CYBER BULLING DETECTION ON SOCIAL MEDIA USING MACHINE LEARNING

PROJECT REPORT

Submitted by

BAKIYA LAKSHMI A(821119104009)

NANDHINI J (821119104029)

SURUTHI S (821119104501)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KINGS COLLEGE OF ENGINEERING, PUNALKULAM

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"CYBER BULLYING DETECTION ON SOCIAL MEDIA USING MACHINE LEARNING"** is the bonafIde work of **" BAKIYA LAKSHMI . A (821119104009), NANDHINI J (821119104029) SURUTHI S (821119104501)** who carried out the project under my supervision during the year 2022 – 2023.

SIGNATURE

Dr.S.M.UMA

ASSOCIATE PROFESSOR,

HEAD OF THE DEPARTMENT

Department of CSE,

Kings College of Engineering,

Punalkulam.

SIGNATURE

MS.R.SUGANTHA LAKSHMI

ASSISTANT PROFESSOR,

SUPERVISOR

Department of CSE,

Kings College of Engineering,

Punalkulam.

Submitted for the Anna University: : Chennai, Practical Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled **"CYBER BULLYING DETECTION ON SOCIAL MEDIA USING MACHINE LEARNING"** is submitted in partial fulfillment of the requirement for the award of the degree in B.E., Anna University, Chennai, is a record of our work carried out by as during the academic year 2022–2023 under the **supervision** of **Ms.R.Sugantha Lakshmi Assistant Professor, Department of Computer Science and Engineering.** The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree or diploma, either in this or any other university.

BAKIYA LAKSHIMI.A	**NANDHINI J**	**SURUTHI S**
(821119104009)	**(821119104029)**	**(821119104501)**

I certify that the declaration made above by the candidates is true.

Ms.R.Sugantha Lakshmi,

Assistant Professor,

Department of Computer Science and Engineering,

Kings College of Engineering,

Punalkulam**.**

ACKNOWLEDGEMENT

We give all glory and honor to the almighty for his blessings and divine help which helped us to complete the project work successfully.

The project work has been undertaken and completed with direct and indirect help of many people and we would like to acknowledge the same.

A special note of thanks to **Shri. T. R. S. Muthukumaar, M.B.A., CEO, Kings College of Engineering, Punalkulam** for his valuable support. We express our deepest gratitude to our secretary **Dr.R. Rajendran, M.A., M.Phil., Ph.D.,** for his moral support.

Our grateful thanks to our Principal **Dr. J.Aruputha Vijaya Selvi, M.E., Ph.D.,** and our Vice Principal **Dr. S. Sivakumar, M.Tech., Ph.D.,** Kings College of Engineering, Punalkulam, for giving permission to do to project work successfully.

We express our warm thanks to **Dr. S.M.Uma, Head of the Department, ComputerScience and Engineering**, for allowing us to do our project.

With immense pleasure, we extend our sincere and heartfelt thanks to our internal project guide & project coordinator **Mrs.R.SUGANTHA LAKSHMI, M.Tech,** we also extend our sincere thanks to all our staff members and technical assistants of CSE department. Our deepest thanks to our parents for uploading us by providing professional education and for their prayerful support that made us to complete the project phase successfully.

ABSTRACT

Now a day's people use social media to create, share and exchanges information and ideas like Instagram, Facebook, Twitter etc. As the technology advances the cyber bullying is getting enhanced. Now a days online harassment ,defame a person with bad words in fake id is a common problem and happens often in social media that leads to several instances .In order to address this problem we are proposing a framework focused on machine learning to analysis emotional content of text and prevents sharing of harassment words. We use dataset of online conversation from Twitter, Wikipedia and kaggle to train and test the model uses Natural Language Processing technique such as sentiment analysis and topic modeling to identify patterns of abusive language and offensive content. The models accuracy is evaluate using precision, recall and F1 score and is found to be effective in detecting cyber bullying with an accuracy of 90%.The results suggest that emotion analysis can be an effective tool for detecting cyber bullying and may help identify and prevent harmful behavior online.

.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

SNO	ACRONYM	ABBREVATION
1	ML	Machine Learning
2	SVM	Support Vector Machine
3	NLP	Natural Language Processing

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Cyber bullying is a form of harassment that takes place online, through social media, messaging apps, or other digital platforms. It involves the use of technology to deliberately and repeatedly harm, humiliate, or threaten another person. Cyber bullying can take many forms, including spreading rumors or false information about someone, sharing embarrassing photos or videos without permission, sending threatening or abusive messages, and impersonating someone online. Victims of cyber bullying can experience a range of negative effects, including anxiety, depression, low self-esteem, social isolation, and even suicidal thoughts or behaviors. It can be especially damaging because it can happen 24/7 and the bullying can be shared with a wide audience. It's important to recognize and address cyber bullying as a serious issue, and to take steps to prevent it from happening in the first place. This includes educating individuals about responsible online behavior, implementing effective policies and procedures to respond to cyber bullying, and providing support and resources to those who have been affected by it.

1.1.1 CYBERBULLYING ON SOCIAL MEDIA SITES

There have been several instances of harassment and abuse on Twitter in Tamil Nadu, India. Here are a few examples of famous harassment incidents on the platform:

In 2019, a Tamil Nadu-based businessman named Kathiresan was targeted with abuse and threats on Twitter after he criticized the ruling party in the state.

Kathiresan received hundreds of hateful messages, and his phone number and address were shared online. He filed a complaint with the police, and several individuals were arrested in connection with the harassment.

Chinmayi Sripada Chinmayi Sripada is a well-known singer and voice actor who has been a vocal advocate against sexual harassment and abuse. She has also spoken out against online harassment and cyber bullying, and has herself been the target of such abuse on social media. In 2018, Sripada filed a police complaint against online trolls who had threatened and harassed her on social media platforms.

The origin of cyber bullying can be traced back to the rise of the internet and social media in the 1990s and early 2000s. As more people began using online platforms to communicate and connect with others, some individuals began using these tools to harass, intimidate, or humiliate others.

Early forms of cyber bullying included sending threatening or insulting emails, creating fake social media profiles to impersonate or mock others, and spreading rumors or lies online. As technology evolved and social media platforms became more popular, the scope and intensity of cyber bullying also grew.

Today, cyber bullying can take many forms, including sending hateful messages on social media, sharing embarrassing photos or videos without someone's consent, posting hurtful comments on public forums or chat rooms, and even using video game chat rooms to harass other players.

The root causes of cyber bullying are complex and multifaceted, and may include factors such as anonymity, social isolation, and a desire for power or control. It's important to recognize that cyber bullying is a serious issue that can have harmful and long-lasting effects on individuals and communities.

1.1.2 DETECTION MODELS FOR CYBER BULLYING

Detection models for cyber bullying using machine learning have become increasingly popular in recent years. These models use natural language processing (NLP) techniques to analyze text data from social media sites and identify instances of cyber bullying.

