Improving Demand Forecasting

Final Report, Capstone Project 2

Table of Contents

1. Problem Statement

What is the seasonality of products in a store? How does demand compare across different stores for the same item? What about seasonally? What will the demand for a product be in the next few months? Accurate forecasting of sales and demand is an important part to managing the supply chain for both online and physical retail.

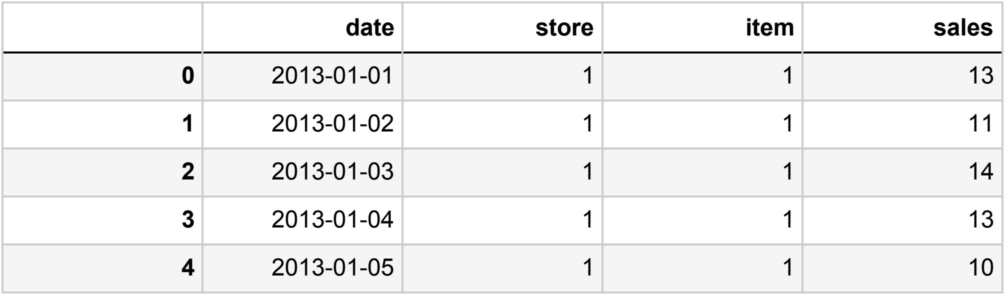
This time series study analyzes the sales of 50 different items at 10 stores across 5 years using algorithms from ARIMA, SARIMA and Facebook Prophet. In addition to tuning the parameters in the modeling functions, each model improves upon the predictions made on item sales.

2. Exploratory Data Analysis

The dataset comes from Kaggle's store item demand forecasting challenge. It describes the sales numbers of 50 items across 10 stores and comes in 2 csv files, train.csv and test.csv.

2.1 Overall Sales

The training dataset contains the date, store number, item number, and sales number for each day of 2013 through 2017, totaling 913,000 records. The dataset contains no details about the items or the stores. This analysis assumes that the stores are from the US.



	date	store	item	sales
0	2013-01-01	1	1	13
1	2013-01-02	1	1	11
2	2013-01-03	1	1	14
3	2013-01-04	1	1	13
4	2013-01-05	1	1	10

Daily sales numbers range from 0 to 231. On average, 52 of any particular item are sold per day. The majority, or middle 50%, of daily item sales numbers fall between 30 and 70, and there are the least number of observations above 100. Density of the sales observations shows the data has non-normal distribution and is slightly left shifted:

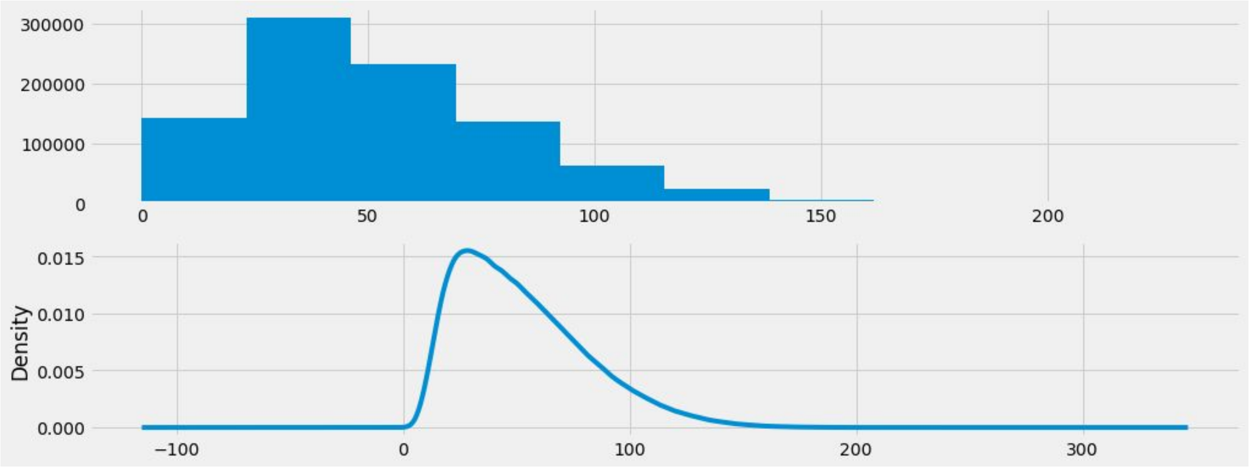


Figure 1.1 *Distribution and density of monthly total sales numbers from 2013 through 2017*

Median sales values have an upward trend over time. Range of total annual sales increased slightly each year, and there is also an increase in the spread, or middle 50% of the data, over time.

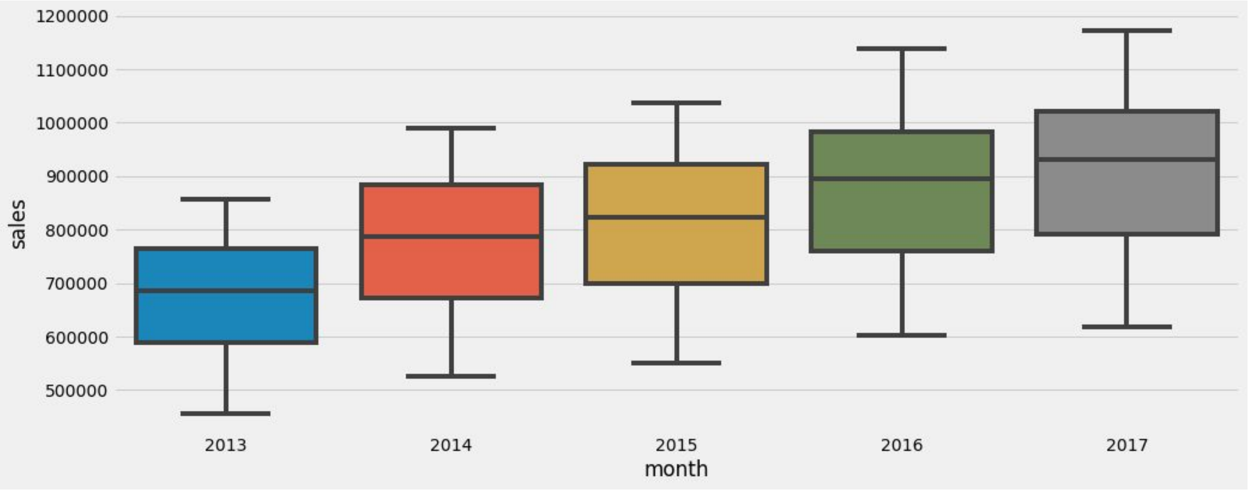


Figure 1.2 *Median and spread of item sales at all stores from 2013 through 2017*

Each year, total item sales climbs to a peak in July from January, and then goes back down to a trough with a slight peak in November. Reviewing the total sales numbers over time for all items in all stores reveals a time series with yearly seasonality and an upwards trend:

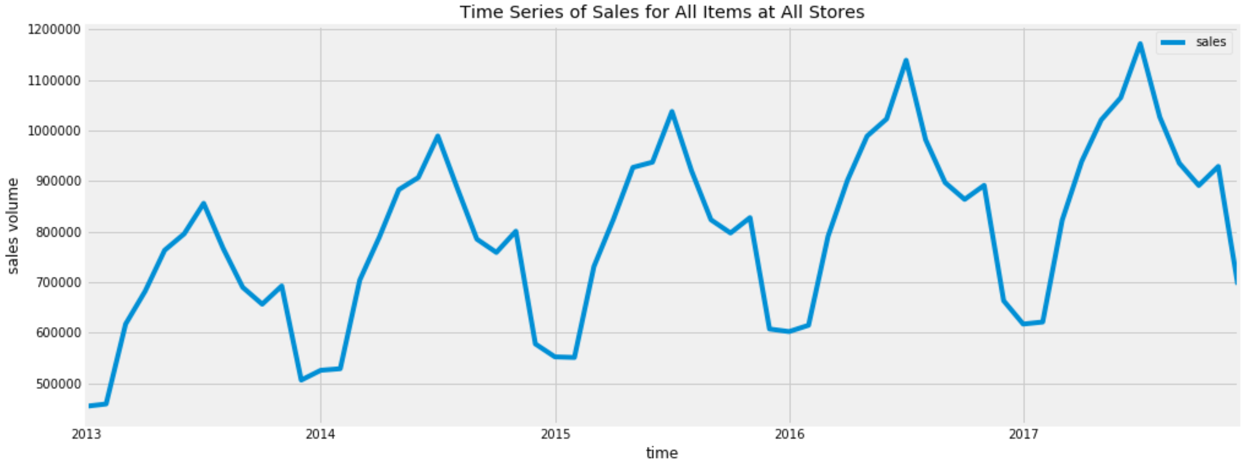


Figure 1.3 *Time Series of sales totals for all items at all stores from 2013 through 2017*

