<u>Boeing Stock Volatility</u>

Intro

The project's intent is to create a model that makes useful predictions about the movement of Boeing's stock. I will be using time-series analysis to predict the 10-day historical volatility of Boeing's stock. If I can actually predict the volatility of the stock, I can use this in a number of ways to help organizations. I would be able to help organizations with useful trading strategies based off these predictions. Buying or selling options can be ways to profit from periods of predicted high or low volatility. Managers can use these insights to increase or decrease their exposure based off the volatility. While this can also lead to profitable trading strategies it can also lead to meaningful drawbacks in existing portfolios if volatility is high. I am extremely interested in Boeing due to the climate it has been in the last couple of years. Obviously with Covid, less people are flying causing the stock to move all over the place. The other big thing is the 737 MAX, we saw the stock tank after the 2 accidents but recently just heard good news about it causing the stock to go back up.

What is volatility?

Volatility is the degree of variation of a trading prices series over time, usually measured by the standard deviation of logarithmic returns. Historical volatility (HV) is the volatility experienced by the underlying stock, stated in terms of annualized standard deviation as a percentage of the stock price. Historical volatility is helpful in comparing the volatility of one stock with that of another stock or to the stock itself over a period of time.

For example, a stock that has a 15 historical volatility is less volatile than a stock with a 25 historical volatility. Additionally, a stock with a historical volatility of 35 is now more volatile than it was when its historical volatility was, say, 20. In contrast to historical volatility, which looks at actual asset prices in the past, implied volatility (IV) looks ahead.

Implied volatility is often interpreted as the market's expectation for the future volatility of a stock. Implied volatility can be derived from the price of an option. Specifically, implied volatility is the expected future volatility of the stock that is implied by the price of the stock's options.

For example, the market (collectively) expects a stock that has a 15 implied volatility to be less volatile than a stock with a 30 implied volatility. The implied volatility of an asset can also be compared with what it was in the past. If a stock has an implied volatility of 40 compared with a 20 implied volatility, say, a month ago, the market now considers the stock to be more volatile.

Implied volatility and historical volatility are studied using a volatility chart. A volatility chart tracks the implied and historical volatility over time in graphical form. It is a helpful visual aid that makes it easy to compare implied volatility and historical volatility.

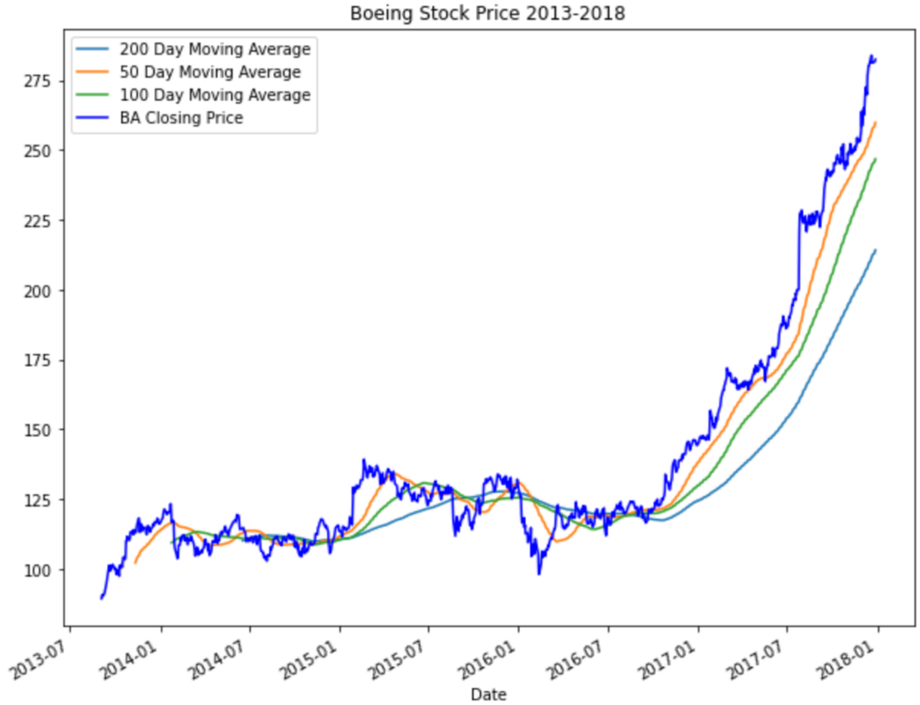
Data Wrangling

I found my data about Boeing's stock that included closing price, high, low, daily volume and several other features. This also provided data on options as well. Quandl.com has free data on stocks traded on exchanges. Quandl is a marketplace for financial, economic, and alternative data delivered in modern formats for today's analysts. The options data that I pulled I paid for using a premium service. I also pulled some data from Yahoo finance just to see the current climate of the stock. The first thing I did was change the date columns to its correct type which was datetime. Every other column was a float. I wanted to make sure there were no null entries in the stock data and there were not. In the options data, the data provided historical and implied volatility at during different maturity length options. In Quandl, the stock data only goes back to 2013 and the option data goes back to 2002. I only used information where they overlapped. There were some null values in the options dataset, which I then cleaned up to have a proper dataset. I then just filtered data from Jan 1, 2013 – December 31, 2017. I then merged the stock and options data into one creating 76 columns and the date as my index. Lastly, I put the data into a pickle format.

