A Minor Project Report on

SentiMeter: An Android Application for Sentiment Analysis of Twitter Data Using KNN and NBayes Classifiers

Submitted in Partial Fulfillment of the Requirements

for the Degree of **Bachelor of Engineering in Software Engineering** under Pokhara University

Submitted by:

Kanchan Singh, 161716

Poshan Pandey, 161724

Priska Budhathoki, 161726

Under the supervision of

Asst. Prof. **Himal Acharya**

Date:

9th Jan 2020

Department of Software Engineering

NEPAL COLLEGE OF INFORMATIONTECHNOLOGY

Balkumari, Lalitpur, Nepal

Abstract

Social media websites have emerged as one of the platforms to raise users' opinions and influence the way any business is commercialized. The opinion of people matters a lot to analyze how the propagation of information impacts the lives in a large-scale network like Twitter. Sentiment analysis of the tweets determines the polarity and inclination of the vast population towards a specific topic, item or entity. These days, the applications of such analysis can be easily observed during public elections, movie promotions, brand endorsements, and many other fields. This project extracts live tweets via creating an app using Twitter developer key and exploit the fast and in-memory computation of Twitter data using KNN (K-Nearest Neighbors) and NBayes (Naive Bayes) classifiers in Java to perform sentiment analysis. The app provides a method for analyzing sentiment score in noisy Twitter streams. The result of analysis of tweets is presented as the rate of positiveness. The accuracy of the system with Kaggle's movie review dataset resulted in 61.8335%.

Keywords: Java, KNN, NBayes, Twitter, Sentiment Analysis

Table of Contents

List of Figures

List of Tables

List of Abbreviations

FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbors
NBayes	Naïve Bayes
NLP	Natural Language Processing
TDM	Term Document Matrix
TF-IDF	Term Frequency Inverse Document Frequency
TN	True Negative
TP	True Positive
UI	User Interface

1 Introduction

1.1 Domain Introduction

Sentiment analysis is also known as “opinion mining” or “emotion Artificial Intelligence” and alludes to the utilization of natural language processing (NLP), text mining, computational linguistics, and bio measurements to methodically recognize, extricate, evaluate, and examine emotional states and subjective information. Sentiment analysis is generally concerned with the voice in client materials; for example, surveys and reviews on the Web and web-based social networks.

As the internet is growing bigger, its horizons are becoming wider. Social Media and Microblogging platforms like Facebook, Twitter, Tumblr dominate in spreading encapsulated news and trending topics across the globe at a rapid pace. A topic becomes trending if more and more users are contributing their opinion and judgments, thereby making it a valuable source of online perception. These projects generally intended to spread awareness.

1.2 Motivation

We have chosen to work with Twitter since we feel it is a better approximation of public sentiment as opposed to conventional internet articles and web blogs. The reason is that the amount of relevant data is much larger for Twitter, as compared to traditional blogging sites. Moreover, the response on Twitter is prompter and also more general (since the number of users who tweet is substantially more than those who write web blogs on a daily basis).

Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analyzing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm’s stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favorable response and in which a negative response (since Twitter allows us to download stream of geotagged tweets for particular locations).

If firms can get this information they can analyze the reasons behind the geographically differentiated response, and so they can market their product in a more optimized manner by looking for appropriate solutions like creating suitable market segments. Predicting the results of popular political elections and polls is also an emerging application to sentiment analysis.

1.3 Problem Statement

Sentiment analysis in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from Twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

1.4 Project Objective

- Extract tweets and preprocess them.
- Estimate the sentiment of those tweets.

1.5 Project Scope and Limitation

This project will be helpful to the companies, political parties as well as to the common people. It will be helpful to political party for reviewing about the program that they are going to do or the program that they have performed. Similarly, companies also can get review about their new product on newly released hardware or software. Also, the movie maker can take review on the

currently running movie. By analyzing the tweets analyzer can get result on how positive or negative or neutral are people about it

Some limitations of this project are:

- Cannot identify humor and sarcasm.
- The current classifier does not consider the neutral sentiments.
- Does not consider the context of tweets.

1.6 Significance of the study

Sentiment Analysis of Twitter Dataset has a number of applications like promotion, politics, election, etc. Twitter Sentiment Analysis can be used to develop their business strategies, to assess customers' feelings towards products or brand, how people respond to their campaigns or product launches and also why consumers are not buying certain products. In politics, Twitter Sentiment Analysis is used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level. Sentiment Analysis Dataset is also used for analyzing election results. Twitter Sentiment Analysis is also used for monitoring and analyzing social phenomena for predicting potentially dangerous situations and determining the general mood of the blogosphere.

1.7 Report Organization

In first chapter we introduced our project. We mentioned its motivation, problem statement, objective, significance and its scope. In Second chapter we will talk about the similar projects. Third chapter is about the methodology used to implement the project. Fourth chapter is about how those methodology is actually implemented in this project. Fifth chapter is about the work division and time schedule. Sixth chapter includes the result and discussion of this project. Seventh chapter concludes and mentions the future works for this project.

2 Literature Review

This section summarizes some of the scholarly and research works in the field of Machine Learning and data mining to analyze sentiments on the Twitter and preparing prediction model for various applications. As the available social platforms are shooting up, the information is becoming vast and can be extracted to turn into business objectives, social campaigns, marketing and other promotional strategies [1]. The benefit of social media to know public opinions and extract their emotions are considered by authors [2] and explained how Twitter gives advantage politically during elections. Further, the concept of the hashtag is used for text classification as it conveys emotion in few words. They suggested how previous research work suffered from lack of training set and misses some features of target data. They opted two-stage approach for their framework- first preparing training data from Twitter using mining conveying relevant features and then propounding the Supervised Learning Model to predict the results of elections held in the USA in 2016. After collecting and preprocessing the tweets, training data set was created first by manual labeling of hashtags and forming clusters, next by using online Sentimental Analyzer VADER which outputs the polarity in percentage. This approach reduced the number of tweets or training set and further they applied Support Vector Machine and Naive Bayes classification algorithm to determine the polarity of tweets. Multistage classification approach was used where an entity classifier receives a general class of tweets and categorize them with respect to individual candidates for comparison. The metric they used to determine the winner was the “Pvt ratio” which is a Positive number of tweets to the total count of tweets for respective candidate.

