A Project-II Report on

DETECTION OF CYBERBULLYING USING MACHINE LEARNING

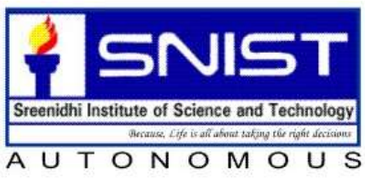
Submitted to the Dept. of Information Technology
in the partial fulfillment of the academic requirements for the award of degree of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

Submitted by

Y. Sai Nithin (19311A12F7)
B. Guru Raghavender (19311A12F8)
T. Sai kumar (19311A12G1)

Under the guidance of
Dr. Sreenivas Mekala
Associate Professor



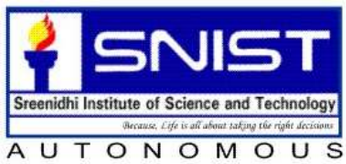
Department of Information Technology
School of Computer Science and Informatics
Sreenidhi Institute of Science and Technology (An Autonomous Institution)
Yamnampet, Ghatkesar Mandal, R. R. Dist., Hyderabad – 501301

affiliated to
Jawaharlal Nehru Technological University Hyderabad
Hyderabad – 500 085

2022–23

Department of Information Technology

School of Computer Science and Informatics
Sreenidhi Institute of Science and Technology



Certificate

This is to certify that the Project-II report on "Detection of Cyberbullying Using Machine Learning " is a bonafide work carried out by Y. Sai Nithin(19311A12F7), B. Guru Raghavrnder(19311A12F8), T. Sai Kumar(19311A12G1) in the partial fulfillment for the award of B.Tech. degree in Information Technology, Sreenidhi Institute of Science and Technology, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad under our guidance and supervision.

The results embodied in the Project-II work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal guide	Project Coordinator	Head of the Department
(Dr. Sreenivas Mekala)	(Dr. Sreenivas Mekala)	(Dr. Sunil Bhutada)

External Examiner
Date:

DECLARATION

We, **Y. Sai Nithin (19311A12F7), B. Guru Raghavender (19311A12F8) and T. Sai Kumar (19311A12G1),** students of **Sreenidhi Institute of Science and Technology, Yamnampet, Ghatkesar,** studying IV year II semester, **Information Technology** solemnly declare that the Project-II work, titled **"Detection of Cyberbullying Using Machine Learning"** is submitted to **Sreenidhi Institute of Science and Technology** for partial fulfillment for the award of degree of Bachelor of Technology in **Information Technology**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

Y. Sai Nithin(19311A12F7)
B. Guru Raghavender(19311A12F8)
T. Sai Kumar(19311A12G1)

Acknowledgements

We would like to express our immense gratitude and sincere thanks to **Dr. Sreenivas Mekala, Associate Professor** in Information Technology for his guidance, valuable suggestions and encouragement in completing the Project-II work within the stipulated time.

We would like to expressour sincere thanks to **Prof. C.V. Tomy**, Director, **Dr. T.Ch. Siva Reddy**, Principal, Dr**. Sunil Bhutada,** Professor &Head of the Department of Information Technology, **Dr. Sreenivas Mekala**, Associate Professor and Project-II Work Coordinator of the Department of Information Technology, Sreenidhi Institute of Science and Technology (An Autonomous Institution), Hyderabad for permitting us to do our Project-II work.

Finally, we would also like to thank the people who have directly or indirectly helped us and parents and friends for their cooperation in completing the Project-II work.

ABSTRACT

Cyberbullying frequently leads to serious mental and physical distress, particularly for women and children, and even sometimes force them to attempt suicide. Online harassment attracts attention due to its strong negative social impact. Many incidents have recently occurred worldwide due to online harassment, such as sharing private chats, rumors, and sexual remarks. The purpose of this research is to design and develop an effective technique to detect online abusive and bullying messages by merging natural language processingand machine learning. Two distinct features, namely Bag-of-Words (BoW) and Term Frequency-Inverse Text Frequency (TFIDF), are used to analyze. With the development of the Internet, the use of social media has increased dramatically over time and has emerged as the most powerful networking tool of the twenty-first century. Social networking sites and online chat programs give users a platform to share their expertise and talents, but a small number of users abuse this opportunity by threatening other users with cyberbullying assaults, which makes it problematic to use these services. Cyberbullying is regarded as a form of technology abuse. According to a recent analysis of global data, the number of incidences of cyberbullying is rising daily.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

