Springboard—DSC Program

Capstone Project 2 Proposal

Demand Forecasting for a Healthcare System Supply Chain

By Laura Eshee

August 2020

Background

"The supply chain generally refers to the resources needed to deliver goods or services to a consumer. In healthcare, managing the supply chain is typically a very complex and fragmented process.

Healthcare supply chain management involves obtaining resources, managing supplies, and delivering goods and services to providers and patients. To complete the process, physical goods and information about medical products and services usually go through a number of independent stakeholders, including manufacturers, insurance companies, hospitals, providers, group purchasing organizations, and several regulatory agencies."[1]

The healthcare supply chain starts with the manufacturers of medical products, goes through the healthcare system's stocking and distribution process and ultimately to the provider's office, the hospital, or the administrative support areas of the system for usage.[1]

"Healthcare supply chain management is unique because each stakeholder has their own interests to protect. Different stages in the supply chain flow may be focused on their own goal. Providers may want to use a specific product because they were trained with it, whereas hospital executives aim to purchase the most affordable quality items.

Since supply chain goals are not always aligned within an organization, the healthcare supply chain management process can be inefficient and fragmented. Healthcare organizations must take into account numerous requests and viewpoints to settle on specific product budgets."[1]

According to an article on the Recycle Intelligence website titled '5 Ways to Improve Healthcare Supply Chain Management,' one way to improve health care supply chain management is to develop effective inventory management. Inventory levels must be managed effectively. Too much stock will result in high costs for storing and maintain it. Too little stock will result in not having enough to meet demand and having to pay a premium to get what is needed quickly. As a result, good forecasting for inventory levels is needed.[2]

Problem Statement

The client is the largest healthcare system in its state. It has eleven hospitals, five health parks, more than three hundred medical offices, nine cancer centers, fifty-five rehabilitation centers, three hospice facilities, twenty-one imaging centers and fifteen urgent care locations. It also has more than twenty-four thousand employees.

In 2018, there were 114,750 hospital admissions across the system. Additionally, it had $4.1 million in assets and received $3.2 million in revenue.

The Supply Chain department of this client consists of one distribution center, as well as, multiple storerooms at the facilities. There are approximately 3,600 items that are stocked at the distribution center. Currently, forecasting for how much of each item should be stocked is done by hand by the manager.

Dataset

The data will be obtained from the client's inventory database using Microsoft SQL Server. A two-year history of the products that have been issued from the distribution center to the facilities will be obtained.

Approach

The approach that will be used to model this problem will be a time series forecasting model. Baseline models will be built using algorithms and features to be defined. Once the performance characteristics of these models are established, other models and/or tuning approaches will also be attempted, and all models built will be compared with respect to performance metrics that align with the business problem. In general, the classical phases associate with many data science problems will be implemented, namely: data acquisition and wrangling, storytelling and applications of inferential statistics, model building and ranking.

Deliverables

As required, all Jupyter notebooks, a written final report, and a presentation slide deck have been submitted.

Data Wrangling

The data was obtained from the client's inventory database using Microsoft SQL Server. A two-year history of the products that have been issued from the distribution center to the facilities was obtained.

The date column (TRANS_DATE) was imported as an object data type. Therefore, the first action was to convert it to a datetime data type. Next, the client requested that the unit of measure (UOM) used for forecasting be the tracked UOM. The tracked UOM is denoted in the data by an X in one of the BUYFL columns. Therefore, the transaction UOM was converted to the tracked UOM using the BUYFL columns.

The data wrangling for this data set was fairly simple. First, the date column data type was converted from object to datetime. Then, the transaction UOM was converted to the tracked UOM and the tracked quantity was computed. The data set was then saved to a csv file. It is now ready for the next step.

Data Story

With the data wrangling completed, it is time to see what story the data tells. The following metrics were calculated on the data set:

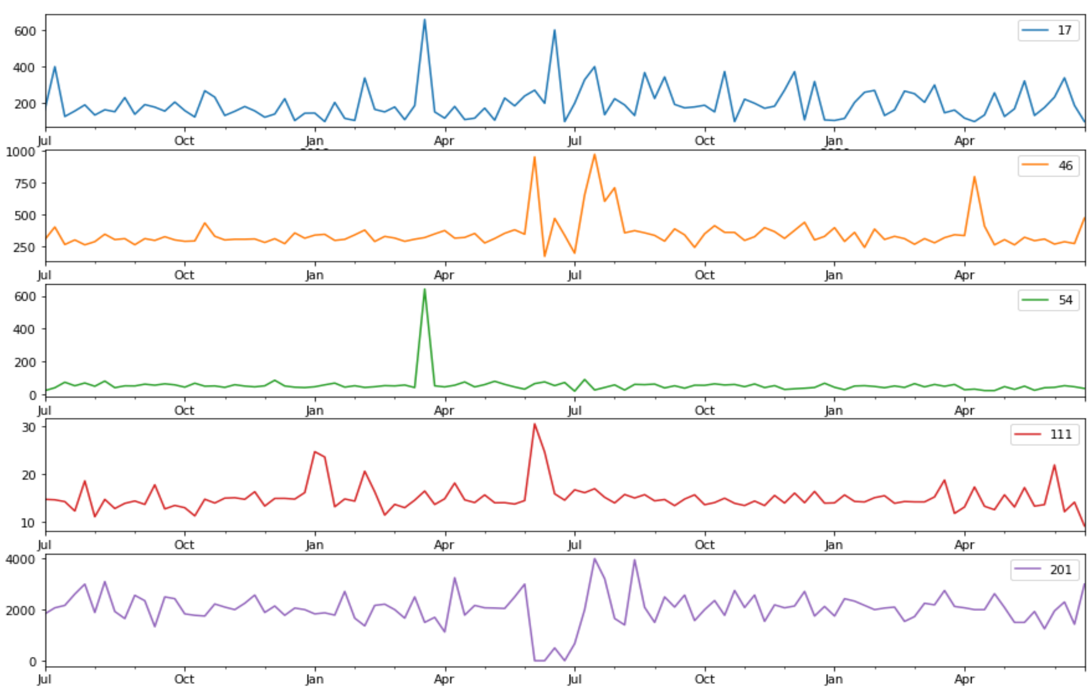
- Weekly forecast: the client asked for the forecasting to be done weekly. Therefore, the week starting date for each Trans Date was calculated. Then, the data set is checked to determine if there are any missing values.

