

# SAS® GLOBAL FORUM 2021

Paper 1180-2021

## Large-scale demand forecasting for e-commerce retailer using SAS Forecast Studio: A Case Study

Ajay Kumar Mishra, Ph.D.; Mohak Narkhede; Chandu Saladi

Core Compete

### ABSTRACT

A leading e-commerce retailer with multiple websites desired to create a systematic, scalable and science-driven product forecasting system. We configured SAS Forecast Studio for this client on a cloud-based server for automated weekly forecasting of tens of thousands of products for the next year. We used SAS for data insights like understanding the distribution of product life and rate of sale, variability in demand and prices, as well as seasonality among the products. Pre-processing included creating demand history by adjusting order history for lost sales using inventory data. Price is an important predictor of demand for this retailer. We modeled holiday events using HPFEvents in SAS Forecast Server Procedures. With a deep understanding of the patterns in the data for this retailer, we configured SAS Forecast Studio including use of test runs to select independent variable form, fit criteria, reconciliation, and hierarchy choices. The heterogeneity of the patterns ensured that ARIMA and UCM with predictor variables, as well as ESM, Intermittent Demand Models and Combined models emerged as winning models for significant portion of products. We showed winning models across history of data and rate of sale. Besides using several standard reports, we created elaborate accuracy reports including measurement of Forecast Value Added. We added steps to identify exception forecasts and replace or modify them. The Forecast Studio run was automated using FSCREATE macro.

### INTRODUCTION

We implemented SAS Forecast Studio at a leading e-commerce retailer with multiple websites, for which orders were fulfilled using shared warehouses. The retailer needed forecasts at the Style-Color-Size (or, SKU), website, and week level. In this paper, we share the details of the implementation. This case study should be useful to data scientists who are implementing SAS Forecast Studio for large-scale automated forecasting or forecasting managers engaged in such an implementation.

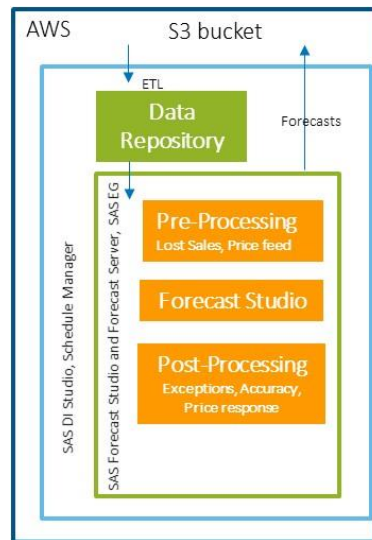
The retailer did not have any organized forecasting process. Planners were responsible for forecasts which was done in Excel. Replenishment decisions were made by them using these forecasts. They were evaluated on the replenishment decisions including product availability, revenue, and markdowns. The forecasts were not stored and not evaluated for accuracy. As a part of their digital transformation initiative, this project was used to create systematic, scalable, and science-driven product demand forecasts and a parallel project was initiated to use these forecasts for recommendations on replenishment. The goal was to equip planners with powerful recommendations to increase product availability and revenue.

### OVERVIEW OF MODULES

The paper will cover the major modules shown in Figure 1: data repository, pre-processing, Forecast Studio, Post-processing, and automation for a weekly production run.

We implemented SAS Forecast Studio on a dedicated AWS cloud server. All tables from client's various systems came into an S3 bucket and our forecast output were stored in an

S3 bucket for transfer to destination systems by the client. This entire flow was automated using SAS Management Console and Schedule Manager. SAS DI Studio was used for ETL. Input files had daily or weekly cadence. A data repository was created for all the data after staging, loading, and validating this data. While the data repository included all the data for the retailer, categories comprising about 10% of the revenue were in scope for the first phase which we were tasked to implement. Relevant data was subset for the forecasting system comprising of pre-processing, forecasting, and post-processing stages. We used SAS Enterprise Guide to create the SAS codes for these modules and SAS Forecast Studio and its backend system called SAS Forecast Server for forecasting.