The first step in building a detection model is to collect a dataset of text messages or posts that contain instances of cyber bullying. This dataset is then used to train the machine learning model. The model is trained using various NLP techniques, such as feature extraction, sentiment analysis, and topic modeling, to identify patterns in the text data that are indicative of cyber bullying. Once the model is trained, it can be used to classify new text messages or posts as either cyber bullying or non-cyber bullying. This is done by feeding the text data into the model, which applies the NLP techniques it has learned during training to identify any patterns that suggest cyber bullying.

There are various types of machine learning models that can be used for cyber bullying detection, including supervised learning, unsupervised learning, and deep learning. Supervised learning models are trained on labeled data, where each example is labeled as either cyber bullying or non-cyber bullying. Unsupervised learning models, on the other hand, are trained on unlabeled data and use clustering or anomaly detection techniques to identify instances of cyber bullying. Deep learning models use neural networks to identify patterns in the text data, and are particularly effective at handling large amounts of data.

CHAPTER 2
LITERATURE SURVEY

2.1 TITLE: Cyber bullying monitoring system for Twitter.

AUTHOR: Prajakta Ingle Ramya Joshi Neha Kaulgud Aarti Suryawanshi Meghana Lokhande

YEAR: 2022

DESCRIPTION: In the recent years Twitter has emerged to be a great source for users to broadcast their daily activities, opinions and feelings via texts and images. Cyber bullying is a harassment that takes prominently happens in social networking sites where cyber bullies target vulnerable victims and it has major psychological and physical effects on the victims. Hence, the Cyber bullying Monitoring System on Twitter is a solution with an aim to identify bullying tweets real time. In our research we have developed a cyber bullying monitoring system. The study reviewed the existing literature for various machine learning algorithms and identified Light GBM as the most efficient. A model for detecting bullying tweets for real time tweets was developed. We considered various twitter and user specific features along with TF IDF embedding for the classification. A detailed report about tweets and analysis was displayed. In future work, a system to classify these tweets into various categories of bullying can be developed.

2.2 TITLE: A Comparative Analysis of Machine Learning Techniques for Cyber bullying Detection on Twitter.

AUTHOR: Amgad Muneer and Suliman Mohamed Fati

YEAR: 2022

DESCRIPTION: The advent of social media, particularly Twitter, raises many issues due to a misunderstanding regarding the concept of freedom of speech. One of these issues is cyber bullying, which is a critical global issue that affects both individual victims and societies. Many attempts have been introduced in the literature to intervene in, prevent, or mitigate cyber bullying; however, because these attempts rely on the victims' interactions, they are practical. Therefore,detection of cyberbullying without the involvement of the victims is necessary. In this study, we 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%.

2.3 TITLE: Detecting Cyberbullying and Cyber aggression in Social Media.

AUTHOR: Despoina Chatzakou , Ilias Leontiadis , Jeremy Blackburn, Emiliano De Cristofaro

YEAR: 2021

DESCRIPTION: Cyber bullying and cyber aggression are increasingly worrisome phenomena affecting people across all demographics. More than half of young social media users worldwide have been exposed to such prolonged and/or coordinated digital harassment. Victims can experience a wide range of emotions with negative consequences such as embarrassment, depression, isolation from other community members, which embed the risk to lead to even more critical consequences, such as suicide attempts. In this work, we take the first concrete steps to understand the characteristics of abusive behavior in Twitter, one of today's largest social media platforms. We analyze 1.2 million users and 2.1 million tweets, comparing users participating in discussions around seemingly normal topics like the NBA, to those more likely to be hate-related, such as the Gamer gate controversy, or the gender pay inequality at the BBC station. We also explore specific manifestations of abusive behavior, i.e., cyber bullying and cyber aggression, in one of the hate-related communities (Gamer gate).

2.4 TITLE: Non-Linguistic Features for Cyber bullying Detection on a Social Media Platform using Machine Learning.

AUTHOR: YuYi Liu , Pavol Zavarsky and Yasir Malik

YEAR: 2021

DESCRIPTION: Cyber bullying on social media platforms has been a severe problem with serious negative consequences. Therefore, a number of researches on automatic detection of cyber bullying using machine learning techniques have been conducted in recent years. While cyber bullying detection has traditionally utilized linguistic features, the cyber bullying on social media does not have only linguistic features. In this paper, a holistic multi-dimensional feature set is developed which takes into account individual-based, social network-based, episode based and linguistic content-based cyber bullying features. To test performance of the proposed multi-dimensional feature set, we designed and built cyber bullying detection models on the KNIME machine learning platform. Six different machine learning algorithms - Naïve Bayes, Decision Tree, Random Forest, Tree Ensemble,Logistic Regression, Support Vector Machines - were used in our cyberbullying detection models. Our experimental results demonstrate that applying the proposed multi-dimensional feature set (i.e. the set not limited to the linguistic features) results in an improved cyber bullying detection for all tested machine learning algorithms.

2.5 TITLE: Comparative Performance of Machine Learning Algorithms in Cyber bullying Detection: Using Turkish Language Preprocessing Techniques

AUTHOR: Emre Cihan Ates , Erkan Bostanci , Mehmet Serdar Güzel

YEAR: 2020

DESCRIPTION: With the increasing use of the internet and social media, it is obvious that cyber bullying has become a major problem. The most basic way for protection against the dangerous consequences of cyber bullying is to actively detect and control the contents containing cyber bullying. When we look at today's internet and social media statistics, it is impossible to detect cyber bullying contents only by human power. Effective cyber bullying detection methods are necessary in order to make social media a safe communication space. Current research efforts focus on using machine learning for detecting and eliminating cyber bullying. Although most of the studies have been conducted on English texts for the detection of cyber bullying, there are few studies in Turkish. Limited methods and algorithms were also used in studies conducted on the Turkish language. In addition, the scope and performance of the algorithms used to classify the texts containing cyber bullying is different, and this reveals the importance of using an appropriate algorithm.

2.6 TITLE: A Study of Machine Learning Approaches to Detect Cyberbullying

AUTHOR: Subbaraju Pericherla and E. Ilavarasan

YEAR: 2020

DESCRIPTION: Social media networks like Face book and Twitter create a great platform to share public views, opinions, feelings by text message, image, video. The public is very much interested to use these networks because of the comfortable Graphical User Interface (GUI) by a single click and taps to share content from their electric gadgets, gizmos, and mostly by their smart phones. On the other hand, some people performing cyber bullying activities like aggressive comments, abusing, trolling. Sometimes, these negative activities lead to cyber bullying victims to attempt suicide. In this paper, the authors are presenting essential approaches to recognize cyber bullying over social media using advanced machine learning and deep learning algorithms. Cyber bullying provokes most of the netizens for suicide attempts. In this connection, this research focuses on the study of various methods and approaches used to detect cyber bullying activities in social media posts through machine learning. Most of the existing techniques are not considering the sarcastic text, considers only a limited number of features in the content for detecting the cyber bullying activity.

