# Stock Market Analysis with Twitter Sentiment Analysis

**FINAL PROJECT REPORT**

**Team Members:**

**Hardeep Singh**

**Vignesh Sriram**

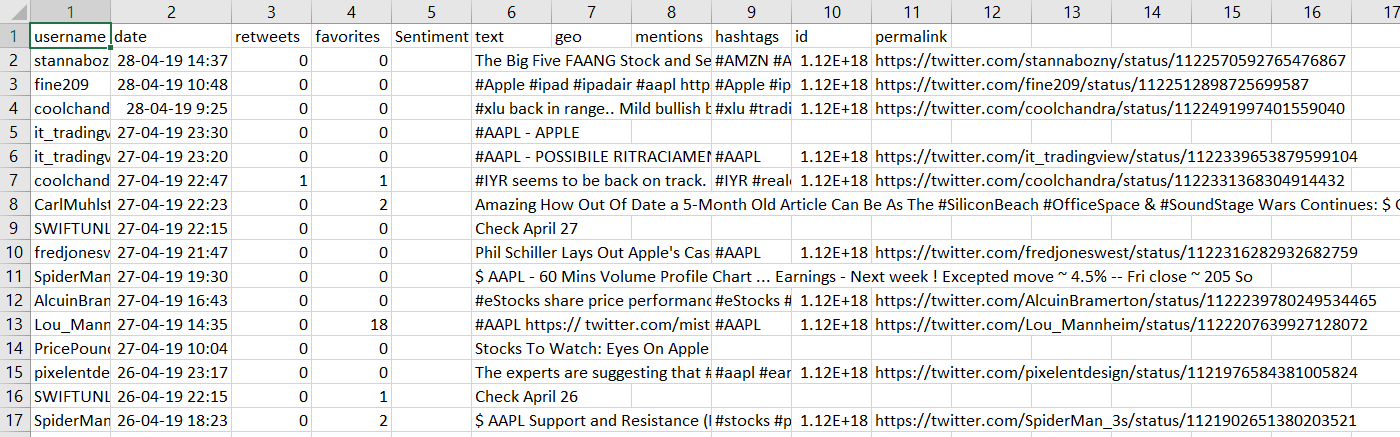
**Sarita Shinde**

**Introduction:**

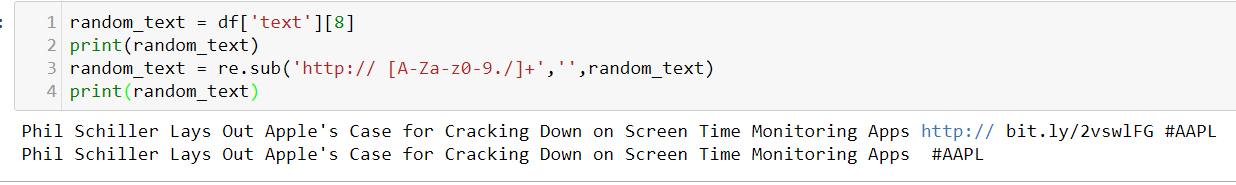
Volatility is one of the main problems which is faced by the stock markets. But having an understanding of the market can open opportunities for profitable investments. As of now there are several factors which tend to influence how the stock market fluctuates. One of It being human interactions on social media. Our project wants to explore how social media(in our case Twitter) influences the stock of a company. Sentiment analysis is how we understand the polarity of a written document, sentence or a phrase. Our final goal is to deduce whether the expression is positive, negative or neutral. We will use various Natural Language Processing tools to analyze this. One of the challenges were that Twitter API cannot fetch enough data. We used a web scrapper to get data required for the project.

We decided to use Twitter as the social media for the analysis because it’s a famous microblogging platform with more than 24 million active users. The users range from a school going kid to celebrities around the world. Twitter has a 280 character limit on its text which makes the data concise and easy to handle and it is a famous platform for people to share and discuss their personal views on a variety of subjects.

Our initial data was obtained using web scraping tools and looked like this.

