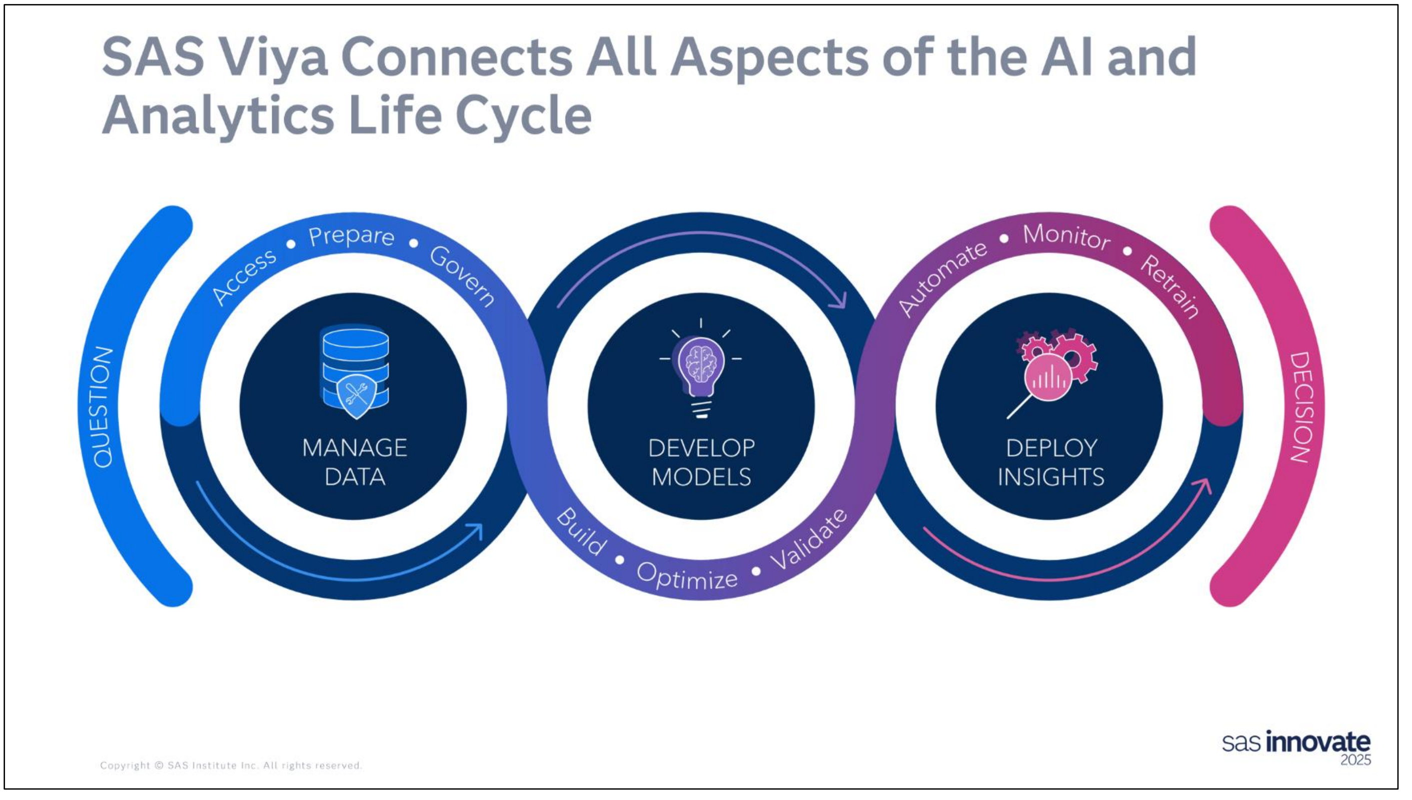
Modeling with Ease: End-to-End Machine Learning in Model Studio

Jeffrey Thompson, PhD
Sr. Analytical Training Consultant

1.1 Introduction

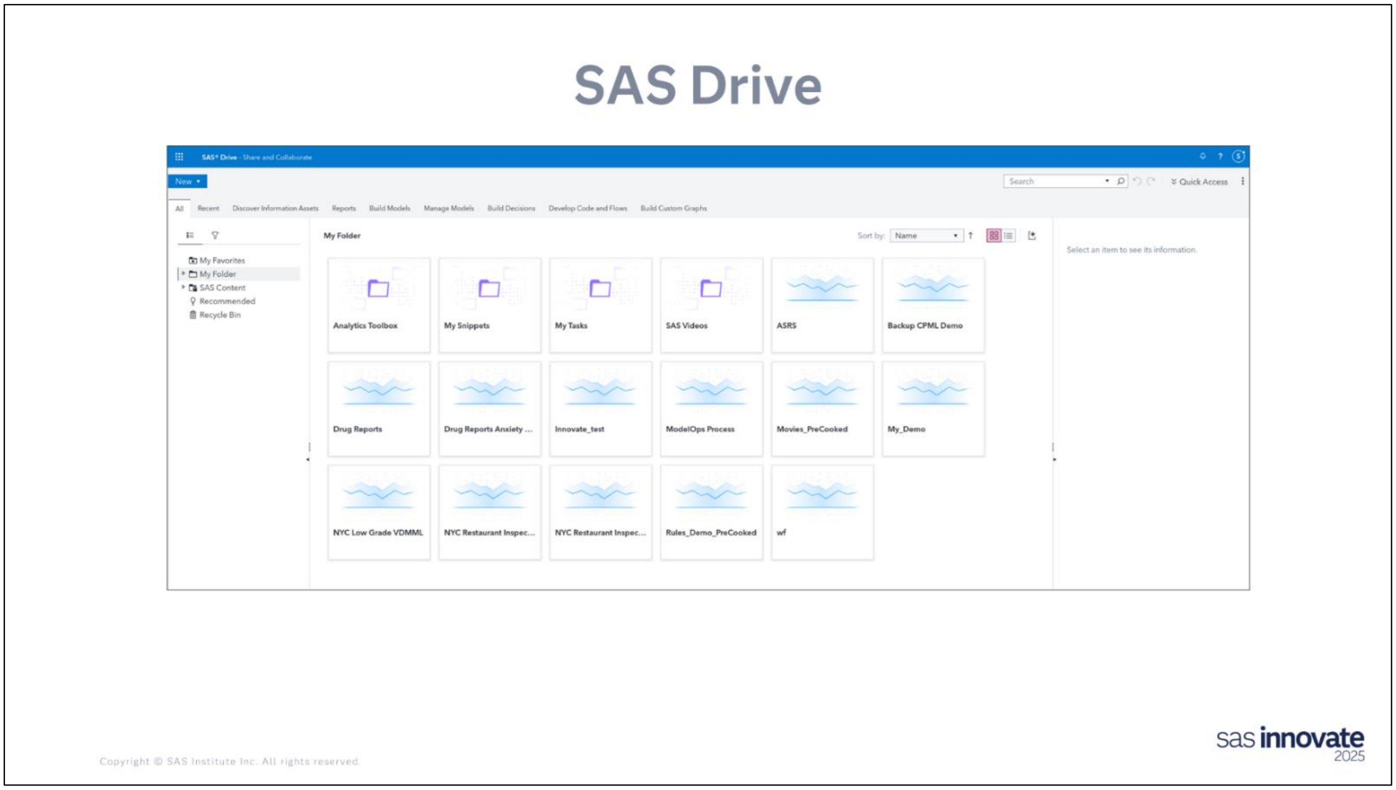
The three phases of the analytics life cycle are ***data***, ***develop***, and ***deployment***. Recognizing and fully supporting all three is necessary to generate impactful insights that come from transforming data into value.



SAS Drive

SAS Drive enables you to quickly access the items and applications that you work with in SAS Viya. The availability of the features on the page depends on your SAS license and the permissions that have been assigned to you by your administrator.

SAS Drive uses the standard sign-in window for SAS applications. To display the sign-in window, enter the URL that is provided by your instructor. After you sign in, you see SAS Drive.

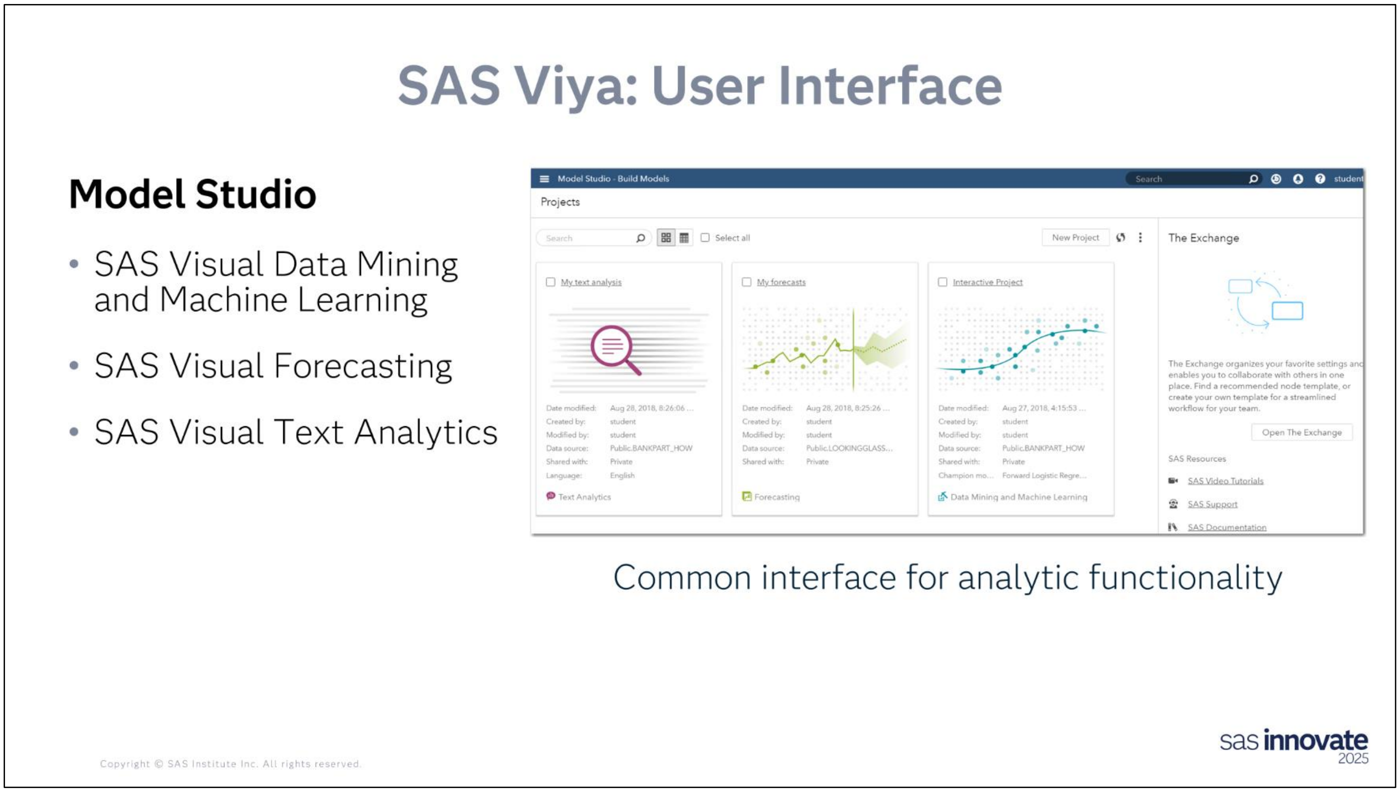


- The application bar at the top enables you to do things such as search for items, view your recent items, and access Help.
- The menu tabs in the main content area are shortcuts that open SAS applications. Note that the tabs represent functional content areas in the analytic workflow. We spend most of our time in this presentation on the Build Models and Manage Models tabs.
- The directory on the left enables you to organize and share your analyses.

For more information about SAS Drive, you can access the SAS Help Center of SAS Drive using the question mark (Help menu icon) on the right side of the application bar at the top of the screen.

Model Studio

Model Studio, included in SAS Viya, is an integrated visual environment that provides a suite of analytic tools to facilitate end-to-end data mining, text, and forecasting analyses. The tools supported in Model Studio are designed to take advantage of the SAS Viya programming and cloud processing environments to deliver and distribute the results of analyses, such as champion models, score code, and results.



Model Studio is a common interface that contains the following SAS solutions:

- SAS Visual Forecasting
- SAS Visual Data Mining and Machine Learning in Model Studio
- SAS Visual Text Analytics

The availability of the functionality in Model Studio depends on your SAS license and the permissions that have been assigned to you by your administrator. You can access the SAS Help Center/Model Studio for more information using the question mark (Help menu icon) on the right side of the application bar.

1.2 BANKPART HOW Data Dictionary

The data to be used for analyses in this workshop consist of observations taken on a large financial services firm's accounts. Accounts in the data represent consumers of home equity lines of credit, automobile loans, and other types of short- to medium-term credit instruments.

The data have been anonymized and transformed to conform to the following description:
A campaign interval for the bank runs for half a year. A campaign is used here to denote all marketing efforts that provide information about and motivate the contracting (purchase) of the bank's financial services products. Campaign promotions are categorized into direct and indirect. *Direct promotions* consist of sales offers to a particular account that involve an incentive. *Indirect promotions* are marketing efforts that do not involve an incentive.

In addition to the account identifier (name: **account**, label: **Account ID**), the data include the following:

A ***target variable*** quantifies account responses over the current campaign season.

Name	Label	Description
B_TGT	Tgt Binary New Product	A binary target variable. Accounts coded with a 1 contracted for at least one product in the previous campaign season. Accounts coded with a zero did not contract for a product in the previous campaign season.

