Springboard Capstone 2

Sales Forecasting Model

by

Vaishali Kalekar

Table of Contents

Purpose

The aim is to build a forecasting model for sales of items' demands across multiple stores. We have had access to historical data for the past 5 years on a daily basis for 10 stores with 50 items in each store. The forecasting needs to be done for these 50 items each for 10 stores for the next 90 days.

Data Wrangling / Cleaning

Data Acquisition

The training and test data for the model is acquired from kaggle.
https://www.kaggle.com/c/demand-forecasting-kernels-only/data

Storage and access

The data is stored locally as well as in the project's Github repository. Through the jupyter notebook, the .csv data file is opened and data is loaded in pandas dataframe.

Description of Data

Once the data is loaded into the dataframe, we see that the data set for this project consists of 913000 observations and 4 features. Sales is our target feature and from the values looks like is a continuous variable. Each row is by the day. So these sales numbers are for a store for an item by the day. So we are looking at the times series forecasting in this case.

Data Types

Further investigation reveals that the data types in the raw data have mostly of the type integer except for the date column.

Missing Values

All the 4 columns have no missing values. We do not have to worry about any value imputing/ deleting this time.

Outliers

We figure out the outliers with The interquartile range (IQR) approach, also called the midspread or middle 50%. It is a measure of the dispersion similar to standard deviation or variance, but is much more robust against outliers. The interquartile range shows how the data is spread about the median. IQR is somewhat similar to Z-score in terms of finding the distribution of data and then keeping some threshold to identify the outlier. Next we remove the outliers identified by the Interquartile Rule for Outliers.

EDA and Feature Analysis

EDA

Univariate and Bivariate Feature Analysis

Next, We will inspect each feature individually.

We will perform the univariate and bivariate analysis using the following steps-

1. Summary statistics
2. Plot individually
3. Plot against our target variable- sales to see if and how both the features are related
4. Modify the feature to make it ready for modelling, if necessary

For individual summary statistics we will use

- .describe()
- .sample(5)
- .nunique()
- Group_by if necessary

1. Store

   This shows there are 10 unique stores in our dataset. Each store has 91300 individual sales recorded in the dataset. The column has no missing values and no special characters etc in the data values. The values are 1-10. So we will not need to do any pre-processing.

After plotting we see that-

a. Store 2 and 8 have high highest sales
b. Store 5 and 7 have lowest sales
c. There is a consistent increase in sales numbers every year at all the stores!

2. item

There are 50 items in each store. The item column values range from 1-50. There are no special characters in the item values, neither are there any outliers. Moreover there are no nulls and each store sells each of these 50 items. So we will not need to do any preprocessing/ transformations before the model training.

3. date

The dates are from Jan 1 2013 to Dec 31 2017. The dates are for a 5 years period with the values ranging by the day in this period. The sales are recorded for each item for each day for all of the 50 stores. Similar to item, there are no special characters in the date values, neither are there any outliers. Moreover there are no nulls and each sale is recorded for a date for an item in a store. So we will not need to do any preprocessing/ transformations before the model training

Seasonality

Next we plot the sales to look at the seasonality for- -Yearly -monthly -weekly. After plotting we observe that

- the sales go up year to year from 2013 to 2017
- there is a steady upward trend.
- there is also a monthly seasonality where sales are going up around middle of the year
- the sales follow the same trend year after year where sales peak up at the middle of the year and taper off with one exception in the month of November. Over the years the sales have gone up basically from year to year but follows the same trend.

Feature Engineering

Feature Transformation

From the univariate analysis of sales, we see that values range from 0 to 235 with a median of 47. After plotting sales, we observe that it has a right skew. Since this is our target variable, we will need to remove the skew. We apply different techniques to remove skew including

