SENTIMENT ANALYSIS OF DATA FOR PREDICTING STOCK MARKET MOVEMENT

B E Project Report

Submitted in partial fulfillment of the requirements of the Degree of

Bachelor of Engineering in Computer Engineering

BY

JUHI GUPTA

YASH BOHRA

ANSHUL JAIN

**Under the Guidance
Of**

MRS. SARITA AMBADEKAR

DEPARTMENT OF COMPUTER ENGINEERING

K. J. SOMAIYA INSTITUTE OF ENGINEERING AND

INFORMATION TECHNOLOGY SION, MUMBAI-22

UNIVERSITY OF MUMBAI, 2017 – 2018

CERTIFICATE

*This is to certify that the project entitled "**SENTIMENT ANALYSIS OF DATA FOR PREDICTING STOCK MARKET MOVEMENT**" is a bonafide work of "**JUHI GUPTA**" (17), "**ANSHUL JAIN**" (22), "**YASH BOHRA**" (10) submitted to University of Mumbai in partial fulfillment of the requirement in **Project II**, for the award of the degree of "**Bachelors of Engineering**" in "**Computer Engineering**"*

_____   _____

Project Guide &
I/c Head of Department   **Principal**
MRS. SARITA AMBADEKAR   **Dr. SURESH UKARANDE**

Place: Sion, Mumbai-400022
Date:

PROJECT APPROVAL FOR B. E.

This Project report entitled "**SENTIMENT ANALYSIS OF DATA FOR PREDICTING STOCK MARKET MOVEMENT**" by **Juhi Gupta, Anshul Jain, Yash Bohra** is approved for the degree of **Bachelors of Engineering in Computer Engineering**.

Examiners   1 _____

                                                      2 _____

Place:

Date:

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

JUHI GUPTA_____

YASH BOHRA_____

ANSHUL JAIN_____

Date:

ACKNOWLEDGEMENT

Before presenting out our project work entitled "**SENTIMENT ANALYSIS OF DATA FOR PREDICTING STOCK MARKET MOVEMENT**", we would like to convey our sincere thanks to many people who guided us throughout the course for this seminar work.

First, we would like to express our sincere thanks to our beloved Principal **Dr. SURESH UKARANDE** for providing various facilities to carry out this report. We would like to express our sincere thanks to **MRS. SARITA AMADEKAR** for her guidance, encouragement, co-operation and suggestions given to us at progressing stages of report.

Finally, we would like to thank our **H.O.D. PROF. SARITA AMBADEKAR** and all teaching, non-teaching staff of the college and friends for their moral support rendered during the course of the report work and for their direct and indirect involvement in the completion of our report work, which made our endeavor fruitful.

<div align="right">

JUHI GUPTA (B1-17)
YASH BOHRA (B1-10)
ANSHUL JAIN (B1-21)

</div>

Place: Sion, Mumbai-400022
Date:

ABSTRACT

Stock market prediction has been an active area of research for a long time. The Efficient Market Hypothesis (EMH) states that stock market prices are largely driven by new information and follow a random walk pattern. Though this hypothesis is widely accepted by the research community as a central paradigm governing the markets in general, several people have attempted to extract patterns in the way stock markets behave and respond to external stimuli.

Efficient Market Hypothesis being popular theory about stock prediction. With its failure much research has been carried in the area of prediction of stocks. Our project takes non quantifiable data such as financial news articles about a company and predicting its future stock trend with news sentiment classification. Assuming that news articles have impact on stock market, this is an attempt to study relationship between news and stock trend. Naïve Bayes gives good result. Experiments are conducted to evaluate various aspects of the proposed model and encouraging results are obtained in all of the experiments. The accuracy of the prediction model is more than 80% and in comparison with news random labeling with 50% of accuracy; the model has increased the accuracy by 30%.

Testing a hypothesis based on the premise of behavioral economics, that the emotions and moods of individuals affect their decision making process, thus, leading to a direct correlation between "public sentiment" and "market sentiment". We perform sentiment analysis on publicly available data to find the public mood and the degree of membership into 4 classes - Calm, Happy, Alert and Kind (somewhat like fuzzy membership). We use these moods and previous days' Dow Jones Industrial Average (DJIA) values to predict future stock movements and then use the predicted values in our portfolio management strategy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Earlier studies on stock market prediction are based on the historical stock prices. Later studies have debunked the approach of predicting stock market movements using historical prices. Stock market prices are largely fluctuating. The efficient market hypothesis (EMH) states that financial market movements depend on news, current events and product re-leases and all these factors will have a significant impact on a company's stock value. Because of the lying unpredictability in news and current events, stock market prices follow a random walk pattern and cannot be predicted with more than 50% accuracy.

With the advent of social media, the information about public feelings has become abundant. Social media is transforming like a perfect platform to share public emotions about any topic and has a significant impact on overall public opinion. Social media platform like Twitter, has received a lot of attention from researchers in the recent times. Twitter is a micro-blogging application that allows users to follow and comment other user's thoughts or share their opinions in real time. The social information exploited are very useful for making predictions.

1.2 BACKGROUND AND MOTIVATION

Short-term fluctuations in stock prices are generally considered to be extremely difficult to predict, primarily due to their nonlinear nature. The authors believe that one of the reasons for such seemingly unpredictable fluctuations is the type of sentiment prevailing amongst traders at that point in time. An attempt has been made in this study to forecast the stock returns using the sentiments expressed on social media.

1.3 PROBLEM STATEMENT

In the finance field, stock market and its trends are extremely volatile in nature. It attracts researchers to capture the volatility and predicting its next moves. Investors and market analysts study the market behavior and plan their buy or sell strategies accordingly. As stock market produces large amount of data every day, it is very difficult for an individual to consider all the current and past information for predicting future trend of a stock. Mainly

there are two methods for Forecasting market trends. One is Technical analysis and other is Fundamental analysis. Technical analysis considers past price and volume to predict the future trend where as Fundamental analysis On the other hand, Fundamental analysis of a business involves analyzing its financial data to get some insights. The efficacy of both technical and fundamental analysis is disputed by the efficient-market hypothesis which states that stock market prices are essentially unpredictable.

This research follows the Fundamental analysis technique to discover future trend of a stock by considering news articles about a company as prime information and tries to classify news as good (positive) and bad (negative). If the news sentiment is positive, there are more chances that the stock price will go up and if the news sentiment is negative, then stock price may go down. This research is an attempt to build a model that predicts news polarity which may affect changes in stock trends. In other words, check the impact of news articles on stock prices. We are using supervised machine learning as classification and other text mining techniques to check news polarity.

1.4 REQUIREMENT ANALYSIS

Economic Feasibility:
It refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for developing the product. If the benefits are the same as the older system, then it is not feasible to develop the product. In this product if we have developed this application then the amount of time spent in preparing the schedules, sending it different branches and monitor the work will be reduced which indirectly increases the production for the company. Since the project is implemented by using Python, which is open source.

Operational Feasibility:
It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real-time environment. It includes the study of additional human resource required and their technical expertise. This application will also work in any environment without any problems since we are implementing this project with python and flask, so it should be operationally feasible.

Technical Feasibility:

It refers to whether the software that is available in the market fully supports the present Application. It studies the pros and cons of using software for the development and its feasibility. It also studies the additional training needed to be given to the people to make the Application work. For this project, it's only the involvement of user that matters the most and the kind of interface that user will be comfortable with and the producing results at very fast speed. Our system only requires high INTEL processor computers for efficient functioning along with SSD and HDD storage tiers.

CHAPTER 2
REVIEW OF LITRATURE

In the paper published by Stanford university named "Sentiment analysis of news articles for financial signal prediction" by Jinjian Zhai , Nicholas Cohen and Anand Atreya. The main focus was on the analysis of publicly-available news reports with the use of computers to provide advice to traders for stock trading. They have developed Java data processing code and used the Stanford Classifier to quickly analyze financial news articles from the New York Times and predict sentiment in the articles. Two approaches were taken to produce sentiments for training and testing. A manual approach was tried using a human to read the articles and classifying the sentiment, and the automatic approach using market movements was used.

The work described in this project can be a module in a larger trading system. This module would extract sentiment from natural language and provide it as an input to that larger system. This project describes two approaches to predicting sentiment from news articles: a manually trained method, and an automatically trained method using market movements. The results of both of these approaches are reported, and the efficacy of using sentiment to predict profitable trading opportunities is discussed. Due to the time and resources limitation of this project, the problem is open-ended. It gives the foundation of the basic ideas of how to solve the problem using natural language processing (NLP) techniques, but does not describe a complete trading system. This work gives an example of how to parse information and link that information with the market, which in this case is defined as the S&P 500 index. A collection of New York Times articles in 2006 relating to business and finance was used as content for training and testing. These articles were used with the Stanford Classifier utilizing maximum entropy to train and predict sentiment.

The articles are taken from The New York Times Annotated Corpus. There corpus contains every article published in The New York Times from Jan 1987 to Jun 2007. Each article is annotated with date, category, and set of tags describing the content of the article. For the purposes of this project, we have chosen to focus exclusively on the articles that are directly related to the stock market, although future work could examine the value of sentiment analysis on other types of content.

In order to classify natural language sentiment of news articles, two methods were tested for determining sentiment: manual and automatic ones using stock market result. Manual classification involved reading each article and assigning it a sentiment tag: positive, neutral, or negative. A class, NYTManualClassifier, was built to aid in this process. Manual classification is obviously time consuming and we were only able to classify two months 3 worth of articles. January and June 2006 were chosen. Some temporal diversity was desired because news situations affecting the market can change, causing different overall sentiments, and because we noticed journalists focus on different types of stories at different times of the year. The classifier is not as good at classifying a mixture as when it is able to specialize on a single team member's classifications. In Automatic Classification Movements of the stock market were also used to generate classifications. They have realized that there data set would not be large enough to study the impact of article sentiment on individual stocks or even industries. For that reason they have decided to use S&P 500 index data to capture movements of the market as a whole. The log return: the log of today's close divided by yesterday's close.

