Lesson Menu Print

Demo Steps for Forecasting Using Model Studio in SAS® Viya®

Course Code: EFVVF42, prepared date: August 5, 2025

Lesson 01

Forecasting Using Model Studio in SAS® Viya®
Lesson 01, Section 1 Demo: Creating a Forecasting Project and Loading the Data

In this demonstration, you create a new forecasting project in Model Studio, **baseline sales forecasts**, and load data into the project. The **baseline sales forecasts** project is used throughout the course.

1. Navigate to the landing page of SAS Viya using the URL and credentials supplied in the virtual lab instructions.

2. Navigate to the upper left corner of the landing page, click **Applications** > **Build Models**. This takes you to Model Studio.

   Model Studio is an integrated visual environment that provides a suite of analytic tools to facilitate end-to-end data mining, text, and forecast analysis. The tools in Model Studio are designed to take advantage of SAS Viya programming and cloud processing environments to deliver and distribute the results of the analysis, such as champion models, score code, and results. It does all this fast.

   Note: If this is your first session, there will be no existing projects unless projects were set up for you. (If projects already exist, the **New Project** button is available in the upper right corner.)

3. Name your project **baseline sales forecasts**.

   Note: Naming your project something relevant and adding a reasonably detailed description of the project is considered a forecasting best practice.

4. For **Type**, select **Forecasting**.

   There are three types: Data Mining and Machine Learning, Forecasting, and Text Analytics. This course will only deal with Forecasting.

   Once you select a type and you create your project, you can't change the Forecasting type. So you would have to reopen a new project. Make sure you've got the right type before you save the project at the bottom of this window.

5. For **Template**, click **Browse**, and choose the **Auto-forcasting**template from the list. Click **OK**.

6. For **Data Source**, click **Browse** to select the modeling data source.

   The Browse Data dialog box is displayed. A list of data sets is displayed in the left-side Available tab. These are data sets that are available in CAS and ready for use in a

Model Studio project. The data set for the **baseline sales forecasts** project is not yet listed and needs to be imported.

7. Click the **Import** data button, and then select **Local files**.

8. Click on **Other Locations**. Click **Computer**. Scroll down and double-click the **Workshop** folder. Double-click the **FVVF** folder. Select the **lookingglass_forecast.sas7bdat** table.

9. Select **Open**.

10. Select **Import Item**. Then click **ADD**.

    Note: If there is a note that the table already exists, you can select the radio button for **Replace file** to overwrite it.

11. From the **Available** area. Select the newly loaded **LOOKINGGLASS_FORECAST** in-memory table. The details list the column names and characteristics.

12. Click **OK** and then **Save** to create the new project. The project now appears.

13. Ensure that the project's **Data** tab is selected to assign variable roles.

    A Note on Variable Assignment:
    - Individual variables can be selected for role assignment by either clicking the variable name or by selecting the corresponding check box.
    - Individual variables are **deselected** after their role is assigned by either **clearing their check box** or **selecting another variable's name**.
    - More than one variable can be selected at the same time using the check boxes.
    - Because selecting a variable using the check box does not deselect others, it is easy for new users to **inadvertently re-assign** variable roles. Taking a few minutes to get comfortable with the variable selection functionality is considered a best practice for using the software.

14. The **Txn_Month** variable is assigned to the role of Time for the project.

    Note: Other time intervals are available by selecting the down arrow next to **Month**. The time interval combined with the Multiplier and Shift options indicates that the desired interval of the time series data is one month and that the 12-month annual cycle starts in January. These options can be changed to modify the time index if it is appropriate for your data.

15. **Sale** is the target for the analysis. Click **sale** in the middle variables list panel. In the right property panel, select **Dependent**.

Note: Missing interpretation options enable the user to interpret, or impute, values for embedded missing values in the series. By default, embedded missing values have no value assigned to them.

16. Deselect **sale**. Assign **productline** and **productname,** in that order, to the **BY Variable** role.

17. Additional variables will be assigned to roles. **Price**, **discount**, and **cost** can be useful as explanatory variables in subsequent analyses. Select these three variables, where the order does not matter, and change their roles to **Independent**.

    Note: For each of these variables, accumulation is accomplished by averaging observed values in each month.

18. Change **Usage in system-generated models** to **Try to Use**.

    If I select Force to use, then each of these three variables will be used in every one of the models for every one of the series. I don't want to do that. Let's see some other options-- Try to use and Use if significant.

    Try to Use will test each of the variables in each of the series. So for each model, for each of the 918 series, in the dataset each of the variables will be tested to see if they're statistically significant in the model, and also whether they benefit the model with respect to a fit statistic, such as Akaike's information criterion. So there are two criteria. If a variable passes both of those tests, they'll be used in the model. If the variable doesn't, it won't be used in the model.

    Try to Use is slightly different from Use if significant. Use if significant only tests to see if the variable is statistically significant, and doesn't check to see if it improves the model's fit statistic.

    So I'm going to use Try to use. Now my data are ready to start my pipelines.

Forecasting Using Model Studio in SAS® Viya®
Lesson 01, Section 2 Demo: Loading an Attributes Table to Subset the Time Series

A unique and useful feature in SAS Visual Forecasting is the ability to visualize the modeling data and operate on generated forecasts outside the hierarchy defined by the project's BY variables. The hierarchical arrangement of the modeling data for this project is defined by product characteristics. However, it is routinely useful to be able to explore and operate on forecasts across facets of the data such as customer demographics or geographic regions.

In the last demonstration, we created a project and added data. The only attributes defined were the BY variables. Now we'd like to add other attributes to subset the time series analyses.

In this demonstration, you incorporate the **LG_ATTRIBUTES** table into the **baseline sales forecasts** project and then use the variables in the table to expand the ways that the modeling data can be visualized.

1. From the landing page of SAS Viya, click the tab for **Build Models** and open the **baseline sales forecasts** project that was created previously by double-clicking it.

2. Change the data source type from **Time Series** to **Attributes** by navigating to the data sources panel, selecting the **New data source menu** and then selecting **Attributes**.

   The attributes data set is not yet here in memory. So once again, I need to import it.

   Note: A default attributes table is created when the BY variables are assigned in the project. The BY variables that define the modeling hierarchy are primary attributes for the project.

3. Click on the **Import** data button to import a new data source.

4. Select **Local File** and click **Other Locations**. Click computer. Scroll down and double-click the **Workshop** folder. Double-click the **FVVF** folder. Select **lg_attributes.sas7bdat** to load the table into memory. Click on the radio button to **Replace file** and then click the button to **Import Item**. Then click **ADD**. Set the Show drop down to Available. Choose **lg_attributes** and click **ADD.**

5. The in-memory table, **LG_ATTRIBUTES**, is now the attributes table for the project. The first two attributes are the by variable that I selected earlier, productline, and productname. This table contains two new attributes: a geographic indicator, **Cust_Region**, and a margin flag, **margin_cat**. The margin flag categorizes the profitability of product names as *LOW*, *MED*, or *HIG* (high).

6. Switch to the **Pipelines** tab by selecting it.

   This first pipeline includes a Data node, Auto-forecasting, Model Comparison, and Output.

7. Right-click and run the **Data** node.

   Note: Pipelines are structured analytic flows and are described in detail later in the course.

8. After the **Data** node runs (you will see a green circle with a check mark inside), right-click the green checkmark and select **Time series viewer**.

   The envelope plot shows the aggregated data at the top level of the hierarchy (918 of 918 series). The colored bands illustrate one and two standard deviations around the aggregated series. The available attribute variables are listed in the left filters panel.

9. The available attribute variables are listed on the left side of the window: **product line**, **Product Name**, **Cust_Region**, and **margin_cat**. You can explore time series in the middle level of the hierarchy by expanding the product line attribute. By default, the product line attribute should already be expanded. Visualize demand for the product line series, Line07. Under the **productline** attribute, select **Line07**.

   The plot changes on the fly to show an aggregation of the four product names contained in Line07: Product 21, Product 22, Product 23, and Product 24. Notice that the Envelope Plot changes because it is now relevant for only the four product lines in Line07.

10. Expand the **Cust_Region** attribute.

    There are two customer regions in Line 07. Those are **Pacific** and **Greater Texas**, three in **Pacific** and one in **Greater Texas**.

11. Selecting **Greater Texas** plots the one product name that flows through both Line07 and the Greater Texas region, product line 24.

    Note: You might need to click the series in the right pane for the image to appear.

12. You can select **Reset** to remove the filters that you created based on attributes and return to 918 series displayed.

13. Click the close button on the Time Series Viewer.

Lesson 02

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 3 Demo: Performing Basic Forecasting with a Pipeline

In this demonstration, you perform basic forecasting with a pipeline.

1. Starting from the landing page of SAS Viya, select **Build Models** and open the **baseline sales forecasts** project created previously.

2. Navigate to the **Pipelines** tab.

   The Auto-Forecasting template is the default pipeline template for Visual Forecasting. It consists of the essential steps in a forecasting analysis:
   - accumulates the data into time series
   - automatically identifies, estimates, and selects forecast models for the time series
   - assesses forecasting results
   - publishes results for use outside the pipeline

   Note: If the modeling data are hierarchically arranged, the identification, estimation, and selection steps in the default forecasting pipeline are done on series in the base level of the hierarchy.

   Remember that the Data node was already run, so we see that green circle with a checkmark inside.

3. Select **Run Pipeline** in the upper right corner of the workspace.

   Then we see the circles, the empty circles start up. And by the time we see all circles filled with green ink and a white checkmark, then the pipeline will be completed. Now that it's done, let's take a look at the Auto Forecasting node.

   Note: If you run into problems with this step, make sure that the modeling and attribute tables are loaded in memory. If the server containing in-memory versions of the modeling and attributes table has been shut down since you last opened the project, tables need to be reloaded.

Auto-forecasting Node Results

1. Right-click on the **Auto-forecasting** node and select **Results**.

   Because the Auto-forecasting node is designed to be run with minimal input from the analyst, relatively few options are surfaced for this node. The Auto-forecasting node automatically identifies, estimates, and generates forecasts for the 918 series in the base or product name level of the modeling hierarchy. Most of the forecast models selected for these series are in the ARIMAX family.

For each series, three families of time series models are considered by default: ARIMAX (ARIMA with exogenous variables), ESM (exponential smoothing models), and IDM(intermittent demand models). The champion model for each series is chosen based on root mean square error. Other selection statistics are available in the Model selection criterion option.

Note: To better match course notes, go to the Model Generation area of the Auto-forecasting Node and turn off IDM.

The MAPE Distribution histogram is located in the upper left-hand corner. The distribution of Mean Absolute Percent Error (MAPE) for forecasts in the product name level of the hierarchy can be used to compare the accuracy of different forecast models. Each of the bars represents the proportion of the series that have a specific range of MAPE values. In general, smaller values of MAPE imply greater accuracy. MAPE is an alternative selection criterion supported in the software. We can see where the majority of the MAPE values are located.

The Model Family histogram is located in the upper right. Among each of these 918 series, mousing over the bars reveal the percentage of series that selected an ARIMA or ESM model as its champion.

The Model Type chart, located in the lower left, summarizes systematic variation found in the identification process. Mousing over the bars reveal what percentage of the forecast models selected at least one of the candidate input variables, have a seasonal pattern, and selected a Trend Model. Now inputs were only permitted for the ARIMA models. So this is nearly all of the ARIMA models that had inputs presence. They can be overlapping series, therefore, we can see that the percentage is summed to over 100%.

The Execution Summary, located in the lower right, provides information about results that are potentially problematic, anomalous, or both. We can see there were 918 series. There weren't any series that failed for forecasting. There were only six series with forecasts equal to zero, meaning in the forecast range, the forecasts were all zero. Then there is a lot of summary information about the number of series that had flat forecasts. Flat forecasts means that in the forecast range, the forecast values were a constant.

2. Click the **Output Data** tab above the MAPE Distribution plot.

   Several output tables are created. You can view them by clicking on them.

   You'll notice that there are a unique product line, product name combination for each line. So that's a unique series identified by its product line and product name. And notice that we have multiple lines of data in this data set, each for a different month. So if I scroll down, I can see that the Time ID, the months go from 2012-2016.

And once we get to 2017, we see the actual values is missing. What that means is that this is the forecast horizon. The forecast horizon, of course, doesn't have any actual values, but it can have predicted values and so on. So that's information you might want to obtain from that forecast table.

