## Contents

**Chapter 1 / Introduction to SAS Open Model Manager**  
About Managing Models ........................................... 1  
Sign In to SAS Open Model Manager ............................. 2  
Manage Application Settings ...................................... 3  
High-Level Support Matrix for Primary Functions ............. 3

**Chapter 2 / Managing Data** ........................................ 7  
About Managing Data ............................................. 7  
Create a CASLIB ................................................... 9

**Chapter 3 / Managing Model Repositories** ..................... 11  
About Model Repositories ........................................ 11  
Create a New Repository ......................................... 11  
Rename a Repository ............................................. 12  
Delete a Repository ................................................ 12

**Chapter 4 / Working with Projects** ......................... 15  
About Projects .................................................... 15  
View Projects Dashboard ......................................... 16  
Create a New Project ............................................. 16  
Creating and Importing Models .................................. 18  
Managing Variables .............................................. 18  
Modifying Project Properties .................................... 20  
Managing Project Versions ........................................ 24  
Delete a Project ................................................... 26  
Rename a Project .................................................. 26  
Search for Projects ............................................... 26

**Chapter 5 / Working with Models** ............................. 31  
About Models ...................................................... 31  
View Models Dashboard ........................................... 32  
Add a New Custom Model ......................................... 32  
Import Models ...................................................... 35  
Export a Model ..................................................... 38  
Move or Copy a Model ............................................ 39  
Delete a Model ..................................................... 40  
Rename a Model .................................................... 40  
Manage Model Content and Versions ............................ 48  
Search for Models .................................................. 48

**Chapter 6 / Evaluating Models** ............................... 53  
About Evaluating Models .......................................... 53  
Compare Models .................................................... 54  
Test Models ........................................................ 55  
Validate Published Models ....................................... 59  
Set Champion and Challenger Models ........................... 61  
Monitoring Performance .......................................... 63
Introduction to SAS Open Model Manager

About Managing Models

Using SAS Open Model Manager, you can import, manage, and evaluate open-source analytical models in a common model repository and organize them within projects and folders. All model development and model maintenance personnel, including data modelers, validation testers, scoring officers, and analysts, can use SAS Open 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 and SAS models, as well as model content that is stored in PMML, ONNX, or RDS formats. In addition to models that are 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. You can create performance data from your operational data, provided that it has the required structure (for example, the data contains a target variable). For more information, see “About Managing Data” on page 7.

SAS Open Model Manager is delivered as a singular, portable image that can be deployed in a Docker Community Edition (Docker CE) or Docker Enterprise Edition (Docker EE) Linux environment. It runs without a full SAS Viya installation. For more information, see SAS Open Model Manager for Containers: Deployment Guide and Open Model Manager Resources GitHub repository.
Here are some of the tasks that you can accomplish with SAS Open Model Manager:

- Use a single interface to access all of your business modeling projects. All models are stored in a common model repository. Models can also be accessed in one place using the **Models** category.
- Import models that you develop using SAS software, open-source software, or an open-source programming language such as Python or R. 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 open-source models to container publishing destinations such as Amazon Web Services and Private Docker, as well as a CAS or SAS Micro Analytic Service publishing destination. You can then validate published models within the publishing destinations or score the models 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.

For more information about open-source models, see “Concepts: Open-Source Models” on page 79.

---

**Sign In to SAS Open Model Manager**

To sign in to SAS Open Model Manager:

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

   **Note:** Contact your system administrator if you need the URL for SAS Open Model Manager. The default URL is `https://host_name/SASModelManager`. The last part of the URL (the application name) is case sensitive.

2. Enter a user ID and password.

3. Click **Sign In**.

4. (Optional) If this is your first time signing in to SAS Open Model Manager, the Welcome to SAS window appears. Here you can set up a profile, which enables you to customize some settings.
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
- “About Model Repositories”

High-Level Support Matrix for Primary Functions

Here is a summary of the primary functions that are supported by SAS Open Model Manager. Additional requirements and restrictions might apply, depending on the function that is being performed.

**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 44.

Open-Source Models and Predictive Model Markup

You can import models that were created using an open-source programming language such as Python and R, or an XML-based predictive model interchange format such as PMML.

<table>
<thead>
<tr>
<th>Programming Language or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMML1</td>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>PMML</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes$^5$</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Python</td>
<td>Python</td>
<td>Yes</td>
<td>Yes$^2$</td>
<td>Yes$^3$</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
### 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 imported into the common model repository using the SAS Open Model Manager web application.

**Note:** Model Studio, SAS Visual Analytics, and SAS Studio are not available with the deployment of SAS Open Model Manager.

---

#### Predictive Model Markup Language (PMML)

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. See “Import Models” on page 35.

Models that have a score code type of Python can be scored, if the Python score code is in the correct format. The Python score code must also be in the correct format for publishing and running performance. See “Scoring Python Models” on page 80.

Supported only for publishing models to CAS, SAS Micro Analytic Service, Amazon Web Services (AWS), and Private Docker publishing destinations. See “Requirements and Restrictions” on page 74.

Supported only for publishing models to Amazon Web Services (AWS) and Private Docker publishing destinations. See “Requirements and Restrictions” on page 74.

Supported only for user-provided scored data. See “Monitoring Performance” on page 63.