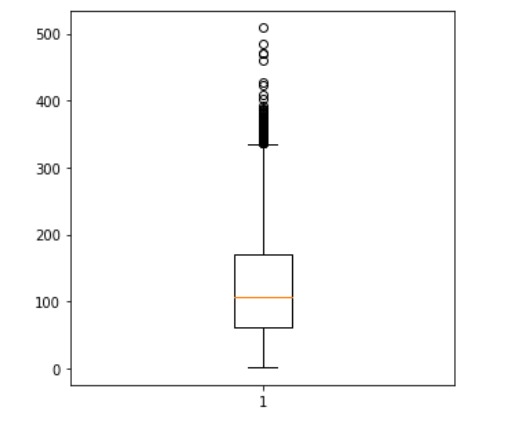


One of our first steps in data cleaning was to remove a @ mentions which are a crucial part of tweeting but do not contribute to the sentiment analysis in any way. We also removed URL links that we included in the tweet because we couldn’t use it for our evaluation.

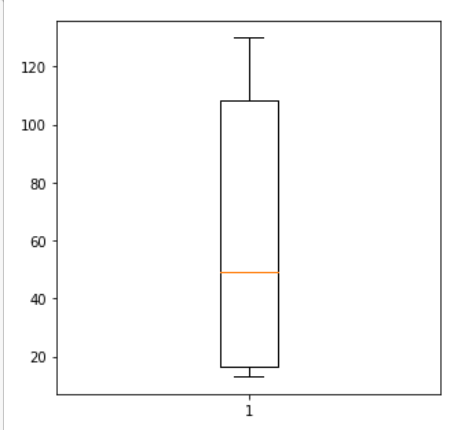


**Exploratory Data Analysis:**

After the initial data processing, we created a box plot to see the difference between the number of characters involved in a tweet in general. The box plot looked as follows before cleaning-



And after the cleaning process, it looked like this-



We observe that the average length of tweets reduces considerably when we get rid of the outliers

**Model Selection and Analysis:**

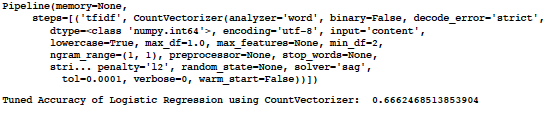
Machine learning algorithms require numeric features for functioning. For performing machine learning on text, we need to convert the text into a vector representation of the value of the text. This is the vectorization process. We have taken into consideration 2 types of vectorizers: Count Vectorizer and TD-IDF Vectorizer.

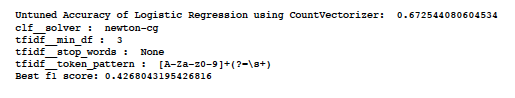
Count Vectorizer- Count Vectorizer tokenizes each message in a document to count the number of times a word occurs in the document. Count Vectorizer is useful when used with neural networks. It also helps to build a vocabulary of known words in the document. The vector that is created has the dimensions equal to the size of the vocabulary. It denotes a values for every time a word is repeated and we will get an accurate count of the words in our text.

TF-IDF Vectorizer- TF-IDF stands for Term Frequency-Inverse Document Frequency. Count Vectorizer evaluates word in standalone fashion not considering their overall effect on the corpus. A better technique is to count the relative frequency of a word in the document against their occurrence n other documents. That is exactly what TF-IDF vectorizer does. The value of the vector goes up increases based on their occurrence in one document but decreases the score if they repeat a lot in other documents too.

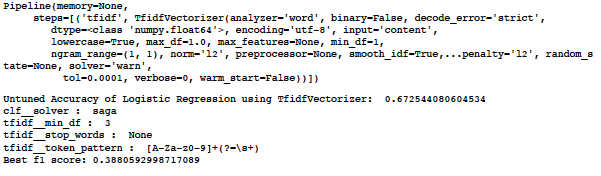
Now we run each of the following models with the count vectorizer as well as the TF-IDF vectorizer to find the model with the highest accuracy.

