# **Sales Volume Forecast****ing**



|  |  |
| --- | --- |
| **University Of Connecticut**    **Adwaith Ajith Menon**  **TABLE OF CONTENTS**   1. **INTRODUCTION ………………………………………………………………. 2** 2. **LITERATURE…………………………………………………………………… 3** 3. **DATA ANALYSIS (PRE-MODELING) ………………………………………. 4**    1. **DATA DESCRIPTION …………………………………………………. 4**    2. **DATA RESTRUCTURING …………………………………………….. 7**    3. **SUMMARY STATISTICS ……………………………………………… 8** 4. **EXPLORATORY DATA ANALYSIS ………………………………………….. 9** 5. **MODELS …………………………………………………………………………. 14**    1. **PREDICTIVE MODELS ………………………………………………... 14**       1. **LINEAR REGRESSION ………………………………………… 14**       2. **XGBOOST MODEL ……………………………………………... 16**       3. **RANDOM FOREST MODEL …………………………………… 19**    2. **TIME SERIES MODELS ……………………………………………….. 20**       1. **ARIMA IN PYTHON ……………………………………………. 20**       2. **ARIMA IN SAS ………………………………………………….. 28** 6. **STORE-WISE RESULTS ………………………………………………………. 32** 7. **MODEL COMPARISON ……………………………………………………….. 33** 8. **RECOMMENDATIONS ………………………………………………………... 34** 9. **CONCLUSIONS & FUTURE WORK …………………………………………. 36** 10. **REFERENCES …………………………………………………………………... 36** 11. **RESPONSIBILITIES OF EACH TEAM MEMBER ………………………… 37** 12. **APPENDIX ……………………………………………………………………… 38** |  |

1. **INTRODUCTION**

Favorita is an Ecuador-based business that specializes in supermarket retail. Our data set is taken from one of Kaggle's ongoing competitions that emphasize the importance of brick and mortar stores to estimate sales.

The dataset includes data from various stores across the country as well as quantitative sales (sales volume) for all families (categories of items in the store) by date. The dataset also includes the number of items on promotion on a specific day at a specific store. Our goal is to forecast sales volume for the upcoming 16 days and use historical data to improve forecasts.

For brick-and-mortar retail stores, which must carefully balance how much inventory to acquire, sales volume estimates are critical. The stores are left with an excess of perishable goods if the forecasts are a little off. Accurate forecasting could help stores ensure they have just enough products at the correct time. If a store can match a product's demand with just the correct amount of supply, there will be no missed sales owing to a lack of inventory and no additional costs due to overstocking. Based on the estimates, the retailers would gain information on how to stock products, eventually retaining clients and ensuring consumer satisfaction. When it comes to perishable commodities, it is essential to ensure that there is enough and that nothing goes to waste owing to rising demand. Through this study, we also intend to assist Favorita in being profitable by examining the data based on the location . Depending on the forecasting model we ultimately select, some recommendations we would want to offer include optimizing the number of stores by boosting the number of items on promotion, implementing customer loyalty programs, and restructuring the staffing schedule.

We also believe that subjective forecasting approaches are insufficient for making decisions and must be supplemented with data science. As more products are introduced to the market and retailers strive to increase sales, complexity rises.

1. **LITERATURE**

A time series is a series of observations taken in chronological order. It can be broken down into three parts.: Trend (Change in Time Series), Seasonality (the repetition of a specific pattern of observations after a certain time interval) ,and Irregular component (Random noise). These three components serve as the basis for decomposition techniques that separate original series into individual components. These techniques make up a significant portion of the forecasting literature.

Time-series forecasting problems cannot be solved with a single optimum method. Each issue may be resolved in a different way to cope with time-series forecasting. One of the simplest prediction methods for time series without obvious seasonal patterns is the Moving Average (MA) method. A more sophisticated variation of MA known as Autoregressive Integrated Moving Average (ARIMA) has been employed in various papers.

Multivariate ARIMA was effectively used by Huber, Gossmann, and Stuckenschmidt (2017)**[1]** to forecast demand for perishable items. Another classic forecasting technique known as Seasonal ARIMA (SARIMA) has been successfully used for applications like forecasting tourism demand (Goh & Law, 2002)**[2]** and predicting the flow of motor traffic (Williams & Hoel, 2003)**[3]**.

Moreover, the forecasting technique of Exponential Smoothing has proven quite effective. For example, Taylor (2011)**[4]** used a Seasonal Exponential Smoothing approach to forecast the daily sales of a supermarket (which was highly volatile).

# **DATA ANALYSIS (PRE-MODELING)**

**3.1 DATA DESCRIPTION:**

Initially, the project had 5 .CSV files.

**Train.csv**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Date | Day wise YYYY-MM-DD | date |
| Store\_nbr | Store ID | Categorical |
| Family | Identify the type of product sold | Categorical |
| sales | Total sales a particular store on a particular day | Continuous |
| onpromotion | Total number of items being promoted | Continuous |

**Oil.csv**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Date | Day wise YYYY-MM-DD | date |
| dcoilwtico | Day-wise oil price | Continuous |

**Holiday.csv**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Date | Day wise YYYY-MM-DD | date |
| type | Type of Holidays & Events | Categorical |
| locale | Regional information | Categorical |
| locale\_name | Regional information | Categorical |
| description | Description of Holiday | Categorical |
| transferred | If a holiday is transformed or not. | Categorical |

**Stores.csv**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Store\_nbr | Store ID | Categorical |
| city | City the store is located in | Categorical |
| state | State the store is located in | Categorical |
| type | Type of the store | Categorical |
| cluster | Grouping of similar stores | Categorical |

**Transactions.csv**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Date | Day wise YYYY-MM-DD | date |
| store\_nbr | Store ID | Categorical |
| transactions | Total number of transactions of a store on a particular day | Continuous |

**3.2 DATA RESTRUCTURING:**

All 5 CSV files were merged by date and the final dataset had 13 features.

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Description** | **Type** |
| Date | Day wise YYYY-MM-DD | date |
| Sales | Total sales a particular store | Continuous |
| Promotion | Total number of items being promoted | Continuous |
| Oil Price | Day-wise oil price | Continuous |
| Type\_x | Holidays & Events | Categorical |
| Transaction | Total transactions by date & store | Continuous |
| locale\_name,city, state | Region information | Categorical |
| Transferred | Whether holiday is transferred to other day | Categorical |
| cluster | Grouping of similar store | Categorical |
| family | Identify the type of product sold | Categorical |
| Store\_nbr | Store ID | Categorical |

We then proceeded with extracting store-level-wise data. Since we had 54 stores, we had 54 individual datasets. After extracting we removed irrelevant features like locale\_name, description, transferred, locale, state, type\_y, and cluster.

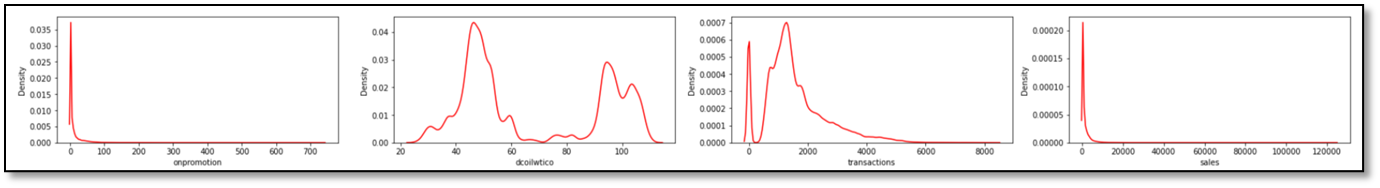
We used as.freq method to convert data to time series with a constant period. Here we used the day as the period. We also imputed the missing values backfill method. Finally, we had around 1684 unique dates starting from 1st January 2013 to 15th August 2017.

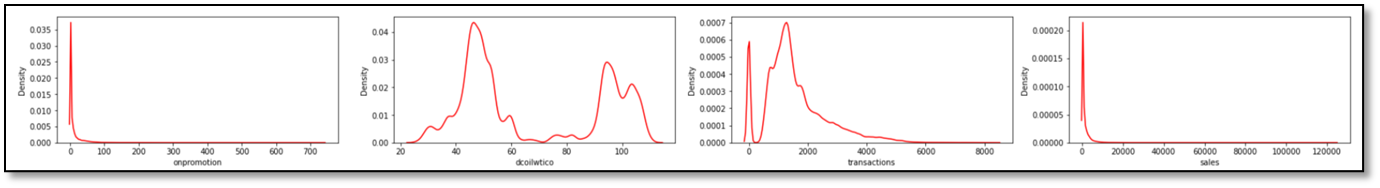
**3.3 SUMMARY STATISTICS**

**Table

Description automatically generated**Below are the summary statistics of the 4 continuous features in the final dataset.We observed that the sales volume is varying from 0 to 124k. Also, a similar trend is observed for onpromotions.

