



SAS[®] Model Manager: User's Guide

2021.2.3 - 2021.2.4*

* This document might apply to additional versions of the software. Open this document in [SAS Help Center](#) and click on the version in the banner to see all available versions.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2022. *SAS® Model Manager: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Model Manager: User's Guide

Copyright © 2022, 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

June 2022

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.

v_011-P1:mdlmgrug

Contents

Chapter 1 / Introduction to SAS Model Manager	1
About Managing Models	1
Sign In to SAS Model Manager	3
SAS Model Manager Home	3
Share Links to SAS Model Manager Content	5
Using SAS Model Manager as a Progressive Web App	5
Manage Application Settings	6
High-Level Model Support Matrix for Primary Functions	7
Chapter 2 / Managing Data	13
About Managing Data	13
Create a CASLIB	14
Import Data	15
Chapter 3 / Managing Model Repositories	17
About Model Repositories	17
Create a New Repository	18
Rename a Repository	18
Delete a Repository	19
Chapter 4 / Working with Projects	21
About Projects	22
View Dashboard and Filter Projects	22
Create a New Project	24
Adding Models	29
Managing Variables	29
Modifying Project Properties	31
Managing Project Files	38
Managing Project Versions	40
Delete a Project	42
Rename a Project	42
Search for Projects	42
Chapter 5 / Working with Models	47
About Models	48
View Dashboard and Filter Models	48
Add a New Custom Model	50
Import Models	53
Export a Model	58
Move or Copy a Model	58
Delete a Model	59
Rename a Model	59
Manage Model Content and Versions	60
Search for Models	73
View Model Relationships and Related Objects	77
Chapter 6 / Comparing and Evaluating Models	79
About Comparing and Evaluating Models	80

Compare Models	80
Test Models	81
Validate Published Models	85
Set Champion and Challenger Models	88
Monitoring Performance	90
Retrain a Project from Model Studio	106
Chapter 7 / Publishing Models	109
About Publishing Models	109
Publishing Requirements and Restrictions	110
Publish a Project Champion Model	113
Publish Models from a Project Version	115
View Model Publish History	117
Deploy a Published Model to Azure Kubernetes Service (AKS)	118
Chapter 8 / Tracking Deployments	121
About Deployments	121
View and Filter Deployments	122
Search for Deployments	123
Chapter 9 / Using SAS Workflow with SAS Model Manager	127
About Using Workflows	127
Requirements	128
Prompts and Attributes	128
Start a New Workflow	130
Working with Tasks	131
Chapter 10 / Concepts	135
Concepts: Open-Source Models	135
Concepts: Performance Monitoring	139
Concepts: PMML Support	155

Introduction to SAS Model Manager

<i>About Managing Models</i>	1
<i>Sign In to SAS Model Manager</i>	3
<i>SAS Model Manager Home</i>	3
<i>Share Links to SAS Model Manager Content</i>	5
<i>Using SAS Model Manager as a Progressive Web App</i>	5
Benefits of Using a Progressive Web App	5
Installing SAS Model Manager as a Progressive Web App	6
<i>Manage Application Settings</i>	6
<i>High-Level Model Support Matrix for Primary Functions</i>	7
Open-Source Models	7
Predictive Model Markup (PMML) Models	8
SAS Viya Models	8
SAS 9.4 Models	9
System-Supplied Score Code Types	10

About Managing Models

SAS Model Manager helps streamline analytical model registration, deployment, monitoring, and management of SAS and open-source models, keeping them in one central place for data scientists and information technology (IT) teams. Using SAS Model Manager, you can store models in a common model repository, and organize them within projects and folders. You can also evaluate models for champion model selection, monitor performance of models, and publish models. All model development and model maintenance personnel, including data modelers, validation testers, scoring officers, and analysts, can use SAS Model Manager.

You can store and manage all types of models within the common model repository. This includes open-source models that are developed using Python and R programming languages, SAS models, as well as model content that is stored in PMML, ONNX, or RDS formats. In addition to models developed with custom Python code, you can use models that are developed with packages such as scikit-learn, TensorFlow, and XGBoost.

Data tables are an integral part of the modeling process. Data tables are used for scoring, publishing validation, and performance monitoring. In addition, data tables are used to record adherence to your modeling methodology for audit compliance. Performance data can be created from your operational data, provided that it has the required structure (for example, the data contains a target variable). For information about preparing and managing your data, see [Getting Started with SAS Data Preparation for SAS Viya](#) and “About Managing Data” on page 13.

Here are some of the tasks that you can accomplish with SAS Model Manager:

- Use a single interface to access all of your business modeling projects and models. All models are stored in a common model repository. Models can also be accessed in one place using the **Models** category. Your most recent model and project items are also available on the **Home** page, along with links to What’s New and how-to (video) articles in SAS Communities, and links to production documentation.
- [Import models](#) that you develop using a SAS application, such as Model Studio, SAS Visual Analytics, and SAS Studio, as well as SAS code, open-source programming language such as Python or R, and PMML models. You can also create a new model with the model’s files in a folder or project.
- [Compare models](#) to assess candidate models.
- [Manage data](#) that is used for scoring, validating published models, and performance monitoring.
- [Run scoring tests](#) to validate models.
- [Publish SAS models and open-source models](#) to supported publishing destinations. You can then score the published models within the publishing destinations or by using external applications or interfaces.
- [Score open-source models](#) within run-time containers.
- [Run monitoring jobs](#) to measure the performance of models on historical and current data.
- Use native Python code in [Jupyter notebooks](#) to import, export, and manage models.
- View various [model](#) and [project](#) metrics such as the number of published models within a dashboard.
- [Create custom workflow definitions](#) to meet your business requirements and to match your business processes. You can then start a workflow process to track the progress of your project.

Sign In to SAS Model Manager

Note: If you are already signed in to SAS Drive, you can access SAS Model Manager by clicking  and selecting **Manage Models**.

To sign in to SAS Model Manager:

- 1 In the address bar of your web browser, enter the URL for SAS Model Manager and press **Enter**. The **Sign In** page appears.

Note: If you are in a single sign-on environment, you are not prompted to sign in. Contact your system administrator if you need the URL for SAS Model Manager. The default URL is `https://host_name:port/SASModelManager`. The last part of the URL (the application name) is case sensitive. If the default port of 80 is used during configuration, you do not need to include the colon and port in the URL.

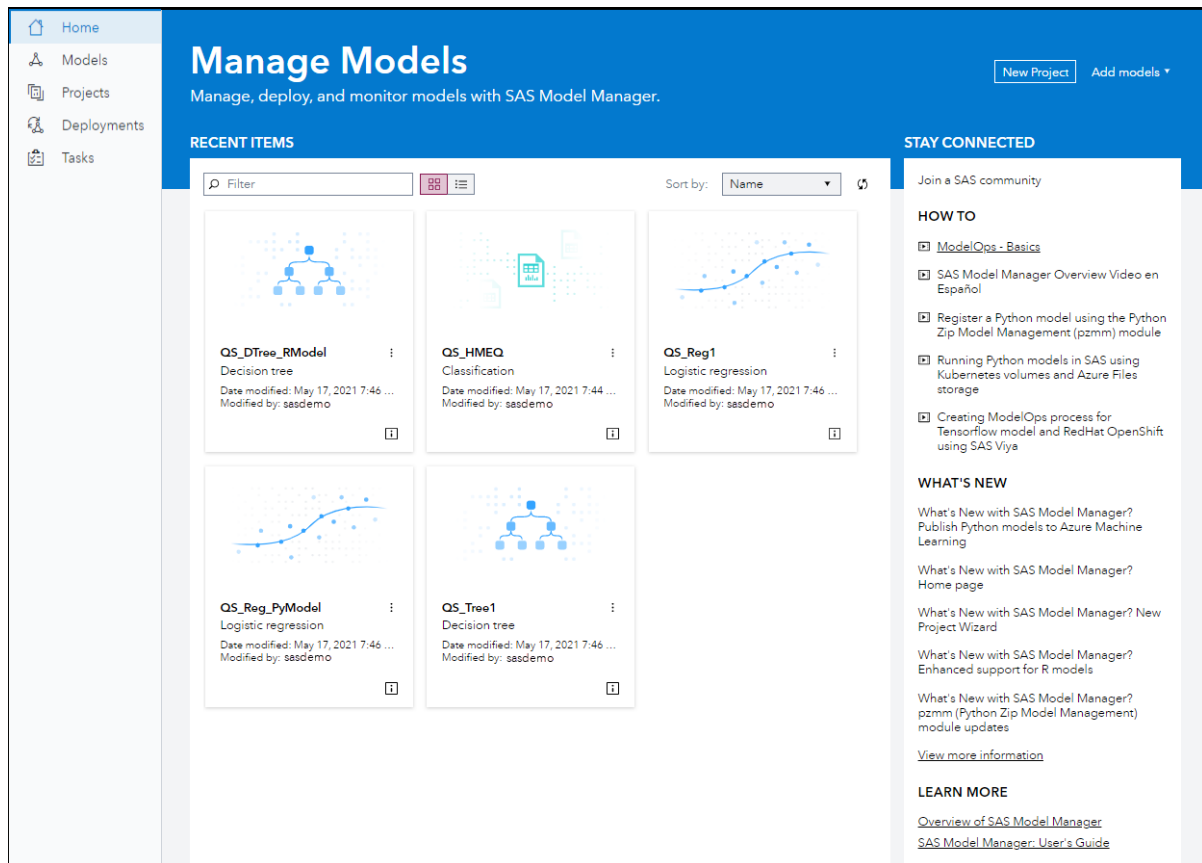
- 2 Enter a user ID and password.
- 3 Click **Sign In**.
- 4 (Optional) If this is your first time signing in to SAS Model Manager, the Welcome to SAS window appears. Here you can set up a profile, which enables you to customize some settings. You will then be taken to the [SAS Model Manager Home](#) page.

TIP You can [install SAS Model Manager as a Progressive Web App \(PWA\)](#).

SAS Model Manager Home

The SAS Model Manager Home page appears the first time you sign in to the web application. The category page that you are on when you sign out of SAS Model Manager is saved. The next time you sign in, the same category page is displayed.



Note: The left navigation bar is also expanded by default the first time you sign in. The next time you sign in, the navigation bar appears in the state that you left it when you signed out.




The Home page contains the following information:

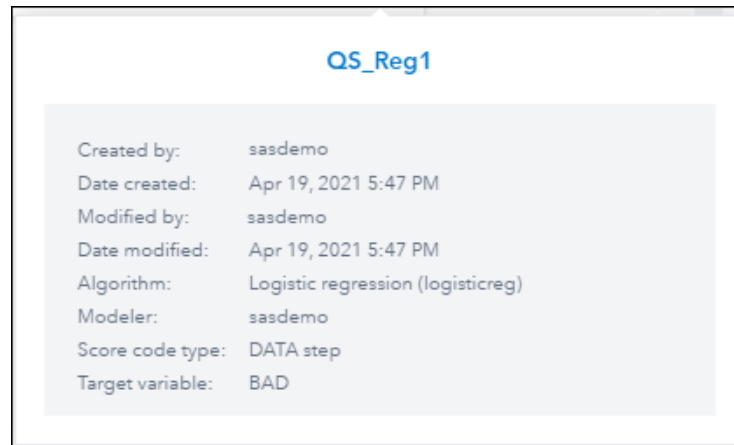
- the most recent items that you opened with the date that they were last modified and by whom
- links to the five most recent how-to videos and What's New articles on SAS Communities
- links to the What's New, overview, and user documentation for SAS Model Manager


Here are some of the actions that you can perform from Home:

- Click  to display the items in a list view.
- Click  to display the items in a tile view.

.....
Note: This is the default view.


- Sort the recent items by name or date modified.
- Click the name of an item to open it.
- Click  to view additional properties for an item.



- Click  : to open an item, remove a recent item, or copy a link to the item.
- [Create a new project](#).
- [Add models](#) to a folder.

Note: Only the 40 most recent items appear on the Home page.

Share Links to SAS Model Manager Content

When the  icon appears in the toolbar, you can perform the following tasks:

- Copy a link to a category view, object, tab, or page to the clipboard.
- Share a link to a category view, object, tab, or page to Microsoft Teams.

Using SAS Model Manager as a Progressive Web App

Benefits of Using a Progressive Web App

The benefits of using SAS Model Manager as a Progressive Web App (PWA) include the following:

- Application persistence – By default, your session will never time out, so you can restart your work more quickly. Your administrator can control the timing with a configuration property in SAS Environment Manager.
- Performance – When installed as a PWA, SAS Model Manager typically starts faster than when accessed in the browser.
- Desktop experience – Installing SAS Model Manager as a PWA confers all the benefits of a traditional installation. You can launch SAS Model Manager from the **Start** menu or Taskbar, and you do not need to sort through countless tabs to find the correct instance of SAS Model Manager.
- Renaming – You can rename each PWA instance to quickly access different environments, such as development, test, or production servers.

Installing SAS Model Manager as a Progressive Web App

SAS Model Manager must be deployed with TLS and HTTPS enabled.

To install SAS Model Manager as a PWA:

- 1 Open SAS Model Manager in a Chromium-based web browser and sign in.
- 2 Open the web browser's **More** menu and select **Install SAS**.

.....
Note: The **Install SAS** option might be located under another menu option. For example, within the Microsoft Edge browser it is located under **Apps**.
.....

SAS Model Manager is now installed as a desktop program named SAS. After one web application has been installed as a PWA, all other SAS applications are accessible in the PWA.

To uninstall the PWA for SAS Model Manager:

- 1 Open the PWA for SAS Model Manager.
- 2 In the PWA menu, select **Uninstall SAS**.

Manage Application Settings

Use the Settings window to edit user preferences or customize accessibility settings for all SAS web applications. You can also manage model repositories.

To access the Settings window, click your name in the application bar and select **Settings**.

For information about settings, see the following documentation:

- [“Settings” in SAS Viya Web Applications: General Usage Help](#)

- [Model Repositories](#)

High-Level Model Support Matrix for Primary Functions

Here is a summary of the primary functions that are supported by SAS Model Manager on SAS Viya. The type of model score code and the assignment of the “Score code” file role can determine which functions you can perform for the different types of models. Additional requirements and restrictions might apply, depending on the function that is being performed. For more information, click the links in the column titles to go to the function-specific section of the user's guide.

IMPORTANT The values in the **Model Score Code Type** column in each table below are associated with the **Score code type** model property. For more information, see [“Set Model General Properties” on page 64](#).

Open-Source Models

You can import models that were created using an open-source programming language such as Python and R and then perform tasks that are associated with the primary functions.

Table 1.1 Support for Primary Functions

Programming Language	Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess	Retrain
Python	Python	Yes	Yes	Yes	Yes	Yes	No
R	R	Yes	Yes	Yes	Yes	Yes	No

Here are some requirements and restrictions for open-source models:

- Models that have a score code type of Python or R can be scored if the score code is in the correct format. The score code must also be in the correct format for publishing and running performance. For more information, see [“Scoring Python Models” on page 136](#) and [“Scoring R Models” on page 138](#).
- Python and R open-source models can be published to CAS, Git, SAS Micro Analytic Service, Amazon Web Services (AWS), Azure, Azure Machine Learning, and Private Docker publishing destinations. For more information, see [“Publishing Requirements and Restrictions” on page 110](#).

Predictive Model Markup (PMML) Models

Predictive Model Markup Language (PMML) is an XML-based predictive model interchange format. PMML models that you create using PMML 4.2 support DATA step score code. When you are importing valid PMML models, the score code type model property is set to DATA step, instead of PMML. PMML models with a score code type of DATA step can be scored and published. For more information, see [“Import Models” on page 53](#).

Table 1.2 Support for Primary Functions

Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess	Retrain
DATA step	Yes	Yes	Yes	Yes	Yes	No
PMML	Yes	No	No	Yes	Yes	No

SAS Viya Models

You can build models with Model Studio, SAS Visual Analytics, or with SAS Viya modeling procedures within SAS Studio. These models can then be registered to the common model repository from those applications or imported into the common model repository using the SAS Model Manager web application.

Table 1.3 Support for Primary Functions

Product or Tool	Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess	Retrain
SAS Visual Data Mining and Machine Learning	DATA step	Yes	Yes	Yes	Yes	Yes	Yes
	DS2 multi-type	Yes	Yes	Yes	Yes	Yes	Yes
	Note: You can build SAS Visual Data Mining and Machine Learning models with Model Studio, SAS Visual Analytics, or with SAS Viya modeling procedures within SAS Studio. When SAS Viya models that contain one or more analytic store files are registered into the common model repository, their score code type is set to DS2 multi-type, instead of Analytic store.						
SAS Visual Statistics	DATA step	Yes	Yes	Yes	Yes	Yes	No

Product or Tool	Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess	Retrain
	Note: You can build SAS Visual Statistics models with SAS Visual Analytics, or with SAS Viya modeling procedures within SAS Studio.						
SAS Visual Text Analytics	DS2 multi-type	Yes	Yes	Yes	No	Yes	No
	Note: You can build SAS Visual Text Analytics models with Model Studio or with SAS Viya modeling procedures within SAS Studio. Text categories, Text concepts, Text sentiment, and Text topics models have a score code type of DS2 multi-type and can be published to CAS, Git, Hadoop, and SAS Micro Analytic Service destinations.						

Here are some restrictions for SAS Viya models:

- Only SAS Visual Data Mining and Machine Learning models that are built and registered using Model Studio can be retrained using SAS Model Manager. For more information, see [“Retrain a Project from Model Studio” on page 106](#).
- Scoring of decisions that contain a model with a score code type of SAS program is not supported by SAS Intelligent Decisioning. For more information about scoring decisions, see [SAS Intelligent Decisioning: User’s Guide](#).

For more information about SAS Viya models, see the following documentation:

- [SAS Visual Data Mining and Machine Learning: User’s Guide](#)
- [SAS Visual Text Analytics: User’s Guide](#)
- [SAS Visual Analytics: Working with SAS Visual Data Mining and Machine Learning](#)
- [SAS Visual Analytics: Working with SAS Visual Statistics](#)
- [“SAS Visual Data Mining and Machine Learning” in SAS Procedures by Name and Product](#)
- [Getting Started with SAS Visual Data Mining and Machine Learning](#)

SAS 9.4 Models

Models that you create with SAS 9.4 can also be imported into SAS Model Manager on SAS Viya. You can then perform tasks associated with the primary functions.

.....
Note:

- When analytic store models are imported using a SASAST file or are registered into the common model repository from Model Studio, SAS Visual Analytics, or SAS Studio, their score code type is set to DS2 multi-type, instead of Analytic store.
- PMML models that you create with PMML 4.2 support DATA step score code. When importing valid PMML models, set the score code type model property to

DATA step, instead of PMML. PMML models with a score code type of DATA step can be scored and published. See [“Import Models” on page 53](#).

Table 1.4 Support for Primary Functions

Product or Tool	Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess	Retrain
Base SAS or other code editor	SAS program	Yes	Yes	No	Yes	Yes	No
SAS Enterprise Miner	DATA step	Yes	Yes	Yes	Yes	Yes	No
	PMML	Yes	No	No	No	Yes	No
	Analytic store	Yes	Yes	Yes	Yes	Yes	No
Note: Applies only to SAS analytic store (SASAST) files. SAS package (SPK) files are not supported.							
SAS HPFOREST and HPSVM procedures	Analytic store	Yes	Yes	Yes	Yes	Yes	No
SAS/STAT linear model procedures	DATA step	Yes	Yes	Yes	Yes	Yes	No

System-Supplied Score Code Types

Note: A model with the score code type of DS2 multi-type contains code files for a DS2 embedded process and a DS2 package. It can also contain one or more analytic stores. When you import or register analytic store models into the common model repository, the DS2 score code is generated and the score code type is set to DS2 multi-type, instead of Analytic store. When adding a model using the new custom model feature, if an analytic store file is included with the model files but the DS2 score code is not included, the model cannot be scored or published.

Table 1.5 Support for Primary Functions

Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess
Analytic store	Yes	Yes	Yes	Yes	Yes

Model Score Code Type	Import	Score	Publish	Monitor Performance	Model Compare and Assess
DATA step	Yes	Yes	Yes	Yes	Yes
DS2 embedded process	Yes	Yes	Yes	Yes	Yes
DS2 multi-type	Yes	Yes	Yes	Yes	Yes
DS2 package	Yes	Yes	Yes	Yes	Yes
PMML ¹	Yes	No	No	Yes	Yes
Python	Yes	Yes ³	Yes	Yes	Yes
R	Yes	Yes ³	Yes	Yes	Yes
SAS program	Yes	Yes ²	No	Yes	Yes
Additional Score Code Types					
Note: Monitoring performance of models with the following score code types is supported only when the performance definition includes input tables that contain scored data. For more information, see “Monitoring Performance” on page 90 .					
C	Yes	No	No	Yes	Yes
CAS language	Yes	No	No	Yes	Yes
Java	Yes	No	No	Yes	Yes
Lua	Yes	No	No	Yes	Yes
MATLAB	Yes	No	No	Yes	Yes

- 1 PMML models that you create with PMML 4.2 support DATA step score code. When you are importing valid PMML models, set the score code type model property to DATA step, instead of to PMML. PMML models with a score code type of DATA step can be scored and published. See [“Import Models” on page 53](#).
- 2 Scoring and publishing of decisions that contain a model with a score code type of SAS program is not supported by SAS Intelligent Decisioning. For more information about scoring or publishing decisions, see [SAS Intelligent Decisioning: User’s Guide](#).
- 3 Models that have a score code type of Python or R can be scored, if the score code is in the correct format. The score code must also be in the correct format for publishing and running performance. See [“Scoring Python Models” on page 136](#) and [“Scoring R Models” on page 138](#).

Managing Data

<i>About Managing Data</i>	13
<i>Create a CASLIB</i>	14
<i>Import Data</i>	15

About Managing Data



The Manage Data window is accessible from both the Models and Projects category views in SAS Model Manager. You can manage existing data, create a new caslib, and import data from a CSV data file or a SAS data set.

Note: You can access the Manage Data window only after a project or model has been added to the common model repository.

CAUTION


Do not add or include files that contain executable or dynamic content that is not valid or approved for your environment. Examples of executable content are batch (.bat) files or executable (.exe) files. Examples of dynamic content are HTML, Microsoft Office documents, JavaScript, dynamic PDF forms, and other forms of executable content. This type of content, when downloaded, could execute in your local environment and provide a way for malicious content to enter it. The Files service is used when adding data files, model files, and project files. As general guidance, SAS recommends that users be vigilant and cautious with any content that is added using the Files service and accessed within SAS Model Manager.

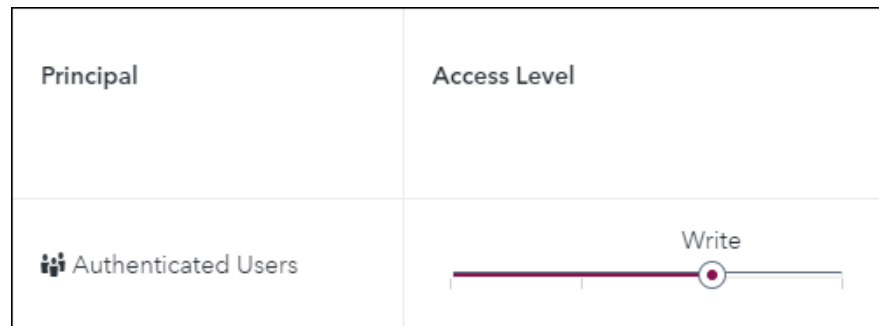
Create a CASLIB

- 1 On the toolbar above the table, click  and select **Manage data**. The Manage Data window appears.
- 2 Click the **Data Sources** tab.
- 3 Click  on the **Data Sources** tab. The Connections Settings window appears.
- 4 Enter a name for the caslib in the **Name** box.
- 5 Accept the default CAS server (cas-shared-default) or select another CAS server from the **Server** drop-down list.
- 6 Select the **File system** option from the **Type** drop-down list.
- 7 Select the **Path** option from the **Source type** drop-down list.
- 8 Select the **Persist this connection beyond the current session** check box to add a global caslib for this connection. For more information about this option, see [“Caslibs on the Data Sources Tab and Import Tab” in SAS Data Explorer: User’s Guide](#).
- 9 Specify the path connection information on the **Settings** tab.

Note: This is the physical path to the remote file system. For a new global caslib, you can specify a directory that does not exist on the target file system, and that directory will be created. However, you must have Read and Write access to the specified location on the file system in order to create a caslib. By default, the `/cas/data` directory path is mounted to store CAS data during deployment. It is recommended that you specify a location within this default directory.

- 10 Click **Test Connection**.
- 11 Click **Save**.
- 12 Grant users permission to write to the library.
 - a On the **Data Sources** tab, select the server where you created the CAS library.
 - b Enter the name of the CAS library in the **Filter** box or scroll to find it.
 - c Right-click the library and select **Edit authorization**. The Edit Authorization window appears.
 - d Move the slider for **Authenticated Users** to include **Write** permission.

Note: You can also click  to grant permissions to other users or groups in this window. For more information, see [“Examples: Manage Access to a Caslib” in SAS Viya: CAS Authorization Window](#).




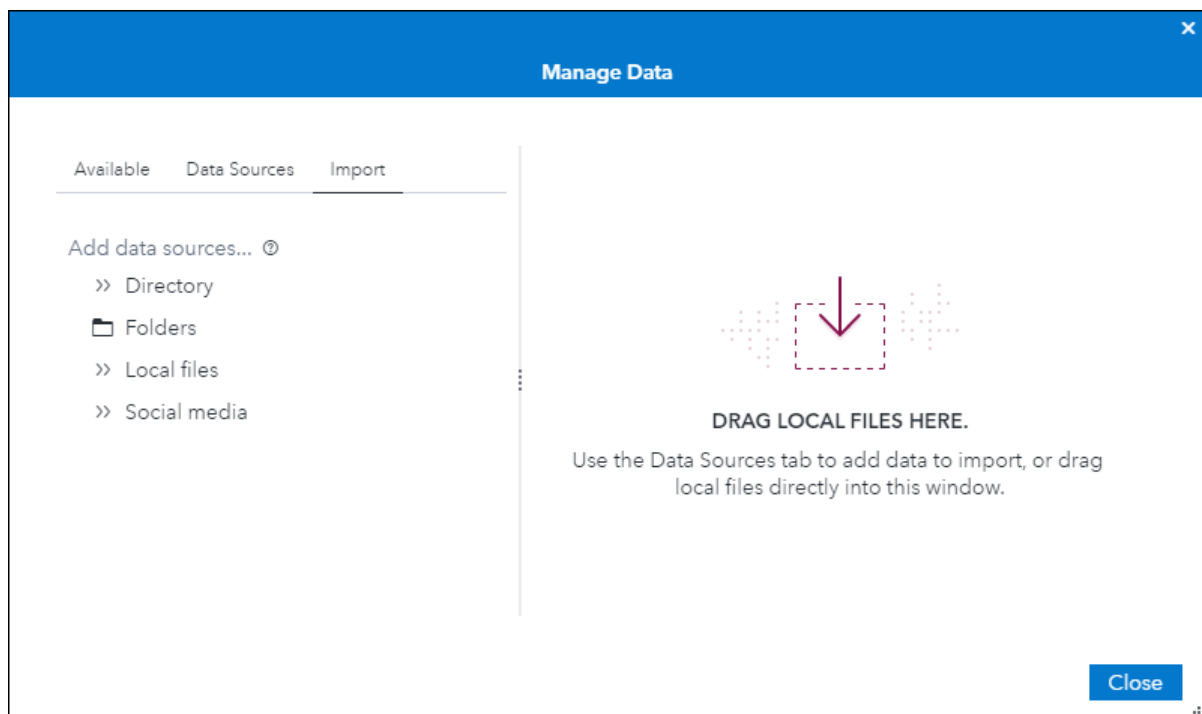
- e Click **Save**.

For more information, see “Making Data Available to CAS” in *SAS Data Explorer: User’s Guide*.

Import Data

Note: Before importing data tables to use for performance monitoring, ensure that the table names meet the requirements. For more information, see “Naming Requirements for Input Data Tables” on page 92.

- 1 On the toolbar above the table, click  and select **Manage data**. The Manage Data window appears.
- 2 Click the **Import** tab.



- 3 Drag your data files into the right pane.

TIP You can also click **Local files** and select **Local file** to navigate to the file location.

- 4 Make sure that the target location is set to the correct library location. The default library location is `cas-shared-default/Public`.

Note: You can accept the default format or select a target format that is appropriate for the source data. Not all source data should be saved to some formats, even if they are supported by the target caslib. For example, the CSV format might be supported, but you cannot save a binary image file to a CSV target file and get useful data. Also, it is recommended to use the SASHDAT format for large data files that you want to use for performing analytic calculations.

- 5 Click **Import All**.
- 6 Click **Close**.

For more information, see [“Working with Data in CAS”](#) in *SAS Data Explorer: User’s Guide*.

Managing Model Repositories

<i>About Model Repositories</i>	17
<i>Create a New Repository</i>	18
<i>Rename a Repository</i>	18
<i>Delete a Repository</i>	19

About Model Repositories


SAS Model Manager provides a secure common model repository to store all types of models that are versioned. The common model repository enables you to search, query, sort, and filter models by attributes that are associated with models across all accessible organizational folders and projects. You can use model repositories to separate your project and model content, as well as to set permissions for objects within a repository. Some examples are having different repositories for test and production environments, or for different organizations. Model repositories are managed within the Settings window of the SAS Model Manager web application. You can add, delete, and rename a repository. The default repository can be renamed, but it cannot be deleted.

Note: Only SAS administrators and other authorized users can create, update, or delete repositories. In addition, Authenticated Users cannot initially access new custom repositories. A SAS administrator must grant access for a user or group to a new custom repository. For more information, see [“Managing Content” in SAS Model Manager: Administrator’s Guide](#).

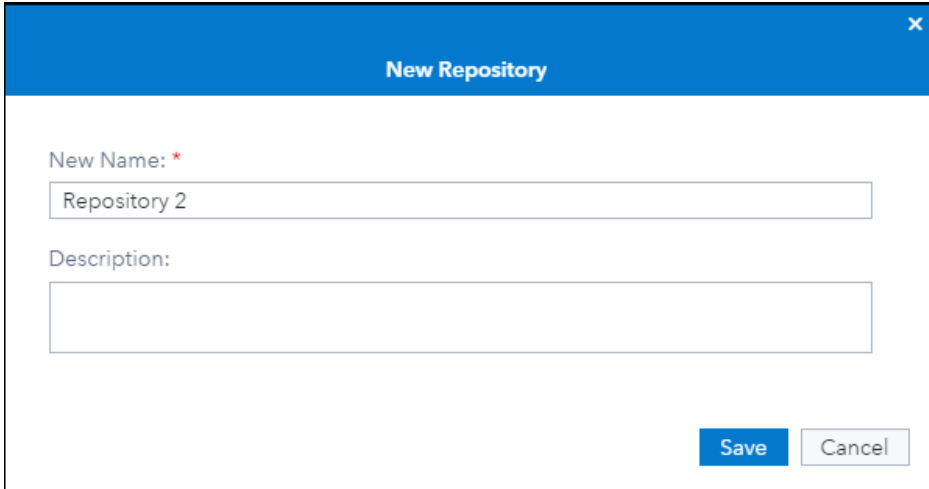
To access the settings for repositories, click the user button in the application bar and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repositories**.

Create a New Repository

Note: By default, only SAS administrators can create new repositories.

- 1 Click the user button in the application bar and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repositories**.
- 2 Click .
- 3 Enter a name for the repository.
- 4 (Optional) Enter a description for the repository.


Note: After you save the new repository, the description cannot be edited.



- 5 Click **Save**.

Rename a Repository

Note: By default, only SAS administrators can rename repositories.

- 1 Click your name in the application bar and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repositories**.
- 2 Select a repository, click , and select **Rename**.
- 3 Enter a new name for the repository.

- 4 Click **Rename**.

Delete a Repository

.....
Note: By default, only SAS administrators can delete repositories. A repository cannot be deleted if it contains model or project content.
.....

- 1 Click your name in the application bar and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repositories**.
- 2 Select a repository and click .
- 3 In the confirmation message, click **Delete**.

Working with Projects

About Projects	22
View Dashboard and Filter Projects	22
View Projects Dashboard	22
Filter the List of Projects Using the Dashboard Charts	23
Create a New Project	24
Adding Models	29
Managing Variables	29
Add Variables from a Data Table	29
Add Custom Variables	30
Edit Variables	30
Delete Variables	31
Modifying Project Properties	31
Set Project Properties	31
Add KPI Rules	36
Add User-Defined Properties	37
Add Keywords	37
Managing Project Files	38
Add Files	39
Delete Project Files	39
Download Project Files	39
Run a Project Code File	39
View File Properties	40
Managing Project Versions	40
Create a New Project Version	41
Manage Project Versions	41
Delete a Project	42
Rename a Project	42
Search for Projects	42
Advanced Search for Projects	43

About Projects

Often numerous statisticians, modelers, and data scientists, working in their tool of choice, are asked to solve a business problem by creating an analytical model. The Projects category view enables users to view models grouped by business problem. It also enables a user to perform an apples-to-apples comparison of models in order to decide which model is the best to use as the champion.

You create projects within folders. The models within a project are associated with a project version. A project version enables you to group models based on business requirements. When you group your models by a specific time period, you can see whether a particular business problem has changed. You can also determine how you addressed those changes and use that insight to govern how you will address these changes in the future.

A project consists of the models, variables, tests, and other resources that you use to determine a champion model. For example, a banking project might include models, data, and tests that are used to determine the champion model for a home equity scoring application. The home equity scoring application predicts whether a bank customer is an acceptable risk for granting a home equity loan.

Note: The **History** tab of a project shows events for the primary actions, and the date on which the project was modified and by whom.

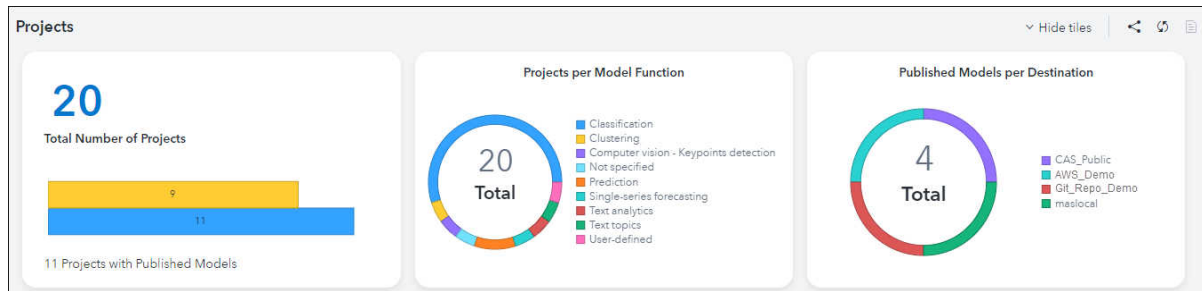
View Dashboard and Filter Projects

View Projects Dashboard

When you first view the Projects category, the dashboard appears at the top of the application window. The dashboard contains a graphical representation of the following metrics:

- the total number of projects
- number of projects with published models
- number of projects per model function
- number of published models per destination

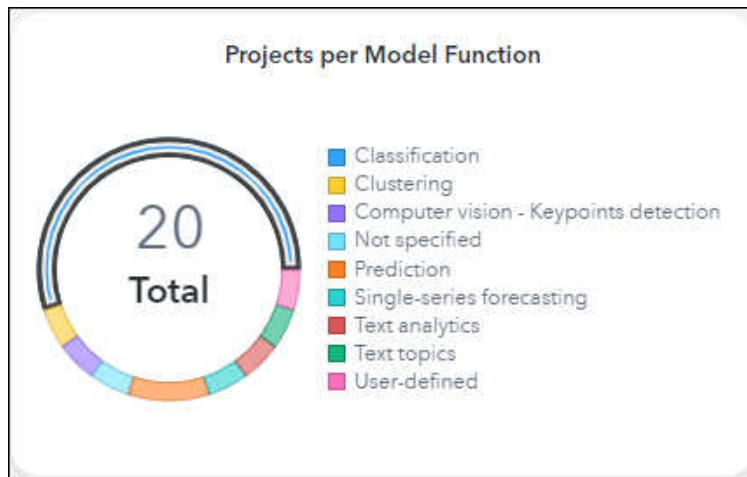
You can click **Hide tiles** or **Show tiles** on the toolbar.