2.7 TITLE: Cyber bullying Detection Using Machine Learning

AUTHOR: Aaminah Ali, Adeel M. Syed

YEAR: 2019

DESCRIPTION: It is an age of the Internet and electronic media, and social media platforms are one of the most frequently used communication medium nowadays. But some people use these sites for malicious purpose and among those negative aspects "Cyber bullying" is prevalent. Cyber bullying is a form of bullying done through electronic means and is used to insult or harm others. Many researchers have proposed solutions and strategies to overcome this menace, but sarcasm is one aspect of it that still needs to be touched. This study aims to highlight previous researchers and to propose an approach to detect cyber bullying along with the element of sarcasm included in it. The results proved that SVM classifier performed better than other classifiers. This particular study aimed to explore cyber bullying detection using machine learning. The previous work donein this regard was also highlighted. Cyber bullying is a vast term and has different aspects. Among those aspects, sarcasm is essential. Sarcasm is a way of insulting someone and has adverse effects on the victim. As per our observation, this aspect of bullying was not considered in the previous researches. Hence this study aimed to include that aspect as well.

2.8 TITLE: Predicting Cyber bullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges

AUTHOR: Mohammed Ali Al-Garadi, Mohammad Rashid Hussain, Nawsher Khan, Ghulam Murtaza

YEAR: 2019

DESCRIPTION: Prior to the innovation of information communication technologies (ICT), social interactions evolved within small cultural boundaries such as geo spatial locations. The recent developments of communication technologies have considerably transcended the temporal and spatial limitations of traditional communications. These social technologies have created a revolution in user-generated information, online human networks, and rich human behavior-related data. However, the misuse of social technologies such as social media (SM) platforms, has introduced a new form of aggression and violence that occurs exclusively online. A new means of demonstrating aggressive behavior in SM websites are highlighted in this paper. The motivations for the construction of prediction models to fight aggressive behavior in SM are also outlined. We comprehensively review cyber bullying prediction models and identify the main issues related to the construction of cyber bullying prediction models in SM.

2.9 TITLE: Cyber bullying Ends Here: Towards Robust Detection of Cyber bullying in Social Media

AUTHOR: Mengfan Yao Chelmis

YEAR: 2018

DESCRIPTION: The potentially detrimental effects of cyber bullying have led to the development of numerous automated, data–driven approaches, with emphasis on classification accuracy. Cyber bullying, as a form of abusive online behavior, although not well–defined, is a repetitive process, i.e., a sequence of aggressive messages sent from a bully to a victim over a period of time with the intent to harm the victim. Existing work has focused on harassment (i.e., using profanity to classify toxic comments independently) as an indicator of cyber bullying, disregarding the repetitive nature of this harassing process. However, raising a cyber bullying alert immediately after an aggressive comment is detected can lead to a high number of false positives. At the same time, two key practical challenges remain unaddressed: (i) detection timeliness, which is necessary to support victims as early as possible, and (ii) scalability to the staggering rates at which content is generated in online social networks. In this work, we introduce Concise, a novel approach for timely and accurate Cyber bullying detection on Instagram media Sessions.

2.10 TITLE: Early detection of cyber bullying on social media networks

AUTHOR: Manuel F. López-VizcaínoFrancisco J. NóvoaVictor CarneiroFidel Cacheda

YEAR: 2018

DESCRIPTION: Cyber bullying is an important issue for our society and has a major negative effect on the victims, that can be highly damaging due to the frequency and high propagation provided by Information Technologies. Therefore, the early detection of cyber bullying in social networks becomes crucial to mitigate the impact on the victims. In this article, we aim to explore different approaches that take into account the time in the detection of cyber bullying in social networks. We follow a supervised learning method with two different specific early detection models, named threshold and dual. The former follows a more simple approach, while the latter requires two machine learning models. To the best of our knowledge, this is the first attempt to investigate the early detection of cyber bullying. We propose two groups of features and two early detection methods, specifically designed for this problem.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Cyber bullying detection on social media using artificial intelligence (AI) and machine learning (ML) is an emerging field that aims to automatically identify and flag potentially harmful content in online platforms. AI and ML algorithms can be trained to detect patterns in social media data, such as the use of abusive language, threatening messages, or images that may be harmful to individuals or groups.

The process of cyber bullying detection using AI involves collecting and analyzing large amounts of social media data, identifying keywords and phrases associated with cyber bullying, and using natural language processing (NLP) techniques to understand the context of the content. The algorithms can also use behavioral analysis to detect patterns of abusive behavior over time, such as repeated instances of online harassment.

Once potentially harmful content is identified, social media platforms can take action to remove the content, warn the user who posted it, or take other measures to prevent further harm. AI-powered cyber bullying detection can help social media platforms to proactively identify and address cyber bullying, which can be difficult to monitor and control manually due to the sheer volume of social media content.Overall, the use of AI and ML for cyber bullying detection on social media holds great potential to improve the safety and well-being of individuals online. However, it is important to ensure that these technologies are used ethically and responsibly, with appropriate safeguards in place to protect user privacy and prevent the misuse of these tools.

3.1.1 DISADVANTAGES

There are many existing systems in cyber bulling detection on social media and have following issues:

ACCURACY

- By using Naive Bays algorithm got 85% accuracy.
- By using SVM algorithm got 85 to 90% accuracy.
- Accuracy depends on dataset.

ANONYMITY

- Huge peoples are in social media its encourage the bullieswithout any fear do harassment ,defame others with fake id.
- To search the victim others also be affected.

OVERFITTING

- Training dataset over close to features will affect the prediction.

LESSFITTING

- Less datasets to be trained doesn’t provide a scalability.

ALGORITHM UTILTY

- By using more algorithms increases the accuracy level.
- Natural Language Processing helps for feature extraction.

FALSE POSITIVES FALSE NEGATIVES

- It is a common problem in machine learning like any ML modelidentifying non-cyber bullying behavior as cyber bullying .

- False negatives (failing to identify cyber bullying behavior). These errors can have negative consequences, such as wrongly accusing someone of cyber bullying or failing to identify instances of cyber bullying that are occurring.

3.2 PROPOSED SYSTEM

The proposed methodology for cyber bullying detection on social media involved data-driven approach using machine learning. The process begins with collecting a diverse dataset of user interactions from social media platforms. This data is then preprocessed to remove noise and standardize the text. Relevant features, such as word frequency and sentiment analysis, are extracted from the preprocessed data. The collected data is labeled as either cyber bullying or non-cyber bullying using manual or automated techniques. Machine learning algorithms are trained on the labeled data, using the extracted features as input. The trained model is evaluated using standard metrics, and then integrated into a real-time system capable of monitoring and detecting instances of cyber bullying. Continuous learning and user feedback are utilized to refine and improve the system over time, ensuring its effectiveness in identifying .