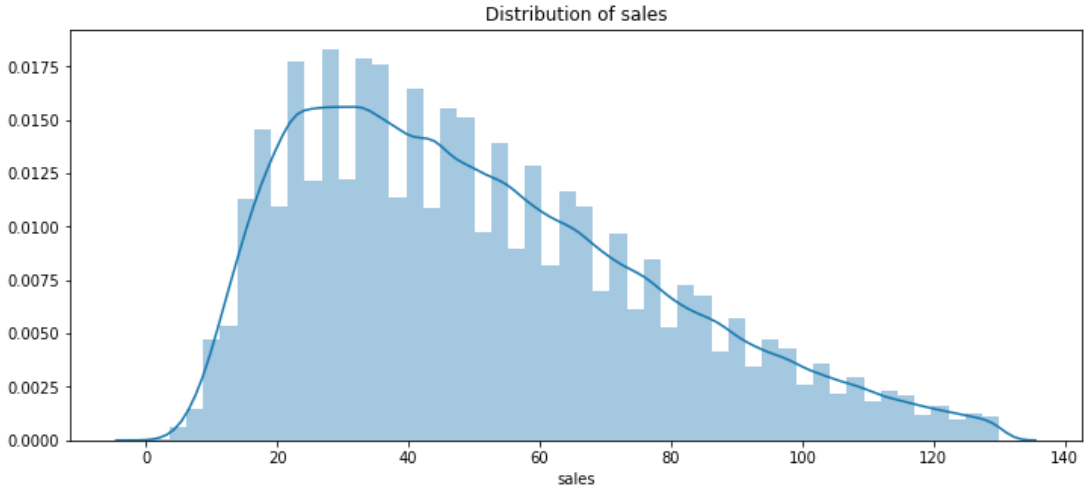
Transformations - Data Skew

As we see, the sales data distribution is a right skewed distribution. We will apply few transformations and evaluate which one is a better fit for removing the skew

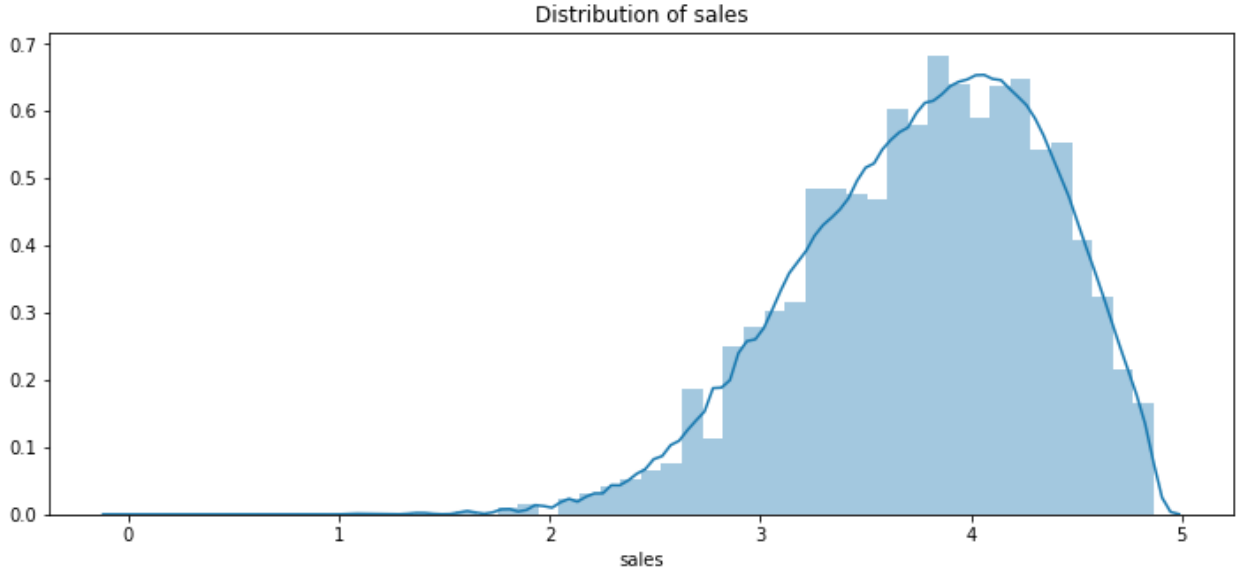
a. Log Transform
b. Square root transform
c. min max scaler transform

Of all three transforms, square root transform works the best in removing the right skew for our dataset values as seen from the plots. So of all, we will use the square root transformation for our ‘sales’ data.

