# HATE SPEECH DETECTION FOR CYBERBULLYING USING NAÏVE BAYES AND SVM

**A PROJECT REPORT**

*Submitted by*

**SNIGDHA BOSE [Reg No: RA1611020010065]**
**SWARANSHI SAXENA [Reg No: RA1611020010165]**

*Under the Guidance of*

**Dr. FEMILDA JOSEPHIN**
(Associate Professor, Department of Software Engineering)

*In partial fulfilment of the Requirements for the Degree Of*

**BACHELOR OF TECHNOLOGY**


**DEPARTMENT OF SOFTWARE ENGINEERING**
**FACULTY OF ENGINEERING AND TECHNOLOGY**
**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**
**KATTANKULATHUR – 603203**

**MAY 2020**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
# KATTANKULATHUR-603203

## BONAFIDE CERTIFICATE

Certified that this project report titled **"HATE SPEECH DETECTION FOR CYBERBULLYING USING NAÏVE BAYES AND SVM"** is the bonafide work of **"SNIGDHA BOSE [Reg No: RA1611020010065], SWARANSHI SAXENA [Reg No: RA1611020010165],** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

**GUIDE**

Dr.J.S. Femilda Josephin

Associate Professor
Dept. of Software Engineering

**HEAD OF THE DEPARTMENT**

Dr.C.LAXMI

Dept. of Software Engineering

Signature of Internal Examiner

Signature of External Examiner

# ABSTRACT

With the increase usage, hate speech is nowadays of keen interest with respect to social media. The anonymity which is provided by the Internet has made it easy for people to comment on anyone in an hateful way to any extent as no one is answerable for their actions. And with the rise of cyber-bullying ,finding ways to detect hate speech automatically is becoming an urgent need.
On Twitter, neagtive tweets are the ones that contain hateful and rude comments targeting selected users like (a person in politics, LGBT,a person with fame, a product) or particular groups (a country, a religion, gender, an organization, etc.). Detecting these kind of hateful comments is important for analysing overall sentiment of an individual or group of people towards another set of groups, and for discouraging associated wrong activities. The manual way of filtering out hateful tweets is not scalable and flexible, motivating researchers to identify automated ways. In this project, our main aim is to classify a tweet as positive or negative text. The task is a bit difficult because of the inherent complexities like different forms of hatred, different kinds of targets, different methods of telling the same meaning. Most of the previous work is about using visual learning methods followed by a linear classifier. However, nowadays deep learning methods and SVM have shown improvements in accuracy. Hence, In order to reduce this problem, we propose a system that can detect cybercrimes from social media automatically using Naive Bayes Classifier and SVM.



## KEYWORDS

Cyberbullying ,Hate Speech Detection, Naïve Bayes, Sentiment Analysis

# ACKNOWLEDGEMENT

We would like to take this opportunity to thank the **Director of E&T**, **Dr. Muthamizehelvan** and **Head of Department of Software Engineering, Dr. C. Lakshmi,** for making this major project course a part of the curriculum and possible for the students.

We would also like to extend our gratitude towards **Mrs. M. Uma, Asst. Professor (S.G)** and **Mrs. Jeyasudha** ,**Asst. Professor(O.G), Mrs. C.G Anupama, Assistant Professor (Sr. G) ,**the Project Co-ordinators for giving us the opportunity to take part in this major project course. We also convey our immense gratitude for **Dr. Femilda Josephin J.S, Associate Professor, Department of Software Engineering** who gave us constant inspiration and suggestions throughout the project work. Under her guidance we were able to complete the project in time and come up with some astonishing research findings. We are highly obliged to her for her help and guidance. She has not only helped us in our project but has also been a great mentor to us.

Lastly, we would like to thank the entire Software Engineering Department who has helped in the completion of our project.

**SNIGDHA BOSE** **RA1611020010065**
**SWARANSHI SAXENA** **RA1611020010165**

# TABLE OF CONTENTS

# LIST OF IMAGES

# LIST OF TABLES



# LIST OF ABBREVIATIONS

| | |
|---|---|
| **ANN** | Artificial Neural Network |
| **AI** | Artificial Intelligence |
| **LGBT** | Lesbian, gay, bisexual and transgender community |
| **SVM** | Support Vector Machine |
| **MLA** | Modern Language Association |
| **NLP** | Natural language Processor |
| **GIF** | Graphics interchange format |
| **SVM** | Support Vector Machine |

# CHAPTER 1

# INTRODUCTION

Cyberbullying activities take place on various online platforms, where users share their thoughts, interests and issues. Some of the users engage in activities such as bullying, harassment, threat and intimidation making others feel bad about themselves, lowering their self-esteem and discouraging them from interacting or expressing themselves. Anonymity has become a major reason which makes this very easy since these bullies never have to face negative consequences. The problem has been growing since the along with the increase in usage of social media. An eye-opening episode occurred when the Formspring; a social network was closed, due to the number of suicides connected with it, due to many messages containing hateful and abusive language. Social media like Twitter, Facebook, Instagram and YouTube are being criticized for not putting much effort in preventing such crimes. They are under pressure to take major actions in order to prevent cyber bullying. The German government has even threatened to fine the such sites, up to 50 million euros if they continue to neglect hateful postings and not take actions within a week. The major motivation of this project is to:

- Increases Security and Control.
- Reduces IT Administration Costs.
- Enhance Privacy.

Cyber bullying is a major problem, which has its roots in growing use of technology around the world. But, technology can as a matter of fact, also help in reducing and also preventing such crimes. Various algorithms can be used for achieving this goal. The most popular algorithms for this purpose are Machine Learning algorithms, Deep Neural Networks in particular can be used nowadays for finding a solution to this problem. Machine learning means teaching a computer to learn by itself without being explicitly programmed. By the use of machine learning patterns in language can be detected which are used by bullies and also to develop some rules in order to automatically detect texts which may have cyber bullying content. In a study that used machine learning to detect cyberbullying, some posts were analysed from FormSpring.me, a social networking site, which contains a very high rate of hate speech content. The texts were labelled using Turk, an AWS (Amazon Web Service). A set of bad words was downloaded from www.noswearing.com which was used to assign severity levels to the texts obtained from social media. The bad words were normalized to train the model. Many machine learning algorithms were applied to build the model, like, decision tree, support vector machine, Naive Bayes algorithm.

The rest of the paper is organized as follows: section 2 includes a theoretical background, and the literary survey which described the problem and helped in requirement gathering and analysis, section 3 will include more of a visual approach to define how the raw data is refined into useful information for hate speech detection. Following this, analysis and recommendations will be presented in section 4. Finally, the paper is concluded by highlighting the probable future research directions.

# CHAPTER 2

# PROJECT OVERVIEW

## 2.1 LITERATURE SURVEY:

- There has been an evident number of life-threatening experiences due to cyberbullying especially among youths throughout the world.
- The recent studies show that this problem is more exaggerated in the USA where about 43% of teens are the targets of cyber bullying .

It is consequently apparent that the readiness of tools that can distinguish practices categorized as cyberbullying can be extremely valuable. Here are a few research papers based on various Hate Speech Detection Techniques:

### 1.” Automatic detection of cyberbullying in social media text” by Cynthia Van Hee and Gilles Jacobs, October 8, 2018.

Objective:

The focus of this paper is on automatic cyberbullying detection in internet apps and texts by taking sample posts of cyber bullying. Binary method is used to find even little cyber bullying in using SVM.

Methodology:

For pre-processing, tokenisation is used, PoS-tagging and lemmatisation have been done using LeTs Preprocess Toolkit. After pre-processing, the extracted features were described using word n-gram, bag-of-words and also Character n-gram bag-of-words.

### 2. Detecting Hate Speech in Social Media” by Shervin Malmasi and Marcos Zampieri, April 2018.

Objective:

This paper aims to discover lexical patterns for detecting cyber bullying by applying supervised Machine Learning models for classification employing a dataset annotated for this purpose.

Methodology:

For features, their model employs character n-grams, word skip-grams and word n-grams. Results were obtained with an accuracy of 78% in identifying posts containing hate speech.

### 3.” Cyberbullying on social media platforms among university students in the United Arab Emirates” by Ghada M. Abaido,14 Sep 2019.

Objective:

The focus of this paper is to identify the quality of spreading of online bullying in UAE among university students and towards filing report against cyberbullying instead of remaining silent.

Methodology:

This research identified data using quantitative collection methodologies to understand the underlying pattern of cyber bullying activities. Questionnaires were designed and distributed among university students to get a better understanding of the situation. These questionnaires included multiple choice questions with scaled responses and also a few open‑ended questions for better understanding. A test was also held to validate the reliability of the information obtained from the survey. In addition to this , a theoretical framework and the study of the existing models were used as a support for this study.

### 4.” Detecting Hate Speech in Social Media In Proceedings of the Recent Advances in NLP” Conference by Shervin Malmasi and Marcos Zampieri, September 2017.

Objective:

The main aim is to work out related baselines for the task of detecting hate speech by using supervised classification methods.

Methodology:

For features, the model employs character n-grams, word skip-grams and word n-grams. The result obtained is an accuracy of 78% in identifying posts classifying texts as hate speech or not. The results show that the major challenge is differentiating among profanity and hate speech.

### 5.” Cyber bullying Detection:- A Step Toward a Safer Internet Yard” by Maral Dadvar and Franciska de Jong,April 2016.

Objective:

Studies show that cyberbullying detection has primarily focused on the content of the text while largely ignoring the users involved in the activity. This study demonstrates inclusion of the users’ information as well as the harassing posts, for example, posting in another social network a reaction of their bullying experience, will help to improve the precision of detecting such activities.

Methodology:

Understanding the users' behaviour by taking into account their activities in a few other online platforms. This can help in finding more information leading to a more accurate detection of hate speech texts.

## 6 “Hate Speech Detection Using NLP Techniques” by Shanita Biere, August, 2018.

Objective:

This paper aims to understand how NLP facilitates in detecting cyber bullying texts. As neural network approaches have shown promising results in existing problems of text classification, a deep learning model, the convolution neural network, has also been used.

Methodology:

The classifier divides the texts into three categories- hate speech, offensive speech and neither. The performance has been tested using the accuracy, recall precision and F-score. The classifier model achieved an accuracy of 91% and a precision of 91%.

## 7.” Detecting and Monitoring Hate Speech in Twitter” by Juan Carlos Pereira-Kohatsu , Lara Quijano-Sánchez , Federico Liberatore and Miguel Camacho-Collados, October 2019.

Objective:

The aim of this study is to design a model, HaterNet, which can identify and also classify hate speech texts on Twitter. Along with this, the model can also monitor and analyze prevailing trends of cyber bullying. This models aims to detect triggers of hate speech, especially against minorities. This information may also be valuable for cyber security agencies and police.

Methodology:

The first module, Hate Speech Detection, gathers tweets and classifies them as either containing hate speech or not containing hate speech. The Selection of Document module does Corpus Collection and Cleaning, Labelling, Feature Selection and document Classification.

