**A Project report on**

# DETECTION AND REPORTING OF CYBERBULLYING ON SOCIAL MEDIA

***in partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

***Submitted by***


| | |
|---|---|
| **M. HARSHA APOORV** | **19B91A05E7** |
| **P. CHANDRA SHEKHAR** | **19B91A05J2** |
| **M. BHARGAV SAI VARMA** | **19B91A05F5** |
| **P. VISHNU VARDHAN** | **19B91A05I7** |
| **N. SAI VENKATA SATYA** | **19B91A05H4** |


***Under the esteemed Guidance of***

**K HARI KRISHNA**

***Assistant Professor***





**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**S.R.K.R ENGINEERING COLLEGE(A)**

ChinaAmiram, Bhimavaram, West Godavari Dist.,A.P.


[2022-2023]

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING S.R.K.R ENGINEERING COLLEGE(A)**

ChinaAmiram, Bhimavaram, West Godavari Dist.,A.P.

[2022-2023]

## BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**DETECTION AND REPORTING OF CYBERBULLYING ON SOCIAL MEDIA**" is the bonafide work of "**M. HARSHA APOORV(19B91A05E7), P. CHANDRA SHEKHAR(19B91A05J2), M. BHARGAV SAI VARMA (19B91A05F5), P. VISHNU VARDHAN (19B91A05I7), N. SAI VENKATA SATYA (19B91A05H4)**" who carried out the project work under my supervision in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

**SUPERVISOR**

K. HARI KRISHNA

Assistant Professor

**HEAD OF THE DEPARTMENT**

Dr. V. CHANDRA SEKHAR

Professor & HOD

# SELF DECLARATION

We hereby declare that the project work entitled "**DETECTION AND REPORTING OF CYBERBULLYING ON SOCIAL MEDIA**" is a genuine work carried out by us in B.Tech., (Computer Science and Engineering) at SRKR Engineering College(A), Bhimavaram and has not been submitted either in part or full for the award of any other degree or diploma in any other institute or University.


**M HARSHA APOORV 19B91A05E7**

**P CHANDRA SHEKHAR 19B91A05J2**

**M BHARGAV SAI VARMA 19B91A05F5**

**P VISHNU VARDHAN 19B91A05I7**

**N SAI VENKATA SATYA 19B91A05H4**

# ABSTRACT

Cyberbullying is a major problem in recent times. As the Internet is easy to use and available to everyone for free. And with access to the internet, everyone has access to social media sites. Not all social media users follow rules. This is the major issue that leads to **Cyberbullying.** Cyberbullying affects teenagers and adults. It also leads to some of the major mishappenings like suicide and depression. The main affected category is teenagers. Cyberbullying is also a cyber-crime that goes unnoticed, but has a huge impact on most people's lives. This brings a huge need in regulation of content on social media platforms. To solve this problem, we are building a machine learning model along with Natural Language Processing to detect and report the occurrence of Cyberbullying in social media comments.

# TABLE OF CONTENTS


| | | |
|---|---|---|
| | **ABSTRACT** | **iii** |
| | **LIST OF TABLES** | **v** |
| | **LIST OF FIGURES** | **vi** |
| 1. | **INTRODUCTION** | **1** |
| 2. | **LITERATURE SURVEY** | **3** |
| 3. | **PROBLEM STATEMENT** | **6** |
| 4. | **SYSTEM ANALYSIS** | **7** |
| 5. | **METHODOLOGY** | **8** |
| 6. | **IMPLEMENTATION** | **17** |
| 7. | **RESULT ANALYSIS** | **23** |
| 8. | **CONCLUSION** | **27** |
| 9. | **REFERENCES** | **28** |
| | **APPENDIX – A:SAMPLE CODE** | **32** |
| | **APPENDIX – B:UML DESIGN** | **37** |

# LIST OF TABLES


| TABLE. NO | NAME OF THE TABLE | PAGE NO |
|---|---|---|
| 6.1 | Detailed Description of Dataset | 17 |
| 6.2 | Train/Test split of the data | 21 |
| 7.1 | Comparison of Evaluation metrics of the models | 26 |

# LIST OF FIGURES


| Fig. NO | FIGURE | PAGE.NO |
|---|---|---|
| 5.1 | System Architecture | 8 |
| 5.2 | Random Forest Classifier | 13 |
| 6.1 | Dataset | 18 |
| 6.2 | Visualization of Instances | 18 |
| 6.3 | Data after Stopwords filtering and without punctuations. | 19 |
| 6.4 | Tokenized Data | 19 |
| 6.5 | Data without Digits | 20 |
| 6.6 | Top 25 words with TF-IDF scores | 20 |
| 7.1 | ROC Curve of Gaussian Naive Bayes | 23 |
| 7.2 | ROC Curve of Logistic Regression | 24 |
| 7.3 | ROC Curve of Decision Tree | 25 |
| 7.4 | ROC Curve of Random Forest | 25 |
| 7.5 | 2-class Precision-Recall curve for Random Forest Classifier | 26 |
| B.1 | Use Case Diagram | 38 |
| B.2 | Activity Diagram | 39 |
| B.3 | Sequence Diagram | 40 |
| B.4 | State Diagram | 41 |

# CHAPTER 1
# INTRODUCTION

More than ever, technology has merged completely into our daily lives as the internet has developed. Social media is popular right now. But just like with everything else, miscreants will emerge occasionally late and occasionally early, but they will exist anyway. Cyberbullying is a widespread problem today. Social networking websites are great resources for interpersonal communication. Although social networking has gained popularity over the years, most people still use it in unethical and immoral ways. This frequently occurs amongst teenagers or even young adults. Bullying each other online is one of their harmful behaviors. Online, it is difficult to tell whether someone is expressing anything only for amusement or whether he may have other motives. They will frequently laugh it off with only a joke or the advice "don't take it so seriously." The use of technology to target, harass, threaten, or humiliate another person is known as cyberbullying. This online conflict frequently turns into threats against some people in real life. Suicide has been used by certain people. At the beginning, these actions must be stopped. Any measures might be made to prevent this, for instance, if a person's tweet or post is deemed inappropriate, his or her account might be closed or stopped for a specific time. Another issue related to the misuse of social media is the spread of false information or fake news.

Social media platforms have become powerful tools for spreading information, both accurate and inaccurate, to a large number of people. Unfortunately, this has also led to the spread of false information, which can be harmful and misleading. Fake news can cause panic, harm someone's reputation, or even influence public opinion in a negative way. Social media platforms have a responsibility to curb the spread of fake news and misinformation by fact-checking information and removing content that violates their policies. Lastly, the excessive use of social media can have negative effects on mental health. The constant use of social media can lead to feelings of anxiety, depression, and low self-esteem, especially among young people. It can also lead to addiction, as people become increasingly dependent on the instant gratification that social media provides. To avoid these negative consequences, it is important to limit social media use and prioritize real-life interactions and activities. Moreover, social media companies should implement features to promote healthy usage habits and provide resources for mental health support.

Cyberbullying is a serious issue that affects people of all ages and backgrounds. It can take many forms, including harassment, threats, and even blackmail. The anonymity of the internet makes it easy for bullies to target their victims without fear of consequences, and the effects can be devastating. Victims of cyberbullying often experience anxiety, depression, and other mental health issues. They may feel isolated and powerless, and may even consider suicide as a way out. It is important for parents, educators, and community leaders to be aware of the signs of cyberbullying and to take action to

prevent it. One effective way to prevent cyberbullying is to educate young people about digital citizenship and responsible online behavior. This includes teaching them how to use social media safely and responsibly, and how to recognize and report cyberbullying. Social media companies also have a role to play in preventing cyberbullying. They can implement policies and features to discourage abusive behavior, such as blocking or reporting tools, and they can work with law enforcement to identify and prosecute cyberbullies. In addition to cyberbullying, social media can also contribute to a culture of harassment and discrimination. Women, people of color, and members of the LGBTQ+ community are often the targets of online hate speech and abuse.

Social media companies can do more to address these issues by enforcing their policies against hate speech and working to promote a more inclusive online community. Overall, it is important to recognize the potential dangers of social media and to take steps to use it responsibly. This includes being aware of the risks of cyberbullying and other forms of online harassment, limiting social media use to avoid negative effects on mental health, and promoting healthy online habits and behaviors. By working together, we can create a safer and more positive online environment for everyone.

# CHAPTER 2
# LITERATURE SURVEY

**Kelly Reynolds et al.** proposed a model by using the data that was collected from the web-site Formspring.me, a question-and-answer formatted website that contains a high percentage of bullying content. The data was labeled using a web service, Amazon's Mechanical Turk and used in conjunction with machine learning techniques provided by the Weka tool kit, to train a computer to recognize bullying content. Both a C4.5 decision tree learner and an instance-based learner were able to identify the true positives with 78.5% accuracy. [1]

**Rahul Ramesh Dalvi et al.** proposed a Machine learning model using Two classifiers i.e., SVM and Naïve Bayes are used for training and testing the social media bullying content. Both Naive Bayes and SVM (Support Vector Machine) were able to detect the true positives with 71.25% and 52.70% accuracy respectively. [2]

**Aaminah Ali and Adeel M. Syed,** aims to propose an approach to detect cyberbullying along with the element of sarcasm included in it. Sarcasm is a way of insulting someone and has adverse effects on the victim. In this study, cyberbullying is detected using machine learning algorithms. First, data was collected that is publicly available on the Internet. After extracting features preprocessing is done which includes removal of special characters, substituting multiple spaces etc. The text was first converted into the numerical form using label encoder. After that dataset was divided into 80% training set and 20% test set. Then classification algorithms SVM, naïve Bayes, Random Forest and ensemble approach were used. Results were presented in a tabular format which depicted SVM and the second one was Logistic Regression with 78% accuracy, then Random Forest with 76.7% while Naïve Bayes performed classification with 76% accuracy. [3]

**Elif Varol Altay et al.** proposed the use of natural language processing techniques and machine learning methods namely, Bayesian logistic regression, random forest algorithm, multilayer sensor, J48 algorithm and support vector machines to determine cyber bullying. The success of these algorithms with different metrics within different experiments have been compared for the first time to the real data. [4]

