

Playing Favorites: Our Top 10 Model Studio Features in SAS® Visual Data Mining and Machine Learning

Wendy Czika and Peter Gerakios, SAS Institute Inc., Cary, NC

ABSTRACT

Model Studio in SAS® Visual Data Mining and Machine Learning provides a pipeline-centric, collaborative modeling environment that enables you to chain together steps (into a pipeline) for preprocessing your data, making predictions by using supervised learning algorithms, and then using the assessment measure of your choice to compare models. After the pipeline has determined the champion model, you can deploy that model to score new data. Since the first release of SAS Visual Data Mining and Machine Learning, many enhancements have been made to Model Studio, from nodes for running code (open-source code, batch code from SAS® Enterprise Miner™, or score code that is generated outside Model Studio) to more integration with other visual environments to allow for seamless exploration and visualization of your data as you build models. Although it is hard to limit the number of our favorite features, we present our top 10 features in Model Studio in SAS Visual Data Mining and Machine Learning 8.3 for relieving your biggest pains and increasing your productivity.

INTRODUCTION

SAS Visual Data Mining and Machine Learning is a collection of algorithms and utilities for data preparation and modeling that run via SAS® Cloud Analytic Services (CAS) actions in a distributed-computing, in-memory infrastructure. There are multiple interfaces to SAS Visual Data Mining and Machine Learning that enable you to programmatically analyze your data by using SAS® procedures, interactively explore and model your data in the visual interfaces of SAS® Visual Analytics and Model Studio, or use an open-source programming language to access SAS analytics.

Model Studio is a visual interface (Wujek, Haller, and Wexler 2018) to SAS® Viya® that leverages the cloud-enabled, in-memory analytics engine of CAS. Model Studio presents a modern approach to data mining that is specifically designed to serve as an extensible and open framework that lets you access data from a variety of common sources. By using CAS actions, you can invoke in-memory analytics not just with SAS, but also with Python, R, and Java.

In this paper, we offer 10 tips to help all users of Model Studio increase productivity and gain better insights into their data.

TIPS FOR CREATING PROJECTS AND PIPELINES

These tips help you efficiently create new projects and pipelines. This means that you can spend less time setting up your projects and more time building better models.

FAVORITE FEATURE #1: THE EXCHANGE AND GLOBAL METADATA

The Exchange

One great strength of Model Studio is its support for sharing and reusing different project components. The Exchange, which was named The Toolbox in Model Studio 8.2, is a space where you can organize your favorite nodes and pipelines, enabling you to collaborate with

others in one convenient location. For example, you can find a best-practice template that your colleague created or create a streamlined workflow that you can share with your team.

| Name | Description | Product | Owner | Last Modified |
|-----------------------------|--|----------------------------------|----------|--------------------------|
| Batch Code | Runs SAS batch code. | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:36 AM |
| Bayesian Network | Fits a Bayesian network model for a class t... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:37 AM |
| C4.5 Forest | | Data Mining and Machine Learning | emduser4 | Aug 27, 2018, 1:05:59 PM |
| C4.5 Gradient Boosting Node | | Data Mining and Machine Learning | scatest | Aug 27, 2018, 9:56:26 AM |
| Decision Tree | Fits a classification tree for a class target o... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:46 AM |
| Forest | Fits a forest model, which consists of multi... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:41 AM |
| GLM | Fits a generalized linear model for an int... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:41 AM |
| Gradient Boosting | Fits a gradient boosting model, which bui... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:41 AM |
| Linear Regression | Fits an ordinary least squares regression ... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:42 AM |
| Logistic Regression | Fits a logistic regression model for a binar... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:42 AM |
| Neural Network | Fits a fully-connected neural network mo... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:43 AM |
| Quantile Regression | Fits a quantile regression model for an int... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:43 AM |
| Score Code Import | Imports SAS score code. | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:44 AM |
| SVM | Fits a support vector machine via interior... | Data Mining and Machine Learning | SAS Node | Sep 6, 2018, 10:22:46 AM |

Display 1. Supervised Learning Templates in The Exchange

Display 1 shows templates for individual nodes and entire pipelines of nodes that you can reuse or share as needed. You or your organization might have a particular set of properties for your Decision Tree nodes that you always want to use instead of the defaults. After specifying these properties once, just click a button to save that node and your Decision Tree node is saved to The Exchange. From then on, you can use your version of the Decision Tree node without needing to change the default properties every time. This concept extends to an entire pipeline.

If you have a best-practice pipeline that you want to start from every time you add a pipeline to a project, you can save it and set that pipeline as the default pipeline. Doing this means that you no longer need to create your pipeline from scratch every time you start a new project or add a new pipeline to an existing project. However, Model Studio also includes a set of pipelines to help you get started. You can duplicate and modify these pipelines to suit your needs, or you can use them as-is.

Model Studio includes basic templates for simpler, more explainable models like regressions and decision trees, and advanced pipelines for more complex models that include the autotuning of hyperparameters. The most recent release of Model Studio includes an automated feature engineering template that tries several different feature engineering methods, then compares these feature sets by using autotuned gradient boosting models.

Global Metadata

When you create multiple projects that use similar data sets or that repeatedly use the same data set, you might find it useful to store the metadata configurations for use across projects. Variable metadata include any setting or information about a variable, such as its role or measurement level. Model Studio enables you to store the metadata settings in the global metadata repository, and then it automatically applies the saved settings to all variables in a new data set that have the same name as a variable in the global metadata repository. In Display 2, you can see that any time a data source contains the variable

Level, that variable is assigned the role of Input and level of Nominal. Similarly, any variable named Gender is assigned the role of Target, and a log transformation is specified for any variable named Amount.

| Variable Name | Label | Type | Role | Level | Order | Transform | Comment | Count | Missing |
|---------------|--------|-----------|--------|----------|---------|-----------|---------|-------|---------|
| amount | | Numeric | Input | Interval | Default | Log | | | 0.0000 |
| city | | Numeric | Input | Binary | Default | Default | | 2 | 0.0000 |
| Gender | Gender | Character | Target | Binary | Default | Default | | 2 | 0.0000 |
| id | | Numeric | ID | Interval | Default | Default | | | 0.0000 |
| is_workday | | Numeric | Input | Binary | Default | Default | | 2 | 0.0000 |
| Level | Level | Character | Input | Nominal | Default | Default | | 12 | 0.0000 |
| Stream | Stream | Character | Input | Nominal | Default | Default | | 15 | 0.0000 |
| weather | | Numeric | Input | Nominal | Default | Default | | 4 | 0.0000 |

Display 2. Global Metadata

FAVORITE FEATURE #2: CREATE A PIPELINE FROM SAS VISUAL ANALYTICS

SAS Visual Analytics provides another interface that enables you to explore and visualize your data. It includes some supervised learning models, such as neural networks and gradient boosting models, from SAS Data Mining and Machine Learning. After you create a model in SAS Visual Analytics, you can import it into a Model Studio pipeline for further model building and comparison.

