Towards partial fulfillment for Undergraduate Degree level Programmed Bachelor of Technology in Computer Engineering

A Final Project Evaluation Report on:

Stock Market Analysis Using Twitter Sentiments

Prepared By:

Admission No	Student Name
U13CO063	AAKASH RANA
U13CO080	ADESH KALA
U13CO081	SHUBHAM GOTHWAL
U13CO084	A. ABISHEK

Class : B.TECH. IV (Computer Engineering) 8th Semester

Year : 2016-2017

Guided By : Dr. DIPTI P. RANA, Dr. RUPA G MEHTA

DEPARTMENT OF COMPUTER ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY,
SURAT – 395 007 (GUJARAT, INDIA)

Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

Sr.	**Admission No.**	**Student Name**
1	U13CO063	AAKASH RANA
2	U13CO080	ADESH KALA
3	U13CO081	SHUBHAM GOTHWAL
4	U13CO084	A. ABISHEK

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	**Student Name**	**Signature of the Student**
1	AAKASH RANA	
2	ADESH KALA	
3	SHUBHAM GOTHWAL	
4	A. ABISHEK	

Certificate

This is to certify that the project report entitled Stock Market Analysis Using Sentiment Analysis *is prepared and presented by*

Sr.	Admission No.	Student Name
1	U13CO063	AAKASH RANA
2	U13CO080	ADESH KALA
3	U13CO081	SHUBHAM GOTHWAL
4	U13CO084	A. ABISHEK

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE:

GUIDE JURY HEAD OF DEPT.

CO-GUIDE

Abstract

In this project, the influence of social media activities on economic indicators is studied. The social media here is the popular microblogging site Twitter and the economic indicators are stock-market events, defined as changes in the price and traded volume of stocks. Specifically, messages related to a number of companies are collected, and is searched for interrelationships between stock-market events for those companies and features extracted from the microblogging messages. In this work, an investigation is done about the relationship between the market indicators for these companies and the volume of tweets mentioning their names or stock symbols. Additionally, other factors are also considered, such as the predicted sentiment of the tweets, the number of followers/friends of the users and the presence of links on the tweets. With all this information, a predictor is trained for each company to estimate the changes in the stock market price. After selecting important Twitter features, together with the stock market indicators at proper timestamps, predictive models are applied, considering features and techniques like relevant tweet filtering, sentiment/emoticon analysis thereby predicting stock market indicators.

Index

List of Figures

List of Tables

Chapter 1: Introduction

Twitter, Inc., an online social networking and a micro blogging service where individuals post a 140-character thought, or “tweet”, is one of the most influential social media platforms out there. It is interesting to note that Twitter now boasts of around 400 million tweets a day by an estimated 313 million monthly active users (June 2016) [1]. These huge number of users are making social media an increasingly significant part of their daily activity and decision making processes, with consumers utilizing recommendations from within their network to help define their purchasing habits. Companies have also followed suit, by analyzing the trends and behavior of consumers to help improve their own product or service accordingly.

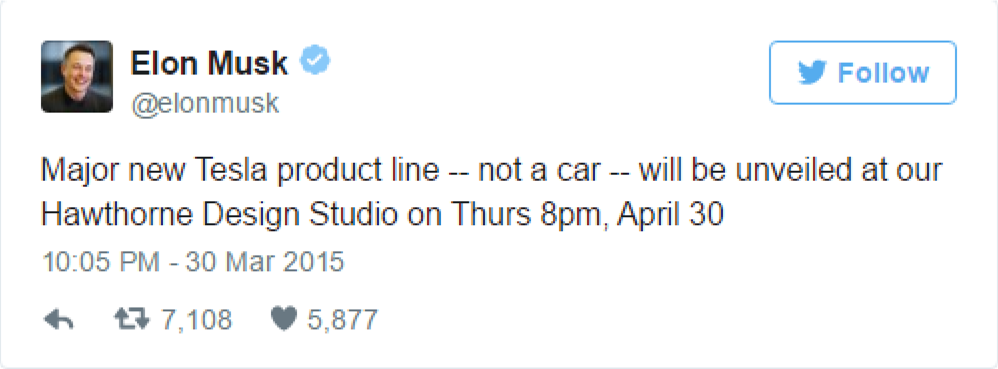
Stock market analysis is the foremost thing which is mandatory prior to any financial investment. To be defined in layman terms, stock market analysis refers to the entire procedure of monitoring and analyzing stocks and public sentiments and thereby calculating the future trends. Twitter is a vast resource of real time emotion, and the instant nature of emotive posts makes data mining easier and scrutiny of this data could help to reveal stock market movements [2]. So, if each tweet is a condensed summary of a person’s mood or opinion about a certain subject, then the aggregate of tweets about the subject should express the collective mood. Words in isolation may have a positive or negative sentiment but once you put them together they can often mean something else. By extension, public mood on social media should be correlated with or be predictive of economic indicators [3].

This project aims to map the correlation between Twitter trends related to some companies and its effect on stock market fluctuations for the same companies. Twitter has now become bigger than ever and with the advent of Digital India, more and more people are coming online, and public opinion is largely formed and spread using online social media.

1.1 Motivation

During the recent phenomena like the General Elections 2014, overnight success of Pokemon Go and product launches by companies like Apple and Samsung, it is observed that social networks like Twitter have massive influence in forming public opinion. The success / failure of a company's product or a political party of the world's largest democracy is dependent upon what the public sentiment regarding them is on Twitter.

Anecdotally, it is possible to identify some situations where messages posted on Twitter have in fact "moved markets." For example, on March 30 2015, a tweet from Elon Musk resulted in an increase of Tesla's capitalization by approximately US$1 billion in just a few minutes. [9]



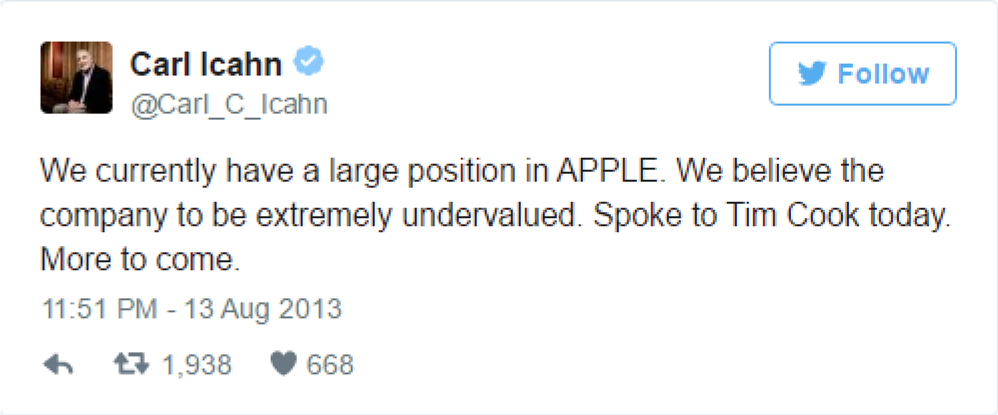
Elon Musk
@elonmusk

Major new Tesla product line -- not a car -- will be unveiled at our Hawthorne Design Studio on Thurs 8pm, April 30

10:05 PM - 30 Mar 2015

Fig.1- Tweet by Elon Musk

A somewhat similar story happened after a 2013 tweet from Carl Icahn led to increased capitalization of Apple of more than $10 billion. [9]



Carl Icahn
@Carl_C_Icahn

We currently have a large position in APPLE. We believe the company to be extremely undervalued. Spoke to Tim Cook today. More to come.

11:51 PM - 13 Aug 2013

Fig.2- Tweet by Carl Icahn

Stock market predictions have always piqued interest of many researchers as well as commercial companies because of the challenging prospects that it presents with. With lots of new research being conducted in Natural Language Processing, better results can be achieved in sentiment analysis of text and even classify a text according to its mood such as sad, happy, and angry. Also, Twitter has now become mature enough to provide us with very powerful APIs which are popular for its ease of use. Twitter Stream APIs can be used for capturing live tweets which can be filtered according to particular topics, making an excellent candidate for our use. Thus, the fairly complicated nature of the task at hand, and the prospect of working on NLP / Sentiment Analysis and applying Machine Learning Algorithms on real-world data and studying about workings of stock markets are some of the things that motivated us to work on this particular project.

1.2 Applications

While utilizing the tools of social media to predict market trends remains an unpredictable entity, individual retail investors can rely on these resources to gain knowledge and actively share information. The ambitions of active online retail traders is diametrically opposed to those of professional practitioners, in so much that they thrive by interacting as part of a large network to share and gather ideas. As there is no more significant or far reaching network than that accessible through social media, it is a natural home for traders looking to build or become part of a community.

1.3 Project Framework

The following Fig.3 shows the framework of overall project that depicts after data collection aggregation of category based tweet is performed followed by sentiment analysis module.