CHAPTER-1

INTRODUCTION

1- INTRODUCTION

1.1 INRODUCTION

On social networking sites, users are able to upload a wide variety of content, including images, videos, documents, and even their own interactions with one another. People browse social media on their desktop computers, mobile phones, and other devices. The social networking platforms Facebook, Twitter, Instagram, and TikTok, amongst many others, are among the most well-known ones today. Today, social media is having an effect in many different spheres, including education, business, and even charitable activities, to name just a few of them. One more manner in which the economy of the arena benefits is from the proliferation of new things to do that has been made possible by social media. The use of social media comes with a number of risks, despite the fact that it has a number of benefits. Consumers with malicious intent intentionally engage in unethical and dishonest behaviors via the media in an effort to intentionally cause emotional distress to other people and smear their reputations. Cyberbullying is currently one of the most important problems facing online communities all over the world. The expressions "cyberbullying" and "cyber harassment" are phrases that are used to describe different types of bullying and harassment that occur online. Online bullying can take a variety of forms, including cyberbullying and cyber harassment. Cyberbullying has become an increasingly common problem, particularly among young adults, as a result of the expansion of the Internet and the increase in the number of individuals in each generation.

1.2. MOTIVATION

The victims of cyberbullying and the people closest to them might suffer life-changing consequences as a result of the behavior. Cyberbullying, like more traditional forms of bullying, can have major psychological repercussions, including a reduction in self-esteem; academic failure; rage; anxiety; despair; school avoidance; violence; and even suicide. The practice of cyberbullying has been connected to a variety of mental and physical health issues, including elevated levels of stress and anxiety, feelings of depression, erratic and hostile behaviors, and a lowered sense of one's own self-worth. The psychological harm done by cyberbullying can linger for a significant amount of time after the harassment has stopped

1.3. PROBLEM DEFINITION

The objective is to create a machine learning model capable of accurately identifying instances of cyberbullying in various online platforms, such as social media, messages, and forums. The model should analyze the content and context of the text, recognizing patterns, offensive language, threats, and other indicators of cyberbullying. By effectively detecting cyberbullying, the model can contribute to the prevention and intervention of harmful online behavior, protecting individuals from psychological distress and fostering a more inclusive and respectful online environment.

1.4. OBJECTIVE

We present a technique to the detection of cyberbullying that is based on machine literacy. This method is able to identify whether or not a specific textbook is pertinent to the subject at hand. Several distinct Machine Learning approaches, including Naive Bayes, Vector Machines for Support, Decision Tree, and Random Forest, have been studied as potential components of the proposed cyberbullying discovery model. We put our methodology to the test by analyzing data obtained from user comments on social media platforms like Twitter and Facebook. When evaluating performance, we make use of both the Bow and the TF-IDF point vectors.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 LITERATURE REVIEW

In 2020, Vimala Balakrishnan et al. presented an automatic cyberbullying detection taking Twitter users' psychological features into account. The three main stages discussed in improving cyberbullying detection are Twitter data collection, feature extractions, and cyberbullying detection and classification. The annotated dataset contained 9484 tweets, out of which 4.5% of users are labelled as bullies, 31.8% as spammers, 3.4% as aggressors, and 60.3% as normal. However, the final dataset contained 5453 tweets as a result of the pre-processing step which included removing non-English tweets, profiles containing no data, and special characters. The features extracted were text features, user features, and network features. The model was executed using WEKA 3.8 with 10-fold cross-validation. Since Naïve Bayes performed poorly during preliminary experimental analysis it was eliminated while Random Forest and J48 continued to perform well. The classifiers were trained using manually annotated data.[1]

In 2020, Jaideep Yadav et al. proposed a novel pretrained BERT model developed by Google researchers that generates contextual embeddings and task-specific embeddings. In the proposed method, for the base model, a deep neural network called the Transformer is used. The Bert contains 12 layers to encode the input data and is built on top of a base model. The data is tokenized and padded accordingly and is fed into the model which generates the final embeddings. The classifier layer classifies the embeddings generated by the previous layers and generates the final output accordingly. Using a pre-trained BERT model, they were able to achieve efficient and stable results in comparison to the previous models to detect cyberbullying.[2]