The first release of Model Studio enabled you to create a pipeline in a new project. This means that with the click of a button, your model in SAS Visual Analytics was imported into a new Model Studio project. In the latest release of Model Studio, you can add a modeling object directly to any Model Studio project that uses the same target.

Regardless of whether you create a new project or use an existing one, the Model Studio project will contain a pipeline that is seeded with an Interactive Data Preparation node that performs any interactive transformations or calculations that were performed in SAS Visual Analytics. The latest release of Model Studio enables you to see the score code that is used to create the transformations and calculations that you applied in SAS Visual Analytics. Following this node is an interactive modeling node that corresponds to the model that you built in SAS Visual Analytics. When you run the interactive model in Model Studio, all the assessment results available in the other Model Studio Supervised Learning nodes are available in the results of this node. In addition, this model is a candidate to become the pipeline and project champion that is deployed to production.

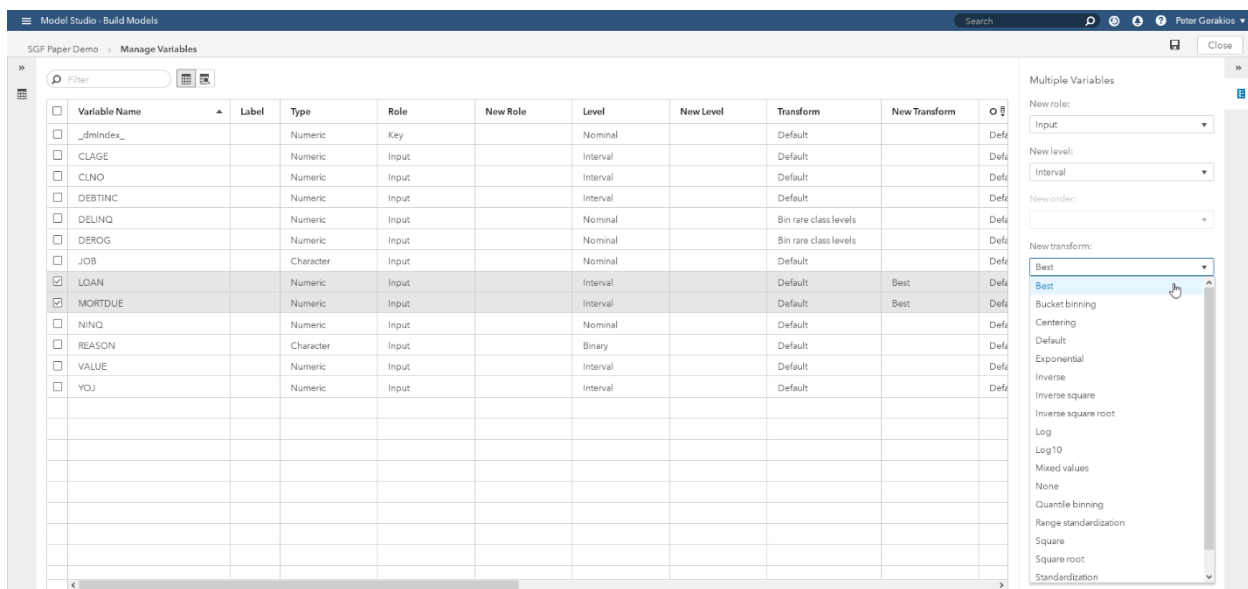
TIPS FOR DATA MINING PREPROCESSING

These tips help you clean and prepare your data for modeling. This means that you can build your models on the best possible data, creating better models.

FAVORITE FEATURE #3: BEST TRANSFORMATIONS

The Transformations node, already one of the most used Data Preprocessing nodes, now includes a very popular SAS Enterprise Miner feature: the ability to apply the “Best” transformation for interval inputs. The “Best” transformation includes the typical Box-Cox transformations and others, such as Centering, Log, Log10, and Standardization. Additionally, you can specify which criterion to use to determine the best transformation, including target-based assessments for either binary or interval targets.

If desired, you can specify a different transformation for each variable. For example, suppose you have two highly skewed variables. One responds best to the Log transformation, and the other responds best to the square root transformation. On the **Data** tab or in a Manage Variables node, you can individually specify the transformation for each input, as shown in Display 3.



Display 3. Best Transformation Selection in the Manage Variables Node

In Display 3, the transformations are specified for some of the inputs, including specifying “Best” for the variable LOAN. In this case, you might not know which transformations are good, much less best, for this variable. Therefore, you want Model Studio to determine the best transformation. Conversely, for variables like DEROG, you know that you want to apply a binning transformation. When the Transformations node is run on these variables, the best transformation of LOAN and MORTDUE is determined and then applied, whereas only the binning transformation is applied to DELINQ and DEROG.

FAVORITE FEATURE #4: ANOMALY DETECTION AND VARIABLE CLUSTERING

This feature highlights two of the lesser-known and undervalued Data Mining Preprocessing nodes: Anomaly Detection and Variable Clustering.

Anomaly Detection Node

The Anomaly Detection node uses a support vector data description method that was first introduced to SAS in PROC SVDD, which is available only in SAS Visual Data Mining and

Machine Learning in SAS Viya. The implementation of PROC SVDD in Model Studio includes some novel enhancements.

