# Emotion Detection in Tweet Replies

**A DISSERTATION**

Submitted in partial fulfillment of the
requirements for the award of the degree
of

**Master of Technology**

in

**INFORMATION SECURITY AND MANAGEMENT**

**BY**
**Manisha Keim**
00902052018

**Under the guidance of**
Mr. Rishabh Kaushal, Assistant Professor
Dr. Niyati Baliyan, Assistant Professor

**DEPARTMENT OF INFORMATION TECHNOLOGY**
**INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN**
**KASHMERE GATE, DELHI-110006**

2018-2020

Keywords: Social engineering, Emotion, Sentiment, Emotion detection, Sentiment detection, Emotion analysis, Sentiment analysis, Machine Learning

# Certificate

This is to certify that the work presented in this M.Tech Dissertation titled **"Emotion Detection in Tweet Replies"** is an authentic record of my own work under the supervision of **Mr. Rishabh Kaushal, Assistant Professor & Dr. Niyati Baliyan, Assistant Professor**, Department of Information Technology. It is submitted in partial fulfillment of the requirements for the degree of **Master of Technology** in **Information Security and Management**, Indira Gandhi Delhi Technical University for Women, Delhi.

Manisha keim

*Student signature*
**Manisha Keim**
**00902052018**

This is to certify that this work has been done under my supervision and guidance. It has not been submitted elsewhere either in part or full, for award of any other degree or diploma to the best of my knowledge and belief.

*Advisor signature*
**Mr. Rishabh Kaushal** **Date: 30 June 2020**
**Assistant Professor**

*Advisor signature*
**Dr. Niyati Baliyan** **Date: 30 June 2020**
**Assistant Professor**

# UNDERTAKING REGARDING ANTI-PLAGIARISM

I, hereby, declare that the material/content presented in the dissertation is free from plagiarism and is properly cited and written in my own words. In case, plagiarism is detected at any stage, I shall be solely responsible for it.

Manisha Keim

*Student signature*
**Manisha Keim**
**00902052018**

## Acknowledgments

I would like to thank my lecturers, **Mr.Rishabh Kaushal & Dr. Niyati Baliyan**, for their guidance through each stage of the process. This dissertation couldn't be completed without the effort and cooperation from your side.

I express my gratitude to my department for providing me the infrastructure to carry out the dissertation work and all those who were directly and indirectly related in enabling me to stay committed to this dissertation work.

**Abstract**

The online presence of users is generating an unprecedented abundance of information, which is inevitably public. The volume of information is increasing because users spend more time and engage more through tweets, replies, etc. Furthermore, the web, today, has gotten a key medium through which individuals express their feelings, sentiments and, perspectives. An individual piece of information is harmless, but once combined with others could lead to prophecies that the user has not imagined.

Commenting platforms, such as Twitter, have emerged as a major online communication platform with millions of users and tweets [5]. We focused on tweets about political leaders like Arvind Kejriwal & Amit Shah and then we selected twitter replies as data set. We have compared the accuracy of several machine learning algorithms, including Logistic Regression, Linear Support Vector Machine, and Naive Bayes for classifying twitter replies based on emotions. Our technique has an accuracy of over 64%, while demonstrating robustness across learning algorithms.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Internet has become an essential part of our life, varieties of amenities (email, shopping, and social networks) can be accessed anytime, anywhere from all kind of devices today. In the interim, users interact with the Internet, information is created and then spread and retained in the network. This information could be in the form of content provided voluntarily from users, such as Facebook posts, Tweets and comments toward Reddit posts [5].

Considering the time, a modern human spends on the Internet, one can expect that this online information, when combined, can reveal significant information. Also, commenting is an enormous activity for some users, who can spend numerous hours every day at it.

In this dissertation, we propose a way to automatically detect the emotion of tweet-replies in commenting platforms. Our input dataset is Tweet-Replies. We have compared the accuracy of several algorithms, including Logistic Regression, Naive Bayes, SVC, Random Forest. Our technique has an accuracy of 64% with Linear SVC, Logistic Regression.

## 1.1 Basic Terminology

A **user** is an account on commenting platform, which enables his/her to leave replies to tweets or create a new tweet expressing his/her views on any topic.

Commenting on online articles/posts is a new form of social interaction. A **commenting platform** is a platform that enables users to write their views in the form of comments. For example: Twitter, Reddit, Facebook, etc. [5]

**Tweets** are short instant messages with 140 characters yet are influential source of expressing emotional state and feelings. [6]

Tweets may also contain the emotional states of users (such as happiness, sadness, anxiety, etc).

Figure 1.1a shows example of a **Sad** political tweet-reply, Figure 1.1b shows example of a **Angry** political tweet-reply & Figure 1.1c shows example of a **Disgust** political tweet-reply.