Filter the List of Projects Using the Dashboard Charts

You can click various sections of the different dashboard charts to filter the list of projects. You can apply filters to each of the charts to create a compound query.


Here is an example of filtering the list of projects to display only projects that have a **Classification** model function:

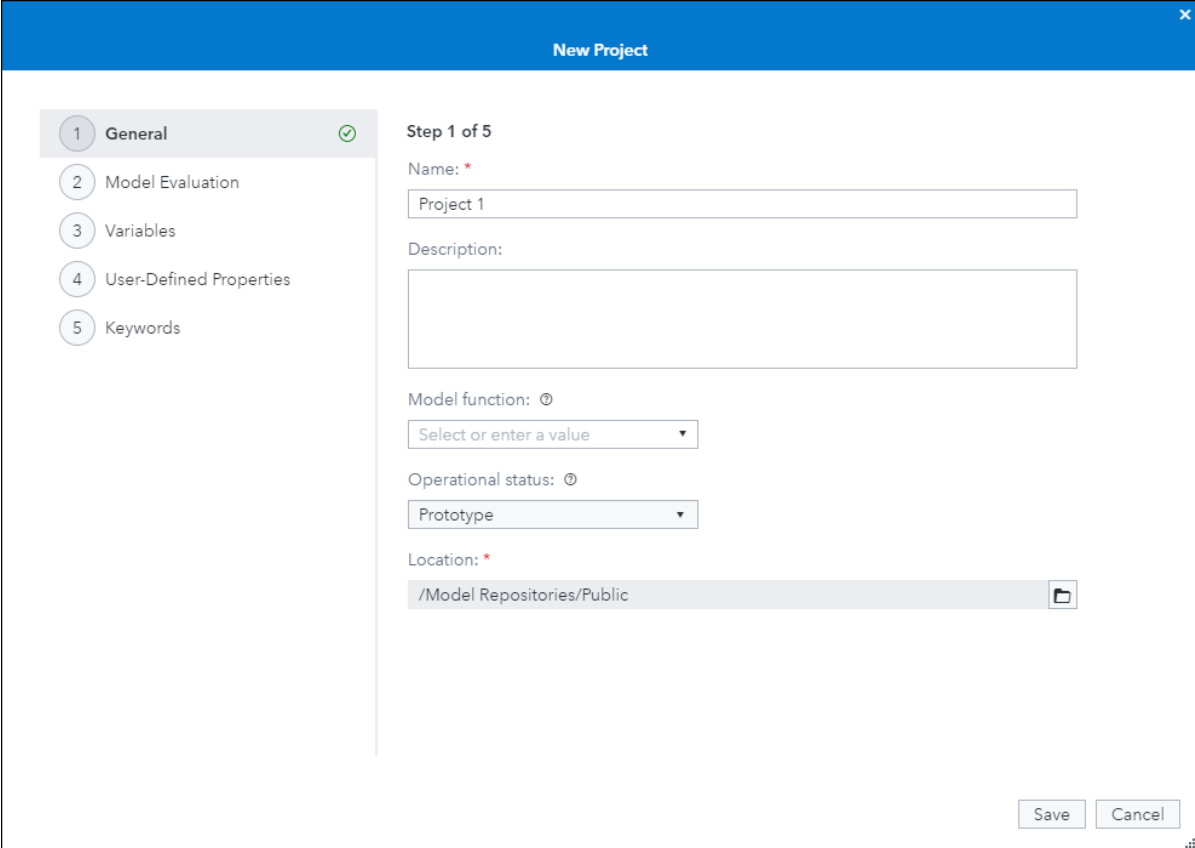


You can also use the Advanced Search for Projects window to specify additional information in order to further filter the list. The filters that you applied using the dashboard charts are reflected in the Advanced Search for Projects window. For more information, see [“Advanced Search for Projects” on page 43](#).

TIP To remove an individual filter from a chart, you can click the empty area beside the chart or you can click the **×** icon within the tokens that are located above the list of projects. You can also remove all filters by clicking the **×** icon, which is located in the right side of the toolbar.

Create a New Project

- 1 Click  to navigate to the Projects category view.
- 2 Click **New Project**. The New Project wizard appears.



The screenshot shows the 'New Project' wizard interface. The title bar reads 'New Project'. The left sidebar lists five steps: 1 General (selected), 2 Model Evaluation, 3 Variables, 4 User-Defined Properties, and 5 Keywords. The main content area is titled 'Step 1 of 5' and contains the following fields:

- Name: ***: A text input field containing 'Project 1'.
- Description:**: A large text area for entering a description.
- Model function: ⓘ**: A dropdown menu with the text 'Select or enter a value'.
- Operational status: ⓘ**: A dropdown menu with the text 'Prototype'.
- Location: ***: A text input field containing '/Model Repositories/Public' and a folder icon.

At the bottom right of the wizard, there are 'Save' and 'Cancel' buttons.

- 3 On the **General** page, enter a name for the project.


TIP You can save a project at any time as long as all of the required fields have been specified.

- 4 (Optional) Enter a description for the project.
- 5 (Optional) Select a model function from the list or enter your own value. The model function indicates the type of output that your predictive model project generates.

IMPORTANT To [monitor the performance](#) of the models within a project, it is recommended to select **Classification** or **Prediction** when creating a

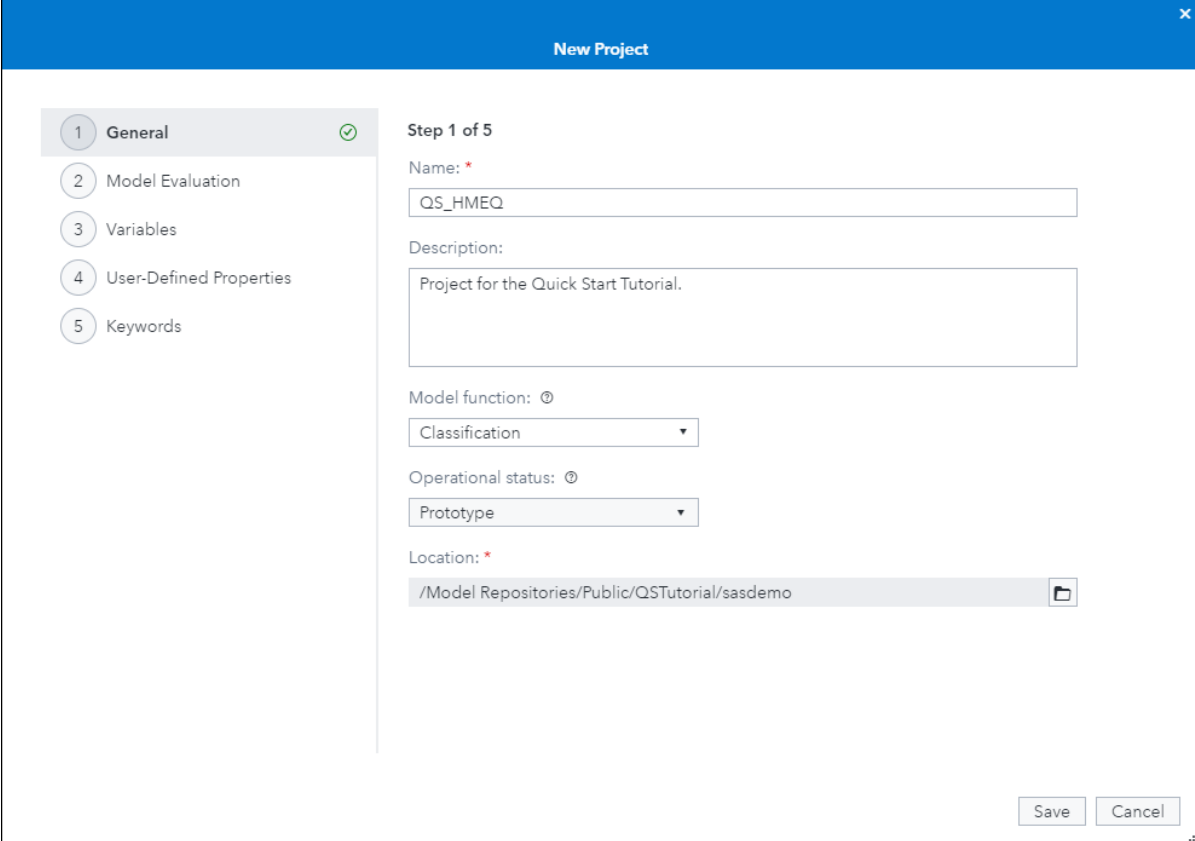
project. Only projects with those model functions can be monitored for performance.

- 6 Accept the default value of **Prototype** for the operational status.
- 7 Accept the default location or select a new location.

To select a new location, click , select the desired repository or folder, and then click **OK**.

Note: In the Choose a Location window, you can create a folder within a repository folder to store projects and models for your organization. Repository folders must be created using the Settings window. For more information, see [“About Model Repositories” on page 17](#).

It is recommended that the **DMRepository** and **VTARespository** repository folders should be reserved for projects and models that are registered from Model Studio. Also, the **VARespository** repository folder should be reserved for models that are registered from SAS Visual Analytics. For more information, see [“Register Models” in SAS Visual Data Mining and Machine Learning: User’s Guide](#).



- 8 (Optional) Click the **Model Evaluation** page and specify values for each setting:
 - a Select a default training table. Click , enter the name of the table in the filter box, and select it. Click **OK**.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see “Working with SAS Data Explorer” in *SAS Data Explorer: User’s Guide*.

- b** Specify values for the target variable, target level, and target event value to use when running performance definition.

Note: If you do not specify values for these settings when creating a project, you must at least specify a value for the target level before running a performance definition. For a project with a classification model function, you must also specify the target event value.

- c** Specify values for the model assessment indicator, alert condition, and alert threshold.

Note: You cannot currently select an assessment indicator when creating a project under these conditions: a model function is not specified, or the model function is Analytical, Text analytics, or Transformation.

The screenshot shows the 'New Project' dialog box, Step 2 of 5: Model Evaluation. The left sidebar shows the navigation menu with 'Model Evaluation' selected. The main area contains the following settings:

- Default training table:** cas-shared-default/Public/HMEQ_TRAIN (with a folder icon and a 'Clear' button)
- Target variable:** BAD (dropdown menu)
- Target level:** Binary (dropdown menu)
- Target event value:** 1 (text input field)
- MODEL ASSESSMENT CRITERIA** (Section Header)
- Assessment indicator:** Misclassification (dropdown menu)
- Alert condition:** Greater than (dropdown menu)
- Alert threshold:** 0.2 (text input field)

At the bottom right, there are 'Save' and 'Cancel' buttons.

- 9** (Optional) Click the **Variables** page and select input and output variables from the default training table or another input table. You can also add custom variables.

Note: If you do not specify project variables on this page, you are prompted to set them when you are setting a model as the project champion or as a challenger.

- a** To select input variables, click **Browse** and select one or more variables, and then click **OK**.

If you did not specify a default training table, here are the other ways that you can select variables.

- Select an input table:

1 Click .

2 Select a data table and click **OK**.

- Add one or more custom variables:

1 Click **Add Custom Variable**.

2 Enter a name for the variable.

3 Select a data type.

4 Expand the **Optional** section to specify a length, measurement, and description for the variable.

5 Click **Add**.

6 Repeat steps 1 through 5 for each variable that you want to add.

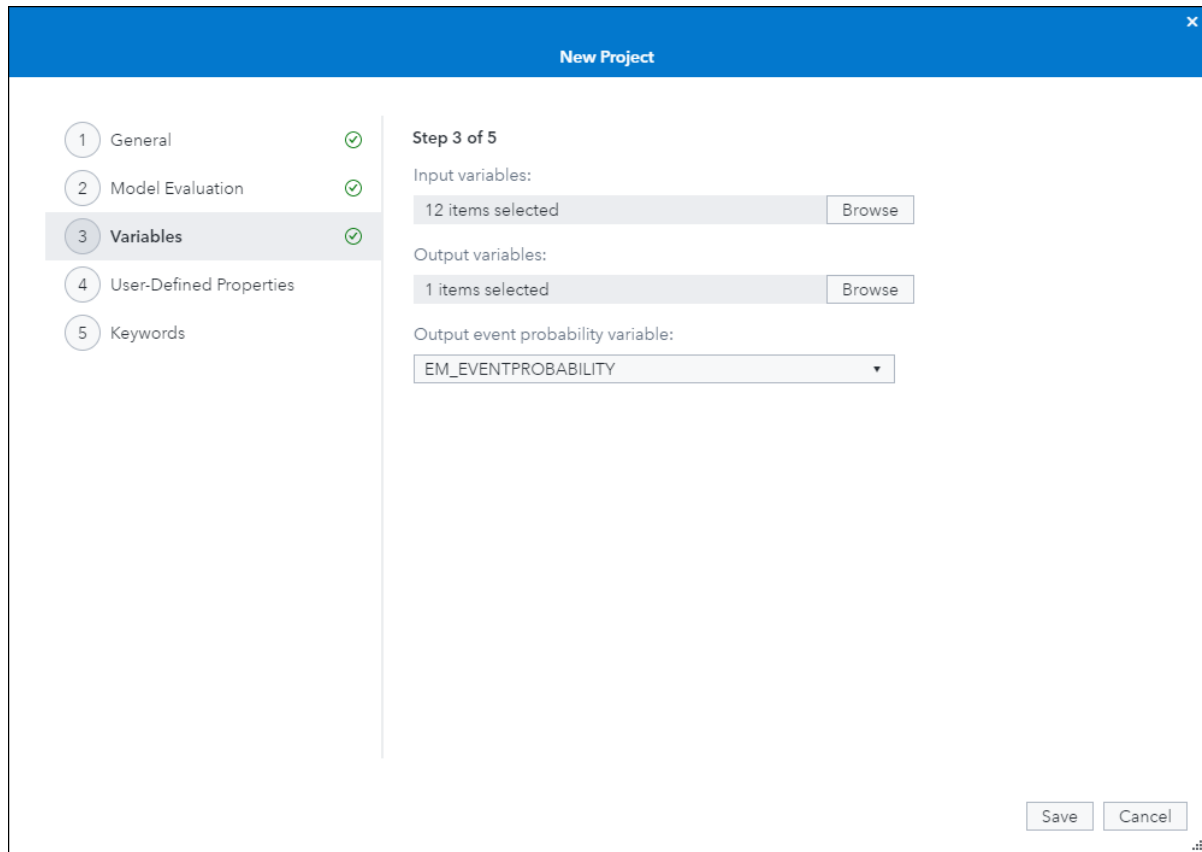
7 Click **OK**.

- b** To select output variables, **Browse** and select one or more variables, and then click **OK**.

Note: You have the same choices for selecting output variables that you had for input variables. See the previous step.

- c** Select a value for each of the project-specific output variables. For example, if you selected **Classification** as the model function on the **General** page, select a value for the **Output event probability variable**.

Note: The model function that you specified determines the project-specific output variables that are displayed. If you do not specify values for the output variables when creating a project, you must specify them before running a performance definition.



10 (Optional) Click the **User-Defined Properties** page.

TIP You can also add user-defined properties after you create a project. For more information, see [“Add User-Defined Properties” on page 37](#).

- a** Click **Add Property**. The Add Property window appears.
 - b** Enter a name for the property.
 - c** Select a data type for the property.
 - d** Enter a value for the property.
 - e** (Optional) Select the **Add another property** check box to add another variable without closing the window.
 - f** Click **Add** to add the property to the list.
 - g** Repeat steps b through f for each property that you want to add.
- 11** (Optional) Click the **Keywords** page.

Enter a value and select an existing keyword, or press Enter to add a new keyword. Repeat this step to add more keywords.

Note: The keyword name can contain only alphanumeric characters, double-byte characters, an underscore (_), a hyphen (-), and a period (.). Spaces are not allowed.

TIP You can also add keywords after you create a project. For more information, see [“Add Keywords” on page 37](#).

12 Click **Save**.

Adding Models

After you create a project, you add models to a project version on the **Models** tab. A project can contain multiple versions. You can also copy a model from a folder or another project version. You can view and compare models in all versions or in one selected version on the **Models** tab. After model evaluation, you set one of the candidate models as the champion model and can also set one or more models as challengers.

Here are some of the tasks that you can perform on the **Models** tab of a project:

- [add a new custom model](#)
- [import models](#)
- [copy a model](#)
- [compare models](#)

Managing Variables


Input variables and output variables can be added to both project and model objects on the **Variables** tab of an object. The same variable name cannot be used for both an input and output variable.

Note: When you add, edit, or delete project variables, you might receive a warning message indicating that the model role will be cleared. This happens when the project champion model and one or more challenger models have been set.

Add Variables from a Data Table

- 1 Click the **Variables** tab.
- 2 Click **Add variable** and select **Data table**. The Choose Data window appears.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see “Working with SAS Data Explorer” in *SAS Data Explorer: User’s Guide*.

- 3 Select the data table that you want to import the variables from and click **OK**. The Choose Variables window appears.
- 4 Select the variables that you want to add and click **➔** to add the variables to the input variables or output variables list. You can also click **➔➔** to add all of the variables from the available items list.
- 5 Click **OK**.
- 6 Click .

Add Custom Variables

- 1 Click the **Variables** tab.
 - 2 Click **Add variable** and select **Custom variable**. The Add Custom Variables window appears.
-
- Note:** If there are no existing variables, click **Add Custom Variables**.
-
- 3 Enter a name for the variable.
 - 4 Select a data type and variable type.
 - 5 Expand the **Optional** section to specify a length, measurement, and description for the variable.
 - 6 Click **Add**.
 - 7 Repeat steps 3 through 6 for each variable that you want to add.
 - 8 Click **OK**.



Edit Variables

To edit variables:

- 1 Click the **Variables** tab.
- 2 Click on the name of the variable that you want to edit. The Edit Variable window appears.
- 3 Edit the properties as needed and click **OK**.

- 4 Click .

Delete Variables

- 1 Click the **Variables** tab.
- 2 Select the check box for the variables that you want to delete, click , and then select **Delete**.
- 3 Click .

Modifying Project Properties

Project properties contain the project metadata. Project metadata includes information such as the name of the project, the type of project (model function), the project owner, the project identifier, the project location, and the names of the tables and variables that are used by project processes. The project properties are organized into these types: General, Model Evaluation, Keywords, and User-Defined.

Set Project Properties

The **Properties** tab of a project contains both system-defined properties that you cannot modify, and project specific properties that can be modified, such as the description of the project. None of the project properties are required, except for the name and location.

To set the project properties, click the **Properties** tab, modify the property values, and then click .

Table 4.1 List of Project Properties

Property	Description
Name	Specifies the name of the project. A project can be renamed only from the Projects category view.
Description	Specifies the description of the project.
Model function	Specifies the type of output that your predictive model project generates. Ensure that the types of models that you are going to use in the project fit within the selected model function type. For

Property	Description
	more information, see Table 4.2 on page 35 .
Operational status	<p>Specifies the current state of the project and its models:</p> <p>Prototype indicates that the project has started, but that the project's models have not yet been deployed. This is the default status when you create a new project.</p> <p>Note: This status replaces the previous status value of "Under development".</p> <p>Deployed indicates that at least one model within the current project has been successfully published to a destination.</p> <p>In production indicates that the project contains models that have been successfully deployed and are being used in a production environment. This status must be set manually.</p> <p>Note: The status replaces the previous status value of "Active".</p> <p>Deactivated indicates that the project contains one or more models that were previously in a production environment and have been temporarily or permanently suspended.</p> <p>Note: This status replaces the previous status values of "Inactive" and "Retired".</p>
Location	Specifies the location of the project in the common model repository.
Champion version	Specifies the project version that contains the champion model.
Champion model name	Specifies the name of the model that is set as the project champion.
UUID	Specifies the universally unique identifier for a project object.
External URL	Specifies a user-defined URL to a project object in another application or to documentation related to the project.

Property	Description
External project ID	Specifies the project ID for a project that was registered from an external application, such as Model Studio.
Model Evaluation	This section contains project properties that are used for comparing, assessing, and evaluating the health of models.
Default training table	Specifies the default training table, which can be used to test models on-demand or to validate models after a user publishes them to a publishing destination. This property is optional.
Target variable	Specifies the name of the target variable that was used to train the model.
Target level	Specifies the target level of binary, nominal, ordinal, or interval.
Target event value	Specifies the target variable value that defines the desired target variable event.
Target values	For class, nominal, ordinal, or interval targets, the set of possible outcome classes, separated by commas. For example, binary class target values might be <code>1, 0</code> or <code>Yes, No</code> . Nominal class target values might be <code>Low, Medium, High</code> . These values are for information only.
Output category variable	Specifies the output variable name, when the Model function property is set to Text categories .
Output event probability variable	Specifies the output event probability variable name, when the Model function property is set to Classification, Analytical, Computer vision - Image classification , or Transformation . Note: This property is also displayed for user-defined model functions.
Output prediction variable	The output prediction variable name, when the Model function property is set to Prediction, Analytical, Computer vision - Image segmentation, Computer vision - Keypoints detection, Computer vision - Object

Property	Description
	<p>detection, Single-series forecasting, or Transformation.</p> <p>Note: This property is also displayed for user-defined model functions.</p>
Output segmentation variable	<p>The output segmentation variable name, when the Model function property is set to Clustering, Analytical, Segmentation or Transformation.</p> <p>Note: This property is also displayed for user-defined model functions.</p>
Output sentiment variable	<p>Specifies the output variable name, when the Model function property is set to Text sentiment.</p>
Output topic variable	<p>Specifies the output variable name, when the Model function property is set to Text topics.</p>
Model Assessment Criteria	<p>This section contains the properties that are used for comparing and assessing models.</p>
Assessment indicator	<p>Specifies performance indicators that are used to assess a model. The applicable indicators are displayed in the drop-down list based on the specified model function.</p>
Alert condition	<p>Specifies the operator to use with the alert threshold value to determine whether the model is performing within an acceptable tolerance for the associated assessment indicator. An example is whether the Misclassification rate for the model is below 25%.</p>
Alert threshold	<p>Specifies the threshold value to use with the alert condition operator to determine whether the model is performing within an acceptable tolerance for the associated assessment indicator.</p>
KPI Rules	<p>This section contains the properties that are used for specifying key performance indicator (KPI) alert notifications.</p> <p>You can specify the rules for key performance indicators (KPIs) to use in order to determine what notifications the system should send to users at the</p>

Property	Description
	<p>project level. Alert notifications are sent when the associated alert condition and threshold are not met.</p> <p>You can add, modify, or delete KPI rules. For more information, see “Add KPI Rules” on page 36.</p>


Table 4.2 Types of Model Functions

Model Function	Description
Analytical	Function for any model that is not Prediction, Classification, Clustering, or Segmentation.
Classification	Function for models that have target variables that contain binary, categorical, or ordinal values.
Clustering	Function for clustering models.
Computer vision - Image classification	Function for computer vision models that are used for image classification.
Computer vision - Object detection	Function for computer vision models that are used for object detection in a classification network.
Computer vision - Keywords detection	Function for computer vision models that are used for keywords detection in neural networks.
Computer vision - Image segmentation	Function for computer vision models that are used for image segmentation.
Prediction	Function for models that have interval targets with continuous values.
Segmentation	Function for segmentation models.
Single-series forecasting	Function for models used to forecast future data based on past data for a single time series.
Text analytics	<p>Function for a SAS Visual Text Analytics modeling project.</p> <p>Note: This model function appears on the Properties tab of a project only when a SAS Visual Text Analytics project contains different types of text models. It</p>


Model Function	Description
	is not displayed as an option in the model function drop-down list
Text categories	Function for SAS Viya Text Analytics categorization models.
Text concepts	Function for SAS Viya Text Analytics concepts models.
Text sentiment	Function for SAS Viya Text Analytics sentiment models.
Text topics	Function for SAS Viya Text Analytics topics models.
Transformation	Function for models that are used to determine mathematical functions, which can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables to improve the fit of your model.


Add KPI Rules

When you add a new rule to the current project, a default rule name is generated that is a combination of the KPI name and the names of values entered in the alert condition and alert threshold fields. The default name can be modified. You can also specify the priority and the number of occurrences that must be reached before sending an alert notification.

- 1 Click **+**. The Add KPI Rule window appears.
- 2 Select a key performance indicator (KPI).
- 3 Select an alert condition.
- 4 Enter a value for the alert threshold.
- 5 Select a priority for the KPI alert.
- 6 Enter a value for the number of occurrences after which you want to set an alert for this KPI rule.
- 7 Accept the default rule name or enter your own rule name.
- 8 Click **OK**.
- 9 Click .

Here are other actions that you can perform:

- Modify the values of existing KPI rules.
- Select one or more KPI rules and click  to delete them from the current project.

Note: After modifying or deleting KPI rules, you must click  to save the changes to the project.


Add User-Defined Properties

You can add your own project or model properties. The property-value pair is metadata for the project or model, both of which can be searched.


To add user-defined properties:


- 1 On the **Properties** tab, select **User-Defined**.
- 2 Click **Add Property**. The Add Property window appears.

TIP If user-defined properties already exist, click **+** above the table.

- a Enter a name for the property.
 - b Select a data type for the property.
 - c Enter a value for the property.
 - d Click **Add** to add the property to the list.
 - e Repeat steps a through d for each property that you want to add.
- 3 Click .

To edit a property:

- 1 Click on a property name within the table.
- 2 Edit the name, the data type, or value of the property.
- 3 Click **OK**.
- 4 Click .

To delete properties, select one or more properties in the table, and then click .

Add Keywords

You can add one or more keywords to a project. When you add keywords to a project, they are also added to a global list of keywords that is available to be added to other projects within the common model repository.

Note: Keyword property values for a project appear as a comma-delimited list in the **Keywords** column of the Projects category view. The functions for sorting projects by the **Keywords** column and searching projects for keyword values are currently not supported.



To add a keyword:

- 1 On the **Properties** tab of a project, select **Keywords**.
- 2 Enter a value and select an existing keyword, or press Enter to add a new keyword.

Note: The keyword name can contain only alphanumeric characters, double-byte characters, the underscore (_), the hyphen (-), and the period (.). Spaces are not allowed.

Repeat this step for additional keywords.

- 3 Click .

To delete a keyword, click  within the token, and then click . The keyword is removed from the project.

Managing Project Files


On the **Files** tab of a project, you can add, modify, and delete custom key performance indicator (KPI) SAS code files. You can also download and run standard and custom KPI SAS code files. The standard KPI file cannot be deleted. The project files are grouped into categories based on their file role and file name. The standard and custom KPI files are used when running performance for models within a project.

CAUTION

Do not add or include files that contain executable or dynamic content that is not valid or approved for your environment. Examples of executable content are batch (.bat) files or executable (.exe) files. Examples of dynamic content are HTML, Microsoft Office documents, JavaScript, dynamic PDF forms, and other forms of executable content. This type of content, when downloaded, could execute in your local environment and provide a way for malicious content to enter it. The Files service is used when adding data files, model files, and project files. As general guidance, SAS recommends that users be vigilant and cautious with any content that is added using the Files service and accessed within SAS Model Manager.

Add Files


Note: Currently, when you add a new file to a project, the file role is set to **Custom KPI**.

- 1 Click **+**. The **Add File** window appears.
- 2 Click , select a file to add to the project, and then click **Open**.

Note: You can add only those files that have a .sas file extension.

- 3 Click **Add**.

Delete Project Files

Select the file and click . In the confirmation message, click **Delete**.

Note: The standard KPI file cannot be deleted.

Download Project Files

Select the file and click . The project file is downloaded to your local machine.

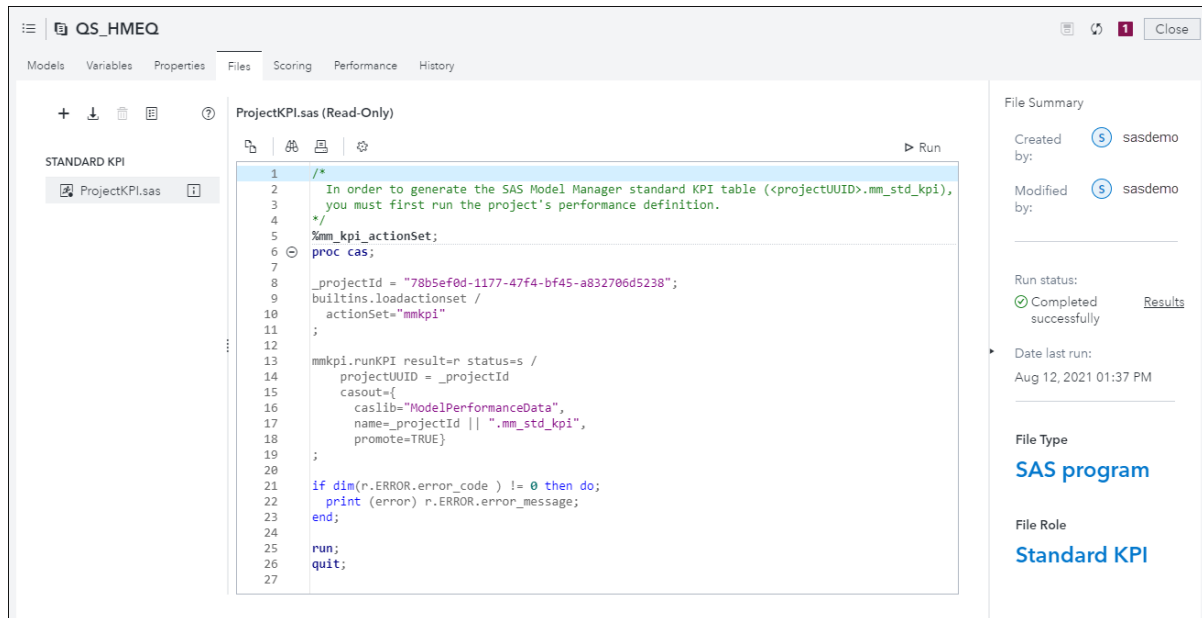
Run a Project Code File

You can run standard KPI and custom KPI code files. However, you cannot edit or delete the standard KPI file (ProjectKPI.sas).

Note: The SAS Model Manager standard KPI table (<projectUUID>.mm_std_kpi) is also generated when running the project's performance definition.

- 1 Select a file and click **>Run** to execute the code within the file.

Note: You can view the status of the running code within the **File Summary** panel. When the run status has a value of completed, the **Results** link appears in this panel. The **View results** button is available in the toolbar





2 Click  **View results** in the toolbar.

The screenshot shows the 'Last Run Results' table in the SAS Studio interface. The table has the following columns: TimeSK, TimeLabel, ProjectUUID, ModelName, and ModelUUID. The data is as follows:

TimeSK	TimeLabel	ProjectUUID	ModelName	ModelUUID
4	Q4	c6e24ba6-382c-490d-b3c4-f7...	QS_Reg1	b1b51bdc-9f7a-4b3c-afbc-2b...
4	Q4	c6e24ba6-382c-490d-b3c4-f7...	QS_Tree1	b300a948-a54e-4dd5-9bd0-d...
3	Q3	c6e24ba6-382c-490d-b3c4-f7...	QS_Reg1	b1b51bdc-9f7a-4b3c-afbc-2b...
3	Q3	c6e24ba6-382c-490d-b3c4-f7...	QS_Tree1	b300a948-a54e-4dd5-9bd0-d...
2	Q2	c6e24ba6-382c-490d-b3c4-f7...	QS_Reg1	b1b51bdc-9f7a-4b3c-afbc-2b...
2	Q2	c6e24ba6-382c-490d-b3c4-f7...	QS_Tree1	b300a948-a54e-4dd5-9bd0-d...
1	Q1	c6e24ba6-382c-490d-b3c4-f7...	QS_Reg1	b1b51bdc-9f7a-4b3c-afbc-2b...
1	Q1	c6e24ba6-382c-490d-b3c4-f7...	QS_Tree1	b300a948-a54e-4dd5-9bd0-d...

View File Properties

To view the file properties, hover over the  icon that appears next to a project file or select a file and then click  in the toolbar.

Managing Project Versions

A project version is a container of models. An initial version is created automatically when you create a project. You can view a list of the project versions on the **Models** tab in the **Version** drop-down list. The latest version is displayed by default. You can also choose to display all versions, create a new version, or manage existing versions from the **Version** drop-down list. When you create a new project version,

you can specify a name and description for the version, such as a time interval for a project cycle.

A version is a sequential number that increments by one each time you add a new version. A project can contain multiple editable versions. A project version is used to differentiate collections of models that are meant to solve the project's problem over time-boundaries. Your version might represent a calendar year, a retail season, or a fiscal quarter. A version contains all of the candidate model resources that you need to determine a champion model as well as all champion model resources. For example, you might develop models for a scoring program that determines whether a customer is eligible for a home equity loan.

Create a New Project Version


- 1 On the **Models** tab, click ▼ in the **Version** drop-down list, and select **New version**. The **New Project Version** window appears.
- 2 Enter a name for the version, or accept the default name (for example, Version 2).
- 3 (Optional) Enter a description for the version.
- 4 Click **Save**.

Manage Project Versions

In addition to creating a new project version, you can edit the description of a version, rename a version, or delete a version.


- 1 On the **Models** tab, click ▼ in the **Version** drop-down list, and select **Manage versions**. The Manage Project Versions window appears.
- 2 (Optional) Edit the description of a version.
- 3 Create a new project version.
 - a Click **[*]**. The New Project Version window appears.
 - b Enter a name for the version, or accept the default name (for example, Version 2).
 - c (Optional) Enter a description for the version.
 - d Click **Save**.
- 4 Rename a project version.
 - a Select a version and click **Rename**. The Rename window appears.
 - b Enter a new name for the version.
 - c Click **Rename**.
- 5 Delete a project version.

Note: When only one project version exists, it cannot be deleted. You must also have the appropriate permissions to delete a version.


- a Select a version and click .
 - b Click **Delete** in the confirmation message.
- 6 Click **Close** to close the Manage Project Versions window.

Delete a Project

Note: You must also have the appropriate permissions to delete a project.

- 1 In the Projects category view, select one or more projects.
- 2 Click  and select **Delete**.
- 3 In the confirmation message, click **Delete**.

Rename a Project

- 1 In the Projects category view, select a project, click , and select **Rename**.

Note: Open objects cannot be renamed.

- 2 Enter a new name for the project.
- 3 Click **Rename**.

Search for Projects


In the **Projects** category view, you can perform multiple types of searches:

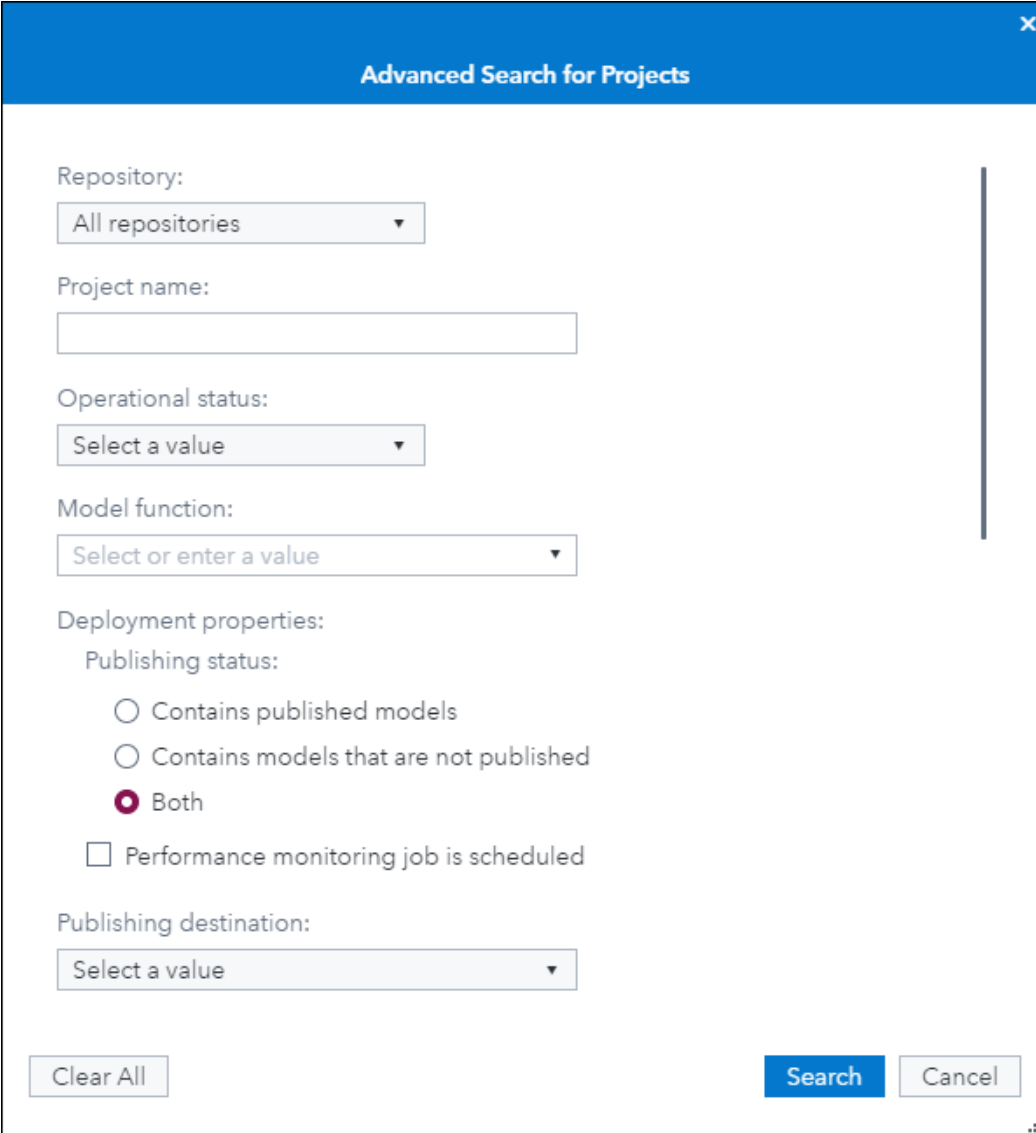
- search for projects by name using the search field above the models list.
- perform an advanced search for projects using project properties, input variables, output variables, or user-defined properties.
- search for projects across applications using the search field in the application bar.