The Anomaly Detection node flags outliers in your data and excludes them from model training so that they do not unduly influence your predictions. To accomplish this, the Anomaly Detection node builds a hypersphere around your training data, and a threshold radius is chosen. Any observations that are farther from the center of the hypersphere than the threshold radius are deemed outliers.

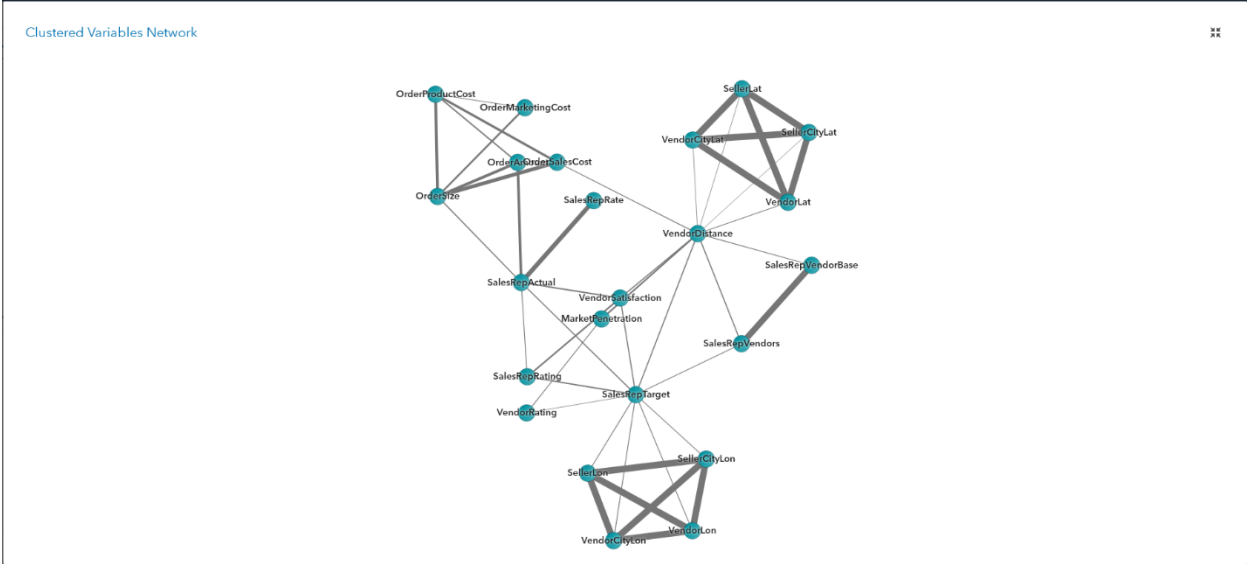
Variable Clustering Node

The Variable Clustering node uses a graphical Lasso approach for clustering inputs and then performs dimension reduction on the basis of the variable clusters. The clustering is performed by PROC GVARCLUS, which is available in SAS Visual Data Mining and Machine Learning in SAS Viya. This feature was available in the first release of Model Studio, but several new features were added in the subsequent release, making this node even more valuable.

Most SAS analytics handle classification or categorical variables by creating dummy variables or class level indicators. But the Variable Clustering node can do the one-hot, or dummy, encoding up front. Thus, when the dimension reduction is performed, you do not have to worry about significantly increasing the size of your data, because only the important, nonredundant levels are kept.

Also, instead of exporting individual variables or class level indicators that represent the variable clusters, you can choose to export cluster components. Cluster components are the first principal component from the variables in each cluster.

Finally, clustering is performed iteratively until a specified stopping criterion is reached, giving you a hierarchical clustering structure. In the most recent version of Model Studio, the optimal clustering configuration can be automatically chosen on the basis of the penalized log-likelihood value. The results of the hierarchy building process are included in the node's results. Display 4 shows the image that Model Studio displays for the final, chosen clustered variable network.



Display 4. Clustered Variable Network

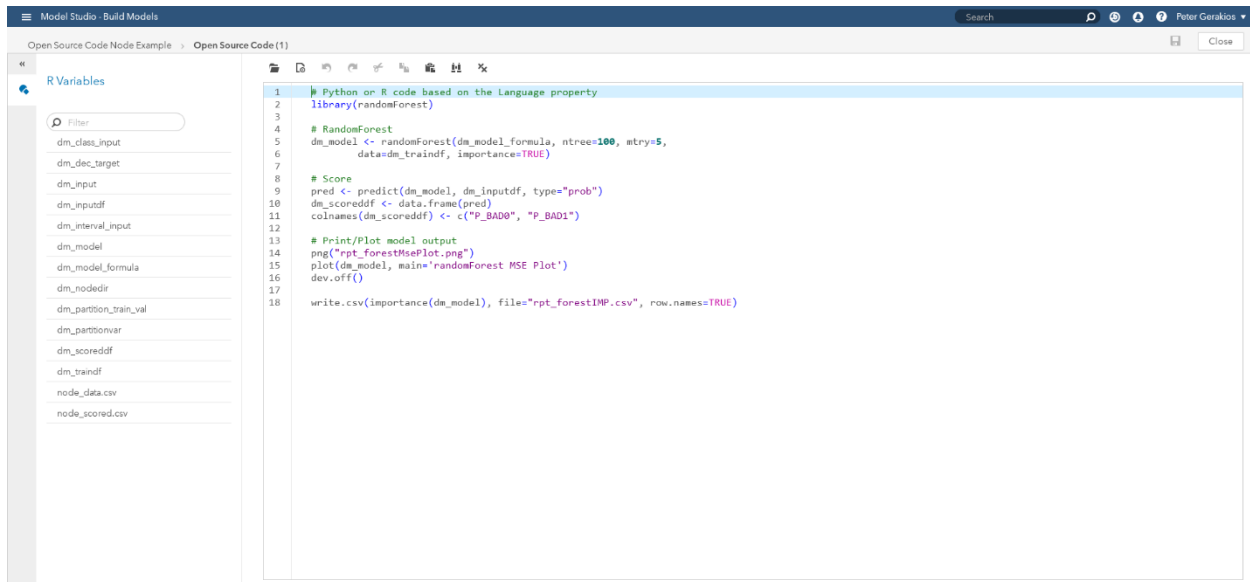
TIPS FOR CUSTOM CODE NODES

These tips help you add your own custom code to a pipeline. This means that you can apply any customizations that you find necessary.

