

SAS[®] Visual Data Mining and Machine Learning 8.5: User's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2019. *SAS® Visual Data Mining and Machine Learning 8.5: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Visual Data Mining and Machine Learning 8.5: User's Guide

Copyright © 2019, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

July 2023

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

8.5-P1:vdmm1ug

Contents

Chapter 1 / What's New in SAS Visual Data Mining and Machine Learning 8.5	1
Overview	1
Chapter 2 / Getting Started with SAS Visual Data Mining and Machine Learning in Model Studio	5
Charitable Giving Example	6
SAS Code Node Examples	25
Integrating Model Studio with SAS Visual Analytics	37
Open Source Code Node Example	45
Risk Modeling Example	52
Chapter 3 / Managing Projects	65
Overview of Model Studio Projects	65
Opening an Existing Project	67
Creating a New Project	68
Sharing a Project	69
Importing and Exporting a Project	72
Deleting a Project	73
Downloading Project Batch API Code	74
Specifying Global Settings	76
Specifying Project Settings	78
Importing a Project from SAS Visual Analytics	80
Chapter 4 / Working with Data	83
Data Management Overview	83
Importing Data	84
User-Defined Formats	85
Replace the Data Source	86
Managing Variable Assignments	87
Managing Global Metadata	91
Integration with SAS Visual Analytics	91
Explore and Visualize Your Data	93
Chapter 5 / Working with Templates	95
Overview of Templates	95
Creating a New Template from a Pipeline	95
Creating a New Template in The Exchange	96
Modifying an Existing Template	96
Available Templates	97
Chapter 6 / Working with Pipelines	105
Overview of Pipelines	105
Creating a New Pipeline	106
Actions on the Pipeline	107
Automated Pipeline Creation	107
Modifying a Pipeline	109
Creating a Template from a Pipeline	110
Running a Pipeline	111

Node Status	111
Comparing Pipelines	112
Insights Tab	112
Managing Models	114
Downloading Logs	118
Chapter 7 / Troubleshooting	119
Enable Debug Reporting	119
Troubleshooting Notes	120
Contact SAS Technical Support	120

What's New in SAS Visual Data Mining and Machine Learning 8.5

Overview	1
General Enhancements	1
New and Enhanced Nodes	2
Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning	3

Overview

SAS Visual Data Mining and Machine Learning 8.5 has many general enhancements and improvements for many existing nodes. Two new nodes were added. Three new Risk Modeling nodes were added as well. A Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning license is required to use the Risk Modeling nodes.

Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning released an update in March 2020.

General Enhancements

- SAS Visual Data Mining and Machine Learning can use automated machine learning to dynamically build a pipeline that is based on your data. This process automatically performs data preparation, model building, model comparison, and model selection on your data to create a pipeline. Once generated, the automated pipeline can either be run or it can be edited to enhance or modify the pipeline.
- Model interpretability properties and results have been updated:
 - You can specify the **Kernel SHAP** method for local interpretability.

- You can specify up to five individual observations to explain for LIME, Kernel SHAP, and ICE.
 - The ICE plot is now overlaid with the PD plot.
- For supervised learning nodes, you can use the exact percentile method for calculating lift and related assessment measures.
- On the **Data** tab, you can specify the **Distribution** method to impute interval input variables.

New and Enhanced Nodes

- The **Feature Machine** node is a Data Mining Preprocessing node that generates new features by performing variable transformations to improve data quality and improve model accuracy. For more information, see [Overview of Feature Machine](#).
- The **Model Composer** node is a Supervised Learning node. It automatically tunes hyperparameters for multiple model types concurrently with optimal allocations of evaluations performed across multiple rounds of hyperparameter tuning. For more information, see [Overview of Model Composer](#).
- **Bayesian Network** node:
 - The maximum value that can be specified for autotuning the **Maximum parents** hyperparameter has been updated to 5.
- **Decision Tree** node:
 - You can now autotune the **Minimum Leaf Size** hyperparameter.
 - The default value for the **Number of interval bins** property has been updated to 50.
- **Forest** node:
 - You can now autotune the **Minimum Leaf Size** hyperparameter.
 - You can now autotune the **Number of Interval Bins** hyperparameter.
 - The default value for the **Number of interval bins** property has been updated to 50.
- **Gradient Boosting** node:
 - You can use the **Interval target distribution** property to specify the distribution of the objective function for an interval target.
 - You can specify new options for performing early stopping:
 - You can use the **Class target metric** property to specify the error metric to use for a class target.
 - When the **Stagnation** method is selected, you can use the **Start from minimum error** property to specify whether to count iterations starting from the iteration that has the smallest validation error.
 - When the **Threshold** method is selected, you can use the **Threshold** property to specify the threshold value and the **Threshold iterations** value to specify the minimum number of training iterations to run before the validation error is compared to the specified threshold.

- ☐ You can now autotune the **Maximum Depth** hyperparameter.
- ☐ You can now autotune the **Minimum Leaf Size** hyperparameter.
- ☐ You can now autotune the **Number of Interval Bins** hyperparameter.
- **Imputation** node:
 - ☐ You can specify the **Distribution** method for interval input variables.
 - ☐ You can use the **Ignore methods in metadata** property to ignore imputation methods that are specified on the **Data** tab, in global metadata, or in a preceding **Manage Variables** node.
- **Open Source Code** node:
 - ☐ You can use the **Use output data in child nodes** property to specify whether to save a copy of the output data that a successive node can use.
 - ☐ You can use the **Drop rejected variables** property to specify whether variables with the role **Rejected** should be dropped from the output data set.
- **SVM** node:
 - ☐ You can now run support vector regression on interval target variables.
- **Transformations** node:
 - ☐ You can use the **Ignore methods in metadata** property to ignore transformation methods that are specified on the **Data** tab, in global metadata, or in a preceding **Manage Variables** node.
- **Variable Clustering** node:
 - ☐ You now select multiple representative variables per cluster with the **Number of representative variables** property.
- **Variable Selection** node:
 - ☐ The default value for the **Number of interval bins** property has been updated to 50 for **Decision Tree Selection** and **Forest Selection**.
 - ☐ For **Gradient Boosting Selection**, the **Interval target distribution**, **Class target metric**, **Start from minimum error**, **Threshold**, and **Threshold iterations** properties can be specified.

Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning

The **March 2020** release of Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning includes the following updates to the risk modeling nodes:

- Several plots in the **Results** now include generated detailed descriptions.
- Basic and advanced risk modeling templates were added.
- For the **Interactive Grouping** node:
 - ☐ You can now specify a special codes mapping data set.
 - ☐ You can now specify **Quantile** grouping for interval and ordinal variables.

- For tree-based grouping, you can now specify how splitting rules handle observations that contain missing values for a variable with the **Missing values** property.

Getting Started with SAS Visual Data Mining and Machine Learning in Model Studio

Charitable Giving Example	6
Tutorial Scenario	6
Create the Project and Import the Input Data	7
Modify Variables	9
Create a Pipeline	11
Generate Descriptive Statistics	11
Replace Missing Values	12
Automatically Train and Prune a Decision Tree	13
Create a Gradient Boosting Model	15
Impute Missing Values	17
Transform Variables	18
Create a Logistic Regression	19
Create a Neural Network	20
Compare Models	21
Interpretability of Champion Model	22
Project Insights	24
Publish the Champion Model	24
SAS Code Node Examples	25
Overview	25
Create the Project and Import the Input Data	25
Modify Variables	28
Create a Gradient Boosting Model	30
Perform Variable Selection	34
Integrating Model Studio with SAS Visual Analytics	37
Overview	37
Download the Sample Data	38
Create the Report	39
Create a Forest	39
Create a Support Vector Machine	40
Continuing in Model Studio	40

Open Source Code Node Example	45
Overview	45
Create the Project and Import the Input Data	45
Modify Variables	48
Create the Pipeline	49
Risk Modeling Example	52
Risk Modeling Overview	52
Create the Project and Import the Input Data	53
Modify Variables	56
Group the Characteristic Variables into Attributes	57
Build an Initial Scorecard	60
Perform Reject Inference	62
Build the Final Scorecard	63

Charitable Giving Example

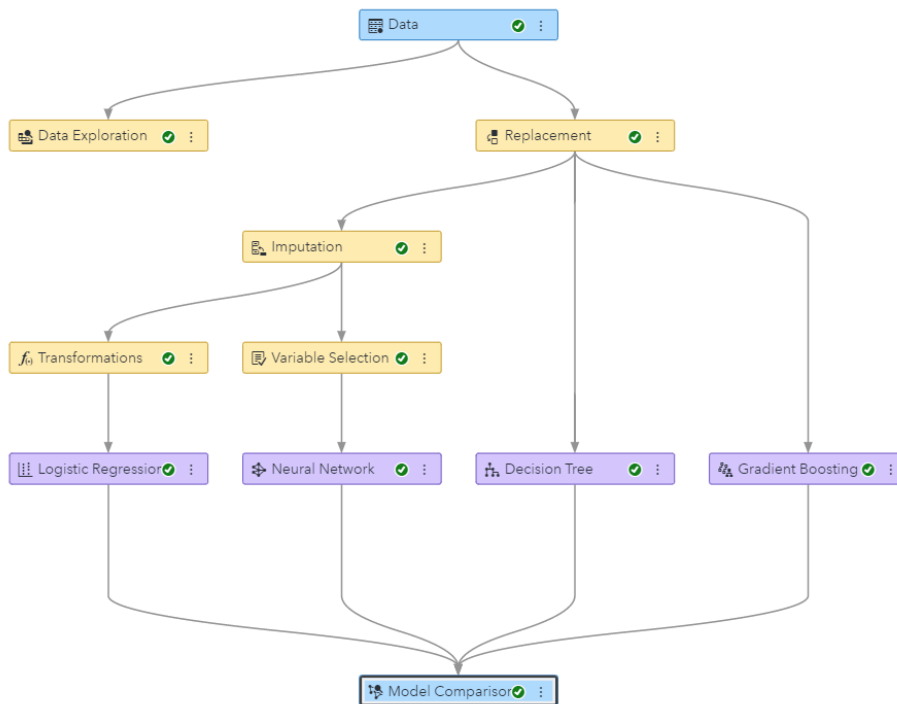
Tutorial Scenario

The Charitable Giving tutorial is intended for new Model Studio users. Although prior data mining experience is beneficial, any user can follow the discussion and complete the steps. The tutorial defines the problem, explores and visualizes the input data, performs data preparation, specifies model fit criteria, and then creates, configures, and trains multiple competing statistical modeling algorithms. A champion model is selected, and then score code is generated. The score code performs the trained champion model's analytic task on new data.


The narrative requires sequential data mining steps. Follow the steps in the order of presentation so that your results match the example. Later, you can modify your final pipeline model settings and add new competing models to see how your changes would affect predicted results.

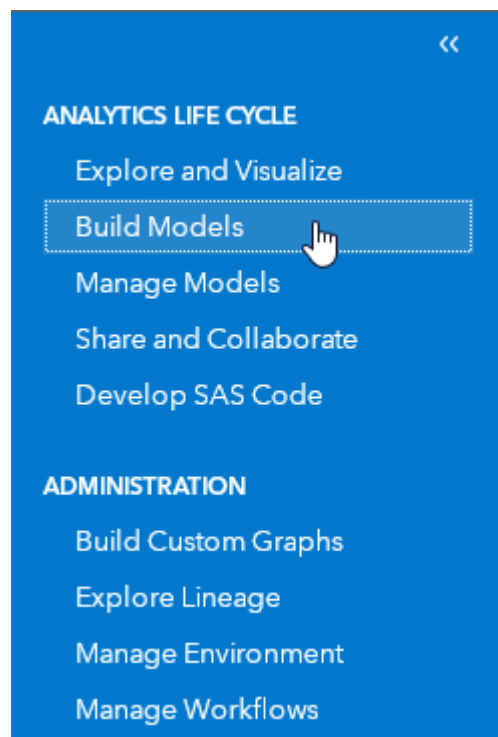
The tutorial uses a training data set with known target variable values named DONOR_DATA. You can download DONOR_DATA from the [SAS Visual Data Mining and Machine Learning product page](#). Click **Example Data for Getting Started with SAS Visual Data Mining and Machine Learning in Model Studio**. Download the ZIP file and extract DONOR_DATA.SAS7BDAT to a directory that your SAS Visual Data Mining and Machine Learning server can access.

In the tutorial, you are a data analyst at a national charitable organization. Your organization wants to use the results of a previous solicitation for donations to better target its next one. You want to determine which of the individuals in your mailing database are the most generous donors. By approaching only these people, your organization can spend less money on the solicitation effort and more money on charitable concerns. The finished pipeline diagram will resemble the one shown here:



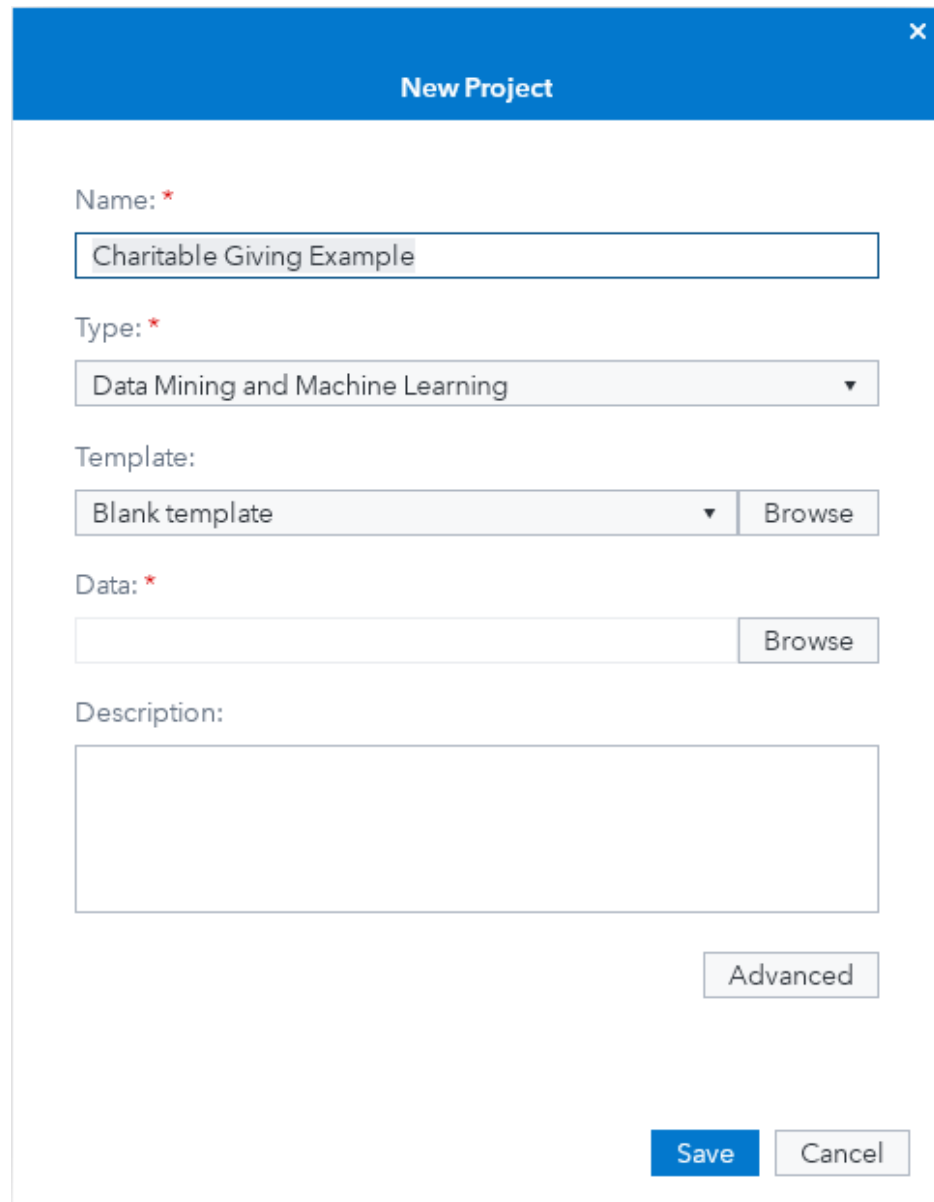
Create the Project and Import the Input Data

This example assumes that you are signed in to SAS Drive. In the upper left corner of the SAS Drive window, click  and select **Build Models**.



You are directed to the Projects page. To create the project that you will use in this example:

- 1 Select **New Project** in the upper right corner of the Projects page.
- 2 Enter **Charitable Giving Example** for **Name** in the New Project window.

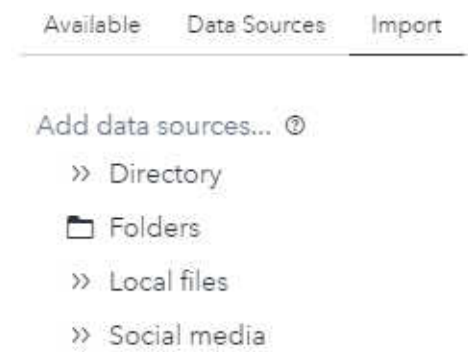


The screenshot shows the 'New Project' dialog box. The 'Name' field is filled with 'Charitable Giving Example'. The 'Type' dropdown is set to 'Data Mining and Machine Learning'. The 'Template' dropdown is set to 'Blank template', with a 'Browse' button next to it. The 'Data' field is empty, with a 'Browse' button next to it. The 'Description' field is a large empty text area. At the bottom right, there is an 'Advanced' button, a 'Save' button, and a 'Cancel' button.

- 3 Select **Data Mining and Machine Learning** for **Type**.

Note: **Forecasting** and **Text Analytics** are additional options if you have licensed those products.

- 4 Ensure that **Blank Template** is specified for **Template**.
- 5 In the **Data** field, select **Browse**. The Choose Data window appears.
- 6 In the upper left corner of the Choose Data window, select **Import**.



- 7 Select **Local files** ⇒ **Local file** and navigate to the folder where DONOR_DATA is stored. Select **DONOR_DATA.sas7bdat** and click **Open**.
- 8 Click **Import Item** in the upper right corner of the Choose Data window. When the data is successfully imported, a note appears, saying that the data is ready for use.

✓ The table was successfully imported on Oct 7, 2019 10:33 AM and is ready for use.


- 9 Once the data set is successfully imported, click **OK** in the lower right corner of the Choose Data window. This brings you back to the New Project window.
- 10 Click **Save** in the New Project window. This directs you to the **Data** tab, where you can modify the variables in your data set.

Modify Variables

On the **Data** tab, variable roles are indicated in the **Role** column. To change the role of a variable:

- 1 Select the TARGET_D variable. Select a variable by clicking the corresponding check box to the left of the **Variable Name** column. The options pane for the selected variable appears to the right of the variable table on the **Data** tab.

>>

TARGET_D 

Role:

Target ▼

Level:

Interval ▼

Order:

▼

Transform:

▼

Impute:

▼

Lower limit:

Enter a decimal value

Upper limit:

Enter a decimal value

- 2 Expand the drop-down list under **Role**, and select the role type that you want to assign to the selected variable. Changes made to each variable are automatically applied and saved. For TARGET_D, select the role **Rejected**.

CAUTION

To avoid making unwanted changes to variable properties, you must manually deselect each variable that you modify when you are finished making changes to its properties.

- 3 Ensure that the variables roles are set to the following values:
 - CLUSTER_CODE is set to **Rejected**.
 - CONTROL_NUMBER is set to **ID**.
 - TARGET_B is set to **Target**.
 - TARGET_D is set to **Rejected**.
 - _dmIndex_ is set to **Key**.
 - _PARTIND_ is set to **Partition**.

In data mining, a strategy for assessing the quality of model generalization is to partition the data source. A portion of the data, called the *training data set*, is used for preliminary model fitting. The rest is reserved for empirical

validation and is often split into two parts: validation data and test data. The validation data set is used to prevent a modeling node from overfitting the training data and to compare models. The test data set is used for a final assessment of the model. In this example, the data has been pre-partitioned where 60% of the data is assigned for training, 30% of the data is assigned for validation, and 10% of the data is assigned for testing.

Select **Map Partition Levels** and ensure that the partition levels are set to the following values:

- ☐ **Training Level** is set to **1**.
 - ☐ **Validation Level** is set to **0**.
 - ☐ **Test Level** is set to **2**.
 - ☐ Click **Save**
 - All other variable roles are set to **Input**.
- 4 Select the following variables:
- FILE_AVG_GIFT
 - LAST_GIFT_AMT
 - LIFETIME_AVG_GIFT_AMT

Set the property **Transform** to **Log10** for these variables.

TIP It is possible to select multiple variables for editing at one time by selecting the box in front of each variable name. By simultaneously selecting FILE_AVG_GIFT, LAST_GIFT_AMT, and LIFETIME_AVG_GIFT_AMT, you can change the **Transform** property for all three variables at once.

Create a Pipeline

- 1 Select the **Pipelines** tab in the upper left corner.
- 2 Right-click the **Data** node and select **Run**.
- 3 Once the **Data** tab has run successfully, continue with the following sections to build your pipeline.

Generate Descriptive Statistics

To see a statistical summary of the input data:

- 1 Right-click the **Data** node and select **Add child node** ⇒ **Miscellaneous** ⇒ **Data Exploration**.
- 2 Right-click the **Data Exploration** node and select **Run**.

- 3 Once the pipeline has run successfully, right-click the **Data Exploration** node and select **Results** from the menu that appears. The following pieces of information about the input data appear either as graphic tiles or rows in a table of projects:
 - **Data Partition Summary**
 - **Important Inputs**
 - **Class Variable Summaries**
 - **Class Variable Distributions**
 - **Interval Variable Moments**
 - **Interval Variable Summaries**
 - **Interval Variable Distributions**
 - **Missing Values**
 - **Target by Input Crosstabulations**
 - **Properties**
 - **Output**
- 4 Click **Close** to return to the pipeline.

Replace Missing Values

In this example, the variables SES and URBANICITY are class variables for which the value ? denotes a missing value. SAS denotes a missing value using a blank for character variables and a period for numeric variables. Because this data uses a question mark instead, Model Studio sees it as an additional level of a class variable. However, the knowledge that these values are missing will be useful later in the model-building process. To implement a **Replacement** node:

- 1 Right-click the **Data** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Replacement**.
- 2 Once created, select the **Replacement** node.
- 3 In the options pane, complete the following:
 - a Set **Replacement value for unknown class levels** to **Missing value**.
 - b Expand **Interval Inputs**.
 - c Ensure that the **Default limits method** property is set to **Standard deviation from the mean**.
- 4 Right-click the **Replacement** node and select **Run**.
- 5 Once the node has run successfully, right-click the node and select **Results** to view detailed information in each of the following sections of interest:
 - **Class Variables** — Displays the replacement values for class variables.

Class Variables

Name	Variable Label	Replace Variable	Type	Replacement Value...
CARD_PROM_12		REP_CARD_PROM_12	N	.
DONOR_GENDER		REP_DONOR_GENDER	C	.
FREQUENCY_STATU S_97NK		REP_FREQUENCY_ST ATUS_97NK	N	.
HOME_OWNER		REP_HOME_OWNER	C	.
INCOME_GROUP		REP_INCOME GROU P	N	.
IN_HOUSE		REP_IN_HOUSE	N	.

- **Interval Variables** — Displays the replacement values for interval variables.

Interval Variables

Name	Variable Label	Replace Variable	Limits Method	Lower Limit
DONOR_AGE		REP_DONOR_AGE	STDDEV	8.5944
FILE_AVG_GIFT		REP_FILE_AVG_GIFT	STDDEV	-12.3594
FILE_CARD_GIFT		REP_FILE_CARD_GIF T	STDDEV	-8.5625
LAST_GIFT_AMT		REP_LAST_GIFT_AMT	STDDEV	-16.8120
LIFETIME_AVG_GIFT_ AMT		REP_LIFETIME_AVG_ GIFT_AMT	STDDEV	-12.3594
LIFETIME_CARD_PR OM		REP_LIFETIME_CARD _PROM	STDDEV	-7.0143

- **Replacement Counts** — Displays the number of observations replaced for each variable.

Replacement Counts

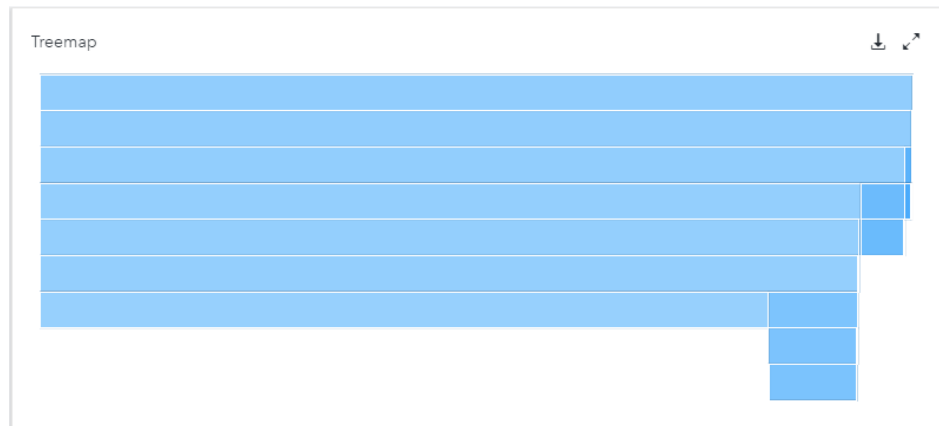
Name	Variable Label	Train	Role	Variable Level
CARD_PROM_12		0	INPUT	NOMINAL
DONOR_AGE		59	INPUT	INTERVAL
DONOR_GENDER		0	INPUT	NOMINAL
FILE_AVG_GIFT		143	INPUT	INTERVAL
FILE_CARD_GIFT		148	INPUT	INTERVAL
FREQUENCY_STATU S_97NK		0	INPUT	NOMINAL
HOME_OWNER		0	INPUT	BINARY

- 6 Close the **Results** window.

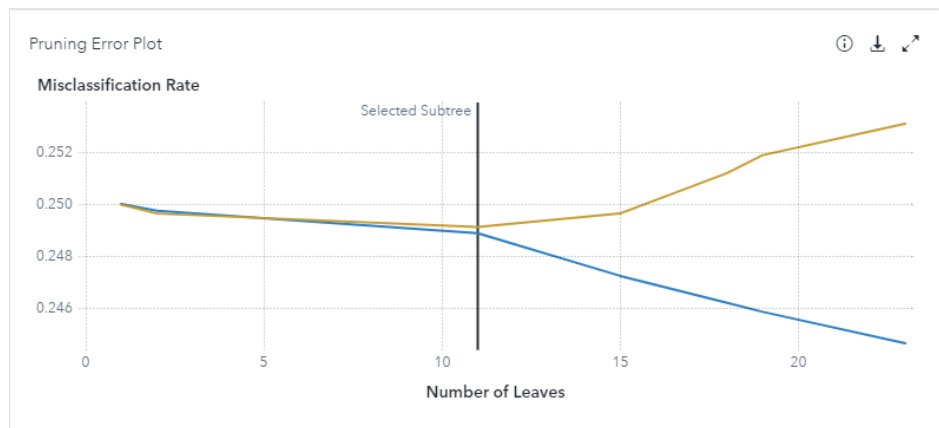
Automatically Train and Prune a Decision Tree

Decision tree models are advantageous because they are conceptually easy to understand, yet they readily accommodate nonlinear associations between input variables and one or more target variables. They also handle missing values without the need for imputation. Therefore, you decide to first model the data using decision trees. You will compare decision tree models to other models later in this example.

Note: A **Model Comparison** node is automatically created when you add a **Supervised Learning** node to your pipeline. Therefore, a **Model Comparison** node will be added to your pipeline when you add the **Decision Tree** node.



- **Pruning Error Plot** — An explanation of this plot is provided.



- **Fit Statistics** — Located on the **Assessment** tab.