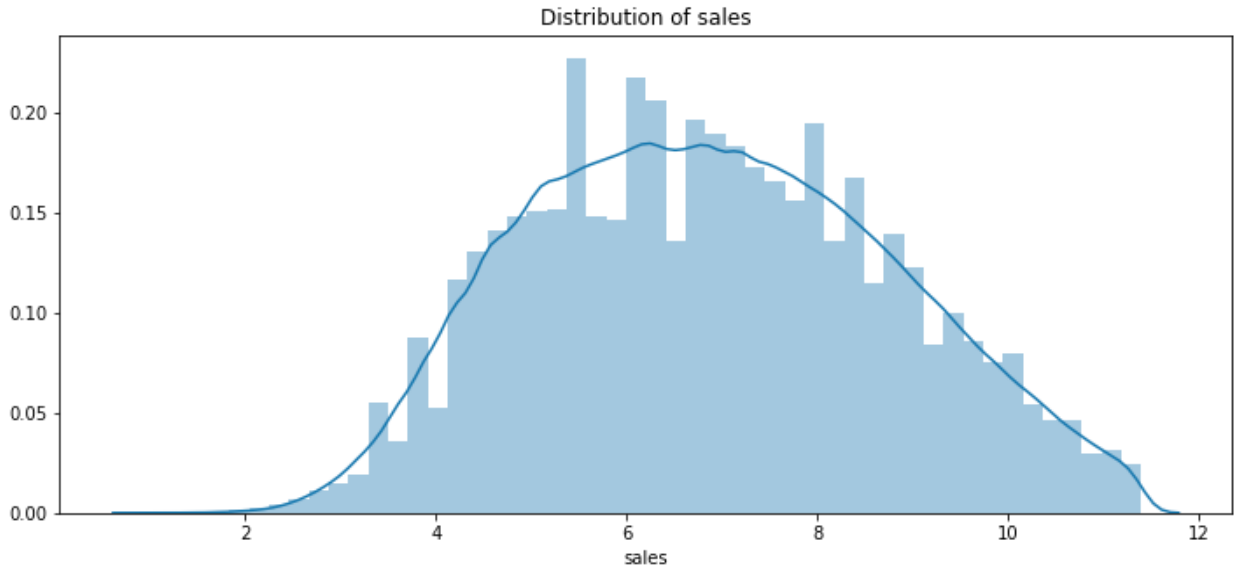
Distribution of sales before removing the skew



