BRAC University

Thesis Report on

Predicting Stock Market Movement using Sentiment Analysis of Twitter Feed

---

Authored by,

PRANJAL CHAKRABORTY
ID: 13301071
pcborty@outlook.com

RASHAD AL HASAN RONY
ID: 13301033
rah.rony@gmail.com

UMMAY SANI PRIA
ID: 13301055
ummaysani@gmail.com

Department of Computer Science and Engineering,
BRAC University

Supervised by,

Dr. MAHBUB ALAM MAJUMDAR
Professor,
Department of Computer Science and Engineering,
BRAC University

Date of submission: 16th April, 2017

Declaration

We declare that, this thesis report is our own work and has not been submitted for any other degree or professional qualifications. All sections of the paper that use quotes or describe an argument or concept developed by another author, have been referenced in the reference section.

_____

Dr. Mahbub Alam Majumdar
Supervisor

_____ _____ _____

Pranjal Chakraborty
Author

Rashad Al Hasan Rony
Author

Ummay Sani Pria
Author

Abstract

Collecting opinions of mass people through the social networking sites has become easy and handy now-a-days. These opinions show the sentimental state of a large number of people, which, according to behavioral economics, will give us the idea about their decision-making process. Twitter is a very popular social networking site and by opinion mining, it is possible to get the sentimental state from the tweets. Moreover, there are publicly available data of Twitter. In this thesis paper, we will try to correlate between this sentimental data and stock market data to predict future movement of stock market.

Keywords: Stock market, Tweets, Sentiment analysis, Decision tree, Decision tree regression, Random forest, Boosted tree, Opinion mining, Machine Learning, and Support Vector Machine (SVM)

Acknowledgement

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 amazing research prospect of Machine Learning.

We are also hugely indebted to Abu Mohammad Hammad Ali, former lecturer of BRAC University, for sharing his ideas and experiences, which helped us to make right choices in the implementation phase.

We are thankful to Rabby Hossain, freelance developer, who helped us with his experience with Twitter.

Nevertheless, 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.

Index

1. Introduction

In the past few years, social networking sites have become an integral part of human life. Twitter, which also features the tag “microblogging site”, is such a social networking site in action since 2006. 100 million active Twitter users update nearly 500 million tweets every day. Users express their opinion, decisions, feeling etc. through these tweets. As time passed, Twitter is being used as a tool to share important information rather than casual tweets. Sharing stock market related tweets is one of those areas, where large number of people get involved in sharing tweets, which contain important stock market information and updates.

Stock market prediction has been an active area of research for a long time. The Efficient Market Hypothesis (EMH) states [24] that stock market prices are largely driven by new information and follow a random walk pattern. Though this hypothesis is widely accepted by the research community as a central paradigm governing the markets in general, several people have attempted to extract patterns in the way stock markets behave and respond to external stimuli.

In the year 2008 [23], the last and the most violent stock market crash took place in United States. On September 20, 2008, the bank bailout bill (Emergency Economic Stabilization Act of 2008) to Congress. The Dow bounced around 11000 until September 29, when the US Senate voted against the bailout bill. The DJIA fell 777.68 in intra-day trading. That was the largest point drop in any single day in history. The month started with the news of Lehman Brothers being declared bankrupt, which took place on September 15, 2008. The Dow dropped 504.48 points.

The Dow hit its pre-recession high on October 9, 2007, closing at 14164.43. Less than 18 months later, it had dropped more than 50 percent to 6594.44 on March 5, 2009.

There had been a lot of speculation and predictions about this great recession by the economists. Apart from the economists, people can now discuss about these issues over the internet and share ideas and feelings. Behavioral economics suggests [16] that the decision-making process of individual mostly depends on his/her emotional or sentimental state. So, based on this hypothesis, in this paper, we will try to show a correlation between public sentiment about stock market and future market movements of those products.

To do this, we will first collect tweets for a long period of time, that are related to stock market. What we are trying to do here, is to analyze what people are talking and thinking about stock market. To do that, we will try to find out the best model for sentiment analysis and apply it on the stock related tweets to distinguish whether a tweet is positive or negative.

After that, these sentiment metrics and the existing stock points data (Dow-Jones Industrial Average for our work) will be used to train a regression model. This model will be used to forecast future movement of stock market, at least a day prior. Then we will analyze our prediction with actual data.

2. Literature Review

[7] examined sentiment analysis on Twitter Data and introduced POS-specific prior polarity features and explored the use of a tree kernel to remove the need for tedious feature engineering. Their introduced features and tree kernel performed approximately at the same level, both outperforming the state-of-art baseline.

[8] proposed a novel machine-learning method that applies text-categorization techniques to just the subjective portions of the document. Extracting these portions can be implemented using efficient techniques for finding minimum cuts in graphs. They used mincut to improve the classification of a sentence into either 'objective' or 'subjective', with the assumption that sentences close to each other tend to have the same class. CNNs can capitalize on distributed representations of words by first converting the tokens comprising each sentence into a vector, forming a matrix to be used as input.

[9] proposed a simple one-layer CNN that achieved state-of-the-art results across several data sets. The very strong results achieved with this comparatively simple CNN architecture suggest that it may serve as a drop-in replacement for well-established baseline models, such as SVM [10] or logistic regression. Kim defined a one-layer CNN architecture that uses pre-trained word vectors as inputs, which may be treated as task-specific or static vectors. In that approach, word vectors are treated as fixed inputs, while in the latter they are 'tuned' for a specific task. Learning task-specific vectors through fine tuning gave them better result.

Efthymios Kouloumpis, Theresa Wilson, Johanna Moore [11] have used Hashtag and Emoticon in tweets to predict sentiment using n-gram, lexicon, parts-of-speech, micro-blogging as features.

Peter D.Turney [12] worked on an unsupervised classification algorithm called PMI_IR(Turney 2001),using the association of adjectives in reviews.

Sida Wang and Christopher D. Manning [13] observed that the performance of Naïve Bayes, SVM varies with bigram feature, length of documents without using stopwords, lexicon.

Our work is mainly inspired by Mittal and Goel's work [1]. They classified the publicly available Twitter data of 2009 into 4 classes – Calm, Happy, Alert and Kind (similar to fuzzy membership). They correlated these values with DJIA values to

predict future stock movements and then they used the predicted values in their portfolio management strategy.

Alongside of that, Bollen's work [2] which received widespread media coverage recently. They also attempted to predict the behavior of the stock market by measuring the mood of people on Twitter. The authors considered the tweet data of all twitter users in 2008 and used the OpinionFinder and Google Profile of Mood States (GPOMS) algorithm to classify public sentiment into 6 categories, namely, Calm, Alert, Sure, Vital, Kind and Happy. They cross validated the resulting mood time series by comparing its ability to detect the public's response to the presidential elections and Thanksgiving Day in 2008. They also used causality analysis to investigate the hypothesis that public mood states, as measured by the OpinionFinder and GPOMS mood time series, are predictive of changes in DJIA closing values. The authors used Self Organizing Fuzzy Neural Networks to predict DJIA values using pre-vious 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).

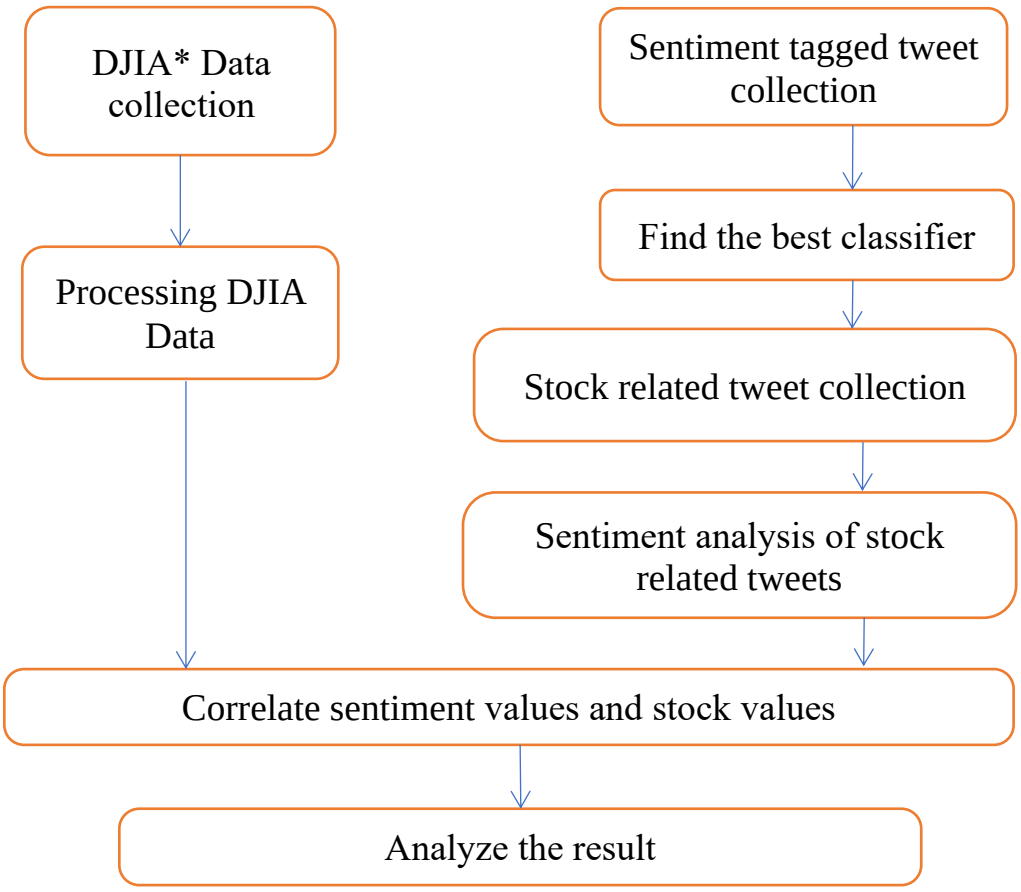
On the other hand, Ali Harb et al. [3] used a different approach which they called AMOD (Automatic Mining of Opinion Dictionaries) where they extracted positive and negative adjectives to identify positive or negative opinions.