2.2 Individual Sales

For item 1 at store 1, daily sales numbers range from 4 to 50 with a mean of 20. The majority, or middle 50%, of daily sales numbers fall between 15 and 24, with the least number of observations below 8 and above 35. Unlike overall sales, density of individual item sales observations at one store shows the data is normally distributed:

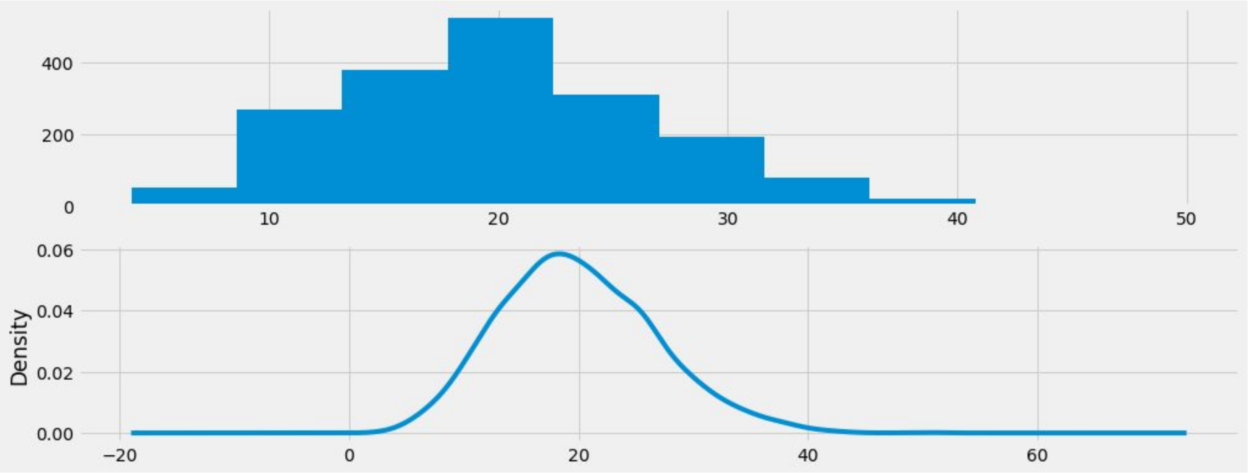


Figure 1.3 *Distribution and density of item 1 sales numbers at store 1 from 2013 through 2017*

Median values of item 1 sales increased from 2013 to 2016 with a slight decrease in 2017. While the range of total annual item 1 sales did not increase steadily over time, the spread or middle 50% of the data increased steadily over time:

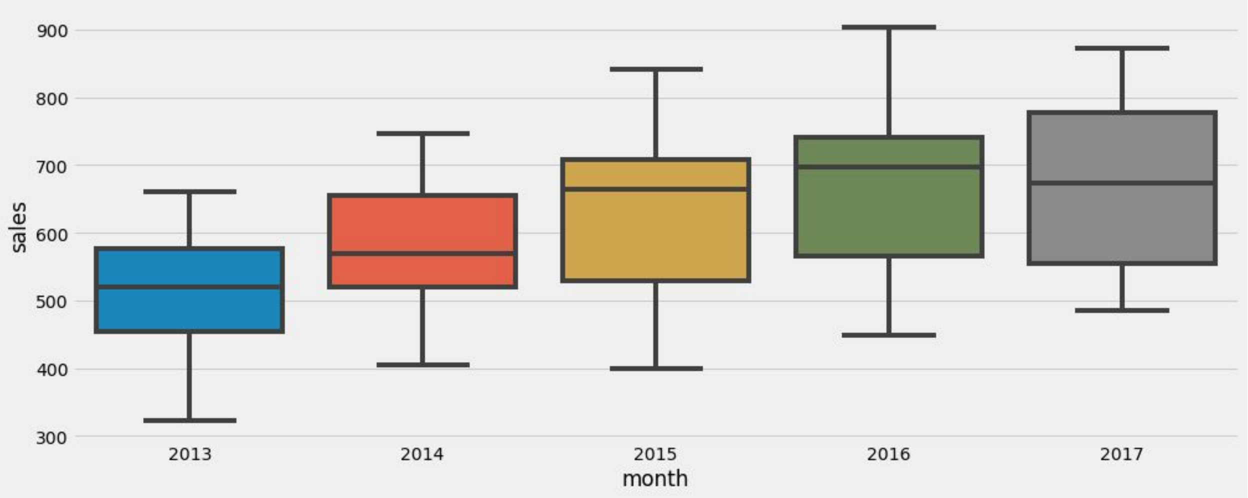


Figure 1.4 *Median and spread of item 1 sales at store 1 from 2013 through 2017*

As expected for both item 1 and item 2 at store 1, the peaks and troughs are more irregular throughout the years compared to those of overall sales. The trends and seasonality of the time series are less pronounced.

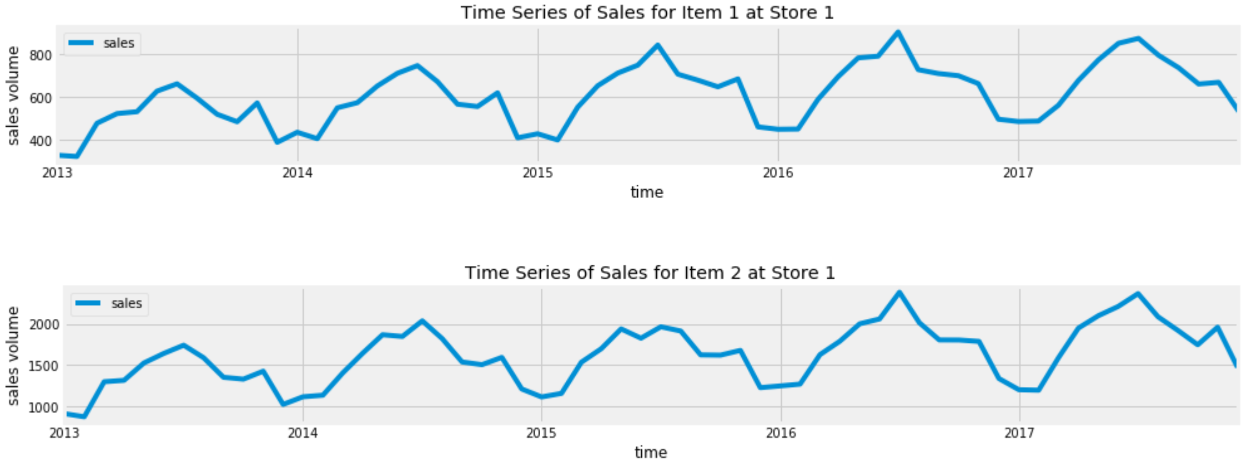


Figure 1.5 *Time series of item 1 and 2 sales at store 1 from 2013 through 2017*

Decomposition of item 1 sales at store 1 confirms an upward trend and strong seasonality from year to year with peaks occurring in July of each year:

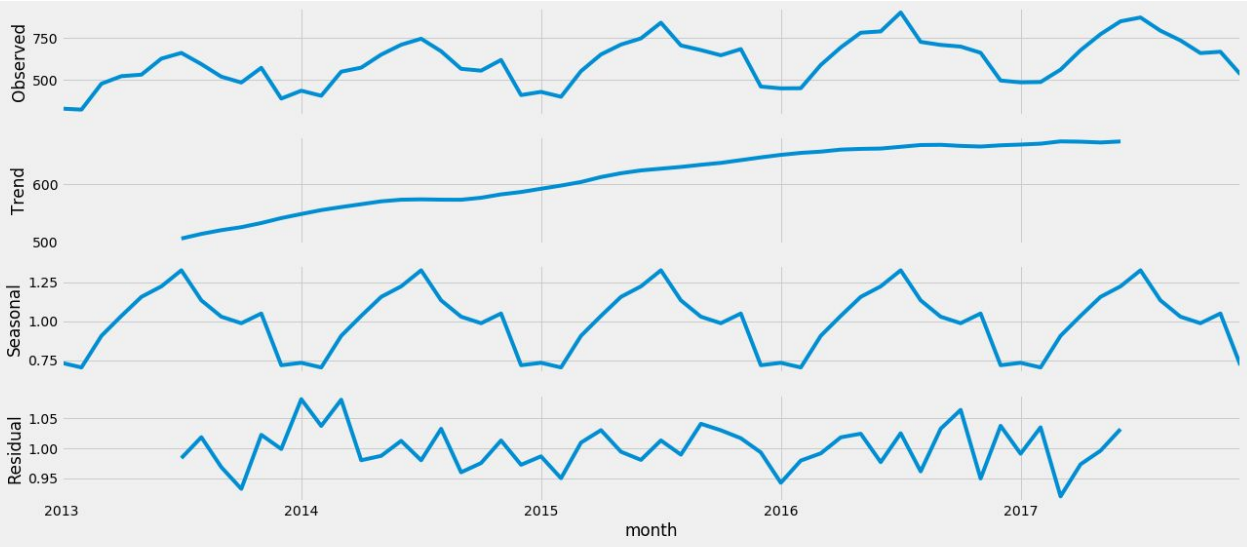
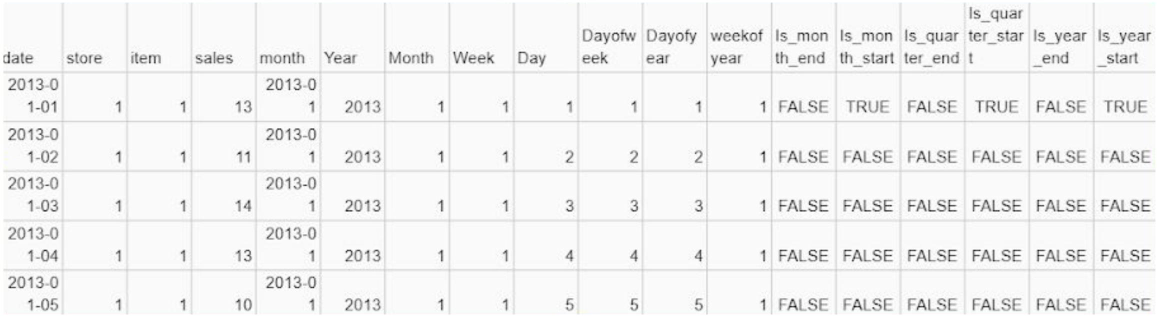


Figure 1.6 *Decomposition of item 1 sales at store 1 from 2013 through 2017*

3. Modeling Preparations

3.1 Extending Date Features

Combining test and training data, the dataset is extended with date features that describe each day with the week, day of year, month, etc. Figuring out how these features impact item sales is a step in preparation for fine tuning the time series models.



date	store	item	sales	month	Year	Month	Week	Day	Dayofweek	Dayofyear	weekofyear	Is_month_end	Is_month_start	Is_quarter_end	Is_quarter_start	Is_year_end	Is_year_start
2013-01-01	1	1	13	2013-01	2013	1	1	1	1	1	1	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
2013-01-02	1	1	11	2013-01	2013	1	1	2	2	2	1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2013-01-03	1	1	14	2013-01	2013	1	1	3	3	3	1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2013-01-04	1	1	13	2013-01	2013	1	1	4	4	4	1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2013-01-05	1	1	10	2013-01	2013	1	1	5	5	5	1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Box cox transformation is applied on the sales values to obtain normal distribution, which is required for training with the LightGBM framework. True and False values in the data are converted to 1 and 0 before training. Tree based learning algorithms from the LightGBM framework are used to determine feature importances.

Plotting the resulting feature importances shows the week, item, and day as the top 3 most important features having the biggest impact on sales numbers. Conversely, the store, day of year, and month are among the least important features:

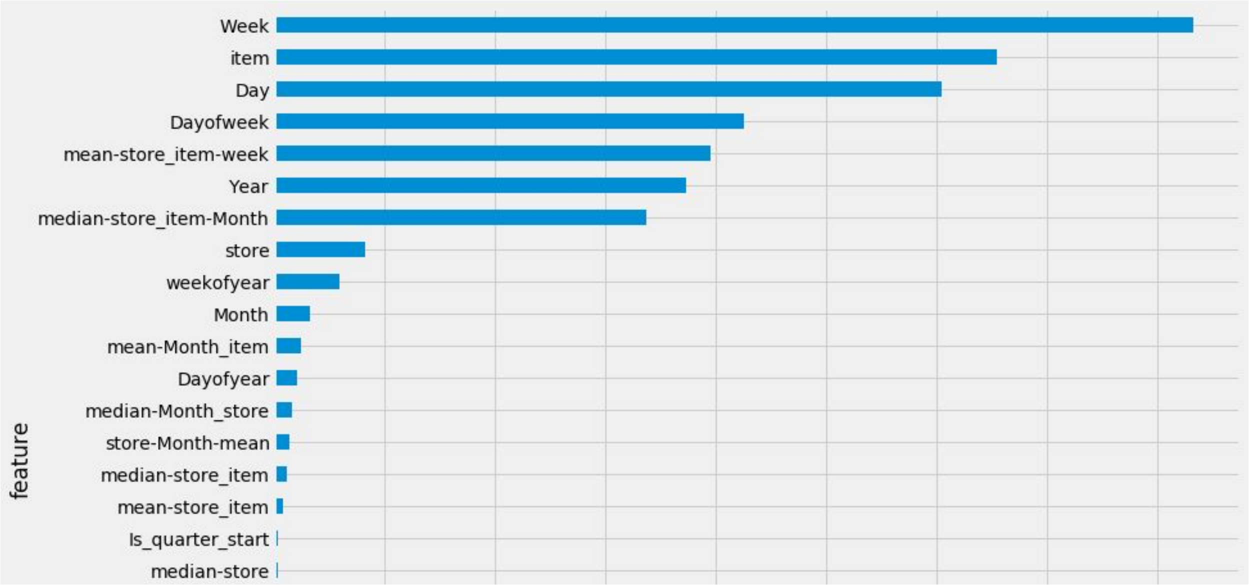


Figure 2.1 *Date features and their importances*

3.2 Data Transformation

In order to model a time series, it must be stationary. Most non-stationary series can be transformed into stationary series. Most time series models work with stationary time series because non-stationarity increases unpredictability. Newer models like Facebook Prophet provide higher flexibility in dealing with non-stationary time series. Stationary means that the distribution of the data doesn't change with time. For a time series to be stationary, it must fulfill 3 criteria:

- Zero trend: the series does not grow or shrink over time
- Constant variance: average distance of the data points from the zero line does not change
- Constant autocorrelation: how each value in the series is related to its neighbors stays the same

Identifying whether a time series is stationary or non-stationary is very important. If it is stationary, then we can use ARMA (autoregressive moving average) models to predict the next values of the time series. If it is non-stationary, then models like Facebook Prophet should be used.

Stationarity of an individual time series (for item 1 at store 1) is checked below using the augmented Dickey-Fuller Test. Here the null hypothesis is that the time series is non-stationary. The test results comprise of a Test Statistic and some Critical Values at different confidence levels. If the 'Test Statistic' is less than the 'Critical Value', then the null hypothesis is rejected and the time series is stationary.

The results for the time series of item 1 at store 1 show the test statistic is -3.16, which is less than the 5% critical value but greater than the 1% critical value, and the p-value 0.02 is greater than 0.05. Since the series has a strong upward trend, the strictest 1% critical value is used to test the hypothesis. Therefore, the null hypothesis is accepted and the time series of item 1 at store 1 is non-stationary. Transformations of individual time series are necessary prior to modeling using ARMA models.

