Improving Demand Forecasting

A Time Series Study by Juichia Che Holland

Define

The Problem

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 in a span of 5 years and goes through a process of fine tuning the forecasts in demand.

The Data

The dataset comes from Kaggle's store item demand forecasting challenge. It describes the daily sales numbers over 5 years of 50 items across 10 stores each and comes in 2 csv files, train.csv and test.csv. 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

0 - 913,000

2013-01-01 to 2017-12-31 per item per store

1 - 10 per item

1 - 50

0 - 231

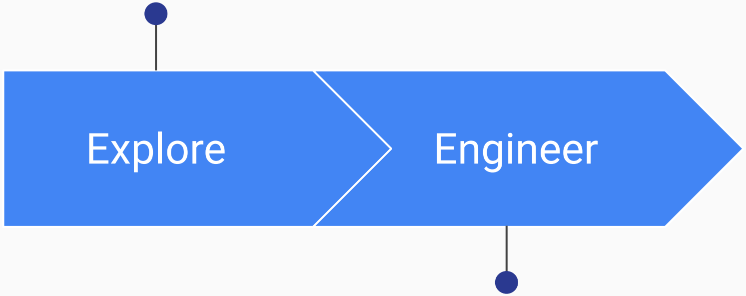
The Process

Explore the descriptive statistics, density, distribution, trend, and seasonality of overall and individual item sales

Explore

The Process

Explore the descriptive statistics, density, distribution, trend, and seasonality of overall and individual item sales

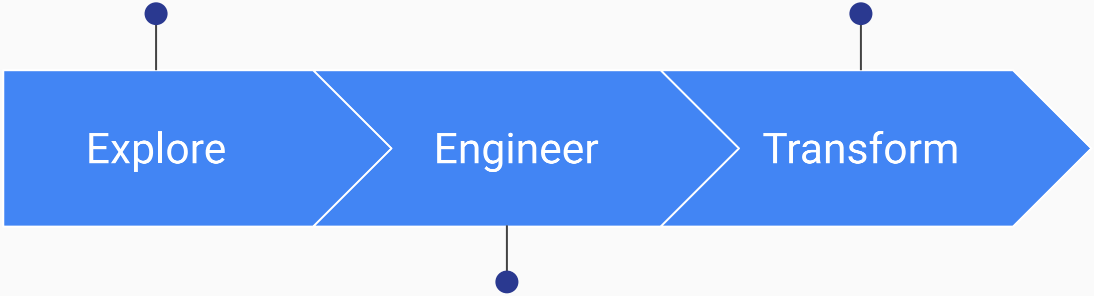


Extend the dataset with descriptive features for the dates. Determine feature importances based on predictive modeling using tree based learning algorithm.

The Process

Explore the descriptive statistics, density, distribution, trend, and seasonality of overall and individual item sales

Test and transform the time series for stationarity.

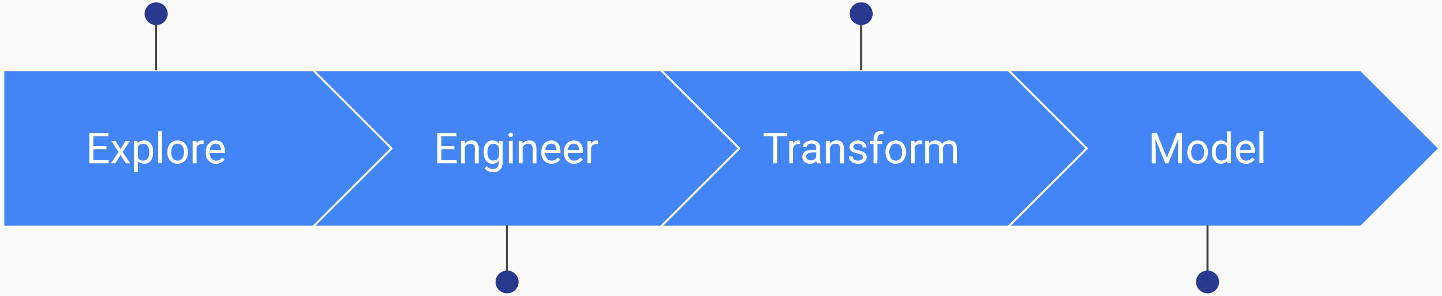


Extend the dataset with descriptive features for the dates. Determine feature importances based on predictive modeling using tree based learning algorithm.

The Process

Explore the descriptive statistics, density, distribution, trend, and seasonality of overall and individual item sales

Test and transform the time series for stationarity.



