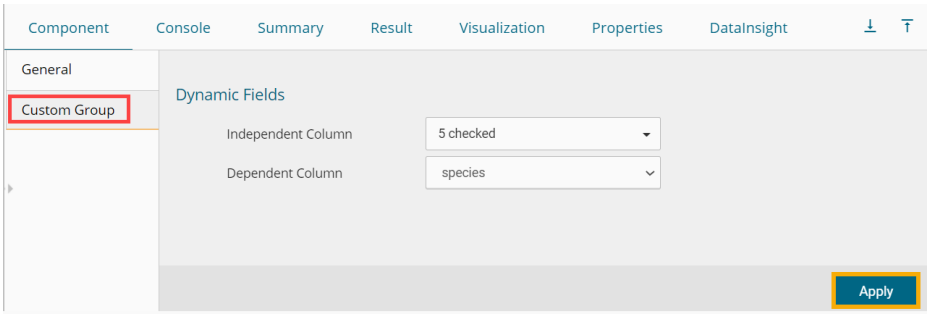
14.1.2.5. LDA Prediction

i) Drag the LDA Prediction component to the workspace and connect it to a configured data source.



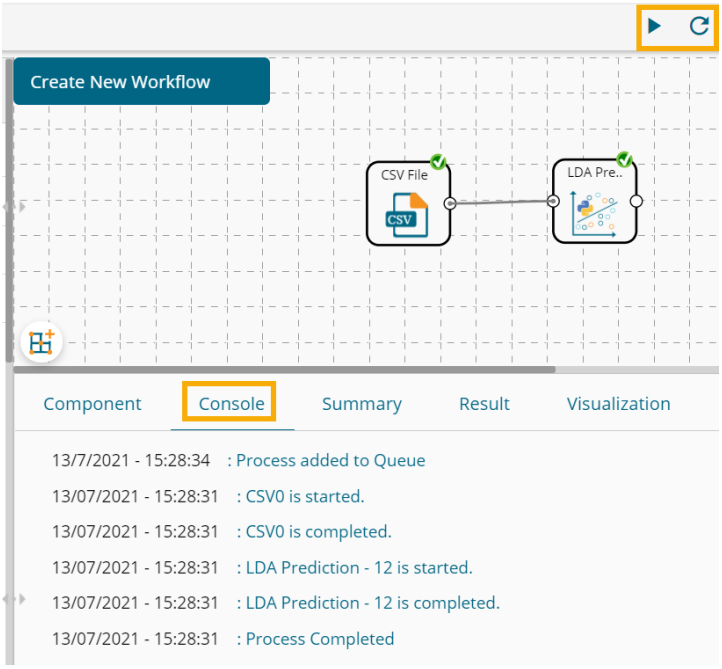
ii) The '**Custom Group**' tab opens.
   a. **Dynamic Fields**
      i. **Dependent Column**: Select the target column on which the regression analysis gets applied.
      ii. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.

iii) Click the '**Apply**' option.

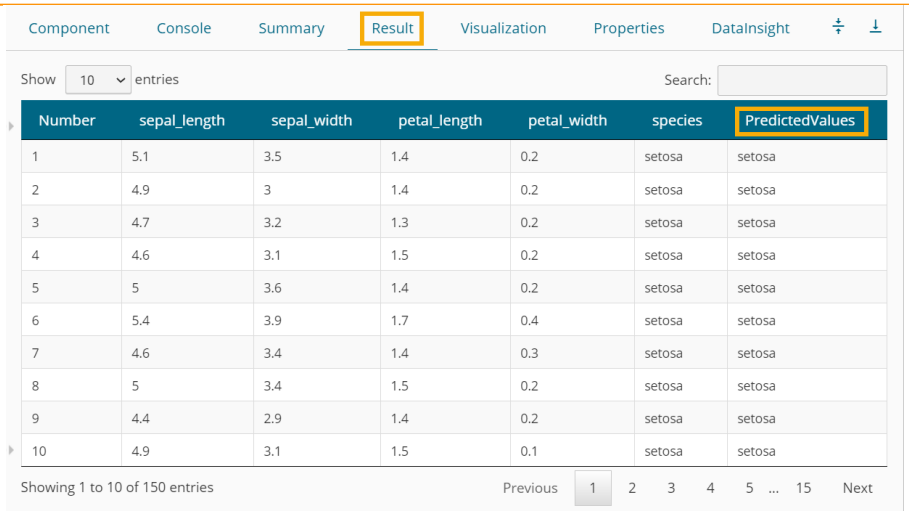


iv) Run the workflow after getting the success message.

v) The user gets the process status under the '**Console**' tab.

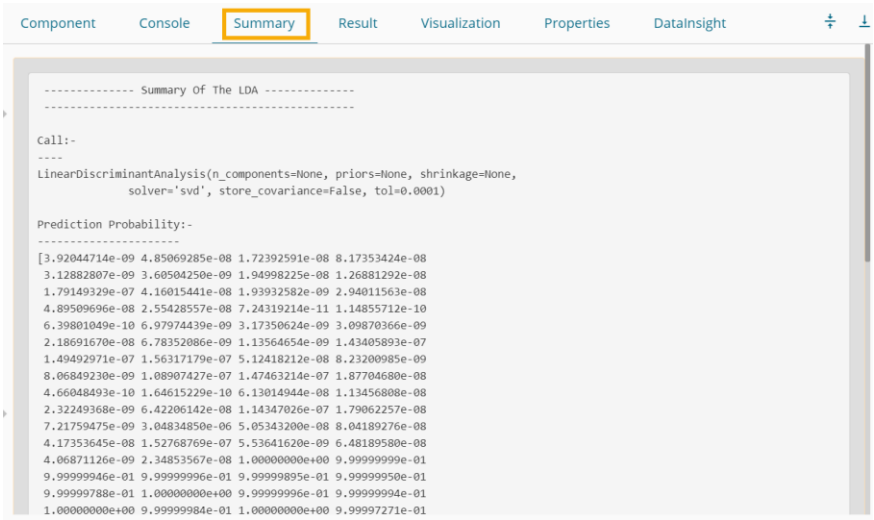


vi) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.
   c. A new column displaying the predicted values gets added to the result view.



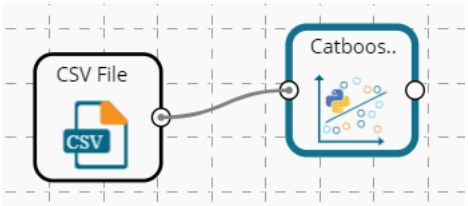
Number	sepal_length	sepal_width	petal_length	petal_width	species	PredictedValues
1	5.1	3.5	1.4	0.2	setosa	setosa
2	4.9	3	1.4	0.2	setosa	setosa
3	4.7	3.2	1.3	0.2	setosa	setosa
4	4.6	3.1	1.5	0.2	setosa	setosa
5	5	3.6	1.4	0.2	setosa	setosa
6	5.4	3.9	1.7	0.4	setosa	setosa
7	4.6	3.4	1.4	0.3	setosa	setosa
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

vii) Click the '**Summary**' tab to display the summary.



14.1.2.6. Catboost Regression

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



ii) The '**Custom Group**' tab opens.

a. **Dynamic Fields**

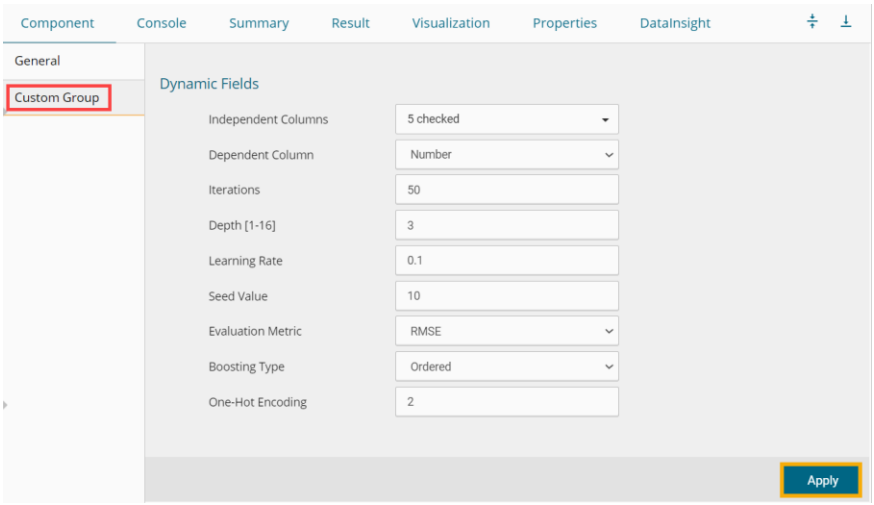
i. **Dependent Column**: Select the target column on which the regression analysis gets applied.

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

iii. Iterations: Provide iteration value.

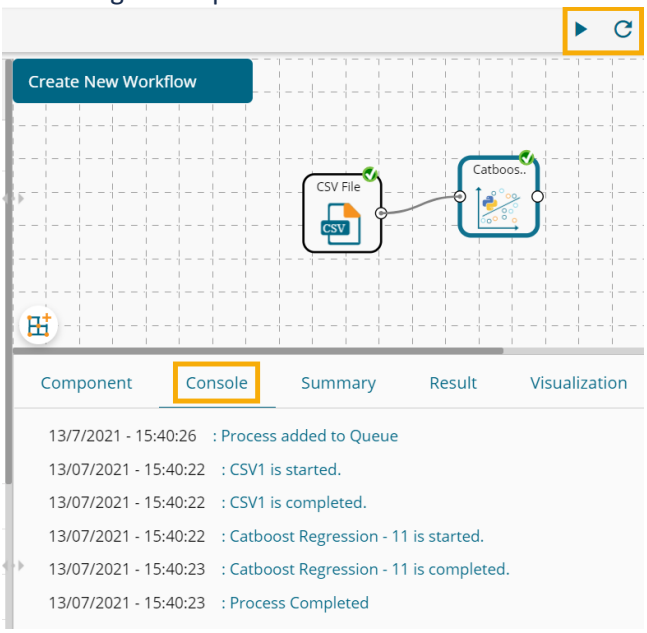
iv. Depth [1-16]: Provide a depth value.
v. Learning Rate: Provide learning rate value.
vi. Seed Value: Provide a number to generate the seed value.
vii. Evaluation Metric: Select an option out of the given choices: RMSE, LogLinQuantile, MAPE, Poisson, Quantile, MSLE, R2, MedianAbsoluteError.
viii. Boosting Type: Select the boosting type out of the '**Ordered**' and '**Plain**' options.
ix. One-Hot Encoding: Provide value to encode the integer representation in the model.

iii) Click the '**Apply**' option.

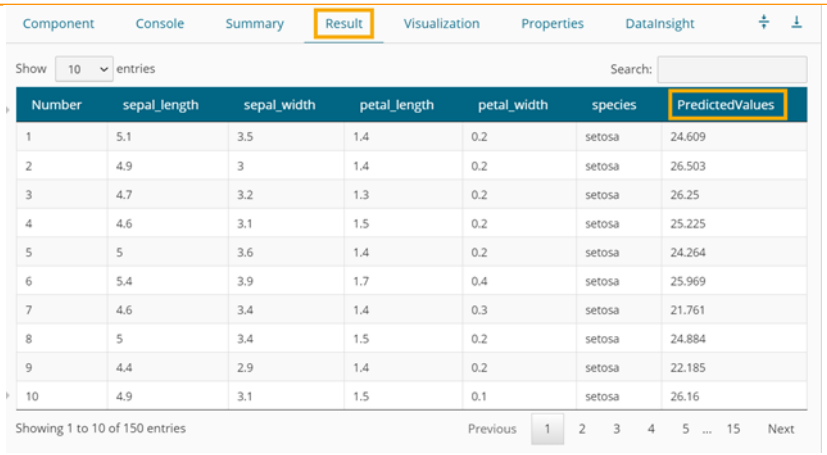


iv) Run the workflow after getting the success message.

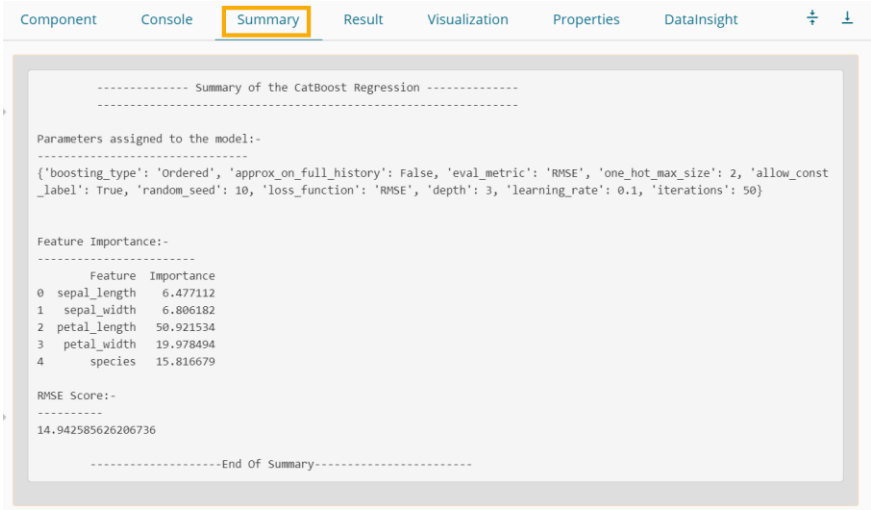
v) The user gets the process status under the '**Console**' tab.



vi) Follow the below-given steps to display the Result view:
a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.
c. A new column displaying the predicted values gets added to the result view.



vii) Click the '**Summary**' tab to display the summary.



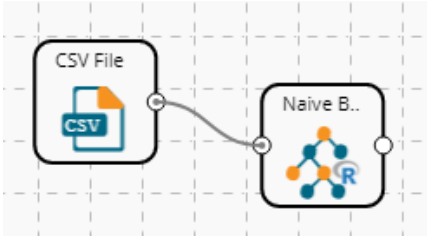
14.1.3. Classification

14.1.3.1. 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**.

Naive Bayes is a leaf node under Classification algorithms under the Algorithm tree node. The component consists of one node for reading data from a data source and another one for giving the Result.

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



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

a. Column Selection

i. **Feature**: Select input columns from the drop-down menu to which the target variable can be compared to performing the analysis.

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

b. Output Information

i. **Show Probability:** Select an option out of True or False (Selecting the 'True' option displays the Probability Column Name field under the 'New Column Information' section).

c. New Column Information

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

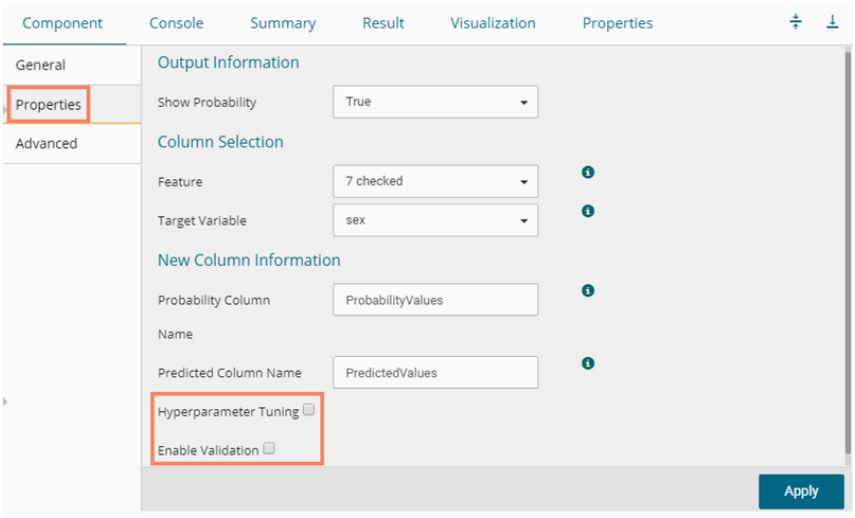
ii. **Predicted Column Name**: Enter a name for the new column containing the predicted values.

d. Hyperparameter Tuning: Apply Hyperparameter Tuning for the model by using a checkmark in the given box.

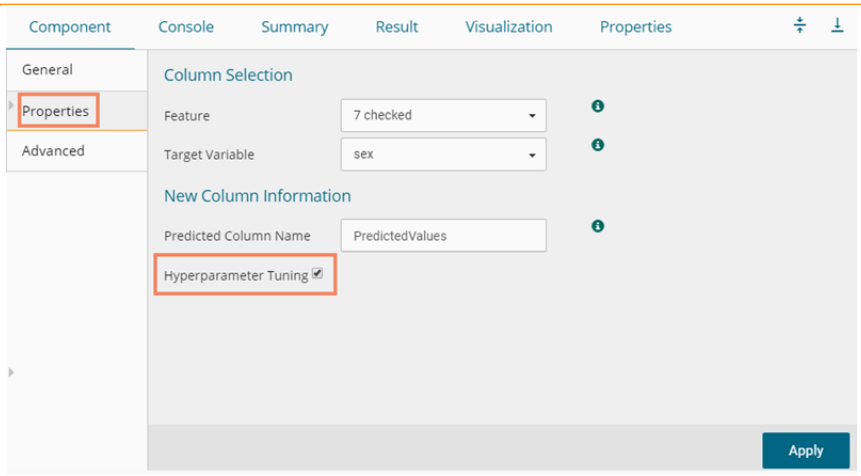
e. Enable Validation: Enable validation by using a checkmark in the given box.

There are three scenarios for the Properties tab to get configured:

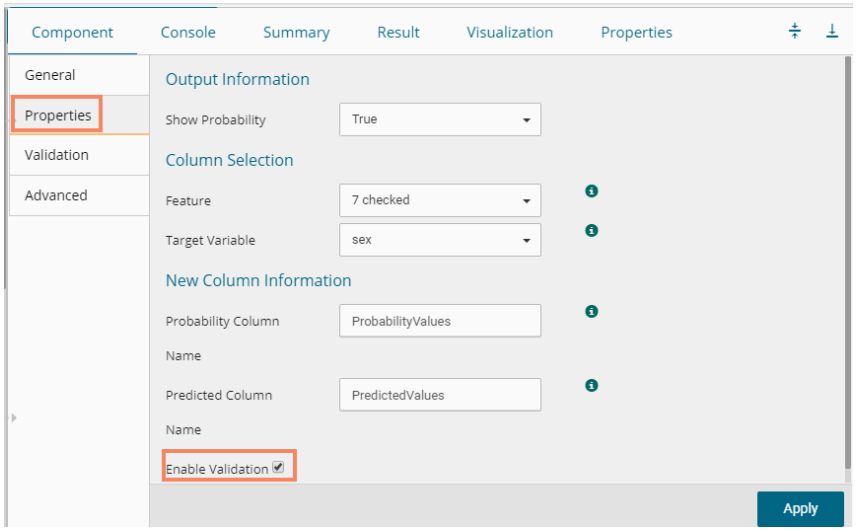
1. Hyperparameter Tuning and Validation are disabled.



2. Hyperparameter Tuning is Applied



3. Validation is enabled

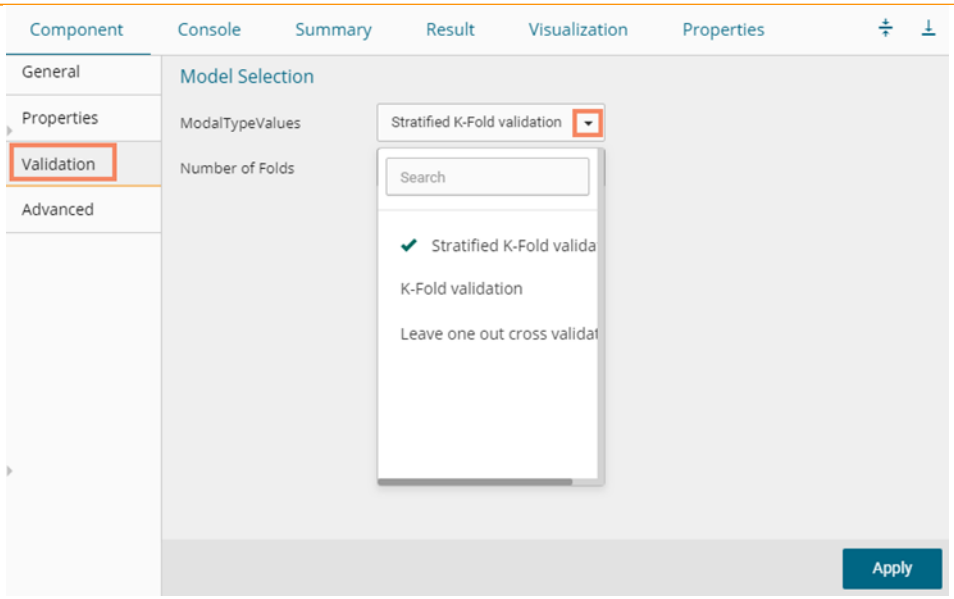


iii) Click the '**Validation**' tab to configure, if it has been enabled from the Component Properties tab. The '**Validation**' tab provides multiple options under the '**Model Type Values**' drop-down menu. The user can select any one of the available options to configure the Validation tab.

a. Model Selection

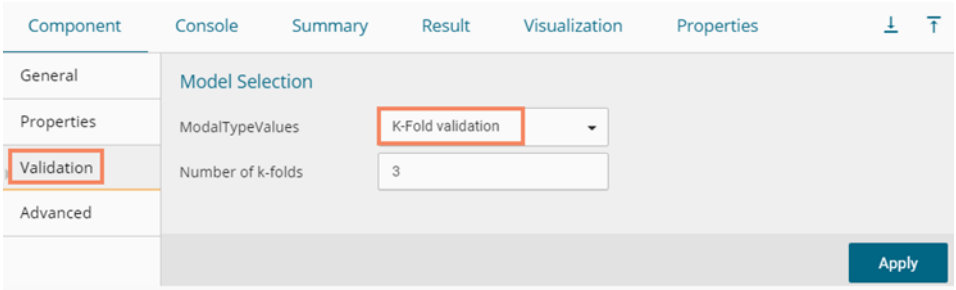
i. **Stratified K-fold Validation**

The user needs to configure the **'Number of Folds'** fields if the selected Model Type Value is **Stratified K-fold Validation.**



ii. **K-fold Validation**

The user needs to configure the **'Number of k-folds'** field if the selected Model Type Values is **K-Fold Validation.**



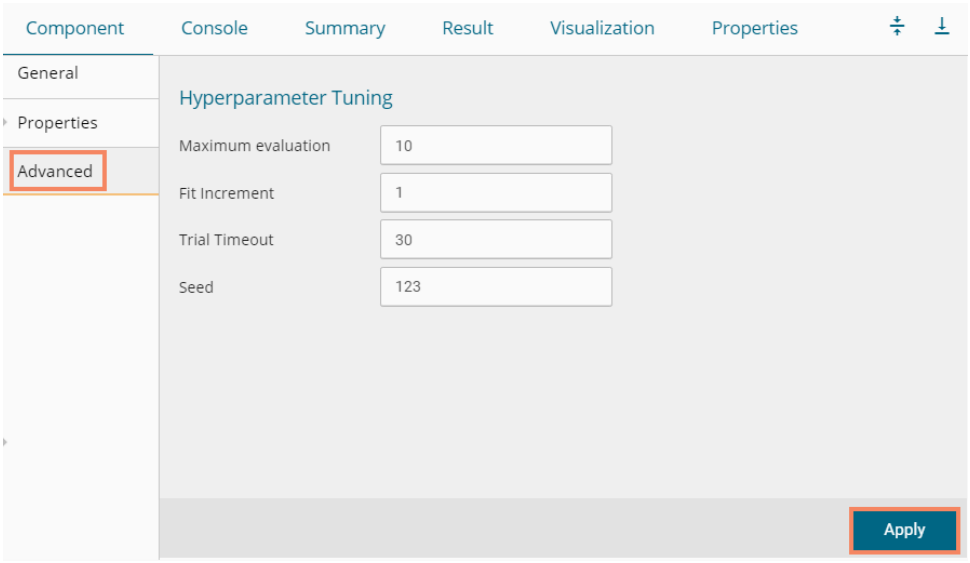
iii. **Leave One Out Cross-Validation**

The user gets to configure no other fields when the selected Model Type Values option is **Leave One Out Cross-Validation**.

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

- **Advanced Tab when 'Hyperparameter Tuning' is Enabled**
  - **a. Hyperparameter Tuning**

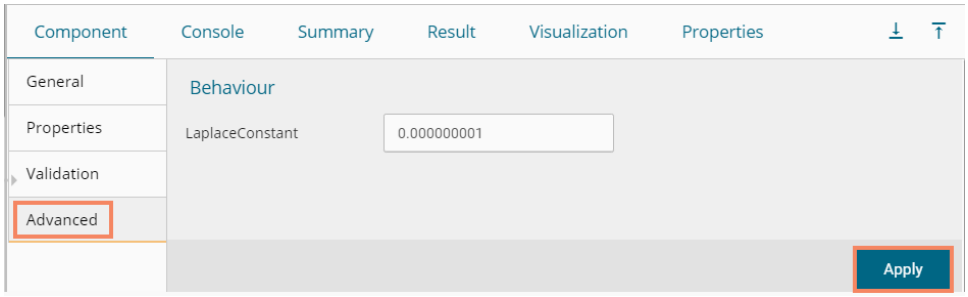
i. **Maximum Evaluation:** Enter optimal evaluation value for defining hyperparameters to search for the ideal model architecture. The default value for this field is 10.
ii. **Fit Increment:** Provide increment value for Hyperparameter model tuning. The default value for this field is 1.
iii. **Trial Timeout:** Set value for the trial timeout field by providing a number. The default value for this field is 30.
iv. **Seed:** Provide value to configure the seed field. The default value for this field is 123.
v. Click the **'Apply'** option.



- **Advanced Tab when 'Validation' is Enabled**

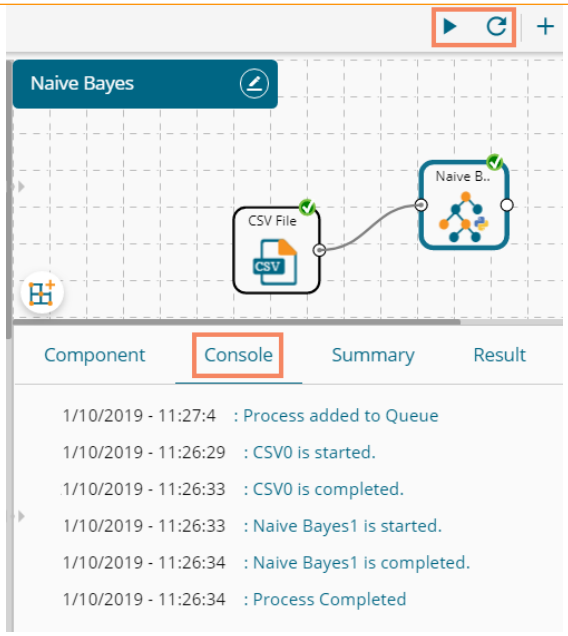
a. Behavior

i. **Laplace Constant:** Enter the smoothing constant for smoothing observations. The smoothing constant must be a double value greater than 0. Entering 0 disables Laplace smoothing.
ii. Click the **'Apply'** option.



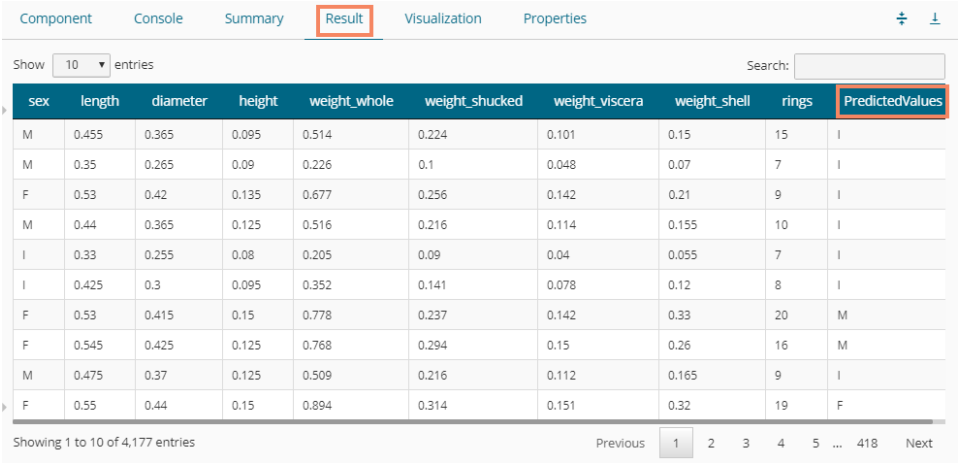
Note: The same field appears when Validation and Hyperparameter Tuning are disabled.

v) Run the workflow and after getting the success message.
vi) The '**Console**' tab opens displaying the steps of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

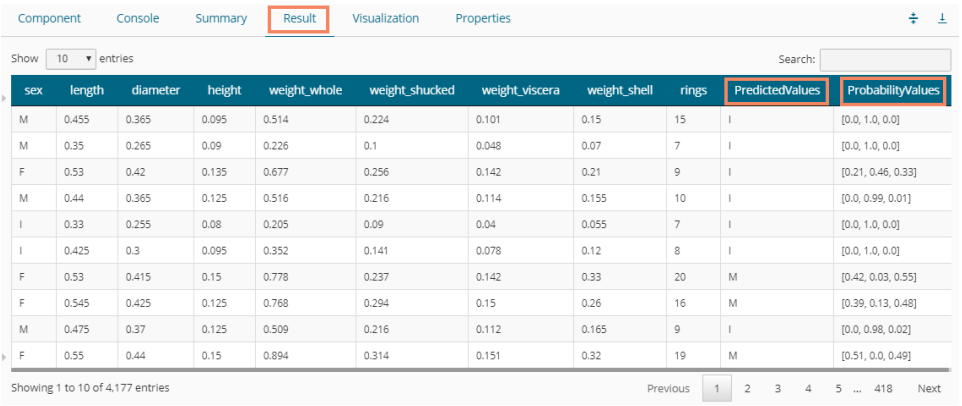


vii) Click the '**Result**' tab to display the dataset in the result view.

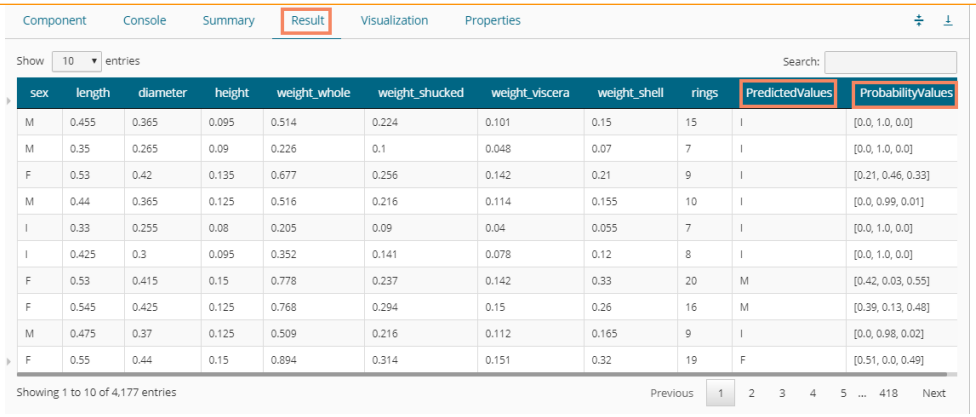
i. **Result View with Validation disabled.**



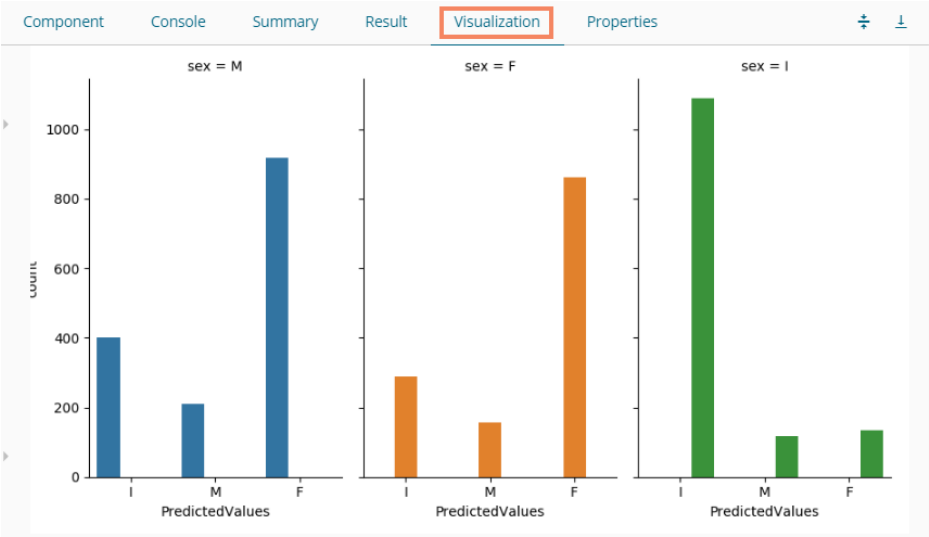
ii. **Result View with Validation enabled.**



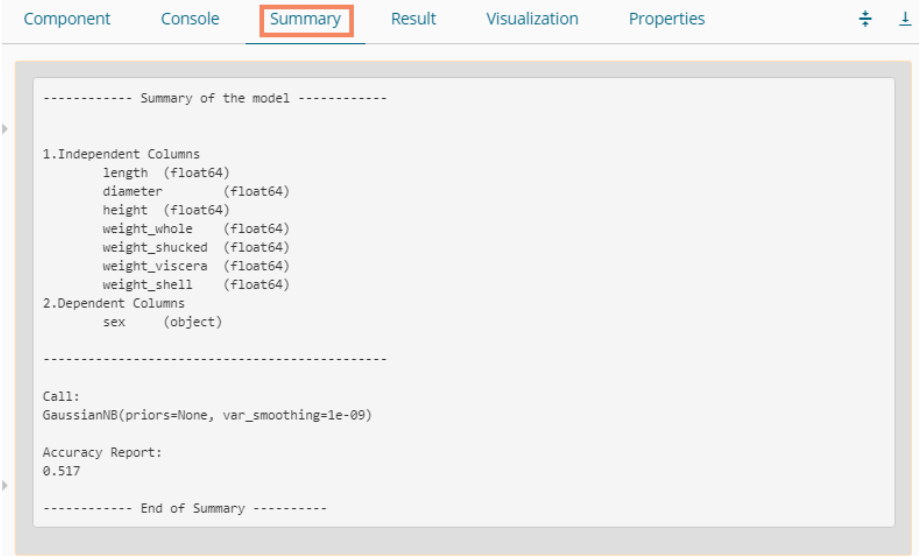
iii. **Result View with Validation and Hyperparameter disabled**



viii) Click the '**Visualization**' tab to see the processed data in the comparative Column charts (the current visualization displays the processed data when '**Validation**' is enabled).

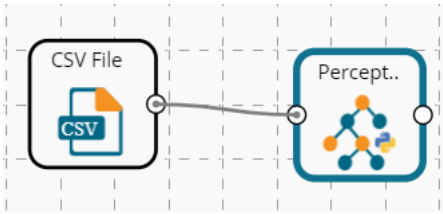


ix) Click the **'Summary'** tab to see the detailed Model Summary.



14.1.3.2. Perceptron

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

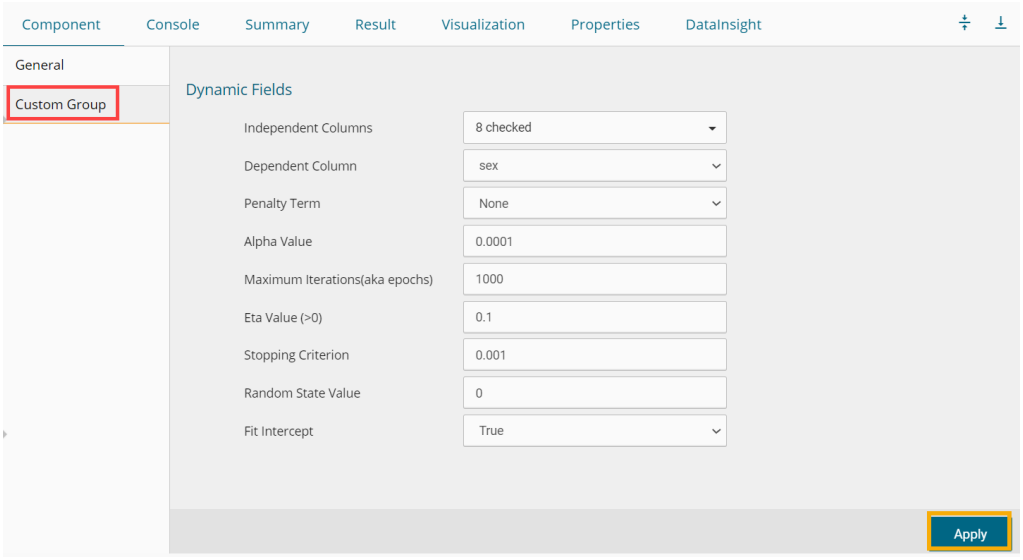


ii) The '**Custom Group**' tab opens.

a. Provide the required details for the **Dynamic Fields**

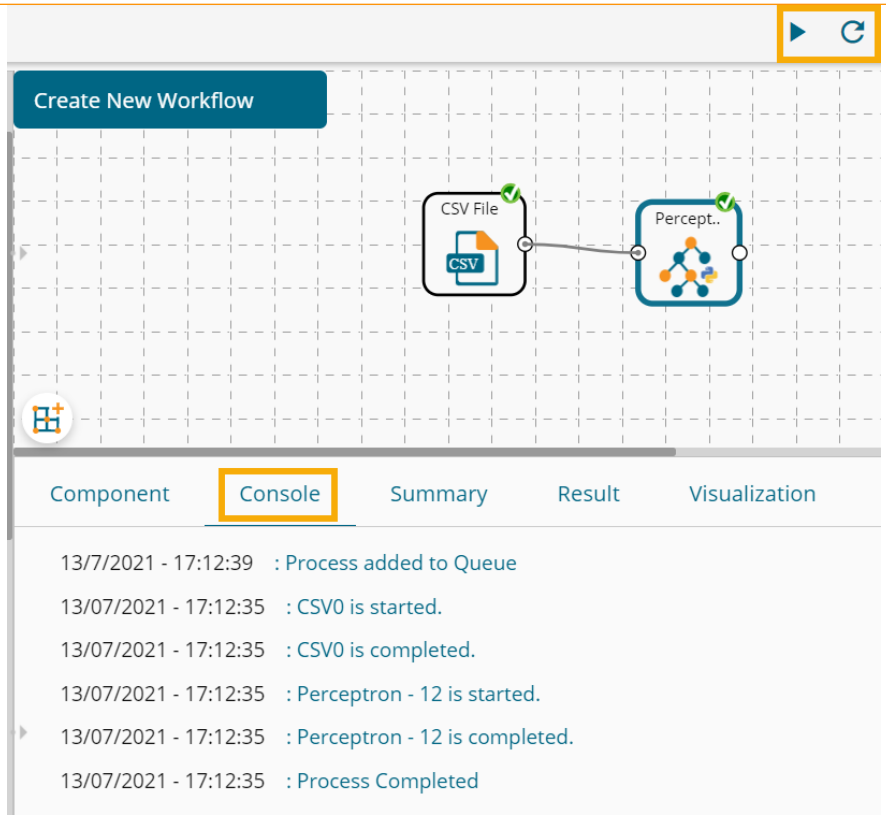
i. **Independent Column**: Select the required input columns against which the regression analysis gets applied to the target column.
ii. **Dependent Column**: Select the target column on which the regression analysis gets applied.
iii. Penalty Term
iv. Alpha Value
v. Maximum Iterations (aka epochs)
vi. Eta Value (>0)
vii. Stopping Criterion
viii. Random State Value
ix. Fit Intercept

iii) Click the '**Apply**' option.



iv) Run the workflow after getting the success message.

v) The user gets the process status under the '**Console**' tab.

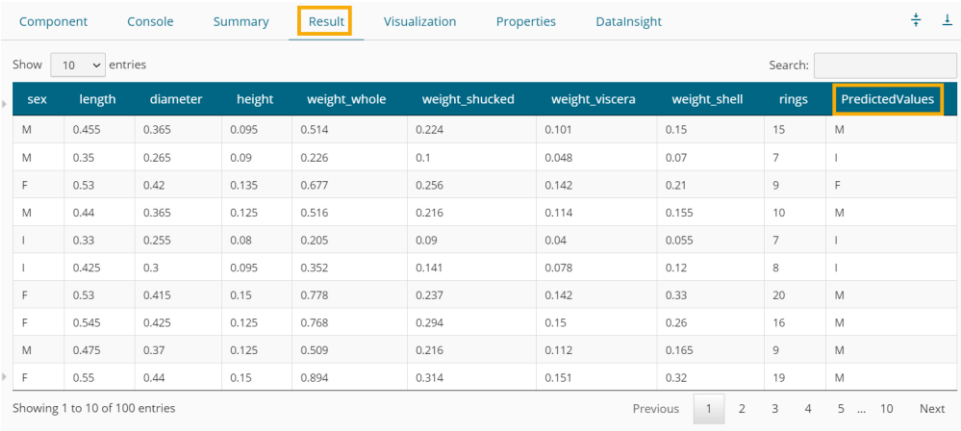


vi) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

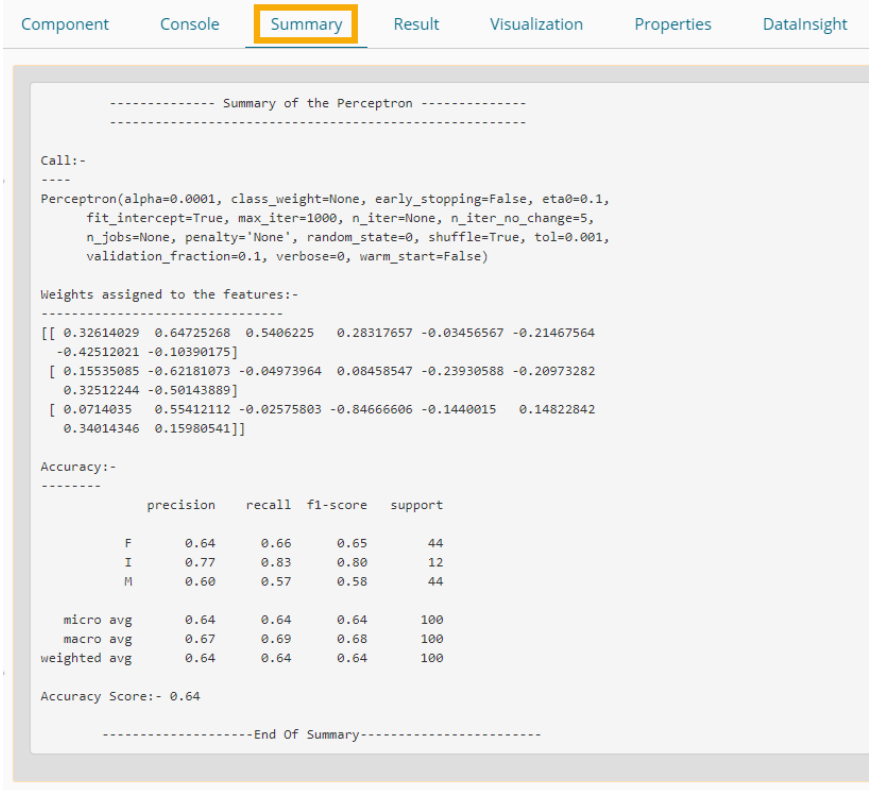
b. Click the '**Result**' tab.

c. A new column displaying the predicted values gets added to the result view.



sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues
M	0.455	0.365	0.095	0.514	0.224	0.101	0.15	15	M
M	0.35	0.265	0.09	0.226	0.1	0.048	0.07	7	I
F	0.53	0.42	0.135	0.677	0.256	0.142	0.21	9	F
M	0.44	0.365	0.125	0.516	0.216	0.114	0.155	10	M
I	0.33	0.255	0.08	0.205	0.09	0.04	0.055	7	I
I	0.425	0.3	0.095	0.352	0.141	0.078	0.12	8	I
F	0.53	0.415	0.15	0.778	0.237	0.142	0.33	20	M
F	0.545	0.425	0.125	0.768	0.294	0.15	0.26	16	M
M	0.475	0.37	0.125	0.509	0.216	0.112	0.165	9	M
F	0.55	0.44	0.15	0.894	0.314	0.151	0.32	19	M

vii) Click the '**Summary**' tab to display the summary.



14.1.3.3. Catboost Classification

i) Drag the Catboost Classification component to the workspace and connect it to a configured data source.

ii) The '**Custom Group**' tab opens.

a. Dynamic Fields

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

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

iii. Iteration**:** Set the iteration value

iv. Depth [1-16]: Set depth. it should be between 1 to 16 (wherein 1 and 16 numbers are also included)

v. Learning Rate: Provide Learning rate value

vi. Loss: Select an option from out of Logloss or Multiclass

vii. Evaluation Metric: Set evaluation metric out of the given choices: Logloss, Crossentropy, Precision, F1, Recall, BalancedAccuracy, BalancedErrorRate

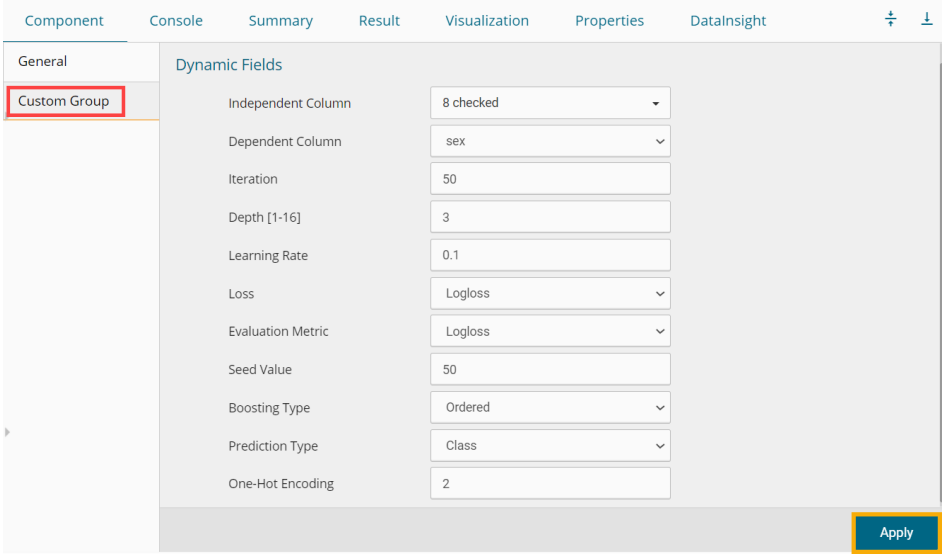
viii. Seed Value: Set a seed value.

ix. Boosting Type: Select a boosting type out of the given choices: Ordered and Plain.

x. Prediction Type: Select a Prediction Type option out of Class, RowFormulaVal, and Probability.

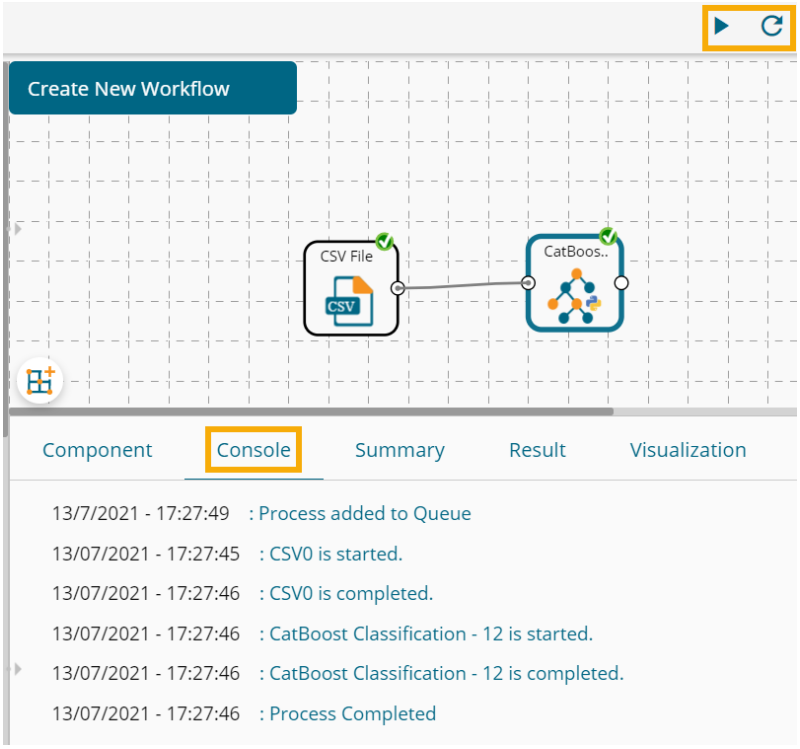
xi. One-Hot Encoding: Provide a value to encode the data.

iii) Click the '**Apply**' option.



iv) Run the workflow after getting the success message.

v) The user gets the process status under the '**Console**' tab.