3. Click the **OUTMODELINFO** data source to open it.

   For each series, the selected model is named and attributes of the model are displayed.

   Once again, we see information for every product line, product name combination. In other words, every series. And in this particular data set, we see the name of the model, or the label for the model, is the type of model that was chosen as the champion model for that particular series. So for Line01, Product01, that particular series, it was an ARIMA model with regression parameters. It is under the ARIMA family. There were no dependent variables here. And we can get information about whether there are seasonal components, whether there are trend components, whether there are inputs presence, and so on.

   So that's information for the Champion model. If I want to see what the competitors were, we can click on the OUTSELECT Data Source. And now you'll see there are three lines for each one of the series. So Line01, Product01 is three different lines. And you can see which of the models were under consideration by looking at the Model column. And then the next column for Selected Status, you can see that the selected row, the selected model for this particular series, as we'd seen before, was the ARIMA model with regression parameters. And you can see why that happened by scrolling farther to the right, each one of these has fit statistics and accuracy statistics calculated. So if we looked at the Mean Absolute Percent Error column, for those first three rows, you can see that the middle value, 3.73, was the smallest of all three. And for MAPE, being absolute percent error, smaller is better, and that is why that particular model won the competition for that particular series.

4. Close the **Results** window.

5. Right-click and open the results of the **Model Comparison** node.

   The Champion Model is the Auto-forecasting model, which is the only one included in the pipeline. WMAE and WMAPE are weighted sums of the MAE and MAPE values across all series. WMAPE and WMAE represent average performance of all the models in a modeling node.

   Note: For the WMAPE and WMAE, the final computation is based on weighted measurements from each time series, where more weight is given to time series with a higher average of the dependent variable.

6. Close the Model Comparison results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 3 Demo: Exploring Generated Models and Forecasts Using the Forecast Viewer

In this demonstration, you will explore generated models and forecasts using the Forecast Viewer. We'll start from **Pipeline 1** in the **Baseline Sales Forecasts** project.

1. Open the Forecast Viewer and explore forecasts for individual Product Names.

    a. Right-click the Auto-Forecasting node and select **Forecast Viewer**.

    Note: Attributes are listed on the left, and what you are seeing in the Envelope plot is listed in the column on the right. What's shown in the plot at this point is a mean of actual values (black hashed line) across all 918 series of Product Names. Range, one standard deviation, and two standard deviations across the 918 series are also shown. Forecast information is not shown here.

    b. Expand **Product Name** under Default Attributes.

    c. Select **Product01**.

    Note: You might need to click the series in the right pane for the image to appear.

    d. The right column heading changes to Series (1 of 918).

    Note: This is one series out of the 918, not the first series. This shows the historical and lead forecasts (line) and actual values (dots) for this individual series. The forecast shown is generated by the champion model for **Product01**.

2. Investigate the Champion model generating the forecast for **Product01**.

    a. Click the Modeling tab.

    b. The Champion-generated model based on the Selection Statistic (MAPE) is **DIAG1_REGARIMA1**. This is an ARIMA regression model including a predictor variable as an input.

    Note: The series of this model included two parameters: p=1 indicates we have an autoregressive model for the first lag, and INPUT indicates that price was an input selected for this particular model. The Selection Statistic that we used was MAPE. So, smaller values are better and that's the reason why REGARIMA became the champion for this particular series. More detailed information like parameter estimates associated with the model can be obtained in generated tables and the Interactive Modeling node, covered later in this course.

3. Navigate around to get comfortable with how the Forecast Viewer works.

a. You are on the Modeling tab.

b. Select **Product02** and **Product03** in addition to **Product01** in the left Attributes column under Product Name.

   Note: The right-hand column heading changes to Series (3 of 918), but the Modeling information doesn't change.

c. Select **Line01:Product02** in the right-hand column to highlight it.

   Note: The Modeling information changes to show the champion and runner-up models for Product02.

d. Click the Forecast tab.

   Note: The plot is a mix of the actual and forecast (blue line and dots) for Line01:Product02 plus the mean, range, and standard deviation measures from the three selected series.

e. Deselect **Product01** and **Product03** under Attributes.

   Note: The forecast and actuals for Product02 are shown. This is confirmed by what's listed in the right-hand column. You may need to click the series in the right pane for images to appear.

f. **Close** the Forecast Viewer.

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 4 Demo: Honest Assessment

In this demonstration, you select models using honest assessment.

1. If it is not still open, reopen the project from the previous demonstration.

2. Click the **Options** ellipsis on the Pipeline 1 tab and select **Duplicate**.

3. Rename the new pipeline by clicking on its **Options** and selecting **Rename**.

4. Rename this pipeline **Honest Assessment Auto**. Click **OK**.

5. Click the **Auto-Forecasting** node to make it active.

6. On the right, in the node options area, expand both the **Model Generation** and **Model Selection** menus.

   Under Model Generation, you can see by default, the Auto Forecasting Pipeline includes just exponential smoothing models and ARIMAX models. Other models that are available to us are IDM, or intermittent demand models, and UCM, or unobserved components models. We won't need to use the IDM models. Those are really only useful when we have a series where many of the time intervals have no data, such as relatively rare events where maybe there are sales of something that don't occur every month. That's not what we have here in these data. So I'm going to leave that box unchecked.

7. Under Model Generation, check the box to **include UCM models**. Uncheck include IDM models.

   So that means that, in addition to checking the ESM models and the ARIMAX models, SAS will also check to see if a UCM model might be the best model for each one of those 918 series.

   Note: UCM models can lead to excessive run time. They are usually reserved for special case or high-value series.

8. We have monthly data. So I'm going to use at least one seasonal cycles worth of data, which includes 12 time points. Under Model Selection, change **Number of data points used in the holdout sample** from **0** (the default) to **12** (a full seasonal cycle's worth of the monthly data) and then either click the return key or click outside of the box.

   When you do this, another box will appear asking you for a percentage of data points to use in the holdout sample. If you also put a value here, the holdout sample will be the smaller of the two.

9. Enter **25** for **Percentage of data points used in the holdout sample**.

   Note: The actual size of the holdout sample is the smaller of the number of data points selected and the percentage of data points. This value can vary from series to series in a project.

10. Leave the model selection criterion as **MAPE** and run this pipeline.

    The Auto Forecasting node you'll notice is going to take a bit longer than the Auto Forecasting node took before. And that's not necessarily because of Model Selection being based on the holdout sample, it is because we've included the UCM models. UCM models cannot be as efficiently run as either ARIMA models or exponential smoothing models. So things will take a little bit longer, but not too much.

11. Right-click and open the results of the **Auto-forecasting** node.

    The ESM model is selected for nearly half of the series. The UCM model accounts for another quarter. Remember that these models were selected on the basis of MAPE on the holdout sample of 12 time points, rather than the fit sample, which was the basis for assessment in the previous pipeline.

12. Close the **Results** window.

13. Right-click and open the results of the **Model Comparison** node.

    WMAE and WMAPE are slightly higher for the honest assessment pipeline than for Pipeline 1. That is to be expected because the data used to assess the models were not the same as the data used to generate the models.

    One thing that you should be aware of, the weighted MAPE is not based on a weighted average of the MAPE for the 918 series using the holdout sample. This is weighted MAPE calculated on the entire sample. So we selected each one of the 918 series. We used as a Selection criterion, the MAPE and the holdout sample. But when we assessed the entire pipeline, the MAPE was recalculated on the entire series, sometimes this could be a little bit misleading or confusing. The consequence of this fact is that you're really not able to compare pipelines or compare nodes in their performance on the holdout sample. So we can select individual models for each individual series using performance on the holdout sample. But when we summarize how well a particular pipeline is doing, we cannot get a result that's based on performance in the holdout sample.

14. Close the Model Comparison results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 5 Demo: Exploring More Pipeline Templates

In this demonstration, you explore pipeline templates other than the default **Auto-forecasting** template. This demonstration proceeds from the end of the previous one.

We already have two pipelines. The auto-forecasting pipeline is Pipeline 1. And another auto-forecasting pipeline is Honest Assessment Auto. But, remember, Honest Assessment Auto used a holdout sample for assessing the best model within each of the series. The next pipeline we'll add uses a different template.

1. From the Pipelines tab in the **baseline sales forecasts** project, click on the plus sign (+) to add a new pipeline.

2. Name the new pipeline **Naïve model forecasts**. The model chosen for all of our time series will be a seasonal random walk model.

3. Select **Browse** from the **Template** drop-down menu. The templates described earlier can be accessed here.

4. Select the **Naïve Forecasting** template and click **OK**.

5. Click **Save** in the New pipeline window.

   A new pipeline is created based on the Naïve modeling node. The subsequent added nodes are described previously.

6. Select the **Naïve model** node and look at the options on the right.

   And you can see, there's always going to be, or nearly always going to be, an option for editing the code. We're not going to be doing that in this course.

   The options on the Naïve modeling node (in the Node options menu on the right) indicate that a Seasonal random walk model will be fitted to each series.

   Note: These models can be useful for providing benchmark measures of forecasting accuracy.

7. Run the **Naïve model forecasts** pipeline by clicking on **Run Pipeline**.

8. Right-click the **Model Comparison** node and select **Results**.

   Holdout samples are not options within the Naïve Model forecasting node, so WMAE and WMAPE are based on the entire samples of each series.

And now we can see the weighted MAPE value and compare it with the previous results.

9. Close the results window.

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 5 Demo: Pipeline Comparison

In this demonstration, you compare pipelines and use accuracy and fit statistics to determine a champion model. This demonstration proceeds from the end of the previous one.

So with a project open, we have now three pipelines: Pipeline 1, Honest Assessment Auto, and Naïve Model Forecasts.

1. Click the **Pipeline Comparison** tab.

   The results indicate that Pipeline 1 is selected as the champion. The Champion pipeline is selected by default using weighted MAPE. So if we look at the Champion column, there is a star where the Champion is. And if you look at the actual values of weighted MAPE, you can see why a pipeline was the champion

   Note: The declaration of a champion pipeline is important for subsequent steps in the forecasting workflow. The forecast table that can be exported from the project is based on the models in the champion pipeline. Also, any overrides that are set will be implemented on the champion model forecasts. Overrides are described in detail later.

   Recall that Pipeline 1 did not use a holdout sample and, therefore, the models were not selected using honest assessment. Because the purpose of these models is forecasting, Pipeline 1 should be excluded from consideration as a champion model. Based on MAPE, the Honest Assessment Auto pipeline would beat the Naïve Model Forecasts pipeline. Manually select that pipeline as the champion pipeline.

2. Right-click the **Honest Assessment Auto** pipeline and select **Set as champion** from the drop-down menu. The pipeline has changed. Click OK on the pop-up window if it appears.

3. To compare summary results and diagnostics across pipelines, select check boxes next to the **Honest Assessment Auto** and **Naïve Model Forecasts** pipelines and then click **Compare**.

   You can now compare the MAPE distributions and Execution Summary results across all selected pipelines in one window.

4. Click **Close** to exit the compare window. If you changed the champion, reset Pipeline 1 as the champion model.

   Note: The pipeline selection criterion can be changed, and the automated choice of the champion pipeline can be overridden. For example, to manually change the champion pipeline, clear the box for **Pipeline 1**, click the **Project pipeline menu**

icon (the three vertical dots in the upper right), and select **Set as champion**. Selecting a new Champion would recreate the data used and override the project.

Forecasting Using Model Studio in SAS® Viya®
Lesson 02, Section 6 Demo: Creating Custom Models with the Interactive Modeling Node

In this demonstration, you will create custom models with the Interactive Modeling node. Start from **Pipeline 1** in the **Baseline Sales Forecasts** project.

1. Add the Interactive Modeling node to the pipeline.

   a. Select **Nodes** (to the upper left of the pipeline diagram).

   b. Expand **Postprocessing**.

   c. Left-click and then drag and drop the **Interactive Modeling** node on the link between the Auto-Forecasting node and the Model Comparison node.

      Note: A plus symbol will appear when you hover over the link between the two existing nodes. This is an indication that it's feasible to drop the node in this part of the Pipeline.

   d. Right-click the **Interactive Modeling** Node and run it.

2. Open the Interactive Modeling node and explore the Forecast tab functionality.

   a. Right-click the **Interactive Modeling** node and open it.