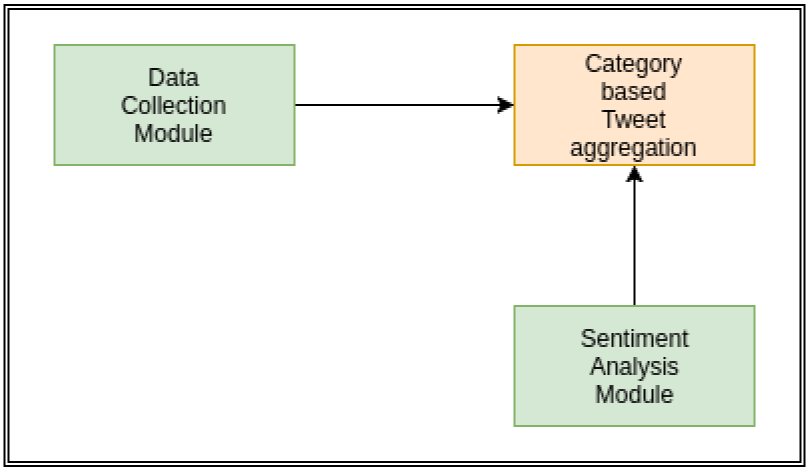


Fig. 3-Overall Framework

The following Fig.4 shows the data collection module which shows data is collected for tweets, related to tweet activity and stock prices which are explained detail in coming chapter.

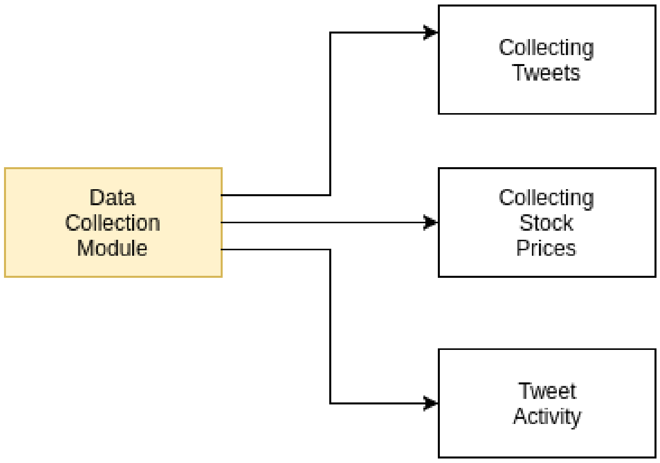


Fig. 4 – Data Collection Module

The following Fig.5 discusses after the data collection, retrieval of tweet features were performed and on that sentiment is calculated as shown in Fig. 6. Then using the range selection and block categorization module the effective stock price is calculated, which are explained detail in coming chapter.

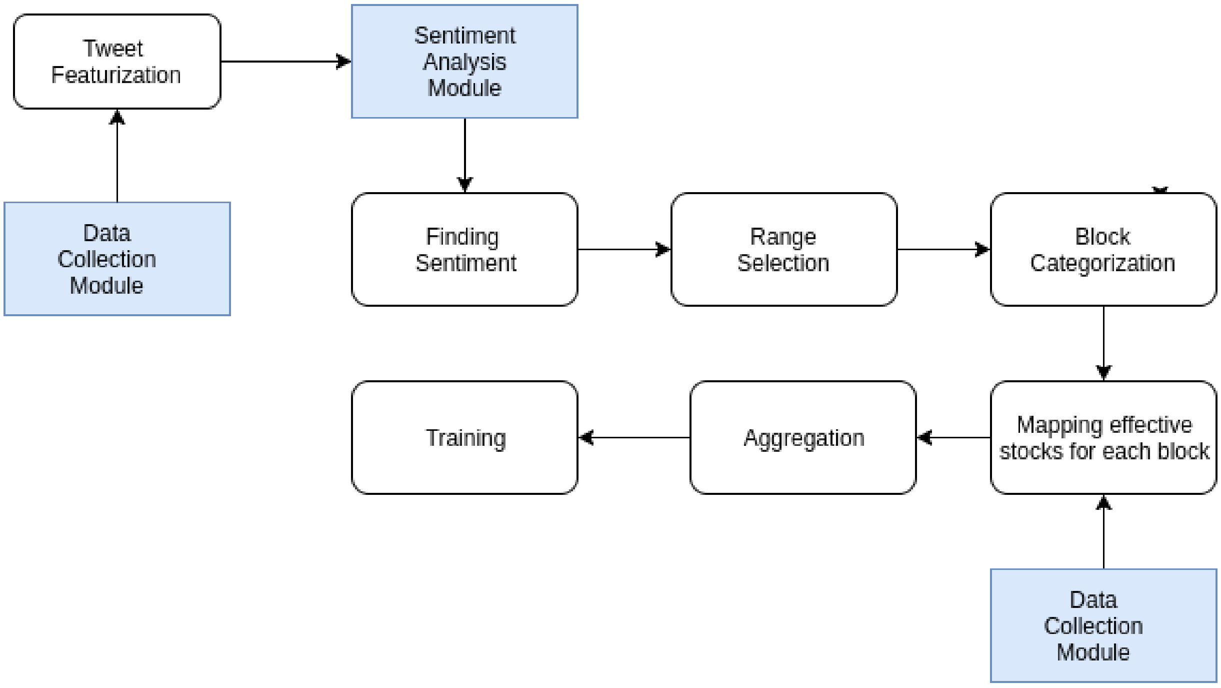


Fig. 5 – Category Based Tweet Aggregation Module

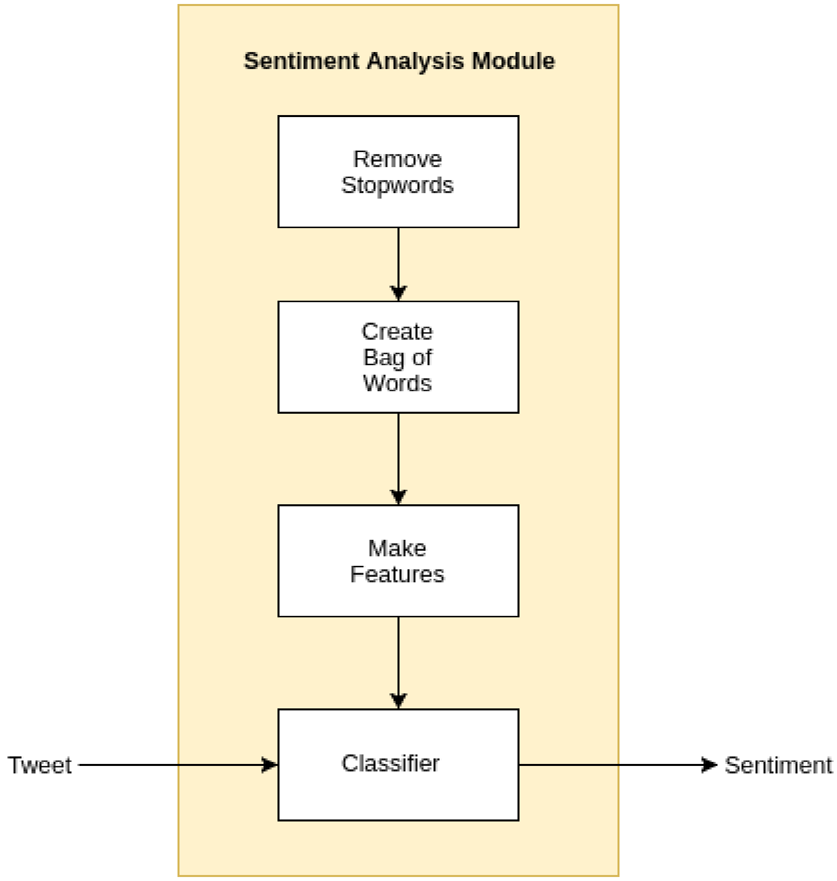


Fig. 6 – Sentiment Analysis Module

1.4 Organization of Project Report

The project is divided into 6 chapters. The current chapter of Introduction is followed by Chapter 2 which discusses about the Literature Survey that tells about the research papers that have been studied related to the project. The existing work shows a promising potential for the scope of the project with Twitter now being used by more number of people than ever before. The Third Chapter discusses about the method used for tracking tweet activity along with the challenges faced and its solutions. The collection of stock price data has also been discussed in the third chapter. The Fourth Chapter consists of the methods used for Sentiment Analysis using Python's NLTK Library. The tweets related to the company that is being tracked are then classified into positive or negative based on the trained sentiment classifier. The Fifth Chapter discusses about the actual feature-sets using two unique approaches namely, Block based tweet Aggregation and Category based Tweet Aggregation. The final results have been discussed in Chapter 6.

Chapter 2: Literature Survey

As a part of literature survey, the following research papers were studied. These research papers have discussed about various methods for sentiment analysis and correlates them with stock market data using various algorithms.

2.1 Twitter mood predicts the stock market

Johan Bollen, Huina Mao and Xiao-Jun Zeng in their research paper [4] have used Opinion Finder and Google Profile Of Mood States (GPOMS) to analyze the sentiments of Twitter users and its effects on stock market of DJIA.

Firstly, they take into account all the tweets which contain user sentiments and filter out the unwanted tweets using regular expressions that match “http”, “www”, etc.

These tweets are assessed by assessment tools like Opinion Finder which measures positive vs negative mood from a given text, GPOMS which measures 6 different mood dimensions from a given text context. This results in 7 mood time series graphs, one from Opinion Finder and 6 from GPOMS. Time series of DJIA closing price is extracted from Yahoo Finance and the hypothesis between public mood and DJIA closing prices is evaluated using Granger causality analysis. The hypothesis between public mood and DJIA stocks is also evaluated using Self Organizing Fuzzy Neural Networks, a nonlinear model.