### 2.2. PROBLEM DESCRIPTION

The use of social media in our daily lives has become excessive. While social media reduces the communication gap, it also increases the vulnerability to online threats, like cyber bullying [blackmail, fraud] etc. Cyberbullying activities take place on various online platforms, where users share their thoughts, interests and issues . Some of the users engage in activities such as bullying, harassment, threat and intimidation making others feel bad about themselves, lowering their self-esteem and discouraging them from interacting or expressing themselves. Anonymity has become a major reason which makes this very easy since these bullies never have to face negative consequences. The problem has been growing since the along with the increase in usage of social media. An eye-opening episode occurred when the Formspring; a social network was closed, due to the number of suicides connected with it, due to many messages containing hateful and abusive language. Social media like Twitter, Facebook, Instagram and YouTube are being criticized for not putting much effort in preventing such crimes. They are under pressure to take major actions in order to prevent and against cyber bullying. The German government has even threatened to fine the such sites, up to 50 million euros if they continue to neglect hateful postings and not take actions within a

week. In order to reduce this problem, we propose a system that can detect cybercrimes from social media automatically.

## 2.3. REQUIREMENT GATHERING

To detect cyberbullying, it is important to identify the types of cyberbullying, and roles in cyberbullying. The types of cyberbullying are of the forms- Threat/Blackmail, Insult, Discrimination, Defamation, Sexual Talk, Encouragement to harasser. The major roles in cyberbullying are- Harasser/Bully, Victim, Bystander- defender/encourager
The requirements are categorized into the functional and non-functional requirements.
The data and texts are taken from twitter comments provided by users and that data is saved as .csv file in our system and any such comment and data in our app is used to find if the data is positive or negative. Also python IDE and Spyder is used for our code to work.Internet facility is also crucial for finding online cybercrimes happening. hence following requirements mentioned above were used in our data.

## 2.4. REQUIREMENT ANALYSIS

We took the data and texts from twitter comments provided by users and that data is saved as .txt file in our system and e use any such comment and data in our app to find if the data is positive or negative. Also we use python IDE and Spyder for our code to work.Internet facility is also crucial for finding online cybercrimes happening .hence following requirements mentioned above were used in our data.

Functional Requirements

- Model should be able to process new social media texts
- Model should be able to analyse and classify text polarity

Non-Functional Requirements

- User friendly
- System should have a higher accuracy
- System should have higher efficiency and response time

Hardware Requirements

- MySQL database server
- New version of Anaconda and all required packages
- Windows/ Linux OS

## 2.5.DATA SOURCE

This classifier is made based on both SVM and Naive Bayes algorithm. The NLTK corpus is a gigantic dump of a wide scope of basic language educational records. Three datasets are used from NLTK corpus which contains various sentiment tweets to train and test the model:
negative_tweets dataset with 5000 tweets with negative emotions
positive_tweets dataset with 5000 tweets with positive emotions
tweets.20150430-223406 dataset with 20000 tweets with no emotions.
The SVM model has been trained on this csv file containing 1000 raw tweets of both positive and negative sentiment and tested upon a csv file containing 250 raw tweets of both positive and negative sentiment extracted from open source. The naive Bayes model has been trained on separate json files of 500 positive and 500 negative tweets each and finally, tested upon a neutral dataset of 20000 tweets, all extracted from open source.

## 2.6. COST ESTIMATION

The model we used to calculate cost is the basic COCOMO Model which is shown as follows:

The basic COCOMO equations take the form

Effort Applied (E) = $a_b(KLOC)^{b_b}$ [ man-months ]

Development Time (D) = $c_b(\text{Effort Applied})^{d_b}$ [months]

People required (P) = Effort Applied / Development Time [count]

| Software project | $a_b$ | $b_b$ | $c_b$ | $d_b$ |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

*Table 2.6.COCOMO model factor values*

our KLOC=1.6(approx)
Software Project=Organic
Effort=2.4*(1.6) ^1.05
=3.931 person-months.

Development time=2.5*(3.93)^0.38
=4.20 months.

## 2.7. PROJECT SCHEDULE

| S. No. | TOPIC | FROM DATE | TO DATE |
|---|---|---|---|
| 1 | LITERATURE SURVEY | 21/11/19 | 25/11/19 |
| 2 | REQ GATHERING | 21/11/19 | 26/11/19 |
| 3 | COST ESTIMATE | 26/12/19 | 03/01/20 |
| 4 | RISK ANALYSIS | 03/01/20 | 23/11/20 |
| 5 | ARCHITECTURE DESIGN | 30/01/20 | 7/02/20 |
| 6 | U I DESIGN | 7/02/20 | 13/02/20 |
| 7 | USE CASE | 14/02/20 | 20/02/20 |
| 8 | SEQUENCE AND CLASS DIAGRAM | 21/02/20 | 27/02/20 |
| 9 | STATE DIAGRAM | 7/03/20 | 14/03/20 |
| 10 | DATA BASE DESIGN | 14/03/20 | 16/03/20 |
| 11 | USER INTERFACE | 16/03/20 | 18/03/20 |
| 12 | MIDDLE WARE | 18/03/20 | 21/03/20 |
| 13 | TESTING | 21/03/20 | 28/03/20 |
| 14 | ANALYSIS | 28/03/20 | 7/04/20 |
| 15 | Mc Calls QUALITY FACTOR | 7/04/20 | 14/04/20 |
| 16 | RESULT | 21/04/20 | 21/04/20 |

| | | | |
|---|---|---|---|
| 17 | RESULT ANALYSIS | 23/04/20 | 25/04/20 |
| 18 | FUTURE ENHANCEMENT | 25/04/20 | 28/04/20 |

*Table 2.7. Project Schedule*

## 2.8. RISK ANALYSIS

**Risk analysis method:** SWOT analysis, Fish Bone

Swot analysis is a common method that are used to analyse risk in major companies. We have also adapted Fish Bone analysis method as through this method we are able to find the cause and effect of our method instantly as it is a graphical method of representing cause and effect.

Risk Analysis is done by Fishbone/ Ishikawa diagram in order to identify all the probable problems/ risks and further sub-problems. The Ishikawa diagram is a cause and effect diagram. It helps to identify the defects, failures and imperfections in a model. The diagram resembles a fish's' skeleton with the head as the main problem and the various causes enlisted down its' spine.

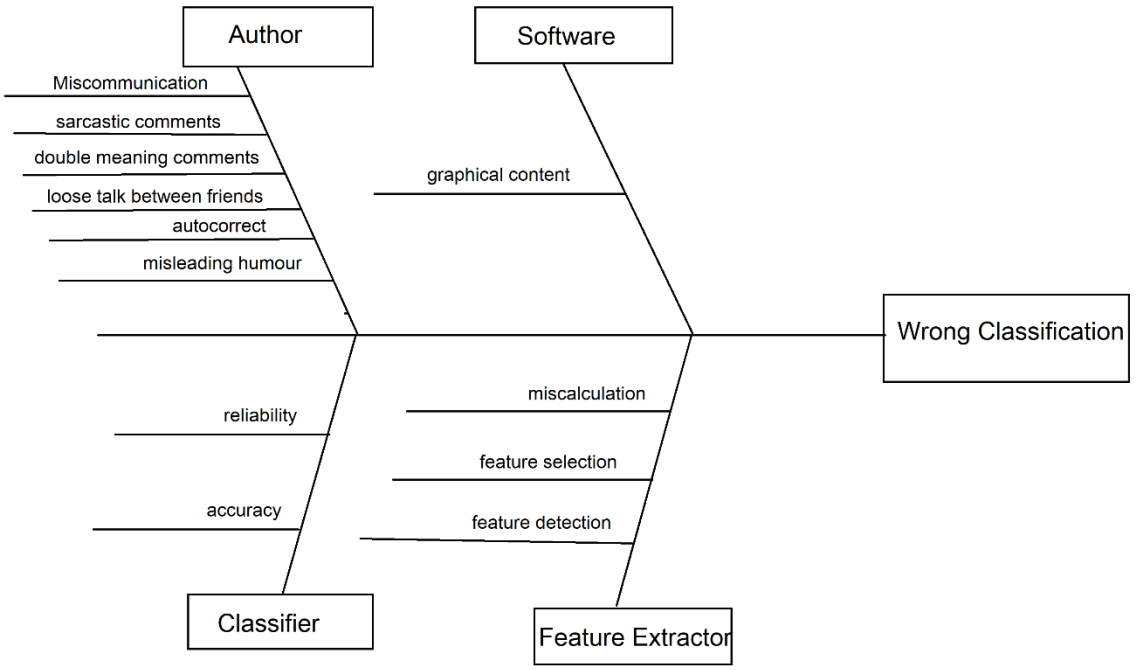


*Figure 2.8. Fishbone Diagram*

## 2.9 Software Requirements Specification

### 2.9.1 Introduction

Cyberbullying activities take place on various online platforms, where users share their thoughts, interests, and issues. Some of the users engage in activities such as bullying, harassment, threat and intimidation making others feel bad about themselves, lowering their self-esteem and discouraging them from interacting or expressing themselves. Anonymity has become a major reason which makes this very easy since these bullies never have to face negative consequences. Hence, in order to reduce this problem, we propose a system that can detect cybercrimes from social media automatically using Naive Bayes Classifier and SVM.

#### 2.9.1.1 Purpose

The purpose of this project is to provide an easy way to find cyber bullying and to eliminate these cases or atleast reduce them by finding such cases automatically.

#### 2.9.1.2 Document Conventions

This document follows MLA (Modem Language Association) format. Bold faced text has been wont to emphasize section and sub-section headings. Highlighting is to means words within the glossary and italicized text is employed to label and recognize diagrams.

#### 2.9.1.3 Intended Audience and Reading Suggestion

The document is to be read by users, developers, staff and the panel of our project team.

### 2.9.2 Overall Description

#### 2.9.2.1 Product Function

- Finding negative comments.
- Giving warning pop up message for hateful comments.
- Reducing number of cyber-bullying cases

#### 2.9.2.2 User Classes and Characteristics

The intended user of the app will be anyone who is currently using twitter to write any comments.

The users should be able to access the following benefits:

- Write positive comment freely.
- Notified if they are writing a hateful comment.
- Reduced cyber-bullying cases.

#### 2.9.2.3 Operating Environment

Operating environment for the Hate Speech Detection System is as listed below.

- Centralized text database
- Operating system: Android, IOS
- Database: Twitter comments
- Platform: Python, Spyder

#### 2.9.2.4 User Documentation

User Documentation will be provided in help section of the app for the non-Technical user. It consists of all the FAQ which the user may have and in case of support whom to ask for help, will consists of all the step by step guidance to search for the location

#### 2.9.2.5 Assumptions and Dependency

Let us assume that this system is used under the following circumstances:

- The user consists of a phone which is either android or iOS
- The company has a basic understanding of English so as to use the app.

### 2.9.3 External Interface Requirement

#### 2.9.3.1 User Interfaces

- Front end: Visual Studio
- Back end: Python, Spyder

#### 2.9.3.2 Hardware Interfaces

- User’s device of choice.

### 2.9.3.3 Software Interfaces

The software used for Hate speech Detection System is listed as follows:

| **Software Used** | |
|---|---|
| | |
| Database | We have used MySQL Database |
| Python | Python was chosen as the language for the backend system as it allows for fast prototyping and allows the programmer to focus on the problems instead of the language. |
| Technologies Used | Tokenisation, SVM, Naïve Bayes, Stop Words, Bag of Words. |

*Table 2.9. Software Used*

**Communication Interfaces**

We intend to communicate through

- Mobile apps
- Internet

## 2.4 Functional Requirement

- System should be able to process new social media texts
- Model should be able to analyse and classify text polarity

### 2.4 Other Non-Functional Requirement

2.5.1 PORTABILITY

The app is made such that it can be downloaded on any phone whether it's android or ios. Thus, making it accessible to end users. No matter what os the user is using it can be used by him.

