Paper SAS2246-2018

Navigating the Analytics Life Cycle with SAS® Visual Data Mining and Machine Learning on SAS® Viya®

Brett Wujek, Susan Haller, and Jonathan Wexler, SAS Institute Inc.

ABSTRACT

Extracting knowledge from data to enable better business decisions is not a single step. It is an iterative life cycle that incorporates data ingestion and preparation, interactive exploration, application of algorithms and techniques for gaining insight and building predictive models, and deployment of models for assessing new observations. The latest release of SAS® Visual Data Mining and Machine Learning on SAS® Viya® accommodates each of these phases in a coordinated fashion with seamless transitions and common data usage. An intelligent process flow (pipeline) experience is provided to automatically chain together powerful machine learning methods for common tasks such as feature engineering, model training, ensembling, and model assessment and comparison. Ultimate flexibility is offered through incorporation of SAS® code into the pipeline, and collaboration with teammates is accomplished using reusable nodes and pipelines. This paper provides an in-depth look at all that this solution has to offer.

INTRODUCTION

With the ubiquity of data these days, companies are racing to ensure that they can apply analytics to derive the insight necessary to provide better products and services, and ultimately to keep pace with, or surpass, the competition. They know they need to "do machine learning," but they frequently don't really know what that entails. Their focus often turns directly to applying the powerful modeling algorithms to their data, resulting in individual eureka moments but neglecting the numerous phases of transforming data into business value in a sustainable manner.

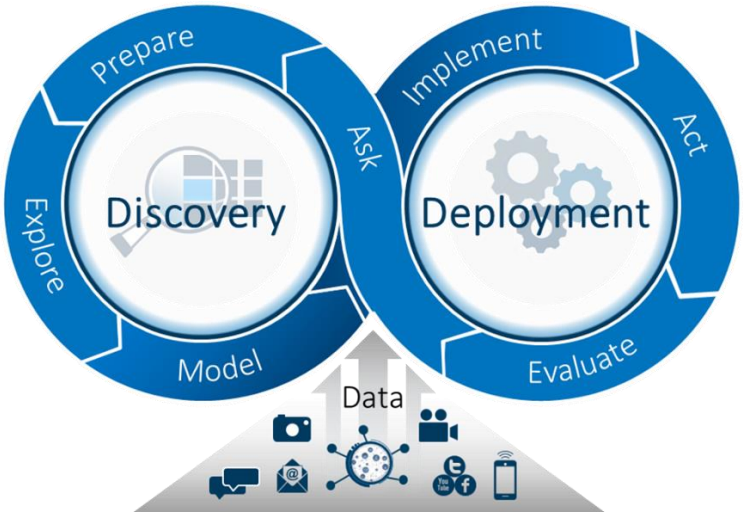

Figure 1. Phases of the Analytics Life Cycle

These important phases make up what is referred to as the analytics life cycle, as illustrated in Figure 1, which consists of the following:

- **Data ingestion:** consuming, merging, and appending data from potentially multiple data sources and formats
- **Data preparation:** cleaning, transforming, aggregating, and creating columns as necessary and appropriate to address the specified business problem
- **Exploration:** profiling, analyzing, and visualizing your data to gain initial insight and understanding of variable distributions and relationships

- **Modeling:** exercising feature engineering techniques, applying algorithms to identify segments and build representations for classifying new observations and making predictions, and assessing and tuning the generated models
- **Model deployment:** selecting champion models and promoting them for use in a production environment to aid in making effective business decisions
- **Model management:** maintaining a version-controlled repository of models, incorporating them into decision-making processes, monitoring their performance over time, and updating them as necessary to ensure that they are adequately and accurately addressing your business problem

Implementing and adhering to a process that accommodates the entire analytics life cycle is a significant undertaking, but a necessary one. Certainly, the public marketplace of analytics packages in open-source languages provides access to an ample supply of algorithms and utilities for data manipulation, exploration, and modeling. But typical business environments require more than individuals working on machine learning applications in silos and using a scattered collection of tools with little governance, lack of data and results lineage, inconsistent formats, collaboration bottlenecks, and hurdles to deployment. In the remainder of this paper, you will see how SAS Visual Data Mining and Machine Learning provides a comprehensive framework of capabilities to navigate this analytics life cycle through a seamless integration of interfaces that focus on each of the aforementioned phases, built on the foundation of SAS Viya. A case study that uses SAS Visual Data Mining and Machine Learning to address the problem of telecommunications customer attrition is presented in the Appendix.

THE FOUNDATION: SAS VIYA

An environment for end-to-end analytics relies on a solid foundation that can provide common access to data, analytics, and results in an efficient, consistent, and open manner. For SAS Visual Data Mining and Machine Learning, that foundation is provided by SAS Viya. SAS Viya is an extension of the SAS platform that offers a distributed, in-memory data access layer in which analytic "actions" can be performed in an efficient distributed and parallel manner through the SAS® Cloud Analytics Services (CAS) execution engine. Figure 2 illustrates the architecture, which is specifically designed to serve as an extensible and open framework in which data can be accessed from a variety of common sources and actions can be invoked in a language-agnostic fashion, and upon which custom and domain-specific applications can be established to exploit the in-memory efficiency and simple and common accessibility of data, actions, and results.

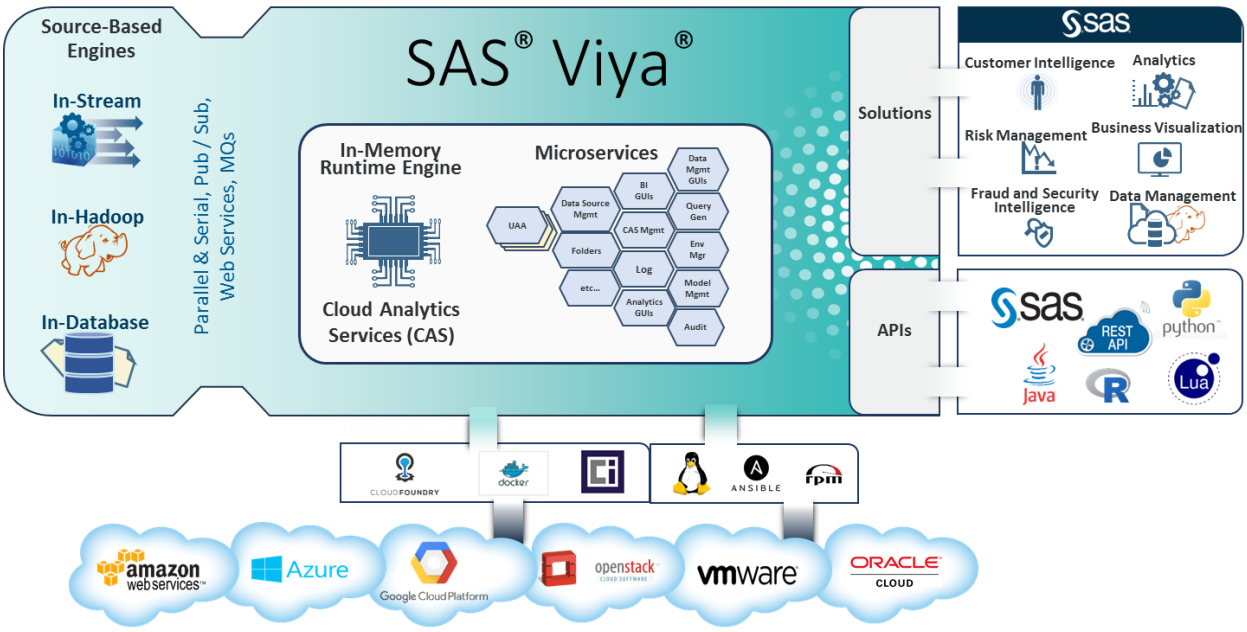

Figure 2. SAS Viya: An Extension of the SAS Platform

SAS Visual Data Mining and Machine Learning is one such application; it assembles a collection of data preparation and modeling actions that are presented through integrated interfaces that are specially designed for each phase of the analytics life cycle, as shown in Figure 3. When your work in one phase is complete, you can directly progress to the next phase, avoiding any hassle (and error-prone process) of transferring (and possibly translating) your data or results, or of launching new applications independently. Because the analytics are performed by invoking actions in CAS, the data preparation and modeling functions can also be executed by writing programs in SAS or other languages for which an API wrapper has been written (Python, R, Java, Lua, and REST). A good example of how you can work on a particular machine learning application across multiple interfaces and programming languages is offered in Wexler, Haller, and Myneni (2017). This is all made possible by SAS Viya providing the common data access layer and open access to a consistent set of analytics actions.

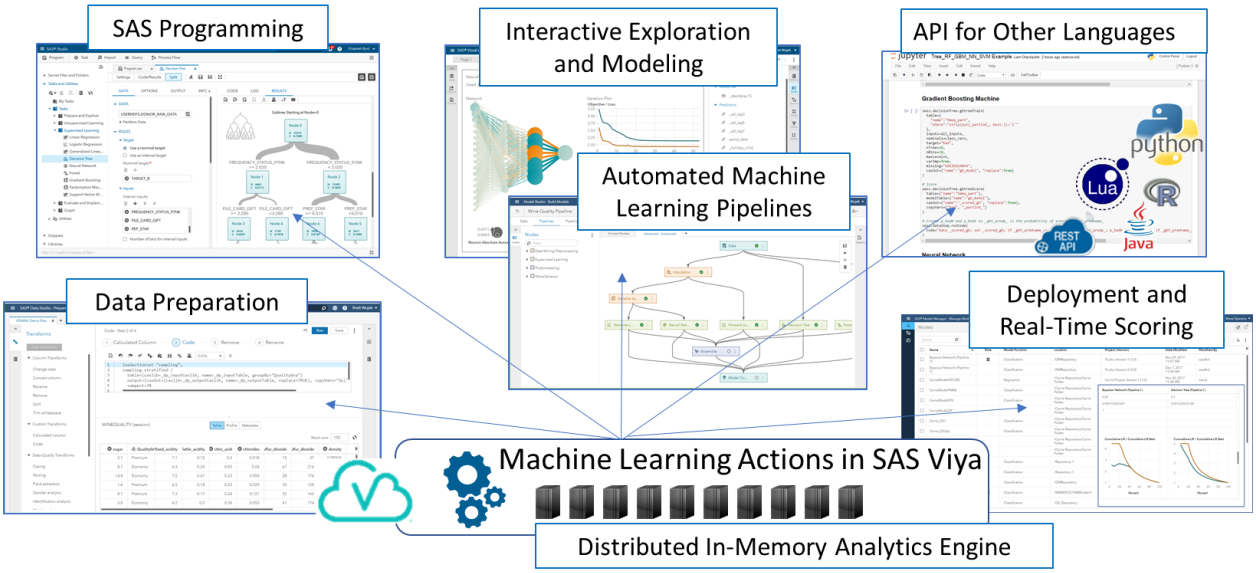

Figure 3. SAS Visual Data Mining and Machine Learning Capabilities and Interfaces

One means of employing the capabilities that comprise SAS Visual Data Mining and Machine Learning is through an integrated collection of actions in a unified web interface that is designed specifically to facilitate the end-to-end analytics life cycle, as depicted in Figure 4. The remainder of this paper navigates through the analytics life cycle with SAS Visual Data Mining and Machine Learning via the user interfaces that are associated with these actions.

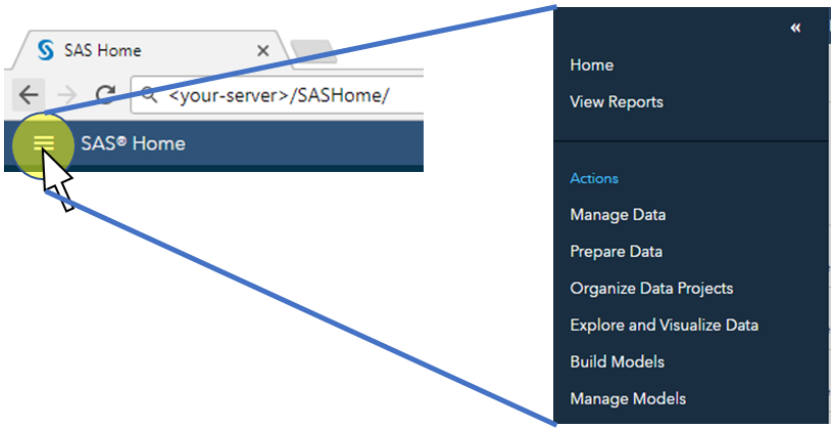

Figure 4. Menu of Actions to Access SAS Visual Data Mining and Machine Learning Capabilities

DATA INGESTION AND PREPARATION

Machine learning applications should be developed and evolve as solutions to well-defined business problems. That is, assuming you have (or can get) the necessary data, what are the most important questions you would like answered to add value to your organization? This is the “Ask” phase of the analytics life cycle shown in Figure 1, and it goes hand-in-hand with collecting the requisite data, identifying the necessary analytical operations, and ensuring that your data are in an appropriate form for these analytics. Although a software platform cannot resolve the “Ask” for you, it *can* support it by the accommodations it provides for ingesting and preparing data for the desired analytical operations in the “Prepare” phase of the analytics life cycle, as shown in Figure 5.

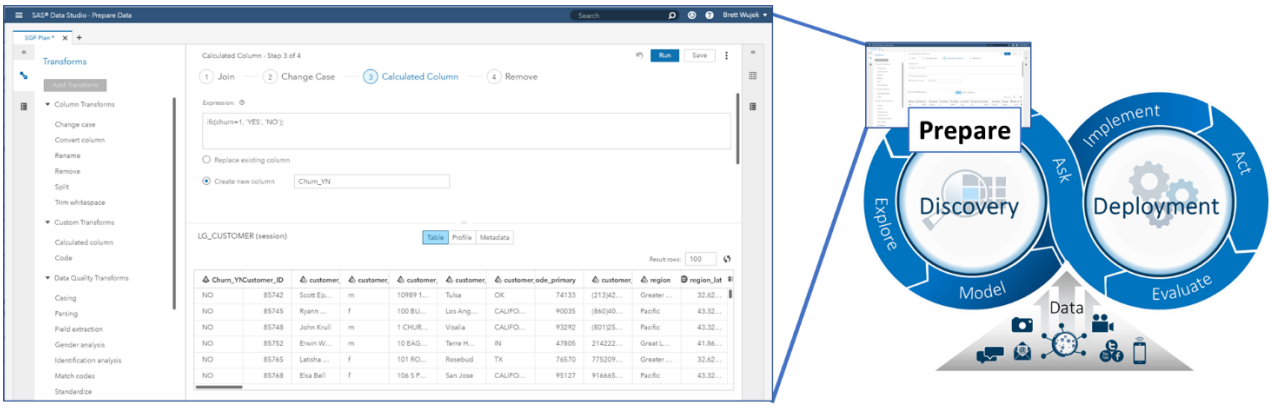

Figure 5. Data Preparation Using SAS® Data Studio in SAS Visual Data Mining and Machine Learning

CONSUMING DATA FROM VARIOUS SOURCES

SAS Visual Data Mining and Machine Learning provides built-in conveniences for browsing available data and importing data from various sources as necessary. A common data browser is used in all interfaces wherever a data table needs to be selected (see Figure 6). For extended data management capabilities, an enhanced form of the data browser can be added to your environment, offered as a dedicated Data Explorer interface, which is accessible from the **Manage Data** action in the actions menu.