Categorical valued inputs summarize account-level attributes related to the propensity to buy products and other characteristics related to profitability and creditworthiness. These variables have been transformed to anonymize account-level information and to mitigate quality issues related to excessive cardinality.

Name	Label	Description
CAT_INPUT1	Category 1 Account Activity Level	A three-level categorical variable that codes the activity of each account. • *X* ➔ high activity. The account enters the current campaign period with a lot of products. • *Y* ➔ average activity. • *Z* ➔ low activity.
CAT_INPUT2	Category 2 Customer Value Level	A five-level (*A* through *E*) categorical variable that codes customer value. For example, the most profitable and creditworthy customers are coded with *A*.

Interval valued inputs provide continuous measures on account-level attributes related to the recency, frequency, and sales amounts (RFM). These variables have been transformed to anonymize account-level information. All measures below correspond to activity prior to the current campaign season.

Name	Label	Description
RFM1	RFM1 Average Sales Past 3 Years	Average sales amount attributed to each account over the past three years
RFM2	RFM2 Average Sales Lifetime	Average sales amount attributed to each account over the account's tenure
RFM3	RFM3 Avg Sales Past 3 Years Dir Promo Resp	Average sales amount attributed to each account in the past three years in response to a direct promotion
RFM4	RFM4 Last Product Purchase Amount	Amount of the last product purchased
RFM5	RFM5 Count Purchased Past 3 Years	Number of products purchased in the past three years
RFM6	RFM6 Count Purchased Lifetime	Total number of products purchased in each account's tenure
RFM7	RFM7 Count Prchsd Past 3 Years Dir Promo Resp	Number of products purchased in the previous three years in response to a direct promotion
RFM8	RFM8 Count Prchsd Lifetime Dir Promo Resp	Total number of products purchased in the account's tenure in response to a direct promotion
RFM9	RFM9 Months Since Last Purchase	Months since the last product purchase
RFM10	RFM10 Count Total Promos Past Year	Number of total promotions received by each account in the past year
RFM11	RFM11 Count Direct Promos Past Year	Number of direct promotions received by each account in the past year
RFM12	RFM12 Customer Tenure	Customer tenure in months

Demographic variables describe the profile of each account in terms of income, homeownership, and other characteristics.

Name	Label	Description
DEMOG_AGE	Demog Customer Age	Average age in each account's demographic region
DEMOG_GENF	Demog Female Binary	A categorical variable that is 1 if the primary holder of the account is female, and 0 otherwise
DEMOG_GENM	Demog Male Binary	A categorical variable that is 1 if the primary holder of the account is male, and 0 otherwise
DEMOG_HO	Demog Homeowner Binary	A categorical variable that is 1 if the primary holder of the account is a homeowner, and 0 otherwise
DEMOG_HOMEVAL	Demog Home Value	Average home value in each account's demographic region
DEMOG_INC	Demog Income	Average income in each account's demographic region
DEMOG_PR	Demog Percentage Retired	The percentage of retired people in each account's demographic region

1.3 Hands-On Workshop Demonstration

Creating a Project and Building, Fitting, and Comparing Predictive Models Using Pipelines in Model Studio

This demonstration illustrates loading data into SAS Viya and building predictive modeling pipelines using Model Studio. The demonstration continues with model comparison in Model Studio and finishes with model deployment in SAS Model Manager.

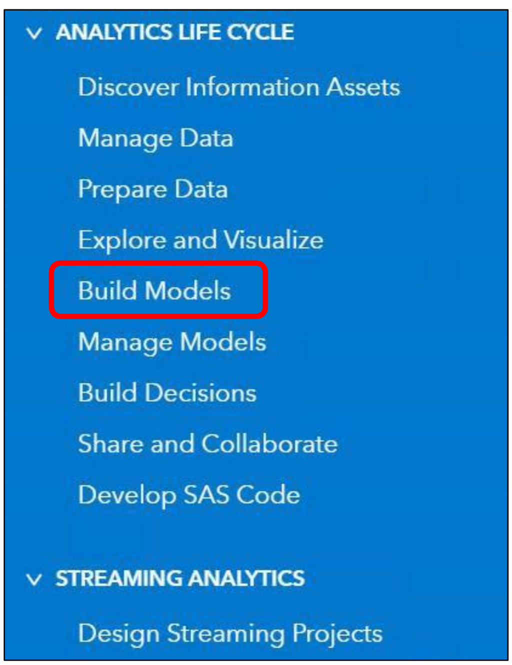
Creating a Project and Building, Fitting, and Comparing Predictive Models Using Pipelines in Model Studio

1. Follow the instructions given by your instructor to access the image desktop.
2. Select the **Google Chrome** shortcut.
3. Select **SAS Drive**.

4. Sign in to SAS using the user name **student** and the password **Metadata0**.
5. If requested to save the password, select **Save**.
6. Select **No** when asked about assumable groups.

Create a Model Studio Project

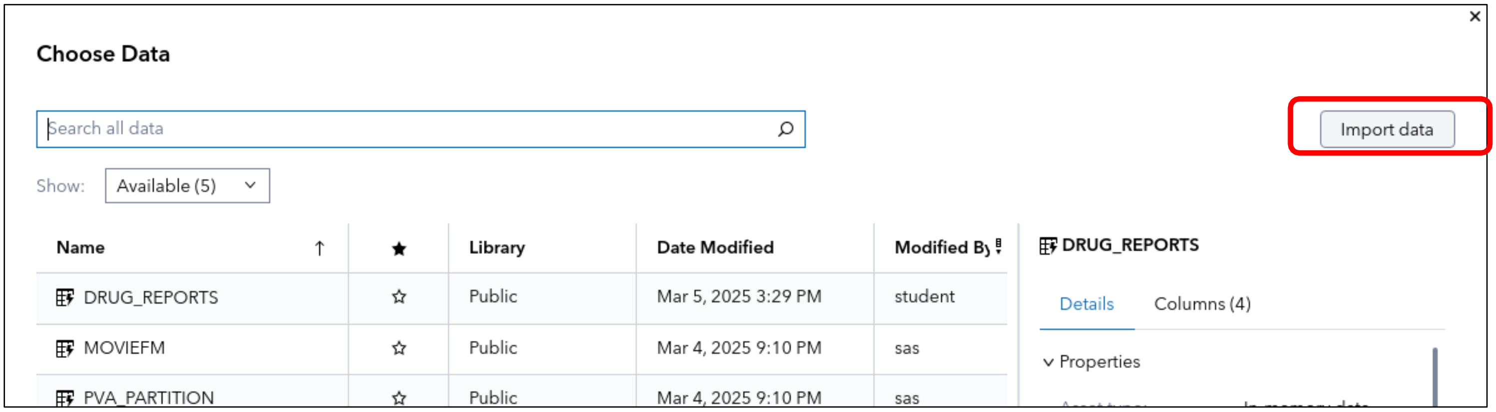
1. Click the **applications menu** icon in the top left and select **Build Models**. This opens the Model Studio web application.



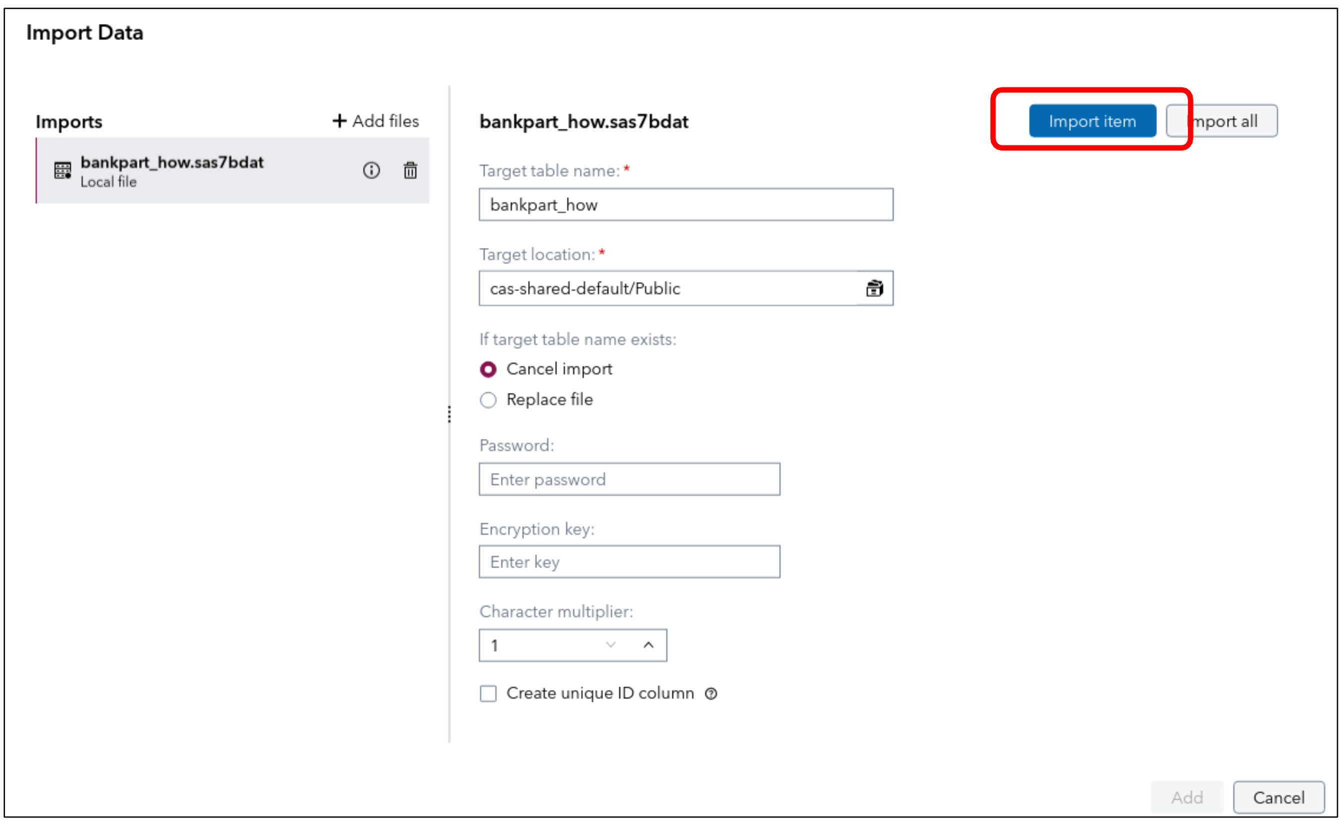
2. From the Model Studio Projects page, select **New Project** (in the upper right corner) to create a new modeling project.
3. Name your project **Model Studio Workshop**.
4. Set Type as **Data Mining and Machine Learning.** (It already should be by default.)
5. Under Template, click **Browse**. Scroll down in the Browse Templates window and select **Blank Template**. Click **OK**.

6. Under the Data option, click **Browse** to choose a data set for this modeling project.

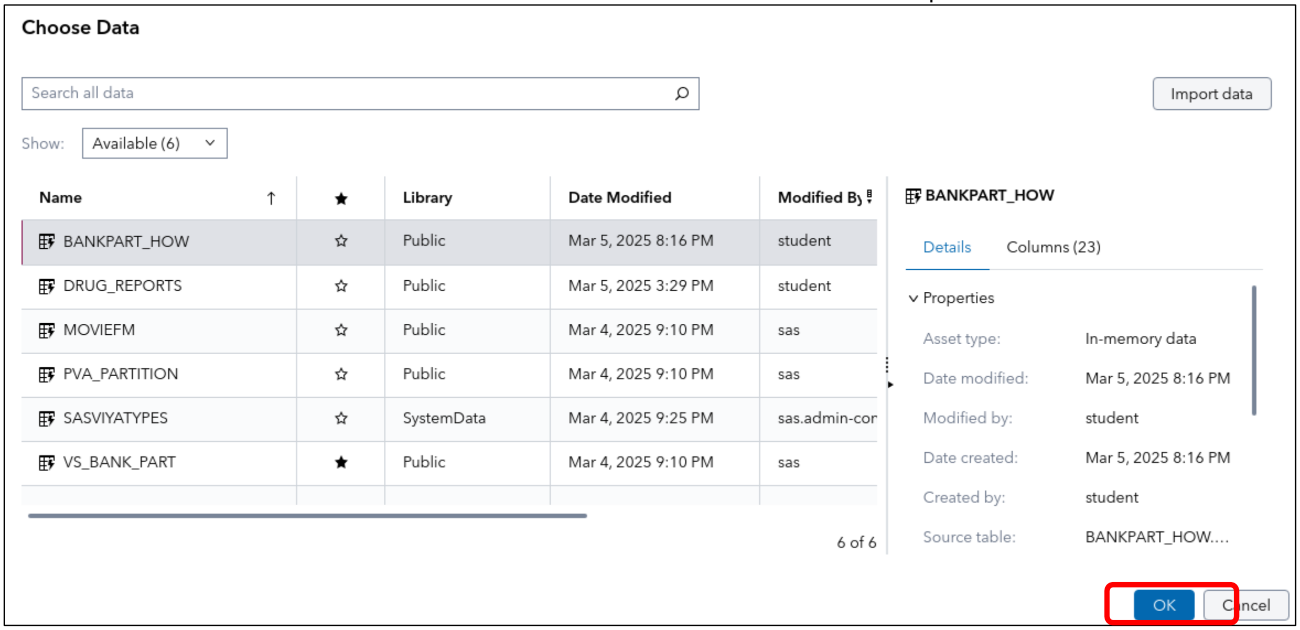
7. We must import and load the BankPart_How data set into CAS memory. From the Choose Data window, select **Import data**.