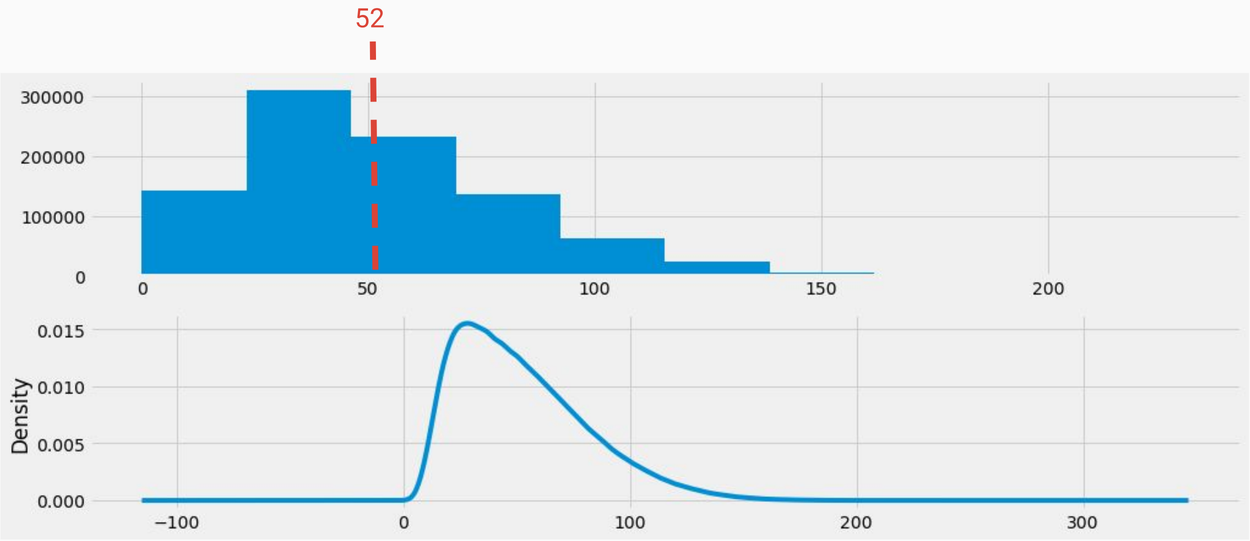
Extend the dataset with descriptive features for the dates. Determine feature importances based on predictive modeling using tree based learning algorithm.

Forecast item sales using a variety of models. Tune model parameters. Evaluate and compare the results of prediction.

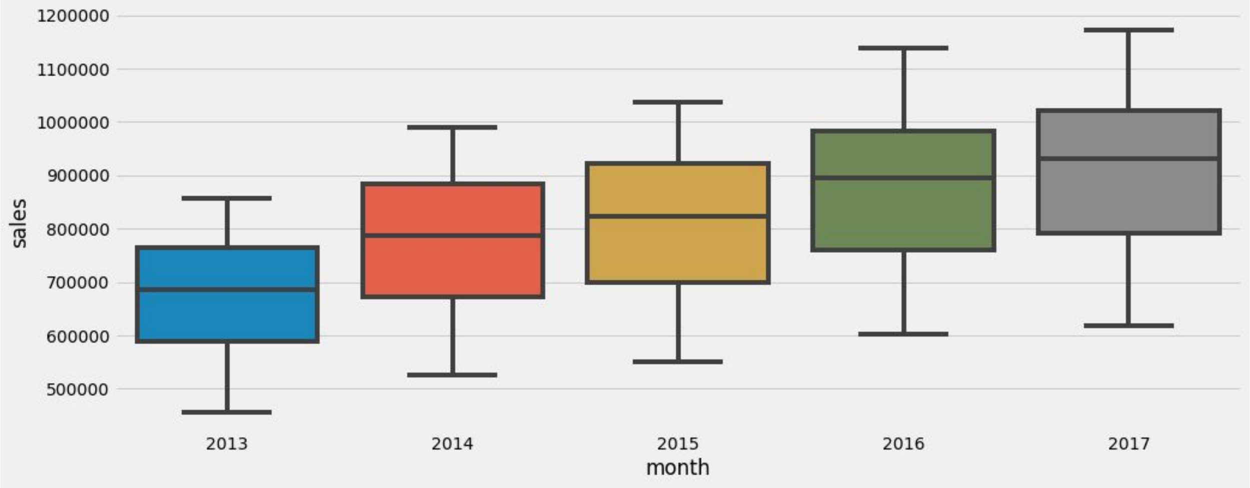
Explore

Overall Sales

Daily item sales range from 0 to 231. On average, 52 of an item are sold per day at any given store. 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.

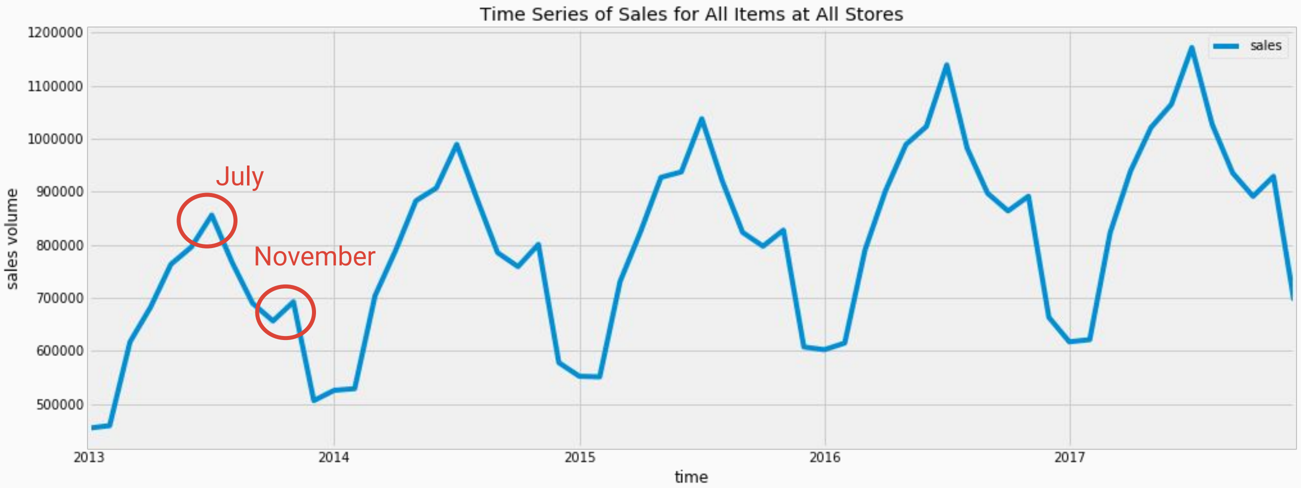


Overall Sales



Median sales values have an upward trend over time. Total annual range of sales increased slightly each year, and there is also an increase in the spread, or middle 50% of the data, over time. This indicates that the time series for total sales has a predictable trend.

