NewgenONE

Data Science Studio

User Guide

Version: 1.0

Disclaimer

This document contains information proprietary to Newgen Software Technologies Limited. User may not disclose or use any proprietary information or use any part of this document without written permission from Newgen Software Technologies Limited.

Newgen Software Technologies Limited makes no representations or warranties regarding any software or to the contents or use of this guide. It also specifically disclaims any express or implied warranties of merchantability, title, or fitness for any particular purpose. Even though Newgen Software Technologies Limited has tested the hardware and software and reviewed the documentation, it does not guarantee or imply that this document is error free or accurate regarding any particular specification. As a result, this product is sold as it is and user, the purchaser, is assuming the entire risk as to its quality and performance. Further, Newgen Software Technologies Limited reserves the right to revise this publication and make changes in its content without any obligation to notify any person, of such revisions or changes. Newgen Software Technologies Limited authorizes no Newgen agent, dealer or employee to make any modification, extension, or addition to the above statements.

Newgen Software Technologies Limited has attempted to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

Copyright © 2023 **Newgen Software Technologies Ltd.** All Rights Reserved.
No part of this publication may be reproduced and distributed without the prior permission of Newgen Software Technologies Ltd.

Newgen Software, Registered Office, New Delhi
E-44/13
Okhla Phase - II
New Delhi 110020
India
Phone: +91 1146 533 200
info@newgensoft.com

Contents

Preface **7**
Revision history 7
About this guide 7
Intended audience 7
Documentation feedback 7
Introduction **9**
Business use case **11**
Accessing the Data Science Studio **14**
Prerequisites 14
Signing in to Data Science Studio 14
Exploring Data Science Studio interface **15**
Designing model pipelines 16
Monitoring pipelines 18
More options 18
Managing projects and pipelines 20
Deploying and Undeploying the model 21
Orchestration 23
My workspace 23
Requesting a feature 25
Raising an issue 26
Productionisation 27
Publishing a model 27
Published model 29
Production for approval 30
Customizing notebook 31
Jupyter notebook 31
Uploading a file 33
Registering Pmml 34
Creating a pipeline **35**
Preparing the data 36
Connecting data 36
Reading data source 36
AutoFeature 37
Wasb 41
Snowflake 42
Salesforce 43
Mssql 44
MySql 45
Sinking data 46
ETL 48

Performing multiple data source operations ... 49
Performing single data source operations ... 50
Cast columns ... 50
Chunker ... 52
Concat column ... 52
Date difference ... 53
Delete duplicate rows ... 54
Date time field extract ... 55
Dummy column ... 56
Delete null rows ... 57
Drop columns ... 57
Document normalizer ... 57
ETL missing value ... 58
Expression ... 61
Column filter ... 62
Lemmatization ... 64
Mapping ... 64
MELT ... 65
NGram ... 66
Column rename ... 66
Remove outliers ... 67
Random sampling ... 68
Sort data frame ... 69
Slice string ... 71
String transformation ... 72
Stemming ... 73
Stop word ... 74
Time series missing value ... 75
Value map operation ... 76
Segmenting data ... 78
Train Test split ... 79
Exploring data ... 80
Data exploration ... 80
Checking data quality ... 88
Profiling data ... 91
Structuring data ... 91
Bucketizer ... 92
Clip outliers ... 93
Missing values ... 94
Normalizer ... 95
One Hot encoding ... 95
Quantile descretizer ... 96
String indexer ... 97
Scaler ... 97
Unstructured data ... 98
Dimensionality reduction ... 100

Handling imbalance data ... 102
Transforming data ... 103
Augmenting data ... 103
Group by function ... 104
Python notebook ... 106
Rule engine ... 106
Deriving variable ... 108
Developing a model ... 111
Deep learning ... 111
Multi-Layer perceptron ... 111
Graph analytics ... 114
Processing natural language ... 115
Allocating latent dirichlet ... 116
Word2Vec ... 117
Evaluator ... 118
Binomial classification evaluation ... 119
Clustering evaluation ... 120
Multinomial classification evaluation ... 121
Regression evaluation ... 122
Machine learning ... 123
Clustering ... 124
Collaborative filtering ... 127
Affinity calculation ... 128
Alternating least squares ... 130
Time series ... 132
Autoregressive integrated moving average ... 132
Autoregressive integrated moving average with eXogenous variables ... 134
Automating ARIMA ... 135
Mining frequent pattern ... 137
Survival analysis ... 139
Classification ... 140
Decision tree classifier ... 141
Logistic regression classifier ... 144
Stacking classifier ... 146
Ensemble classifier ... 147
Gradient boosting classifier ... 147
Naïve bayes classifier ... 149
Support vector machine ... 151
Random forest classifier ... 152
Regression ... 155
Decision tree regressor ... 155
Generalized linear regressor ... 157
Stacking regressor ... 159
Ensemble regressor ... 160
Linear regressor ... 160

Gradient boosting regressor 163
Random forest regressor 164
Saving a pipeline 167
Running a pipeline 169
Model inference 170
Viewing model batch 170
Appendix 174
Data Science Studio constraints 174
Glossary 177

Preface

This chapter provides information about the purpose of this guide, details on the intended audience, and revision history for NewgenONE Data Science Studio.

Revision history

Revision date	Description
November 2023	Initial publication

About this guide

This guide explains how to create, publish, deploy and maintain data models using the NewgenONE Data Science Studio platform.

Intended audience

This guide is intended for data scientists and data analysts. The reader must have an understanding of data manipulation techniques.

Documentation feedback

To provide feedback or any improvement suggestions on technical documentation, write an email to docs.feedback@newgen.co.in.

To help capture your feedback effectively, share the following information in your email:

- Document name
- Version
- Chapter, topic, or section
- Feedback or suggestions

Introduction

NewgenONE Data Science Studio is a robust visual data science platform that accelerates data science project execution and enables you to quickly reach insights from raw data. The platform's advanced AI (Artificial Intelligence) capabilities increase productivity and encourage collaboration.

The NewgenONE Data Science Studio platform provides a variety of capabilities to meet data science requirements, including:

- **Data preparation** — Allows you to work with large amounts of data efficiently. It helps you blend, integrate, cleanse, and explore data on a massive scale. You can connect various data sources like relational databases and Azure Blob Storage using the built-in Data connectors. Once connected, you can use visual tools to prepare the data for modeling, which involves tasks like creating dummy variables and removing outliers, and more. NewgenONE Data Science Studio also supports integrating different types of data formats, such as relational, NoSQL, SQL, and various file formats.
  The platform offers built-in data sink nodes to save your data, model, or model outputs to preferred destinations like Elasticsearch or Blob Storage. The data processing capabilities are governed by , allowing you to read, integrate, and manipulate data without needing to store a local copy. This scalability makes NewgenONE Data Science Studio a powerful and flexible platform for handling large datasets.
- **Data visualization** — Makes data visualization easy and interactive. It helps you explore and cleanse data while analyzing large amounts of information. The platform offers various Visualization techniques, such as bubble charts, histograms, and scatter plots, to help you understand and analyze the data effectively.
  Additionally, NewgenONE Data Science Studio includes comprehensive Data cleaning operations that can be applied at both the column level and the entire dataset level. It enables you to perform data pre-processing tasks like predicting missing values and dropping columns with zero variance.
- **Model training** — With its comprehensive AI modeling design studio, it offers various data profiling options for both structured and unstructured data,

necessary for machine learning. The platform features an intuitive drag-and-drop interface that makes model development easy. You can drag and drop building blocks (nodes) onto the canvas and configure them to build a model pipeline.

NewgenONE Data Science Studio provides a rich set of modeling algorithms and techniques, including machine learning, deep learning, and graph modeling. The platform also includes a robust model evaluation capability that offers various performance metrics for multi-model experimentation and evaluation.

- **Model deployment and monitoring** — Simplifies the process of deploying models and monitoring their performance. It allows one-click deployment and integrated model performance monitoring. You can save the final models on local or Cloud infrastructure and serve them in batches or API modes.

  The platform supports deploying multiple models together, ensuring that the best-performing model is always served. You can also schedule and govern model retraining on new datasets. Additionally, NewgenONE Data Science Studio provides detailed insights into the behavior of all deployed models.
- **Effective automation** — Excels in automation, making many tasks easier for you. It automatically creates features from different datasets by analyzing the columns in the base datasets. The platform also allows you to add custom rules for creating new features.

  It supports machine learning-based feature selection and automatic feature validation capabilities, helping you choose the most relevant features for your models.

  NewgenONE Data Science Studio automatically selects the best-performing algorithms, configure parameters, and optimize for optimal model performance. Furthermore, it offers automatic model retraining and selection based on predefined frequencies.
- **Easy collaboration** — Promotes seamless collaboration among multiple stakeholders throughout the data science project lifecycle. Each user gets their own workspace to manage projects and perform various tasks, such as running, scheduling, monitoring, and publishing model pipelines. This collaborative approach ensures effective teamwork and smooth project management.

Related topic(s)

- Accessing Data Science Studio
- Exploring Data Science Studio interface

Business use case

Refer to the following use case to gain a better understanding of the NewgenONE Data Science Studio platform.

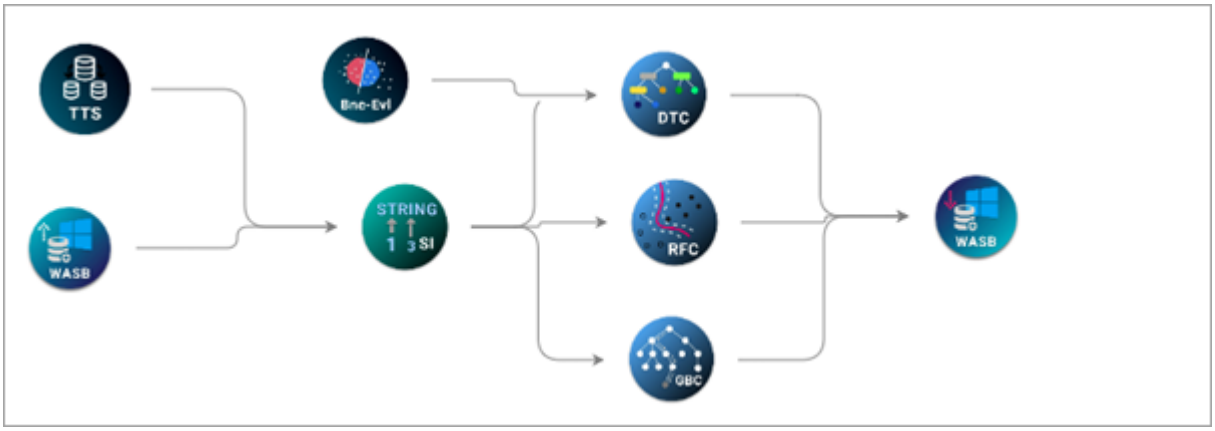
Problem Statement — To identify the dataset within the bank that indicates whether customers have ceased doing business with the bank or are still active.

NewgenONE Data Science Studio solution — The NewgenONE Data Science Studio Solution contains all the datasets with information about customers at a bank. The dataset also features a Boolean column named *Attrition_Flag*, indicating whether a customer has left the bank (churned) or not.

Within the dataset, there are two categories of information:

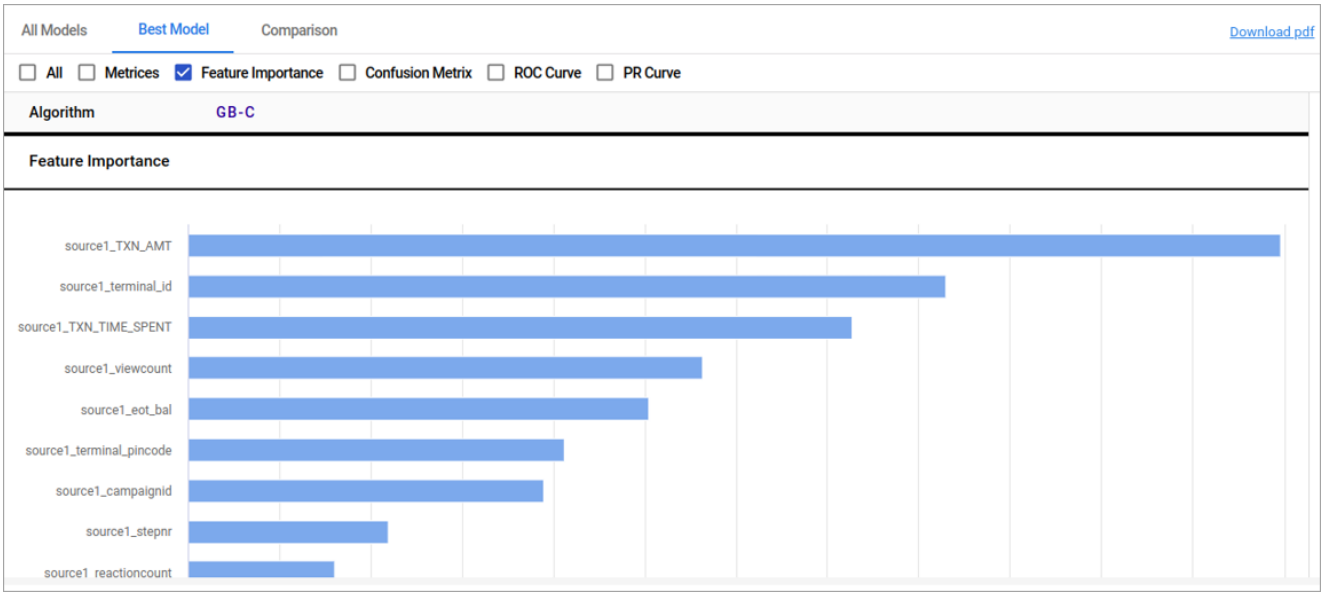
- **Personal Details** — This category includes information such as the customer's name, age, marital status, employment status, and more.
- **Transactional Details** — This category includes information about the customer's account balance with the bank.

Below is the sample pipeline to solve the above problem statement.



1. The dataset reads data from the Wasb source. For more information about datasources, refer to the Reading data source section.
2. The source dataset contains several string columns, such as Marital Status, Educational Category, Income Status, and more, which require a string indexer to obtain the numeric indexes of the values in these columns.

3. The train-test split node splits the original dataset into train and test datasets. The string indexer is applied separately to the training and testing datasets.
4. For this classification problem, there are three classification algorithms have been used in the pipeline:
    a. Decision Tree Classifier (DTC)
    b. Random Forest Classifier (RFC)
    c. Gradient Boosting Tree classifier (GBC)
5. The BNC-Evl classification evaluator node is connected to the DTC and can link to any algorithm node in the pipeline. It assesses the three models generated by these algorithms and selects the best one according to its evaluation settings.
6. In this case, it saves the final trained model to the sink location, which is Blob Storage. For more information, refer to Sinking data section.

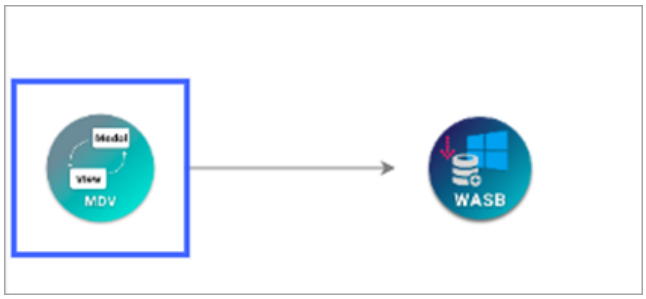


The chart illustrates the results of the pipeline, revealing crucial features (variables) related to the churn prediction problem within the provided dataset.

- In the depicted chart, the Gradient Boosting Classification (GBC) algorithm showcases the importance of features in the following order:
    - TXN_AMT— Displays the highest influence in determining churn prediction.
    - terminal_id — Displays the second-highest influence on determining churn prediction.
    - TXN_TIME_SPENT — Displays third in terms of influence on determining churn prediction.

- The chart also presents various metrics, including:
  - Confusion Matrix
  - ROC Curve
  - PR Curve
  - Accuracy
  - F1 Score
  - Precision
  - Recall, and more
- Furthermore, it indicates the utilization of three algorithms in the pipeline, with the GBC algorithm emerging as the most effective model based on the available dataset.

7. In the above pipeline, the trained model can be served on new data. For more information, refer to the Viewing model batch section.



8. The data contains no missing values, outliers, or other characteristics that requires additional data preparation steps beyond the String Indexer node.

> It is not mandatory to use DTC, RFC, and GBC classifiers only. You can use any other algorithm(s) described under classification for the same problem.

Accessing the Data Science Studio

The NewgenONE Data Science Studio is accessible online through web browser. This chapter includes the following topics:

- Prerequisites
- Signing in to Data Science Studio

Prerequisites

Following are the prerequisites to access the NewgenONE Data Science Studio:

- Valid NewgenONE Data Science Studio URL
- Registered Username and Password
- Supported browsers are:
  - Microsoft Edge 113 and above
  - Google Chrome 113 and above
  - Mozilla Firefox 113 and above

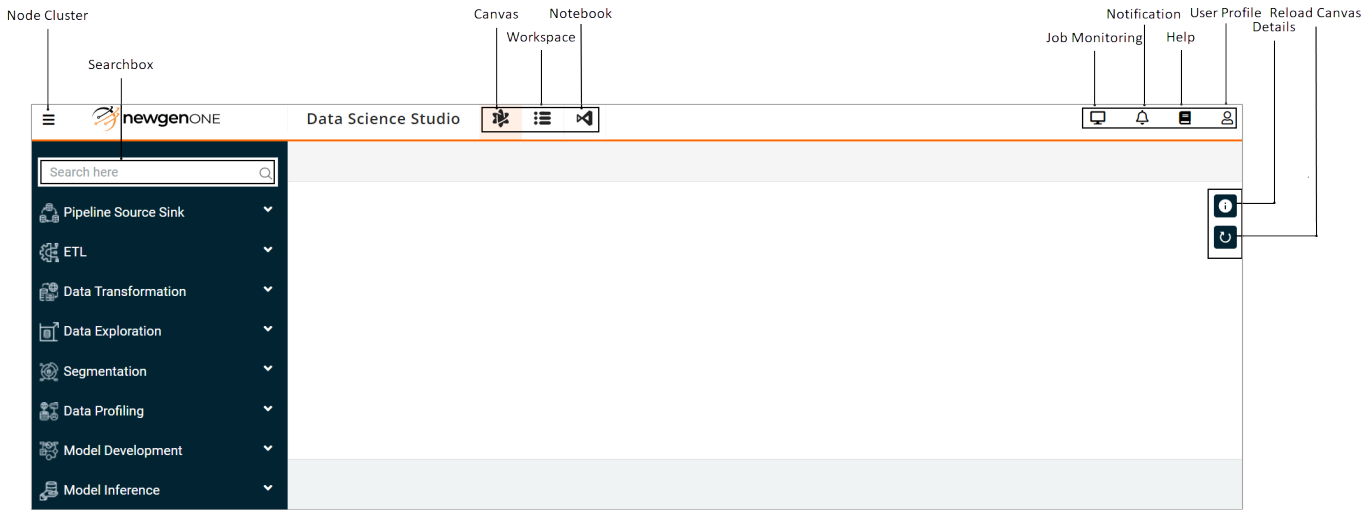
Signing in to Data Science Studio

To sign in to NewgenONE Data Science Studio, perform the following steps:

1. Open a web browser.
2. Enter the NewgenONE Data Science Studio URL in your browser. The Sign in dialog appears.
3. Enter your registered email ID and then click **Next**.
4. Enter your password and then click **Sign in**. On successful sign in, the landing page appears.

Exploring Data Science Studio interface

The landing page consists of the following menu options:

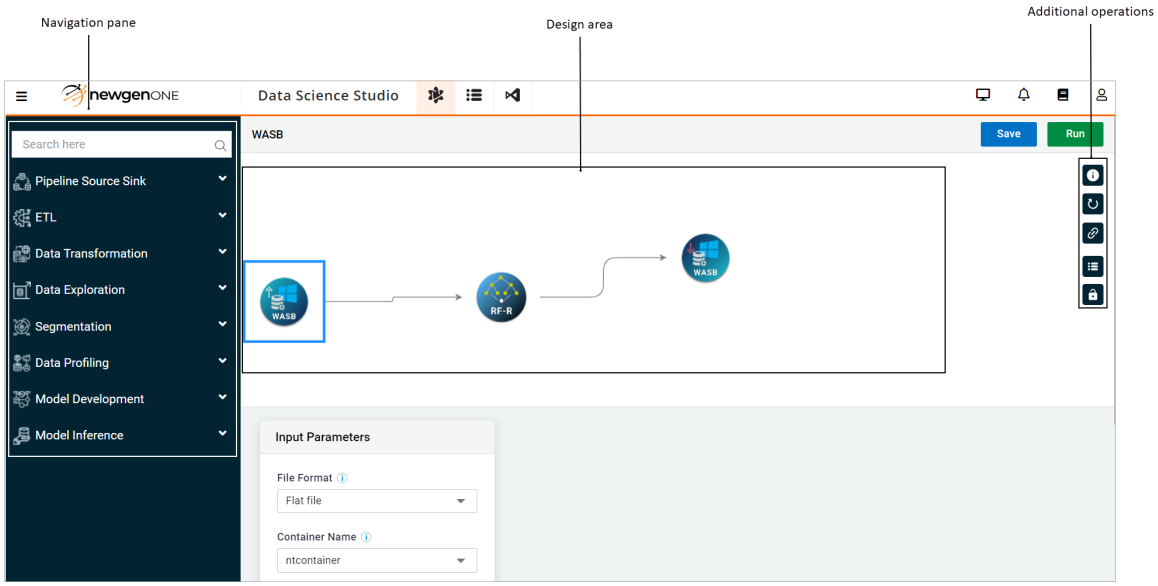


Options	Description
Node Cluster	Allows you to expand and minimize the node panel. For more information, refer to the Creating a pipeline section.
Searchbox	Allows you to search the nodes by its name.
Canvas	Allows you to open the canvas. The Canvas consists of the following options: • Details — Displays the pipeline details once you open a saved pipeline from the workspace. For example, project name, pipeline name, version and more. • Reload Canvas — Allows you to reload the canvas and reset the canvas to blank. For more information, refer to the Designing data or model pipelines section. The canvas becomes blank upon reloading. Ensure that you save the opened pipeline before reloading the canvas.

Options	Description
Workspace	Allows you to open or access the workspace. For more information, refer to the Managing projects and pipelines section.
Notebook	Allows you to browse the jupyter notebook. For more information, refer to the Jupyter Notebook section.
Job Monitoring	Allows you to monitor the pipeline executions. For more details, refer to the Designing data or model pipelines section.
Notification	Allows you to browse the notification of pending approval to publish a pipeline. For more information, refer to the Productionisation section. This option is visible to all users, but only administrators (approvers) selected by the signed-in users receive the notifications when publishing a pipeline.
Help	Allows you to open the document and training concepts to understand the NewgenONE Data Science Studio.
User Profile	Allows you to logout from the NewgenONE Data Science Studio platform.

Designing model pipelines

The Canvas allows you to design data or model pipelines required in a data science life cycle.



The Canvas consists of the following options:

Options	Description
Navigation pane	Contains the list of all the nodes clustered into different groups that allows you to perform data access from the source, basic to complex data transformation operations, and model experimentation as well as development actions. It also provides you a search box to find the required node.
Design area	Allows you to drag and drop the required nodes from the Node Cluster available in the navigation pane and connect those nodes to form a logical workflow to build a data or model pipeline.
Additional operations	Displays the parameter settings for the selected nodes on the canvas as follows: • Details — Displays the pipeline name, project name, and version. • Reload Canvas — Allows you to reset the canvas to blank. • Pipeline Operations — Allows you to enable or disable the particular settings and it vary for each node. • Workspace — Allows you to navigate to your workspace with selected project and pipeline. • Lock Pipeline — Allows you to lock the pipeline so any updates made to a node's parameters does not affect the succeeding nodes. The Pipelines Operations, Workspace, and Lock Pipeline operations appear if you click or select the any node on the canvas.

There are two options available on the canvas:

- Save Pipeline
- Run Pipeline

Monitoring pipelines

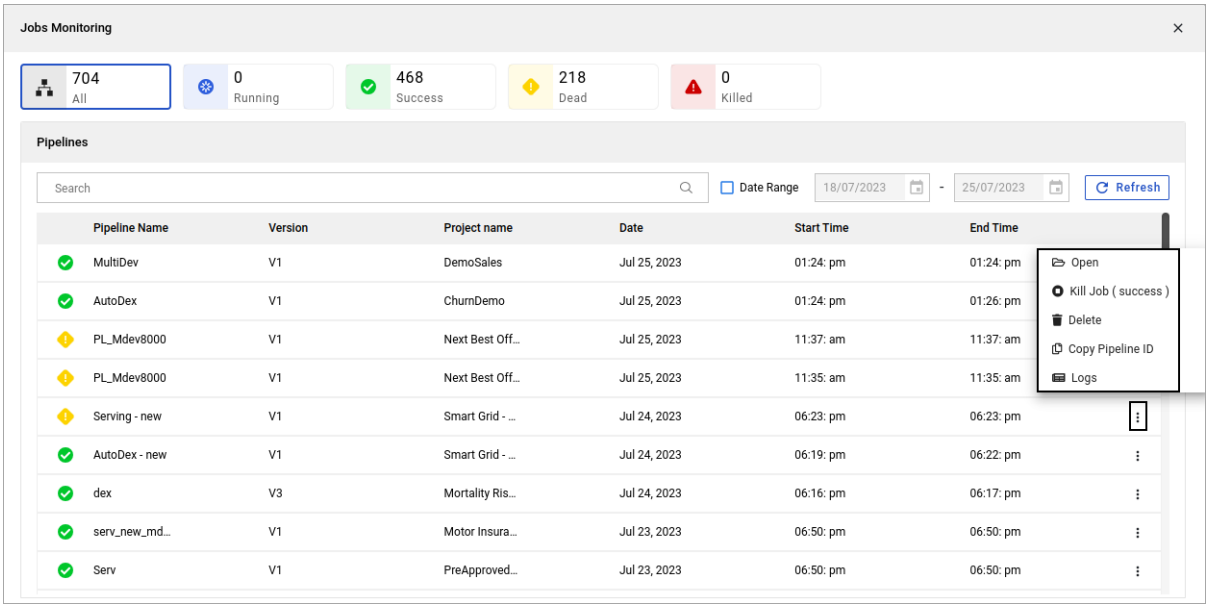
Monitoring pipelines allows you to check the status of the running pipeline.

To access the pipelines monitoring, open the canvas and click **Job Monitoring** icon. The Jobs Monitoring page appears with the following options:

Options	Description
Pipeline status	Indicates the pipeline status as follows: • All — Shows the count of all available pipelines. • Running — Shows the count of running pipelines. • Success — Shows the count of successfully executed pipelines. • Dead — Shows the count of pipelines that encountered errors during execution. • Killed — Shows the count of pipelines stopped by the system or the signed-in user.
Search box	Allows you to search any pipeline with its name.
Date Range	Allows you to view the pipelines by applying the date filter using the date range.
Refresh	Allows you to reload the pipelines data.
More options	Allows you perform multiple operations such as viewing logs and opening pipeline in canvas. For more information, refer to the More options section.

More options

This section describes the various operations that can be performed on the pipelines. To access the operations, go to **Canvas** and click **Job Monitoring**. The Job Monitoring dialog appears.

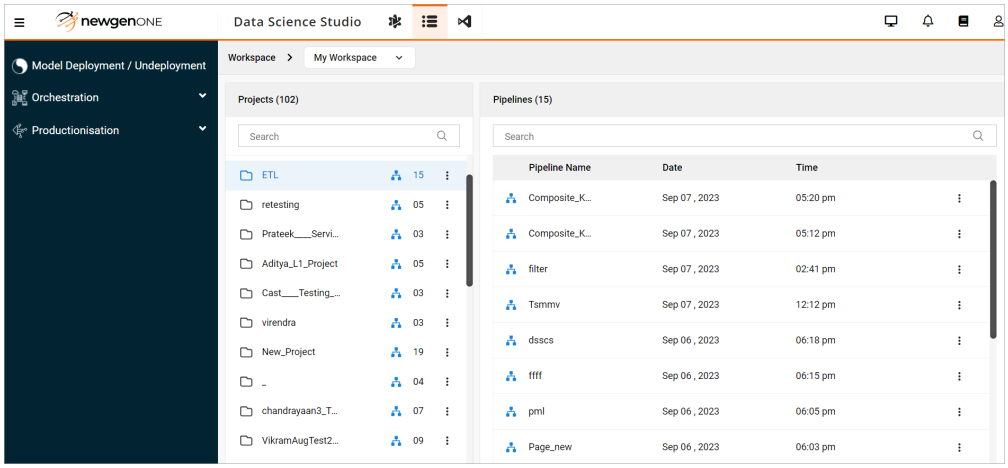


The options are as follows:

- **Open** — Allows you to open the pipeline in the canvas and modify the node information according to your requirement.
- **Kill Job (success)** — Allows you to stop the running pipelines.
- **Delete** — Allows you to delete the pipeline from the list available in the job monitoring.
- **Copy Pipeline ID** — Allows to copy the pipeline ID.
- **Logs** — Displays the logs that consist of the type of error along with its description, including the date and time of the error.

Managing projects and pipelines

The Workspace allows you to manage projects and pipelines.



To access the Worskpace, perform the following steps:

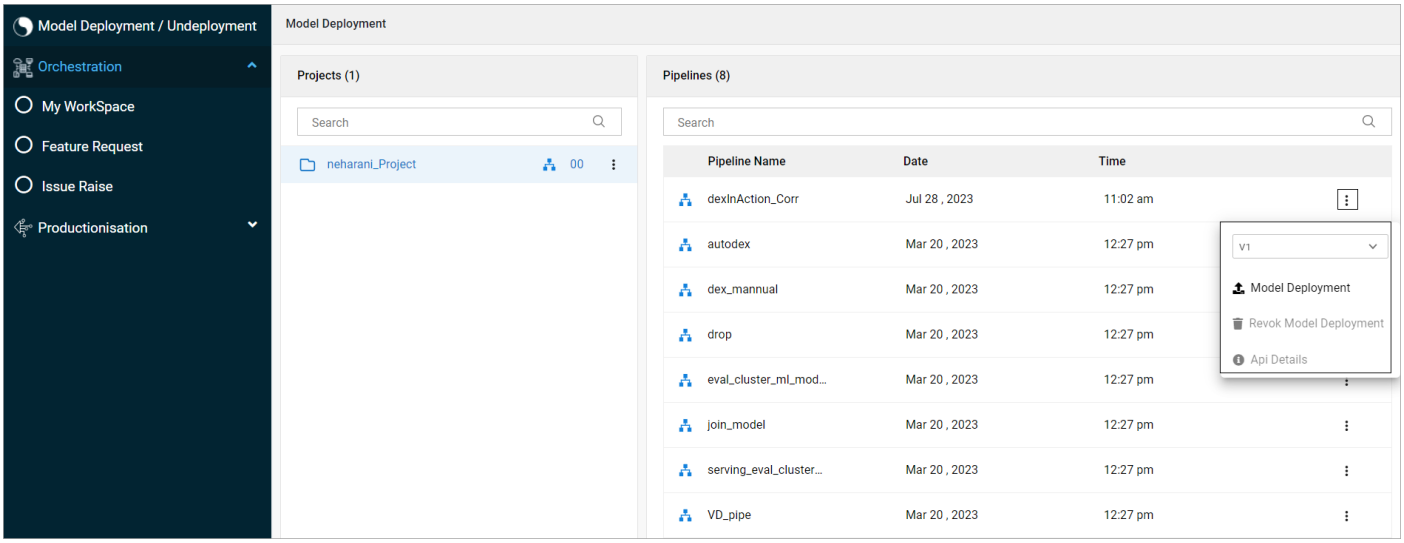
1. Open the NewgenONE Data Science Studio using the credentials. The landing page appears.
2. Click the Workspace icon given on upper-left pane. The Workspace page appears with the following options:

Options	Description
Workspace	Allows you to select the following workspace using the dropdown: • **My Workspace** — Comprises the list of projects created by signed-in users. My Workspace appears by default. • **Tenant** — Comprises the list of projects shared with all users in the application. • **Shared With Me** — Comprises the list of all pipelines shared with the signed-in user by other users of the application. It also comprises the following more options: ◦ import — Under the more options ⋮ icon, click the **import** to import the pipelines into projects in your workspace. ◦ preview — Under the more options ⋮ icon, click the **preview** to preview the pipelines.

Options	Description
Projects	Shows the projects present in the selected workspace and allows you to perform the following operations: • Search box — Allows you to search the project by its name. • Rename — Allows you to change the project name by clicking the more options icon given next to the project. • Delete — Allows you to delete the project by clicking the more options icon given next to the project.
Pipelines	Displays the pipelines present in the selected project. For more information, refer to the My Workspace section.
Navigation pane	Allows you to navigate to the following tabs on the left pane: • Model Deployment/ Undeployment • Orchestration • Productionisation

Deploying and Undeploying the model

Model Deployment and Undeployment provide a list of projects and pipelines present under the workspace of the signed-in user. It shows a list of only those pipelines with **PMML Save** set to **Yes** in the sink node of the pipeline.



To deploy or undeploy the pipelines, perform the following steps:

1. Go to **Workspace**. The My Workspace page appears.
2. Select the **Model Deployment/Undeployment** tab given on the left-pane. The Model Deployment page appears on the right pane.
   The model deployment page consists of the following:
   - Projects — Comprises the available project. Also, you can search the project with it's name using the searchbox.
   - Pipelines — Comprises the available pipelines on the selected project. Also, you can search the pipeline with it's name using the searchbox.
3. Click the more options ⋮ icon given against the pipelines. The dialog appears with the following options:
   - Model Deployment — Use this option to deploy the Predictive Model Markup Language (PMML) model.
   - Revoke Model Deployment — Use this option to undeploy the PMML model.
   - Api Details — Use this option to view the API details in the JSON and Curl format.