      Note: There are three tabs: Forecast, Modeling, and Series Analysis. It should open showing the Forecast tab results, but if it doesn't, click the Forecasting tab.

   b. Expand **Default** attributes.

   c. Select **Product01**.

      Note: You might need to click the series in the right pane for the image to appear.

      Note: The functionality on the Forecast tab is the equivalent of the Forecast tab in the Forecast Viewer. Recall that for **Product01**, there is evidence of promotion or Price effects, but no discernable seasonality or trend.

3. Explore the Modeling tab functionality.

   a. Click the Modeling tab. We've seen three of the models shown in the Forecast Viewer, but the champion seems to have changed into a **PREDECESSOR** model, and its type is Inherited.

Note: The IM node is running another Auto-Forecasting node with default settings 'under the hood'. In this case, the models shown (the Model Selection list for Product01) are identical to the three we've seen in the Forecast Viewer plus the Champion model from the predecessor Modeling node. Because there are two equivalent Auto-Forecasting runs with default settings in the Pipeline, the **PREDECESSOR** Champion model and the **DIAG1_REGARIMA** model are the same. Note the in-sample MAPE values. This can change when the Interactive Modeling node is attached to an Auto-Forecasting node whose default setting have been changed or when it's attached to a different type of Modeling node.

b. Select the **DIAG1_REGARIMA1** model.

c. Click the **View diagnostic plot/table** button.

d. Select **Model Fit** > **Parameter Estimates**.

e. Select the **DIAG1_ARIMAX1** model. The Parameter Estimates table changes to show results for the selected model. It has two estimated parameters.

f. Select the **View diagnostic plot/table** button > **Model Fit** > **Statistics of Fit**. This table shows fit statistics associated with the selected model. Scroll over in the table to view the available fit measures.

Note: In addition to MAPE, there are over 50 reported statistics of fit.

g. Select the **DIAG1_REGARIMA1** model.

h. Select the **View diagnostic plot/table** button > **Basic error analysis** > **Prediction Errors**.

Note: There are a couple of large residuals that lie outside the Two Standard Error band. These might warrant further investigation.

i. Select the **View diagnostic plot/table** button > **Error autocorrelation analysis** > **White noise probability test (log scale)**.

Note: Students who are comfortable with ARIMA model identification will find familiar plots like the ACF, PACF, and IACF for the residuals of the model among the diagnostics listed. We'll find some of these same diagnostics on the Series Analysis tab, but there the generated ACF, PACF, and IACF are based on the time series. The white noise test indicates that the model's residuals are not white noise. Some spikes exceed the 0.05 threshold line. This tells us that it might be useful to add terms to the model.

4. Create a custom model specification based on a generated specification.

a. Click the **View selected model** button in the top right of the Model list. The Model Details have similar information to the Parameter Estimates table plus information about differencing and the transfer function specification for inputs.

b. Click the pop-up window.

c. Click the **Copy** icon.

   Note: Generated models cannot be directly edited. First, a copy of the model is made and then the copy can be edited.

d. Name the model **myREGARIMA1_1**.

e. Select **Independent Variables**.

f. Select **Edit** (pencil) next to the pre-selected **input** variable.

g. Change Simple Numerator Factors from 0 to **1**.

   Note: We've changed the way that Price enters the model with a Numerator order 1 term. This specifies that when Price jumps in a given month, Sales are impacted in that month and in the month following.

h. Select **Save** > **Save**.

   Note: The Custom model myREGARIMA1 has the best MAPE of the models listed, but it is not declared the Champion model for Product01. It can be made the Champion using the following steps:

   I. Make sure **myREGARIMA1_1** is selected and click the **Set as champion** icon.

   II. Select **Commit Changes**.

   III. Select **Commit**.

   IV. Click the Forecast tab.

   Note: The forecasts shown are generated by the new Champion, **myREGARIMA1_1** custom specification.

5. Look at Series Analysis diagnostics.

   a. Click the **Series Analysis** tab.

Note: A Series Analysis is pre-loaded for the dependent variable, Sale, for the selected series, Line01:Product01. There are three tabs on the left of the main window: Filters, Model Inputs, and Analyses. It's opened in Series Analysis by default.

b. Click the **Add analysis** (plus sign) button next to the Unit Sale icon.

Note: In addition to standard diagnostics like the ACF, PACF, and White Noise test, available analyses include decompositions and seasonal adjustments.

c. Scroll through the pop-up list to see the available diagnostics for the series.

Note: Additional series can be analyzed by dragging and dropping them from the Model Inputs column into the Series Analysis diagram.

6. Create a new custom model.

   a. Click the **Modeling** tab.

   b. Click the **Create Model** icon.

   Note: Selecting one of the listed families of models provides a point-and-click interface for creating a new custom model and adding it to the model selection list. This is a way to create a custom model that is not based on an existing Generated model.

   c. Close the Interactive Modeling Node

Lesson 03

Forecasting Using Model Studio in SAS® Viya®
Lesson 03, Section 1 Demo: Generating Hierarchical Forecasts with the Default Settings

In this demonstration, you use the BY variables, **productline** and **productname**, to perform hierarchical forecasting using the Hierarchical Forecasting node.

1. Starting from the landing page of SAS Viya, create the **LG hierarchy forecast** project. This project will use the same in-memory tables and variable metadata as the **baseline sales forecasts** project created earlier. For convenience, a summary of the project creation steps is below.
    - Navigate to **Build Models** and start a new project.
    - Name the project **LG hierarchy forecast**, set the type to **Forecasting**, and provide a reasonably detailed description. Set template to **Auto-forcasting**.
    - Navigate to the **LOOKINGGLASS_FORECAST** table on the Available tab and click **OK**.
      Note: If the **LOOKINGGLASS_FORECAST** table is not on the Available tab, you need to load it into memory following steps that were shown previously in the course.
    - Save the new project.
    - Assign variable roles. **Txn_Month** is the time variable, the dependent variable is **sale**, and the BY variables are **productline** and **productname**. Set **Reconciliation level** to **productline**.
    - Select and edit **price**, **discount**, and **cost**. Set the role of these variables to **Independent** and change **Usage in system-generated models** to **Try to Use**.

2. Navigate to the Pipelines tab, and select **Run Pipeline**. This pipeline is identical to the ones used previously, but running it now allows you to compare this pipeline to the Hierarchical pipeline that is used later.

3. Click the plus (+) to add a new pipeline.

4. Name the new pipeline Pipeline 2.

5. Select **Hierarchical Forecasting** for the template and click **Save**.

6. In contrast to the Auto-forecasting node, the Hierarchical Forecasting node allows extensive customization. Select the **Hierarchical Forecasting** node and scrutinize the options on the right side of the screen.

7. Click on **Run Pipeline**.

8. When it finishes running, right-click the **Hierarchical Forecasting** node and select **Results**.

9. Results are given on both the **productline** (middle) and **productname** (base) levels of the hierarchy. Model Type and Model Family results are added to the previously introduced diagnostics.

   Note: Recall that the modeling hierarchy was set when the **productline** and **productname** variables were assigned as BY variables in the project.

   The average Weighted MAPE over all of the series is shown for the productline hierarchy level, and the Weighted MAPE is shown for on the hierarchy level productname. Within each of those, you can see the MAPE Distribution of all the possible series used.

   The Model Family information indicates that this is still a selection among simple models. The best models selected were mostly ESM models.

   Looking at the Model Types, mousing over the bars reveals the percentage of the models that used the independent, or input, variables, had seasonal components, and had a trend. And we can look at the same information on the hierarchy level productname.

10. Close the results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 03, Section 2 Demo: Adding Combined Models to the Hierarchical Forecasting Pipeline

The previous demonstration generated forecasts for all series in the three-level hierarchy under the default settings for the Hierarchical Forecasting node. In this demonstration, you try to improve the fit of the forecasts by adding combined models to the pipeline. For each series, the combined model combines the generated forecasts from default families of models considered for that series to produce a new forecast. The default combination method is a simple average of forecasts.

This demonstration proceeds from Pipeline 2, created in the previous demonstration.

1. Expand the **Nodes** menu on the left side of the workspace. Expand **Forecasting Modeling**.

2. Click and then drag and drop a **Hierarchical Forecasting** node on top of the **Data** node.

3. Right-click and rename the new Hierarchical Forecasting node to **Hierarchical Forecasting with combined models**. Click **OK**.

4. Select the **Hierarchical Forecasting with combined models** node, and expand the **Model Generation** options on the Node Options panel on the right.

   Notice that the default options for Model Generation include ARIMA and ARIMAX models, and ESM models. The sliders for UCM and external models are not slid to the right, so those models are not included.

5. Scroll down to the **Include combined models** option and slide the toggle to **on**.

   With combined models, you can average the results from all of the ARIMA and ESM models. Often, the combined models can perform better in forecast than the ARIMA models and the ESM models individually.

   Keep the default methods for combination, a straight average of all the models. Keep all the other statistics and options as they are by default.

6. Select **Run Pipeline** to run the updated components.

7. Right-click the **Hierarchical Forecast with combined models** node and open **Results**. The Model Family results show that the majority of forecast models selected for the series in the base and middle levels of the hierarchy are generated by Combined (comb) forecasts.

The aggregated, or weighted, MAPE measures have improved, relative to the forecasts generated under the default settings, for both levels of the hierarchy.

Mousing over the bars shows the percentage of combined models that were chosen as the best model for the series. We can also see how many were ARIMA, and the exponential smoothing.

8. Close the results.

9. Right-click on the **Model Comparison** node and select **Results**. The Hierarchical Forecasting with combined models node is the champion for the pipeline.

10. You can compare results at the base level of the hierarchy across the two pipelines in the diagnostics. Close the results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 03, Section 2 Demo: Selecting Models Based on Forecast Accuracy

One potential issue with the selection of the **Hierarchical Forecasting with combined models** node as the champion in the previous demonstration is that the selection criterion reflects how well the models fit the series in the training data.

In this demonstration, you split each time series in the data into two parts: training and validation. The Champion modeling node is selected based on aggregated, out-of-sample performance, or accuracy. This demonstration proceeds from Pipeline 2, created in the previous demonstration.

1. Select the default **Hierarchical Forecasting** node, expand the right Node Options panel, and expand the **Model Selection** options.

2. Change the Model selection criterion to **RMSE (Root Mean Squared Error)**.

3. Change the number of data points used in the holdout sample to **12**. The number of data points is set to 12 because this is monthly data, and the holdout sample should include at least one seasonal cycle of data.

4. Change the percentage of data points used in the holdout sample to **25**. The holdout sample typically includes a maximum of about 25% data points.

   Note: When you choose both criteria for a number of data points and percentage of data points, the smaller number of observations generated by either of these restrictions is used as the holdout sample size for each series.

5. Select the **Hierarchical Forecasting with combined models** node, expand the right Node Options panel, and expand the **Model Selection** options.

6. Change the Model selection criterion to **RMSE (Root Mean Squared Error)**.

7. Change the number of data points used in the holdout sample to **12** and the percentage of data points used in the holdout sample to **25**.

8. Rerun the pipeline by clicking on **Run Pipeline**.

9. Right-click on the **Hierarchical Forecasting with combined models** node and click on **Results**. The Model Family and Model Type results are similar, but the MAPE distributions and aggregated MAPE values have changed over the base and middle levels of the hierarchy.

   The distributions of the models that are selected are slightly different. The combined models within productline are not necessarily the majority of the models that were selected. These diagnostics are now based on residuals generated over the holdout sample region for each series. That is, they are accuracy statistics. In general, the MAPE values tend to be a bit larger when working on a holdout sample.

10. Close the results.

11. Right-click on the **Model Comparison** node and select **Results**. Although the choice of the champion pipeline has not changed, this result is more relevant. The pipeline with the models that extrapolate best onto data that they have not seen before is chosen as the champion.

12. Close the **Model Comparison** results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 03, Section 2 Demo: Sharing a Custom Pipeline via the Exchange

In this demonstration, you use the pipeline developed in the previous demonstration as a custom template for other forecasting projects. The Exchange provides a repository for collecting and sharing project objects with others. This demonstration proceeds from Pipeline 2, created in the previous demonstration.

1. From Pipeline 2 in the **LG hierarchy forecast** project, click on Options. Select **Save to The Exchange**.

2. Name the pipeline **LG Hierarchical Forecasting with Combined Models**. Add a description and click **Save**.

   Note: Providing a representative name and a detailed description is always useful.

