# Predicting Stock market trend from twitter feed and Building a framework for Bangladesh

Inspiring Excellence

*Authored by:*

Shabab Karim(14101138)

shababkarim93@gmail.com

Tahmid Abdullah(14101142)

sakib.9078@gmail.com

Umme Tayaba(14101034)

tayebasadna8@gmail.com

*Supervised by:*

Dr. Mahbub Alam Majumdar

Professor, Department of Computer Science and Engineering

BRAC University

May 6, 2018

# Declaration

We do hereby declare that the thesis titled “Predicting Stock market trend form twitter feed and Building a framework for Bangladesh” is submitted to the Department of Computer Science and Engineering of BRAC University in partial fulfillment of the completion of Bachelors of Science in Computer Science and Engineering. We hereby declare that this thesis is based on results obtained from our own work. Due acknowledgement has been made in the text to all other materials used. This thesis, neither in whole nor in part has been previously submitted to any University or Institute for the award of any degree or diploma and sources are properly acknowledged and mentioned by reference.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Dr. Mahbub Alam Majumdar
(Supervisor)

______________________________ ______________________________

Shabab Karim(14101138) Tahmid Abdullah(14101142)

______________________________

Umme Tayaba(14101034)

# Acknowledgement

First and foremost, we express our solemn to almighty Allah, the merciful, who has given us the ability, strength and opportunity to perform this thesis work.

We are grateful to our supervisor, Dr. Mahbub Alam Majumdar, for his immense support and valuable ideas throughout the work. He made us to get out of our comfort zones and to push the limit. Without his guidance, it would have been impossible to get introduced to the research prospect of Machine Learning.

We want to thank our parents for their immense support and believe throughout, to reach our goal for this research.

Lastly, we would like to express our gratitude to Department of Computer Science and Engineering, BRAC University and our teachers for helping us with all the necessary support.

# Abstract

*Social media has become an integral part in our day to day lives. What we share in these media are what we believe in and give others a window of opportunity to predict what is going on in our mind or how our actions will be in the future. Twitter is amazing at this job because usually people tend to write exactly what they are thinking when the character limit is only hundred and forty characters. This is particularly helpful when we want to analyze the trends of stock market. In this thesis paper, we tried to come up with a solution to better predict stock market trends analyzing the sentiments from twitter feeds obtained from StockTwits.*

**Keywords:** Stock Market, Sentiment Analysis, Classifier, Regression, Machine Learning, logistic regression, tweets.

# Contents

# Chapter 1

# Introduction

## 1.1 Human Behaviour

Our ability to learn and to get better task through experiences is a part of being a human. [8] When we are born we don't know anything but as days go by we become more capable of doing new things every day. But it is surprising to imagine that even computers could do the same. Just as our brain tells us to complete a task, so do computers. Suppose someone is asked to describe the differences between an Elephant and a lion we could begin by differentiating them in terms of their characters and thus recognizing them as different creatures. A computer program to learn and see the statistical pattern between two creatures that will enable it to recognize whether it is an elephant or a lion. That might figure out that elephants have bigger ears and lion has smaller and every information representing them numerically organizing in bit space. But its computer not the program that identifies that pattern and establishes the algorithm by which the data will be sorted. One example of a simple yet highly effective algorithm is to finding out lines separating elephants and lions; when the computer sees a new picture it changes the direction of the line and then says it's either an elephant or a lion. The more data a computer receives the more accurate it can be in its predictions. The world is full of data like music, pictures, clips, spreadsheets and it does not look like it is going to slow down anytime soon.

## 1.2 Machine Learning

Machine learning brings the promise of deriving meaning from all of that data. Machine learning is an important and advanced topic in the study of Computer Science which is closely related to Artificial Intelligence. Machine learning and AI can be interchangeably used. There is a lot of data now-a-days generated not only by people also by technologies like phone computers and other devices. This study is necessary for implementing important and complex functions in a more linear and flexible way other than programming or without being programmed. But how is this possible? The one way to think about this is to think about the differences between how will we normally program and what things we would like to get from a machine learning algorithm. In normal programming I am going to give the computer the data and it gives the output but in Machine learning the idea is I am going to give computer the output, by giving examples of what I want the program to do, levels of data, characterization of different classes of things. What a computer can do is by the given classes and different characterization of data the machine learning can actually produce a program the way I want. A program that I can use to infer various information. The problems are solved by Machine learning through mathematical calculations. This is extremely useful for predicting the condition of future outputs by determining the current and past states statistically. In order to predict these thoroughly, machine learning uses elaborated and composite formulas and also algorithms through which analysts, developers and engineers can make correct decision of advancement about the state of a particular data. It is a branch of Artificial Intelligence that automatically improves programs using data. For an instance tagging people or objects by their names on Facebook or any photos can be made by the study of machine learning. Of course perhaps the biggest example of all is Google search. We are using a system that has many machine learning system as required. For example-data mining is one sort of machine learning task. This system could be trained in an email messages to distinguish between spammed and non-spammed massages and after learning it can be used to classify new email messages into spam and non-spam folders. But it is not always correct. Sometimes good emails or inbox massages can get into spam folders or spammed messages can be mixed up in inbox. But this is an example or a task of a very small level. But now things are changing. Now what we are seeing in our day to day life is the advanced level of machine learning. If we talk about machine learning at a broad scale there are couple of components that we need to be worried about.

## 1.3 Applications of Machine Learning

There are different fields where machine learning is frequently used like medical diagnosis, brain-machine interfaces, self-driving cars, stock market Analysis, recommendation Engine using Machine learning can be used to recommend the right product for the right customer. Overtime as more data is used by the algorithm the performance in the system improves. It can also be used for the technology behind facial recognition, text and speech recognition, online shopping with reviewing recommendations, credit card number detection and so much more. When we are on deep learning and have access to better data we see dramatic breakthroughs. For example we recently launched Google translator using machine translation and deep learning systems the translation quality has improved dramatically than what it was in the past ten years. So the ability for computers to do these kind of tasks be it voice recognition, speech recognition, nature recognition is getting to a new point using machine learning. Advancing in machine learning will make big differences in many fields. Recently the study has been established to help diagnose diabetic retinopathy and its condition that causes blindness using machine learning. If we detected it earlier we could completely cure it, otherwise it causes blindness and also the fastest growing blindness in the world. Today we need advanced software knowledge to detect these conditions. But using machine learning we can detect it very accurately than a regular doctor using instruments. So there are kinds of changes to happen when we apply machine learning to all kinds of fields.

# Chapter 2

# Literature review

Just like a doctor identifies a disease by first examining and understanding the core of human biology, we as researchers must target the domain we are researching on. For that we found Hasbrook, Sofianos and Sosebee's[1] paper on New York Stock Exchange Systems and Trading Procedures to be quite helpful in understanding the ins and outs of the New York Stock Exchange.

Stock market has been subject of many scrutiny by scholars since its inception on 8th of March, 1817. Early works include papers from Jennings[2] which touches on different class struggles that people had to deal with the government and the big fishes in order to find the "American Dream". Since the early days it seems that the stock market has been volatile according to Lockwood and Linn[3], where they concluded that the volatility falls from the opening hour until early afternoon and rises thereafter and is significantly greater for intra-day versus overnight periods. Market variance is also shown to change significantly over time, rising after NASDAQ began in 1971.

With the usage of computers becoming ever more present[4] and with its huge computational power gave us a gateway to greater opportunities to bridge the gap between smart investments and sheer luck. Researchers from Goldman Sachs have always tried to predict the market from historical data but as we have seen in recent stock market crashes, using that approach is flawed because it does not take into account the sentiment of the public, after all it is emotions and feeling of trust that drives the market.

With this in mind we are starting our work on predicting stock market prices based on the sentiments of the general public. The data will be mined from social media site ”Twitter”, which is micro-blogging website. Our work is mainly based on Serban, Gonzalez and Wu’s[4] work on predicting stock market changes using sentiment analysis on twitter feed. They collected tweets related to stock market and classified them in six categories, which are: Calm, Alert, Sure, Vital, Kind and Happy. Then they correlated the data according to DIJA and found significant similarities. In their paper they address questions such as: How should we analyze and interpret the sentiment of thousands of emotional tweets? What is the intrinsic relationship between emotional tweets and stock market? What class of models can we expect to perform well on such stock price and trading volume prediction across several stocks? What metrics are suitable for evaluating a social media based model for stock market prediction?

A similar approach was taken in [5] where the positive and negative mood of tweets on Twitter is analyzed and compared with stock market indices such as Dow Jones, S&P 500, and NASDAQ over a period of 5 months. They found that the number of positive tweets is much higher than that of negative ones, more than double on average. However, the mood indicators (both positive and negative) proved to be always negatively correlated with DJIA, NASDAQ and S&P500.

Another paper published by Mittal and Goel[6] tried to find the correlation between ”public sentiment” and ”market sentiment”. Their paper shows that they used Self Organizing Fuzzy Neural Networks [10] to predict DJIA values using its closing 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).

We will also be taking help from Bollen et al[7]. The raw DJIA values are first fed into the preprocessor to obtain the processed values. At the same time, the tweets are fed to the sentiment analysis algorithm which outputs mood values for the four mood classes for each day. These moods and the processed DJIA values are then fed to our model learning framework which uses SOFNN to learn a model to predict future DJIA values using them.

Ali Harb et al. [27] used a different approach which they called AMOD (Automatic Mining of Opinion Dictionaries) where they extracted positive and negative adjectives to identify positive or negative opinions.

In another approach Vivek Sehgal and Charles Song [8] developed a new measure known as ”Trust Value” which assigns trust to each message based on its author. They used “TrustValue” to learn the relationship between sentiments.

Peter D.Turney [9] worked on an unsupervised classification algorithm called PMI_IR (Turney 2001), using the association of adjectives in reviews.

Sida Wang and Christopher D. Manning [10] observed that the performance of Naïve Bayes, SVM varies with bigram feature, length of documents without using stop-words, lexicon.

Authors Jasmina Smailović, Miha GrčarNada Lavrač, Martin Žnidaršič [32] in their paper, Predictive Sentiment Analysis of Tweets: A Stock Market Application, used positive sentiment probability as a new indicator to be used in predictive sentiment analysis in finance. They used Support vector machine(SVM) mechanism to classify tweets into 3 different categories to improve the classifiers in the stock market prediction.