(a) Sad


(b) Angry


(c) Disgust

Figure 1.1: Examples of Emotion Tweet-Replies

## 1.2 Problem Statement

The research question in our work is: can we automatically know the sentiment/emotion of tweet-replies in commenting platforms. In our work, we build a model to automatically classify emotions based on tweet replies on a particular tweet.

The input to our model is the tweet-reply information of the users which contains tweet-reply ID, date & time it was posted, who posted the reply and the tweet reply text. The ouptut is to detect emotional behavior. Detecting emotional behavior is a critical building block for ensuring that these platforms serve their primary purpose, which is the honest and safe exchange of thoughts among users. [5]

Ekman et al. [1] proposed a model of six fundamental feelings after analyzing facial expressions observed in distinct cultures: happiness, sadness, anger, fear, disgust and surprise [1]. Sentiment is classified either as positive, negative or neutral. Our work provides an approach for automatically classifying Twitter replies into distinct emotional categories(anger, sadness, disgust and happiness).

## 1.3 Motivation

The proposed method can be used to know the mood of a person which may support researchers or therapists to get some insights.

It can also be used by healthcare counsellors to monitor patient's emotional state whether a person is going through some stress.

# Chapter 2

# Literature Survey

This section describes the prior works on classifying emotion from text.

Ching Li et al. [5] focuses on commenting platforms like Disqus for detection of misbehaviour. Their work has two key novelties: (a) propose two classifications methods, with one following a two-stage approach, which first maps observable features to behaviors and then maps these behaviors to user roles, and (b) use a comprehensive set of 73 features.

Hasan et al. [6] proposes a algorithm to automatic labelling of twitter messages to the emotions of users and compares the accuracy of machine learning algorithms including SVM, KNN, Decision Tree, and Naive Bayes. They have divided the dataset to four classes: Happy-Active, Happy-Inactive, Unhappy-Active, and Unhappy-Inactive. They have also considered emoticon features, punctuation features & negation features. Among various algorithms, KNN achieved the accuracy 0f 90%.

Badugu et al. [7] focus on rule-based classifier which identifies the feeling or state of mind of the tweet and classifies the twitter message under appropriate emotional category. The system consists of four classes of emotions namely (c1) Happy-Active, (c2) Happy-Inactive, (c3) Unhappy-Active, and (c4) Unhappy-Inactive, it gives the accuracy over 85.1%.

Ranganathan et al. [8] develops a corpus based on the National Research Council and classifies emotions from tweets based on decision tree classifier, decision forest, and rule-based classifier for automatic classification of emotion based on the labeled dataset.

Gaydhani et al. [10] propose an algorithm to automatically classify tweets into three classes: hateful, offensive and clean. They considered n-grams as features and passing their term frequency-inverse document frequency (TFIDF) values to various machine learning models.

Table 2.1 summarizes some basic emotion models used in the literature and they express mostly

Table 2.1: Emotion Models.

| **Emotions** | **References** |
|---|---|
| Anger, Disgust, Fear, Happiness, Sadness, and Surprise | [6] |
| Nervous, Anxious, Tension, Afraid, and Fear | [7] |
| Anger, Fear, Anticipation, Trust, Surprise, Sadness, Joy, and Disgust | [8] |
| Surprise, Disgust, Happiness, Fear, Sadness, and Anger | [9] |

Table 2.2: Feature Analysis.

| **Feature** | **References** |
|---|---|
| Offensive words/ Hostile language | [2], [3] |
| Number of statuses | [2], [5] |
| Number of followers | [2], [3], [5] |
| Number of favorites | [2], [3] |
| Number of URLs/tweet | [2], [5] |
| Number of hastags/tweet | [2], [3] |
| Suspended/deleted accounts | [2], [3] |
| Denouncing speech | [3] |
| Number of posts since account creation | [3], [5] |
| Account creation date | [2], [3] |

Table 2.3: Evaluation Metrics.

| **Algorithm** | **References** |
|---|---|
| Logistic Regression | [3], [10] |
| Naive Bayes | [4], [6] |
| Support Vector Machines | [3], [10] |
| Random Forest (RF) | [3], [4], [5] |
| XGBoost(XGB) | [3] |
| CatBoost(CB) | [3] |
| Extra-Tree (ET) | [3] |
| Decision Tree | [4], [8] |
| TF-IDF | [10] |

Table 2.4: Datasets.

| **Datasize** | **References** |
|---|---|
| 100386 users | [2] |
| 558 hate tweets received a total of 1711 replies | [3] |
| Twitter Streaming API | |
| (a) 1M random tweets | |
| (b) hate-related 650K tweets based on 309 hashtags | [4] |
| 7M comments | [5] |
| 16,5300 tweets | [6] |
| 10,48576 tweets | [7] |

all human emotions. This table maps emotions to which author have used. The most common features among all of them are anger, fear, happiness, etc. Some authors have used hashtags like #enjoy, #happytweet to label Happy-Active class, ,#peaceful, #satisfied to label Happy-Inactive class, #anxious, #distress to label Unhappy-Active class and #sad, #nothappy to label Unhappy-Inactive class [6].