By distant supervision, A. Go et al. [4] classified the sentiment of twitter feeds into positive and negative sentiment. They took tweets with emoticons, where emoticons are used as noise label.

In another approach Vivek Sehgal and Charles Song [5] has developed a new measure known as "Trust Value" which assigns trust to each message based on its author. To learn the relationship between sentiments and stock behavior they used "TrustValue" with the sentiments.

3. Methodology

Our course of work will be to find the best model for sentiment classification of the tweets in the first place. We will some the well-known classifiers and choose the one with the best performance. After that, we will collect stock market related tweets and analyze their sentiment with the chosen model. We will then try to correlate the day wise sentiment of the stock market and actual stock values to predict the stock value of a day at least a day before.

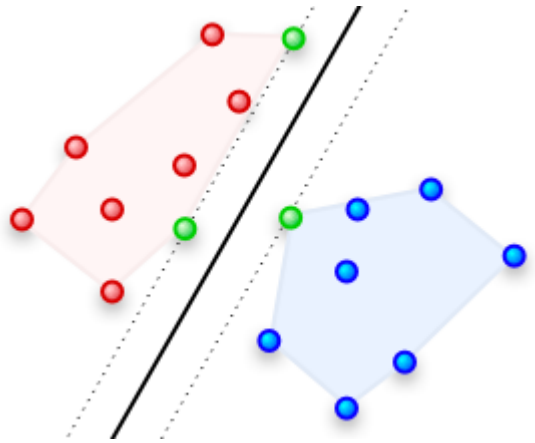


*DJIA = Dow-Jones Industrial Average

Fig.: Workflow of the project

Tweets are at a large full of words, characters, slangs, links which are irrelevant to specify any mood or sentiments. So, processing of the tweets is important for good classifier model. Preprocessed DJIA data, applying some threshold level, will be used to indicate the base for calculating ups and downs of stock market index. The calculated sentiment values of the stock related tweets will be mapped to stock movements with linear regression model. Using linear regression, we will determine probable stock market movement. Some of the basic terminologies, that we need to know to learn about our work, are mentioned here-

4a. Support Vector Machine (SVM):



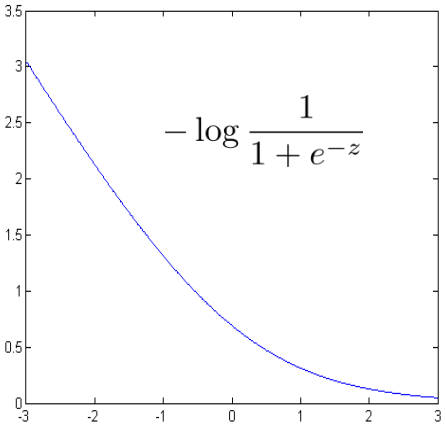
This algorithm is used to classify data. It plots a hyperplane to distinguish between classes. The larger distance between closest data points to the hyperplane the better classification. The math behind this algorithm is about vectors, vector dot product, orthogonal projection. For our sentiment analysis, the feature vector is X, which consists of the words in tweets. Θ vector is the parameter vector that is used to balance the relation of hypothesis function,

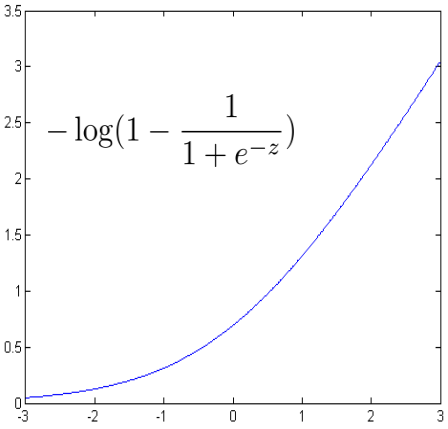
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

SVM cost function:

$$-(y \log h_\theta(x) + (1 - y) \log(1 - h_\theta(x)))$$

$$= -y \log \frac{1}{1 + e^{-\theta^T x}} - (1 - y) \log(1 - \frac{1}{1 + e^{-\theta^T x}})$$





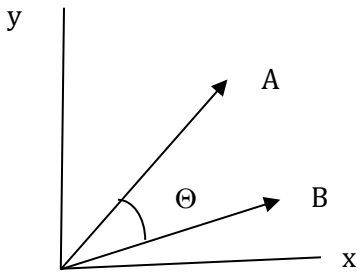
For all of the data points in training set the SVM cost function:

$$\min_{\theta} C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

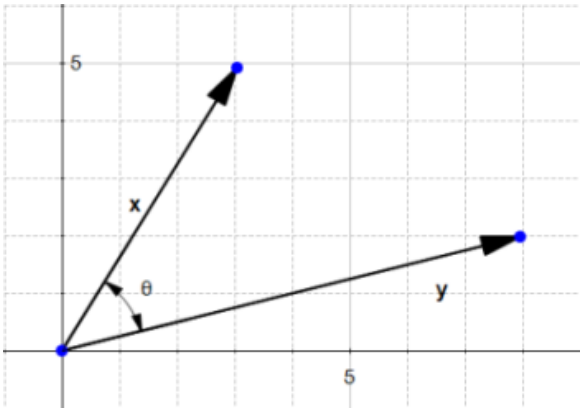
Vector dot product: Dot product is also called the inner product of two vectors. For SVM which is almost like logistic regression, the inner product of $\theta^T X$ is essential to classify.

If A, B are vectors,

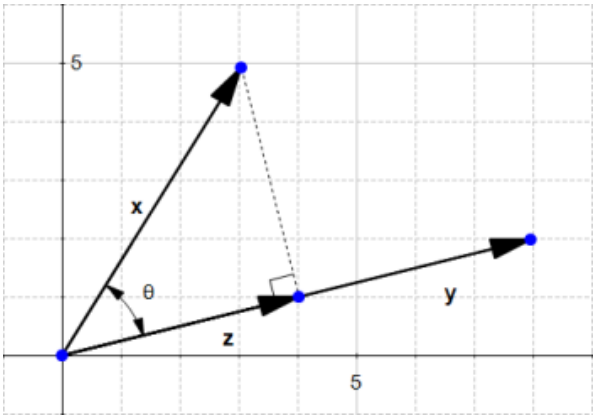
$$A \,.\, B = |A| \,.\, |B| \,.\, cos\Theta$$
$$= x_1 y_1 + x_2 y_2$$



Orthogonal Projection: Given two vectors x and y, we would like to find the orthogonal projection of x onto y.

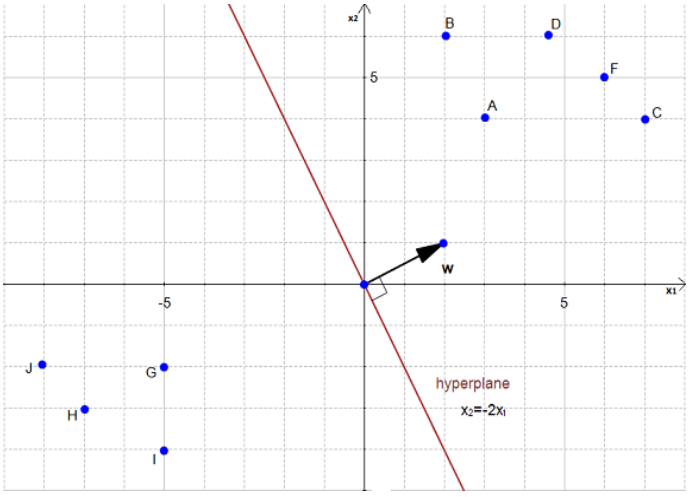


To get the orthogonal projection we draw a perpendicular line from x to y which gives us the vector z.

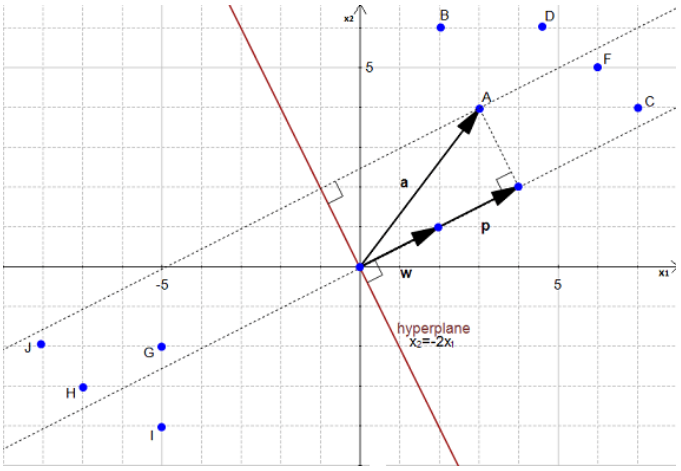


The vector $z = (u.x)$. u is the orthogonal projection of x onto y.

<u>Computing the distance from hyperplane to a data point:</u> For computing distance from a hyperplane we have assumed , we will use the orthogonal projection.

