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**

**INFORMATION**

**TECHNOLOGY**

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. In this project, we are going to extract live tweets via creating an app using Twitter developer key and we are going to exploit the fast and in-memory computation of Twitter data using classifiers KNN (K-Nearest Neighbors) and NBayes (Naive Bayes) in Java to perform sentiment analysis. The primary aim is to provide a method for analyzing sentiment score in noisy twitter streams from android application. This paper reports on the design of sentiment analysis and extracting a vast number of tweets. Results classify user's perception via tweets into positive and negative. Secondly, we discuss various techniques to carry out a sentiment analysis on twitter data in detail.

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

**Table of Contents**

[1 Introduction 1](#_Toc29462363)

[1.1 Domain Introduction 1](#_Toc29462364)

[1.2 Motivation 1](#_Toc29462365)

[1.3 Problem Statement 2](#_Toc29462366)

[1.4 Project Objective 2](#_Toc29462367)

[1.5 Project Scope and Limitation 3](#_Toc29462368)

[1.6 Significance of the study 3](#_Toc29462369)

[1.7 Report Organization 4](#_Toc29462370)

[2 Literature Review 5](#_Toc29462371)

[3 Methodology 7](#_Toc29462372)

[3.1 Flowchart 7](#_Toc29462373)

[3.2 Use Case Diagram 8](#_Toc29462374)

[3.3 Software Development Life Cycle: Waterfall Model 9](#_Toc29462375)

[3.4 Classifiers Used 10](#_Toc29462376)

[3.4.1 Naïve Bayes (NBayes): 10](#_Toc29462377)

[3.4.2 K-Nearest Neighbors (KNN): 10](#_Toc29462378)

[3.5 Dataset 12](#_Toc29462379)

[3.6 Accuracy 13](#_Toc29462380)

[3.6.1 Confusion Matrix 13](#_Toc29462381)

[4 System Implementation 14](#_Toc29462382)

[4.1 System Architecture 14](#_Toc29462383)

[4.1.1 Input 14](#_Toc29462384)

[4.1.2 Tweets Retrieval 14](#_Toc29462385)

[4.1.3 Data Preprocessing 15](#_Toc29462386)

[4.1.4 Classification Algorithm 16](#_Toc29462387)

[4.1.5 Classified Tweets 18](#_Toc29462388)

[4.1.6 Graphical Representation 18](#_Toc29462389)

[4.2 Tools Used 19](#_Toc29462390)

[4.3 Technologies Used 19](#_Toc29462391)

[5 Project Task and Time Schedule 20](#_Toc29462392)

[5.1 Work Division 20](#_Toc29462393)

[5.2 Gantt Chart 21](#_Toc29462394)

[6 Result and Discussion 22](#_Toc29462395)

[6.1 Final Outcomes 22](#_Toc29462396)

[6.2 Accuracy 24](#_Toc29462397)

[7 Conclusion and Future Scope 26](#_Toc29462398)

[8 References 27](#_Toc29462399)

**List of figures**

[Figure 1: Flowchart 7](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462400)

[Figure 2: Use Case Diagram 8](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462401)

[Figure 3: Waterfall Model 9](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462402)

[Figure 4: Part A;How KNN Works? 11](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462403)

[Figure 5: Part B: How KNN Works? 12](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462404)

[Figure 6: Relation between value of K and error 12](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462405)

[Figure 7: System Architecture 14](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462406)

[Figure 8: Logging in with Twitter Account 22](#_Toc29462407)

[Figure 9: Selecting Topic and searching key 23](#_Toc29462408)

[Figure 10: Displaying Tweets and then Result 24](file:///D:\Desktop\New%20folder\Introduction.docx#_Toc29462409)

**List of Tables**

[Table 1: Confusion Matrix 13](#_Toc29462410)

[Table 2: Tools Used 19](#_Toc29462411)

[Table 3: Technologies Used 19](#_Toc29462412)

[Table 4: work Division 20](#_Toc29462413)

[Table 5: Gantt Chart 21](#_Toc29462414)

[Table 6: Calculated result for Movie Review dataset 25](#_Toc29462415)

**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

# Introduction

## 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.

Large organizations and firms take advantage of people's feedback to improve their products and services which further help in enhancing marketing strategies. One such example can be leaking the pictures of the upcoming iPhone to create a hype to extract people's emotions and market the product before its release. Thus, there is a huge potential of discovering and analyzing interesting patterns from the infinite social media data for business-driven applications.

