Stock Prediction Using Sentimental Analysis

*Submitted in partial fulfilment of the requirements*

*of the degree of*

BACHELOR OF ENGINEERING

*in*

INFORMATION TECHNOLGY

(A.Y. 2016-2017)

by

|  |
| --- |
| Aakash Kamble (64)  Darshan Vakharia (65)  Rachit Verma (78) |

Under the Guidance of

Dr.Rajesh S. Bansode

Associate Professor, HOD-I.T Department, TCET



**University of Mumbai**

**Department of Information Technology**



Project Report Approval for Bachelor of Engineering

This project report entitled ***Stock Prediction Using Sentimental Analysis*** by***Aakash Kamble, Darshan Vakharia & Rachit Verma*** is approvedfor the degree of ***BACHELOR OF ENGINEERING inINFORMATION TECHNOLGY***.

|  |  |
| --- | --- |
| Signature :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name : Dr. Rajesh S. Bansode  HOD-IT | Signature :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Name : Dr. Rajesh S. Bansode  HOD-IT |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Name : Dr. B. K. Mishra  Principal,  Thakur College of Engineering and Technology. | |

|  |
| --- |
| **Examiners** |
| 1. Signature :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Name : |
| 1. Signature :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Name : |

Date:

Place:

**Declaration**

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I 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. I 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.

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Aakash Kamble, 64)

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Darshan Vakharia, 65)

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Rachit Verma, 78)

Date:

**C O N T E N T S**

**Chapter No. Topic Pg. No.**

* + - * Abstract
      * Certificate
      * Industry Certificate
      * List of Figures
      * List of Tables
      * Abbreviations and symbols
      * Acknowledgement

**Chapter1 Overview 01**

**1.1 Introduction**

1.2 Importance of the Project

1.3 Literature Survey

1.4 Motivation

**Chapter2 Proposed Work 05**

**2.1 Introduction**

2.2 Problem definition

2.3 Data Flow Diagram

2.4 Methodology

**2.5 Summary**

**Chapter3 Analysis and Planning 17**

**3.1 Introduction**

3.2 Feasibility Study

3.3 Project planning

3.4 Scheduling (Timeline chart)

**3.5 Summary**

**Chapter4 Results& Discussion 27**

4.1 Expected Results

4.2 Future Scope

**Chapter5 Conclusion 28**

5.1 Conclusion

Abstract

Stock market prediction has been a vital requirement of the investors. Computer science plays an important role in it. Well-organized and EMH had been one of the prominent theory about stock prediction. Collapse of it had resulted in research in the area of prediction of stocks. Yahoo finance API and google sentimental analysis API. The 5 min delay of yahoo finance API is the setback for real time capture of stock info.

The idea of project is about taking non quantifiable data such as financial twitts about a company and predicting its future stock trend with twitts (twitter) sentiment classification. Considering the fact that twitter data have impact on stock market, this is an attempt to figure out relationship between twitter data and stock trend. Classification models are created which depict polarity of twitts being positive or negative. Proposed model is assessed for various data and promising results are obtained. The project is powered by Google Sentimental analysis API which provided solidity.

#### Various AI and data mining techniques had been used till data. Prediction methodologies fall into three broad categories which can (and often do) overlap. They are [fundamental analysis](https://en.wikipedia.org/wiki/Fundamental_analysis), [technical analysis](https://en.wikipedia.org/wiki/Technical_analysis) (charting) and technological methods. Fundamental and technical analysis are based on traditional approaches and mainly used by the experts with through knowledge in the field. Technological is based on the various data mining technologies. They can be classified as Internet-based data sources for stock market prediction and Applications of Complexity Science for stock market prediction.

The prediction using neural network, based on the expert reviews and news data are some of the previous attempts made with successful results. They had being unable to get raw and abundant data. Our work is based on [1] Bollen et al’s famous paper which predicted the same with 87% accuracy. The twitter data was used in his prediction method. The goal is to automate the work and improve the accuracy.

The project serves as a platform where both the experts and the common users can make use of the data. The deficiencies in the previous techniques are reviewed and the attempt is to minimize them and make the algorithm as efficient as possible.



**CERTIFICATE**

This is to certify that Mr. Aakash Kamble, Mr. Darshan Vakharia, Mr. Rachit Verma are bona-fide students of Thakur College of Engineering and Technology, Mumbai. They have satisfactorily completed the requirements of the PROJECT-I as prescribed by University of Mumbai while working on “Stock Prediction Using Sentimental Analysis”.

|  |  |  |
| --- | --- | --- |
| **Dr. Rajesh Bansode**  **(Guide)** | **Dr. Rajesh Bansode**  **( HOD - IT)** | **Dr. B. K. Mishra**  **(Principal)** |

**Internal Examiner External Examiner**

**(Name and Signature with Date) (Name and Signature with Date)**

Thakur College of Engineering and Technology Kandivali(E), Mumbai-400101.

PLACE: Mumbai

DATE:



Date: August 4th, 2016

To Whomsoever It May Concern,

We at Sty-Labs are pleased to inform you that the following students:-

1. Aakash Kamble
2. Darshan Vakharia
3. Rachit Verma

from ‘Thakur College Of Engineering And Technology’ will be working with us for their Final Year B.E Project titled: ‘Stock Rates Prediction Using Sentimental Analysis’ under the guidance of the skilled software developers here at Sty-Labs.

The working hours for the same would be as follows:

Monday: 9:30 AM to 6:30 PM

Friday: 3:30 PM to 7:30 PM

Alternate Saturday: 9:30 AM to 6:30 PM

(Until the completion of the project)

All the rights of the project developed will solely belong to the above mentioned group members. The students are required to be present at the given days and time to prevent any loss to their academics and to the company’s working hours.

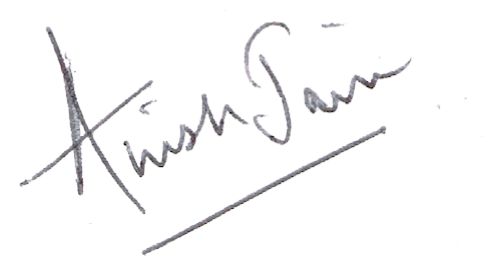
The organization expects full support from the college for the same.

The college can feel free to contact Mr. Vignesh Prabhu who will be overlooking their project work on the organizations behalf for any progress reports or details regarding the project.

Mr. Vignesh Prabhu : 9619524024

With best wishes,

For StyLabs Info Solution Pvt Ltd.



**Anish Jain  
Head of Product**

**List of Figures**

Figure 2.1 : Data Flow Diagram (Layer 0)

Figure 2.2 : Data Flow Diagram (Layer 1)

Figure 2.3 : Core Mechanism

Figure 2.4 : Process Flow

Figure 2.5 : Classification Engine of proposed Sentiment Analysis Model

Figure 3.1 : Gantt Chart

**List of Tables**

Table I: Technical Feasibility

Table II: Cost requirement for developing required system

Table III: Result Analysis

Table IV: Output, Outcome Mapping

**Abbreviations**

1. AI : Artificial Intelligence
2. API : Application Program Interface
3. DA: Data Analytics
4. DJIA: Dow Jones Industrial Average
5. EMH: Efficient Market Hypothesis
6. GPOMS: Google Profile of Mood States
7. IEEE: Institute Of Electrical And Electronics Engineers
8. MKL: Math Kernel Library
9. RF: Random Forest
10. SOFNN: Self-Organizing Fuzzy Neural Network

**ACKNOWLEDGEMENT**

This project was supported by Thakur College Of Engineering & Technology and Sty-Labs. We are thankful to our guide Dr. Rajesh Bansode who provided  expertise that greatly assisted the research. We also thank the project coordinators for arranging the necessary facilities to carry out the project work.

We have to express out appreciation to the Mr. Vignesh Prabhu for sharing their pearls of wisdom with us during the course of this project. We are also immensely grateful to Mr. Anish Jain for their comments on an earlier versions of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed professionals.

We thank the Dr. Rajesh Bansode (HOD - IT), Dr. R.R Sedamkar (Dean Academic), Dr. B. K. Mishra (Principal) and the college management for their support.

**(Aakash Kamble, 64)**

**(Darshan Vakharia, 65)**

**(Rachit Verma, 78)**

**Chapter 1**

**Overview**

**1.1 INTRODUCTION**

Stock market and its trends have extremely volatile nature. It lures researchers to capture the volatility and predicting its next moves. Investors and market analysts study the market behaviour 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 analysing 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.

**1.2 Importance of the project**

The 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 twitter 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 twitts polarity which may affect changes in stock trends. In other words, check the impact of twitter data on stock prices. We are using supervised machine learning as classification and other text mining techniques to check twitts polarity. And also be able to classify unknown twitts, which is not used to build a classifier. Three different classification algorithms are implemented to check and improve classification accuracy.

**1.3 Literature Survey**

Stock price trend prediction is an active research area, as more accurate predictions are directly related to more returns in stocks. Therefore, in recent years, significant efforts have been put into developing models that can predict for future trend of a specific stock or overall market. Most of the existing techniques make use of the technical indicators. Some of the researchers showed that there is a strong relationship between news article about a company and its stock prices fluctuations. Our project is supported by Bollen et al’s [1] theory. Following is discussion on previous research on sentiment analysis of text data and different classification techniques.

Nagar and Hahsler in their research [3] presented an automated text mining based approach to aggregate news stories from various sources and create a News Corpus. The Corpus is filtered down to relevant sentences and analyzed using Natural Language Processing (NLP) techniques. A sentiment metric, called TwitterSentiment, utilizing the count of positive and negative polarity words is proposed as a measure of the sentiment of the overall news corpus. They have used various open source packages and tools to develop the news collection and aggregation engine as well as the sentiment evaluation engine. They also state that the time variation of TwitterSentiment shows a very strong correlation with the actual stock price movement. Yu et al [4] present a text mining based framework to determine the sentiment of news articles and illustrate its impact on energy demand. News sentiment is quantified and then presented as a time series and compared with fluctuations in energy demand and prices. J. Bean [5] uses keyword tagging on Twitter feeds about airlines satisfaction to score them for polarity and sentiment. This can provide a quick idea of the sentiment prevailing about airlines and their customer satisfaction ratings. We have used the sentiment detection algorithm based on this research.