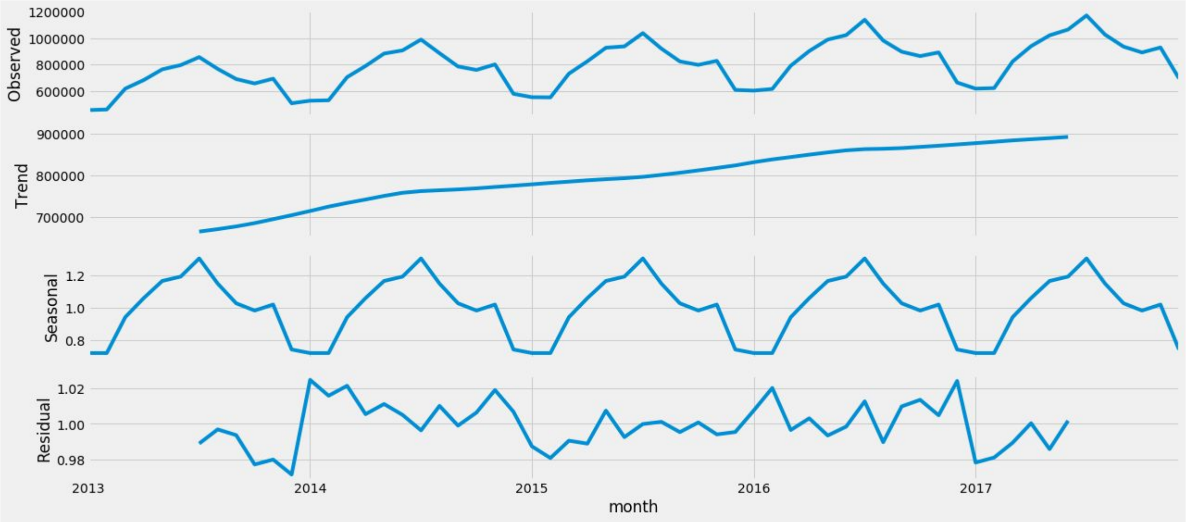
Overall Sales



Each year, total 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.

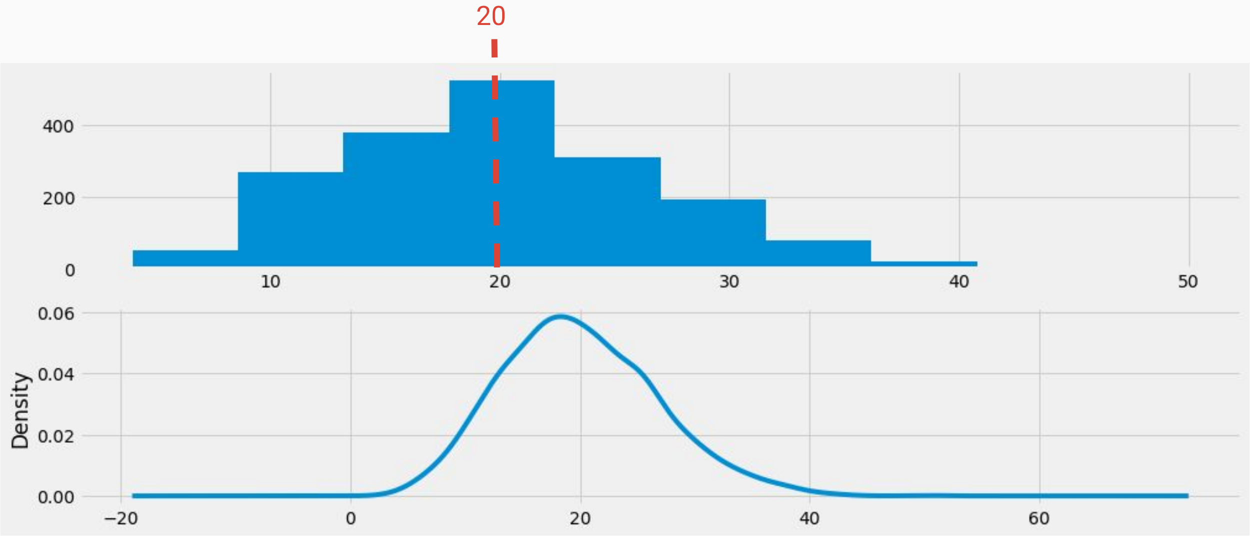
Decomposition

An upwards trend and strong seasonality are visible with decomposition of the time series for the sales totals of all items at all stores. This indicates that transformations might be required prior to modeling, and that models which take seasonality into consideration should be utilized when making forecasts.

