Paper 2575-2018

Extending the Capability of SAS Forecast Studio® to Enable Custom Exception Handling

Divya Guru Rajan, CoreCompete; Christopher Houck, CoreCompete

ABSTRACT

SAS Forecast Studio allows for the generation of a large number of forecasts in a fast and automated way. However, examining forecast exceptions can become very time-consuming. This paper demonstrates how to automate the process of selecting alternate models and adjust forecasts for different types of exceptions. See how our method uses model related datasets and catalogs generated as part of a Forecast Studio project based on data from a large fashion retailer. In a programmatic way, alternate models are selected and the related datasets are updated. The resulting alternate model selections can be viewed in Forecast Studio. This paper describes the exception types addressed, the overall process flow for selecting forecast exceptions, and the method used to select alternate models. In addition, out-of-sample accuracy results demonstrate how applying this approach improves forecast results.

INTRODUCTION

As retail companies use time-series to generate a large number of forecasts in a short amount of time using powerful software such as SAS Forecast Studio, identifying and addressing exceptions through manual intervention can be cumbersome. This paper shows an automated way to identify and address forecast exceptions while continuing to use the visual power of the SAS Forecast Studio user interface.

PROBLEM

Shipments data from the retail industry often consists of irregular spikes and dips, short history, of insufficient frequency as well as overall noise, making it difficult to use automated time-series models. Although using functionality built into SAS Forecast Studio that allows the use of holdout sample for model selection usually improves performance, in cases where the series are not stable, the generated models and related forecasts can be based on anomalies seen in small holdout period.

Another issue is that the models selected using an automated forecasting process can be unusually sensitive to outlier data points found towards the end of the historical period. The model chosen then can produce forecasts with a significantly high or low trend based on a small number of sparse and anomalous data points which may not conform with the reality of the business environment and hence will need to be adjusted.

Although it is possible within SAS Forecast Studio for users to select alternate models to address exception forecasts by manually clicking through different series and models, doing so across a large number of series may not be feasible. However, while it is possible to select an alternate model using code written outside of the forecasting system, doing so does not allow a user to see the change reflected in the Forecast Studio user interface.

To sum up, there are two specific issues in accounting for forecast exceptions – a. Identifying and addressing exceptions in an automated way and, b. Enabling viewing of the final model selected in SAS Forecast Studio's user interface.

This paper addresses both of the above issues through a method using the model datasets and catalogs generated as part of a Forecast Studio project to select alternate models in an automated way and also allow users to see the alternate selections in the user interface. We first describe the types of exceptions we seek to address, the overall process flow for selecting forecast exceptions along with the method used to select alternate models and finally provide the out-of-sample accuracy results.