Results of Dickey-Fuller Test	**Before Transformation**	**After Transformation**
Test Statistic	-3.157671	-1.267679e+01
p-value	0.022569	1.210928e-23
#Lags Used	23.000000	2.200000e+01
Number of Observations Used	1802	1802
Critical Value (1%)	-3.433984	-3.433984e+00
Critical Value (5%)	-2.863145	-2.863145e+00
Critical Value (10%)	-2.567625	-2.567625e+00

Differencing is the technique used to transform the individual time series for stationarity. In this technique, the difference of the observation at a particular instant with that at the previous instant is taken. After first order differencing using Pandas, the test statistic is less than the 1% critical value and the p-value is much smaller than 0.05. Now the null hypothesis is rejected and the transformed time series is stationary. The series also shows constant variance over time upon plotting its rolling statistics:

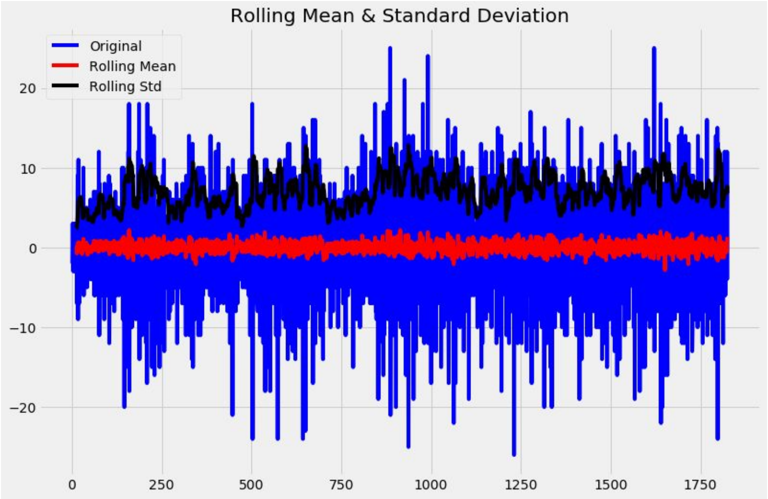


Figure 2.2 *Rolling statistics of item 1 sales at store 1*

4. Modeling with ARIMA and SARIMA

ARIMA model includes the AR (Auto Regression) term, the I (Integrated) term, and the MA (Moving Average) term. The I term is a full difference derived by subtracting one instant's value from another instant's value. The AR term is a partial difference. The coefficient on the AR term explains the percent of a difference needed to be taken. A MA term in a time series model is a past error (multiplied by a coefficient).

The 3 terms (AR, I, and MA) of the ARIMA model correspond to 3 parameters (p, d, and q) in the modeling function. These parameters help model the major aspects of a times series: seasonality, trend, and noise.

Parameter p is associated with the AR aspect of the model, which incorporates past values i.e lags of dependent variable. If p is 5, the predictors for x(t) will be x(t-1)….x(t-5). Evaluating the PACF plot can help specify the value for p.

Parameter d is associated with the I term of the model, which affects the amount of differencing to apply to a time series. The transformation for trend stationarity that was done on this time series prior to modeling was a first order differencing, which means that in this case d is equal to 1.

Parameter q is the size of the MA window of the model. If q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average and actual value at ith instant.

4.1 Determining p, d, q for ARIMA

ACF and PACF plots provide diagnostics that help determine the important parameters in the ARIMA modeling function. According to the rules for identifying parameters for ARIMA models:

1. ACF becomes zero after lag = 3, so q= 3
2. PACF becomes almost zero after lag = 6, so p = 6
3. d is the order of differencing, so d=1 if original time series' is used for model fitting, d=0 if transformed time series is used for model fitting.

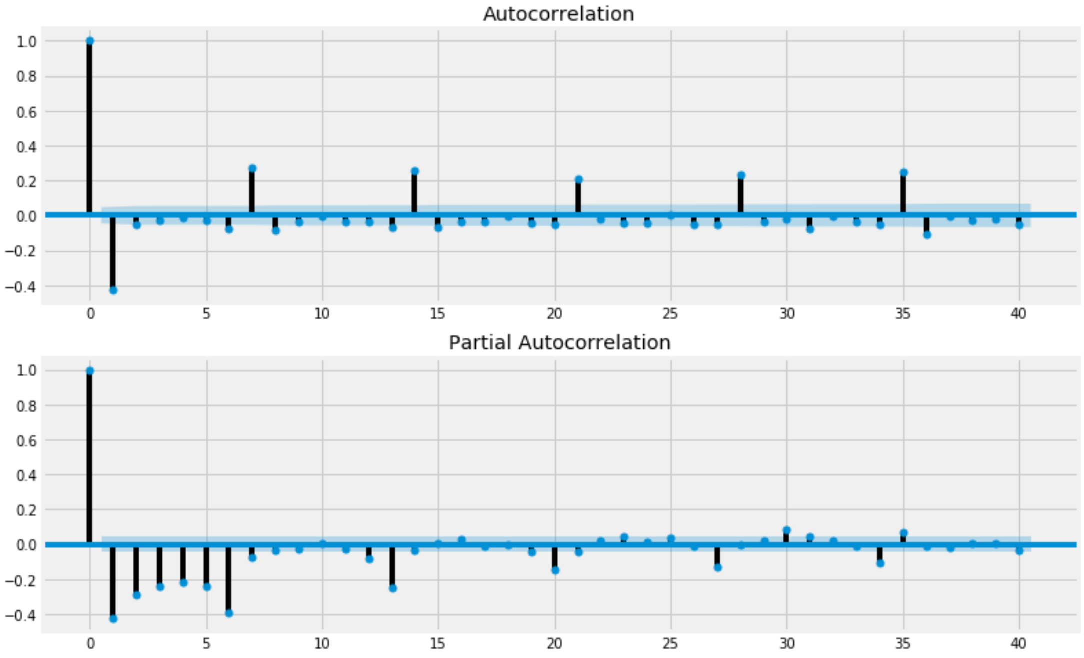


Figure 3.1 *ACF and PACF of transformed time series of item 1 at store 1*

Fitting an ARIMA model with order = (6,0,3) and plotting its residuals shows that the model’s residuals are not random. Most of the residuals are not of constant location and scale:

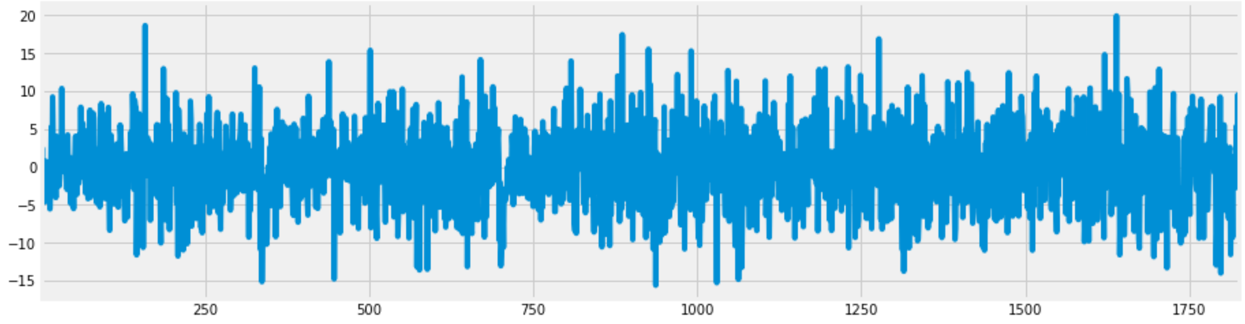


Figure 3.2 *Residuals of fitted ARIMA model using p = 6, d = 0, and q = 3*

A diagnosis of the residuals can be done using the ACF and PACF plots as well as the LJung-Box test. The ACF and PACF plots show that for the first 50 lags, not all sample autocorrelations fall inside the 95% confidence interval, which indicates that the residuals are not random:

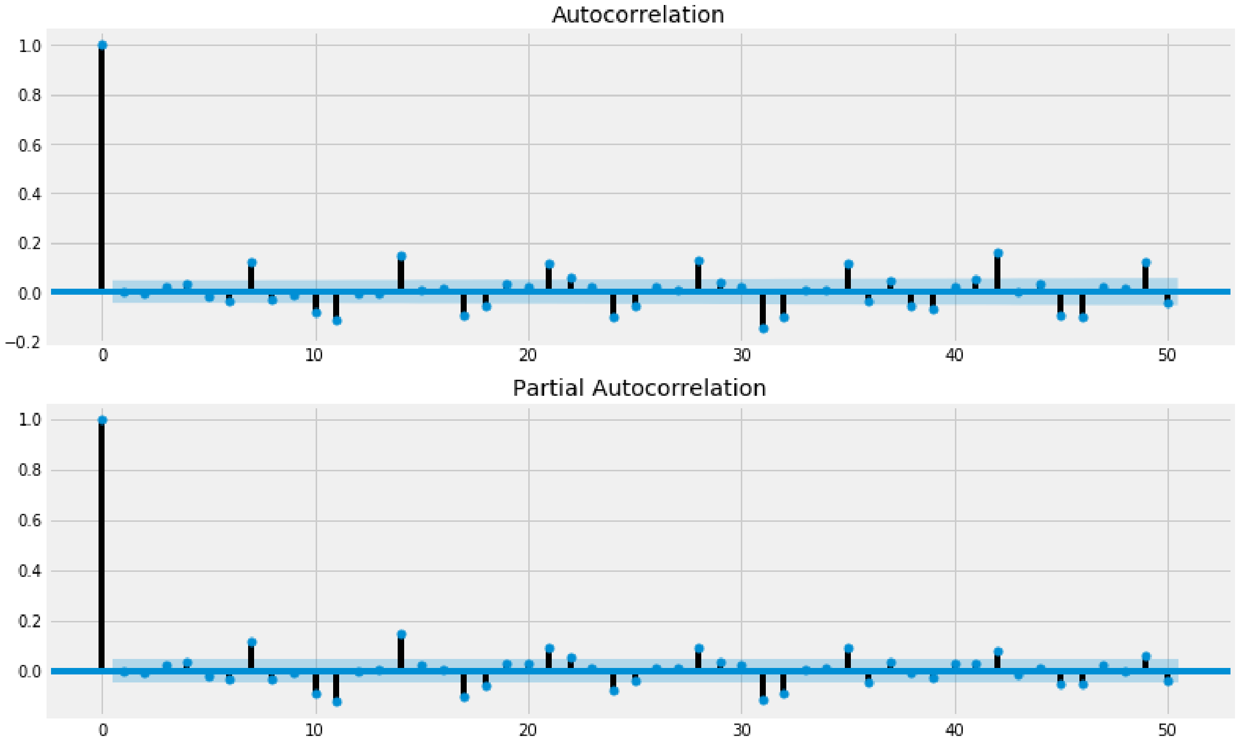


Figure 3.3 *ACF and PACF of ARIMA(6,0,3) residuals*

The LJung-Box test result shows that for more than 45 of the first 50 lags, the Ljung-Box statistics are 0 which is lower than the 0.05 threshold, indicating that the residuals are not random:

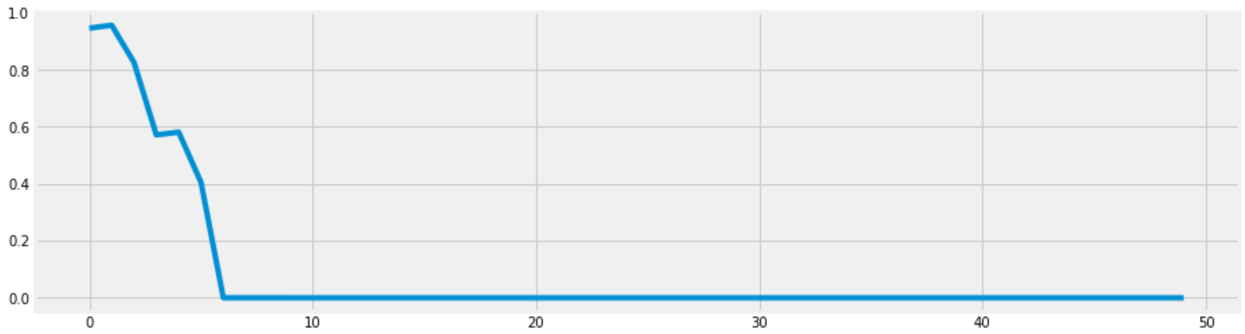


Figure 3.4 *LJung-Box test statistics of ARIMA(6,0,3) residuals*

Diagnostics of the residuals of the ARIMA(6,0,3) fitted model on the transformed time series of item 1 at store 1 indicate that the ARIMA model does not provide an adequate fit to the data.

The ACF and PACF plots of the transformed time series both show recurring patterns every 7 periods, which indicates a weekly pattern and significant seasonality even after differencing transformation for stationarity. Next, the SARIMAX model is considered for dealing with seasonality.

4.2 Forecasting with SARIMA

There are four seasonal elements that are part of the SARIMA modeling function that can be configured:

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period. If m is 7 for daily data, then there is weekly seasonality.

Previously the ACF and PACF plots showed strong seasonality on a weekly cycle, so m equals 7 is used in the seasonal order of the SARIMA model.

Using the SARIMA model to take seasonality into account, ACF and PACF plots of the fitted model's residual show that most sample autocorrelations fall within the 95% confidence interval, which indicates that the residuals are random:

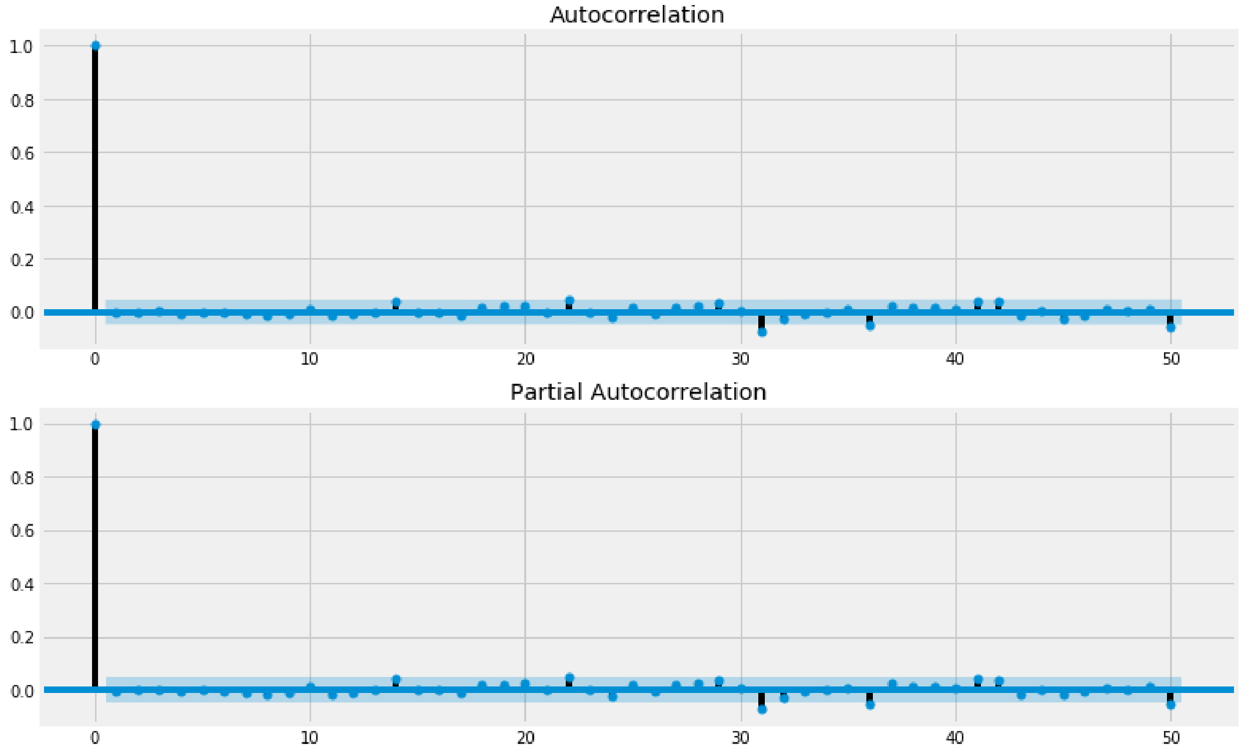


Figure 3.5 *ACF and PACF of SARIMA(6,1,3,1,1,1,7) residuals*

The Ljung-Box test result shows that for all observations in the first 50 lags, the Ljung-Box statistics are above 0.5 which is higher than the 0.05 threshold, indicating that the residuals are random.

