IST 652 Project Report

“Stocks Prices and Finance Tweets Analysis”

Submitted by:

Lakshya Kumar Gupta
Qingqing Hu

I. Introduction

“In the long run, valuations may drive stock prices, but in the short term it is market sentiment that moves prices.”

The objective of this project is to analyze tweets' sentiments of five blue chip companies and study the effect of these sentiments on the stock prices of these companies. We are analysing five well known companies (Amazon, Microsoft, Apple, Advanced Micro Devices(AMD), Tesla) in this project.

In order to achieve the goal of the project we performed various analyses on tweets as well as financial stock prices. We performed word frequency analysis to analyse the content present in the tweets. We used VADAR's Sentiment Intensity Analyzer on these tweets to calculate sentiment scores. We used these sentiment scores to analyse relationships between market sentiment and stock prices. In addition to this, we also tried to forecast the stock prices using a multivariate time series model, called Vector Auto Regression (VAR) which uses past values of a group of time-dependent variables to forecast future values. In this case we are using tweet sentiments and past stock prices of the companies to forecast future prices.

II. Dataset Description

In this project, we will use two datasets.

1. Financial Tweets dataset from Kaggle
2. Stock prices data from Yahoo Finance

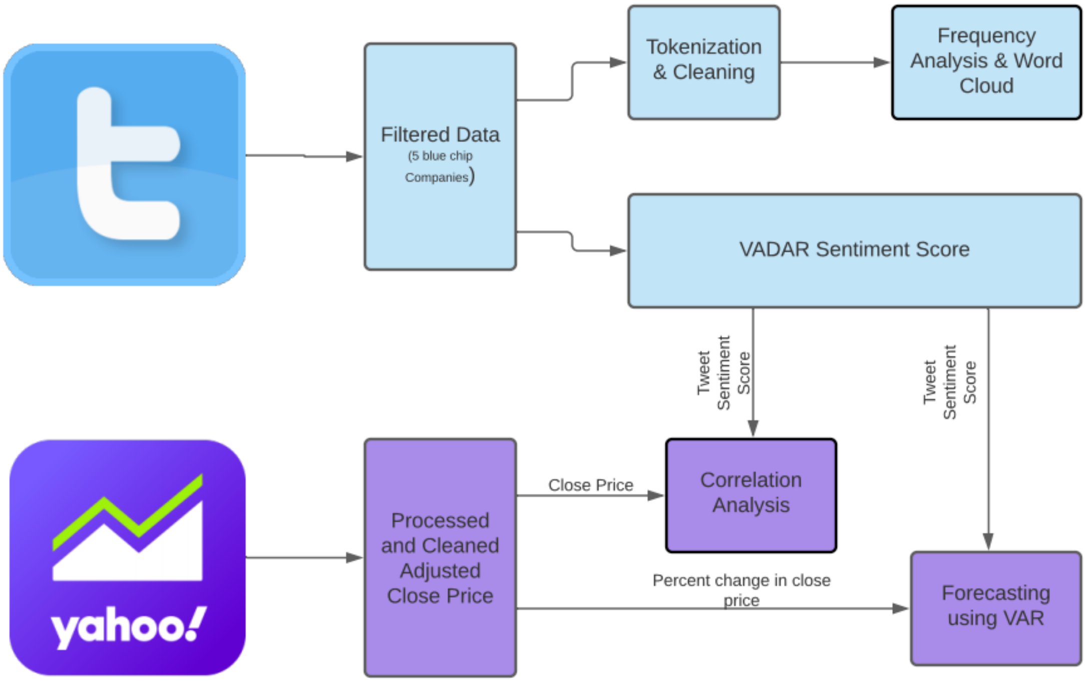
Financial Tweets dataset from Kaggle contains more than 92300 tweets which were collected between April 9 and July 16, 2020 using not only the SPX500 tag but also the top 25 companies in the index and “#stocks”.
Dataset link:
https://www.kaggle.com/utkarshxy/stock-markettweets-lexicon-data?select=tweets_labelled_09042020_16072020.csv

Historical stock prices data for the five blue chip companies of the same period was downloaded from Yahoo Finance (https://finance.yahoo.com/). We used the pandas_datareader package to fetch the data between required dates from Yahoo Finance. The dataset contains the company name, date, closing price, opening price, adjusted price and volume. Because the adjusted price is more representative, we will focus on this attribute in the analysis and comparisons.

III. Methodology

We will be using CRISP-DM methodology to analyze our dataset and give best results. The CRISP-DM stands for CRoss Industry Standard Process for Data Mining that involves following phases of data science life cycle,

1) Business Understanding
 2) Data Understanding
3) Data Preparation
4) Modeling
5) Evaluation



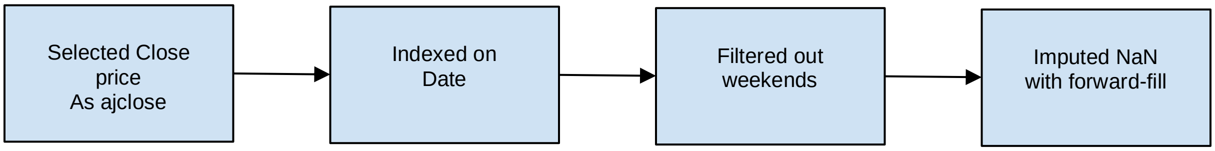
Flow Diagram

Data Exploration and Preprocessing

Stocks Data

Data Fetching:
We leveraged the pandas_datareader package to fetch the data between required dates from Yahoo Finance.We choose Amazon, Tesla, Microsoft, Apple and AMD, and select the data on April 9, 2020 and solstice on July 16, 2020. Through the DataReader function, the stock price information of these five companies during this period is intercepted from Yahoo Finance.

