Additional Nodes in Model Studio

Model Studio offers additional nodes that are not mentioned earlier in this course. Three of these nodes are described in detail below: the Save Data node, the SAS Code node, and the Open Source Code node.

For more information about other useful nodes, like the following, see the documentation for SAS Visual Data Mining and Machine Learning:

- Batch Code: The Batch Code node is a Supervised Learning node. It enables you to import external SAS models that are saved in batch code format.
- Score Code Import: The Score Code Import node is a Supervised Learning node that enables you to import external models that are saved as SAS score code.
- Ensemble: The Ensemble node is a Postprocessing node. It creates new models by combining the posterior probabilities (for class targets) or the predicted values (for interval targets) from multiple predecessor models.

Save Data Node

The Save Data node is a Miscellaneous node that enables you to save the training table that is produced by a predecessor node to a caslib. This table could be partitioned into training, validation, or test sets based on the project settings. In that case, the table contains the **_partind_** variable that identifies the partitions.

By default, the training table produced by a pipeline is temporary and exists only for the duration of the run of a node, and has local session scope. The Save Data node enables you to save that table to disk in the location associated with the specified output library. This table is then available to other applications for further analysis or reporting.

In the Output Library Project settings, you can specify the default output caslib where tables are to be saved. You can overwrite this location using the Output library property.

In addition, you can load the table in memory and promote the table to have global scope in the specified caslib. This enables multiple CAS sessions to access this table.

If you run the node, the results consist of an output table containing information about the saved table, including a list of variables and their basic attributes.

The properties of the Save Data node are as follows:

- **Output library** specifies the output caslib where the table will be saved on disk. Use **Browse** to navigate to the proper library. If the user has specified an output library under **Project Settings**, this library is used by default.
- **Table name** specifies the name for the CAS table being saved. The default value is **tmpSaveData**.
- **Replace existing table** specifies whether to override an existing CAS table with the same name when saving. By default, this option is deselected.
- **Promote table** specifies whether to load the table in memory and promote the table to global space. By default, this option is deselected.

After running the node, you can open the Results window, which has two tabs: Properties and Output.

- The Properties tab specifies the properties that were selected before running the node. These include the output library, the table name, whether to replace or promote the table, and the CAS session ID.
- The Output tab displays the SAS output of the saved data run.

SAS Code Node

The SAS Code node is a Miscellaneous node that enables you to incorporate new or existing SAS code into Model Studio pipelines. The node extends the functionality of Model Studio by making other SAS procedures available for use in your data mining analysis. You can also write SAS DATA steps to create customized scoring code, conditionally process data, or manipulate existing data sets. The SAS Code node is also useful for building predictive models, formatting SAS output, defining table and plot views in the user interface, and for modifying variables' metadata. The node can be placed at any location within a pipeline (except after the Ensemble or Model Comparison nodes). By default, the SAS Code node does not require data. The exported data that are produced by a successful SAS Code node run can be used by subsequent nodes in a pipeline.

To indicate that the SAS Code node produces a model that should be assessed, right-click the **SAS Code** node and select **Move > Supervised Learning**. If a SAS Code node that is marked as a Supervised Learning node does not generate any score code, either as DS1 or as an analytical store (astore), then no assessment reports nor model interpretability reports are generated. If the node produces score code that does not create the expected predicted or posterior probability variables, then the node will fail.

The properties of the SAS Code node are as follows:

- **Code editor** invokes the SAS Code Editor.
- **Train only data** specifies whether the node should receive the training observations only if the data are partitioned. By default, this option is deselected. Currently, this property is unavailable for this node. To specify that the node receive only

training data, add the following WHERE clause to your code:

```
where &dm partitionvar.=1;
```

The Code Editor window is opened from a property in the properties panel. The Code Editor window enables the user to view a Macros table and a Macro Variable table from the left column, which contain a list of macros and macro variables, respectively, that are available to the SAS session.

Additional options are available as shortcut buttons on the top of the Editor window. These options enable you to do the following:

- browse
- control settings, which include general (such as showing line numbers and font size) and editing (such as enabling autocomplete and auto indention) code options
- undo and redo
- cut, copy, and paste
- find and replace
- clear all code

User-written code is saved through a Save shortcut button in the upper right corner of the Editor window.