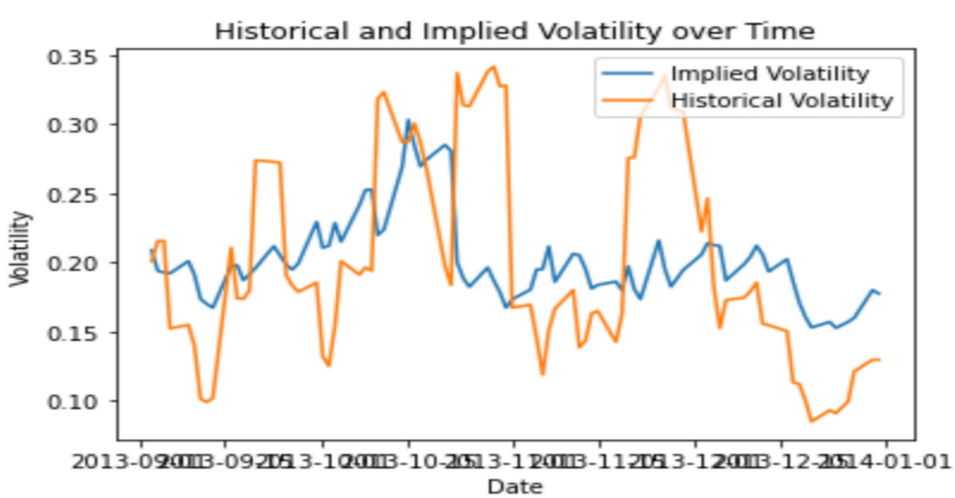
Exploratory Data Analysis

This step required me to begin exploring the datasets through visualizations. My goal was to find noticeable patterns in the historical volatility of Boeing's stock that could be visualized. I also wanted to find the relationship between historical volatility and other features within the dataset.

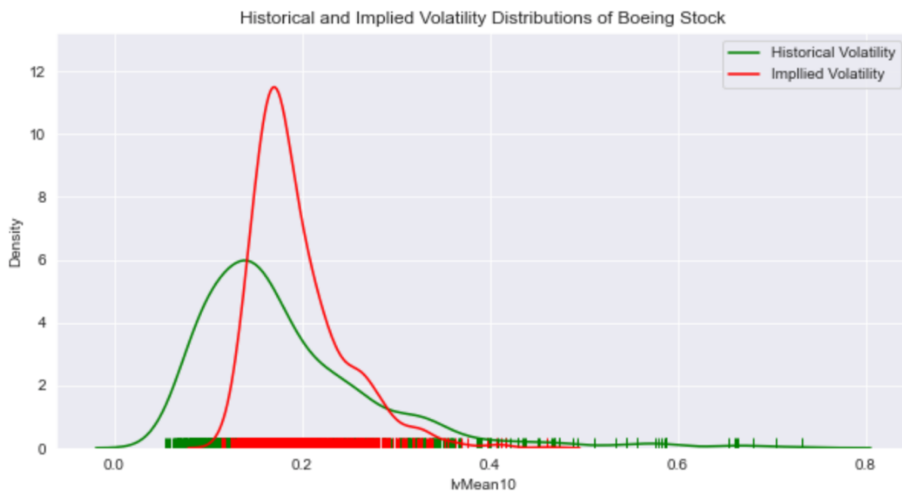
I had to unpack the pickled data from the Data Wrangling to make sure that it was okay. I plotted the range for a fixed period, March 2017, to the end of the dataset, to get an idea of the daily ranges of the stock. I also included the 50, 100, and 200 day moving averages of the stock prices, knowing that these features are often used to identify patters in the price. Here you can see the chart below.