---

#### SAS Viya Models

<table>
<thead>
<tr>
<th>Product or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Visual Data Mining and Machine Learning</td>
<td>Analytic store</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>DS2 multi-type</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Product or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Visual Statistics</td>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Note:** You can build SAS Visual Statistics models with SAS Visual Analytics, or with SAS Viya modeling procedures within SAS Studio.
### High-Level Support Matrix for Primary Functions

<table>
<thead>
<tr>
<th>Product or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Visual Text Analytics</td>
<td>SAS program</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> You can build SAS Visual Text Analytics models with Model Studio or with SAS Viya modeling procedures within SAS Studio.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SAS 9.4 Models

Models that you create with SAS 9.4 can also be imported into SAS Open Model Manager on SAS Viya.

<table>
<thead>
<tr>
<th>Product or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base SAS or other code editor</td>
<td>SAS program</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>PMML¹</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Analytic store</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Note:</strong> Applies only to SAS analytic store (SASAST) files. SAS package (SPK) files are not supported.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product or Tool</th>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
<th>Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS HPFOREST and HPSVM procedures</td>
<td>Analytic store</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

¹ 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 35.
System-Supplied Score Code Types

**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.

<table>
<thead>
<tr>
<th>Model Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Monitor Performance</th>
<th>Model Compare and Assess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic store</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DATA step</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DS2 embedded process</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes(^3)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DS2 multi-type</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DS2 package</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes(^4)</td>
<td>Yes(^6)</td>
<td>Yes</td>
</tr>
<tr>
<td>PMML(^1)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
<td>Yes</td>
</tr>
<tr>
<td>Python</td>
<td>Yes</td>
<td>Yes(^2)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R</td>
<td>Yes</td>
<td>No</td>
<td>No(^5)</td>
<td>Yes(^6)</td>
<td>Yes</td>
</tr>
<tr>
<td>SAS program</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Additional Score Code Types**

<table>
<thead>
<tr>
<th>Score Code Type</th>
<th>Import</th>
<th>Score</th>
<th>Publish</th>
<th>Compare and Assess</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
</tr>
<tr>
<td>CAS language</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
</tr>
<tr>
<td>Java</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
</tr>
<tr>
<td>Lua</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes(^6)</td>
</tr>
</tbody>
</table>

\(^1\) 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 35.

\(^2\) Models that have a score code type of Python can be scored, if the Python score code is in the correct format. The Python score code must also be in the correct format for publishing and running performance. See “Scoring Python Models” on page 80.

\(^3\) Supported only for publishing models to CAS publishing destinations. See “Requirements and Restrictions” on page 74.

\(^4\) Supported only for publishing models to SAS Micro Analytic Service. See “Requirements and Restrictions” on page 74.

\(^5\) Supported only for publishing models to Amazon Web Services (AWS) and Private Docker publishing destinations. See “Requirements and Restrictions” on page 74.

\(^6\) Supported only for user-provided scored data. See “Monitoring Performance” on page 63.
Managing Data

About Managing Data

The Manage Data window is accessible from both the Models and Projects category views in SAS Open 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.

Create a CASLIB

1. In 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) 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 Administration: CAS Authorization.

e Click Save.

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

1. In 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

About Model Repositories

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 Open 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 Open Model Manager: Administrator’s Guide.

To access the settings for repositories, click the user button in the application bar and select Settings $\Rightarrow$ SAS Open Model Manager $\Rightarrow$ 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 $\Rightarrow$ SAS Open Model Manager $\Rightarrow$ 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 Open 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 Open Model Manager ➔ Repositories.
2 Select a repository and click ⌘.
3 In the confirmation message, click Delete.
# Working with Projects

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Projects</td>
<td>15</td>
</tr>
<tr>
<td>View Projects Dashboard</td>
<td>16</td>
</tr>
<tr>
<td>Create a New Project</td>
<td>16</td>
</tr>
<tr>
<td>Creating and Importing Models</td>
<td>18</td>
</tr>
<tr>
<td>Managing Variables</td>
<td>18</td>
</tr>
<tr>
<td>Add Variables from a Data Table</td>
<td>18</td>
</tr>
<tr>
<td>Add Custom Variables</td>
<td>19</td>
</tr>
<tr>
<td>Edit Variables</td>
<td>19</td>
</tr>
<tr>
<td>Delete Variables</td>
<td>19</td>
</tr>
<tr>
<td>Modifying Project Properties</td>
<td>20</td>
</tr>
<tr>
<td>Set Project General Properties</td>
<td>20</td>
</tr>
<tr>
<td>Add Tags</td>
<td>23</td>
</tr>
<tr>
<td>Add User-Defined Properties</td>
<td>24</td>
</tr>
<tr>
<td>Managing Project Versions</td>
<td>24</td>
</tr>
<tr>
<td>Create a New Project Version</td>
<td>25</td>
</tr>
<tr>
<td>Manage Project Versions</td>
<td>25</td>
</tr>
<tr>
<td>Delete a Project</td>
<td>26</td>
</tr>
<tr>
<td>Rename a Project</td>
<td>26</td>
</tr>
<tr>
<td>Search for Projects</td>
<td>26</td>
</tr>
<tr>
<td>Advanced Search for Projects</td>
<td>26</td>
</tr>
</tbody>
</table>

## About Projects

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.