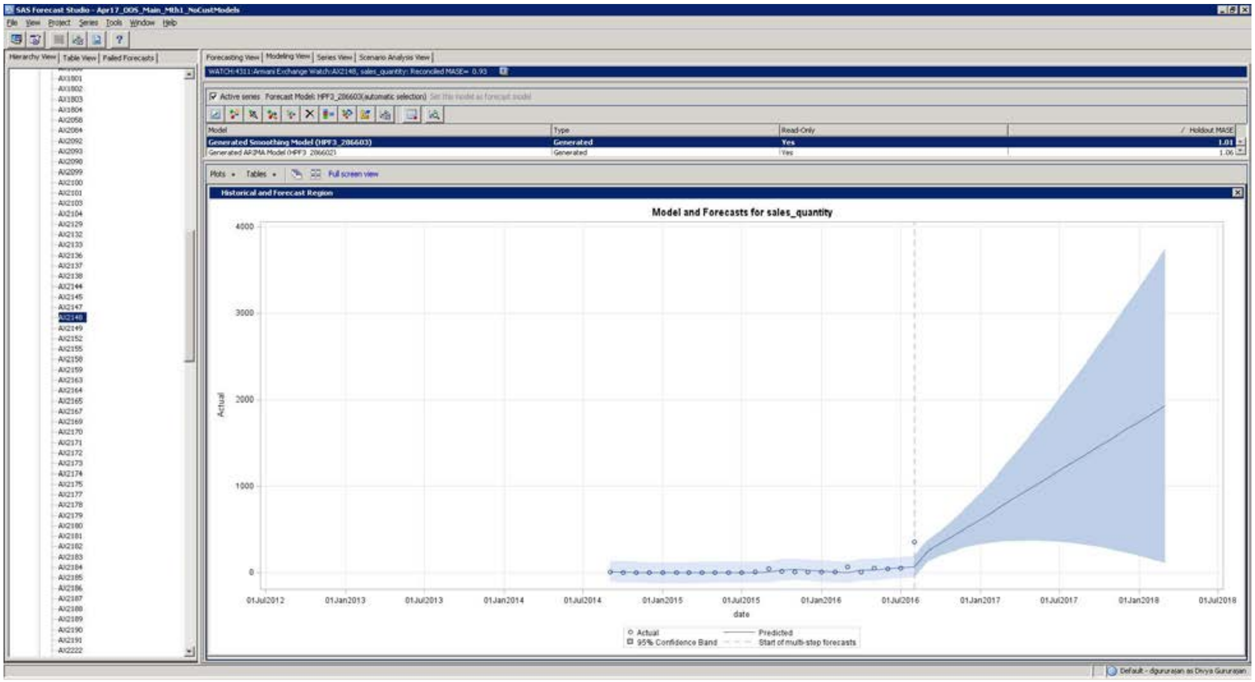
SOLUTION

The data is monthly shipments spanning a historical period, ranging from 1-3 years for three different fashion retail products. Accuracy based on business requirements is related to the forecast for the fourth month outward or the 90-day offset (starting from the month of forecast generation). Here, we check accuracy for three such consecutive out-of-sample periods.

TYPES OF EXCEPTIONS

In selecting exceptions to examine, three different criteria are checked:

1. "Runaway" Forecasts: The first type of exception (shown in Fig. 1) is when the forecasts – with a model that was automatically selected, likely based on a small number of data points towards the end of the historical period – are either much higher or lower than the actuals in the historical period. The approximate threshold we use for this exception is when the minimum of the forecasts is greater than the maximum of the actuals and vice-versa. The forecast generated using an alternate model selected through our exceptions handling process is shown in Fig. 2.
2. Forecast Average Significantly Different from Actuals Average: The second type of exception (shown in Fig. 3) is when the average of the forecasts is significantly (for example, 50%) greater or lesser than the average of the actuals from the historical period. We find that this type of exception usually occurs when the model automatically selected by Forecast Studio is based on a holdout period that is a small, non-representative sample of the overall series. The forecast generated using to address this exception using an alternate model as part of our process is shown in Fig. 4.
3. Complex Models Based on Input Variables Containing External Forecasts: The third type of exception (shown in Fig. 5) that we address, occurs when externally generated forecasts are included among the set of forecasts for the software to pick from. In this situation, Forecast Studio may select complex ARIMA models using transformations of the input variable that contains the forecast from the external model. The forecast generated by the software then does not really provide a forecast based on the external model's correct functional form. The forecasts generated in this example shown are from a naïve benchmark model, and the alternate model selected to address this exception is shown in Fig. 6.


