

Accelerate Your End-to-End Enterprise Decisions

David R. Duling, Stephen Sparano, SAS Institute Inc.

ABSTRACT

Do you need to add speed to your enterprise decision systems? SAS® Viya to the rescue. This paper guides you through the entire process of building and deploying decisions systems that integrate advanced analytical models with precise business rules, using software available only on SAS Viya. This software includes SAS® Visual Data Mining and Machine Learning, SAS® Model Manager, and SAS® Decision Manager. Decision processes can be deployed to SAS® Cloud Analytic Services, Teradata and Apache Hadoop databases, SAS Micro Analytic Service for on-demand processing, and SAS® Event Stream Manager for integration into high-speed data streams.

INTRODUCTION

All businesses depend on quality decision making. In many cases, computer automated decision logic can be embedded into production business processes to speed and improve those processes. That decision logic can now be completely developed and deployed using the tools already available in SAS Viya. This paper discusses the entire process with a focus on predictive models and business rules. SAS Viya now contains all the essential tools for building and managing critical business decisions.

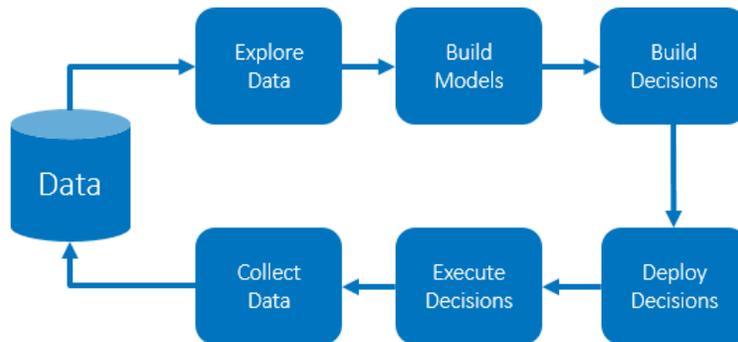


Figure 1. The lifecycle of a business decisions process.

DATA

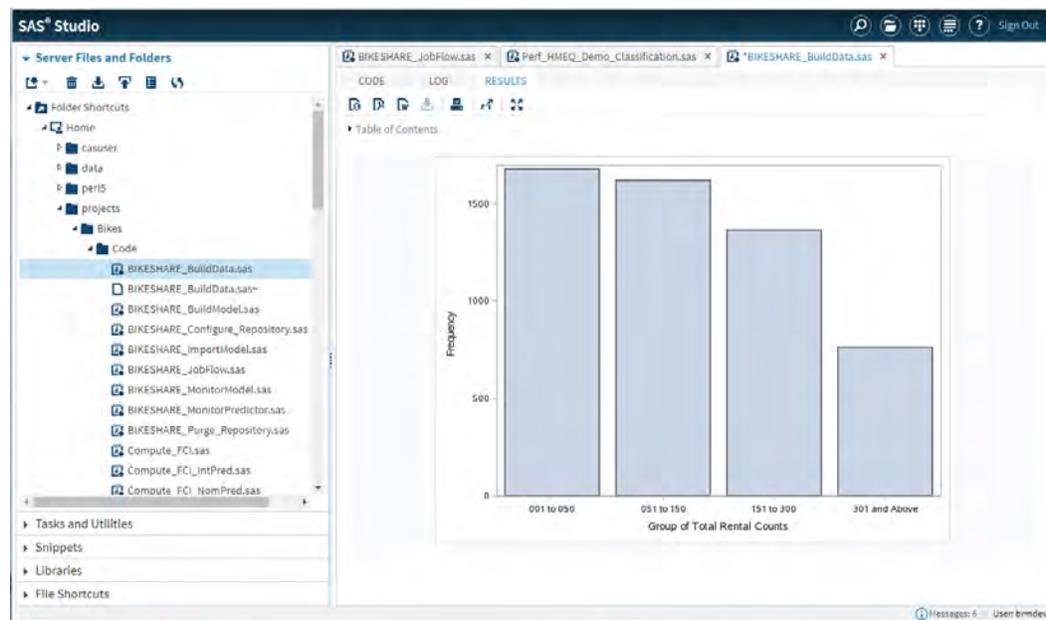
This analysis examines a bike sharing company data set. Bike sharing has become a popular service in many urban areas and data has become a frequent topic for data analysis. This particular data was inspired by those examples. It has 5422 observations containing variables including date, time, conditions, and bike rental counts. The goal of this project is to manage the bike rental supply to match demand. We need to create an automated decision making process that will alert staff to potential bike shortages. The data was originally available as a CSV table that was imported into a SAS Studio session on a SAS Viya system to begin the analysis. The following variables were used in this analysis

Variable Name	Variable Label
datetime	hourly date + timestamp
Season	1 = spring, 2 = summer, 3 = fall, 4 = winter
Holiday	whether the day is considered a holiday

workingday	whether the day is neither a weekend nor holiday
weather	1: Clear, Few clouds, Partly cloudy, Partly cloudy ; 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist ; 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds; 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
Temp	temperature in Celsius
Atemp	"feels like" temperature in Celsius
humidity	relative humidity
windspeed	wind speed
Casual	number of non-registered user rentals initiated
registered	number of registered user rentals initiated
Count_group	number of total rentals

Error! Reference source not found.. Variables in the BIKESHARE_TRAIN data used in this paper.