**Amgad Muneer and Suliman Mohamed Fati** attempted to explore this issue by compiling a global dataset of 37,373 unique tweets from Twitter. Moreover, seven machine learning classifiers were used, namely, Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), Random Forest (RF), AdaBoost (ADB), Naive Bayes (NB), and Support Vector Machine (SVM). Each of these algorithms was evaluated using accuracy, precision, recall, and F1 score as the performance metrics to determine the classifiers' recognition rates applied to the global dataset. The experimental results show the superiority of LR, which achieved a median accuracy of around 90.57%. [5]

**Md Manowarul Islam et al.** investigated the automated detection of social media posts including cyberbullying by taking into account two aspects, BoW and TF-IDF. Bag-of-Words (BoW) and term frequency-inverse document frequency (TFIDF) are two distinctive features that are used to analyze the accuracy of four different machine learning techniques. SVM for both BoW and TF-IDF and four machine learning techniques are employed to detect bullying material. In the future, they intended to create a system that uses deep learning algorithms to automatically identify and categorize cyberbullying in Bengali literature. [6]

**Neelakandan S et al.** have provided instructions on how to use a novel FSSDL-CBDC technique for identifying and categorizing cyberbullying in social media. Phases of the proposed FSSDL - CBDC technique include preprocessing, feature selection, and classification, among others. Additionally, the overall classification results are greatly improved by developing the BCO-FSS approach to select the ideal collection of features from the preprocessed data. The feature -reduced subset is delivered to and classified by the SSA-DBN model at the same time as the other models. The use of the SSA to adjust the DBN model's hyperparameter led to better results when compared to the traditional DBN model. On a benchmark dataset, several simulations were run to evaluate the improved detection performance. The simulation results showed that the FSSDL-CBDC technique performed much better in classification than the other state-of-the-art approaches when compared to them. In the future, the FSSDL-CBDC technique's performance might be improved by adding approaches for outlier identification and feature reduction to the algorithm. [7]

**Lu Cheng1 and Ahmadreza Mosallanezhad** used a reinforcement learning technique. They suggested a context-aware and model-independent debiasing strategy that does not require any additional resources or annotations beyond a predefined list of sensitive triggers frequently used to identify cases of cyberbullying. Empirical analyses demonstrate that the proposed technique can simultaneously improve detection performance and lessen the effects of unexpected biases. Empirical

evaluations demonstrated that their approach can mitigate unintended bias in the data without impairing a model's prediction accuracy. [8]

**John Hani and Mohamed Nashaat** proposed a supervised machine learning approach for detecting and preventing cyberbullying. They used several classifiers to train the model and recognize bullying actions. Their proposed approach on cyberbullying dataset shows that the Neural network performs better and achieved accuracy of 92.8% and SVM achieved accuracy of 90.3%. And also Neural network outperformed other classifiers on the same dataset. [9]

**Trisna Febrian and Arif Budiarto** both have proposed a model, during the 2019 Indonesia elections, so much hate speech is circulated in social media through Twitter. In order to recognise hate speech this paper presented a dataset in which LDA algorithm is used to extract the tweets related to debates and pretrained sentiment analysis is used to separate positive and negative tweets. [10]

**Rohit Pawar et al.** proposed A distributed cyberbullying detection system that will detect bullying messages and drop them before they are sent to the intended receiver. The prototype has been created using the principles of NLP, Machine Learning and Distributed Systems. [11]

**Vijay Banerjee et al.** suggested the uses of a deep learning model for cyberbullying detection. It uses Convolutional Neural Network where the dataset is given to the input layer, the embedding layer converts them into vector representation for understandable by the machine.Convolutional layer makes use of Rectified Linear Unit. For feature reduction, a max pooling layer is used. Dropout layer is used after the embedding layer and before the output layer. Value of the dropout layer is 0.5. It is then given to the fully connected layer or the dense layer which is the output layer and the last layer for classification purposes. The accuracy scored in this model is 93.97%. The training process may take a lot of time if the system does not contain a better GPU. [12]

**Mohammad Ali, Mohammad Rashid et al.** considered the challenges like human data characteristics, culture effect and language dynamics, they have developed a model by using different ML algorithms like SVM, Random Forest, NB algorithm, Decision Tree, K-nearest neighbor, logistic regression classification. [13]

# CHAPTER 3
# PROBLEM STATEMENT

With increase in the usage of the Internet, there has been an exponential increase in the use of online platforms, including social media (Facebook, Twitter, Koo, Tinder). Irrespective of the age, gender, race and ethnicity. Along with it Cyberbullying is also increased which goes unnoticed and has a huge impact on people's lives. Efforts to address cyberbullying are crucial in today's digital age. It's important to recognize that cyberbullying can have serious consequences, including depression, anxiety, self-harm, and even suicide. Developing a model that can detect and classify bullying comments online can be a significant step towards creating a safer online environment. To address this issue we are building a model which observes the comments and classifies whether it’s bullying or not. Based on the result, it is reported to the higher officials.

# CHAPTER 4
# SYSTEM ANALYSIS

## 4.1 EXISTING SYSTEM

In Existing system, they used various machine learning algorithms and approaches to detect language patterns of the bullies and hence generated a model to automatically detect cyberbullying actions. They proposed a supervised machine learning approach for detecting and preventing cyberbullying.
The main drawback of this existing system is that after detecting cyberbullying there are no measures taken to report the user or person who is doing that particular cybercrime.

## 4.2 PROPOSED SYSTEM

The proposed application is a model that can detect the occurrence of real-time cyberbullying by using Machine learning approaches with the help of Twitter API and notify the official authorities of Cyber Security. We are using Natural Language Processing, ensemble learning techniques along with traditional supervised machine learning algorithms. The main focus of our model is to not only detect but also take action on the users who committed the cyber-crime.

### 4.2.1 ADVANTAGES OF PROPOSED SYSTEM

- **Real-time detection:** The proposed system enables the real-time detection of cyberbullying incidents as they occur on Twitter. This allows for prompt intervention and resolution of the situation.
- **Improved accuracy:** The use of natural language processing (NLP), ensemble learning techniques, and supervised machine learning algorithms can significantly improve the accuracy of cyberbullying detection.
- **Automatic notification:** The proposed system automatically notifies the official authorities of cyber security, allowing for timely and efficient action to be taken against the perpetrators.
- **Reduced human bias:** By using machine learning approaches, the proposed system can reduce human bias and subjectivity in the detection of cyberbullying incidents.
- **Scalability:** The proposed system can be easily scaled to detect cyberbullying incidents across multiple social media platforms, not just Twitter.

# CHAPTER 5
# METHODOLOGY

## 5.1 SYSTEM ARCHITECTURE

The following is the System Architecture for Detection and Reporting of Cyberbullying.

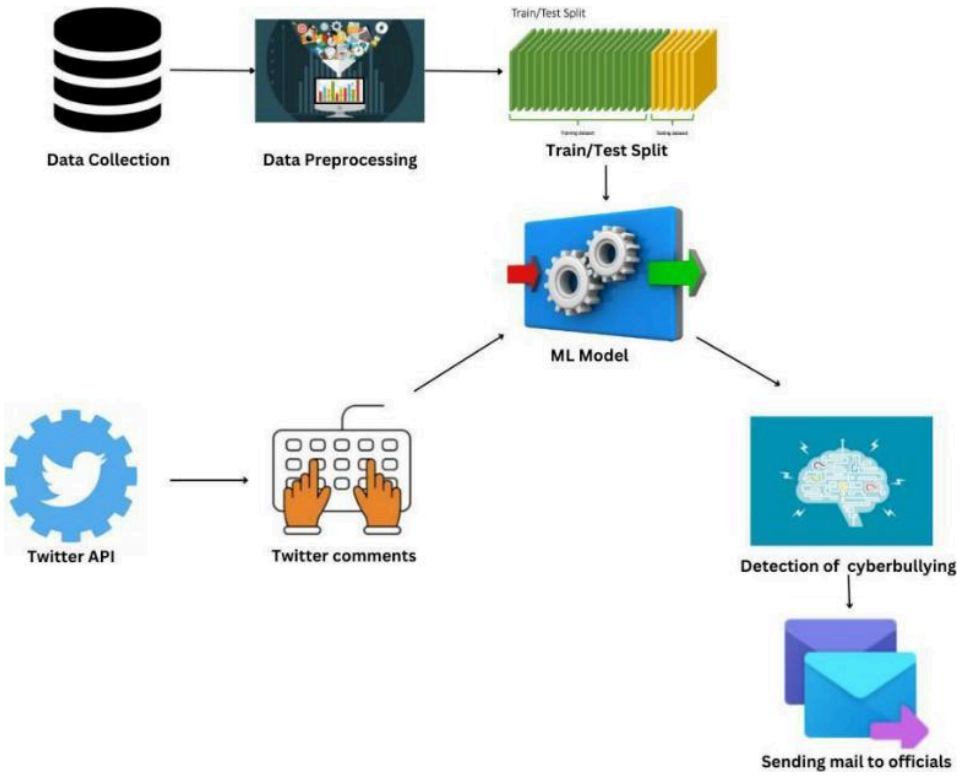



**Fig. 5.1 System Architecture**

**System architecture Description:**

1. Data Collection: This step involves collection of data from Dataturks' Tweet Dataset for Cybertroll Detection obtained from Kaggle for reaching the final results. Because of the seriousness of the issue we aim to resolve, it was crucial to choose a dataset that was complete, reliable, relevant, and to the point. While we considered many other datasets as well, many of them either had missing attributes, were too low in quality, or were found to have irrelevant data after manual inspection.
2. Data Preprocessing: The preprocessing steps were done as follows using the nltk library along with regex:
   i. Word Tokenization
   ii. Stop words filtering

iii. Removing punctuation

iv. Stemming

v. Digit removal

vi. Feature Extraction

vii. Attribute evaluation

3. Train/Test Split: The dataset of 20001 is split into Training data and Test data in the ratio 80:20, where 80% of the data is Training Data and 20% of the data is Test data.
4. Build the Model: Model is built using the training data and the machine learning algorithms. The algorithms used are Gaussian Naive Bayes, Logistic Regression, Decision Tree classifier, Random Forest Classifier.
5. Extracting realtime comments: Using twitter API, real time comments are extracted and given to the model for detection of cyberbullying.
6. Notifying Officials: If the model detects cyberbullying, then the comment along the account details are sent to the Cyber Security officials to take action regarding the issue.