SAS Forecast Studio - Apr17_OOS_Main_Mth1_NoCustModels

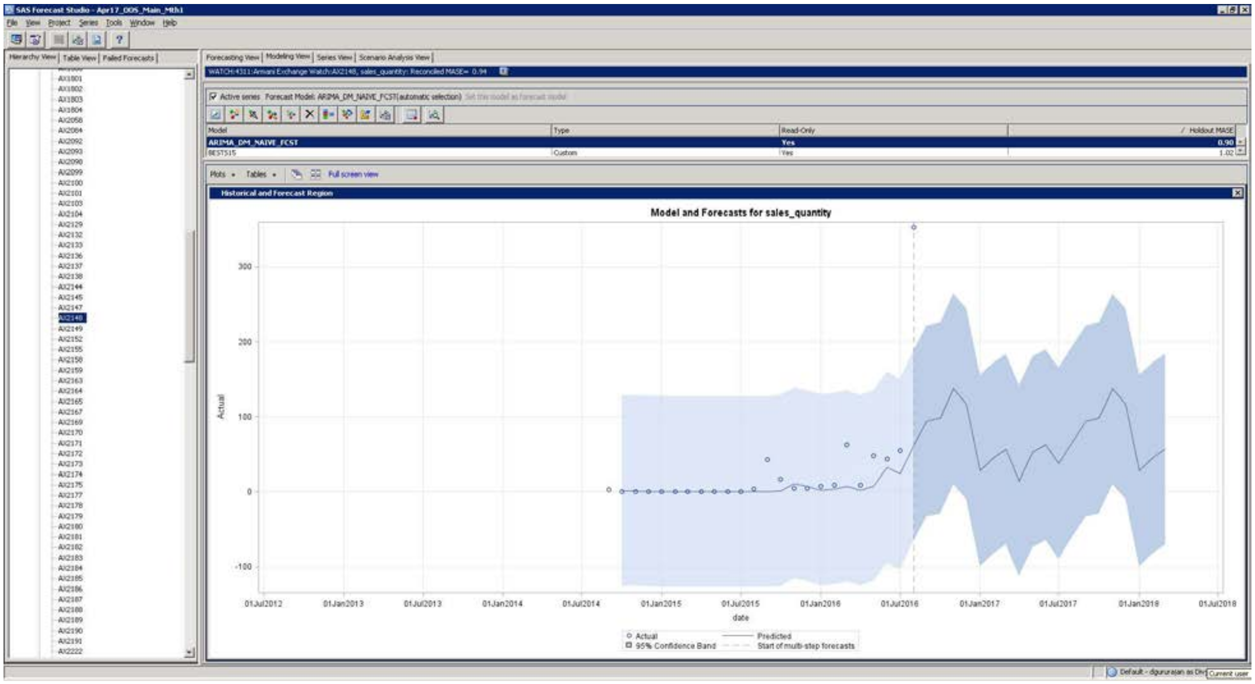
WATCH:4311:Armani Exchange Watch:AX2148, sales_quantity: Reconciled MASE= 0.93

☑ Active series Forecast Model: HPF3_286603(automatic selection) Set this model as forecast model

Model	Type	Read-Only	Holdout MASE
Generated Smoothing Model (HPF3_286603)	Generated	Yes	1.01
Generated ARIMA Model (HPF3_286602)	Generated	Yes	1.06

Default - dgururajan as Divya Gururajan

Figure 1. Example of a "runaway" forecast using trend based on a couple of data points at the end of the history period