3.2.1 PROPOSED METHOD

DATA COLLECTION

- Data's are collected from real time such as Twitter ,Wikipedia and Kaggle.
- Do the preprocessing work effectively.

Some preprocess steps are

Preprocessing is an essential step in machine learning that involves preparing and transforming the raw data into a suitable format for the learning algorithm. Preprocessing helps to improve the quality of the data, remove noise, handle missing values, and make the data compatible with the chosen machine learning

model. Here are some common preprocessing steps:

1. DATA CLEANING:

- Handling missing data: Decide on a strategy to deal with missing values, such as imputation (replacing missing values with estimated values) or removing instances with missing data.
- Handling outliers: Identify and handle outliers, which are data points significantly different from other observations, by either removing them or transforming them.

2. DATA TRANSFORMATION:

- Feature scaling: Scale numerical features to a standard range (e.g., between 0 and 1) to ensure that features with different scales do not dominate the learning process. Common scaling techniques include normalization (min-max scaling) and standardization (z-score scaling).
- Feature encoding: Convert categorical features into a numerical representation since most machine learning algorithms work with numerical data. Common techniques include one-hot encoding, label encoding, or ordinal encoding.
- Feature engineering: Create new features or transform existing features to extract more meaningful information that can improve the learning process. This could involve techniques such as polynomial features, logarithmic transformations, or interaction terms.

3. DIMENSIONALITY REDUCTION:

- Feature selection: Choose the most relevant features that contribute the most to the prediction task, removing irrelevant or redundant features. This helps to reduce overfitting.

- Feature extraction: Transform the high-dimensional data into a lower-dimensional space while preserving the most important information. Techniques like Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) are commonly used for feature extraction.

4. HANDLING IMBALANCED DATA:

- In scenarios where the classes in the target variable are imbalanced (i.e., one class has significantly more instances than the other), techniques such as oversampling the minority class or undersampling the majority class can be applied to balance the dataset.

5. SPLITTING DATA:

- Divide the dataset into training, validation, and testing sets. The training set is used to train the model, the validation set is used to fine-tune hyperparameters, and the testing set is used to evaluate the model's performance on unseen data.
- These preprocessing steps are not exhaustive, and the specific techniques used may vary depending on the nature of the data, the problem domain, and the requirements of the machine learning algorithm being used.

USING COMBINATIONS OF ALGORITHM

- By using various algorithm such as Support VectorMachine(SVM),Random forest, Logistic regression using Natural Language Processing(NLP) technique provides the accuracy level.
- Decision tree helps to make prediction accurately.

FEATURE EXTRACTION

- It analyze the raw data using patterns, relationships and relevantcharacteristics in textual message.
- The feature will be extracted by Bag of words, Term frequency, Inverse document frequency and word2vec models.

SENTIMENT ANALYSIS

- It is trained by labeled data that will classified into positivenegative and neutral categories.
- By understanding the emotions expressed in a message, it is possible to identify instances of cyber bullying.

BEHAVIORAL ANALYSIS

- It focuses on identifying patterns of behavior that are associated with cyber bullying through analyzing the content of individual messages.
- Clustering techniques can group users who exhibit similar behavior, such as posting aggressive or offensive content.

PRECISION AND RECALL

- Precision and recall are measures of a model's performance in identifying positive and negative instances.
- Precision measures the proportion of positive predictions that are correct, while recall measures the proportion of actual positive instances that are correctly identified by the model.

3.2.2 ADVANTAGES

- Easily detectable.
- Effective with Accuracies.
- To detect personal attack social Medias.

CHAPTER 4

SYSTEM ANALYSIS

4.1 HARDWARE SYSTEM CONFIGURATION

- Processor - Pentium –IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

4.2 SOFTWARE SYSTEM CONFIGURATION

- Operating System : Windows 7 or 8
- Front End : Python Idle , Flask
- Back End : My sql

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

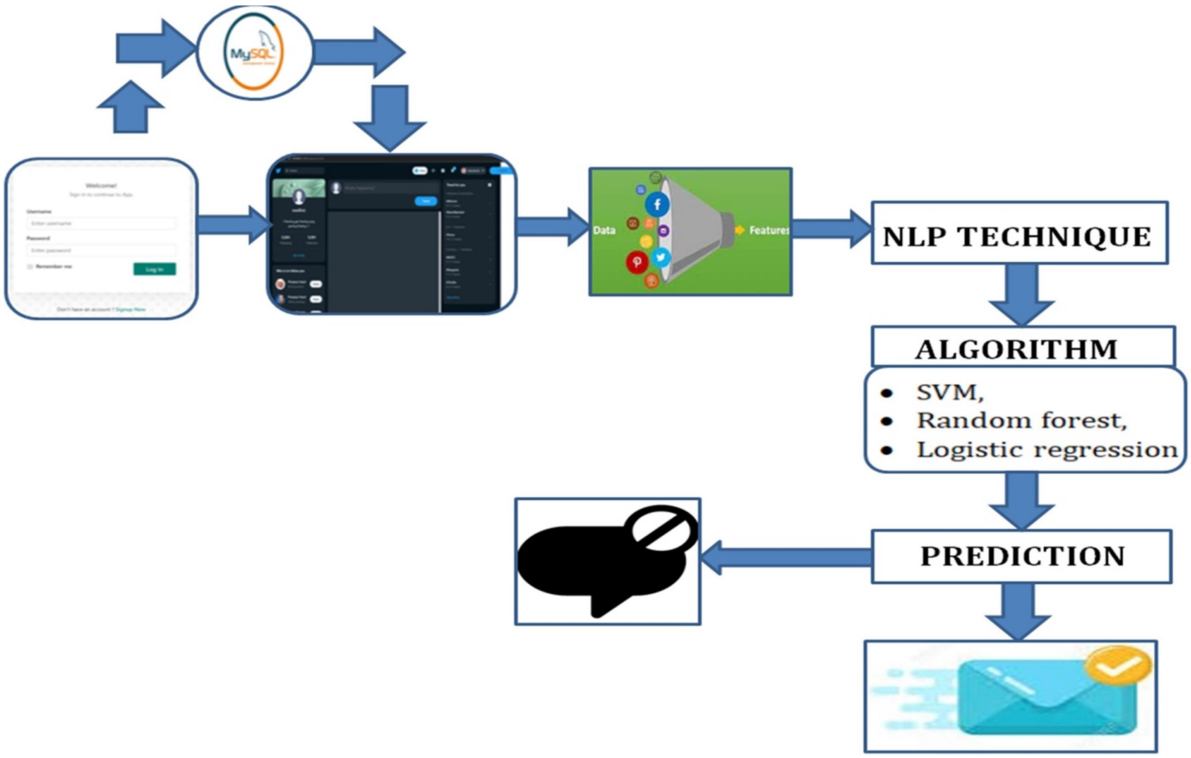


Fig No :5.1

- The architecture proposes the system work briefly such as first step is login page after it will connect to the mysql (xampp server) at the backend .
- User details will be evaluated.
- Then the commands will extracted by bag of words, word 2 vector.
- Then NLP technique. After algorithm make a prediction whether cyber bullying or not.
- If it is cyber bullying the message will be blocked or the message will be send.