They have designated log returns of greater than 0.0031 as positive, less than -0.0045 as negative, and anything in between as neutral. The value was set to make 36% of the dates positive, 28% neutral and 36% negative as we would like to test with higher volatility. They have realized that there is a lag in news publishing and news articles will typically be discussing the previous day's market movements. The article contains language about selling stock as part of the offering, which may have thrown off the classifier. Stock offerings would likely come up frequently and it might improve results to add features delineating stock offerings from articles about stock selloffs. In addition, feature writing does not have much to do with day to day stock movements. Filtering by metadata, to do this, we carefully examined the breadth of articles in the NYT corpus, looking for title, body text, and metadata features that would narrow down the articles to those likely to contain information of direct import to the stock market. The first feature that was identified was the "desk" metadata tag. Analysis showed that articles from The New York Times' "Financial Desk" were highly relevant to the stock market, while those from other departments at the newspaper were of significantly lower importance

Filtering by content, given the improvement that the metadata filtering made on our classification results, we wanted to examine further ways to improve article selection and thus potentially further improve our results. Having carefully examined the metadata of the articles, we turned our attention back to the content of the article. Combining this with the metadata filtering, we again improved our performance. In this case, while the performance on the positive class went down slightly (though it remained far above the original performance), the performance on neutral and negative increased appreciably. These improved filtering methods allowed us to get much better performance from relatively simple changes.

There are a number of ways this work could be improved. One idea tries to address the fact that most articles are much more nuanced than simply expressing positive or negative sentiment. As a result, it is worth analyzing the potential performance improvements if we perform classification at a more nuanced level - either by separately classifying different portions of an article or by allowing for classifications such as "somewhat positive". Another promising idea concerns article selection. Since we were focusing on the S&P 500 index, it is likely that we could improve our accuracy by further selecting articles exclusively about companies that are actually in that index. Furthermore, we could weigh the importance of such articles based on their relative proportions in the index. This would allow us to better model the relationship between sentiment and market performance.

This work has demonstrated the difficulty of extracting financially-relevant sentiment information from news sources and using it as a market predictor. While news articles remain a useful sort of information for determining overall market sentiment, they are often difficult to analyze and, since they are often focused on conveying nuanced information, may contain mixed messages. Furthermore, the success of this model relies largely upon the exploitation of market inefficiencies, which often take a great deal of work to identify if they are to be reliable. Thus, while our system provides interesting analysis of market sentiment in hindsight, it is less effective when used for predictive purposes. Nonetheless, given the coarse signals produced by our model, it is important to note that it is not necessary to trade directly using the values produced from our model. The sentiment results we produce could instead be an input to another trading system or simply be given to human traders to aid their judgments.

In a recent study by Anshul Mittal and Arpit Goel, we have studied how they have applied sentimental analysis and machine learning principles to find the correlation between public sentiment and market sentiment. According to the Efficient Market Hypothesis (EMH) states that stock market price are largely driven by the new information and follow a random pattern and we also use previous day DJIA values to predict the stock market movement using Self Organizing Fuzzy Neural network(SOFNN) on twitter feed and DIJA value we have implemented a naïve portfolio management strategy based on our management values.

In this paper, we have studied a hypothesis based on the premise of behavioural economics, that the emotions and moods of individuals affect their decision making process, thus, leading to a direct correlation between "public sentiment" and "market sentiment". We perform sentiment analysis on publicly available Twitter data to find the public mood and the degree of membership into 4 classes - Calm, Happy, Alert and Kind (somewhat like fuzzy membership). We use these moods and previous days' Dow Jones Industrial Average (DJIA) values to predict future stock movements and then use the predicted values in our portfolio management strategy.

The technique used in this paper builds directly on the one used by Bollean et al. The raw DJIA values are first fed into the pre-processor to obtain the processed values. At the same time, the tweets are fed to the sentiment analysis algorithm which outputs mood values for the four mood classes for each day. These moods and the processed DJIA values are then fed to our model learning framework which uses SOFNN to learn a model to predict future DJIA values using them. The learnt model as well as the previous DJIA and mood values is used by the portfolio management system which runs the model to predict the future value and uses the predicted values to make appropriate buy/sell decisions. The data obtained from the above mentioned sources had to be pre-processed to make it suitable for reliable analysis. While the Twitter data was available for all days lying in the giving period, the DJIA values obtained using Yahoo! Finance was (understandably) absent for weekends and other holidays when the market is closed.

In order to complete this data, approximated the missing values using a concave function. This approximation is justified as the stock data usually follows a concave function, unless of course at anomaly points of sudden rise and fall.

If we observe the general movement of stock markets, it is associated with a few sudden jumps/falls and a brief period of small fluctuations around the new value. However, such jumps/falls are due to some major aber-rations and cannot be predicted. Moreover, as we know the public memory is very short and even though the market may be trading at a much higher level than the previous year, that does not mean that calmness will be much higher than previous year; public mood is a very local metric. Sentiment Analysis: For sentiment analysis we have four moods: calm, happy, alert and kind. Using Opinion finder and sentiwordnet was inadequate and therefore they have developed their own analysis code.

They had followed following steps:
Word List Generation -> Tweet Filtering(n gram) -> Daily score Completion ->Score Mapping generation

Score Mapping, They have mapped the score of each word to the six standard POMS states using the mapping techniques specified in the POMS questionnaire. We then map the POMS states to our four mood states using static correlation rules (for example, happy is taken as sum of vigour and negation of depression). Model Learning and Prediction, They basically have used linear regression, but correlation between stock and mood is non-linear. Causality relation between past 3 day's mood and current day stock price, they tried 4-Learning algorithm. (Logic regression, Logist,regression, SVM,SOFNN). Portfolio Management, Naïve Greedy Strategy based on stock assumption that we can hold at most one stock at any given time.

We have investigated the causative relation between public mood as measured from a large scale collection of tweets from twitter.com and the DJIA values. Results in the paper show that firstly public mood can indeed be captured from the large-scale Twitter feeds by means of simple natural language processing techniques, as indicated by the responses towards a variety of socio-cultural events during the year 2009. Secondly, among the observed dimensions of moods, only calmness and happiness are Granger causative of the DJIA by 3-4 days. Thirdly, a Self Organizing Fuzzy Neural Network performs very good in predicting the actual DJIA values when trained on the feature set consisting of the DJIA values, Calm mood values and Happiness dimension over the past 3 days.

using, English speaking people. It's possible to obtain a higher correlation if the actual mood is studied. It may be hypothesized that people's mood indeed affects their in-vestment decisions, hence the correlation. But in that case, there's no direct correlation between the people who invest in stocks and who use twitter more frequently, though there certainly is an indirect correlation - investment decisions of people may be affected by the moods of people around them, ie. the general public sentiment. All these remain as areas of future research.

Twitter mood predicts the stock market [4]. In this paper we learn about behavioral economics that tells us that emotions can deeply affect individual behavior and decision making. In this paper we investigate whether the measurements of collective mood states derived from tweets and correlated to the value of DJIA (Dow Jones Industrial Average). To track the mood state, we have used two mood tracking tools.

(i) Opinion Finder

(ii) GPOMS (Google Profile of Mood States)

The paper has explained the process over three phases which are as follows:

1 First phase

(i) Collection of daily tweets subjected to two mood assessment tools.

(ii) Also extract time series of daily DJIA closing values from Yahoo Finance.

2 Second phase

(i) To find out mood measured and investigate whether it satisfies the hypothesis in predicting future DJIA values.

3 Third phase

(i) Deploy self-organizing Fuzzy Neural Network to test hypothesis that prediction of values can be improved.

Generating Public Mood Time Series Opinion Finder: We select positive and negative words that are marked as "weak" or "strong" from Opinion Finder lexicon. Then for each tweet we determine whether it contains any number of positive or negative tweets from OF lexicon. For each occurrence we increase score of positive or negative tweets by one and calculate the ratio of positive vs negative messages for tweets posted on same day.

GPOMS (Google profile of mood states): GPOMS can measure human mood state in terms of six different mood dimensions, namely calm, alert, sure, vital, kind and happy. So it can calculate much wider variety of natural occurring mood terms in tweets and map them to their respective POMS (profile of mood state) mood dimension.

Cross validating Opinion-Finder and GPOMS time-series against large sociocultural events: We first validate OF and GPOMS to capture various aspects of public mood. For this purpose, certain time period was chosen specifically that includes events such as US presidential election and thanksgiving

The result shows that OF successfully identifies the publics emotional response in relation to the presidential election. While GPOM results reveal a more differentiated public mood response. It characterized significant drop in calm mood state indicating high anxiety levels. To quantitatively determine the relation between GPOM's mood dimension and the OF mood trends, we test the correlation between mood trends using multiple regression.Through this paper we can speculate that the general public is presently as strongly invested in DJIA as financial experts and therefore there mood states will directly affect their investment decision and thus stock market values.

CHAPTER 3
PROJECT ANALYSIS AND DESIGN

A timeline represents the flow of the working of the project with its list of events in chronological order, also known as a project artifact. It is a graphical design that shows the dates with the tasks completed on their sides. The highlighted areas indicate the period of time wherein they were completed.

Following is the detailed design of the project including project timelines and task distribution.

3.1 PROJECT TIMELINES AND TASK DISTRIBUTION

Phase	Task Problem	Months									
		July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr
1	Problem Definition	██	██								
	Rigorous Study &Analysis		██	██	██	██					
2	Project Planning			██	██	██					
	Designing				██	██	██				
3	Implementation					██	██	██	██	██	
	Testing							██	██	██	██
	Modification								██	██	
	Deployment										██

Table 3.1: Project Timeline

3.2 TASK DISTRIBUTION

Task List	Assigned to	Status
Problem Definition	Juhi Gupta Anshul Jain Yash Bohra	Completed
Algorithm selection	Juhi Gupta Anshul Jain Yash Bohra	Completed
Collection of Data	Yash Bohra	Completed
GUI designing	Anshul Jain	Completed
Database designing	Anshul Jain	Completed
Preparing pickeled data and	Juhi Gupta	Completed
Implementing machine learning algorithm and saving	Juhi Gupta	Completed
Predicting Stock Price for the current date	Juhi Gupta Anshul Jain Yash Bohra	Completed
Paper publishing	Juhi Gupta Anshul Jain Yash Bohra	Completed
Documentation	Juhi Gupta Anshul Jain Yash Bohra	Completed

Table 3.2: Task distribution

The task distribution table is given below. It is a distributed view of the individual subtasks of the entire project and the group member(s) who have completed them. The 1-year plan has been broken down into manageable parts and helps us set measurable goals that are realistic and can be accomplished easily in small amounts of time. Table 3.2 shows this required data.