Model Deployment

To deploy the model, perform the following steps:

1. Click the **Model Deployment**. The dialog appears.
2. Click **Confirm**. The Model gets deployed.

Revoke Model Deployment

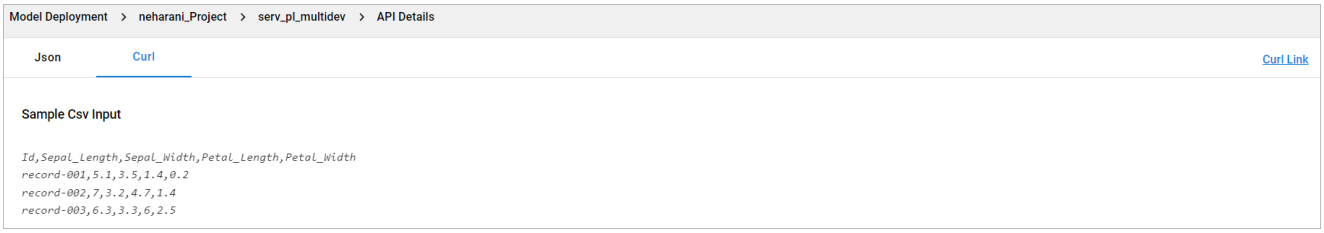
To undeploy or revoke the model, perform the following steps:

1. Click the **Revoke Model Deployment**. The dialog appears.
2. Click **Confirm**. The deployed model gets revoked successfully.

Api details

To view the API details, perform the following steps:

1. Click the **Api details**. The API Details page appears with Json and Curl tabs.



2. Go to **Json** to view the API request and response body in JSON format. Also, click **Json Link** given on right-pane to copy the Json link.
3. Go to **Curl** to view the Sample Csv Input in Curl format. Also, click **Curl Link** to copy the Curl Link.

Orchestration

This section consists of the following:

- My Workspace
- Feature Request
- Issue Raise

My workspace

My Workspace provides another navigation path to access the workspace of the signed-in user.

To access the Workspace and modify the pipelines, perform the following steps:

1. Go to **Orchestration** given on the navigation pane.
2. Click **My Workspace**. My Workspace page appears.
3. Under the Projects, select the project according to your requirement.

4. Under the Pipelines, click the **more options** ⋮ icon against any pipeline. The dialog appears with the following options:

Options	Description
View Result	Shows the results of the last executed pipeline. This option appears for the already executed pipeline.
Open	Opens the pipeline in the canvas. For more information, refer to the Navigating Canvas section.
Run	Runs the selected pipeline.
View Pipeline Info & Edit	Shows the pipeline details such as project name, pipeline name, created on, tags, and comments.
Copy	Copies the selected pipeline into another project or a new project within your workspace.
Delete	Deletes the pipeline from the project.
Publish Pipeline	Allows you to share the pipeline with a user or tenant (all the users of the application). This option appears when the pipeline is executed.
Json	Downloads the selected pipeline in a JSON format. To make changes to the JSON, download it from the submenu, make the required changes, and then upload the updated JSON back to the pipeline using the upload submenu. The changes get reflected in the pipeline once you upload the updated JSON. This option appears when the pipeline is executed successfully.
PMML Download	Downloads pipeline models in PMML format for flexible deployment. This option appears only when PMML flag is set to true in the sink node settings, and pipeline execution is successful.

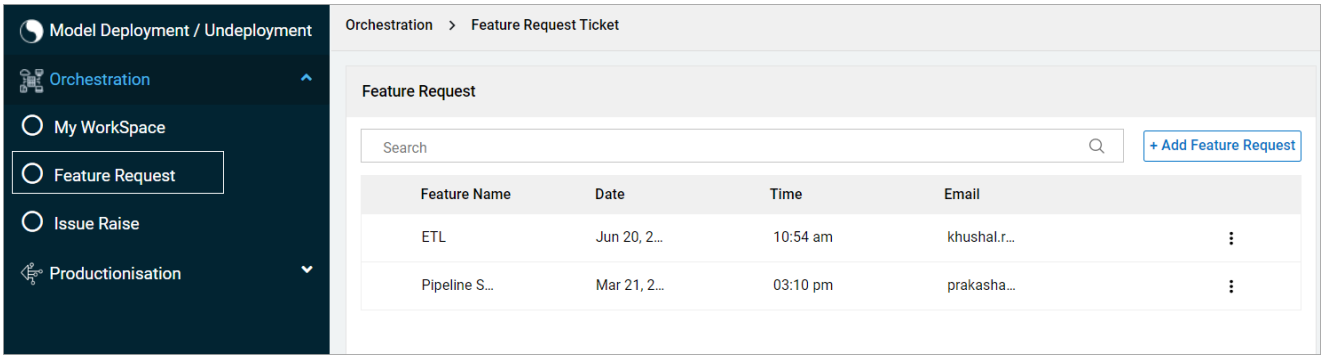
Requesting a feature

Feature Request allows you to raise a request for a new feature or update an existing feature.

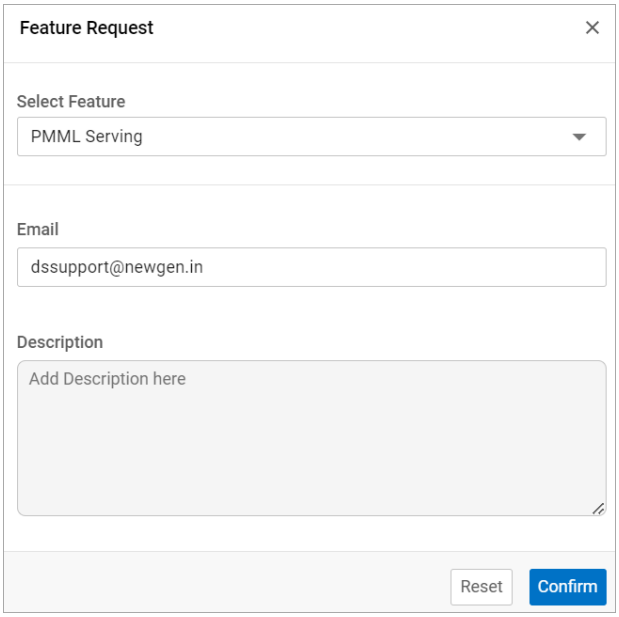
Newgen representative shares the email address for feature requests.

To raise a feature request, perform the following steps:

1. Go to **Orchestration** given on the navigation pane.
2. Select the **Feature Request**. The Feature Request Ticket page appears.



3. Click the **+Add Feature Request**. The Feature Request dialog appears.



4. Select the feature using the dropdown.

5. Enter the support email ID that you have received during NewgenONE Data Science Studio on-boarding to send the feature update requests.
6. Click **Confirm**. The feature requests get added.

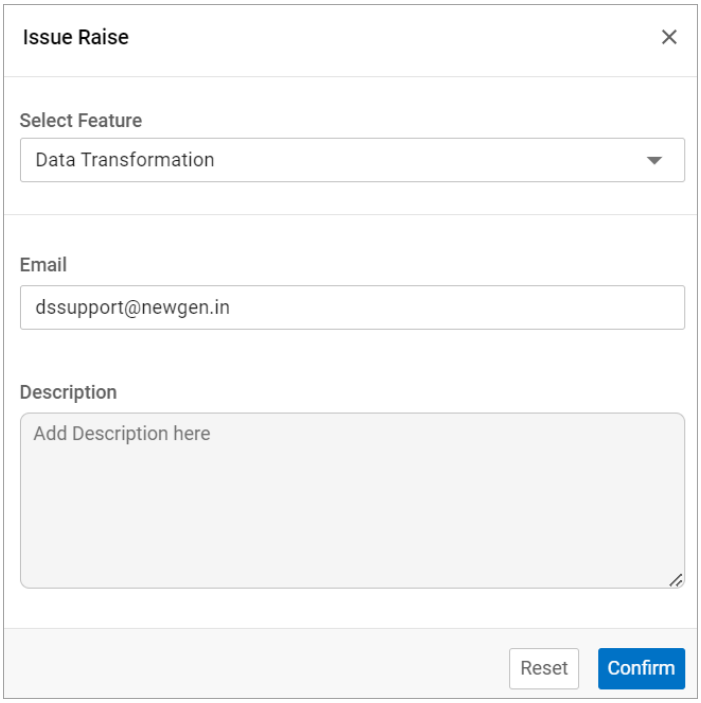
Raising an issue

Issue Raise allows you to raise an issue with the existing features.

Newgen representative shares the email address for issues raised.

To raise an issue on existing features, perform the following steps:

1. Go to **Orchestration** given on the navigation pane.
2. Select the **Issue Raise**. The Issue Raise page appears.
3. Click the **+Raise Issue**. The Raise Issue dialog appears.



4. Select the feature using the dropdown and enter the email ID that you have received during NewgenONE Data Science Studio on-boarding to raise an issue.
5. Click **Confirm**. The issue gets raised.

Productionisation

Productionisation manages the pipeline scheduling. With the scheduling, the pipeline can be served for production data.

This section consists of the following topics:

- Model publish
- Published model
- Production for approval

Publishing a model

Model Publish allows you to schedule pipelines in your workspace. Multiple pipelines can be scheduled together.

To publish a model, perform the following steps:

1. Go to **Workspace**.
2. Click the **Productionisation** and select the **Model Publish** option. The model Publish page appears.
3. Select the required project or you can search the project by its name using the searchbox.
4. After selecting the required project, drag the pipelines and drop it to the **Deploy** section given on the right pane.

   To search the pipeline with its name, use the searchbox given in the Pipelines section.

5. Click **Deploy**. The following options appear to set your scheduling parameters:

Options	Description
Project Name	Enter the project name for production.

Options	Description
Edit Sources	Update the following parameters using this option: • **File Format** — Select the file type. • Container Name — Select the container name. • **File Path** — Enter the file path for the pipeline where your file located in the container. • **Delimiter** — Enter the delimiter that is, commas (,), tab, space, semicolon (;), and quotes (".'). • **Mod**e — Select any of the following as the mode of source: ◦ DROPMALFORMED ◦ FAILFAST PERMISSIVE ℹ This field appears on selecting file format as JSON. • **Row Tag** — Select the tag of row. • **Root Tag** — Select the root tag. ℹ The row tag and root tag fields appear on selecting file format as XML. • **Source Name** — Enter the source name. • Click **Confirm** after setting all source parameters.
Drools	Select one of the following: • Yes — To apply the drools • No — To not apply the drools ℹ By default, it is set to No.
Change Frequency	Set the frequency according to your requirement. ℹ By default, it is set to Daily.
Start Date	Set the start date using datepicker. ℹ You must set the start and end date if you change frequency to daily.
End Date	Set the end date using datepicker.
Run Hourly Basis	Select **Yes** to run the pipeline on hourly basis.
Frequency	Enter the frequency.
Buffer Time	Enter the buffer time in minutes.

Options	Description
Max Retries	Enter the maximum number that you want to allow for retry.
Admin Users	Select the administrators using the dropdown. The administrators you select here further approves your pipelines for deployment.
Enter Time	Enter the time when you want to run the pipeline in a day.

6. After setting all the parameters, click **Deploy**. The pipeline gets deployed.

After deploying the pipelines, all administrators receive notifications for approval. Once approved, the pipelines appear in the Published Models list.

Published model

Published Model provides the list of projects and pipelines, along with the details of the project that is deployed to production. It allows you to search the projects and filter them based on their approval status.

To view the published pipeline or model, perform the following steps:

1. Go to **Productionisation**.
2. Click the **Published model**. The Published Model page appears with the following tabs:
   • Projects — Comprises the list of all projects.

   > Click the sort order icon given against the searchbox to view projects with the all, pending, accepted, and rejected pipelines.

   • Pipelines — Comprises the list of all pipelines available on selected project.
3. Select the project or you can search the project with its name using searchbox. After selecting the projects, the deployed pipelines appear on the right pane.
4. Select the required pipeline or you can search the pipeline in the searchbox by its name.
5. Click the **more options** icon given next to the pipeline.
6. Click the **View Pipeline** option. The published pipeline page appears with the following details:
   • Project Name — Shows the project name of the published pipeline.

- Pipeline Name — Shows the published or deployed pipeline name.
- Pipeline Type — Shows the type of pipeline such as development.
- Created On — Shows the time and date when the pipeline is created.
- Tags — Displays the tags used in the pipeline.
- Comments — Displays the comments if any.

Production for approval

The Production For Approval allows you to view the approval status of the projects and pipelines. The pipelines in production allows you to view and modify the production pipelines from the production URL of the platform. The URL is different than the platform URL.

To view the pipelines in production, perform the following steps:

1. Go to **Workspace** and select the **Productionisation**.
2. Click the **Production For Approval**. The Product For Approval page appears.
3. Select the desired project. The pipelines present in project appears in the Pipelines section.
4. Click the **more options** icon given next to the pipeline. The following options appear:

Options	Description
View Result	Shows the results of the last successful run of the pipeline.
Open	Opens the pipeline in canvas. It only opens in view mode, so you cannot save any edits done in the pipeline. To edit the pipeline, go to the platform and republish the pipeline for production.
Dashboard	Allows you to navigate to the dashboard for the pipeline, where you can see the visual representation of various metrics of the model. You can also view the ROC, LIFT, and GAIN curves of the pipeline model.

Customizing notebook

The Notebook allows you to write custom Python code, which you can then use in a Python node during data transformation in the pipeline.

This section consists of the following:

- Jupyter Notebook
- Uploading a file
- Registering Pmml

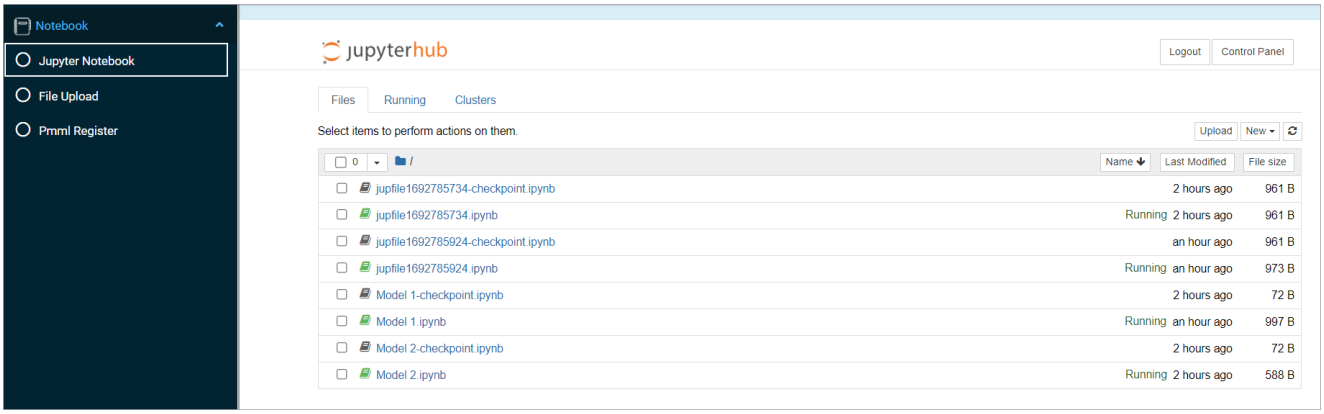
Jupyter notebook

The Jupyter Notebook allows you to write the custom code in python and embed that as a node into the model pipeline itself.

Keep the Running, Clusters and Control Panel as default. Thus, do not make any changes in them.

To write the code in the Jupyter Notebook, perform the following steps:

1. Go to **Notebook** given on the navigation pane.
2. Select the **Jupyter Notebook** using the dropdown. The Jupyter Notebook page appears with the list of notebook.



3. Click the **New** option given against the Upload on the right pane. The dialog appears with the following options:
    a. Notebook — Click the **Python 3 (ipykernel)** option. The Jupyterhub notebook page opens and allows you to create notebook as well as write the code in it.
    b. *(Optional)* Other — Select one of the following options:
        i. Text File — The text files allows you to keep notes for your reference.



Keep the Text file as the default and do not make any changes.

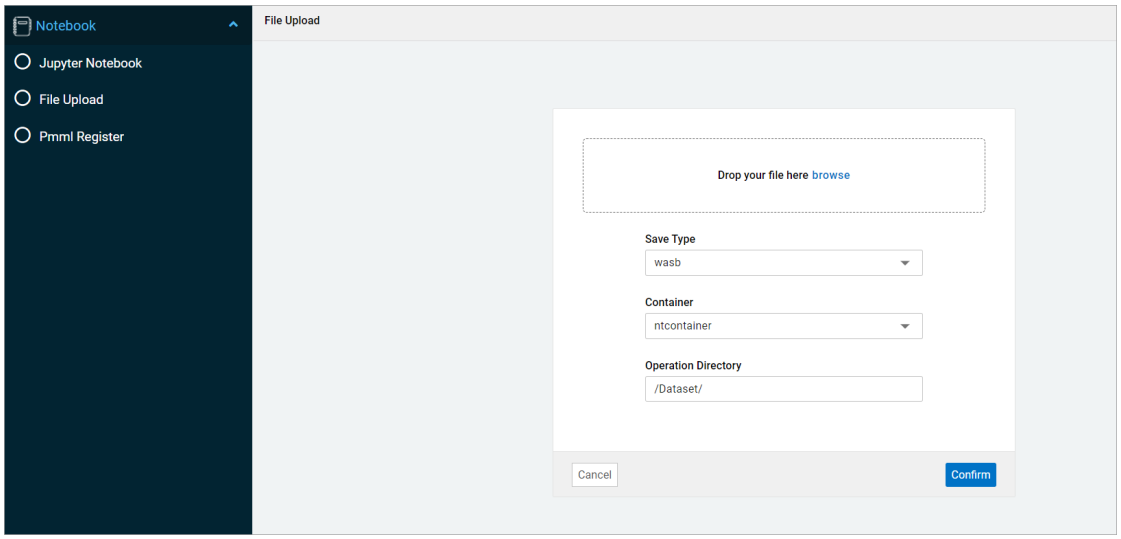
        ii. Folder — Use this option to create a new folder by following the below steps:
            i. Select the checkbox given against the files or folders. The following options appear:
            - Duplicate — Use this option to make a duplicate of the selected code file or folder.
            - Shutdown — Use this option to stop the running files.
            - View — Use this option to view the code available in the file or folder.
            - Edit — Use this option to edit or rewrite the code.
            - Delete — Use this option to delete the file or folder.
        iii. Terminal — Keep the Terminal as the default and do not make any changes.
4. Click **Upload** to add the custom code from your local system to the NewgenONE Data Science Studio platform.
5. Click the **Refresh Notebook List icon** to refresh the page of existing code files.
6. Click the **Name** to sort the files in alphabetical order that is, A to Z or Z-A.
7. Click the **Last Modified** to view the time of the modified files from ascending to descending order or vice-versa.

8. Click the **File size** to view the files size from ascending to descending order or vice-versa.
9. Click the **Logout** option given on the upper-right pane to sign out from the Jupyterhub notebook.

> By default, you can access the jupyter notebook and requires no credentials to sign in. It uses the NewgenONE Data Science Studio platform credentials to access the jupyter notebook. When signing in to Jupyter on a local on-premise deployment of the platform, you can use only the username (without @domain) of MSAD to sign-in.

Uploading a file

The File Upload feature allows you to upload the files such as *video*, *zip*, *csv*, and more directly on the Window Azure blob storage. Further, you can use them to fetch the data in the WASB data sources.



To upload a file on the Azure Blob storage, perform the below steps:

1. Go to **Notebook** and click the dropdown.
2. Click the **File Upload** given on the navigation pane. The File Upload page appears.
3. Click the **browse** and select the file from your local system to upload on the Azure Blob storage.
4. Select the **Save Type** as wasb using the dropdown.
5. Select the **Container** using the dropdown where you want to store the uploaded file.
6. Enter the **Operation Directory** where you want to save the uploaded file in the container.

7. Click **Confirm**. The file gets uploaded successfully.

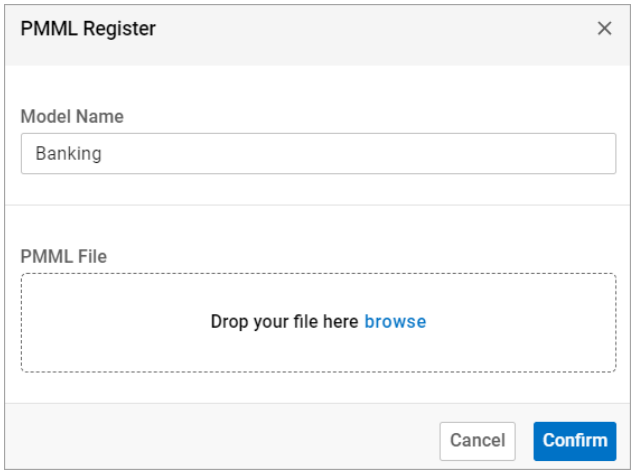
Registering Pmml

It allows you to register the Predictive Model Markup Language (PMML) file that does not exist on the platform. Then, you can use it to view the API details of the PMML.

To create a PMML file, save the data in Wasb sink node as PMML during pipeline creation. Hence, you can register those PMML files. For more information, refer to the Reading data source section.

To register the PMML, perform the following steps:

1. Go to **Notebook**.
2. Select the **Pmml Register** on the navigation pane. The Registered Pmml List page appears.
3. Click the **+Register New PMML**. The PMML Register dialog appears.

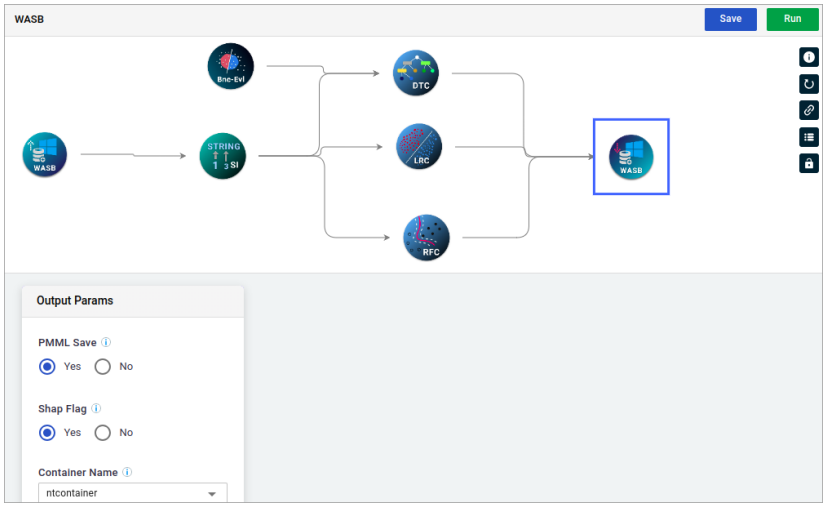


4. Enter the **Model Name** and upload the **PMML file**.
5. Click **Confirm**. The PMML gets registered.

Creating a pipeline

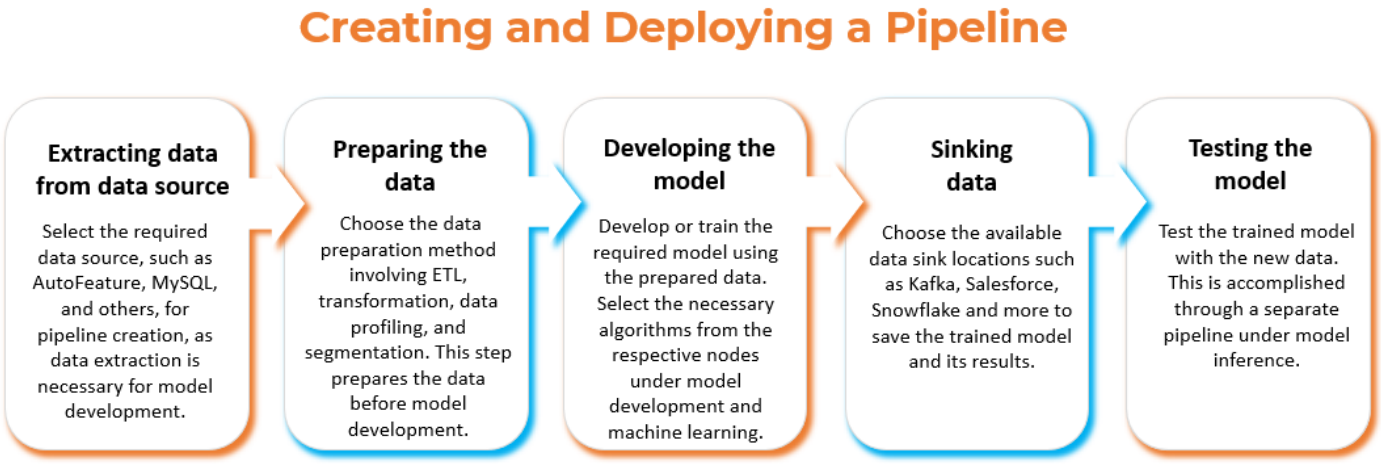
The Canvas allows you to create a pipeline using the various pre-built nodes as well as custom code.

To create a pipeline, it must include nodes in a sequential manner.



Creating and deploying a pipeline workflow

The following workflow depicts the typical pipeline creation and deployment using the NewgenONE Data Science Studio platform.



Preparing the data

Data preparation is the first and primary step for model pipeline creation. This section describes the following steps and processes involved in creating a data pipeline:

- Connecting data
- Joining data
- Refining data
- Exploring data
- Profiling data
- Transforming data
- Segmenting data

Connecting data

Connecting data includes creating a data pipeline. First, connect to data sources and then define a data sink (destination) to save the data.

NewgenONE Data Science Studio provides a variety of data source and sink connectors (nodes) for you to choose from and configure. These data source and data sink nodes are available in the Pipeline Source Sink on the Canvas.

The clusters of nodes are:

- Data Source Read
- Data Sink

Reading data source

The Data Source Read cluster provides you with various data connector nodes that allows you to insert data in the pipeline from various sources.

To read the data source, perform the following steps:

1. Go to **Canvas**.
2. Click the **Pipeline Source Sink** on the navigation pane.

3. Select the **Data Source Read**. The following nodes appear:
   - AutoFeature
   - Wasb
   - Snowflake
   - Salesforce
   - Mssql
   - MySql

AutoFeature

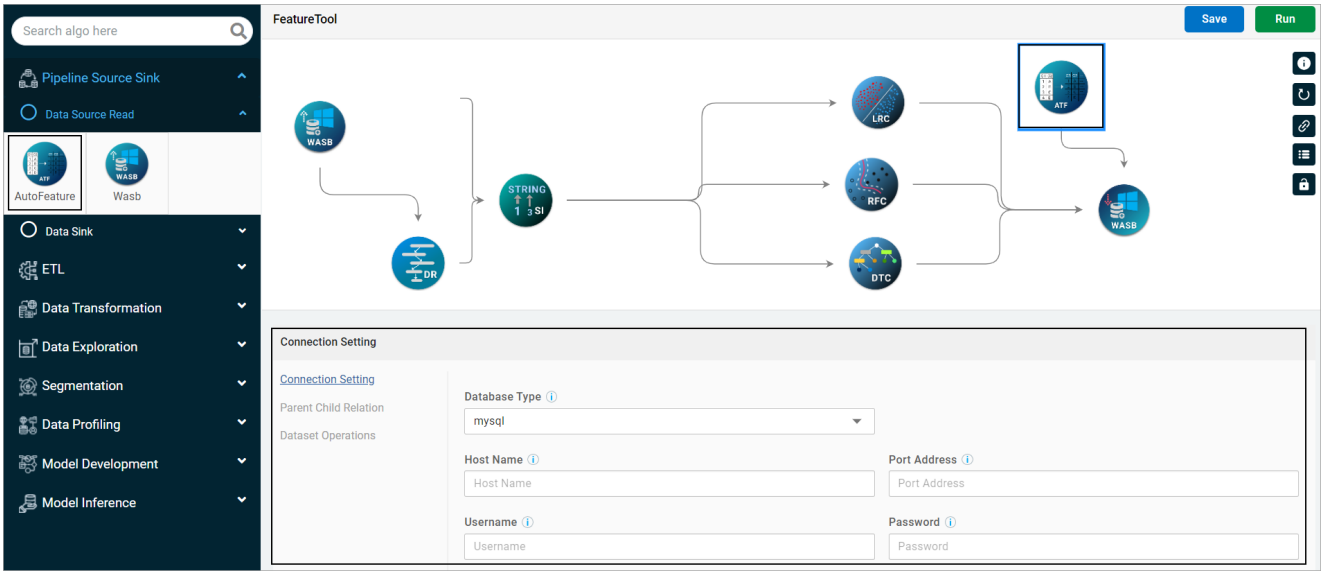
This section explains how to configure the autofeature.

Prerequisites — You must have MySQL.

The AutoFeature generates features automatically and operates based on the concept of entity-based feature engineering. It organizes data into entities (similar to database tables) and relationships (defining how these entities are related to each other). By using these relationships, Featuretools automatically generates new feature based on aggregations, transformations, and combinations of existing features.

To read the data from AutoFeature node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.



2. Drag the **AutoFeature** node and drop it on the Canvas.

You can only read data from the AutoFeature node when it's added to a connected pipeline.

3. Click the **AutoFeature** node within a connected pipeline to define its possible value. The Connection Setting section appears with the following tabs:
   - Connection Setting
   - Parent Child Relation
   - Dataset Operations
4. The AutoFeature node requires MySQL database connections and can only be connected to the following nodes:

• RF-r	• RF-c	• SVM	• GMM
• GB-r	• GB-c	• DT-c	• WasbSink
• Dt-r	• LR-c	• K	• LocalSink
• GLM	• NBC	• BI-k	• Dex

- To access the RF-r, GB-r, Dt-r, and GLM, refer to the Regression section.
- To access the RF-c, DT-c, LR-c, SVM, and NBC, refer to the Classification section.
- To access the BI-K, GMM, and K, refer to the Clustering section.

Connection Setting

The Connection Setting allows you to connect the AutoFeature node to the data source.

To connect the node to the database, perform the following steps:

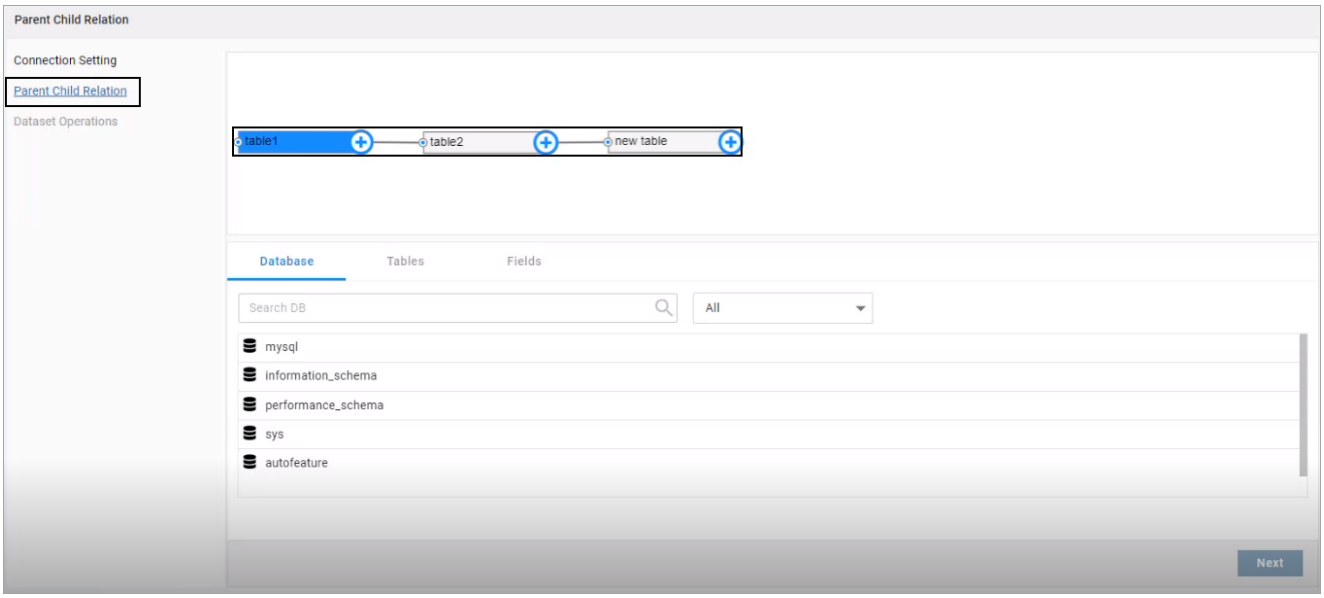
1. In the **Connection Setting** section, specify the following:
   - Database Type — Select the database type from the dropdown. For example, MySQL.
   - Host Name — Enter the selected database's hostname. For example, the MySQL host name can be, localhost.
   - Port Address — Enter the port address for the selected host.
   - Username — Enter the MySQL username.
   - Password — Enter the password associated with the above-specified username.
2. After specifying the above parameters, click **Connect**. The Connection established successfully message appears.

Parent Child Relation

The Parent Child Relation allows you to add tables and compare their columns. To compare table columns, ensure that they have the same data type.