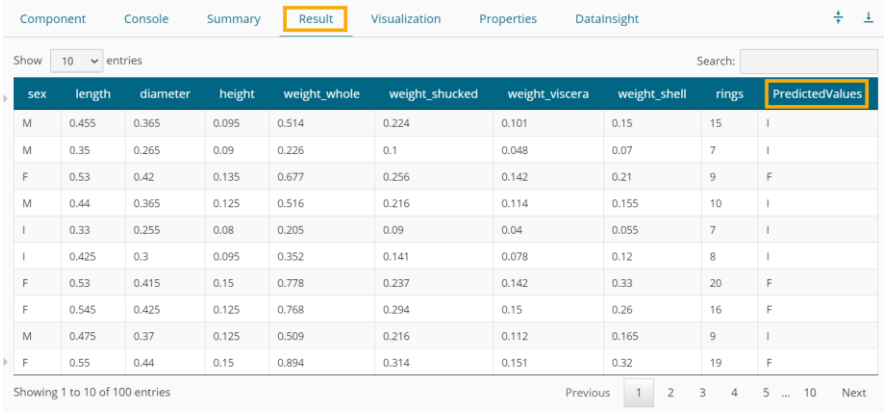


vi) Follow the below-given steps to display the Result view:

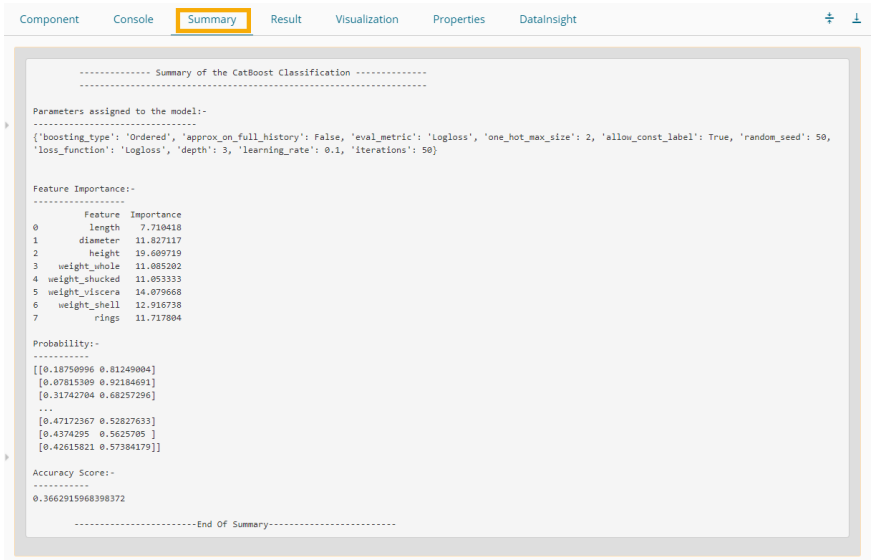
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

c. A new column displaying the predicted values gets added to the result view.



vii) Click the '**Summary**' tab to display the summary.



14.1.4. Tree-Based modeling

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

14.1.4.1. Random Forest

14.1.4.1.1. Random Forest with Classification as Algorithm Type

i) Drag the Random Forest component to the workspace and connect it with a configured data source.

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

a. Output Information

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

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

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

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

b. Column Selection

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

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

c. New Column Information

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

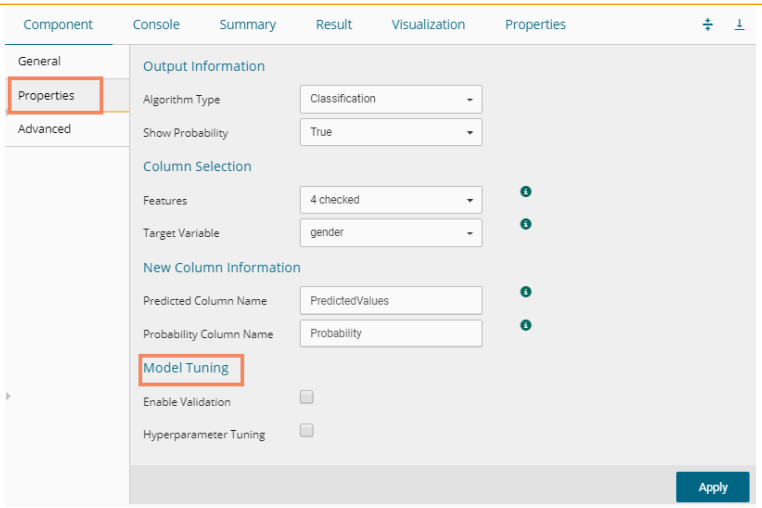
ii. **Probability Column Name:** Enter a name for the new column containing the probability values.

d. Model Tuning

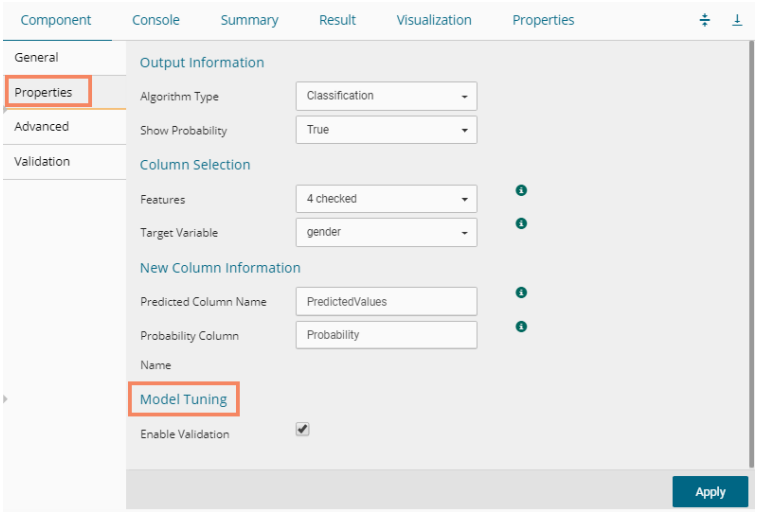
i. **Enable Validation:** Enable validation as a model tuning option by a checkmark in the given box.

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

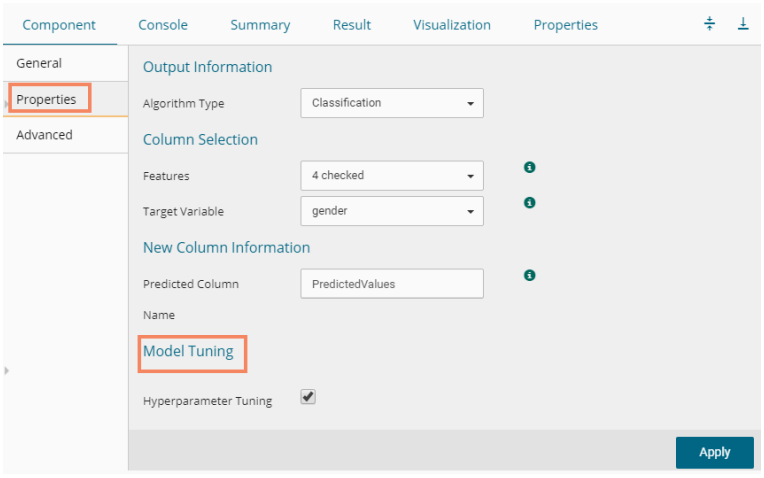
Properties Tab when Model Tunning is not Enabled



Properties Tab when Validation is Enabled as Model Tuning



Properties Tab when Hyperparameter Tuning is Enabled as Model Tuning



Note:

a. The '**Show Probability**' field appears only if, '**Classification**' option is selected via the '**Algorithm Type**' drop-down menu.
b. The '**Show Probability**' field disappears in the following scenarios:
   i. If the selected **Algorithm Type** is **Regression**
   ii. If the selected Model Tuning option is **Hyperparameter Tuning**.

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

- **Advanced Tab when both the Model Tuning options are Disabled**

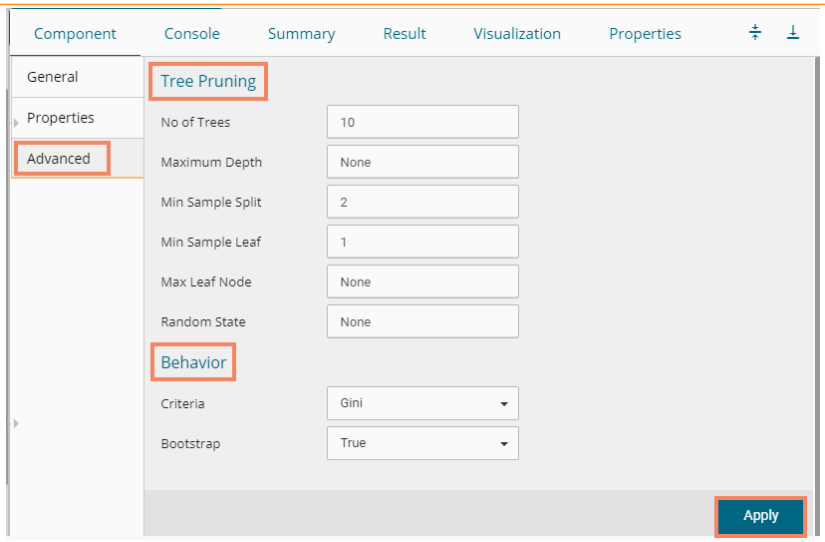
a. Tree Pruning

i. **No. of Trees:** It is a numerical value that defines the structural size of your tree. The higher number of trees gives you better performance but makes your code slower.
ii. **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.)
iii. **Min Sample 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.
iv. **Min Sample Leaf:** Leaf is the end node of a decision tree. A smaller leaf makes the model more prone to capturing noise in train data.
v. **Max Leaf Node:** Select an option from the given choices: '**int**' or '**None**' (The field is optional, and the default option for the field is '**None**').
vi. **Random State:** This parameter makes a solution easy to replicate. A definite value of random_state produces the same results if given with the same parameters and training data. The default value for this field is **None**.

b. Behavior

i. **Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'** tab.
   The splitting index can be:
   1. **Gini:** Select this option to measure inequality among values of randomly chosen elements from a set.
   2. **Entropy:** Select this option to measure impurities for exploratory analysis.
ii. **Bootstrap:** Select an option from the drop-down menu out of True/False (the default value for this field is '**True**').

iv) Click the **'Apply'** option**.**



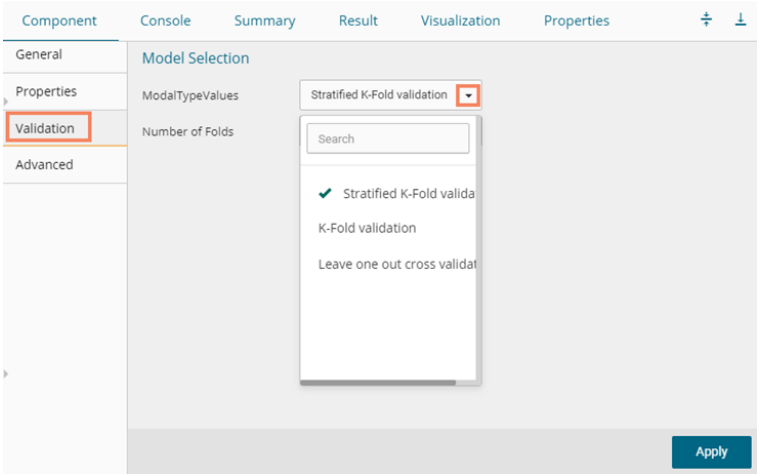
Note: The '**Advanced**' tab remains the same as displayed when both the model tuning options are disabled or when Validation is enabled.

v) Click the '**Validation**' tab and configure the required fields.

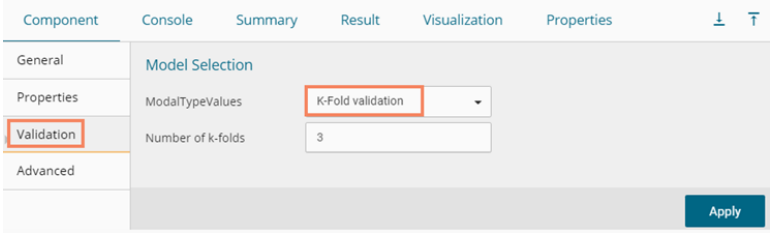
vi) Click the '**Validation**' tab to configure, if it has been enabled from the Component Properties tab. The '**Validation**' tab provides multiple options under the '**Model Type Values**' drop-down menu. The user can select any one of the available options to configure the Validation tab.

a. Model Selection

i. **Stratified K-fold Validation**
The user needs to configure the **'Number of Folds'** fields if the selected Model Type Value is **Stratified K-fold Validation.**



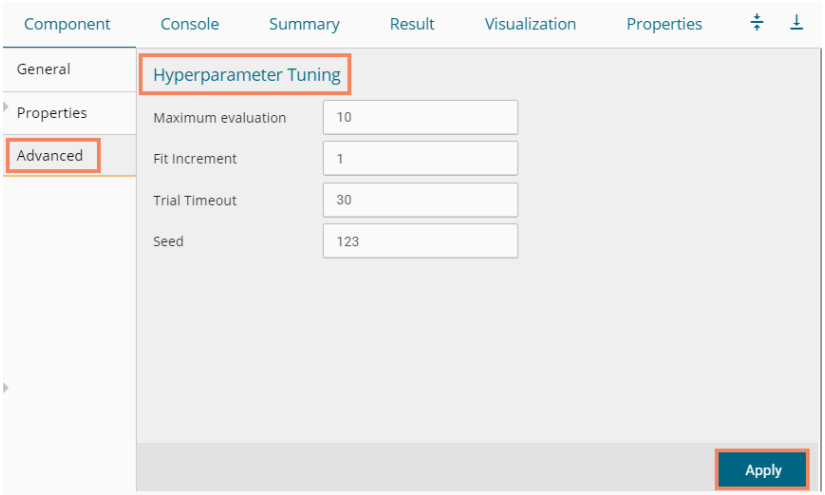
ii. **K-fold Validation**
The user needs to configure the **'Number of k-folds'** field if the selected Model Type Values are **K-Fold Validation.**



iii. **Leave One Out Cross-Validation**
The user gets to configure no other fields when the selected Model Type Values option is **Leave One Out Cross-Validation**.

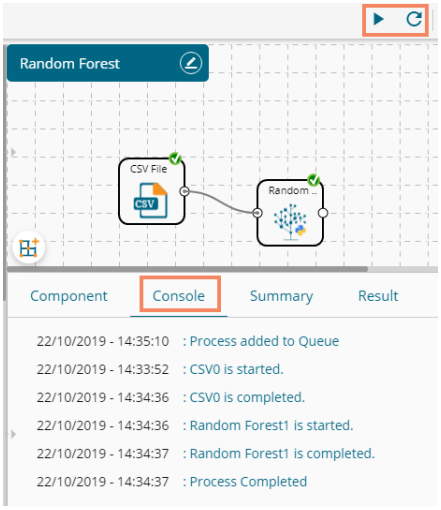
- **Advanced Tab when Hyperparameter Tuning is enabled**
  - **a. Hyperparameter Tuning**
    - i. Maximum evaluation: Provide a numerical value set to indicate the maximum value of model evaluation (The default value for this field is 10).
    - ii. Fit Increment: Provide a numerical value set as the increment to model fitting (The default value for this field is 1).
    - iii. Trial Timeout: Provide a numerical value set for the process timeout (usually in seconds (The default value for this field is 30).
    - iv. Seed: A numerical value set as the initialization state of a pseudo-random number generator (the default value for this field is 123).

vii) Click the '**Apply**' option to configure the 'Advanced' tab (if required).



viii) Run the workflow after getting the success message.

ix) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

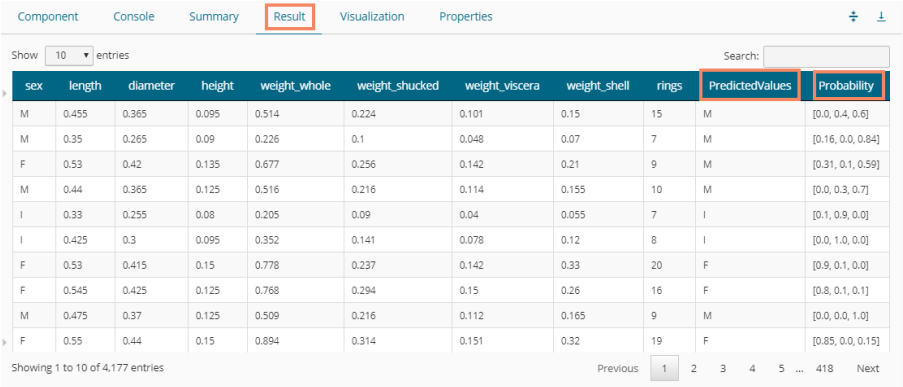


x) Follow the below-given steps to display the Result view:

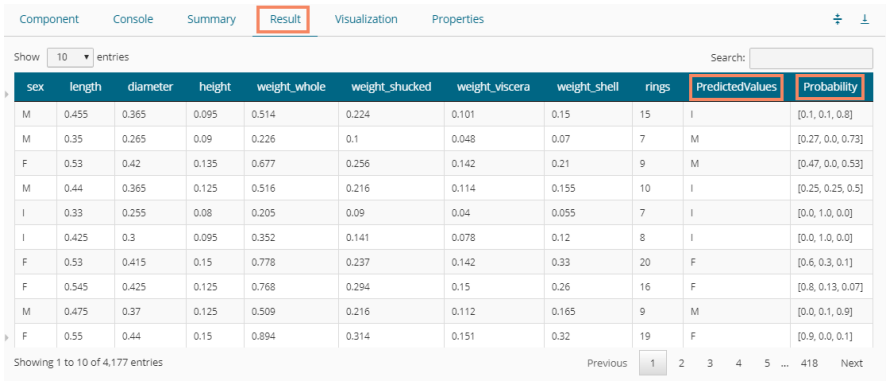
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

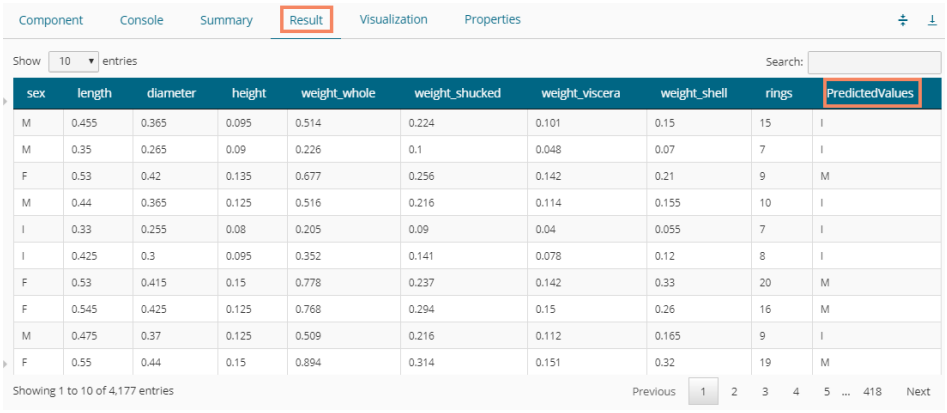
i. Result view when both the Model Tuning options are disabled.



ii. Result view with the '**Validation**' option enabled.



iii. Result view with the '**Hyperparameter Tuning**' option enabled

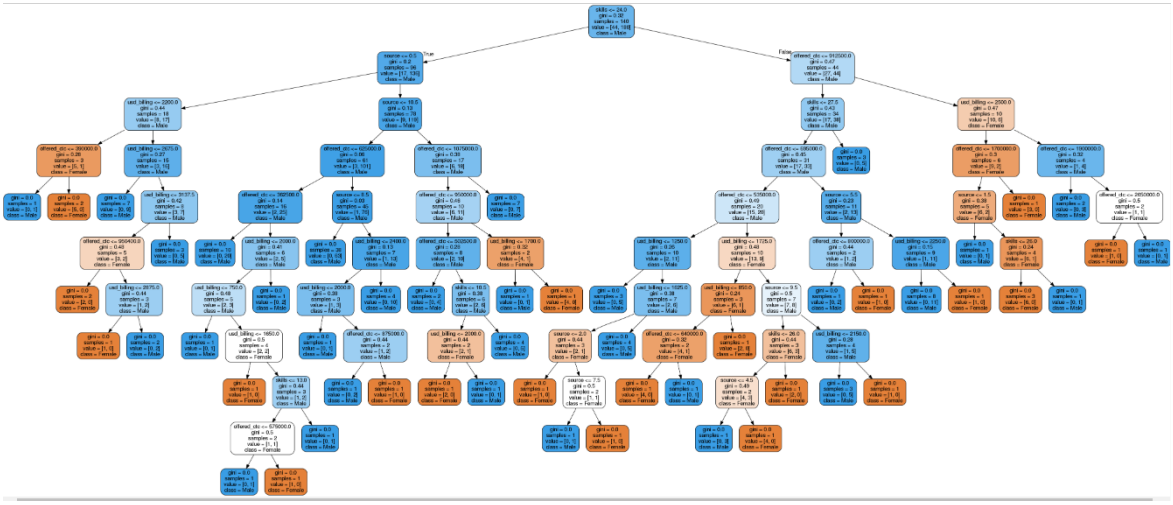


Note: The Probability column displays data in the Array format when Validation is enabled.

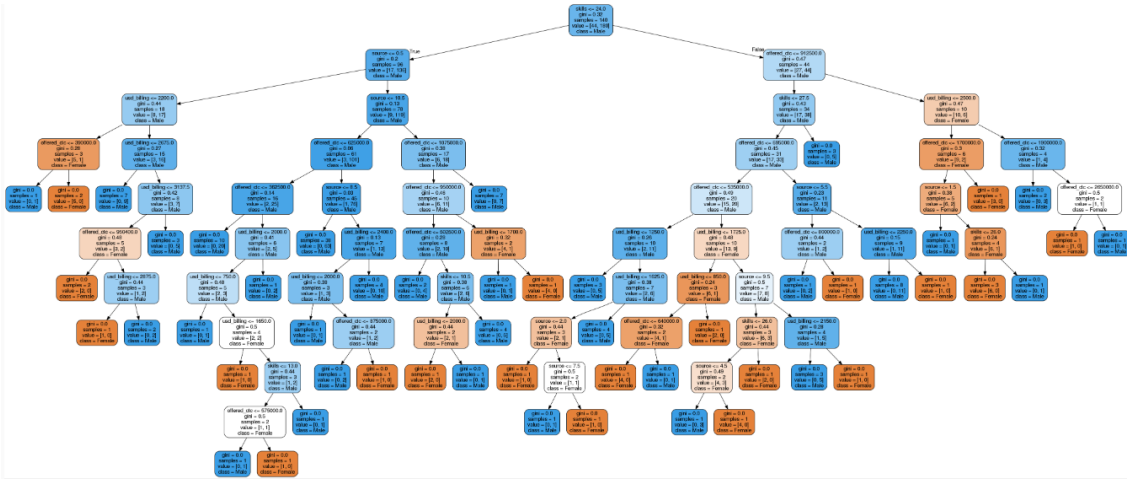
xi) Click the '**Visualization**' tab.

xii) The Result data gets displayed via the tree chart.

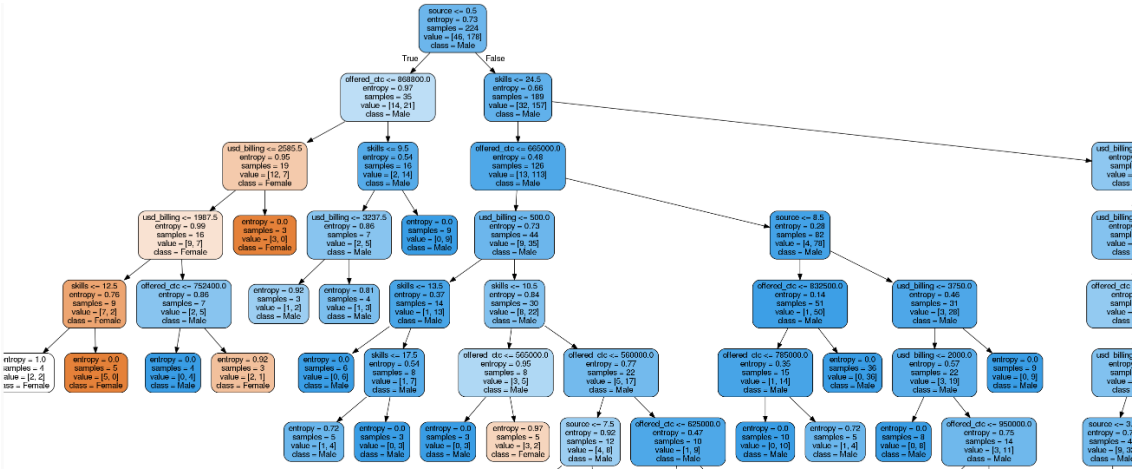
a. Visualization when no Model Tuning option is enabled



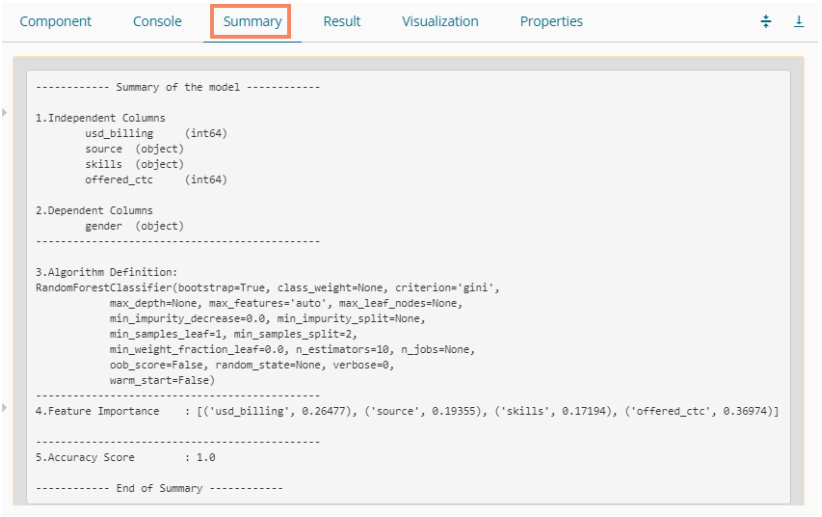
b. Visualization when the 'Validation' option is enabled



c. Visualization when Hyperparameter Tuning is enabled



xiii) Click the '**Summary**' tab to open the model summary.



14.1.4.1.2. Regression as Algorithm Type for Random Forest

i) Drag the Decision 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 the dependent column as the categorical values.
2. **Regression:** Select this option if users want to pass the dependent column as numerical values.

b. Column Selection

i. **Independent Columns**: Select input columns from the drop-down list to which the target column can be compared to perform the analysis.

ii. **Dependent Columns**: Select the target column for which the analysis is performed.

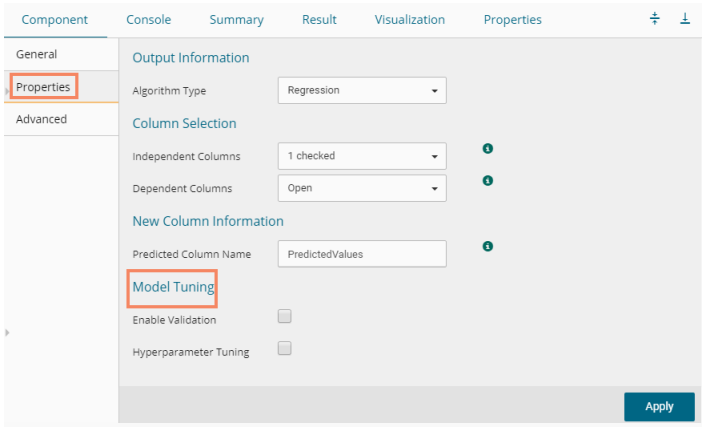
c. **New Column Information**

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

d. **Model Tuning**

i. **Enable Validation:** Enable validation by a checkmark in the given box.

ii. **Hyperparameter Tuning:** Enable the Hyperparameter Tuning option by a checkmark in the given box.



Note: Other possible scenarios to configure the Properties tab can be when either of the Model Tuning options is enabled.

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

- **Advanced Tab when both the Model Tuning options are Disabled**

a. **Tree Pruning**

i. **No. of Trees:** It is a numerical value that defines the structural size of your tree. The higher number of trees gives you better performance but makes your code slower.

ii. **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.)

iii. **Min Sample 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.

iv. **Min Sample Leaf:** Leaf is the end node of a decision tree. A smaller leaf makes the model more prone to capturing noise in train data.

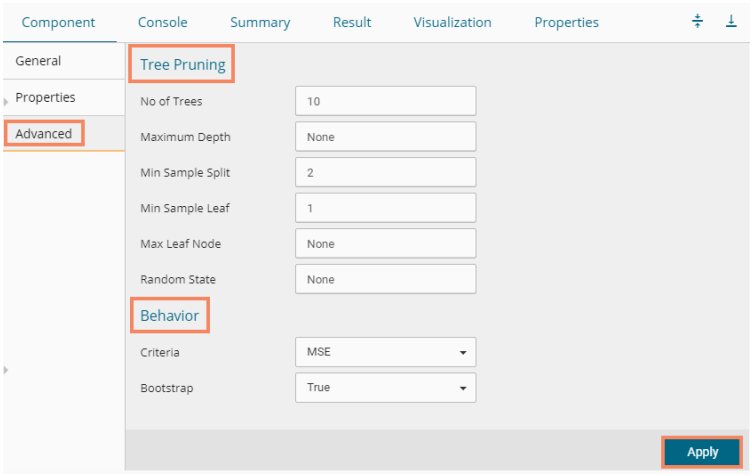
v. **Max Leaf Node:** Select an option from the given choices: '**int**' or '**None**' (The field is optional, and the default option for the field is '**None**').

vi. **Random State:** This parameter makes a solution easy to replicate. A definite value of random_state produces the same results if given with the same parameters and training data. The default value for this field is **None**.

b. **Behavior**

i. **Criteria:** It is an optional field that depends on the selected algorithm type from the **'Properties'**.
The available splitting index options are:
1. **MSE**
2. **MAE**

ii. **Bootstrap:** Select an option from the drop-down menu out of True/False (the default value for this field is '**True**').



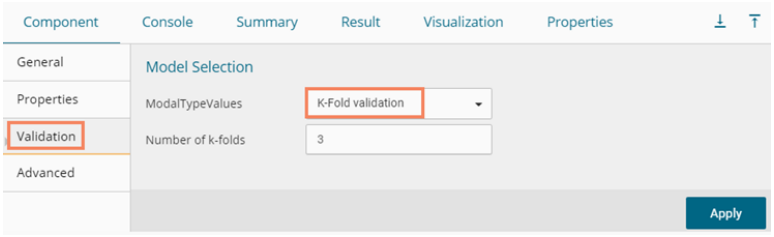
Note: The Advanced tab remains the same when '**Validation**' is enabled**.**

iv) Click the '**Validation**' tab to configure, if it has been enabled from the Component Properties tab. The '**Validation**' tab provides multiple options under the '**Model Type Values**' drop-down menu. The user can select any one of the available options to configure the Validation tab.

a. Model Selection

i. **K-fold Validation**

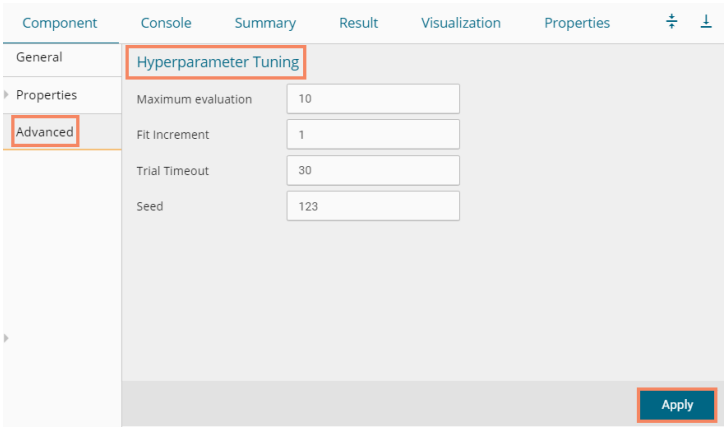
The user needs to configure the **'Number of k-folds'** field if the selected option for the **'Model Type Values**' is **K-Fold Validation.**



ii. **Leave One Out Cross-Validation**

The user gets to configure no other fields when the selected Model Type Values option is **Leave One Out Cross-Validation**.

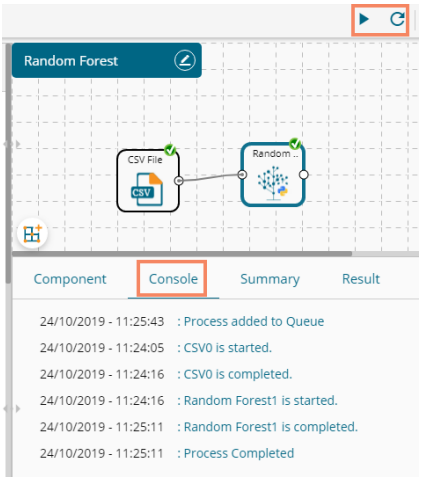
- **Advanced Tab when Hyperparameter Tuning is Enabled**
  - **a. Hyperparameter Tuning**
    - i. Maximum evaluation: Provide a numerical value set to indicate the maximum value of model evaluation (The default value for this field is 10).
    - ii. Fit Increment: Provide a numerical value set as the increment to model fitting (The default value for this field is 1).
    - iii. Trial Timeout: Provide a numerical value set for the process timeout (usually in seconds (The default value for this field is 30).
    - iv. Seed: A numerical value set as the initialization state of a pseudo-random number generator (the default value for this field is 123).



v) Click the **'Apply'** option after configuring the Properties, Advanced (if required), and validation (if enabled) tabs.

vi) Run the workflow after getting the success message.

vii) The '**Console**' tab opens.

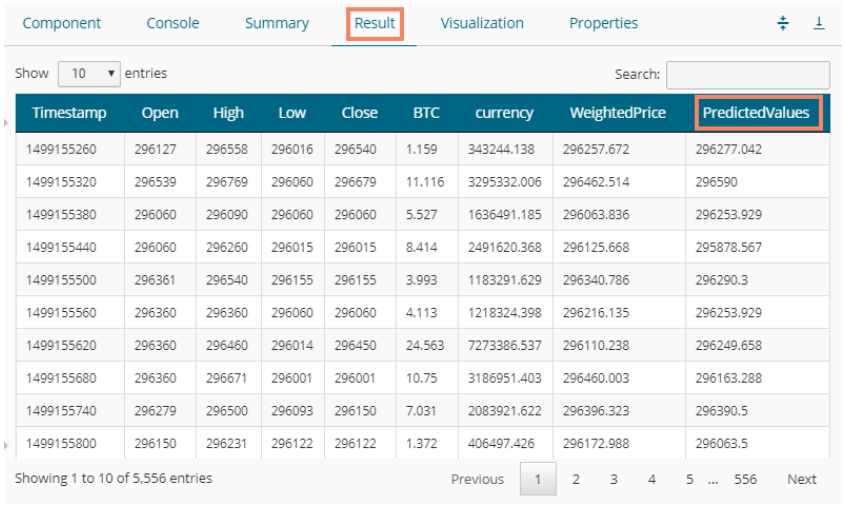


viii) Follow the below-given steps to display the Result view:

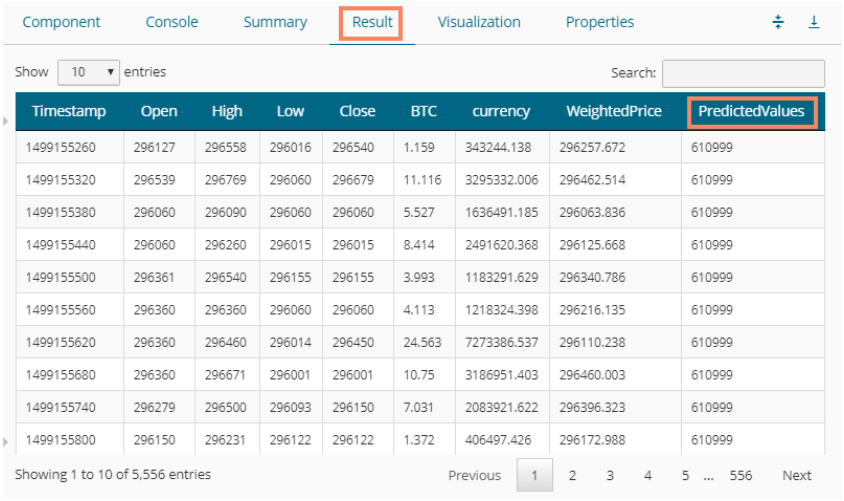
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

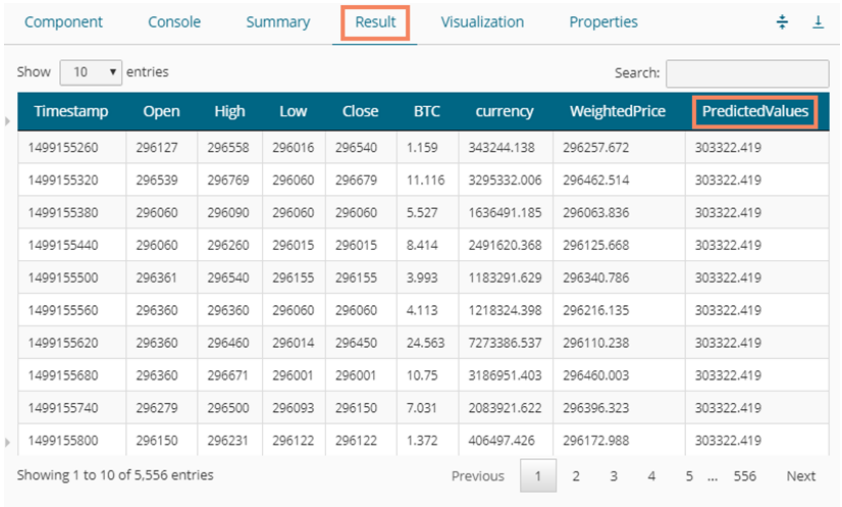
i. Result View when both the Model Tuning options are disabled



ii. Result view when the '**Validation**' option is enabled



iii. Result view when '**Hyperparameter Tuning**' is enabled.

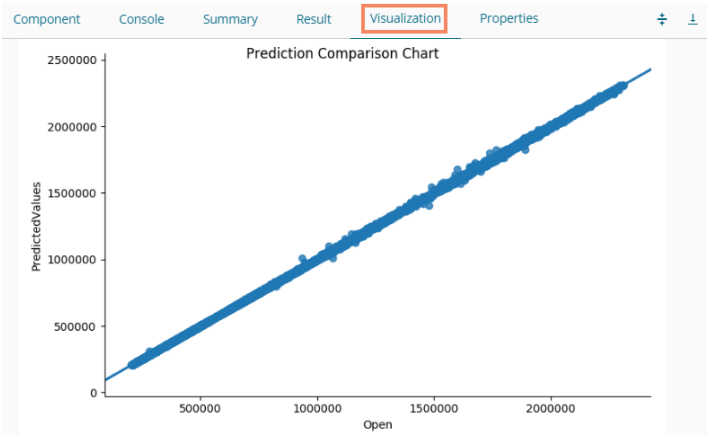


Note: The Probability column is displayed in the Array format while enabling the '**Validation**' option.

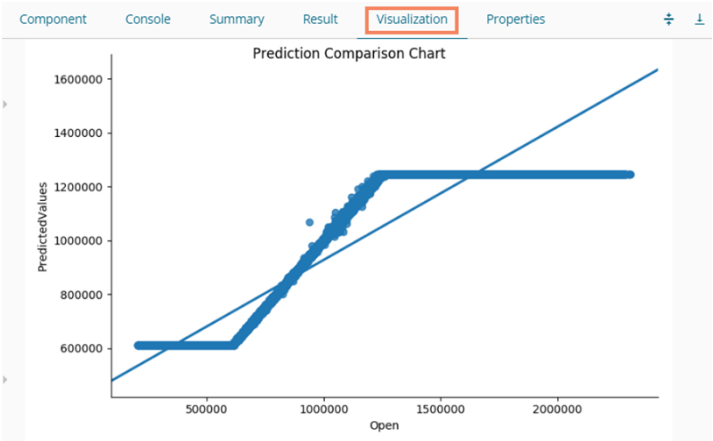
ix) Click the '**Visualization**' tab.

x) The Result data gets displayed via the tree chart (The following visualization displays result in data when no Model Tuning option is enabled).

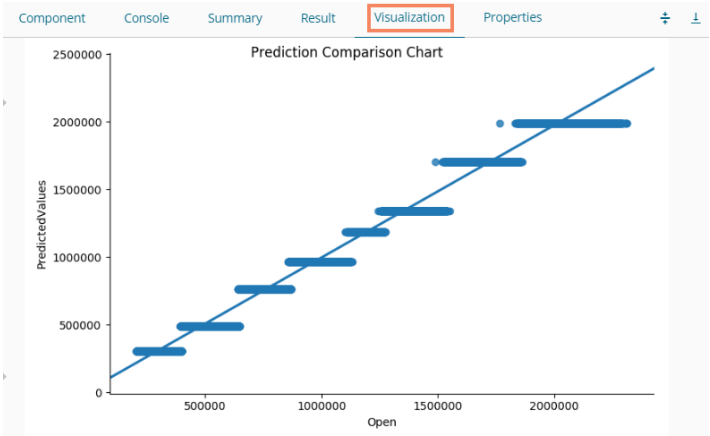
a. Visualization tab when no Model Tuning option is enabled.



b. Visualization tab when Validation is enabled.



c. Visualization tab when Hyperparameter Tuning is enabled.

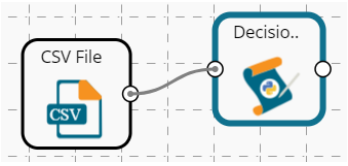


14.1.4.2. Decision Tree

14.1.4.2.1. Decision Tree with Classification Model Type

i) Drag the Decision Tree component to the workspace and connect it to a configured data source.

ii) Click the Decision Tree algorithm component.



iii) The '**Custom Group**' tab opens.

a. Dynamic Fields

i. Type of Model: Select a model out of the given choices.
   1. **Classification**: Select this option if users want to pass the dependent column as the categorical values.
   2. **Regression:** Select this option if users want to pass the dependent column as numerical values.

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

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

iv. Criterion ['mse' for Regresssion]: Select an option to measure inequality among values of randomly chosen elements from a set out of gini, entropy, or mse.

v. 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 valueless than 30 rpart for 32 bit-machines).

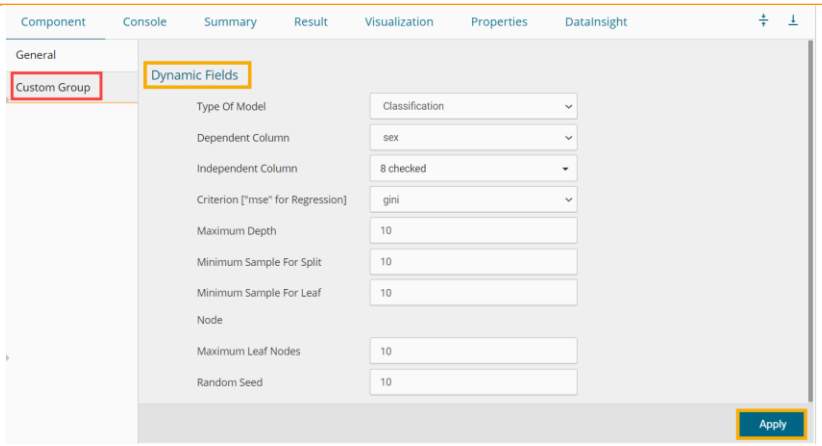
vi. Minimum Sample for Split: It indicates a minimum number of observations within a single node for a split to be attempted. The default value for this field is 10.

vii. Minimum Sample for Leaf node: It indicates the minimum number of the sample for a leaf node.

viii. Maximum Leaf Nodes: It indicates the maximum number of the sample for a leaf node.

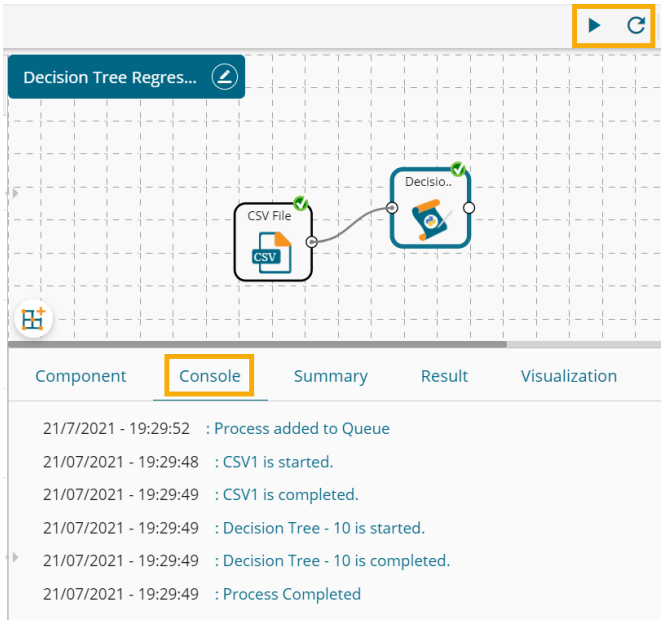
ix. Random Seed: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

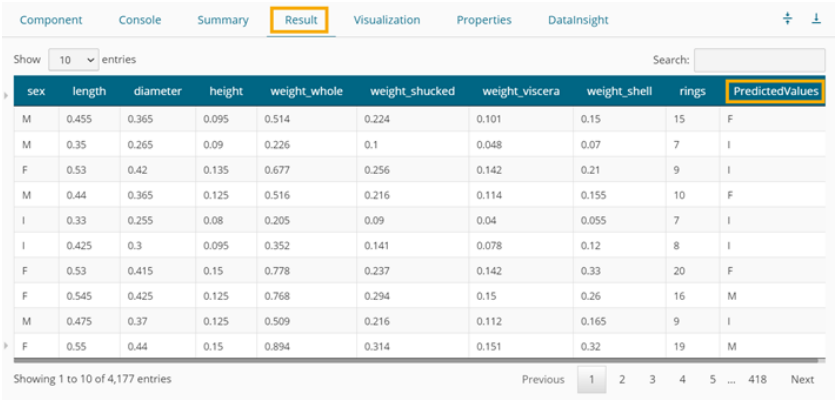
vi) The user gets the process status under the '**Console**' tab.



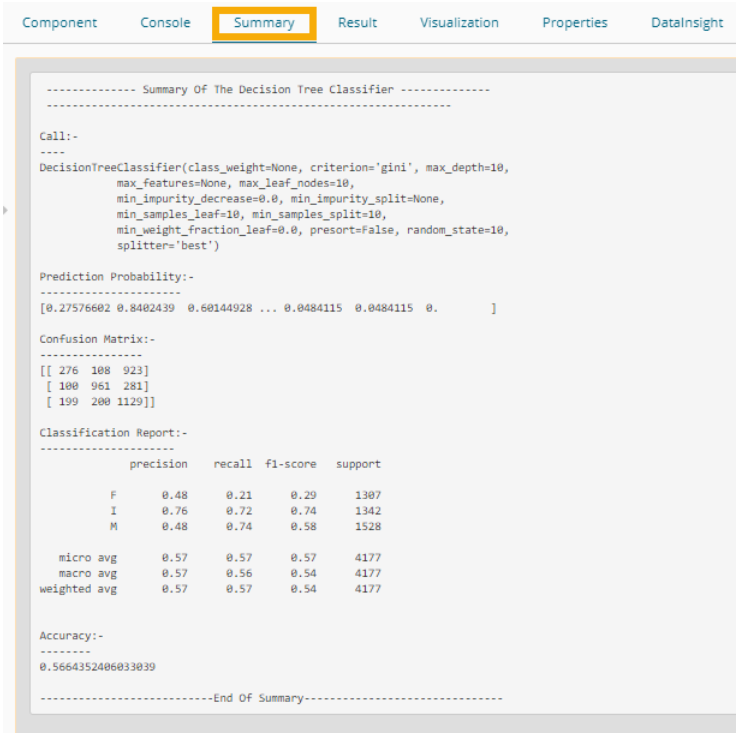
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. A new column displaying the predicted values gets added to the result view.



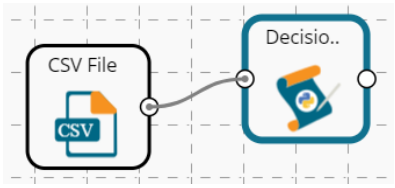
viii) Click the '**Summary**' tab to display the summary.



14.1.4.2.2. Decision Tree with Regression Model Type

i) Drag the Decision Tree component to the workspace and connect it to a configured data source.

ii) Click the Decision Tree algorithm component.

