Sentiment Analysis on Stock Market using Twitter Data

Project proposal

Class: CSCE 5290 Section 002

Group #5

Team Members

Sweety Makwana

Gopi Krishna Sure

Gayathri Katukojwala

Nikhil Rangineni

GitHub Repository: https://github.com/Smakwana30/CSCE-5290-NLP-Project/blob/main/Project%20Proposal%20-%20Sentiment%20Analysis%20on%20Stock%20Market%20using%20Tweets.pdf

Motivation

Twitter is a very well-known place where traders tweet about stocks and financial instruments they care about. Their tweets reveal their sentiment about the stocks they are tweeting. Which can help decide overall trend for the particular stock if analyzed correctly. The stock market which is seen relatively high volatile in recent years could provide chances for making money or losing money. There are various factors that affects the direction of the stock market, or a single stock as follows:

- fundamental of the publicly traded company
- technical indicators
- news and media release for the company
- public sentiment - social media posts such as Twitter, Facebook, etc.

While the field of AI has grew tremendously in recent times, an AI engineer could take advantage of available AI technologies to help predict the trend of stock market by analyzing sentiment of traders from twitter.

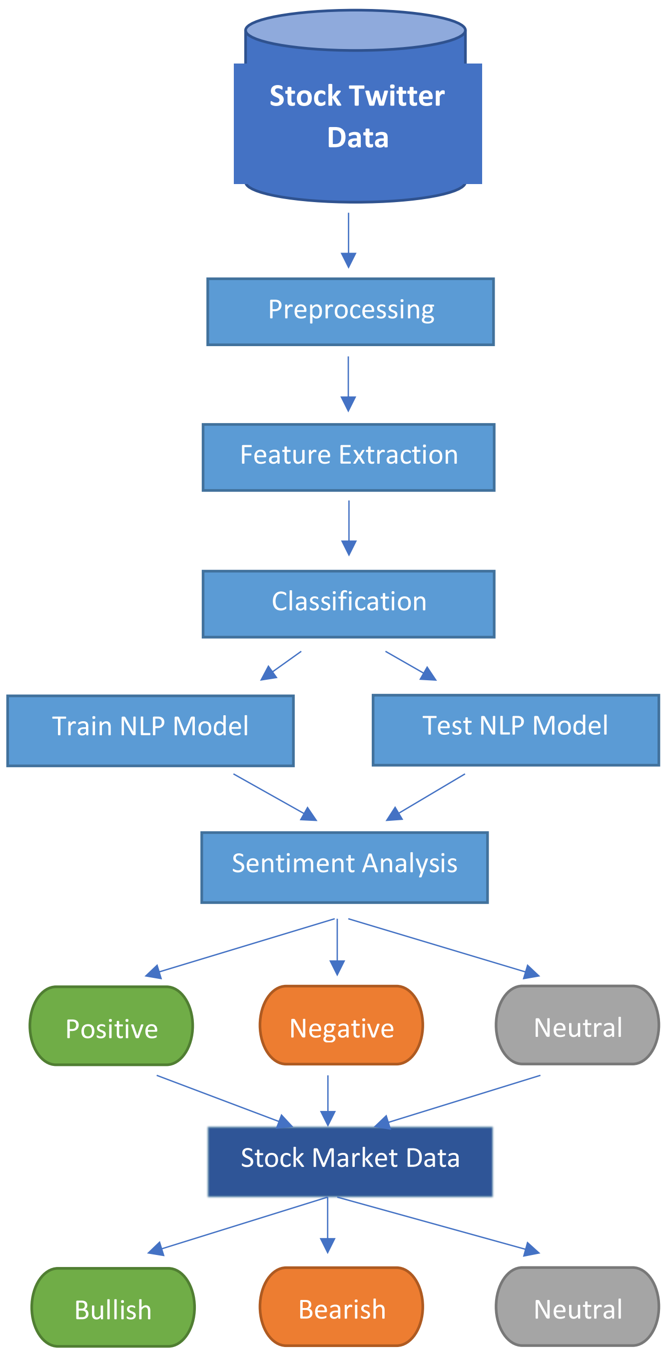
Significance

As Abraham Lincoln said, "With public sentiment nothing can fail. Without it, nothing can succeed." This also applies to the stock market. If public sentiment is positive about the stock market it means the stock market is expected to trend in a bullish manner, else if the public sentiment is negative, once could expect the stock market to trend in a bearish manner. If the sentiment is neutral, then stock market may not see much volatility and may trend neutral. Twitter data is linguistic and unstructured data which can be translated into structured and meaningful data using NLP models. Public sentiment could be classified into three categories as: positive, negative and neutral via text classification and text summarization tools available under the umbrella of Machine Learning field. Without the help of NLP model, the trader needs to read each and every tweet about the particular stock, analyze them manually and then decide the trend of the stock based on the sentiment score he/she comes up with. Not to forget here, the stock market data is constantly changing and so public sentiment about it. It is a very tedious process for anyone to manually keep track of all the tweets and analyze them constantly to help predict the stock prices.

Objectives

To overcome the problems faced by traders and help them decide the trend of a particular stock price using twitter data, our team would like to propose a project to build an NLP based model to derive public sentiment from tweets. Which would help traders to decide the trend of the stock and prevent financial losses. We would acquire pre-labeled Twitter data for a particular financial instrument such as, Apple (AAPL), Tesla (TSLA), Amazon (AMZN), or S&P500 index from Kaggle to train and validate the NLP model. We would obtain historical data for an interested stock from Yahoo Finance to observe possible co-relation between the tweet sentiment analysis and stock prices. The historical data for stock would include daily price for Open, High, Low, Close, Volume and the date. We would yet to incorporate a type of the classifier to classify twitter data. We may integrate appropriate neural network such as long Short-term Memory (LSTM) or Self Organizing Fuzzy Neural Networks (SOFNN).