Sentiment Analysis by researchers Imran et al. [3] exploited the technology 'Apache Spark' for fast streaming of tweets and presented the approach Stream Sensing to handle real-time data in the unstructured and noisy form. They conducted the approach on Twitter data to find some useful and interesting trends which further can be generalized to any real-time text stream. The unsupervised learning approach is used to locate interesting patterns and trends from tweets processed on Apache Spark. Inspired by the approach described by Zhu et al. [4] and Li et al. [5] for mining data by selecting time window, authors [3] opted for sliding window method for capturing the live streams of tweets. The common approach found in almost all relevant research works constitutes data collection using Twitter API, preprocessing of data, filtering of data then

approaches in feature extraction, classification and pattern analysis makes the distinction. Authors used a sliding window of 5 minutes during data collection and further created Term Document Matrix (TDM) for feature extraction. The pattern analysis was carried out by using the score of TF-IDF for finding the most important keywords as explained by Wu et al [5]. The trending topic or hashtag is fed and tweets relevant to it are filtered to form TDM and computing the weights of TF-IDF to find the most important words is the key idea of this sentiment analysis.

3 Methodology

3.1 Flowchart

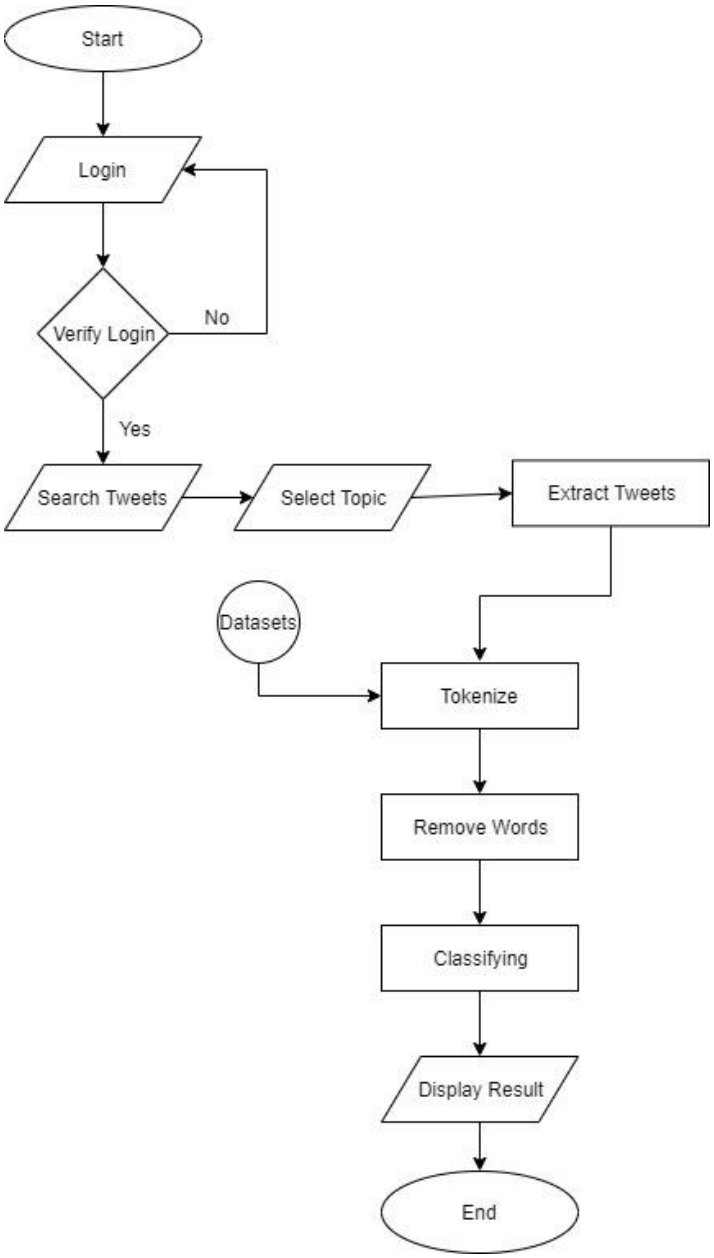


Figure 1: Flowchart

3.2 Use Case Diagram

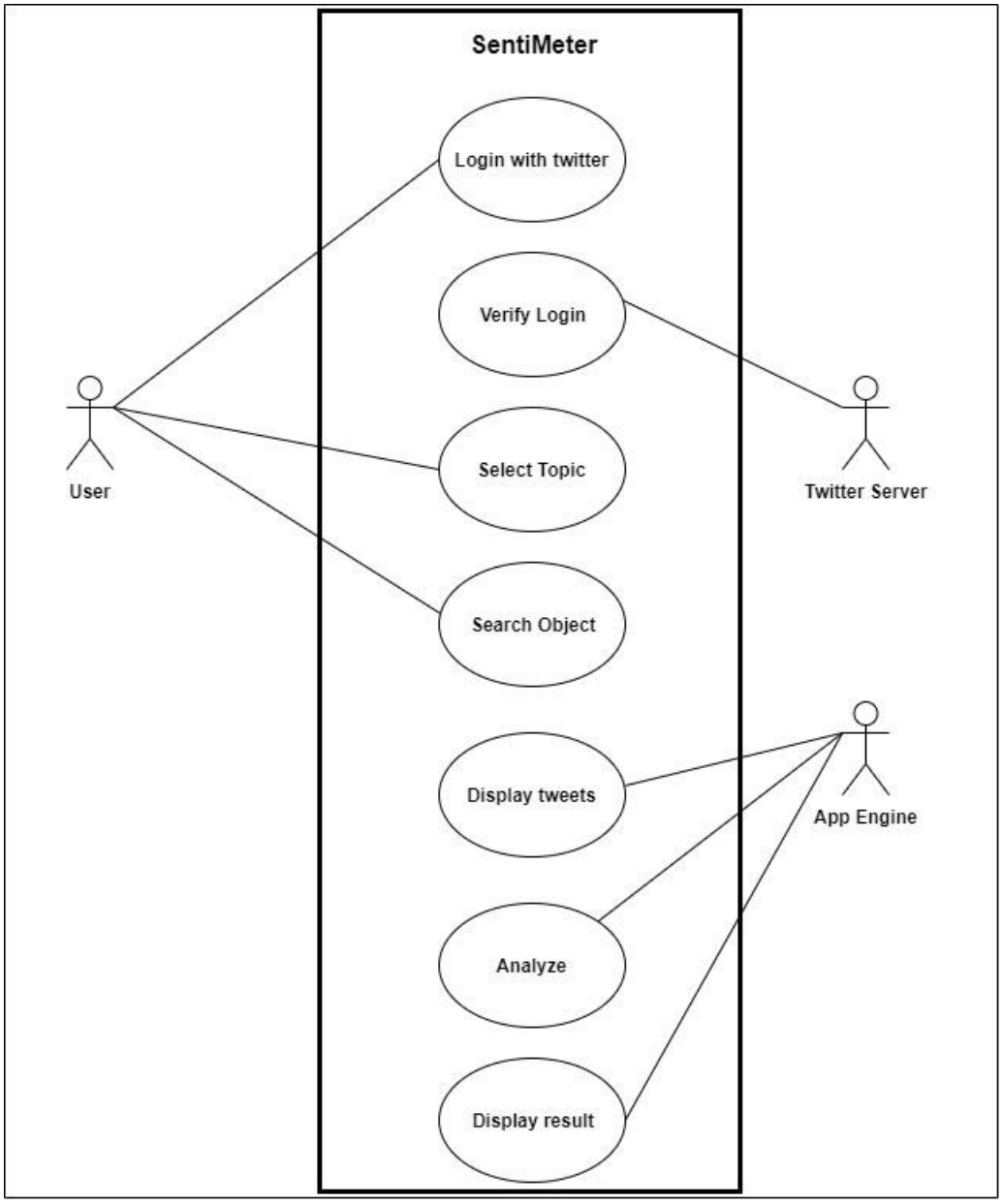


Figure 2: Use Case Diagram

3.3 Software Development Life Cycle: Waterfall Model

In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. The above illustration is a representation of the different phases of the Waterfall Model. We will be using the waterfall model approach for the development of our project. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure the success of the project.

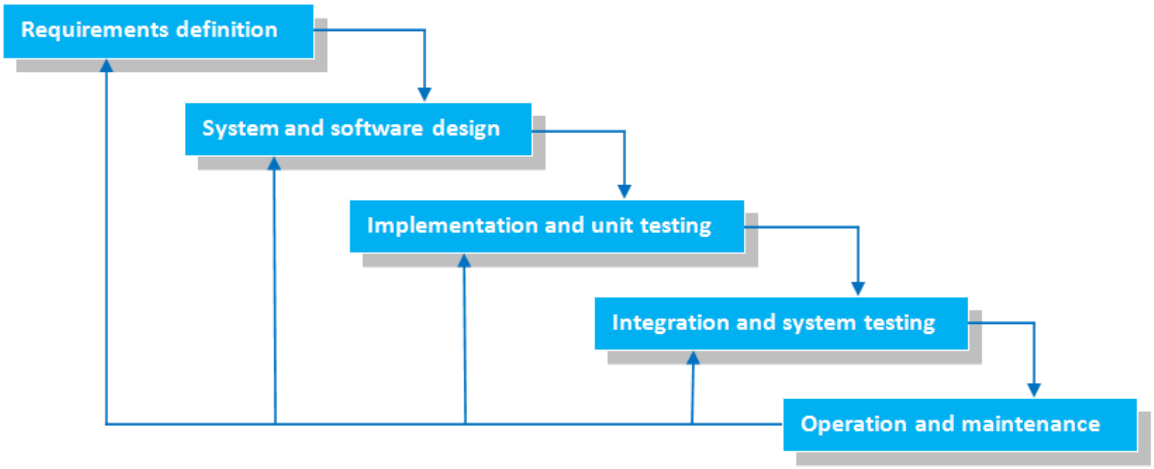


Figure 3: Waterfall Model

The sequential phases in the Waterfall model are:

- **Requirement definition -** All possible requirements of the system to be developed were captured in this phase and documented properly.
- **System and Software Design** – In this phase we designed our basic UI and imported the Twitter data.
- **Implementation and unit testing** – In this stage we implemented our models in the UI and analyzed tweets using those models.
- **System Testing** – We tested our project for several time using different values of k or different datasets in this phase.

- **Deployment of system** – Finally we extracted the APK file of this project in this phase.
- **Operation and Maintenance** – We then refined our project for bugs at this stage.

3.4 Classifiers Used

3.4.1 Naïve Bayes (NBayes):

This is a classification method that relies on Bayes' Theorem with strong (naive) independence assumptions between the features. A Naive Bayes classifier expects that the closeness of a specific feature (element) in a class is disconnected to the closeness of some other elements. For instance, an organic fruit might be considered to be an apple if its color is red, its shape is round and it measures approximately three inches in breadth. Regardless of whether these features are dependent upon one another or upon the presence of other features, a Naïve Bayes classifier would consider these properties independent due to the likelihood that this natural fruit is an apple. Alongside effortlessness, the Naive Bayes is known to out-perform even exceedingly modern order strategies. The Bayes hypothesis is a method of computing for distinguishing likelihood P(a|b) from P(a), P(b) and P(b|a) as follows:

$$P\left(\frac{a}{b}\right) = \frac{P\left(\frac{b}{a}\right) * P(a)}{P(b)}$$

Where P(a/b) is the posterior probability of class given as given predictor b and P(b/a) is the likelihood that is the probability of predictor b given the class a. The prior probability of given class a is denoted by p(a) and that of predictor b is P(b). The Naïve Bayes is widely used in the task of classifying texts into multiple classes and was recently utilized for sentiment analysis classification.