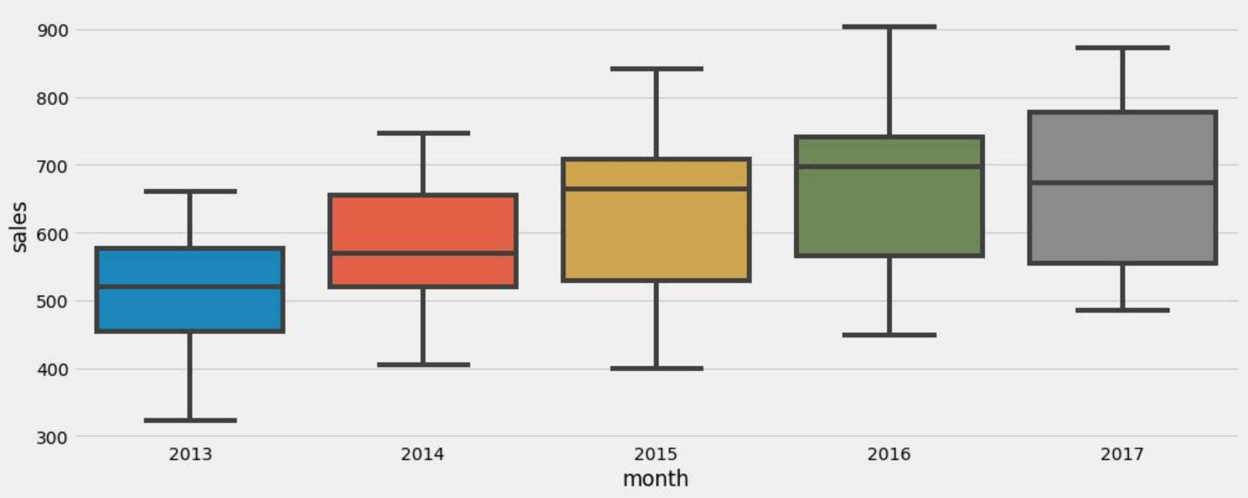


Individual Sales

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.

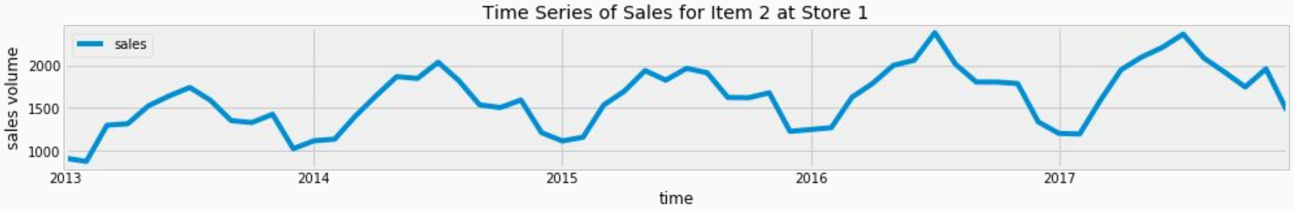


Individual Sales



Both the median and range of annual sales for item 1 at store 1 decreased from 2016 to 2017. There is an overall increase in the spread over time. This could mean that the time series of individual items is less predictable in trend.

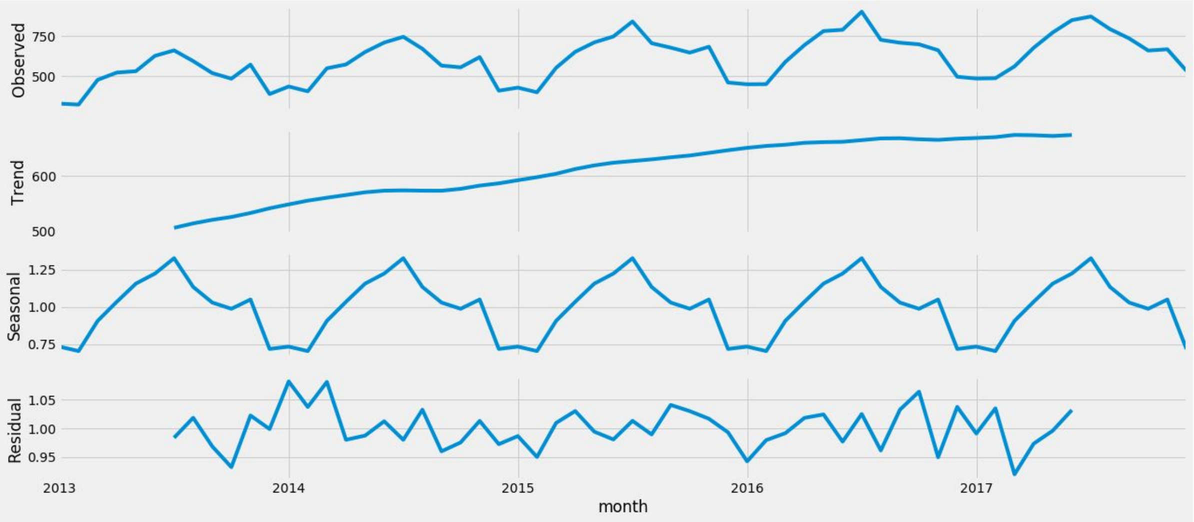
Individual Sales



For both item 1 and item 2 at store 1, the peaks and troughs are irregular throughout the years. The trends and seasonality of the time series are less pronounced.

Decomposition

An upwards trend and strong seasonality became more visible with decomposition of the time series for the sales of item 1 at store 1. This indicates that transformations might be required prior to modeling, and that models which take seasonality into consideration should be utilized when making forecasts.