Workflow



First, we would collect our datasets and visualize it. By using Google Colab platform, we would start implementing data preprocessing and building NLP model to derive sentiments via stock twitter data. Following train, test, and validation the NLP model should be able to detect sentiment of a particular stock or indexes and associated price prediction as such if the sentiment is positive, the stock is poised to be bullish; sentiment is negative, the stock is poised to be bearish; and if the sentiment is neutral, the stock would experience zero or low volatility in short to medium term.

Features:

- Visualize Twitter data
- Preprocess and classify Tweets using NLP model: Positive, Negative, Neutral
- Predict price for a particular stock using public (Tweet) sentiments
- Provide speed and scalability for financial decision to make profit and minimize losses
- Accuracy
- Visualize stock price chart along with Twitter Sentiment Polarity to predict the trend of the stock

Project Increment

Background

One of the most significant areas of study in academic and professional circles is stock price prediction. Various data mining techniques are often used in research. But the machine learning/deep learning technique gives a more accurate, precise and easier way to solve such stock and market price questions.

Now-a-days, information about public opinions is vastly available on social media. This is acting as a platform where public shares their emotional ideas and feelings which can potentially impact overall stock market and/or particular individual stock price. Users of the microblogging service Twitter can follow and comment on the ideas of other users and express their own opinions in real time. More than a million users send more than 140 million tweets every day. Estimation investigation of twitter information and sentiment classification is the assignment of judging conclusion in a chunk of content as positive, negative or unbiased.

In this project a method for predicting stock prices is developed using Twitter tweets about various companies. There are many scholarly research done in this are to predict sentiment of mass people using their public posts from Twitter and other social media. To name a couple of these research articles:

1. Go, Alec, Lei Huang, and Richa Bhayani. "Twitter sentiment analysis." *Entropy* 17 (2009): 252.
   This paper was presented in 2009 when Twitter was a new microblogging platform where users share their sentiments on any related topics including stock prices. They developed a machine learning algorithm to classify tweets for marketers and reviews for the company.

2. Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!." *Proceedings of the international AAAI conference on web and social media*. Vol. 5. No. 1. 2011.
   This paper investigates usefulness of sentiment analysis on Twitter data as being inspired from other domains where strong link has been found between part-of-speech tagging and sentiment lexicons in other domains.

Dataset:

We are using the Twitter API to collect our datasets for the purpose of categorizing the tweets into positive, neutral, and negative tweets later using the NLP models we build along the process. We have taken the keys to access the twitter APIs and then search the tweets based on specific keywords and then add them to a text file to feed into our model for pre-processing and perform sentimental analysis.

In the dataset, we have searched the keywords 'Tesla', '#TSLA', and combination of the words 'YahooFinance and tesla'. As we have seen that there are different types of tweets related to the Tesla stock, so we filtered the tweets based on the above keywords.

We have stored the filtered tweets in a text file to perform analysis and pre-processing later which is used to feed the model with the processed data.

As seen in Fig. 1 below, 3 code cells are used to fetch the tweets from the twitter, we have set the limit of 1000 and then used the 'search_tweets' method of the twitter to fetch the API and passed the parameter to the items method. Later it converted to the cursor object to iterate over the tweets. Later we have opened a file and written the tweets into the file line by line and after finishing we closed the file object.

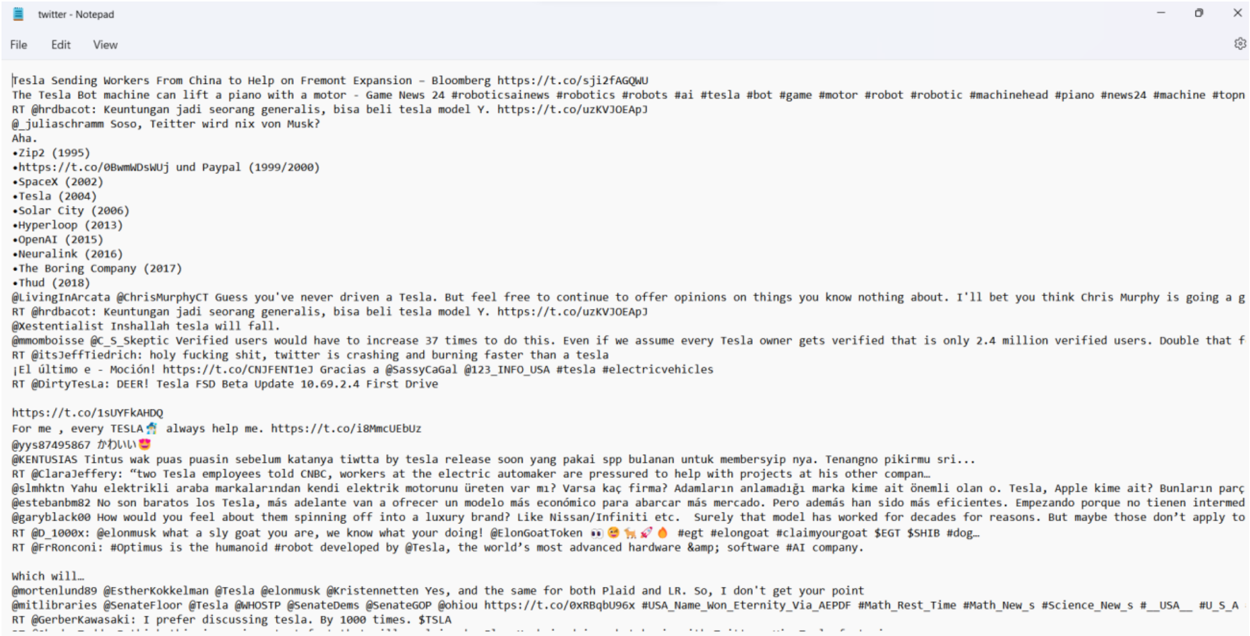
```
[6] #code to search the Tweets with keyword tesla and add them to a text file
    keyword = 'Tesla'
    limit=1000
    tweets = twep.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
    file = open('twitter.txt', 'w', encoding="utf-8")
    for tweet in tweets:
        file.write(tweet.full_text+'\n')
    file.close()
```

```
[7] #code to search the Tweets with keyword #TSLA and add them to a text file
    keyword = '#TSLA'
    limit=1000
    tweets = twep.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
    file = open('twitter.txt', 'a', encoding="utf-8")
    for tweet in tweets:
        file.write(tweet.full_text+'\n')
    file.close()
```

```
[8] #code to search the Tweets with keyword yahoofinance and Tesla and add them to a text file
    keyword = 'yahoofinance and Tesla'
    limit=1000
    tweets = twep.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
    file = open('twitter.txt', 'a', encoding="utf-8")
    for tweet in tweets:
        file.write(tweet.full_text+'\n')
    file.close()
```