3.4.2 K-Nearest Neighbors (KNN):

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

- Ease to interpret the output
- Calculation time
- Predictive Power

Let us take a few examples to place KNN in the scale:

KNN algorithm fairs across all parameters of considerations. It is commonly used for its ease of interpretation and low calculation time.

Let's take a simple case to understand this algorithm. Following is a spread of circles (RC) and squares (GS):

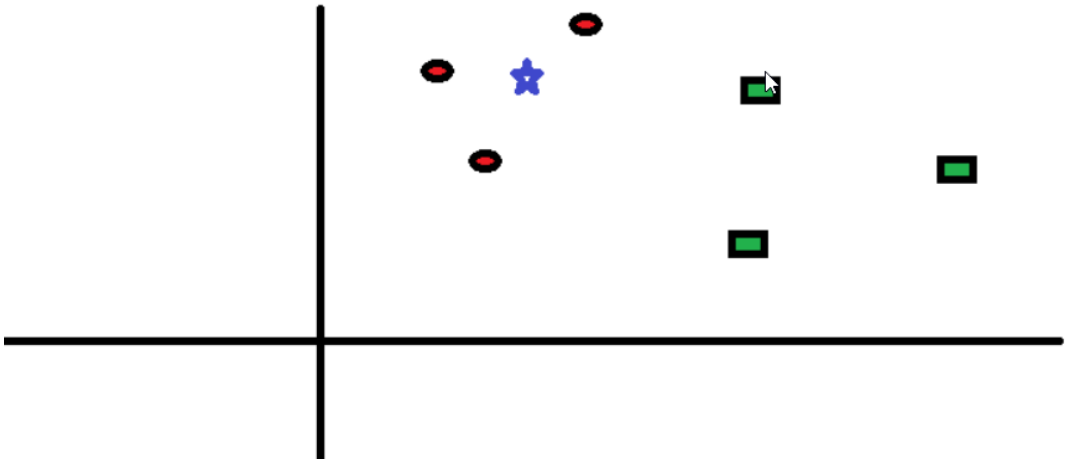


Figure 4: Part A;How KNN Works?

You intend to find out the class of the star (BS). BS can either be RC or GS and nothing else. The "K" is the KNN algorithm is the nearest neighbors we wish to take a vote from. Let's say K = 3. Hence, we will now make a circle with BS as center just as big as to enclose only three data points on the plane. Refer to the following diagram for more details:

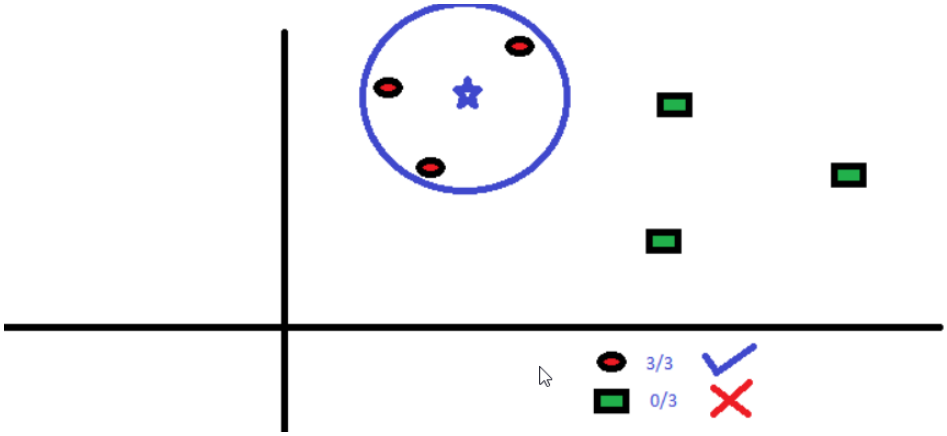


Figure 5: Part B; How KNN Works?

The three closest points to BS are all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm.

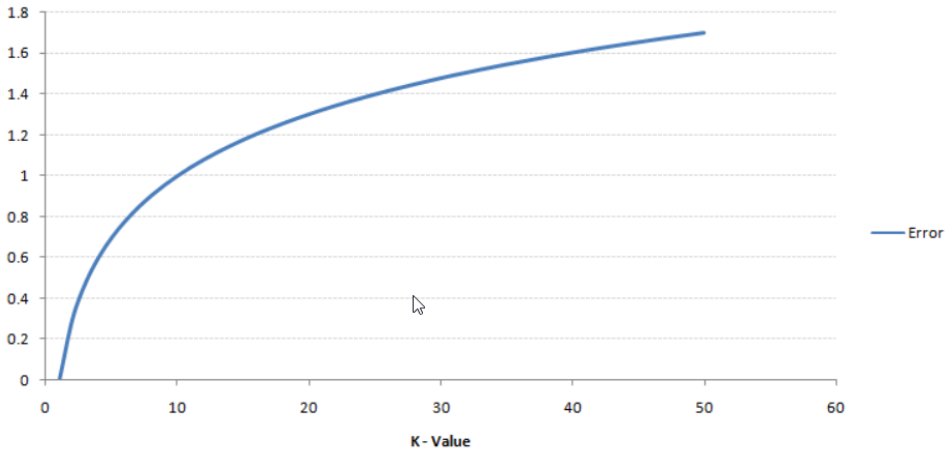


Figure 6: Relation between value of K and error

3.5 Dataset

For the dataset we used the dataset of popular Twitter sentiment analysis project of Mr. Jeffrey Breen for Kaggle [6]. As well as to improve the accuracy we used political [7] and IMDB labelled [8] datasets for specific topics.