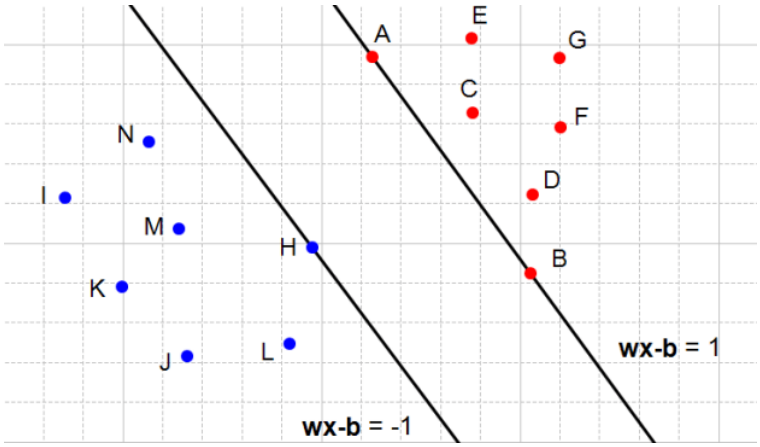


The equation of a hyperplane is defined by: $w^T x = 0$. If we take $0 = x_2 + 2x_1$ as the hyperplane then, $w^T x = x_2 + 2x_1$.



Our goal is to find the distance between the point $A(3,4)$ and the hyperplane. We can see in figure that; this distance is the same thing as $||p||$ and p is the orthogonal projection of A on w.

<u>Computing the best hyperplane:</u>



For each vector x_i, either:

$w.x_i + b \geq 1$, for x_i having the class $y_i = 1$ (1)

or,

$w.x_i + b \leq 1$, for x_i having the class $y_i = -1$ (2)

And multiply both sides by y_i (which is always -1 in equation (2))

$y_i(w.x_i + b) \geq y_i(-1)$ (3)

$y_i(w.x_i + b) \geq 1$, for all $1 \leq i \leq n$ (4)

In inequation (1), as $y_i = 1$, multiplying by y_i doesn't change the sign of the inequation (1).

$y_i(w.x_i + b) \geq 1$, for x_i having the class 1 (5)

We combine equations (4) and (5):

$y_i(w.x_i + b) \geq 1$, for all $1 \leq i \leq n$ (6)

when $||w||$ is the smallest, for example for the point $x_i = H.A$; w is the smallest and we find the best supporting hyperplanes. The hyperplane drawn from the midpoint of these two hyperplanes are the optimal ones.

Minimize in (w, b), $\| w \|$, subject to $y_i(w \cdot x_i + b) \geq 1$ (for any $i = 1,2, \ldots, n$)

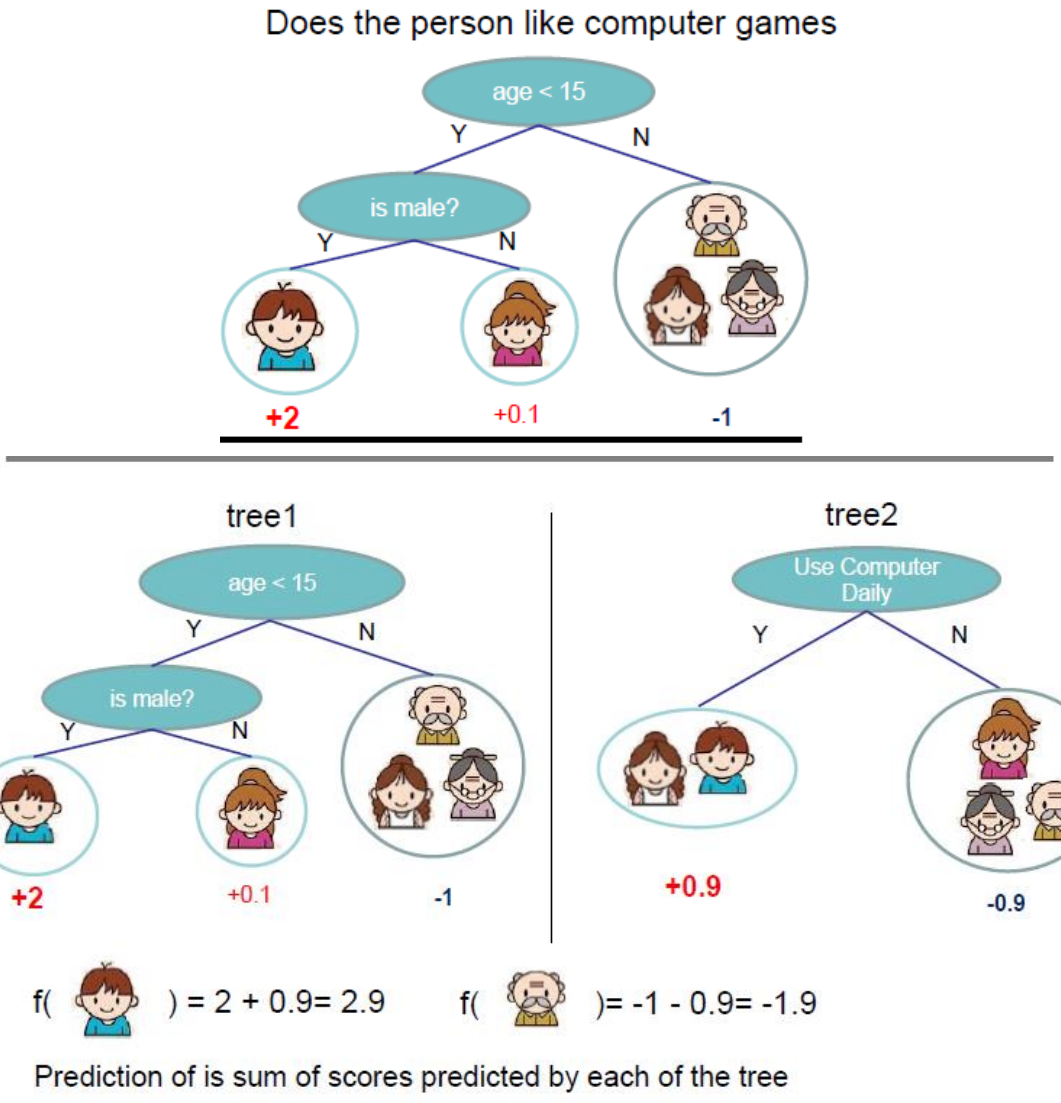
4b. Boosted Tree regression: Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman [18] that boosting can be interpreted as an optimization algorithm on a suitable cost function.

Assuming we have K trees then prediction,

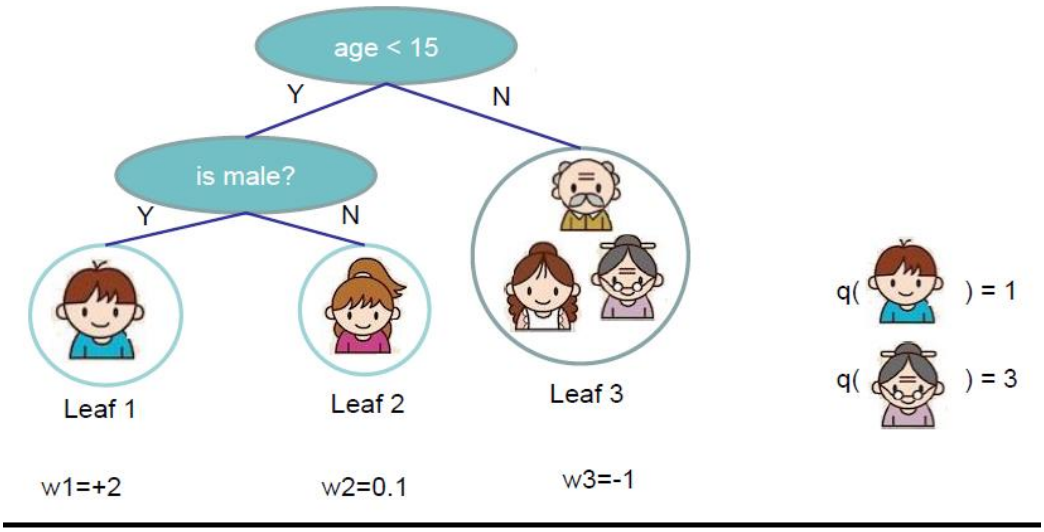
$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

$\mathcal{F}$ is the space of functions, containing all regression trees.



$f_t(x)$: We define tree by a vector of scores in leaves, and a leaf index mapping function that maps an instance to a leaf

$$f_t(x) = w_{q(x)}, \quad w \in \mathbf{R}^T, q : \mathbf{R}^d \to \{1, 2, \cdots, T\}$$



Objective:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

$l(y_i, \hat{y}_i)$ is training loss, the rest of the equation is complexity of the trees (Number of nodes in the tree, depth).

Loss on training data:

$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

Starting from constant prediction; adding a new function for new rounds each time,

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

Now, how do we decide which function to add? To do that, we need to optimize the objective.

The prediction at round t is-

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \end{aligned}$$

Considering square loss-

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} \left(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i))\right)^2 + \Omega(f_t) + const \\ &= \sum_{i=1}^{n} \left[2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2\right] + \Omega(f_t) + const \end{aligned}$$

Now, our goal is to find f_t to minimize our objective.

