***Springboard Capstone Project II*:** Predict Stock Price Using Tweet Sentiments

---

How does twitter sentiments affects stock price

-Balaji Santhanagopal

Table of contents

Problem Description

Predicting stock market prices has been a topic of interest among both analysts and researchers for a long time. Stock prices are hard to predict because of their high volatility influenced by diverse political and economic factors, leadership changes, investor sentiment, and many other factors. Predicting stock prices based on either historical data or textual information alone has proven to be insufficient.

Market sentiment is a qualitative measure of the attitude and mood of investors to financial markets in general, and specific sectors or assets in particular. Positive and negative sentiment drive price action, and also create trading and investment opportunities for active traders and long-term investors. Existing studies in sentiment analysis have found that there is a strong correlation between the movement of stock prices and the publication of news articles.

The purpose of the following project is to attempt to create a model that makes useful predictions about the movement of stock based on the tweet sentiments. In particular the focus was on using time-series analysis to predict the stock price based on previous day tweets about that stock. From a business perspective this could be useful in a number of ways. If tweet sentiments can be used in stock price predictions, trading strategies can be structured around those predictions. In particular buying or selling options would be ways to profit from periods of predicted high or low stock prices. From the many experiments that have been performed, Apple stock has been selected, to be studied in more depth.Additionally, while we focus on Apple stock, the techniques used here could be applied to any individual stock or even to indexes or futures, and could be used to choose stocks when constructing a diverse portfolio.

Data Acquisition and Data Wrangling

The first step in preparing for the analysis was to find appropriate data. This dataset I used is a part of the paper published in 2020 IEEE International Conference on Big Data Special Session and is available in Kaggle. This dataset contains tweets that are written about Amazon, Apple, Google, Microsoft, and Tesla by using their appropriate share tickers. This dataset contains over 3 million unique tweets with their information

such as tweet id, author of the tweet, post date, the text body of the tweet, and the number of comments, likes, and retweets of tweets matched with the related company.

	tweet_id	writer	post_date	body	comment_num	retweet_num	like_num	ticker_symbol
0	550441509175443456	VisualStockRSRC	1420070457	lx21 made $10,008 on AAPL -Check it out! htt...	0	0	1	AAPL
1	550441672312512512	KeralaGuy77	1420070496	Insanity of today weirdo massive selling. $aap...	0	0	0	AAPL
2	550441732014223360	DozenStocks	1420070510	S&P100 #Stocks Performance HD LOW $SBUX$ TGT...	0	0	0	AMZN
3	550442977802207232	ShowDreamCar	1420070807	GM TSLA: Volkswagen Pushes 2014 Record Recal...	0	0	1	TSLA
4	550443807834402816	i_Know_First	1420071005	Swing Trading: Up To 8.91% Return In 14 Days h...	0	0	1	AAPL

The tweet writer column had a lot of missing values,as I was not planning to use this column, I decided to drop it.

The post_date is in the form seconds since epoch. I converted it to a date object.

Dropped the twitter id as it was not required any more.

Performed the following operations on the tweets in order to clean them:

1. Convert them into lowercase.
2. Remove all the links starting with either http or pic.twitter.com or https
3. Remove all the special characters
4. Remove all the hashtags (#), @ symbol.
5. Remove words like: , click here, moneycontrol, markets update, live updates, News Alert, active traders.
6. Remove all the numbers.

After the cleaning and preprocessing our data looks like this:

	writer	post_date	body	comment_num	retweet_num	like_num	ticker_symbol	Date
0	VisualStockRSRC	2015-01-01 00:00:57	lx made on aapl check it out learn howtotrade ...	0	0	1	AAPL	2015-01-01
1	KeralaGuy77	2015-01-01 00:01:36	insanity of today weirdo massive selling aapl ...	0	0	0	AAPL	2015-01-01
2	DozenStocks	2015-01-01 00:01:50	s p stocks performance hd low sbux tgt dvn ibm...	0	0	0	AMZN	2015-01-01
3	ShowDreamCar	2015-01-01 00:06:47	gm tsla volkswagen pushes record recall tally...	0	0	1	TSLA	2015-01-01
4	i_Know_First	2015-01-01 00:10:05	swing trading up to return in days swingtradin...	0	0	1	AAPL	2015-01-01

Since statistical analysis of stocks is an area of high interest, there are a wealth of resources on the internet detailing the movement of stock prices. I was able to find a

source of data that provided pricing data about these stocks that included closing price, high, low, daily volume and several others. Yahoo finance contains free data on many stocks traded on exchanges.

The first step in working with yahoo finance data was to determine how to use their API. I found a Python module to download stock data from Yahoo! Finance. Once this step was completed the stock data could be pulled into a notebook by way of a csv file. After converting the csv into a Pandas dataframe, I checked the newly created dataframe for NaN values. A quick check revealed no NaN values.

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	ticker
2015-01-02	44.574001	44.650002	42.652000	43.862000	23822000	0.0	0.0	TSLA
2015-01-05	42.910000	43.299999	41.431999	42.018002	26842500	0.0	0.0	TSLA
2015-01-06	42.012001	42.840000	40.841999	42.256001	31309500	0.0	0.0	TSLA
2015-01-07	42.669998	42.956001	41.956001	42.189999	14842000	0.0	0.0	TSLA
2015-01-08	42.562000	42.759998	42.001999	42.124001	17212500	0.0	0.0	TSLA

Exploratory Data Analysis

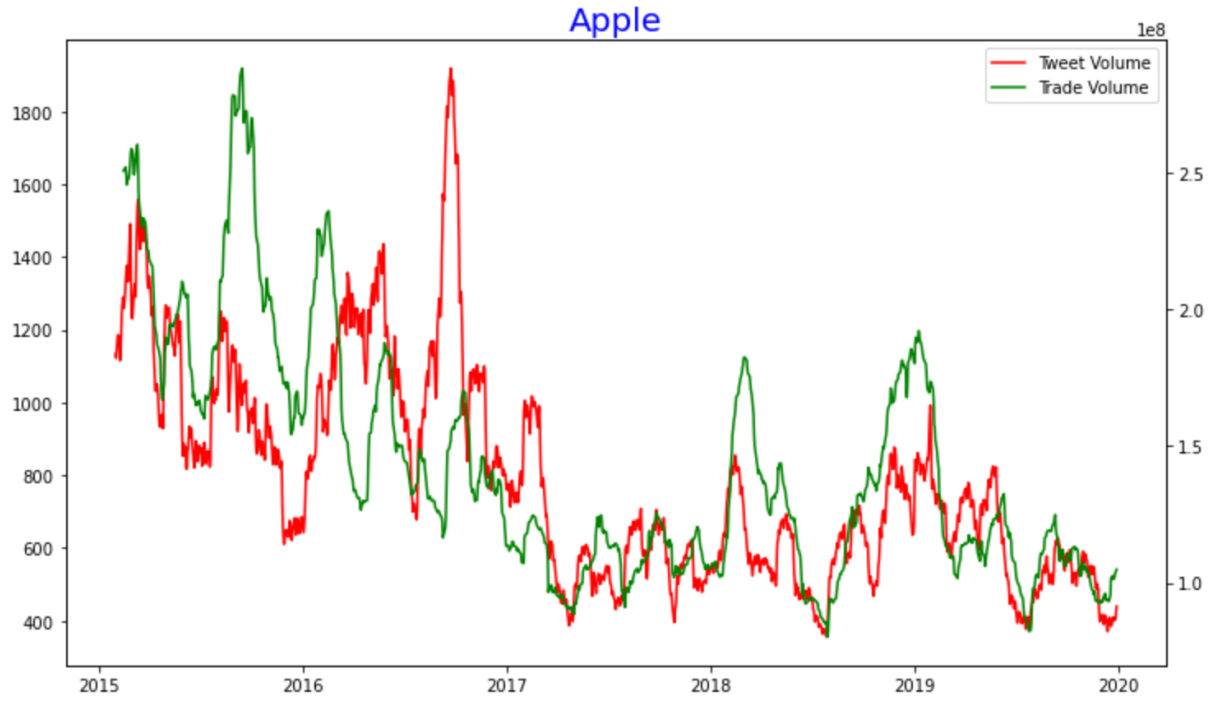
Now I visually analyzed to see if there is any trend in the volume of a stock traded and the volume of its corresponding tweets on the previous day. I got the average 30 days window for both tweet volume and stock volume to get a feel of the trend.