FAVORITE FEATURE #5: THE OPEN SOURCE CODE NODE

The Open Source Code node, which is included in the Miscellaneous group of nodes, gives you the ability to run R or Python code in a Model Studio pipeline. You can place the node anywhere in a pipeline, and it does not need to be used for supervised learning. However, you can create a predictive model with R or Python code and move that Open Source Code node into the Supervised Learning lane. When you move the Open Source Code node to the Supervised Learning lane, it is treated as a modeling node, meaning that you receive the model assessment and model comparison results alongside those of SAS models. Because your R or Python code is not executed in CAS, a sample of the data is created and downloaded to the compute server as a CSV file. If you choose, the input data can be read into a data frame.

You will find a GitHub repository with examples of using the Open Source Code node at the following location: https://github.com/sassoftware/sas-viya-dmml-pipelines/tree/master/open_source_code_node. Included in the repository is an example of the R package `randomforest` that contains the code to name the prediction columns so that the model can be assessed and plots can be created.



Display 5. Open Source Code Editor

Display 5 shows the Code Editor before any code has been added. You can type or paste your code in the panel on the right. On the left is a list of variables that are available to use in your code. For example, instead of listing all your input variables, you can use the `dm_input` list in your code to represent all the input variables. Additionally, you can use the `dm_dec_target` variable to represent your target variable instead of hard-coding it.

The following code creates a random forest model in R:

```
# Python or R code based on the Language property
library(randomForest)

# RandomForest
dm_model <- randomForest(dm_model_formula, ntree=100, mtry=5,
                          data=dm_traindf, importance=TRUE)

# Score
pred <- predict(dm_model, dm_inputdf, type="prob")
dm_scoreddf <- data.frame(pred)
colnames(dm_scoreddf) <- c("P_BAD0", "P_BAD1")

# Print/Plot model output
png("rpt_forestMsePlot.png")
plot(dm_model, main='randomForest MSE Plot')
dev.off()

write.csv(importance(dm_model), file="rpt_forestIMP.csv", row.names=TRUE)
```

Besides calling the randomforest package, it creates the predictions (two columns of posterior probabilities, because this is a binary target) and names them P_BAD0 and P_BAD1. These are the names that Model Studio needs in order to assess the model. The MSE plot is included as a PNG image, and the variable importance table is saved to a CSV file. Both are displayed in the node's results, shown in Display 6, because they contain the rpt_ prefix.

The screenshot displays the 'Open Source Code Results' node in Model Studio. It contains four main sections:

- Summary:** A plot titled 'randomForest MSE Plot' showing Mean Squared Error (MSE) on the y-axis (ranging from 0 to 2) against the number of trees on the x-axis (ranging from 0 to 100). The plot shows a green line for training error and a red line for test error, both decreasing as the number of trees increases.
- Output Data:** A table titled 'rpt_forestIMP.csv' showing variable importance metrics for 10 variables. The columns are VARI, 0, 1, MeanDecreaseAcc..., and MeanDecreaseGini.
- R code:** The R script used to create the model, including library loading, model training, prediction, and saving of results.
- R output:** The execution output of the R code, showing the version of the randomForest package (4.6-14) and the null device output.

| VARI | 0 | 1 | MeanDecreaseAcc... | MeanDecreaseGini |
|-------------|---------|---------|--------------------|------------------|
| IMP_DELIHQ | 29.6390 | 25.5152 | 34.9364 | 120.5452 |
| IMP_DEROG | 21.3476 | 9.7853 | 21.5348 | 59.5721 |
| IMP_JOB | 17.6037 | 2.4475 | 16.3608 | 42.5148 |
| IMP_NINQ | 25.2360 | 8.4527 | 26.0458 | 63.9370 |
| IMP_REASON | 8.4671 | 3.6451 | 8.6791 | 10.3042 |
| IMP_CLAGE | 20.3451 | 15.9458 | 23.2921 | 107.8610 |
| IMP_CLNO | 28.1194 | 5.4346 | 26.2112 | 75.2657 |
| IMP_DEBTINC | 51.9656 | 33.5742 | 54.5498 | 317.0742 |

Display 6. Open Source Code Node Results

FAVORITE FEATURE #6: THE SAS CODE NODE

Similar to the Open Source Code node is the SAS Code node, which gives you the ultimate flexibility to do anything not already covered by a dedicated Model Studio node. As with the Open Source Code node, you can place the SAS Code node anywhere in a pipeline, and it can be moved into the Supervised Learning lane. When the SAS Code node is in the Supervised Learning lane, it is treated as a modeling node, meaning that you receive the model assessment and model comparison results alongside those of all other models.

For example, you can use the SAS Code node to programmatically change metadata. That is, you can use a SAS procedure to calculate the skewness of a variable, determine whether its value is above a certain threshold, and set the Log transformation as the desired transformation. You can create your own score code to perform any kind of transformation or feature engineering that you want. You can create your own plots to explore or summarize your data.

You will find a GitHub repository with examples of using the SAS Code node at the following location: https://github.com/sassoftware/sas-viya-dmml-pipelines/tree/master/sas_code_node.

The code editor for the SAS Code node includes SAS macros and macro variables that you can use in your code to represent various elements, such as data sets, CAS tables, or a list of variables. The macros and macro variables available here are similar to what you would find in the SAS Enterprise Miner code editor. In addition, the SAS Code node's code editor is color-coded for SAS code and can autocomplete code statements.