Fig. 1: Codes for Data Collection

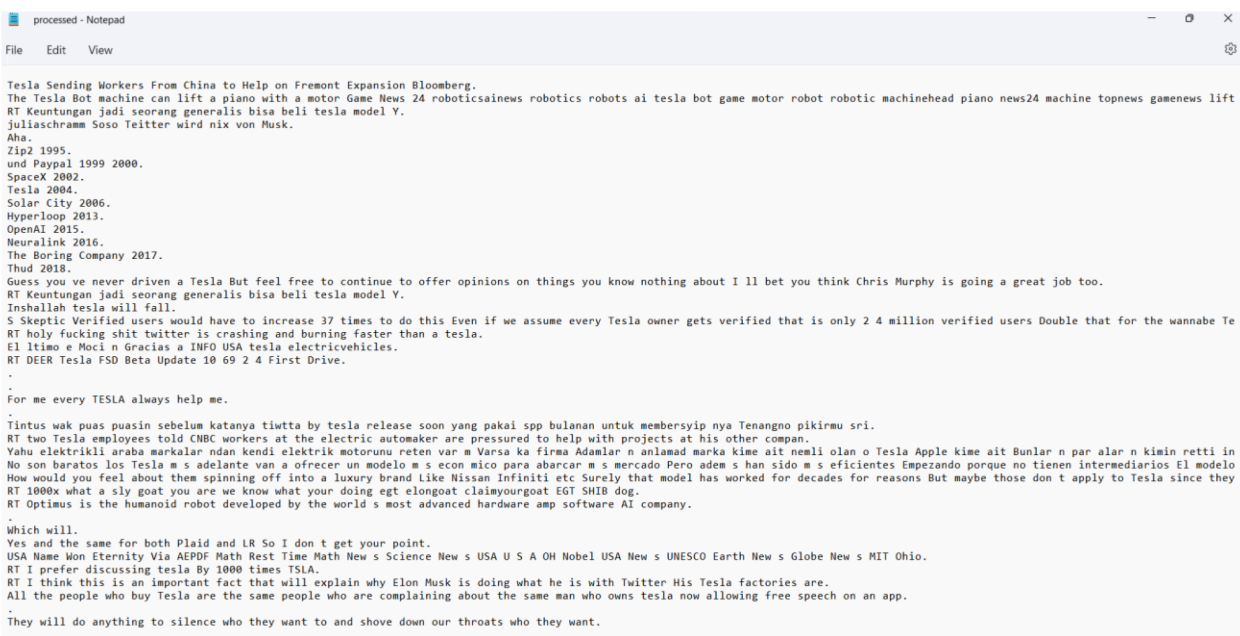
Below images Fig. 2. shows the snapshot of our datasets as originally abstracted tweets (twitter.txt) on Tesla company.



```
twitter - Notepad
File Edit View

Tesla Sending Workers From China to Help on Fremont Expansion - Bloomberg https://t.co/sji2fAGQWU
The Tesla Bot machine can lift a piano with a motor - Game News 24 #roboticsainews #robotics #robots #ai #tesla #bot #game #motor #robot #robotic #machinehead #piano #news24 #machine #topn
RT @hrdbacot: Keuntungan jadi seorang generalis, bisa beli tesla model Y. https://t.co/uzKVJOEApJ
@_juliaschramm Soso, Teitter wird nix von Musk?
Aha.
•Zip2 (1995)
•https://t.co/0BwmWDsWUj und Paypal (1999/2000)
•SpaceX (2002)
•Tesla (2004)
•Solar City (2006)
•Hyperloop (2013)
•OpenAI (2015)
•Neuralink (2016)
•The Boring Company (2017)
•Thud (2018)
@LivingInArcata @ChrisMurphyCT Guess you've never driven a Tesla. But feel free to continue to offer opinions on things you know nothing about. I'll bet you think Chris Murphy is going a g
RT @hrdbacot: Keuntungan jadi seorang generalis, bisa beli tesla model Y. https://t.co/uzKVJOEApJ
@Xestentialist Inshallah tesla will fall.
@mmomboisse @C_S_Skeptic Verified users would have to increase 37 times to do this. Even if we assume every Tesla owner gets verified that is only 2.4 million verified users. Double that f
RT @itsJeffTiedrich: holy fucking shit, twitter is crashing and burning faster than a tesla
¡El último e - Moción! https://t.co/CNJFENT1eJ Gracias a @SassyCaGal @123_INFO_USA #tesla #electricvehicles
RT @DirtyTesLa: DEER! Tesla FSD Beta Update 10.69.2.4 First Drive

https://t.co/1sUYFkAHDQ
For me , every TESLA always help me. https://t.co/i8MmcUEbUz
@yys87495867 かわいい😍
@KENTUSIAS Tintus wak puas puasin sebelum katanya tiwtta by tesla release soon yang pakai spp bulanan untuk membersyip nya. Tenangno pikirmu sri...
RT @ClaraJeffery: “two Tesla employees told CNBC, workers at the electric automaker are pressured to help with projects at his other compan…
@slmhktn Yahu elektrikli araba markalarından kendi elektrik motorunu üreten var mı? Varsa kaç firma? Adamların anlamadığı marka kime ait önemli olan o. Tesla, Apple kime ait? Bunların parç
@estebanbm82 No son baratos los Tesla, más adelante van a ofrecer un modelo más económico para abarcar más mercado. Pero además han sido más eficientes. Empezando porque no tienen intermed
@garyblack00 How would you feel about them spinning off into a luxury brand? Like Nissan/Infiniti etc.  Surely that model has worked for decades for reasons. But maybe those don’t apply to
RT @D_1000x: @elonmusk what a sly goat you are, we know what your doing! @ElonGoatToken  #egt #elongoat #claimyourgoat $EGT $SHIB #dog…
RT @FrRonconi: #Optimus is the humanoid #robot developed by @Tesla, the world’s most advanced hardware &amp; software #AI company.

Which will…
@mortenlund89 @EstherKokkelman @Tesla @elonmusk @Kristennetten Yes, and the same for both Plaid and LR. So, I don't get your point
@mitlibraries @SenateFloor @Tesla @WHOSTP @SenateDems @SenateGOP @ohiou https://t.co/0xRBqbU96x #USA_Name_Won_Eternity_Via_AEPDF #Math_Rest_Time #Math_New_s #Science_New_s #__USA__ #U_S_A
RT @GerberKawasaki: I prefer discussing tesla. By 1000 times. $TSLA
```