In 2020, Sudhanshu Baliram Chavan et al. proposed the approach to detect cyberbullying on Twitter. The required dataset was collected from sources like GitHub, Kaggle. Initially, the data is pre-processed and features are extracted using a TFDIF vectorizer algorithm. These tweets are then passed through the naive Bayes and SVM model and are classified accordingly. When a tweet is categorized as bullying, ten other tweets from that users' account will be fetched and passed through naive Bayes and SVM classifiers again. If the overall probability of that user's tweets lies above 0.5 then it will be considered as a bullied tweet. Based on the accuracy score and the results it was evident that the SVM model outperformed the naive Bayes with the accuracy score of 71.25%.[3]

In 2019, John Hani et al. presented a supervised learning approach to detect cyberbullying. As a part of the pre-processing step, data is cleaned by removing the noise and unnecessary text. This is performed using tokenization, lowering text, stop words along with encoding cleaning and word correction. The second step is the feature extraction step which is done using TF IDF and sentiment analysis technique including NGrams

for considering different combinations of the words like 2- Gram, 3-Gram, and 4- Gram. The cyberbullying dataset from Kaggle is split into ratios (0.8, 0.2) for train and test. SVM and Neural networks are used as classifiers that run on a different n-gram language model. Accuracy, recall and precision, and f-score are the performance measures. It is found that Neural Network performed better than the SVM classifier. Neural Network achieved an average f-score of 91.9% and SVM achieved an average f-score of 89.8%.[4]

In 2018, Monirah Abdullah Al-Ajlanet and Mourad Ykhlef proposed a novel algorithm CNN-CB which is based on a convolutional neural organization and adapts the idea of word embedding. The architecture comprises four layers - Embedding, Convolution Layer, Max Pooling Layer, and Dense Layer. The first layer, word embedding, creates a vector space of vocabulary which is the input to the subsequent layer, the convolutional layer, which compresses the input vector without losing significant features. The third layer, the Max pooling layer, takes the output of the second layer as its input and finds the maximum value of the chosen region to save just significant highlights. The last layer, the Dense layer, does the classification. This gave a precision of 95%. [5]

In 2018, Monirah A. Al-Ajlan et al. proposed optimized Twitter cyberbullying detection based on deep learning (OCDD) which does not extract features from tweets instead, it represents a tweet as a set of word vectors that are fed to a convolutional neural network (CNN)for classification. Hence the feature extraction and selection phases are eliminated in this approach. To represent the semantics between words, word embedding is used and is generated using (GloVe) technique. CNN uses a lot of parameters and to optimize these values, a metaheuristic optimization algorithm is used to find optimal or near-optimal values that will be used for classification. CNN showed great results. [6]

In 2017, Yee Jang Foong and Mourad Oussalah presented an automated cyberbullying detection that uses natural language processing techniques, text mining, and machine learning. For dataset ASKfm, a social media platform where users can anonymously ask questions and view a sample of a user's profile is used. As a part of the pre-processing procedure web links and unknown characters are removed, incorrect wordings in case any are corrected, and also lexicons are replaced with equivalent textual expressions. A combination of features has been used which includes TF-IDF, Unusual capitalization count, LIWC, and Dependency parser. The data set is split into a 70% training set and 30% testing set. [7]

In 2016, X. Zhang et al. proposed a novel approach based on a pronunciation-based convolutional neural network (PCNN). Word-to-Pronunciation conversions done to group a set of words spelled incorrectly, which have the same meaning and pronunciation, together with the corrected word. Two separate CNN is used to establish a baseline. For the first baseline feature set, word-embedding based on 5 Google's word-vector was used. For the creation of the feature set of the second baseline, CNN Random, arbitrarily generated vectors were used. The phoneme codes were arbitrarily introduced into vectors for the feature set for PCNN. To handle class imbalance three techniques were implemented- threshold moving, cost function adjust, and a hybrid

solution, out of which cost function adjusting is most effective. [8]