Fit Statistics

Target Name	Data Role	Partition Indicator	Formatted Parti...	Sum of Frequen...
TARGET_B	TEST	2	2	1,937
TARGET_B	TRAIN	1	1	11,623
TARGET_B	VALIDATE	0	0	5,812

- 6 Close the **Results** window.

Create a Gradient Boosting Model

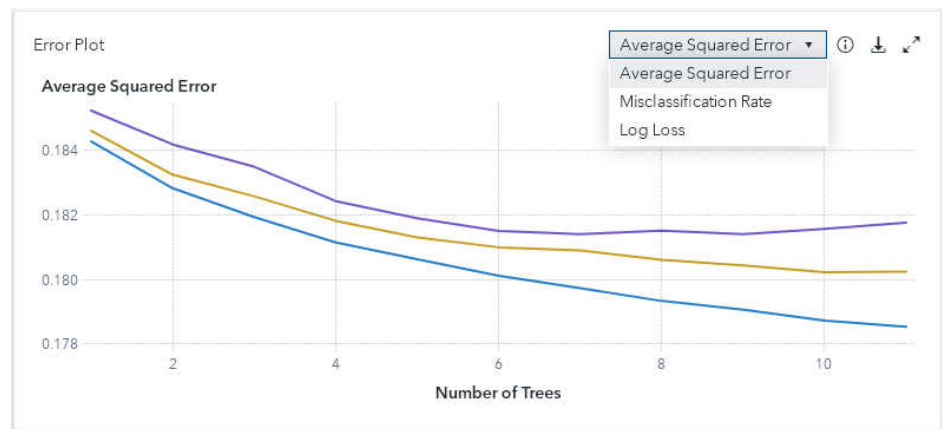
The Gradient Boosting node uses a partitioning algorithm to search for an optimal partition of the data for a single target variable. Gradient boosting is an approach that resamples the analysis data several times to generate results that form a

weighted average of the resampled data set. Tree boosting creates a series of decision trees that form a single predictive model. Like decision trees, boosting makes no assumptions about the distribution of the data. Boosting is less prone to overfit the data than a single decision tree. If a decision tree fits the data fairly well, then boosting often improves the fit. For more information, see [Overview of Gradient Boosting](#).

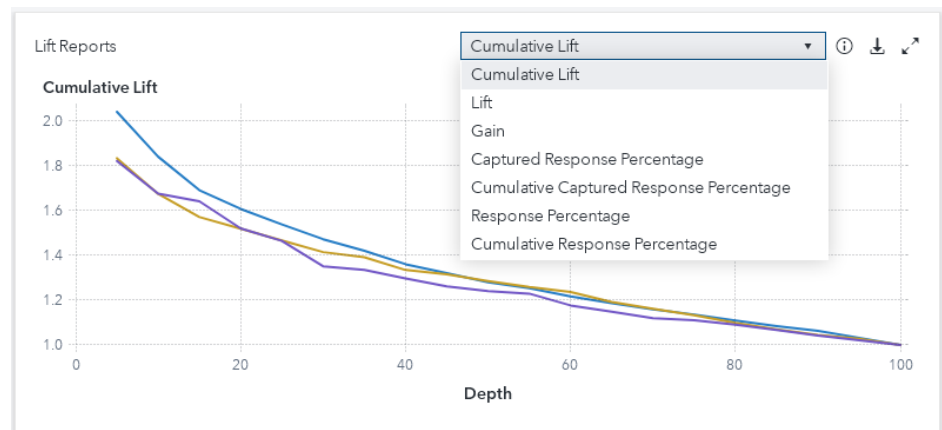
To create a gradient boosting model of the data:

- 1 Right-click the **Replacement** node and select **Add child node** ⇒ **Supervised Learning** ⇒ **Gradient Boosting**.
- 2 Once created, select the **Gradient Boosting** node.
- 3 In the options pane, expand **Tree-splitting Options**. Set **Maximum depth** to **10**.
- 4 In the options pane, enable the **Perform Autotuning** option. Autotuning automatically adjusts the hyperparameters that govern the model training process to the best values. For more information, see [Overview of Autotuning](#).
- 5 Right-click the **Gradient Boosting** node and select **Run**.
- 6 Right-click the **Gradient Boosting** node and select **Results** to view components such as the following:

■ Error Plot



■ Lift Reports — Located on the **Assessment** tab.



■ Variable Importance

Variable Label	Role	Variable Name	Training Import...	Importance Sta...
Replacement: RECENT_RESPONSE_COUNT	INPUT	REP_RECENT_RESPONSE_COUNT	8.2212	39.7246
Replacement: PEP_STAR	INPUT	REP_PEP_STAR	3.5768	13.3957
Replacement: CARD_PROM_12	INPUT	REP_CARD_PROM_12	2.8652	6.6038
Replacement: LIFETIME_MAX_GIFT_AMT	INPUT	REP_LIFETIME_MAX_GIFT_AMT	2.1722	5.3370
Replacement:				

■ Autotune Best Configuration

Parameter	Value
Evaluation	62
Number of Variables to Try	24
Learning Rate	0.5050
Sampling Rate	1
Lasso	5
Ridge	5
Number of Bins	100
Maximum Tree Levels	3

Because of the nondeterministic behavior of SAS Viya, your results might not match these results.

- 7 Close the **Results** window.

Impute Missing Values

For decision trees, missing values are not problematic. Surrogate splitting rules enable you to use the values of other input variables to perform a split for observations with missing values. In Model Studio, however, models such as regressions and neural networks ignore observations that contain missing values, which reduces the size of the training data set. Less training data can substantially weaken the predictive power of these models. To overcome this obstacle of missing data, you can impute missing values before you fit the models.

TIP Impute missing values before fitting a model that ignores observations with missing values if you plan to compare those models with a decision tree. Model comparison is most appropriate between models that are fit with the same set of observations.

To impute missing values:

- 1 Right-click the **Replacement** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Imputation**.
- 2 Once created, select the **Imputation** node.
- 3 In the options pane, under **Interval Inputs**, set **Default method** to **Median**.
- 4 Right-click the **Imputation** node and select **Run**.
- 5 Once the **Imputation** node has successfully run, right-click the node and select **Results**. Explore the following:
 - **Input Variable Statistics**

Input Variable Statistics

Input Variable	Variable Level	Number of Mis...	Percent Missing	Imputable
REP_CARD_PROM_12	NOMINAL	0	0	0
REP_DONOR_AGE	INTERVAL	2,925	25.1656	1
REP_DONOR_GENDER	NOMINAL	0	0	0
REP_FILE_AVG_GIFT	INTERVAL	0	0	0
REP_FILE_CARD_GIFT	INTERVAL	0	0	0

■ **Imputed Variables Summary**

Imputed Variables Summary

Imputed Variable	Method	Input Variable	Value	Percent Missing
IMP_REP_DONOR_AGE	MEDIAN	REP_DONOR_AGE	60	25.1656
IMP_REP_INCOME_GROUP	COUNT	REP_INCOME_GROUP	5	22.6103
IMP_REP_MONTHS_SINCE_LAST_PROM_R	MEDIAN	REP_MONTHS_SINCE_LAST_PROM_RESP	18	1.1701
IMP_REP_WEALTH_RATING	COUNT	REP_WEALTH_RATING	9	45.7885

- 6 Close the **Results** window.

Transform Variables

At the beginning of this example, you decided to transform the variables FILE_AVG_GIFT, LAST_GIFT_AMT, and LIFETIME_AVG_GIFT_AMT using the **Log10** methodology. To execute the transformation of these variables:

- 1 Right-click the **Imputation** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Transformations**.

- 2 Right-click the **Transformations** node and select **Run**.

Create a Logistic Regression

As part of your analysis, you want to include some parametric models for comparison with the decision trees that you built earlier in this example. Because it is familiar to the management of your organization, you have decided to include a logistic regression as one of the parametric models. To do so:

- 1 Right-click the **Transformation** node and select **Add child node** ⇒ **Supervised Learning** ⇒ **Logistic Regression**.
- 2 Right-click the **Logistic Regression** node and select **Run** from the resulting menu.
- 3 Once the node has successfully run, right-click the **Logistic Regression** node and select **Results**. Explore the following:

■ Parameter Estimates

Effect	Parameter	t Value	Sign	Estimate
REP_MEDIAN_HOME_VALUE	REP_MEDIAN_HOME_VALUE	6.8297	+	0.0002
REP_FREQUENCY_STATUS_97NK	REP_FREQUENCY_STATUS_97NK_1	6.5356	-	-0.4958
REP_MONTHS_SINCE_LAST_GIFT	REP_MONTHS_SINCE_LAST_GIFT	5.8012	-	-0.0393
REP_RECENT_CARD_RESPONSE_PRO	REP_RECENT_CARD_RESPONSE_PRO	4.2368	+	0.6058
REP_PEP_STAR	REP_PEP_STAR_0	4.2137	-	-0.2382

■ Regression Fit Statistics

Statistic	Description	Training	Validation	Testing
M2LL	-2 Log Likelihood	12,708.6535	6,342.1266	2,113.7133
AIC	AIC (smaller is better)	12,728.6535	6,362.1266	2,133.7133
AICC	AICC (smaller is better)	12,728.6725	6,362.1645	2,133.8275
SBC	SBC (smaller is better)	12,802.2609	6,428.8017	2,189.4022
ASE	Average Square Error	0.1814	0.1813	0.1809

- **Fit Statistics** — Located on the **Assessment** tab.

Target Name	Data Role	Partition Indicator	Formatted Parti...	Sum of Frequen...
TARGET_B	TEST	2	2	1,937
TARGET_B	TRAIN	1	1	11,623
TARGET_B	VALIDATE	0	0	5,812

- 4 Close the **Results** window.

Create a Neural Network

A neural network model is a stronger predictive model than the other models that you have generated up to this point. Neural networks are a class of parametric models that can accommodate a wider variety of nonlinear relationships between a set of predictors and a target variable than can logistic regression.

You know that the management of your organization prefers a stronger predictive model. You choose to add a neural network model even though it will be more complicated to explain than a regression model or a decision tree.

Building a neural network model involves two main phases. First, define the network configuration. You can think of this step as defining the structure of the model that you want to use. Then, iteratively train the model. The **Neural Network** node trains a specific neural network configuration, and is best used when you know a lot about the structure of the model that you want to define.

Before creating a neural network, you will reduce the number of input variables with the **Variable Selection** node. Performing variable selection reduces the number of input variables and saves computer resources. To use the **Variable Selection** node followed by the **Neural Network** node:

- 1 Right-click the **Imputation** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Variable Selection**.
- 2 Once created, right-click the **Variable Selection** node and select **Run**.
- 3 Right-click the **Variable Selection** node and select **Add child node** ⇒ **Supervised Learning** ⇒ **Neural Network**.
- 4 Once created, select the **Neural Network** node.
- 5 Take these actions in the options pane:
 - a Set **Number of hidden layers** to 5.
 - b Expand **Target Layer Options** and select **Direct connections**.

▼ Target Layer Options

☒ Direct connections

Interval target standardization:

Midrange ▼

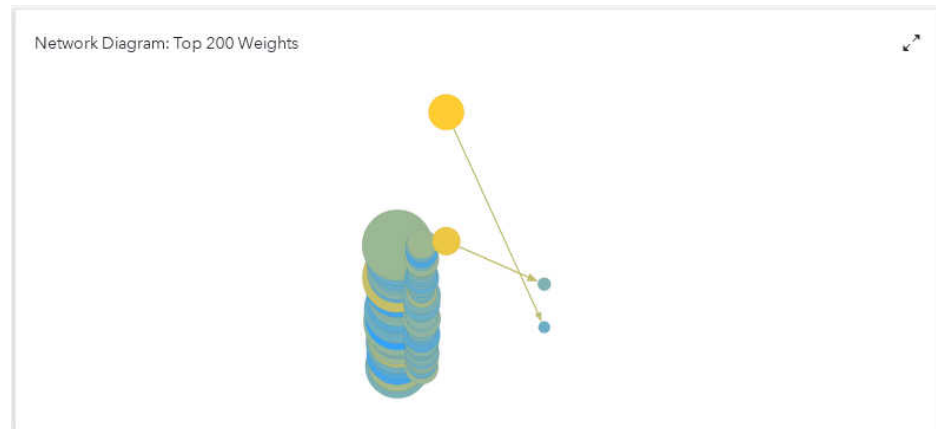
Interval target error function:

Normal ▼

Interval target activation function:

Identity ▼

- 6 Right-click the **Neural Network** node and select **Run**.
- 7 Once the **Neural Network** node has successfully run, right-click the node and select **Results** to view components such as the **Network Diagram**.




- 8 Close the **Results** window.

Compare Models

To use the **Model Comparison** node to compare the models that you have built in this example and to select one as the champion model:

- 1 Right-click the **Model Comparison** node that was created when you first created the **Decision Tree** node and select **Run**.
- 2 Right-click the **Model Comparison** node and select **Results**.
- 3 In the **Model Comparison** pane, you can see that the **Logistic Regression** model is selected as the champion model. In the Model Comparison node, Model Studio selects the champion model based on the value of a single statistic. You can specify which statistic to use for selection in the options pane. Because you did not change the value of this property, the default statistic (Kolmogorov-Smirnov statistic) was used.

Note: The selection statistic can also be specified in the **Rules** section within the Project Settings window. To edit the project settings, click  in the upper right corner of the window and click **Project settings**. Modifications to the project settings are applied to all applicable areas throughout the project.

- 4 Close the **Results** window.

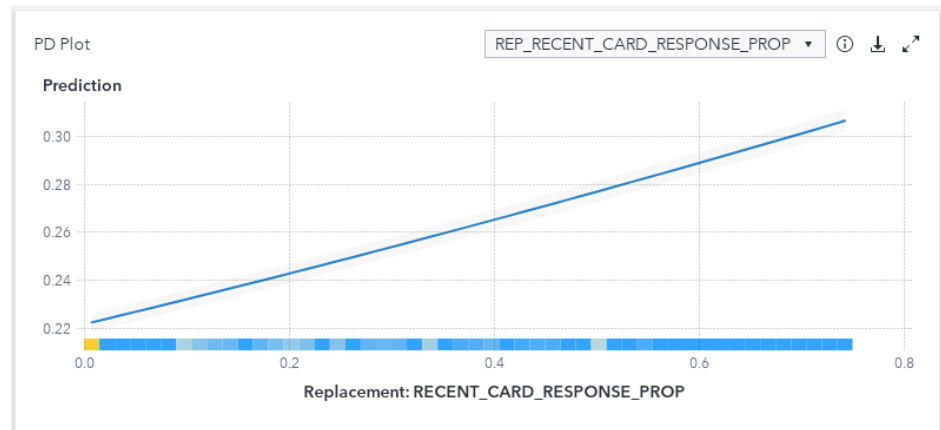
Interpretability of Champion Model

Now that you have selected the Logistic Regression model as the Champion Model, you want to bring to management more information about the interpretation of the model. There are several model interpretability plots and reports that can be generated. More details about the model interpretability plots available in Model Studio can be found in [Model Interpretability Plots](#).

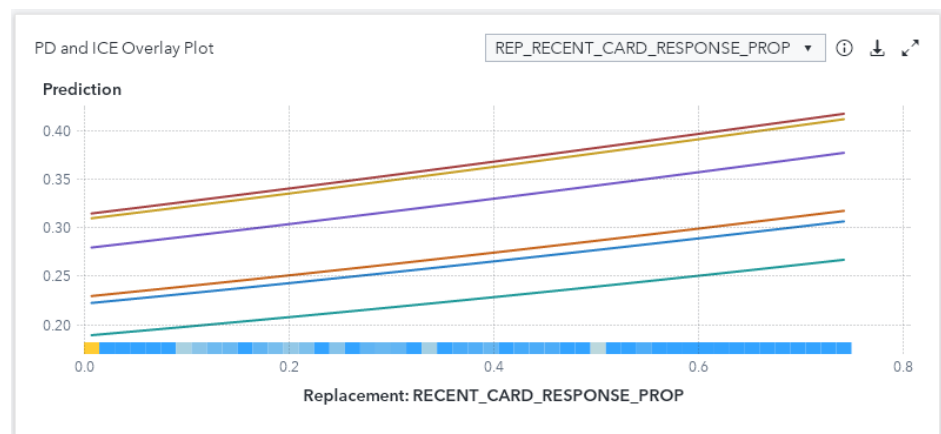
- 1 Click the **Logistic Regression** node.
- 2 Expand the **Global Interpretability** options and select **Variable Importance** and **PD plots**.
- 3 Expand the **Local Interpretability** options and select **ICE plots** and **LIME**.
- 4 Right-click the **Logistic Regression** node and select **Run**.
- 5 Once the **Logistic Regression** node has successfully run, right-click the node and select **Results**. Explore the following on the **Model Interpretability** tab:
 - **Surrogate Model Variable Importance**

Variable Label	Variable Name	Relative Import...	Role	Variable Level
Replacement: RECENT_CARD_RESPONSE_PROP	REP_RECENT_CARD_RESPONSE_PROP	1	INPUT	INTERVAL
Replacement: MEDIAN_HOME_VALUE	REP_MEDIAN_HOME_VALUE	0.4930	INPUT	INTERVAL
Replacement: FREQUENCY_STATUS_97NK	REP_FREQUENCY_STATUS_97NK	0.2309	INPUT	NOMINAL
Replacement: NUMBER_PROM_12	REP_NUMBER_PROM_12	0.1725	INPUT	INTERVAL

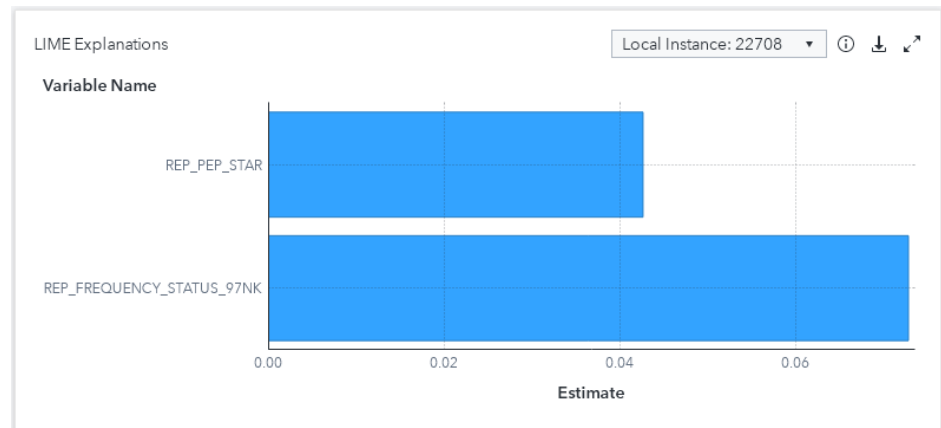
- **Partial Dependence**



■ Individual Conditional Expectation



■ LIME Explanations



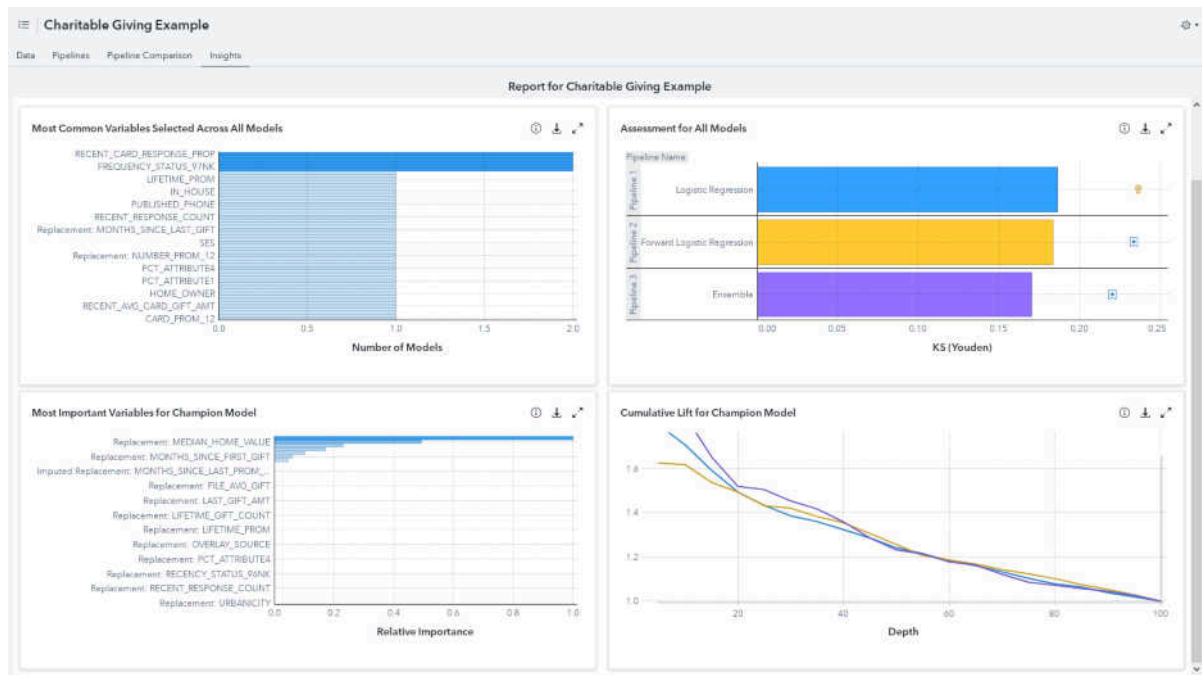
For more information about the model interpretability plots generated in the **Logistic Regression** node, see [Logistic Regression Results](#).

- 6 Close the **Results** window.

Project Insights

The **Insights** tab provides a summarized report of the project. The report includes the following information:

- the most commonly selected variables across all pipeline champion model
- a comparison of the assessment statistic across all pipeline champion models
- the most important variables for the selected champion model
- the cumulative lift plot for the selected champion model



In the image above, the **Forward Logistic Regression** model was created by the **Advanced template for class target** that is included with SAS Visual Data Mining and Machine Learning. The **Ensemble** model used automated machine learning to dynamically build a pipeline that is based on your data.

Publish the Champion Model

Before completing this section, ensure that an administrator has set up a location where you can publish your model. For more information, see [SAS Viya Administration: Publishing Destinations](#).

Publishing a model enables you to execute the model in various run-time engines. You can also manage a published model in SAS Model Manager, if it is licensed. For more information, see [Publish Models on page 115](#).

To publish a model:

- 1 Navigate to the **Pipeline Comparison** tab.
- 2 Ensure that the champion model is selected at the top of the **Pipeline Comparison** tab.
- 3 In the upper right corner of the **Pipeline Comparison** tab, click **:** and select **Publish models**.
- 4 The Publish Models window appears. Select the destination where you want your model to be published.
- 5 Select **Publish**.
- 6 The Publishing Results window appears. This window shows the name and published name of your model, as well as the status of your model (publishing, published successfully, and so on).
- 7 Close the **Publishing Results** window.

SAS Code Node Examples

Overview

In this section, you use the **SAS Code** node included with Model Studio to perform two different tasks. In the first example, you create a gradient boosting model with the GRADBOOST procedure. In the second example, you use the FOREST procedure to perform variable selection.


Both examples use the DONOR_DATA data set that is found on the [SAS Visual Data Mining and Machine Learning product page](#). Click **Example Data for Getting Started with SAS Visual Data Mining and Machine Learning in Model Studio**. Download the ZIP file and extract the contents to a directory that your SAS Visual Data Mining and Machine Learning server can access. This data includes names of those who were solicited for charitable donations and indicates whether they donated.

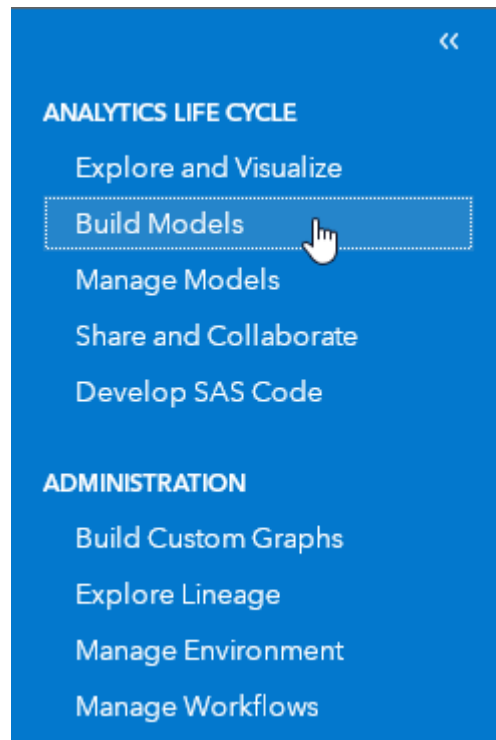
Download the ZIP file and extract the contents to a directory that your SAS Visual Data Mining and Machine Learning server can access.

Both examples require you to complete the steps in the [Create the Project and Import the Input Data](#) and [Modify Variables](#) sections. After completing these sections, you can complete the examples in any order.

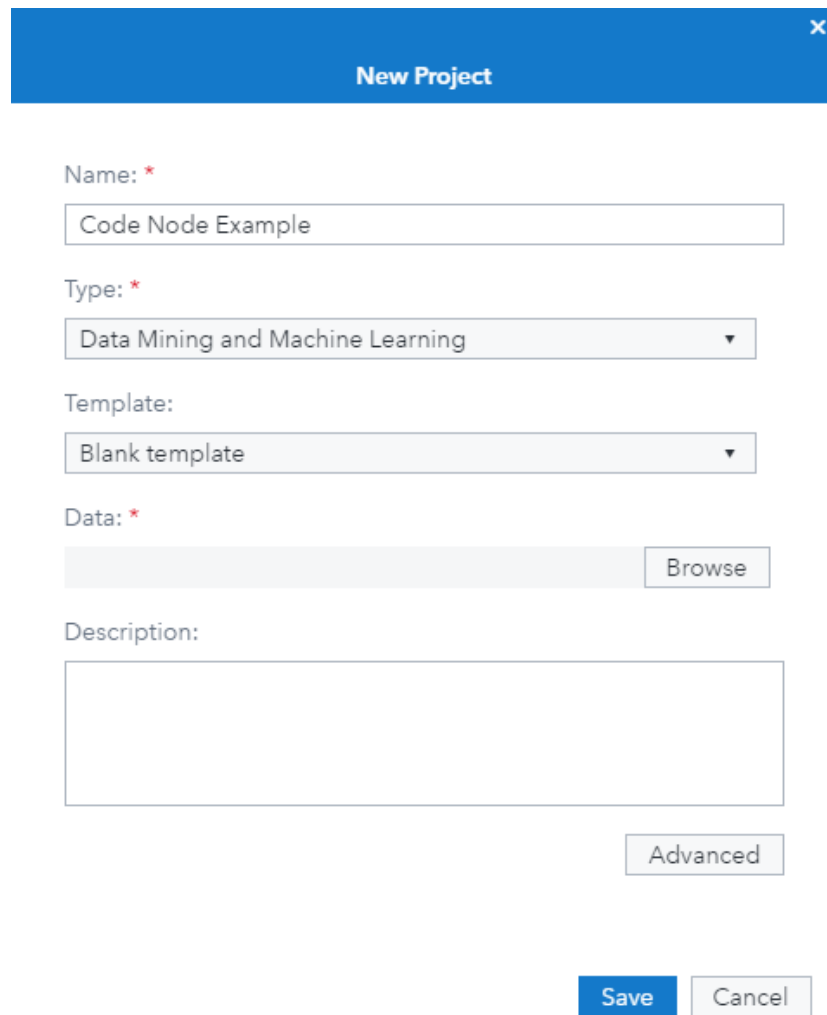
Create the Project and Import the Input Data

This example assumes that you are signed in to SAS Drive. To create the project that you will use in this example:

- 1 In the upper left corner of the SAS Drive window, click , and select **Build Models**.



- 2 Select **New Project** in the upper right corner of the page.
- 3 Enter Code Node **Example** for **Name** in the New Project window.