The following code sets the transformation of skewed variables to Log:

```
proc cardinality data=&dm_data outcard=&dm_datalib..outcard;
    var %dm_interval_input;
run;

filename deltac "&dm_file_deltacode";

data skewed;
    file deltac;
    set &dm_datalib..outcard end=eof;

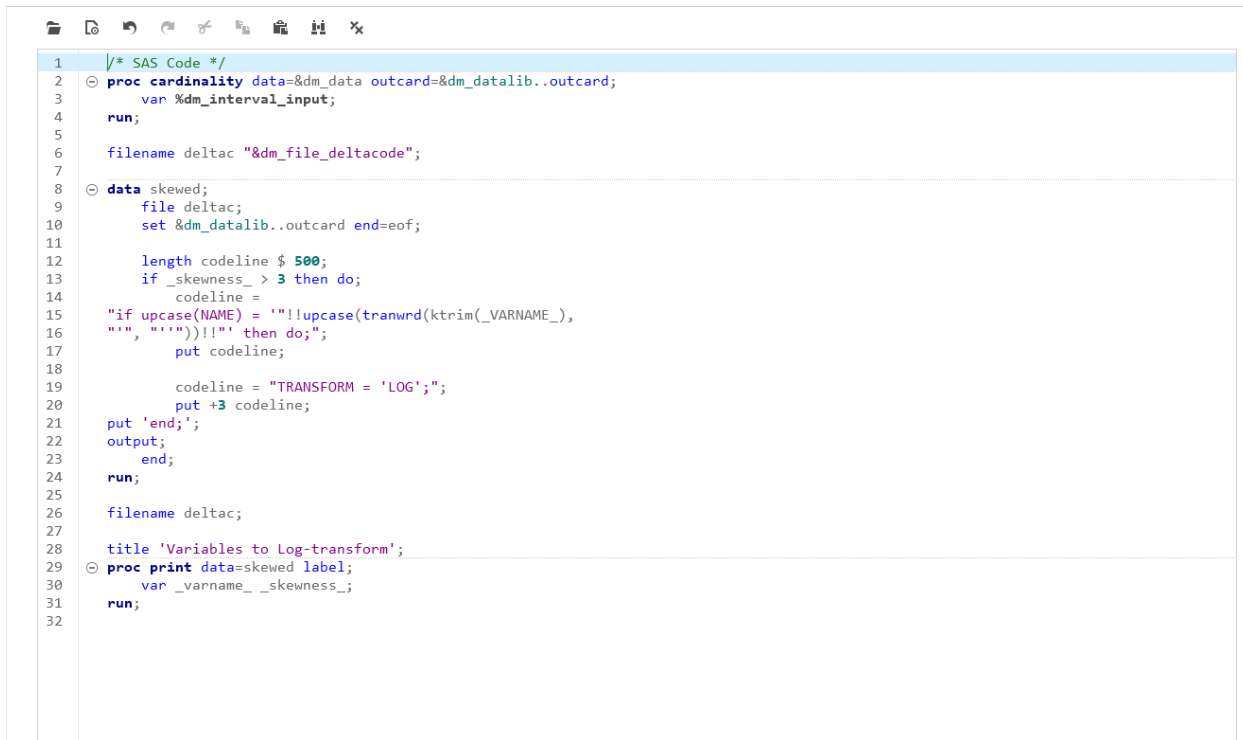
    length codeline $ 500;
    if _skewness_ > 3 then do;
        codeline =
"if upcase(NAME) = '!!upcase(tranwrd(ktrim(_VARNAME_),
'", "''))!!' then do;";
        put codeline;

        codeline = "TRANSFORM = 'LOG'";
        put +3 codeline;
        put 'end;';
        output;
    end;
run;

filename deltac;

title 'Variables to Log-transform';
proc print data=skewed label;
    var _varname_ _skewness_;
run;
```


Notice that in Display 7, the formatting and color-coding are applied to the SAS code.



```
1  /* SAS Code */
2  proc cardinality data=&dm_data outcard=&dm_datalib..outcard;
3      var %dm_interval_input;
4      run;
5
6      filename deltac "&dm_file_deltacode";
7
8  data skewed;
9      file deltac;
10     set &dm_datalib..outcard end=eof;
11
12     length codeline $ 500;
13     if _skewness_ > 3 then do;
14         codeline =
15         "if upcase(NAME) = "!!upcase(tranwrd(ktrim(_VARNAME_),
16         " ", ""))!!" then do;";
17         put codeline;
18
19         codeline = "TRANSFORM = 'LOG'";
20         put +3 codeline;
21     put 'end';
22     output;
23     end;
24     run;
25
26     filename deltac;
27
28     title 'Variables to Log-transform';
29     proc print data=skewed label;
30         var _varname_ _skewness_;
31     run;
32
```