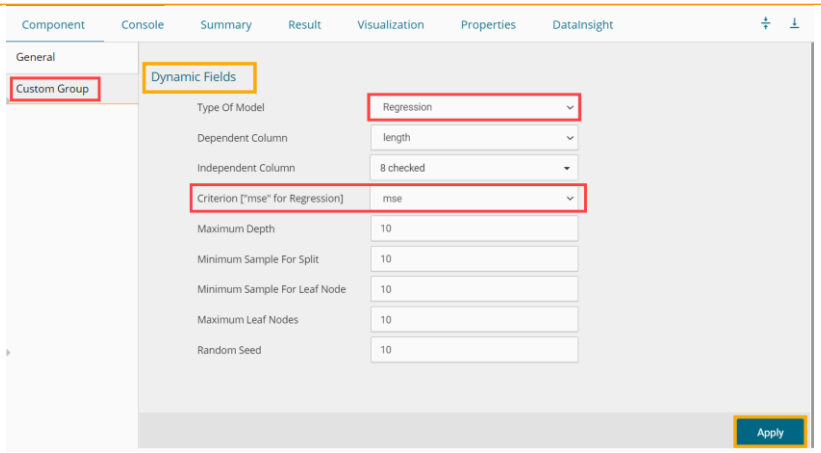


iii) The '**Custom Group**' tab opens. Configure the **Dynamic Fields** with Regression selected as the Model type.

a. Make sure that the selected Criterion option is 'mse'

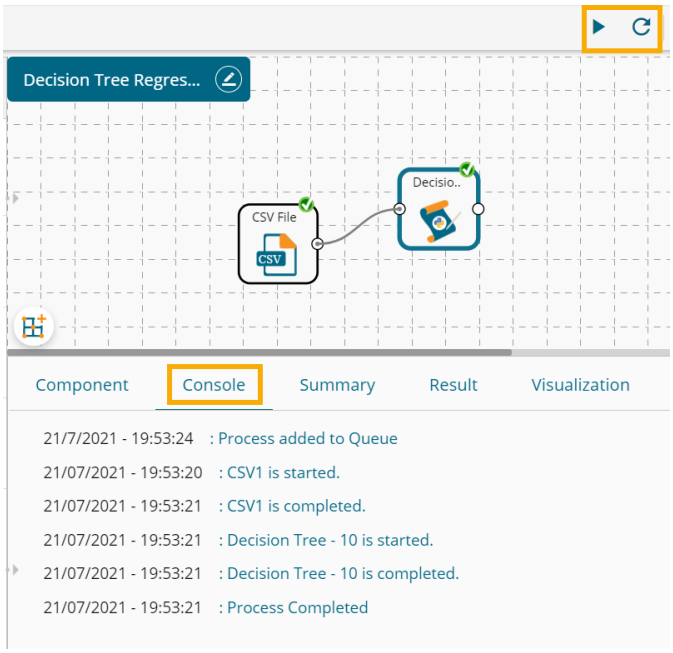
Note: All the other fields configuration remains the same.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

vi) The user gets the process status under the '**Console**' tab.

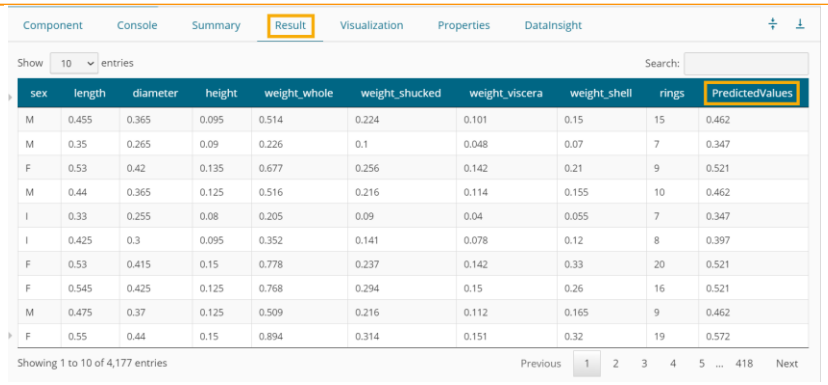


vii) Follow the below-given steps to display the Result view:

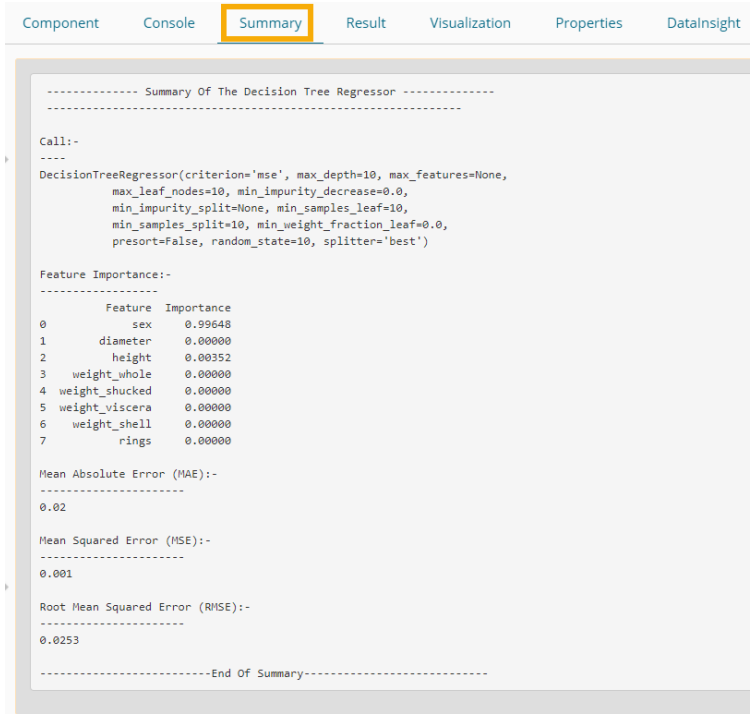
a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

c. A new column displaying the predicted values gets added to the result view.



viii) Click the '**Summary**' tab to display the summary.

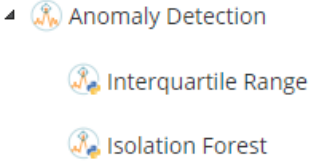


14.1.5. Anomaly Detection

Anomaly Detection is a data mining algorithm type that identifies data points, events, and/or observations that deviate from the normal behavior of a dataset.

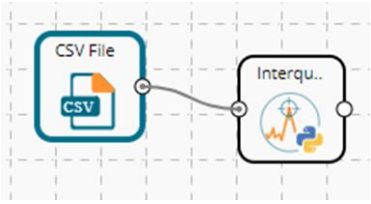
The Anomaly Detection tree node contains two sub-algorithms as shown in the image given below:

1. Interquartile Range
2. Isolation Forest



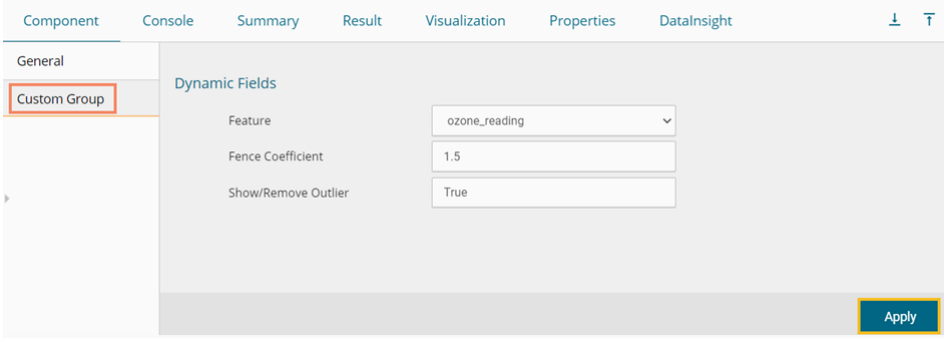
14.1.5.1. Interquartile Range

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



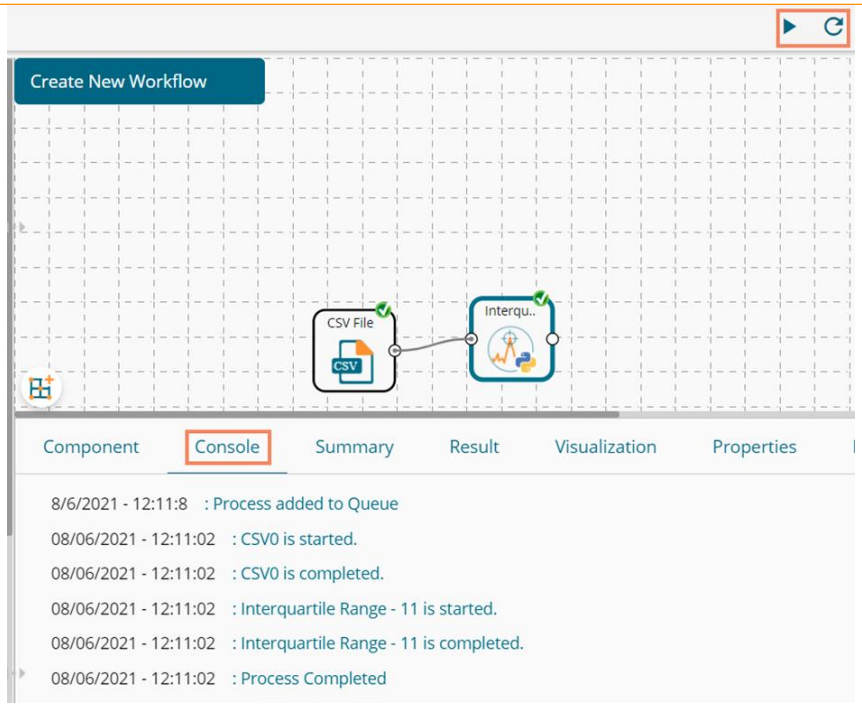
ii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a. Select a feature column.
   b. Set the fence coefficient.
   c. Select an option to either show or remove outliers.

iii) Click the '**Apply**' option.



iv) Run the workflow after getting the success message.

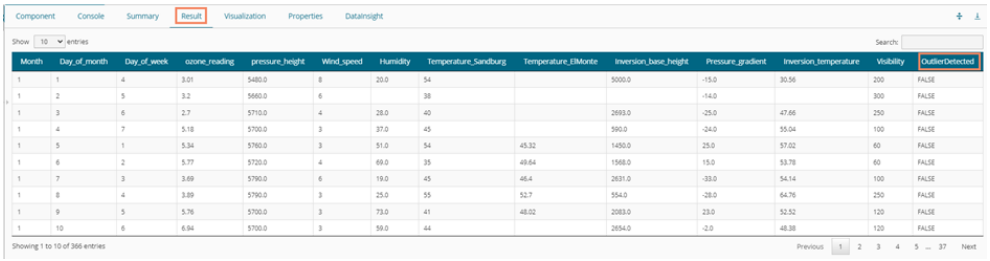
v) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



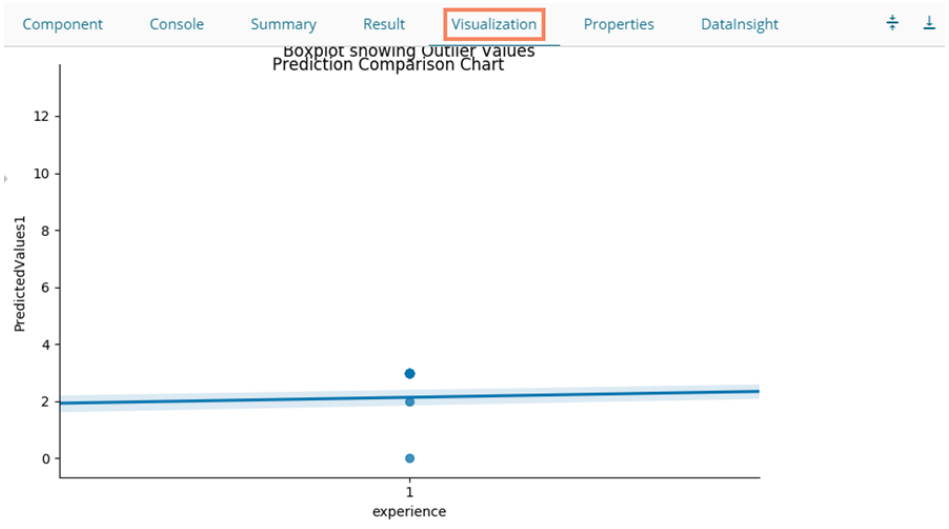
vi) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

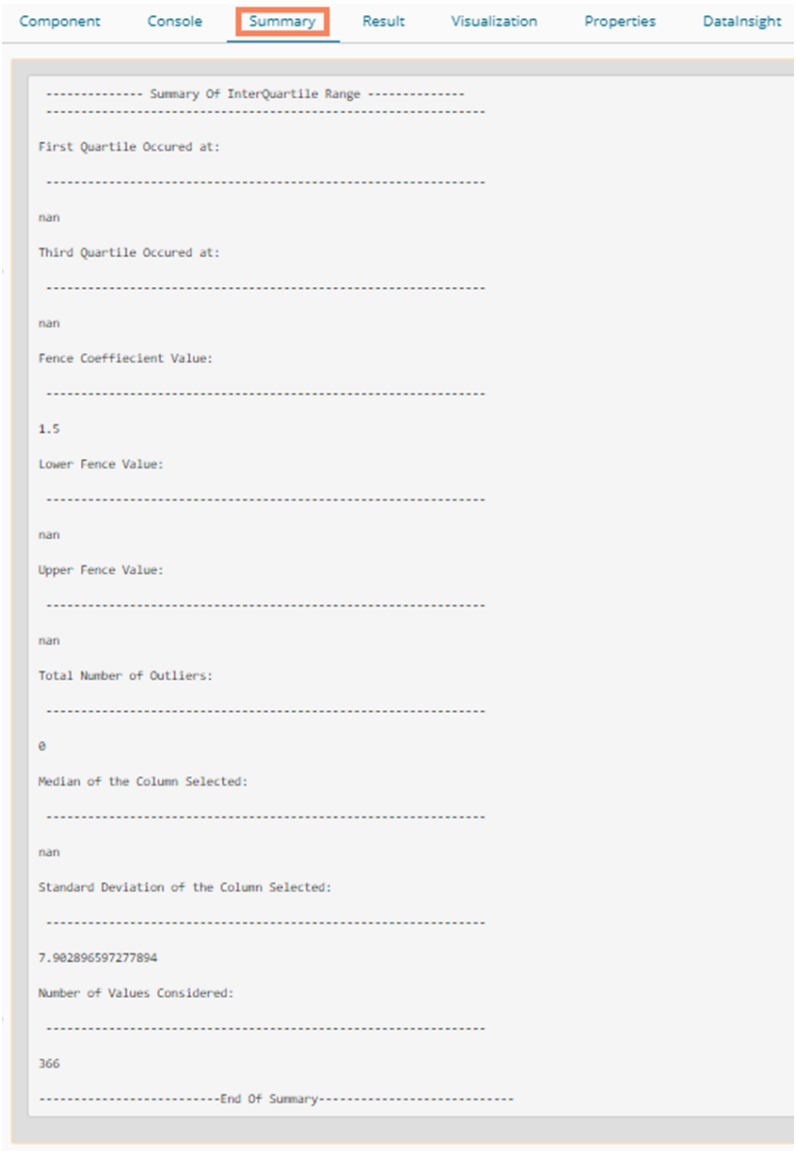
b. Click the '**Result**' tab.



vii) Click the '**Visualization**' tab. The Result data gets displayed via the Boxplot chart displaying the outlier values. It is also known as the Prediction Comparison chart. The following chart shows predicted values by experience.



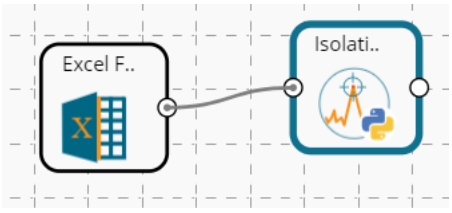
viii) Click the '**Summary**' tab to open the model summary.



14.1.5.2. Isolation Forest

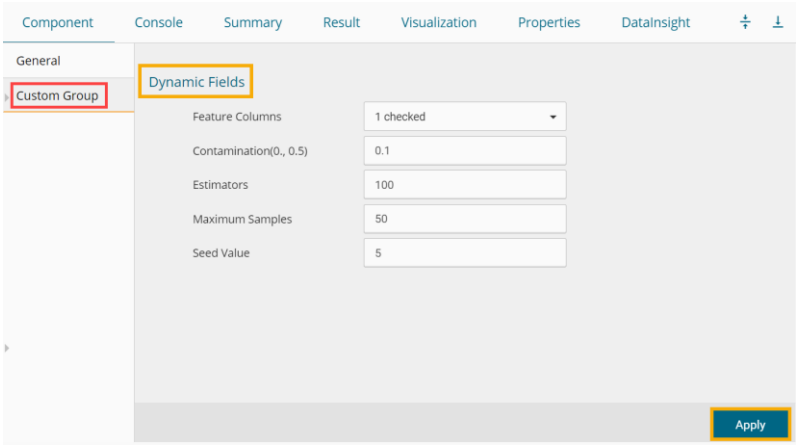
i) Drag the Isolation Forest algorithm component to the workspace and connect it with a configured data source.

ii) Click the Isolation Forest component.



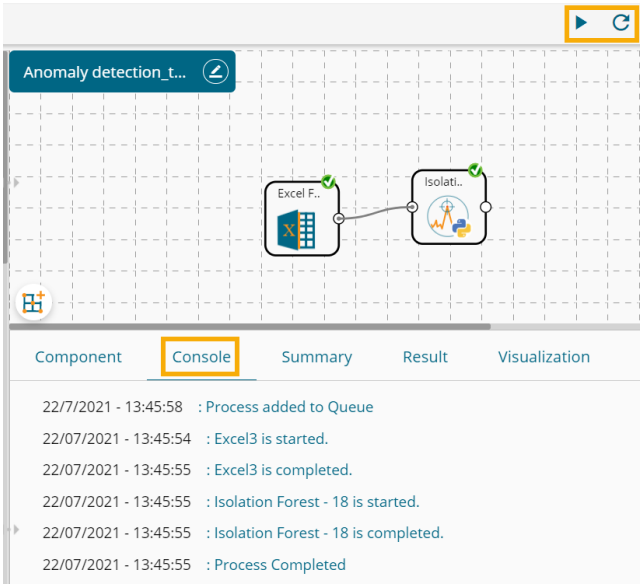
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a. Select a feature column.
   b. Set the contamination value (the default value for this field is 0.1).
   c. Set the value for the Estimators.
   d. Set value for the Maximum Samples (the default value for this field is 50).
   e. Set a Seed Value (the default value for this field is 5).
   f. Select an option to either show or remove outliers.

iv) Click the '**Apply**' option.

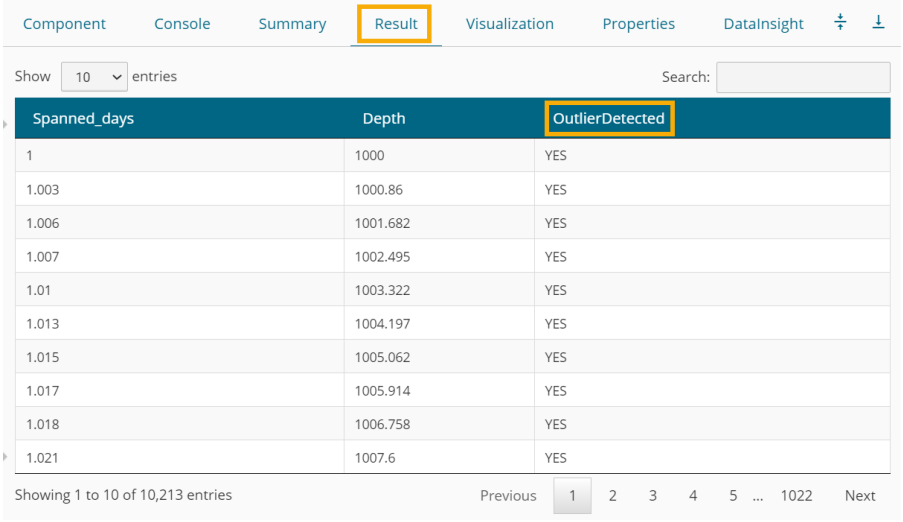


v) Run the workflow after getting the success message.

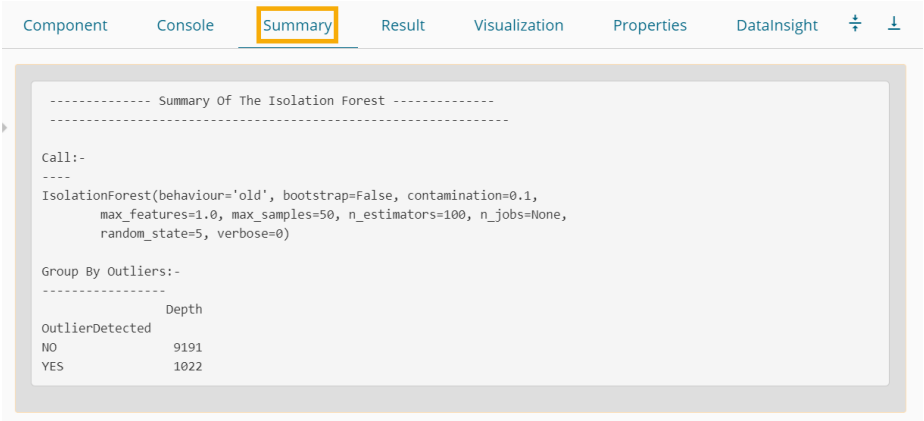
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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) Click the '**Summary**' tab to open the model summary.

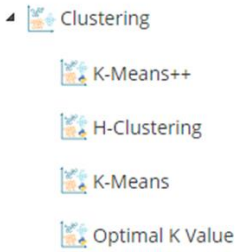


14.1.6. Clustering

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).

The Clustering algorithm in the Python Workspace contains the following algorithms:

1. K-Means++
2. H-Clustering
3. K-Means
4. Optimal K Value

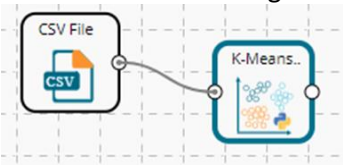


14.1.6.1. K-Means ++

The K-Means ++ algorithm ensures a smarter initialization of the centroids and improves the quality of the clustering. Apart from initialization, the rest of the algorithm is the same as the standard K-means algorithm. That is K-means++ is the standard K-means algorithm coupled with a smarter initialization of the centroids.

i) Drag the K-Means++ algorithm component to the workspace and connect it with a configured data source.

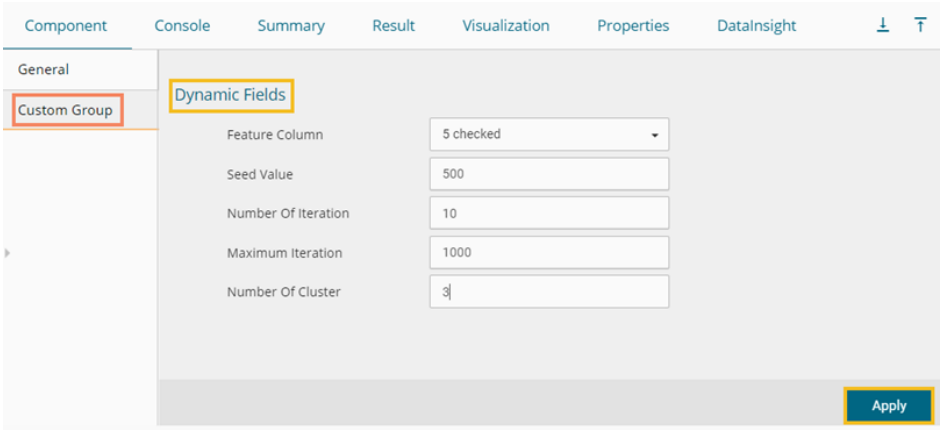
ii) Click the K-Means++ algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

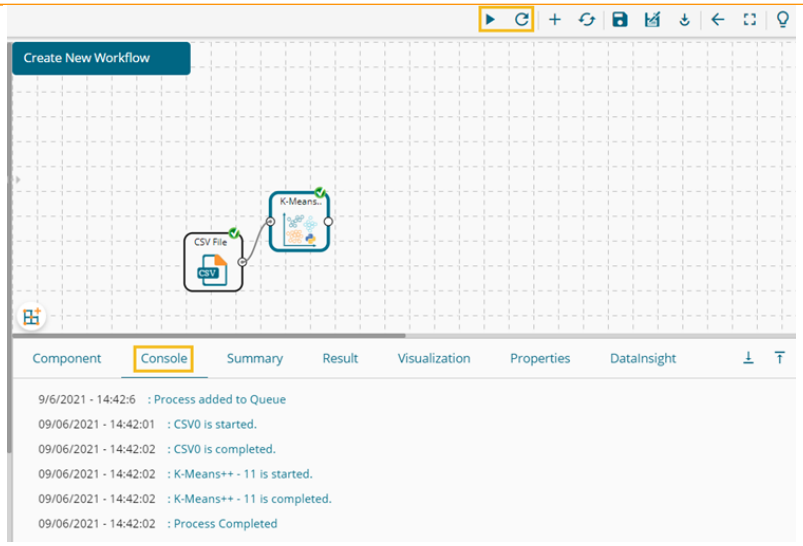
a. Feature Column: Select the feature columns from the input data set
b. Seed Value: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. Set the initial seed value (the default seed value for this field is 500).
c. Number of Iteration: Set the number of Iteration to run the data set repetitively (the default value for this field is 10).
d. Maximum Iteration: Enter the number of iterations allowed for discovering clusters. (The default value for this field is 1000).
e. Number of Clusters: Set the number of clusters to be created.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

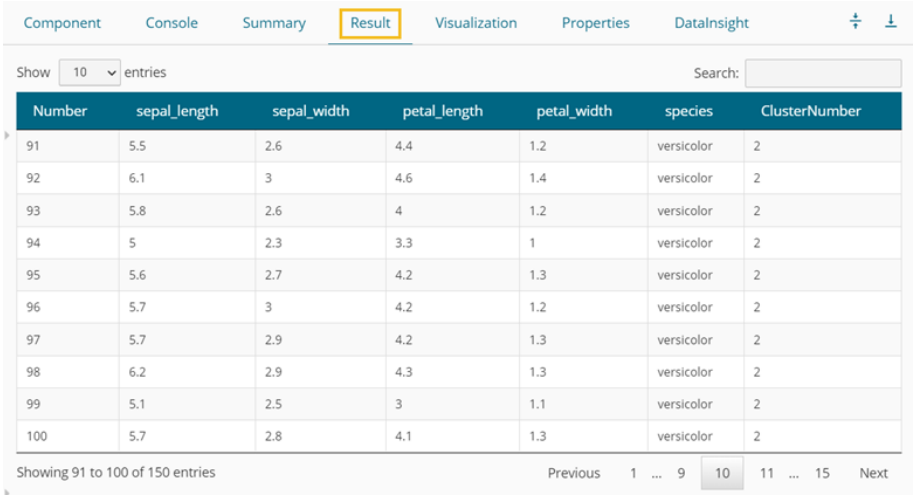
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



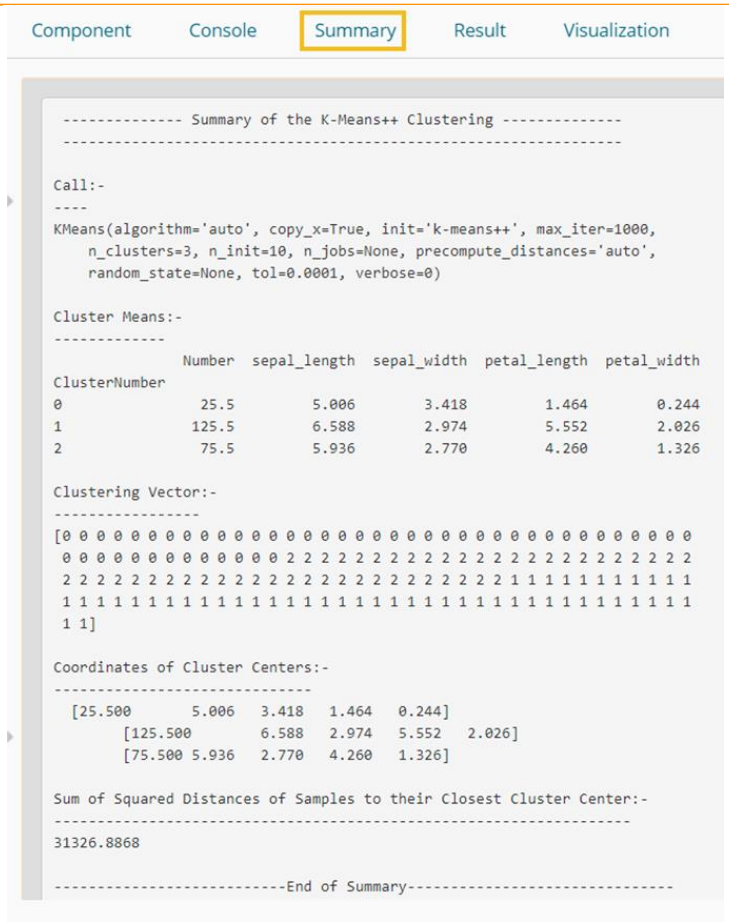
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



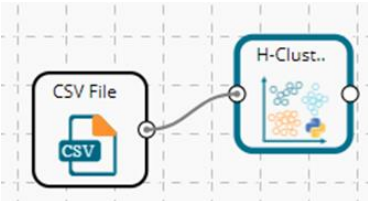
viii) Click the '**Summary**' tab to open the model summary.



14.1.6.2. H-Clustering

i) Drag the H-Clustering algorithm component to the workspace and connect it with a configured data source.

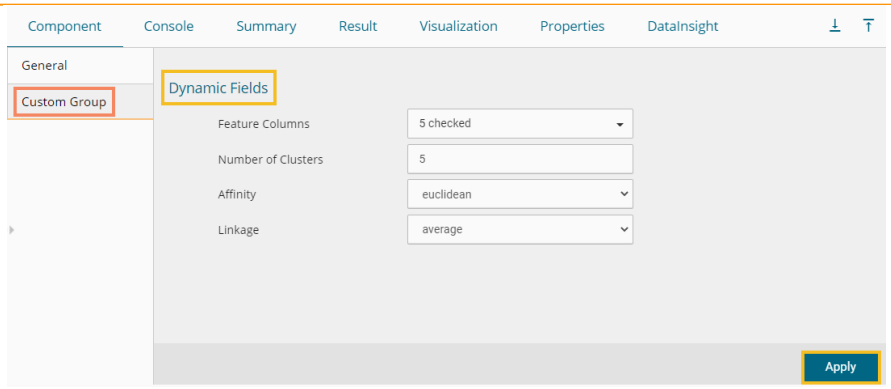
ii) Click the H-Clustering algorithm component.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

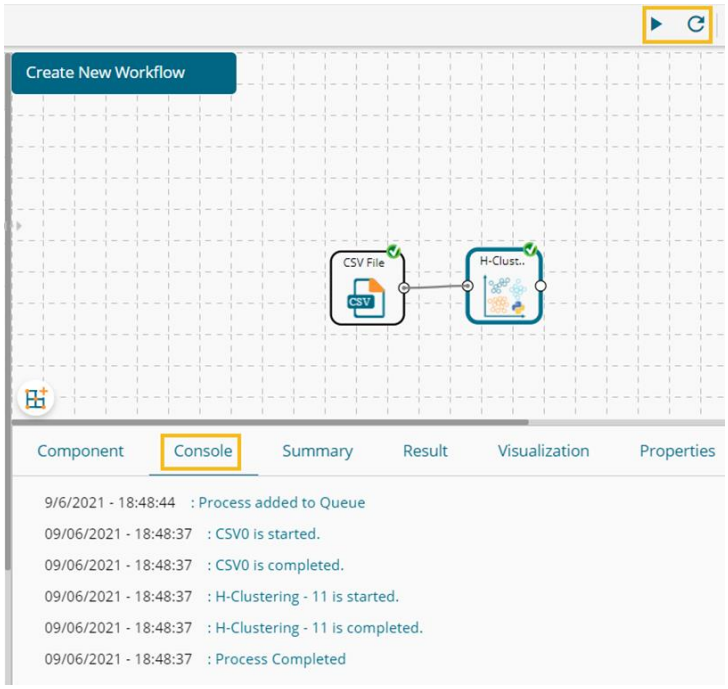
a. Feature Columns: Select the feature columns from the input dataset
b. Number of Clusters: Set the number of clusters to be created
c. Affinity: Select an affinity option out of the given choices (Euclidean, l1, l2, manhattan, cosine, precomputed)
d. Linkage: Select a linkage option out of the given choices (average, ward, complete)

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

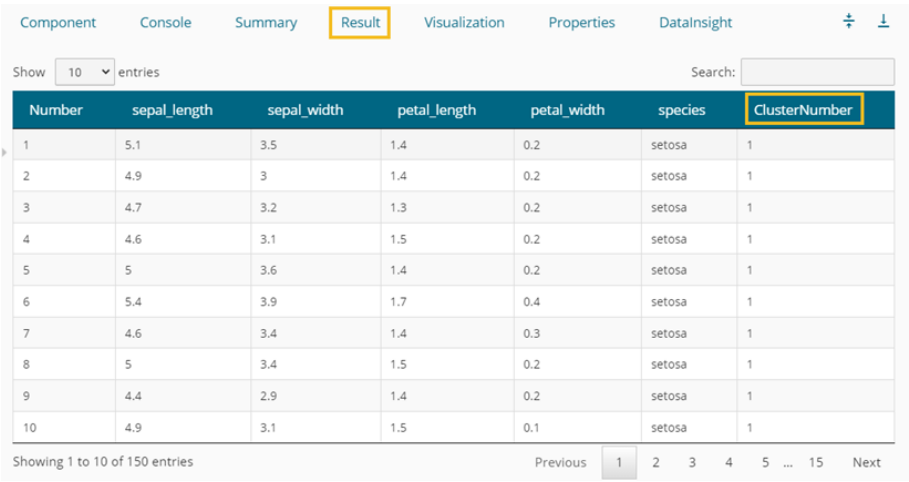
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



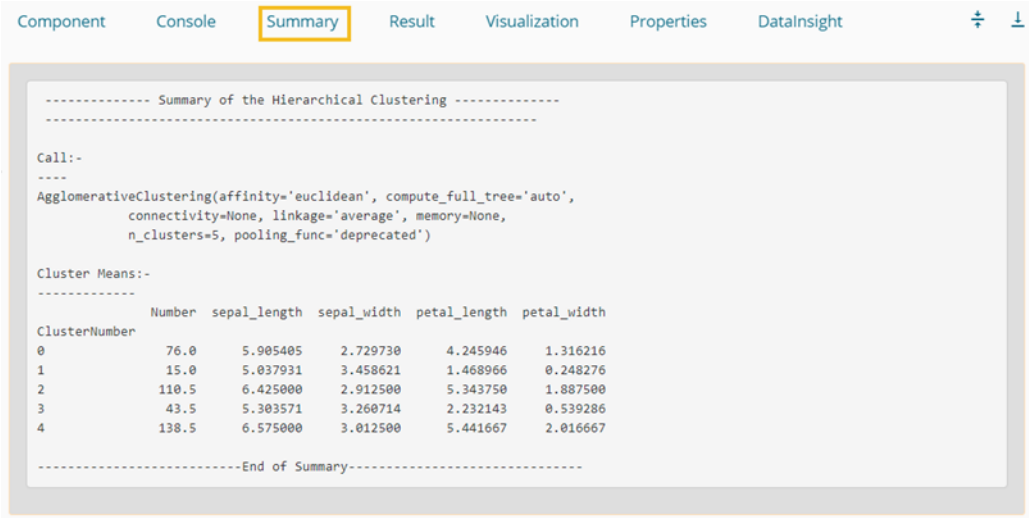
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

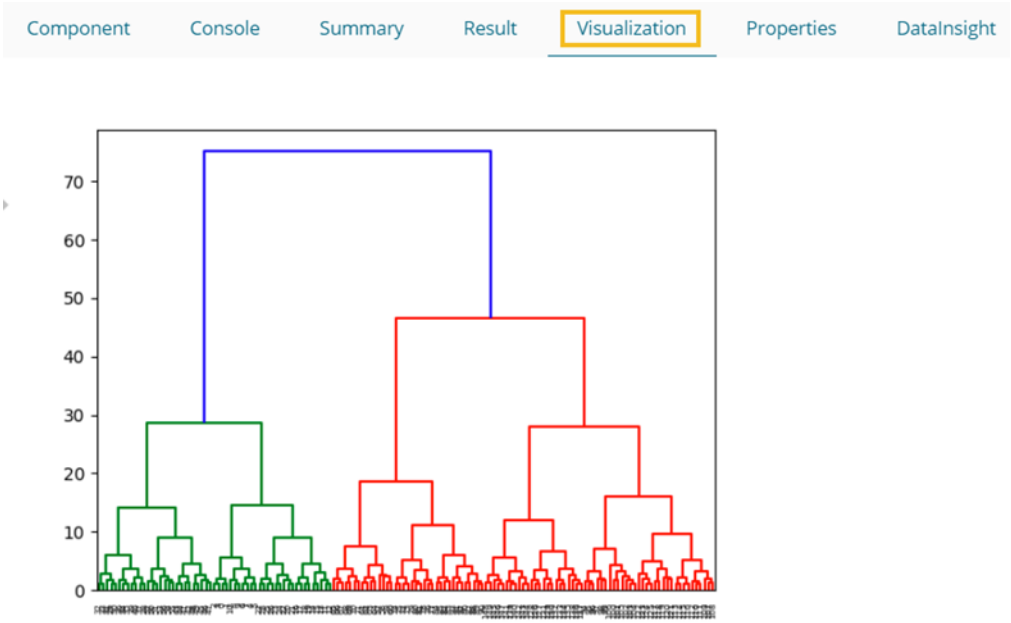
b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



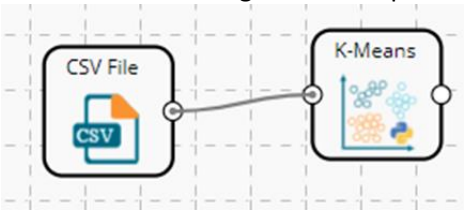
ix) Click the '**Visualization**' tab to open the result view via chart.



14.1.6.3. K-Means

x) Drag the K-Means algorithm component to the workspace and connect it with a configured data source.

xi) Click the K-Means algorithm component.

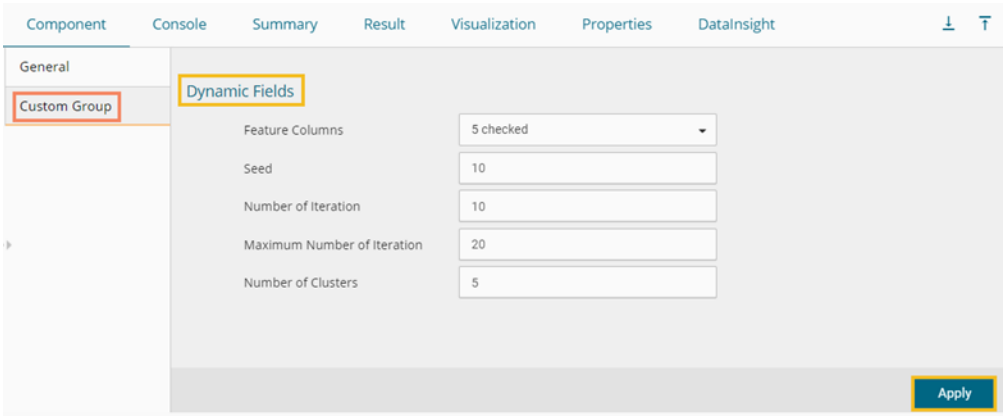


xii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. Feature Columns: Select the feature columns from the input data set

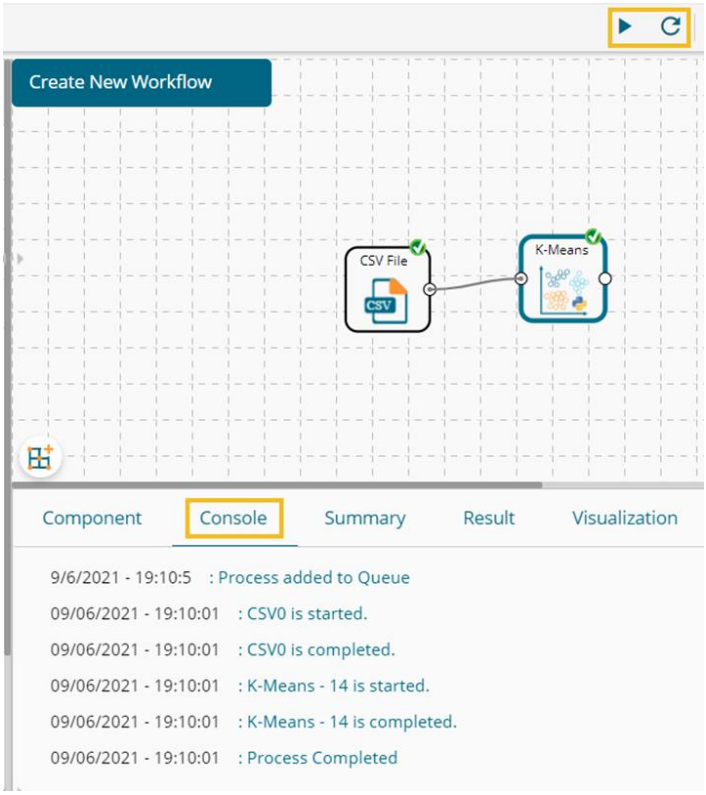
b. Seed: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. Set the initial seed value (the default seed value for this field is 10).
c. Number of Iteration: Set the number of Iteration to run the data set repetitively (the default value for this field is 10).
d. Maximum Iteration: Enter the number of iterations allowed for discovering clusters. (The default value for this field is 20).
e. Number of Clusters: Set the number of clusters to be created (The default value for this field is 5).

xiii) Click the '**Apply**' option.



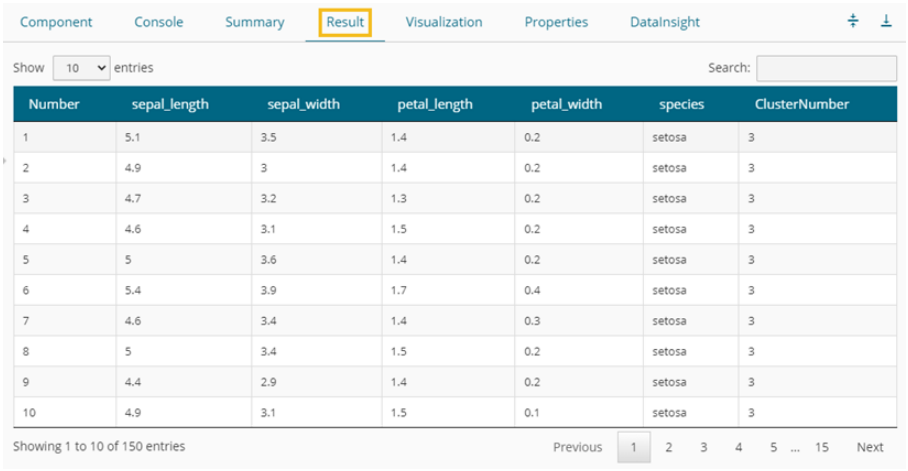
xiv) Run the workflow after getting the success message.

xv) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



xvi) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.
b. Click the '**Result**' tab.



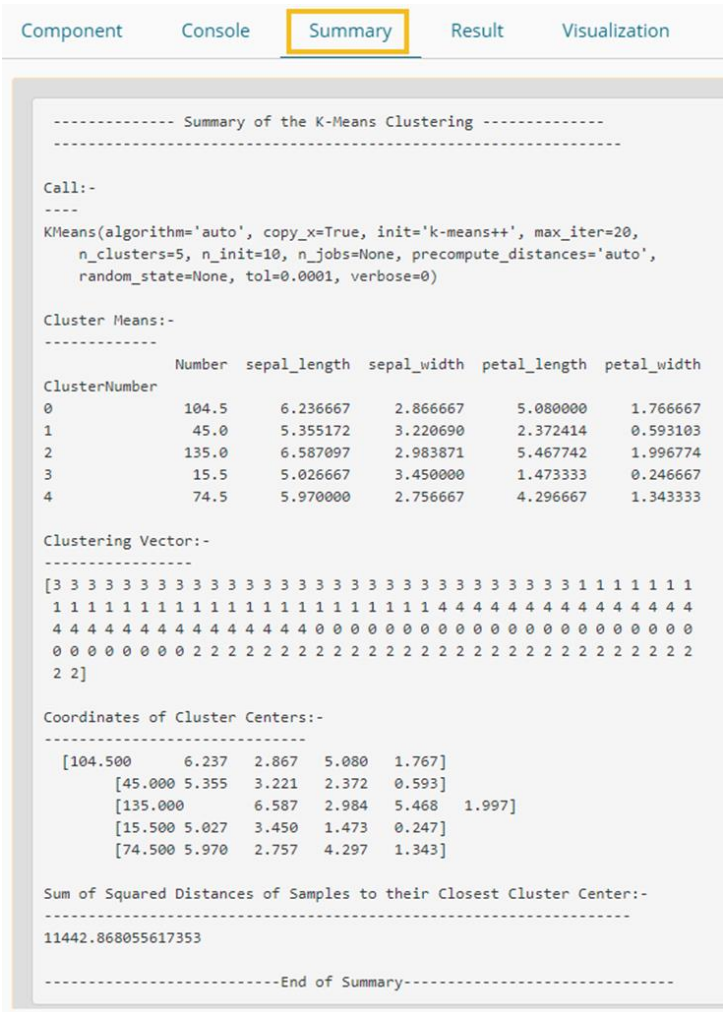
Component | Console | Summary | Result | Visualization | Properties | DataInsight

Show 10 entries Search:

Number	sepal_length	sepal_width	petal_length	petal_width	species	ClusterNumber
1	5.1	3.5	1.4	0.2	setosa	3
2	4.9	3	1.4	0.2	setosa	3
3	4.7	3.2	1.3	0.2	setosa	3
4	4.6	3.1	1.5	0.2	setosa	3
5	5	3.6	1.4	0.2	setosa	3
6	5.4	3.9	1.7	0.4	setosa	3
7	4.6	3.4	1.4	0.3	setosa	3
8	5	3.4	1.5	0.2	setosa	3
9	4.4	2.9	1.4	0.2	setosa	3
10	4.9	3.1	1.5	0.1	setosa	3

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

xvii) Click the '**Summary**' tab to open the model summary.