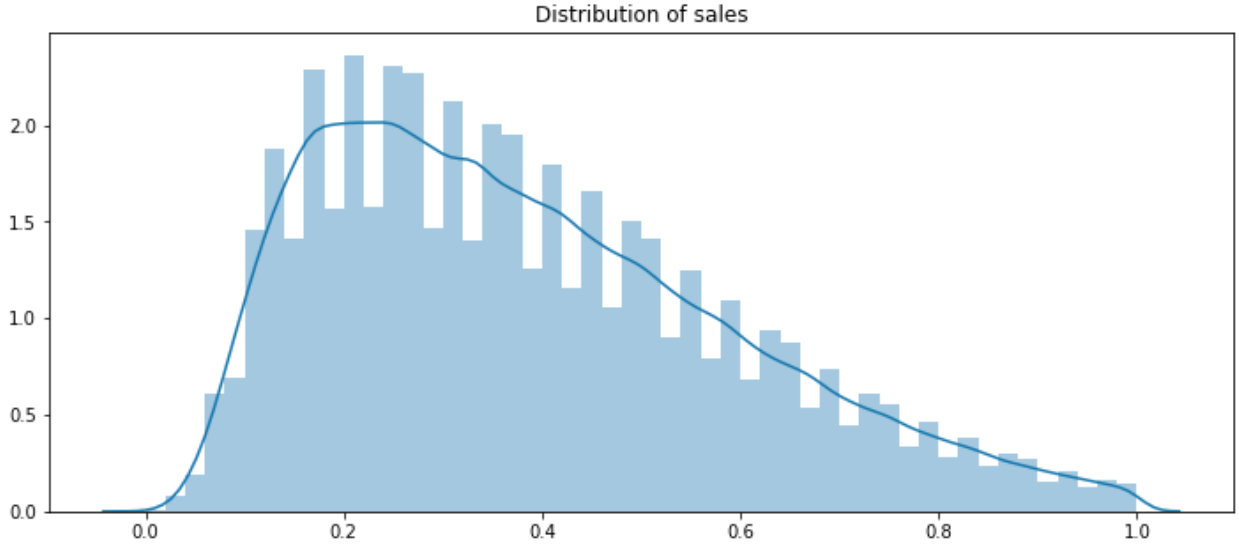
After log transformation



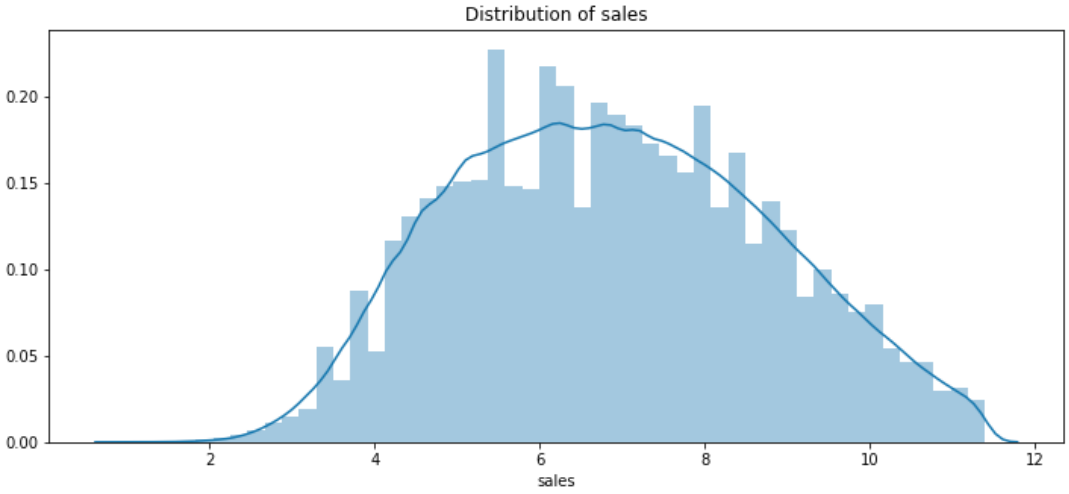
After square root transformation



After min max scaler transformation



Distribution of sales after removing the skew



Inferential Statistics - Two-Sample Independent t-Test

Test:

We will use two sample t statistic test for this problem
A two-sample independent t-test can be run on sample data from a normally distributed numerical outcome variable to determine if its mean differs across two independent groups. We could see if the mean sales differs between 2 stores by collecting a sample of items and recording their sales.

Hypothesis:

Ho: Null Hypothesis: The population mean of one group equals the population mean of the other group, or $\mu 1 = \mu 2$

HA: Alternate Hypothesis : The population mean of one group does not equal the population mean of the other group, or $\mu 1 \neq \mu 2$

Significance:

Confidence Level : 95%

Analysis Plan:

Reject the null hypothesis if p value< a : alpha will be = 0.05

Note: the test is appropriate because the sampling method was simple random sampling, the samples were independent, each population was at least 10 times larger than its sample

Interpretation:

After running the test, and comparing the P-value to the significance level alpha, we see that the P-value is greater than the significance level. (p>a) So we accept the null hypothesis. That means Both sample proportions are equal i.e. with 95% confidence we can say that the two samples come from the same population.

Machine Learning Modeling

Overview and Approach

The aim is to forecast the sales for items sold in the stores. We will have to forecast the sales for the next 90 days.
We have been given the historical sales numbers for the past 5 years for 50 items across 10 stores. The model will forecast the sales for these items across the stores for next 3 months.
The target variable 'sales' is a continuous variable so this is a regression problem.
First we start off with Naive Approach and then move on to fbProphet.

Choice of Performance Metric

Some of the metrics for regression performance are - Root Mean Squared Error (RMSE),Mean Absolute Error (MAE),R Squared (R^2),Adjusted R Squared (R^2),Mean Squared Percentage Error (MSPE),Mean Absolute Percentage Error (MAPE) While MSE deals with absolute errors, MPSE and MAPE work with relative errors.It can also be thought of as weighted versions of MAE.

- MAPE-The mean absolute percent error (MAPE) expresses accuracy as a percentage of the error. Because the MAPE is a percentage, it can be easier to understand than the other accuracy measure statistics. For example, if the MAPE is 5, on average, the forecast is off by 5%.
- The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values.

The MAPE (mean absolute percentage error) is not scale-dependent and is often useful for forecast evaluation. MAPE is considered to be a popular measure of forecast accuracy. We will start off by using RMSE and then move on to using MAPE.

1. Baseline Model - Naive Approach

Naive method assumes that the next expected point is equal to the last observed point. This approach uses the value of last day sales and uses it to estimate sales for next day. So we have y ^ @t+1 = y @t

2. Prophet Baseline

We build the model using the Facebook Prophet package. It allows specifying multiple seasonalities and special events and allows making predictions.