Data Preprocessing:
We selected the closing price as the research focus, so we first selected the closing prices of these five companies to form a new data set called adjclose. At the same time, since the stock market will not be open on weekends, we first extract the weekdays during this period and re-sort the entire data set according to weekdays. Since reindexing will insert missing values (NaN) for the dates that were not present in the original set. To resolve this, we filled the missing by replacing them with the latest available price for each instrument.


Selected Close price As ajclose
Indexed on Date
Filtered out weekends
Imputed NaN with forward-fill

Tweets dataset

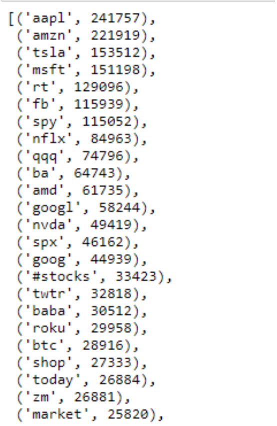
For tweets dataset, download the csv file from Kaggle, then read this csv file into pandas dataframe. This Dataset contains financial tweets and contains tweets of many other companies which are not in the scope of this project. So, we have to apply filters to get tweets of our desired companies. We used a substring match of name and ticker of the company to map with their tweets. And finally created a dataframe containing filtering tweets

Data Analysis

Word Frequency and Word cloud

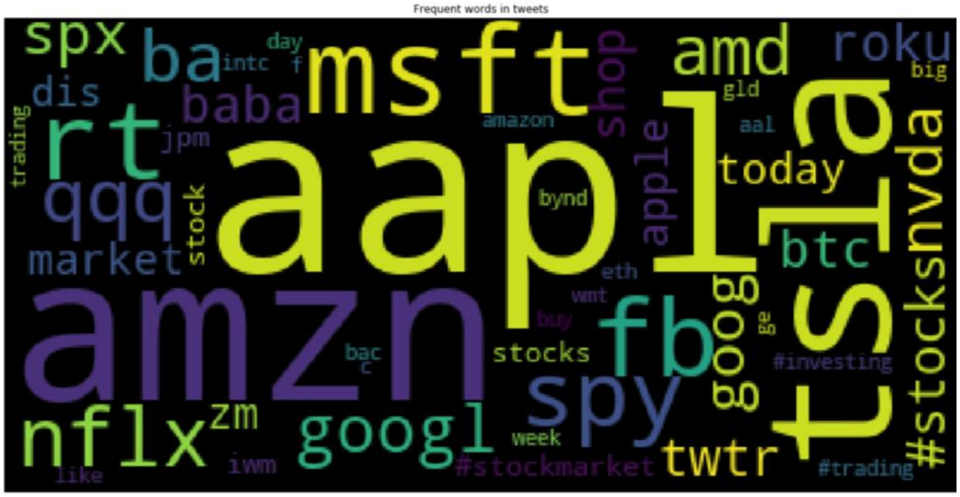
In the next step, we will conduct a more detailed analysis of the text. In this step, we will use the TweetTokenizer from nltk. We first split the text, then removed the stop words, and finally tokenize the lower tweets and store tokens into the flat list allTokens. We also used an alpha filter to filter out non alphabetic tokens.

Next, we will make a statistic of the words that appear in the text, frequency of each word appears, and then sort according to the frequency, shown as below:



```
[('aapl', 241757),
 ('amzn', 221919),
 ('tsla', 153512),
 ('msft', 151198),
 ('rt', 129096),
 ('fb', 115939),
 ('spy', 115052),
 ('nflx', 84963),
 ('qqq', 74796),
 ('ba', 64743),
 ('amd', 61735),
 ('googl', 58244),
 ('nvda', 49419),
 ('spx', 46162),
 ('goog', 44939),
 ('#stocks', 33423),
 ('twtr', 32818),
 ('baba', 30512),
 ('roku', 29958),
 ('btc', 28916),
 ('shop', 27333),
 ('today', 26884),
 ('zm', 26881),
 ('market', 25820),
```

In addition to that, we built a word cloud and visualized it according to the frequency of the words appearing.