To check the accuracy of our classifier we used the dataset of movie reviews from Kaggle [9]. For Instance, our positive keyword datasets are:

```
a+ abound abounds abundance abundant accessable accessible acclaim
acclaimed acclamation accolade accolades accommodative accomodative
accomplish accomplished accomplishment accomplishments accurate
accurately achievable achievement achievements achievible acumen
adaptable adaptive adequate adjustable admirable admirably admiration
admire admirer admiring admiringly adorable adore adored adorer adoring
adoringly adroit adroitly adulate adulation adulatory advanced advantage
advantageous advantageously advantages adventuresome adventurous
advocate advocated advocates affability affable affably affectation
affection affectionate affinity affirm affirmation affirmative affluence
affluent afford affordable affordably afordable agile agilely agility
agreeable agreeableness agreeably all-around alluring alluringly
altruistic altruistically amaze amazed amazement amazes amazing
amazingly ambitious ambitiously ameliorate amenable amenity amiability
amiabily amiable amicability amicable amicably amity ample amply amuse
amusing amusingly angel angelic apotheosis appeal appealing applaud
appreciable appreciate appreciated appreciates appreciative
appreciatively appropriate approval approve ardent ardently ardor
articulate aspiration aspirations aspire assurance assurances assure
assuredly assuring astonish astonished astonishing astonishingly
astonishment astound astounded astounding astoundingly astutely
attentive attraction attractive attractively attune audible audibly
auspicious authentic authoritative autonomous available aver avid avidly
award awarded awards awe awed awesome awesomely awesomeness awestruck
awsome backbone balanced bargain beauteous beautiful beautifullly
```

Figure 7: Positive Keywords Datasets

3.6 Accuracy

3.6.1 Confusion Matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix is used. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.

Table 1: Confusion Matrix

	Predicted Values		
Actual Values		**Positive**	**Negative**
	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

4 System Implementation

4.1 System Architecture

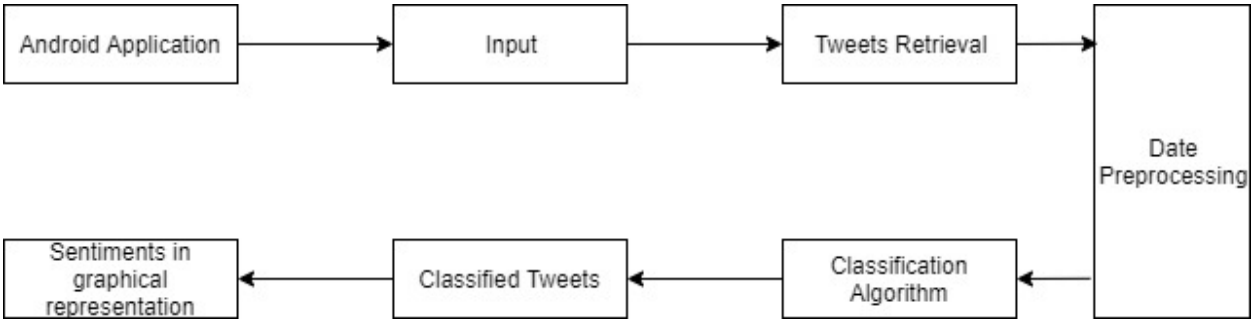


Figure 8: System Architecture

4.1.1 Android Application

The application is made for the android operating system. The result is displayed in one of the pages. For developing application, we used XML and Java for android development.

4.1.2 Input

At this stage user will input some keywords for searching tweets about that keywords.

4.1.3 Tweets Retrieval

Data in the form of raw tweets is acquired by using the Twitter developer's API which provides a package for simple Twitter streaming API. The User Interface for the list of the tweets is provided by the Twitter. The retrieved tweet is at first tested with the pretrained datasets from Kaggle. If it matches more than 60% then its positiveness is directly passed to KNN else it is sent to the tokenizer.

4.1.4 Data Preprocessing

All the tweets are at first translated into lowercase then symbols like (!, ?, # etc.) are removed. Then irresponsible words are removed. For Example, we have a tweet “I love PM KP Oli !!!”. It is preprocessed as “i love pm kp oli”

4.1.5 Classification Algorithm

We used two classification algorithms as follows:

4.1.5.1 NBayes

NBayes classifier is used to classify the words from Twitter to express their positivity or negativity for each tweet. Here the model is trained using the datasets. Irresponsible words are removed and score for each word is calculated to be either 1 or 0 comparing with the datasets. For above example NBayes classifies it as “love 1”.

4.1.5.2 KNN

After the NBayes classifier finds the score of each words KNN then analyzes the various distances like taxicab, Euclidian, hamming and finds out the positivity or the negativity of the tweets as a whole i.e. The score of all the tweets. While experimenting the values of K, Maximum accuracy is acquired at K = 3. For above example the KNN final value is automatically 1 since it has only one value. Then for all the tweets KNN compares all the tweets using Taxicab, Hamming and Euclidian distance and measures its positiveness.

4.1.6 Classified Tweets

The classified tweets score is then passed to a graphical model that displays the result.

4.1.7 Graphical Representation

The received result is the displayed in result page in graphical format.

4.2 Tools Used

Table 2: Tools Used

S. N	Tools	Purpose
1	Android Emulator	Testing App
2	Android Studio	IDE for Android Development
3	Draw.io	Drawing Charts and tables
4	Figma	Designing UI look
5	GitHub	Managing Team work
6	Microsoft Visio	For Gantt Chart

4.3 Technologies Used

Table 3: Technologies Used

S. N	Technology	Purpose
1	Java	For android development and algorithm implementation.
2	Twitter Developer API	For Twitter data extraction
3	XML	For android UI development.