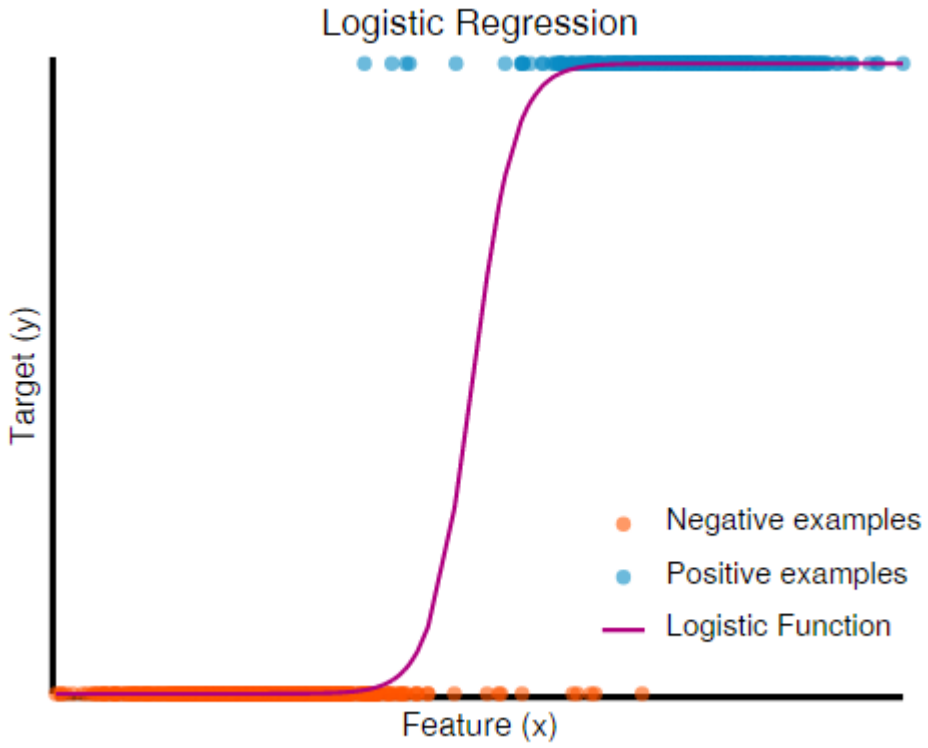
$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Number of leaves L2 norm of leaf scores

4c. Linear Regression: Linear regression attempts [17] to model the relationship between two variables by fitting a linear equation to observed data. It can also be fitted in a quadratic equation and still it will be called linear regression. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

A linear regression line with a linear equation is of the form, $y = a + bx$, where x is the explanatory variable and y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$).

4d. Logistic Regression: Logistic regression [17] is the type regression analysis, where the dependent variable is dichotomous or binary. In logistic regression, the probability that a **binary target is True** is modeled as a logistic function of a linear combination of features. The following figure illustrates how logistic regression is used to train a 1-dimensional classifier. The training data consists of positive examples (depicted in blue) and negative examples (in orange). The decision boundary (depicted in pink) separates out the data into two classes.



The logistic regression can be understood simply as finding the β parameters that best fit:

$$y = \begin{cases} 1 & \beta_0 + \beta_1 x + \varepsilon > 0 \\ 0 & else \end{cases}$$

where, ε is an error distributed by the standard logistic distribution.

Given a set of features x_i, and a label $y_i \in \{0,1\}$, logistic regression interprets the probability that the label is in one class as a logistic function of a linear combination of the features:

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

Analogous to linear regression, an intercept term is added by appending a column of 1's to the features and L1 and L2 regularizers are supported. The composite objective being optimized for is the following:

$$min_\theta \sum_{i=1}^{n} f_i(\theta) + \lambda_1 ||\theta||_1 + \lambda_2 ||\theta||_2^2$$

Where λ_1 is the L1_penalty and λ_2 is the L2_penalty.

4e. Decision Trees: A decision tree reaches [17] its decision by performing a sequence of tests. Each internal node in the tree corresponds to a test of the value of one of the input attributes, A_i, and the branches from the node are labeled with the possible values of the attribute, $A_i = V_{ik}$. Each leaf node in the tree specifies a value to be returned by the function.

Thus, the Information Gain(IG) function would be

$$IG\ (A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

where,

$$remainder\ (A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n}\ I\ (\frac{p_i}{p+n}, \frac{n_i}{p+n})$$

4. Sentiment Analysis

4a. Data:

Our dataset contains, as mentioned earlier, 1.5 million (1,578,614) hand-tagged tweets, collected through Sentiment140 API [21] and Kaggle [19]. The tweets are tagged ‘0’ and ‘1’ for being ‘positive’ and ‘negative’ respectively. Then we performed a 90%-10% random split over the dataset, to divide the dataset into training dataset, containing 1,499,337 tweets and testing dataset, containing 79,277 tweets.

The distribution of positive and negative tweets is almost equal in both training and testing data.

	Training	Testing
Positive	750,527	39,651
Negative	748,810	39,626

Table 4.1: Data distributions

Tweets are often ill-punctuated, grammatically incorrect and, because of the character-limit, very concise. Some of the entries of our dataset is as follows.

Sentiment	**SentimentText**
0	is so sad for my APL friend.....
0	I missed the New Moon trailer...
1	omg its already 7:30 :O
0	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)...
0	i think mi bf is cheating on me!!! T_T
0	or i just worry too much?
1	Juuuuuuuuuuuuuuuussssst Chillin!!

Table 4.2: Unprocessed data frame

4b. Preprocessing:

The tweets have gone through following pre-processing steps-

(i) All characters have been changed to lower case letters to avoid repetition of same words in feature vector.

(ii)All urls, starting with <http://>, <https://> and <www.> have been stripped off from the text.

(iii) Consecutive whitespaces are replaced by single whitespace. Apart from that, more than two consecutive characters, symbols, punctuation marks have been replaced by just two of them.

((iv)Usernames (starts with <@>) are removed. Hashtags are kept stripping off the <#> sign. Emoticons are kept the same.

(v) Apart from these, all retweet signs ('RT'), and double (<">) quote signs are stripped off. All non-ascii characters are removed too.

After applying the filtering technique mentioned above, the tweets took a form like this.

Sentiment	SentimentText
0	is so sad for my apl friend..
0	i missed the new moon trailer..
1	omg its already 7:30 :o
0	.. omgaga. im soo im gunna cry. i've been at this dentist since 11.. i was suposed 2 just get a crown put on (30mins)..
0	i think mi bf is cheating on me! t_t
0	or i just worry too much?
1	juusst chillin!

Table 4.3: First 8 rows of the filtered data frame

4c. Experiment and Results

GraphLab Create [20] has been used to work with this large amount of data. We used the Bag of Words model as the feature set. At first, we are trying to create a logistic classifier with the training data and gather information of 20 iterations.

Number of examples:	1424961
Number of classes:	2
Number of unpacked features:	264041
Number of coefficients:	264042

With the step size of 1.0, the training accuracy reached the saturation at around 0.845, which is not really overfitting.

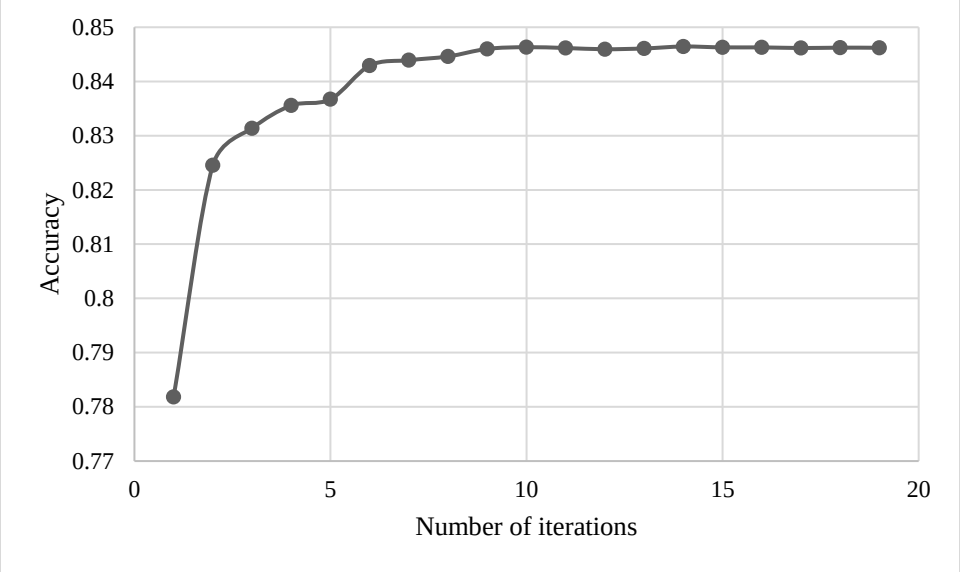


Figure 4.1: Training Accuracy

Before starting the training, we defined randomly chosen 5% of the training data as a validation set to predict the testing accuracy and determine the iteration count for creating the classifiers.