Approach

Since there are 10 stores with 50 items each, we have 500 time series. For training our model we will take the representation of these data series and build our model. Then we will compare and evaluate the model performances. Once we have finalized the best model depending on the performance, we will continue on the best model to forecast our 90 day period for the rest of the stores.
Representative data

We have Store 2 and 8 with highest sales and Store 5 and 7 with lowest sales. We will choose highest selling and lowest selling items from each of these stores
-So we have -
Store 2- top 15 bottom 5 , top is 28 and second top is 15
Store 8- top 15 bottom 5
Store 5- top 15 bottom 5
Store 7- top 15 bottom 5
For the consistency purpose, and going with the majority we will be choosing item number 15 and 5 from each store.

Model Improvements

We will do 2 things for model improvements

1. Transformations
2. Add Holidays and yearly special events to the model

3. Prophet Additive with Holidays

Looking at the sales trend for 'Black Fridays' and over the weekends, we will see if adding holidays feature in prophet will improve the performance of the model or not. We will now add seasonalities like yearly, monthly and daily. In addition we will also add the holidays to better improve the performance.

4. Prophet multiplicative

By default Prophet fits additive seasonalities, meaning the effect of the seasonality is added to the trend to get the forecast. However, in our dataset the time series has a clear yearly cycle, but the seasonality in the forecast is small at the start of the time series, increases gradually at the peak and small again at the end. In this time series, the seasonality is not a constant additive factor as assumed by the Prophet, rather it shows slight growth and then shrinks again at the end. (with multiplicative seasonality, it keeps growing with the trend.) This might be the case of multiplicative seasonality. We will investigate this further and see if this change will bring any model improvement.
Prophet can model multiplicative seasonality by setting seasonality_mode='multiplicative' in the input arguments.

Results Summary

Looking at the yearly plot, it looks like the shopping sales go higher in the month of July. And start tapering off till the end of the month with an exception of the month of November. This might be due to the "black Friday' sales happening in Nov. So understandably the sales picked up in that month. Also if we look at the weekely plot, there is a peak in sales on Saturday and Sunday. The 'trend' plot shows consistent data as observed in our analysis that the sales have picked up year after year with a steady growth.

It seems there is not much difference in additive Vs multiplicative models in terms of the performance improvements.
We choose the additive model with holidays to train the rest of our models.

Comparison of Base Model to “Final” Model

Model	RMSE	MAPE
Baseline Model - Naive Approach	22.90	54.08 %
Prophet - Additive Model	10.11	8.13%
Prophet - Additive with seasonalities + holidays + special events and after removing the skew	0.52	4.12%
Prophet - Multiplicative with seasonalities + holidays + special events and after removing the skew	0.53	4.19%

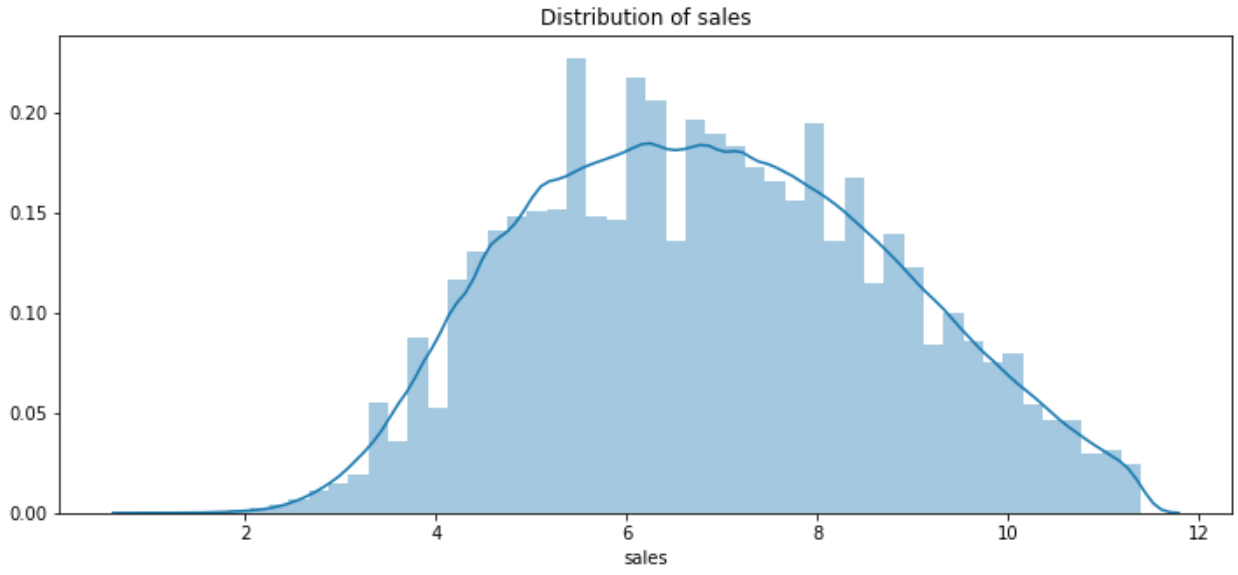
There is a huge improvement from base model performance to both our final models.
We see that prophet additive with seasonalities and prophet multiplicative with seasonalities performed almost similarly. However Prophet - Additive with seasonalities + holidays + special events and after removing the skew gave the best performance.