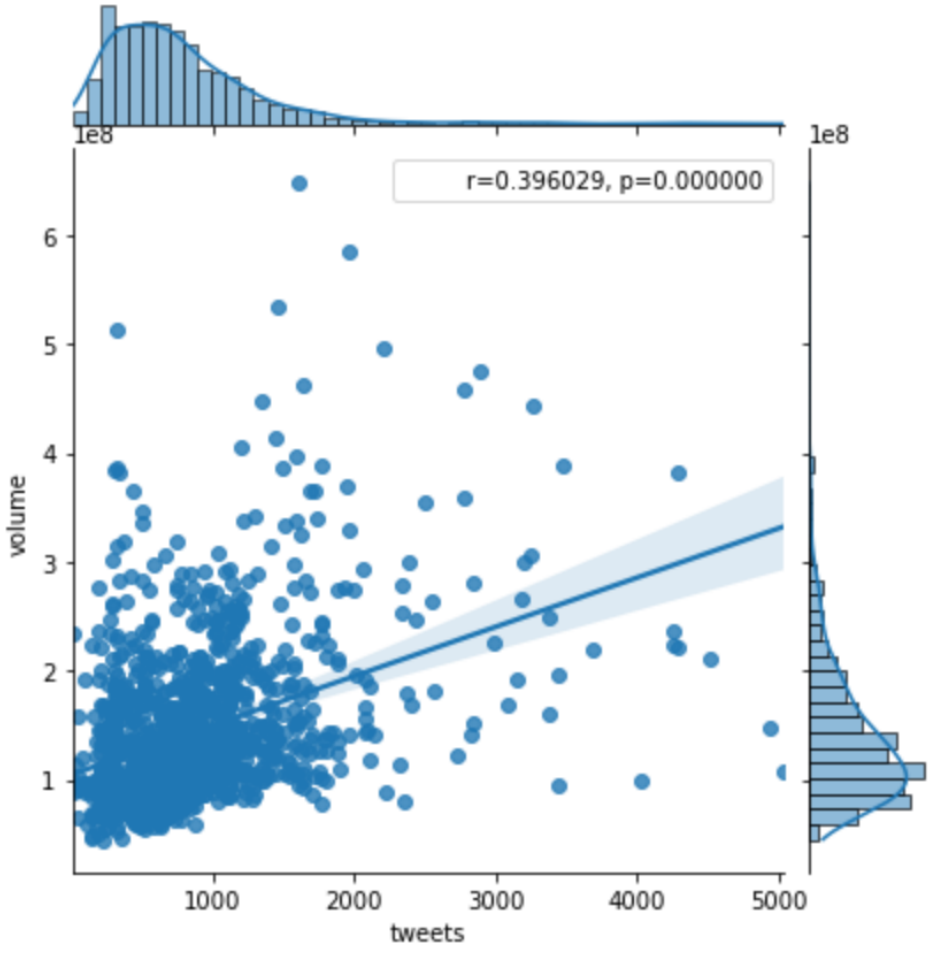
Also, I checked if there was any correlation between the volume of tweets with the volume of stock traded the next day. Here a linear relationship between the variables is not assumed, although a monotonic relationship is possible. The widely used Pearson correlation coefficient measures the strength of the linear relationship between normally distributed variables. When the variables are not normally distributed or the relationship between the variables is not linear, it is appropriate to use the Spearman rank correlation method. Our hypothesis for the test is as follows:

$\mathbf{H_{Null}}$: Our hypothesis would be that there is no relation between tweet volume and the stock volume traded.

$\mathbf{H_{Alternative}}$: Here our hypothesis is that there is a relation between the volume of tweets and the volume of stock traded.

I calculated the p-value of the spearman correlation and if the p-value was below 0.05, the null hypothesis will be rejected and we can fairly conclude that there is a positive/negative correlation between the stock volume and trade volume. The figure below shows the relationship between trade volume and tweet volume for Apple.





The calculated values for other stocks is given in the table below.

Stock	P-value	r
AAPL	0.000	0.396
TSLA	0.000	0.596
AMZN	0.000	0.215
MSFT	0.296	0.029
MSFT	0.000	0.234

It looks like the volume of tweets has a positive correlation with the trade volume. However, the strength of the correlation is relatively low. Additionally, it is not certain that volume of tweets is always correlated with the share price as demonstrated in Microsoft's case where the p-value exceeded the predetermined threshold of 0.05. The significance of values observed above the threshold means the relationship is not statistically significant.This means that the Hypothesis is rejected and the null Hypothesis is accepted.

Next, I analyzed how the sentiment of the tweets impact the stock price. I categorized the tweet sentiments as either positive, negative or neutral.