## 5.2 PROPOSED METHODOLOGY

Cyberbullying is a serious problem that affects many people, particularly children and adolescents. It can have a negative impact on mental health, social relationships, and academic performance. Detecting cyberbullying is a challenging task, as it involves analyzing large volumes of text data in various forms, including social media posts, emails, and instant messages.

Machine learning can be a useful tool for detecting cyberbullying, as it can learn patterns and characteristics of cyberbullying from a large dataset and apply them to new data. Here is the proposed methodology for detection and reporting of cyberbullying using machine learning:

1. Acquisition of Dataset.
2. Data pre-processing and feature extraction.
3. Training a model.
4. Evaluation of the model.
5. Applying the model in a real-world scenario.
6. Sending automated mail to the officials.

### 5.2.1 ACQUISITION OF DATASET

We have used Dataturks' Tweet Dataset for Cybertroll Detection obtained from Kaggle for reaching the final results. Because of the seriousness of the issue we aim to resolve, it was crucial to choose a dataset that was complete, reliable, relevant, and to the point. While we considered many other datasets

as well, many of them either had missing attributes, were too low in quality, or were found to have irrelevant data after manual inspection.

Here is the Detailed Description of the dataset:

1. It is a partially manually labeled dataset.
2. Total Instances: 20001.

The dataset has 2 attributes- tweet and label [0 corresponds to No while 1 corresponds to Yes]

### 5.2.2 DATA PREPROCESSING AND FEATURE EXTRACTION

The data needs to be preprocessed. We used the nltk library of Python to preprocess the dataset. There are different steps to preprocess the data

1. Word Tokenization: A Token is a single entity that is building blocks for a sentence or paragraph. Word Tokenization converts our text to separate words in a list.
2. Stop words filtering is done using nltk.corpus.stopwords.words('english') to fetch a list of stopwords in the English dictionary, after which they are removed. Stop words are words such as "the", "a", "an", "in", which are not significant and do not affect the meaning of the data to be interpreted.
3. To remove punctuation, we save only the characters that are not punctuation, which can be checked by using string.punctuation.
4. Stemming: Stemming is a process of linguistic normalization, which reduces words to their root word. We stem the tokens using nltk.stem.porter.PorterStemmer to get the stemmed tokens. For example, connection, connected, connecting word reduce to a common word "connect".
5. Digit removal: We also filtered out any numeric content as it doesn't contribute to cyberbullying.
6. Now the next step was to extract features so that it can be used with ML algorithms, for which we used TF-IDF Transform using Python's sklearn library. TF-IDF is a statistical measure to evaluate the relevance of a word, which is basically calculated by multiplying the number of times that words appeared in the document by the inverse document frequency of the word. TF-IDF uses the method of diminishing the weight (importance) of words appearing in many documents in common, considering them incapable of discerning the documents, rather than simply counting the frequency of words as CountVectorizer does. The outcome matrix consists of each document (row) and each word (column) and the importance (weight) computed by tf * idf (values of the matrix). If a word has high tf-idf in a document, it has most of the times

occurred in given documents and must be absent in the other documents. So the words must be a signature word.

7. Attribute evaluation is done manually as can be seen where we have printed the top 25 words according to the calculated tf-idf score. Some Top ranked words for the dataset were: [hate, fuck, damn, suck, ass, that, lol, im, like, you, it, get, what, no, would, bitch]

### 5.2.3 TRAINING A MODEL

The entire dataset is divided into training and test data in the ratio 80:20 with 16000 instances of training data and 4001 instances of test data. To train the model we are using supervised machine learning algorithms.

A. **Gaussian Naive Bayes**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem of mathematics. In simple words, the Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called naive Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution. This extension of Naive Bayes is called Gaussian Naive Bayes. Beside the Gaussian Naive Bayes there are also the Multinomial naive Bayes and the Bernoulli naive Bayes. We picked the Gaussian Naive Bayes because it is the most popular one and one of the simplest to implement because we only need to estimate the mean and the standard deviation from the training data. The classifier was implemented using the "sklearn.naive_bayes" package.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters $\sigma_y$ and $\mu_y$ are estimated using maximum likelihood.

B. **Logistic Regression**

Regression analysis is a predictive modeling technique that analyzes the relation between the target or dependent variable and independent variable in a dataset. Regression analysis techniques get used when the target and independent variables show a linear or non-linear relationship between each other, and the target variable contains continuous values. Regression analysis involves determining the best fit line, which is a line that passes through all the data points in such a way that distance of the line from each data point is minimized. Logistic regression is one of the types of regression analysis technique, which gets used when the dependent variable is discrete. Example: 0 or 1, true or false, etc. This means the target variable can have only two values, and a sigmoid curve denotes the relation between the target variable and the independent variable, by mapping any real value to a value between 0 and 1. We chose Logistic Regression as the size of our data set was large, and it had almost equal occurrence of values to come in target variables. Moreover, there was no correlation between independent variables in the dataset. The classifier was implemented using the "sklearn.linear_model" package.

**C. Decision Tree Classifier**

A Decision Tree is constructed by asking a series of questions with respect to the dataset. Each time an answer is received, a follow-up question is asked until a conclusion about the class label of the record. The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges. It has 3 types of nodes: Root, Internal, and Leaf nodes. In a decision tree, each leaf node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. Using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG) (reduction in uncertainty towards the final decision). In an iterative process, we can then repeat this splitting procedure at each child node until the leaves are pure. This means that the samples at each leaf node all belong to the same class. The classifier was implemented using the "sklearn.tree" package.

**D. Random Forest Classifier**

Random Forest Classifier is a popular machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple decision trees to produce a more accurate and robust model. In Random Forest Classifier, a set of decision trees is constructed using a

random subset of features from the dataset. Each decision tree is built independently and uses a different subset of the features. To make a prediction, the algorithm combines the predictions made by each tree, either by taking a majority vote (for binary classification) or by averaging the predictions (for multiclass classification). As its name implies, Random Forest Classifier consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The low correlation between models is the key as they can produce ensemble predictions that are more accurate than any of the individual predictions, as the trees protect each other from their individual errors. The process of Bagging is used to diversify models as each individual tree is allowed to randomly sample from the dataset with replacement. The classifier was implemented using the "sklearn.ensemble" package.

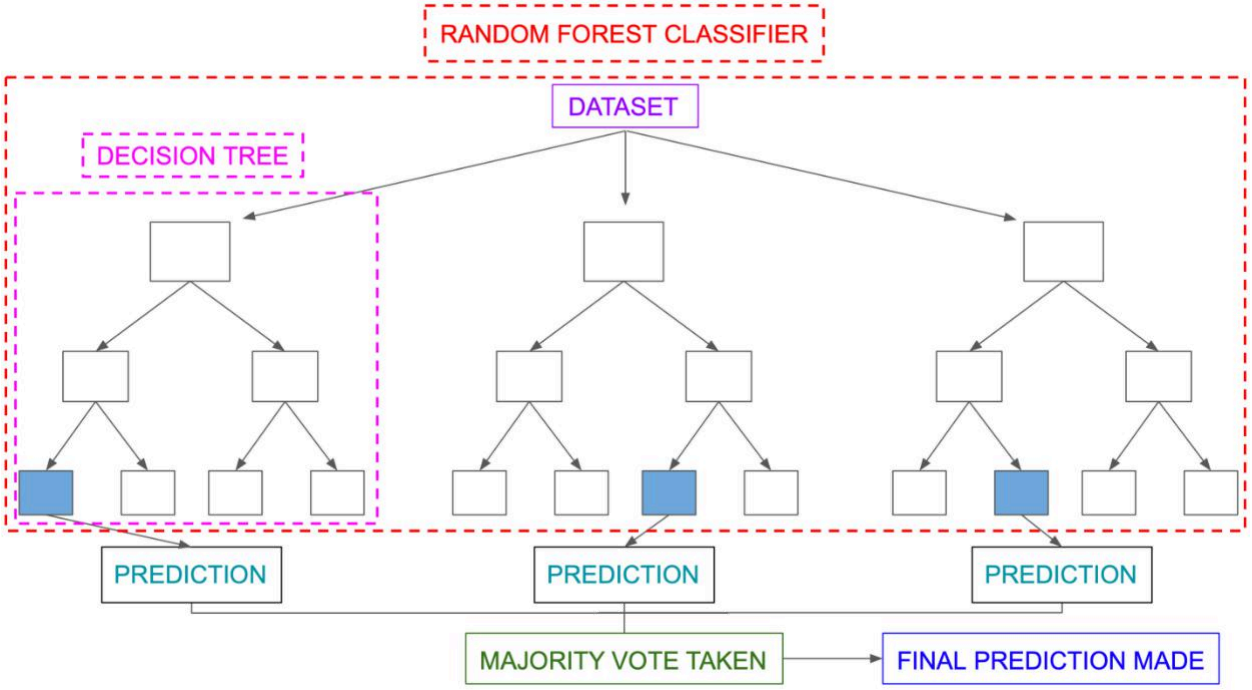



**Fig. 5.2 Random Forest Classifier**

**ALGORITHM:**

**Input**: Training set D = {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi is a vector of features and yi is the corresponding class label.

**Output**: Random Forest Classifier model f(x).

**Steps:**

1. For each tree in the forest:
    - Randomly select a subset of features from the dataset.

- Build a decision tree based on the selected features using a criterion such as information gain or Gini impurity.
- Repeat steps a and b to build a set of decision trees, each using a different subset of features.