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. The grouping of the models can be for a specific period of time.
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 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 📊 in the toolbar to hide or show the dashboard panel.

---

### Create a New Project

1. Click ☑️ to navigate to the Projects category view.
2. Click New Project. The New Project window appears.
3. Enter a name for the project.

   Note: The initial version 1.0 is displayed and reflects the level for sequential versions.

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.
Note: The value for Model function cannot be modified after you click Save. If you want to monitor performance of a model, it is recommended to select Classification or Prediction when creating a project. Only projects with a model function of Classification or Prediction can be monitored for performance.

6 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 within the Settings window. For more information, see “About Model Repositories” on page 11.

7 Click Save.
Creating and Importing Models

After you create a project, you import models into 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 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.

For more information, see the following:

- "Add a Custom Model within a Project" on page 34
- "Import Models into a Project" on page 37
- "Copy a Model" on page 39
- "About Evaluating Models" on page 53

Managing Variables

Input variables and output variables can be added to both project and model objects on the Variables tab. 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 does not exist, 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 Select Variables window appears.
4 Select input or output for the variable type.
5 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.
6 Click OK.
7 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 

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 three types: General, Tags, and User-Defined.

Set Project General Properties

**General Properties** 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 general properties, click the **Properties** tab, modify the property values, and then click .

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specifies the name of the project. A project can be renamed only from the Projects category view.</td>
</tr>
<tr>
<td>Description</td>
<td>Specifies the description of the project.</td>
</tr>
<tr>
<td>Model function</td>
<td>Specifies the type of output that your predictive model project generates. After it has been declared, the <strong>Model function</strong> property for a project cannot be changed. Ensure that the types of models that you are going to use in the project fit within the selected model function type. For more information, see Table 4.2 on page 22.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Operation status</td>
<td>Specifies the current state of the project:</td>
</tr>
</tbody>
</table>
|                           | Under Development  
|                           | indicates that the project has started but a champion model is not yet in production.                                                             |
|                           | Active  
|                           | indicates that a champion model for this project is in production.                                                                               |
|                           | Inactive  
|                           | indicates that the champion model is temporarily suspended from production.                                                                     |
|                           | Retired  
|                           | indicates that the champion model for this project is no longer in production.                                                                   |
| 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.                                                                           |
| Default train table       | Specifies the **Default train table** is also 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. |
| Target variable           | Specifies the name of the target variable that was used to train the model.                                                                    |
| 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 1, 0 or Yes, No. Nominal class target values might be Low, Medium, High. These values are for information only. |
| Target level              | Specifies the target level of binary, nominal, ordinal, or interval.                                                                           |
Table 4.2  Types of Model Functions

<table>
<thead>
<tr>
<th>Model Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>Function for any model that is not Prediction, Classification, or Segmentation.</td>
</tr>
<tr>
<td>Model Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Classification</td>
<td>Function for models that have target variables that contain binary, categorical, or ordinal values.</td>
</tr>
<tr>
<td>Clustering</td>
<td>Function for segmentation or clustering models.</td>
</tr>
<tr>
<td>Forecasting</td>
<td>Function for models used to forecast future data based on past data.</td>
</tr>
<tr>
<td>Prediction</td>
<td>Function for models that have interval targets with continuous values.</td>
</tr>
<tr>
<td>Text analytics</td>
<td>Function for SAS Visual Text Analytics models.</td>
</tr>
<tr>
<td>Transformation</td>
<td>Function for models used to determine mathematical functions that can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables to improve the fit of your model.</td>
</tr>
</tbody>
</table>

Add Tags

You can add one or more tags to a project. When you add tags to a project, they are added to a master list of tags that is available to be added to other projects within the same repository.

To add a tag:

1. On the **Properties** tab of a project, select **Tags**.
2. Select an existing tag or enter a name for a tag, and then click ✪.

   **Note:** The tag 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 tags.

3. Click ✖.

To delete a tag, click ✖ at the right-side of the row, and then click ✖. The tag is removed from 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 **Add** 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 **Add**.
4. Click **OK**.

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 **OK**.

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

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.
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 from the Actions menu within the toolbar.

Advanced Search for Projects

To perform an advanced search for projects:
1. Click and select **Search for projects**. The Search for Projects window appears.

   ![Search for Projects window](image)

   - **Repository**: The default is **All repositories**.
   - **Name**:
   - **Description**:
   - **Model function**:
   - **Target variable**:
   - **Input variables**: Select or enter variables
   - **Output variables**: Select or enter variables

   No projects contain user-defined properties.

2. To search for a project in a specific repository, click in the **Repository** dropdown list, and select the desired repository. **All repositories** is selected as the default for the advanced search.

3. Enter a possible value in the **Name** field to search for a project by name.

4. Enter a possible value in the **Description** field to search for a project by description.

5. Enter a value in the **Model function** field to search for a project by model function.

6. Enter a value in the **Target variable** field to search for a project by target variable.

7. To search for a project by input 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 \( \rightarrow \). You can also click \( \leftrightarrow \) 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 \( \leftarrow \). Click \( \leftarrow \) to remove all of the variables from the available items list.