In 2016, Michele Di Capua et al. presented an unsupervised approach to detect cyberbullying using a design model inspired by Growing Hierarchical SOMs. Firstly, features are divided into four groups: Syntactic features, Semantic features, Sentiment features and Social features. Growing Hierarchical Self Organizing Map (GHSOM) network algorithm, which is well suited for a large collection of documents that has to be classified, is used. It uses a hierarchical structure of multiple layers, where each layer consists of a variety of independent SOMs. A single SOM is employed at the root layer. For every unit, during this map, a SOM could be added to the subsequent layer of the hierarchy. GHSOM Network is trained and tested concerning a K-folded dataset, applying a K-fold partitioning of data. [9]

In 2014, Sourabh Parime and Vaibhav Suri presented an approach of using data mining and machine learning techniques to detect cyberbullying. Text mining is performed on unstructured data using machine learning techniques to extract knowledge from the text which includes multiple stages like document clustering, data pre-processing, attribute generation for which an in-built classifier is used to generate labels from the features fed into it and occurrences are counted and a weight is assigned to each label and irrelevant attributes are removed which helps to estimate the nature of the comments. Sentiment analysis is used for determining the tone of the given text. [10]

AUTHORS	TITLE AND YEAR OF PUBLICATION	ALGORITHM (CLASSIFIERS)
Balakrishnan, Vimala et al. [1]	Improving cyberbullying detection using Twitter user's psychological features and machine learning (2020)	Random Forest, J48
J. Yadav, D. Kumar et al. [2]	Cyberbullying Detection using PreTrained BERT Model (2020)	Pre-trained BERT
Sudhanshu Baliram Chavan et al. [3]	Detecting A Twitter Cyberbullying Using Machine Learning (2020)	SVM, Naive Bayes
John Hani, Mohamed Nashaat et al. [4]	Social Media Cyberbullying detection using machine learning (2019)	SVM, Neural Network
Monirah Abdullah AlAjlanet and Mourad Ykhlef [5]	Deep Learning algorithm for cyberbullying detection (2018)	CNN-CB
M. A. Al-Ajlan and M. Ykhlef [6]	Optimized Twitter Cyberbullying Detection based on Deep Learning (2018)	CNN
Yee Jang Foong and Mourad Oussalah [7]	Cyberbullying System Detection and Analysis (2017)	SVM
X. Zhang et al. [8]	Cyberbullying detection with a pronunciation based convolutional neural network (2016)	CNN, PCNN
Michele Di Capua, E. Di Nardo et al. [9]	Unsupervised Cyberbullying detection in social networks (2016)	GHSOM Network Algorithm
Sourabh Parime and Vaibhav Suri [10]	Cyberbullying detection and prevention: Data Mining and psychological perspective (2014)	SVM

CHAPTER-3

ANALYSIS

3. ANALYSIS

3.1. EXISTING SYSTEM

The current system consists of switches to control the system and an electrical wiring circuit that connects street lights with a power supply. The switch controls the entire system since it decides whether the circuit is open or closed. From the place where the system is physically located, it must be manually operated. Other than the actual position of the switches given for the system or the power source, this system cannot be managed from anywhere. It is also apparent that this arrangement necessitates operating each switch for each of the matching streetlights separately. Given that the entire electronic circuit is made up of manually operated switches, the system has the potential to become intelligent. They could be thought of as the system's overall control. Here, the ability to manage the switch remotely and thereby the complete system led to the creation of this project. Cyberbullying affects over 50% of kids in America. The victim of this bullying suffers both psychologically and physically. The pain of cyberbullying, which is difficult to sustain, leads the victims to pursue self-destructive behaviors like suicide. Therefore, it's crucial to recognize and stop cyberbullying if you want to keep teenagers safe. Decision tree techniques are used in the current machine learning application for cyberbullying detection, but this algorithm is ineffective in categorizing the cyberbullying texts.

Disadvantages of Existing System:

1. A change in the data that has a significant impact on the structure of the decision tree is one of the potential factors that can lead to an unstable decision tree.
2. The computations that go into a decision tree can become quite complex, particularly when contrasted with those of alternative approaches.
3. The training process for a decision tree model takes significantly more time.
4. Because of the intricacy of the process and the amount of time that is required, the cost of training a decision tree is relatively expensive.
5. The approach of decision trees is not useful for making accurate predictions of continuous values or regressions.