**Figure 1. Overview of implementation modules**

## **BENEFITS OF SAS FORECAST STUDIO**

In order to create an automated forecasting system, the retailer looked into many options including open-source packages. SAS Forecast Studio is a front end of the forecasting system with a graphical user interface (GUI), running on top of a backend engine called SAS Forecast Server, which has many powerful and flexible procedures or macros. Some of the strengths of SAS Forecast Studio due to which it was chosen are given below:

- It has a GUI to run and visualize forecasts.
- A large-scale automated forecasting system can be created using SAS Forecast Studio and SAS Forecast Server using the SAS suite of products for data flow, automation, and analysis.
- It fits powerful families of time series models, ARIMAX, Exponential Smoothing (ESM), Unobserved Components Model (UCM), Intermittent Demand Model (IDM), and Combined. ARIMAX and UCM allow for independent variables.
- It does automatic optimization of parameters and selection of models, which is a time-consuming process in open-source and many other systems which might have the same time series models.
- Using SAS Forecast Studio saves significantly on development time. In the front-end GUI of the SAS Forecast Studio there are many options which can be set or selected to configure a system. The same options can be selected via programming using FS macros like FSCREATE.
- With easy configuration, SAS Forecast Studio does hierarchical forecasting and

reconciliation. This is particularly useful in retail.

- Events like public holidays and major repeating promotions like annual fall sale or annual spring sale are easily handled by this system using the front-end and a backend macro in SAS Forecast Server called PROC HPFEVENTS. Handling moving holidays for retail forecasting is a major development effort for open-source systems which do not handle them automatically.
- In the GUI for SAS Forecast Studio there are several additional features like overrides, filters, scenario analysis, standard reports.

## DATA FOR FORECASTING

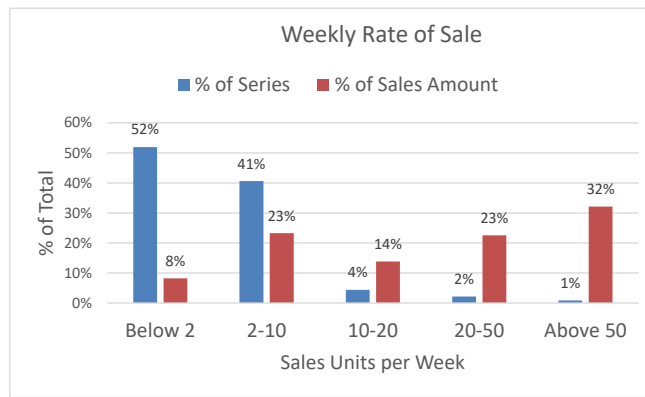
Besides sales history, retail forecasting has specific data requirements. In addition, as an e-commerce retailer additional data was available. The following are the key data sources used in the implementation:

- Sales history at SKU/Website/Date level was available for more than 2 years. Sales units and sales amount were available with additional flags to identify returns.
- Inventory snapshot was available at SKU/Date level. This helped identify whether a customer order can be fulfilled in a few days.
- Product hierarchy was available to identify the family of each SKU. The product hierarchy was SKU (or, Size), Style-Color, Style, Class, Department, and Division. Retail forecasting uses location hierarchy as well which in this case was simple, each website and all websites. The time hierarchy or retail calendar had day, fiscal week, fiscal month, fiscal quarter, and fiscal year.
- Future price data at SKU/Website/Week level was available for several weeks ahead. This could change in the future but in any week, the latest version was available.
- Website views were available at Style-Color – Website – Week level. This information was not available for the future.
- Clearance flag was available at SKU/Date level.
- The retailer had SKU segmentation which we used for reporting.

Product hierarchy was refreshed whenever it was received. For other tables, incremental data was received and appended to historical data. This created an up-to-date data lake which was useful to the client team's data scientists for purposes other than forecasting. Selected Classes were in scope for the project. Only data for those Classes were filtered and sent forward to the forecasting module.