Engineer

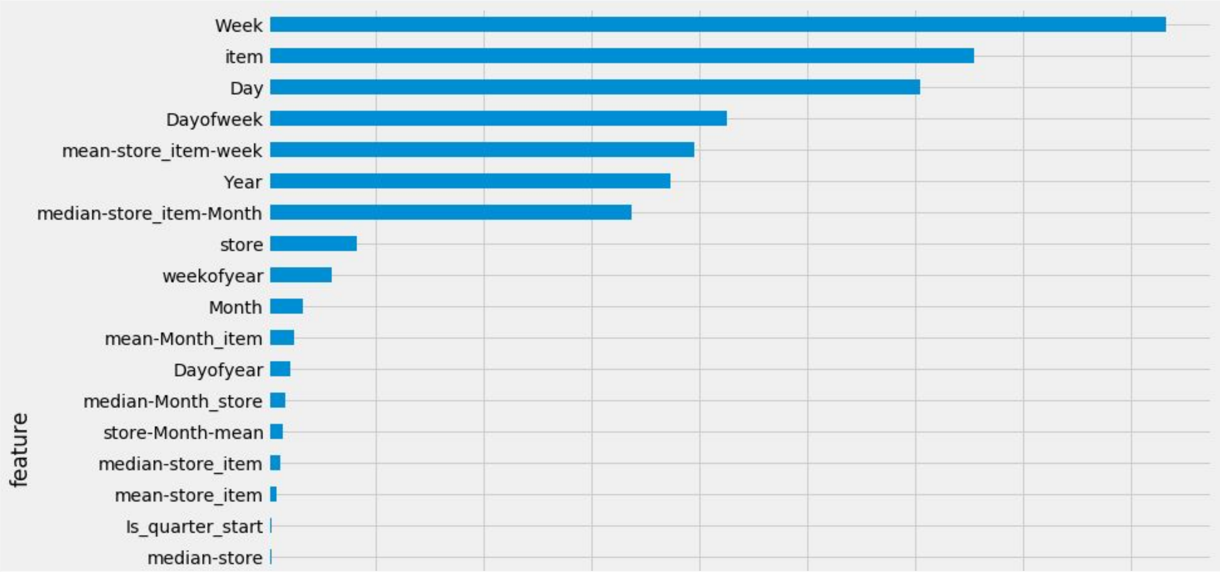
Extending Date Features

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

Feature Importances

Tree based learning algorithms from the LightGBM framework are used to determine feature importances.

The week, item, and day are 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



Transform

Test for Stationarity

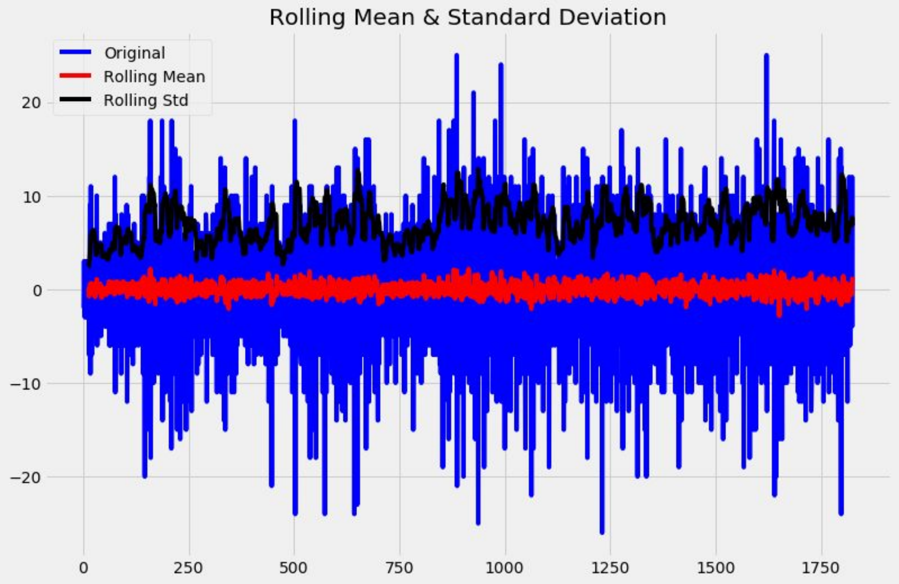
Stationarity of an individual time series (for item 1 at store 1) is checked using the augmented Dickey-Fuller Test. Here the null hypothesis is that the time series is non-stationary.

Since the series has a strong upward trend, we will use the strictest 1% critical value to test the hypothesis. We accept the null hypothesis and the time series of item 1 at store 1 is non-stationary. Transformations of individual time series are necessary prior to modeling using ARIMA models.

Results of Dickey-Fuller Test:

Test Statistic	-3.157671
p-value	0.022569
#Lags Used	23.000000
Number of Observations Used	1802.000000
Critical Value (1%)	-3.433984
Critical Value (5%)	-2.863145
Critical Value (10%)	-2.567625