- Missing values: it was found that several of the conversion columns were missing values, since not every item has the same number of UOMs. This finding was expected.

- Number of orders for each item: the number of orders for each item was determined. The item with the highest number of orders was Item #932, UNDERPAD INCONT30X36IN MOD ABS, with 24,728 orders, and the item with the least number of orders was Item #89, STENT PANCR 5-5 W INTNL, with 89 orders.

- Missing dates: whether there are missing dates for each item was determined. It was found that forty-five of the items were not ordered during one or more weeks in the two-year time period. It was decided to put zero for the quantity during the weeks where the items were not ordered.

- Plotting: the data frame was now reshaped so that time plots and box plots could be created for each item. A sample of the time plots and box plots are included here. To see the plots for all of the items, go to https://github.com/lookingglass01/springboard/blob/master/CapstoneProjects/CP2/code/Demand%20Forecasting%20Data%20Story.ipynb.



In summary, this data set has 2 years' worth of orders for 92 items in the warehouse. The data was sorted by item and week starting date. Then, it was inspected for missing dates. The data was reshaped so that it could be plotted with time plots and box plots, and the values for the missing dates were filled in with zeros. The final data set was saved so that it will be ready for the next step in the project, which in inferential statistics.

Hypothesis Testing

Next, hypothesis testing was performed on the data set. It was decided to test the mean orders of items in the same class to see if the observed difference in the means is statistically significant. The results are as follows:

- The difference in the means of the quantities of 2 different catheters were analyzed:
  - Ho: mu1 = mu2, Ha: mu1 != mu2
  - Outliers were removed from both items
  - Results:
    - Statistic: -37.572
    - P-value: 0.0
    - Reject Ho. The difference in the means is statistically significant.
- The difference in the means of the quantities of 2 different gowns were analyzed:
  - Ho: mu1 = mu2, Ha: mu1 != mu2
  - Outliers were removed from both items
  - Results:
    - Statistic: -19.149
    - P-value: 0.0
    - Reject Ho. The difference in the means is statistically significant.
- The difference in the means of the quantities of 2 different hypodermic syringes were analyzed:
  - Ho: mu1 = mu2, Ha: mu1 != mu2
  - No outliers were found for these items
  - Results:
    - Statistic: -16.241
    - P-value: 0.0
    - Reject Ho. The difference in the means is statistically significant.

Modeling

Once the data story was completed, a first algorithm was built for selected products. It was decided to model two products using the Auto-ARIMA and Prophet. The products chosen were Item #17 - APPLICATOR FBRTP 6IN STRL 2 P and Item #130173 - SYRINGE HYPODERM 3ML 23GX1IN. Item 17 was chosen because its time plot showed much variation, outliers and skewness, all of which would be a challenge to forecast. Item 130173 was chosen because it was more normally distributed with no outliers and would be easier to forecast. The intent was to compare the forecasts for the two items to see how the algorithm s performed with more challenging forecasts and easier forecasts.

Auto-ARIMA

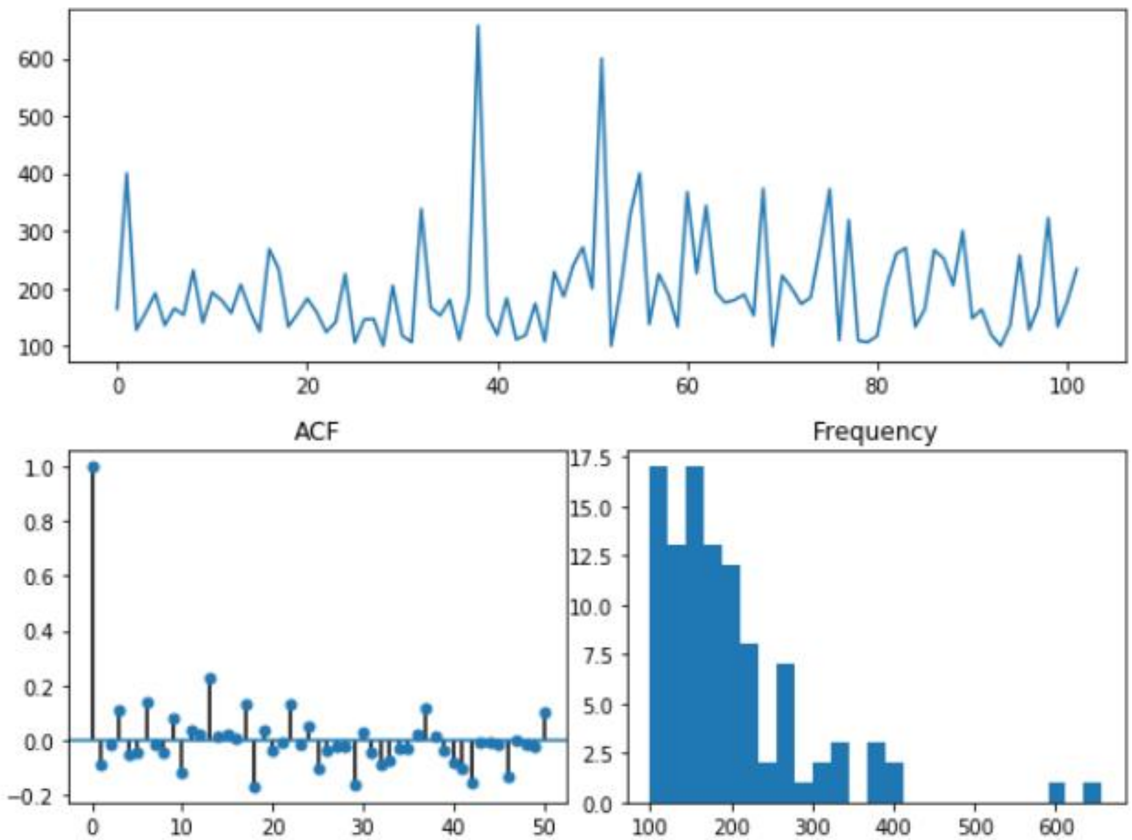
Auto-ARIMA was the first algorithm that was used. ARIMA stands for 'Auto Regressive Integrated Moving Average'. It uses the product's history, lags and lagged forecast errors to forecast future quantities. It uses the variables p, d and q to calculate the forecast.