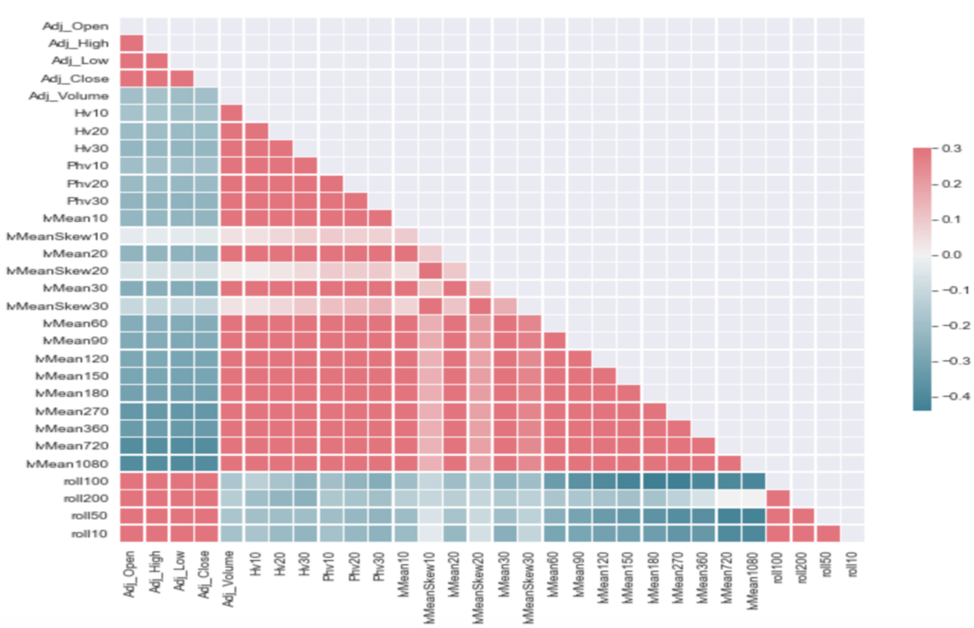
I then wanted to see a chart that had Historical Volatility in respect to the Implied Volatility of the options. The next chart shows a volatility of the year 2013. You can see here the Historical Volatility would have massive spikes from time to time compared to Implied.



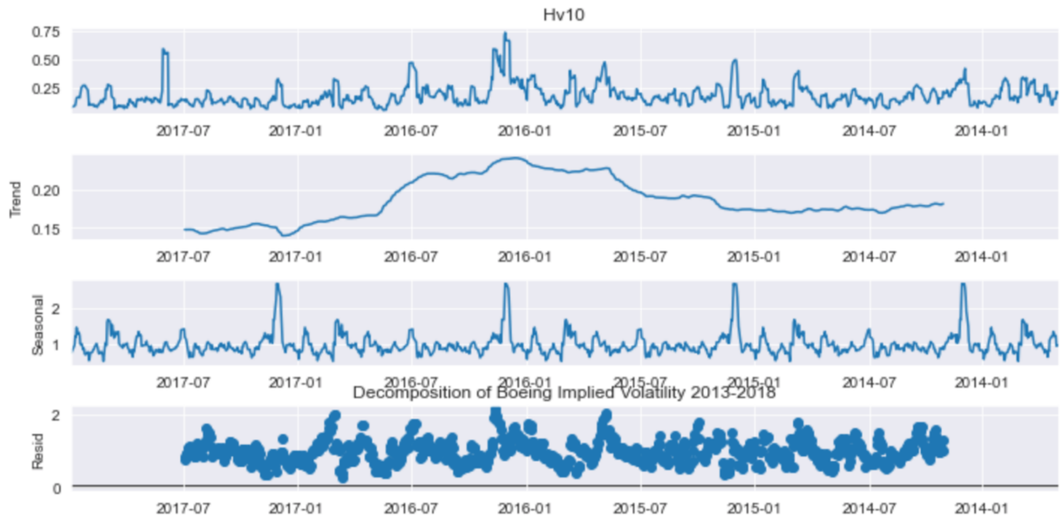
I then plotted the two volatilities on a histogram. You will see here that the historical volatility spikes above the implied volatility in tail events. Usually, however, historical volatility is slightly lower than implied volatility. The two have similar means but you can see below the curves on top of each other are very different.



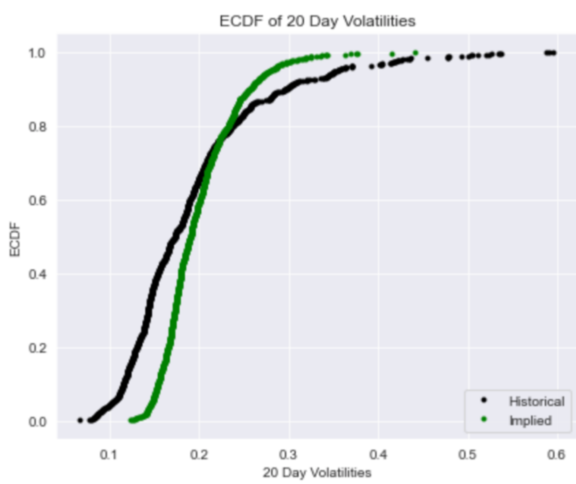
I then wanted to build a heatmap to visualize the Pearson's correlation coefficients of each feature related to another. The Historical volatility to stock prices has no correlation. Stocks that trend higher have a lower volatility. Historical volatility has positive correlation to implied volatility. The higher volume correlates to greater historical volatility.