Table 2.2 lists the features used in the literature. Many of the features include followers count, favorites count, URLs count, Hashtags count, etc. For example, the dataset of author in [7] includes polarity, id, date, query, username and tweet.

Table 2.3 lists the various algorithms used in models to get the desired accuracy. Various authors have trained their models through Decision Tree Classifier and author in [8] have implemented decision table majority (DTM) rule-based classification method which includes schema and body where Schema is set of features and Body is set of labels. DTM classifier searches for exact match if any unlabeled instance is present in dataset.

Table 2.4 shows the dataset distributions taken by authors. The distribution gives an idea about how much data should be divided into training and test data.

# Chapter 3

# Data Collection and Annotation

## 3.1 Collecting Tweet Replies

We searched verified profiles for the target political leaders(Arvind Kejriwal and Amit Shah) and manually collected 15 unique tweets for each of the leader.

After this, we collected tweet replies from Twitter API, web-scraping using Selenium, BeautifulSoup in two-part process:

- First, we collect tweet-reply ID from tweet.
- Then, we pass the tweet-reply ID to Twitter API to collect the tweet-reply text(known as 'tweetText' in our dataset), username, date & time of tweet reply and store it in a CSV file.

We consider tweets about 2 political leaders:

- Arvind Kejriwal
- Amit Shah

Dataset Specification:

- No. of tweets: 30
- No. of tweet-replies: 3200
- Date of data collection: September 2019 to 07th June 2020.
- Frequency of data update: Weekly.

## 3.2 Manual Annotation of Tweet Replies & Data Visualization

We aim to build two types of classifiers:

1. Binary classifier to classify the political tweet replies as "happy" and "not happy".
2. Multi-class classifier to classify tweet replies into four finer-grained classes of emotion - anger, sadness, disgust and happiness.

By emotion of a political tweet reply we mean whether it can be said, by just reading the tweet, that the user is emotionally happy or not.

To build these classifiers, we need to have a training dataset that is already labelled, and for that we need to manually annotate this data. Though, we are currently working on the binary classifier, the annotation process should be conducted for both the classifiers together to maintain coherency and avoid bias.

**Process:**
We selected a random set of political tweets-replies that are used for manual annotation, training and testing of the classification models. To have an uniform distribution of tweet-reply of each political leader in the sample, we randomly select 750 political tweet-replies for each political leader.

The annotator is given access to only the "tweetText" of the tweet and no other information about the user like his name, date & time of tweet reply, etc. This is done to ensure that the annotation is done by only analyzing the text of the tweet reply.

We did the following labelling after analyzing the tweets:

1. Label 1: To classify a tweet reply as happy or unhappy.
   - 0: happy
   - 1: unhappy
2. Label 2: If the value of Label 1 is 'unhappy', then we define three emotional roles:
   - 1: Sad
   - 2: Angry
   - 3: Disgust

Table 3.1: Some examples where Label 1 is marked 1.

| **Reply** | **Label** |
|---|---|
| Wastage of funds. Build statues and Temples instead. | Unhappy |
| U people r doing less n speak much louder n other hand @narendramodi @TajinderBagga he work hard to unite India n u people donate like its really scary situation again let u ask to think about ur strategy n make sure India first n All ur other agenda comes second. | Unhappy |

Table 3.2: Some examples where Label 1 is marked 0.

| **Reply** | **Label** |
|---|---|
| Sir Indeed.. Ladies in Delhi are really enjoying and overwhelmed. Wish each state and central govt. in our country someday throws away their political ideologies and EGOs for the sake of #INDIA's future and implement similar educational structure... | Happy |
| We didn't get a #smarteducationalsystem.. Hope the next gen gets it... #Indiawillshineagain... | Happy |

Table 3.3: Some Examples of Label 2 annotation.

| **Reply** | **Label** |
|---|---|
| Wastage of funds. Build statues and Temples instead. | Sad |
| I am a modi supporter but really like your work. Just keep distance from congress. I will support you. | Sad |
| What for. They did nothing. They are making fool . They are liars. No college was opened by them. They are outsiders. | Angry |
| @ArvindKejriwal In unauthorized colonies largely illegal constructions carried out by property mafia. Govt of Delhi do not want to touch this issue as govt land is sold by builder mafia and big vote bank politics loss. | Disgust |
| @ArvindKejriwal Bribing voters will not help | Disgust |

Figure 3.1 shows the distribution of tweet-replies on the basis of Sentiment - negative, positive and neutral. This takes into consideration the 'tweetText' of dataset.