## DATA SUMMARY

In a typical week around 55,000 series (SKU/Website) needed to be forecasted for the Classes in scope for our work. We had 2 years of history for all the series, including those which were not active. Most of the series had history of 6-12 months and a fair number had less than 6 months of history. About 20% of the series and 50% of the revenue came from series with more than a year of history. As shown in Figure 2, weekly rate of sale had a mix of high and low rate of sale as well as intermittent demand. On sales variability, a third of the series accounting for 50% of sales amount had coefficient of variation of sales units below 1. On price variability, a third of the series accounting for 78% of sales amount had coefficient of variation of price below 0.5. Finally, there was significant seasonality in the sales pattern throughout the year as is common in retail. However, the peak for Black Friday was about 4 times a regular week and the 4 weeks covering Black Friday and Cyber Monday had about 20% of the annual revenue.



**Figure 2. Rate of sale distribution**

## PRE-PROCESSING FOR FORECASTING

Several tasks were completed in pre-processing steps before the data could be used for forecasting in SAS Forecast Studio.

The client team decided which series were eligible to receive a forecast. We cannot make this decision based on activity since some are slow sellers and others might have ended their planned life recently. The client defined forecast eligibility at SKU level even though the forecasts are made at the SKU/Website level. The criteria were that the SKU's end date must be in the future, the SKU's first sale date must be in the past, and the product must be active, which in turn was defined as having at least one sale in the last 52 weeks. SKU's start date was not deemed reliable enough for this purpose: while 70% of the SKUs had the planned start date and the first sale in the same week, for others the planned start date was either well ahead of the first sale week or after the first sale week. In a pre-processing step, we processed the data so that only the history of items eligible for forecasts went to Forecast Studio.

We created independent variables during pre-processing. Price and Website views were two independent variables. Weekly future price of each SKU-Website was available as a feed. We processed this feed to create price for each week for which this feed was available. For the weeks beyond the feed, the last known price was used. For history, Average Price, defined as Sales amount/Sales units, was used since last minute promotions are common and the feed may not be reliable for the past. Website views were available at the Style/Website/Week level and hence all the SKU/Website/Week had the same value for the SKUs which were part of a style. We used 4-week moving average to smooth out views. This information was not available for the future.

The requirement was to forecast non-clearance sales only. We removed clearance sales using the Clearance flag available at SKU/Date level.

Forecast Studio can create complete sales history including weeks for which there might be no records since the sales was 0. However, we did it explicitly using SAS code for greater control on the process, including some steps in lost sales adjustment described below.

## LOST SALES ADJUSTMENT USING RULES

Forecasts need to be based on demand rather than sales. For many reasons including out-of-stock, sales for a SKU might be low on specific days. If we do not adjust the sales upwards, future forecasts will be lower than demand and out-of-stocks will perpetuate. However, making meaningful adjustments to sales history is not easy. We used some business rules to make these lost sales adjustments at SKU/Website level. We erred on the safe side to make adjustments only when there was a strong case. Significant upward revision of sales history can lead to over-forecasting. Three-fourths of series had one or

more dates adjusted. In total we increased sales units by 20% and sales amount by 15%, and got buy-in from the client team by showing aggregate adjustments at a Class level.

We restricted adjustments to scenarios where the stock was 0 and sales was deemed low, including zero sales. We flagged out-of-stock dates for each series, using inventory data. Products can have continuous sales or be seasonal. Using time between sales as a metric, we used no sales for 4 or more months to demarcate distinct seasons; any adjustment was made only within a season, not between seasons. We identified dates with 0 or low positive values relative to its neighborhood weeks for a SKU/Website as dates with lost sales. We excluded such dates in a long stretch of no sales since making adjustments to a large number of weeks requires greater analytical complexity than was in scope for this project. In addition, we used additional data cleaning steps like minimum inventory and total sales in life, to adjust only significant series. We used moving average of previous 14 dates with positive sales to get smooth but recent level of sales to make a reasonable estimate of what the sales could have been. Finally, we created summaries to check the extent of adjustment. Since this rule-based adjustment worked at a SKU/Website level, there was a likelihood that many series were simultaneously adjusted upwards for the same set of weeks while in reality if all of the SKUs had been in stock, there would be substitution and the total sales would not have increased significantly. We fine tuned the rules to ensure that adjustments were reasonable and erred on the side of less adjustment. Figure 3 illustrates upward adjustment in sales for a sample series. It shows conservative adjustments too.