8. Click **Local files**. Navigate to **workshop** > **SIWMLS_EndToEnd** and select **bankpart_how.sas7bdat**. Click **Open**.
9. In the Import Data window, click **Import item**.



10. Click **Add**. In the Choose Data window, from the Available list, select the in-memory **BANKPART_HOW** table and then click **OK**.



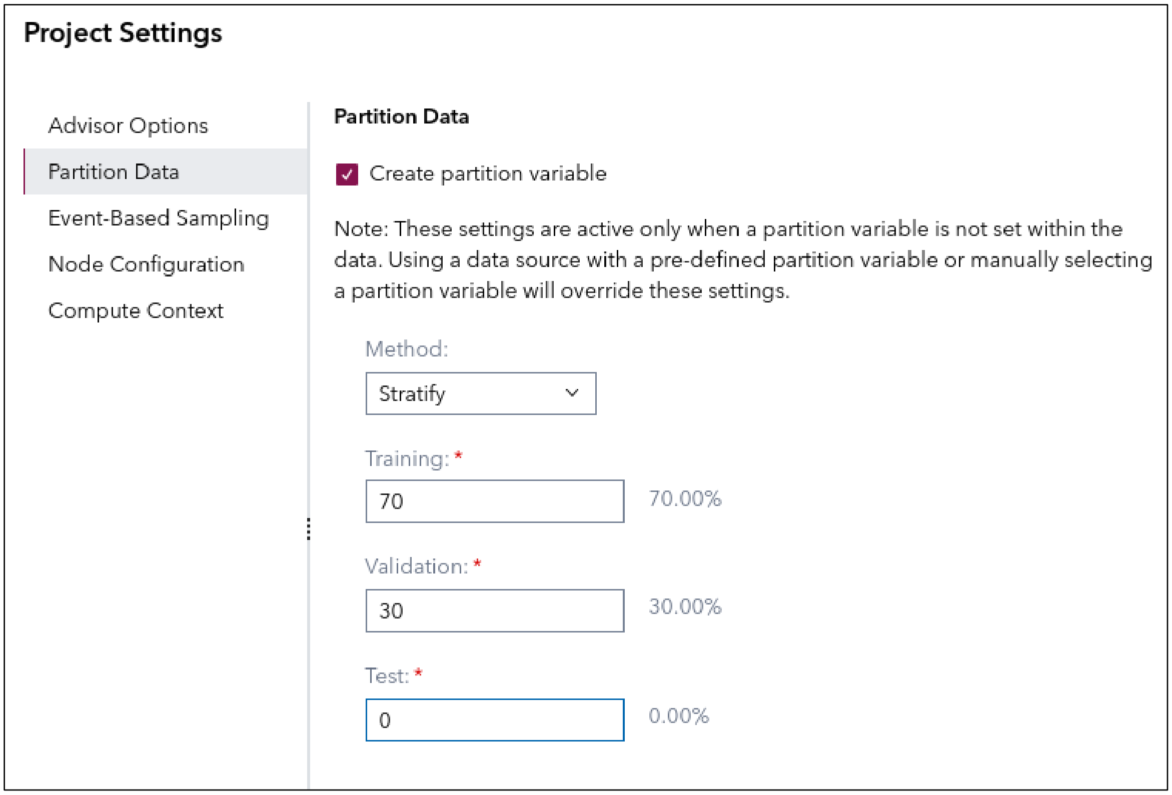
11. The table now appears in the New Project window.

Note: If the table is not loaded into memory (displaying the symbol), you might have to load it into memory by clicking the **Load Into Memory** icon prior to being able to select and load the table into memory.

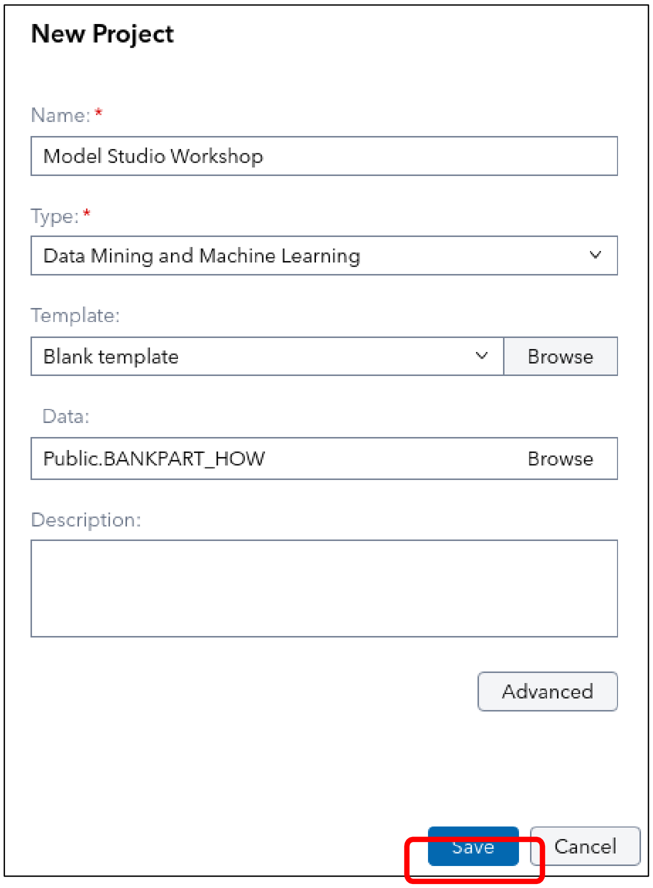
12. Click **Advanced** to open additional Project Settings.

13. Select **Partition Data** to modify partition options.

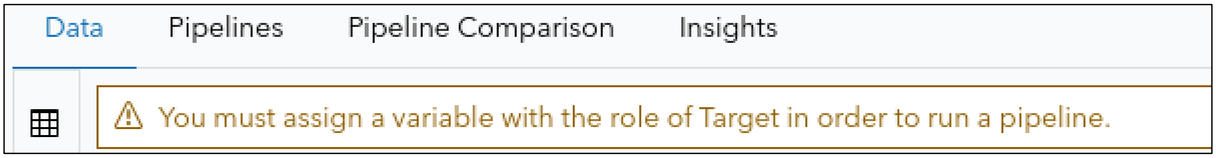
14. Notice that a partition variable is automatically created. Change the Training partition to **70%**, the Validation partition to **30%**, and the Test partition to **0%.**



15. Click **Save** to return to the New Project window.
16. Your settings should match the screenshot below. Click **Save** in the New Project window to create the modeling project.

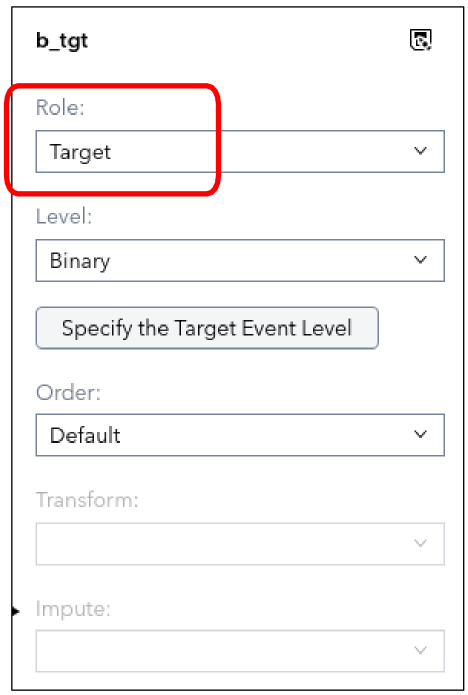

New Project
Name: *
Model Studio Workshop
Type: *
Data Mining and Machine Learning
Template:
Blank template
Browse
Data:
Public.BANKPART_HOW
Browse
Description:
Advanced
Save
Cancel

17. Model Studio opens on the Data tab and displays a warning at the top indicating a variable must be assigned a role of target.



18. Select **b_tgt** (**Binary New Product**) by clicking the check box next to the variable name. In the right-hand pane, set its role to **Target**. (Click the down arrow under Role and choose Target from the drop-down menu.)

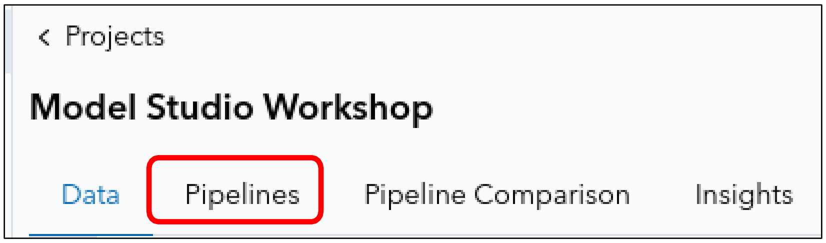
Note: The Data tab contains metadata about the columns in the **BANKPART_HOW** data. Model Studio automatically makes some decisions about metadata but needs the user to specify a target.



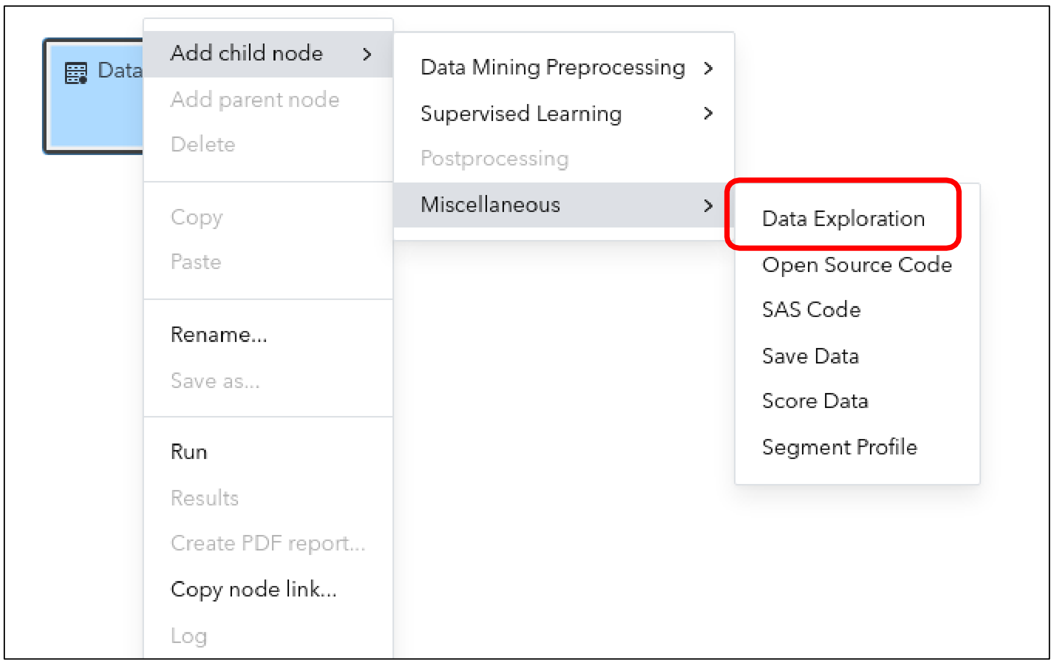
The Role column for b_tgt updates to Target and the warning at the top of the window goes away.

Build and Fit Predictive Models Using Pipelines with Model Interpretability Plots

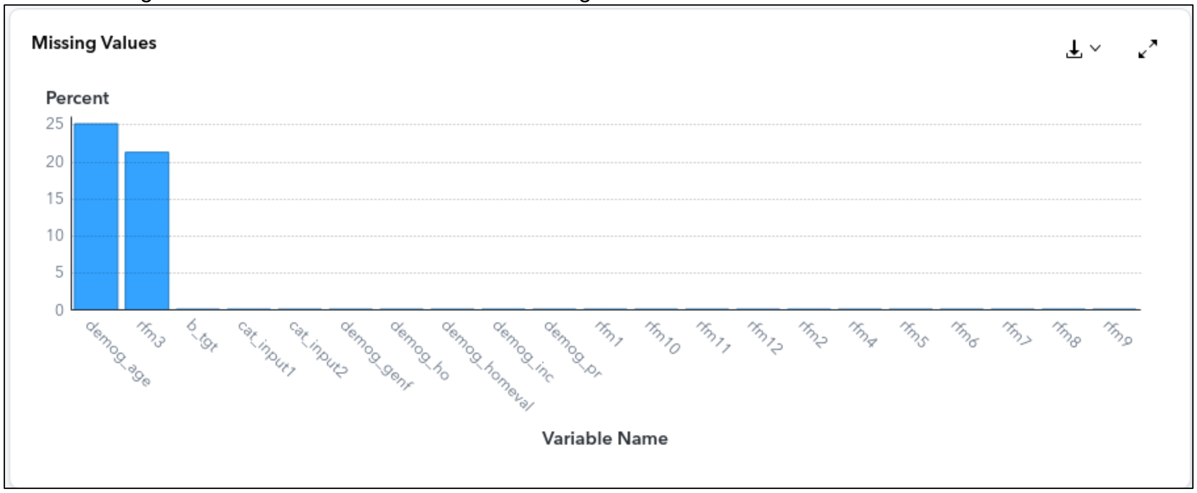
1. Select **Pipelines** on the top left of the page.



2. The default (blank) pipeline starts out with just a Data node. Let's start by exploring the data. Right-click the **Data** node and select **Add child node** > **Miscellaneous** > **Data Exploration**.

