

Introducing SAS® Model Manager 15.1 for SAS® Viya®

Glenn Clingroth, Robert Chu, Steve Sparano, David Duling

SAS Institute Inc.

ABSTRACT

SAS® Model Manager has been a popular product since 2006 for customers who need software for managing their collections of valuable analytical models with functions for organization, search, editing, testing, deployment, and monitoring performance. Some customers have thousands of models. SAS® Viya brings a new version of Model Manager with a modern set of features and capabilities. This paper will tour those new features including a new web application for managing models and projects; integration with SAS® Visual Statistics, Model Studio, SAS® Decision Manager, and SAS® Event Stream Manager; computations based on the SAS® Cloud Analytic Server; model deployment to multiple operational servers; and compatibility with open source environments such as Python. Model Manager is now cloud-ready, multi-tenant capable, and provides a full REST API for custom application integration.

INTRODUCTION

The primary goal for SAS® Model Manager 15.1 is to provide model management capabilities for customers working in the SAS® Viya® environment – the newest version of the SAS system that provides a modern compute server and cloud ready services environment. Effective organizations not only build great models, they also incorporate these models into production processes for automated decision processing. Model Manager has been a successful product for bridging the gap between data scientists and business operations. This paper describes the major features of SAS Model Manager 15.1 and describes how it helps you manage and incorporate models into your organization's processes.

MODEL MANAGEMENT

A model is defined as a programmatic function that has been created by an analytical operation. For example, a data scientist might use a machine learning operation to build a model that predicts the probability that a customer will fail to repay a loan. The function will have a set of input variables and a set of output variables. The final model function may be saved in several forms: a full SAS program, a SAS data step, a SAS DS2 package, or a SAS Analytic Store file. All these models can be managed and executed by SAS Model Manager. In addition, models created with open source and third-party systems such as Python, R, MATLAB, and many others can be stored and managed in Model Manager.

Model function code is often referred to as *score code*, and the execution of that code is termed *scoring*.

A model can be more than just a programmatic function. A model can also contain properties such as who created and updated the model, the key attributes of the model, when the model was created, updated, and deployed, and why the model was created.

The following table shows the contents of a basic model. When the model is scored, the output variable, probability is used by downstream processes to make automated decisions. The score code is shown as SAS data step code, but the code could be in any programming language.

Score Code File	Properties
LENGTH LOAN MORTDUE P PROBABILITY 8 ; p= 0.735 + 1.681E-5 * LOAN + 4.199E-6 * MORTDUE ;	Name: Baseline logistic Description: Predict loan default based on two simple variables. Tool: SAS Viya

<pre> if p > 0 then p= 1 / (1 + exp(-p)) ; else p= exp(p) / (1 + exp(p)) ; probability= p ; </pre>	Function: Classification Algorithm: Logistic Regression Score code type: SAS Datastep Input Variables: LOAN (N), MORTDUE (N) Output Variables: PROBABILITY (N) Modeler: Glenn Clingroth Date: April 8, 2018
---	---

Table 1. A simple model showing the score code file and model properties.

In practice, useful models frequently have hundreds to thousands of lines of score code, one or more SAS Analytic Store files, other resource files, and dozens of properties and documents.

Modern businesses can have tens to thousands of these models, depending on how complex the business processes are and how often they create models. These models are critical assets to the business and must be governed as such.

This paper describes several key new features in SAS Model Manager 15.1 for SAS Viya.

ARCHITECTURE

SAS® Model Manager 15.1 is a core SAS Viya application that benefits from the services of the SAS Viya environment. The application requires the SAS® Visual Analytics package, the SAS® Cloud Analytic Services, and the SAS language compute service. All SAS Viya applications work together in a common services framework that provides scalability, manageability, and interoperability.

Figure 1 shows the three key pillars of a complete model management application.