We binned several variables into logical groups, including the rental count variable which had a broad distribution. Contact the authors for the complete SAS code files. The resulting variable, Count_group, contains only four levels. This variable will be the target of our analysis.



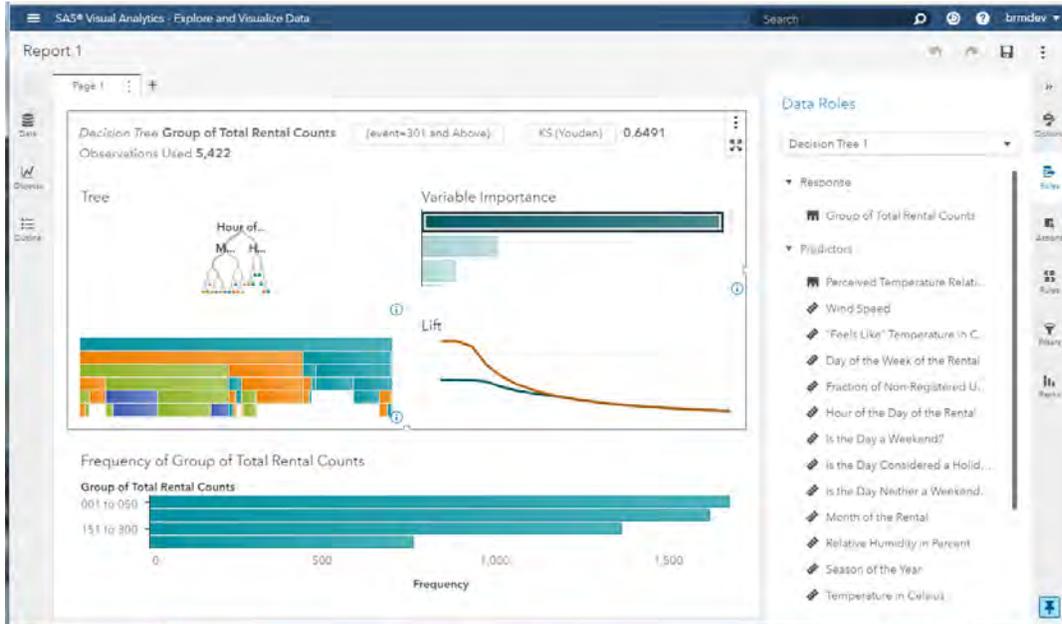
Display 1. The distribution of the four levels of the Count_group variable.

EXPLORE DATA

We used SAS ® Visual Analytics to explore initial predictive models for the data. This software easily creates reports, statistical analysis, and basic machine learning models. We selected the BIKESHARE_TRAIN data created in our SAS Studio session. With Visual Analytics, we created a decision tree model to validate that there is a predictive relationship between the input variables and the dependent variable Count_group. Simply drag the Decision Tree object, found in the Visual Statistics list,

onto the working area. Select the Count_group for the response variable. The Tree will be automatically updated.

The following display shows the resulting Decision Tree model. The depth of the tree, the number of variables selected, the KS value, and the relatively high values of lift indicate that a useful predictive model can be generated for this data. At this point we can choose to save this model or to continue analysis in SAS Model Studio. For this exercise, we can choose the latter. From the model options, choose 'Create Pipeline' to initiate transfer the data and model into a new Model Studio project.