3.2 PROPOSED SYSTEM

The technique used in our project follows the following steps. The raw DJIA values are first fed into the preprocessor to obtain the processed values. At the same time, the Sentimental data are fed to the sentiment analysis algorithm which outputs mood values for the four mood classes for each day. These moods and the processed DJIA values are then fed to our model learning framework which uses algorithm to learn a model to predict future DJIA values. The learnt model runs to predict the future value and uses the predicted values to make appropriate buy/sell decisions.

3.2.1 Algorithm

Following are the steps that needs to be followed to implement our system

1. Collection of Sentimental data of a selected stock (using NYTimes API)

2. Collection of stock price of the selected stock (using Google Finance)

3. Performing Data Pre-Processing on the collected data

4. Applying sentimental Analyzer to extract the sentiments from the dataset (Vader Algorithm)

5. Applying Co-relational Analyzer on the data (Implenting MLP Machine learning Algorithm)

6. Predict stock market movement (using saved pickled learned MLP model)

3.3 DETAILED DESIGN

3.3.1 Flow Chart
A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. Common alternative names include: flow chart, process flowchart, functional flowchart, process map, process chart, functional process chart, business process model, process model, process flow diagram, work flow diagram, business flow diagram. The terms "flowchart" and "flow chart" are used interchangeably

The two most common types of boxes in a flowchart are:
1. A processing step, usually called activity, and denoted as a rectangular box
2. A decision, usually denoted as a diamond.

Fig 3.1: Flow Chart

3.3.2 DFD

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.[2] DFDs can also be used for the visualization of data processing (structured design).A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.
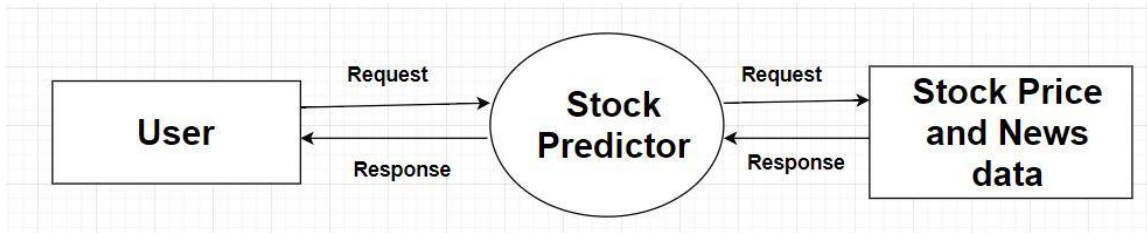
Fig 3.2: DFD Level 0 of Proposed System
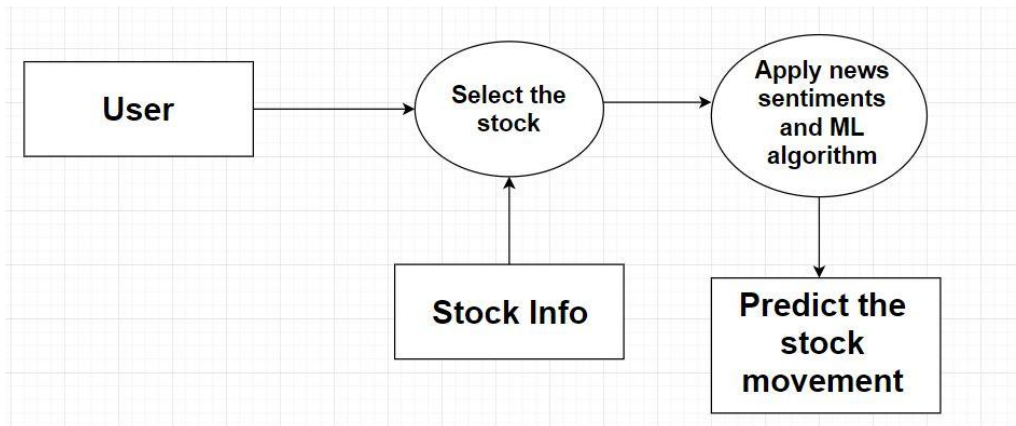
Fig 3.3: DFD Level 0 of Proposed System
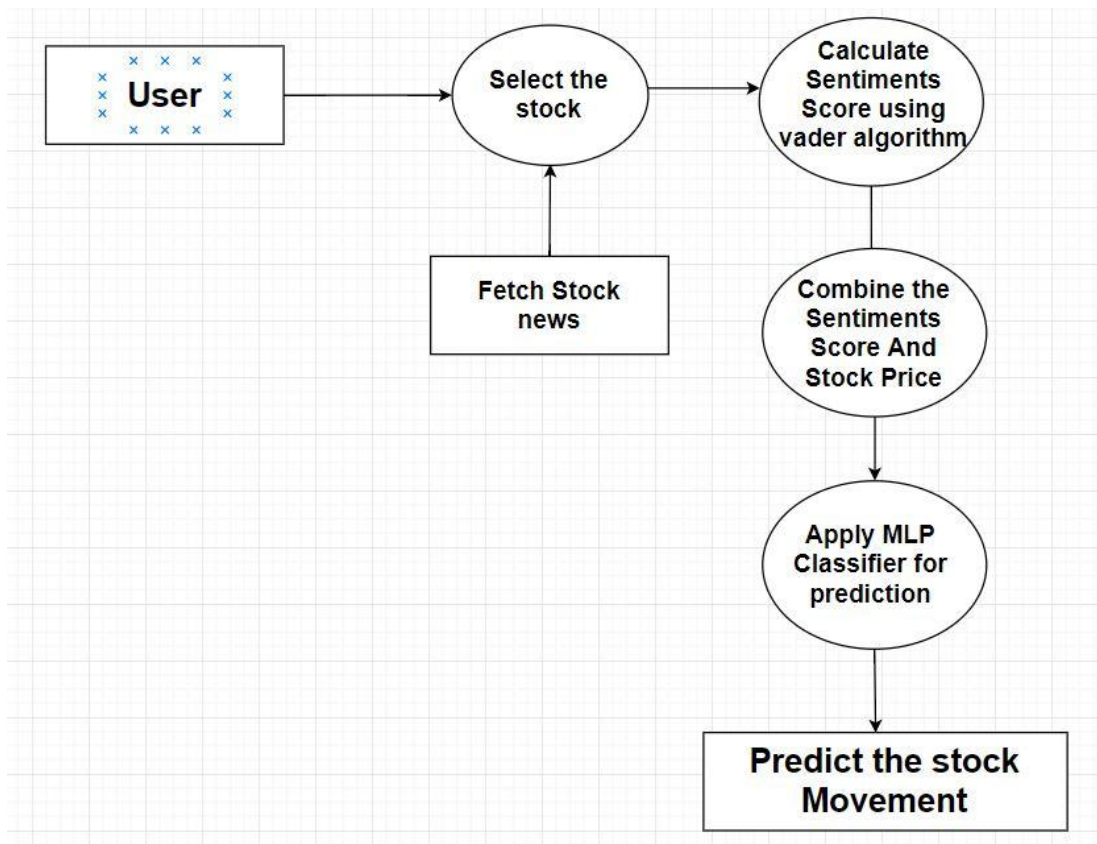
Fig 3.4: DFD Level 2 of Proposed System

3.3.3 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.
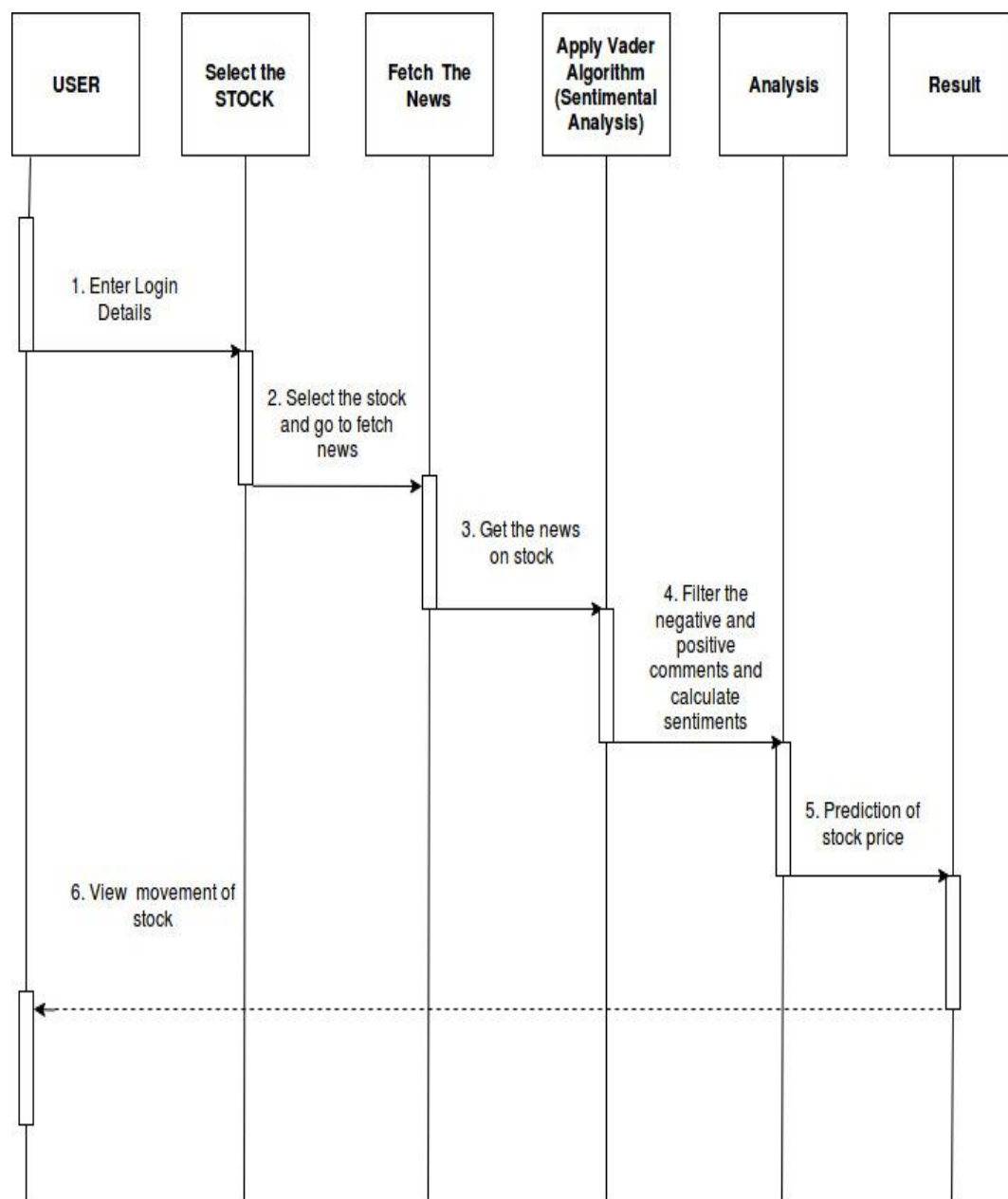
Fig 3.5: Sequence Diagram