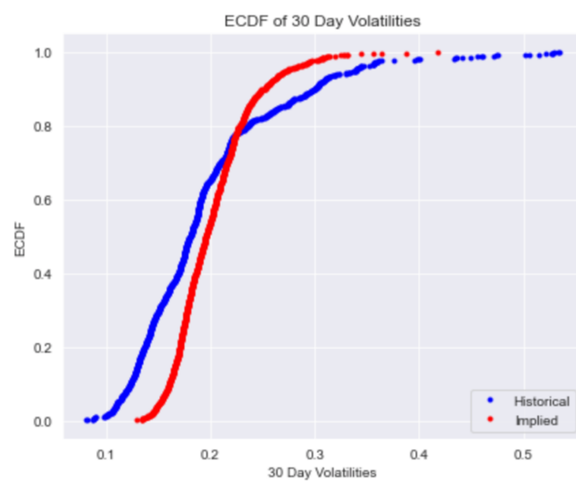


I then used the statsmodels seasonal decomposition model examine Boeing stock prices. This chart shows observed, trends, seasonal, and residual charts.

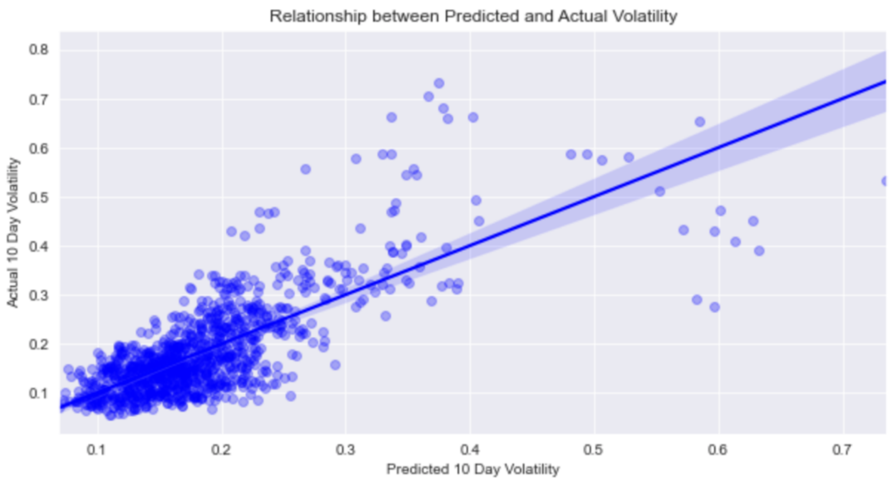


The seasonal trend for implied volatility is higher at the end of the calendar year towards Christmas and then goes down after the New year.





An ECDF is an estimator of the Cumulative Distribution Function. The ECDF essentially allows you to plot a feature of your data in order from least to greatest and see the whole feature as if is distributed across the data set. This is the empirical distribution function of the 20- and 30-day volatilities. There is a greater amount of days where the implied volatility is below historical volatility. Given time, historical volatilities will regress back to normal and these are just outliers.



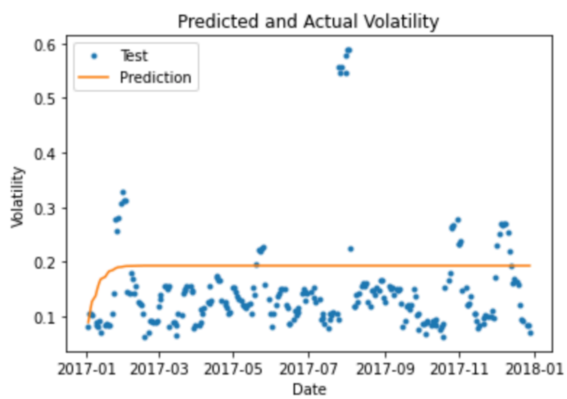
This is a regression plot of the predicted 10-day volatility vs the actual 10-day volatility. When the actual volatility goes above 30%, the model has trouble predicting the actual volatility. There are however are lot of prediction in the 15 - 30% range that are higher than actual volatility.