1. Tools in the Model Creation category build models that are typically stored as a set of special purpose files. Model Manager works with both SAS and third party model development tools.
2. Tools in the Model Management category are included in SAS Model Manager. They provide content management, model selection, governance, testing, publishing, and monitoring functions. The Model Repository, Model Management, and Model Publish services provide the core functionality. They can be used by developers to create custom applications. The model management application provides a sophisticated user interface for all model management functions.
3. Tools displayed in the Model Execution category provide model run-time capabilities for different production environments. Some processes are run offline in scheduled batch processes. Some processes provide real-time scoring for online and streaming data applications. Some processes run inside external environments such as commercial data bases and open source pipelines. SAS Viya provides interfaces to run in all of these environments.

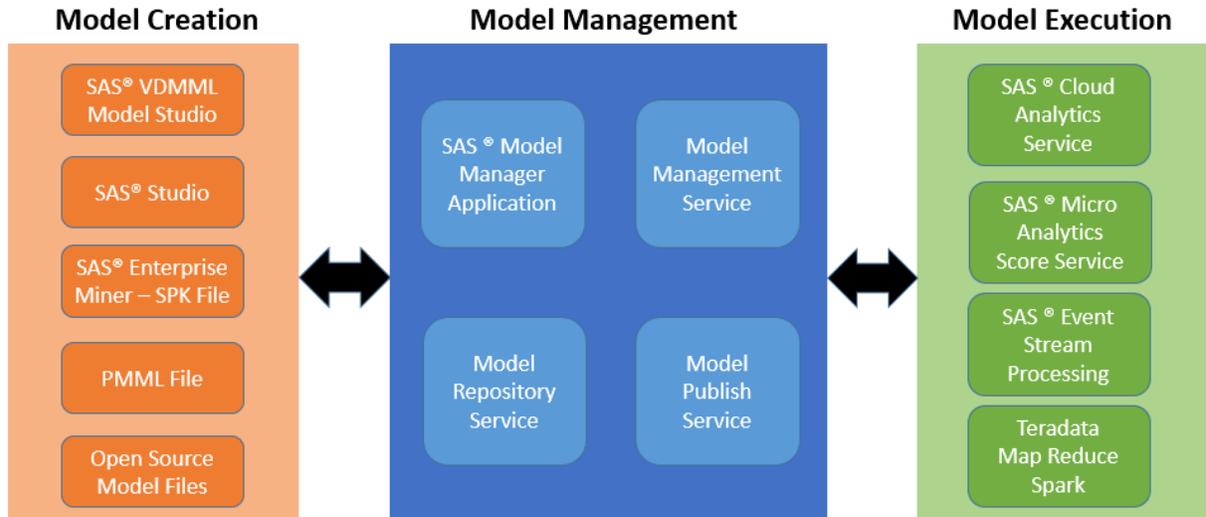


Figure 1. Model management architecture showing tools for model creation, management, and execution.

SAS® MODEL MANAGER

SAS Model Manager is delivered in HTML5 that passes rigorous security standards, is multi-tenant capable, cloud-ready, and scalable. The application provides user actions for all functionality delivered by the solution.

The following display shows the main view of SAS Model Manager. Models have been added from several different sources, including Model Studio and SAS® Studio.

The screenshot shows the SAS Model Manager interface for a project named 'bikeshare'. The main view is a table of models, with a navigation bar on the left and a top toolbar.

Name	R...	Model Function	Project Version	Algorithm	Date Modified	Modif... B:
Decision Tree (Pipeline 1)		Classification	Version 1 (1.0)	Decision tree	Mar 15, 2018 11:55 PM	brmdev
EM_AUTONEURAL		Classification	Version 1 (1.0)	AutoNeural	Mar 15, 2018 11:54 PM	brmdev
Logistic Regression (Pipeline 1)		Classification	Version 1 (1.0)	Logistic regression	Mar 12, 2018 11:50 PM	brmdev
Neural Network (Pipeline 1)		Classification	Version 1 (1.0)	Neural networks	Mar 12, 2018 11:50 PM	brmdev
PMML_Logistic_Regression		Classification	Version 1 (1.0)		Mar 14, 2018 10:07 PM	brmdev
Tree From SAS Studio		Classification	Version 1 (1.0)	Decision tree	Mar 13, 2018 12:07 AM	brmdev
bikereg		Classification	Version 1 (1.0)		Mar 14, 2018 09:55 PM	brmdev

Figure 2. A SAS® Model Manager project containing models from multiple sources.