## 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.

## Problem Statement

Sentiment analysis of 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.

## Project Objective

The objective of this project is to extract data from twitter and use those data to find the real-time trend and the opinion of the public so that to use them in business objectives, social campaigns, marketing, and other promotional strategies. It can be used during elections, movie premier, promotions, etc. to find the opinions of the audience or public and act accordingly. Our aim is to provide the people with a means to find the opinion of the public about their product or ideology or principle.

## 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 3 on the currently running movie. By analyzing the tweets analyzer can get result on how positive or negative or neutral are peoples 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.

## 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.

## 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.
* Finally, eighth chapter includes the references of this report.

# 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 as explained in [1]. The benefit of social media to know public opinions and extract their emotions are considered by authors in [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.

# Methodology

## Flowchart

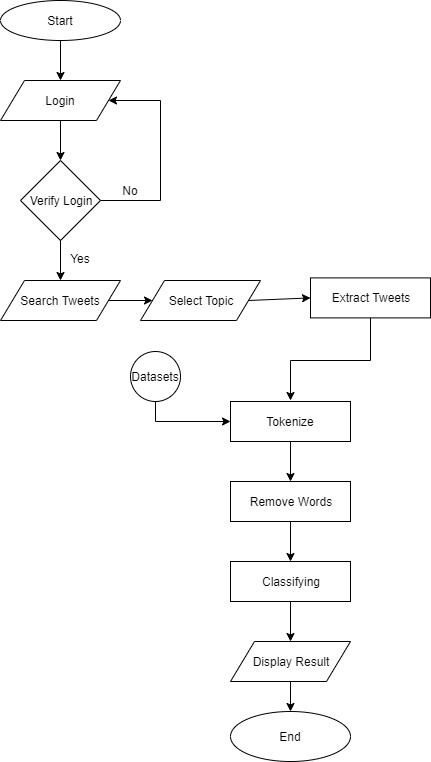


Figure 1: Flowchart

## Use Case Diagram

Figure 2: Use Case Diagram

## 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.

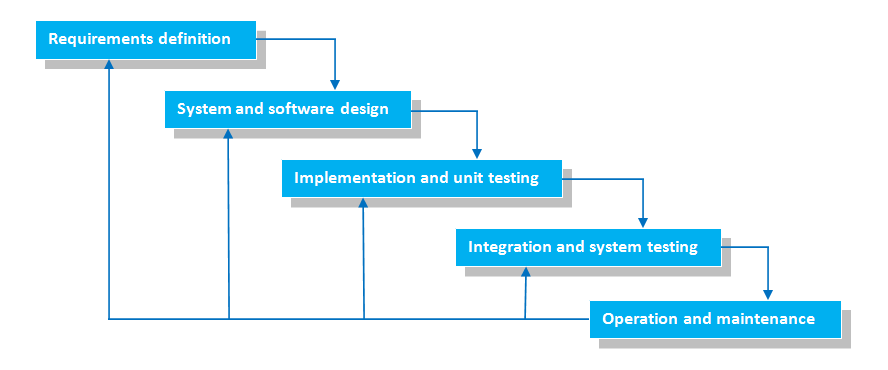


Figure 3: Waterfall Model

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.

## Classifiers Used

### 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:

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.

### 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.

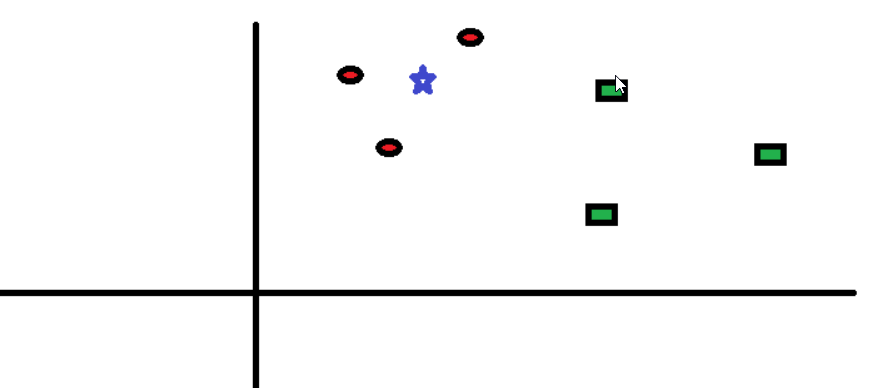
Let’s take a simple case to understand this algorithm. Following is a spread of circles (RC) and squares (GS):