This research paper [6] studies how the results of financial forecasting can be improved when Twitter data with different levels of relevance to the target stock are used simultaneously. They used multiple kernels learning technique to partition the information extracted from five different categories of news articles based on sectors, sub-sectors, industries etc. Twitter data are divided into the five categories relevant to a targeted stock, its sub industry, industry, group industry and sector while separate kernels are employed to analyze each one. The experimental results show that the cumulative use of various tweets categories increases the prediction performance in comparison with methods based on a lower number of news categories. It shows that highest prediction accuracy and returns gained per trade were achieved for MKL when all of the five categories of tweets were utilized with two separate kernels of the polynomial and Gaussian types used for each tweets category.

Chan., Wong., and Lam., implemented a neural network model using the technical analysis variables for listed companies in Shanghai Stock Market. In this paper performance of two learning algorithm and two weight initialization methods are compared. The results reported that prediction of stock market is quite possible with both the algorithm and initialization methods but the performance of the efficiency of the back propagation can be increased by conjugate gradient learning and with multiple linear regression weight initializations.  
Other prominent literatures are that of Siekmann et al. (2001) who used fuzzy rules to split inputs into increasing, stable, and decreasing trend variables. Siekmann et al. (2001) implemented a network structure that contains the adaptable fuzzy parameters in the weights of the connections between the first and second hidden layers.

Kim and Han (2000) used a genetic algorithm to transform continuous input values into discrete ones [7]. The genetic algorithm was used to reduce the complexity of the feature space.  
Forecasting Of Indian Stock Market Index Using Artificial Neural Network. Kishikawa and Tokinaga (2000) used a wavelet transform to extract the short-term feature of stock trends.  
Kim and Han (2000) used neural network modified by Genetic Algorithm. Kim and Chun (1998) used refined probabilistic NN (PNN) to predict a stock market index [8]. Pantazopoulos et al. (1998) presented a neurofuzzy approach for predicting the prices of IBM stock.  
Chenoweth, Tim., Obradovic, Zoran., used specialized neural network as preprocessing component and a decision rule base. The preprocessing component determine the most relevant features for stock market prediction, remove the noise, and separate the remaining patterns into two disjoint sets.

Next, the two neural networks predict the market's rate of return, with one network trained to recognize positive and the other negative returns. Some work has also been reported in portfolio construction, for Roman, Jovina and Jameel, Akhtar in their paper proposed a new methodology to aid in designing a portfolio of investment over multiple stock markets [9]. For that they used backpropagation and recurrent network and also the contextual market information. They developed a determinant using the accuracy of prediction of the neural network and the stock return of the previous year and used it to select the stock market among other markets

In the work by Refenes et al, (1997), they compared the performance of backpropagation network and regression models to predict the stock market returns. Desai, V. S. (1998), compared the performance of linear regression with that of the neural network in forecasting the stock returns.

In many papers ARIMA model has been used as a benchmark model in order to compare the forecasting accuracy of the ANN. Jung-Hua Wang; Jia-Yann Leu developed a prediction system of recurrent neural network trained by using features extracted from ARIMA analysis. Then after differencing the raw data of the TSEWSI series and then examining the autocorrelation and partial autocorrelation function plots, they identified the series as a nonlinear version of ARIMA (1,2,1). Neural networks were trained by using  
second difference data and were seen to give better predictions than otherwise trained by using raw data. Jingtao Yao, Chew Lim Tan and Hean-Lee Poh developed a neural network that was used to predict the stock index of Kuala Lumpur Stock Exchange. The used trading strategies to a paper profit were recorded and were compared with that of the ARIMA model. The results showed that the performance of the neural net was better than that of the ARIMA. It was also asserted that useful prediction can be made even without the use of extensive data or knowledge.

Researchers have tested the accuracy of ANN in predicting the stock market index return of most developed economies across the globe. Literatures are available for forecasting index returns of U.S markets like NYSE, FTSE, DJIA, S&P500. Few papers are also available in context to Asian stock markets like Hang Seng Stock Exchange, Korea Stock Exchange Tokyo Stock Exchange and Taiwan Stock Exchange.

Some literatures are also available in Indian context. Panda, C. and Narasimhan, V. used the artificial neural network to forecast the daily returns of Bombay Stock Exchange (BSE) Sensitive Index (Sensex) [10]. They compared the performance of the neural network with performances of random walk and linear autoregressive models. They reported that neural network out-performs linear autoregressive and random walk models by all performance measures in both in-sample and out-of-sample forecasting of daily BSE Sensex returns.

In another paper, Dutta,G. et.al. studied the efficacy of ANN in modeling the Bombay Stock Exchange (BSE) SENSEX weekly closing values [11]. They developed two networks with inputs as the weekly closing value, 52-week moving average of the weekly closing SENSEX values, 5-week moving average of the same, Forecasting Of Indian Stock Market Index Using Artificial Neural Network. and the 10-week Oscillator for the past 200 weeks for one neural net. And for the other network the inputs are the weekly closing value, 52-week moving average of the weekly closing SENSEX values, 5-week moving average of the same and the 5-week volatility for the past 200 weeks.

To assess the performance of the networks they used the neural networks to predict the weekly closing SENSEX values for the two-year period beginning January 2002. The root mean square error (RMSE) and mean absolute error (MAE) are chosen as indicators of performance of the networks. The proposed network has been tested with stock data obtained from the Indian stock Market BSE Index. Bishnoi T. R., et al has analyzed the behavior of daily and weekly returns of five Indian stock market indices for random walk during April-1996 to June-2001. They have tested the indices for normality, autocorrelation using Qstatistic & Dickey-Fuller test and analyzed variance ratio using homoscedastic and heteroscedastic test estimates. The results support that Indian stock market indices do not follow random walk. The previous studies have used various forecasting techniques in order to predict the stock market trends.

Some attempted to forecast the daily returns where others developed forecasting models to predict the rate of returns of individual stocks. In many papers it was also found that researchers have attempted to compare their results with other statistical tools. And these findings provide strong motivation for modeling forecasting tools for stock market prediction.  
The uniqueness of the research comes from the fact that the research employs a neural network based forecasting approach on National Stock Exchange index (CNX S&P Nifty 50)

Furthermore, as not much work has been done on the forecasting of Indian stock market indices using neural network, this paper will actually help to understand the microstructure of Indian market.

**1.4 Motivation**

The main motivation for this project is to develop a better stock options price prediction system, that investors as well as speculators can use to maximize their returns.

Many research papers have been written to help investors predict stock price as well as pick portfolio to maximize their returns. However, the there isn’t much papers that exist for maximizing returns on stock options or predicting stock option prices. Looking at NIPS and ICML, I saw there is a papers for better comping up with a better stock price predicting, based on the given conditions; however, it assume that you already knowing the stock prices for the option.

The goal of this project is develop a system that will help predict the percent price change for a stock and use that to predict the percent price change of the underlying option for the stock. This can also be used in order to help predict best option portfolio to help or develop an options trading strategy to maximize returns.

**Chapter 2**

**Proposed Work**

* 1. **Introduction**

The Efficient Market Hypothesis (EMH) [12, 14] 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. These moods and previous days’ Dow Jones Industrial Average (DJIA) values are used to predict future stock movements and then use the predicted values in our portfolio management strategy.

**Related work** Our work is based on Bollen et al’s strategy [1] which received widespread media coverage recently. They also attempted to predict the behavior of the stock market by measuring the mood of people on Twitter. The authors took into consideration the tweet of all twitter users in 2008 and used Opinion Finder and Google Profile of Mood States algorithm to classify sentiment of public into six categories, namely, Happy, Vital, Alert, Kind, Calm, and Sure. Cross validation of resulting mood time series by comparing its ability to detect the response of public to the presidential elections and Thanksgiving Day in 2008 was done by them. Also they used causality analysis to investigate the hypothesis that public mood states, as measured by the Opinion Finder and GPOMS mood time series[7], are predictive of changes in DJIA closing values. Self Organizing Fuzzy Neural Networks is used by the researchers to predict DJIA values using previous values. A remarkable accuracy of nearly 87% in predicting the up and down changes in the closing values of Dow Jones Industrial Index (DJIA) is shown by their results [3].

**2.2 Problem Definition**

Market price of the equity shares is very difficult to predict and it is based on historical prices of the stock but it is an important tool for short term investors for achieving maximum profits. MLP neural networks have been the existing method for price predictions. However, MLP neural network does not always give accurate prediction in case of volatile markets. In our app we are introducing a new way of collaborating both neural networks and decision tree to forecast the stock market price more accurately than MLP.