- p—the order of the auto regressive term.

- d—the order of the moving average term.
- q—the number of differencing required to make the times series stationary, which is a process of subtracting the previous value from the current value and using the result to smooth the data.[3]

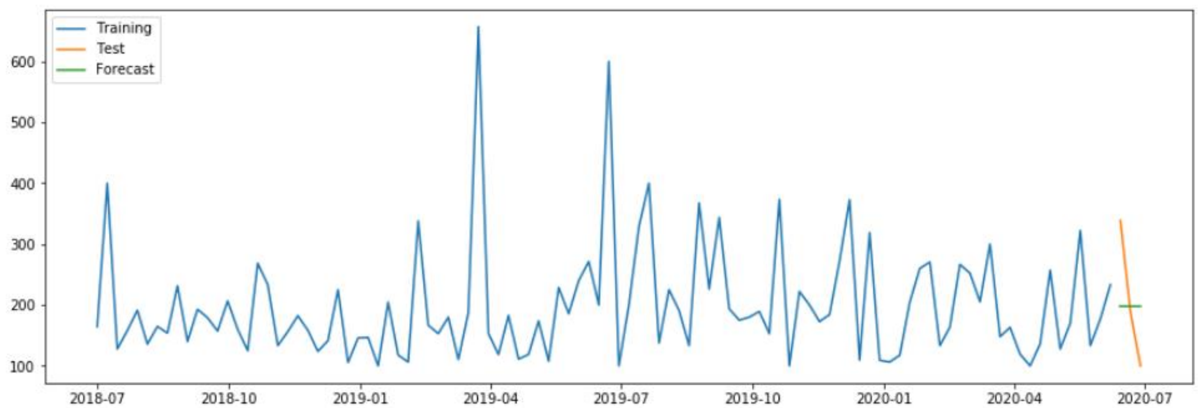
The Auto-ARIMA algorithm calculates p, d and q automatically from the data.

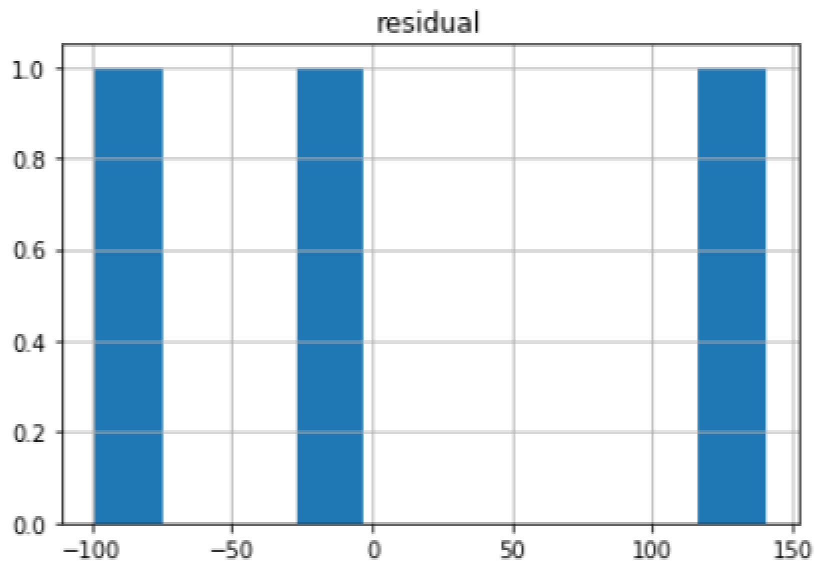
A time plot, ACF plot and Frequency plot were done for the Item 17. The time plot shows the variation of order quantities over the last 2 years. The ACF plot shows the residuals. The frequency plot shows the outliers and skewness.



Since this was the first time the Auto-ARIMA algorithm had been run on the data, it was decided to run it first with raw data. The predicted quantities are shown in the table below. Also, a plot of the training and test sets, and the predictions, as well as a plot of the residuals are show below.