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:

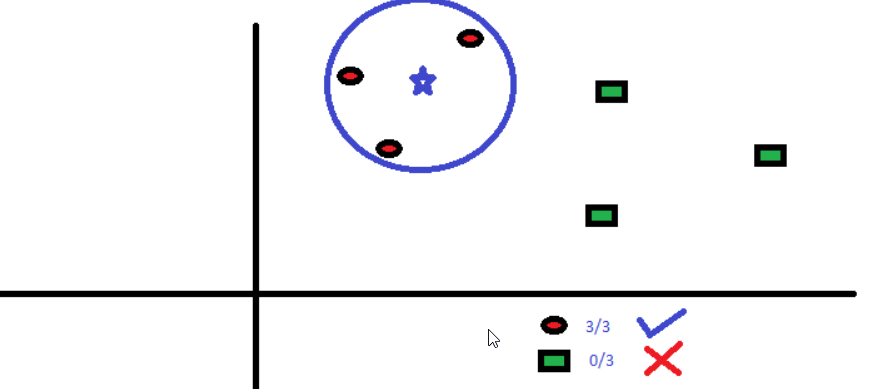
   
The three closest points to BS are all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm.

Figure 5: Part B: How KNN Works?

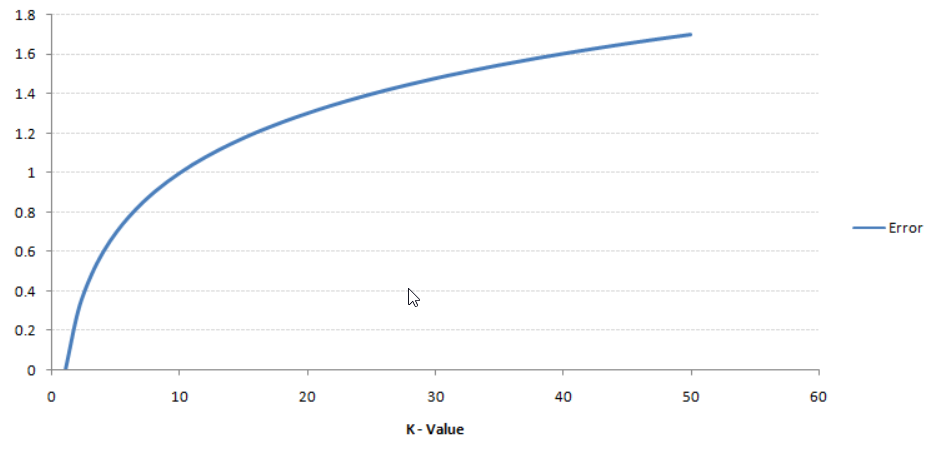


Figure 6: Relation between value of K and error

## 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].

## Accuracy

### 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** |

# System Implementation

## System Architecture

Figure 7: System Architecture

### Input

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

### 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. Here is the code snippet for importing the twitter data. For Now we have set our limit to 200 tweets but that can be extended as required.

**import** com.twitter.sdk.android.Twitter;  
**import** com.twitter.sdk.android.core.Callback;  
**import** com.twitter.sdk.android.core.Result;  
**import** com.twitter.sdk.android.core.TwitterAuthConfig;  
**import** com.twitter.sdk.android.core.TwitterException;  
**import** com.twitter.sdk.android.core.TwitterSession;  
**import** com.twitter.sdk.android.core.identity.TwitterLoginButton;

**private static final** String ***TWITTER\_KEY*** = **"FVMeO6JX5EpC2sSaFzOVQvpgx"**; **private static final** String ***TWITTER\_SECRET*** =**"s4y5fcRqtmzVoyiM5efetp2IWgl9cXm6Ky4hf9q3CE3an8D8CJ"**;

### Data Preprocessing

At this stage we preprocess the tweets. We tokenize the tweets and remove the symbols like (#, ., ,, ) .A small code snippet at this stage is:

**public static void** main(String[] args,Context ctx){  
  
*/\* Intialize the Key array \*/* Key[] keys = **new** Key[4000];  
 **for**(**int** j=0;j<keys.**length**;j++){  
 keys[j] = **new** Key();  
 }  
  
 **try**{  
 *// FileInputStream fis = new FileInputStream(args[0]);* FileInputStream fis = ctx.openFileInput(args[0]);  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(fis));  
 **int** cl\_type, i=0, j;  
 String line = **null**;  
  
 **while** ((line = br.readLine()) != **null**) {  
  
 */\* Input from file is dependent on format of source \*/* String[] columndet = line.split(**"\t"**);  
 columndet[0] = columndet[0].toLowerCase();  
 *//System.out.println("String: " + columndet[0]);* cl\_type=Integer.*parseInt*(columndet[1]);  
 *//System.out.println("Class Type:" + cl\_type);* String delims = **"[ -.,?!]+"**;  
 columndet[0]=columndet[0].replaceAll(**"not "**,**"not\_"**);  
 String[] tokens = columndet[0].split(delims);  
  
 */\* token[] stores the array of Keywords \*/* **for**(String s : tokens){  
 **if**(s.length()>1 && !*isSilly*(s,ctx)){  
 **if**(s.length()>1 && (i==0 || *exists*(keys,s)==-1) ){  
 *//System.out.println(s + s.length());* keys[i].**key**=s;  
 keys[i].**freq** += 1;  
 i++;  
 }  
 **else**{  
 keys[*exists*(keys,s)].**freq** += 1;  
 }  
 }  
 }  
 }

### Classification Algorithm

We used two classification algorithms as follows:

#### 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. Code snippet of NBayes classifier is:

**public class** nbayes{  
  
 **public static double** cprob(**double** x[], **double**[] y){  
 */\* x is train y is test \*/* **double** d=1, diff;  
 **for**(**int** i=0;i<x.**length**;i++){  
 **if**((x[i]!=0 && y[i]!=0) )  
 d \*= (1000-i)\*y[i]\*x[i];  
 }  
 **return** d;  
 }  
  
 **public static int** main(**double**[][] d,**double**[] test){  
  
  
 **double** d0=0, d1=0;  
  
 d0 = *cprob*(d[0],test);  
 d1 = *cprob*(d[1],test);  
  
 **if**(d0>d1)  
 **return** 0;  
 **else  
 return** 1;  
  
 }  
}

#### 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. Code snippet of the KNN is:

**public class** knn{  
  
 **public static double** taxicab(**double** x[], **double**[] y){  
  
 **double** dist=0, diff;  
 **for**(**int** i=0;i<x.**length**;i++){  
 diff = Math.*abs*(x[i]-y[i])\*(1000-i);  
 }  
  
 **return** dist;  
 }  
  
 **public static double** euclidian(**double** x[], **double**[] y){  
  
 **double** dist=0, diff;  
 **for**(**int** i=0;i<x.**length**;i++){  
 diff = (x[i]-y[i]);  
 dist += diff\*diff\*(1000-i);  
 }  
  
 dist = Math.*sqrt*(dist);  
  
 **return** dist;  
 }  
  
 **public static double** hamming(**double** x[], **double**[] y){  
  
 **double** dist=0, diff;  
 **for**(**int** i=0;i<x.**length**;i++){  
 **if**((x[i]!=0 && y[i]!=0) || (x[i]==0 && y[i]==0))  
 diff = 0;  
 **else** diff = 1000-i;  
 dist += diff;  
 }  
 **return** dist;  
 }  
  
 **public static int** main(**double**[][] d,**double**[] test,String S){  
  
 **double** d0=0, d1=0;  
  
 **if**(S.equals(**"euclidian"**)){  
 d0 = *euclidian*(d[0],test);  
 d1 = *euclidian*(d[1],test);  
 }  
 **else if**(S.equals(**"taxicab"**)){  
 d0 = *taxicab*(d[0],test);  
 d1 = *taxicab*(d[1],test);  
 }  
 **else if**(S.equals(**"hamming"**)){  
 d0 = *hamming*(d[0],test);  
 d1 = *hamming*(d[1],test);  
 }  
 **if**(d0<d1)  
 **return** 0;  
 **else  
 return** 1;  
 }

}

### Classified Tweets

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

### Graphical Representation

Finally, the calculated result is displayed in graphical representation.