The SAS Model Manager user interface provides the following features:

- SAS Model Manager displays categories for models, projects, and workflow tasks in the navigation bar on the left of the window.

- Figure 2 shows the main view of a model project. A project contains candidate and champion models, tests, and reports. A project enables you to import and modify models, create and manage versions of a model, test and monitor models, and publish models to the model repository.
- The models listed in Figure 2 are for project “bikeshare” version 1.0. Additional versions of the project will contain new sets of models. Project versions are used for evaluating models at different points in time.
- The New Model and Import buttons enable you to add models to the project versions.
- The Models page displays the list of models available in a project. In Figure 2, the EM_AUTONEURAL model is the current champion, and the Decision Tree (Pipeline 1) model is the current challenger.
- The Variables page displays the list of required or allowed variables for models in a project.
- The Tests page displays all of the tests that have been run with the models in a project. You can run tests to verify that a model is performing properly before publishing it to run-time environments.
- The Performance page displays the current state of the model monitoring jobs. A monitoring job processes production data in order to detect changes and trends in data, model output, and model accuracy. This information is used to plan downstream business processes and schedule rebuilding of the model when performance has significantly degraded.
- The Workflow page displays the workflows have been started, completed, or terminated for a project. SAS Model Manager can start and stop a single workflow for each project. Each workflow can schedule the tasks of multiple users and execute automated tasks. SAS workflows conform to the BPMN standard.
- The History page lists the important events in the life cycle of a project, including model repository, model management, and model publishing events.

MODEL INTEGRATION

SAS Model Manager 15.1 enables you to manage any type of analytical model from any source. You may add models that were developed in tools such as SAS Viya, SAS® 9.4, Python, R, Scala, Google, Amazon, MATLAB®, Java™, IBM® SPSS, and other possible source of analytical content. Several SAS applications have been engineered to directly add models to the repository including SAS® Visual Data Mining and Machine Learning, SAS® Visual Text Analytics, and SAS® Studio.

- **SAS Visual Statistics and Model Studio** provide point-and-click environments for building data mining, text analysis, and forecasting models in the SAS Viya environment. After models are created and selected, you can register the models to the model repository service and open the models in SAS Model Manager.
- **SAS® Studio** provides an interactive environment for developing SAS programs. Developers can build models using the SAS language and then use custom macros supplied with SAS Model Manager to save these models to the model repository service. After a model is saved to the model repository service, it can be managed by SAS Model Manager. By using the custom macros, you can build automated systems that train and manage thousands of models in batch processes. You can also use SAS Model Manager to import SAS program files, data step code files, or DS2 code files. In SAS Model Manager, you can import any number of files and create new model definitions.
- **SAS® Enterprise Miner** on SAS 9.4 can rapidly build numerous models. Use the Create Package option to save models in an SPK format that contains all of the model information

including a full description, all score code, and all results for each step of the process. In SAS Model Manager, you can import and read the SPK file.

- **PMML** files created by SAS or external modelling tools can be imported into SAS Model Manager. If the PMML file conforms to the latest specification, Model Manager can create data step score code for execution in numerous environments. You can import PMML model files into SAS Model Manager.
- **Open Source and external vendor models**, including models written in Python, R, MATLAB®, and so on, can be saved as a collection of special purpose files. In SAS Model Manager, you can import any number of files and create new model definitions in the model repository.

The following display shows the main elements of a model.