**Figure 3. Illustration of lost sales adjustment**

## CONFIGURATION OF FORECAST STUDIO

SAS Forecast Studio allows configuration from the GUI but we used matching options in FSCREATE macro which can be used in the backend SAS code and batch runs, which was useful in automation. Since we had to forecast around 55,000 series, we used one Forecast Studio project. In larger implementations, we would have used multiple Forecast Studio projects to keep the runtime reasonable and performance of the GUI acceptable; we would configure them separately to get the best forecast for each project.

Sales Units was the dependent variable to be forecast. Price and Website Views were independent variables. In addition, events were defined as explained later. Forecasts were created for 52 weeks (Lead = 52). Forecast Studio does not have an option to give different end dates for each series. If a SKU had end date within those 52 weeks and hence was not eligible beyond the end date, in post-processing, we set those forecasts to 0.

After discussion with the client, we used the following hierarchy for forecasting: Website, Department, Class, Style, Style-Color, SKU. This ensured that at the lowest level, there was

a SKU-level forecast for each Website. The team recognized that SKU-level history would be more granular than Style-Color, thus reducing the quality of forecasts. We could have forecast at Style-Color/Website level if we had robust size profiles to split Style-Color/Website forecast to SKU/Website. Since in history series start in different weeks, as more series start, at higher levels it looks as if sales is increasing in time when in reality, incremental sales are coming due to greater count of series. Hence, for aggregation we chose the Average option which gives average sales per series. Reconciliation across hierarchies is an important decision. After conducting trials at various levels, we chose Middle-Out reconciliation at Department level. For disaggregation, we chose proportion to determine the loss function used for reconciliation.

SAS Forecast Studio has sophisticated and robust implementations of major families of methods in time-series forecasting: ARIMAX, ESM, UCM, IDM, and Combined. This is done via its backend engine SAS Forecast Server which comprises of SAS procedures for these methods. E.g., HPFARIMASPEC procedure does ARIMAX modeling while HPFESMSPEC procedure models the ESM family. Auto Regressive Integrated Moving Average, including predictor variables (also called ARIMAX) is a powerful family of models which creates excellent time series forecasts. It is built to handle typical structures in time series data, besides predictor variables. ESM or Exponential Smoothing Model family models time series as a function of horizontal level, trend, seasonality. This family includes many variations including the simplest form, Simple Exponential Smoothing, and the most general form, Holt-Winter. It does not consider predictor variables. It often emerges as a winning method when there is significant seasonality, which is usually true at higher levels in the product hierarchy like Class or Department. UCM or Unobserved Components Model models time series as a function of horizontal level, trend, seasonality, cyclicity, all of which can change over time rather than be static. It allows for predictor variables too. IDM or Intermittent Demand Models is suited to demand which arises occasionally, i.e., there are lots of zeros in history. Finally, Combined has composite models combining two or more models. A differentiator for SAS Forecast Server is that for each of these families, SAS optimizes the parameters automatically making it efficient for a data scientist on both effort and run-time.

Model Family	Share of Forecasts in a sample
ESM	36%
ARIMA	27%
IDM	19%
UCM	10%
COMBINED	4%
NONE (Uses Mean)	3%

**Figure 4. Methods used to create forecasts at the SKU/Website level**

Additional salient settings are listed below. See Forecast Server Administrator’s Guide to understand these settings in FSCREATE.