* 1. **Data Flow Diagram**

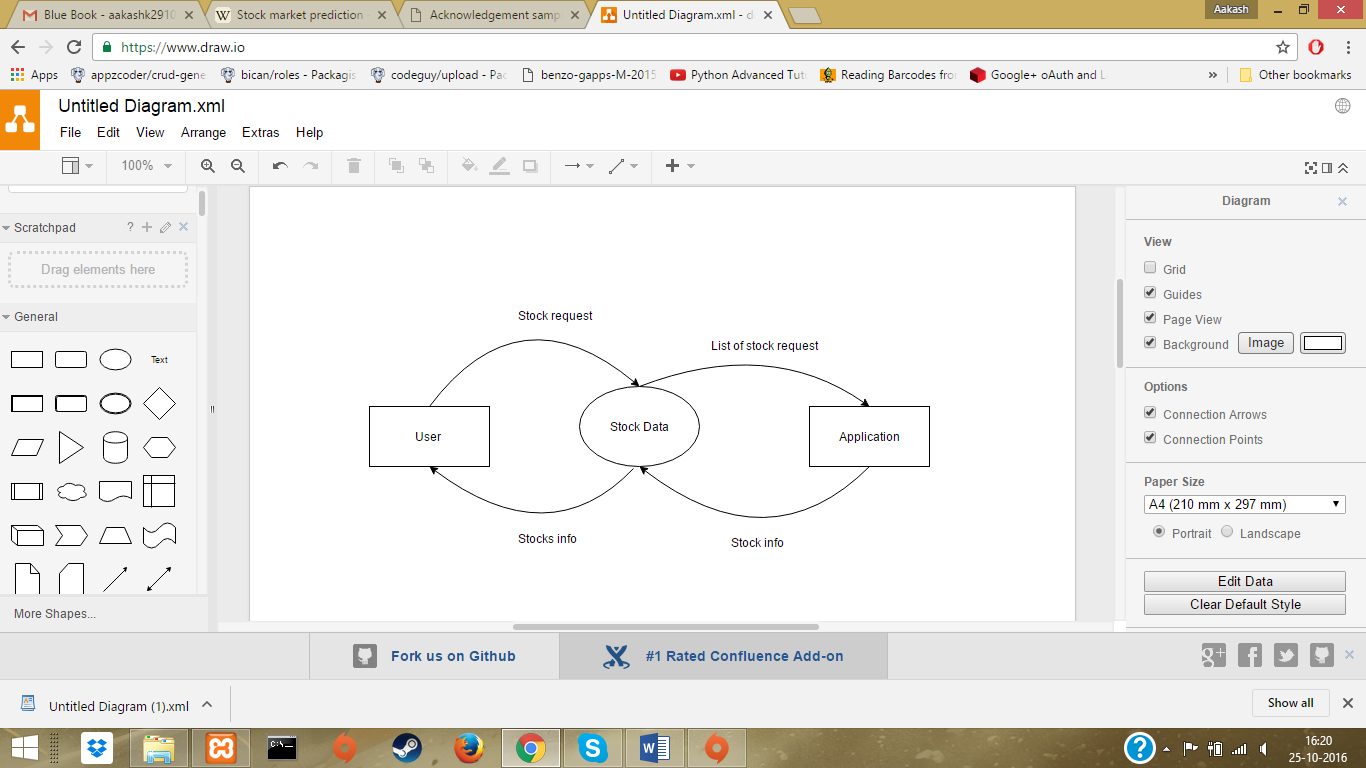


Figure 2.1 Data Flow Diagram (Layer 0)

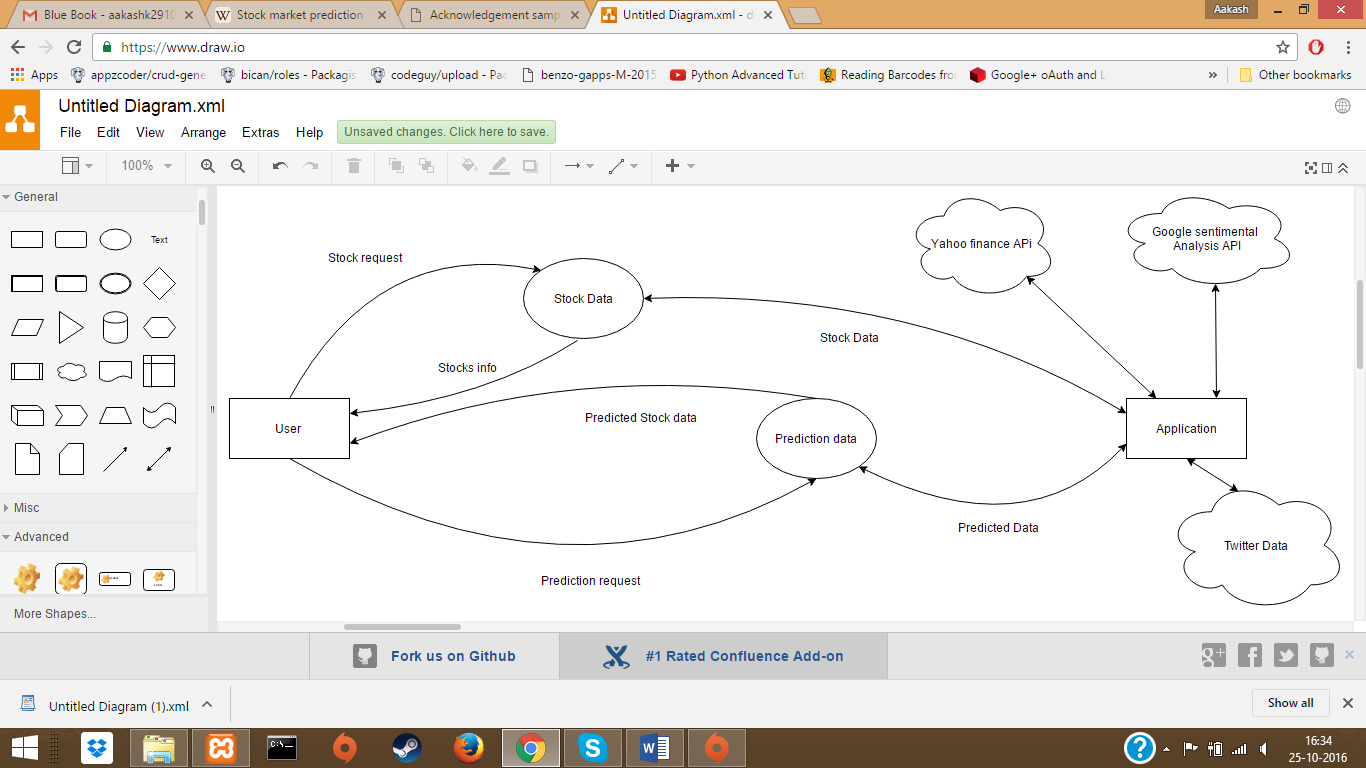


Figure 2.2 Data Flow Diagram (layer 1)

Figure 2.3 Core mechanism

This design can logically be seen as harmonic working of three blocks, first is the twitter data, second is sentimental analysis model and third is the stock data in form of DJIA. The twitter data in form of raw text from twitts is collected feed into sentimental analysis model. The output is twitter data with its’ polarity score. The relationship between the stock and data with polarity score is obtained. The stock data and the twitter data is plotted and the predicted output is obtained.

* 1. **Methodology**

Figure 2.4**:** Process Flow

1. **Sentiment Analysis Model:**

Figure 2.5**:** Classification Engine of proposed Sentiment Analysis Model

The four steps of the Twitter tweets classification [13].

* 1. **Global heuristics:**

Emoticons are a strong indications of sentiment. They also come in a variety of orthographic forms which require careful handling by defining methods. All non-positive sentiments present in the tweet followed by smiley overrides negative sentiment is among the most influential sentiment in the tweets. These is the main reason of registering the emoticons at the end.

* 1. **Evaluation of hashtags:**

Hashtags hold a special interest in twitter tweets due to the fact that they responsible for singling out a semantic unit of specific significance in the tweet. Big issue in exploiting semantic in hashtags is that it can conflate several terms present in a tweet, as in #greatthing or #notverysurprising. Appling a series of heuristics pairing parts of the hashtag with lexicons. In the case of #notverysurprising, the initial letters ‘not’ would be classified as one of the terms in the lexicon for non-positive terms. In like manner, the letters ‘very’ will be classified as one of the terms present in the lexicon for “strength of sentiment” ‘surprising’ will be identified as positive sentiment in the lexicon. Culmination of ‘not very surprising’ will lead to a detection of a negative sentiment for this particular hashtag. This is recorded and would be combined with determination of other features of the tweet at a later stage.

* 1. **Decomposition in Ngrams:**

The text present in a tweet segregated into list of uni-grams, bi-grams, tri-grams and quadri-grams. For example, the tweet ‘This job leaves to be needed’ will be segregated in list of the following: “This, job, leaves, to, be, needed, This job, job leaves, leaves to, to be, be needed, This job leaves, job leaves to, leaves to be, to be needed, This job leaves to, job leaves to be, leaves to be needed” The reason of this segregation is that sentiment are contained in emotions consisting of various terms. In the example above, to be needed is a negative detection recorded as such in the lexicon for negative sentiment, while needed is a positive sentiment. The model loops through all the n-grams of the tweet and verifies for their existence in lexicons. If an n-gram is found to be present in any of the lexicons, the heuristic present with this term in this lexicon is executed, and a classification (positive, negative, or other semantic) is returned.

* 1. **Post-processing:**

By this stage, the methods described above would have returned a high number of conflicting sentiment group for one single tweet. For instance, in the example.

*“This job leaves to be needed, the evaluation of the n-grams returns a positive sentiment classification (needed) and negative (to be needed) as well.”*

A sequence of heuristics verifies which of the conflicts indicated in sentiments should be retained till the end. In the above case, co-presence of positive and non-positive sentiments without indication is resolved as the tweet being of a non-positive sentiment. If the presence of a moderator is detected in the tweet (such as though, but, even if), rules of a complex nature are utilized.

* 1. **Prediction Algorithm**

**inputs :** stockItem, stockSentiment

**local variables:** predictedPrice, priceChange, var, count

priceChange 🡨 0

count 🡨 0

**while :** count < 5

var 🡨 stockSentiment.getPositive() – stockSentiment.getNegative()

priceChange 🡨 priceChange + (var \* 100) / stockSentiment.getTotalTweets()

count ++

**end**

priceChange 🡨 priceChange/5;

predictedPrice 🡨 stockitem.currentPrice + priceChange;

**return** predictedPrice;

* 1. **Summary**

Moods and previous days’ Dow Jones Industrial Average (DJIA) values are used to predict future stock movements and then use the predicted values in our portfolio management strategy results show a remarkable accuracy of nearly 87% in predicting the up and down changes in the closing values of Dow Jones Industrial Index (DJIA). App introduces a new way of collaborating both neural networks and decision tree to forecast the stock market price more accurately than MLP.