For more information about searching, see “Search” in *SAS Viya Web Applications: General Usage Help*.

Advanced Search for Projects

To perform an advanced search for projects:

- 1 Click  **Advanced search**, which is to the right of the search box. The Advanced Search for Projects window appears.



Advanced Search for Projects

Repository:
All repositories ▼

Project name:

Operational status:
Select a value ▼

Model function:
Select or enter a value ▼

Deployment properties:

Publishing status:

Contains published models

Contains models that are not published

Both



Performance monitoring job is scheduled

Publishing destination:
Select a value ▼

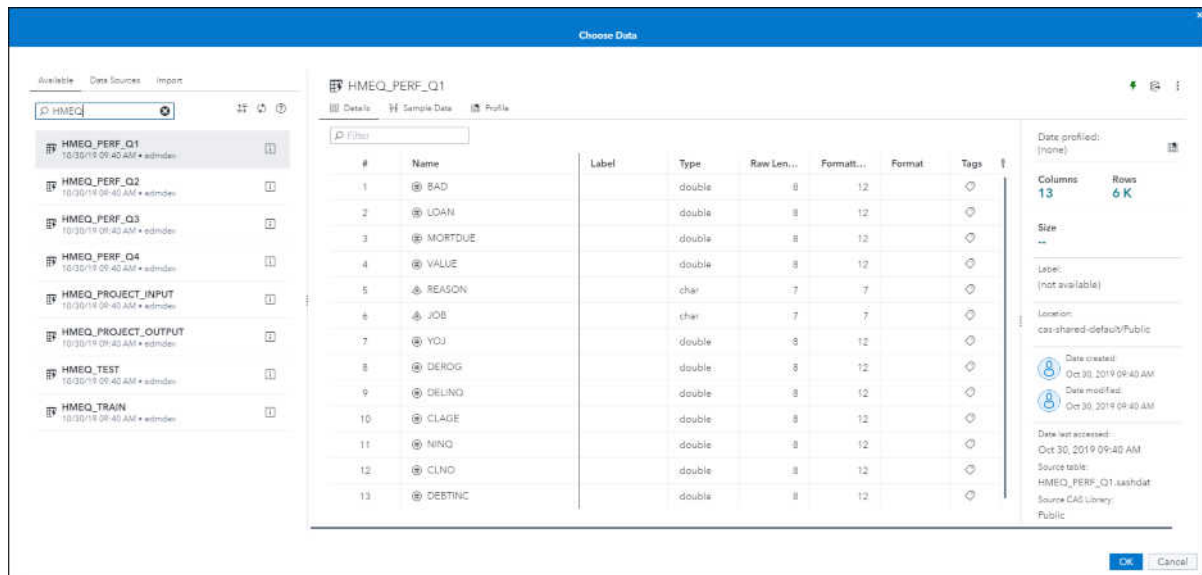
Clear All Search Cancel

- 2 To search for a project in a specific repository, click ▼ in the **Repository** drop-down list, and select the desired repository. **All repositories** is selected as the default for the advanced search.
- 3 Enter a possible value in the **Project name** field to search for a project by name.
- 4 Specify values for one or more of the following fields that you want to search for:

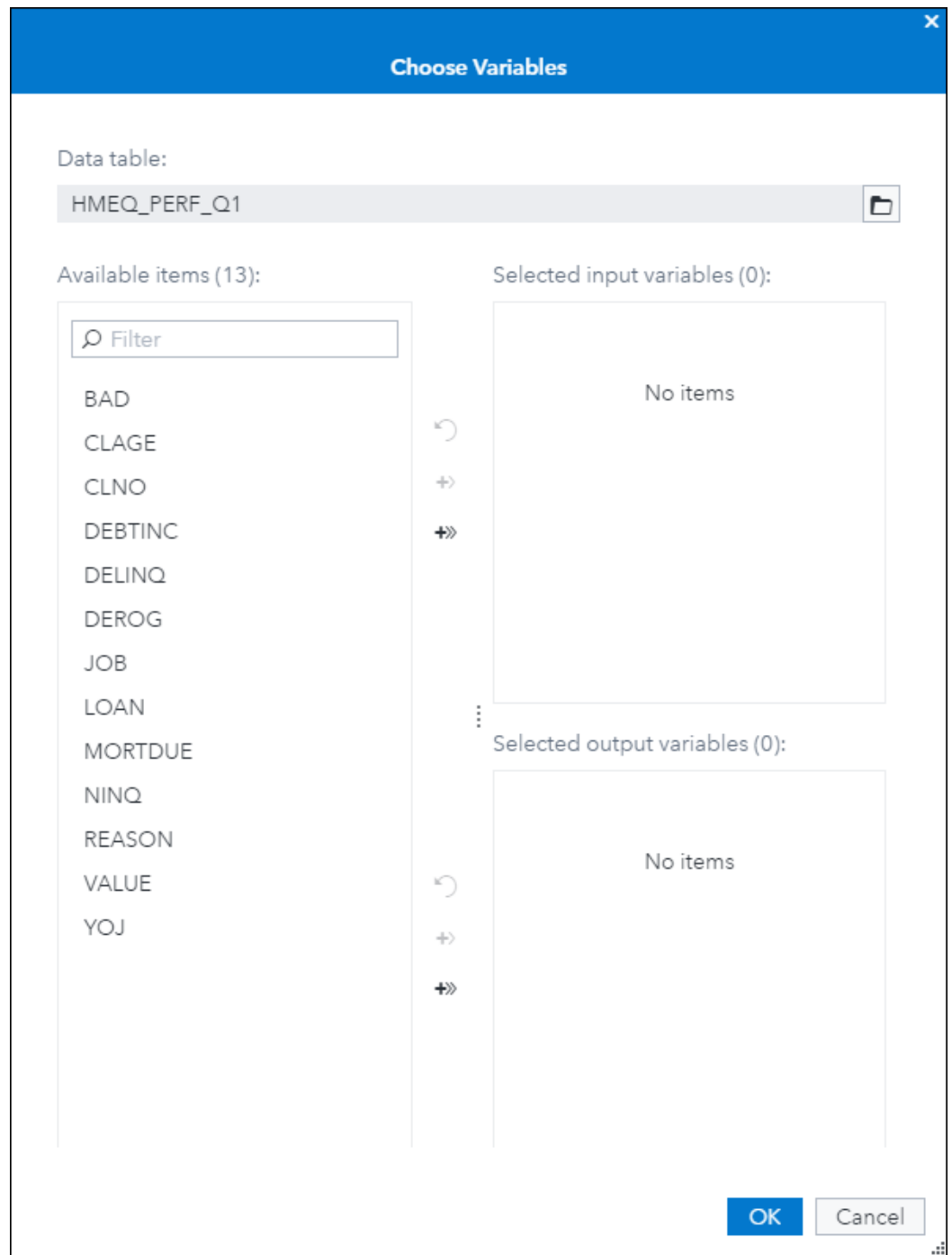
Note: If you previously used the dashboard charts to filter the list of projects, the filters are applied to the associated fields in this window.

- **Operational status**
 - **Model function**
 - **Deployment properties**
 - **Publishing destination**
 - **Date created (range)**
 - **Modeler**
- 5 To search for a project by **Target variable**, enter a value or click  to select a variable from a data table.
 - 6 To search for a project by input variables or output variables, select a data table and choose the variables that are associated with models or project objects.
 - a Click  next to the **Input variables**, or **Output variables** field. The Choose Data window appears.
 - b Select a data table from the list on the left-hand side. The details of the selected data appear on the right-hand side. The sample data and available profiles of the data table are also viewable.

TIP Use the search bar to locate the desired data table and filter search results.




- c When you have selected your data table, click **OK**. The Choose Variables window appears.




- d Select the variables that you want to add and click **+**. You can also click **+** to add all of the variables from the available items list.
- e To remove variables, select the variables that you want to remove and click **←**. Click **←** to remove all of the variables from the available items list.

TIP Click **↶** to undo your last action.

- f Click **OK** to close the Choose Variables window. The variable or variables that you selected appear in the **Input variables** and **Output variables** fields.

TIP Click  next to the **Input variables** or **Output variables** fields in the Search for Projects window to clear the selected variables.

- 7 To search for user-defined properties that are associated with project objects, specify one or more property name and value pairs, or specify property names only.
 - a Click **+** to add a row to the **User-defined properties** table.
 - b Click **▼** in the **Name** drop-down list, and select a user-defined property.
 - c (Optional) Enter the value for the user-defined property in the **Value** column.
 - d Select a row and click  to remove the user-defined property from the search criteria.
- 8 Click **Search**. The Projects category view opens and the project or projects that meet the advanced search criteria are listed.

TIP Click **Clear All** to clear all previously entered search criteria in the Search for Projects window.

Working with Models

About Models	48
View Dashboard and Filter Models	48
View Models Dashboard	48
Filter Models List Using the Dashboard Charts	50
Add a New Custom Model	50
Add a Model within a Folder	51
Add a Custom Model within a Project	52
Import Models	53
Restrictions	54
Import Models into a Folder	55
Import Models into a Project	56
See Also	57
Export a Model	58
Move or Copy a Model	58
Move a Model	58
Copy a Model	59
Delete a Model	59
Rename a Model	59
Manage Model Content and Versions	60
Managing Model Files	60
Managing Variables	62
Modifying Model Properties	64
Managing Model Versions	70
Publish an Open Model Object	71
Search for Models	73
Advanced Search for Models	73
View Model Relationships and Related Objects	77

About Models

You can add new custom models or import existing models using the SAS Model Manager web application. Models can be stored within a folder or project version in the SAS Model Manager common model repository.

The Models category view enables you to access all of the models in the common model repository in one place. The models can be located in a folder, or a project version. You can import models, add new custom models, compare models, and export models. You can also view model relationships and related objects, as well as search for models. Models that are within a project version can be published from the Projects category view or when opened from the Models category view. Models within folders cannot be published.

Several options exist for importing open-source models such as Python and R into SAS Model Manager. You can place your model files in a ZIP file and then import the model, or you can add a new custom model and add local files. You can also use native Python code in a Jupyter notebook to submit REST API requests to the Model Repository API. For more information, see [model-management-resources](#) GitHub repository.

Note: Models can also be built using Model Studio, SAS Visual Analytics, or SAS Studio and then registered into the common model repository.

For more information, see the following documentation:

- [Getting Started with SAS Visual Data Mining and Machine Learning](#)
 - [Welcome to Model Studio](#)
 - [SAS Visual Analytics: Getting Started with Analytical Models](#)
 - [SAS Studio: Task Reference Guide](#)
-

View Dashboard and Filter Models

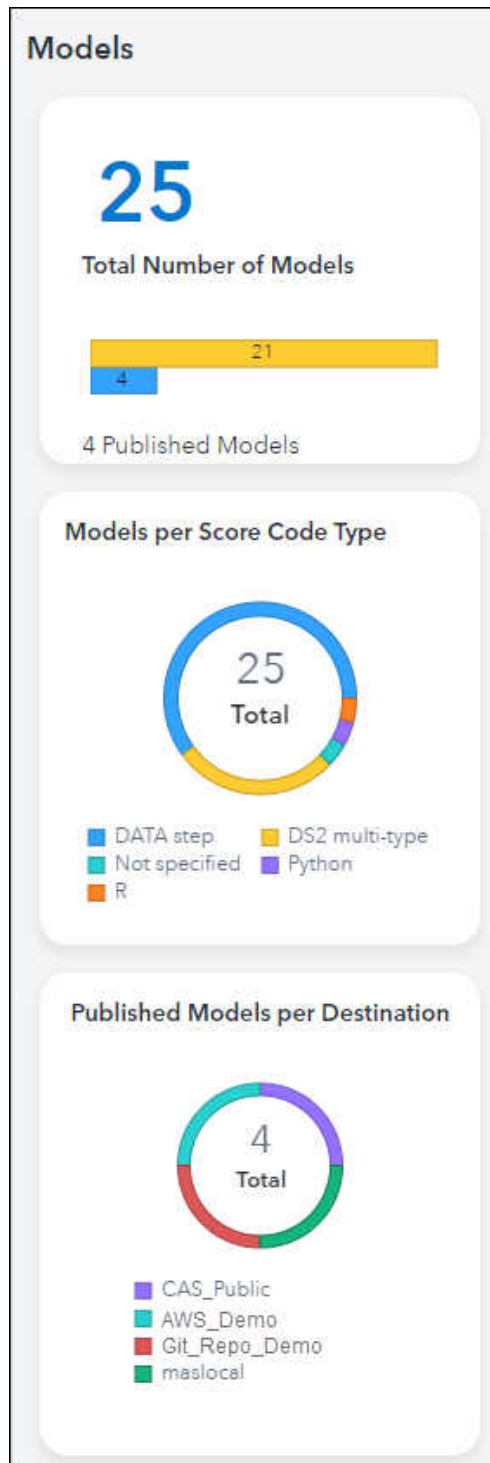
View Models Dashboard

When you first view the Models category the dashboard appears at the top of the application window. The dashboard contains a graphical representation of the following metrics:

- total number of models
- number of published models

- number of models per score code type
- number of published models per destination

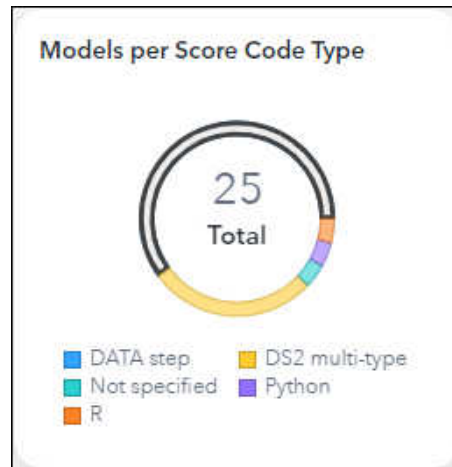
You can click **Hide tiles** or **Show tiles** on the toolbar.



Filter Models List Using the Dashboard Charts

You can click various sections of the different dashboard charts to filter the list of models. You can apply filters for each of the charts to create a compound query.

Here is an example of filtering the list of models to display only models that have a score code type of **DATA step**:



You can also use the Advanced Search for Models window to specify additional information in order to further filter the list. The filters that you applied using the charts are reflected in the Advanced Search for Models window. For more information, see [“Advanced Search for Models” on page 73](#).

TIP To remove an individual filter from a chart, you can click the empty area beside the chart or you can click the × icon within the tokens that are located above the list of models. You can also remove all filters by clicking the × icon, which is located in the right side of the toolbar.

Add a New Custom Model

You can add a new custom model from one or more model files and store it within a folder or within a project version. When you add a new custom model in the Models category view, you can choose a repository and folder to store the new model. When you create a model from the **Models** tab of a project, you can select a project version to store the model.


CAUTION

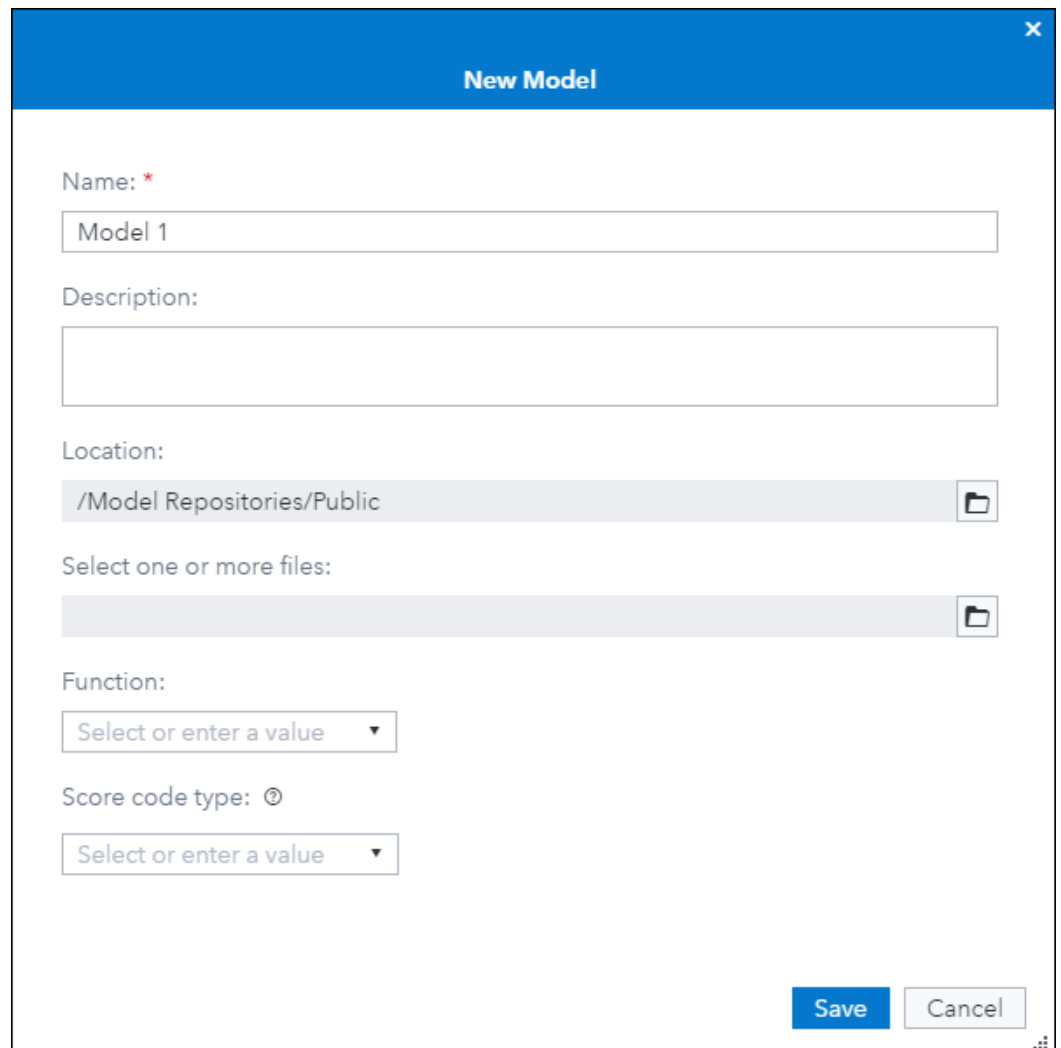
Do not add or include files that contain executable or dynamic content that is not valid or approved for your environment. Examples of executable content are batch (.bat) files or executable (.exe) files. Examples of dynamic content are HTML,

Microsoft Office documents, JavaScript, dynamic PDF forms, and other forms of executable content. This type of content, when downloaded, could execute in your local environment and provide a way for malicious content to enter it. The Files service is used when adding data files, model files, and project files. As general guidance, SAS recommends that users be vigilant and cautious with any content that is added using the Files service and accessed within SAS Model Manager.

Add a Model within a Folder

To add a new custom model in a repository or folder:

- 1 Click  to navigate to the Models category view.
- 2 Click **Add models** and select **New custom model**. The New Model window appears.



The screenshot shows the "New Model" dialog box. The title bar is blue with the text "New Model" and a close button (X). The main area is white and contains the following fields:

- Name: ***: A text input field containing "Model 1".
- Description:**: A large, empty text area.
- Location:**: A text input field containing "/Model Repositories/Public" and a folder icon.
- Select one or more files:**: A text input field and a folder icon.
- Function:**: A dropdown menu with the text "Select or enter a value".
- Score code type: @**: A dropdown menu with the text "Select or enter a value".




At the bottom right, there are two buttons: "Save" (blue) and "Cancel" (white).

- 3 Enter a name for the new custom model.
- 4 (Optional) Enter a description for the new model.

- 5 Click  to select a location to store the new model, and then click **OK**.

Note: In the Choose a Location window, you can create a folder within a repository folder to store projects and models for your organization. Repository folders must be created using the Settings window. For more information, see [“About Model Repositories” on page 17](#).


It is recommended that the `DMRepository` and `VTARespository` repository folders should be reserved for projects and models that are registered from Model Studio. Also, the `VARespository` repository folder should be reserved for models that are registered from SAS Visual Analytics. For more information, see [“Register Models” in SAS Visual Data Mining and Machine Learning: User’s Guide](#).




- 6 (Optional) Click , select one or more files to include within the new model, and then click **Open**.
- 7 To select a function, click  in the **Function** drop-down list, and select the desired function.
- 8 To select a score code type, click  in the **Score code type** drop-down list, and select the desired score code type.
- 9 Click **Save**. The new model object opens.

Add a Custom Model within a Project

To add a new custom model within a project version:

Note: By default, models are added within the latest project version. You can select a different project version from the **Version** drop-down list.

- 1 Click  to navigate to the Projects category view.
- 2 Open a project.
- 3 Click **Add models** and select **New custom model**. The New Custom Model window appears.

- 4 Enter a name for the new model.
- 5 (Optional) Enter a description for the new model.
- 6 (Optional) Click , select one or more files to include within the new model, and then click **Open**.
- 7 Select or enter a value for the function. To select a function, click  in the **Function** drop-down list, and select the desired function.
- 8 To select a score code type, click  in the **Score code type** drop-down list, and select the desired score code type..
- 9 Click **Save**. The new model object opens.

TIP After you edit the model content, click  to return to the project.

Import Models

You can import models into a project or into a folder. However, only specific file types can be used when you import models into a project or folder. Models that are imported into a folder can then later be moved into a folder or project version. If a model already exists in a project version with the same value for the external model

ID property, a new model version is created. The model name is updated only when a new name is specified and the external model ID is the same. Otherwise, a model with the same name cannot be imported.

Note: You cannot score, publish, or monitor performance of models that are within a folder. Models must be within a project version in order for you to perform these tasks.

Restrictions

Here are the file types that can be used to import models:

CAUTION

Do not add or include files that contain executable or dynamic content that is not valid or approved for your environment. Examples of executable content are batch (.bat) files or executable (.exe) files. Examples of dynamic content are HTML, Microsoft Office documents, JavaScript, dynamic PDF forms, and other forms of executable content. This type of content, when downloaded, could execute in your local environment and provide a way for malicious content to enter it. The Files service is used when adding data files, model files, and project files. As general guidance, SAS recommends that users be vigilant and cautious with any content that is added using the Files service and accessed within SAS Model Manager.

PMML XML File

an XML file that contains PMML model information. Predictive Modeling Markup Language (PMML) is an XML-based standard for representing data mining results. You can import PMML models that are produced by other applications. PMML 4.2 is supported. Some classification and prediction models that are created using PMML 4.2 are converted to DATA step score code during the model import process. The file extensions can be .xml or .pmml, provided that the file contains valid PMML XML code. When you are importing valid PMML models, the score code type model property is set to DATA step, instead of PMML. PMML models with a score code type of DATA step can be scored and published. For more information, see [“Concepts: PMML Support” on page 155](#).

Note: PMML XML files cannot be edited using the code editor. They are read-only. Also, files with the .pmml file extension cannot be viewed within the editor.

SAS analytic store (ASTORE) file

a file that is created by using the ASTORE procedure, as well as by using some SAS Visual Data Mining and Machine Learning modeling procedures or CAS actions. An ASTORE file is a binary representation of a model that encapsulates in a portable format the scoring logic and metadata such as input and output variables. The file extensions can be .sasast or .astore, provided that the file contains a valid analytic store model.

SAS package (SPK) file

a compressed container file that contains a mining result and model component files. SPK files can be created using SAS Enterprise Miner or SAS 9.4 modeling procedures and macros.



ZIP file

an archive file that contains model files. Model files that are associated with a specific model are stored within the ZIP file. The ZIP file can contain model folders at the same level or in a hierarchical folder structure. Each model folder within the ZIP file is imported as a separate model object that contains the contents of the model folder. When you import models from a ZIP file into a project version, the hierarchical folder structure is ignored.

Import Models into a Folder


To import models into a folder:

Note: You can import models only into repository folders within the common model repository (**Model Repositories** folder) or folders within a repository folder.

- 1 Click  to navigate to the Models category view.
- 2 Click **Add models** and select **Import**. The Import Models window appears.
- 3 Click  to select a location to store the models, and then click **OK**.

Note: In the Choose a Location window, you can create a folder within a repository folder to store projects and models for your organization. Repository folders must be created using the Settings window. For more information, see [“About Model Repositories” on page 17](#).

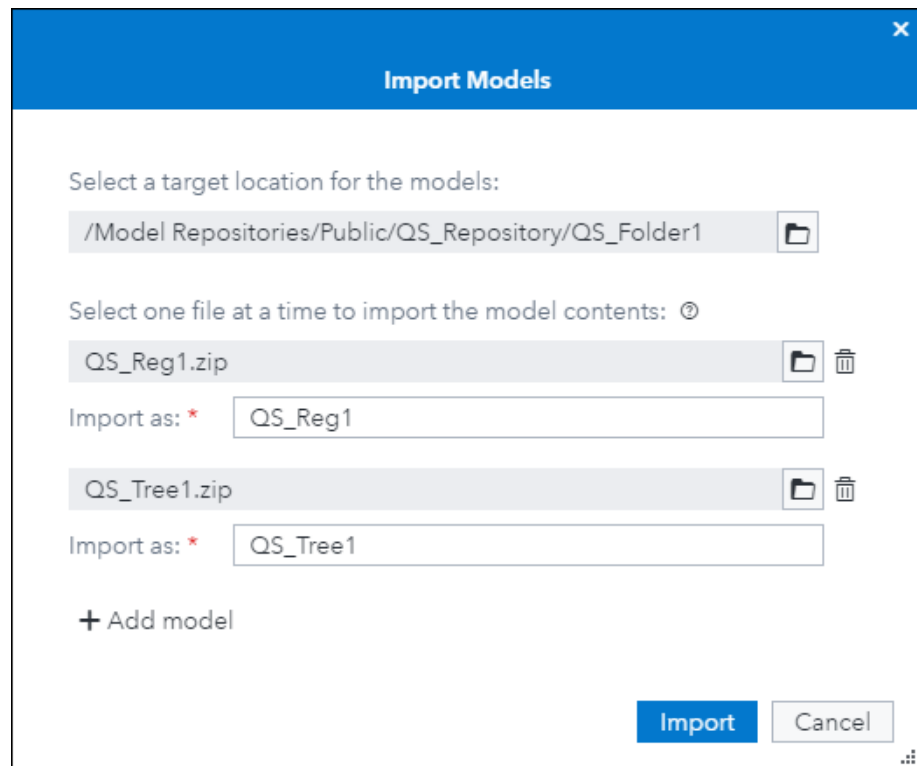
It is recommended that the **DMRepository** and **VTARespository** repository folders should be reserved for projects and models that are registered from Model Studio. Also, the **VARespository** repository folder should be reserved for models that are registered from SAS Visual Analytics. For more information, see [“Register Models” in SAS Visual Data Mining and Machine Learning: User’s Guide](#).

- 4 Click  to select a file that contains your model contents. Select only one file at a time in the Open window. The name of the selected file is used as the default model name.

Click **Open**.

Note: If you previously exported an analytic store model from SAS Model Manager, the analytic store files are included in the ZIP file. When you re-import the analytic store model to the same system or another system, the analytic store files are copied to the ModelStore caslib.

- 5 Click **+ Add model** to add rows so that you can import more models.
- 6 Repeat steps 4 and 5 until you have selected all of the models that you want to import.





TIP To remove extra lines, click  before you click **Import**.

- 7 Click **Import**.

Import Models into a Project

To import model into a project version:

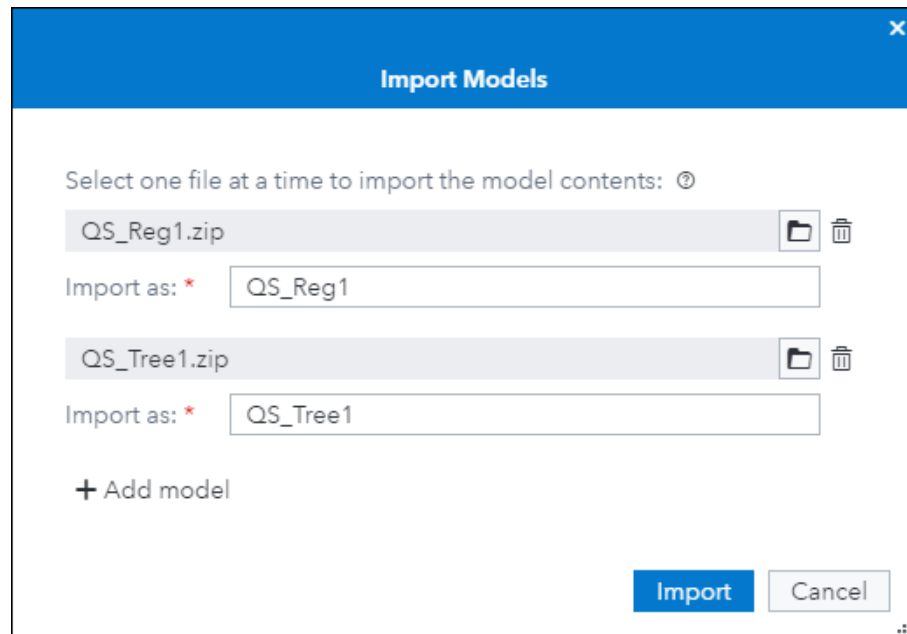
- 1 Click  to navigate to the Projects category view.
- 2 Open a project.
- 3 Click **Add models** and select **Import**. The Import Models window appears.
- 4 Click  to select a file that contains your model contents. Select only one file at a time in the Open window. The name of the selected file is used as the default model name.


Click **Open**.

Note: If you previously exported an analytic store model from SAS Model Manager, the analytic store files are included in the ZIP file. When you re-import the analytic store model to the same system or another system, the analytic store files are copied to the ModelStore caslib.

- 5 Click **+ Add model** to add rows so that you can import more models.

- 6 Repeat steps 4 and 5 until you have selected all of the models that you want to import.



TIP To remove extra lines, click  before you click **Import**.

- 7 Click **Import**.

Note: If a model already exists in a project version with the same value for the external model ID property, a new model version is created. The model name is updated only when a new name is specified. However, if the external model ID property values do not match for the model with the same name, the model cannot be imported. For more information about the external model ID property and other model properties, see [Table 5.1 on page 64](#).

See Also


- “Register a Model” in *SAS Studio: Task Reference Guide*
- %MM_IMPORT_MODEL Macro
- %MM_IMPORT_ASTORE_MODEL Macro

Export a Model

You can export one model at a time from the Models category view or from the **Models** tab of a project.

Note: When you export an analytic store model, the analytic store files are included in the ZIP file. If you re-import the analytic store model to the same system or another system, the analytic store files are copied to the ModelStore caslib.

To export a model:

- 1 Select a model from the list.
- 2 Click  and select **Export as ZIP**.

The contents of the model in a ZIP file is downloaded to your local machine.

Move or Copy a Model



You can move a model from a folder to another folder or project version using the Models category view. Only SAS Administrators and users who have Delete permission for the source location where the model resides and Write permission for the target location can move a model. By default, all other users can copy a model from a folder or another project from the **Models** tab of a project only.

For more information, see [“Managing Permissions” in SAS Model Manager: Administrator’s Guide](#).

Move a Model




To move a model:

Note: Only models located within a folder can be moved.

- 1 Click  to navigate to the Models category view.
- 2 Select a model from the list.
- 3 Click  and select **Move**. The Choose a Location window appears.
- 4 Navigate to the folder or project version that you want to move the model to.
- 5 Click **OK**.

Copy a Model

To copy a model from a folder or another project:


- 1 Open a project and click the **Models** tab.
- 2 Click **Add models** and select **Copy from** from the drop-down list. The Choose a Model window appears.
- 3 Click  to navigate to a folder or a project version.
- 4 Click  for the model folder and select the model object. The model object is indicated by the icon .
- 5 Click **OK**.

Note: Only the latest version of the source model is copied in to the project as a new model object. The initial version for the model is 1.0.

Delete a Model

You can delete one or more models at a time from the Models category view or from the **Models** tab of a project.

Note: The project champion model and open model objects cannot be deleted.


- 1 Select one or more models.
- 2 Click  and select **Delete**.
- 3 In the confirmation message, click **Delete**.

Rename a Model

You can rename one model at a time from the Models category view or from the **Models** tab of a project.

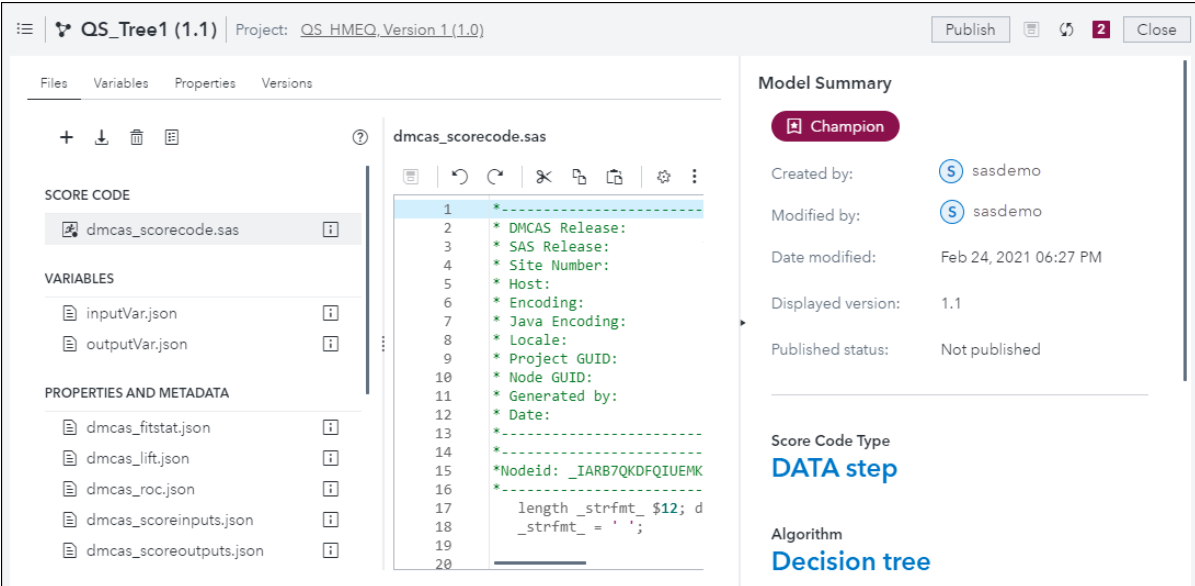
Note: Open objects cannot be renamed.

To rename a model:


- 1 Select a model, click , and select **Rename**.
- 2 Enter a new name for the model.
- 3 Click **Rename**.

Manage Model Content and Versions

When you open a model, you can manage model files, add model input and output variables, modify the model properties, and add or view model versions. You can also publish a model, if it is located within a project version. Models within a folder cannot be published. You can open a model from the Models category view and from the **Models** tab of a project. The primary information about the displayed version of a model is shown in the **Model Summary** panel.