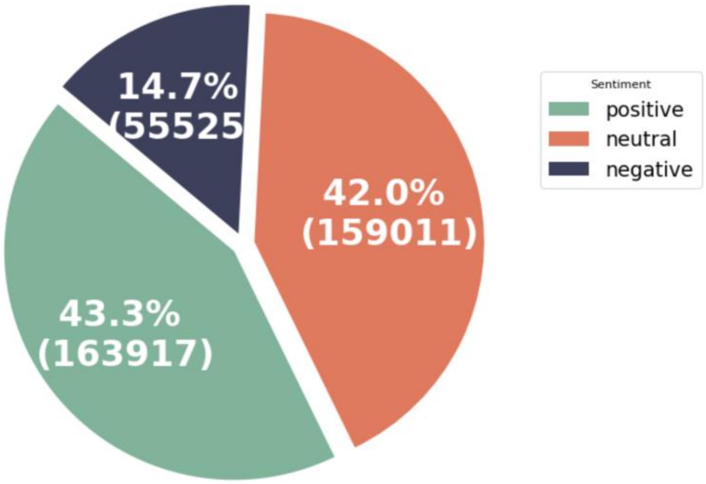

Frequent words in tweets

Sentiment Analysis

We leveraged VADER Sentiment analyzer to get sentiments from the filtered tweet of 5 companies. VADER is short for Valence Aware dictionary and sentiment reasoning. Vader is identified as a lexicon and rule-based sentiment analysis tool used to analyze the sentiment of a text. Lexicon is a list of lexical features (words) which are labeled with positive or negative based on semantic meaning.VADER is intelligent enough to understand the emphasis of capitalization and punctuation, therefore, it was not a necessity to use VADER on cleaned texts.

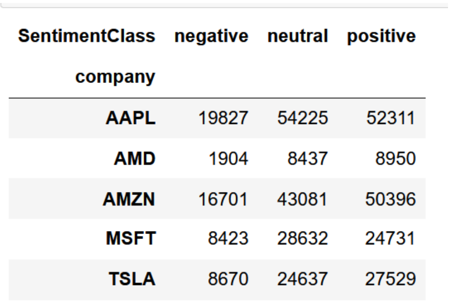
The SentimentIntensityAnalyzer(). polarity_scores() in nltk.sentiment.vader is the dictionary that has a ‘*compound*’ as a key which signifies the normalized score of sentiments ranging from -1 (most extreme negative) to 1 (most extreme positive).

We also used this compound score to classify tweets, values from –1 to -0.05 classified as negative sentiment, 0.05 to 1 represents positive sentiment and remaining represents neutral sentiment. To preliminarily understand the distribution of sentiment class for all Twitter data, we counted the number of tweets per class and plotted using a pie for positive, negative and neutral sentiments.

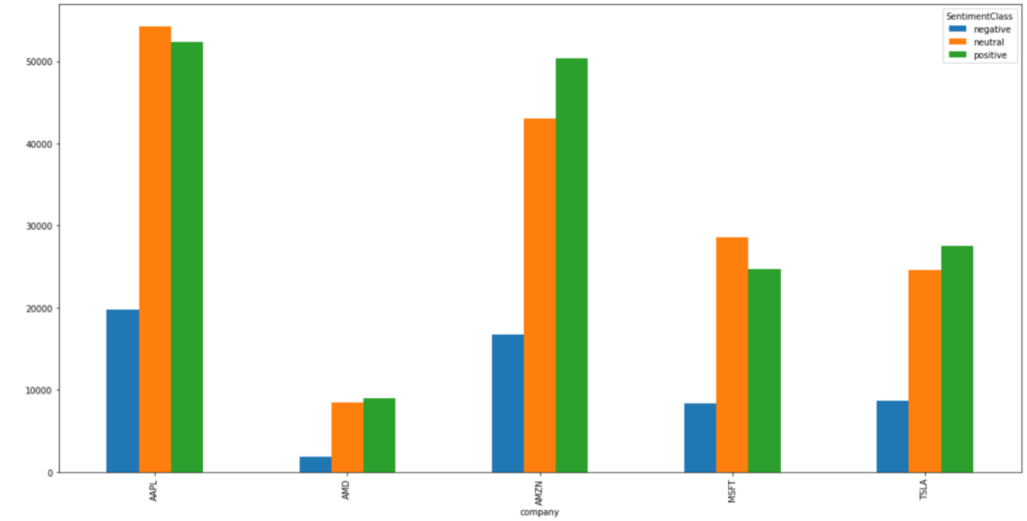


According to the above figure, we can see that both neutral and positive are close to 43%, and the number of tweets defined as negative is relatively small, accounting for only about 15%.

Then, we want to understand more details of the Twitter distribution of each company's three classes. So, we use the company name as the index and the three levels of positive, negative and neutral as the columns to make a pivot table. Based on the table, we use a histogram to show the sentiment classes distribution of each company's Twitter.

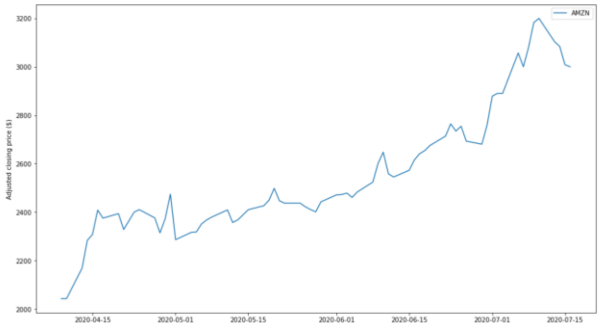


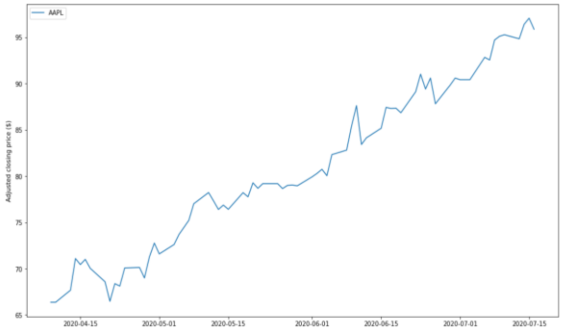
SentimentClass	negative	neutral	positive
company			
AAPL	19827	54225	52311
AMD	1904	8437	8950
AMZN	16701	43081	50396
MSFT	8423	28632	24731
TSLA	8670	24637	27529