Fig. 2: Snapshot of raw data collected in ‘twitter.txt’ file

The image below, Fig. 3. shows the snapshot of partially processed data from ‘processed.txt’ file, where we have removed special characters, URLs, Links and separated the sentences.



```
processed - Notepad
File Edit View

Tesla Sending Workers From China to Help on Fremont Expansion Bloomberg.
The Tesla Bot machine can lift a piano with a motor Game News 24 roboticsainews robotics robots ai tesla bot game motor robot robotic machinehead piano news24 machine topnews gamenews lift
RT Keuntungan jadi seorang generalis bisa beli tesla model Y.
juliaschramm Soso Teitter wird nix von Musk.
Aha.
Zip2 1995.
und Paypal 1999 2000.
SpaceX 2002.
Tesla 2004.
Solar City 2006.
Hyperloop 2013.
OpenAI 2015.
Neuralink 2016.
The Boring Company 2017.
Thud 2018.
Guess you ve never driven a Tesla But feel free to continue to offer opinions on things you know nothing about I ll bet you think Chris Murphy is going a great job too.
RT Keuntungan jadi seorang generalis bisa beli tesla model Y.
Inshallah tesla will fall.
S Skeptic Verified users would have to increase 37 times to do this Even if we assume every Tesla owner gets verified that is only 2 4 million verified users Double that for the wannabe Te
RT holy fucking shit twitter is crashing and burning faster than a tesla.
El ltimo e Moci n Gracias a INFO USA tesla electricvehicles.
RT DEER Tesla FSD Beta Update 10 69 2 4 First Drive.
.
.
For me every TESLA always help me.
.
Tintus wak puas puasin sebelum katanya tiwtta by tesla release soon yang pakai spp bulanan untuk membersyip nya Tenangno pikirmu sri.
RT two Tesla employees told CNBC workers at the electric automaker are pressured to help with projects at his other compan.
Yahu elektrikli araba markalar ndan kendi elektrik motorunu reten var m Varsa ka firma Adamlar n anlamad marka kime ait nemli olan o Tesla Apple kime ait Bunlar n par alar n kimin retti in
No son baratos los Tesla m s adelante van a ofrecer un modelo m s econ mico para abarcar m s mercado Pero adem s han sido m s eficientes Empezando porque no tienen intermediarios El modelo
How would you feel about them spinning off into a luxury brand Like Nissan Infiniti etc Surely that model has worked for decades for reasons But maybe those don t apply to Tesla since they
RT 1000x what a sly goat you are we know what your doing egt elongoat claimyourgoat EGT SHIB dog.
RT Optimus is the humanoid robot developed by the world s most advanced hardware amp software AI company.
.
Which will.
Yes and the same for both Plaid and LR So I don t get your point.
USA Name Won Eternity Via AEPDF Math Rest Time Math New s Science New s USA U S A OH Nobel USA New s UNESCO Earth New s Globe New s MIT Ohio.
RT I prefer discussing tesla By 1000 times TSLA.
RT I think this is an important fact that will explain why Elon Musk is doing what he is with Twitter His Tesla factories are.
All the people who buy Tesla are the same people who are complaining about the same man who owns tesla now allowing free speech on an app.
.
They will do anything to silence who they want to and shove down our throats who they want.
```