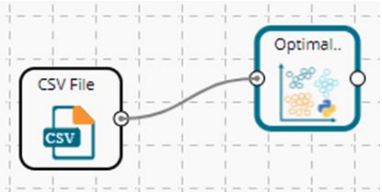
Component | Console | Summary | Result | Visualization

```
 -------------- Summary of the K-Means Clustering --------------
 ---------------------------------------------------------------

Call:-
----
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=20,
    n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)

Cluster Means:-
-------------
                Number  sepal_length  sepal_width  petal_length  petal_width
ClusterNumber
0                104.5      6.236667     2.866667      5.080000     1.766667
1                 45.0      5.355172     3.220690      2.372414     0.593103
2                135.0      6.587097     2.983871      5.467742     1.996774
3                 15.5      5.026667     3.450000      1.473333     0.246667
4                 74.5      5.970000     2.756667      4.296667     1.343333

Clustering Vector:-
-----------------
[3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]

Coordinates of Cluster Centers:-
------------------------------
  [104.500      6.237   2.867   5.080   1.767]
       [45.000 5.355   3.221   2.372   0.593]
       [135.000        6.587   2.984   5.468   1.997]
       [15.500 5.027   3.450   1.473   0.247]
       [74.500 5.970   2.757   4.297   1.343]

Sum of Squared Distances of Samples to their Closest Cluster Center:-
--------------------------------------------------------------------
11442.868055617353

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

14.1.6.4. Optimal K Value

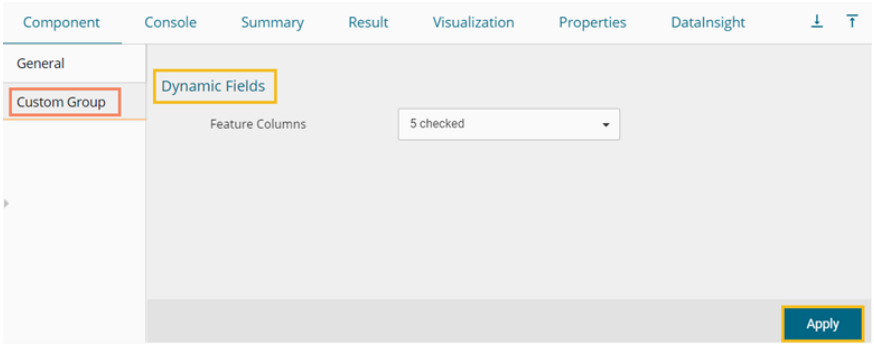
i) Drag the Optimal K Value algorithm component to the workspace and connect it with a configured data source.

ii) Click the Optimal K Value algorithm component.



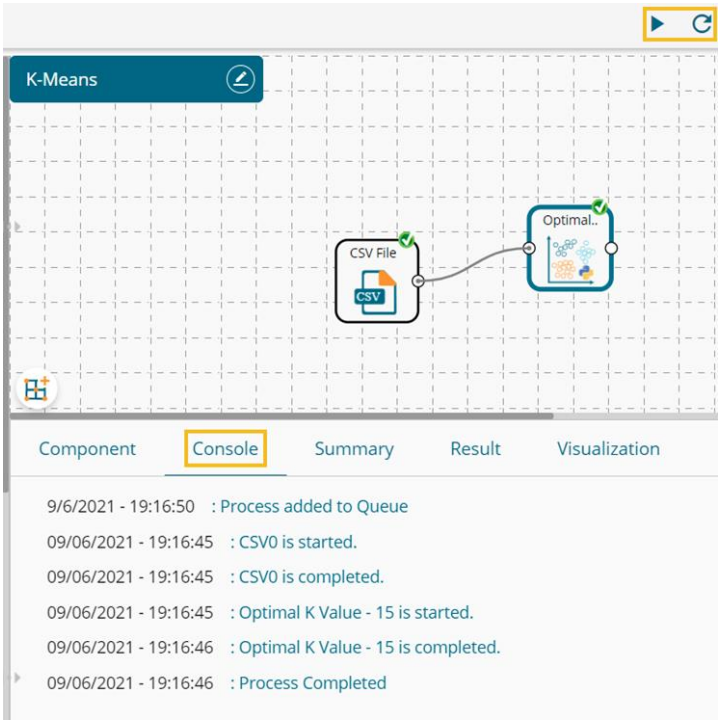
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a. Feature Columns: Select the feature columns from the input data set

iv) Click the '**Apply**' option.



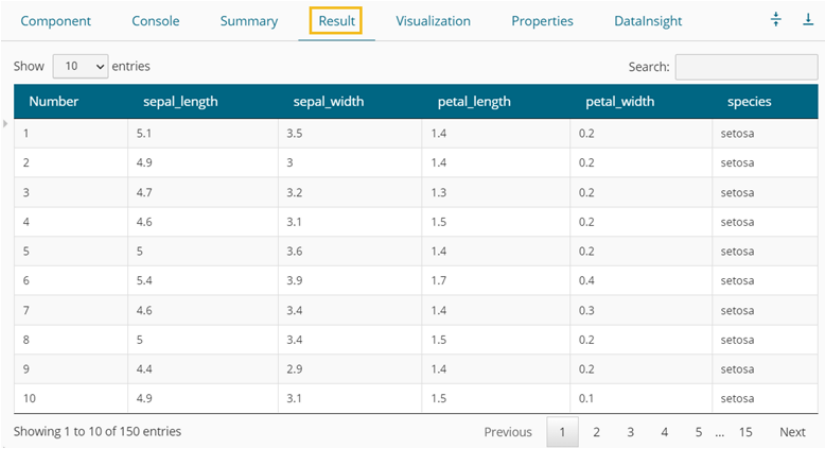
v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



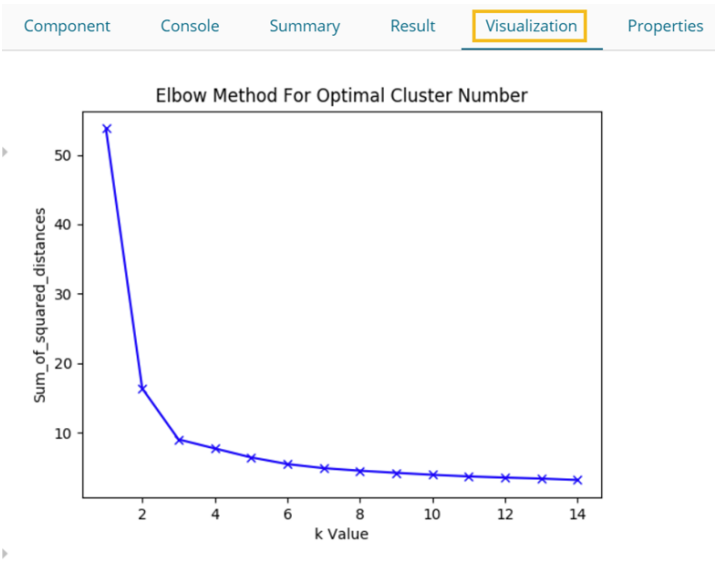
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



Number	sepal_length	sepal_width	petal_length	petal_width	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

viii) Click the '**Visualization**' tab to open the result data in the chart form.



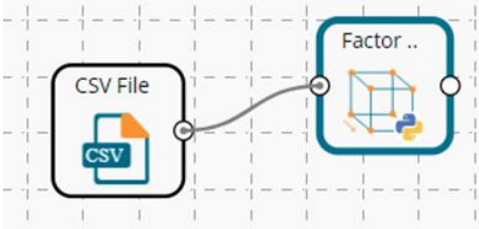
14.1.7. Dimensionality Reduction

The Dimensionality Reduction algorithm has the following sub-types in the Python Workspace as given below:

14.1.7.1. Factor Analysis

i) Drag the Factor Analysis component to the canvas and connect it to a configured data source to get the data.

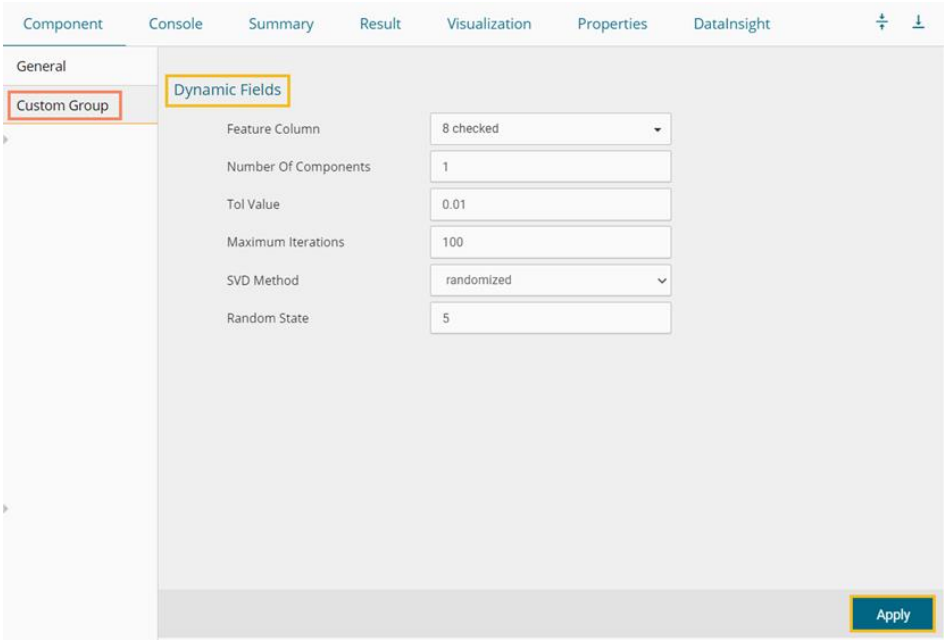
ii) Click the Factor Analysis algorithm component to get the configuration fields.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

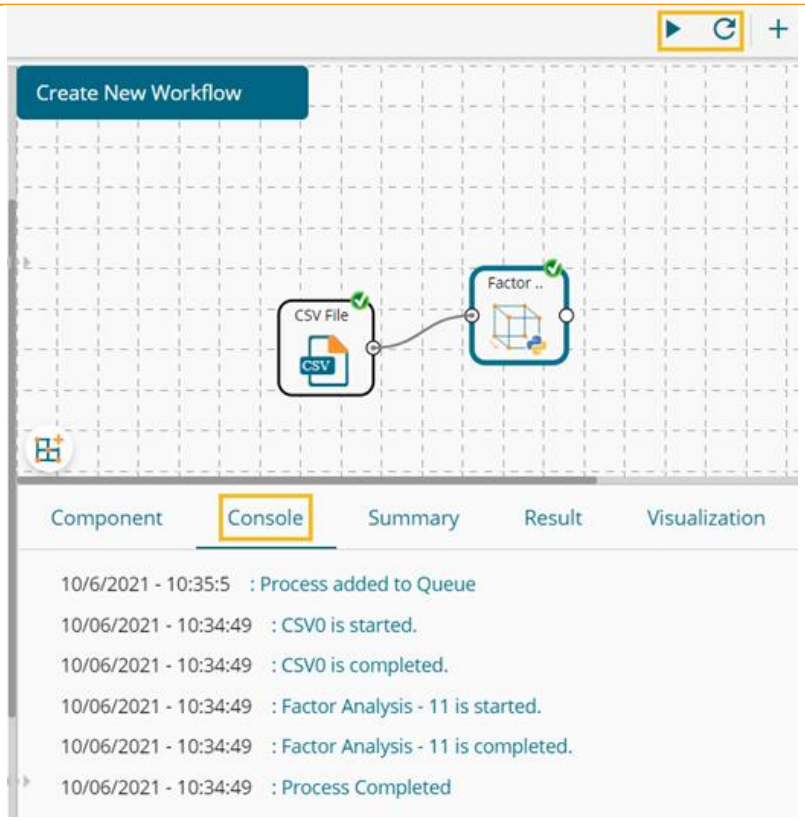
a. Feature Columns: Select required columns from the input data source against which the analysis gets applied.
b. Number of Components: Provide the number of components to be taken (the default number for this field is 1).
c. Tol Value: Set the stopping tolerance value for log-likelihood increase. Datatype by default is float (the default value for this field is 1e-2 or 0.01)
d. Maximum Iterations: Set value for the maximum iterations allowed to the selected data source (the default value for this field is 100).
e. SVD Method: Select an option out of the given choices - auto/full/randomized/arpack
f. Random State: Provide random state value (the default value for this field is 5)

iv) Click the '**Apply**' option.

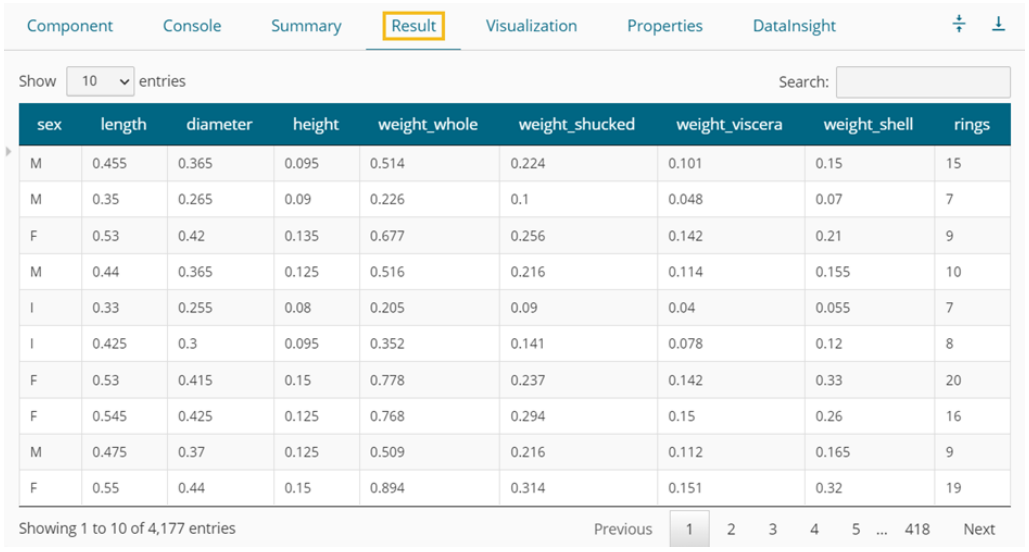


v) Run the Workflow after getting the success message.

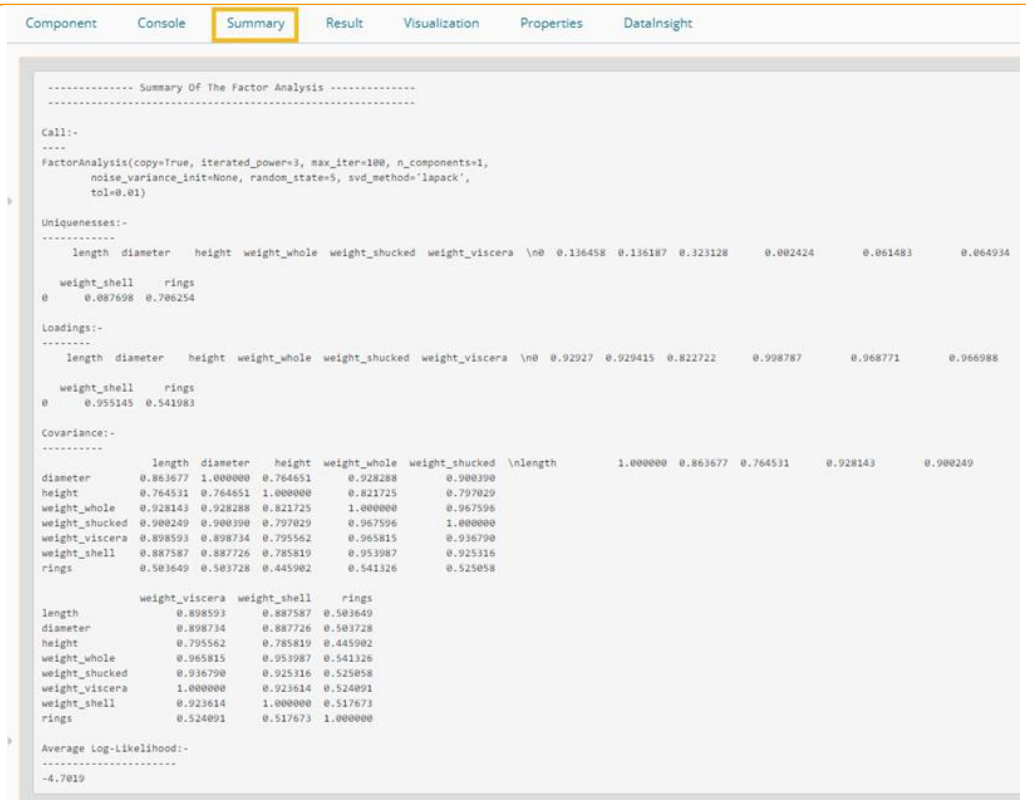
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:
viii) Click the dragged algorithm component onto the workspace.
ix) Click the '**Result**' tab.



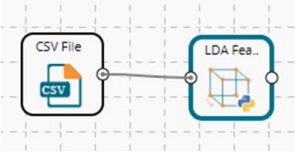
x) Click the '**Summary**' tab to open the model summary.



14.1.7.2. LDA Feature Selection

i) Drag the LDA Feature Selection component to the canvas and connect it to a configured data source to get the data.

ii) Click the LDA Feature Selection algorithm component to get the configuration fields.

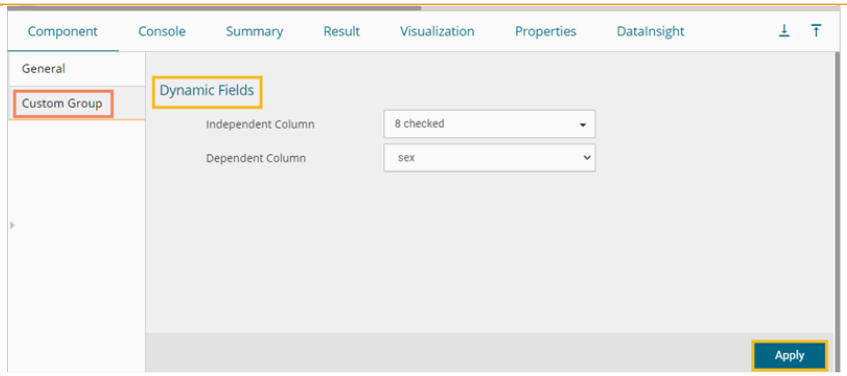


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

a. Independent Columns: Select the required input columns against which the analysis gets applied to the target column.

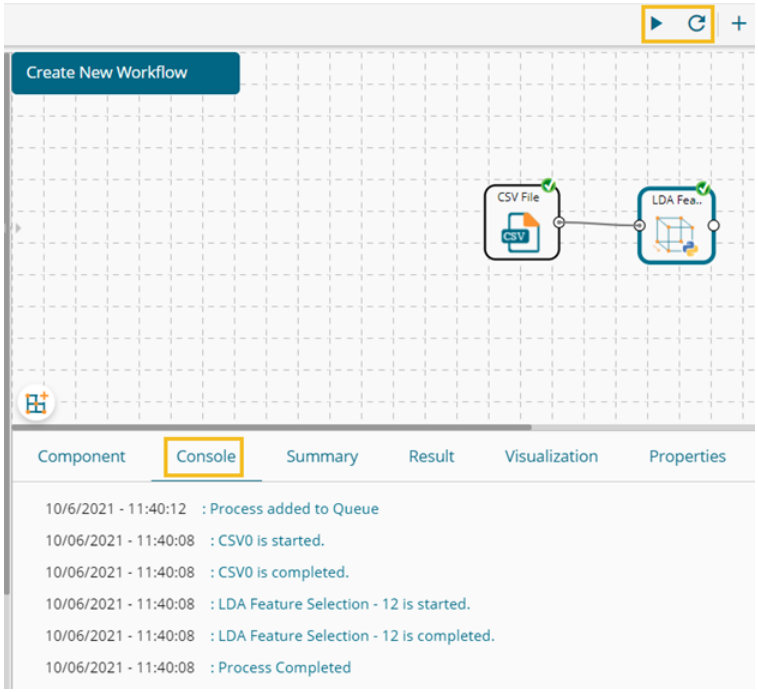
b. Dependent Column: Select the target column on which the regression analysis gets applied.

iv) Click the '**Apply**' option.



v) Run the Workflow after getting the success message.

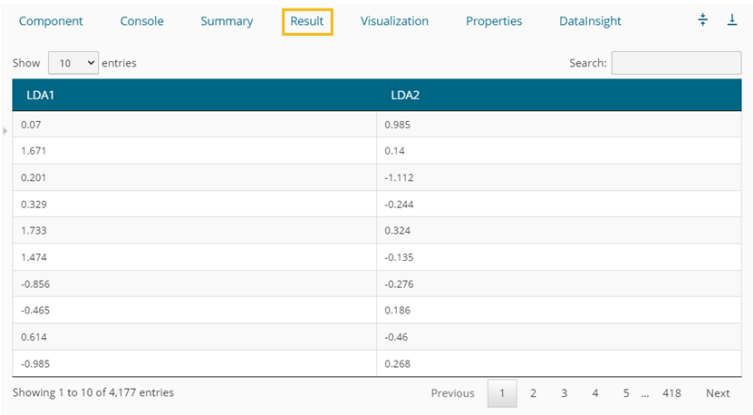
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



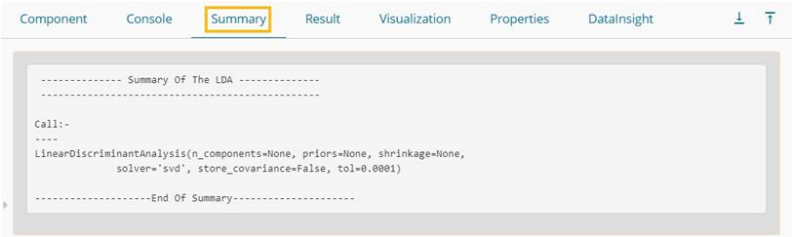
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



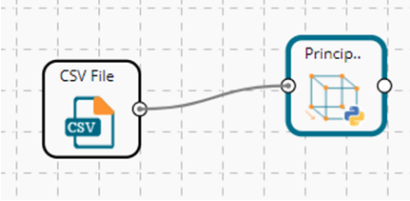
viii) Click the '**Summary**' tab to open the model summary.



14.1.7.3. Principle Component Analysis

i) Drag the Principle Component Analysis component to the canvas and connect it to a configured data source to get the data.

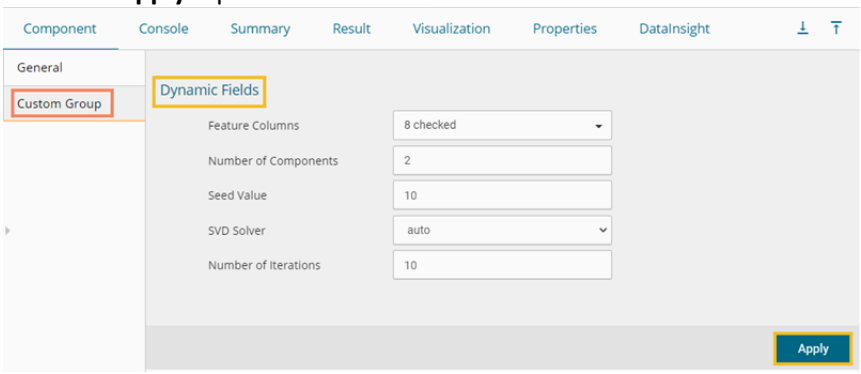
ii) Click the Principle Component Analysis algorithm component to get the configuration fields.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

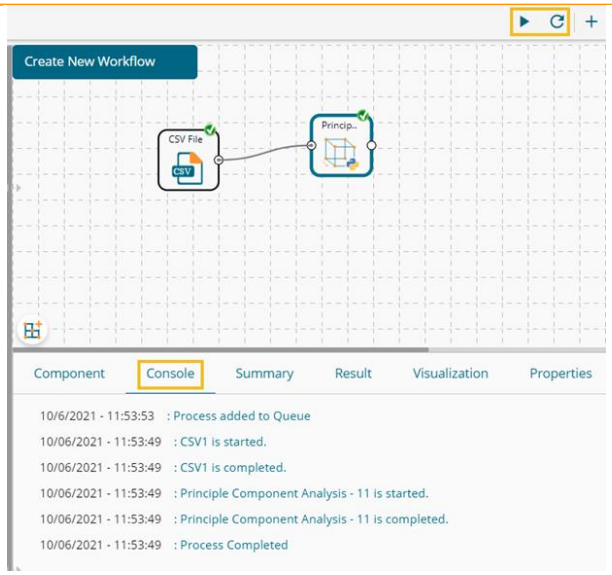
a. Select Feature Columns: Select required columns from the input data source against which the analysis gets applied.
b. Seed Value: Provide a number (or vector) to initialize the process of the pseudo-random number generator (the default value for this field is 10).
c. SVD Solver: Select an option out of the given choices- auto/full/randomized/arpack.
d. Number of Iteration: Set a number to indicate the iterations for the model (the default number for this field is 10).

iv) Click the '**Apply**' option.



v) Run the Workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Click the '**Summary**' tab to open the model summary.

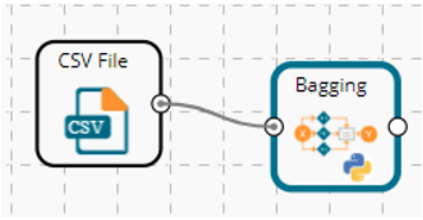
Note: There is no processed input data to display under the '**Result**' tab.

14.1.8. Ensemble Models

The Ensemble Models Algorithm contains the following sub-types under the Python Workspace:

14.1.8.1. Bagging

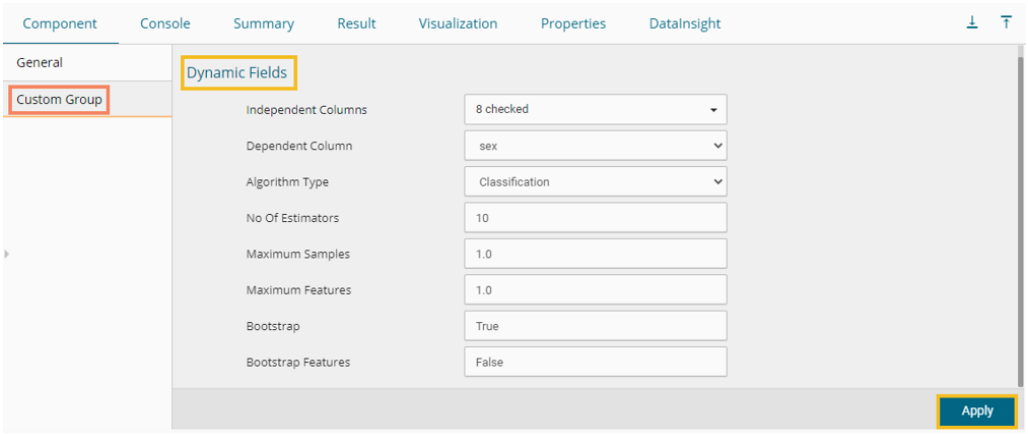
i) Drag the Bagging algorithm component to the workspace and connect it with a configured data source.
ii) Click the algorithm component to get the configuration fields.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
a. Independent Columns: Select the required input columns against which the analysis gets applied to the target column.

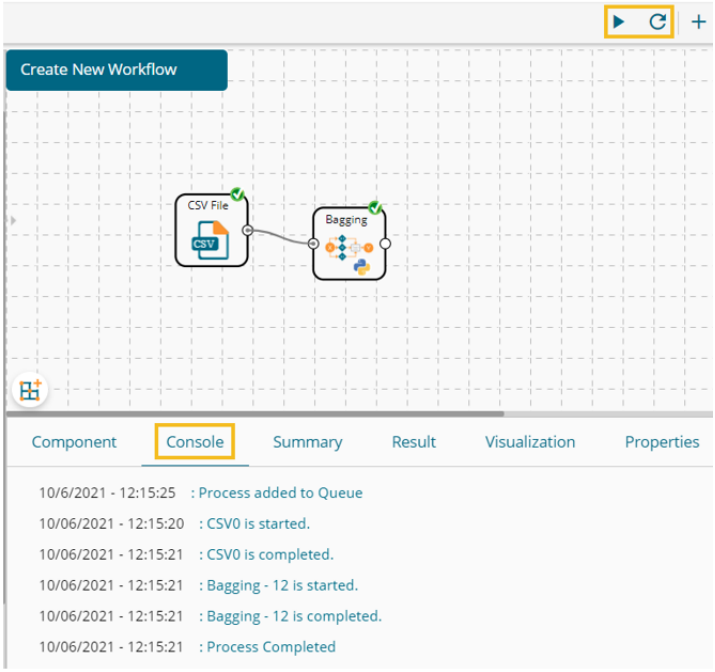
b. Dependent Column: Select the target column on which the analysis gets applied.
c. Algorithm Type: Select an algorithm type out of Classification or Regression.
d. No. of Estimators: Set a number indicating the number of estimators. The default value for this field is
e. Maximum Samples: Provide a number indicating the maximum samples. The default value for this field is
f. Maximum Features: Provide a number indicating the Maximum Features. The
g. Bootstrap: Configure with the 'True' option to select Bootstrap as a modeling method. Configure with the 'False' option if do not wish to select it. The default value for this field is '**True**'.
h. Bootstrap Features: Mention 'True' or 'False' to configure this field. the default option for this field is 'False'.

iv) Click the '**Apply**' option.

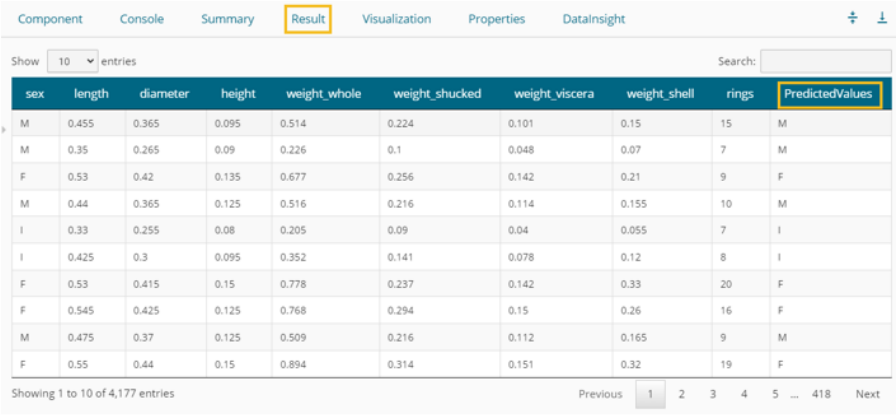


v) Run the workflow after getting the success message.

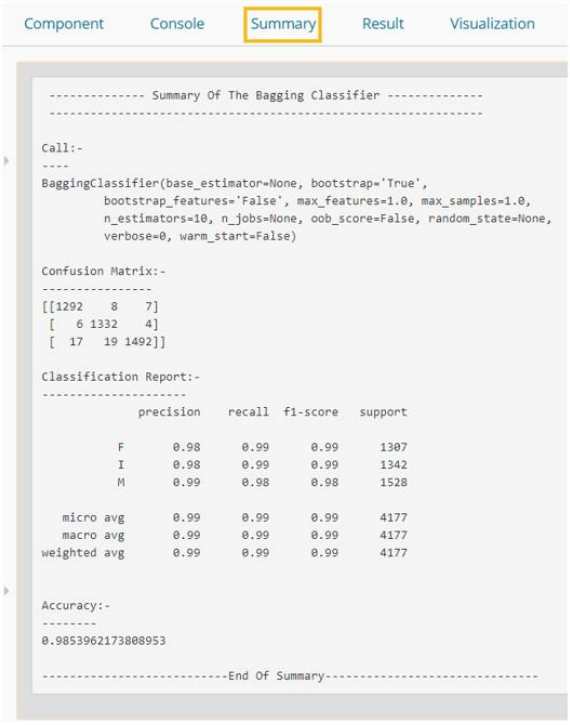
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.



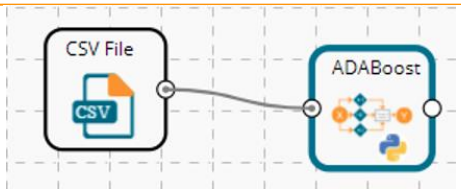
viii) Click the '**Summary**' tab to open the model summary.



14.1.8.2. ADABoost

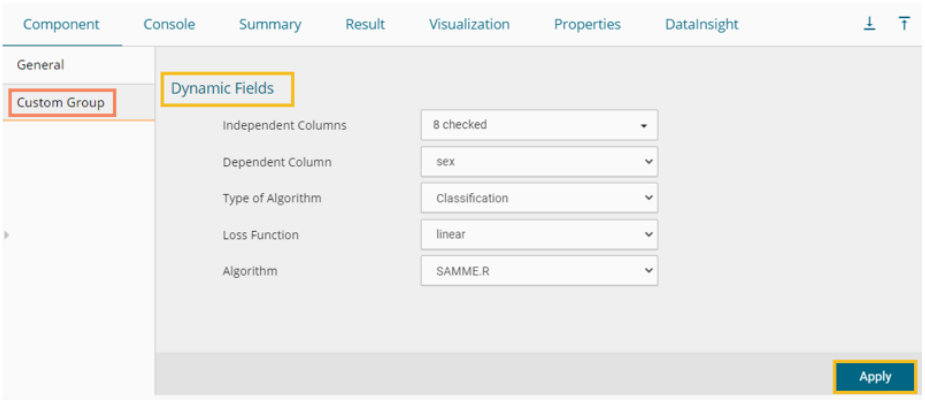
14.1.8.2.1. Algorithm Type Classification

i) Drag the ADABoost algorithm component to the workspace and connect it with a configured data source.
ii) Click the algorithm component to get the configuration fields.
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

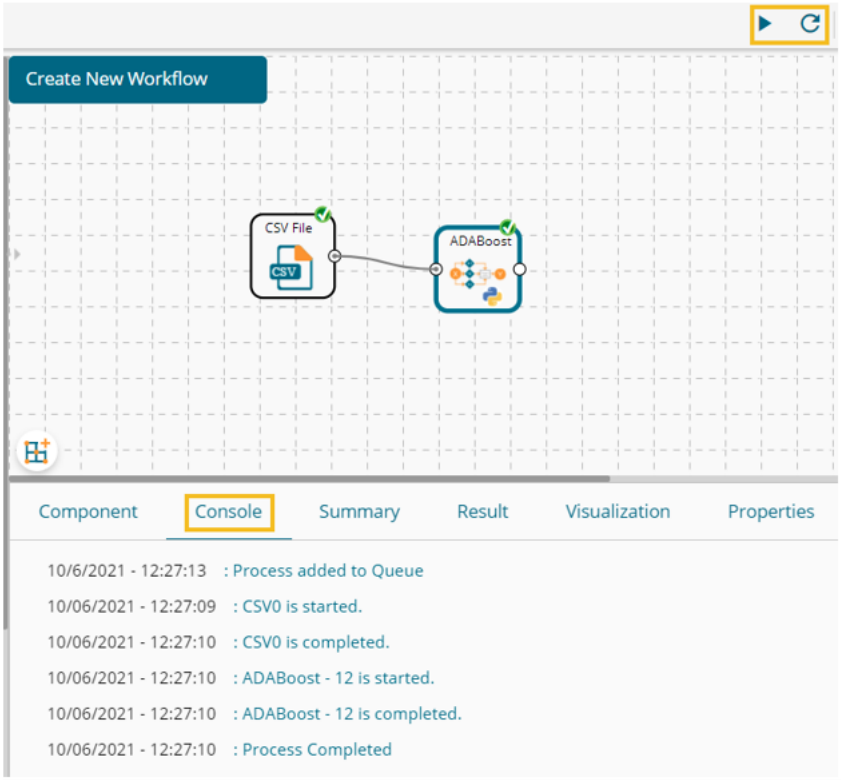


a. Independent Columns: Select the required input columns against which the analysis gets applied to the target column.
b. Dependent Column: Select the target column on which the analysis gets applied.
c. **Types of Algorithms:** The selected algorithm type is **Classification.**
d. Loss Function: Select an option out linear, square, or exponential
e. Algorithm: Select an option out of SAMME.R and SAMME (the default option for this field is SAMME.R)

iv) Click the '**Apply**' option.



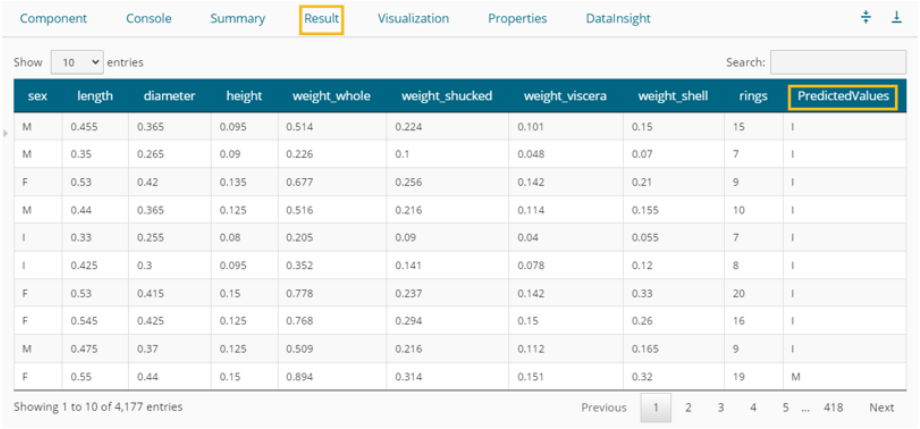
v) Run the workflow after getting the success message.
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



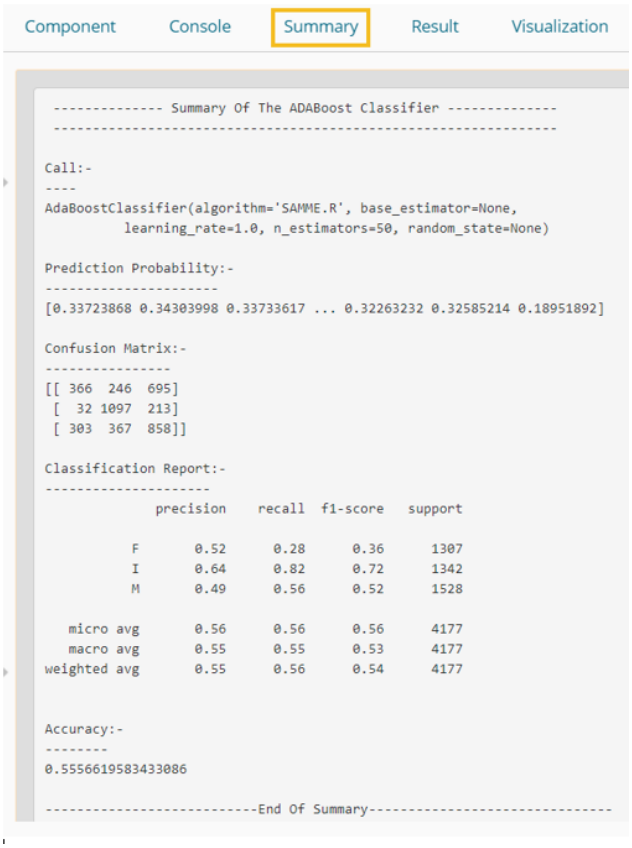
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

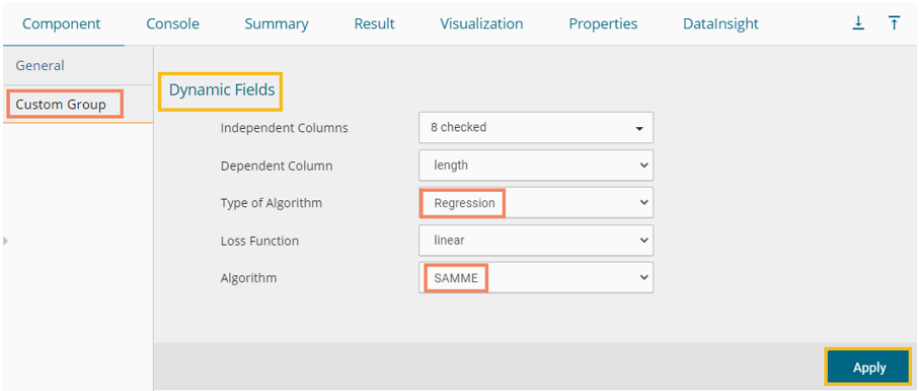


viii) Click the '**Summary**' tab to open the model summary.



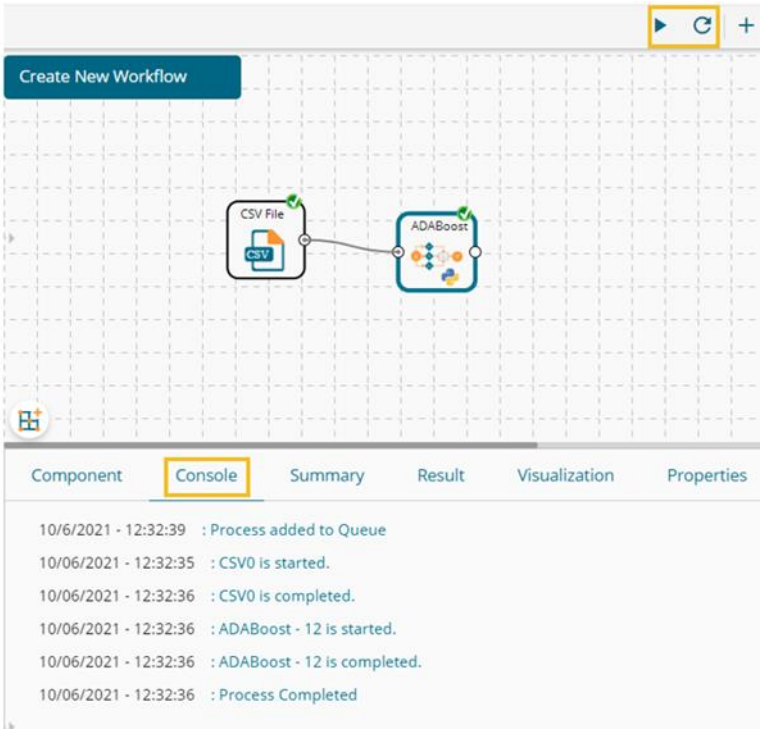
14.1.8.2.2. Algorithm Type Regression

i) Drag the ADABoost algorithm component to the workspace and connect it with a configured data source.
ii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
iii) Independent Columns: Select the required input columns against which the analysis gets applied to the target column.
iv) Dependent Column: Select the target column on which the analysis gets applied.
v) **Types of Algorithms:** The selected algorithm type is **Regression.**
vi) **Loss Function:** Select an option out linear, square, or exponential
vii) **Algorithms:** Select an option out of SAMME.R and SAMME (the default option for this field is SAMME.R)
viii) Click the '**Apply**' option.



ix) Run the workflow after getting the success message.

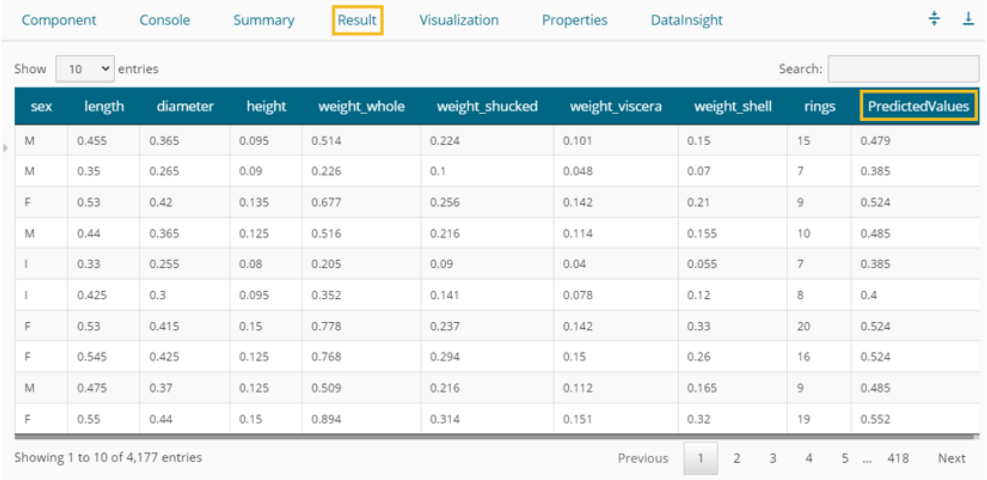
x) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



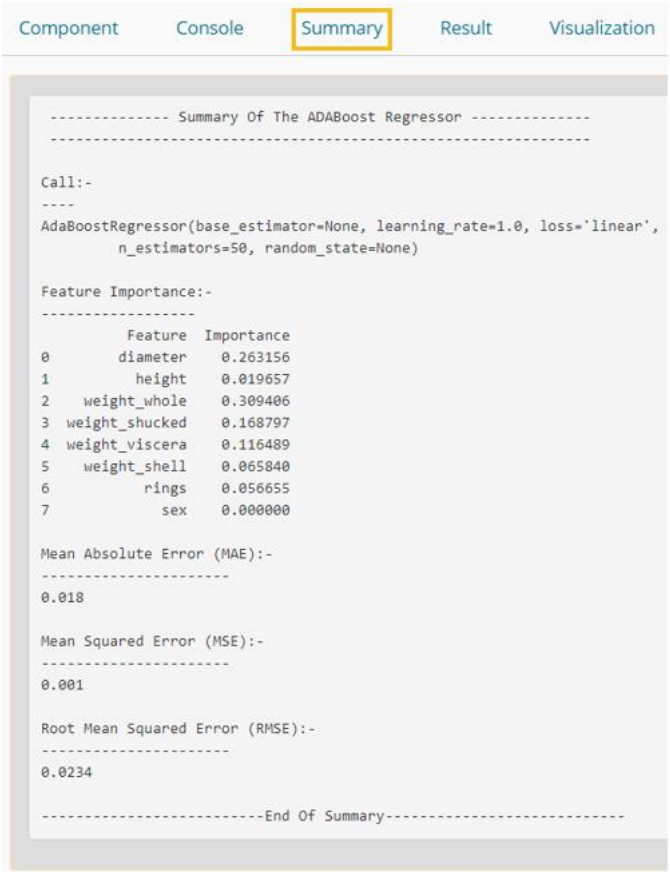
xi) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the **'Result'** tab.



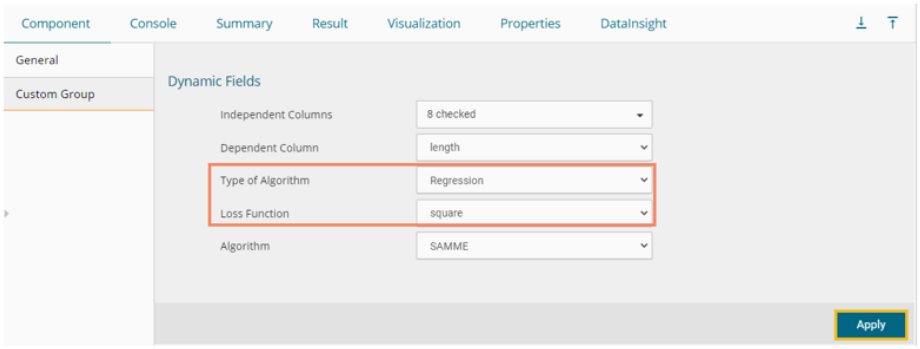
xii) Click the '**Summary**' tab to open the model summary.



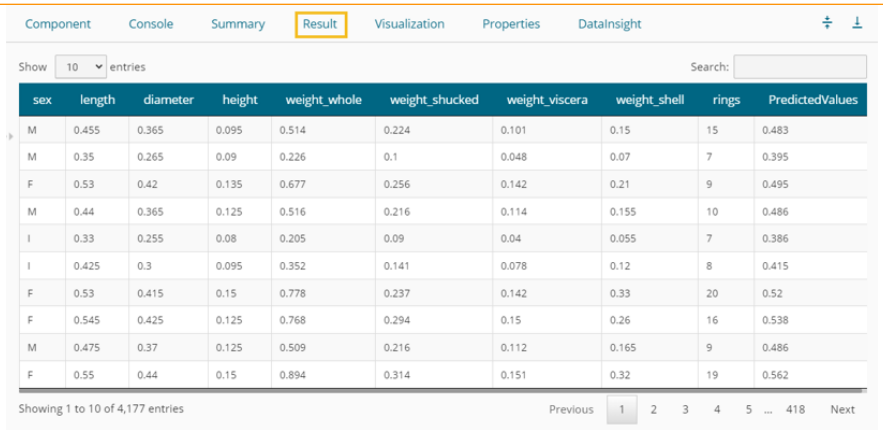