2.5.2 RELIABILITY

The software tends to works failure free in most of the environmental conditions.

2.5.3 PERFORMANCE

The software works efficiently and updates the users as fast as possible

2.5.4 Security

The software adapts most of the security principles and keeps user data safe.

**2.6 Software Quality Attributes**

- AVAILABILITY: The software system is available at the user's reach.
- CORRECTNESS: The software is expected to provide accurate detection of the activity
- MAINTAINABILITY: The administrator and the faculties are expected to maintain proper information for the smooth usage of the software and keeping it maintained.
- USABILITY: The software should satisfy the security concerns as well as allow for easier security monitoring.

**Business Rules**

- The app is available to user at zero cost.
- Adds will be provided in between  to earn development cost.

**Other Requirements**

- Power supply to the system (electricity).
- A maintenance engineer needs to be present take care of the system.

# CHAPTER3

# ARCHITECTURE & DESIGN

## 3.1. SYSTEM ARCHITECTURE

## MODULES-

- **Feature Extractor-**

Feature Extraction is a process to reduce dimensionality of text by converting it into more manageable groups for further processing. It combines the variables into features, reducing the amount of data to be processed while still precisely describing original data. This reduces amount of redundant data. All the raw tweets in the csv file are converted into feature vectors to fit into the training dataset for SVM.

- **Data Preparation-**

Initially, the data needs to be prepared for classification. This is done by tokenization, data cleaning, removing noise and regular expressions.

- **Classifier Model-**

Classifier model is used to classify the text as ‘hate speech’ or ‘non-hate speech’ according to the percentage of embedded sentiment assigned to it by the sentiment analyzer. The classifier model is prepared by training SVM and Naive Bayes on training datasets and finally evaluating their performance on the testing dataset.

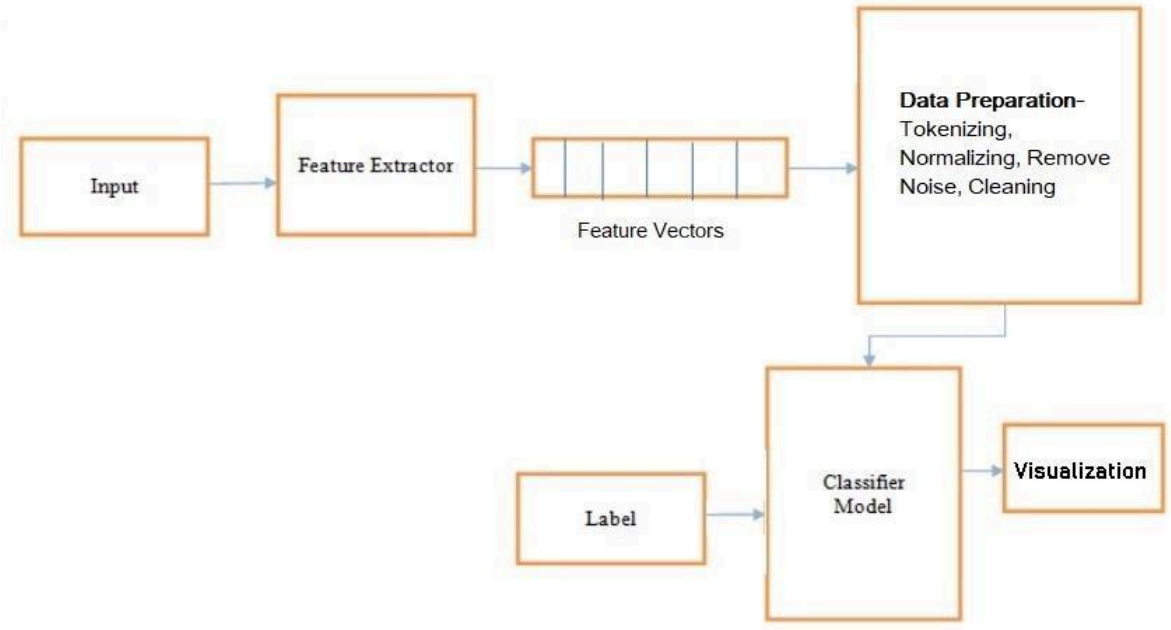

*Figure 3.1.a. Architecture Diagram*

This process can be explained with the help of an example like: -

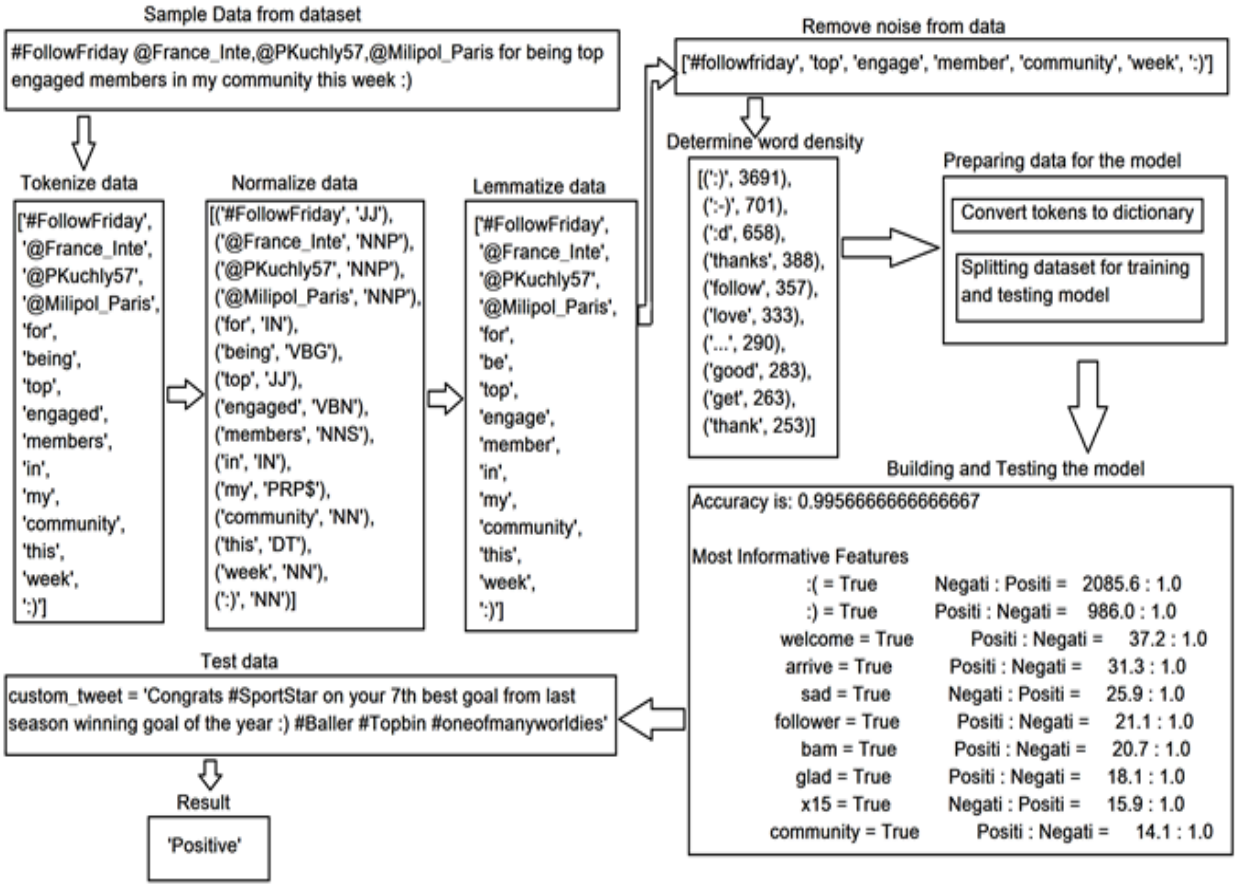


*Figure 3.1.b.Same Text Implementation*

Firstly, the data is tokenized, that is, it is split up into smaller parts called tokens. The punkt tokenizer package is used, which splits the strings into sentences and words. Normalization groups together words with same meaning, for example, “run”, “ran” and “runs”. This reduces the redundancy in the training data. Lemmatization analyses the structure of the word and converts it into normalized form. The tagging algorithm evaluates the context of the words in the data. For example, NN for noun and VB for verb. In the figure, it can be noticed that, “being” changes to “be”, which is its’ root word. Noise means unnecessary text. Noise reduction is done via Regular Expressions to remove hyperlinks, social media handles, punctuation and more. In this step, all the text is converted into lowercase. The common words appearing in negative and positive texts according to the training data are identified using the FreqDist class of NLTK. From the figure, it can be seen that smiling emoticon is commonly present in positive sentences. The negative to positive ratios are assigned to the words of the training dataset. The cleaned data are converted into a dictionary, which is randomly shuffled into a dataset and split for training and testing.

## 3.2. INTERFACE PROTOTYPING (UI)


*Figure 3.2.a. UI-sentiment analysis*

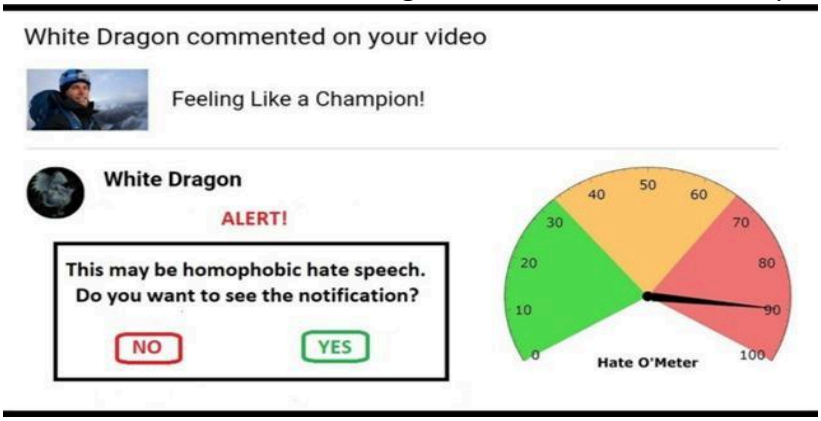


*Figure 3.2.b. UI-Cyber bullying detection*

## 3.3.DATA FLOW DESIGN

In Data flow Diagram it is explained how the raw data flows and eventually turns into the required information that we need for hate speech detection. First the raw data that is tweets are taken and its features are extracted which are basically the words without verbs etc. Then the rest of the data is cleaned. And when we get only cleaned and meaningful words from the text, then we apply classification algorithm to understand the polarity of the sentence and also whether it points towards cyberbullying or not.