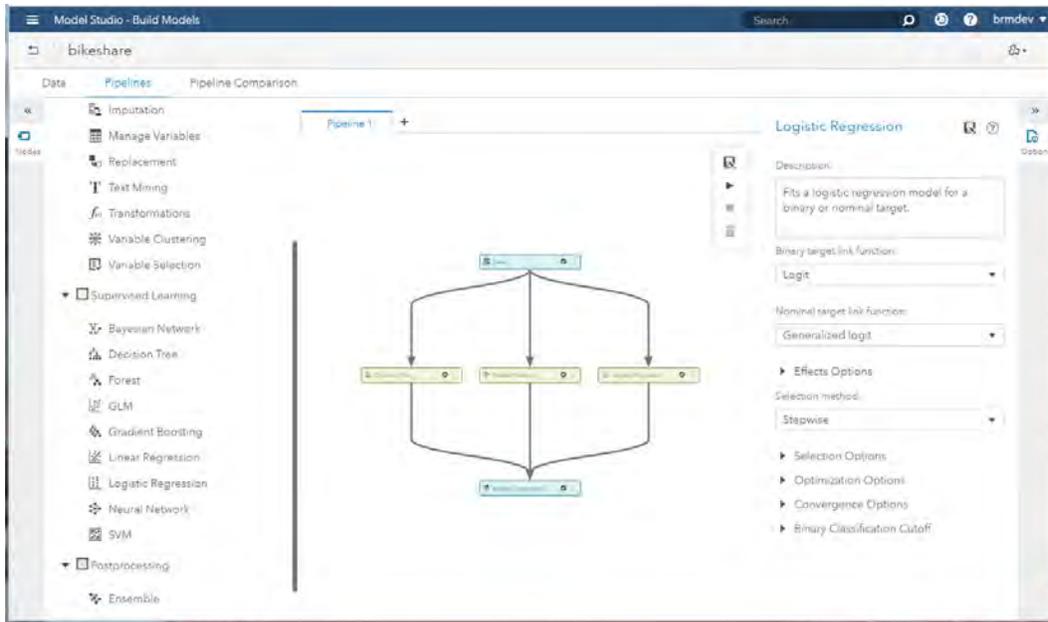


Display 2. The Decision Tree model in SAS Visual Analytics.

BUILD MODELS

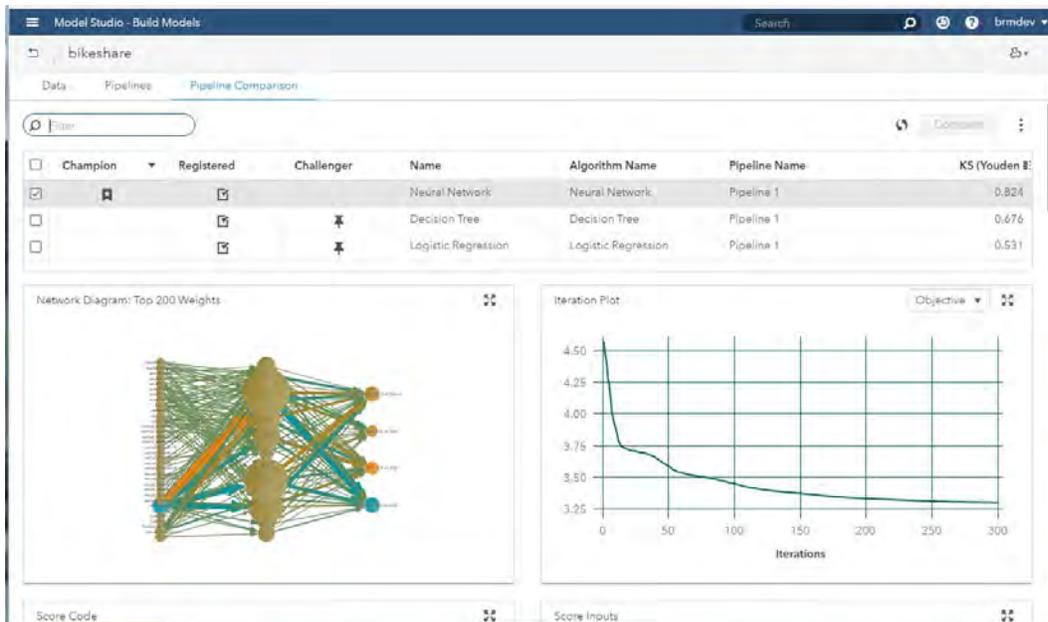
To build a candidate set of final models, we use the SAS Model Studio application which is part of the SAS® Visual Data Mining and Machine Learning package. This application has a rich set of tools for preparing and cleaning data, transforming variables, and a wide variety of modern machine learning algorithms. For this paper, we will select basic models to maintain focus on the entire decision management process. Since we already prepared the data using SAS Studio, we can proceed directly to the supervised modeling tools.

SAS Model Studio uses a process flow diagram for all critical model building operations. You will drag nodes to the diagram to implement the modeling process. You then run the entire process to produce the models. In this case, we selected the Decision Tree, Neural Network, and Logistic Regression nodes.



Display 3. Creating models with a process flow diagram.