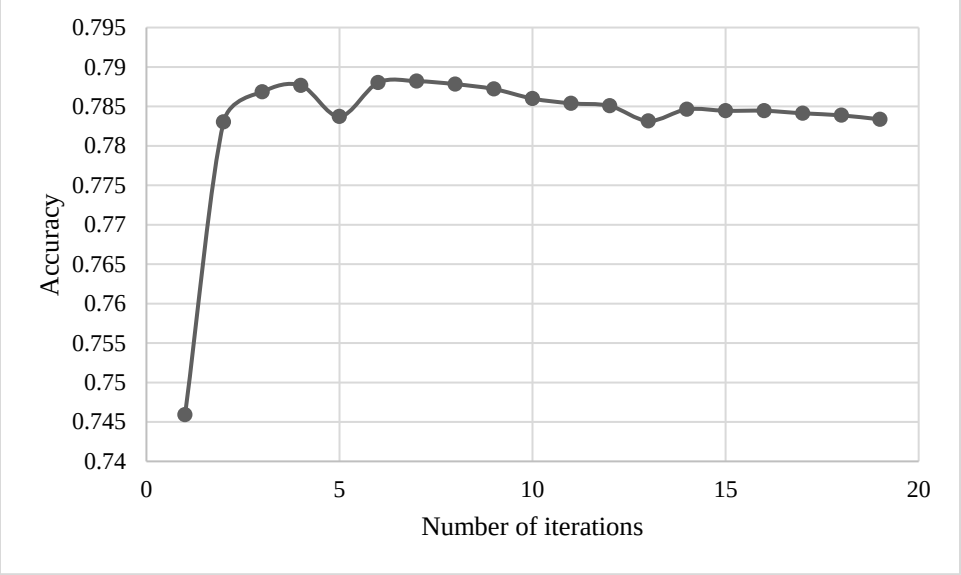


Figure 4.2: Validation Accuracy

Here, over the validation set, 8th iteration achieves the highest accuracy. So, the model created with 8 iteration is expected to perform the best over the test data. Separate logistic regression models have been created for each number of iterations and tested over the testing dataset.

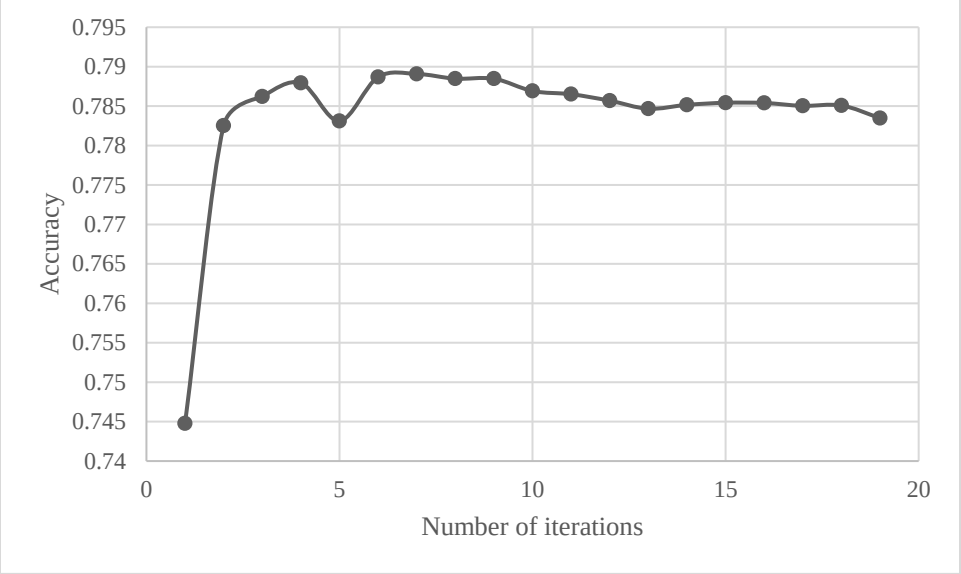


Figure 4.3: Test Accuracy

We have created separate learning models with Logistic Regression, SVM, Random Forest, Boosted Tree and Decision Tree with same training dataset and till 8 iterations. Then we tested those classifiers over same training dataset.

Figure 4.4: Comparison with other classifiers

We also evaluated the accuracy with bigram and trigram model to extracted features. The accuracy with bigram and trigram model are not as much as the unigram model, as the sentences are not well structured.

Models	**Unigram**		**Bigram**		**Trigram**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.8452	0.7890	0.984	0.7646	0.988	0.7382
SVM	0.8463	0.7910	0.9832	0.7595	0.9873	0.74
Decision Tree	0.639	0.6415	0.524	0.522	0.526	0.5253
Boosted Tree	0.55	0.5499	0.604	0.6029	0.547	0.5403
Random Forest	0.644	0.6428	0.534	0.535	0.5314	0.5344

Table 4.4: n-gram (n=1-3) model outputs

Here, n-gram models (n>1) starts to overfit in logistic regression and SVM. Both train and test accuracy decreases in tree based classifiers and test accuracy drops to the accuracy of randomly chosen class (0.5 for binary classes). So, we are continuing with unigram bag of words model.

The detailed information of the test data evaluation with unigram model is as follows-

Models	Accuracy	F1 score	Precision	Recall
Logistic Regression	0.7890	0.7881	0.7915	0.7849
SVM	0.7910	0.7908	0.7921	0.7897
Decision Tree	0.6415	0.6649	0.6244	0.7112
Boosted Tree	0.684	0.6962	0.6705	0.7239
Random Forest	0.6428	0.6688	0.6236	0.7211

Table 4.5: Complete information of the test data evaluations

Details of the predictions-

Target label	Predicted Label	Logistic Regression	SVM	Decision Tree	Boosted Tree	Random Forest
Positive	**Positive**	31431	31406	22654	25459	23338
Positive	**Negative**	8195	8220	16972	14167	16288
Negative	**Positive**	8531	8339	11457	10889	11717
Negative	**Negative**	31120	31312	28194	28762	27934

Table 4.6: Details of the predictions in numbers

Here, clearly SVM performs the best. So, for evaluating the stock related tweets, we will create another SVM classifier with the whole dataset of tagged tweets.

We also considered using stopwords filtering for the tweets, but tweets contain very low amount of but significant information, which tends to get lost due to stopwords filtering. The results with stopwords filtering is as follows.

	Accuracy	F1 score	Precision	Recall
Logistic Regression	0.7507	0.7558	0.7412	0.7709
SVM	0.7546	0.77614	0.7412	0.7828
Decision Tree	0.5550	0.6853	0.5302	0.9686
Boosted Tree	0.6197	0.7087	0.5744	0.9249
Random Forest	0.5702	0.691	0.5395	0.9606

Table 4.7: Complete information of the test data evaluation with stopwords filtering

Here, recall (sensitivity) has substantially increased in the tree based classifiers, because these classifiers done significantly better at detecting negative tweets.

Details of the predictions using stopwords-

Target label	Predicted Label	Logistic Regression	SVM	Decision Tree	Boosted Tree	Random Forest
Positive	**Positive**	28954	28786	5593	12449	7118
Positive	**Negative**	10672	10840	34033	27177	32508
Negative	**Positive**	9084	8611	1245	2974	1563
Negative	**Negative**	30567	31040	38406	36677	38088

Table 4.8: Details of the predictions in numbers with stopwords filtering

Now, we will use this whole dataset with 1.5 million tweets as training set for the next step and we will SVM classifier for sentiment classification.

5. Stock market movement analysis

From our sentiment analysis model, we found that SVM worked best on our training data along with unigram feature. So, for sentiment analysis of the stock related tweets, we used SVM with unigram approach.

We have gathered tweets with phrases ‘stock market’, ‘stocktwits’ and ‘AAPL’ from January to December 2016. We used tweets from January to August along with DJIA closing points to train our model and tweets from September to December for testing. Tweets containing ‘stock market’, ’stocktwits’ were trained to predict general stock market points closing difference. Besides, tweets containing ‘AAPL’ were trained to predict Apple Inc. closing stock points difference. Tweets were ignored of the days when the stock market or Apple’s stock prices was closed. We have downloaded at most a thousand tweets of a day for keywords ‘stock market’, ‘stocktwits’, ‘AAPL’ respectively.

The ‘stocktwits’ tagged tweets that has been used was formatted in the following way. Others data frames with tweets containing ‘stock market’ and ‘AAPL’ were formatted the same way.

Date	Tweets
1/1/2016	@christina commented on stocktwits : $$$ Sry grabbed the wrong bitlink…….
1/1/2016	RT @destinblu commented on stocktwits : !! you got to know when to fold em ;)
1/1/2016	@stocktwits : the markets in 2015 $ spy (s&p 500) -2% $ qq (nasdaq) +6% $ iwm (russell) -6% $ tlt (bonds) -3% $ gld (go
1/1/2016	RT @doep commented on stocktwits : When it makes the headlines on the shitty-republican economic/business paper i am oka….
1/1/2016	happy new year to all @clueless8_trading followers & stocktwits staff for their continued suppo! pic.twitter.com/vjxjxprliq

Table 5.1: Data frame of the tweets containing ‘stocktwits’

5a. Sentiment analysis:

We tagged each tweet of these data frames as either '0' if positive or '1' if negative. To classify these tweets, we have used the same preprocessing technique that has been used in the sentiment analysis phase. As mentioned before, we used the SVM classifier, trained with the 15 million manually sentiment tagged tweets that has been used earlier and the feature vector was created (also for the stock related tweets) with unigram model.