3. On the left side of the window, click the icon for **The Exchange**.

4. Under **Templates** on the left panel, expand **Pipelines** and select **Forecasting**. The custom pipeline that was saved from the **LG hierarchy forecast** project is now available to others.

5. Return to the projects area of **Model Studio**.

Forecasting Using Model Studio in SAS® Viya®
Lesson 03, Section 3 Demo: Implementing the Hierarchical Modeling Node and Generating Reconciled Forecasts for the Project

This demonstration proceeds from the previous demonstration, Sharing a Custom Pipeline via the Exchange.

Creating a Pipeline, Adding, and Running the Hierarchical Modeling Node

1. Open the **LG hierarchy forecast** project if needed.

2. Select the **Pipelines tab**, and then select **Add new pipeline (+)**.

3. Name the new pipeline **Hierarchical Modeling**.

4. Next to template, select the **Browse** button, and then expand **Base Forecasting**. This will create a pipeline with just a Data node. We will be adding the Hierarchical Modeling node to this pipeline.

5. Select **OK**, and then select **Save**.

6. On the left pane, select the **Nodes menu**, and then expand **Forecasting Modeling**.

7. Left click, drag and drop a **Hierarchical Modeling node** on top of the **Data** node.

8. Run the pipeline.

Exploring the Hierarchical Modeling Node

1. When the run completes, right click the **Hierarchical Modeling** node and select **Open**.

   The three levels in the data hierarchy are listed in the table. Additionally, a note reports that the Reconciliation process was completed. Weighted MAPE, denoted WMAPE, and Reconciled WMAPE values are listed for each level of the hierarchy.

   Note: WMAPE is a measure that aggregates MAPE values associated with the champion model's forecast for each series in the corresponding level of the hierarchy. Reconciled WMAPE is a measure that aggregates MAPE values on the reconciled forecasts in the corresponding level of the hierarchy.

2. Check the box next to the **TOP** level, and then select the **Forecast Viewer** button.

3. Select **Top** in the right-hand-panel. This will activate the plots on the page.

The plot shows the actuals and forecasts associated with the data at the top level of the data hierarchy. Top level **SALES** seem to be trending up nicely. Also, the total **SALES** series has a seasonal pattern that the champion model captures and extrapolates into the lead forecast horizon.

Note: The TOP level data is an aggregation of Unit Sales (SALE) across all BY groups in the project's data. It is a single series that results from the process of aggregation and represents total unit sales across all **productname** or **productline** series.

4. Expand the drop-down list next to **Hierarchy level** and select **productline**.

   This takes you to the **Forecast Viewer** for the pipeline associated with the middle of the project's data hierarchy. There are 270 **productline** series in the middle level of the data hierarchy. You can explore forecasts and actual values for different **productline** series using steps shown previously in the Exploring Generated Models and Forecasts Using the Forecast Viewer demonstration.

5. Expand the drop-down list next to **Hierarchy level** and select **productname**.

   This takes you to the **Forecast Viewer** for the pipeline associated with the bottom level of the project's data hierarchy. There are 918 **productname** series in the middle level of the data hierarchy.

6. Select **Close**.

Exporting the Reconciled Forecasts for the Project

The three Forecast Viewers that we just explored are associated with three different pipelines, one for each level of the data hierarchy. Each pipeline is generated using an **AUTO-FORECASTING** template.

Recall that when this project was created, the Reconciliation Level was set to **productline**. When the **Hierarchical Modeling** node was run, automatic model generation, model selection and forecast generation were accomplished for all three levels of the data hierarchy. As a final step, the forecasts at the **Top** and **productname** levels were reconciled to the forecasts at the **productline** level. Now, we're going to export the generated reconciled forecasts for this project.

Saving the Forecast Data from the Top-level Pipeline

1. Select the check box next to the **TOP** level in the **Hierarchy Levels** table.
2. Select the **Open Pipeline** button on the top right.
3. Inside the TOP Pipeline, select the **Nodes** button and then expand **Miscellaneous**.
4. Left click, drag, and drop a **Save Data** node on top of the **Auto-forecasting** node.

5. Select the **Edit save options** button in the properties of the **Save Data** node.

   Note: the functionality of the **Save Data** node is discussed in more detail later.

6. Expand **Time Series** under **Output Tables** and select **Forecasted values**.
7. Check the box next to **Include in output**.
8. Rename the table; **Top_RECFOR**.

   Note: Because we've set reconciliation to the middle, or **productline**, level, the **Top** level forecasts are impacted by the process of reconciliation. **Top** level reconciled forecasts are the **Predicted Values (PREDICT)** in the **OUTFOR** table from this level of the hierarchy.

9. Select the Browse button to select an output CAS library.
10. Expand **cas-shared-default** and select the **Public** library.
11. Select **OK**, and then select **OK** again to close the save options.
12. Close the **Top Pipeline**.

Saving the Forecast Data from the *productline* Pipeline

1. Select the check box next to the **productline** level in the **Hierarchy Levels** table.
2. Select the **Open Pipeline** button on the top right.
3. Inside the productline Pipeline, select the **Nodes** button and then expand **Miscellaneous**.
4. Left click, drag, and drop a **Save Data** node on top of the **Auto-forecasting** node.
5. Select the **Edit save options** button in the properties of the **Save Data** node.
6. Expand **Time Series** under **Output Tables** and select **Forecasted values**.
7. Check the box next to **Include in output**.
8. Rename the table **productline_OUTFOR**.

Note: Because we've set reconciliation to the middle, or productline, level, the productline level forecasts are not impacted by the process of reconciliation; productline level statistical forecasts are the Predicted Values (PREDICT) in the OUTFOR table from this level of the hierarchy.

9. Select the Browse button to select an output CAS library.
10. Expand **cas-shared-default** and select the **Public** library.
11. Select **OK**, and then select **OK** again to close the save options.
12. Close the **productline Pipeline**.

Saving the Forecast Data from the *productname* Pipeline

1. Select the check box next to the **productname** level in the **Hierarchy Levels** table.
2. Select the **Open Pipeline** button on the top right.
3. Inside the Productline Pipeline, select the **Nodes** button and then expand **Miscellaneous**.
4. Left click, drag, and drop a **Save Data** node on top of the **Auto-forecasting** node.
5. Select the **Edit save options** button in the properties of the Save Data node.
6. Expand **Time Series** under **Output Tables** and select **Forecasted values**.
7. Check the box next to **Include in output**.
8. Rename the table **productname_RECFOR**.

The **productname** level forecasts are impacted by the process of reconciliation; **productname** level reconciled forecasts are the **Predicted Values (PREDICT)** in the **OUTFOR** table from this level of the hierarchy.

9. Select the Browse button to select an output CAS library.
10. Expand **cas-shared-default** and select the **Public** library.
11. Select **OK**, and then select **OK** again to close the save options.
12. Close the **productname Pipeline**.

Exporting the Reconciled Forecasts from the Project

1. Close the **Hierarchical Modeling** node.

2. **Run** the Hierarchical Modeling Pipeline.

   Tables exported by the Save Data node are promoted by default.

3. Select the **Applications** menu and then select **Explore and Visualize**.

4. Select **New report**.

5. Click **Data Select Menu**, select **Add Data**.

   The three in-memory reconciled forecast tables we exported from our project are available here for further processing and reporting.

6. Click Cancel and return to Model Studio.

Lesson 04

Forecasting Using Model Studio in SAS® Viya®
Lesson 04, Section 1 Demo: Adding the Attributes Table to a Project

Attributes are useful for visualizing the data outside the dimensions defined by the modeling hierarchy. This demonstration proceeds from the **LG hierarchy forecast** project created in the previous lesson.

1. Start from the landing page of **SAS Viya,** and open the **LG hierarchy forecast** by double-clicking it.

2. Click the **Data** tab. Click the **New data source menu** button and select **Attributes**.

3. Select **LG_Attributes** from the list of available data sources. Click **OK**. The **LG_Attributes** table is now the attributes table for the project.

   Notice that there is a list of attributes. The attributes are productline and productname, which you should recognize as the BY variables from before. BY variables can be thought of as special cases of attributes. You can think of an attribute as a way of being able to drill down, describe, or summarize the times series that are being modeled and forecasted.

   So for productline, there are multiple product lines, multiple product names. And from the LG_ATTRIBUTES table, we added the Cust_Region and margin_cat, which was LOW, MED, and HIG. Those four variables together can be used to allow us to drill-down into the time series of interest.

   In particular for this demonstration, we can apply overrides just to the time series that we need overrides to be applied to. We haven't done any forecasting yet. We have two pipelines: Pipeline 1 and Pipeline 2. And at this point, you'll notice that Pipeline 2 and Pipeline 1 are not run, so we need to rerun the pipelines.

4. Click the **Pipelines** tab, and open **Pipeline 2**. Rerun the pipeline.

   Before I run Pipeline 1, remember that we made some changes to Pipeline 2.Specifically, we made some changes by using a holdout sample, and we calculated the accuracy statistics based on the holdout sample. When you do accuracy statistics on a holdout sample, the accuracy statistics don't look as good. In order for me to be able to compare these two pipelines, I really need to do the same type of modifications to this pipeline as I did to Pipeline 2.

5. Open **Pipeline 1** and perform the modifications that you made to **Pipeline 2** earlier.
   - Click on **Auto-forecasting** and expand the Node options.
   - Expand **Model Selection**.
   - Change the model selection criterion to **RMSE(Root Mean Square Error)**.
   - Change the number of data points used in the holdout sample to **12**.

- Change the percentage of data points used in the holdout sample to **25**.
- Rerun the pipeline.

6. Select **Pipeline Comparison**. Pipeline 2 is the champion pipeline. Forecasts shown on the Overrides tab are generated by the champion node from the champion pipeline in a project.

7. Click the **Overrides** tab. The plot shows an aggregation of the 918 series in the base level of the hierarchy.

   Now, you'll notice that the attributes are to the left. We can filter based on those attributes. The attribute of productline is open already. There are five different lines, Line 02, 03, 04, 07, and 08. Productnames, we can expand those and see the five different products. Customer Region, there are five regions: South, Great Lakes, Pacific, Mid Atlantic, and Greater Texas. And of course, we have our three categories for margin_cat, LOW, MED, and HIG. You can notice under the Forecast Overrides table, we have statistical forecasts based on our models, based on our champion model, for the months January 2017 through December of 2017.

Forecasting Using Model Studio in SAS® Viya®
Lesson 04, Section 1 Demo: Applying Overrides to Generated Forecasts

This demonstration proceeds from the previous demonstration. Attributes can be helpful in post-modeling tasks such as applying overrides.

The Overrides functionality basically works in two steps: creation and implementation. In the following steps, two overrides are created:

- Forecasts in the South customer region will be reduced by 20% for the first three months of 2017 to accommodate a pending strike among delivery drivers.
- High-margin products in the Greater Texas region will be increased by 15% in response to pending promotional activity that will occur in July of 2017.

These overrides will be implemented, and an impact analysis of their effects on the model's forecasts will be reviewed.

1. To implement the first override, expand the **Cust_Region** attribute and select the **South** region. The plot changes on the fly to show an aggregation of the **197 productname** series in the South region.

2. Right-click the **Override** cell under 01/01/2017 and select **Override Calculator**.

3. Click **Filter** and name the item **South Override**.

4. Click **Properties**. Add **02/01/2017** and **03/01/2017** to apply the override to the first three months of 2017 using the plus button +. Click **OK**.

5. Because the goal is to reduce forecasts in the South region by 20% during the time range specified above, select **Adjust based on an existing forecast value** and then select **Final Forecast**.

   Note: In this case, final forecasts are statistical forecasts that have been adjusted for reconciliation.

6. Set **Aggregate final forecast lock** to **on**.

   Note: Here, the forecast lock is a restriction on the aggregated final forecast of all **productname** series in the South region. Forecasts for individual series in the override group are free to vary, but they must sum to the override values.

7. Set **Adjustment** to **-20%**. Click **OK**.

8. Click **OK**.

9. The overrides are currently pending. Right-click on any of the three override cells and select **Submit All**. And now, we see the override values are now blanked out, because we've already applied the overrides. The final forecast is now modified, it's no longer an override.

   The second override is a 15% increase for high-margin forecasts in the Greater Texas region in JUL2017.