Differencing

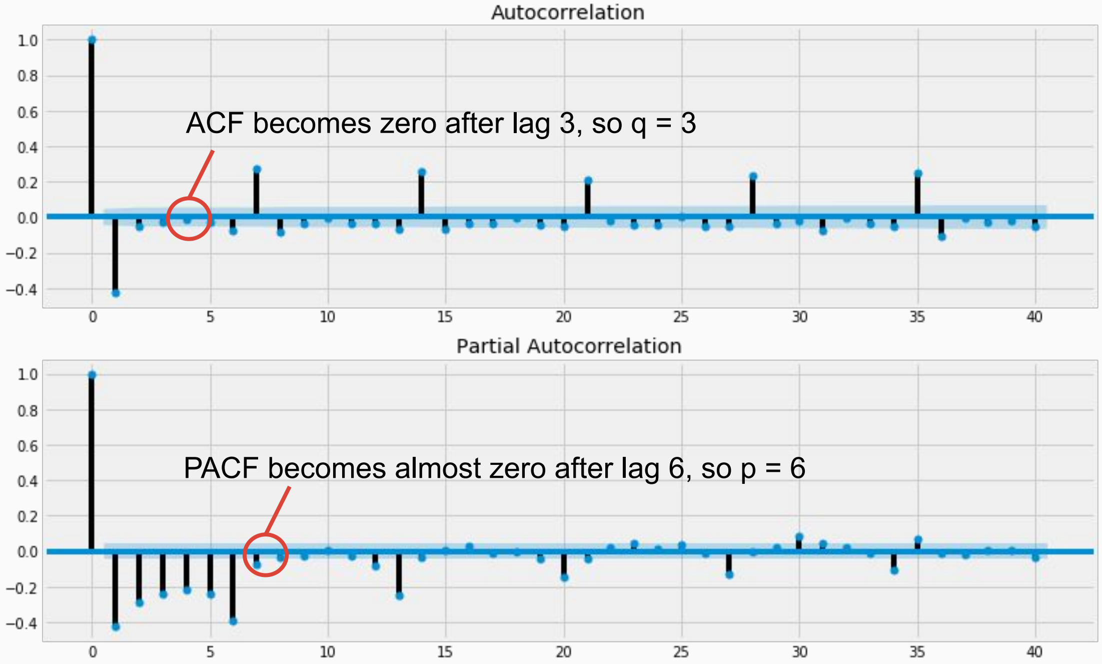


To transform the series of sales numbers of each item at each store, we take the difference of the observation at a particular instant with that at the previous instant using first order differencing in Pandas. The transformed time series shows constant variance over time.

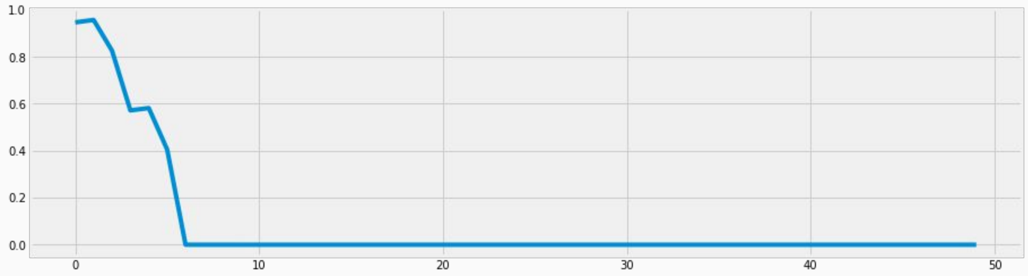
Model

Determining ARIMA Parameters

ACF and PACF plots provide diagnostics that help determine the important parameters in the ARIMA modeling function - p, d, and q. 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.



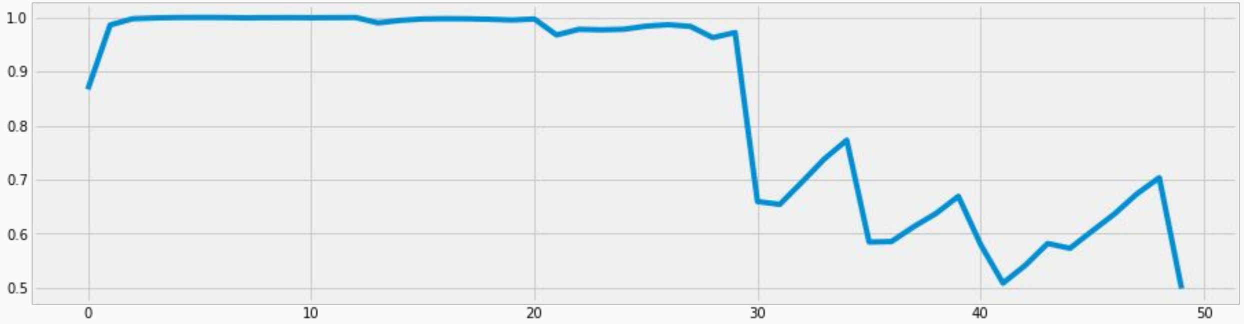
Evaluating Fit of ARIMA



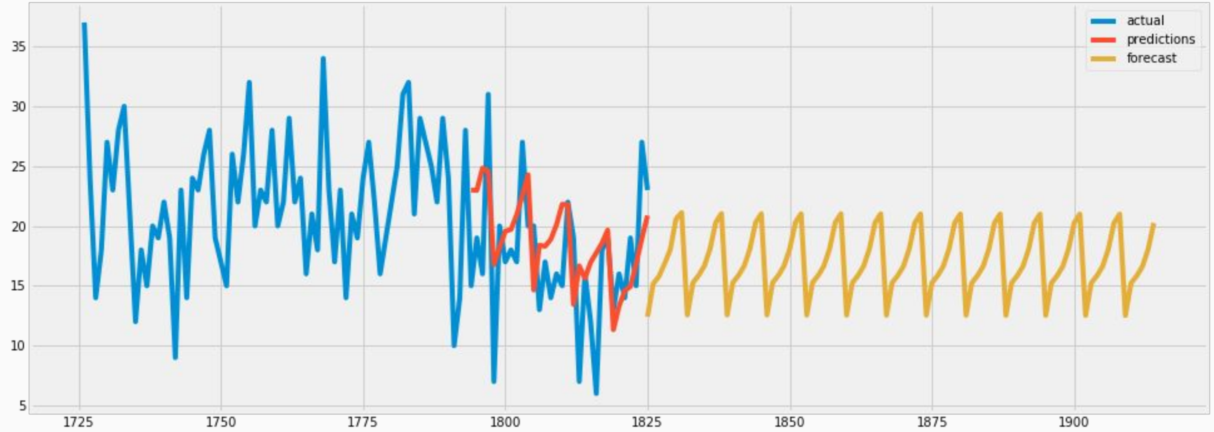
The Ljung-Box test result for fitted ARIMA(6,0,3) model shows that for more than 45 of the first 50 lags, the Ljung-Box statistics are 0 which is lower than the 0.05 threshold, indicating that the residuals are not random. Diagnostics of the residuals using this test indicates that the ARIMA model does not provide an adequate fit to the data. Therefore, this model is not further utilized in making predictions.