To establish a parent-child relation within a table column, perform the following steps:

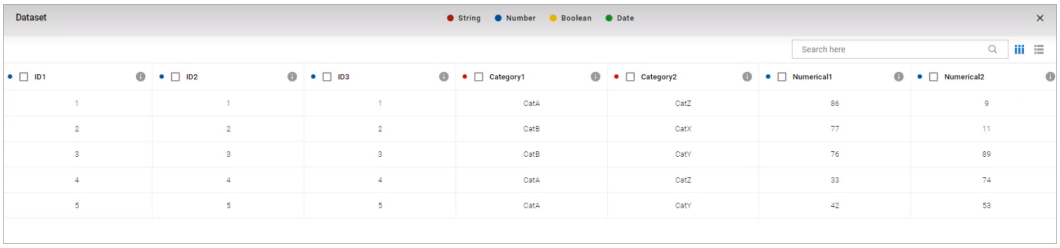
1. In the **Parent Child Relation** section, add the table using the **plus icon** ⊕. The new table gets added.



2. Click the table where you want to select a column. The following tabs appear:
   a. Database — Select the database from which you want to fetch the table columns. For example, autofeature.
      After selecting the database, click **Next**. The Tables tab appears.
      - Searchbox — Use this option to search the database by its name.
      - All — Use this option to select all, added, or not added databases.
   b. Tables — Select the table from the existing table list and click **Next**. The Fields tab appears.

      You can also search for tables by name in the searchbox.

   c. Fields — In the **Fields** section, select the following:
      i. Primary Id — Click **Primary Id** textbox. The Dataset dialog appears. Select the column checkbox that you want to set as the primary ID.
         - Searchbox — Use this option to search for columns by name.



- Dataset types — Shows the following dataset types:
  - Red — String dataset
  - Blue — Number dataset
  - Yellow — Boolean dataset
  - Green — Date dataset
- Row View — Use this icon to view the fields in rows.
- Column View — Use this icon to view the fields in columns.
- Searchbox — Use this option to search the data by name.
- Close — Use this icon to close the dataset dialog.

ii. Ref Id — Click **Reference Id** textbox. The Dataset dialog appears. Select the column checkbox that you want to set as the reference ID.

iii. Click **Next**. The Dataset Operations tab appears.

Dataset Operations

The Dataset Operations section allows you to select columns and their corresponding function values.

To configure the dataset operations, perform the following steps:

1. In the **Dataset Operations** section, the following fields appear:
   - Non-Features Columns
   - Select Function
2. Click the **Non-Features Columns** textbox. The Dataset dialog appears.
3. Select the data column checkbox. The Columns gets selected.

> You can select one or more columns according to your requirement.

4. Click the **Select Function** textbox. The dropdown appears with the following options:
   - DeleteAllNullRows
   - FillNullRowWithValue

> When selecting this option, define the user input value to replace null rows..

- ImputeWithMedian
- ImputeWithMean

5. Click **Save** to save the dataset operations settings.

Wasb

Windows Azure Blob Storage is a flat file system storage on the Azure cloud, where you can store data irrespective of its structure and the schema.

To read the data from Wasb node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.
2. Drag the **Wasb** node and drop it on the Canvas.

   You can only read Wasb node data if you add it to a connected pipeline.

3. Click the **Wasb** node placed in a connected pipeline to define its input parameter values. The Input Parameters section appear.
4. Select one of the **File Formats** from the dropdown:
   - Flat file
   - Parquet
   - Xlsx
   - Avro
   - Orc
   - Json
   - Xml
5. Select the **Container name** using the dropdown.
6. Select the **File Path** or enter the file name to add in the datasource.
7. If required, select the Column Category to categorize each column in your data as categorical or continuous in the dataset.

   The system automatically manages categorization based on column content by default.

8. Enter the **Delimiter value**, such as comma (,), tab ( ), and colon (:).

   This parameter appears when the selected file format is flat file.

9. In the **Mode** field, select one of the following Modes:
   - DROPMALFORMED — To drop the rows that cannot be parsed properly or not matching with the specified schema.
   - FAILFAST — To detect or terminate the errors.
   - PERMISSIVE

   This parameter appears when the selected file format is *JSON* or *XML*.

10. Enter the row tag and root tag for the selected column.

    This parameter appears when the selected file format is *XML*.

11. Select the **Non-features Column**. The selected column(s) gets removed from the dataset.
12. Enter a **source name**. It is preceded with each column name in the data.

Snowflake

Snowflake allows you to store, share and analyze large amounts of data using a cloud-native architecture.

To read the data from Snowflake node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.
2. Drag the **Snowflake** node and drop it on the Canvas.

   You can only read data from the Snowflake node when it is added to a connected pipeline.

3. Click the **Snowflake** node within the connected pipeline to define its input parameters value. The Input Parameters section appear.
4. Specify the following parameters:
   - Snowflake URL — Enter the URL of the snowflake warehouse.
   - Snowflake Account — Enter your Snowflake account name.
   - Snowflake Role — Enter the user's role in Snowflake, such as ACCOUNTADMIN, SYSADMIN, and others.
   - Username — Enter the username for the sign in.
   - Password — Enter the password for the username.
   - Warehouse — Enter the warehouse to use in snowflake. For example – COMPUTE_WH

- After specifying the above parameters, click **CONNECT** after specifying the above parameters. The connection gets successful if all details are correct. Then, specify the following fields:
  - Database Name — Select a database.
  - Schema — Select the schema corresponding to the above selected database.
  - Table Name — Select the table name, from where it reads the data.
  - Non Feature Columns — Select the non-feature column(s).
  - Execute Query radio button — Select **Yes** or **No**.
  - If you select **Yes**, enter the SQL query in the **Query** field to retrieve data from the selected database.
  - If you select **No**, provide the following details:
    - Where — Enter the SQL WHERE clause (if applicable) for filtering.
    - Source Name — Enter a source name. It is preceded with each column name in the data.

Salesforce

Salesforce is a widely-used CRM (customer relationship management) platform.

To read the data from Snowflake node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.
2. Drag the **Salesforce** node and drop it on the Canvas.

   You can only read data from the Salesforce node if you add it to a connected pipeline.

3. Click the **Salesforce** node placed in a connected pipeline to define its input parameters value. The Input Parameters section appear.
4. Specify the following parameters:
   - Salesforce Username — Enter the username for signing in.
   - Salesforce Password — Enter the corresponding password for signing in.
   - Salesforce Security Token — Enter the security token for an extra layer of authentication along with the password.

- After specifying the above parameters, click **CONNECT**. The connection gets successful if all details are correct. Then, specify the following fields:
  - Salesforce Api Name — Select the Salesforce API name based on your data requirements.
  - Non Features Columns — Select the non-feature column(s).
  - Source Name — Enter a source name. It is preceded with each column name in the data.

Mssql

Microsoft SQL Server (Mssql) is a relational database management system (RDBMS).

To read the data from Mssql node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.
2. Drag the **Mssql** node and drop it on the Canvas.

   You can only read data from the Salesforce node if you add it to a connected pipeline.

3. Click the **Mssql** node placed in a connected pipeline to define its input parameters value. The Input Parameters section appear.
4. Specify the following parameters:
   - **Machine IP** — Enter the IP or fully qualified name of the host (or server) of a SQL.
   - **Driver Name** — Enter the driver name to connect the SQL server. This is prefilled.
   - **Port** — Enter the SQL server host port number with that you want to connect.
   - **mssql User** — Enter the username for the SQL Server sign-in.
   - **Password** — Enter the password of the mssql user.
   - **Connect** — Click this button to establish a connection with the SQL server.
   - **Database Name** — Select the database using the dropdown.
   - **Table Name** — Select the table name using dropdown. This table belongs to the above selected database.
   - **Execute Query** — Select one of the following options:
     - **Yes** — To write a SELECT SQL query.
       - Query — Write the SELECT SQL query to retrieve the data from the SQL server.
     - **No** — To keep this as the default.
       - Where — Write the WHERE clause such as x>200, y==3, and more.

- The Query field appears when you select Yes in Execute Query.
- The Where field appears when you select No in Execute Query.

- **Select Column Category** — Click this option. The Dataset dialog appears. Then, select the column(s) checkbox that you want as the Categorical column(s) and close.
- **Non features Column**s — Enter the source name for database such as mssqlSource. It is required if you have selected two or more data source nodes in the pipeline. It is preceded with each column name in the data.
- **Source Name** — Enter the source name for this datasource node such as mssqlSource. By default, it is prefilled with Source1.

In case of two or more dastasource nodes in the pipelines. Then, you must define the source name as it gets preceded with each column name in the data.

MySql

MySQL is an open-source relational database management system (RDBMS).

To read the data from Mysql node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Reading data source section.
2. Drag the **Mysql** node and drop it on the Canvas.

   You can only read data from the Salesforce node if you add it to a connected pipeline.

3. Click the **Mysql** node placed in a connected pipeline to define its input parameters value. The Input Parameters section appear.
4. Specify the following parameters:
   - **Machine IP** — Enter the IP or fully qualified name of the host (or server) of a MySQL.
   - **Mysql Driver Name** — The driver name to connect the MySQL server. This is prefilled.
   - **Username** — The username for the MySQL sign-in.
   - **Password** — Enter the password of the MySQL user.
   - **Port** — Enter the MySQL server host port number with that you want to connect.
   - **Connect** — Click this button to establish a connection with the MySQL server.
   - **Database Name** — Select the database using the dropdown.

- **Table Name** — Select the table name using dropdown. This table belongs to the above selected database.
- **Execute Query** — Select one of the following options:
    - **Yes** — To write a SELECT MySQL query.
        - Query — Write the SELECT MySQL query to retrieve the data from the MySQL server.
    - **No** — To keep this as the default.
        - Where — Write the WHERE clause such as x>200, y==3, and more.

> - The Query field appears when you select Yes in Execute Query.
> - The Where field appears when you select No in Execute Query.

- **Select Column Category** — Click this option. The Dataset dialog appears. Then, select the column(s) checkbox that you want as the Categorical column(s) and close.
- **Non features Column**s — Enter the source name for database such as mysqlSource. It is required if you have selected two or more data source nodes in the pipeline. It is preceded with each column name in the data.
- **Source Name** — Enter the source name for this datasource node such as mysqlSource. By default, it is prefilled with Source1.

> In case of two or more dastasource nodes in the pipelines. Then, you must define the source name as it gets preceded with each column name in the data.

Sinking data

The Data Sink cluster comprises a set of nodes that allow you to save the output data in various sources.

To access the supported data sink types, perform the following steps:

1. Go to **Canvas**.
2. Click the **Pipeline Source Sink** on the navigation pane.
3. Select the **Data Sink**. The following nodes appear.
    - ES
    - Wasb
    - LocalSink
    - Salesforce
    - Snowflake

4. Drag the desired node that you want to connect in the pipeline and drop it on the Canvas.

   You can only sink the node data if you add that to a connected pipeline.

5. Double-click the node on the canvas. The Input Parameters section appear.
6. Specify the following parameters:

Nodes	Description	Input parameters
ES (Elastic Search)	Stores the model that serves output like Arima, AutoArima, and Arimax due to the fast search requirement.	Elasticsearch Index — Enter the indexname for data saving. After saving the data, you can retrieve it from there.
LocalSink	Stores the models locally on your machine. This node appears for using the on-premises application deployment.	• Sink Type — Select the *csv* or *parquet* • outputStorageDir — Name of the local directory(not the folder path) that saves the data. If the local directory specified here doesn't exist, the application creates that directory under the system-defined path and appends it to the system-defined output path.
Wasb	Stores the AI model's insight for consumption by other applications in a flat file form. Microsoft Azure Blob Storage allows you to store the data irrespective of its structure and the schema.	• Sink Type — Select the *csv* or *parquet*. • Container Name — Select the azure storage container to save the data in it. • outputStorageDir — Name of the local directory(not the folder path) that saves the data. If the local directory specified here doesn't exist, the application creates that directory under the system-defined path and appends it to the system-defined output path.

Nodes	Description	Input parameters
Snowflake	Enables users and organizations to store and share large amounts of data and analyze it using a cloud-native architecture. It is an easy-to-use, scalable warehouse with efficient performance.	For input parameters, refer to the Snowflake section. The parameters appear till Warehouse field only.
Salesforce	Enhances the decision-making and drive business strategies.	For input parameters, refer to the Salesforce section and refer to the following for additional parameters: • Sink Type — Select the *csv* or *parquet*. • output object name — Enter a name for the object created in salesforce with output data. Some fields might not appear for Salesforce data sink node.
MsSql	A relational database management system (RDBMS) that works as a data sink for NK pipelines.	For input parameters, refer to the Mssql section. However, there is no need to define the Table Name parameter.
MySQL	An open-source relational database management system that works as a data sink for ML pipelines.	For input parameters, refer to the Mysql section. However, there is no need to define the Table Name parameter.

ETL

ETL allows you to join and refine the data. For procedural details, refer to the following:

- Joining data — Joining data is necessary for data pipeline creation. It allows you to use data from multiple sources and it is necessary to perform operations on the consolidated data. For procedural details, refer to the Performing multiple data source operations section.
- Refining data — Data refinement involves operations to ensure data relevance. It also ensures data homogeneity and eliminates discrepancies to achieve

meaningful results from the model. For procedural details, refer to the Performing single data source operations section.

Performing multiple data source operations

Multiple Data source operations provide two operations that are Data Join and Data Union to collate data from two different sources.

To create multiple data sources, perform the following steps:

1. Go to **Canvas**.
2. Click the **ETL** on the navigation pane. The following sections appear:
   - Single Data Source operations
   - Multiple Data Source operations
3. Under the ETL, click the **Multiple Data Source operations**. The following nodes appear.
   - DJ (Data Join) — Allows you to join data from two data sources.
   - DU (Data Union) — Allows you to perform data union on multiple data sources. This operation does not require any input parameters.
4. Drag the desired node and drop it on the Canvas.
5. Click the **Data Join** node. The Input Parameters section appears.
6. Specify the following parameters:
   a. Under the Select Join Type, select one of the following join:
      - Inner join — Combines all the matching rows on the selected columns from both datasets.
      - Left join — Combines all the matching rows on the selected columns from both the datasets and includes unique rows from the first (left) dataset.
        - Select the source table using the dropdown.

          This parameter appears if you select the **Left Join** as the join type.

      - Right join — Combines all the matching rows on the selected columns from both datasets and includes unique rows from the second (right) dataset.
      - Full outer join — Combines all the unmatched rows on the selected columns from both datasets.

   b. Under the Common Column, select the **Source List** using the dropdown list. Joining data with sources helps retrieve complete information.
      - source 1 — Select the column in source 1 to apply the join.

- source 2 — Select the column in source 2 to apply the join.

Performing single data source operations

To create a single data source, perform the following steps:

1. Go to **Canvas**.
2. Click the **ETL** on the navigation pane. The following options appear:
   - Single Data Source operations
   - Multiple Data Source operations
3. Under the ETL, click the **Single Data Source operations**. The nodes appear on the navigation pane.
   - CA
   - CHR
   - CO
   - DD
   - DDR
   - TSMV
   - DE
   - DMC
   - DN
   - DOC
   - DR
   - VM
   - ETLMV
   - EXP
   - FI
   - LMN
   - M
   - MELT
   - NGram
   - RE
   - RO
   - RS
   - SDF
   - SS
   - ST
   - STM
   - SWR

Cast columns

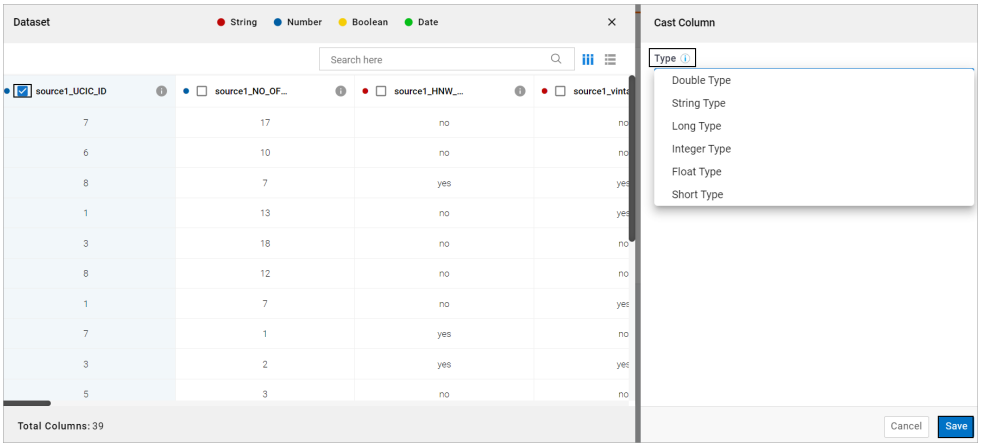
The Cast Columns (CA) node assists in casting the data type of an existing column into another data type. Illegal casting results in a null value.

For example, casting a column comprising alphabets into IntegerType or DoubleType shows a NULL result since it is an illegal casting. Legal casting includes converting a double type to an integer type that is, 12.231--> IntegerType --> 12.

To define the parameters in the CA node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **CA** node from navigation pane and drop it on the canvas.
3. Click the **CA** node. The Input Parameters section appears.

4. Click the **Cast Column** textbox. The Dataset dialog appears.



5. Select the column(s) checkbox. The Cast Column section appears on the right pane. You can also use the following options based on your requirements:
   - Dataset types — Shows the following dataset types:
     - Red — String dataset
     - Blue — Number dataset
     - Yellow — Boolean dataset
     - Green — Date dataset
   - Row View — Use this icon to view the fields in rows.
   - Column View — Use this icon to view the fields in columns.
   - Searchbox — Use this option to search the column by column name.
   - Close — Use this icon to close the dataset dialog.
6. In the Type textbox, select one of the following casting type:
   - **Double Type** — It stores floating-point numbers with double precision, typically using 64 bits to store data.
   - **String Type** — It stores sequences of characters, such as text. For example, Hello.
   - **Long Type** — It stores whole numbers, typically greater than the standard integer data type.
   - **Integer Type** — It stores whole numbers (positive or negative) without a fractional part, typically using 32 bits to store data.
   - **Float Type** — It stores floating-point numbers with single precision.
   - **Short Type** — It stores whole numbers, similar to integers but typically using fewer bits.
7. After selecting the casting type, click **Save**. The data type gets selected for column casting.

Chunker

Chunker (CHR) node allows you to extract phrases from unstructured text, enabling the analysis of sentences (only common noun and proper noun). However, it does not specify their internal structure or their role in the main sentence. Therefore, you must provide the column for chunking and the regex to use for the chunking process.

To define the parameters in the CHR node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **CHR** node from navigation pane and drop it on the canvas.
3. Click the **CHR** node. The Input Parameters section appears.
4. Click the **Chunker Input Columns** textbox. The Dataset dialog appears.
5. Select the **column(s)** checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.
7. Add the Regex (Regular Expression) Parsers by selecting the following checkbox from the dropdown
   - <NNP>+ (proper noun, singular)
   - <NNS>+ (noun plural)

Concat column

The Concat Column (CO) node allows you to concatenate the values of two columns. The resulting column is always be in string datatype.

To define the parameters in the CO node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **CO** node from navigation pane and drop it on the canvas.
3. Click the **CO** node. The Input Parameters section appears.

4. Click the **Concat value** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.
7. In the **Concat Delimiter** field, enter the **delimiter** for concatenation. The supported delimiter include (" ", @, ' ', #).

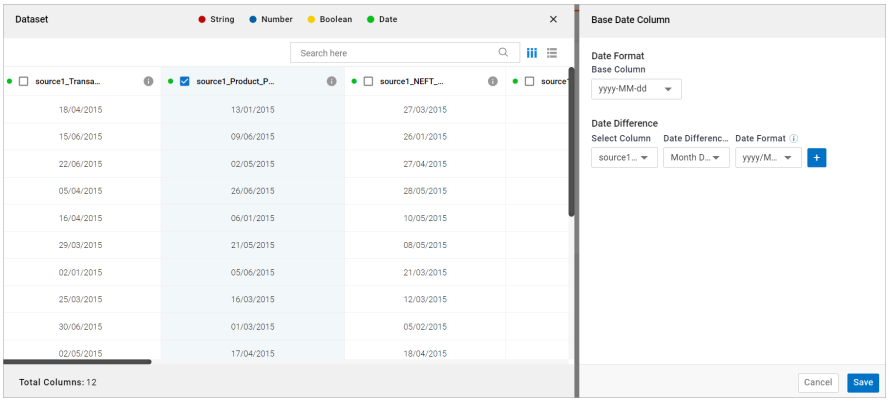
> Only a single delimiter is supported. In the case of multiple delimiters, the user-specified delimiter takes precedence.

Date difference

The Date Difference (DD) node allows you to determine the difference between two dates.

To define the parameters in the DD node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DD** node from navigation pane and drop it on the canvas.
3. Click the **DD** node. The Input Parameters section appears.
4. Click the **Base Date Column** textbox. The Dataset dialog appears.



5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View

- Column View
- Searchbox
- After selecting the column(s), the Base Date Column section appears on the right.

6. Specify the following fields:

Fields	Description
Date Format	Select the base column to determine the difference using the dropdown.
Date Difference	Select the following date difference: • Select Column — Select the column to determine the difference from the base column. • Date Difference Type — Select the required difference types from day, month, and year options. • Date Format — Select the date format for the value difference.

7. After specifying the fields, click **Save** and **close** the dataset dialog.

Delete duplicate rows

The Delete Duplicate Rows (DDR) node allows you to delete duplicate rows based on a subset of columns.

To define the parameters in the DDR node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DDR** node from navigation pane and drop it on the canvas.
3. Click the **DDR** node. The Input Parameters section appears.
4. Click the **Delete Duplicate Rows** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox to delete the duplicate rows in them. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox

6. After selecting the column(s), click **Close**.

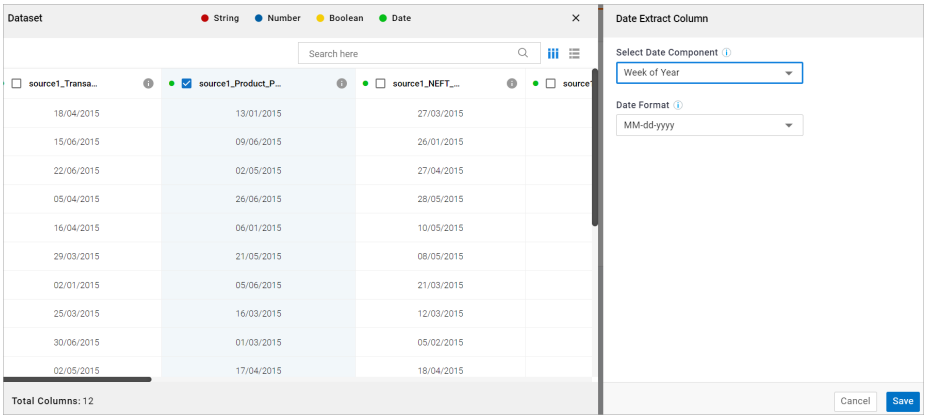
Date time field extract

The Date Time Field Extract (DE) node allows you to extract various date components from a date column.

For example, you can extract the week, day, quarter of the year and more.

To define the parameters in the DE node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DE** node from navigation pane and drop it on the canvas.
3. Click the **DE** node. The Input Parameters section appears.
4. Click the **Date Extract Column** textbox. The Dataset dialog appears.



5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), specify the following fields appear on the right.
   - Select Date Component — Select the required date component(s) that you want to extract from the following options:
     - Week of Year
     - Day of Month
     - Day of Year

- Year
- Month
- Quarter
- Date

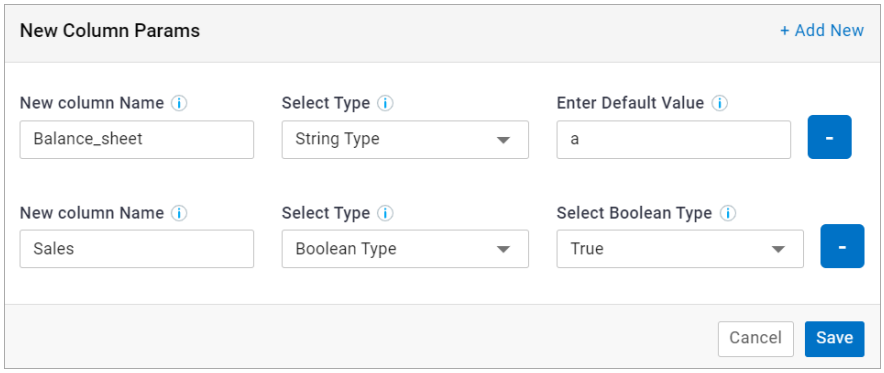
- Date Format — Select the date format for the selected column.

Dummy column

The Dummy Column (DC) node allows you to create dummy columns.

To define the parameters in the DC node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DC** node from navigation pane and drop it on the canvas.
3. Click the **DC** node. The Input Parameters section appears.
4. Click the **Dummy Column** textbox. The New Column Params dialog appears.



5. Enter the **new column name** without space and you can use underscore (_).
6. Select the column type from the following options:
   - Double Type
   - Integer Type
   - String Type
   - Boolean Type
7. Enter the **default value** for the column type.
8. Select **True** or **False** for boolean type.

> The Select Boolean Type field appears if you select *Boolean Type* in the *Select Type*.

Delete null rows

The Delete Null Rows (DN) node allows you to remove all null rows from a data frame.

To add the DN node in datasource, perform the steps from 1 to 3 as described in the Single data source operations section.

This node requires no input.

Drop columns

The Drop Columns (DR) node allows you to remove columns from a data frame.

To define the parameters in the DR node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DR** node from navigation pane and drop it on the canvas.
3. Click the **DR** node. The Input Parameters section appears.
4. Click the **Drop Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.

Document normalizer

The Document Normalizer (DOC) node allows you to normalize textual features by using regex patterns to remove unwanted texts and convert the data into lower case, if required.

To define the parameters in the DOC node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **DOC** node from navigation pane and drop it on the canvas.
3. Click the **DOC** node. The Input and Algorithm Parameters section appears.
4. Click the **DocNorm Input Columns** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.
7. Specify the following Algorithm Parameters:
   - Pattern — Enter the pattern. Then, it removes the regex patterns from the document if they match.
   - LowerCase — Select one of the following options:
     - Yes — Converts the string to lowercase
     - No — Does not convert the string to lowercase

ETL missing value

The ETLMV (ETL Missing Value) node fills in all the missing values in the selected columns or the entire dataset. Additionally, it enables you to choose a value from existing options, such as the mean, median, mode, a constant value, and more.

To define the parameters in the ETLMV node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **ETLMV** node from navigation pane and drop it on the canvas.
3. Click the **ETLMV** node. The Input Parameters section appears.

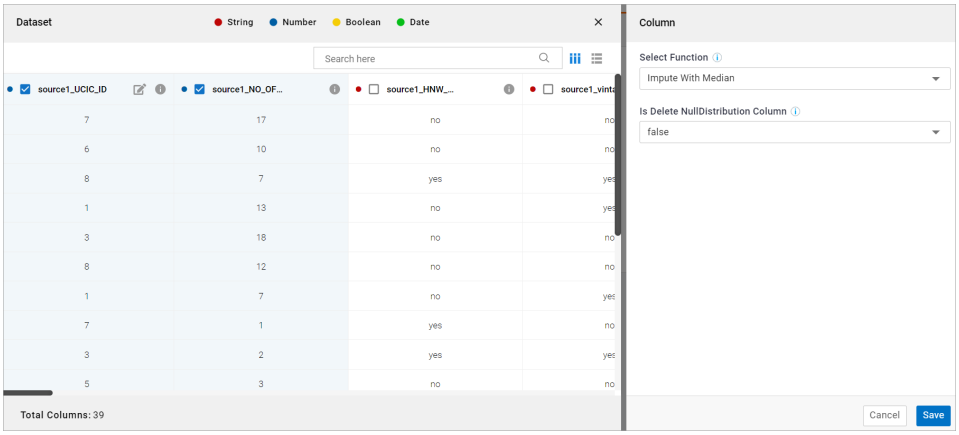
4. In the **Apply Function on** field, select one of the following option from the dropdown:
    a. Specific Column — This option allows you to apply functions to specific column(s) within a dataset. For procedural details, refer to the Specific Column section.
    b. Complete Data — This option allows you to apply functions to the complete dataset. For procedural details, refer to the Complete Data section.

Specific Column

Under the **Apply Function on**, if you select the **Specific Column**. Then, the Column field appears.

To define parameters in Column field, perform the following steps:

1. Click the **Column Field**. The Dataset dialog appears.



2. Select the **column(s)** checkbox. The Column section appears on the right. The following options appear:
    • Dataset types
    • Row View
    • Column View
    • Searchbox

3. Specify the following fields:

Fields	Description
Select Function	Select one of the following function to update the missing values: • Impute With Median • Impute With Mean • Impute With Mode • Fill Null Row With Value • Impute With Group Mean • Delete Null Rows With Column • Impute With Group Median
User Input Value	Enter the user input value for the selected column. This field appears if you select the **Fill Null Row With Value** in the Select Function field.
Select Column	Select the categorical column having numeric values. This field appears if you select the **Impute With Group Mean** and **Impute With Group Median** in the Select Function field.
Is Delete NullDistribution Column	Select one of the following options: • Yes — Enables the null distribution for the selected column(s). • No — Disables the null distribution for the selected column(s).
Percentage of Null Allow	Enter the allowed percentage between o to 100 or 0.0 to 1.0. This field appears if you select **True** in the Is Delete NullDistribution Column field.

4. After specifying the functions, click **Save**.

Complete Data

Under the **Apply Function on**, if you select the **Complete Data**. Then, the Select Function field appears.

To define parameters in Select Function field, perform the following steps:

1. Select one of the following function to apply on Complete Data:
   - **Delete all Null Row** — Deletes all rows that contain null values in any column.
   - **Fill Rows of Null values** — Fills null values in each row with the specified value.
   - **Impute With Mean and Mode** — Imputes numerical values with the mean and categorical values with the mode.
   - **Impute with Median and Mode** — Imputes numerical values with the median and categorical values with the mode.
   - **Delete All Null Distribution Column** — Deletes columns with null values exceeding a specified threshold percentage.
2. After selecting the function, specify the following fields:

Fields	Description
User Input String Value	Enter the string value. This field appears if you select the *Fill Rows of Null Value* in Select Function.
User Input Integer Value	Enter the integer value. This field appears if you select the *Fill Rows of Null Value* in Select Function.
Percentage of Null Allow	Enter the percentage of allowed nulls in any column. This field appears if you select the *Delete All Null Distribution Column* in Select Function.
Fill Function Name	Select between Impute with Mean and Mode, Impute with Median, and Mode. This field appears if you select the *Delete All Null Distribution Column* in Select Function.

Expression

The Expression (EXP) allows you to perform various functions with two columns.

For example, you can add the values of two integer type columns using the "+" operator.

To add the EXP node in datasource, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **EXP** node from navigation pane and drop it on the canvas.
3. Click the **EXP** node. The Input Parameters section appears.
4. In the Input Parameters section, specify the following parameters:
    a. Expression — Select the first column from the Search Columns dropdown.
        i. Click the **Search Columns** dropdown and select the second column.
        ii. Close the **Search Column** dialog.
        iii. Enter the operator (for example, "+", "-", "*", "/" and more) in between the selected columns in the Expression field.

Column filter

The Column Filter (FI) node allows you to filter column values based on a specified condition.

To define properties of the FI node in datasource, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **FI** node from navigation pane and drop it on the canvas.
3. Click the **FI** node. The Input Parameters section appears.
4. Select one of the following radio button:
    • Filter
    • Regular Expression
5. Specify the following fields:

Fields	Description
Filter	Select this checkbox to apply filter on column(s).

Fields	Description
Filter Value	Specify the filter value by clicking this field. • Search Columns — Select the column from dropdown on which you want to apply filter. • Filter Value — Enter the filter condition. The Filter condition includes **>**, **<**, **<=**,**>=**, And, **==** operators. • Click **Confirm** to save the added filter value. ⓘ This field appears if you select the **Filter** checkbox.
Group By Column	Select this checkbox to apply conditions on column(s).
Regex Value	Specify the regular expression by clicking this field. • Search Columns — Select the column from dropdown fo which you want to specify the regular expression. • Regex Value — Enter the regular expression. ⓘ Column name must start with ab and end with xy. • Click **Confirm** to save the added regex value.
The below fields appear if you select the **Group By Column** checkbox.	
Grouping By Column	Click and select the required column(s) checkbox from the dataset dialog. The following options appear: • Dataset types • Row View • Column View • Searchbox
Operations Column Name	Allows you to select the column and apply the operation accordingly. • Click and select the required column(s) checkbox from the dataset dialog. • After selecting column(s), select one of the following type in Operations Column Name. ◦ sum ◦ min ◦ max ◦ count ◦ avg • Click **Save** to perform the operation on the selected column.

Fields	Description
Operations Performed Column	Allows you to rename the column based on your operation. 1. Click this textbox. The Operation Performed Column dialog appear. • Select Columns — Select the column using dropdown. • Operation Performed Column — Enter the condition, if required. Filter condition can have **>, <, <=,>=, And, ==** operators. 2. Click **Confirm** to apply the operation.

Lemmatization

The Lemmatization (LMN) node allows you to convert all words in the data to their lemma (base form) while keeping the context of the word. For example, it converts caring to care.

To define the parameters in the LMN node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **LMN** node from navigation pane and drop it on the canvas.
3. Click the **LMN** node. The Input Parameters section appears.
4. Click the **Lemmatizer Input Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   • Dataset types
   • Row View
   • Column View
   • Searchbox
6. After selecting the column(s), click **Close**.

Mapping

The M (Mapping Null to 0) node allows you to populate null fields with 0.