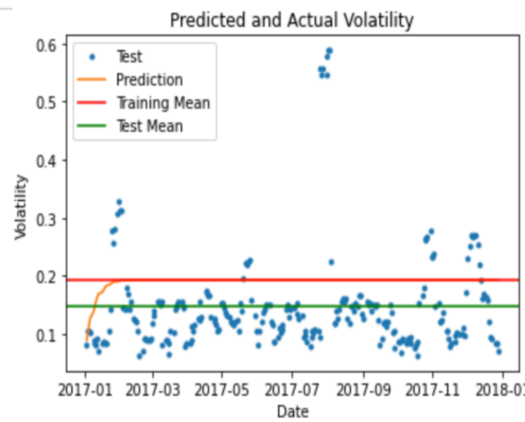
Machine Learning

This last phase required model-building to predict the historical volatility. I focused on ARIMA which is Autoregressive Integrated Moving Average and then on the Prophet model which is built by Facebook. I determined that there were seasonal and trend components to the movement of the Boeing stock. Specifically, I would use auto-ARIMA, a model with built-in parameter tuning. This could help find the best parameters for the model using a methodology similar to grid search.

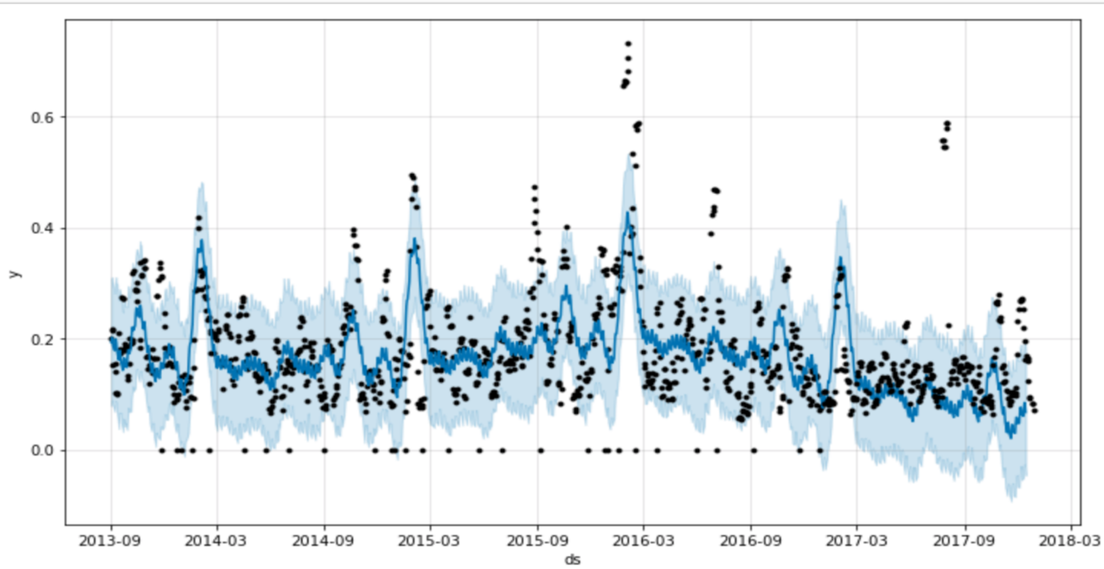
I ran two ordinary least squares models using the statsmodel package. One that focused on historical volatility and the other on multiple features including implied volatility, volume, and skew. That second model that had more features outperformed the more basic model. This made me believe that additional features might help us predict volatility going forward.

I started with a baseline using an auto-ARIMA model to predict Boeing's 10-day historical volatility. The initial model seemed to start off very close to the actual volatility, but it then quickly gravitated parallel to the training mean (2013-2016) where is stayed through most of 2017. The straight line brought an issue that would be in several of the models. A forecast building a full year's data based on data for the previous 4 years , rather than taking into account the most recent measure the stock's volatility. It was tracking well for a few days, but quickly degraded.