5.2 UML DIAGRAM

A Use Case diagram is a visual representation of the functional requirements of a system, showcasing the interactions between actors (users or external systems) and the system itself. In the case of cyber bullying detection on social media, a Use Case diagram can illustrate the various actions and roles involved. Here's an example of how a Use Case diagram for this scenario might look.

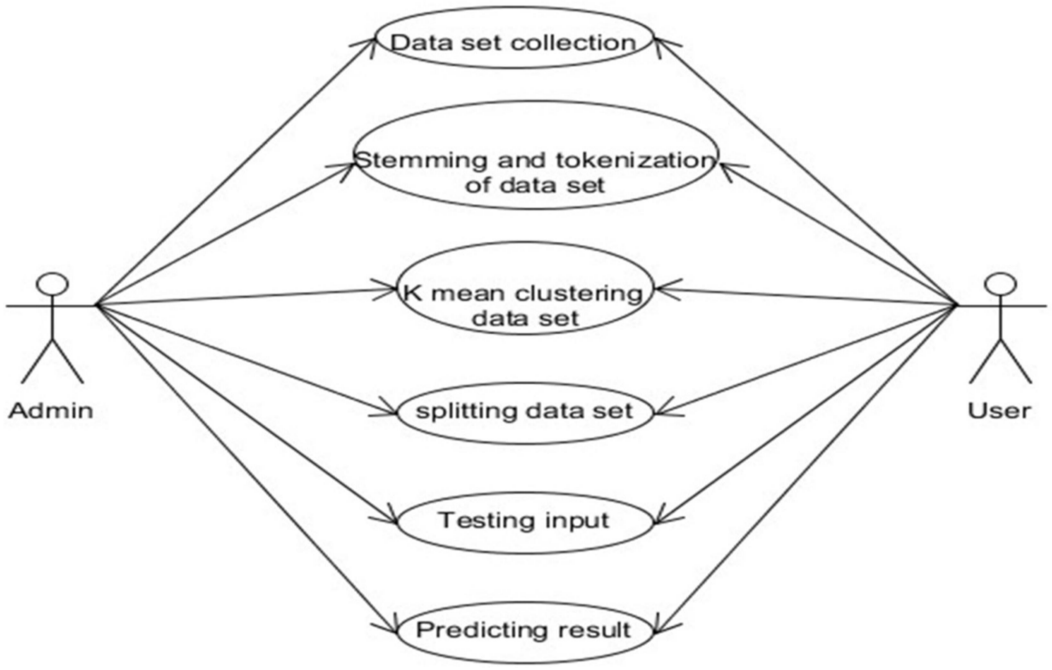


Fig No :5.2

5.3 SEQUENCE DIAGRAM

A Sequence diagram is a type of interaction diagram that shows the order of interactions between objects or components in a system over time. It illustrates the flow of messages exchanged between these objects, emphasizing the time-based aspect of the interactions. In the context of cyber bullying detection on social media, a Sequence diagram can depict the sequence of events and communication between different entities. Here's an example of how a Sequence diagram for this scenario might look.

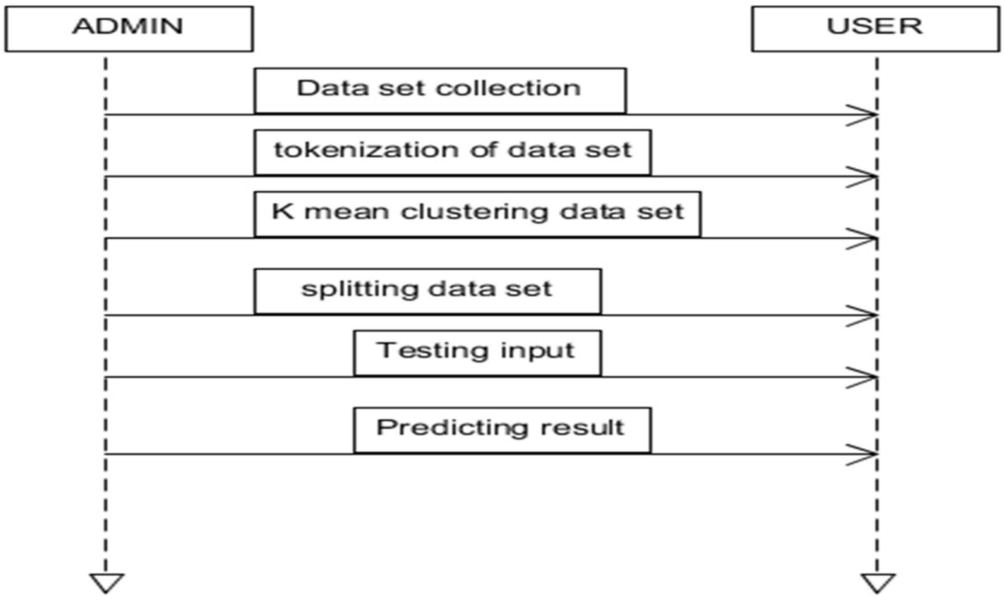


Fig No :5.3

5.4 CLASS DIAGRAM

A Class diagram is a type of structural diagram that shows the classes, interfaces, attributes, and relationships between them in a system. It is used to describe the static structure of a system and the objects that make up the system. In the context of cyber bullying detection on social media, a Class diagram can be used to illustrate the different classes and their relationships in the system. Here's an example of how a Class diagram for this scenario might look:

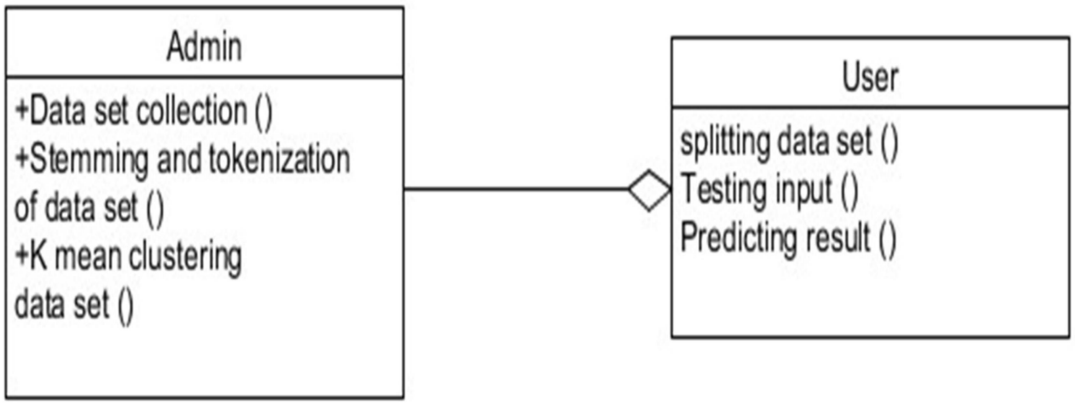


Fig No :5.4

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 MODULES

6.1.1 Data Collection

6.1.2 Data Preprocessing.

6.1.3 NLP Processing.

6.1.4 Model Selection.

6.1.5 Model Deployment.

6.1.1 DATA COLLECTION

- Data's are collected from real time such as twitter,Wikipedia ,kaggle.
- It is an labeled dataset used to classify cyber bullying ornot.
- 1057 data's were trained in this framework.