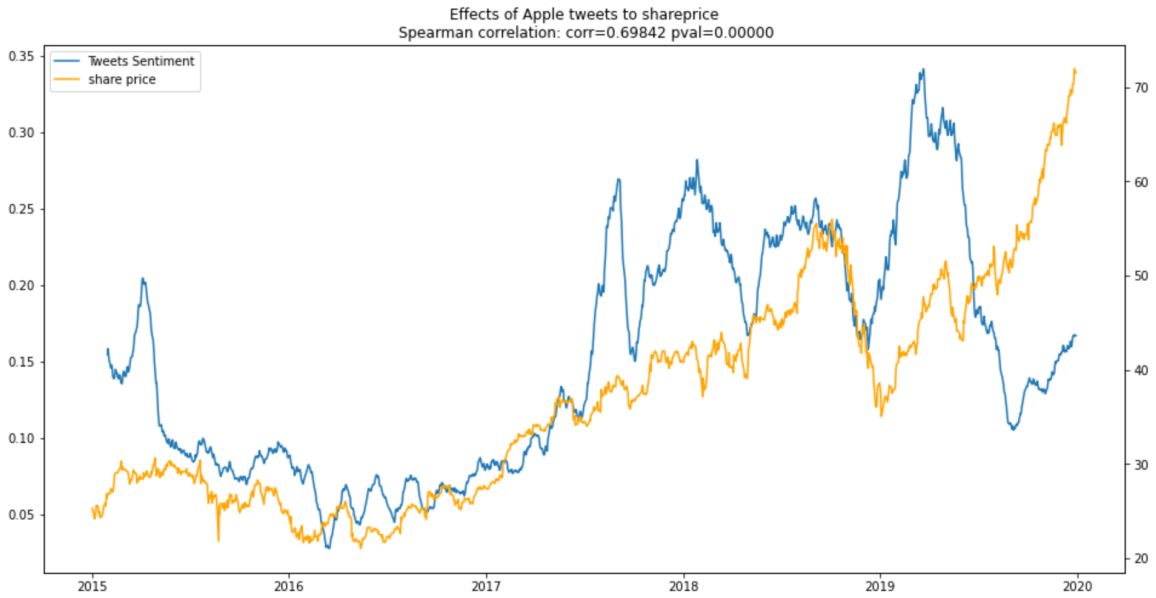
$\mathbf{H_{Null}}$: The sentiment of the tweet has no correlation with the shareprice of the company.

$\mathbf{H_{Alternative}}$: The sentiment of the tweet has a correlation with the shareprice of the company.

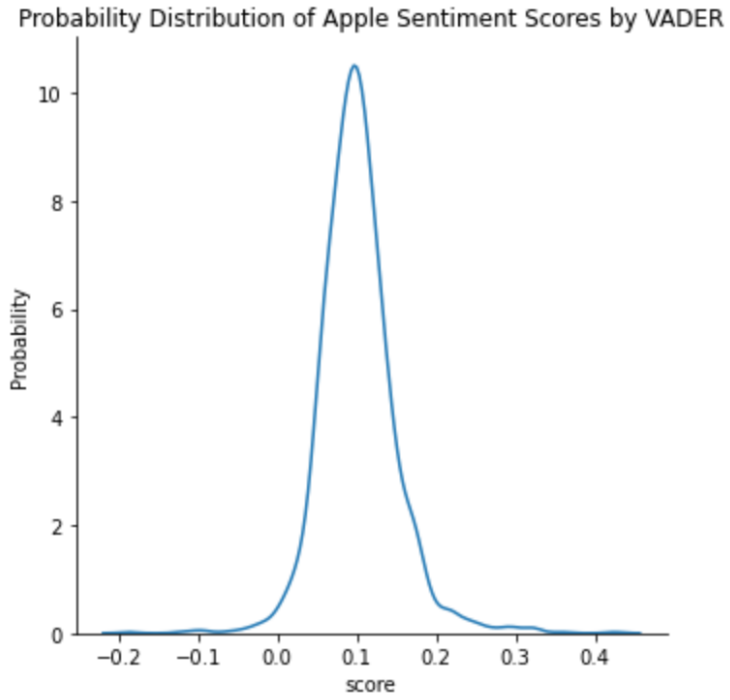
I did not tokenize, remove stopwords, and got bigrams, etc as I used a pre-trained sentiment analyzer VADER here since the data is unsupervised.**VADER** stands for **Valence Aware Dictionary and sEntiment Reasoner. Vader performs well** for the analysis of sentiments expressed in social media. These sentiments must be present in the form of comments, tweets, retweets, or post descriptions.**VADER** is a lexicon and rule-based analysis tool.VADER gives a compound score for each paragraph. Score = -1 signifies negative news and score = 1 signifies positive news. Positive news should raise the index prices and vice versa.

I found the tweet sentiment for each tweet and got the mean sentiment of the tweets per day for the stock.

The figure below shows the correlation and the p-value for Apple.



The probability distribution of the tweets on Apple sentiment is shown below.



For the majority of news, VADER is confident in detecting either positive or negative sentiment since most of the points lie on the boundary. This shows accurate and confident prediction from VADER library

Ticker Symbol	p-value	correlation
APPL	0.000	0.120
TSLA	0.000	0.698
AMZN	0.000	0.575
GOOGL	0.000	0.629
MSFT	0.000	0.512

There seems to be a stronger correlation between the sentiment of the tweets and the share price of the company as compared to the previous hypothesis.

I then performed a time series analysis to see if twitter sentiments can be used to predict stock prices. The analysis of time series is based on the assumption that successive values in the data file represent consecutive measurements taken at equal time intervals. I built the following models

- ARIMA
- Facebook Prophet
- V**ector Autoregression (VAR)**

First, I built the univariate models using the stock data of Apple.

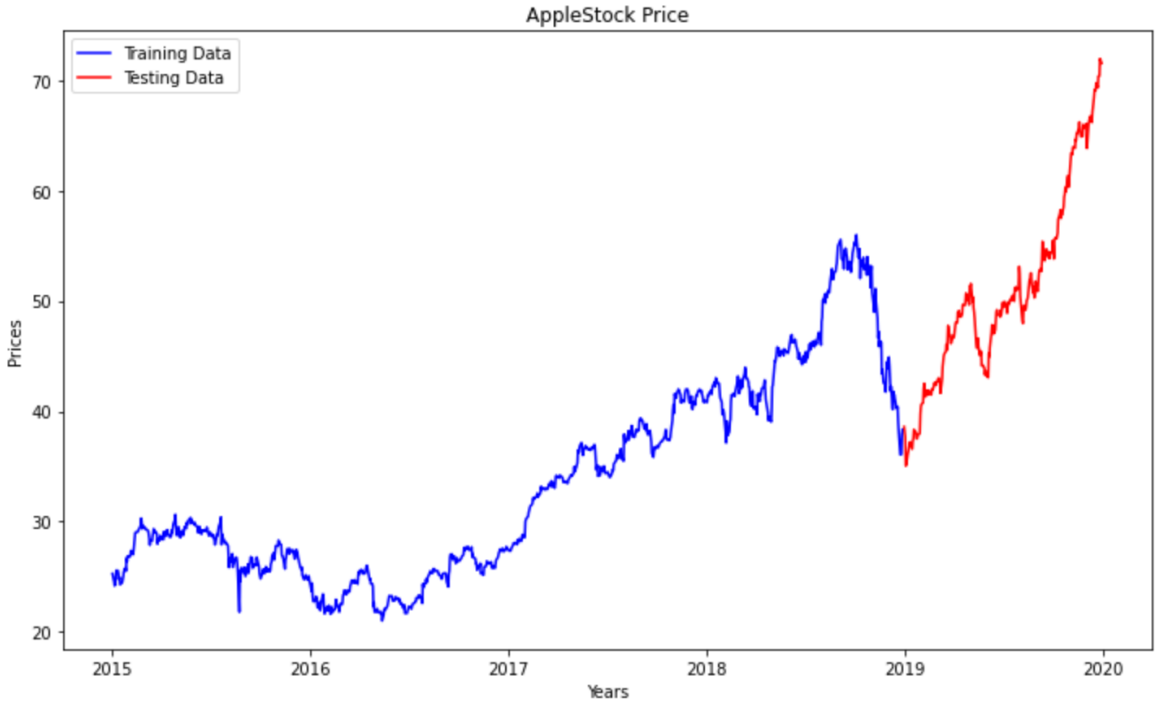
Then, I used V**ector Autoregression (VAR)** which is a multivariate forecasting algorithm that is used when two or more time series influence each other. Here, I used the Apple stock data and the relevant tweet sentiment to build the VAR model.