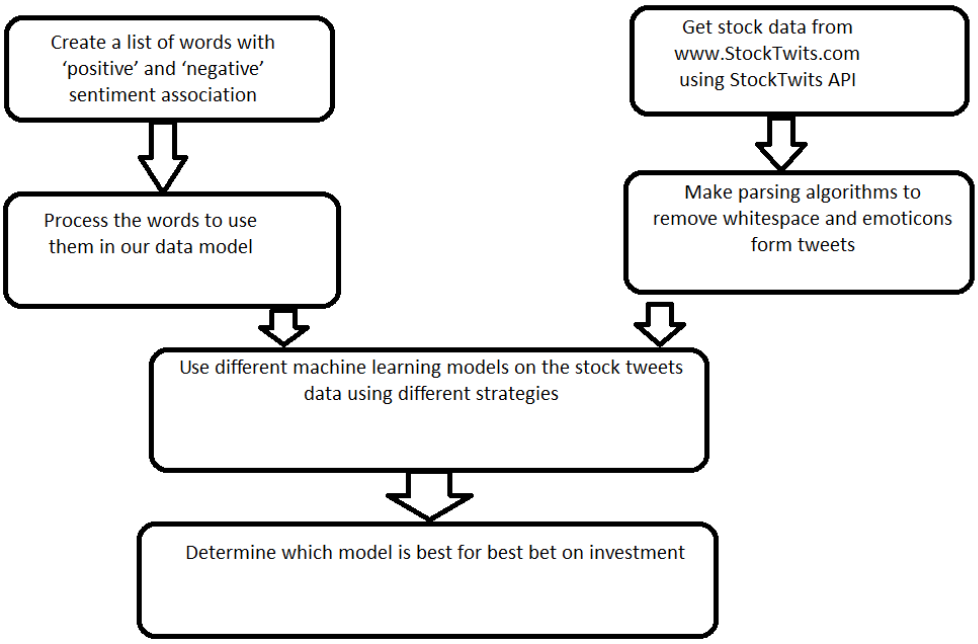
In the paper, Using sentiment analysis for stock exchange prediction [33], writers, Milson L. Lima, Thiago P. Nascimento, Sofiane Labidi, Nadson S. Timbó, Marcos V. L. Batista, Gilberto N. Neto, Eraldo A. M. Costa and Sonia R. S. Sousa, used natural language processing algorithms(LPN) to determine the collective mood of assets, as most of the mechanisms were based on statistical data. Later, with the help of SVM algorithm, they extracted the patterns in a attempt to predict the active behaviour of the investors.

# Chapter 3

# Work flow

## 3.1 Studying Machine Learning

We made a lot of use in machine learning libraries by studying Python. So this was something we should be fairly familiar with so that we don't get stuck in programming aspects. The second things is we should have a little bit of familiarity with statistics. We mostly assume that when we introduce our statistics concepts we are given some explanations or intuitions .But if we are taking something like descriptive or complex then we are right on point. There are some multiple steps of Machine learning. The first and foremost step is gathering data. This step is important because the quality and quantity we gather that can directly determine how good our productive model can be. In this case the data we collect can will yield us a table. And this is our training data. The next step of a machine learning is Data Preparation where we load our data into a suitable place and prepare for use in our machine learning training. We will first put our data together .Then randomize the ordering .We would not want the ordering of our data to affect how we learn just to determine the correct result. This is also good for a perfect visualization of our data and to help find the relevant relationships between different variables. Also shows if there is any data imbalance. We need to split the data into two categories as well. The first part will be a training model .The second part we will be using for evaluating our trained model performances. We don't want to use the same data that the model was trained on for evaluation. Sometimes the data we collect needs other form of adjusting and manipulation such as duplication, normalization, error collection .These are all included in data preparation step. The third step in the workflow is choosing a model.


Create a list of words with 'positive' and 'negative' sentiment association
Get stock data from www.StockTwits.com using StockTwits API
Process the words to use them in our data model
Make parsing algorithms to remove whitespace and emoticons form tweets
Use different machine learning models on the stock tweets data using different strategies
Determine which model is best for best bet on investment

## 3.2 Working with the dataset

Evaluation allows us to test the model against the data that has never been used for the training. This metric allows us to see how the model perform against data that has not yet seen. This is meant to be representing how the model could perform in the real world. Once we are done with Evaluation it is possible that we would like to see if we can further approve our training in any way. We can do this by tuning some parameters .One example of parameter we can tune is how many times we can run through the training set during training showing the data multiple times. By doing that we will potentially lead to higher accuracies .Another parameter defines how far we shift the line during each step based on the information from the previous training step. These values play the role of how accurate and perfect the model can become and how long the training takes. For more difficult models initial conditions can play a crucial role as well in determining the result of the training. We can also see the differences depending on whether a model starts off training with initial values 0's versus some distribution of those values. As we can see there are many consideration in the phase of training and we will determine what makes a model good for us. These parameters can be called as hyper parameters. The adjustment or tuning of these hyper parameters are complex. After we are done with training parameters or hyper parameters with the help of Evaluation part it is now the step to do something that comes of some use. Machine learning is using data to answer questions. So prediction or inference is that step where we finally getting our answers to the questions. By combining all these steps finally we can use machine learning to predict more thoroughly rather than human judgment. Now let's come to our topic and how it is related to machine learning. And to begin we are going to draw the attention to an incident. The day our famous singer of the history Michael Jackson died people started mourning by tweeting on twitter networking site, some gave status with grief on Facebook walls. We humans seeing those posts or tweets could understand that something bad has happened using our human brains ad intelligence. But this is also possible for computers to predict and assume the same like humans even more accurately.

We also took help from Bollen et al [4]. The raw DJIA values are first fed into the preprocessor to obtain the processed values. At the same time, the tweets are fed to the sentiment analysis algorithm which outputs mood values for the four mood classes for each day. These moods and the processed DJIA values are then fed to our model learning framework which uses SOFNN to learn a model to predict future DJIA values using them.

## 3.3 Understanding the Sentiment

This is where our topic comes. By stock market we see the tweets and bring an overall outline of whether the tweet is positive, neutral or negative. We divide the tweet between these three divisions. In this case we are using some special keywords like adjectives to determine the type of the tweet. And this data is collected every day following the seven steps of machine learning which is typically denoted as data gathering as we discussed earlier. In this step we are putting all the tweets from different decisions in a table. Then we randomize the ordering of the tweets to see the approximate relationships between those tweets. After gathering the tweets we go into the next step data preparation where some sort of editing, normalization process happens. And then after gathering the data every day we give a rating to each of the divisions. Which is actually denoted by data training. In this step we will be using a graph with appropriate equations according to the result of those tweets in the division and there will be many slops I the graph as the features are many and we are collecting the tweets every day. By gathering all the slopes of possible tweets we will gather them into Weight Matrix and the y intercepts into Biases matrix. In the training process the tweets arranged by the randomized orders are explicitly seen and analyzing that the outputs are determined that are supposed to be made. And this step repeats every time with each iteration in a cyclic order which is an important step of Machine learning. Gradually by collecting more data and making more graphs and matrixes we will be getting more and more training sets and will be able to see the line of each training set are shifting and changing the direction which particularly reflects the ideal separation between lines which says the tweets posted daily are different from each other. After completing this crucial step of training it's time for evaluation all those tweets positive, negative or neutral closely and send them for Evaluation process. By evaluation we will be testing whether our assumption about the tweets between those divisions are right or wrong. And then we will be looking for little more better performance of our training about stock market tweets. Every rating will be rounded up to 1. Through machine learning we will be able to find out the specific importance to our variables. And at last we will be making the predictions seeing the tweets. Twitter data influences stock market in many ways. In order to explain this we will draw up an example of a drastic incident that happened in early 2008 on 16th September when the stock-market crashed in America for which a massive downfall of stock prices occurred which caused great loss of economy. And as a result the markets were forced to shut down for a short time. But before the crash took place the experts already assumed such thing might happen by analyzing the conditions of the stock market and the prices and the reviews as well. But no one paid any heed to it. After the crash they felt the consequences .They had to face a great crisis and the larger banks has been bankrupted. If there was such model to figure out the consequences of the stock-market then we could have restricted such calamity to happen. Using stock market analysis we could have gathered all the information through twitter data and then go through the machine learning parts step by step. These would have been easier for us to predict whether the stock market will collapse or not and based on that we could take the decision that the market should shut down automatically in order to defend the financial losses. There are different stock markets in the world and each and every one is very competitive to others. That's why adjusting in this market is difficult to make profits and stay in the competition. Imagining the ability to know what the stock-market is going to do before it was using the power of twitter data. In general we are very interested in the notion of collecting mood states like the society can by crying one day, smiling one day, happy the other day. So we developed a methodology to measure those mood stated from people's tweets on how they formulate a particular tweets from words we will be validating our results by looking at whether they matched with the people from other areas. And thus we started co-relating different tweets on different stages and finding out the relations. We made the fluctuations of the mood state and we compared the fluctuations and then finding out the significant co-relations. So what the algorithm does is the computer rate these tweets which does not happen manually and the computer reads through millions of tweets looking at those dramatic patterns, particular terms of use, combinations of those terms and for that entire day they build the picture of whether the average result is particularly positive or negative. And these dimensions can be used to predict whether the market will go up or down quite a few days later. The most basic issue is understanding what we are using these correlations for. A lot of people in the financial industry are unwilling to use something that they fundamentally don't understand. And we don't particularly understand how the fluctuations in the twitter mood state can co-relate with the fluctuations of twitter price. So to understand that kind of connections in the development model of why we are making such predictions these co-relations are very important to us. There are millions of tweeter data published every day on the network now so there is a huge amount of information. Through twitter data we can easily highlight the importance of ongoing tweeting activity and also to find out the relationships through the analysis. [17] For example someone tweeted "I didn't like the offer at all". The stock market will analyze such reviews first, gather some more tweets and then find out the similarities between them. And then it will be collecting more for a particular period of time say 6 months with 50000 tweets let's say and assigning them in a training set starting from parameters -50 to +50, whereas -50 denotes a highly negative tweet and +50 denotes a highly positive tweet. These can add up some more parameters in between like negative, neutral, slightly negative, slightly positive, positive. And the decision will be taken based on the weight of each parameter. So if the negative or highly negative parameter raises on top of other parameters then the prediction will be made that there are maximum chances for stock-market to fall. If the neutral level rises high then the prediction can be made that there are chances partly either the stock market will fall or will not. Similarly if the parameter slightly positive or positive or extremely positive rises high then we can come to the prediction that the stock-market is making profits and so it will in the near future. We will be opting for three basic different models in order to predict the price in stock market and after that we will be differentiating the outputs and then determine which one gives the best predictive ability for future prices. There is a policy of creating and investment and we have to go deep into it. After we already have decided to make an investment we are likely to find out how many companies there are and which one among all will assumedly would give us good outputs. Then we decide to take that company in order to make profits. In order to do so we will be keeping track of all the past records ,the past trade-off ,designs and the reputation records made by that company, also the new innovations by analyzing its overall feature. Then we will observe how stock prices have changed overtime looking into the records of specific periods. Also by looking into what other investors or experts or analysts are saying or predicting about the future of this company seeing their tweets and reviews. Finally, we will gather all these data to make prediction about the future price. Good traders will use predictive models while deciding where to invest.