Evaluating Fit of SARIMA

The Ljung-Box test result of the fitted SARIMAX(6,1,3,1,1,1,7) model with weekly seasonality 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 and that the model is an adequate fit for the data.



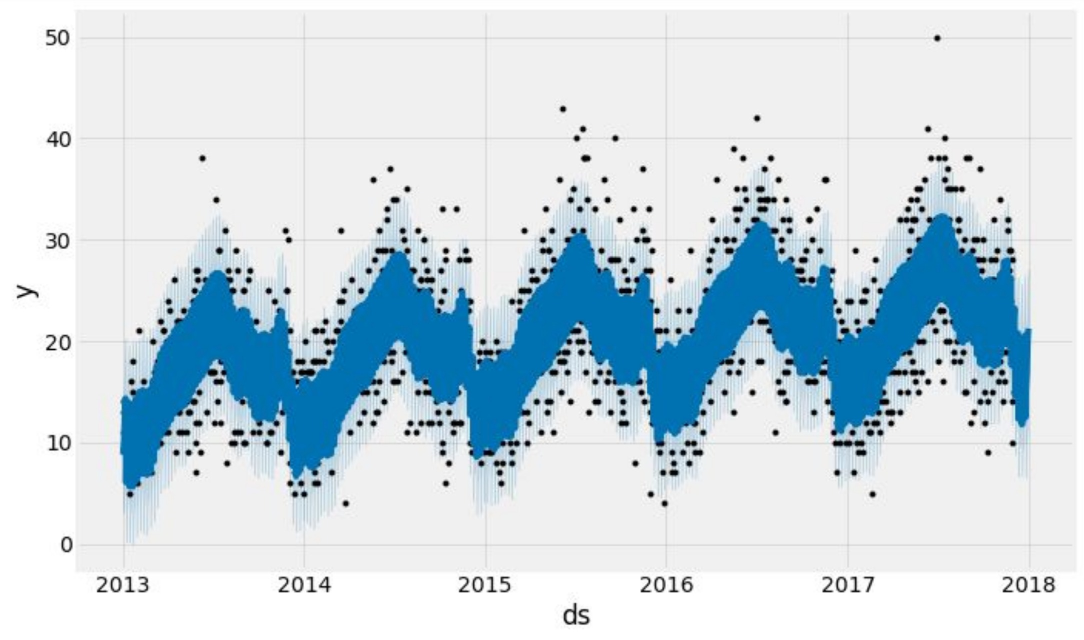
Evaluating SARIMA Forecast



The relative error of the SARIMAX(6,1,3,1,1,1,7) model's forecast (MAPE) is about 49%, and on average its forecast is wrong by 6 predicts (MAE).

Predicting Sales with Facebook Prophet

Prophet's sales predictions (in dark blue) come with the forecasted value, an upper bound, and a lower bound value. The forecasted range gives Prophet an advantage in accuracy.