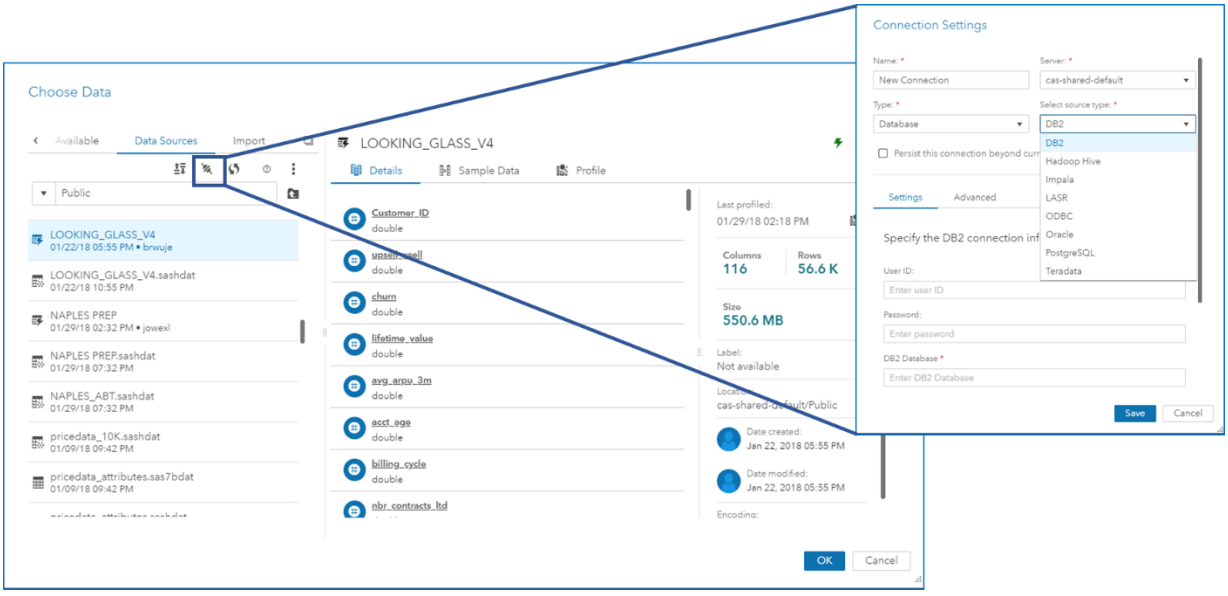

Figure 6. Browsing and Loading Data

To prepare, explore, and perform analytics on data in SAS Viya, the data must be loaded into memory as a CAS table. The data browser displays data tables that are available to use immediately (data sets that have been loaded into CAS tables), denoted by the icon next to the table name. Data can be made available by defining connections to new **Data Sources** by clicking the "Connect" button (), and referencing data sets that reside in those data sources. SAS Viya supports several types of data sources by using data connectors (depending on SAS/ACCESS® licensing), including the following:

- **File system:** DNFS, HDFS, Path
- **Database:** DB2, Hadoop Hive, Impala, LASR, ODBC, Oracle, PostgreSQL, Teradata

Once a data source is defined, a CAS library (caslib) serves as a reference to it and is presented for you to browse available tables. Although you can browse *all* tables (including SAS data sets) that reside in the data source, you can select only tables that are loaded as in-memory CAS tables to use within the application.

Tip: *To load a data set that resides in a specified data source but is not yet loaded (that is, it has an icon other than next to it), right-click it and select* ***Load****.*

You can also import data from local files, such as the commonly used CSV (comma-separated values) format or other text files that contain data in a tabular format, or directly from social media feeds such as Twitter, Facebook, Google, and YouTube. The main thing to keep in mind is that a data set must be loaded into memory as a CAS table before you can work with it.

For a selected table, the data browser displays all the column names along with their corresponding data types, in addition to information about the size of the table, as shown in Figure 6. You can run a profile of the table to get an initial indication of the cardinality, number of missing values, and basic descriptive statistics for each variable. The profile provides some insight as to what type of data preparation you might need to exercise before applying analytical operations on or modeling the data.

TRANSFORMING AND ENHANCING YOUR DATA

Data preparation is such an important and necessary step in machine learning applications (Wujek, Hall, and Gunes 2016) that you will find capabilities to transform and augment your data in various forms throughout different interfaces in SAS Visual Data Mining and Machine Learning. Often, interactive visual inspection of distributions and other aspects of the data is necessary in order to understand which analytical transformations are required, and other specialized forms of data manipulation, such as feature engineering techniques, are more closely associated with the model building phase. For data preparation to be done in a systematic and repeatable fashion so that it can be applied consistently to new data in the future, SAS Visual Data Mining and Machine Learning provides a powerful and convenient interface, SAS Data Studio, for preparing your data. SAS Data Studio enables you to build a data plan that consists of a sequence of well-defined, repeatable steps that apply transforms to the source data table that is loaded. These transforms are organized in the following categories:

- **Column Transforms** to modify the values in existing columns in common ways
- **Row Transforms** to filter rows on the basis of variable values or to create columns through transposition
- **Multi-input Transforms** to join/merge or append tables
- **Data Quality Transforms** to standardize values and apply common data cleansing operations by using a SAS® Quality Knowledge Base
- **Custom Transforms** to calculate new columns by using simple expressions or custom code

The out-of-the-box transforms provide a convenient way to quickly transform (and clean) the values in columns and create new columns through aggregation and calculations. For any data preparation actions that are not directly available as transforms, the **Code** transform (one of the **Custom Transforms**) provides ultimate flexibility by enabling you to write SAS DATA step or CASL code to prepare your data as necessary.

Tip: *When writing code for the **Code** transform in order to prepare data, you must use the variables _dp_inputCaslib, _dp_inputTable, _dp_outputCaslib, and _dp_outputTable to refer to the input and output tables.*

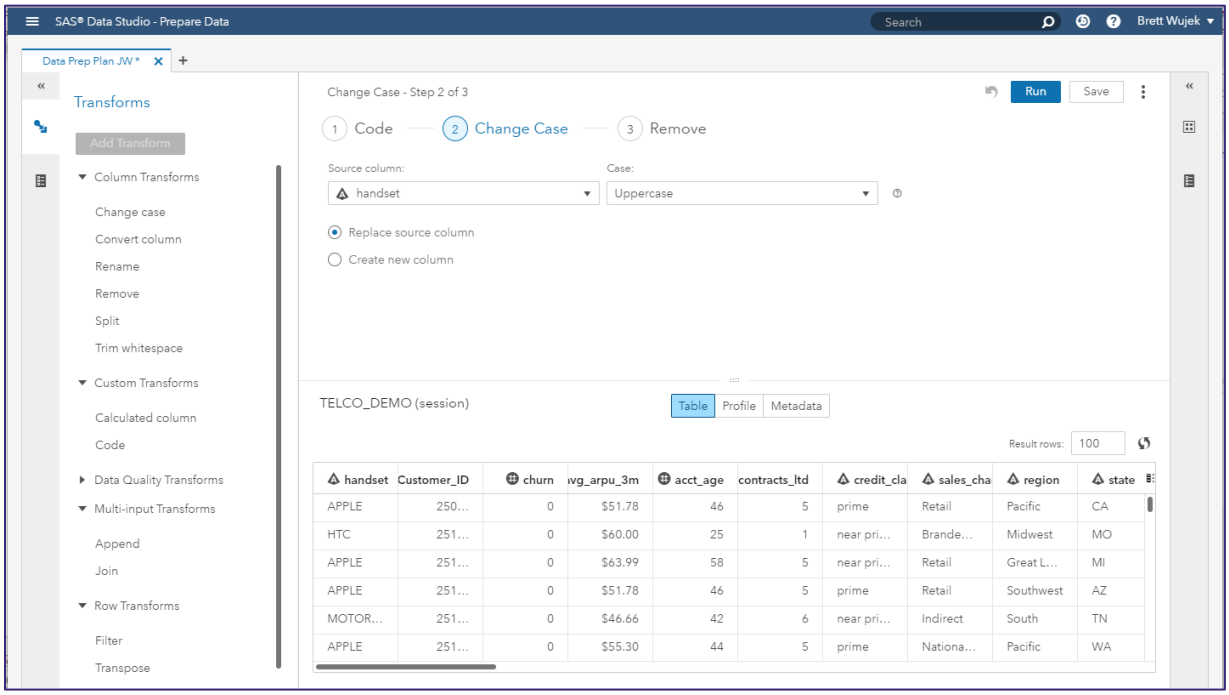

Figure 7. Building and Applying a Data Preparation Plan in SAS Data Studio

Each transform that is added as a step in the data plan must be defined and then run so that an updated version of the table is available for a subsequent step. The plan maintains a reference to the unaltered source data table while it creates and updates a new table as a result of applying the steps. Information about the source table can be seen on the left, and information about the result table can be viewed on the right. Profiles of the source and result tables can be run to view information about the variables.

Tip: *If the result table is not as expected or desired, you can roll back the list of steps from last to first by clicking the undo button .*

USING YOUR PREPARED DATA

A major advantage of SAS Visual Data Mining and Machine Learning is the ability to seamlessly navigate from one phase of the analytics life cycle to another. Once you have defined the data preparation plan, you can save the plan so that it can be applied to new data tables, and you can use the actions menu ( ) to progress directly to other phases that will use the result table, as shown in Figure 8. To continue navigating through the analytics life cycle, you can select **Explore and Visualize Data** to get a good sense of the nature of your data and the relationships among the variables.

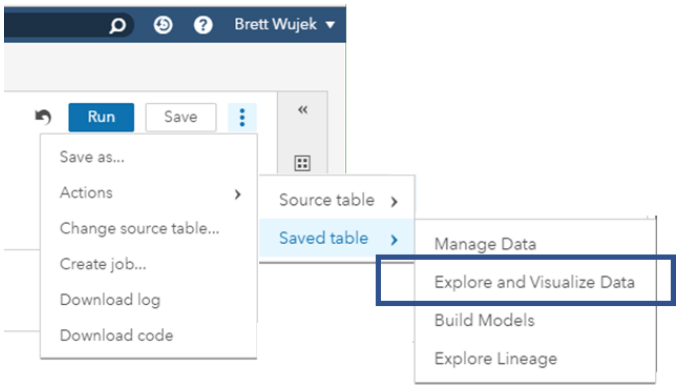

Figure 8. Actions in SAS Data Studio for Navigating through the Analytics Life Cycle

INTERACTIVE EXPLORATION AND MODELING

Extracting value from your data requires a certain degree of understanding the data, and although descriptive statistics and other forms of profiling are useful tools for this, there is no substitute for exploring your data interactively in a visual manner. The “Explore” phase of the analytics life cycle sets the stage for more in-depth analysis and modeling, enabling you to gain some initial insights from variable distributions and relationships, and providing a realization of the potential payoff that can be expected of predictive modeling.

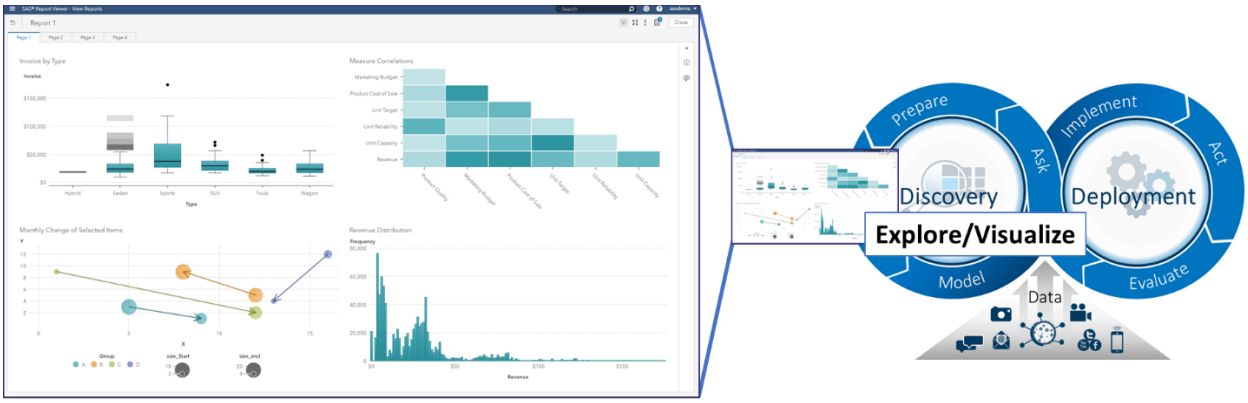

Figure 9. Exploring Your Data in SAS Visual Data Mining and Machine Learning

EXPLORING YOUR DATA

SAS Visual Data Mining and Machine Learning provides very powerful and intuitive visual exploration capabilities in the SAS® Visual Analytics interface. You can very quickly view the nature of your variables by using bar charts of distributions, and you can understand the relationships among variables by using objects such as scatter plots and correlation matrices. You can also get a sense for observational groupings by using crosstabulation tables (crosstabs), parallel coordinate plots, and clustering algorithms.