**Chapter 3**

**Analysis and Planning**

* 1. **Introduction**

Planning and Analysis, provides basic process and analytic frameworks to support Corporate/Division/Business Unit managers in both their day-to-day management tasks and their quarterly/annual financial management responsibilities. Decision is madeof how best to manage the process. This concentration was particularly relevant for us to plan on working within the finance and accounting function of our project. We have worked with strategic planning, budget, and forecast processes, preparing for monthly/quarterly/annual strategic and operational review processes, and create day-to-day decision analysis support, pricing, cost control, outsourcing, and capital spending.

* 1. **Feasibility Study**

1. **Executive Summary**

Project is related to ‘Stock Prediction using Sentimental Analysis Model’.

1. The project maintains two levels of users:-

Administrator Level-Owner

Users

1. Main facilities available in this project are:-
2. Maintaining records of users.
3. Maintaining stocks details.
4. Providing various stock analysis.
5. Providing specific stock details to users.
6. Maintaining backup of data as per user requirements (for selected stocks).
7. User or Administrator can search a stock record by name or rank.
8. **Define business problem or opportunity**

The Software is for the automation of Stock Prediction.

1. It maintains two levels of users:-
2. Administrator Level
3. User Level
4. The Software includes:-
5. Maintaining user details.
6. Providing and maintaining all kinds of stock analysis for a user.
7. Prediction and Report generation.
8. **Requirements and purpose of the study**
9. Hardware Requirements
   * + 1. Processor RAM, Disk Space
       2. Server
       3. Hosting
       4. Android Mobile
10. Software Requirements
11. Operating System
12. Android Studio
13. Android Libraries
14. Emulator
15. Twitter API
16. Stock API
17. Google Sentimental Analysis API
18. Purpose

App Predictor provide following facilities like:-

* + - 1. Analysis of stocks
      2. Stock Prediction
      3. Record maintenance of stocks

1. **Description of the options assessed**

Based on research conducted over the Internet we identified the following alternatives for the SPS:

1. Use Traditional Softwares.
2. Involve Experts to predict the stocks and analysis
3. **Assumptions used in the study**

Every user should be:

1. Comfortable of working with stocks.
2. He must have knowledge in stock transfers.
3. He must also have basic knowledge of English too.
4. **Audience impacted**

It can be used by any person be it an expert or a laymen dealing with or who wish to deal with stocks.

1. **Financial obligation**

The financial and the economic questions during the preliminary investigation are verified to estimate the following:

1. The cost to conduct a full system investigation.
2. The cost of hardware and software for the class of application being considered.
3. The benefits in the form of reduced cost.
4. The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide profits.
5. **Recommended action**
6. Time evaluation is the most important consideration in the development of project.
7. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.
8. A reliable Stock Prediction System can be developed in the considerable amount of time.
   1. **Project Planning**

Table of Contents

1. Project Name
2. Stakeholders
3. Project Description
4. Measurable Organization Value
5. Project Scope
6. Project Schedule Summery
7. Project Budget
8. Resources
9. Assumption Risk
10. Project Administration
11. Acceptance & Approval

Project Name:

Stock Prediction System

Stakeholders:

The stakeholders include follows:-

1. Project sponsor, project leader, & project team.
2. Quality assurance department
3. E & Y Consultants for internal Audits.
4. Customers

**Project Description**

The main purpose of the Stock Prediction System is to provide a fully automated environment to analyse stocks and predict the stock market in near future. Stock Prediction System allows to track multiple stocks and analyse in an efficient manner without wastage of time to check each one manually. Also a thorough analysis of stocks is ensured.

It is user-friendly and provides many options for the users. Tracking can be scheduled as per the convenience of users.

**Measurable Organizational Value:-**

| **Organizational**  **Impact** | **Value** | **Metric** | **Time Frame** |
| --- | --- | --- | --- |
| Strategic | Increase reliability and the image of Stock Predictor | Reduce load time and analysis time from 35% to 20%  Reduce schedule changes from 40% to 20%. | 24 months  12 months |
| Customer | Reduce the steps to complete the task.  Provide better analysis report.  Provide desired prediction | Reduce time to load.  Reduce the average wait time for report generation.  Increase availability of matching stock data from 40% to 60% | 12 months  12 months  12 months |
| Financial | Reduce personnel & communication cost of scheduling.  Reduce total load.  Reduce bandwidth wastage | Reduce personnel &communication cost of development by 73%.  Reduce bandwidth wastage or bots by 90% | 12 months  60 months  24 months |
| Operational | Reduce the scheduling cycle time.  Recognize specific delivery boys for their contribution. | Reduce scheduling time from 3 hours to 1 hour.  Increase the accuracy of contribution recognition program to 95% | 12 months  24 months |
| Social | Increase the number of customers served. | Increase the number of customer served from 100 to 150 | 24 months |

Table I: Technical Feasibility

**Project Scope**

Following would be included in the scope:-

1. Easy & safe way for customer’s registration & verification.
2. Provide better prediction system.
3. Increase the proper utilization of available data.
4. Reduce the maintenance cost of operations.
5. Reduce deviation from actual stock rates.

Project Schedule Summary

1. Project Start date: -
   1. July 20 2016
2. Project End date: -
   1. March 18 2017
3. Timeline of Project:-

The entire project is divided into modules as per project development lifecycle and spans over 36 days of duration.

**Project Budget**

1. Total Project Budget:-

The startup investment for AeroPredict is Rs 5,000. This includes the development, implementation and testing.

**Resources**

People:-

1. 1 Project Manager (project duration)—plan, manage, control project
2. 1 Project Administrator (project duration)—facilitate project planning, execution, and communication
3. 1 Programmers—design and develop UMS according to the requirements
4. 1 Testers – for the purpose of testing individual modules

Facilities:-

1. Equipments- Hardware and software.
2. Resources:-
   1. Computers
   2. Laptops
   3. Web Server
   4. Hosting
   5. DB Server

Name of the Provider:-

GoDaddy Online Hosting.

Date to be provided:-

1st August, 2016.

# **Assumptions and Risk**

Assumptions:-

1. Management will ensure that project team members are available as needed to complete project tasks and objectives.
2. The members are willing to change business operations to take advantage of the functionality offered by the new improved technology.
3. The core committee will participate in the timely execution of the Project Plan (i.e., timely approval cycles and meeting when required).
4. Failure to identify changes to draft deliverables within the time specified in the project timeline will result in project delays.
5. Project team members will adhere to the Communications Plan.

Constraints:-

The following represent known project constraints:

1. Project funding sources are limited, with no contingency.
2. Due to the nature of law enforcement, resource availability is inconsistent

Risks:-

AeroPredict Commitments- Accurate identification of AeroPredict requirements, interpretation and design depend on the patience and commitment of the people and staff in focus.

**Project Administration**

1. Communications Plan:-

The communication among the team members as well as the client and developers is very important. Proper communication is a must for effective development of the project. The Project manager is at the apex of the communication tree. He should effectively communicate with the Project administrator who in turn communicates with the programmers as well as the testers. If effective communication fails, the end result would be a total chaos.

1. Scope Management Plan: -

The scope of the software project is described above. The Project Administrator takes appropriate care that the project doesn’t go out of the scope so that the timeframe does not gets disturbed as it would cause issues. It is an important to see that nothing outside of the scope be touched upon.

1. Quality Management Plan: -

Quality management can be considered to have four main components: quality planning, quality control, quality assurance and quality improvement. Quality management is focused not only on product/service quality, but also the means to achieve it. Again it is up to the Project Administrator to maintain high quality so that it maintains the customer satisfaction.

1. Change Management Plan: -

Changes in any software project are an integral part of it. The changes are caused due to the change in requirements of the Users. Whenever any change is arrived the project administrator understands the C.R. first and will set priority to the same. Next he would assign the change request to any of the developer who after developing the same would give it to the tester. Just in case the changes are too many then project manager would coordinate with the users.

* 1. **Scheduling (Timeline Chart)**

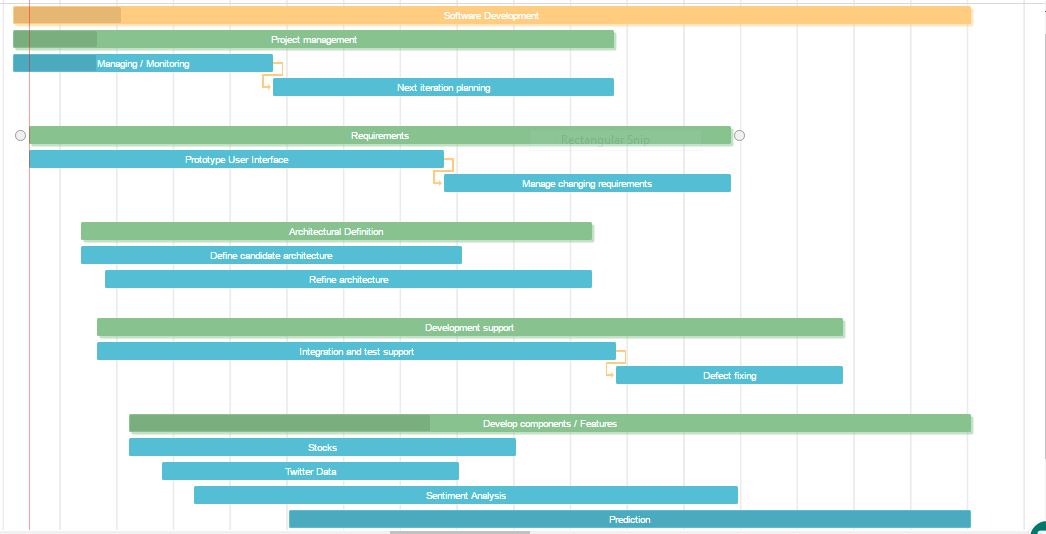
****

Figure 3.1: Gantt Chart

* 1. **Summary**

Investigation is done on the causative relation between public mood as measured from a large scale collection of tweets from twitter.com and the DJIA values. Our results 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. The planning and analysis of project is done accordingly.