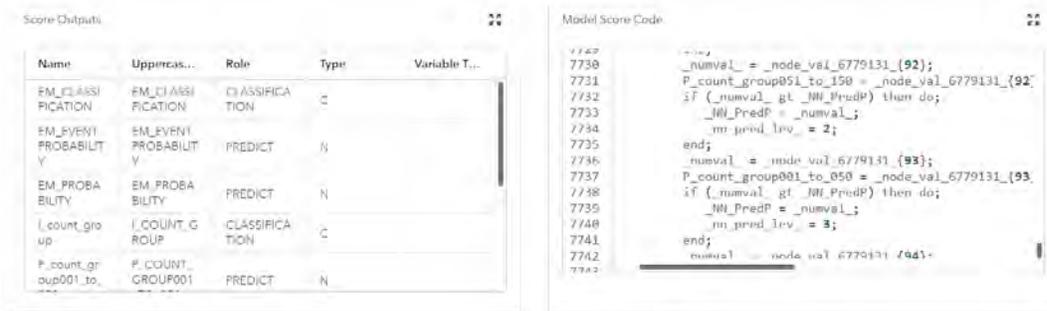
Each model will be trained and a prospective champion model will be selected based on selectable statistical criteria. You can also tag the alternative models as challengers to list all models in the Pipeline Comparison page as shown in Display 4. We can see that the Neural Network has a superior value for the KS (Kolmogorov-Smirnov, a measure of accuracy) statistic, and has been selected as the champion model.



Display 4. The Pipeline Comparison page shows the champion and challenger models.

In the pipelines list, all models have a check mark in the registered column. This indicates that the models have been saved to the SAS Model Repository service. This is a common service in SAS Viya for storage of analytical models and serves as an exchange point for SAS applications. A SAS Studio user can also save models to the model repository using SAS code and macros supplied with SAS Viya.

One more detail about the model should be examined. The Pipeline Comparison page shows the Model Score Code which is an essential feature of any model that will be used in scoring or decision making processes. The Score Outputs display shows the list of variables that are created by this code. We will use this code and these output variables in our decision logic.



Display 5. Score Output variables and the Model Score Code.

MANAGE MODELS

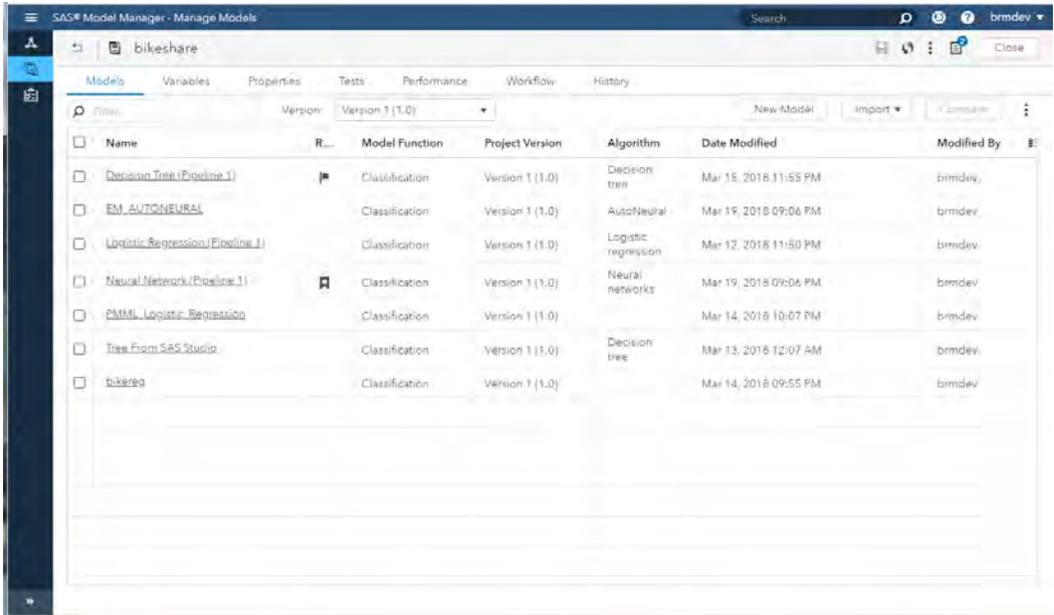
SAS® Model Manager 15.1 is the primary product for managing a collection of models used in scoring and production systems which are used in automated decision making processes. SAS Model Manager enables you to manage any type of analytical model from any source. You can add models that were developed in tools such as SAS® Viya, SAS 9.4, Python, R, Scala, Google, Amazon, Matlab, Java, IBM® SPSS, and any 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.

A primary reason for managing your inventory of models is governance, which 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.