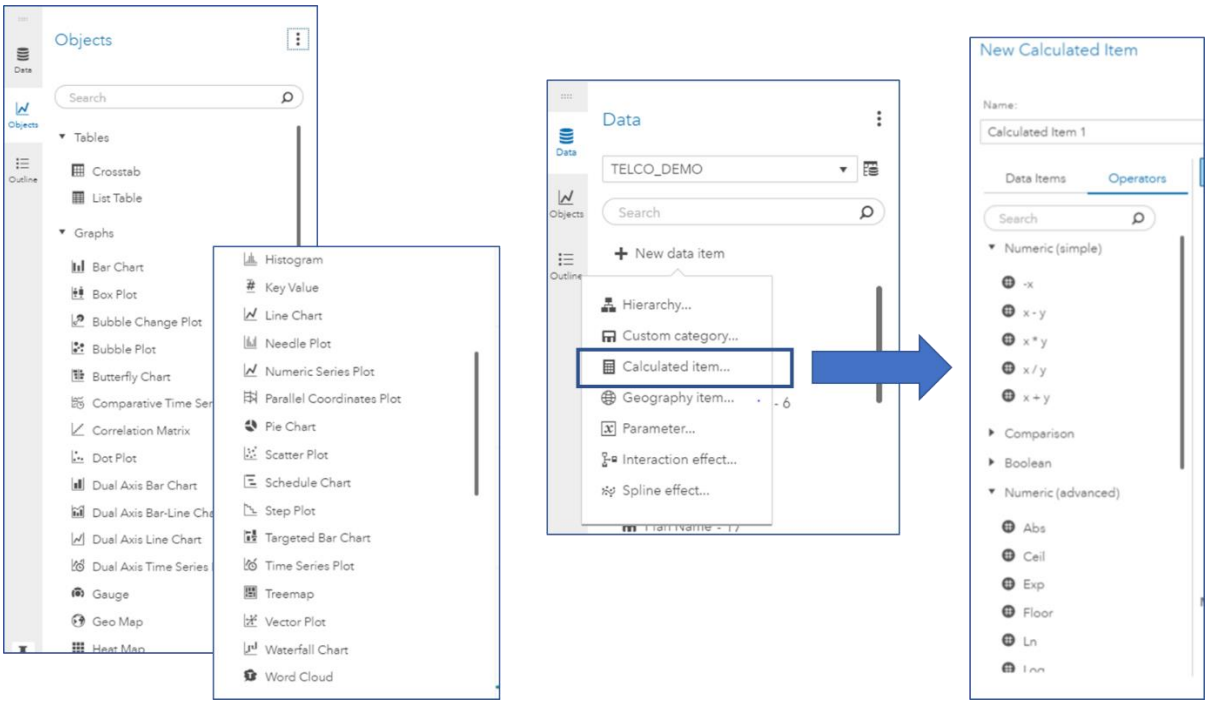

Figure 10. Visualization and Data Preparation in SAS Visual Analytics

Based on insight gained by exploring your data, or possibly from prior domain knowledge, you might need to transform columns or augment your data with new variables that are functions of existing variables. As previously stated, data preparation occurs in many forms in different phases of the analytics life cycle. During interactive exploration, you can create new data items very simply (without any programming) by selecting from a wide array of mathematical, comparison, date and time, text, and aggregation operators to build expressions that generate new columns. Persistent attention to the representation of your data leads to better, more meaningful predictive modeling results.

BUILDING PREDICTIVE MODELS

Ultimately, moving from *descriptive* analytics (understanding the nature of your data in terms of historical behavior and trends) to *predictive* analytics (realizing what might happen in the future based on the historical data) involves building models to represent the relationships between input variables and a target of interest. Using such models to classify new observations or predict target values is, of course, the focus of machine learning. One of the goals of SAS Visual Data Mining and Machine Learning is to offer these modeling capabilities in different forms that can be consumed by users who have various levels of expertise. To that end, you can build several different types of predictive models in SAS Visual Analytics in an interactive fashion with no programming required.

A crucial step in building predictive models is to ensure that you hold out data from the training process to honestly assess the accuracy of the model on data that was not seen during training. For this purpose, SAS Visual Analytics enables you to define partitions in your data by one of the following methods:

- Select a category variable that has two or three levels and select **Set as partition column**.

  Tip: *Convert a Measure variable to Category in order to select it as the partition variable.*

- Click the button next to the data source and select **Add partition data item** to create a new column to use to define the partitioning.

Models can be trained without a validation partition, but doing so is highly discouraged because overfitting to the training data will most likely occur and your model will not generalize well (that is, accuracy of predictions will diminish).

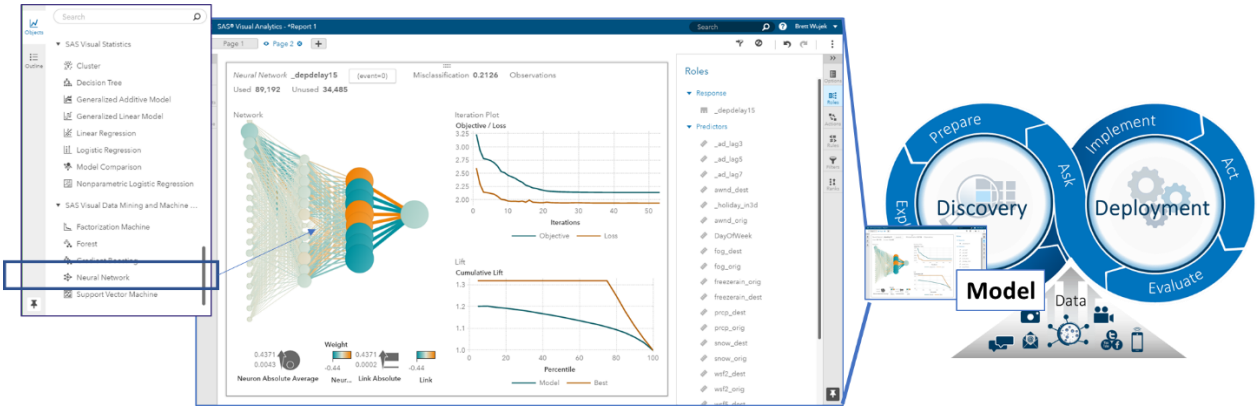

Figure 11. Interactive Modeling Using SAS Visual Analytics in SAS Visual Data Mining and Machine Learning

As shown in Figure 11, various objects for building predictive models are available in SAS Visual Analytics; the more modern machine learning algorithms fall under SAS Visual Data Mining and Machine Learning, whereas the basic regression techniques and decision tree fall under SAS® Visual Statistics. Training a model in SAS Visual Analytics is as simple as dragging a modeling object onto a page (or double-clicking it) and selecting the **Response** (target variable to predict) and **Predictors** (input variables) in the **Roles** tab on the right, as shown in Figure 11. You also need to specify the variable that serves as the **Partition ID** if one has been defined. Once you have selected the desired roles for your model, you can conveniently create other types of models that have the same roles by right-clicking and selecting **Duplicate as**.

Tip: *To create the model on a new page, press the Alt key when you right-click.*

The **Options** tab presents many algorithm settings, called *hyperparameters*, that can be adjusted to drive the training process. Some models provide right-click menu options for editing the model hyperparameters; for example, you can add, remove, or edit hidden layers by right-clicking a neural network diagram. Modifying the hyperparameters and immediately observing the resulting change in model accuracy through various metrics and assessment plots gives you an interactive means of gaining insight regarding the types of models you can build. Although you can manually explore different combinations of these settings, the number of all possible configurations makes manual exploration infeasible. To address this issue, for several algorithms you can select **Autotune** to instruct the software to automatically adjust the hyperparameters by using an intelligent search strategy to find the best model, as described in Koch et al. (2017). Since autotuning requires training numerous candidate models, invoking it disables interaction with this modeling object until the process completes, which could take several minutes. However, several measures are taken in the autotuning implementation to make the process as efficient as possible by managing parallel use of computing resources and using early stopping techniques (Koch, Wujek, and Golovidov 2018).

COMPARING AND USING YOUR MODELS

Training effective predictive models is somewhat of an art. Beyond applying different data preparation steps, employing feature engineering techniques, and finding the best hyperparameter settings to use, assessment of a model can be carried out using several different metrics. As you interactively build different models, often with different modeling algorithms, direct comparison can be cumbersome as you look at each model independently. SAS Visual Analytics provides a **Model Comparison** object that enables you to easily compare your models side-by-side in a variety of ways by presenting multiple standard assessment plots, different metrics, and cutoff and percentile selections. Models that have the same variables defined for the roles can be selected to include in the comparison.

Tip: *Modifying and retraining a model after the **Model Comparison** object has been created requires that you create a new **Model Comparison** object; it cannot be updated.*

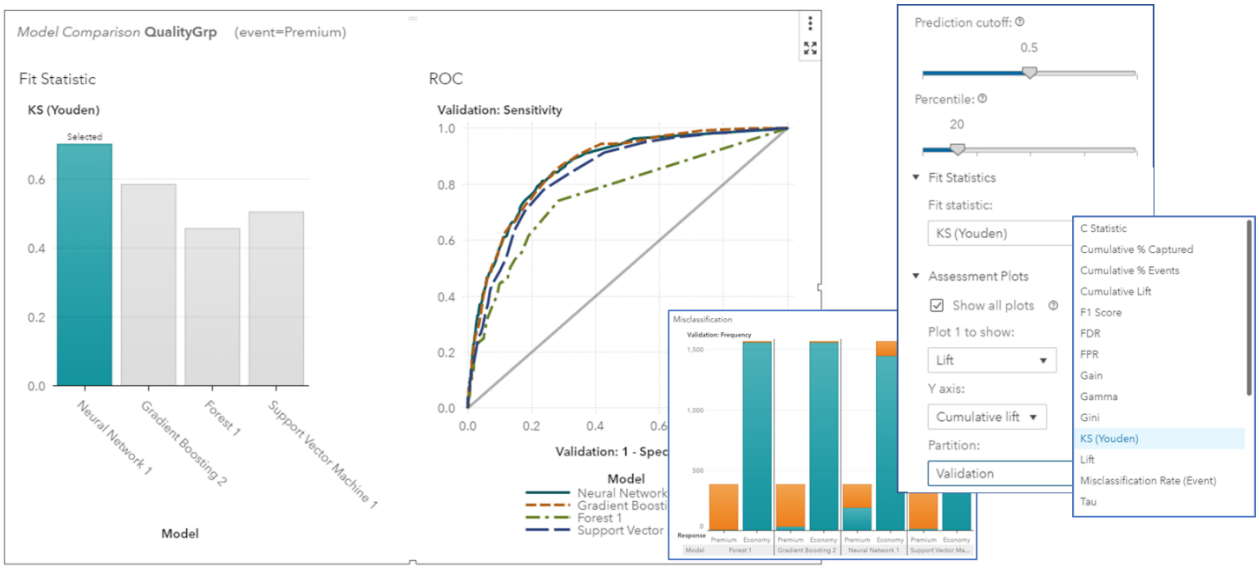

Figure 12. Comparing Models in SAS Visual Data Mining and Machine Learning

Once you have identified a model that you want to use for scoring new observations, you can right-click on the model and select **Export model**. This generates the score code for the model and downloads it for you to use as desired. Models that use the binary "astore" (analytic store) format save the model to the Models caslib in addition to downloading the SAS code that can be used to invoke the astore model for scoring.

Interactively building and assessing models within SAS Visual Analytics gives you a sense of the level of predictive power you can hope to achieve and the effect that different algorithms and their associated hyperparameters have in building accurate predictive models. In some cases, the resulting models are sufficient for use in your production environment. However, quite often they are best used as starting points for constructing more detailed and sophisticated machine learning applications by enhancing them to incorporate advanced feature engineering techniques and effective ensembling methods. Building pipelines to represent the flow of data through sequences of connected nodes that apply these techniques is an effective way of automating the process and ensuring repeatability. In SAS Visual Analytics, you can right-click on any model and select **Create pipeline** to progress to the next level of modeling, as depicted in Figure 13.

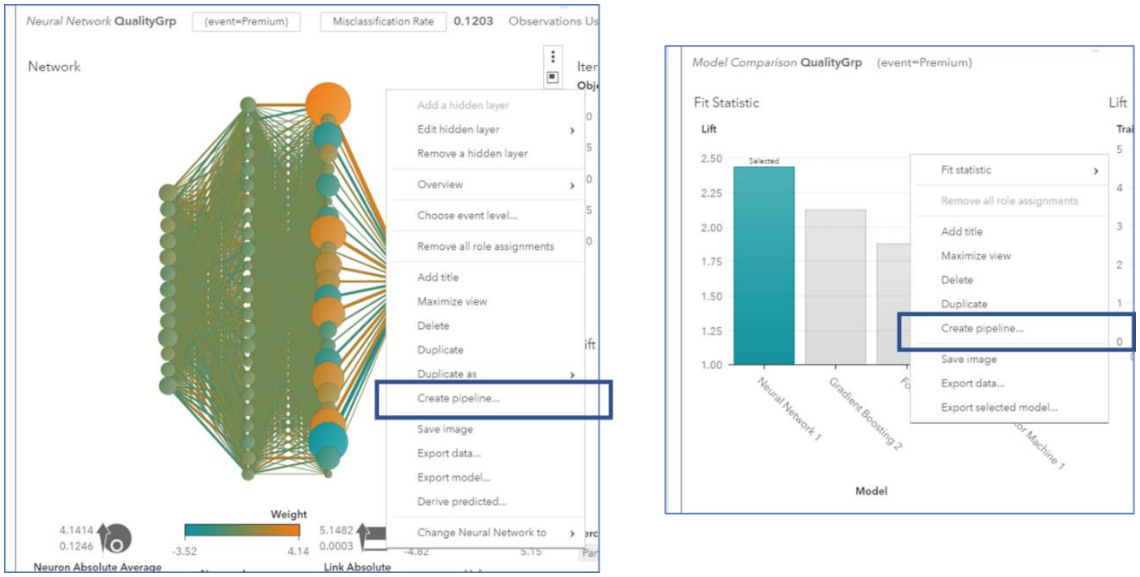

Figure 13. Taking Interactive Modeling to the Next Level by Creating Pipelines

AUTOMATED MODELING PIPELINES

Pipelines serve as self-documenting, automated, repeatable processes; they offer flexibility in the steps taken to build, compare, and ultimately choose the model for your business problem. Overall, they are tremendous productivity enhancers. To take you beyond interactive, ad-hoc modeling, SAS Visual Data Mining and Machine Learning offers a pipeline-centric, collaborative modeling environment in Model Studio. This feature-rich interface enables you to build automated processes that exercise feature engineering techniques, to apply algorithms to identify segments and build representations for classifying new observations and making predictions, and to assess and compare the generated models.