GPOMS consists of 6 mood dimensions namely Calm, Alert, Sure, Vital, Kind and Happy. GPOMS mood dimensions and lexicon are derived from psychometric instrument, namely the Profile of Mood States (POMS). The score of each dimension is obtained by mapping each word with a lexicon. To enable the comparison between OF and GPOMS time series, each dimension of GPOMS is normalized by their Z-score.

$$Z_{X_t} = \frac{X_t - X_{mean}(\,X_{t\pm k}\,)}{\sigma(X_{t\pm k})}$$

Correlation between OF and GPOMS mood analysis can be found by using linear regression with the Z-score values of GPOMS and the values generated by OF. To correlate the time series of OF and GPOMS with the closing values of DJIA, the econometric technique of Granger causality analysis is applied. Granger causality analysis rests on the assumption that if a variable X causes Y then changes in X will systematically occur before changes in Y. The lagged values of X will exhibit a statistically significant correlation with Y.

Granger causality analysis suggests a predictive relation between certain mood dimensions and DJIA. However, Granger causality analysis is based on linear regression whereas the relation between public mood and stock market values is certainly non-linear. Considering these nonlinear effects and assess the public sentiments in DJIA values, comparisons are made with Self-Organizing Fuzzy Neural Network (SOFNN) model that predicts DJIA values on the basis of two sets of inputs: the past 3 days of DJIA values, and the same combined with various permutations.

Combination of two or more mood values are taken into account for predictive analysis. For example, Happy may not be independently linearly related with DJIA, but it may nevertheless improve the SOFNN prediction accuracy when combined with Calm. All the permutations of mood values are taken as an input to SOFFN model, first of which being I_0 represents a naive, baseline model that has been trained to predict DJIA values at time t from the historical values at time t-1, t-2, t-3.

2.2 Analyzing Stock Market Movements Using Twitter Sentiment Analysis

Tushar Rao and Saket Srivastava in [5] analyzed sentiments of 4 million tweets and found high correlation between social media activity and market indices. They classified their tweets as negative and positive using TwitterSentiment and calculated Bullishness, Volume and Agreement among positive and negative tweets for the given time period. The above parameters were then used to identify their correlation with stock/index parameters like Trading Volume and Returns.

Taking time period as monthly average, their approach showed strong correlation values. Further, Bivariate Granger Causality Analysis rejected with high confidence the null hypothesis that Twitter features do not affect returns in the financial market. The Expert Model Mining System(EMMS), which incorporates Exponential Smoothing, Auto Regressive Integrated Moving Average and seasonal ARIMA models, is applied twice, once with the tweets features and second time without it, so as to analyze improvement in prediction. They observed that there was a significant reduction in Mean Absolute Percentage Error and Maximum Absolute Percentage error between prediction and actual data when tweets features were used.

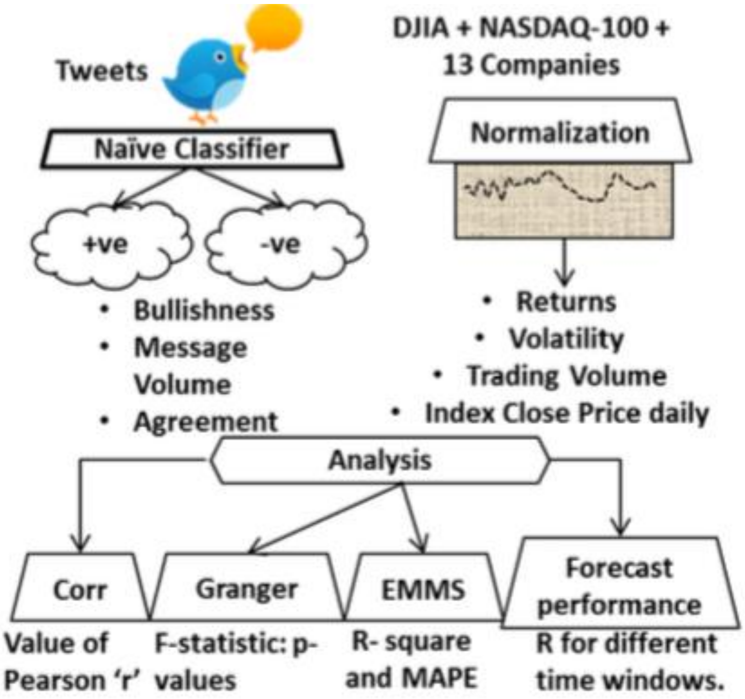


Fig. 7 - Workflow [5]

2.3 Prediction of changes in the stock market using twitter and sentiment analysis

In this paper [6], Iluian Vlad Serban, David Sierra Gonz'lez and Xuyang Wu have tried to mine the derived vast source of data from twitter for different purposes.

In this work, they have performed exhaustive feature selection procedure. The companies taken under consideration includes IBM, Intel and General Electric (GE). A predictor is trained for each company to estimate the change in the stock market price. The number of tweets are weighted by the number of friends. They analyzed twitter for positive and negative mood of the tweets and compared with stock market indices such as Dow Jones, S&P 500 and NASDAQ. The result showed that the number of positive tweets is much higher than the negative tweets. However, the mood indicators (both positive and negative) proved to be always negatively correlated with DJIA, NASDAQ and S&P 500.

Mao et al. [7] investigated the correlations between the number of tweets and stock indicators. Using a simple linear regression model with the tweet counts for a short period of 17 days, reported an accuracy of 68% in predicting the direction of change in the daily closing price at the stock market level.

Project Workflow:

- Initially, the dataset is parsed and processed.
- The sentiment of each English tweet related to the selected companies is analyzed.
- Extract feature from tweets.
- Finally, we use the models to predict the price and volume changes over a fixed test period of time.

2.3.1 Feature Extraction:

A step by step incremental approach is used for extracting features and building models. The raw twitter data contains a lot of information for each tweet, most of which is not relevant for this work. The study found that tweets tend to be more credible when they cite external sources, i.e. when they provide an URL with the information they are propagating, and when they are retweeted many times. Tweets also tend to be more credible if they are sent by users with many friends and followers. The tweets are weighted differently according to the number of friends and followers, and according to if they contain URL's or not.

Features extracted are:

- Tweet creation time.
- Content of the tweet.
- Number of friends.
- Number of followers.
- Contains URL.

The tweets containing company's stock names ($IBM, $INTC or $GE) are analyzed. After that, tweets are searched for any URL. Finally, tweet texts are converted to lower cases for performing sentiment analysis.

2.3.2 Sentiment Analysis:

For sentiment analysis, two different off-the-shelf platforms are used: *Stanford's Deeply Moving* and *LingPipe*. Stanford's Deeply Moving is a deep based on a Recursive Neural Network that builds on top of grammatical structures. It builds up a representation of whole sentences based on the sentence structure and computes the sentiment based on how words compose the meaning of longer phrases. It was trained on the dataset Stanford Sentiment Treebank.

LingPipe [6] is a toolkit for processing text using computational linguistics. LingPipe is used to do tasks like: find the names of people, organizations or locations in news, automatically classify Twitter search results into categories and suggest correct spellings of queries. *DynamicLMClassifier* is used for sentiment classification.

2.3.3 Sentiment Features:

The weight of friends and followers used are either linear or logarithmic. Linear weighting would imply, for example, that twice as many friends will make a tweet twice as credible. Logarithmic weighting would imply, for example, that more friends correspond to more credible tweets, but that each additional friend only adds a decreasing marginal credibility. This would be the case if users having more than a certain number of friends, say a thousand friends, were all equally credible. To keep the maximum amount of information we will consider the linear and logarithmic weightings, as well as no credibility weighting which would correspond to weighting each tweet with a constant. This yield 5 × 2 × 2 × 3 = 60 features in total. [6]

Features can be partitioned according to:

- Weighting factor.
- URL
- Tweet type
- Sentiment

2.3.4 Evaluation:

Evaluating predictive performance on financial time-series depends to a large extend on the purpose of the prediction. The evaluation metrics used were:

- Mean squared-error
- Mean absolute error
- Positive vs. negative accuracy

The above research papers were studied which provided a useful insight into the complexity of the project and many ideas were incorporated into our project workflow. The major thing that was noticed is that they considered only historical data instead of live Twitter Stream.

Chapter 3: Data Collection and Tracking Tweet Activity

We have two types of data to collect:

- Twitter Data
- Stock market data for specific companies

3.1 Twitter Stream

All the previous research that was done used past Twitter Data that was collected before the actual performing of the analysis. The new approach that is done here is to collect live stream of Twitter data. The stock market effects in many cases happen at a millisecond scale and real-time tweets would help develop a model that could give better accuracy.

A step by step incremental approach must be used for extracting features and building models. The raw Twitter data contains a lot of information for each tweet, most of which is not relevant for this work. It needs to be known if tweets tend to be more credible when they cite external sources, i.e. when they provide an URL with the information they are propagating, or when they are retweeted many times. Tweets may tend to be more credible if they are sent by users with many friends and followers. The tweets should be weighted differently according to the number of followers and favorites count.

The main aim here is to increase the quality of the data and consider only those tweets which matter for this project i.e. it should be related to the company which is being studied. Twitter provides Streaming API which provides a persistent flow of tweet data. The streaming process gets the input tweets and performs parsing, filtering, then is stored in persistent data storage. We are using a MySQL storage for our use. The HTTP handling process queries the data store for results in response to user requests. While this model is complex, the benefits from having a real-time stream of Tweet data make the integration worthwhile for our stock market prediction.