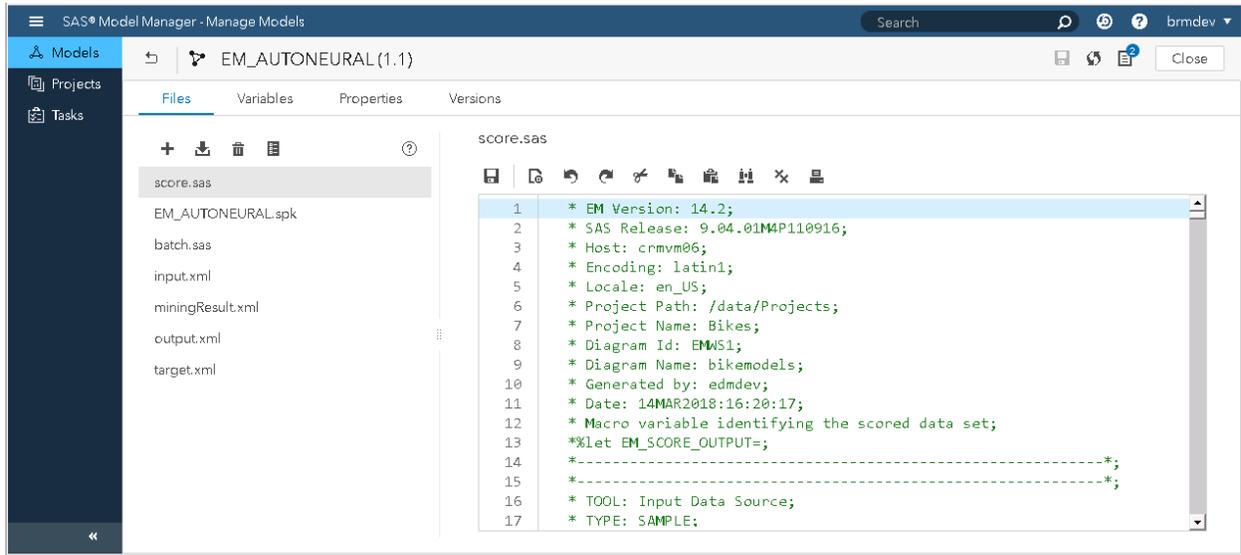


Figure 3. A model comprising several model files and properties.

Figure 3 shows the following features:

- The **Files page** displays all of the content that was loaded into the model definition. The score.sas file is the model score code. The batch.sas file is the model retraining code. Models from different sources will have different sets of files. This model was generated using SAS® Enterprise Miner. The user may interactively edit the score code or batch code to fix errors or make necessary changes.
- The **Variables page** displays the input and output terms used by the model. You can interactively edit the list of variables to conform to changes in the model score code or address inconsistencies in the model score code.
- The **Properties page** displays information about the model such as the model function, algorithm, score code language, and software tool. You can edit these properties.
- The **Versions page** lists the versions of a model.

MODEL GOVERNANCE

Governance is particularly important in the financial services. However, this practice now extends to other industries that have become dependent on analytical models to meet business objectives and expectations. Most businesses need to govern the model creation and deployment process in order to demonstrate that they are complying with both internal and external regulations, policies, and procedures. Model governance requires that all production models be tracked and managed.

SAS Model Manager is designed to provide the core tools that are needed to meet governance demands during analytical model lifecycle management. Being able to manage the analytical model lifecycle is a big step toward achieving regulation compliance on analytical models.

With Model Manager, model versions can be created manually or automatically. For example, when a model is published to a production environment, a static snapshot of the model version is created. This model version has a unique ID. Given a model version ID in the production environment, you can query the model repository to get the exact content of the deployed model

SAS Model Manager also supports scoring job history, publish history, and audit trails that help you track which models have been published, when those models were used, and by whom. Model Manager integrates BPMN-compliant workflow features to help you define, manage, and monitor the processes in the analytical model lifecycle.

Typically, organizations might build many analytical models with tools from many vendors or open source projects. SAS Model Manager supports all types of model registration with versioning.