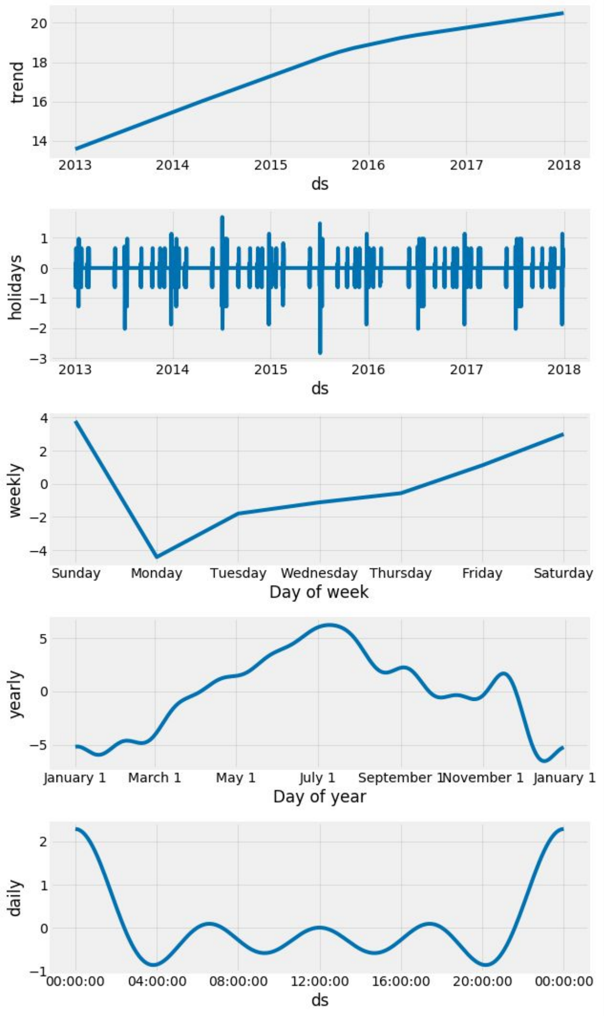


Diagnosing Prophet Components

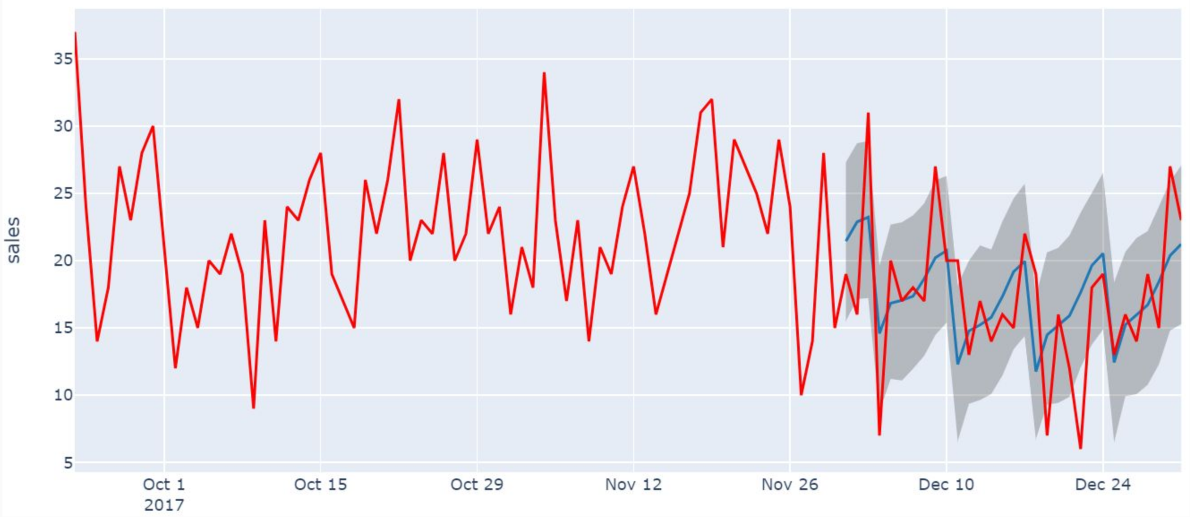
The yearly component in Prophet model's decomposition suggests that Independence Day, the play offs, the Super Bowl, and Thanksgiving can be added as holidays to improve the forecast. The weekly component also makes clear the peaks that occur on Saturday and Sunday.

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.

Prophet's model is configured to account for yearly, weekly, daily seasonality along with holidays.



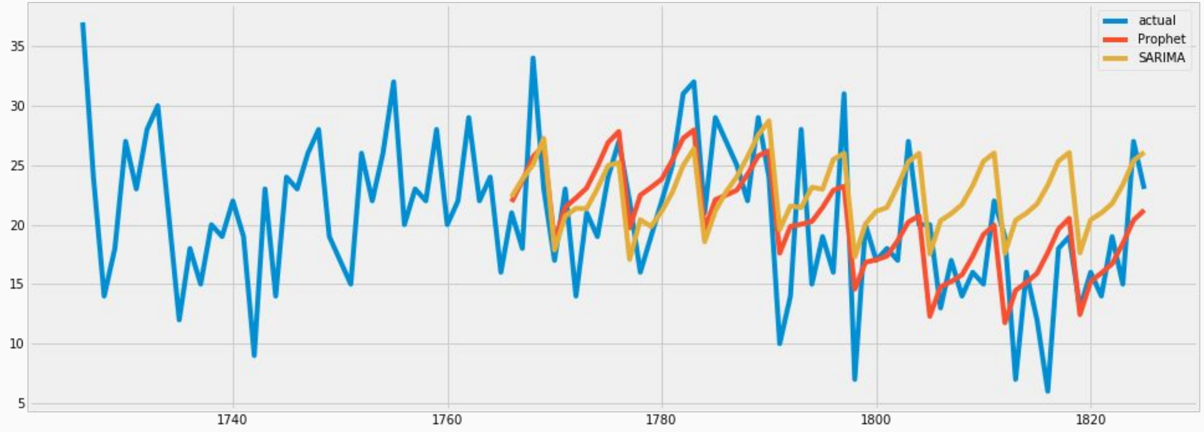
Evaluating Prophet Predictions



The relative error of the forecasts of the Prophet model with yearly, weekly, daily seasonality, and holidays (MAPE) is around 27.5%, and on average it is wrong by 3.54 predicts (MAE).

Prophet vs. SARIMA

Although both Prophet and SARIMA predicted similarly for seasonality, the Prophet model has more predictions 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 where as SARIMA only accounts for one seasonal order. The SARIMA model is wrong by 6 predicts on average, which is almost double that of the Prophet model.



Summary

ARIMA(6,0,3) fitted model and its non-random residuals proved to be an inadequate fit for the data. SARIMAX(6,1,3,1,1,1,7) fitted model accounts for weekly seasonality with parameter m = 7 and its random residuals proved it to be an adequate fit for the data. The relative error of the SARIMAX(6,1,3,1,1,1,7) model's forecast is about 49%, and on average its forecast is wrong by 6 predicts.

Facebook Prophet provided flexibility in dealing with complex seasonalities and trends. The Prophet model was configured to account for year, weekly, as well as daily seasonality. Holidays that occur around the annual peaks in July and November are also passed into the modeling function. The relative error of Prophet's forecast is around 27.5%, and on average it is wrong by 3.5 predicts.

While SARIMA and Prophet both accurately predicted the weekly climbs and drops, overall the Prophet model made more predictions closer to actual sales. The two models started out predicting equally closely to the actuals, but as time went on the predictions grew further apart, with Prophet making the more accurate predictions. This could be due to the fact that Prophet is able to take into account multiple seasonalities.

When dealing with non-stationary time series that have strong trends and multiple seasonalities, modeling functions that process **high and multiple seasonal orders** seem to perform better over time.

Thanks!

juichiaholland@gmail.com
https://www.linkedin.com/in/juichiache/