2. For classification tasks, the final prediction is the mode of the predictions made by each tree. For regression tasks, the final prediction is the average of the predictions made by each tree.
3. The hyperparameters such as the number of trees, the maximum depth of each tree, and the minimum number of samples required to split a node can be tuned using cross-validation techniques to find the optimal values for a given dataset and task.
4. To make predictions on new data, pass the data through each tree in the forest and aggregate the predictions as described in step 2.

## 5.4 EVALUATION OF MODEL

To determine the performance of a model, we need to evaluate the model based on its results as compared to the actual results. The Metrics used for determining the performance of models are as follows:

1. Accuracy: Accuracy is a commonly used metric in machine learning and statistical classification tasks, which measures the percentage of correct predictions made by a classifier. It is defined as the ratio of the number of correct predictions to the total number of predictions made.

   Accuracy can be calculated using the following formula:

   **Accuracy = (Number of correct predictions) / (Total number of predictions)**

   Accuracy values range from 0 to 1, where a value of 1 indicates that all predictions made by the classifier are correct and a value of 0 indicates that none of the predictions are correct.

   While accuracy is a useful metric for evaluating the overall performance of a classifier, it can be misleading in cases where the class distribution is imbalanced. In such cases, a classifier that always predicts the majority class may have a high accuracy even though it is not very useful in practice. Therefore, it is often necessary to consider other metrics such as precision, recall, and F-measure in addition to accuracy.
2. F–measure: The F-measure is a widely used metric in information retrieval and machine learning for evaluating the performance of a binary classification system. It combines precision and recall into a single score, which helps to balance the tradeoff between the two

metrics. The F-measure is defined as the harmonic mean of precision and recall, where precision is the number of true positives divided by the number of true positives and false positives, and recall is the number of true positives divided by the number of true positives and false negatives. The F-measure is calculated as:

**F-measure = 2 * (precision * recall) / (precision + recall)**

The F-measure ranges from 0 to 1, where a score of 1 indicates perfect precision and recall, and a score of 0 indicates no precision or recall. A high F-measure score indicates that the classifier has a good balance between precision and recall, and is therefore effective at identifying relevant instances while minimizing false positives and false negatives.

3. Precision: Precision is a statistical measure that reflects how close a set of measurements are to each other. In other words, it is a measure of how consistent or reproducible a set of measurements are. The precision is calculated as:

**Precision = True Positive / Positive Samples**

It is also worth noting that precision is not always the most appropriate metric to use, especially when the dataset is imbalanced or when the costs of false positives and false negatives are not equal. In these cases, other metrics such as recall or F1 score may provide a more comprehensive evaluation of the model's performance.

4. Recall: In machine learning, recall is a performance metric that measures the ability of a binary classification model to correctly identify positive instances from a dataset. It is the ratio of the true positives (correctly identified positive instances) to the sum of true positives and false negatives (positive instances that were incorrectly classified as negative). Recall is particularly useful when the cost of false negatives (incorrectly predicted negative instances) is high. The recall is calculated as:

**Recall = True Positive / (True Positive + False Negative)**

5. ROC Curve: A Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classification model. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different classification thresholds. The TPR is the proportion of true positives (correctly classified positive instances) to the total number of actual positive instances in the dataset. The FPR is the proportion of false positives (incorrectly classified negative instances) to the total number of actual negative instances in the dataset. The ROC curve is useful for evaluating the trade-off between TPR and FPR, and for selecting an appropriate classification threshold based on the specific requirements of the application. An ideal ROC curve would have a TPR of 1 (indicating that all positive instances are correctly classified) and an FPR of 0 (indicating that no negative instances are incorrectly

classified as positive). The area under the ROC curve (AUC) is also commonly used as a metric for evaluating the overall performance of a binary classification model. The AUC ranges from 0 to 1, with a value of 0.5 indicating a random classifier and a value of 1 indicating a perfect classifier.

# CHAPTER 6
# IMPLEMENTATION

The proposed methodology for detection and reporting of cyberbullying involves various steps.

1. Acquisition of dataset: We collected a dataset which is available online that consists of various comments that are considered as bullying. It has a total of 20001 instances with 2 labels(0 corresponds to No whereas 1 corresponds to Yes).
2. Data pre-processing and feature extraction: There are several pre-processing steps involved
   i. Word Tokenization - Word tokenization converts the text into list of words
   ii. Stop words filtering is done using nltk.corpus.stopwords.words('english')
   iii. To remove punctuation, we considered only the letters that are not punctuation.
   iv. Digit removal, removal of numerics because those are not related to bullying.
   v. The Data is then given for feature extraction by using TF-IDF.
3. Design and implementation of a classification model: The proposed algorithms for cyberbullying detection are Gaussian Naive Bayes, Logistic Regression, Decision Tree Classifier, Adaboost Classifier and Random Forest Classifier.
4. Evaluation of model: After building the model, we have to evaluate each model based on metrics like Accuracy, F1 score, precision, recall and roc curve.
5. Applying in real world scenarios: Using twitter APIs we can obtain the real time comments to detect cyberbullying.
6. Notifying officials: Creating a script that can send automated mails to the concerned officials for taking appropriate action on the users who committed the cyber-crime.

## 6.1 DATASET

The dataset used was set in a json format. Since the fields of the dataset were relatively simple to interpret, the original set of fields in the annotation attribute was removed, and filled with the label values to simplify the next step. The number of instances for each class are mentioned in table 1.

| | |
|---|---|
| **Total Instances** | 20001 |
| **CyberBullying Instances** | 7822 |
| **Non-CyberBullying Instances** | 12179 |

**Table 6.1 Detailed Description of Dataset**

| | content | annotation |
|---|---|---|
| 0 | Get fucking real dude. | 1 |
| 1 | She is as dirty as they come and that crook ... | 1 |
| 2 | why did you fuck it up. I could do it all day... | 1 |
| 3 | Dude they dont finish enclosing the fucking s... | 1 |
| 4 | WTF are you talking about Men? No men thats n... | 1 |
| ... | ... | ... |
| 19996 | I dont. But what is complaining about it goi... | 0 |
| 19997 | Bahah yeah i&;m totally just gonna&; get pis... | 0 |
| 19998 | hahahahaha >:) im evil mwahahahahahahahahaha | 0 |
| 19999 | What&;s something unique about Ohio? :) | 0 |
| 20000 | Who is the biggest gossiper you know? | 0 |

20001 rows × 2 columns

**Fig. 6.1 Dataset**

The figure 6.1 is a labeled dataset which consists of 2 columns and 20001 instances. Where the first column is the content or the comment and the second column determines whether a particular comment is cyberbullying or not.

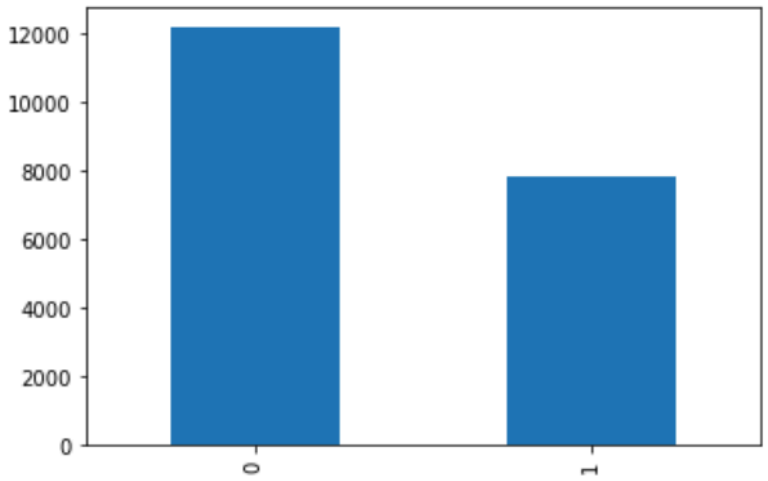



**Fig. 6.2 Visualization of Instances**

The figure 6.2 shows the visualization of instances as cyberbullying instances and non-cyberbullying instances.

## 6.2 DATA PRE-PROCESSING

The Data is then preprocessed by following various steps like word tokenization, Stopwords filtering, Removal of punctuations and Digits for making the data clean and ready for feature extraction.

As the stopwords such as “is”, “as”, “the” etc.., or the punctuations or the digits don’t have any impact on the cyberbullying they are removed which is an essential step in data preprocessing.

The Important features are extracted from the pre-processed data by using TF-IDF(Term Frequency-Inverse Document Frequency).

| | annotation | content_without_puncs |
|---|---|---|
| **0** | 1 | Get fucking real dude |
| **1** | 1 | She dirty come crook Rengel Dems fucking corru... |
| **2** | 1 | fuck up I could day too Lets hour Ping later s... |
| **3** | 1 | Dude dont finish enclosing fucking showers I h... |
| **4** | 1 | WTF talking Men No men thats menage thats gay |
| **...** | ... | ... |
| **19996** | 0 | I dont But complaining going do |
| **19997** | 0 | Bahah yeah im totally gonna get pissed talking... |
| **19998** | 0 | hahahahaha im evil mwahahahahahahahahaha |
| **19999** | 0 | Whats something unique Ohio |
| **20000** | 0 | Who biggest gossiper know |

20001 rows × 2 columns

**Fig. 6.3 Data after Stopwords filtering and without punctuations.**

The figure 6.3 shows the data after Stop words filtering and after removal of punctuations which have no impact on the detection of Cyberbullying.

| | annotation | content_tokenize |
|---|---|---|
| **0** | 1 | get fuck real dude |
| **1** | 1 | she dirti come crook rengel dem fuck corrupt ... |
| **2** | 1 | fuck up i could day too let hour ping later s... |
| **3** | 1 | dude dont finish enclos fuck shower i hate ha... |
| **4** | 1 | wtf talk men no men that menag that gay |
| **...** | ... | ... |
| **19996** | 0 | i dont but complain go do |
| **19997** | 0 | bahah yeah im total gon na get piss talk you ... |
| **19998** | 0 | hahahahaha im evil mwahahahahahahahahaha |
| **19999** | 0 | what someth uniqu ohio |
| **20000** | 0 | who biggest gossip know |