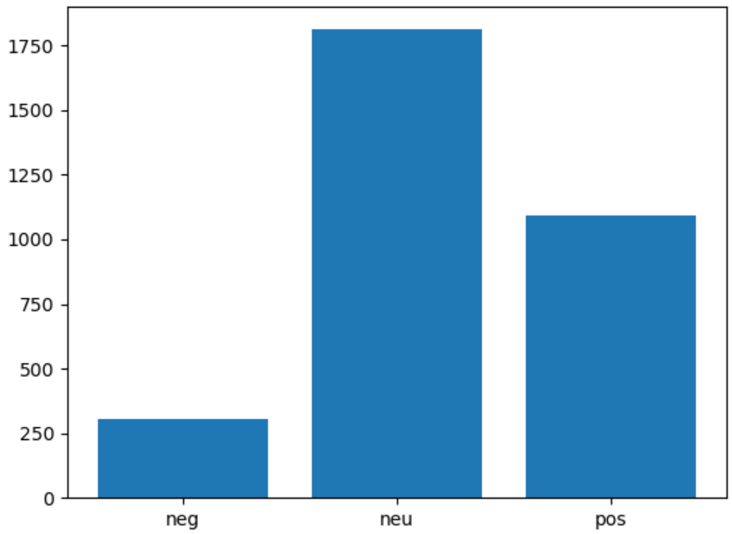


Figure 3.1: Sentiment Analysis on dataset

Figure 3.2 displays the word cloud of negative tweet-replies in the dataset.

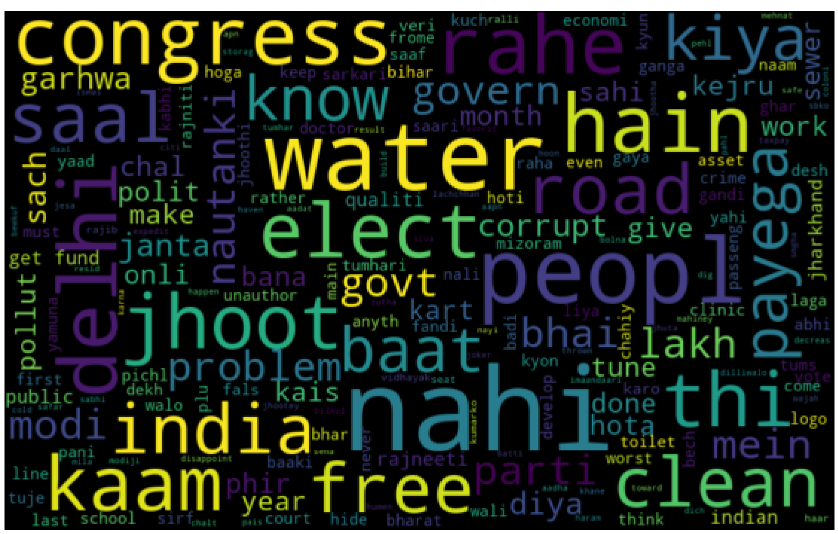


Figure 3.2: Wordcloud from negative tweet-replies

The frequent hashtags extracted from datset are:

- #NoRoadNoVote
- #maharashtracrisis
- #amprapalihome
- #dillikapaanizehrila

- #terroristpragyathakur

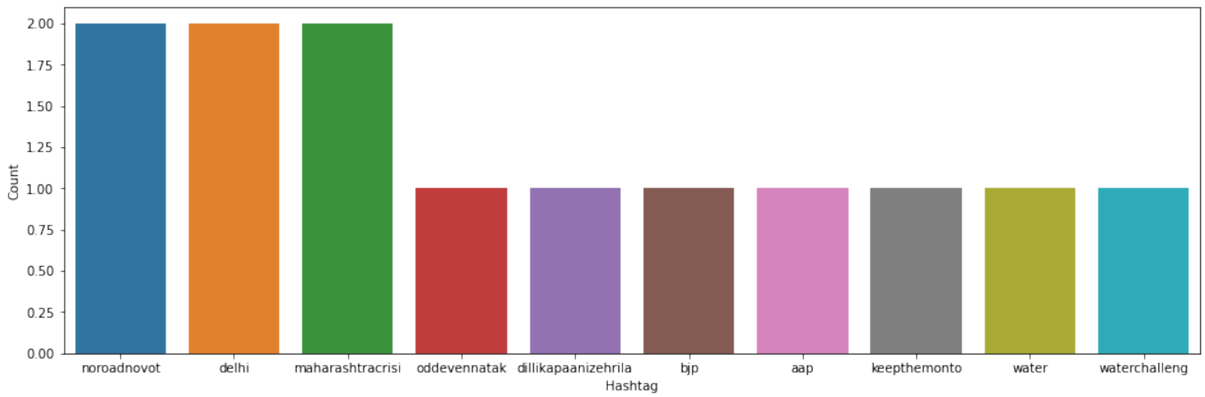


Figure 3.3: Hashtags extracted from tweet-replies

# Chapter 4

# Proposed Methodology

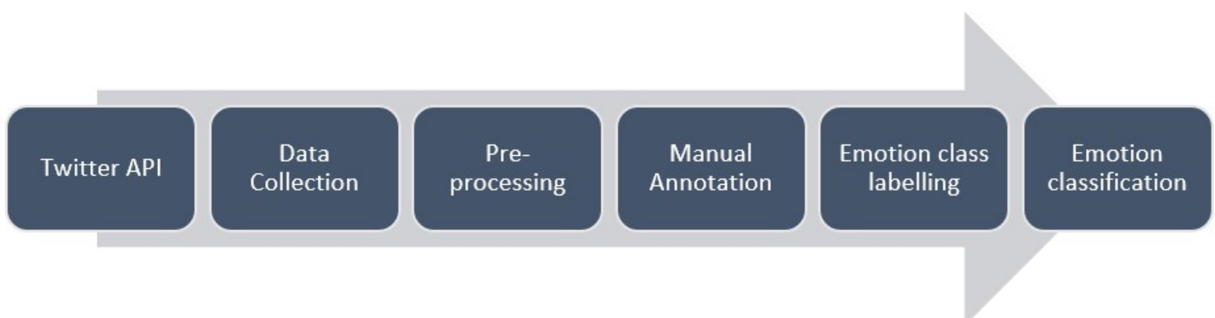


Figure 4.1: Methodology

## 4.1 Basic Pre-processing

The tweet replies obtained from API are pre-processed to transform them into semantic vectors. The pre-processing process is divided into intermediate steps:

1. Removing punctuation, URLs and user mentions.
2. Conversion of all tweet-replies into lower case.
3. Removal of stop words. For instance words like 'a', 'an', 'is', 'the' etc. These words don't give us any information about the content of the text. Thus it should not matter if we remove these words for the text.
4. Translation of Hindi & Hinglish words into corresponding English words through **google-trans**.
5. Transformation of pre-processed tweets into a word vector representation.

6. The final step is the formation of a word vector sequences that can be provided as input to classifier.

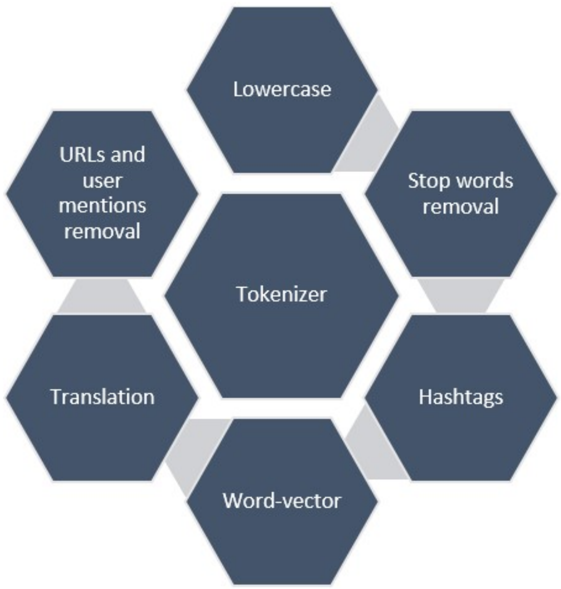


Figure 4.2: Pre-processing

## 4.2 Word Vector Representations

### 4.2.1 Bag-of-Words(BoW)

BoW is an unigram model where we consider individual words into account and give each word a unique number.

CountVectorizer is a part of sklearn package, which converts collection of documents to tokens.

For example:

We provide input the following sentence and in return we get the result assigning a unique number to every word in the sentence.

$$input = [''The\ quick\ brown\ fox\ jumped\ over\ the\ lazy\ dog.'']$$

$$output = 'dog' : 1, 'fox' : 2, 'over' : 5, 'brown' : 0, 'quick' : 6, 'the' : 7, 'lazy' : 4, 'jumped' : 3$$

It also displays the count of every word, which is a sparse representation:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

,

### 4.2.2 TF-IDF

TF-IDF stands for "Term Frequency — Inverse Document Frequency".

- **Term Frequency:** This summarizes how often a given word shows up within a document.
- **Inverse Document Frequency:** This down-scales words that are common across documents.

This is another strategy which depends on the frequency method but it is distinctive to the bag-of-words approach in the sense that it takes into account, not just the occurrence of a word in a single tweet-reply but in the whole corpus.