New Project

Name: *
Code Node Example

Type: *
Data Mining and Machine Learning

Template:
Blank template

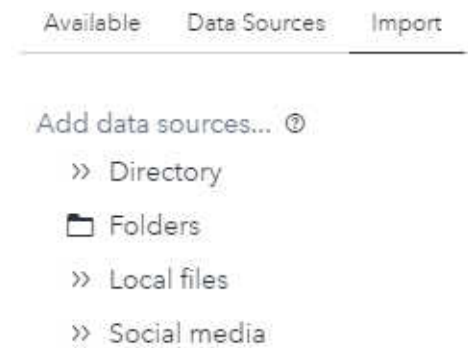
Data: *
Browse

Description:

Advanced

Save Cancel

- 4 Select **Data Mining and Machine Learning** for **Type**.
- 5 Ensure that **Blank Template** is specified for **Template**.
- 6 In the **Data source** field, select **Browse**. The Choose Data window appears.
- 7 In the upper left corner of the Choose Data window, select **Import**.



Available Data Sources Import

Add data sources... ⓘ

>> Directory


>> Folders

>> Local files

>> Social media

- 8 Select **Local files** ⇒ **Local file**, and navigate to the folder where DONOR_DATA is stored. Select **DONOR_DATA.sas7bdat** and click **Open**.

- 9 Click **Import Item** in the upper right corner of the Choose Data window. When the data is successfully imported, a note appears, saying that the data is ready for use.

 The table was successfully imported on Oct 7, 2019 10:33 AM and is ready for use.


- 10 Once the data set is successfully imported, click **OK** in the lower right corner of the Choose Data window. This brings you back to the New Project window.
- 11 In the lower right corner of the New Project window, click **Save**. You are redirected to the **Data** tab, where you can modify the variables in your data set.

Modify Variables

On the **Data** tab, variable roles are indicated in the **Role** column. To change the role of a variable:

- 1 Select a variable by clicking the corresponding check box to the left of the **Variable Name** column. The options pane for the selected variable appears to the right of the variable table on the **Data** tab. Select the TARGET_D variable.

>>

TARGET_D 

Role:

Target ▼

Level:

Interval ▼

Order:

▼

Transform:

▼

Impute:

▼

Lower limit:

Enter a decimal value

Upper limit:

Enter a decimal value

- 2 Expand the drop-down list under **Role**, and select the role type that you want to assign to the selected variable. Changes made to each variable are automatically applied and saved. For TARGET_D, select the role **Rejected**.

CAUTION

To avoid making unwanted changes to variable properties, you must manually deselect each variable that you modify when you are finished making changes to its properties.

- 3 Ensure that the variables roles are set to the following values:
 - CLUSTER_CODE is set to **Rejected**
 - CONTROL_NUMBER is set to **ID**
 - TARGET_B is set to **Target**
 - TARGET_D is set to **Rejected**
 - _dmIndex_ is set to **Key**.
 - _PARTIND_ is set to **Partition**.

In data mining, a strategy for assessing the quality of model generalization is to partition the data source. A portion of the data, called the *training data set*, is used for preliminary model fitting. The rest is reserved for empirical

validation and is often split into two parts: validation data and test data. The validation data set is used to prevent a modeling node from overfitting the training data and to compare models. The test data set is used for a final assessment of the model. In this example, the data has been pre-partitioned where 60% of the data is assigned for training, 30% of the data is assigned for validation, and 10% of the data is assigned for testing.

- All other variable roles are set to **Input**.
- 4 On the **Data** pane, select the `_PARTIND_` variable. On the options pane, select **Map Partition Levels** and ensure that the partition levels are set to the following values:
 - **Training Level** is set to **1**.
 - **Validation Level** is set to **0**.
 - **Test Level** is set to **2**.
 - 5 Set the property **Transform** to **Log10** for the following variables:
 - `FILE_AVG_GIFT`
 - `LAST_GIFT_AMT`
 - `LIFETIME_AVG_GIFT_AMT`

TIP It is possible to select multiple variables for editing at one time by selecting the box in front of each variable name. By simultaneously selecting `FILE_AVG_GIFT`, `LAST_GIFT_AMT`, and `LIFETIME_AVG_GIFT_AMT`, you can change the **Transform** property for all three variables at once.

Create a Gradient Boosting Model

This example requires you to complete the steps in the [Create the Project and Import the Input Data](#) and [Modify Variables](#) sections. This example also assumes that you have not created any other pipelines before starting this section. To create the gradient boosting model:

- 1 Select the **Pipelines** tab in the upper left corner.
- 2 Right-click the **Data** node and select **Run**.
- 3 Once the **Data** node has run successfully, continue with the following steps to build your pipeline.
- 4 Right-click the **Data** node and select **Add child node** ⇒ **Miscellaneous** ⇒ **SAS Code**.
- 5 Right-click the **SAS Code** node and select **Move** ⇒ **Supervised Learning**. A **Model Comparison** node is automatically added to the pipeline.
- 6 Select the **SAS Code** node. On the options pane, click **Open Code Editor**.
- 7 In the Training code editor, enter the following code:

```

proc gradboost data=&dm_data
  numBin=20 maxdepth=6 maxbranch=2 minleafsize=5
  minuseinsearch=1 ntrees=10 learningrate=0.1 samplingrate=0.5 lasso=0
  ridge=0 seed=1234;
  %if &dm_num_interval_input %then %do;
    input &dm_interval_input / level=interval;
  %end;

  %if &dm_num_class_input %then %do;
    input &dm_class_input / level=nominal;
  %end;

  %if "&dm_dec_level"="INTERVAL" %then %do;
    target &dm_dec_target / level=interval ;
  %end;

%else %do;

  target &dm_dec_target / level=nominal;
%end;

&dm_partition_statement;
ods output
  VariableImportance = &dm_lib..VarImp
  Fitstatistics       = &dm_data_outfit
;
savestate rstore=&dm_data_rstore;
run;


%dmcas_report(dataset=VarImp, reportType=Table,
  description=%nrquote(Variable Importance));

%dmcas_report(dataset=VarImp, reportType=BarChart, category=Variable,
  response=RelativeImportance, description=%nrquote(Relative
Importance Plot));

```

This code uses the following Model Studio macros:

- **DM_DATA** — A macro variable that identifies the CAS training table. If partitioned, the table contains the `_partInd_` variable that identifies which observations are used for training, validation, and test. This table is transient and is dropped when the node finishes running.
- **DM_NUM_INTERVAL_INPUT** — A macro variable that identifies the number interval input variables.
- **DM_NUM_CLASS_INPUT** — A macro variable that identifies the number of class input variables.
- **DM_DEC_LEVEL** — A macro variable that identifies the measurement level (binary, interval, ordinal, or nominal) of the target variable.
- **DM_PARTITION_STATEMENT** — A macro variable that identifies the partition statement. This variable is blank if the data is not partitioned.
- **DM_LIB** — A macro variable that identifies the SAS library where the variable importance table is saved. This table is named `VarImp`.

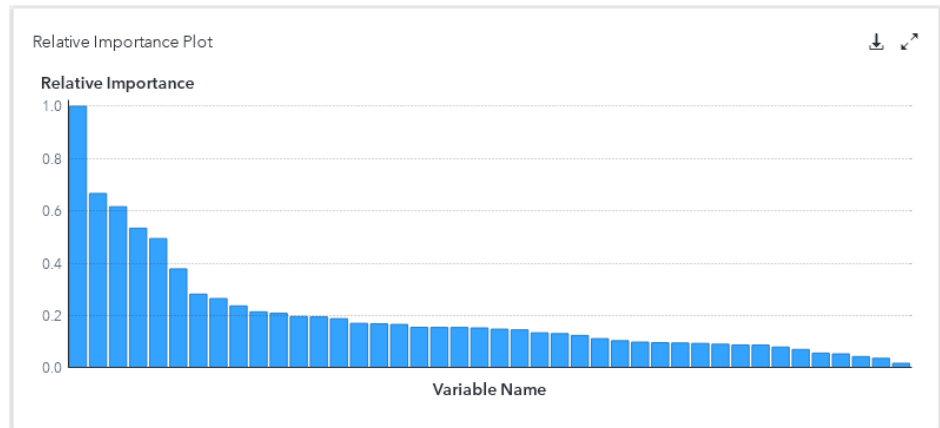
- **DM_DATA_OUTFIT** — A macro variable that identifies the fit statistics data set. This is the data set that is used by the model comparison node to select the best model in the pipeline.
 - **DM_DATA_RSTORE** — A macro variable that identifies the remote analytic store that is created by the GRADBOOST procedure. This table is used by Model Studio to score and assess the model.
 - **%DM_INTERVAL_INPUT** — A macro that identifies the interval input variables.
 - **%DM_CLASS_INPUT** — A macro that identifies the class input variables.
 - **%DM_DEC_TARGET** — A macro that identifies the project target variable.
 - **%DMCAS_REPORT** — A macro that enables the addition of more reports to the Results window.
- 8 In the upper right corner of the code editor, click .
- 9 Click **Close**.
- 10 Right-click the **SAS Code** node and select **Run**.

There are two DMCAS_REPORT calls to display the contents of the variable importance table VarImp.

- **reportType=Table** — This call adds the Variable Importance table to the results, as indicated in the DESCRIPTION argument.
 - **reportType=BarChart** — This call adds a bar chart that contains Relative Importance for each input.
- 11 Right-click the **SAS Code** node and select **Results**. Review the following results:
- **Variable Importance**

Variable Name	Training Importance	Importance Standard...	Relative Importance
RECENT_RESPONSE_COUNT	29.0716	9.8146	1
WEALTH_RATING	19.3900	1.6354	0.6670
CARD_PROM_12	17.9211	3.7937	0.6164
INCOME_GROUP	15.5319	1.8072	0.5343
RECENT_CARD_RESPONSE_COUNT	14.3824	1.9043	0.4947
MEDIAN_HOME_VALUE	11.0102	2.1206	0.3787

- **Relative Importance Plot**



- **Path EP Score Code** — Because you specified that the **SAS Code** node is a supervised learning node, it automatically creates EP Score Code. More specifically, this code produces an analytic store as its score code.

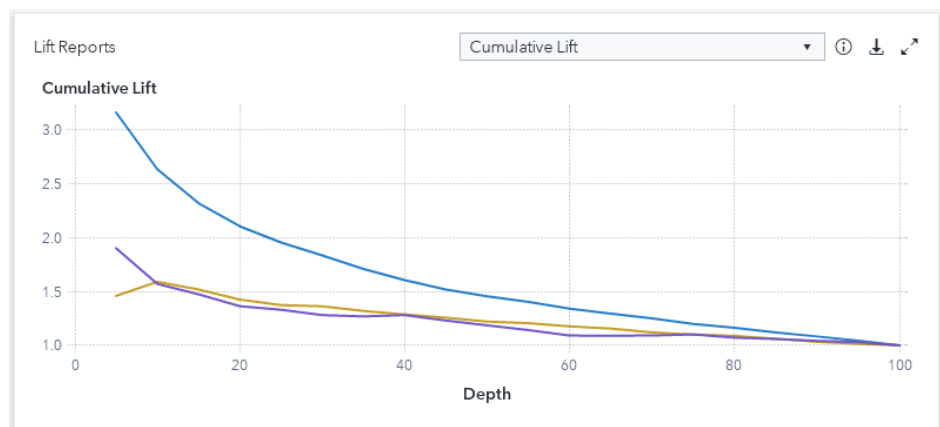
Path EP Score Code

```

1  data sasep.out;
2      dcl package score _5TSQYL3SD802T6KEVS1G8JBQ4();
3      dcl double "P_TARGET_B1" having label n'Predicted: TARGET_B=1';
4      dcl double "P_TARGET_B0" having label n'Predicted: TARGET_B=0';
5      dcl nchar(32) "I_TARGET_B" having label n'Into: TARGET_B';
6      dcl nchar(4) "_WARN_" having label n'Warnings';
7      dcl double "EM_EVENTPROBABILITY" having label n'Probability for TARGET_B =1';
8      dcl nchar(32) "EM_CLASSIFICATION" having label n'Predicted for TARGET_B';
9      dcl double "EM_PROBABILITY" having label n'Probability of Classification';
10     varlist allvars [_all_];
11
12
13     method init();
14         _5TSQYL3SD802T6KEVS1G8JBQ4.setvars(allvars);
15

```

- **Lift Reports, ROC Reports, Fit Statistics, and Event Classification** — These plots are automatically created because the **SAS Code** node is a supervised learning node. Click **Assessment** in the upper left corner of the SAS Code Results window to access these plots.





12 Click **Close** in the upper right corner to exit the results.

Perform Variable Selection

This example requires you to complete the steps in the [Create the Project and Import the Input Data](#) and [Modify Variables](#) sections. This example also assumes that you have not created any other pipelines before starting this section.

- 1 Select the **Pipelines** tab in the upper left corner.
- 2 Right-click the **Data** node and select **Run**.
- 3 Once the **Data** tab has run successfully, continue with the following sections to build your pipeline.
- 4 Right-click the **Data** node and select **Add child node** ⇒ **Miscellaneous** ⇒ **SAS Code**.

- 5 Select the **SAS Code** node. On the options pane, click **Open Code Editor**.
- 6 In the Training code editor, enter the following code:

```
proc forest data=&dm_data
  minleafsize=5 minuseinsearch=1 seed=12345 loh=0 numbin=20
  ntrees=100 maxdepth=20 inbagfraction=0.6 ;
  partition fraction (valid=0.3 seed=12345);
  %if &dm_num_interval_input %then %do;
    input %dm_interval_input / level=interval;
  %end;

  %if &dm_num_class_input %then %do;
    input %dm_class_input / level=nominal;
  %end;

  %if "&dm_dec_level"="INTERVAL" %then %do;
    target %dm_dec_target / level=interval ;
  %end;

  %else %do;
    target %dm_dec_target / level=nominal;
  %end;
  grow IGR;
  ODS output VariableImportance = &dm_lib..forestvarimportance ;
run;


&dmcas_report(dataset=forestvarimportance, reportType=BarChart,
category=Variable, response=RelativeImportance,
sortDirection=descending, sortBy=RelativeImportance,
description=%nrbquote(Relative Importance Plot));

filename _frf "&dm_file_deltacode";
data _null_;
  length string $200;
  file _frf;
  set &dm_lib..forestvarimportance ;
  where RelativeImportance < 0.3;
  string = 'if NAME "!!kstrip(Variable)!!" then ROLE="REJECTED";';
  put string;
run;
filename _frf;

&dmcas_report(file=&dm_file_deltacode, reportType=CodeEditor,
description=%nrbquote(Metadata Changes));
```

This code uses the following Model Studio macros:

- **DM_FILE_DELTACODE** — A macro variable that identifies the file that contains the DATA step code to modify the columnsmeta information that is exported by the node.

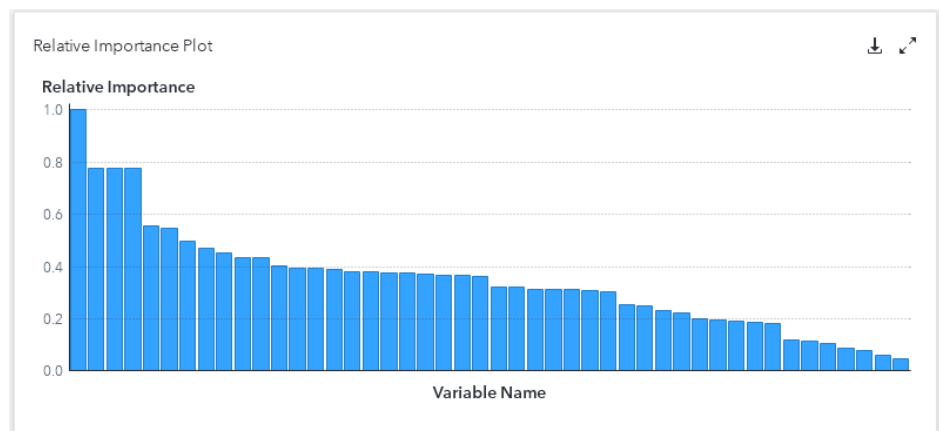
- 7 In the upper right corner of the code editor, click .

This code uses PROC FOREST to identify the relative variable importance for all variables in the input data. Those variables with a relative importance less than 0.3 are assigned the role **Rejected**. All other variables are kept.

- 8 Click **Close**.
- 9 Right-click the **SAS Code** node and select **Run**.
- 10 Right-click the **SAS Code** node and select **Results**.

There are two DMCAS_REPORT calls to display the contents of the variable importance table VarImp.

- **reportType=BarChart** — This call adds the Relative Importance Plot to the results.



- **reportType=CodeEditor** — This call adds the Metadata Changes information to the results. All of the variables that are dropped from the analysis are listed here.

Metadata Changes	
1	if NAME "RECENCY_STATUS_96NK" then ROLE="REJECTED";
2	if NAME "SES" then ROLE="REJECTED";
3	if NAME "LIFETIME_GIFT_RANGE" then ROLE="REJECTED";
4	if NAME "LIFETIME_MIN_GIFT_AMT" then ROLE="REJECTED";
5	if NAME "OVERLAY_SOURCE" then ROLE="REJECTED";
6	if NAME "MONTHS_SINCE_ORIGIN" then ROLE="REJECTED";
7	if NAME "MOR_HIT_RATE" then ROLE="REJECTED";
8	if NAME "RECENT_AVG_CARD_GIFT_AMT" then ROLE="REJECTED";
9	if NAME "PCT_ATTRIBUTE1" then ROLE="REJECTED";
10	if NAME "PEP_STAR" then ROLE="REJECTED";
11	if NAME "DONOR_GENDER" then ROLE="REJECTED";
12	if NAME "RECENT_STAR_STATUS" then ROLE="REJECTED";
13	if NAME "IN_HOUSE" then ROLE="REJECTED";
14	if NAME "HOME_OWNER" then ROLE="REJECTED";
15	if NAME "BIBLISHED_PHONE" then ROLE="REJECTED";

There is no score code available for this node because it is not a supervised learning node. There are also no assessment plots.

- 11 Click **Close** to exit the results.

Integrating Model Studio with SAS Visual Analytics

Overview

The example in this section shows the typical process of creating a model in SAS Visual Analytics, copying it to Model Studio, and then continuing your analysis. This example demonstrates the steps to complete this task and you are encouraged to repeat the process with several additional models.

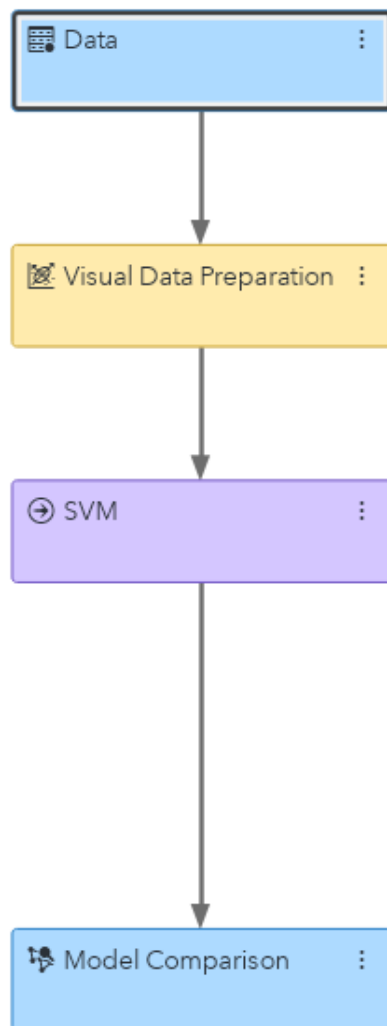
SAS Visual Analytics enables you to create many different types of reports that demonstrate the efficacy of various statistical models. For more information, see [Working with SAS Visual Statistics](#) in the SAS Visual Analytics documentation. Many of the reports created in SAS Visual Analytics can be sent to Model Studio for comparison against other Model Studio models.

In SAS Visual Analytics, you start by identifying the data that you want to model. Next, you can adjust certain characteristics of that data or create new data items. Then, you add one or more objects to the workspace and assign data items to those objects. Objects vary in complexity from simple tables to more complex statistical models.

Reports in SAS Visual Analytics can range from single-page reports that contain a single object to a multi-page reports that contain several dependencies and inter-object connections. However, when you copy a model from SAS Visual Analytics to Model Studio, a four-node pipeline is always created. The four nodes in the pipeline are as follows:

- The **Data** node
- The **Visual Data Preparation** node
- A model node
- The **Model Comparison** node

The data preparation and model nodes are discussed in more detail in this example. Below is a four-node pipeline that has been created from SAS Visual Analytics. In this case, the modeling algorithm used was Support Vector Machine.



What you cannot determine from this pipeline is that the SAS Visual Analytics report that generated the pipeline contained two pages, each containing its own model. This report, which you will re-create, uses a forest model to create inputs that are used by a support vector machine model. The creation of these inputs is captured in the Visual Data Preparation node, even though the Forest node does not appear in the pipeline.

This example uses the HMEQ data set, which contains 5,960 mortgage applications and indicates whether the applicant defaulted on the loan. The HMEQ data set can be found at <http://support.sas.com/documentation/onlinedoc/viya/examples.htm>. To complete this example, you must follow the proceeding sections in the order in which they appear.


Download the Sample Data


- 1 In a web browser, navigate to the [SAS Viya Example Data Sets](#) page.
- 2 Download the file **hmeq.csv** to your local machine.

Create the Report


This example assumes that you have already signed in to SAS Drive.

To create the report:

- 1 In the upper left corner of the SAS Drive window, click , and select **Explore and Visualize**. This opens SAS Visual Analytics, and enables you to choose a data source, create a new report, or load an existing report in the Explore and Visualize window.
- 2 Click the **Start with Data** button to load your data. The Choose Data window appears, enabling you to select the data source for this project.
- 3 On the **Import** tab, select **Local files** ⇒ **Local file**. Navigate to the location where you saved **hmeq.csv** and select **hmeq.csv**.
- 4 In the Choose Data window, click **Import Item**. After the table is successfully imported, click **OK**.
- 5 By default, the report is named **Report 1**, which is displayed in the upper middle portion of the screen. Before continuing with the example, rename the project by saving it.





Click  in the upper right corner of the page, and then select **Save**. This opens the Save As window. In the **My Folder** pane, navigate to a location where you have Write permission. In the **Name** field, enter **Integration Example**, and click **Save**.

Typically, you can save your work in **My Folder**.

- 6 In the **Data** pane, right-click **BAD** and select **Convert to category**.
- 7 Click  in the upper right corner of the window to save the report.





Create a Forest

To create a forest:

- 1 From the left pane, click  to select an object. Drag the  icon onto the canvas to create a forest.
- 2 Click  in the right pane. For **Response**, click **Add**, and select **LOAN**.
- 3 For **Predictors**, click **Add**, and select every variable except **BAD**. Click **OK**.
- 4 In the **Variable Importance** plot, right-click and select **Derive predicted**. In the New Prediction Items window, review the new data items and click **OK**.
- 5 Click  to save the report.

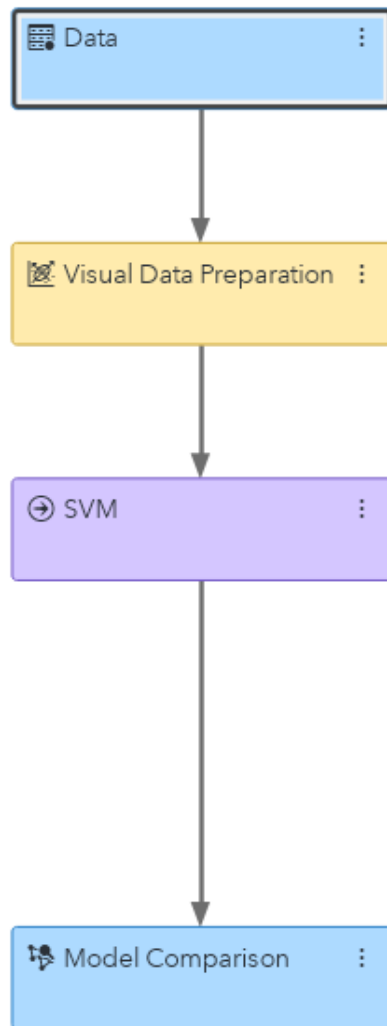
Create a Support Vector Machine

To create a support vector machine:

- 1 Click **+** to add a new page to the report.
- 2 From the left pane, click  to select an object. Drag the  icon onto the canvas to create a support vector machine.
- 3 Click  in the right pane. For **Response**, click **Add**, and select **BAD**.
- 4 For **Predictors**, click **Add**, and select **CLAGE**, **DEBTINC**, **DELINQ**, **DEROG**, and **Predicted: LOAN**. Click **OK**.
- 5 Click  to save the report.
- 6 In the **Relative Importance** plot, right-click and select **Create pipeline** ⇒ **Add to new project**. This action copies the model and all data preparation steps to Model Studio. Model Studio automatically opens.

Continuing in Model Studio

After creating the project and copying the necessary information from SAS Visual Analytics to Model Studio, you should see a pipeline that resembles the following image. Your Model Studio project is also named Integration Example, as that title is inherited from SAS Visual Analytics.



As discussed earlier, this pipeline contains four nodes. The **Visual Data Preparation** node contains the score code necessary to create the prediction and residual variables that are used as inputs for the support vector machine. From this point, you can modify the pipeline as if it were created in Model Studio. The only restriction is that you cannot edit the properties of the **Visual Data Preparation** node.

- 1 In the upper left corner, click **Data** to open the **Data** tab. This tab displays all the information that Model Studio knows about the input data set.
- 2 Notice that SAS Visual Analytics created two new variables: **_dmIndex_** and **_va_d__E_BAD**. The original target variable, **BAD**, has been assigned the role **Rejected**. SAS Visual Analytics creates temporary variables as needed to complete the tasks that you want to perform. In this case, changing BAD from a measure to a category necessitated the creation of **_va_d__E_BAD**.

These name changes also require careful consideration if you want to use a holdout data set. That holdout data set must have a target variable name that exactly matches the target variable name created by SAS Visual Analytics. The target variable is used for generating assessment statistics, not scoring.

- 3 Also, notice that all other variables are assigned the role **Input**. This does not match the support vector machine that you created in SAS Visual Analytics. The

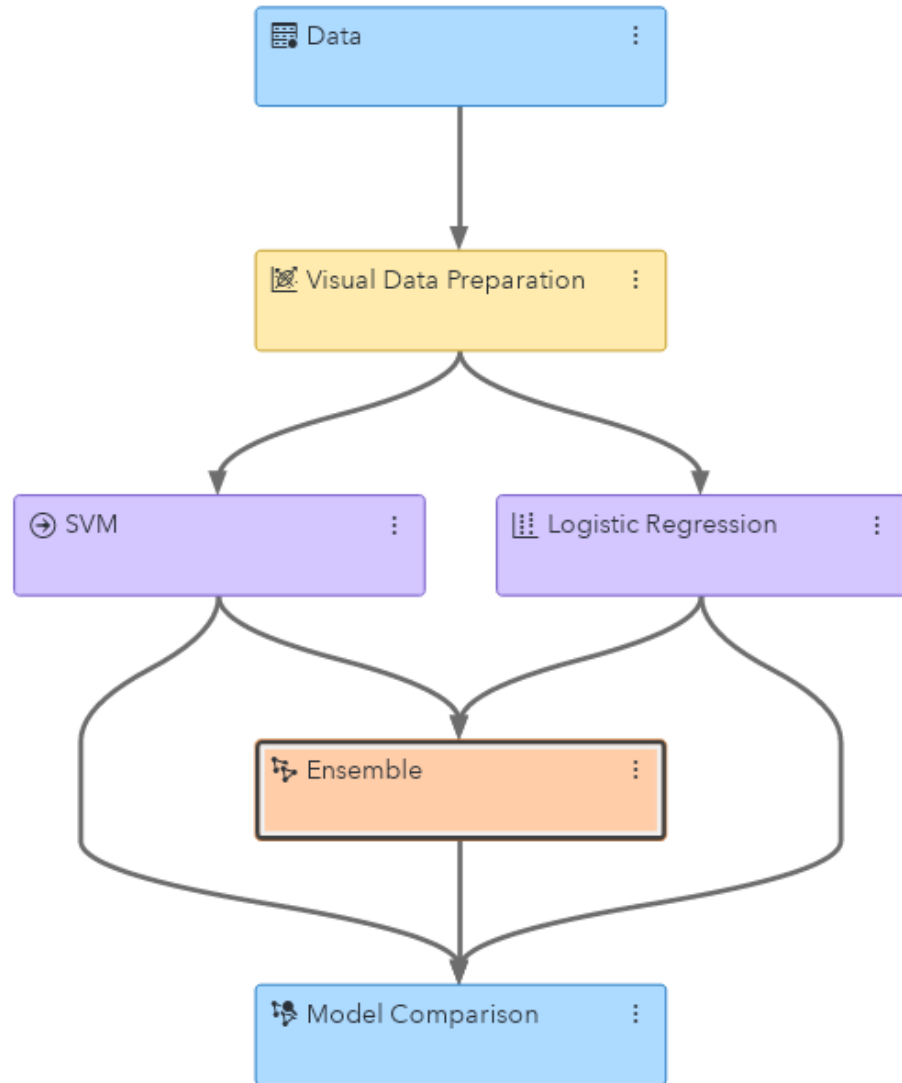
Visual Data Preparation node handles the role assignments when you run the pipeline.