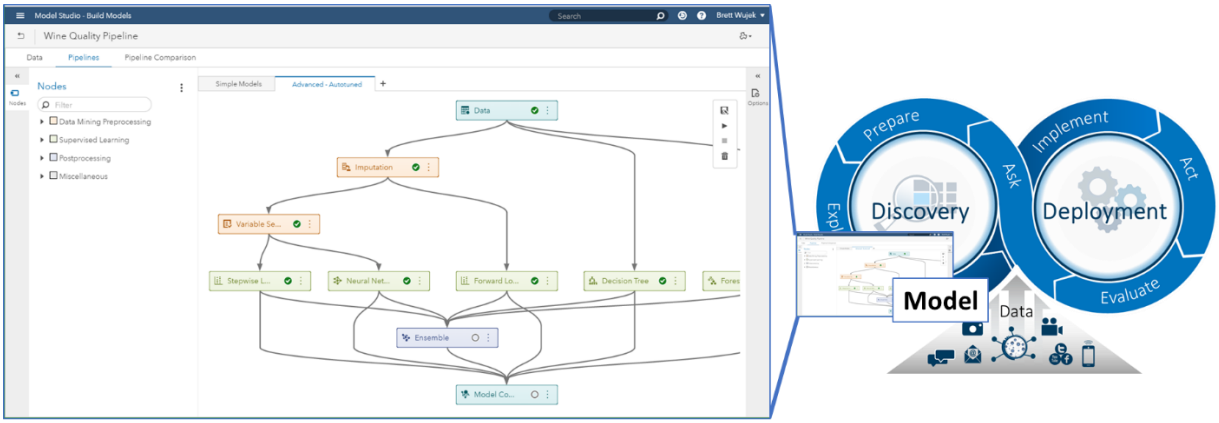

Figure 14. Automated Modeling Using Model Studio in SAS Visual Data Mining and Machine Learning

PROJECTS

The primary object for defining and managing pipelines for machine learning applications in Model Studio is a *project*. The Projects page serves as the top-level home page in Model Studio; it presents all the projects that you have created or that are accessible to you, either as a table or as a display of tiles. Since Model Studio is an interface that enables you to build pipelines for other domains—namely forecasting (for SAS® Visual Forecasting) and text analytics (for SAS® Visual Text Analytics)—you might see a mixture of projects of different types listed, depending on the products you have licensed. The color of the tile (or the Type column in the table view) indicates the type of project; for “Data Mining and Machine Learning” projects, the tile also presents a thumbnail image that corresponds to the type of model that is determined to be the *champion* (best) model for that project. The champion model is determined by assessing and comparing all models that have been trained in the project based on a specified metric.

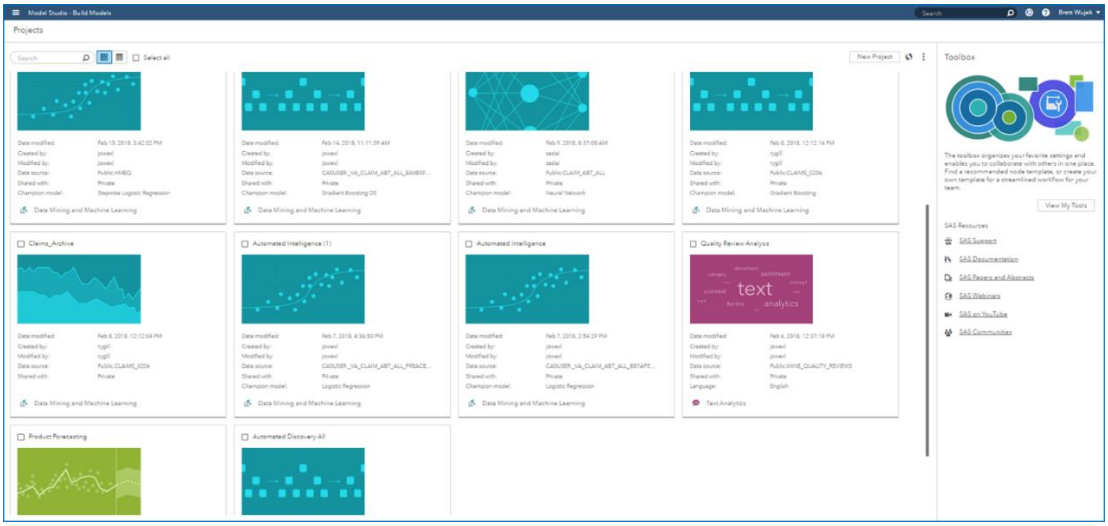

Figure 15. Model Studio Projects Page

By default, projects are private, meaning they are visible only to the creator and administrators, and editable only by the creator. However, Model Studio is designed and developed to be a collaborative environment, so projects can be shared with defined groups by selecting the project, clicking **Share** in the actions menu, and specifying the group with which it is to be shared, as illustrated in Figure 16. You can share a project as read-only or you can let others edit it; Model Studio ensures that only one person can edit a project at a time.

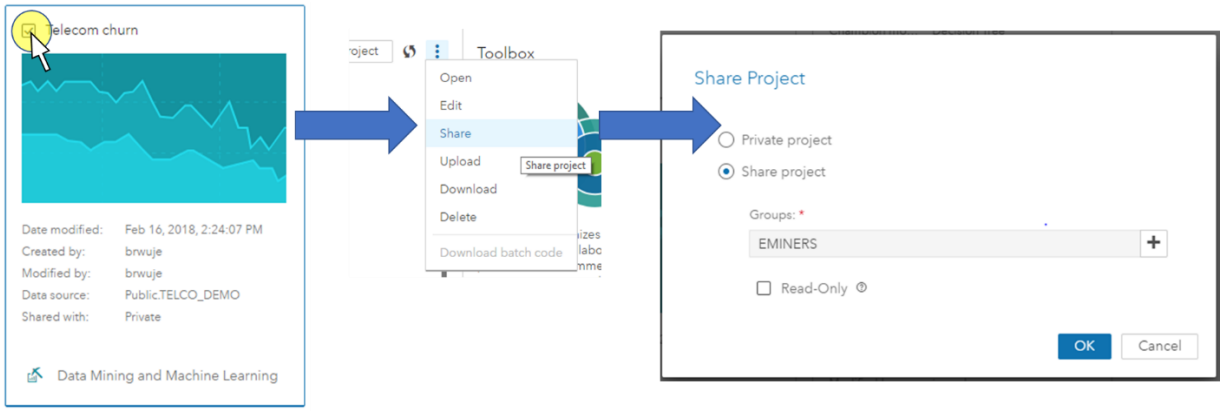

Figure 16. Sharing Model Studio Projects in SAS Visual Data Mining and Machine Learning

To archive projects offline or transfer them among servers, you can download a selected project by using the actions menu (⋮). The project is downloaded as a ZIP file that contains JSON files with all the information necessary to recreate the project. When uploading a project ZIP file, you simply need to specify the data source by selecting a CAS table available on the CAS server.

A well-defined and completed project can serve as a mechanism for generating a production-worthy model that is deployed to support making business decisions. Given that models can become “stale” (decay in accuracy) over time, you might need to re-execute your project, using new source data to generate a new model to consider for production. To avoid the need to use the Model Studio user interface for this, you can download batch code that contains the necessary RESTful API calls—invoked from SAS, Python, or REST (representational state transfer) code—to invoke the services to re-execute the project and generate new models. The code also provides API calls to check the status of the project execution and to obtain the new champion model after it is complete.

DATA DEFINITION

The first step in building models is to understand and define the metadata for variables that you will be using within the pipelines in your project. On the **Data** tab of Model Studio, you will find high-level summary statistics for each variable within your table, such as the count of unique levels, percentage of missing values, and the minimum, maximum and mean for your continuous variables. These metrics determine the default measurement level and roles that are assigned to each variable. For example, variables that contain more than 50% missing values are automatically assigned a role of **Rejected** and excluded from your analysis. All numeric variables that have more than 20 levels are assigned a measurement level of **Interval**.

Although Model Studio automates the generation of this metadata, you can also interact with this metadata and make modifications to the default definitions that are provided. To do so, highlight one or more variables within your table and select **Edit variable**.

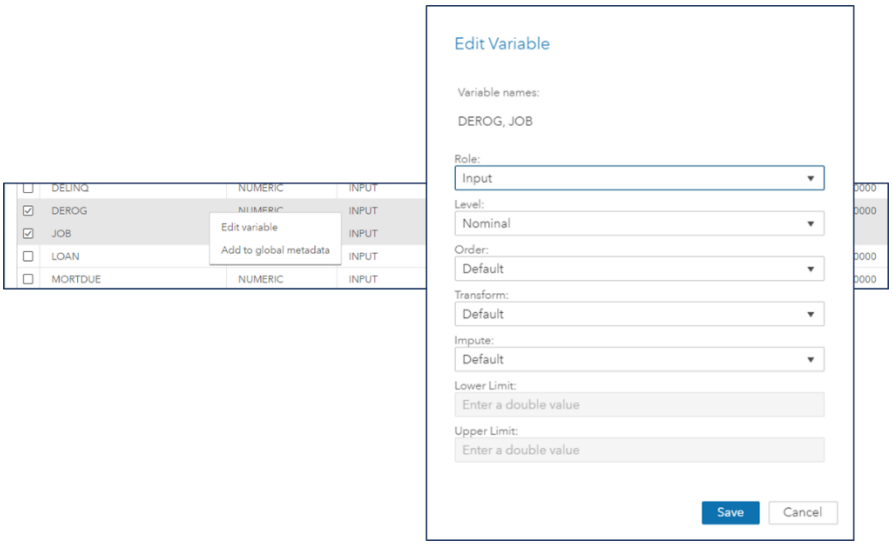

Figure 17. Editing Variables on the Data Tab in Model Studio

Tip: *One target variable and at least one input variable are required to run a pipeline in Model Studio. Once a target is defined and a pipeline has been executed, the target cannot be modified.*

You can modify the default settings for your metadata, and you can also define variable-specific imputation and transformation strategies that can be used within your pipelines. The imputation and transformation methods that are offered depend on the measurement level of the variable you are editing. It is important to note that the imputation and transformation methods defined here are not applied to your variables until you include the corresponding Imputation or Transformation nodes in your pipeline, as described in the next section. When these nodes are executed, the variable-specific methods are applied where defined and node-specific methods are applied to all other variables.

Tip: *To see additional columns in the **Data** tab in Model Studio, you can customize the display by using the Manage Columns button ( ) found in the upper right corner of the table itself. This button enables you to add or remove columns from the table and to designate their order.*

There are many cases in which data definitions for a variable might span multiple projects or even multiple data sources themselves. You can avoid defining this information in each instance by selecting **Add to global metadata** from the actions menu ( ) in the upper right. Each time a new project is created and metadata are generated, Model Studio checks for variables that match global metadata definitions by name (case-sensitive) and type, and it pulls in and reuses the information that is stored in the global metadata when it assigns values. Global metadata that has been defined can be managed in the Toolbox, which is accessible from the Projects page in Model Studio.

PIPELINES

As previously mentioned, Model Studio enables you to build automated process flows, called pipelines, that accommodate all the steps that are involved in a typical machine learning application. The **Pipelines** tab of Model Studio presents a visual representation of your pipelines as you construct them using "nodes." All pipelines start with a Data node to inject the project data into the flow and perform any specified partitioning (done only once for the project), and each step in a pipeline is represented by a node from one of the following categories:

- **Data Mining Preprocessing:** Nodes for manipulating and studying the data before building any models

  Note: The "Data Mining" prefix is used to differentiate these nodes from preprocessing nodes for other domains such as forecasting and text analytics. The prefix is omitted for the remainder of this paper.

- **Supervised Learning:** Nodes for building models to predict your specified target
- **Postprocessing:** Nodes for performing operations on models that are built by upstream nodes; currently this is dedicated to building ensemble models
- **Miscellaneous:** Nodes for various useful auxiliary capabilities

The specific available nodes within these categories are listed in Figure 18.

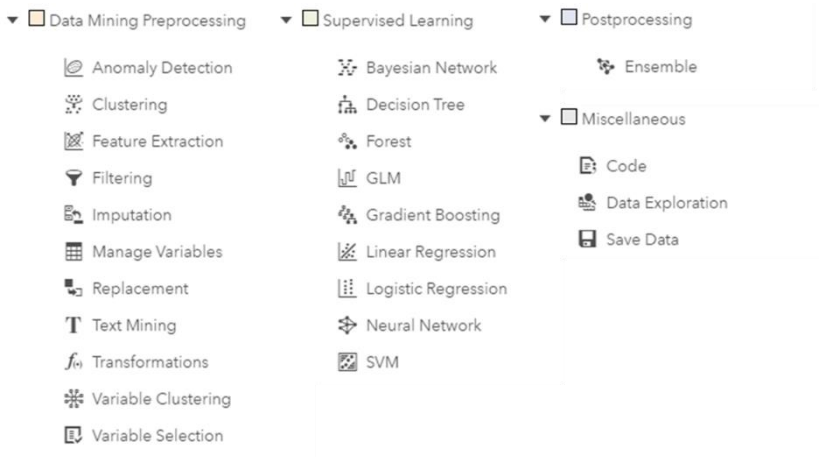

Figure 18. Categories of Nodes Available to Build Pipelines in Model Studio

Pipeline Construction

Instead of just providing an empty canvas that allows free-form pipeline construction in which nodes are added and connected in any desired fashion, Model Studio takes the deliberate approach of enforcing that nodes be connected only in a meaningful fashion, based on their categories. Supervised Learning nodes cannot be added before Preprocessing nodes, Postprocessing nodes must come after Supervised Learning nodes, Supervised Learning nodes cannot be run in sequence (that is, in the same branch), and the Data Exploration and Save Data nodes are terminal nodes. These rules ensure that the logic of your pipelines remains intact. Pipelines can take full advantage of the distributed execution environment of SAS Viya by executing different independent nodes in parallel branches so that they can execute simultaneously.

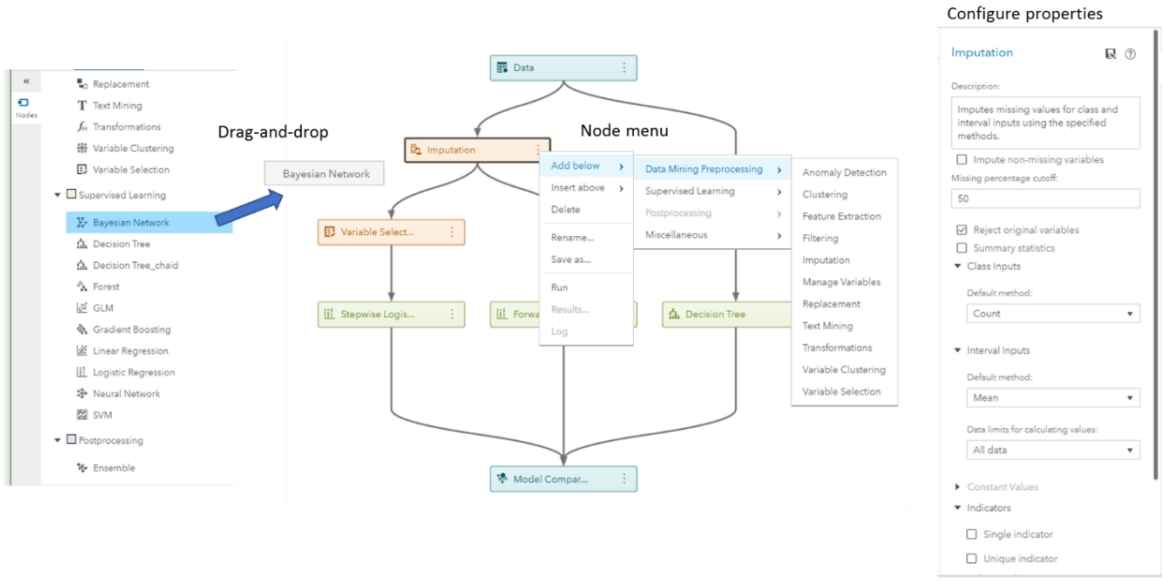

Figure 19. Constructing Machine Learning Pipelines in Model Studio

Creating pipelines is as simple as dragging nodes onto the pipeline canvas on top of the node that you want it to follow, or using the node menu (⋮) to insert nodes below or above an existing node. The pipeline rules described earlier ensure that you are adding nodes in appropriate locations, and parallel branches are automatically created when multiple nodes are added after a node. Because the primary goal of Model Studio is to automate and facilitate building multiple candidate models and comparing them to identify deployment-worthy models, all Supervised Learning nodes are automatically connected to a Model Comparison node that assesses the models and presents the results for comparison. This is discussed further in the section “Model Comparison.”