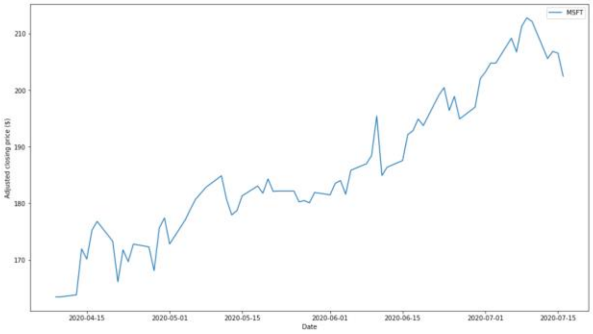


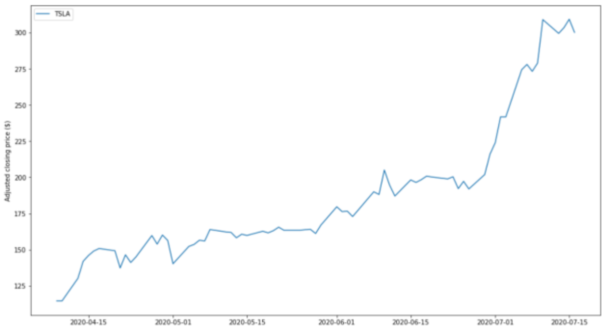
Stock Price Analysis

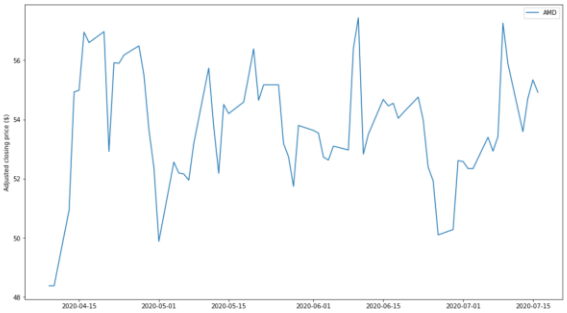
After preprocessing the stock prices of the five companies, we get the time series of each company, and then draw the line charts of stock prices for all five companies in this period with the date as the X axis and the adjusted price as the Y axis. Stock price plots are shown as below:



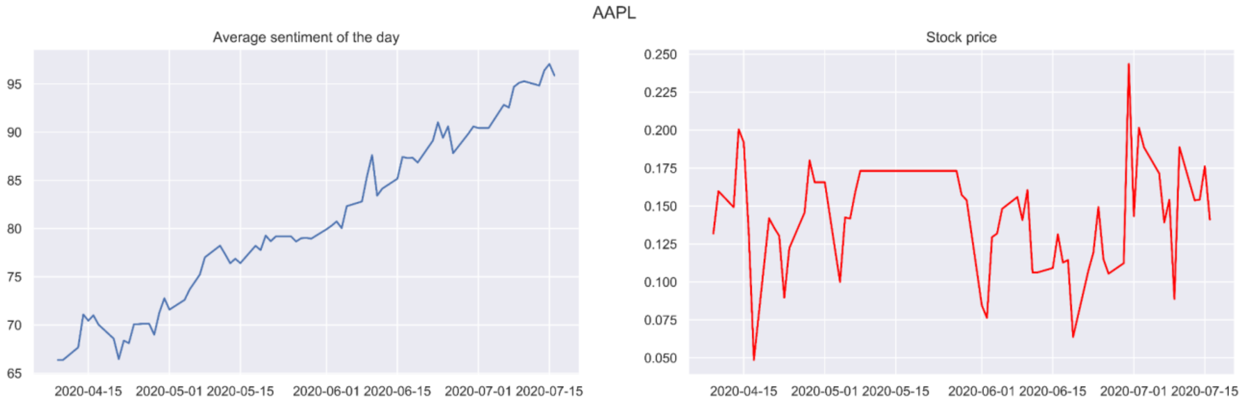




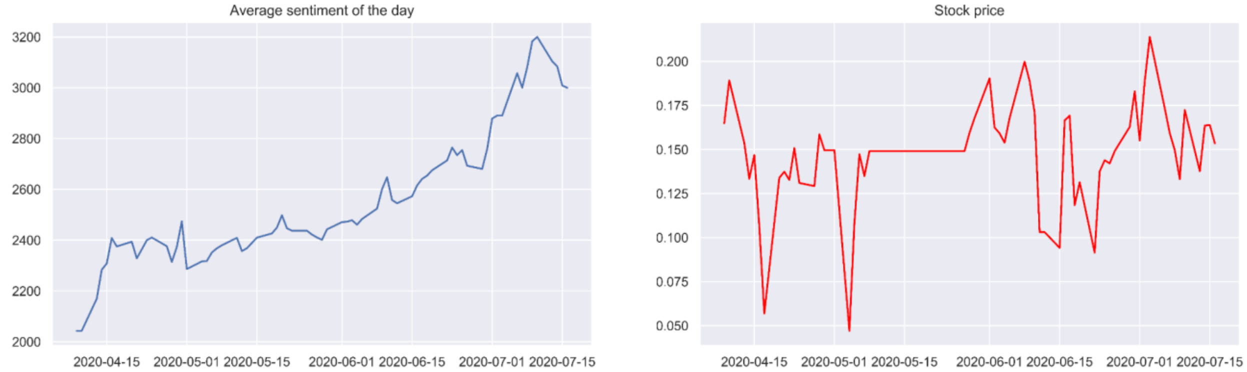




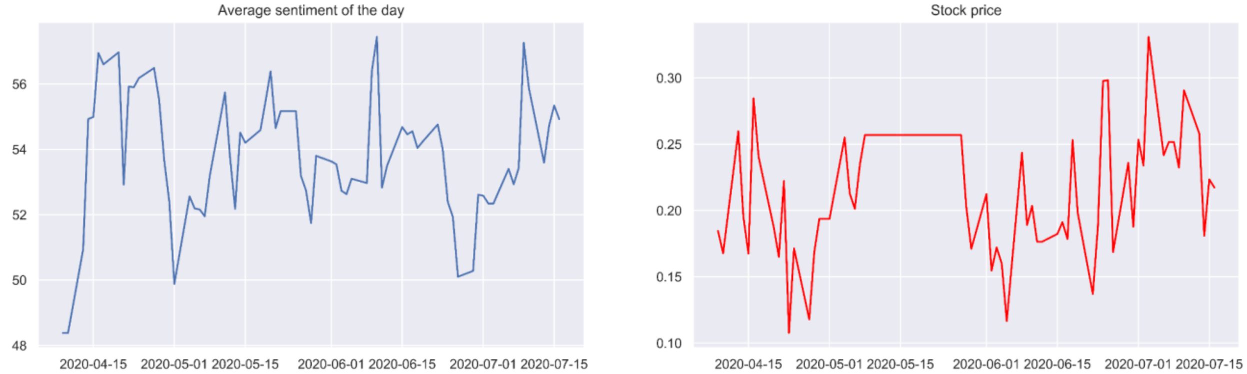
Now, we want to understand the trend of the compound score of each company's tweets during this period. Therefore, we set up a pivot table with date as index, company name as columns, and took the average value compound for every date. Then we draw sentiment line graphs below for these five companies.