Below are the distribution plots of sales, onpromotion, dcoilwtico, and transactions. From the distribution plots, we can see that the distribution curve for both onpromotion and sales are right-skewed. From the distribution plot of oil price, we can observe bimodal distribution which means there are two events where high prices of oil were recorded.

****

****

1. **EXPLORATORY DATA ANALYSIS**

* **Correlations between continuous variables**

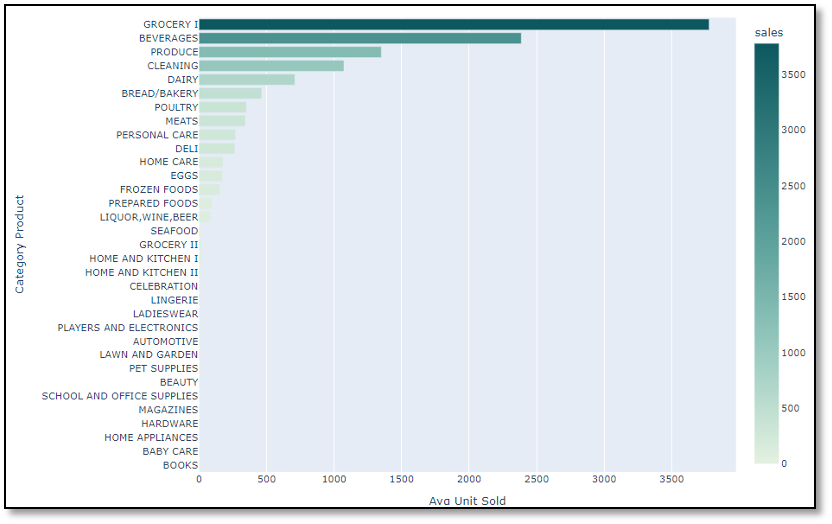
We have plotted a correlation plot for all the continuous features. From the plot, we have observed that onpromotion (0.43) and transaction (0.23) have a certain positive correlation with sales, which means that the growth of promotion and transaction can lead to the growth of the total sales. On the other hand, there is a very slight negative correlation between oil price and sales (-0.076).

* **Oil price vs Sales**

****

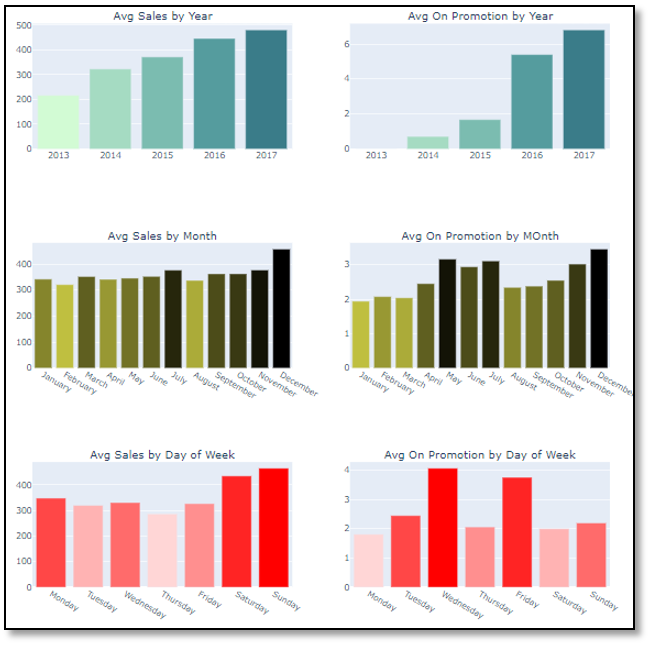
We have plotted a scatter plot with oil price against sales. From the plot, we have observed that lower oil prices mean more sales and higher oil prices have relatively fewer sales. In addition to people driving out less, the rise in oil prices has a great impact on operating costs and might be the reason for fewer sales.

* **Popular Product Family**



This image shows the total sales of different types of goods in all stores. Grocery1, beverage, and produce are the three categories of commodities with the highest sales. These are all daily necessities related to food.

* **Average Sales & Promotion vs Time**



This plot is a time exploration of sales and promotion. It can be seen from the graph in the first row that promotion and sales are increasing with time. Sales is growing steadily and slowly, but promotion is growing exponentially. The mismatch between the promotion growth and the sales growth may be related to the product attributes. Because the best-selling products are necessities of life, which are a kind of goods with very small demand elasticity, many promotions have not brought about significant sales growth.

The second row and the third row show the sales and promotion changes at the monthly and daily levels. December is the time when the sales is the highest in the whole year, winter and summer vacations are the time when the promotion is the most, which may be related to holidays. The weekly sales peak occurs on weekends, but the promotion peak occurs on Wednesday . This may be a strategy to attract customers on workdays.

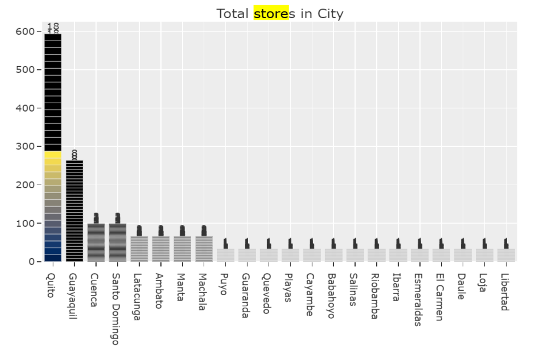
* **Average Sales vs Location & Product Family & Oil Price**



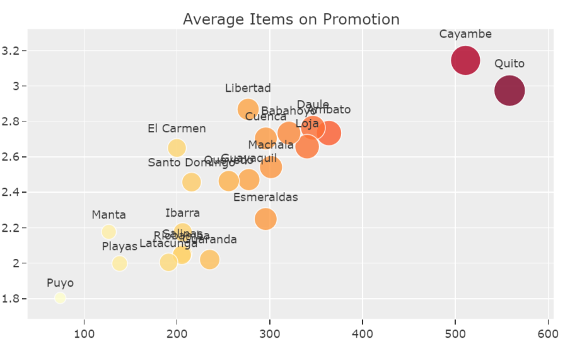
The first row is the analysis related to location. The best-selling stores appear in Quito, Pichincha State. Quito is the capital of Ecuador.

The bottom right graph shows oil prices versus average sales over time, which confirms the negative correlation between oil prices and sales.

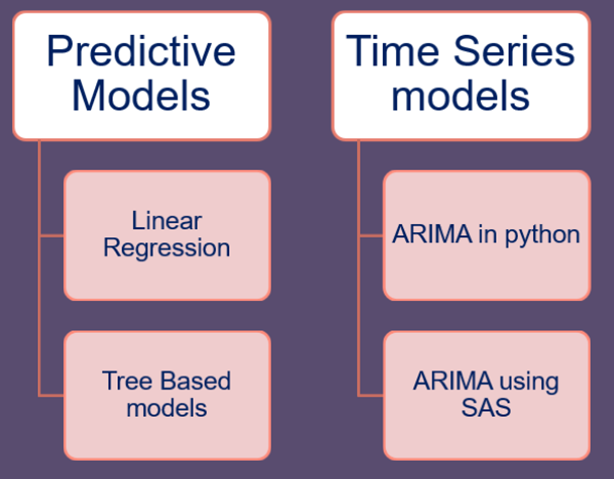
* **Stores distribution**



We have 54 stores in total, and Quito occupies one-third, which is far more than other cities.

* **Average Promotion per City**

From the above plot, we can observe that Quito and Calamba are the cities with the most promotions, which is one of the main reasons for the high sales of stores in these two cities.

1. **MODEL****S**

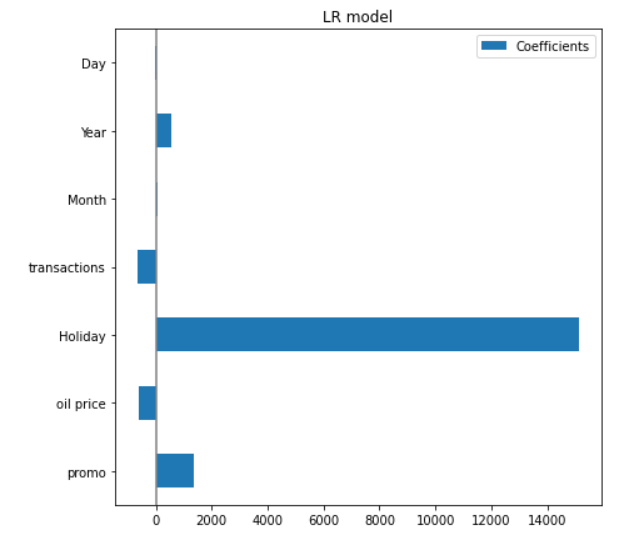
Two types of modeling are performed. One set of models belongs to predictive models and another set belongs to Time series models. The following models are applied on store10, remaining store results are shown in the appendix section.