The Project completion is represented graphically using Gantt chart. The progress mapping is done. The monitoring is done and it is observed that the project is almost on schedule and the schedule broken the buffer time allocated for the stages.

**Chapter 4**

**Results & Discussion**

**4.1 Expected Results**

Our work is based on Bollen et al’s famous paper which predicted the same with 87% accuracy. They also used causality analysis to investigate the hypothesis that public mood states, as measured by the OpinionFinder and GPOMS mood time series, are predictive of changes in DJIA closing values. The authors used Self Organizing Fuzzy Neural Networks to predict DJIA values using previous values. Their results show a remarkable accuracy of nearly 87% in predicting the up and down changes in the closing values of Dow Jones Industrial Index (DJIA).

As sentiment API provides improved results we are expecting to improve the accuracy of the prediction.

We are also expecting that the project will have a huge market value due to SaaS Architecture. The application will work over cloud and will provide a multi-tenant environment.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. no** | **Entity** | **Numbers of tweets** | **Predicted Stock price** | **Actual Stock price** | **Deviation** | **Percentage Deviation** |
| 1 | AlphabetInc**.** | 675 | 809.44 | 815.91 | 6.47 | 0.795 |
| 2 | Facebook Inc. | 900 | 128.42 | 128.74 | 0.32 | 0.249 |
| 3 | Yahoo Inc. | 900 | 42.08 | 42.26 | 0.18 | 0.02 |
| 4 | Apple Inc. | 898 | 120.22 | 120.03 | 0.19 | 0.158 |
| 5 | Nike | 1000 | 56 | 55.9 | 0.10 | 0.178 |
| 6 | Adidas | 1000 | 177.5 | 177.30 | 0.20 | 0.113 |
| 7 | Puma | 1000 | 343.5 | 343.85 | 0.35 | 0.102 |

Table III : Result Analysis

The comparison is done among companies. Prediction of stock prices is done using the proposed algorithm. The reading of Actual price is taken next day. The average deviation is 1.859 and mean present deviation is 0.305%. It has been observed that in case of analysis data with low number of tweets the deviation is found to be more than that observed otherwise. With increase in the dataset and available of more structured stock data the accuracy will increase and proportionally the deviation will decrease accordingly.

The correlation between the sentiment and the change in price is found to be high. With the accuracy being 100% for the considered entities.

|  |  |  |
| --- | --- | --- |
| **Output** | **Outcome** | **Improvement** |
| Efficient Market Hypothesis | Predict random walkthrough of market prices. | This techniques improved the prediction of the market changes. Percent improvement of **10%** was observed. |
| Semantic Classification | Classification of incoming to be analysed into semantic classes. | Semantic classes help to provide better classification. The **19%** improvement of prediction results was observed. |
| Pre-processing of Data | The data was pre-processed and rather than data information was used for analysis. | A drastic increase in the percent improvement of about **69%** was observed |
| Multi-level sentiment Analysis model | A multilayer sentiment analysis model enabled efficient and quality sentimental output | Percentage improvement of about **73.2%** was observed as compared to regular sentiment analysis. |