Note: The Regression algorithm together with the '**square**' and '**exponential**' Loss Function options bring slightly changed processed data under the result and summary tabs. Please look into the following image for the same:

14.1.8.2.2.1. Regression Algorithm Type with 'Square' as Loss Function

Dynamic Fields:

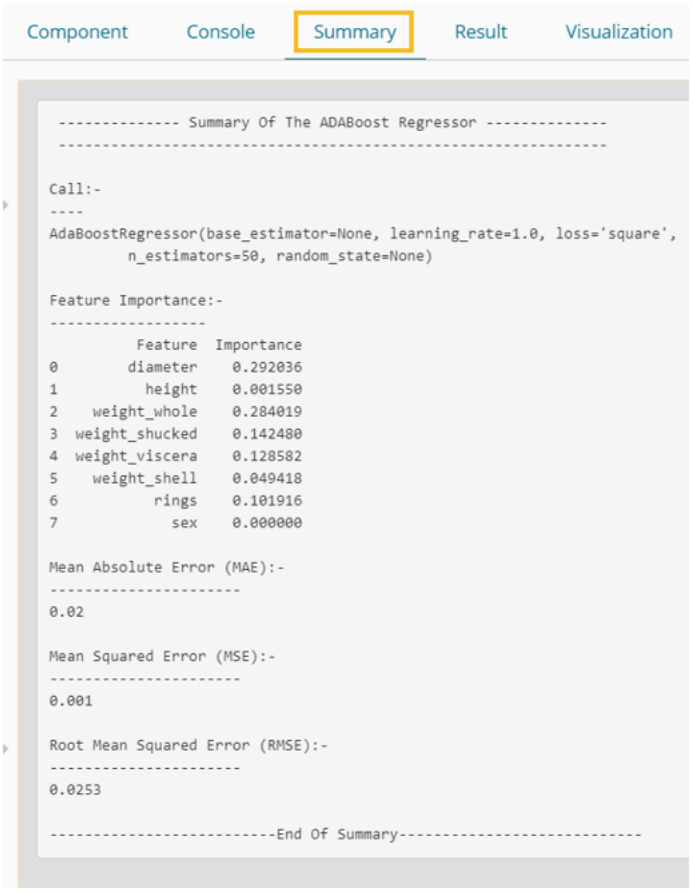


Result Tab:



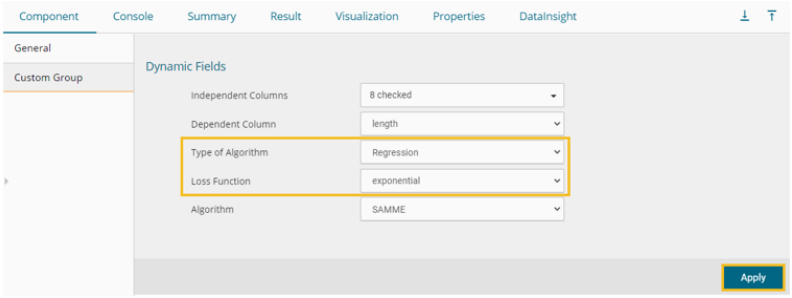
sex	length	diameter	height	weight_whole	weight_shucked	weight_viscera	weight_shell	rings	PredictedValues
M	0.455	0.365	0.095	0.514	0.224	0.101	0.15	15	0.483
M	0.35	0.265	0.09	0.226	0.1	0.048	0.07	7	0.395
F	0.53	0.42	0.135	0.677	0.256	0.142	0.21	9	0.495
M	0.44	0.365	0.125	0.516	0.216	0.114	0.155	10	0.486
I	0.33	0.255	0.08	0.205	0.09	0.04	0.055	7	0.386
I	0.425	0.3	0.095	0.352	0.141	0.078	0.12	8	0.415
F	0.53	0.415	0.15	0.778	0.237	0.142	0.33	20	0.52
F	0.545	0.425	0.125	0.768	0.294	0.15	0.26	16	0.538
M	0.475	0.37	0.125	0.509	0.216	0.112	0.165	9	0.486
F	0.55	0.44	0.15	0.894	0.314	0.151	0.32	19	0.562

Summary:

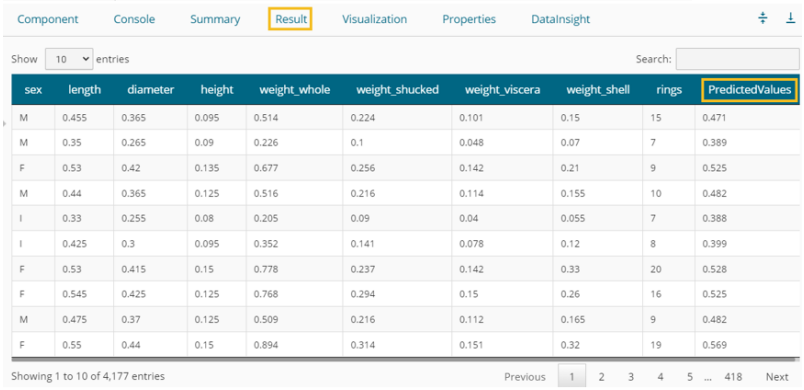


14.1.8.2.2.2. Regression Algorithm with 'exponential' as Loss Function

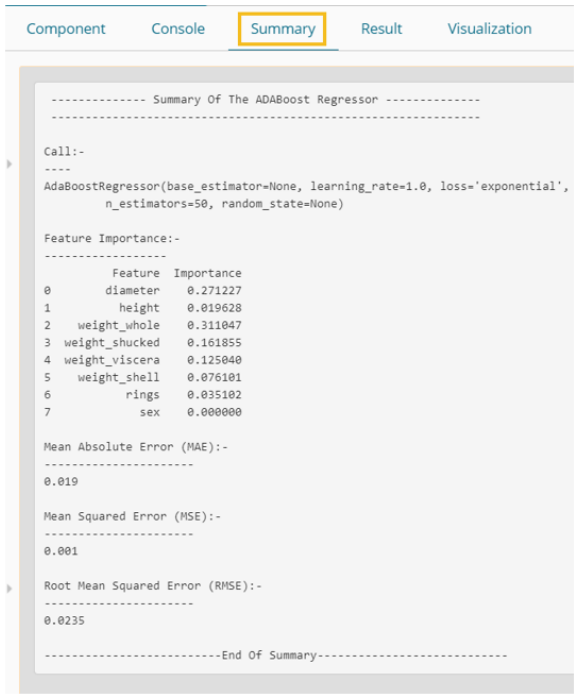
Dynamic Fields:



Result Tab:



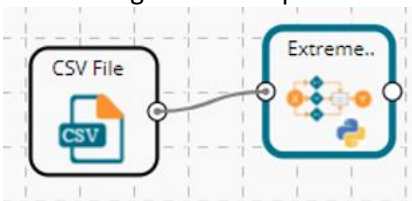
Summary Tab:



14.1.8.3. Extremely Randomized Trees

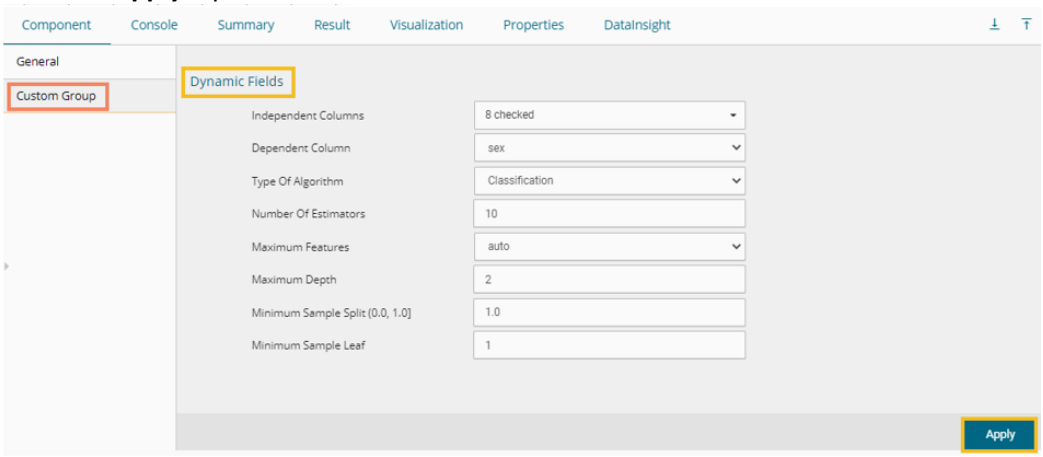
i) Drag the Extremely Randomized Trees algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



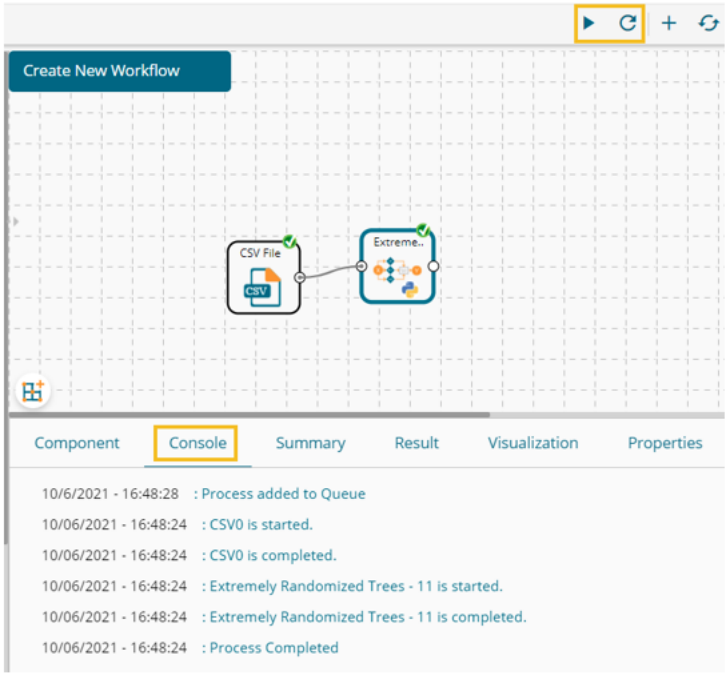
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

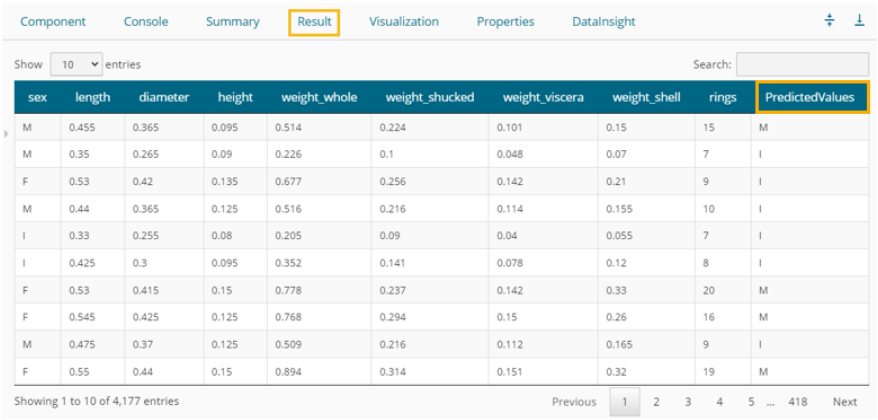
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



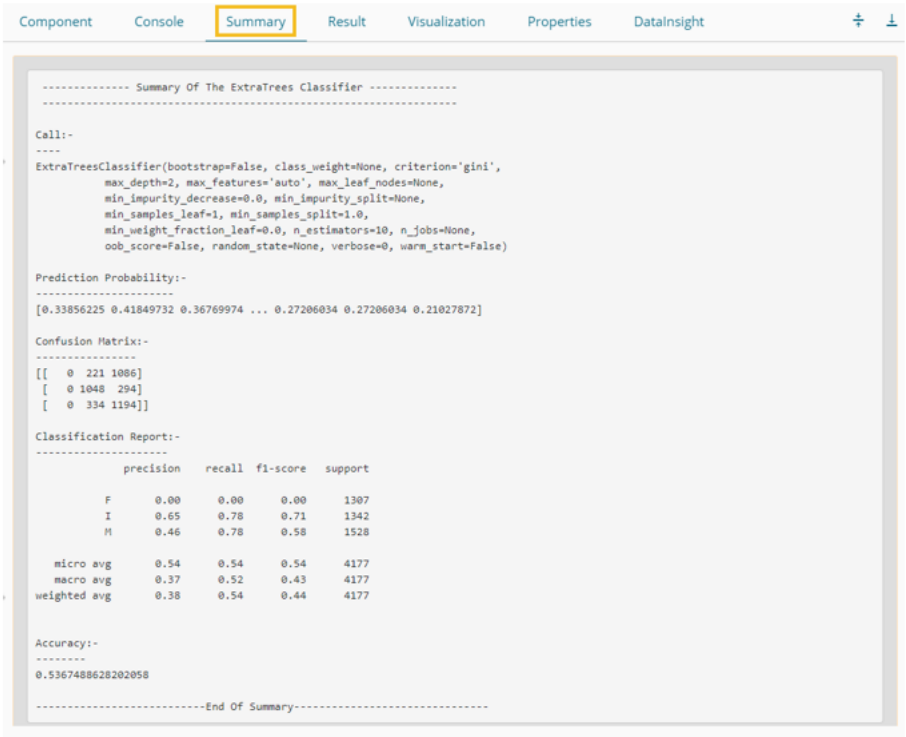
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.

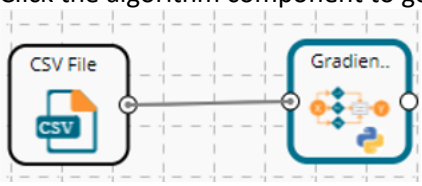


Note: The Extremely Randomized Algorithm component can be configured with the selection of the Regression as algorithm type as well.

14.1.8.4. Gradient Boosting Model

i) Drag the Gradient Boosting Model algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to get the configuration fields.

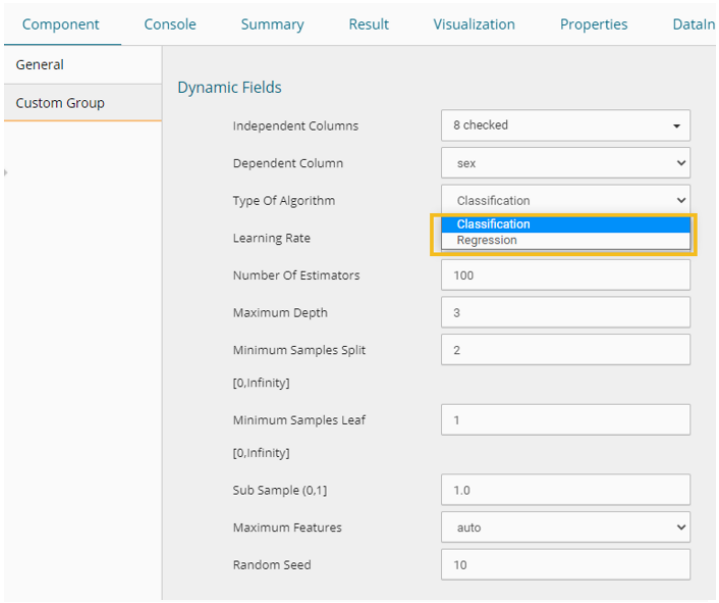


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

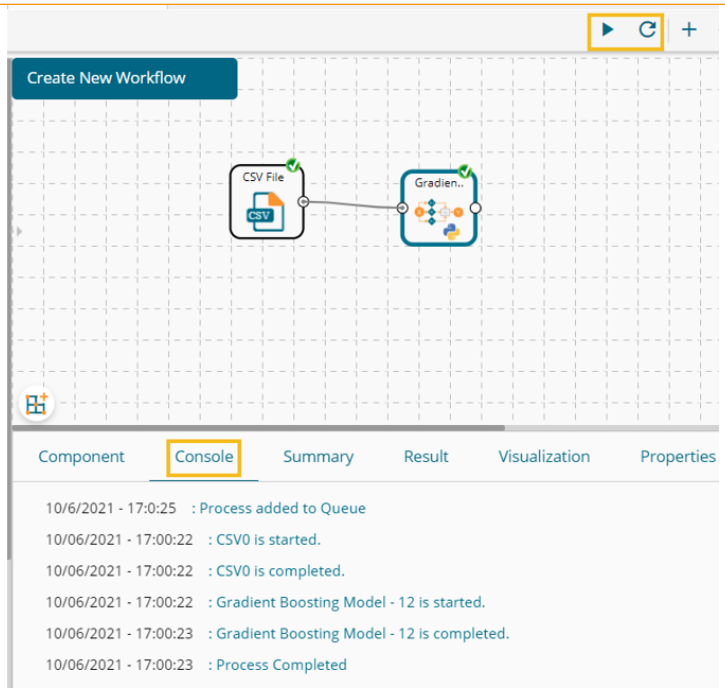
a. Independent Column: Select the required input columns against which the analysis gets applied to the target column.

b. Dependent Column: Select the target column on which the regression analysis gets applied.
c. Type of Algorithms (Select an algorithm type out of Classification or Regression)
d. Learning Rate:
e. Number of Estimators (the default value for this field is 100)
f. Maximum Depth (the default value for this field is 3)
g. Minimum Samples Split (0. infinity) - (the default value for this field is 2)
h. Minimum Sample Leaf (0. infinity) - (the default value for this field is 1)
i. Sub Sample (0,1]- (the default value for this field is 1.0)
j. Maximum Features: Select an option out of auto, sqrt, log2
k. Random Seed: Provide a number (or vector) used to initialize a pseudorandom number generator (the default value for this field is 10).

iv) Click the '**Apply**' option.



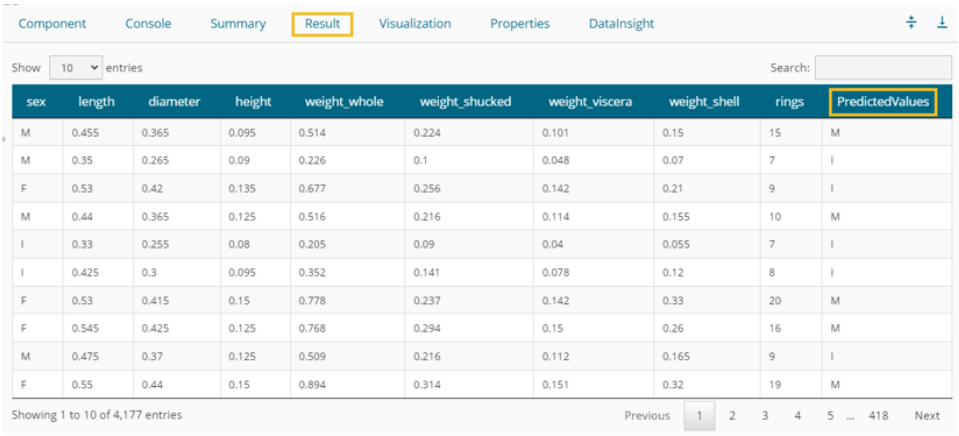
v) Run the workflow after getting the success message.
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



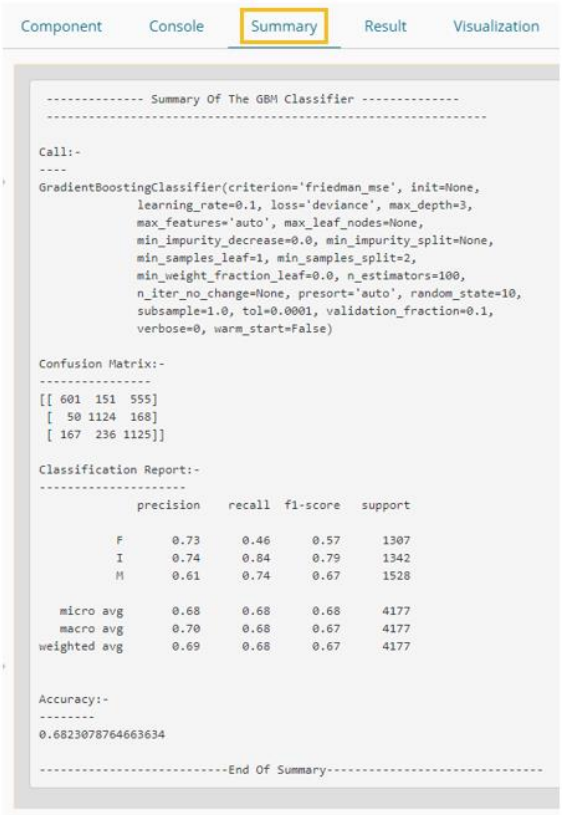
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.

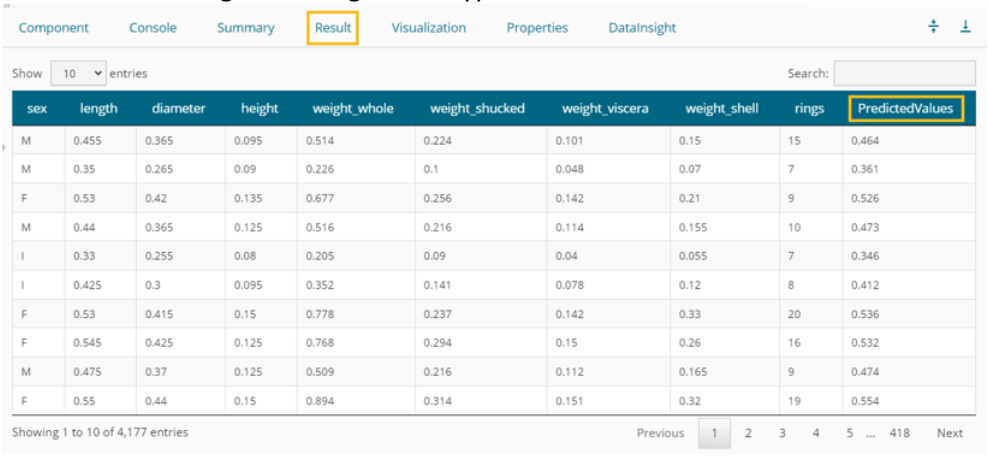


viii) Click the '**Summary**' tab to open the model summary.



Note: The users can run the model using the Regression Algorithm Type from the component configuration fields. The processed data under the Result tab will slightly get changed and the Summary will get displayed for the Regressor as given below:

a. Result Tab with Regression Algorithm Type:



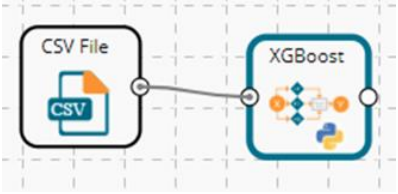
b. Summary of the model created choosing the Regression Algorithm type:

14.1.8.5. XGBoost

The XGBoost is an algorithm that uses a gradient boosting framework designed for speed and performance.

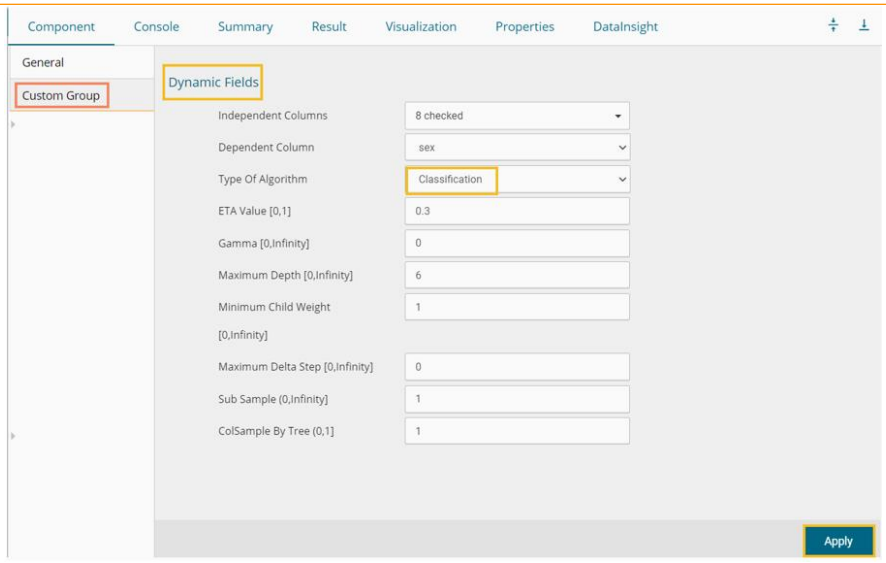
i) Drag the XGBoosting algorithm component to the workspace and connect it with a configured data source.

ii) Click on the algorithm component to open the configuration fields.



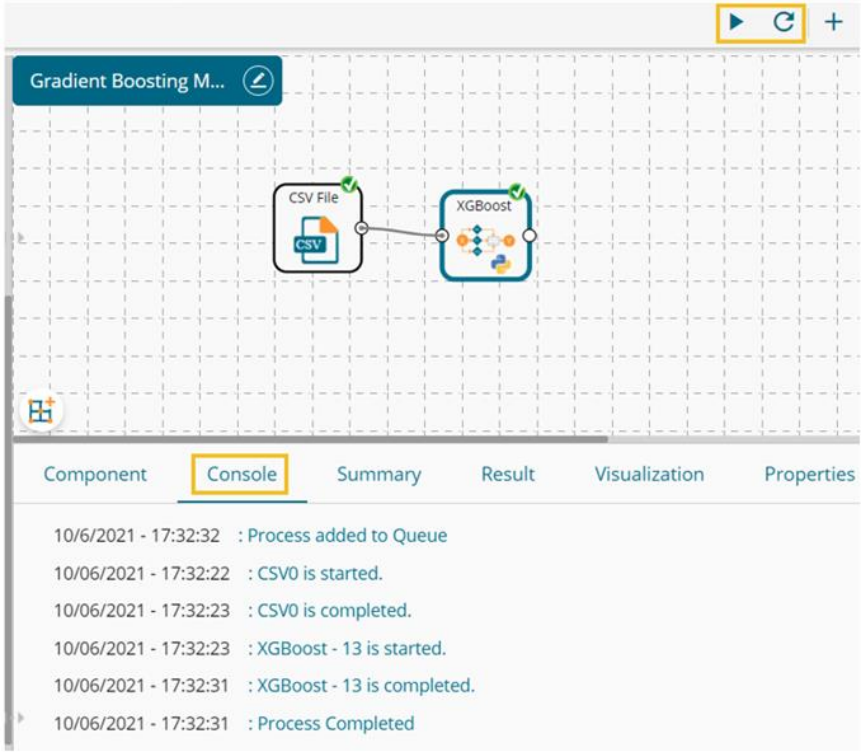
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

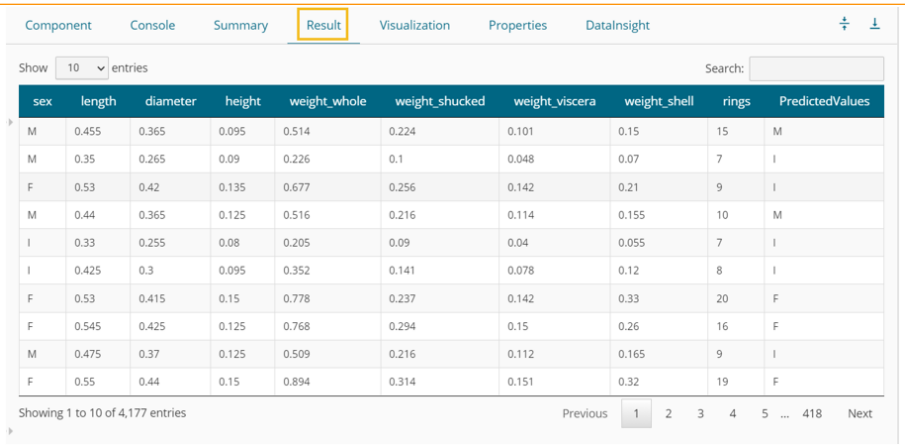
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



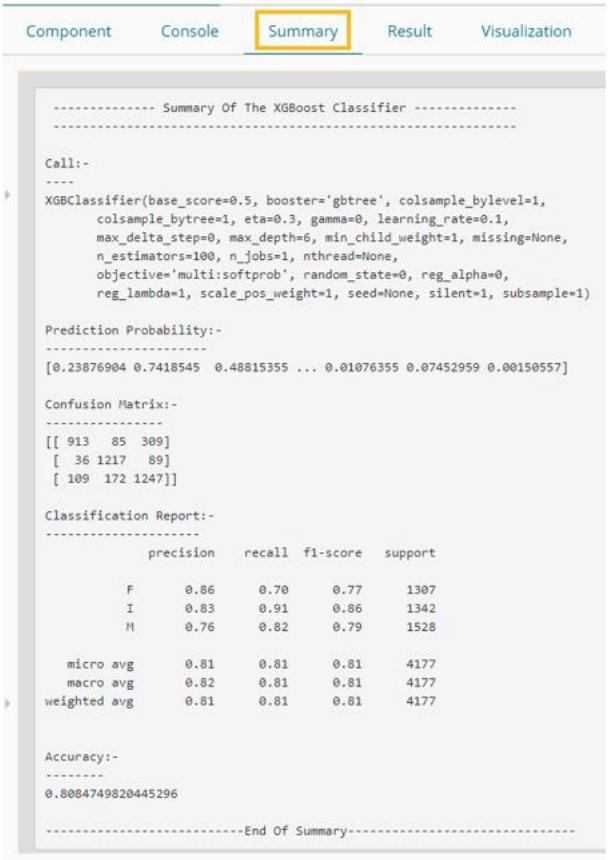
vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

b. Click the '**Result**' tab.



viii) Click the '**Summary**' tab to open the model summary.



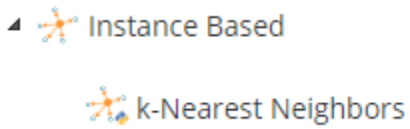
14.1.9. Instance-based Algorithm

The Instance-based algorithms also known as memory-based algorithms compare new problem instances with the instances already seen in training and have been stored in memory. These algorithms are sometimes referred to as 'Lazy' computation gets postponed until a new instance is observed.

The Data Science Workbench provides the K-Nearest Neighbor as displayed in the below-given image:

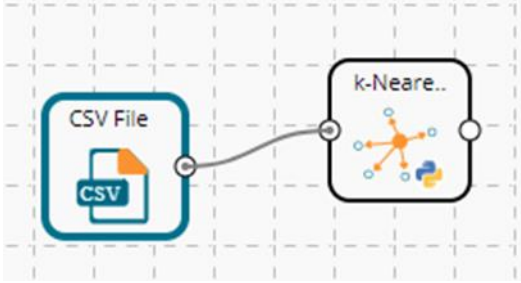
14.1.9.1. K-Nearest Neighbor (KNN)

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to understand and use, but it becomes significantly slow while dealing with a large amount of data.



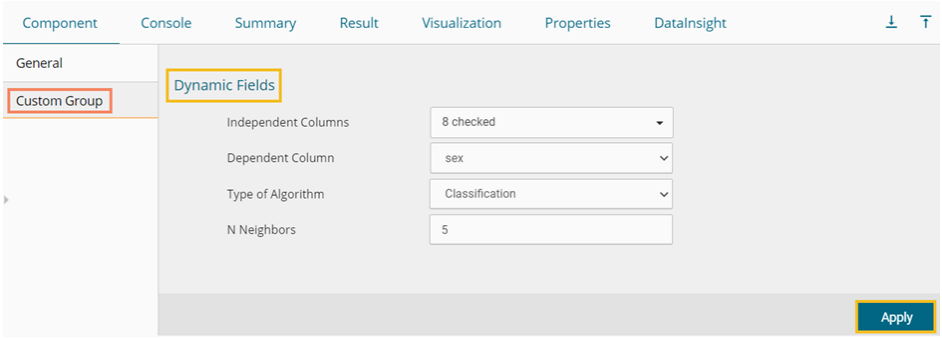
i) Drag the K-Nearest Neighbor (KNN) algorithm component to the workspace and connect it with a configured data source.

ii) Click the K-Nearest Neighbor (KNN) algorithm component.



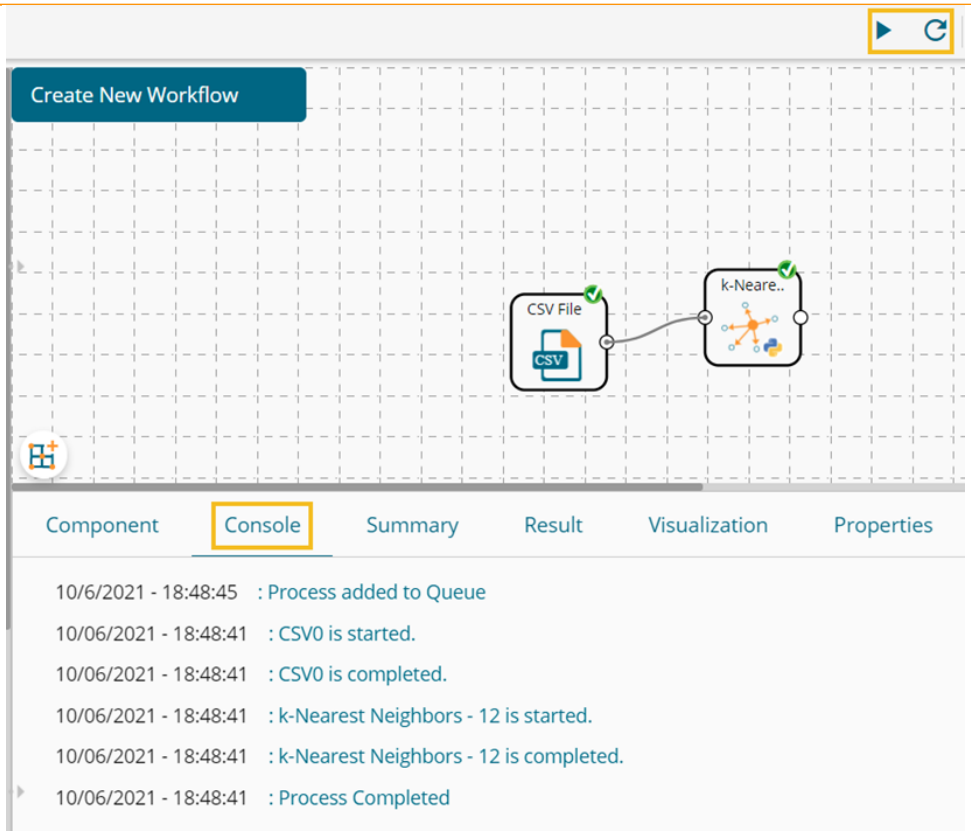
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component.
   a) Independent Columns: Select the required input columns against which the analysis gets applied to the target column.
   b) Dependent Column: Select the target column on which the analysis gets applied.
   c) Algorithm Type: Select an algorithm type out of Classification or Regression.
   d) N Neighbors: Set a value for the N Neighbors (the default value for this field is 5).

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

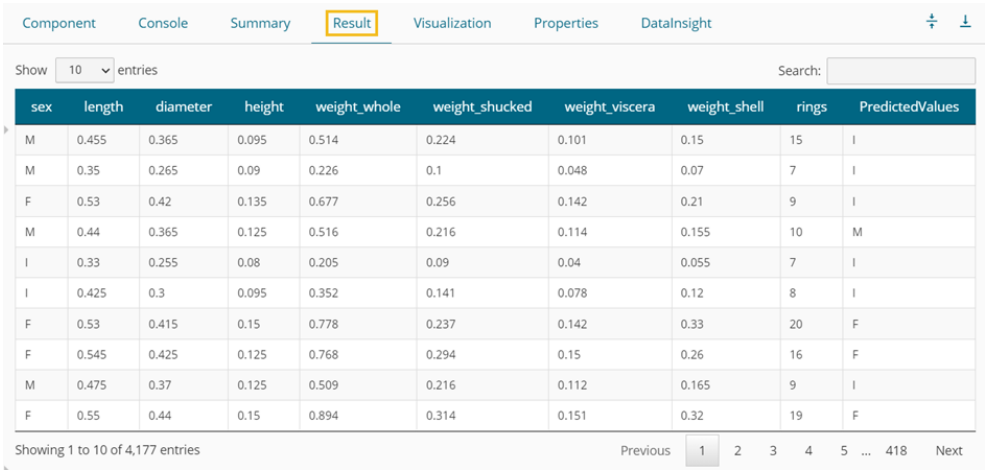
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



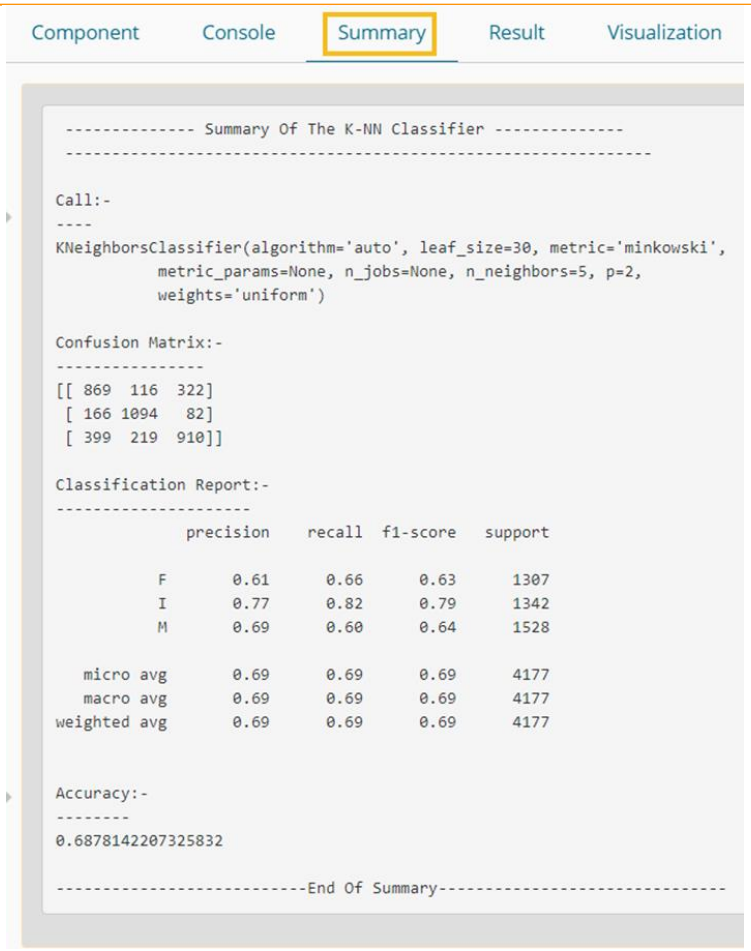
vii) Follow the below-given steps to display the Result view:

a) Click the dragged algorithm component onto the workspace.

b) Click the '**Result**' tab.

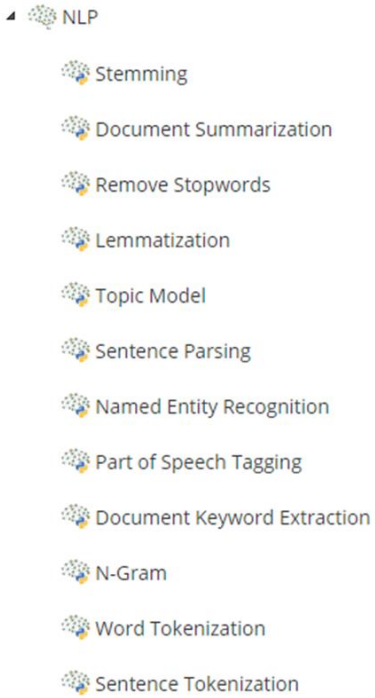


viii) Click the '**Summary**' tab to open the model summary.



14.1.10. NLP

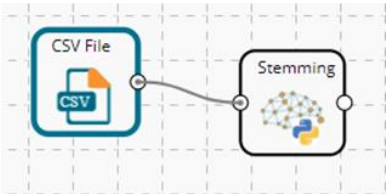
The following algorithms are included under the NLP algorithm tree node.



14.1.10.1. Stemming

i) Drag the Stemming algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.

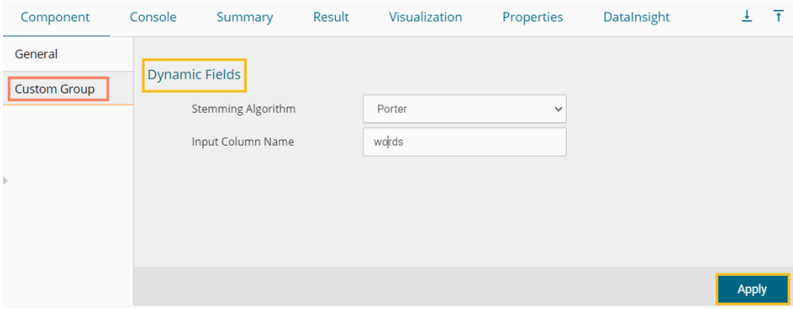


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:

a. Stemming Algorithms

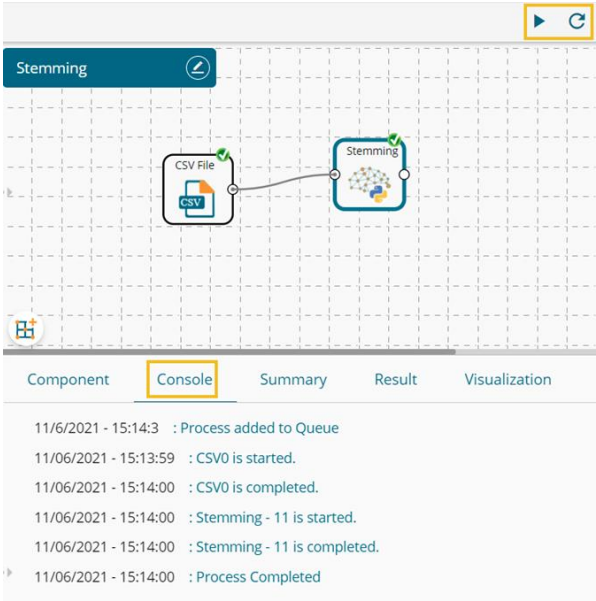
b. Input Column Name

iv) Click the '**Apply**' option.

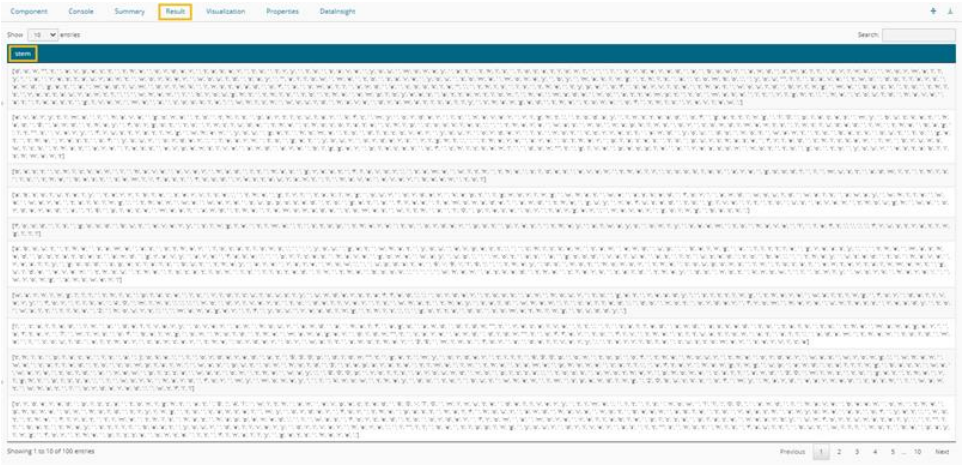


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



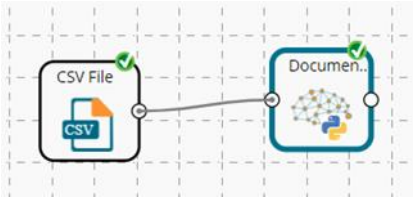
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.2. Document Summarization

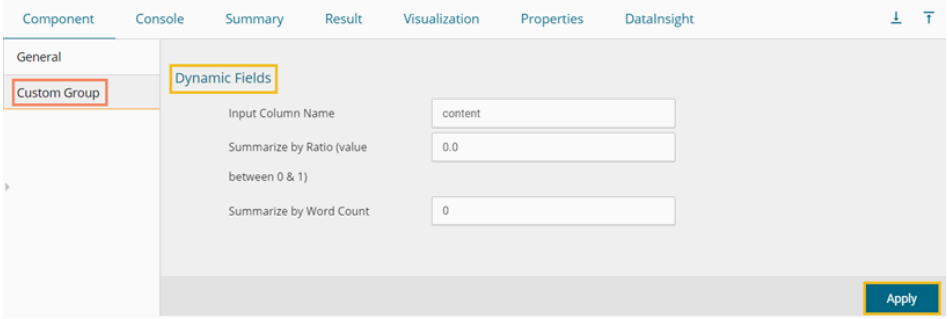
i) Drag the Document Summarization algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



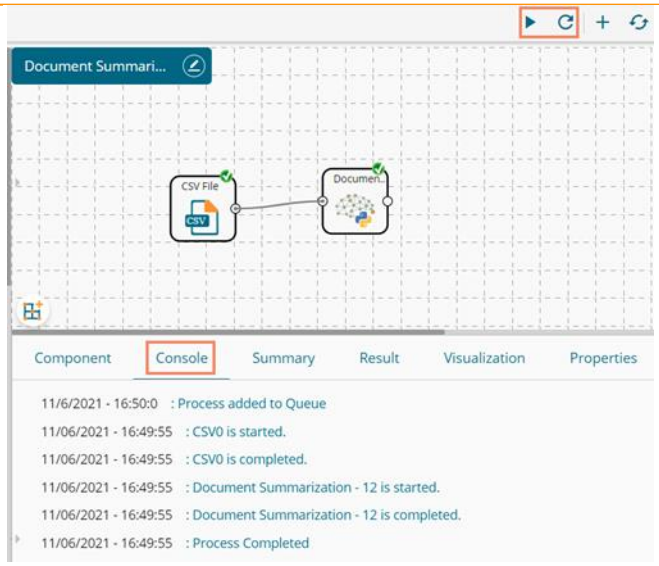
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name
   b. Summarize by Ratio (value between 0 to 1)
   c. Summarize by Word Count

iv) Click the '**Apply**' option.

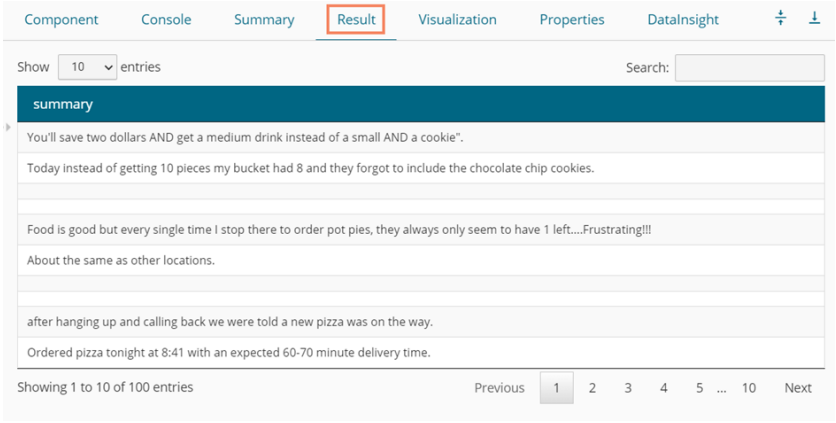


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

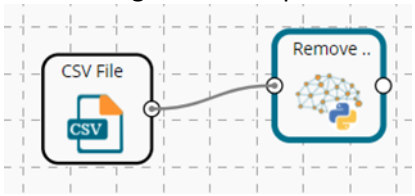


vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.

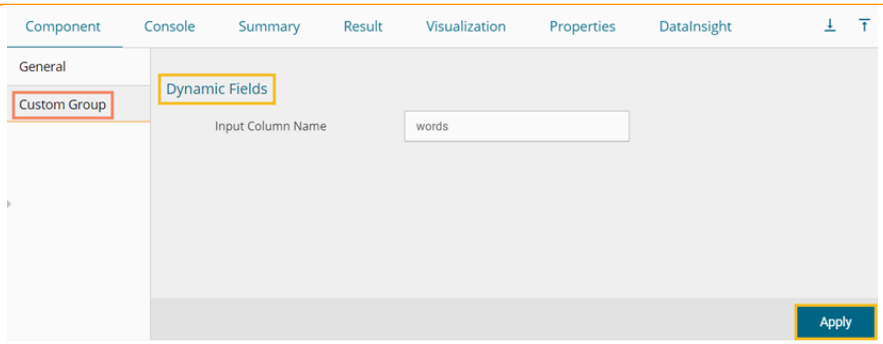


14.1.10.3. Remove Stopwords

i) Drag the algorithm component to the workspace and connect it with a configured data source.
ii) Click the algorithm component to open the configuration tab.

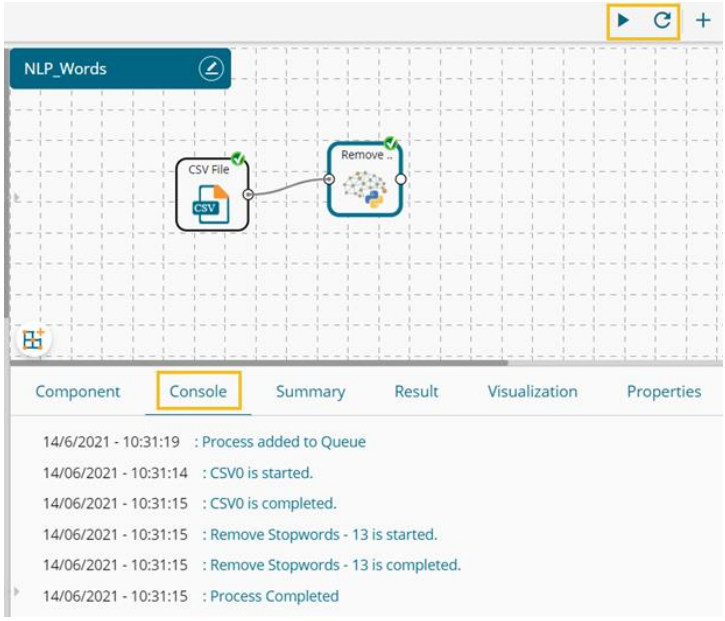


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name
iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

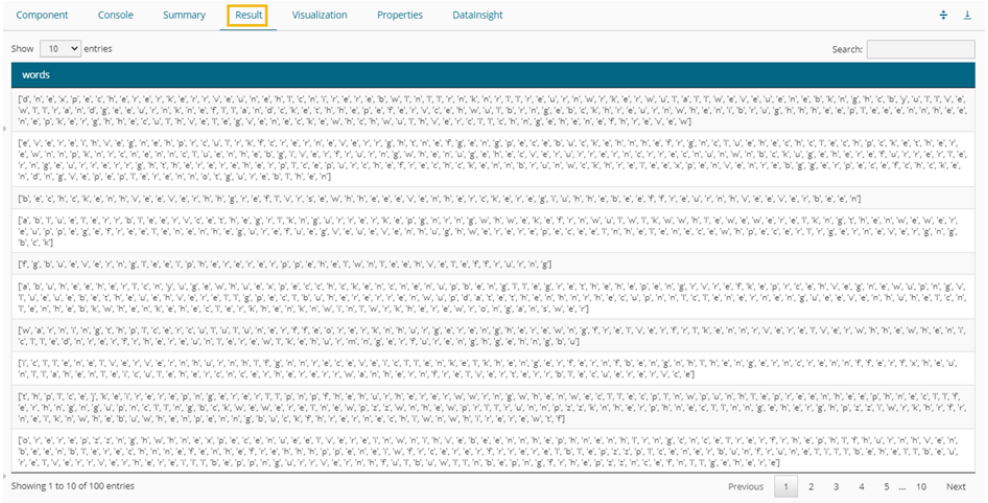
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

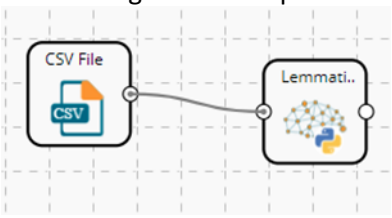
b. Click the '**Result**' tab.



14.1.10.4. Lemmatization

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

ii) Click the algorithm component to open the configuration fields.

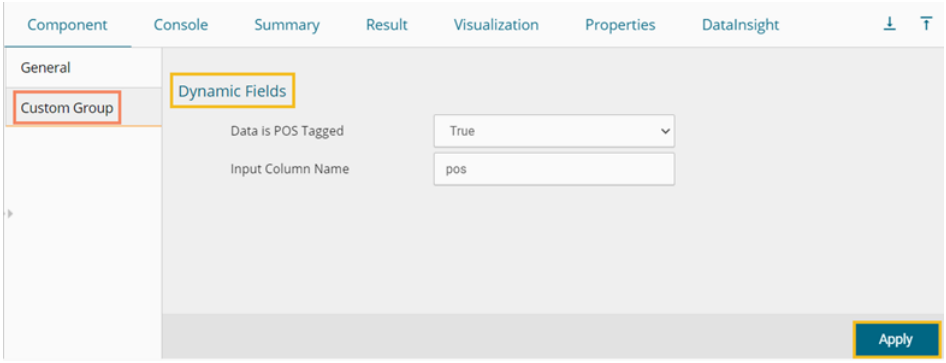


iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:

a. Data is POS Tagged: Select either the 'True' or 'False' option from the given menu.

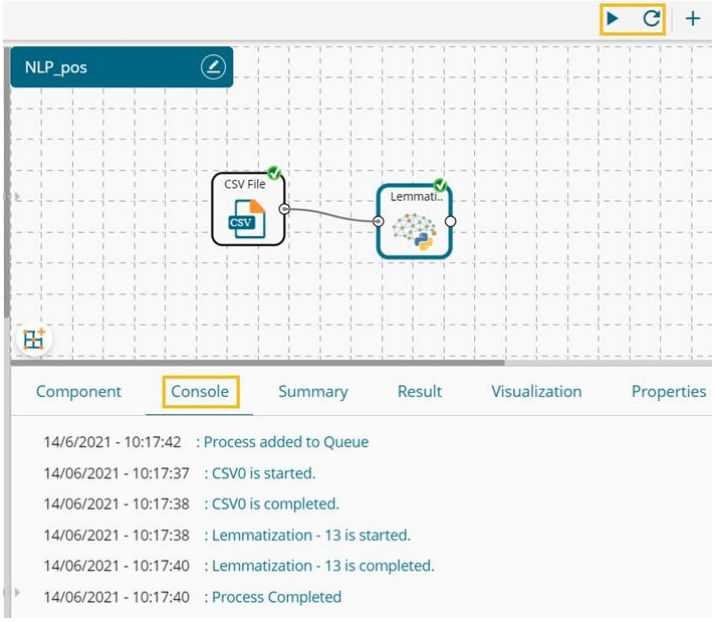
b. Input Column Name: Provide the name for the input column.

iv) Click the '**Apply**' option.

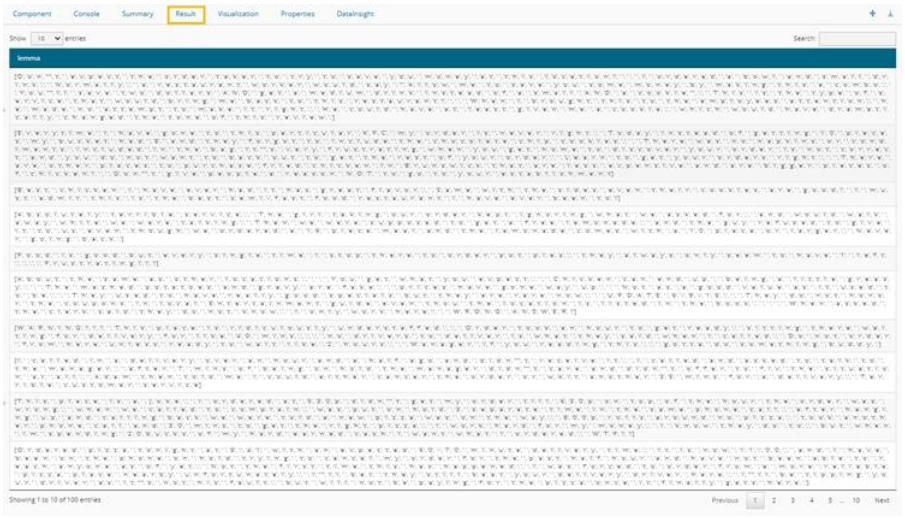


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



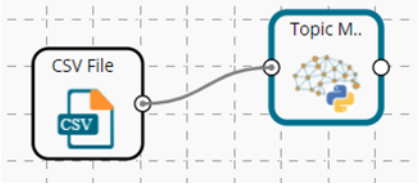
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.5. Topic Model

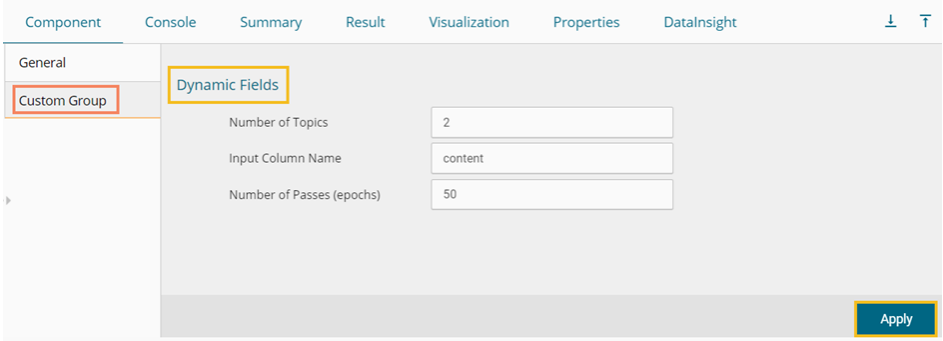
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



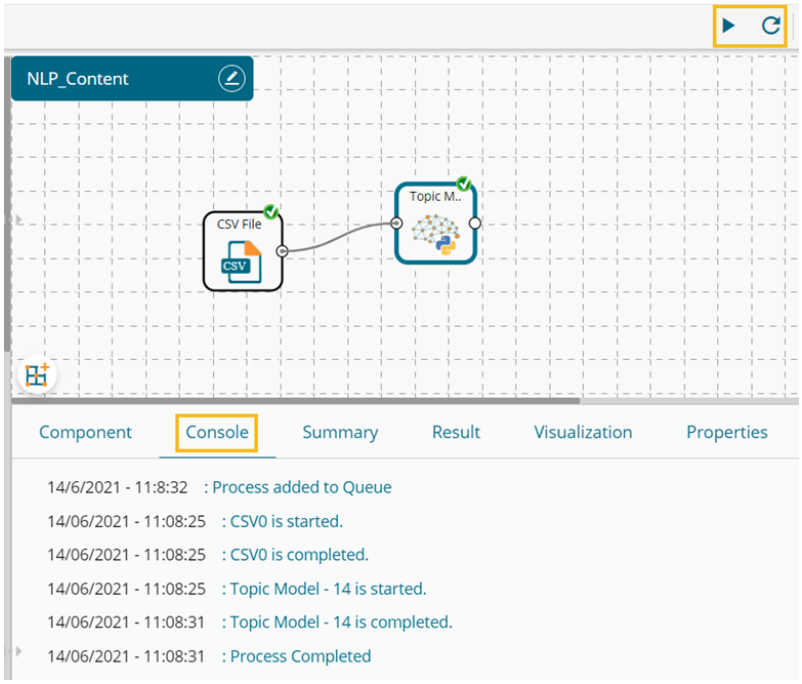
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Number of Topics
   b. Input Column Name
   c. Number of Passes (epochs)

iv) Click the '**Apply**' option.

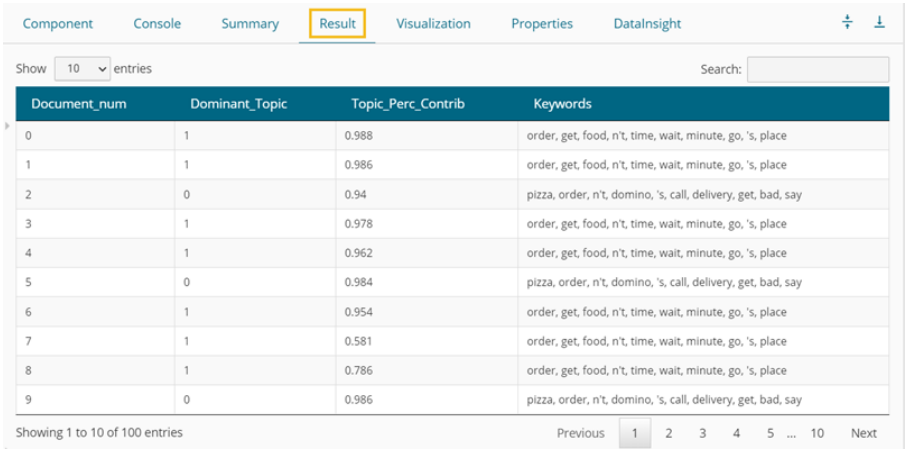


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

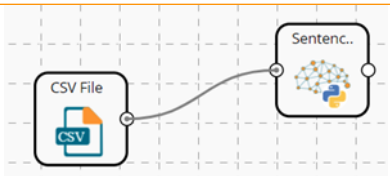


vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



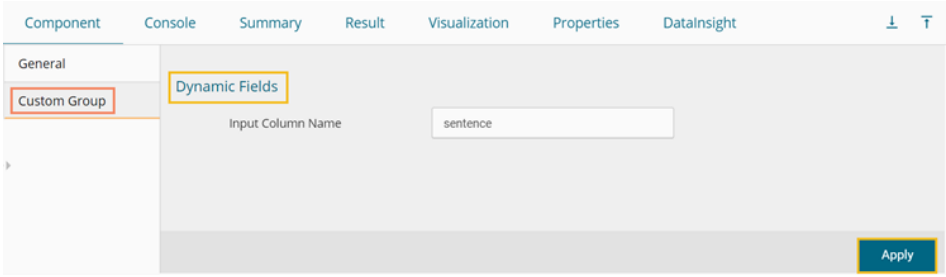
14.1.10.6. Sentence Parsing

i) Drag the algorithm component to the workspace and connect it with a configured data source.
ii) Click on the algorithm component to open the configuration fields.



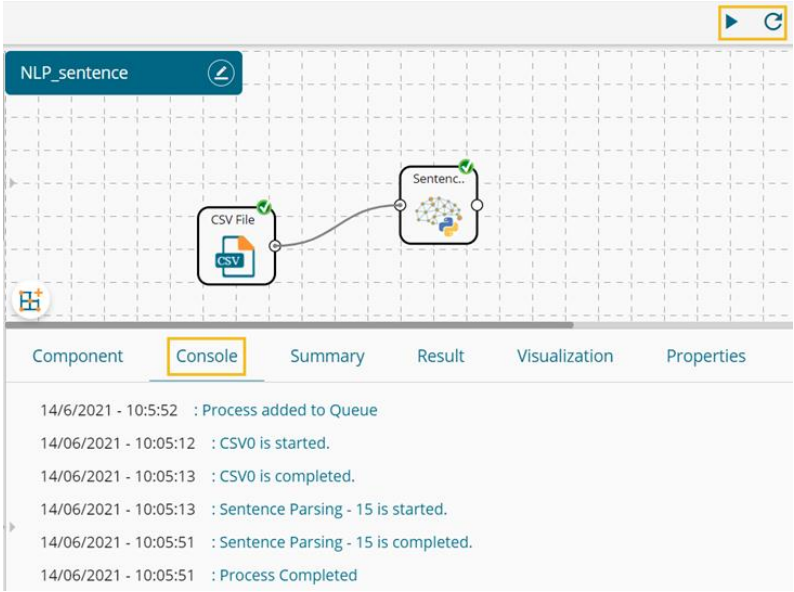
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name

iv) Click the '**Apply**' option.

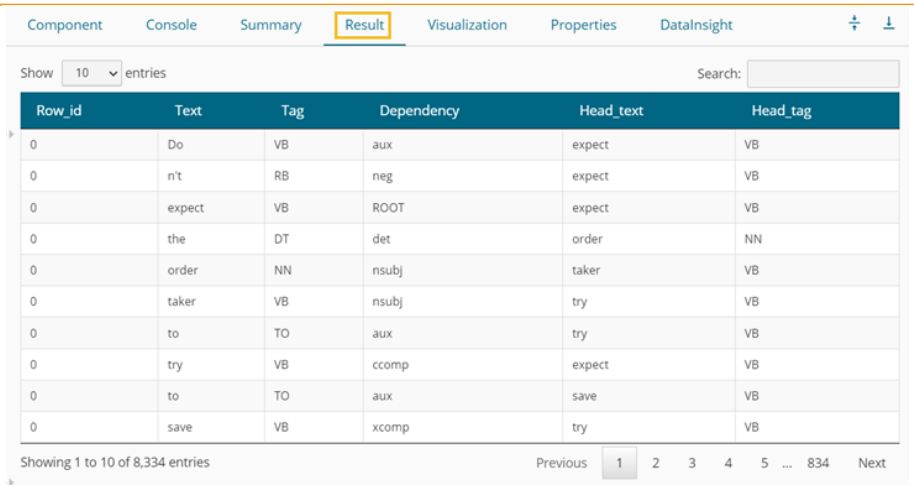


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



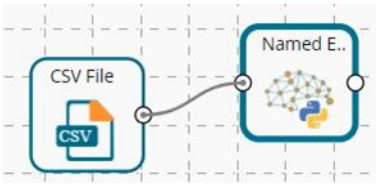
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.7. Named Entity Recognition

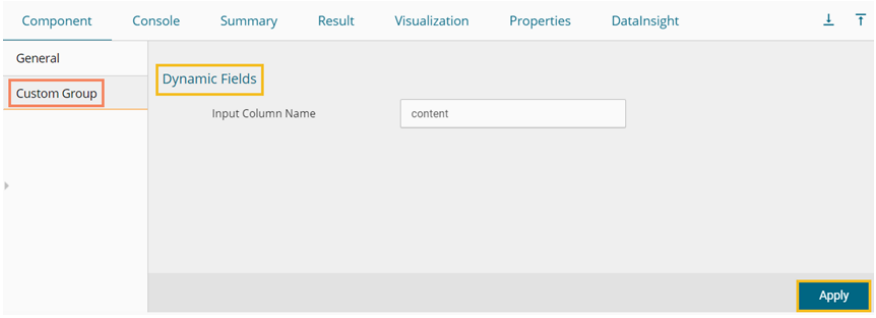
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