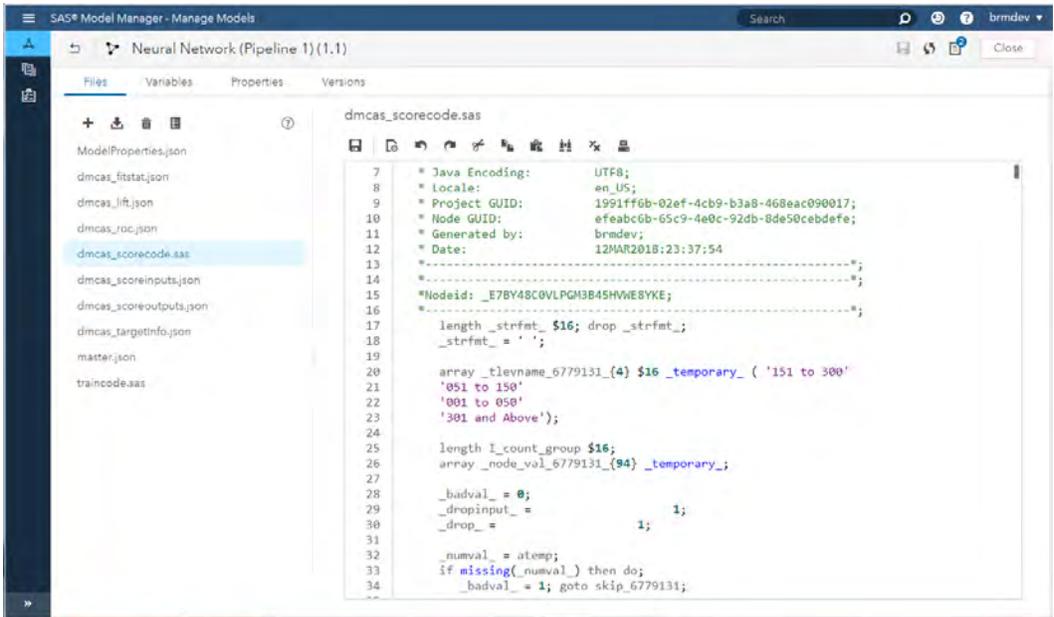
With SAS Model Manager, model versions can be created manually or automatically. For example, when a model is published to a production environment, a static snapshot of the model version is created. This model version has a unique ID. Given a model version ID in the production environment, you can query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

Display 6 shows a project in SAS Model Manager that was created when the models were registered from SAS Model Studio. These models all have the term 'Pipeline' in their title. The other models were added from additional sources including SAS Enterprise Miner, SAS Studio and an external PMML file. The pipeline Neural Network and Decision Tree models show the champion and challenger icons, respectively. Inside SAS Model Manager, you can change the champion and challenger models, compare properties, and test the models for expected performance and accuracy.



Display 6. SAS Model Manager showing the “bikeshare” project and list of models.

From the list of models, you can drill into the details of any model. Display 7 shows the pipeline Neural Network model. In this case we can see the same score code, statistics, variables and properties we could see in SAS Model Studio. In Model Manager, every model is stored as a collection of files that can be created and modified by the application or user. All changes are tracked with version management.



Display 7. The Neural Network model shown in SAS Model Manager.

BUILD DECISIONS

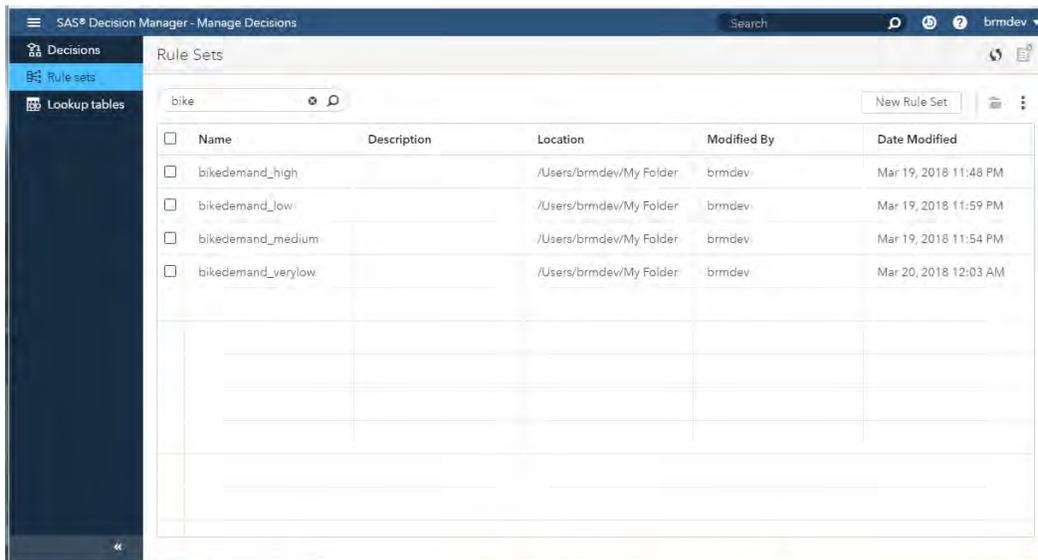
A model produces a useful and valid estimate of a probability or a prediction, such as the probability that a customer will fail to repay a loan or respond to a discount offer, or the demand for bikes at bikeshare locations. There are thousands of reasons to build predictive models. However the model does not make business decisions. It needs to be combined with business rules to become a useful analytically-driven decision. These rules encode dynamic business logic that cannot be included in a model. A common pattern is to compute the model probability and then apply rules that create an outcome. In this example, the model score is divided into three types of actions.

```
IF MODEL_SCORE < 0.5 THEN ACTION=NONE;  
ELSE IF MODEL_SCORE < 0.75 THEN ACTION=REVIEW;  
ELSE IF MODEL_SCORE >= 0.75 THEN ACTION=ALERT;
```