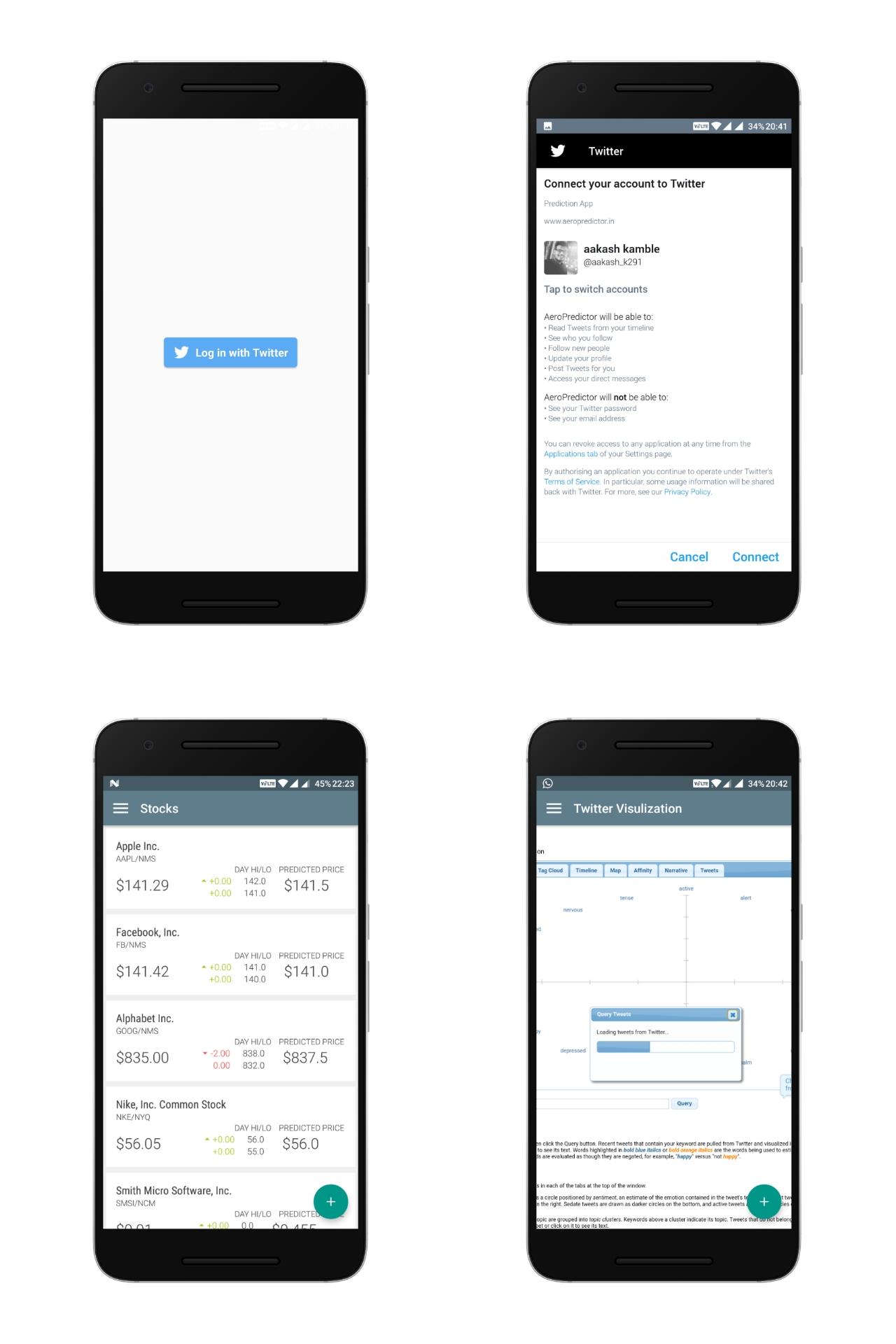
Table IV: Output, Outcome mapping table

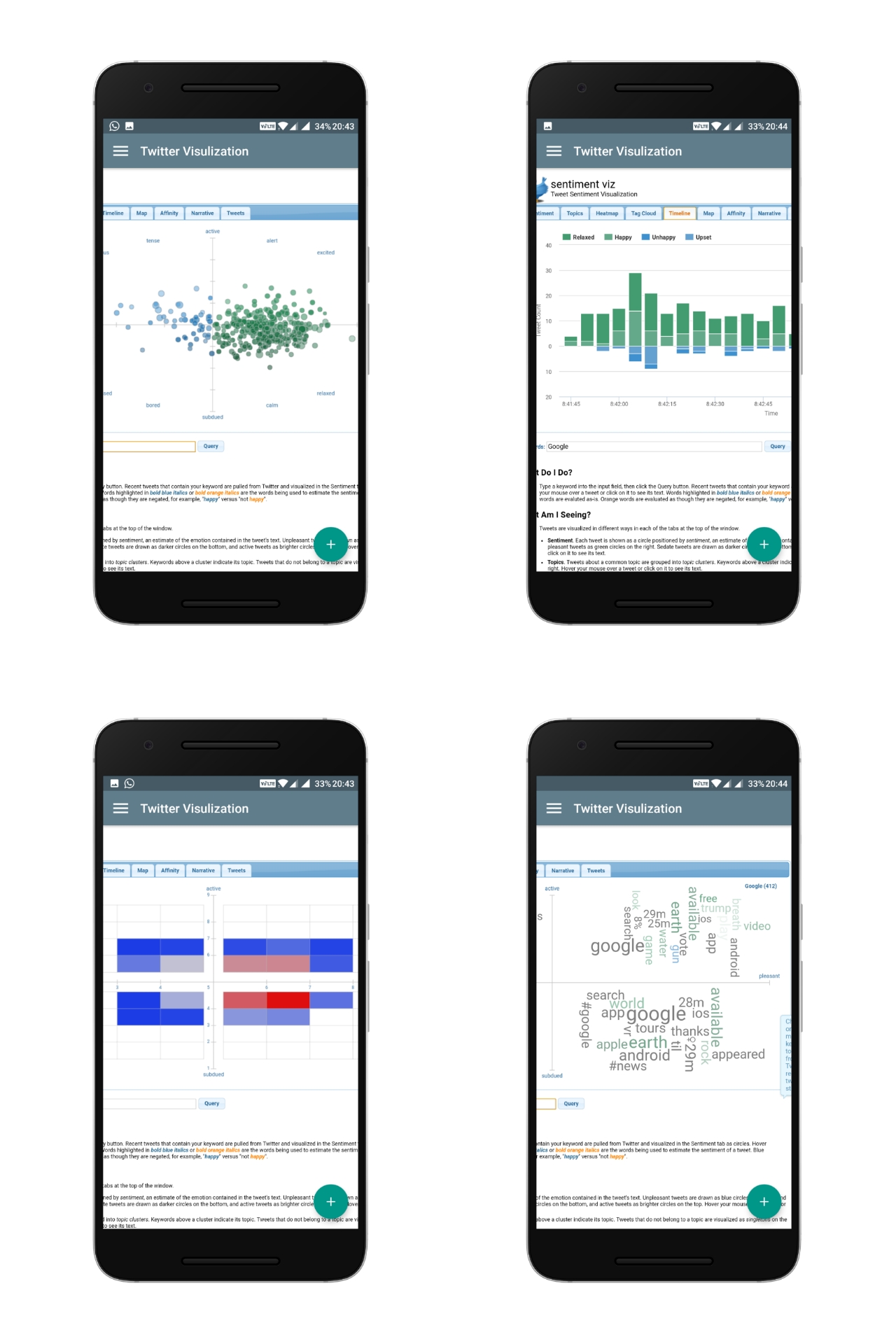
* 1. **Future Scope**

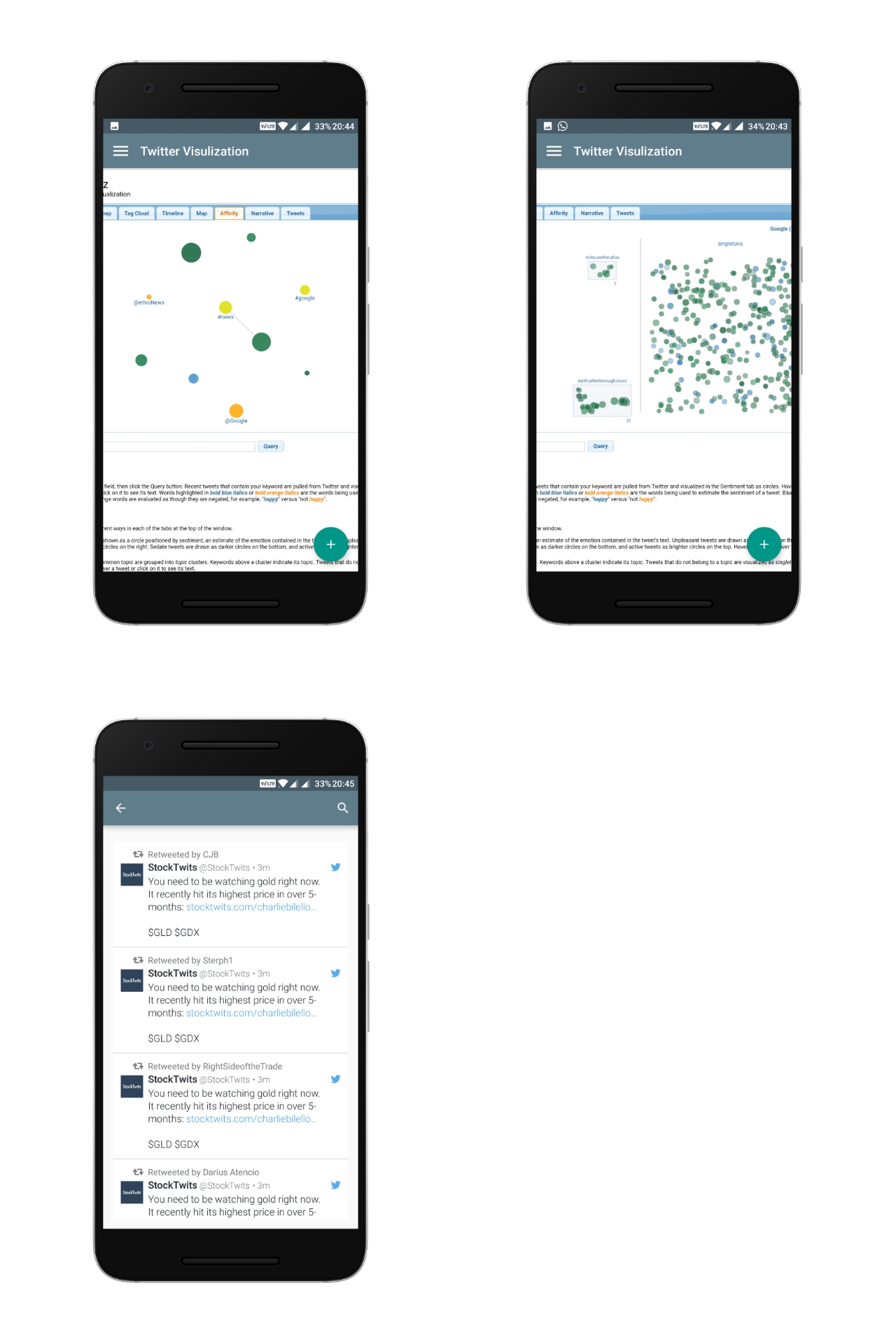
The neural network models for forecasting stock market are at an evolutionary stage and there are future possibilities of improvement in the prediction accuracy and reliability of the neural network based forecasting.

In this software we have investigated the basic aspect of the prediction problem of a stock market with artificial neural network. We have only used the Historic prices of the Index values for prediction. Other macro-economic factors and other international stock market data as input variables can also be used as input variables in order to improve the accuracy of the model. Application of Chaos Theory, Fractal analysis and wavelet analysis in feature selection of the input data set will also give a possibility of improvement in the performance. The various trend indicators of the technical analysis can also be used in the input variables and can be checked for improvement in the performance of the network

We would like to extend this research by adding more company’s data and check the prediction accuracy. For those companies where availability of twitter data is a challenge, we would be using yahoo finance news data for similar analysis. We can also incorporate similar strategies for algorithmic trading.

****

****

****

**Chapter 5**

**Conclusion**

**5.1 Conclusion**

Our results are in some conjunction with , but there are some major differences as well. Firstly our results show a better correlation between the calm and happy mood dimensions with the DJIA values, unlike their result, which showed high correlation with only calm mood dimension

It’s worth mentioning that our analysis doesn’t take into account many factors. Firstly, our dataset doesn’t really map the real public sentiment, it only considers the twitter 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 affect their investment decisions, hence the correlation.

The Android Application has provided us a platform to reach wide range of audience and a user friendly environment for users. The application has a user-friendly User Interface and a simple workflow to add/delete stocks and maintain portfolio. The prediction algorithm is simple with a Time complexity of O(1) and Space complexity of O(1). The algorithm natively calculates the predicted price and help users suggest the likely rise and fall of the stock prices.

**REFERENCES**

[1] J. Bollen and H. Mao., “Twitter mood as a stock market predictor,” *IEEE Computer, vol. , no. 44(10):91–94*

[2] Anshul Mittal and Arpit Goel Stanford University., “Stock Prediction Using Twitter Sentiment Analysis,” *https://pdfs.semanticscholar.org/4ecc/55e1c3ff1cee41f21e5b0a3b22c58d04c9d6.pdf*

[3] Anurag Nagar, Michael Hahsler, “Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams,”  *IPCSIT vol., no. XX (2012) IACSIT Press, Singapore*

[4] W.B. Yu, B.R. Lea, and B. Guruswamy, “A Theoretic Framework Integrating Text Mining and Energy Demand Forecasting,” *International Journal of Electronic Business Management, vol., no. 2011,5(3): 211-224,*

[5] J. Bean, “R by example: Mining Twitter for consumer attitudes towards airlines,” *In Boston Predictive Analytics Meetup Presentation, Feb 2011*

[6] Yauheniya Shynkevich, “T.M. McGinnity, Sonya Coleman, Ammar Belatreche, Predicting Stock Price Movements Based on Different Categories of News Articles,” *2015 IEEE Symposium Series on Computational Intelligence*

[7] A. Lapedes and R. Farber, "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling", *Technical Report, LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987.*

[8] Kim, K, J and I Han (2000), “Genetic algorithm approach to feature discretization in artificial neural network for the prediction of stock price index.” Published by *Elsevier science, Ltd., Experts systems with application, 19, 125-132.*

[9] Roman, Jovina and Jameel, Akhtar Backpropagation and Recurrent Neural Networks in *Financial Analysis of Multiple Stock Market Returns.*

[10] Panda, C. and Narasimhan, V. (2006), “Predicting Stock Returns : An Experiment of the Artificial Neural Network”, *Indian Stock Market South Asia Economic Journal, Vol. 7, No. 2, 205-218.*

[11] Goutam Dutta, Pankaj Jha, Arnab Kumar Laha and Neeraj Mohan, “Artificial Neural Network Models for Forecasting Stock Price Index”, *The Bombay Stock Exchange Journal of Emerging Market Finance, Vol. 5, No. 3, 283-295.*

[12] Malkiel, Burton G. (2003). "The Efficient Market Hypothesis and Its Critics”. *CEPS Working paper No 91.*

[13] Clement Levallois, Umigon. Retrieved from *http://www.clementlevallois.net/download/umigon.pdf*

[14] T. Rao and S. Srivastava, "TweetSmart: Hedging in markets through Twitter," 2012 Third International Conference on Emerging Applications of Information Technology, Kolkata, 2012, pp. 193-196. doi: 10.1109/EAIT.2012.6407894