**5.1 PREDICTIVE MODELS**

**5.1.1 LINEAR REGRESSION MODEL**

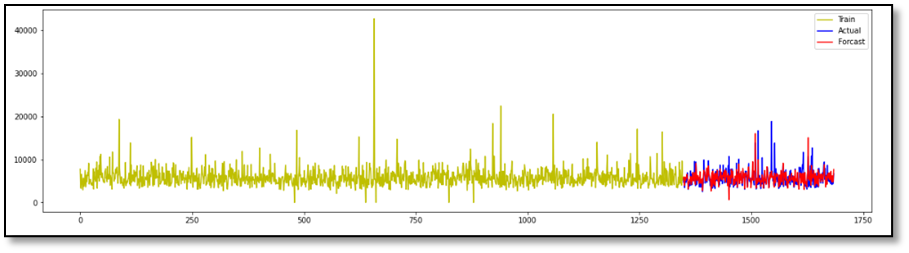
In statistics, linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.

If the goal is prediction, forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

Below is the feature contribution plot for the Linear regression model for store 10.

Chart, scatter chart

Description automatically generatedBelow is the Predicted vs Actual Values plot for Store Number 10.

The above plot shows Actual vs Predicted results for Linear Regression.

**5.1.2 XGBOOST REGRESSION MODEL**

XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e. how far the model results are from the real values. The most common loss functions in XGBoost for regression problems is reg:linear, and that for binary classification is reg:logistics. Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods. XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancel out and better one sums up to form final good predictions. Some of the important parameters are highlighted below:

* **eta [default=0.3]**

Analogous to learning rate in GBM. Makes the model more robust by shrinking the weights on each step. Typical final values to be used: 0.01-0.2

* **min\_child\_weight [default=1]**

Defines the minimum sum of weights of all observations required in a child. This is similar to min\_child\_leaf in GBM but not exactly. This refers to min “sum of weights” of observations while GBM has min “number of observations”. Used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.

* **max\_depth [default=6]**

The maximum depth of a tree, same as GBM. Used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample. Should be tuned using CV. Typical values: 3-10

* **max\_leaf\_nodes**

The maximum number of terminal nodes or leaves in a tree. Can be defined in place of max\_depth. Since binary trees are created, a depth of ‘n’ would produce a maximum of 2^n leaves. If this is defined, GBM will ignore max\_depth.

* **gamma [default=0]**

A node is split only when the resulting split gives a positive reduction in the loss function. Gamma specifies the minimum loss reduction required to make a split. Makes the algorithm conservative. The values can vary depending on the loss function and should be tuned.

* **max\_delta\_step [default=0]**

In maximum delta step we allow each tree’s weight estimation to be. If the value is set to 0, it means there is no constraint. If it is set to a positive value, it can help making the update step more conservative. Usually this parameter is not needed, but it might help in logistic regression when class is extremely imbalanced. This is generally not used but you can explore further if you wish.

* **subsample [default=1]**

Same as the subsample of GBM. Denotes the fraction of observations to be randomly samples for each tree. Lower values make the algorithm more conservative and prevents overfitting but too small values might lead to under-fitting. Typical values: 0.5-1

* **colsample\_bytree [default=1]**

Similar to max\_features in GBM. Denotes the fraction of columns to be randomly samples for each tree. Typical values: 0.5-1

* **colsample\_bylevel [default=1]**

Denotes the subsample ratio of columns for each split, in each level. I don’t use this often because subsample and colsample\_bytree will do the job for you. but you can explore further if you feel so.

* **lambda [default=1]**

L2 regularization term on weights (analogous to Ridge regression). This used to handle the regularization part of XGBoost. Though many data scientists don’t use it often, it should be explored to reduce overfitting.

* **alpha [default=0]**

L1 regularization term on weight (analogous to Lasso regression). Can be used in case of very high dimensionality so that the algorithm runs faster when implemented

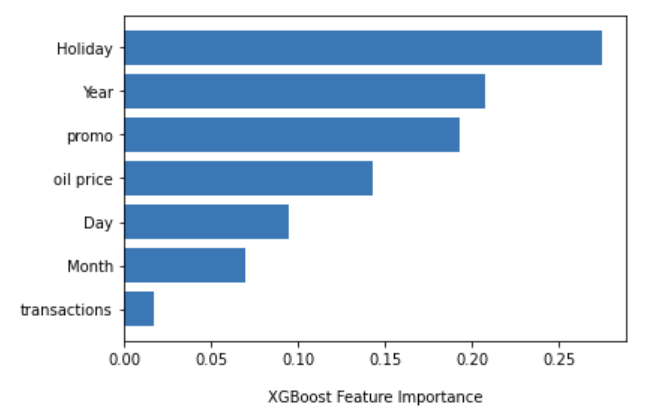
* **scale\_pos\_weight [default=1]**

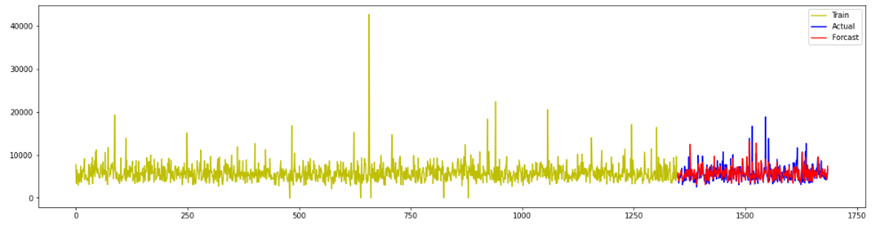
A value greater than 0 should be used in case of high class imbalance as it helps in faster convergence.

When we ran the XGBoost regression model with the default parameters for store 10 we achieved a MAPE of 0.128, RMSE of 1295.44 ,and R2 of 0.586.

After Hyperparameter tuning we found silent=1, eta=0.02, max\_depth=5, subsample=1 to be the best parameters. After fitting the model with the above parameters, we observed that MAPE improved to 0.121, R2 to 0.591 and, RMSE to 1286.93.

Below is the feature importance plot for the XG Boost model. From the plot we can see that holiday, year and promo are the top 3 important features and the transactions feature has the least importance.





The above plot shows Actual vs Predicted results for XGBoost

**5.1.3 RANDOM FOREST REGRESSION MODEL**

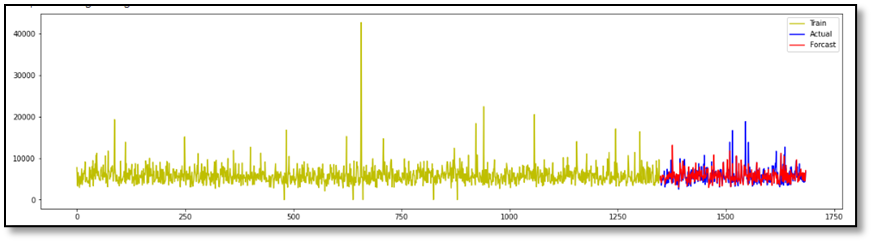
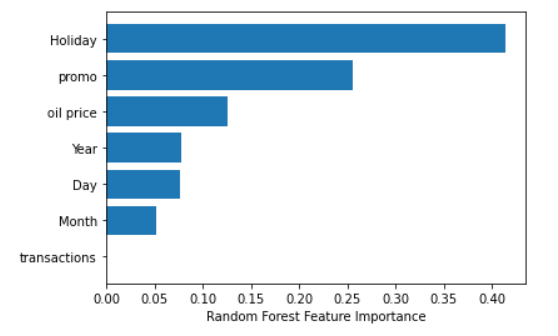
We will use the sklearn module for training our random forest regression model, specifically the RandomForestRegressor function. The RandomForestRegressor documentation shows many different parameters we can select for our model. Some of the important parameters are highlighted below:

* n\_estimators — the number of decision trees you will be running in the model
* criterion — this variable allows you to select the criterion (loss function) used to determine model outcomes. We can select from loss functions such as mean squared error (MSE) and mean absolute error (MAE). The default value is MSE.
* max\_depth — this sets the maximum possible depth of each tree
* max\_features — the maximum number of features the model will consider when determining a split
* bootstrap — the default value for this is True, meaning the model follows bootstrapping principles (defined earlier)
* max\_samples — This parameter assumes bootstrapping is set to True, if not, this parameter doesn’t apply. In the case of True, this value sets the largest size of each sample for each tree.
* Other important parameters are min\_samples\_split, min\_samples\_leaf, n\_jobs, and others that can be read in the sklearn’s RandomForestRegressor documentation

When we ran the Random Forest regression model with the default parameters for store 10 we achieved an MAPE of 0.127, RMSE of 1280.14 and R2 of 0.59.