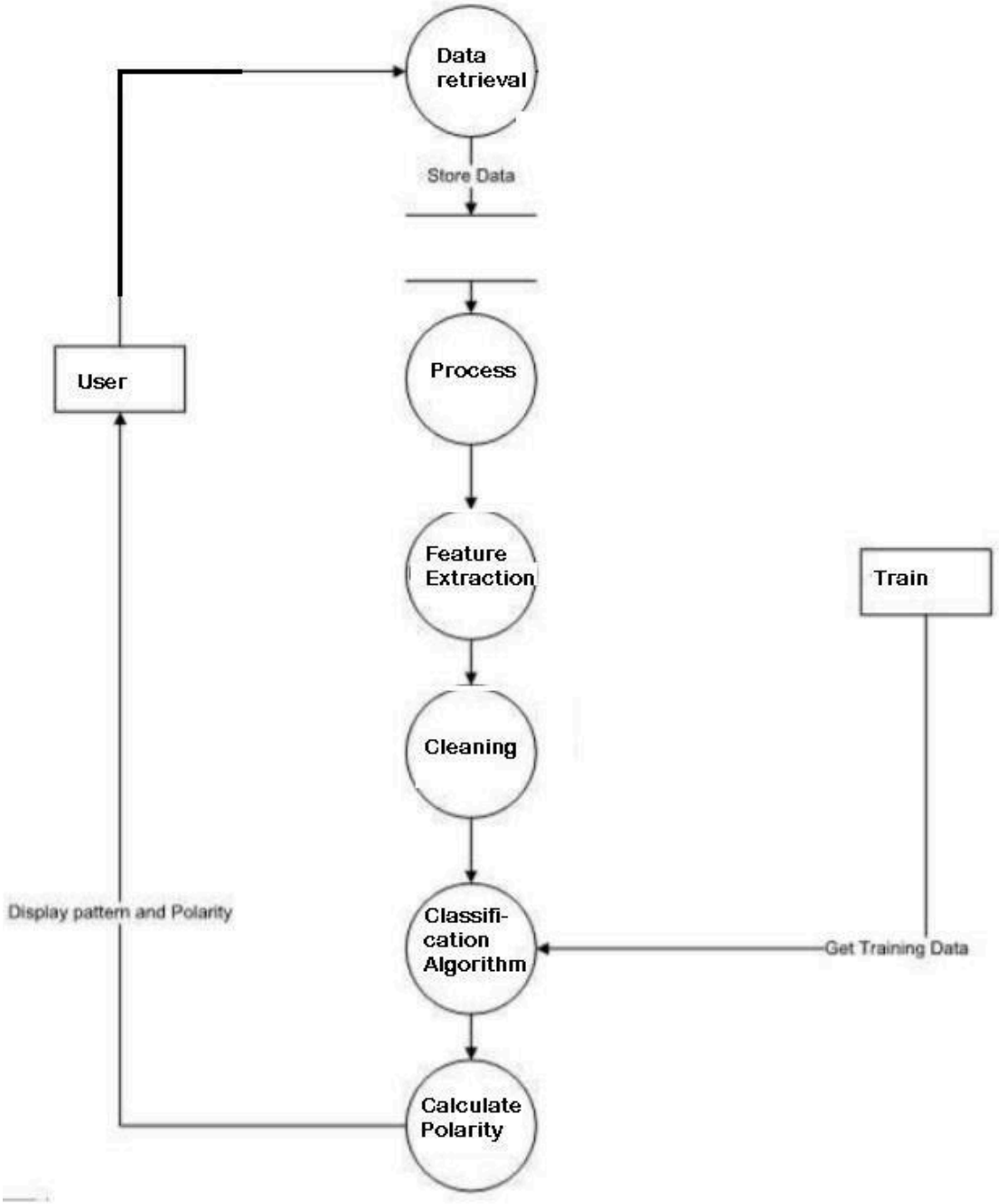


*Figure 3.3. Data flow diagram*

## 3.4.USE CASE DIAGRAM

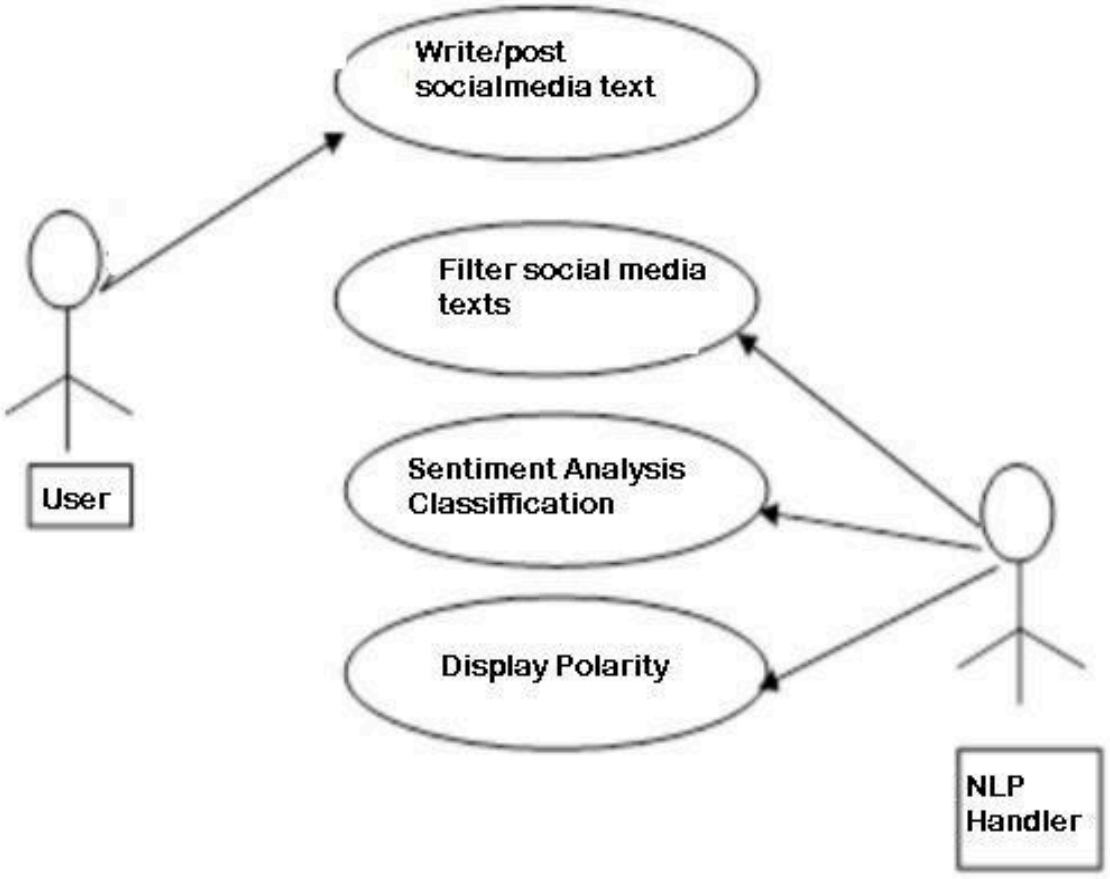


*Figure 3.4. Use Case diagram*

In this diagram, it is explained that the user simply puts the comment on social media. Then Natural Language Processor Handler filters the text and performs sentiment Analysis to finally display polarity of the text.

## 3.5. SEQUENCE DIAGRAM

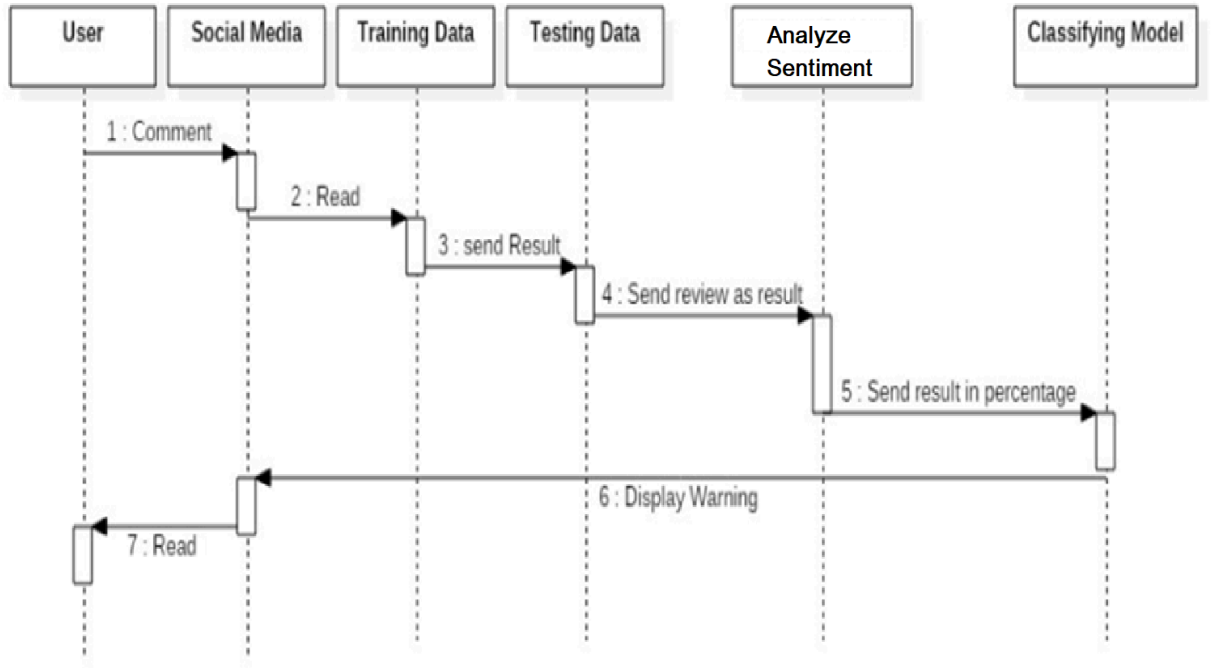


*Figure 3.5. Sequence diagram*

Sequence diagram depicts how the data flor occurs in a sequence. Initially the user comments on social media, Then the data is read and tested. The result of the test is sent to analyse sentiment of the text. After this classifier modl finally displays the polarity of he text and warning is displayed on the screen in case the text shows positive to hate speech detection.

## 3.6.CLASS DIAGRAM

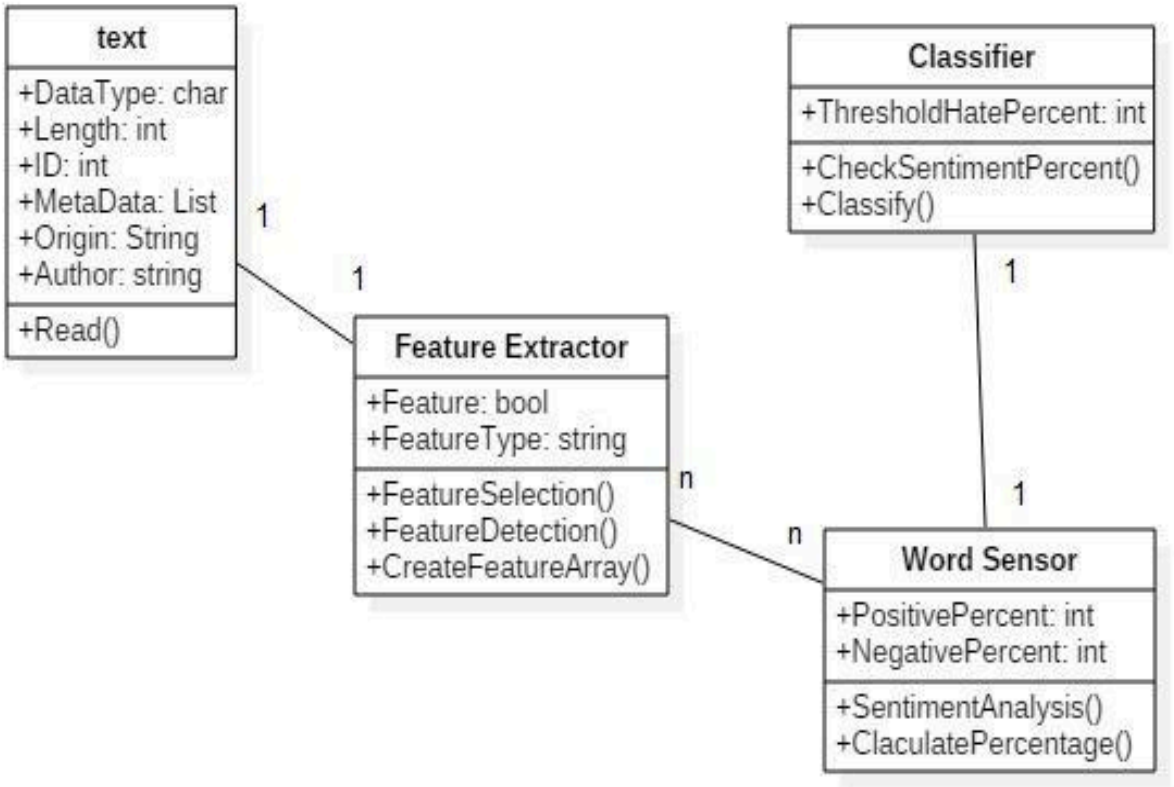


*Figure 3.6. Class diagram*

The class diagram represents the entities and classes that are involved in Hate Speech Detection process. The classes have their properties and respective functions which are shown in the class diagram above.

