CHAPTER 1

1.1 GENERAL

In financial stock markets, it is considered to be an impossible task to predict most-likely stock market movement. There are generally two approaches for predicting the future market movements. One approach relates to the analysis of current sentiments related to a firm like that of Apple Inc. from social media and news. The other approach involves using Time Series Forecasting methods on previous historical data to find patterns in short-term seasonal intervals to predict the most likely bounded price change along with the returns for any number of days with very high accuracy and very low percentage errors.

1.2 SENTIMENT ANALYSIS APPROACH

In the first approach of the proposed model, big data from social media sites like Twitter is used for analysis. Positive and negative sentiments are identified for companies using Natural Language Processing through word dictionaries and n-grams model. When someone buys a stock, he/she generally buys it because of something he/she heard about on the news, social media or through their friends. This external information is used for stock market prediction. Based on this a correlation is found between Daily Price Change and sentiment score to predict next day's most likely movement.

Sentiment analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. It is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, company, stocks etc. is positive, negative, or neutral. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service. It generally aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. Linguistics is involved in this domain to preprocess data through cleaning, removing redundancy, converting encodings, textual analysis, etc., along with the extraction of knowledge in the sources like positive and negative opinions. It is also

referred to as opinion mining in the field of Natural Language Processing. Sentiments are defined to be awesome, very good, good, neutral, bad and poor/very negative feeling in this proposed project work.

1.2.1 Twitter Historical Data

Twitter feed is a social media platform used online for sharing recent news through the usage of “tweets”. More than a billion users are responsible for sending and reading of these “tweets”, which are short 140-character messages. Hence Twitter is a public platform with a mine of public opinion of people all over the world and of all age categories. In Twitter Sentiment Analysis, the emotional tone of people behind a series of word sets inside a particular tweet is determined, to gain an understanding of the feelings, sentiments, and emotions expressed through an online hashtag/mention/reference of the stocks related to technology giant Apple Inc., USA. Data is extracted from Twitter API. The tweets are collected using Twitter API and filtered by performing sentiment analysis using both Python and R programming. Twitter provides two possible ways to gather tweets, the Streaming API or the Search API. Tweets are classified as positive, negative and neutral based on the sentiment present.

Tweets are the best example of Unstructured Data with high velocity and variety. They generally consist of a lot of symbols, acronyms, hashtags, shorthand notations, emojis and emoticons, annotations, encodings and unnecessary information like images, videos and URL links. Some tweets could also be retweeted multiple times by others creating redundancy. So the tweets are preprocessed and filtered to represent correct public opinions, emotions, feelings, and sentiments. For tweet preprocessing, multiple levels of extraction like corpus formation, obtaining a stream of tokens, and stop words removal, and conversion to ASCII encodings for removing special characters were used to filter the right data. After this, the word corpus of tweets is ready. From this word corpus, frequent words used in the sentiments are filtered and show in the form of word cloud, which can be used for outlier and robot analysis along with the list of people responsible for these sentiments. For preprocessing of tweets, tokenization and stop words removal is done for removing special characters. In tokenization, tweets are split into individual words based on the space and irrelevant symbols like emoticons are removed. We form a

list of individual words for each tweet. Words that do not express any emotion are called Stop words. After splitting a tweet, words like a, is, the, with etc. are removed from the list of words.

After data preprocessing, tweets are labelled through “positive” or “negative” based on sentiment scores. The net sentiment of a tweet is sum of all word sentiments. For extracting and filtering these features, labelling is done on basis of N-grams model. After labelling the data and feature extraction, we build a sentiment analyzer that determines the label of the tweet. The sentiment analyzer is a binary classifier which classifies the tweet into positive or negative.We used two types of classifiers: Naive Bayes Bernoulli and Support Vector Machine.

1.3 TIME SERIES FORECASTING APPROACH

For the second approach, Historical stock market changes dataset is used for forecasting by assuming that certain patterns have bearings on the future for short-term linear intervals. It is similar to the weather forecasting approach. This is what the ARIMA (Auto Regressive Integrated Moving Average) prediction model for time series forecasting is based on and is famous for. The stationary time series was used to forecast the closing price and returns in logarithms based on training of historical data and combined with Neural Network to get an accuracy of 87%.

ARIMA and Neural Network are built on historical stocks in R-Studio. A large number of analysts and researchers make use of R language to take care of most difficult problems in the fields ranging from computational science and big data analytics to quantitative computing and weather forecasting. The value of differencing parameter in Integrated sub-model predicted in ARIMA suggested the presence of Random Walk in the time series of stock prices for short-term forecasting. ETS (Exponential Timeseries Smoothening) forecasting model is used on the logarithmic returns obtained from ARIMA to find the existence of drift through data visualizations in R. This drift is also the basis for the popular model of Random Walk.

The most probabilistic frequency of stationary deviations in closing price, found from the Ensemble model of ARIMA and Neural Network, is predicted through Random Walk

usage on 40 years stock prices of Apple Inc. on a Hadoop cluster. These approaches are combined with the prediction of Stock returns to uncover market patterns, plan strategies for future investments, and forecast the value of next 20-50 days' closing price and daily returns respectively.

1.3.1 Historical Stock Datasets

Popular sites like Yahoo! News & Google news have large chunks of research data which try to predict the stock market in real time. Huge volumes of historical data related to stocks of a company can be downloaded from these websites for time series forecasts. Before going into their details, the difference between DJIA (Dow Jones Industrial Average) and NASDAQ (National Association of Securities Dealers Automated Quotations) is mentioned herewith. The latter has security stocks which former doesn't have and also has a large no of companies registered with it. NASDAQ Historical Securities Stock Quotes provides real-time market information and analysis of stock quotes and financials. Company news like of Apple Inc. along with investing tools are featured on the Nasdaq Stock Market website.

Yahoo! Finance provides us the NYSE Quote of Apple Inc's stocks ($AAPL). Yahoo! Finance features historical data including stock quotes, financial news, and reports. One of the use case of this API with respect to Apple Inc. is for giving past 6 years stock details of Apple Inc. for probabilistic Time-Series logistic regression-based forecasting in Ensemble model of ARIMA and Neural Network in R-Studio to predict the closing prices for next 20-50 days. The other use case is in the MapReduce implementation on Hadoop framework to find the most likely percentage change occurring daily in a Random Walk model by loading huge volume of past 40 years historical stock dataset in Hadoop cluster and calculating frequencies of the daily drifts on the cluster nodes. This gives us the forecasts for daily returns for any number of days.

CHAPTER 2

2.1 RESEARCH IN FORECASTING OF MARKET MOVEMENTS THROUGH TWITTER SENTIMENT ANALYSIS

2.1.1 Fuzzy Neural Networks

The most popular research published in prediction through sentiment analysis area is by Bollen's [2] thorough investigation of the public sentiments into 6 dimensions of anxiousness, calmness, vitalness, kindness, sureness, and happiness obtained from feeds of Twitter data and its correlation with the Dow Jones Industrial Average (DJIA) Index values, found through usage of a Fuzzy Neural Network and Granger Causality Analysis for predicting and proving the existing correlation with an accuracy of 86.7 percent in market movements and closing values with very low MAPE (Mean Average Percentage Error). Behavioral economics tells us that emotions can profoundly affect individual behavior and decision-making. They verified whether this also applies to societies at large, i.e. can societies experience mood states that affect their collective decision making. By extension of it, they studied if the public mood is correlated or even predictive of economic indicators and investigated whether measurements of collective mood states derived from large-scale Twitter feeds are correlated to the value of the Dow Jones Industrial Average (DJIA) over time. For this, the text content of daily Twitter feeds was analyzed by two mood tracking tools, namely Opinion-Finder that measures positive vs. negative mood and Google-Profile of Mood States (GPOMS) that measures mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). The resulting mood time series was cross-validated by comparing their ability to detect the public's response to the presidential election and Thanksgiving Day in 2008. A Granger causality analysis and a Self-Organizing Fuzzy Neural Network are then used to investigate the hypothesis that public mood states, as measured by the Opinion-Finder and GPOMS mood time series, are predictive of changes in DJIA closing values. Their results indicate that the accuracy of DJIA predictions can be significantly improved by the inclusion of specific public mood dimensions but not others. In particular, the variations along the public mood dimensions of Calm and Happiness as measured by GPOMS seem to have a predictive effect, but not general happiness as measured by the Opinion-Finder tool. The accuracy found by them was of 87.6% in predicting the daily up

and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error by more than 6%.