To add the M node in datasource, perform the steps from 1 to 3 as described in the Single data source operations section.

This node requires no input.

MELT

The MELT node allows you to unpivot a data frame from wide format to long format. Additionally, it assists you in transforming a data frame where some columns serve as identifier variables while the remaining columns represent measured variables. Consequently, these measured variables are unpivoted to the row axis. As a result, two non-identifier columns remain as variable and value.

To define parameters for the MELT node, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Single data source operations section.
2. Drag the **MELT** node from navigation pane and drop it on the canvas.
3. Click the **MELT** node. The Input Parameters section appears.
4. Specify the following Input Parameters:

Parameters	Description
Grouping Column	Allows you to select the column that you want to preserve. • Click the textbox. The Dataset dialog appears. • Select the required column(s). The following options appear: ◦ Dataset types ◦ Row View ◦ Column View ◦ Searchbox
Value Variables	Allow you to select the column for melting. • Click the textbox. The Dataset dialog appears. • Select the required column(s). The following options appear: ◦ Dataset types ◦ Row View ◦ Column View ◦ Searchbox

Parameters	Description
Variables Name	Enter the name of the column that holds the melted column name.
Value Name	Enter the column name holding the melted column.

NGram

The NGram node allows you to simplify all combinations of adjacent words or letters of length n that you can find in your source text.

For example, in the sentence **fox is animal**, all 2-grams (or **bigrams**) are **foxis** and **isanimal**.

To define the parameters in the NGram node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **NGram** node from navigation pane and drop it on the canvas.
3. Click the **NGram** node. The Input Parameters section appears.
4. Click the **NGram Column Name** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.
7. In the **Nth Value**, enter the **number of terms**. The N is the number of terms in N-gram like ab, bc, cd, de, ef is 2-gram from string abcde.

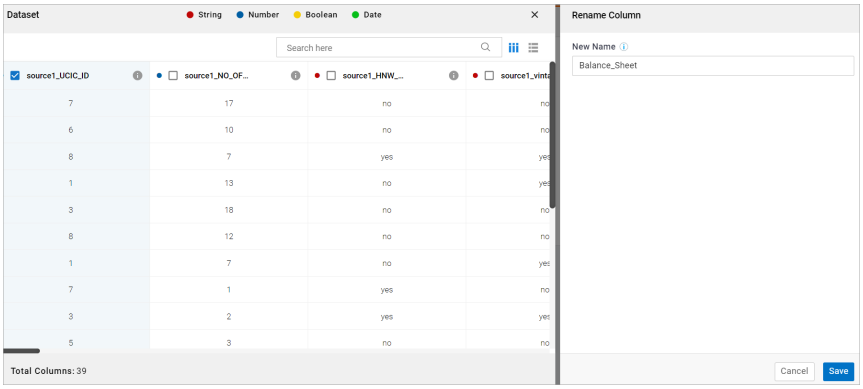
Column rename

The Rename Column (RE) node allows you to rename columns.

To define the parameters in the RE node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **RE** node from navigation pane and drop it on the canvas.
3. Click the **RE** node. The Input Parameters section appears.
4. Click the **Rename Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox

   After selecting the column(s), the Rename Column section appears.
6. Enter the new column name without any space and use the underscore (_).



7. Click **Save**. The column gets renamed.

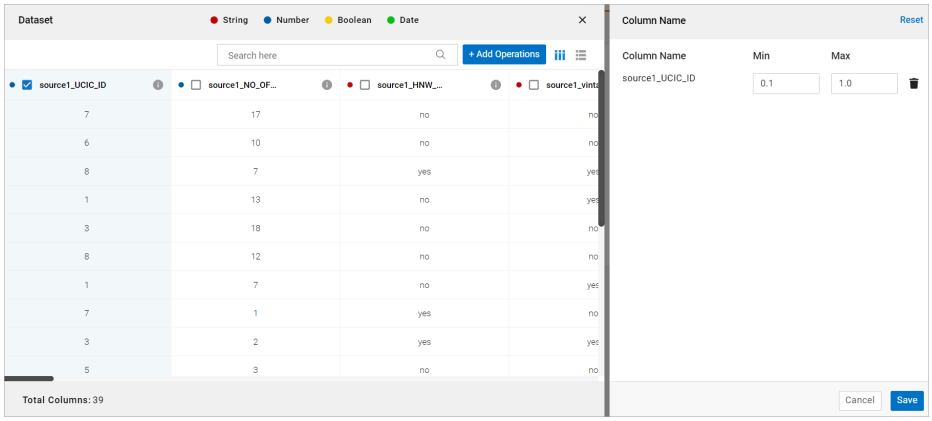
Remove outliers

The Remove Outliers (RO) node allows you to remove outliers based on the default quartile range.

For example, if you set the first quartile at 0.25 and the third quartile at 0.75, it calculates the Interquartile Range (IQR), which represents the range between the first and third quartiles. Data points outside this range are considered as outliers and are removed.

To define the parameters in the RO node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **RO** node from navigation pane and drop it on the canvas.
3. Click the **RO** node. The Input Parameters section appears.
4. Click the **Column Name** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **+Add Operations** button given against the searchbox. The Column Name section appears on right.



7. Enter the **0.1** as the min(minimum) and **1.0** as the max(maximum) value for the selected column.
8. After specifying the values, click **Save**.

Random sampling

The Random Sampling (RS) node allows you to offer an equal probability for selecting each sample in the dataset. It randomly selects the sample, creating an unbiased representation of the total population.

To use this feature, you must specify the desired percentage of the population and provide a seed number. You can also enable the option for replacement, allowing sampling units to occur more than once.

To define the parameters in the RS node, perform the following steps:

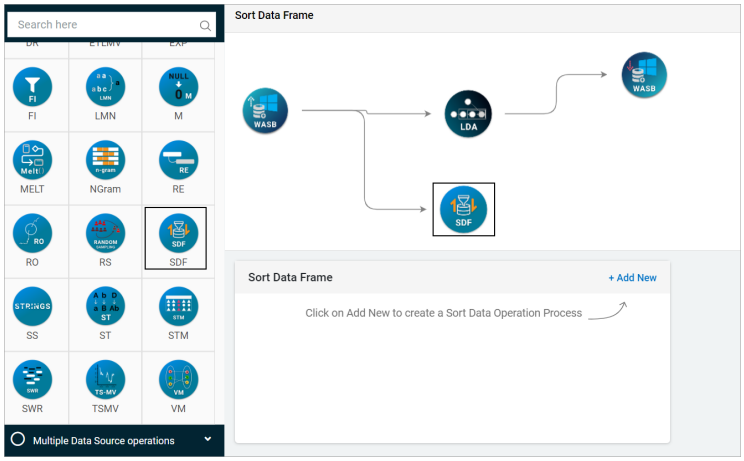
1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **RS** node from navigation pane and drop it on the canvas.
3. Click the **RS** node. The Input Parameters section appears.
4. Select one of the following:
   - Yes — Enable replacement
   - No — Disable replacement
5. Specify the following fields
   - Fraction — Enter the fraction value. By default, the value is set to *0.2*.
   - Seed — Enter the seed value for random number generation. By default, the value is set to *1234*.

Sort data frame

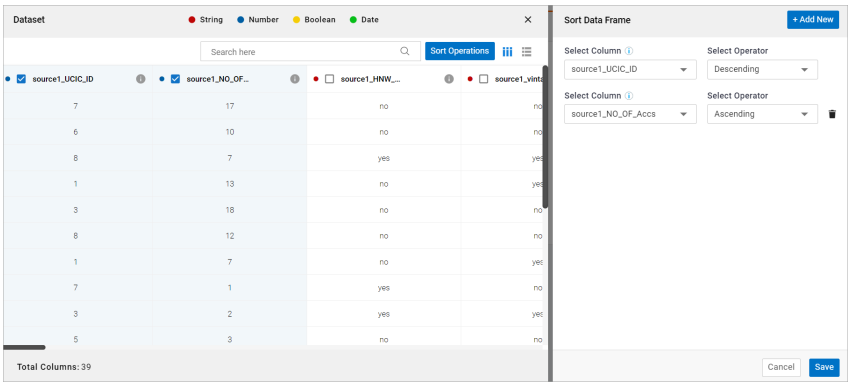
The Sort Data Frame (SDF) node allows you to sort the numerical values of a column in ascending or descending order.

To define the parameters in the SDF node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **SDF** node from navigation pane and drop it on the canvas.
3. Click the **SDF** node. The Sort Data Frame section appears.



4. Click **+Add New** option. The Dataset dialog appears.
5. Select required column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. Click the **Sort Operations** button given against the searchox. The Sort Data Frame section appears on right.

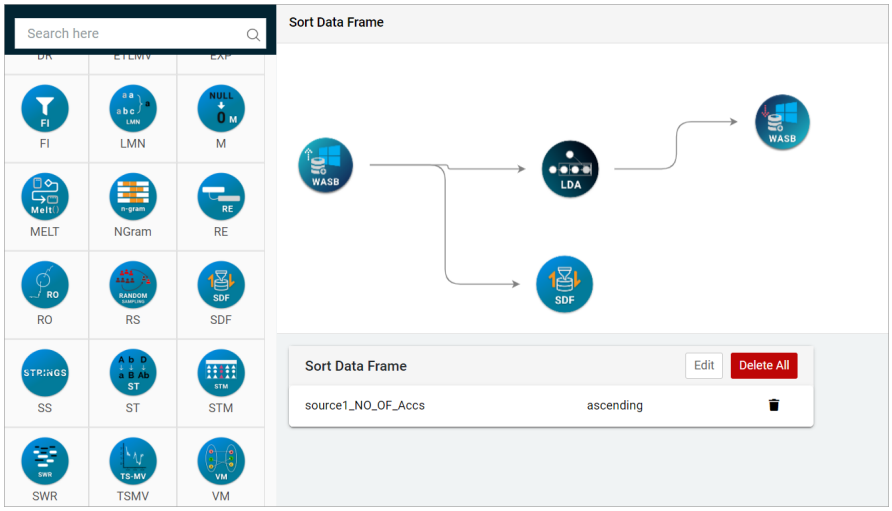


7. Select the Column using dropdown.
8. Select the operator as ascending or descending for the selected column.
9. To add more sort data, click **+Add New**. Then, select column and operator.
10. After selecting the sort operations, click **Save**. The added sort operations appears in the Sort Order Frame section.
11. In the Sort Data Frame section, click **Edit** to modify the added sort data frame operations.
12. To remove the added sort data frame, refer to the Deleting the sort data frame section.

Deleting the sort data frame

To delete the sort data frame, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the sort data frame.



2. In the Sort Data Frame section, click the **delete icon** to delete the sort operation. The Confirmation dialog appears.
3. Click **Confirm**. The sort data frame gets deleted successfully.
4. To delete all sort operations, click the **Delete All**. The Confirmation dialog appears.
5. Click **Confirm**. The sort data frame gets deleted successfully.

Slice string

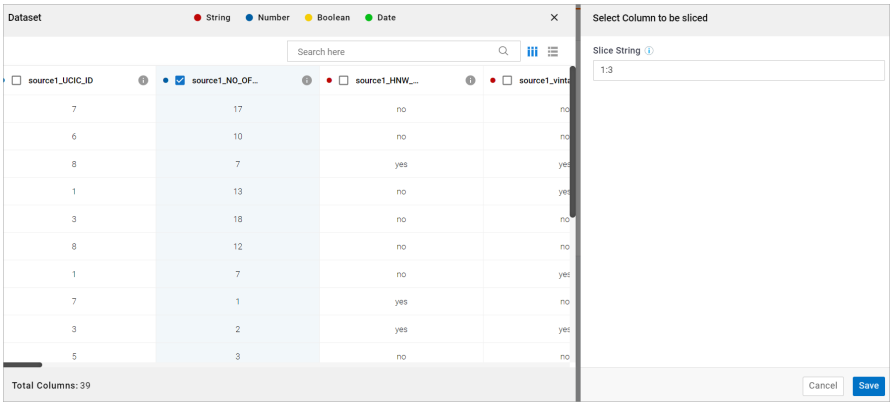
The Slice String (SS) node allows you to slice string values with the indexes given. The string index starts from 0 to length(string)-1.

For example, the string is **Number Theory** and input given to node is (3,8), then the Sliced String is **ber Th**.

To define the parameters in the SS node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **SS** node from navigation pane and drop it on the canvas.
3. Click the **SS** node. The Input Parameters section appears.

4. Click the **Select Column to be sliced** textbox. The Dataset dialog appears.



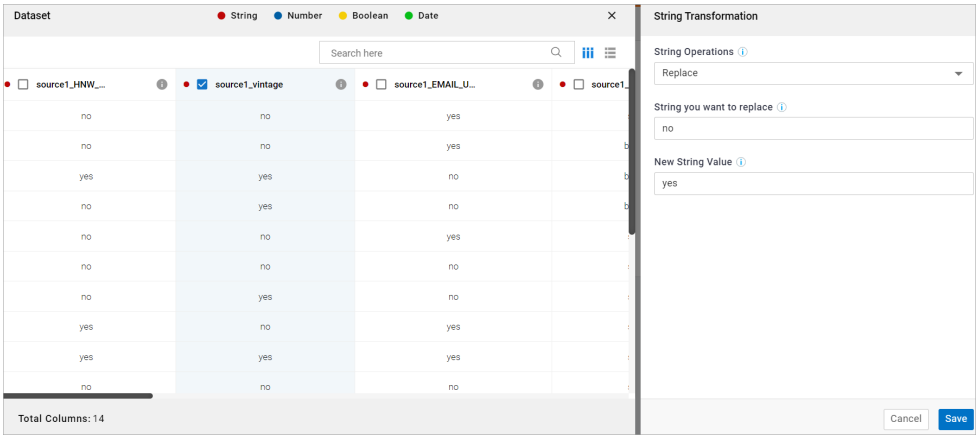
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. Enter the **value** in the Slice String. The format must be a:b where a is the starting index and b is the end index and string includes index from a to b-1.
   A string of length n has indexing from 0 to n-1.
7. Click **Save**. The Slice String value gets saved for the selected column(s).

String transformation

The String Transformation (ST) node allows you to perform various transformation operations on string type data. Using this, you can transform string type data to either uppercase or lowercase, you can trim the spaces as well as replace a string completely or partially.

To define the parameters in the ST node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **ST** node from navigation pane and drop it on the canvas.
3. Click the **ST** node. The Input Parameters section appears.
4. Click the **String Transformation** textbox. The Dataset dialog appears.



5. Select the required column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox

   After selecting the column(s), the String Transformation section appears on the right.
6. Select one of the following String Operations:
   - Lowercase
   - Uppercase
   - Trim
   - Replace
7. If you select Replace as string operation, then specify the following:
   - String you want to replace — Enter the string that you want to replace.
   - New String Value — Enter the new string for the selected string replacement.
8. Once you have selected the string operation, click **Save**.

Stemming

The Stemming (STM) node allows you to remove either the end or the beginning of a word. It utilizes a list of common prefixes and suffixes typically found in inflected words.

For example, it converts **caring** to **car**, **chocolatey** to **chocolate**, and more.

To define the parameters in the STM node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.

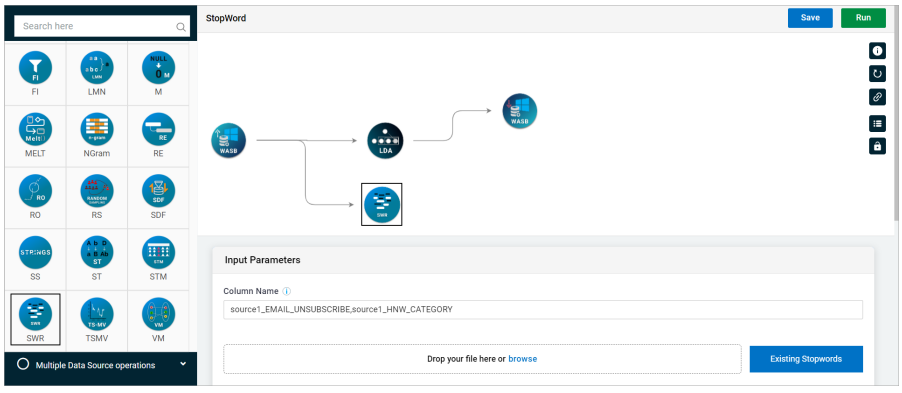
2. Drag the **STM** node from navigation pane and drop it on the canvas.
3. Click the **STM** node. The Input Parameters section appears.
4. Click the **Stemmer Input Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox
6. After selecting the column(s), click **Close**.

Stop word

The Stop Word (SWR) node allows you to remove stop words from the selected column. Furthermore, you can incorporate your own stop words by uploading a text file containing all the custom stop words you want to eliminate. The format of the text file must consist of one word per line.

To define the parameters in the STM node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **STM** node from navigation pane and drop it on the canvas.
3. Click the **STM** node. The Input Parameters section appears.



4. Click the **Stemmer Input Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View

- Column View
- Searchbox

6. After selecting the column(s), click **Close**.
7. Click **Browse** to upload the stop word file. The supported format is *txt*.

Time series missing value

The Time Series Missing Value (TSMV) node allows you to generate new data points based on the missing data points in the date column.

For example, if there's a series of dates with some values corresponding to those dates, and a few dates are missing in between, the TSMV node creates those missing dates with the corresponding values. Values can be filled with constants, backfill, or padding.

To define the parameters in the TSMV node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **TSMV** node from navigation pane and drop it on the canvas.
3. Click the **TSMV** node. The Input Parameters section appears.
4. Click the **Time Series Missing Value Column** textbox. The Dataset dialog appears.
5. Select the column(s) checkbox. The following options appear:
   - Dataset types
   - Row View
   - Column View
   - Searchbox

   If the data source doesn't contain any date column then the blank screen appears with the *There is no column available for further operation* message.

6. After selecting the column(s), specify the following fields appear on the right.

Fields	Description
Start Date	Select the start date using the date picker.
End Date	Select the end date using the date picker.
Select Column	Select the required column using the dropdown.

Fields	Description
Fill NAN Value of Split Series	Select one of the following options: • Back Fill • Pad • Fill with constant value
Constant Value	Enter the constant value.

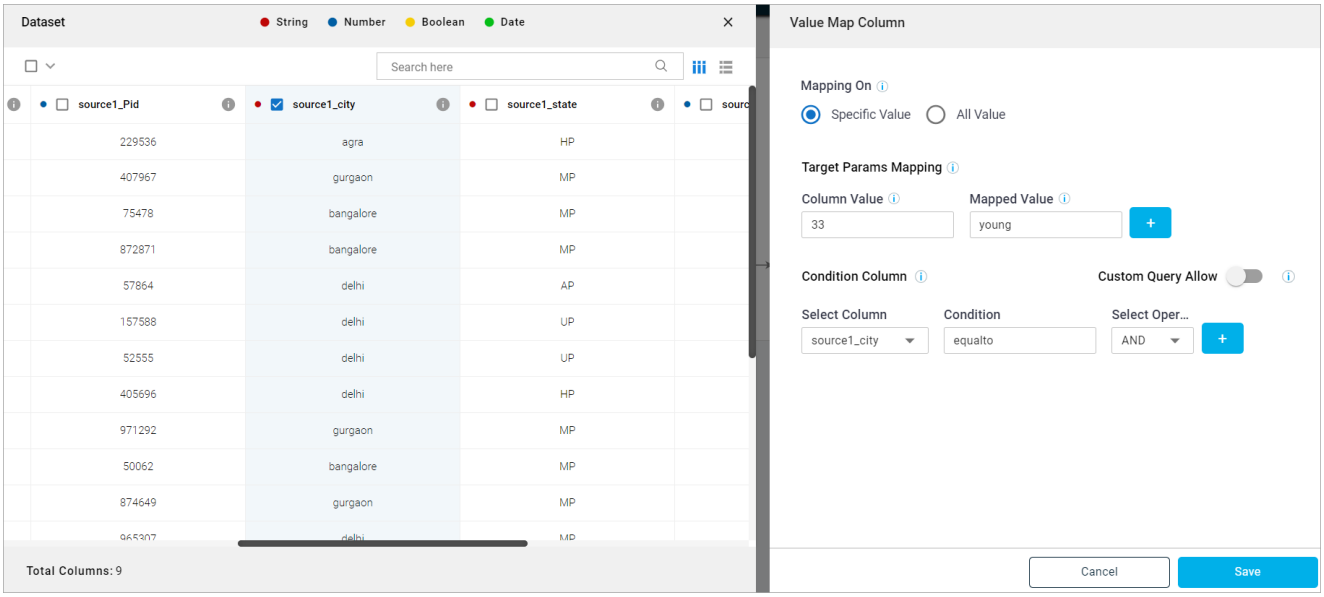
7. Click **Save**. The value gets added.

Value map operation

The Value Map Operation (VM) node allows you to replace values in the selected column, either all or specific, and even apply custom conditions or queries for value mapping.

To map the column values, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Single data source operations section.
2. Drag the **VM** node from navigation pane and drop it on the canvas.
3. Click the **VM** node. The Input Parameters section appears.
4. In the Input Parameters section, click the **Value Map Column** textbox. The Dataset dialog appears.
5. Select the column checkbox to map values.
   • Total Columns — Shows the available columns in the datasource.
   • Searchbox — Use this option to search the column by its name.
   • Row View — Use this icon to view the fields in row view.
   • Column View — Use this icon to view the fields in column view.
6. After selecting the column. The Value Map Column dialog appears.



7. Specify the following fields:
   a. Mapping On — Select one of the following:
      - Specific Value
      - All Value
   b. Under the Target Params Mapping, define the following fields:
      - Column Value — Enter the value for the selected column.

      The Column Value field appears if you have selected Specific Value.

      - Mapped Value — Enter the value for the mapped column.

      Click **+** icon to add more column and mapped value.

   c. Under the Condition Column, define the following fields:
      - Select Column — Select a column from the dropdown containing all the dataset column names where the condition must be applied.
      - Condition — Enter the value.
      - Select Operator — Select the operator from AND, OR, NOT, AND NOT, OR NOT, IN, NOT IN, and NONE.

      - Click the (+) button to enter multiple condition.
      - Click (-) button to remove the multiple condition.

   d. Enable the **Custom Query Allow** option to write the custom query. Then, specify the following options:
      i. Select Query Column — Select the required column(s).

ii. Query — Enter the condition. A condition can have **>,<, <=, >=, ==** operators along with AND,OR,IN, NOT IN, NOT, AND NOT, OR NOT logical operators according to your requirement.

e. Click **Save**. The value gets mapped in column.

Segmenting data

Segmentation is a process of dividing a dataset into distinct groups or segments based on the certain characteristics or features. It allows you to identify patterns or similarities within each segment and differences between segments. Additionally, it allows targeted analysis, personalized marketing, and a better understanding data subgroups.

To segment the data, perform the following steps:

1. Go to **Canvas**.
2. Click the **Segmentation** on the navigation pane. The SC (Segment Creation) node appears.
3. Drag the node and drop it on the canvas.
4. Click the node. The input Parameters appear. To specify the parameters of the SC node, refer to the Segment Creation section.

Segment Creation

It splits the data into different segments using segment conditions. These conditions are applied to both categorical and numerical columns, allowing the data to be split based on categories or values.


You can use this node at any stage of the data preparation.

To split data into different segment, specify the following parameters:

1. In the Input Parameter section, specify the following:
   a. Segment Condition — Select the column to apply the segmentation. Categorical columns have no conditions. The application segments the data by grouping it according to the values of the selected categorical column. For continuous columns, the segmentation is based on the conditions set in this parameter using the <, >, <=, >= operators. Multiple conditions are separated by commas in this case. It is possible to apply segmentation based on both

categorical and continuous columns. In this case, the application segments the data based on the values of the categorical column(s) and the condition(s) set for continuous column(s).

- To segment date columns, you can use these set of conditional parameters such as <,>,>=,<=. After selecting date columns, select the date format dd/mm/yy value during the conditions.

b. Rule Present — Select **Yes** if you want to choose the rule engine file.

> In case of no changes, keep the rule present default that is **No**.

Train Test split

The Train Test split or Split Train provides the various mechanism to split the data into training and testing data.

To define the parameters of the SplitTrain node, perform the following steps:

1. Go to **Canvas**.
2. Click the **Train Test Split** on the navigation pane. The SplitTrain nodes appear.
2. Drag the **SplitTrain** node from the navigation pane and drop it on the canvas.
3. Click the **SplitTrain** node. The Input Parameters appear.
4. Under the K-Fold cross validation Flag, select one of the following options. Here, K-Fold assess the performance of a reductive model and solves issues in the model evaluation process.
    a. Yes — Select this option to apply K fold cross validation for training.
        i. Number of Fold — Enter the number of folds.
    b. No — Select this option to apply training data into two parts based on possible strategies according to the parameter and specify the following:
        i. Split Type — Select one of the following split type:
        - randomSplit — Breaks the data into train and test parts randomly. After selecting this option, specify the following:
            - Test Percentage
            - Seed
        - continuousSplit — Indicates that all data above a specific index (particular row) assigns itself to the training dataset, while the lower part, which contains as many data points as the specified percentage, is assigned to the test dataset. After selecting this option, specify the following fields:

    - Test Percentage
- rowIndexWiseSplit — Resembles a continuous split with slight differences. After selecting this option, specify the following fields:
    - Train Start Index
    - Train End Index
    - Test End Index
- equalLabelDistributionSplit — Describes a custom or hypothetical function for splitting a dataset into training and testing sets while ensuring an equal distribution of class labels between the two sets. After selecting this option, specify the following fields:
    - Test Percentage
    - seed
- dateTypeSplit — Performs the time series modeling and requires split data based on the date column. After selecting this option, specify the following fields:
    - Time Start Date
    - Train End Date
    - Test End Date
    - Train Test Date Format
    - Train Test Date Column



After performing the train-test split, if a categorical column's category in the test dataset does not exist in the training dataset, then move the corresponding row from the test dataset to the training dataset to ensure category balance. You can select the column type (categorical or continuous) in the source node.

Exploring data

This section consists of the following topics:

- Data exploration
- Data quality

Data exploration

Data exploration helps in understanding the dataset, identifying patterns, detecting anomalies, and gaining insights before building and training models.

To define the data exploration, perform the following steps:

1. Go to **Canvas**.
2. Click the **Data Exploration** on the navigation pane.
   - Data Exploration
   - Data Quality
3. Under the Data Exploration, click the **Data Exploration**. The following nodes appear.
   - Cross Tab — Computes a pair-wise frequency table of the given columns. This table, known as a contingency table, is a matrix displaying variable frequency distribution.
   - Dex — Explores your data through various visualizations and statistical operations.
   - Dex-Action — Performs actions like remove Outlier, drop column based on Variance, drop column based on unique count, and impute missing values on auto-dex. It performs actions based on EDA on data in data exploration node.
4. Drag the desired node and drop it on the Canvas.
5. Click the node to define their input parameters. The Input Params section appears for the selected node on the canvas.
6. Specify the following parameters:

Nodes	Input parameters
CrossTab	• Group By Column — Select one or more group by column names. • Pivot Column (Label Column) — Select one or more label column names. • Column Operation — Select one or multiple columns to perform mathematical operation according to your requirement.
Dex	• Data Exploration — Select one of the following: ◦ **Auto** — Select this option if you want application to explore all operations (mean, median, quantiles, and more) across all columns in the dataset. ◦ **Manual** — Select this option if you want to provide the column name in additional field called **Exploration column** along with setting all operations you want to explore on this column. For more information, refer to the Manual Dex Operations section. • Exploration Column — Select the column names for manual data exploration. Also, specify the operational details to explain what can be explored when a column is selected.

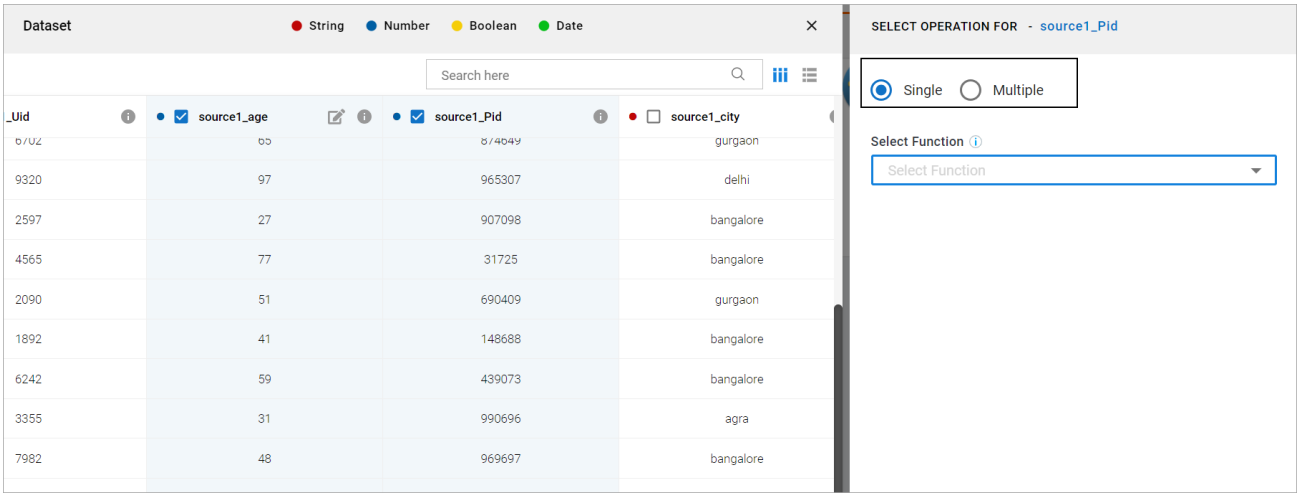
Nodes	Input parameters
Dex-Action	• Variance • Outlier • Column drop based on count • Feature scaling • Missing value imputation • Column drop based on null value • Correlation with label column For more information, refer to the *Dex Exploration Action Operations* section.

Manual Dex Operations

Manual Dex Operations explain the single and multiple column selection and its options.

To apply the Manual Dex Operations on single and multiple column, perform the below steps:

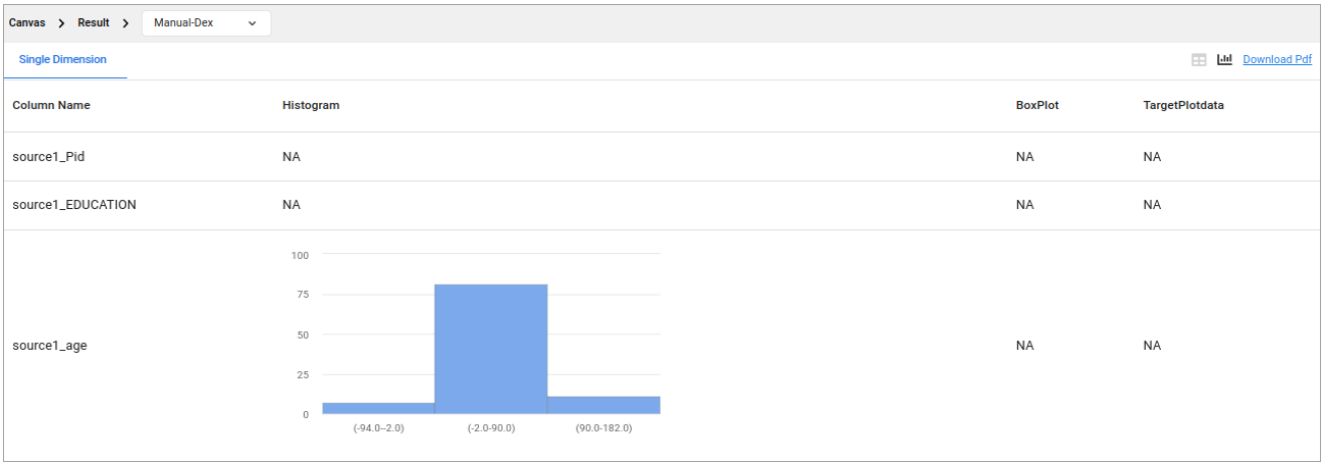
1. Perform the steps from 1 to 6 as mentioned in the Data exploration section.
2. In the Dataset dialog, select the required column(s). The Select Operation section appears in the right.
3. Select one of the following options:
   • Single
   • Multiple



4. After selecting the Single option, select the following required function using the dropdown:

Function	Description
CoefficientOfVariation (CV)	Defines the relative variability of a dataset column in relation to its mean (average). The CV calculation formula is (standard deviation/mean) x 100.
Describe	Works with Dex-in-action and the settings are configured in the dex-in-action node only. For more information, refer to the Dex-Action section.
Median	Calculates the median of the selected column.
Q_quantile	Requires **quantile** values to create bins of equal size for the selected column. Upon selecting this function, enter the percentile values that you want to calculate, such as 0.25, 0.50, 0.20, 0.40, 0.60, and 0.80.
DistinctValues	Calculates the distinct values in the selected column.
Variance	Calculates the variance in the selected column and defines the spread of values in the selected column. It uses the mean of values in the selected column to calculate the variance.
Kurtosis	Defines the shape of the probability distribution of the selected column.
Skewness	Calculates the asymmetry in the probability distribution in the selected column.
FrequentItems	Provides the list of values with occurrences exceeding the defined fraction in the Frequent Item Percentage Value for the selected column. Upon selecting this function, enter the frequent item percentage.
SumDistinct	Calculates the sum of distinct values in the selected column.
Sum	Calculates the total sum of values in the selected column.
MeanAbsoluteDeviation	Calculates the mean of the absolute deviation of each value in the column from the mean of the column values.
Range	Calculates the absolute difference between the maximum and minimum values of the selected column.