**TIP** Click \( \times \) 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.
To search for user-defined properties that are associated with models or project objects, specify one or more property name and value pairs, or specify property names only.

a. Click \( \text{➕} \) to add a row to the **User-defined properties** table.

b. Click \( \text{▼} \) 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 \( \text{-minus sign} \) to remove the user-defined property from the search criteria.

9. 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.
About Models

You can add new custom models or import existing models using the SAS Open Model Manager web application. Models can be stored within a folder or project version in the SAS Open 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 search for models. Models that are within a project version can be published from the Projects category view.
Several options exist for importing open-source models such as Python and R into SAS Open 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 Open Model Manager Resources GitHub repository.

**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 📅 in the toolbar to hide or show the dashboard.

![Dashboard Image]

**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.

**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.

3 Enter a name for the new custom model.

4 (Optional) Enter a description for the new model.

5 Click **Folder** 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 within the Settings window. For more information, see “About Model Repositories” on page 11.

6 (Optional) Click **Folder**, 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.

![New Custom Model window]

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. 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. Only specific file types can be used to 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.

**Note:** You cannot score, publish, monitor performance, or compare 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:

**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. Models that are created using PMML 4.2 support DATA step score code. The file extensions can be .xml or .pmml, provided that the file contains valid PMML XML code.

**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 (SASAST) file**

a file that is created by using the ASTORE procedure or SAS Visual Data Mining and Machine Learning modeling procedures. The SASAST file contains the model scoring files and model input and output variables for an 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 within the Settings window. For more information, see “About Model Repositories” on page 11.

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 Open Model Manager, the analytic store file is included in the ZIP file. When you re-import the analytic store model to the same system or another system, the analytic store file is 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.
To import model into a project version:

1. Click \(\text{Projects}\) 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 \(\text{Add model}\) 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 Open Model Manager, the analytic store file is included in the ZIP file. When you re-import the
analytic store model to the same system or another system, the analytic store file is 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.

![Import Models dialog box](image)

**TIP** To remove extra lines, click ![Remove extra lines](image) before you click **Import**.

7 Click **Import**.

---

**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 file is included in the ZIP file. If you re-import the analytic store model to the same system or another system, the analytic store file is copied to the ModelStore caslib.

To export a model:

1 Select a model from the list.

2 Click ![Export](image) 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 Open Model Manager: Administrator’s Guide.

Move a Model

To move a model:

1. Note: Only models located within a folder can be moved.
2. Click to navigate to the Models category view.
3. Select a model from the list.
4. Click and select Move. The Choose a Location window appears.
5. Navigate to the folder or project version that you want to move the model to.
6. 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 Import 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 into 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 open a model from the Models category view and from the Models page of a project.
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 any type of file 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 viewed by another application.

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.

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. 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 does not exist, 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 Select Variables window appears.

4. Select input or output for the variable type.

5. Select the variables that you want to add and click \( \rightarrow \). You can also click \( \leftrightarrow \) to add all of the variables from the available items list.

6. Click OK.

7. 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, 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, no values are displayed in the Model Output Variable column, and 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 two types: General and User-Defined.
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>
<thead>
<tr>
<th><strong>Table 5.1</strong> List of General Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property</strong></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Project name</td>
</tr>
<tr>
<td>Project version</td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>Score code type</td>
</tr>
</tbody>
</table>

*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.

Depending on score code type, you can score, monitor, or publish a model. For more information, see “High-Level Support Matrix for Primary Functions” on page 3.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train table</td>
<td>Specifies the <strong>Train table</strong> 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.</td>
</tr>
<tr>
<td>Train code type</td>
<td>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.</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Specifies the computational algorithm that is used for the selected model.</td>
</tr>
</tbody>
</table>
| Input variable type              | 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 **score** and **train**.  
**Note:** If both files are included with the registered model, the property value is set to **score**. |
| 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 event probability variable | Specifies the output event probability variable name, when the **Model function** property is set to **Classification**, **Analytical**, **Forecasting**, or **Transformation**. |
| Output prediction variable       | The output prediction variable name, when the **Model function** property is set to **Prediction**, **Analytical**, **Forecasting**, or **Transformation**. |
### Property Description

- **Output segmentation variable**: The output segmentation variable name, when the Model function property is set to Clustering, Analytical, Forecasting, or Transformation.

- **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

<table>
<thead>
<tr>
<th>Model Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical</td>
<td>Function for any model that is not Prediction, Classification, or Segmentation.</td>
</tr>
<tr>
<td>Classification</td>
<td>Function for models that have target variables that contain binary, categorical, or ordinal values.</td>
</tr>
<tr>
<td>Clustering</td>
<td>Function for segmentation or clustering models.</td>
</tr>
<tr>
<td>Forecasting</td>
<td>Function for models used to forecast future data based on past data.</td>
</tr>
<tr>
<td>Prediction</td>
<td>Function for models that have interval targets with continuous values.</td>
</tr>
<tr>
<td>Text categorization</td>
<td>Function for SAS Visual Text Analytics categorization models.</td>
</tr>
<tr>
<td>Model Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Text extraction</td>
<td>Function for SAS Visual Text Analytics concepts models.</td>
</tr>
<tr>
<td>Text sentiment</td>
<td>Function for SAS Visual Text Analytics sentiment models.</td>
</tr>
<tr>
<td>Text topics</td>
<td>Function for SAS Visual Text Analytics topics models.</td>
</tr>
<tr>
<td>Transformation</td>
<td>Function for models used to determine mathematical functions that can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables to improve the fit of your model.</td>
</tr>
</tbody>
</table>

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:

2. Click Add Property. The Add Property window appears.

   **TIP** If user-defined properties already exist, click Add 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 Add.
4. 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 .
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 Open 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:

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.