We are collecting the following data:

- ❖ The tweet content
- ❖ Number of followers on the account

- Retweet Count
- Favorite Count
- Language
- Time of tweet

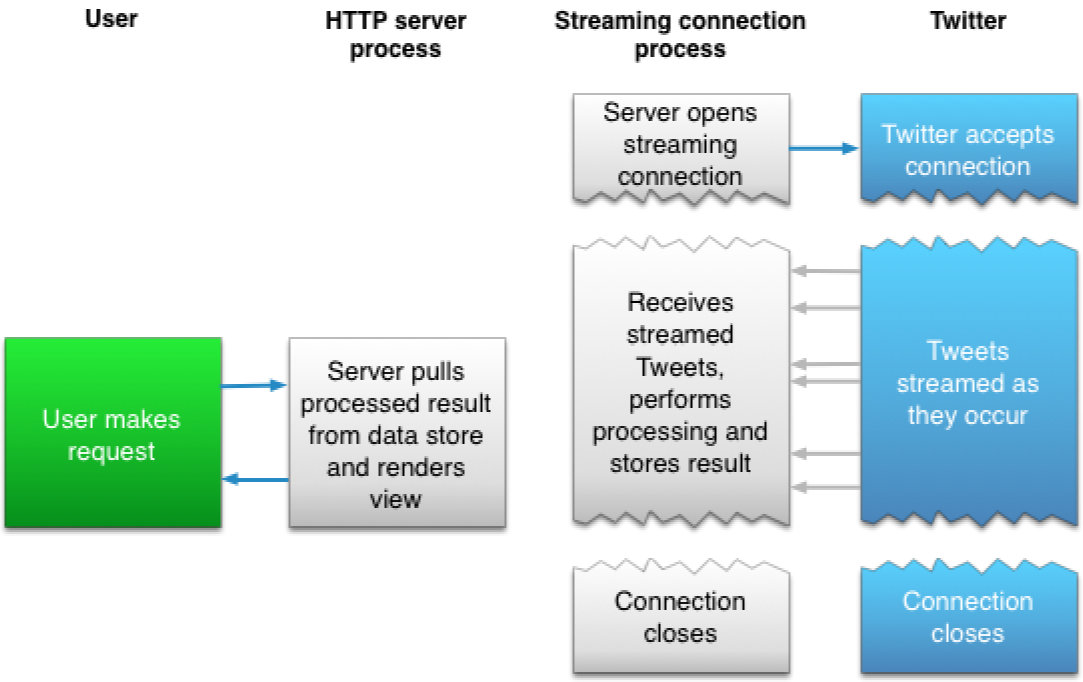

User
HTTP server process
Streaming connection process
Twitter
Server opens streaming connection
Twitter accepts connection
Receives streamed Tweets, performs processing and stores result
Tweets streamed as they occur
User makes request
Server pulls processed result from data store and renders view
Connection closes
Connection closes

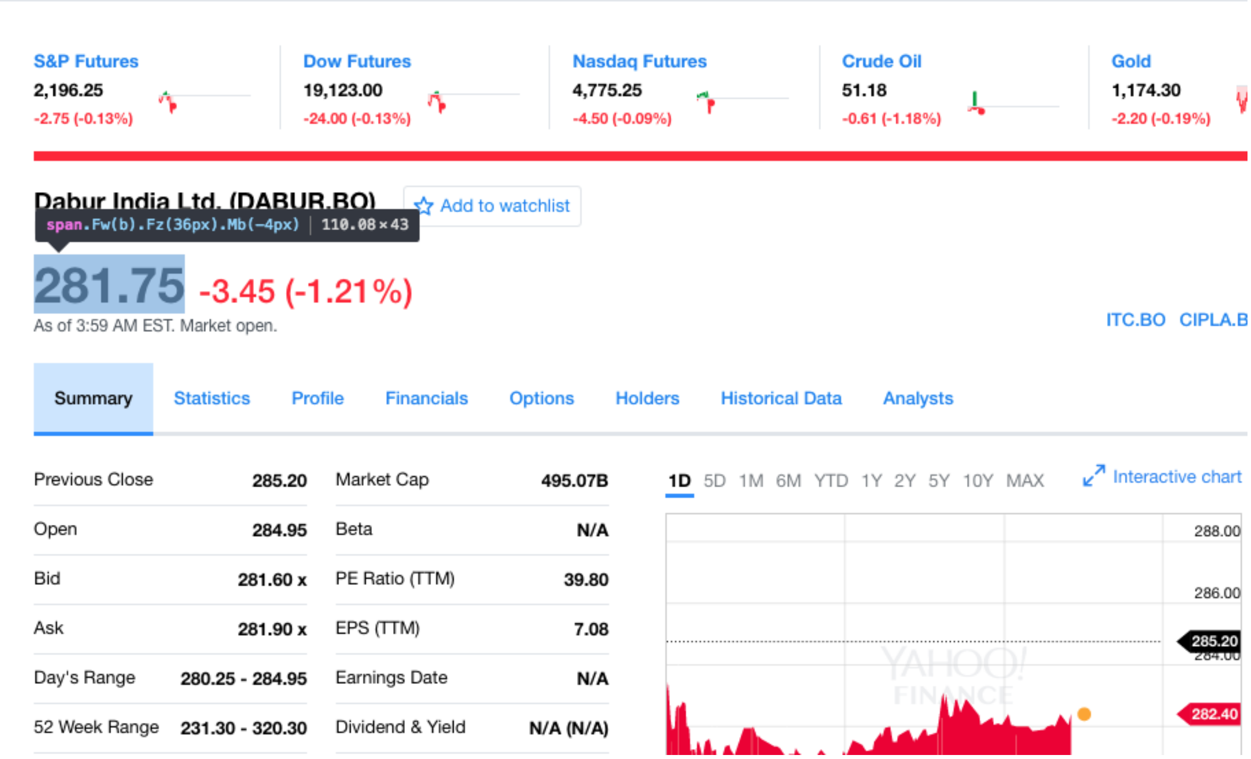
Fig. 8 – Twitter data Collection Workflow [10]

For this project, data for the company 'SBI' and 'Microsoft' are collected. The data collection script finds any mention of the word 'SBI' or '#SBI' or 'Microsoft' or '#Microsoft' in the tweet without considering the case in which it was written, fetches these tweets along with the above data and stores in a MySQL Database. The timestamp values can be used for correlation with timestamp values of stock market indices.

3.2 Stock Market Data

Main source for getting stock market data is Yahoo Finance [11]. For each company, Yahoo Finance has a dedicated page which provides data like percentage rise/fall of stock price, stock volume, earnings, market cap, etc. For the project, we are considering only the stock price and the rise/fall of the same. The financial stock market data must be collected for every 5 minutes.

This is essential as on days with high tweet activity, stocks can show big variations, with every minute bringing about big changes. As no API can provide such data, we will have to scrape the data from stock market websites and save it in database.



S&P Futures	Dow Futures	Nasdaq Futures	Crude Oil	Gold
2,196.25	19,123.00	4,775.25	51.18	1,174.30
-2.75 (-0.13%)	-24.00 (-0.13%)	-4.50 (-0.09%)	-0.61 (-1.18%)	-2.20 (-0.19%)

Dabur India Ltd. (DABUR.BO)

span.Fw(b).Fz(36px).Mb(-4px) | 110.08 × 43

281.75 -3.45 (-1.21%)

As of 3:59 AM EST. Market open.

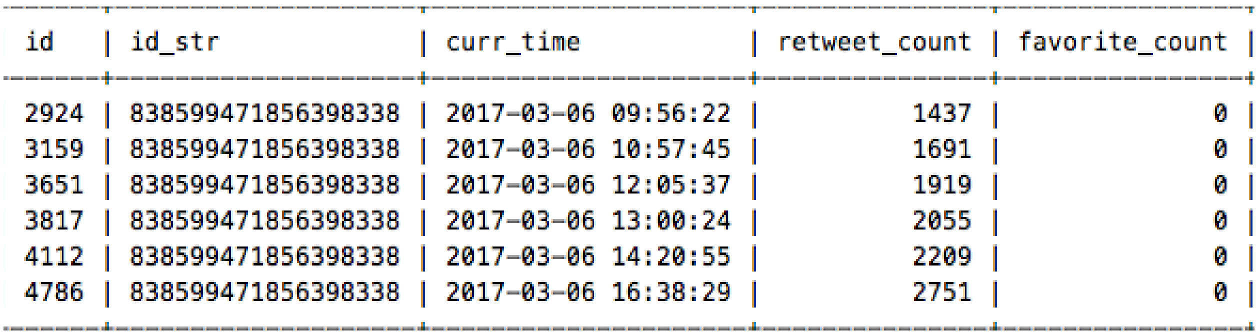
Previous Close	285.20	Market Cap	495.07B
Open	284.95	Beta	N/A
Bid	281.60 x	PE Ratio (TTM)	39.80
Ask	281.90 x	EPS (TTM)	7.08
Day's Range	280.25 - 284.95	Earnings Date	N/A
52 Week Range	231.30 - 320.30	Dividend & Yield	N/A (N/A)

Fig 9 - Getting the stock price using Yahoo Finance [11]

Here, a GET request is made to the link for the respective stock symbol and the stock market data is scraped off using the above span class parameter.