After sentiment tagging and preprocessing, the data frame looks like this-

Date	Tweets	Sentiment
1/1/2016	commented on stocktwits : sry grabbed the wrong bitlink..	1
1/1/2016	commented on stocktwits : you got to know when to fold em	1
1/1/2016	stocktwits : the markets in 2015 spy (s&p 500) -2% $ qq (nasdaq) +6% $ iwm (russell) -6% $ tlt (bonds) -3% $ gld (go	0
1/1/2016	commented on stocktwits : when it makes the headlines on the shitty-republican economic/business paper i am oka..	1
1/1/2016	happy new year to all clueless8_trading followers & stocktwits staff for their continued suppo!	0

Table 5.2: Data frame of tweets containing 'stocktwits' after preprocessing and sentiment tagging

5b. Creating training data:

We have created separate training and result predicting data frames for the tweets containing 'stock market', 'stocktwits', and 'AAPL'. Our training data consist of tweets from January 1, 2016 to August 31, 2016. The training data frames contain

average marginal value of at most one thousand tweets a day and closing stock value difference of that day and the next day. So, the hypothesis is, after getting marginal value of the tweets of one day, we can predict how much stock market will rise or fall the next day. Here, we are getting the marginal values of tweets through sentiment analysis with SVM over them. The tables below show how we have formatted our data to train the model for stock market points (difference) prediction. The tables show the first few entries to train separate Boosted Tree Regression models.

Date	Average Marginal Value of tweets	Actual Closing price difference
04/01/2016	8.978216556	-2.64
05/01/2016	30.08666345	-2.01
06/01/2016	5.077105478	-4.25
07/01/2016	0.800589583	0.510002
08/01/2016	19.76872346	1.57
11/01/2016	81.77476036	1.43
12/01/2016	13.22260718	-2.57

Table 5.3: First few entries to train Boosted Regression Tree model of the tweets containing ‘AAPL’

Date	Average Marginal Value of tweets	Actual Closing price difference
04/01/2016	31.61296056	9.72
05/01/2016	8.431343781	-252.15
06/01/2016	-5.815192181	-392.41
07/01/2016	10.8736832	-167.65
08/01/2016	-52.30292855	52.12
11/01/2016	-15.69335651	117.65
12/01/2016	20.65283859	-364.81

Table 5.4: First few entries to train Boosted Regression Tree model of the tweets containing ‘stock market’

Date	Average Marginal Value of tweets	Actual Closing price difference
04/01/2016	-5.119520943	9.72
05/01/2016	13.2967218	-252.15
06/01/2016	22.6936427	-392.41
07/01/2016	18.15792941	-167.65
08/01/2016	17.38733424	52.12
11/01/2016	9.017401005	117.65
12/01/2016	93.87863623	-364.81

Table 5.5: First few entries to train Boosted Regression Tree model of the tweets containing ‘stocktwits’

Some dates are not considered as the stock market was closed on those days. Tweets of those days are discarded too eventually.

The tables, as shown above, is formatted such a way that, for a date, the sentiment value is of that day and the difference of closing price is between that day and the next day. Which means, in the table of ‘stocktwits’, the average marginal value of tweets -5.119520943 of 04/01/2016 and the next day closing price increased by 9.72 points, so that, the dataset is trained to predict the stock value difference of the next day. So, if given today's tweets based on our training set we can predict tomorrow's stock price difference in advance.

5c. Predicted results:

In order to predict the results, we created the dataset only with the happiness index (average marginal values of tweets) as input from September 1, 2016 to December 31, 2016. Our target is to assign the predicted difference generated by respective Boosted Tree Regression model for a particular date, where the input is the happiness index of the day.

After evaluation, separate table will be created with predicted difference and actual difference. The input and output datasets for each of the ‘stock market’, ‘stocktwits’ and ‘AAPL’ categories are formatted as follows.

Date	Average Marginal Value of tweets
02/09/2016	-42.6213
06/09/2016	25.81552
07/09/2016	6.229185
08/09/2016	19.41263
09/09/2016	9.25263
12/09/2016	-17.8348
13/09/2016	20.81668

Table 5.6: First few lines of the input data frame of the tweets containing 'AAPL' for the Boosted Regression Tree model trained with the tweets containing 'AAPL'

Date	Actual Closing difference	Predicted value
02/09/2016	-0.030006	0.084494352
06/09/2016	0.660004	0.36068821
07/09/2016	-2.840004	-1.625716209
08/09/2016	-2.39	-0.73836863
09/09/2016	2.310005	-0.753412366
12/09/2016	2.509995	0.389092475
13/09/2016	3.82	0.049129248

Table 5.7: First few lines of the output data frame of the tweets containing 'AAPL' for the Boosted Regression Tree model trained with the tweets containing 'AAPL'

Date	Average Marginal Value of tweets
02/09/2016	69.77249387
06/09/2016	25.58993275
07/09/2016	72.68831828
08/09/2016	37.23681252
09/09/2016	47.71251388
12/09/2016	83.45846297
13/09/2016	41.7261997

Table 5.8: First few lines of the input data frame of the tweets containing ‘stock market’ for the Boosted Regression Tree model trained with the tweets containing ‘stock market’

Date	Actual Closing difference	Predicted value
02/09/2016	46.16	83.37435
06/09/2016	-11.98	-16.4363
07/09/2016	-46.23	-139.138
08/09/2016	-394.46	29.32628
09/09/2016	239.62	4.388528
12/09/2016	-258.32	25.82002
13/09/2016	-31.98	8.013132

Table 5.9: First few lines of the output data frame of the tweets containing ‘stock market’ for the Boosted Regression Tree model trained with the tweets containing ‘stock market’

Date	Average Marginal Value of tweets
02/09/2016	31.54802
06/09/2016	-19.5212
07/09/2016	5.687261
08/09/2016	9.668922
09/09/2016	18.07009
12/09/2016	18.30938
13/09/2016	2.104533

Table 5.10: First few lines of the input data frame of the tweets containing ‘stocktwits’ for the Boosted Regression Tree model trained with the tweets containing ‘stocktwits’

Date	Actual Closing difference	Predicted value
02/09/2016	46.16	86.58723
06/09/2016	-11.98	114.4555
07/09/2016	-46.23	-5.4833
08/09/2016	-394.46	17.87045
09/09/2016	239.62	-3.50789
12/09/2016	-258.32	-3.50789
13/09/2016	-31.98	-36.6365

Table 5.11: First few lines of the output data frame of the tweets containing ‘stocktwits’ for the Boosted Regression Tree model trained with the tweets containing ‘stocktwits’

5d. Plotting and Evaluation:

We have plotted the output data with both actual stock difference and predicted stock difference in a time series, time is in the x-axis and difference of stock points in the y-axis.

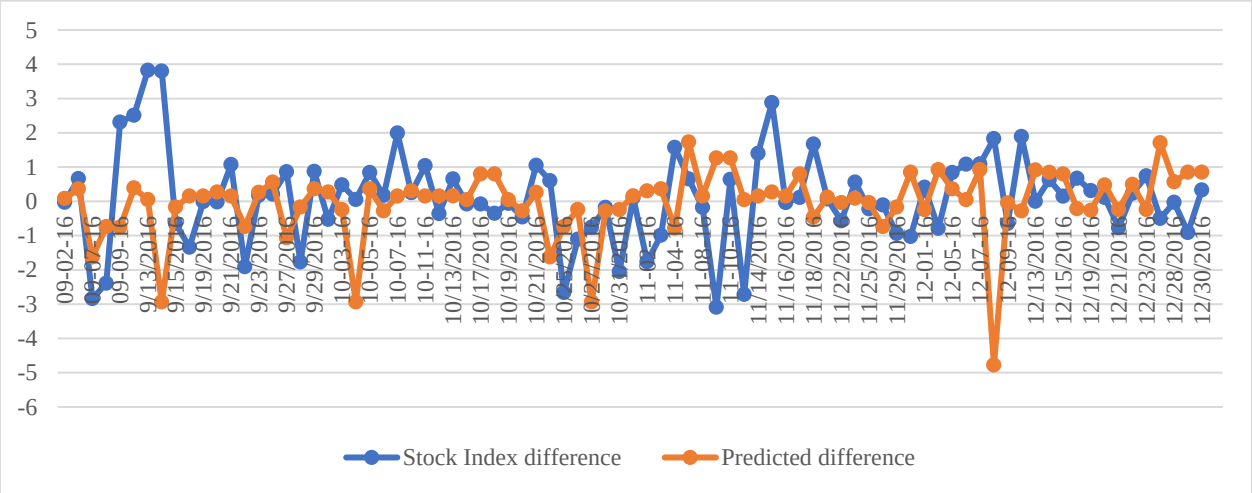


Figure 5.1: Company specific ('AAPL') closing stock price difference prediction using tweets containing 'AAPL' and Boosted Regression Tree.

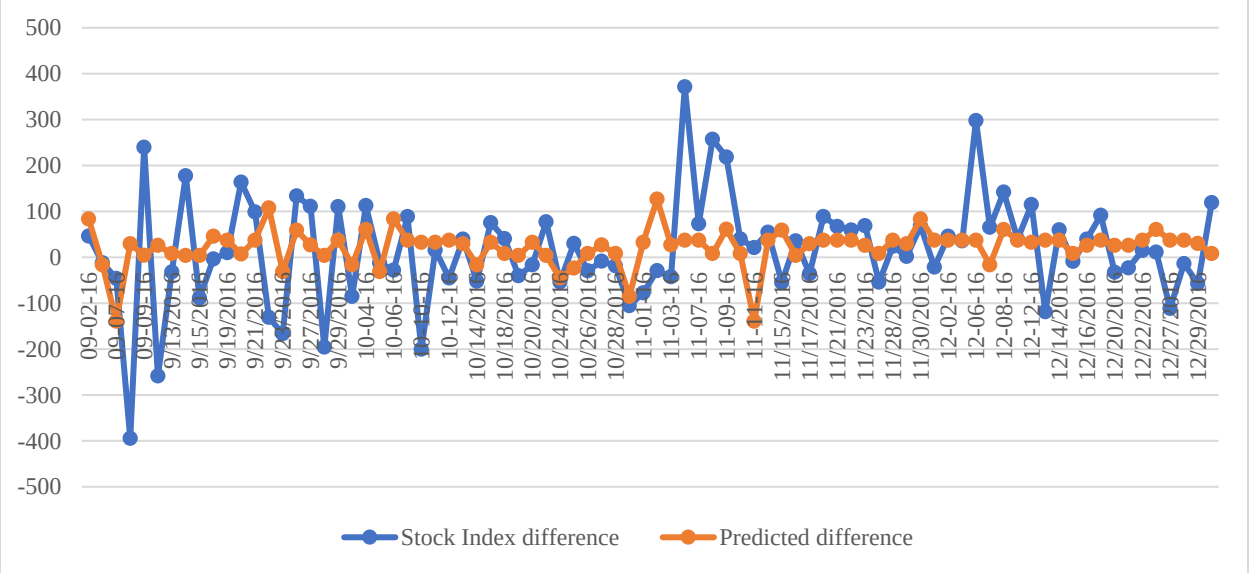


Figure 5.2: Prediction of Closing Stock Price difference using value (Boosted Regression Tree) on tweets containing 'stock market'.