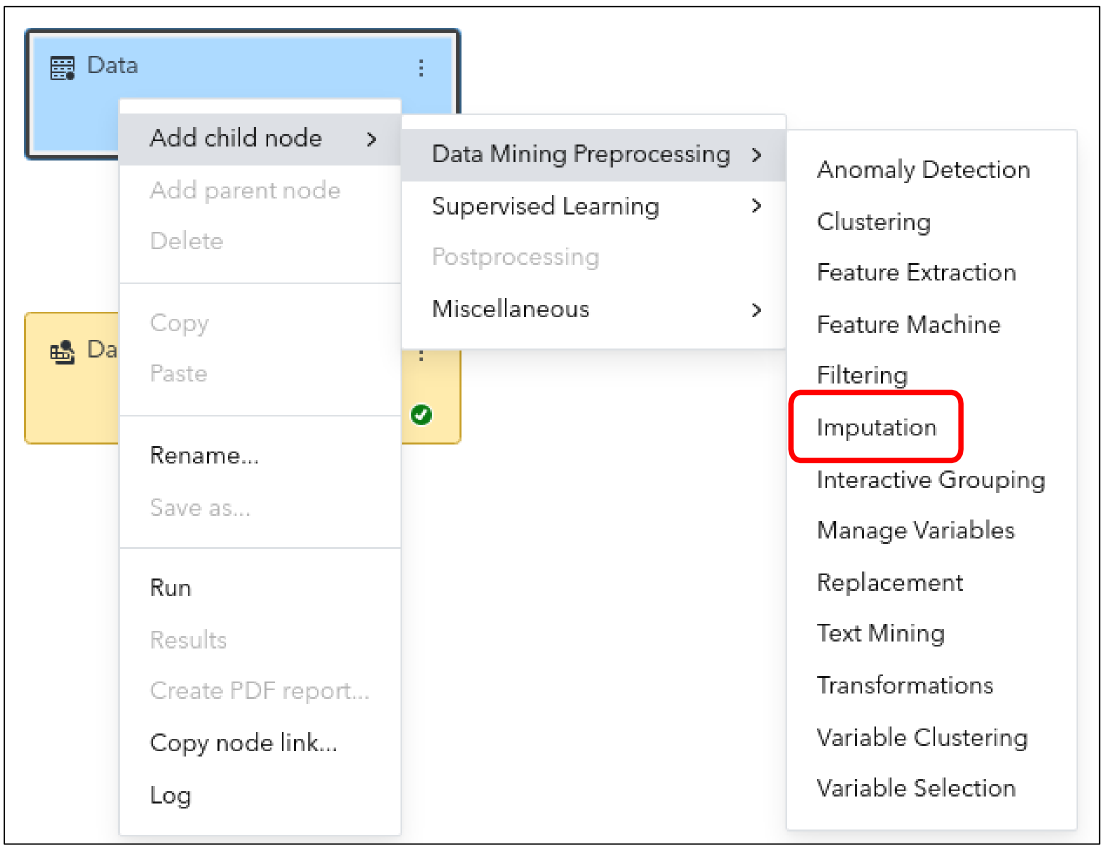


3. Click **Run Pipeline**.
4. When the run is complete, right-click the Data Exploration node and select **Results**. The node provides both numerical and graphical summaries of the data. Scroll down to see the Missing Values plot. Some variables in the data set have missing values.



We will correct for missingness later. Examine other results from the Data Exploration node, such as Important Inputs, Class Variable Distributions, Interval Variable Moments, and Interval Variable Summaries, as you want.

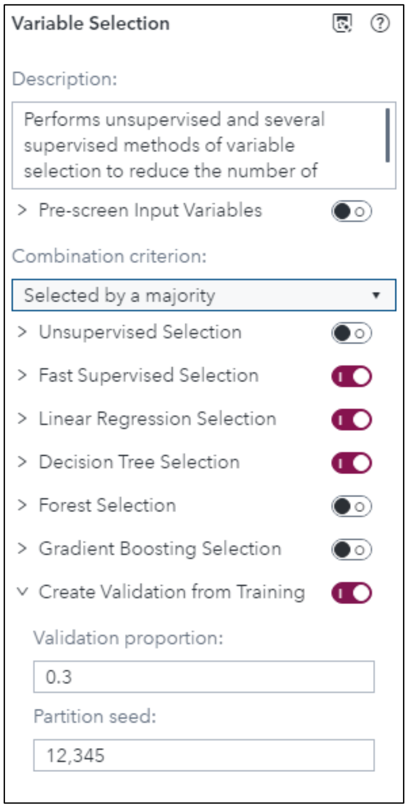
5. Let's do some data preprocessing by first correcting missing values by imputing. Right-click the **Data** node and select **Add child node** > **Data Mining Preprocessing** > **Imputation**.



Click the **Imputation** node to display the settings for imputing missing values in the right-hand pane. By default, missing class inputs are replaced with the mode (**Count** refers to the categorical level with the highest count) and missing interval inputs are replaced with the mean.

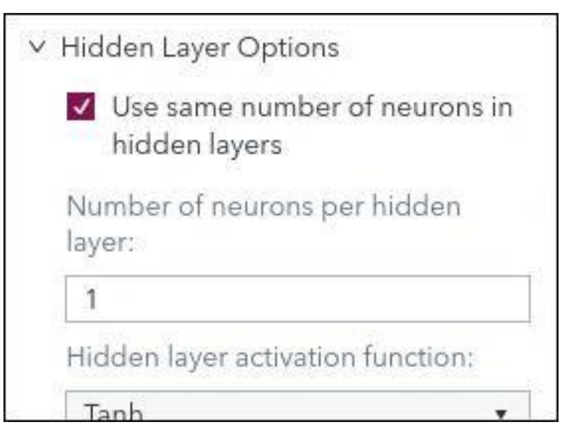
6. Let's do more data preprocessing by selecting a subset of input variables. Right-click the **Imputation** node and select **Add child node** > **Data Mining Preprocessing** > **Variable Selection**.

7. Click the **Variable Selection** node to display the settings. From the **Combination criterion** menu in the right pane, select **Selected by a majority**. Leave **Fast Supervised Selection** on and turn on **Linear Regression Selection** and **Decision Tree Selection** by clicking the toggle switch next to those methods. (When you turn on a property, additional settings for the property are shown. You can hide these additional settings by clicking the down arrow next to the property name.)

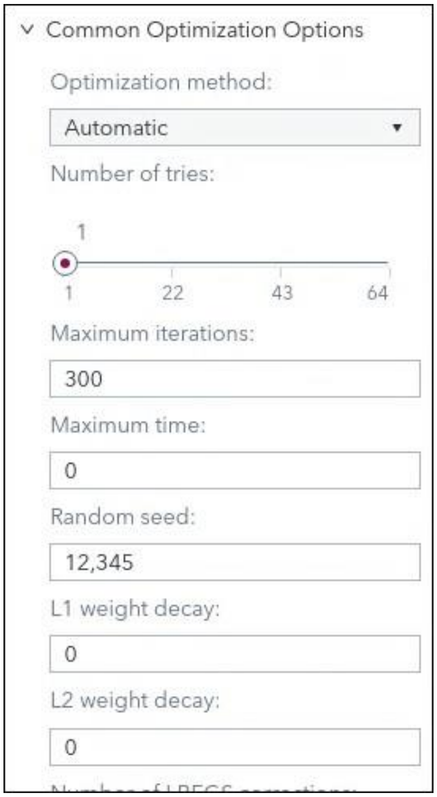


8. Let's build a supervised machine learning model, a neural network. Right-click the **Variable Selection** node and select **Add child node** > **Supervised Learning** > **Neural Network**. A Model Comparison node is automatically added at the end of the pipeline.

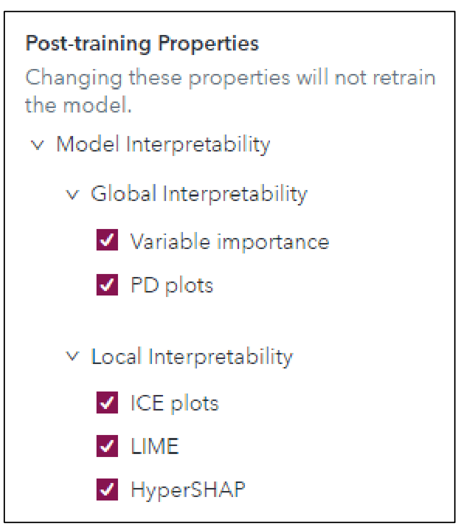
9. Click the **Neural Network** node and in the properties to the right, under **Hidden Layer Options,** set **Number of neurons per hidden layer** to **1**.



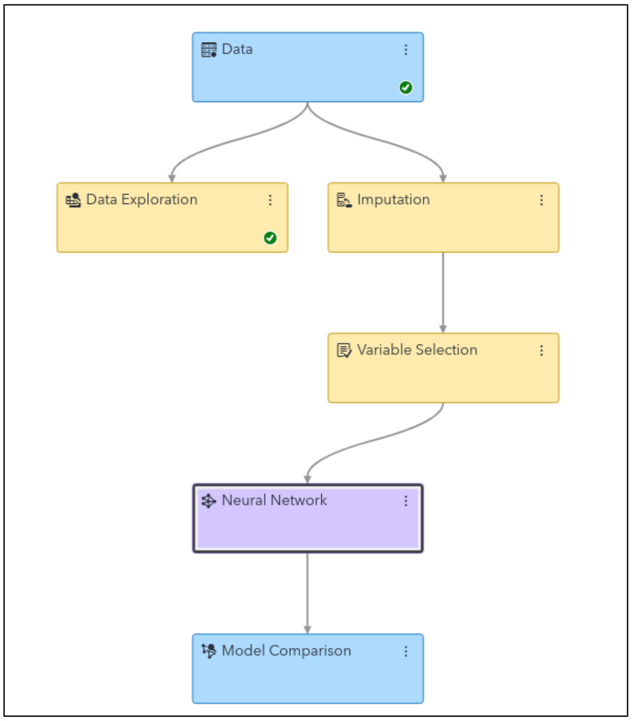
10. Open **Common Optimization Options** (you may need to scroll down in the properties) and set **L2 weight decay** to **0**.



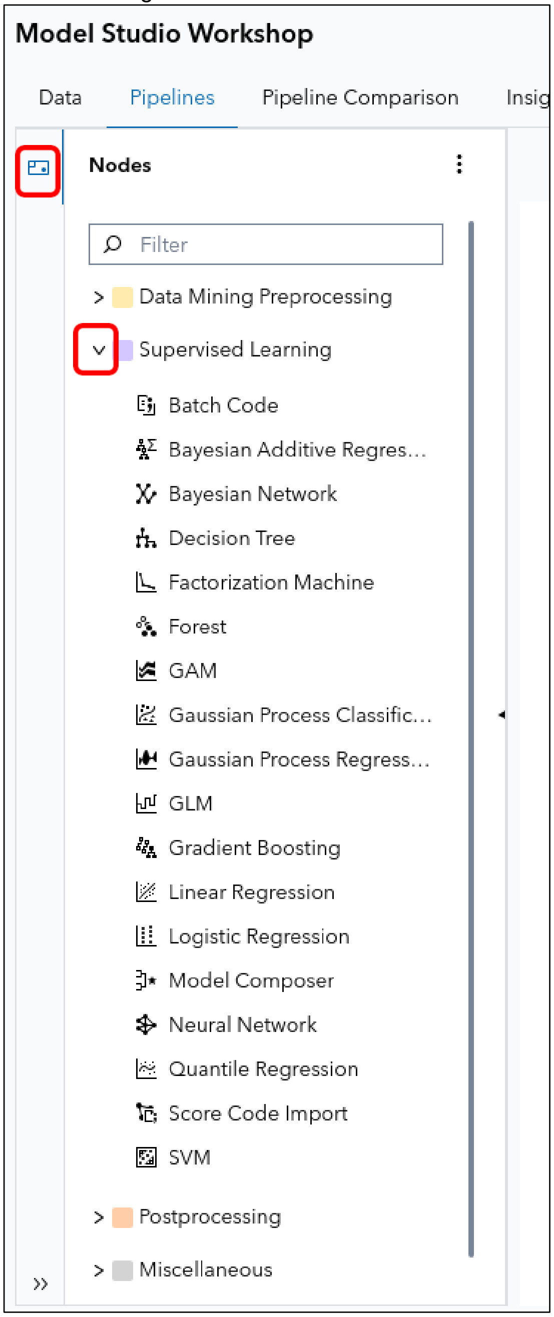
11. Scroll down to the bottom of the options on the right. Under **Post-training Properties**, open **Model Interpretability**. Below that, open **Global Interpretability** and **Local Interpretability**.
12. Select **Variable importance** and **PD plots** under **Global Interpretability**. This turns on a Variable Importance table as well as a Partial Dependence plot.
13. Select **ICE plots**, **LIME**, and **HyperSHAP** under **Local Interpretability**. This turns on the Individual Conditional Expectation plots, the Local Model-Agnostic Explanations, and the Shapley Values plots.



14. Your pipeline should appear as follows:



15. Open the **Nodes** menu on the left of the pane and expand the **Supervised Learning** list to see the supervised machine learning modeling capabilities currently available in Model Studio.



The *Data Mining Preprocessing* nodes provide tools for creating features, selecting variables, and changing values in the data. This includes some tools for unsupervised learning, although the graphical interface is fundamentally designed for supervised learning.

The *Supervised Learning* nodes provide tools for building supervised machine learning models.