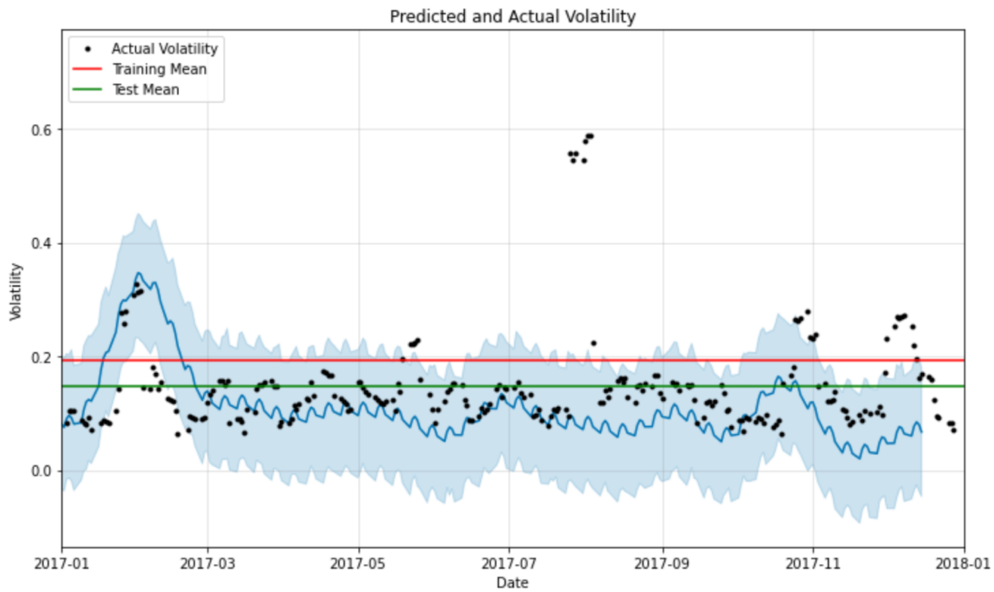




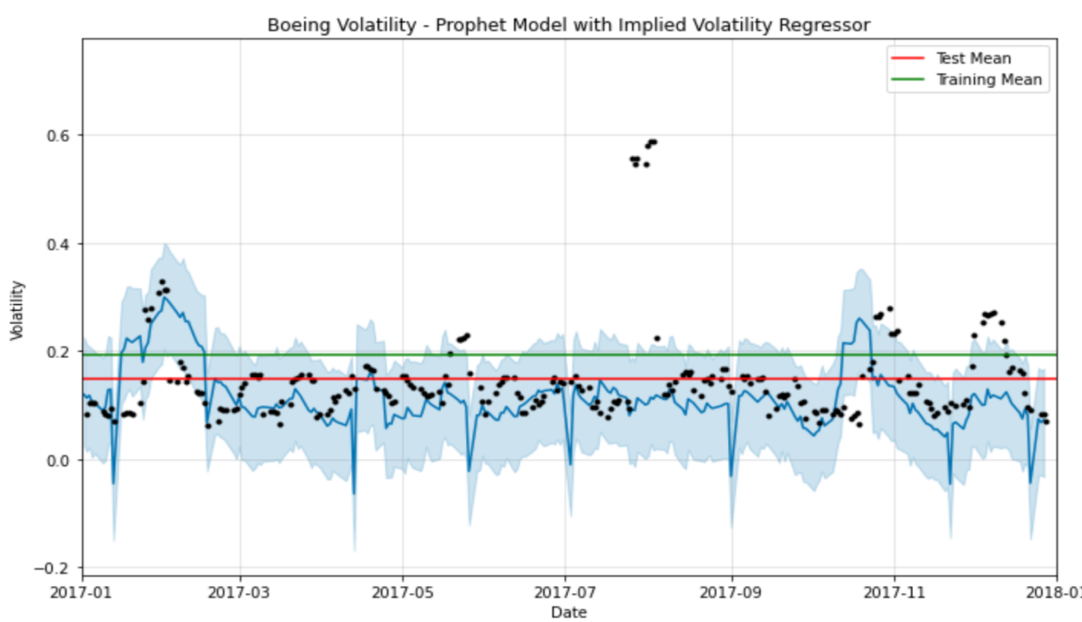
The next model that I used was the Facebook Prophet model. Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.



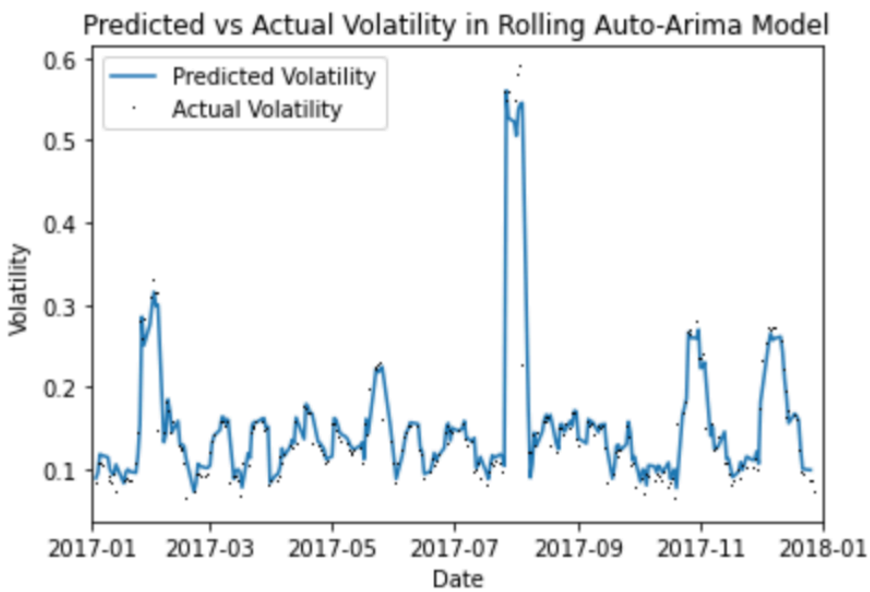
This model showed significant improvement in terms of showing the market trend volatility. It also showed improved in the root mean squared error and r-squared compared with the baseline model.