While building pipelines from scratch offers the flexibility to introduce whatever logic is deemed necessary for your machine learning application, existing pipelines can often serve as good starting points to avoid continually building similarly structured flows. Pipelines that have proven to be effective at building good models for one problem, or data set, can serve as “templates” to be applied to another problem. Model Studio comes supplied with pipeline templates for a number of scenarios and levels of modeling. When you add a new pipeline to your project, you can select a template as a starting point and then modify it and configure the node properties as desired.

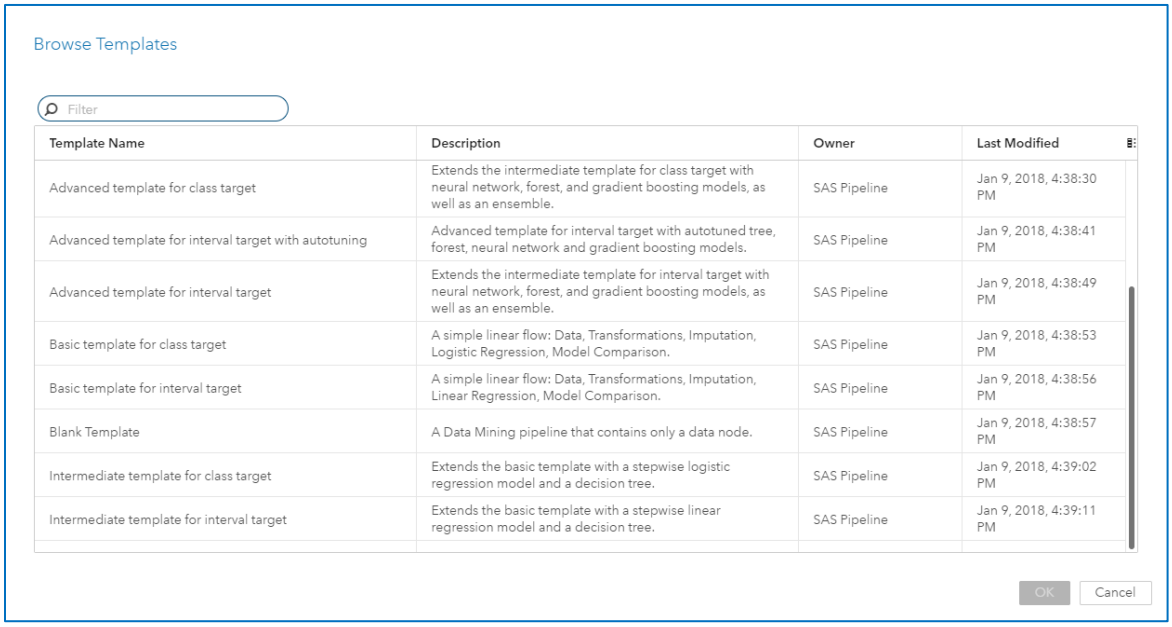

Figure 20. Pipeline Templates in Model Studio

Collaboration

SAS Visual Data Mining and Machine Learning was designed and developed with the mindset that data mining and machine learning projects are a collaborative effort, and that these projects often produce artifacts that are useful to other projects. The pipeline templates that are included in Model Studio are a basis that you can extend by saving pipelines that you create, allowing them to be used to create new pipelines in other projects, possibly by other users. To save a pipeline so that it can be used in other projects, click the Save button (💾) in the pipeline toolbar and provide a name and description for the template. The entire pipeline structure and all properties of all the nodes are retained so that new instances will start as exact copies of this template. This template will appear along with the predefined templates (with the corresponding creator specified) when you select a template to create a new pipeline. You can manage templates in the Toolbox, which is accessible from the Projects page in Model Studio.

Beyond sharing and reusing pipelines as templates, often you will find that a certain configuration of a node is very effective and you would like to use it in other pipelines or share it with others. For example, if you typically like a certain method and settings for feature extraction, you can configure a Feature

Extraction node and save a copy of it to the Toolbox by clicking the Save button () at the top of the Properties panel for that node. A copy of that node will then reside in the Toolbox and will be accessible in the appropriate category in the Nodes panel for selection and use in building pipelines.

Some Noteworthy Nodes

A detailed discussion of all available nodes is beyond the scope of this paper. Most of the nodes serve as convenient interfaces to underlying CAS actions that are associated with common data mining and machine learning capabilities. However, the following handful of nodes are highlighted because they provide specialized functionality to supplement and enrich your pipelines.

Manage Variables

As described previously, the **Data** tab enables you to specify information (metadata) about how your variables should be used in the project. However, often you want to specify different metadata in order to use the variables in special ways in different branches of a pipeline, or for different pipelines. For example, you might want to use different inputs for different models, or you might want to apply different transformations from those that are defined on the **Data** tab. The Manage Variables node enables you to do just that. After you insert a Manage Variables node, you must initially execute it in order to allow it to import the current metadata information; you can then view an editor that enables you to modify the variable metadata.

Tip: *Once a pipeline has been run in a project, metadata for the variables of the project cannot be modified on the **Data** tab. To modify the variable metadata for a specific pipeline, use the Manage Variables node within that pipeline.*

Data Exploration

In-depth interactive exploration is best done in SAS Visual Analytics as discussed in the section "Exploring Your Data." But often you are working on a pipeline in Model Studio and you want to get a sense of some intermediate form of your data. The Data Exploration node is a Miscellaneous node that displays summary statistics and plots for variables in the data table that is provided by the preceding node. The Data Exploration node selects a subset of variables to provide a representative snapshot of the data. Variables can be selected to show the most important inputs, or to indicate "suspicious variables" (that is, variables that have anomalous statistics). You can use the Data Exploration node to identify good candidate variables for inclusion in predictive models as well as variables you might want to exclude. This node can suggest variables that might require transformation (for example, variables that have skewed distributions) or imputation of missing values.

Tip: *To explore a model's predicted values, connect the Data Exploration node to the Supervised Learning node that produces that model.*

Save Data

By default, the table that is produced by a node in a pipeline is temporary; it exists only for the duration of the run of the node and has local session scope. You can connect a Save Data node to another node in order to save the output table to disk in the location that is associated with the specified output library. This table can then be used later by other applications for further analysis or reporting (for example, in SAS Visual Analytics).

Tip: *To allow the saved table to be seen in other CAS sessions, select the option to **Promote table**.*

Ensemble

Quite often, the most accurate predictions are not provided by a model that is trained from a single instance of an algorithm, but instead are provided by combining the predictions of multiple models into an

ensemble model. The Ensemble node is a Postprocessing node that enables you to create a new model by using a functional combination (aggregation) of posterior probabilities (for class targets) or predicted values (for interval targets) from multiple models in a pipeline. You can add an Ensemble node after any Supervised Learning node and then continue to add other existing models to it by using the **Add Models** option in the node menu (⋮). General model assessment statistics are provided in the results, and the generated ensemble model is treated just like other models in terms of comparison and potential selection as a champion. Score code for the Ensemble node is produced by combining the score code (DATA step or analytic store) of the constituent models.

Code

The SAS language is very expansive and contains many useful and powerful capabilities that are not directly available in the nodes that Model Studio provides. The SAS Code node provides ultimate flexibility in what you can incorporate into your pipelines. You can execute SAS procedures and write SAS DATA steps to create customized scoring code, conditionally process data, or manipulate existing data sets. The Code node is also useful for building predictive models that are not supported by existing nodes, for formatting SAS output, for defining table and plot views in the user interface, and for modifying variable metadata. This node can create output plots and tables that show up as results just as they do for other nodes, and data tables that are produced by a successful Code node execution can be used by subsequent nodes in a pipeline.

Tip: *Because the Code node can be used to run any SAS code you want, it defaults to being used for preprocessing. However, you can move it to be a Supervised Learning node by clicking the node menu (⋮) and selecting **Move**. This allows it to automatically connect to the Model Comparison node.*

MODEL COMPARISON

All supervised learning models automatically feed into a Model Comparison node to allow for side-by-side assessment and comparison of all candidate models within your pipeline.

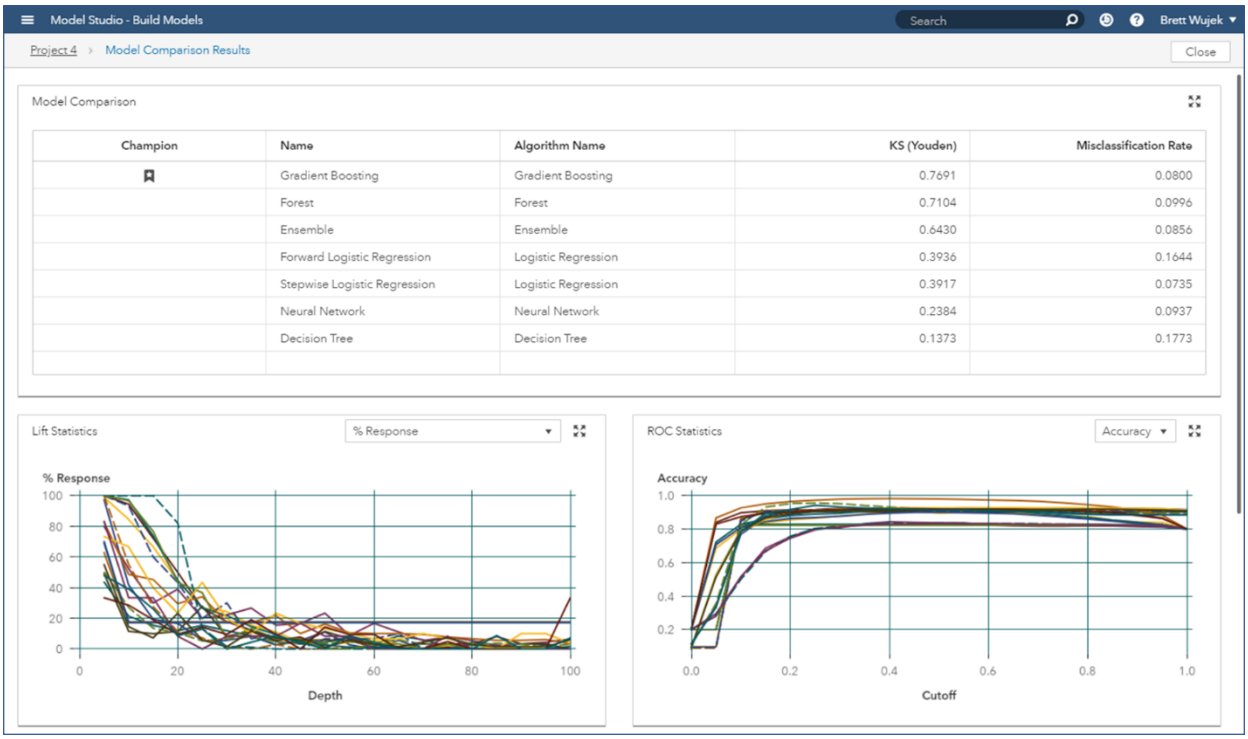

Figure 21. Comparing Models Generated by a Pipeline in Model Studio

The first table you see in the results of the Model Comparison node contains high-level information for each of your candidate models. The first column indicates the champion model that is automatically selected for you on the basis of your requested model selection statistic. You can configure this model selection statistic in the properties of the Model Comparison node; by default, it is based on the Kolmogorov-Smirnov statistic for class targets and the average squared error for interval targets. You can also select the partition that is used for champion selection; by default, the validation partition is used if it is available. In addition, the results of this node contain a comprehensive comparison of fit statistics and standard assessment plots across each of your candidate models, including lift, gain, %Response, and ROC, to name a few for a class target.

Tip: *You can change the default statistic, partition, and cutoff that are used for comparing models and selecting a champion in the user settings for Model Studio. To set the default values of these options as desired, select <**your user name**>→**Settings** in the upper right.*

The Model Comparison node automatically identifies and flags a champion model for each of the *pipelines* in your project, but your ultimate goal is to identify an overall champion for your *project*. This can be done on the **Pipeline Comparison** tab of Model Studio.

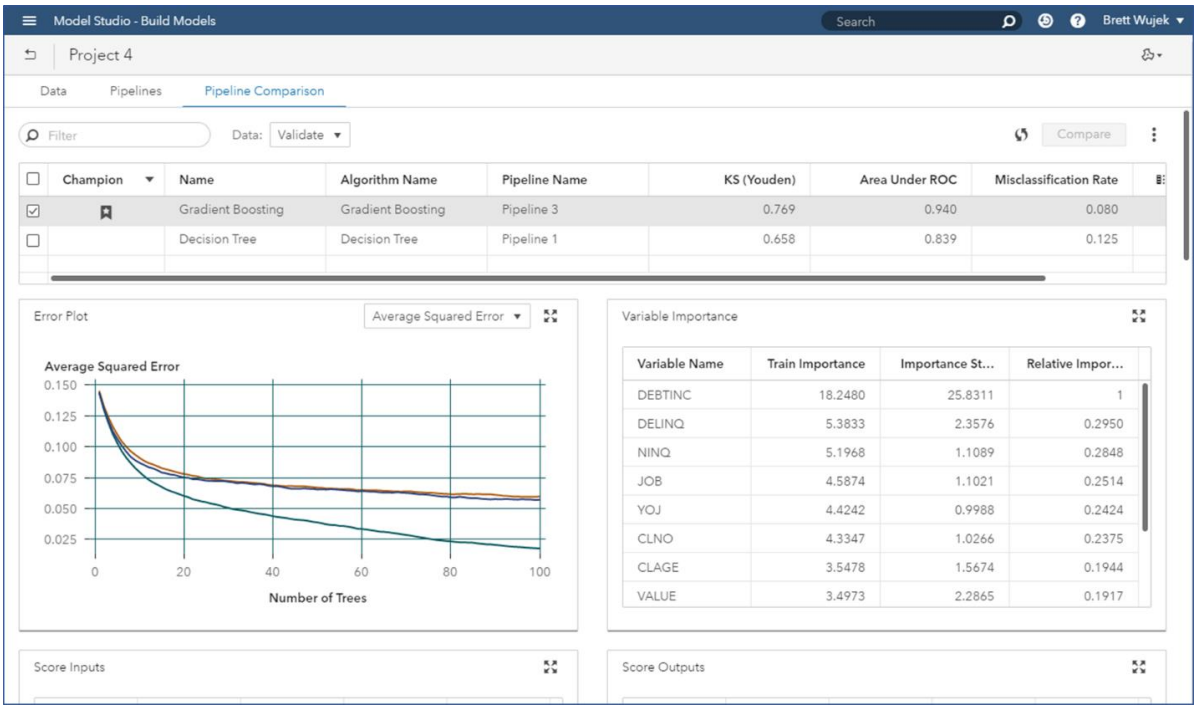

Figure 22. Comparing Models from All Pipelines in the Project in Model Studio

The look and feel of this tab is very similar to the results of the Model Comparison node (shown in Figure 21). The table at the top of this tab contains information for each of the candidate models in the project. In this case, your candidate models are the champion models that were identified by the Model Comparison node in each of your pipelines. As is done in the Model Comparison node, an overall project champion is identified from your candidate models based on your specified model selection criterion.