After Hyperparameter tuning we found n\_estimators=200, max\_depth=9, max\_features=4, min\_samples\_leaf=2, min\_samples\_split=10 to be the best parameters. After fitting the model with the above parameters, there was no improvement in the MAPE, RMSE, and R2 values. So, we proceeded with the default model.

The below is the feature importance plot of the Random Forest model. As we can observe, holidays feature is the most important parameter and transactions feature has negligible importance for Store 10.

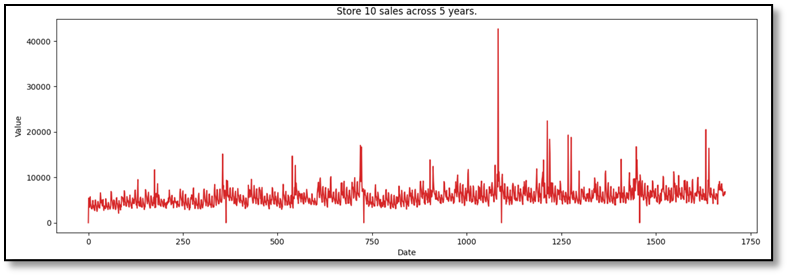


The above plot shows Actual vs Predicted results for XGBoost model.

**5.2 TIME SERIES MODELS**

**5.2.1 ARIMA IN PYTHON**

The first step while performing Time Series Analysis is to have a look at how it is varying. Plotted the sales plot against time to understand this. When closely observed, very abrupt changes in the sales are recorded.



To get a clear picture of seasonality, trend, and residuals. It is needed to inspect seasonal decomposition in python using statsmodels.

**TIME SERIES DECOMPOSITION**

Time series decomposition is a technique that splits a time series into several components, each representing an underlying pattern category, trend, seasonality, and noise.

**Seasonality:**

Seasonality describes the periodic signal in your time series.

**Trend:**

Trend describes whether the time series is decreasing, constant, or increasing over time.

**Noise:**

Noise describes what remains behind the separation of seasonality and trend from the time series. In other words, it’s the variability in the data that cannot be explained by the model.

Graphical user interface, application, Word

Description automatically generated

**Observations**

* In the figure, we can identify a very slight change in trend.
* The seasonality pattern which we are seeing is because of repeated short cycles within time series.

**AUGMENTED DICKEY FULLER TEST (ADF Test)**

In time series forecasting, the first step is to determine the number of differencing required to make the series stationary. ADF is a test for stationarity.

The null hypothesis for this test is that there is a unit root. The alternate hypothesis states that the time series is stationary (or trend-stationary).

Below are results of ADF test:

*ADF Statistic: -5.45*

*n\_lags: 2.5e-06*

***p-value: 2.57e-06***

*Critical Values: 1%, -3.43*

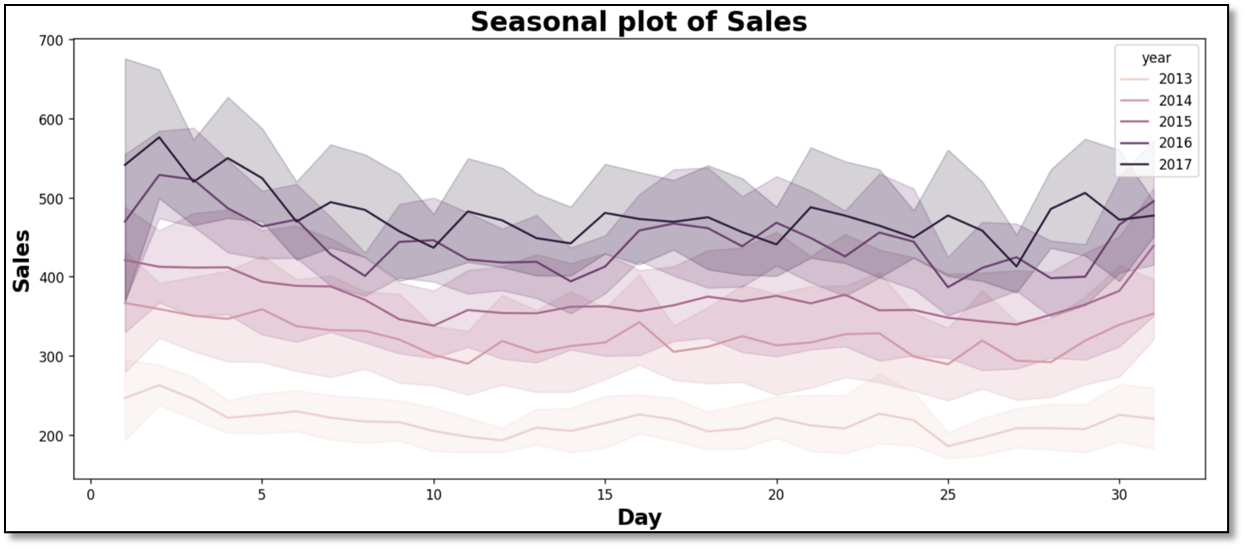
***Critical Values: 5%, -2.81***

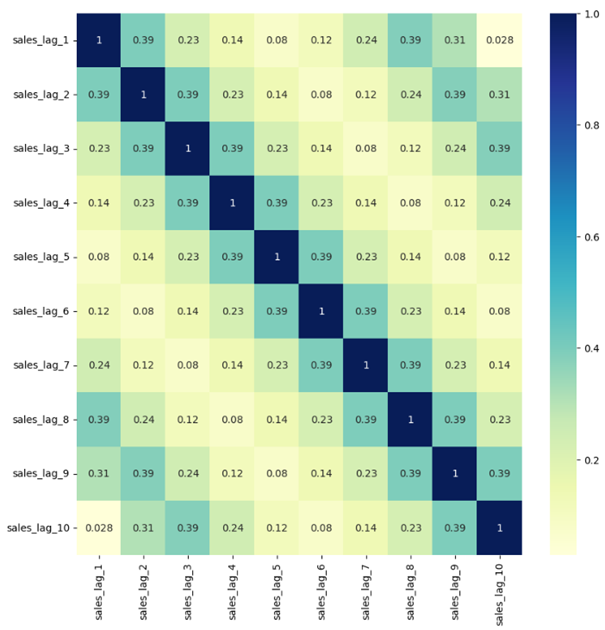
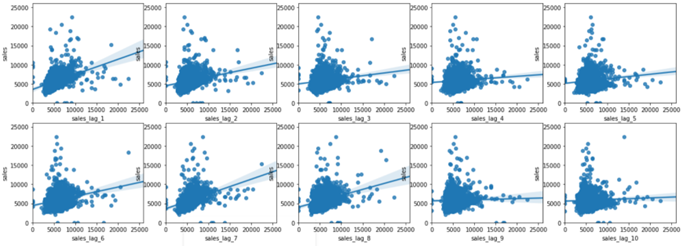
*Critical Values: 10%, -2.5*

**Observations:**

From the result, p-value is found significant which favors the alternate hypothesis that the series is stationary. However from the seasonal decomposition, it is evident that the series has seasonal components.

**SEASONALITY & LAGGED PLOTS**

From day wise plots across years, it is evident there is seasonality within sales. We can see some repetitive patterns of sales going up and down.