Scoring Metric

Root Mean Square Error (RMSE)was used to compare the performance of various models.

The expectation here is that twitter sentiment used along with stock data should give us a lower RMSE score, than when calculated with the univariate models ARIMA and Facebook Prophet.

Train Test Split



Since it is a time-series data, we cannot split it randomly. Hence, we are considering the last 20% data as a test set and the first 80% data as a train set.
Train Data Shape: (1005, 2)
Test Data Shape: (252, 2)

Time series Analysis

- Time series is a collection of data points that are collected at constant time intervals.
- It is time dependent.

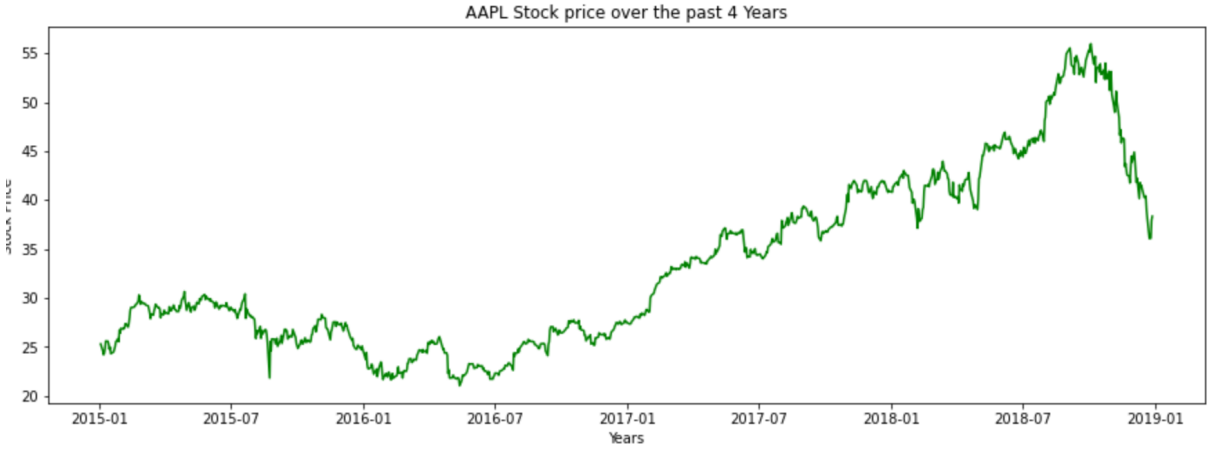
- Most time series have some form of **seasonality trends**. For example, if we sell umbrellas, we expect there will be higher sales in the rainy season. The sales time series has seasonality trends.

There are three basic criteria for a time series to understand whether it is a stationary series or not. Statistical properties of time series such as mean and variance should remain constant over time to call time series stationary.

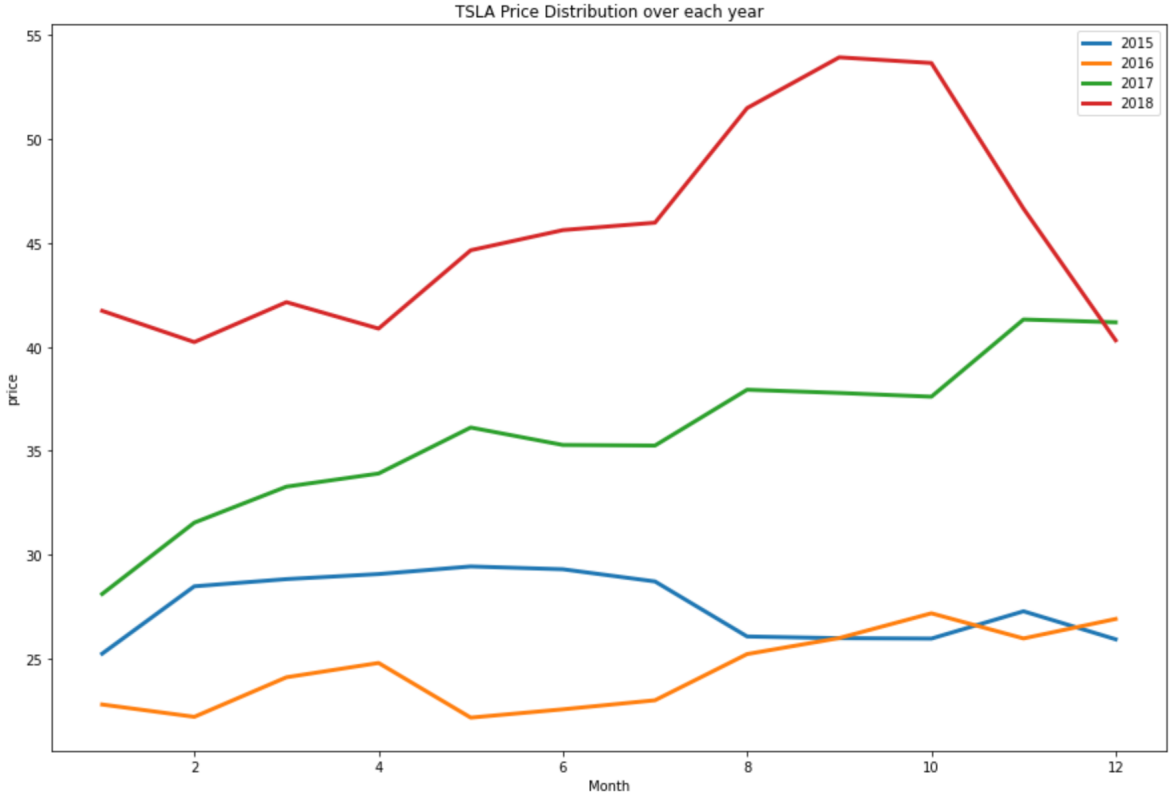
Following are the **3 qualities of a stationary time series**:

- Constant mean
- Constant variance
- Autocovariance that does not depend on time. Autocovariance is the covariance between the time series and lagged time series.

I visualized and checked the seasonality and trend of the time series first.