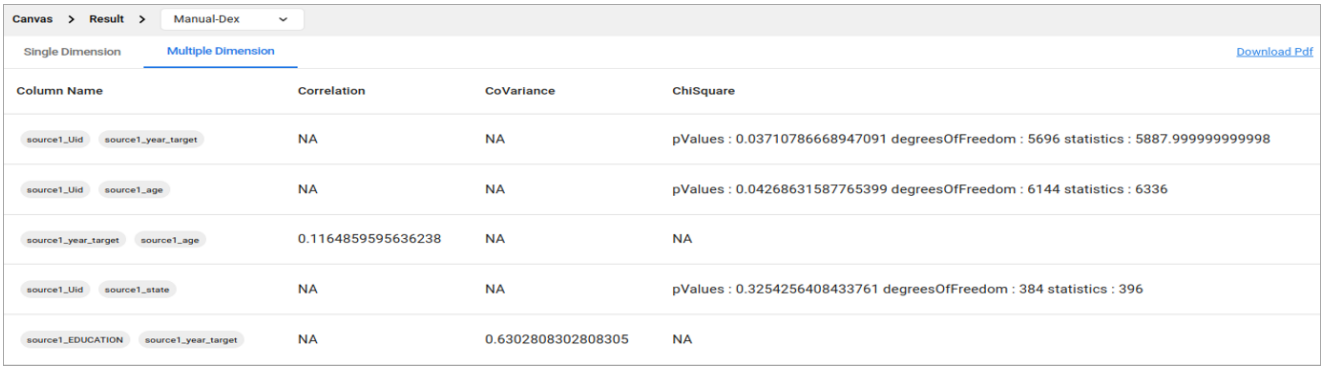
Function	Description
DetectOutliers	Provides the outlier percentage for the selected column.
Histogram	Creates the histogram for the selected column based on the number of bins defined in the additional field. The result screen displays the histogram. Upon selecting this function, enter the bin numbers that you want to create.
BoxPlot	Creates the boxplot for the selected column, defining its values under different categories from another column selected while setting the BoxPlot operation. The result screen displays the box plot against the selected primary continuous column.
InformationValue	Provides the information value to understand the values distribution for the selected column based on the other column selected to retrieve the information. For this other column, select a binary column for calculating information value. Information value is calculated for events and non-events only.



- The string types columns have only the following operations:
  - Distinct Values
  - Frequent Items
  - Describe
  - Information Value

5. After selecting the Multiple option, select the following required function using the dropdown:

Function	Description
Correlation	Calculates the correlation value between the selected column and the other subsequent column(s) chosen during the setup process.
CoVariance	Calculates the value that defines the degree to which two variables change together. In this case, the other variable(s) are the subsequent column(s) selected during the setup process of the CoVariance operation.
ChiSquare	Measures the independence between variables. In this case, the other variable(s) are the subsequent column(s) selected during the setup process of the ChiSquare operation.



6. After selecting the function, click **Save**. The Manual Dex operations get applied successfully.

Dex Exploration Action Operations

The Dex-Action node can be connected to the data exploration (dex) node only. It applies additional operation that can be performed based on the data exploration (dex) operations.

To apply the data exploration (dex) operations on single and multiple column, perform the below steps:

1. Perform the steps from 1 to 6 as mentioned in the Data exploration section.

2. In the input parameters, the select the following operations as required:

Operation	Description
Variance	Measures the values spread of a data feature statistically and have following fields: • **Variance Threshold** — Enter a value below or equal to the variance of the column. Then, the column gets dropped from the dataset. • **Action Columns** — Select the data exploration column name(s). These are subset of those columns that are selected for Dex with Variance operation.
Outlier	Appears if you select Q_quantile single column operation for Dex node in manual exploration. The fields for Outlier in Dex-in-Action are as follows: • **Outlier dropdown** — In the Outlier dropdown, select row based outlier or clip outlier according to your requirement. • **Action columns** — Select the data exploration column name(s). These are subset of those columns that are selected for Dex with Q_quantile operation.
Column Drop Based on Count	Appears if you select the describe or Distinct Values operation for single column operation in dex node in manual exploration. The available fields are as follows: • **Drop Column** — Select criterion to add a column. The value available is *uniqueCountEqualsRowCount* for **describe** single column operation in dex node while *uniqueCountEqualsOne* for **Distinct Values** single column operation in dex node for manual exploration. • **Action Columns** — Select the data exploration column name(s). These are subset of those columns that are selected for Dex with **describe** and/or **Distinct Values** operation.
Missing Value Imputation	Appears if you select the describe or median for single column operation in dex node for manual exploration. The available fields are as follows: • **Missing Value Imputation** — Select the value to impute the missing value. The value available is *mean* for **describe** single column operation in dex node while *median* value is available for **median** single column operation in dex node node for manual exploration. • **Action Columns** — Select the data exploration column name(s). These are subset of those columns which were selected for Dex with **describe** and/or **median** operation.

Operation	Description
Column Drop Based On Null Value	Appears if you select the describe operation for single column operation in dex node for manual exploration. The available fields are as follows: • **Percentage Of Null Supported** — Enter the threshold to drop the column based on null values available. If percentage of null is above threshold, then it gets removed. • **Action Columns** — Select the data exploration column name(s). These are subset of those columns that are selected for Dex with **describe** operation.
Correlation With Label Column	Appears if you select the correlation operation for multiple column operation in dex node for manual exploration. The available fields are as follows: • **Label Column** — It's the column with which the correlation operation on multiple columns, is defined in manual dex. • **Minimum Correlation** — Enter the threshold to drop the column based on correlation between label and action column. • **Action Columns** — Select the column on which you want to apply the operation or action.

3. Click **Save**. The operation gets applied. The following table summarizes the Dex-in-Action operations along with Dex operations:

Dex-in-Action Operations	Dex Operations – Single Column	Dex Operations – Multiple Column
Variance	Variance	-
Column Drop Based on Null Values	Describe	-
Column Drop Based on Count	Describe, Distinct Values	-
Missing Value Imputation (use mean value)	Describe	-
Missing Value Imputation (use median value)	Median	-
Outlier	Q_quantile	-
Correlation With Label Column	-	Correlation

Operations not listed in the above table for dex do not have any associated operations in dex-in-action. Therefore, no operations are enabled for those unlisted operations in the dex-in-action node.

Checking data quality

Data quality greatly influences the performance, accuracy, and reliability of the models built on that data. Inadequate data quality can lead to biased, inaccurate, and unreliable predictions, making it essential to ensure data quality throughout the entire machine learning process.

To define data quality properties, perform the following steps:

1. Go to **Canvas**.
2. Click the **Data Exploration** on the navigation pane.
   - Data Exploration
   - Data Quality
3. Under the Data Exploration, click the **Data Quality**. The Data Validation node appear.
4. Drag the Data Validation node and drop it on the Canvas.
5. Click the node to define its input parameters. The Input Params section appears.
6. Specify the following parameters:

Node	Description	Input parameter
(DV) Data Validation	It checks the data including checking alphanumeric data, date format, alphabetic data, presence of special characters, finding the length, most occurring values, distinct count, checking the percentage of missing values, and more.	• Single Column Operations ◦ check date format ◦ check alpha numeric ◦ check alphabetic ◦ contain special character ◦ find length ◦ most occurring value ◦ distinct count ◦ range of column ◦ check by data type ◦ missing value percentage column ◦ Is unique ◦ navigate column • Multiple Column Operations ◦ check Duplicate column ◦ check duplicate row For more information, refer to the Data Validation Operations section.

Data Validation Operations

The Data Validation Operations provide you the data validation results and doesn't perform any action on the data. Therefore, data validation pipelines are created separately (with source and data validation node) from the regular model development pipelines.

To achieve the data validation results, perform the following steps:

1. Perform the steps from 1 to 6 as mentioned in the Checking data quality section.
2. In the single column operations, specify the following fields:

Fields	Description
Check date format	Works on the date type column and validates the date column according to the format selected in the additional field named, **Date Format**. The validation result appears as follows: • True — The date format in the column matches the one selected in the dropdown. • False — The date format in the column does not match the one selected in the dropdown.
Check alpha numeric	Checks whether the selected column contains any alphanumeric values. • True — Column has alphanumeric values • False — Column does not have the alphanumeric values
Check alphabetic	Checks whether the selected column contains any alphabetic values. • True — Column has alphabetic values • False — Column does not have the alphabetic values
Contain special character	Checks whether the selected column contains have a special character. • True — Column has special character • False — Column does not have the special character
Find length	Provides the length of the string or numeric values in the column.
Most occurring value	Provides the value of the column with the highest number of occurrences along with the count of occurrences.

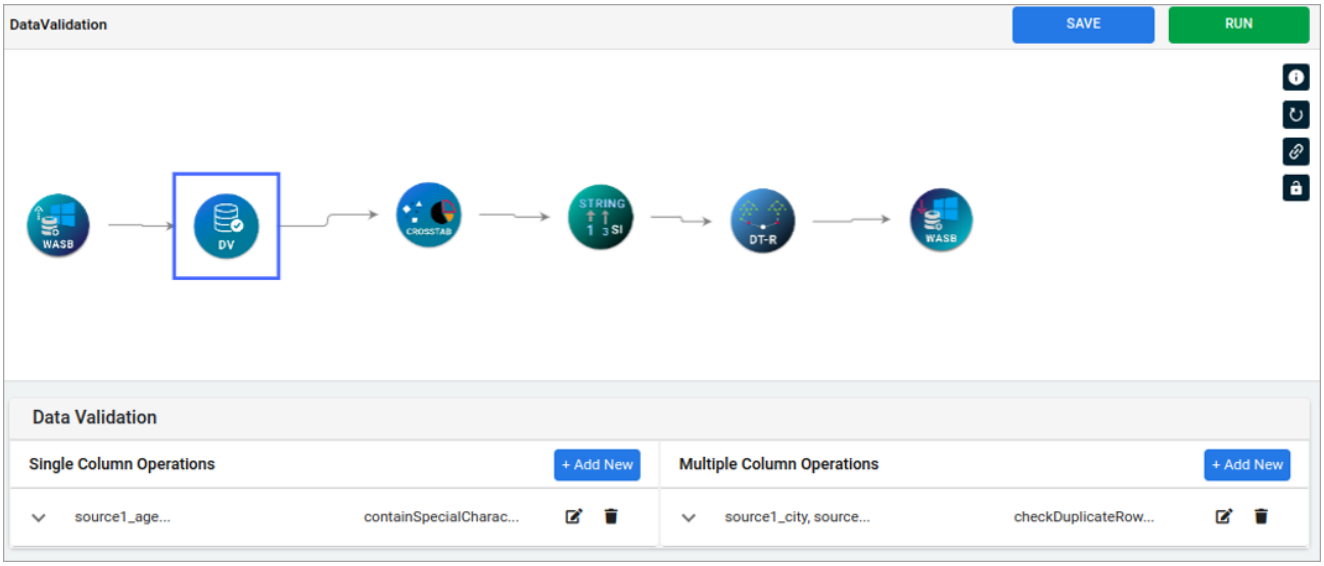
Fields	Description
Distinct count	Calculates the count of each value in the selected column and displays the result on the data validation screen.
Range of column	Provides the minimum and maximum values of the selected column.
Check by datatype	Determines whether the selected column values have same data types. • True — Column has special character • False — Column does not have the special character • OR — Provide the data type of the selected column
Missing value percentage	Provides the percentage of missing values in the selected column. A value of 0.0 in the data validation result indicates that there are no missing values in the column.
Missing value percentage per column	Determines the missing value percentage per column. • True — Column contains the missing value • False — Column does not have missing value
Is Unique	Validates whether all the values of the selected column are unique or not. • True — Values are unique • False — Values are not unique
Negative column	Determines the negative column as follows: • True — Column contains the negative value • False — Column does not have any negative value
Shape of data	Provides the shape of data in validation result screen.

After applying the data validation operation to a column, the column becomes unavailable for additional validations when you click the **Add New** button. However, you can apply multiple single-column validation operations to a column. Despite performing validation operations on multiple columns, the columns remain in the table.

3. In the Multiple column operations, specify the following fields:

Fields	Description
Check duplicate column	Determines whether the selected columns have duplicate values or not.

Fields	Description
Check duplicate row	Determines whether there are duplicate values in the above selected columns or not. The validation result displays the tuples of the duplicate indexes. (Optional) click the **Select All Functions** checkbox to select both of above functions.



Profiling data

Data profiling is the process of pre-processing or transforming data for modeling. It allows you to analyze data for completeness, accuracy, and accessibility. NewgenONE Data Science Studio provides you with various pre-built nodes to handle structured and unstructured datasets.

Structuring data

Structure data has a well-defined schema or format data that includes data in excel, csv, relational database tables, and more.

To structure the data, perform the following steps:

1. Go to **Canvas**.

2. Click the **Data Profiling** on the navigation pane.
3. Select the **Structured Data** using dropdown. The following nodes appears on the navigation pane:
    - B
    - CLO
    - MV
    - N
    - OHE
    - SI
    - SLR
4. Under the Structured data, drag the desired node and drop it on the canvas.
5. Click the node. The Input Parameters appear to define their values.

Bucketizer

The Bucketizer (B) transforms a column of continuous features to a column of feature-specified buckets by the signed-in user. By default, the bucket's lower limit is considered as negative infinity, while the upper limit is considered as the positive infinity.

For example, if you provide buckets like '30,50', then three buckets are created as follows:

- -infinity,30
- 30,50
- 50,+infinity

Values less than equal to 30 are placed in the first bucket, that is, -infinity, 30.

The splits must be increasing.

To define the input parameters for Bucketizer, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
    - Bucketizer Column — Select the column to bucketize. The Dataset dialog appears.
        - Search box — Allows you to search the required database by its name.

            - Click this  icon to view the dataset in column view.

- Click this  icon to view the dataset in list view.

- Specify the following in the Bucketizer Column:
  - Split — Enter the required split range. For example For split 12,27,50. There can be 4 buckets - [min bound value-12), [12,27),[27,50), [50, max bound value).
  - Min Bound of Split — Enter the minimum bound of the split.
  - Max Bound of Split — Enter the maximum bound of the split and then click **Save**.

> The -inf and inf are automatically passed as min_bound, max_bound if those parameters are blank.

Clip outliers

The Clip outliers (CLO) enables you to clip the outlier values to the nearest quantile range value.

To define the input parameters for Clip outliers, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
   - Select Columns to be applied — Select the columns that you want to clip based on the mentioned threshold. The dialog appears.
   - On the dialog, you can search the required column by its name using the search box.
   - Specify the following parameters:
     - Min — Enter the minimum clip threshold. For example, 0.25.
     - Max — Enter the maximum clip threshold. For example, 0.75.

> - Click this  icon to view the dataset in column view.
> - Click this  icon to view the dataset in list view.

3. Once done, click **Save**. The changes get saved.

Missing values

The Missing Values (MV) enables you to remove or fill missing values using the mean, median, mode, or a custom value provided by user.

To define the input parameters for Missing Values, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
   - Apply Function on — Select one of the following column using the dropdown:
     - Specific Column — Select this column and specify the following parameters:

Parameters	Description
Fill Null row with value	• Select this option to impute the missing row values with the provided input values. • User Input Value — Enter the string or integer value to impute null rows based on the data type of the column selected to apply missing value functions.
Impute With Mean	Select this option to impute missing values with a mean value in the selected column.
Impute With Median	Select this option to impute missing values with a median value in the selected column.
Impute With Mode	Select this option to impute missing values with mode values in the selected column.
Impute With Group Mean	Select this option to impute with group mean and select the other group column for calculating mean.
Impute With Group Median	Select this option to impute with group median and select the other group column for calculating median.

     - Complete Data — Select this function and specify the following parameters:

Parameters	Description
Fill Rows of Null Value	• User Input String Value — Enter the string value to use it in the null values in the rows for string. • User Input Integer Value — Enter the integer value to use it in the null values in the numeric type columns.
Impute With Mean and Mode	Select this option to impute missing values with a mean for numeric columns and mode value for string columns.
Impute With Median And Mode	Select this option to impute missing values with a median for numeric columns and mode value for string columns.

◦ Once done, click **Save**. The changes get saved.

Normalizer

The Normalizer (N) is a process that ensures each data sample has the same impact on the model, regardless of its original scale. It is commonly used to bring all feature vectors to a common scale.

To define the input parameters for Normalizer, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
   • p norm — Enter the integer number less than or equal to the number of columns in the dataset (excluding non-feature columns).
   • Exclude Columns or Categorical Columns — Select the excluded columns from the normalization.

One Hot encoding

One Hot Encoding (OHE) enables you to create a new binary feature value for each categorical value in a categorical column.

To define the input parameters for One Hot Encoding, perform the following steps:

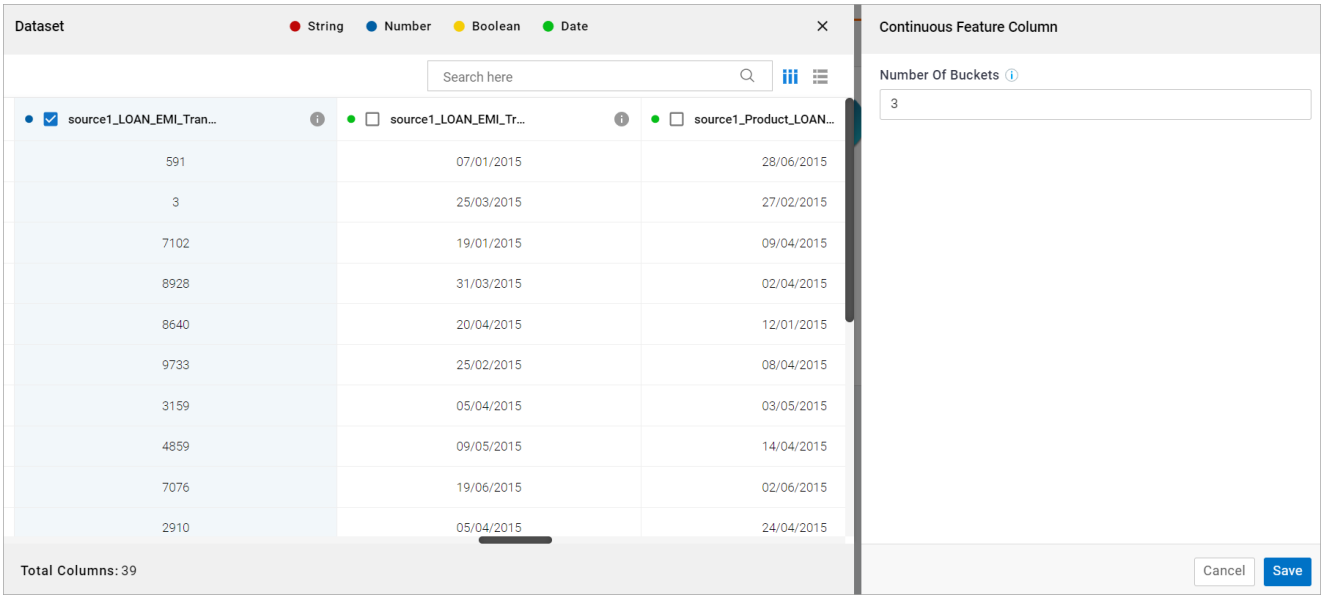
1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
   - Categorical Column — Select the columns to apply OHE. It must be a categorical column instead of continuous column.
   - Select Invalid Type — Select one of the following:
     - keep — To keep any extra categorical value(s) in the test dataset in serving.
     - error — To throw error if there are extra categorical value(s) in the test dataset and not present in the selected column (the categorical column used for OHE) in the train data used for the model development.

Quantile descretizer

Quantile Descretizer(QD) allows you to perform variable binning using the quantile values.

To define the input parameters for Quantile Descretizer, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Click the **Continous Feature Column**. The Dataset dialog appears.



3. Select the desired column checkbox on which you intend to perform variable binning.

- To search the data, use the searchbox and enter the data name.
- To view data in column view, click this icon.
- To view data in row view, click this icon.

4. After selecting the column, the Number of Buckets feature appear.
5. Enter the number of bins and click **Save**.

String indexer

String Indexer (SI) allows you to convert categorical values into numerical values ranging from 0.

To define the input parameters for String Indexer, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following parameters:
   - Categorical Column — Select the columns to apply string indexer. Numeric values must be assigned to the string values in this column.
   - Select Invalid Type — Select one of the following:
     - keep — To keep any extra categorical value(s) in the test dataset in serving.
     - error — To throw error if there are extra categorical value(s) in the test dataset and not present in the selected column (the categorical column used for SI) in the train data used for model the development.

Scaler

Scaler (SLR) allows you to scale the values using a variety of scaling operations available within the platform.

To define the input parameters for Scaler, perform the following steps:

1. Perform the steps from 1 to 5 as described in the Structured Data section.
2. Specify the following input parameters:
   a. Standard Scaler — Standardize features by removing the mean and scaling to unit variance using column summary statistics on the samples in the training set. Under Standard Scaler parameter, specify the following fields to apply standard scaler algorithm:

- Standard Deviation Value — Select **Yes** to scale the data to unit standard deviation. Else, select **No** to keep this value as default,
- Mean Value — Select **Yes** to center the data with mean before scaling. Then, it builds a dense output. Else, select **No** to keep the mean value as default.
- Standard Scaler Input Column — Click this column. The Dataset dialog appears. Then, select the column that you intend to standarize for scaling.

b. MaxAbs Scaler — Rescale the each feature individually to range (-1,1) by dividing through the largest maximum absolute value in each feature. It does not shift or center the data, and thus does not destroy any sparsity. To select the column, click the **MaxAbs Scaler** column. The Dataset dialog appears and allows you to select the column.

c. MinMax Scaler — Rescale each feature individually to a common range (min, max) linearly using column summary statistics. Under the MinMax Scaler parameter, specify the following fields:

- Minimum Scaler Value — Enter the minimum scalar value which acts as a lower bound after transformation.
- Maximum Scaler Value — Enter the maximum scalar value which acts as a upper bound after transformation.
- MinMax Scaler Input Column — Click this column, the dataset dialog appear. Then, select the column on which you want to apply the scaler functionality.

Unstructured data

Unstructure data has the combination of any kind of data such as data with images, videos, text documents, and more. The unstructured data is non-relational data.

To unstructure the data, perform the following steps:

1. Go to **Canvas**.
2. Click the **Data Profiling** on the navigation pane.
3. Select the **Unstructured Data** using dropdown. The following nodes appears on the navigation pane:
   - CVR
   - Doc2Vec

- TI

Count Vectorizer

The CountVectorizer (CVR) allows you to transform a text into a vector based on the frequency (count) of each word in the entire text.

To define the input parameters for CVR, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Unstructured data section.
2. Under the Unstructured data, drag the **CVR** node and drop it on the canvas.
3. Click the **CVR** node. The Input Parameters appear to define their values.
4. Click the **Count Vectorized Input Column** textbox. The Dataset dialog appears.
5. Select the column to apply the CountVectorizer. Also, you can use the following options to search the desired column:
   - Searchbox — Use this searchbox to find the data by its name.
   - Column View — Use this icon to view dataset in column.
   - Row View — Use this icon to view dataset in row.
6. Enter the minimum number of documents in the MinDf. The specifed value then appear in the vocabulary.

Term Frequency Inverse Document Frequency

The Term Frequency Inverse Document Frequency (TI-IDF) is a numerical statistic that reflects the importance of a word in a document within a collection or corpus. Primarily, text analysis uses TF-IDF, as it is useful for scoring words in machine learning algorithms for Natural Language Processing (NLP). Additionally, it increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This helps adjust for the fact that some words appear more frequently in general.

To define the input parameters for TI-IDF, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Unstructured data section.
2. Under the Unstructured data, drag the **TI-IDF** node and drop it on the canvas.
3. Click the **TI-IDF** node. The Input Parameters appear to define their values.
4. In the delimiter, enter the **delimiter** such as comma (,), colon (:), or more to separate the fields.
5. Enter the **number of features**. Then, the specified number of feature dimensions create the feature vector or number of hashing buckets in hashing TF.

6. Click the **Unstructured Column Name**. The Dataset dialog appears.
7. Select the unstructured column to apply the CountVectorizer. Also, you can use the following options to search the desired column:
   • Searchbox — Use this searchbox to find the data by its name.
   • Column View — Use this icon to view dataset in column.
   • Row View — Use this icon to view dataset in row.

Document as a vector

The Doc2Vec is a tool that represents documents as a vector and is an extension of a word2vec algorithm.

To define the input parameters for Doc2Vec, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Unstructured data section.
2. Under the Unstructured data, drag the **Doc2Vec** node and drop it on the canvas.
3. Click the **Doc2Vec** node. The Input Parameters appears to define their values.
4. Click the **Doc2Vec Input Columns**, select the column that you want to create as a vector.

   In pipeline execution, selecting multiple columns causes an error.

5. In the **Vector Size**, enter the integer value. The specified value decides the size of the output vector.

Dimensionality reduction

Dimensionality reduction helps in reducing the number of input features in a dataset.

To reduce the dimensionality, perform the following steps:

1. Go to **Canvas**.
2. Click the **Data Profiling** on the navigation pane.
3. Select the **Dimensionality Reduction** using dropdown. The following nodes appears on the navigation pane:
   • PCA
   • SVD

Principal Component Analysis (PCA)

Principal component analysis (PCA) is an analytical method that focuses on extensive datasets comprising a substantial quantity of dimensions or features for each observation. By enhancing data interpretability and retaining a maximal volume of information, PCA facilitates the visualization of multidimensional data. To elaborate, PCA stands as a statistical approach for diminishing dataset dimensionality. This process entails a linear transformation of the data into a fresh coordinate framework, wherein a reduced number of dimensions captures a significant portion of data variability compared to the original dataset.

This node empowers you to select the desired quantity of features. Subsequently, PCA identifies and delivers the top k features, streamlining data representation.

To define the input parameters for PCA, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Dimensionality Reduction section.
2. Under the Dimensionality Reduction, drag the **PCA** node and drop it on the canvas.
3. Click the **PCA** node. The Input Parameters appear to define their values.
4. In the **Number of principal components**, enter the required **number of principal components**.

Singular Value Decomposition (SVD)

It is matrix factorization technique which decomposes the data into three matrices. It allows you to perform singular value decomposition on the necessary dataset.

To define the input parameters for SVD, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Dimensionality Reduction section.
2. In the **Dimensionality Reduction**, drag the **SVD** node and drop it on the canvas.
3. Click the **SVD** node. The Input Parameters appear.
4. In the Input parameters section, specify the Number of singular Values. Enter the number of singular values that must be less than the number of features.

Handling imbalance data

An imbalanced data is a dataset where one data class holds more data points than another.

To handle the imbalance data, perform the following steps:

1. Go to **Canvas**.
2. Click **Data Profiling** on the navigation pane.
3. Select **Handling Imbalance Data** using dropdown. The SMOTE (Synthetic Minority Oversampling Technique) node appears. This node performs data augmentation by creating synthetic data points based on the original data points. This increases the minority class samples based on the up-scaling percentage. The SMOTE node works only on the encoded categorical column values.
4. Drag the **SMOTE** node from the navigation pane and drop it on the canvas.
5. Click the **SMOTE** node. The Input Parameters appear to define their values.
6. Specify the following input parameters:

Parameters	Description
Label Column name	Allows you to select the label column name. • Click this textbox. The Dataset dialog appears. • Select the required column checkbox.
K (nearest neighbours)	Enter the number of nearest neighbours.
Number of Iterations	Enter the number of iterations (required by the K-NN stage).
Random State in smote node	Enter the number for random state generation.
Over Sampling Percentage	Enter the over sampling percentage to modify the minority class.

SMOTE does not work with DP missing values. Therefore, if your data contains any null values, you must use ETL to impute or remove the missing values before SMOTE.

Transforming data

Data transformation is the process of altering the structure and format of the data for modelling. Newgen AI provides you the various pre-built nodes to apply custom transformation on the data.

To transform the data, perform the following steps:

1. Go to **Canvas** and click **Data Transformation** on the navigation pane.
2. Select the **Data Transformation** using dropdown. The following nodes appear:
   - GBF
   - DA
   - VD
   - Python
   - Rule Engine

Augmenting data

Data Augmentation (DA) can be used to generate tabular synthetic data. The primary goal of DA is to generate realistic synthetic data that preserves the statistical properties and patterns of the original dataset. It addresses the challenges of generating tabular data, where each column can have different data types, such as categorical, continuous, and other types, and complex interdependencies between features. The architecture consists of a generator and a discriminator, similar to a traditional GAN (Generative Adversarial Network). The generator takes random noise as input and attempts to generate synthetic samples that resemble the original data distribution. The discriminator, on the other hand, aims to distinguish between real and synthetic samples.

To define the input parameters for the DA node, perform the following steps:

1. Perform the steps 1 and 2 as mentioned in the Transforming data section.
2. Drag the **DA node** from navigation pane and drop it on the canvas.
3. Click the **DA node**. The Input Parameters section appear.

4. Specify the following parameters:

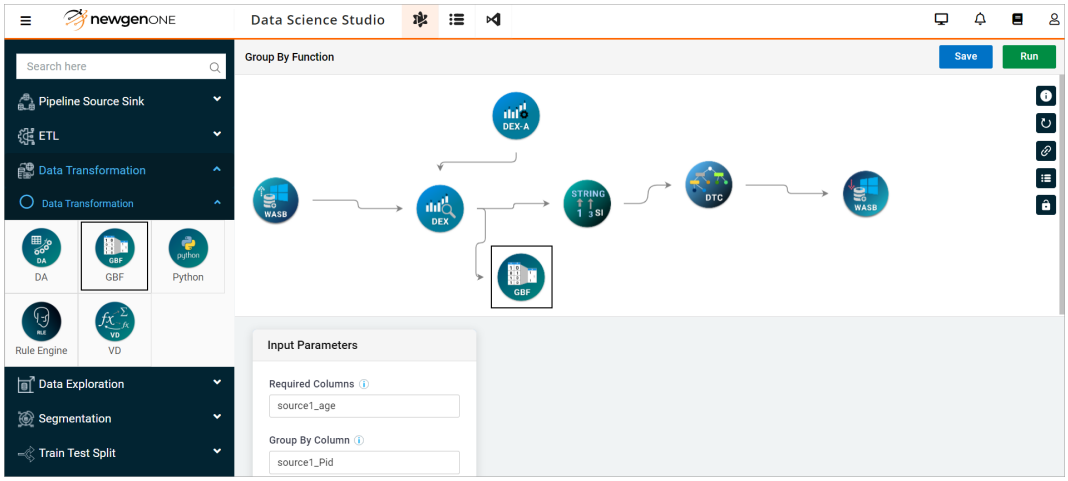
Parameters	Description
Number of Rows Required	Enter the required number of synthetic rows to generate.
Max Iter	Enter the number of iterations for the model to perform to optimize its parameters. For better results, enter a value between 300 and 500.
Batch Size	Enter the number of samples used in each step. For fast training, enter a high number like 1000.
Generator Dim	Enter the size of the output samples for each one of the Residuals. A Residual Layer gets created for each of the provided values provided. The default value is {256, 256,256}.
Discriminator Dim	Enter the sample size in the output for each of the Discriminator Layers. A Linear Layer gets created for each of the provided values. The default value is {256, 256,256}.
Embedding Dim	Enter the size of the random sample passed in the generator.
Log Frequency	Log frequency represents the audio data. Select one of the following options: • Yes — Select **Yes** to use the log frequency of categorical levels in conditional sampling. This argument affects how the model processes the frequencies of the categorical values used to condition the remaining values. • No — Select **No** to not use the log frequency.

Group by function

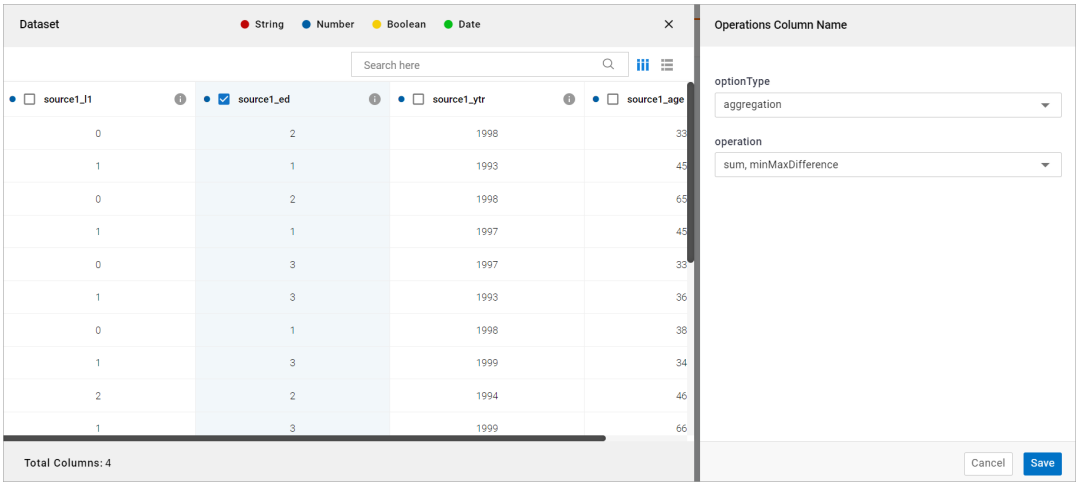
The Group By Function (GBF) groups data by a common characteristic and applies an aggregation function to the data in each group.

To define the input parameters for the GBF node, perform the following steps:

1. Perform the steps 1 and 2 as mentioned in the Transforming data section.
2. Drag the **GBF node** from navigation pane and drop it on the canvas.