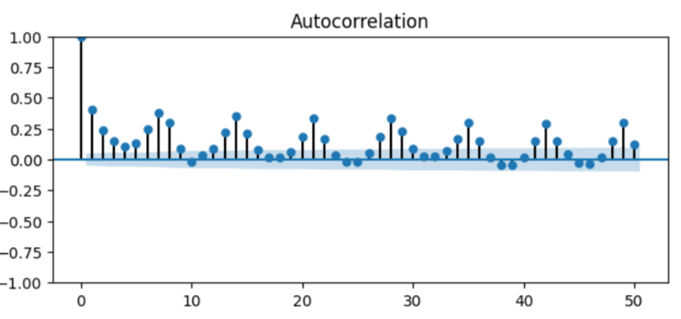
**Lagged plots**

**Observations:**

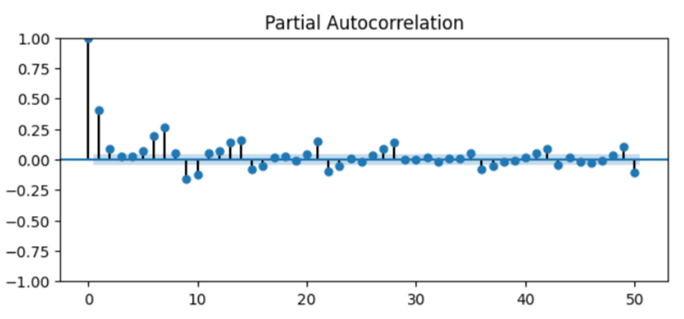
Here it can be seen correlation with different lags, especially the highest correlation of 0.39 was observed every consecutive day and once in every eight days. There are also days with no correlation for example no correlation is observed after every 5 and 10 days.

**ACF & PACF plots:**

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. We plot these values along with the confidence band. PACF is a partial auto-correlation function. Basically instead of finding correlations of present with lags like ACF, it finds correlation of the residuals with the next lag value hence ‘partial’ and not ‘complete’ as we remove already found variations before we find the next correlation.



ACF with significant values at lags at multiples of 7, implies there is seasonality every 7 days.

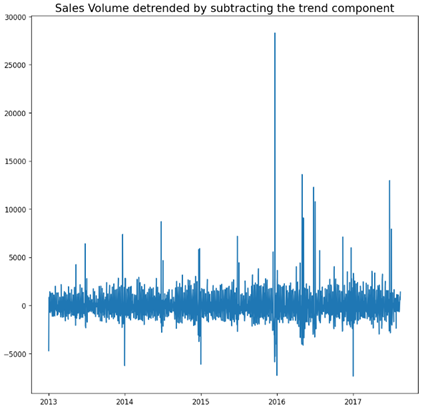


A negative value at lag period 9 which implies higher sales values 9 days ago will cause lower values today.

**DETREND & DESEASONALIZED time series**

A stationary process has the property that the mean, variance and autocorrelation structure do not change over time. Stationarity can be defined in precise mathematical terms, but for our purpose we mean a flat looking series, without trend, constant variance over time, a constant autocorrelation structure over time and no periodic fluctuations. To ensure stationarity we have to detrend and deseasonalize.

**DETREND**

****Fit a line and subtract series from the fitted line to cancel the trend

**DESEASONALIZE**

Take a moving average with length as the seasonal window. This will smoothen in series in the process. Seasonal difference in the series (subtract the value of previous season from the current value). Divide the series by the seasonal index obtained from STL decomposition



**CAUSALITY TEST:**

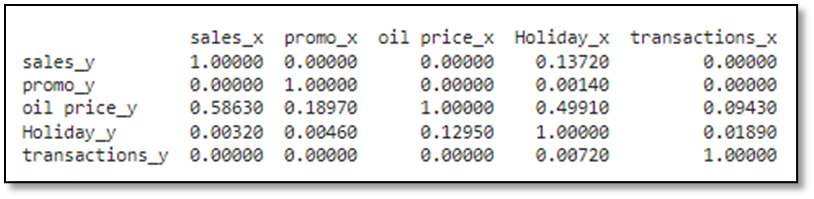
Since we have multiple predictors for example promotions, transactions we performed the Granger Causality test to determine whether a one-time series is useful for forecasting another.

Null Hypothesis (H0): Time series *x* does not Granger-cause time series *y*

Alternative Hypothesis (HA): Time series *x* Granger causes series *y*

“Granger-causes” means that knowing the value of time series *x* at a certain lag is useful for predicting the value of time series *y* at a later time period.

The row is the response (y) and the columns are the predictors (x). If a given p-value is < significance level (0.05), for example, take the value 0.0 in (row 1, column 2), we can reject the null hypothesis .

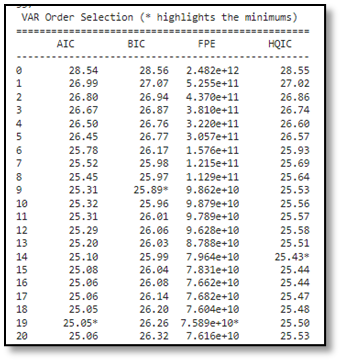
Holiday, promotions, transactions are shown significant

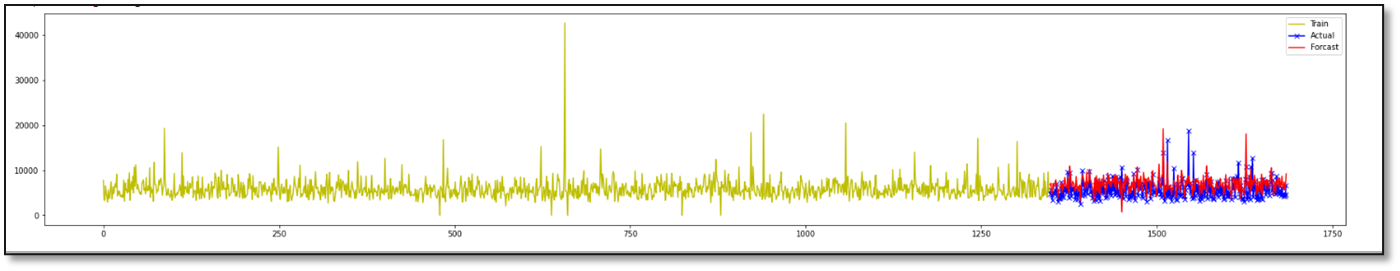
**VECTOR ARIMA**

Vector autoregression (VAR) is a statistical model used to capture the relationship between multiple quantities as they change over time. VAR is a type of stochastic process model. VAR models generalize the single-variable (univariate) autoregressive model by allowing for multivariate time series.

Like the autoregressive model, each variable has an equation modeling its evolution over time. This equation includes the variable's lagged (past) values, the lagged values of the other variables in the model, and an error term. VAR models do not require as much knowledge about the forces influencing a variable as do structural models with simultaneous equations. The only prior knowledge required is a list of variables which can be hypothesized to affect each other over time.

Results: The best model is chosen depending upon lowest AIC

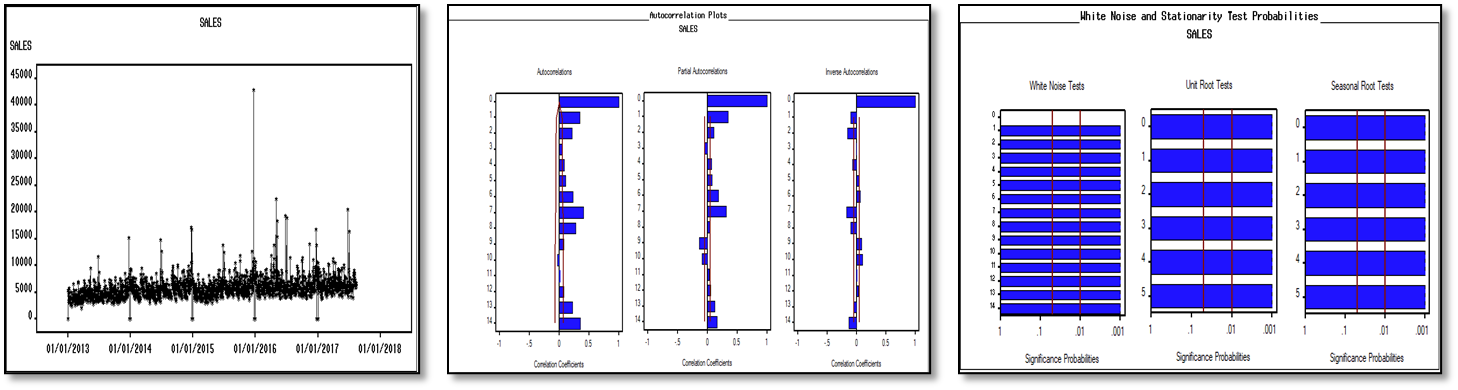


****

**This plot shows actual vs forecast in Vector ARIMA model.**

**5.2.2 ARIMA IN SAS**

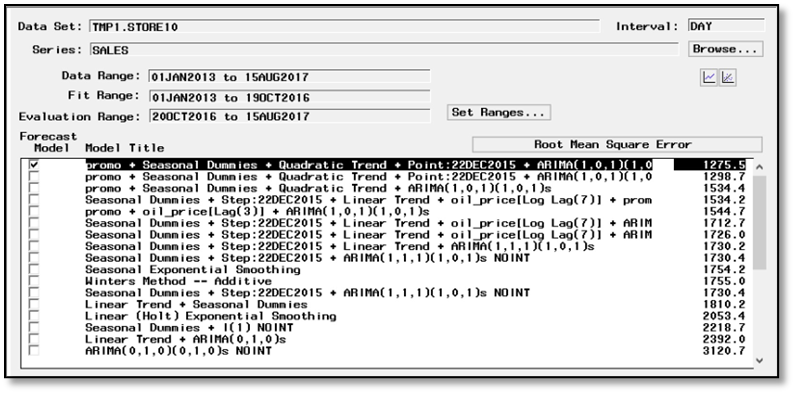
We can observe outputs after establishing the model in SAS, giving us insight into trends, seasonality, outliers, white noise, and autocorrelation. For our analysis, we have taken into account one of the stores i.e. store\_10. The original trend, seasonality and autocorrelation in the data can be seen as follows -