Note: If you change the target variable information or the partition variable information on the **Data** tab and try to run your pipeline, it will fail. The score code that is generated and applied in the **Visual Data Preparation** node requires the partition and variable information that was known in SAS Visual Analytics when you created the pipeline. After the project is created (the project advisor code is run), you cannot modify any data item on the **Data** tab. Therefore, if you altered a data item and your pipeline failed, you need to delete your Model Studio project. Open your SAS Visual Analytics project, create a new pipeline, and try any modifications again.

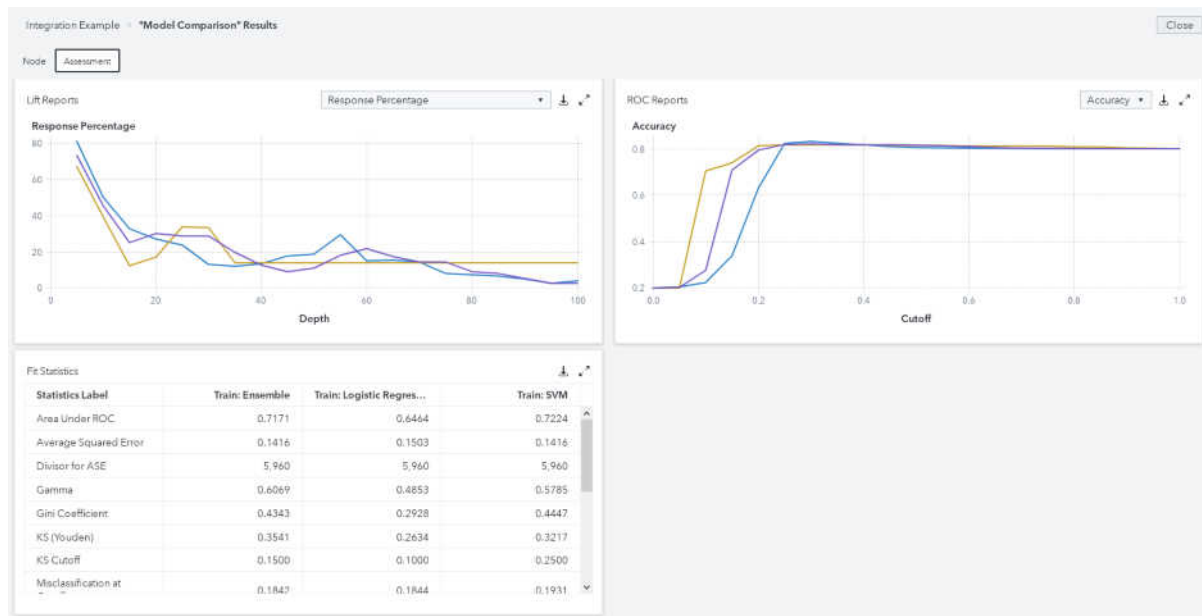
- 4 In the upper left corner, click **Pipelines** to open the **Pipelines** tab.

The Visual Data Preparation node ensures that all subsequent nodes see the data as it existed in SAS Visual Analytics. This node executes all of the model score code from all of the objects in SAS Visual Analytics that were used to create the copied model. This code can be DS1 code, analytic store code, or some combination of the two. The Visual Data Preparation node ensures that this code runs in the proper order and guarantees that subsequent nodes receive the proper information. This means that the Logistic Regression node that you add in the next step sees the data as it existed in SAS Visual Analytics.

- 5 Right-click the **Visual Data Preparation** node and select **Add child node** ⇒ **Supervised learning** ⇒ **Logistic Regression**. This adds a new supervised learning model to the pipeline. The data preparation steps used in SAS Visual Analytics are applied to the **Logistic Regression** node.
- 6 Right-click the **Logistic Regression** node and select **Add child node** ⇒ **Postprocessing** ⇒ **Ensemble**.
- 7 Right-click the **Ensemble** node and select **Add Models** ⇒ **SVM**. At this point, your pipeline should resemble the following:



- 8 Right-click the **Model Comparison** node and select **Run**. This action runs all of the nodes preceding the **Model Comparison** node.
- 9 Right-click the **Model Comparison** node and select **Results**. This brings you to the **Node** tab, which displays the **Model Comparison** table. The table shows you which model is your champion model. Click **Assessment** in the upper left corner of the Model Comparison Results window to see the **Lift Reports** and **ROC Reports** charts, as well as the **Fit Statistics** table.



Review the results computed for each of the three models in the diagram. The **Ensemble** node was chosen as the champion model.

Click **Close** to exit the Results window.

- 10 Suppose you want to modify the support vector machine model to improve model performance. Right-click the **SVM** node and select **Enable Properties**. Click **Yes** when the Enable Properties window appears.

Note: If you enable the properties of a node created in SAS Visual Analytics, the model is retrained, which might yield new results. Once properties have been enabled, the model cannot be switched back to use the original SAS Visual Analytics score code.

- 11 Click the **SVM** node. In the **Options** pane, enable **Perform Autotuning**. Autotuning automatically chooses the optimal values for the hyperparameters used to build your predictive model. For more information, see [Overview of Autotuning](#).
- 12 Right-click the **Model Comparison** node and select **Run**. This action runs all of the nodes preceding the **Model Comparison** node.
- 13 Right-click the **Model Comparison** node and select **Results**. Notice the improved misclassification rate of the **SVM** node model.
- 14 Click **Close** to exit the Results window.

Open Source Code Node Example

Overview


In this section, you use the **Open Source Code** node to create a logistic regression model in R. You also compare this model with a **Logistic Regression** node. Though a champion is chosen, the intent of the example is not to build the best model. Rather, this example demonstrates how models from Python or R are executed and compared in Model Studio.

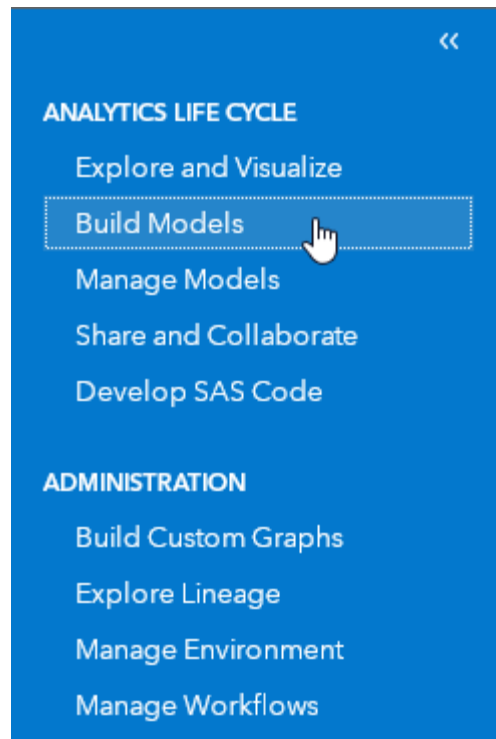
This example uses the HMEQ data set, which has a binary target that indicates whether a mortgage loan defaulted. The inputs include information such as loan amount, reason for loan, years at present job, and debt-to-income ratio.

If you are unsure about whether the HMEQ data set is available on your system, navigate to the [SAS Viya Example Data Sets](#) page and download **hmeq.csv**. Note where you saved this file.

Create the Project and Import the Input Data

This example assumes that you are signed in to SAS Drive. To create the project that you use in this example:

- 1 In the upper left corner of the SAS Drive window, click , and select **Build Models**.



- 2 Select **New Project** in the upper right corner of the page.
- 3 Enter `Open Source Code Node Example` for **Name** in the New Project window.

✕
New Project

Name: *

Type: *

Data Mining and Machine Learning ▼

Template:

Blank template ▼

Data: *

Browse

Description:

Advanced

Save

Cancel

- 4 Select **Data Mining and Machine Learning** for **Type**.
- 5 Ensure that **Blank Template** is specified for **Template**.
- 6 In the **Data** field, select **Browse**. The Choose Data window appears.
- 7 If **HMEQ** is listed on the **Available** tab of the Choose Data window, select the HMEQ data set and click **OK**.
 If the HMEQ data set is not listed on the **Available** tab, import it:
 - a Click the **Import** tab.
 - b Select **Local files** ⇒ **Local file**.
 - c In the Open window, navigate to the location where you saved the HMEQ data set and select **hmeq.csv**.
 - d Click **Open**.
 - e In the upper right corner, click **Import Item**.
 - f After the data set is successfully imported, click **OK**.
- 8 Click the **Advanced** button below the **Description** text box, and the New Project Settings window appears. Select **Partition Data** in the upper left corner of the window.

New Project Settings

Advisor Options
Partition Data
 Event-Based Sampling
 Node Configuration

Partition Data

☒ Create partition variable

Note: These settings are active only when a partition variable is not set within the data. Using a data source with a pre-defined partition variable or manually selecting a partition variable will override these settings.

Method:
 Stratify

Training:
 60 60.00%

Validation:
 30 30.00%

Test:
 10 10.00%

Save Cancel

- 9 Ensure that the **Create partition variable** option is selected, and click **Save** in the lower right corner of the window. This brings you back to the New Project window.
- 10 In the lower right corner of the New Project window, click **Save**. You are redirected to the **Data** tab, where you can modify the variables in your data set.

Modify Variables

On the **Data** tab, variable roles are indicated in the **Role** column. To change the role of a variable:

- 1 Select a variable by clicking the corresponding check box to the left of the **Variable Name** column. The options pane for the selected variable appears to the right of the variable table on the **Data** tab.
- 2 Expand the drop-down list under **Role**, and select the role type that you want to assign to the selected variable. Changes made to each variable are automatically applied and saved.

CAUTION

To avoid making unwanted changes to variable properties, you must manually deselect each variable that you modify when you are finished making changes to its properties.

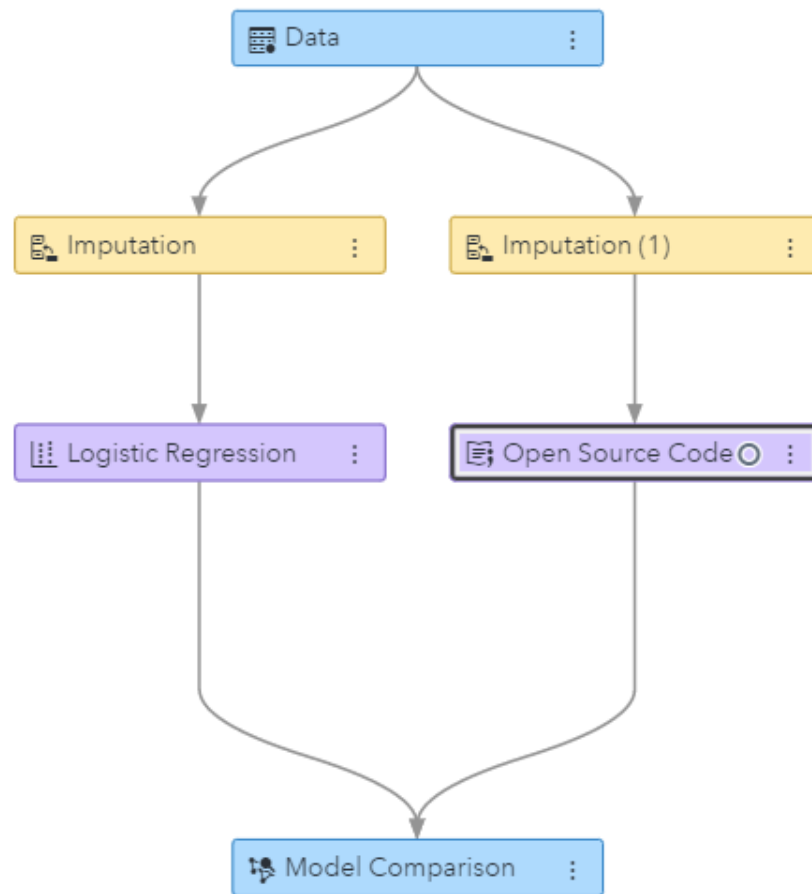
- 3 Using the steps above, adjust the property of **Role** for each of the following variables:
 - Set BAD to **Target**.
 - Ensure that all other variables are set to **Input**.

Create the Pipeline

This example requires you to complete the steps in the previous sections. This example also assumes that you have not created any other pipelines before starting this section. To create the pipeline that contains an open-source model:

- 1 Navigate to the **Pipelines** tab. This tab should contain a single pipeline with only a **Data** node.
- 2 Right-click the **Data** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Imputation**.
- 3 Right-click the **Imputation** node and select **Add child node** ⇒ **Supervised Learning** ⇒ **Logistic Regression**.
- 4 Right-click the **Data** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Imputation**. This ensures that all missing data is imputed because some open-source packages cannot handle missing data.
- 5 Right-click the **Imputation** node that you added in step 4 and select **Add child node** ⇒ **Miscellaneous** ⇒ **Open Source Code**.
- 6 Right-click the **Open Source Code** node and select **Move** ⇒ **Supervised Learning**. This ensures that the node performs model assessment and can be compared to the **Logistic Regression** node.

Your current pipeline should resemble the following image.




- 7 Select the **Imputation** node that is above the **Open Source Code** node. In the options pane, select **Impute non-missing variables**.
- 8 Select the **Open Source Code** node. In the options pane, set the value of **Language** to **R**.
- 9 In the options pane, click **Open Code Editor**. Enter the following code in the code editor:

```
# Build logistic regression model
dm_model <- glm(BAD ~ IMP_CLAGE + IMP_CLNO + IMP_DEBTINC +
  IMP_LOAN + IMP_MORTDUE + IMP_VALUE + IMP_YOJ,
  data=dm_traindf, family=binomial(link="logit"))

# View model output
summary(dm_model)

# Score
pred <- plogis(predict(dm_model, dm_inputdf))
dm_scoreddf <- data.frame(pred)
colnames(dm_scoreddf) <- c("P_BAD1")
dm_scoreddf$P_BAD0 <- 1 - dm_scoreddf$P_BAD1
```

- 10 In the upper right corner of the code editor, click .
- 11 Click **Close**.
- 12 Right-click the **Model Comparison** node and select **Run**.

13 After the pipeline has successfully run, right-click the **Open Source Code** node and select **Results**.

The screenshot shows the 'Open Source Code' Results window in SAS Model Studio. The 'Node' tab is selected, displaying the following content:

R Code

```

1 #-----
2 # Language: R
3 #-----
4 # Set R-work to node-work directory
5 dm_nodedir <- '/opt/sas/viya/config/var/tmp/compsrv/default/e0fea'
6 setwd(dm_nodedir)
7
8 # Variable declarations
9 dm_dec_target <- 'BAD'
10 dm_partitionvar <- '_PartInd_'
11 dm_partition_train_val <- 1
12
13 dm_class_input <- c("IMP_DELIQ", "IMP_DEROG", "IMP_JOB", "IMP_NINQ")
14 dm_interval_input <- c("IMP_CLAGE", "IMP_CLNO", "IMP_DEBTINC", "IMP_I
15

```

R Output

```

1
2 Call:
3 glm(formula = BAD ~ IMP_CLAGE + IMP_CLNO + IMP_DEBTINC + IMP_LOAN
4     IMP_MORTDUE + IMP_VALUE + IMP_YOJ, family = binomial(link = "l
5     data = dm_traindf)
6
7 Deviance Residuals:
8      Min       1Q   Median       3Q      Max
9  -1.1035   -0.7089   -0.5776   -0.4000    3.9216
10
11 Coefficients:
12              Estimate Std. Error z value Pr(>|z|)
13 (Intercept) -1.765e+00  2.637e-01  -6.694 2.17e-11 ***
14 IMP_CLAGE   -6.307e-03  6.544e-04  -9.637 < 2e-16 ***
15

```

Properties

Property Name	Property Value
osCode_Language	r
sampMethod	STRATIFY
sampType	NUM
sampObs	10,000
sampPercent	10
exportWithFormat	true

Output

Predicted Variables							
Obs	Predicted Variable Name	Variable Name	Level Frequency	Level Frequency Percent	Raw Numeric Value	Raw Character Value	Formatted Value
1	P_BAD1	BAD	1189	19.9497	1		1
2	P_BAD0	BAD	4771	80.0503	0		0
3	I_BAD	BAD

- a Expand the **R code** results to view the actual code that was generated by Model Studio and submitted. Notice that this code is a combination of precursor, user, and posterior code. The precursor and posterior code is added based on the node properties and whether the node is in the **Preprocessing** or **Supervised Learning** group.
 - b Expand the **R Output** results to view the input variables that are significant in the model.
 - c On the **Assessment** tab, notice that assessment measures such as lift and ROC were computed for the open-source model.
 - d Close the Results window.
- 14** Right-click the **Model Comparison** node and select **Results**. The SAS logistic regression model was chosen as the champion model based on having a better Kolmogorov-Smirnov statistic.
- The **Assessment** tab lets you compare the results of the open-source model against the SAS logistic regression model .
- 15** Close the Results window. The example is now complete.

Risk Modeling Example

Risk Modeling Overview

Note: Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning is not included with SAS Visual Data Mining and Machine Learning. If your site has not licensed Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning, the risk modeling nodes do not appear in your SAS Visual Data Mining and Machine Learning software.

The Risk Modeling Add-on for SAS Visual Data Mining and Machine Learning is used to evaluate the level of risk associated with applicants or customers. It provides statistical odds, or probability, that an applicant with any given score will be good or bad. In its simplest form, a scorecard is built from a number of characteristics (that is, input or predictor variables). Characteristics are initially grouped into attributes. For example, age is a characteristic, and 25-33 is an attribute. Each attribute is associated with a number of scorecard points. These scorecard points are statistically assigned to differentiate risk, based on the predictive power of the characteristic variables, correlation between the variables, and business considerations.

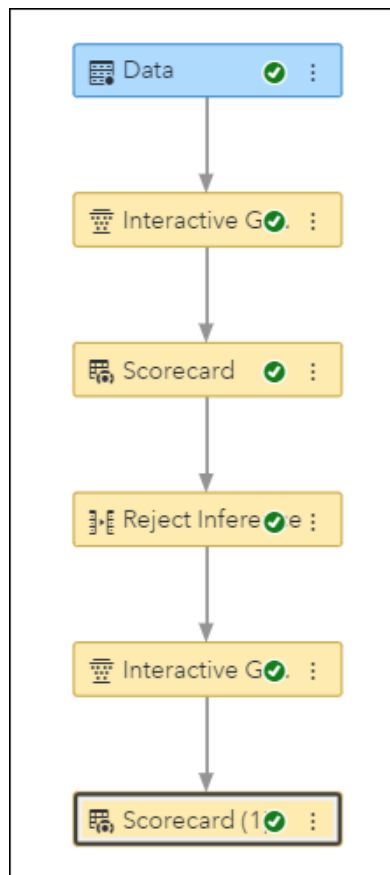
The accepts data set is used to train the initial scorecard model. It consists of just the credit applicants who have been previously extended credit. The accepts data set labels the applicants as event or nonevent (that is, good or bad). The rejects data set is used to remedy selection bias in your data. The rejects data contains credit applicants that were denied credit. The rejects data is scored using the initial scorecard to infer whether the rejected applicants would have been good or bad credit risks. These observations are added to the accepts data and this augmented data set is used to build the final scorecard model.

This example uses accepts and rejects data sets that are called `rm_accepts` and `rm_rejects`. You can download the data for this example from the [SAS Visual Data Mining and Machine Learning product page](#). Click **Data for Risk Modeling Example**. Download the ZIP file and extract `rm_accepts.SAS7BDAT` and `rm_rejects.SAS7BDAT` to a directory that your SAS Visual Data Mining and Machine Learning server can access.


To build a scorecard model in this example:

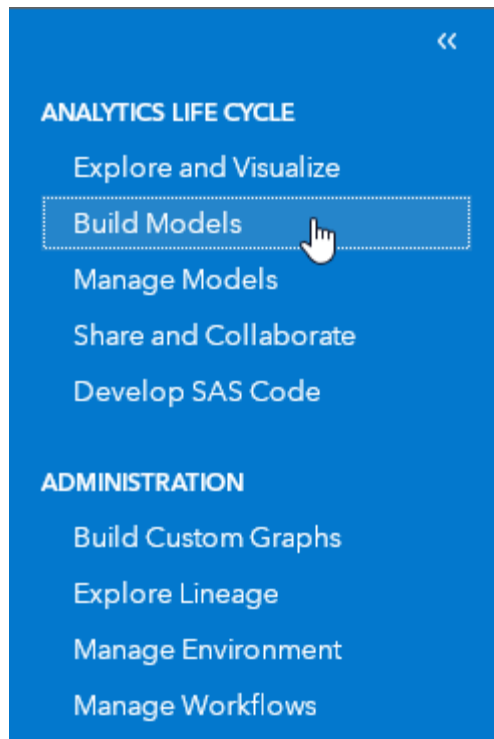
- 1 Group the characteristic variables into attributes.
- 2 Use a logistic regression model to create an initial scorecard.
- 3 Perform reject inference on the initial model.
- 4 Create the final scorecard using the information that is obtained in the previous steps.

The finished pipeline diagram will resemble the one that is shown here:



Create the Project and Import the Input Data

This example assumes that you are signed in to SAS Drive. In the upper left corner of the SAS Drive window, click  and select **Build Models**.



You are directed to the Projects page. To create the project that you will use in this example:

- 1 Select **New Project** in the upper right corner of the Projects page.
- 2 Enter **Risk Modeling Example** for **Name** in the New Project window.

New Project

Name: *

Type: *

Template:

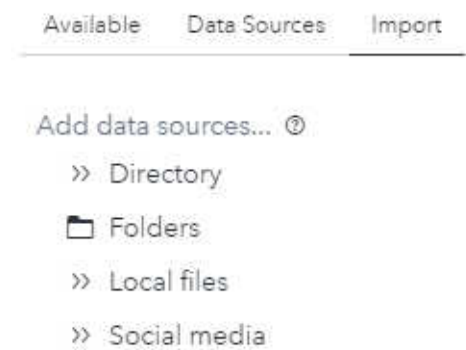
Data: *

Description:

- 3 Select **Data Mining and Machine Learning** for **Type**.

Note: **Forecasting** and **Text Analytics** are additional options if you have licensed those products.

- 4 Ensure that **Blank template** is specified for **Template**.
- 5 In the **Data** field, click **Browse**. The Choose Data window appears.
- 6 In the upper left corner of the Choose Data window, select **Import**.



- 7 Select **Local files** ⇒ **Local file** and navigate to the folder where `rm_accepts` is stored. Select `rm_accepts.sas7bdat` and click **Open**.
- 8 Click **Import Item** in the upper right corner of the Choose Data window. When the data is successfully imported, a note appears, saying that the data is ready for use.

✓ The table was successfully imported on Oct 7, 2019 10:33 AM and is ready for use.

- 9 When the data set is successfully imported, click **OK** in the lower right corner of the Choose Data window. This returns you to the New Project window.
- 10 Click **Save** in the New Project window. This directs you to the **Data** tab, where you can modify the variables in your data set.

Modify Variables

On the **Data** tab, variable roles are indicated in the **Role** column. To change the role of a variable:

- 1 Select a variable by clicking the corresponding check box to the left of the **Variable Name** column. The options pane for the selected variable appears to the right of the variable table on the **Data** tab.
- 2 Expand the drop-down list under **Role**, and select the role type that you want to assign to the selected variable. Changes made to each variable are automatically applied and saved.

CAUTION

To avoid making unwanted changes to variable properties, you must manually deselect each variable that you modify when you finish changing its properties.

- 3 Using the steps above, adjust the property of **Role** for each of the following variables:
 - Set GB to **Target**.
 - Ensure that all other variables are set to **Input**.

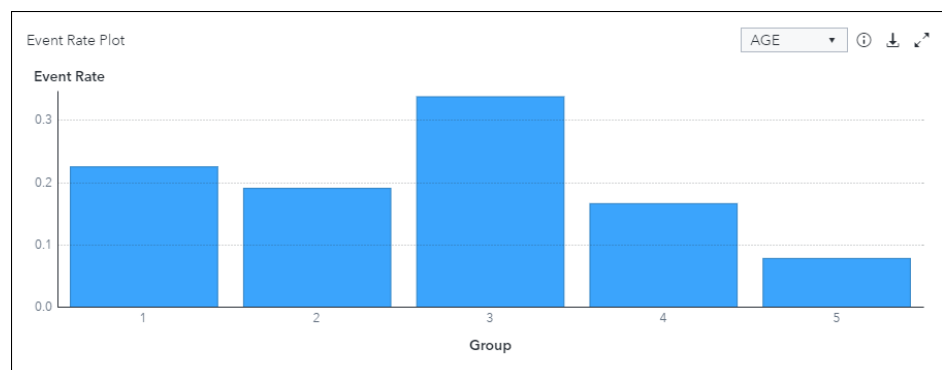
Group the Characteristic Variables into Attributes

You first want to group the characteristic values into attributes that can be used as input variables for your scorecard model:

- 1 Click the **Pipelines** tab in the upper left corner.
- 2 Right-click the **Data** node and select **Run**.
- 3 When the **Data** node has run successfully, right-click the **Data** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Interactive Grouping**.
- 4 Right-click the **Interactive Grouping** node and select **Run**.
- 5 When the node has run successfully, right-click the node and select **Results** to view detailed information about the variable groups that were created:
 - **Output Variables** — Displays all characteristics and their corresponding information value (IV) and Gini statistic. IV is a measure of the predictive value of a characteristic and the Gini statistic is used to measure how equal the event rates are across the attributes of a characteristic. In this example, characteristics with an IV greater than 0.1 are used as input variables in subsequent modeling steps. All other attributes are rejected. In the options pane, the **Variable Selection Options** enable you to adjust the variable selection method and cutoff values.

Variable	Gini Statistic	Information Value	Level for Interactive	Calculated Role
AGE	33.3240	0.3780	INTERVAL	INPUT
STATUS	23.9680	0.2330	NOMINAL	INPUT
INC1	24.0280	0.2040	NOMINAL	INPUT
TMJOB1	22.3940	0.1950	INTERVAL	INPUT
CARDS	18.3330	0.1680	NOMINAL	INPUT
INCOME	22.4510	0.1650	INTERVAL	INPUT
INC	20.9420	0.1560	NOMINAL	INPUT
PERS_H	19.8870	0.1530	NOMINAL	INPUT

- **Event Rate Plot** — Displays the event rate for each variable group.