3.2. PROPOSED SYSTEM

In our proposed setup, we take advantage of the LDR's ability to change its resistance based on the amount of light hitting it. Our proposed approach makes use of LDR to determine day from night. An infrared (IR) sensor detects passing vehicles and triggers the streetlight to turn on and off automatically. An infrared sensor

detects a lack of pedestrian or vehicle traffic and automatically dims the headlights. However, if the car is on the road, the signal will activate.The foundation for cyberbullying detection is made up of two parts: natural language processing (NLP) and machine learning (ML). Raw text is useless to machine learning algorithms. Therefore, we must first transform the methods into vectors or integers before applying them. The processed data is transformed into a Bag-of-Words for use in the succeeding phase. We also take into account the TF-IDF attribute in our model. To do this, you can use the statistical technique known as TF-IDF (phrase Frequency-Inverse Document Frequency), which calculates how important a given phrase is to each document in a given set of documents. Tools like SMOTE and Xgboost are used to make enhancements.

Methodology:

- Natural Language processing: Real-world posts and communications often contain a variety of extraneous symbols, letters, and words. For instance, neither numerals nor punctuation have any bearing on the ability to recognise bullying. Before employing the machine learning approaches, we must first clean and prepare the comments for the detection phase. In this phase, several processing tasks are carried out, such as the removal of stop words, punctuation, and digits, tokenization, stemming, etc. The two most important components of the texts are set up after pre-processing as follows:
- 1 Bag-of-Word.:
- • Machine learning methods cannot be applied directly to the raw text. In order to use the algorithms, they must first be transformed into vectors or numbers. Bag-of-Words (BoW) is an intermediate format after data processing is complete.
- 2. TF-IDF:
- For our model, we additionally consider this factor. Term relevance in a set of documents can be determined using the TF-IDF (Term Frequency-Inverse Document Frequency) statistical method. In a bag of words, each word has the same importance, but in a TF-IDF, the more often occurring words should be given more importance because they are more suited for categorization.
- Machine Learning: In this study, indicators of bullying are looked for in text and messages through the application of machine learning techniques such as Random Forest. We establish the best classifier for a particular dataset relating to cyberbullying that is accessible to the public.

3.3. SOFTWARE REQUIREMENT SPECIFICATION

Programming Language : Python
IDE : Jupyter
UML Design : Argo UML

Tools : PIP

3.3.1. Purpose

The primary goal of this project is to design and put into practise a technique that makes use of machine learning and natural language processing to consistently identify instances of cyberbullying and abusive speech in online conversations. This will be accomplished through a combination of these two approaches. Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TFIDF) are two of the variables that are utilised in this study to evaluate the degree of precision achieved by four various machine learning approaches.

3.3.2. Scope

According to several widely used words, this programme would be capable of reliably identifying social media messages as positive or bad. uses five terms with varying degrees of polarity with a primary focus on bullying. It operates in a real-world setting. Accurate models are trained using the most recent machine learning techniques. The procedure for detecting cyberbullying is automatic, takes little time, and is accurate.

3.3.3. Overall Description

This cyberbullying detection system is intended to recognize and categorize cyberbullying behaviors such as age, gender, and religion on social media. The Xgboost algorithm is used to detect cyberbullies, while the TF-IDF classifier is used to categorize their behavior.

CHAPTER-4

DESIGN

4.DESIGN

4.1. UML DIAGRAMS

UML is an abbreviation for "Unified Modelling Language," which is its full name. Object-oriented software developers often use a common general-purpose modelling language called the Unified Modelling Language (UML). The Object Management Group created this standard and is in charge of maintaining it. The purpose of UML, or the Unified Modelling Language, is to standardize the language used in expressing models of OO software. In its current form, the Unified Modelling Language consists mostly of a notation and a meta-model. The Unified Modelling Language (UML) could grow to incorporate other methods or become associated with existing ones in the future. When it comes to describing, visualizing, constructing, and documenting software system artefacts, the Unified Modelling Language (UML) is the language of choice. It's the benchmark of excellence. Business modelling and the design of non-software systems are two more common UML uses. The Unified Modelling Language (UML) can be used to simulate large and complex systems since it is a collection of proven engineering best practices.