Trend: This time series showed an upward trend. This was a non-stationary time series. I needed to convert it to stationary to forecast accurately. Then I checked for the seasonality.



Seasonality: The time series had a slight seasonal variation.

I observed that the stock price increases in the later half of the year.

Now I checked the stationarity of the time series. The stationarity can be checked using the following methods:

- **Plotting Rolling Statistics**: We have a window, let's say window size is 6 and then we find rolling mean and variance to check stationary.
- **Dickey-Fuller Test**: The test results comprise of a Test Statistic and some Critical Values for difference confidence levels. If the test statistic is less than the critical value, we can say that time series is stationary. Our hypothesis is

$\mathbf{H_{Null}}$: The time series is non-stationary.

$\mathbf{H_{Alternate}}$: The time series is stationary.

For time series to be stationary we should get a p-value of less than 5% to reject the null hypothesis.


```
   Augmented Dickey-Fuller Test on "Stock Price"
   ----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -1.4224
No. Lags Chosen       = 22
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.5715. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

- The first criteria for stationarity is constant mean. It fails because the mean is not constant as you can see from the plot (black line) above.
- Second is constant variance. It looks constant. (Green Graph above)
- Third one is that if the test statistic is less than the critical value, it can be said that the time series is stationary. Here: test statistic = 0.674 and critical values = {'1%': -3.431667761145687, '5%': -2.8621223070279247, '10%': -2.5670799628923104}. Test statistics are bigger than the critical values. So, no stationarity.

It was clear that the time series was not stationary. I had to make sure that the time series was stationary.

Two methods which can help make it stationary, they are:

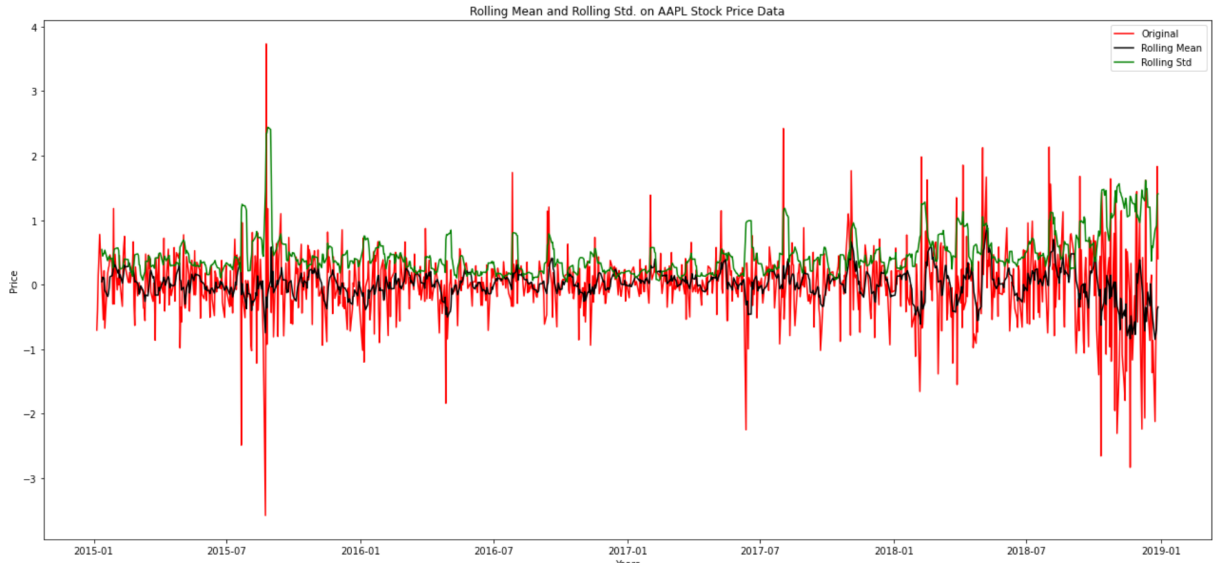
- **Moving Average Method**
- **Differencing Method**


```
   Augmented Dickey-Fuller Test on "Stock Price"
   ----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -1.8403
No. Lags Chosen       = 10
Critical value 1%     = -3.441
Critical value 5%     = -2.866
Critical value 10%    = -2.569
=> P-Value = 0.3606. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

Moving Average method

Mean is constant over time. There is no trend visible, the p-value is also less than 5%. But the test statistic is not less than Critical Value. Variance is not constant. The time series is still not stationary.

I tried the differencing method.



Differencing method

```
   Augmented Dickey-Fuller Test on "Stock Price"
   ----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -4.8086
No. Lags Chosen       = 21
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.0001. Rejecting Null Hypothesis.
=> Series is Stationary
```

Now there was no trend visible,the time series was almost stationary now. This time series could be used for forecasting.

Modeling

i. ARIMA

An ARIMA model is a class of statistical models for analyzing and forecasting time series data.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

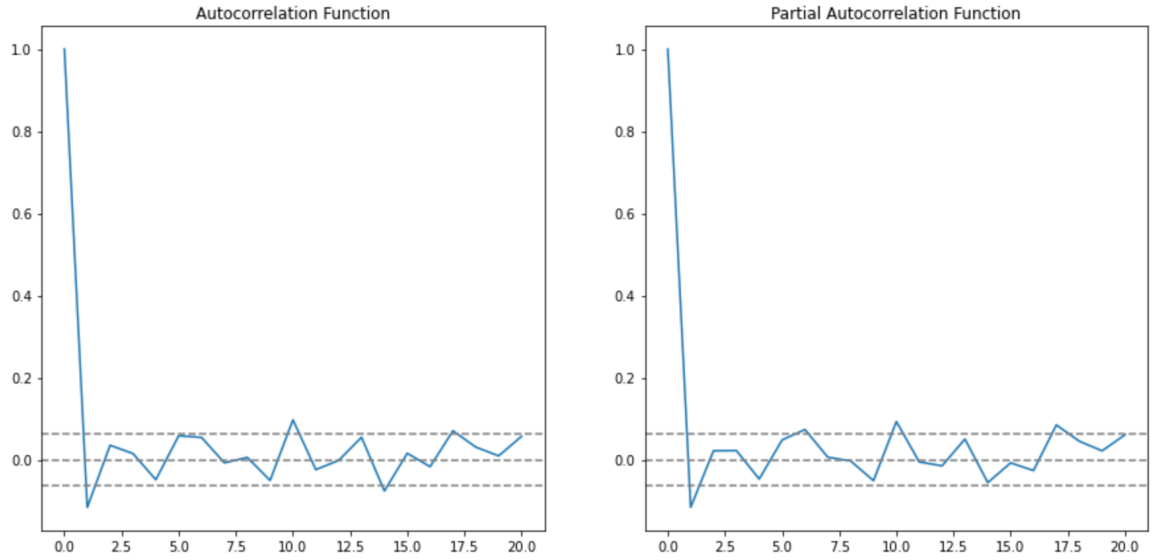
This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
- **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from observation at the previous time step) in order to make the time series stationary.
- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components is explicitly specified in the model as a parameter. The parameters of the ARIMA model are defined as follows:

- **p**: The number of lag observations included in the model, also called the lag order.
- **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
- **q**: The size of the moving average window, also called the order of moving average.