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 from the Actions menu within the toolbar.

Advanced Search for Models
To perform an advanced search for models:

1. Click  and select Search for models. The Search for Models window appears.

2. To search for a model in a specific repository, click  in the Repository dropdown list, and select the desired repository. All repositories is selected as the default for the advanced search.

3. Enter a possible value in the Name field to search for a model by name.

4. Enter a possible value in the Algorithm field to search for a model by algorithm.

5. Enter a user name in the Modeler field to search for a model by modeler.
6 Enter a value in **Target variable** field to search for a model by target variable.

7 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.

   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 \( \rightarrow \). You can also click \( \leftrightarrow \) 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 \( \leftarrow \). Click \( \leftrightarrow \) to remove all of the variables from the available items list.

TIP Click \( \leftrightarrow \) 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.
To search for user-defined properties that are associated with models or project objects, specify one or more property name and value pairs, or specify property names only.

- Click + to add a row to the User-defined properties table.
- Click ▼ in the Name drop-down list, and select a user-defined property.
- (Optional) Enter the value for the user-defined property in the Value column.
- Select a row and click to remove the user-defined property from the search criteria.

Click Search. The Models category view opens and the model or models that meet the advanced search criteria are listed.

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

### About 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 Open 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

<table>
<thead>
<tr>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Evaluating Models</td>
<td>53</td>
</tr>
<tr>
<td>Compare Models</td>
<td>54</td>
</tr>
<tr>
<td>Test Models</td>
<td>55</td>
</tr>
<tr>
<td>About Testing Models</td>
<td>55</td>
</tr>
<tr>
<td>Create and Run a New Test</td>
<td>56</td>
</tr>
<tr>
<td>Edit a Test</td>
<td>58</td>
</tr>
<tr>
<td>Delete a Test</td>
<td>58</td>
</tr>
<tr>
<td>Validate Published Models</td>
<td>59</td>
</tr>
<tr>
<td>About Validating Published Models</td>
<td>59</td>
</tr>
<tr>
<td>Edit and Run a Publishing Validation Test</td>
<td>61</td>
</tr>
<tr>
<td>Delete a Publishing Validation Test</td>
<td>61</td>
</tr>
<tr>
<td>Duplicate a Publishing Validation Test</td>
<td>61</td>
</tr>
<tr>
<td>Set Champion and Challenger Models</td>
<td>61</td>
</tr>
<tr>
<td>About Champion and Challenger Models</td>
<td>61</td>
</tr>
<tr>
<td>Set a Champion Model</td>
<td>62</td>
</tr>
<tr>
<td>Set a Challenger Model</td>
<td>62</td>
</tr>
<tr>
<td>Clear Model Role</td>
<td>63</td>
</tr>
<tr>
<td>Monitoring Performance</td>
<td>63</td>
</tr>
<tr>
<td>About Monitoring Performance</td>
<td>63</td>
</tr>
<tr>
<td>Set Project Properties</td>
<td>64</td>
</tr>
<tr>
<td>Naming Requirements for Input Data Tables</td>
<td>64</td>
</tr>
<tr>
<td>Create a New Performance Definition</td>
<td>65</td>
</tr>
<tr>
<td>Specify Definition Details for System Scores Data</td>
<td>67</td>
</tr>
<tr>
<td>Specify Definition Details for User Provides Scored Data</td>
<td>68</td>
</tr>
<tr>
<td>Run Performance and View Results</td>
<td>69</td>
</tr>
<tr>
<td>Edit Performance Definition</td>
<td>70</td>
</tr>
<tr>
<td>Clear Definition Content or Performance Results</td>
<td>71</td>
</tr>
</tbody>
</table>
- create and run a scoring test on a model
- monitor performance of a model
- publish models to CAS, SAS Micro Analytic Service, and container destinations
- 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 Python packages such as sklearn and scipy.stats. 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 Open Model Manager.

**Note**: You can create the JSON files that are needed to display the fit statistics, and lift and ROC plots by using Python code and packages such as sklearn and scipy.stats. For more information, see the examples in the Open Model Manager Resources GitHub repository.

To compare models:

1. Select one or more models.
2. From the Models category view, click ![compare_icon](compare.png) 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 Stat
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 Open 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, models must be assigned 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, or Python. For more information, see “Assign Model File Roles” on page 41 and “Set Model General Properties” on page 44.

Note: Models that have a score code type of Python can be scored only when the Python score code is in the correct format. For more information, see “Scoring Python Models” on page 80.

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 59.

Note: R models can be scored only within a container publishing destination. For more information, see “Scoring R Models” on page 81.
Create and Run a New Test

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 Window](image)

   - **Name:** Enter a name for the test if you do not want to use the default name.
   - **Description:** Enter a description for the test.
   - **Model:** Click Choose Model and select a model to test.
   - **Input table:** Select the input table for the test, and click OK.