The screenshot shows the SAS Model Studio interface for a model named "dmcas_scorecode.sas". The interface is divided into several panels:

- Files Panel (Left):** Shows a list of files categorized into "SCORE CODE" (dmcas_scorecode.sas), "VARIABLES" (inputVar.json, outputVar.json), and "PROPERTIES AND METADATA" (dmcas_fitstat.json, dmcas_lift.json, dmcas_roc.json, dmcas_scoreinputs.json, dmcas_scoreoutputs.json).
- Code Editor (Center):** Displays the SAS code for "dmcas_scorecode.sas". The code includes comments for DMCAS Release, SAS Release, Site Number, Host, Encoding, Java Encoding, Locale, Project GUID, Node GUID, and Generated by. It also contains a macro call: `length _strfmt_ $12; d _strfmt_ = ' ';`.
- Model Summary Panel (Right):** Provides metadata for the model, including:
 - Champion: 
 - Created by: sasdemo
 - Modified by: sasdemo
 - Date modified: Feb 24, 2021 06:27 PM
 - Displayed version: 1.1
 - Published status: Not published
 - Score Code Type: **DATA step**
 - Algorithm: **Decision tree**

Managing Model Files


On the **Files** tab of a model, you can add, delete, and download files, as well as assign roles to model files. The model files are grouped into a few primary categories based on their file role and file name. You can add most file types to a model. You can also edit supported file types in the code editor. File types that are not supported by the editor or that are read-only can be downloaded, and they can be viewed by another application.

CAUTION



Do not add or include files that contain executable or dynamic content that is not valid or approved for your environment. Examples of executable content are batch (.bat) files or executable (.exe) files. Examples of dynamic content are HTML, Microsoft Office documents, JavaScript, dynamic PDF forms, and other forms of

executable content. This type of content, when downloaded, could execute in your local environment and provide a way for malicious content to enter it. The Files service is used when adding data files, model files, and project files. As general guidance, SAS recommends that users be vigilant and cautious with any content that is added using the Files service and accessed within SAS Model Manager.

Add Files

- 1 Click **+**. The **Add Model Files** window appears.
- 2 Click , select one or more files to add to the model, and then click **Open**.
- 3 Click **Add**.


Assign Model File Roles

Place your pointer over  next to a model file to view the file properties. Roles might need to be assigned for your model files. To assign a role, select the file and click . Enter a value for the file role or select the file role from the drop-down menu. Click **Save**.

Some roles such as **Score code** and **Score resource** are automatically assigned when you import or create new models based on their file names. The files are also grouped into a few primary categories based on their file role and file name.

Note: The file type for the file that is assigned the file role of **Score code** must be associated with the value that is specified for the score code type model property. For example, if the file name is myScoreCode.py, then the value for the score code type should be Python. This ensures that the correct score code file is used and that valid code is generated for a scoring job.

Delete Model Files

Select the file and click . In the confirmation message, click **Delete**.

Download Model Files

Select the file and click . The model file is downloaded to your local machine.

Managing Variables


Input variables and output variables can be added to both project and model objects on the **Variables** tab of an object. The same variable name cannot be used for both an input and output variable. You can also map model output variables to project output variables for models with a score code type of DATA step, SAS program, or PMML.

Note: When you add, edit, or delete model variables, you might receive a warning message indicating that the model role will be cleared. This happens when the model is set as the project champion or as a challenger.

Add Variables from a Data Table

- 1 Click the **Variables** tab.
- 2 Click **Add variable** and select **Data table**. The Choose Data window appears.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see [“Working with SAS Data Explorer” in SAS Data Explorer: User’s Guide](#).

- 3 Select the data table that you want to import the variables from and click **OK**. The Choose Variables window appears.
- 4 Select the variables that you want to add and click ➔ to add the variables to the input variables or output variables list. You can also click ➤ to add all of the variables from the available items list.
- 5 Click **OK**.
- 6 Click .

Add Custom Variables


- 1 Click the **Variables** tab.
- 2 Click **Add variable** and select **Custom variable**. The Add Custom Variables window appears.

Note: If there are no existing variables, click **Add Custom Variables**.



- 3 Enter a name for the variable.
- 4 Select a data type and variable type.
- 5 Expand the **Optional** section to specify a length, measurement, and description for the variable.
- 6 Click **Add**.
- 7 Repeat steps 3 through 6 for each variable that you want to add.
- 8 Click **OK**.

Edit Variables

To edit variables:

- 1 Click the **Variables** tab.
- 2 Click on the name of the variable that you want to edit. The Edit Variable window appears.
- 3 Edit the properties as needed and click **OK**.
- 4 Click .

Delete Variables



- 1 Click the **Variables** tab.
- 2 Select the check box for the variables that you want to delete, click , and then select **Delete**.
- 3 Click .

Map Output Variables

To set a model as the project champion or as a challenger, run performance, or publish a model from a project, you must map the model output variables to the project output variables. If you do not map the output variables after importing a model, you are prompted to map them when setting a model as the project champion or as a challenger.

Note: You can modify the model output variable mappings only for a model with a score code type of DATA step, SAS program, or PMML. The names of model output variables must also match the names of the project output variables for other types of models. Otherwise, you cannot run performance for a model or publish a model.

To map output variables:

- 1 Click the **Variables** tab of a model.
- 2 Click  and select **Map output variables**. The Map Output Variables window appears.
- 3 Select the model output variables to map with each of the project output variables.
- 4 Click **OK**.
- 5 Click .

Modifying Model Properties

Model properties contain the model metadata. Model metadata includes information such as the name of the model, the type of model, the modeler, the model identifier, the name and path of the repository, and of the tables and variables that are used by model processes. The model properties are organized into three types: General, Relationships, and User-Defined.

IMPORTANT The **Relationships** page is available only when SAS Model Risk Management is installed and you have access to it.

Set Model General Properties

General Properties contains both system-defined properties that you cannot modify, and model specific properties that can be modified, such as the description of the project.


To set the model general properties, click the **Properties** tab, modify the property values, and then click .

Table 5.1 List of General Properties

Property	Description
Name	Specifies the name of the model. It can be renamed only from the Models category view or the Models tab of a project.
Description	Specifies the description of the model.
Location	Specifies the location of the model in the common model repository.
Project name	Specifies the name of the project that contains the model.

Property	Description
Project version	Specifies the project version that contains the model.
Function	Specifies the type of output that your model generates. For more information, see Table 5.2 on page 66 .
Score code type	<p>Specifies the type of score code that your model uses. A value must be specified in order for you to be able to publish a model, run a test, or monitor performance for a model. You can select a value from the list or enter your own value. User-defined values are not added to the list. Instead, they are stored within the model properties.</p> <p>Note: A model with the score code type of DS2 multi-type can contain code files for a DS2 embedded process, a DS2 package, and one or more analytic stores.</p> <p>Depending on score code type, you can score, monitor, or publish a model. For more information, see “High-Level Model Support Matrix for Primary Functions” on page 7.</p>
Train table	Specifies the Train table that is used to validate scoring functions or scoring model files when a user publishes the associated project champion model or challenger models to a database. This property is optional.
Train code type	Specifies the type of train code that your model uses. This property is for informational purposes only. You can select a value from the list or enter your own value. User-defined values are not added to the list. Instead, they are stored within the model properties.
Algorithm	Specifies the computational algorithm that is used for the selected model.
Input variable type	<p>Specifies the type of input variables and whether the variables come from the trainInputVar.json model file or from the inputVar.json model file. Models that are registered from SAS Studio might contain the trainInputVar.json model file. Valid values are <code>score</code> and <code>train</code>.</p> <p>Note: If both files are included with the registered model, the property value is set to <code>score</code>.</p>
Target variable	Specifies the name of the target variable.
Target event value	Specifies the target variable value that defines the desired target variable event.
Target level	Specifies the target level of binary, nominal, ordinal, or interval.
Output category variable	Specifies the output variable name, when the Model function property is set to Text categories .

Property	Description
Output event probability variable	Specifies the output event probability variable name, when the Model function property is set to Classification, Analytical, Computer vision - Image classification, or Transformation .
Output prediction variable	The output prediction variable name, when the Model function property is set to Prediction, Analytical, Computer vision - Image segmentation, Computer vision - Keypoints detection, Computer vision - Object detection, Single-series forecasting, or Transformation .
Output segmentation variable	The output segmentation variable name, when the Model function property is set to Clustering, Analytical, Segmentation, or Transformation .
Output sentiment variable	Specifies the output variable name, when the Model function property is set to Text sentiment .
Output topic variable	Specifies the output variable name, when the Model function property is set to Text topics .
Modeler	Specifies the user ID for the user that built the model.
Tool	Specifies the tool that was used to build the model. An example is Model Studio.
Tool version	Specifies the version number of the tool that is specified in the Tool property.
UUID	Specifies the universally unique identifier for a model object.
External model ID	Specifies the model ID for a model that was registered from an external application, such as Model Studio.
External URL	Specifies a user-defined URL to a model object in another application or to documentation related to the model.

Table 5.2 Types of Model Functions

Model Function	Description
Analytical	Function for any analytical model that is not associated with another specific model function.
Classification	Function for models that have target variables that contain binary, categorical, or ordinal values.
Clustering	Function for segmentation or clustering models.


Model Function	Description
Computer vision - Image classification	Function for computer vision models that are used for image classification.
Computer vision - Object detection	Function for computer vision models that are used for object detection in a classification network.
Computer vision - Keywords detection	Function for computer vision models that are used for keywords detection in neural networks.
Computer vision - Image segmentation	Function for computer vision models that are used for image segmentation.
Prediction	Function for models that have interval targets with continuous values.
Segmentation	Function for segmentation models.
Single-series forecasting	Function for models used to forecast future data based on past data for a single time series.
Text categories	Function for SAS Visual Text Analytics categorization models.
Text concepts	Function for SAS Visual Text Analytics concepts models.
Text sentiment	Function for SAS Visual Text Analytics sentiment models.
Text topics	Function for SAS Visual Text Analytics topics models.
Transformation	Function for models that are used to determine mathematical functions. These functions can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables to improve the fit of your model.

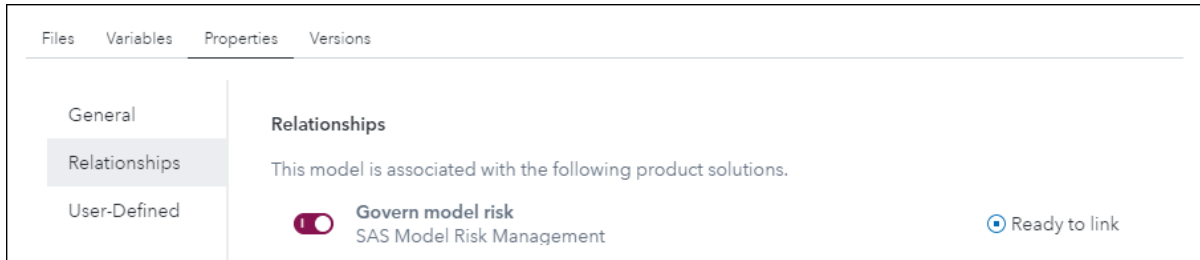
Establish Model Relationships



You can use model relationships to establish an association with objects in other product solutions. When the **Govern model risk** property setting is enabled on the **Relationships** page of a model object, SAS Model Risk Management users can then establish a relationship with the model by linking to it.

IMPORTANT The **Relationships** page is available only if your company has licensed SAS Model Risk Management and you have the appropriate permissions to it.


Here is an example user scenario:

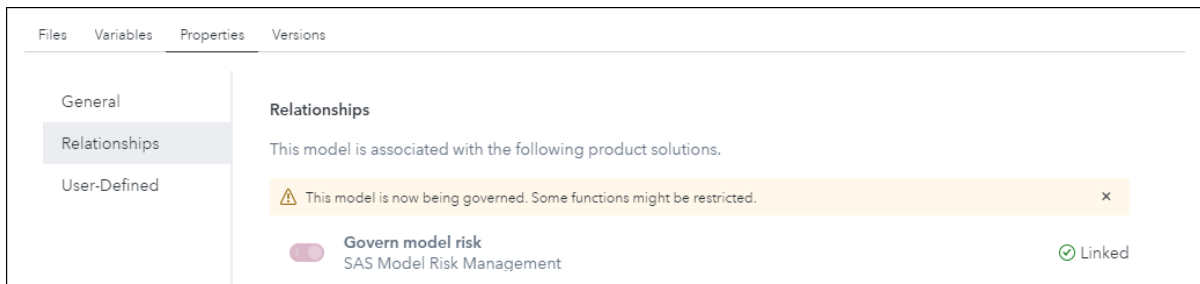
- 1 Open a model.
- 2 On the **Properties** tab, select **Relationships**.
- 3 Enable the **Govern model risk** toggle button and click . The model is now ready to be linked to from a model object in SAS Model Risk Management.




- 4 Click  and select **Govern Model Risk** to switch to SAS Model Risk Management.
- 5 Click  and open the model that you want to establish a relationship with.

TIP For information about the SAS Model Risk Management model object properties and content, see [SAS Model Risk Management: User's Guide](#).

- 6 Click the **Repository Details** tab of the model object.
- 7 Click **Link** and select a source model from the common model repository. Only models that are enabled to govern model risk in SAS Model Manager and that are not already linked are available in the models list.
- 8 Click **Save**.
- 9 Click  and select **Manage Models** to return to SAS Model Manager.
- 10 In the Models category view, click the name of the model that you previously linked to from the SAS Model Risk Management model object. The model object opens
- 11 Return to the **Relationships** page of the model **Properties** tab. The relationship status now indicates that the model is linked.



- 12 Click **Close**.
- 13 In the list of models, click  to the right of a model to view the model risk card for the related object that the model is associated with.

The **Model Risk Card for Related Object** window appears.

14 Click **Close**.


Add User-Defined Properties

You can add your own project or model properties. The property-value pair is metadata for the project or model, both of which can be searched.


To add user-defined properties:


- 1** On the **Properties** tab, select **User-Defined**.
- 2** Click **Add Property**. The Add Property window appears.

TIP If user-defined properties already exist, click **+** above the table.

- a** Enter a name for the property.
 - b** Select a data type for the property.
 - c** Enter a value for the property.
 - d** Click **Add** to add the property to the list.
 - e** Repeat steps a through d for each property that you want to add.
- 3** Click .

To edit a property:

- 1 Click on a property name within the table.
- 2 Edit the name, the data type, or value of the property.
- 3 Click **OK**.
- 4 Click .

To delete properties, select one or more properties in the table, and then click .

Managing Model Versions

The current version of a model is the latest version in which the model properties and file contents are editable. If you add a new model version manually or perform an action that automatically creates a new model version, a snapshot of the model's contents is taken and a version number is assigned. For example, a new model version is automatically created when you set a model as the champion model or publish a champion model from the project level. However, the contents of the new model version that is created can no longer be edited. You can view the contents of the new model version only. Model versions cannot be deleted.

Set the Displayed Version

The displayed version is the version whose information is displayed on the other tabs, such as the **Files**, **Variables**, and **Properties** tabs. The version number for the displayed version appears next to the model name in the object title bar. On the **Versions** tab, a ✓ indicates the displayed version. To change the displayed version, select the version that you want to view, and click **Set Version**.

Note: Here are a few restrictions when creating a new model version.

- The current version of an object is the version that has the highest version number. When you create a new version, SAS Model Manager locks the current version before it creates the new version.
 - You cannot modify a locked version.
 - You cannot unlock a version.
-

Create a New Version

To create a new version:

- 1 On the **Versions** tab, click **New Version**. The New Version window appears.
- 2 Select the version type: **Minor** or **Major**. Version numbers follow the format *Major.Minor*. If you select **Major**, the number to the left of the period is incremented. If you select **Minor**, the number to the right of the period is incremented.

- 3 Click **Save**.

Publish an Open Model Object

When you open a model that is located within a project version, you can publish it from the object toolbar. Models that are located directly within a folder and that are not members of a project cannot be published. For information about publishing models from the **Models** tab of a project or from the Projects category view, see “About Publishing Models” on page 109.

IMPORTANT Before you can publish a Python model or an R model, the source code must be in the correct format. For more information, see “Concepts: Open-Source Models” on page 135.

- 1 Open a model.
- 2 Click **Publish**. The Publish Models window appears.
- 3 Select a destination.

Note: If you have Read and Write permissions to the caslib that is specified in a publishing destination, the destination is shown in the list.

For more information, see *SAS Viya: CAS Authorization Window* and “Configuring Publishing Destinations” in *SAS Model Manager: Administrator’s Guide*.


TIP Click **Details** to view the destination details.

- 4 (Optional) Assign tags to the published model by entering a tag value and selecting an existing tag, or press Enter to add a new tag.

TIP Tag names cannot start with a period or a hyphen, and cannot contain more than 128 characters. They can contain only alphanumeric characters, the underscore (_), the hyphen (-), and the period (.).

- 5 Specify a value for any properties that might appear after you select a destination.

Property	Description	Destination Type	Required
Azure Machine Learning workspace	Indicates the workspace in the Azure Machine Learning cloud-based environment to publish the selected items to.	Azure Machine Learning	Yes

Property	Description	Destination Type	Required
	Select a value from the drop-down list.		
Git directory	<p>Indicates the directory path within the Git repository to publish the selected items to.</p> <p>Click  and navigate to a folder within the Git repository. Click OK.</p> <p>Note: If a folder is not selected within the Git repository, the items are published to the root directory.</p>	Git	No

- 6 (Optional) Update the published name in the **Items to Publish** section. The maximum length and character restrictions differ, depending on your destination. For more information, see [Table 7.1 on page 112](#).
- 7 (Optional) If you have previously published the model with the same published name, you can replace the model. In the **Items to Publish** section, enable the **Replace item with the same name** toggle in order to replace the previously published item of the same name in the same destination.

Note: When publishing models to a container, an Azure Machine Learning, or a Git publishing destination, if a published model with the same name already exists within the destination, a sequential number is added as a tag to indicate the version of the published model.

- 8 Click **Publish**. The Publishing Results window appears. The status of the publishing request is displayed in the **Status** column.

Note: When you select a CAS destination and click **Publish**, the CAS destination table is automatically reloaded and the newly published item is made available to other applications. If the table contains models that are currently in use by SAS Model Manager or another application, you might not want to reload the table at the same time that you publish content. In that case, select **Publish without reloading**. However, you must manually reload the table in order for the newly published content to be accessible.

When you are publishing to a SAS Micro Analytic Service destination, the **Micro Analytic Module** column is also displayed with a URL to the published model.

- 9 When the status changes to **Published successfully**, click **Close**.

Note: The following is true for models that are within a project version: after a model has been published, a new version of the model is created and a

publishing validation test is also created. For more information, see [“Managing Model Versions” on page 70](#) and [“Validate Published Models” on page 85](#).

Search for Models


In the **Models** category view, you can perform multiple types of searches:

- search for models by name using the search field above the models list.
- perform an advanced search for models using model properties, input variables, output variables, or user-defined properties.
- search for model objects across applications using the search field in the application bar.

For more information about searching, see [“Search” in SAS Viya Web Applications: General Usage Help](#).

Advanced Search for Models



To perform an advanced search for models:

- 1 Click  **Advanced search**, which is to the right of the search box. The Advanced Search for Models window appears.

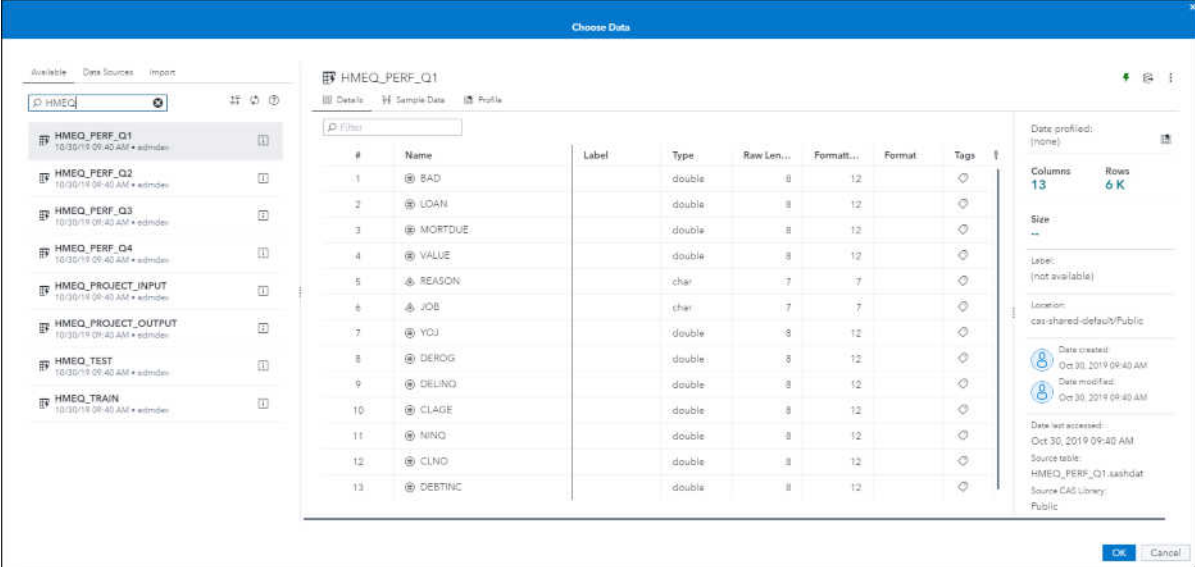
- 2 To search for a model in a specific repository, click ▼ in the **Repository** drop-down list, and select the desired repository. **All repositories** is selected as the default for the advanced search.
- 3 Specify values for one or more of the following fields to include them in your search criteria:

.....
Note: If you previously used the dashboard charts to filter the list of projects, the filters are applied to the associated fields in this window.

- **Model name**
- **Model function**
- **Algorithm**
- **Score code type**
- **Deployment properties**
- **Publishing destination**

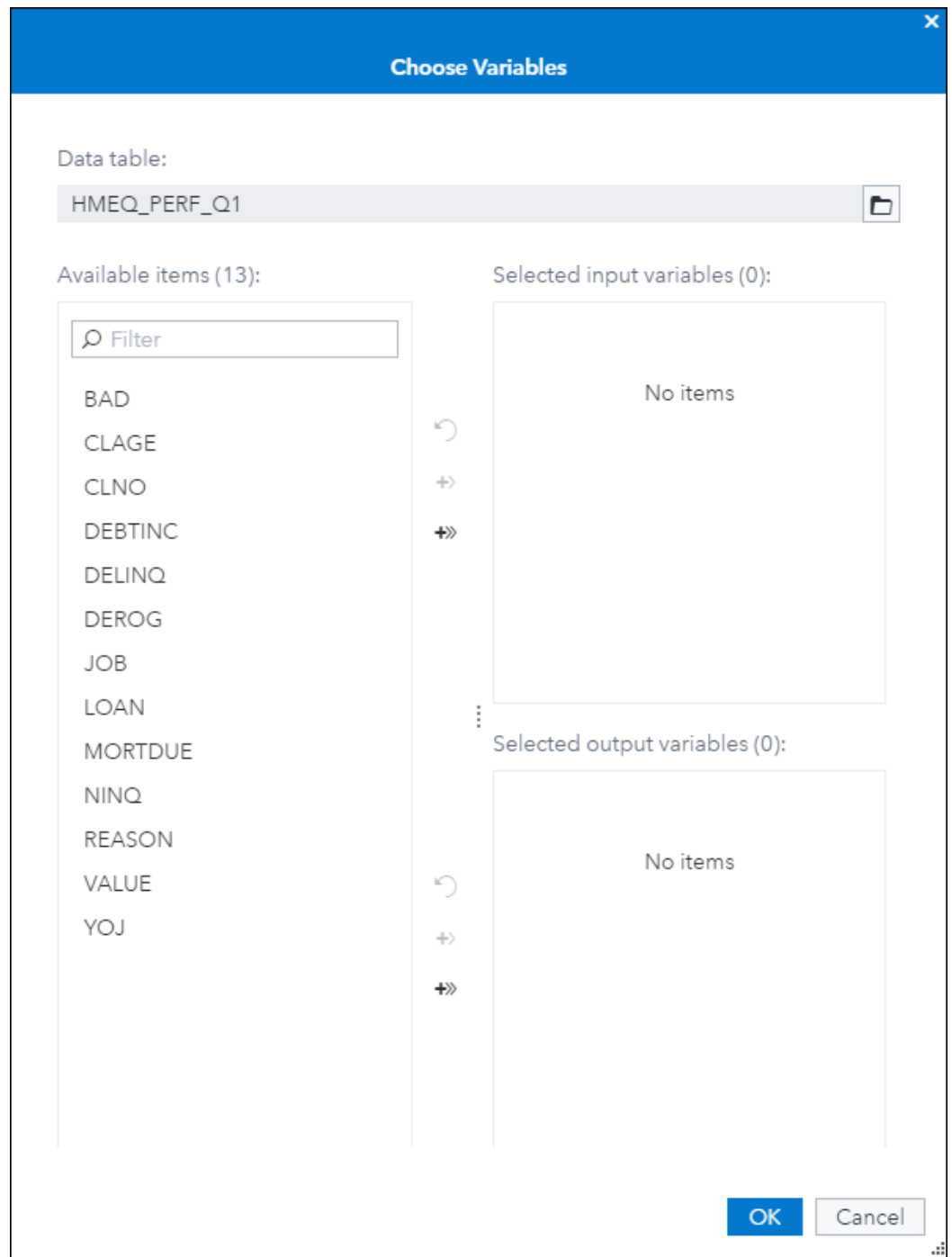
- **Date created (range)**
 - **Modeler**
- 4 To search for a project by **Target variable**, enter a value or click  to select a variable from a data table.
 - 5 To search for a model by input or output variables, select a data table and choose the variables that are associated with models or model objects.
 - a Click  next to the **Input variables** or **Output variables** field. The Choose Data window appears.
 - b Select a data table from the list on the left-hand side. The details of the selected data appear on the right-hand side. The sample data and available profiles of the data table are also viewable.

TIP Use the search bar to locate the desired data table and filter search results.



#	Name	Label	Type	Raw Len...	Format...	Format	Tags
1	BAD		double	8	12		
2	LOAN		double	8	12		
3	MORTDUE		double	8	12		
4	VALUE		double	8	12		
5	REASON		char	7	7		
6	JOB		char	7	7		
7	YOJ		double	8	12		
8	DEROG		double	8	12		
9	DELING		double	8	12		
10	CLAGE		double	8	12		
11	NING		double	8	12		
12	CLND		double	8	12		
13	DEBTINC		double	8	12		


- c When you have selected your data table, click **OK**. The Choose Variables window appears.




- d Select the variables that you want to add and click **+**. You can also click **+** to add all of the variables from the available items list.
- e To remove variables, select the variables that you want to remove and click **←**. Click **←** to remove all of the variables from the available items list.

TIP Click **↶** to undo your last action.


- f Click **OK** to close the Choose Variables window. The variables that you selected appear in the **Input variables** and **Output variables** fields.

TIP Click  next to the **Input variables** or **Output variables** fields in the Advanced Search for Models window to clear the selected variables.

- 6 To search for user-defined properties that are associated with model objects, specify one or more property name and value pairs, or specify property names only.
 - a Click **+** to add a row to the **User-defined properties** table.
 - b Click **▼** in the **Name** drop-down list, and select a user-defined property.
 - c (Optional) Enter the value for the user-defined property in the **Value** column.
 - d Select a row and click  to remove the user-defined property from the search criteria.
- 7 Click **Search**. The Models category view displays the list of models that meet the advanced search criteria.

TIP Click **Clear All** to clear all previously entered search criteria in the Advanced Search for Models window.


View Model Relationships and Related Objects

In the Models category view, if model relationships have been established, a **Relationships** token appears in the **Tags** column. Also, the  icon appears to the right of the model name.



Name	Role	Project (Version)	Model Function	Tags
QS_Reg1 		QS HMEQ, Version 1 (1.0)	Classification	Relationships (1) 

Here are some tasks that you can perform:

- Click  to view the model risk card for the related object.
- Click the **Relationships** token to view the model relationships.

For more information, see [“Establish Model Relationships” on page 67](#).

Comparing and Evaluating Models

<i>About Comparing and Evaluating Models</i>	80
<i>Compare Models</i>	80
<i>Test Models</i>	81
About Testing Models	81
Create and Run a New Test	82
Edit a Test	85
Delete a Test	85
<i>Validate Published Models</i>	85
About Validating Published Models	85
Edit and Run a Publishing Validation Test	86
Delete a Publishing Validation Test	87
Duplicate a Publishing Validation Test	88
<i>Set Champion and Challenger Models</i>	88
About Champion and Challenger Models	88
Set a Champion Model	89
Set a Challenger Model	89
Clear Model Role	90
<i>Monitoring Performance</i>	90
About Monitoring Performance	90
Set Project Properties	91
Naming Requirements for Input Data Tables	92
Create a New Performance Definition	93
Run Performance and View Results	96
Edit Performance Definition	100
Clear Performance Results	100
Edit Layout of Performance Results	100
Schedule Performance Monitoring	103
<i>Retrain a Project from Model Studio</i>	106

About Comparing and Evaluating Models

The goal of a modeling project is to identify a champion model that an external scoring application uses to predict an outcome. SAS Model Manager provides tools to evaluate candidate models and declare a project champion model.

Here are the tasks that you can perform to evaluate your models:

- compare and assess models
- create and run a scoring test on a model
- monitor performance of a model
- validate the published models within the destination
- access the published models from external scoring applications

Compare Models

You can compare and assess one or more models. When you compare models, the model comparison output includes model properties, user-defined properties, and variables. The model comparison output might also include fit statistics, and lift and ROC plots for the models if the required model files are available. The fit statistics, as well as plots for lift and ROC, can be produced using the ASSESS procedure or Python packages such as sklearn and scipy.stats. The fit statistics and plots are available by default for the SAS Visual Data Mining and Machine Learning models that are created using Model Studio. On the **Files** tab of a model object, notice that JSON files (dmcas_fitstat.json, dmcas_lift.json, dmcas_roc.json) are included. These JSON files are used to show the fit statistics and plots when comparing models in SAS Model Manager.

Note: You can create the JSON files that are needed to display the fit statistics as well as Lift and ROC plots. Here are some methods that you can use to create the JSON files: the ASSESS procedure, the %MM_MODEL_ADD_JSONFILES macro, or Python code and packages such as sklearn and scipy.stats.

For more information, see the following documentation and examples:

- [The ASSESS Procedure](#) in *SAS Visual Statistics: Procedures*
- ["%MM_MODEL_ADD_JSONFILES Macro"](#) in *SAS Model Manager: Macro Reference*
- Examples in the [model-management-resources](#) GitHub repository

To compare models:

- 1 Select one or more models.

- 2 From the Models category view, click  and select **Compare**.

On the **Models** tab of project, click **Compare**.

The **Compare** page appears. The default is to **Show All** of the comparison model content.

- 3 Click **Show Differences**.

- 4 Review the differences for the following model information:



.....
Note: Section titles appear whether or not the section contains content.

- Model Properties
- User-Defined Properties
- Input Variables
- Output Variables
- Target Variable
- Fit Statistics

.....
Note: This section compares the fitness statistics for the chosen models.

- Plots

.....
Note: This section contains both Lift and ROC plots.

TIP Click  to expand a section, and click  to collapse a section.

- 5 Click **Close**.

Test Models

About Testing Models

The purpose of a test is to run the score code of a model and produce scoring results that you can use for scoring accuracy and performance analysis. The test uses the input data table to generate the test output table. If your environment has its own means of executing the score code, then your use of the SAS Model Manager scoring tests is mostly limited to testing the score code. Otherwise, you can use the tests both to test your score code and execute it in a production environment.

To be scored, tested, or validated, a model must have a file that is assigned to the score code model file role and have the following score code types: DATA step, SAS program, DS2 package, DS2 embedded process, DS2 multi-type, Analytic store, Python, or R. For more information, see [“Assign Model File Roles” on page 61](#) and [“Set Model General Properties” on page 64](#).

When you publish a model, the system creates a publishing validation test. You can edit the publishing validation test to select a test data table and output library. This enables you to score and validate models within the publishing destination that they were previously published to. For more information, see [“Validate Published Models” on page 85](#).

Note: Models that have a score code type of Python or R can be scored, tested, or validated only when the score code is in the correct format. For more information, see [“Scoring Python Models”](#) and [“Scoring R Models”](#).

Create and Run a New Test

By default, only the user who creates a test definition can view, update, or delete the test definition, as well as run the test and view the test results. For more information, see [“Default Permissions” in SAS Model Manager: Administrator’s Guide](#).

Note: If you have one or more tests, you can select the check box and click **Run**, to run them all at the same time.

- 1 On the **Scoring** tab of a project, click the **Tests** tab, and then click **New Test**. The New Test window appears.

New Test

Name: *

Description:

Model:

Input table: *

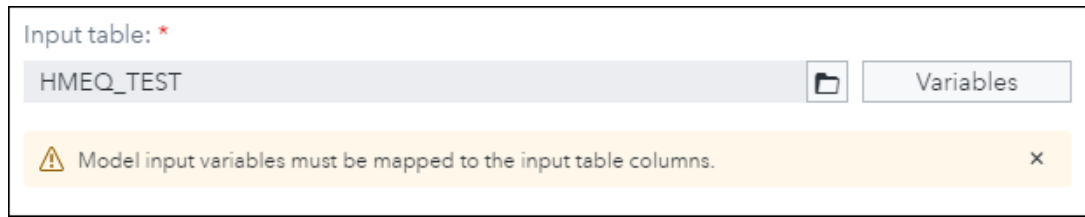
Output data library: *

- 2 Enter a name for the test if you do not want to use the default name.
- 3 (Optional) Enter a description for the test.
- 4 Click **Choose Model** and select a model to test.
- 5 Click , select the input table for the test, and click **OK**.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see [“Working with SAS Data Explorer”](#) in *SAS Data Explorer: User’s Guide*.


- 6 Map variables.

Note: SAS Model Manager automatically maps model input variables to the columns in the input table when the names and data types of the variables match those of the table columns. If any input variables cannot be mapped automatically, a warning message is displayed. When you are creating a scoring test for a model with the score code type of SAS program, if the column names in the input table do not match the model input variable names, a warning message is displayed and the **Variables** button is disabled. The exception is input variables for SAS Visual Text Analytics models with a score code type of SAS program. Those can be mapped.



TIP You can change the automatic variable mappings.







To map variables:


- a Click **Variables**. The Variable Mappings window appears.
 - b For each input variable, select the table column to which the variable should be mapped.
 - c Click **OK**.
- 7 (Optional) By default, the library for the output data is the same as that for the input data table. Click  to specify a different library to store the new test output table that is created when the test is run. The input data table and output library must be on the same CAS server.
- 8 Click **Run** to save and run the test. Alternatively, click **Save** to save the test definition without running it.

Note: You can also select a check box for a test and click **Run** in the toolbar to rerun a test.

The status of the test is indicated by the icon in the **Status** column.

Table 6.1 Test Statuses

Icon	Status
	The test is not ready to run. The test definition is not complete, or it might contain errors.
	The test is defined correctly and is ready to run.
	The test is running.
	The test completed successfully.
	The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page.
	The test did not run successfully. Check the SAS log for information. The URI to the log file is shown on the Test Results page.

- 9 Click  in the **Results** column to view the results of the test. The **Output** page is displayed by default. The **Test Results** page displays information about the

test, including the URIs for the test definition and test results. It also includes URIs to the SAS code that was run by SAS Model Manager, the output data set, and the SAS log that was generated when the code was run.

You can also click the **Code** or **Log** pages to view the test result details.


You can also work with the output table in other SAS applications to analyze the data, create and compare models, discover relationships hidden in the data, and generate reports based on the data.

Edit a Test

- 1 On the **Scoring** tab of a project, click the **Tests** tab.
- 2 Click on a test name. The Edit Test window appears.
- 3 Edit the test properties as needed, and then click **Save** or **Run**.

Note: You can also select a check box for a test and click **Run** in the toolbar to rerun a test.

Delete a Test

- 1 On the **Scoring** tab of a project, click the **Tests** tab.
- 2 Select one or more tests and click .

Validate Published Models

About Validating Published Models

When you publish a model, the system creates a publishing validation test. You can edit the publishing validation test to select a test data table and output library. The validation of the model runs within the publishing destination that the model was previously published to. The test data table must be loaded in CAS memory in order to run a publishing validation test.

Note: A Git publishing destination is a repository and it has no means of executing code. Therefore, when a model is published to a Git destination, no publishing validation test is created.