In practice, business rule systems might have thousands of rules based on hundreds of variables. An organization might have dozens of business analysts with the title of 'rule writer' and a designated 'rule master' that enforces policies and procedures. The combination of models and rules are the basis of very powerful decision management systems. The output of our bikeshare demand model, applied to the model training data, creates four possible values. We will create a business rule set for each of these predictions.

Model Prediction	Count	Percent	Rule Set
'301 and Above'	644	11.8	bikedemand_high
'151 to 300'	1388	25.6	bikedemand_medium
'051 to 150'	1651	30.5	bikedemand_low
'001 to 050'	1739	32.1	bikedemand_verylow

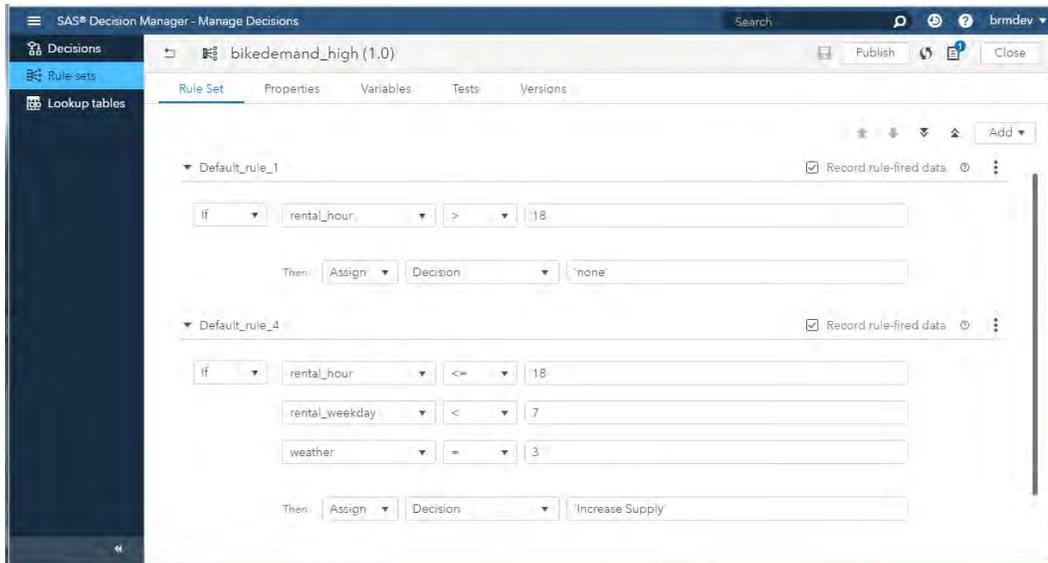
Display 8 shows the main screen of SAS Decision Manager with categories for Decisions, Rule sets, and Lookup tables. The list of rule sets is filtered to show only the items containing the term 'bike' in their name.



Display 8. A list of rules sets containing 'bike' in the name.

Opening the 'bikedemand_high' ruleset, we can find the rules shown in Display 9. The output of the rules is the variable 'Decision'. The main rule indicates that if the time is before 6pm (18) and the day is not a Sunday (7) and weather is good (3) then we should increase the supply of bicycles. The rules are used

to apply business logic in conjunction with the model. Similar rule sets were created for the other prediction values. You can modify the list of variables to change the input and output lists or add new variables to the rules, and you can test the business rules with historical or simulated data to make sure the expressions evaluate correctly and provide useful value. Finally, you can save versions of the rules and recall or rollback to prior versions as needed. Versioning is a key governance capability of a business rules system.