I plotted the Autocorrelation and Partial Autocorrelation Plot to identify the above parameter values.

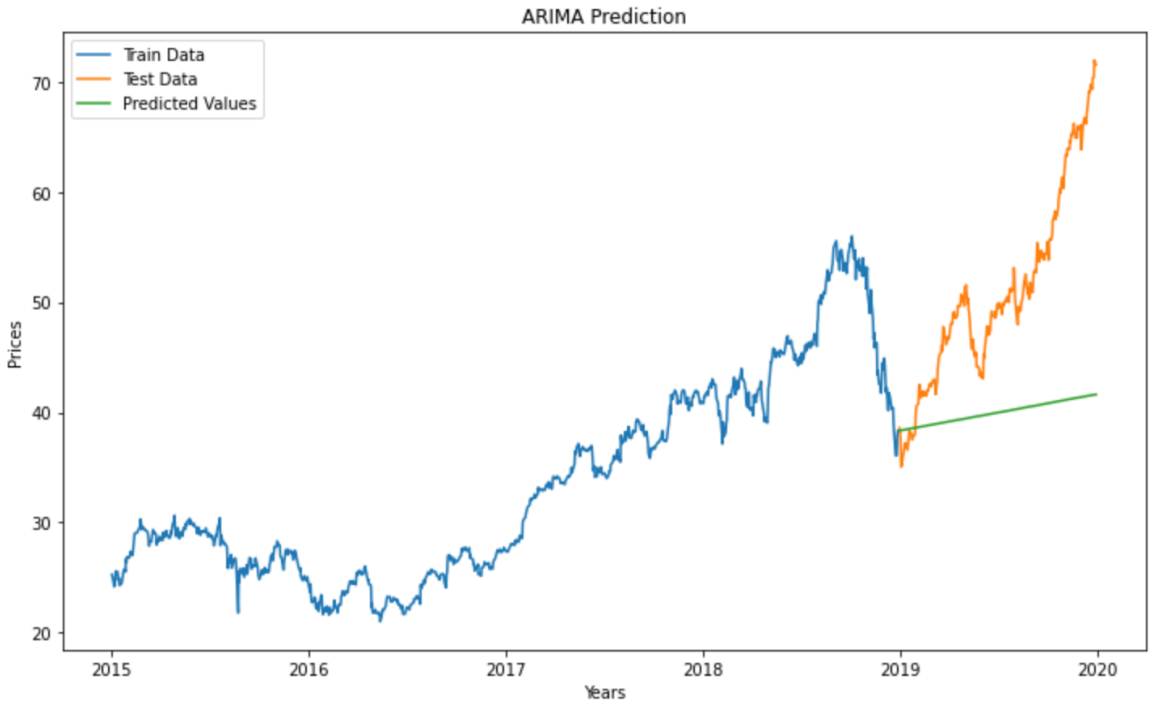


p – The lag value where the PACF chart crosses the upper confidence interval for the first time. If you notice closely, in this case p=1.

q – The lag value where the ACF chart crosses the upper confidence interval for the first time. If you notice closely, in this case q=1.

d - In differencing method, shift of 1 period produced a stationary time series. So I used d = 1.

I forecasted the stationary time-series which I got after the differencing method using ARIMA. Then transformed the results to get the original time series.