Fig. 3: Partially Pre-processed data saved in to ‘processed.txt’ file

Detail Design of Features

- Collect the dataset based on specific keywords related to a stock company from Twitter API
- Preprocess the data to remove special characters, URLs, links, stopwords, lemmatize, etc.
- Visualize and analyze Twitter data using "WordCloud", Histogram, Unique words search, etc.
- Build NLP model for sentiment analysis
- classify Tweets using NLP model: Positive, Negative, Neutral
- Predict price for a particular stock using public (Tweet) sentiments *(To be implemented in next increment)*
- Provide speed and scalability for financial decision to make profit and minimize losses
- Accuracy *(To be implemented in next increment)*
- Visualize stock price chart along with Twitter Sentiment Polarity to predict the trend of the stock *(To be implemented in next increment)*

Analysis

In the analysis part we have visualized the text in different formats and did some calculations to get the frequency of most occurred words and the frequency of the length of sentences in the over Train data.

1. In the below screenshot Fig. 4., we have imported the 'WordCloud' class, which is used to display the highest frequency words in different color and words with high frequency have larger size than the lower frequency words. In the below screenshot we have filtered the Positive sentiment words to form a word cloud that are repeated more number of times.

```
In [26]: #created the wordcloud method to display the words with the sentiment values
         from wordcloud import WordCloud
         from matplotlib import pyplot as plt

         def DisplayWordCloud(input,bcol):
             plt.figure(figsize=(10,10))
             wocl=WordCloud(background_color=bcol,max_words=50, min_word_length=2, contour_width=1, contour_color='orange')
             wocl.generate(" ".join(input))
             plt.imshow(wocl)
             plt.axis("off")
```

```
In [27]: DisplayWordCloud(df[df.Sentiment=="positive"].Text, 'white')
```

Fig. 4: Analysing positive unique words from dataset

2. In the below Fig. 5., we have formed the word cloud by filtering the negative sentiment tweets and displayed the words with more frequency with white background.


```
[26] DisplayWordCloud(df[df.Sentiment=="negative"].Text, 'white')
```

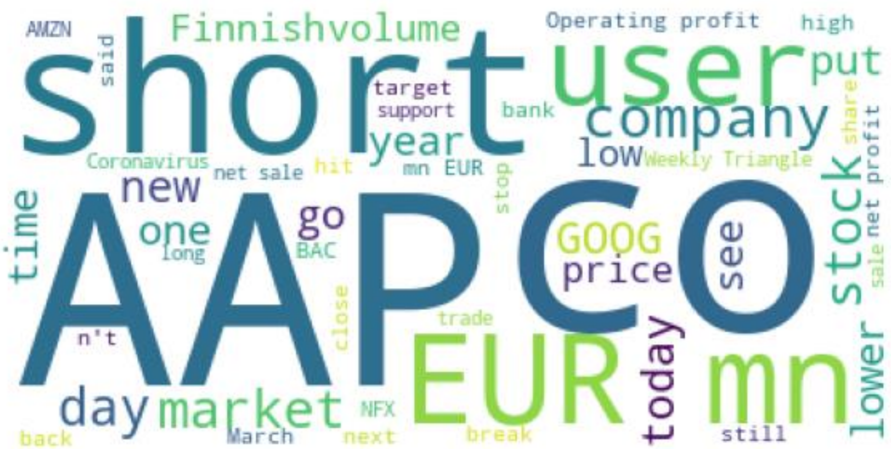


Fig. 5: Analysing Negative unique words from Dataset

3. In the below Fig. 6., we have displayed the word cloud with neutral sentiment text and displayed the words with more frequency and white back ground.

```
[27] DisplayWordCloud(df[df.Sentiment=="neutral"].Text, 'white')
```

Fig. 6: Analysing Neutral unique words from Dataset

4. In the below Fig. 7., we have plotted a bar chart where it shows the frequency of the length of sentences in the whole training corpus. Using the for loop we have found the length of each sentence and then stored them in a dictionary, later the dictionary is ordered to sort the dictionary based on the items values. Finally using the 'plt.bar' method we have displayed the frequency and length of sentences.

```
[68] #frequency of length of sentences in the Train dataset
     from collections import OrderedDict
     freq = {}
     for line in Train_X:
       l=len(line)
       if (l in freq):
         freq[l] += 1
       else:
         freq[l] = 1
     final_dict = OrderedDict(sorted(freq.items()))
     plt.bar(final_dict.keys(), final_dict.values(), 10, color='g')
     plt.show()
```