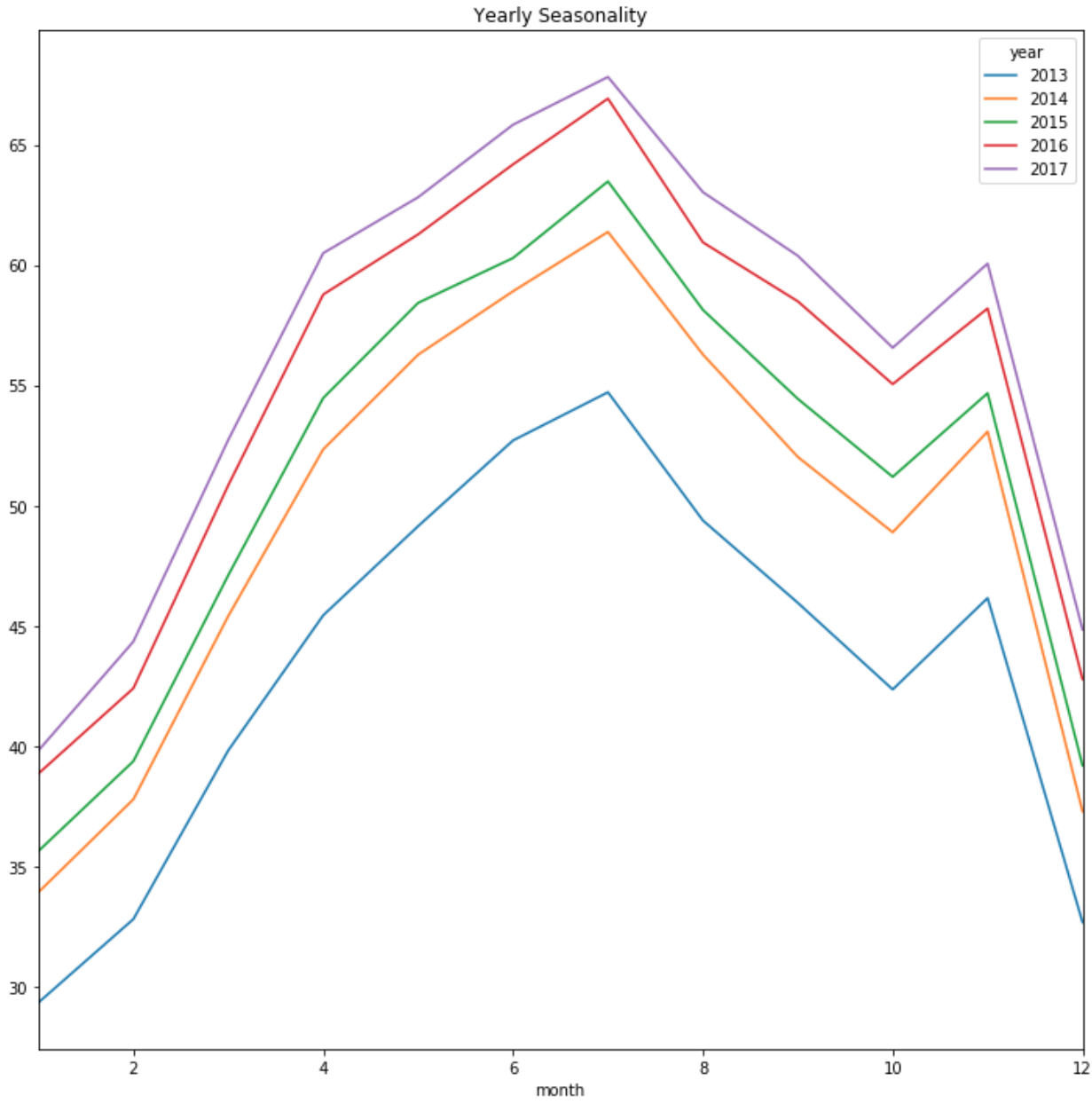
Future Improvements

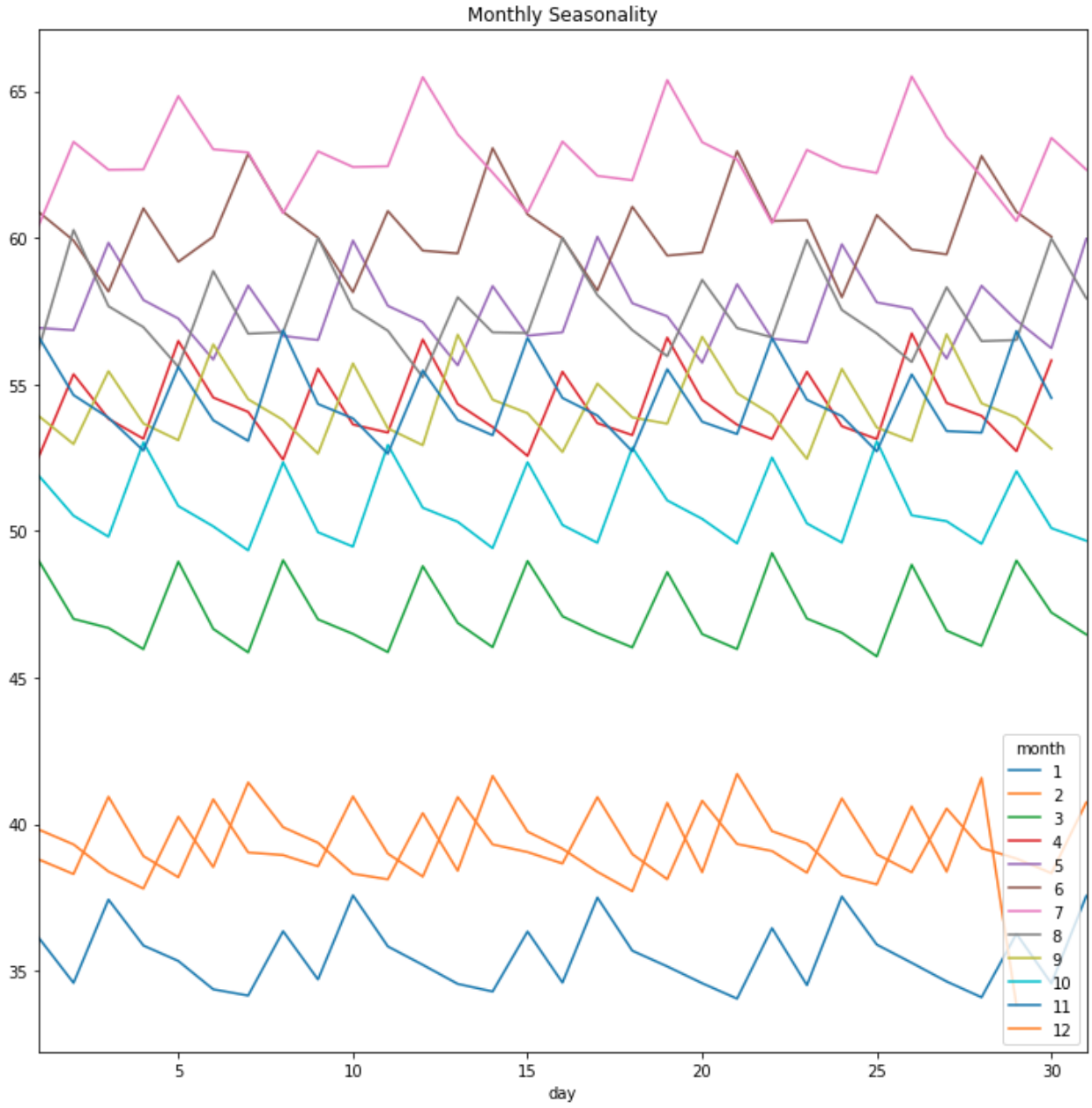
As we have seen earlier, our dataset had 500 time series for 10 stores with 50 items each. Right now we treated each time series separately and trained/ forecasted the sales accordingly. However in future if fbProphet comes with any improvements of handling multiple time series together in one go, that will make the whole job of predicting in such scenarios a whole lot easier. And we will have to make changes accordingly too.