ARIMAX = YES, ESM = YES, UCM = YES, COMBINE = YES

MINOBSNONMEAN = 12, MINOBSSEASON = 1, MINOBSSTREND = 12,

CRITERION = MAPE

INTERMITTENT = 2, SEASONTTEST = 0.02, DETECTOUTLIERS = YES, MAXNUMOUTLIERS = 2, MAXPCTOUTLIERS = 2

TRANSTYPE = NONE

ALLOWNEGATIVE = NO



For choosing several settings we conducted controlled test runs to compare between options. For fit, we compared in-sample vs hold-out sample, and in-sample fit won. For accuracy criterion, MAPE won over RMSE. Among transformation options, None was a winner over either Boxcox or Auto, where Auto picks log or none. We explored modeling average price as a discount but retained price as the format for the predictor variable.

Holidays or major annual promotion events have a significant impact on retail sales. When these move to different fiscal weeks year over year, it is an analytical challenge to estimate the lift due to the holiday and apply it in the right week in the future, for high forecast accuracy. Easter is a common example. Since it moves across years and is usually significant for retail forecasts, the Easter effect needs to be applied in the correct week in the future. When holidays do not move, they are captured by seasonality and do not require special modeling effort. The flexibility in event modeling is a major advantage in SAS Forecast Server. Forecast Studio allows specification using the GUI but upload of holiday file gives more flexibility in modeling holidays. We used HPFEVENTS procedure from SAS Forecast Server to create an external event repository for use with Forecast Studio. Sample definition of an event is shown below. There are several holidays in the library, like Thanksgiving for the U.S., stored as keywords which can be used directly; for others dates must be given. In this example Black Friday is specified as a temporary correction which grows starting two weeks before the week of Thanksgiving with peak in the week of Thanksgiving, given as maximum amplitude of Value = 1, 0.75 in the week before and  $(0.75)^2 = 0.5625$  in the preceding week.

```
EVENTDEF BlackFriday = THANKSGIVING/ TYPE=TC BEFORE=(DURATION=2  
SLOPE=GROWTH ) TCPARM=0.75 PULSE=WEEK VALUE=1;
```

Visualization of forecasts in SAS Forecast Studio is excellent. We saw that hierarchical reconciliation improved the accuracy significantly, particularly for series with less than 12 months of history; reconciliation gave much better shape to such series.

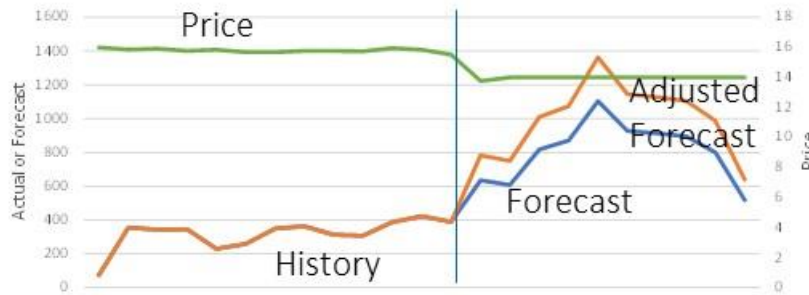
## POST-PROCESSING AND ACCURACY REPORTING

SAS Forecast Studio creates forecasts using the data created in pre-processing steps, as per the configuration given in the section above. Outputs are stored in a large number of output tables, for each level of hierarchical forecast. We added some post-processing steps before formatting the output in the format desired by the client for downstream applications.

Statistical forecasting methods sometimes create values which seem odd and require exception handling. We checked for Size/Website/Week level exceptions like missing values or unreasonably high values and Size/Website level exceptions like extreme values or unexpected number of zeros. We replaced exceptions with rule-based forecasts using recent level and seasonality index.

Forecast output needed prediction errors which are part of the output from SAS Forecast Studio. Mostly the prediction errors were reasonable relative to prediction mean, which are the forecasts. Large prediction errors usually occurred when the forecasts were small. We used prediction error analysis and capping to adjust them.

We noticed that some series do not have price variation in history due to which forecasts like those driven by Exponential Smoothing family or insignificant predictor variable in ARIMAX, would ignore price change in the future. We added price-based adjustment of forecasts. If price variation in history was low, we adjusted the forecasts using Class-level price elasticity given by the client. An example is in Figure 5 below. Price variation in history was not significant but in the forecast period there was a sharp drop. Our adjustment increased the forecast.