10. Select **Reset all** from the attributes menu on the left, and then select the **Greater Texas** region and the **high (HIG) margin** category. The plot changes to show forecasts and actual values of the 16 high-margin series that flow through the Greater Texas region.

11. Select the **Override** cell under **07/01/2017**, and right-click it to access the Override Calculator.

12. Set **Aggregate final forecast lock** to **on**.

13. Select **Adjust** based on an existing forecast value. Then select **Final Forecast.**

14. Change the **Adjustment** value to **+15%**. Click **Filter** and name this override **OverrideTXHIG**.

15. Click **OK**.

16. A message box might appear, warning about pending overrides. If it does appear, select **Submit All**. If not, you have created a pending override. Right-click the cell with the pending override value, and select **Submit All**. The final forecast and the forecast plot now reflect the JUL2017 promotion override.

17. Click the **Override Management** tab. The newly created override is added to the list. Overrides can also be modified from here. The Override Calculator and the Delete overrides button are on the top right of this page.

Forecasting Using Model Studio in SAS® Viya®
Lesson 04, Section 1 Demo: Resolving Override Conflicts

Override conflicts occur when two or more overrides create a forecasting outcome that is infeasible for one or more time intervals. Conflicts arise with locked forecast overrides. In this demonstration, you will add another override to forecasts for series in the South region for the first month of 2017 to illustrate a conflict. This demonstration proceeds from the end of the previous one.

Assume that you have information that LOW margin series in the South region are somehow exempt from the pending strike for the first month of 2017, and that these products are also going on promotion in this month. The net effect of these two phenomena is hypothesized to be an increase of 60%.

1. Back on the **Overrides** tab, click **Reset all**, and then select the **South** customer region and the **LOW** margin category. The plot changes to show the 141 time series in this cross section of the data.

2. Right-click the **01/01/2017 Override** cell, and access the Override Calculator.

3. Click **Filter** and name the item **OverrideSouthLOW**.

4. Click **Properties**. Set the adjustment to the final forecast to **+60%**, and **lock the aggregate final forecast** for this subset of series.

5. Click **OK**.

6. Right-click any **Override** cell, and select **Submit all**.

7. The two locked overrides submitted for JAN2017 on the cross section of **South** and **South and LOW margin** have created an infeasible final forecast outcome. The two options for resolution are listed below. If the Conflicts Detected box does not appear, go back and make sure that you locked both of the previous overrides.

8. Select **Resolve Automatically**.

   Note: Selecting **Resolve Manually** takes you back to the Override Calculator to implement a conflict solution. Selecting **Resolve Automatically** calls an optimization algorithm to find a feasible solution for the conflict that is as close to the desired override restrictions as possible.

9. Right-click the **01/01/2017 Override** cell, and select **Impact Analysis**.

   The impact analysis for Group 3 (the 141 LOW margin series in the South region for JAN2017) shows that the final forecast is a compromise between the Previous Final

Forecast (first override) and the second override, applied above. The Delta shows the net effect of the two overrides.

10. Select **Filter3** (or whichever filter is associated with Group 3) to see the plot of the final forecasts for these 141 series.

Forecasting Using Model Studio in SAS® Viya®
Lesson 04, Section 1 Demo: Exporting Forecasts

Following the override process, the final forecasts from the champion pipeline are consistent with business knowledge and are ready to be disseminated. Making the forecasts available to other team members and project stakeholders is straightforward. This demonstration proceeds from the end of the previous one.

1. From the Overrides tab in the **LG hierarchy forecast** project, click **More** (the "snowman") and select **Export all data**.

2. Select the **Public** directory or another directory to which you have Write access. Keep the default name for the exported table, **LG Hierarchy Forecast_OUTFOR**. Click **Export**.

   Note: The **Promote table** option is selected. This means that the table is accessible by other team members and in other tools, such as SAS Visual Analytics.

3. Navigate to **Explore and Visualize Data**. Click **New Report**.

4. This functional area provides access to SAS Visual Analytics. Click **Data Source Menu** then choose **Add Data**. The exported data are loaded in memory and are available.

5. Start typing the name of the saved data set in the search bar.

6. Select **LG HIERARCHY FORECAST_OUTFOR** from the Available area and click **Add**.

7. Click and drag the **Prediction Errors** variable into the workspace. The default chart option for this variable displays a histogram of the forecast errors.

   Notice the prediction errors are symmetrically distributed around the value 0.

8. Click the **More** button in the upper right and select **close report**. You do not need to save this report. Return to Model Studio.

Lesson 05

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 1 Demo: Incorporating More Filters into the Time Series Viewer

In this demo, we create more filters, beyond the primary and secondary attribute variables, for use in the Time Series Viewer.

1. Starting from the SAS Viya landing page, create the **Additional Topics** project. This project will use the same in-memory tables and variable metadata as the baseline sales forecasts project created earlier. For convenience, a summary of the project creation steps is below.
    a. From the Applications Menu drop-down menu, select **Build Models**.

    b. Click the **New Project** button.

    c. Name the project **Additional Topics**, set the type to **Forecasting**, and provide a reasonably detailed description.

    d. Select **Auto-forecasting** as the template.

    e. Navigate to the **LOOKINGGLASS_FORECAST** table on the Available tab and click **OK**. Then click **Save**.

    Note: If the **LOOKINGGLASS_FORECAST** table is not on the Available tab, you need to load it into memory following steps that were shown previously in the course.

    f. Assign variable roles. **Txn_Month** is the time variable, the dependent variable is **sale**, and the BY variables are **productline** and **productname**. Set Reconciliation level to **productline**.

    g. Select **price**, **discount**, and **cost**. Set the role of these variables to **Independent** and change Usage in system-generated models to **Use if significant**.

2. Click the **New data source** menu button and select **Attributes**.

3. Select **LG_Attributes** from the list of available data sources. Once again, if that is not listed as available, you can always reimport it.

4. Click **ADD**. The **LG_Attributes** table is now the attributes table for the project.

5. Click **Descriptive statistics** under Attributes. Select **MEAN** and **STDDEV** from the attribute list. Change the drop-down selection under Display attribute by default to **Yes**.

6. Click **Model attributes** under Attributes. Select **_SEASONAL_**, **_TREND_**, and **_INPUTS_** from the attribute list. Change the drop-down selection under Display attribute by default to **Yes**.

7. Click **Forecast attributes** under Attributes. Select **MAE** and **MAPE** from the attribute list. Change the drop down selection under Display attribute by default to **Yes**.

8. Navigate to the Pipelines tab and select **Run Pipeline**.

9. Right-click the Data node and select **Time Series Viewer**.

   Right now we see all 918 series, some summary statistics along with the envelope plot in the middle.

10. Expand the Descriptive statistics filter. Expand the Mean Value of Series filter.

11. Using the slider, set the lower and upper bounds to be **185** and **300** respectively. You can double-click the slider endpoint to open a text box where you can enter the exact value for the endpoint.

    Now I see an envelope plot just to the series, whose mean values are between 185 and 300.

12. Expand the Standard Deviation of Series filter. Using the slider, set the lower and upper bounds to **25** and **50** respectively. You can double-click the slider endpoint to open a text box where you can enter the exact value for the endpoint.

    Now the number of series that fit both of those criteria are 45.

13. Click **Close** to return to the pipeline view.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 1 Demo: Using Filters in the Forecast Viewer

In this demo, we use more filters, beyond the primary and secondary attribute variables, within the Forecast Viewer. This demo continues from the **Additional Topics** project created in the previous demonstration.

1. From the opening page of Model Studio, select the **Additional Topics** project.

2. From the pipeline view, right-click the Auto-forecasting node and select the **Forecast viewer**.

3. Expand the Model attributes filter. Expand the Seasonal Model filter.

   Zero means no seasonal parameter in the model, and one means there is a seasonal parameter.

4. Select **1** to filter only to models containing Seasonal components.

5. Expand the Trend Model filter. Select **1** to filter to models containing a Trend component.

6. Expand the Inputs Present filter. Select **0** to filter to models not containing Input variables (exogenous components).

7. Expand the Forecast attribute filter. Expand the Mean Absolute Percent Error filter.

8. Using the slider, set the lower and upper bounds to be **5** and **5.5** respectively. You can double-click the slider endpoint to open a text box where you can enter the exact value for the endpoint.

9. On the right, select the first series that matches the filters. Click **Actuals**, **Predicted**, and **Confidence limit** to control what appears on the envelope plot.

10. Click the **Save Filter** icon above the Attributes section to save the current filter combination. In the text box, call this filter **Custom Filter**. This filter is now saved for use later and within the Overrides section.

11. Click **Close** to return to the pipeline view.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 2 Demo: Exporting Automatically Generated Tables

In this demo, we will export automatically generated tables from Modeling nodes and the Project. This demo continues from the **Additional Topics** project created in the previous demonstration.

1. From the opening page of Model Studio, select the **Additional Topics** project.

2. Export an **OUTFOR** table from a Hierarchical Forecasting Node.

    a. Click the **Pipelines** tab.

    b. Select **Add new pipeline** (+).

    c. Name the pipeline **Hierarchical** and then click the **Browse** button.

    d. Select a **Hierarchical Forecasting** template, and click **OK**.

    e. Select **Save** and then the **Run** pipeline button.

    f. Open the Hierarchical Forecasting node's model Results.

    g. Click the **Output Data** tab and then select **OUTFOR** under Forecasts.

    h. Click the **Save** button.

    i. In the Save Output Data window, expand cas-shared-default.

    j. Select **Public**.

    k. Change the table name to **OUTFOR_LG_Forecasts_Node**.

    l. Select **Save**.

    Note: Forecasts exported here are statistical forecasts from this node's generated, champion models. There's not currently a way to promote this table. We'll have to load it from the HDAT version in the next step.

3. Access the saved table in Visual Analytics.

    a. Select the **Applications Menu** > **Explore and Visualize**.

    b. Select **New Report**.

c. Click the Data Sources area then select **Add Data**. Use the search field to browse to and select **OUTFOR_LG_Forecasts_Node.sashdat**.

d. Click **Add**.

e. From the report, select **Objects**.

f. Click **Time Series plot** and drag it into the viewing area.

g. Select **Assign Data**.

h. Under Time Axis, click **Add** > **Time ID Values**.

Now, what about the y-axis? The y-axis is displaying, by default, frequency of the time ID values. That's not what I want. So I'm going to switch that.

i. Left-click **Frequency** to replace it with **Predicted Values**. Click Close.

j. Select **More** (vertical ellipsis on the top right) > **Close** > **Don't Save**.

4. Export the **OUTFOR** table for the project via the Outlier tab.

a. Select the **Applications Menu** > **Build Models**. This re-opens the **Additional Topics** project.

b. Click **Close**.

c. Click the Overrides tab.

d. Select **More** > **Export All Data**.

e. Expand cas-shared-default and select **Public**.

f. Name the table **OUTFOR_LG_Forecasts_Project**.

g. Select **Export**.

Note: This table is promoted by default. Forecasts exported here are for the project. That is, they are generated from models in the overall champion modeling node in the champion pipeline, and they have been adjusted for any applied Overrides.

h. Select **Applications Menu** > **Manage Data**.

Note: **OUTFOR_LG_Forecasts_Project** shows up under the Available tab as an In-Memory table. Manage Data seems like the easiest way to confirm this, but you could also use Visual Analytics if you are more comfortable with it.

i. Return to Model Studio.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 2 Demo: Exporting Tables Generated by Request

In this demo, we will export tables generated by request from the Save Data node. This demo continues from the **Additional Topics** project created in the previous demonstration.

1. Reopen the **Additional Topics** project.

2. Request the **OUTEST** table.

   a. Click the **Pipelines** tab and open the Hierarchical Pipeline.

   b. Select the Hierarchical Forecasting node and navigate to Properties.

   c. Expand the Output Tables group and expand Attributes.

   d. Select **Parameter estimates**. Run the Hierarchical Forecasting Node.

3. Drag a Save Data node into the pipeline, configure and run it.

   a. Click the **Nodes** icon.

   b. Expand Miscellaneous.

   c. Left-click and then drag and drop the Save Data node on the Hierarchical Forecasting node. Right-click on the Hierarchal Forecasting node and select Run.

   d. Select the **Save Data** node. Click the **Edit save options** button in the properties of the Save Data node.

   e. Select **Parameter estimates** under Attributes.

   f. Check the box next to **Include in output**.

