



Managing Models and Building Decisions in SAS® Viya®

Course Notes

Managing Models and Building Decisions in SAS® Viya® Course Notes was developed by Peter Christie. Additional contributions were made by Kristen Aponte, Jeanne Ferneyhough, and Ari Zitin. Instructional design, editing, and production support was provided by the Learning Design and Development team.

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

Managing Models and Building Decisions in SAS® Viya® Course Notes

Copyright © 2018 SAS Institute Inc. Cary, NC, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Book code E71275, course code LWVMDM51/VMDM51_001, prepared date 12Jul2018.

LWVMDM51_001

ISBN 978-1-63526-866-9

Table of Contents

To learn more.....	v
Chapter 1 Making Decisions.....	1-1
1.1 Data-Based Decisions	1-3
Chapter 2 SAS® Model Manager.....	2-1
2.1 Introduction.....	2-3
2.2 Project Setup	2-7
Demonstration: Access the Data Files.....	2-10
Demonstration: Create a Model Repository.....	2-16
Demonstration: Create a New Project	2-18
Exercises.....	2-20
2.3 Import Models	2-23
Demonstration: Import Models into a Project.....	2-27
2.4 Model Properties	2-29
Demonstration: Set Model Properties	2-34
2.5 Evaluate Models.....	2-36
Demonstration: Compare Models	2-37
Demonstration: Test a Model	2-38
Demonstration: Set a Champion Model	2-41
Demonstration: Publish a Champion Model	2-43
Exercises.....	2-46
2.6 Creating a Model Performance Report	2-48
Demonstration: Run Performance Reports.....	2-52
2.7 Import, Enable, and Use a Workflow	2-58
Demonstration: Add a Workflow Definition	2-60
Demonstration: Start a Workflow.....	2-62

Demonstration: Complete Workflow Tasks	2-63
2.8 Solutions	2-64
Solutions to Exercises.....	2-64
Solutions to Student Activities (Polls/Quizzes).....	2-71
Chapter 3 Decision Manager.....	3-1
3.1 Introduction.....	3-3
3.2 Accessing Data	3-7
Demonstration: Access the Data Files.....	3-9
3.3 Creating Rule Sets	3-12
Demonstration: Create a Rule Set.....	3-16
Exercises.....	3-21
3.4 Creating and Testing Decisions.....	3-26
Demonstration: Create and Test a Decision	3-28
Exercises.....	3-31
3.5 Lookup Tables (Self-Study).....	3-38
Demonstration: Use a Lookup Table.....	3-43
3.6 Solutions	3-44
Solutions to Exercises.....	3-44

To learn more...



For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at <http://support.sas.com/training/> as well as in the Training Course Catalog.

For a list of SAS books (including e-books) that relate to the topics covered in this course notes, visit <https://www.sas.com/sas/books.html> or call 1-800-727-0025. US customers receive free shipping to US addresses.

Chapter 1 Making Decisions

1.1 Data-Based Decisions.....1-3

1.1 Data-Based Decisions

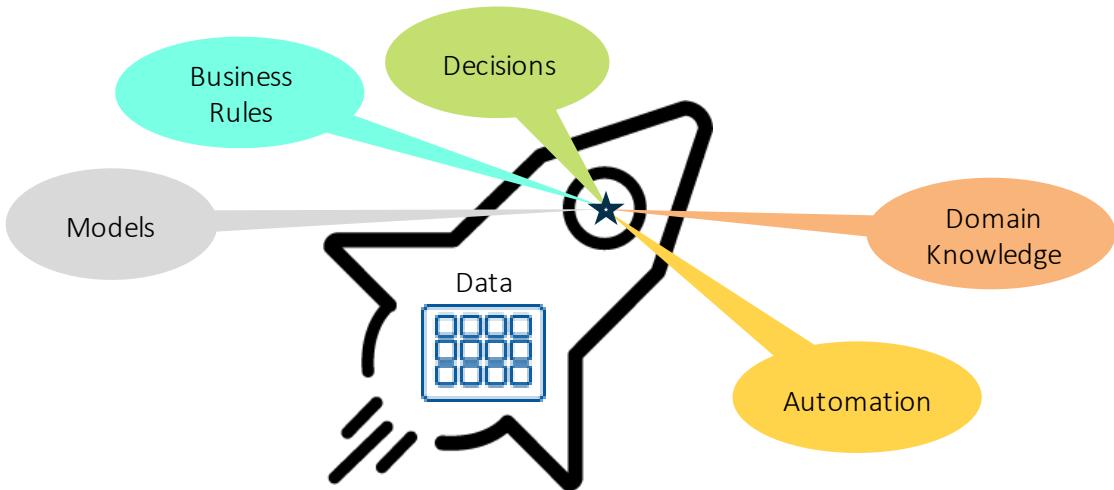
Objectives

- Describe the analytical landscape.
- List the components available to help with business decisions.
- Describe the analytical life cycle.

2



Reaching Your Goal with Advanced Analytics



3



Data (in motion or at rest) is the fuel consumed by advanced analytics to provide insights that help us do what we do better.

To work smarter, you can use various combinations of the following:

- models (predictive and forecasting)
- business rules
- decisions
- domain knowledge
- automation

Discussion

Describe a situation where only a subset of these factors would be used to build a decision.

The diagram shows a central black star with a blue dot in the middle. Six arrows point from surrounding ovals to the star: a cyan arrow from 'Business Rules', a green arrow from 'Decisions', a grey arrow from 'Models', a yellow arrow from 'Automation', an orange arrow from 'Domain Knowledge', and a light blue arrow from 'Data' (represented by a grid icon). The background is white with a faint dotted pattern.

4

Copyright © SAS Institute Inc. All rights reserved.

sas

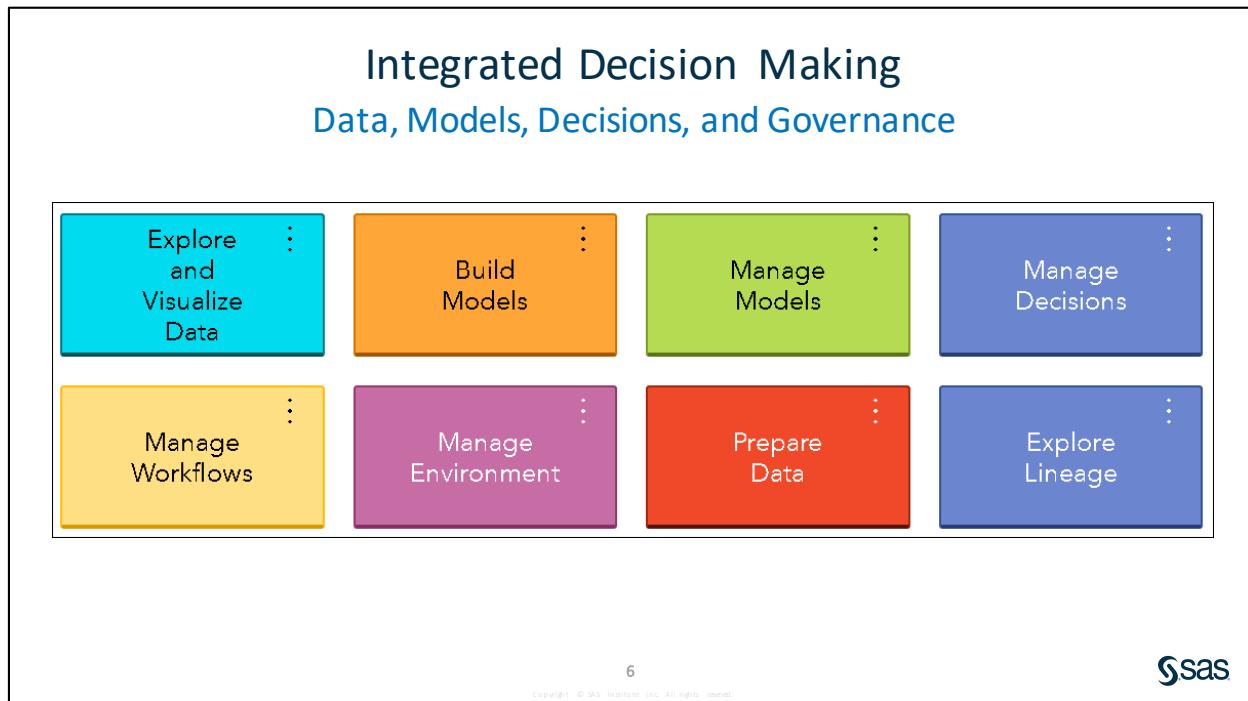
Scope of the Analytical Landscape

Stakeholders	Operations	Targets
<p>Data Scientists</p>	<p>Customer Service</p> <p>Retail</p> <p>Logistics</p> <p>Promotions</p>	<p>Customers</p> <p>Suppliers</p> <p>Partners</p> <p>Stockholders</p>
<p>Management</p> <p>IT / Network</p> <p>Business</p>	<p>Governance</p> <p>Manufacturing</p>	

Copyright © SAS Institute Inc. All rights reserved.

sas

People from across the organization are making choices that impact the connected world: business users, IT operations, manufacturing, and customer service. An enterprise-wide process is necessary to support the different steps in the analytical ecosystem. Collaboration becomes key in successful implementations of analytical-based choices in the production environment.



Using a single environment (such as SAS Viya, pictured above) to manage analytics facilitates the activities from cleansing the data through deploying business decisions. You might use the following:

- models for making predictions and choices
- rules with if-then constructs
- workflows for review and approval cycles
- impact analysis to determine the impact of a change in your decision-making process across the environment

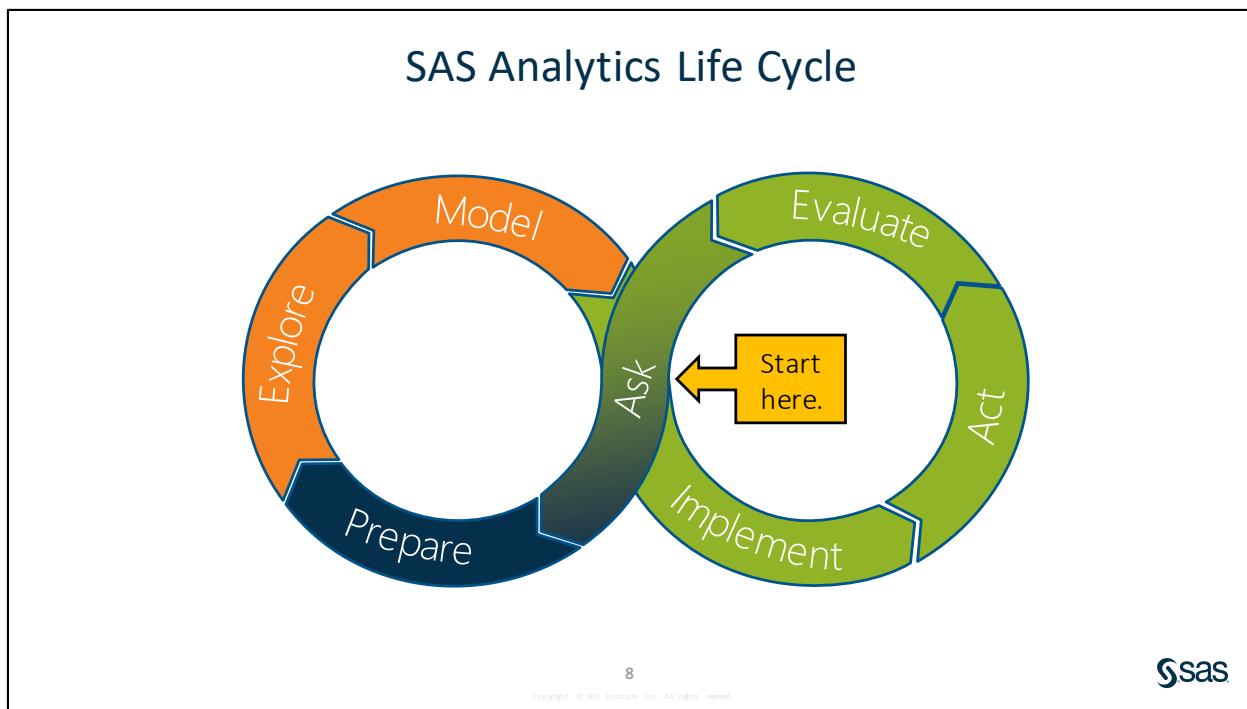
Discussion

What aspects of the integrated decision making slide are you most interested in?

Explore and Visualize Data	Build Models	Manage Models	Manage Decisions
Manage Workflows	Manage Environment	Prepare Data	Explore Lineage

Copyright © SAS Institute Inc. All rights reserved.

sas



1. Ask questions and define the problem.
2. Prepare the data. This phase is both critical to success and frustratingly time-consuming. You have data sitting in databases, on desktops, or in Hadoop. Plus, you want to capture live-streaming data.

3. Explore with self-service visualization tools to serve a wide range of users, from business analysts to data scientists. In this phase, you search for relationships, trends, and patterns to gain a deeper understanding of your data. You also develop and test hypotheses through rapid prototyping in an iterative process.
4. Build predictive models.
5. Implement the model while using repeatable, automated, and governed processes.
6. Act. In this phase, there are two types of decisions: operational decisions that are automated, and strategic decisions where individuals determine a long-term impact.
7. Evaluate. Enter a feedback loop to monitor, evaluate, and possibly retrain the model to always keep your business running at optimal performance.

Model management in the analytics life cycle encompasses implementing, acting, and evaluating results of models. Implementing involves validating models, selecting a champion model, and deploying your models (in batch or real time).

Acting on the results means making decisions based on the model results that impact stakeholders downstream from where the business actions takes place.

Evaluating and monitoring the performance of your models show the effectiveness of your choices. These results let you know when to retire or create new models. The analytics life cycle begins again as additional insights into how to improve are discovered.

Chapter 2 SAS® Model Manager

2.1	Introduction	2-3
2.2	Project Setup.....	2-7
	Demonstration: Access the Data Files	2-10
	Demonstration: Create a Model Repository	2-16
	Demonstration: Create a New Project.....	2-18
	Exercises.....	2-20
2.3	Import Models.....	2-23
	Demonstration: Import Models into a Project.....	2-27
2.4	Model Properties.....	2-29
	Demonstration: Set Model Properties	2-34
2.5	Evaluate Models	2-36
	Demonstration: Compare Models.....	2-37
	Demonstration: Test a Model.....	2-38
	Demonstration: Set a Champion Model.....	2-41
	Demonstration: Publish a Champion Model	2-43
	Exercises.....	2-46
2.6	Creating a Model Performance Report.....	2-48
	Demonstration: Run Performance Reports	2-52
2.7	Import, Enable, and Use a Workflow	2-58
	Demonstration: Add a Workflow Definition.....	2-60
	Demonstration: Start a Workflow.....	2-62
	Demonstration: Complete Workflow Tasks.....	2-63
2.8	Solutions	2-64
	Solutions to Exercises	2-64
	Solutions to Student Activities (Polls/Quizzes)	2-71

2.1 Introduction

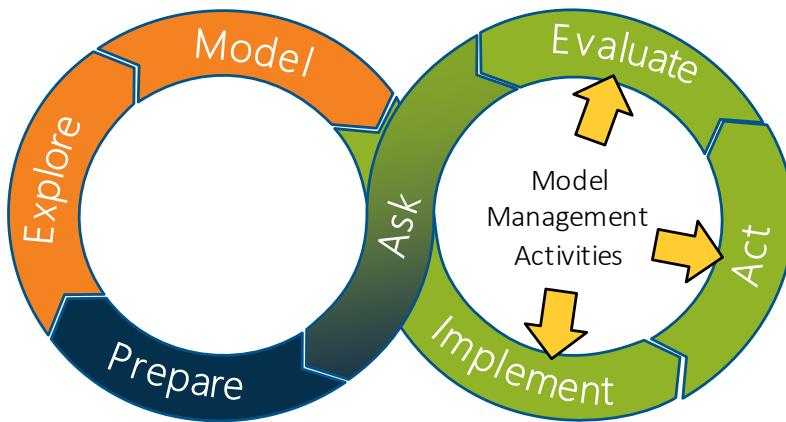
Objectives

- List the services provided by SAS Model Manager.
- Name the three key areas of the Model Manager screen and their purpose.

3



SAS Analytics Life Cycle



4



Model Management focuses on the implement, act, and evaluate elements of the analytics life cycle. For models that you have already created, items such as the model score code, data attributes, documentation, and other model properties are stored in your organization's Model Manager repository. Production score code used for scoring new data with your models resides in Model Manager.

2.01 Multiple Answer Poll

Select your area (or areas) of involvement in model management.

- a. use the model results in a business process
- b. data preparation
- c. manager or approver
- d. system administrator / DBA
- e. run models in production
- f. other

5



Services That Model Manager Provides

- Store models in a common model repository
- Add models from Model Studio, SAS code, PMML, Open Source
- Document a new model based on the model's files
- Run scoring tests* to validate models
- Identify a model as champion or challenger
- Publish models* to SAS Cloud Analytic Services (CAS), Hadoop, SAS Micro Analytic Service, and Teradata
- Use a workflow to track progress through a project

* Certain restrictions apply

6



To be scored by Model Manager, models must meet both of the following requirements:

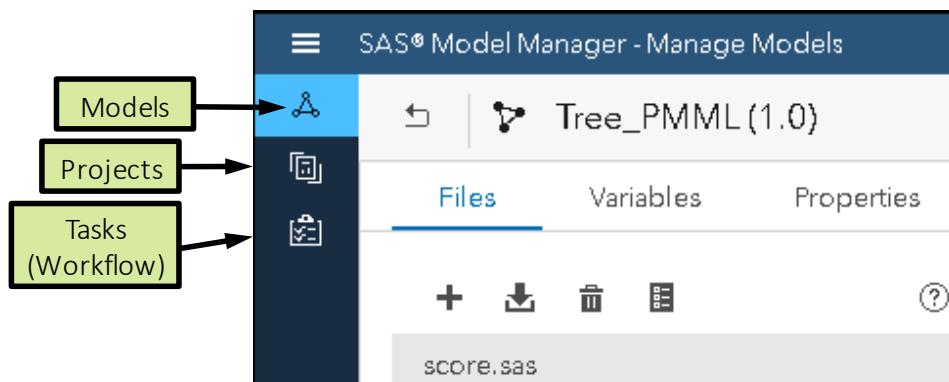
- The score code model file role must be assigned.
- The model must have one of the following score code types: DATA step, SAS program, DS2 package, DS2 embedded process, DS2 multi-step, or Analytic store.

Model publishing restrictions

- Models with a score code type of SAS program cannot be published.
- Models with a score code type of DS2 package cannot be published to SAS Cloud Analytic Services (CAS), Hadoop, or Teradata.
- Models with score code types of Analytic store and DS2 embedded process cannot be published to SAS Micro Analytic Service.

continued...

Model Management Categories



sas

There are three category views in SAS Model Manager. The **models** category (which is selected in the display capture on the slide) shows attributes of an individual model. The **Return** icon returns to the original table of all models in the system regardless of what repository they are in.

The **projects** category includes project details. The projects and models views are often used together when working in Model Manager. Some of the windows (Variables and Properties) have the same name in the projects and models views.

Model Management Categories

The screenshot shows the SAS Model Manager interface with the title "Model Management Categories". On the left, there is a sidebar with several icons. One icon, which looks like a checklist or workflow, is highlighted with a green box and labeled "Tasks (Workflow)". An arrow points from this label to the icon. The main area is titled "Tasks" and shows a table with one row. The table has columns for Name, Workflow, Date Started, Claimed By, and Associated Objects. The single entry is "Import models" under "Workflow", "May 17, 2018 11:16 AM" under "Date Started", and "Project: OS PVA" under "Associated Objects". There are also standard toolbar icons at the top right.

8
Copyright © SAS Institute Inc. All rights reserved.

sas

The **tasks** category shows workflow information for a user. The workflow contains the checklist of steps to guide you through your specific model management analytical life cycle. In the window on the slide above, the user is reminded to import models for a project that is following the steps outlined in the Model Lifecycle workflow. Clicking the **Associated Objects** link on the right takes the user to the project. Using a workflow is optional. Refer to the section on workflows for more information.

2.2 Project Setup

Objectives

- Sign in to SAS Model Manager.
- Add data sources and access data.
- Create a model repository.
- Create a new project.

10



Data Sources

Data must be in SAS Cloud Analytic Services (CAS) for Model Manager to use it. This means that the data must exist in computer memory if you want to run model scoring.

If the data is not currently in memory:

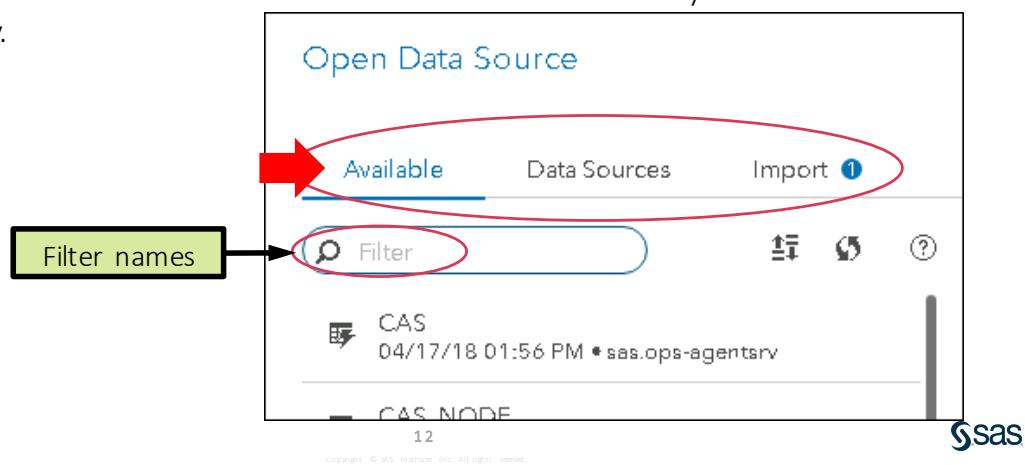
- You can copy a recognized data source into CAS memory.
- You can import new data from a computer that is accessible from the CAS environment.

11



Data Sources

This window appears in many parts of the SAS Viya environment (that is, anytime that you have to specify a data source that will be used for a specific task). Data sources under the Available window are already loaded into memory.



From the Available window, you can filter on the file names to quickly see whether a data source is loaded into memory and is ready for use by SAS Model Manager. The lightning bolt icon indicates that the data is currently loaded into memory.

Data Sources

The screenshot shows the SAS Data Explorer interface with the 'Open Data Source' window open. The 'Data Sources' tab is selected, indicated by a red circle. Below it, the 'Available' tab is shown with a dropdown menu set to 'Public'. Under 'Available', there is a list of datasets: HMEQ_PERF_Q1.sashdat, HMEQ_PERF_Q2.sashdat, HMEQ_PERF_Q3.sashdat, HMEQ_PERF_Q4.sashdat, HMEQ_TEST.sashdat (which is highlighted with a red circle), and HMEQ_TRAIN.sashdat. To the right, a detailed view of the HMEQ_TEST.sashdat table is displayed. The table has six columns: BAD, LOAN, MORTDUE, VALUE, REASON, and JOB. The 'Profile' section shows 'Last profiled: Never', 'Columns: ..', 'Rows: ..', 'Size: ..', 'Label: Not available', and 'Location: cas-shared-default/Public'. At the bottom right of this panel, there is a lightning bolt icon with a hand cursor over it, also highlighted with a red circle. The SAS logo is visible in the bottom right corner of the window.

This window shows the **HMEQ_TEST** table under Data Sources, which is going to be used in a scoring test for a project. If this table did not show up under Available, then it needs to be loaded into CAS for use in Model Manager. Clicking the lightning bolt icon in the top right loads the selected table into memory.

For more information about data sources, refer to the SAS Data Explorer documentation:
<http://documentation.sas.com/?cdclId=vacdc&cdcVersion=8.2&docsetId=datahub&docsetTarget=n01gdrmxI7lhszn1gny5h764yisv.htm&locale=en>



Access the Data Files

Note: The instructor will guide you through the steps necessary to make the data files available on your computer. The next three steps have already been completed on the classroom computers.

If the files do not already exist in D:\Workshop\Winsas\Viya_DM\QuickStartTutorial on the classroom computer, download QuickStartTutorial.zip onto your computer from <http://support.sas.com/documentation/onlinedoc/modelmgr/>.

1. Create a folder on your machine to store the tutorial files: **D:\Workshop\Winsas\Viya_DM**.
2. Save the QuickStartTutorial.zip file into the folder that you created.
3. In Windows Explorer, right-click **QuickStartTutorial.zip** and select **WinZip** \Rightarrow **Extract to here**.

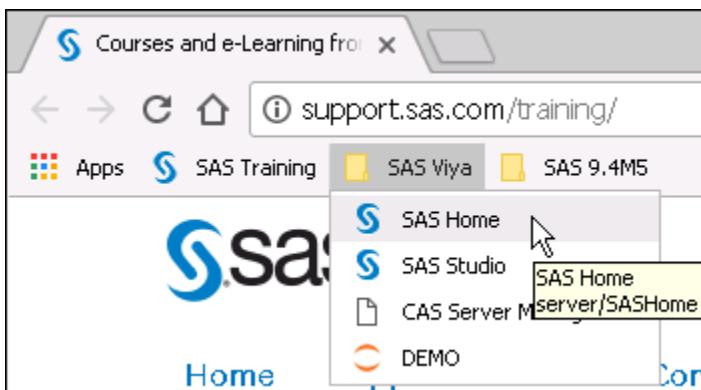
Windows creates a folder named **QuickStartTutorial** that contains the tutorial data sets.

Sign In to SAS Model Manager

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

To sign in to SAS Model Manager:

1. From the Google Chrome web browser, select **SAS Viya** \Rightarrow **SAS Home** as shown below.



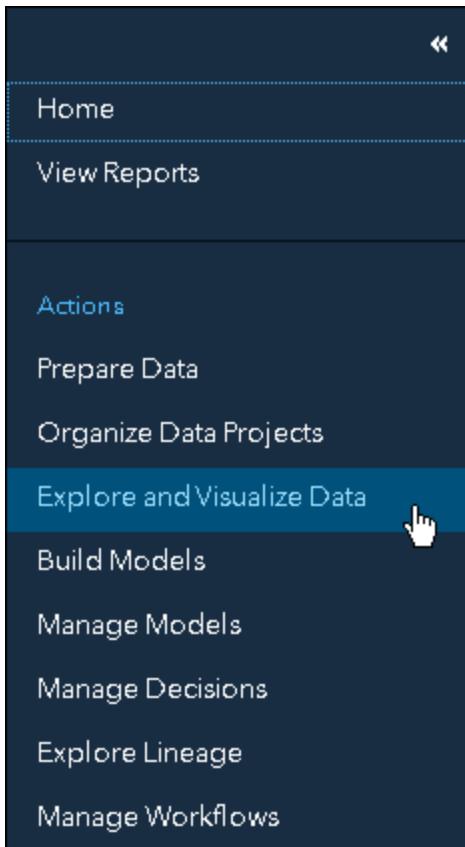
The Sign In page appears.