5 Project Task and Time Schedule

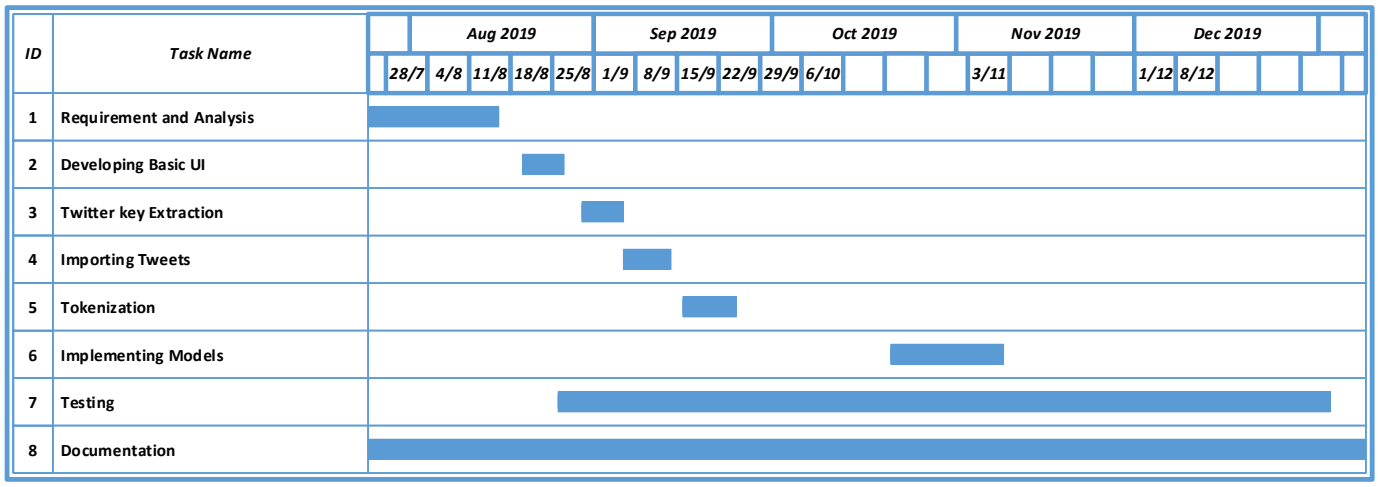
5.1 Work Division

Table 4: Work Division

S. N	Task Name	Kanchan Singh	Poshan Pandey	Priska Budhathoki
1	Requirement Analysis	✓	✓	✓
2	Developing Basic UI	✓		✓
3	Twitter Key Extraction		✓	
4	Importing Tweets		✓	
5	Tokenization		✓	
6	Implementing Models	✓		✓
7	Coordinating Models		✓	
8	Testing	✓	✓	✓
9	Documentation	✓	✓	✓

5.2 Gantt Chart

Table 5: Gantt Chart



6 Result and Discussion

6.1 Final Outcomes

Finally, the proposed project is completed. The project is able to analyze the tweets and obtain the sentiment of those tweets. Here are the screenshots on how our product works.

At first user should login via their Twitter account as follow:

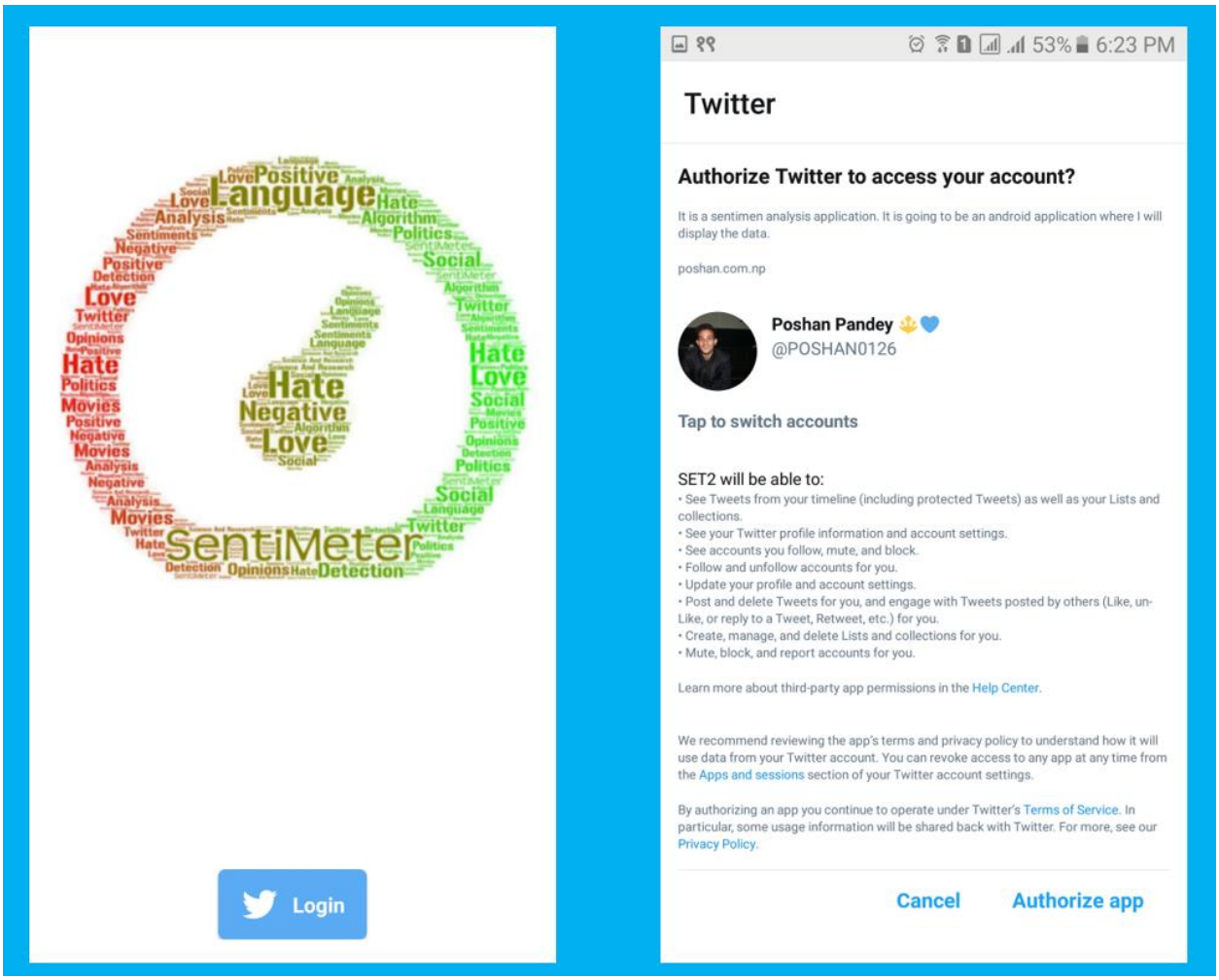


Figure 9: Logging in with Twitter Account

Then user should select a specific topic i.e. either movies or politics for now. Topic is included so that tweets with similar pattern can be analyzed fast comparing with datasets [7][8]. Then user should search the keyword as follow:

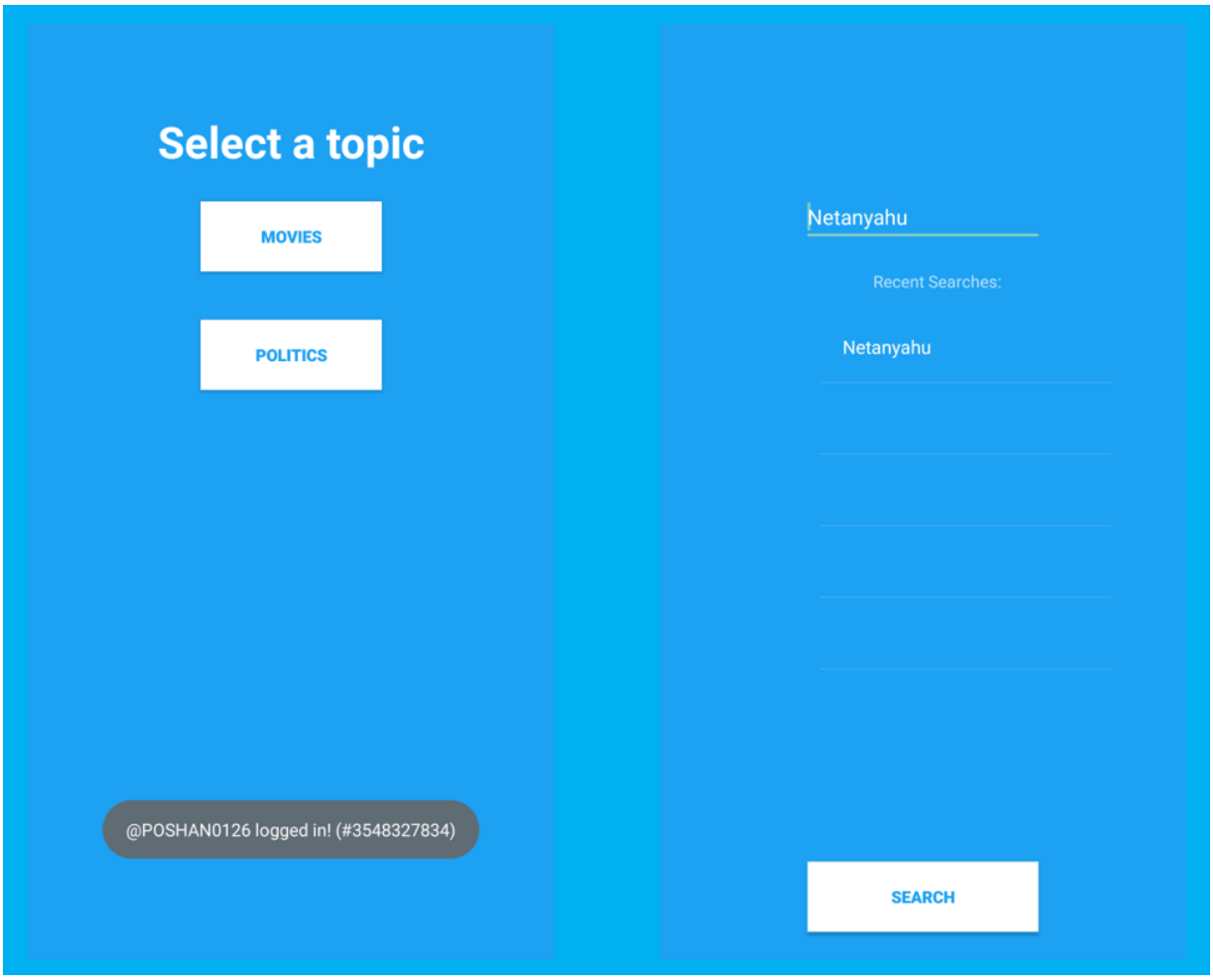


Figure 10: Selecting Topic and searching key

After that the latest tweets of that topic is showed to the user and when they press the analyze button, result is displayed as below:

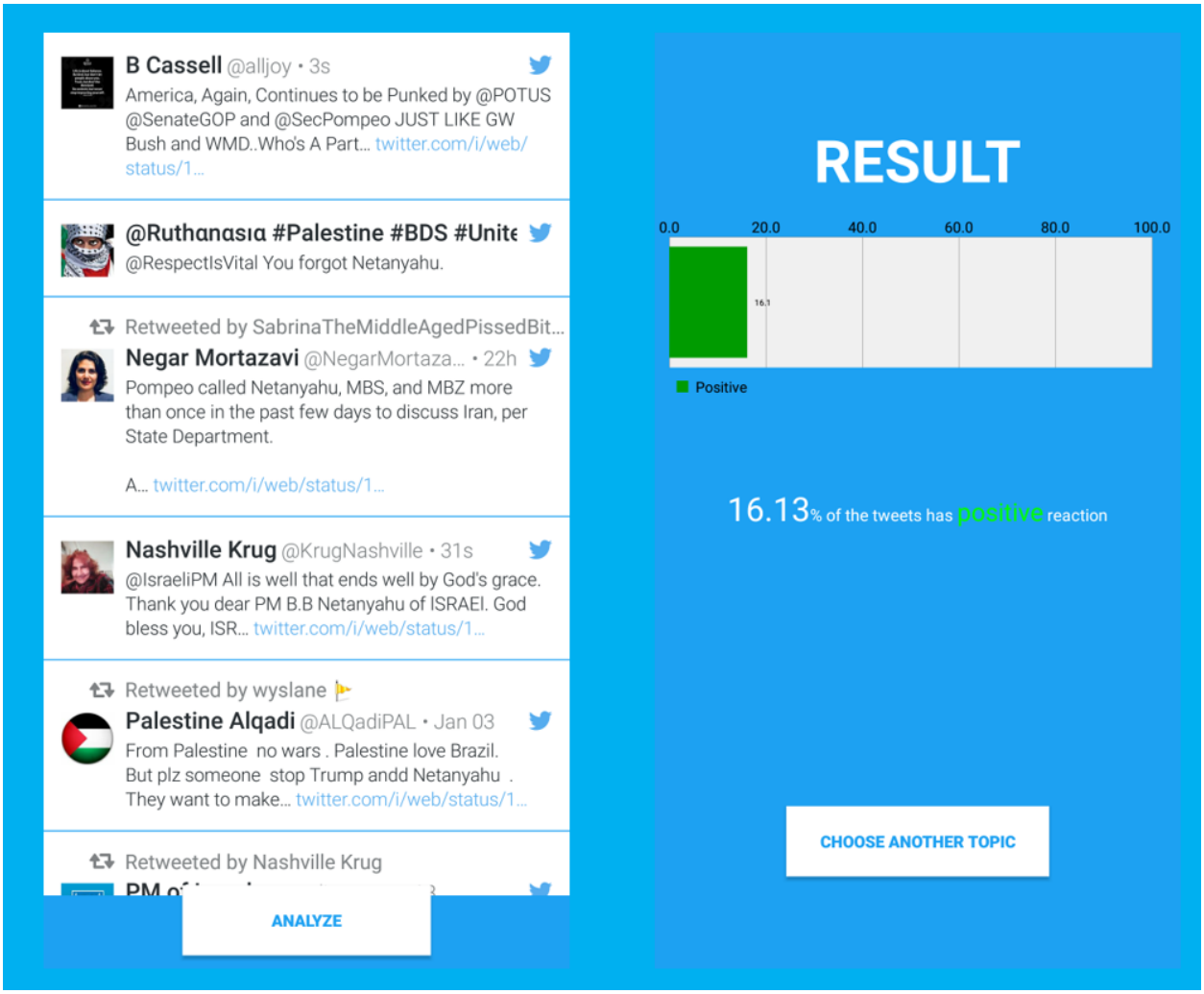


Figure 11: Displaying Tweets and then Result

6.2 Accuracy

We used our model to check the sentiment of movies reviews datasets from Kaggle in which there were 50% positive review and 50% negative review [9]. The calculated results are as follow:

Table 6: Calculated result for Movie Review dataset

	Predicted		
Actual		**Positive**	**Negative**
	Positive	**82.667%**	**17.333%**
	Negative	**59%**	**41%**

The calculated accuracy for the used datasets is 61.8335%.

7 Conclusion and Future Work

The outcome of the project is a furnished mobile application. The mobile application is able to retrieve tweets and analyze those tweets to calculate the sentiment. The result is presented as the rate of positiveness. SentiMeter, upon testing with Kaggle's movie review dataset, resulted in 61.8335% accuracy.

7.1 Future Work

The task of sentiment analysis, especially in the domain of micro-blogging, is still in the developing stage and far from complete. So, we propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance. Right now, we have worked with only the very simplest unigram models; we can improve those models by adding extra information like closeness of the word with a negation word. We could specify a window prior to the word (a window could for example be of 2 or 3 words) under consideration and the effect of negation may be incorporated into the model if it lies within that window. The closer the negation word is to the unigram word whose prior polarity is to be calculated, the more it should affect the polarity.

For example, if the negation is right next to the word, it may simply reverse the polarity of that word and farther the negation is from the word the more minimized ifs effect should be. Apart from this, we are currently only focusing on unigrams and the effect of bigrams and trigrams may be explored. As reported in the literature review section when bigrams are used along with unigrams this usually enhances performance.

8 References

[1] Mtibaa, M. May, C. Diot and M. Ammar, "PeopleRank: Social Opportunistic Forwarding", 2010 Proceedings IEEE INFOCOM, 2010.

[2] J. Ramteke, S. Shah, D. Godhia, and A. Shaikh, “Election result prediction using Twitter sentiment analysis,” in Inventive Computation Technologies (ICICT), International Conference on, 2016, vol. 1, pp. 1–5.

[3] Dr. Khalid N. Alhayyan & Dr. Imran Ahmad “Discovering and Analyzing Important Real-Time Trends in Noisy Twitter Stream” n.p

[4] Li, H.-F. and Lee, S.-Y. (2009). Mining frequent itemsets over data streams using efficient window sliding techniques. Expert Syst. Appl. 36, 2, 1466–1477.

[5] Alexander Pak and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining". In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), May 2010.

[6] https://github.com/jeffreybreen/Twitter-sentiment-analysis-tutorial-201107/tree/master/data/opinion-lexicon-English [Accessed: 5:35 PM 29th December]

[7] https://www.kaggle.com/yemregundogmus/turkey-political-opinions [Accessed: 5:40 PM 29th December]

[8] https://www.kaggle.com/mwallerphunware/imbd-movie-reviews-for-binary-sentiment-analysis [Accessed: 5:45 PM 29th December]

[9] https://www.kaggle.com/blanderbuss/positive-and-negative-movies-reviews/ [Accessed: 5:35 PM 29th December]