## 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 |

## 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. |

# Project Task and Time Schedule

## Work Division

Table 4: work Division

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

## Gantt Chart

Table 5: Gantt Chart



# Result and Discussion

## Final Outcomes

Finally, the proposed project is completed. We are 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:

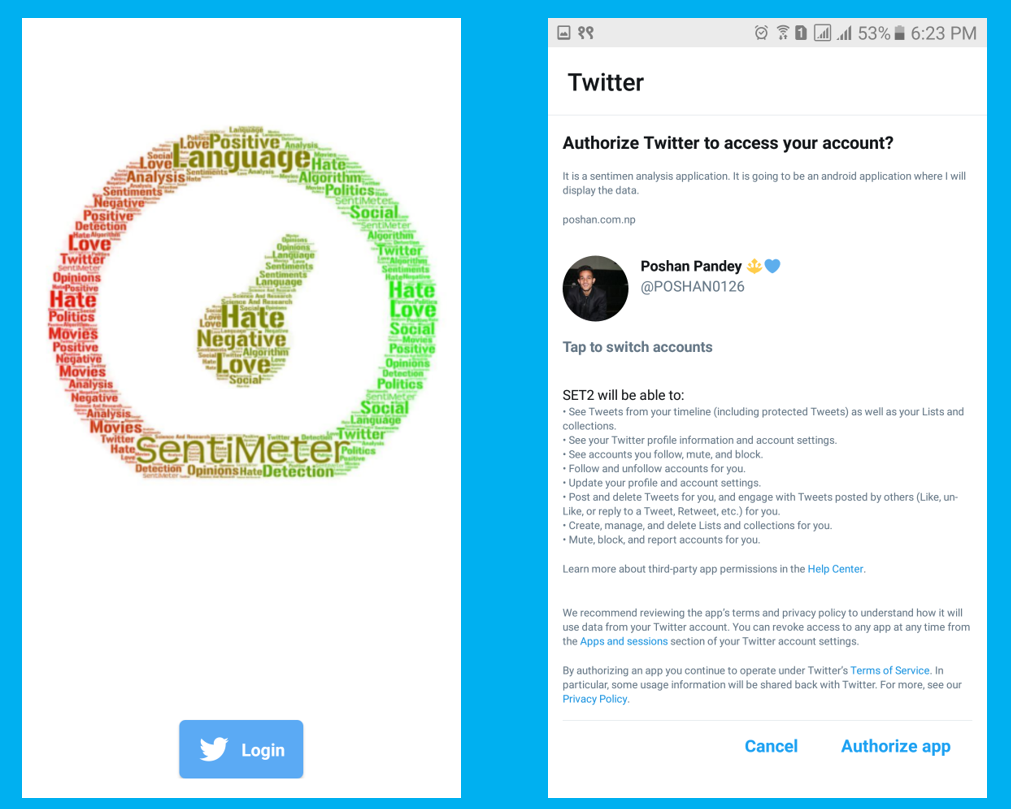


Figure 8: Logging in with Twitter Account

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:

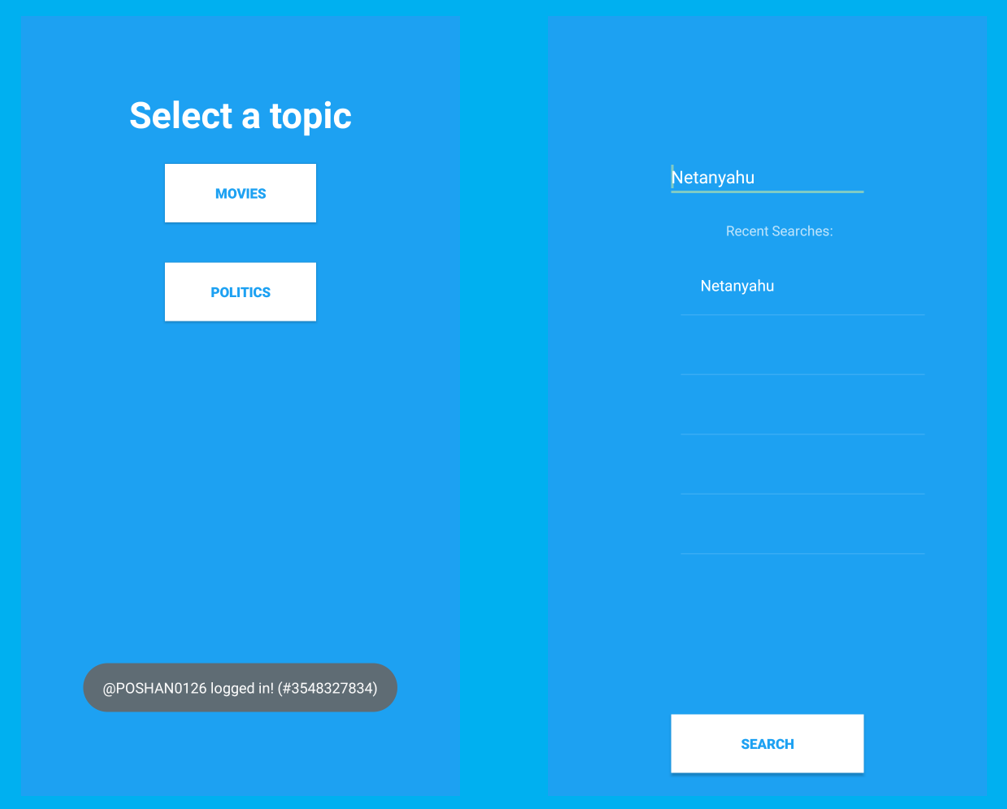


Figure 9: Selecting Topic and searching key

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:

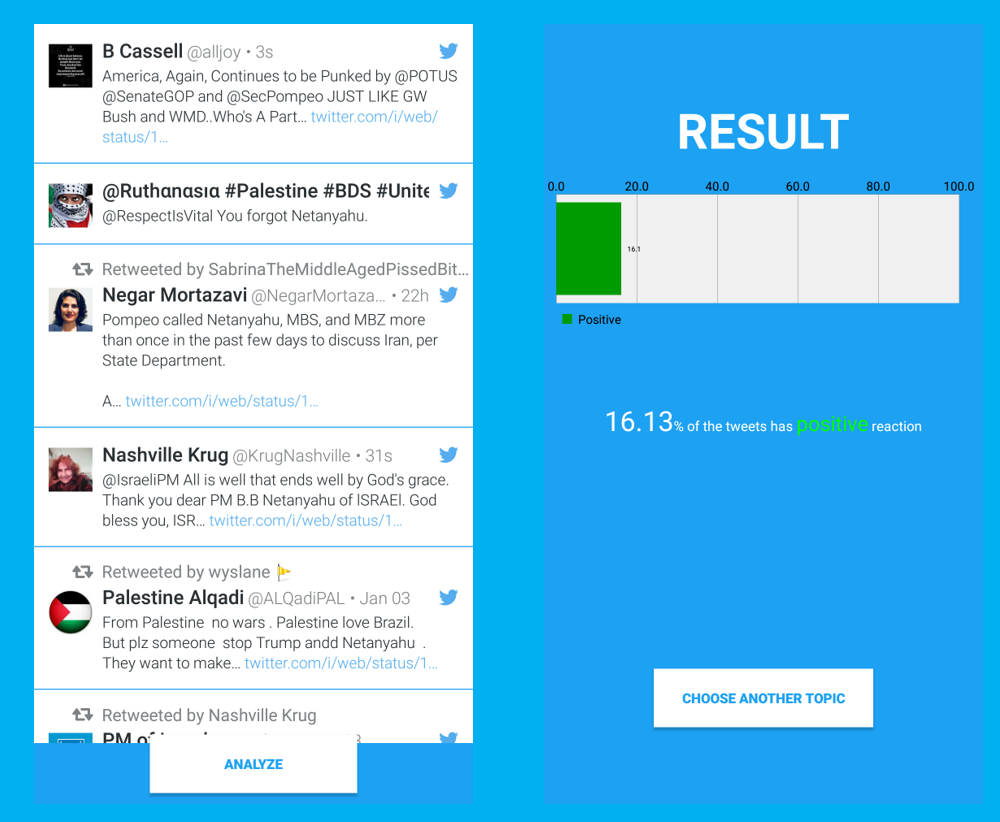


Figure 10: Displaying Tweets and then Result

## 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** | **63%** | **37%** |

The calculated accuracy for the used datasets is 59.8335%.

# Conclusion and Future Scope

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.

# 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]