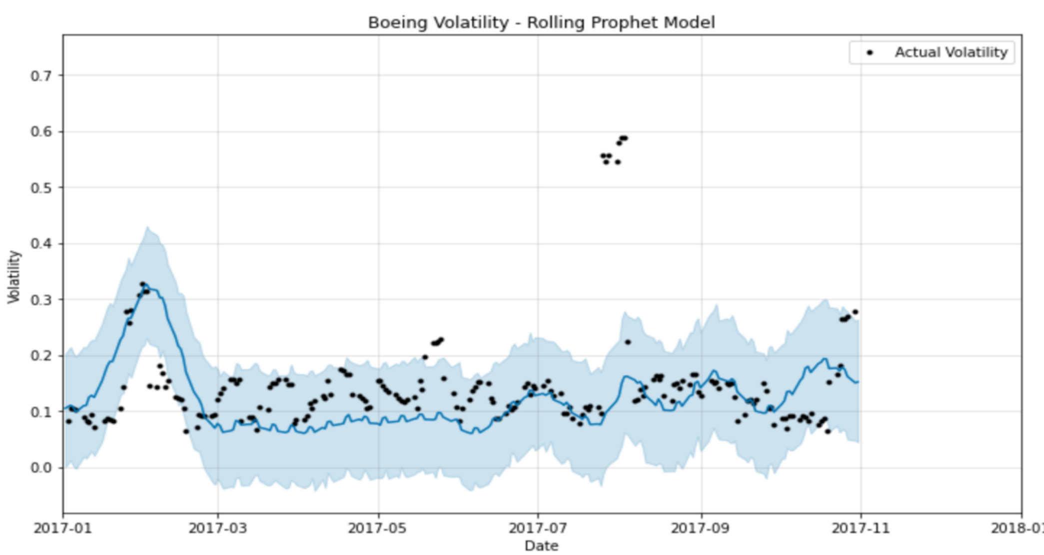
Now that I see that the prophet model outperforms the baseline, I believed that there were ways I can improve the prophet even further. I can do this by adding additional features. The heatmap showed a reasonable correlation between implied volatility and historical volatility, I thought that implied volatility would be a good feature to add. This was the most accurate model so far with RMSE and r-squared but the r-squared is still negative.



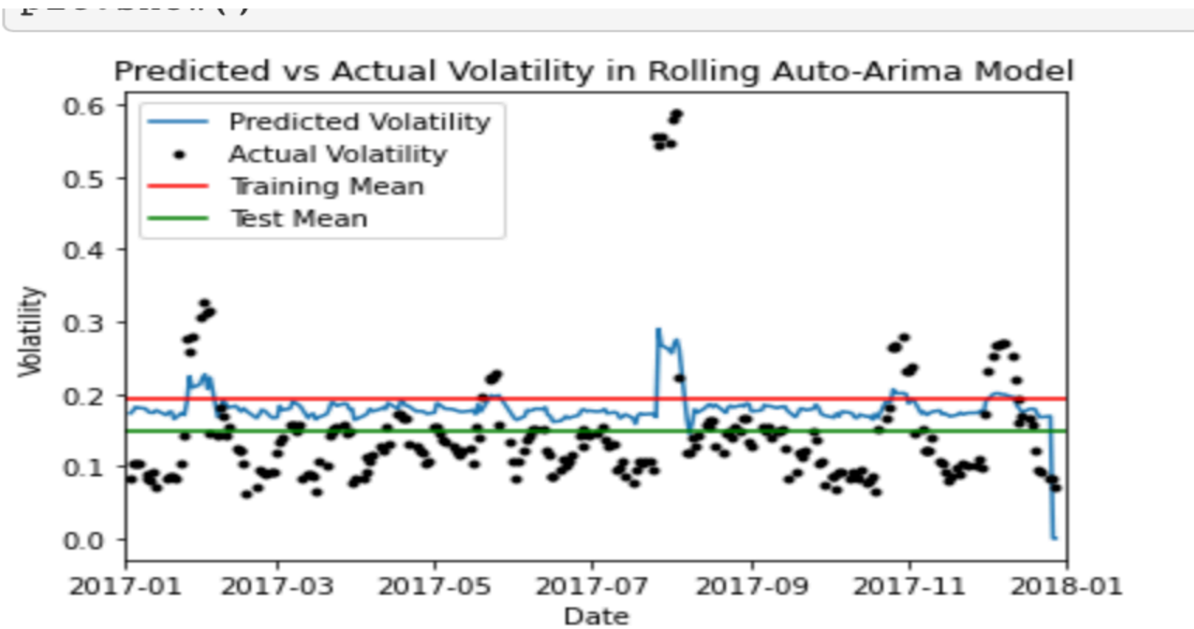
The models did a really good job building in seasonal and trend patterns but did not apply as much weight to recent values as I thought would be appropriate. The models essentially used data from 2013 through 2016 to build a forecast for the entirety of 2017. To try to fix this, I fitted the model on all the data up to the day of the prediction. I built a loop that would represent a new auto-Arima model for each day and output the result of a new dataframe.