# Chapter 4

# Data handling

## 4.1 Data collection

Our data consists of mainly daily tweets about the stock market. Initially we are going to try to analyze the NYSE then upon perfection we will be using our model with the most accuracy to data from Bangladesh and implement a system to help investors of Bangladesh. We started off with finding a tagged data, unfortunately there were no free sources to find such. What we did then was built our own web script to scrap tweets from the Twitter API.

We run this script every day for 8 hours and we collect a huge amount of tweet daily. Then we store it in JavaScript Object Notation (JSON) to our file storage and parse it. The format looks like this:

```
{"text": "Are you prepared for the crash of the Stock
   ↪ Market? Then listen to Final Battle https://t.co/
   ↪ fYGkS8FoC4 #MusicLyrics", "created_at": "Sun Oct 01
   ↪ 12:25:59 +0000 2017"}
{"text": "RT @NYSE: Without looking it up, tell us what
   ↪ year this company went public on the NYSE #
   ↪ StockCertificateSunday https://t.co/V3TQAv0LjP", "
   ↪ created_at": "Sun Oct 01 12:26:02 +0000 2017"}
{"text": "Kindest Quarter Arrives for Stock Market That
   ↪ Nothing Can Rattle https://t.co/90CvgCivdD via
   ↪ @bllshbrsh", "created_at": "Sun Oct 01 12:26:07 +0000
   ↪  2017"}
{"text": "Have you considered investing outside the stock
   ↪ market? https://t.co/ ofsMI8pdwB", "created_at": "Sun
   ↪  Oct 01 12:26:07 +0000 2017"}
{"text": "Will You Be Ready When the Stock Market Crashes
   ↪ Again? - The Wall Street Journal https://t.co/3k7Bai5
   ↪ crQ", "created_at": "Sun Oct 01 12:26:14 +0000 2017"}
{"text": "RT @marketstocknews: Dow Jones Industrial Average
   ↪  turns positive in midday trading https://t.co/
   ↪ vimzQlBUfh", "created_at": "Sun Oct 01 12:27:02 +0000
   ↪  2017"}
{"text": "RT @ZyiteGadgets: Dow Jones Industrial Average
   ↪ turns positive in midday trading https://t.co/zKGfa5
   ↪ LGm9", "created_at": "Sun Oct 01 12:27:03 +0000
   ↪ 2017"}
```

As we can see that it consists of text body of the tweet and meta-data such as timestamp. The next job we will be doing is to score the tweet data.

## 4.2 Data pre-processing

As we can see on figure 4A.1 that the tweet texts consists a lot of inconsistency and noisy words. These noisy words will interfere with our learning algorithm. Also the biasness seems to increase as a result of including these words. If we include these words then our learning algorithm seems to look for quantifiers like “a”, “an”, “this” etc. Also auxiliary verbs seems to be of no interest to us. So we need only words that give a sense of “good”, “bad” or “neutrality”. The following list of things we had to do in order to pre-process our data:

1. Delete all hyper-text links from the tweet data. For example: “http://” or “https://t.co/3k7Bai5crQ” are removed from the tweet feed.

2. Change all word blocks from the tweet data to lower case. This increases

uniformity and changes helps us to remove repetitions if present.

3. Removing white spaces from the tweet data. We keep the emoticons because they provide helpful insights about the tweet.
4. We remove punctuations marks like commas, full stop etc.
5. Clear out any tag to any person in the tweet data. Tags starting with “@” are removed. We keep hashtags because sometimes tags like “#Stock #Crash” helps us better understand the mood.
6. Do take retweets in our data sweets. So remove tweets with “RT” from the data set.

| **Tweet Data** | **Parsed Data** |
|---|---|
| @ZyiteGadgets: Dow Jones Industrial Average turns positive in midday trading https:// t.co/ zKGfa5LGm9. | dow jones industrial average turns positive midday trading |
| @RealDonaldTrump making America great again by increasing jobs in coal mines. | making america great again increasing jobs coal mines. |
| Kindest Quarter Arrives for Stock Market That Nothing Can Rattle https://t.co/90CvgCivdD via @bllshbrsh | kindest quarter arrives stock market nothing can rattle via |
| Will You Be Ready When the Stock Market Crashes Again? - The Wall Street Journal https://t.co/3k7Bai5crQ | will you ready when stock market crashes again wall street journal |

Table 4.1: actual tweets and parsed tweets

## 4.3 Data scoring

Our approach for scoring a tweet was simple and effective. The first problem with tweets in the JSON files were that they contain a lot of noisy words which have little to no significance to the actual context. We have to remove them to get words of interest to us. In light of this objective we first collected a list of positive, negative and neutral words in the dictionary. Then what we did was scored based on the unique positive, negative and neutral words on the tweets and our list of words. Suppose, the tweet consists of n words. Now considering the score for positive, negative and neutral score be Scorepos, Scoreneg and Scoreneu respectively and notate the set of all positive, negative and neutral words as listpos, listneg and listneu and frequency of positive, negative and neutral words as frequencypos, frequencyneg and frequencyneu respectively we come up with the following formula for scoring the data:

$$score_{pos} = \frac{\sum_1^n Frequency_{pos}}{\sum_1^n Frequency_{pos} + \sum_1^n Frequency_{neg} + \sum_1^n Frequency_{neu}}$$

$$score_{neg} = \frac{\sum_1^n Frequency_{neg}}{\sum_1^n Frequency_{pos} + \sum_1^n Frequency_{neg} + \sum_1^n Frequency_{neu}}$$

$$score_{neu} = \frac{\sum_1^n Frequency_{neu}}{\sum_1^n Frequency_{pos} + \sum_1^n Frequency_{neg} + \sum_1^n Frequency_{neu}}$$

Using this simple set of formula we are able to score our tweets. A sample of the scoring is given below.

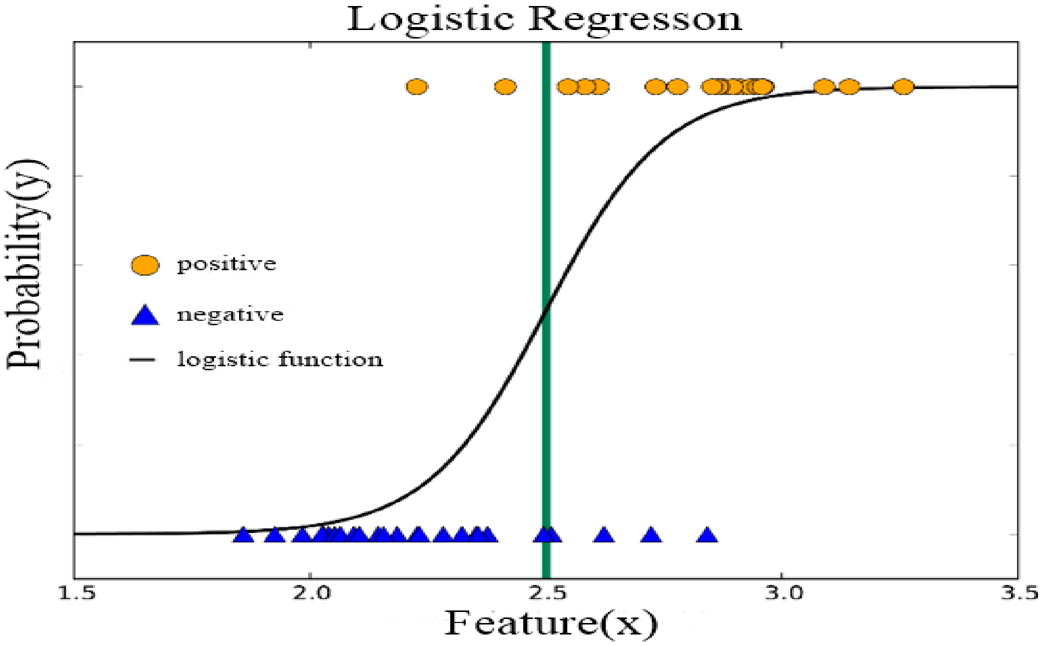
| **Tweet number** | **Positive score** | **Negative score** | **Neutral score** |
|---|---|---|---|
| Tweet A | 0.1 | 0.2 | 0.7 |
| Tweet B | 0.5 | 0.1 | 0.4 |
| Tweet C | 0.0 | 1.0 | 0.0 |

Table 4.2: a sample scoring table of data

# Chapter 5

# Logistic regression

Logistic regression [9] is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). In logistic regression, the dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, etc.) or 0 (FALSE, failure, etc.).



The main goal of logistic regression is to find the best fitted model to explain the relationship between the dependent variable and a set of independent variable. This regression model is also known as binary logistic model as the outcome is 1 or 0. It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor. To illustrate logistic regression let us look at a diagram. In the given diagram, it is illustrated how to train a 1-dimensional classifier. Here, the training data is divided into two segments, positive (in orange circles) and negative (in blue triangle), where a positive score means a probability of 1 and negative means probability of 0. The Black line represents the decision boundary of the logistic regression which separates the data into two classes.