16

3.4 DEVLOPMENT METHODOLOGY

3.4.1. MODULE 1 (DATA COLLECTION OF PUBLIC OPINION): To perform processing and prediction, we need to have data set to perform all the operations upon. Data are collected and then stored in a format which could help in easy processing of data. This will complete our task of building up Dataset which will be used in further subsequent modules.

```
┌─────────────────────────┐
│       Installing,       │
│  Authenticating of key to│
│   collect News from API │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    Adding constrain to  │
│     filter out content  │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│                         │
│     Store the Dataset   │
│                         │
└─────────────────────────┘
```
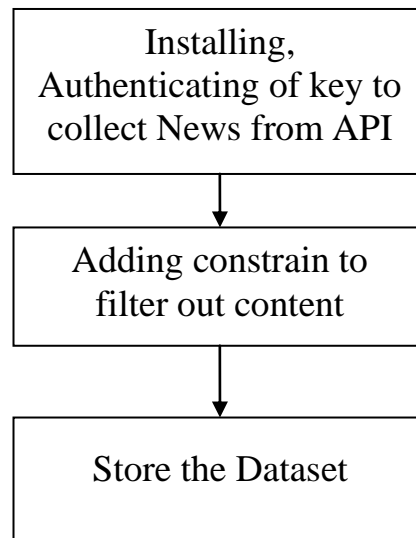
Fig 3.6: Flow chart of Module 1

3.4.2. MODULE 2 (DATA COLLECTION OF STOCK PRICES): To perform processing and prediction, we need to have data set to perform all the operations upon. Data are collected and then stored in a format which could help in easy processing of data. This will complete our task of building up Dataset which will be used in further subsequent modules.

```
┌─────────────────────────┐
│       Installing,       │
│  Authenticating of key to│
│  collect stock prices from│
│       required API      │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    Adding constrain to  │
│     filter out content  │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│     Store the Dataset   │
└─────────────────────────┘
```
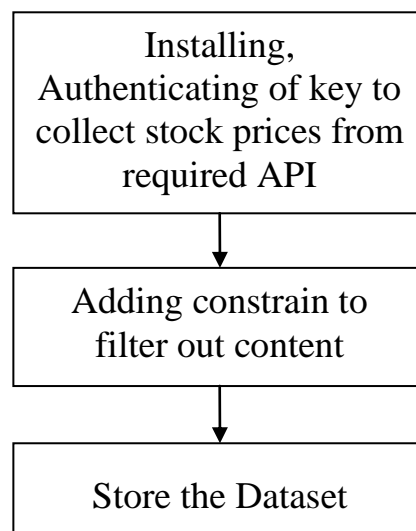
Fig 3.7: Flow chart of Module 2

3.4.3. MODULE 3 (SENTIMENTAL ANALYSIS & FEATURE EEXTRACTION):

Dataset which is available after the execution of previous modules will be used for sentimental analysis. We are going to apply sentimental analysis algorithm which will help us to extract emotions and calculate the score. Stock process of the stock are also used to extract the features which will be used in the consecutive modules

.

```
┌─────────────────────────┐
│        Dataset          │
│  (Sentimental news and  │
│   price of a particular │
│         stock)          │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Sentiment and Feature  │
│       Extraction        │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    Store the Extracted  │
│        Features         │
└─────────────────────────┘
```
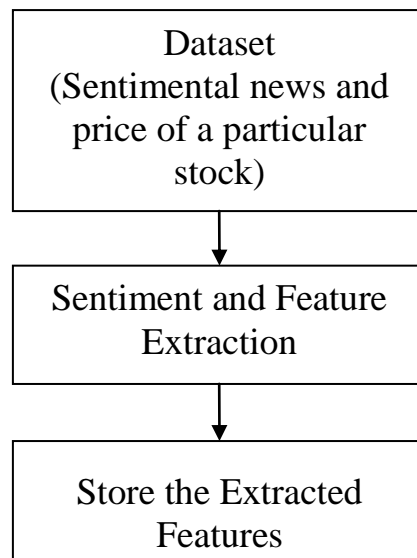
Fig 3.8: Flow chart of Module 3

3.4.4. MODULE 4 (TRAINING AND TESTING THE MODEL):

We are going to apply machine learning algorithm to predict the movement. Large amount of Dataset will be use to training the model and a small set of dataset will be used for testing the model

```
┌─────────────────────────┐
│  Dataset (Stock Price)  │
│   And Extracted Features │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    Training the Model   │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    Testing the Model    │
└─────────────────────────┘
```
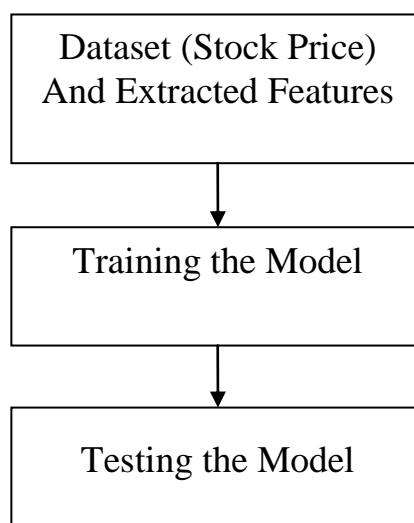
Fig 3.9: Flow chart of Module 4

18

3.4.5. MODULE 5 (PREDICTING THE STOCK MARKET MOVEMENT): This module basically gives the output, this module will tell us the short term prediction of stock. This will help us in portfolio management which is our desired aim.

```
┌─────────────────────┐
│  Live Dataset (Stock │
│  Price & Sentimental │
│       News)         │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│  Trained & Tested   │
│       Model         │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ Stock market Prediction │
└─────────────────────┘
```
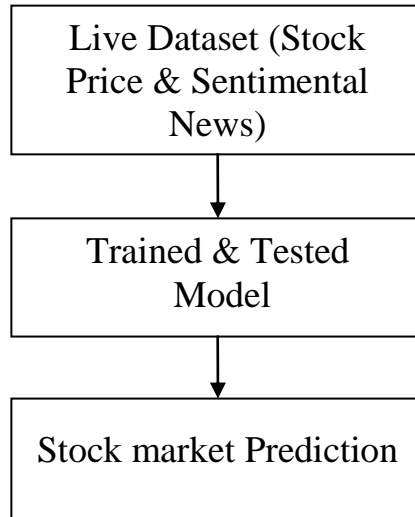
Fig 3.10: Flow chart of Module 5

CHAPTER 4
SYSTEM REQUIRMENTS

4.1   HARDWARE REQUIREMENTS

Processor   :   Intel Processor

 Memory   :   4GB RAM and above

Hard Disk   :   1 TB

4.2   SOFTWARE REQUIREMENTS

Operating System   :   Mac OS/ Windows OS/ Ubuntu

Front end   :   Flask in Python

Back end   :   Python

Software Required   :   Anaconda

4.3   COST ESTIMATION

Cost Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable. Cost Estimation determines how much money, effort, resources, and time it will take to build a specific system or product. Estimation is based on

1. Past Data/Past Experience

2. Available Documents/Knowledge

3. Assumptions

4. Identified Risks

The four basic steps in Software Project Estimation are

1. Estimate the size of the development product.

2. Estimate the effort in person-months or person-hours.

3. Estimate the schedule in calendar months.

4. Estimate the project cost in agreed currency.

Observations on Estimation

1. Estimation need not be a one-time task in a project. It can take place during −

1.1. Acquiring a Project.

    1.2. Planning the Project.

    1.3. Execution of the Project as the need arises.

2. Project scope must be understood before the estimation process begins. It will be helpful to have historical Project Data.

3. Project metrics can provide a historical perspective and valuable input for generation of quantitative estimates.

4. Planning requires technical managers and the software team to make an initial commitment as it leads to responsibility and accountability.

5. Past experience can aid greatly.

6. Use at least two estimation techniques to arrive at the estimates and reconcile the resulting values. Refer Decomposition Techniques in the next section to learn about reconciling estimates.

7. Plans should be iterative and allow adjustments as time passes and more details are known.

General Project Estimation Approach

The Project Estimation Approach that is widely used is Decomposition Technique. Decomposition techniques take a divide and conquer approach. Size, Effort and Cost estimation are performed in a stepwise manner by breaking down a Project into major Functions or related Software Engineering Activities.

Step 1 − Understand the scope of the software to be built.

Step 2 − Generate an estimate of the software size.

    2.1 Start with the statement of scope.

    2.2 Decompose the software into functions that can each be estimated individually.

    2.3 Calculate the size of each function.

2.4 Combine function estimates to produce an overall estimate for the entire project.

Step 3 − Generate an estimate of the effort and cost. You can arrive at the effort and cost estimates by breaking down a project into related software engineering activities.

3.1 Identify the sequence of activities that need to be performed for the project to be completed.

3.2 Divide activities into tasks that can be measured.

3.3 Estimate the effort (in person hours/days) required to complete each task.

3.4 Combine effort estimates of tasks of activity to produce an estimate for the activity.

3.5 Obtain cost units (i.e., cost/unit effort) for each activity from the database.

3.6 Compute the total effort and cost for each activity.

3.7 Combine effort and cost estimates for each activity to produce an overall effort and cost estimate for the entire project.