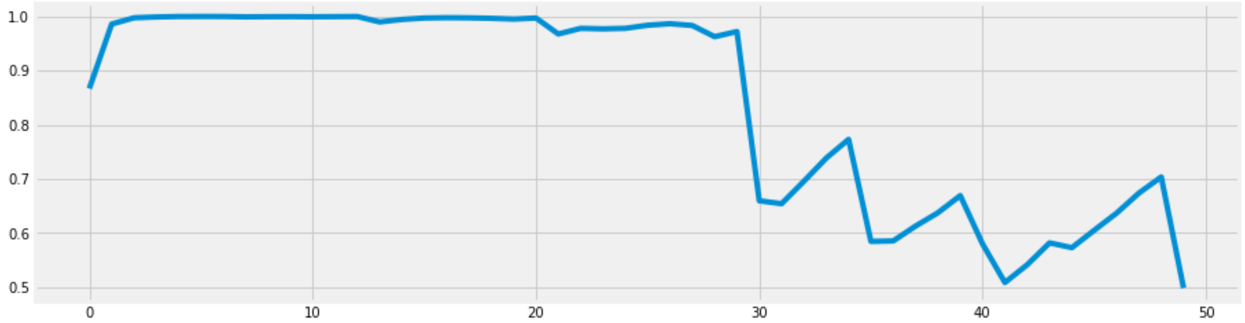


Figure 3.6 *Ljung-Box test statistics of SARIMA(6,1,3,1,1,1,7) residuals*

Diagnostics of the residuals of the SARIMAX(6,1,3,1,1,1,7) fitted model on the transformed time series of item 1 at store 1 indicate that the model provides a more adequate fit to the data than the ARIMA model. This model is used for forecasting.

Forecasting 90 days into the future for the sales of item 1 at store 1 using the fitted SARIMAX model above, the forecast appears to be quite inaccurate and the model was not able to take much of the seasonality into account despite SARIMA being a better fit for the data than ARIMA.

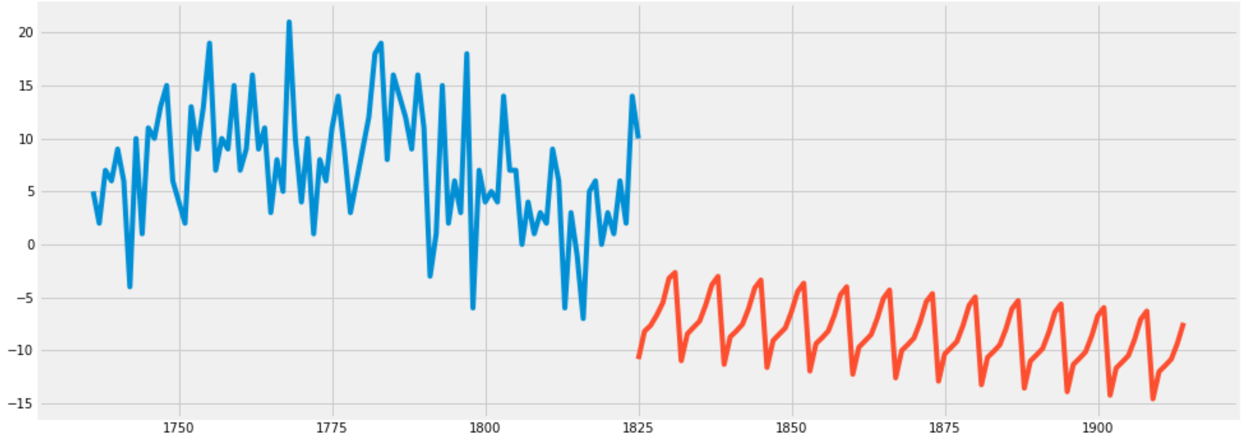


Figure 3.7 *Forecasting 90 days into the future with SARIMA(6,1,3,1,1,1,7)*

5. Modeling with Facebook Prophet

Facebook's time series modeling package Prophet uses a trend-seasonality decomposition. It also provides more flexibility in that data with high seasonality need not be transformed prior to modeling.

5.1 Forecasting with and without Holidays

To prepare the dataset for modeling with Prophet, the column with sales values is copied into a new column named y, and the column with dates is copied into a new column named ds (for datestamp). The dataset is split into training and validation sets to prepare for making predictions.

Prophet’s predictions shown in dark blue in the below plot demonstrates that the model is a better fit for the data than SARIMA and it takes into account most of the seasonality and anomalies:

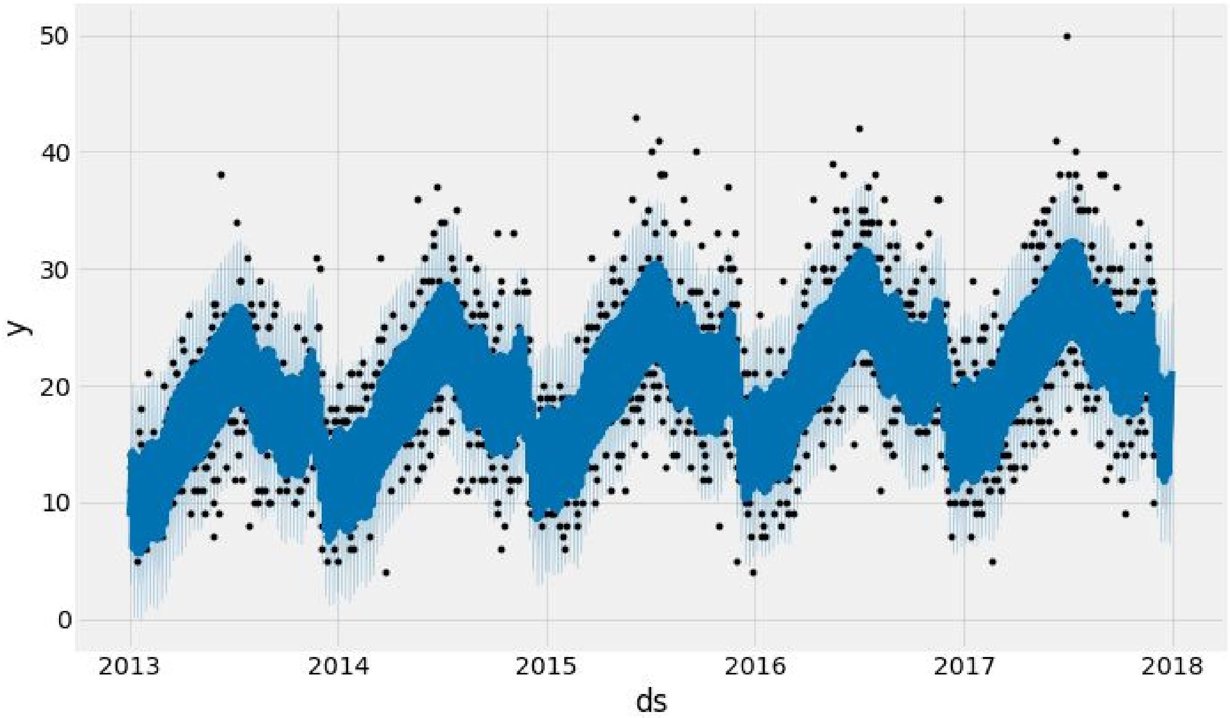


Figure 4.1 *Predictions of sales of item 1 at store 1 using Prophet*

Based on Prophet's decomposition of seasonality, the forecast shows that sales of item 1 at store 1 peak weekly on Saturday and Sunday, and yearly in July and in November:

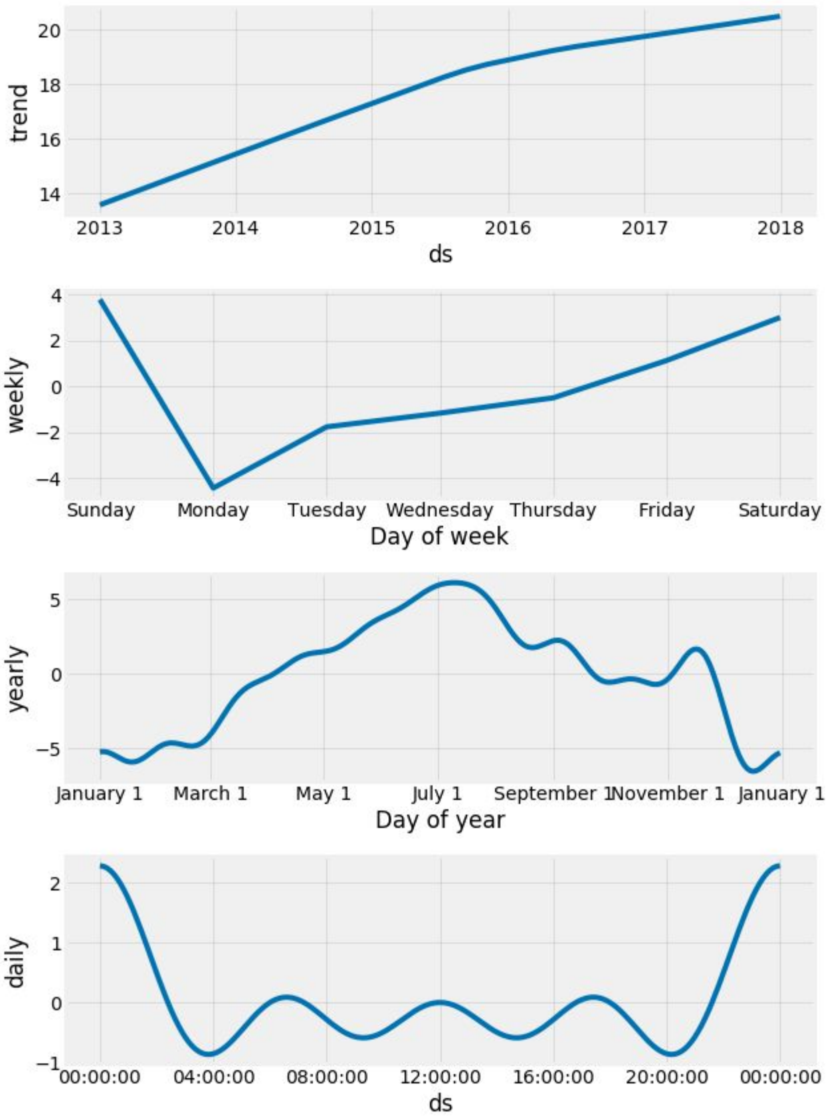


Figure 4.2 *Components of Prophet's forecast on the sales of item 1 at store 1*

Prophet's decompositions suggest that annual events and holidays in July and November could be taken into consideration when making predictions for this sales data. Assuming that the sales data is for stores based in the US, Independence Day, the play offs, the Super Bowl, and Thanksgiving should be added as holidays to improve the forecast.

A dataframe of the dates and corresponding holidays is created and passed into the holiday parameter in the Prophet modeling function. Prophet model's components now include a decomposition of the holidays that were passed in:

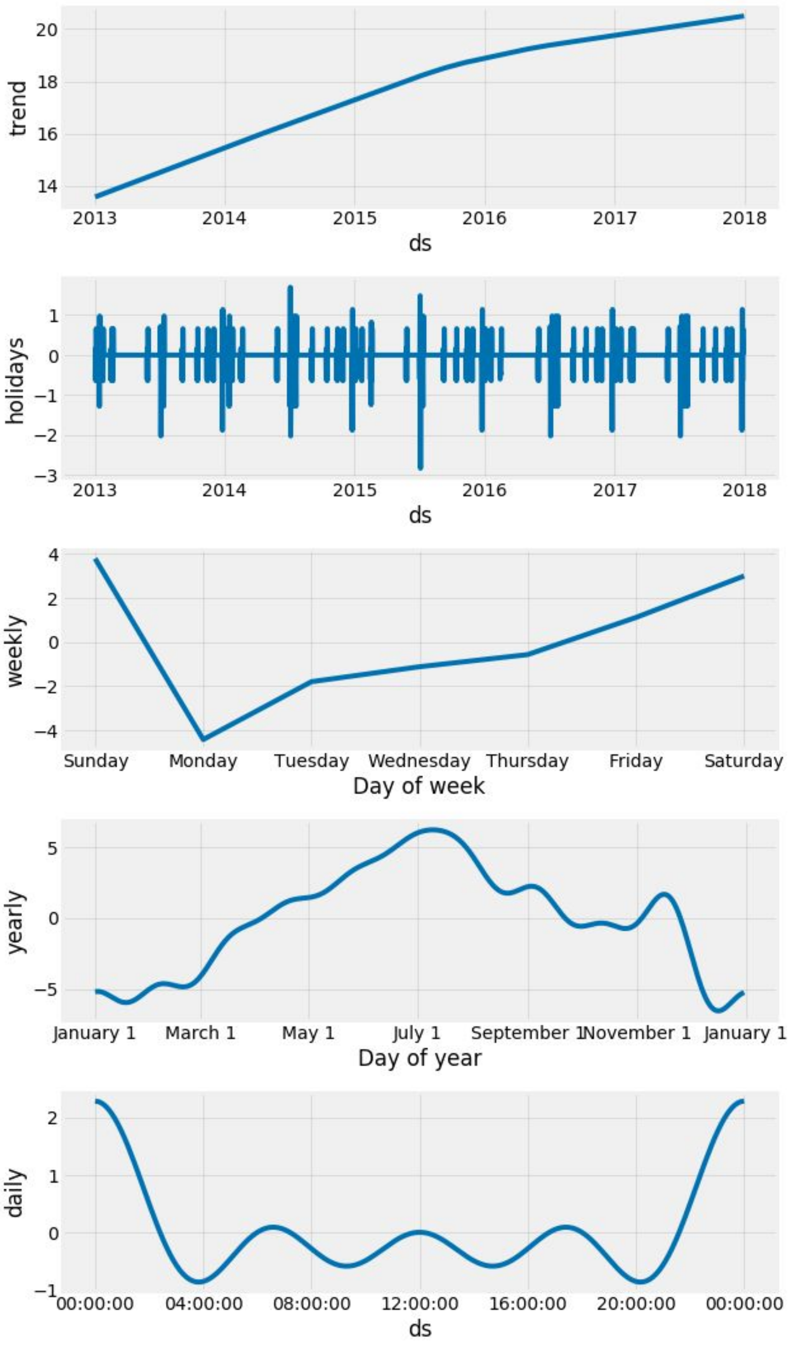


Figure 4.3 *Components of Prophet's forecast on the sales of item 1 at store 1 with holidays*

Prophet's forecasts with holidays taken into consideration look similar to those without holidays:

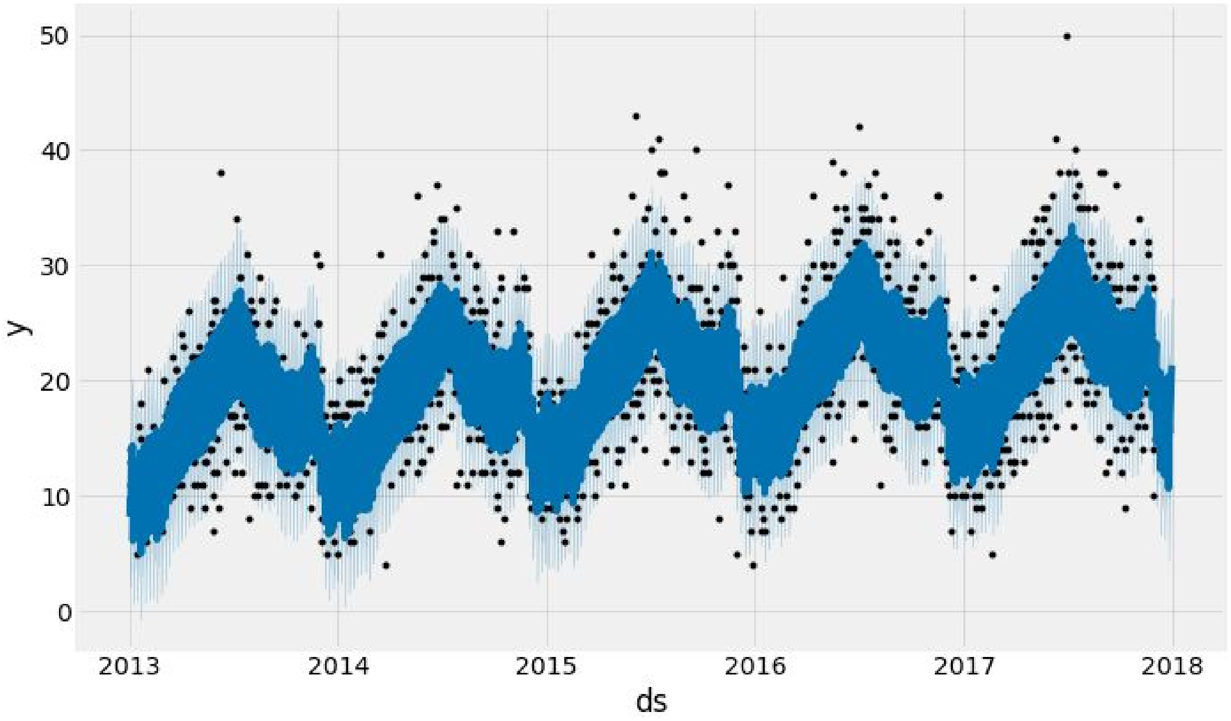


Figure 4.4 *Predictions of sales of item 1 at store 1 using Prophet with holidays*