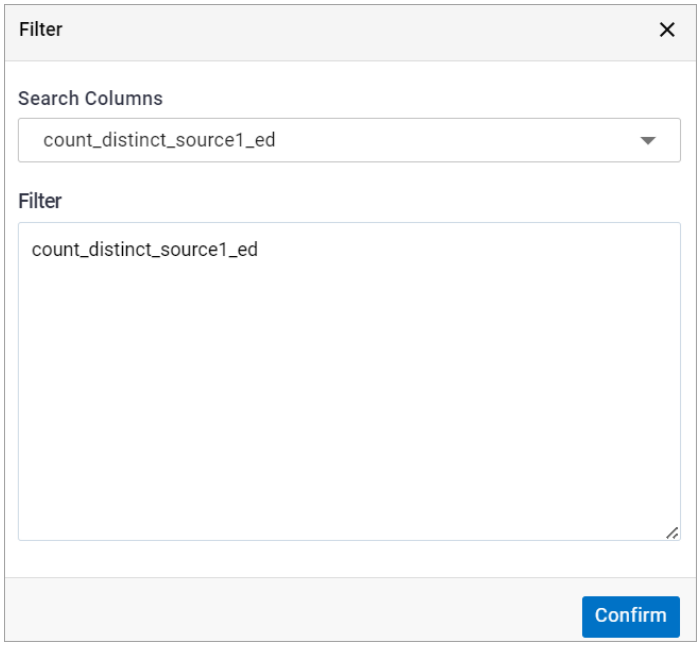


3. Click the **GBF node**. The Input Parameters section appear.
4. Click the **Required Columns** textbox. The Dataset dialog appear.
    a. Select the required column. You can also use the Searchbox or the Column View and Row View options to find the desired column:
5. Click the **Group By Column** textbox. The Dataset dialog appear.
    a. Select the required column for the group by operation.
6. Click the **Operations Column Name** textbox. The Dataset dialog appear.



    a. Select column(s) for applying Aggregate functions. The Operations Column Name section appears.
        i. *(Optional)* Select the timeDifference for a date column.
        ii. Select the operation(s) such as sum, min, max, count, and more using the dropdown.
        iii. In case of date column, select the dateFormat and dateColumn using the dropdown.
7. Select the **Order By Column** using dropdown. Here, you can select the column that you have selected in the above parameters.

8. Under the **Ordering Type** parameter, select the ascending or descending order.
9. In the **Filter**, click this textbox and the filter dialog appears.



   a. Enter filter condition using the dropdown that you intend to apply on group by column. The filter condition can be >, <, <=,>=, And, == operators.
   b. Click **Confirm**.
10. In the **Regex Condition**, click this textbox and the Regex dialog appears.
    a. Enter the regular expression and click **Confirm**. The defined regex condition gets applied on the columns to find and match the patterns in text.

Python notebook

Python (Python Notebook) allows you to create a python notebook for yourself and customize the functions accordingly.

Specify the Jupyter hub's notebook frame. In this, write the custom code according to your requirement. For more information, refer to the Customizing notebook section.

Rule engine

Rule Engine node plays a vital role in Production and Serving, handling various business rules.

For example, retraining becomes possible only when the number of new data points surpasses a specific threshold.

Another example, the default threshold for converting probability to a predicted label is 0.5. By utilizing this node, you can modify the threshold to any value between 0 and 1. You can log in to the KIE server and implement custom logic through this node.

The Rule Engine node encompasses the following types of scenarios:

Scenario 1

You can apply rules to each segmented dataset in the development pipeline. The expectation is that all segmented data must satisfy the rule to create an ML model. Currently, the conditions based on the total row count support is available. For example, "total count >= 10,000"

Scenario 2

You can apply rules to the production pipeline to validate retraining model conditions. The expectation is that all source or sources' data (for multiple sources) must satisfy the rule to retrain the ML model. To apply the rule, store a snapshot of the last developed ML model data. The Supported rule conditions include:

- For categorical columns, each category's percentage must be greater than the previous category percentage. For example:
  - Churn_rate.percentage.0 >= 10 OR Churn_rate.percentage.1 >= 20 (anycolumnName.percentage.anyCategoryBelongsToTheColumn)
  - Churn_rate.percentage.0 > 10 AND Churn_rate.percentage.1 > 20
- The total row count must be greater compared to the previous count. For example, anyColumnName.total.percentage >= 10
- A combination of the total row count and categorical columns, where each category's percentage must be greater than the previous percentage. For example, anyColumnName.total.percentage >= 10 AND (Churn_rate.percentage.0 >= 10 OR Churn_rate.percentage.1 >= 10)

Deriving variable

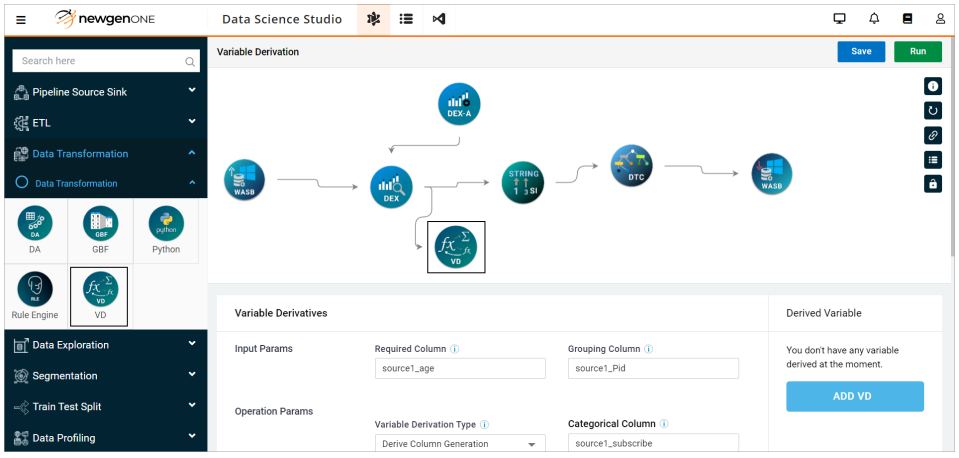
Variable Derivation (VD) allows you to create new variables from the existing variables using dates or without dates. It derives new columns by applying any of the available aggregation functions like sum, count distinct, mean, min, max, and more, and it can also make new variables from the variables you just created with the VD.

For example, you have data on all the transactions made by a person in a year.

1. Derive New features without using date — You can derive the total expenditure incurred by a person in a year.
2. Derive New features using date — You can derive total expenditure incurred by a person in a month, quarterly, every 4 months, and more. You can specify the date range as well as the interval.
3. Derive New features from previously generated VD features — You can derive the total expenditure incurred in the 1st quarter and 3rd quarter.

To define the input parameters for the VD node, perform the following steps:

1. Perform the steps 1 and 2 as mentioned in the Augmenting data section.
2. Drag the **VD node** from navigation pane and drop it on the canvas.
3. Click the **VD node**. The Variable derivatives section appear.



4. Specify the following parameters for the Variable Derivates:

Parameters	Description
Input Params	
Required Column	Select the required column.
Grouping Column	Select the required columns on which you want to perform the group by operation
Operation Params	
Variable Derivation Type	Keep the default value, Derived Column Generation.
Categorical Column	Select the categorical column on which you want to perform the group by function.
Operation Column	Select the column for which summarization is required in derived columns. Summarization can be done using one or more of sum, count, min, max, and average.
Date Params	
Derivation Type By Date	Select whether you want to derive the column with the date range or without the date range.
Date Column	Select the date column corresponding to the selected categorical column.
Date Format	Select the date format using the dropdown.
Interval Type	Select the interval time between the day or the month.
Start Date	Select a past date from which you want to fetch the new columns.
End Date	Select the date up to which the new columns must be derived.
Interval Duration	Enter the number in which you want to divide the derive column. For example, if you select n months in interval type and y as interval duration. Then, it creates n/y variables.
Is Last Bin Required	Select one of the following options: • Yes — Select Yes if you want data summarization that is not filtered within the specified date range. • No — Select No to keep this as default.

5. After specifying the above parameters, click **ADD VD**. The variable derivate gets added.

Developing a model

Model development involves creating, training, and refining a machine learning or statistical model to make predictions, classify data, or gain insights from data. It enables you to create a model. This chapter includes the following:

- Deep Learning
- Graph Analytics
- NLP
- Evaluator
- Machine Learning

Deep learning

Deep Learning is a crucial part of machine learning, and it operates through artificial neural networks (ANNs). The ANNs are also known as deep neural networks (DNNs). The concept of neural networks focuses on emulating the structure and functioning of the human brain, thereby helping in learning complex patterns and relationships within data.

To define properties for deep learning, perform the following steps:

1. Go to **Canvas**.
2. Click the **Model Development** on the navigation pane.
3. Select the **Deep Learning** using dropdown. The Multi-Layer Perceptron node appears.

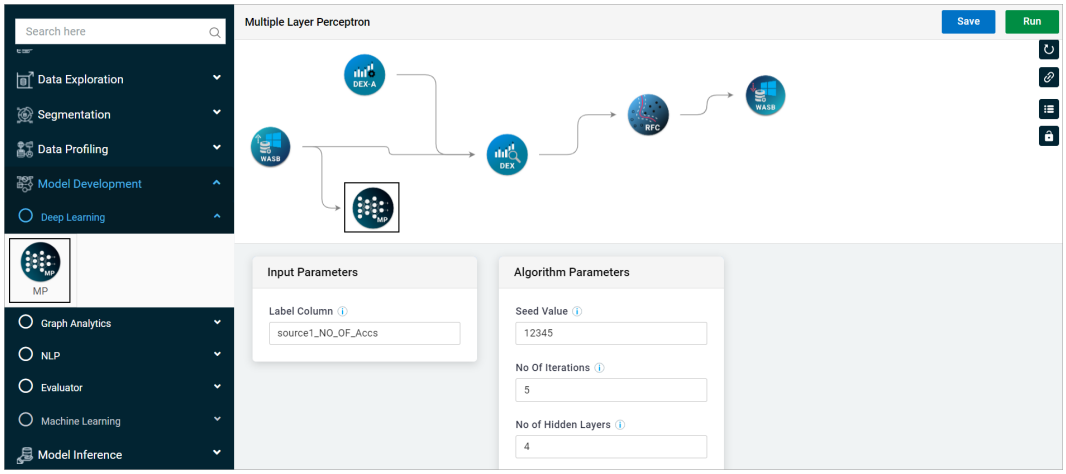
Multi-Layer perceptron

Multi-Layer Perceptron (MP) node is a type of feedforward neural network that consists of one or more hidden layers of nodes between the input and output layers. Each node

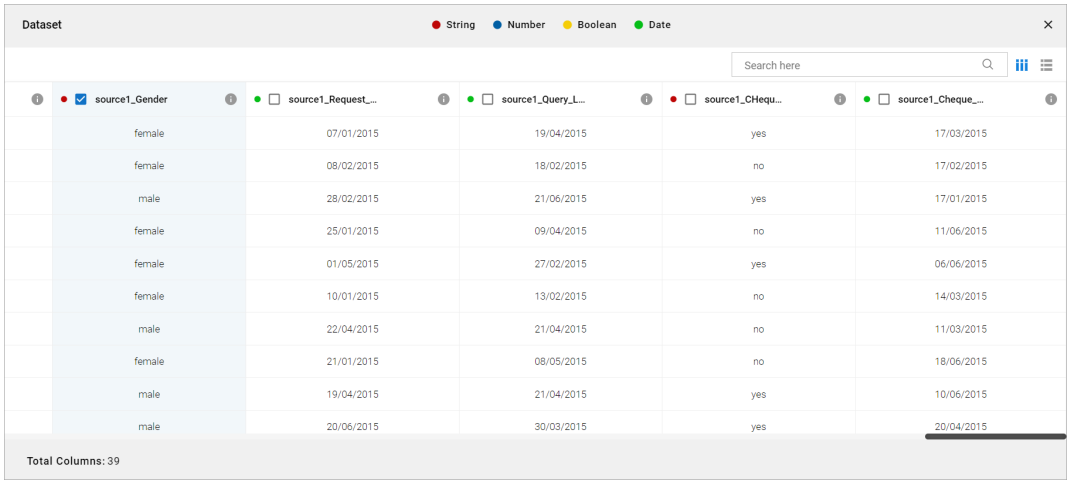
in the hidden layer receives input from all the nodes in the previous layer and generates output that is passed to all the nodes in the next layer.

To define the parameters of the MP Node, perform the following steps:

1. Perform steps from 1 to 3 as mentioned in the Deep learning section.
2. Drag the **MP** Node and drop it on the canvas.



3. Click the **MP** Node. The Input and Algorithm Parameters appear.
4. In the **Input Parameter** section, click **Label Column** textbox. The Dataset dialog appear.



5. Select the required **column** as label column. The Label Column contains the target variable or the variable predicted by the algorithm.

6. Specify the following possible values in the Alogrithm Parameters:

Algorithm Parameters	Description	Possible values
Seed Value	Determines the starting point for the random number generator.	Non negative integer value
No Of Iterations	Determines the quantity of weight updates that the MLP undergoes based on the training data.	Positive integer value (>=1)
No of Hidden Layers	Determines the depth of the neural network.	Positive integer value (>=1)
Tolerance Level	Determines the stopping criterion for MLP during training. Additionally, it represents the minimum improvement in the objective function that must be achieved between iterations to continue training.	>0
Perceptron Solver	Determines the optimization algorithm to update the weights in MLP during training.	• l-bfgs • gd
Block Size	Determines the number of samples used in each mini-batch during training. A mini-batch is a subset of the training data used to update the weights in MLP.	>=10
Perceptron Multiclass	Enables the process of handling multiclass classification problems.	• Yes • No
Condition Positive	Represents the number of positive instances in the training dataset and it calculates the performance metrics such as precision, recall, and F1-score. This field appears if you select **No** in the **Perceptron Multiclass**.	0 to 1

Algorithm Parameters	Description	Possible values
Condition Negative	Represents the number of negative instances in the training dataset and it calculates the performance metrics such as precision, recall, and F1-score. This field appears if you select **No** in the **Perceptron Multiclass**.	0 to 1

Graph analytics

Graph analytics is implementation of ML techniques to structure the data as graphs or networks. The graphs consists of vertices and connections which are used to represent relationships and dependencies among entities.

To define the properties for the graph analytics, perform the following steps:

1. Go to **Canvas**.
2. Click the **Model Development** on the navigation pane.
3. Click **Graph Analytics** using the dropdown. The following nodes appear.
   - PR
   - TC

Page Rank

The Page Rank (PR) determines the relative importance of vertices within a graph. By using hyperparameters such as the User Vertex Column, Maximum Iterations, and PageRank Probability, you can fine-tune the algorithm to better suit their specific needs. By understanding the workings of this algorithm, you can gain valuable insights into the structure and importance of complex networks.

To define the parameters of the PR node, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Graph Analytics section.
2. Drag the **PR** node and drop it on the canvas.
3. Click the **node**. The Input Parameters section appears.
4. In the Input Parameters, click **User Vertex Column** textbox. The Dataset dialog appears.

5. Select the desired column checkbox. The selected column specifies the vertex column in the graph used as the starting point for the PageRank calculation.
6. Enter the **positive integer value ≥1** in the Maximum Iterations textbox. It specifies the maximum number of iterations that the PageRank algorithm must run before terminating.
7. Enter the value from **0 to 1** in the PageRank Probability textbox. It specifies the probability that a user continue clicking on links from a given page.

Triangle Counting

The Triangle Counting (TC) node is designed to find triangles within a graph. A triangle consists of three nodes, where each node maintains a relationship with the other two. Triangle Counting determines the quantity of triangles that traverse through each vertex, offering a measure of clustering.

To define the input parameters of the TC node, perform the following steps:

1. Perform the steps from 1 to 3 as described in the Graph Analytics section.
2. Drag the **TC** node and drop it on the canvas.
3. Click the node. The Input Parameters section appears.
4. In the Input Parameters, click **User Vertex column**. The Dataset dialog appears.
5. Select the desired column checkbox. The selected column specifies the vertex column in the graph used as the starting point for the PageRank calculation.

Processing natural language

Natural Language Processing (NLP) is a field within artificial intelligence (AI) and linguistics. It facilitates interaction between computers and human language, enabling machines to generate human language in a meaningful and useful manner.

To define the properties for NLP, perform the following steps:

1. Go to **Canvas**.
2. Click the **Model Development** on the navigation pane.
3. Select the **NLP** using dropdown. The following nodes appear:
   - Lda
   - Word2Vec

Allocating latent dirichlet

Latent Dirichlet Allocation (LDA) classifies the text in a document into particular topics, making it an example of a topic model. It constructs a topic per document model and a words per topic model, both modeled as Dirichlet distributions.

To define the parameters of the LDA Node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Processing natural language section.
2. Drag the **LDA** Node and drop it on the canvas.
3. Click the **LDA** node. The Input and Algorithm Parameters appear.
4. In the Input Parameters, click **Index of Lda Column** textbox. The Dataset dialog appear.
5. Select the required column checkbox. The selected column specifies the output column containing the word vectors generated by the algorithm.
6. Specify the following Alogrithm Parameters:

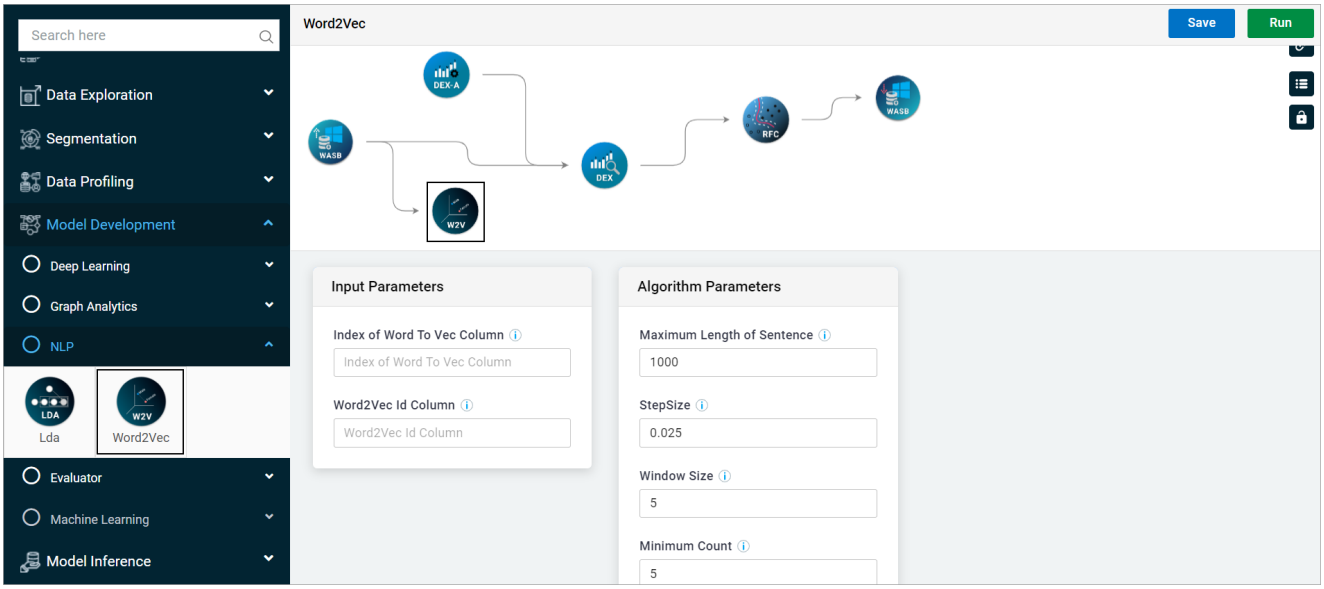
Alogrithm parameters	Description	Possible values
Max Iterations	Controls the maximum number of iterations that the algorithm performs before stopping them. A higher value might result in more accurate results, but also increase the processing time.	Positive integer value
Max Words	Sets the maximum number of words allowed in a single topic.	Positive integer value
Lda Optimizer	Determines the optimization method used to train the LDA model.	• em (Expectation Maximum) • online
Lda Learning Decay	Controls the Learning rate decay during training. A smaller value results in a slower but more stable learning process.	0,1
Lda Learning Offset	Determines the starting learning rate during training.	Positive integer value
Number of Topics	Determines the number of topics the model generates.	10≥

Word2Vec

Word2Vec generates word embeddings in NLP. This algorithm has several hyperparameters that can be tuned to optimize its performance.

To define the parameters of the Word2Vec Node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Processing natural language section.
2. Drag the **Word2Vec** Node and drop it on the canvas.
3. Click the **Word2Vec** node. The Input and Algorithm Parameters appear.



4. In the Input Parameters, click the **Index of Word To Vec Column** textbox. The Dataset dialog appear.
5. Select the required column checkbox. The selected column specifies the index of the input column containing the words. The input column must be of StringType and contain words as strings.
6. In the Input Parameters, click **Word2Vec Id Column** textbox. The Dataset dialog appear.
7. Select the required column checkbox. The selected column specifies the output column that holds the word vectors generated by the algorithm.

8. Specify the following Algorithm Parameters:

Algorithm parameters	Description	Possible values
Maximum Length of Sentence	Defines the maximum number of words that can be considered in a sentence.	Positive integer value
Stepsize	Defines the learning rate for the Word2Vec algorithm and specifies the amount of change in the weights of the model for each update. The default value is 0.025.	Positive float value
Windowsize	Specifies the maximum distance between the target and context words used for training the model. The default value is 5.	Positive integer value
Minimum Count	Specifies the minimum number of times a word must appear in the corpus for including it in the Word2Vec model. The default value is 5.	Positive integer value
Vector Size	Specifies the dimensionality of the word vectors generated by the model. The default value is 100.	Positive integer value
Maximum Iterations	Specifies the maximum number of iterations to run the Word2Vec algorithm. The default value is 5.	Positive integer value

Evaluator

An Evaluator refers to a performance metric or scoring function that measures and assesses the performance of a model or algorithm. Evaluators play a vital role in gauging the effectiveness of a machine learning model on a specific task and deciding whether the model is fit for deployment or needs additional enhancements.

To define properties for the Evaluator, perform the following steps:

1. Go to **Canvas**.
2. Click the **Model Development** on the navigation pane.

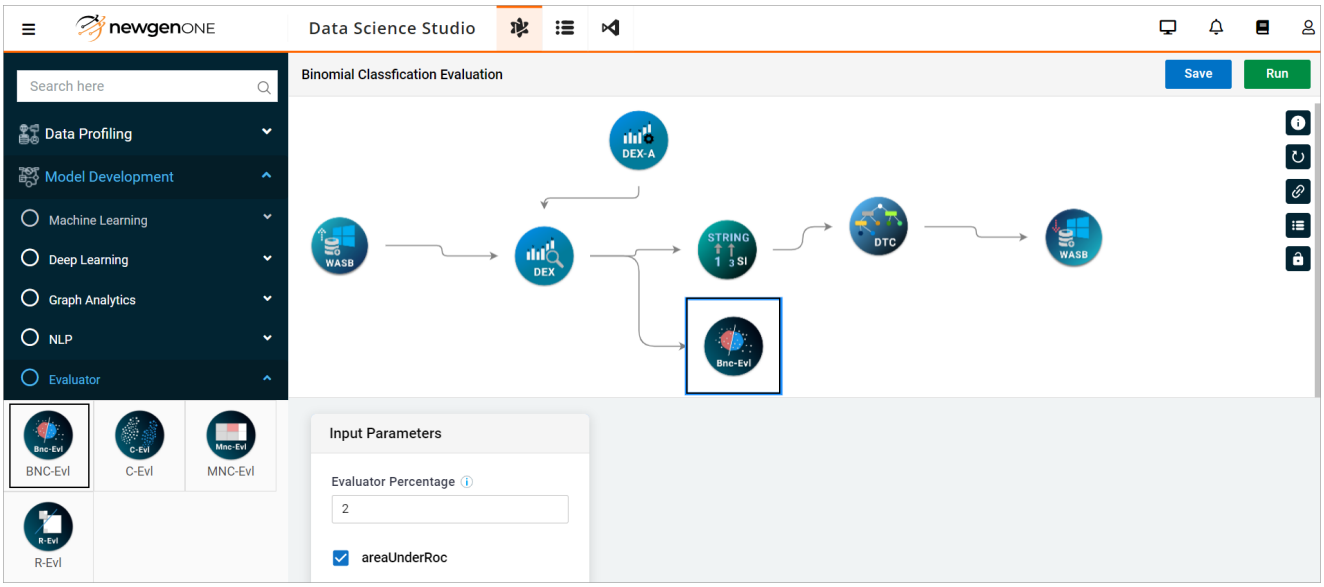
3. Select the **Evaluator** using dropdown. The following nodes appear:
   - BNC-Evl
   - C-Evl
   - R-Evl
   - MNC-Evl

Binomial classification evaluation

The Binomial Classification Evaluation (BNC-Evl) Node evaluates the performance of a binary classfication model.

To define the parameters of the BNC-Evl node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Evaluator section.
2. Drag the **BNC-Evl** node from the navigation pane and drop it on the canvas.
3. Click the **BNC-Evl** node. The Input Parameters appear.
4. Under the **Evaluator percentage** textbox, enter the **percentage** that you want to improve in the model performance and then replace it with the retained model.



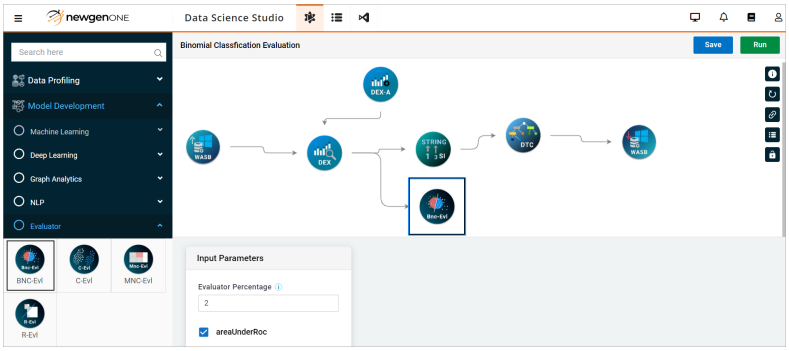
5. Select the following metrics checkbox as required:
    a. areaUnderRoc — Select the area under the receiver operating curve checkbox to use it in binary classification problems and specify the following:
        • First Time Eval Value — Enter the minimum threshold for corresponding metric.
        • Evaluator Weightage — Enter the evaluation weightage. It becomes the weightage of this metric.
    b. Accuracy — Select this checkbox to measure accuracy of the model and specify the First Time Eval Value and Evaluator Weightage fields.
    c. Precision — Select this checkbox to measure the proportion of true positive predictions among all instances predicted as positive by the model. Specify the First Time Eval Value and Evaluator Weightage fields.
    d. Recall — Select this checkbox to measure the proportion of the true positive predictions among all actual positive instances in the dataset. Specify the First Time Eval Value and Evaluator Weightage fields.
    e. areaUnderPR — Select the areaUnderPR (Area Under the Precision-Recall Curve) checkbox to use this in the scenarios where both precision and recall are crucial or while dealing with imbalanced datasets. Specify the First Time Eval Value and Evaluator Weightage fields.

Clustering evaluation

The Clustering Evaluation (C-Evl) Node evaluates the quality and effectiveness of clustering algorithms in unsupervised machine learning.

To define the parameters of the C-Evl node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Evaluator section.
2. Drag the **C-Evl** node from the navigation pane and drop it on the canvas.
3. Click the **C-Evl** node. The Input Parameters appears.



4. Under the **Evaluator percentage** textbox, enter the **percentage** that you want to improve in the model performance and then replace it with the retained model.
5. Select the silhouetteDistance checkbox to measure the performance of cluster algorithms and specify the First Time Eval Value and Evaluator Weightage fields.

Multinomial classification evaluation

The Multinomial Classification Evaluation (MNC-Evl) node evaluates the performance of a multi-class model.

To define the parameters of the MNC-Evl node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Evaluator section.
2. Drag the **MNC-Evl** node from the navigation pane and drop it on the canvas.
3. Click the **MNC-Evl** node. The Input Parameters appear.
4. Under the **Evaluator percentage** textbox, enter the **percentage** that you want to improve in the model performance and then replace it with the retained model.
5. Select the following checkboxes against required measure for evaluations:
    a. fmeasureMicroAvg — It is a F1-Measure Micro Average, a variant of the F1-Score that considers micro-averaging. Select the area under the receiver operating curve checkbox and specify the following:
        • First Time Eval Value — Enter the minimum threshold for corresponding metrics.
        • Evaluator Weightage — It's a weightage of the metric. Enter the percentage to evenly divide the evaluator.
    b. fmeasureMacroAvg — It is a F1-score Macro Average, a variant of the F1-score that considers macro-averaging. Select this checkbox to measure accuracy of the model. Then, specify the First Time Eval Value and Evaluator Weightage fields.

c. precisionMicroAvg — It is the metric to evaluate precision of a classification model. It is calculated by dividing the sum of true positives for all classes by the sum of the sums of true positives for all classes and the false positives for all classes. Select this checkbox to measure the proportion of true positive predictions among all instances predicted as positive by the model. Then, specify the First Time Eval Value and Evaluator Weightage fields.
d. precisionMacroAvg — It is the metric to evaluate precision of a classification model. It is the average (mean) of the sum of the precision values of each of the class in multi-class classification model. Select this checkbox to measure the proportion of the true positive predictions among all actual positive instances in the dataset. Specify the First Time Eval Value and Evaluator Weightage fields.
e. recallMicroAvg — It is the evaluation metrics to assess the sensitivity of a classification model. It is calculated by dividing the sum of true positives for all classes by the sum of sums of true positives for all classes and the false negatives for all classes. Select the areaUnderPR (Area Under the Precision-Recall Curve) checkbox to use this in the scenarios where both precision and recall are crucial or while dealing with imbalanced datasets. Specify the First Time Eval Value and Evaluator Weightage fields.
f. recallMacroAvg — It is the evaluation metric to assess the sensitivity of a classification model. It is the average (mean) of the sum of the recalls of the all classes. Select the areaUnderPR (Area Under the Precision-Recall Curve) checkbox to use this in the scenarios where both precision and recall are crucial or while dealing with imbalanced datasets. Specify the First Time Eval Value and Evaluator Weightage fields.
g. Accuracy — Determined by dividing the proportion of accurate predictions from the overall number of predictions. It provides an overall assessment of model correctness but might not be suitable for imbalanced datasets. Select the areaUnderPR (Area Under the Precision-Recall Curve) checkbox to use this in the scenarios where both precision and recall are crucial or while dealing with imbalanced datasets. Specify the First Time Eval Value and Evaluator Weightage fields.

Regression evaluation

The Regression Evaluation (R-Evl) node evaluates the performance and accuracy of a regression model.

To define the parameters of the R-Evl node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Evaluator section.
2. Drag the **R-Evl** node from the navigation pane and drop it on the canvas.
3. Click the **R-Evl** node. The Input Parameters appear.
4. Under the **Evaluator percentage** textbox, enter the **percentage** that you want to improve in the model performance and then replace it with the retained model.
5. Select the following metrics checkbox as required:
    a. RMSE — It is a root mean squared error. It calculates the square root of the average of the squared differences between predicted and actual values. Select the area under the receiver operating curve checkbox to use it in binary classification problems. The, specify the following:
    • First Time Eval Value — Enter the minimum threshold for corresponding metric.
    • Evaluator Weightage — It is the weightage of this metric. Enter the percentage to evenly divide the evaluator percentage.
    b. R-squared — The metric used to assess the goodness of fit of a regression model quantifies the proportion of the variance in the dependent variable (target) that the independent variables (features) in the model explain. Its values range from 0 to 1, with higher values indicating a better fit. Select this checkbox to measure accuracy of the model. Specify the First Time Eval Value and Evaluator Weightage fields.
    c. MAE — It is mean absolute error. It calculates the average of the absolute differences between predicted and actual value, making it less sensitive to outliers. Select this checkbox to measure the proportion of true positive predictions among all instances predicted as positive by the model. Then, specify the First Time Eval Value and Evaluator Weightage fields.

Machine learning

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models. It enables computers to learn and make predictions or decisions without getting explicitly programmed.

To define properties for machine learning, perform the following steps:

1. Go to **Canvas**.

2. Click the **Model Development** on the navigation pane.
3. Select the **Machine Learning** using dropdown. The following tabs appear:
   - Clustering
   - Collaborative Filtering
   - Time Series
   - FP Mining
   - Survival Analysis
   - Classification
   - Regression

Clustering

Clustering identifies data points that share certain feature characteristics among them. Then, it groups these data points together without knowing the labels of the data points. Clustering algorithms utilize various metrics to determine the similarity among the data points.

To define properties for clustering, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. Under the **Machine Learning**, select the **Clustering** using dropdown. The following nodes appear:
   - BI-K
   - GMM
   - K

Bisecting Kmeans

The Bisecting Kmeans (BI-K) modifies the K-Means algorithm which is a hybrid approach between partitional and hierarchical clustering. It begins with a single cluster containing all points in the model training set. Iteratively, it identifies divisible clusters at the bottom level and then bisects each of them using the k-means algorithm until there are a total of k leaf clusters or no leaf clusters are further divisible.

To define the parameters of the BI-K node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the clustering section.
2. Drag the **BI-K** node from the navigation pane and drop it on the canvas.

3. Click the **BI-K** node. The Algorithm Parameters appear.
4. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Bi Kmeans seed	Enter the value to ensure the reproducibility.
Number of leaf clusters	Enter the number of clusters greater than 1.
Number of Iterations	Enter the iteration number after which algorithm gets stopped.
Silhouette Distance Curve	Select one of the following options: • Yes — Select **Yes** to generate the silhouette curve in the output. • No — Select **No** to exclude the silhouette curve from the output.