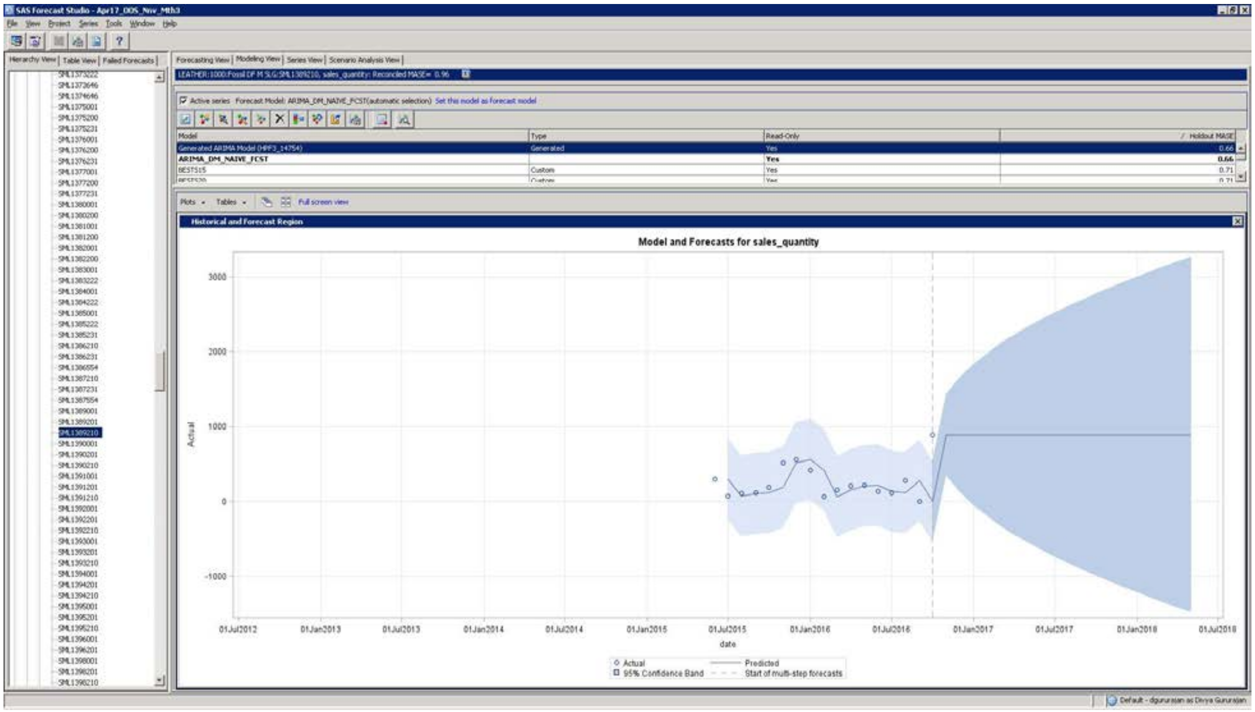
SAS Forecast Studio - Apr17_OOS_Main_Mth1

WATCH:4311:Armani Exchange Watch:AX2148, sales_quantity: Reconciled MASE= 0.94

☑ Active series Forecast Model: ARIMA_DM_NAIVE_FCST(automatic selection) Set this model as forecast model

Model	Type	Read-Only	Holdout MASE
ARIMA_DM_NAIVE_FCST		Yes	0.90
BESTS15	Custom	Yes	1.02

Figure 2. Alternate model selection to address the issue of a 'Runaway Forecast'



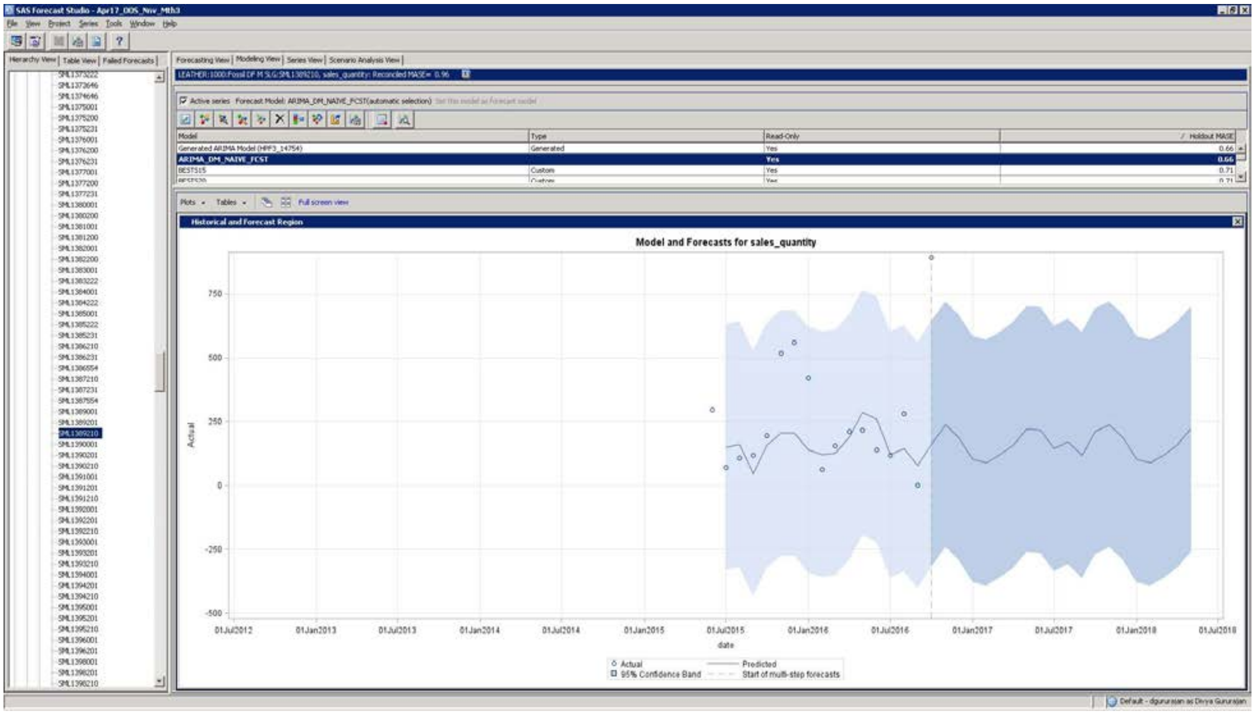
SAS Forecast Studio - Apr17_OOS_Nov_Mth3