2.1.2 Sentiment Word-Net

Chen and Lazer [8] built a prediction model to derive the investment strategies through the observation and classification of the twitter feeds into Financial positive and negative sentiments. They begin their sentiment analysis by applying Alex Davies' word list in order to see if a simple approach is sufficient enough to correlate to market movement. For this, a pre-generated word list was used of roughly five thousand common words along with log probabilities of 'happy' or 'sad' associated with the respective words. The process worked as follows. First, each tweet was tokenized into a word list. The parsing algorithm separated the tweets using whitespace and punctuation, while accounting for common syntax found in tweets, such as URLs and emoticons. Next, each token's log-probability was looked up in the word list; as the word list was not comprehensive, they chose to ignore the words that do not appear in the list. The log probabilities of each token was simply added to determine the probability of 'happy' and 'sad' for the entire tweet. These were then averaged per day to obtain a daily sentiment value. As expected, this method resulted in highly uncorrelated data (with correlation coefficients of almost zero). They tried to improve this by using a more comprehensive and accurate dictionary for positive and negative sentiments. Specifically, the initial word list was swapped with a sentiment score list which was generated using Sentiment-WordNet, which consisted of over 400 thousand words. This list considered relationships between each word and included multi-word expressions, because of which it provided better results. They also tried representing the daily sentiment value in a different way - instead of averaging the probabilities of each tweet, they counted the frequency of 'happy' tweets (such as using a threshold probability of above 0.5 for happy) and represented this as a percentage of all tweets for that day. While this did not improve the output's correlation with stock market data, it did provide them with more insight into their Twitter data. For example, they were able to see a spike in the percentage 'happy' tweets toward the end of each month. They did not find news events which could have caused these spikes; however, upon investigating the source of Twitter data, they found that it had been pre-filtered for a

previous research project (i.e. there may be some bias in what they assumed to be raw Twitter data). Due to a lack of access to better Twitter data, they concluded that using the frequency of happy tweets is not a reliable indicator of sentiment for their application and reverted back to their averaging method.

2.1.3 Data Mining

On the basis of industry type in which people work, Bing et al. [9] studied twitter dataset to predict prices of stock markets. Some of the industries they used were Technology, Finance, Information Technology, etc. In their research, they proposed a method to mine Twitter data for answers to the questions of predicting movements of stock markets from public sentiments on social media to finding out which company has more predictable stocks. As one of the most popular social media, more than millions of users post over 140 million tweets every day, so according to them, this situation makes Twitter like a corpus with valuable data. More specifically, some social media sites like Facebook and Twitter are explicitly designed for social interactions, while others like Flickr are more tendentiously designed for content sharing. Attributed to social media's high level of ease to use, reach, richness and immediacy, public opinions and discourses are changing rapidly, and its influences are extended to various domains such as politics, environment, entertainment industry, stock market, etc. The availability of massive amounts of data has drawn great attentions on researching social media statistically. Specifically, they proposed to use a data mining algorithm to determine if the price of a selection of 30 companies listed in NASDAQ and the New York Stock Exchange can actually be predicted by using the collected 15 million records of tweets (i.e., Twitter messages). For this, they extracted ambiguous textual tweet data through NLP techniques to define public sentiment, then made use of a data mining technique to discover patterns between public sentiment and real stock price movements. With their proposed algorithm, they managed to discover that it is possible for the stock closing price of some companies to be predicted with an average accuracy as high as 76.12%.

2.1.4 Distant Supervised Learning

A strong negative correlation pattern was found out by Zhang [10], between public opinions of worriedness, fearfulness, and hopefulness in tweets and values of DJIA

indices. Twitter messages, or tweets, can provide an accurate reflection of public sentiment on when taken in aggregation. In their research paper, they primarily examined the effectiveness of various machine learning techniques on providing a positive or negative sentiment on a tweet corpus collected from Twitter API. Additionally, they applied the extracted twitter sentiment to accomplish two tasks. First one was to look for a correlation between twitter sentiment and stock prices by joining them using same days. Secondly, they determined which words in tweets correlate to changes in stock prices by doing a post analysis of price change and tweets. They accomplished this by mining tweets using Twitter's search API and subsequently processing them for analysis. For the task of determining sentiment, they tested the effectiveness of three machine learning techniques: Naive Bayes classification, Maximum Entropy classification, and Support Vector Machines. They discovered that SVMs give the highest consistent accuracy through cross validation among all of the techniques, but not by much. They also discussed the various approaches used in training these classifiers. They applied their findings after that to an intra-day market scale and it was found that there was very little direct correlation between stock prices and tweet sentiment on specifically an intra-day scale. Next, they improved on the keyword search approach by reverse correlating stock prices to individual words in tweets, and finding reasonably, that certain keywords are more correlated with changes in stock prices. Lastly, they discussed the various challenges posed by looking at twitter for performing stock market predictions.

2.1.5 N-gram, word2vec and Random Forest

A research in this field was also done by Brian et al. [11] recently, based on the correlation of stock price rise and fall with the public sentiments through the study of Pearson correlation coefficient for stocks. In their study, they seek to predict a sentiment value for stock related tweets on Twitter and demonstrate a correlation between this sentiment and the movement of a company's stock price in a real time streaming environment. Both n-gram and "word2vec" textual representation techniques were used alongside a random forest classification algorithm to predict the sentiment of tweets. These values were later evaluated for correlation between stock prices and Twitter sentiment for that each company. There were significant correlations between price and

sentiment for several individual companies. Some companies such as Microsoft and Walmart showed strong positive correlation, while others such as Goldman Sachs and Cisco Systems showed strong negative correlation. This suggested that consumer facing companies are affected differently than other companies. Their Stanford model makes use of deep neural networks. Deep neural networks are an expansion on early neural networks such as Perceptrons. Advances in hardware processing speeds, particularly graphics processing units, as well as an increasing interest in parallel processing have brought resurgence in the use of artificial neural networks by enabling the addition of hidden layers of neurons, and backpropagation. The additional layers allow these models to become more highly non-linear fitting closer to the data, while backpropagation enhances training efficiency on labeled data in deep neural networks.

2.1.6 Self Organizing Fuzzy Neural Network

In a paper of Mittal et al. [12], it was proven that the mechanism of predicting with accuracy rate to be around 75 percent with a usage Fuzzy neural networks on DJIA Index and Twitter Feeds. They proposed a new cross validation method for financial data and obtained 75.56% accuracy using Self Organizing Fuzzy Neural Networks (SOFNN) on the Twitter feeds and DJIA values from the period June 2009 to December 2009. They also implemented a naive portfolio management strategy based on their predicted values. The raw DJIA values were first fed into the preprocessor to obtain the processed values. At the same time, the tweets were 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 were then fed to their 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 were 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.

2.2 RESEARCH IN PREDICTING CLOSING PRICE FROM TIME SERIES FORECASTING

2.2.1 Ant Colony Optimization

Research in the prediction of Stock Markets has also been done with help of algorithms in Artificial Intelligence field by Bouktif et al. [17]. They combined Bayesian Classifiers with the Ant Colony Optimization algorithm on the public mood states obtained from Twitter giving a significant performance in prediction accuracy of concerned stock values. They built a new prediction model for the same stock market problem based on single models combination. Their proposed approach to build such model was simultaneously promoting both performance and interpretability. By interpretability, the ability of a model to explain its predictions is referred. They compared their approach against the best Bayesian single model, model learned from all the available data, bagging and boosting algorithms. The test results indicated that their proposed model for stock market prediction performs better than those derived by alternatives approaches.

2.2.2 Convolutional Neural Network

The approach related to Time series forecasting has also been researched through Deep Neural Networks by Gunduz et al. [18]. Convolutional Neural Network was used for prediction of the direction of movement of the closing price of GARAN, THYAO and ISCTR stocks. The daily movement directions of three frequently traded stocks (GARAN, THYAO and ISCTR) in Borsa Istanbul were predicted using deep neural networks. Technical indicators obtained from individual stock prices and dollar-gold prices were used as features in the prediction. Class labels indicating the movement direction were found using daily close prices of the stocks and they were aligned with the feature vectors. In order to perform the prediction process, the type of deep neural network, Convolutional Neural Network, was trained and the performance of the classification was evaluated by the accuracy and F-measure metrics. In the experiments performed, using both price and dollar-gold features, the movement directions in GARAN, THYAO and ISCTR stocks were predicted with the accuracy rates of 0.61, 0.578 and 0.574 respectively. Compared to using the price-based features only, the use of dollar-gold features by them improved the classification performance.

2.2.3 LSTM Neural Network

LSTM Neural Networks were also used in time series forecasts by Nelson et al. [20] for predicting future trends in the market through stock price history and analysis indicators

for forecasting of movement direction. They studied the usage of LSTM networks on that scenario, to predict future trends of stock prices based on the price history, alongside with technical analysis indicators. For that goal, a prediction model was built, and a series of experiments were executed and theirs results analyzed against a number of metrics to assess if this type of algorithm presents improvements when compared to other Machine Learning methods and investment strategies. The results that were obtained were promising, getting up to an average of 55.9% of accuracy when predicting if the price of a particular stock is going to go up or not in the near future.

2.2.4 Gene Expression Programming

An Ensemble model of forecasting has been studied by Bautu et al. [19] by ensembles of Gene Expression Programming (GEP) evolved models in deep learning and artificial intelligence fields through binary classification of stock prices. They investigated new ways to obtain models for time series of stock price index movements. They approached the problem in a supervised learning fashion, as binary classification, and applied Gene Expression Programming (GEP) to obtain classifier models. This evolutionary approach works with a population of candidate models and evolves them in epochs, by means of genetic operations, towards an optimal solution. The models evolved make use of any of the possible features, at any time during the algorithm. Unlike traditional techniques, GEP performs a global search of the space of models. Candidate solutions are evaluated against a fitness function, so that the algorithm discerns between variously fit models. The genetic operators allowed the inheritance of valuable information from prior generations, the exchange of information between peers in the same generation and also the birth of new knowledge to be stored in the population in each epoch. The performances of pure GEP evolved classifiers were acceptable when compared with those of classifiers induced by other machine learning techniques. Combining the output of multiple models when making predictions is a common technique in machine learning, used to obtain more robust and efficient, yet complex, classifiers based on more primitive ones. They used many ensemble techniques to enhance the generalization power of the GEP evolved classifiers. The obtained meta-models were empirically tested on real-world stock data. Several state-of-the-art machine learning methods—Naive Bayes, Support

Vector Machines, Multi-Layer Perceptron, Decision Table and Random Forrest—were applied to the same task. The experiments performed on real-world stock market data showed that the ensembles of GEP evolved classifier models are competitive to classifiers trained by state-of-the-art machine learning methods.