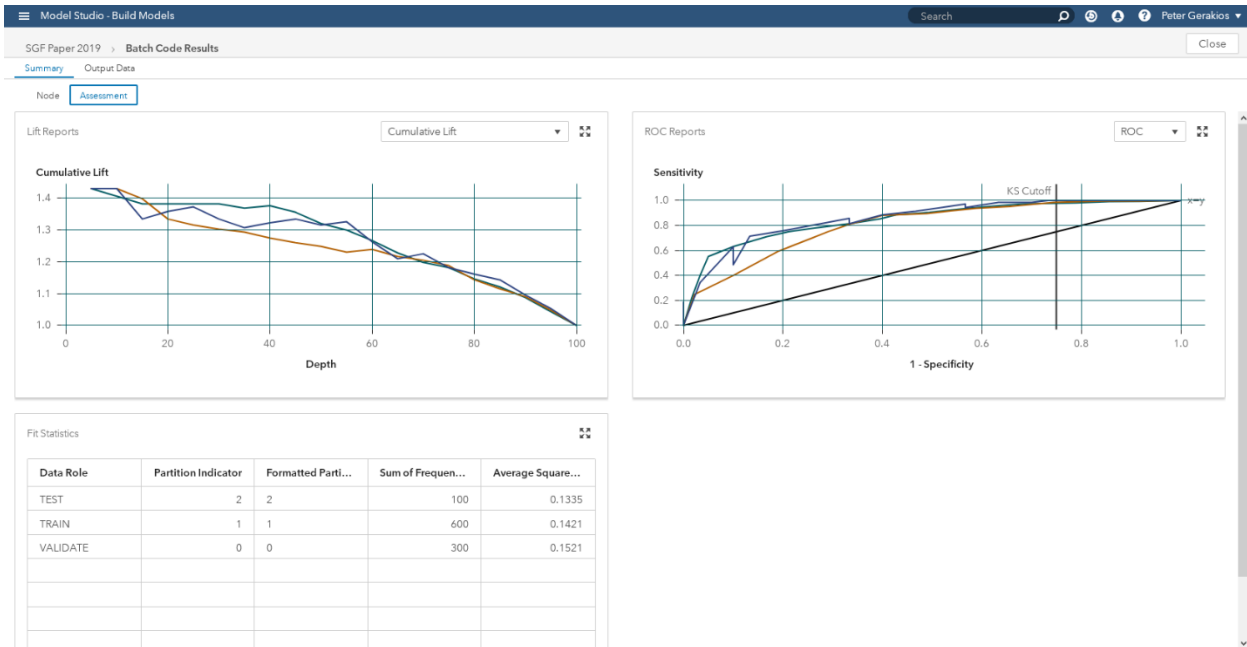
Display 7. SAS Code Editor

FAVORITE FEATURE #7: THE BATCH CODE AND SCORE CODE IMPORT NODES

The Batch Code and Score Code Import nodes enable you to import SAS Enterprise Miner and other models into Model Studio. After you import them, you can assess and compare these models against any other models that you build in Model Studio.

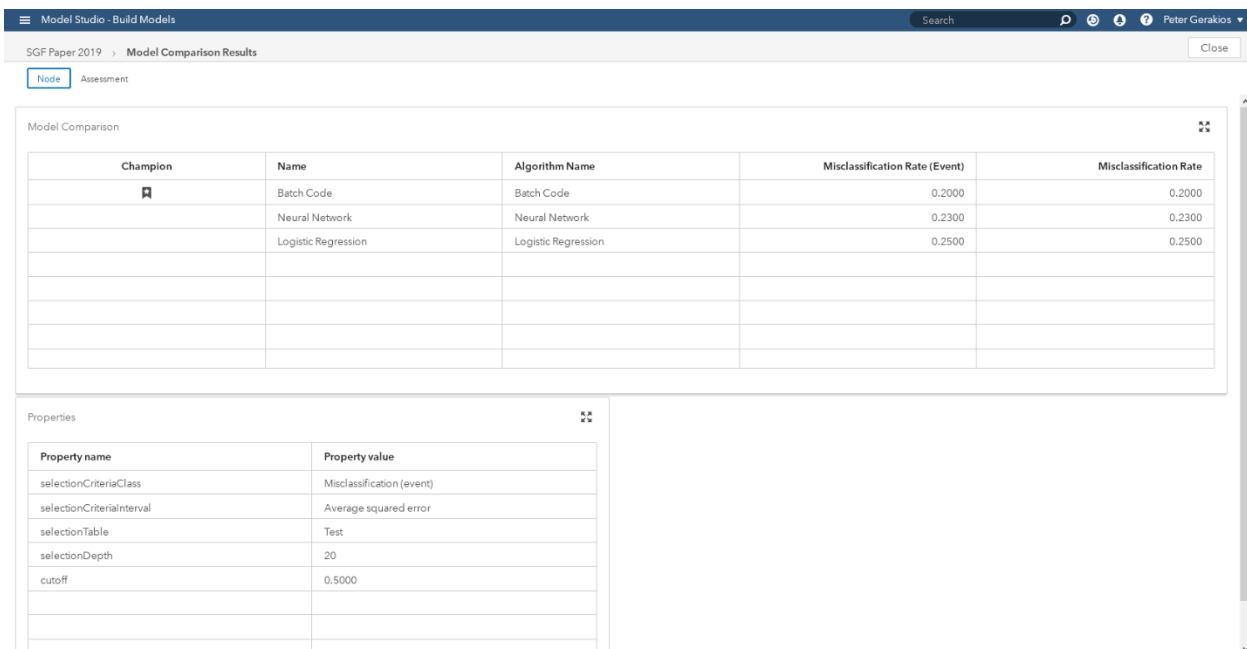
The Batch Code node enables you to import batch processing code for models that were created in SAS Enterprise Miner. Because SAS Enterprise Miner runs on SAS 9.4 and, unlike SAS Viya, does not operate on data in memory, the Batch Code node uses a sample of the input data that is based on the properties that you specify. This sample is provided to the SAS 9.4 client, and the batch code that represents the SAS Enterprise Miner process flow diagram is run on the sample. After the batch code runs, Model Studio retrieves the model score code files and uses that information to score the project data. The scored data are used to produce the assessment results.

The Score Code Import node enables you to import external models—those created outside Model Studio—that are saved as SAS score code. These models can be in the form of a single SAS DATA step code file or a SAS analytic store (ASTORE) file, which is a binary representation of your model. The analytic store file must also be accompanied by embedded process (EP) score code. When the node runs, the score code is used to score the project data, and the model is assessed and compared against other models that you build in Model Studio.



Display 8. Batch Code Assessment Results

In Display 8, you can see the assessment statistics that are included in the results of the Batch Code node—and these are the same plots that you get with other Supervised Learning nodes and a nominal target. You can view the various measures of lift and gain across depths, and you can view the receiver operating characteristic (ROC) curves and related statistics. Not shown here is a table of other fit statistics that are included for each partition. The Model Comparison node even selects the Batch Code node as the champion for that pipeline if it has the best value of the selection statistic, as you can see in Display 9.



Display 9. Model Comparison Results

TIPS FOR MODEL TUNING

FAVORITE FEATURE #8: AUTOMATIC HYPERPARAMETER TUNING

Available in five of the Supervised Learning nodes, automatic hyperparameter tuning (autotuning) is an algorithmic search to determine the best model settings for your data. Hyperparameters are properties that affect the training process, and thus they affect the quality of the resulting predictive model. Examples of hyperparameters include learning rate, regularization parameters, and the number of trees in a forest.

To reduce the manual effort that is required in finding the hyperparameter values that give you the most accurate model, you can enable the Perform Autotuning property for certain modeling nodes. This ensures that the optimal values of the available hyperparameters are chosen automatically. You can control which hyperparameters to autotune, the range of values to tune across, and the initial hyperparameter value to try. Autotuning (Koch et al. 2017) seeks to minimize or maximize a chosen objective function (typically a measure of model error) by using search methods such as a genetic algorithm, random sampling, Latin hypercube sampling, Bayesian kriging, and grid search (available only in the latest release).