20001 rows × 2 columns

**Fig. 6.4 Tokenized Data**

The figure 6.4 shows the data after performing tokenization for which a tokenizer is used.

| | annotation | content_tokenize | content |
|---|---|---|---|
| **0** | 1 | get fuck real dude | get fuck real dude |
| **1** | 1 | she dirti come crook rengel dem fuck corrupt ... | she dirti come crook rengel dem fuck corrupt ... |
| **2** | 1 | fuck up i could day too let hour ping later s... | fuck up i could day too let hour ping later s... |
| **3** | 1 | dude dont finish enclos fuck shower i hate ha... | dude dont finish enclos fuck shower i hate ha... |
| **4** | 1 | wtf talk men no men that menag that gay | wtf talk men no men that menag that gay |
| **...** | ... | ... | ... |
| **19996** | 0 | i dont but complain go do | i dont but complain go do |
| **19997** | 0 | bahah yeah im total gon na get piss talk you ... | bahah yeah im total gon na get piss talk you ... |
| **19998** | 0 | hahahahaha im evil mwahahahahahahahahaha | hahahahaha im evil mwahahahahahahahahaha |
| **19999** | 0 | what someth uniqu ohio | what someth uniqu ohio |
| **20000** | 0 | who biggest gossip know | who biggest gossip know |

20001 rows × 3 columns

**Fig. 6.5 Data without Digits**

The figure 6.5 shows the data after digit removal, because numerical data has no impact on the data for detection of cyberbullying.

```
hate                                    Score: 533.8157298036014
fuck                                    Score: 503.76150769255435
damn                                    Score: 482.3875012051478
suck                                    Score: 407.37790877127185
ass                                     Score: 337.54089621427744
that                                    Score: 311.6250930420745
lol                                     Score: 298.0085779872157
im                                      Score: 296.0216055277791
like                                    Score: 287.8183474868775
you                                     Score: 284.7850587424088
it                                      Score: 254.75722294501585
get                                     Score: 253.19747902607998
what                                    Score: 221.43673623523864
know                                    Score: 211.53595900888456
would                                   Score: 202.5073882820925
bitch                                   Score: 193.08800391463464
ye                                      Score: 182.22364463196365
love                                    Score: 181.49014270754344
go                                      Score: 180.2588319545915
haha                                    Score: 179.29466045019018
think                                   Score: 178.9039058038677
one                                     Score: 174.16019276608517
do                                      Score: 160.57524593088053
time                                    Score: 160.1100301847739
gay                                     Score: 159.5820454915121
peopl                                   Score: 151.04499856119287
```

**Fig. 6.6 Top 25 words with TF-IDF scores**

The figure 6.6 shows the top 25 words along with their TF-IDF scores in descending order.

## 6.3 TRAINING THE MODELS

The preprocessed data is then split into training data and test data with 80:20 ratio, where training data is used to train the model whereas the test data is used to test the working of a model and evaluate the model.

| | Test | Training |
|---|---|---|
| **Total Instances** | 4001 | 16000 |
| **Cyberbullying Instances** | 2429 | 9750 |
| **Non Cyberbullying Instances** | 1572 | 6250 |

**Table 6.2 Train/Test split of the data**

This step can be done by using train_test_split().

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

The training data is used to build the model. The algorithms used to train the dataset are Gaussian Naive Bayes, Logistic Regression, Decision tree classifier, Random Forest Classifier. The training of the models is done as shown below.

1.Gaussian Naive Bayes

```
gnb = GaussianNB()
gnbmodel = gnb.fit(X_over, y_over)
y_pred = gnbmodel.predict(X_test)
```

2.Logistic Regression

```
lgr = LogisticRegression()
lgr.fit(X_over, y_over)
y_pred = lgr.predict(X_test)
```

3.Decision Tree Classifier

```
dtc = DecisionTreeClassifier()
dtc.fit(X_over, y_over)
y_pred = dtc.predict(X_test)
```

4.Random Forest Classifier

```
rfc = RandomForestClassifier(verbose=True) #uses randomized decision trees
rfcmodel = rfc.fit(X_over, y_over)
y_pred = rfc.predict(X_test)
```

## 6.4 MODEL EVALUATION

Model evaluation is used to understand the performance of a model. Model evaluation is the process of assessing the performance of a machine learning model on a particular task. It involves using various metrics to measure the effectiveness of the model, such as accuracy, precision, recall, F1 score, and others. The purpose of model evaluation is to determine how well the model is performing on the task it was trained to do. This is important because a model that performs poorly on the task may not be useful or may even be harmful if it is used to make important decisions. Model evaluation typically involves splitting the dataset into a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the model's performance on unseen data. This helps to ensure that the model is not simply memorizing the training data, but is instead learning to generalize to new data.

```
def getStatsFromModel(model):
  print(met.classification_report(y_test, y_pred))
  #disp = met.plot_precision_recall_curve(model, X_test, y_test)
  #disp.ax_.set_title('2-class Precision-Recall curve: '
                     #'AP={0:0.2f}')

  logit_roc_auc = met.roc_auc_score(y_test, model.predict(X_test))
  fpr, tpr, thresholds = met.roc_curve(y_test, model.predict_proba(X_test)[:,1])
  plt.figure()
  plt.plot(fpr, tpr, label='(area = %0.2f)' % logit_roc_auc)
  plt.plot([0, 1], [0, 1],'r--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic')
  plt.legend(loc="lower right")
  plt.savefig('Log_ROC')
  plt.show()
```

# CHAPTER 7
# RESULT ANALYSIS

The models used for training are Gaussian Naive Bayes, Logistic Regression, Decision Tree classifier and Random Forest Classifier. After training and testing the models, ROC Curve and Accuracy are used to evaluate the models. The result analysis of the models is as follows:

- Gaussian Naive Bayes is not suitable for detection of Cyberbullying.
- Logistic Regression and Decision Tree classifier are almost similar because of their similar ROC Curve and similar accuracy.
- Ensemble learning models perform better than the normal supervised machine learning models.
- Random Forest classifier delivers the best performance with high accuracy compared to other machine learning models.

The ROC curves along with the confusion matrix and other evaluation metrics of different machine learning models are shown below:

1. Gaussian Naive Bayes

```
Score: 0.6163459135216196
Confusion Matrix:
 [[ 925 1504]
 [  31 1541]]
              precision    recall  f1-score   support

           0       0.97      0.38      0.55      2429
           1       0.51      0.98      0.67      1572

    accuracy                           0.62      4001
   macro avg       0.74      0.68      0.61      4001
weighted avg       0.79      0.62      0.59      4001
```

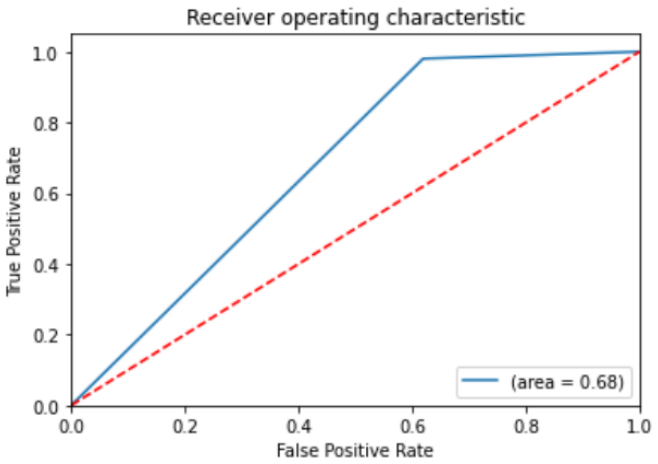



**Fig. 7.1 ROC Curve of Gaussian Naive Bayes**

The figure 7.1 shows the ROC curve of Gaussian Naive Bayes and the accuracy of Gaussian Naive Bayes is 61%.

2. Logistic Regression

```
Accuracy:  0.8002999250187454
Confusion Matrix:
 [[1901  528]
 [ 271 1301]]
              precision    recall  f1-score   support

           0       0.88      0.78      0.83      2429
           1       0.71      0.83      0.77      1572

    accuracy                           0.80      4001
   macro avg       0.79      0.81      0.80      4001
weighted avg       0.81      0.80      0.80      4001
```

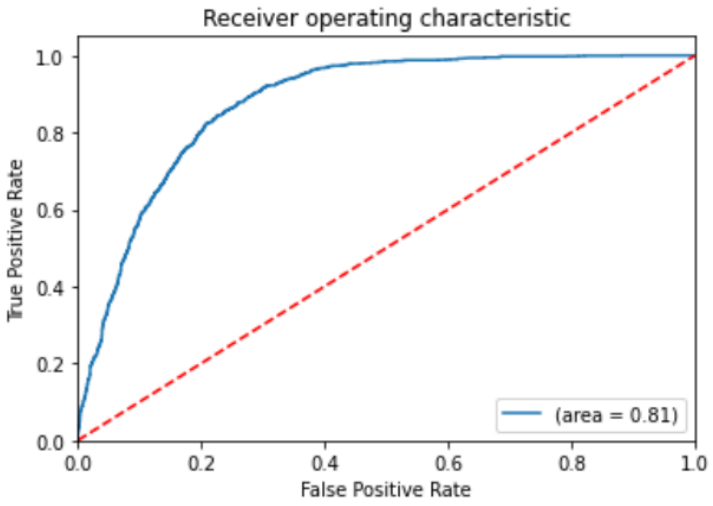



**Fig. 7.2 ROC Curve of Logistic Regression**

The figure 7.2 shows the ROC curve of Logistic Regression and the accuracy of Logistic Regression is 80%.