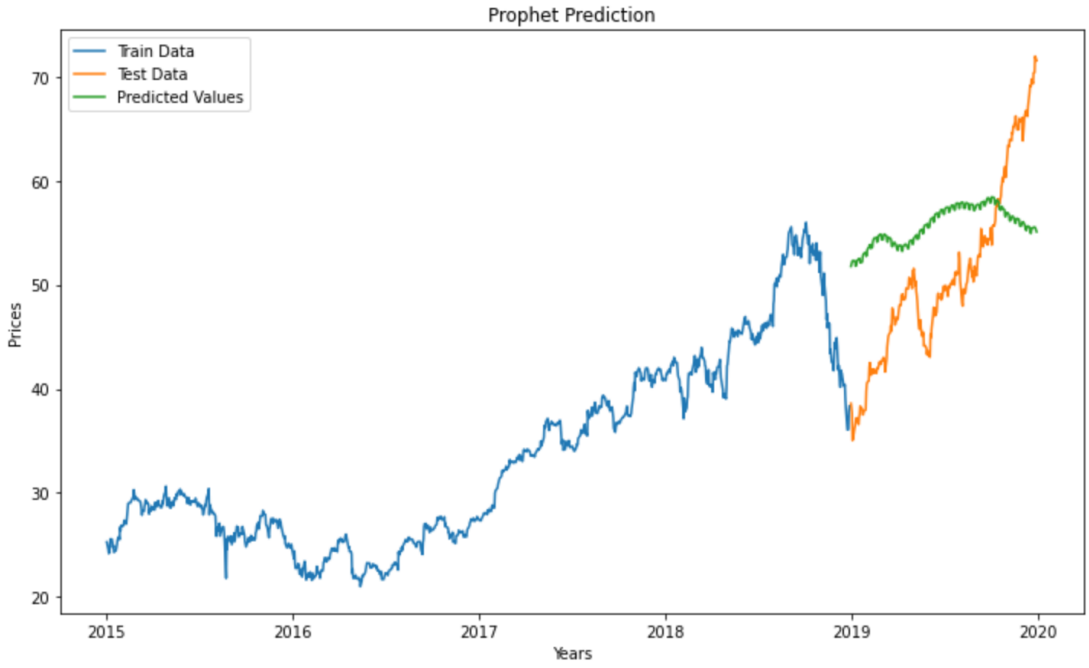


RMSE from ARIMA model = 13.02

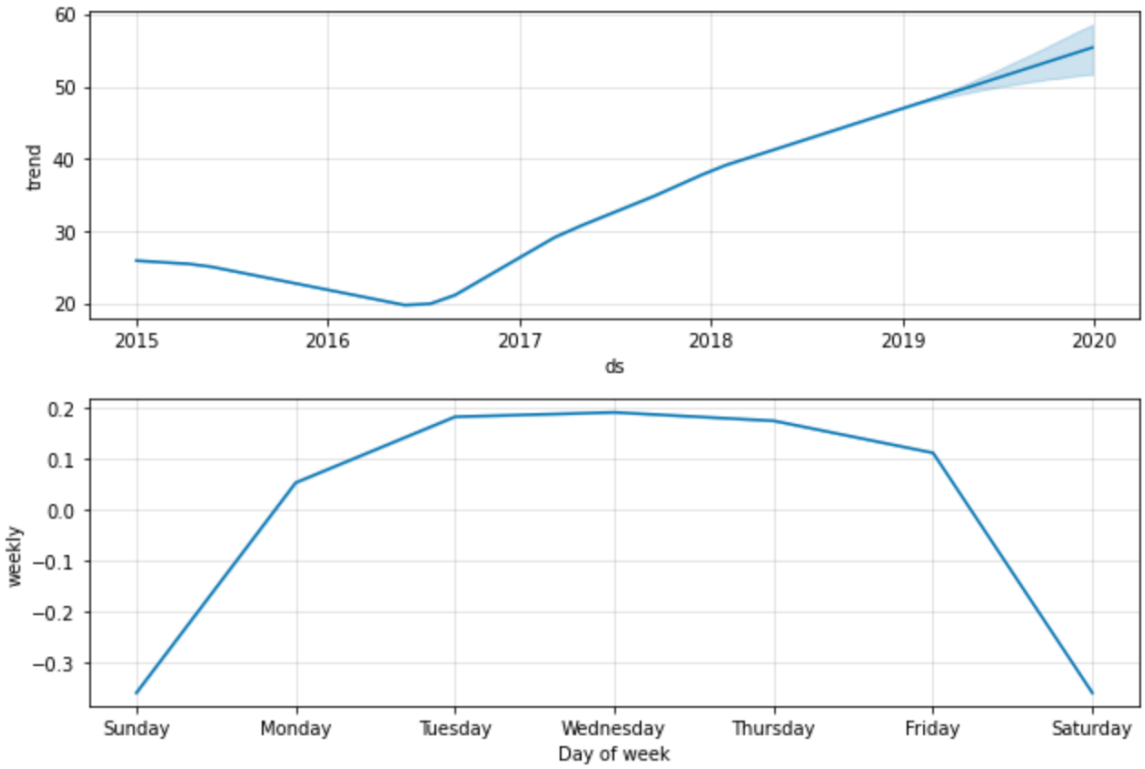
The next step was to see if this can be improved using Facebook Prophet.

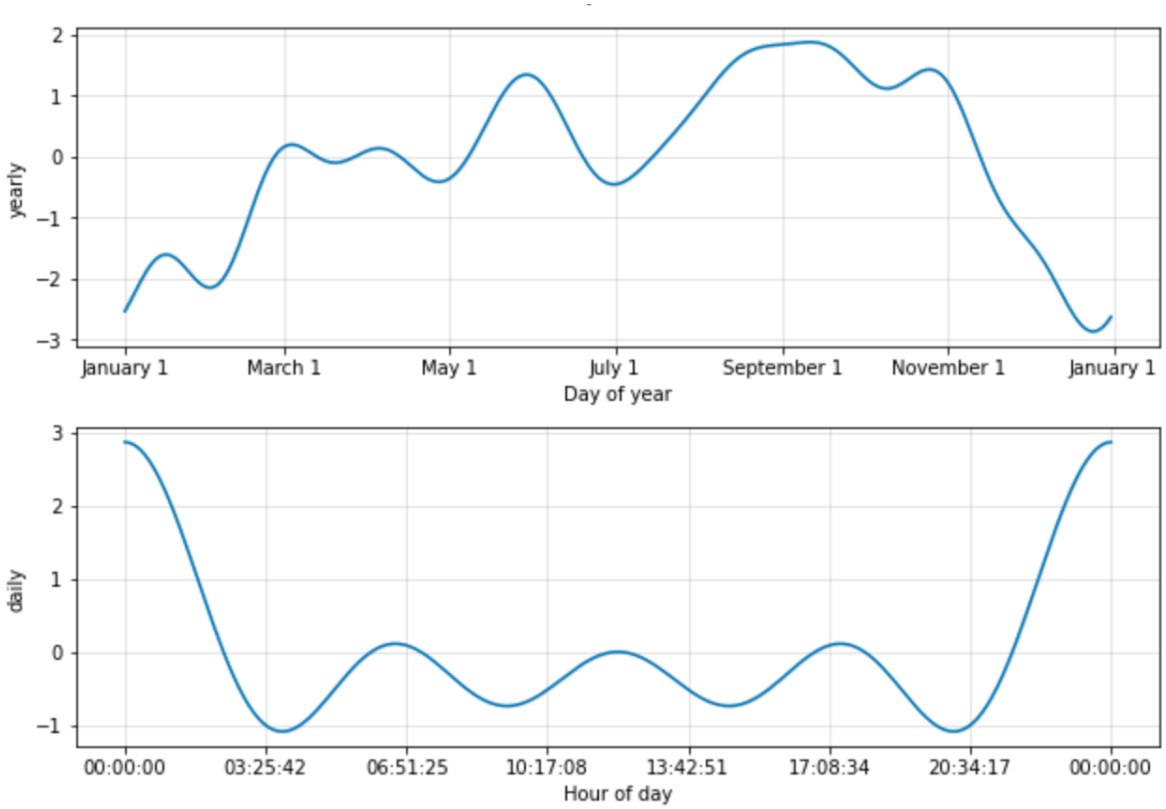
ii. Facebook Prophet

The prophet is an open-source library published by Facebook that is based on **decomposable (trend+seasonality+holidays) models**. It provides us with the ability to make time-series predictions with good accuracy using simple intuitive parameters and has support for including the impact of custom seasonality and holidays!



R**RMSE from Facebook Prophet =** 9.18


60
50
40
30
20
trend
2015
2016
2017
2018
2019
2020
ds
0.2
0.1
0.0
−0.1
−0.2
−0.3
weekly
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week



The data has some seasonal information present.
Following points can be observed from the above graphs:

1. The data shows an upward trend.
2. Stock price goes higher normally towards mid- week.
3. There is a high chance to observe a 52 week high Stock Price in the last quarter of the year.
4. Stock Price fluctuates during the whole day.

iii. Vector Autoregression Model

A Vector autoregressive (VAR) model is useful when one is interested in predicting multiple time series variables using a single model.

Testing Causation using Granger's Causality Test

The basis behind Vector AutoRegression is that each of the time series in the system influences each other. That is, you can predict the series with past values of itself along with other series in the system.

Using Granger's Causality Test, first test the relationship before even building the model. The null hypothesis here is that the coefficients of past values in the regression equation is zero.

Check Granger Causality of all possible combinations of the Time series. The rows are the response variable, columns are predictors. The values in the table are the P-Values. P-Values lesser than the significance level (0.05), implies the Null Hypothesis that the coefficient of the corresponding past values is zero, that is, the X does not cause Y, can be rejected.

data : pandas dataframe containing the time series variables
variables : list containing names of the time series variables.

	score_x	Open_x
score_y	1.0000	0.183
Open_y	0.4387	1.000

Since the p-value is < significance level (0.05),we can say twitter sentiment impacts the stock price and we can reject the null hypothesis.

Cointegration test helps to establish the presence of a statistically significant connection between two or more time series. When two or more time series are cointegrated, it means they have a long run, statistically significant relationship.