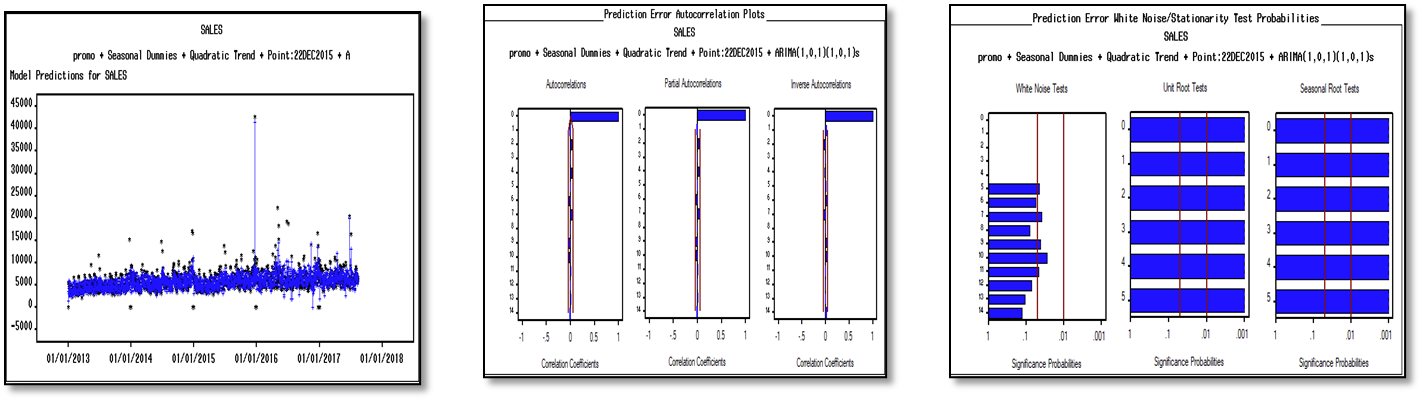
* Slight trend can be observed
* Data is seasonal
* A cycle happens every seven days
* Data is not white noise

After trying several models for achieving the best performance, we were able to tune the model with RMSE as low as 1275.5, MAPE: 0.127, and R^2: 0.663. Below is the information on parameters we used for building the best-performing model

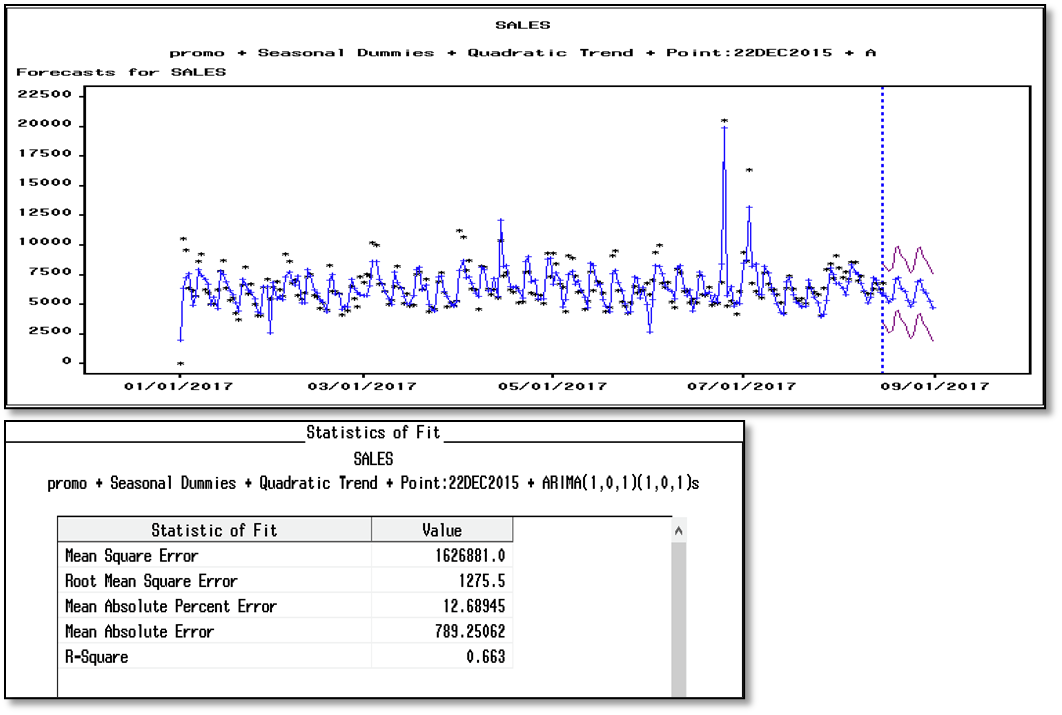
* Regressors - Promo
* Seasonal Dummies
* Quadratic Trend
* Interventions - Point: 22DEC2015
* ARIMA(1,0,1)(1,0,1)s



**Model Output -**



* No autocorrelation
* No partial autocorrelation
* Residual is white noise
* Actual vs Predicted

**Model Forecast -** Data forecasted for next 16 days i.e. 2.5 weeks

1. **STORE-WISE RESULTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Store No** | **Method** | **MAPE** | **RMSE** | **R-Square** |
| **1** | **Linear Regression** | 8.36 | 2188.8 | 0.61 |
|  | **Random Forest** | 1.77 | 1599.9 | 0.79 |
|  | **XGBoost** | 1.51 | 1564.2 | 0.8 |
|  | **ARIMA (parameters) SAS** | 8.194 | 1451.3 | 0.829 |
| **10** | **Linear Regression** | 0.15 | 1494.1 | 0.45 |
|  | **Random Forest** | 0.13 | 1280.1 | 0.6 |
|  | **XGBoost** | 0.12 | 1286.9 | 0.59 |
|  | **ARIMA (parameters) SAS** | 0.125 | 1275.5 | 0.663 |
| **11** | **Linear Regression** | 0.11 | 3715.7 | 0.55 |
|  | **Random Forest** | 0.08 | 3065.3 | 0.69 |
|  | **XGBoost** | 0.09 | 3286.5 | 0.64 |
|  | **ARIMA (parameters) SAS** | 9.6 | 3347 | 0.71 |
| **14** | **Linear Regression** | 1.75 | 1499.5 | 0.66 |
|  | **Random Forest** | 1.19 | 1311.9 | 0.74 |
|  | **XGBoost** | 4.07 | 1487.3 | 0.66 |
|  | **ARIMA (parameters) SAS** | 13.509 | 1512.1 | 0.63 |
| **3** | **Linear Regression** | 0.18 | 8329.6 | 0.54 |
|  | **Random Forest** | 0.13 | 6816.6 | 0.69 |
|  | **XGBoost** | 0.11 | 6169.3 | 0.75 |
|  | **ARIMA (parameters) SAS** | 9.24 | 7112.2 | 0.669 |

1. **MODEL COMPARISON:**

**STORE 10**

|  |  |  |  |
| --- | --- | --- | --- |
| **Models** | **RMSE** | **MAPE** | **R2** |
| **Linear Regression** | 1494.1 | 0.15 | 0.45 |
| **Random Forest** | 1280.1 | 0.13 | 0.6 |
| **XGBoost** | 1286.9 | 0.12 | 0.59 |
| **Time Series in Python** | 1770 | 0.32 | 0.40 |
| **Time Series in SaS** | 1275.5 | 0.127 | 0.663 |

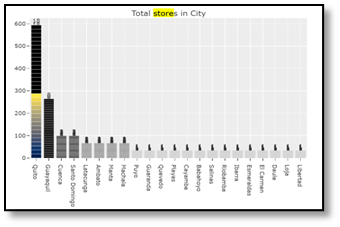
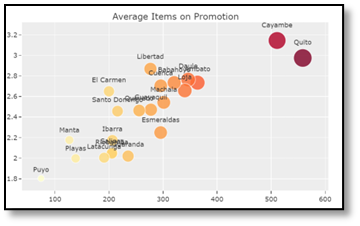
1. **RECOMMENDATIONS**

**8.1 Optimizing Shelf Space**

According to our data, the least sold products are those from the families of Hardware, Clothing, and Beauty. We advise giving shelf space for these products in accordance with the volume of sales. The shelf space should be assigned in such a way that the best-selling products, such as Groceries and Beverages, are placed at the back of the store. Customers would travel inside the store to purchase these items because they are the most popular, and the least sold items, such as hardware and clothing, should be placed on shelves and aisles that can attract customers' attention (near the store's entrance) so that sales volume for these items can be increased.