The logistic regression can be understood simply as finding the $\beta$ parameters that best fit:

$$y = \begin{cases} 1, & \text{if } \beta_0 + \beta_1 x + \epsilon. \\ 0, & \text{otherwise.} \end{cases}$$

Where, $\epsilon$ is an error distributed by the standard logistic regression. Logistic regression is named at the core of the method logistic function. Logistic function is S-shaped curve that can take any real value number and then, map it to a value that is between 0 and 1. But never exactly the limit values. The logistic function $\sigma(t)$ is defined as follows:

$$\sigma(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t}$$

Where $e$ is the base of the natural logarithm. And $t$ is the actual numeric value that has to be transformed. Let us assume that $t$ is a linear function of a single explanatory variable $x$. Then $t$ can be written as follows:

$$t = \beta_0 + \beta_1 x$$

And the [12] logistic function can be written as:

$$y = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

Where $y$ is the predicted value and $\beta_0$ is the bias and $\beta_1$ is the co-efficient for the single input value $x$. Now, given a set of inputs, $x_i$ and a label $y_i \in 1, 0$, logistic regression interprets the probability that the label is in one class as a logistic function of a linear combination [11] of features:

$$p(y_i = 1|x) = \frac{1}{1+e(-\theta^T x)}$$

A column of 1's is added with the features $x_i$, similar to linear regression. This probability function has to be transformed to binary values of 0 and 1 and this transformation is done using the logistic function.

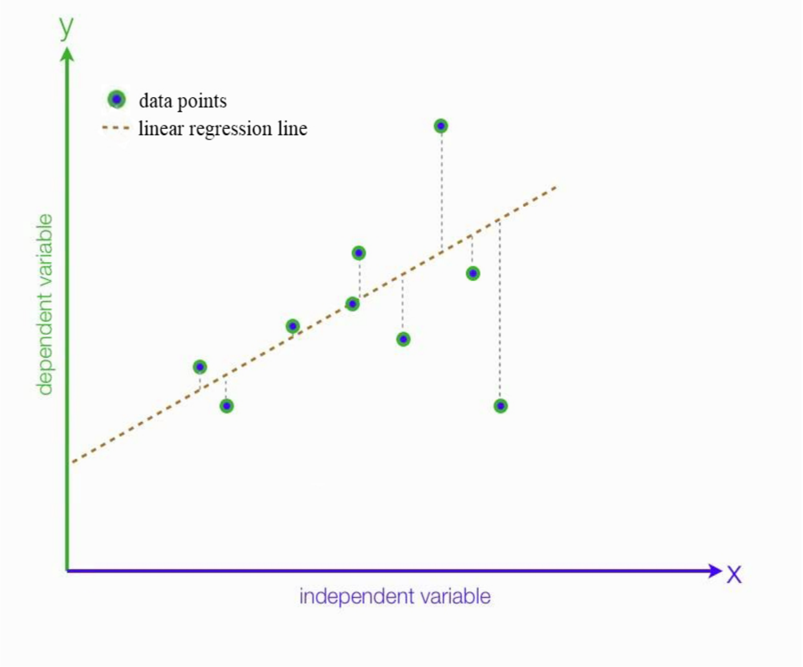
Since we have understood the concepts we will be using python libraries to implement this model. With time we will work on tweaking this algorithm further if need. Right now, we are in a phase of analyzing different models. Our python implementation is as follows:

```
feature_column_names = [’pos’, ’neg’, ’neu’]
predicted_class_name = [’change’]
file_list = (gb.glob("final/*.pkl")) #getting all pickles
   ↪ file list from folder
file = file_list[0]
X = []
Y = []
with open(file, ’rb’) as pick:
   final_list = pickle.load(pick)
   X = np.array([ [obj[’pos’], obj[’neg’], obj[’neu’] ] for
      ↪ obj in final_list])
   Y = np.array([ obj[’change’] for obj in final_list])
   h = .02
   train_sz = floor(len(Y) * (0.7));
   X_train = X[0:train_sz]
   X_test = X[train_sz:len(Y)]
   Y_train = Y[0:train_sz]
   Y_test = Y[train_sz:len(Y)]
   logreg = linear_model.LinearRegression()
   # we create an instance of Neighbours Classifier and fit
      ↪ the data.
   logreg.fit(X_train, Y_train)
   Z = logreg.predict(X_test)
   print("Mean␣squared␣error:␣%.2f" % mean_squared_error(
      ↪ Y_test, Z))
   # Explained variance score: 1 is perfect prediction
   print(’Variance␣score:␣%.2f’ % r2_score(Y_test, Z))
```

# Chapter 6

# Linear regression

A linear approach that would model the relationship between a scalar dependent variable and one or more independent variables [13]. The independent variables are also known as, explanatory variables. If the number of independent variable is one, then it is known as simple linear regression whereas for multiple number of independent or explanatory variables, the model is called multiple linear regression.



It is the most basic form of regression. To illustrate simple regression model, let us take a look at the given figure, where x denotes the independent variable and y denotes the dependent variable. The green dots on the graph are the scattered plots of the data points for which there would occur changes to the value of dependent variable y based on value of x. The dotted line represents the regression line also known as best fitted line. [9] In simple linear regression the data set would be modelled as:

$$y = b_0 + b_1 x$$

Where $y$ is the predicted output and $x$ is the independent variable or explanatory variable. $b_0$ and $b_1$ are the coefficient that would move the regression line to be best fitted. As $b_0$ determines where the line intercepts the $y$-axis in the graph, it is call the intercept. $b_1$ Is called the slope as it determines the slope of the regression line. To estimate $b_1$ :

$$b_1 = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_1^n (x_i - \bar{x})}$$

Where $\bar{x}$ and $\bar{y}$ are the average of the respected variable values in the dataset. And we can find the estimated value of $b_0$ as follows:

$$b_0 = \bar{y} - b_1 \bar{x}$$

Now, for multiple linear regression model, the linear equation can be written as:

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_n x_n + \epsilon$$

Here, $n$ is the number of total independent variables. So the total data can be represented in a 2-d matrix and the system could be represented in $n$ equations in matrix notation as follows:

$$Y = XB + \epsilon$$

where,

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & x_{12} & \vdots & x_{1n} \\ 1 & x_{21} & x_{22} & \vdots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \vdots & x_{nn} \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Here, vector $B$ contains the regression co-efficient. $B$ is to be known to form the regression model. It is estimated using least square estimates and the equation is as follows:

$$\hat{b} = (X^T X)^{(-1)} X^T y$$

Where, $X^T$ refers to the transpose matrix of $X$ matrix and $-1$ is used to refer to the inverse matrix. With this, the multiple linear regression model can be estimated as:

$$\hat{y} = X\hat{b}$$

Just like we had done previously we would be looking for a solution that has already been built in for us instead of building our own work around the problem.

```
feature_column_names = [’pos’, ’neg’, ’neu’]
predicted_class_name = [’change’]
file_list = (gb.glob("final/*.pkl")) #getting all pickles
   ↪ file list from folder
file = file_list[0]
X = []
Y = []
with open(file, ’rb’) as pick:
   final_list = pickle.load(pick)
   X = np.array([ [obj[’pos’], obj[’neg’], obj[’neu’] ] for
      ↪  obj in final_list])
   Y = np.array([ obj[’change_round’] for obj in final_list
      ↪ ])
   h = .02 # step size in the mesh
   X_train = X[0:train_sz]
   X_test = X[train_sz:len(Y)]
   Y_train = Y[0:train_sz]
   Y_test = Y[train_sz:len(Y)]
   logreg = linear_model.LogisticRegression(C=1e5)
   # we create an instance of Neighbours Classifier and fit
      ↪  the data.
   logreg.fit(X_train, Y_train)
   Z = logreg.predict(X_test)
   # Put the result into a color plot
   print Z
   print("Mean␣squared␣error:␣%.2f%%" % ((1 -
      ↪ mean_squared_error(Y_test, Z)) * 100))
```

# Chapter 7

# Neural network

Neural network model is inspired from the structure of the brain. [7] As we know, our brain is just collection of cells known as neurons, certain parts of neurons light up when we see something familiar to us. This is just not limited to seeing but by all 6 senses, each has a group of neurons associated to it. These neurons are inter-connected and communicate with each other to find a common goal with the most accuracy.

Just like neurons in our brain we implement the neural network. Just like neurons in our brain we have a bunch of “nodes” connected to each other holding an absolute value from 0 to 1. [15] It is also known as a “connectionist” computer network because of its nodes having connection with each other and passing in values from one another.

$$f(x) = \begin{cases} 0, & \text{if } x < 0. \\ 1, & \text{otherwise.} \end{cases}$$

Then we look into the sigmoid function just like we had done previously in the previous learning models. [16] In modern times there is a prevalence of RELL function instead of sigmoid due to its slow nature but we will stick to the basics.

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Let’s also look at why we are going to use sigmoid function for neural networks. There is a neat property about sigmoid functions. The derivative of the sigmoid is represented as the function itself. This helps us to implement the back propagation needed for neural networks.

$$
\begin{aligned}
\sigma^{'}(z) &= \frac{d}{dz}(\frac{1}{1+e^{-z}}) \\
&= -\frac{1}{(1+e^{-z})^{-2}}\frac{d}{dz}(e^{-z}) \\
&= \frac{e^{-z}}{(1+e^{-z})}\frac{1}{(1+e^{-z})} \\
&= \frac{(e^{-z}+1-1)}{(1+e^{-z})}\frac{1}{(1+e^{-z})} \\
&= (\frac{(1+e^{-z})}{(1+e^{-z})} - \frac{1}{(1+e^{-z})})\frac{1}{(1+e^{-z})} \\
&= (1 - \frac{1}{(1+e^{-z})})\frac{1}{(1+e^{-z})} \\
&= (1-\sigma(z))\sigma(z)
\end{aligned}
$$

Before we go deep down this rabbit hole, we need to define “learning” in a neural network. Prior to that let us understand how we humans learn. [18] We usually start with a random assumption about something that we don’t know about. When we learn that something, we change our assumptions and modify our understanding accordingly. Neural networks learn in the same way and the parameter that is being learned is the weights of the various connections to a neuron. From the transfer function equation, we can observe that in order to achieve a needed output value $z$ for a given input $x$, the weight has to be changed. This is a very simple example to understand the obviousness. A simple signal flow in a neural network starts with giving the inputs to the input neurons and obtain an output. Based on the degree of deviation from the desired output, the weights inside the network are changed (in a defined way) to better fit the output. This is the actual learning that takes place inside a neural network. The layers we used in our neural network are described below:

- Input Layer: In the input layer we have nodes which match with the number of features that we have, which is 3. This the layer where the different values for the three different features will be given as an output.