3. Decision Tree Classifier

```
Accuracy:  0.8505373656585854
Confusion Matrix:
 [[1881  548]
 [  50 1522]]
              precision    recall  f1-score   support

           0       0.97      0.77      0.86      2429
           1       0.74      0.97      0.84      1572

    accuracy                           0.85      4001
   macro avg       0.85      0.87      0.85      4001
weighted avg       0.88      0.85      0.85      4001
```

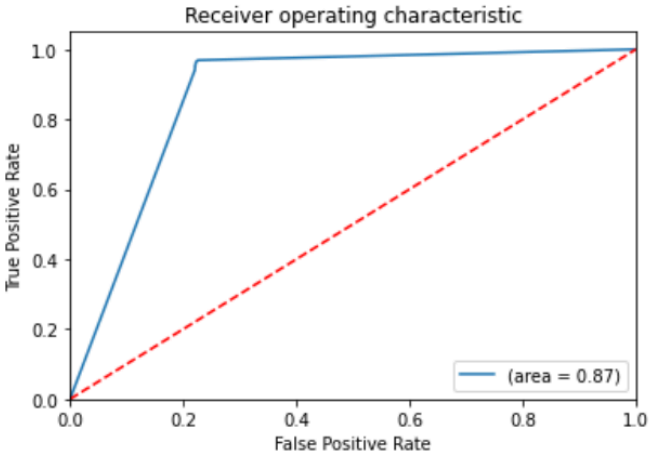



**Fig. 7.3 ROC Curve of Decision Tree Classifier**

The figure 7.3 shows the ROC curve of the Decision Tree Classifier and the accuracy of the Decision Tree Classifier is 85%.

4. Random Forest Classifier

```
Score: 0.918270432391902
Confusion Matrix:
 [[2175  254]
 [  73 1499]]
              precision    recall  f1-score   support

           0       0.97      0.90      0.93      2429
           1       0.86      0.95      0.90      1572

    accuracy                           0.92      4001
   macro avg       0.91      0.92      0.92      4001
weighted avg       0.92      0.92      0.92      4001
```




**Fig. 7.4 ROC Curve of Random Forest Classifier**

The figure 7.4 shows the ROC curve of the Random Forest Classifier and the accuracy of the Random Forest Classifier is 92%.

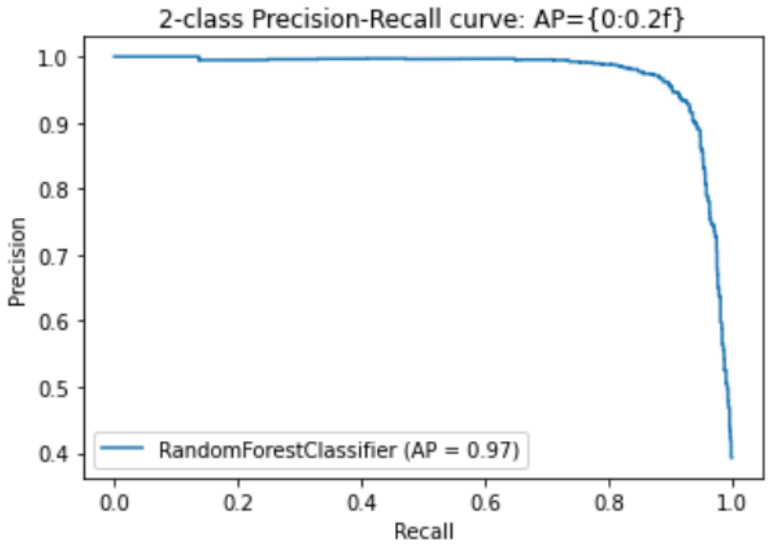



**Fig. 7.5 2-class Precision-Recall curve for Random Forest Classifier**

| | Naive Bayes | Logistic Regression | Decision Tree | Random Forest |
|---|---|---|---|---|
| Accuracy | 0.62 | 0.80 | 0.85 | 0.92 |
| Precision | 0.79 | 0.81 | 0.88 | 0.92 |
| Recall | 0.62 | 0.80 | 0.85 | 0.92 |
| F1-Score | 0.59 | 0.81 | 0.85 | 0.92 |
| ROCArea | 0.68 | 0.81 | 0.87 | 0.92 |

**Table 7.1 Comparison of Evaluation metrics of the models**

# CHAPTER 9
# CONCLUSION

Our project aims to develop a system that can detect cyberbullying based on comments. The existing systems have limitations on detection. To address this, we collected a dataset of various comments. Our proposed model achieves a higher accuracy of 92%, compared to existing models which have accuracy of 87% and 89%. Additionally, many of the previous models require large amounts of data and computational power to train, whereas our model, which uses machine learning algorithms, requires less computational power. Therefore, our model overcomes the limitations of the existing systems and successfully detects cyberbullying with a high level of accuracy. Overall, our system is an effective solution for detecting cyberbullying based on comments.

# CHAPTER 10
# REFERENCES

[1] Hani, John, Nashaat Mohamed, Mostafa Ahmed, Zeyad Emad, Eslam Amer, and Mohammed Ammar. “Social media cyberbullying detection using machine learning.” International Journal of Advanced Computer Science and Applications 10, no. 5 (2019).

[2] Reynolds, Kelly, April Kontostathis, and Lynne Edwards. “Using machine learning to detect cyberbullying.” In 10th International Conference on Machine learning and applications and workshops, vol. 2, pp. 241-244. IEEE.

[3] Talpur, Bandeh Ali, and Declan O’Sullivan. “Cyberbullying severity detection: A machine learning approach.” PloS one 15, no. 10 (2020): e0240924.

[4] Muneer, Amgad, and Suliman Mohamed Fati. “A comparative analysis of machine learning techniques for cyberbullying detection on Twitter.” Future Internet 12, no. 11 (2020): 187.

[5] Ali, Aaminah, and Adeel M. Syed. “Cyberbullying detection using machine learning.” Pakistan Journal of Engineering and Technology 3, no. 2 (2020): 45-50.

[6] Dalvi, Rahul Ramesh, Sudhanshu Baliram Chavan, and Aparna Halbe. “Detecting Twitter cyberbullying using machine learning.” In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 297-301. IEEE, 2020.

[7] Altay, Elif Varol, and Bilal Alatas. “Detection of cyberbullying in social networks using machine learning methods.” In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp. 87-91. IEEE, 2018.

[8] Islam, Manowarul & Uddin, Md Ashraf & Islam, Linta & Akhter, Arnisha & Sharmin, Selina & Acharjee, Uzzal. (2021). Cyberbullying Detection on Social Networks Using Machine Learning Approaches. 10.1109/CSDE50874.2020.9411601.

[9] Islam, Md Manowarul, Md Ashraf Uddin, Linta Islam, Arnisha Akter, Selina Sharmin, and Uzzal Kumar Acharjee. “Cyberbullying detection on social networks using machine learning approaches.” In

2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1-6. IEEE, 2020.

[10] Vijay Banerjee, JuiTelavan, Pooja Gaikwad, Pallavi Vartak.(2019). “Detection of Cyberbullying Using Deep Neural Network”. 5th International Conference on Advanced Computing & Communication Systems (ICACCS). DOI: 10.1109 ICACCS.2019.8728378.IEEE 2019

[11] M. A. Al-Garadi et al., “Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges” in IEEE Access, vol. 7, pp. 70701-70718, 2019, doi: 10.1109/ACCESS.2019.2918354.

[12] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, andYi Chang. Abusive language detection in online user content. InProceedings of the 25th international conference on the world wide web,pages 145–153. International World Wide Web Conferences SteeringCommittee, 2016.

[13] Rui Zhao, Anna Zhou, and Kezhi Mao. Automatic detection of cyber-bullying on social networks based on bullying features. In Proceedings Of the 17th international conference on distributed computing and networking, page 43. ACM, 2016.

[14] Sourabh Parime and Vaibhav Suri. Cyberbullying detection and preven-tion: Data mining and psychological perspective. In Circuit, Power andComputing Technologies (ICCPCT), 2014 International Conference on,pages 1541–1547. IEEE, 2014.

[15] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. Detecting offensive language in social media to protect adolescent online safety. In Privacy,Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom),pages 71–80. IEEE, 2012.

[16] I-Hsien Ting, Wun Sheng Liou, Dario Liberona, Shyue-Liang Wang,and Giovanny Mauricio Tarazona Bermudez. Towards the detection of cyberbullying based on social network mining techniques. In Behav-ioral, Economic, Socio-cultural Computing (BESC), 2017 InternationalConference on, pages 1–2. IEEE, 2017.

[17] Harsh Dani, Jundong Li, and Huan Liu. Sentiment informed cyberbullying detection in social media. In Joint European Conference onMachine Learning and Knowledge Discovery in Databases, pages 52–67. Springer, 2017.

[18] Maral Dadvar and Franciska De Jong. Cyberbullying detection: a step toward a safer internet yard. In Proceedings of the 21st International Conference on the World Wide Web, pages 121–126. ACM, 2012.

[19] Maral Dadvar, de FMG Jong, Roeland Ordelman, and Dolf Trieschnigg.Improved cyberbullying detection using gender information. In Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop(DIR 2012). University of Ghent, 2012.

[20] Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciskade Jong. Improving cyberbullying detection with user context. In Eu-ropean Conference on Information Retrieval, pages 693–696. Springer,2013.

[21] Maral Dadvar, Dolf Trieschnigg, and Franciska de Jong. Experts and machines against bullies: A hybrid approach to detect cyberbullies.In Canadian Conference on Artificial Intelligence, pages 275–281.Springer, 2014.

[22] Nektaria Potha and Manolis Maragoudakis. Cyberbullying detection using time series modeling. In Data Mining Workshop (ICDMW), 2014 IEEE International Conference on, pages 373–382. IEEE, 2014.

[23] Vikas S Chavan and SS Shylaja. Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In Advances in computing, communications and informatics(ICACCI), 2015 International Conference on, pages 2354–2358. IEEE,2015.

[24] Youssef Bassil and Mohammad Alwani. Post-editing error correction algorithm for speech recognition using bing spelling suggestion. arXivpreprint arXiv:1203.5255, 2012.

[25] Akiko Aizawa. An information-theoretic perspective of tf–idf measures.Information Processing & Management, 39(1):45–65, 2003.

[26] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual polarity in phrase-level sentiment analysis. In Proceedings ofHuman Language Technology Conference and Conference on EmpiricalMethods in Natural Language Processing, 2005.