Note: Contact your system administrator if you need the URL for SAS Model Manager. The default URL is <http://server/SASHome/SASModelManager>.

2. Enter the user ID **Student** and the password **Metadata0**.
3. Click **Sign In**.
4. If prompted, select **Yes** to the Assumable Groups prompt.

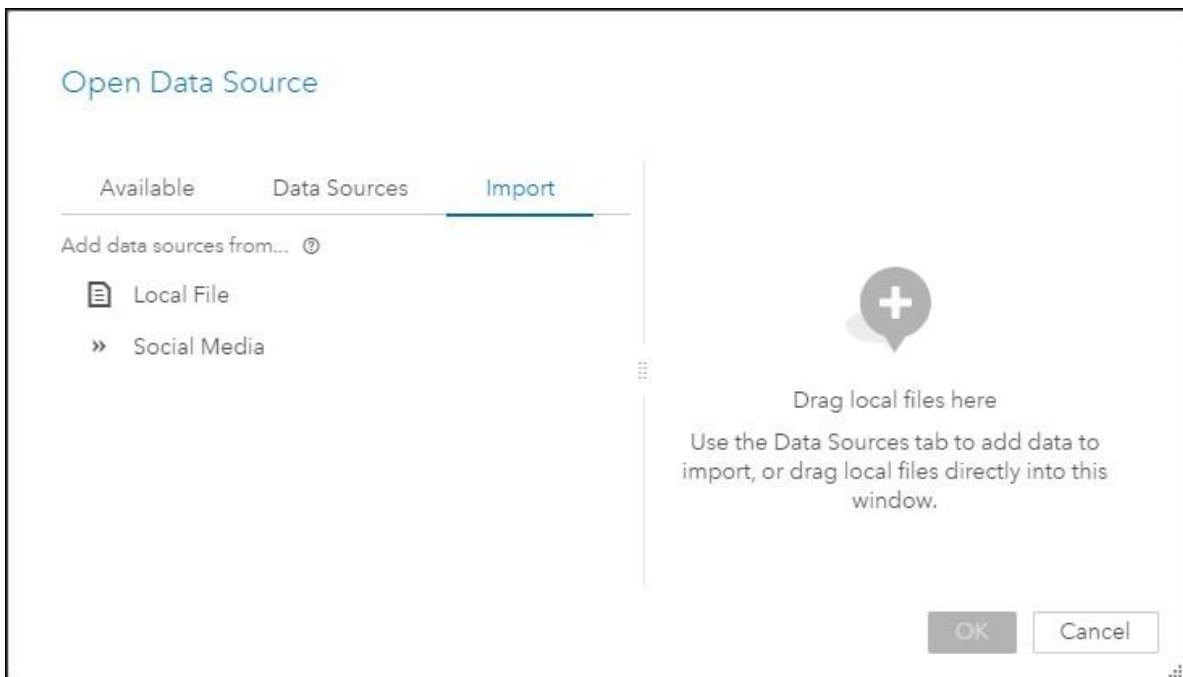
Import Data Sources

1. Click  and select **Explore and Visualize Data**.



2. Click **Data**. The Open Data Source window appears.

3. Click **Import** ⇒ **Local File** and navigate to **D:\Workshop\Winsas\Viya_DM\QuickStartTutorial\Data**.



4. Select the **hmeq_test** SAS data set and click **OPEN**.



5. Click **Import Item**. Wait until the import completes and then click **OK** to close the Open Data Source window.

By default, the table is imported into the target destination library, **cas-shared-default/Public**.

6. (Optional Information) Your instructor might direct you to load the data into memory if it was already imported. In this case, the table will already exist in data tables. Select the table and click the lightning bolt to load it into memory.

The screenshot shows the SAS Model Manager interface. On the left, there's a navigation bar with 'Available', 'Data Sources' (which is highlighted with a red oval), and 'Import'. Below this is a dropdown menu set to 'Public'. The main area lists several data sources:

- HMEQ_PERF_Q1.sashdat (02/16/18 08:12 PM)
- HMEQ_PERF_Q2.sashdat (02/16/18 08:12 PM)
- HMEQ_PERF_Q3.sashdat (02/16/18 08:12 PM)
- HMEQ_PERF_Q4.sashdat (02/16/18 08:12 PM)
- HMEQ_TEST.sashdat (02/13/18 04:52 PM)** (highlighted with a red oval)
- HMEQ_TRAIN.sashdat

To the right of the list is a detailed view for 'HMEQ_TEST.sashdat'. It includes tabs for 'Details', 'Sample Data', and 'Profile'. The 'Profile' tab is active. It displays the following information:

- Last profiled: Never
- Columns: 2
- Rows: 2
- Size: 2
- Label: Not available
- Location: cas-shared-default/Public
- Date created: 02/13/2018 04:52 PM

A red oval also highlights the lightning bolt icon in the 'Profile' section, which is used to load the data into memory.

7. Click and select **Manage Models** to return to SAS Model Manager.

See Also

- [Getting Started with SAS® Data Preparation for SAS® Viya®](#)
- [“Data Administration: How to \(SAS Environment Manager\)” in SAS® Viya® Administration: Data](#)

End of Demonstration

continued...

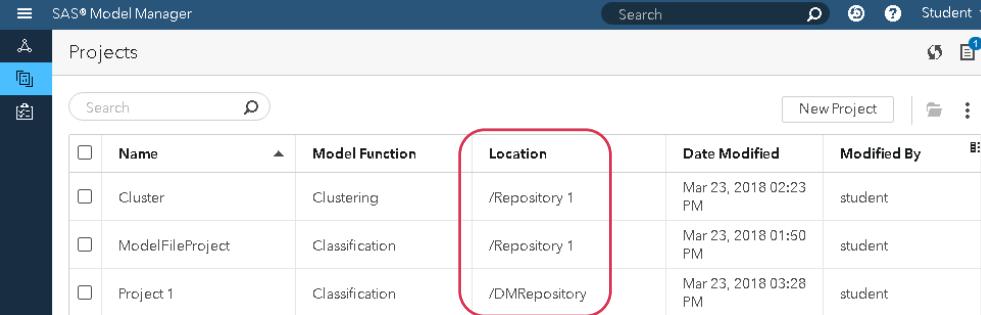
Working with Model Repositories and Folders

Projects and model content are stored in a model repository.

The default repository is named **Repository 1**.

Pipelines created in Model Studio are stored in **DMRepository**.

(These two standard repositories are included with the software.)



Name	Model Function	Location	Date Modified	Modified By
Cluster	Clustering	/Repository 1	Mar 23, 2018 02:23 PM	student
ModelFileProject	Classification	/Repository 1	Mar 23, 2018 01:50 PM	student
Project 1	Classification	/DMRepository	Mar 23, 2018 03:28 PM	student



Many projects usually reside in a repository. This course shows how to create a repository even though this activity is not performed very often. The DMRepository should be reserved for use by pipelines created in Model Studio.

Working with Model Repositories and Folders

You can add repositories for different organizations, or for test and production environments.

In this course, you create a repository named **QS_Repository**.

You can create folders in a repository to organize modeling projects.

New Repository

New Name: *

Description:

Save Cancel

18

Copyright © SAS Institute Inc. All rights reserved.



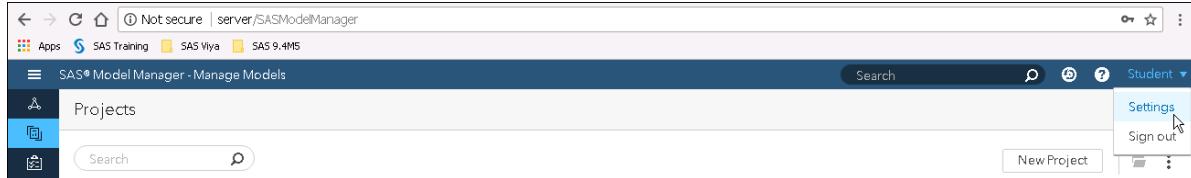
Users with proper authorization can rename and delete repositories. A model repository usually contains many Model Manager projects. After you save the new repository, you cannot edit the description.

Authenticated users can delete only repositories that they have created and only if the repositories contain no projects.



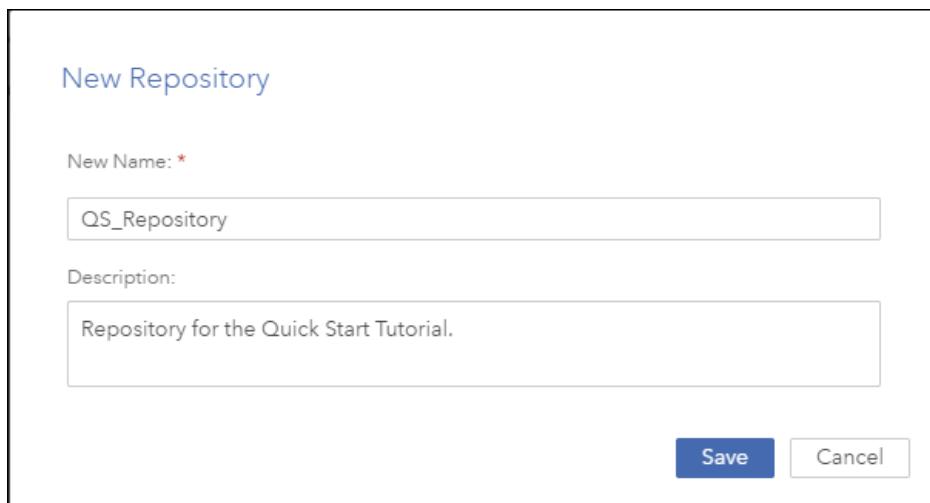
Create a Model Repository

1. Click the arrow next to **Student** in the application bar (top right of window) and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repository**.



2. Click .
3. Enter **QS_Repository** as the name of the repository.
4. Enter **Repository for the Quick Start Tutorial** for the description of the repository.

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



New Repository

New Name: *

Description:

Save **Cancel**

5. Click **Save**.

See Also

- “Managing Model Repositories” in *SAS® Model Manager: User’s Guide*

End of Demonstration

Projects

A project stores model score code, variables, documentation, and scoring test results within a project version.

<input type="checkbox"/>	Name	Role	Model Function	Project Version	Algorithm	Date Modified
<input type="checkbox"/>	QS_Reg1		Classification	Version 1 (1.0)	Logistic regression	Feb 13, 2018 12:00:00
<input type="checkbox"/>	QS_Tree1		Classification	Version 1 (1.0)	Decision tree	Feb 13, 2018 02:00:00

20



A project version represents a time-based container of models. If you have a policy that requires models be replaced each year, then the version would represent a year.

Project Properties

General and user-defined project properties are recorded under properties.

21



User-defined properties can be used as search arguments for locating projects in the repository.

Note: The UUID of the project is recorded in the project properties. This UUID is needed when creating performance reports later.



Create a New Project

In this demo, you create a folder named **sasdemo** in QS_Repository to store your projects.

1. Click to navigate to the projects category view.
2. Click **New Project**. The New Project window appears.
3. Enter **QS_HMEQ** for the name of the project.
The initial version is displayed and reflects the level for sequential versions.
4. (Optional) Enter a description for the project.
5. Select **Classification** for the model function.
6. Click the **Location folder** icon and click the **QS_Repository** repository folder.
7. Click the **New folder** icon and enter **Sasdemo** to create a new folder in the repository. Click anywhere in the white space of the window and then reselect the **Sasdemo** folder and click **OK**.

The screenshot shows the 'New Project' dialog box. It has the following fields:

- Name:** *
- Description:**
- Initial version:**
- Model function:**
- Location:**
- Buttons:**

8. Click **Save**. The project opens.

See Also

- “Working with Projects” in *SAS® Model Manager: User’s Guide*

End of Demonstration



Exercises

1. Accessing Data and Creating a Model Repository and a New Project

a. Access the data files.

Follow the steps in the **Import Data Sources** section in the “Accessing the Data Files” demonstration, which accesses the course data.

b. Create a model repository.

- 1) Click your name in the application bar and select **Settings** ⇒ **SAS Model Manager** ⇒ **Repository**.
- 2) Click
- 3) Enter **Training_Repository** as the name of the repository.
- 4) Enter **Repository for student exercises** for the description of the repository.

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

New Repository

New Name: *

Description:

- 5) Click **Save** ⇒ **Close**.

c. Create a new project.

- 1) Click to navigate to the Projects category view.
- 2) Click **New Project**. The New Project window appears.
- 3) Enter **QS_PVA** for the name of the project.

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

- 4) (Optional) Enter a description for the project.
- 5) Select **Classification** for the model function.
- 6) For the location, select the **Training_Repository** and click to create a new folder and enter **Exercises** as the folder name. Click the white space in the window, and then click back on the Exercises folder you just created to make sure it is selected. Click **OK**.

- 7) Click **Save**. The project opens.

New Project

Name: *
QS_PVA

Description:

Initial version:
1.0

Model function: ②
Classification

Location:
/Model Repositories/Training_Repository/Exercises

Save Cancel

2. Importing a Data Source and Profiling the Data

- a. Go to **Explore and Visualize data** from the applications menu .

Note: If you performed the steps in the previous demo, click the down arrow next to the displayed data source to add another data source.

Click **Data** \Rightarrow **Open Data Source** \Rightarrow **Import**, select **Local File**, and load the **pva_train** local data source from **D:\Workshop\WinsasViya_DM\QuickStartTutorial\Data**.

- b. Select **Import item** and click **OK**.

- c. Click **Available** and scroll down to **PVA_TRAIN**.

Notice that the icon has a lightning bolt, indicating that the data source is loaded into memory.

- d. Select **Sample Data** to view data in the **pva_train** table. Do any of the variables have missing data?

- e. From Data Sources, run a profile against the **pva_train** data source. This provides summary statistics on the contents of the file.

- f. What is the mean of **DONOR_AGE**? What is the maximum value of **RECENT_AVG_GIFT_AMT**?

- g. Click **OK**
- h. Click  and select **Manage Models** to return to SAS Model Manager.

End of Exercises

2.3 Import Models

Objectives

- State multiple ways of attaching models into Model Manager.
- Import models into a project.

25



Working with Models

The models category view lists all models and specifies the repository, project, and version for each.

To manage models, you use the projects category view and the models category view.

	Name	Model Function	Location	Project (Version)
<input type="checkbox"/>	Forward Logistic Regression (Pipeline 1)	Classification	/DMRepository	hmmv_bu, Version 1 (1.0)
<input type="checkbox"/>	PVA_Logistic	Classification	/Training_Repository/Exercises	QS_PVA, Version 1 (1.0)

Annotations in the screenshot:

- 'Models category' points to the sidebar button.
- 'Projects category' points to the sidebar button.
- 'Add a model' points to the 'Add a model' button.
- 'New Model' points to the 'New Model' button.
- 'Import' points to the 'Import' button.
- 'Repository' points to the 'Repository' column header in the table.
- 'Version' points to the 'Project (Version)' column header in the table.
- 'Sas' is the SAS logo in the bottom right corner.

Models can be added by selecting individual model files using the **New Model** button. They can be imported if they are in the form of a PMML XML, SAS Package, or ZIP format model by using the **Import** button. A model imported from the models category can be added later to an appropriate project with the projects category.

Note: Some table columns have been excluded for educational purposes.

Working with Models

Models can be added either by selecting individual model files using the New Model button or by using the Import button if the model is in PMML, SAS Package, or ZIP format.

A model imported from the models category can be added later to an appropriate project with the projects category.

The screenshot shows two dialog boxes side-by-side. On the left is the 'New Model' dialog, which has fields for 'Name:' (containing 'Model 4') and 'Location:' (containing '/Model Repositories/Repository 1'). There is also a 'Select one or more files:' field with a file browser icon. At the bottom are 'Save' and 'Cancel' buttons. On the right is the 'Import Models' dialog, which has a 'Select a target location for the models:' field (containing '/Model Repositories/Repository 1'), a 'Select one file at a time to import the model contents:' field with a file browser icon, and a '+ Add model' button. At the bottom are 'Import' and 'Cancel' buttons. A small SAS logo is in the bottom right corner.

You can import models from Model Studio, PMML (Predictive Modeling Markup Language), SPK (SAS package file), or a ZIP archive file that contains model files. Multiple models can be imported at the same time into the models category view. The /DMRepository should be reserved for projects and models that are registered from Model Studio.

Working with Models

If you import a properly formed PMML version 4.2 model file, the following additional files are also created:

- score.sas
- input.xml
- output.xml
- target.xml

The score.sas file is assigned the role of **Score code**.

The screenshot shows the 'SAS® Model Manager - Manage Models' interface. On the left is a sidebar with icons for 'New Model', 'Import Model', and 'Edit Model'. The main area shows a list of files for a model named 'Neural_PVA_pmml (1.0)'. The 'Files' tab is selected. The list includes 'score.sas' (highlighted in blue), 'Neural_PVA_pmml.xml', 'input.xml', 'output.xml', and 'target.xml'. At the bottom are buttons for '+', 'Upload', 'Delete', and 'Properties'.

This PMML model can be scored in its project under the Test window because the additional files are automatically created. 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.

Working with Models

Models can be added directly from the projects category using the selections under the icon at the right of the screen if using a small browser window.

The screenshot shows the SAS Model Manager interface. The title bar says "SAS Model Manager". The main area has tabs for "Models", "Variables", "Properties", "Tests", "Performance", and "Workflow". A red arrow points to the left edge of the window, where the vertical scroll bar is located. In the top right corner of the main content area, there is a three-dot menu icon. A context menu is open at this location, with "Import" highlighted. Other options visible in the menu include "New Model", "Import >", "Compare", and "Actions >". The "Models" tab is selected, showing a table with columns: Role, Mod..., Proj..., AI..., Date Modified. One row is visible: Role is "Classification", Mod... is "P...", Proj... is "Version 1 (1.0)", AI... is "Logistic regression", and Date Modified is "Apr PM". The "Version" dropdown shows "Version 1 (1.0)". The bottom right of the window has the SAS logo.

Note: The Import and Copy from selections appear under the icon in this window because the browser has been resized to be small and prevents the entire window from appearing. Notice the > to the right of the Workflow menu option. If the browser was resized to be large enough to show the entire application, the New Model and Import buttons would appear as shown in the next screen.

Working with Models

This is the projects category as it appears in a large browser window. Notice that the New Model and Import buttons are now displayed, whereas they were not rendered in the previously shown small browser view.

Name	Role	Model Function	Project...	Algorithm	Date Modified	Modifi...
Neural_PVA_pmml		Classification	Version 1 (1.0)	Nonlinear Optimization	May 18, 2018 11:39 AM	student
PVA_LogisticRegres...	★	Classification	Version 1 (1.0)	Logistic regression	Apr 19, 2018 06:00 PM	student

30

Working with Models

You can use Model Studio pipelines to register models that were created there. The Manage Model option creates the Model Manager project.

31

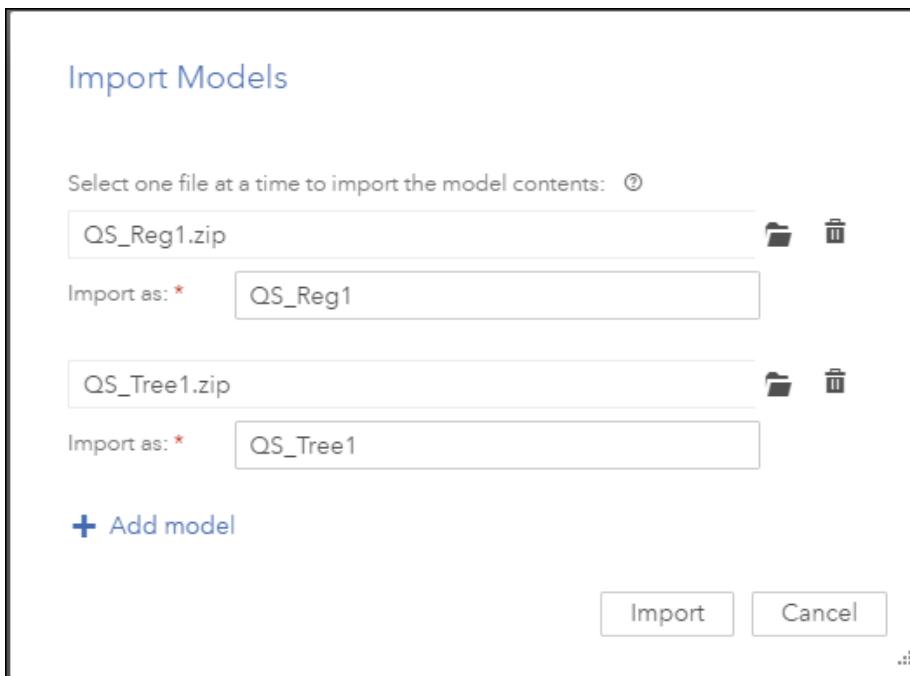
Model Studio pipelines are used to build and compare models. You can register your champion model and then select **Manage Model** to create a complete model manager project.



Import Models into a Project

Note: By default, the models are imported into the latest project version (for example, **Version 1 (1.0)**).

1. Open the **QS_HMEQ** Model Manager project.
2. On the Models tab, click **Import** and select **Import** from the drop-down list. The Import Models window appears.
3. Select the models to import.
 - a. Click and navigate to the location of the model files:
D:\Workshop\Winsas\Viya_DM\QuickStartTutorial\Models
 - b. Select the **QS_Reg1.zip** file that contains your model contents.
- Note:** Select only one ZIP file at a time in the Open window. The name of the selected file is used as the default model name.
4. Click **Open**.
5. Click **Add model** to add rows so that you can import more models.
6. Repeat steps 3 and 4 to import the model from the **QS_Tree1.zip** file.



7. Click **Import**.

TIP To remove extra lines that do not contain a file to import, click before you click **Import**.

8. Click **Close** in the confirmation message.

Here is an example of the Models tab after the models have been imported.

The screenshot shows the SAS Model Manager interface with the title bar "SAS® Model Manager - Manage Models". The left sidebar has icons for Home, Projects, and Data. The main area is titled "QS_HMEQ". The top navigation bar includes "Search", "Student", and other icons. Below the title is a breadcrumb trail "QS_HMEQ". The main content area has tabs: "Models" (selected), "Variables", "Properties", "Tests", "Performance", "Workflow", and "History". A search bar says "Filter" and a dropdown says "Version: Version 1(1.0)". Buttons include "New Model", "Import", "Compare", and a three-dot menu. A table lists two models:

<input type="checkbox"/>	Name	Role	Model Function	Project Version	Algorithm	Date Modified	Modified By
<input type="checkbox"/>	QS_Reg1		Classification	Version 1(1.0)	Logistic regression	Apr 19, 2018 04:07 PM	student
<input type="checkbox"/>	QS_Tree1		Classification	Version 1(1.0)	Decision tree	Apr 19, 2018 04:07 PM	student

See Also

- “Import Models” in *SAS® Model Manager: User’s Guide*

End of Demonstration

2.4 Model Properties

Objectives

- Update properties of models that have been imported into Model Manager.

34



continued...

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 screenshot shows the SAS Model Manager application window. The title bar says "SAS Model Manager". The main area has tabs: Models, Variables, Properties (which is highlighted with a red circle), Tests, Performance, Workflow, and History. Below the tabs, there's a "General" section with a "User-Defined" header. The "Properties" tab contains the following data:

Name:	QS_Complete
Description:	(empty)
Model function:	Classification
Operation status:	Under development
Created by:	student
Date created:	Feb 13, 2018 11:48 AM
Date modified:	Feb 13, 2018 11:48 AM
Location:	/Model Repositories/Repository 1
Champion version:	Version 1
Champion model name:	QS_Tree1
Default train table:	cas-shared-def... <input type="button" value="Edit"/> <input type="button" value="Clear"/>

35



Model Properties

The Score code type property and details are shown below. The general model properties are included in the course notes for your reference.

Score code type

Models with a Score code type of DATA step, SAS program, DS2 package, DS2 embedded process, DS2 multi-type, and Analytic store can be scored.
 Models with a Score code type of SAS program cannot be published.
 Models with a Score code type of DS2 package cannot be published to CAS, Hadoop, or Teradata.
 Models with a Score code type of Analytic store and DS2 embedded process cannot be published to SAS Micro Analytic Service.

List of General Properties

Property	Description
Name	Specifies the name of the model. It can be renamed from the models category view only or on the Models tab of a project.
Description	Specifies the description of the model.
Location	Specifies the location of the model in the common model repository.
Project name	Specifies the name of the project that contains the model.
Project version	Specifies the project version that contains the model.
Function	Specifies the type of output that your model generates. For more information, see “Types of Model Functions.”

List of General Properties	
Property	Description
Score code type	<p>Specifies the type of score code that your model uses. A value must be specified in order for you to be able to publish a model, run a test, or monitor performance for a model. You can select a value from the list or enter your own value. User-defined values are not added to the list. Instead, they are stored within the model properties. When you are specifying a score code type, here are some restrictions to be aware of:</p> <p>Only models with a score code type of DATA step, SAS program, DS2 package, DS2 embedded process, DS2 multi-type, and Analytic store can be scored.</p> <p>Models with a score code type of SAS program cannot be published.</p> <p>Models with a score code type of DS2 package cannot be published to CAS, Hadoop, or Teradata.</p> <p>Models with score code types of Analytic store and DS2 embedded process cannot be published to SAS Micro Analytic Service.</p>
Train table	Specifies the Train table that is used to validate scoring functions or scoring model files when a user publishes the associated project champion model or challenger models to a database. This property is optional.
Train code type	Specifies the type of train code that your model uses. This property is for informational purposes only. You can select a value from the list or enter your own value. User-defined values are not added to the list. Instead, they are stored within the model properties.
Algorithm	Specifies the computational algorithm that is used for the selected model.
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.

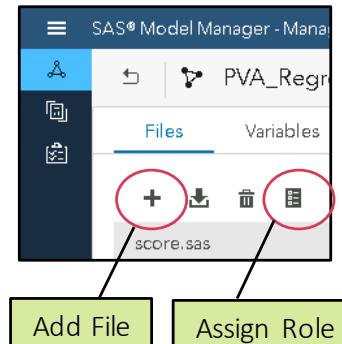
List of General Properties	
Property	Description
Output prediction variable	The output prediction variable name, when the Model function property is set to Prediction, Analytical, Forecasting, or Transformation.
Output segmentation variable	The output segmentation variable name, when the Model function 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.