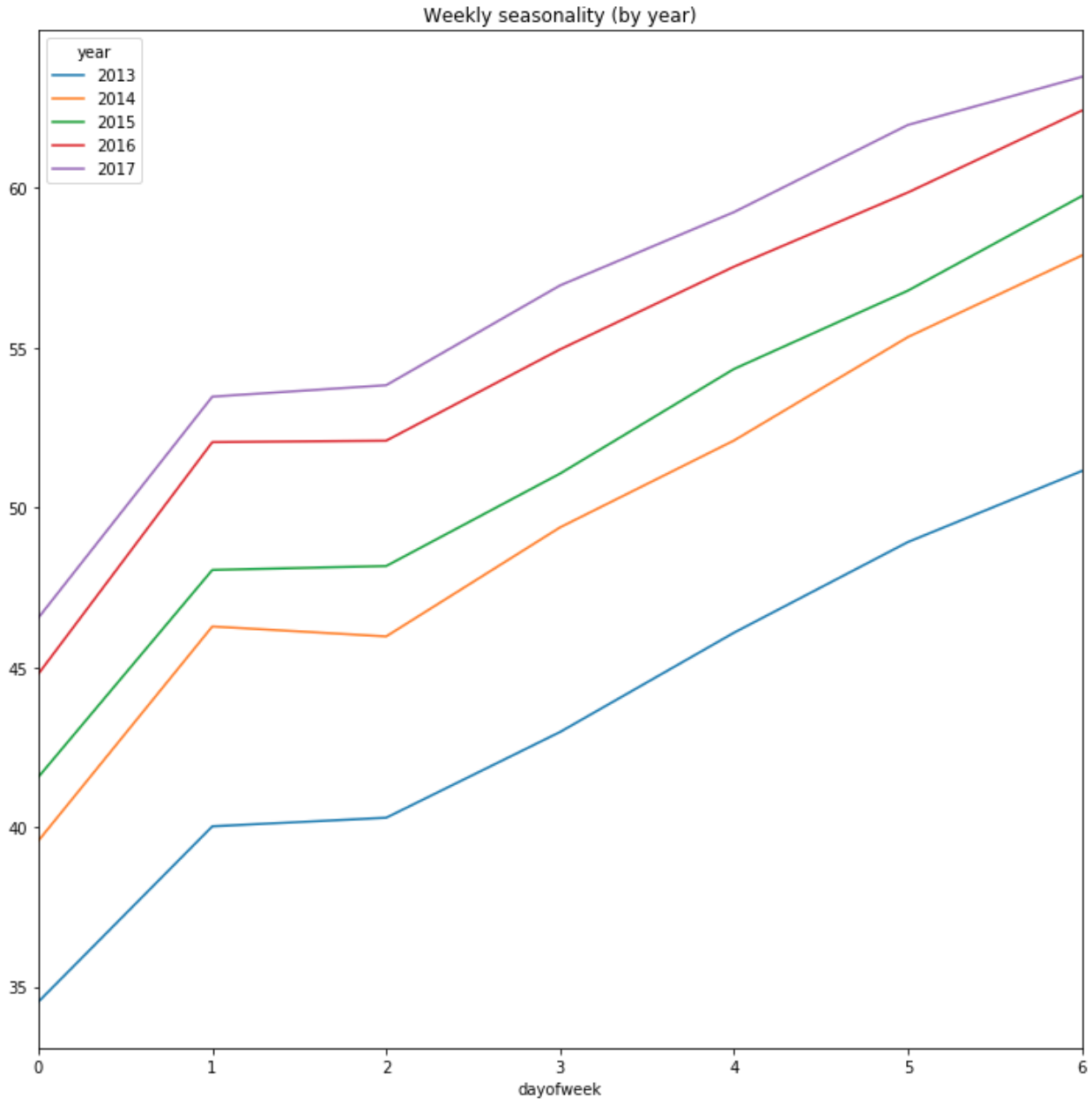
Appendix : Distribution of sales and seasonality plots

Distribution of sales after removing outliers and right skew




Yearly Seasonality
year
2013
2014
2015
2016
2017
65
60
55
50
45
40
35
30
2
4
6
8
10
12
month


Monthly Seasonality
65
60
55
50
45
40
35
5
10
15
20
25
30
day
month
1
2
3
4
5
6
7
8
9
10
11
12


Weekly seasonality (by year)
year
2013
2014
2015
2016
2017
60
55
50
45
40
35
0
1
2
3
4
5
6
dayofweek

Yearly sales for each store (5 years- 2013 to 2017)