[27] Steven Loria, P Keen, M Honnibal, R Yankovsky, D Karesh,E Dempsey, et al. Textblob: simplified text processing. SecondaryTextBlob: Simplified Text Processing, 2014.

[28] William B Cavnar, John M Trenkle, et al. N-gram-based text categoriza-tion. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, volume 161175. Citeseer, 1994.

[29] V. K. Singh, S. Ghosh, and C. Jose, "Toward multimodal cyberbullying detection," in Proceedings of the 2017 CHI Conference Extended Ab-stracts on Human Factors in Computing Systems, 2017, pp. 2090–2099.

[30] H. Rosa, J. P. Carvalho, P. Calado, B. Martins, R. Ribeiro, and L. Co-heur, "Using fuzzy fingerprints for cyberbullying detection in social networks," in 2018 IEEE International Conference on Fuzzy Systems(FUZZ-IEEE). IEEE, 2018, pp. 1–7.

[31] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE transactions on systems, man, and cybernetics,vol. 21, no. 3, pp. 660–674, 1991.

[32] I. Rish et al., "An empirical study of the naive bayes classifier," IJCAI2001 workshop on empirical methods in artificial intelligence, vol. 3,no. 22, pp. 41–46, 2001.

[33] M. Pal, "Random forest classifier for remote sensing classification,"International journal of remote sensing, vol. 26, no. 1, pp. 217–222,2005.

[34] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," Neural processing letters, vol. 9, no. 3, pp. 293–300, 1999.

[35] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Rapid Health data repository allocation using predictive machine learning,"Health Informatics Journal, p. 1460458220957486, 2020.

[36] M. Ashraf Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian,"Dynamically recommending repositories for health data: a machine learning model," in Proceedings of the Australasian Computer ScienceWeek Multiconference, 2020, pp. 1–10.

# APPENDIX-A: SAMPLE CODE

Step 1: Loading and understanding data and setting up input directories

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer, TfidfVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

from nltk.stem.porter import PorterStemmer
import nltk
import re, string
from nltk.corpus import stopwords

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report
from sklearn import metrics
```

Step 2: Data Visualization

```
url = 'https://drive.google.com/uc?export=download&id=12fBlhsa5GIdtme1jT3KlPPIgIdjzqhv1'
url = 'Dataset.json'
#df = pd.read_json(url, lines= True,orient='columns')
df = pd.read_json(url)
df.head
for i in range(0,len(df)):
    if df.annotation[i]['label'][0] == '1':
        df.annotation[i] = 1
    else:
        df.annotation[i] = 0
df.drop(['extras'],axis = 1,inplace = True)
df['annotation'].value_counts().sort_index().plot.bar()
```

Step 3: Data Preprocessing

```
nltk.download('stopwords')
stop = stopwords.words('english')
regex = re.compile('[%s]' % re.escape(string.punctuation))
def test_re(s):
    return regex.sub('', s)
df ['content_without_stopwords'] = df['content'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
df ['content_without_puncs'] = df['content_without_stopwords'].apply(lambda x: regex.sub('',x))
del df['content_without_stopwords']
del df['content']
df

#Stemming
porter_stemmer = PorterStemmer()
#punctuations
nltk.download('punkt')
tok_list = []
size = df.shape[0]
```

```
for i in range(size):
  word_data = df['content_without_puncs'][i]
  nltk_tokens = nltk.word_tokenize(word_data)
  final = ''
  for w in nltk_tokens:
    final = final + ' ' + porter_stemmer.stem(w)
  tok_list.append(final)

df['content_tokenize'] = tok_list
del df['content_without_puncs']
df

noNums = []
for i in range(len(df)):
  noNums.append(''.join([i for i in df['content_tokenize'][i] if not i.isdigit()]))

df['content'] = noNums
df

tfIdfVectorizer=TfidfVectorizer(use_idf=True, sublinear_tf=True)
tfIdf = tfIdfVectorizer.fit_transform(df.content.tolist())

df2 = pd.DataFrame(tfIdf[2].T.todense(), index=tfIdfVectorizer.get_feature_names_out(),
columns=["TF-IDF"]) #for second entry only(just to check if working)
df2 = df2.sort_values('TF-IDF', ascending=False)
print (df2.head(10))

def display_scores(vectorizer, tfidf_result):
    scores = zip(vectorizer.get_feature_names_out(),
                 np.asarray(tfidf_result.sum(axis=0)).ravel())
    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    i=0
    for item in sorted_scores:
        print ("{0:50} Score: {1}".format(item[0], item[1]))
        i = i+1
        if (i > 25):
```

```
        break

#top 25 words
display_scores(tfIdfVectorizer, tfIdf)

X=tfIdf.toarray()
y = np.array(df.annotation.tolist())
#Splitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

Step 4: Training and Calculating Scores

```
def getStatsFromModel(model):
  print(met.classification_report(y_test, y_pred))
  #disp = met.plot_precision_recall_curve(model, X_test, y_test)
  #disp.ax_.set_title('2-class Precision-Recall curve: '
                #'AP={0:0.2f}')

  logit_roc_auc = met.roc_auc_score(y_test, model.predict(X_test))
  fpr, tpr, thresholds = met.roc_curve(y_test, model.predict_proba(X_test)[:,1])
  plt.figure()
  plt.plot(fpr, tpr, label='(area = %0.2f)' % logit_roc_auc)
  plt.plot([0, 1], [0, 1],'r--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic')
  plt.legend(loc="lower right")
  plt.savefig('Log_ROC')
  plt.show()
```

```
#Gaussian Naive Bayes
gnb = GaussianNB()
gnbmodel = gnb.fit(X_over, y_over)
y_pred = gnbmodel.predict(X_test)
print ("Score:", gnbmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(gnb)

#Logistic Regression
lgr = LogisticRegression()
lgr.fit(X_over, y_over)
y_pred = lgr.predict(X_test)
print("Accuracy: ",met.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(lgr)

#Decision Tree Classifier
dtc = DecisionTreeClassifier()
dtc.fit(X_over, y_over)
y_pred = dtc.predict(X_test)
print("Accuracy: ",met.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(dtc)

#RandomForestClassifier
rfc = RandomForestClassifier(verbose=True) #uses randomized decision trees
rfcmodel = rfc.fit(X_over, y_over)
y_pred = rfc.predict(X_test)
print ("Score:", rfcmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(rfc)
```

# APPENDIX-B: UML DESIGN

## 7.1 UML DIAGRAMS

Unified Modeling Language is an acronym for Uml. Simple terms, UML is a new method to software modeling and documentation. It's one of the most widely used business process modeling methods. It is based on software component diagramsWe can better grasp potential defects or problems in software or business processes by employing visual representations.

The Unified Modelling Language is a general purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It includes choices and knowledge of systems that must be built. It is used to study, plan, browse, establish, maintain, and manage data about these systemsThe Unified Modelling Language is very important parts of developing object oriented 7 software and the software development process

The confusion that surrounded software development and documentation led to the creation of UML. There were various approaches to represent and record software systems in the 1990s. The need for a more unified way to visually depict those systems arose, and the UML. was created by three Rational Software software developers in 1994-1996. In 1997, it was chosen as the standard, and it has remained the standard ever since, with only minor revisions.

## GOALS:

The main goals of the uml design are listed below:

1. Ready-to-use visual modeling language should be provided for the user which allows them to create and share meaningful models.
2. Provide means for extending and specializing the key notions.
3. Be unaffected by programming languages or development processes.
4. Establish a formal foundation for comprehending the modeling language.
5. Promote the expansion of the market for 00 tools.
6. Encourage the use of higher-level development ideas like collaborations, frameworks, patterns, and components.
7. Apply methodologies.

## B.1 USE CASE DIAGRAM

A use case diagram is a form of behavioral diagram that is defined and produced from a UML.Use-case analysis.

Providing a graphical representation of a system's configuration in terms of actors, objectives (represented as use cases), and relationships between those use cases is the main motive behind it.