Assigning Model File Roles

You might need to assign one of the following roles to your model files:

- Score code
- Score resource
- Format XML
- Format item store
- Format catalog
- Train code
- Train resource
- Input variables**

- Output variables
- Target variables
- Analytic store
- Python pickle
- PMML
- Documentation
- Data plan



sas

These are the model file roles that you can specify as properties for any file included with a model. You can attach a PDF document to a model and assign it the role of Documentation. You can attach files to a model without specifying a role. Some roles, such as Score code, might be assigned automatically when importing a model.



Set Model Properties

To set the model properties:

1. Click the name of the model **QS_Reg1**. The model object opens.
2. The Files window shows various JSON files describing input and output variables, and it also contains the **dmcas_scorecode.sas** program.

The screenshot shows the SAS Model Manager interface with the title bar "SAS® Model Manager - Manage Models". Below the title bar, the model name "QS_Reg1 (1.0)" is displayed. The interface has four tabs: "Files" (selected), "Variables", "Properties", and "Versions". In the "Files" tab, there is a list of files: "ModelProperties.json", "dmcas_fitstat.json", "dmcas_lift.json", "dmcas_roc.json", "dmcas_scorecode.sas" (which is highlighted in blue), and "dmcas_scoreinputs.json". To the right of the file list, the content of the "dmcas_scorecode.sas" file is shown in a code editor. The code consists of several lines starting with asterisks (*), which are comments indicating system information such as DMCAS Release, SAS Release, Site Number, Host, Encoding, Java Encoding, Locale, and Project GLUTD.

```

*-----*
* DMCAS Release
* SAS Release:
* Site Number:
* Host:
* Encoding:
* Java Encoding
* Locale:
* Project GLUTD.

```

3. Click the **Variables** tab. For this model, the input and output variables were recognized from the ZIP file for this model and are populated for us.
4. Click the **Properties** tab.
5. On the General page, update the values so that the following properties are set:

Property	Value
Score code type	DATA step
Target variable	BAD
Target event value	1
Target level	Binary

Property	Value
Output event probability variable	EM_EVENTPROBABILITY

The Output event probability variable is used for classification models scoring tests.

1. Click .
2. To return to the main project, click .
3. Repeat steps 1 through 7 for the **QS_Tree1** model.

See Also

- “Modifying Model Properties” in *SAS® Model Manager: User’s Guide*

End of Demonstration

2.5 Evaluate Models

Objectives

- Run scoring tests of models.
- Define a model as the champion model.
- Publish a model for use by other applications.

40



Evaluating Models

One reason for completing a modeling project is to identify a champion model that an external scoring application uses to predict an outcome.

You can compare and assess models, run a test of a model's scoring code, and monitor performance of a model.

Models can be published to CAS, Hadoop, SAS Micro Analytic Service, and Teradata for production scoring applications.

41





Compare Models

1. On the Models tab of the QS_HMEQ project, select the **QS_Reg1** and **QS_Tree1** models.
2. Click **Compare**. The Compare page appears.
3. Click **Show Differences**. The default is to show all of the comparison model content.
4. Review the differences for the following model information:
 - Model properties
 - User-defined properties
 - Input variables
 - Output variables
 - Target variable
 - Fit statistics
 - Lift and ROC plots
5. Click **Close**.

See Also

- “Compare Models” in *SAS® Model Manager: User’s Guide*

End of Demonstration



Test a Model

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

1. On the Tests tab of the QS_HMEQ project, click **New Test**. The New Test window appears.
2. Enter **QS_Test1** for the name of the test.
3. (Optional) Enter a description for the test (for example, **Test to determine the project champion model.**).
4. Click **Choose Model** and select the **QS_Tree1** model to test.
5. Under Data source, click , select **HMEQ_TEST** from the PUBLIC caslib as the input table for the test, and click **OK**. (If you do not see the data set, try clicking the refresh button .)
6. Map the variables.

Note: In this case, all variable names in the model matched names in the data source, so mapping is not required because it completed automatically. Continue with step 7.

Note: SAS Model Manager automatically maps model input variables to the columns in the input table when the names and data types of the variables match those of the table columns. If any input variables cannot be mapped automatically, a warning message is displayed.

Data source: *

HMEQ_TEST

Variables

⚠ Input variables must be mapped to table columns. **X**

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) Expand the **Advanced** section to display the advanced options.

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

New Test

Name: *

Description:

Test to determine the project champion model.

Model: *

Choose Model

Data source: *

Variables

▼ Advanced

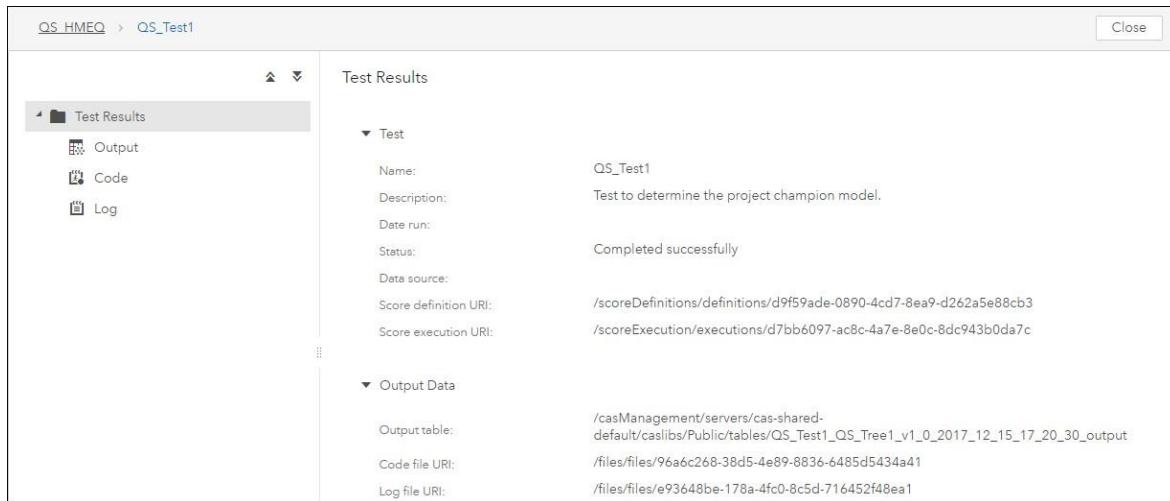
Output data library: *

Save Run Cancel

9. Click **Run** to save and run the test.

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

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



The screenshot shows the 'Test Results' window for a test named 'QS_Test1'. The left sidebar has a tree view with 'Test Results' expanded, showing 'Output', 'Code', and 'Log' options. The main area is titled 'Test Results' and contains sections for 'Test' and 'Output Data'. The 'Test' section shows the following details:

Detail	Value
Name:	QS_Test1
Description:	Test to determine the project champion model.
Date run:	
Status:	Completed successfully
Data source:	
Score definition URI:	/scoreDefinitions/definitions/d9f59ade-0890-4cd7-8ea9-d262a5e88cb3
Score execution URI:	/scoreExecution/executions/d7bb6097-ac8c-4a7e-8e0c-8dc943b0da7c

The 'Output Data' section shows the following URIs:

URI Type	URI Value
Output table:	/casManagement/servers/cas-shared-default/caslibs/Public/tables/QS_Test1_QS_Tree1_v1_0_2017_12_15_17_20_30_output
Code file URI:	/files/files/96a6c266-38d5-4e89-8836-6485df434a41
Log file URI:	/files/files/e93648be-178a-4fc0-8c5d-716452f48ea1

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

11. Click **Close** to return to the Tests window.

See Also

- “Test Models” in *SAS® Model Manager: User’s Guide*

End of Demonstration



Set a Champion Model

1. Click the **Models** tab of the QS_HMEQ project.
2. Select the **QS_Tree1** model, click at the top right of the window, and select **Set as champion**. The Select Project Output Variables window appears.
3. Select **EM_EVENTPROBABILITY** from the **Model Output Variables** column and then change the name of the project output variable to **score** in the **Project Output Variables** column.

Select Project Output Variables

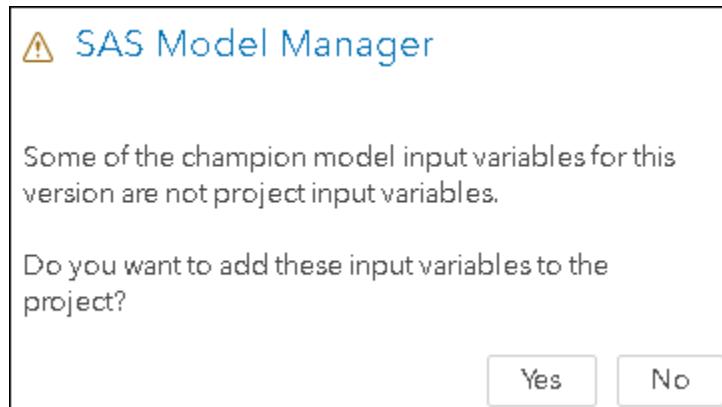
Select the model output variables to use as project level output variables. The project output variables are used to generate output results when running performance monitoring reports.

<input type="checkbox"/>	Model Output Variables	Project Output Variables
<input type="checkbox"/>	EM_CLASSIFICATION	EM_CLASSIFICATION
<input checked="" type="checkbox"/>	EM_EVENTPROBABILITY	score
<input type="checkbox"/>	EM_PROBABILITY	EM_PROBABILITY
<input type="checkbox"/>	I_BAD	I_BAD
<input type="checkbox"/>	P_BAD0	P_BAD0
<input type="checkbox"/>	P_BAD1	P_BAD1

Save **Cancel**

4. Click **Save**.

5. 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.

The screenshot shows the SAS Model Manager interface with the "Models" tab selected. The window title is "QS_HMEQ". The top menu bar includes "File", "Edit", "View", "Model", "Variables", "Properties", "Tests", "Performance", "Workflow", and "History". Below the menu is a toolbar with icons for "New Model", "Import", "Compare", and "Close". A filter bar allows filtering by "Name", "Role", "Model Function", "Project Version", "Algorithm", "Date Modified", and "Modified By". The main area displays a table of models:

Name	Role	Model Function	Project Version	Algorithm	Date Modified	Modified By
QS_Reg1		Classification	Version 1 (1.0)	Logistic regression	Dec 14, 2017 03:33 PM	edmdev
QS_Tree1		Classification	Version 1 (1.0)	Decision tree	Dec 15, 2017 05:44 PM	edmdev

See Also

- “Set Champion and Challenger Models” in *SAS® Model Manager: User’s Guide*

End of Demonstration



Publish a Champion Model

Note: Before you can complete this activity, a CAS publish destination must be configured by a SAS administrator. For more information, see “[Configuring Publish Destinations](#)” in *SAS® Model Manager: Administrator’s Guide*.

1. Select the check box next to the champion model, **QS_Tree1**.
2. Click and select **Publish**.

Name	Role	Model Functi...	Project Version	Algorithm	Date Modified	Modi
QS_Reg1		Classification	Version 1 (1.0)	Logistic regression	Apr 19, 2018 04:11 PM	stude
<input checked="" type="checkbox"/> QS_Tree1		Classification	Version 1 (1.0)	Decision tree	Apr 19, 2018 04:30 PM	stude

The 'Publish' option is highlighted in the context menu.

The Publish Models window appears.

3. Select a publish destination (for example, maslocal).

Note: The publish destinations on your computer might be different from those shown here.

Destination:		
Name	Type	CAS Server
maslocal	Micro Analytic Service	

4. (Optional) If you have previously published a model, expand the **Items to Publish** section, and select the check box in the **Replace** column for each model that you want to replace.

The screenshot shows the 'Publish Models' dialog box. At the top, it says 'Destination:' followed by a table with three rows:

Name	Type	CAS Server	CAS Library	Model Ta...
_CAS_PUBLIC_	CAS	cas-shared-default	Public	publishPublic
TERADATA	Teradata	cas-shared-default	CASUSER	supermode

The row for '_CAS_PUBLIC_' is highlighted. Below the table, it says 'Selected: _CAS_PUBLIC_'. Under the heading '▼ Items to Publish', there is a table with one row:

Name	Published Name	Replace
QS_Tree1 (1.1)	QS_Tree1	<input type="checkbox"/>

At the bottom right are 'Publish' and 'Cancel' buttons.

Publish destinations are created when the software is installed, or they can be added by an administrator later. Three destinations are shown in the display capture above.

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

The screenshot shows a modal window titled "Publishing Results". It contains a table with three columns: "Name", "Published Name", and "Status". There is one row in the table. The "Name" column contains "QS_Tree1", the "Published Name" column contains "QS_Tree1", and the "Status" column contains "Published successfully". A "Close" button is visible at the bottom right of the window.

Name	Published Name	Status
QS_Tree1	QS_Tree1	Published successfully

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

See Also

- “Publishing Models” in *SAS® Model Manager: User’s Guide*

End of Demonstration



Exercises

3. Importing Models from SAS Package File, PMML, and ZIP Formats

- Open the QS_PVA project created in the previous exercise.
- Import a model from a SAS Package file. Import the **PVA_Regression.spk** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Were input and output variables documented when this model was imported?
- Check and modify the model properties as necessary, paying close attention to those shown below.

Target variable:	TARGET_B
Target event value:	1
Target level:	Binary
Output event probability variable:	P_TARGET_B1

Save any changes that you make.

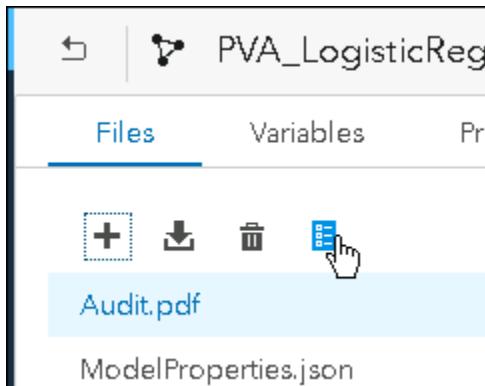
- Go back to the QS_PVA project and import a model in PMML format. Import the **PVA_Tree_pmml.xml** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Were the input and output variables correctly specified?
- What additional files beside the XML file were created during the import?
- Import a model containing all relevant files in ZIP format. Import the **PVA_LogisticRegression.zip** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Set and save the model properties as necessary.
- Compare the available attributes of all models in your project. Which model uses the least number of input variables? Which model or models contain fit statistics for the validation data? Where does this information come from?
- Run tests of the models to make sure the score code is valid. Use the **pva_score** data set located on the computer local directory **D:\Workshop\Winsas\Viya_DM\Exercise\Data\pva_score**.

Note: You will have to import the data source before creating the tests.

Hint: You can import data sources directly from the test dialog box.

- Set **PVA_LogisticRegression** as the champion model and publish it. Use the variable **P_TARGET_B1** as the project output variable.

- j. Add a PDF document to your champion model files as supplemental information for background on the model. Use the PDF located in **D:\Workshop\Winsas\Viya_DM\Exercise\Data\ Audit.pdf**. Assign the role of documentation to this file.
- k. Update the label by clicking the **Properties** icon shown below.



- I. From the list of available attributes, select **Documentation**.
- m. Save the model.

End of Exercises

2.6 Creating a Model Performance Report

Objectives

- Prepare model performance reporting data.
- Download model score code.
- Use SAS Studio to access data and create performance reports.

48



Model Performance Reporting

You can monitor model performance by using a performance reporting macro in SAS Studio in your SAS Viya environment with these steps:

1. Prepare the performance data source.
2. Obtain the project and model UUIDs from the properties in Manage Models.
3. Upload the score code from SAS Model Manager to the SAS Studio session.
4. Establish a connection to CAS.
5. Generate the performance data sets.
6. Create and run your report code using the performance reporting macros.

49



The performance results tables that are produced using the macros can then be viewed in SAS Studio, SAS Environment Manager, or SAS Visual Analytics.

For an example of monitoring the performance of a champion model, see “[Performance Monitoring Macros](#)” in SAS® Model Manager: Macro Reference.

Performance Monitoring Macro Syntax

```
%MM_PERFORMANCE_MONITOR (
    PERFLIB=monitoring-input-caslib,
    PERFDATANAMEPREFIX=input-data-prefix,
    MM_MART=monitoring-output-caslib,
    ASTOREFREF=fileref-to-analytic-store-file,
    SCORECODEFILEREF=fileref-to-score-code-file,
    RUNSCORE=flag-for-scoring,
    SAVERESULT=flag-for-saving-performance-results
);
```

50

Copyright © SAS Institute Inc. All rights reserved.



Required Arguments

PERFLIB=monitoring-input-caslib

specifies the caslib for the performance input data.

perfLib=mmlib

PERFDATANAMEPREFIX=input-data-prefix

specifies the prefix of the performance input table name.

MM_MART=monitoring-output-caslib

specifies the caslib for the performance monitor results.

ASTOREFREF=fileref-to-analytic-store-file

specifies the fileref for the model analytic store file.

SCORECODEFILEREF=fileref-to-score-code-file

specifies the fileref for the model score code file.

scoreCodeFref=ep

When the DS2 embedded process code contains a reference to an analytic store file, you can create a fileref for the DS2 EP code. You can then pass the fileref to the %MM_PERFORMANCE_MONITOR macro.

filename ep '/r/ge.unx.sas.com/vol/vol120/u12/scnkuj/Models/forest_ep.sas';

RUNSCORE=flag-for-scoring

specifies the flag that indicates whether scoring should be run or not. The values are Y or N.

SAVERESULT=flag-for-saving-performance-results

specifies to save the performance results out of CAS memory. The values are Y or N.

Create a CAS Session

```

1  ****
2  /* Set the options necessary for creating a connection to a CAS server. */
3  /* Once the options are set, the cas command connects the default session */
4  /* to the specified CAS server and CAS port, for example the default value */
5  /* is 5570. */
6  ****
7
8  options cashost=server.demo.sas.com casport=5570;
9  cas;
10
11 ****
12 /* Start a session named mySession using the existing CAS server connection */
13 /* while allowing override of caslib, timeout (in seconds), and locale */
14 /* defaults. */
15 ****
16
17 cas mySession sessopts=(caslib=casuser timeout=1800 locale="en_US");
18 ****
19 /* Create a default CAS session and create SAS librefs for existing caslibs */
20 /* so that they are visible in the SAS Studio Libraries tree. */
21 ****
22
23 cas;
24 caslib all assign;

```

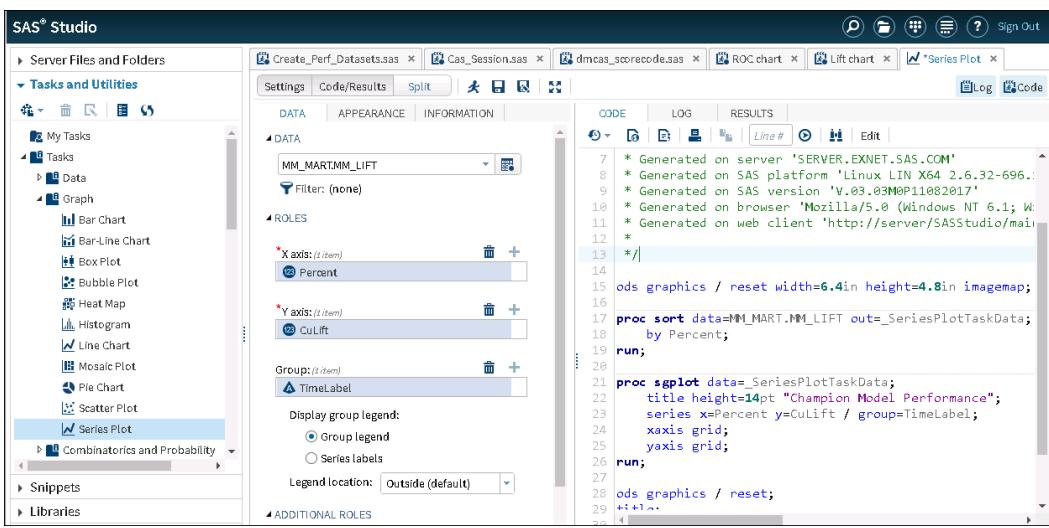
Copyright © SAS Institute Inc. All rights reserved.



These statements when run in SAS Studio make the data sources that you used in Model Manager accessible. The performance data sets need to be in CAS memory to run the performance reports.

The rest of the code used to create reports is shown in the demonstration.

Creating Reports with SAS Studio



The screenshot shows the SAS Studio interface with the 'Series Plot' task selected in the left sidebar. The central workspace has two tabs: 'DATA' and 'CODE'. The 'DATA' tab shows roles assigned to the X and Y axes. The 'CODE' tab displays the following SAS code:

```

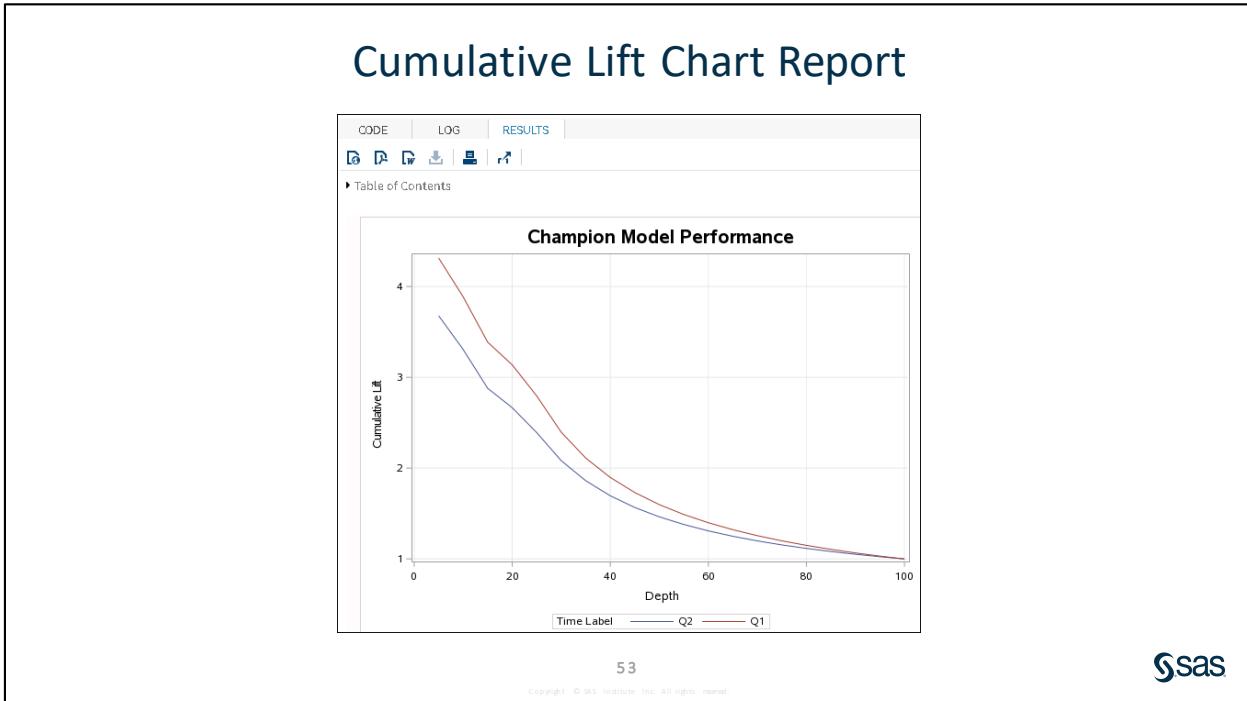
7  * Generated on server 'SERVER.EXNET.SAS.COM'
8  * Generated on SAS platform 'Linux LIN X64 2.6.32-696.-
9  * Generated on SAS version 'V.03.03M0P11082017'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 6.1; W-
11 * Generated on web client 'http://server/SASStudio/main-
12 */
13
14 ods graphics / reset width=6.4in height=4.8in imagemap;
15
16 proc sort data=MM_MART.MM_LIFT out=_SeriesPlotTaskData;
17   by Percent;
18 run;
19
20 proc sgplot data=_SeriesPlotTaskData;
21   title height=14pt "Champion Model Performance";
22   series x=Percent y=Culift / group=TimeLabel;
23   xaxis grid;
24   yaxis grid;
25 run;
26
27 ods graphics / reset;
28
29 */

```

Copyright © SAS Institute Inc. All rights reserved.



After generating the performance data sets, you can use SAS Studio to create your own report. In the demonstration, some of the generated report code has been provided for the lift and ROC charts.





Run Performance Reports

Create the performance report data sets and run a performance report on these data sets in SAS Studio.

1. Record the project and model UUIDs from their respective properties in Model Manager and save them to a text file using Notepad on your computer.

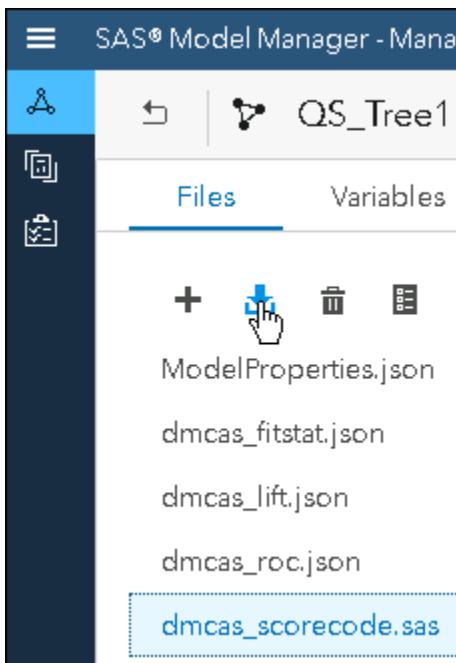
Project Properties

General	
Target level:	(none)
Output event probability variable:	(none)
UUID:	9c71c5a3-5d58-4674-a156-d18413afe5bb

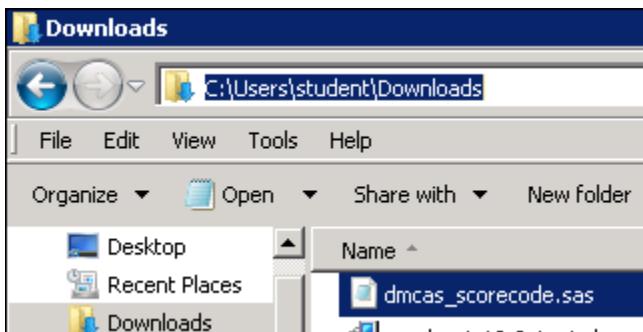
Model Properties

General	
Tool:	Model Studio
Tool version:	8.2
UUID:	c93644d0-528e-4e50-a098-729225134529
External model ID:	ee34d555-cefd-4aef-9475-03141ca86c84

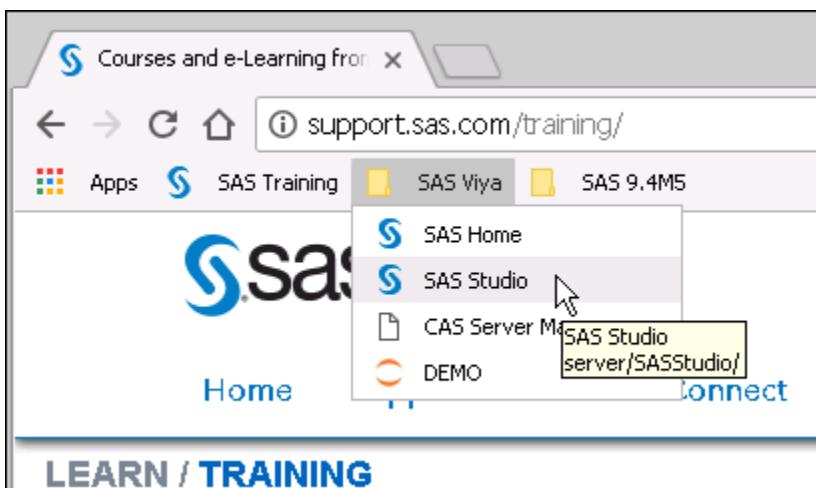
2. Download a copy of the score code for the QS_Tree1 model to the local machine and note the file location.