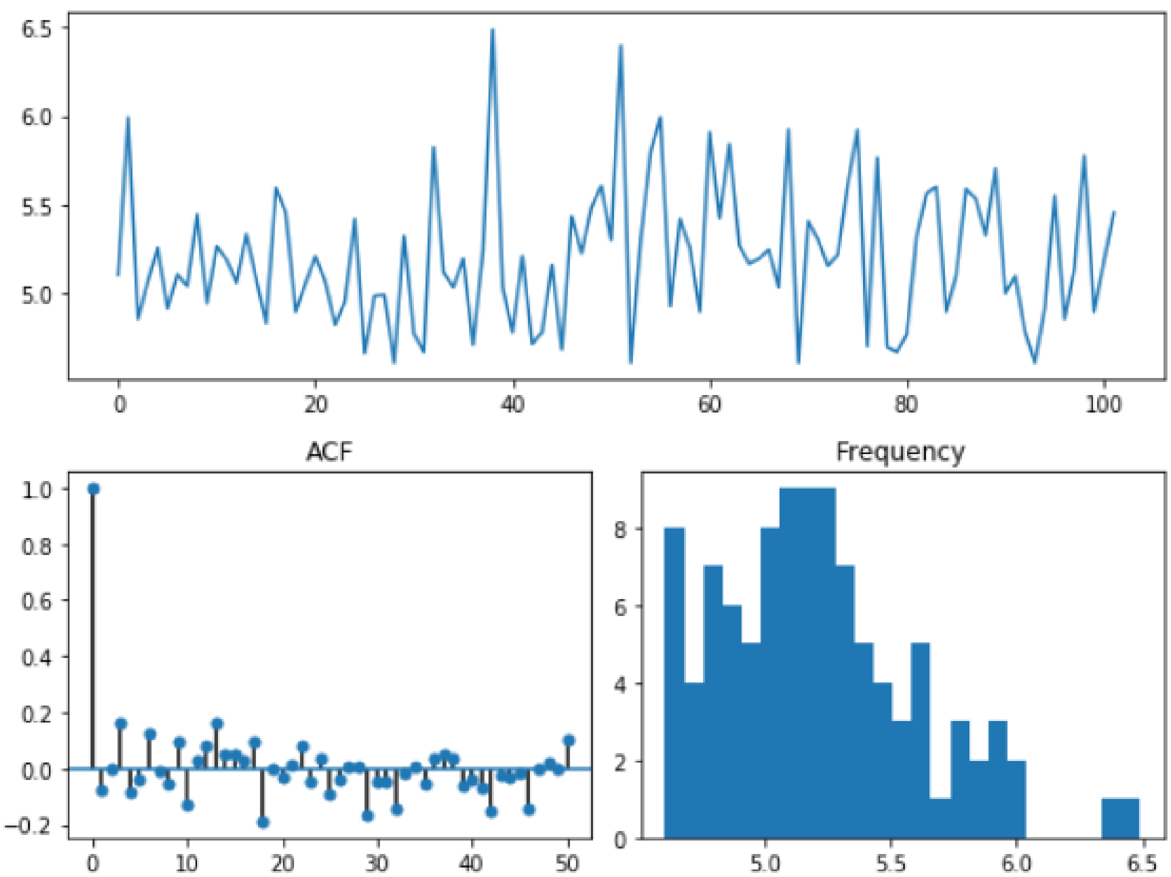
	Predicted_Quantity
Week_Start	
2020-06-14	198.683041
2020-06-21	198.683041
2020-06-28	198.683041



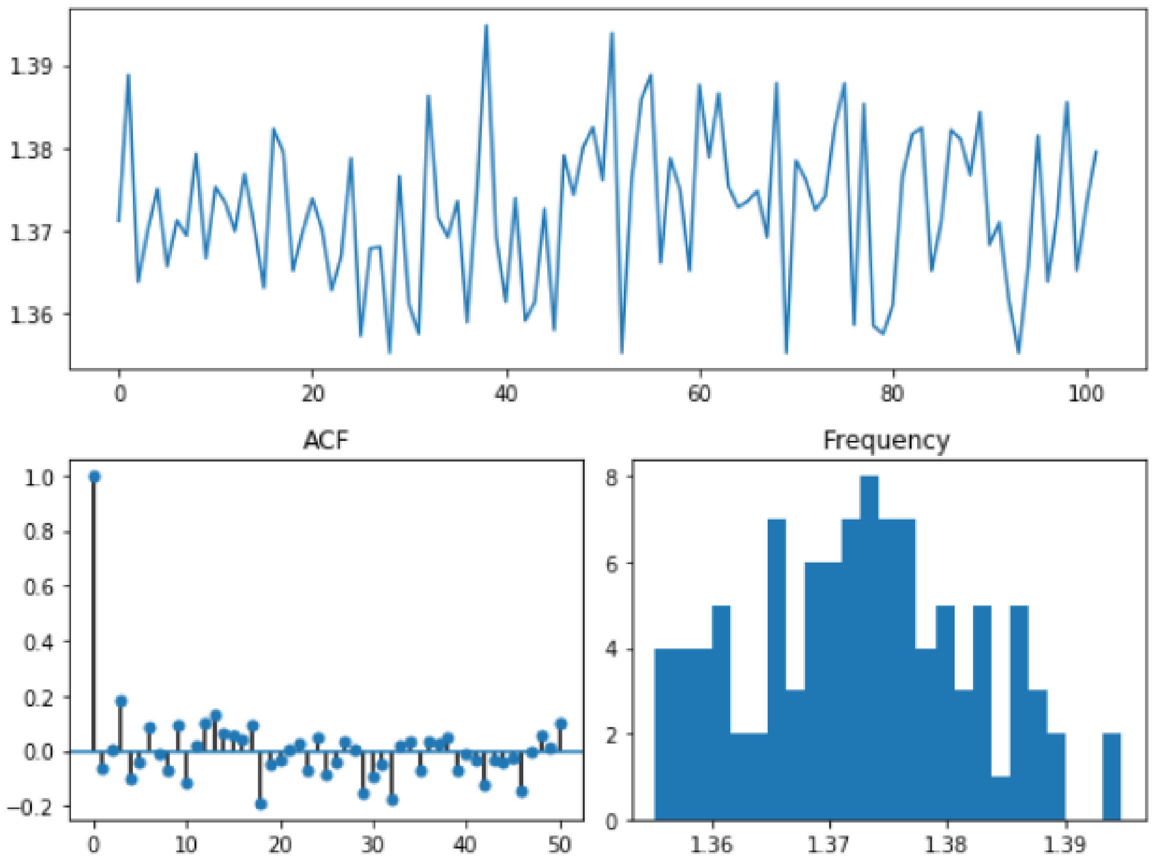


It is easy to see that the residuals do not cluster around zero as well as would be preferred. Therefore, the data will be transformed by taking its log and by using the Box-Cox transformation to see which will transform the data into a more normal distribution. The plots are below:

Log Transformation

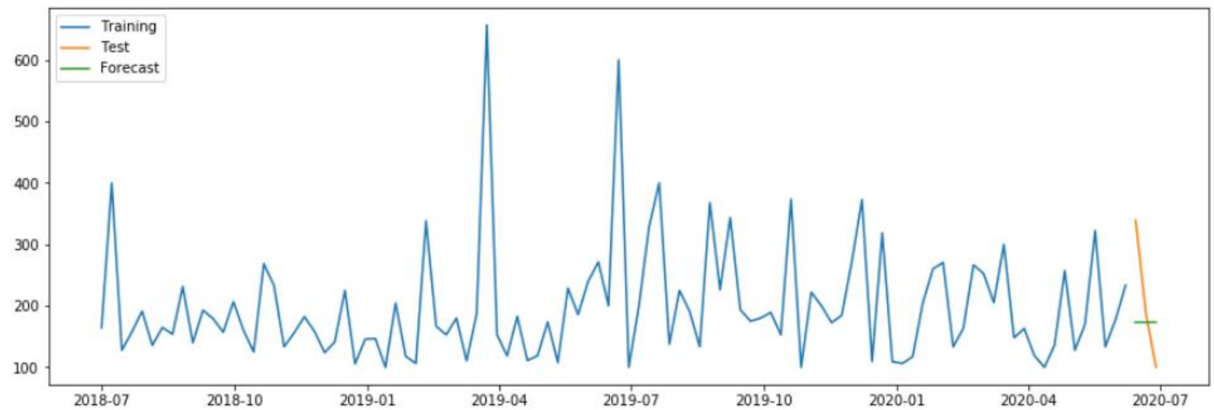


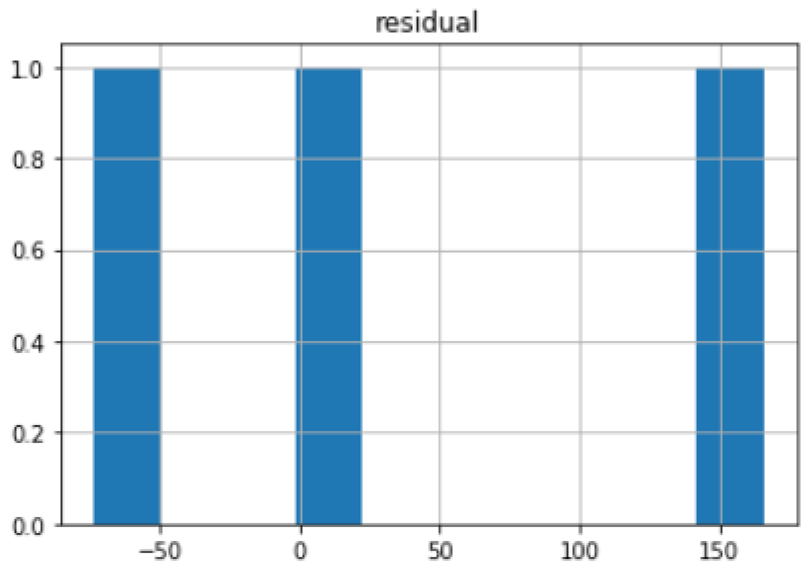
```
Box-Cox Transformation
```



Since the Box-Cox transformation results in a more normal distribution, it is the one that will be used to fit the algorithm for Item 17. The results are as follows:

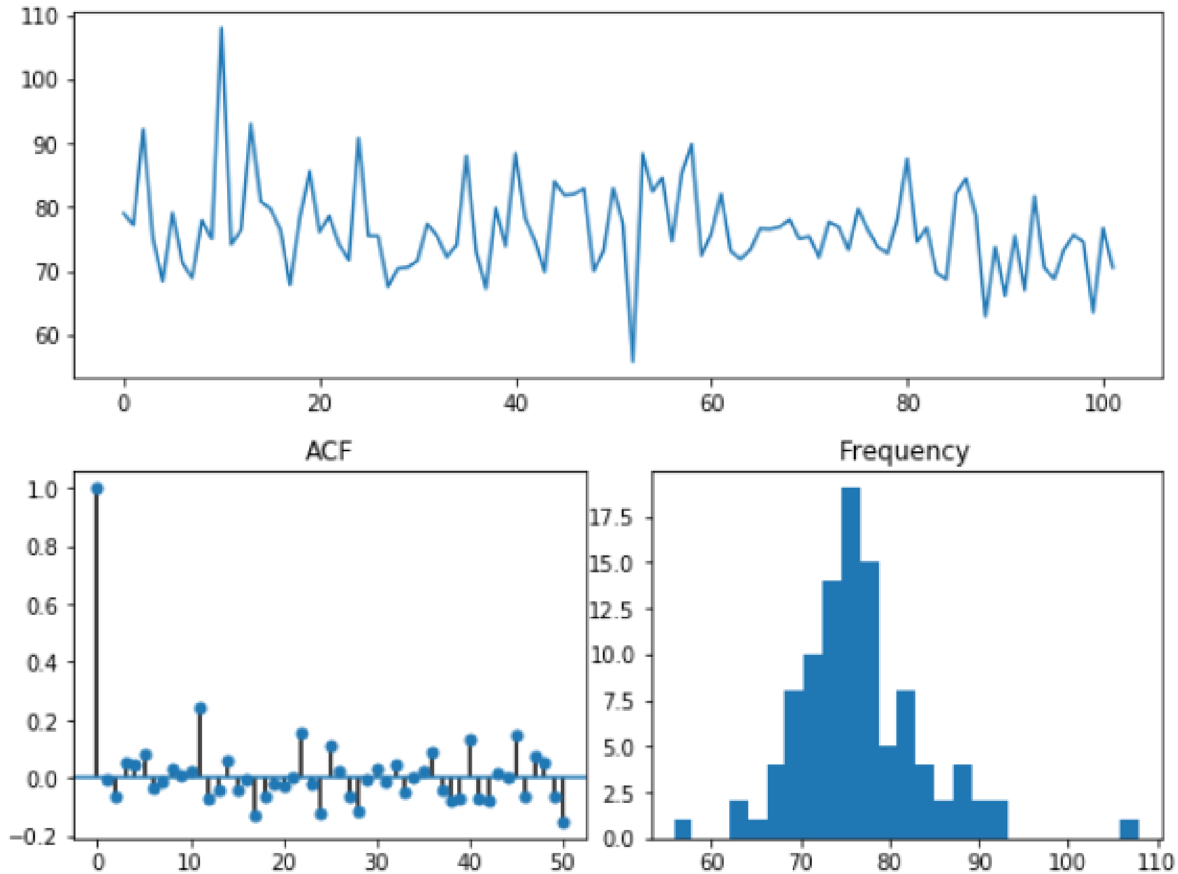
Week_Start	Predicted_Quantity
2020-06-14	173.237237
2020-06-21	173.237237
2020-06-28	173.237237