MODEL PUBLISHING

Model publishing is the process of preparing and staging a model into a production system for execution by run-time systems. For batch processing, the goal is often to stage the model into the native system where the production data is located in order to eliminate large data movement between production and analytical systems. This requires services in the production environment that can process and execute the model functions. For real-time processing, the goal is to stage the model into a scalable, low-latency execution server by using standard APIs.

SAS Model Manager provides functions for publishing models into both batch and real-time execution systems. The following services are currently supported:

- SAS Cloud Analytic Services. The model is staged in a table that is used by the run model actions on the server.
- Teradata database. The model is staged to a database table and the model score code is executed inside the database using scoring functions provided by SAS.
- Hadoop Map Reduce. The model is staged to a file in the designated directory and the model score code is executed inside the database using scoring functions provided by SAS
- Hadoop Spark. The model is staged to a file in the designated directory and the model score code is executed inside the database using scoring functions provided by SAS
- SAS Micro Analytic Score Service. The model is directly loaded and compiled into a dedicated real-time scoring service by using a REST API. The model is available immediately for score execution.
- SAS® Event Stream Processing. The model is loaded by SAS Event Stream Processing for integration into ultra-high speed streaming data applications for complex event processing.

After it is published, the model can be integrated with existing processes and executed directly against the data in the production system.

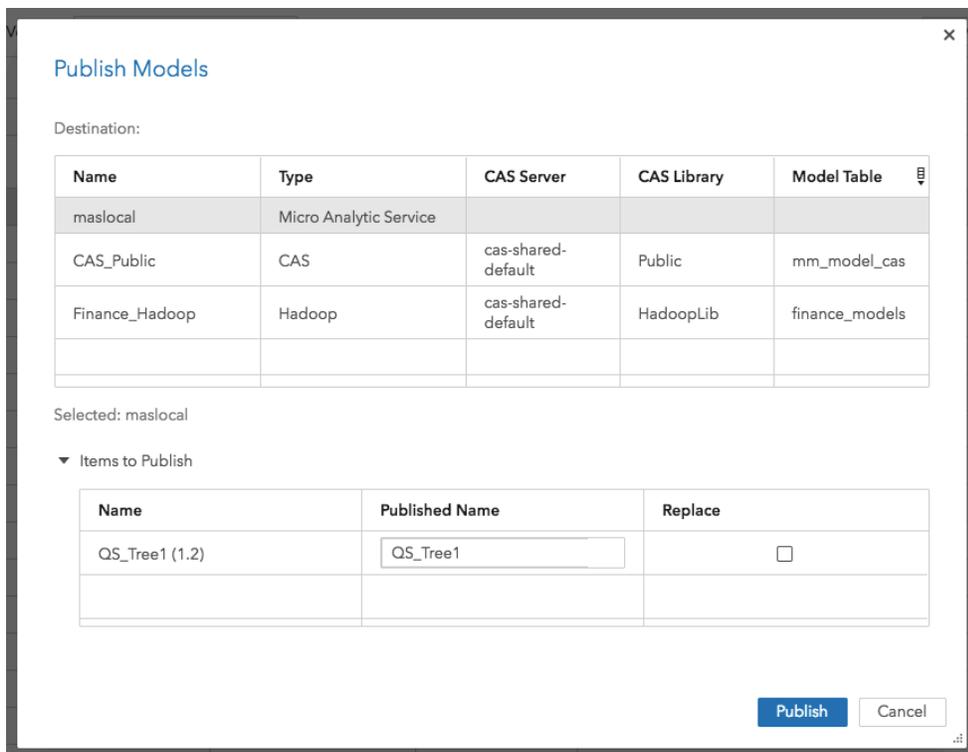


Figure 4. The model publish window showing three different publish destinations.