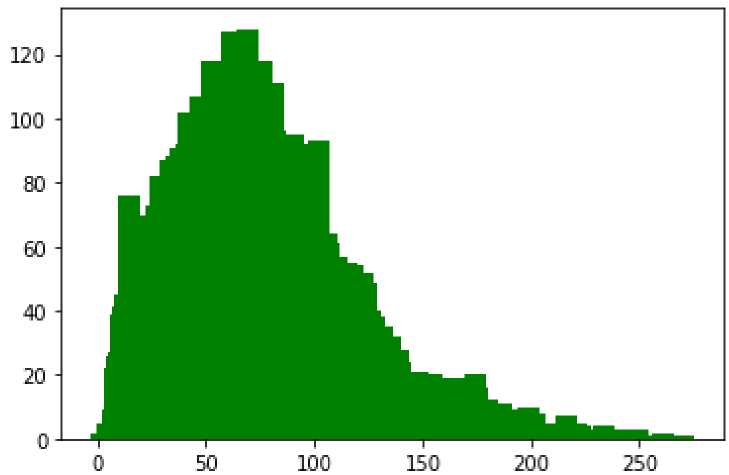


Fig. 7: Analysing length of sentences using Histogram

5. In the below Fig. 8., we found the most occurred words and displayed the count and word name. We have downloaded the stopwords and the iterated over the corpus to form the corpus into list of words, later used the counter method to get the counts of the words and used the most_common method to get the frequent occurred words and plotted them into a horizontal bar type graph.

```
[84] #most common words in the twitter text
     from collections import Counter
     import nltk
     import seaborn as sns

     nltk.download('stopwords')
     stop=set(stopwords.words('english'))
     Input_str=[]
     for line in Train_X:
         word_list= line.split()
         for word in word_list:
           Input_str.append(word)
     count=Counter(Input_str)
     common=count.most_common()
     x, y= [], []
     for word,count in common[:20]:
         if (word not in stop):
             x.append(word)
             y.append(count)

     sns.barplot(x=y,y=x)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
<matplotlib.axes._subplots.AxesSubplot at 0x7f7d751c0890>
```

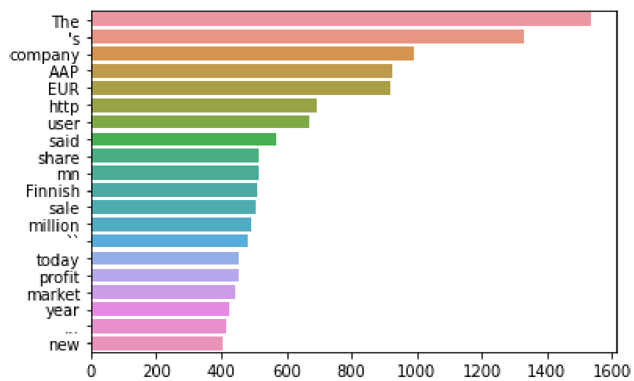


Fig. 8: Analysing Frequency of Unique words

Implementation

Installed the Tweepy package that is required to get the Tweets from twitter. (Fig.. 9)

```
In [1]: pip install tweepy

Requirement already satisfied: tweepy in c:\users\gopi krishna\anaconda3\lib\site-packages (4.10.1)
Requirement already satisfied: requests<3,>=2.27.0 in c:\users\gopi krishna\anaconda3\lib\site-packages (from tweepy) (2.28.
1)
Requirement already satisfied: oauthlib<4,>=3.2.0 in c:\users\gopi krishna\anaconda3\lib\site-packages (from tweepy) (3.2.0)
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in c:\users\gopi krishna\anaconda3\lib\site-packages (from tweep
y) (1.3.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\gopi krishna\anaconda3\lib\site-packages (from requests<3,>=2.27.0->
tweepy) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\gopi krishna\anaconda3\lib\site-packages (from requests<3,>
=2.27.0->tweepy) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\gopi krishna\anaconda3\lib\site-packages (from requests<3,>=2.
27.0->tweepy) (2021.10.8)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\gopi krishna\anaconda3\lib\site-packages (from requests<
3,>=2.27.0->tweepy) (2.0.4)
Note: you may need to restart the kernel to use updated packages.

In [2]: pip show tweepy

Name: tweepy
Version: 4.10.1
Summary: Twitter library for Python
Home-page: https://www.tweepy.org/
Author: Joshua Roesslein
Author-email: tweepy@googlegroups.com
License: MIT
Location: c:\users\gopi krishna\anaconda3\lib\site-packages
Requires: oauthlib, requests, requests-oauthlib
Required-by:
Note: you may need to restart the kernel to use updated packages.
```

Fig. 8

In the below Fig. 9, In the first cell Imported the required packages.

In the second cell, copied the keys that are required to access the Twitter API to get the tweets.

In the third cell, Passed the keys for the authentication and created a API object to access the twitter API.

```
In [3]: import os
        import tweepy as tw
        import pandas as pd

In [4]: consumer_key= 'Y3RJeJFRHnA9QnpZu8z9S0Skb'
        consumer_secret= 'mSHz1KdYkTNmYWuxMwX4fMzzGc9qv1OqgyQxWU26pheoGRcLuE'
        access_token= '1287803472805367808-EedGVpYxgeXKILni2gt9HvJacgsmxx'
        access_token_secret= 'K9jXNXr6i72odGz9XAiUnN5841eIUiDKcrj52RozRO743'

In [5]: auth = tw.OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_token_secret)
        api = tw.API(auth, wait_on_rate_limit=True)
```