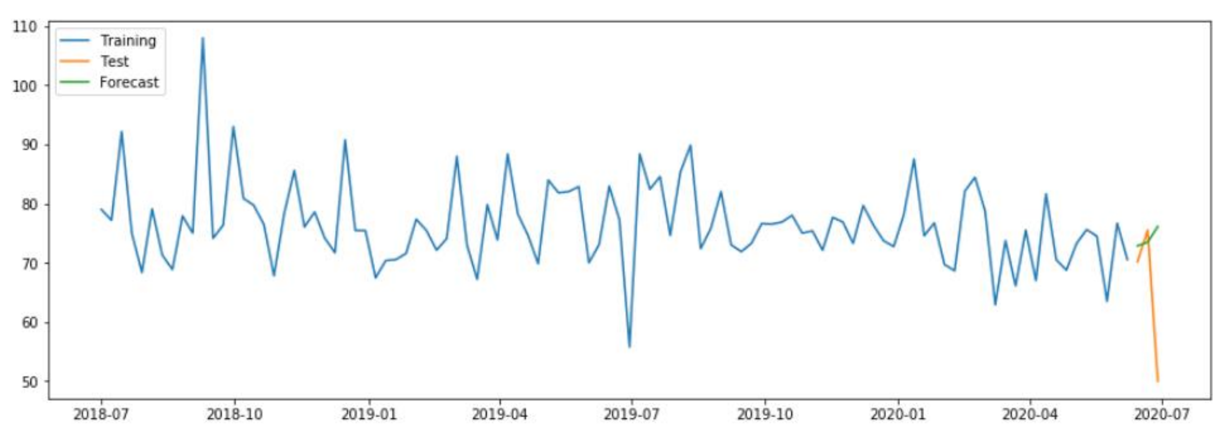
Using the Box-Cox transformed data results in a lower quantity forecast. Also, the residual plot did not improve very much.

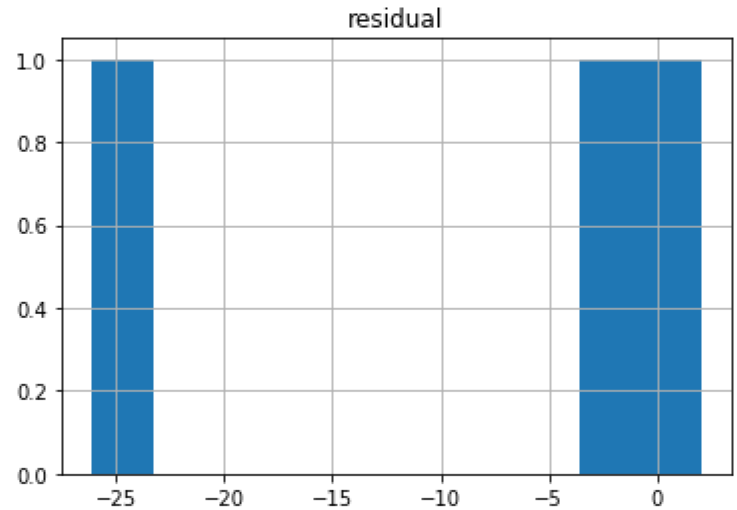
Next, a time plot, ACF plot and Frequency plot were done for the Item 130173.



The frequency plot shows that this item is normally distributed with one outlier. Therefore, no transformation will be needed. The results of the fitted model are below:

Week_Start	
2020-06-14	72.844240
2020-06-21	73.479548
2020-06-28	76.113295





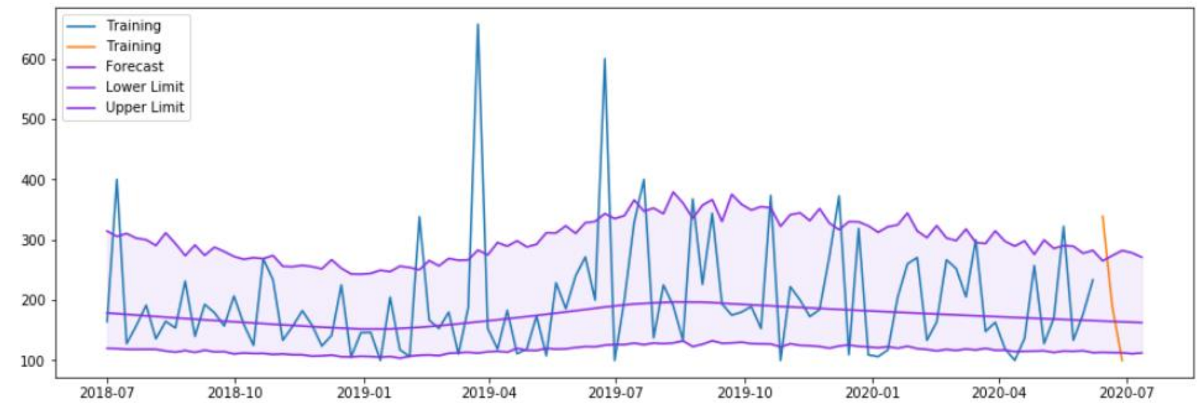
The results for this forecast are better than for Item 17. The residual plot shows that the differences between the actual vs. the predicted values are reasonable. In the next step, both products will be forecasted using Prophet to see if the results are different.