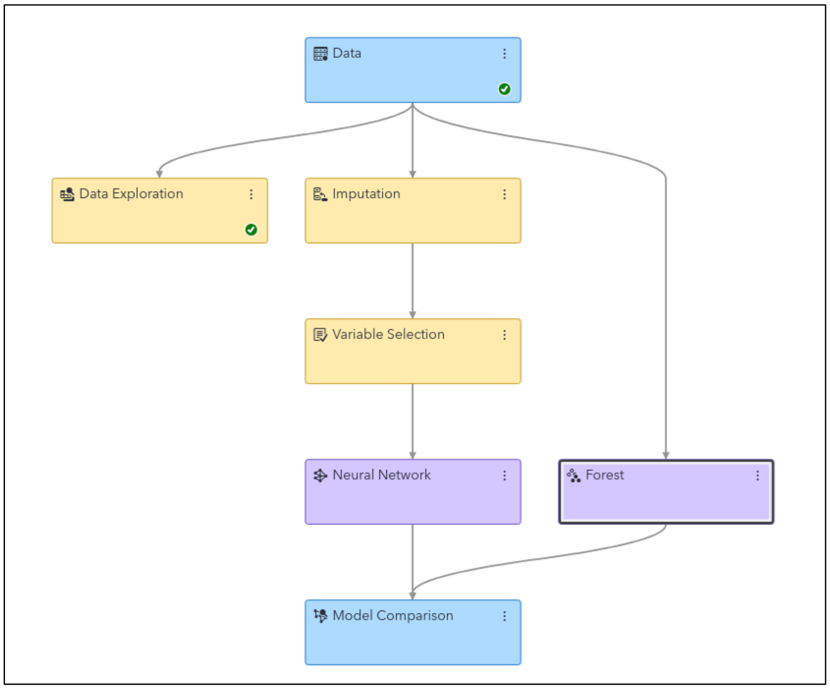
The *Postprocessing* node enables you to create an ensemble of supervised machine learning models.

The *Miscellaneous* nodes enable you to explore data, incorporate SAS or open-source code into the pipeline, and integrate the pipeline with other SAS tools. We explore the Open Source Code node later in this demonstration.

Although we won't have time to discuss most of the nodes available in Model Studio, you are encouraged to experiment by adding any of the nodes that interest you to the pipeline.

16. Drag and drop the **Forest** node from the Supervised Learning list onto the **Data** node. (You can hide the Nodes menu to gain space for the pipelines.)

    Note: The neural network model requires imputation because it ignores any rows with missing values. The forest model can handle missing values and has built-in variable selection, so it can skip the Imputation and Variable Selection nodes.



17. Click **Run Pipeline** in the top right.
18. When the Variable Selection node finishes running, right-click the **Variable Selection** node and select **Results**.

    The Variable Selection table indicates which variables are kept as inputs and which variables are rejected.

Variable Selection

Name	Variable Label	Variable Level	Role	Reason
RFM9	Months Since Last Purchase	INTERVAL	INPUT	
ACCOUNT	Account ID	NOMINAL	ID	
DMINDEX		NOMINAL	KEY	
PARTIND	Partition Indicator	BINARY	PARTITION	
CAT_INPUT1	Account Activity Level	NOMINAL	REJECTED	Combination Criterion
CAT_INPUT2	Customer Value Level	NOMINAL	REJECTED	Combination Criterion

The Variable Selection Combination Summary table indicates which methods chose to keep or reject the input variables. In this case, the Fast Supervised selection, the Linear Regression selection, and the Decision Tree selection methods were active. If a majority of the methods fail to select an input, that input is set to **REJECTED** in the Variable Selection table.

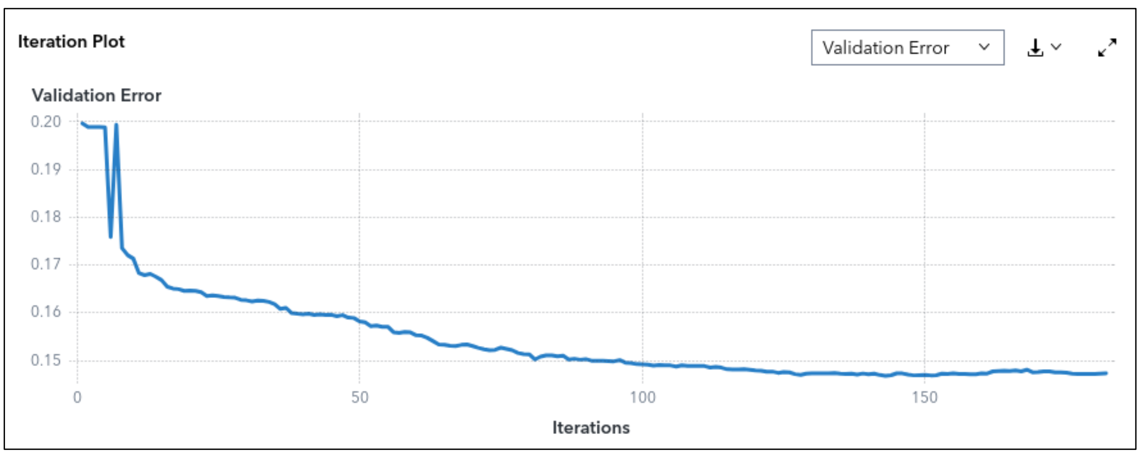
Variable Selection Combination Summary

Name	Variable Label	Fast	Linear Regression	Decision Tree
CAT_INPUT1	Account Activity Level	REJECTED	INPUT	REJECTED
CAT_INPUT2	Customer Value Level	REJECTED	INPUT	REJECTED
DEMOG_GENF	Female Binary	REJECTED	REJECTED	REJECTED
DEMOG_HO	Homeowner Binary	REJECTED	INPUT	REJECTED
DEMOG_HOMEVAL	Home Value	INPUT	INPUT	INPUT
DEMOG_INC	Income	REJECTED	REJECTED	REJECTED
DEMOG_PR	Percentage Retired	REJECTED	REJECTED	REJECTED
IMP_DEMOG_AGE	Imputed Customer	REJECTED	REJECTED	REJECTED

19. Close the results of the Variable Selection node.
20. Right-click the **Neural Network** node and select **Results**.

    The Network Diagram plot provides a visual of the neural network architecture but does not provide any way to interpret the model. We will add model interpretability plots to the neural network results in the next section.

    The Iteration plot indicates that the model tends to improve performance on validation data (new data that it has not seen before) as the neural network is trained.



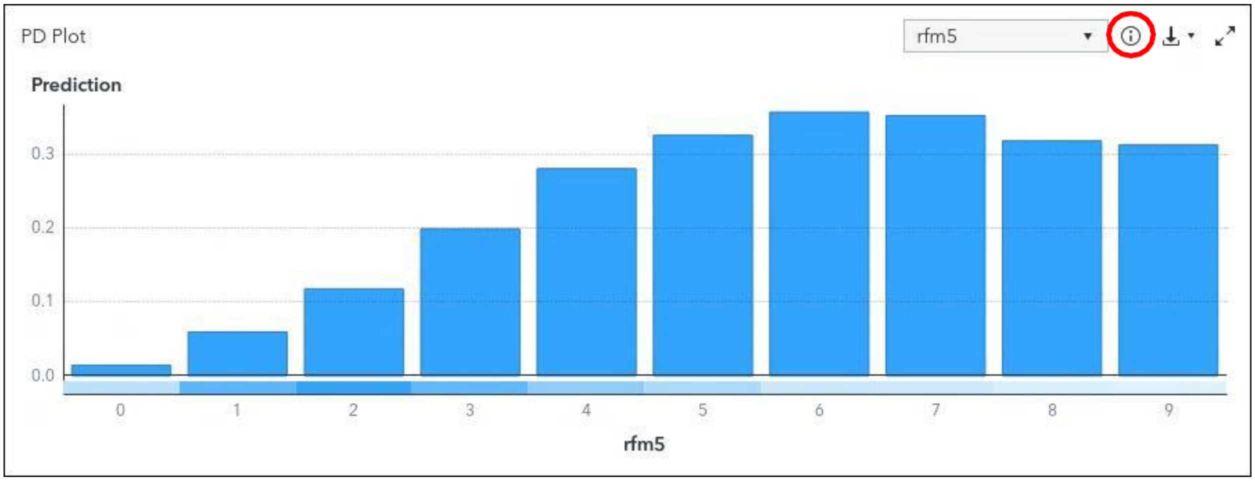
21. Select **Assessment** in the top center to look at model assessment plots and tables that can be created for all predictive models in Model Studio.
22. Click the **Expand** button  on the Fit Statistics table to see a collection of assessment statistics for the model.

Fit Statistics

Target ...	Data Role	Partitio...	Format...	Numbe...	Averag...	Divisor ...	Root A...	Misclas...	Multi-C...	KS (You...
b_tgt	TRAIN	1	1	370,726	0.1047	370,726	0.3236	0.1469	0.3398	0.5920
b_tgt	VALIDATE	0	0	159,107	0.1046	159,107	0.3235	0.1468	0.3393	0.5954

23. Close the Fit Statistics table and click the tab for **Model Interpretability**. This tab appears because we are chose to turn on the model interpretability tools in Model Studio.

24. Explore the different types of model interpretability plots available in SAS Viya. For example, here is the Partial Dependence plot for the imputed **rfm5** variable. (For an automated description of each of the plots, click the **info** button circled below for the corresponding plot.)



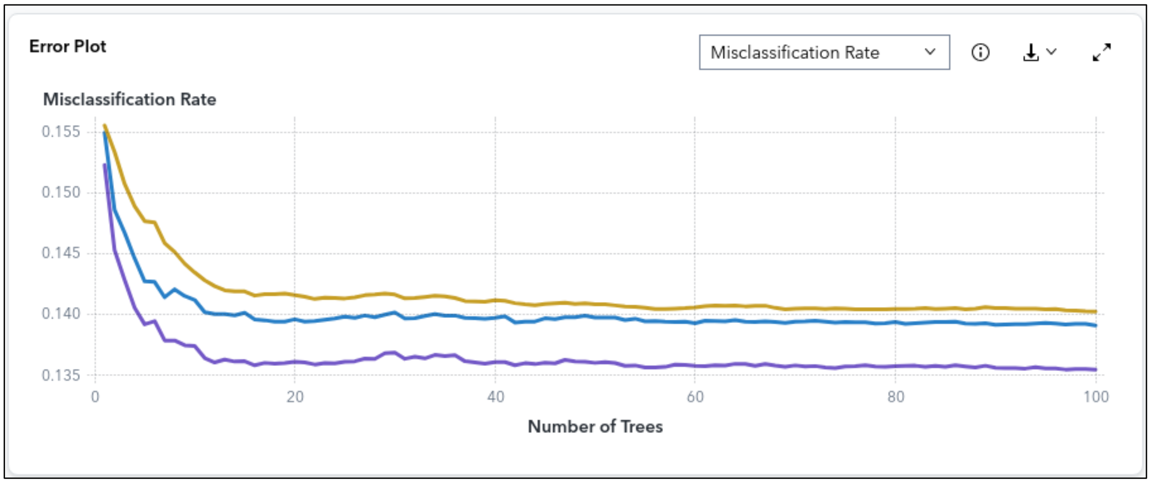
We have more demonstration topics to cover, so we don't have time to explain each of the model interpretability plots in detail. If you want to learn more about model interpretability in SAS, please refer to our tutorial video on these tools: https://www.youtube.com/watch?v=6LcyVSLwVck

25. Close the results of the Neural Network node to return to the pipeline.

26. Right-click the **Forest** node and select **Results**.

27. On the Error plot, change **Average Squared Error** to **Misclassification Rate**.

The Error plot indicates that model performance improves over-all as decision trees are added to the forest. The menu can be used to choose between assessment statistics of misclassification rate and average squared error.

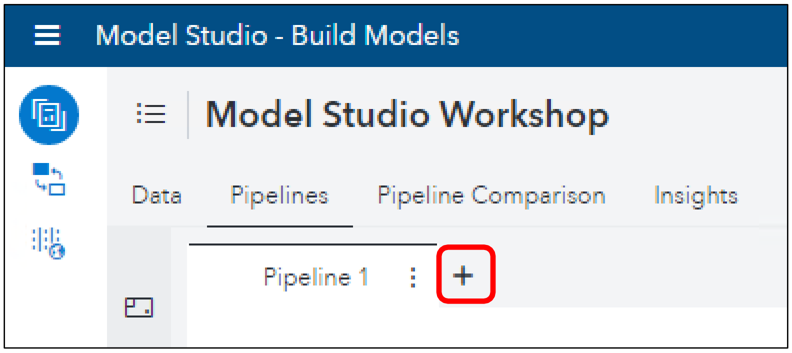


Note: The legend on this plot shows up only if the plot is expanded.

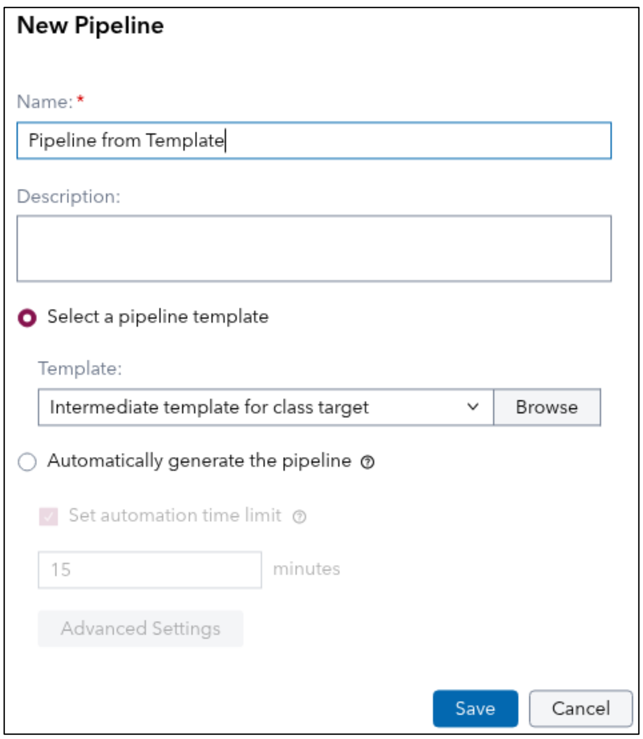
28. Close the results of the Forest node to return to the pipeline.