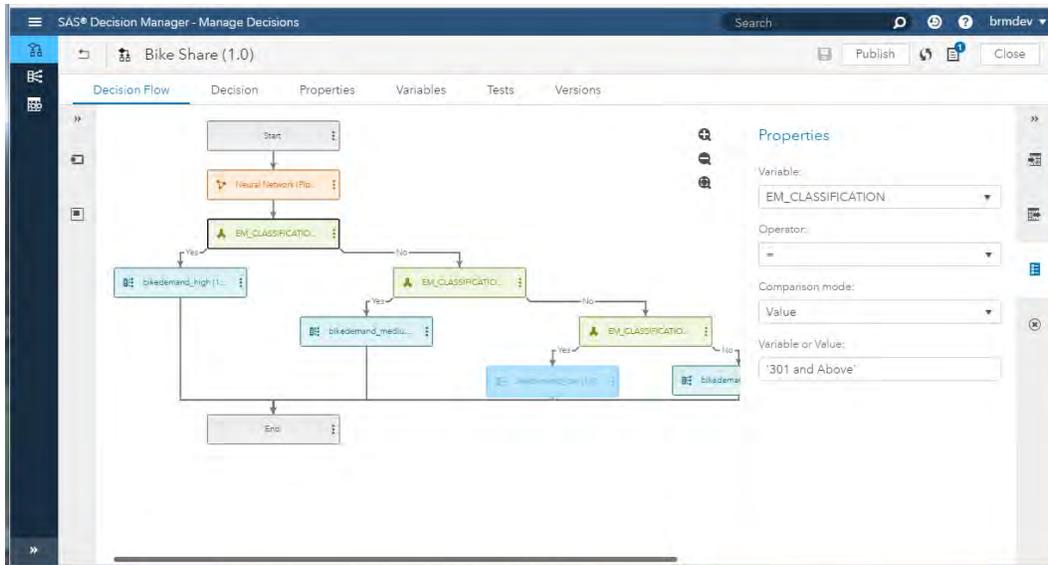


Display 9. An example of business rules for the bike sharing application.

A key benefit is that business users can modify the rules as strategy and conditions change, without changing the prediction model created by the data scientist.

Once the business rules have been created, they can be combined with the model predictions in a decision flow. This flow is represented by a process flow diagram that can contain models, rules, conditional branches, and custom SAS code. You can create this flow by dragging these elements to the diagram. When you drag a ruleset node, model node, or custom code node to the diagram, you will be prompted to select the corresponding object from the SAS file navigator. When you drag a condition node to the diagram you will be prompted for the variable and values that define the condition. You can combine these elements into any processing order to satisfy any logical needs.

Display 10 shows the diagram that has been created for the bike share application. The top node represents the entry point to the process. The next node is the Neural Network model which produces the bike demand predictions. The next node is a condition node that tests for the high demand condition '301 and Above'. If the condition is true then the 'bikeshare_demandhigh' rule set is executed. The remaining branches apply to the remaining prediction values and corresponding rule sets.



Display 10. The decision process flow diagram.

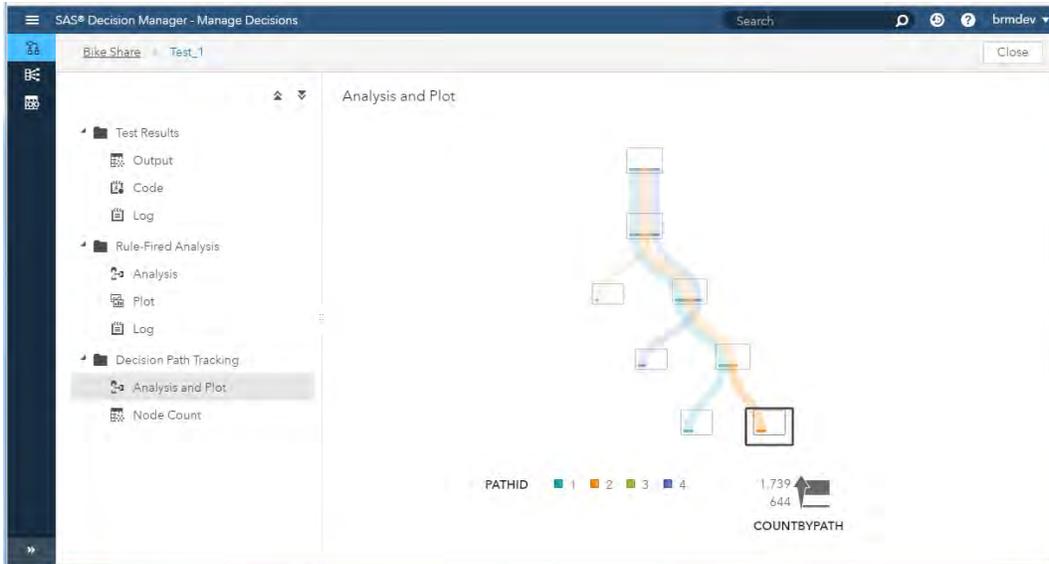
As the diagram and rules are being created, a good practice is to run tests with historical or simulated data. You can create a new test any time after the decision has been saved. You can create multiple tests each with a different input data set to review performance with different combinations or types of data values. Display 11 shows the results of a decision test. The Rule-Fire Analysis shows the rules that fired for each row of input data. This is a very powerful and useful tool for tracing and validating rule logic.