- Output Layer: In this layer we get the output as a 1 or a 0. This signifies whether there will be an increase or decrease in prices of the stock market.

- Hidden Layers: In this layer is where the neural network does its magic. As per convention we take the number of hidden layers as the mean of the number of the features. We are taking ceil of the mean of the features, which equals to 2 hidden layers. In the first hidden layer we are using the sigmoid function as the activation function and bipolar step as the activation function in the second layer.

This is our implementation of neural networks.

```
feature_column_names = ['pos', 'neg', 'neu']
predicted_class_name = ['change']
file_list = (gb.glob("final/*.pkl")) #getting all pickles
   ↪ file list from folder
file = file_list[0]
X = []
Y = []
with open(file, 'rb') as pick:
    final_list = pickle.load(pick)
    X = np.array([ [obj['pos'], obj['neg'], obj['neu'] ] for
       ↪  obj in final_list])
    Y = np.array([ obj['change_round'] for obj in final_list
       ↪ ])
    h = .02 # step size in the mesh
    X_train = X[0:train_sz]
    X_test = X[train_sz:len(Y)]
    Y_train = Y[0:train_sz]
    Y_test = Y[train_sz:len(Y)]
    clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
       ↪ hidden_layer_sizes=(2,), random_state=1)
    clf.fit(X_train, Y_train)
    Z = clf.predict(X_test) # Put the result into a plot
    print Z
    print("Mean␣squared␣error:␣%.2f%%" % ((1 -
       ↪ mean_squared_error(Y_test, Z)) * 100))
```

# Chapter 8

# Gradient boosting decision tree

Later we realized that to get a better result we should try an ensembling technique for better results. We will ensemble Linear regression followed by Gradient Boosting on the data set. Gradient boosting is one of the latest advancements in modern data modeling and mining[13]. In gradient boosting we define the loss function as:

$$Loss = MSE = \sum (y_i - y_i^p)^2$$

where, $y_i$ is the ith target value, $y_i^p$ is the ith prediction and $L(y_i - y_i^p)$ is loss function. By using gradient descent we can decide which branch to expand which might not seem statistically correct but logically so. We can find our minimum squared error is minimum at:

$$\begin{aligned} y_i^p &= y_i^p + \alpha * \sigma \sum (y_i - y_i^p)^2 / \sigma y_i^p \\ &= y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p) \end{aligned}$$

So, we are basically updating the predictions such that the sum of our residuals is close to 0 (or minimum) and predicted values are sufficiently close to actual values. The algorithm used is given below:

**Algorithm** Gradient Boosting algorithm

```
Inputs:
    $\{(x_i, y_i)_{i=1}^{n}\}$, training set
    $L(y, F(x))$, differentiable loss function
    $M$, number of iterations
Initialize:
    $F_0(x) = arg_{\gamma} min \sum_{i=1}^{n} L(y_i, \gamma)$
for $m = 1$ to M do
    1. Compute pseudo-residuals:
    $r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{f(x)=f_{m-1}(x)}$ , for $i = 1, \ldots, n$
    2. Fit base leaner $h_m(x) topseudo - residuals, trainitusingtrainingset$
    3. Compute multiplier $\gamma_m$ by solving one-dimentional optimization problem:
    $\gamma_m = arg_{\gamma} min \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$
    Update the model:
    $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$
end for
Output $F_m(x)$
```

# Chapter 9

# Fuzzy K-Nearest neighbour

The idea is to find K similar data point from the training set and use them to interpolate the output value, which is either the majority value for categorical output, or average (or weighted average) for numeric output. K is a tunable parameter which needs to be cross-validated to pick the best value.

Decision theory using Bayes' model gives us a better error detection model. In those cases where this information is not present, many algorithms make use of distance or similarity among samples as a means of classification. The K-nearest neighbor decision rule has often been used in these pattern recognition problems. One of the difficulties that arises when utilizing this technique is that each of the labeled samples is given equal importance in deciding the class memberships of the pattern to be classified, regardless of their 'typicalness'[31]. The theory of fuzzy sets is introduced into the K-nearest neighbor technique to develop a fuzzy version of the algorithm.

Given a universe $U$ of objects, a conventional crisp subset $A$ of $U$ is commonly defined by specifying the objects of the universe that are members of $A$. An equivalent way of defining $A$ is to specify the characteristic function of $A$, $u_a : U \rightarrow \{0, 1\}$ where for all $x \in U$

$$u_a(x) = \begin{cases} 1, & \text{if } x \in A. \\ 0, & \text{if } x \notin A. \end{cases}$$

Fuzzy sets are derived by generalizing the concept of a characteristic function to a membership function $u : U \rightarrow [0, 1]$ .An example of a fuzzy set is the set of real numbers much larger than zero, which can be defined with a membership function as follows:

$$u(x) = \begin{cases} \frac{x^2}{x^2+1}, & \text{if } x \geq 0. \\ 0, & \text{otherwise.} \end{cases}$$

Numbers that are not at all larger than zero are not in the set ($U = 0$), while numbers which are larger than zero are partially in the set based on how much larger than zero they are. Thus the impetus behind the introduction of fuzzy set theory was to provide a means of defining categories that are inherently imprecise. Since the introduction of fuzzy set theory the terms hard and crisp have been used to describe sets conforming to traditional set theory.

Given a set of sample vectors, $x_1, x_2, \ldots, x_n$, a fuzzy $c$ partition of these vectors specifies the degree of membership of each vector in each of $c$ classes. It is denoted by the $c$ by $n$ matrix $U$, where $u_{ik} = u_i(x_k)$ for $i = 1, \ldots, c$ and $k = 1 \ldots n$ is the degree of membership of $x_k$ in class $i$. The following properties must be true for $U$ to be a fuzzy $c$ partition:

1. $\sum_{i=1}^{c} u_{ik} = 1$

2. $0 < \sum_{k=1}^{m} u_{ik} < n$

3. $u_{ik} \in [0, 1]$

Let $W = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ labeled samples. The crisp K-NN algorithm is as follows:

**Algorithm** The crisp $K$-$NN$ algorithm

```
Inputs:
    $y$, of unknown classification.
Set:
    $K, 1 \leq K \leq n$
Initialize:
    $i \leftarrow 1$
while ($K$-nearest neighbour found) do
    Compute distance from $y$ to $x_i$
    if ($i \leq K$) then
        include $x_i$ in the set of $K$-nearest neighbours
    else if ($x_i$ is closer to $y$ then any previous nearest neighbour) then
        Delete farthest in the set of $k$-nearest neighbours
        Include $x_i$ in the set of $k$-nearest neighbours
    end if
    increment $i$
end while
```

# Chapter 10

# Support vector machine

Recently, support vector machines (Vapnik, 1995; Vapknik, 1998a; Vapnik, 1998b) have been introduced for solving pattern recognition problems. In this method one maps the data into a higher dimensional input space and one constructs an optimal separating hyperplane in this space. This basically involves solving a quadratic programming problem, while gradient based training methods for neural network architectures on the other hand suffer from the existence of many local minima.

Besides the linear case, SVM's based on polynomials, splines, radial basis function networks and multilayer perceptrons have been successfully applied. Being based on the structural risk minimization principle and capacity concept with pure combinatorial definitions, the quality and complexity of the SVM solution does not depend directly on the dimensionality of the input space.

Given a training set of $N$ data points $\{y_k, x_k\}_{k=1}^{N}$, where $x_k \in \mathbb{R}^n$ is the $k$th input pattern and $y_k \in \mathbb{R}$ is the $k$th output pattern, the spport vector method approach aims at constructing a classifier of the form:

$$y(x) = [\sum_{k=1}^{N} \alpha_k y_k \Psi(x, x_k) + b]$$

Here we introduce a least squares version to the SVM classifiers by formulating the classification problem as:

$$\min_{x,b,e} g_3(w, b, e) = \frac{1}{2} w^t w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

subject to the equality constrains

$$y_k[w^t \varphi(x_k) + b] = 1 - e_k, k = 1, \ldots, N.$$

One defines the Lagrangian

$$\mathcal{L}_3(w, b, e; \alpha) = g_3(w, b, e) - \sum_{k=1}^{N} \alpha_k \{y_k[w^T \varphi(x_k) + b] - 1 + e_k\}$$

where $\alpha_k$ are Lagrange multipliers(which can be either positive or negative now due to the equality constrains as follows from the Kuhn-Tucker conditions(Fletcher, 1987)).

The algoritm we used is as follows:

```
Algorithm SVM algorithm
  Initialize:
      n-dimensional hyperplane, where, n ← number of features
      learning rate
  Repeat until the closest point on all axes is furthest from the hyperplane:
  for all i in n do
      maximize $f(c_1 \ldots c_n) = \sum_1^n c^i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i c_i (x_i . x_j) y_j c_j$,
      subject to $\sum_{i=1}^n c_i y_i = 0$ and $0 \leq c_i \leq \frac{1}{2n\lambda}$
      Increase generation count
  end for
```

# Chapter 11

# Results and analysis

The experimental setup consists of two components. The first one is collecting data and scoring it. We collect data from Twitter feed and score it. The second component being the learning models. Some of the models are pretty simple and some cutting edge. Based on the learning models we have worked on previously we have come up with interesting results. We used rapidTable [21] and meta-charts [22] to make visual representations. We have used three learning models on data that we have collected from a dedicated machine for 3 months. We have divided seventy percent of it for training set and the rest 30% for test set. Based on these training set we have come up with the following results.

| **Learing Model** | **Accuracy** |
|---|---|
| Logistic Regression | 62% |
| Linear Regression | 54% |
| Neural Network | 66% |
| SVM | 64% |
| Gradient boosting decision tree | 68% |
| Fuzzy K-Nearest Neighbor | 59% |

Table 11.1: comparison of different classifiers

A bar chart for comparison is a follows:

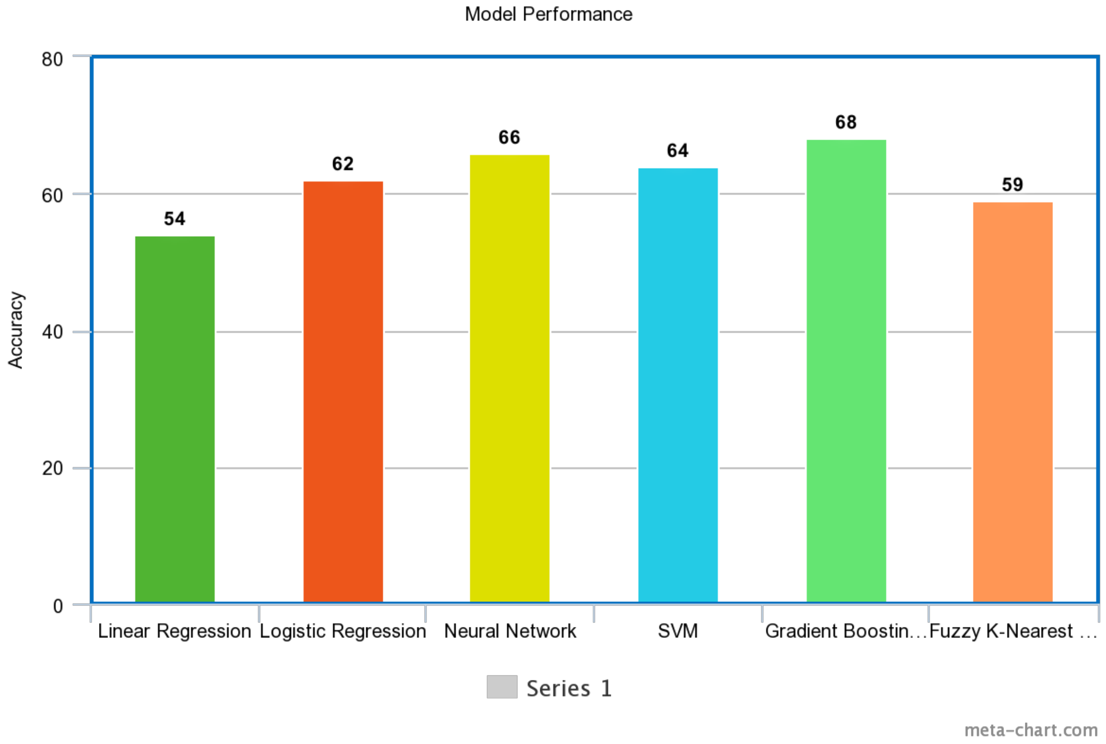


Figure 11.1: Bar charts representing differences among different classifiers

To improve our accuracy and decrease the bias of our data set we try to improve performance by randomly K-folding our data set. Then after many generations of K-folding we find a slight boost in performance.

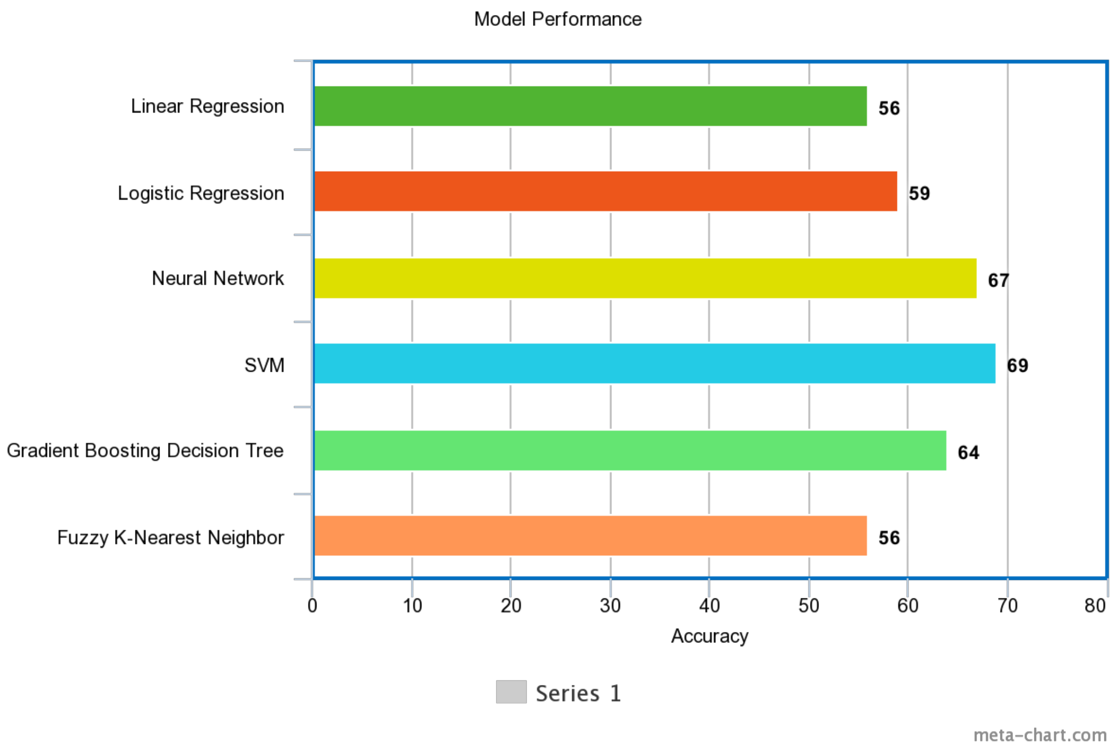


Figure 11.2: performance after K-folding

This came as surprising to us that the accuracy of neural networks came out worse off than much simpler models such as Linear or Logistic regression. But on further analysis we understood that the problem was with the lack of enough availability of data which led to the neural network not learning enough to make accurate predictions.

A week by week prediction accuracy is given below.

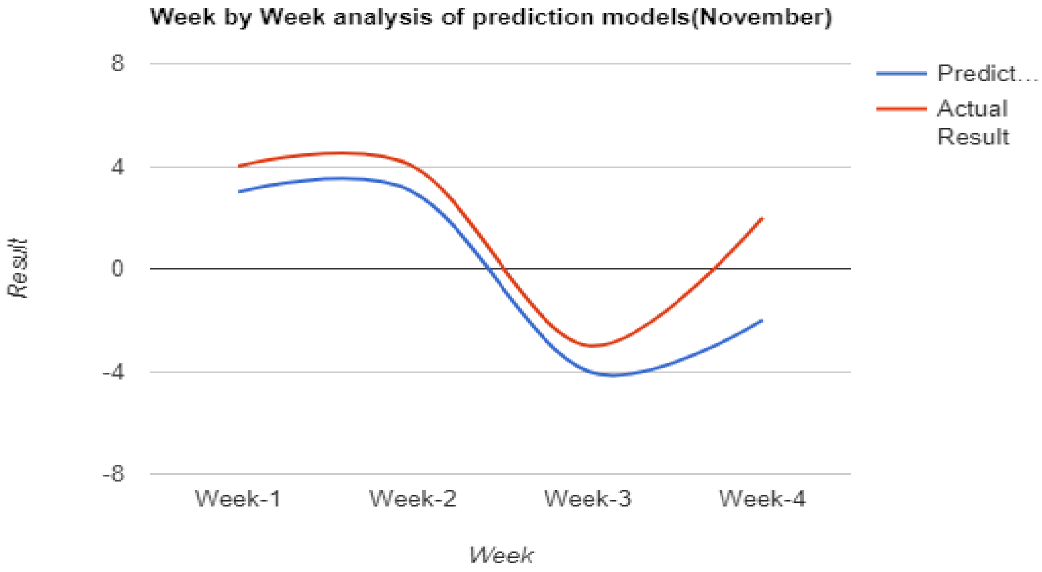


Figure 11.3: Week by Week prediction comparison for the month of November by Linear Regression

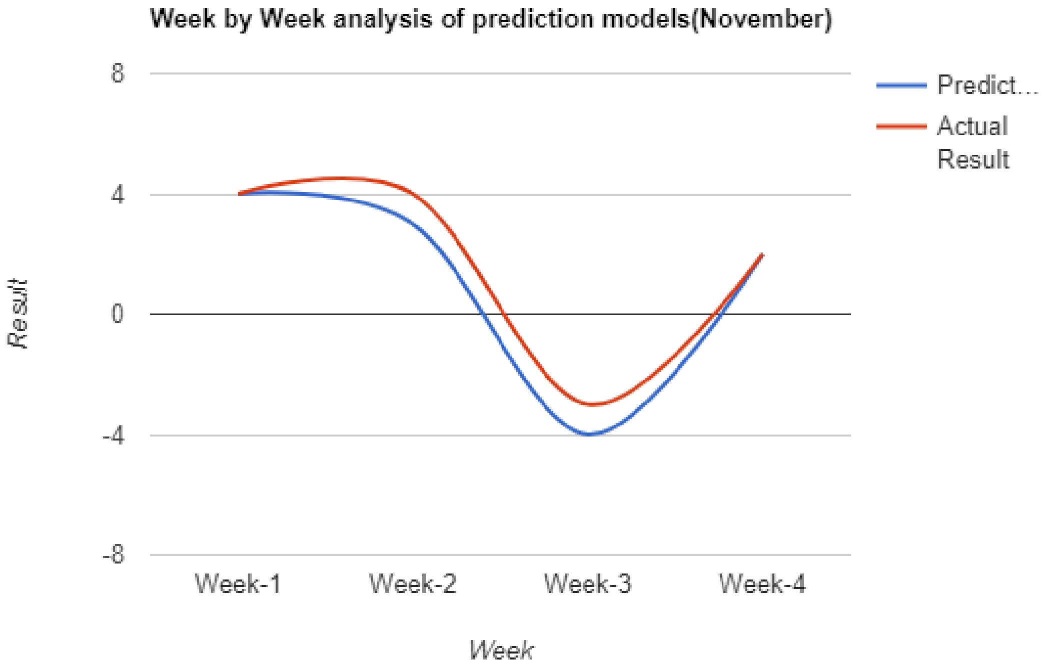


Figure 11.4: Week by Week prediction comparison for the month of November by Logistic Regression