Display 10 shows the autotuning properties of the Gradient Boosting node. You can see that there are six hyperparameters that can be autotuned. There are also other options related to the search method, and then a subset of general autotuning options.

The image shows a configuration panel for the Gradient Boosting node. It is organized into several sections:

- Perform Autotuning:** A toggle switch that is turned on.
- Hyperparameters:** Six items, each with a right-pointing arrow and a toggle switch that is turned on:
 - L1 Regularization
 - L2 Regularization
 - Learning Rate
 - Number of Inputs per Split
 - Number of Trees
 - Subsample Rate
- Search Options:**
 - Search method:** A dropdown menu set to "Genetic algorithm".
 - Number of evaluations per iteration:** A text input field containing "10".
 - Maximum number of evaluations:** A text input field containing "50".
 - Maximum number of iterations:** A text input field containing "5".
- General Options:**
 - Validation method:** A dropdown menu set to "Partition".
 - Training data proportion:** A text input field containing "0.7".
 - Validation data proportion:** A label with no input field visible.

Display 10. Autotuning Properties of the Gradient Boosting Node

If you have done any autotuning, then the results of your node will contain the tables shown in Display 11. These tables display the best configuration of hyperparameters that was found and the other sets of values that were tried, along with their selection statistics.

| Parameter | Value |
|----------------------------|--------|
| Evaluation | 34 |
| Number of Trees | 99 |
| Number of Variables to Try | 10 |
| Learning Rate | 0.2227 |
| Sampling Rate | 0.8063 |
| Lasso | 6.0046 |
| Ridge | 8.3224 |
| Kolmogorov-Smirnov | 0.4936 |

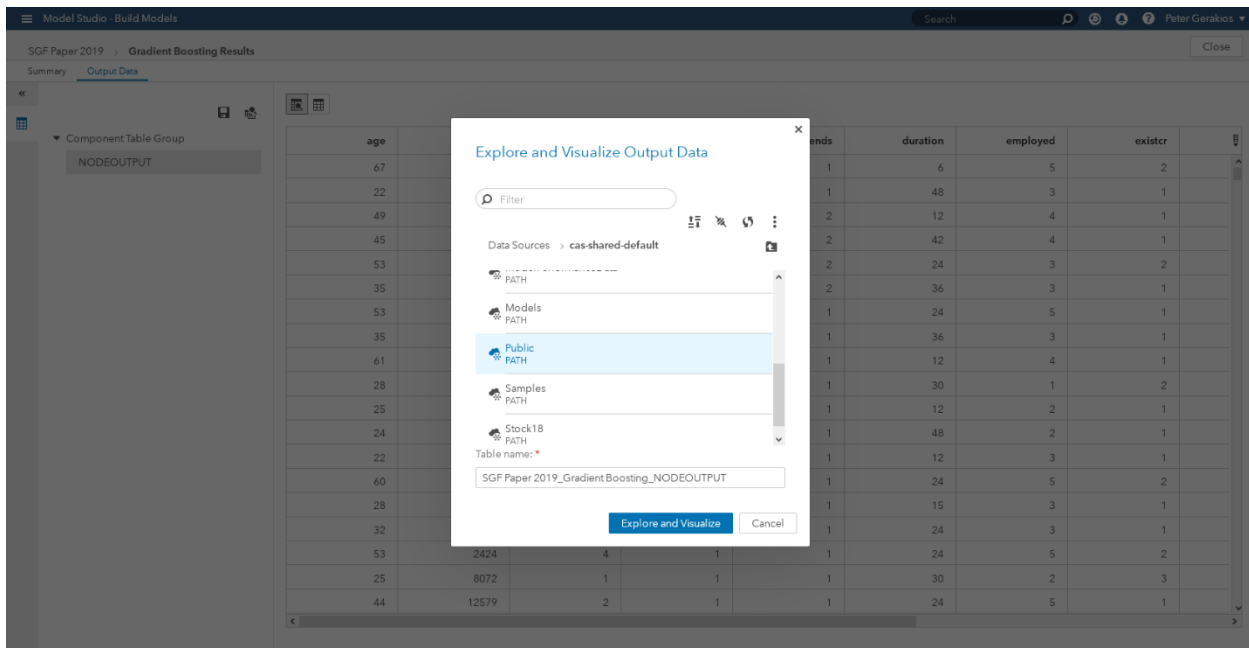
| Evaluation | Number of Trees | Number of Vari... | Learning Rate | Sampling Rate |
|------------|-----------------|-------------------|---------------|---------------|
| 0 | 100 | 20 | 0.1000 | 0.5000 |
| 34 | 99 | 10 | 0.2227 | 0.8063 |
| 18 | 98 | 10 | 0.2457 | 0.8143 |
| 39 | 98 | 10 | 0.2457 | 0.8143 |
| 44 | 98 | 10 | 0.2452 | 0.8141 |
| 46 | 98 | 10 | 0.2445 | 0.8139 |
| 45 | 99 | 10 | 0.2215 | 0.8059 |
| 6 | 78 | 7 | 0.7800 | 1 |

Display 11. Autotuning Results of the Gradient Boosting Node

TIPS FOR VIEWING AND DEPLOYING RESULTS

FAVORITE FEATURE #9: EXPLORE AND VISUALIZE OUTPUT DATA

New to the latest release of Model Studio is the ability to explore and visualize the data that are exported from each node, as shown in Display 12. Again, SAS Enterprise Miner users are probably familiar with this popular capability. The score code of the current node and all predecessor nodes is applied to the available data. This means that you can view columns generated by the score code, such as predicted values, residuals, and imputed or transformed variables. Instead of just looking at a table of values, you can take advantage of the exploration and visualization capabilities of SAS Visual Analytics.