2.2.5 ARIMA and EMMS models

ARIMA model (Auto-Regressive Integrated Moving Average) was studied for forecasting in R by Angadi et al. [7] in depth and was found to be very accurate for short-term forecasting of stock market trends. They proposed a model for forecasting the stock market trends based on the technical analysis using historical stock market data and ARIMA model. Their model automated the process of direction of future stock price indices and provides assistance for financial specialists to choose the better timing for purchasing and/or selling of stocks. The results were shown in terms of visualizations using R programming language. The obtained results revealed that the ARIMA model has a strong potential for short-term prediction of stock market trends.

EMMS (Expert Model Mining System) combining AR (Auto-Regression), I (Integration), MA (Moving Average) with ES (Exponential Smoothening) was researched by Rao et al. [1] reducing MAPE by combining it with social media Sentiment Analysis Correlations for NASDAQ (National Association of Securities Dealers Automated Quotations) and DJIA values with 75.56 percent accuracy through learning algorithms of Support Vector Machines and Neural Networks for Portfolio Management implemented in MATLAB software. They investigated the complex relationship between tweet board literature (like bullishness, volume, agreement etc) with the financial market instruments (like volatility, trading volume and stock prices). They analyzed sentiments for more than 4 million tweets between June 2010 to July 2011 for DJIA, NASDAQ-100 and 13 other big cap technological stocks. The results showed high correlation (upto 0.88 for returns) between stock prices and twitter sentiments. Further, using Granger's Causality Analysis, they validated that the movement of stock prices and indices are greatly affected in the short term by Twitter discussions. Finally, they have implemented Expert Model Mining System (EMMS) to demonstrate that the forecasted returns give a high value of Rsquare (0.952) with low Maximum Absolute Percentage Error (MaxAPE) of 1.76% for Dow

Jones Industrial Average (DJIA). The selection criterion for the EMMS was coefficient of determination (R squared) which is square of the value of pearson-'r' of fit values (from the EMMS model) and actual observed values. The EMMS was applied twice - first with tweets features as independent predictor events and second time without them. They have applied simplistic message board approach by defining bullishness and agreement terminologies derived from positive and negative vector ends of public sentiment w.r.t. each market security or index terms (such as returns, trading volume and volatility). Their method was not only scalable but also gave more accurate measure of large scale investor sentiment that can be potentially used for short term hedging strategies. This gave clear distinctive way for modeling sentiments for service-based companies such as Google in contrast to product-based companies such as Ebay, Amazon and Netflix. The aim of their work, was to quantitatively evaluate the effects of twitter sentiment dynamics around a stocks indices/stock prices and use it in conjunction with the standard model to improve the accuracy of prediction.

CHAPTER 3

3.1 SENTIMENT ANALYSIS

With no doubt, though uninteresting individually, tweets can provide a satisfactory reflection of public sentiment when taken in aggregate. Tweets are mined using Twitter's Search API and subsequently processed them for further analysis, which included Natural Language Processing (NLP) and Sentiment Analysis. Thereafter, we applied Naive Bayes and SVM to predict each tweet's sentiment. Correspondingly the financial data from Yahoo Finance was collected. Correlation is then examined between normalized average sentiment score and daily returns.

3.1.1 Data Collection

Historical twitter data is collected through Twitter API for python called Tweepy. For the process of collecting tweets, Twitter provides two possible ways to gather tweets, the Streaming API or the Search API. The Streaming API allows users to obtain real-time access to tweets from an input query. The user first requests a connection to a stream of tweets from the server. Then, the server opens a streaming connection and tweets are streamed in as they occur, to the user. However, there are a few limitations of the Streaming API. First, language cannot be specified, resulting in a stream that contains Tweets of all languages, including a few non-Latin based alphabets, that complicates further analysis. Because of these issues, we decided to go with the Twitter Search API instead. The Search API is a REST API which allows users to request specific queries of recent tweets. The Search API allows filtering based on language, region, geolocation and time. There is a rate limit associated with the query, but we handle it in the code. Higher rate limits are obtained by using application only authentication. A query only returns 100 tweets. More tweets are obtained by saving the id of the last tweet fetched. The request returns a list of JSON objects that contain the tweets and their metadata. This includes a variety of information, including username, time, location, retweets, and more. For our purposes, we mainly focus on the time and tweet text. We use as query the ticker of the company in front of which we add the dollar sign to gather the most "financial" tweets.

The historical stock prices are obtained from Yahoo Finance website [16]. This dataset consists of open, high, low, close values for each day.

3.1.2 Data Preprocessing

The text of each tweet includes a lot of words that are irrelevant to its sentiment. For example, some tweets contain URLs, tags to other users, or symbols that have no meaning. In order to better determine a tweet's sentiment score, before anything else we had to exclude the "noise" that occurred because of these words. For this to happen, we relied on a variety of techniques using the Natural Language Tool Kit (NLTK) for Python and regular expressions. We first do some general pre-processing on tweets which are as follows.

- Convert the tweets to lower case
- Replace 2 or more dots (.) with space.
- Strip spaces and quotes (" and ') from the ends of the tweet.
- Replace 2 or more spaces with a single space.

3.1.2.1 Handling URL

Users often share hyperlinks to other web pages in their tweets. Any particular URL is not important for text classification as it would lead to very sparse features. Therefore, we replace all the URL's in tweets with the word URL.

3.1.2.2 Handling User Mention

Every Twitter user has a handle associated with them. Users often mention other users in their tweets by @handle. We replace all user mentions with the word USER_MENTION.

3.1.2.3 Handling Emoticons

Users often use a number of different emoticons in their tweet to convey different emotions. We replace the matched emoticons with either EMO_POS or EMO_NEG depending on whether it is conveying a positive or a negative emotion.

3.1.2.4 Handling Hashtags

Hashtags are unspaced phrases prefixed by the hash symbol (#) which are frequently used by users to mention a trending topic on Twitter. We replace all the hashtags with the words with the hash symbol. For example, #hello is replaced by hello.

3.1.2.5 Handling Retweets

Retweets are tweets which have already been sent by someone else and are shared by other users. Retweets begin with the letters RT. We remove RT from the tweets as it is not an important feature for text classification.

After applying tweet level pre-processing, we processed individual words of tweets as follows

- Strip any punctuation [‘ " ? ! , . ( ) : ;] from the word.
- Convert 2 or more letter repetitions to 2 letters. Some people send tweets like I am sooooo happpppy adding multiple characters to emphasize on certain words. This is done to handle such tweets by converting them to I am soo happy.
- Remove – and ’. This is done to handle words like t-shirt and their’s by converting them to the more general form tshirt and theirs. Check if the word is valid and accept it only if it is. We define a valid word as a word which begins with an alphabet with successive characters being alphabets, numbers or one of dot and underscore
- Use lemmatization to convert a word to its base form.
- Remove stop words from the tweet.

Returns are obtained from the stock dataset as follows, where $R_d$is the returns for the current day d and P_d is the closing price of the stock on day d.

$$R_d = \frac{P_d - P_{d-1}}{P_d}$$

3.1.3 Training Dataset Collection

To train a sentiment analyzer, we need a collection of labelled tweets . As the collected tweets are huge in number , manual labelling of tweets is not possible. Hence we resorted to two methods to label the tweets. The first method is labelling tweets using emojis. As we discussed in preprocessing, tweets containing labels EMO_POS are labelled as positive and tweets containing EMO_NEG are labelled as negative. The second method is using AFINN word dictionary [22] which consists of positive and negative words, their sentiment labelled between -5 to 5. The net sentiment of a tweet is sum of all word sentiments. If it is greater than zero, the tweet is labelled as positive.If it is less than zero,

tweet is labelled as negative. We used second method as the number of tweets containing emojis are very low.

3.1.4 Feature Extraction

A unigram is simply an N-gram of size one. For each unique tokenized word in a tweet, a unigram feature is created for the classifier. For example, if a negative tweet contains the word “bad”, a feature for classification would be whether or not a tweet contains the word “bad”. Since the feature came from a negative tweet, the classifier would be more likely to classify other tweets containing the word “bad” as negative. Likewise, a bigram is an N-gram of size two and a trigram is an N-gram of size three. That means that in the case of bigrams the feature vector for the classifier is made of a two-word combination and in the case of trigrams is made of a three-word combination respectively. For example, if a negative tweet contains the combination “not perfect”, in the case of the bigram feature extraction it would be classified as a negative tweet. Instead, if only unigram features were used, the tweet would have been classified as positive since the term “not” has a neutral sentiment and the term “perfect” a positive one.

3.1.5 Feature Filtering

With the method described above, the feature set grows larger and larger as the dataset increases leading to the point where it becomes difficult and unnecessary to use every single unigram, bigram, as a feature to train our classifier. So we decided to use only the n most significant features for training. We used a chi-squared test, Pearson’s chi-squared test in particular to score each unigram, bigram. NLTK helped us to determine the frequency of each feature. Having now, the features ordered by score, we selected the top-10000 to use for training and classification.