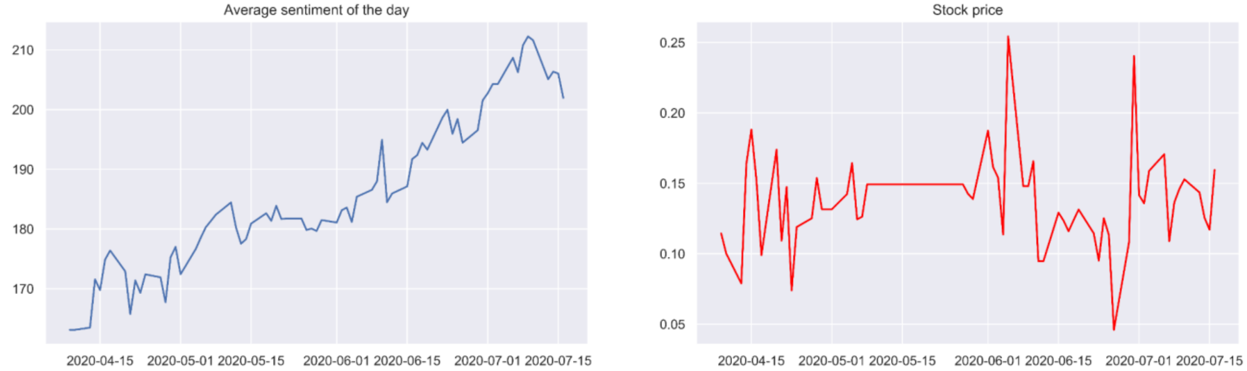
AMZN



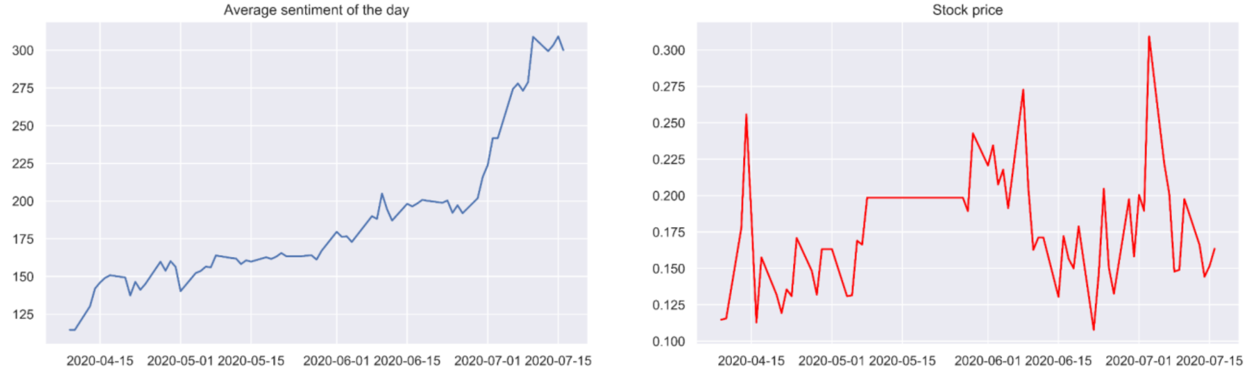
AMD



MSFT



TSLA

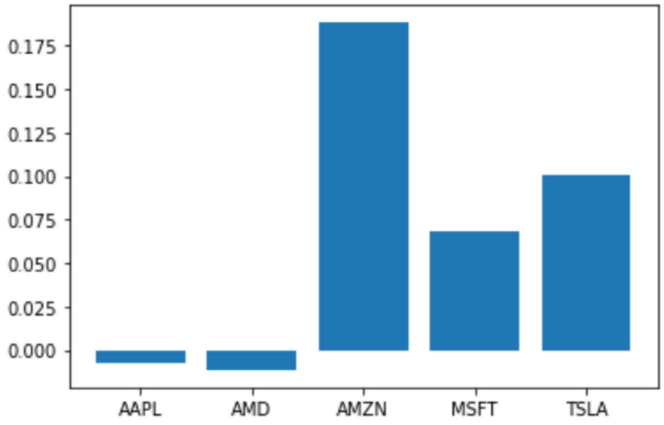


Correlation between Stocks and Sentiment

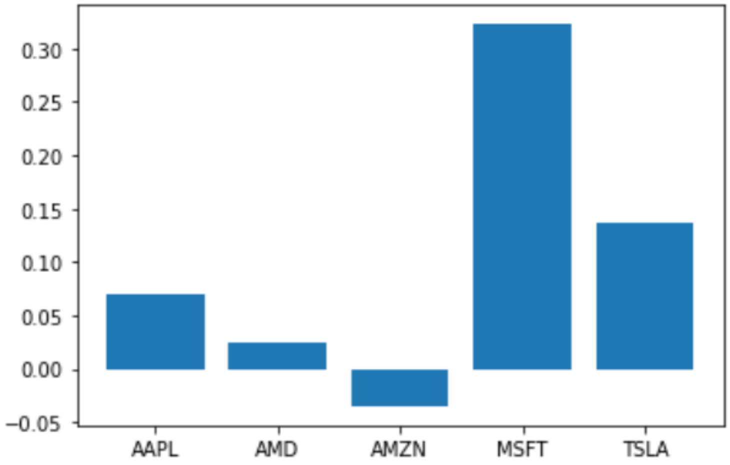
According to the above line chart of stock prices and average sentiment, the relationship between the two is not very intuitively displayed, so it is difficult for us to conclude whether the two are related or how they are related.

- Firstly, we try to search for the relationship between stock price and sentiment with correlation analysis.
- To better dig out the correlation inside for sentiment and stock price, we then make a correlation analysis about the percentage change for close price and sentiment score.
- At last, similarly we analyze the correlation again after taking one day lag. Three outputs are shown as below:

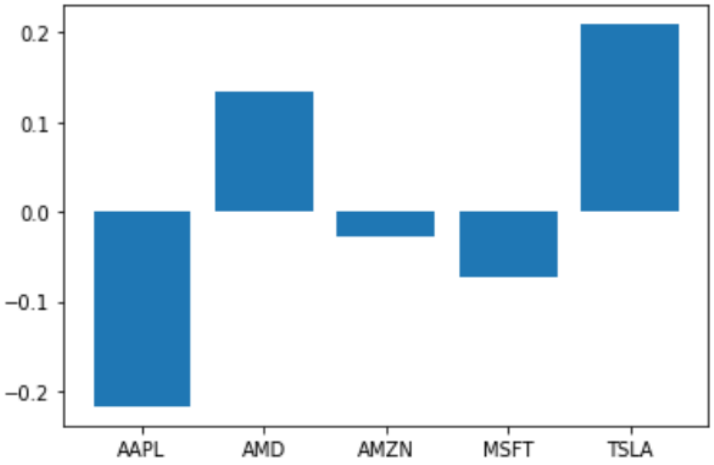
Correlation between sentiment and Stock price:



Taking percentage Change for both close price and sentiment score:



Taking 1 day lag of percentage change in sentiment with respect to percentage change in price

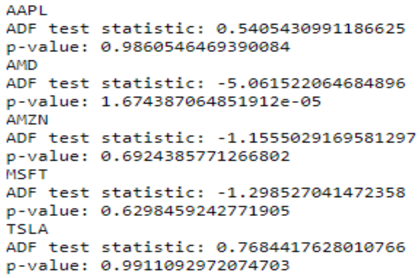


Price Forecasting

In time series forecasting, we generally use past values of a time dependent variable to forecast its future values. But in this project, we also want to consider an exogenous factor which is sentiment of tweets along with stock prices. This classifies our analysis as a multivariate time series analysis. We generally use multivariate time series analysis to model and explain the interesting interdependencies and co-movements among the variables. We applied a multivariate time series method, called Vector Auto Regression (VAR) on this dataset.