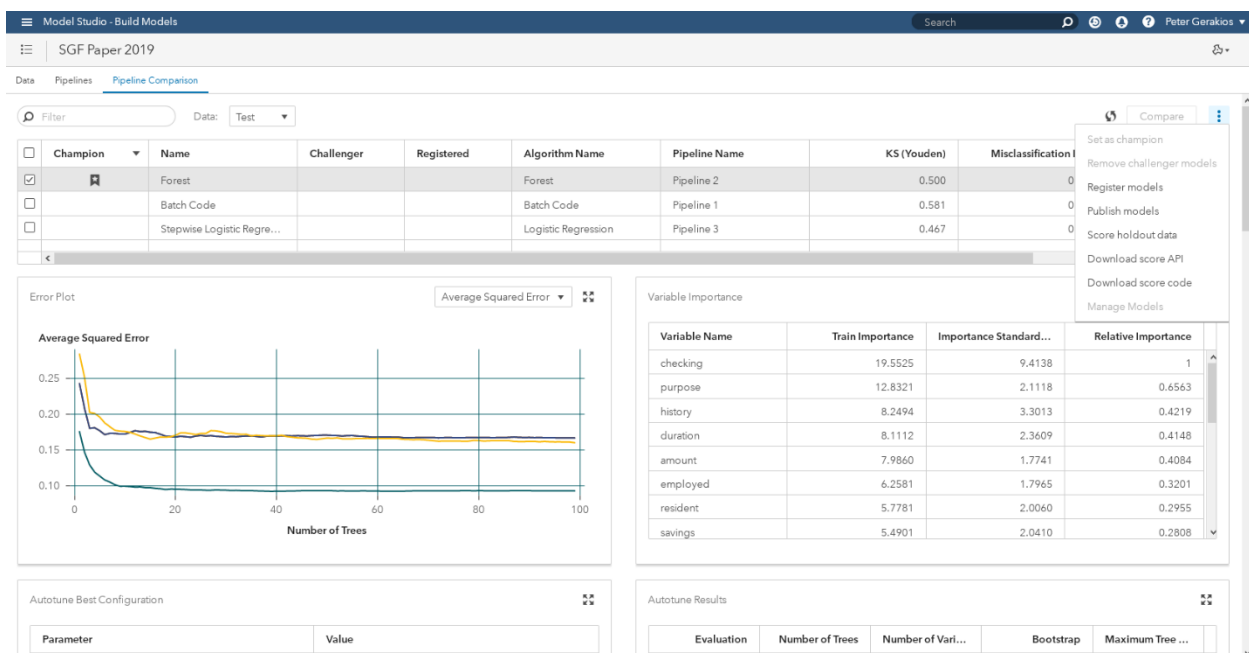


Display 12. Explore and Visualize Output Data Window

Inside SAS Visual Analytics, you can, for example, plot your predicted values against the actual target values with just a couple of clicks. When you're done exploring the data, you can seamlessly switch back to Model Studio and pick up where you left off.

FAVORITE FEATURE #10: SCORING AND DEPLOYMENT

Last, but definitely not least, are the many options available for scoring and deploying models, which is one of the most important parts of the analytics life cycle. You can download scoring APIs written in SAS, Python, or REST that can be embedded in your own applications. Models can be registered to SAS® Model Manager, where they are stored in the common model repository (CMR). After a model is placed in the CMR, you can monitor and test it, or retrain it if you notice model degradation. You can also publish models directly in databases such as Hadoop and Teradata. You can download the score code for use in a SAS program. And finally, you can score an additional holdout data set.



Display 13. Pipeline Comparison Tab

Each of these scoring or deployment methods is available on the **Pipeline Comparison** tab, shown in Display 13. This tab automatically includes the champion model from each pipeline and identifies the overall champion model for the project. You can add other challenger models to the tab, which lets you deploy that model in any of the previously mentioned methods.

BONUS TIPS

In creating this list, we were forced to leave some of our favorites on the cutting-room floor. Here are a few of the runners-up that deserve a special mention.

MODEL INTERPRETABILITY

All the Supervised Learning nodes now include options for model interpretability. This feature lets you see which inputs are most important and can explain the predictions at a cluster level. SAS is very interested in this area and is actively researching improvements.

DATA EXPLORATION

The Data Exploration node gives you options to display a three-dimensional, nonlinear projection of your interval inputs by using t -distributed stochastic neighbor embedding (t -SNE).

DETAILED NODE DESCRIPTIONS

Each node provides a detailed description of the node and its capabilities in the same pane as the node's properties.

VARIABLE SELECTION

The Variable Selection node can combine multiple variable selection techniques and will choose a winner on the basis of a selected voting method.

QUANTILE REGRESSION

The new Quantile Regression node models an interval target when you are interested in understanding what predicts the median of the target.

GLOBAL AND PROJECT SETTINGS

You can specify global settings that control several aspects of the model creation process. This includes specifying the model comparison rules, setting the default logging options, or configuring the metadata advisor rules that automatically apply variable level and role assignments.

TABBED RESULTS

SAS is always working to make viewing your results a better experience. Now that the results plots and tables are organized into tabs, you no longer need to scroll back and forth to find the report or plot that you want.

CONCLUSION

The Model Studio interface in SAS Visual Data Mining and Machine Learning offers a powerful environment for building pipelines. The highlights we have shared here just scratch the surface of the capabilities that it offers. Please check our SAS Data Mining and Machine Learning Community site often at https://communities.sas.com/t5/SAS-Data-Mining-and-Machine/bd-p/data_mining, where we post more tips, or you can share your own.

REFERENCES

Wujek, B., Haller, S., and Wexler, J. 2018. "Navigating the Analytics Life Cycle with SAS Visual Data Mining and Machine Learning on SAS Viya." In *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc. Available at <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2246-2018.pdf>.

Koch, P., Wujek, B., Golovidov, O., and Gardner, S. 2017. "Automated Hyperparameter Tuning for Effective Machine Learning." In *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available at <https://support.sas.com/resources/papers/proceedings17/SAS0514-2017.pdf>.

RECOMMENDED READING

- *Getting Started with SAS Visual Data Mining and Machine Learning in Model Studio*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Wendy Czika
SAS Institute Inc.
wendy.czika@sas.com

Peter Gerakios
SAS Institute Inc.
peter.gerakios@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.