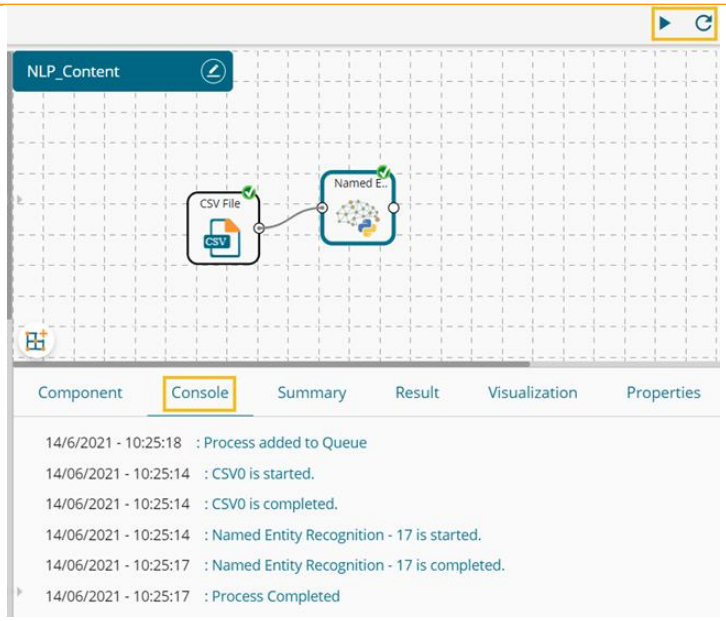
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name

iv) Click the '**Apply**' option.

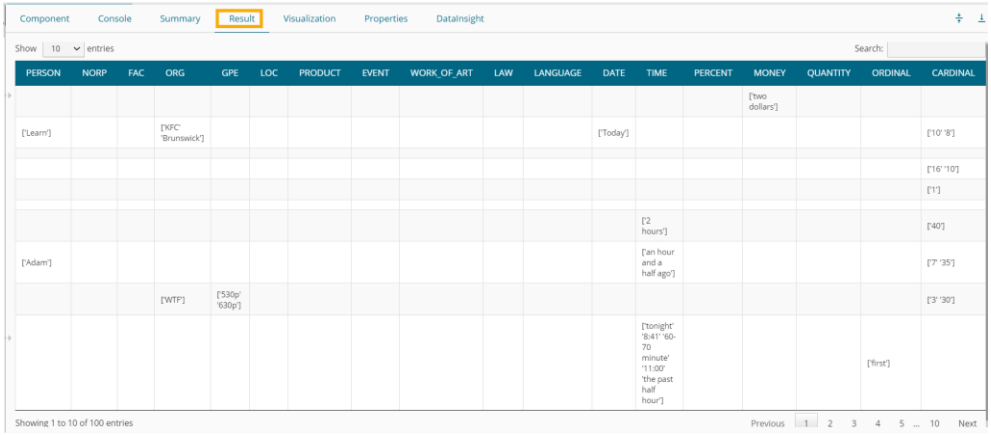


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



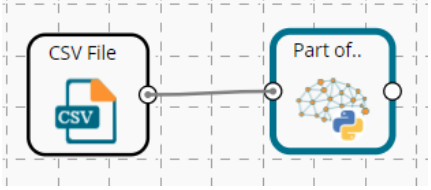
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.8. Part of Speech Tagging

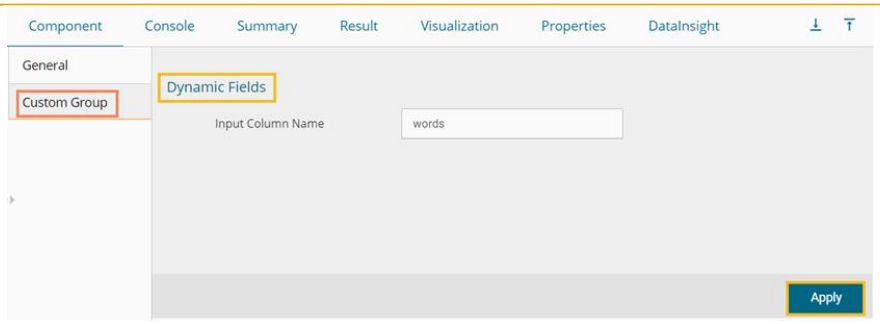
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



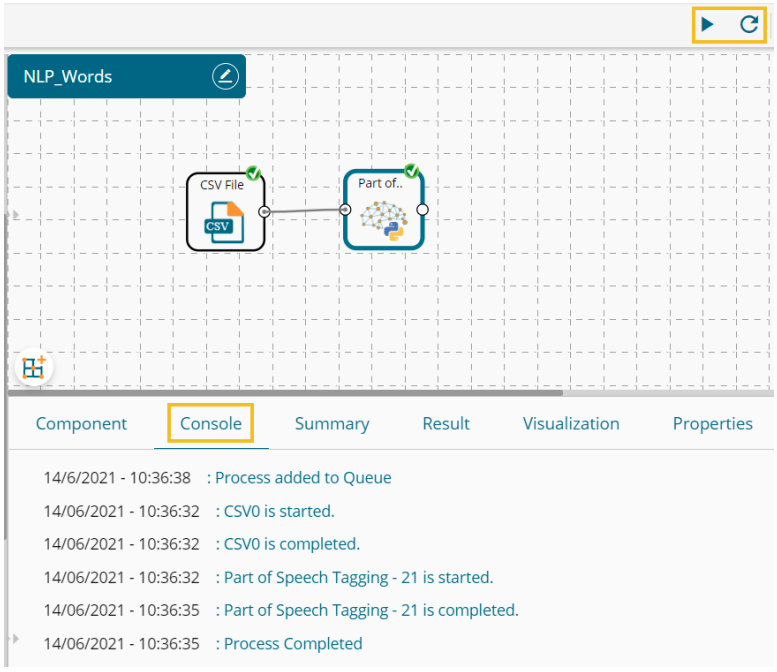
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

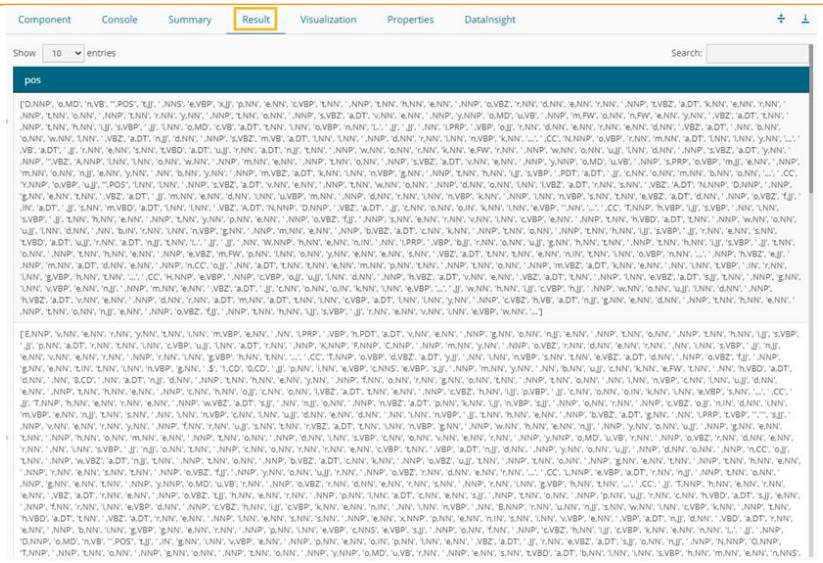
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

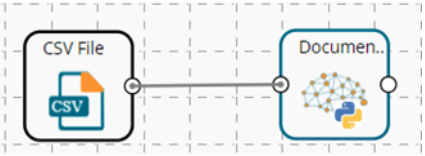
b. Click the '**Result**' tab.



14.1.10.9. Document Keywords Extraction

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

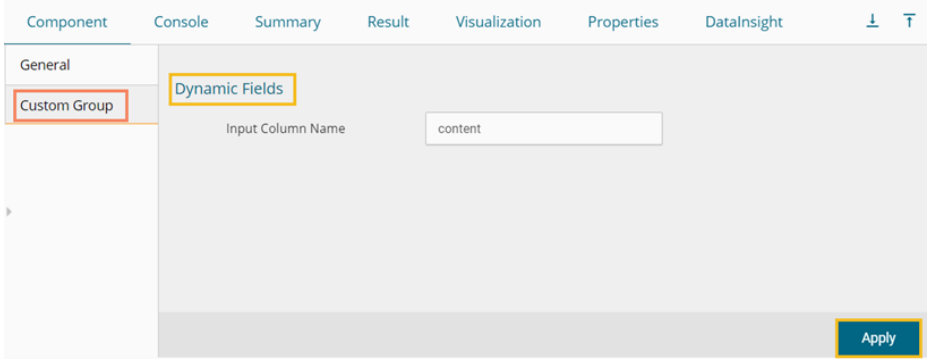
ii) Click the algorithm component to open the configuration fields.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:

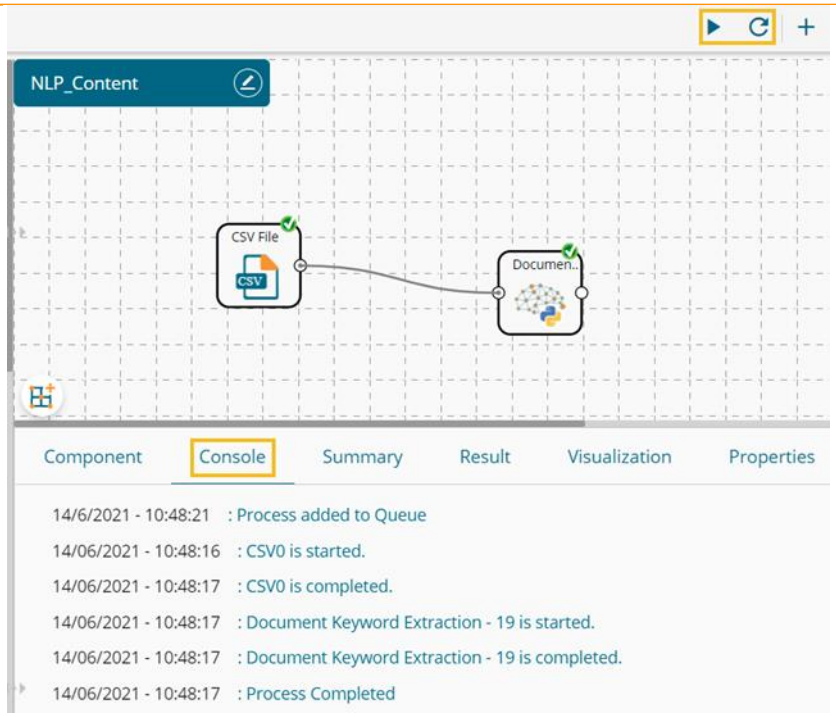
a. Input Column Name

iv) Click the '**Apply**' option.

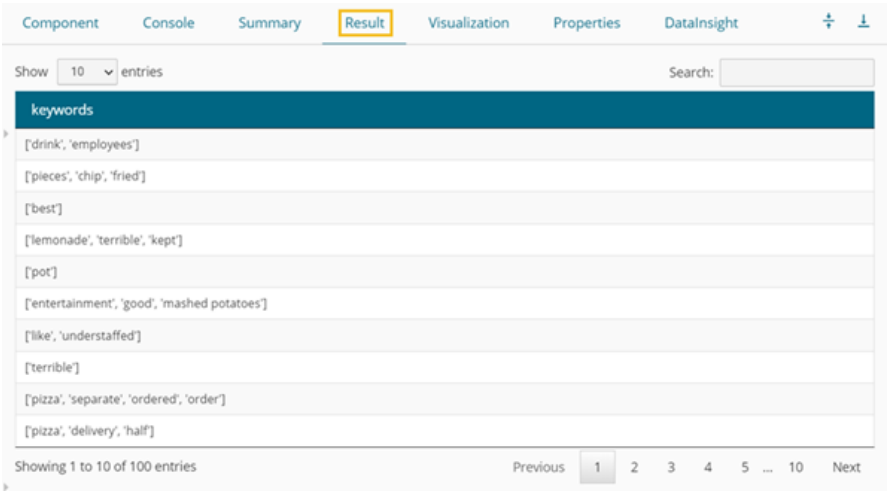


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.

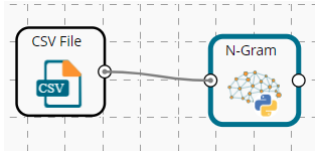


vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.10. N-Gram

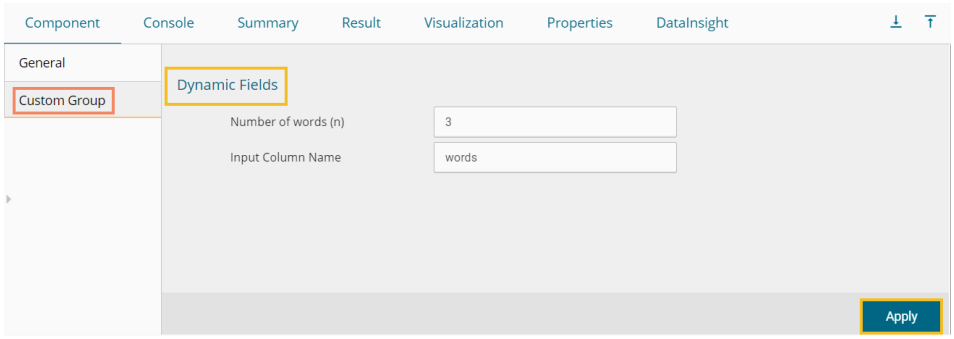
i) Drag the algorithm component to the workspace and connect it with a configured data source.
ii) Click on the algorithm to open the configuration fields.



iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Number of words (0)

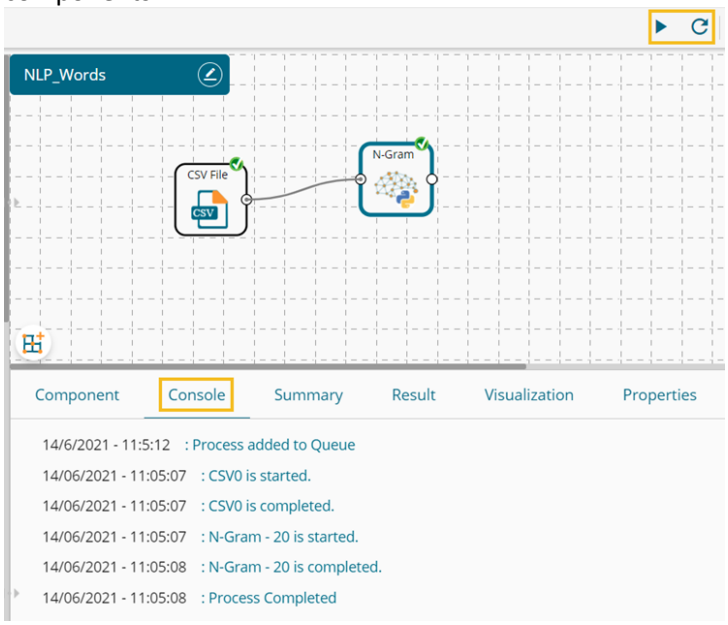
b. Input Column Name

iv) Click the '**Apply**' option.



v) Run the workflow after getting the success message.

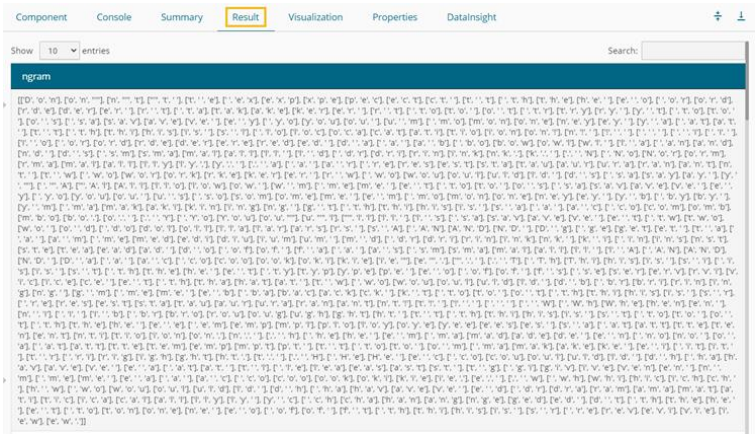
vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



vii) Follow the below-given steps to display the Result view:

a. Click the dragged algorithm component onto the workspace.

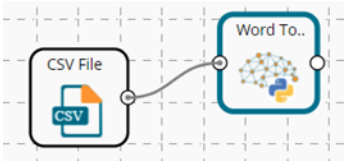
b. Click the '**Result**' tab.



14.1.10.11. Word Tokenization

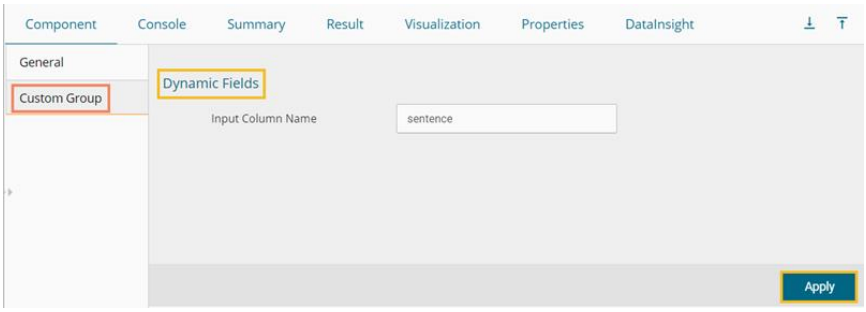
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



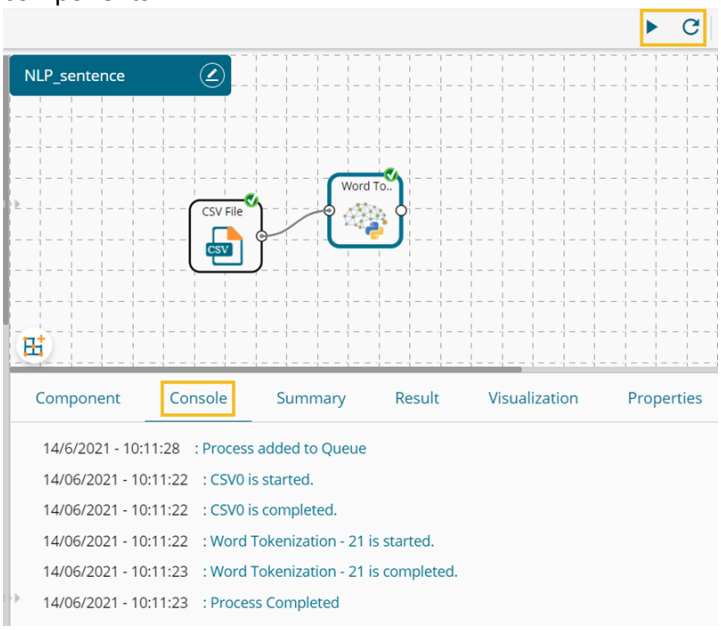
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name

iv) Click the '**Apply**' option.

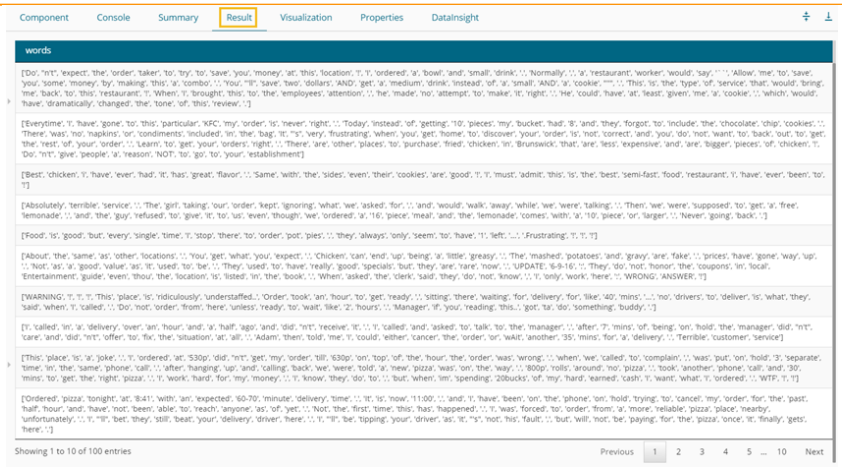


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



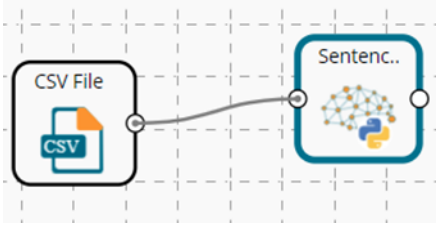
vii) Follow the below-given steps to display the Result view:
   a. Click the dragged algorithm component onto the workspace.
   b. Click the '**Result**' tab.



14.1.10.12. Sentence Tokenization

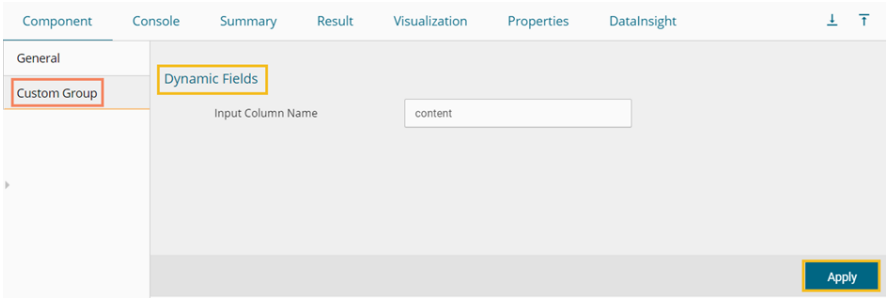
i) Drag the algorithm component to the workspace and connect it with a configured data source.

ii) Click the algorithm component to open the configuration fields.



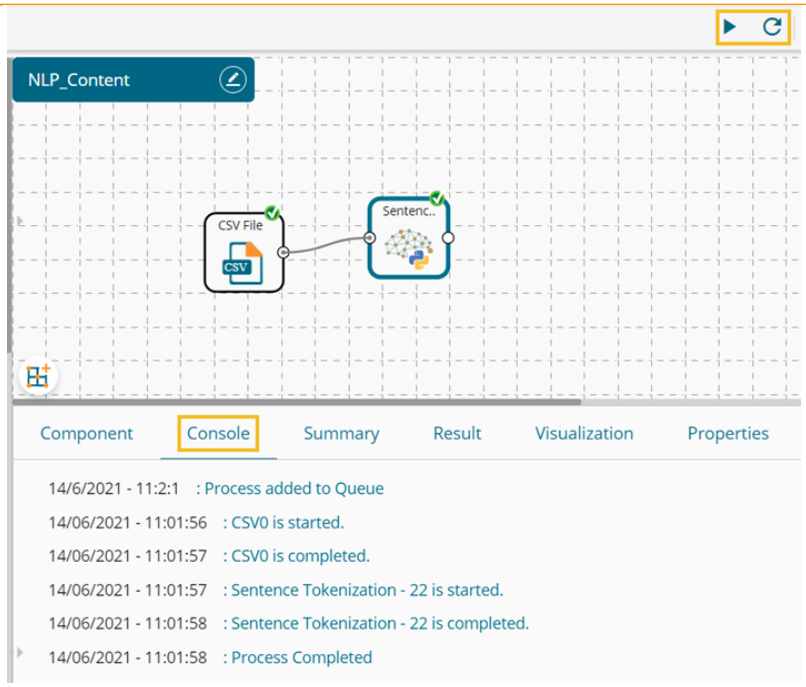
iii) Configure the '**Dynamic Fields**' of the dragged algorithm component by providing the following details:
   a. Input Column Name

iv) Click the '**Apply**' option.

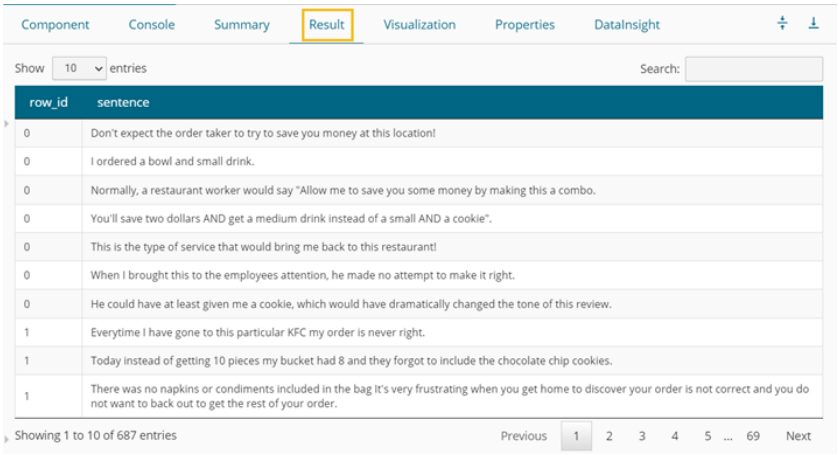


v) Run the workflow after getting the success message.

vi) The Console tab opens displaying the step-by-step completion of the process. The completion of the console process gets marked by the green checkmarks on the top of the dragged components.



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.

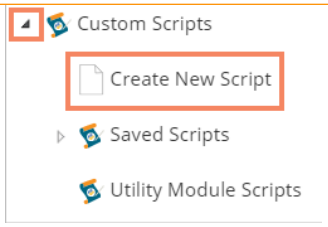


14.2. Custom Scripts (Python Scripts)

The 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 Workspace.

14.2.1. Creating a New Python Script

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



iii) The users get the '**Component**' tab.

iv) Configure the following fields in the '**General**' tab:

a. Basic

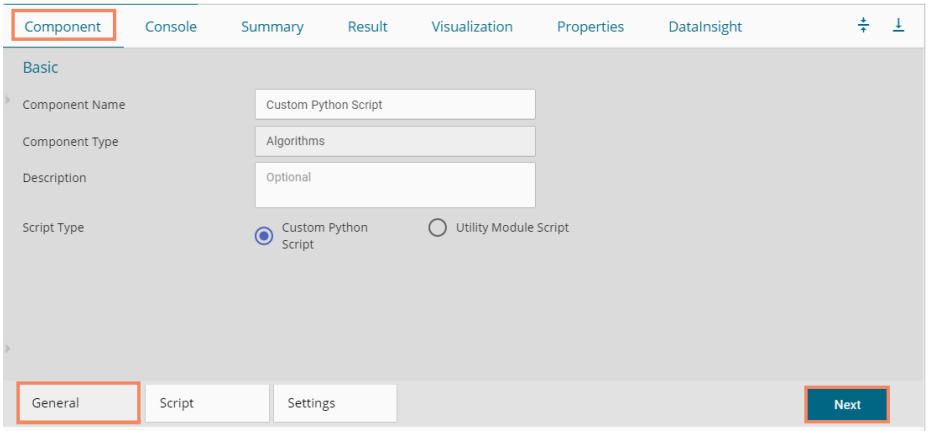
i. **Component Name:** Enter a name or title that you wish to give a saved Python Script.

ii. **Component Type:** Default Component type will be displayed in this field.

iii. **Description:** Describe the Component (It is an optional field).

iv. **Script Type:** Select one option out of '**Custom Python Script'** or '**Utility Module Script**' for the script to get saved under the selected script type.

v) Click the **'Next'** option.



vi) The users get redirected to the '**Script**' tab.

vii) Provide the following information:

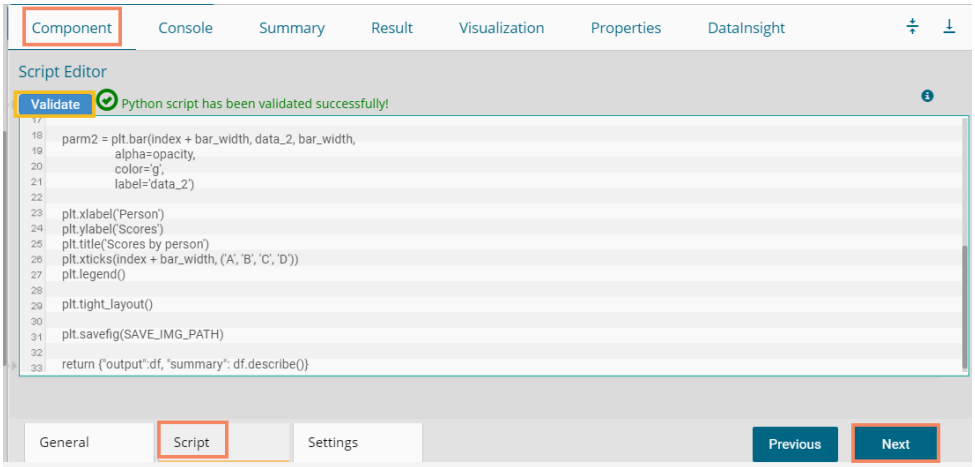
a. Script Editor

i. Write the required python script in the given space under the **'Script Editor.'**

ii. Click the '**Validate'** option.

iii. A success message should appear after the validation (as shown in the below image).

iv. Click the '**Next**' option.



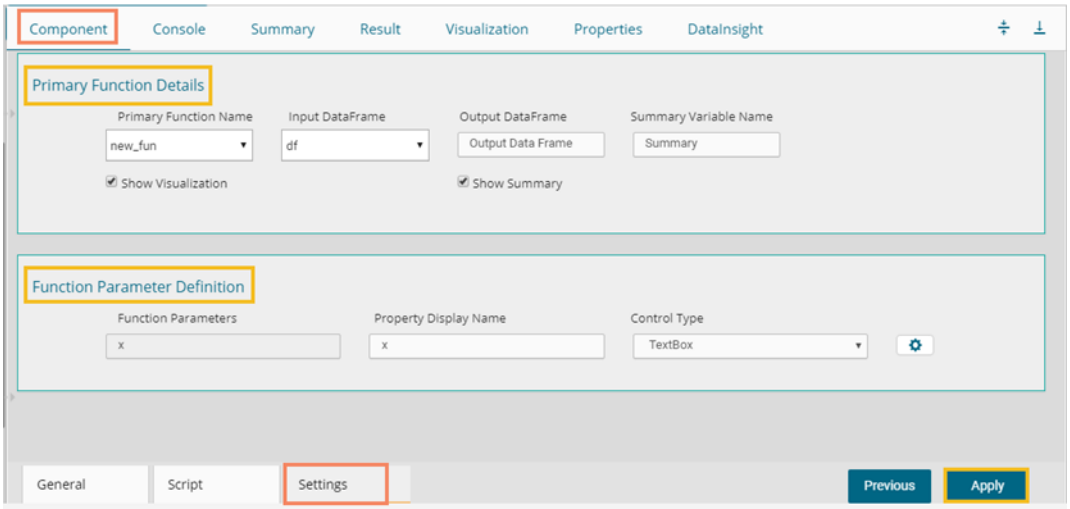
b. Configure the required **'Primary Function Details'** to embed the customized Python script into a function.

i. **Primary Function Name:** Select the name of the created functio**n** from the drop-down menu.

ii. **Input Data Frame:** Select a dataset (that has been used above) from a drop-down menu.

(The '**Output Data Frame**' option and the '**Model Variable Name**' are pre-selected for the Primary Function Details)

viii) Click the **'Next'** option (The users can click the '**Previous**' option if wish to open the previous page).



ix) The users get directed to the '**Settings**' tab.

x) Configure the following fields:

a. Primary Function Details

This option configures the following details:

i. Primary Function Name: Select an option from the drop-down menu.

ii. Input Data Frame: Select an option from the drop-down menu.

iii. Output Data Frame: Provide a name for the Output Data Frame.

iv. Summary Variable Name: Provide a name for the Summary Variable Name.

v. The user can select the '**Show Visualization**' and '**Show Summary**' options from this section.

b. Function Parameter Definition

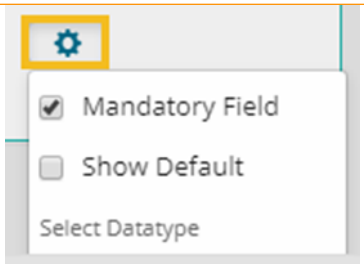
i. **Function Parameters**: Actual names of parameters configured in the script.

ii. **Property Display Name**: Parameter name to be displayed while configuring the saved script as a component.

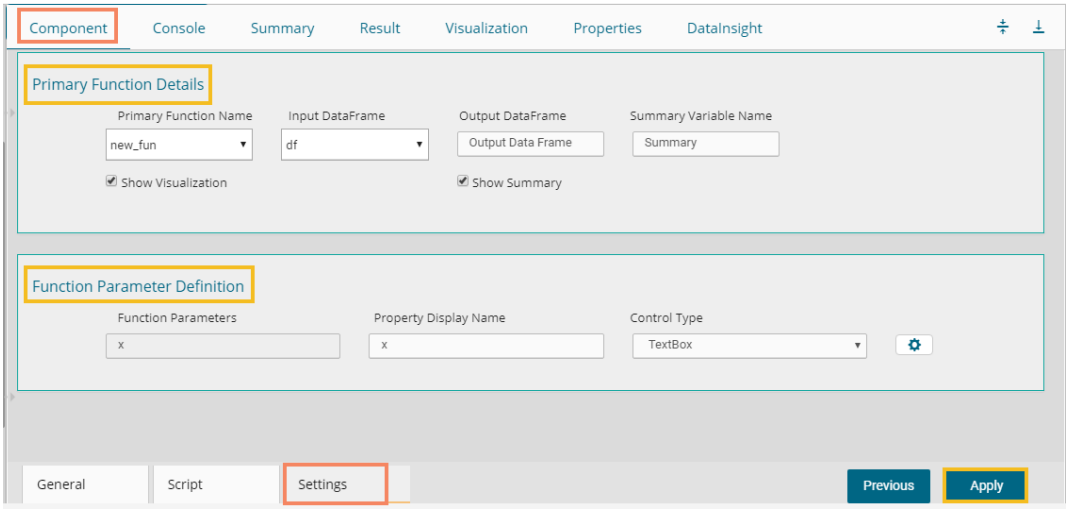
iii. **Control Type:** User can select out of the following options:

1. Text box,
2. Drop-down menu,
3. Column Selector (single),
4. Column Selector (multiple).

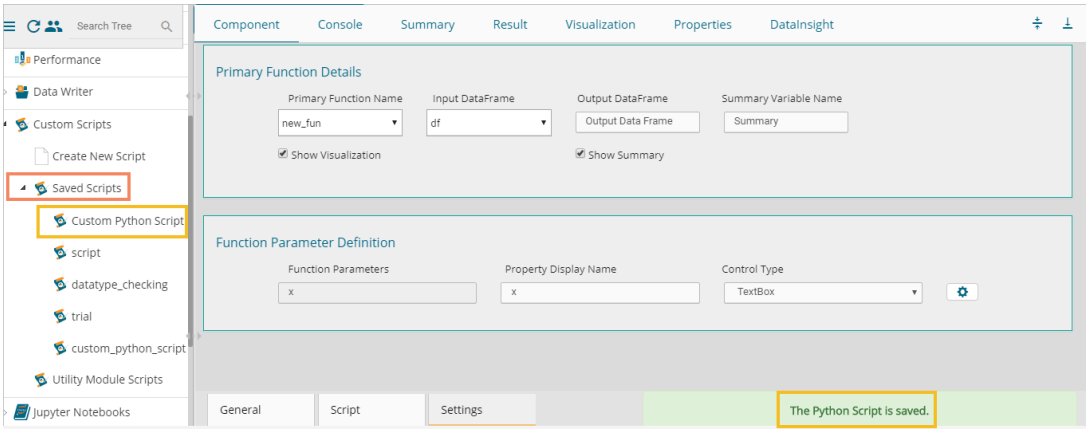
iv. **Settings option**  **:** To set the display for mandatory fields and validate the datatype for the input column. This field is associated with function parameters.



xi) Click the '**Apply**' option.



xii) A message appears to notify that the newly created Python script has been saved successfully.
xiii) The newly created Python Script gets saved in the **'Saved Scripts'** list.



Guidelines for Writing a Python Script

1. The first argument of the function should be a data frame.
2. The Python script needs to be written inside a valid Python function. E.g., the entire code body should be inside the proper indentation of the function (Use 4 spaces per indentation level.)
3. The Python script should have at least one primary function. Multiple functions are acceptable, and one function can call another function, but it should be written above the calling function body (if the called function is an outer function) or above the calling statement (if the called function is an inner function).

4. Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets, and braces, or using a hanging indent. When using a hanging indent, the following should be considered; there should be no arguments on the first line, and further indentation should be used to distinguish itself as a continuation line clearly.
5. Spaces are the preferred indentation method.
6. Limit all lines to a maximum of 79 characters. The Python standard library is conservative and requires limiting lines to 79 characters (and doctrines/comments to 72).
7. Do not use "type" as the function argument, as it is a predefined keyword.
8. In Python, single-quoted strings and double-quoted strings are the same.
9. All the packages used in the function need to import explicitly before writing the function.
10. The Python script should return data in the form of a data Frame only and should define while writing the function.
11. The column names should remain the same while creating new columns in the Output Table Definition.
12. If users need to define column selector (Multiple), then in the definition ': List[String]' should be used and the body of the function should be in '.to Array'.
13. If users need to define a column selector (Single), then 'String' must be used in the definition.

Note:

a. Click the '**Information**' button to get the rules to write a valid Python script.

b. All the supported date data types are listed in date formats in the data type definition, all other date formats are considered string data types.

c. Mssql data types are considered string data types.

14.2.2. Saved Python Scripts

14.2.2.1. Viewing a Saved Python Script

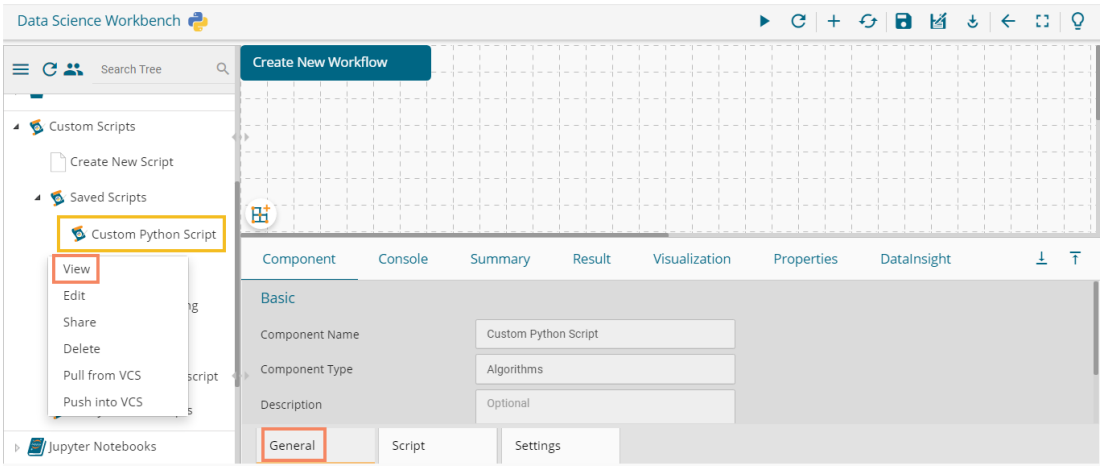
i) Select a Script from the '**Saved Scripts**' list.

ii) Use right-click on the selected Script.

iii) A context menu opens.

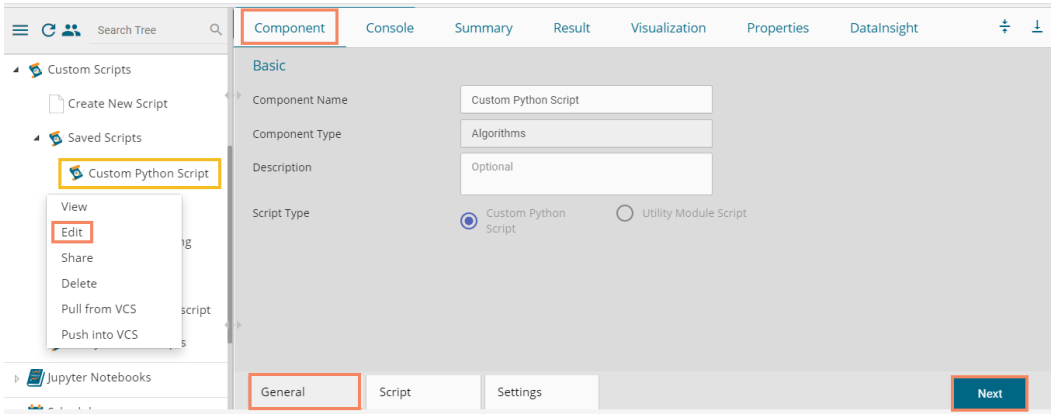
iv) Select the '**View**' option.

v) The users get redirected to the '**Component**' tab.



14.2.2.2. Editing a Saved Python Script

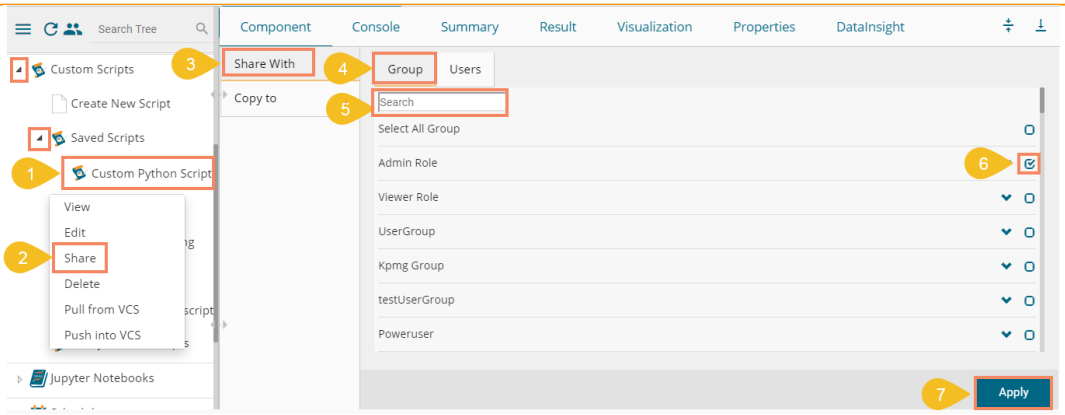
i) Select a Script from the list of '**Saved Scripts**' list.
ii) Use a right-click on the selected script.
iii) A context menu opens.
iv) Select the '**Edit**' option.
v) The users get redirected to the '**Component**' tab.
vi) The users can edit the required fields provided under the **General**, **Script**, and **Settings** tabs.



14.2.2.3. Sharing a Saved Python Script

The users can share a custom Python script with other users and groups using the Share option. The following options are available to share a custom Python script:

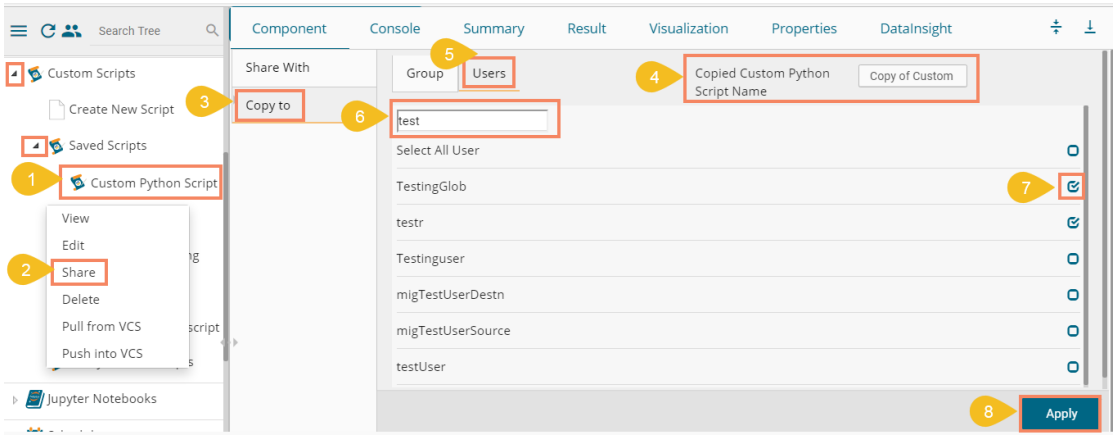
1. **Share With**: This option allows the user to share a custom Python script with selected users or user groups. Any changes made to the custom Python script will be transferred to all the users with whom the custom Python script has been shared.