6.1.2 DATA PREPROCESSING

- In ML most of work is in preprocessing.
- Another from dataset it is way to increase the accuracy.

6.1.3 NLP PROCESSING

FEATURE EXTRACTION

6.1.3.1 BAG OF WORDS MODEL

- In this model, a piece of text is represented as a "bag" (multiset) of its words, disregarding grammar and word order but keeping track of the frequency of each word.
- Unigram model where single words and Bigram modeluses two words and N-gram model is the generalized mode.

6.1.3.2 TF-IDF MODEL

- Term frequency(Tf) is a calculation of frequency of a word in a document. It is measured as chance of finding a text word inside adocument.
- Inverse document frequency (Idf) shows how frequent or rare a word is throughout the corpus. It is used to identify rare words incorpus. Idf value is higher for rarer words.

6.1.3.3 WORD2VEC

- These vector representations capture the semantic and syntactic relationships between words, making them useful for a wide range of natural language processing tasks.
- Continuous bag of words (CBOW) and skip-gram. In CBOW, the algorithm predicts the target word based on its context, whereas in skip-gram, it predicts the context words based on the target word. Both architectures have their advantages and disadvantages depending on the specific taskand dataset.

6.1.4 MODEL SELECTION

ALGORITHMS

6.1.4.1 SUPPORT VECTOR MACHINE(SVM):

- Support Vector Machines (SVMs) are a popular algorithm in machine learning used for classificationand regression analysis.
- The hyperplane is chosen such that it maximizes the margin, which is the distance between the hyperplaneand the closest data points from each class.

6.1.4.2 LOGISTIC REGRESSION

- Logistic regression is a statistical technique used for modeling the probability of a binary response variablebased on one or more predictor variables.
- The sigmoid function maps any input value to a value between 0 and 1,can be interpreted as the probability of the response variable taking on a certainvalue.

6.1.4.3 RANDOM FOREST

- Random forest is a popular machine learning algorithmused for classification, regression, and other tasks. It is an ensemble method that combines multiple decision trees to make a final prediction.
- In a random forest, multiple decision trees are built on different subsets of the training data, with different setsof randomly selected features.
- This helps to reduce over fitting and increase the generalization performance of the model.

6.1.5 MODEL DEPLOYMENT

- It's a developed by a python program and Jupiter filedeployed as an JSON file.
- Web page developed with basic HTML and CSS.

CHAPTER 7

TESTING

7.1 UNIT TESTING

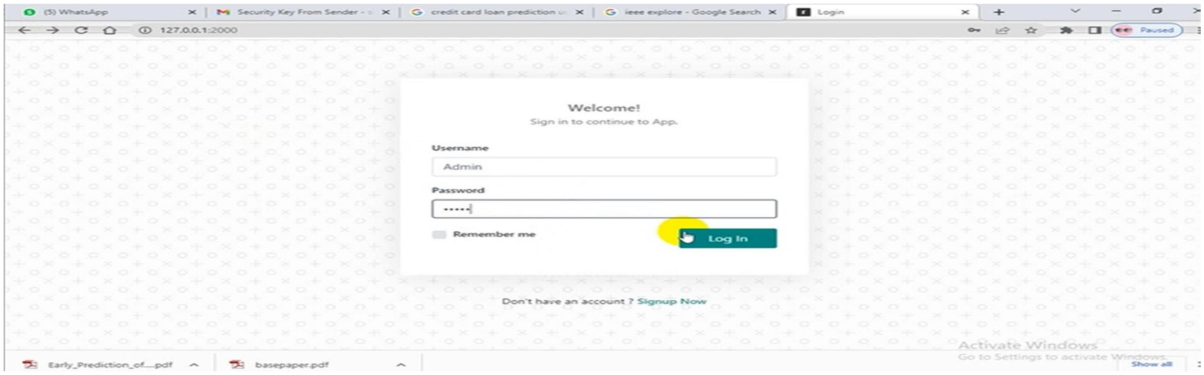
Testing in machine learning refers to the process of evaluating the performance and generalization ability of a trained machine learning model on unseen data. The purpose of testing is to assess how well the model can make predictions or classifications on data it has not been trained on, and to estimate its performance in real-world scenarios. Here are the key aspects of testing in machine learning:

Unit testing in machine learning refer to the process of testing individual components are function of machine learning system in isolation to ensure the they are working as expected .The goals is to identify and fix errors or bugs early in the development of cycle , before they become bigger problems down the line .By unit testing random forest give the accuracy level high. SVM accuracy level is around 85 to 90 percentage .linear regression gives 80 to 85 percentage.