**Figure 5. Price-based adjustment of forecasts**

## **CUSTOM ACCURACY REPORTING**

We enabled standard reports which are in-built in SAS Forecast Studio. Custom accuracy reports for review by forecasting analysts and business leaders were part of the requirements. We created a module on accuracy that ran after creating forecasts. It used past forecasts and newly available actual sales to create accuracy metrics, and output formatted tables in Excel using SAS code.

In consultation with the client, Weighted Mean Absolute Percentage Error (MAPE), weighted by revenue was chosen as the primary metric. We added bias and mean percentage error too. MAPE was calculated at the Size/Website level but reported at Department level as a weighted metric. While we created forecasts for the next 52 weeks, for accuracy assessment the horizon chosen was 4 weeks. We used 4-week ahead forecasts, taken cumulatively and compared it with actual for those weeks. These decisions were suitable because supply chain and replenishment decisions from domestic fulfillment had lead time of 2-4 weeks and decisions used demand over multiple weeks rather than focus on a specific week.

We reported Forecast Value Added as well to show improvement in accuracy over simple forecast created using 4-week moving average or last-year's value, if it was available. This simple metric was highly effective for short-term forecasts but our model driven forecasts showed improvement of 5 to 10% in Weighted MAPE over the simple forecast. Forecast Value added increased as trend, seasonality, and association with price increased.

We made the accuracy measurement report meaningful by judging model performance fairly. We made two adjustments in consultation with the client team. We defined price error as a scenario where actual Average Price varied by more than 5% from the price given to the model and dropped it for accuracy reporting. We defined availability error as the scenario where an item was out-of-stock, in which case forecasts would exceed actual and hence should not be used in measurement. While these adjustments dropped 20%-40% of the series, the metric was meaningful for the end users to assess the efficacy of the forecasts.

We added cross-tabulations of MAPE buckets with magnitude of actual sales units, product life length, weekly rate of sale, and client's product segmentation. E.g., we saw that weighted MAPE was higher than normal when the actual units were less than 10. This was reassuring to the client team since being off significantly on small numbers is not as big a concern as high errors when the actual units are high. As we would expect, MAPE was lower when product life length was longer or rate of sale was higher.

## **CONCLUSION**

In this paper, we presented a case study on how we implemented SAS Forecast Studio for an e-commerce retailer. SAS Forecast Studio is easy to configure using GUI and matching FS macros for automated batch runs. It fits important families of forecasting algorithms,



with automatic parameter optimization. Analysts can tune a Forecast Studio project for better forecasts by trying various combinations of configuration settings. Automatic handling of hierarchical forecasting, reconciliation, and events makes the project efficient and improves quality of forecasts. Using SAS programming, pre-processing and post-processing can be done in a flexible manner to meet client's business requirements and improve adoption.

## RECOMMENDED READING

- *Brenda Wolfe, B., M. Leonard, and P. Fahey, Introducing SAS® Forecast Studio Paper 193-30, Proceedings of SAS Global Conference SUGI 30*
- *Leonard, M., Large-Scale Automatic Forecasting: Millions of Forecasts, SAS White Paper*
- *Looking Inside SAS® Forecast Studio, SAS White Paper*
- *Forecasting 101: How to Get More Accurate Forecasts With Less Cost and Effort, SAS White Paper*
- *Forecast Value Added Analysis: Step by Step, SAS White Paper*
- *SAS Forecast Studio 15.1: User's Guide*
- *SAS Forecast Server Procedures 15.1: User's Guide*
- *SAS Forecast Server 15.1: Administrator's Guide*

## CONTACT INFORMATION

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

Ajay Kumar Mishra  
Core Compete, 5001 South Miami Blvd., Suite 410, Durham, NC 27703  
Ajay.Mishra@CoreCompete.com  
www.CoreCompete.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.