Gaussian Mixed Models

The Gaussian Mixed Models (GMM) describes a composite distribution that involves selecting points from a set of k Gaussian sub- distributions, each having its distinct probability. The Gaussian mixture model operates as a probabilistic framework, assuming that data points emerge from a blend of finite Gaussian distributions with unspecified parameters. Basically, mixture models is as an extension of k- means clustering, incorporating details about both the data's covariance structure and the centers of the latent Gaussians.

To define the parameters of the GMM node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the clustering section.
2. Drag the **GMM** node from the navigation pane and drop it on the canvas.
3. Click the **GMM** node. The Algorithm Parameters appear.
4. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Seed Value	Enter the seed value for random number generation. The default value is 12345.

Algorithm parameters	Description
Number of Clusters	Enter a number that signifies the desired subpopulation count for the algorithm to identify, influencing clustering quality. Setting clusters too low can combine subpopulations, causing under-segmentation, while setting them too high can lead to over- segmentation, introducing noise and redundancy in results.
Number of Iterations	Enter a number that enables algorithm to iteratively adjust the Gaussian distribution parameters to maximize data likelihood within the model until convergence. Insufficient iterations might lead to suboptimal clustering, while excessive iterations might increase computation time despite reaching optimal parameters.

K-Means

K-means is a centroid based clustering algorithm that partitions data points into clusters based on their similarity of centroids.

The K-Means node solves the clustering problems in machine learning or data science. Here, K defines the number of pre- defined clusters to create in the process. K-means is a centroid-based algorithm, or a distance- based algorithm, which allows you to calculate the distances to assign a point to a cluster. In K- Means, each cluster is associated with a centroid.

To define the parameters of the K-Means node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the clustering section.
2. Drag the **K-Means** node from the navigation pane and drop it on the canvas.
3. Click the **K-Means** node. The Algorithm Parameters appear.
4. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Number of Clusters	Enter a number that signifies the desired subpopulation count for the algorithm to identify, influencing clustering quality. Setting clusters too low can combine subpopulations, causing under-segmentation, while setting them too high can lead to over-segmentation, introducing noise and redundancy in results.
Number of Iterations	Enter a number that enables algorithm to iteratively adjust the Gaussian distribution parameters to maximize data likelihood within the model until convergence. Insufficient iterations might lead to suboptimal clustering, while excessive iterations might increase computation time despite reaching optimal parameters.
Seed Value	Enter the seed value.
Silhouette Distance Curve	Select one of the following options: • Yes — Select **Yes** to generate the silhouette curve in the output. • No — Select **No** to exclude the silhouette curve from the output.

Collaborative filtering

Collaborative Filtering allows the generation of personalized predictions or recommendations based on the preferences of similar users or items. It is commonly used in scenarios involving a large dataset of users or items, with the aim of predicting how a user might rate or interact with an item they have not yet encountered.

To define properties for collaborative filtering, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. Under the **Machine Learning**, select the **Collaborative Filtering** using dropdown. The following nodes appear:
   • A
   • ALS

Affinity calculation

Affinity Calculation (A) involves the process of data mining and retrieves insightful correlations between different variables based on their co-occurrence within the feature values in the dataset. Additionally, it observes transaction behavior or patterns to identify affinity, which is then employed for recommendations using generalized Linear Regression.

To define the parameters for the A node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Collaborative Filtering section.
2. Drag the **A** node from the navigation pane and drop it on the canvas.
3. Click the **A** node. The Input and Algorithm Parameters appears.
4. Specify the following Input Parameters:

Input parameters	Description
Customer Column	The column name that includes the customer IDs. The possible values can be any string or numerical values that uniquely identify the customers or users.
Item Column	The column name that includes the item IDs. The possible values can be any string or numerical values that uniquely identify the items or products.
Label Column Name	The column name that includes the ratings or scores. The column including the rating or score values given the customers or users to the items of products. This field appears when you select the BinningValues in Affinity Calculation type.
Binning Operation	The operation type to perform on the ratings or scores. The possible values includes fixed width binning, adaptive binning, equal frequency binning, clustering-based binning, custom binning, temporal binning, attribute-based binning, and hybrid binning. This field appears when you select the BinningValues in Affinity Calculation type.

5. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Affinity Calculation type	Select one of the following: • BinningValues — Refers to the process of organizing and grouping data into bins. • ZerosAndOne — Represents the two possible states of a binary value. • FrequencyValues — Represents how many times a particular value, event, or observation occurs within a dataset.
The below parameters appears if you select the *BinningValues* in the Affinity Calculation type.	
Affinity logo	Represents the algorithm that calculates the affinity or similarity between feature values. Select the GeneralizedLinearRegression. It is an extension of traditional linear regression (LIR) model. It uses wider range of data distributions and relationships between the variables. It is normally used when the assumptions of classical linear regressions are not met.
Hyperparameter tuning	For tuning the model, select one of the following options: • Yes — Enables model tuning • No — Disables model tuning By default, the *Hyperparameter tuning* is set to No.
GLM Family Name	Specifies the conditional distribution for the response by selecting one the following distribution functions. • poisson — • gamma —
GLM max iteration value	Requires the maximum iteration value for generalized linear regression that helps to make attempts until the set limit. For non-hyperparameters tuning scenarios, it takes one value and for hyperparameters tuning scenarios, it takes an array separated by comma (,). Therefore, provide the value according to your requirement and it must be positive integer value >=1. By default, the *GLM max iteration value* is set to 10.

Algorithm parameters	Description
Normalization scale max value	Enter the maximum value for normalizing scale. It defines the upper limit for the scale of normalized range where the affinity or similarity values must be scaled. The value can be any positive integer.
Normalization scale min value	Enter the minimum value for normalizing scale. It defines the lower limit for the scale of normalized range where the affinity or similarity values must be scaled. The value can be any positive integer.
GLM Standardization	Process of standardizing the data or features that makes the convergence faster. To set the GLM standardization, select one of the following: • True — Enables the GLM Standardization • False — Disables the GLM Standardization By default, the *GLM Standardization* is set to *True*.
GLM reg param	Provide the value to regularize the GLM parameter. It indicates the Regularization parameter for L2 regularization term and that is **0.5*regParam*L2norm(coefficients)^2**. By default, the value is set to 1. The value must be greater than 0.
GLM Tolerance	Enter the value greater than 0. This specifies the convergence tolerance of iterations. A smaller value leads to higher accuracy at the cost of more iterations.
GLM FitIntercept	Select one of the following to calculate the intercept for this model. • True — Includes intercept in calculations • False — Does not includes intercept in calculation. For example, it is expected that data is centered. By default, the value is set to *True*.

Alternating least squares

Alternating Least Squares (ALS) algorithm allows you to solve overfitting issues in sparse data and increase prediction accuracy. You can also use it for predicting the user's ratings of different items.

The ALS algorithm divides the user-item interactions into two lower-dimensional metrics to predict the user-item interactions.

To define the parameters for the A node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Collaborative Filtering section.
2. Drag the **A** node from the navigation pane and drop it on the canvas.
3. Click the **A** node. The Input and Algorithm Parameters appears.
4. Specify the following Input Parameters:

Input parameters	Description
Als Item Id Col	Click and select the column from the dataset dialog. The selected column name contains the item ID in the input DataFrame.
Als User Column	Click and select the column from the dataset dialog. The selected column name contains the user ID in the input DataFrame.
Ratings Column	Click and select the column from the dataset dialog. The selected column name contains the rating values in the input DataFrame.

5. Specify the following Algorithm Parameters:

Algorithm parameters	Description
ALS Number Iterations	Enter any positive integer value as the number of iterations to run the ALS algorithm. By default, the value is set to 5.
ALS Number of Recommendations	Enter any positive integer value as the number of recommendations to generate for each customer. By default, the value is set to 4.
ALS Lambda	Enter any positive value from 0.01 to 1.0. The specified value is set for the regularization parameter that controls the extent of overfitting.By default, the value is set to 0.1.
ALS Rank	Enter any positive integer value. The specified value is set as the number of latent features that must be computed from observed features using matrix factorization. By default, the value is set to 30.

Algorithm parameters	Description
Minimum Rating	Enter the lowest possible score (positive integer value) that can be assigned to an item in the recommendation system. The default value is set to 1.
Maximum Rating	Enter the highest possible score (positive integer value) that can be assigned to an item in the recommendation system. The default value is set to 5.
Ignore used products	Specifies whether to ignore products that the customizer has previously used from the recommended products. • Yes — Select this to ignore used products • No — Select this to not ignore used products. By default, the ignore used products is set to No.

Time series

Time series analysis includes the data points collected over time and focuses on understanding and forecasting patterns, trends, and behaviors within the time sequence.

To define the properties for Time series, perform the following steps:

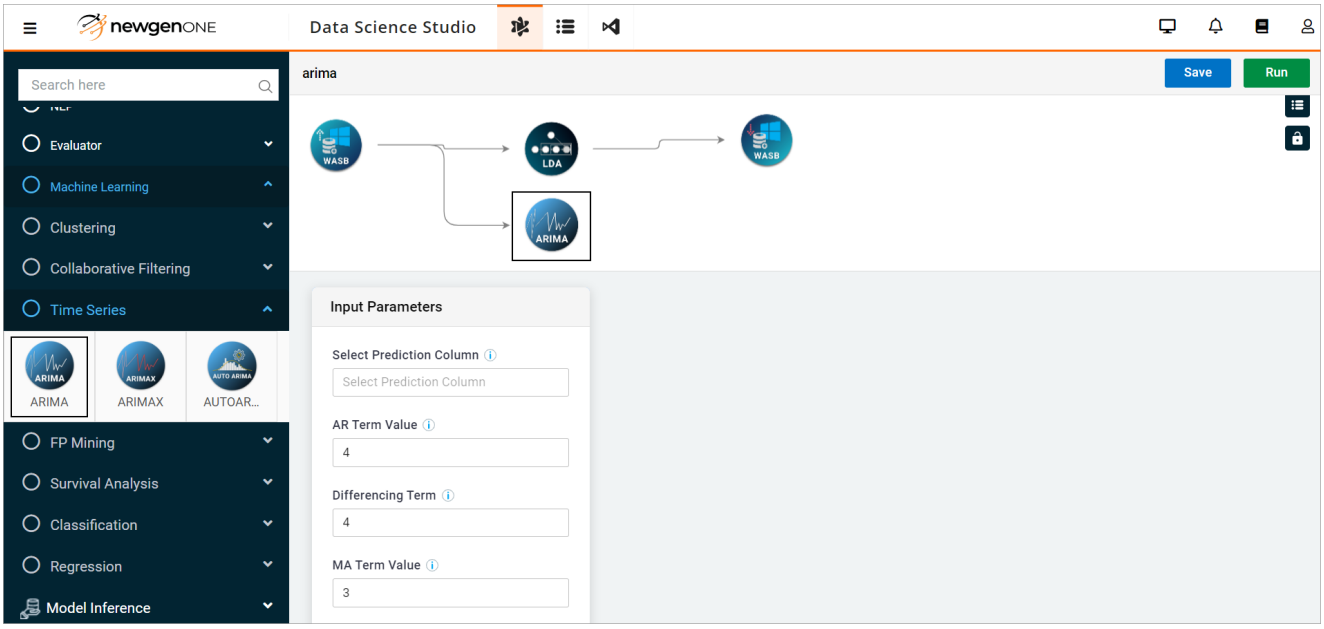
1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. Under the **Machine Learning**, select the **Time Series** using dropdown. The following nodes appear:
   • ARIMA
   • ARIMAX
   • AUTO ARIMA

Autoregressive integrated moving average

The ARIMA (Autoregressive Integrated Moving Average) model is a time series forecasting model that captures the relationship between a dependent variable and its previous values, along with the model's errors or residuals.

To define the parameters for the ARIMA node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Time Series section.
2. Drag the **ARIMA** node from navigation pane and drop it on the canvas.
3. Click the **ARIMA** node. The Input Parameters appears.



4. Click the Select Prediction Column textbox. The Dataset dialog appears.
5. Select the column for prediction of the time series data.
6. Enter the **positive integer value** in the **AR Term Value**. The AR Term value determines the number of lagged values of the time series to include in the model.
7. Enter the **non-negative integer value** in the **Differencing Term**. The Differencing Term determines the number of times the time series must be differenced to achieve stationarity.
8. Enter the **non-negative integer value** in the **MA Term Value**. The value of the moving average (MA) term is used in the model. It determines the number of lagged errors to include in the model.
9. Select one of the following options to include an intercept term in the model.
   - **Yes** — Includes an intercept term
   - **No** — Does not include an intercept term
10. Under the Arima Method, select one of the following options to use for optimization by minimizing and maximizing the objective function.
    - **lbfgs (Broyden-Fletcher-Goldfarb-Shanno)** — Estimates the parameters of a model.

- **cg (Conjugate Gradient)** — Estimates model parameters, particularly suited for large datasets.
- **nm (Nelder-Mead)** — Estimates model parameters, especially useful for nonlinear optimization problems.
- **powell** — Estimates model parameters, particularly for numerical optimization.

11. Enter the **positive integer** in the **Max Iterations**. The value becomes the maximum number of function evaluations.
12. Enter the **positive integer** in the **Number of Predictions**. The value becomes the number of time steps to predict into the future.

Autoregressive integrated moving average with eXogenous variables

The ARIMAX (AutoRegressive Integrated Moving Average with eXogenous variables) model is a time series forecasting model that enhances the ARIMA model by incorporating extra explanatory variables. It serves as a valuable tool for analyzing and predicting time series data that contain seasonal or trend components, while also integrating external factors that might impact the data.

To define the parameters for the ARIMAX node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Time Series section.
2. Drag the **ARIMAX** node from navigation pane and drop it on the canvas.
3. Click the **ARIMAX** node. The Input Parameters appears.
4. Click the **Select Prediction Column** textbox. The Dataset dialog appears.
5. Select the column for prediction of the time series data.
6. Enter the **positive integer value** in the **AR Term Value**. The AR Term value determines the number of lagged values of the time series to include in the model.
7. Enter the **non-negative integer value** in the **Differencing Term**. The Differencing Term determines the number of times the time series must be differenced to achieve stationarity.
8. Enter the **non-negative integer value** in the **MA Term Value**. The value of the moving average (MA) term is used in the model. It determines the number of lagged errors to include in the model.
9. Select one of the following options to include an intercept term in the model.

- **Yes** — Includes an intercept term
- **No** — Does not include an intercept term

10. In the Arima Method, select one of the following options to use for optimization by minimizing and maximizing the objective function.
    - lbfgs
    - cg
    - nm
    - powell
11. Enter the **positive integer** in the **Max Iterations**. The value becomes the maximum number of function evaluations.
12. Enter the **positive integer** in the **Number of Predictions**. The value becomes the number of time steps to predict into the future.
13. Click **Exogenous Columns** textbox. The Dataset dialog appears.
14. Select the column(s). The selected column(s) includes the exogenous variables in the model.

Automating ARIMA

The Auto ARIMA model automates the selection of optimal values for the ARIMA hyperparameters based on a given dataset, enabling time series forecasting.

To define the parameters for the ARIMAX node, perform the following steps:

1. Perform the steps from 1 to 3 mentioned in the Time Series section.
2. Drag the **ARIMAX** node from navigation pane and drop it on the canvas.
3. Click the **ARIMAX** node. The Input Parameters appears.
4. Specify the following Input Parameters:

Input Parameters	Description
Select Prediction Column	Click and select the column from the dataset dialog. The selected column work as prediction column that contains time series data.
Number Of Predictions	Enter the positive integer value. The selected value becomes the number of predictions in future.

Input Parameters	Description
Order of First Differencing (d)	Enter the positive integer value. In case *d* is set as *None* the value gets selected based on the test parameter results automatically.
Order of Time Lags (start_p)	Enter the positive integer value. It becomes the maximum order of the AR(p) term in the ARIMA model. For example, the defined value is the allowed limit to make predictions.
Order of Moving Average (start_q)	Enter the positive integer value. It becomes the maximum order of the MA(q) term in the ARIMA model.
Max Order of Lags (max_p)	Enter the positive integer value>=start_p. It becomes the maximum order of lags to consider in the PACF plot for the AR(p) term.
Max Order of Differencing (max_d)	Enter the positive integer value>=start_d. It becomes the maximum order of differencing to apply on the time series. For example, the defined value is the allowed time to make your data look right.
Max Order of Moving Average (max_q)	Enter the positive integer value>start_q. It becomes the maximum order of the MA(q) term in the ARIMA model.
Order of Seasonal Lag (start_P)	Enter the positive integer value. It becomes the maximum order of the seasonal AR(P) term in the ARIMA model.
Order of Seasonal Differencing (D)	Enter the positive integer value. It becomes the order of differencing to apply on the seasonal component of the time series. For example, the defined value is the allowed limit to rearrange the data to get a right data.
Period of Seasonal Differencing (m) *m* stands for append in the parameter name.	Enter the positive integer value for m. It becomes the period of the seasonal differencing of the time series. The default value is 1. For example, Company A receives more orders during the summer or festive season than in winter, and this pattern repeats on a yearly basis. Thus, the defined value, such as a year, represents the period it takes for this pattern to repeat itself.

Input Parameters	Description
Seasonal	Allows you to fit a seasonal Arima by selecting one of the following: • **True** — m must be greater than 1 • **False** — m is equal to seasonal
Test	Select the Kpss (kwiatkowski-Philips-Schmidt-Shin) using dropdown. This is the statistical test that determines the presence of seasonality in the time series. For example, this test helps to check how right the seasonal predictions at guessing the future.
Stepwise	Allows you to use a stepwise approach to select the optimal ARIMA hyperparameters. • **Yes** — Enables stepwise approach • **No** — Disables stepwise approach
Trend	Whether to include a trend in the ARIMA model. Enter the 1,1,0,1 in the textbox to include a trend (a+t+t^3) input in the ARIMA model. The default value is set to 1.

Mining frequent pattern

Frequent Pattern Mining (FP) enables the discovery of frequent itemsets or patterns from transactional databases or datasets containing sets of items.

To define properties of the Frequent pattern, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. In the **Machine Learning**, select the **FP Mining** using dropdown. The FPG node appears.

Frequent Pattern Growth

The Frequent Pattern Growth (FPG) discovers the frequent patterns in transactional datasets. The FPG is based on the Apriori algorithm and is known for its efficiency in handling large datasets.

To define parameters of FPG, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Mining Frequent Pattern section.
2. Drag the **FPG** node and drop it on canvas.
3. Click the **FPG** node. The Input and Algorithm Parameters section appears.
4. Specify the following Input Parameters:

Input parameters	Description
Transaction Columns	Allows you to select the column(s) that comprises the transaction data. • Click the Transaction Columns textbox. The Dataset dialog appears. • Select the required column(s) checkbox.
Delimiter	Enter the character to separate items in the transaction data. The valid characters are: • comma (,) • semi-colon (;) • colon (:) • space () • inverted comma (" ")

5. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Minimal Support	Enter the positive integer value or decimal value between 0 to 1. The set value is the minimum threshold value for the support of frequent itemsets. Itemsets with a support count greater than or equal to this threshold are considered frequent.
Number of Partitions	Enter the positive integer value. The set value is the number of partitions that are used in the computation of frequent itemsets. Increasing the number of partitions speed up the computation but also increases the memory requirements.

Survival analysis

Survival Analysis analyzes and conducts modeling of time-to-event data. It helps you identify the moment when a particular event takes place, such as the duration until a machine fails, the time until a patient recovers, the period until a customer churns, and more. This approach proves valuable in situations where you aim to examine the time-to-event aspect and the factors that impact the event's incidence.

To define properties for Survival Analysis, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. In the **Machine Learning**, select the **Survival Analysis** using dropdown. The AFT node appears.

Accelerated Failure Time Survival Regressor

The Accelerated Failure Time Survival Regressor (AFT) node is a regression model that predicts time-to-event for censored data.

To define parameters of AFT, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Survival Analysis section.
2. Drag the **AFT** node and drop it on canvas.
3. Click the **AFT** node. The Input and Algorithm Parameters section appears.
4. Specify the following Input Parameters:

Input parameters	Description
Label Column	Allows you to select the column index that becomes as label in input dataset for classification. • Click the Label Column textbox. The dataset dialog appears. • Select the required column(s) checkbox.
Common Censor Column	Allows you to select the common censor column that indicates that the event has occurred or not. • Click the Common Censor Column textbox. The dataset dialog appears. • Select the required column(s) checkbox.

5. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter tuning	Select one of the following options to specify whether to use the hyperparameter tuning method or not: • Yes • No The default value is No.
AFT Quantile Probability	Enter the value for quantile probabilities array. It can be any non negative value between 0 to 1 and the array must be non-empty.
AFT MaxIter	Enter the maximum iterations number. • For no hyperparameter tuning, it takes one value. • For Hyperparameter tuning, it takes array separated by comma (",").
AFT Tolerance	Allows you to set the convergence tolerance of iterations. A smaller value results in higher accuracy at the cost of increased iterations. • For no hyperparameter tuning scenario, it requires one value. • For hyperparameter tuning scenarios, it requires an array separated by commas (",").
AFT FitIntercept Value	Select one of the following options to calculate the intercept for this model • True — To use the intercept in calculations • False — To not use the intercept in calculations

Classification

Classification allows categorizes the data into discrete classes or categories based on labeled training examples.

To define properties for classification, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.

3. Under the **Machine Learning**, select the **Classification** using dropdown. The following nodes appear:
    - DT-C
    - NB-C
    - ENS-C
    - LR-C
    - SVM
    - RF-C
    - STK-C
    - GB-C

Decision tree classifier

The Decision Tree Classifier (DT-C) node is a supervised learning algorithm that solves problems using a tree representation. In this representation, each internal node of the tree corresponds to an attribute, while each leaf node corresponds to a class label.

To define parameters of DT-C, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **DT-C** node and drop it on canvas.
3. Click the **DT-C** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select an option to include multiple hyperparameters for enhancing the algorithm's performance. • **Yes** — Additional fields with the required metrics. • **No** — No additional fields with the required metrics. By default, the *Hyperparameter Tuning* is set to *No*.
Seed Value	Enter the positive integer value to generate a seed value for a random number. By default, the seed value is 12345.

Algorithm parameters	Description
Maximum Depth of Tree	Enter the positive integer value from 0 to 30 and tree grows upto the set depth. The default value is 5. In case the value is not specified, then tree continues splitting until all leaves are pure or until all leaves contain less than the minimum number of samples per leaf. Setting a maximum depth can help prevent overfitting and improve the model's generalization performance.
Maximum Number of Bins	Enter the positive integer value >=2and >= number of categories in categorical feature. The set value is used to split Continuous variables. The default value is 50. In this case, the DT-C algorithm finds the best split by computing the information gained from each possible bin. Thus,increasing the parameter value can enhance data information capture but can also lead to overfitting.
Minimum Information gain	Enter the non negative integer value. It is required for considering a split as valid. In case, the minimum information gain is less than the set threshold, then it does not performs the split. Increasing the parameter value can help prevent overfitting but can also lead to underfitting if the value is too high.
Is MultiClass	Select one of the following options to enable classifier training for multi-class classification: • **Yes** — Use this option when dealing with more than two classes. It employs a one-vs-all approach to train separate binary classifiers for each class label. • **No** — Use this option when dealing with only two classes. It does not employ a one-vs-all approach to train separate binary classifiers for each class label.

Algorithm parameters	Description
Metric Name	Select one of the following metric name to evaluate the model: 1. When *Is MultiClass* is set to *Yes*, then the available metric names are: • f1 • weightedPrecision • weightedRecall • accuracy 2. When *Is MultiClass* is set to *No*, then the available metric names are: • areaUnderROC • areaUnderPR This field appears if you select *Yes* in *Hyperparameter Tuning*.
Impurity Method	Select one of the following options to use impurity measure for computing the information gain. By default, the *Impurity Method* is set to *gini*. • gini • entropy
Minimum Instances Per Node	Enter a positive integer value greater than or equal to 1. It is required to split a node. In case a node has fewer instances than this threshold, it becomes a leaf node. Setting a higher value can help prevent overfitting. The default value is 1.
Condition Positive	Enter either 0 or 1 for a condition positive value. It calculates the precision, recall, and F1 score for the classifier with the positive class label. The default value is 0. This field appears if you select *No* in the *Is MultiClass*.
Condition Negative	Enter either 0 or 1 for a negative value. It calculates the precision, recall, and F1 score for the classifier with negative class label. The default value is 1. This field appears if you select *No* in the *Is MultiClass*.

Logistic regression classifier

The Logistic Regression Classifier (LR-C) node is a supervised machine learning algorithm used for solving binary and multiclass classification problems. It models the probability of the outcome of a binary or multiclass dependent variable based on one or more independent variables. The model uses a logistic function to map the input features to the probability of the output class, and it estimates the parameters of the function using the maximum likelihood method.

To define parameters of LR-C, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **LR-C** node and drop it on canvas.
3. Click the **LR-C** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to use the optimal hyperparameters of the model. By default, the *Hyperparameter Tuning* is set to *No*. • Yes • No
Is MultiClass	Select one of the following options to identify whether the problem is multiclass classification. By default, the *Is MultiClass* is set to *No*. • Yes — Model use a multiclass logistic regression algorithm • No — Model does not use a multiclass logistic regression algorithm

Algorithm parameters	Description
Evaluation Metric Name	Select one of the options to represent the logistic regression evaluation metric name for hyperparameter tuning: 1. When *Is MultiClass* is set to *Yes*, then the available metric names are: • f1 • weightedPrecision • weightedRecall • accuracy By default, the *Evaluation Metric Name* is *f1*. 2. When *Is MultiClass* is set to *No*, then the available metric names are: • areaUnderROC • areaUnderPR • By default, the *Evaluation Metric Name* is *areaUnderROC*. • The *Evaluation Metric Name* field appears if you select *Yes* in *Hyperparameter Tuning*.
Elastic Params	Allows you to control the trade-off between L1 (Lasso) and L2 (Ridge) regularization. The default value is 1.0. The possible values are: • 0 — L2 Penalty • 1 — L1 Penalty • 0 and 1 — Combination of L1 and L2
Tolerance Value	Enter the positive float value for stopping criteria. If the change in the objective function between iterations is below 0, then the model stops iterating. The default value is 1E-6.
MaxIter Parameter	Enter the value greater than 0. The specified values becomes the maximum number of iterations for the optimization algorithm. The default value is 25.
Fit Intercept Parameter	Select one of the following options to fit an intercept. By default, the *Fit Intercept Parameter* is set to *True*. • True — Fits an itercept term • False — Does not fits an intercept term

Algorithm parameters	Description
Standardization Parameter	Select one of the following options to standardize the input variables. By default, the *Standardization Parameter* is set to *True*. • True • False
Regularization Parameter	Enter the positive float value to regularize the parameters for this model. The default value is 0.01. • For scenarios without hyperparameter tuning, it requires one value. • For hyperparameter tuning scenarios, it requires an array separated by commas.
Condition Positive	Enter either 0 or 1 for a condition positive value. It calculates the precision, recall, and F1 score for the classifier with the positive class label. The default value is 0. This field appears if you select *No* in the *Is MultiClass*.
Condition Negative	Enter either 0 or 1 for a negative value. It calculates the precision, recall, and F1 score for the classifier with the negative class label. The default value is 1. This field appears if you select *No* in the *Is MultiClass*.

Stacking classifier

The Stacking Classifier (STK-C) node involves stacking the outputs of individual classifiers and computing the final prediction using another classifier. The predictions from the other models serve as inputs for the stacking model.

To define parameters of STK-C, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **STK-C** node and drop it on canvas.
3. Click the **STK-C** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.

6. In the Algorithm Parameter section, select one of the following options as the base classifier for the Stacking Classifier:
   • GradientBoostingTreeClassification
   • RandomForestClassification
   • DecisionTreeClassification

Ensemble classifier

Ensemble Classifier (ENS-C) node is a machine learning technique that combines the predictions of multiple individual classifiers (base models) to make more accurate and robust predictions than any single classifier can achieve on its own. The idea behind an ensemble methods is to exploit the diversity and collective intelligence of the individual classifiers to improve robustness and overall performance.

To define parameters of ENS-C, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **ENS-C** node and drop it on canvas.
3. Click the **ENS-C** node. The Input Parameters section appears.
4. In the **Ensembling Type for Serving** field, select one of the following options to ensemble the probability columns of different models:
   • max
   • avg

Gradient boosting classifier

The Gradient Boosting Classifier (GB-C) is an ensemble method of machine learning that combines multiple weak classifiers into a single strong classifier. It adds decision trees to a model sequentially and each tree corrects the errors done by the previous tress simultaneously. This algorithm optimizes a loss function by adjusting both the weights of data points and the parameters of the tree. Hence, the final prediction is a weighted sum of the predictions of all the trees.

To define parameters of GB-C, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.

2. Drag the **GB-C** node and drop it on canvas.
3. Click the **GB-C** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Seed Value	Enter the positive integer value to initialize the random number generator, which can affect algorithm results. Setting a fixed seed value ensures that algorithm results are reproducible across different runs. By default, the seed value is 12345.
Number of Iterations	Enter the positive integer value greater than 1 to set the number of iterations (number of trees) in the model. The default value is 2. Specifying this parameter controls the model complexity and avoids overfitting.
Maximum Depth of Tree	Enter the positive integer value from 0 to 30 to specify the maximum depth of the gradient boosting tree. The depth of a tree represents the number of splits it makes before reaching a leaf node. The default value is 5.
Maximum Number of Bins	Enter the positive integer value >=2 which is equal to number of categories in any categorical feature. The set value becomes the maximum number of bins (maximum number of intervals) that can be created for each numeric feature during the split process. This parameter controls the split granularity and helps to avoid overfitting. The default value is 50.
Minimum Information Gain	Enter a non negative float value. The set value becomes the minimum information gain which is the threshold for the split decision. The splits with information gain greater than this set value are allowed. Additionally, this parameter controls the split quality and avoids overfitting. The default value is 0.01.

Algorithm parameters	Description
Minimum Instance per Node	Enter a positive integer value greater than or equal to 1. It is required for creating a leaf node. Additionally, this parameter controls the generalization power of the model and avoids overfitting. The default value is 4.
Sub Sampling Rate	Enter the float value ranging from 0.0 to 1.0 to set the sub sampling rate. This rate fraction of data points is used to train each tree. The default value is 0.1.
Feature Subset Strategy	Select one of the following options to select the features used for each tree: • auto • all • sqrt • log2 • onethird The default value is set to all.
Condition Positive	Sspecify the positive class label for binary classification problems. It is used to compute the precision, recall, and F1 score for the classifier. It can be 0 or 1. The default value is 0.
Condition Negative	Specify the negative class label for binary classification problems. It is used to compute the precision, recall, and F1 score for the classifier. It can be 0 or 1. The default value is 1.

Naïve bayes classifier

Naïve Bayes classifiers (NBC) node are a family of supervised learning algorithms that use Bayes' theorem to classify data. They are called naïve because they make the assumption that the features of a data point are independent of each other.

To define parameters of NBC, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **NBC** node and drop it on canvas.
3. Click the **NBC** node. The Input and Algorithm Parameters section appears.

4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to use the optimal hyperparameters of the model. By default, the *Hyperparameter Tuning* is set to *No*. • Yes • No
Lambda	Enter the value from 0.01 to 1.0 to avoid zero-frequency problems in the calculation of conditional probabilities. The default value is 1.0.
Model Type	Select one of the following model type to use in Naïve Bayes algorithm for classification: • multinomial — Assumes that the features are discrete counts. For example, word frequency in text classification. • bernoulli — Use this option when feature vectors are binary (Boolean) values. For example, presence or absence of a word in text classification.
Naive Bayes MultiClass	Select one of the following options to classify the problems with more than two classes. By default, *Naive Bayes MultiClass* is set to *No*. • Yes — Use this option to select Multinomial classification • No — Use this option to select Binomial classification This parameter appears if you select the *multinomial* in the *Model Type*.

Algorithm parameters	Description
Metric Name	Select one of the options to specify the metric name for model evaluation: 1. When *Naive Bayes MultiClass* is set to *Yes*, then the available metric names are: • f1 • weightedPrecision • weightedRecall • accuracy 2. When *Naive Bayes MultiClass* is set to *No*, then the available metric names are: • areaUnderROC • areaUnderPR This field is available in case of hyperparameter tuning.
Condition Positive	Enter either 0 or 1 for a condition positive value. It calculates the precision, recall, and F1 score for the classifier with the positive class label. The default value is 0. This field appears if you select *No* in the *Naive Bayes MultiClass*.
Condition Negative	Enter either 0 or 1 for a negative value. It calculates the precision, recall, and F1 score for the classifier with the negative class label. The default value is 1. This field appears if you select *No* in the *Naive Bayes MultiClass*.

Support vector machine

A Support Vector Machine (SVM) node performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are called support vectors. This algorithm handles both linear and non-linear decision boundaries.

To define parameters of SVM, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.
2. Drag the **SVM** node and drop it on canvas.
3. Click the **SVM** node. The Input and Algorithm Parameters section appears.

4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Number of Iterations	Enter the integer value greater than 0. It is set to determine that how many times the algorithm updates the model parameters. The default value is 5.
Step Size for Gradient Descent	Enter the value greater than or equal to 1. It is set to determine the step size taken during the optimization process. The default value is 1.
Regularization Value	Enter the float value greater than 0. It is set to control the overfitting in the model. The default value is 0.01.
Svm Min BatchFraction	Enter the float value between 0 to 1. It is set as the minimum fraction of data that must be used in each iteration of the algorithm. The default value is 0.3
Condition Positive	Enter either 0 or 1 for a condition positive value. It calculates the precision, recall, and F1 score for the classifier with the positive class label. The default value is 0.
Condition Negative	Enter either 0 or 1 for a negative value. It calculates the precision, recall, and F1 score for the classifier with the negative class label. The default value is 1.