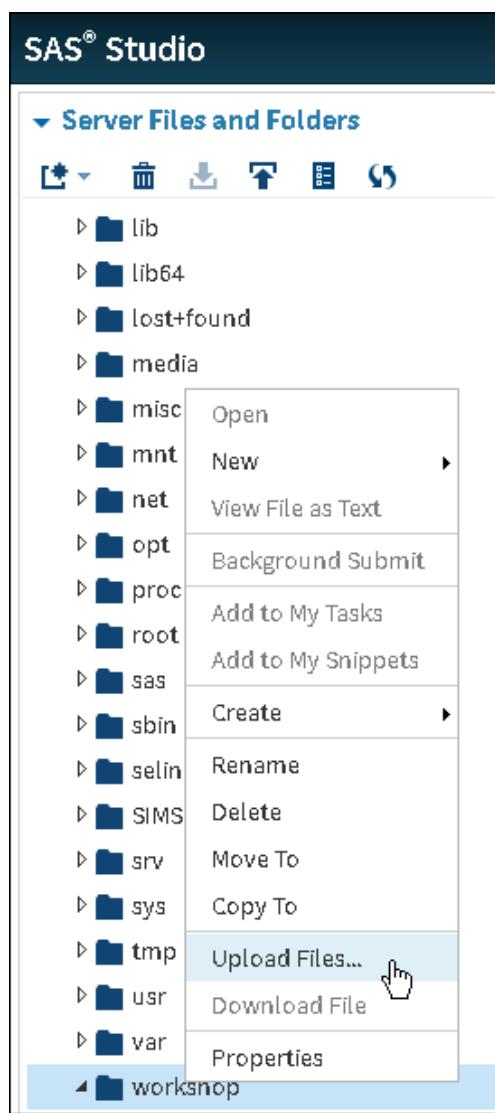
File Download containing model score code:



3. Open SAS Studio (on SAS Viya) and upload the score code to the server **files** list under **/workshop**.



Upload score code to the CAS Server:



The screenshot shows the SAS Studio interface with the title "SAS® Studio". In the top navigation bar, there is a dropdown menu labeled "Server Files and Folders". Below the navigation bar is a toolbar with icons for file operations: New, Open, Save, Copy, Paste, Find, and Refresh. The main area displays a hierarchical list of server files and folders. A context menu is open over the "misc" folder, listing options such as Open, New, View File as Text, Background Submit, Add to My Tasks, Add to My Snippets, Create, Rename, Delete, Move To, Copy To, Upload Files..., Download File, and Properties. The "Upload Files..." option is highlighted with a blue background and a cursor icon pointing at it.

Upload Files

Upload files to: /workshop

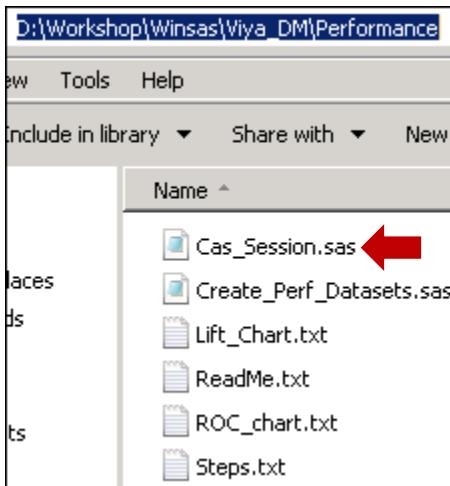
Choose Files

Selected files:

1	SAS	dmcas_scorecode.sas	13.2 kb
---	-----	---------------------	---------

Upload Cancel

4. Open the **Cas_Session.sas** program into SAS Studio. Run it to establish the CAS session.



5. In the same manner as in steps 3-4, upload the **Create_Perf_Datasets.sas** job into the /workshop directory. ***Update it with the project and model UIDs*** that you saved into Notepad previously and ***check the input and output references*** in the program for accuracy and adjust as necessary. This job generates the performance results data sets.

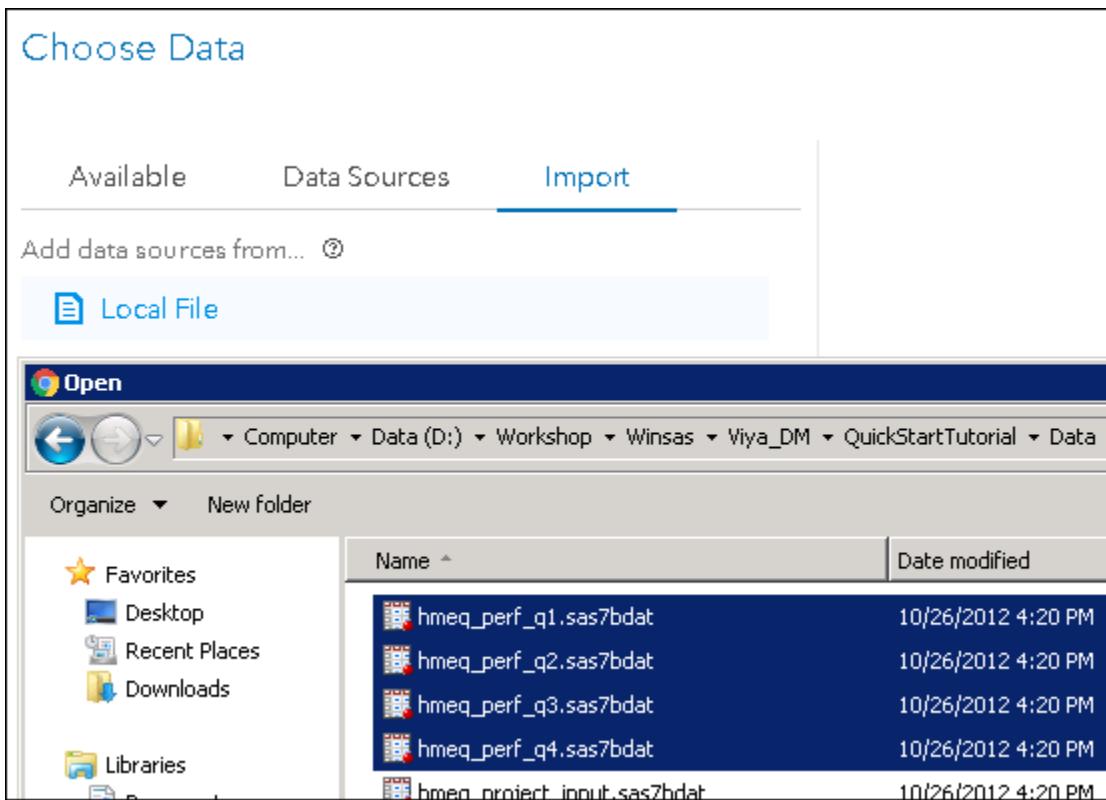
```

Create_Perf_Datasets.sas x Cas_Session.sas x dmcas_scorecode.sas x
CODE LOG RESULTS OUTPUT DATA
cas _mmcas_;
caslib _all_ assign;
%let _mm_projectuuid=%nrstr(9c71c5a3-5d58-4674-a156-d18413afe5bb);
%let _mm_modelid=%nrstr(b25dc1d7-3d9c-40ed-9914-da3b7ffc0cbb);
%let mm_modelflag = 0.

```

6. Add the performance data sets and upload them into CAS. They are located in **D:\workshop\winsas\viya_DM\ QuickStartTutorial\Data**. The files are called **hmeq_perf_q1 - q4**.

7. Go back into Explorer and click **Visualize Data** \Rightarrow **Add Data Source** \Rightarrow **Data** \Rightarrow **Import** \Rightarrow **Local File** \Rightarrow D:\Workshop\Winsas\Viya_DM\QuickStartTutorial\Data.



8. Select the four **hmeq_Perf** data sets and click **Open**. Select **Import All** and **OK**.
9. Run the **Create_Perf_Datasets** job.

10. The performance data sets contain the name of the Model Manager project and are shown below in SAS Studio under Server Files and Folders.

The screenshot shows the SAS® Studio interface with the title "SAS® Studio". Under the "Server Files and Folders" section, the "casuser" folder is expanded, revealing its contents:

- images
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_fitstat.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_job_history.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_ks.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_lift.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_meta.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_model_indicator.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_model_status.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_roc.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_var.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_var_deviation.sashdat
- 53a94885-e539-4256-9627-f35e517b2ec3.mm_var_summary.sashdat
- Best_Model_gbt.sashdat
- gbt_model.sashdat

11. Upload the **Lift_Chart.txt** program into SAS Studio. Rename it to a .sas file and run it to create and view the chart.
12. Upload the **Roc_Chart.txt** program into SAS Studio. Rename it to a .sas file and run it to create and view the chart.

End of Demonstration

2.7 Import, Enable, and Use a Workflow

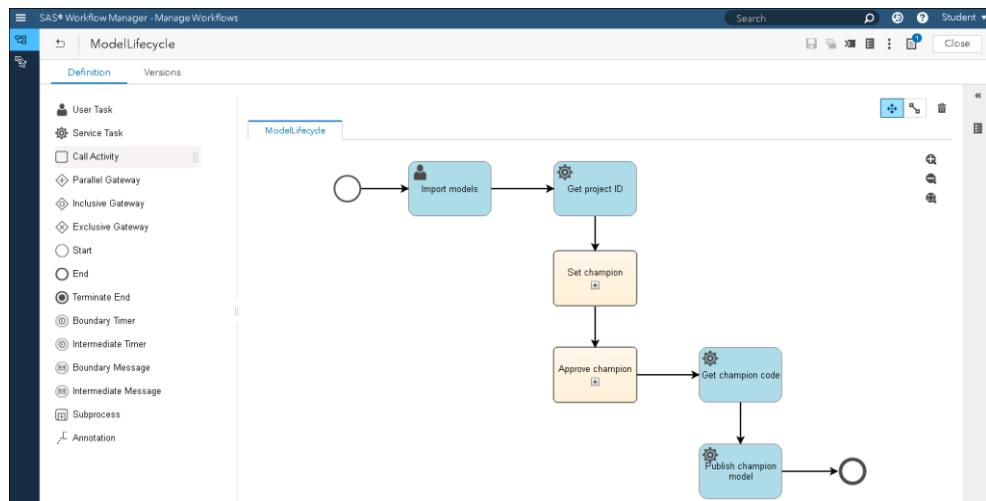
Objectives

- Import a workflow definition.
- Attach a workflow to a Model Manager project.
- Track Model Manager tasks using a workflow.

56



Workflow Definition

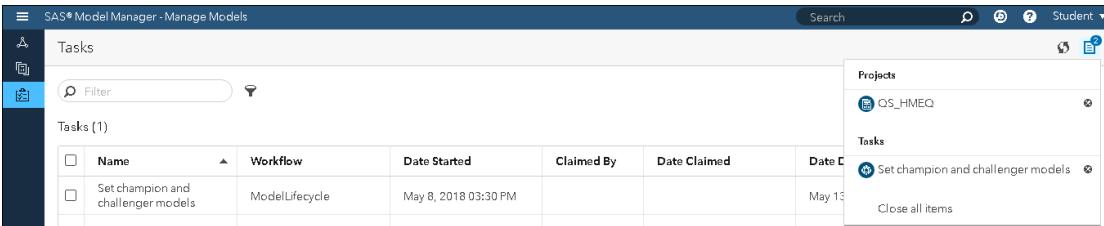


57



In Manage Workflows, you can upload a workload definition or create a new one. This is a workflow definition that we will attach to Model Manager and use it to track steps in our process flow.

Workflow Task Navigation



The screenshot shows the SAS Model Manager interface with the title "SAS® Model Manager - Manage Models". The left sidebar has icons for Home, Projects, and Tasks. The main area is titled "Tasks" with a sub-section "Tasks (1)". A table lists one task:

Name	Workflow	Date Started	Claimed By	Date Claimed	Date Due
Set champion and challenger models	ModelLifecycle	May 8, 2018 03:30 PM			May 13

On the right, there are sections for "Projects" (QS_HMEQ) and "Tasks" (Set champion and challenger models). Below the table are buttons for "Close all items" and "Close all items".

sas

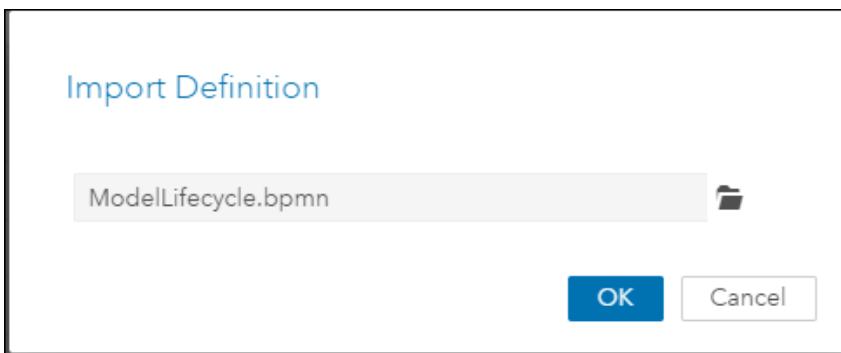
The Set champion and challenger models task shown above must be done by a user who is in the SAS Workflow Editors or the SAS Workflow Definition Administrators user group. For more information, see [SAS® Workflow Manager: Administrator's Guide](#).

If you want to create your own workflow definition, see [SAS® Workflow Manager: Quick Start Tutorial](#).



Add a Workflow Definition

1. Click and select **Manage Workflows**.
2. Click **Import**. The Import Definition window appears.
3. Click and navigate to where the workflow definition is stored (for example, D:\Workshop\Winsas\Viya_DM\QuickStartTutorial\Workflow Definitions).
4. Select the **ModelLifecycle.bpmn** file and click **Open**.



5. Click **OK**.
6. Open the workflow definition that you just imported (for example, **ModelLifecycle**).

TIP Use the **Filter** field to find the workflow definition that you imported.

7. Set the client identifier.

Note: The client identifier denotes the application that starts the workflow. You must specify a client identifier for a workflow definition before it can be enabled.

- a. Click in the object toolbar, which is in the top upper right of the window. The Definition Properties window appears.
- b. Select **SAS Model Manager** for the client identifier and click **OK**.
8. Enable the workflow definition.
 - a. Click the **Versions** tab. Notice that SAS Workflow Manager has saved your workflow definition as the **Current** version.
 - b. Click to create a static numbered version of the workflow definition. SAS Model Manager assigns the number 1.0 to the version.
 - c. Click to enable version 1.0 of the workflow definition.

The check mark in the **Enabled** column indicates the version that is enabled.
- d. Click **Save** and **Close** to close the workflow definition.

TIP If your workflow has multiple versions, you can view different versions by selecting the version on the Versions page, and then clicking the **Definitions** tab. The selected check box indicates the version that is displayed on the Definitions tab.

See Also

- [SAS® Workflow Manager: Quick Start Tutorial](#)

End of Demonstration



Start a Workflow

You must be in the SAS Workflow Process Administrators group to complete this demonstration. For more information, see [SAS® Workflow Manager: Administrator's Guide](#).

1. Click and select **Manage Models**.
2. Open the **QS_HMEQ** project and click the **Workflow** tab.
3. Click **Start Workflow** and select the **ModelLifecycle** workflow definition from the list. The Start Workflow window appears.
4. Specify values for any prompts that are displayed.

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

For example, specify a value for the project name.

The screenshot shows a 'Start Workflow' dialog box. At the top, it says 'Start Workflow'. Below that, it says 'Name: ModelLifecycle'. Underneath, it says 'Project name: *' followed by a text input field containing 'QS_HMEQ'. At the bottom right, there are two buttons: a blue 'Start' button and a white 'Cancel' button.

5. Click **Start**. The workflow is added to the list with a status of **In progress**.

The screenshot shows the SAS Model Manager interface with the 'Workflow' tab selected. The title bar says 'QS_HMEQ'. The 'Workflow' tab has a 'Start Workflow ▾' button. The main area displays a table with columns: Name, Status, Started By, Date Started, Date Ended, and Description. One row is shown: 'ModelLifecycle' with 'In progress' status, started by 'sasdemo' on 'Dec 15, 2017 06:34 PM', and a description 'Sample workflow for performing actions in SAS Model Manager.'

Name	Status	Started By	Date Started	Date Ended	Description
ModelLifecycle	In progress	sasdemo	Dec 15, 2017 06:34 PM		Sample workflow for performing actions in SAS Model Manager.

See Also

- [“Start a New Workflow” in SAS® Model Manager: User’s Guide](#)

End of Demonstration



Complete Workflow Tasks

You must be in the SAS Workflow Process Administrators group to complete this exercise. For more information, see [SAS® Workflow Manager: Administrator's Guide](#).

1. Click .
2. Click a task to open it.
3. Click to claim the task.
4. Specify values for any prompts that are displayed on the Prompts tab.
5. Click the **Properties** tab to view task properties, including the associated object.
6. Click the name of the associated object (for example, **Project: QS_HMEQ**) to open it.
7. Complete any actions that are associated with the task. An example is importing models into a project.
8. Click to switch back to the task object.
9. Click **Complete**.
10. Click **Complete** in the confirmation message.
11. Repeat steps 2 through 10 until all of the workflow tasks have been completed.

See Also

- “Working with Tasks” in [SAS® Model Manager: User’s Guide](#)

End of Demonstration

2.8 Solutions

Solutions to Exercises

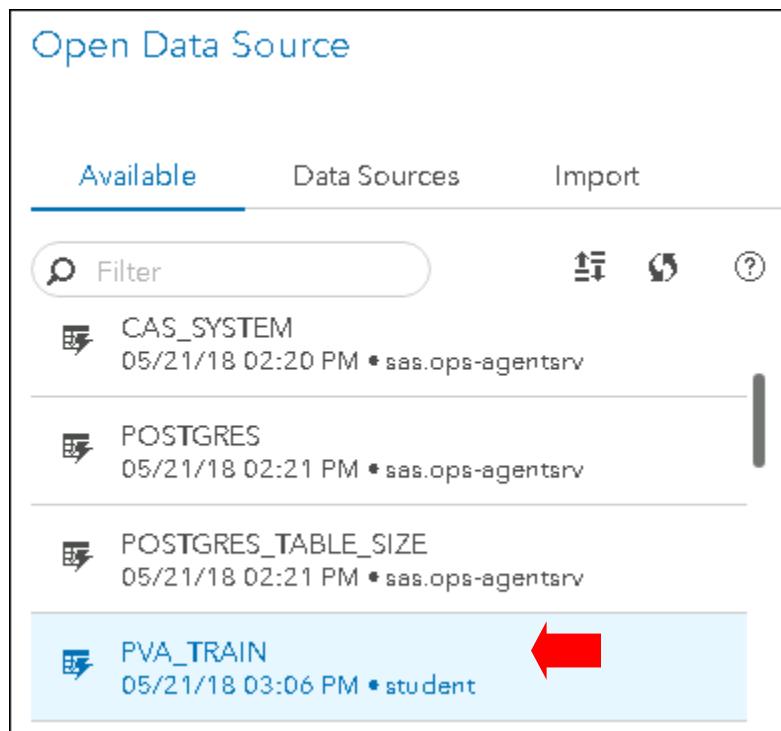
1. Accessing Data and Creating a Model Repository and a New Project

The complete instructions are in the exercise.

2. Importing a Data Source and Profiling the Data

- a. Go to **Explore and Visualize data** from the applications menu . Click **Data** \Rightarrow **Import**, select **Local File**, and load the **pva_train** local data source from **D:\Workshop\Winsas\Viya_DM\QuickStartTutorial\Data**.
- b. Select **Import item** and click **OK**.
- c. Click **Available**, and scroll down to **PVA_TRAIN**.

Notice that the icon has a lightning bolt, indicating that the data source is loaded into memory.



The screenshot shows the "Open Data Source" dialog box with the "Available" tab selected. There are three tabs at the top: "Available" (which is highlighted), "Data Sources", and "Import". Below the tabs is a search bar labeled "Filter" with a magnifying glass icon. To the right of the search bar are three icons: a lightning bolt, a gear, and a question mark. The main area displays four data sources in a list:

- CAS_SYSTEM
05/21/18 02:20 PM • sas.ops-agentsrv
- POSTGRES
05/21/18 02:21 PM • sas.ops-agentsrv
- POSTGRES_TABLE_SIZE
05/21/18 02:21 PM • sas.ops-agentsrv
- PVA_TRAIN
05/21/18 03:06 PM • student

A red arrow points to the "PVA_TRAIN" entry in the list.

- d. Select **Sample Data** to view data in the **pva_train** table. Do any of the variables have missing data?

Donor_Age and Income_Group have missing values represented by . (a period) in the Sample Data table.

The screenshot shows the 'Open Data Source' dialog with the 'Available' tab selected. A list of data sources is on the left, and the 'PVA_TRAIN' table is selected. On the right, the 'Sample Data' tab is active, displaying a preview of the table with 100 rows. The columns are labeled: CONTROL, DONOR_AGE, SES, INCOME_GROUP, HOME_VALUE, PEP_STAR, ATUS_97NK, GIFT_AMT, and DONEE. Some cells contain missing values represented by a period ('.').

CONTROL	DONOR_AGE	SES	INCOME_GROUP	HOME_VALUE	PEP_STAR	ATUS_97NK	GIFT_AMT	DONEE
0000005	87	?	2	654	1	1	15	
00000142	.	2	.	475	1	2	10	
00000038	.	2	6	1688	1	3	13.44	
00000159	81	2	4	530	1	4	9.64	
00000041	74	2	2	514	0	1	17.5	

- e. From Data Sources, run a profile against the **pva_train** data source. This provides summary statistics on the contents of the file.
- f. What is the mean of **DONOR_AGE**? **58.89**

What is the maximum value of **RECENT_AVG_GIFT_AMT**? **15.41**

The screenshot shows the 'Profile' tab for the PVA_TRAIN data source. It displays summary statistics for several columns:

Column	Mean	Median
CARD_PROM_12	5.35	
CONTROL_NUMBER		
DONOR_AGE	58.89	
FILE_CARD_GIFT	5.26	

Column	Mean
RECENT_AVG_GIFT_AMT	15.41

- g. Click **OK**.
- h. Click **≡** and select **Manage Models** to return to SAS Model Manager.

3. Importing Models from SAS Package File, PMML, and ZIP Formats

- Open the **QS_PVA** project created in the previous exercise.
- Import a model from a SAS Package file. Import the **PVA_Regression.spk** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Were input and output variables documented when this model was imported?

Yes. Select the Variables window when viewing the model to see which variables are inputs and outputs.

<input type="checkbox"/>	Name	Data Type	Input	Output
<input type="checkbox"/>	CARD_PROM_12	# Decimal	✓	
<input type="checkbox"/>	DONOR AGE	# Decimal	✓	
<input type="checkbox"/>	FREQUENCY STATUS 97NK	# Decimal	✓	
<input type="checkbox"/>	INCOME GROUP	# Decimal	✓	
<input type="checkbox"/>	MEDIAN HOME VALUE	# Decimal	✓	
<input type="checkbox"/>	MONTHS SINCE FIRST GIFT	# Decimal	✓	
<input type="checkbox"/>	MONTHS SINCE LAST GIFT	# Decimal	✓	
<input type="checkbox"/>	PEP_STAR	# Decimal	✓	
<input type="checkbox"/>	RECENT CARD RESPONSE PROP	# Decimal	✓	
<input type="checkbox"/>	SES	▲ Character	✓	
<input type="checkbox"/>	IMP_DONOR AGE	# Decimal		✓
<input type="checkbox"/>	L_TARGET_B	▲ Character		✓
<input type="checkbox"/>	M_DONOR AGE	# Decimal		✓

- Check and modify the model properties as necessary, paying close attention to those shown below.

Target variable:	TARGET_B
Target event value:	1
Target level:	Binary
Output event probability variable:	P_TARGET_B1

Save any changes that you make.

These selections are made in the Properties window of the PVA_Regression model. A portion of the window is shown here for reference.

General	
Name:	PVA_Regression
Description:	
Created by:	student
Date created:	Apr 19, 2018 05:30 PM
Date modified:	Apr 19, 2018 05:35 PM
Location:	/Model Repositories/Training_Repository/Exercises
Project name:	<u>OS_PVA</u>
Project version:	Version 1
Function:	Classification
Score code type:	DATA step

- d. Import a model in PMML format. Import the **PVA_Tree_pmml.xml** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Were the input and output variables correctly specified?

Yes

- e. What additional files beside the XML file were created during the import?

The five total files are listed in the window below.

SAS® Model Manager - Manage Models

PVA_Tree_pmml (1.0)

Files Variables Properties

+ Download Delete Details

score.sas

PVA_Tree_pmml.xml

input.xml

output.xml

target.xml

- f. Import a model containing all relevant files in ZIP format. Import the **PVA_LogisticRegression.zip** model from **D:\Workshop\Winsas\Viya_DM\Exercise\Models**. Set and save the model properties as necessary.
- g. Compare the available attributes of all models in your project. Which model uses the least number of input variables? Which model or models contain fit statistics for the validation data? Where does this information come from?

Examine the Files window for each model in the project. Any model that has a file with fitstat as part of the name contains a JSON file with this information. This information, if provided, was part of the model when it was imported. Models coming from Model Studio or SAS Enterprise Miner usually contain this type of information.

- h. Run tests of the models to make sure the score code is valid. Use the **pva_score** data set located on the computer local directory **D:\Workshop\Winsas\Viya_DM\Exercise\Data\pva_score**.

Note: You will have to import the data source before creating the tests.

Hint: You can import data sources directly from the test dialog box.

Choose a Data Source

Available Data Sources Import 1

Import(1) +

pva_score.sas7bdat

The table was successfully imported on Apr 19, 2018 05:40 PM and is ready for use.