      Note: The table is Promoted by default.

   g. Rename the table **MY_OUTEST**.

   h. Click the **Browse** button to specify the output CASlib.

   i. Expand cas-shared_default and select **Public**.

   j. Select **OK** and then **OK**.

k. Right-click on the Save Data node and select **Run**.

4. Explore the exported **MY_OUTEST** table.

    a. Select the **Applications Menu** > **Manage Data**.

    b. Select **MY_OUTEST** from the Available tab.

    Note: You might need to click **Refresh** to see **MY_OUTEST**. You can also use the Filter box to search for the data.

    c. Click the Sample Data tab to view the data.

    Notice that there are multiple rows for each product line/product name combination. That's because there is a row for each parameter selected for each one of those series-- within each of those products within each of the series. So there are three parameters for Product01. There were three parameters for Product02, only two parameters for Product03, and so on.

    Note: There's one row for each parameter estimated in each champion model from the Hierarchical Forecasting node. Examine the contents of the table after expanding and grouping _**PARM**_, _**EST**_, _**PVALUE**_, and _**MODEL**_ variables.

    d. Return to Model Studio.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 2 Demo: Customizing Exported Tables

In this demo, we will customize exported tables using filters and the Save Data node. This demo continues from the **Additional Topics** project created in the previous demonstration.

1. From the opening page of Model Studio, select the **Additional Topics** project. Continue with the pipeline modified with the Save Data node in the previous demonstration.

2. Create the filter.

    a. Select the Hierarchical Pipeline and then right-click the Hierarchical Forecasting node.

    b. Select **Forecast Viewer**.

    c. Under Attributes, expand **LG_ATTRIBUTES** and expand margin_cat.

    d. Select **HIG (high margin)**.

    Note: There are 109 products with the attribute **HIG**.

    e. Click the **Save Filter** button and name the filter **High Margin Products**. Select **OK**.

    f. Close the Forecast Viewer.

3. Create a table with custom content in the Save Data node.

    a. Select the Save Data node and click **Edit save options** in the properties.

    b. Select **Statistics of fit** under Attributes.

    c. Select **High Margin Products** from the Apply an existing filter list.

    d. Check the box next to **Include in output**.

    e. If Public is not prepopulated, click the **Browse** button and expand cas-shared-default.

    f. When Public is populated, select **OK**.

    g. Rename the table **OUTSTAT_High_Margin**.

h. Select **OK** to close the window.

i. Right-click on the Save Data node and select **Run**.

4. Explore the performance of the champion models associated with high-margin products.

    a. Select the **Applications Menu** > **Explore and Visualize**.

    b. Click **New Report**. Click Data Source menu then select **Add Data**.

    c. Using the search field, select **OUTSTAT_HIGH_MARGIN** under the Available tab. Click **Add**.

       Note: You might need to refresh to see **OUTSTAT_HIGH_MARGIN**. There are 109 product name levels listed next to this variable under Category.

    d. Left-click and drag the **Mean Absolute Percent Error** variable into the main window (under Measure). The generated histogram shows MAPE across the 109 products in the high-margin category.

    e. Close the report. You do not need to save the report. Return to Model Studio.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 3 Demo: Updating Models and Forecasting

In this demo, we build a Hierarchical Forecast Project based on monthly data through December, 2016. We then acquire three months of additional data and update the forecasts in three different ways. First, we use the selected models and the parameter estimates from December, 2016. Next, we use the selected models from December, 2016, with parameter estimates updated using the additional three months of data. Finally, we use newly selected forecasting models and parameters from all data including the additional three months.

1. Starting from the SAS Viya landing page, let's create the **Updating Models** project. This project will use the same in-memory tables and variable metadata as the **baseline sales forecasts** project created earlier. For convenience, a summary of the project creation steps is below:

    a. From the Applications Menu, select **Build Models**.

    b. Click the **New Project** button.

    c. Name the project **Updating Models**, set the type to **Forecasting**, and provide a reasonably detailed description.

    d. Select **Hierarchical Forecasting** as the template.

    Note: If Hierarchical Forecasting is not listed, you will need to browse to select it.

    e. Navigate to the **LOOKINGGLASS_FORECAST** table on the Available tab and click **OK**. Then click **Save**.

    Note: If **LOOKINGGLASS_FORECAST** does not appear on the Available tab, then you need to load it into memory following steps that were shown previously in the course.

    f. Assign variable roles. **Txn_Month** is the time variable, the dependent variable is **sale**, and the BY variables are **productline** and **productname**. Set Reconciliation level to **productline**.

    g. Select and edit **price**, **discount**, and **cost**. Set the role of these variables to **Independent** and change Usage in system-generated models to **Use if significant**.

2. Navigate to the Pipelines tab.

3. Click the Hierarchical Forecasting node and note the settings in the right panel.

4. Verify that the Forecast task is set to **Diagnose**.

   When you select a forecasting task Diagnose, there are several things that will happen. There will be three candidate models for each series. Among those candidate models, based on the criterion that you select, one of those models will be selected for each one of the series. The parameter estimates will be estimated from the data and then forecasts will be generated.

   We're going to start by assuming that we have not yet reached April of 2017. We don't have the data through March of 2017. So let's just generate the model from the data that we had from December of 2016 and before.

5. Expand Model Selection and type **12** for the number of data points used in the holdout sample.

6. Type the number **25** for the Percentage of data points used in the holdout sample.

7. Click the button for **Run pipeline**.

8. Click the Overrides tab and note the statistical forecasts for the first few months of 2017, focusing on April, 2017.

   Note: There is no historical data for these months. By default, the forecast horizon of 12 months ends at December, 2017.

   In the next part of the demo, a new data file is read in that contains all the data from LOOKINGGLASS_FORECAST, plus data for three additional months, through March, 2017. This simulates the experience of continuing to use a model for several months, even as new data are collected. In this example, we will use both the selected models for each series, and also the parameter estimates obtained from the LOOKINGGLASS_FORECAST data.

9. Return to the Data tab.

10. Click the **New data source menu** button and select **Time series** from the menu.

11. Load the **lg_fct_ext3mon** data into memory.

    a. Click the Import Data button.

    b. Select **Local files**. Click on **Other Locations**. Click on Computer. Scroll down and double-click on the Workshop folder. Open the FVVF folder.

    c. Select **lg_fct_ext3mon.sas7bdat**.

d. Click **Open**.

e. Select **Replace File** under If target table name exists.

f. Click the button for **Import Item**.

g. Click **ADD**.

h. Ensure that the Show area is set to Available. Select **lg_fct_ext3mon.sas7bdat**. Click **ADD**. If a pop-up window appears, close it.

12. Return to the Pipelines tab.

13. Click the Hierarchical Forecasting node and go to its settings in the right panel.

14. Set Forecast Task to **Forecast**.

    By doing this, we're going to be taking the additional three months of data, including it in our forecast for April, May, June, and so on. So we should see the forecast change for April, May, June, and beyond. Not only that, the forecast horizon now will be until March of 2018 rather than December of 2017.

15. Run the pipeline.

16. Click the Overrides tab.

17. Click **Yes** to refresh overrides using the new data.

    Note: There are differences between the plot and tables displayed now and the previous plot and tables. There is now historical data for January 2017 through March 2017. The April data statistical forecasts (and beyond) have changed, due to the addition of three time points of historical data. The forecast horizon now extends to March 2018.

    I said I wanted to pay attention to the April forecast. All I've done so far when I clicked on Forecast is to update the forecast based on the new data. So with new data, the forecasted values should change a little bit. But what hasn't changed is the model itself. So the selected models for each of the 918 series are exactly the same. There has not been a new selection of any of the models. None of the candidate models for each of the series have changed. And the parameters have not changed either. The parameters are not estimated. So it's not a big surprise that there's not a big change in the forecasted values. The one thing that you will note that's different is that now, the forecast horizon starts in April rather than January. And it extends to March of 2018.

So what if we do want the flexibility? With new data, parameter estimates could be readjusted.

In the next part of the demonstration, we will rethink our forecasting approach. We do not want to re-select champion models, but we will allow the parameter estimates to be adjusted to account for any changes in the last three months.

18. Return to the Pipelines tab.

19. Click the Hierarchical Forecasting node and go to its settings in the right panel.

20. Set Forecast Task to **Fit**.

    So with Fit, and as we refit the models, in other words, we re-estimate the parameters for the model. So that should change the forecasts a bit more than when we used Forecast as the forecast task.

21. Run the pipeline.

22. Click the Overrides tab.

23. Click **Yes** to refresh overrides using the new data.

    Note: There are differences between the plot and tables displayed now and the previous plot and tables. The April data statistical forecasts (and beyond) have changed, due to the addition of three time points of historical data and re-estimation of the model parameters.

    So now there's a bigger change in the forecast for April. Once again, the difference here is the parameters were allowed to be re-estimated based on the three more months of data. Whereas, using Forecast for the forecast task, those parameter estimates were not allowed to be changed. And if you pay attention, the shape of the forecasts here is much different now that I've changed the parameter estimates for the table. So that makes a big difference. And that's a decision that you need to make as far as, after a certain amount of time, how do you want to update your forecasts? It's probably not a good idea each time you get new data to update the entire model, so re-estimate, try to find new candidate models, and so on. But you might want to do things like at least re-estimate the parameters based on the new data.

    Now what if you did want to perhaps reselect? Or remember, one of the things that's done when you first run a model at the very beginning is that there are three candidate models that are generated for each one of this series. And then, based on some statistic, maybe MAPE or MAE, you select from among those three candidate models for each one of the series. Now, with new data, it might be that maybe the MAPE for one of the other candidate models now does better than the previous champion model for a particular series. So in order to allow me to change the entire model or at least within the three candidates that were already produced, so I still am limited to the three candidates that were already available to me based on the original LG data. I can select a new one among those three. So I'm going to be using the Forecast task of select this time and see how that changes things.

In the next part of the demo, based on recent information, we decide that perhaps new models would work better. The newly selected models, however, are limited to the three candidates that came from the Diagnose process for the **LOOKINGGLASS_FORECAST** data.

24. Return to the Pipelines tab.

25. Click the Hierarchical Forecasting node and go to its settings in the right panel.

26. Set Forecast Task to **Select**.

27. Run the pipeline.

28. Click the Overrides tab.

29. Click **Yes** to refresh overrides using the new data.

    Note: There are differences between the plot and tables displayed now and the previous plot and tables. All statistical forecasts have changed, due to the re-selection of the forecast models.

    Let's see how much of a difference this makes in the April forecast. Well, not that much of a difference. But it does make some difference. Remember that in many cases of those 918 series, the same one of the three candidates will be selected. But you can see the shape of the forecast is a little bit different.

    In the final part of the demo, we consider the old models stale and restart model selection and estimation using all available data.

30. Return to the Pipelines tab.

31. Click the Hierarchical Forecasting node and go to its settings in the right panel.

32. Set Forecast Task to **Diagnose**.

33. Run the pipeline.

34. Click the Overrides tab.

35. Click **Yes** to refresh overrides using the new data.

Note: There are differences between the plot and tables displayed now and the previous plot and tables. All statistical forecasts have changed, due to the re-selection of the forecast models.

Let's see how much we change in April. The shape of the forecast is definitely changing. You can see in the plot. It's up to you to determine when and how much to change parameters, the selected models, or refresh the entire forecast.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 4 Demo: Using Keyword Event Variables in an Automatic Forecasting System

In this demo, we will add predefined event variables to the project. This demo continues from the **Additional Topics** project created previously.

1. From the opening page of Model Studio, select the **Additional Topics** project.

2. Add event variables and include them as candidates in the model generation process.

    a. Click the Data tab and expand the Events section..

    b. Click on **Predefined Events**.

    c. Select **Christmas Day**, **Independence Day (US)**, and **Thanksgiving Day** by checking the boxes next to these events.

    d. Set Usage in system-generated models to **Try to use**.

    With Try to Use and Use if Significant, in both cases, the event or the independent variable will be selected if it is statistically significant in the model. If you use Try to Use, another criterion will be a predefined improvement in a model fit statistic, like Akaike's information criterion.

    e. Select the Pipelines tab.

    f. Add a Hierarchical Forecasting pipeline by clicking on **Add a Pipeline** (+).

    g. Change Template to **Hierarchical Forecasting**, and name the new pipeline **Hierarchical with event variables**.

    h. Click **Save**.

    i. Run the pipeline.

    j. Open the Hierarchical Modeling node by right-clicking it and selecting **Results**.