Step 4 − Reconcile estimates: Compare the resulting values from Step 3 to those obtained from Step 2. If both sets of estimates agree, then your numbers are highly reliable. Otherwise, if widely divergent estimates occur conduct further investigation concerning whether

4.1 The scope of the project is not adequately understood or has been misinterpreted.

4.2 The function and/or activity breakdown is not accurate.

4.3 Historical data used for the estimation techniques is inappropriate for the application, or obsolete, or has been misapplied.

Step 5 − Determine the cause of divergence and then reconcile the estimates.

4.3.1 FUNCTION POINT

A Function Point (FP) is a unit of measurement to express the amount of business functionality, an information system (as a product) provides to a user. FPs measure software size. They are widely accepted as an industry standard for functional sizing.For sizing software based on FP, several recognized standards and/or public specifications have come into existence.

ISO Standards

1. COSMIC − ISO/IEC 19761:2011 Software engineering. A functional size measurement method.

2. FiSMA − ISO/IEC 29881:2008 Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method.

3. IFPUG − ISO/IEC 20926:2009 Software and systems engineering - Software measurement - IFPUG functional size measurement method.

4. Mark-II − ISO/IEC 20968:2002 Software engineering - Ml II Function Point Analysis - Counting Practices Manual.

5. NESMA − ISO/IEC 24570:2005 Software engineering - NESMA function size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis

Object Management Group Specification for Automated Function Point: Object Management Group (OMG), an open membership and not-for-profit computer industry standards consortium, has adopted the Automated Function Point (AFP) specification led by the Consortium for IT Software Quality. It provides a standard for automating FP counting according to the guidelines of the International Function Point User Group (IFPUG).

1. Function Point Analysis (FPA) technique quantifies the functions contained within software in terms that are meaningful to the software users. FPs consider the number of functions being developed based on the requirements specification.

2. Function Points (FP) Counting is governed by a standard set of rules, processes and guidelines as defined by the International Function Point Users Group (IFPUG). These are published in Counting Practices Manual (CPM).

FP Counting Process involves the following steps:

Step 1 − Determine the type of count.

Step 2 − Determine the boundary of the count.

Step 3 − Identify each Elementary Process (EP) required by the user.

Step 4 − Determine the unique EPs.

Step 5 − Measure data functions.

Step 6 − Measure transactional functions.

Step 7 − Calculate functional size (unadjusted function point count).

Step 8 − Determine Value Adjustment Factor (VAF).

Step 9 − Calculate adjusted function point count.

Note : General System Characteristics (GSCs) are made optional in CPM 4.3.1 and moved to Appendix. Hence, Step 8 and Step 9 can be skipped.

Cost Estimation of Proposed System

Measurement Parameter	Count		Weighting Factor			
			Simple	Average	Complex	
Number of User Inputs	2	x	○ 3	○ 4	◉ 6	= 12
Number of User Outputs	1	x	○ 4	◉ 5	○ 7	= 5
Number of User Inquiries	1	x	◉ 3	○ 4	○ 6	= 3
Number of Files	225	x	○ 7	○ 10	◉ 15	= 3375
Number of External Interfaces	2	x	○ 5	○ 7	◉ 10	= 20
Count = Total					3415	

Rate each factor (Fi, i=1 to14) on a scale of 0 to 5

0	1	2	3	4	5
No Influence	Incidental	Moderate	Average	Significant	Essential
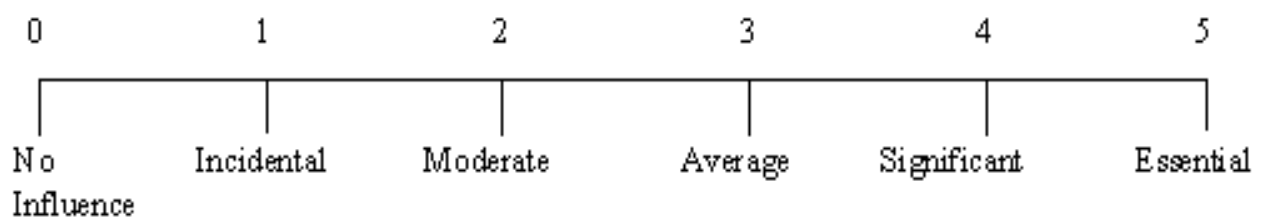
F1. Does the system require reliable backup and recovery? `3`

F2. Are data communications required? `4`

F3. Are there distributed processing functions? `4`

F4. Is performance critical? `5`

F5. Will the system run in a existing, heavily utilized operational environment? `1`

F6. Does the system require on-line data entry? `1`

F7. Does the on-line data entry require the input transaction to be built over multiple screens or operations? `4`

F8. Are the master files updated on-line? `4`

F9. Are the inputs, outputs, files or inquiries complex? `4`

F10. Is the internal processing complex? `4`

F11. Is the code designed to be reusable? `3`

F12. Are conversion and installation included in the design? `4`

F13. Is the system designed for multiple installations in different organizations? `1`

F14. Is the application designed to facilitate change and ease of use by the user? `4`

According to the Function Point Cost Estimation our Project has: 3790 FP

TEST EFFORTS

Test efforts are not based on any definitive timeframe. The efforts continue until some pre-decided timeline is set, irrespective of the completion of testing. This is mostly due to the fact that conventionally, test effort estimation is a part of the development estimation. Only in the case of estimation techniques that use WBS,such as Wideband Delphi, Three-point Estimation, PERT, and WBS, you can obtain the values for the estimates of the testing activities.

If we have obtained the estimates as Function Points (FP), then as per Caper Jones,

Number of Test Cases = (Number of Function Points) $\times$ 1.2

$$= 3790 \text{ x } 1.2$$

$$= 4548$$

Thus Number of test cases is 4548

Once you have the number of test cases, you can take productivity data from organizational database and arrive at the effort required for testing.

Percentage of Development Effort Method

Test effort required is a direct proportionate or percentage of the development effort. Development effort can be estimated using Lines of Code (LOC) or Function Points (FP). Then, the percentage of effort for testing is obtained from Organization Database. The percentage so obtained is used to arrive at the effort estimate for testing.

Estimating Testing Projects

Several organizations are now providing independent verification and validation services to their clients and that would mean the project activities would entirely be testing activities.

Estimating testing projects requires experience on varied projects for the software test life cycle. We are considering :

1. Team skills
2. Domain Knowledge
3. Complexity of the application
4. Historical data
5. Bug cycles for the project
6. Resources availability
7. Productivity variations
8. System environment and downtime

CHAPTER 5
IMPLEMENTATION AND SOLUTION

1 MODULE 1 (DATA COLLECTION OF PUBLIC OPINION): To perform processing and prediction, we need to have data set to perform all the operations upon. Data are collected and then stored in a format which could help in easy processing of data. This will complete our task of building up Dataset which will be used in further subsequent modules.

In order to collect news from News from NYTimes, We need to register with their API and collect access key
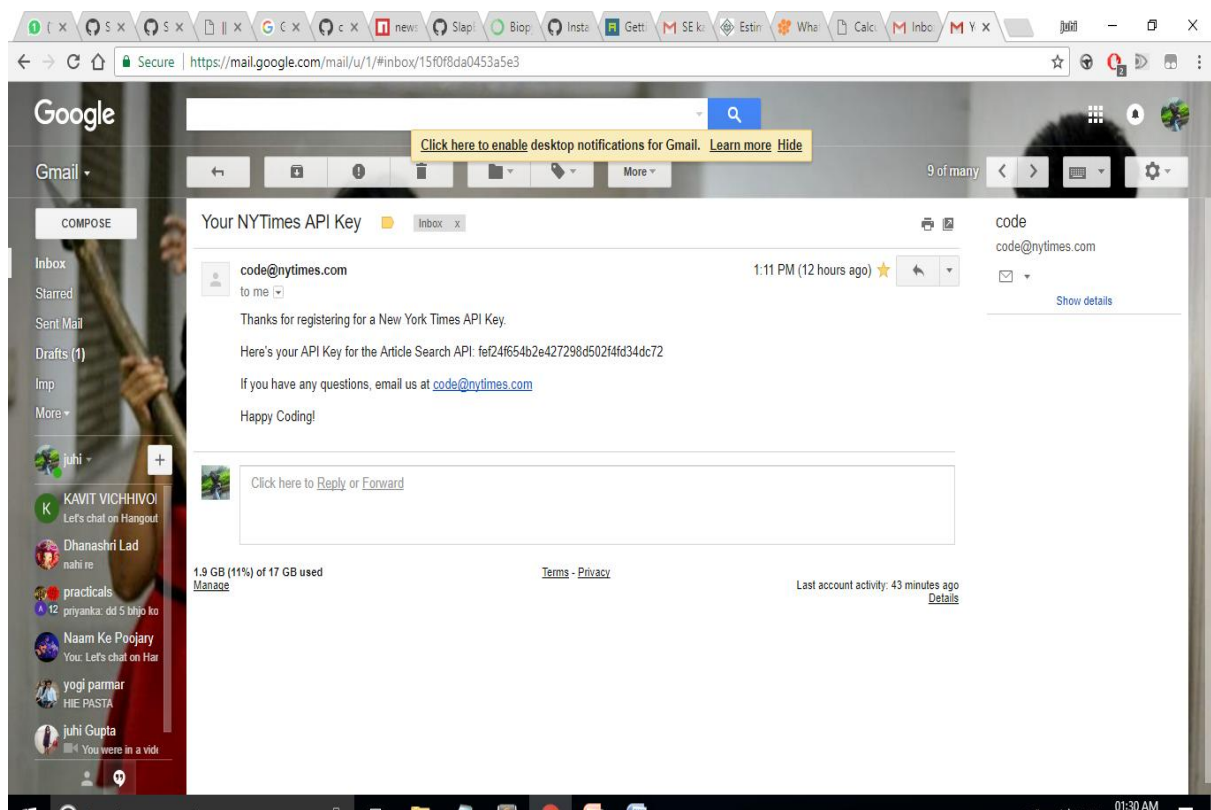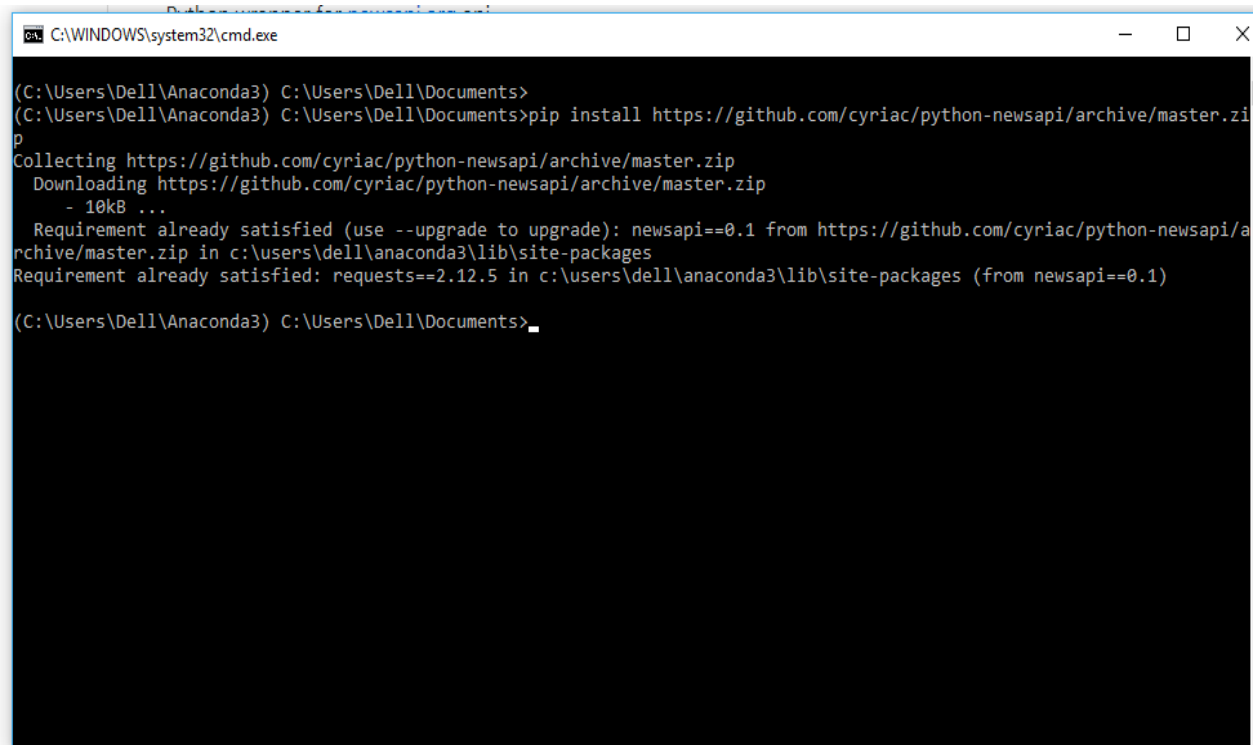
Fig 5.1: Authorization Key of NYTimes