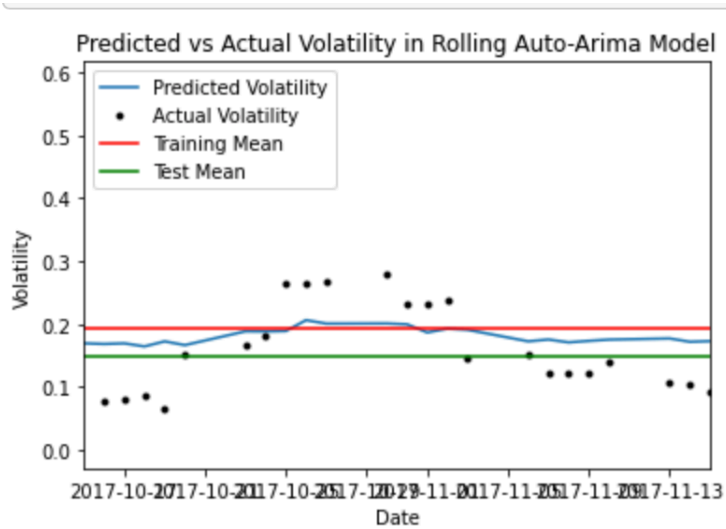
This improved the model tremendously with an RMSE of .04 and an R Squared of .74 .



The rolling prophet model falls short as the R-squared is negative. It is better than previous ones but falls short of the most recent ARIMA model. The model seems to apply more weight on the trend and seasonality and less weight on the previous day's volatility than the auto-ARIMA model. I was amazed by the improvement from the baseline model to the rolling auto-ARIMA generating a positive R-squared. The one issue with the rolling auto-ARIMA is that by fitting the model to the last day before the prediction, it was given access to 9 of the 10 days included in that average. It was able to fit itself to 90% of the test set before making a prediction. I then built a new version that predicted 10 days out each time.



The model had a worse r squared but had many positive aspects to it. The model shows that this model is actually taking into account using it in the shorter-term prediction. The overall downtrend has the predicted volatility below the mean of the training set. The model applies to much early weight to the previous period of higher volatility, but it learns to predict better as the model moves forward. The model makes accurate predictions about the direction of volatility but is unable to accurately predict the scale of those moves. You can see below.



	A	B	C	D
1	Model		RMSE	R-Squared
2				
3	Auto ARIMA baseline model		0.097	-0.221
4	Auto ARIMA rolling 10 day		0.079	0.174
5	Prophet with Implied Volatility Regressor		0.103	-0.356
6	Prophet Rolling		0.102	-0.206
7	Auto ARIMA Rolling 1 Day Window		0.045	0.75

The prophet has its statistical advantages, but I can see the model with a 1-day window adapted best to the market conditions. Both could use further work though.

Conclusions

This was extremely complex and fell short of obtaining the volatility of the stock. I did make improvements from the baseline model. The biggest improvement was shortening from a year to 10 days to even 1 day. Adding features also made improvements and could expanded on for the future.

Future Work

Of course, the biggest thing would be adding more features to improve the models. This could improve the predictive accuracy of these forecasts going forward. This can focus more on current market conditions than on longer term data. Another thing to consider would applying different training sets. In this case our training set was 4 years and test set were 1. I wonder having a shorter training set it would be a better predictor for 10 days. This may however lose some seasonality visualizations. As for customers, I did fall short in providing accurate predictions, but I provide some import insights and trends that could provide some sort of guidance. These might be of use when making predictions but not necessarily the scale of moves. Overall, this project was fascinating to me given my interest in stocks, finance, and data science.