Build Models Using a Pipeline Template and add an Open-Source Model

1. Click the plus button next to **Pipeline 1** in the top left of the project to create a new pipeline.



2. Name your new pipeline **Pipeline from Template**.
3. Under the **Select a pipeline template** option, click **Browse**.
4. In the Browse Templates window, scroll down and select **Intermediate Template for class target**. Click **OK**.



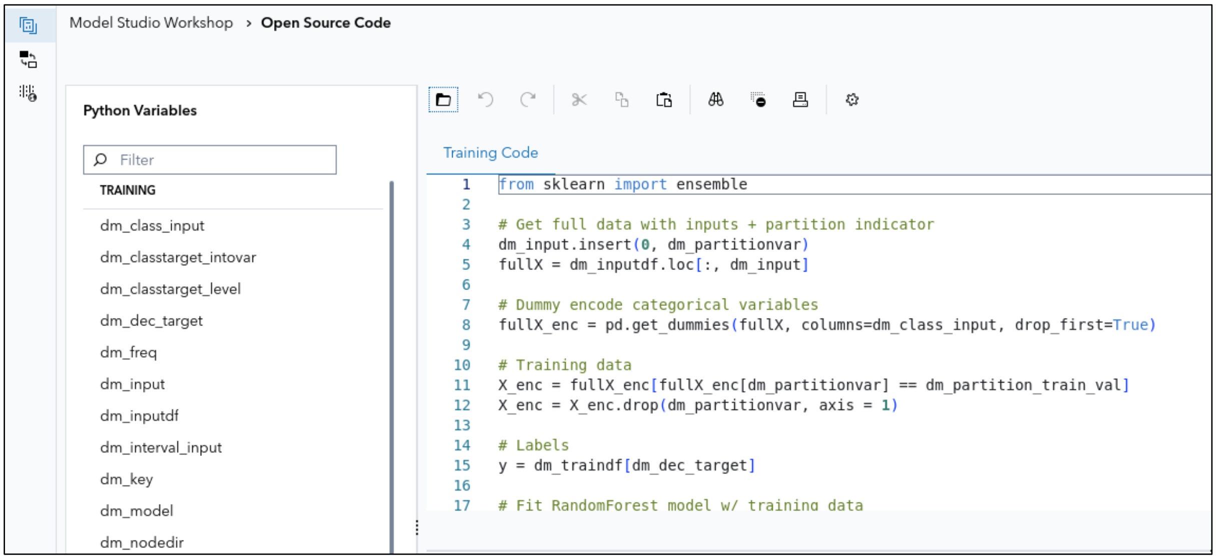
There is also the option to automatically generate the pipeline. This process uses automated machine learning to dynamically build a pipeline that is based on your data. With this option, you can select the amount of time that the pipeline generation process is allowed to run. Be cautioned that this process can be resource intensive.

5. In the New Pipeline window, click **Save** to create a new pipeline from the template.

6. Right-click the **Variable Selection** node and select **Add child node > Miscellaneous > Open Source Code**.

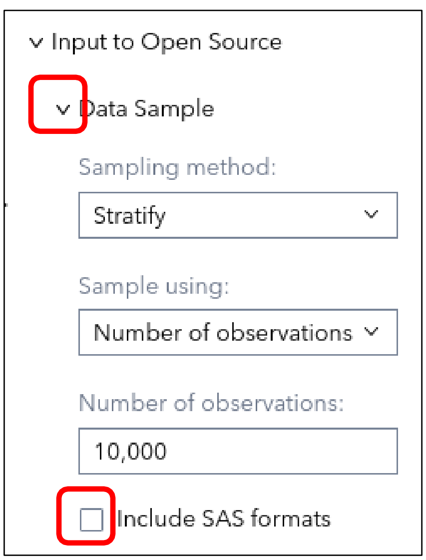
7. In the properties pane of the Open Source Code node, select **Open code editor**.

8. Click in the Training Code window and then click the **Load Source code file** short-cut button . Navigate to **Computer > SIWMLS_EndToEnd**. Select **Python_Forest_Code.py** and click **Open**. Examine the Python forest code.



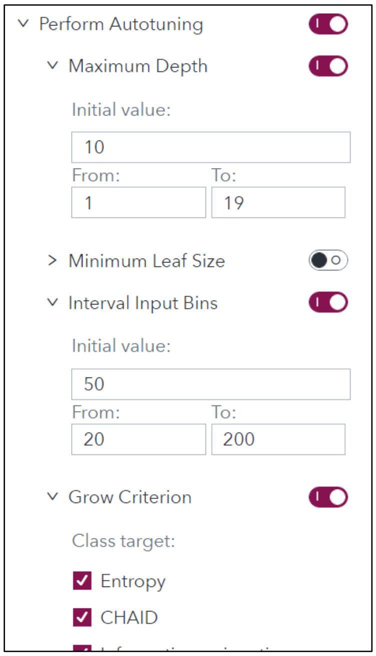
9. Click **Save** and then **Close**.

10. In the properties pane, expand **Data Sample** and deselect **Include SAS formats**. Also, notice that the Language property is set to **Python** (the default setting). R is another option for this property.



11. Right click on the Open Source Code node and select **Move > Supervised Learning**. The node changes to purple indicating it is a supervised model, and the node automatically connects to the Model Comparison node.

12. Click the **Decision Tree**. In addition to selecting options for the decision tree hyperparameters, you can take advantage of autotuning the model.



Autotuning is available for many machine learning models. For a decision tree, many of the tree splitting and tree growing hyperparameters can be autotuned.

13. At this time, <u>do not perform autotuning</u>.
14. Click **Run Pipeline** in the top right.
15. When the pipeline finishes running, right-click the **Model Comparison** node and select **Results**.
16. Expand the Model Comparison table to see additional fit statistics for the four listed models in this pipeline.

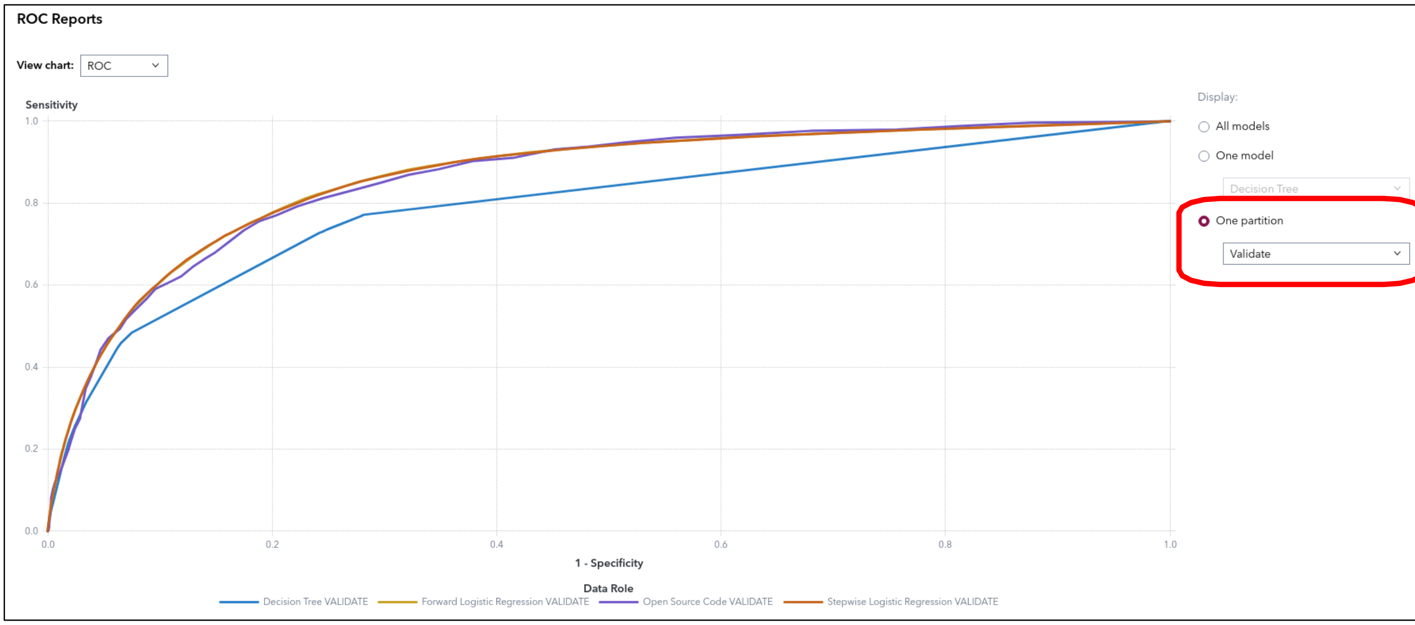
Model Comparison

Cha...	Name	Algorit...	KS (You...	Accuracy	Averag...	Area U...	Cumula...	Cumula...	Cutoff	Data Role	Depth	F1 Score	False
★	Forward Logistic Regression	Logistic Regression	0.5822	0.8486	0.1076	0.8633	3.6228	36.2284	0.5000	VALIDATE	10	0.5276	0.3
	Decision Tree	Decision Tree	0.4901	0.8402	0.1194	0.7907	3.4202	34.2025	0.5000	VALIDATE	10	0.5279	0.3
	Stepwise Logistic Regression	Logistic Regression	0.5789	0.8484	0.1077	0.8629	3.6247	36.2473	0.5000	VALIDATE	10	0.5243	0.3
	Open Source Code	Open Source Code	0.5699	0.8178	0.1186	0.8602	3.6516	36.5159	0.5000	VALIDATE	10	0.1824	0.1

The star next to the Forward Logistic Regression model indicates that it is the pipeline champion (based on the default KS statistic from validation data). Notice that assessment statistics are generated and displayed for all models in the pipeline.

17. Close the Model Comparison table and navigate to the Assessment tab.

18. Expand the **ROC Reports** plot. Select **One Partition > Validate**.

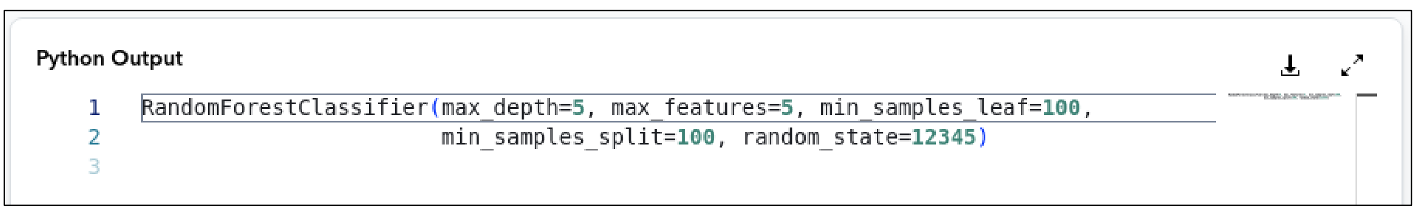


Note: Assessment plots are created for the open source model as well as for the SAS models.

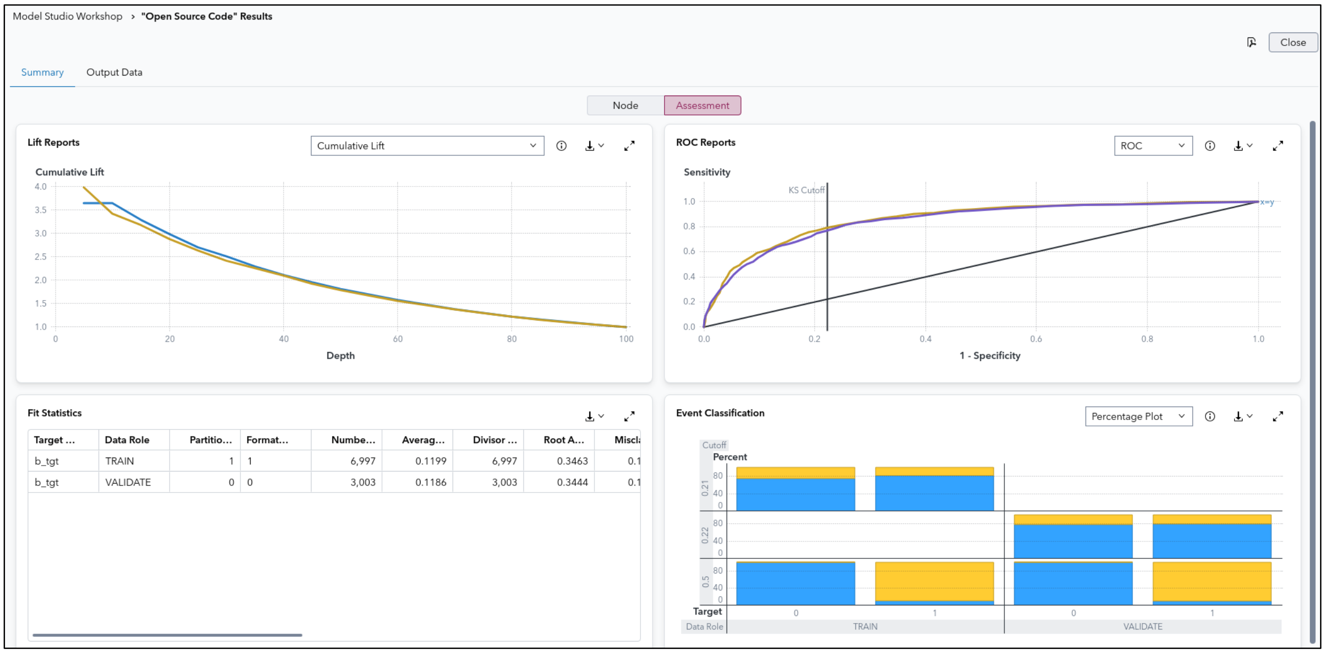
19. Close the plot and close the results of the Model Comparison node.