After installation, we set up the entire required environment using Anaconda. We will install the required python libraries and there dependencies through Anaconda Prompt.

We will need following python Libraries for twitter collection

1. NyTimes

2. NyTimesArticles

3. NewsAPI

Fig 5.2: Command to install NewsAPI

Fig 5.3: Command to install NYTimes and NYTimesarticles

Fig 5.4: Executing code for Collecting NYtimesarticles Part 1

Fig 5.5: Executing code for Collecting NYtimesarticles Part 2
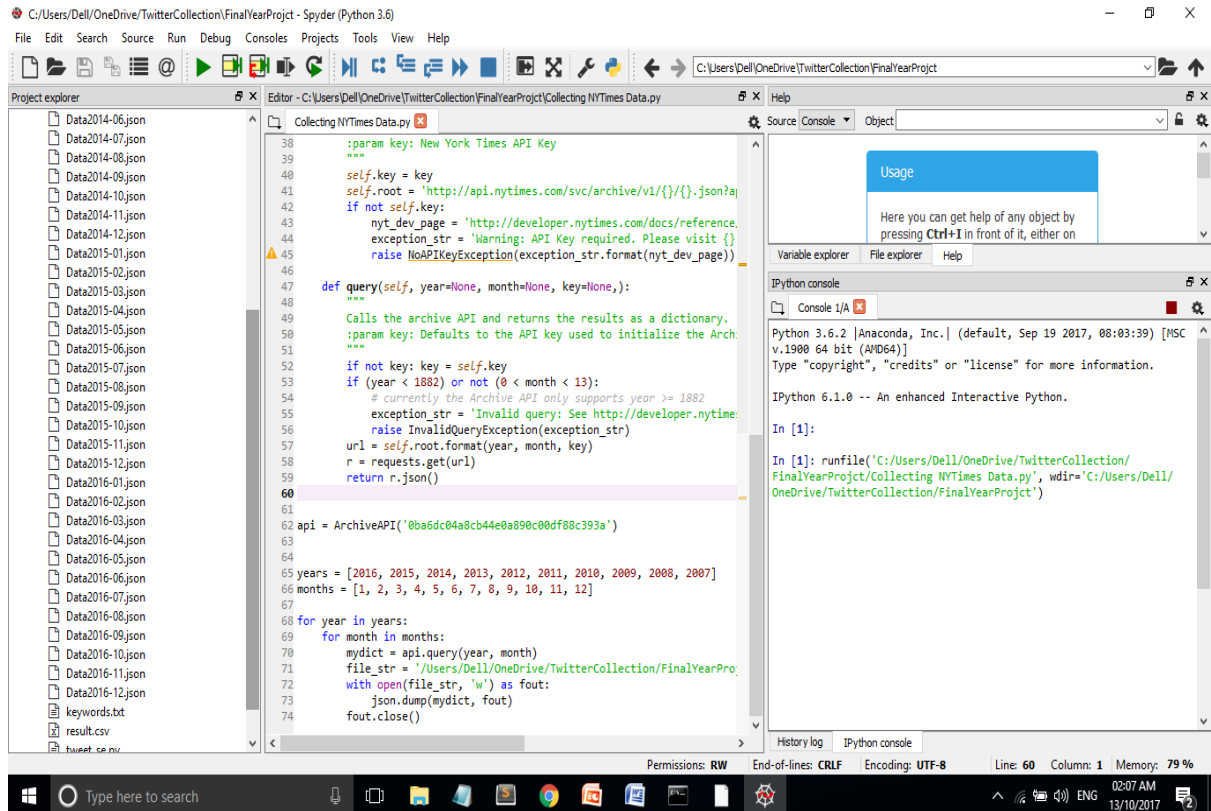
Fig 5.6: Collected nytimesarticles

Fig 5.7: Collected NYtimesarticles

Fig 5.8: Sample NYTimes Article
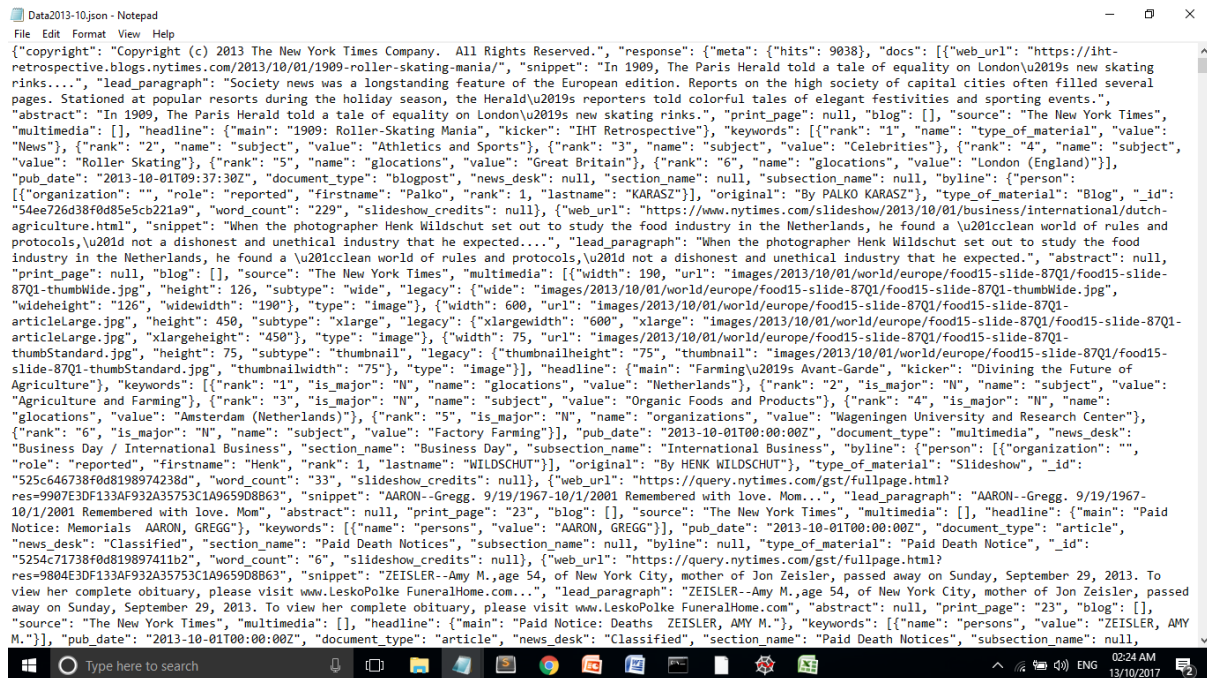
2 UI DESIGNING