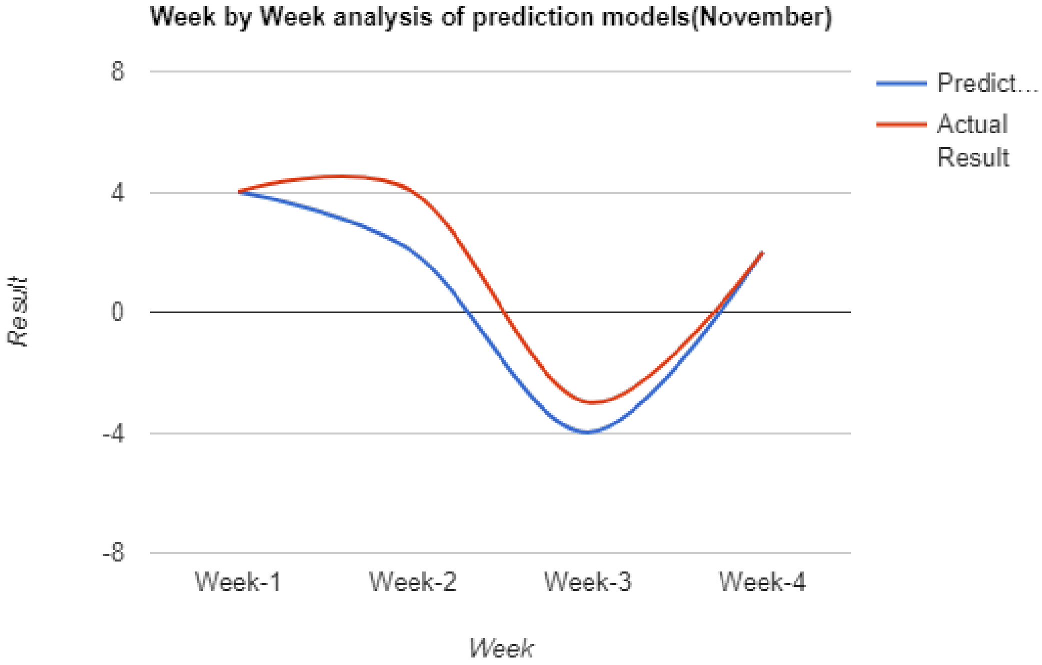


Figure 11.5: Week by Week prediction comparison for the month of November by Neural Network

Our performance on SVM can be shown in the following picture:

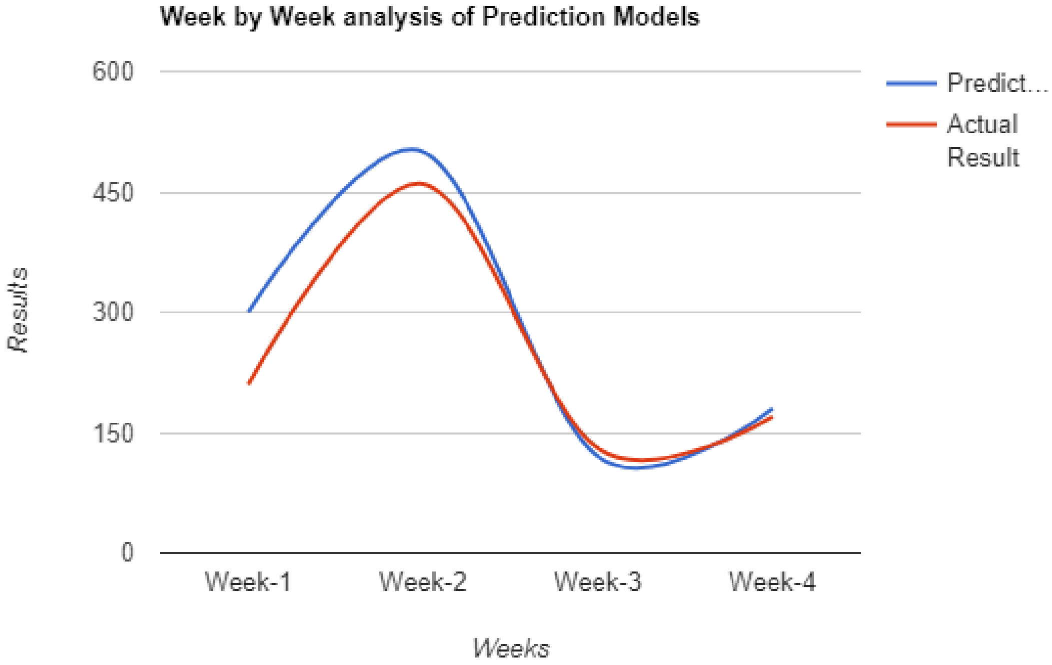


Figure 11.6: Week by Week prediction comparison for the month of November by SVM

Then we looked into KNN. Although this is not a proper machine learning model and it takes a huge overhead to calculate the optimal solution we did find good results.

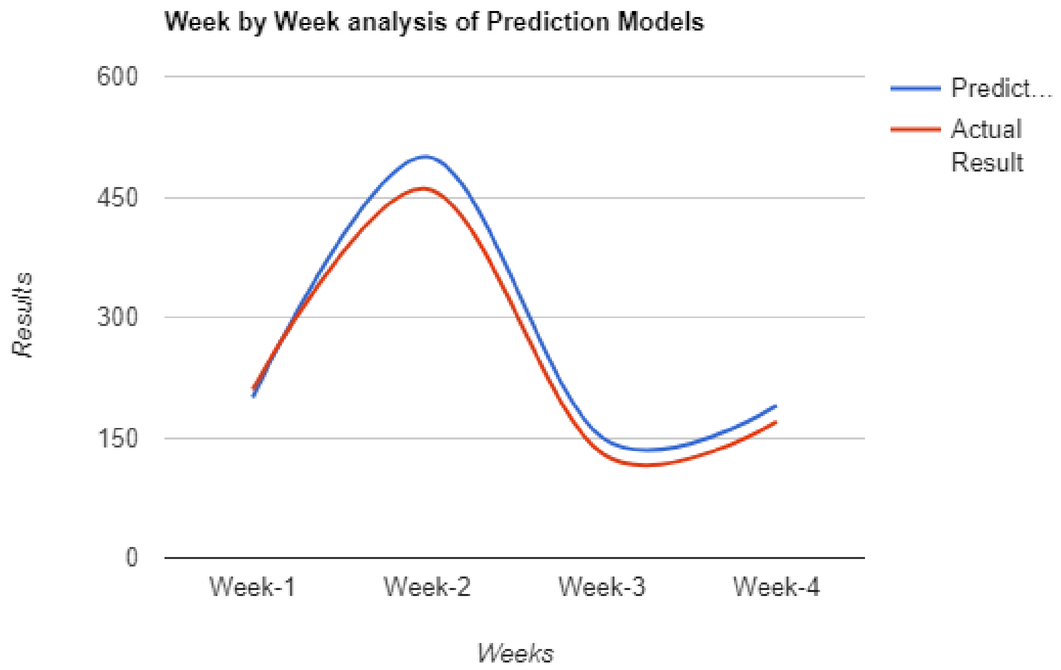


Figure 11.7: Week by Week prediction comparison for the month of November by Fuzzy K-nearest neighbour

We end our analysis with gradient boosting. After we set up our initial simple model, in our case Linear Regression we use Gradient Boosting Decision Tree to minimize our error. We use around 50 iterations of the data set and see what happens.

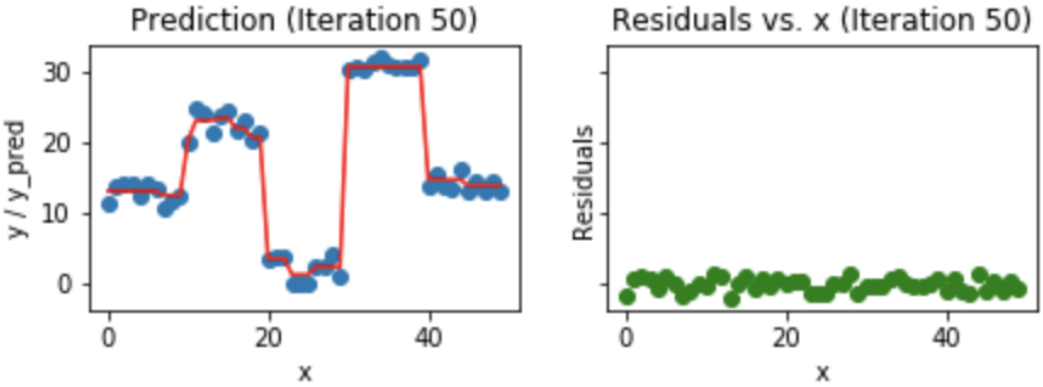


Figure 11.8: Gradient boosting after 50th iteration

This is particularly interesting because this result is similar to what we got in the 20th iteration. So for the rest 30 iterations our algorithm could not find a better result. At this point it seems like a big risk to let the algorithm run any more times because then we might be over fitting the data set.
As of today this model has been giving us the most accurate results. A pictorial example is shown below:

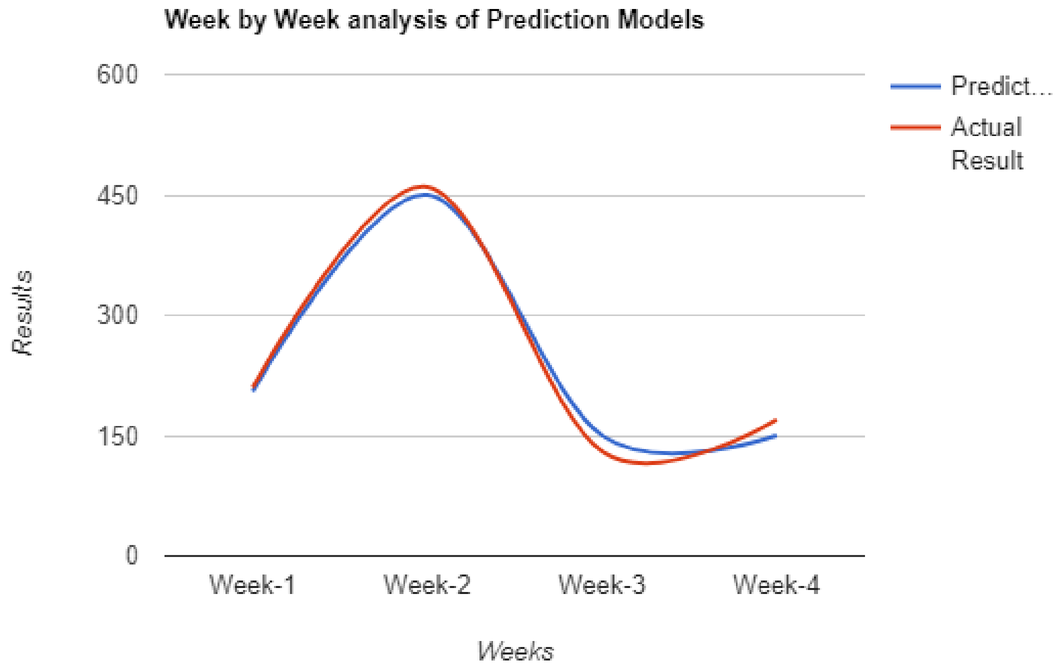


Figure 11.9: Week by Week prediction comparison for the month of November by Gradient boosting decision tree