Only these models can be validated:

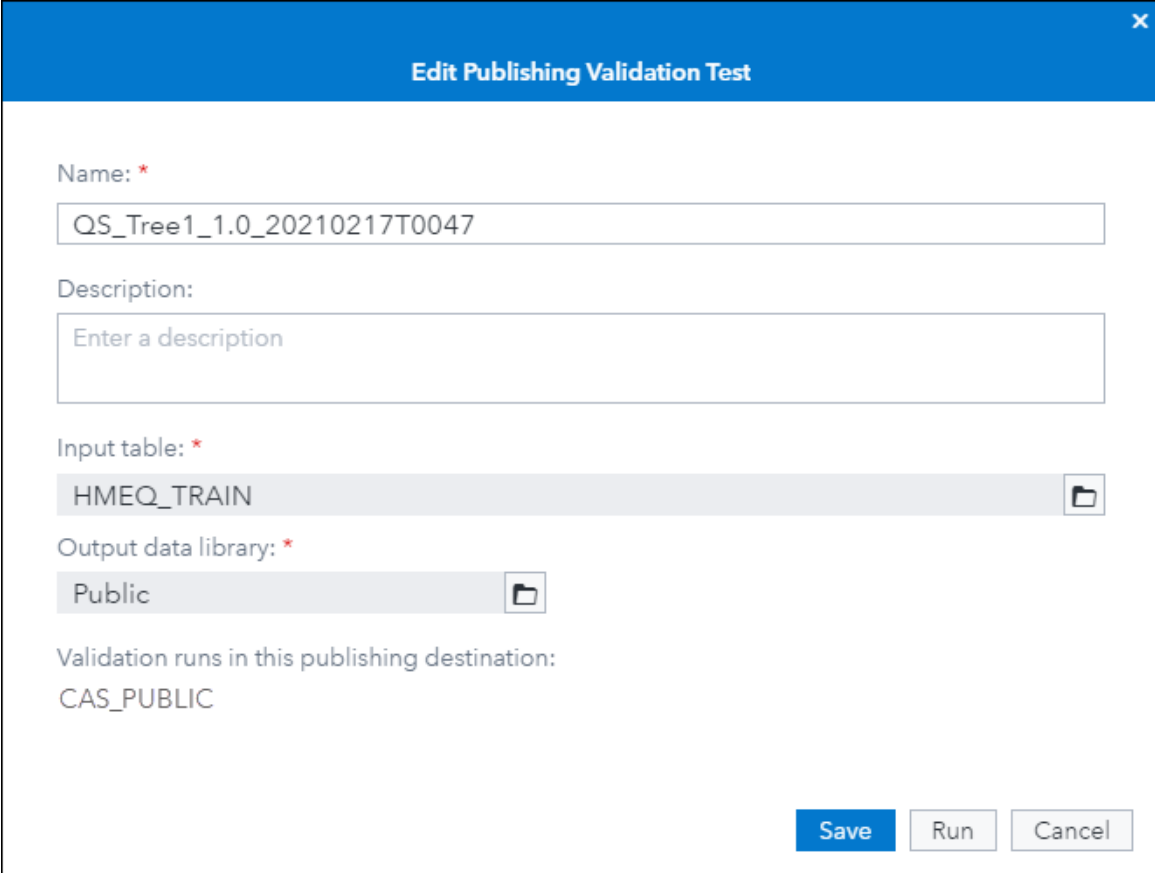
- those that have the score code model file role assigned
- those that have a score code type of DATA step, SAS program, DS2 package, DS2 embedded process, DS2 multi-type, Analytic store, Python, and R.

For more information, see [“Assign Model File Roles” on page 61](#) and [“Set Model General Properties” on page 64](#).

Note: Models that have a score code type of Python or R can be validated if the score code is in the correct format. For more information, see [“Scoring Python Models” on page 136](#) and [“Scoring R Models” on page 138](#).

Edit and Run a Publishing Validation Test


- 1 Click the **Scoring** tab of a project, and then click the **Publishing Validation** tab.
- 2 Click a test name. The Edit Publishing Validation Test window appears.




Edit Publishing Validation Test

Name: *


Description:

Input table: *
 


Output data library: *
 

Validation runs in this publishing destination:
 CAS_PUBLIC

Save


- 3 (Optional) Change the name of the test.
- 4 (Optional) Enter a description for the test.
- 5 Click , select the input table for the test, and click **OK**.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see [“Working with SAS Data Explorer” in SAS Data Explorer: User’s Guide](#).

- 6 Specify a value for any properties that might appear for the publishing destination and are required for validation. For example, if you select an Azure Machine Learning destination, a property appears for you to select a compute resource.
- 7 (Optional) By default, the library location is the same as that for the input data table. Click  to specify a different library to store the new test output table that is created when the test is run. The input data table and output library must be on the same CAS server.
- 8 Click **Run** to run the test. Alternatively, click **Save** to save the test definition without running it.

Note: You can also select a check box for a test and click **Run** in the toolbar to rerun a test.

The status of the test is indicated by the icon in the **Status** column. For more information, see [Table 6.1 on page 84](#).


- 9 Click  in the **Results** column to view the results of the test. The **Test Results** page displays information about the test, including the URIs for the test definition and test results. It also includes URIs to the SAS code that was run by SAS Model Manager, the output data set, and the SAS log that was generated when the code was run.

You can click the **Output**, **Code**, or **Log** pages to view the test result details.


You can also work with the output table in other SAS applications to analyze the data, create and compare models, discover relationships hidden in the data, and generate reports based on the data.

Note: When a publishing validation test is run for a model that was published to an Azure Machine Learning destination, the published model is deployed to Azure Kubernetes Service (AKS) in order to perform the validation. After validation is complete, the published model is removed from AKS. For more information about deploying models, see [“Deploy a Published Model to Azure Kubernetes Service \(AKS\)” on page 118](#).

Delete a Publishing Validation Test

- 1 On the **Scoring** tab of a project, click the **Publishing Validation** tab.
- 2 Select one or more publishing validation tests, click , and select **Delete**.
- 3 In the confirmation message, click **Delete**.

Duplicate a Publishing Validation Test

- 1 On the **Scoring** tab of a project, click the **Publishing Validation** tab.
- 2 Select one of the publishing validation tests, click , and select **Duplicate**.

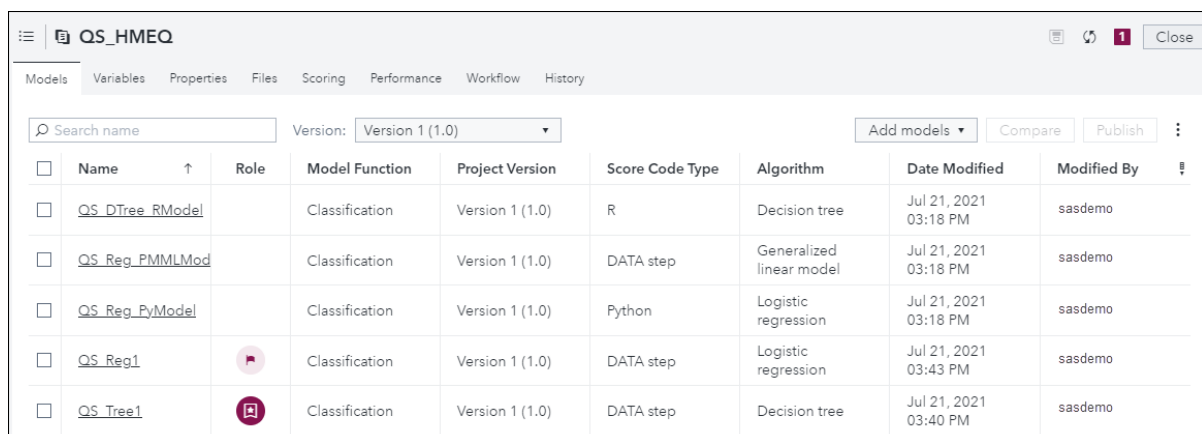
Set Champion and Challenger Models



About Champion and Challenger Models

The champion model is the best predictive model that is chosen from a pool of candidate models. Before you identify the champion model, you can evaluate the structure, performance, and resilience of candidate models. When a champion model is ready for production scoring, you set the model as the champion model. The project version that contains the champion model becomes the champion version for the project. You can publish the champion model to a publishing destination.


You use challenger models to test the strength of champion models. The champion model for a project can have one or more challenger models. A model can be flagged as a challenger model only after a champion model for the project has been selected. A challenger model can be located in any version of a project.

Note: A SAS Visual Data and Machine Learning project champion model is automatically set as a champion within a SAS Model Manager project when you register the models from a Model Studio project, as long as a project champion does not already exist. If a project champion already exists, the new Model Studio project champion is set as a candidate champion.



<input type="checkbox"/>	Name	Role	Model Function	Project Version	Score Code Type	Algorithm	Date Modified	Modified By
<input type="checkbox"/>	QS_DTree_RModel		Classification	Version 1 (1.0)	R	Decision tree	Jul 21, 2021 03:18 PM	sasdemo
<input type="checkbox"/>	QS_Reg_PMMLMod		Classification	Version 1 (1.0)	DATA step	Generalized linear model	Jul 21, 2021 03:18 PM	sasdemo
<input type="checkbox"/>	QS_Reg_PyModel		Classification	Version 1 (1.0)	Python	Logistic regression	Jul 21, 2021 03:18 PM	sasdemo
<input type="checkbox"/>	QS_Reg1		Classification	Version 1 (1.0)	DATA step	Logistic regression	Jul 21, 2021 03:43 PM	sasdemo
<input type="checkbox"/>	QS_Tree1		Classification	Version 1 (1.0)	DATA step	Decision tree	Jul 21, 2021 03:40 PM	sasdemo

Set a Champion Model

- 1 Click the **Models** tab of a project.
- 2 Select a model, click , and select **Set as champion**.
- 3 If the Select Project Output Variables window appears, select the model output variables to use as project level output variables. You can use the same variable names or specify different names for the project output variables.

.....

Note: You can modify the names of the project output variables only for a model with a score code type of DATA step, SAS program, and PMML.

.....


Click **Save**.

- 4 If the model input variables are not project input variables, you are prompted to add the input variables to the project.
In the confirmation message, click **Yes**.


.....

Note: If you click **No**, the model is not set as the project champion.

.....

- 5 When the champion model has been set,  is displayed in the **Role** column.

Set a Challenger Model

- 1 Click the **Models** tab of a project.
- 2 Select a model, click , and select **Set as challenger**.
- 3 If the **Select Project Output Variables** window appears, select the model output variables to use as project level output variables. You can use the same variable names or specify different names for the project output variables.

.....

Note: You can modify the names of the project output variables only for a model with a score code type of DATA step, SAS program, and PMML.

.....


Click **Save**.

- 4 If the model input variables are not project input variables, you are prompted to add the input variables to the project.
In the confirmation message, click **Yes**.


.....

Note: If you click **No**, the model is not set as a challenger model.

.....

- 5 When the challenger model has been set,  is displayed in the **Role** column.

Clear Model Role

- 1 Click the **Models** tab of a project.
- 2 Select the champion model or a challenger model, click , and select **Clear role**.

.....
Note: If you clear the role of the project champion model, all of the associated challenger models are cleared as well. The candidate champion role cannot be cleared.
.....

- 3 In the confirmation message, click **Yes**.

Monitoring Performance

About Monitoring Performance

Models can begin to degrade the moment that you put them in production. Using SAS Model Manager, you can monitor the performance of your analytical models to see whether they continue to behave as expected after changes occur in market conditions or customer behavior, new data becomes available, or there is concept drift. By monitoring their performance, you can avoid model decay and re-evaluate the business value of your models in production to keep them performing at the highest levels. Comparing and assessing model performance over time is a best practice that preserves the value that your data science and IT teams originally provided.

SAS Model Manager enables you to monitor and evaluate model performance of your champion, challenger, and other models to ensure that your models are performing efficiently. You begin by collecting performance data that has been created by the champion model at intervals. Your organization determines what those intervals are. You then use performance data to assess model prediction accuracy. It includes all of the required variables as well as a target variable. For example, you might want to create performance data tables monthly or quarterly and then use SAS Model Manager to create a performance definition that includes each time interval.

Next, you create a performance definition, and then you run it. The output from running a performance definition includes a summary of performance and model assessment information, performance results tables, and several charts for each model that is specified within the definition. The summary information includes model assessment criteria, scoring observations, performance definition high-level details, and the performance rank of models within the containing project. The results include charts such as Variable Distribution, Characteristic, Stability, Lift,

Gini, ROC, Kolmogorov-Smirnov (KS), Average Squared Error (ASE), and Feature Contribution (FCI) charts for each model that was specified within the definition. Additional charts are displayed for prediction models with an interval target, and custom charts can be configured to be displayed as well. Only the charts for each model are displayed on the **Performance** tab of a project. You can explore and visualize the performance results tables using SAS Visual Analytics.

Note that model performance can sometimes be improved by tuning or refitting your models, or by using a new champion model.

You can also conduct partial model performance monitoring based on the input data. If your input data contains only the input variables, then the system computes and returns the characteristic analysis. If your input data contains the score output variable, then the system also computes and returns the stability analysis. If your input data contains the response variables, then the system also computes and returns the accuracy measures, such as Gini, ROC, Lift, Kolmogorov-Smirnov (KS), Average Squared Error (ASE), and Feature Contribution Index (FCI). The charts that are displayed depend on the model function and whether there is a binary or interval target variable defined within a project. Additional charts are displayed for a prediction model with an interval target.

Alternatively, you can also create performance monitoring output by writing your own SAS program using the performance monitoring macros that are provided with SAS Model Manager. You can then submit your program using SAS Studio. The performance results tables that are produced using the macros can then be viewed in SAS Studio, SAS Environment Manager, or SAS Visual Analytics. For more information, see [“Performance Monitoring Macro” in SAS Model Manager: Macro Reference](#).

Note: When performance monitoring is run, the system can score models only with the following score code types: **SAS program**, **DATA step**, **DS2 embedded process**, **DS2 multi-type**, **Analytic store**, **Python**, or **R**. For more information, see [“Set Model General Properties” on page 64](#).

For more information and examples of the performance monitoring charts, see [“Concepts: Performance Monitoring” on page 139](#).

Set Project Properties

- 1 Click the **Properties** tab of a project.
- 2 On the **General** page, specify the following properties:

Classification project

- Target variable
- Target event value
- Target level
- Output event probability variable

Prediction project

- Target variable
- Target level
- Output prediction variable

3 Click .

Naming Requirements for Input Data Tables

Rules for Data Table Names

Here are rules for data table names:

- They must have at least two levels.
- The value for the *prefix* can contain underscores, but spaces are not recommended in the prefix name or the table name.
- The name can contain alphanumeric characters, double-byte characters, and special characters, except for the following characters: / \ , ; ' "
- The second level must be a number.
- An underscore is treated as a delimiter between the levels within the name.
- The sequence number and time label must be unique across all of the data table names.
- The time label is used to represent the data in the charts. If a time label is not provided, the sequence number is displayed in the charts.

Supported Formats

Use one of the following formats for the name of the data tables:

- *prefix_sequenceNumber*
- *prefix_sequenceNumber_timeLabel*

Descriptions and Restrictions

Here are the descriptions and restrictions for the different levels of a table name:

prefix

The prefix is the first level of the table name and specifies which data tables in a data library to use for performance. The prefix can contain underscores, but spaces are not recommended in the prefix name.

sequence number

The sequence number is the second level of the table name and specifies the order in which the data tables should be used for performance monitoring. The sequence number must be the second level of the table name. The sequence number must be unique across all of the data table names. It is recommended that you start with the number 1 and increase from there.

time label

The time label is the third level of the table name and specifies the label to use to represent the data in the performance charts. The time label must be unique across all of the data table names. The label can be a period of time such as Q1 or Q12020, or another meaningful label. Spaces are not recommended in the label name. If a time label is not provided, the sequence number is displayed in the charts.

Create a New Performance Definition

- 1 Click the **Performance** tab of a project and then click **New Definition**. The New Performance Definition window appears.
- 2 On the **General** page, accept the default name for the performance definition or enter a name of your choosing.
- 3 (Optional) Enter a description.

- 4 Click the **Tables** page and select an input data method.


IMPORTANT Before selecting an input data method, make sure that the input data tables follow the naming requirements. For more information, see [“Naming Requirements for Input Data Tables”](#) on page 92.

Use a single table

Click , select a data table and click **OK**.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see “Working with SAS Data Explorer” in *SAS Data Explorer: User’s Guide*.

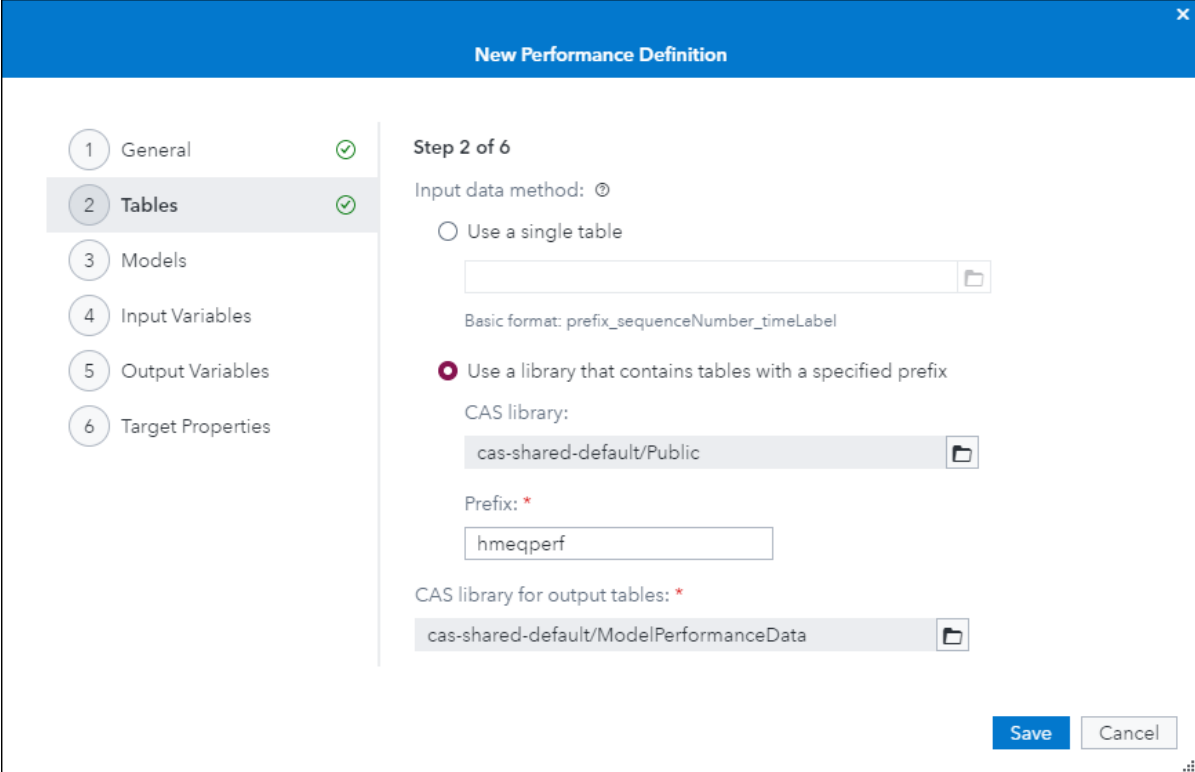
Use a library that contains tables with a specified prefix

- Click , select the same server as the data table, and select a library. Click **OK**.
- Enter a prefix.

Note: The value for the *prefix* can contain hyphens and underscores, but spaces are not recommended in the prefix name or the table name.


Note: If the data tables that were specified in the performance definition exist in a library, but are not loaded in a CAS session, they are loaded at run time. This condition might occur when the CAS server is restarted.

- 5 (Optional) Verify that the CAS library for the output tables is set. The default value is `cas-shared-default/ModelPerformanceData`. The input data source library and the output data library must be on the same CAS server.



New Performance Definition

Step 2 of 6

Input data method: 

Use a single table

Use a library that contains tables with a specified prefix

CAS library:
cas-shared-default/Public

Prefix: *
hmeqperf

CAS library for output tables: *
cas-shared-default/ModelPerformanceData

Save Cancel

- 6 Click the **Models** page and select an option for the data scoring method.

Note: Partial save is supported. After you complete this step of the wizard, you can click **Save**.

System scores data using selected models

When you choose this data method, the model score code is used to score the data before generating the performance results.

Use scored data from input tables

When you choose this data method, the data tables should contain the predicted values for the scored model.

- 7 Select a model-selection option.

TIP If model roles or selected models are changed after a performance definition is saved, you must edit the performance definition and save it, in order for the changes to take effect.

Use referenced models

The default option to use the current champion. You can also select to include all challenger models.

Select specific models from this project

This option enables you to select one or more models to include in your performance definition for model analysis.

New Performance Definition ✕

1 General ✔

2 Tables ✔

3 **Models** ✔

4 Input Variables

5 Output Variables

6 Target Properties

Step 3 of 6

Data scoring method: ⊙

System scores data using selected models

Use scored data from input tables

Model-selection options: *

Use referenced models ⊙

Current champion: QS_Tree1 (1.1)

All challengers: QS_Reg1 (1.0)

Select specific models from this project

Name	Role	Project Version	Date Modified
QS_Reg1 (1.0)	⊙	Version 1 (1.0)	Jul 21, 2020 12:06 PM
QS_Reg_PMMLM odel (1.0)		Version 1 (1.0)	Jul 21, 2020 12:01 PM
QS_Reg_PyModel (1.0)		Version 1 (1.0)	Jul 21, 2020 12:01 PM
...	⊙	...	Jul 21, 2020

0 of 4 items selected

Save
Cancel

- 8 Click the **Input Variables** page to view the input variables to include in the report results. Select the project input variables that you want to use for analysis when generating the performance results.

TIP By default, all of the project input variables are selected when you are creating a new definition.

- 9 Click the **Output Variables** page to view the output variables to include in the report results. Select the project output variables that you want to use for analysis when generating the performance results.

TIP By default, all of the project output variables are selected when you are creating a new definition.

- 10 (Optional) Click the **Target Properties** page to view the properties that are being used to compute stability and accuracy measures to assess model prediction accuracy.

TIP You can modify the values of the target properties on the **Properties** tab of this project. If the values are changed after a performance definition is created, you must edit the performance definition and save it, in order for the changes to take effect.

- 11 Click **Save**. You are returned to the **Performance** tab.

Run Performance and View Results

Note: Data tables are loaded at run time, if necessary, when they are referenced within a performance definition and are not available in a CAS session.

- 1 On the **Performance** tab, click **Run** and select **Run now**.

TIP You can also schedule performance jobs to run at a specific frequency such as daily, weekly, monthly, or quarterly. Other types of frequencies are also available. For more information, see [“Schedule Performance Monitoring” on page 103](#).

- 2 Click **:** and select **View job history** to view the current status of the performance job.

Job Name	Input Library	Table Prefix	Model	Data Table	Output Library	Status	Date Started	Date Completed	Submitted By	Log	Code
QS_HMEQ_Performance	Public	hmqperf	QS_Reg1 (1.0)		ModelPerformanceData	🟢	Oct 15, 2021 02:46 PM	Oct 15, 2021 02:50 PM	sasdemo	📄	🔗
QS_HMEQ_Performance	Public	hmqperf	QS_Tree1 (2.0)		ModelPerformanceData	🟢	Oct 15, 2021 02:46 PM	Oct 15, 2021 02:50 PM	sasdemo	📄	🔗

- When the job is complete, click **Close** to return to the **Performance** tab and view the results.

The **Summary** page contains assessment information about the models that are included in the project's performance results. It also includes scoring observations, performance definition high-level details, and the performance rank of models within the containing project.

Note:

The model performance rank is based on the specified model assessment criteria in the project's properties. If the assessment criteria are not specified or user-defined criteria is specified in the project's properties, the misclassification KPI is used to determine the rank of models in a project with a classification model function. Likewise, the average squared error (ASE) KPI is used to determine the rank of models in a project with a prediction model function.

The screenshot displays the 'Performance' tab for the 'QS_HMEQ' project. The main content area is titled 'QS_HMEQ_Performance' and includes a 'Summary' tab. The summary is divided into three main sections:

- Scored Observations:** A large blue number '23,840'.
- Definition Details:** A table with the following information:

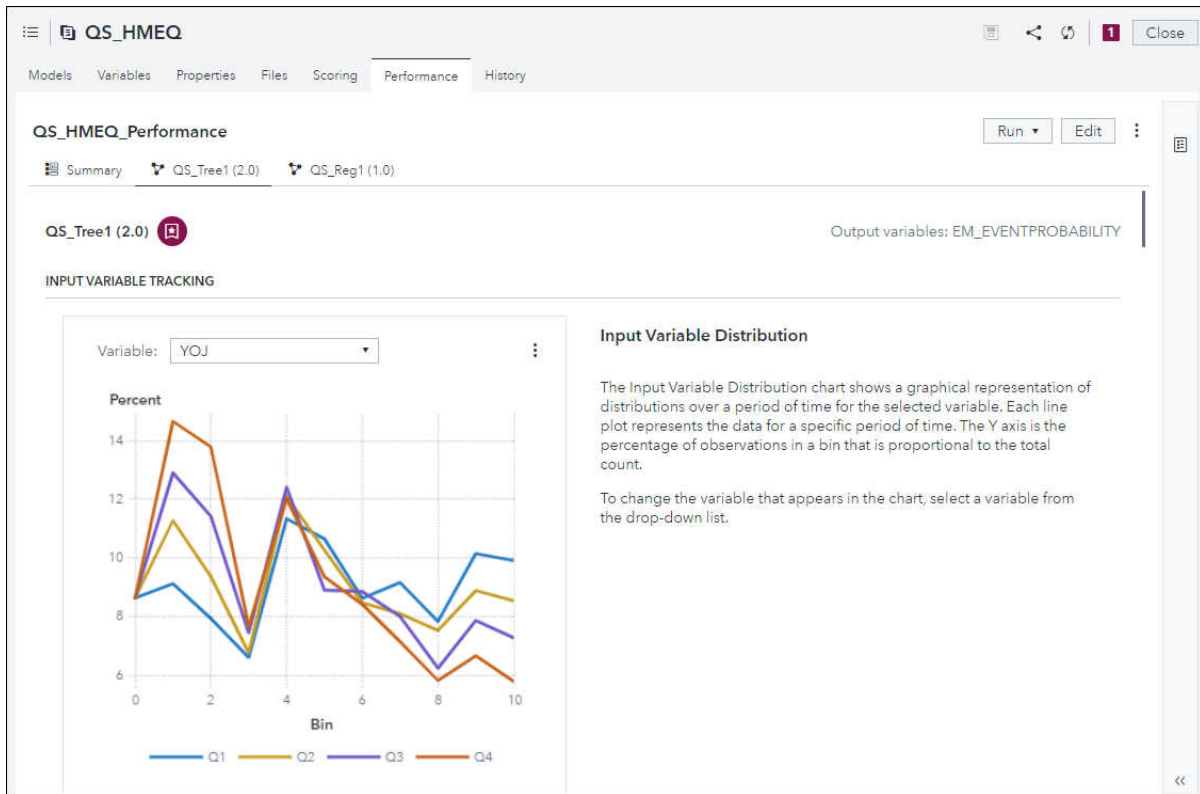
Created by:	sasdemo
Last updated:	Oct 15, 2021 02:50 PM
Models:	QS_Tree1 (2.0) QS_Reg1 (1.0)
- Model Performance Rank:** A table showing the ranking of models:

1	QS_Reg1
2	QS_Tree1

Below these sections are two model-specific performance cards:

- QS_Reg1 (1.0):**
 - Misclassification: 27.11%
 - False positive rate: 23.49%
 - Recall: 43.95%
 - Last updated: Oct 15, 2021 11:48 AM
 - Modified by: sasdemo
 - Created by: sasdemo
 - Score code type: DATA step
- QS_Tree1 (2.0):**
 - Misclassification: 22.23%
 - False positive rate: 13.76%
 - Recall: 51.21%
 - Last updated: Oct 15, 2021 11:53 AM
 - Modified by: sasdemo
 - Created by: sasdemo
 - Score code type: DATA step

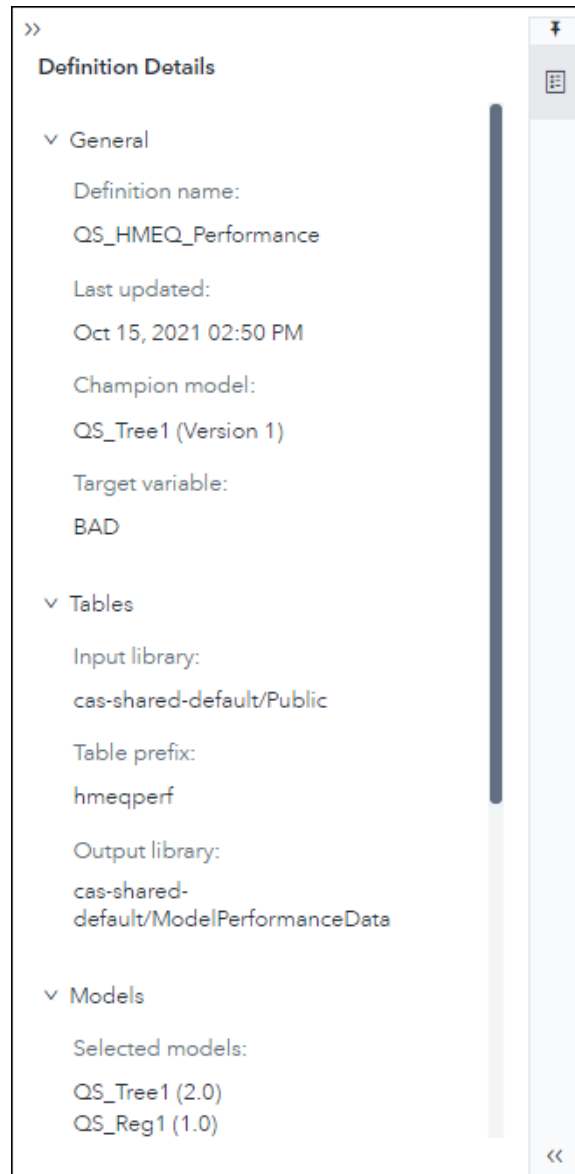
- Click a model object tab to view the performance results for that specific model.




Note:

For the Input Variable Distribution and Feature Contribution charts, you can select variables from the drop-down list to change the content of the chart. This is an example of the YOJ variable distribution.


- b** Click  to view the performance definition details.



Note: You can conduct partial model performance monitoring based on the input data. If your input data contains only the input variables, then the system computes only the characteristic analysis. If your input data contains the score output variable, then the system also computes the stability analysis. If your input data contains the response variables, then the system also computes the accuracy measures, such as Gini, ROC, Lift, Kolmogorov-Smirnov (KS), Average Squared Error (ASE), and Feature Contribution Index (FCI). Additional charts are available for running performance for a prediction model with an interval target. For more information, see [“Model Monitoring” on page 143](#).

- 4 (Optional) Click  and select **Explore and Visualize**.

Note: SAS Visual Analytics opens within the same browser window, and the performance data for the associated chart is shown. For more information, see [SAS Visual Analytics: Working with Report Content](#).

To return to your project, click  and select **Manage Models**.

- 5 Click **Close**.

Edit Performance Definition

Here are some reasons to edit the performance definition content:

- The model role has changed or the model you previously selected is no longer available.
- The project properties or variables have changed.


To edit the definition:

- 1 Click **Edit**.
- 2 Modify the performance definition content.
- 3 Click **Save**.

For information about the definition content, see [“Create a New Performance Definition” on page 93](#).

Clear Performance Results


You can clear all of the performance results and delete the data tables from the CAS output library.

- 1 On the **Performance** tab, click  and select **Clear all**.
- 2 Click **Clear** in the confirmation message.

After the performance results are cleared and the data tables are deleted, an information message is displayed indicating that you can edit the definition. Otherwise, click **Run** and select **Run now** to generate the results for the existing definition content. See: [“Edit Performance Definition”](#).

Edit Layout of Performance Results

Edit Layout of Charts

- 1 On the **Performance** tab, click  and select **Edit layout** ⇨ **Model charts**.

Note: Currently, the layout of the **Summary** page cannot be modified and the **Comparison Charts** page is not available. Therefore, the submenu options are disabled.

✕
Edit Layout of Model Charts

Select the charts that you want to display on the Performance tab. You can drag and drop the items in the list to rearrange the order in which you want them to appear.

Input Variable Tracking
Output Variable Tracking

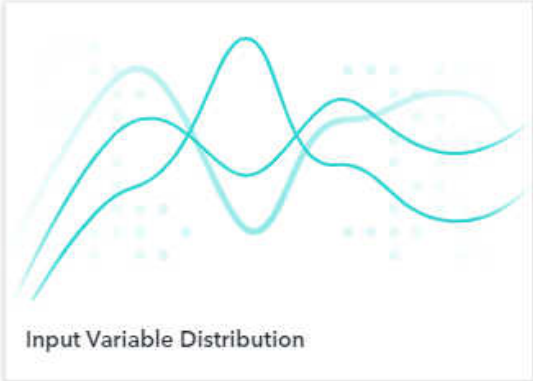
Charts

+ Add Custom Chart

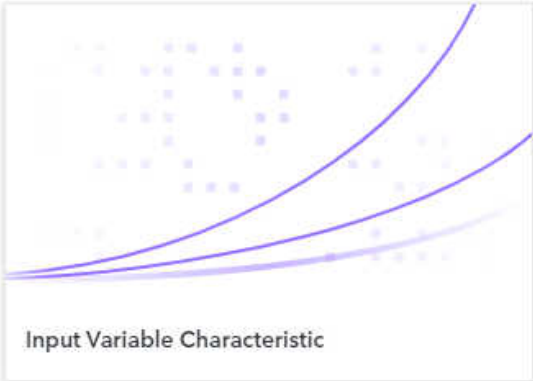
Input Variable Distribution

Input Variable Characteristic ⓘ

Preview



Input Variable Distribution





Input Variable Characteristic

Save
Cancel

- 2 Select or deselect input and output variable tracking charts to be displayed on the **Performance** tab.
- 3 Arrange the charts to appear in the order that you want by using drag and drop in the **Charts** panel.



TIP Click the Help icon ⓘ next to a chart to view information about that specific chart in the **Charts** panel.

Add Custom Charts

- 1 Copy an object link from a report in SAS Visual Analytics.
 - a Click  and select **Explore and Visualize**.
 - b Open a report.
 - c Click  on the object toolbar, and then select **Copy link**. The Copy Link window is displayed.
 - d Select these options:
 - **Interactive report**
 - **Embeddable web component**
 - e Click **Copy Link** and paste the link into a text file so that you can use the link when creating a custom chart in SAS Model Manager.

Note: The SAS Visual Analytics report object that you want to link to must be on the same host as SAS Model Manager.

For more information, see “[Copy a Link to an Object](#)” in *SAS Visual Analytics: Designing Reports*.

- 2 Click  and select **Manage Models** to return to SAS Model Manager.
- 3 On the **Performance** tab of your project, click  and select **Edit layout** ⇒ **Model charts**. The Edit Layout of Model Charts window appears.
- 4 Click **+ Add Custom Chart**. The Add Custom Chart window appears.

✕

Add Custom Chart

Specify a name for the chart, and the link to the report object, to use for the custom tile.

Name: *

Help description: ⓘ

Report object link: * ⓘ

- 5 Paste the object link in the **Report object link** box.
- 6 Enter a name for the new custom chart.
- 7 Enter a Help description to provide information about your custom chart.
- 8 Click **Add**.

Schedule Performance Monitoring

After you create a performance definition on the **Performance** tab of a project, you can schedule the performance job to run at a specific frequency such as daily, weekly, monthly, or quarterly. Other types of frequencies are also available.


- 1 Click **Run** and select **Schedule for later**. The Schedule Job window appears.

- 2 Accept the default name for the job schedule or specify your own name.
- 3 Select an option from the **Frequency** drop-down list to specify how often to trigger the job to run (such as a specified number of minutes, hours, or days).
- 4 Depending on your choice for the frequency interval, different fields appear in the window to enable you to specify a frequency for running the job. For example, if you select **Yearly** in the **Frequency** field, you can specify a day of a month (such as the first of January), the last day of a month, or a specific weekday in a month (such as the third Thursday in February). If you specify **Minutes** in the **Frequency** field, you can specify that the job runs every 5, 10, 15, 20, or 30 minutes. Use these fields to specify the criteria for the trigger interval.