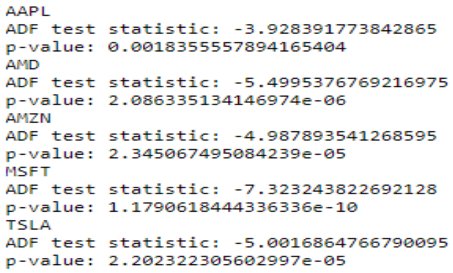
The Vector Auto Regression (VAR) model is a stochastic process that represents a group of time-dependent variables as a linear function of their own past values and the past values of all the other variables in the group.

Before applying VAR, both the time series variable should be stationary. Both the series are not stationary since both the series do not show constant mean and variance over time. We can also perform a statistical test like the Augmented Dickey-Fuller test (ADF) to find stationarity of the series using the AIC criteria.



```
AAPL
ADF test statistic: 0.5405430991186625
p-value: 0.9860546469390084
AMD
ADF test statistic: -5.061522064684896
p-value: 1.674387064851912e-05
AMZN
ADF test statistic: -1.1555029169581297
p-value: 0.6924385771266802
MSFT
ADF test statistic: -1.298527041472358
p-value: 0.6298459242771905
TSLA
ADF test statistic: 0.7684417628010766
p-value: 0.9911092972074703
```

Result of ADF test on stock prices



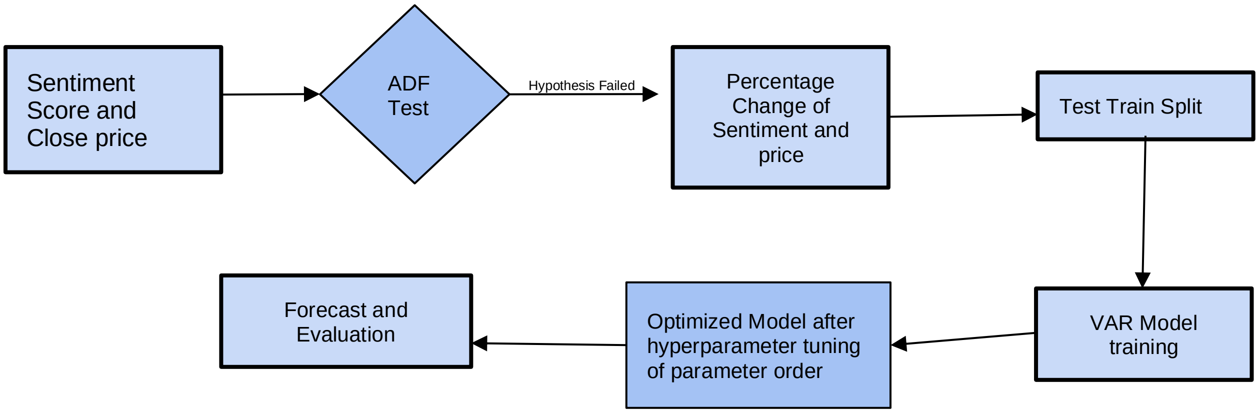
```
AAPL
ADF test statistic: -3.928391773842865
p-value: 0.0018355557894165404
AMD
ADF test statistic: -5.4995376769216975
p-value: 2.086335134146974e-06
AMZN
ADF test statistic: -4.987893541268595
p-value: 2.345067495084239e-05
MSFT
ADF test statistic: -7.323243822692128
p-value: 1.1790618444336336e-10
TSLA
ADF test statistic: -5.0016864766790095
p-value: 2.202322305602997e-05
```

Result of ADF test on stock prices

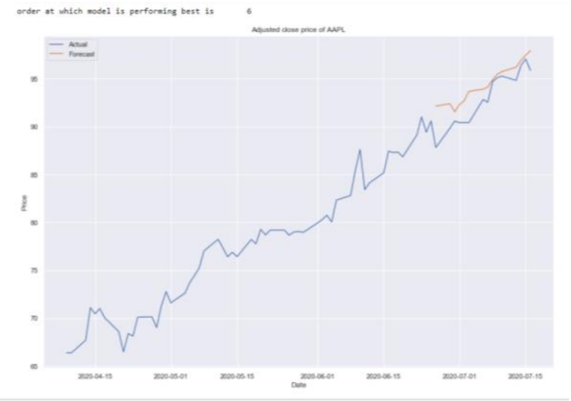
In both cases, the p-value is not significant enough, meaning that we cannot reject the null hypothesis and conclude that the series are non-stationary.

As both the series are not stationary, we perform differencing. We will use the percentage change that we have calculated earlier.

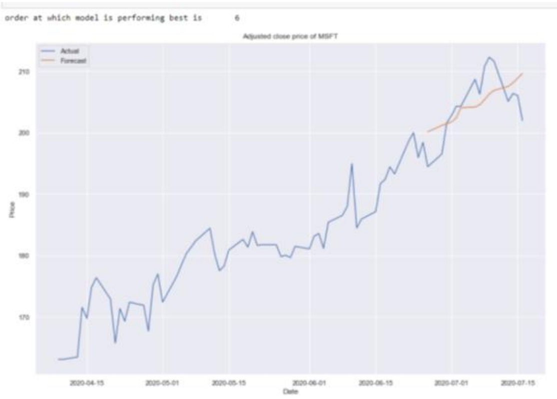
We carry-out the train-test split of the data and keep the last 15-days as test data and train VAR model with the training data. To get an optimized model we have to hyper tune parameters of the VAR model. In this case we have to optimize the parameter named order. We used MAPE (Mean Absolute Percentage Error) as our evaluation metric for hyperparameter tuning and evaluation. We finally forecasted the future prices using the optimized model and evaluated using MAPE.