    Note: Mousing over the histogram we can see the percentage of the forecast models at the base and middle level of the hierarchy contain at least one event variable.

    k. Close the Results.

3. Assess the impact of event variables on goodness of fit.

    a. Click on **Nodes**.

    b. Expand Forecasting Modeling.

    c. Drag a Hierarchical Forecasting node into the pipeline onto the Data node.

    d. Right-click on the new node and select **Rename**.

    e. Rename the new node **Hierarchical no predefined events** and click **OK**.

    f. Right-click the new node and select **Modify event usage**.

    g. Select all event variables.

    h. Set Usage in system-generated models to **Do not use**.

       In this way, I can compare my champion model that allowed for the events to the champion models that will come from not allowing the events.

    i. Close Modify event usage and rerun the pipeline.

    j. Right-click on the Model Comparison node and select **Results**.

       Notice the difference in various diagnostics based on inclusion of event variables. The champion was Hierarchical Forecasting, so having the events improved those models a little bit. If you look at the WMAPE, the weighted MAPE, the difference is not very great at all, but there was some difference.

    k. Close the Model Comparison results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 4 Demo: Adding Custom Event Variables in an Automatic Forecasting System

In this demo, we will add custom event variables to the project. This demo continues from the **Additional Topics** project created previously.

1. From the opening page of Model Studio, select the **Additional Topics** project.

2. Import the custom event variables table into the project.

   a. Select the Data tab.

   b. Select **New Data Source** > **Events**.

   c. Click the **Import** data button > **Local files**.

   d. Browse to the FVVF directory. (Click on **Other Locations**. Click computer. Scroll down and double-click on the **Workshop** folder. Find and double-click the **FVVF** folder.) Select **lg_eventdat.sas7bdat** and then **Open**.

   e. Click the **Import Item** button and then **ADD**. Ensure that the Show area is set to Available. Choose **lg_eventdat.sas7bdat**. Click **ADD**.

   f. Select both **SUMMER** and **TC_XMAS**.

   g. Set Usage in system-generated models to **Try to Use**.

   h. Click the **View Table** button.

      Note: The Keyword and Duration of Event (before and after) columns indicate that Summer is a three-month pulse that starts in June of each year, and TC_Xmas is a temporary change type event variable that starts in November and has short persistence.

3. Update system generated models to include the Custom event variables as candidates.

   a. Click the Pipelines tab.

   b. Rerun the Hierarchical with event variables pipeline from the previous demonstration.

      Now that the new events have been defined and they have been defined as Try to Use, Try to Use will be valid for both Hierarchical forecasting node and

Hierarchical node predefined events. So the Hierarchical no predefined events will allow no predefined events but will allow for the events that I just added.

c. Right-click the **Hierarchical no predefined events** node and select **Forecast viewer**.

d. Click the Modeling tab.

e. Select the **Line01:Product02** series.

Note: The champion forecast model contains the **TC_XMAS** event variable. So this shows you that at least for this series, that event variable was an improvement on the model.

f. Close the Forecast Viewer.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 5 Demo: Exploring IDM Models in Model Studio

In this demo, we will explore Intermittent Demand Models (IDM) in Model Studio. This demo continues from the **Additional Topics** project created previously.

1. From the opening page of Model Studio, select the **Additional Topics** project.
2. Explore the Projects results for intermittent series.
   a. Click the Pipelines tab.
   b. Expand Forecasting Modeling.
   c. Select **Hierarchical Forecasting (Pluggable)**. Drag and drop it onto the Data node.
   d. Select the **Hierarchical Forecasting (Pluggable)** node, expand Model Generation, and ensure that the **Include IDM models** box is checked.
   e. Run the pipeline.
   f. Right-click on the Hierarchical Forecasting (Pluggable) node and select **Results**.

      Note: About one percent of the series at the **PRODUCTNAME** level have been detected as Intermittent.
   g. Close the Results.
   h. Select the **Hierarchical Forecasting (Pluggable)** node, expand Model Generation, and expand Intermittency test.

      Note: The Intermittency test is on by default and the Sensitivity level for intermittency test is set at 2.
   i. Right-click on the Model Comparison node and select **Results**.
   j. Close the Results.
3. Assess other IDM Sensitivity settings.
   a. Expand Forecasting Modeling in Nodes.
   b. Click on **Hierarchical Modeling (Pluggable)** and drag it onto the Data node.

c. Right-click on the new node and select **Rename**.

d. Rename the new node **Hierarchical Forecasting (Pluggable) (Modified IDM)** and click **OK**.

e. Select the **Hierarchical Forecasting (Pluggable) (Modified IDM)** node, expand Model Generation, and expand Intermittency test.

f. Change the Sensitivity level for intermittency test to **6**.

g. Rerun the pipeline.

h. Right-click on the the Hierarchical Forecasting (Pluggable) (Modified IDM) node and select **Results**.

   Note: No series are classified as IDM based on results in the Model Family plots.

i. Close the Results.

j. Right-click the Model Comparison node and select **Results**.

   Note: The Hierarchical Forecasting (Pluggable) (Modified IDM) node is the champion. This can be interpreted as evidence that for this particular data set, the default sensitivity settings are sub-optimal.

k. Close the Results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 6 Demo: Adding Outlier Detection to the Forecasting System

In this demo, we will add Outlier Detection to the forecasting project. This functionality is accessed from the Hierarchical Forecasting (Pluggable) node and is activated by editing the in-line code. This demo continues from the **Additional Topics** project created previously.

1. From the opening page of Model Studio, select the **Additional Topics** project.

2. Bring a Hierarchical Forecasting (Pluggable) node into a pipeline and edit the In-Line code.

    a. Open the **Additional Topics** project.

    b. Select the Pipelines tab.

    c. Select **Add a New Pipeline**.

    d. Select the **Hierarchical Forecasting Template**.

    e. Name the pipeline **Outliers**.

    f. Select **Save**.

    g. Select **Nodes**.

    h. Expand the Forecasting Modeling section.

    i. Click on **Hierarchical Forecasting (Pluggable)** and then drag and drop a Hierarchical Forecasting (Pluggable) node onto the Data node.

    j. Click **Open Code Editor** in the Properties of the Hierarchical Forecasting (Pluggable) node.

    k. Scroll down to the "Define the diagnose part of script to run in TSMODEL" comment in the code.

    Note: This comment is at line 95 approximately.

    l. Change the arguments of the setARIMAXOutlier method to `rc = diagSpec.setARIMAXOutlier('DETECT', 'YES');`

    This will perform outlier detection, and slightly modify the models when I do that.

m. Click **Save** and then close the In-Line code window.

3. Run the node and explore the Outlier Detection results.

   a. Run the Pipeline.

   b. Right-click the Hierarchical Forecasting (Pluggable) node and select **Results**.

   c. Navigate to the Model Type results.

      Note: About 11% of the series at the base level of the hierarchy have a champion model that contains at least one Outlier variable.

      Now not only are the outliers detected, but also when outliers are detected, there are some modifications to the models in order to improve the model or optimize the model for correction of the outliers.

   d. Close the Results.

   e. Right-click the Model Comparison node and select **Results**.

      Note: The Hierarchical Forecasting (Pluggable) node is now the champion node for the pipeline. WMAPE for the new, champion node including Outlier Detection improves relative to the Hierarchical Modeling node that does not include this functionality.

   f. Close the Results.

Forecasting Using Model Studio in SAS® Viya®
Lesson 05, Section 7 Demo: Using the Ensemble Node

In this demo, we will add an Ensemble node to a pipeline in the **Additional Topics** forecasting project.

Create a New a Pipeline

1. From the opening page of Model Studio, select the **Additional Topics** project.
2. Open the **Additional Topics** project.
3. Select the Pipelines tab.
4. Select **Add a New Pipeline**.
5. Name the Pipeline **Ensemble node**.
6. Select **Browse**. Select the **Auto-Forecasting Template** and then select **OK**.
7. Select **Save**.

Add Modeling Nodes for the Ensemble

1. Right click the Auto-forecasting node and choose rename. Give it the name **Auto-forecasting default** and then select **OK**.
2. On the left pane, select the **Nodes** menu and expand **Forecasting Modeling**.
3. Left click, drag and then drop a **Hierarchical Forecasting node** on top of the Data node.
4. Right click and rename the Hierarchical Forecasting node **ARIMA with Outliers**.
5. Select **OK**.
6. Make sure that ARIMA with OUTLIERS node is selected. On the right pane locate and expand the **Model Generation** properties.
7. Expand **Include ARIMAX models** and then change the criterion for outlier detection to **Yes**.
8. Toggle the option off to **Include ESM models**.
9. Return to the left side pane and select the **Nodes** menu.

10. Left click, drag and then drop an **Auto-forecasting node** on top of the Data node.

11. Right click and rename this Auto-forecasting node **UCM only** and then select **OK**.

12. With the **UCM only** node selected, expand the **Model Generation** properties on the right pane.

13. De-select all checked boxes next to the listed models.

14. Check the box next to **Include UCM models**.

Add and Run an Ensemble Node

1. From the left pane, select the **Nodes** menu.

2. Expand **Postprocessing**.

3. Left-click, drag and drop an **Ensemble node** on top of the Auto-forecasting default node. This will connect this node to the Ensemble node. However, we want to add more.

4. Right-click the Ensemble node and select **Add Models**.

5. Select **ARIMA with Outliers**.

6. Right-click the Ensemble node and select **Add Models**.

7. Select **UCM Only**.

8. **Run** the Pipeline.

   Note that **RMSE** is the listed Metric in the properties of the Ensemble node. By default, the champion forecast from the predecessor modeling nodes with the lowest RMSE is chosen as the Ensemble champion for each series. RMSE for each candidate forecast is calculated over the entire history for each series. Several other fit measures or Metrics are available.

   Note: If periods are excluded from modeling in the Project Settings, the Ensemble node calculates the selection statistic on the excluded observations or over the out of sample range.

Assess the Ensemble Node Pipeline Results

1. Right click the Ensemble node and select **Results**.

2. The Modeling Node Weights table shows the frequency of forecasts contributed by predecessor modeling nodes. Mousing over **ARIMA with Outliers** shows what percentage of it is of the 918 of the Ensemble's champion forecasts. Mousing over **UCM only** shows its percentage of the Ensemble's champion forecasts.

3. Select the **Output Data** tab.

   The **OUTFOR** table lists the Predicted Values, Confidence Limits and so on for the forecasts selected by the Ensemble node.

4. Left click the **OUTWEIGHT** table to open it in the viewer.

   This table provides a mapping between the predecessor modeling node that supplies the forecast and the corresponding BY group.

   Note: The OUTEST table is not available out of the box for export from the Ensemble node. However, mapping information from the OUTWEIGHT table could be combined with the OUTEST tables from each predecessor modeling node to compile the Ensemble node's OUTEST information.

5. **Close** the Ensemble node results.

6. Right click the Model **Comparison node** and select **Results**.

   The Ensemble node is the Champion modeling node for this pipeline.

Lesson 06

Forecasting Using Model Studio in SAS® Viya®
Lesson 06, Section 0 (Self-Study) Demo: Auto-forecasting Code Overview

A straightforward version of the code can be accessed from the Auto-forecasting node. One of the things that make this version of the generated code straightforward is that it operates on only the base level of the data hierarchy. That is, there is no time series creation in the middle and upper levels of the hierarchy (aggregation) or forecast reconciliation accommodated in the generated code. We consider a more general version of the code in a subsequent demonstration.

1. In the baseline sales forecast project, click the **Pipelines** tab and navigate to **Pipeline 1**. Pipeline 1 contains the default Auto-forecasting template. Run the pipeline.

2. Select the **Auto-forecasting** node and find the code editor option to the right. Select Open.

Part 1: Set Up and Data Accumulation

1. The **PROC TSMODEL** statement specifies the in-memory table to be used for analysis.

   The PROC statement references macro variables that resolve to a caslib and the in-memory table that contains the transactional modeling data, **LOOKINGGLASS_FORECAST**. Output objects are also listed. These are CAS table names that will contain the results of the automatic modeling and forecasting process.

```
proc tsmodel data = &vf_libIn.."&vf_inData"n
     outobj = (
              outfor  = &vf_libOut.."&vf_outFor"n
              outstat = &vf_libOut.."&vf_outStat"n
              outSelect = &vf_libOut.. "&vf_outSelect"n
              outmodelinfo = &vf_libOut..outmodelinfo
               )
                 ;
```

2. The next steps define the time series that result from the process of accumulating the transactional data in the **LOOKINGGLASS_FORECAST** table.