3.1.6 Machine Learning Algorithms

After labelling the data and feature extraction, we build a sentiment analyzer that determines the label of the tweet. The sentiment analyzer is a binary classifier which classifies the tweet into positive or negative.We used two types of classifiers: Naive Bayes Bernoulli and Support Vector Machine. We chose to focus on these algorithms because according to [21], they are the state of the art for Sentiment Analysis.

3.1.6.1 Naive Bayes Classifier

- A Naive Bayes classifier is based on Bayes rule and simple form of Bayesian network. It is a simple probabilistic model which relies on the assumption of feature independent to each other in order to classify data.
- The algorithm assumes that each feature is independent of presence or absence of other feature in input data, so this assumption is known as 'naive'.
- Bayes classifier use Bayes theorem, which is

$$\mathrm{P(c|F)} = \frac{P(F/c\ )P(c)}{P(F)}$$

P(c/F) = probability of feature F being in class c

P(F/c) = probability of generating feature F given by class c

P(c) = probability of occurrence of class c

P(F) = probability of feature F occurring

- In above context, we are looking for class c, so we find probable class given for given feature, F. Denominator does not depend on class, so we treat it as a constant. Numerator depends on class so we focus to determine the value of P(F/c). For a class c_j, the features are conditionally independent of each other, hence

$$P\left(f1, f2 \ldots \frac{fn}{cj}\right) = \prod_i P(\frac{fi}{cj})$$

From these, we classify tweet statistics with label c* with a maximum posterior decision taking the most probable label from all labels C.

$$c* = arg \max_{cj \in C} P(cj) \prod_i P(fi/cj)$$

The Naïve Bayes is very simple and its conditional independence assumptions are not realistic in the real world. However, it gives better accuracy for stock prediction.

3.1.6.2 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. Unlike the Naive Bayes classifier, the SVM is a large margin classifier, rather than probabilistic. In previous works, SVMs have been shown to be very effective for text categorization. The SVM is a classifier that attempts to find a separation between a linearly separable set of data, with as wide of a gap as possible between them, called a margin. An SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. With our training set as an input, the SVM finds the hyperplane such that each point is correctly classified and the hyperplane is maximally far from the closest points. The name ”support vector” comes from points on the margin between the hyperplane and the nearest data points, which are called support vectors. The SVM looks for a parameter vector a that, again, maximizes the distance between the hyperplane and every training point. In essence, it is an optimization problem:

$$\text{Minimize } \frac{1}{2}\, a * a$$

$$y * (ax_i + b) \geq 1$$

where y is the class label (−1, 1) for negative and positive. Once the SVM is built, classification of new tweets simply involves determining which side of the hyperplane that they fall on. In our case, there are only two classes, so there is no need to go to a non-linear classifier.

3.2 TIME-SERIES FORECASTING

The problem of forecasting the future price of securities on the stock market (or currency exchange rates, and so on). Markets have very different statistical characteristics similar to natural phenomena such as weather patterns. Machine learning and Deep learning combination can be used to forecast markets, through access to publicly available data and when it comes to markets, past performance can be used as a good predictor of future returns by using the changes in small seasonal intervals.

3.2.1 ARIMA Model

Machine learning is applicable to datasets where the past is a good predictor of the future by dividing past years data into small seasonal intervals like ARIMA model's moving distributed lagged dataset. ARIMA stands for a combination of Autoregressive Models (AR), Integrated Models (I), and Moving Average Models (MA) & Seasonal Regression Models. ARIMA is used in financial time series because it can be viewed as piecewise stationary or short-time stationary movements. It is a type of the Distributed Lags Model. The three models combined were:

3.2.1.1 Auto Regression (AR) Models– A type of random process is represented by describing certain time-varying processes. The output variable of given time series is regressed on its own lagged values [Fig.3.1]. The number of time lags is denoted by the "p" value in the model.

3.2.1.2 Differencing or Integrated (I) Models – It indicates that the data values were replaced with the difference between their values and the previous ones, known as differencing. Distributed regression of the time series is involved to convert a non-stationary time series to a stationary one. The degree of differencing is denoted by the "d" value in the model.

Non-seasonal ARIMA models are represented by ARIMA (p, d, q). Seasonal ARIMA models have another factor "m", where m is the number of periods in each season. Some of the standard values for ARIMA are (1,1,0), (1,2,1), (1,0,0), (0,1,0). To get most accurate values for (p, d, q), the value on each of the small subparts of the time series curve was calculated and used for that part itself.

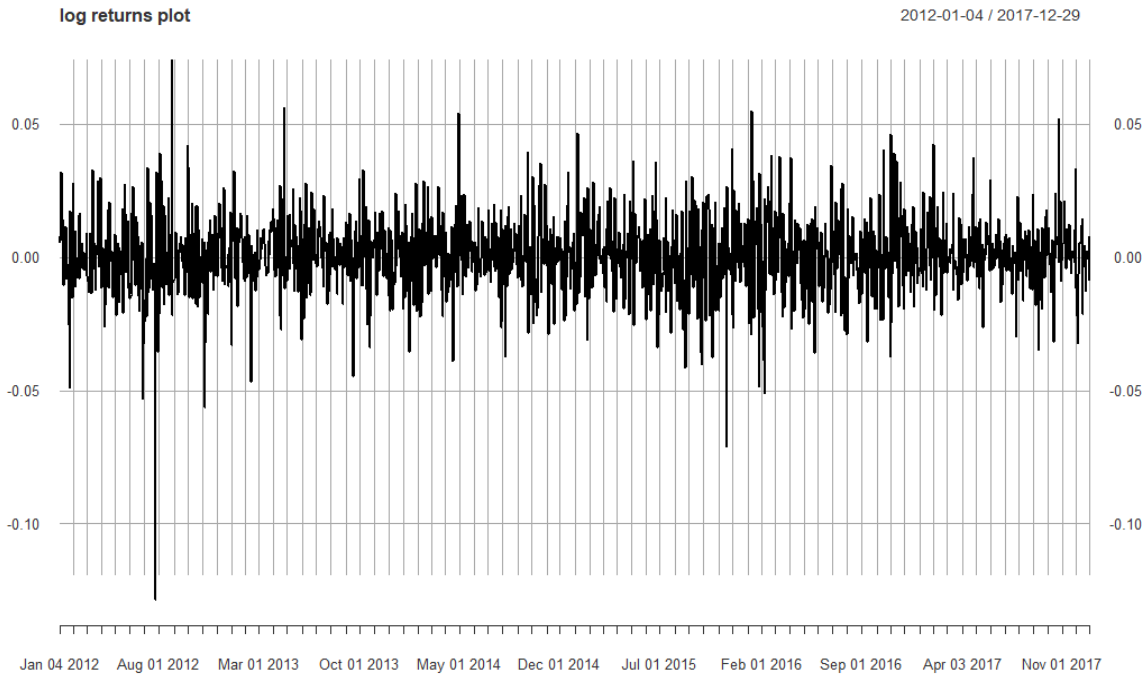


Fig.3.1 Logarithmic Returns Plot for Apple Inc. from Yahoo! Finance

3.2.1.3 Moving Average (MA) Models – It gives regression error as a linear combination of past error terms. The number of lagged values of the error term is denoted by the "q" value in the model

The general equation for ARIMA can be given as-

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t = \delta + \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t.$$

where this defines an ARIMA (p, d, q) process with drift equal to $\delta/1-\Sigma\,\varphi_i$

Here, ACF plot and PACF plot [Fig.3.2] were used to give the accurate lagged values of the error term for getting "q" value. The values of "p", "d" and "q" were adjusted automatically in each seasoned interval, for improving the accuracy of the overall prediction. The testing of ARIMA gave an idea of how well the model fits the data through AIC , AICc and BIC values. All the values obtained were very low, and lower

the values of these terms, the better the ARIMA model fits. The obtained values of (p,d,q) were (2,1,2) respectively.