Note: If you select **Date List** in the **Frequency** field, you cannot select a date more than once.

- 5 In the **Start time** field, specify when the job schedule should start. Click the entry in the **Start time** field to select a time. Times are specified in 24-hour format.
For example, if you use the **Frequency** fields to specify that the job runs every hour, and you specify **10:15** in the **Start time** field, the job runs at 10:15, 11:15, 12:15, and so on. If you use the **Frequency** fields to specify that the job runs every 20 minutes, and you specify **09** in the **Start time** field, the job runs at 9:00, 9:20, 9:40, and so on.
- 6 Specify the time zone to use when evaluating the time for the job schedule and the date on which it starts.

Note: If you select **Date List** in the **Frequency** field, you must select the same value in the **Time zone** field for every scheduled date.


- 7 Click  and select a start date.
- 8 Specify when the job schedule ends. You can specify that the job schedule never ends, that it ends after a certain number of times, or that it ends on a specific date.



Schedule Job ✕


Job schedule name: *

Frequency: Interval: months

Occurrence:

Time: * 


Time zone:  

Start date: 

End:

- 9 Click **Save**.

TIP Only one job schedule can exist for each project performance definition. To view or edit an existing performance job schedule, click **Run Job** and select **Schedule for later**. Make your changes and click **Save**.

You can also edit or delete a job schedule using SAS Environment Manager. Additional triggers can be added to an existing job schedule only by using SAS Environment Manager. To access SAS Environment Manager, click  and select **Manage Environment**. For more information, see [“Schedule Jobs” in SAS Environment Manager: User’s Guide](#).


CAUTION

Do not delete job requests in SAS Environment Manager. If you delete a job request in SAS Environment Manager, the performance monitoring feature does not work as designed.

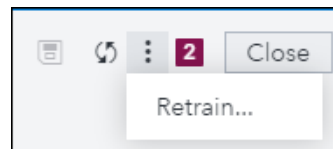
Retrain a Project from Model Studio



Only SAS Visual Data Mining and Machine Learning projects that have been registered from Model Studio into the SAS Model Manager common model repository can be retrained. Model studio projects and their models are stored in the `DMRepository` repository folder when they are registered.

To send a retrain request for a Model Studio project:

- 1 Click  to navigate to the Projects category view.
- 2 Open a project that was registered from Model Studio.

Note: If a project has been registered from Model Studio, an actions menu button appears to the left of the open items icon.



- 3 Click  and select **Retrain**.
- 4 Select an option to send a retrain request to Model Studio.
 - Select the **Set the project retrain state to needed** option if you want to indicate that the project and its models are ready for retrain.
 - Select the **Retrain now with a new data table** option, if you want to select a new data table and send a request to Model Studio to retrain the project now. Click  to choose a data table.

TIP If the desired data table is not available, you can import and load a table from the Choose Data window. Once loaded, the table is listed on the **Available** tab of the Choose Data window and can be selected as the data table for the project. For more information, see [“Working with SAS Data Explorer”](#) in *SAS Data Explorer: User’s Guide*.

Retrain Project

Select an option and send a request to the Model Studio application for retraining.

Set the project retrain state to needed

Retrain now with a new data table

cas-shared-default/Public/HMEQ_TRAIN

Send Cancel

5 Click **Send**.

Note: When you select the **Retrain now with a new data table** option, all pipelines in the associated Model Studio project are rerun. The previous pipeline results are overwritten. When the pipelines have finished running in Model Studio, a new champion model is selected during pipeline comparison. The new champion model is then auto-registered into a new project version within the associated SAS Model Manager project.

Publishing Models

<i>About Publishing Models</i>	109
See Also	110
<i>Publishing Requirements and Restrictions</i>	110
Model Score Code Types and Publishing Destinations	110
Published Name	112
<i>Publish a Project Champion Model</i>	113
<i>Publish Models from a Project Version</i>	115
<i>View Model Publish History</i>	117
<i>Deploy a Published Model to Azure Kubernetes Service (AKS)</i>	118

About Publishing Models

You can publish models to a publishing destination so that you can validate the models. You can also access published models from other applications to perform tasks such as scoring. Models can be published to destinations that are defined for CAS, Git repository, Hadoop, SAS Micro Analytic Service, and Teradata. They can also be published to an Azure Machine Learning cloud-based provider, and containers such as Amazon Web Services, Azure, Google Cloud Platform (GCP), and Private Docker. For information about defining publishing destinations, see [“Configuring Publishing Destinations” in SAS Model Manager: Administrator’s Guide](#).

Here are the different locations that models can be published from within the SAS Model Manager web application:

- The project champion model and its challengers can be published from the Projects category view.
- The **Models** tab of a project.
- An open model object, if the model is located within a project version.

Note: For information about what types of models are supported for publishing, see “[High-Level Model Support Matrix for Primary Functions](#)” on page 7 and “[Publishing Requirements and Restrictions](#)” on page 110.

The **SAS Micro Analytic Service (maslocal)** publishing destination is available by default. Other publishing destinations can be defined by a SAS administrator. A global caslib is required for CAS, Hadoop, and Teradata publishing destinations. For more information, see “[Configuring Publishing Destinations](#)” in *SAS Model Manager: Administrator’s Guide*.

Here are several things to keep in mind:

- When you publish a model from a project version, the system creates a publishing validation test.

Note: A Git publishing destination is a repository, and thus it has no means of executing code. Therefore, when a model is published to a Git destination, no publishing validation test is created.

- You can edit the publishing validation test to select a test data table and output library.
- Model validation runs within the publishing destination that the model was published to.

For more information, see “[Validate Published Models](#)” on page 85 and “[Schedule Jobs](#)” in *SAS Environment Manager: User’s Guide*.

See Also

- “[Publishing Python Models](#)”
- “[Publishing R Models](#)”

Publishing Requirements and Restrictions

Model Score Code Types and Publishing Destinations

Before you can publish a model, you must first set its score code type in the model’s properties and make sure that the score code file role is set. Only models with a score code type of DATA step, DS2 package, DS2 embedded process, DS2 multi-

type, Analytic store, Python, and R and that contain a valid score code file can be published.

Here are a few things to be aware of:

- The model score code type can affect which destinations you can publish your models to.
- A model with the score code type of DS2 multi-type can contain code files for a DS2 embedded process, a DS2 package, and one or more analytic stores. For example, when SAS Visual Data Mining and Machining Learning models that contain an analytic store are registered into the common model repository, their score code type is set to DS2 multi-type, instead of Analytic store.
- Models that have a score code type of Python or R can be published if the score code is in the correct format. For more information, see [“Scoring Python Models” on page 136](#) and [“Scoring R Models” on page 138](#).
- In order to validate models with a score code type of Python within a container publishing destination, you must first add a requirements.json file to the model before you publish it. The requirements.json file should include the install statements for the Python packages, which contain the modules that are used in the Python score code file and its score resource files. For more information, see [“Scoring Python Models” on page 136](#).
- Predictive Model Markup Language (PMML) is an XML-based predictive model interchange format. Some classification and prediction models that you create using PMML 4.2 are converted to DATA step score code during the model import process. When you are importing valid PMML models, the score code type model property is set to DATA step, instead of PMML. PMML models with a score code type of DATA step can be scored and published. For more information, see [“Import Models” on page 53](#) and [“Concepts: PMML Support” on page 155](#).

Here is a list of the model score code types and the types of destinations that you can publish your models to:

Note: The supported container publishing destinations are Amazon Web Services, Azure, Google Cloud Platform (GCP), and Private Docker.

Model Score Code Type	CAS	Git Repository	Hadoop	SAS Micro Analytic Service	Teradata	Azure Machine Learning	Container Destinations
Analytic store	No	Yes	No	No	No	No	Yes
DATA step	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DS2 embedded process	Yes	Yes	Yes	No	Yes	Yes	No
DS2 multi-type	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DS2 package	No	Yes	No	Yes	No	Yes	Yes

Model Score Code Type	CAS	Git Repository	Hadoop	SAS Micro Analytic Service	Teradata	Azure Machine Learning	Container Destinations
PMML	No	No	No	No	No	No	No
Python	Yes	Yes	No	Yes	No	Yes	Yes ¹
R	Yes	Yes	No	Yes	No	Yes	Yes ¹
SAS program	No	Yes	No	No	No	No	No

¹ SAS Viya models that contain Python or R open-source code cannot be published to a container destination.

Note: Here are the types of SAS models that are currently supported for publishing to container destinations:

- SAS Visual Data Mining and Machine Learning models that are registered from Model Studio, SAS Visual Analytics, or SAS Studio
- SAS Visual Statistics models that are registered from SAS Visual Analytics or SAS Studio
- DATA step models in SAS package files that are created using SAS Enterprise Miner and then imported into SAS Model Manager

For more information, see [“Set Model General Properties” on page 64](#).

Published Name



The maximum length and character restrictions for the published name differ, depending on your destination.

Table 7.1 Requirements and Restrictions for Published Names

Destination	Maximum Length	Requirements and Restrictions
Container destinations	128	<p>The published name must start with a letter or an underscore. It cannot contain spaces, multi-byte characters, or special characters other than the underscore.</p> <p>Note:</p> <ul style="list-style-type: none"> ■ The supported container destinations are Amazon Web Services, Azure, and Private Docker. ■ When publishing SAS models, the published name that you enter is prepended with a forward slash (/) and then is assigned to the SAS_SCR_APP_PATH environment variable. Therefore, the published name that you enter can be only 127 characters long. The value of this variable determines the module name. For more information, see “Changing the Endpoint Name for a

Destination	Maximum Length	Requirements and Restrictions
		Container” in <i>SAS Container Runtime: Programming and Administration Guide</i> .
Azure Machine Learning	128	The published name must start with a letter or an underscore. It cannot contain spaces, multi-byte characters, or special characters other than the underscore.
Git	128	The published name cannot contain colons (:) or double quotation marks.
SAS Micro Analytic Service	100	The published name cannot contain the following characters: ! @ # \$ % ^ & * () = ~ ` \ / . { } " ' ;
SAS Cloud Analytic Services (CAS)	128	The published name cannot contain single or double quotation marks.
Hadoop	128	The published name cannot contain colons (:) or double quotation marks.
Teradata	128	The published name must start with a letter or an underscore. It cannot contain spaces, multi-byte characters, or special characters other than the underscore.

Publish a Project Champion Model

- 1 Click  to navigate to the Projects category view.
- 2 Select a project, click , and select **Publish**. The **Publish Models** window appears.

Note: The project champion model and its challengers appear in the **Items to Publish** section.

- 3 Select a destination.

Note: If you have Read and Write permissions to the caslib that is specified in a publishing destination, the destination is shown in the list.


For more information, see [SAS Viya: CAS Authorization Window](#) and “Configuring Publishing Destinations” in *SAS Model Manager: Administrator’s Guide*.

TIP Click **Details** to view the destination details.

- 4 (Optional) Assign tags to the published models by entering a tag value and selecting an existing tag, or press Enter to add a new tag.

TIP Tag names cannot start with a period or a hyphen, and cannot contain more than 128 characters. They can contain only alphanumeric characters, the underscore (_), the hyphen (-), and the period (.).

- 5 Specify a value for any properties that might appear after you select a destination.

Property	Description	Destination Type	Required
Azure Machine Learning workspace	Indicates the workspace in the Azure Machine Learning environment to publish the selected items to. Select a value from the drop-down list.	Azure Machine Learning	Yes
Git directory	Indicates the directory path within the Git repository to publish the selected items to. Click  and navigate to a folder within the Git repository. Click OK . Note: If a folder is not selected within the Git repository, the items are published to the root directory.	Git	No

- 6 (Optional) In the **Items to Publish** section, edit the **Published name** if you do not want to use the default name for the published module. The maximum length and character restrictions differ, depending on your destination.

For more information, see [Table 7.1 on page 112](#).

- 7 (Optional) If you have previously published the project champion model or its challengers, enable the **Replace item with the same name** toggle in order to replace the previously published item of the same name in the same destination.

Note: When publishing models to a container, an Azure Machine Learning, or a Git publishing destination, if a published model with the same name already

exists within the destination, a sequential number is added as a tag to indicate the version of the published model.

- 8 Click **Publish**. The Publishing Results window appears. The status of the publishing request is displayed in the **Status** column.

Note: When you select a CAS destination and click **Publish**, the CAS destination table is automatically reloaded and the newly published item is made available to other applications. If the table contains models that are currently in use by SAS Model Manager or another application, you might not want to reload the table at the same time that you publish content. In that case, select **Publish without reloading**. However, you must manually reload the table in order for the newly published content to be accessible.

When you are publishing to a SAS Micro Analytic Service destination, the **Micro Analytic Module** column is also displayed with a URL to the published model.

- 9 When the status changes to **Published successfully**, click **Close**.


TIP If you close the Publishing Results window before publishing has completed, you can still see whether the model was published successfully on the **History** tab of the project.


Note: The following is true for models that are within a project version: after a model has been published, a new version of the model is created and a publishing validation test is also created. For more information, see [“Managing Model Versions” on page 70](#) and [“Validate Published Models” on page 85](#).

Publish Models from a Project Version

You can publish models from the **Models** tab of a project, including the champion model.

IMPORTANT Before you can publish a Python model or an R model, the source code must be in the correct format. For more information, see [“Concepts: Open-Source Models” on page 135](#).

- 1 Click  to navigate to the Projects category view.
- 2 Open a project.
- 3 (Optional) Select a version from the **Versions** drop-down list. By default, the models within the current displayed version appear in the list.

- 4 Select one or more models on the **Models** tab, click , and select **Publish**. The Publish Models window appears.
- 5 Select a destination.

Note: If you have Read and Write permissions to the caslib that is specified in a publishing destination, the destination is shown in the list.


For more information, see [SAS Viya: CAS Authorization Window](#) and “Configuring Publishing Destinations” in [SAS Model Manager: Administrator’s Guide](#).

TIP Click **Details** to view the destination details.

- 6 (Optional) Assign tags to the published models by entering a tag value and selecting an existing tag, or press Enter to add a new tag.

TIP Tag names cannot start with a period or a hyphen, and cannot contain more than 128 characters. They can contain only alphanumeric characters, the underscore (_), the hyphen (-), and the period (.).

- 7 Specify a value for any properties that might appear after you select a destination.

Property	Description	Destination Type	Required
Azure Machine Learning workspace	Indicates the workspace in the Azure Machine Learning environment to publish the selected items to. Select a value from the drop-down list.	Azure Machine Learning	Yes
Git directory	Indicates the directory path within the Git repository to publish the selected items to. Click  and navigate to a folder within the Git repository. Click OK . Note: If a folder is not selected within the Git repository, the items are published to the root directory.	Git	No

- 8 (Optional) Update the published name in the **Items to Publish** section. The maximum length and character restrictions differ, depending on your destination.

For more information, see [Table 7.1 on page 112](#).

- 9 (Optional) If you have previously published a model with the same published name, you can replace the model. In the **Items to Publish** section, enable the **Replace item with the same name** toggle for each model that you want to replace.

Note: When publishing models to a container, an Azure Machine Learning, or a Git publishing destination, if a published model with the same name already exists within the destination, a sequential number is added as a tag to indicate the version of the published model.

- 10 Click **Publish**. The Publishing Results window appears. The status of the publishing request is displayed in the **Status** column.

Note: When you select a CAS destination and click **Publish**, the CAS destination table is automatically reloaded and the newly published item is made available to other applications. If the table contains models that are currently in use by SAS Model Manager or another application, you might not want to reload the table at the same time that you publish content. In that case, select **Publish without reloading**. However, you must manually reload the table in order for the newly published content to be accessible.

When you are publishing to a SAS Micro Analytic Service destination, the **Micro Analytic Module** column is also displayed with a URL to the published model.


- 11 When the status changes to **Published successfully**, click **Close**.

Note: The following is true for models that are within a project version: after a model has been published, a new version of the model is created and a publishing validation test is also created. For more information, see [“Managing Model Versions” on page 70](#) and [“Validate Published Models” on page 85](#).

View Model Publish History

When a user publishes a model, an event is added to the **History** tab of the project.

Note: Only models that are associated with a project can record an event on the **History** tab.

- 1 Click  to navigate to the Projects category view.
- 2 Open a project.
- 3 Click the **History** tab of the project.


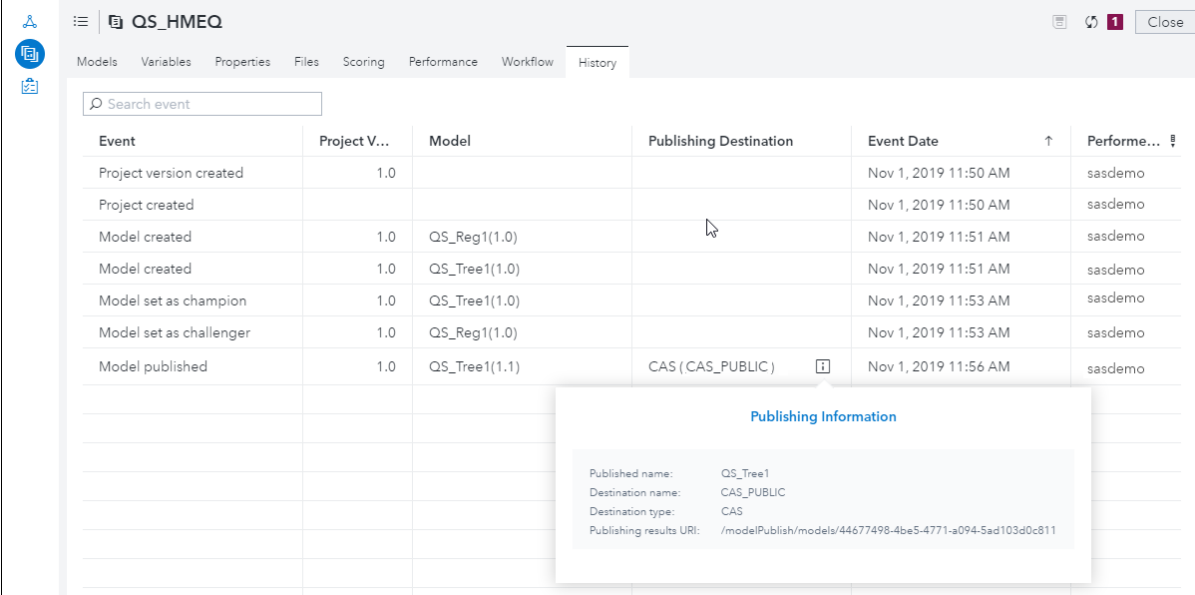
- 4 (Optional) Enter `published` in the **Search event** box to search for models that have been published.
- 5 Click  in the **Publishing Destination** column to view the publishing information for a model.
- 6 Click anywhere on the screen to close the information pop-up window.

Figure 7.1 Example of the Publishing Information for a CAS Publishing Destination




Event	Project V...	Model	Publishing Destination	Event Date	↑	Performe... ↓
Project version created	1.0			Nov 1, 2019 11:50 AM		sasdemo
Project created				Nov 1, 2019 11:50 AM		sasdemo
Model created	1.0	QS_Reg1(1.0)		Nov 1, 2019 11:51 AM		sasdemo
Model created	1.0	QS_Tree1(1.0)		Nov 1, 2019 11:51 AM		sasdemo
Model set as champion	1.0	QS_Tree1(1.0)		Nov 1, 2019 11:53 AM		sasdemo
Model set as challenger	1.0	QS_Reg1(1.0)		Nov 1, 2019 11:53 AM		sasdemo
Model published	1.0	QS_Tree1(1.1)	CAS (CAS_PUBLIC)	Nov 1, 2019 11:56 AM		sasdemo

Publishing Information

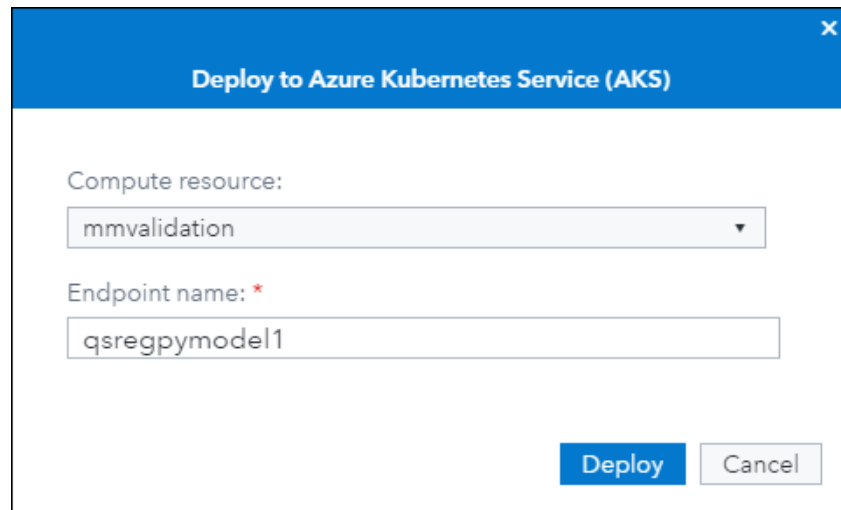
Published name: QS_Tree1
 Destination name: CAS_PUBLIC
 Destination type: CAS
 Publishing results URI: /modelPublish/models/44677498-4be5-4771-a094-5ad103d0c811

Deploy a Published Model to Azure Kubernetes Service (AKS)

After you publish a model to an Azure Machine Learning publishing destination you can deploy the published model to AKS. This enables you to make the published model available so that you score it using custom code or another application outside the SAS Viya environment. It also enables you to deploy the published model in a production environment.

- 1 Click the **Scoring** tab of a project, and then click the **Publishing Validation** tab.
- 2 Select a publishing validation test, click , and select **Deploy to AKS**. The Deploy to Azure Kubernetes Service (AKS) window appears.
- 3 Select a compute resource that you have access to.
- 4 Enter a value for the endpoint name.

Note: The endpoint name can contain only alphanumeric characters, and letters must be lowercase.



Deploy to Azure Kubernetes Service (AKS)

Compute resource:
mmvalidation

Endpoint name: *
qsregpymodel1

Deploy Cancel

5 Click **Deploy**.

Note: A toast message is displayed indicating that the published model is being deployed from the Azure Machine Learning to Azure Kubernetes Service (AKS). You can check the deployment status within the Azure Machine Learning destination.

Tracking Deployments

<i>About Deployments</i>	121
<i>View and Filter Deployments</i>	122
View Deployments	122
Filter Deployments Using the Graph	122
View Deployment Related Content	123
<i>Search for Deployments</i>	123
Advanced Search for Deployed Models	124

About Deployments

The goal of a modeling project is to identify models that can predict the best data-driven outcomes and deploy them in critical business systems. SAS Model Manager provides tools to evaluate candidate models, declare champion models, and inform your scoring officer that a predictive model is ready for validation or production.

The Deployments category view enables you to access all of your model deployments in one place, view key metrics about your organization's model deployments, and track their status.

Note: Deployments are not available until after you have published models from a project.

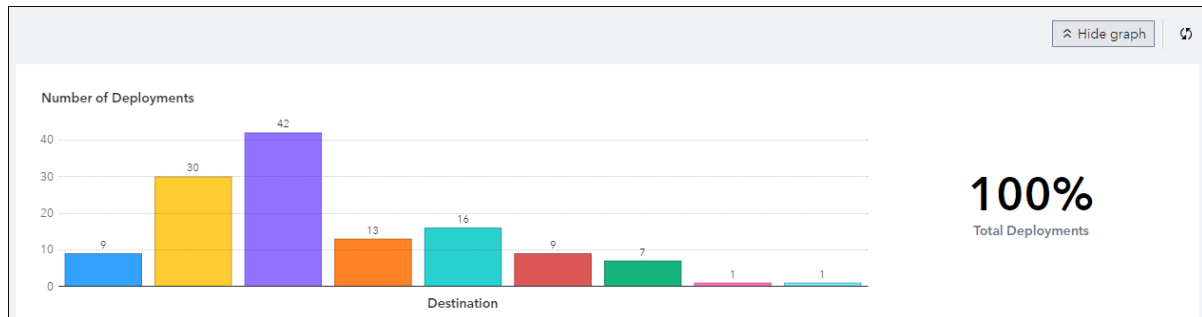
View and Filter Deployments

View Deployments

When you view the Deployments category, a graph appears at the top of the application window and tiles with key information are displayed in the left panel. The graphs and tiles provide the following metrics:

- total deployments
- number of deployments for each destination
- percentage of total deployments
- number of projects and percentage of projects with published models
- number of published models and percentage of published models with successful validation
- number of published models and percentage of published models without validation

You can click **Hide graph** or **Show graph** on the toolbar.

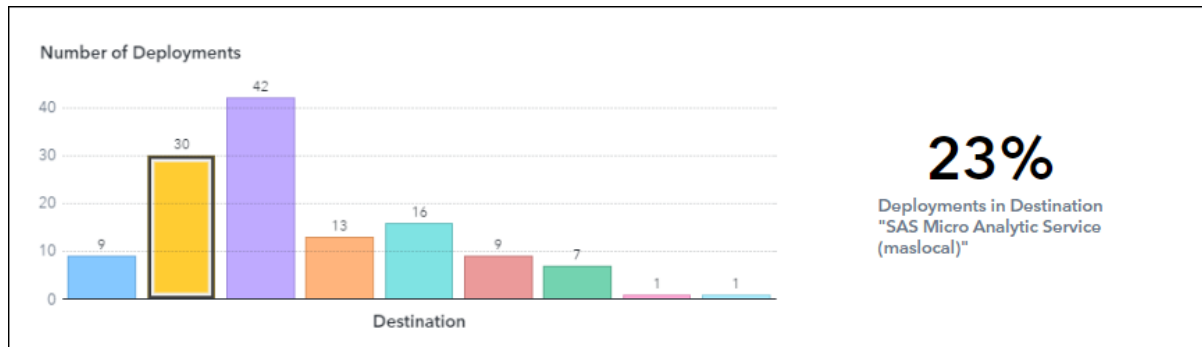


Filter Deployments Using the Graph

You can click sections of the graph to filter the list of models. Tokens are added to the search bar above the list of items.

TIP To select more than one bar in the graph select CTRL+

Here is an example of filtering the list to display only deployments in the SAS Micro Analytic Service (maslocal) destination.



You can also use the Advanced Search for Deployment Models window to specify additional information in order to further filter the list. The filters that you applied using the graph are reflected in this window. For more information, see [“Advanced Search for Models” on page 73](#).

TIP To remove an individual filter from a graph, click the empty area beside the graph or click the × icon within the tokens that are located above the list of deployments. To remove all filters, click the × icon, which is located in the right side of the toolbar.

View Deployment Related Content

Values in some of the columns are links that you can click to display the associated content. Here is the content that is displayed for each type of link:

Column Name	Content Displayed
Model Name (Version)	Opens the associated model object.
Published Name	Opens the associated project and displays the Publishing Validation page of the Scoring tab.
Project (Version)	Opens the associated project and displays the Models tab.

Search for Deployments

In the **Deployments** category view, you can perform multiple types of searches:


- search for deployments by destination and published name using the search field above the list.
- perform an advanced search for deployments using destinations, published name, score code type, date published (range), or published by properties.

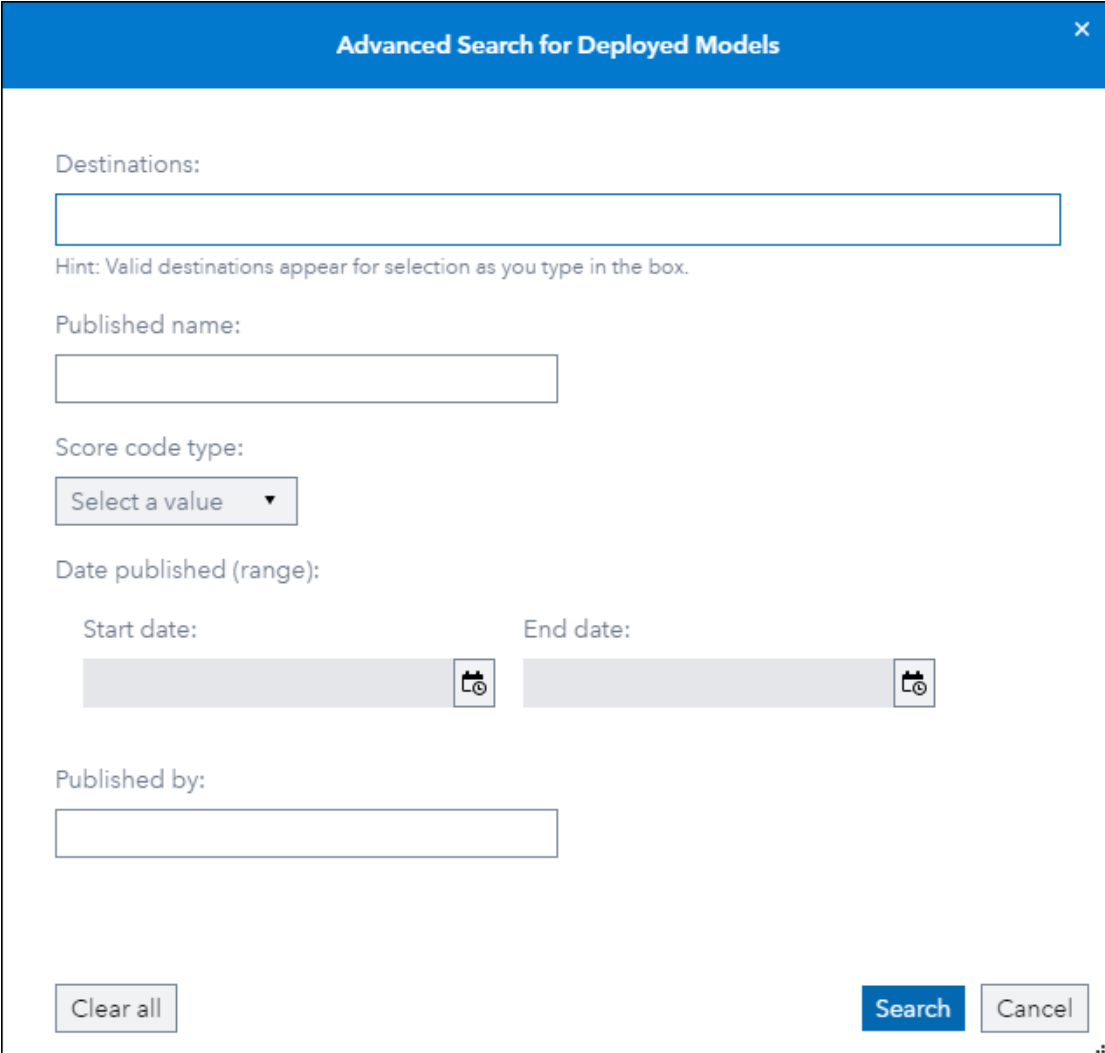
- search for objects across applications using the search field in the application bar.

For more information about searching, see [“Search” in SAS Viya Web Applications: General Usage Help](#).

Advanced Search for Deployed Models

To perform an advanced search for models:

- 1 Click  **Advanced search**, which is to the right of the search box. The Advanced Search for Deployed Models window appears.





Advanced Search for Deployed Models ×

Destinations:

Hint: Valid destinations appear for selection as you type in the box.

Published name:

Score code type:
Select a value ▼

Date published (range):
Start date:  End date: 

Published by:

Clear all Search Cancel

- 2 To search for deployed models within specific destinations, enter a destination in the **Destinations** box and select it from the list that appears.
- 3 Enter a published name or part of a published name to include it in the search criteria.
- 4 Select a value for the score code type.

- Specify a range for the date published in order to limit the list of deployments to specific dates and times.

Advanced Search for Deployed Models
✕

Destinations:

SAS Micro Analytic Service (maslocal) ✕
CAS (CAS_Public) ✕

Hint: Valid destinations appear for selection as you type in the box.

Published name:

QS_

Score code type:

Select a value
▼

Date published (range):

Start date:

Oct 8, 2021, 6:18 PM
✕
📅

End date:

Oct 15, 2021, 6:18 PM
✕
📅

Published by:

sasdemo

Clear all

Search

Cancel

- Enter a user ID to include the user that the models were published by in the search criteria.
- Click **Search**. The Deployments category view is displayed with the list of deployments that meet the advanced search criteria.

TIP Click **Clear All** and then click **Search** to clear all previously entered search criteria in the Advanced Search for Deployed Models window.

🔍 Advanced search

(2)
✕

Destinations ▼ ✕
QS_ ✕
Date published (range): Oct 8, 2021 6:18 PM - Oct 15, 2021 6:18 PM ✕

<input type="checkbox"/>	Destination	Model Name (Ve...	Published Name	Role	Project (Version)	Project Status	Date Published
<input type="checkbox"/>	maslocal	QS_Reg_PyMode...	QS_Reg_PyModel		QS_HMEQ(1.0)	Prototype	Oct 15, 2021 5:19 PM
<input type="checkbox"/>	CAS_Public	QS_Tree1 (1.1)	QS_Tree1	🚩	QS_HMEQ(1.0)	Prototype	Oct 15, 2021 11:52 AM

Using SAS Workflow with SAS Model Manager

<i>About Using Workflows</i>	127
<i>Requirements</i>	128
<i>Prompts and Attributes</i>	128
<i>Start a New Workflow</i>	130
<i>Working with Tasks</i>	131
About Tasks	131
Complete a Task	132
Release a Task	132
Filter Tasks	132

About Using Workflows

SAS Model Manager uses the **Workflow** tab within a project and the Tasks category view to interface with SAS Workflow. A workflow is an instance of a workflow definition. A workflow can be used to track the progress of objects, such as projects. An authorized user can use SAS Workflow Manager to create workflow definitions and to make them available to SAS Model Manager for use. Workflow definitions contain the set of tasks, participants, and data objects that comprise a business task. The status that you select when completing a task determines the next task in the workflow. All users can access the Tasks category view.

For information about creating workflow definitions, see [SAS Workflow Manager: User's Guide](#).

Requirements

Before users can use SAS Workflow with SAS Model Manager, a system administrator must complete the following tasks:

- Configure a user account for a workflow client to have service execution privileges. The Workflow service uses this account to make service task calls when running a workflow instance.
- Grant users or a user group workflow administrative privileges.
- Specify the SAS Model Manager client identifier and enable a workflow definition to make it available to SAS Model Manager.

Note: The developer of a workflow definition can also complete this task.


For more information, see [“Configure SAS Workflow” in SAS Model Manager: Administrator’s Guide](#).

Prompts and Attributes

You can add prompts and attributes to the Start element and user tasks in order to customize fields within workflow tasks. SAS Model Manager provides attributes that enable you to prompt users to select a model or select potential owners from a list within the Start element or a user task as part of the workflow process.

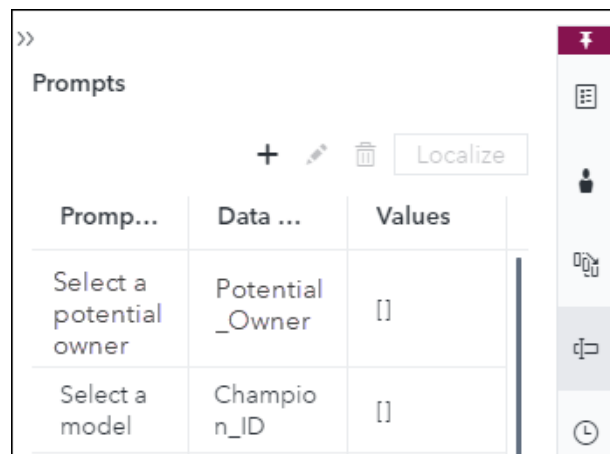
Here are the controls and attribute values that are supported:

Type of Control	Attribute Value	Description
Choose Models	show_model_list	<p>The show_model_list attribute value and the associated control enable you to configure a prompt as part of a workflow user task that enables a user to select a model from a list.</p> <p>A prompt with the specified name is displayed with the Choose Models control within the associated user task. The Choose Models button launches the Choose Models window, which contains a list of all models for the associated project and its containing versions. This enables</p>

Type of Control	Attribute Value	Description
		a user to select a model from a list within a workflow user task.
Choose Members	select_potential_owner	<p>The select_potential_owner attribute value and the associated control enable you to configure a prompt as part of a workflow user task that enables a user to select a user or group as a potential owner during the workflow process.</p> <p>A prompt with the specified name is displayed with the Choose Members control within the associated user task. The user button  launches the Choose Members window, which contains a list of users and groups. This enables a user to select a user or group as a potential owner as part of the user task.</p>

Here are the high-level steps for adding attributes to a workflow user task to utilize the controls that SAS Model Manager provides:

- 1 Add a user task to your workflow diagram.
- 2 Add a prompt and associate it with a data object.