LEATHER:1000:Fossil DF M SLG:SML1389210, sales_quantity: Reconciled MASE= 0.96

☑ Active series Forecast Model: ARIMA_DM_NAIVE_FCST(automatic selection) Set this model as forecast model

Model	Type	Read-Only	Holdout MASE
Generated ARIMA Model (HPF3_14754)	Generated	Yes	0.66
ARIMA_DM_NAIVE_FCST		**Yes**	**0.66**
BESTS15	Custom	Yes	0.71
BESTS20	Custom	Yes	0.71

Figure 3. Example of a forecast that is uniformly (on average) significantly higher than the average of actuals


SAS Forecast Studio - Apr17_OOS_Nsv_Mth3

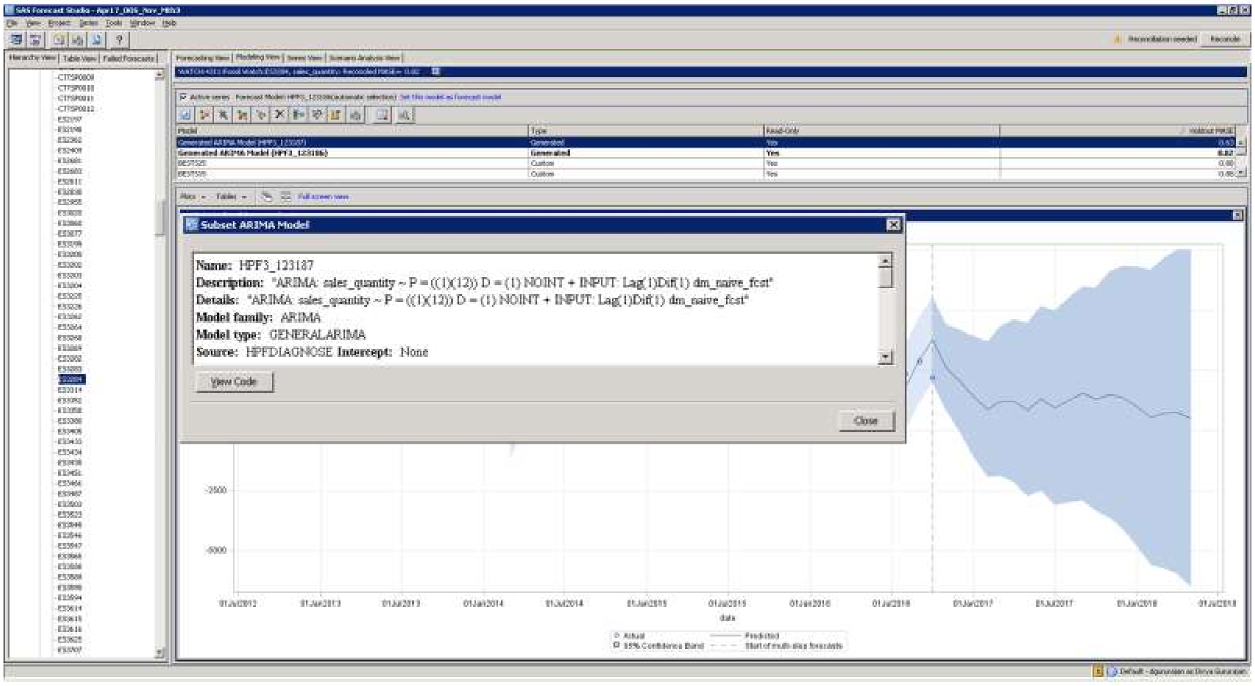
LEATHER:1000:Fossil DF M SLG:SML1389210, sales_quantity: Reconciled MASE= 0.96