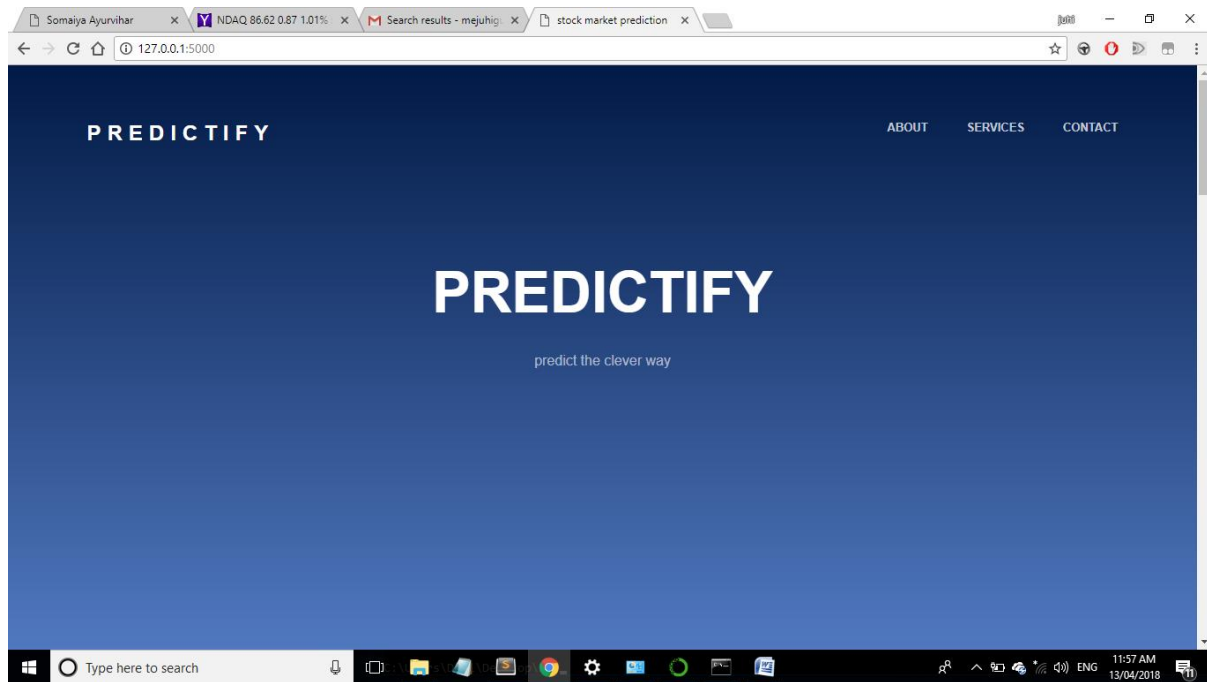STRATEGY   CONSULTING   STOCK PRICE ESTIMATION

Fig 5.9: UI Designing Part 1

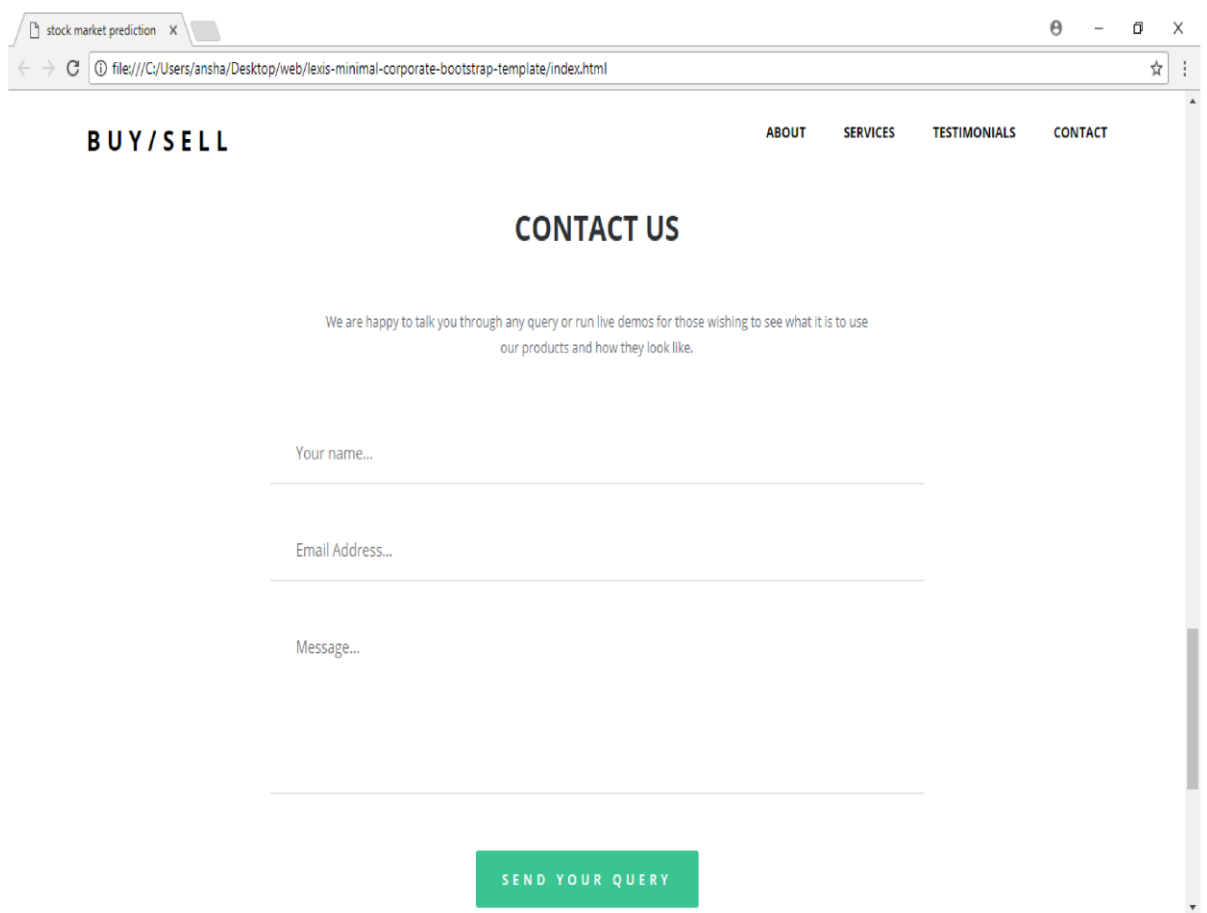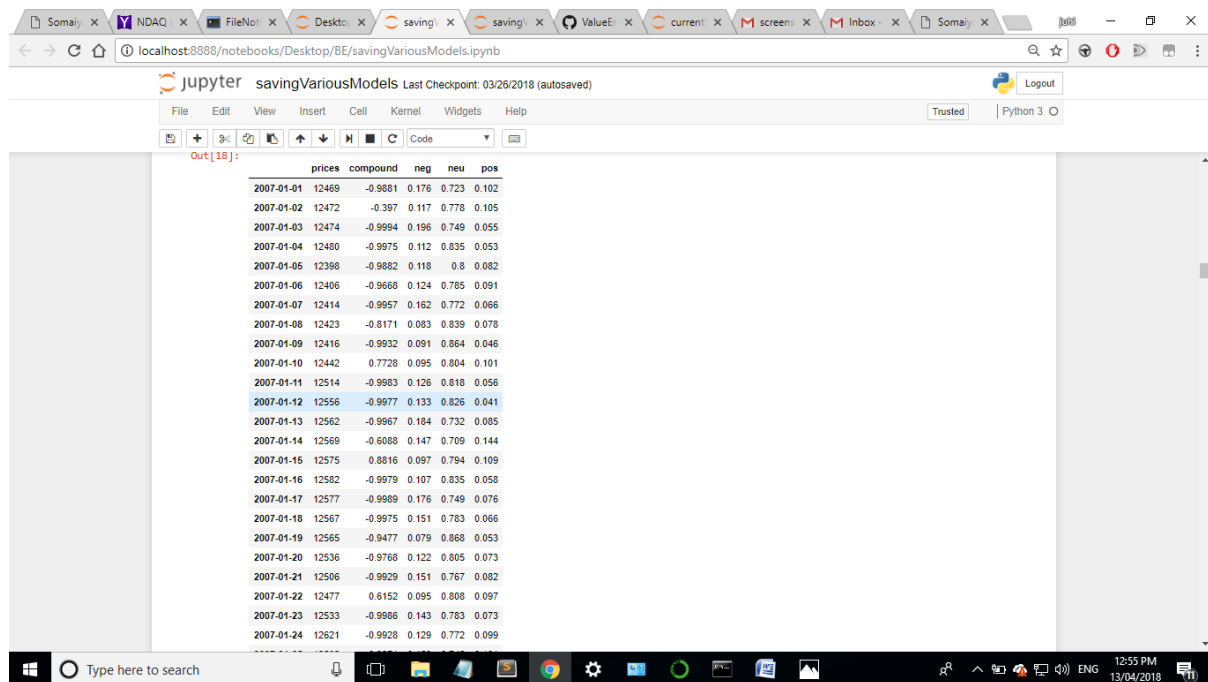Fig 5.10: UI Designing Part 2

Fig 5.11: UI Designing Part 3

3 MODULE 2 (DATA COLLECTION OF STOCK PRICES): To perform processing and prediction, we need to have data set to perform all the operations upon. Data are collected and then stored in a format which could help in easy processing of data. This will complete our task of building up Dataset which will be used in further subsequent modules.

We will be using Google Finance to collect the price of stock. We will collect the data and store the data. Google finance API needs to be installed and implemented in order to gather the data

4 MODULE 3 (SENTIMENTAL ANALYSIS & FEATURE EEXTRACTION): Dataset which is available after the execution of previous modules will be used for sentimental analysis. We are going to apply sentimental analysis algorithm which will help us to extract emotions and calculate the score. Stock process of the stock are also used to extract the features which will be used in the consecutive modules.