   **TIP** If the desired data table does not exist, 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 Open 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.

Note: SAS Open 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.

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 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.

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 The test is not ready to run. The test definition is not complete, or it might contain errors.</td>
<td></td>
</tr>
<tr>
<td>🔄 The test is defined correctly and is ready to run.</td>
<td></td>
</tr>
<tr>
<td>🔄 The test is running.</td>
<td></td>
</tr>
<tr>
<td>Icon</td>
<td>Status</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>✔</td>
<td>The test completed successfully.</td>
</tr>
<tr>
<td>🔄</td>
<td>The test completed, but warnings were issued in the SAS log. The URI to the log file is shown on the Test Results page.</td>
</tr>
<tr>
<td>✗</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 Open 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.

---

**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.

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 41 and “Set Model General Properties” on page 44.

Note: Models that have a score code type of Python can be validated, if the Python score code is in the correct format. For more information, see “DS2 Interface to Python” in SAS Micro Analytic Service: Programming and Administration Guide.

Edit and Run a Publishing Validation Test

1. Click the Scoring tab of a project, and then click the Publishing Validation tab.
2. Click on a test name. The Edit Publishing Validation Test window appears.
3 (Optional) Change the name of the test.

4 (Optional) Enter a description for the test.

5 Click \( \square \), select the input table for the test, and click OK.

**TIP** If the desired data table does not exist, 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 (Optional) Expand the Advanced section to display the advanced options.

7 (Optional) By default, the library location is the same as that for the input data table. Click \( \square \) to specify a different library to store the new test output table that is created when the test is run.

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 57.

9 Click **in the Results column** to view the results of the test. The **Test Results** page displays information about the test, including the URLs for the test definition and test results. It also includes URLs to the SAS code that was run by SAS Open 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.

---

**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.

Set a Champion Model

1. Click the Models tab of a project.
2. Select a model, click 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, a is displayed in the Role column.

Set a Challenger Model

1. Click the Models tab of a project.
2. Select a model, click 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, \( \text{role} \) 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 \( \text{role} \), 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.

3 In the confirmation message, click Yes.

---

Monitoring Performance

About Monitoring Performance

SAS Open 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 Open 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 performance results tables and several charts, such as Variable Distribution, Characteristic, Stability, Lift, Gini, ROC, Kolmogorov-Smirnov (KS), and Average Squared Error (ASE) charts. Only the charts are displayed on the Performance tab of a project.

Note that model performance can sometimes be improved by tuning or refitting the model, 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), and Average Squared Error (ASE).
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, or Python. If a model has a score code type such as R, you must provide scored data as input when running performance monitoring. For more information, see “Set Model General Properties” on page 44.

For more information and examples of the performance monitoring reports, see “Concepts: Performance Monitoring” on page 82.

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 done.

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 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 should 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
- prefix_sequenceNumber_timeLabel_modelUUID
- prefix_sequenceNumber_timeLabel_modelUUID_modelRole

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 should 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 Q12019, or another meaningful label. Spaces are not recommended in the label name.

model UUID
The model UUID is the fourth level of the table name. If you provide user-scored data and select a library to use for performance monitoring, the data table names must contain the UUID of the model.

model role
The model role is the fifth level of the table name, and is optional. Valid values are champion or challenger.

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. Select an option from the Data scoring method drop-down list.

   **System scores data**
   When you choose this data method, the model score code is used to score the data before generating the performance results.
User provides scored data
When you choose this data method, the data tables must contain the predicted values for the scored model.

3 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 64.

Use a single table
Click , select a data table and click OK.

**TIP** If the desired data table does not exist, 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 cannot contain underscores, but spaces are not recommended in the prefix name or the table name.

Note: When you select both User provides scored data and Use a library that contains tables with a specified prefix, be aware of the following special requirement. Your data table names must contain the UUID of the model. If they do not, the system assumes that the data is associated with the project champion model.

4 Click OK. The Definition page appears.

5 Specify the definition details for the chosen data processing method.

- System scores data
- User provides scored data

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.
Specify Definition Details for System Scores Data

1. Enter a name for the performance definition (for example, My_Performance_Definition1).

2. (Optional) Enter a description.

3. (Optional) View the Tables section. The single data table or the library and prefix that you previously specified are displayed.

   **IMPORTANT** If you select a different input data table, make sure that it follows the naming requirements. For more information, see “Naming Requirements for Input Data Tables” on page 64.

4. In the Models section, select the Use referenced models option.

5. Accept the default option to use the current champion.

6. (Optional) In the Report Settings section, you can view the input and output variables to include in the report results.

   **TIP** By default, all of the project input variables and output variables are selected. Click Choose Variables to modify which variables are included in the report results and click OK.

7. View the Project Properties section to verify that the required properties have been set.

8. (Optional) View the Output Settings section title to verify the CAS library for the output tables. The default value is cas-shared-default/ModelPerformanceData. The input data table and the output data library must be on the same CAS server.

9. Click ✅. You are returned to the Performance tab.

10. Run performance and view results.
Specify Definition Details for User Provides Scored Data

1. Enter a name for the performance definition (for example, `My_Performance_Definition1`).

2. (Optional) Enter a description.