☑ Active series Forecast Model: ARIMA_DM_NAIVE_FCST(automatic selection) Set this model as Forecast model

Model	Type	Read-Only	Holdout MASE
Generated ARIMA Model (HPF3_14754)	Generated	Yes	0.66
ARIMA_DM_NAIVE_FCST		**Yes**	**0.66**
BEST515	Custom	Yes	0.71
BEST520	Custom	Yes	0.71

File View Project Series Tools Window Help
Hierarchy View | Table View | Failed Forecasts
Forecasting View | Modeling View | Series View | Scenario Analysis View
Plots Tables Full screen view
Historical and Forecast Region
Model and Forecasts for sales_quantity
Actual
750
500
250
0
-250
-500
01Jul2012
01Jan2013
01Jul2013
01Jan2014
01Jul2014
01Jan2015
01Jul2015
01Jan2016
01Jul2016
01Jan2017
01Jul2017
01Jan2018
01Jul2018
date
○ Actual
□ 95% Confidence Band
Predicted
Start of multi-step forecasts
Default - dgururajan as Divya Gururajan

Figure 4. Alternate model selection that addresses the problem of a forecast that is uniformly (on average) significantly higher than the average of actuals


SAS Forecast Studio - Apr17_OOS_Nsv_Mth3

WATCH:4311:Fossil Watch:ES3304, sales_quantity: Reconciled MASE= 0.82

☑ Active series Forecast Model: HPF3_123186(automatic selection) Set this model as Forecast model

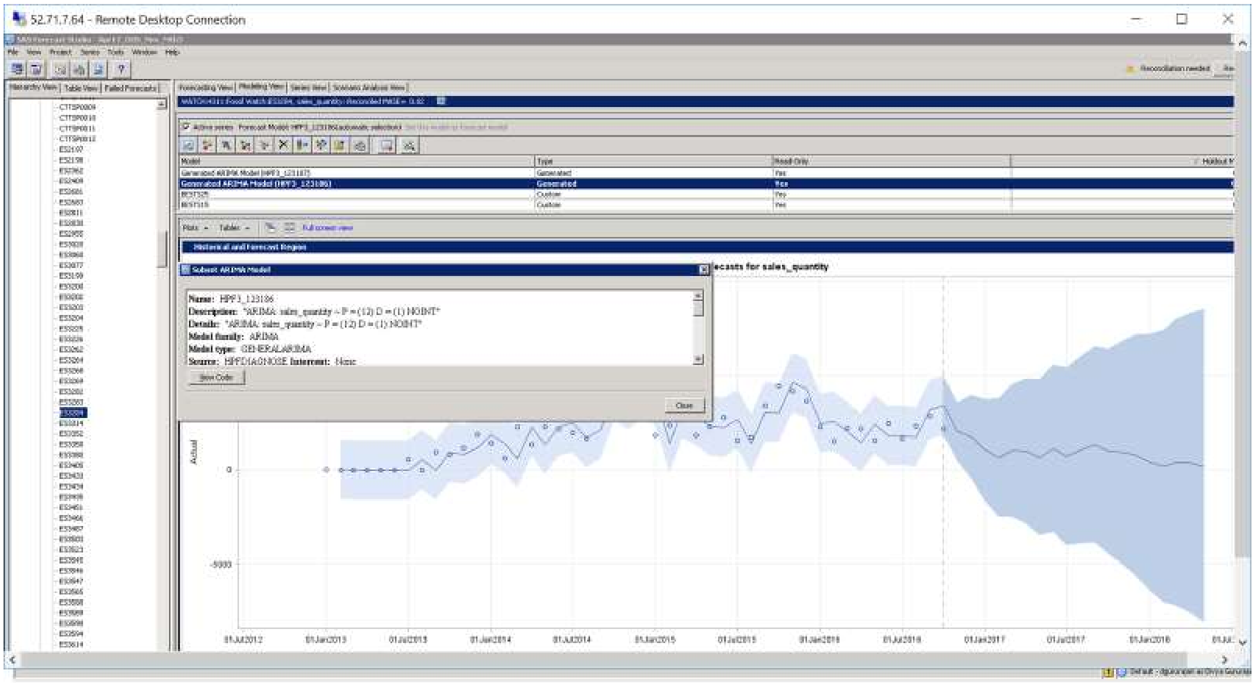
Model	Type	Read-Only	Holdout MASE
Generated ARIMA Model (HPF3_123187)	Generated	Yes	0.83
Generated ARIMA Model (HPF3_123186)	**Generated**	**Yes**	**0.82**
BEST520	Custom	Yes	0.86
BEST505	Custom	Yes	0.86