Tip: *You can also manually add challenger models (models that you want to be evaluated and compared to determine a champion) from any pipeline into the Pipeline Comparison table by selecting the associated node menu (⋮) and selecting **Add challenger model**.*

As you select individual models within this table, model-specific results are displayed in addition to assessment tables and plots, scoring code, and a list of required inputs and generated output variables for your model. If you select multiple models within this table, you can generate a side-by-side comparison of these models by clicking **Compare**.

There are times when you might want to include models that are generated outside of Model Studio to consider in the determination of your overall project champion. A perfect example would be when you want to compare models you have created in SAS® Enterprise Miner™ to new models that are generated in Model Studio. The score code for these external models can be easily imported into the Pipeline Comparison table by selecting **Import score code** from the actions menu (⋮), as shown in Figure 23.

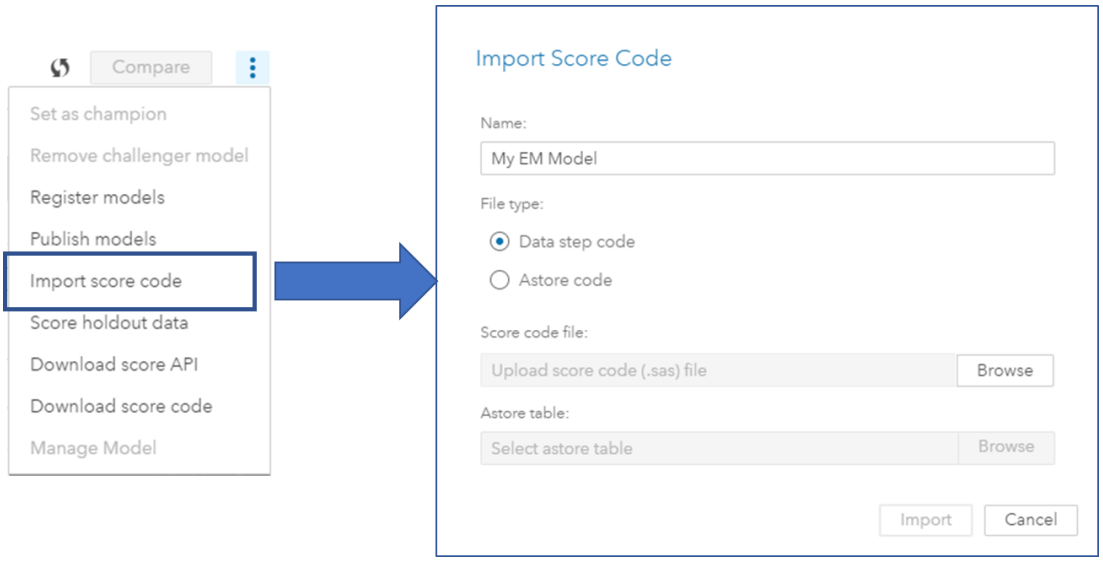

Figure 23. Importing Model Score Code in Model Studio

After this scoring code has been imported, it is applied to the project data, each of the partitions is assessed, and fit statistics are calculated. This model is now also included in the table of project candidates and is considered like any other candidate model during the overall project champion selection. No associated pipeline is created to visualize this external model.

In order to truly assess how well a model will generalize and perform, the model needs to be applied to a separate holdout table—a table that was not used in the creation of the models or in the validation for automatic selection of the champion. You can choose this table by selecting **Score holdout data** from the actions menu, as shown in Figure 23. This option opens a data browser that enables you to drill into available libraries and choose the appropriate CAS table to be used as a holdout. After the table is chosen, the score code for all project candidate models is applied to this holdout data. After the score code has been applied, the table and all assessment plots and tables for each model are updated to include values for the holdout sample.

Tip: *Although Model Studio automatically identifies a champion model for your project based on a model selection statistic, you can manually override the champion by highlighting the desired model and selecting* ***Set as champion*** *from the action menu* (⋮).

MODEL DEPLOYMENT AND MANAGEMENT

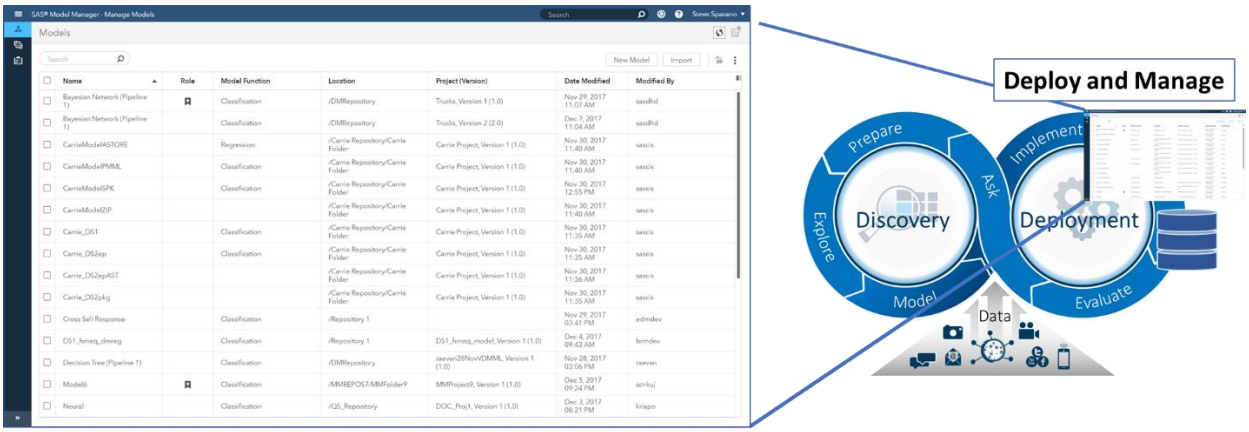

Figure 24. Managing Models Deployed by SAS Visual Data Mining and Machine Learning

Now that you have built candidate models and determined a champion, the next step in the analytics life cycle is to deploy and manage your models to aid in making effective business decisions. Deployment capabilities are surfaced through several options on the action menu (⋮) on the **Pipeline Comparison** tab in Model Studio.

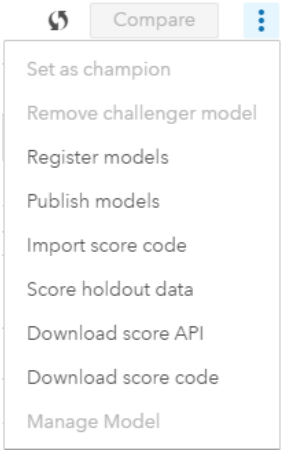

Figure 25. Actions Available for Using Models Created in Model Studio

A powerful way of deploying your models to your production environment is to publish them to destinations that support executing them to score new data. If you select **Publish models**, you are prompted to select a destination, which can be CAS, Hadoop, or Teradata, depending on the existence of a license for the corresponding SAS® Scoring Accelerator. As part of the publishing process, the model is translated into scoring code that can be seamlessly executed within the production environment.

Tip: *Before you can publish models to a caslib for CAS, Hadoop, or Teradata, that caslib must be specified as a publishing destination by an administrator.*

Alternately, you can execute models directly via a scoring web service call by selecting **Download score API**. This action provides you with the code that uses a model to score data by issuing a REST call, wrapped in SAS, Python, or simply the REST call itself. You can also download the SAS score code itself to execute in a SAS environment; for models in the form of analytic stores, the associated astore file is saved to the Models caslib.

Although using your models to score new data is a fundamental goal, it is just as important to maintain a level of organization for the models and a measure of control over them. If you have a license for SAS®

Model Manager, you can also register your models into a common model repository. This repository supports version control of your models, and it enables you to monitor stability and performance over time, and to update and retrain models when necessary, as described in Clingroth (2018). Proper management of the models that are used in your production environment is a necessary step to ensure that they are adequately and accurately addressing your business problem. With this management in place, you can confidently incorporate these models with other business rules into your decision process and operational workflows by using SAS® Decision Manager (if licensed) to automate analytic-based decision making.

CONCLUSION

SAS Visual Data Mining and Machine Learning is not just a set of algorithms. It is not a specific user interface or programming module. Rather, it is a comprehensive, fully integrated assembly of capabilities and interfaces that accommodate the entire analytics life cycle, enabling you to smoothly navigate from data to decisions. Built on the foundation of SAS Viya, it exploits the full power of your available computing resources with the latest innovations in in-memory analytics for efficient execution on distributed data. It offers full flexibility with support for interactive, programmatic, or automated approaches to applying analytics to your data. The collaborative environment enables users of all levels of expertise, from programmer to business analyst, to participate in the process of using modern machine learning techniques to turn data into real business value.

Figure 26. End-to-End Navigation of the Analytics Life Cycle with SAS Visual Data Mining and Machine Learning

APPENDIX: CASE STUDY

This case study illustrates how you can use SAS Visual Data Mining and Machine Learning on SAS Viya to solve a modern business problem. It takes you through the process of preparing data, interactively exploring data and building models, building automated modeling pipelines, and deploying models—all within one environment.

Imagine that you are a data scientist at a telecommunications company and you are charged with the task of identifying the most likely customers to drop their service within the next year. You have data sets that represent account information and usage patterns.

Tip: *The data and other artifacts that are associated with this case study can be found at* https://github.com/sassoftware/sas-viya-machine-learning/tree/master/case_studies/telecom

PREPARE DATA

Two tables have been loaded into memory:

- TELECOM_ACCOUNTS, which contains information about account attributes
- TELECOM_USAGE, which contains usage attributes.

These two tables are joined together by customer_id in the Data Studio interface. If there were additional data tables to join, they could also be joined in here.

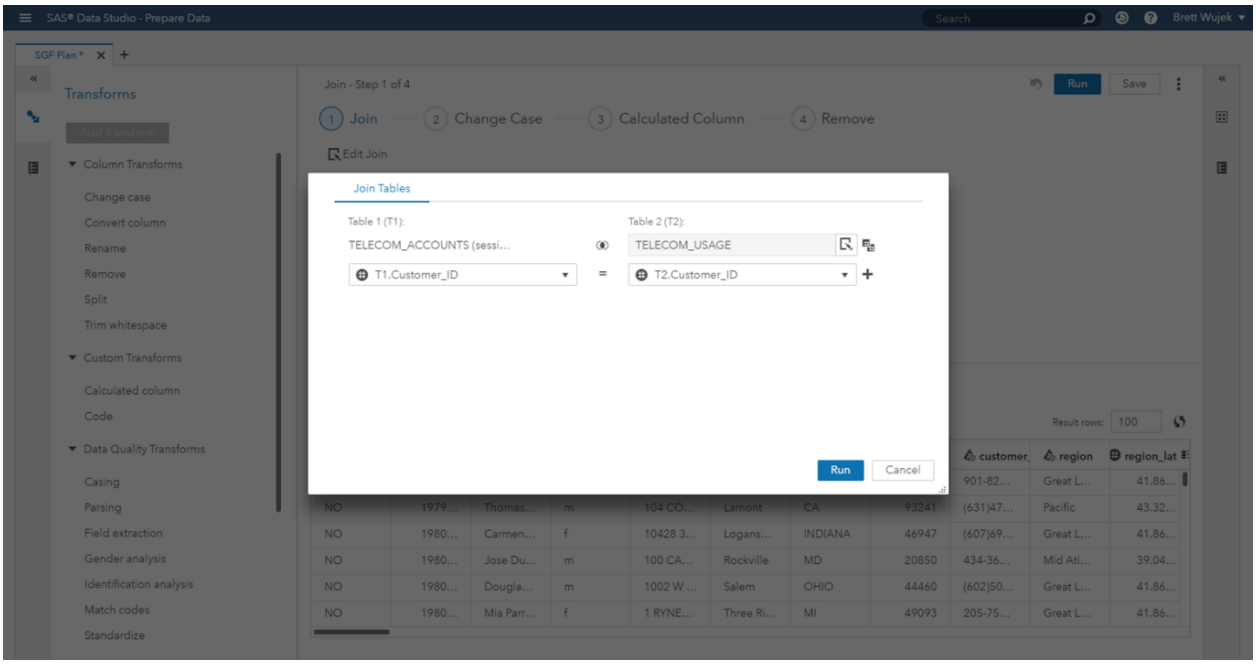

Figure 27. Join Multiple Tables Using SAS Visual Data Mining and Machine Learning on SAS Viya

An important aspect of data preparation is to identify potential areas for data cleansing. For example, you might have data such as city and state that are in mixed case and have varying spellings and abbreviations. Figure 28 shows a visual profile of the data. As you can see, the variable customer_state contains values of mixed case. The predictive models would treat values such as Massachusetts and MASSACHUSETTS as different values. You can automatically standardize the values by using the **Change case** transform.