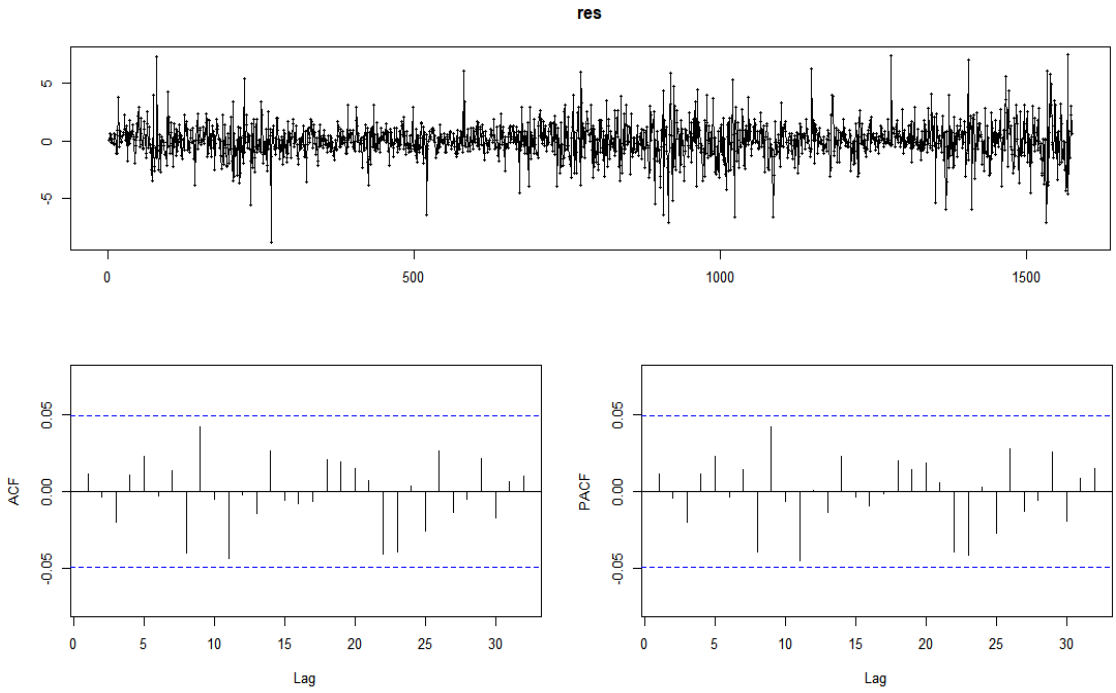


Fig.3.2 Residuals from Logarithmic Returns along with ACF and PACF Plots of them

AR (2) meant that logarithmic regression was based on previous 2 values, d=1 meant that Random Walk was present in the time series of Closing prices, and MA (2) implied that 2 regression values were used as a combination of lagged error terms in differencing. After that, an extensible time series (xts) object is initialized for Actual log returns and a data frame for the forecasted return series. A loop was then run through each seasoned interval for training and later testing the forecasted values of daily returns in logarithms.

3.2.2 Neural Network

A feed-forward Neural Network was used on the forecasted logarithmic returns obtained from ARIMA through the (p, d, q) parameters passed to the Neural Network. The network was constructed with a single hidden layer with ARIMA's trained lagged inputs for forecasting and number of states equal to half of the stock prices for training. Number

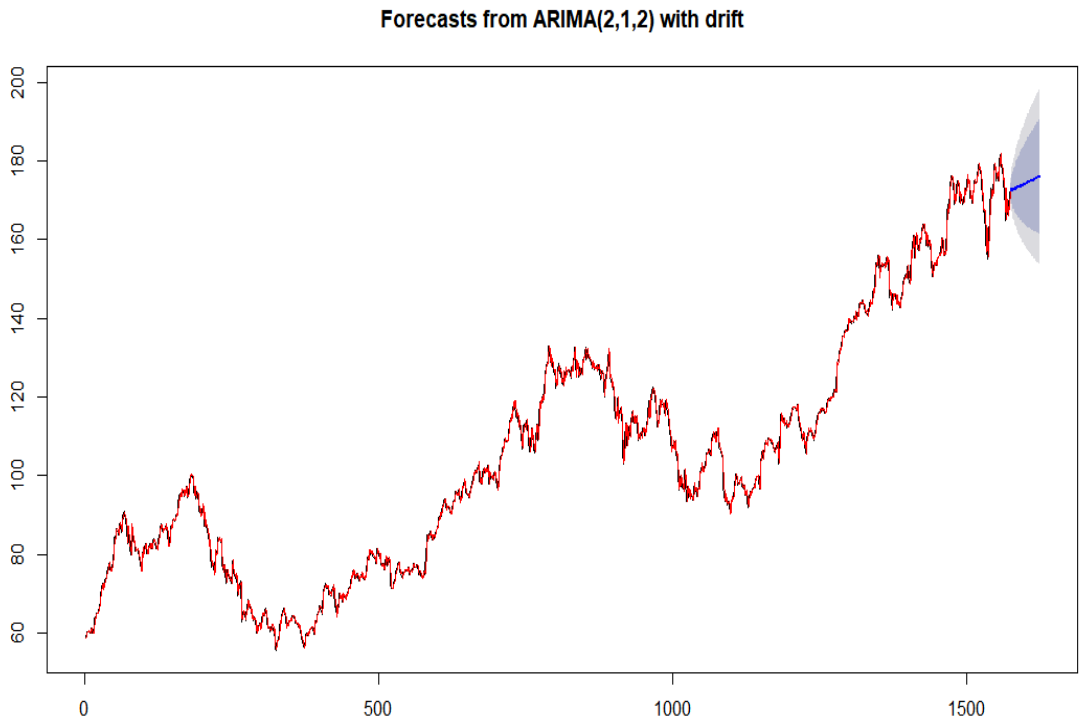
of non-seasonal lags passed from ARIMA results of ‘p’, and 20 networks were fitted with random start weights using logistic sigmoid function.

$$y(v_i) = (1 + e^{-v_i})^{-1}.$$

Equation of the logistic sigmoid function ranges from 0 to 1. Here y(v[i]) is the output of the i^{th} node (neuron) and v[i] is the weighted sum of the input connections inside hidden layer, each node in one layer connects with a certain weight w[i, j] to every node in the following layer.

The model predicts the Closing price for next 50 days along with the logarithmic Daily change for any number of days. The results obtained predicted the market direction which was verified on the NASDAQ and forecasted Closing Price, which was very near to the actual value

3.2.3 Random Walk Model



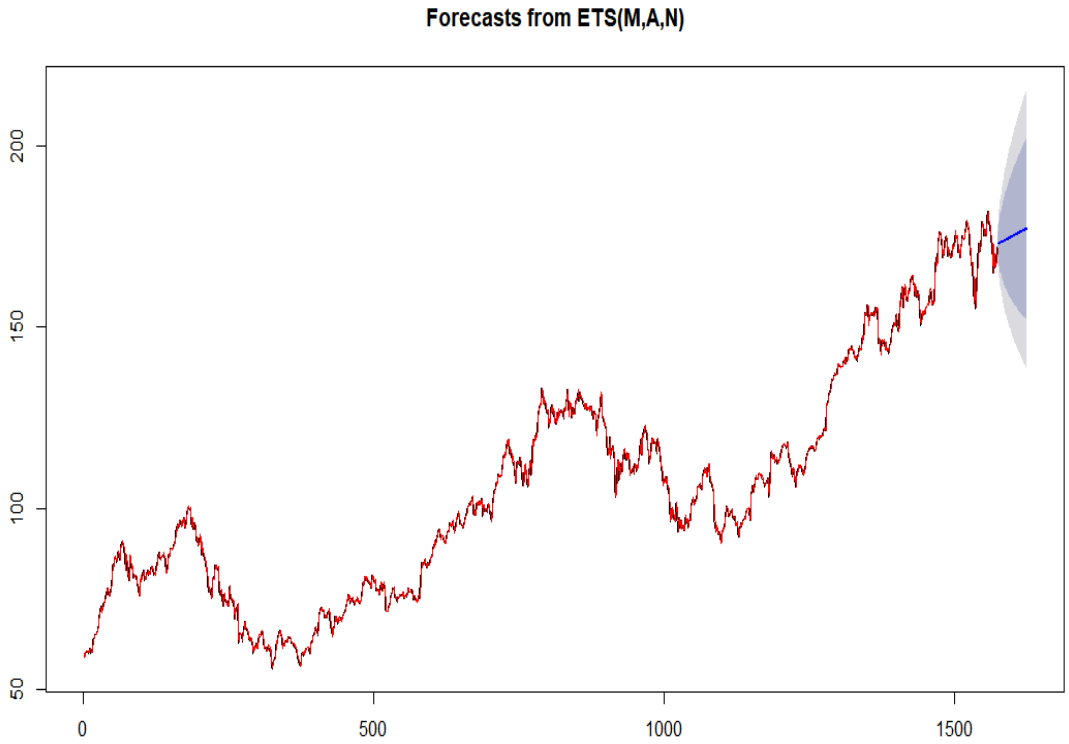


Fig.3.3 The Drift boundaries of ARIMA, verified by Exponential Smoothening (ETS) predicted by Random Walk of Ensemble Model

The Random Walk Model is used in Ensemble for finding the Drift boundaries obtained [Fig. 11] in ARIMA forecasts. It is a famous theory among financial hypotheses which says that time-series data is formed from the random or stochastic process, that consists of a series of random steps on the financial historical dataset space such as the Daily Changes. It connects the next day's opening price to be previous day's closing price with most likely deviation to be maximum frequency/probability of percentage change which occurred on daily basis.

$$X_t = \mu + X_{t-1} + \epsilon_t$$

X[t] is the prediction of Deviation in current day, X[t-1] is previous day's closing price, u is the most probabilistic frequent boundary value of drift and error term is added along with it for stationary time series.

Through a MapReduce implementation of frequencies of past Daily changes on Hadoop framework, the most likely boundary values of percentage change for the region of 95 percent and 90 percent occurring daily is found from the Random Walk model from the historical data of past 40 years.

CHAPTER 4

4.1. Sentiment Analysis Results

4.1.1 Tweet statistics

After gathering the tweets, they are analysed to see the sentiment distribution.Majority of the tweets are neutral as observed by the following plot.