Subset ARIMA Model

Name: HPF3_123187
Description: "ARIMA: sales_quantity ~ P = ((1)(12)) D = (1) NOINT + INPUT: Lag(1)Dif(1) dm_naive_fcst"
Details: "ARIMA: sales_quantity ~ P = ((1)(12)) D = (1) NOINT + INPUT: Lag(1)Dif(1) dm_naive_fcst"
Model family: ARIMA
Model type: GENERALARIMA
Source: HPFDIAGNOSE **Intercept:** None

File View Project Series Tools Window Help
Reconciliation needed
Reconcile
Hierarchy View | Table View | Failed Forecasts
Forecasting View | Modeling View | Series View | Scenario Analysis View
Plots Tables Full screen view
View Code
Close
-2500
-5000
01Jul2012
01Jan2013
01Jul2013
01Jan2014
01Jul2014
01Jan2015
01Jul2015
01Jan2016
01Jul2016
01Jan2017
01Jul2017
01Jan2018
01Jul2018
date
○ Actual
□ 95% Confidence Band
Predicted
Start of multi-step forecasts
Default - dgururajan as Divya Gururajan

Figure 5. Example of a complex ARIMA model that is automatically generated based on an input variable



52.71.7.64 - Remote Desktop Connection

Model	Type	Read Only
Generated ARIMA Model (HPF3_123187)	Generated	Yes
Generated ARIMA Model (HPF3_123186)	Generated	Yes
BEST525	Custom	Yes
BEST515	Custom	Yes

Subset ARIMA Model

Name: HPF3_123186
Description: "ARIMA: sales_quantity ~ P = (12) D = (1) NOINT"
Details: "ARIMA: sales_quantity ~ P = (12) D = (1) NOINT"
Model family: ARIMA
Model type: GENERALARIMA
Source: HPFDIAGNOSE **Intercept:** None

Figure 6. Alternate model selection to address the problem of a complex ARIMA model that is automatically generated based on an input variable

We next describe the process we developed to select alternate models to address exceptions and finally update the Forecast Studio user interface with the alternate selection. We first identify exceptions fitting our different criteria at all levels of the hierarchy, starting from the reconciliation level and progressively moving to the forecasting level. We then select alternate models using the model related datasets and catalogs generated by Forecast Studio as part of the project. Next, we check the forecasts generated by each of the alternate models for each of the series that falls into any one of the exception buckets and select the model that minimizes the error related to the specific exception. If there are multiple models that minimize the exception error to the same level, the model with the smallest error statistic is selected. Finally, the datasets with the model parameters that underlie the Forecast Studio project are updated to reflect the alternate model selection and the changes are then reflected in the Forecast Studio user interface, when the project is re-forecasted.