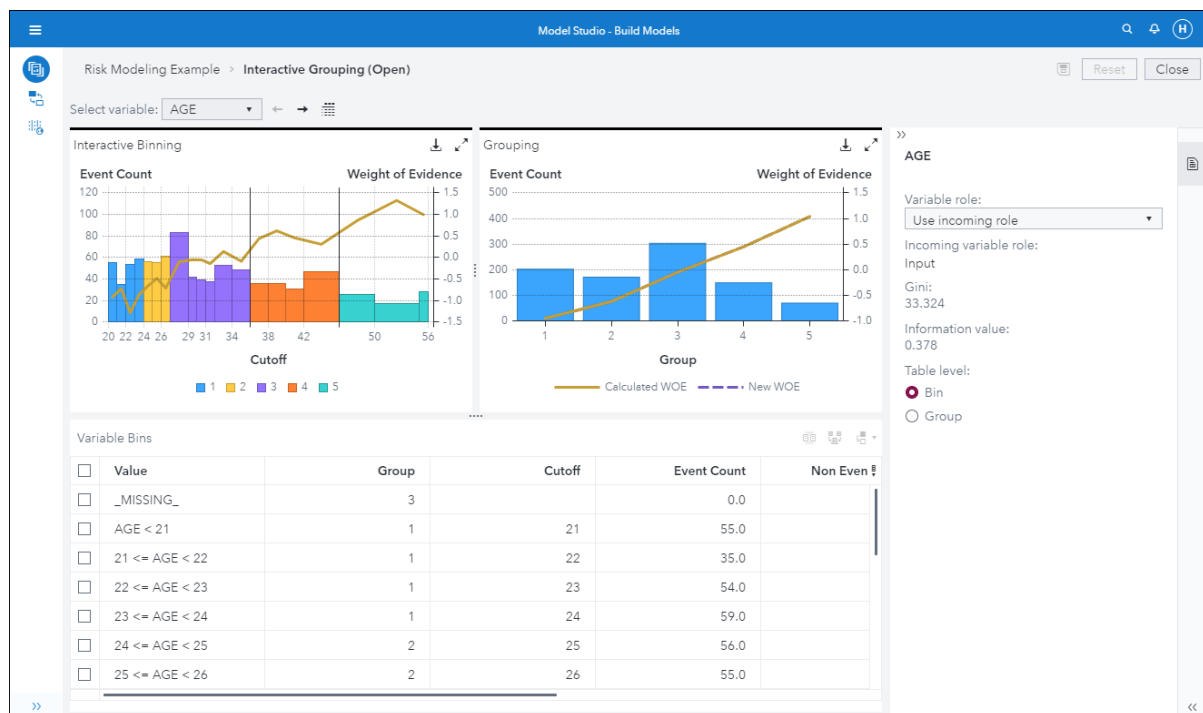
- **Statistics Table** — Displays several statistics such as the event count, nonevent count, and weight of evidence (WOE) for each variable group.

Variable	Group	Event Count	Non-Event Count	Weight of Evidence
AGE	1	203	79	-0.9438
AGE	2	172	93	-0.6149
AGE	3	304	291	-0.0437
AGE	4	150	236	0.4532
AGE	5	71	201	1.0406
BUREAU	1	610	592	-0.0300
BUREAU	2	290	308	0.0602
CAR	1	245	150	-0.4906

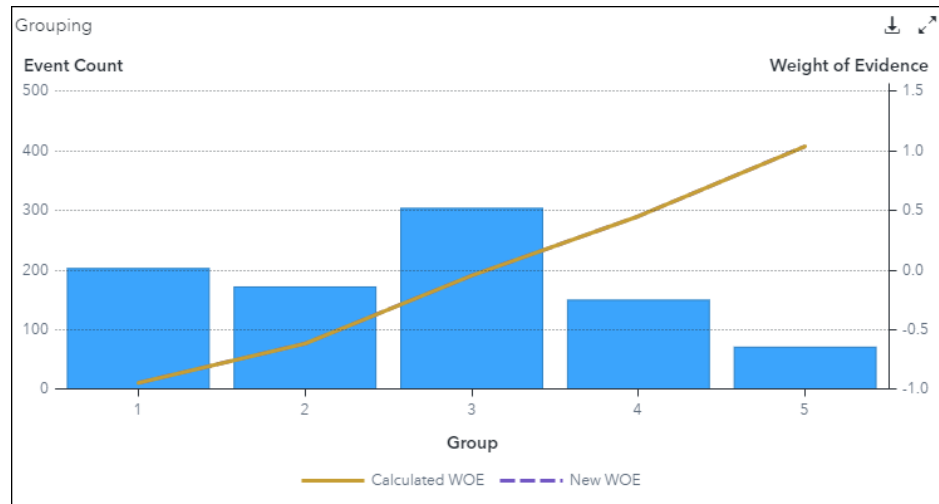
6 Close the **Results** window.

7 After running the node and viewing the results, you might decide that you want to adjust the variable groups that were generated for operational or business reasons. To open the Interactive Grouping editor and manually adjust variable groups, right-click the **Interactive Grouping** node, and select **Manage Group Bins**.

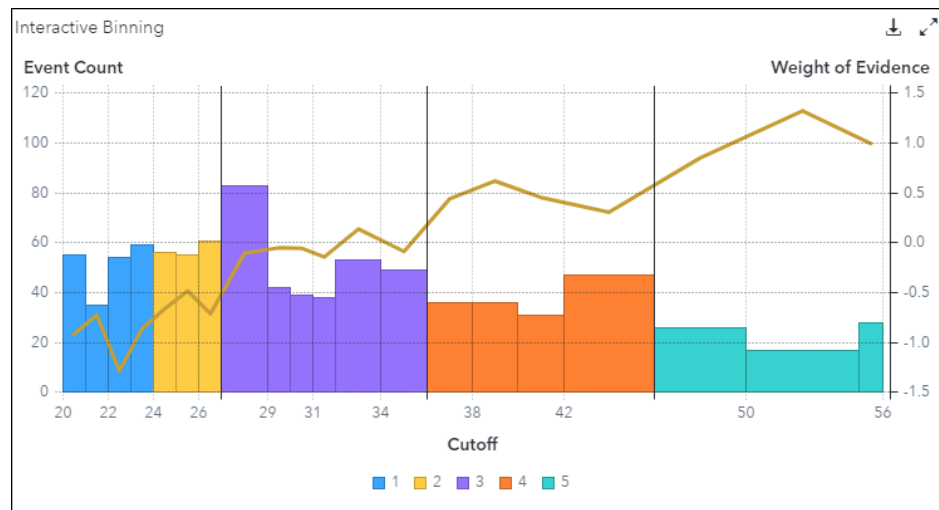
The Interactive Grouping editor opens:



The Grouping chart displays the event count and WOE for each group of the characteristic AGE. WOE measures the strength of an attribute of a characteristic in differentiating good risk and bad risk applicants. Negative values indicate that the particular group contains a higher proportion of bad risk applicants than good risk applicants. That is, negative WOE values indicate that the applicants in the group present a greater risk.



The Interactive Binning chart displays the event count and WOE for each bin of the variable AGE.



- 8 The Interactive Grouping editor enables you to split, merge, or change the groups of the selected bins. Each time you adjust a variable bin, the Gini statistic and IV of the characteristic are updated in the right pane.

To split a bin:

- Select **46 ≤ AGE < 50**.
- Click . The Split Bin window appears.
- Enter **48** for the **New Cutoff Value** and click **Save**. Group 5 now has an additional bin with a cutoff value of 48.


To merge bins:


- Select the three bins in group 2.
- Click . Group 2 now contains just one bin.

To change the group of selected bins:

- Select the last two bins in group 3.

- b Click  and select **Create new group**. There are now six groups.

Note: When you click , you can also merge the selected bins into existing groups.

- 9 Click  and then click **Close** to exit the Interactive Grouping editor.

Build an Initial Scorecard

Next, you want to use the grouped characteristics to build your initial scorecard. You can choose to use the WOE or GRP variables that were generated by the **Interactive Grouping** node for the analysis. The WOE variables contain the weight of evidence of each binned variable and the GRP variables contain the group ID.

The **Scorecard** node generates the scorecard with logistic regression and scaling. The regression coefficients are used to scale the scorecard. Scaling makes the scorecard conform to a particular range of scores. The two main elements of scaling a scorecard are determining the odds at a certain score and determining the points that are required to double the odds. The **Odds**, **Scorecard Points**, and **Points to Double the Odds** properties control the scaling.

This example uses the default settings for the logistic regression and scaling:

- 1 Right-click the **Interactive Grouping** node and select **Add child node** ⇒ **Miscellaneous** ⇒ **Scorecard**.
- 2 In the options pane, set **Scorecard type** to **Detailed**.
- 3 Right-click the **Scorecard** node and select **Run**.
- 4 When the node has run successfully, right-click the node and select **Results** to view detailed information in each of the following sections of interest:
 - **Scorecard** — Displays the scorecard points, WOE, event rate (percentage of bad risk applicants in that score range), percent (percentage of bad risk applicants that have a score higher than the lower limit of the score range), and the regression coefficient for each attribute.

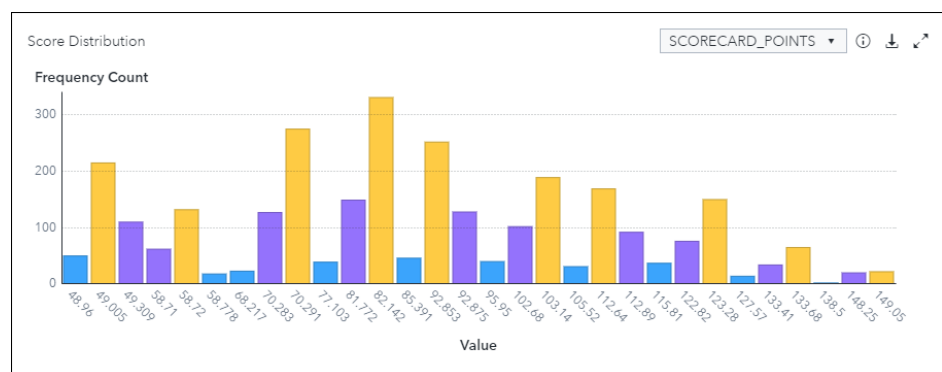
Scorecard

		Group	Scorecard Points	Weight of Evidence	Event Rate GB = 1	Percent	Coefficient
Age	AGE < 24	1.00	5	-0.94	72.75	15.90	-0.63
	24 <= AGE < 27	2.00	11	-0.61	63.88	13.93	-0.63
	27 <= AGE < 32, _MISSING_	3.00	20	-0.09	52.40	20.80	-0.63
	32 <= AGE < 36	4.00	22	0.04	46.74	11.77	-0.63
	36 <= AGE < 46	5.00	30	0.45	39.91	22.13	-0.63
	46 <= AGE	6.00	41	1.04	27.80	15.47	-0.63
Credit Cards	AMERICAN EXPRESS, NO CREDIT CARDS, VISA MYBANK, VISA OTHERS, _MISSING_						

- **Empirical Odds Plot** — Displays the empirical odds plotted against the average values of the scorecard points. The negative slope implies that good risk applicants have larger scores than bad risk applicants.



- **Score Distribution** — Located on the **Assessment** tab. Displays the distribution of the scorecard points, the event odds, and the log of the event odds.



- **Gains Table** — Located on the **Assessment** tab. Displays several statistics for each score bucket.

Bucket	Data Role	Score Bucket	Count	Event Count
25	TRAIN	Score >= 140	29	1
24	TRAIN	136 <= Score < 140	15	2
23	TRAIN	132 <= Score < 136	1	0
22	TRAIN	128 <= Score < 132	60	11
21	TRAIN	124 <= Score < 128	55	12
20	TRAIN	120 <= Score < 124	77	21
19	TRAIN	116 <= Score < 120	12	4
18	TRAIN	112 <= Score < 116	114	39

- **Average Predicted Probability** — Located on the **Assessment** tab. As the score increases, the predicted probability of a bad risk applicant decreases.



- 5 Close the **Results** window.

Perform Reject Inference

The accepts data set that is used to develop a credit scoring model is structurally different from the through-the-door population to which the credit scoring model is applied. The number of events and nonevents in the target variable that is created for the credit scoring model is based on the records of applicants who were all accepted for credit. However, the population to which the credit scoring model is applied includes applicants who would have been rejected under the scoring rules that were used to generate the initial model.

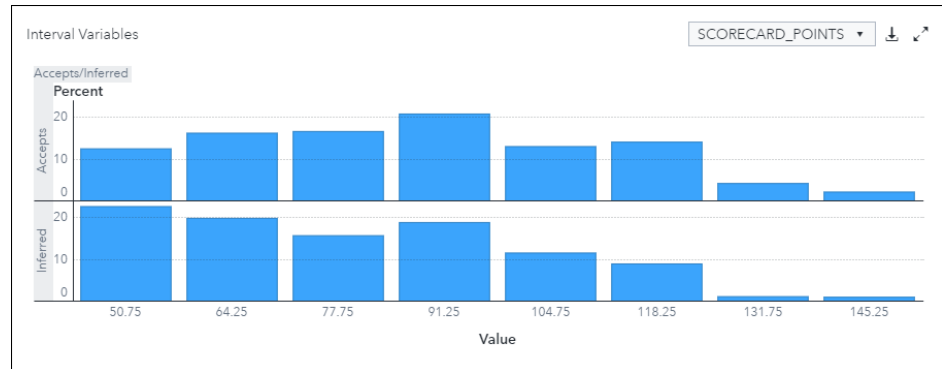
The reject inference approach uses the model that was trained using the accepted applications to score the rejected applications. The observations in the rejects data set are classified as inferred events and inferred nonevents. The inferred observations are then added to the accepts data set, which contains the actual event and nonevent records, to form an augmented data set. This augmented data set, which represents the through-the-door population, serves as the training data set for a second scorecard model.

To perform reject inference:

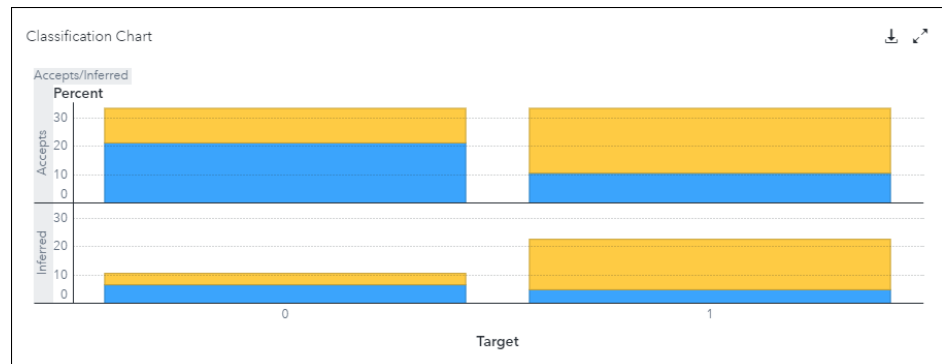
- 1 Right-click the **Scorecard** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Reject Inference**.
- 2 Import the rejects data set:
 - a In the options pane, under **Score data table name**, click **Browse**. The Choose Data window appears.
 - b In the upper left corner of the Choose Data window, select **Import**.
 - c Select **Local files** ⇒ **Local file** and navigate to the folder where rm_rejects is stored. Select **rm_rejects.sas7bdat** and click **Open**.
 - d Click **Import Item** in the upper right corner of the Choose Data window. When the data is successfully imported, a note appears, saying that the data is ready for use.
 - e When the data set is successfully imported, click **OK**.
- 3 In the options pane, set **Inference method** to **Parceling**. The parceling method distributes the scored rejects into equal sized buckets that are defined by the **Score range method** that you specify. This example uses the default method to define the buckets based on the score range of the rm_accepts data set. The scored rejects are then randomly classified as an event or nonevent.
- 4 In the options pane, set **Event rate increase** to 1.2. The proportion of event and nonevents in the rm_rejects data set is not expected to approximate the proportion of event and nonevents in the rm_accepts data set. You would expect that the event rate of the rm_rejects data set is higher than that of the rm_accepts data set. A value of 1.2 specifies that the event rate in the rejects data should be 20% greater than what is observed in the accepts data.
- 5 Right-click the **Reject Inference** node and select **Run**.

- 6 When the node has run successfully, right-click the node and select **Results** to view detailed information in each of the following sections of interest:

- **Interval Variables** — Displays the distribution of the scorecard points, probability of the event, and probability of the nonevent for the accepts and inferred data sets.



- **Classification Chart** — Displays the distribution of the predicted event and nonevent for the accepts and inferred data sets.



- 7 Close the **Results** window.

Build the Final Scorecard

To build the final scorecard, you must repeat the grouping and scorecard creation steps on the augmented data set:

- 1 Right-click the **Reject Inference** node and select **Add child node** ⇒ **Data Mining Preprocessing** ⇒ **Interactive Grouping**.
- 2 Right-click the **Interactive Grouping** node and select **Add child node** ⇒ **Miscellaneous** ⇒ **Scorecard**.
- 3 In the options pane, set **Scorecard type** to **Detailed**.
- 4 Right-click the **Scorecard** node and select **Run**.
- 5 When the node has run successfully, right-click the node and select **Results** to view detailed information about the final scorecard that is based on both accepted and rejected applicants.

Scorecard

		Group	Scorecard Points	Weight of Evidence	Event Rate GB = 1	Percent	Coefficient
Age	AGE < 24	1.00	-5	-1.27	81.08	20.09	-0.69
	24 <= AGE < 28	2.00	9	-0.59	67.86	17.98	-0.69
	28 <= AGE < 37, _MISSING_	3.00	22	0.11	53.74	28.82	-0.69
	37 <= AGE < 49	4.00	35	0.74	39.59	21.44	-0.69
	49 <= AGE	5.00	43	1.16	29.52	11.67	-0.69
Credit Cards (CARDS)	AMERICAN EXPRESS, NO CREDIT CARDS, VISA CITIBANK, VISA MYBANK, VISA OTHERS, _MISSING_, _UNKNOWN_	1.00	13	-0.24	61.75	75.76	-0.99

6 Close the **Results** window.

Managing Projects

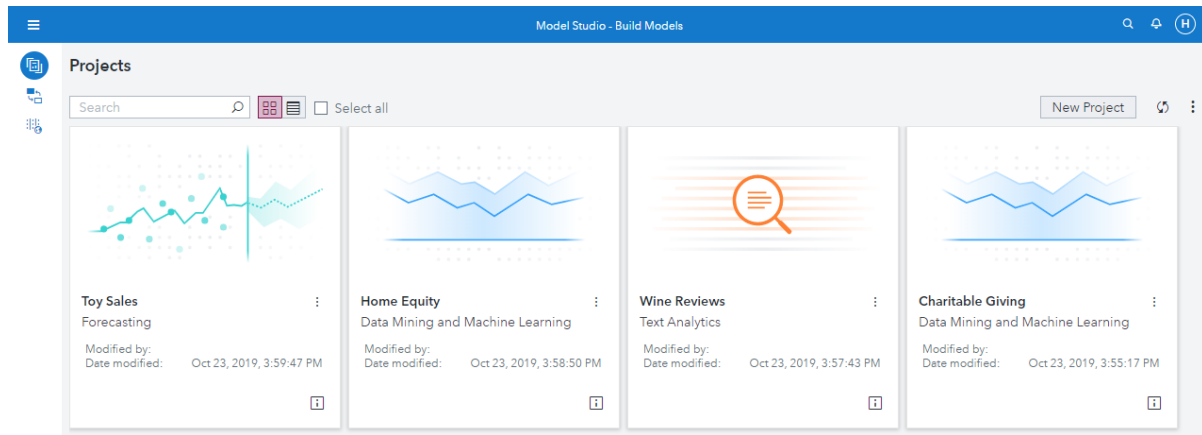
<i>Overview of Model Studio Projects</i>	65
<i>Opening an Existing Project</i>	67
<i>Creating a New Project</i>	68
<i>Sharing a Project</i>	69
<i>Importing and Exporting a Project</i>	72
<i>Deleting a Project</i>	73
<i>Downloading Project Batch API Code</i>	74
<i>Specifying Global Settings</i>	76
<i>Specifying Project Settings</i>	78
<i>Importing a Project from SAS Visual Analytics</i>	80

Overview of Model Studio Projects



A *project* is a top-level container for your analytic work in Model Studio. You can view projects in the Model Studio Projects page.

Model Studio projects can be one of three types: Forecasting projects, Data Mining and Machine Learning projects, and Text Analytics projects. The project types that appear in your Model Studio installation depend on the SAS licensing for your site.

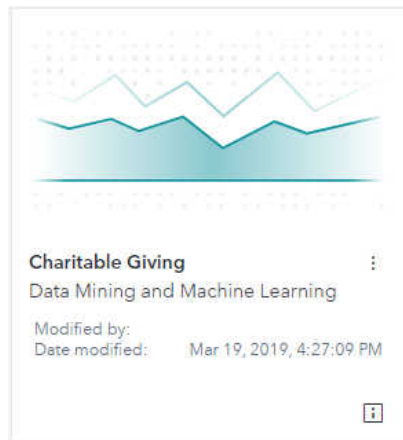
Depending on your project filter setting, existing projects in your environment appear either as graphic tiles or rows in a table of projects.




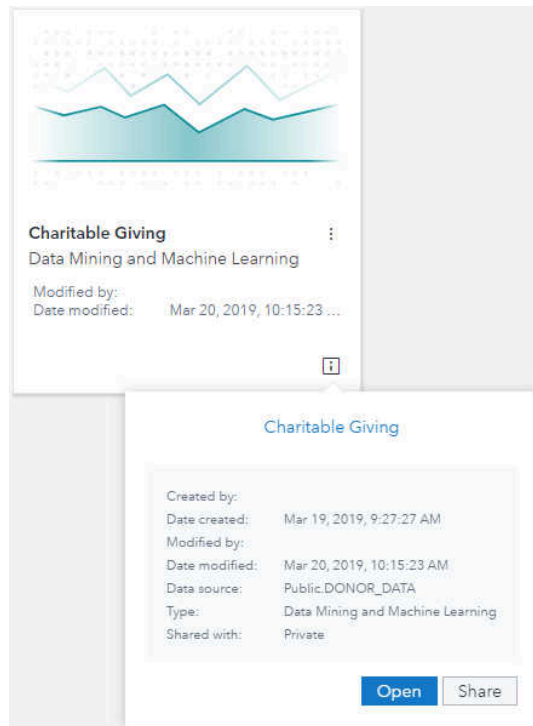
To alternate between table and tile project displays:

- Click  near the top of the page to show existing Model Studio projects in a graphic tile matrix.
- Click  near the top of the page to show Model Studio projects in a tabular list.

A Model Studio project contains the data source, the pipelines that you create, and related project metadata (such as project type, project creator, share list, and last update history). If you create more than one pipeline in your project, analytic results that compare the performance of multiple pipelines are also stored in the project.

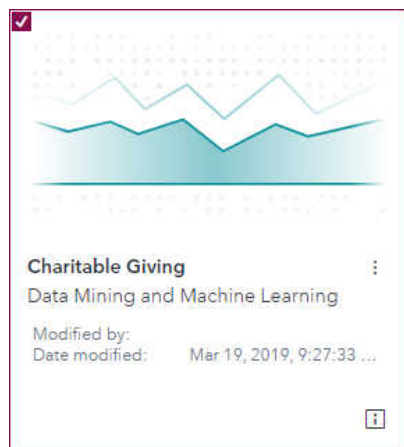


Click  in the lower right corner of the project tile to view additional information about the project.



Opening an Existing Project

You use the Model Studio Projects page to access existing projects. If your Model Studio Projects page displays project tiles, double-click the tile that you want to open, or click the link to the project name within the tile. Model Studio opens the selected project.



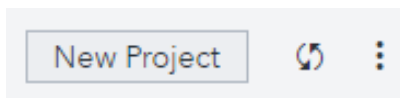
If your Model Studio Projects page displays a project table, double-click the row that contains the desired project, or click the link of the project name to open it. Model Studio opens the selected project to the last visited tab.

Search							New Project		
<input type="checkbox"/>	Name	Type	Modified By	Created By	Shared With	Data Source	Date Modified		
<input type="checkbox"/>	Toy Sales	Forecasting	Holly Sweeney	Holly Sweeney	Private	Public.INSIGHTTOYDEMO	Mar 20, 2019, 9:35:35 AM		
<input type="checkbox"/>	Home Equity	Data Mining and Machine Learning	Holly Sweeney	Holly Sweeney	Private	Public.HMEQ	Mar 20, 2019, 9:31:44 AM		
<input type="checkbox"/>	Wine Reviews	Text Analytics	Holly Sweeney	Holly Sweeney	Private	Public.WINE_QUALITY_REVIEWS	Mar 20, 2019, 9:28:51 AM		
<input type="checkbox"/>	Charitable Giving	Data Mining and Machine Learning	Holly Sweeney	Holly Sweeney	Private	Public.DONOR_RAW_DATA	Mar 20, 2019, 9:27:46 AM		

Creating a New Project

You create new Model Studio projects from the Projects page. To create a new project:

- 1 Click the **New Project** button in the upper right corner.



- 2 The New Project window appears.

New Project

Name: *

Toy Company

Type: *

Data Mining and Machine Learning

Template:

Blank template

Data: *

Browse

Description:

Advanced

Save

Cancel

Enter a name for your new project in the **Name** field.

- 3 Select a project type from the **Type** list. The choices are **Forecasting**, **Data Mining and Machine Learning**, and **Text Analytics**.
- 4 Select a template to use for your pipeline. *Templates* are pipelines that are pre-populated with configurations that can be used to create models quickly. The default template is the **Blank template**. The **Blank template** is the baseline pipeline that is pre-populated with only the **Data** node. For more information about the templates available in Model Studio, see [Available Templates on page 97](#).

- 5 You must identify the data source that you want to use. A project can have only one data source. However, some nodes can point to additional data sources. Select the **Browse** button to open the Choose Data window. Use the Choose Data window to select your data source and click **OK**. For more information, see [Getting Started with the Choose Data Window](#).

Note: The data source name cannot contain any of the following characters: / \ * ? " < > | : - . &

- 6 If you want to enter information about the project that might be useful to others, enter that content in the **Description** field.
- 7 Click **Advanced** to specify additional project creation options. In the New Project Settings window, you can specify the following settings:
 - **Advisor Options** — These settings enable you to specify settings such as the maximum number of class level values and the maximum percentage of missing values. Another option in this group enables you to control the interval cutoff value for numeric variables. If the number of unique levels exceeds the specified cutoff value, the variable is assigned the measurement level interval.
 - **Partition Data** — These settings enable you to partition the data set into subsets used for training, validation, and test. You can also specify the partition method. After the **Data** node is run, you can view the partition proportions, but you cannot edit them until you reload the data source. For more information, see [Replace the Data Source on page 86](#).
 - **Event-based Sampling** — These settings enable you to specify event-based sampling for the model. All of the observations that have the target event are included, and the observations with non-events are sampled to match the percentages specified for **Event** and **Non-event**.
 - **Node Configuration** — These settings enable you to specify configuration code in Python. This Python code is automatically prepended to every **Open Source Code** that you include in your pipeline when the **Language** property is set to **Python**. The Python code can be a maximum of 4000 characters.
- 8 Click the **Save** button to create your new project using the name, project type, and data source name that you specified.

After you create your new project, Model Studio takes you to the **Data** tab of your new project page. Here, you can make adjustments to data source variable names, labels, type, role, and level assignments. For more information about the **Data** tab, see the [Data Management Overview](#) section.


Sharing a Project

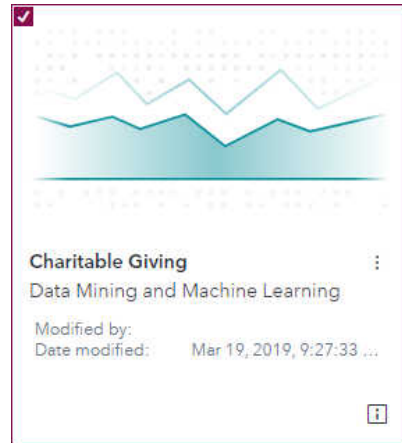
After creating a project, you can share it with others in your organization. Model Studio enables you to share projects with user-defined groups.