3. (Optional) View the Tables section title to expand it. The single data table or the library and prefix that you previously specified are displayed.

   IMPORTANT If you select a different input data table, make sure that it follows the naming requirements. For more information, see “Naming Requirements for Input Data Tables” on page 64.

4. In the Models section, click Choose Model, and select a model. Click OK.

   ![Choose Models](image)

   Items selected: 0

<table>
<thead>
<tr>
<th>Name</th>
<th>Project Version</th>
<th>Model Function</th>
<th>Role</th>
<th>Modified By</th>
<th>Date Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>QS.Reg1 (1.0)</td>
<td>Version 1 (1.0)</td>
<td>classification</td>
<td>sasdemo</td>
<td>Oct 31, 2019 01:19 PM</td>
<td></td>
</tr>
<tr>
<td>QS.Tree1 (3.0)</td>
<td>Version 1 (1.0)</td>
<td>classification</td>
<td>Champion</td>
<td>Oct 31, 2019 02:09 PM</td>
<td></td>
</tr>
</tbody>
</table>

5. (Optional) In the Report Settings section, you can view the input and output variables to include in the report results.

   TIP By default, all of the project input variables and output variables are selected. Click Choose Variables to modify which variables are included in the report results and click OK.

6. View the Project Properties section to verify that the required properties have been set.

7. (Optional) View the Output Settings section to verify the CAS library for the output tables. The default value is `cas-shared-default/ModelPerformanceData`. 
8 Click  You are returned to the **Performance** tab.

9 **Run performance and view results.**

---

**Run Performance and View Results**

**Note:**

If the data tables that were previously specified in the performance definition exist in a library, but are not loaded in a CAS session, they are loaded at run time.

1 Click **Run** on the **Performance** tab.

2 Click **View Job History** to view the current status of the performance job.

3 When the job is complete, click **Close** to return to the **Performance** tab and view the results.
Figure 6.1  Example of Performance Results for a Classification Model with a Binary Target

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), and Average Squared Error (ASE).

4 Click Close.

Edit Performance Definition

Here are some reasons to edit the performance definition details:

- The model role has changed or the model you previously selected is no longer available.
- The project properties or variables have changed.
Click **Edit Definition**, modify the performance definition details, and then click .

For information about the definition details, see “Specify Definition Details for System Scores Data” on page 67 and “Specify Definition Details for User Provides Scored Data” on page 68.

---

### Clear Definition Content or Performance Results

You can clear the content of an existing performance definition, or clear performance results and delete output tables.

1. Click **Clear All** on the **Performance** tab.

2. Select the check box for the definition content or data that you want to clear.

   ![Clear All dialog]

   Select an option to indicate which content to clear:

   - ✔ Clear performance definition content
   - □ Clear performance results and delete output tables

3. Click **Clear**.
About Publishing Models

You can publish models to a publishing destination, so that the destination can be used by other applications for tasks such as scoring. Models can be published to destinations that are defined for CAS and SAS Micro Analytic Service, as well as container destinations such as Amazon Web Services and Private Docker.

Models can be published from different locations within the SAS Open Model Manager web application. The project champion model and its challengers can be published from the Projects category view. Models can also be published from the Models tab of a project.

By default, two publishing destinations are available: SAS Micro Analytic Service (maslocal) and CAS (CAS_Public). Before you publish a model to the SAS Micro Analytic Service (maslocal) destination, you must first update the number of threads for SAS Micro Analytic Service. Other publishing destinations can be defined by a SAS administrator. For more information, see “Configuring Publishing Destinations” in SAS Open Model Manager: Administrator’s Guide.

Note: When you publish a model from a project version, the system creates a publishing validation test. 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 59.
Requirements and Restrictions

Model Score Code Type

Before you can publish a model, you must first set its score code type. Only models with a score code type of DATA step, DS2 package, DS2 embedded process, DS2 multi-type, Analytic store, Python, and R can be published.

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.

Depending on the model score code type, the following restrictions could affect where you can publish a model:

- Models with the following score code types can be published to CAS: DATA step, DS2 embedded process, DS2 multi-type, or Python.
- Models with the following score code types can be published to SAS Micro Analytic Service: DATA step, DS2 package, DS2 multi-type, or Python.
- Models with the following score code types can be published to a container destination such as Amazon Web Services or Private Docker: Python or R. For more information, see “Concepts: Open-Source Models” on page 79.

For more information, see “Set Model General Properties” on page 44.

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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Maximum Length</th>
<th>Requirements and Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container destinations</td>
<td>128</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The supported container destinations are Amazon Web Services and Private Docker.</td>
</tr>
<tr>
<td>SAS Micro Analytic Service</td>
<td>100</td>
<td>The published name cannot contain the following characters: ! @ # $ % ^ &amp; * ( )</td>
</tr>
<tr>
<td>SAS Cloud Analytic Services (CAS)</td>
<td>128</td>
<td>The published name cannot contain single or double quotation marks.</td>
</tr>
</tbody>
</table>

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 table.

3. (Optional) In the Items to publish table, 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 75.

4. (Optional) If you have previously published the project champion model or its challengers, select the check box in the Replace column in order to replace the previously published item of the same name in the same destination.