Target table name: * pva_score

Target destination: * cas-shared-default/Public

Import Item

i. Set **PVA_LogisticRegression** as the champion model and publish it.

These actions are taken from the Models window of the project. Select the check box next to the model that you want to work with, click the **Actions** icon at the top right of the window, and make your selection from the menu that appears. See the window below for details.

Name	Role	Model Function	Project ...	Algorithm	Date Modified
Neural_PVA_pmml		Classification	Version 1 (1.0)	Nonlinear Optimization	May 18, 2018 11: AM
PVA_LogisticRegression		Classification	Version 1 (1.0)	Logistic regression	Apr 19, 2018 06: PM
PVA_Regression		Classification	Version 1 (1.0)	Regression	Apr 19, 2018 05: PM
PVA_Tree_pmml		Classification	Version 1 (1.0)		Apr 19, 2018 05: PM
modelIn2_fromZip					May 18, 2018 11:30 student

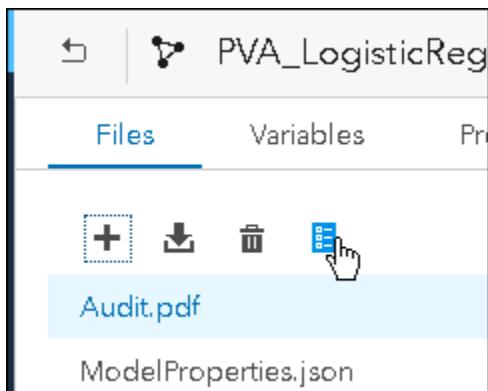
j. Add a PDF document to your champion model files as supplemental information for background on the model. Use the PDF located in **D:\Workshop\Winsas\Viya_DM\Exercise\Data\ Audit.pdf**. Assign the role of documentation to this file.

Add Model Files

Select one or more files:

Add Cancel

- k. Update the label by clicking the **Properties** icon shown below.



- l. From the list of available attributes, select **Documentation**.

A screenshot of the 'Documentation' properties dialog box. On the left, there's a vertical list of attributes: 'Analytic store', 'Python pickle', 'PMML', 'Documentation' (which is highlighted with a light blue background and has a hand cursor pointing at it), and 'Data plan'. Below this list is a dropdown menu with the placeholder 'Select or enter a value'. On the right side of the dialog box, there are two sections: 'Date modified:' with the value 'April 19, 2018 06:00:31 PM' and 'Modified by:' with the value 'student'. At the bottom right of the dialog box is a blue 'Save' button.

- m. Save the model.

End of Solutions

Solutions to Student Activities (Polls/Quizzes)

2.02 Poll – Correct Answer

Data must be loaded into memory before it can be used by Model Manager.

- True
- False

Chapter 3 Decision Manager

3.1	Introduction	3-3
3.2	Accessing Data.....	3-7
	Demonstration: Access the Data Files	3-9
3.3	Creating Rule Sets.....	3-12
	Demonstration: Create a Rule Set.....	3-16
	Exercises.....	3-21
3.4	Creating and Testing Decisions	3-26
	Demonstration: Create and Test a Decision.....	3-28
	Exercises.....	3-31
3.5	Lookup Tables (Self-Study)	3-38
	Demonstration: Use a Lookup Table.....	3-43
3.6	Solutions	3-44
	Solutions to Exercises	3-44

3.1 Introduction

Objectives

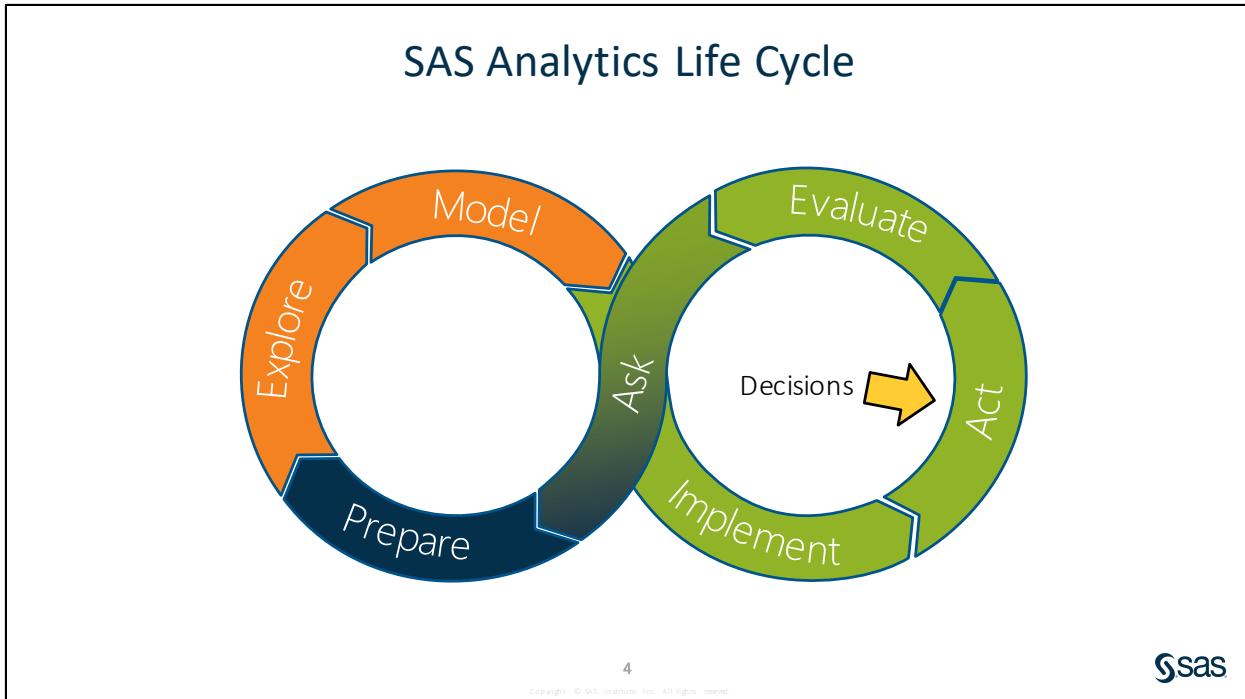
- Describe a rule set.
- Describe a decision.

3

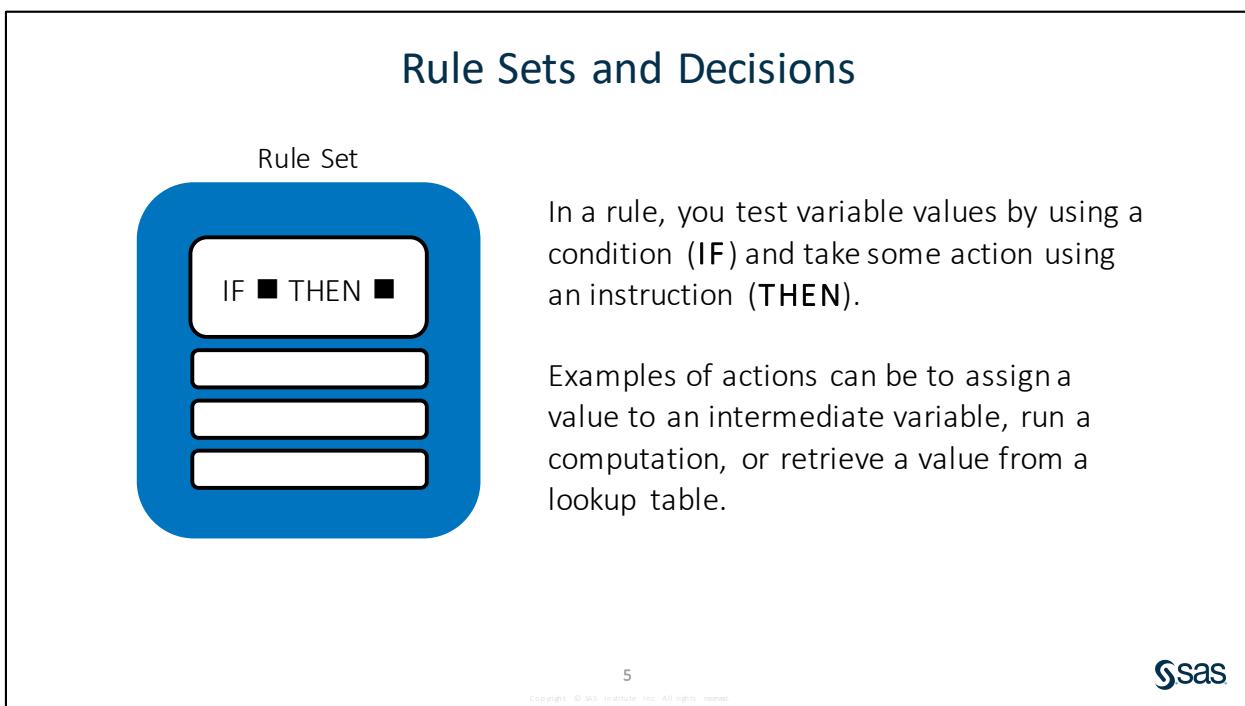


At the completion of this chapter, you create three rule sets and combine these rule sets together with conditional logic into a decision. The decision evaluates residential loan applications. It identifies loan applications that need to be reviewed manually, and it approves or rejects other loan applications based on certain criteria.

This section introduces the concepts of rule sets and decisions.



In the Act phase of the analytics life cycle introduced in Chapter 1, we use two types of decisions: operational decisions that are often automated, and strategic decisions that have a long-term impact. Decisions can use output from predictive models, business rule logic, conditions based on values read from the prepared data, or any combination of these.



A rule set is just a collection of one or more business rules. Business rules specify conditions to be evaluated and actions to be taken if those conditions are satisfied.

Rule Sets and Decisions

Rule Set

Debt-to-Income Ratio

IF █ THEN █

Equity

Conditional Logic

Decision

6

Copyright © SAS Institute Inc. All rights reserved.

sas

For example, if the applicant for a residential loan has a high debt-to-income ratio and little equity in the property, the loan application might be denied or flagged for additional review.

Rule Sets and Decisions

Rule Set

Decision

Rule Sets

Analytical Models

Conditional Logic

7

Copyright © SAS Institute Inc. All rights reserved.

sas

A decision combines rule sets, analytical models, and conditional logic into the decision-making process.

You can publish a decision to create an entity in another environment. This entity can be controlled, automated, and monitored. For example, if you publish a decision to the SAS Micro Analytic Service destination, SAS Decision Manager creates a DS2 package that can be managed and run through the SAS Micro Analytic Service interface.

For more information, see “[Working with Business Rules](#)” in [SAS Decision Manager: User’s Guide](#) and “[Working with Decisions](#)” in [SAS Decision Manager: User’s Guide](#).

3.2 Accessing Data

Objectives

- Sign in to SAS Decision Manager.
- Access data using Decision Manager.

9



Copyright © SAS Institute Inc. All rights reserved.

In this section, you learn how to place data in SAS Cloud Analytic Services (CAS) memory so that it can be accessed by SAS Decision Manager

Data Sources

Open Data Source

Available **Data Sources** Import

Public

HMEQ_PERF_Q1.sashdat
02/16/18 08:12 PM

HMEQ_PERF_Q2.sashdat
02/16/18 08:12 PM

HMEQ_PERF_Q3.sashdat
02/16/18 08:12 PM

HMEQ_PERF_Q4.sashdat
02/16/18 08:12 PM

HMEQ_TEST.sashdat
02/13/18 04:52 PM

HMEQ_TRAIN.sashdat

HMEQ_TEST.sashdat

Details **Sample Data** **Profile**

BAD double

LOAN double

MORTDUE double

VALUE double

REASON char

JOB char

Last profiled:
Never

Columns Rows

Size

Label:
Not available

Location:
cas-shared-default/Public

Date created:
Feb 13 2018 04:52 PM

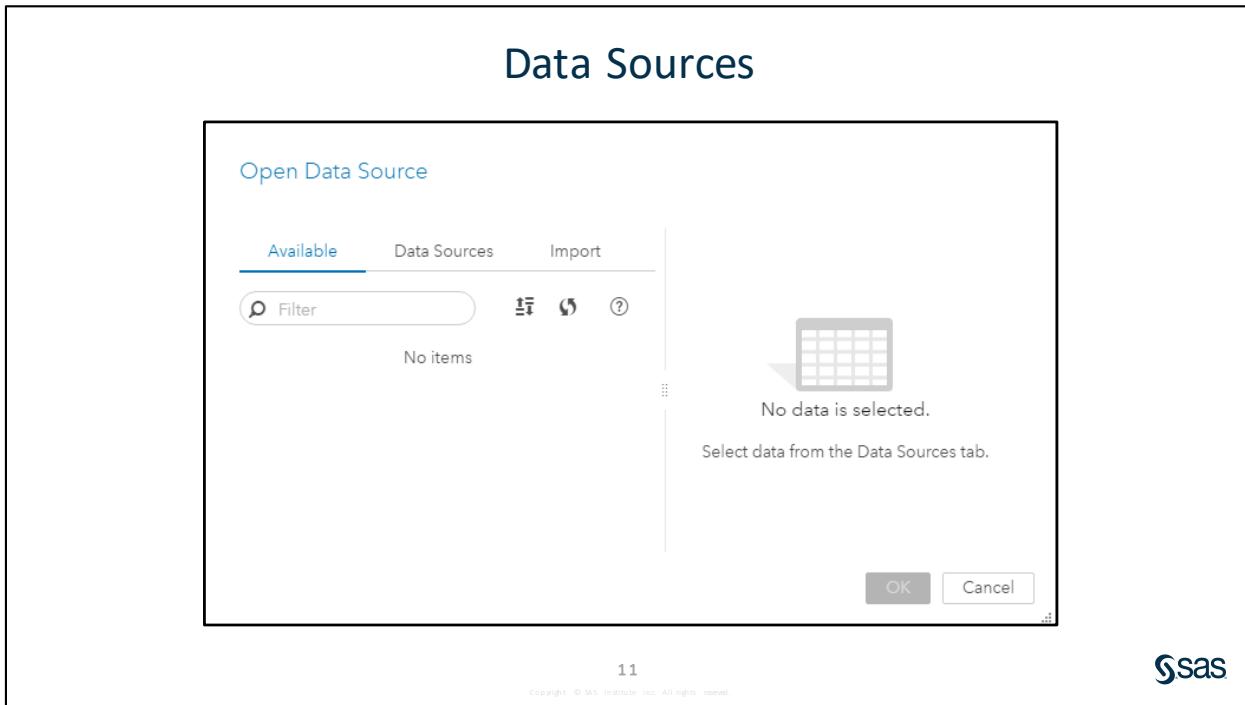
10



Copyright © SAS Institute Inc. All rights reserved.

The Open Data Source window appears at multiple points in SAS Viya, such as within the Explore and Visualize Data section under SAS Viya. Another example would be selecting the option to test a decision, where the testing wizard requires you to specify a data source. The window shown in the slide above will appear at this point, enabling you to access data. This data might already be loaded into CAS memory, it might be another registered data source that needs to be placed into memory before it can be used (as shown here), or it might be a new data table that you want to import from your local computer.

Previously imported data appears under Data Sources. Notice that the Public folder is shown here near the top left of the screen. Data sources that are loaded into memory appear twice in the list. The lightning bolt icon (top right) toggles between loading and unloading data in memory. Data can be imported from a local computer using the Import window. Summary statistics can be generated for any data table using the Profile screen.



Only data that resides in CAS memory appears under the Available window selected in this display capture.



Access the Data Files

The instructor will guide you through any steps necessary to make the data files available on your classroom computers. The following three steps can be used to load the data onto your own computer.

If the files do not already exist in D:\Workshop\Winsas\Viya_DM\QuickStartTutorial on the classroom computer, download QuickStartTutorial.zip onto your computer from <http://support.sas.com/documentation/onlinedoc/edm/>.

1. Create a folder on your machine to store the tutorial files, **D:\Workshop\Winsas\Viya_DM**.
2. Save the QuickStartTutorial.zip file into the folder that you created.
3. In Windows Explorer, right-click **QuickStartTutorial.zip** and select **WinZip** \Rightarrow **Extract to here**.

Windows creates a folder named QuickStartTutorial that contains the tutorial data sets. You can use SAS Visual Analytics to import these data sets.

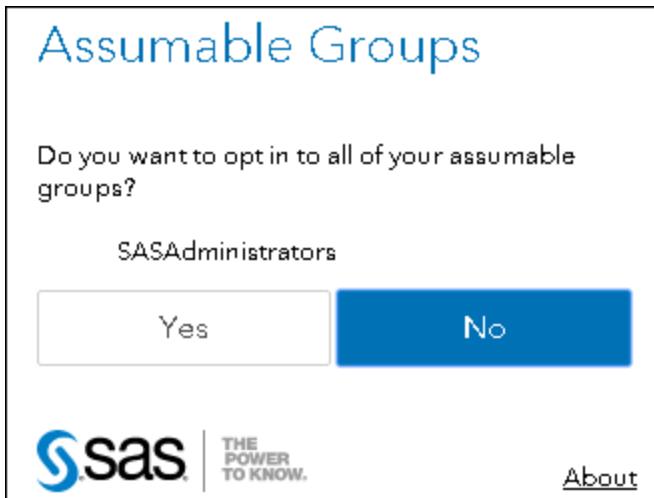
Signing In to SAS Decision Manager

To sign in to SAS Decision Manager:

1. In the address bar of your web browser, enter the URL for SAS Decision Manager <http://server/SASHome> and press Enter. The Sign In page appears.

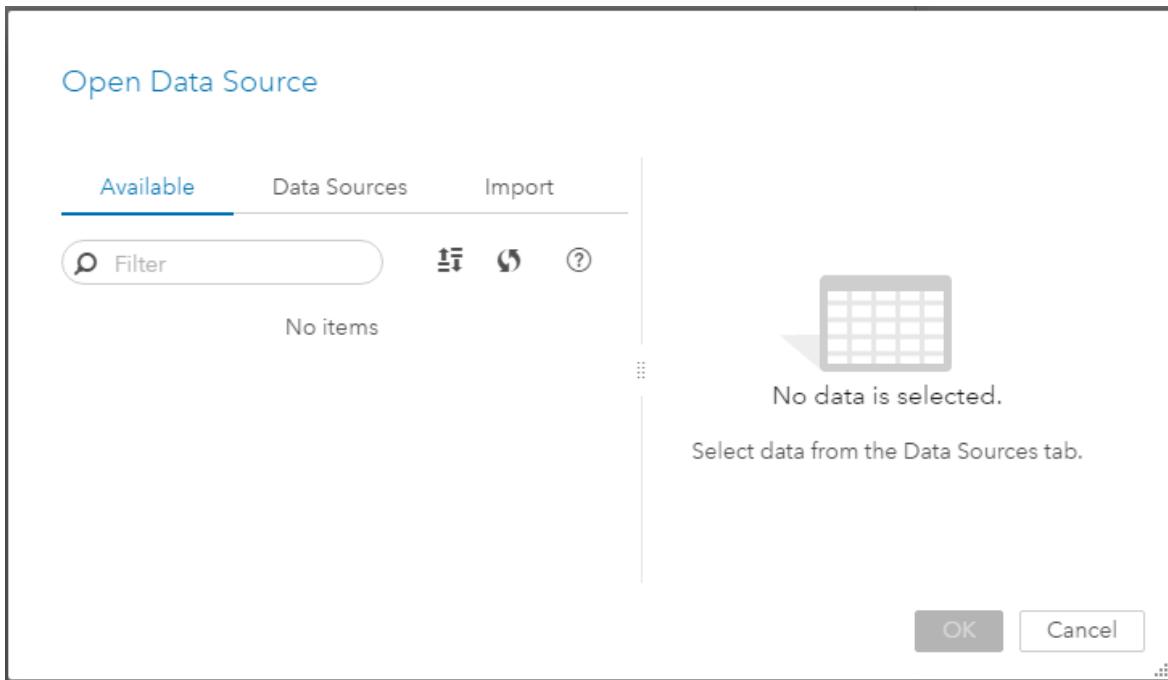
Note: Contact your system administrator if you need the URL for SAS Decision Manager. The default URL is <http://server/SASHome/SASDecisionManager>.

2. Enter **Student** for the user ID and **Metadata0** for the password.
3. Click **Sign In**.
4. If prompted, click **Yes** in the Assumable Groups window.



Importing the Data Source

1. Click and select **Explore and Visualize Data**. If you have not worked in SAS Visual Analytics, the Welcome to SAS Visual Analytics window appears.
2. Click **Data**. The Open Data Source window appears.

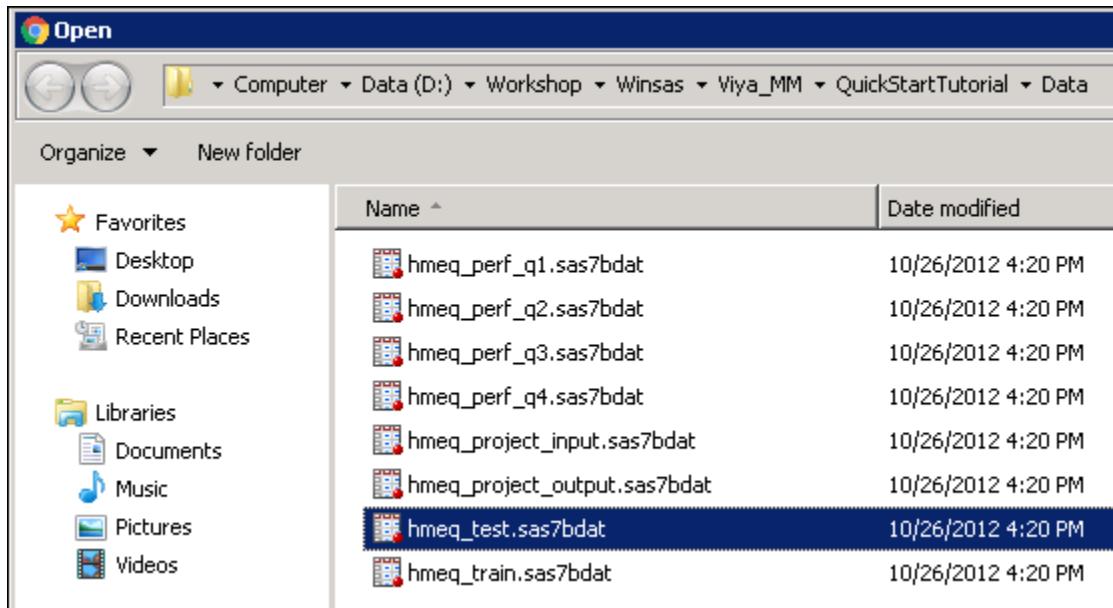


3. (Optional) Your instructor might direct you to load the data into memory if it was already imported. In this case, the table already exists in data tables. Select the table and click the lightning bolt to load it into memory.

Column	Type
BAD	double
LOAN	double
MORTDUE	double
VALUE	double
REASON	char
JOB	char

4. If the **HMEQ_TEST** table does not appear under Data Sources, click the **Import** tab and click **Local file**.

5. Navigate to the location where you saved the contents of the ZIP file in “[Download the Tutorial Files.](#)”



6. Select the data set **hmeq_test.sas7bdat** and click **Open**.
7. Click **Import Item**. By default, the table is imported into the library **cas-shared-default/Public**.
8. Click **OK** to close the Open Data Source window.
9. Click **≡** and select **Manage Decisions** to return to SAS Decision Manager.

End of Demonstration

3.3 Creating Rule Sets

Objectives

- Create and use rule sets and conditions in a decision.
- Test the decision.
- Practice building your own decision in the exercise.

14

Copyright © SAS Institute Inc. All rights reserved.



continued...

Overview of Tasks

With the exception of Steps 1 and 3, these are typical steps that you would complete to add content to your decision database. In this chapter, you complete the following eight steps:

1. Download the tutorial files.
2. Sign in to SAS Decision Manager.
3. Import the **HMEQ_TEST** data set.

15

Copyright © SAS Institute Inc. All rights reserved.



Overview of Tasks

4. Create the **Loans_To_Review** rule set. This rule set identifies loan applications that must be reviewed manually. **NOTE: This step will be performed by the instructor. The others are done in the exercise.**
5. Create the **Low_Ratio** rule set. This rule set evaluates applications for which the applicant has a low debt-to-income ratio.
6. Create the **High_Ratio** rule set. This rule set evaluates loans for which the applicant has a high debt-to-income ratio.
7. Create the **Evaluate_Loans** decision. The decision incorporates the three rule sets and uses a condition to control which rule sets are run.
8. Test and publish the decision.

16



For more information about defining rule sets and decisions, see [Working with Business Rules](#) in *SAS® Decision Manager: User's Guide* and [Working with Decisions](#) in *SAS® Decision Manager: User's Guide*.

The **Loans_to_Review** rule set identifies loans for which key information is missing, but the available data suggests that the application should be approved. These loans need to be reviewed by an underwriter. This rule set flags a loan application for review if all the conditions listed on the slide below are true.

Loans to Review Rule Set Details

- The value of the applicant's property is greater than or equal to \$120,000 (**VALUE > 120000**).
- The total amount of debt that the applicant has is less than 35.5% of their total income (**DEBTINC < 35.5**).
- The data for the applicant's total amount of debt is not missing (**DEBTINC != .**).
- The data for either the number of delinquent credit lines or the number of major derogatory reports is missing (**DEROG=. OR DELINQ=.**).

17



This slide lists four rules, each containing its own condition and action. As a designer of business rules, you decide whether you want to combine the rules into a single rule set, or to save each one into its own individual rule set.

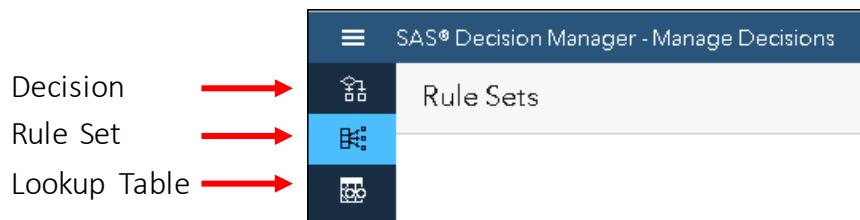
You might find that you use the same group of rules in the same order in multiple decisions. In this case, rather than duplicating the same rules over and over, a best practice would be to consider building a library of rule sets designed to be reused. Naming the rule sets with some convention that you agree on will help you find rules that address a specific purpose.