PROCESS OF ALTERNATE MODEL SELECTION

We next describe the process we developed to select alternate models to address exceptions and finally update the Forecast Studio user interface with the alternate selection. We first identify exceptions fitting our different criteria at all levels of the hierarchy, starting from the reconciliation level and progressively moving to the forecasting level. We then select alternate models using the model related datasets and catalogs generated by Forecast Studio as part of the project. Next, we check the forecasts generated by each of the alternate models for each of the series that falls into any one of the exception buckets and select the model that minimizes the error related to the specific exception. If there are multiple models that minimize the exception error to the same level, the model with the smallest error statistic is selected. Finally, the datasets with the model parameters that underlie the Forecast Studio project are updated to reflect the alternate model selection and the changes are then reflected in the Forecast Studio user interface, when the project is re-forecasted.

The specific steps – coded as a SAS macro – followed in selecting alternate models after generating forecasts in Forecast Studio to address exception forecasts are as follows –

1. Check forecasts generated for each series and obtain series that fit any of the exceptions
2. Start by addressing exceptions (in a pre-specified order) identified at the reconciliation level

3. Obtain the names of all runner-up models (from the outstatselect dataset) for each series fitting an exception
4. Obtain the actual model specification for each of the runner-up models using the model catalog (autolevmodrep) generated by Forecast Studio
5. Place the alternate model specification in a custom model repository that contains only the alternate model's specification
6. Generate forecasts using Proc HPFEngine with the model repository created in Step 5
7. Run Steps 3 to 6 for all alternate models
8. Obtain forecasts from all runner-up models for all series fitting the exceptions
9. Check forecasts generated from runner-up models for each series to find model with minimum exception error
    a. In case of multiple winning models (same level of exception-related error), pick model with minimum error statistic (E.g. Mean Absolute Scaled Error etc).
    b. In case no model addresses exception, select model from a list of models (including the original) that minimizes the exception.
10. Replace the "Outest" dataset (related to the specific hierarchy level) with the parameters from the alternate model. If a series did not fit an exception, the parameters remain the same
11. Re-forecast all series using the updated "Outest" dataset and code generated in the 'Forecast_Do_Not_Import.sas' program
12. Repeat Steps 1-11 for each series until exceptions at the final forecasting level have been addressed and revised forecasts have been generated.

OUT-OF-SAMPLE ACCURACY

We next compare increases in out-of-sample accuracy from following the above process to handle exceptions to the accuracy from prior to running the exceptions algorithm as well as the accuracy of a naïve benchmark model. We find an overall substantial increase in accuracy as seen in Table 1 with an overall approximately 6% difference compared to the forecasts from prior to running the exceptions process and about a 11% compared to the naïve benchmark model.

Product / Out-of-Sample Date	**Compared to Naïve Model**	**Compared to forecasts prior to running Exceptions Process**
Product 1	**0.22%**	**11.97%**
Period 1	7.29%	0.76%
Period 2	-0.60%	29.98%
Period 3	-11.47%	2.68%
Product 2	**38.87%**	**2.33%**
Period 1	53.35%	-1.42%
Period 2	13.91%	-1.90%
Period 3	21.74%	11.67%
Product 3	**-1.86%**	**4.30%**
Period 1	10.98%	2.16%
Period 2	-12.58%	11.61%
Period 3	-11.27%	-1.56%
Overall	**11.58%**	**5.78%**

Table 1. Change in Out-of-Sample Accuracy (using MAPE) Across Models

CONCLUSION

SAS Forecast Studio provides an efficient way to generate forecasts in an automated manner for a large number of series. However, when forecast exceptions resulting from noisy data and business requirements, need to be addressed, significant manual intervention may be required. This paper addresses this problem by providing an automated and efficient alternate method to handle forecast exceptions based on any set of user-defined criteria and in doing so, also allows users to visually inspect the forecasts generated by the alternate models.

ACKNOWLEDGMENTS

We wish to thank Alex Chien, Director of Forecasting at SAS Institute for his contributions to this paper.

CONTACT INFORMATION

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

Divya Guru Rajan
CoreCompete
divya.gururajan@corecompete.com

Christopher Houck
CoreCompete
chris.houck@corecompete.com