## 3.7. INTERACTION DIAGRAM

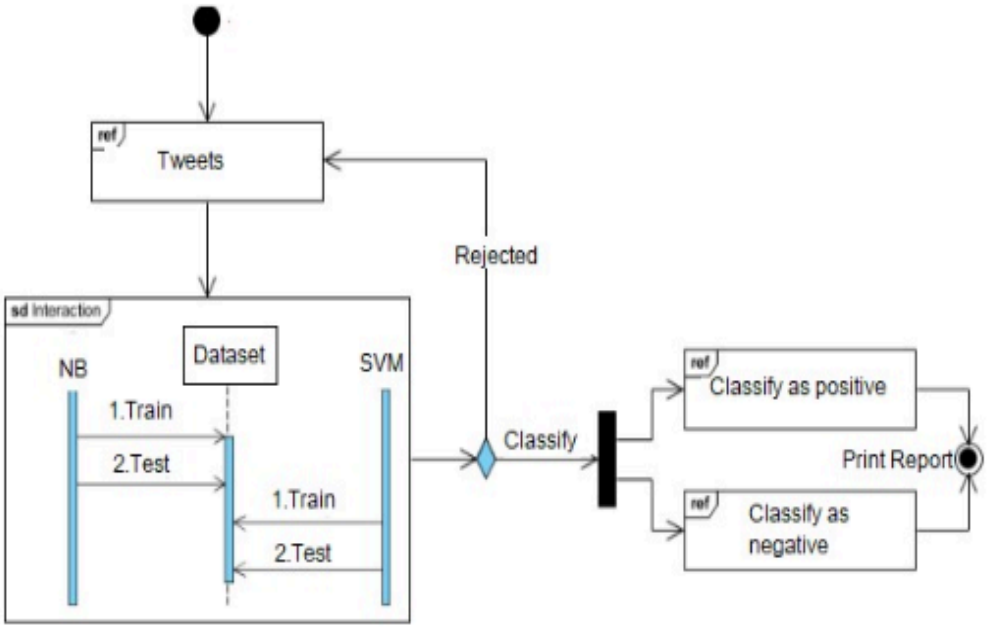


***Figure 3.7. Interaction diagram***

Interaction Diagram explains how the elements interact between each other. Tweets directly interact with Feature extraction and then then they are classified and if they are proven to be negative comments then they are printed as warning.

## 3.8. STATE/ ACTIVITY DIAGRAM

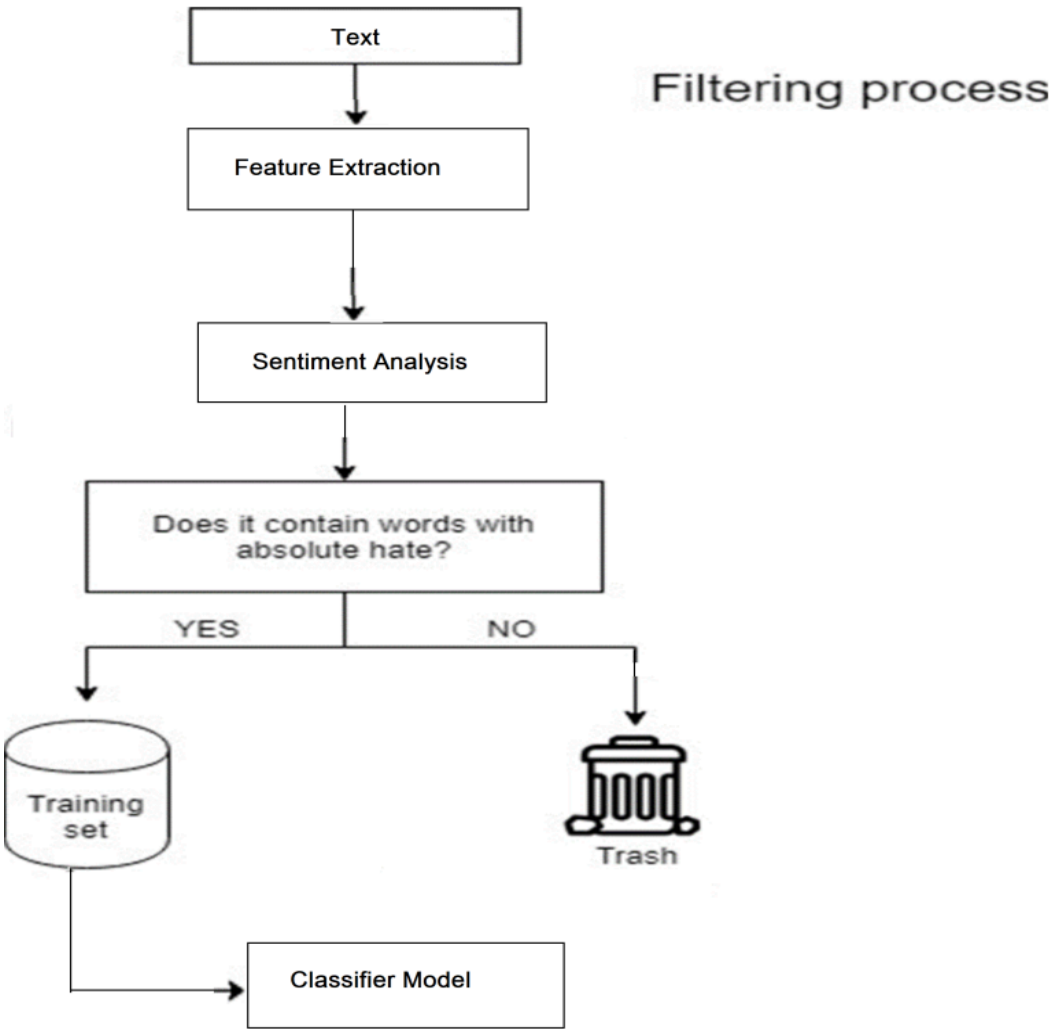


*Figure 3.8. Activity diagram*

Activity Diagram are used describe the steps in a use case diagram which can be either sequential or concurrent. First the raw data that is tweets are taken and its features are extracted which are basically the words without verbs etc. Then the rest of the data is cleaned. And when we get only cleaned and meaningful words from the text, then we apply classification algorithm to understand the polarity of the sentence and also whether it points towards cyberbullying or not.

## 3.9. COMPONENT & DEPLOYMENT DIAGRAM

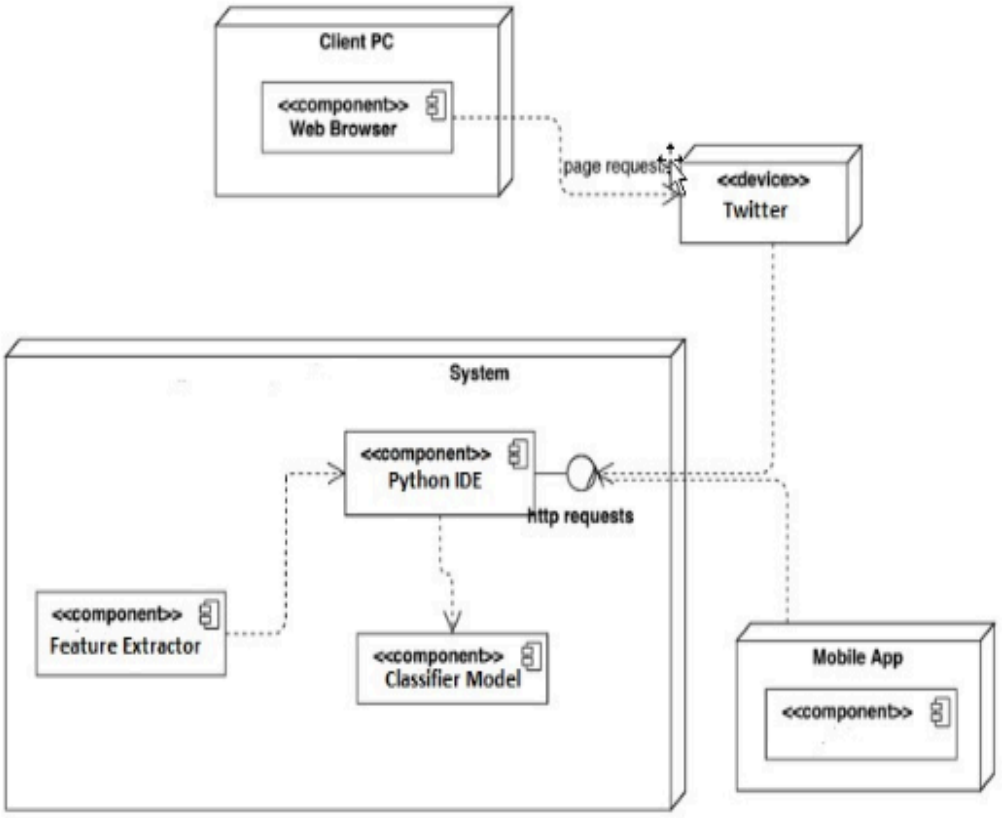


*Figure 3.9. Component diagram*

Component Diagram is used to tell the physical aspects of a system.Here all the physical aspects like Python IDE, Feature Extractor, Classifier model and mobile application are used as shown in the diagram.

# CHAPTER4

# IMPLEMENTATION

## MODULES-

1) **Feature Extractor-**

   Feature Extraction is a process to reduce dimensionality of text by converting it into more manageable groups for further processing. It combines the variables into features, reducing the amount of data to be processed while still precisely describing original data. This reduces amount of redundant data.
   The method of feature extraction used in this project is Bag-of-words. It uses NLP to extract words and classify them by frequency of occurrence.

2) **Sentiment Analyzer-**

   Sentiment Analysis is the most common text classification tool that analyses the social media text and indicates if the underlying sentiment of the text is positive or negative.

3) **Classifier Model-**

   Classifier model is used to classify the text as 'hate speech' or 'non-hate speech' according to the percentage of embedded sentiment assigned to it by the sentiment analyzer. This will be done by the application of both, Naïve Bayes and SVM model.

The methodology behind the annotation process is simplistic. Natural language processing (NLP) is used to convert text into numbers and vectors which are easily understandable by the machine. To proceed with our motivation, data cleaning must be done. Feature Extraction is a process to reduce dimensionality of text by converting it into more manageable groups for further processing. It combines the variables into features, reducing the amount of data to be processed while still precisely describing original data. This reduces amount of redundant data.