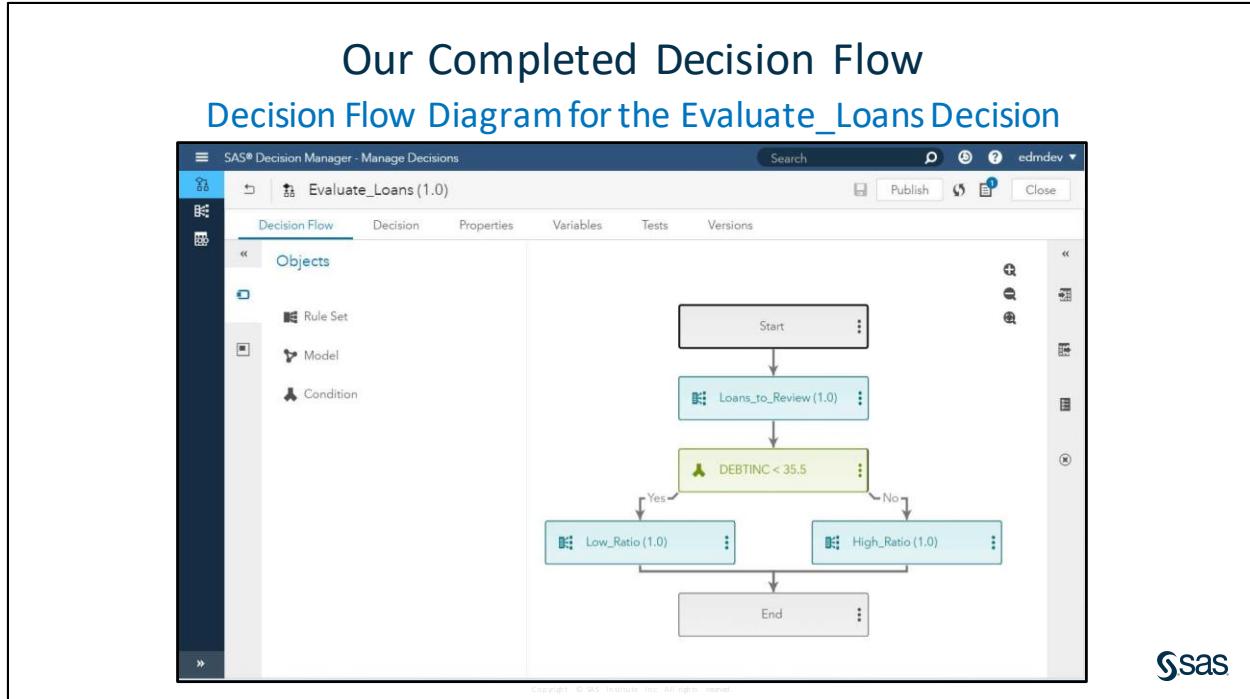
This requires some planning and foresight, and you might want to keep this in mind as you build a library of decisions in your company.

Rule Set

Steps to Create and Test a Rule Set

1. Create a new rule set.
2. Add variables to the rule set.
3. Define the rule set. (A rule set can contain more than one rule.)
4. Test the rule set.







Create a Rule Set

Creating a New Rule Set

1. From Manage Decisions, click  on the navigation bar.
2. Click **New Rule Set**. The New Rule Set window appears.
3. Enter **Loans_to_Review** for the rule set name.
4. Click  and select the folder where you want to save the rule set. Use **/My Folder** as the location.
5. Click **Save**. The application opens the new rule set and displays the Rule Set tab.

Adding Variables to the Rule Set

You can import existing variables from data sources or from other rule sets and decisions, or you can create new variables. In this demo, you import the variables that are in the table **HMEQ.TEST** and then create a custom variable that does not exist in the table.

1. On the Variables tab, select **Add Variable** \Rightarrow **Data source**. The Choose a Data Source window appears, and the list of CAS tables that are loaded into memory is displayed on the Available tab.
2. Enter **HMEQ** in the search field to subset the list of available tables.

If the **HMEQ_TEST** table does not appear in the list of available tables, complete these steps:

- a. On the Data Sources tab, click  beside **cas-shared-default**, and then click  beside **Public**.
- b. Right-click the **HMEQ_TEST** table and select **Load**. After the table is loaded into memory, it should appear on the Available tab.
- c. Click the **Available** tab.

If the **HMEQ_TEST** table does not appear in the list of available tables, click .

3. Select the **HMEQ_TEST** table, and click **OK**. The Add Variables window appears.
4. Click  to add all the variables in the table to the rule set, and then click **OK**. All the variables in the **HMEQ_TEST** table are added to the rule set as both input and output variables. Input variables are variables that appear in the input data. Output variables are variables that appear in the output of the rule set.
5. Select **Add Variable** \Rightarrow **Custom variable**. The Add Variables window appears.
6. Enter **REVIEW** as the variable name, select **Boolean** as the data type, and click **Add**.
7. Clear the check box in the **Input** column.

When the **Input** check box is selected for a variable, the application attempts to map the variable to a column in the input table. The variable **REVIEW** does not appear in the **HMEQ.TEST** input table, so this variable is only an output variable.

In this demo, you create an output-only variable. In other rule sets, you might need a temporary variable that exists only while the rule set is running. To create temporary variables, you clear both the **Input** and **Output** check boxes.

8. Click **OK** to add the output variable **REVIEW** to the rule set.

Variables	Data Type	Input	Output
<u>DELINQ</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>DEROG</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>JOB</u>	Character	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>LOAN</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>MORTDUE</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>NINQ</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>REASON</u>	Character	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>REVIEW</u>	Boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>VALUE</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>YOJ</u>	Decimal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Defining the Business Rule

1. On the Rule Set tab, click **Add Rule**. The application adds a new rule that contains an IF-THEN statement.
2. In the field next to the IF rule operator, select the variable **VALUE**.
3. Select **>=** as the expression operator.
4. Enter **120000** in the expression field.

If **VALUE** **>=** **120000**

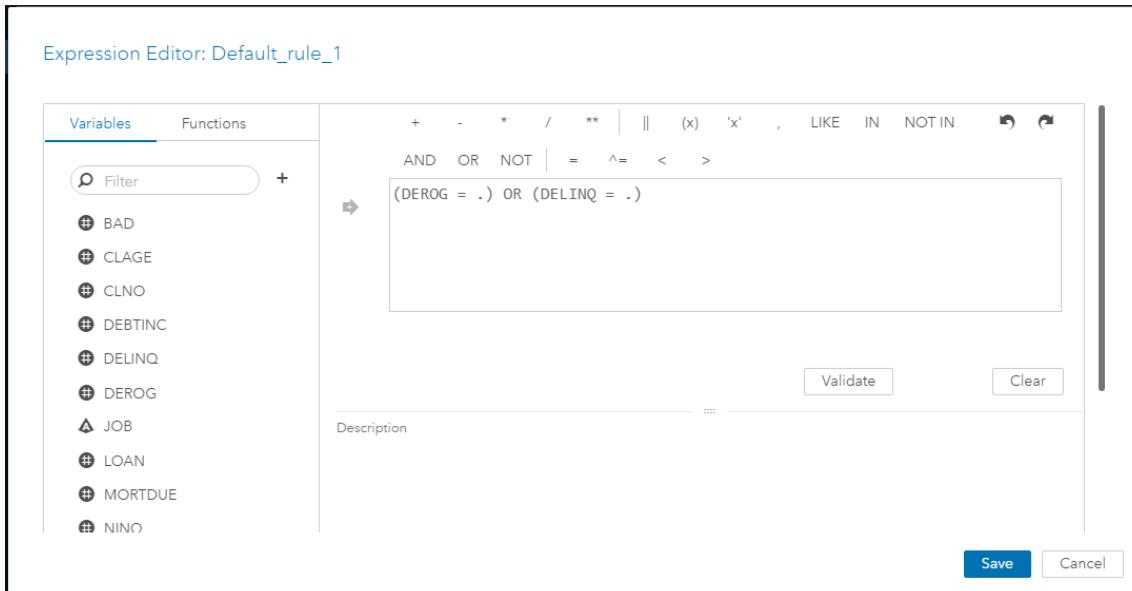
Then **Assign** **BAD**

TIP By default, the **Record rule-fired data** check box is selected on all new rules. If a rule's conditions evaluate to True, SAS Decision Manager executes the rule's actions, and the rule is said to have *fired*.

If the **Record rule-fired data** check box is selected for that rule, then a rule-fired record is generated. When you test a rule set, you can analyze the rule-fired data by running a rule-fired analysis. For this demo, the **Record rule-fired data** check boxes remain selected. When you test the final decision, you run a rule-fired analysis test.

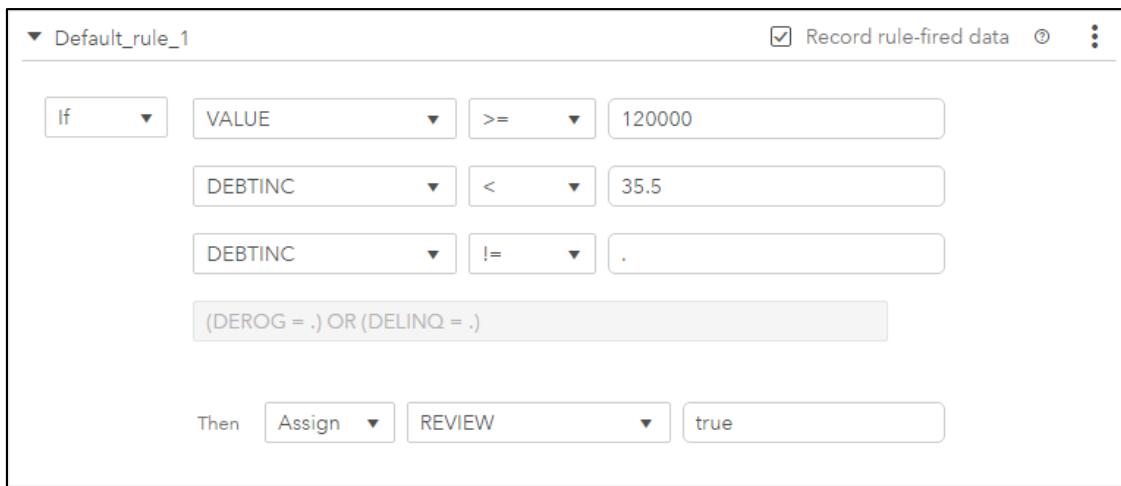
5. Select **Add \Rightarrow Condition**.
6. For the second condition, select **DEBTINC** as the variable, select $<$ as the expression operator, and enter **35.5** in the expression field.
7. Select **Add \Rightarrow Condition**.
8. Select **DEBTINC** as the variable, select **\neq** (not equal to) as the expression operator, and enter **.** (a period) in the expression field. The period denotes a missing numeric value.
9. Select **Add \Rightarrow Condition**.
10. To enter this condition expression, you must use the expression editor. Click  to open the expression editor.
11. Enter **(DEROG = .) OR (DELINQ = .)** in the expression field, and click **Save**.

Alternatively, you can construct the expression by selecting variable names on the Variables tab and selecting operators in the rows above the expression field.



12. To construct the THEN clause, select **Assign** as the action, select **REVIEW** as the variable, and enter **true** in the expression field.

13. Click  to save the rule set.



The screenshot shows the 'Default_rule_1' rule set in SAS Decision Manager. The rule conditions are defined as follows:

- If **VALUE** \geq **120000**
- DEBTINC** $<$ **35.5**
- DEBTINC** \neq **.**
- (DEROG = .)** OR **(DELINQ = .)**

The action section shows:

- Then**
- Assign** \rightarrow **REVIEW**
- true**

A checkbox for 'Record rule-fired data' is checked, and a three-dot menu icon is visible in the top right corner.

Testing the Rule Set

Testing a rule set is optional, but it enables you to discover any problems with a rule set before you incorporate it into a decision.

1. On the Tests tab, click **New Test**. The New Test window appears.
2. (Optional) Enter a name for the test if you do not want to use the default name.
3. Click , select the **HMEQ_TEST** table, and click **OK**.

SAS Decision Manager automatically maps input variables in the rule set to 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, an error message is displayed, and you can map the variables manually. You can map variables and view variable mappings by selecting **Variables** in the New Test window.

4. Click **Run** to run the test. If the test runs successfully, SAS Decision Manager displays the check mark  icon in the Status column.
5. Click  under Results.
6. Click **Rule-Fired Analysis** \Rightarrow **Run Rule-Fired Analysis**.

7. Click **Analysis**. Select a row in the top table and see the corresponding details of the rule that fired in the bottom table.

The screenshot shows the SAS Decision Manager interface. The left sidebar has sections for Test Results (Output, Code, Log) and Rule-Fired Analysis (Analysis, Plot, Log). The main area is titled 'Analysis' and shows 'Output Records'. A table titled 'Output Records' displays data with columns: Rules Fired Count, REVIEW, BAD, and C. The last row shows values: 1, 1, 0, and 181.0707. Below this is a section titled 'Rules Fired for the Selected Output Record' with a table showing Rule Set Name (Loans_to_Review), Rule Name (Default_rule_1), Rule Order (1), and Rule Logic (IF { VALUE >= 120000 } AND { DEBTINC < 35.5 } AND { DEBTINC ^= . } AND { (DEROG = .) OR (DELINQ = .) } THEN REVIEW = true).

Rules Fired Count	REVIEW	BAD	C
1	1	0	181.0707
1	1	0	170.7005
1	1	0	170.945
1	1	0	185.5726

Rule Set Name	Rule Name	Rule Order	Rule Logic
Loans_to_Review	Default_rule_1	1	IF { VALUE >= 120000 } AND { DEBTINC < 35.5 } AND { DEBTINC ^= . } AND { (DEROG = .) OR (DELINQ = .) } THEN REVIEW = true

8. Click **Close** to close the rule set.

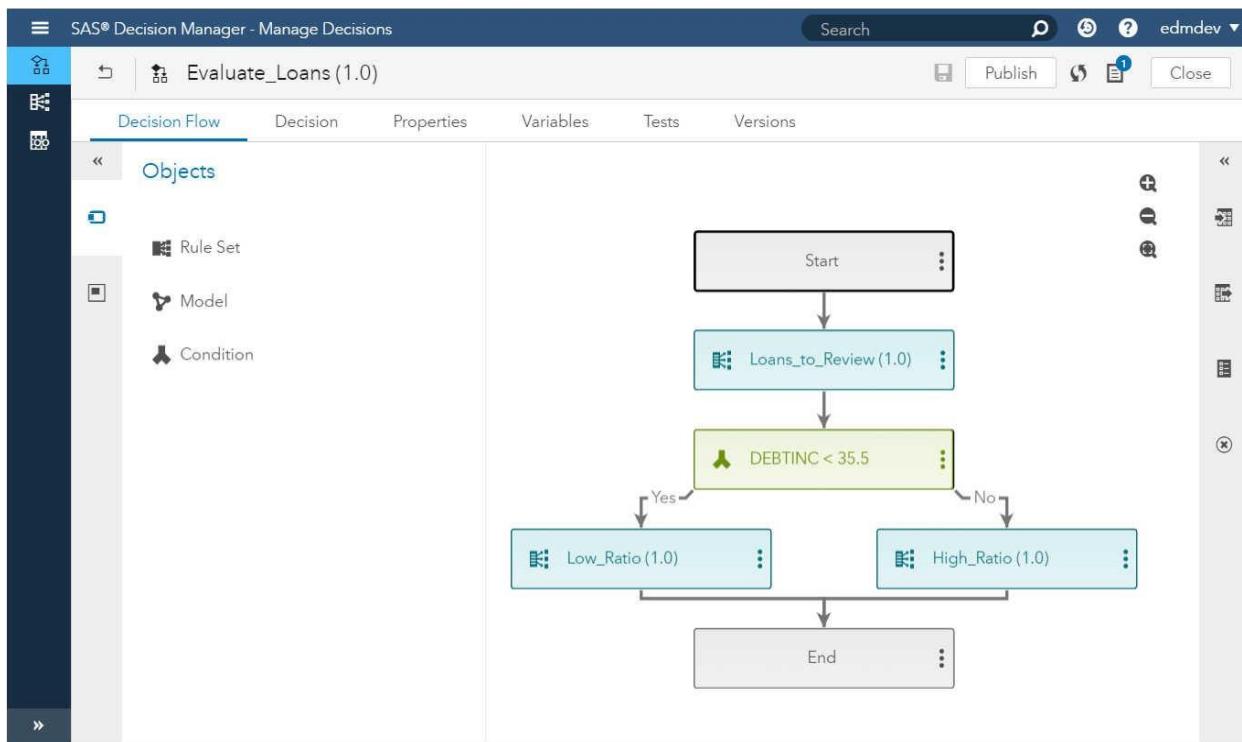
End of Demonstration



Exercises

Before starting this exercise, build the **Loans_to_Review** rule set from the previous demo. (Option: Test the rule set.)

In this exercise, create and test two new rule sets: **Low_Ratio** and **High_Ratio**.



1. Creating the Low_Ratio Rule Set Containing Two Rules

The **Low_Ratio** rule set evaluates loans for which the applicant's debt-to-income ratio is less than 35.5% of their total income.

This rule set defines two rules.

The **first rule** approves a loan application if all of the following conditions are true:

- The current value of the applicant's property is greater than or equal to \$120,000 (**VALUE >= 120000**).
- The reason for the loan is either debt consolidation or home improvement (**REASON IN ('DebtCon','HomeImp')**).
- The applicant's current number of credit lines is less than 15 (**CLNO < 15**).

The **second rule** in the **Low_Ratio** rule set rejects a loan application if all the following conditions are true:

- The current value of the applicant's property is less than \$120,000 (**VALUE < 120000**).
- The number of derogatory reports for the applicant is greater than 2 (**DEROG > 2**).
- The number of delinquent credit lines for the applicant is greater than or equal to 2 (**DELINQ >= 2**).

- Create a new rule set.
 - Name the new rule set **Low_Ratio**.
 - Put it in the same metadata folder where you saved the **Loans_to_Review** rule set.
- Add variables to the rule set.

Note: For this rule set, you can import the variables from the **Loans_to_Review** rule set and define a custom Boolean variable named **REJECT**.

- Define the **Approve** rule. (The finished rule is shown below.)

The screenshot shows the 'Approve' rule configuration in SAS Decision Manager. The rule consists of two conditions and one action:

- If** **VALUE** **>=** **120000**
- REASON** **IN** **'HomeImp','DebtCon'**
- CLNO** **<** **15**
- Then** **Assign** **REJECT** **false**

Note: The values in the IN clause of the second condition are case sensitive and must be entered as shown: **'HomeImp','DebtCon'**. SAS Decision Manager automatically adds parentheses around the set of values that you enter.

- Click **:** and select **Rename rule**.

TIP Assigning logical names to the rules makes it easier to determine which rules fired when you review rule-fired data.

- Enter **Approve** as the rule name, and click **Rename**.
- Save the rule set but do not close it.

- Define the **Reject** rule. (The finished rule is shown below.)

The screenshot shows the 'Reject' rule configuration in SAS Decision Manager. The rule consists of three conditions and one action:

- If** **VALUE** **<** **120000**
- DEROG** **>** **2**
- DELINQ** **>=** **2**
- Then** **Assign** **REJECT** **true**

- Click **:** for this rule and select **Rename rule**.
- Enter **Reject** as the rule name, and click **Rename**.
- Click **□** to save the rule set.

- e. (Optional) Test the rule set. Follow the instructions for testing a rule set in “[Create the Loans_to_Review Rule Set](#).”
 - f. Click **Close** to close the rule set.
- 2. Creating the High_Ratio Rule Set**

The **High_Ratio** rule set evaluates loans for which the applicant’s debt-to-income ratio is greater than 35.5% of the applicant’s total income. This rule set defines one rule with both IF and ELSE conditions.

This rule **approves** a loan application if all the following conditions are true:

- The applicant has no more than one delinquent credit line (**DELINQ <= 1**).
- The data for the number of delinquent credit lines is not missing from the application (**DELINQ != .**).
- The applicant’s oldest credit line is at least 290 months old (**CLAGE >= 290**).

The ELSE part of the rule **rejects** a loan application if both of the following conditions are true:

- The applicant has more than one delinquent line of credit (**DELINQ > 1**).
 - The applicant’s oldest credit line is less than 178 months old (**CLAGE < 178**).
- a. Create a new rule set. Enter **High_Ratio** for the rule set name.
 - b. Add variables to the rule set. For this rule set, import the variables from the **Low_Ratio** rule set.
 - c. Define the **Approve** rule so that the rule set contains the following conditions and action:

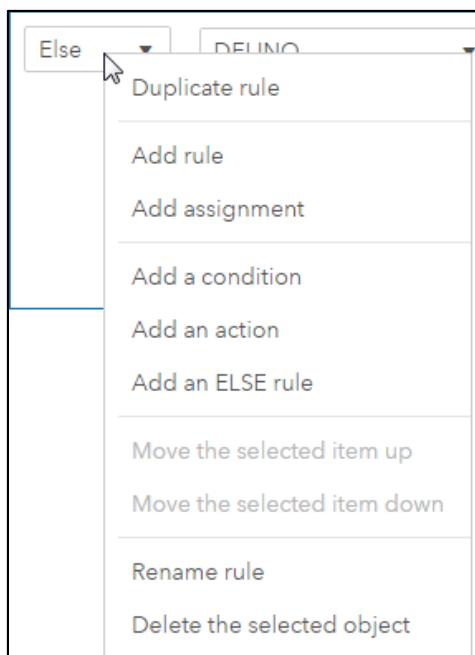
▼ Approve			
If	DELINQ	<=	1
	DELINQ	!=	.
	CLAGE	>=	290
Then	Assign	REJECT	false

- 1) Click to save the rule set.

- d. Define the **Reject** rule using the Else construct below the **Approve** rule.

The screenshot shows the 'Approve' rule configuration in the SAS Decision Manager. The rule consists of two main sections: 'If' and 'Else'. The 'If' section contains three conditions: 'DELINQ <= 1', 'DELINQ != .', and 'CLAGE >= 290'. The 'Then' part of the 'If' section is an 'Assign' action with 'REJECT' assigned to 'false'. The 'Else' section contains two conditions: 'DELINQ > 1' and 'CLAGE < 178'. The 'Then' part of the 'Else' section is an 'Assign' action with 'REJECT' assigned to 'true'.

- 1) With the **Then** line of the **Approve** rule selected, click **Add** and select the **Else** rule.
- 2) Click **Add** and select **Condition**.
- 3) Build the rest of the rule as shown above.
- 4) Right-click the **Else** rule operator and select **Rename rule**.



- 5) Enter **Reject** as the rule name, and click **Rename**.

TIP Rule names for ELSE clauses do not appear in the rule set editor.

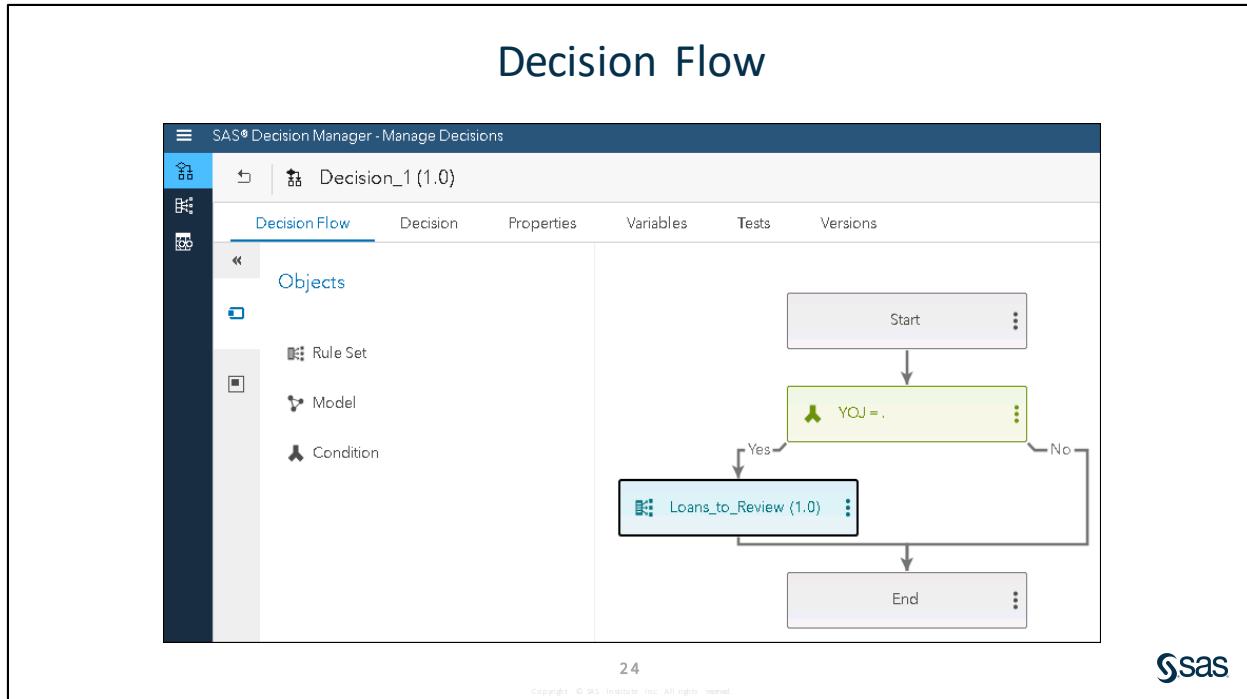
- 6) Click  to save the rule set.
- e. (Optional) Test the rule set. Follow the instructions for testing a rule set in “[Create the Loans_to_Review Rule Set](#).”
- f. Click **Close** to close the rule set.

End of Exercises

3.4 Creating and Testing Decisions

Objectives

- Use rule sets and conditions in a decision.
- Test the decision.
- Practice building your own decision in the exercise.



A decision flow can contain rule sets, models, and condition objects. The decision flow shown in the slide above starts with a condition that checks for missing values in the Years on Job variable (**YOJ**). The left branch of the condition (the Yes branch) then runs the **Loans to review** rule set against the records satisfying the condition.

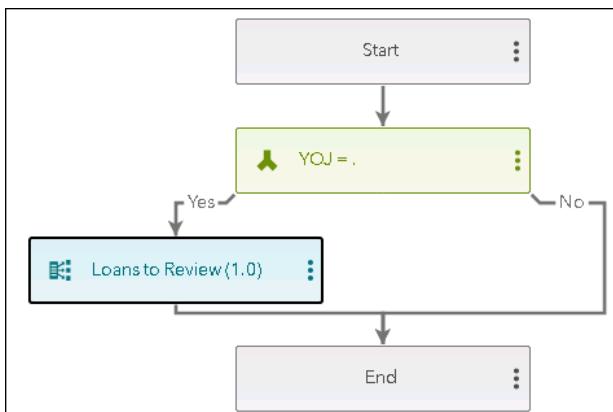
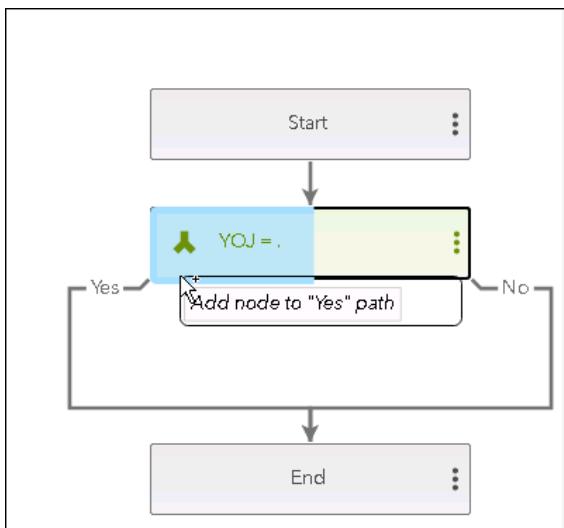
You can include variables from a data source, rule set, another decision, or as a custom variable.