20. Right-click the **Open Source Code** node and select **Results**.

    The Python Output window shows any output created by Python from the open-source code node. In this case, it indicates that a forest model was built and lists the settings used.



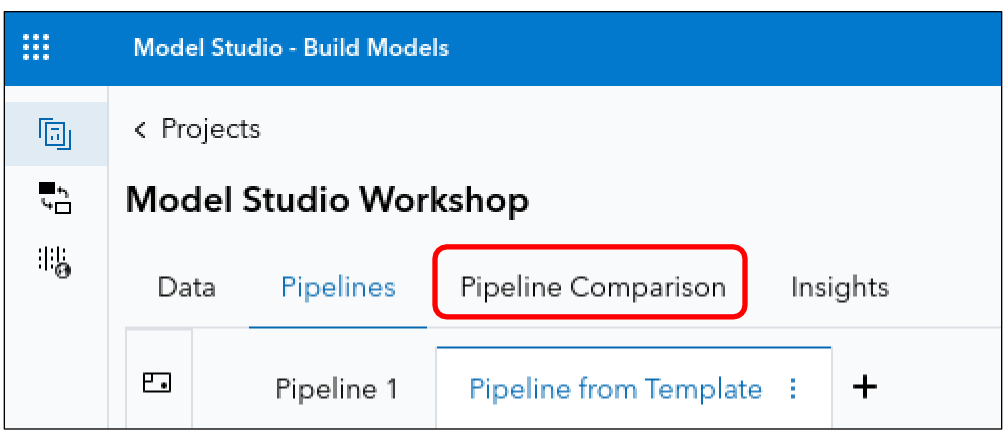
21. Navigate to the Assessment tab to see lift plots, ROC plots, and fit statistics for the open-source model. Notice that the Open Source Code node still produces the same assessment plots and statistics as all the other modeling nodes that SAS provides which allows for a fair comparison between open-source and SAS models.



22. Close the results of the Open Source Code node.

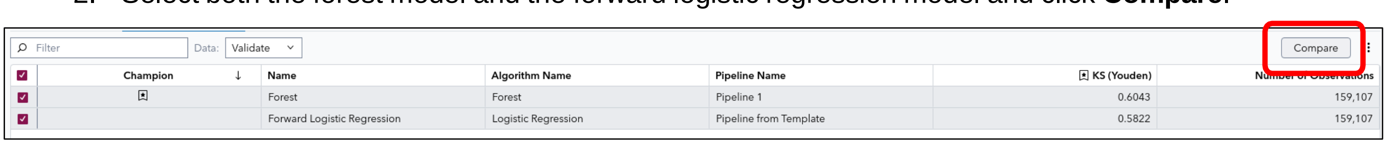
Compare Models and Assess Model Performance

1. Navigate to the Pipeline Comparison tab by selecting **Pipeline Comparison** in the top left of the project.



The forest model from Pipeline 1 is selected as the champion model based on the KS statistic on validation data. Results for the selected model are displayed on the Pipeline Comparison tab.

2. Select both the forest model and the forward logistic regression model and click **Compare**.

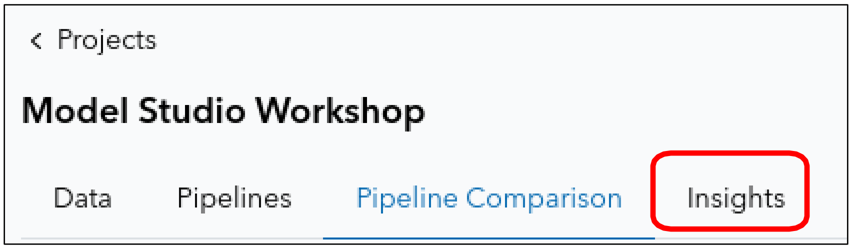


The same type of results that you see in the Model Comparison node appear. This time, the results compare the two selected models (the champions for each pipeline) from the Pipeline

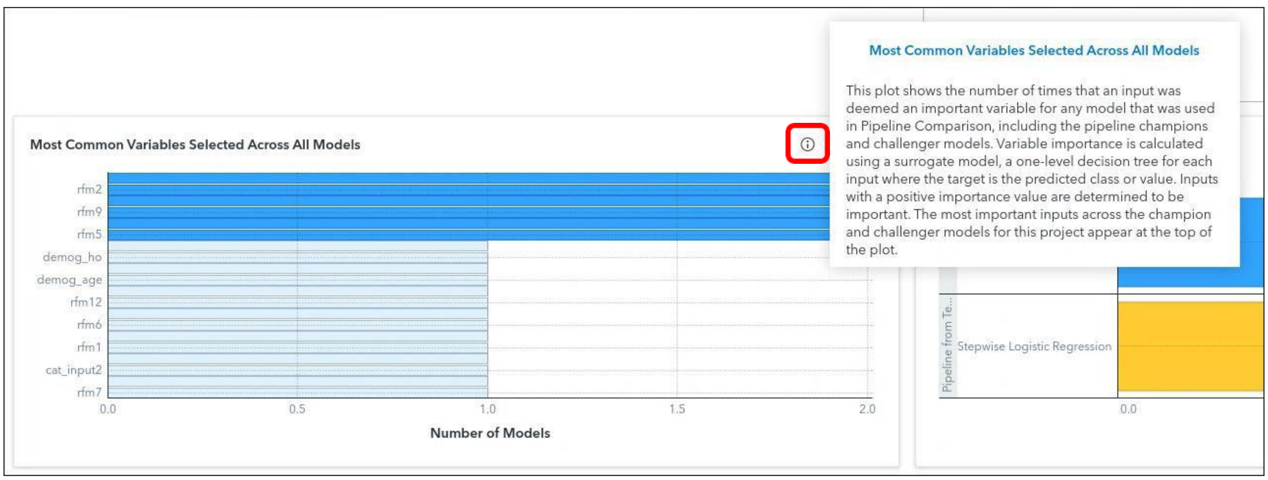
Comparison tab.

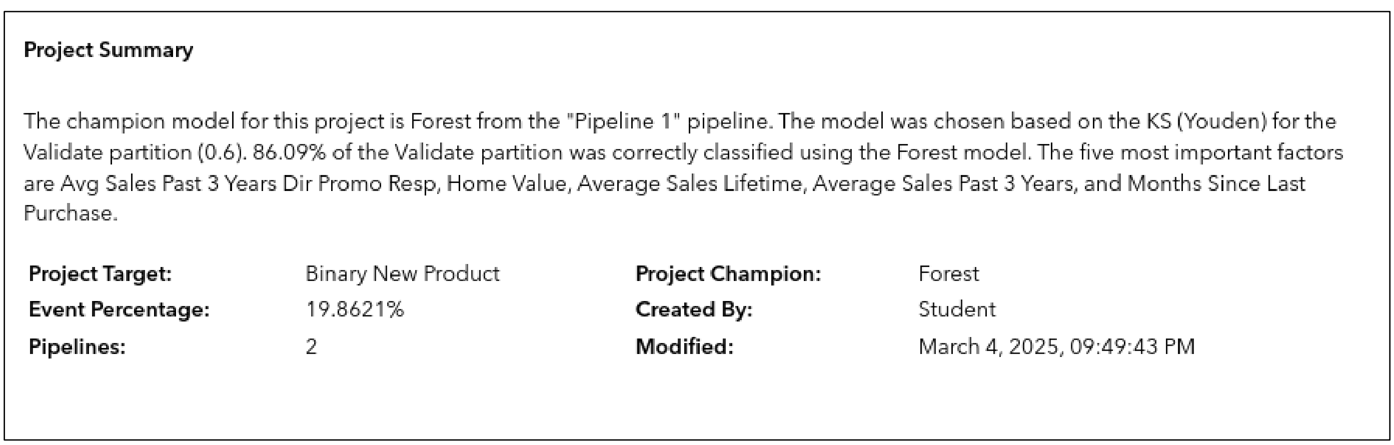
3. Expand the **Fit Statistics** table and notice that the forest model has a validation misclassification of 13.91% and the logistic regression model has a validation misclassification of 15.14%.
4. Close the Fit Statistics table and close the results of the comparison.

5. Navigate to the project Insights tab by selecting **Insights** in the top left of the project. The Insights tab gives a summary of work done in the Model Studio project, with a focus on the champion model selected by the Pipeline Comparison tab.



The Project Summary window contains automatically generated text that explains the champion model and the most important variables used in that model.





The Most Common Variables Selected Across All Models plot shows which variables are used in the models on the Pipeline Comparison tab. Note that the only models that are discussed or described on the project Insights tab are the models that appear on the Pipeline Comparison tab.

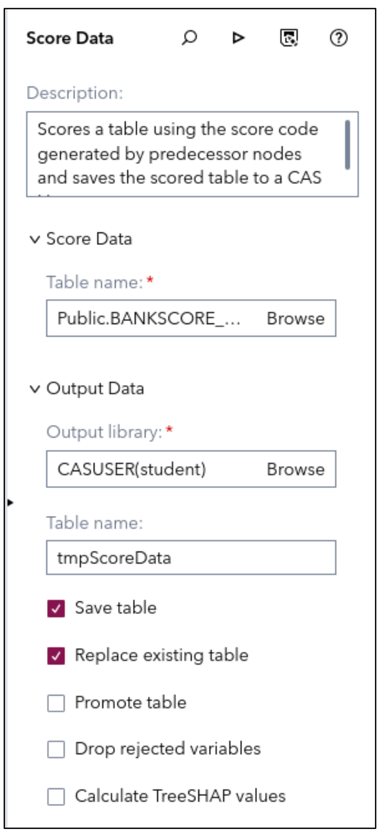
Click the **View an automated description of the results** button to see an automatically generated explanation of the plot contents.

The Project Notes window enables you to write your own description or analysis of the project or its results. This is a good place to put any notes that you found about the data or the modeling effort so that people who open your project later can quickly review what you learned.

Note: In general, the project Insights tab is a good place to start when looking at a project that someone else has created. It provides a high-level overview of the best models and highlights important variables used to create those models. The Project Summary also gives you information about when the project was created and who created it.

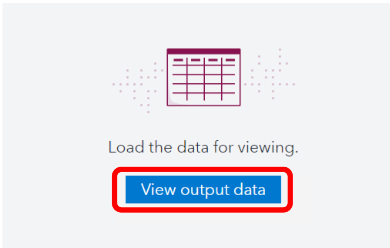
Scoring Data in a Pipeline

1. Navigate back to the Pipelines tab and select **Pipeline 1**.
2. Right-click the **Forest** node and select **Add child node** > **Miscellaneous** > **Score Data**.
3. In the options pane on the right, select **Browse** for **Table name**.
4. We must import and load the table to be scored, BankScore_How, into CAS memory. From the Choose Data window, select **Import data**. Click **Local files**. Navigate to **workshop** > **SIWMLS_EndToEnd** and select **bankscore_how.sas7bdat**. Click **Open**.
5. Click **Import item**. When the import is complete, click **Add**.
6. Select **BANKSCORE_HOW** in the Available list. Click **OK**.
7. Select **Browse** under **Output library**. Select **cas-shared-default** > **CASUSER(student)** and then click **OK**. This is the location of where the scored table will be place in CAS memory.
8. Click the **Replace existing table** check box. Your options pane should resemble the following:



9. Click **Run Pipeline** in the top right.
10. Right-click the **Score Data** node and select **Results**. Click the **Output Data** tab.

11. Select **View output data**.



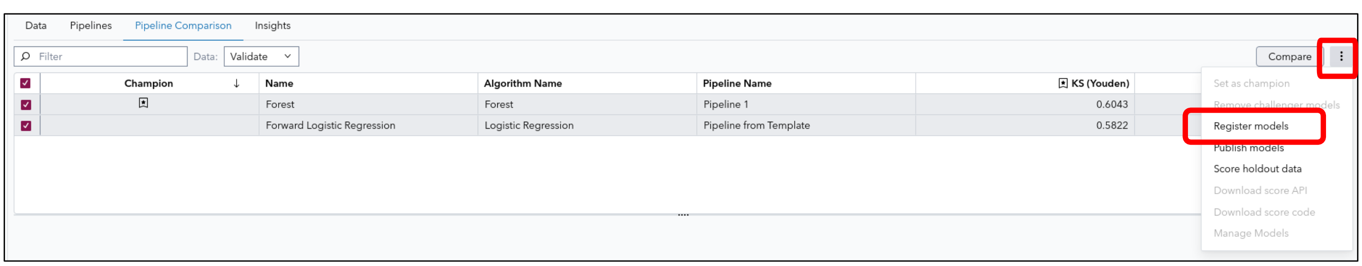
12. Select **View output data** again in the Sample Data window. Do not enable sampling.
13. The first column in the table, **Probability for b_tgt=1**, is the predicted event probability for each case scored; that is, the probability that the customer purchases at least one financial product. You can also scroll to the right of the table to view other scored columns of the data.