   i) Select a Python script from the list of **Saved Scripts**.
   ii) Select the '**Share**' option from the context menu.
   iii) The '**Share With**' option gets displayed (by default).
   iv) Select either the '**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 when the '**Group**' option has been selected.
      b. The users can be excluded by not selecting a username from the list when the '**Users**' option has been selected.
   v) Search for specific users or groups by using the Search space.
   vi) Select a specific user or group from the list by check-marking the box.
   vii) Click the '**Apply**' option.



viii) The selected Python script gets shared with the chosen user(s)/group(s).

2. **Copy To**: This option creates a copy and shares a 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.

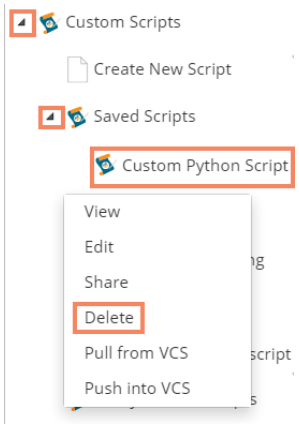
   i) Select a Python script from the list of '**Saved Scripts**'.
   ii) Select the '**Share**' option from the context menu.
   iii) Select the '**Copy To**' option.
   iv) The copied custom Python script name will be displayed in a box.
   v) Select either the '**Group**' or '**Users**' tab.
      a. By selecting a group, all group members inside the group will be listed. Users can be excluded by not selecting them from the group when the '**Group**' option has been selected.
      b. Users can be excluded by not selecting a username from the list when the '**Users**' option has been selected.
   vi) Search for a user or group by using the search space.
   vii) Select a specific user or group from the list by check-marking the box.
   viii) Click the '**Apply**' option.



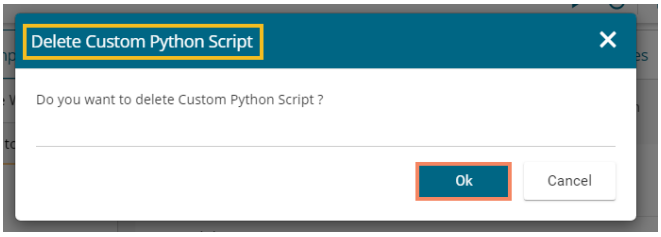
14.2.2.4. Deleting a Saved Python Script

i) Select a Python Script from the '**Saved Scripts**' list.
ii) Right-click on the selected Scala Script.

iii) A context menu opens.
iv) Select the '**Delete**' option.



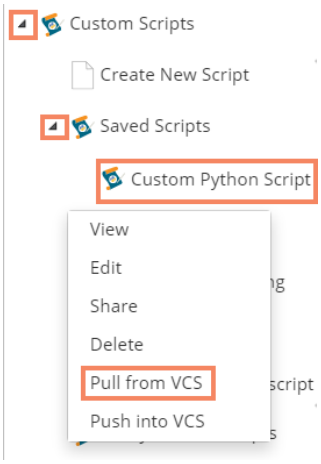
v) The Delete Custom Python Script window opens to assure the deletion.
vi) Click the '**Ok**' option.



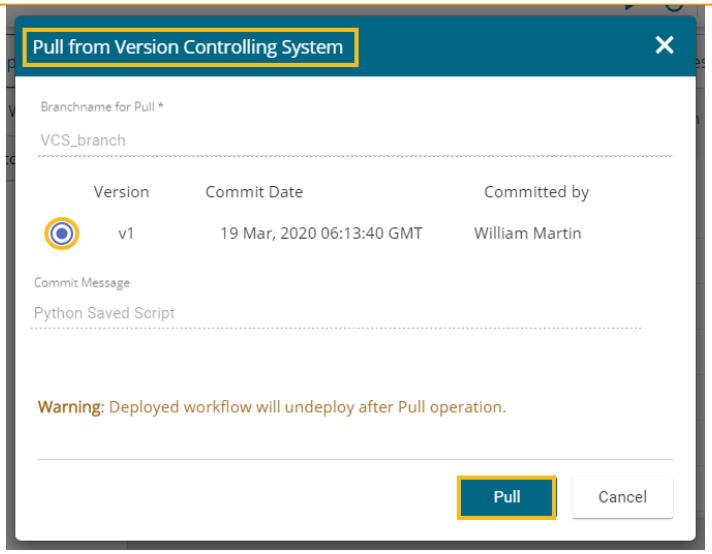
vii) The selected script gets deleted.

14.2.2.5. Pull from VCS

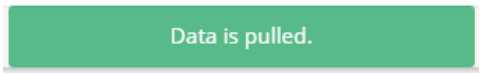
i) Select a saved script.
ii) Select the '**Pull from VCS**' option from the context menu.



iii) The '**Pull from Version Controlling System**' window opens.
iv) Select the version(s) of the script that you wish to pull.
v) Click the '**Pull**' option.



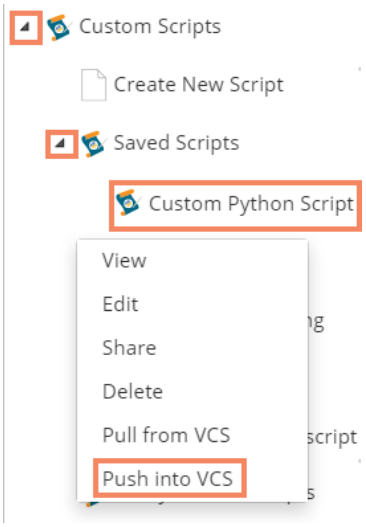
vi) A message appears to confirm that the data is pulled.



14.2.2.6. Push into VCS

vii) Select a saved script.

viii) Select the '**Push into VCS**' option from the context menu.

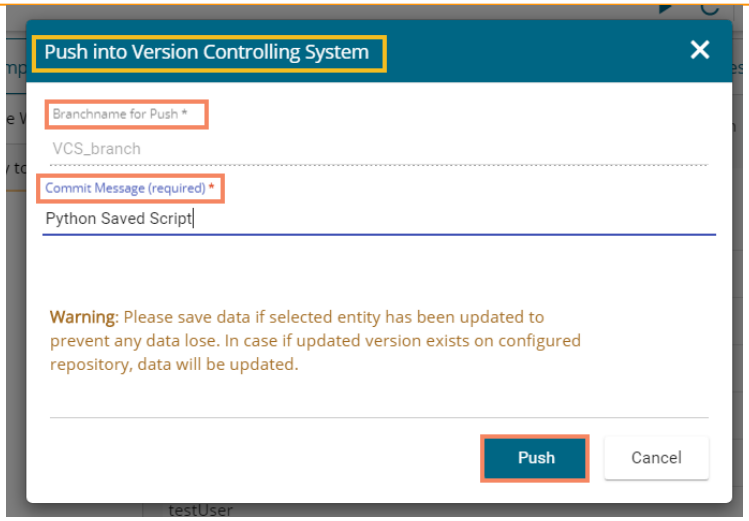


ix) The '**Push into Version Controlling System**' window opens.

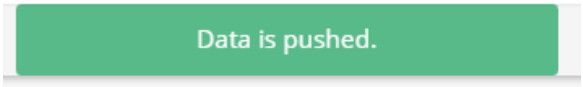
x) Select a branch name for the push.

xi) Provide the commit message.

xii) Click the '**Push**' option.

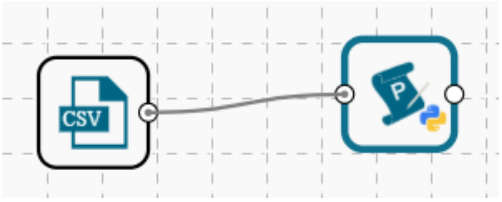


xiii) A success message appears to confirm that the data has been pushed.

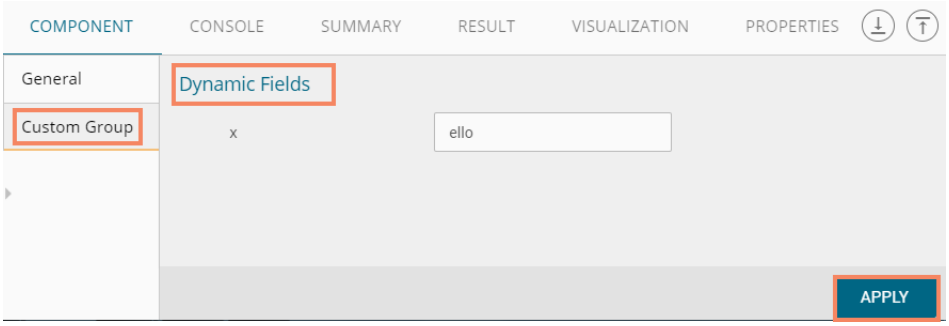


14.2.2.7. Connecting Saved Python Script with a Data Source

i) Click the **'Custom Python Script'** tree node.
ii) Select and drag a saved Python script to the workspace.
iii) Connect the Python Script to a configured data source.
iv) Click the dragged '**Python Script'** component.

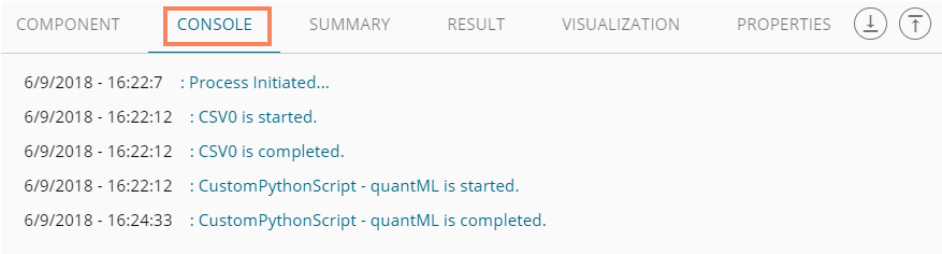


v) Configure the required fields in the '**Custom Group**' tab.
vi) Click **'APPLY'**



vii) After getting the success message run the workflow

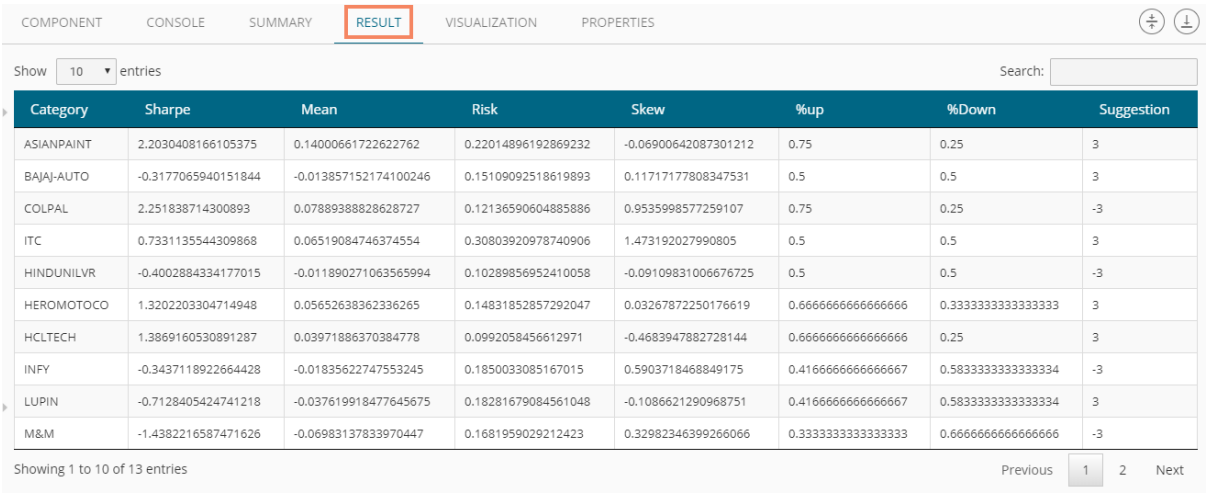
viii) Users will get the process status under the '**CONSOLE**' tab



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

a. Click the dragged Python component on the workspace.

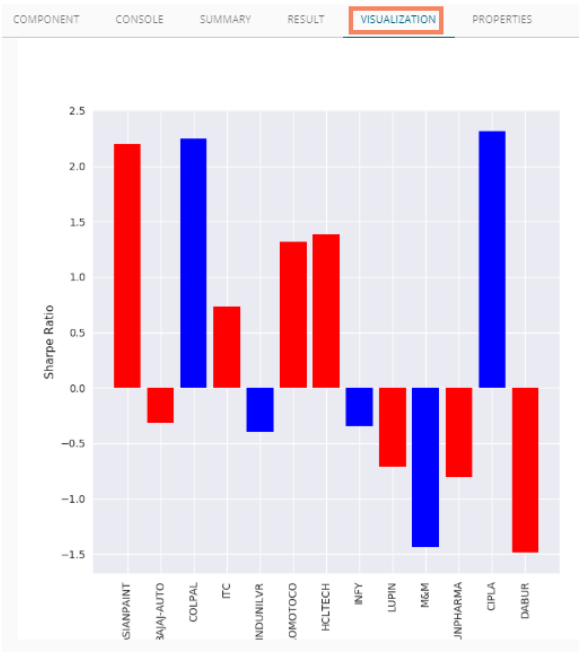
b. Click the '**RESULT**' tab.



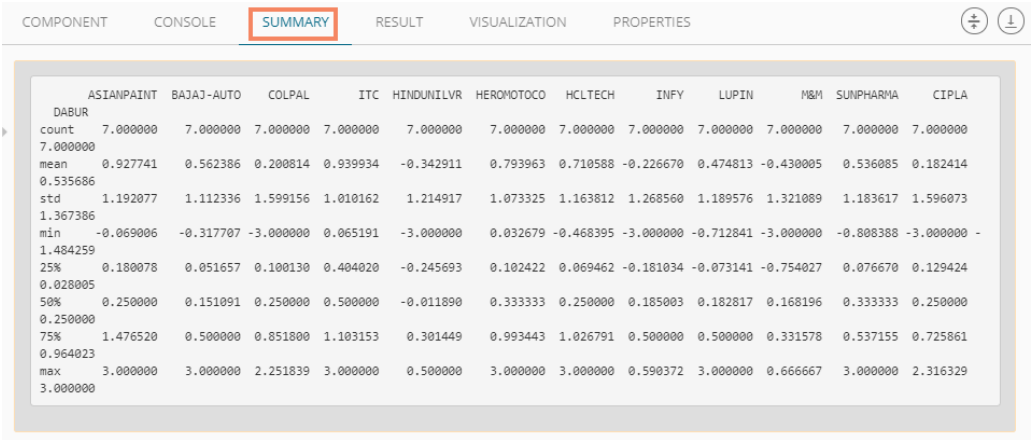
Category	Sharpe	Mean	Risk	Skew	%up	%Down	Suggestion
ASIANPAINT	2.2030408166105375	0.14000661722622762	0.22014896192869232	-0.06900642087301212	0.75	0.25	3
BAJAJ-AUTO	-0.3177065940151844	-0.013857152174100246	0.15109092518619893	0.11717177808347531	0.5	0.5	3
COLPAL	2.251838714300893	0.07889388828628727	0.12136590604885886	0.9535998577259107	0.75	0.25	-3
ITC	0.7331135544309868	0.06519084746374554	0.30803920978740906	1.473192027990805	0.5	0.5	3
HINDUNILVR	-0.4002884334177015	-0.011890271063565994	0.10289856952410058	-0.09109831006676725	0.5	0.5	-3
HEROMOTOCO	1.3202203304714948	0.05652638362336265	0.14831852857292047	0.03267872250176619	0.6666666666666666	0.3333333333333333	3
HCLTECH	1.3869160530891287	0.03971886370384778	0.0992058456612971	-0.4683947882728144	0.6666666666666666	0.25	3
INFY	-0.3437118922664428	-0.01835622747553245	0.1850033085167015	0.5903718468849175	0.4166666666666667	0.5833333333333334	-3
LUPIN	-0.7128405424741218	-0.037619918477645675	0.18281679084561048	-0.1086621290968751	0.4166666666666667	0.5833333333333334	3
M&M	-1.4382216587471626	-0.06983137833970447	0.1681959029212423	0.32982346399266066	0.3333333333333333	0.6666666666666666	-3

Showing 1 to 10 of 13 entries Previous 1 2 Next

x) Click the '**VISUALIZATION**' tab to display the result data through a column chart.



xi) Click the '**SUMMARY**' tab to view a summary of the process.

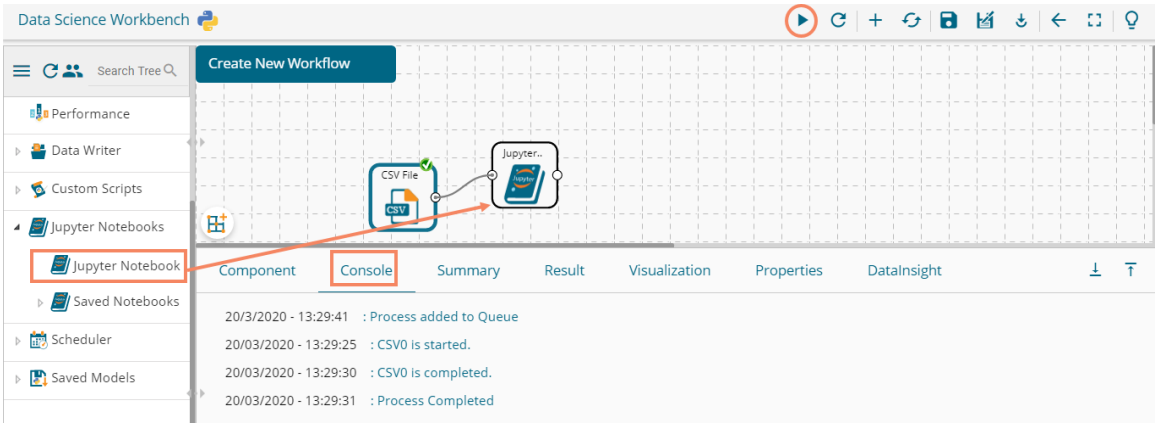


	ASIANPAINT	BAJAJ-AUTO	COLPAL	ITC	HINDUNILVR	HEROMOTOCO	HCLTECH	INFY	LUPIN	M&M	SUNPHARMA	CIPLA	DABUR
count	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000
mean	0.927741	0.562386	0.200814	0.939934	-0.342911	0.793963	0.710588	-0.226670	0.474813	-0.430005	0.536085	0.182414	0.535686
std	1.192077	1.112336	1.599156	1.010162	1.214917	1.073325	1.163812	1.268560	1.189576	1.321089	1.183617	1.596073	1.367386
min	-0.069006	-0.317707	-3.000000	0.065191	-3.000000	0.032679	-0.468395	-3.000000	-0.712841	-3.000000	-0.808388	-3.000000	-1.484259
25%	0.180078	0.051657	0.100130	0.404020	-0.245693	0.102422	0.069462	-0.181034	-0.073141	-0.754027	0.076670	0.129424	0.028005
50%	0.250000	0.151091	0.250000	0.500000	-0.011890	0.333333	0.250000	0.185003	0.182817	0.168196	0.333333	0.250000	0.250000
75%	1.476520	0.500000	0.851800	1.103153	0.301449	0.993443	1.026791	0.500000	0.500000	0.331578	0.537155	0.725861	0.964023
max	3.000000	3.000000	2.251839	3.000000	0.500000	3.000000	3.000000	0.590372	3.000000	0.666667	3.000000	2.316329	3.000000

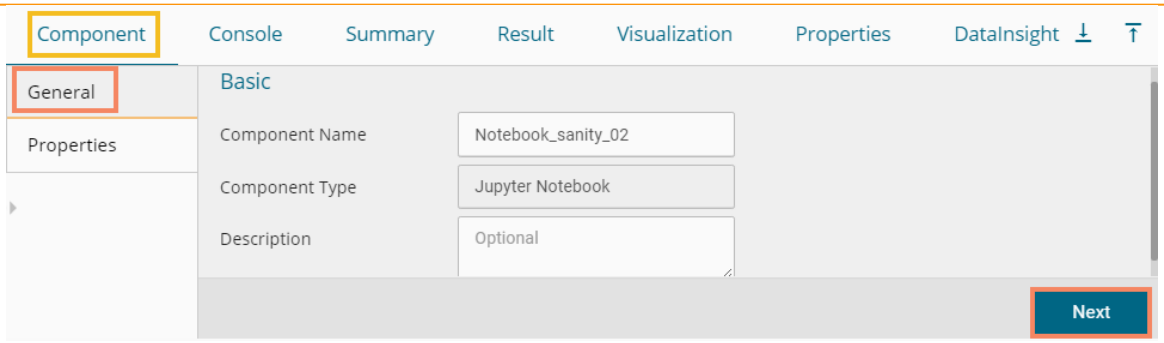
14.3. Jupyter Notebooks

The integrated Jupyter Notebook tree-node allows the users to create and share documents that contain live code, equations, visualizations, and narrative text. It can be used in numerical simulation, statistical modeling, data visualization, machine learning. The key motive is to introduce live coding inside the Data Science workbench and more efficiently use it as a component.

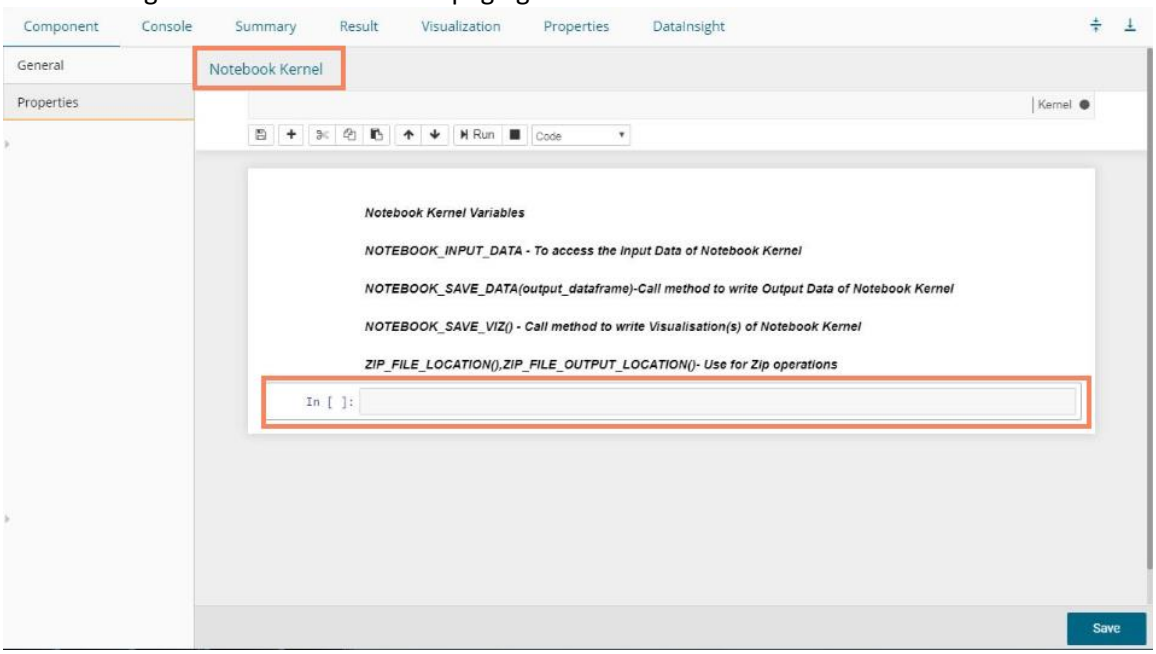
i) Upload a Data Source and run it.
ii) Connect the Jupyter Notebook component to it.



iii) Configure the General tab for the Jupyter Notebook component.
iv) Provide the Component name.
v) Click the '**Next**' option to load the Jupyter Notebook Kernel. This will trigger the Notebook Kernel in the backend and start it.

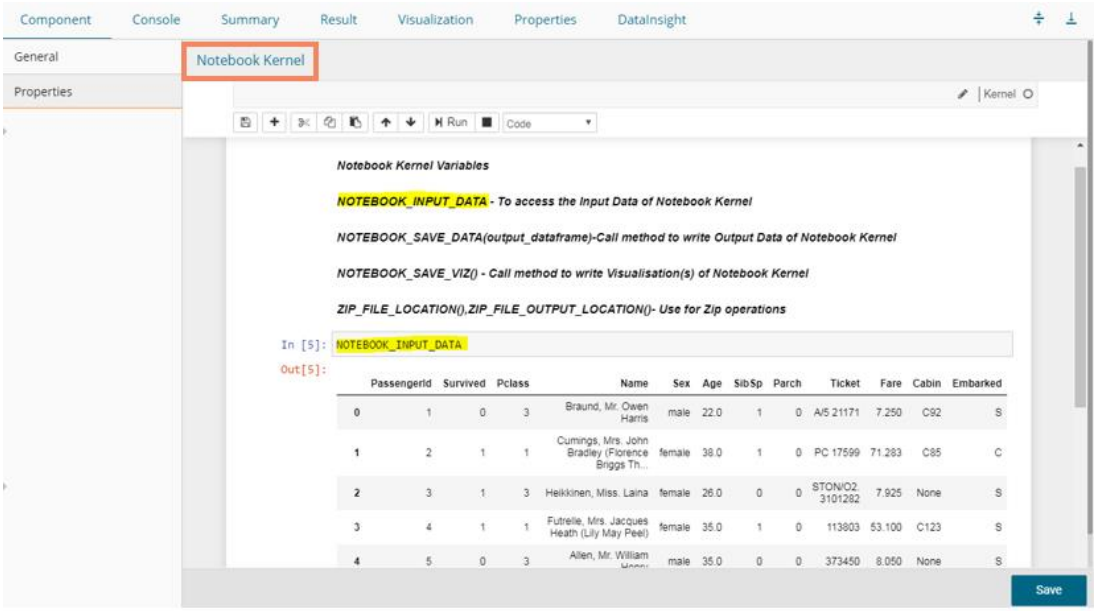


vi) After loading Notebook Kernel a new page gets listed in the footer tab as shown below:

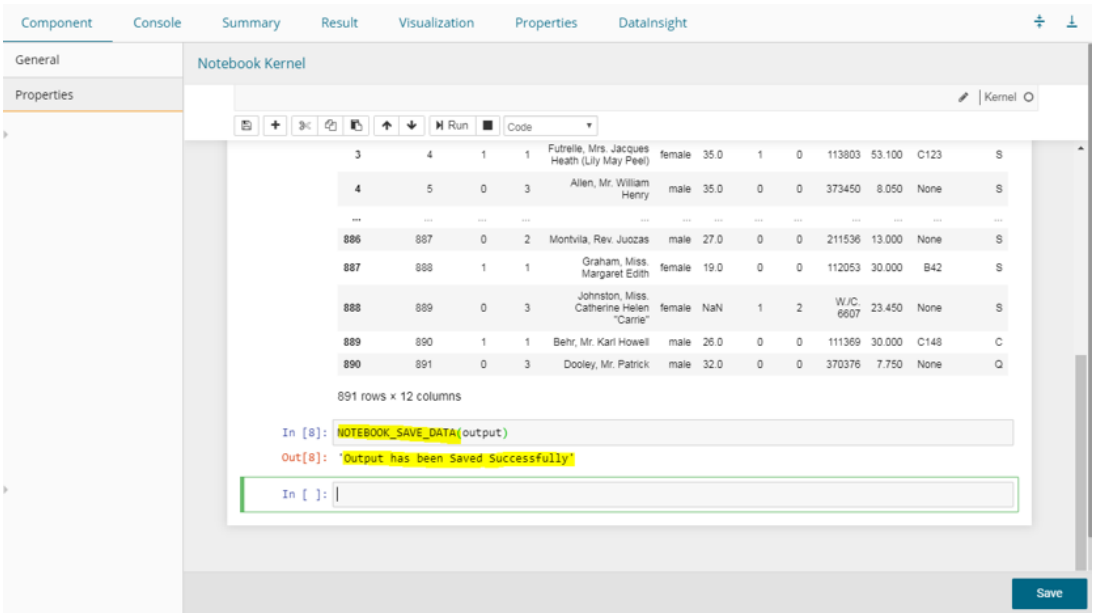


vii) Provide the script with proper Input, Output, and Save functions. The user must follow the instructions given for the Notebook Kernel Variables to move further.

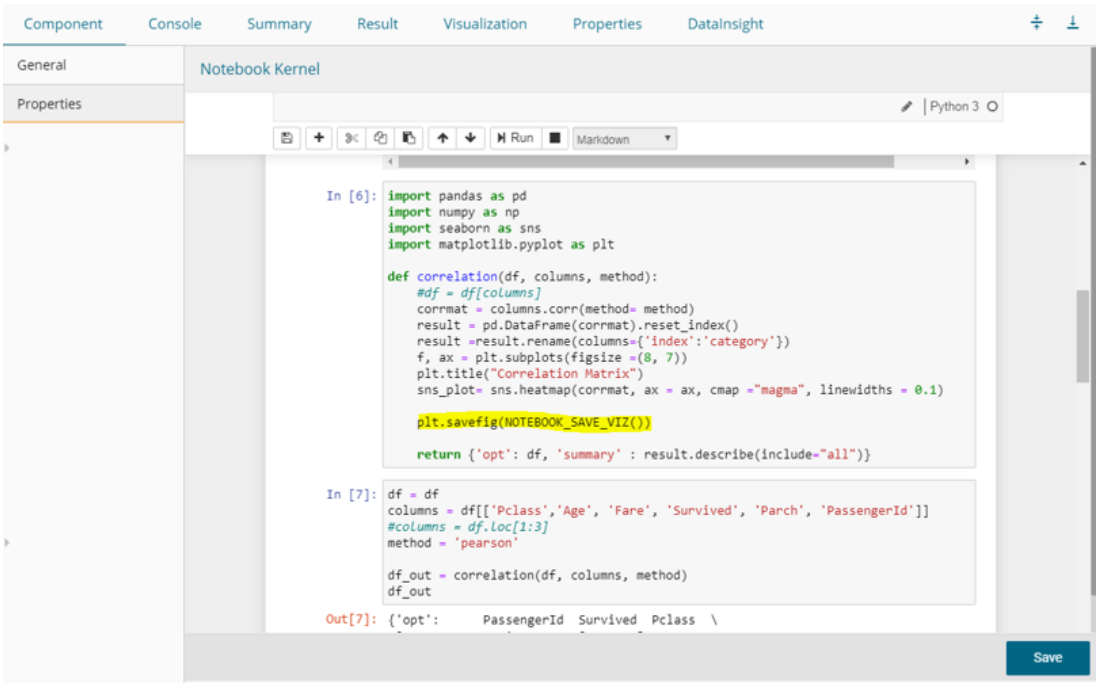
1. To load the input data using **NOTEBOOK_INPUT_DATA**



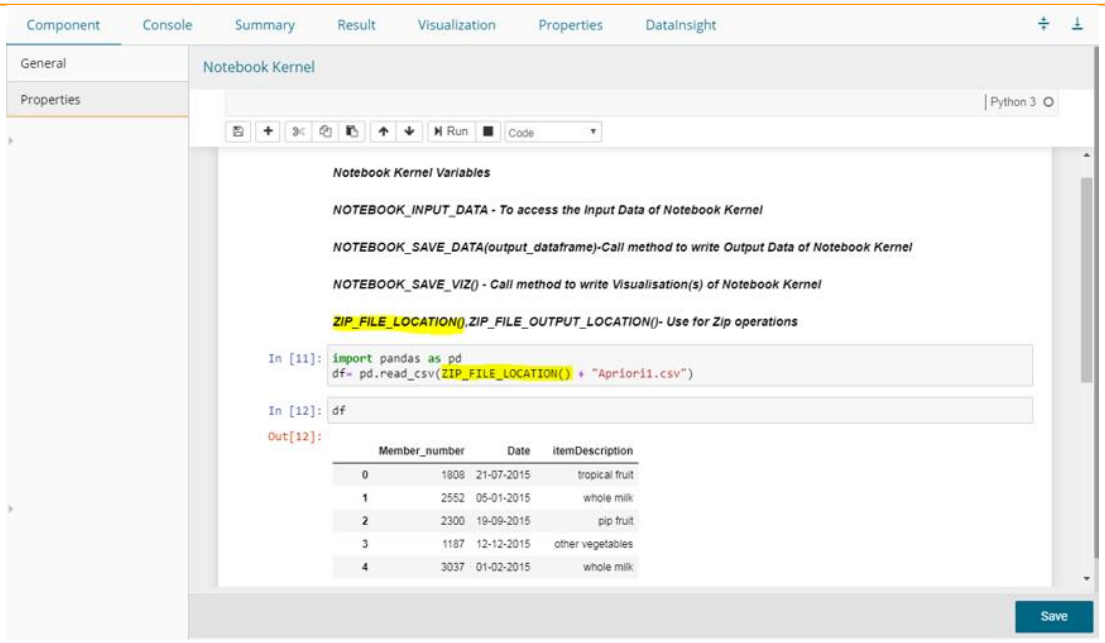
2. To save the output data use ***NOTEBOOK_SAVE_DATA(output_dataframe)***



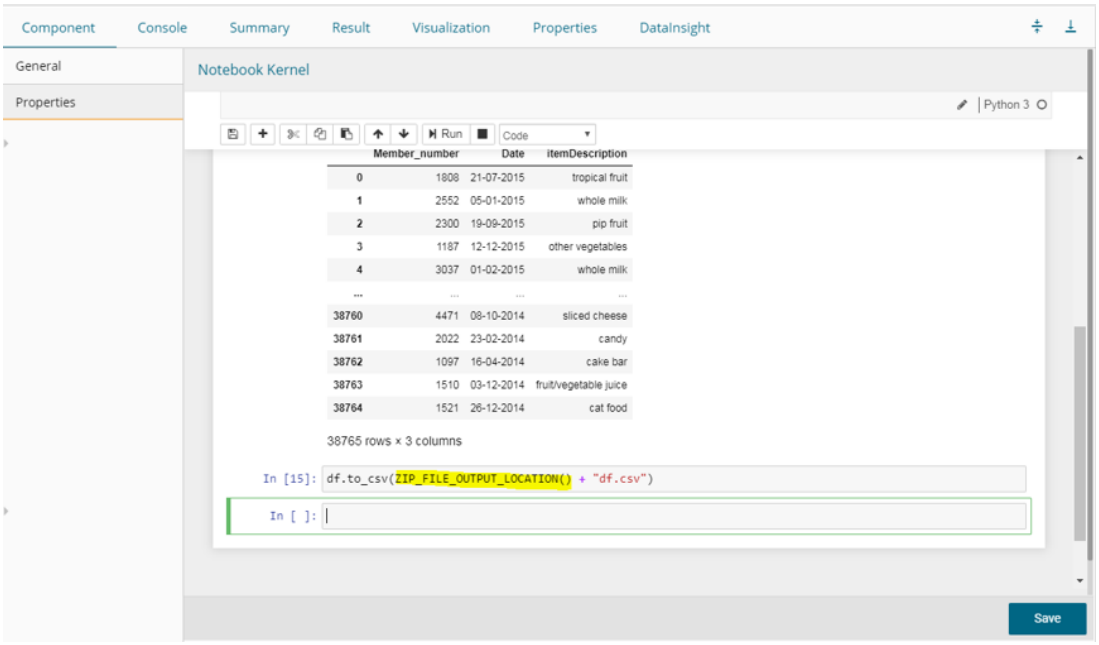
3. To call method to write Visualizations use ***NOTEBOOK_SAVE_VIZ()***



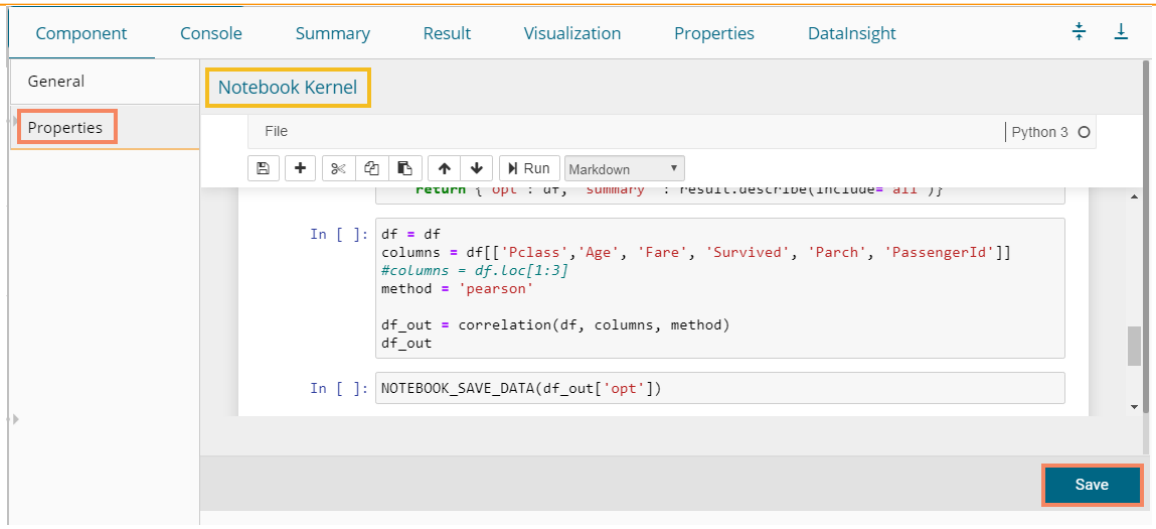
4. To load input data from a zip reader use ***ZIP_FILE_LOCATION()***



5. To save the output data to a zip file location use ***ZIP_FILE_OUTPUT_LOCATION()***

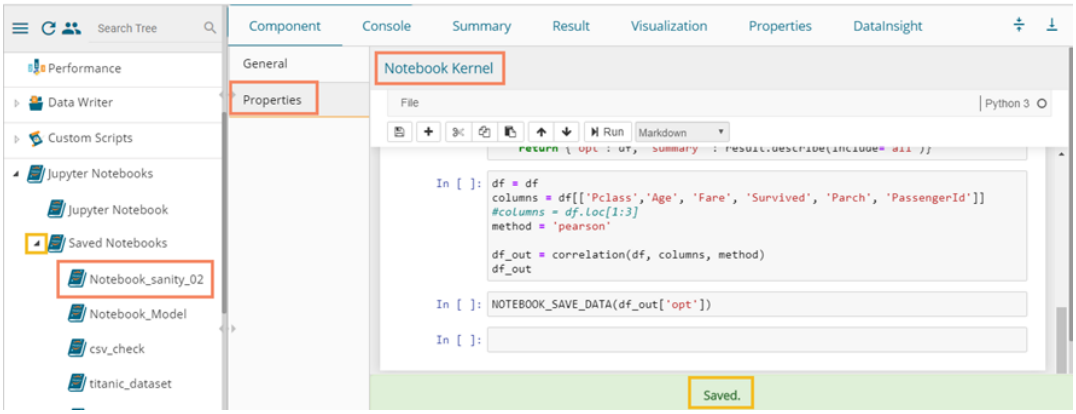


viii) Once you have saved the output of the Jupiter Notebook, click the '**Save**' option.



ix) A message appears to inform that the Jupyter Notebook has been saved.

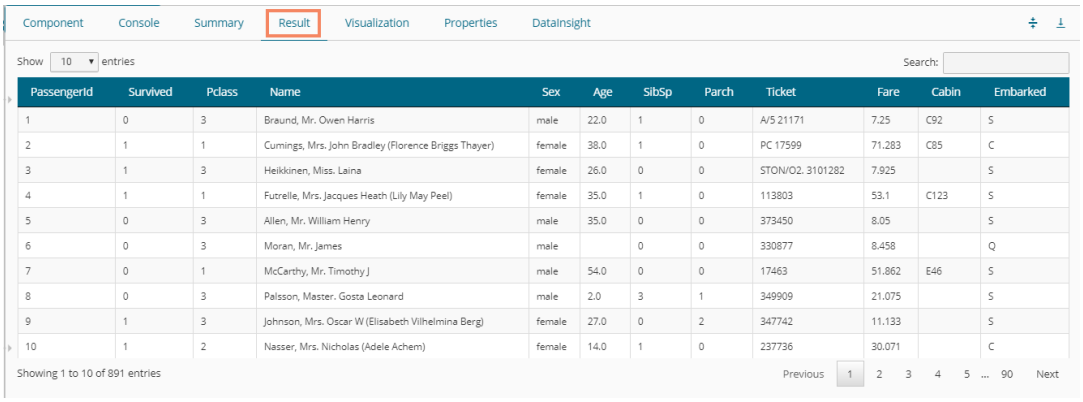
x) The Jupyter Notebook component gets added to the **Saved Notebooks** section.



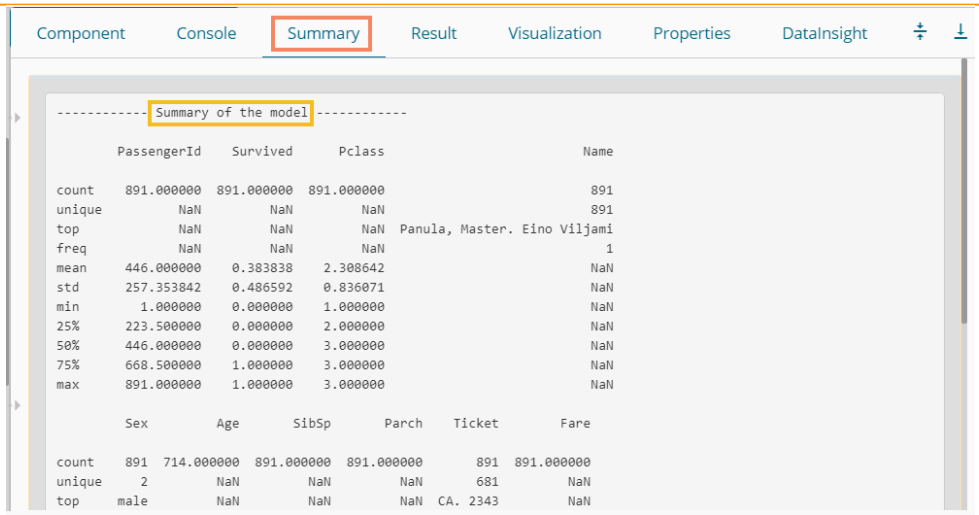
xi) Run the workflow.

xii) Once the Workflow runs successfully the user can see Summary, Result, Visualization, and DataInsight for the newly saved Jupyter Notebook component.

a. Click the '**Result**' tab to see the processed data.

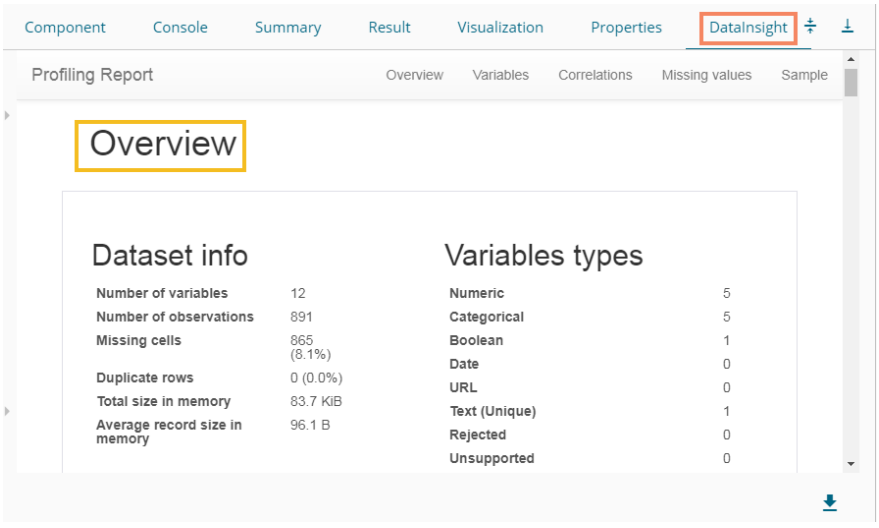


b. Click the '**Summary**' tab to see the model summary.



c. Click the '**Visualization**' tab to see the visual presentation of the data.

d. Click the '**DataInsight**' tab to see the data insights.

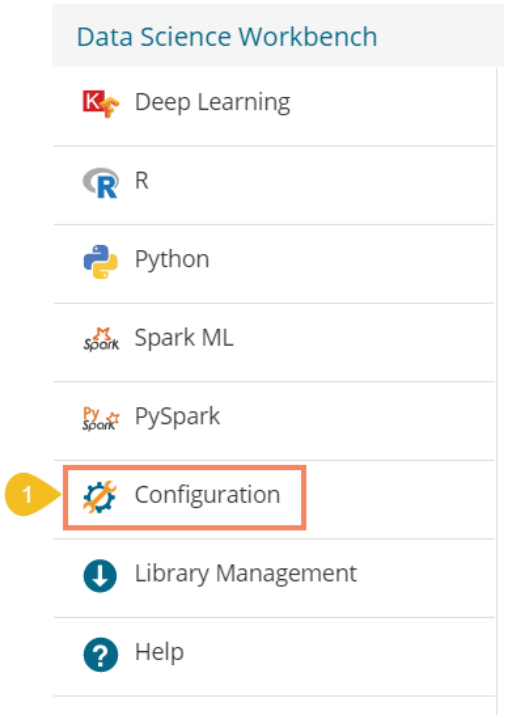


15. Configuration

The user gets redirected to the Admin module containing the server configuration option for the Data Science plugin.

15.1. Configuring Python Server

i) Click the '**Configuration**' option from the Data Science Workbench homepage.

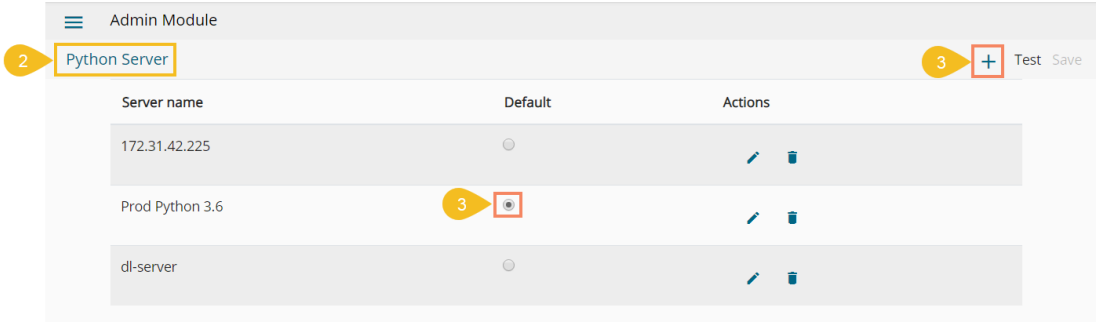


ii) By default, the Python Server details open under the **Admin Module**.

iii) The user can select another Python server from the available server list by selecting the radio button icon

or

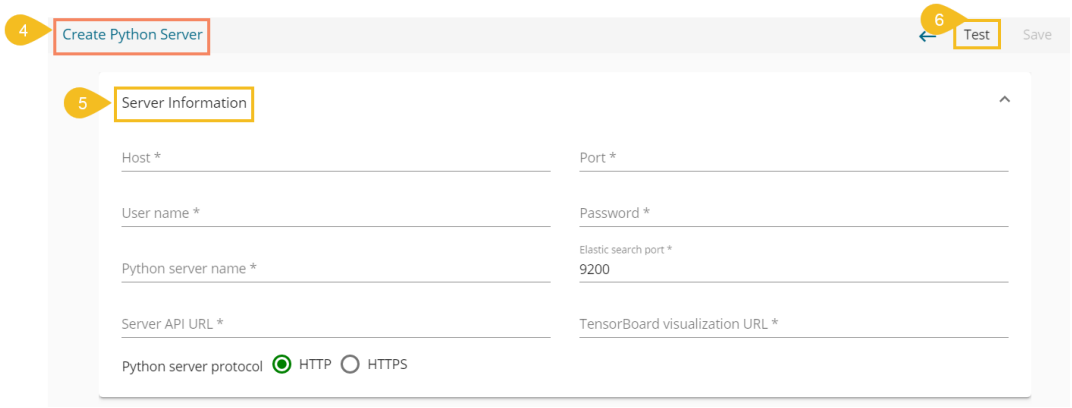
Click the '**Add new server**' + icon to configure a new Python server.



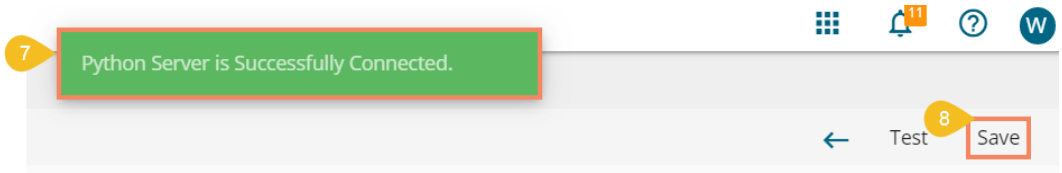
iv) The '**Create Python Server**' page opens by clicking the '**Add new server**' option.
v) Provide the following information:

i. Host: Host address of the Spark server
ii. Port: Spark server's port number
iii. Username: Enter a username to log in to the Spark server
iv. Password: Enter the password for the above username
v. Python Server Name: Provide Python Server Address
vi. Elastic Search Port: Provide the elastic search port number
vii. Server API URL: Provide the server API URL link
viii. Tensor Board Visualization URL: Provide the Tensor Board Visualization URL link