3.3 Tracking Tweet Activity

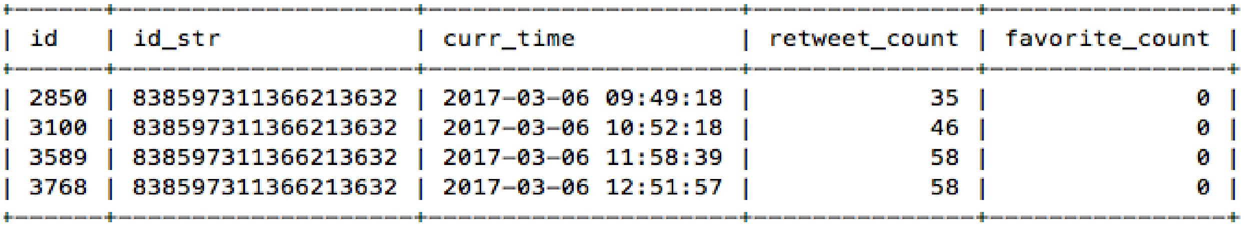
A model for prediction of the stock market price based on sentiments needs tweets data not just at an instant but over a longer period of time. This is because stock markets may show a delayed effect on its value after the tweets have been posted and not necessarily an instant effect. Also, some tweets by twitter handles with many followers may have a lot of retweet activity right from the time it is posted, while some customer related complaint might pick up over time. A system is needed that can adapt to these time changes and not be static.



```
-------+---------------------+-----------------------------+----------------+-----------------+
 id    | id_str              | curr_time                   | retweet_count | favorite_count |
-------+---------------------+-----------------------------+----------------+-----------------+
 2924 | 838599471856398338 | 2017-03-06 09:56:22 |           1437 |               0 |
 3159 | 838599471856398338 | 2017-03-06 10:57:45 |           1691 |               0 |
 3651 | 838599471856398338 | 2017-03-06 12:05:37 |           1919 |               0 |
 3817 | 838599471856398338 | 2017-03-06 13:00:24 |           2055 |               0 |
 4112 | 838599471856398338 | 2017-03-06 14:20:55 |           2209 |               0 |
 4786 | 838599471856398338 | 2017-03-06 16:38:29 |           2751 |               0 |
-------+---------------------+-----------------------------+----------------+-----------------+
```

Fig. 10 - Twitter trend of tweet 1

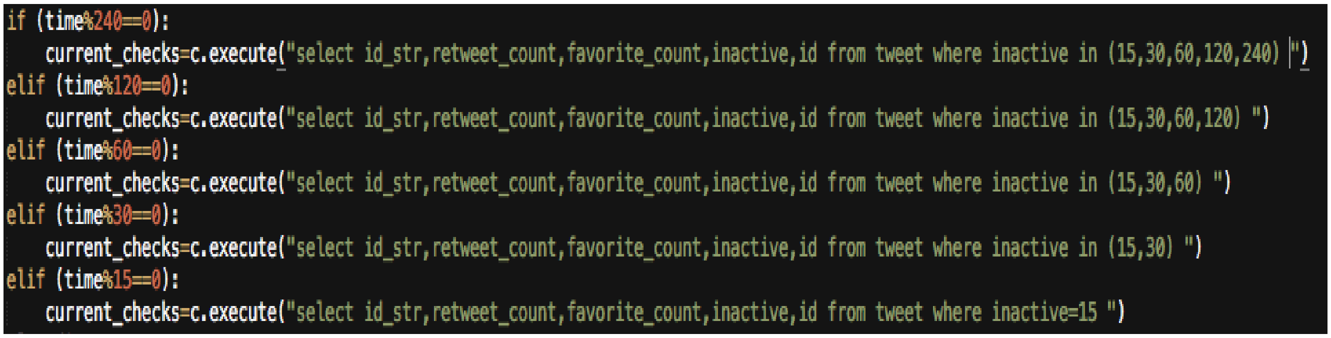
Taking a static time window of random value will lead to incorrect data as it is unable to track when the tweets make the real impact. For example, if 30 minutes' block size for tweets is taken, all tweets in 2:00 pm-2:30 pm time block will be mapped to the stock price at 2:30 pm. This could have ambiguous results. This ambiguity is because the tweet might have not had any impact in that time, and its effect might be much more prolonged than the 30 minutes' window. A tweet can be inactive when finally, its rate of change of activity becomes 0. At that point, it can be safely assumed that the tweet may not have any more effect on the stock market. Moreover, a tweet will keep having impact as and when it is being retweeted and thus cannot be justified with one independent value of stock price.



```
+------+--------------------+---------------------+---------------+----------------+
| id   | id_str             | curr_time           | retweet_count | favorite_count |
+------+--------------------+---------------------+---------------+----------------+
| 2850 | 838597311366213632 | 2017-03-06 09:49:18 |            35 |              0 |
| 3100 | 838597311366213632 | 2017-03-06 10:52:18 |            46 |              0 |
| 3589 | 838597311366213632 | 2017-03-06 11:58:39 |            58 |              0 |
| 3768 | 838597311366213632 | 2017-03-06 12:51:57 |            58 |              0 |
+------+--------------------+---------------------+---------------+----------------+
```

Fig. 11 - Twitter trend of tweet 2

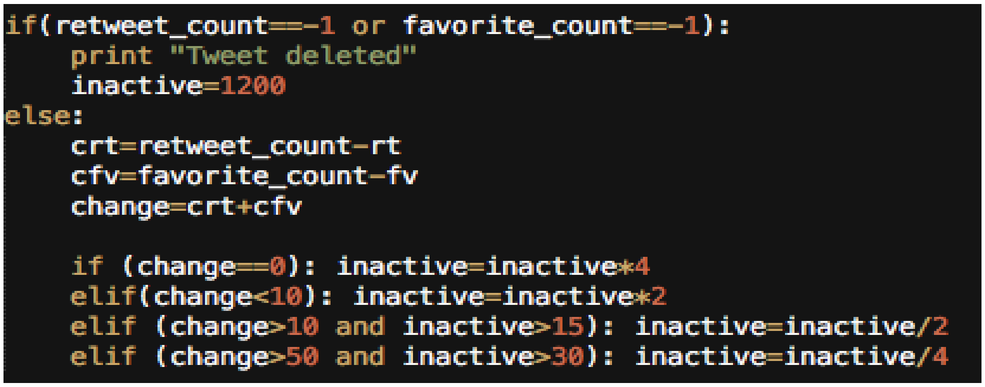
As it can be seen in the Fig. 10, the tweet is showing a lot of volatile activity and is continuously getting many retweets. This tweet thus should have an impact over a wider time period and stock. The above Fig. 11 started strong but as soon as it stopped showing any activity, no more information is retrieved for it.



```
if (time%240==0):
    current_checks=c.execute("select id_str,retweet_count,favorite_count,inactive,id from tweet where inactive in (15,30,60,120,240) ")
elif (time%120==0):
    current_checks=c.execute("select id_str,retweet_count,favorite_count,inactive,id from tweet where inactive in (15,30,60,120) ")
elif (time%60==0):
    current_checks=c.execute("select id_str,retweet_count,favorite_count,inactive,id from tweet where inactive in (15,30,60) ")
elif (time%30==0):
    current_checks=c.execute("select id_str,retweet_count,favorite_count,inactive,id from tweet where inactive in (15,30) ")
elif (time%15==0):
    current_checks=c.execute("select id_str,retweet_count,favorite_count,inactive,id from tweet where inactive=15 ")
```

Fig. 12 - The appropriate tweets are picked according to the above logic

The Fig. 12 denotes the algorithm used to find the relevant time tweets. After these tweets are stored in the database, a python script is run. Every 15 minutes, the activity (retweet and favorite count) is checked for the selected number of tweets. The initial activity of each tweet is set at 15 minutes. This inactivity is changed by the below given Fig. 13. The basis of it is that a tweet which is active must be monitored repeatedly while a tweet showing less activity must be searched through less actively as it would be unnecessary.



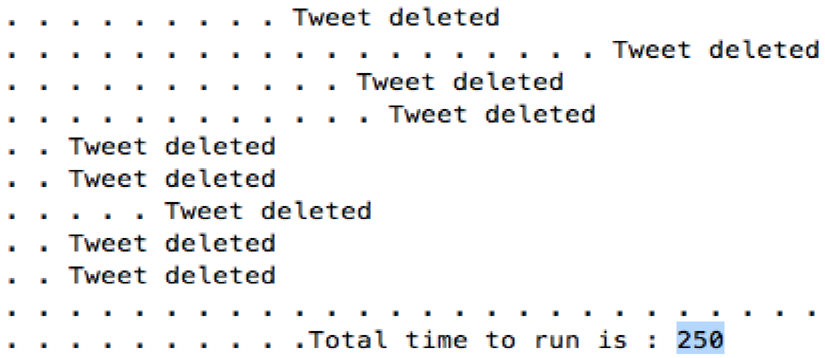
```
if(retweet_count==-1 or favorite_count==-1):
    print "Tweet deleted"
    inactive=1200
else:
    crt=retweet_count-rt
    cfv=favorite_count-fv
    change=crt+cfv

    if (change==0): inactive=inactive*4
    elif(change<10): inactive=inactive*2
    elif (change>10 and inactive>15): inactive=inactive/2
    elif (change>50 and inactive>30): inactive=inactive/4
```

Fig. 13 - Algorithm for determining value of inactivity

For example, if a certain tweet is not showing any activity at all, then it’s inactive time is made 4 times the current time. If some tweet suddenly starts showing some activity, it is to be ensured that its activity is retrieved more often and thus the time of inactivity should be significantly reduced. Using the above algorithm, a good time distribution of how tweet has made its activity and how the process is going on is obtained. This would be critical in the later stages of prediction of stock prices.