- 3 Add an attribute with the same name as the prompt and specify an attribute value that SAS Model Manager supports.

IMPORTANT The name of the attribute and the prompt must match, in order for the control that is associated with the attribute to appear in the workflow task.

Name	Value
Select a potential owner	select_potential_owner
Select a model	show_model_list

Here is an example of a user task with the **Select a model** prompt.

Note: Prompts are disabled until a user claims a task.

For an example of how to add a prompt as part of a workflow user task and associate the attribute with the prompt, see [“Add the Set Champion and Challenger Models User Task”](#) in *SAS Workflow Manager: Quick Start Tutorial*. You can also use the workflow definition BPMN file that is included in the QuickStartTutorial.zip file and follow the steps in [“Add the Set Champion and Challenger Models User Task”](#) in *SAS Workflow Manager: Quick Start Tutorial*.

For more information about adding attributes and prompts, see [“Managing Attributes”](#) in *SAS Workflow Manager: User’s Guide*.

Start a New Workflow

When you start a new workflow, it is associated with the project. For a specific project, only one workflow can be in progress at a time. The tasks within a workflow must be completed or the in-progress workflow process must be terminated, before a new workflow can be started.

To start a workflow:

- 1 Open a project and click the **Workflow** tab.

Note: The **Workflow** tab appears only for users who have workflow administrative privileges when workflow definitions are available. For more information, see [“Requirements” on page 128](#).

- 2 Click **Start Workflow** and select a workflow definition from the list. The Start Workflow window appears.
- 3 Specify values for any prompts that are displayed.

Note: What is displayed in the **Start Workflow** window depends on what is configured in the workflow definition start node. If prompts are not configured for the start node, the default text is “Are you sure you want to start this workflow?”.

- 4 Click **Start**. The workflow is added to the list with a status of “In progress”.

Name	Status	Started By	Date Started	Date Ended	Description
Model_Lifecycle	In progress	sasdemo	Jan 27, 2021 08:49 PM		

Working with Tasks

About Tasks



The Tasks category view displays the tasks for workflows that are in progress, and that you have been assigned to as a potential owner or that have been claimed by you.

Name	Workflow	Date Started	Claimed By	Date Claimed	Date Due	Associated Objects
Import models	Model_Lifecycle	Nov 1, 2019 02:47 PM				Project: QS...

In the Tasks category view, you can perform the following:



- claim a task
- open a task
- release a task
- view the object that is associated with a task

Complete a Task


- 1 Click  to view the **Tasks** category.
- 2 Click on a task to open it.
- 3 Click  to claim the task.
- 4 Specify values for any prompts that are displayed on the **Prompts** tab.


.....
Note: Prompts are disabled until you claim the task.
.....

- 5 Click the **Properties** tab to view task properties, including the associated object.
- 6 Click on the name of the associated object to open it.
- 7 Complete any actions that are associated with the task. An example is importing models into a project.
- 8 Click **Complete**.
- 9 Click **Complete** in the confirmation message. You are returned to the task category view.

.....
Note: Alternatively, you can select one or more tasks and can click  to open them or click  to claim them.
.....

Release a Task

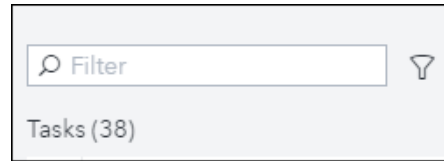
- 1 Select one or more tasks.
- 2 Click .


.....
Note: Alternatively, if you already have the task open, you can click .

Filter Tasks

From the Tasks category view, you can filter the tasks that are displayed in the list. Here are the two options available for filtering tasks:

- Enter a value in the **Filter** field above the list to filter the list by task name.

Figure 9.1 Example of Filtering by Task Name

- Click  to filter the list by workflow name, date started, date claimed, or date due.

Concepts

Concepts: Open-Source Models	135
Python Models	136
R Models	138
Concepts: Performance Monitoring	139
About Performance Monitoring	139
Data Composition	140
Model Monitoring	143
Feature Contribution	152
Population Stability	153
Standard KPI Trend	155
Concepts: PMML Support	155
Overview	155
PROC PSCORE Functionality	156
Supported Versions	156
Supported PMML Models	156
Requirements for PROC PSCORE	157
PROC PSCORE Usage	157
PROC PSCORE Example	157
See Also	157

Concepts: Open-Source Models

You can use the Python and R open-source programming languages to develop analytic models. This includes model content that is stored in PMML, ONNX, or RDS formats and Python models that are developed with packages such as scikit-learn, TensorFlow, and XGBoost. You can then import those models into SAS Model Manager web applications to compare, test, and evaluate the performance of the models before publishing them to a test or production environment. When you publish open-source models to a container publishing destination, they can then be scored within run-time containers.

Several options exist for importing open-source models into SAS Model Manager. You can place your model files in a ZIP file and then import the model, or you can

create a new model and add local files. For more information, see [“Add a New Custom Model” on page 50](#) and [“Import Models” on page 53](#). You can also create models using the Model Repository API. The API can be called in several ways, including using Python programs or using Python in a Jupyter notebook. For more information, see [model-management-resources](#) GitHub repository and [Model Repository API](#) documentation.

Python Models

Scoring Python Models

After you import a Python model, you can run an on-demand scoring test for the model. You can also publish a model to a publishing destination to score and validate it. When you publish a model from a project version to a CAS, SAS Micro Analytic Service, Azure Machine Learning, or a container publishing destination such as Amazon Web Services (AWS), Azure, or Private Docker, a publishing validation test is created. The publishing validation test can then be run to validate the model within the publishing destination.

In order to run an on-demand scoring test for a model from within the SAS Model Manager web application, the Python score code must be in the correct format. Here are the format requirements for what must be included in the Python score code:

- import statements for any modules that are needed.
- a score function definition with a list of the input variables.

.....
Note: Variable names cannot be more than 32 characters long.

- an Output statement enclosed in quotation marks that lists the output variables and directly follows the score function definition. List all output parameters as comma-separated values. Do not specify None if there is no output parameter.
- the lists of variables are case insensitive and must be comma separated.

Here is a Python code example:

```
import numpy
import pickle
import settings

def scoreModel(CLAGE, DEBTINC, NINQ, VALUE, LOAN, MORTDUE, YOJ):
    "Output: EM_EVENTPROBABILITY, EM_CLASSIFICATION"
```

If your Python model includes a pickle file, you must also include the following code, which specifies the name of the pickle file that should be loaded:

```
try:
    _thisModelFit
except NameError:
    with open(settings.pickle_path + "hmeq_logistic.pickle", 'rb') as _pFile:
        _thisModelFit = pickle.load(_pFile)
```

After you publish a model to a container destination, a publishing validation test is created. The `settings.py` file and DS2 package wrapper files are also generated and added to the model. You can use the publishing validation test to score and validate your model within a run-time container.

IMPORTANT In order to validate Python models within a container publishing destination, the Python packages which contain the modules that are used in the Python score code file and its score resource files must be installed in the run-time container. You can install the packages when you publish a Python model or decision that contains a Python model to a container publishing destination by adding a `requirements.json` file that includes the package install statements to your model. For information about adding files, see [“Managing Model Files” on page 60](#) and [Add a Model File](#) in the `devsascom-rest-api-samples` GitHub repository.

Here is an example:

```
[
  {
    "step": "install statsmodels ",
    "command": "pip3 install statsmodels"
  }
]
```

For additional examples of how to format your Python score code file and the `requirements.json` file, see the Python model samples in the [model-management-resources](#) GitHub repository.

See Also

- [“Test Models” on page 81](#)
- [“DS2 Interface to Python” in *SAS Micro Analytic Service: Programming and Administration Guide*](#)

Publishing Python Models

Python models can be published to CAS, SAS Micro Analytic Service, Git, Azure Machine Learning, or a container publishing destination such as Amazon Web Services (AWS), Azure, or Private Docker. When you import a Python model, the score code type is set to Python.

See Also

- [“About Publishing Models” on page 109](#)
- [“Validate Published Models” on page 85](#)

R Models

Scoring R Models

After you import an R model, you can run an on-demand scoring test for the model. You can also publish a model to a publishing destination to score and validate it. When you publish a model from a project version to a CAS, SAS Micro Analytic Service, or a container publishing destination such as Amazon Web Services (AWS), Azure, or Private Docker, a publishing validation test is created. The publishing validation test can then be run to validate the model within the publishing destination. Models that are published to a container publishing destination can be scored within run-time containers.

In order to run an on-demand scoring test for a model from within the SAS Model Manager web application, the R score code must be in the correct format.

Note: If an R model is transformed to PMML 4.2 format and the PMML XML file is imported into SAS Model Manager, the score code type is set to DATA step. PMML models with a DATA step score code can be scored using a scoring test within the SAS Model Manager web application.

Here are the format requirements for what must be included in the R score code:

- import statements for any libraries that are needed.
- a score function definition with a list of the input variables.

Note: Variable names cannot be more than 32 characters long.

- an Output statement that lists the output variables must be declared on the first line within the body of the score function definition. List all output parameters as comma-separated values. Do not specify None if there is no output parameter. Example: `#output: outvar1, outvar2`
- the lists of variables are case insensitive and must be comma separated.
- the path to the RDA or RDS file must be specified. Example: `paste(rdsPath, 'rds_file_name.rds', sep = '')`

Here is an R code example:

```
scoreFunctionName <- function(var1, var2, var3)
{
  #output: outvar1, outvar2

  if (!exists("myClassTree"))
  {
    assign("myClassTree", readRDS(file = paste(rdsPath,
'hmeq_classtree_r.rds', sep = '')), envir = .GlobalEnv)
  }
}
```

```
# Include scoring logic here to get a list of the output variables.  
  
output_list <- list('outvar1' = outvar1, 'outvar2' = outvar2)  
return(output_list)  
}
```

Publishing R Models

R models can be published to CAS, a Git repository, SAS Micro Analytic Service, or a container publishing destination such as Amazon Web Services (AWS), Azure, or Private Docker. When you import an R model, the score code type is set to R. To be published to CAS or SAS Micro Analytic Service, R models must have a score code type of R and an R score code file that contains a scoring function in the correct format.

See Also

- [“About Publishing Models” on page 109](#)
- [“Validate Published Models” on page 85](#)

Concepts: Performance Monitoring

About Performance Monitoring

You can monitor the effectiveness of models by analyzing the performance results. You can create a performance definition on the **Performance** tab of a project. A performance job is then created for each model when you save the definition. When you run a performance job, the summary information and the charts for each model that appear on the **Performance** tab represent the performance results data that is generated. The key performance indicators (KPIs) help you to assess the health and effectiveness of your models within the project.

The following types of charts are included in the performance results:

Data Composition

The [Variable Distribution](#) chart shows you the distributions for a variable in one or more time periods, this enables you to see the differences and changes over time.

The [Characteristic and Stability](#) charts detect and quantify shifts in the distribution of variable values that occur in input data and scored output data over time. By analyzing these shifts, you can gain insights on scoring input and output variables.

Model Monitoring

The model monitoring charts are a collection of performance assessment charts that evaluate the predicted and actual target values. The model monitoring charts create several charts:

- [Lift](#)
- [Gini](#)
- [ROC \(Receiver Operating Characteristic\) on page 144](#)
- [KS](#)
- [Average Squared Error \(ASE\) of the Interval Target](#)
- [Mean of the Predicted Interval Target](#)
- [Standard Deviation of the Predicted Interval Target](#)
- [Proportion of Predicted Interval Target Values](#)

Feature Contribution Index

The feature contribution index (FCI) measures the relationships between input and output variables. This enables you to evaluate the importance of predictors within that model.

Standard Key Performance Indicator (KPI)

The Standard KPI Trend chart shows the trend of the selected standard key performance indicator (KPI) that is calculated over a period of time.

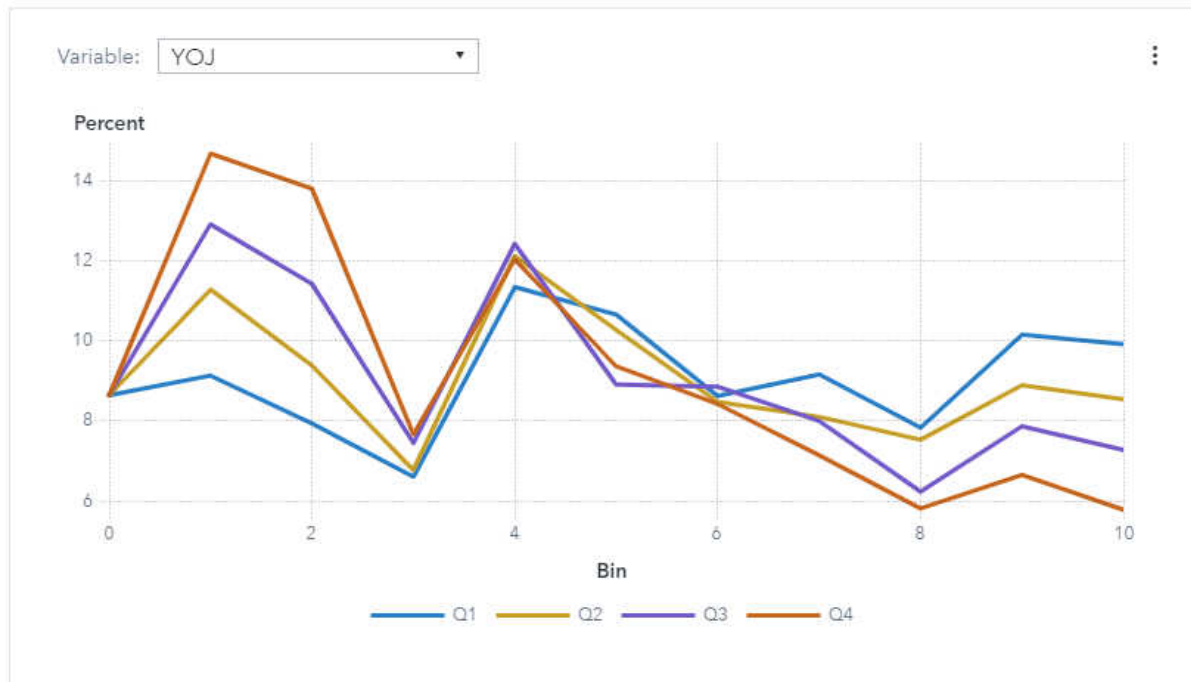
Data Composition

Variable Distribution

On the **Performance** tab of a project, you can view the variable distribution of a model. The Variable Distribution chart is a graphical representation of distributions over a period of time for the selected variable. Each line plot represents the data for a specific period of time. The Y axis is the percentage of observations in a bin that is proportional to the total count.

To change the variable that appears in the chart, select a variable from the drop-down list.

Here is an example of an Input Variable Distribution chart.



Characteristic and Stability

Together, the Input Variable Characteristic and Output Variable Stability charts detect and quantify shifts that can occur in the distribution of model performance data, scoring input data, and the scored output data that a model produces.

Note: For each time period that you run performance, SAS Model Manager creates a new point on the charts. Line segments between points in time do not appear on the charts unless you specify at least three data sources and collection dates as part of the performance definition.

Characteristic

The Input Variable Characteristic chart detects and quantifies the shifts in the distribution of variable values in the input data over time. These shifts can point to significant changes in customer behavior that are due to new technology, competition, marketing promotions, new laws, or other influences.

To find shifts, the Characteristic chart compares the distributions of the variables in these two data sets:

- the training data set that was used to develop the model
- a current data set

If large enough shifts occur in the distribution of variable values over time, the original model might not be the best predictive or classification tool to use with the current data.

The Characteristic chart uses a deviation index to quantify the shifts in a variable's values distribution that can occur between the training data set and the current data set. The deviation index is computed for each predictor variable in the data set, using this equation:

$$\text{Deviation_Index} = \sum \left(\% \text{ Actual} - \% \text{ Expected} \right) \times \ln \left(\frac{\% \text{ Actual}}{\% \text{ Expected}} \right)$$

Numeric predictor variable values are placed into bins for frequency analysis. Outlier values are removed to facilitate better placement of values and to avoid scenarios that can aggregate most observations into a single bin.

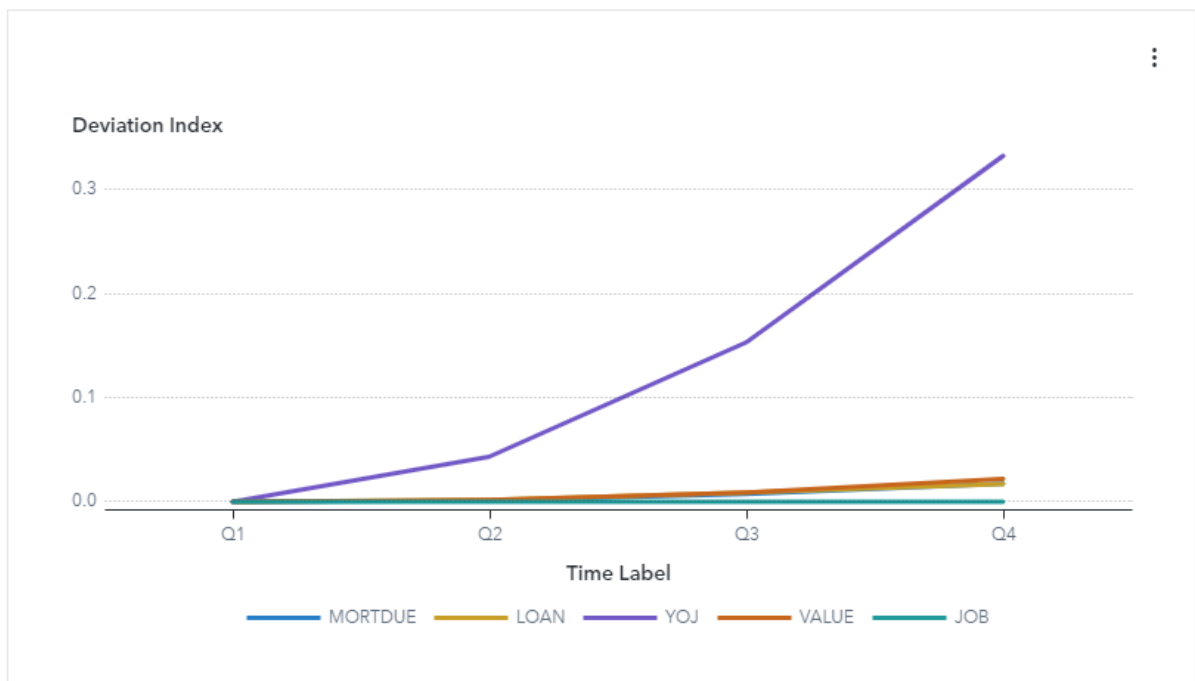
If the training data set and the current data set have identical distributions for a variable, the variable's deviation index is equal to 0. A variable with a deviation index value that is $P1 > 2$ is classified as having a mild deviation. The Characteristic chart uses the performance measure P1 to count the number of variables that receive a deviation index value that is greater than 0.1.

A variable that has a deviation index value that is $P1 > 5$ or $P25 > 0$ is classified as having a significant deviation. A performance measure P25 is used to count the number of variables that have significant deviations, or the number of input variables that receive a deviation index score value that is greater than or equal to 0.25.

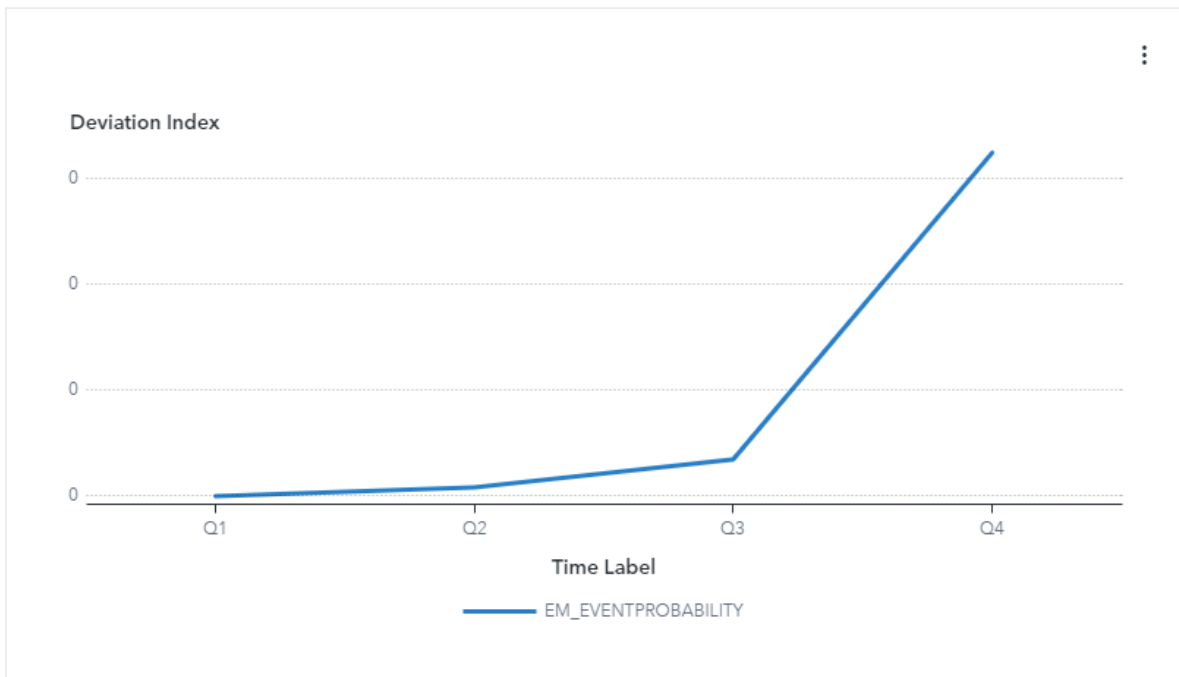
Stability

The Output Variable Stability chart evaluates changes in the distribution of scored output variable values as models score data over time, and detects and quantifies shifts in the distribution of output variable values in the data that is produced by the models. If an output variable from the training data set and the output variable from the current data set have identical distributions, then that output variable's deviation index is equal to 0. An output variable with a deviation index value that is greater than 0.10 and less than 0.25 is classified as having a mild deviation. A variable that has a deviation index value that is greater than 0.30 is classified as having a significant deviation. Too much deviation in predictive variable output can indicate that model tuning, retraining, or replacement might be necessary.

Here is an example of the Input Variable Characteristic chart.



Here is an example of the Output Variable Stability chart.



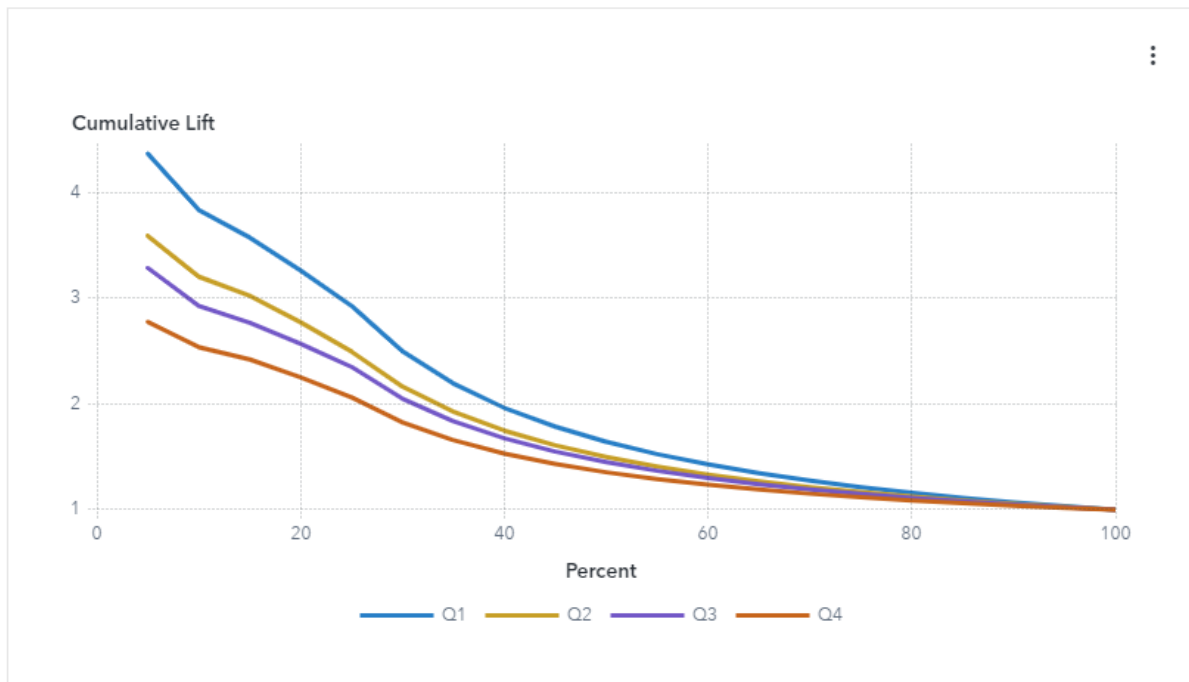
Model Monitoring

Lift

The Lift chart provides a visual summary of the usefulness of the information that is provided by a classification model for predicting a binary outcome variable. Specifically, the report summarizes the utility that you can expect by using the champion model as compared to using baseline information only. Baseline information is the prediction accuracy performance of the initial performance monitoring definition or batch program using operational data.

A monitoring Lift chart can show a model's cumulative lift at a given point in time or the sequential lift performance of a model's lift over time. The Lift performance indexes Lift5Decay, Lift10Decay, Lift15Decay, and Lift20Decay are used to detect model performance degradation. The performance indexes are not displayed in the Lift chart, but are available in the mm_model_indicator performance results data table. The data that underlies the Lift chart is contained in the mm_lift performance results data table.

Here is an example of a monitoring Lift chart.



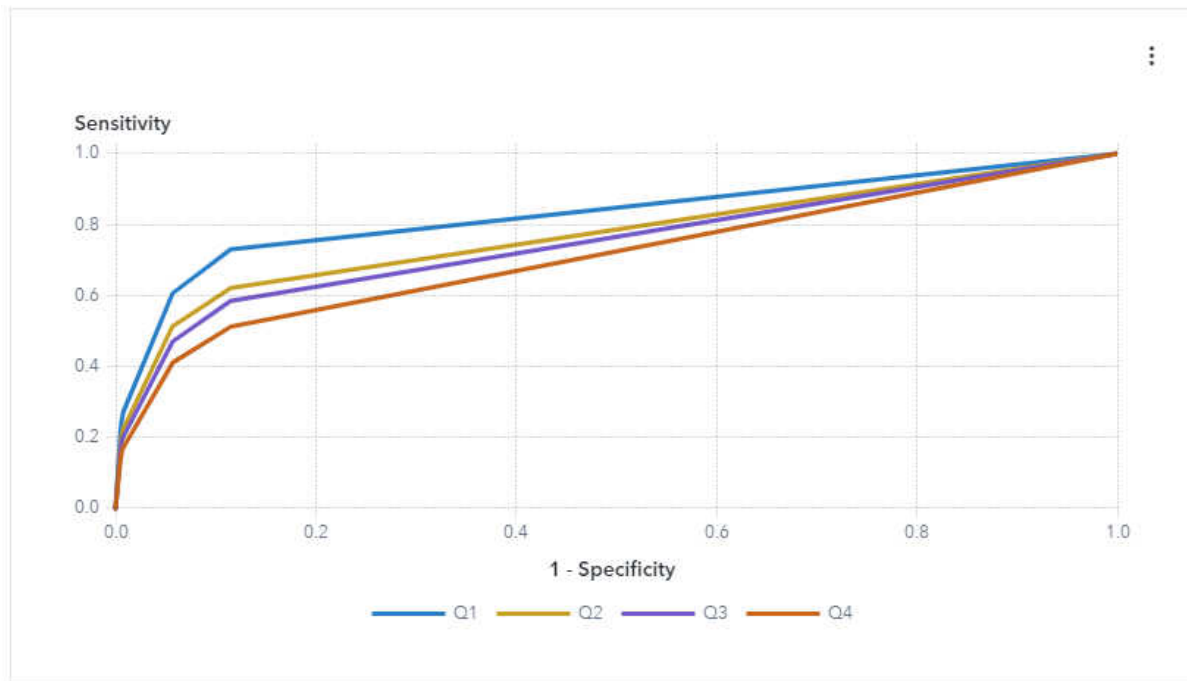
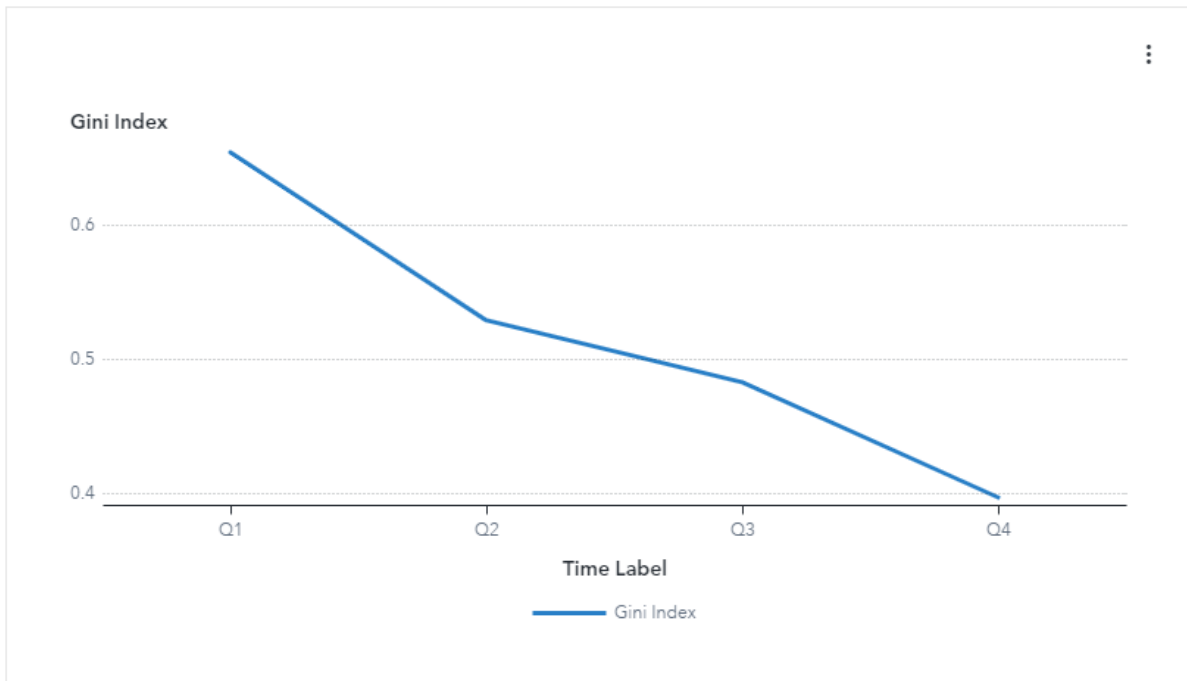
Gini and ROC

The Gini and ROC charts show you the predictive accuracy of a classification model that has a binary target. The plot displays sensitivity information about the Y axis and 1-Specificity information about the X axis. Sensitivity is the proportion of true positive events. Specificity is the proportion of true negative events. The Gini index is calculated for each ROC curve. The Gini coefficient is a benchmark statistic that can be used to summarize the predictive accuracy of a model, and is directly related to the area under the ROC curve ($2 \cdot \text{AUC} - 1$).

Use the monitoring Gini chart to detect degradations in the predictive power of a model.

The data that underlies the monitoring Gini and ROC charts are contained in the `mm_roc` performance results data table.

Here are examples of the monitoring **Gini** and **ROC** charts.



KS

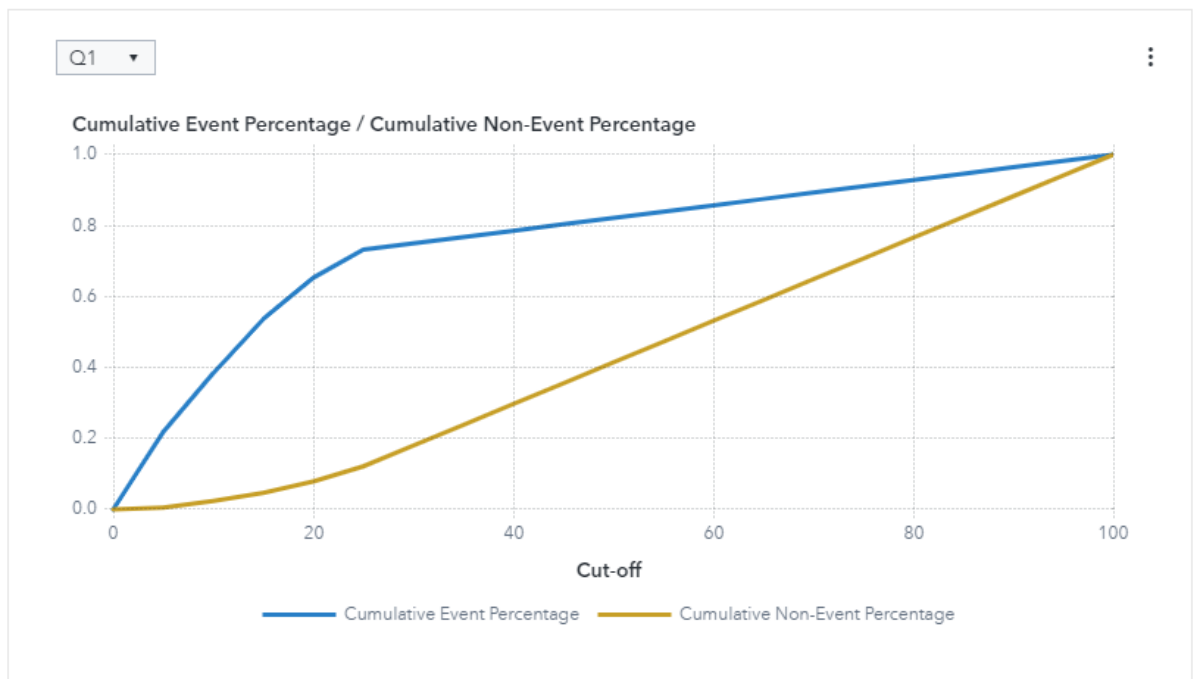
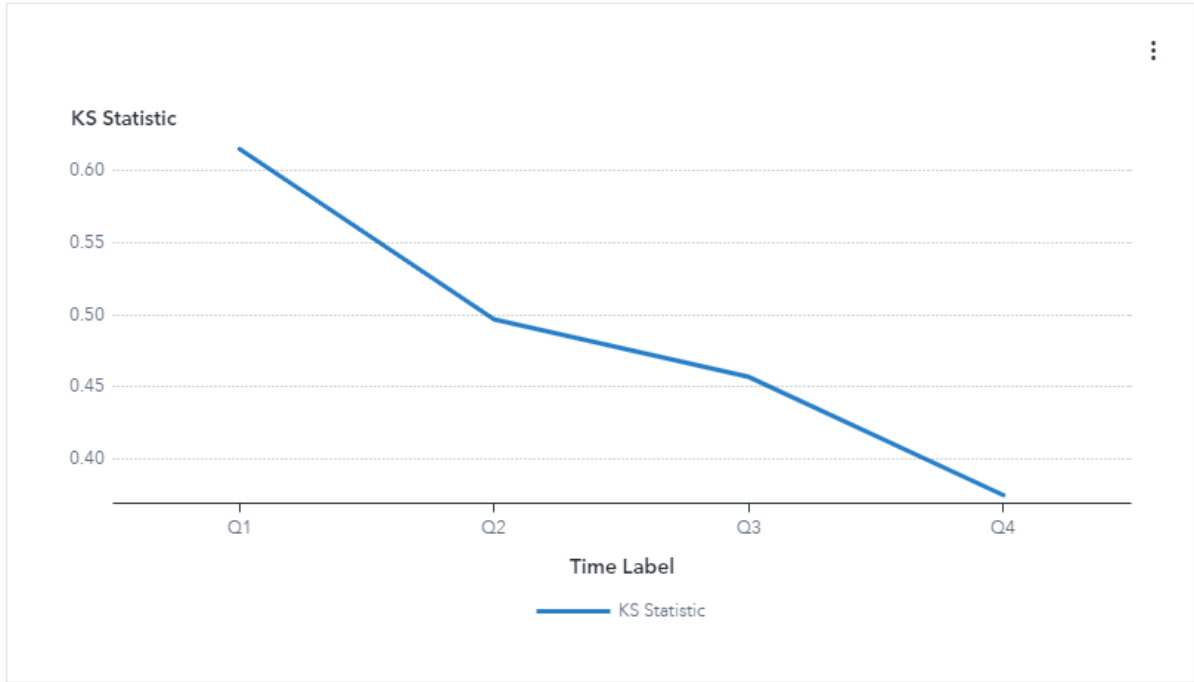
The KS charts contains the Kolmogorov-Smirnov (KS) test plots for models with a binary target. The KS statistic measures the maximum vertical separation, or deviation between the cumulative distributions of events and non-events. This trend chart uses a summary data set that plots the KS statistic and the KS probability cutoff values over time.