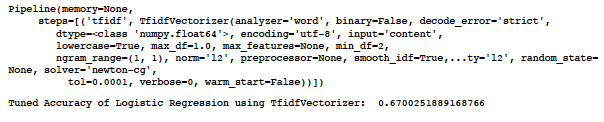
Logistic Regression+CountVectorizer



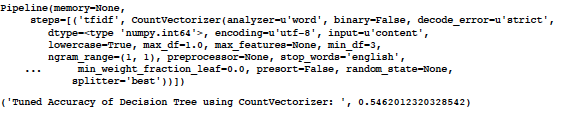


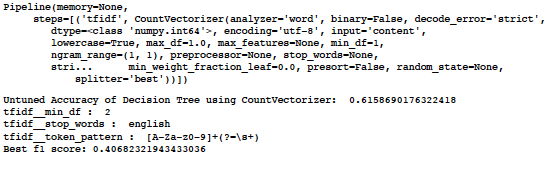
Logistic Regression+TF-IDF Vectorizer



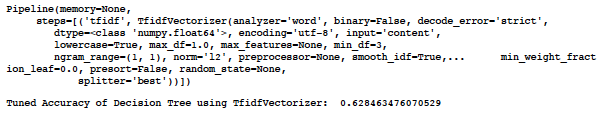


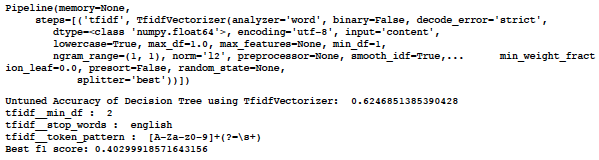
Decision Tree+CountVectorizer



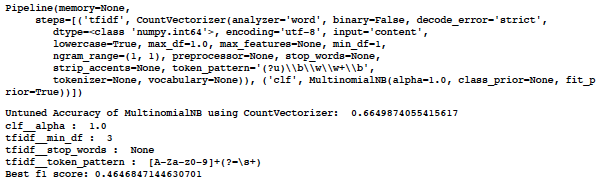


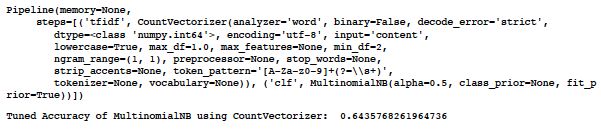
Decision Tree+ TF-IDF Vectorizer



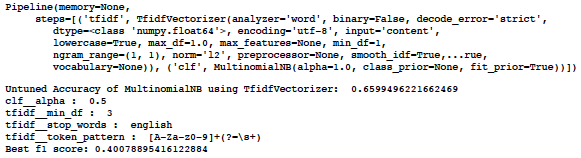


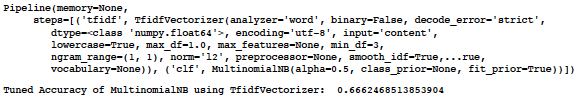
Multinomial Naïve Bayes+ CountVectorizer



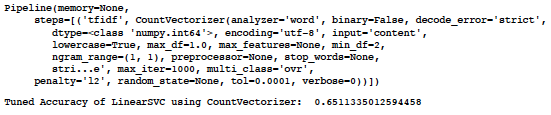


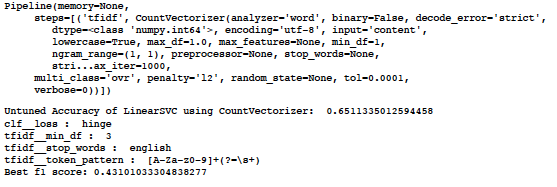
Multinomial Naïve Bayes+TF-IDF Vectorizer