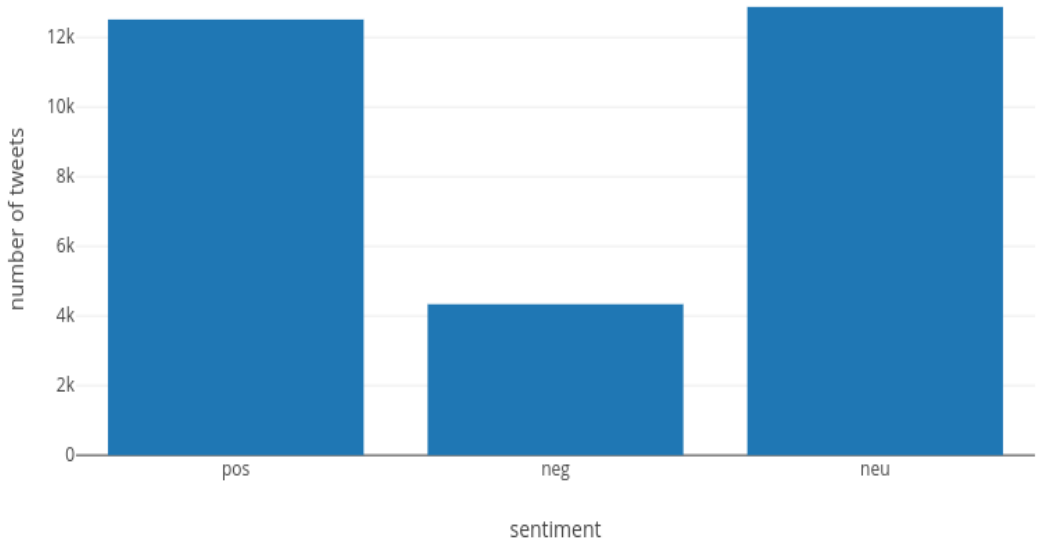


Fig 4.1 Total distribution of sentiments for Apple Inc

Each day as well ,the majority of the tweets are neutral. The following is the distribution of sentiments for Apple Inc over time.

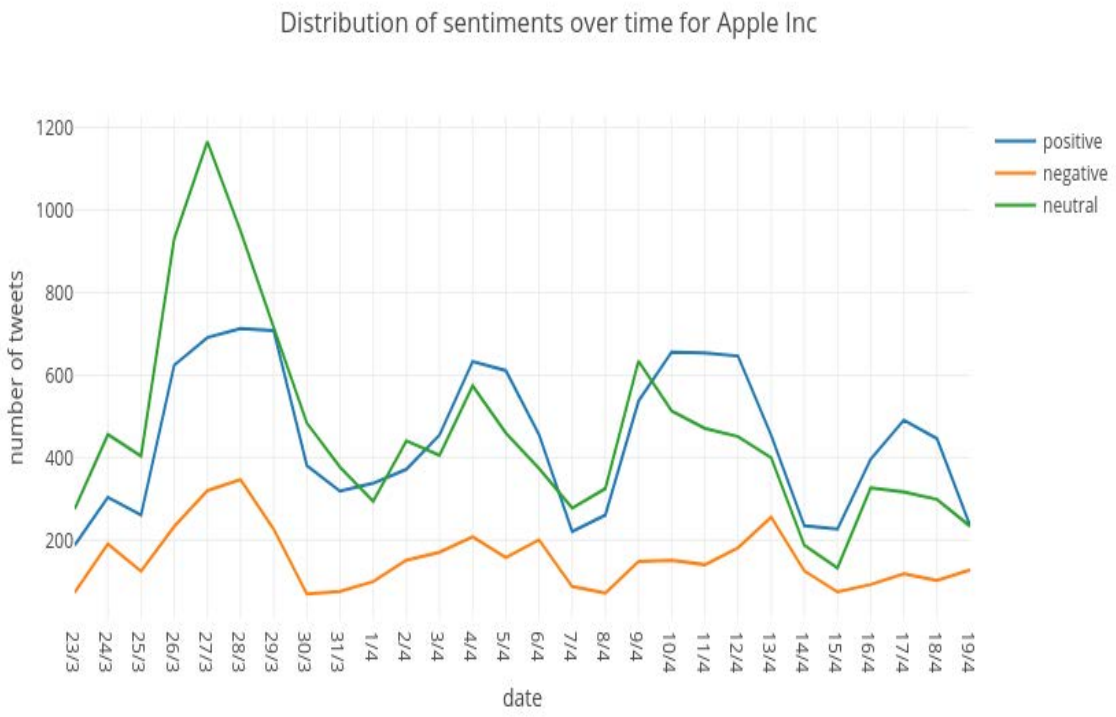


Fig 4.2 Distribution of sentiments over time for Apple Inc

The document-term matrix was formed for words in all the tweets from the word corpus and ranked based on high frequency. A text cloud or word cloud is a visualization of word frequency in a given text as a weighted list. A data frame was created for each word containing the word list along with their frequencies sorted in reverse order. After this, the "wordcloud2" package was used in R to visualize the words which appeared in the word cloud [Fig.4.3] of Apple Inc along with their frequencies. Bigger size indicates more frequent words.

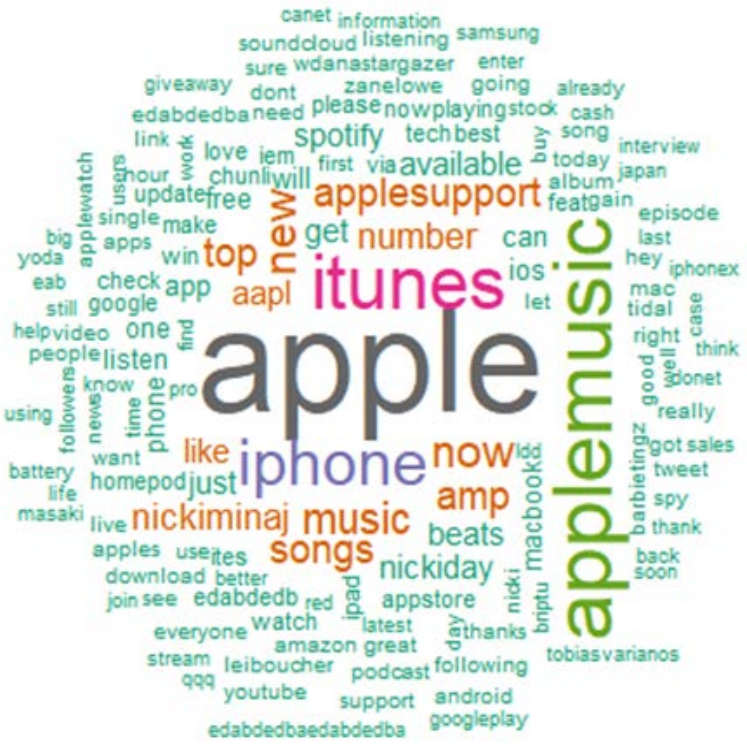


Fig 4.3 Word Cloud of Apple Inc.

4.1.2 Classifier Evaluation

After building sentiment analyzer, Accuracy of both the models is tested using 10-fold cross validation. SVM yielded better accuracy marginally. Below are the obtained metrics.

Table 4.1 Classifier Evaluation

	Naïve Bayes	SVM
Accuracy	0.79	0.81
Precision	pos : 0.78 neg : 0.54	pos :0.79 neg : 0.61
Recall	pos : 0.58 neg : 0.81	pos : 0.62 neg : 0.83
F-measure	pos : 0.68 neg : 0.57	pos : 0.69 neg : 0.75

4.1.3 Examining Correlation

Financial tweets are collected for Apple Inc by querying through $AAPL ticker for a period of 19 days .After finding the sentiments ,correlation is observed between normalized sentiment score and daily returns. The normalized sentiment score for a given day is calculated as ratio of difference between number of positive tweets and negative tweets to sum of them.

$$S_d = \frac{pos - neg}{pos + neg}$$

where S_d is the normalized net sentiment on a given day d . pos and neg are the number of positive and negative tweets respectively. A plot of S_d and R_d (described in section 3.1.2) versus date is plotted over a period of 19 days. As observed, there is a significant correlation between them.

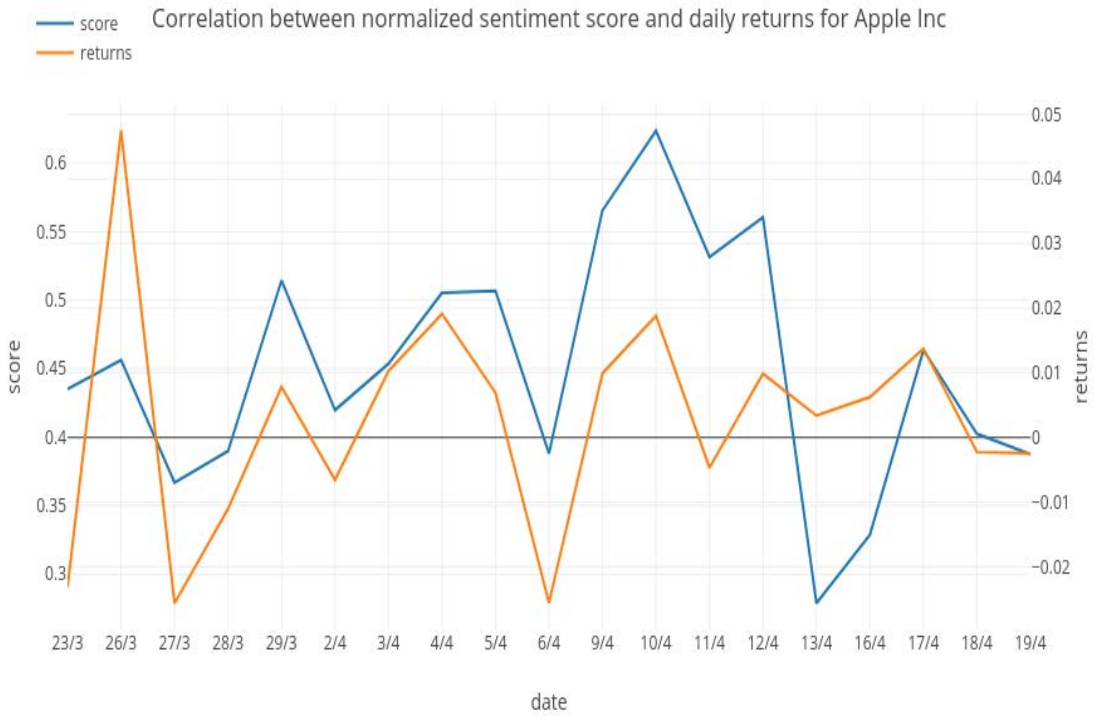


Fig 4.4 Correlation between normalized sentiment scores and daily returns for Apple Inc.