The Model Studio implementation of sharing is distinct from project sharing as performed in SAS Drive. Any projects that you share using SAS Drive do not retain the same settings for user groups in Model Studio. Also, any projects that you share

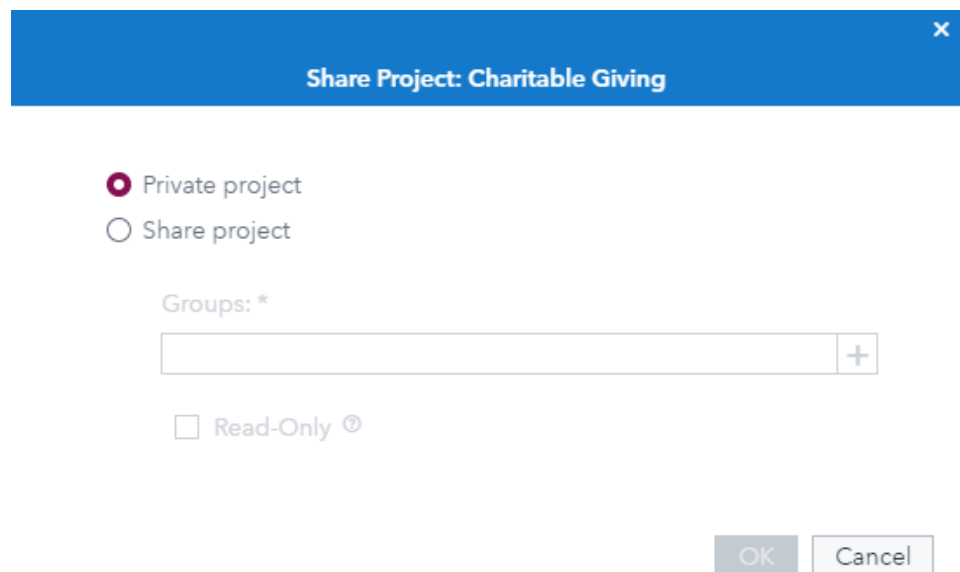
using Model Studio do not retain the same settings for users in SAS Drive. For more information about the authorization service, see [SAS Viya Administration: General Authorization](#). For more information about SAS Drive, see [SAS Drive: Documentation](#).

To share a project:

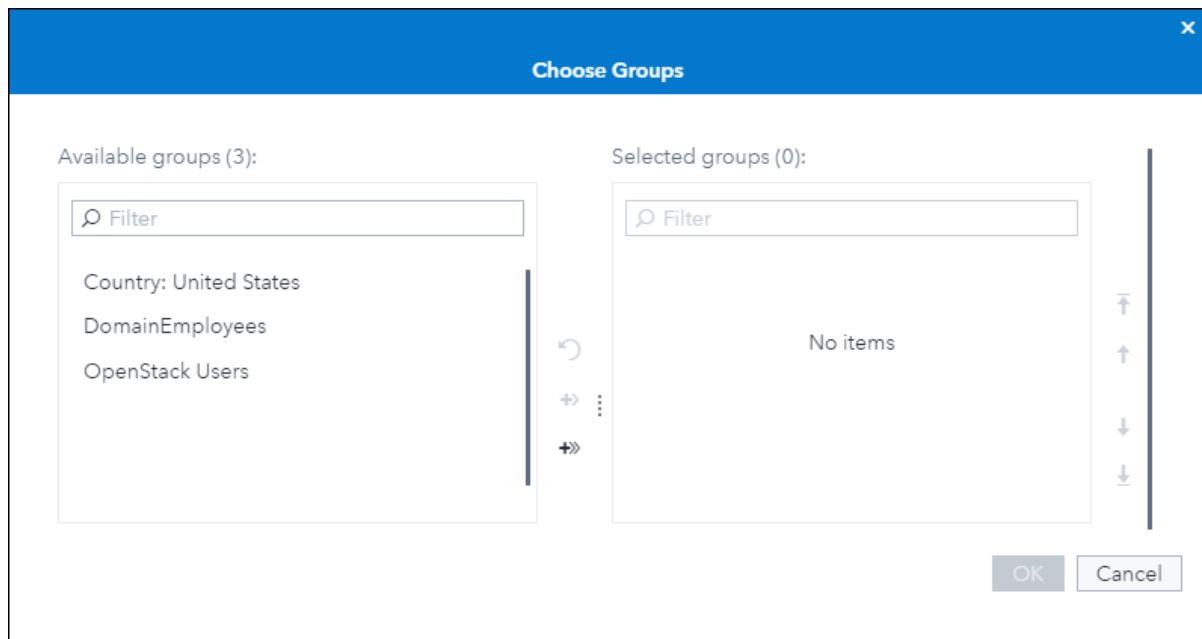
- 1 Select the desired project by clicking the check box in the project tile, and then click  next to the Project page Toolbox.



- 2 Select **Share**.
- 3 The Share Project window appears.




- 4 Select **Share project**.
- 5 Configure the groups by clicking **+**. Use the Choose Groups window to select which groups you want to share access with.



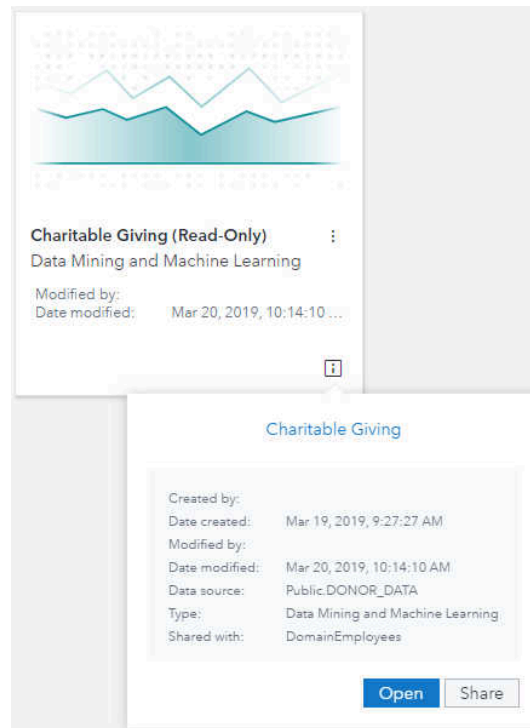
Once groups have been configured, click **OK**.

- 6 By default, group members can modify the shared project. To disable this feature, select **Read-Only**.

Note: The following features apply to shared projects:

- Project sharing can be changed only by the project owner or an administrator.
- Only the administrators and the owner of a shared project can delete that project.
- If a project is not shared in **Read-Only** mode, then only one person can have the project open at a time. Shared projects that are currently open are indicated with a  icon on the Projects page.
- If a project is shared in **Read-Only** mode, nobody can make changes to the project, including the project owner.

- 7 Once the configurations are set on the Share Project window, click **OK** to share. You can see that your project has been shared on the project tile.



You can also remove sharing of a project. To do this, repeat steps 1 through 3 above, but in the Share Project window select **Private project** and click **OK**. This removes shared access to the project.

Importing and Exporting a Project

The source data for a project must already be loaded to the CAS server before you import the project.

To export a project:

- 1 On the Projects page, select the project that you want to export by clicking the check box in the project tile.
- 2 Click **:** in the upper right corner of the window, and select **Export**.

The project files will immediately begin to download. SAS Visual Data Mining and Machine Learning projects are stored as JSON files.

Only SAS Administrators can import SAS Visual Data Mining and Machine Learning projects. To import a project:


- 1 Click **:** in the upper right corner of the window, and select **Import**. If you also have SAS Visual Forecasting or SAS Visual Text Analytics installed, select **Import** ⇒ **Data Mining and Machine Learning**.
- 2 In the Import Data Mining Project window, specify the location of the project and an associated data set. When you import a project, you must specify the ZIP file that was saved when you exported the original project.

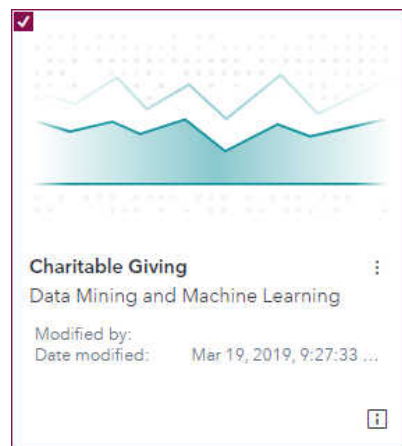
3 Click **Import**.

Note: These steps apply only to importing projects that were created in the same version of Model Studio. For information about how to import projects from previous versions of Model Studio, see [Promotions within SAS Viya](#).

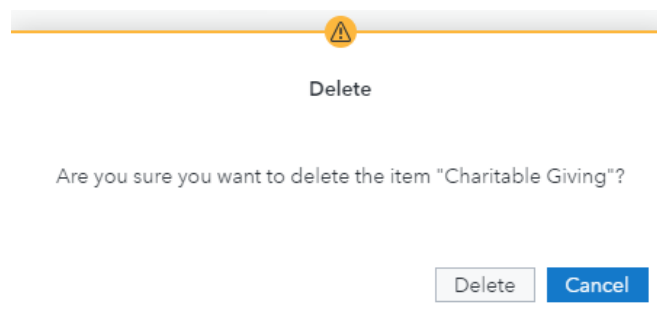
Deleting a Project

To delete a project:

- 1 Select the desired project by clicking the check box in the project tile, and then click  next to the Project page Toolbox.




- 2 Select **Delete**.
- 3 The Delete window appears, asking for confirmation of deletion.

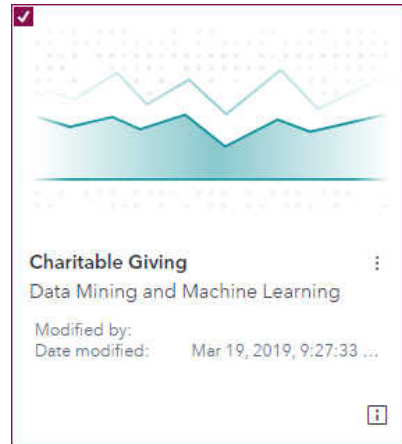


- 4 Click **Delete**.

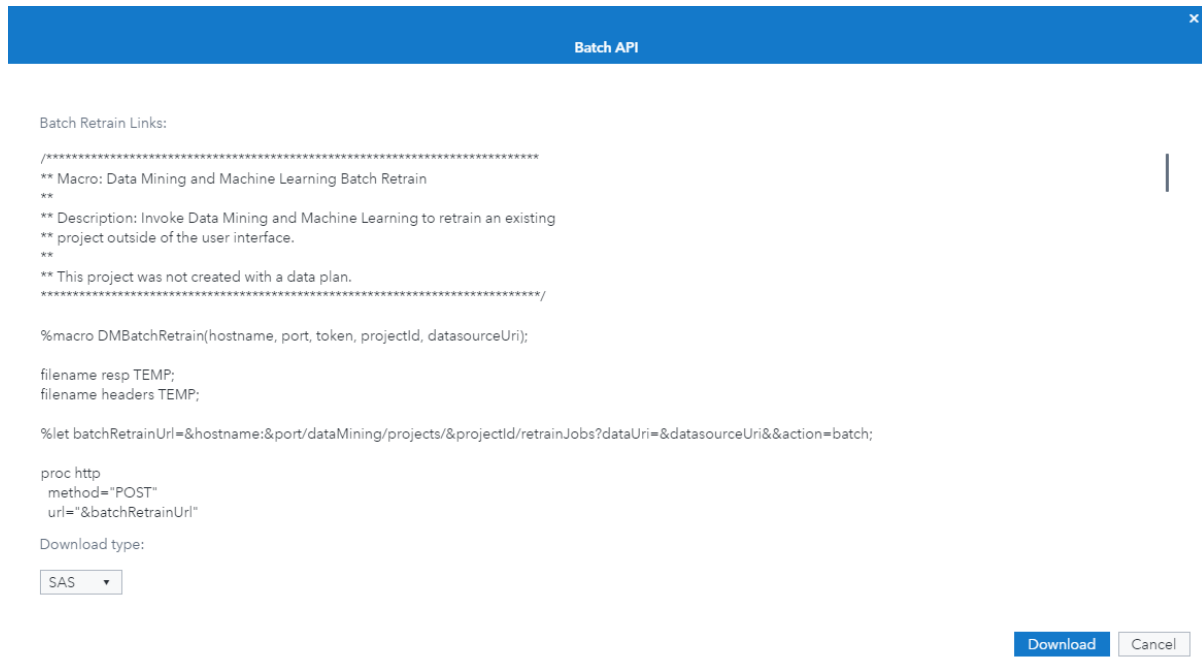
Downloading Project Batch API Code

To download project batch API code:

- 1 Select the desired project by clicking the check box in the project tile, and then click  next to the Project page Toolbox.



- 2 Select **Batch API**.
- 3 The Batch API window appears, which displays the batch API code for the project. Batch API code can be given in the following types:
 - **SAS** — Downloads code to run in a SAS programming environment.
 - **Python** — Downloads code to run in a Python programming environment.
 - **REST** — Downloads a text file with example REST calls that you can use in an application.



- 4 Select the download type for the batch code, and click **Download**. The code will begin downloading immediately.

To run the batch code, supply the host name of the SAS Visual Data Mining and Machine Learning server and the user name and password.

SAS

Update the host, username, and password macro variables at the end of the code.

```

%let protocol = http;
%let host = test.example.com;
%let port = 80;
%let username = my_username;
%let password = my_password;
%let projectId = a0548b2f-a669-4a10-a4dd-052c671c0c00;
%let datasourceUri = /dataTables/dataSources/cas-fs-cas-shared-default~fs-Public/
tables/myTable
    
```

Python

Specify these parameters on the command line. For example, if `download.py` is the file name of the Python batch code, issue the command:

```

$ python download.py --host test.example.com
--username my_username --password my_password
    
```

If you have overrides in the project, they might generate conflicts when they are submitted by the batch code. You can update the batch code to automatically resolve these conflicts by adding `autoresolve="true"` to the code.

SAS

Find the PROC HTTP procedure that includes the following IN option with `firstTransaction` and `lastTransaction`.

```

in="{ "firstTransaction": "@first", "lastTransaction": "@last" }"
    
```

Add the `autoResolve` setting, as follows:

```

in="{ "firstTransaction": "@first", "lastTransaction": "@last",
    
```

```
, "autoResolve": true}
```

Python

Find the `resubmit_overrides` function:

```
def resubmit_overrides(env):
```

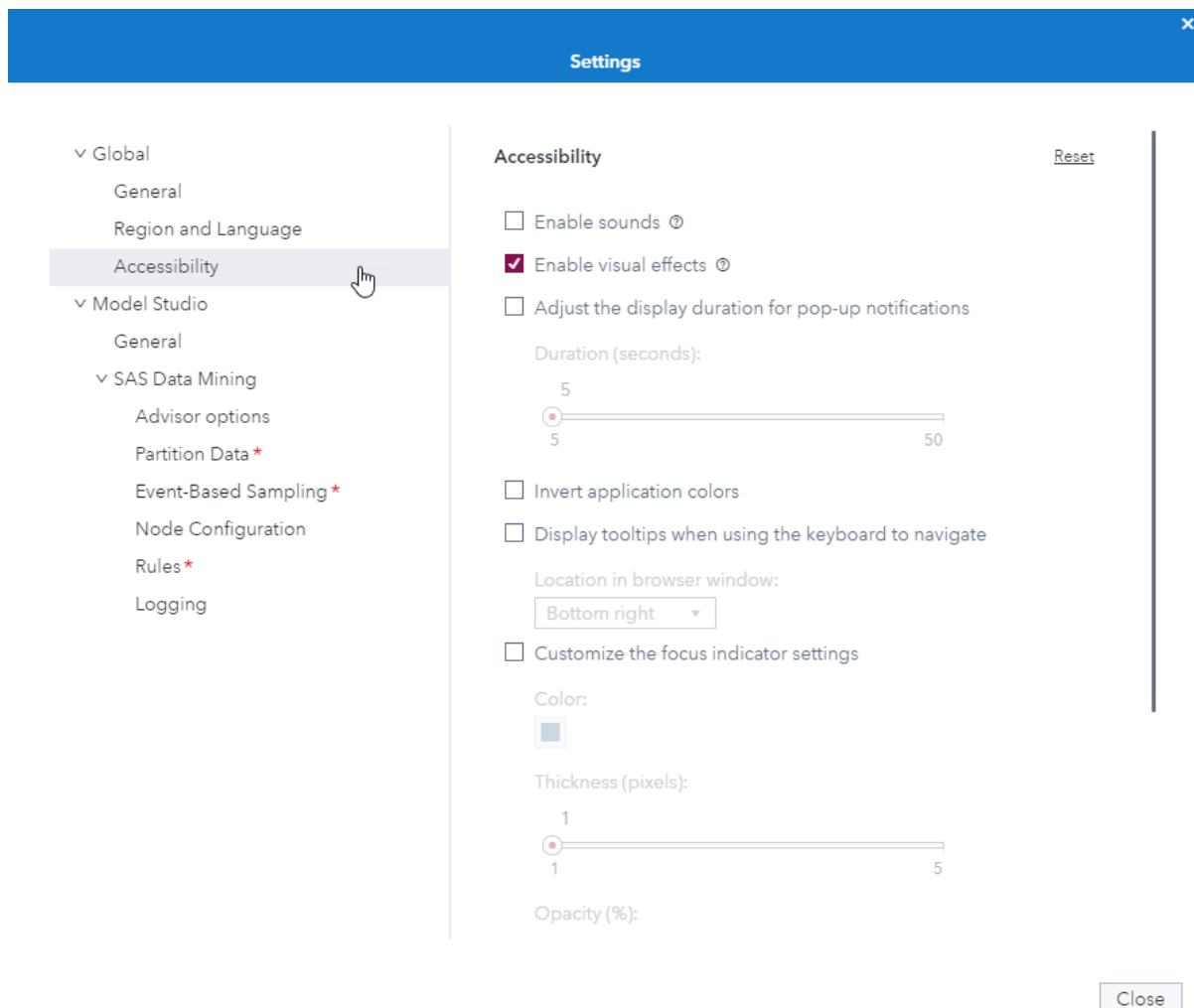
Update the body statement with `autoResolve` assignment.

```
body = '{ "firstTransaction": "@first", "lastTransaction": "@last",  
  "autoResolve": true }'
```

Specifying Global Settings

When setting up your Model Studio account, you might want to modify global settings for your account instance. To edit global settings:

- 1 In the upper right corner of the window, click the user name button, and select **Settings**.
- 2 The Settings window appears, enabling you to alter global settings.



- 3 For changing global settings, the following options are available:
- **General** — These settings enable you to set your interface theme, reset messages, and choose a profile picture. To return all settings to their default values, click **Reset**.
 - **Region and Language** — These settings enable you to set locales for your browser and Java Runtime Environment. To return all settings to their default values, click **Reset**.
 - **Accessibility** — These settings enable you to specify sounds and visual effects for your interface. You can also adjust the display duration for pop-up notifications, invert colorings, display tooltips, and customize the focus indicator. To return all settings to their default values, click **Reset**.
- 4 For changing Model Studio project settings at a global level, the following options are available for SAS Data Mining users:
- **General** — Select **Reset Column Preferences** to reset table column preferences to their default values.
 - **Advisor options** — These settings enable you to specify settings such as the maximum number of class level values and the maximum percentage of missing values. Another option in this group enables you to control the interval cutoff value for numeric variables. If the number of unique levels exceeds the specified cutoff value, the variable is assigned the measurement level interval. To return all settings to their default values, click **Reset**.
 - **Partition Data** — These settings enable you to partition the data set into subsets used for training, validation, and test. You can also specify the partition method. These proportions apply to new projects and can be changed at any time. To change the partition proportions in existing projects, see [Specifying Project Settings on page 78](#). By default, the data is partitioned as follows:

Training	60%
Validation	30%
Test	10%

To return all settings to their default values, click **Reset**.


- **Event-based Sampling** — These settings enable you to specify event-based sampling for the model. All of the observations that have the target event are included, and the observations with non-events are sampled to match the percentages specified for **Event** and **Non-event**. By default, event-based sampling is disabled. If enabled, the **Event** and **Non-Event** percentages are both 50% by default. To return all settings to their default values, click **Reset**.
- **Node Configuration** — These settings enable you to specify configuration code in Python. This Python code is automatically prepended to every **Open Source Code** that you include in your pipeline when the **Language** property is set to **Python**. To return all settings to their default values, click **Reset**.

Note: The Python code can be a maximum of 4000 characters.

- **Rules** — These settings enable you to specify the rules used for comparing pipeline models, the default binary classification cutoff value, and the number of cutoff values to use for the ROC assessment charts. For more information, see [Overview of Model Comparison](#) in the Model Studio reference documentation. To return all settings to their default values, click **Reset**.
 - **Logging** — These settings enable you to specify debug reporting, including options to resolve macro variables, add timings and headers, and retain temporary tables. To return all settings to their default values, click **Reset**.
- 5 Once the settings are appropriately configured, click **Close**. Settings are automatically saved.

Specifying Project Settings

For certain Model Studio projects, you might need to modify project settings to set up models properly. To edit project settings:

- 1 Open the project, and then click  in the upper right corner of the window, under the user name button, and click **Project settings**.

x

Project Settings

Partition Data
 Event-Based Sampling
 Node Configuration
 Rules
 Output Library
 Logging

Partition Data

☒ Create partition variable

Note: These settings are active only when a partition variable is not set within the data. Using a data source with a pre-defined partition variable or manually selecting a partition variable will override these settings.

Method:

Stratify ▼

Training:

60.00%

Validation:

30.00%

Test:

10.00%

Save

Cancel

- 2 The Edit Project Settings window contains several properties that might need to be altered for your projects. The following options are available:

- **Partition Data** — These settings enable you to partition the data set into subsets used for training, validation, and test. You can also specify the partition method. By default, the data is partitioned as follows:

Training	60%
Validation	30%
Test	10%

The **Partition Data** proportions can be edited after creating a project, but only before the **Data** node is run. After the **Data** node is run, you can view the partition proportions, but you cannot edit them until you reload the data source. For more information, see [Replace the Data Source on page 86](#).

- **Event-Based Sampling** — These settings enable you to specify event-based sampling for the model. All of the observations that have the target event are included, and the observations with non-events are sampled to

match the percentages specified for **Event** and **Non-event**. By default, event-based sampling is disabled. If enabled, the **Event** and **Non-Event** percentages are both 50% by default.

- **Node Configuration** — These settings enable you to specify configuration code in Python. This Python code is automatically prepended to every **Open Source Code** that you include in your pipeline when the **Language** property is set to **Python**.

Note: The Python code can be a maximum of 4000 characters.

- **Rules** — These settings enable you to specify the rules used for comparing pipeline models, the default binary classification cutoff, and the number of cutoff values to use for the ROC assessment charts. For more information, see [Overview of Model Comparison](#) in the Model Studio reference documentation.
- **Output Library** — These settings enable you to specify the library for output from the model.
- **Logging** — These settings enable you to specify debug reporting, including options to resolve macro variables, add timings and headers, and retain temporary tables.

- 3 Once the settings are appropriately configured, click **Save**.

Note: The settings configured at the project level override any global settings that you configured for your Model Studio instance.

Importing a Project from SAS Visual Analytics

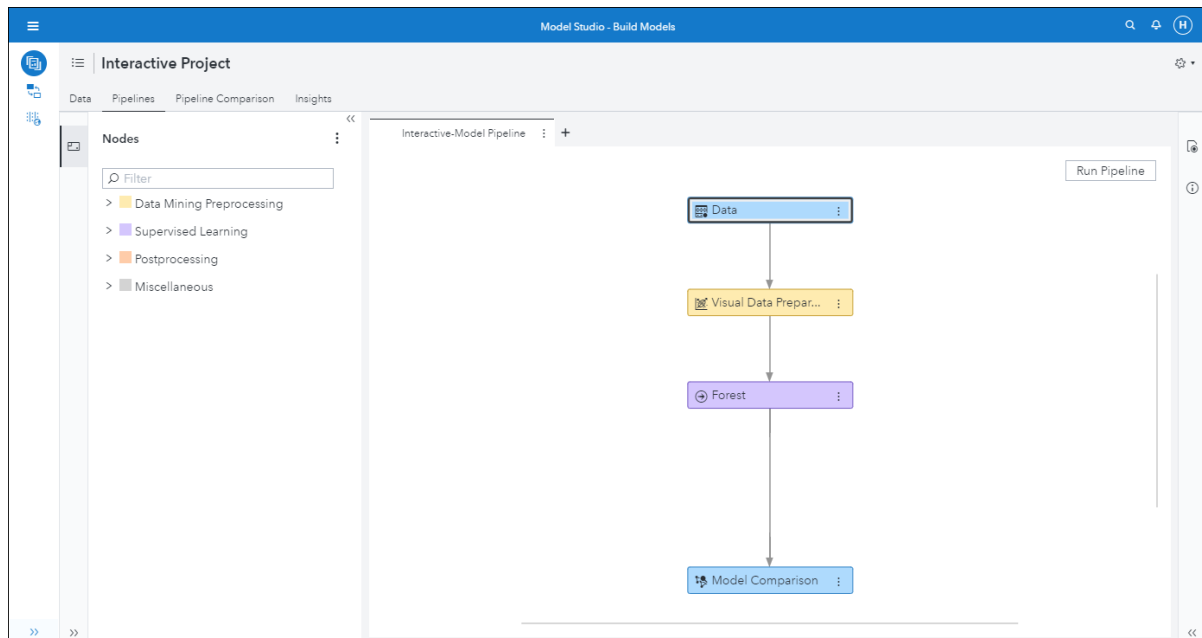
Model Studio users can use their SAS Visual Analytics reports to create projects. The following SAS Visual Analytics objects can be used to generate pipelines in Model Studio:

- **Bayesian Network**
- **Decision Tree**
- **Forest**
- **Generalized Linear Model**
- **Gradient Boosting**
- **Linear Regression**
- **Logistic Regression**
- **Model Comparison**
- **Neural Network**

■ Support Vector Machine

The SAS Visual Analytics objects listed above correspond directly to nodes in Model Studio. To export an object:

- 1 In SAS Visual Analytics, click the **Create Pipeline** button in the upper left corner of the object canvas, and select either **Add to new project** or **Add to existing project**.
- 2 If **Add to new project** is selected, the Model Studio interface opens, and a project is created in Model Studio. The name matches the name of the saved SAS Visual Analytics report. If the report in SAS Visual Analytics has no name, the project is named *Interactive Project* in Model Studio. If **Add to existing project** is selected, the Model Studio interface opens, and you are prompted to select an existing project. Only Model Studio projects where the data node has previously been run and where the target variable name, type, and event level match will be available. A pipeline is then created, and named *Interactive-Model Pipeline*.



This pipeline looks identical to a normal Model Studio pipeline, except for two key differences:

- Model Studio generates a Data Mining Preprocessing node called **Visual Data Preparation**. The **Visual Data Preparation** node runs the score code necessary to perform all the data preparation steps that were performed in SAS Visual Analytics. The properties of this node are not available to edit.
- The SAS Visual Analytics objects are also represented. Here, **Forest** corresponds to the Forest object in SAS Visual Analytics. Supervised learning nodes can be edited in Model Studio. Right-click the supervised learning node, and then select **Enable properties**. If you enable the properties of a node, the model is retrained, which might yield new results. Once properties have been enabled, the model cannot be switched back to use the original SAS Visual Analytics score code. **Model Interpretability** properties do not require retraining the model, so they are automatically enabled and do not affect the original SAS Visual Analytics score code. This pipeline can be run as-is, or nodes can be added to the pipeline to be run for comparison purposes.