Fig 5.12: Sentimental score Snapshot

To extract the feature we need to apply various NLP algorithms like NaiveBayse Algorithm, Word set needs to be dined and feature extraction should be implemented to understand the sentiments (emotion) of the news and to determine its impact on the stock price

5 MODULE 4 (TRAINING AND TESTING THE MODEL): We are going to apply machine learning algorithm to predict the movement. Large amount of Dataset will be use to training the model and a small set of dataset will be used for testing the model
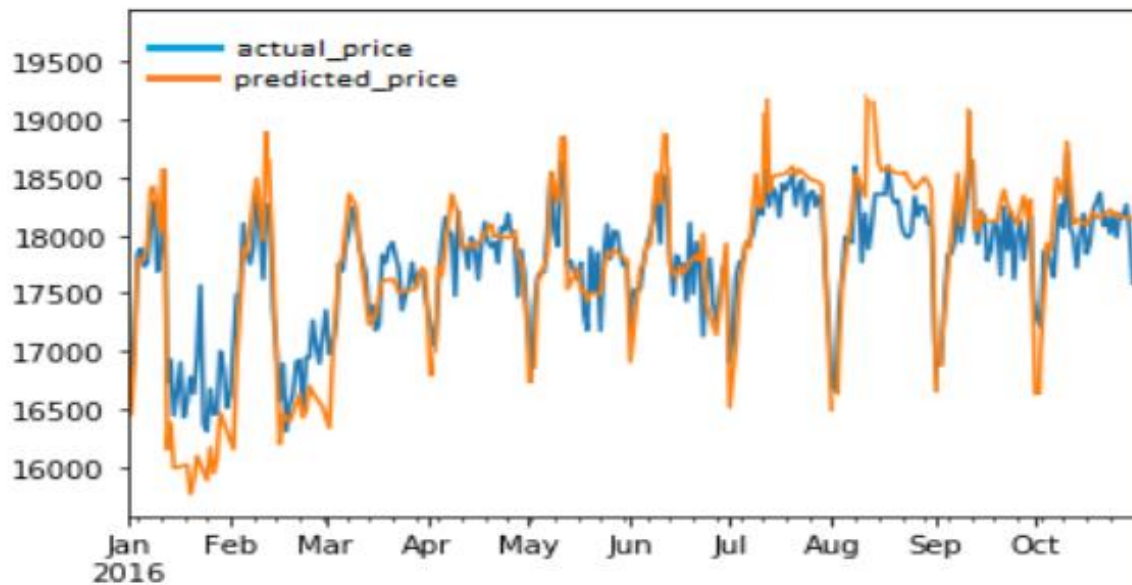

Fig 5.13: Result of testing data on learned model

We will be collecting around 10 years of dataset. We will use 7 years of data to train the system and then use 3 years of data to test the system. This will help us to get properly trained and tested model which will work efficiently and give good performance.
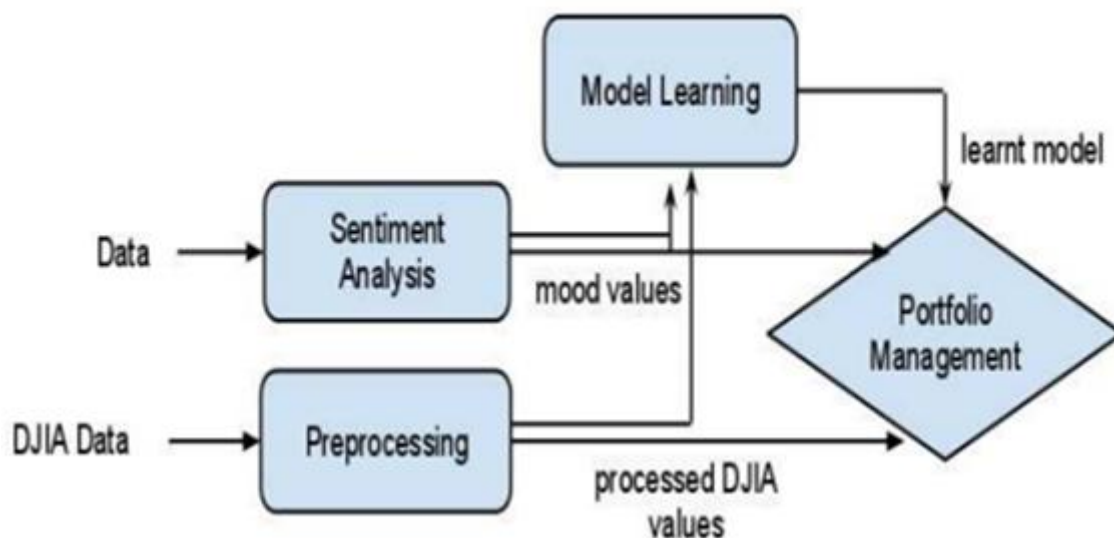
Fig 5.14: Representation of Model 4

6 MODULE 5 (PREDICTING THE STOCK MARKET MOVEMENT): This module basically gives the output, this module will tell us the short term prediction of stock. This will help us in portfolio management which is our desired aim.

Fig 5.15: Prediction of Stock Market Movement

Trained and tested model can take live data and give out short term predictions based on the current live data input. It will help us to predict the market and provide ways for portfolio management

Fig 5.16: Plotting stock market price

5.2 TESTING:

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

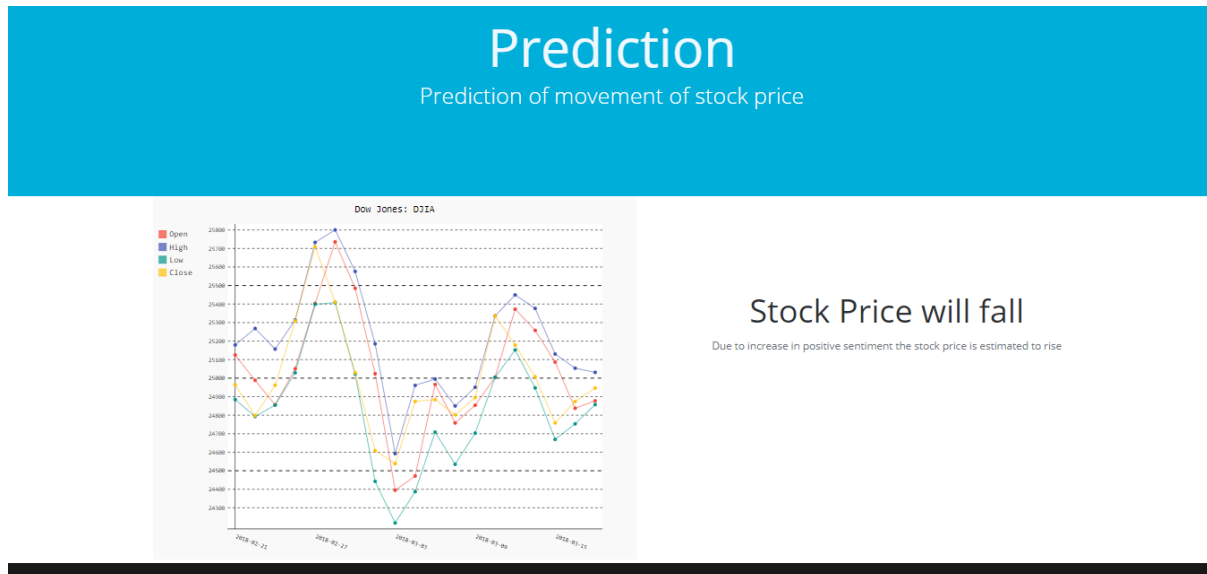**Black box testing:** Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

Specification-based testing: Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks.

Advantages and disadvantages: The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, and/or some parts of the back-end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

White box testing: White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.
Types of white box testing. The following types of white box testing exist:
• API testing (application programming interface) - Testing of the application using Public and Private APIs

• Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

• Fault injection methods - improving the coverage of a test by introducing faults to test code paths

• Mutation testing methods

• Static testing - White box testing includes all static testing

Code completeness evaluation

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

• Function coverage, which reports on functions executed

• Statement coverage, which reports on the number of lines executed to complete the test they both return a code coverage metric, measured as a percentage.

Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.

2. Acceptance testing performed by the customer, often in their lab environment on their own HW, is known as user acceptance testing (UAT).

Test Case ID	TEST	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	To Extract the news.	News should be collected	News are extracted and given for Sentiments calculation.	PASS
2	To Calculate Sentiment Of News Accurately	Sentimental Score should be Accurate	Sentiment Score is calculated correctly	PASS
3	To train and test the MLP Classifier algorithm	Tested data should have accurate output	Tested data is having good accuracy	PASS
4	To Plot the Graph	Able to show the historical price of given Stock.	Graph is plotted correctly showing the historical price.	PASS
5	To Predict Stock Movement Correctly	Able to Predict the Movement	Stock Movement is predicted properly.	PASS

Table 5.1 Test case

CHAPTER 6
SUMMARY

In this project, we have shown that a strong correlation exists between rise/fall in stock prices of a company to the public opinions or emotions about that company expressed on social platforms. The main contribution of our work is the development of a sentiment analyzer that can judge the type of sentiment present in the news. The News are classified into various categories. At the beginning, we claimed that positive emotions or negative sentiment of public in platform about a company would reflect in its stock price. Our speculation is well supported by the results achieved and seems to have a promising future in research.

Our results show that firstly public mood can indeed be captured in the large-scale feeds by means of simple natural language processing techniques. Secondly a good correlation Analyzer will work efficiently in predicting the actual stock values when trained on the feature set consisting of the stock values. Thirdly results obtained from the following approach is highly efficient than conventional method.

By using our Webapp even a naïve user gets stock market prediction at his finger tips, this could help him to develop a profile and do his portfolio management easily. Our project proves a platform to provide transparency to the user. User could either use strategy planning to see how well stock has done over a period of time and can also use prediction function which will tell user if that particular's stock price will rise or fall.

CHAPTER 7
FUTURE SCOPE

Sometimes its very difficult for a naive user to start with the stock market analysis. Our project gives a platform and automates the task of the user of going through the news and historic price of the stock. This method of analyzing social data with historic price will give us good results like 65% accuracy in the face value and 80% accuracy in predicting the right movement. But our project considers only aspect ie. news and historic price to predict the stock market movement.

If we are able to build the platforms which considers more aspects than mentioned above then we would be able to easily predict the stock market price with accuracy of 95%. If the model is able to consider twitter data, data from other social media as well such as instagram, facebook, trending topics from google trend, analytical data from google analytics we would be able to boost up the performance of the model we have trained eventually increasing the accuracy. Although constraint of mining the fake data posted by users on the social media should be handled otherwise it will yield wrong prediction.

REFERENCES

Journal Papers

[1] Jinjian (James) Zhai Nicholas (Nick) Cohen and Anand Atreya "Sentiment analysis of news articles for financial signal prediction" ,nlp.stanford.edu

[2] International conference on Signal Processing, Communication, Power and Embedded System "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements", (SCOPES)-2016

[3]Graham Bowley, "Wall Street Computers Read the News, and Trade on It", New York Times, Dec 21, 2010.

[4] Anshul Mittal "Stock Prediction Using Twitter Sentiment Analysis", Stanford University

[5] Twitter mood predicts the stock market, arXiv:1010.3003v1 [cs.CE] 14 Oct 2010

[6] Jasmina Smailovic Miha GrcarNada Lavrac Martin Znidarsic "Predictive Sentiment Analysis of Tweets: A Stock Market Application",

[7] Alec Go, Lei Huang and Richa Bhayani, "Twitter Sentiment Analysis", CS224N Final Report, 2009.

[8] Negative Feeling words, http://eqi.org/fw neg.htm

[9] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T1