The method of feature extraction used in this project is Bag-of-words. It uses NLP to extract words and classify them by frequency of occurrence. This includes removing stop words, stemming, tokenization, etc. Once the data is clean, the method of classification to be used is sentiment analysis for hate speech detection. Sentiment analysis is the process of analysing text I order to determine its' emotional tone. It will categorize a line as positive or negative using sentiment score which reflects the depth of emotions in the text. Sentiment Analysis is mostly used as a classification tool that analyses the social media text and indicates if the emotion behind the text is positive or negative.
After the data is separated based on sentiment, now we need to classify the data as hateful or not and for this task we need a classifier model .Classifier model is used to classify the text as 'hate speech' or 'non-hate speech' according to the percentage of embedded sentiment assigned to it by the sentiment analyser.

The rule based algorithms' aim is to identify and utilize a set of related rules that is capable of representing the information gained by the system. After this is done, we need a classifier model for classifying the data and for that we have used is Naïve Bayes Algorithm. This classifier was created using existing computing tools. We used to scikit-learn a python

package than implements most of the machine learning algorithms including naïve Bayes algorithm and feature extraction techniques. The classifier model is created using the multinomial () function which is part the scikit-learn package. The package also contains functions like Count Vectorizer () for bag of words implementation and transforming documents to feature vectors.

In addition to Naïve Bayes Algorithm, SVM is also used as a comparative analysis for greater accuracy of classification. SVM (Support Vector Machine) is a supervised learning algorithm used for classification purposes. The values are plotted and a hyperplane is chosen in such a way that it maximizes the margin of the training data. Once the training is done, the input data is processed by the classifier as positive or negative indicating presence of cyber bullying or not.

Both, SVM (Support Vector machine) and Naïve Bayes algorithm have been used to create a classification model. The test data is classified by both the models separately, and, on comparison of the results, a cumulative result is shown; whether text is 'cyber bullying detected' or ' cyber bullying not detected'.

## 4.1. DATABASE DESIGN

### 4.1.1.ER DIAGRAM

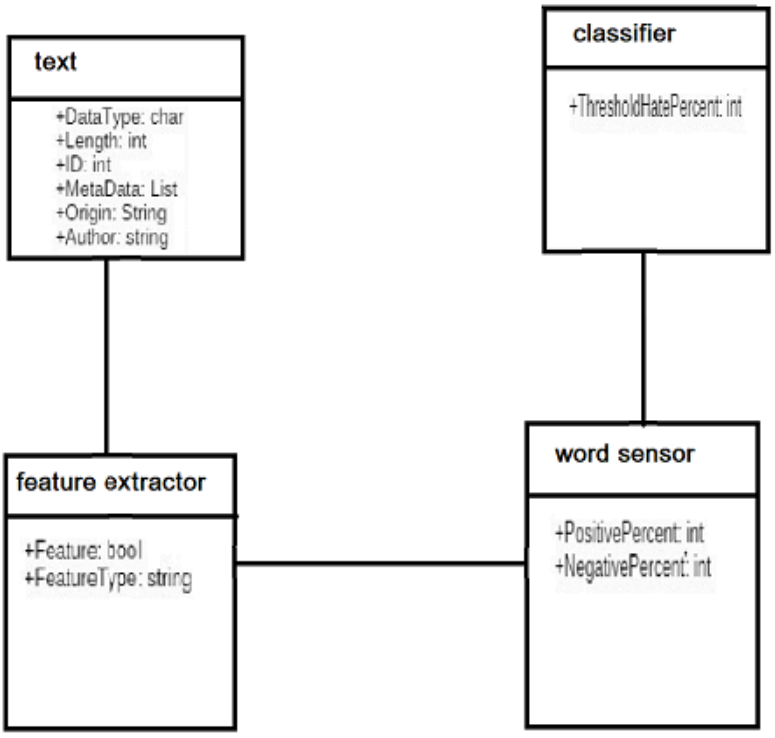


*Figure 4.1.1. ER diagram*

ER Diagram explain how all the entities interact with each other in the diagram with the respective functions they have.

## 4.1.2RELATIONAL DIAGRAM

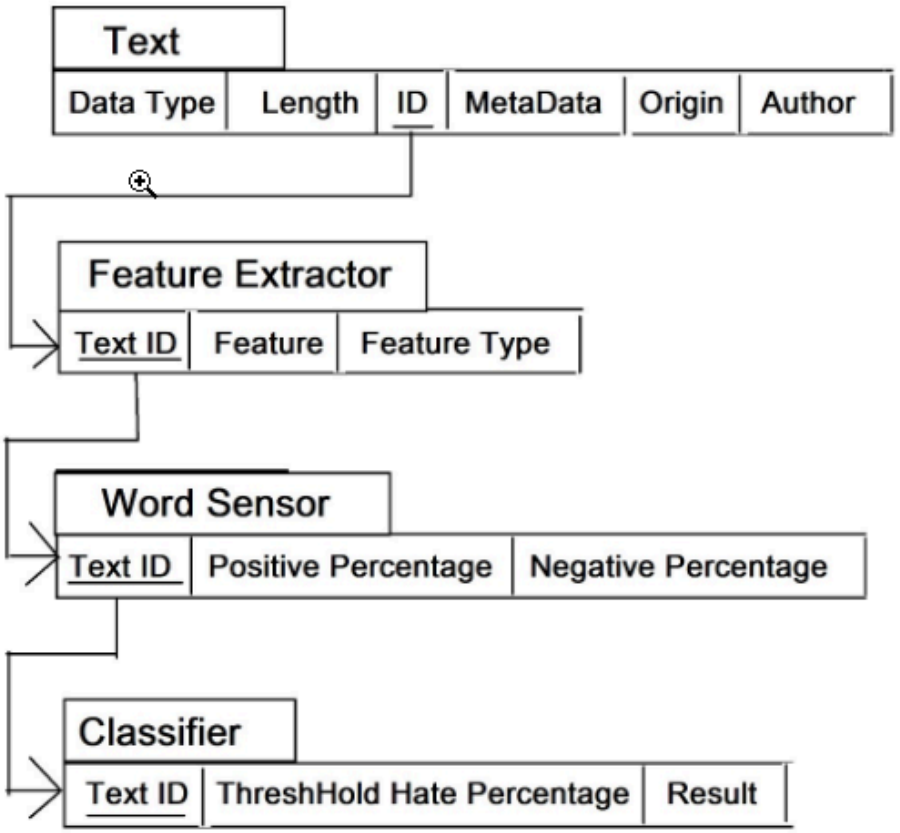


*Figure 4.1.2. Relational diagram*

Relational Diagram basically shows the visual representation of the entities in the system Hence the diagram shows all the entities along with their properties used.

## 4.2.USER INTERFACE


*Figure 4.2.a. UI-sentiment analysis*

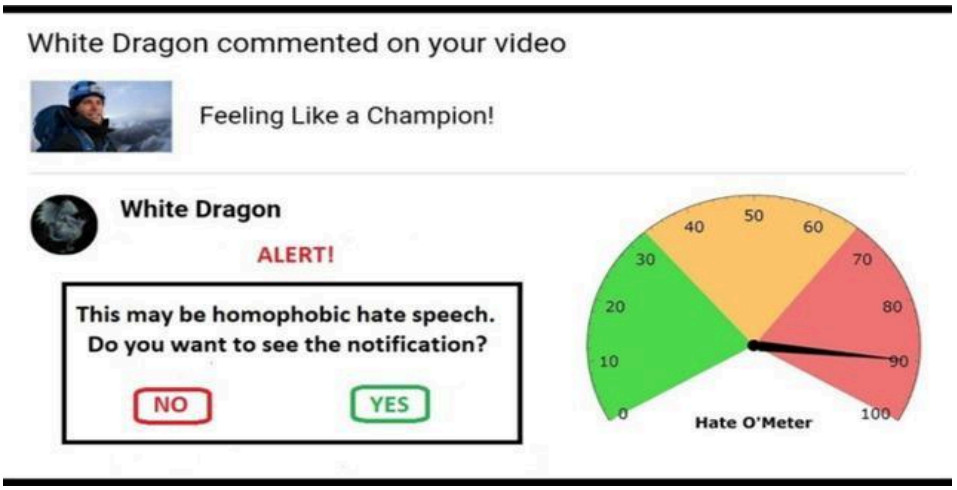


*Figure 4.2.b. UI-Cyber bullying detection*

The user can interface through the mobile app where he can look whether the data is hateful or not.

## 4.3. MIDDLEWARE-

- **Anaconda:** is a package manager, a Python/R data science distribution, and an assortment of more than 7,500 open-source bundles
- **Spyder:** Spyder (Scientific Python Development Environment) is a free IDE that's included within Anaconda.
- **Python: It** is a deciphered, high-level, general universally useful programming language It gives builds that empowers simple programming on both small and large scales.

# CHAPTER5

# VERIFICATION & VALIDATION

## 5.1. UNIT TESTING

Unit Testing is an integral part of the testing. It helps to test each module separately. We adopt python pytest testing methodology and prepare some test cases to check the behaviour after analysing the behaviour we divided the test cases into pass and fail.

| Test ID | Test Scenario | Test Step | Test Data | Expected Output | Actual Output |
|---|---|---|---|---|---|
| 1 | Cyber bullying Detection | Enter Social Media text | I hate you. You are so fat @raashi_77 ! | Detect Cyber bullying | Cyber bullying Detected |
| 2 | Cyber bullying Detection | Enter Social Media text | Hey Guys please follow for more updates.Love you peeps. Keep RepostingJ | No Cyber bullying Detected | No Cyber bullying Detected |
| 3 | Cyber bullying Detection | Enter Social Media text | Wow This is so amazing! ! <3 | No Cyber bullying Detected | No Cyber bullying Detected |
| 4 | Cyber bullying Detection | Enter Social Media text | @rinamalhotra is so ugly. I dont undertand who voted for her. She doesn't deserve the crown. She must be eliminated. Boo! | Detect Cyber bullying | Cyber bullying Detected |
| 5 | Cyber bullying Detection | Enter Social Media text | Congrats @SportChampionRio on your yet again amazing goal. Tis the winning goal of championship:) #BESTPLAYER #BIGGESTFAN #ILY | No Cyber bullying Detected | No Cyber bullying Detected |
| 6 | Cyber bullying Detection | Enter Social Media text | @rakshitshah is a failure. A mere joke in the name of a stand up comic. Shut up please! :P | Detect Cyber bullying | Cyber bullying Detected |

| | | | | | |
|---|---|---|---|---|---|
| 7 | Cyber bullying Detection | Enter Social Media text | OMG !! @shrutirani is so damn BEAUTIFUL. <3 <3 | No Cyber bullying Detected | No Cyber bullying Detected |
| 8 | Cyber bullying Detection | Enter Social Media text | Disgusting. Hope they Shoot themselves. #LOL | Detect Cyber bullying | Cyber bullying Detected |
| 9 | Cyber bullying Detection | Enter Social Media text | Your song is so annoying @harshraj65 . Please do not post your stupid songs XD | Detect Cyber bullying | Cyber bullying Detected |
| 10 | Cyber bullying Detection | Enter Social Media text | Just Checked in at @hotelgrande. Feeling blessed #sunkissed #beachready | No Cyber bullying Detected | No Cyber bullying Detected |