You can run a test to evaluate the rule-fired analysis for the flow.



Create and Test a Decision

1. Select **Decisions** \Rightarrow **New Decision**. Enter **MISSING_YOJ** as the name and save.
2. Click the **Variables** tab. Click **Add Variable** \Rightarrow **Rule set**.
3. Select **Loans_to_Review**, select all variables, and click **OK**.
4. Click the **Decision Flow** tab.
5. Right-click the **Start** object and select **Add Condition**.
6. In the properties window, select the variable **YOJ**, select the **=** operator, and set the value as **''**.
7. Drag the **Rule Set** object onto the Yes side of the condition. Select the **Loans_to_Review** rule set and save the decision .



8. Click the **Decision** tab for an alternate view of the decision.

If YOJ = Value . Then
Loans_to_Review (1.0)

Else
No content

9. Click **Test** \Rightarrow **New test**, add the **HMEQ_TEST** data source, and click **Run**.
10. When the test is finished running, open the results. Click **Rule_Fired Analysis** \Rightarrow **Run Rule_Fired Analysis**.
11. Click **Analysis** and select an output record in the top table to view the rule logic in the bottom table.

Rules Fired Count	REVIEW	BAD
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0

Rule Set Name	Rule Name	Rule Order	Rule Logic
Loans_to_Review	Default_rule_1	1	IF (VALUE >= 120000) AND (DEBTINC < 35.5) AND (DEBTINC ^= .) AND (DEROG = .) OR (DELINQ = .) THEN REVIEW = true

12. The plot shows that the decision fired for eight records.

13. Close the test.

End of Demonstration

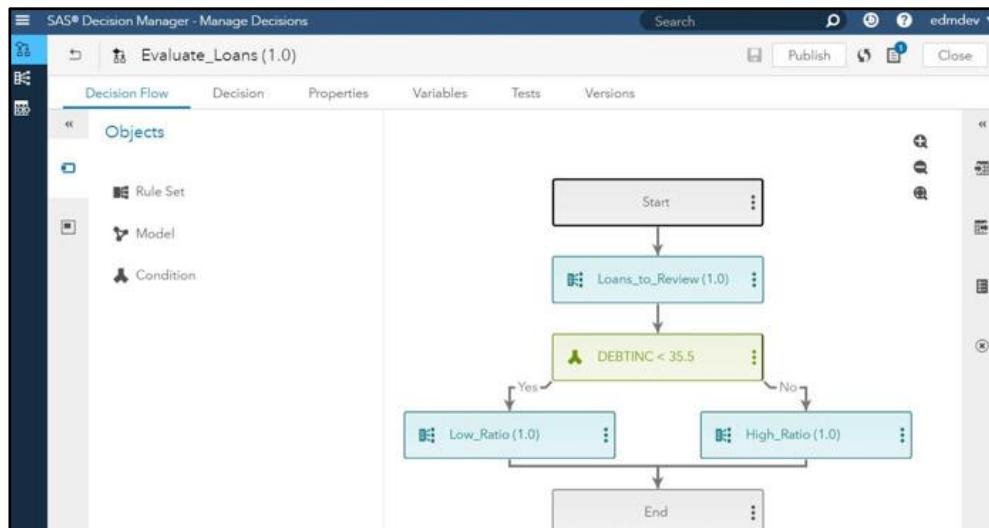
Exercise Setup: Create a Decision

The **Evaluate_Loans** decision combines these three rule sets with conditional logic to evaluate loan applications:

- **Loans_to_Review**
- **Low_Ratio**
- **High_Ratio**

The conditional logic determines which rule set is run based on the value of the **DEBTINC** variable.

Exercise: Result





Exercises

3. Creating the Evaluate_Loans Decisions

- a. Create a new decision.
 - 1) Click  on the navigation bar.
 - 2) Click **New Decision**.
 - 3) Enter **Evaluate_Loans** as the decision name.
 - 4) Click  and select the same folder where you saved the **Low_Ratio** rule set.
 - 5) Click **Save**. The new decision opens in the decision flow editor. A Start node and an End node are automatically added to decision.

TIP There are two views that you can use to edit decisions. The **Decision Flow** tab enables you to edit a decision using a graphical editor. The **Decision** tab enables you to edit the decision using a tabular view similar to the rule set editor. This tutorial uses the graphical view on the Decision Flow tab.

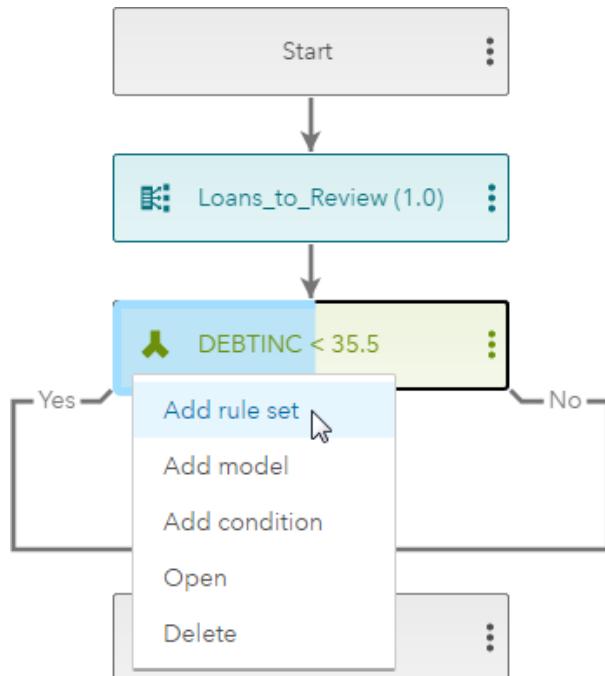
- b. Add the **Loans_to_Review** rule set.
 - 1) Drag the **Rule Set** object from the list of objects onto the Start node in the diagram. The Select a Rule Set window appears.
 - 2) Navigate to the location where you saved the rule sets, select the **Loans_to_Review** rule set, and click **OK**.
- c. Add a condition.

In decision conditions, you can compare the value of a variable to a literal value or to the value of another variable. The **comparison mode** specifies which method to use. In this exercise, we compare the value of **DEBTINC** to the literal value 35.5.

- 1) Drag the **Condition** object from the list of objects onto the **Loans_to_Review** rule set in the diagram. The condition is added to the decision flow below the **Loans_to_Review** rule set, and the Properties pane for the condition appears.
- 2) In the Properties pane, select **DEBTINC** for the variable.
- 3) Select **<** as the expression operator.
- 4) Select **Value** for the comparison mode, and enter **35.5** for the value.
- 5) Click  to save the decision.

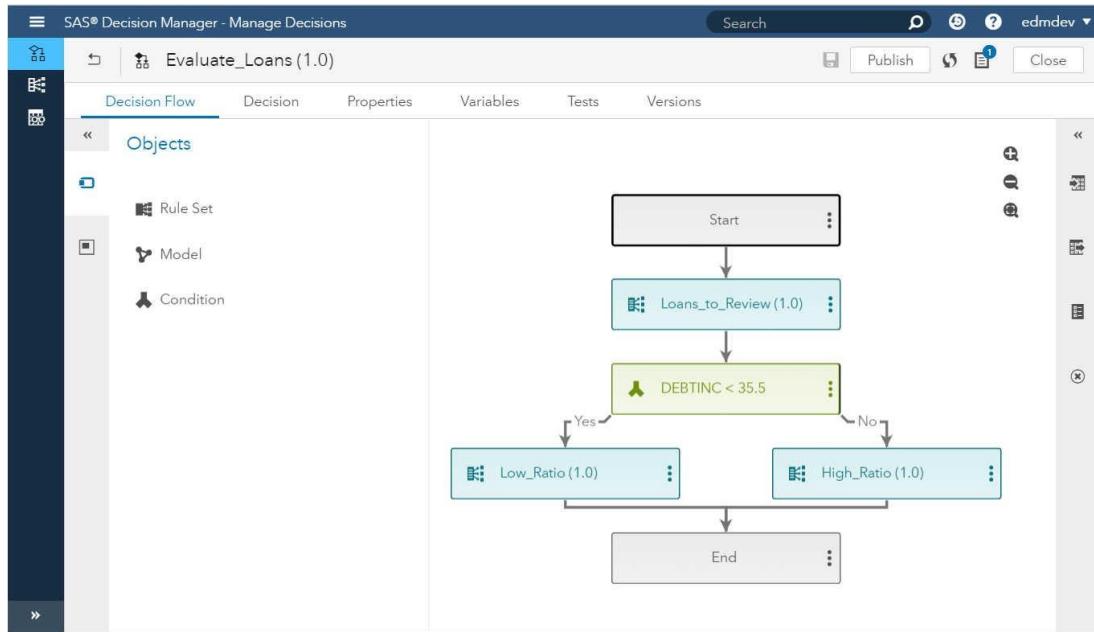
d. Add the **Low_Ratio** and **High_Ratio** rule sets.

- 1) Right-click the Yes side of the condition node, and select **Add rule set**. The Select a Rule Set window appears.



- 2) Select the **Low_Ratio** rule set, and click **OK**.
- 3) Right-click the No side of the condition node, and select **Add rule set**. The Select a Rule Set window appears.
- 4) Navigate to the location where you saved the rule sets, select the **High_Ratio** rule set, and click **OK**.

- 5) Click  to save the decision.



4. Testing and Publishing the Decision

- a. Create and run a new test.

Testing a decision is optional but highly recommended because it enables you to discover any problems with a decision before it is published and incorporated into a production system.

- 1) On the Tests tab, click **New Test**. The New Test window appears.
- 2) (Optional) Enter a name for the test if you do not want to use the default name.
- 3) Click , select the **HMEQ_TEST** table, and click **OK**. SAS Decision Manager automatically maps input variables in the decision to 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, an error message is displayed, and you can map the variables manually. You can map variables and view variable mappings by selecting Variables in the New Test window.
- 4) Click **Run** to run the test. If the test runs successfully, SAS Decision Manager displays the check mark  icon in the **Status** column.
- 5) 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 Decision Manager, the output data set, and the SAS log that was generated when the code was run.

- b. Run a rule-fired analysis.

If a rule's conditions evaluate to True, then the rule is said to have *fired*. Rule-fired data includes summary information about how many times each rule fired and detailed information for each time that a rule evaluates to True. See “[How Rules Are Evaluated and When Rule-Fired Records Are Generated](#)” in *SAS® Decision Manager: User’s Guide* for more information.

- 1) Click **Rule-Fired Analysis** in the navigation pane.

- 2) Click **Run Rule-Fired Analysis**. SAS Decision Manager analyzes the test results to determine which rules fired for each row in the **HMEQ_TEST** table.
- 3) Click **Analysis** in the navigation pane. The analysis page displays the output table at the top of the page and the rule-fired table at the bottom. If you select a row in the output table, the rules that fired for that output row are displayed in the bottom table.

For example, in this exercise, there is one output record for which two rules fired. The two rules are the rule in the **Loans_to_Review** rule set and Rule 2 in the **Low_Ratio** rule set.

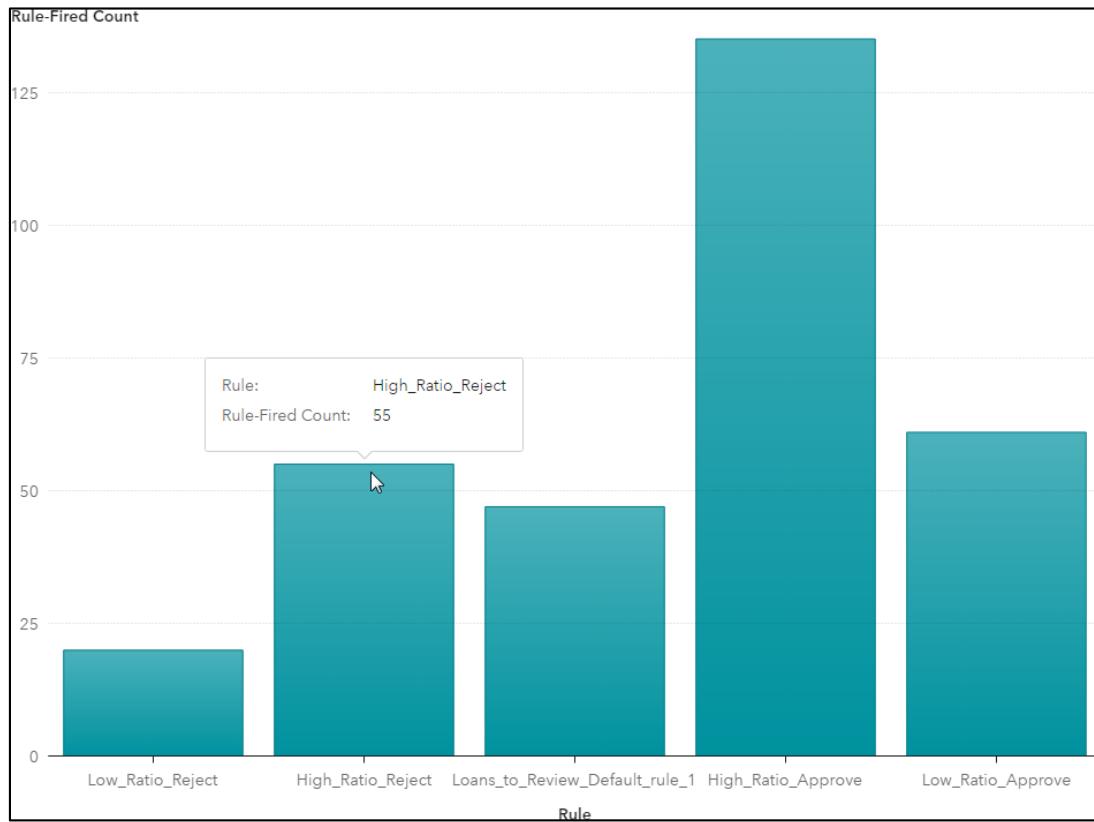
Output Records

Rules Fired Count	REJECT	REVIEW
1	U	
2	0	1
1	0	
...		

Rules Fired for the Selected Output Record

Rule Set Name	Rule Name	Rule Order	Rule Logic
Loans_to_Review	Default_rule_1	1	IF (VALUE >= 120000) AND (DEBTINC < 35.5) AND (DEBTINC ^= .) AND ((DEROG = .) OR (DELINQ = .)) THEN REVIEW = true
Low_Ratio	Approve	2	IF (VALUE >= 120000) AND (REASON in ('HomeImp', 'DebtCon')) AND (CLNO < 15) THEN REJECT = false

- 4) Click **Plot** in the navigation pane. SAS Decision Manager displays a bar chart that shows how many times each rule in the decision fired. Position your mouse pointer over a bar to display the name of the rule and the number of times that the rule fired.

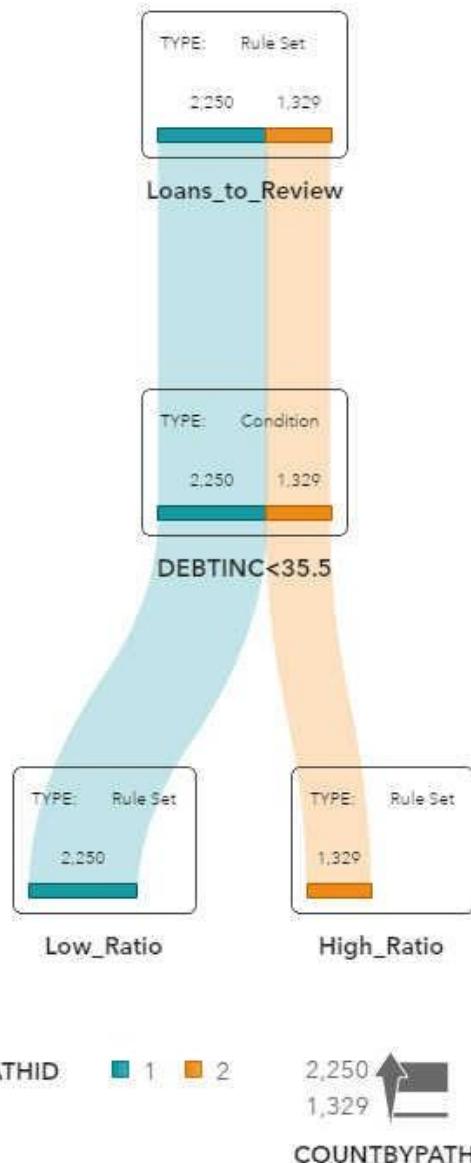


- c. Run a decision path tracking analysis.

Decision path tracking shows you the route that input records take through the rule sets and conditions in your decision.

- 1) Click **Decision Path Tracking** in the navigation pane.
- 2) Click **Run Path Tracking** to run a decision path analysis.

- 3) Click **Analysis** ⇒ **Plot** to display a Sankey diagram that shows the flow of the input records through the rule sets and conditions in the decision. The numbers in the diagram are the number of rows in the input table that followed each path.



- 4) Click **Node Count** in the navigation pane to display a table showing the number of input records evaluated at each node in the decision.
- 5) Click **Close** to close the Test Results window.

d. Publish the **Evaluate_Loans** decision.

Publishing content makes it available to other applications. In this exercise, you publish only the decision, but you could also publish the rule sets. Publishing a decision or a rule set creates an entity that can be managed and run in another environment. For example, if you publish content to the SAS Micro Analytic Service destination, SAS Decision Manager creates a DS2 package that can be managed and run through the SAS Micro Analytic Service interface.

- 1) Click **Publish**. The Publish Decision window appears.

- 2) Select the destination where you want to publish the decision, and click **Publish**. The Publishing Results window displays the name of the published decision, its status, and information about any issues encountered while publishing the decision.
- 3) Click **Close** to close the Publishing_Results window.
- 4) Click **Close** to close the decision.

End of Exercises

3.5 Lookup Tables (Self-Study)

Objectives

- State the purpose of lookup tables.
- Create a lookup table.
- Create a business rule that uses a lookup table.

31



Lookup Table

Lookup tables are tables of key-value pairs. You can use them to search for a key and retrieve its associated value. For example, you can retrieve a part name based on its part number.

The benefits of using lookup tables:

- update rule outcomes as business conditions change
- update the values that the rule/logic uses just by updating the lookup table

32



Lookup tables are not designed to account for missing values.

Lookup Table

Lookup tables are usually imported from a CSV file with two columns. The first column is the key (an expected value from your data table). The second column is the value that can be retrieved into an output variable in your business rule.

	A	B
1	ProfExe	Executive
2	Other	Services
3	Sales	Sales
4	Mgr	Manager
5	Office	Administrative
6	Self	Self Employed

33



Copyright © SAS Institute Inc. All rights reserved.

Lookup Table

From the Lookup Table navigation bar, you create a table by either importing a CSV file or using the + sign to build the table one row at a time.

Key	Value
	Unspecified
Mgr	Manager
Office	Administrative
Other	Services
ProfExe	Executive
Sales	Sales
Self	Self Employed

34



Copyright © SAS Institute Inc. All rights reserved.

Entries in a lookup table must be 100 characters or less. A single lookup table should contain fewer than 5,000 entries for best performance. The Export button creates a saved copy of your lookup table as a CSV file.

continued...

Lookup and LookupValue Functions

There are two functions that you can use with lookup tables:

LOOKUP

enables you to verify that a key is present in a lookup table. This function is valid only in condition expressions.

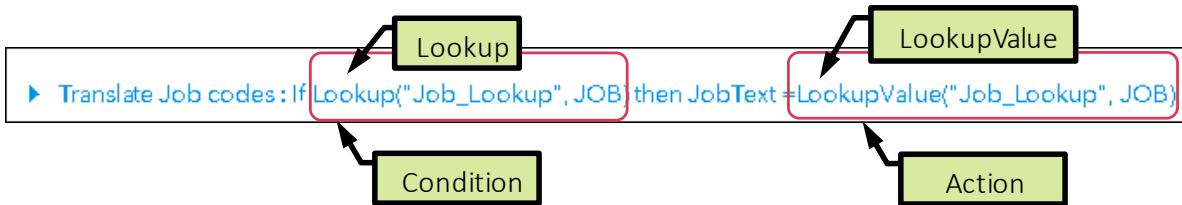
LOOKUPVALUE

enables you to retrieve a value that is associated with a key. This function is valid only in action expressions.

Lookup and LookupValue Functions

For example, the “Translate Job codes” rule set uses a lookup table in two places.

The rule **if** condition has the **Lookup(“Job_Lookup”,JOB)** function to read the value of job from the data set. If it finds a match in the table, the **then** action applies the **LookupValue (“Job_Lookup”,JOB’)** function to retrieve the expanded value for Job from the lookup table into the **JobText** variable.



Lookup and LookupValue

This window shows how a rule set appears in the application.

The screenshot shows the SAS Decision Manager interface. A rule set named 'Looks (1.0)' is open. The 'Rule Set' tab is selected. Inside, a rule named 'Default_rule_1' is expanded. The 'If' section contains a 'Lookup' node, which is highlighted with a green box and an arrow. The 'Then' section contains an 'Assign' dropdown set to 'JobText', followed by a 'LookupValue' node, also highlighted with a green box and an arrow. Below the 'Then' section is a 'Lookup expression editor/selector' icon, which is highlighted with a green box and an arrow. The bottom right corner of the interface has the SAS logo.

When the If or Then row is selected, the Lookup expression editor icon appears on the row. Use this icon to select an item from your list of lookup tables and to build the expression.

Expression Editor

The screenshot shows the 'Lookup Expression Editor' window titled 'Translate Job codes'. On the left, there are two tabs: 'Lookup Tables' (selected) and 'Variables'. Under 'Lookup Tables', there is a 'Filter' input field and two items listed: 'Job_Lookup' and 'Lookup_table_1'. To the right, a large text area displays the expression 'LookupValue("Job_Lookup", JOB)'. At the bottom of the editor are 'Validate' and 'Reset' buttons. The bottom right corner of the window has the SAS logo.

The expression editor lets you choose a lookup table, and lets you choose a variable for the lookup expression. This screen uses the `LookupValue` function because it was opened from the “action” portion of the rule set. The Variables link lists names of variables that can be added into the expression with a double-click. The Validate button at the bottom checks the syntax of your expression.

Test Result

Test results indicate that the rule successfully used the lookup table to translate the values in the **JOB** variable into the **JobText** variable.

JOB	JobText	Rules Fired Count
Other	Services	1
Sales	Sales	1
Office	Administrative	1
Mgr	Manager	1
Other	Services	1

Sas

Lookup tables can be used in any rule. If a lookup table is updated, the rule uses the new lookup values the next time that it is run. To update a lookup table, create a new version, update the contents, and activate the new table version. A lookup table must be activated before it can be used by running processes.

⚠ Activate

The active version is the version that is used when the lookup table is referenced by running processes.
Activating a version locks it.

Do you want to activate version 1.1?



Use a Lookup Table

Create a lookup table from a CSV file and retrieve values from the table in a simple business rule.

1. Create a lookup table. Click **Lookup Tables** \Rightarrow **New Lookup Table** in SAS Decision Manager.
2. Delete the row containing a blank key and the value *Unspecified*. Lookup tables cannot be used to check for missing values.
3. Name the table **Job_Lookup** and click **Save**.
4. Click **Import**. Select **D:\workshop\winsas\Viya_DM\Lookup.csv**. Click **Import**.
5. Click **Activate** \Rightarrow **Yes**.
6. Create a rule set. Click **Rule sets** \Rightarrow **New Rule Set**.
7. Enter **Job_Lookup_Rule** as the name and click **Save**.
8. Click **Variables** \Rightarrow **Add Variable** \Rightarrow **Data source**. Click **Hmeq_test** \Rightarrow **OK**. Select all the variables and click **OK**.
9. Click **Add Variable** \Rightarrow **Custom variable**. Enter **JobText** as the name and character as the data type and click **Add**. Clear the **Input** check box, click **OK**, and save the rule set.
10. Click the **Rule Set** tab. Click **Add Rule**. Click the data entry area to the right of the IF expression and click the **Lookup Expression Editor** icon.

11. Add **Lookup("Job_Lookup",JOB)** as the expression. Validate and save the rule set.
12. Click the action section of the rule and select the **JobText** variable for Assign. Click the **Lookup Expression Editor** icon.

13. Enter **LookupValue("Job_Lookup",JOB)** as the expression. Validate and save the rule set. Save the rule.
14. Test the rule using the **Hmeq_test** table. Click **Tests** \Rightarrow **New Test** \Rightarrow **Data Source** \Rightarrow **HMEQ_TEST** \Rightarrow **OK** \Rightarrow **Run**.
15. Review the test results.

End of Demonstration

3.6 Solutions

Solutions to Exercises

1. Creating the Low_Ratio Rule Set Containing Two Rules

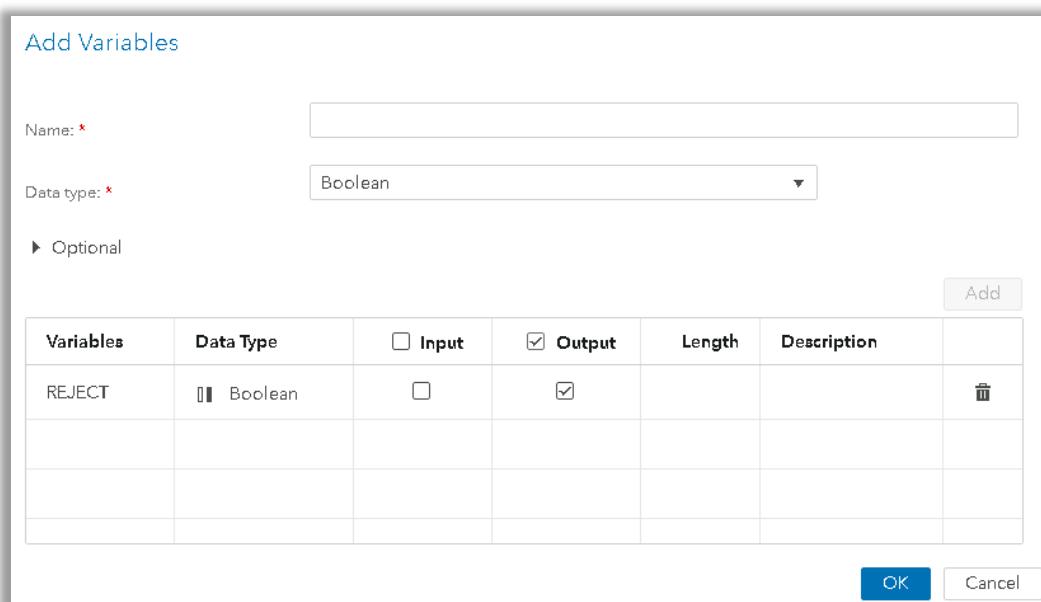
a. Create a new rule set.