Rules Fired Count	Decision	Demandgroup
1	none	high
1	none	high
1	none	high
1	Increase Supply	high
1	none	high
1	none	high

Rule Set Name	Rule Name	Rule Order	Rule Logic
bikedemand_high	Default_rule_4		IF (rental_hour <= 18) AND (rental_weekday < 7) AND (weather = 3) THEN Decision = 'Increase Supply'

Display 11. An analysis of rule fire execution on test data.

Decision flow execution can also be analyzed. Display 12 shows a graphical flow view of test data through the decision flow. The top node represents the start of the flow and each node in the flow that receives data is represented in this graph. The node count in Display 13 confirms that the flow of decision data matches the expected values from the model output and validates the branching logic in the diagram.



Display 12. Graphical view of the flow of test data through the decisions flow.

Node Count

counts	node_name	node_type
5422	Neural Network (Pipeline 1)	Model
1739	bikedemand_verylow	Ruleset
644	bikedemand_high	Ruleset
1651	bikedemand_low	Ruleset
1388	bikedemand_medium	Ruleset

Display 13. Node counts from the path diagram.

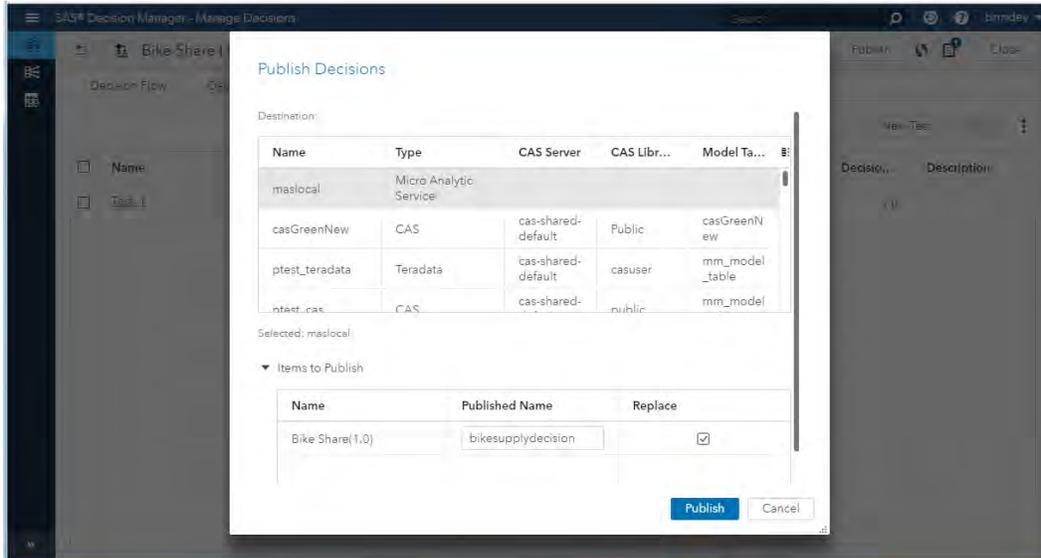
DEPLOY DECISIONS

Once you have fully developed and validated the business rules and decision flow logic, you need to deploy your decision process to a production environment. In this case, deploy simply means a process to generate the final run time code and stage it in a selected data processing server. SAS Decision Manager can use a common Viya publish service for deployment to the maximum number of environments. The following run time services are currently supported:

- SAS® Cloud Analytic Service. 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 or Spark. The model is staged to files in the designated directory and the model score code is executed using native scoring functions provided by SAS
- SAS Micro Analytics 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. This service is a real time web decision service that is a component of SAS Decision Manager

After publishing, the decision process can be integrated with existing processes and executed directly against data in the production system. Display 14 shows the publishing of the bike share decision to the Micro Analytic Score service. The decision is now a live RESTful endpoint that can be queried and executed by custom transaction processing applications. The API documentation is available from SAS.



Display 14. Publish the decision to a run time service.

EXECUTE DECISIONS

Once a decision has been published to any of the run time destinations, it can be used as a function directly in the destination, or called from a Cloud Analytic Service action for processing directly in the destination. Run time processes are typically created by the host organization's information technology department for integration into production systems. SAS is committed to supporting a wide variety of production run time systems for model and decision processing.

CONCLUSION

This paper has demonstrated the essential steps required to build a complete decision from data to model to business rules to decision to execution. A real business process would require more complex elements and many more of them. There are many more capabilities available in these products than described in this sample. We hope that you will be able to build upon this example to create your own custom decision processes using the full functionality of SAS Viya.

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>

Taylor, James. 2011. *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics*. IBM Press.

RECOMMENDED READING

- *SAS Model Manager 15.1: User's Guide*
- *SAS Decision Manager 5.1: User's Guide*
- *SAS Micro Analytics Service 5.1: Programming and Administration Guide*
- *SAS Visual Data Mining and Machine Learning 8.2: User's Guide*

CONTACT INFORMATION

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

David Duling
SAS Institute, Inc.
919-677-8000
David.Duling@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.