4.2 Time Series Forecasting Results from Ensemble Model

4.2.1 ARIMA forecasts

ARIMA used 97% data for training and 3% for testing from 6 years data. The Accuracy Percentage of the forecast was calculated by aggregating values in a table. The predicted stock returns for testing logarithmic returns data [Fig. 4.5] are also compared with their actual returns in the testing phase of ARIMA. The accuracy was 50 percent until here and was increased through the usage of Neural Network in the Ensemble Model.

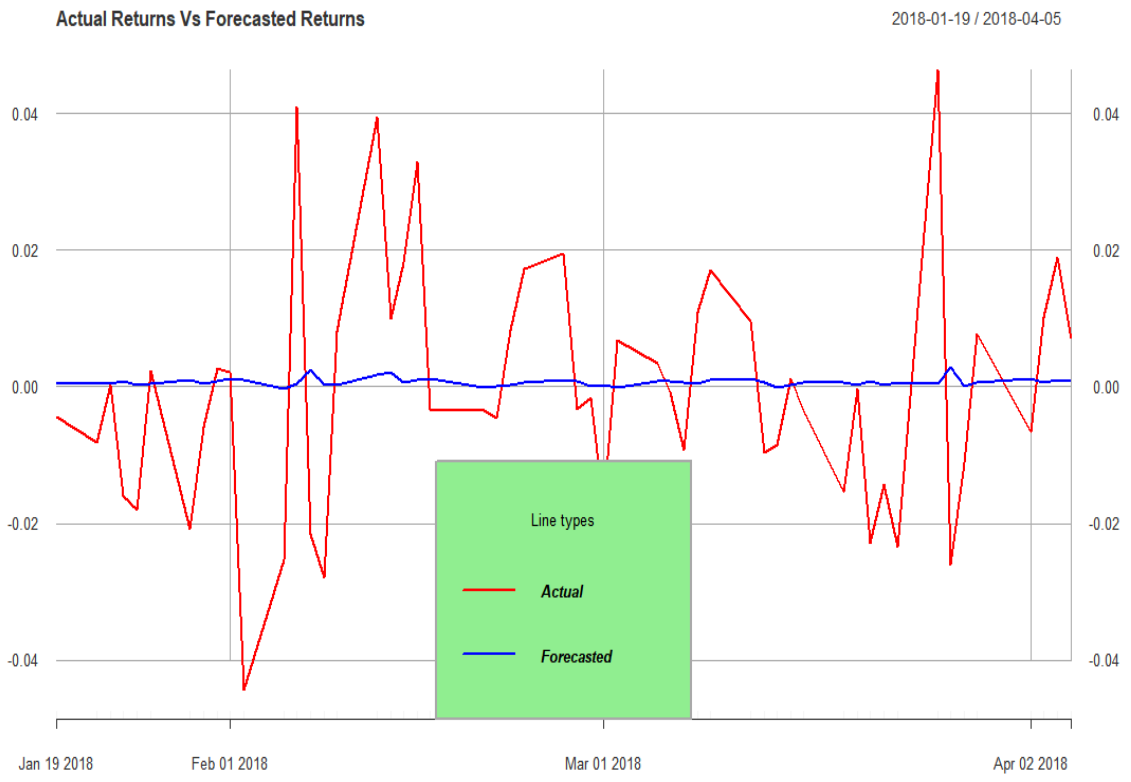


Fig.4.5. Actual Returns Vs Forecasted Returns for Apple Inc. from ARIMA

4.2.2 Neural Network forecasts and predictions

The Neural network was constructed with a single hidden layer with ARIMA's trained lagged inputs for forecasting and number of states equal to half of the stock prices for training. Number of non-seasonal lags passed from ARIMA results of 'p', and 20 networks were fitted with random start weights using logistic sigmoid function. The testing proved the results to be more accurate from the Ensemble of ARIMA and Neural Network to be 89 percent accurate [Fig.4.6] with very low MAPE (Mean Average Percentage Error) = 2.75 and RMSE (Root Mean Square Error) = 5.43.

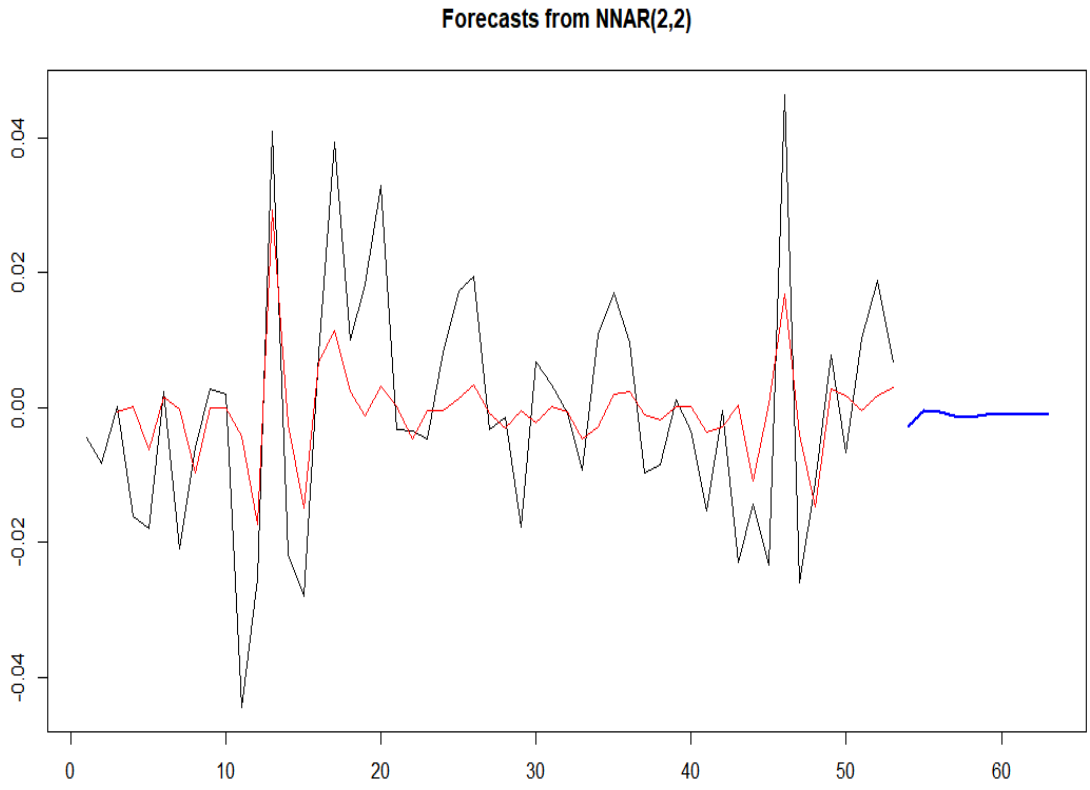


Fig.4.6 Actual Returns Vs Forecasted Returns for Apple Inc. from Ensemble of Neural Network and ARIMA

The model predicted the Closing price for next 50 days along with the logarithmic Daily change for any number of days. The results obtained predicted the market direction to gradually decrease which was verified on the NASDAQ and forecasted Closing Price for 9th April was 172.01 which was very close to the actual value of 170.05 [Fig.4.7] which was very near to the actual value. The drift in the forecast of 9th April is of magnitude 1.96. These drift boundary values with probabilities were calculated through Random Walk.