If we analyze our result based on phrases from StockTwits in we can see that there is indeed a relationship between the stock market and public sentiment. Public sentiment tends to change how the stock market is affected. A tabular way of expressing our data is as follows:

| **Tweets with phrase** | **Mean Absolute Error of Predicting Closing Stock price difference** |
|---|---|
| DJI | 52.23 |
| NYSE | 67.63 |
| Stock Market | 60.31 |

Table 11.2: Results for specific phrases with Linear Regression

| **Tweets with phrase** | **Mean Absolute Error of Predicting Closing Stock price difference** |
|---|---|
| DJI | 35.74 |
| NYSE | 42.16 |
| Stock Market | 56.31 |

Table 11.3: Results for specific phrases with Logistic Regression

| **Tweets with phrase** | **Mean Absolute Error of Predicting Closing Stock price difference** |
|---|---|
| DJI | 52.23 |
| NYSE | 67.63 |
| Stock Market | 60.31 |

Table 11.4: Results for specific phrases with Neural Network

# Chapter 12

# Limitations in Bangladesh for a platform to predict stock market

In Bangladeshi context, building a predictive model to analyze stock market shifts and changes is a difficult task. To make a predictive model for Bangladesh, we have to work with Dhaka Stock Exchange (DSE) [19]. But, there is no financial communication platform available in Bangladesh, where the shareholders are communicating with each other over the market share like the one we have work on for the United States share market, StockTwits [20]. So, to work on a predictive stock market platform, the system in not well structured in context of Bangladesh. Twitter is not a popular social media in Bangladesh. Seldom people use twitter to share emotions via twitter. Another major setback is that, the data required for prediction are rarely open source in Bangladesh. For example, we wanted to access the closing change of the stock market in Dhaka, but there were not any available resources where this data is stored over a period of time and that data is publicly available. Individual companies do gather this type of data, but in almost all cases, these data are not publicly shared. So. Lack of open data is a big problem towards building a predictive platform for Bangladesh. Another problem arises through corruption. As investors are not always honest about their share returns, the data becomes corrupted as well as hard to determine some predictions.

An alternative approach we thought was to use the Facebook API to get user data related to stock market in Bangladesh, because Facebook is more popular platform in Bangladesh than Twitter. But Facebook API has made its data sharing to a bare minimum in the recent times. So, getting the exact data we need to create a predictive model of stock market was not possible at this time. Moreover, there are not a lot of resource related to Bangladeshi stock market in Facebook. Most, of the resources are limited to different Facebook groups and using Facebook graph API, we can only extract information about an individual user data from that group and use only one group at a time. Facebook posts are not limited in words as well, so the sentiment analysis using those posts do not produce very accurate results. Another problem is that, individual users do not post about stock markets in their regular posts, most of the posts are more like informative rather then expressive about certain stock market trends which also make sentiment analysis error prone. Therefore, using Facebook API did not seem a good idea.

To actually make stock market predictive models based on Bangladesh data, we need a platform for stock related discussions among investors, which is at this time not available. We also need a platform of stock related information which will give access to not only the stock prices of that day only, but historical data of stock market in Bangladesh for an extended period of time. And, as we understood during our research, this might take some time because most of the investors in Bangladesh are not used to this type of discussion in social media. So, even if someone build a platform, the investors might not use that at all. All of this limited us to not being able to work on stock market of Bangladesh. But, this same model could be used in future, if stock market data and user opinions are available then.

# Chapter 13

# Discussion and Future plan

We have worked with the twitter data of a very minimum time period, which made it difficult to predict major fluctuations in the stock market. To predict movements more precisely, we will require more data from a longer time span. Although, historical data is available of stock data twitter has restricted data access for a long time period. The twitter API [20] only allows real time twitter mining now, but to collection historical tweet, it is only possible by purchasing different subscription packages.

To predict stock market trends, we only focused on twitter feeds. But, there are a lot of other variables that effects the stock market changes. For example, a natural calamity like tsunami might affect stock market of a region as well. So, alongside tweets, other stock market related news should also be included in the predictions. In future we would like to work on more features like this. Moreover, tweets are often misspelled or there exists grammatical mistakes in them, this makes the classification of the tweet a lot more complicated with traditional classifiers. We would like to use boosted SVM classifier [21] for text to make a more accurate sentiment classification.

Lastly, our work is still in its early phrases and is not suitable for general users use. We would like to build an online tool that general users can use from home. Via this, they can perceive stock market trends and decide whether we should buy, sell or retain any stock. And the tool will update automatically given new market information. But, all of these are areas of future research.

# Chapter 14

# References

[1]December 1965, The New York Stock Exchange and the Commission Rate Struggle, Richard W. Jennings, Berkely Law [2]LOCKWOOD, L. J. and LINN, S. C. (1990), An Examination of Stock Market Return Volatility During Overnight and Intraday Periods, 1964–1989. The Journal of Finance, 45: 591-601. doi:10.1111/j.1540-6261.1990.tb03705.x [3]Volume, volatility, liquidity and efficiency of the Singapore Stock Exchange before and after automation - Naidu, G.N Rozeff, Michael S. JO - Pacific-Basin Finance Journal [4] Prediction of changes in the stock market using twitter and sentiment analysis. Iulian Vlad Serban, David Sierra Gonzalez, and Xuyang Wu. University College London.

[5] Xue Zhang, Hauke Fuehres, and Peter A. Gloor. Predicting stock market indicators through twitter "i hope it is not as bad as i fear". Procedia - Social and Behavioral Sciences, 26:55–62, January 2011.

[6] Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis. 2012.

[7] J. Bollen and H. Mao. Twitter mood as a stock market predictor. IEEE Computer, 44(10):91–94.

[8] V. Sehgal and C. Song, "SOPS: Stock Prediction Using Web Sentiment," Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, NE, 2007, pp. 21-26.

[9] P. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," 2002. 37

[10] S.Wang, C.D. Manning, “Baselines and Bigrams: Simple, Good Sentiment and

Topic Classification," 2012.

[11] G. P. Gang Leng and T. M. McGinnity. An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Networks, 17(10):1477–1493.

[11] Schoen Harald, Gayo-Avello Daniel, Takis Metaxas Panagiotis, Mustafaraj Eni, Strohmaier Markus, Gloor Peter, (2013) "The power of prediction with social media", Internet Research, Vol. 23 Issue: 5, pp.528-543, doi: 10.1108/IntR-06-2013-0115.

[12] Hosmer DW Jr, Lemeshow S, Sturdivant RX (2013) Applied Logistic Regression. Third Edition. New Jersey: John Wiley & Sons.

[13] Long JS (1997) Regression Models for categorical and limited dependent variables. Thousand Oaks, CA: Sage Publications.

[14] Pampel FC (2000) Logistic regression: A primer. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-132. Thousand Oaks, CA: Sage Publications.

[15] Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR (1996) A simulation study of the number of events per variable in logistic regression analysis. Journal of Clinical Epidemiology 49:1373-1379

[16] Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates

[17] Charles Darwin. The Variation of Animals and Plants under Domestication. (1868) (Chapter XIII describes what was known about reversion in Galton's time. Darwin uses the term "reversion".)

[18] Templeton AR: Epistasisand complex traits. In: Epistasisand Evolutionary Process (Edited by: Wade M, Brodie III B, Wolf J). Oxford, Oxford University Press 2000.

[19] Sherriff A, Ott J: Applicationsof neural networks for gene finding. Adv Genet 2001, 42: 287–298.

[20] Utans J, Moody J: Selecting neuralnetwork architectures via the prediction risk application to corporatebond rating prediction. In: ConferenceProceedings on the First International Conference on ArtificialIntelligence Applications on Wall Street 1991.

[21] Hastie T, Tibshirani R, Friedman JH: The Elementsof Statistical Learning. NewYork, Springer-Verlag 2001.

[23] “Dhaka Stock Exchange” 2017. [Online]. Available: http://www.dsebd.org [Accessed 03-dec-2017]

[24] ”StockTwits message”, StockTwits, 2017. [Online]. Available: http:// stocktwits.com. [Accessed: 03-dec- 2017].

[25] “Twitter API”, 2017. [Online]. Available: https://developer.twitter.com/en/ docs [Accessed: 02-july-2017]

[26] C. Bishop, “Pattern Recognition and Machine Learning,” Springer, 2006.

[27] ”line graph”, Rapid table, 2017. [Online]. Available: https://www. rapidtables. com/ tools/line-graph.html [Accessed: 28-nov-2017].

[28] ”bar chart”, 2017. [Online]. Available: https://www.meta-chart.com/bar. [Accessed: 28-dec- 2017].

[29] A. Harb, M. Plantié, G. Dray, M. Roche, F. Trousset, and P. Poncelet, ”Web opinion mining: How to extract opinions from blogs?” 2008, p. 7.

[30] Adankon M., Cheriet M. (2009) Support Vector Machine. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA

[31] ”A fuzzy K-nearest neighbour algorithm” [Online]. Available: https:// www. researchgate.net/ publication/244956382_A_Fuzzy_K-Nearest_Neighbor_Algorithm [ Accessed Mar 02 2018].

[32] Smailović J., Grčar M., Lavrač N., Žnidaršič M. (2013) Predictive Sentiment Analysis of Tweets: A Stock Market Application. In: Holzinger A., Pasi G. (eds) Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data. Lecture Notes in Computer Science, vol 7947. Springer, Berlin, Heidelberg

[33] Milson L. Lima, Thiago P. Nascimento, Sofiane Labidi, Nadson S. Timbó, Marcos V. L. Batista, Gilberto N. Neto, Eraldo A. M. Costa and Sonia R. S. Sousa (2016) USING SENTIMENT ANALYSIS FOR STOCK EXCHANGE PREDICTION. In: International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 7, No. 1, January 2016. DOI : 10.5121/ijaia.2016.7106