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. Before you publish a model to the SAS Micro Analytic Service (maslocal) destination, you must first update the number of threads for the SAS Micro Analytic Service.

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 Open 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.

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:** A publishing validation test is created after a model has been published. For more information, see “Validate Published Models” on page 59.

---

**Publish Models**

You can publish models from the **Models** tab of a project, including the champion model. When you publish a model, a new version of the model is created.

**IMPORTANT** Before you can publish a Python model to SAS Micro Analytic Service, you must make sure that the Python source code is in the correct format. For more information, see “Python Models” on page 80.

---

**Publish Models from the Projects Category**

1. Click to navigate to the Projects category view.
2. Open a project.
3 Select one or more models on the **Models** tab, click **Publish**, and select **Publish**. The Publish Models window appears.

4 (Optional) Update the published name.
   For more information, see Table 7.1 on page 75.

5 (Optional) If you have previously published a model with the same published name, you can replace the model. In the **Items to publish** table, select the check box in the **Replace** column for each model that you want to replace.

6 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. Before you publish a model to the **SAS Micro Analytic Service (maslocal)** destination, you must first update the number of threads for the SAS Micro Analytic Service.
   For more information, see “CAS Authorization: How To (Authorization Window)” in **SAS® Viya Administration: CAS Authorization** and “Configuring Publishing Destinations” in **SAS Open Model Manager: Administrator’s Guide**.

7 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 Open 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.

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

   Note: For models that are within a project version, a publishing validation test is created after a model has been published. For more information, see “Validate Published Models” on page 59.

---

**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 **Projects** 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.
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 Open 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 Open 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 32 and "Import Models" on page 35. 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 Open Model Manager Resources GitHub repository and Model Repository API documentation.
Python Models

Scoring Python Models

After you import a Python model, you can run a scoring test on-demand to score a 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) or Private Docker, a publishing validation test is created. The publishing validation test can then be run to score and validate the model within the publishing destination.

In order to score a model on-demand from within the SAS Open 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 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 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:

```python
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:

```python
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. The Python packages that are used in the Python score code file must be installed in the run-time container. You can install the packages at run time by adding a requirements.json model file that includes the install statements to your model.

Here is an example:
For information about how to format your Python score code file and the requirements.json file, see the examples in the Open Model Manager Resources GitHub repository.

See Also

- “Test Models” on page 55
- SAS Open Model Manager: Quick Start Tutorial
- “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, or a container publishing destination such as Amazon Web Services (AWS) or Private Docker. When you import a Python model, the score code type is set to Python.

See Also

- “About Publishing Models” on page 73
- “Validate Published Models” on page 59

R Models

Scoring R Models

After you import an R model, you can publish the model to a container publishing destination such as Amazon Web Services (AWS) or Private Docker to score and validate it. When you publish a model from a project version, a publishing validation test is created, which you can run to score and validate the model within the publishing destination. Models that are published to a container publishing destination can be scored within run-time containers.

For information about how to format your R score code file, see the examples in the Open Model Manager Resources GitHub repository.

Note: If an R model is transformed to PMML 4.2 format with a score code type of DATA step, you can score the model using a scoring test within the SAS Open Model Manager web application.
Publishing R Models

When you import an R model, the score code type is set to R. R models with a score code type of R can be published only to a container publishing destination such as Amazon Web Services (AWS) or Private Docker. If an R model is transformed to PMML 4.2 format with a score code type of DATA step, it can also be published to a CAS or SAS Micro Analytic Service publishing destination.

See Also

- “About Publishing Models” on page 73
- “Validate Published Models” on page 59

Concepts: Performance Monitoring

Performance Monitoring Reports

About Performance Monitoring Reports

You can monitor the performance of models by analyzing the performance results. You can create a performance definition on the Performance tab of a project. When you run performance for models, the charts that appear on the Performance tab represent the performance results data that is generated.

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

Data Composition Reports

The Variable Distribution chart shows you the distributions for a variable in one or more time periods, which enables you to see the differences and changes over time. The Characteristic and Stability reports 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 Reports

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

- Lift
- Gini
- ROC (Receiver Operating Characteristic)
- KS
- ASE (Average Squared Error) for prediction models
Data Composition Reports

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 a Variable Distribution chart.

![Variable Distribution Chart](image)

Characteristic and Stability

Together, the Characteristic and Stability reports 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 Open 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 Report**

The Characteristic report 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 report 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 report 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:

\[
\text{Deviation Index} = \sum (\% \text{Actual} - \% \text{Expected}) \times \ln \left( \frac{\% \text{Actual}}{\% \text{Expected}} \right)
\]

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 report 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 Report**

The Stability report 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 are examples of the Characteristic and Stability charts.
Model Monitoring Reports

Lift

The Lift report provides a visual summary of the usefulness of the information that is provided by a 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 report 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.

![Lift Chart]

Gini

The Gini and ROC reports show you the predictive accuracy of a 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*AUC-1).

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

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

Here are examples of the monitoring Gini and ROC charts.
The KS charts contain 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 report uses a summary data set that plots the KS statistic and the KS probability cutoff values over time.

Use the KS report 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

The Average Squared Error (ASE) report 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 Average Squared Error 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 Open 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;