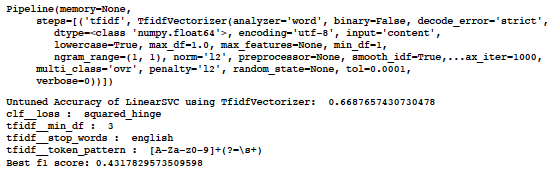


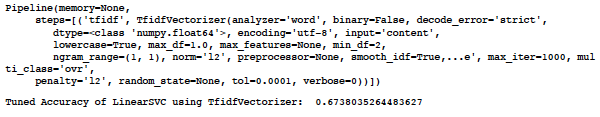
Linear SVC+Count Vectorizer



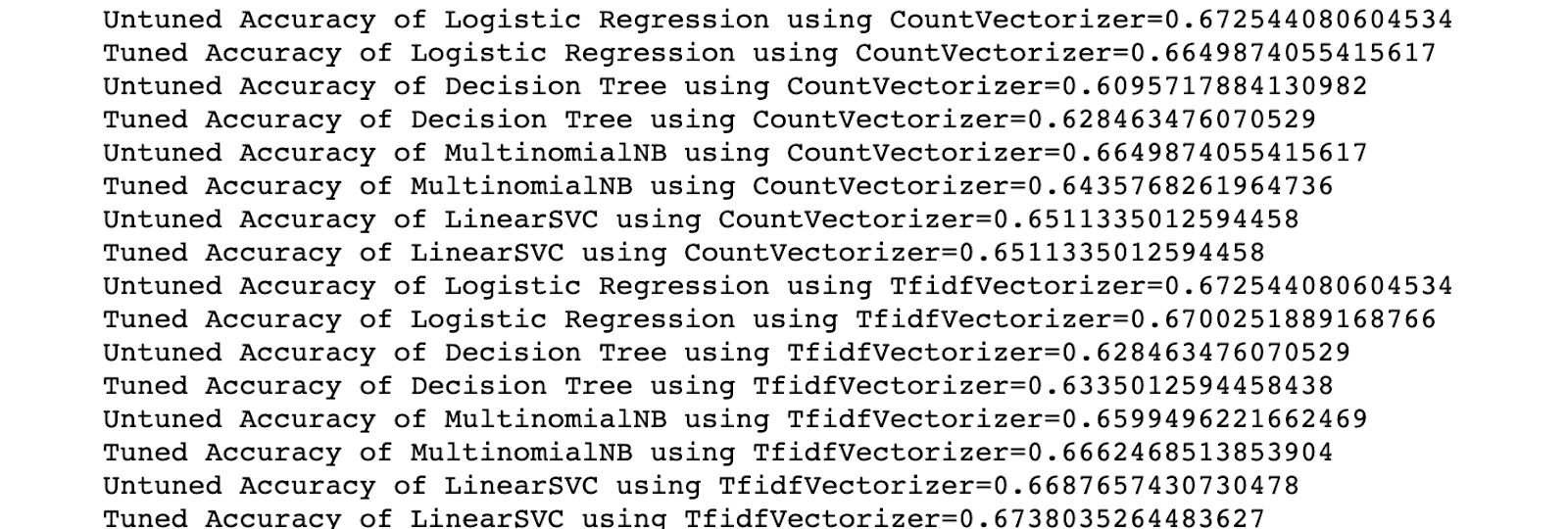


Linear SVC+TF-IDF Vectorizer

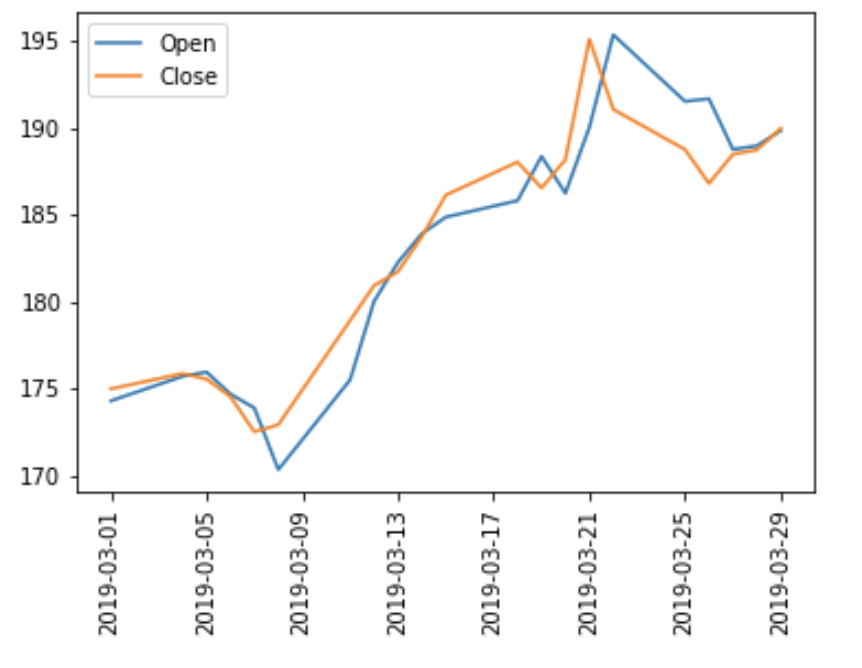


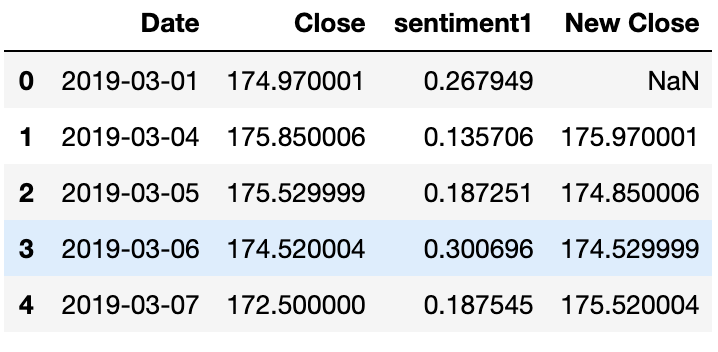


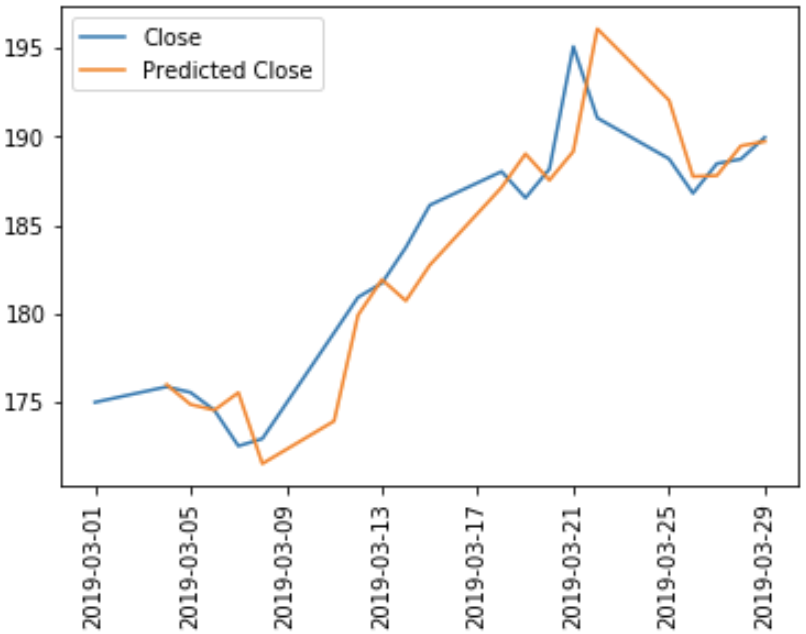
**Model Comparison:**



**Stock Prediction Based on Sentiment Analysis:**







Conclusion and Future Scope:

As we can see from the above results, we get the best results using LinearSVC using TfidfVectorisor

Currently our Sentiment analysis model is capable of only predicting the stock prices for the next day, but we can improve this by fetching news from a larger stock.

Also, what we noticed while we fetched the data was that a lot of the tweets were made by common people who didn’t have much impact. On the other hand, there were a very less number of people whose tweets were influential. The way to differentiate these tweets could be the use of something called as retweet. In the future we can assign a higher weightage to the tweets which have higher retweets.

Also a lot of users prefer using emoticons which were not taken into consideration by our model. But this is one untapped territory which has a lot of potential to improve the quality of sentiment analysis.