Working with Data

<i>Data Management Overview</i>	83
<i>Importing Data</i>	84
<i>User-Defined Formats</i>	85
<i>Replace the Data Source</i>	86
<i>Managing Variable Assignments</i>	87
Assigning Variable Metadata	87
Assigning Variable Metadata Details	87
<i>Managing Global Metadata</i>	91
<i>Integration with SAS Visual Analytics</i>	91
<i>Explore and Visualize Your Data</i>	93

Data Management Overview

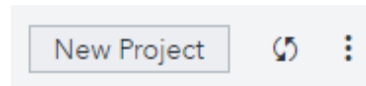
Model Studio provides several options for managing and modifying data. A project can have only one data source. However, some nodes can point to additional data sources. The Data tab enables you to modify variable assignments and manage global metadata. You can also retrain a model with new data, as long as the target variable in the new data set is the same as the original data set.

Note: Typically, the column names of the input data are not translated. However, certain system-generated UI labels and column names are automatically translated when you are using Model Studio in a non-English language. If your input data contains a column name that matches one of these system-generated UI labels or column names, you might encounter unexpected translations of the column names of your input data.

Importing Data

To add a new data set to the repository for use in a new project:

- 1 On the Model Studio Projects page, select **New Project** in the upper right corner.



- 2 The New Project window appears.

New Project

Name: *

Toy Company

Type: *

Data Mining and Machine Learning

Template:

Blank template

Data: *

Browse

Description:

Advanced

Save

Cancel

- 3 Click **Browse** to open the Choose Data window.
- 4 The Choose Data window appears. Select the **Import** tab.
- 5 Drag the desired local data set directly to the window. Model Studio parses the data set and pre-populates the window with data set configurations.

- 6 After configuring the data set import properties, click **Import Item**. After successful import, the following message appears:

✓ The table was successfully imported on Mar 19, 2019 03:17 PM and is ready for use.

Click **OK**.

To finish creating a project, follow the steps outlined in [Creating a New Project on page 68](#).

Note: Data set names and variable names can have a maximum length of 32 characters.

You cannot create a project if the data set name contains any of the following characters: / \ * ? " < > | : - . &

User-Defined Formats

Your data sets might include formats that are not natively supported by SAS Visual Data Mining and Machine Learning. To enable SAS Visual Data Mining and Machine Learning to recognize these formats:

- 1 Upload your format to a CAS format library.

- 2 Move your format to the compute server machine. If the formats are Windows formats and the machine is UNIX, then you need to use PROC CPORT and PROC CIMPORT to move the format. For more information, see the [PROC CPORT](#) and [PROC CIMPORT](#) documentation in *Base SAS 9.4 Procedures Guide*.
- 3 Add the format to the compute server session search path. For example, you can add the following code to the `autoexec_usermods.sas` autoexecutable SAS file in `/opt/sas/viya/config/etc/compssrv/default`:

```
libname format '</home/filepath/casuser/>';
options fmtsearch=(format.emfmt);
```



See [Managing User-Defined Formats in SAS Viya](#) in the SAS Viya Administration documentation for more information.

Replace the Data Source

A project can have only one data source. You might want to replace the data source in your project to do one of the following:

- Retrain a model with new data.
- Change the way that the data is partitioned after a pipeline is run.
- Reload the input table with additional records.
- Reload the input table with additional variables.
- Load the input table with a different name.

To replace the data source of a project:

- 1 Open a project. On the Data tab, click  to open the Data Sources pane.
- 2 In the upper left corner of the Data Source pane, click .
- 3 The Choose Data window appears. If the desired data set has already been added to the Model Studio repository, select it from the list on the **Available** tab. If the desired data set is local to your environment, select the **Import** tab, and follow the instructions in [Importing Data on page 84](#).
- 4 Click **OK**. The Data tab now displays details about the new data set.

Note: To retrain a model with a new data set, the new data set must use the same target variable as the original data set. For more information, see [Managing Global Metadata on page 91](#).

Managing Variable Assignments

Assigning Variable Metadata

To specify variable properties:

- 1 On the **Data** tab, select the desired variables.
- 2 The right pane enables you to specify several properties of the variables, including the following:
 - **Role**
 - **Level**
 - **Order**
 - **Transform**
 - **Impute**
 - **Lower Limit**
 - **Upper Limit**

For the **Transform**, **Impute**, **Lower Limit**, and **Upper Limit** properties, altering these values in the **Data** tab does not directly modify the variable. Instead, this sets metadata values for these properties. The Data Mining Preprocessing nodes that use metadata values (**Transformations**, **Impute**, **Filter**, and **Replacement**) might use these parameters if the corresponding action is requested.

Assigning Variable Metadata Details

- The following options are available when specifying variable roles:
 - **Assessment** — Specifies that the variable be used for decision processing. The Assessment role is currently not used in Model Studio.
 - **Classification** — Specifies that the variable be used for model classification for a class target. For example, if the variable BAD is set as the target variable, the I_BAD variable has the 0 or 1 prediction based on the predicted probabilities and the cutoff used. The classification cutoff is applied only to binary targets.
 - **Filter** — Specifies that the variable be used for filtering. For variables with the role of Filter, observations are filtered out when the value = 1 and kept when the value = 0. The Filter role is used in the **Filtering** node and the **Anomaly Detection** node.
 - **ID** — Specifies that the variable is an ID variable.

- ❑ **Input** — Specifies that the variable be used as an input variable in your pipeline.
- ❑ **Key** — Specifies that the variable is a unique identifier for all observations. The Key role is used by the **Text Mining** node and is used in the generation of the observation-based Model Interpretability reports.
- ❑ **Offset** — Specifies that the variable is a numeric variable that is used by the **GLM**, **Gradient Boosting**, and **Logistic Regression** nodes. An offset variable is typically used for a covariate with a known slope. The variable that is specified is not estimated, but instead is added directly to the model.
- ❑ **Partition** — Specifies that the variable be used for partitioning your data set.
- ❑ **Prediction** — Specifies that the variable is used during model assessment. This variable is the prediction for an interval target or the posterior probabilities for a class target.
- ❑ **Rejected** — Specifies that the variable be excluded from all analysis in your pipeline.
- ❑ **Residual** — Specifies that the variable is an error residual. This role is used only for informational purposes.
- ❑ **Segment** — Specifies that the variable is a segment variable. Segment variables are created by the **Clustering** node for cluster IDs created. The segment variable is also used in the **Segment Profile** node.
- ❑ **Target** — Specifies that the variable is the target variable.
- ❑ **Text** — Specifies that the variable is a text variable. The Text role is used by the **Text Mining** node.
- ❑ **Time ID** — Specifies that the variable is a time variable. This role is used only for informational purposes.

Nominal variables that are assigned the role **Target** also have the option **Specify the Target Event Level**. This enables you to choose which level to assign as the event level.

The variable role **Partition** can be assigned only to variables with three or fewer unique levels. You can select **Map Partition Levels** to choose which levels to assign to the Training, Validation, and Test partitions.

Note: Frequency variables are not supported.

Note: Interval variables that have a DATE, DATETIME, or TIME format can be assigned only the variable roles **ID** and **Rejected**.

- The following options are available when selecting variable levels:
 - ❑ **Binary**
 - ❑ **Interval**
 - ❑ **Nominal**
 - ❑ **Ordinal**

Note: All nodes analyze **Ordinal** variables as **Nominal** variables.

- The following options are available when selecting variable order:

- ☐ **Ascending**
- ☐ **Default**
- ☐ **Descending**
- ☐ **Formatted Ascending**
- ☐ **Formatted Descending**
- The following options are available when selecting variable transformations:
 - ☐ For class variables, the options are as follows:
 - **(none)**
 - **Bin rare nominal levels**
 - **Default**
 - **Level count encoding**
 - **Level encoding**
 - **Level proportion encoding**
 - **Target encoding**
 - **WOE encoding**

The **Bin rare nominal levels** option is available only for nominal level variables.

- ☐ For interval variables, the options are as follows:
 - **(none)**
 - **Best**
 - **Bucket binning**
 - **Centering**
 - **Default**
 - **Exponential**
 - **Inverse**
 - **Inverse square**
 - **Inverse square root**
 - **Log**
 - **Log10**
 - **Quantile binning**
 - **Range standardization**
 - **Square**
 - **Square root**
 - **Standardization**
 - **Tree-based binning**

The **Best** transform option performs several transformations and uses the transformation that by default, minimizes the skewness of the variable distribution.

Note: **Log**, **Log 10**, **Square root**, and **Inverse square root** add an offset to variables to ensure positive values. **Inverse** and **Inverse square** add an offset to variables to ensure nonzero values. This prevents creating missing values during the transformation when input variable values are zero.

- The following options are available when you are selecting how to impute missing variable values:

- For class variables, the options are as follows:

- (none)
- Cluster count
- Count
- Custom constant value
- Default
- Default constant value
- Distribution

Cluster count, **Count**, and **Distribution** are not available for unary variables.

- For interval variables, the options are as follows:

- (none)
- Cluster mean
- Custom constant value
- Default
- Default constant value
- Distribution
- Maximum
- Mean
- Median
- Midrange
- Minimum
- Trimmed maximum
- Trimmed mean
- Trimmed midrange
- Trimmed minimum
- Winsorized maximum
- Winsorized mean
- Winsorized midrange
- Winsorized minimum


You can specify a **Lower limit** and an **Upper limit** to use for the **Metadata limits** method in the **Filtering** node or the **Replacement** node.

Managing Global Metadata


In Model Studio, *metadata* is defined as the set of variable roles, measurement levels, and other configurations that apply to your data set.

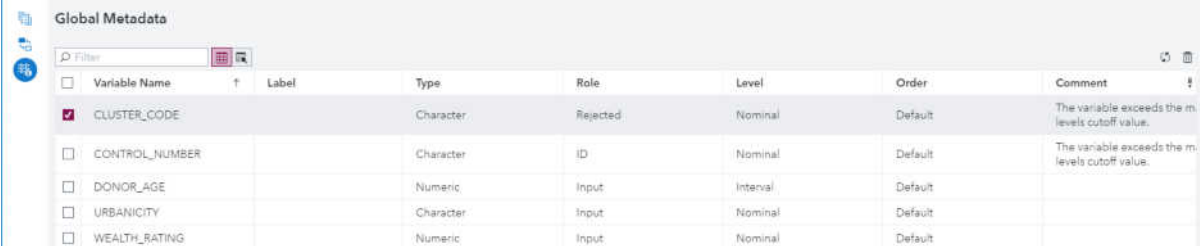
When creating multiple projects using similar data sets (or when using a single data set), you might find it useful to store the metadata configurations for usage across projects. Model Studio enables you to do this by collecting the variables in a repository called Global Metadata. By storing your metadata configurations as global metadata, the configurations apply to new data sets that contain variables with the same name.

To save a variable as global metadata:

- 1 On the **Data** tab, select the desired variables.
- 2 In the right pane, set the desired variable metadata assignments.
- 3 Click  in the upper right corner of the right pane.
- 4 A window appears, confirming that the operation was successful.


To examine and manage the variables designated as global metadata:

- 1 Click  in the upper left corner of the window.
- 2 The Global Metadata window appears.



Variable Name	Label	Type	Role	Level	Order	Comment
<input checked="" type="checkbox"/> CLUSTER_CODE		Character	Rejected	Nominal	Default	The variable exceeds the m. levels cutoff value.
<input type="checkbox"/> CONTROL_NUMBER		Character	ID	Nominal	Default	The variable exceeds the m. levels cutoff value.
<input type="checkbox"/> DONOR_AGE		Numeric	Input	Interval	Default	
<input type="checkbox"/> URBANICITY		Character	Input	Nominal	Default	
<input type="checkbox"/> WEALTH_RATING		Numeric	Input	Nominal	Default	

This window displays a table that contains all variables specified as global metadata, as well as their metadata assignments.

- 3 To remove a variable from the global metadata repository, select the desired variable and click  in the upper right corner of the window.

Integration with SAS Visual Analytics

SAS Visual Analytics enables you to transfer certain analytical models from SAS Visual Analytics to Model Studio. To move a model from SAS Visual Analytics to Model Studio, click the **Create pipeline** button.

This action creates a new project in Model Studio that contains the following elements:

- the active data set
- score code to apply all data processing, filtering, and transformations
- score code to run the model that was exported

The properties of the modeling nodes can be edited, and subsequently models can be retrained in Model Studio. Right-click the modeling node, and then select **Enable properties** to edit the node. Model interpretability properties are always automatically enabled. You can add and delete nodes in this pipeline as in any other Model Studio pipeline. You can use the Model Studio model comparison and pipeline comparison tools to evaluate your transferred models against any new models.

At this time, the supported models are Bayesian Network, Decision Tree, Forest, Generalized Linear Model, Gradient Boosting, Linear Regression, Logistic Regression, Neural Network, and Support Vector Machine. There also exist exceptions within these models:

- You cannot copy a Decision Tree with a binned measure response.
- You cannot copy a Generalized Linear Model, Linear Regression, or Logistic Regression that uses a frequency, weight, offset, or group by variable.
- You cannot copy a Logistic Regression with a non-binary response variable.
- You cannot copy a Neural Network with a weight variable.


There are a few caveats to note when transferring a model from SAS Visual Analytics to Model Studio.

- In order to add a SAS Visual Analytics model to an existing Model Studio project, the target variable name, type, and event level must match. The data node in the existing Model Studio project must have also been previously run.
- Instead of using the variable name that exists in the original data set, SAS Visual Analytics prefers to use the variable label. However, Model Studio prefers to use the variable name as it exists in the original data. Therefore, if the variable names and variable labels in your input data are different, you might experience some unexpected naming issues when a model is transferred. Model Studio displays both the variable name and the variable label in the **Variables table** layout of the Data pane.
- SAS Visual Analytics creates a custom name for target variables. This new variable is indicated with a label in Model Studio.
- When you are exporting from the Model Comparison object, only the champion model is exported.
- Partition variables must be numeric variables that contain only the values 0 for training data, 1 for validation data, and 2 for testing data. The testing data can be omitted. If the partition information from SAS Visual Analytics differs from Model Studio, then the partitioning of the Model Studio project is used.
- Category target variables cannot contain any special characters, including a comma, semicolon, open parenthesis, or close parenthesis. Special characters in the target variable of a Model Studio pipeline cause model creation to fail.
- You cannot transfer a model from Model Studio to SAS Visual Analytics. However, you can copy the input data to SAS Visual Analytics for exploration and visualization.


- Certain actions that create a data item in SAS Visual Analytics are performed in the **Visual Data Preparation** node in Model Studio. For example, when you derive a cluster ID in SAS Visual Analytics, a data item is created in the **Data** pane. If you specify this created data item in a SAS Visual Analytics model that is transferred to Model Studio, it does not appear in the **Data** pane of your project. Instead, it is re-created when the **Visual Data Preparation** node runs.
- If you enable the properties of a transferred SAS Visual Analytics modeling node in Model Studio, the model is retrained, which might yield new results. Once properties have been enabled, the model cannot be switched back to use the original SAS Visual Analytics score code. **Model Interpretability** properties do not require retraining the model. Therefore, they are automatically enabled and do not affect the original SAS Visual Analytics score code.
- You cannot save your pipeline as a template to the exchange.
- If you transfer a Gradient Boosting, Decision Tree, or Forest model, the value for the **Maximum levels** property that is mapped to Model Studio is one fewer than the value that you specified in SAS Visual Analytics.

Explore and Visualize Your Data

The **Output Data** tab is located in the upper left corner of the **Results** window. It enables you to generate output data. A row is created for each observation, and all original and created columns are displayed. The created columns can contain imputation results, model prediction results, or any other information created by the node.

In order to generate output data, click the **Output Data** tab, and then click **View Output Data** in the middle of the screen. When the output data is successfully generated, the table containing that data appears. You can save the output data for later use by clicking  in the upper left corner of the **Data sources** tab.

Note: Table names cannot exceed 247 bytes.

You can also visualize your output data by clicking  in the upper left corner of the **Data sources** tab. The Explore and Visualize Output Data window appears, and you are prompted to select a CAS library that you want to save your output table in.

Note: To use the **Explore and Visualize** functionality in Model Studio, you must be able to create and edit reports in SAS Visual Analytics.

When you have selected a CAS library, click **Explore and Visualize** in the lower right corner of the window. This redirects you to SAS Visual Analytics, where you can use a variety of tools to model your data. For information about using SAS Visual Analytics, see [SAS Visual Analytics: Getting Started with Reports](#).

Working with Templates


<i>Overview of Templates</i>	95
<i>Creating a New Template from a Pipeline</i>	95
<i>Creating a New Template in The Exchange</i>	96
<i>Modifying an Existing Template</i>	96
<i>Available Templates</i>	97

Overview of Templates

Model Studio supports templates as a method for creating statistical models quickly. A *template* is a special type of pipeline that is pre-populated with configurations that can be used to create a model. A template might consist of multiple nodes or a single node. Model Studio includes a set of templates that represent frequent use cases, but you can also create models themselves and save them as templates in the toolkit.

Creating a New Template from a Pipeline



To create a template from a pipeline:

- 1 Click  next to the pipeline tab in the upper left corner of the canvas.
- 2 Select **Save to The Exchange**.
- 3 In the Save Pipeline to The Exchange window, enter a **Name** and **Description** for the new template.
- 4 Click **Save**.

You can also create templates from singular nodes. To create a template from a node:


- 1 Right-click on the desired node. Select **Save As**. The Save Node to The Exchange window appears.
- 2 In the Save Node to The Exchange window, enter a **Name** and **Description** for the new template.
- 3 Click **Save**.

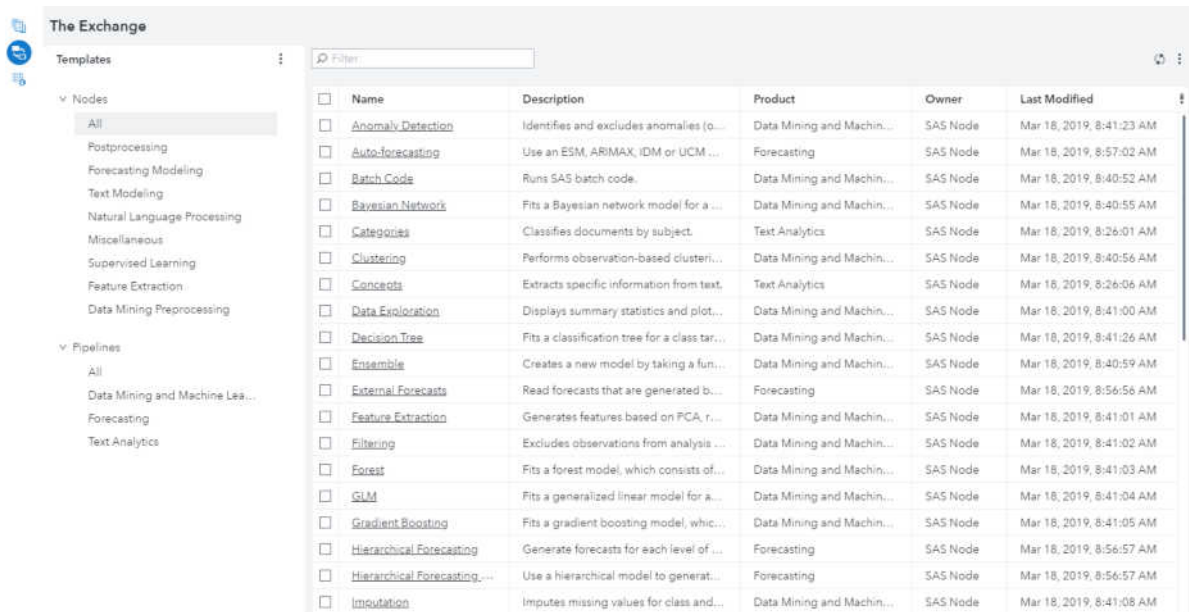
Creating a New Template in The Exchange

- 1 Click  in the upper left corner of the screen. The Exchange page opens. This page enables you to examine all available templates. The Exchange stores node and pipeline templates for SAS Visual Data Mining and Machine Learning, SAS Visual Text Analytics, and SAS Visual Forecasting applications.
- 2 To create a new template, select the existing template most similar to your desired template. You will duplicate and modify this template.
- 3 Click  in the upper right corner of the screen and select **Duplicate**
- 4 For node templates, the Duplicate Node window appears. For pipeline templates, the Duplicate Pipeline window appears. Enter a name and a description for the new template.
- 5 Click **Save**. Your new template appears in the list of templates.

Modifying an Existing Template

If you have sufficient permissions, you can modify existing templates. To modify a template:

- 1 Click  in the upper left corner of the screen.
- 2 The Exchange page opens. This page enables you to examine all available templates. The Exchange stores node and pipeline templates for SAS Visual Data Mining and Machine Learning, SAS Visual Text Analytics, and SAS Visual Forecasting applications.



Name	Description	Product	Owner	Last Modified
Anomaly Detection	Identifies and excludes anomalies (o...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:23 AM
Auto-forecasting	Use an ESM, ARIMAX, IDM or UCM ...	Forecasting	SAS Node	Mar 18, 2019, 8:57:02 AM
Batch Code	Runs SAS batch code.	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:40:52 AM
Bayesian Network	Fits a Bayesian network model for a ...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:40:55 AM
Categories	Classifies documents by subject.	Text Analytics	SAS Node	Mar 18, 2019, 8:26:01 AM
Clustering	Performs observation-based clusteri...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:40:56 AM
Concepts	Extracts specific information from text.	Text Analytics	SAS Node	Mar 18, 2019, 8:26:06 AM
Data Exploration	Displays summary statistics and plot...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:00 AM
Decision Tree	Fits a classification tree for a class tar...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:26 AM
Ensemble	Creates a new model by taking a fun...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:40:59 AM
External Forecasts	Read forecasts that are generated b...	Forecasting	SAS Node	Mar 18, 2019, 8:56:56 AM
Feature Extraction	Generates features based on PCA, r...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:01 AM
Filtering	Excludes observations from analysis ...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:02 AM
Forest	Fits a forest model, which consists of...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:03 AM
GLM	Fits a generalized linear model for a...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:04 AM
Gradient Boosting	Fits a gradient boosting model, whic...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:05 AM
Hierarchical Forecasting	Generate forecasts for each level of ...	Forecasting	SAS Node	Mar 18, 2019, 8:56:57 AM
Hierarchical Forecasting, ...	Use a hierarchical model to generat...	Forecasting	SAS Node	Mar 18, 2019, 8:56:57 AM
Imputation	Imputes missing values for class and...	Data Mining and Machin...	SAS Node	Mar 18, 2019, 8:41:08 AM

- 3 To access a particular template, click the template name. This opens the Node Template or Pipeline Template window. If you do not have Edit privileges for a given template, you will see **(Read-Only)** displayed in the window.

In the Node Template or Pipeline Template window, you can make changes and configure the nodes in the pipeline. Changes are saved automatically to the template.

Note: While you are editing a template, nodes can be re-configured, but no nodes can be added or deleted.

Available Templates

The following Node templates are included with Model Studio:

Node Name	Node Description	Product
Anomaly Detection	Identifies and excludes anomalies (observations) using the support vector data description.	Data Mining and Machine Learning
Auto-forecasting	Use an ESM, ARIMAX, IDM, or UCM model to generate forecasts.	Forecasting
Batch Code	Runs SAS batch code.	Data Mining and Machine Learning

Node Name	Node Description	Product
Bayesian Network	Fits a Bayesian network model for a class target.	Data Mining and Machine Learning
Categories	Classifies documents by subject.	Text Analytics
Clustering	Performs observation-based clustering for segmenting data.	Data Mining and Machine Learning
Concepts	Extracts specific information from text.	Text Analytics
Data Exploration	Displays summary statistics and plots for variables in your data table.	Data Mining and Machine Learning
Decision Tree	Fits a classification tree for a class target or a regression tree for an interval target.	Data Mining and Machine Learning
Ensemble	Creates a new model by taking a function of posterior probabilities (for class targets) or the predicted values (for interval targets) from multiple models.	Data Mining and Machine Learning
External Forecasts	Reads forecasts that are generated by an external source.	Forecasting
Feature Extraction	Generates features based on PCA, robust PCA, SVD, or autoencoders to use as inputs. Note that PCA, SVD, and RPCA use interval inputs only.	Data Mining and Machine Learning
Feature Machine	Generates features that address one or more identified transformation policies.	Data Mining and Machine Learning
Filtering	Excludes observations from analysis based on specified criteria.	Data Mining and Machine Learning
Forest	Fits a forest model, which consists of multiple decision trees based on different samples of the data and different subsets of inputs.	Data Mining and Machine Learning
GLM	Fits a generalized linear model for an interval target with a specified target distribution and link function.	Data Mining and Machine Learning

Node Name	Node Description	Product
Gradient Boosting	Fits a gradient boosting model, which builds a sequential series of decision trees.	Data Mining and Machine Learning
Hierarchical Forecasting	Generates forecasts for each level of the specified hierarchy.	Forecasting
Hierarchical Forecasting (Pluggable)	Generates forecasts using hierarchical forecasting model.	Forecasting
Imputation	Imputes missing values for class and interval inputs using the specified methods.	Data Mining and Machine Learning
Linear Regression	Fits an ordinary least squares regression model for an interval target.	Data Mining and Machine Learning
Logistic Regression	Fits a logistic regression model for a binary or nominal target.	Data Mining and Machine Learning
Manage Variables	Modifies the metadata of variables.	Data Mining and Machine Learning
Model Composer	Automatically tunes hyperparameters for multiple model types concurrently with optimal allocations, and then selects the top model.	Data Mining and Machine Learning
Multistage Model	Generates forecasts using a multistage forecasting model.	Forecasting
Naive Model	Generates forecasts using naive model.	Forecasting
Neural Network	Fits a fully connected neural network model.	Data Mining and Machine Learning
Non-seasonal Model	Generates forecasts using a non-seasonal ESM, ARIMAX, or UCM model.	Forecasting
Open Source Code	Runs Python or R code.	Data Mining and Machine Learning
Panel Series Neural Network	Generates forecast using fully connected neural network model.	Forecasting

Node Name	Node Description	Product
Quantile Regression	Fits a quantile regression model for an interval target.	Data Mining and Machine Learning
Regression for Time Series	Uses a regression model to generate forecasts	Forecasting
Replacement	Replaces data values such as outliers and unknown class levels with specified values.	Data Mining and Machine Learning
Retired Series	Generates forecasts for retired series using a specified value.	Forecasting
SAS Code	Runs SAS code.	Data Mining and Machine Learning
Save Data	Saves data exported by a node in a pipeline to a CAS library.	Data Mining and Machine Learning
Score Code Import	Imports SAS score code.	Data Mining and Machine Learning
Score Data	Scores a table using the score code generated by predecessor nodes and saves the scored table to a CAS library.	Data Mining and Machine Learning
Seasonal Model	Generates forecasts using a seasonal ESM, ARIMAX, or UCM model.	Forecasting
Segment Profile	Examines segmented data and enables identification of factors that differentiate the segments from the population.	Data Mining and Machine Learning
Sentiment	Analyzes attitudes expressed in documents.	Text Analytics
Stacked Model (NN + TS) Forecasting	Generates forecasts using stacked model (Neural Network + Time Series).	Forecasting
SVM	Fits a support vector machine via interior-point optimization for a binary target.	Data Mining and Machine Learning
Temporal Aggregation Model	Generates forecasts using a temporal aggregation model.	Forecasting

Node Name	Node Description	Product
Text Mining	Parses and performs topic discovery to prepare text data for modeling.	Data Mining and Machine Learning
Text Parsing	Prepares text for terms analysis.	Text Analytics
Topics	Assigns documents to topics.	Text Analytics
Transformations	Applies numerical or binning transformations to input variables.	Data Mining and Machine Learning
Variable Clustering	Performs variable clustering to reduce the number of inputs.	Data Mining and Machine Learning
Variable Selection	Performs unsupervised and several supervised methods of variable selection to reduce the number of inputs.	Data Mining and Machine Learning

The following Pipeline templates are included with Model Studio:

Pipeline Name	Pipeline Description	Product
Advanced template for class target	Extends the intermediate template for class target with neural network, forest, and gradient boosting models, as well as an ensemble.	Data Mining and Machine Learning
Advanced template for class target with autotuning	Advanced template for class target with autotuned tree, forest, neural network, and gradient boosting models.	Data Mining and Machine Learning
Advanced template for interval target	Extends the intermediate template for interval target with neural network, forest, and gradient boosting models, as well as an ensemble.	Data Mining and Machine Learning
Advanced template for interval target with autotuning	Advanced template for interval target with autotuned tree, forest, neural network, and gradient boosting models.	Data Mining and Machine Learning
Advanced template for risk modeling	Risk modeling pipeline that extends the basic risk modeling template with a Reject Inference node and subsequent Interactive Grouping and Scorecard nodes.	Data Mining and Machine Learning

Pipeline Name	Pipeline Description	Product
Auto-forecasting	Forecasting pipeline with automatic modeling.	Forecasting
Auto-forecasting (Intermittent)	Forecasting pipeline with automatic, intermittent modeling.	Forecasting
Base Forecasting	Forecasting pipeline with no modeling components added by default.	Forecasting
Basic template for class target	A simple linear flow: Data, Imputation, Logistic Regression, Model Comparison.	Data Mining and Machine Learning
Basic template for interval target	A simple linear flow: Data, Imputation, Linear Regression, Model Comparison.	Data Mining and Machine Learning
Basic template for risk modeling	Risk modeling pipeline that contains an Interactive Grouping node followed by a Scorecard node.	Data Mining and Machine Learning
Blank Template	A Data Mining pipeline that contains only a data node.	Data Mining and Machine Learning
Demand Classification	Forecasting pipeline with demand classification segmentation.	Forecasting
External Forecasts	Forecasting pipeline with external forecasts.	Forecasting
External Segmentation	Forecasting pipeline with external segmentation.	Forecasting
Feature engineering template	Data mining pipeline that performs feature engineering.	Data Mining and Machine Learning
Hierarchical Forecasting	Forecasting pipeline with hierarchical modeling.	Forecasting
Intermediate template for class target	Extends the basic template with a stepwise logistic regression model and a decision tree.	Data Mining and Machine Learning
Intermediate template for interval target	Extends the basic template with a stepwise linear regression model and a decision tree.	Data Mining and Machine Learning

Pipeline Name	Pipeline Description	Product
Naive (Moving Average) Forecasting	Forecasting pipeline with naive, moving average modeling.	Forecasting
Naive Forecasting	Forecasting pipeline with naive modeling.	Forecasting
Non-seasonal Forecasting	Forecasting pipeline with non-seasonal modeling.	Forecasting
Regression Forecasting	Forecasting pipeline with regression modeling.	Forecasting
Retired Forecasting	Forecasting pipeline with retired modeling.	Forecasting
Seasonal Forecasting	Forecasting pipeline with seasonal modeling.	Forecasting
Text Analytics: Assisted Concept Rule Creation	Use Textual Elements to quickly generate custom concept rules.	Text Analytics
Text Analytics: Data Access	Text Analytics pipeline that contains a single Data node.	Text Analytics
Text Analytics: Generate Concepts, Topics, and Categories	Text Analytics pipeline for model generation with Concepts, Text Parsing, Sentiment, Topics, Categories.	Text Analytics
Text Analytics: Topic Discovery	Text Analytics pipeline that uses text parsing and machine learning to discover topics.	Text Analytics

Working with Pipelines

<i>Overview of Pipelines</i>	105
<i>Creating a New Pipeline</i>	106
<i>Actions on the Pipeline</i>	107
<i>Automated Pipeline Creation</i>	107
<i>Modifying a Pipeline</i>	109
<i>Creating a Template from a Pipeline</i>	110
<i>Running a Pipeline</i>	111
<i>Node Status</i>	111
<i>Comparing Pipelines</i>	112
<i>Insights Tab</i>	112
<i>Managing Models</i>	114
Register Models	114
Publish Models	115
Export Models for Production	116
Score Holdout Data	117
Download Score API	118
<i>Downloading Logs</i>	118

Overview of Pipelines

Model Studio projects are built around one or more pipelines. A *pipeline* is a process flow diagram that can be used to represent a sequence of analytical tasks. These analytical tasks are represented as individual nodes in a pipeline.