TF-IDF works by penalizing the common words by assigning them lower weights while giving significance to words which are uncommon in the whole corpus yet show up in great numbers in few documents.

$$input = ["The\ quick\ brown\ fox\ jumped\ over\ the\ lazy\ dog.", "The\ dog.", "The\ fox"]$$

$$output = 'fox' : 2, 'lazy' : 4, 'dog' : 1, 'quick' : 6, 'the' : 7, 'over' : 5, 'brown' : 0, 'jumped' : 3$$

Following are the inverse document frequencies are calculated for each word in the vocabulary, assigning the lowest score of 1.0 to the most frequently observed word: "the" at index 7:

$$\begin{bmatrix} 1.69314718 & 1.28768207 & 1.28768207 & 1.69314718 & 1.69314718 & 1.69314718 & 1.69314718 & 1.0 \end{bmatrix}$$

## 4.3 Emotion Classification

We divided the annotated dataset into two folds, one for training data and other for testing data. Then training data is passed to classifiers using various classification algorithms.

We calculated the accuracy for each classification algorithm. We have also calculated precision, recall, and F1-score mentioned below in table.

### 4.3.1 Binary Emotion Classifier

We provide the input as the annotated dataset which is fed to train the classifier. We have used Logistic Regression & Naive-Bayes Classifier.

Our first classifier is a binary classifier that classifies whether the political tweet reply depicts emotion as "happy(0)" or "not-happy(1)".

Figure 4.3 shows the distribution of tweet replies as happy & not-happy.

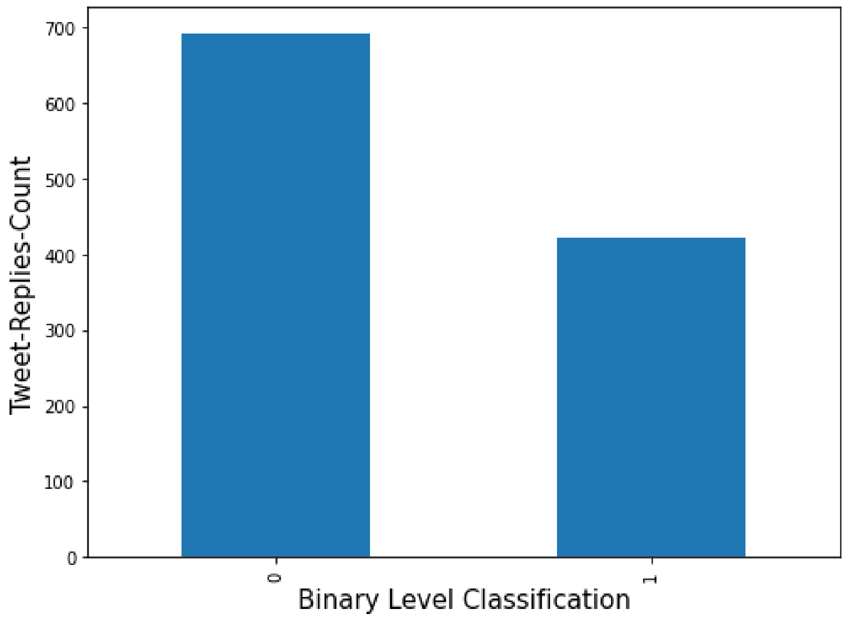


Figure 4.3: Distribution of tweet replies

#### 4.3.1.1 Binary Classifier using Logistic Regression

We apply Logistic Regression classifier for supervised classification.

**Logistic Regression using Bag-of-Words features**
Giving an accuracy of 75% and the other scores are mentioned in Table 4.1

**Logistic Regression using TF-IDF features**
Giving an accuracy of 71% and the other scores are mentioned in Table 4.1

After performing our model on Logistic Regression we have find out that it was finding only features pertinent to emotion but it did not perform well when finding the particular emotion(eg. Sad, Angry or Disgust) from tweet-replies.

#### 4.3.1.2 Binary Classifier using Naive Bayes

We apply Naive Bayes classifier for supervised classification.
Giving an accuracy of 79% and the other scores are mentioned in Table 4.1

Out of all 3 classifiers, Naive Bayes performs better on binary classification.

### 4.3.2 Multi-Class Classifier

To train the the Multi-Class classifier, we use TF-IDF weighted vectors and then apply train on 'tweetText' and predict the emotion.

After the data transformation, we have all the features and labels which is fed to train the classifier. We have used Naive-Bayes Classifier which is most suitable for multinomial variant.

Table 4.1: Classifiers & their scores.

| **Classifiers** | **Label** | **precision** | **recall** | **f1-score** |
|---|---|---|---|---|
| LR using Bag-of-Words | Happy(0) | 0.84 | 0.74 | 0.79 |
| | Unhappy(1) | 0.63 | 0.76 | 0.69 |
| LR using TF-IDF | Happy(0) | 0.92 | 0.54 | 0.68 |
| | Unhappy(1) | 0.61 | 0.94 | 0.74 |
| Naive Bayes | Happy(0) | 0.78 | 0.90 | 0.84 |
| | Unhappy(1) | 0.80 | 0.62 | 0.70 |

Figure 4.4 shows the distribution of tweet replies as 0(Happy), 1(Sad), 2(Angry) & 3(Disgust).