   &vf_timeID resolves to **TXN_MONTH**, and the interval is monthly. The **VF_varsTSMODEL** macro lists the dependent and candidate independent variables with their corresponding accumulation methods. Recall the BY variables that are defined for the project. **&vf_byVars** resolves to **productline** and **productname**.

```
*define time series ID variable and the time interval;
 id &vf_timeID interval = &vf_timeIDInterval
               Setmissing = &vf_setMissing trimd = LEFT;

 *define time series and the corresponding accumulation methods;
 %vf_varsTSMODEL;

 *define the BY variables if exist;
 %if "&vf_byVars" ne "" %then %do;
    by &vf_byVars;
 %end;
```

3. The **REQUIRE** statement specifies that the ATSM package be used. The **SUBMIT** statement flags the beginning of DATA-step-like or external functionality in the TSMODEL procedure.

```
*using the ATSM (Automatic Time Series Model) package;
    require atsm;

    *starting user script;
    submit;
```

4. Recall that packages contain objects. Objects need to be declared and then initialized. After that, methods can be run on the objects. Below, ATSM objects, needed for subsequent forecasting steps, are declared.

   TSDF indicates a time series data frame object type. The first **DECLARE** statement below declares a **tsdf** object type and names it **dataframe**. A description of other objects declared is provided in the syntax comments.

```
*declaring the ATSM objects;
        /*
        TSDF:     Time series data frame used to group series
              variables for DIAGNOSE and FORENG objects
        DIAGNOSE: Automatic time series model generation
        FORENG:   Automatic time series model selection and
              forecasting
        DIAGSPEC: Diagnostic control options for DIAGNOSE object
        OUTFOR:   Collector for FORENG forecasts
        OUTSTAT:  Collector for FORENG forecast performance
              statistics
        */
```

```
        declare object dataframe(tsdf);
        declare object diagnose(diagnose);
        declare object diagspec(diagspec);
        declare object inselect(selspec);
        declare object forecast(foreng);
```

5. In the next step, the **dataframe** object is initialized, and **addY** and **addX** (via the **addXTSMODEL** macro) methods are run on it to populate it with the dependent and independent variables in the project.

```
*initialize the tsdf object and assign the time series
         roles;
        rc = dataframe.initialize();
        rc = dataframe.addY(&vf_depVar);
        *add independent variables to the tsdf object if there
              are any;
        %if "&vf_indepVars" ne "" %then %do;
            %vf_addXTSMODEL(dataframe);
        %end;
```

Part 2: Diagnose (Create) Model Specifications

1. The **diagspec** object regulates the model identification step in the project. The default methods applied to the **diagspec** object create an exponential smoothing and an ARIMAX model specification each series in the data.

```
rc = diagSpec.open();
        %if %UPCASE("&_esmInclude") eq "TRUE"  %then %do;
            rc = diagSpec.setESM('METHOD', 'BEST');
        %end;
        %if %UPCASE("&_arimaxInclude") eq "TRUE"  %then %do;
            rc = diagSpec.setARIMAX('IDENTIFY', 'BOTH');
        %end;
        %if %UPCASE("&_idmInclude") eq "TRUE" %then %do;
            rc = diagSpec.setIDM('INTERMITTENT',
                                 &_intermittencySensitivity);
            rc = diagSpec.setIDM('METHOD', "&_idmMethod");
        %end;
        %else %do;
            rc = diagSpec.setIDM('INTERMITTENT', 10000);
        %end;
        %if %UPCASE("&_ucmInclude") eq "TRUE" %then %do;
            rc = diagSpec.setUCM();
        %end;
```

```
        rc = diagSpec.setOption('CRITERION',
            "&_modelSelection_criteria");
        rc = diagSpec.close();
```

Note: **rc** represents a return status code. Return codes are numeric values that are returned when a method associated with an object is called.

2. The diagnose object, **diagnose**, is initialized. Model identification restrictions contained in the **diagspec** object are read into **diagnose**. The **run** method is then set on the **diagnose** object to kick off the creation of model specifications for the project.

```
*set the diagnose object using the diagspec object and run the
            diagnose process;
        rc = diagnose.initialize(dataframe);
        rc = diagnose.setSpec(diagspec);
        …
        rc = diagnose.run();
```

Part 3: Automatic Model Selection and Forecast Generation

1. Forecast objects are used to do automatic model selection and forecasting. Here, the best model for each time series in the data is selected based on model selection lists associated with the **diagnose** object.

   The **forecast** object is initialized, and the results of the diagnose step are loaded. Next, forecasting, model selection, and other options are set using **setOption** methods. These options include defining the lead forecast horizon and the model selection criterion. The **run** method is set on the forecast object to kick off the automatic model selection and forecasting process.

```
*initialize the forecast object with the diagnose result and run
         model selecting and generate forecasts;
        rc = forecast.initialize(dataFrame);
        rc = forecast.setOption('criterion', …
          …&_modelSelection_criteria");
        rc = forecast.setOption('lead',&vf_lead);
        rc = forecast.setOption('horizon',&vf_horizonStart);
        rc = forecast.setOption('minobs.trend',&_minobsTrend);
        rc = forecast.setOption('minobs.mean',&_minobs);
        %if "&vf_allowNegativeForecasts" eq "FALSE" %then %do;
            rc = forecast.setOption('fcst.bd.lower',0);
        %end;
   …
        rc = forecast.run();
```

2. In the final step before the ENDSUBMIT statement, forecast results are collected into tables defined at the start of the syntax.

```
*collect the forecast and statistic-of-fit from the forgen
    object run results;
        rc = outfor.collect(forecast);
        rc = outstat.collect(forecast);
        rc = outSelect.collect(forecast);
        rc = outmodelinfo.collect(forecast);
    endsubmit;
quit;
```

Forecasting Using Model Studio in SAS® Viya®
Lesson 06, Section 0 (Self-Study) Demo: Modifying the Auto-forecasting Code and Creating a Custom Forecast Node

Suppose that model interpretability is important to you and your forecast project's stakeholders. One way to potentially improve model interpretability is to prioritize the identification of the regression part of diagnosed ARIMAX models. Let's show how to modify the code provided by the auto-forecasting node to do this. Subsequent steps show that the Exchange in Model Studio can be used as a repository to share customized nodes.

1. Return to Pipeline 1 of the **LG Hierarchy Forecast** project and access the Auto-forecasting node code via the options.

2. Open the code editor.

3. The **diagSpec** objects are set starting on line 152. Modify the **setARIMAX** action as below.

```
/*open the diagspec object and enable ESM, IDM, UCM, ARIMAX
         model class for diagnose; */
   rc = diagSpec.open();
   %if %UPCASE("&_esmInclude") eq "TRUE"  %then %do;
        rc = diagSpec.setESM('METHOD', 'BEST');
   %end;
   %if %UPCASE("&_arimaxInclude") eq "TRUE"  %then %do;
        rc = diagSpec.setARIMAX('IDENTIFY', 'REG');
   %end;
...
```

Note: Further information about ATSM objects, actions, and associated options can be found in the TSMODEL procedure documentation (*SAS® Visual Forecasting 8.4: Forecasting Procedures*. 2019).

4. Select **Save** and then close the Code window.

5. Run the pipeline and then open the results of the Auto-forecasting node. The Model Type chart indicates that the inputs are selected into almost 70% of the generated ARIMAX models for this node.

6. Close the results.

7. Right-click the **Auto-forecasting** node and select **Save as**.

8. Provide the name **Auto-forecasting with REG-ARMA** and a reasonably detailed description, such as "This node prioritizes the identification of the regression (X) part of generated ARIMAX models."

9. Click **Save**.

10. Expand **Nodes** > **Forecasting Modeling**. Note that the new node is now available in other pipelines within this project.

11. Select **View all projects** to navigate to the Model Studio main page.

12. Select **The Exchange** on the menu. The custom node is also available to other authorized users and across projects.

Forecasting Using Model Studio in SAS® Viya®
Lesson 06, Section 0 (Self-Study) Demo: Overview and Modification of the Hierarchical Forecasting (Pluggable) Node

Another node that provides code access is Hierarchical forecasting (pluggable). The code that this node contains provides more functionality than the previous code that we examined. It accommodates a hierarchical approach to time series generation and forecasting and also includes reconciliation functionality. Because the code is more general, it is arranged in a series of handy macros that regulate the different steps in a forecasting project.

1. Open **Pipeline 2** in the **LG hierarchy forecast** project.

2. Expand the nodes menu, and then select, drag, and drop a **Hierarchical forecasting (pluggable)** node on top of the **Data** node.

3. Select the **Hierarchical forecasting (pluggable)** node, and open the code editor from the options menu. The header comments provide useful information, including macro variable names for popular properties and output table names. It also lists three macros in the code.

4. The first macro, **tsmodelCodeDiagnose**, defines the diagnose part of the forecasting analysis. This macro's syntax is similar to diagnose code that we have seen before. A **diagspec** object is opened, and actions are run on it that regulate the model creation or diagnosis portion of the analysis. The **diagspec** object is then closed and loaded into a **diagnose** object via a **setspec** option. We return to this macro later in the demonstration to add a model family that will be considered for each series in the hierarchy.

```
/*-----------------------------------------------------------------
 * Define the diagnose part of script to run in TSMDOEL
 *---------------------------------------------------------------*/
%macro tsmodelCodeDiagnose;

    /*setup time series diagnose specifications*/
    rc = diagSpec.open();
    %if "&_esmInclude" eq "TRUE" %then %do;
        rc = diagSpec.setESM('METHOD', 'BEST');
    %end;
    %if "&_arimaxInclude" eq "TRUE" %then %do;
        rc = diagSpec.setARIMAX('IDENTIFY', 'BOTH');
    %end;
    %if "&_idmInclude" eq "TRUE" %then %do;
        rc = diagSpec.setIDM('INTERMITTENT',
```

```
                                &_intermittencySensitivity);
    %end;
    %else %do;
        rc = diagSpec.setIDM('INTERMITTENT', 10000);
    %end;
    %if "&_ucmInclude" eq "TRUE" %then %do;
        rc = diagSpec.setUCM();
    %end;
    rc = diagSpec.close();

    /*diagnose time series to generate candidate model list*/
    rc = diagnose.initialize(dataFrame);
    rc = diagnose.setSpec(diagSpec);
    rc = diagnose.setOption('BACK', &_forecastBack);
    rc = diagnose.setOption('minobs.trend',&_minobsTrend);
    rc = diagnose.setOption('minobs.season',&_minobsSeason);
    rc = diagnose.Run();
    ndiag = diagnose.nmodels();

    /*setup combined model*/
    %if "&_combInclude" eq "TRUE" %then %do;
        declare object comb(combSpec);
        rc = comb.open(ndiag);
        rc = comb.AddFrom(diagnose);
        rc = comb.close();
    %end;

    /*Run model selection and forecast*/
    rc = inselect.Open(ndiag);
    rc = inselect.AddFrom(diagnose);
    rc = inselect.close();

%mend;
```

5. The second macro, **tsmodelCodeSelectOption**, defines how holdout sample selection works, if the holdout sample options are changed from 0.

6. The **tsmodelCode** option defines a script that contains the main components necessary for running PROC TSMODEL with the ATSM package: declaring and initializing objects, loading the target and explanatory variable names into the **tsdf** object, and so on.

7. Different levels of the data hierarchy are operated on independently with regard to automatic model diagnosis and model selection. Series in each level of the hierarchy get their statistical forecasts generated with separate calls of the **hf_tsmodel** macro.

8. After the statistical forecasts are generated, forecast reconciliation across hierarchy levels can be accomplished with one of the two reconciliation macros, **hf_reconcile_td** or **hf_reconcile_bu**, given a selected reconciliation level.

9. The **vf_hier_forecast** macro embeds the relevant macros described above, and it is the main macro function call for generating statistical and reconciled forecasts for series in the data hierarchy.

10. Select **Close** to return to Pipeline 2.