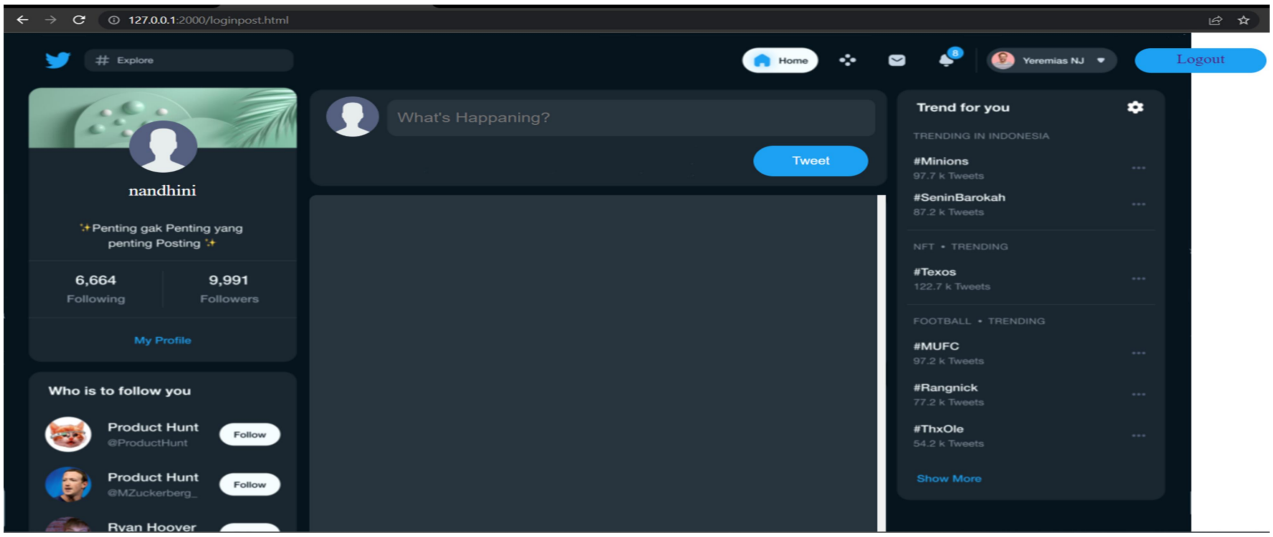
CHAPTER 8

RESULT AND DISCUSSIONS

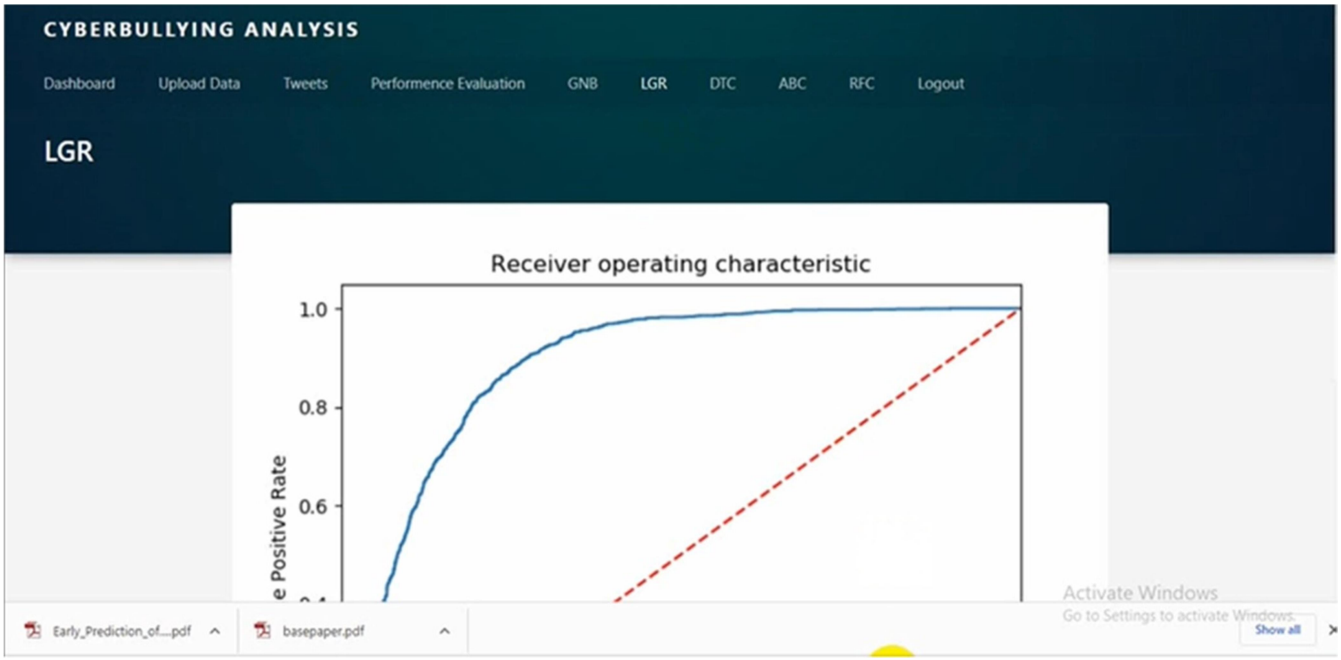
LOGIN PAGE



OUTPUT PAGE



ANALYSIS PROOF PAGE



CHAPTER 9

APPENDIX

9.1 SOURCE CODE

```
from better_profanity import profanity

from flask import Flask,render_template,request,redirect,url_for

import mysql.connector

import os

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.svm import LinearSVC
```

```
    from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import cross_val_score
    from sklearn.metrics import accuracy_score
    from sklearn.metrics import precision_recall_curve
    from sklearn.metrics import plot_precision_recall_curve
    from sklearn.metrics import roc_auc_score
    from sklearn.metrics import roc_curve
    from sklearn.metrics import classification_report
    from sklearn import metrics
    from imblearn.over_sampling import RandomOverSampler
    import datetime


    UPLOAD_FOLDER = 'static/file/'
    app = Flask(__name__)
    app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER


    mydb                                                                    =
    mysql.connector.connect(host="localhost",user="root",password="root",database=
    "cyber")
    mycursor = mydb.cursor()
```

```
@app.route('/')
def login():
    return render_template('login.html')


@app.route('/loginpost.html', methods = ['POST','GET'])
def userloginpost():
    global data1
    if request.method == 'POST':
        data1 = request.form.get('username')
        data2 = request.form.get('password')
        sql = "SELECT * FROM `users` WHERE `name` = %s AND `password` =
%s"
        val = (data1, data2)
        mycursor.execute(sql,val)
        account = mycursor.fetchone()
        if account:
            return render_template('twitter.html')
        elif data1 == 'Admin' and data2 == 'Admin':
            return render_template('dashboard.html')
        else:
```

```
        return render_template('login.html',msg = 'Invalid')
@app.route('/pages-register.html')
def reg():
    return render_template('pages-register.html')


@app.route('/reg',methods=['POST','GET'])
def register():
    if request.method == 'POST':
        name = request.form.get('username')
        phone = request.form.get('phone')
        password = request.form.get('password')
        sql = "INSERT INTO users (`name`, `phone`, `password`) VALUES (%s, %s,
%s)"
        val = (name,phone,password)
        mycursor.execute(sql, val)
        mydb.commit()
        return render_template('login.html')


@app.route('/send',methods=['POST','GET'])
def send():
```

```
    if request.method == 'POST':
        msg = request.form.get('msg')
        censored = profanity.censor(msg)
        now = datetime.datetime.now()
        sql = "INSERT INTO `tweets` (`name`, `date`, `tweet`) VALUES (%s, %s, %s)"
        val = (data1, now, msg)
        mycursor.execute(sql, val)
        mydb.commit()
        if '*' in censored:
            return render_template('twitter.html',view = 'style=display:block', value = 'Hello user! You send wrong word Please Change it!')
        else:
            return render_template('twitter.html',view = 'style=display:block', value = 'Post Tweeted')


@app.route('/tweet')
def tweet():
    sql = 'SELECT * FROM `tweets`'
    mycursor.execute(sql)
    result = mycursor.fetchall()
```

```
    if result:
        return render_template('tweet.html', data = result)
    return render_template('tweet.html', msg = 'No tweets')


@app.route('/upload.html')
def up():
    return render_template('upload.html')


@app.route('/upload',methods=['POST','GET'])
def upload():
    global df
    if request.method == 'POST':
        if os.path.exists('static/file/perform.png'):
            os.remove('static/file/perform.png')
        if os.path.exists('static/file/abc.png'):
            os.remove('static/file/abc.png')
        if os.path.exists('static/file/dtc.png'):
            os.remove('static/file/dtc.png')
        if os.path.exists('static/file/gnb.png'):
            os.remove('static/file/gnb.png')
```

```
if os.path.exists('static/file/lgr.png'):
    os.remove('static/file/lgr.png')
if os.path.exists('static/file/rfc.png'):
    os.remove('static/file/rfc.png')
file1 = request.files['jsonfile']
if file1:
    jsonfile = os.path.join(app.config['UPLOAD_FOLDER'], file1.filename)
    file1.save(jsonfile)
else:
    jsonfile = 'static/file/Dataset.json'
df = pd.read_json(jsonfile)
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()
plt.savefig('static/file/perform.png')
```

```
# pre processing


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']


#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']

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

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(), columns=["TF-IDF"]) #for second entry only(just to check if working)
df2 = df2.sort_values('TF-IDF', ascending=False)
```

```
dfx = pd.DataFrame(tfIdf.toarray(), columns = tfIdfVectorizer.get_feature_names())


def display_scores(vectorizer, tfidf_result):
    scores = zip(vectorizer.get_feature_names(),
        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
display_scores(tfIdfVectorizer, tfIdf)


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