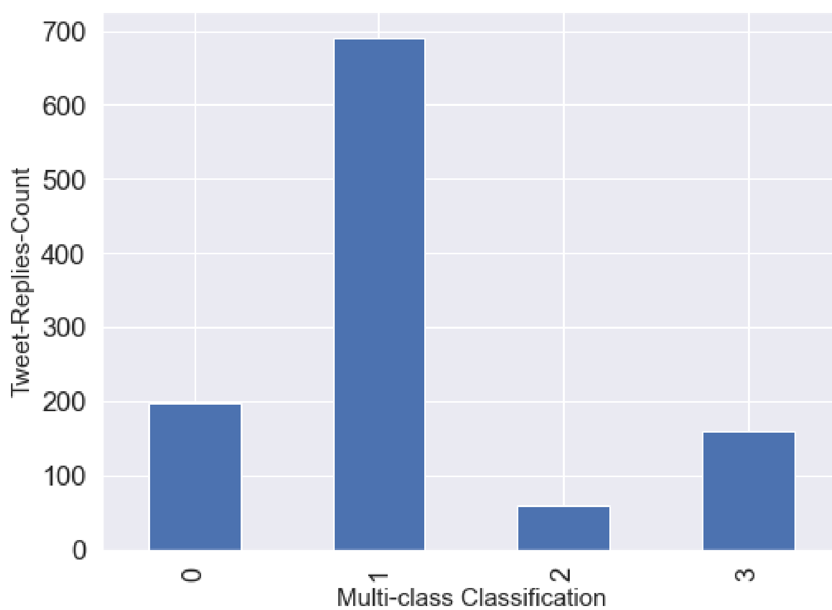


Figure 4.4: Distribution of tweet replies

Following are the features that are most correlated with each of the emotions, it's extracted after training 'tweetText' from our corpus(we have used **chi2** [11] to extract these features):

- Angry
  - Most correlated unigrams:
    - 'jhoot'
    - 'elections'
  - Most correlated bigrams:
    - 'ground reality'
    - 'sewer line'

- Disgust
  - Most correlated unigrams:
    * 'corruption'
    * 'muft'
  - Most correlated bigrams:
    * 'ground reality'
    * 'clean water'
- Sad
  - Most correlated unigrams:
    * 'respect'
    * 'help'
  - Most correlated bigrams:
    * 'vote nahi'
    * 'kisi vote'

In Figure 4.5, we can see the discrepancies between the predicted and actual emotions/labels through the confusion matrix.

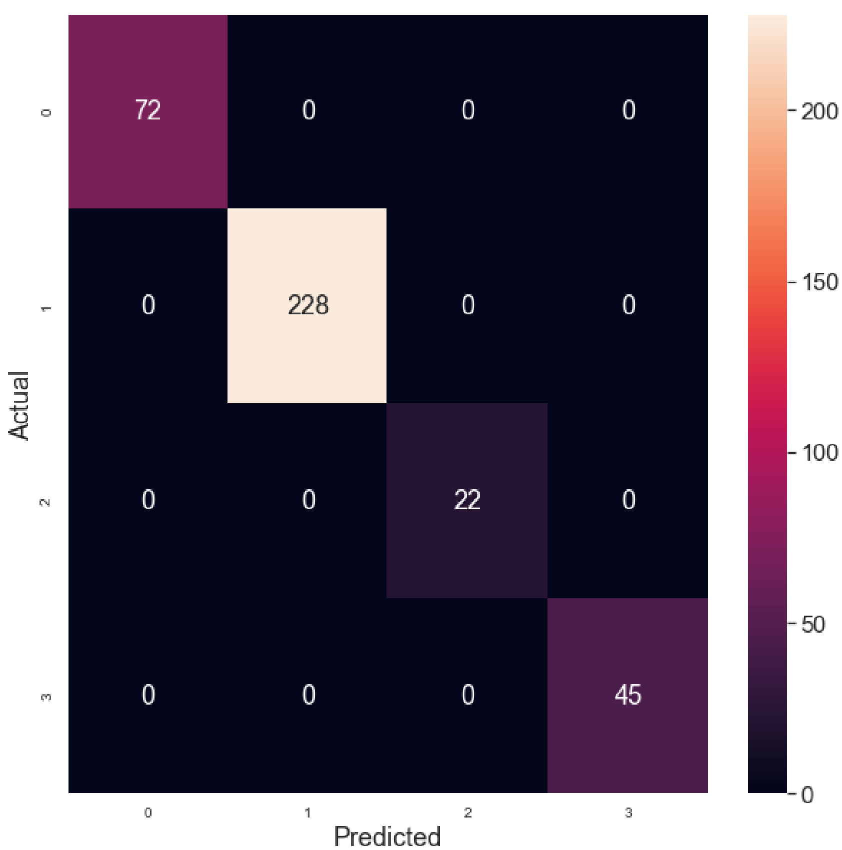


Figure 4.5: Confusion Matrix'