Not all model code types can be published to all destinations from SAS Model Manager. The following table shows the combinations of model score code types and destinations that are supported by SAS Model Manager. CAS refers to the SAS Cloud Analytic Services; MAS refers to the SAS Micro Analytic Score Service, and ESP refers to the SAS Event Stream Processing server.

Source Code Type	Publish Destination Type					
	SAS	CAS	MAS	ESP	Teradata	Hadoop
SAS Program	Yes					
SAS Data Step and DS2	Yes	Yes	Yes	Yes	Yes	Yes
SAS Astore (machine learning model)	Yes	Yes	Yes *	Yes	Yes	Yes
PMML (converted to Data Step)	Yes	Yes	Yes	Yes	Yes	Yes
Python	Yes		Yes	Yes		

Table 2. Publishing Support in SAS® Model Manager

REST INTERFACES

Do you need to build or customize business applications using SAS services? In SAS Viya, all services provide RESTful APIs that can be used by organizations for custom integration. These are the same APIs that are used by the SAS solutions. Each of the core model management services--model repository, model publishing, and model management--provide a comprehensive REST API that, together, covers all of the functionality in the SAS Model Manager web application.

Model management functions are separated into three APIs:

- **MODEL REPOSITORY API**

The Model Repository API provides the core RESTful API for model repository management and supports registering, organizing, and managing models. The main functional areas of the API are Repositories, Projects, and Models.

- **MODEL MANAGEMENT API**

The Model Management API provides a RESTful API that works with the model repository. The main functional areas of the API are model performance and comparison, workflow management, and code retrieval for scoring and publishing.

- **MODEL PUBLISH API**

The Model Publish API provides a RESTful API that supports publishing models for use in production data servers such as CAS, Hadoop, SAS Micro Analytic Service, and Teradata. The main functions of the API are destination management and model publishing.

Model management is a core feature in SAS Viya. To provide a robust model management environment, the SAS Model Manager APIs use the core SAS Viya platform APIs, such as authorization, folders, and files in addition to specialized APIs such as score execution, workflow, and compute service.

Documentation for these services conforms to the OpenAPI standard and is available at <https://developer.sas.com/guides/rest.html>.

INTEGRATION WITH SAS[®] DECISION MANAGER AND SAS[®] EVENT STREAM PROCESSING

Applications also query and read content *from* the SAS Model Manager services. SAS[®] Decision Manager and SAS[®] Event Stream Processing do not add models to the model services. They read models from the services and integrate the models into their own processes. This is a prime example of how any application can include model execution (scoring) into their business processes.

SAS Decision Manager provides advanced capabilities for building rules and decision flows. Using predictive models in decision flows is an effective way to make decisions that are smarter and more effective. When you add a model to a decision flow, SAS Decision Manager displays the list of models from which to choose. The variables from the model that you choose are added to the decision flow, and you can use the output variables from the model in any business rule or condition in the decision. For example, the decision flow in Figure 5 uses the model named DS1_HMEQ_DMREG. This model calculates the value of EM_EVENTPROBABILITY. The value of this variable is used in the condition immediately following the model. You can test and deploy the decision flow to the same servers that are available to SAS Model Manager.

SAS[®] Event Stream Processing integrates models in a similar way. Using SAS Event Stream Processing Studio, you can create a streaming data continuous query that executes on the Event Stream Processing server. You can select a champion model from a model project to be included in the query flow. Models are automatically deployed with the query definition. In addition, when a new champion model is defined, a message is sent to the SAS[®] Event Stream Manager that fetches the new model from the repository and deploys it into the running Event Stream Processing servers. This dynamic update is a very new and powerful feature.