The one major shortcoming of this approach is that there is a single process making a request via Twitter API to get the tweet information and then updating a database. For 100 tweets, using a single process took approximately 300 seconds, i.e. 3 seconds per tweet. Thus, if information of 10,000 tweets are to be retrieved, it can take us hours. This need a better solution as requests are made every 15 minutes.

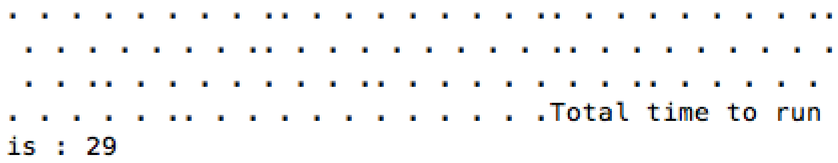


```
. . . . . . . . . Tweet deleted
. . . . . . . . . . . . . . . . . . . Tweet deleted
. . . . . . . . . . . Tweet deleted
. . . . . . . . . . . . Tweet deleted
. . Tweet deleted
. . Tweet deleted
. . . . . Tweet deleted
. . Tweet deleted
. . Tweet deleted
. . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . .Total time to run is : 250
```

Fig. 14 - Results without multiprocessing

This solution adopts the use a pool of ten processes. As retrieving tweets data is independent of each other, ten concurrent processes can retrieve this data and save this in the database. The approach speeds up the process by a magnitude of 10 and now the task can get done in 1/10th of the time.

This solution seems easy and trivial but its implementation raises some issues of its own. One major problem is dealing with concurrent access of database where some process can hold the database and other process must wait for it. Optimization of code and timing constraints are needed to prevent this kind of error. The other major problem that occurred is that the python network library uses keep alive to reuse the connection. So, if a connection gets a problem due to rate limitations, then it is still reused. The solution to this was to close that process and make it wait for the other processes to end. The closed process is no more allocated any more chunks of tweets and this way it does not cause any problems.



```
. . . . . . . . .. . . . . . . . .. . . . . . . . ..
 . . . . . . . .. . . . . . . . .. . . . . . . . .
 . . .. . . . . . . . .. . . . . . . . .. . . . . .
. . . . . .. . . . . . . . . . . .Total time to run
is : 29
```

Fig. 15 - After Optimizations with multiprocessing

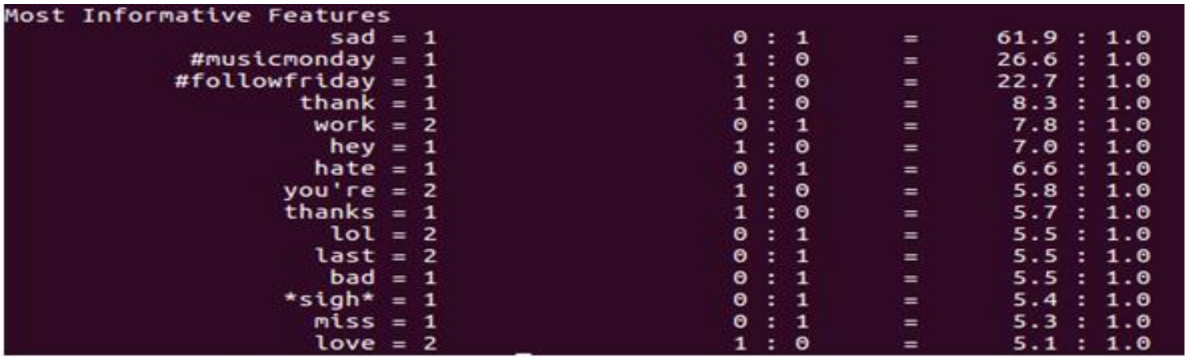
Overcoming the problems of the initial script, the modified script effectively tracks the activity of the tweet and this tweet activity plays a crucial role in the later training phases.

Chapter 4: Sentiment Analysis using NLTK

Sentiment analysis can use natural language processing, artificial intelligence, text analysis and computational linguistics to identify the attitude of a writer with respect to a topic. It’s an important cornerstone of behavioral finance, where theorists believe that markets are irrational and that asset prices are driven by human emotion (e.g., fear, greed, hope and overconfidence, among others). With the growth in global conversation on social media - Twitter in particular - where a vast amount of real-time market conversation occurs on a daily basis, academics and practitioners have been studying and measuring the global conversation to understand if it can meaningfully impact markets. Most concur that Twitter sentiment is correlated to asset price moves, but the debate has been about the predictive nature of Tweets on price. Well, the results are in and the early movers in this space are seeing success.

4.1 Preprocessing and training of the dataset

A dataset of around 1.5 million tweets which are hand-labelled as positive or negative has been used. This dataset contains informal language, SMS language and 140-character limit making it like the type of data we are working on. In a previous project iteration, a bag-of-words of size 50 was used to create the feature-set for sentiment analysis. After running the code, it was found that a small bag size isn’t good for getting high accuracy. After removing all the stop-words, all the extra spaces, all characters like ‘&’, ‘;’, etc. which do not carry any sentiment value, obtained data set was cleaned. The size of the bag in the new updated model is changed to 5000 words. The bag is formed by taking the most frequent 5000 words in the cleaned dataset. Then the data is trained using the Naive Bayes Classifier from the NLTK library and the output is below.



```
Most Informative Features
                     sad = 1                   0 : 1      =     61.9 : 1.0
             #musicmonday = 1                  1 : 0      =     26.6 : 1.0
            #followfriday = 1                  1 : 0      =     22.7 : 1.0
                   thank = 1                   1 : 0      =      8.3 : 1.0
                    work = 2                   0 : 1      =      7.8 : 1.0
                     hey = 1                   1 : 0      =      7.0 : 1.0
                    hate = 1                   0 : 1      =      6.6 : 1.0
                  you're = 2                   1 : 0      =      5.8 : 1.0
                  thanks = 1                   1 : 0      =      5.7 : 1.0
                     lol = 2                   0 : 1      =      5.5 : 1.0
                    last = 2                   0 : 1      =      5.5 : 1.0
                     bad = 1                   0 : 1      =      5.5 : 1.0
                  *sigh* = 1                   0 : 1      =      5.4 : 1.0
                    miss = 1                   0 : 1      =      5.3 : 1.0
                    love = 2                   1 : 0      =      5.1 : 1.0
```

Fig. 16 - Most Informative features

The above figure shows a snapshot of the output of the Naive Bayes Classifier. The model gave ~75% accuracy and this also shows a list of most informative features. For example, if the tweet contains the word 'sad', there is 62 times more chance that the tweet is negative than positive. Since training on 1.5 million tweets with feature size of 5000 takes a lot of time, the pickle tool of Python is used. The data is trained once and saved in a file. Whenever it is needed, it is loaded back using pickle. If there is any change in the model, the pickle file is updated by training the data again. The output of the tweet sentiment on an unlabeled tweet is used to make our second training data set which does the actual prediction of stock price movement. When making the training dataset for predicting stock price for a company, the following entities are considered. (This is an example data.)

Sr No	Tweet Sentiment	Followers Count	Retweet Count	Favorites Count	Stock Price (Up/Down)
1	Positive	125	35	12	Up
2	Negative	51	15	5	Down
3	Positive	10000	250	102	Up
4	Negative	10	1	0	Up
5	Positive	25	4	2	Up

Table 1 - Sample tweets

If thought intuitively, a single tweet alone shouldn't result in the stock price going up or down. It is usually many positive tweets' combined effect that resulted in the stock price going up or many negative tweets' combined effect that resulted in the stock price going down. Hence it doesn't make sense to map a single tweet sentiment value to the stock price, but instead there is a requirement for 'aggregation' that considers a combined effect of all the tweets and map it with a stock price value.

We have come up with various approaches for aggregation each with its pros and cons. For understanding these approaches, a few terms have been defined for easier understanding of the following report.

4.2 Sentiment Analysis: Decreasing Memory Complexity

The original memory complexity of the model was Big O (Number of tweets X 5000). This is the reason why all the 1.5 million tweets were not able to be trained. For each tweet only 15-20 words will have a positive frequency count. So, instead of considering all the 5000 bag of words, it is optimal to consider those which have a positive frequency count.

After performing this optimization, the new reduced memory complexity is Big O (Number of tweets X K), where K is the expected number of words in a tweet. Only these frequent words are to be considered as features for our model. Using this approach, an accuracy of around 75% is achieved.

4.3 Improvements to Sentiment Analysis Model

The words frequencies in the previous model were calculated using single word, i.e., the 1-gram technique. To improve the accuracy, the bigram technique is implemented in the current model along with the 1-gram. A bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words. A bigram is an n-gram for n=2.

Example: ‘bad’ is usually associated with a negative sentiment. If our 1-gram method finds ‘not bad’, it will map this ‘bad’ word to negative sentiment whereas it is used in a positive way in this case.

Bigrams approach will also be able to handle such cases, thus improving the accuracy.