Use the KS chart to detect degradations in the predictive power of a model. To scroll through a successive series of KS performance depictions, select a time interval from the **Time Interval** list box. If model performance is declining, it is indicated by the decreasing distances between the KS plot lines.

The ksDecay performance index detects model performance degradation. The ksDecay performance index is not displayed on the KS chart, but is available in the mm_model_indicator performance results data table.

The data that underlies the KS chart is contained in the mm_ks performance results data table.

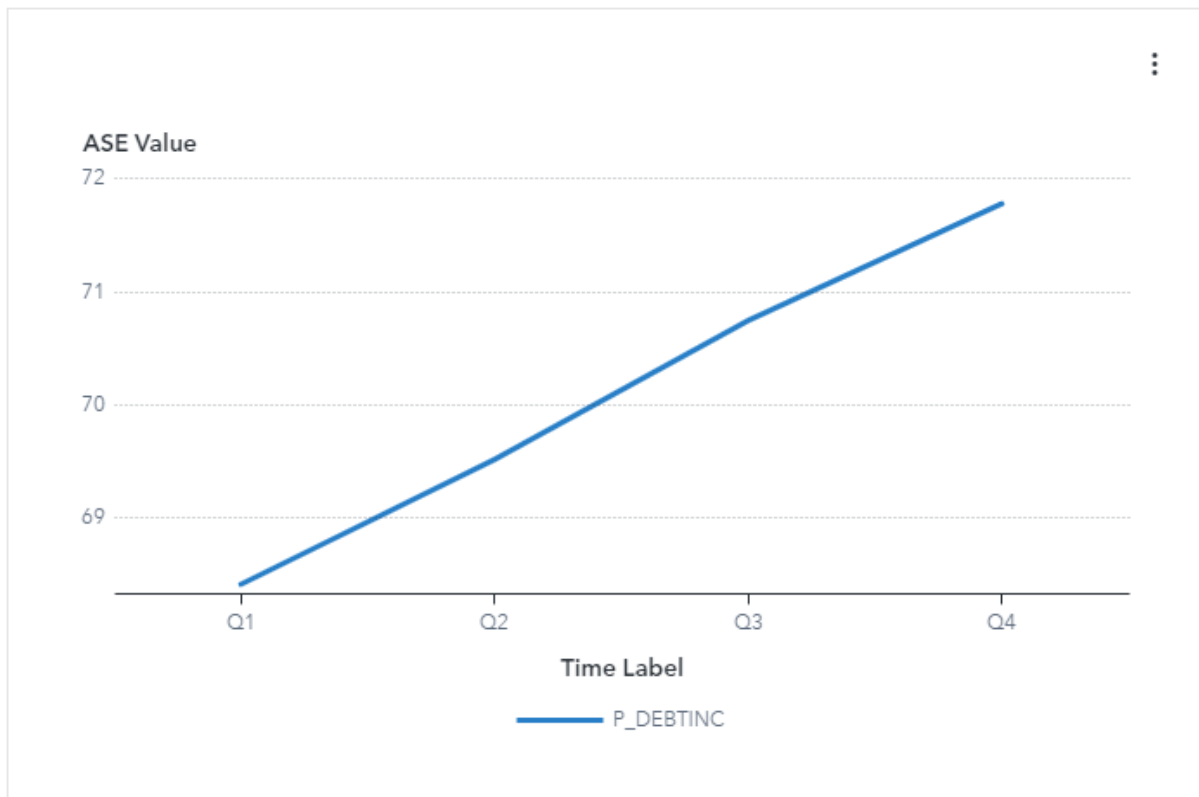
Here are examples of the KS charts.



Average Squared Error (ASE) of the Interval Target

The Average Squared Error (ASE) chart checks the accuracy of a prediction model with an interval target by comparing the estimation derived from the test data and the actual outcomes that are associated with the test data for different time periods.

Here is an example of the chart.



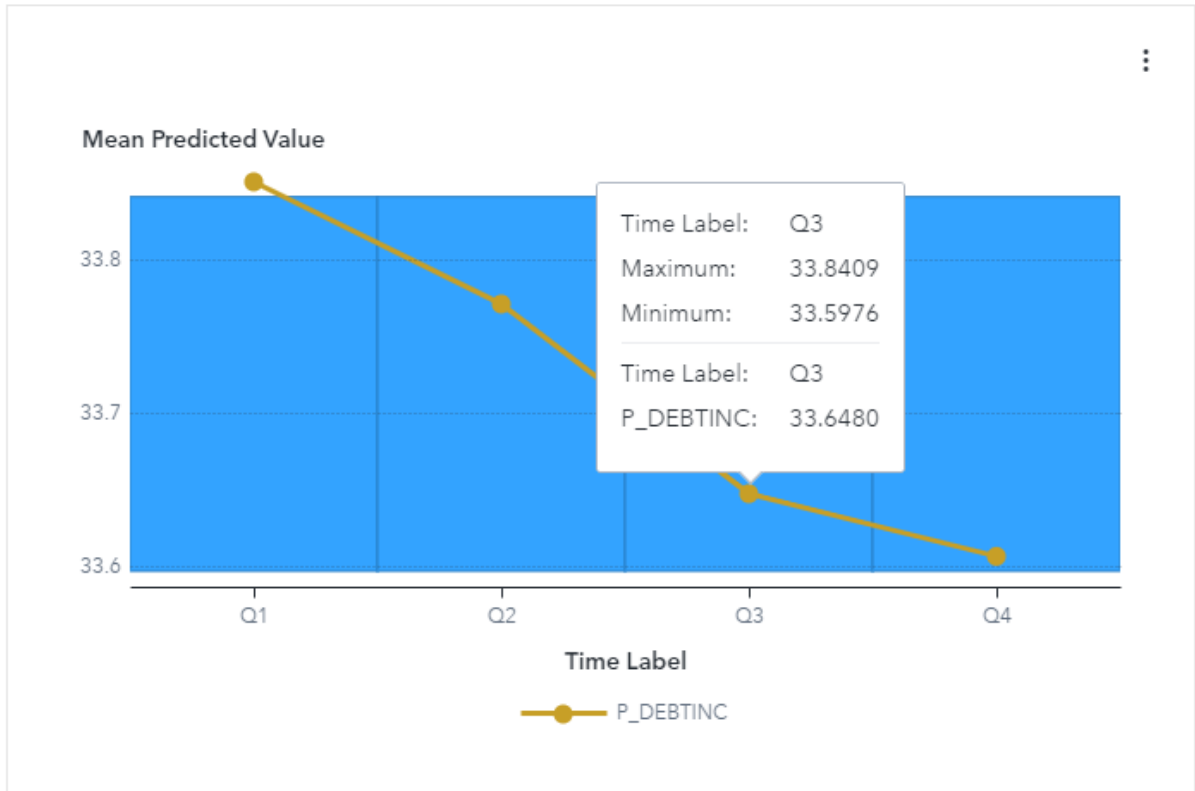
Mean of the Predicted Interval Target

The Mean of the Predicted Interval Target chart shows the mean of values for a prediction model with an interval target for different time periods. The central horizontal line represents the weighted average of the means. The shaded band represents the control limits. If a point in the line chart is outside the shaded band, it indicates that the predicted interval target values at the corresponding time period have deviated significantly. You can then further scrutinize the model outcomes at that time period.

If the underlying data distribution has not changed over time and the current model is still effective in predicting the interval target, you should not see any significant changes in the overall spread of the predicted interval target values.

Concepts from the statistical process control (SPC) are used along with the XSCHEM statement in the SHEWHART procedure to monitor the means and the standard deviation of the predicted interval target over time. For more information, see [The SHEWHART Procedure – Constructing Charts for Means and Standard Deviations](#) in *SAS/QC User's Guide* and [Statistical Process Control Action Set](#) in *SAS Visual Statistics: Programming Guide*.

Here is an example of the Mean of the Predicted Interval Target chart.

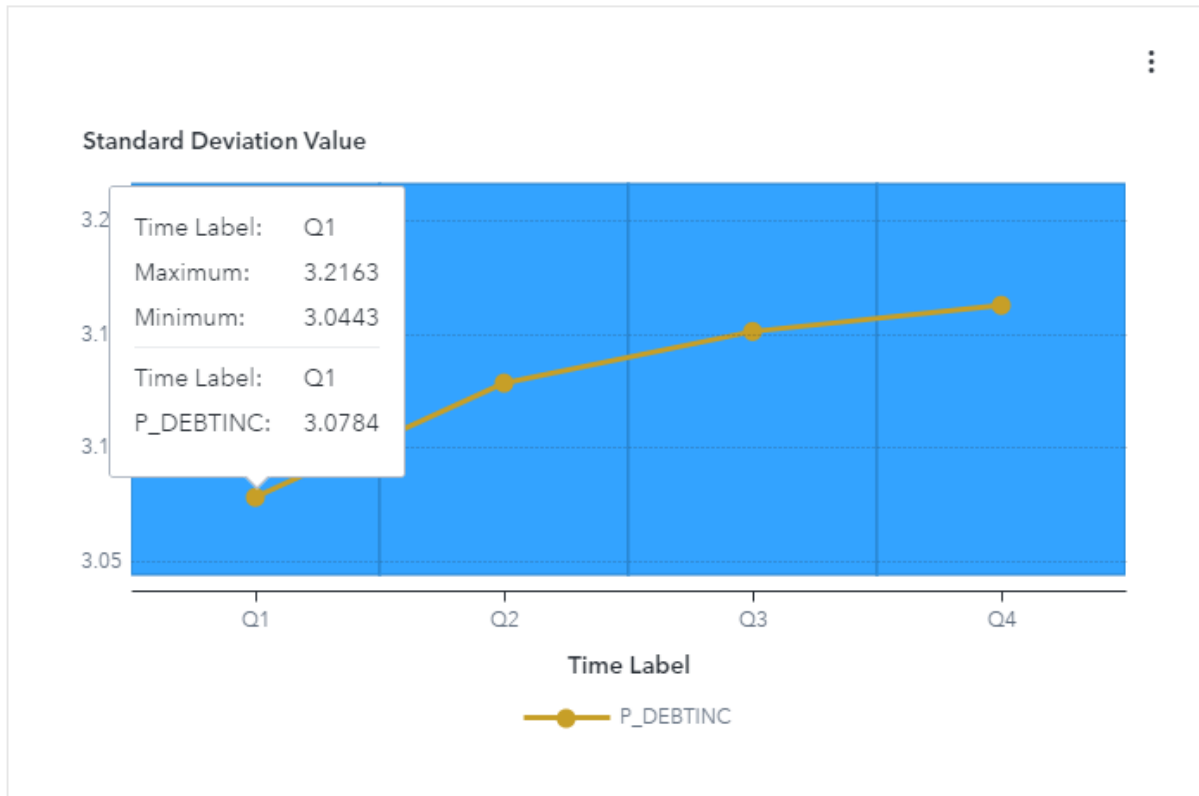


Standard Deviation of the Predicted Interval Target

The Standard Deviation of the Predicted Interval Target chart shows the standard deviation of values for a prediction model with an interval target for different time periods. The central horizontal line represents the weighted average of the standard deviations. The shaded band represents the control limits. If a point in the line chart is outside the shaded band, it indicates that the spread of the predicted interval target values at the corresponding time period has deviated significantly. You can then further scrutinize the model outcomes at that time period.

Concepts from the statistical process control (SPC) are used along with the XSCHEM statement in the SHEWHART procedure to monitor the standard deviation of the predicted interval target over time. For more information, see [The SHEWHART Procedure - Methods for Estimating the Standard Deviation](#) in *SAS/QC User's Guide*.

Here is an example of the Standard Deviation of the Predicted Interval Target chart.



Proportion of Predicted Interval Target Values

The Proportion of Predicted Interval Target Values charts represent the proportion of values for a prediction model with an interval target that are above the 25th, 50th, and 75th percentiles, over a period of time. When the underlying data distribution has changed over time, you need to know how the distribution changed. For example, the distribution might shift its location, the spread might become smaller or larger, the skewness might change, or the kurtosis might change. Therefore, you can use these charts to monitor the proportion of the predicted interval target values that are above specified thresholds. Without loss of generality, the 25th, the 50th, and the 75th percentiles of the predicted interval target values are represented at the first monitoring time (commonly known as the Time 0).

Concepts of the statistical process control (SPC) are used with the p chart to monitor the proportions over time. For more information, see [The SHEWHART Procedure - Constructing Charts for Proportion Nonconforming \(p Charts\)](#) in *SAS/QC User's Guide*

Here are the percentiles that are represented in the Proportion of Predicted Interval Target Values charts:

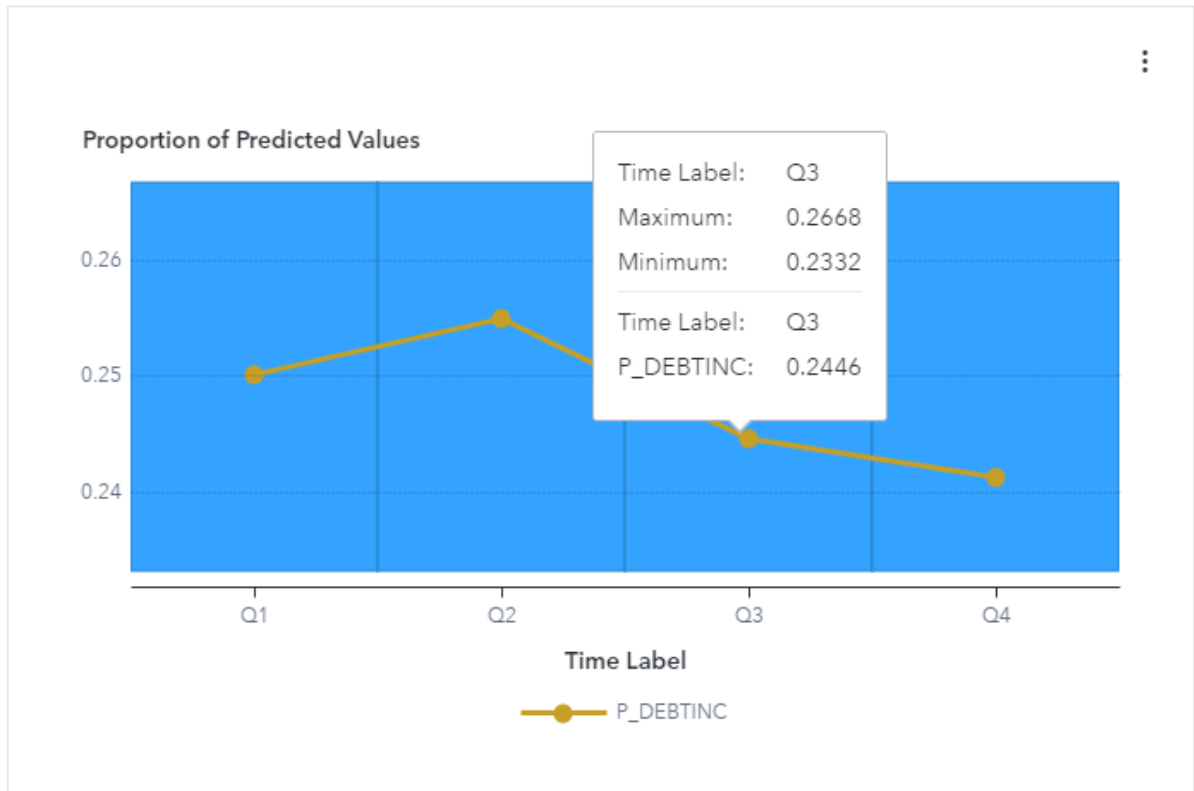
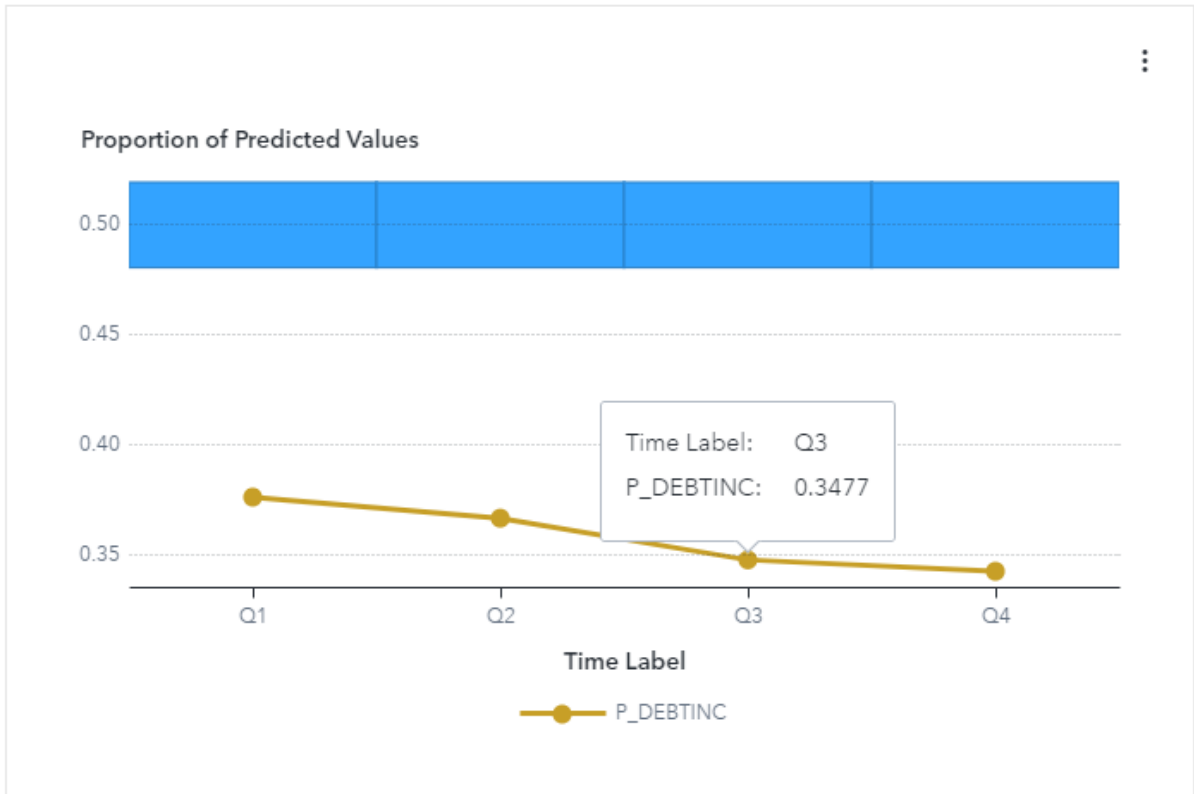
- Above the 25th Percentile (first quartile)
- Above the 50th Percentile (median)
- Above the 75th Percentile (third quartile)

It is recommended that you look at the p charts together. For example, if the 25th percentile p chart at time period Q3 is out of bounds, but the 50th and the 75th percentile p charts at time period Q3 are not, then you can infer that the distribution has moved the values that are below the 50th percentile, but not those that are above the 50th percentile.

Note: When there is a small sample size, the proportion might not reach the expected value.

Here is an example of the Proportion of Predicted Interval Target Values charts.



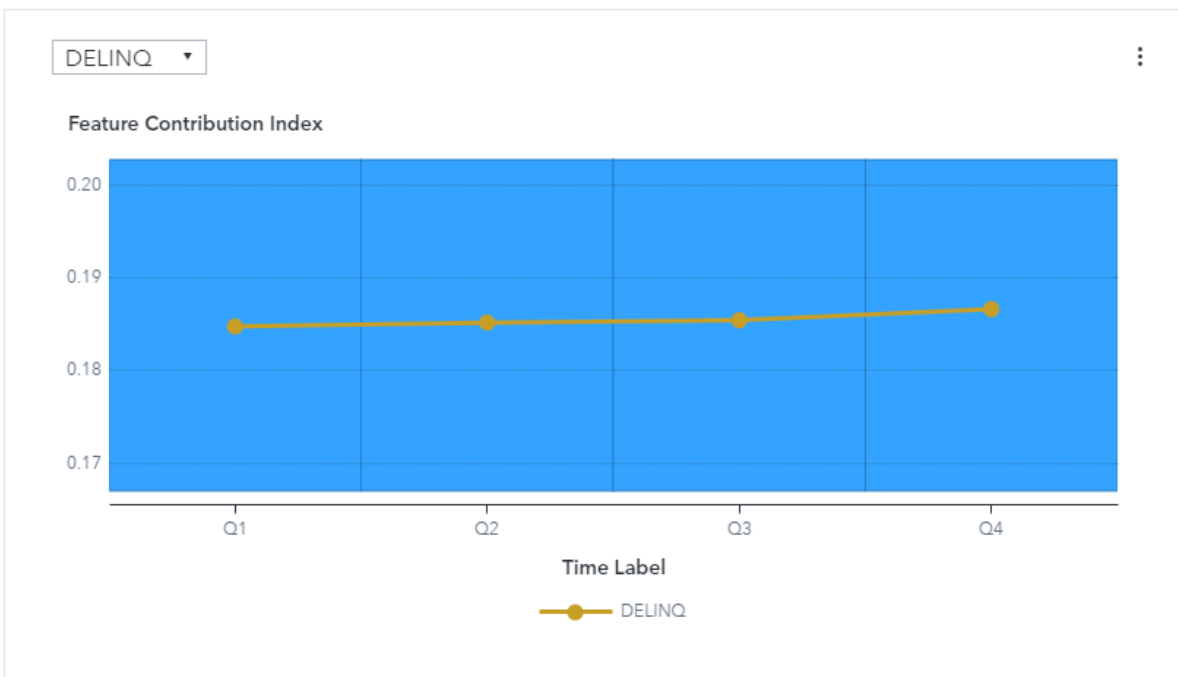


Feature Contribution

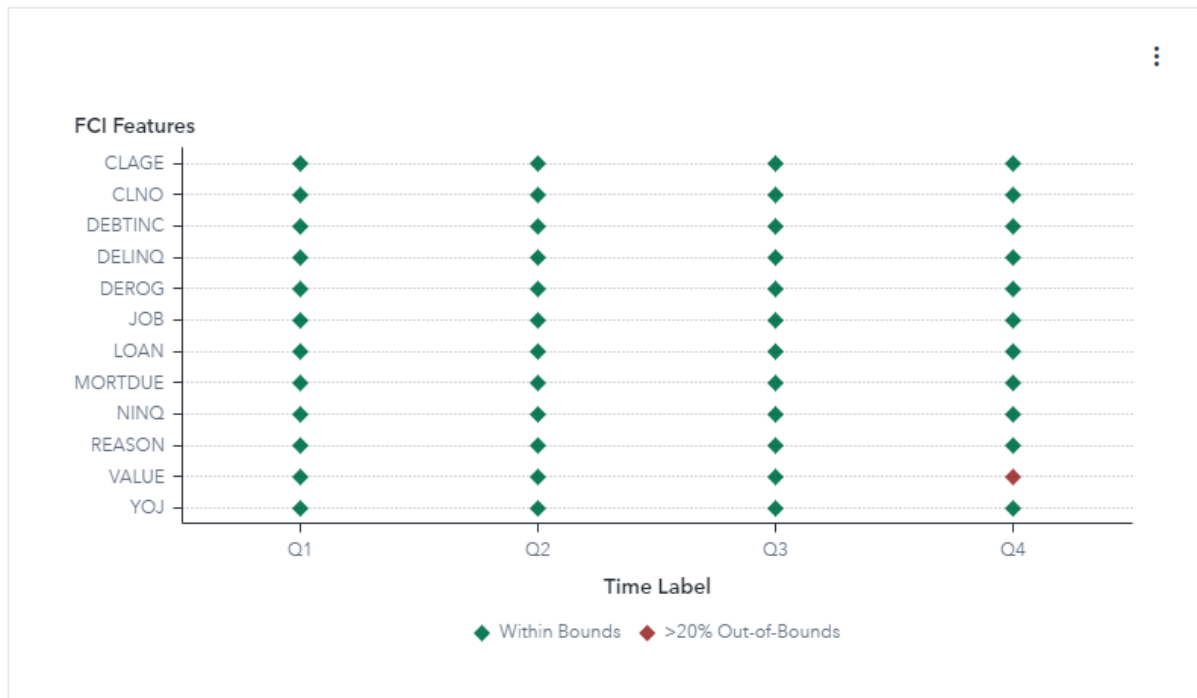
The feature contribution index (FCI) measures the strength of the relationship between each input variable and the predicted numeric output variables. When you are running performance, the FCI macros are used to compute the feature contribution indices for interval and nominal predictors and return the indices in the output data. The output data is then used to create the Feature Contribution chart. The top nine variables with the maximum absolute contribution for first data point are displayed in the drop-down list in descending order. The input data must contain the scored model data or else you must score the model when running performance, in order for the feature contribution index to be computed for the model variables. Otherwise, the Feature Contribution Index chart is not displayed.

The Feature Contribution Index Out-of-Bounds Indicators chart enables you to identify the features that need additional scrutiny.

Here is an example of the Feature Contribution chart.



Here is an example of the Feature Contribution Index Out-of-Bounds Indicators chart.



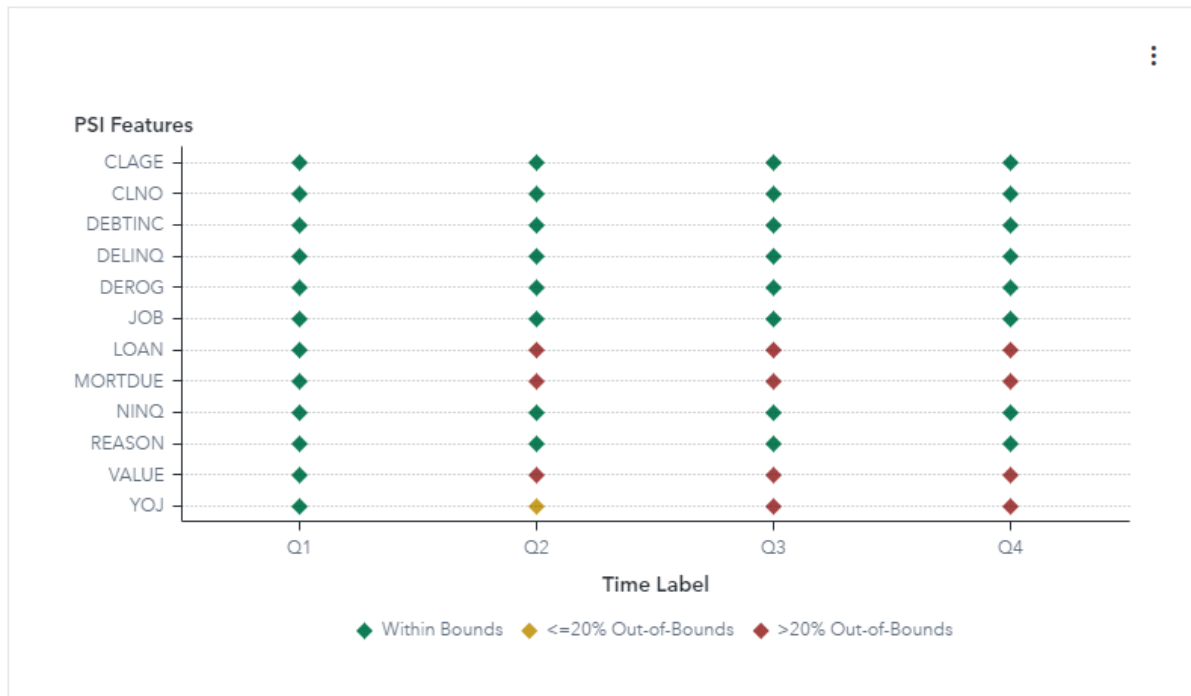
See Also

[“Feature Contribution Index Macros” in SAS Model Manager: Macro Reference](#)

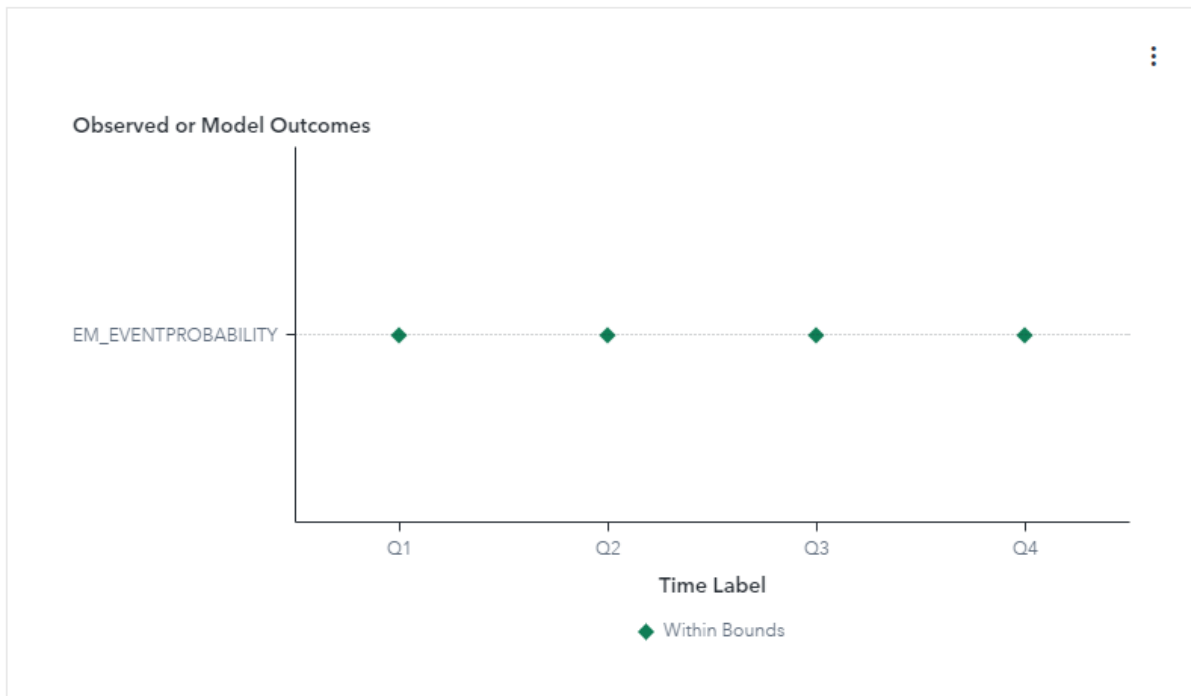
Population Stability

The population stability index (PSI) indicates the degree of changes of a distribution from a baseline. The Population Stability Index Out-of-Bounds Indicators charts enable you to identify the features that need additional scrutiny.

Here is an example of the Population Stability Index Out-of-Bounds Indicators for Input Variables chart.



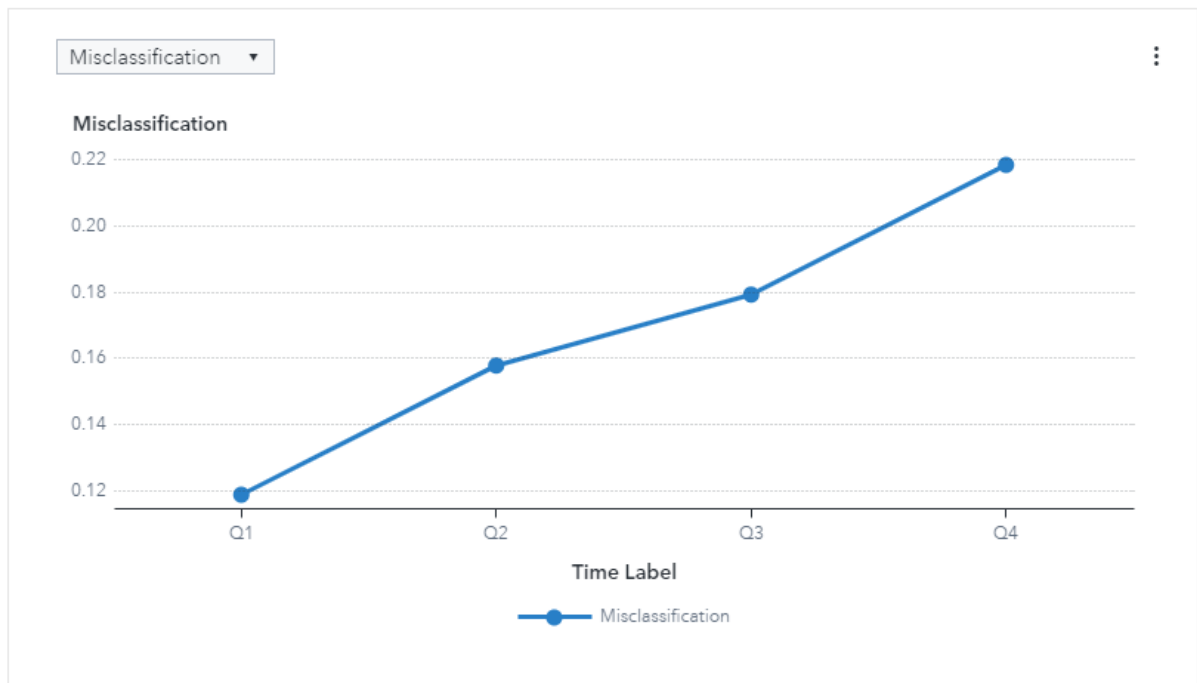
Here is an example of the Population Stability Index Out-of-Bounds Indicators for Output Variables chart.



Standard KPI Trend

The Standard KPI Trend chart shows the trend of the selected standard key performance indicator (KPI) that is calculated over a period of time.

Here is an example of the Standard KPI Trend chart.



Concepts: PMML Support

Overview

PMML is an XML markup language that was developed to exchange predictive and statistical models between modeling systems and scoring platforms. Users can import the majority of standard-compliant PMML models and score them within a SAS environment via the SAS PSCORE procedure.

PROC PSCORE Functionality

The SAS PSCORE procedure generates SAS DATA step score code that is functionally equivalent to the PMML model. The generated score code can be executed on all platforms that are supported by SAS to score the data sets. You can submit the score code in SAS Enterprise Miner via the Program Editor, SAS Enterprise Miner Project code, or within a SAS Enterprise Miner Process Flow Diagram, via the SAS Code node. However, the SAS Enterprise Miner UI environment is not necessary to run the score code.

Note: The PSCORE procedure generates both DATA step code and DS2 code. However, only DATA step model score code is generated when you are registering a PMML model into SAS Model Manager.

Supported Versions

PROC PSCORE currently supports the use of PMML 4.2. Other versions of PMML are not supported for use with PROC PSCORE.

Supported PMML Models

SAS PROC PSCORE supports the following types of PMML models:

- Regression
- Trees
- Neural Networks
- Clustering models
- Scorecard
- Vector Machine
- Naïve Bayes
- Baseline models

The following models are supported on an experimental basis:

- Time Series
- General Regression

Requirements for PROC PSCORE

In order to use PROC PSCORE, you must have SAS 9.4M2 or later, a well formed PMML modeling file, and Write access to the output directory for the DATA step score file. A SAS Enterprise Miner license is not necessary to run PROC PSCORE. SAS 9.4 procedures can also be run on SAS Viya 3.4 or later using SAS Studio 4.4.

PROC PSCORE Usage

```
PROC PSCORE PMML FILE = "<full-pathname-of-PMML-file>"
      DS FILE = "<full-pathname-of-output-DS-file>"
```

PROC PSCORE Example

```
/*Run the PSCORE procedure on a generated PMML file*/
PROC PSCORE PMML FILE = "C:\temp\heart_pmml1.xml"
      DS FILE = "C:\temp\ds_heart_score.sas";
run;
```

See Also

[“SAS Enterprise Miner 15.2 PMML Support” in SAS Enterprise Miner: Reference Help](#)