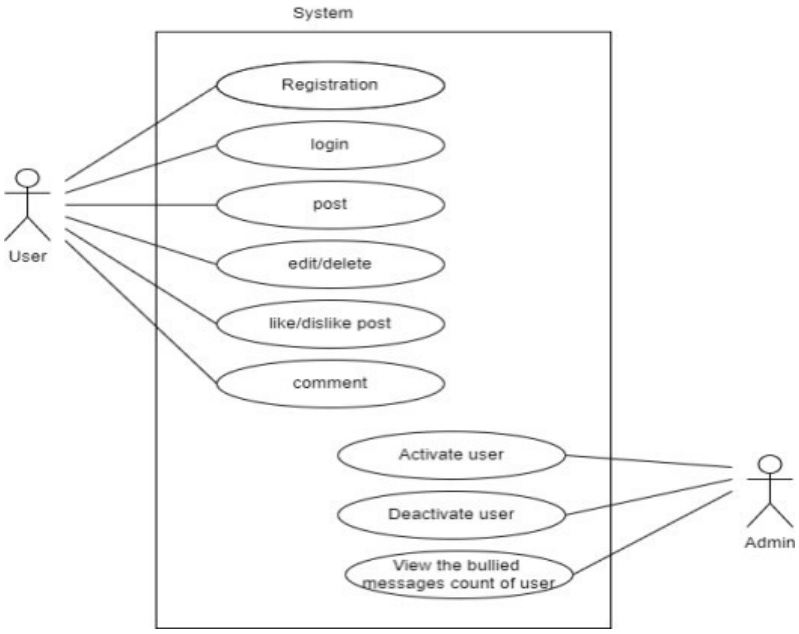
Principal Goals:

Following are some of the primary motivations behind the UML layout:

1. Enable users to create and share insightful models by providing them with an accessible visual modelling language.
2. Offer means for expanding and specializing the fundamental ideas.
3. Show a lack of concern for certain programming languages or development methodologies.
4. Develop a rigorous method for learning the modelling language.
5. Boost the development of the OO tool market.
6.Create a list of recommended Procedures
7. Encourage the application of ideas from more advanced stages of growth, such as collaboration, frameworks, patterns, and components.

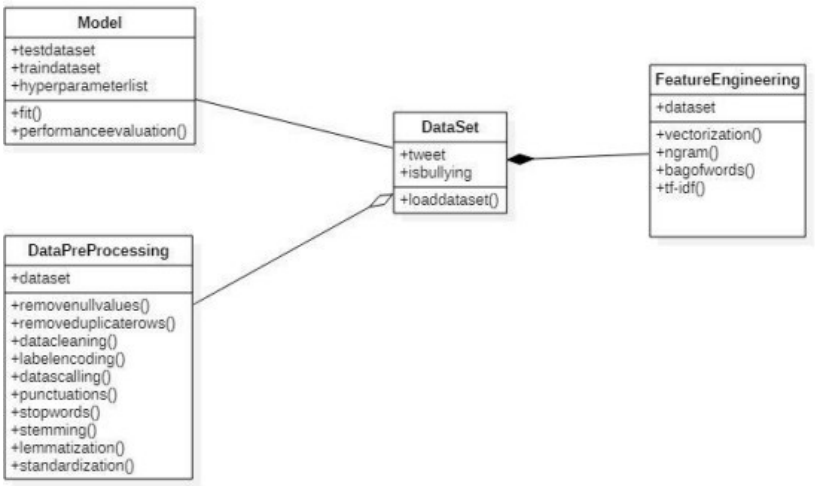
4.1.1 Use Case Diagram

A use case diagram, which is part of the Unified Modelling Language (UML), can be used to summarise the users of your system, who are often referred to as actors, as well as their interactions with the system. You can't just slap one together; you'll need a particular set of connectors and symbols for it to work properly.