```
#Training data biasness
unique_elements, counts_elements = np.unique(y_train, return_counts=True)
unique_elements, counts_elements = np.unique(y_test, return_counts=True)

oversample = RandomOverSampler(sampling_strategy='not majority')
X_over, y_over = oversample.fit_resample(X_train, y_train)

unique_elements, counts_elements = np.unique(y_over, return_counts=True)

def getStatsFromModel(model):
    # print(classification_report(y_test, y_pred))
    disp = 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 = roc_auc_score(y_test, model.predict(X_test))
    fpr, tpr, thresholds = 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('static/file/roc.png')


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))
plt.title('GaussianNB')
getStatsFromModel(gnb)
plt.savefig('static/file/gnb.png')


lgr = LogisticRegression()
lgr.fit(X_over, y_over)
y_pred = lgr.predict(X_test)
# print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
```

```
# print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
plt.title('Logistic Regression')
getStatsFromModel(lgr)
plt.savefig('static/file/lgr.png')

dtc = DecisionTreeClassifier()
dtc.fit(X_over,  y_over)
y_pred = dtc.predict(X_test)
# print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
# print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
plt.title('Decision Tree Classifier')
getStatsFromModel(dtc)
plt.savefig('static/file/dtc.png')

abc = AdaBoostClassifier()
abc.fit(X_over, y_over)
y_pred = abc.predict(X_test)
# print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
# print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
plt.title('AdaBoost')
```

```
        getStatsFromModel(abc)
        plt.savefig('static/file/abc.png')

        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)
        plt.savefig('static/file/rfc.png')

        return render_template('upload.html',msg='File Upload Successfully...')

@app.route('/dashboard.html')
def dashboard():
    return render_template('dashboard.html')

@app.route('/performence')
def performence():
    return render_template('perform.html',path='static/file/perform.png')
```

```
@app.route('/gnb')
def gnb():
    return render_template('gnb.html',path='static/file/gnb.png')
@app.route('/lgr')
def lgr():
    return render_template('lgr.html',path='static/file/lgr.png')
@app.route('/dtc')
def dtc():
    return render_template('dtc.html',path='static/file/dtc.png')
@app.route('/abc')
def abc():
    return render_template('abc.html',path='static/file/abc.png')
@app.route('/rfc')
def rfc():
    return render_template('rfc.html',path='static/file/rfc.png')

if_____name_== '_main_': app.run(d)
```

CHAPTER 10

CONCLUSION AND FUTURE ENCHANCEMENTS

10.1 CONCLUSION

In conclusion, machine learning can be a powerful tool for detecting cyber bullying on social media platforms. By training models on large datasets of labeled examples, machine learning algorithms can learn to automatically identify patterns in text and detect instances of cyberbullying with high accuracy. The benefits of using machine learning for cyberbullying detection include faster and more efficient identification of instances of cyberbullying, which can help social media platforms take action to protect their users. Overall, machine learning has the potential to play an important role in addressing the problem of cyberbullying on social media, but it should be used as part of a comprehensive approach that includes education, community engagement, and policy changes to create a safer and more respectful online environment.

10.2 FUTURE ENCHANCEMENTS

- Plan to attach framework with Twitter and provide non cyber bullying platform.
- Plan to establish non cyber bulling positive social media application.

REFERENCES

[1] H. Ting, W. S. Liou, D. Liberona, S. L. Wang, and G. M. T. Bermudez, "Towards the detection of cyberbullying based on social network mining techniques," in Proceedings of 4th International Conference on Behavioral, Economic, and SocioCultural Computing, BESC 2017, 2017, vol. 2018-January, doi: 10.1109/BESC.2017.8256403.

[2] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, network: Application to a real case of cyberbullying," 2014, doi: 10.1007/978-3-

[3] A. Mangaonkar, A. Hayrapetian, and R. Raje, "Collaborative detection of cyberbullying behavior in Twitter data," 2015, doi: 10.1109/EIT.2015.7293405.

[4] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," 2016, doi: 10.1145/2833312.2849567.

[5] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, "Detection of Cyberbullying Using Deep Neural Network," 2019, doi: 10.1109/ICACCS.2019.8728378.

[6] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," 2011, doi: 10.1109/ICMLA.2011.152.

[7] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," 2020, doi: 10.1109/ICESC48915.2020.9155700.

[8] M. Dadvar and K. Eckert, "Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study," arXiv. 2018.

[9] S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," arXiv. 2018.

[10] Y. N. Silva, C. Rich, and D. Hall, “BullyBlocker: Towards the identification of cyberbullying in social networking sites,” 2016, doi: 10.1109/ASONAM.2016.7752420.

[11] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” 2016, doi: 10.18653/v1/n16-2013.

[12] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” 2017. [13] E. Wulczyn, N. Thain, and L. Dixon, “Ex machina: Personal attacks seen at scale,” 2017, doi: 10.1145/3038912.3052591.

[14] A. Yadav and D. K. Vishwakarma, “Sentiment analysis using deep learning architectures: a review,” Artif. Intell. Rev., vol. 53, no. 6, 2020, doi: 10.1007/s10462-019-09794-5.

[13] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” 2017. [13] E. Wulczyn, N. Thain, and L. Dixon, “Ex machina: Personal attacks seen at scale,” 2017, doi: 10.1145/3038912.3052591.

[15] A. Yadav and D. K. Vishwakarma, “Sentiment analysis using deep learning architectures: a review,” Artif. Intell. Rev., vol. 53, no. 6, 2020, doi: 10.1007/s10462-019-09794-5.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.