Fig. 9

In the below 3 cells (Fig. 10), Fetched the tweets that are related to the TESLA by passing the 3 types of arguments that are related to the tesla stock. After getting the tweets, the tweets are written into a file called twitter.txt and then later after the tweets are written into the file, the file object is closed. Given 10000 as parameter to fetch the tweets of that count.

```
In [6]: keyword = 'Tesla'
        limit=10000
        tweets = tw.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
        file = open('twitter.txt', 'w', encoding="utf-8")
        for tweet in tweets:
            file.write(tweet.full_text+'\n')
        file.close()

In [7]: keyword = '#TSLA'
        limit=10000
        tweets = tw.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
        file = open('twitter.txt', 'a', encoding="utf-8")
        for tweet in tweets:
            file.write(tweet.full_text+'\n')
        file.close()

In [8]: keyword = 'yahoofinance and Tesla'
        limit=10000
        tweets = tw.Cursor(api.search_tweets, q=keyword, tweet_mode='extended').items(limit)
        file = open('twitter.txt', 'a', encoding="utf-8")
        for tweet in tweets:
            file.write(tweet.full_text+'\n')
        file.close()
```

Fig.10

In the below cell (Fig. 11) created a new text file with name processed.txt that contains the tweets from the twitter.txt after processing. All the special characters and the links are removed in this cell and then written into the processed.txt.

```
In [9]: file = open('processed.txt', 'w', encoding="utf-8")

In [10]: #preprocessing
         import re
         with open('twitter.txt','r', encoding="utf-8") as f:
             lines = f.readlines()
         f.close()
         for line in lines:
             content=' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", line).split())
             file.write(content+'.'+'\n')
         file.close()
```

Fig. 11

In the below cells (Fig. 12), the processed text is changed into the list format and then added to the test variable which needs to be classified.

```
In [11]: # opening the file in read mode
         my_file = open("processed.txt", "r")

         # reading the file
         data = my_file.read()

         # replacing end splitting the text
         # when newline ('\n') is seen.
         data_into_list = data.split("\n")
         my_file.close()

In [12]: Test_X=[]
         for x in data_into_list:
             Test_X.append(x)
```

Fig. 12

In the first cell (Fig. 13), the text is tokenized using the word tokenize method and then removed the stop words and then finally appened the remaining words into a sentence and then added back to the test_data.

In the second cell removed all the duplicate tweets if there are any to avoid the wrong results.

```
[13]:  import nltk
       from nltk.stem import WordNetLemmatizer
       from nltk.corpus import stopwords
       from nltk.tokenize import word_tokenize
       nltk.download('stopwords')
       stopword = set(stopwords.words('english'))
       Test_data=[]
       for x in Test_X:
           tokens = word_tokenize(str(x))
           final_tokens = [w for w in tokens if w not in stopword]
           wordLemm = WordNetLemmatizer()
           finalwords=[]
           for w in final_tokens:
               if len(w)>1:
                   word = wordLemm.lemmatize(w)
                   finalwords.append(word)
           Test_data.append(' '.join(finalwords))

       [nltk_data] Downloading package stopwords to C:\Users\Gopi
       [nltk_data]     krishna/nltk_data...
       [nltk_data]   Package stopwords is already up-to-date!

[14]:  #Removing single words
       Test_X=[]
       for x in Test_data:
           if len(x)>10:
               Test_X.append(x)
       Test_X = [*set(Test_X)]
```

Fig. 13

As seen in Fig. 14, to train the model we have collected 2 different sets from the Kaggle and then merged the 2 data sets.One is All-data.csv and other is stock_data.csv. Merged the 2 datasets to form a dataset of size 10K. Collected the text into the Train_X and then copied the sentiment to the Train_Y variables to train the model. For one dataset changed the sentiment values from 1 to positive and -1 to negative.

```
In [15]: twit = pd.read_csv("all-data.csv", encoding = "latin-1")

In [16]: Train_Y=twit["Sentiment"]

In [17]: Train_X=twit["Text"]

In [18]: twit = pd.read_csv("stock_data.csv", encoding = "latin-1")

In [19]: for ind in twit.index:
             if(twit['Sentiment'][ind]==-1):
                 twit['Sentiment'][ind]="negative"
             else:
                 twit['Sentiment'][ind]="positive"

C:\Users\GOPIKR~1\AppData\Local\Temp/ipykernel_44488/3332717168.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  twit['Sentiment'][ind]="positive"
C:\Users\Gopi krishna\anaconda3\lib\site-packages\pandas\core\indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  self._setitem_single_block(indexer, value, name)

In [20]: Train_X=Train_X.append(twit["Text"])
         Train_Y=Train_Y.append(twit["Sentiment"])

In [21]: Train_X.shape
Out[21]: (10637,)
```

Fig. 14

In the below cell (Fig. 15) cleaned the Train_X tweets by removing the stopwords, first downloaded the stopwords and then compared with the tokens in the text, if they are matched the tokens are removed and then appended to form a sentence and then added to the Train_X variable.

```
In [22]: import nltk
         from nltk.stem import WordNetLemmatizer
         from nltk.corpus import stopwords
         from nltk.tokenize import word_tokenize
         nltk.download('stopwords')
         stopword = set(stopwords.words('english'))
         Train_data=[]
         for x in Train_X:
             tokens = word_tokenize(str(x))
             final_tokens = [w for w in tokens if w not in stopword]
             wordLemm = WordNetLemmatizer()
             finalwords=[]
             for w in final_tokens:
                 if len(w)>1:
                     word = wordLemm.lemmatize(w)
                     finalwords.append(word)
             Train_data.append(' '.join(finalwords))
         Train_X= Train_data

[nltk_data] Downloading package stopwords to C:\Users\Gopi
[nltk_data]     krishna/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Fig. 15

As seen in Fig. 16 below, the model is built with the pipeline having vector inputs as parameters and multinomial classification. And then the model is trained by using the Train_X and Train_Y variables.

After training the model the Test data is passed which we collected from the twitter API and then the labels are predicted which can classified as Positive, Negative and Neutral.