Sentiment Score and Close price
ADF Test
Hypothesis Failed
Percentage Change of Sentiment and price
Test Train Split
VAR Model training
Optimized Model after hyperparameter tuning of parameter order
Forecast and Evaluation

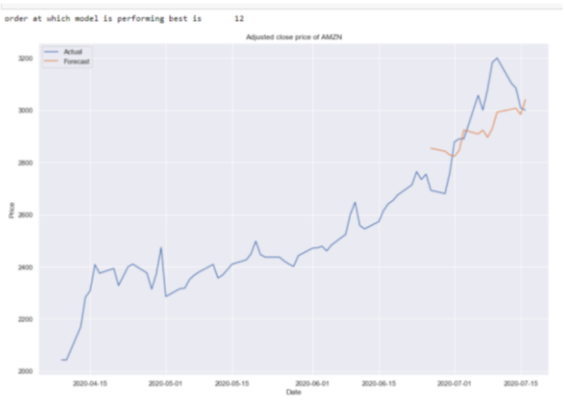
Plots of forecasted stock prices are below:



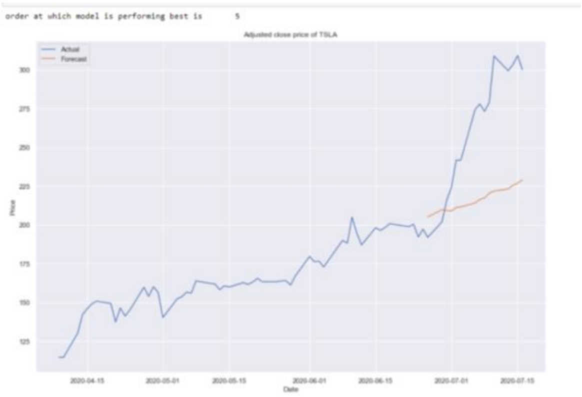
```
order at which model is performing best is      6
```



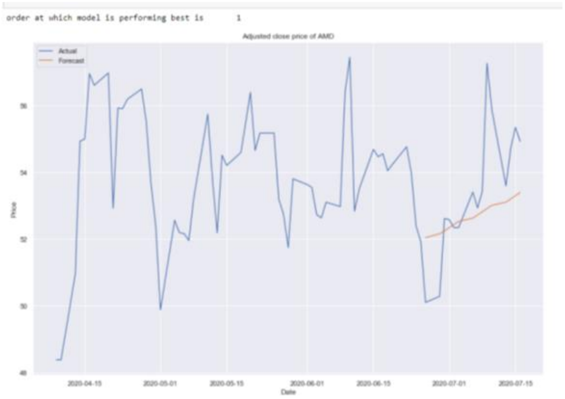
```
order at which model is performing best is      6
```



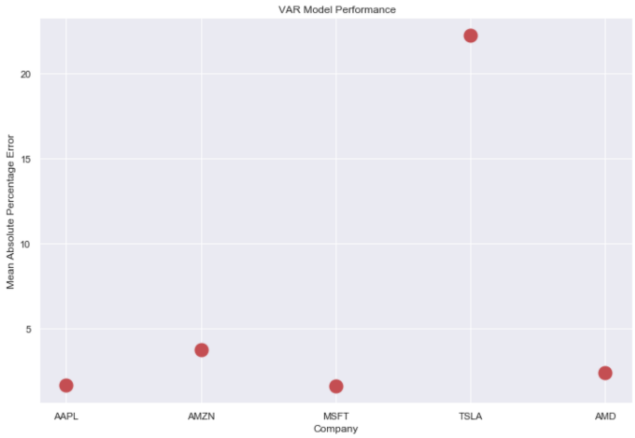
```
order at which model is performing best is      12
```



```
order at which model is performing best is      5
```



```
order at which model is performing best is      1
```



VAR Model Performance

Except Tesla, MAPE (Mean Absolute Percentage Error) on price forecasts are low and acceptable.

V. Conclusion

We performed multiple analyses and experiments on both tweets and stocks price data. From sentiment analysis on tweets, it was seen that there are low percentage counts of negative tweets (15%). In addition to that, stock of all companies has a positive trend, this hints that positive market sentiment has pushed up the stock prices.
Although from the correlation values, it was not very much evident that stock price and compound sentiment score are strongly correlated. This may be due to non-linearity in compound score as the correlation computation assumption signifies the linear vector.
Forecasting using the VAR model with sentiment and stock prices resulted in acceptable error in all the companies except Tesla.

VI. Reference

https://towardsdatascience.com/multivariate-time-series-forecasting-456ace675971

https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/

https://towardsdatascience.com/a-comprehensive-guide-to-downloading-stock-prices-in-python-2cd93ff821d4

https://einvestingforbeginners.com/stock-correlation-daah/