**Appendix**

**StockItem.java**

*/\*\*  
 \* Custom stock object POJO  
 \*/***public class** StockItem {  
  
 **private** String **name**;  
 **private** String **currency**;  
 **private** String **symbol**;  
 **private** String **stockExchange**;  
 **private double price**;  
 **private long avgVolume**;  
 **private long volume**;  
 **private double dayHigh**;  
 **private double dayLow**;  
 **private double yearHigh**;  
 **private double yearLow**;  
 **private double open**;  
 **private double previousClose**;  
 **private double marketCapitalisation**;  
 **private double earningsPerShare**;  
 **private double change**;  
 **private double changeInPercent**;  
 **private double annualYield**;  
 **private double annualYieldPercentage**;  
  
  
 **public** StockItem() {}  
  
 **public** StockItem(String name,  
 String currency,  
 String symbol,  
 String stockExchange,  
 **double** price,  
 **long** avgVolume,  
 **long** volume,  
 **double** dayHigh,  
 **double** dayLow,  
 **double** yearHigh,  
 **double** yearLow,  
 **double** open,  
 **double** previousClose,  
 **double** marketCapitalisation,  
 **double** earningsPerShare,  
 **double** change,  
 **double** changeInPercent,  
 **double** annualYield,  
 **double** annualYieldPercentage) {  
  
 **this**.**name** = name;  
 **this**.**currency** = currency;  
 **this**.**symbol** = symbol;  
 **this**.**stockExchange** = stockExchange;  
 **this**.**price** = price;  
 **this**.**avgVolume** = avgVolume;  
 **this**.**volume** = volume;  
 **this**.**dayHigh** = dayHigh;  
 **this**.**dayLow** = dayLow;  
 **this**.**yearHigh** = yearHigh;  
 **this**.**yearLow** = yearLow;  
 **this**.**open** = open;  
 **this**.**previousClose** = previousClose;  
 **this**.**marketCapitalisation** = marketCapitalisation;  
 **this**.**earningsPerShare** = earningsPerShare;  
 **this**.**change** = change;  
 **this**.**changeInPercent** = changeInPercent;  
 **this**.**annualYield** = annualYield;  
 **this**.**annualYieldPercentage** = annualYieldPercentage;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getCurrency() {  
 **return currency**;  
 }  
  
 **public void** setCurrency(String currency) {  
 **this**.**currency** = currency;  
 }  
  
 **public** String getSymbol() {  
 **return symbol**;  
 }  
  
 **public void** setSymbol(String symbol) {  
 **this**.**symbol** = symbol;  
 }  
  
 **public** String getStockExchange() {  
 **return stockExchange**;  
 }  
  
 **public void** setStockExchange(String stockExchange) {  
 **this**.**stockExchange** = stockExchange;  
 }  
  
 **public double** getPrice() {  
 **return price**;  
 }  
  
 **public void** setPrice(**double** price) {  
 **this**.**price** = price;  
 }  
  
 **public long** getAvgVolume() {  
 **return avgVolume**;  
 }  
  
 **public void** setAvgVolume(**long** avgVolume) {  
 **this**.**avgVolume** = avgVolume;  
 }  
  
 **public long** getVolume() {  
 **return volume**;  
 }  
  
 **public void** setVolume(**long** volume) {  
 **this**.**volume** = volume;  
 }  
  
 **public double** getDayHigh() {  
 **return dayHigh**;  
 }  
  
 **public void** setDayHigh(**double** dayHigh) {  
 **this**.**dayHigh** = dayHigh;  
 }  
  
 **public double** getDayLow() {  
 **return dayLow**;  
 }  
  
 **public void** setDayLow(**double** dayLow) {  
 **this**.**dayLow** = dayLow;  
 }  
  
 **public double** getYearHigh() {  
 **return yearHigh**;  
 }  
  
 **public void** setYearHigh(**double** yearHigh) {  
 **this**.**yearHigh** = yearHigh;  
 }  
  
 **public double** getYearLow() {  
 **return yearLow**;  
 }  
  
 **public void** setYearLow(**double** yearLow) {  
 **this**.**yearLow** = yearLow;  
 }  
  
 **public double** getOpen() {  
 **return open**;  
 }  
  
 **public void** setOpen(**double** open) {  
 **this**.**open** = open;  
 }  
  
 **public double** getPreviousClose() {  
 **return previousClose**;  
 }  
  
 **public void** setPreviousClose(**double** previousClose) {  
 **this**.**previousClose** = previousClose;  
 }  
  
 **public double** getMarketCapitalisation() {  
 **return marketCapitalisation**;  
 }  
  
 **public void** setMarketCapitalisation(**double** marketCapitalisation) {  
 **this**.**marketCapitalisation** = marketCapitalisation;  
 }  
  
 **public double** getEarningsPerShare() {  
 **return earningsPerShare**;  
 }  
  
 **public void** setEarningsPerShare(**double** earningsPerShare) {  
 **this**.**earningsPerShare** = earningsPerShare;  
 }  
  
 **public double** getChange() {  
 **return change**;  
 }  
  
 **public void** setChange(**double** change) {  
 **this**.**change** = change;  
 }  
  
 **public double** getChangeInPercent() {  
 **return changeInPercent**;  
 }  
  
 **public void** setChangeInPercent(**double** changeInPercent) {  
 **this**.**changeInPercent** = changeInPercent;  
 }  
  
 **public double** getAnnualYield() {  
 **return annualYield**;  
 }  
  
 **public void** setAnnualYield(**double** annualYield) {  
 **this**.**annualYield** = annualYield;  
 }  
  
 **public double** getAnnualYieldPercentage() {  
 **return annualYieldPercentage**;  
 }  
  
 **public void** setAnnualYieldPercentage(**double** annualYieldPercentage) {  
 **this**.**annualYieldPercentage** = annualYieldPercentage;  
 }  
  
  
}

**Suggestion:**

**Resultset.java**

**import** java.util.List;  
  
**public class** ResultSet {  
  
 List<StockSymbol> **Result**;  
  
 **public** List<StockSymbol> getResult() {  
 **return Result**;  
 }  
  
}

**StockSyambol.java**

**public class** StockSymbol {  
  
 **private** String **symbol**;  
 **private** String **name**;  
 **private** String **exch**;  
 **private** String **type**;  
 **private** String **exchDisp**;  
 **private** String **typeDisp**;  
  
 **public** StockSymbol() { }  
  
 **public** StockSymbol(String symbol, String name, String exch, String type, String exchDisp, String typeDisp) {  
 **this**.**symbol** = symbol;  
 **this**.**name** = name;  
 **this**.**exch** = exch;  
 **this**.**type** = type;  
 **this**.**exchDisp** = exchDisp;  
 **this**.**typeDisp** = typeDisp;  
 }  
  
 **public** String getSymbol() {  
 **return symbol**;  
 }  
  
 **public void** setSymbol(String symbol) {  
 **this**.**symbol** = symbol;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getExch() {  
 **return exch**;  
 }  
  
 **public void** setExch(String exch) {  
 **this**.**exch** = exch;  
 }  
  
 **public** String getType() {  
 **return type**;  
 }  
  
 **public void** setType(String type) {  
 **this**.**type** = type;  
 }  
  
 **public** String getExchDisp() {  
 **return exchDisp**;  
 }  
  
 **public void** setExchDisp(String exchDisp) {  
 **this**.**exchDisp** = exchDisp;  
 }  
  
 **public** String getTypeDisp() {  
 **return typeDisp**;  
 }  
  
 **public void** setTypeDisp(String typeDisp) {  
 **this**.**typeDisp** = typeDisp;  
 }  
  
 @Override  
 **public** String toString() {  
 **return** String.*format*(**"Symbol: %s, name: %s"**, getSymbol(), getName());  
 }  
}

**SuggestionQuery.java**

**package** com.aerotron.aeropredictor.model.suggestion;  
  
  
*/\*\*  
 \* http://autoc.finance.yahoo.com/autoc?query=ab&callback=YAHOO.Finance.SymbolSuggest.ssCallback&region=US&lang=en-US  
 \*  
 {  
 "ResultSet":{  
 "Query":"ab",  
 "Result":[  
 {  
 "symbol":"AB",  
 "name":"AllianceBernstein Holding L.P.",  
 "exch":"NYQ",  
 "type":"S",  
 "exchDisp":"NYSE",  
 "typeDisp":"Equity"  
 },  
 {  
 "symbol":"ABC",  
 "name":"AmerisourceBergen Corporation",  
 "exch":"NYQ",  
 "type":"S",  
 "exchDisp":"NYSE",  
 "typeDisp":"Equity"  
 },  
 {  
 "symbol":"ABX",  
 "name":"Barrick Gold Corporation",  
 "exch":"NYQ",  
 "type":"S",  
 "exchDisp":"NYSE",  
 "typeDisp":"Equity"  
 },  
 {  
 "symbol":"ABBV",  
 "name":"AbbVie Inc.",  
 "exch":"NYQ",  
 "type":"S",  
 "exchDisp":"NYSE",  
 "typeDisp":"Equity"  
 },  
 {  
 "symbol":"ABT",  
 "name":"Abbott Laboratories",  
 "exch":"NYQ",  
 "type":"S",  
 "exchDisp":"NYSE",  
 "typeDisp":"Equity"  
 },  
  
 {  
 "symbol":"ABG.MC",  
 "name":"ABENGOA -A-",  
 "exch":"MCE",  
 "type":"S",  
 "exchDisp":"Madrid StockItem Exchange CATS",  
 "typeDisp":"Equity"  
 }  
 ]  
 }  
 }  
  
\*/***public class** SuggestionQuery {  
  
 **private** com.aerotron.aeropredictor.model.suggestion.ResultSet **ResultSet**;  
  
 **public** ResultSet getResultSet() {  
 **return ResultSet**;  
 }  
  
}

**Network:**

**GetStockQuoteThread.java**

**import** android.content.Context;  
**import** android.content.SharedPreferences;  
**import** android.os.Process;  
**import** android.preference.PreferenceManager;  
  