```
In [23]: from sklearn.feature_extraction.text import TfidfVectorizer
         from sklearn.naive_bayes import MultinomialNB
         from sklearn.pipeline import make_pipeline

         model = make_pipeline(TfidfVectorizer(), MultinomialNB())

In [24]: model.fit(Train_X, Train_Y)
         labels = model.predict(Test_X)
```

Fig. 16

Preliminary Results:

The below screenshot (Fig. 17) gives us the overall stock direction after the classification, but we also need to perform the sentiment analysis to get the accurate results of the project.

The Labels are converted to the list and then count is collected for each class. Using those values, a bar graph is displayed which shows the count of each class of the tweets that are collected from the twitter API.

```
In [25]: Final_lables=labels.tolist()

In [26]: pcount=Final_lables.count("positive")
         ncount=Final_lables.count("negative")
         necount=Final_lables.count("neutral")
         pcount,ncount,necount

Out[26]: (1362, 63, 43)

In [27]: import matplotlib.pyplot as plt
         fig = plt.figure()
         ax = fig.add_axes([0,0,1,1])
         Sentiment = ['Positive', 'Negative','Neutral']
         Count = [pcount,ncount,necount]
         ax.bar(Sentiment,Count)
         plt.show()
```

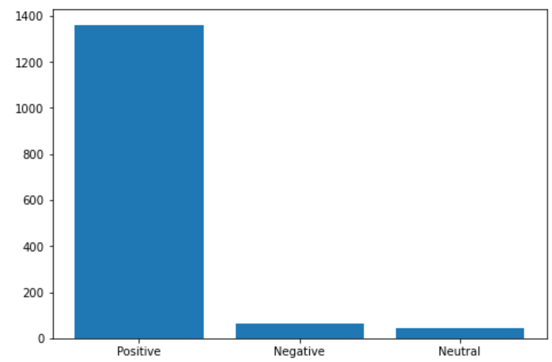


Fig. 17

Project Management and Responsibility

To manage this project, we have divided our responsibilities amongst four team members as mentioned earlier in this project document. We communicated in person as well as via online medium such as WhatsApp and Zoom to cooperate with one another. We broke down the project into smaller tasks and worked on each task as assigned primarily. Each task was peer reviewed by all four of us as mentioned below in Fig. 18

Task Description	**Primary Responsible (First Name)**	**% Of work done as Primary Responsible member**	**Peer Reviewed by (First Name)**
Brain Storming Ideas	Gopi, Sweety, Nikhil, Gayathri	25% each member	Gopi, Sweety, Nikhil, Gayathri
Data Collection	Sweety, Gopi, Nikhil, Gayathri	25% each member	Nikhil, Gayatri
Topic Research and Related work	Gopi, Sweety, Nikhil,	34% each member	Gopi, Sweety, Nikhil
Workflow and Project Outline	Sweety, Gopi, Nikhil	34% each member	Gopi, Nikhil, Sweety
Feature Selection	Gopi, Nikhil, Sweety	34% each member	Gayathri
Data Analysis	Gopi, Sweety, Nikhil, Gayathri	25% each member	Gopi, Sweety, Nikhil, Gayathri
Implementation	Gopi, Sweety, Nikhil, Gayathri	25% each member	Gopi, Sweety, Nikhil
Results	Gopi, Sweety, Nikhil	34% each member	Gopi, Sweety, Nikhil, Gayathri

Fig. 18

Future Work:

We have completed the Data analysis, preprocessing and the classification part for the NLP model. We need to complete the Sentimental analysis part and the stock prediction which we will complete in the final increment part of the project. Also, we will include some more analysis related to the test and train data of the tweets. We all the four members will coordinate the tasks for future work and report it in the next increment proportionately.

Demos:

Video 1: Gopi Krishna Sure

https://drive.google.com/file/d/1_joq2kKPsiqaeSe3H_pzCqPk1A_iGQQB/view?usp=share_link

Video 2: Sweety Makwana

https://drive.google.com/file/d/1lAxt2agXdiEwk3kLAbPOl5BfFCxDIeAS/view?usp=sharing

Video 3: Nikhil Rangineni

https://drive.google.com/file/d/1I9Pj7yAeoZ-a79H7a61_DYm5S57XbNnX/view?usp=share_link

Video 4: Gayathri Katukojwala

https://drive.google.com/file/d/1xrTqE6RCz5t-Eai1ulyz__7RYQpwdiNB/view

Google Colab:

1. Ipynb file https://drive.google.com/file/d/1rDRb_TPlAnqMVb-PRzOPww9XAWsZMU3j/view?usp=share_link

2. Pdf file https://drive.google.com/file/d/1pMDfsF0vagwyC5D59mpx22MnEBuhxlH8/view?usp=share_link

References:

- https://www.kaggle.com/
- https://finance.yahoo.com/
- https://data.world/crowdflower/apple-twitter-sentiment
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7959635/
- https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news
- https://monkeylearn.com/blog/sentiment-analysis-of-twitter/
- Go, Alec, Lei Huang, and Richa Bhayani. "Twitter sentiment analysis." *Entropy* 17 (2009): 252.
- Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!." *Proceedings of the international AAAI conference on web and social media*. Vol. 5. No. 1. 2011.
- https://www.nltk.org/
- https://developer.twitter.com/en/docs/twitter-api