5.2 Evaluating Prophet Forecasts

A custom function is written to calculate the MAPE (mean absolute percentage error) and MAE (mean absolute error) of Prophet's forecasts with and without holidays. The relative error of Prophet's forecast (MAPE) is about 27.5%, and on average the model is wrong by 3.54 predicts (MAE). The two Prophet models are comparable in results:

	Prophet	**Prophet with Holidays**
MAPE	27.49667012665557	27.64935663329023
MAE	3.5474778019043165	3.5466413004468964

For the sales of item 1 at store 1, Prophet covers the peaks and troughs with its predicted upper and lower bounds, and it more accurately models the weekly seasonality and trend than SARIMA:

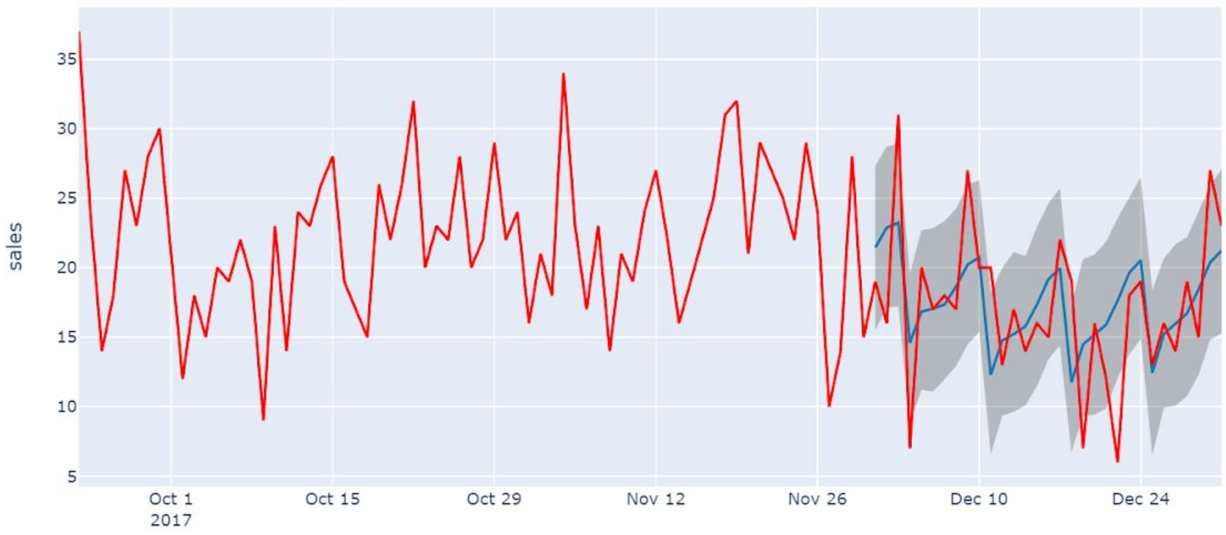


Figure 4.5 *Actuals, forecasts, upper and lower bounds of Prophet predictions*

5.3 Comparing Prophet and SARIMA

In overlaying the actual sales of item 1 at store 1 with the forecasts of Prophet and SARIMA, although both Prophet and SARIMA predicted similarly for seasonality, the Prophet model has predicted values that are closer to the actuals in the validation set. This might be due to the fact that the Facebook Prophet modeling function was configured to take into account yearly, weekly and daily seasonality.

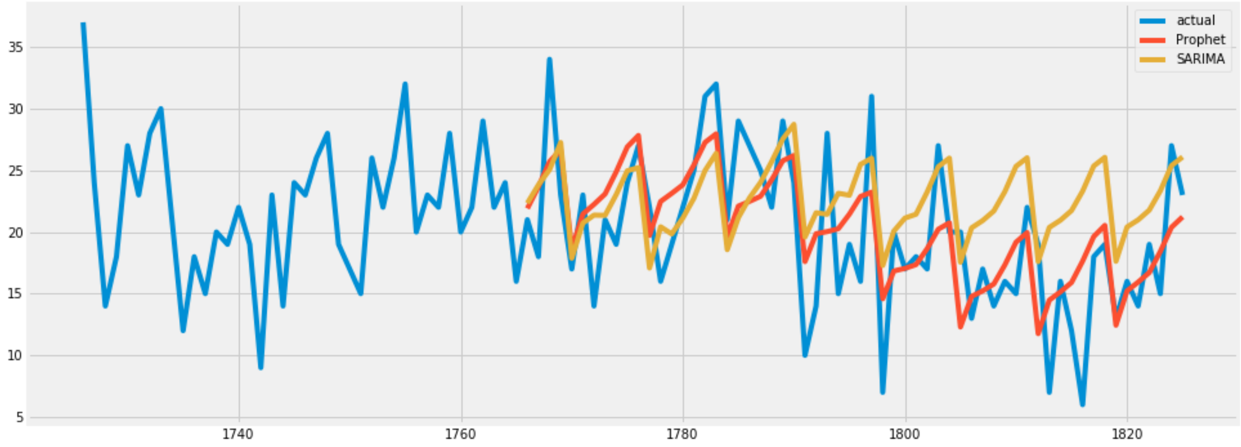


Figure 4.6 *Actual sales of item 1 at store 1 and forecasts of Prophet and SARIMA*

The relative error of the SARIMA forecast (MAPE) is about 49%, and on average its model is wrong by 6 predicts (MAE), almost double that of the Prophet model:

	Prophet	SARIMA
MAPE	27.49667012665557	49.00306558565962
MAE	3.5474778019043165	6.012676922341632

6. Summary

The time series for the sales of 50 items across 10 stores from 2013 to 2017 has an overall upward trend and yearly seasonality. Total sales volume increased from year to year. Sales climbed to a peak in July from January each year, and went back down to a trough at the end of the year with a slight peak in November.

For individual items, the seasonal sales peaks and troughs are less regular (harder to predict) throughout the years, and the upward trend is slight and less pronounced. Decomposition confirms an upward trend and seasonality, which indicates that transformation is required prior to modeling with ARIMA.

Features are added to the dataset that describe the day, week, week of year, month, etc of the dates. Tree based learning algorithms from the LightGBM framework are used to determine importances of these date features. The resulting feature importances shows the week, item, and day as the top 3 most important features having the biggest impact on sales numbers. Conversely, the store, day of year, and month are among the least important features. Therefore, sales should be modeled by item while taking into consideration weekly and daily seasonality.

ARIMA is the first model evaluated for predicting this time series. Therefore, stationarity of the time series needs to be checked and transformed if necessary. Using the augmented Dickey-Fuller test, the null hypothesis is accepted and the time series is non-stationary.

First order differencing is done to transform the non-stationary time series. Using the augmented Dickey-Fuller test on the transformed time series, the null hypothesis is rejected and the transformed time series is stationary. This allows ARIMA and SARIMA to be used to model the transformed time series.

By evaluating the ACF and PACF plots of the transformed time series, parameters for the ARIMA modeling function are determined, with p = 6, d = 0, and q = 3. ARIMA(6,0,3) is fitted and diagnostics of the model's residuals show them as not random. This concludes the ARIMA model to provide an inadequate fit to the data.

Feature importances indicated week as a feature having great impact on the sales of an item. The seasonal order parameter m is set to 7 is the model SARIMA(6,0,3,1,1,1,7) and fitted.

Diagnostics of the model show that the residuals are random. This confirms the SARIMA model as an adequate fit to the data. However, upon forecasting item sales with the SARIMA model, it became clear that much of the seasonality is still not taken into account, as SARIMA only provides configuration of one type of seasonality.

Facebook Prophet is the final model evaluated for this time series. Prophet provides the flexibility to configure many seasonal orders and to model non-stationary time series directly without transformation. Prophet models with and without holidays provided comparable predictions in sales, with a relative error of 27.5%. On average the Prophet model is wrong by 3.54 predicts.

Comparatively, on average the SARIMA model is wrong by 6 predicts. Although both Prophet and SARIMA predicted similarly for seasonality, the Prophet model has predicted values that are closer to the actuals in the validation set. This might be due to the fact that the Facebook Prophet modeling function was configured to take into account yearly, weekly and daily seasonality. By providing upper and lower bounds of the predictions, Prophet also covers more of the data’s peaks and troughs than SARIMA.