In this testing we have taken10 test cases with different datasets and have analysed the expected output .After running the code we check if the actual output matches with the expected output and result of each of the test cases is shown as follows:

## 5.2. INTEGRATION TESTING

Integration Testing is used to test the overall behaviour of our product. We adopted manual testing methodology for our integration testing. Here is the table of our test cases:

*Table 5.2.a. Integration Testing- Manual Testing*

All the texts provided are successfully predicted as shown above.

```
In [36]: runfile('C:/Users/Snigdhabose/Anaconda3/lib/ssl.py', wdir='C:/Users/Snigdhabose/Anaconda3/lib')
-------------SVM Classifier-----------------

Sample Train Data for SVM Classifier:
                                                Content Label
1016  synopsis : a humorless police officer's life c...   neg
57    'pleasantville' ( 1998 ) taps into hollywood's...   pos
79    118 minutes ; not rated ( though i suspect it ...   pos
243   i want to correct what i wrote last year in my...   pos
452   plot : a peculiar french girl grows up lonely ...   pos

SVM Classifier Report:-
Training time: 8.445000s; Prediction time: 0.855000s
*NOTE: F1 = 2 * (precision * recall) / (precision + recall)
('positive: ', {'recall': 0.91, 'f1-score': 0.9145728643216081, 'support': 100L, 'precision': 0.9191919191919192})
('negative: ', {'recall': 0.92, 'f1-score': 0.9154228855721394, 'support': 100L, 'precision': 0.9108910891089109})

-------------NB Classifier-----------------
Naive Bayes Classifier Report:-
('Accuracy is:', 0.996)
**MOST COMMON INFORMATIVE FEATURES ARE:-**
Most Informative Features
                    :( = True           Negati : Positi =   2054.0 : 1.0
                   sad = True           Negati : Positi =     33.5 : 1.0
              follower = True           Positi : Negati =     20.0 : 1.0
               welcome = True           Positi : Negati =     15.1 : 1.0
                   x15 = True           Negati : Positi =     14.9 : 1.0
               awesome = True           Positi : Negati =     13.9 : 1.0
             community = True           Positi : Negati =     13.8 : 1.0
                  blog = True           Positi : Negati =     13.8 : 1.0
                   via = True           Positi : Negati =     12.4 : 1.0
                 enjoy = True           Positi : Negati =     12.4 : 1.0
None

-------------Testing the models-----------------
SAMPLE TEXT1:
@shivangi234 is so fat. hahaha. She didn't even deserve the title of Miss glam2020. She is so ugly and fat too. All these shows are a scam.
('NB Classifier Result:', 'Negative')
('SVM Classifier Result:', 'Negative')
**RESULT**
CyberBullying is Detected (using SVM Classifier and naive Bayes Classifier)
```

*Figure 5.2.b. Integration Testing-Output*

## 5.3.USER TESTING

We let our customers test our services. We collected feedbacks and analyse them.

```
In [10]:

In [10]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9973333333333333)
('I ordered just once from TerribleCo, they screwed up, never used the app again.',
'Negative')

In [11]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.995)
('wow this is so amazing !!', 'Positive')

In [12]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.992)
('I hate you. You are so fat.', 'Negative')

In [13]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9953333333333333)
('Hey guys. Please follow me for more updates. love you peeps. keep reposting :)',
'Positive')

In [14]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9946666666666667)
("@rinamalhotra is so ugly. I dont undertand who voted for her. She doesn't deserve
the crown. She must be eliminated. Boo!", 'Negative')

In [15]:
```

*Figure 5.3. User Testing*

## 5.4. SIZE-LOC

| | **Expected** | **Actual** |
|---|---|---|
| App KLOC | 20 | 14 |
| Web KLOC | 200 | 170 |
| Backend KLOC | 40 | 43 |
| Total KLOC | 260 | 218 |

*Figure 5.4. Size-LOC*

## 5.5. COST ANALYSIS

**Effort Applied (E)** = $a * (KLOC)^{b}$

**Development Time (D)** = $c * (Effort\ Applied)^{d}$

*Software Project is organic*

| | **Expected** | **Actual** |
|---|---|---|
| **KLOC** | 20(App) + 200(Web) + 40(Backend) = 260 | 14(App) + 170(Web) + 43(Backend) = 216 |
| **E** | $2.4 * (270)^{1.05}$ = 824.01 man-months | $2.4 * (218)^{1.05}$ = 494.92 man-months |
| **D** | $2.5 * (824.01)^{0.38}$ = 32.06 months | $2.5 * (494.92)^{0.38}$ = 26.41 months |
| | | |

*Figure 5.5. Cost Analysis*

## 5.6. DEFECT ANALYSIS

Risk Analysis is done by Fishbone/ Ishikawa diagram in order to identify all the probable problems/ risks and further sub-problems. The Ishikawa diagram is a cause and effect diagram. It helps to identify the defects, failures and imperfections in a model. The diagram resembles a fishs' skeleton with the head as the main problem and the various causes enlisted down its' spine.

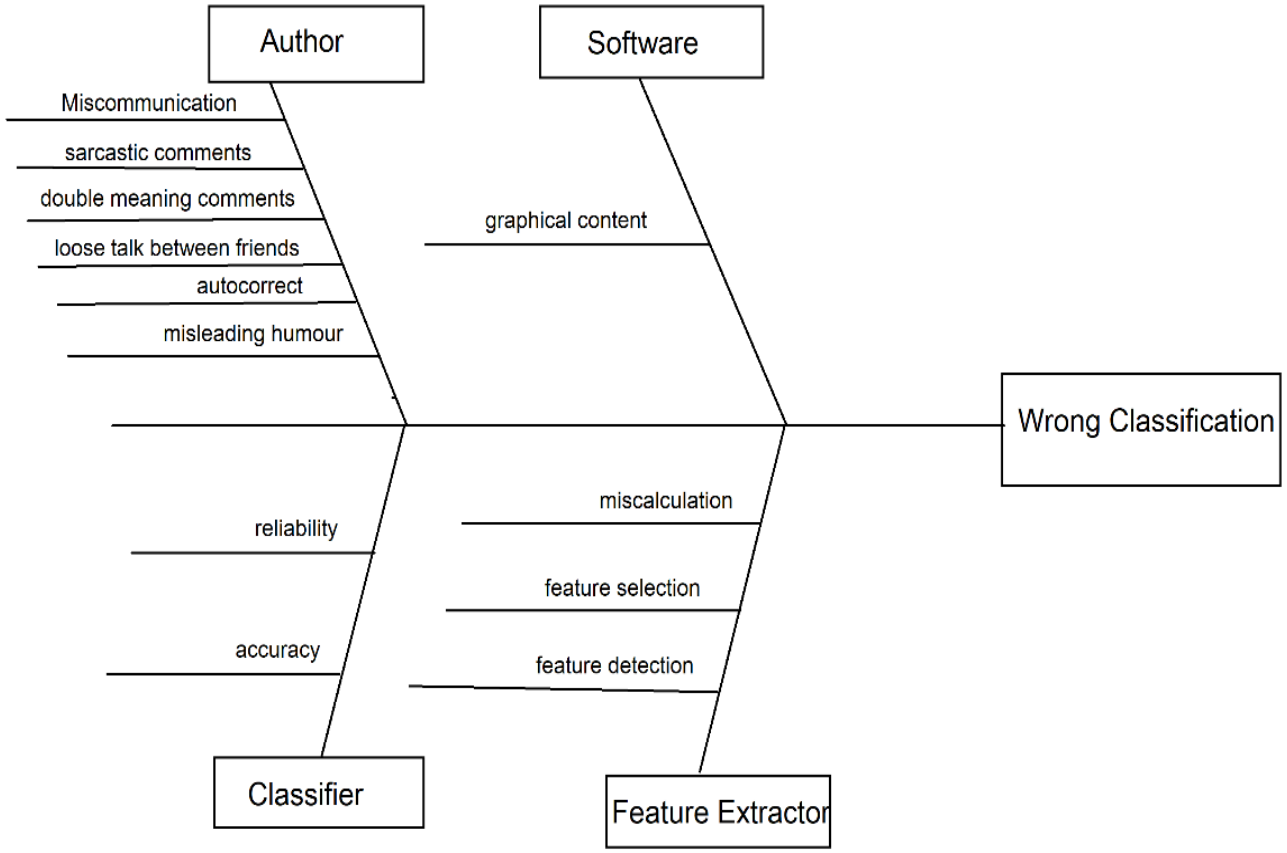


*Figure 5.6. Fishbone diagram*

## 5.7.MC CALL'S QUALITY FACTORS

- Efficiency-
  Efficiently detect 99% of negative and abusive texts
- Correctness-
  The results are accurate. It determines accurate percent of hatefulness in all the texts provided and can distinguish between different positive and negative texts appropriately.
- Usability-
  The app is available to user 24x7 for parking.
- Flexibility-
  The app works on any platform whether its android or ios . It available at all devices. If the user changes its devices, then also its data can be easily transferred from one device to other.

# CHAPTER6

# EXPERIMENT RESULTS & ANALYSIS

## 6.1.RESULTS

The model successfully identifies the most informative features and builds a naive bayes and SVM classification on the basis of the training data provided.

```
In [8]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
[(u':)', 3691), (u':-)', 701), (u':d', 658), (u'thanks', 388), (u'follow', 357),
(u'love', 333), (u'...', 290), (u'good', 283), (u'get', 263), (u'thank', 253)]
('Accuracy is:', 0.994)
Most Informative Features
                    :( = True           Negati : Positi =   2042.5 : 1.0
                    :) = True           Positi : Negati =   1008.2 : 1.0
              follower = True           Positi : Negati =     36.6 : 1.0
              followed = True           Negati : Positi =     27.0 : 1.0
                   sad = True           Negati : Positi =     18.1 : 1.0
                justin = True           Negati : Positi =     17.9 : 1.0
             community = True           Positi : Negati =     13.3 : 1.0
             goodnight = True           Positi : Negati =     13.3 : 1.0
                 didnt = True           Negati : Positi =     13.3 : 1.0
          unfortunately = True           Negati : Positi =     12.7 : 1.0
None
('I ordered just once from TerribleCo, they screwed up, never used the app again.',
'Negative')
```

*Figure 6.1.a. Result output*

Accuracy of both the models are high. They predict same outcomes in all tested cases. Comparative Analysis is shown here below-

```
In [36]: runfile('C:/Users/Snigdhabose/Anaconda3/lib/ssl.py', wdir='C:/Users/Snigdhabose/Anaconda3/lib')
-------------SVM Classifier-----------------

Sample Train Data for SVM Classifier:
                                                Content Label
1016  synopsis : a humorless police officer's life c...   neg
57    'pleasantville' ( 1998 ) taps into hollywood's...   pos
79    118 minutes ; not rated ( though i suspect it ...   pos
243   i want to correct what i wrote last year in my...   pos
452   plot : a peculiar french girl grows up lonely ...   pos

SVM Classifier Report:-
Training time: 8.445000s; Prediction time: 0.855000s
*NOTE: F1 = 2 * (precision * recall) / (precision + recall)
('positive: ', {'recall': 0.91, 'f1-score': 0.9145728643216081, 'support': 100L, 'precision': 0.9191919191919192})
('negative: ', {'recall': 0.92, 'f1-score': 0.9154228855721394, 'support': 100L, 'precision': 0.9108910891089109})

-------------NB Classifier-----------------
Naive Bayes Classifier Report:-
('Accuracy is:', 0.996)
**MOST COMMON INFORMATIVE FEATURES ARE:-**
Most Informative Features
                     :( = True           Negati : Positi =   2054.0 : 1.0
                    sad = True           Negati : Positi =     33.5 : 1.0
               follower = True           Positi : Negati =     20.0 : 1.0
                welcome = True           Positi : Negati =     15.1 : 1.0
                    x15 = True           Negati : Positi =     14.9 : 1.0
                awesome = True           Positi : Negati =     13.9 : 1.0
              community = True           Positi : Negati =     13.8 : 1.0
                   blog = True           Positi : Negati =     13.8 : 1.0
                    via = True           Positi : Negati =     12.4 : 1.0
                  enjoy = True           Positi : Negati =     12.4 : 1.0
None

-------------Testing the models-----------------
SAMPLE TEXT1:
@shivangi234 is so fat. hahaha. She didn't even deserve the title of Miss glam2020. She is so ugly and fat too. All these shows are a scam.
('NB Classifier Result:', 'Negative')
('SVM Classifier Result:', 'Negative')
**RESULT**
CyberBullying is Detected (using SVM Classifier and naive Bayes Classifier)
```