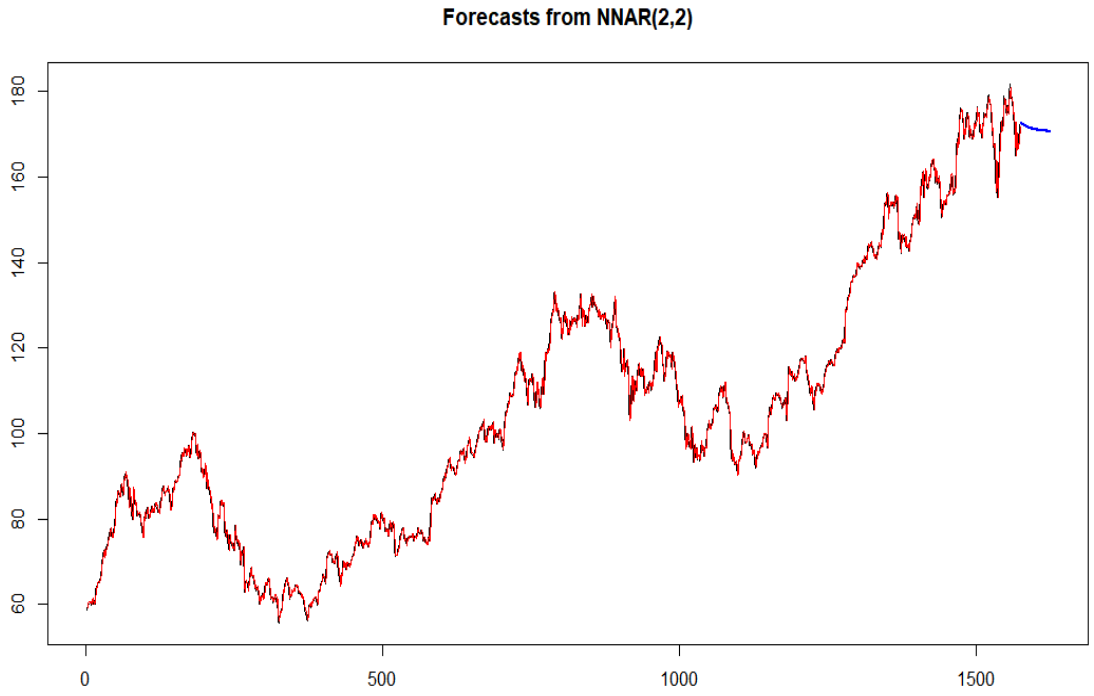


Fig.4.7 Closing Price of Next 50 days for Apple Inc. forecasted from the Ensemble Model

4.2.3. Random Walk Drift predictions

The drift boundary values were found from past 40 years data through Map-Reduce on Hadoop of Daily Change. The boundary values indicated a deviation of -9.31% to +9.91% [Fig.4.8] to be most frequent for 9th April and the magnitude of 1.96 is well within the interval [-9.31%,+9.91%] of 170.05 value of closing price.

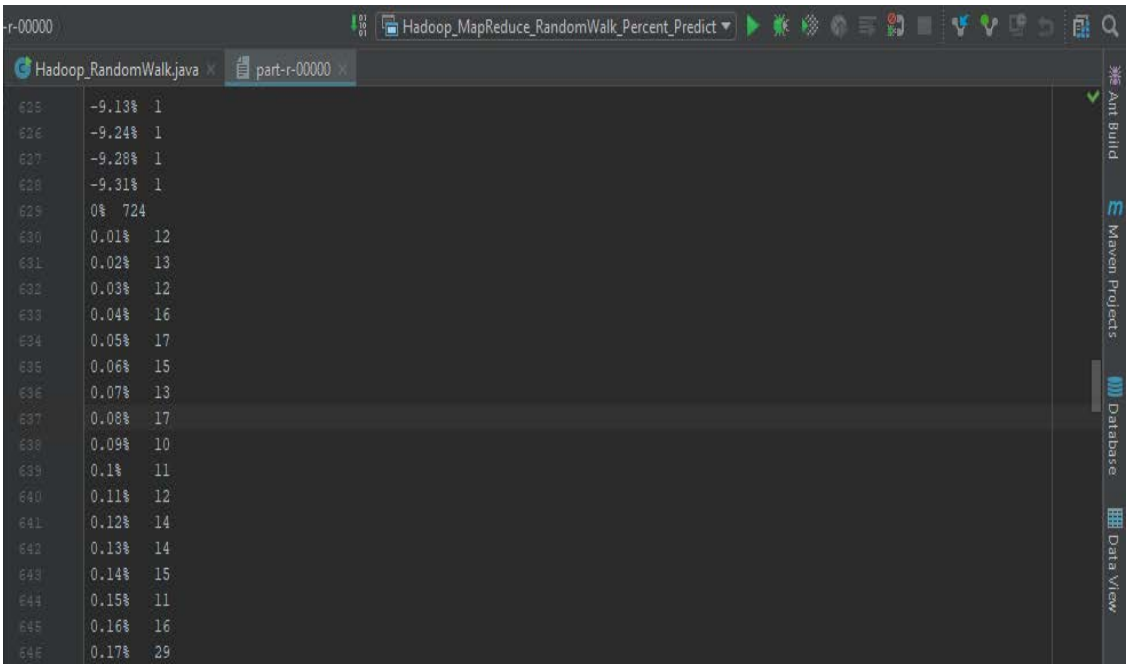


Fig 4.8 Frequencies of probabilistic Daily Drift calculated from Random Walk

So, the forecasted result from the Ensemble Model was 172.01 with -9.31% to +9.91% deviation for Apple Inc.'s Closing Price of 9th April.

CHAPTER 5

In this proposed project, an attempt is made to develop a prediction and forecasting model for finding the future stock market movements and their values based on the sentiment analysis of opinions and emotions expressed on Twitter feeds for a technology industry, like Apple Inc., using opinion mining and time series analysis using historical stock market data. The predicted outputs show the proposed model's potential to forecast the stock market movements for the short-term analysis of future, like for the next day, helping investors in their profitable investments in securities of stock markets and decisions related to buying/selling/holding a stock share, and thereby they can also contribute to advancements in technology through investing in the best Technology industry, and compete successfully with other emerging prediction and forecasting techniques. Other industry sectors like Healthcare could also be used for investments in saving lives.

5.1 Scope for Future Works

Other opportunities for improvements, which could be done on the proposed model, include using sentiments from News sources. Furthermore, it is worth mentioning that our analysis does not take into consideration many factors. First of all, our dataset does not really extract the real public sentiment, it only considers the twitter using English speaking people. Also, outliers and robots can be identified by validating the people responsible for the sentiments from the past week to be humans or not.

REFERENCES

[1] Rao, Tushar, and Saket Srivastava. “Analyzing stock market movements using twitter sentiment analysis.” In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp. 119-123. IEEE Computer Society, August 2012.

[2] Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." Journal of computational science 2, no. 1 (2011): 1-8.

[3] Makrehchi, Masoud, Sameena Shah, and Wenhui Liao. "Stock prediction using event-based sentiment analysis." In Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, vol. 1, pp. 337-342. IEEE, December 2013.

[4] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford 1, no. 12 (2009).

[5] Liu, Bing, Minqing Hu, and Junsheng Cheng. "Opinion observer: analyzing and comparing opinions on the web." In Proceedings of the 14th international conference on World Wide Web (WWW-2005), pp. 342-351. ACM, May 10-14, 2005.

[6] Sahni, Tapan, Chinmay Chandak, Naveen Reddy Chedeti, and Manish Singh. "Efficient Twitter sentiment classification using subjective distant supervision." In Communication Systems and Networks (COMSNETS), 2017 9th International Conference on, pp. 548-553. IEEE, June 2017.

[7] Angadi, Mahantesh C., and Amogh P. Kulkarni. "Time Series Data Analysis For Stock Market Prediction Using Data Mining Techniques With R." International Journal of Advanced Research in Computer Science 6, no. 6 (2015), August 2015.

[8] Chen, Ray, and Marius Lazer. "Sentiment analysis of twitter feeds for the prediction of stock market movement." stanford. edu. Retrieved January 25 (2013), Cs 229, pp. 15, 2013.

[9] Bing, Li, Keith CC Chan, and Carol Ou. "Public sentiment analysis in Twitter data for prediction of a company's stock price movements." In e-Business Engineering (ICEBE), 2014 IEEE 11th International Conference on, pp. 232-239. IEEE, 2014.

[10] Zhang, Linhao. "Sentiment analysis on Twitter with stock price and significant keyword correlation." Ph.D. dissertation, pp. 130, 2013.

[11] Dickinson, Brian, and Wei Hu. "Sentiment analysis of investor opinions on twitter." Social Networking 4, no. 03 (2015): 62..

[12] Mittal, Anshul, and Arpit Goel. "Stock prediction using twitter sentiment analysis." Standford University, CS229 (2011 http://cs229.stanford. edu/proj2011/GoelMittalStockMarketPredictionUsingTwitterSentimentAnalysis. pdf) 15 (2012).

[13] List of Positive Words - Positive Words Research, http://positivewordsresearch.com/list-of-positive-words/.

[17] Bouktif, Salah, and Mamoun Adel Awad. "Ant colony based approach to predict stock market movement from mood collected on Twitter." In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pp. 837-845. IEEE, 2013.

[18] Gunduz, Hakan, Zehra Cataltepe, and Yusuf Yaslan. "Stock market direction prediction using deep neural networks." In Signal Processing and Communications Applications Conference (SIU), 2017 25th, pp. 1-4. IEEE, 2017.

[19] Bautu, Elena, Andrei Bautu, and Henri Luchian. "Evolving gene expression programming classifiers for ensemble prediction of movements on the stock market." In Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, pp. 108-115. IEEE, 2010.

[20] Nelson, David MQ, Adriano CM Pereira, and Renato A. de Oliveira. "Stock market's price movement prediction with LSTM neural networks." In Neural Networks (IJCNN), 2017 International Joint Conference on, pp. 1419-1426. IEEE, 2017.

[21] A. Pak and P. Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Lrec, pages 1320–1326, 2010.

[22] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

Historical Data

[14] NASDAQ Stock Quote for Apple Inc, https://www.nasdaq.com/symbol/aapl.

[15] Google Finance Apple Inc.'s NYSE Quote, http://www.google.com/finance?q=NYSE:AAPL .

[16] Yahoo! Finance Stock Details for Apple Inc https://finance.yahoo.com/q?s=AAPL.