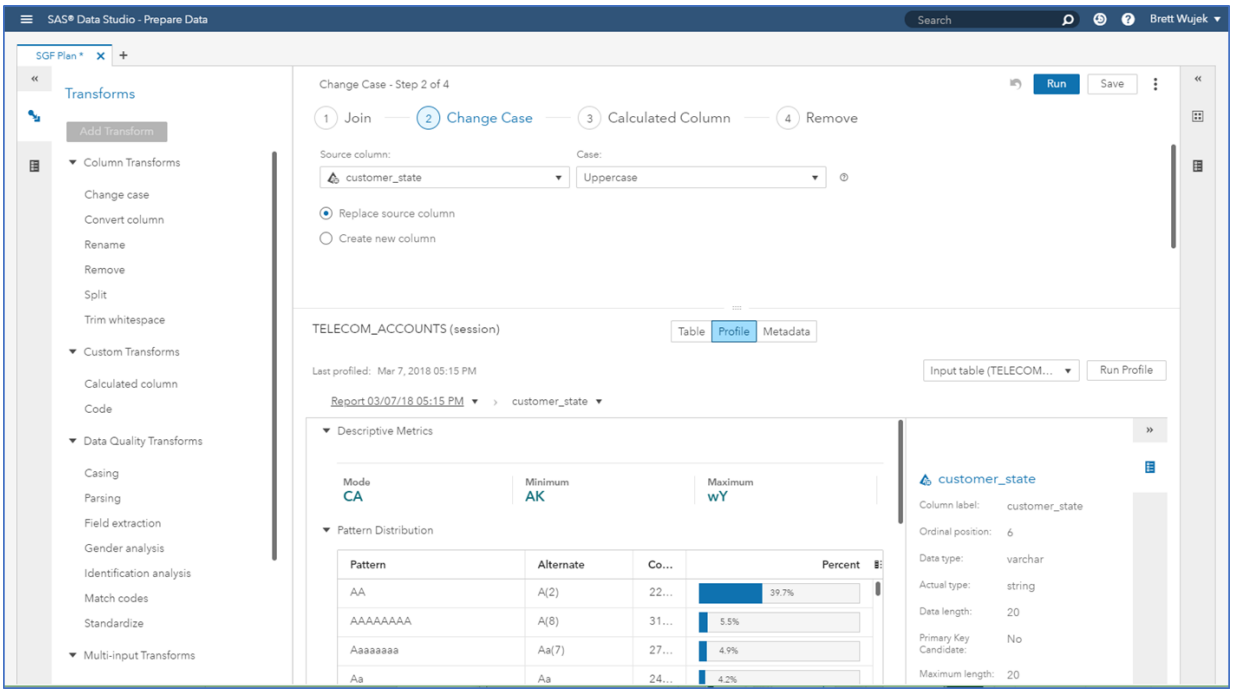

Figure 28. Generate Variable Profiles and Identify and Fix Data Quality Issues

The column that contains the churn information contains two numeric values: 1 and 0, which represent YES and NO respectively. Use the **Calculated column** transform to add custom SAS code that executes the IFC SAS function to transform the values to YES and NO.

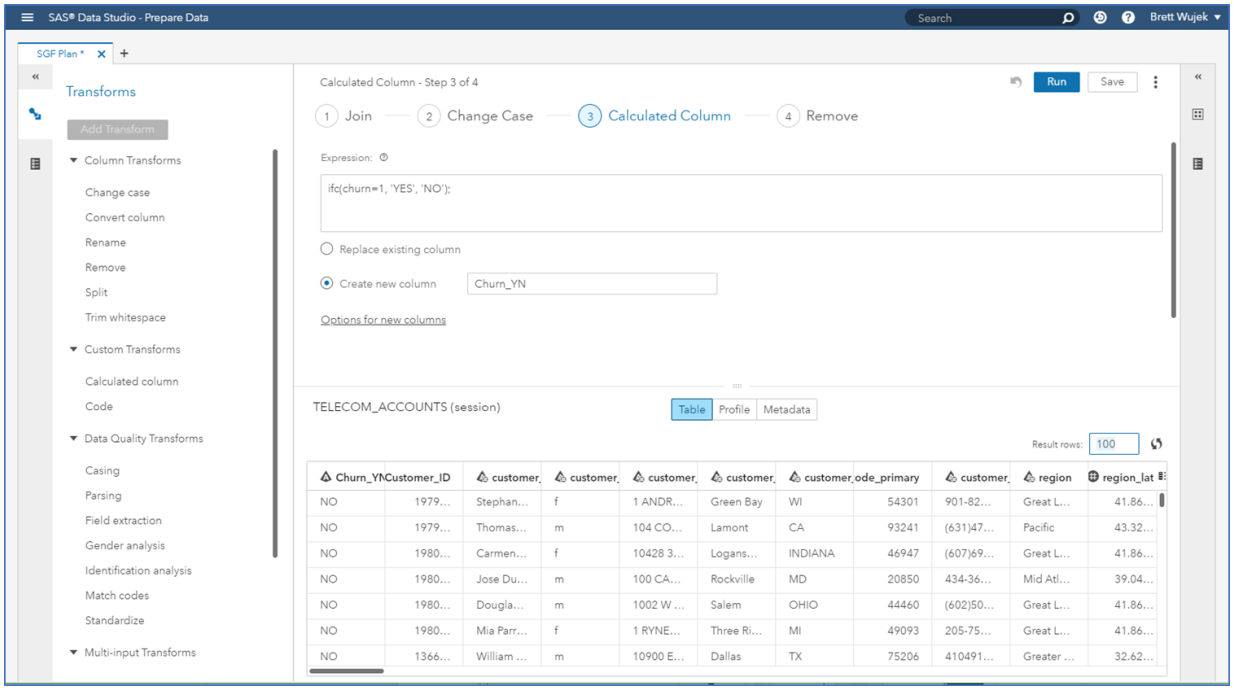

Figure 29. Create Calculated Columns by Using Custom SAS Programming Functions

Now that you have joined your data tables together, standardized the values of customer_state, and created the target column, you can save this data preparation “plan” for future use. This analytic base table can be shared with other users in the system for collaborating and running analyses in parallel.

Sharing the table reduces the need to manually copy data, thus preserving a single center of truth. With preliminary data preparation complete, you can now explore and visualize the data.

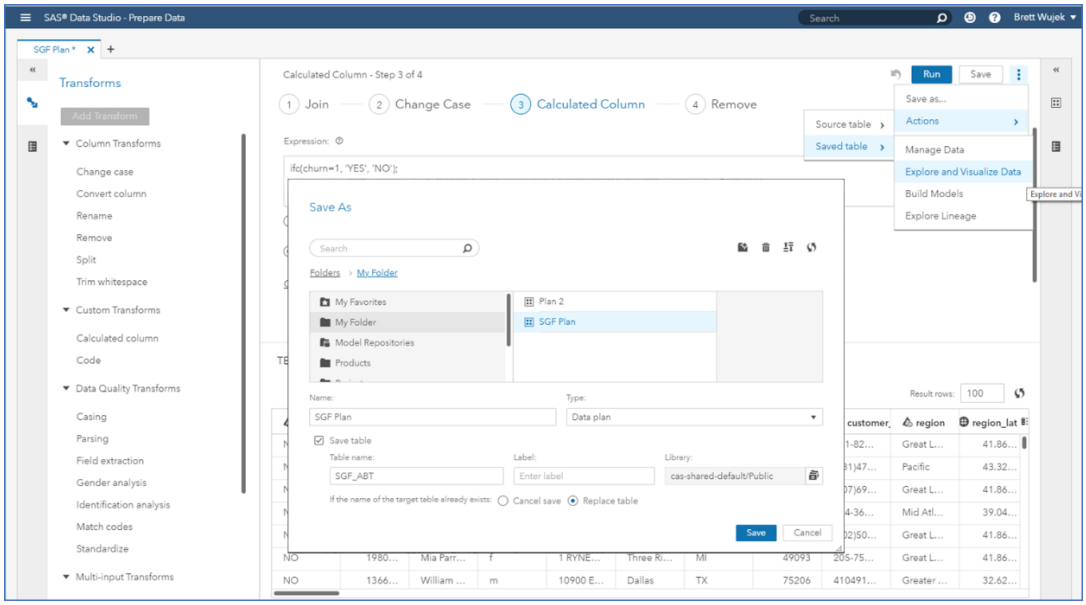

Figure 30. Save Data and Plan for Additional Analyses

EXPLORE AND VISUALIZE

Once you click **Explore and Visualize Data**, another interface automatically appears and presents the data that you just prepared. The data table has not been copied; rather the application launched and pointed to the same data table in memory.

Figure 31 shows a bar chart that was created to present the distribution of the target variable. Approximately 4.16% of the customers churned within a year. A correlation matrix of the continuous attributes in the data was also created. The darker shaded cells indicate a strong correlation between the variables.

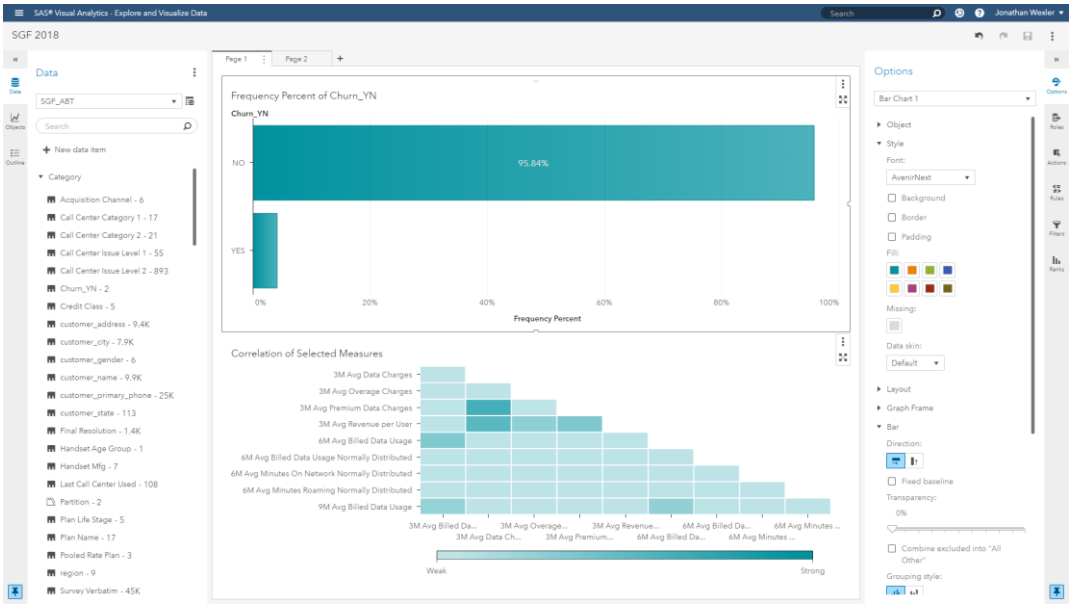

Figure 31. Interactively Explore and Visualize Patterns in Data Using SAS Visual Data Mining and Machine Learning on SAS Viya

To identify the customers most likely to churn, you can build a gradient boosting machine model and partition the data to hold out 30% for validation. This partition information is represented in the data as a new column.

As you can see in Figure 32, the Validation Misclassification is 0.0230, meaning 2.3% of your validation set observations were not correctly classified. The default settings of the models were used, so this gives you a good baseline to start. You can see that Times Suspended Last 6M and Total Voice Charges are the two most important factors in identifying customer churn. The variable importance column is interactive, so you can remove columns that do not add value to the model.

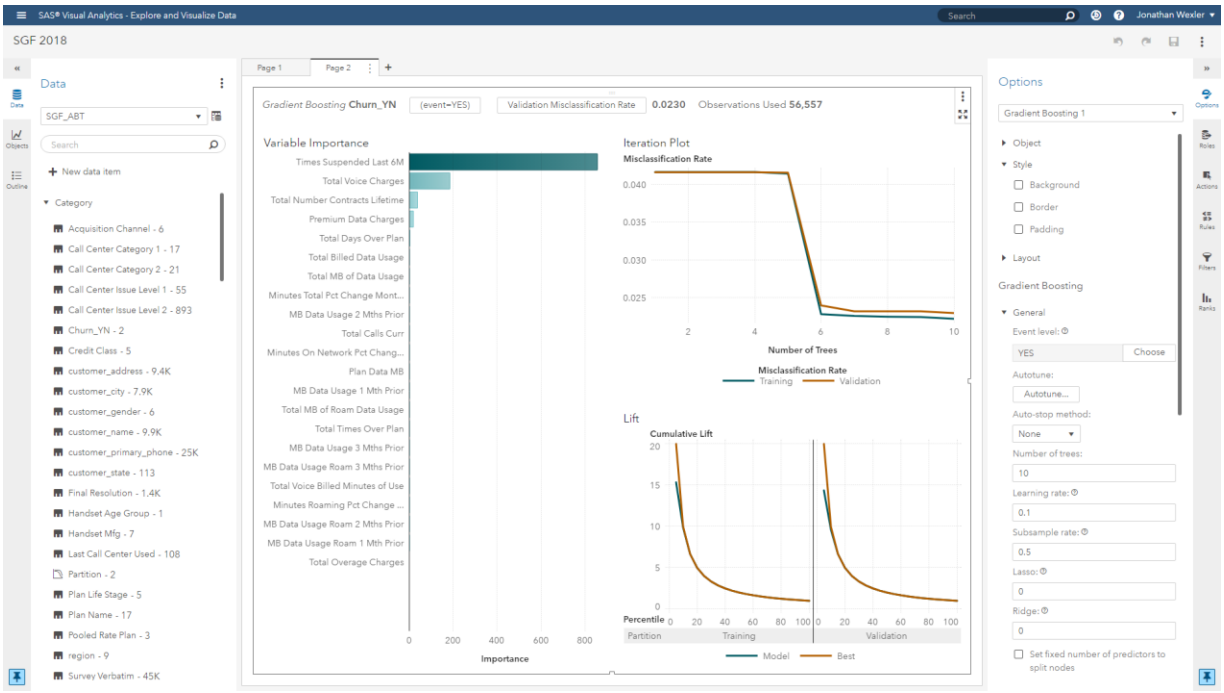

Figure 32. Interactively Build and Tune a Gradient Boosting Machine Model

You could spend time manually adjusting algorithm settings (hyperparameters) for this model to minimize misclassification. Instead, you can invoke this process automatically by using autotuning (click **Autotune**), which uses optimization methods to intelligently search for the optimal set of hyperparameters for your model. In Figure 33, you can see that using autotuning improved Validation Misclassification to 0.0223. In this case, the best set of hyperparameter values were four auto-stop iterations, 150 trees, a learning rate of 0.421, a subsample rate of 1, and lasso and ridge regularization parameters of 5.34 and 6.84, respectively. Manually experimenting to find these values by trial-and-error would be infeasible.