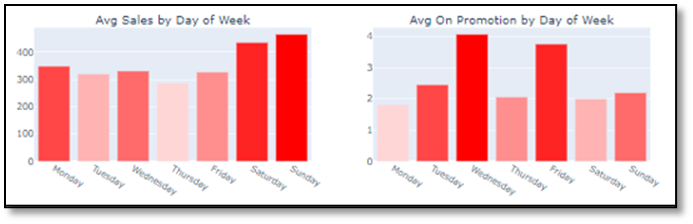
**8.2 Leveraging Number of Stores**

Despite having the second highest number of stores, the average sales for Guayaquil are lower than the average sales for Cayambe, Ambato, and Daule. We discovered that this was due to a greater number of items on promotion for the cities of Cayambe, Ambato, and Daule. We would suggest stores in Guayaquil to increase promotions in order to leverage their store count within the city for increased sales volume.



**8.3 Thrifty Thursdays**

Thursdays have the lowest average daily sales. The reason for this is that there are fewer promotions on Thursdays. We recommend that stores launch a campaign such as "Thrifty Thursdays" to attract customers, where the number of items on promotion across all product families is highest on Thursdays, resulting in an increase in sales volume.



**8.4 Reasonable Work Schedule**

Revamping the staffing schedule is one method to cut labor costs. For instance, compared to weekends, weekdays have lower sales volume. When opposed to weekends, we would recommend fewer active counters during the weekdays, which would result in lower overall labor costs.

**8.5 Customer Retention**

As oil prices rose, we noticed a decline in sales volume (due to an increase in logistics costs). During such inflationary periods, we would advise the stores to introduce membership-based loyalty or rewards programs, as this would encourage customers to make more repeat purchases from the same stores. Implementing such programs can significantly help stores to retain customers & maintain sales volume.

1. **CONCLUSIONS & FUTURE WORK**

* When comparing results at the store level, our best model had the lowest RMSE when compared to most of the Kaggle submissions.
* With enhanced computational power, we can explore advanced models such as Recurrent Neural Networks (RNN) and Seasonal Artificial Neural Networks (Seasonal-ANN) to improve forecasting accuracy.

1. **REFERENCES**

1.[https://www.sciencedirect.com/science/article/pii/S2667096822000027#bib0027](https://www.sciencedirect.com/science/article/pii/S2667096822000027)

2.[https://www.sciencedirect.com/science/article/pii/S2667096822000027#bib0022](https://www.sciencedirect.com/science/article/pii/S2667096822000027)

3.[https://www.sciencedirect.com/science/article/pii/S2667096822000027#bib0073](https://www.sciencedirect.com/science/article/pii/S2667096822000027)

4.[https://www.sciencedirect.com/science/article/pii/S2667096822000027#bib0066](https://www.sciencedirect.com/science/article/pii/S2667096822000027)

5.<https://blog.hubspot.com/service/retain-customers-in-inflation>

6.<https://www.sciencedirect.com/science/article/pii/S2667096822000027>

7.<http://cs229.stanford.edu/proj2017/final-reports/5244336.pdf>

8.<https://www.researchgate.net/publication/350935708_Time_series_forecast_of_sales_volume_based_>

9.<https://towardsdatascience.com/random-forest-regression-5f605132d19d>

10.<https://www.geeksforgeeks.org/xgboost-for-regression/>

11. <https://en.wikipedia.org/wiki/Linear_regression>

1. **RESPONSIBILITIES OF EACH TEAM MEMBER**

* **Adwaith Ajith Menon**

Adwaith took complete responsibility in implementing predictive models at all the store levels. He also contributed to the recommendations, introduction sections in the final report.

* **Di Zhao**

Di Zhao took complete responsibility in implementing the Time Series Models in SAS, she also actively contributed in documentation of the final report and also assisted in preparing slides.

* **Jahnavi Adabala**

Jahnavi took complete responsibility in gathering up relevant literature for our problem. She implemented prediction models for entire panel level data and also contributed in documentation

* **Kartiki Bhargava**

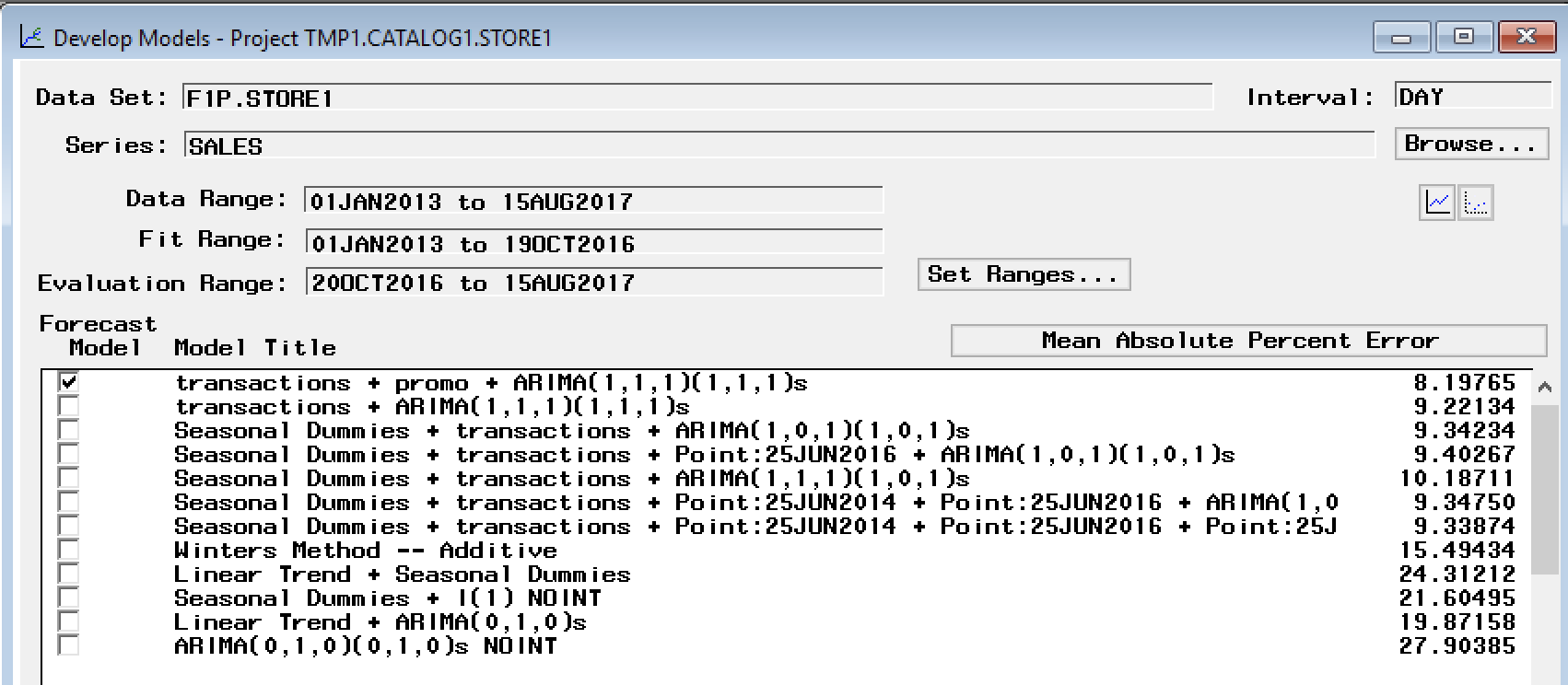
Kartiki took part in implementation of Time Series Models in SAS and also collaborated in documentation and presentation

* **Sai Jahnavi Gamalapati**

Sai Jahnavi performed Exploratory Data Analysis and also explored time series analysis in python. She also assisted in the preparation of the presentation and final report.