ix. Python Server Protocol: Select a protocol option by using the radio option out of **HTTP** and **HTTPS**

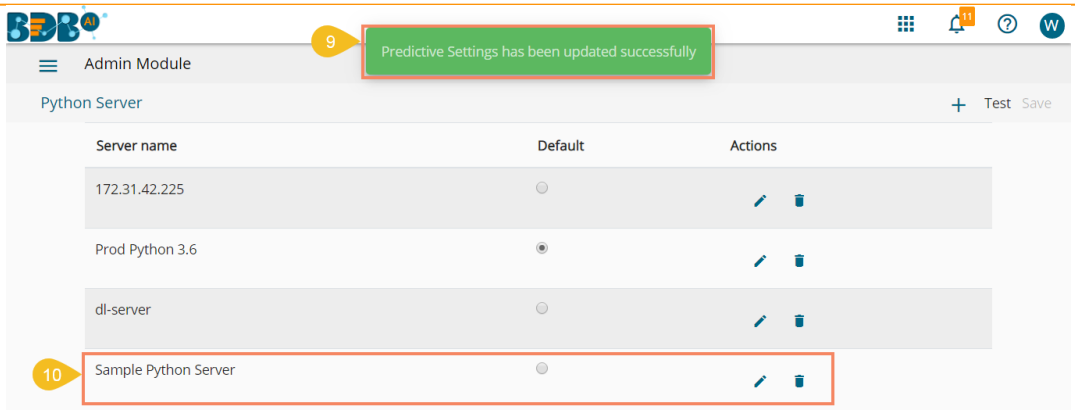
vi) Click the '**Test**' option to verify the connection.



vii) A success message appears to assure about the Python Server connection.
viii) Click the enabled '**Save**' option to save the verified Python server information.



ix) A success message appears to ensure that the predictive settings got updated.
x) The newly configured Python Server gets added to the '**Python Server**' window.



Note:

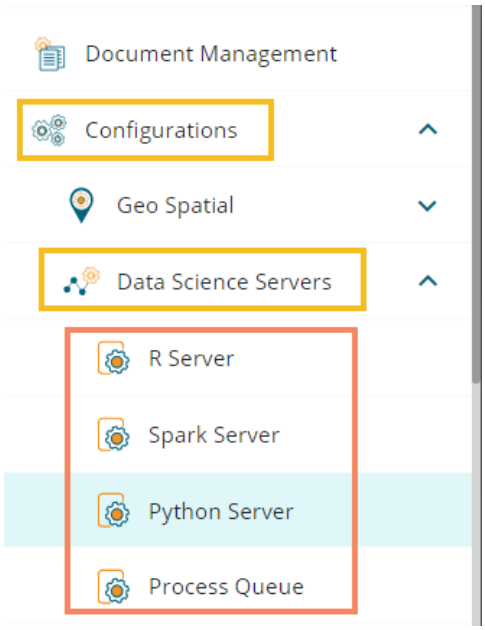
a. Click the '**Edit**' icon to modify an existing python server configuration

b. Click the '**Delete**' icon to remove the selected Python server details from the list.

xi) To access the other Data Science Servers, the user can click on the '**Configurations**' option provided under the Admin Module.

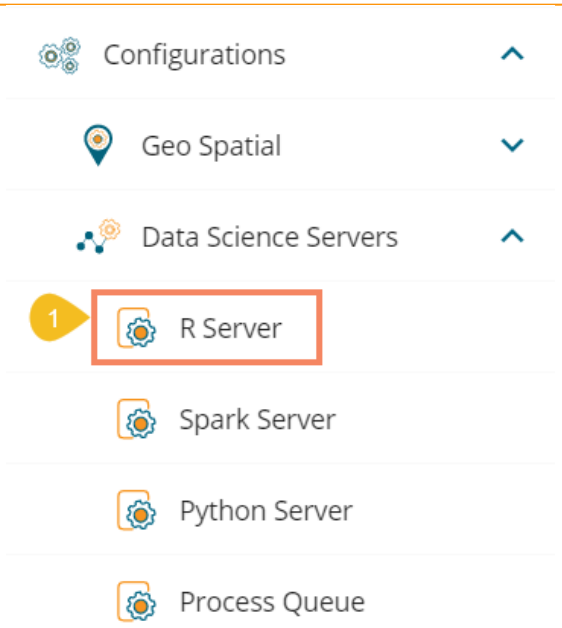
xii) Choose the '**Data Science Servers**' option.

xiii) All the available options appear as displayed below:



15.2. Configuring R Server

i) Select the **R Server** option from the Data Science server list.

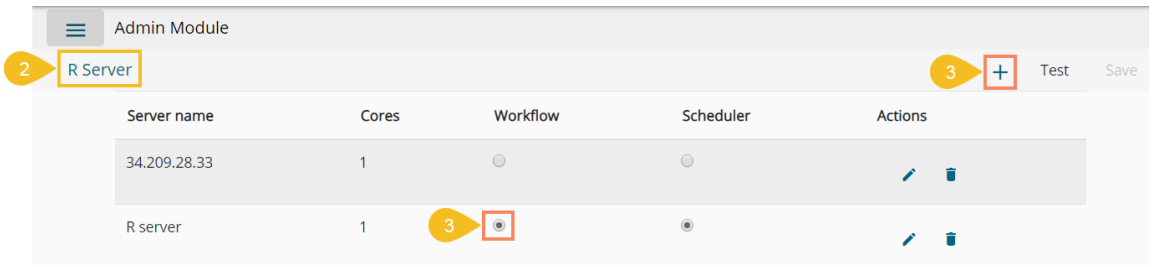


ii) The R Server page opens.

iii) The user can select another R server from the available server list by selecting the radio button icon

or

Click the '**Add new server**' + icon to configure a new R server.



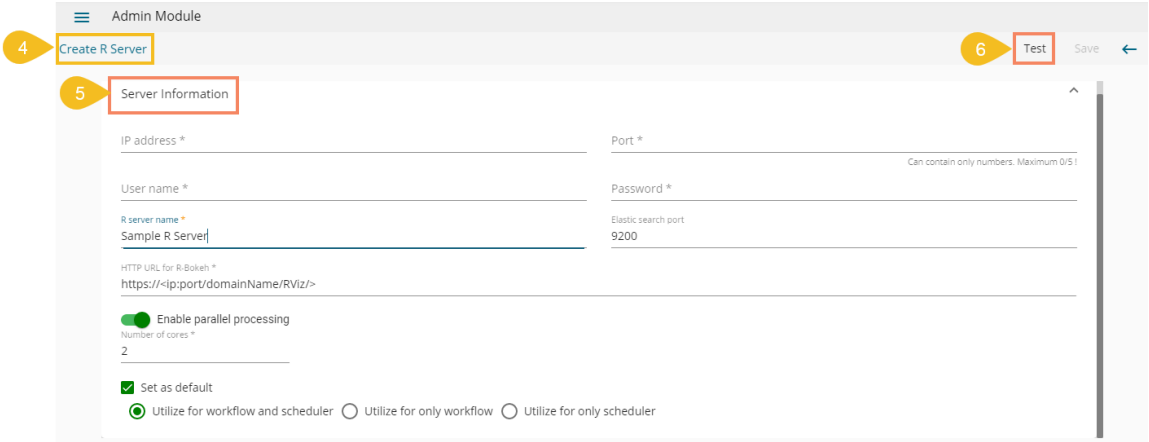
iv) The '**Create R Server**' page opens by clicking the '**Add new server**' option.

v) Provide the following information to configure a new R server:

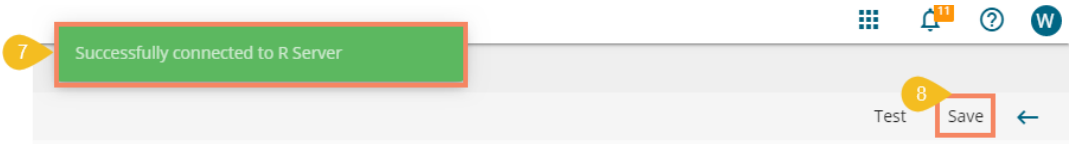
i. IP Address: IP address of the R-server
ii. Port: R-Server's port number
iii. Username: Enter a username to log in to the R- server
iv. Password: Enter the password for the above username
v. R Server Name: Provide the R- Server address
vi. Provide HTTP URL for R-Bokeh: Provide R Visualization URL
vii. Elastic Search Port: Provide an elastic search port number
viii. R Visualization URL: Provide HTTP URL for R-Bokeh
ix. Enable Parallel Processing: Avail this option by using the enable/disable button
    1. By enabling the Parallel Processing, it asks to configure 'Number of Cores'
x. Set as Default: Select this option by using a checkmark in the box
xi. The user gets further options for the Parallel enabled Processing:
    1. Utilize for Workflow and Scheduler

2. Utilize for the only Workflow
3. Utilize for the only Scheduler

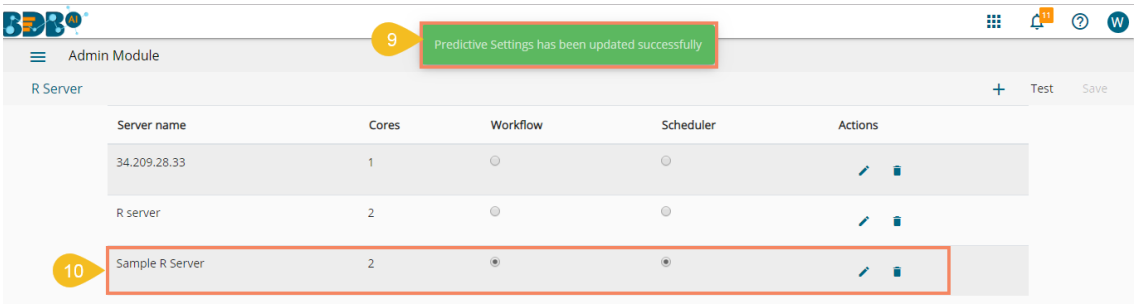
vi) Click the '**Test**' option to verify the R-Server connection.



vii) A success message appears to assure about the R Server connection.
viii) Click the enabled '**Save**' option to save the verified R server information.



ix) A success message appears to ensure that the predictive settings got updated.
x) The newly configured R Server gets added to the '**R Server**' window.

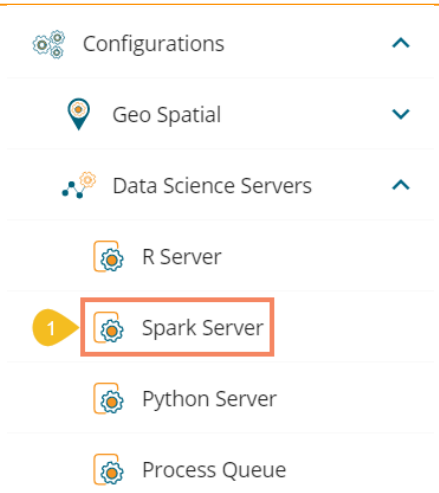


Note:

a. Click the '**Edit**' icon to modify an existing R server configuration
b. Click the '**Delete**' icon to remove the selected R server details from the list.

15.3. Configuring Spark Server

i) Select the **Spark Server** option from the Data Science server list.

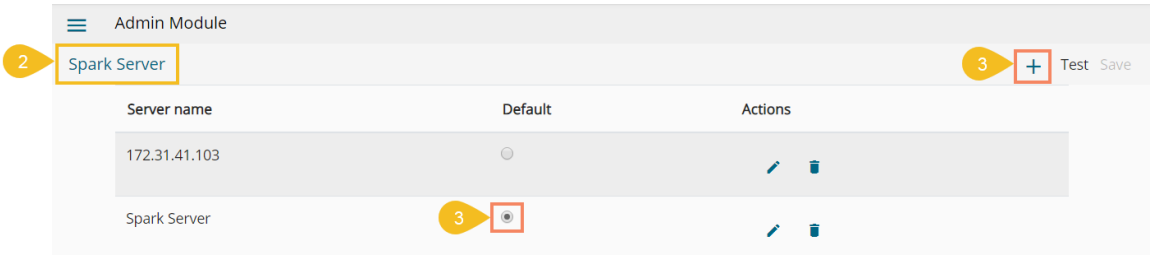


ii) The Spark Server page opens.

iii) The user can select another Spark server from the available server list by selecting the radio button icon.

or

Click the '**Add new server**' icon to configure a new Spark server.

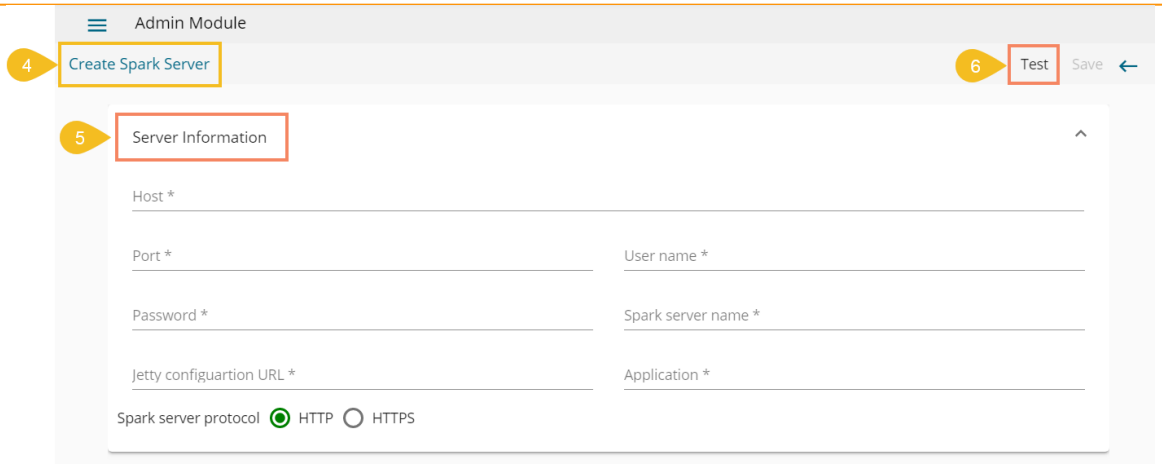


iv) The '**Create Spark Server**' page opens by clicking the '**Add new server**' option.

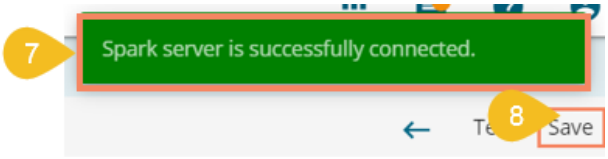
v) Provide the following information to configure a new Spark server:

i. Host: Host address of the Spark server
ii. Port: Spark server's port number
iii. Username: Enter a username to log in to the Spark server
iv. Password: Enter the password for the above username
v. Spark Server Name: Provide Spark Server Address
vi. Jetty Confirmation URL: Provide Jetty confirmation URL link
vii. Application: Provide the application name
viii. Spark Server Protocol: Select a protocol option by using the radio option

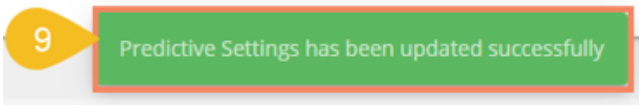
vi) Click the '**Test**' option to verify the Spark Server connection.



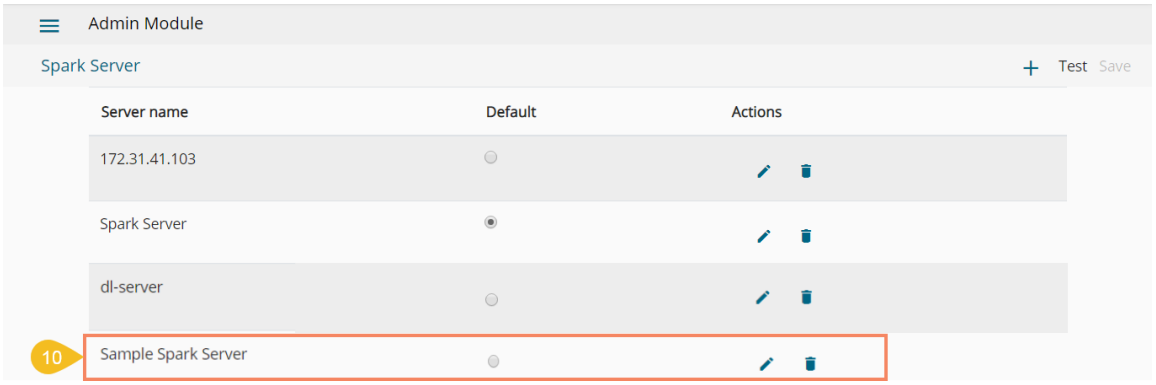
vii) A success message appears to assure about the Spark Server connection.
viii) Click the enabled '**Save**' option to save the verified Spark server information.



ix) A success message appears to ensure that the predictive settings got updated.



x) The newly configured Spark Server gets added to the '**Spark Server**' window.



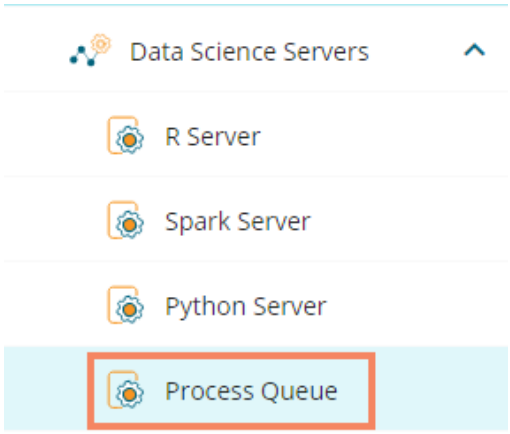
Note:

a. Click the '**Edit**' icon to modify an existing Spark server configuration
b. Click the '**Delete**' icon to remove the selected Spark server details from the list.

15.4. Configuring Process Queue

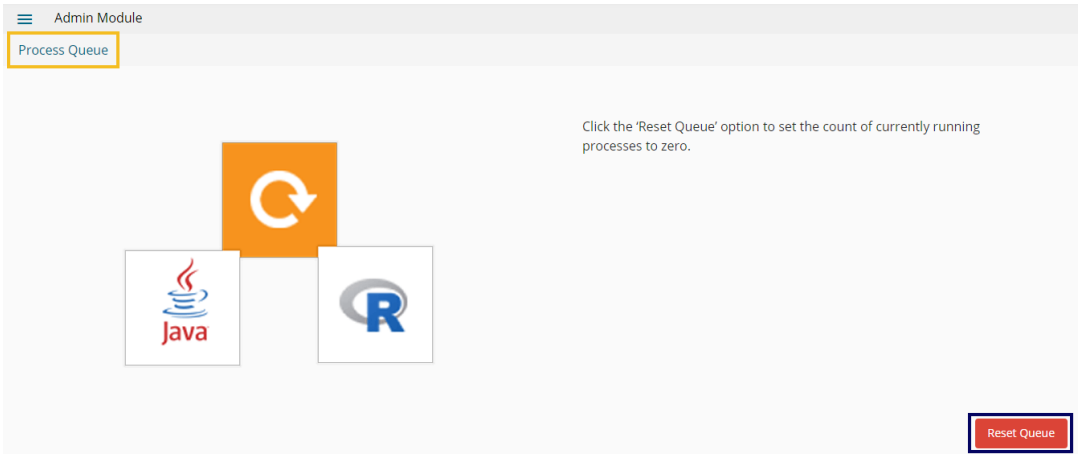
The user can reset the Predictive process queue through this Predictive Settings option.

i) Click the '**Process Queue'** option from the Data Science Servers configuration options.



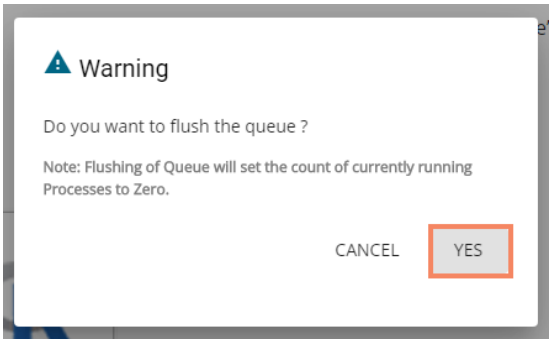
ii) The Process Queue page opens.

iii) Click the '**Reset Queue**' option.



iv) A warning message appears, asking whether the user wants to flush the queue.

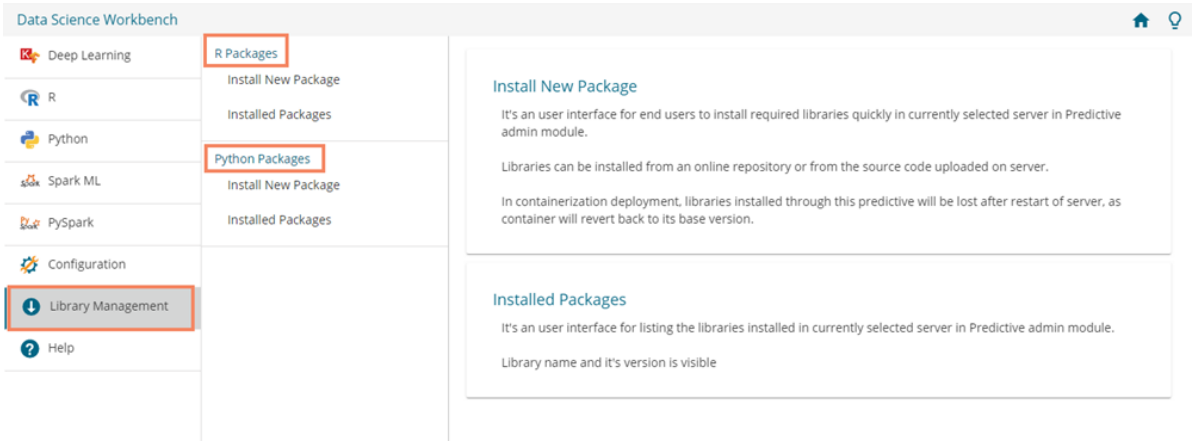
v) Click the '**YES**' option to set the count of currently running processes to Zero for the Data Science Workbench.



16. Library Management

The Library Management option facilitates the user to install R and Python libraries from an online repository or the source code uploaded on the server to the R and Python Data science servers.

Click the Library Management option from the Data Science Workbench. Details to open R and Python Packages get displayed.



i) R Packages

a) Navigate to the Library Management page.

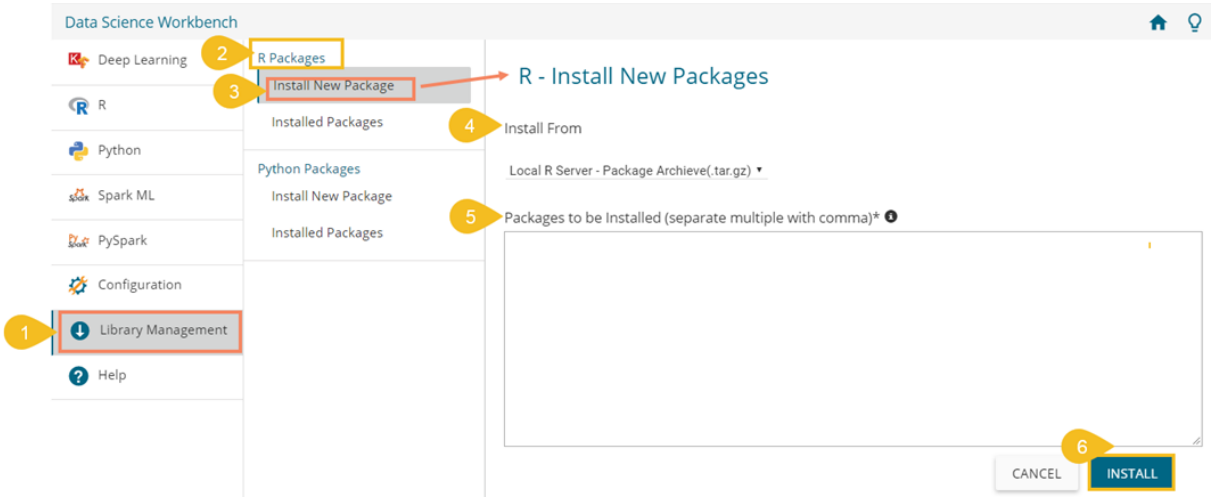
b) The R Packages option displays.

c) Click the '**Install New Package**' option to open the R-Install New Packages screen.

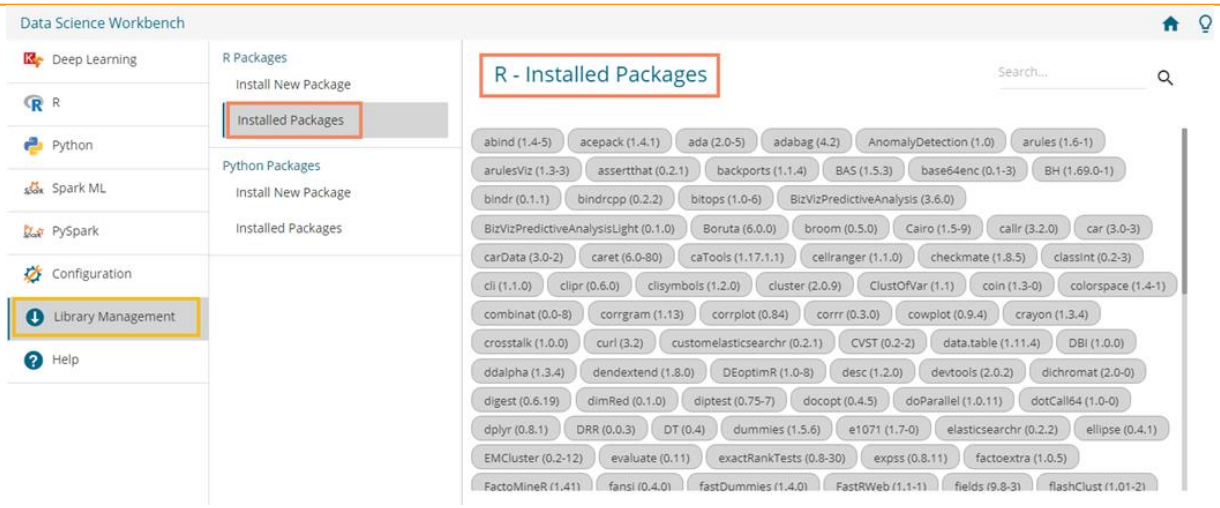
d) Select an option from where you want to install the package using the drop-down list.

e) Provide the package names in the given box. Use a comma to separate multiple packages.

f) Click the '**INSTALL**' option to install the packages.



g) Click the '**Installed Packages**' to display all the installed packages.



ii) Python Packages

a) Select the '**Install New Package**' option from the Python Packages.

b) The '**Python- Install New Packages**' fields open.

c) Select an option from where you want to install.

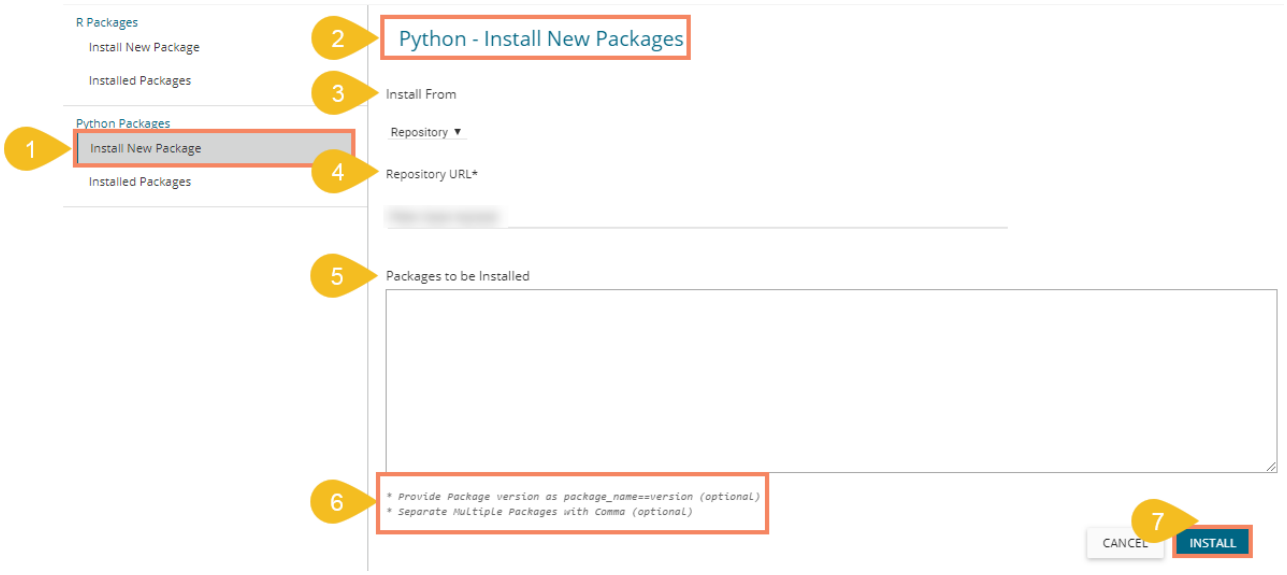
d) If the selected '**Install From**' option is '**Repository**,' it displays the Repository URL link.

e) Mention the packages you want to install in the given box.

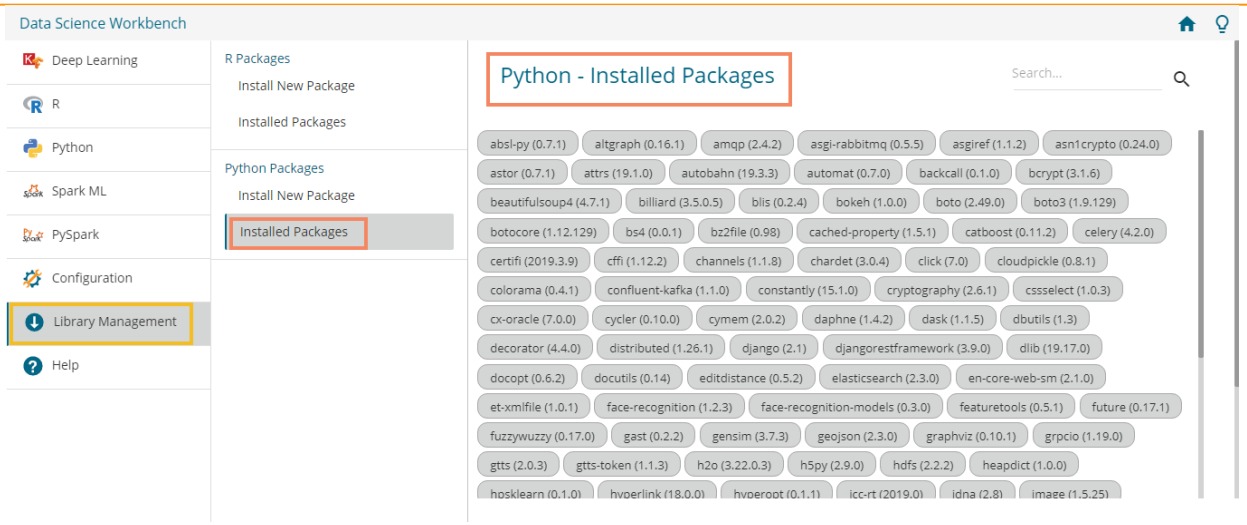
f) Follow the below-given rules:

1. Provide Package version as 'package name==version' (optional)
2. Separate Multiple Packages with Comma (optional)

g) Click the '**INSTALL**' option to install the new Python packages.



h) Click the '**Installed Packages**' option to display all the installed Python Packages.



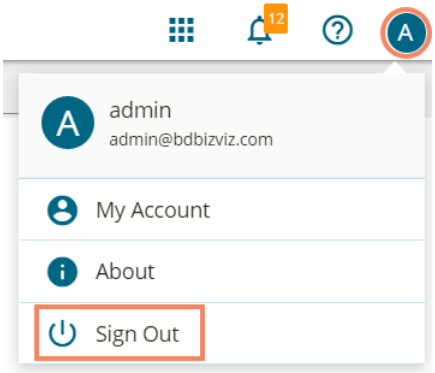
Note: The containerized deployment does not support the libraries installed through this option as the container reverts to its base version.

17. Signing Out

The following steps describe how to Sing-off from the BDB Platform.

i) Click the '**User Profile**' icon on the Platform homepage.

ii) Click the '**Sign Out**' option.

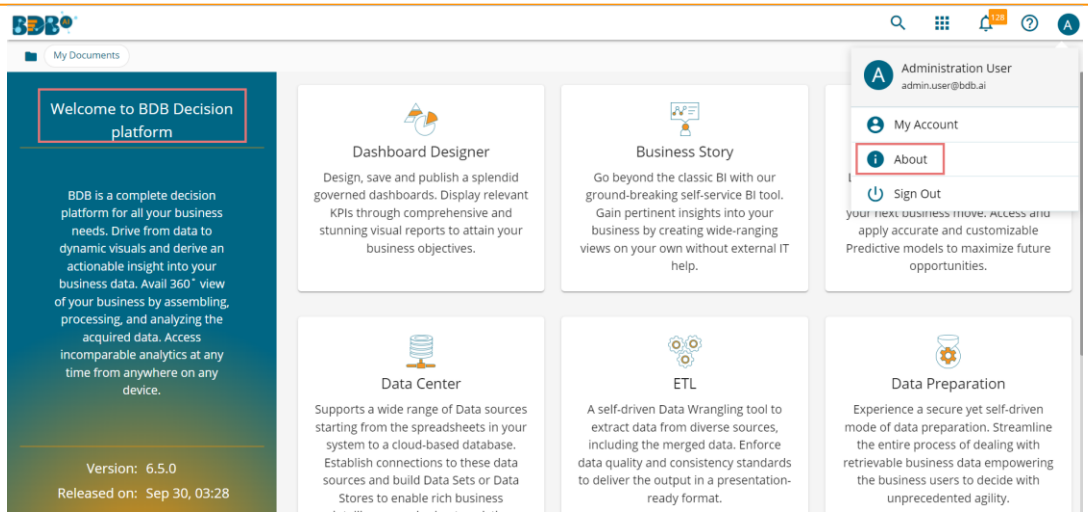


iii) The user successfully signs off from the **BDB Platform.**

Note:

a. By clicking the '**Sign Out**' option, the user gets back to the Sign-in page of the BDB platform.

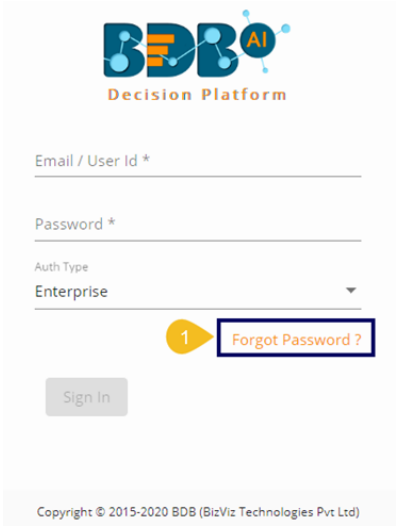
b. Click the '**About**' option to open the default homepage for the BDB Platform.



17.1. Forgot Password Option

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

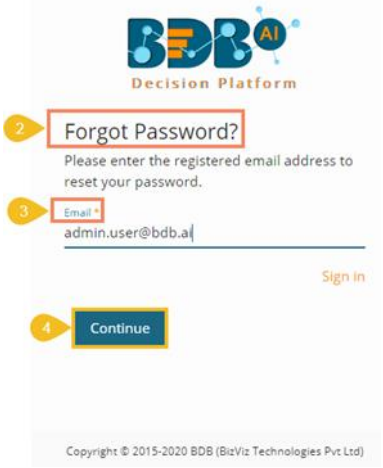
i) Click the '**Forgot Password?**' option from the Sign In page.



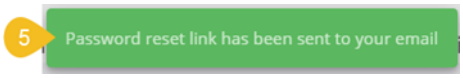
ii) The '**Forgot Password?**' page opens.

iii) Provide the email id that is registered with BDB to send the reset password link.

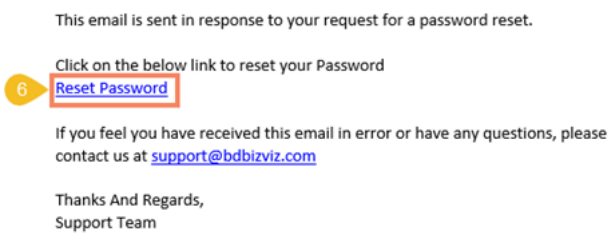
iv) Click the 'Continue' option.



v) The user may be redirected to select a space in case of multiple spaces under one server link(The user needs to select a space and click the '**Continue**' option once again). If a user does not have multiple spaces then, a message appears to notify the user that the password reset link (The users receive the reset link via their registered email.)

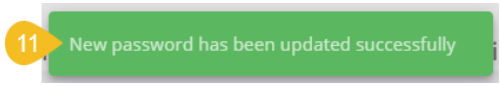


vi) Click the link from your registered email.



vii) The user gets redirected to the '**Reset Password**' page to set a new password.
viii) Set a new password.
ix) Confirm the newly set password.
x) Click the '**Continue**' option.

xi) The password for the selected BDB account gets reset and a message appears to inform the user.



Note: The user gets redirected back to the Sign In page after successfully resetting the password.

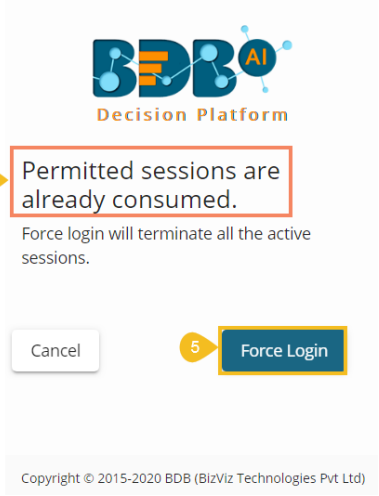
17.2. Force Login

The '**Force Login**' functionality has been introduced to control the number of active sessions up to three. The users can access only 3 sessions at a time when they try to access the 4th session, a warning message displays

to inform that the user has consumed the permitted sessions and, a click on the '**Force Login**' would kill all those active sessions.

i) Navigate to the BDB Platform Login page.
ii) Enter the valid credentials to log in.
iii) Click the '**Sign In**' option.

iv) The user gets the following message if the permitted active sessions (3 sessions at a time) are consumed.
v) Click the '**Force Login**' option.



vi) A warning message appears the currently active sessions get killed, and the user gets redirected to the BDB Platform Sign In page.
vii) The user needs to provide valid credentials once again and click the '**Continue**' option to access the platform.