This is the basic premise on which Vector Autoregression(VAR) models are based on. So, it is common to implement the cointegration test before starting to build VAR models.

```
Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
score  ::  48.25     > 12.3212   =>   True
Open   ::  6.16      > 4.1296    =>   True
```

Test Train Split

The VAR model will be fitted on df_train and then used to forecast the next 4 observations. These forecasts will be compared against the actuals present in test data.
Train Data Shape: (1005, 2)
Test Data Shape: (252, 2)

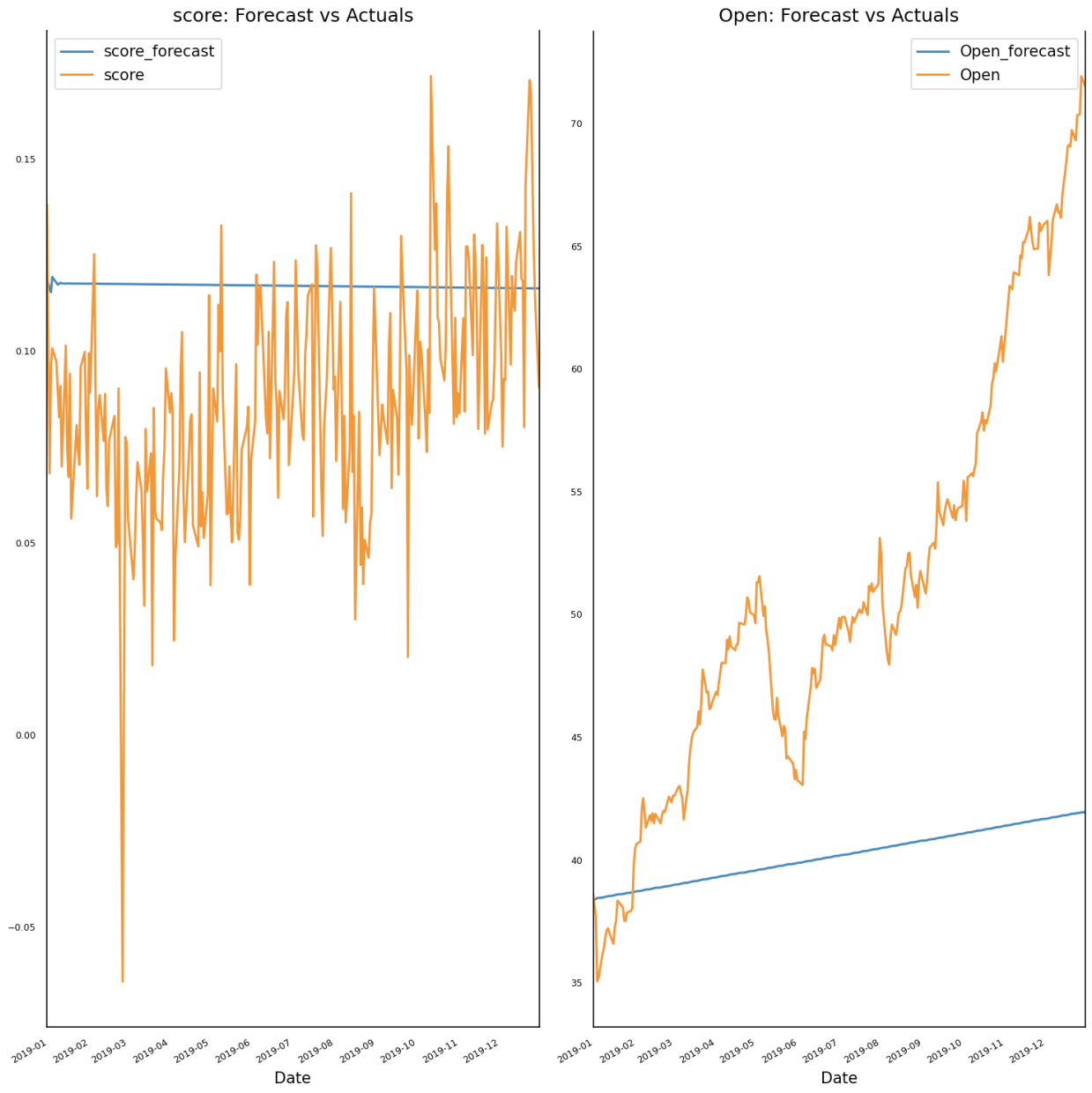
The VAR model requires the time series we want to forecast to be stationary. I used Augmented Dickey-Fuller Test (ADF Test) to test the time series.

```
   Augmented Dickey-Fuller Test on "score"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -6.7624
No. Lags Chosen       = 6
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Open"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -1.4224
No. Lags Chosen       = 22
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.5715. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

It is clear that the AAPL stock price is not stationary. I used the Differencing method to make the series stationary. The results are as shown below.

```
   Augmented Dickey-Fuller Test on "score"
   ---------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -11.3807
No. Lags Chosen       = 22
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Open"
   ---------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -4.8086
No. Lags Chosen       = 21
Critical value 1%     = -3.437
Critical value 5%     = -2.864
Critical value 10%    = -2.568
=> P-Value = 0.0001. Rejecting Null Hypothesis.
=> Series is Stationary.
```

The plot of Forecast vs Actuals is as given below.



RMSE for Vector Autoregression = 13.1736

Model Performance

Model	RMSE
ARIMA	13.02
Facebook Prophet	9.18
Vector autoregression	13.17

Conclusion

I used time-series data and built univariate and multivariate models for our analysis. Though I found some correlation between stock price and tweet sentiments, when it comes to predicting the stock price this relationship remained inconclusive as seen from our model results.

Further Improvements

Here are some of the leads to improve the results from the our discussed solution:

1. Collect news data for more years to have more data points.
2. We have limited stock price data. To do a more extensive stock analysis, we can take hourly stock price data instead of daily stock price data to increase the data points. This can improve accuracy.
3. Use Deep learning models like LSTM. We can tinker with the LSTM architecture and hyperparameters to improve the model accuracy.
4. Instead of using a pre-trained VADER Sentiment Analyzer, we can train our own model by first creating training data. A custom trained model should give better sentiment results as it will get trained on the stock market news language.
5. There is some research that shows using GAN(Generative Adversarial Network), Reinforcement Learning can also be used to predict stock prices better.

Customer Recommendations

The goal of this exploration was to build a model that could predict stock prices based on tweet sentiments that would be actionable in the trading arena for a client. While I believe the project fell short in that regard, the modeling did provide some useful insight into the market. I was able to identify yearly and weekly patterns as well as overall trends that could provide guidance.

Also, while the VAR model did not provide the accuracy I had hoped for, I do believe that it made useful predictions about the direction of stock price, and that these predictions might be of use when making decisions. To be useful, the model need not necessarily predict the scale of moves.

In future we can use Deep learning models like LSTM combined with a custom trained sentiment analyser model to get better results.

References:

Vector Autoregression (VAR) – Comprehensive Guide *By Selva Prabhakaran*
https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python/

Time Series Prediction Tutorial with EDA -
https://www.kaggle.com/kanncaa1/time-series-prediction-tutorial-with-eda