1. **APPENDIX**

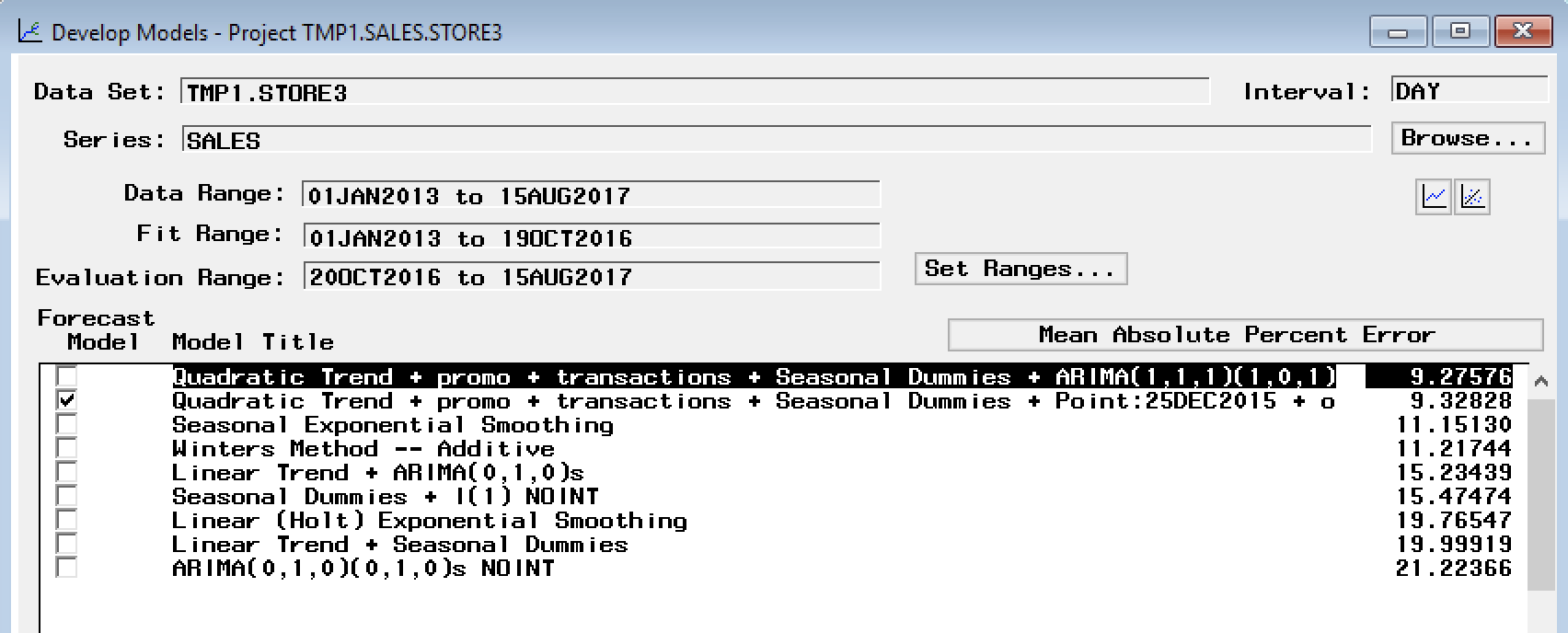
**Store\_1**



**Best Model:** transactions + promo + ARIMA(1,1,1)(1,1,1)s

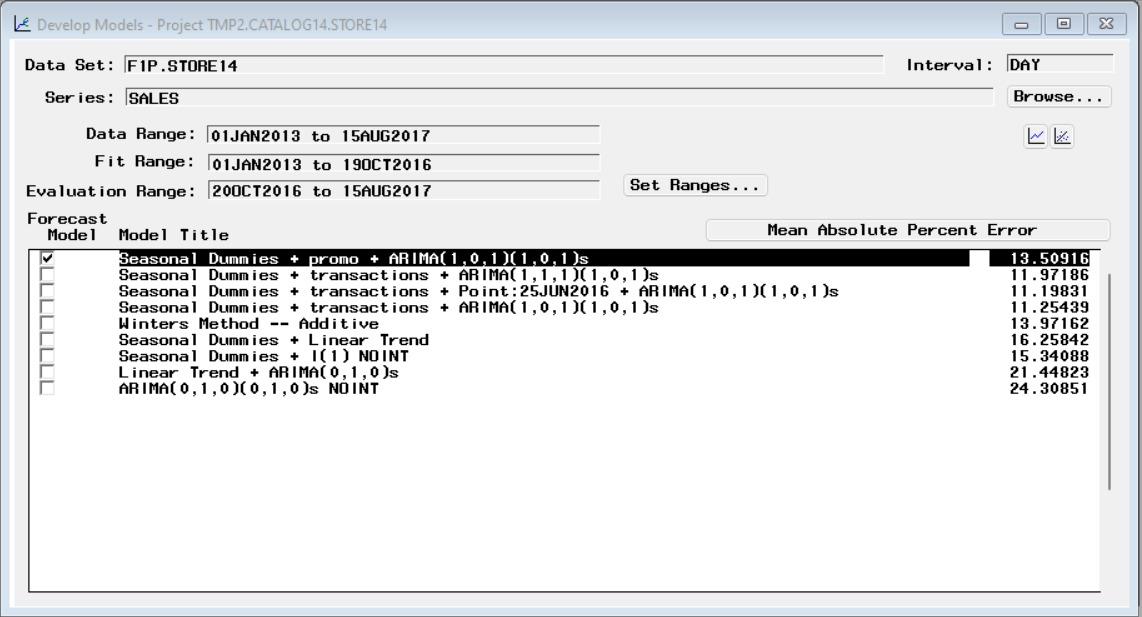
**MAPE:** 8.198

**Store\_3**



**Best Model:** Quadratic Trend + promo + transactions + Seasonal Dummies + ARIMA(1,1,1)(1,0,1)

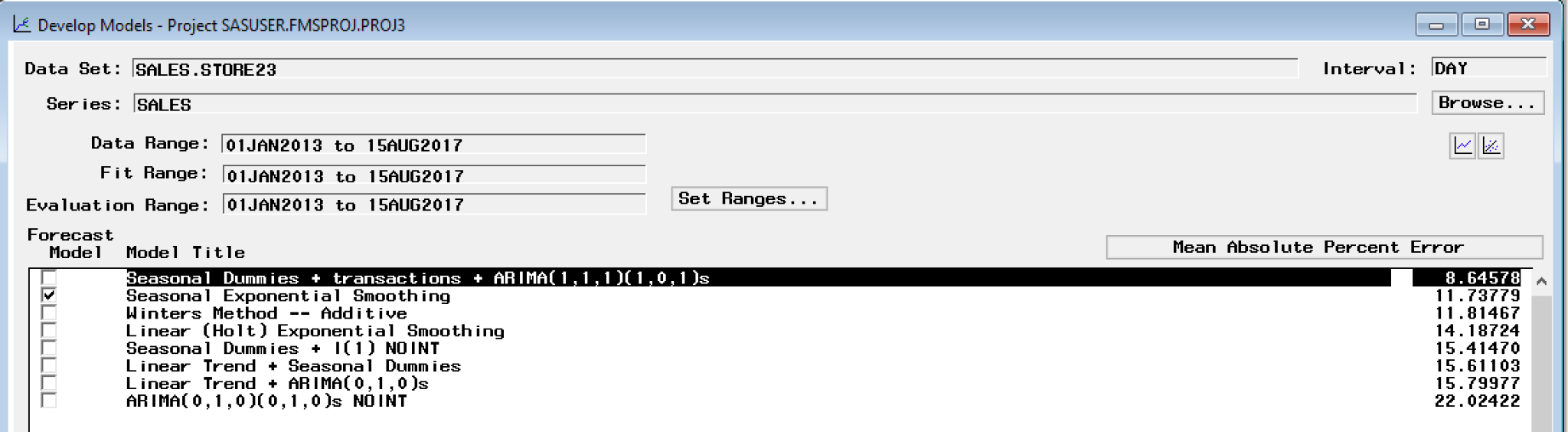
**MAPE:** 9.276

**Store\_14**

**Best Model:** Seasonal Dummies + transactions + ARIMA(1,0,1)(1,0,1)s + Point:25JUN2016

**MAPE:** 11.20

**Store\_23**



**Best Model:** Seasonal Dummies + transactions + ARIMA(1,1,1)(1,0,1)s

**MAPE:** 8.646