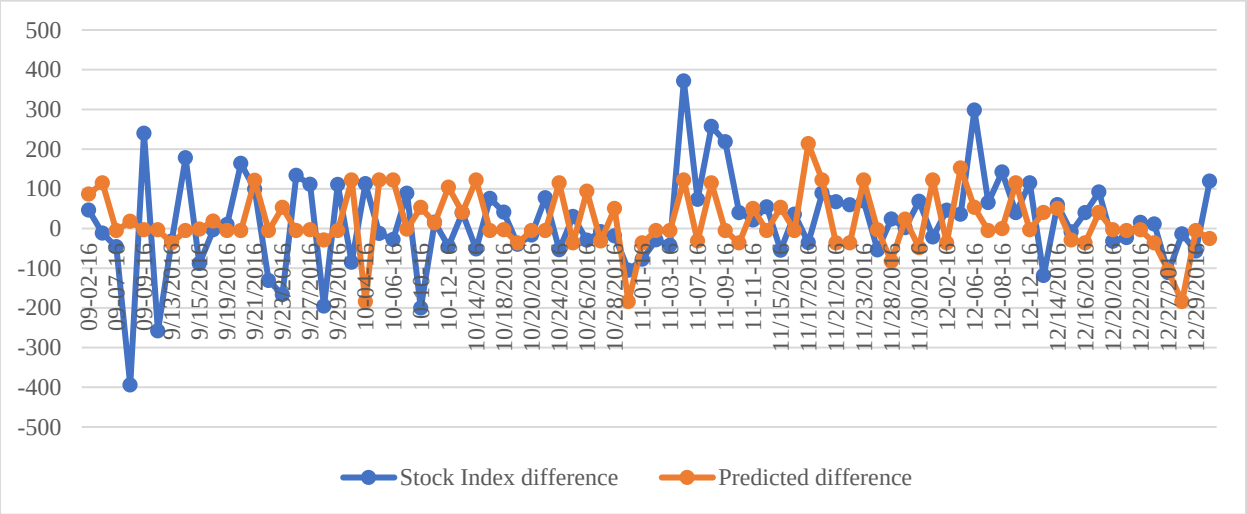


Figure 5.3: Prediction of Closing Stock Price difference using value (Boosted Regression Tree) on tweets containing ‘stocktwits’.

As we have seen, the output for overall stock value prediction works far better than for particular product. After adjusting these predicted differences and actual differences with the stock market values of the previous day, we find the desired output.

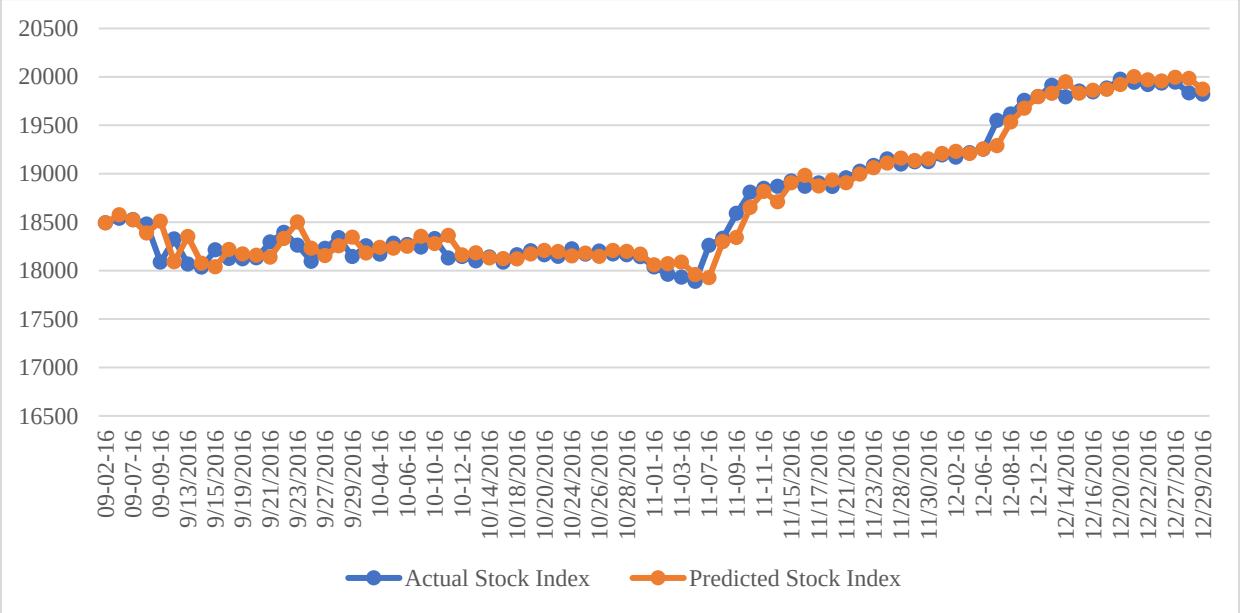


Figure 5.4: Actual stock index vs. Predicted stock index graph of the tweets containing ‘stock market’

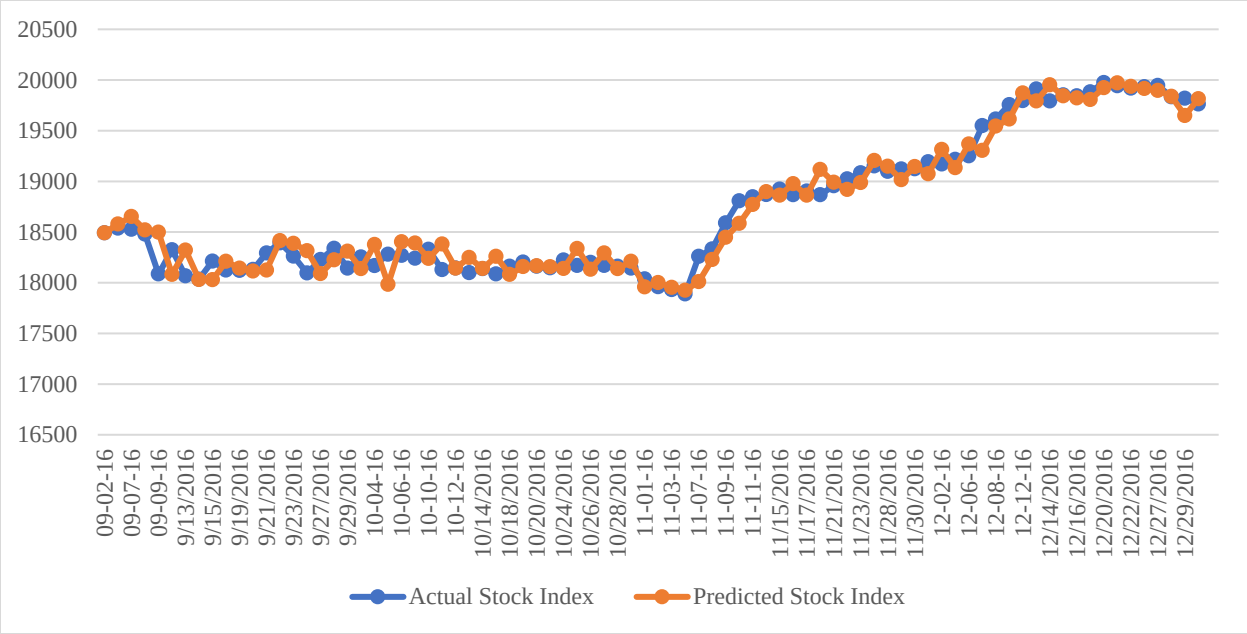


Figure 5.5: Actual stock index vs. Predicted stock index graph of the tweets containing 'stock twits'

A closer look to a one month range (October 11, 2016 to November 11, 2016) might get us a better view-