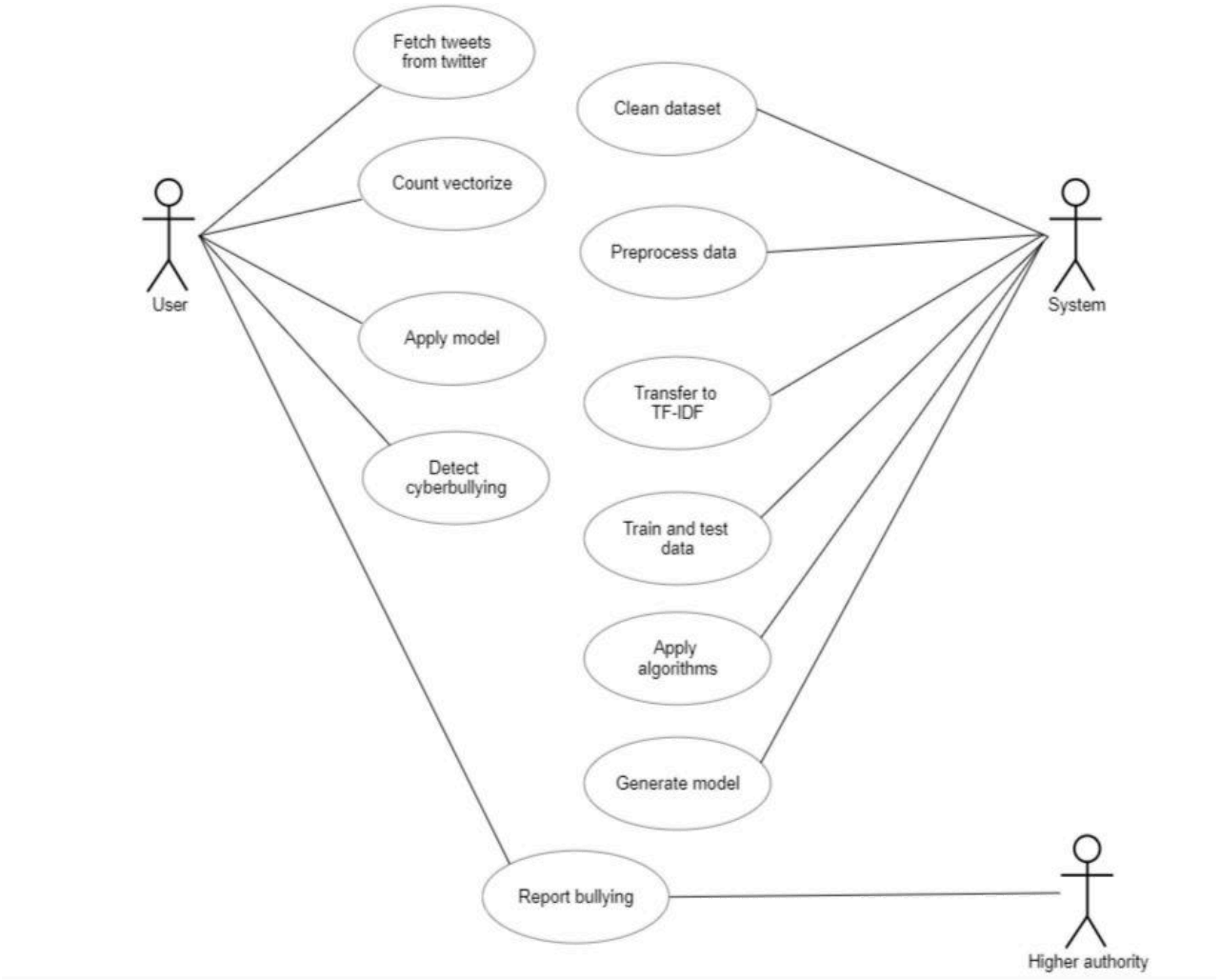



**Fig. B.1 Use Case Diagram**

## B.2 ACTIVITY DIAGRAM

An activity diagram is a type of diagram that illustrates the flow of activities and actions performed by a system or a process. It is commonly used in software engineering to model business processes, system workflows, and user interactions. Activity diagrams are composed of nodes and edges, where nodes represent activities and edges represent the flow of control between them. The activity diagram is drawn as a flowchart-like structure where the nodes are connected by edges that indicate the flow of control. The direction of the arrows represents the flow of control from one activity to another. Activity diagrams are useful in analyzing and designing complex systems or processes. They help in understanding the various steps involved in a process, identifying potential bottlenecks and areas for improvement, and optimizing the process flow.

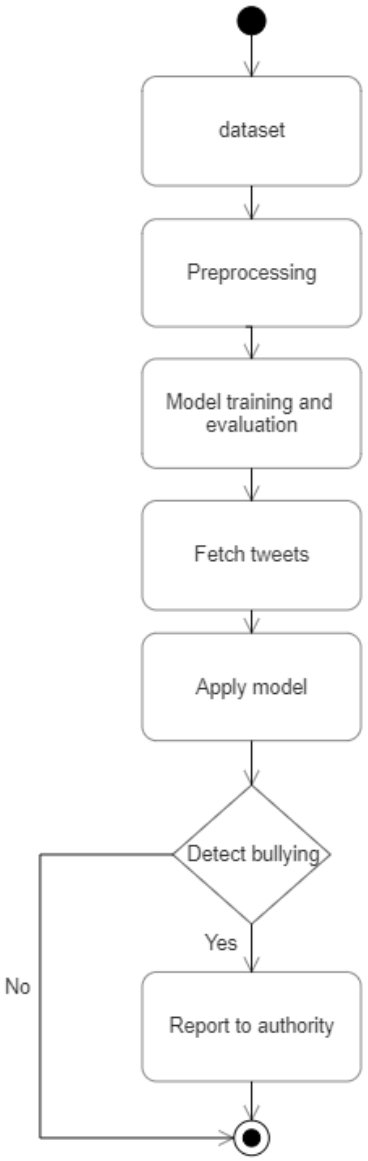



**Fig. B.2 Activity diagram**

## B.3 SEQUENCE DIAGRAM

A Sequence diagram is a graphical view of a scenario that shows object interaction in a time based sequence: what happens first, what happens next. The most important characteristic of a sequence diagram is that it is time-ordered.

The interaction between distinct objects in the system is represented by a sequence diagram. The most important characteristic of a sequence diagram is that it is time-ordered. This implies that the exact order in which the items interact is described step by step. The sequence diagram's various elements communicate with one another by passing "messages."

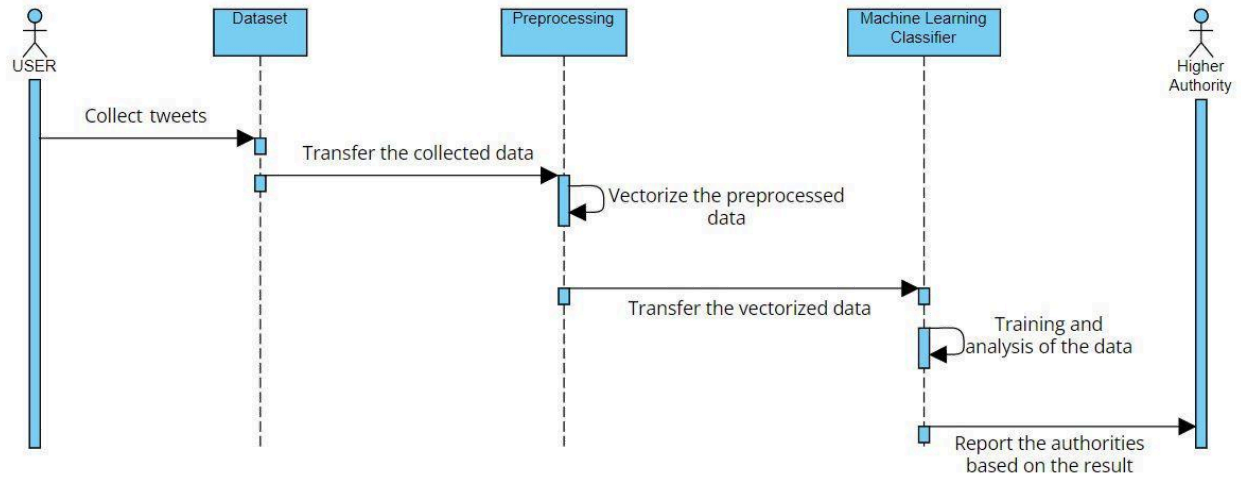



**Fig. B.3 Sequence Diagram**

## B.4 STATE DIAGRAM

A state diagram is a sort of diagram used to illustrate the behavior of systems in computer science and related subjects. State diagrams assume that the system being described has a finite number of possible outcomes; this is not always the case, but it is occasionally an acceptable assumption. State diagrams come in a variety of forms, each with its own meanings.

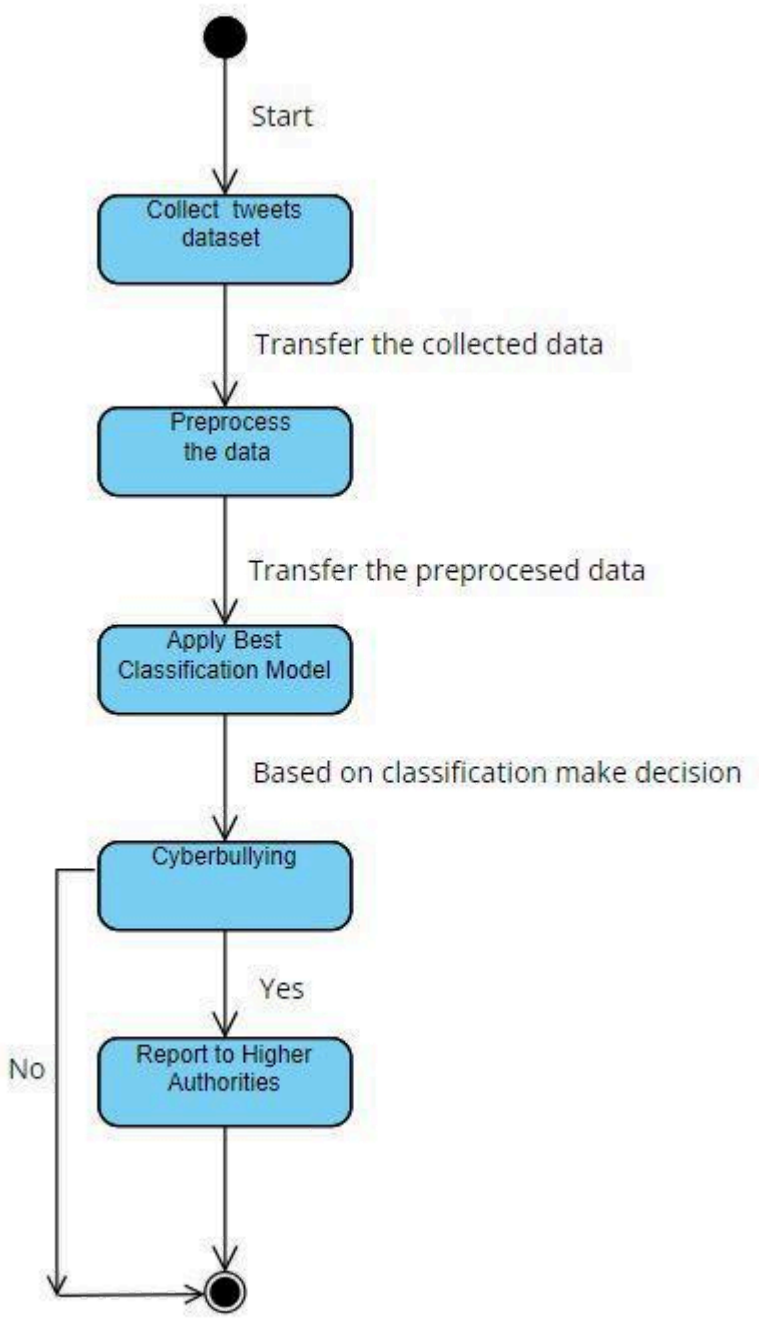



**Fig. B.4 State diagram**