Probability f...	Customer Age	Predicted fo...	Home Value	Probability ...	Income	Percentage ...	Average Sal...	Count Total ...	Count Dir
0.1228346991	.	0	$66,854	0.8771653009	$0	33	$14.00	14	
0.2124477273	.	0	$57,396	0.7875522727	$0	29	$10.00	13	
0.0270270752	28	0	$0	0.9729729248	$59,758	25	$20.00	8	
0.0281356798	43	0	$78,486	0.9718643202	$45,762	19	$25.00	13	
0.0449085804	58	0	$67,192	0.9550914196	$41,713	33	$15.00	11	
0.4306054618	47	0	$45,793	0.5693945382	$0	27	$8.08	18	
0.0299829226	29	0	$78,422	0.9700170774	$55,214	34	$22.50	12	

14. Click **Close** to close the Results window.

Manage and Deploy Models

1. Navigate back to the Pipeline Comparison tab.
2. Select the **Forest** model and the **Forward Logistic Regression** model using the check boxes on the left (if not already selected). Click the **More options** button ⋮ on the right and select **Register models**.



3. Click **OK** to use the default location, **/Model Repositories/DMRepository**, to store the registered models.

4. When the activity is completed in the Register Models window and you see that the two models

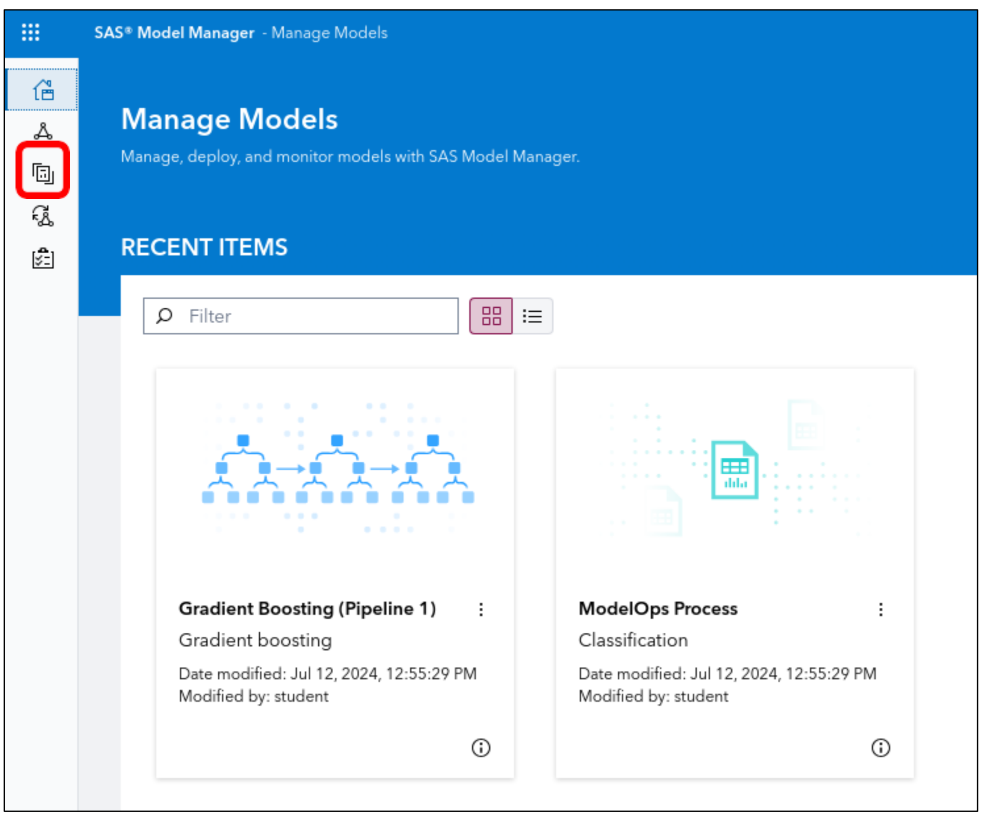
were registered successfully, click **Close** to close the Register Models window.

5. Click the **applications menu** icon in the upper left corner and select **Manage Models** to open the SAS Model Manager web application.

   SAS Model Manager streamlines the managing, deploying, monitoring, and operating aspects of using analytical models. It enables you to maintain a repository of SAS and open-source models for analytical projects. The repository enables easy comparison of registered models, and it has tools to monitor the performance of production models. In the event of model degradation, models can be efficiently retrained and redeployed.

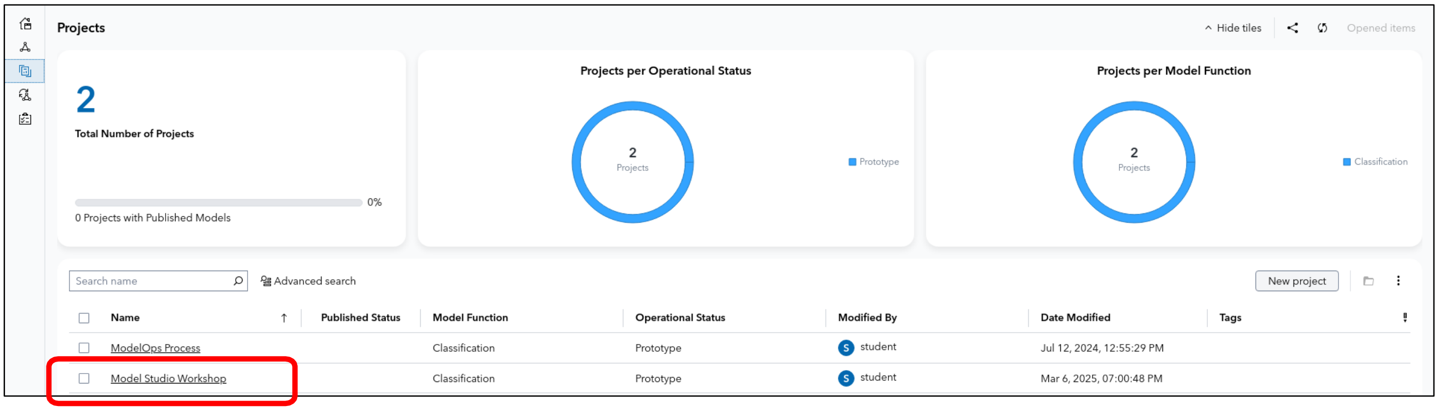
   With SAS Model Manager, we can automate and streamline the analytics life cycle so that models get into production faster. Automated monitoring and governance of models followed by triggering retrain, rebuild, and replace cycles ensure ongoing value to your business.

6. Select the **Projects** page icon in the left column. (Third from the top.)



7. The projects home page opens. It shows that a Model Manager project named Model Studio Workshop (the name of our modeling project in Model Studio) has been created automatically from registering our models. This project contains the two models we registered in Model

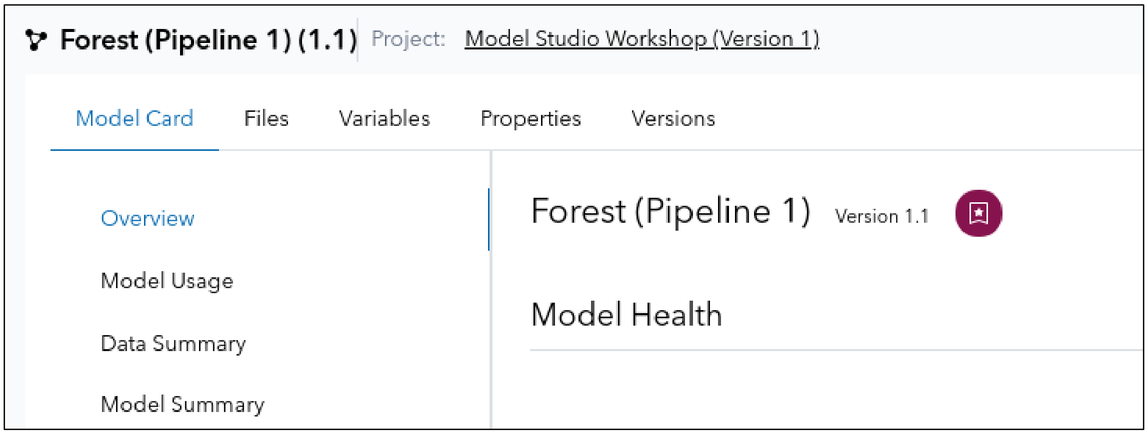
Studio. (The number of projects that exist on your computer may be different from what is shown below.)



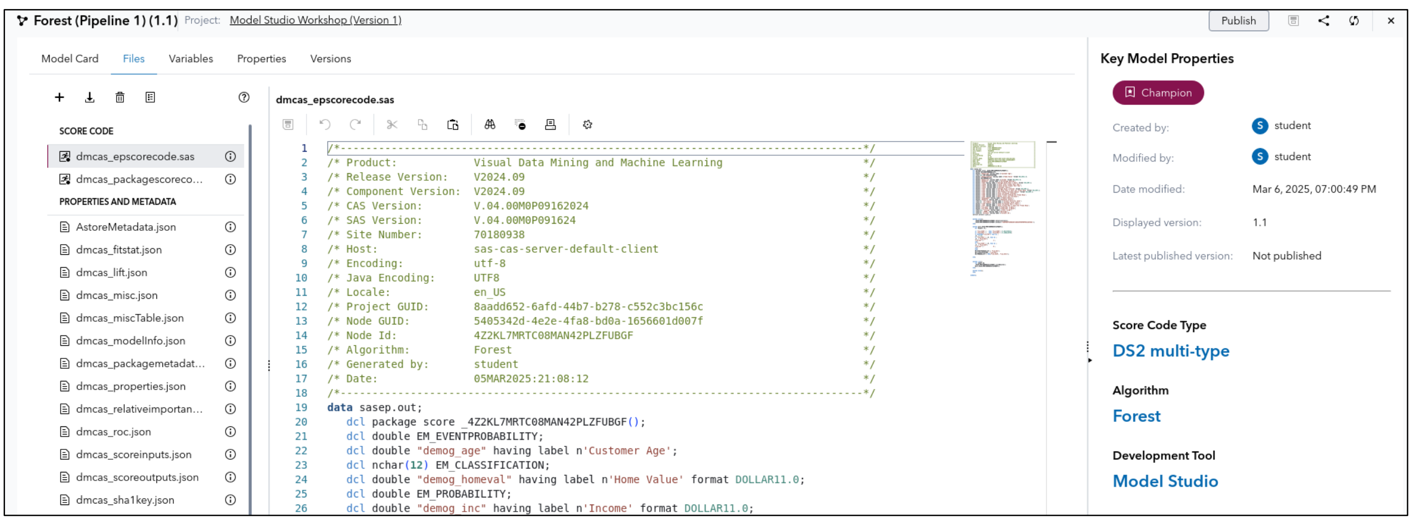
8. Select the **Model Studio Workshop** project to open it.

   You have registered the forward logistic regression model and the forest model into Model Manager from your Model Studio project.

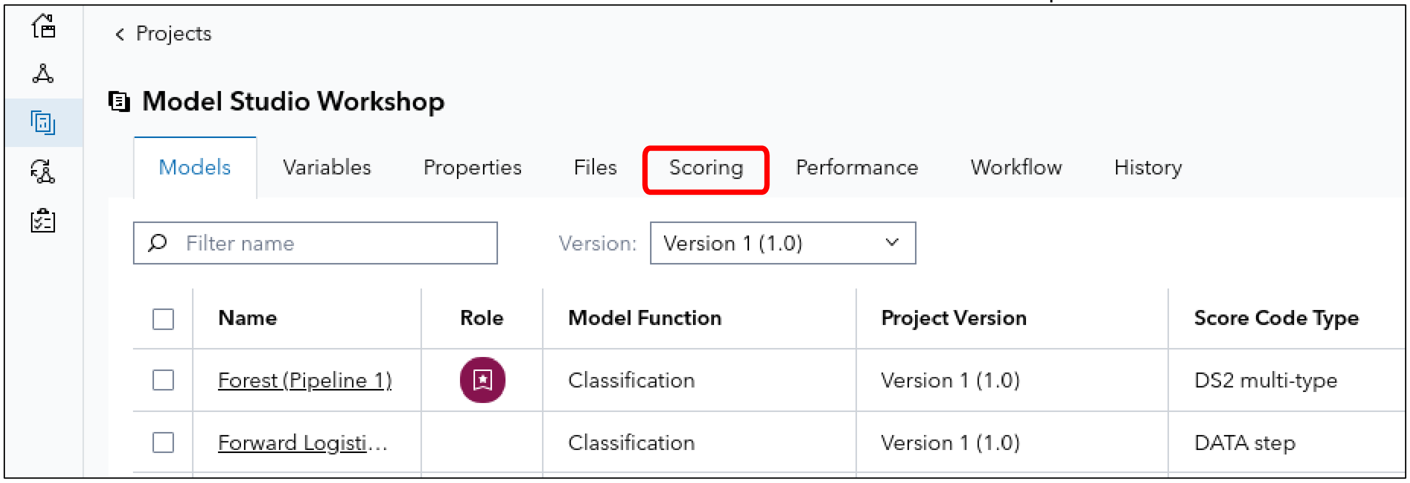
9. Select the **Forest (Pipeline 1)** model. The view changes to the Models view, as indicated in the left column, where artifacts about the forest model are stored. The Model Card for the forest model is shown by default. Click the **Files** tab.



10. The list of files on the left indicates artifacts stored for the model. The file **dmcas_epscorecode.sas** is the Base SAS score code (written in DS2) that can be used to deploy this model anywhere that SAS code can be executed. The file **dmcas_packagescorecode.sas** is score code packaged slightly differently and can be used for SAS Micro Analytic Service, for example, if the model were to score new cases in real time. The model score code can be published to a scoring destination from this view in a single click.



11. Return to the Projects view by clicking the **Projects** button in the left column. Click the **Scoring** tab.



12. From this page, a scoring test of the score code could be run as well as a publishing validation test, if the model score code was published to a scoring destination. We will not perform either in this workshop.

    For more information about SAS Model Manager, please see SAS Model Manager: User's Guide.

End of Demonstration