- 1) Click  on the navigation bar.
- 2) Click **New Rule Set**. The New Rule Set window appears.
- 3) Enter **Low_Ratio** for the rule set name.
- 4) Click , and select the same folder where you saved the **Loans_to_Review** rule set.
- 5) Click **Save**. The application opens the new rule set and displays the Rule Set tab.

b. Add variables to the rule set.

For this rule set, import the variables from the **Loans_to_Review** rule set and define a custom variable named **REJECT**.

- 1) Click the **Variables** tab and select **Add Variable** \Rightarrow **Rule Set**. The Add Variables window appears.
- 2) Select the **Loans_to_Review** rule set.
- 3) Click any variable name to enable the  button.
- 4) Click , and then click **OK**.
- 5) Select **Add Variable** \Rightarrow **Custom variable**. The Add Variables window appears.
- 6) Enter **REJECT** as the name, select **Boolean** as the data type, and click **Add**.
- 7) Clear the check box in the **Input** column.



Variables	Data Type	<input type="checkbox"/> Input	<input checked="" type="checkbox"/> Output	Length	Description	
REJECT	Boolean	<input type="checkbox"/>	<input checked="" type="checkbox"/>			

- 8) Click **OK** to close the Add Variables window.

c. Define the **Approve** rule.

- 1) On the Rule Set tab, click **Add Rule**. The application adds a new rule that contains an IF-THEN statement.
 - 2) To construct the IF condition, select **VALUE** as the variable, select **>=** as the operator, and enter **120000** in the expression field.
 - 3) Select **Add \Rightarrow Condition**.
 - 4) To construct the second condition, select **REASON** as the variable, select **IN** as the operator, and enter '**HomeImp','DebtCon**' in the expression field. When you use the IN operator, SAS Decision Manager automatically adds parentheses around the set of values that you enter.
- Note:** The values in the IN clause are case sensitive and must be entered as shown.
- 5) Select **Add \Rightarrow Condition**.
 - 6) Select **CLNO** as the variable, select **<** as the operator, and enter **15** in the expression field.
 - 7) To construct the THEN clause, leave **Assign** as the action, select **REJECT** as the variable, and enter **false** in the expression field.
 - 8) Click **:** and select **Rename rule**.

TIP Assigning logical names to the rules makes it easier to determine which rules fired when you review rule-fired data.

- 9) Enter **Approve** as the rule name, and click **Rename**.

The screenshot shows the 'Approve' rule configuration in the SAS Decision Manager interface. The rule has two conditions defined under the 'If' section:

- Condition 1: Variable **VALUE**, Operator **>=**, Expression **120000**
- Condition 2: Variable **REASON**, Operator **IN**, Expression **'HomeImp','DebtCon'**

The 'Then' section specifies the action as **Assign** and the variable as **REJECT**, with the expression **false**.

- 10) Click to save the rule set.

d. Define the **Reject** rule.

- 1) Select **Add \Rightarrow Add rule**. The application adds a new rule that contains an IF-THEN statement.
- 2) To construct the IF condition, select the variable **VALUE**, select **<** as the operator, and enter **120000** in the expression field.
- 3) Select **Add \Rightarrow Condition**.
- 4) Select **DEROG** as the variable, select **>** as the operator, and enter **2** in the expression field.

- 5) Select **Add** \Rightarrow **Condition**.
- 6) Select **DELINQ** as the variable, select **\geq** as the operator, and enter **2** in the expression field.
- 7) To construct the THEN clause, leave **Assign** as the action, select **REJECT** as the variable, and enter **true** in the expression field.
- 8) Click **:** for this rule and select **Rename rule**.
- 9) Enter **Reject** as the rule name, and click **Rename**.

The screenshot shows the 'Reject' rule configuration in the SAS Decision Manager. The rule consists of three conditions stacked vertically:

- If **VALUE** $<$ **120000**
- DEROG** $>$ **2**
- DELINQ** \geq **2**

Below the conditions, the 'Then' section is set to **Assign** **REJECT** with the value **true**. There is also a checked checkbox for **Record rule-fired data**.

- 10) Click **Save** to save the rule set.
- e. (Optional) Test the rule set. Follow the instructions for testing a rule set in “[Create the Loans_to_Review Rule Set](#).”
 - f. Click **Close** to close the rule set.
- ## 2. Creating the High_Ratio Rule Set
- a. Create a new rule set.
 - 1) Click **Rule Sets** on the navigation bar and click **New** to return to the Rule Sets table.
 - 2) Click **New Rule Set**. The New Rule Set window appears.
 - 3) Enter **High_Ratio** for the rule set name.
 - 4) Click **OK**, and select the same folder where you saved the **Loans_to_Review** rule set.
 - 5) Click **Save**. The application opens the new rule set and displays the Rule Set tab.
 - b. Add variables to the rule set. For this rule set, you import the variables from the **Low_Ratio** rule set.
 - 1) On the Variables tab, select **Add Variable** \Rightarrow **Rule Set**. The Add Variables window appears.
 - 2) Select the **Low_Ratio** rule set.
 - 3) Click any variable name to enable the **Add** button.
 - 4) Click **Add**, and then click **OK**.

c. Define the **Approve** rule.

The screenshot shows a rule editor window with the following configuration:

- If** condition: **DELINQ** \leq **1**
- Then** action: **Assign** **REJECT** **false**
- Else If** condition: **DELINQ** \neq **.**
- Else If** condition: **CLAGE** \geq **290**

- 1) On the Rule Set tab, click **Add Rule**. The application adds a new rule that contains an IF-THEN statement.
- 2) To construct the IF condition, select the variable **DELINQ**, select the \leq operator, and enter **1** in the expression field.
- 3) Select **Add \Rightarrow Condition**.
- 4) Select the variable **DELINQ**, select the \neq (not equal to) operator, and enter **.** (a period) in the expression field. The period denotes a missing numeric value.
- 5) Select **Add \Rightarrow Condition**.
- 6) Select **CLAGE** as the variable, select \geq as the operator, and enter **290** in the expression field.
- 7) To construct the THEN clause, select **Assign** as the action, select **REJECT** as the variable, and enter **false** in the expression field.
- 8) Click **:** and select **Rename rule**.
- 9) Enter **Approve** as the rule name, and click **Rename**.
- 10) Click **□** to save the rule set.

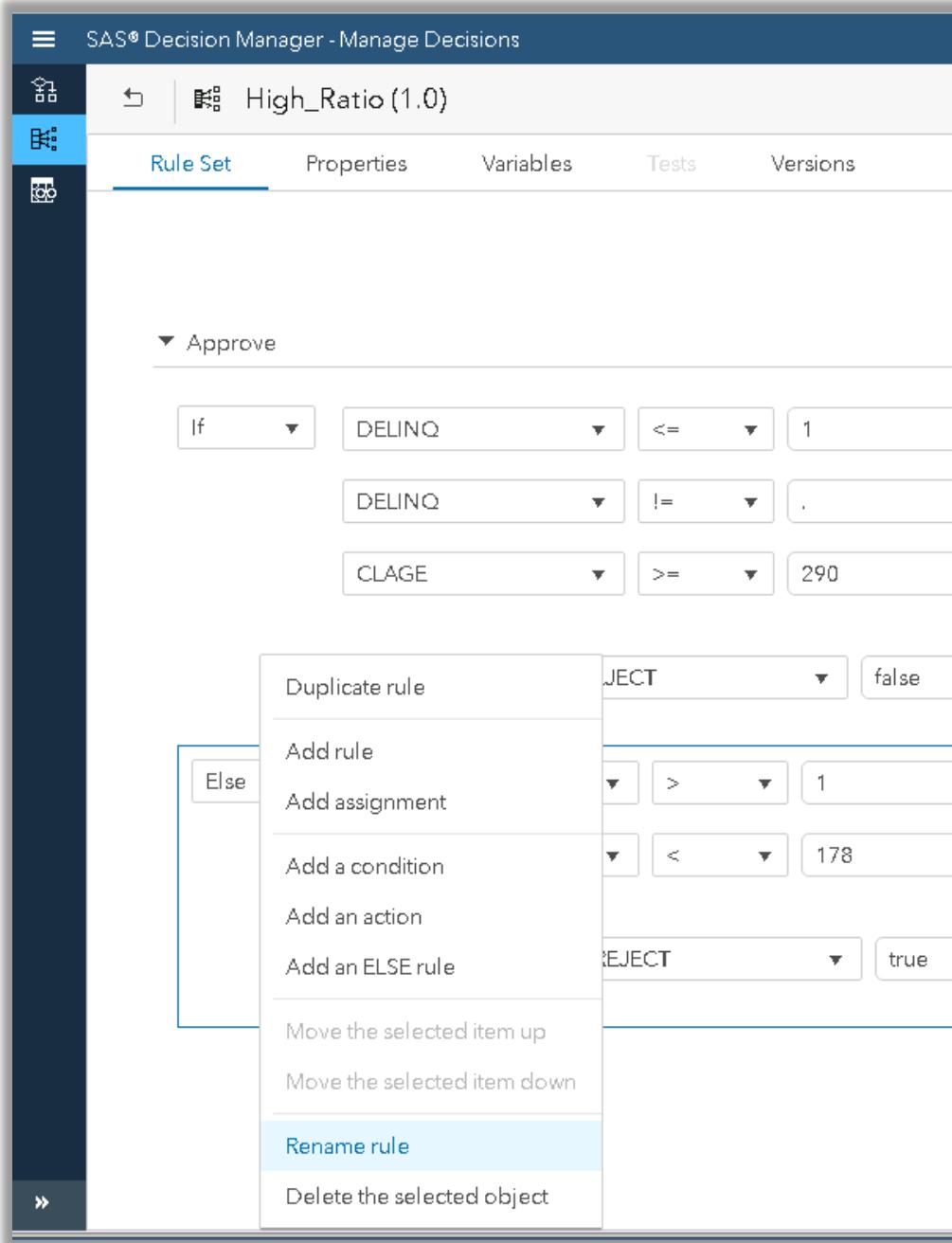
- d. Define the **Reject** rule using the Else construct below the **Approve** rule.

The screenshot shows the 'Approve' rule configuration in the SAS Decision Manager. The rule consists of two main sections: 'If' and 'Else' conditions, each followed by a 'Then' clause.

- If Condition:**
 - Variable: DELINQ, Operator: <=, Value: 1
 - Variable: DELINQ, Operator: !=, Value: .
 - Variable: CLAGE, Operator: >=, Value: 290
- Then Clause:** Action: Assign, Variable: REJECT, Value: false
- Else Condition:**
 - Variable: DELINQ, Operator: >, Value: 1
 - Variable: CLAGE, Operator: <, Value: 178
- Else Then Clause:** Action: Assign, Variable: REJECT, Value: true

- 1) With the Then line of the **Approve** rule selected, click **Add** and select **Else rule**.
- 2) Click **Add** and select **Condition**.
- 3) Select the variable **DELINQ**, select **>** as the operator, and enter **1** in the expression field.
- 4) Select **Add \Rightarrow Condition**.
- 5) Select **CLAGE** as the variable, select **<** as the operator, and enter **178** in the expression field.
- 6) To construct the THEN clause, select **Assign** as the action, select **REJECT** as the variable, and enter **true** in the expression field.

- 7) Right-click the **Else** rule operator and select **Rename rule**.



- 8) Enter **Reject** as the rule name, and click **Rename**.

TIP Rule names for ELSE clauses do not appear in the rule set editor.

- 9) Click  to save the rule set.



The screenshot shows the 'Approve' rule set editor. It contains two main sections of conditional logic:

- If:** DELINQ <= 1 AND DELINQ != . AND CLAGE >= 290. Then Assign REJECT false.
- Else:** DELINQ > 1 AND CLAGE < 178. Then Assign REJECT true.

A checkbox for "Record rule-fired data" is checked at the top right.

- e. (Optional) Test the rule set. Follow the instructions for testing a rule set in “[Create the Loans_to_Review Rule Set](#).”
 - f. Click **Close** to close the rule set.
- 3. Creating the Evaluate_Loans Decision**
- The **Evaluate_Loans** decision combines the **Loans_to_Review**, **Low_Ratio**, and **High_Ratio** rule sets with conditional logic to evaluate loan applications. The conditional logic determines which rule set is run based on the value of the **DEBTINC** variable.
- a. Create a new decision.
 - 1) Click  on the navigation bar.
 - 2) Click **New Decision**. The New Decision window appears.
 - 3) Enter **Evaluate_Loans** as the decision name.
 - 4) Click  and select the same folder where you saved the **Low_Ratio** rule set.
 - 5) Click **Save**. The new decision opens in the decision flow editor. A Start node and an End node are automatically added to decision.

TIP There are two views that you can use to edit decisions. The **Decision Flow** tab enables you to edit a decision using a graphical editor. The **Decision** tab enables you to edit the decision using a tabular view similar to the rule set editor. This tutorial uses the graphical view on the Decision Flow tab.

b. Add the **Loans_to_Review** rule set.

- 1) Drag the **Rule Set** object from the list of objects onto the Start node in the diagram. The Select a Rule Set window appears.
- 2) Navigate to the location where you saved the rule sets, select the **Loans_to_Review** rule set, and click **OK**.

c. Add a condition.

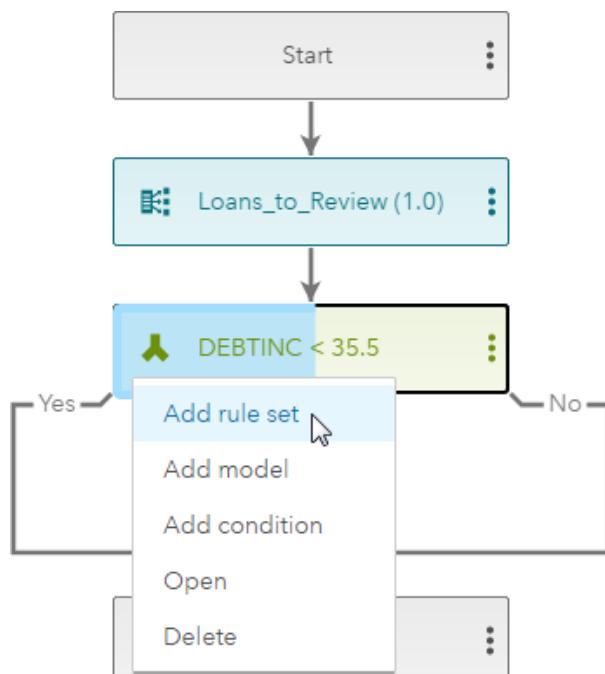
- 1) Drag the **Condition** object from the list of objects onto the **Loans_to_Review** rule set in the diagram. The condition is added to the decision flow below the **Loans_to_Review** rule set, and the Properties pane for the condition opens.
- 2) In the Properties pane, select **DEBTINC** for the variable.
- 3) Select **<** as the expression operator.
- 4) Select **Value** for the comparison mode, and enter **35.5** for the value.

If you wanted to compare **DEBTINC** to another variable, you would select **Variable** as the comparison mode and enter the variable name instead of a literal value.

- 5) Click  to save the decision.

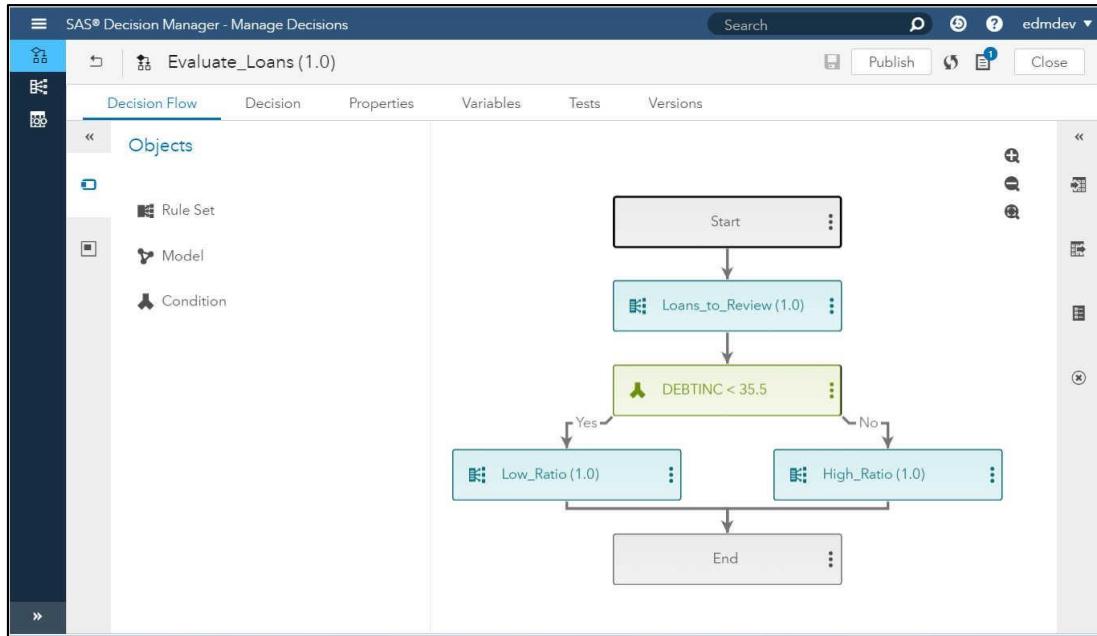
d. Add the **Low_Ratio** and **High_Ratio** rule sets.

- 1) Right-click the **Yes** side of the condition node, and select **Add rule set**. The Select a Rule Set window appears.



- 2) Navigate to the location where you saved the rule sets, select the **Low_Ratio** rule set, and click **OK**.
- 3) Right-click the **No** side of the condition node, and select **Add rule set**. The Select a Rule Set window appears.
- 4) Navigate to the location where you saved the rule sets, select the **High_Ratio** rule set, and click **OK**.

- 5) Click  to save the decision.



4. Testing and Publishing the Decision

- a. Create and run a new test.

Testing a decision is optional, but highly recommended because it enables you to discover any problems with a decision before it is published and incorporated into a production system.

- 1) On the Tests tab, click **New Test**. The New Test window appears.
- 2) (Optional) Enter a name for the test if you do not want to use the default name.
- 3) Click , select the **HMEQ_TEST** table, and click **OK**. SAS Decision Manager automatically maps input variables in the decision to 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, an error message is displayed, and you can map the variables manually. You can map variables and view variable mappings by selecting **Variables** in the New Test window.
- 4) Click **Run** to run the test. If the test runs successfully, SAS Decision Manager displays the check mark  icon in the **Status** column.
- 5) 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 Decision Manager, the output data set, and the SAS log that was generated when the code was run.

- b. Run a rule-fired analysis.

If a rule's conditions evaluate to True, then the rule is said to have *fired*. Rule-fired data includes summary information about how many times each rule fired and detailed information for each time that a rule evaluates to True. See ["How Rules Are Evaluated and When Rule-Fired Records Are Generated"](#) in *SAS Decision Manager: User's Guide* for more information.

- 1) Click **Rule-Fired Analysis** in the navigation pane.

- 2) Click **Run Rule-Fired Analysis**. SAS Decision Manager analyzes the test results to determine which rules fired for each row in the **HMEQ_TEST** table.
- 3) Click **Analysis** in the navigation pane. The analysis page displays the output table at the top of the page and the rule-fired table at the bottom. If you select a row in the output table, the rules that fired for that output row are displayed in the bottom table.

For example, in this exercise, there is one output record for which two rules fired. The two rules are the rule in the **Loans_to_Review** rule set and Rule 2 in the **Low_Ratio** rule set.

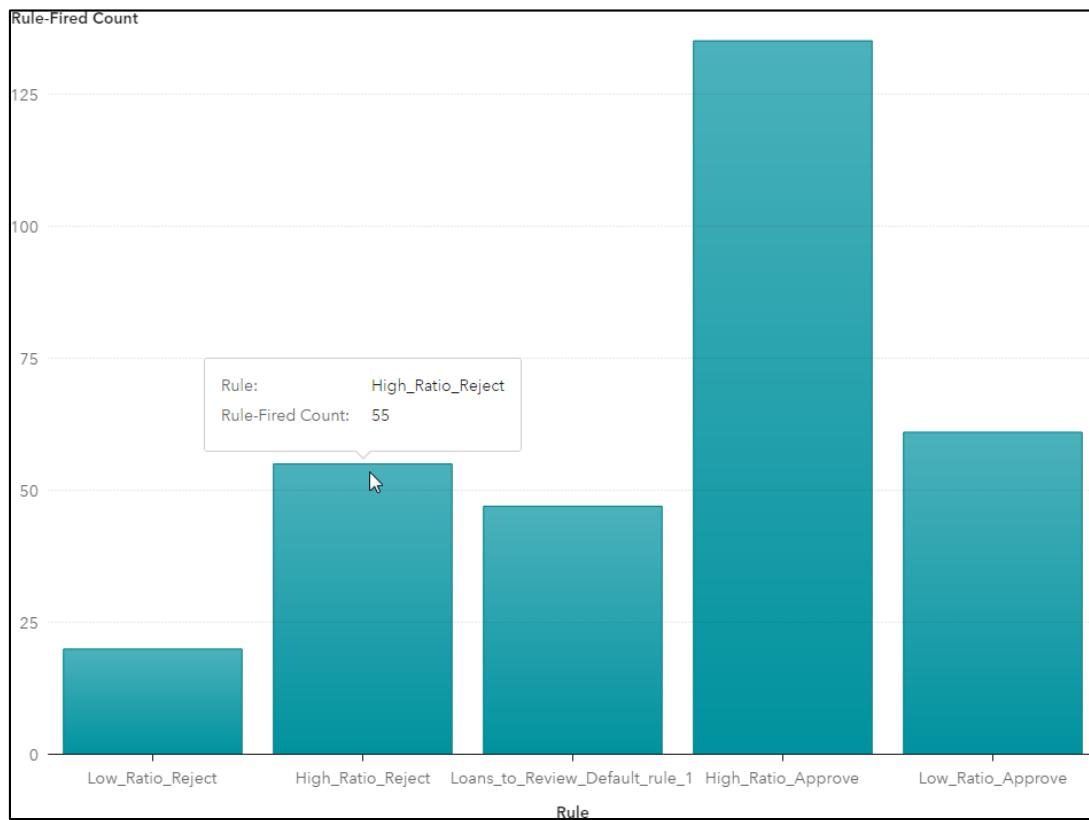
Output Records

Rules Fired Count	REJECT	REVIEW
1	0	
2	0	1
1	0	
...		

Rules Fired for the Selected Output Record

Rule Set Name	Rule Name	Rule Order	Rule Logic
Loans_to_Review	Default_rule_1	1	<pre>IF (VALUE >= 120000) AND (DEBTINC < 35.5) AND (DEBTINC ^= .) AND ((DEROG = .) OR (DELINQ = .)) THEN REVIEW = true</pre>
Low_Ratio	Approve	2	<pre>IF (VALUE >= 120000) AND (REASON in ('HomeImp', 'DebtCon')) AND (CLNO < 15) THEN REJECT = false</pre>

- 4) Click **Plot** in the navigation pane. SAS Decision Manager displays a bar chart that shows how many times each rule in the decision fired. Position your cursor over a bar to display the name of the rule and the number of times that the rule fired.

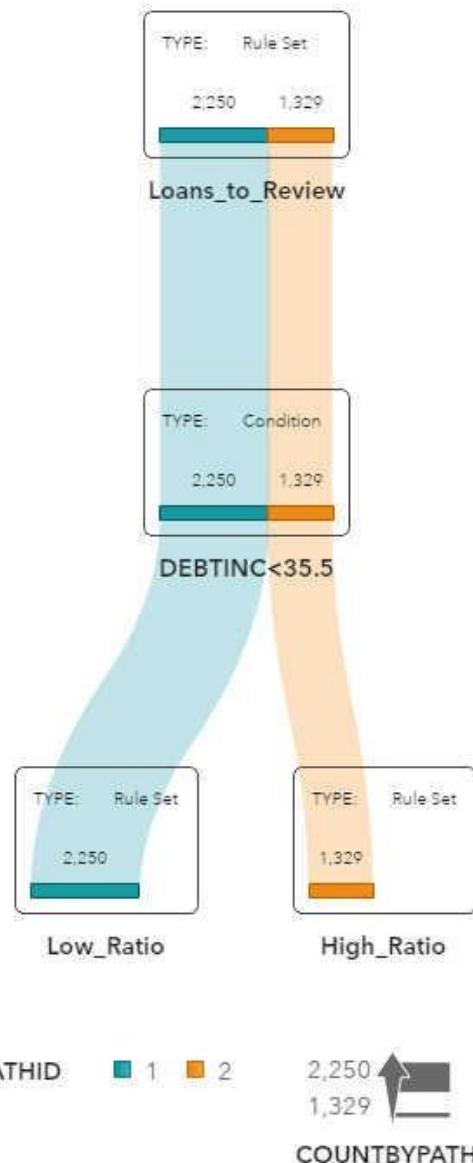


- c. Run a decision path tracking analysis.

Decision path tracking shows you the route that input records take through the rule sets and conditions in your decision.

- 1) Click **Decision Path Tracking** in the navigation pane.
- 2) Click **Run Path Tracking** to run a decision path analysis.

- 3) Click **Analysis** \Rightarrow **Plot** to display a Sankey diagram that shows the flow of the input records through the rule sets and conditions in the decision. The numbers in the diagram are the number of rows in the input table that followed each path.



- 4) Click **Node Count** in the navigation pane to display a table showing the number of input records evaluated at each node in the decision.
- 5) Click **Close** to close the Test Results window.
- d. Publish the **Evaluate_Loans** decision.

Publishing content makes it available to other applications. In this exercise, you publish only the decision, but you could also publish the rule sets. Publishing a decision or a rule set creates an entity that can be managed and run in another environment. For example, if you publish content to the SAS Micro Analytic Service destination, SAS Decision Manager creates a DS2 package that can be managed and run through the SAS Micro Analytic Service interface.

- 1) Click **Publish**. The Publish Decision window appears.

- 2) Select the destination where you want to publish the decision, and click **Publish**. The Publishing Results window displays the name of the published decision, its status, and information about any issues encountered while publishing the decision.
- 3) Click **Close** to close the Publishing_Results window.
- 4) Click **Close** to close the decision.

End of Solutions