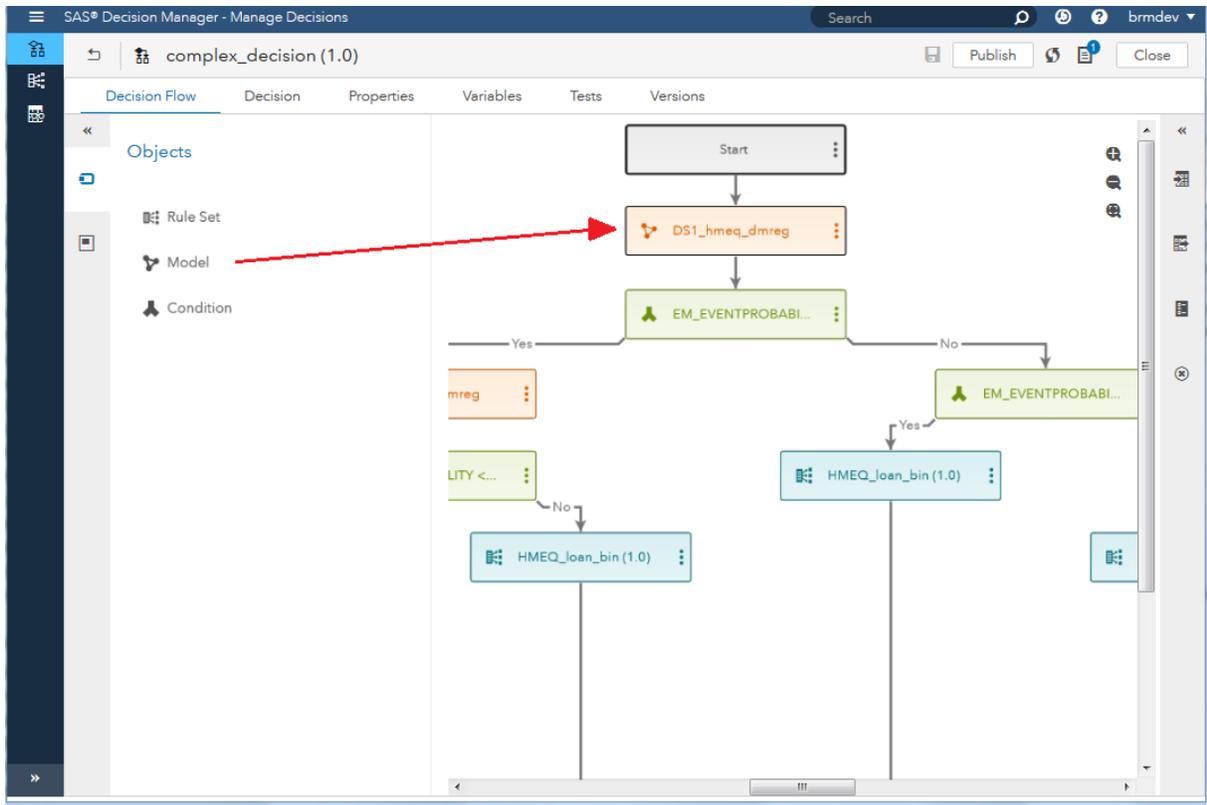


Figure 5. Adding a model to a SAS® Decision Manager decision flow.

MODEL MONITORING

After a model is published, a key activity is monitoring the model performance over time to detect changes in data that might be causing degradation in the model performance. All data changes over time so all models will show degradation. SAS Model Manager provides functions for measuring data change and model degradation over time. Data change is monitored by use of a Population Stability Index (PSI) measurement. Model change is monitored by calculating common model accuracy data such as Lift and ROC statistics at each time point. This data is appended to a set of dedicated tables maintained in the MMLIB library on the Cloud Analytic Service. Applications such as SAS® Visual Analytics and SAS® Studio can use this data to create advanced reports and run analyses.

For example, the graph in Figure 6 was created by running monitoring code in SAS Studio and writing custom code to create the graph using PROC SGPLOT in SAS/GRAPH®. The ROC curve is an indicator of overall model discrimination between the binary states of the dependent (target) variable. A higher curve indicates a better model than a lower curve. Data scientists and business analysts examine the monitoring output for signals that the model needs to be retrained or recreated. The retrained or recreated model goes through the same model management process in order to be validated and moved into production. SAS Model Manager will support the entire model lifecycle.