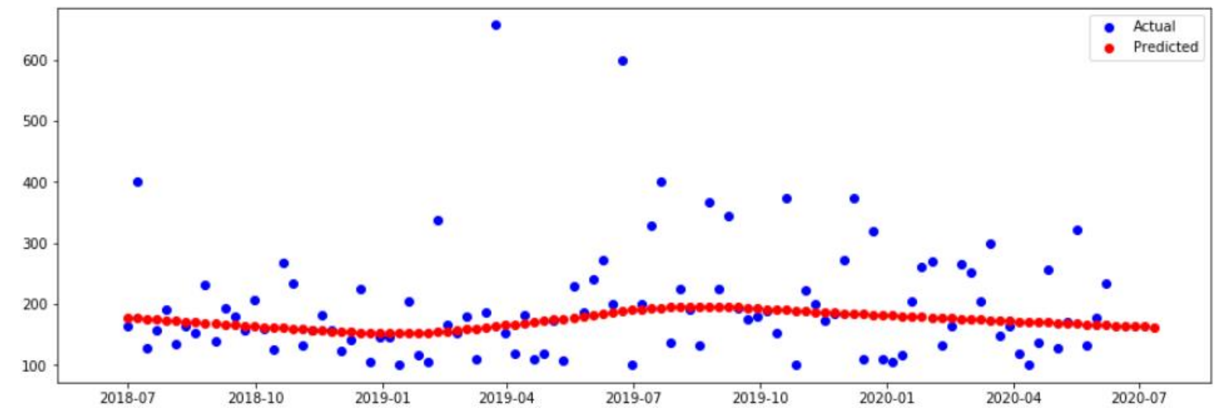
Prophet

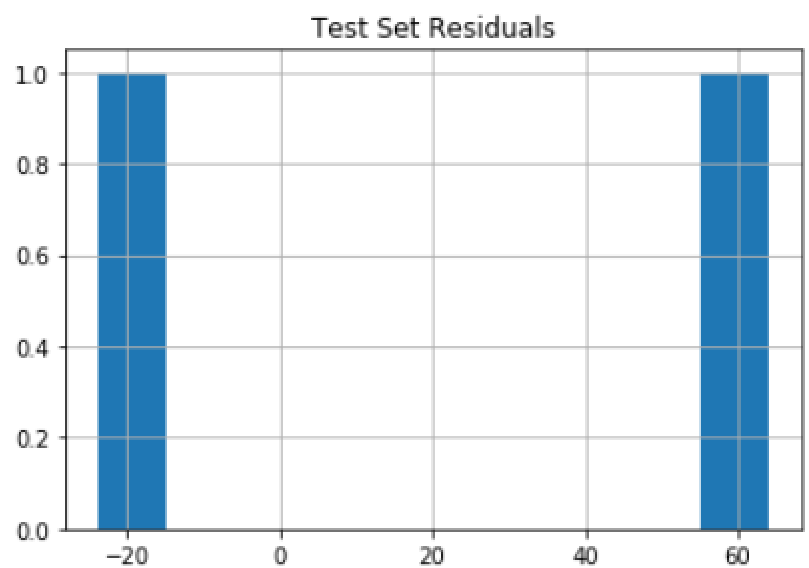
Prophet is an algorithm for forecasting time series data based on an additive algorithm, where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. Prophet is open source software released by Facebook's Core Data Science team.[4]

As with the Auto-ARIMA algorithm, Item 17 was transformed using Box-Cox and fitted to the Prophet algorithm. The results are below:

	ds	yhat	yhat_lower	yhat_upper
102	2020-06-14	165.206126	112.693411	289.951335
103	2020-06-21	164.564483	113.443048	283.172374
104	2020-06-28	163.927090	111.060929	281.858653
105	2020-07-05	163.293906	110.240731	266.603461
106	2020-07-12	162.664892	111.878927	267.716158