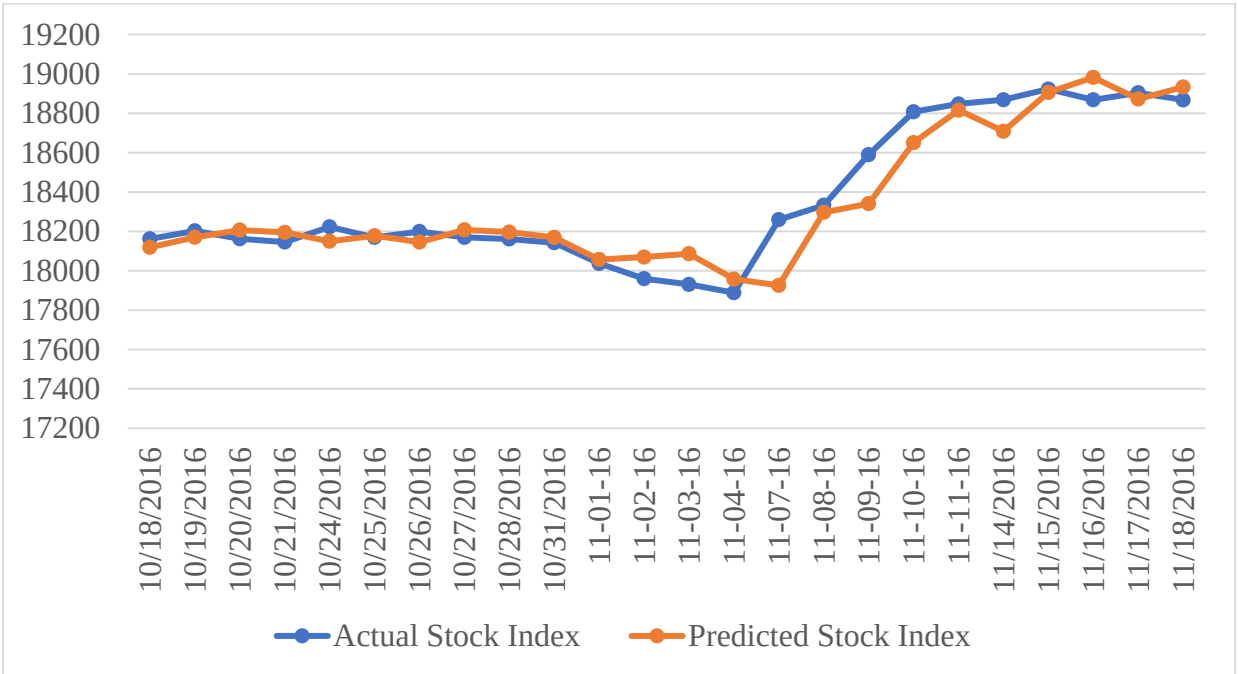


Figure 5.6: Actual stock index vs. Predicted stock index graph of the tweets containing 'stock market' from October,11 to November, 11 of 2016

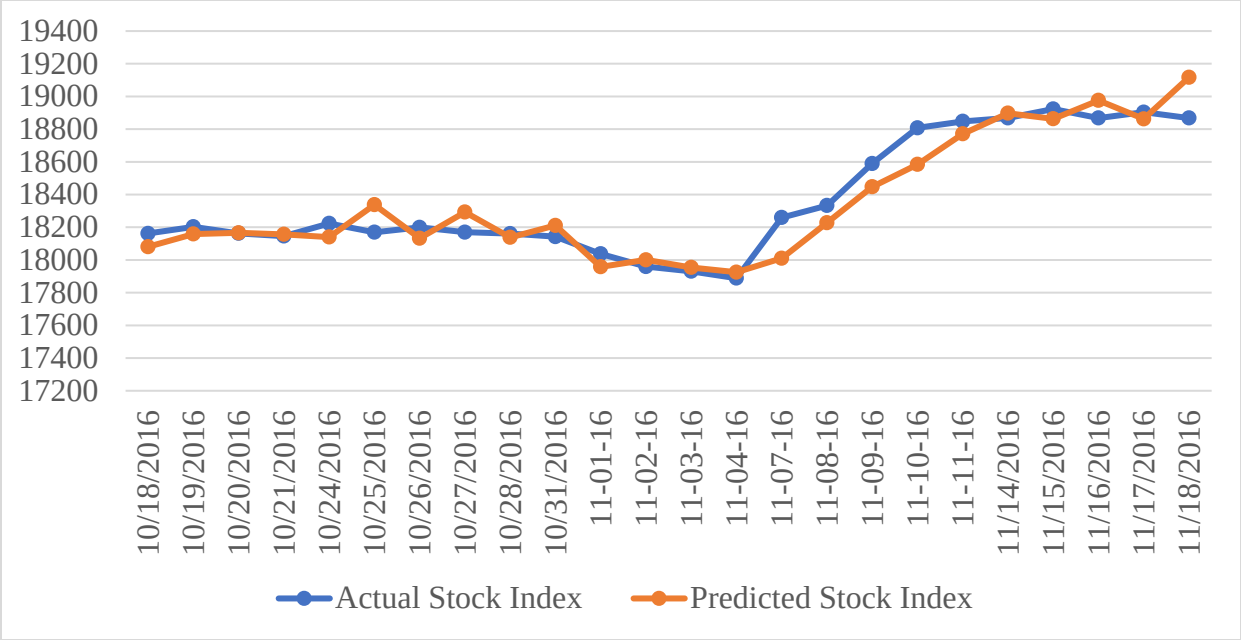


Figure 5.7: Actual stock index vs. Predicted stock index graph of the tweets containing 'stocktwits' from October,11 to November, 11 of 2016

Here, most of the values went really close through the actual values. Very few of them went far or in the wrong direction from the actual stock point.

The error metrics of the predictions is shown below-

Tweets with phrase	**Mean Absolute Error of Predicting Closing Stock price difference**	**RMSE**
'stock market'	82.95103119	116.1048303
'stocktwits'	103.3456242	131.3553
'AAPL'	1.201669284	1.731554777

Table 5.12: Error metrics

Although 'AAPL' seems to give less error, the stock data that we have used to work on the tweets containing 'AAPL' were only of the AAPL, not the overall stock points. As it shows, the approach with tweets containing 'stock market' performed the best to predict general stock market movement.

6. Discussion and Future Works

In our thesis work, we tried to predict the future movement of the stock market of United States (Dow-Jones) by analyzing Twitter sentiment about the stock market. Our initial plan was to do the work on Dhaka Stock Exchange (DSE), but Twitter is not as popular in Bangladesh as in United States. Moreover, there is no financial communication platform like StockTwits [22] (as a result of Twitter not being popular) in Bangladesh.

We have worked on the data of only one year with very little major fluctuations in stock market. Although one day prior predictions worked well, to predict the probable movement long before, we need data of longer timespan. Stock market data for longer timespan is available, but Twitter has restricted the historical tweet collection. Collecting historical tweets using Twitter API is now only possible by purchasing different subscription packages.

Alongside of that, Tweets are generally misspelled and often grammatically incorrect, which makes it harder to classify with traditional classifiers. In future, we will try to implement a boosted SVM classifier with logistic regression [17] for text and apply it on this dataset for better accuracy in sentiment classification.

We did not get very good results with specific companies. There is scope for progress there too. Only twitter data might not be much helpful in this case. We are planning to use stock market related news as another feature in this case for better prediction both for specific companies or the overall stock market.

Moreover, we did not use any neural networks, the most cutting edge learning tool in our work. It is mainly because, neural networks are widely used for image classification. We could not develop any text classification model using neural network. Based on [25], our future work would be to implement a sentiment classifier with RNN (recurrent neural network).

Lastly and most importantly, our work is not in a condition to use as a general user. We are planning to build an online tool, which will update itself with new market information and users can perceive market trends at home and decide whether to buy, sell or retain any stock. All these remains areas of future research.

7. References

[1] A. Mittal, A. Goel, “Stock Prediction Using Twitter Sentiment Analysis” in proceeding of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2013.

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

[3] 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.

[4] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford 1, 2009.

[5] 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.

[6] “A Twitter sentiment analysis,” 2013. [Online]. Available: http://www.sentiment140.com. Accessed: Nov. 2, 2016.

[7] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment Analysis of Twitter Data," LSM ’11 Proceedings of the Workshop on Languages in Social Media, pp. 30–38, Jun. 2011.

[8] Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In ACL-2004.

[9] Y. Kim, "Convolutional neural networks for sentence classification," 2014. [Online]. Available: https://arxiv.org/abs/1408.5882.

[10] T. Joachims, "Text categorization with support vector machines: Learning," 1998.

[11] E. Kouloumpis, T. Wilson, J. Moore, “Twitter Sentiment Analysis: The Good the Bad and the OMG!,” 2011.

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

[13] S.Wang, C.D. Manning, “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification,” 2012.

[14] Y. I. Chang, "Boosting SVM Classifiers with Logistic Regression,".

[15] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters," Computational Intelligence, vol. 22, no. 2, pp. 110–125, May 2006.

[16] P. Diamond, “Behavioral Economics,” Massachusetts Institute of Technology, Dept. of Economics, 2008.

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

[18] L. Breiman, “Arcing the Edge,”.

[19] “Your home for data science,” 2016. [Online]. Available: http://www.kaggle.com. Accessed: Nov. 2, 2016.

[20] “GraphLab create,” Turi, 2016. [Online]. Available: https://turi.com/products/create/. Accessed: Dec. 13, 2016.

[21] "API - sentiment140 - A Twitter sentiment analysis tool,". [Online]. Available: http://help.sentiment140.com/api. Accessed: Dec. 20, 2016.

[22] "StockTwits message", StockTwits, 2017. [Online]. Available: http://stocktwits.com. [Accessed: 03- Apr- 2017].

[23] P. Temin, "The Great Recession and the Great Depression", National Bureau of Economic Research, vol. 15645, 2010.

[24] B. Malkiel, "Efficient Market Hypothesis", The New Palgrave: Finance. Norton, New York, pp. 127-134, 1989.

[25] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification", Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.