4.4 Word Cloud

The word clouds are usually created using word frequencies. From a bunch of documents in each of the topics, the number of times each word occurs is counted and used for visualization. The area occupied by the word in the cloud is directly proportional to the frequency count of the word in the document. Here, a word cloud is prepared by using the most used words from the tweet data that was collected. It is clear that words like ‘good’, ‘get’, ‘want’, ‘love’, etc. are some of the most used words in the tweets and are a major representation of sentiments in them.

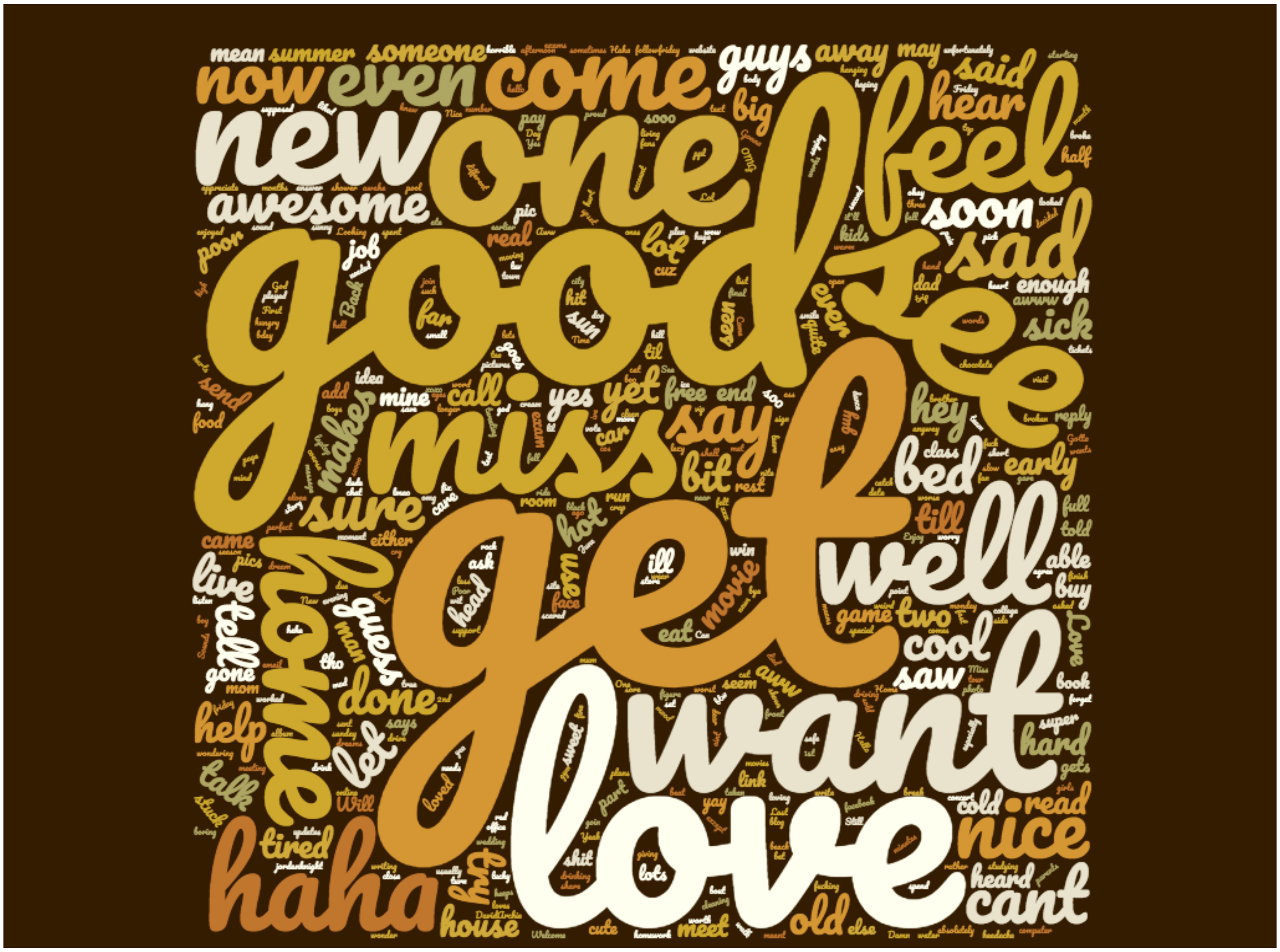


Fig. 17 - Word cloud

4.5 Training of Dataset Using Different Classifiers

i) Maximum Entropy Text Classifier:

The Max Entropy classifier is a discriminative classifier commonly used in Natural Language Processing, Max Entropy classifier performs very well for several Text Classification problems such as Sentiment analysis.

The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naive Bayes classifier, the Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the principle of maximum entropy and from all the models that fit our training data, selects the one which has the largest entropy. Due to the minimum assumptions that the Maximum Entropy classifier makes, it is regularly used when anything about the prior distributions is not known and when it is unsafe to make any such assumptions. Moreover, Maximum Entropy classifier is used when the conditional independence of the features can't be assumed. The Max Entropy requires more time to train comparing to Naive Bayes. Accuracy of around 74.8% is obtained.

ii) Decision Trees:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

iii) Multinomial NB:

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). Multinomial Naive Bayes is a specialized version of Naive Bayes that is designed more for text documents. Whereas simple naive Bayes would model a document as the presence and absence of words, multinomial Naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with.

Using bag of words technique for determining the features of the tweet and using Naïve Bayes classifier for training yields a respectable accuracy.

Chapter 5: Approaches to Enhance Training Features

Here the various approaches to train the dataset will be discussed and the most appropriate one will be chosen based on their pros and cons.

5.1 Definitions

- **Block:** A group/collection of tweets along with its attributes that are tweeted in a fixed interval of time.

  E.g. Tweets that were tweeted between 2:00 PM to 2:30 PM can be considered as a single block with block-size of 30 minutes.

- **Aggregation:** The sum of values of all attributes of the tweets that belong to a specific sentiment class.

  E.g. Let's consider all tweets of block 2:00-2:30 PM for a particular day. Now the sum of all values of the attribute 'followers count' for all the positive tweets in this time period. Similarly, sum of values for all attributes.

- **Range:** An attribute divided into specified number of groups on the basis of some upper or lower limits for the chosen attribute.

  E.g. Dividing follower count attribute into range-groups like [0,100], [100,250], etc.

- **Delay:** The time after which a tweet or a block of tweets will affect the stock price Value.

  E.g. Suppose a tweet on time 2 PM got popular throughout the day, and the stock price rose/fell at the starting of the next day (9 AM). This time difference is defined as delay.

- **Category:** Defines the type of a tweet based on the range of values, that a tweet's attribute falls under.

  E.g. If a tweet has a follower count in range [>250] and the retweet count under [0,50] then the tweet is said to be in the category ([>250], [0,50]).

Tweets can be categorized as (high followers, high retweets), (low followers, high retweets), etc. based on the ranges selected.

- **Effective Stock Price:** The stock price for a block is said to increase if the count of increase in stock price per 5 minutes is greater than the count of decrease in stock price.

  E.g. Consider a block size of 15 minutes. Let's say the stock price increased twice and decreased once, the effective stock is considered to raise.

These terms will become clearer when they are used in the following approaches.

5.2 Block based Tweet Aggregation

In this approach, all time-stamped tweets are put into blocks according to some decided time intervals. For example, if the block size is 30 mins, all tweets between 2:00 PM to 2:30 PM will be in the same block. Similarly, all tweets between 2:30 PM to 3:00 PM will be in the same block.

After this, the aggregation process is applied on all the tweets of the same block and are converted into a single row. The aggregation process includes summing up all values of all attributes that belong to positive class and summing up all values of all attributes that belong to negative class. The following figure shows the aggregation of the above table.

Followers Count (Positive)	Retweet Count (Positive)	Favorites Count (Positive)	Followers Count (Negative)	Retweet Count (Negative)	Favorites Count (Negative)	Stock Price
10150	289	116	71	16	5	Up

Table 2: Example of Tweet Featurization 1

With this, the effects of a block of tweets can be aggregated instead of working with a single tweet's effect which might not be a good indicative of the stock price value. We can map this block to the stock price at some time equal to the ending time of the block + the delay value.

Example: If the block contains tweets from 2:00 PM to 2:30 PM, and our delay value is 24 hours, the stock price value will be of 2:30 PM for the next day.

5.3 Category based Tweet Aggregation:

There were 2 major problems in the last approach. The first problem is that assume only the tweets of a company on a particular day are taken for training and the block size being 30 minutes, the number of rows in the training dataset is 48. It is impossible to train the model with 48 values. The next problem is that grouping different users together might not fetch accurate results. Users with large number of followers or tweets with high retweet counts will make a higher impact on the stock market compared to others. For normal users/tweets their cardinality will matter the most ie the number of positive/negative tweets by the normal user.

Each attribute is divided into a set of ranges based on the distribution of the number of tweets/users for that attribute. For example, 10 percentiles of the active twitter users have less than 3 followers and 99 percentile of twitter users have less than 3000 followers. The median of the distribution is around 60 followers [8].