Random forest classifier

The Random forest classifier (RF-C) is a supervised learning algorithm that uses multiple decision trees to make predictions. Each decision tree is trained on a different subset of the training data, which helps to reduce overfitting and improve the accuracy of the model. The predictions from each decision tree are then combined to generate a final prediction.

To define parameters of SVM, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Classification section.

2. Drag the **SVM** node and drop it on canvas.
3. Click the **SVM** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to use the optimal hyperparameters of the model. By default, the *Hyperparameter Tuning* is set to *No*. • Yes • No
Seed Value	Enter the positive integer value to generate a seed value for the algorithm and ensures that the algorithm produces the same results each time it runs. By default, the seed value is 12345.
Maximum Depth of Tree	Enter the positive integer value for maximum depth of the decision trees in the forest. The default value is 5.
Maximum Number of Bins	Enter the positive integer value. The set value becomes the maximum number of bins to use when discretizing continuous features. The default value is 50.
Impurity Method	Select one of the following options to use impurity measure for computing the information gain. By default, the *Impurity Method* is set to *gini*. • gini • entropy
Minimum Information Gain	Enter the non negative float value to specify the minimum information gain required for decision tree splits. The default value is 0.01.
Number of Trees	Enter a positive integer value ≥ 1 to specify the number of decision trees to build in the forest. The default value is 3.
Minimum Instance per Node	Enter a positive integer value ≥ 1 to specify the minimum number of instances required for a decision tree node to split. The default value is 4.

Algorithm parameters	Description
Sub Sampling rate	Enter the value from 0 to 1. The set value becomes the fraction of data to use for each tree in the forest. The default value is 0.1.
Is MultiClass	Select one of the following options to identify whether the problem is multiclass classification. By default, the *Is MultiClass* is set to *No*. • **Yes** — Model use a multiclass random forest classifier algorithm • **No** — Model does not use a multiclass random forest classifier algorithm
Metric Name	Select one of the options to specify the metric name for model evaluation: 1. When *Is MultiClass* is set to *Yes*, then the available metric names are: • f1 • weightedPrecision • weightedRecall • accuracy By default, the *Evaluation Metric Name* is *f1*. 2. When *Is MultiClass* is set to *No*, then the available metric names are: • areaUnderROC • areaUnderPR • By default, the *Evaluation Metric Name* is *areaUnderROC*. • The *Evaluation Metric Name* field appears if you select *Yes* in *Hyperparameter Tuning*.
FeatureSubset Strategy	Select one of the following options to specify the strategy for feature selection in each tree with the forest. The default value is all. • auto • all • sqrt • log2 • onethird

Algorithm parameters	Description
Condition Positive	Enter either 0 or 1 for a condition positive value. It calculates the precision, recall, and F1 score for the classifier with the positive class label. The default value is 0. This field appears if you select *No* in the *Is MultiClass*.
Condition Negative	Enter either 0 or 1 for a negative value. It calculates the precision, recall, and F1 score for the classifier with the negative class label. The default value is 1. This field appears if you select *No* in the *Is MultiClass*.

Regression

Regression is a supervised machine learning technique. It predicts continuous numerical values and establishes a relationship between input features (independent variables) and a target output (dependent variable) by learning from labeled training data.

To define properties for Regression, perform the following steps:

1. Go to **Canvas**.
2. Go to **Model Development** on the navigation pane.
3. Under the **Machine Learning**, select the **Regression** using dropdown. The following nodes appear:
    - DT-R
    - LIR
    - ENS-R
    - GLM
    - GB-R
    - STK-R
    - RF-R

Decision tree regressor

A Decision Tree Regressor (DT-R) node is a type of machine learning algorithm that is used for regression problems. It builds a tree-like model of decisions and their predicted outcomes, based on the features of the input data.

To define parameters of DT-R, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **DT-R** node and drop it on canvas.
3. Click the **DT-R** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to include several hyperparameters that you can tune to improve the algorithm's performance. By default, the *Hyperparameter Tuning* is set to *No*. • **Yes** — Includes the additional fields with the required metrics. • **No** — Does not include the additional fields with the required metrics.
Seed Value	Enter the random positive integer value to initialize the random number generator, which can affect algorithm results. Setting a fixed seed value ensures that algorithm results are reproducible across different runs. By default, the seed value is 12345.
Maximum Depth of Tree	Enter the positive integer value to specify the maximum depth of the decision tree. A deeper tree can model more complex relationship between input features and Target variables but might lead to overfitting. The default value is 5.
Maximum Number of Bins	Enter the positive integer value >=1 to specify the maximum number of bins that can be used for each feature. A larger number of bins can capture more information about the feature, but can also increase the tree complexity. The default value is 5.

Algorithm parameters	Description
Minimum Information Gain	Enter a non negative float value. The set value becomes the minimum information gain required for a split. Information gain is a measure of how much a particular split reduces the uncertainty in the target variable. A higher minimum information gain can lead to a simpler tree with fewer splits, but might also result in lower predictive accuracy. The default value is 0.01.
Metric Name	Select one of the following metric name to evaluate the model: • mse • rmse • R-Squared • MAE (Mean Absolute Error) This field appears if you select *Yes* in *Hyperparameter Tuning*.
Impurity Method	Select variance to calculate the impurity of a split.
Minimum Instances Per Node	Enter a positive integer value greater than 1. It is required to split a node. On split, each of the children nodes in the tree must have atleast the entered value in this field. A higher minimum number of instances can lead to a simpler tree with fewer splits but might result in lower predictive accuracy. The default value is 1.

Generalized linear regressor

The Generalized Linear Regressor (GLM) node is a type of regression algorithm that extends the ordinary linear regression model to handle non-normal response variables, such as binary, count, or continuous data with non-constant variance. The algorithm uses a link function to map the linear predictor to the response variable and a probability distribution function to model the variability of the response variable.

To define parameters of GLM, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **GLM** node and drop it on canvas.

3. Click the **GLM** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to include several hyperparameters that you can tune to improve the algorithm's performance. By default, the *Hyperparameter Tuning* is set to *No*. • **Yes** — Includes the additional fields with the required metrics. • **No** — Does not include the additional fields with the required metrics.
Family Name	Select one of the following options as the name of the probablity distribution function. It is used to model the response variable. By default, the *Family Name* is *gamma*. • poisson • gamma
Link Name	Select one of the following options as the name of the link function. It maps the linear predictor to the response variable. • identity • log
GLM Evaluation Metric Name	Select one of the following metric name to evaluate the model: • rmse • R-Squared • MAE (Mean Absolute Error) This field appears if you select *Yes* in *Hyperparameter Tuning*.
GLM Max Iteration Value	Enter the integer value greater than 0. It takes a single value, but for hyperparameter tuning scenarios, it takes an array separated by commas (,). The default value is 10.

Algorithm parameters	Description
GLM Standardization	Select one of the following options to standardize the input variables with a mean of zero and a standard deviation. This helps the algorithm to converge faster and improve performance. By default, the *GLM Standardization* is set to *True*. • True • False
GLM reg param	Enter the positive integer value greater than 0. The default value is 1. Setting this parameter controls the strength of the penalty term in the objective function, preventing overfitting, and improving the model's generalization performance.
GLM Tolerance	Enter the positive float value to determine the stopping criterion for the solver. If the change in the objective function is below this value, the solver considers the optimization problem to be converged. The default value is 1E-6.
GLM FitIntercept	Select one of the following options to fit the intercept term in the linear model. By default, the *GLM FitIntercept* is set to *False*. • True — Algorithm fits the intercept term • False — Intercept term is set to zero

Stacking regressor

Stacking Regressor (STK-R) node is an ensemble learning technique used in machine learning. It combines multiple regression models to improve the overall predictive performance. The basic idea behind stacking is to use the predictions from several base regression models as inputs to a higher-level regression model, called a meta-regressor, which then outputs the final prediction.

To define parameters of STK-R, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **STK-R** node and drop it on canvas.

3. Click the **STK-R** node. The Input and Algorithm Parameters section appears.
4. In the **Input Parameters** section, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. In the **Algorithm Parameter** section, select one of the following options as the base regressor for the Stacking Regressor
   - GradientBoostedRegression
   - RandomForestRegression
   - DecisionTreeRegression

Ensemble regressor

The Ensemble Regressor (ENS-R) node combines the predictions of multiple individual regression models (base models) to make more accurate and robust predictions than any single model might achieve on its own.

To define parameters of ENS-R, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **ENS-R** node and drop it on canvas.
3. Click the **ENS-R** node. The Input Parameters section appears.
4. In the **Ensembling Type for Serving** field, select one of the following options to ensemble the probability columns of different models:
   - max
   - avg

Linear regressor

Linear Regressor (LIR) node predicts the value of a dependent variable (y) based on a given independent variable (x). Thus, this regression technique identifies a linear relationship between the input x and the output y.
Linear Regression is a statistical approach used for modeling a relationship between a scalar response and one or more explanatory variables, also known as dependent and independent variables. The objective of linear regression is to minimize the sum of squared errors between the predicted and actual values.

To define parameters of LIR, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **LIR** node and drop it on canvas.
3. Click the **LIR** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to include several hyperparameters that you can tune to improve the algorithm's performance. By default, the *Hyperparameter Tuning* is set to *No*. • **Yes** — Includes the additional fields with the required metrics. • **No** — Does not include the additional fields with the required metrics.
Max Iteration Value	Enter the maximum number of iterations for the solver to converge. The solver iterates until the convergence criterion is met or this number of iterations is reached. The default value is 25.
Elastic Net	Elastic net is a regularization method that combines the L1 and L2 penalties of the Lasso and Ridge regression methods. It adds a hyperparameter to balance the contributions of L1 and L2 penalties. Enter the value from 0 to 1 to combine the L1 and L2 priors as regularizer. • 0 — L2 penalty • 1 — L1 penalty • 0 to 1 — L1 and L2 penalty combination The default value is set to 1.0.

Algorithm parameters	Description
Evaluation Metric Name	Select one of the following metric name to evaluate the model. By default, the *Evaluation Metric Name* is set to *r2*. • rmse • R-Squared • MAE (Mean Absolute Error) This field appears if you select *Yes* in *Hyperparameter Tuning*.
Standardization Value	Select one of the following options to standardize the input variables with a mean of zero and a standard deviation of one. This helps the algorithm to converge faster and improve performance. By default, the *Standardization value* is set to *True*. • True • False
Reg Param Value	Enter the any non negative integer value between 0 to 1. The default value is 0.3. Setting this parameter controls the strength of the penalty term in the objective function, preventing overfitting, and improving the model's generalization performance.
Solver Value	Select one of the following options to use the algorithm in the optimization problem. By default, the *Solver Value* is set to *l-bfgs*. • l-bfgs • normal • auto
Tolerance Value	Enter the positive float value to determine the stopping criterion for the solver. If the change in the objective function is below this value, the solver considers the optimization problem to be converged. The default value is 1E-6.
Fit Intercept Parameter	Select one of the following options to fit the intercept term in the linear model. By default, the *Fit Intercept* is set to *False*. • True — Algorithm fits the intercept term • False — No Intercept term is used

Gradient boosting regressor

Gradient Boosting Regressor (GB-R) node builds a decision tree in a step-by-step process where each new tree tries to correct the errors made by the previous trees. The process first builds a simple decision tree on the training data, and then calculating the errors between the predicted values and the actual values. A second decision tree is then built to correct the errors made by the first tree, and more.

To define parameters of GB-R, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **GB-R** node and drop it on canvas.
3. Click the **GB-R** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

Algorithm parameters	Description
Seed Value	Enter the positive integer value to initialize the random number generator, which can affect algorithm results. Setting a fixed seed value ensures that algorithm results are reproducible across different runs. By default, the seed value is 12345.
Number of Iterations	Enter the integer value greater than 1 to set the number of iterations (number of trees) in the model. The default value is 2. Specifying this parameter controls the model complexity and avoids overfitting.
Maximum Depth of Tree	Enter the positive integer value to specify the maximum depth of the decision tree. A deeper can model more complex relationship between input features and target variables but might lead to overfitting. The default value is 5.

Algorithm parameters	Description
Maximum Number of Bins	Enter the positive integer value >=1 which is equal to number of categories in any categorical feature. The set value becomes the maximum number of bins that can be used for each feature. A larger number of bins can capture more information about the feature, but can also increase the tree complexity. The default value is 5.
Minimum Information Gain	Enter any non negative float value. The set value becomes the minimum information gain required for a split. Information gain is a measure of how much a particular split reduces the uncertainty in the target variable. A higher minimum information gain can lead to a simpler tree with fewer splits, but might also result in lower predictive accuracy. The default value is 0.01.
Minimum Instance per Node	Enter a positive integer value greater than 1. It is required to split a node. A higher minimum number of instances can lead to a simpler tree with fewer splits but might result in lower predictive accuracy. The default value is 4.
Sub Sampling Rate	Enter the value between 0 to 1 to set the sub sampling rate (feature sampling rate). It controls the fraction of features and trains each decision tree in the forest. The default value is 0.1.
Feature Subset Strategy	Select one of the following options to reduce the correlation between the trees in the forest: • all • sqrt • log2 • onethird

Random forest regressor

Random Forest Regressor (RF-R) node is a Supervised learning algorithm that uses an ensemble learning method for regression. It uses multiple decision tree regressors to create models with improved accuracy. These decision trees are trained on distinct

subsets of the training data. The predictions from each of these decision trees are then combined to produce the final prediction.

To define parameters of RF-R, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Regression section.
2. Drag the **RF-R** node and drop it on canvas.
3. Click the **RF-R** node. The Input and Algorithm Parameters section appears.
4. In the Input Parameters, click the **Label Column** textbox. The dataset dialog appears.
5. Select the required column checkbox.
6. Specify the following Algorithm Parameters:

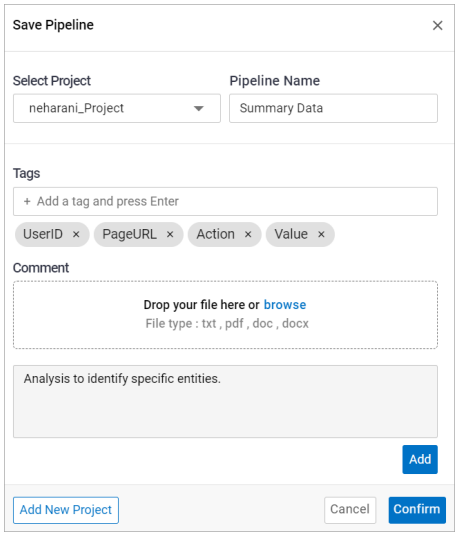
Algorithm parameters	Description
Hyperparameter Tuning	Select one of the following options to include several hyperparameters that you can tune to improve the algorithm's performance. By default, the *Hyperparameter Tuning* is set to *No*. • **Yes** — Includes the additional fields with the required metrics. • **No** — Does not include the additional fields with the required metrics.
Seed Value	Enter the random positive integer value to initialize the random number generator, which can affect algorithm results. Setting a fixed seed value ensures that algorithm results are reproducible across different runs. By default, the seed value is 12345.
Metric Name	Select one of the following metric name to evaluate the model: • mse • rmse • r2 • mae This field appears if you select *Yes* in *Hyperparameter Tuning*.
Number of Trees	Enter the positive integer value greater than or equal to 1. The set value become the limit for creating trees for random forest regression. By default, the Number of Trees is set to 3.

Algorithm parameters	Description
Minimum Information Gain	Enter a non negative float value. The set value becomes the minimum information gain required for a split. Information gain is a measure of how much a particular split reduces the uncertainty in the target variable. A higher minimum information gain can lead to a simpler tree with fewer splits, but might also result in lower predictive accuracy. The default value is 0.01.
Maximum Depth of Tree	Enter the positive integer value to specify the maximum depth of the decision tree. A deeper can model more complex relationship between input features and target variables but might lead to overfitting. The default value is 5.
Maximum Number of Bins	Enter the positive integer value >=2 which is equal to number of categories in any categorical feature. The set value becomes the maximum number of bins that can be used for each feature. A larger number of bins can capture more information about the feature, but can also increase the tree complexity. The default value is 5.
Impurity Method	Select variance to calculate the impurity of a split.
Minimum Instance per Node	Enter a positive integer value greater than 1. It is required to split a node. A higher minimum number of instances can lead to a simpler tree with fewer splits but might result in lower predictive accuracy. The default value is 4.
Sub Sampling Rate	Enter the value between 0 to 1 to set the sub sampling rate (feature sampling rate). It controls the fraction of features and trains each decision tree in the forest. The default value is 0.1.
Feature Subset Strategy	Select one of the following options to reduce the correlation between the trees in the forest: • auto • all • sqrt • log2 • onethird

Saving a pipeline

To save a pipeline, perform the following steps:

1. Open a pipeline on the canvas, the SAVE button appears on the upper-right.
2. Click **Save**. The Save Pipeline dialog appears.



3. Specify the following fields:

Fields	Description
Select Project	Select an existing project for the pipeline using the dropdown.
Pipeline Name	Enter the pipeline name.
Tags	Provide the tag name to the pipeline and press Enter.
Comment	Enter the comment in the textbox.
Drop your file here	Upload the pipeline information in the form of a *txt*, *pdf*, *doc*, or *docx* format.
Add New Project	Click **Add New Project** to add a new project in which you want to add a pipeline. The Create New Project dialog appears. Enter the project name and click **Confirm**. The New Project gets created.

Fields	Description
Confirm	Click **Confirm** to save the pipeline. When you update an existing pipeline and save it, the save options allows you to choose between two actions. Firstly, you can save the pipeline under the same (current) version, for example, v1, by clicking **Confirm**. Alternatively, you can click **Create New Version** to save the updated pipeline as a different version, say v2.

Running a pipeline

To run or execute a pipeline, perform the following steps:

1. Open a pipeline on the canvas, the Run button appears on the upper-right pane.
2. Click **Run**. The pipeline gets executed.

Model inference

Model inference or Model Serving tests the model with new data that it has not encountered during model development. This data is applied to the trained model separately from the training data.

Viewing model batch

To define properties of the Model View Batch, perform the following steps:

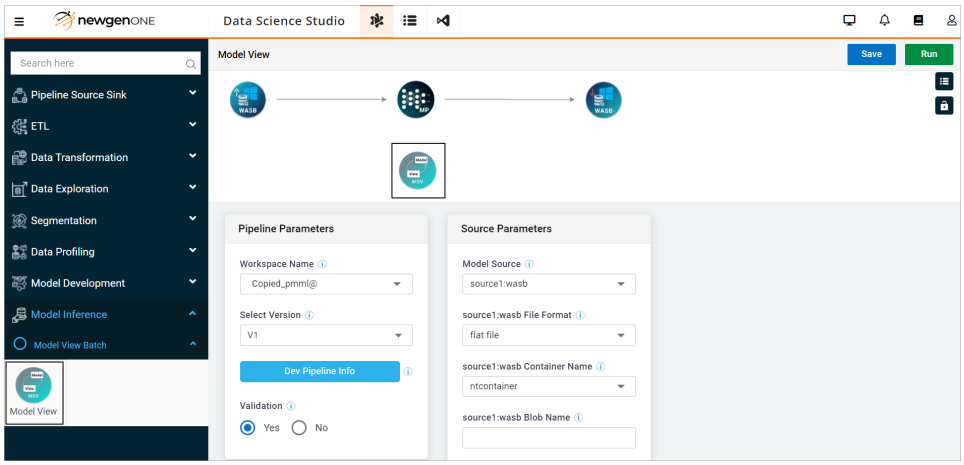
1. Go to **Canvas**.
2. Click the **Model Inference** on the navigation pane.
3. Select **Model View Batch** using dropdown. The Model View node appears.

Model View

The Model View allows you to predict or infer from the model. After completing model training, it performs inference or prediction on another set of data.

To define parameters for the Model View node, perform the following steps:

1. Perform the steps from 1 to 3 as mentioned in the Viewing model batch section.
2. Drag the **Model View** node from navigation pane and drop it on canvas.
3. Click the **Model View** node. The Pipeline Parameters section appears.



4. Specify the following Pipeline Parameters:

Pipeline parameters	Description
Workspace Name	Select the successfully executed model pipeline for workspace.
Select Version	Select the version of the selected model pipeline.
Dev Pipeline Info	Click to view the selected development pipeline details. This option appears if you select the version.
Validation	Select one of the following option to validate the model: • Yes • No

The Source Parameters appear only if you select the version and workspace name.

5. Specify the following Source Parameters:

Source parameters	Description
Model Source	Select the source1:wasb using dropdown. The model source is similar to the model pipeline source type.
File Format	Select one of the following file format for the selected source: • flat file • parquet • JSON • XML This field appear if you select the source in Model Source parameter.
Container Name	Select the Azure Blob container or the container on which you have access using the dropdown.
Blob Name	Select the file name that exists in the selected container with the file extension type.
Delimiter	Select one of the following delimiter to organize data: • comma (,) • colon (:) • semi-colon (;) • inverted comma (' ') • space ( )

Source parameters	Description
Non-Features Column	Select the features that you don't want to use for modelling. • Click the Non-Feature Column textbox. The Dataset dialog appears. • Select the required column(s) checkbox. You can also use the following options: ◦ Searchbox — Use the searchbox to find the data by its name. ◦ Column View — Use this icon to view dataset in column. ◦ Row View — Use this icon to view dataset in row.
Id Column	Select the Column ID using the dropdown. The selected Column ID work as identifier for the dataset.
Prediction Limit	A probability threshold is set for the predictions. Above this threshold, the final prediction is considered a positive condition, while below it, the final prediction is considered a negative condition, as selected in the development pipeline for binary classification.
The following parameters appear if you select the AutoFeature in the Model Source.	
Connection String	Specify the following details for connection that exist in the database: • Database Type — Select the database such as MySQL and Hive. • Hostname — Enter the server address. • Port — Enter the port number of the selected server. • Username — Enter the database username. • Password — Enter the username password.
Parent Child Relation	Select the required database or the database list on which the database user has access, with a search field available above the list. It is required to define the parent child relationship between tables of database.

Source parameters	Description
Dataset Operations	Contains the required operation details for the dataset created through the parent-child relation combination. Thus, specify the following fields: • Non-feature column — Select the required column(s) to exclude from dataset. • Id Column — Select the column ID that contains unique identification of every row from the dataset. • Select function — Select one of the following operation to perform on dataset. The values are: ◦ ImputeWithMedian ◦ DeleteAllNullRows ◦ FillAllNullRowWith ◦ ImputeWithMean • UserInputValue — Enter the user input value. This parameter appears if you select the FillAllNullRowWith in the select function.

Appendix

This appendix describes the limitations or restrictions you might encounter when you add or update the columns.

Data Science Studio constraints

The following table lists the constraints for the Data Science Studio:

<table>
<tr><th>Constraints</th><th colspan="2">Description</th></tr>
<tr><td rowspan="14">Special characters support in column name</td><td colspan="2">The column name supports the following special characters and can also be converted to an underscore (_) if required.</td></tr>
<tr><td>Special characters</td><td>Description</td></tr>
<tr><td>(</td><td>Left parenthesis or Open parenthesis</td></tr>
<tr><td>]</td><td>Right bracket or Close bracket</td></tr>
<tr><td>)</td><td>Right parenthesis or Close parenthesis</td></tr>
<tr><td>[</td><td>Left bracket or Open bracket</td></tr>
<tr><td>!</td><td>Exclamation</td></tr>
<tr><td>$</td><td>Dollar</td></tr>
<tr><td>@</td><td>At symbol</td></tr>
<tr><td>#</td><td>Hash or Pound</td></tr>
<tr><td>%</td><td>Percent</td></tr>
<tr><td>&</td><td>Ampersand</td></tr>
<tr><td>^</td><td>Caret or Circumflex</td></tr>
<tr><td>~</td><td>Tilde</td></tr>
</table>

Constraints	Description	
Special characters support in column name	`	Grave accent
	?	Question mark
	<	Less than
	>	Greater than
	-	Hyphen or Minus
	+	Plus
	/	Slash or Forward slash
	,	Comma
	.	Period or Dot
	*	Asterisk
	:	Colon
Column name	It must not be blank.	
Column value	Each column must contain at least one value in the dataset.	
Required space in column name	In column names, avoid the use of spaces in prefixes or suffixes. For instance, the following formats are not acceptable: • Any preceding spaces before the column name such as ' colname' • Any succeeding space after the column name such as 'colname ' • Any preceding and succeeding spaces before and after like ' colname '	
Integer (int) data	An integer data must be less than 9 x 10^8.	
Delimiter support	There's only a support of single delimiter. If the data contains multiple delimiters, then only user-selected delimiter are allowed. The supported delimiter are as follows: • , (Comma) • \t (Tab) • @ (At symbol) • # (Hash or Pound)	

Constraints	Description
Null values	Empty values are allowed for the null value case (missing value), and the following values are considered as strings: • NA (Not-a-Number) • N/A (Not Available or Non Applicable) • NULL • Space
Characters	Data must not contain the following characters: • \s (Space) • \t (Tab) • \n (Newline)

Glossary

Accuracy

Accuracy is determined by dividing the proportion of accurate predictions by the overall number of predictions. It provides an overall assessment of model correctness but might not be suitable for imbalanced datasets.

areaUnderPR

areaUnderPR is a metric that evaluates the performance of a classification model, especially in imbalanced datasets. A higher AUC-PR indicates better model performance in situations where positive and negative class sizes are imbalanced.

areaUnderRoc

areaUnderRoc is a measure of a model's ability to differentiate between positive and negative classes. A higher AUC-ROC indicates better model discrimination, with a value of 1 representing perfect performance.

Categorical variable

Categorical variables are data that can be divided into distinct categories or groups. These variables have a limited number of distinct categories or levels.

Classification

Classification is a supervised learning task where the goal is to assign a predefined label or category to input data based on its characteristics or features.

Continuous variables

Continuous variables are numerical variables that can take an infinite number of values within a specified range. They are often measured with decimal precision, and statistical operations like addition, subtraction, multiplication, and division are meaningful for continuous variables.

Correlation with label column

Correlation with the label column involves assessing the relationship between a specific column and the target or label column in a dataset. It helps determine the feature correlation efficiency with the target variable.

Data cleaning

Data Science Studio offers comprehensive data cleaning operations that you can apply at both the column level and the entire dataset level.

Data connectors

Data connectors allow you to connect with various data sources like relational databases and Azure Blob Storage using the built-in data connectors.

Data migration

Data migration transfers data from one system or source to another. It involves various methods and techniques depending on the type of data and data sources, such as MySQL or Azure Blob Storage.

Data preparation

Data preparation allows you blend, integrate, cleanse, and explore data on a massive scale.

Data profiling

Data profiling is a process of analyzing and examining a dataset to gain insights into its characteristics, including completeness, accuracy, and accessibility. It helps understand the data before applying machine learning techniques.

Data quality

Data quality refers to the overall reliability, accuracy, and completeness of a dataset. It is a critical aspect of machine learning that greatly impacts the performance, accuracy, and reliability of models built on the data. Inadequate data quality can lead to biased, inaccurate, and unreliable predictions.

Data refinement

Data refinement is a process of cleaning and preparing data to ensure its quality, relevance, and consistency. It is crucial for obtaining meaningful results from data models and analyses.

Data source operations

Data source operations are actions or processes performed on data sources. It also includes operations like data joining and data union to collate data from different sources.

F1

F1 is a metric required to assess the overall performance of a classification model during the imbalance between the classes. It provides a balance between precision (ability to minimize false positives) and recall (ability to capture true positives) of a classification model.

Feature scaling

Feature scaling is a process of normalizing or standardizing numerical features in a dataset to ensure that they have similar scales. This prevents certain features from dominating others in machine learning algorithms.

GAN (Generative Adversal Network)

GAN (Generative Adversarial Network) is an algorithm that consists of two neural networks: the generator, which creates synthetic data, and the discriminator, which monitors the difference between original and synthetic data. It prevents generating duplicate data. These two models were trained competitively in an adversarial manner.

Handling Imbalance Data

Handling Imbalanced Data addresses situations in machine learning where the distribution of data classes in a dataset is highly skewed or imbalanced. This leads to biased model predictions, and techniques are employed to mitigate the impact of class imbalance, such as oversampling, undersampling, or using specialized algorithms.

Homogeneous data

Homogeneous data is data of a similar type or format that makes them consistent and suitable for analysis.

Hyperparameter tuning

Hyperparameter tuning involves defining specific configurations before model training, which exerts a substantial impact on the resultant model's performance. It occurs before model training and controls the step size or rate at which a model's parameters are updated during training.

Illegal casting

It occurs during an attempt to convert a column into a data type that is not compatible with its current data.

Information gain

Information gain quantifies how much information a feature provides about the target variable or class labels in a dataset. The higher the Information Gain, the more valuable the feature is for decision making or prediction.

K-Fold cross validation

K-Fold cross-validation is a resampling method that evaluates the model's performance in machine learning by partitioning the dataset into k roughly equal-sized subsets. The model undergoes training and validation k times, with each iteration employing a distinct subset as the validation set while utilizing the remaining (k-1) subsets as the training set.

Learning rate

Learning rate is a parameter required for model optimization. It controls the step size or rate at which a model's parameters are updated during training.

Macro average

Macro average is a method that aggregates performance metrics by calculating the metrics for each class separately and then averaging them.

MAE (Mean Absolute Error)

MAE calculates the average of the absolute differences, making it less sensitive to outliers.

Mean

The mean is the arithmetic average of a set of numbers, calculated by summing all the numbers in the dataset and then dividing by the count of numbers.

Median

The median is the middle number in a sorted dataset. If the dataset has an odd number of values, the median is the middle number. If the dataset has an even number of values, the median is the average of the two middle numbers.

Micro average

Micro average is a method that aggregates performance metrics by treating all individual data points or predictions equally. It gives equal weight to each data point, making it suitable for imbalanced datasets where some classes might have significantly more or fewer instances

Missing value imputation

Missing value imputation is a process of filling in missing or null values in a dataset with estimated or calculated values. It helps maintain the integrity of the dataset for analysis.

Mode

The mode is the value that appears most frequently in a dataset. A dataset can have one mode, more than one mode, or no mode at all.

MSE

MSE calculates the average of the squared differences between predicted values and the corresponding actual values from a dataset.

Outlier

An outlier is an observation or data point that differs from other observations within a dataset. Identifying and managing outliers in data exploration ensures that analysis results are unbiased.

Precision

Precision is a crucial metric that assesses the ratio of accurate positive predictions among all model-generated positive predictions. This metric is required in scenarios where the consequences of false positives can be expensive.

Predictors

Precision is a crucial metric. It assess the ratio of accurate positive predictions among all model generated positive predictions. This metric requires in scenarios where the consequences of false positives can be expensive.

Principal component analysis

Principal Component Analysis is a dimensionality reduction technique used in statistics and machine learning to reduce the number of features or variables in a dataset while preserving the most important information. It transforms the original features into a new set of orthogonal variables called principal components.

R-Squared

R-Squared is a metric that assesses the health of a regression model. It measures how much of the dependent variable's variance is explained by the independent variables in the model, with values between 0 and 1, where higher values indicate better performance.

Recall

Recall measures the ratio of accurate positive predictions among all actual positive instances present in the dataset.

Regex (Regular Expression)

Regex is a tool that matches patterns and manipulates text data. It also allows you to define specific search patterns for extracting or modifying text.

Regression

Regression is a supervised learning technique used for predicting a continuous target variable (also known as the dependent variable) based on one or more independent variables (predictors or features).

Regularization

Regularization prevents overfitting of a model by adding a penalty term to the loss function during model training to prevent the model from fitting the training data too closely.

RMSE

RMSE (Root Mean Squared Error) calculates the square root of the average of the squared differences between predicted and actual values.

Silhouette Distance Curve

Silhouette Distance Curve measures the similarity of data points within their own cluster compared to other clusters. Its aim is to identify the number of clusters (k) that maximizes the silhouette score, representing the optimal number of clusters for the given dataset.

Softmax

Softmax is an activation function and a probability distribution function used in the output layer of a neural network for classification problems.

Structure data

Structured Data refers to data organized and formatted consistently, typically residing in relational databases or tabular formats, making it easy to analyze and process. Structured data consists of well-defined data types in columns and rows.

Supervised learning

Supervised Learning involves algorithms learning from labeled data, where the training data includes both input features and corresponding target labels or outcomes.

Target variables

Target Variables are critical concepts representing the variables you try to predict or model in a supervised learning task. They are also known as dependent variables or response variables and are referred to as the Label Column in the Data Science Studio platform.

Time series

Time Series focuses on modeling and forecasting data points collected or recorded at regular time intervals.

Unsupervised learning

A type of machine learning where the algorithm learns patterns, structures, or representations in the data without explicit supervision or labeled target outcomes. The example are Clustering and Dimensionality Reduction.

Variance

A statistical measure that quantifies the amount of spread or dispersion in a dataset. In the context of data exploration, it identifies columns with low variance, indicating that they do not vary much and might not be informative for analysis.

Visualization techniques

Data Science Studio offers various visualization techniques, such as bubble charts, histograms, and scatter plots.

Weighted metrics

Weighted Metrics includes F1, Precision, and Recall. Each class is assigned a weight based on its importance or prevalence in the dataset. These weights reflect the relative significance of each class. Then, it calculates the weighted average of respective metric.

Standard Deviation

The standard deviation is a numeric value that measures the spread of data around its mean. A lower standard deviation value indicates that most of the data is closer to the mean, while a larger value indicates that the data is widely spread.