**import** com.aerotron.aeropredictor.event.AppMessageEvent;  
**import** com.aerotron.aeropredictor.model.StockItem;  
**import** com.aerotron.aeropredictor.model.data.StockDataCache;  
**import** com.aerotron.aeropredictor.util.Constants;  
  
**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.HashSet;  
**import** java.util.List;  
**import** java.util.Map;  
**import** java.util.Set;  
**import** java.util.TreeSet;  
  
**import** de.greenrobot.event.EventBus;  
**import** timber.log.Timber;  
**import** yahoofinance.Stock;  
**import** yahoofinance.YahooFinance;  
  
*/\*\*  
 \* References:  
 \* [1] http://financequotes-api.com/  
 \* [2] http://financequotes-api.com/javadoc/yahoofinance/YahooFinance.html  
 \*/***public class** GetStockQuoteThread **extends** Thread{  
  
 **private** Context **mContext**;  
 **private** SharedPreferences **mPrefs**;  
  
 **public** GetStockQuoteThread(Context context) {  
 **mContext** = context.getApplicationContext();  
 }  
  
 @Override  
 **public void** run() {  
 **super**.run();  
 android.os.Process.*setThreadPriority*(Process.***THREAD\_PRIORITY\_BACKGROUND***);  
  
 **if**(!isInterrupted()) {  
  
 **synchronized** (**this**) {  
 **mPrefs** = PreferenceManager.*getDefaultSharedPreferences*(**mContext**);  
 }  
  
 List<StockItem> stockItemList = **new** ArrayList<>();  
 StockItem stockItem;  
  
 *// empty array list should nothing be found in shared preferences* Set<String> defaultPortfolio = **new** HashSet<>();  
 defaultPortfolio.addAll(**new** ArrayList<String>());  
  
 *// fetch user preferences, otherwise pass in the default* Set<String> temp = **mPrefs**.getStringSet(Constants.***PREFS\_STOCK\_PORTFOLIO\_SET***, defaultPortfolio);  
 Set<String> portfolio = **new** TreeSet<>(temp); *// sort* String[] symbols = portfolio.toArray(**new** String[portfolio.size()]);  
  
 **if** (symbols.**length** == 0) {  
 *// if the returned string set is empty, post a message to the user* EventBus.*getDefault*().post(**new** AppMessageEvent(Constants.***STOCK\_PORTFOLIO\_NOT\_DEFINED***));  
 **return**;  
 }  
  
 **try** {  
 *// query the Yahoo Finance API* Map<String, Stock> stocks = YahooFinance.*get*(symbols);  
 Stock stock;  
  
 **if** (stocks != **null**) {  
 **for** (String symbol : symbols) {  
 stock = stocks.get(symbol);  
 stockItem = buildCustomStockItem(stock);  
 stockItemList.add(stockItem);  
 }  
  
 *// stash the data to the cache, let any interested parties know* StockDataCache.*getStockDataCache*().setStocks(stockItemList);  
 EventBus.*getDefault*().post(**new** AppMessageEvent(Constants.***STOCK\_DOWNLOAD\_COMPLETE***));  
  
 } **else** EventBus.*getDefault*().post(**new** AppMessageEvent(Constants.***STOCK\_DOWNLOAD\_FAILED***));  
  
 } **catch** (IOException e) {  
 Timber.*e*(e, **"Failed to retrieve quote data: %s"**, e.getMessage());  
 EventBus.*getDefault*().post(**new** AppMessageEvent(Constants.***STOCK\_DOWNLOAD\_FAILED***));  
 }  
 }  
 }  
  
  
 **private** StockItem buildCustomStockItem(Stock stock) {  
  
 *// convert BigDecimal to longs, Realm does NOT support BigDecimal (or java.util.Calendar)* **return new** StockItem(  
 stock.getName(),  
 stock.getCurrency(),  
 stock.getSymbol(),  
 stock.getStockExchange(),  
 stock.getQuote().getPrice().doubleValue(),  
 stock.getQuote().getAvgVolume(),  
 stock.getQuote().getVolume(),  
 stock.getQuote().getDayHigh().longValue(),  
 stock.getQuote().getDayLow().longValue(),  
 stock.getQuote().getYearHigh().longValue(),  
 stock.getQuote().getYearLow().longValue(),  
 stock.getQuote().getOpen().longValue(),  
 stock.getQuote().getPreviousClose().longValue(),  
 stock.getStats().getMarketCap().longValue(),  
 stock.getStats().getEps().longValue(),  
 stock.getQuote().getChange().longValue(),  
 stock.getQuote().getChangeInPercent().longValue(),  
 stock.getDividend().getAnnualYield().longValue(),  
 stock.getDividend().getAnnualYieldPercent().longValue()  
 );  
  
 }  
  
}

**Prediction.java**

**public class** Prediction {  
  
 **private static** String *SEARCH\_QUERY*;  
 **private static final** String ***SEARCH\_RESULT\_TYPE*** = **"recent"**;  
 **private static final int *SEARCH\_COUNT*** = 1000;  
 **private long maxId**;  
 ListView **SearchList**;  
 **private** String **t**;  
 **private int positive** =0, **neagtive**=0;  
  
 **public** Double getPrediction(StockItem item){  
  
 **final** SearchService service = Twitter.*getApiClient*().getSearchService();  
 service.tweets(*SEARCH\_QUERY*, **null**, **null**, **null**, ***SEARCH\_RESULT\_TYPE***, ***SEARCH\_COUNT***, **null**, **null**, **maxId**, **true**, **new** Callback<Search>() {  
 @Override  
 **public void** success(Result<Search> searchResult) {  
  
 List<Tweet> tweets = searchResult.**data**.**tweets**;  
 **for** (Tweet i:  
 tweets) {  
 **t** = i.**text**;  
 HttpResponse r = getSentiment(**t**);  
  
 **if**(r.polarity >0){  
 **positive**++;  
 }  
 **else if**(r.polarity < 0) {  
 **neagtive**++;  
 }  
 }  
 }  
  
 @Override  
 **public void** failure(TwitterException error) {  
  
 **positive** = **neagtive** = 0;  
 }  
 }  
 );  
  
 **return** item.getPrice() + ((**positive** - **neagtive**)\*100.0 / 1000);  
 }  
  
 **public** HttpResponse getSentiment(String tweet){  
 *// These code snippets use an open-source library. http://unirest.io/java* HttpResponse<JsonNode> response = post(**"https://community-sentiment.p.mashape.com/text/"**)  
 .header(**"X-Mashape-Key"**, **"ipoeHPWlmfmshIqzTFZHcUm6eZy5p1yW6MDjsnFaYoNXRYyp3h"**)  
 .header(**"Content-Type"**, **"application/x-www-form-urlencoded"**)  
 .header(**"Accept"**, **"application/json"**)  
 .field(**"txt"**, tweet)  
 .asJson();  
  
 **return** response;  
 }  
  
  
}

**StockDataCache.java**

*/\*\*  
 \* Singleton used to hold the stock data  
 \* Reference:  
 \* [1] https://www.safaribooksonline.com/library/view/android-programming-the/9780134171517/ch07s03.html  
 \* [2] https://www.safaribooksonline.com/library/view/android-programming-the/9780134171517/ch09.html  
 \*/***public class** StockDataCache {  
  
 **public static final** String ***STOCK\_OBJECT*** = **"stock"**;  
 **private static** StockDataCache *sStockDataCache*;  
 **private static** List<StockItem> *sStockItems*;  
  
 **private** StockDataCache() { }  
  
 **public static** StockDataCache getStockDataCache() {  
 **if**(*sStockDataCache* == **null**) {  
 *sStockDataCache* = **new** StockDataCache();  
 *sStockItems* = **new** ArrayList<>();  
 }  
 **return** *sStockDataCache*;  
 }  
  
  
 **public** List<StockItem> getStocks() {  
 **return** *sStockItems*;  
 }  
  
 **public void** setStocks(List<StockItem> stockItems) {  
 *sStockItems*.clear();  
 *sStockItems*.addAll(stockItems);  
 }  
  
 **public** StockItem getStock(String symbol) {  
 **if**(symbol != **null**) {  
 **for** (StockItem stockItem : *sStockItems*) {  
 **if**(stockItem.getSymbol().equals(symbol)) {  
 **return** stockItem;  
 }  
 }  
 }  
 **return null**;  
 }  
  
}

**TwitterLogin.java**

**public class** Splash **extends** Activity {  
  
 **private** TwitterLoginButton **loginButton**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(com.aerotron.aeropredictor.R.layout.***splashh***);  
  
  
 **loginButton** = (TwitterLoginButton) findViewById(R.id.***twitter\_login\_button***);  
 **loginButton**.setCallback(**new** Callback<TwitterSession>() {  
 @Override  
 **public void** success(Result<TwitterSession> result) {  
 *// The TwitterSession is also available through:  
 // Twitter.getInstance().core.getSessionManager().getActiveSession()* TwitterSession session = result.**data**;  
 *//* ***TODO: Remove toast and use the TwitterSession's userID*** *// with your app's user model* String msg = **"@"** + session.getUserName() + **" logged in! (#"** + session.getUserId() + **")"**;  
 Toast.*makeText*(getApplicationContext(), msg, Toast.***LENGTH\_LONG***).show();  
 Intent intent = **new** Intent(getApplicationContext(), MainActivity.**class**);  
 startActivity(intent);  
  
 }  
 @Override  
 **public void** failure(TwitterException exception) {  
 Toast.*makeText*(getApplicationContext(), **"Unable to Login"**, Toast.***LENGTH\_LONG***).show();  
 Log.*d*(**"TwitterKit"**, **"Login with Twitter failure"**, exception);  
*/\* Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivity(intent);\*/* }  
 });  
  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 *// Make sure that the loginButton hears the result from any  
 // Activity that it triggered.* **loginButton**.onActivityResult(requestCode, resultCode, data);  
 }  
  
}