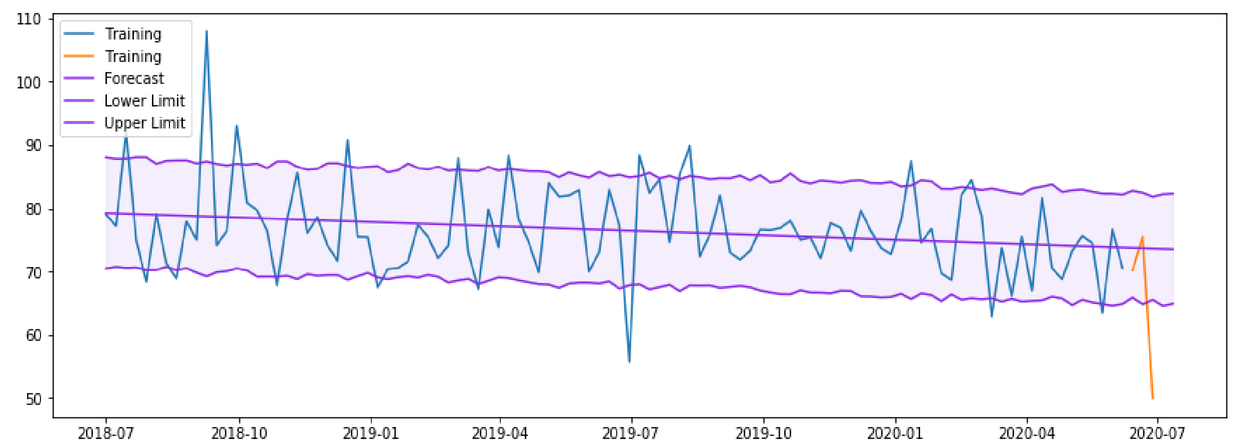


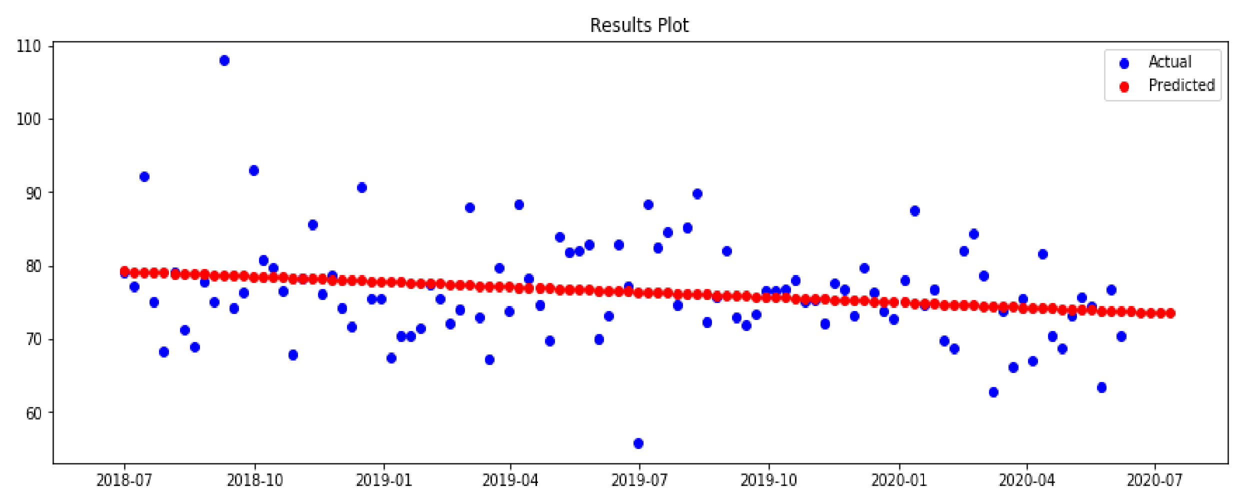


In this model, yhat is the forecasted quantity. Also, yhat_lower and yhat_upper are the upper and lower limits of the forecast, respectively. These limits represent a 95% confidence interval for the forecast. In other words, there is a 95% certainty that the correct forecast lies between these limits. A graph that includes the limits is included above. The limits and the area they represent are shaded in purple. Also, it can be seen in the Residuals plot that the residuals are closer to zero with this model.

Next, Item 130173 was fitted to the Prophet algorithm with the following results:

	ds	yhat	yhat_lower	yhat_upper
102	2020-06-14	73.736156	65.418828	82.937484
103	2020-06-21	73.682172	64918795	82032048
104	2020-06-28	73.628188	65.181435	82.341897
105	2020-07-05	73.574203	64.984683	82.537769
103	2020-07-12	73.520219	64.941713	82.540307





Once again, the residuals are much closer to zero with this model.

Below are tables that summarize the findings for each forecasting model built, for each item:

Auto-ARIMA

Date	Item #17 Forecast	Item #17 Residual	Item #130173 Forecast	Item #130173 Residual
2020-06-14	173	165.65	73	-2.67
2020-06-21	173	14.99	73	2.03
2020-06-28	173	-73.24	76	-26.11

Item	R^2	MAE	RSME	MAPE
17	-0.1318	84.63	104.92	43.36
130173	-0.9143	10.27	15.20	18.76

Prophet

Date	Item #17 Forecast	Item #17 Residual	Item #130173 Forecast	Item #130173 Residual
2020-06-14	165	-10.42	74	-2.82
2020-06-21	165	-67.48	74	3.25
2020-06-28	164	-173.68	74	3.56
2020-07-05	163	-23.68	74	-1.82
2020-07-12	163	63.93	74	23.63

Item	R^2	MAE	RMSE	MAPE
17	-1.8826	135.36	0.0025	0.0025
130173	-0.5861	9.67	2.75	2.75

The forecasts for the two algorithms are similar. Also, the r^2 and mean absolute errors seem to suggest that the Auto-ARIMA algorithm might be a better algorithm. However, the differences in the RMSE and MAPE indicate that the Prophet algorithm is better. Given that the Prophet algorithm is more robust and more easily provides upper and lower limits for its forecast, it has been decided to use this algorithm for future forecasting this data.

Conclusion and Recommendations

The purpose of this project was to help the client forecast product demand for its distribution center.

The process used to accomplish the purpose was to obtain a 2-year history of supply orders for two products from the client's database, prepare the data set for analysis, create plots to show the story the data told, perform inference testing and build forecasting models, using two different algorithms, to predict the quantities of the items to be kept in the warehouse.

The results show initial evidence that good performing forecasting models can be created using Prophet.

It is recommended that the development of the project move into the next phase which will be to algorithm and predict forecasts for the top 92 items in the client's distribution center. The goal of the project is to provide automated forecasting for all the items in the distribution center that can reasonably be modeled.

Future Work

Future work for this project includes the following:

- Further develop the algorithmic process so that multiple items can be modeled with one process rather than individually.
- Roll the process out to all items that have enough history and frequency of orders to be modeled effectively.

Footnotes:

1. 'Exploring the Role of Supply Chain Management in Healthcare', *Recycle Intelligence*, August 5, 2016, https://revcycleintelligence.com/news/exploring-the-role-of-supply-chain-management-in-healthcare (accessed June 16, 2020)

2. '5 Ways to Improve Healthcare Supply Chain Management', *Recycle Intelligence*, May 19, 2016, https://revcycleintelligence.com/news/5-ways-to-improve-healthcare-supply-chain-management, (accessed June 16, 2020)

3. 'ARIMA Model – Complete Guide to Time Series Forecasting in Python', *Machine Learning Plus*, https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/, (accessed July 29, 2020)

4. 'Forecasting at scale', *Facebook*, https://facebook.github.io/prophet/#:~:text=Prophet%20is%20a%20forecasting%20procedure%20implemented%20in%20R%20and%20Python.,-It%20is%20fast&text=Prophet%20is%20a%20procedure%20for,daily%20seasonality%2C%20plus%20holiday%20effects., (accessed July 30, 2020).