SAS Viya users have access to more power than they might realize. All SAS Enterprise Miner and SAS/STAT procedures are included with a Visual Data Mining and Machine Learning license on SAS Viya. This means that by using the SAS Code node in a pipeline, users have access to the Enterprise Miner procedures that are specific to that product and to the entire suite of tools available with SAS®9 SAS/STAT software.

The in-memory CAS table would require being copied to a location accessible by these procedures in the form of a SAS data set.

Open Source Code Node

Open source in SAS Viya supports Python and R languages and requires Python or R and necessary packages to be installed on the same machine as the SAS Compute Server. It downloads data samples from SAS Cloud Analytic Services for use in Python or R code and transfers data by using a data frame or CSV file using the Base SAS Java Object.

The Open Score Code node enables you to import external code that is written in Python or R. The version of Python or R software does not matter to the node, so any version can be used as the code is passed along. The Python or Rscript executable must be in system path on Linux, or the install directories can be specified with PYTHONHOME or RHOME on Windows.

The Open Source Code node is used to run Python or R code in a pipeline.

The node enables the user to prototype machine learning algorithms that might exist in open source languages but have not yet been vetted to be included directly as a node in Model Studio. This node can subsequently be moved to a Supervised Learning group if a Python or R model needs to be assessed and included to be part of model comparison. The node can execute Python or R software regardless of their versions.

After selecting the language (Python or R) from properties, use the Open button to enter respective code in the editor. Because this code is not executed in CAS, a data sample (10,000 observations by default) is created and downloaded to avoid movement of large data. Use Data Sample properties to control the sample size and method. Apply caution and do not specify full data or a huge sample when the input data are large. When performing model comparison with other Supervised Learning nodes in the pipeline, note that this node might not be using full data.

Input data can be accessed by the Python or R code via a CSV (comma-separated-value) file or as a data frame. When **Generate data frame** is selected, a data frame is generated from the CSV file, and input data are available in dm_inputdf, which is a pandas data frame in Python or an R data frame. When data are partitioned, an additional data frame, dm_traindf, is also available in the editor. That frame contains training data. If a Python or R model is built and needs to be assessed, corresponding predictions or posterior probabilities should be made available in the **dm_scoreddf** data frame. To do so, right-click and select **Move > Supervised Learning** to indicate that model predictions should be merged with input data and model assessment should be performed. Note that the number of observations in dm_inputdf and dm_scoreddf should be equal for successful merge to occur.

Note that this node cannot support operations such as **Download score code**, **Register models**, **Publish models**, and **Score holdout data** from the Pipeline Comparison tab because it does not generate SAS score code.

The properties of the Open Source Code node are as follows:

- **Code editor** invokes the SAS Code Editor.
- **Language** specifies the open source language to be used. Available options for this property are R and Python. The default setting is R.
- **Generate data frame** specifies whether to generate an R data frame or a pandas data frame in Python. In addition, categorical inputs are encoded as factors in R. If this option is disabled, the input data should be accessed as a CSV file. By default, this option is enabled.
- **Data Sample** controls sampling of the data. By default, this property is collapsed. Thee Data Sample property has been expanded in the screen capture above. When expanded, the subcategories are shown. The subcategories are as follows:
    - **Sampling Method** specifies the sampling method. When the input data has a partition variable or a class target (or both), the sample is stratified using them. Otherwise, a simple random sample is used. The available settings are None, Simple Random and Stratify. The default setting is Stratify.

- **Sample using** specifies whether to sample using the number of observations or the percent of observations from input data. The available settings are Number of observations and Percent of Observations. The default setting is Number of observations.
- **Number of Observations** or **Percent of Observations** depends on the setting for the Sample using property. When **Sample using** is set to **Number of Observations**, this property specifies the number of observations to sample from input data. The default in this case is 10,000, and the user can enter numeric values manually. When **Sample using** is set to **Percent of Observations**, this property specifies the percent of observations to sample from input data. In this case, a slide bar appears that has values ranging from 1 to 100. The default setting is 10.
- **Include SAS formats** specifies whether to include SAS formats in input data to downloaded CSV files, when passing data to open source software. By default, this option is enabled.

Like the SAS Code node, for the Open Source Code node, the Code Editor window is opened from a property in the properties panel. The Code Editor window enables the user to view a list of R variables or Python variables (depending on which open source language is used) that are available to the editor session.

Additional options are available as shortcut buttons that are identical to those described earlier for the SAS Code node.