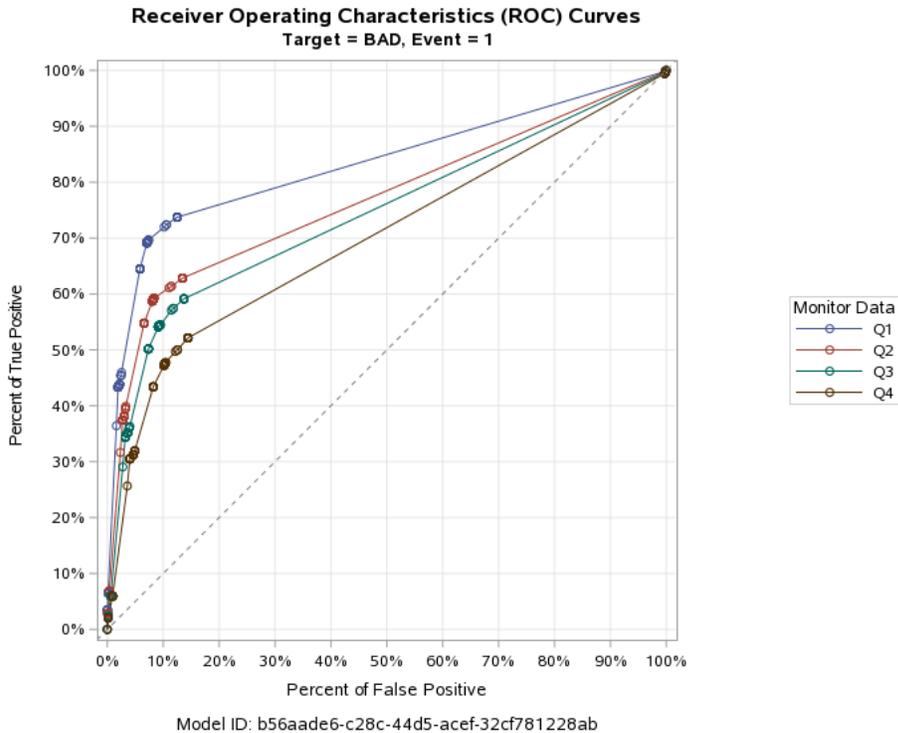


Figure 6. The model monitoring job produces measures such as these ROC curves.

CONCLUSION

SAS® Model Manager 15.1 is a major step forward for SAS Viya users. It provides features for all of the main activities of model management including storage, management, testing, publishing, and monitoring. The core model management services provide a REST API that enables developers to build custom applications. Both SAS and third party software can be included in the same projects. Models can be tested and deployed to multiple execution environments. The Cloud Analytics Service provides big data processing, machine learning, and artificial intelligence. With SAS Viya and SAS Model Manager, you can find solutions for nearly every model management and model execution project.

REFERENCES

Smith, Stephen. "Operationalizing Data Science." Eckerson Group. Accessed February 1, 2018. Available at <https://www.eckerson.com/articles/stephen-smith-operationalizing-data-science>.

Lukic, Rad. "7 Key Components of a Sound Model Governance Framework." S&P Global Market Intelligence. Accessed February 1, 2018. Available at <https://marketintelligence.spglobal.com/blog/7-key-components-of-a-sound-model-governance-framework>

ACKNOWLEDGMENTS

The authors thank Kristen Aponte and Jeanne Ferneyhough for content and editing.

RECOMMENDED READING

- SAS® Model Manager 15.1: User's Guide
- SAS® Decision Manager 5.1: User's Guide
- SAS® Micro Analytic Service 5.1: Programming and Administration Guide

- SAS® *Visual Data Mining and Machine Learning 8.2: User's Guide*
- SAS® *Event Stream Processing 5.1 documentation*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Steve Sparano
SAS Institute, Inc.
919-677-8000
Steve.Sparano@SAS.COM

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.