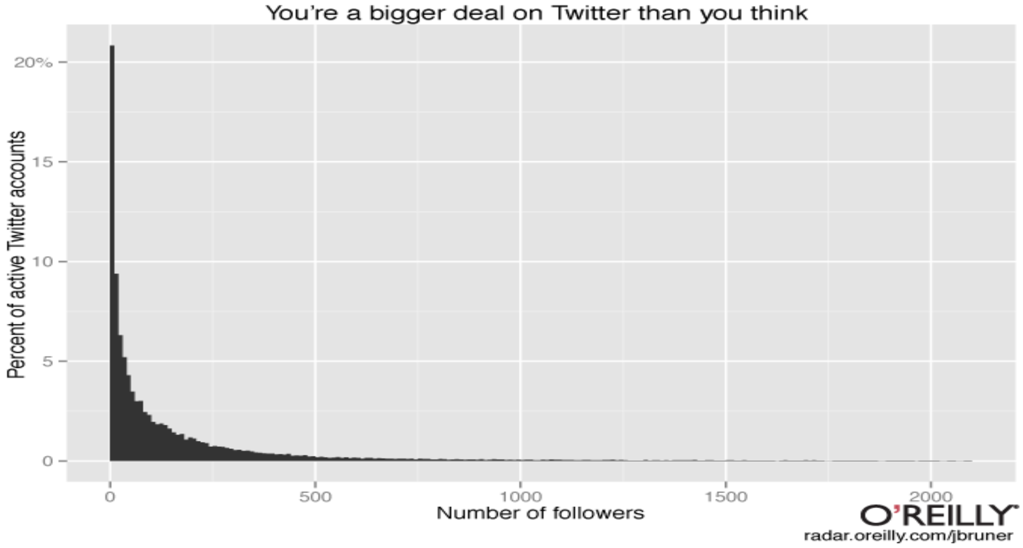


Fig 18 - Number of followers vs active twitter accounts [8]

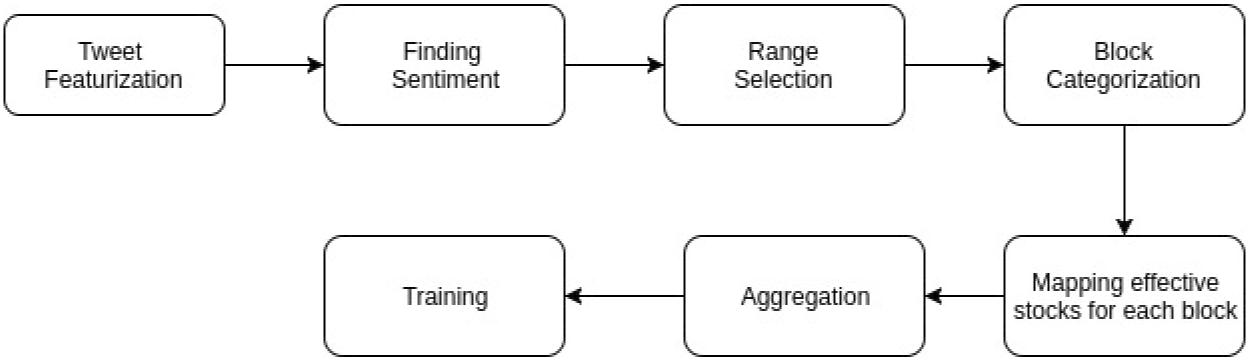


Fig 19 - Steps involved in Category based Tweet Aggregation

5.3.1 Categorization of Blocks

The three main attributes are number of followers of a user, retweet count and favorites count of the tweet. If number of followers is divided into x1 ranges, retweet count and favorites count of tweets are divided into x2 and x3 ranges respectively, then the total number of rows in a single block equals x1 * x2 * x3. For example:

Followers Range:

1. [0 – 3]

2. [3 – 200]

3. [200 - 3000]

4. [> 3000]

Retweet Range:

1. [0 – 50]

2. [50 – 500]

3. [500 – 2000]

4. [> 2000]

Favourites Range:

1. [0 – 10]

2. [10 – 300]

3. [300 – 4000]

4. [> 4000]

The total number of categories per block = 4 * 4 * 4 = 64. Every tweet will lie in one of these 64 categories. A tweet tweeted by a user with 250 followers, retweeted 650 times and with favorites count 4500 will fall in category 2 of follower range, category 3 and 4 of retweet range and favorites range respectively. Thus, every tweet will fall under some category of every attribute and each row will contain only the aggregation of the tweets that fall in that category. Aggregation means the sum of values of all tweets of every attribute that fall in some given category.

5.3.2 Mapping Blocks with Stock Price

- All tweets from 12 am - 9am is considered as a single block and is mapped with the effective stock price at 9am.

- Tweets from 9am - 4pm are divided into 1 hour blocks and are mapped with the effective stock price corresponding to the end of the block.

- Tweets after 4pm will be mapped with the opening stock price of the next day.

5.3.3 Aggregation Technique

Let n be the maximum of number of ranges considering all the features of the tweet. We concatenate the range numbers of all the features. This number is the base n representation of the row in which the features must be aggregated.

Sr No	Followers Count	Retweet Count	Favorite Count
1	<=10	<=10	<=10
2	>10	>10	>10

Table 3 - Example for categorization of tweets

Here the value of n is 2. All the eight possibilities of categories can be represented in binary form.

Example:

Sr No	Followers Count	Retweet Count	Favorite Count
1	3	100	100
2	100	9	1

Table 4 - Examples of tweets

In the first example,

Followers count <=10 (Range 0)
Retweet Count >10 (Range 1)
Favorite Count > 10 (Range 1)

Concatenating all the range numbers gives the binary number 011. Thus row 3 is aggregated.

In the second example,

Followers count >10 (Range 1)
Retweet Count <= 10 (Range 0)
Favorite Count <= 10 (Range 0)

Concatenating all the range numbers gives the binary number 100. Thus row 4 is aggregated.

5.4 Training Using Category based Tweet Aggregation

The data is aggregated according to the above method, and a cleaned dataset is obtained which is ready for training using various machine learning models. The features consist of fields like positive followers' aggregation, negative followers' aggregation, positive and negative retweet count and favorites count aggregation. This row is mapped to a Boolean value 0 if the stock market price went down and 1 if stock market price went up. This dataset is trained using Decision Tree Classifier and Random Forests Classifier.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [12]. Accuracy of around 67% is obtained on the said dataset.

Classifier	Accuracy
Decision Tree Classifier	63.12%
Random Forests Classifier	67.39%

Table 5: Accuracy with different classifiers

Category based Tweet Aggregation is more flexible approach as compared to Block based Tweet Aggregation. It gives better results with Random Forests Classifier.

Conclusion and Future Work

Twitter Analysis for stock markets has found many market participants in the last few years and they are actively leveraging the predictive nature of tweets for trading and investment. This project aimed at prediction of the upward or downward stock price movement for a particular company and it has been achieved with respectable accuracy. Twitter data alone cannot be taken as a deterministic way of predicting stocks, but coupled with other factors, it can be a great tool for getting an early lead in a market where every second matters. The above approaches achieve an accuracy rate of 67% in predicting the effective stock price of the company.

The open-ended nature of this project makes improvement possible at all stages of the project: data collection, sentiment analysis and prediction. There is a need to extract relevant tweets and only consider them for prediction. This can be done using Named Entity Recognition. The relationship effects of different companies on each other must be considered if they are in the same domain. e.g. Jio announcing a new offer could affect stock prices of Airtel. Deep learning methods can be used to improve the sentiment analysis classification. Also, prediction of the actual stock price of a company can be done instead of just predicting the rise/fall.

References

[1] *Number of monthly active Twitter users worldwide from 1st quarter 2010 to 2nd quarter 2016*: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] *Twitter and Stock Trading: A Real Strategy? By Ryan C. Fuhrmann, CFA | June 19, 2015* http://www.investopedia.com/articles/active-trading/061915/twitter-and-stock-trading-real-strategy.asp

[3] *Does Social Media Affect Consumer Decision Making? | Patarawadee Sema, Johnson & Wales University | 20-7-2013* http://scholarsarchive.jwu.edu/cgi/viewcontent.cgi?article=1023&context=mba_student

[4] Twitter mood predicts the stock market | Johan Bollen, Huina Mao and Xiao-Jun Zeng: http://arxiv.org/pdf/1010.3003v1.pdf

[5] Analyzing Stock Market Movements Using Twitter Sentiment Analysis | Tushar Rao, Saket Srivastava: http://eprints.lincoln.ac.uk/11274/1/ASONAM%202012.pdf

[6] Prediction of changes in the stock market using twitter and sentiment analysis | Iulian Vlad Serban, David Sierra Gonzalez, and Xuyang Wu: http://blueanalysis.com/iulianserban/Files/twitter_report.pdf

[7] Yuexin Mao, Wei Wei, Bing Wang, and Benyuan Liu. Correlating S&P 500 stocks with twitter data. In Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, HotSocial ’12, pages 69–72, New York, NY, USA, 2012. ACM.

[8] Tweets loud and quiet https://www.oreilly.com/ideas/tweets-loud-and-quiet

[9] Can Twitter Help you Beat the Stock Market? https://newrepublic.com/article/124860/can-twitter-help-beat-stock-market

[10] Twitter Developer Documentation: https://dev.twitter.com/streaming/overview

[11] Yahoo Finance http://finance.yahoo.com

[12] Random Forests Classifier Wikipedia page http://en.wikipedia.com/wiki/Random_forest

Acknowledgement

The completion of this project would not have been possible without the continued support and guidance of our guide Dr. Dipti P. Rana and co-guide Dr. Rupa G. Mehta. Their fresh ideas and innovative solutions helped us successfully complete the project and we would like to thank them for the same.