*Figure 6.1.b. SVM and Naive bayes output*

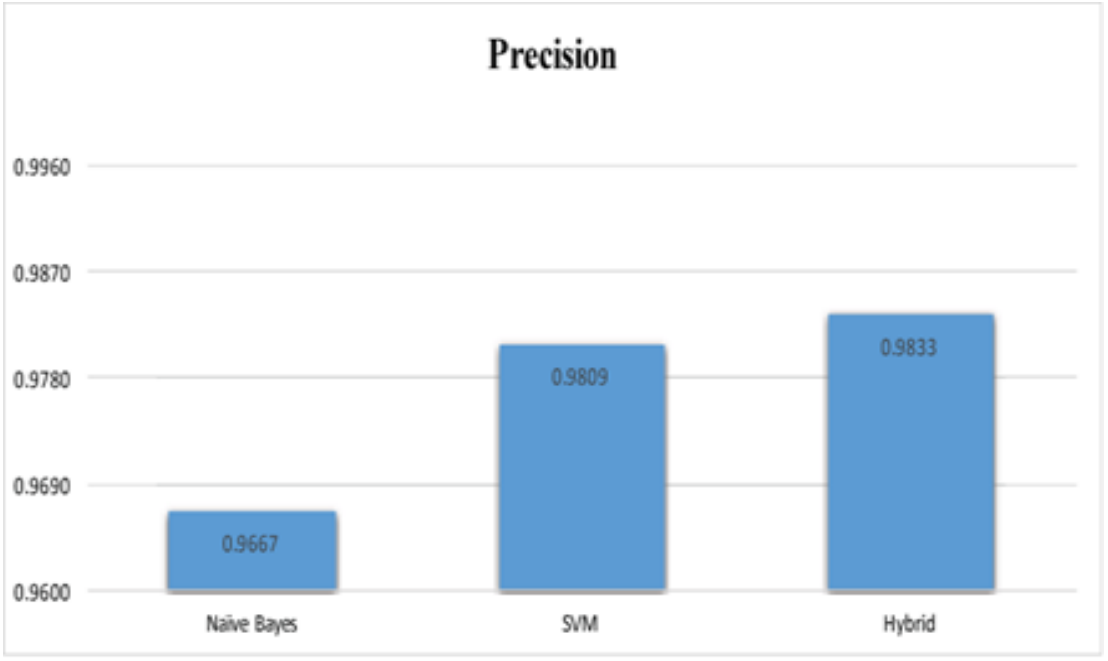


*Figure 6.1.c. SVM and naive bayes – precision comparison*

The model is able to successfully classify sentences as positive and negative as shown here below-

```
In [10]:

In [10]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9973333333333333)
('I ordered just once from TerribleCo, they screwed up, never used the app again.',
'Negative')

In [11]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.995)
('wow this is so amazing !!', 'Positive')

In [12]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.992)
('I hate you. You are so fat.', 'Negative')

In [13]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9953333333333333)
('Hey guys. Please follow me for more updates. love you peeps. keep reposting :)',
'Positive')

In [14]: runfile('C:/Users/Snigdhabose/Desktop/major proj/code/untitled1.py',
wdir='C:/Users/Snigdhabose/Desktop/major proj/code')
('Accuracy is:', 0.9946666666666667)
("@rinamalhotra is so ugly. I dont undertand who voted for her. She doesn't deserve
the crown. She must be eliminated. Boo!", 'Negative')

In [15]:
```

*Figure 6.1.d. Sample Results*

## 6.2.RESULT ANALYSIS

TP : TRUE POSITIVE

TN : TRUE NEGATIVE

FP : FALSE POSITIVE

FN : FALSE NEGATIVE

Accuracy means overall how many time the algorithm was correct

Accuracy= (TP + TN)/ (TP + TN + FP + FN)

Precision means how often is the output correct

Precision= (TP)/ (TP + FP)

## CONFUSION MATRIX OF NAÏVE BAYES

| n= 10000 | Predicted: NO | Predicted: YES |
|---|---|---|
| Actual: NO | 4975 | 25 |
| Actual: YES | 25 | 4975 |

*Figure 6.2.a Confusion matrix- Naïve Bayes*

Accuracy = 95%

Precision = 81%

| n= 10000 | Predicted: NO | Predicted: YES |
|---|---|---|
| Actual: NO | 4995 | 5 |
| Actual: YES | 5 | 4995 |

*Figure 6.2.b Confusion matrix- SVM*

Accuracy = 99%

Precision = 90%

| dataset of 10000 texts | | |
|---|---|---|
| | SVM Classifier | Naive Bayes Classifier |
| Training time | 8.4 s | 4.2 s |
| Testing Time | 0.85 s | 0.83 s |
| Precision score | 0.90 | 0.81 |
| | | |

*Figure 6.2.c. Comparison Metrics- SVM and Naive Bayes*

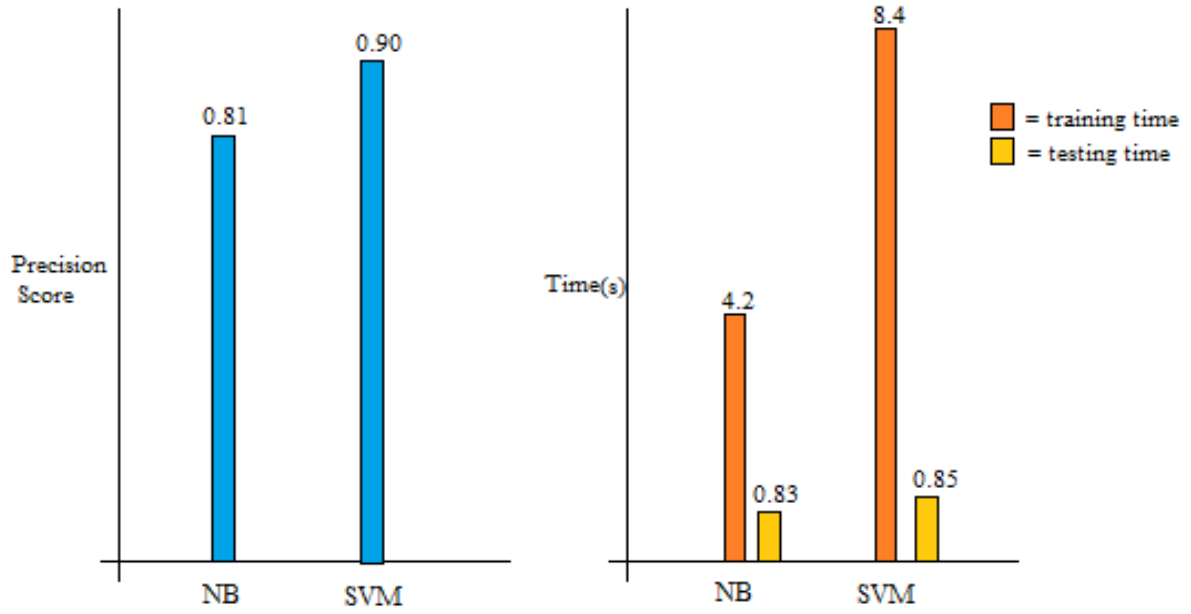


*Figure 6.2.d. Comparison plots- SVM and Naïve Bayes*

## 6.3. CONCLUSION & FUTURE WORK

Based on the objectives of this study, we were able to create a hate speech classifier using Naive Bayes algorithm and SVM. We were also able to collect and label a number of tweets from twitter website and used the labelled dataset to test and train the hate speech classifier. We managed to get an average precision score of 95%.
One of the limitations when using Naïve Bayes algorithm is the assumption of independent features. This is not the ideal situation when dealing with tweets or any others documents.

Despite this limitation, Naïve Bayes still performs well and can be enhanced by using more trigrams or longer word combinations.
Although SVM is good for text classification but SVM algorithm is not suitable for large data sets and large texts .Also one major drawback that is faced using this system is that it does not read false positive and true negative comments. For example .if the text says ,"I am not very happy", then the application might take the word "very" and "happy" and consider it to be positive which is wrong. Also, the application is unable to read GIFS and audio and videos .Another area where it can be improved in future s that it presently reads and analyses text which is written only in English language and people also sometimes use regional language which then changes the sentiment of the overall text sometimes.
Also, the application is unable to read and understand any sarcastic comment. For example ,if the text says ,"oh my god ,you are so intelligent ,hahaha" ,the application would consider it as a positive comment .All these are areas which show a way for improvement in this application in future.

This research will add to the body of knowledge in the field of curbing online hate, online hate speech monitoring, social media data mining and application of machine learning algorithms to solve real life problems.

# CHAPTER9

# REFERENCES


[1].Deepika Mallampati," An Efficient Spam Filtering using Supervised Machine Learning Techniques", International Journal of Scientific Research in Computer Science and Engineering , April 2018.

[2]. Shafigh Askia, Navid Khalilzadeh Souratib," Proposed efficient algorithm to filter spam using machine learning techniques", Pacific Science Review A: A Natural Science and Engineering, July 2016.

[3] .Varsha Malik, Sanjay Kumar," A Review of Spam Detection using Machine Learning", International Journal of Digital Application & Contemporary Research September 2016.

[4]. Hong Yang, Qihe Liu, Shijie Zhou and Yang Luo," A Spam Filtering Method Based on Multi-Modal Fusion", American Chemical Society, February 2016.

[5] Izzat Alsmadi, Ikdam Alhami," Clustering and classification of email contents", Journal of King Saud University – Computer and Information Sciences, January 2015.

[6] Asmeeta Mali, "Spam Detection Using Baysian with Pattren Discovery", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-3, July 2013.

[7] F. Smadja, H. Tumblin, "Automatic spam detection as a text classification task", in: Proc. of Workshop on Operational Text Classification Systems, 2002.

[8] Ann Nosseir , Khaled Nagati and Islam Taj-Eddin, "Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013 ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784.

[9] Paula Fortuna," Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes.", 2017.

[10] Sean McElwee. "The case for censoring hate speech. https://www. alternet.org/civil-liberties/case-censoring-hate-speech",2013.

[11] David Robinson, Ziqi Zhang, and Jonathan Tepper," Hate speech detection on twitter: Feature engineering v.s. feature selection.", 2018.

[12] Anna Schmidt and Michael Wiegand," A survey on hate speech detection using natural language processing. ",2017.