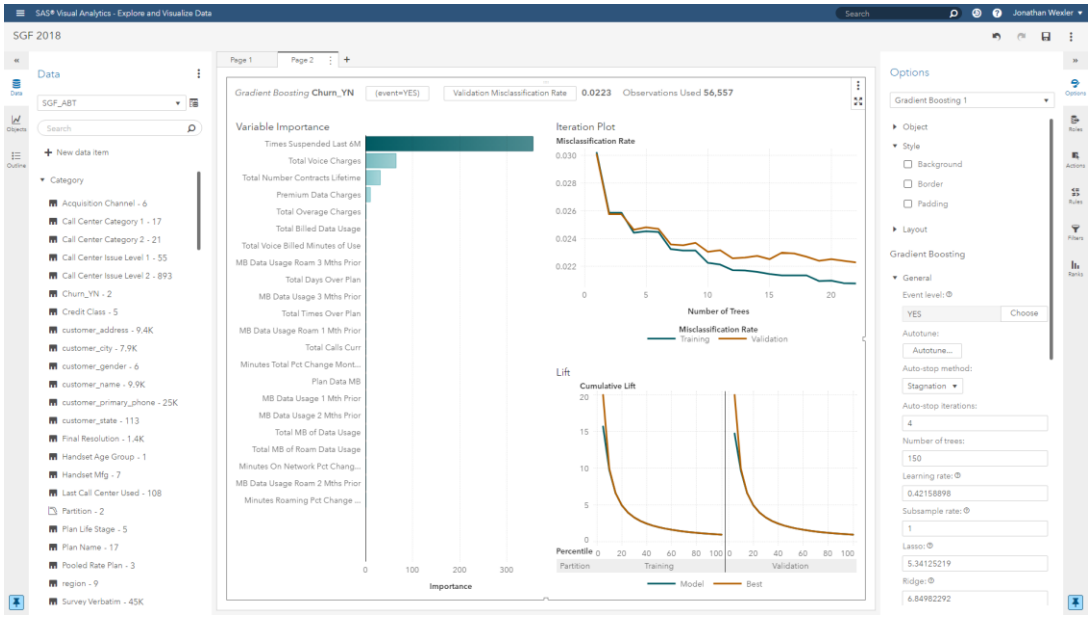

Figure 33. Use Autotuning to Find the Optimal Set of Hyperparameters for Your Model

BUILD MODELS

You could spend additional time tuning your gradient boosting machine and building more models such as neural networks and forests. However, at this point you might want to capture your interactive process and preserve it for reuse. When you right-click on the model results and select **Create pipeline**, any interactive steps that were executed for data preparation and the generated gradient boosting machine model are represented as a pipeline, as shown in Figure 34. If you had created a data transformation, that would be represented in the Interactive Data Prep node. Note that the score code for the gradient boosting machine that was generated interactively is automatically incorporated as a modeling (Supervised Learning) node in the pipeline. This node can be used for assessment and comparison, but the model properties cannot be changed for retraining in this software release (SAS Visual Data Mining and Machine Learning 8.2).

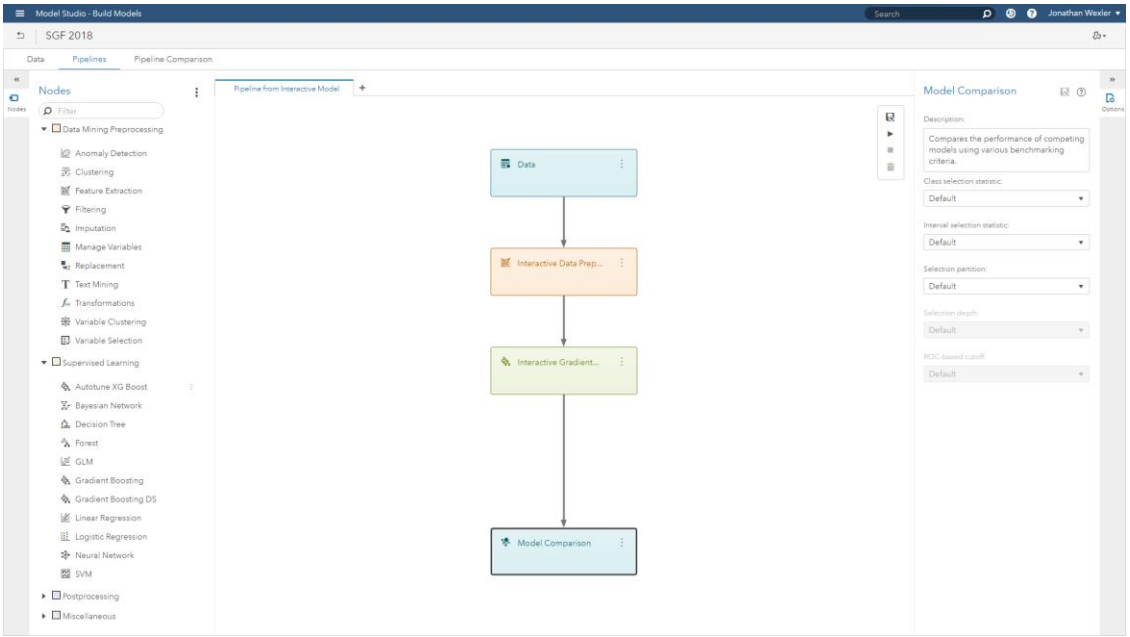

Figure 34. Generate Automated Pipeline for Additional Feature Engineering and Modeling

In Figure 35, this pipeline is enhanced by imputing missing values, adding tree-based binning interval transformations, and feeding the transformed data into nodes to generate a stepwise logistic regression model and an autotuned gradient boosting machine model.

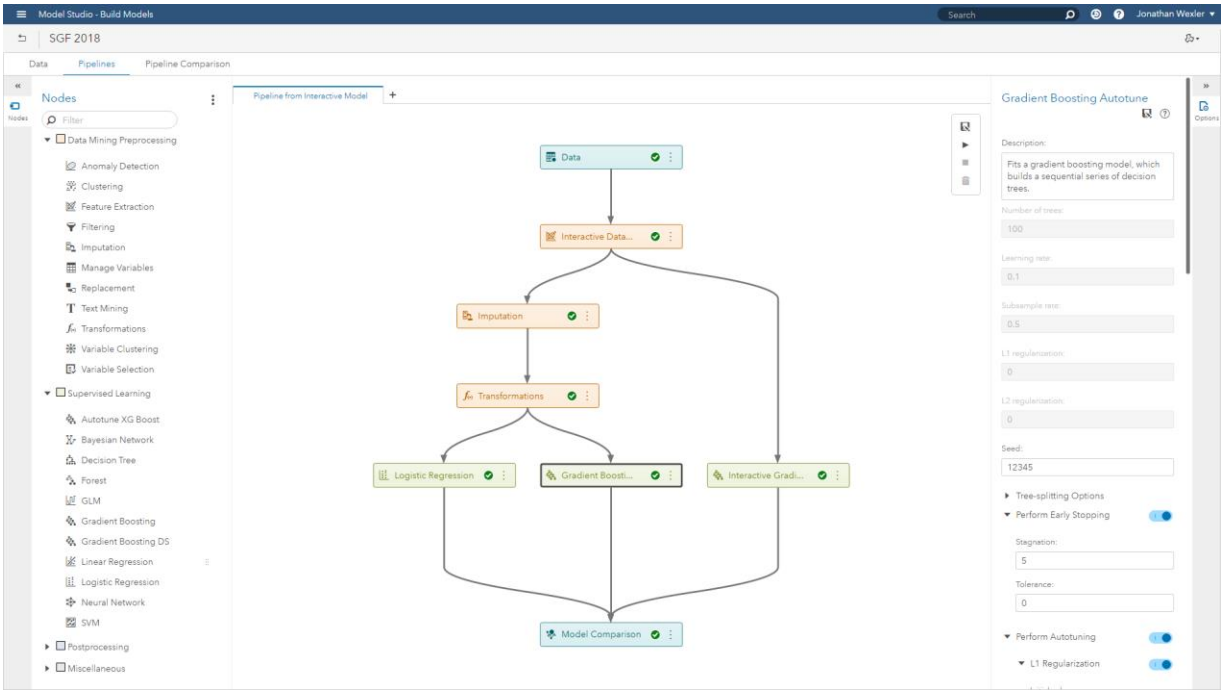

Figure 35. Enhance Pipeline with Additional Analytic Methods and Automatically Choose the Best Model

The pipeline automatically chose the best performing model, based on lowest validation misclassification rate. The gradient boosting autotune model had the lowest misclassification rate at 0.0217.

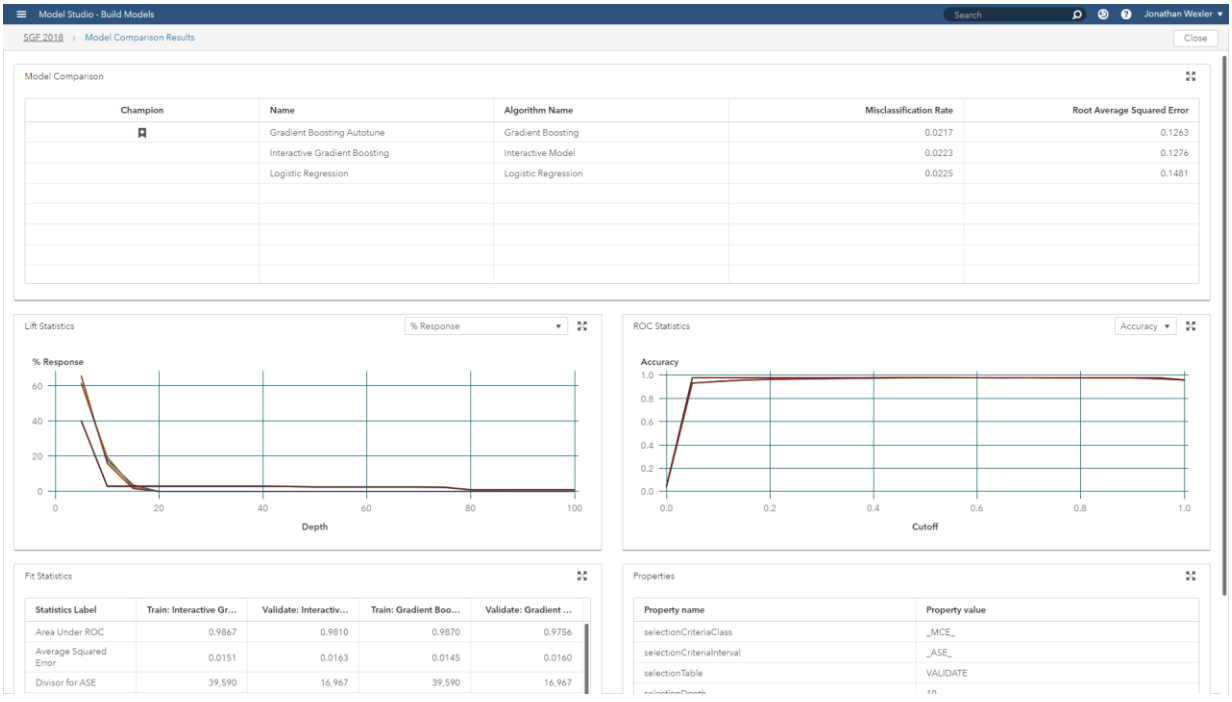

Figure 36. Automatically Select Champion Model Based on Assessment Statistics

After you have trained several models, including a selected champion model and any potential challenger models, you have complete flexibility with respect to model deployment. Figure 37 shows the code that is generated to invoke a SAS scoring API. For example, if you had a web application, you could use this code to call back into SAS to automatically score new records. You could just as easily use the Python or REST APIs to deploy your models. If your production data that needed to be scored resided in Hadoop or some relational database, you could also publish these pipelines automatically with just two mouse clicks, depending on SAS licensing. The pipelines would be embedded in the production systems, with no manual recoding.

Note that when the time comes to update your models, you can also retrain this entire pipeline by using a batch retrain API that the application generates.

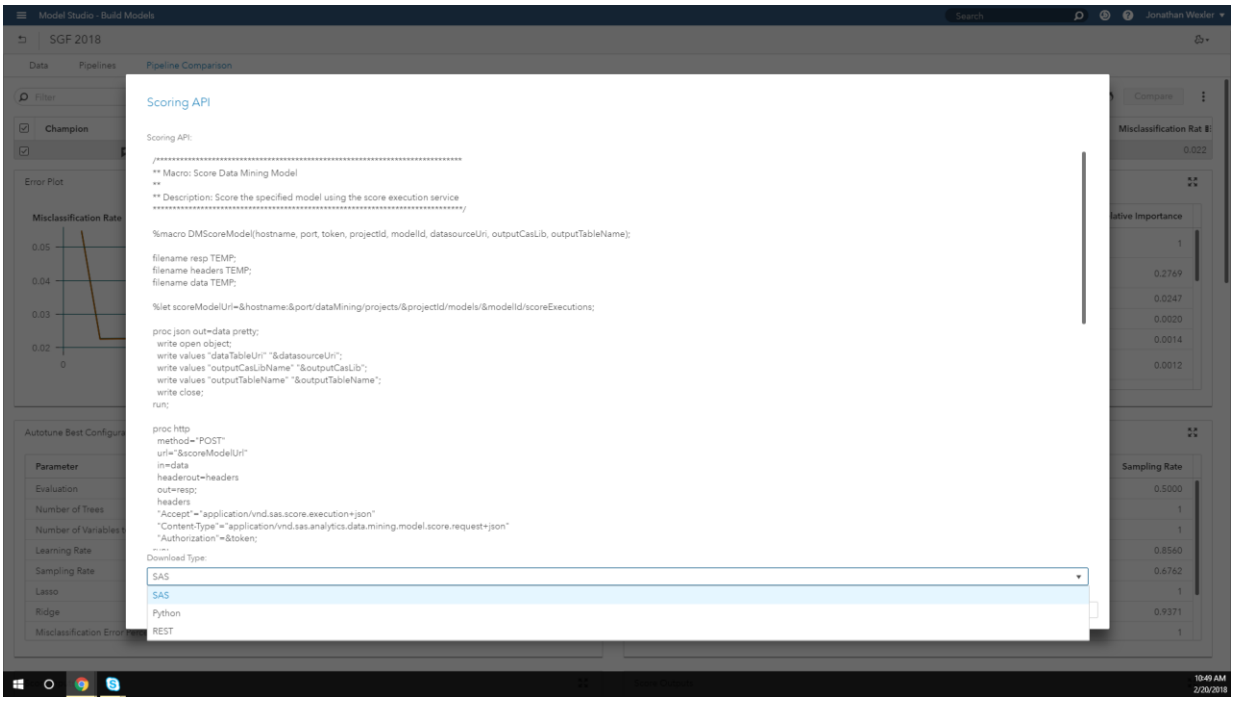

Figure 37. Deploy Pipelines to Production Using In-Database/Hadoop Publishing or Scoring APIs

REFERENCES

Clingroth, G. "Introducing SAS Model Manager 15.1 for SAS Viya." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc.

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

Koch, P., Wujek, B., and Golovidov, O. (2018). "Managing the Expense of Hyperparameter Autotuning." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc.

Wexler, J., Haller, S., and Myneni, R. 2017. "An Overview of SAS Visual Data Mining and Machine Learning on SAS Viya." *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings17/SAS1492-2017.pdf.

Wujek, B., Hall, P., and Güneş, F. (2016). "Best Practices in Machine Learning Applications." *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings16/SAS2360-2016.pdf.

ACKNOWLEDGMENTS

The authors would like to thank Anne Baxter for her contributions to this paper.

CONTACT INFORMATION

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

Brett Wujek
SAS Institute Inc.
brett.wujek@sas.com

Susan Haller
SAS Institute Inc.
susan.haller@sas.com

Jonathan Wexler
SAS Institute Inc.
jonathan.wexler@sas.com