After testing our model with Naive Bayes, we can test it with other models : Logistic Regression (Multinomial) Naive Bayes Linear Support Vector Machine Random Forest

Table 4.2: Various models and their accuracy.

| **Model Name** | **Accuracy** |
|---|---|
| LinearSVC | 0.64 |
| LogisticRegression | 0.64 |
| MultinomialNB | 0.63 |
| RandomForestClassifier | 0.62 |

Here, Linear Support Vector Machine & Logistic Regression performs better than the other two classifiers.

# Chapter 5

# Conclusion

In our work, we have used Naive Bayes, Logistic Regression, Random Forest Classifier, Linear Support Vector Machine. Bayesian classifiers are probabilistic classifiers based on features in different classes. For our multi-class classification problem, we have chosen Multinomial Naive Bayes, it is part of sklearn package. It is suitable for classification with discrete features.

**Results:**
We have collected 3200 tweet-replies into consideration. The number of tweet-replies got decreased due to pre-processing. We have assumed that our training labels are mutually exclusive i.e one tweet-reply corresponds to only one emotion. After pre-processing, we spit our corpus into training and testing set. We have applied various Machine Learning algorithms to our corpus, we calculated accuracy, precision, recall, and F1-score. Table 4.1 shows the results of binary classification using Logistic Regression - Bag of Words features, Table 4.1 shows the results of binary classification using Logistic Regression - TF-IDF features, Table 4.1 shows the results of binary classification using Naive Bayes. The results from these classifiers helps us understand how we can automatically know the sentiment/emotion(happy or unhappy) of tweet-replies.

Another classification is done by multi-class classification problem to know the fine-grained class of emotion like sad, angry or disgust. The accuracy of classifiers applied on this dataset is shown in Table 4.2.

Detecting emotion in Hinglish tweet-replies is one of the challenging problem to solve, and we faced various challenges during data collection and classification like:

- Informal language users use in tweet-reply.
- Tweet-reply which don't have any emotion.
- Language ambiguity.

# Chapter 6

# Future Work

Our work can be improved for better results and accuracy as there are some limitations which can be improved in future, for example:

1. We may increase the size of our Hinglish dataset.
2. User features like hashtags, emoticons, #followers, tweet reply length, #user-mentions, username, date & time of tweet reply, can be considered to extract some meaning out of them.
3. Users usually post images, videos & GIFs as part of tweet-reply which can also be a part of corpus.
4. Language ambiguity.
5. In the future, we can use Transfer Learning/Deep Learning approach to automatically detect emotion.

# Bibliography


[1] Ekman, P., Friesen, W. V., Ellsworth, P. (1972). Emotion in the Human Face. 1 ed. Pergamon.

[2] Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A. and Meira Jr, W (2017). "Like Sheep Among Wolves": Characterizing Hateful Users on Twitter.

[3] B Mathew, N Kumar, P Goyal, A Mukherjee. (2018). Analyzing the hate and counter speech accounts on Twitter.

[4] D Chatzakou, N Kourtellis, J Blackburn. (2017). Detecting aggressors and bullies on Twitter.

[5] T. C. Li, J. Gharibshah, E. E. Papalexakis, and M. Faloutsos. (2018). TrollSpot: Detecting misbehavior in commenting platforms. *ASONAM '17: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017* Pg. 171-175.

[6] Hasan, M., Rundensteiner, E., and Agu, E. (2014). EMOTEX: Detecting Emotions in Twitter Messages. *Cybersecurity Conference, Stanford University.*

[7] Badugu, S. and Suhasini, M. (2017). Emotion Detection on Twitter Data using Knowledge Base Approach. *International Journal of Computer Applications (0975 – 8887)* Volume 162 – No 10, March 2017.

[8] Ranganathan, J., Hedge, N., Irudayaraj A.S. and Tzacheva A.A. (2018). Automatic Detection of Emotions in Twitter Data - A Scalable Decision Tree Classification Method. *ACM RevOpID '2018, Baltimore, Maryland, USA.*

[9] Gaind, B. and Syal, V. and Padgalwar S. (2019). Emotion Detection and Analysis on Social Media. *Global Journal of Engineering Science and Researches* (ISSN 2348 - 8034, Pg. 78-89).

[10] Gaydhani, A., Doma, V., Kendre, S., and Bhagwat, L. (2018). Emotion Detection and Analysis on Social Media. *Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach.*

[11] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.