By default, the initial pipeline for a project uses the template that was specified when the project was created. You can create new pipelines using different templates, and you can make changes to the initial pipeline.

Creating a New Pipeline

In Model Studio, pipelines contain the nodes that process data and create models. A project can contain multiple pipelines.

To create a new pipeline:

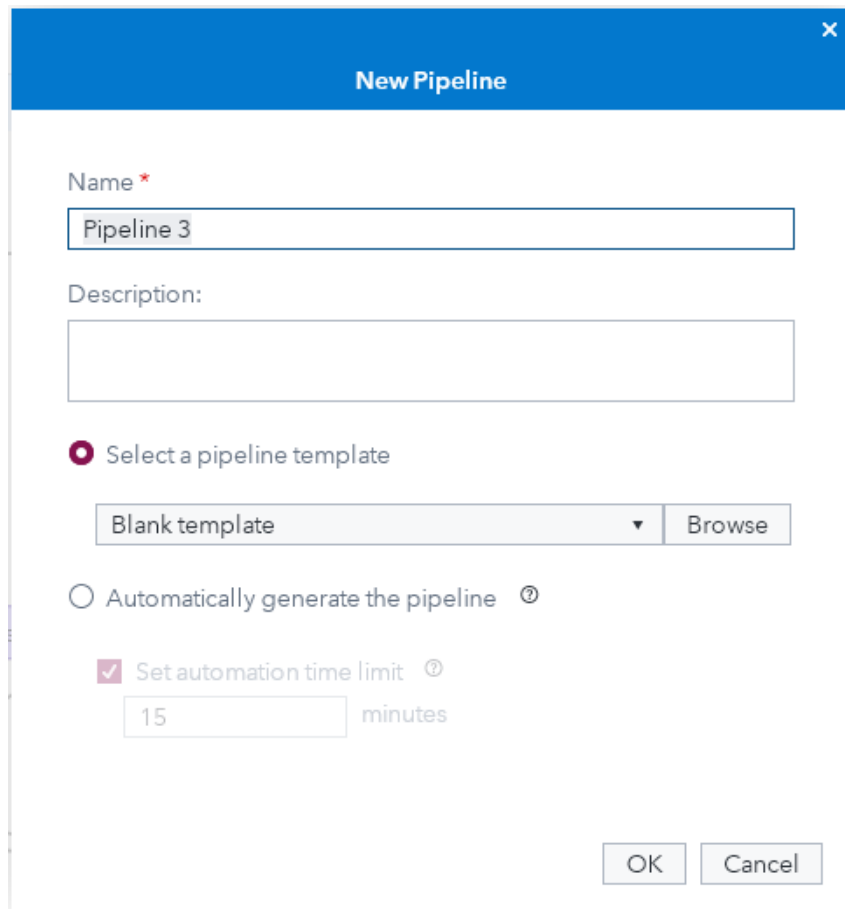
- 1 Navigate to the **Pipelines** tab.
- 2 Click **+** next to the current pipeline tab in the upper left corner of the canvas.




The New Pipeline window appears.

- 3 Give the pipeline a name and an optional description.
- 4 Click **OK**.

In the **Template** field, your recently used templates are available. To use a template that you have not used recently, select **Browse templates** and select a template in the Browse Templates window.

A screenshot of the 'New Pipeline' dialog box. The dialog has a blue header with the title 'New Pipeline' and a close button (X). The main area contains several fields and options: a 'Name' field with a red asterisk, containing the text 'Pipeline 3'; a 'Description' field; a radio button selected for 'Select a pipeline template' with a dropdown menu showing 'Blank template' and a 'Browse' button; an unselected radio button for 'Automatically generate the pipeline'; and a checked checkbox for 'Set automation time limit' with a value of '15' and the unit 'minutes'. At the bottom right are 'OK' and 'Cancel' buttons.

Note: Automatic pipeline generation is available only in SAS Visual Data Mining and Machine Learning projects.

You can also duplicate a pipeline. Click  next the current pipeline tab in the upper left corner of the canvas and click **Duplicate**.

Note: The duplicate functionality is not available in SAS Visual Text Analytics 8.5.

Actions on the Pipeline

Click  on the current pipeline tab to perform the following actions:

- **Run** — Runs the entire pipeline.
 - **Stop** — Stops the run when the pipeline is running.
 - **Duplicate** — Creates a duplicate pipeline. The name is appended with a number. You can rename the duplicate after it is created.
-

Note: The duplicate functionality is not available in SAS Visual Text Analytics 8.5.

- **Rename** — Renames the pipeline.
- **Save to The Exchange** — Saves the pipeline with the nodes and any settings applied to those nodes as a template to The Exchange. The new templates can be used in other projects.
- **Delete** — Deletes the pipeline. This option is available when you have more than one pipeline in your project.
- **Show overview map** — Places a map of the pipeline in the upper left corner of the canvas.
- **Expand header** — Provides a space at the top of the tab to add a description or other text that might be useful. The text can be formatted.
- **Unlock** — For automatically generated pipelines, select this option to edit the pipeline.
- **Logs** — For automatically generated pipelines, select this option to view the pipeline creation log.

Automated Pipeline Creation

SAS Visual Data Mining and Machine Learning can use automated machine learning to dynamically build a pipeline that is based on your data. This process

automatically performs data preparation, model building, model comparison, and model selection on your data to create a pipeline. During pipeline construction, shared projects are locked to the user who started the process, even if that user exits the project.

To enable SAS Visual Data Mining and Machine Learning to automatically create your pipeline, select **Automatically generate the pipeline** in the New Pipeline window.

The screenshot shows the 'New Pipeline' dialog box. It has a blue header bar with the text 'New Pipeline' and a close button (X). The main area is white. It contains a 'Name *' label followed by a text input field containing 'Pipeline 3'. Below this is a 'Description:' label followed by a larger text input field. There are two radio button options: 'Select a pipeline template' and 'Automatically generate the pipeline'. The 'Automatically generate the pipeline' option is selected and has a help icon (i). Below this is a checked checkbox for 'Set automation time limit' with a help icon (i), followed by a text input field containing '15' and the word 'minutes'. At the bottom right are 'OK' and 'Cancel' buttons.

Pipelines that are created by SAS automation are indicated by the ⓘ icon on the name tab.

After your pipeline is created, it is in a locked state and has not been run. This means that the pipeline cannot be edited and results are not immediately available. The pipeline also includes a note that indicates it was initially generated by SAS automation. If you set a time limit or manually stop the pipeline generation, the best pipeline that is generated in that time frame is displayed, but it might not be the optimal pipeline.

To run the pipeline, click the **Run Pipeline** button in the project workspace. You do not need to unlock the pipeline to run it.

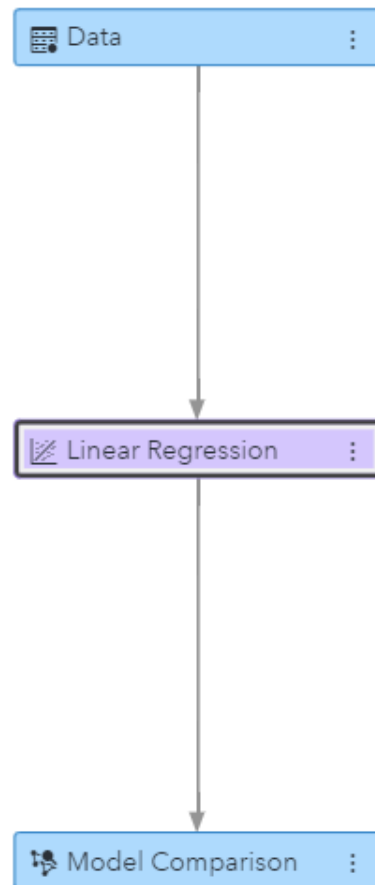
To edit this pipeline, including the note, you must unlock it first. To unlock a pipeline, click the ⋮ button on the pipeline tab and select **Unlock**. Unlocking a pipeline enables you to edit the pipeline as if it were a user-created pipeline. The original note remains until you edit or remove it.

Note: The data advisor must finish running, and a variable must be assigned the role **Target** before the option to select automated pipeline creation is available.

Modifying a Pipeline

After creating a new pipeline, you are ready to create functionality by adding nodes to the pipeline in either of the following ways:

- 1 Expand the **Nodes** pane on the left side of the canvas. By default, the **Blank Template** still contains a data node. These steps assume that you are starting with the **Blank Template**.
- 2 Select a node from the **Data Mining Preprocessing**, **Supervised Learning**, or **Miscellaneous** sections, click and drag it so that the node icon is positioned over the **Data** node, and release the cursor. The new node is added to the canvas, automatically connected to the **Data** node.



Similarly, you can add more nodes to the pipeline, either connected to the **Data** node, or to the other nodes. Click and drag the new node so that the icon is positioned over the existing node. The new node is added to the pipeline,

connected to the node that it was positioned over. See the restrictions (described below) that govern how nodes can be connected to each other.


- 3 Alternatively, right-click on the **Data** node, and select **Add child node**. Select a node from the **Data Mining Preprocessing**, **Supervised Learning**, or **Miscellaneous** options.

Similarly, you can connect more nodes to existing nodes by either selecting **Add child node** (creating more successor nodes) or **Add parent node** (creating predecessor nodes). As with the other method for adding nodes, there are some restrictions as to how nodes can be connected to each other.

You can also delete nodes from your pipeline by right-clicking the node, selecting **Delete**, and clicking **Delete** on the Delete window.


Model Studio has a series of rules that govern the positioning of nodes:

- 1 **Data Mining Preprocessing** nodes can follow the **Data** node or other **Data Mining Preprocessing** nodes. They cannot follow **Supervised Learning** or **Postprocessing** nodes.
- 2 **Supervised Learning** nodes can follow the **Data** node or **Data Mining Preprocessing** nodes. They cannot follow **Postprocessing** nodes or other **Supervised Learning** nodes.
- 3 **Postprocessing** nodes can follow only **Supervised Learning** nodes. They are invalid elsewhere.
- 4 **Miscellaneous** nodes can follow any Model Studio nodes except for the **Model Comparison** node. The **Model Comparison** node is generated when any **Supervised Learning** node is added to the pipeline, and is automatically connected to follow the added **Supervised Learning** node.
- 5 The **Data Exploration**, **Model Comparison**, **Save Data**, and **Score Data** nodes are terminal nodes—that is, no nodes can follow these nodes in your pipeline.

You can also add notes to your pipeline. This can be useful if multiple users are working in the same project. Click  next the current pipeline tab in the upper left corner of the canvas and click **Expand header**. Click **Collapse header** to hide the notes.



Creating a Template from a Pipeline

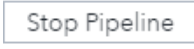
To create a template from a pipeline:

- 1 Click  next to the pipeline tab in the upper left corner of the canvas.
- 2 Select **Save to The Exchange**.
- 3 In the Save Pipeline to The Exchange window, give the template a name and an optional description.
- 4 Click **Save**.

Running a Pipeline

There are two ways to run a pipeline:

- 1 Run all the nodes of the pipeline sequentially, starting with the **Data** node. This is done by clicking the  button in the upper right corner of the canvas. This can also be done by clicking  next to the current pipeline tab in the upper left corner of the canvas and clicking **Run**.
- 2 Run one branch of the pipeline, running only the selected node, and all nodes preceding that node by arrows. This is done by right-clicking a node, and selecting **Run**.


To interrupt a running pipeline, click the  button in the upper right corner of the canvas.


Node Status

As you look at a pipeline, hold your pointer over each node to see its status.

- When you first create the pipeline, all nodes have a status of **Initialized**.
- After a node in the pipeline has successfully finished execution, the node shows a status of **Successful**.
- A node with a status of **Pending** is waiting for other nodes in the pipeline to complete before starting.
- When a node starts executing its functions, its status is **Running**. You cannot make any changes to the node until it finishes running.
- If a pipeline run fails, check to see which node has a **Failed** status.
- If you make any changes to the project settings, variable assignments, or the project training table the nodes change to **Out-of-date** status, and the pipeline must be run again.
- If a node fails for any reason, while a pipeline is running, the status of the subsequent nodes in the pipeline show **Canceled**.

Comparing Pipelines

Once you have successfully run a **Model Comparison** node in at least one pipeline, you can compare pipelines with different models to see which model gives the optimal result. You can even “compare” a single pipeline with itself; this displays the results for the single model. To see a pipeline comparison, select the **Pipeline Comparison** tab. The **Pipeline Comparison** tab displays the champion model, the algorithm used, and error statistics. The selection statistic used to determine the champion model is specified in the **Rules** section within the Project Settings window. To edit the project settings, click  in the upper right corner of the window and click **Project settings**.

To see alternate statistics, click , and select **Manage columns**. Use the Manage Columns window to add or remove alternate statistics about the pipelines.

The **Pipeline Comparison** tab also displays the various results of the champion model. For more information about these results, see the **Results** section for the given champion model in the [SAS Visual Data Mining and Machine Learning: Reference Help](#) documentation.

Note: The **Pipeline Comparison** tab compares only the champion models for each pipeline by default. If you have multiple algorithms in a single pipeline, use the **Model Comparison** node to compare the performance of each of these individual models. For more information, see [Overview of Model Comparison](#) in the Model Studio reference documentation. If you want to add models to the **Pipeline Comparison** tab, right-click the model node in your pipeline, and select **Add challenger model**.

Another feature available on the **Pipeline Comparison** tab is the Compare window. The Compare window enables users to examine the various accuracy statistics of each of the pipelines directly. To compare multiple pipelines:

- 1 Select at least two pipelines in the left-most column of the comparison table.
- 2 Click **Compare** above the table. The Compare window appears. The Compare window contains a table of fit, lift, and ROC statistics. The window also contains line graphs of the statistics, comparing the data roles for each pipeline.


Insights Tab

You can access a project summary report after successfully running the **Model Comparison** node in at least one pipeline. To see the report, select the **Insights** tab. There are six panels with summary information about the project and champion models.

- **Project Summary** — The Project Summary provides information about the project champion model and the most important input variables. Additional project level information is provided:
 - **Project Target** — The target variable specified in the project.
 - **Overall Average** (interval target) — The average of the target variable values in the project data.
 - **Event Percentage** (class target) — The percentage of events in the project data.
 - **Pipelines** — The total number of pipelines in the project.
 - **Project Champion** — The champion model selected in the project.
 - **Created By** — The user ID of the person who originally created the project.
 - **Modified** — The date and time of the most recent modification made to the project.
- **Project Notes** — The Project Notes panel enables you add notes to the project. This can be useful if multiple users are working in the same project.

Note: The content of the Project Notes panel cannot exceed 4000 characters.

- **Most Common Variables Selected Across All Models** — A bar chart that shows the number of times that an input variable was determined to be an important variable for any model on the **Pipeline Comparison** tab. This includes the champion model for each pipeline and any challenger models added. Variable importance is calculated using a surrogate model, which is a one-level decision tree for each input variable where the target is the predicted class or value. Input variables with a positive importance value are determined to be important.
- **Assessment for All Models** — A bar chart that shows the values of the selection statistic for all models on the **Pipeline Comparison** tab. The models are grouped by pipeline.
- **Most Important Variables for Champion Model** — A bar chart that shows the important variables for the project champion model. The relative importance is determined from the actual model when the champion is a decision tree, gradient boosting, or forest model. Otherwise, relative importance is calculated using a one-level decision tree for each input variable to predict the predicted value as a global surrogate model.
- **Cumulative Lift for Champion Model** (class target) — A chart that displays the cumulative lift as a function of the depth for the project champion model. The data is sorted by the predicted probabilities and then divided into 20 quantiles where each quantile contains 5% of the data. The number of events in each quantile is calculated. The cumulative lift for a particular quantile is the ratio of the number of events across all quantiles up to and including the current quantile and the number of events that would be in the quantile at random.
- **Actual vs. Predicted for Champion Model** (interval target) — A chart that displays the actual mean and the predicted mean of the target variable for the project champion model. The data is sorted by the predicted target values and then divided into 20 quantiles where each quantile contains 5% of the data. The actual mean and the predicted mean of the target are plotted for each quantile.

You can download the data that is used to build each chart by clicking . A CSV file with the chart data will immediately begin downloading.

Several charts have detailed descriptions available that explain how the charts were generated. The descriptions also provide information about how to interpret the charts that is custom to your particular data and models. You can access these detailed descriptions in one of the following ways:

- Expand an individual chart by clicking ↕. The detailed description is in the right panel.
- Click ⓘ on an individual chart. A window appears with the detailed description.

Note: The detailed descriptions are available only in the English language.

Managing Models

Register Models

To register a model, click ⋮ in the upper right corner of the **Pipeline Comparison** tab and select **Register models**. When a model is registered, the model name and pipeline name are concatenated. You cannot register a model if the length of this concatenated string is greater than 58 characters.

If SAS Model Manager is installed on the system, this action registers the model in SAS Model Manager. SAS Model Manager is used to store and organize models in a common model repository. It allows for model governance and model change control over time. To view the selected models in SAS Model Manager, click ⋮ in the upper right corner of the **Pipeline Comparison** tab and select **Manage Models**. For more information, see [SAS Model Manager: User's Guide](#).

If the CASHostAccountRequired group has been created, then Read and Write permissions must be configured for the ModelStore caslib and the ModelPerformanceData caslib. This enables members of the CASHostAccountRequired group to register analytic store models. For more information about how to configure permissions, see [File System Directory Permissions](#).

Note: If you update and re-register a model during the same user session, you must refresh the model in **SAS Model Manager** to see the updates.

The Model Repository service is always available, so you can still register models if SAS Model Manager is not licensed. For more information, see [SAS Viya Administration: Models](#).

Publish Models


You might want to publish a model so that the model can be executed in various run-time engines. To publish models, you must first create a publishing destination. The types of publishing destinations that are supported are CAS, Hadoop, SAS Micro Analytic Service, or Teradata. Information about how to configure publishing destinations can be found in [SAS Viya Administration: Publishing Destinations](#).

In order to publish analytic stores to SAS Micro Analytic Service, a separate location must be created to store the analytic stores. For more information, see [Access to Analytic Store Model Files](#).

If the CASHostAccountRequired group has been created, then Read and Write permissions must be configured for the ModelStore caslib and the ModelPerformanceData caslib. This enables members of the CASHostAccountRequired group to publish analytic store models. For more information about how to configure permissions, see [File System Directory Permissions](#).

Note: You cannot publish to SAS Micro Analytic Service if you are using user-defined formats.

To publish a model once you have created a publishing destination:

- 1 Click  in the upper right corner of the **Pipeline Comparison** tab and select **Publish models**.
- 2 The Publish Models window appears. Select the model publish destination that you created and the model that you want published.

Note: You cannot publish more than one model at a time.

✕
Publish Models

Items to publish:

Name	Published Name	Replace
Gradient Boosting	Gradient_Boosting_896	<input type="checkbox"/>

Destination: ⓘ

CAS (AACASDestination) ▼ [Details](#)

Publish ▼

Cancel

- 3 Click **Publish**. The model is published immediately. If you do not want to reload the table at the same time that you publish content, select **Publish without reloading**. However, when you publish an item to a CAS destination, you must reload the table in order for the newly published content to be accessible.

When you publish a model, that model is also automatically registered to the Model Repository service.


Note: If you are publishing a model to CAS, you must publish the model to the same CAS server where your data is located. Copying data from one CAS server to another is not supported.

Export Models for Production

Some models in SAS Visual Data Mining and Machine Learning are packaged in a single downloadable DATA step code file. Other models are packaged in two parts, score code and a binary file, for efficiency.

Models can be exported from both the **Pipelines** tab and the **Pipeline Comparison** tab.

- On the **Pipelines** tab, select the pipeline that contains your target model. Right-click the champion node, and select **Download Score Code**.

- On the **Pipeline Comparison** tab, select one pipeline, click  in the upper right corner of the **Pipeline Comparison** tab, and then select **Download Score Code**.

Both methods listed above download a ZIP file to the client that contains the model score code. The model score code contains the code generated by the supervised learning node, as well as any data mining preprocessing nodes preceding it.


For models packaged in two parts, the ZIP file contains DS2 code and the analytic stores are saved in the MODELS caslib. The DS2, or DATA step 2, code is also referred to as analytic store code or EP score code. The analytic store score code is a representation of the model pipeline including any pre- or post-processing steps. The second part of the two-part model is the analytical store. Each analytic store is a binary file that contains the state of an analytic procedure after training. There can be multiple analytic stores in a single model. Together, the analytic store code and the analytic store represent your model, and can be used to score new data. The **Download Score Code** action saves the analytic stores in the MODELS caslib. In addition, the analytic store score code is compressed and downloaded to the client. The analytic store score code contains a comment that specifies names of binary files. These binary files are required for use with the score code, and can be found in the MODELS caslib.

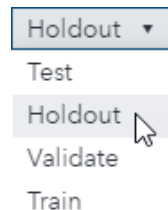
You can identify the path associated with the MODELS caslib by using SAS Environment Manager. For more information, see [SAS Viya Administration: Using SAS Environment Manager](#).

In the CAS server's file system, navigate to the location of the MODELS caslib to find the analytic stores for your model. Each analytic store specified in the analytic store score code is in the MODELS directory with an extension of SASHDAT.

Score Holdout Data


To score holdout data:

- 1 Click  in the upper right corner of the **Pipeline Comparison** tab.
- 2 Select **Score holdout data**.
- 3 The Choose Data window appears. Select the data set that contains the holdout data that you want scored.
- 4 Click **OK**.
- 5 Model Studio scores the holdout data. To see the results of this process, use the **Data** menu below **Pipeline Comparison** to select **Holdout**.




Download Score API

To download the model API:

- 1 Click  in the upper right corner of the **Pipeline Comparison** tab.
- 2 Select **Download score API**. The Scoring API window appears.
- 3 Select the **Download Type**. The following choices are available:
 - **SAS** — Downloads code to run in a SAS programming environment.
 - **Python** — Downloads code to run in a Python programming environment.
 - **REST** — Downloads a text file with example REST calls that you can use in an application.
- 4 After selecting the **Download type**, click **Download**. The model API downloads immediately.

Downloading Logs

To download project logs:

- 1 In the project, click  in the upper right corner (this is accessible on the Data, Pipelines, and Pipeline Comparison tabs).
- 2 Select **Project logs**. You can then select the **Project Partitioning** logs, the **Project Data Advisor** log, or the **Scoring and Assessing** model logs. For pipelines that have not yet run, only the **Project Data Advisor** log is available.
- 3 Once the log is selected, click **Download log**. The log downloads immediately, saved as a TXT file.

Troubleshooting

<i>Enable Debug Reporting</i>	119
<i>Troubleshooting Notes</i>	120
<i>Contact SAS Technical Support</i>	120

Enable Debug Reporting

By default, node logs summarize content in order to save space. To troubleshoot a problem, you might need to enable debug reporting:

- 1 In the upper right corner of the window, click the user name, and select **Settings**.
- 2 Under **Logging**, select **Enable debug reporting**.
- 3 Click **Close**. Debug reporting is now enabled for new pipelines.

Note: In order to enable debug reporting for nodes in an existing pipeline, you must complete the above steps and then rerun the pipeline. If you do not rerun the pipeline, debug reporting is not enabled.

After you enable debug reporting, you can view additional information in the node log:

- 1 Right-click a node and select **Log**. When you view the node log with debug reporting enabled, a Log Size Warning window is displayed.
- 2 Click **Download log**, and use a text editor to view the contents. Additional details that can help you correct problems are displayed in the log.

Troubleshooting Notes

Troubleshooting notes are available on the SAS Support site: <http://support.sas.com/notes>. To search, include the error text along with the words *visual data mining*.

Contact SAS Technical Support

To contact SAS Technical Support:

- 1 In the upper right corner of the window, click the user name, and select **About**.
- 2 Copy the following information:
 - **Product name**
 - **Release**
 - **SAS Viya release**
 - **Build date**
 - **Site name**
 - **Site number**
- 3 Create a track using this form: <https://support.sas.com/ctx/supportform/createForm>. Include the following information:
 - The information from step 2
 - The debug-reporting enabled log if your problem involves a node
 - A description of your pipeline and screenshots