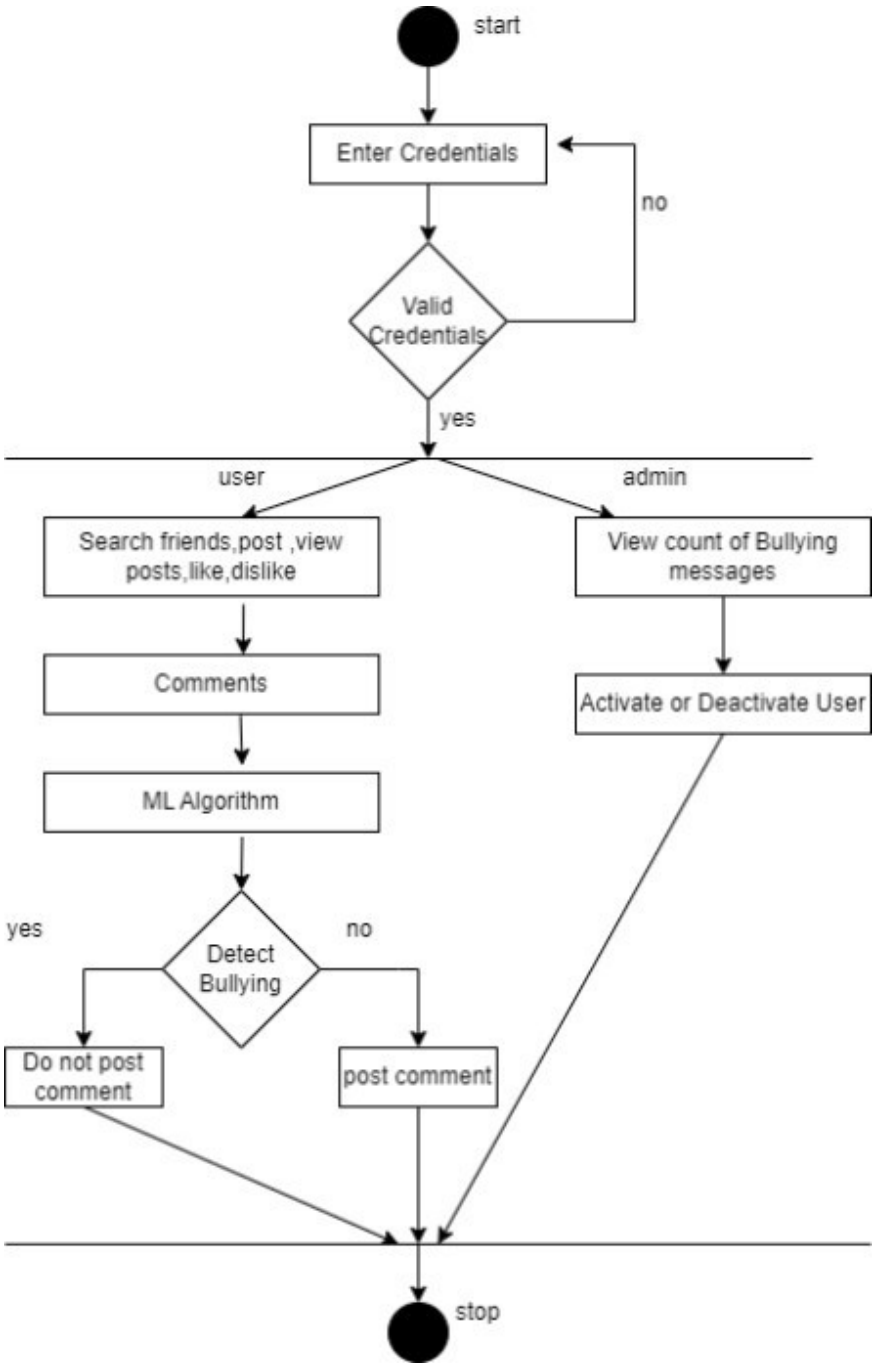
4.1.2 Class Diagram

The class diagram is a still image of the programme in its current state. It illustrates the many components that make up the system as well as the relationships between those components. A class is able to both create its own objects and inherit the characteristics and behaviours of other classes in addition to its own. Class diagrams are utilised in the construction of software code as well as in the visualisation, description, and documentation of the numerous distinct components that make up the system.



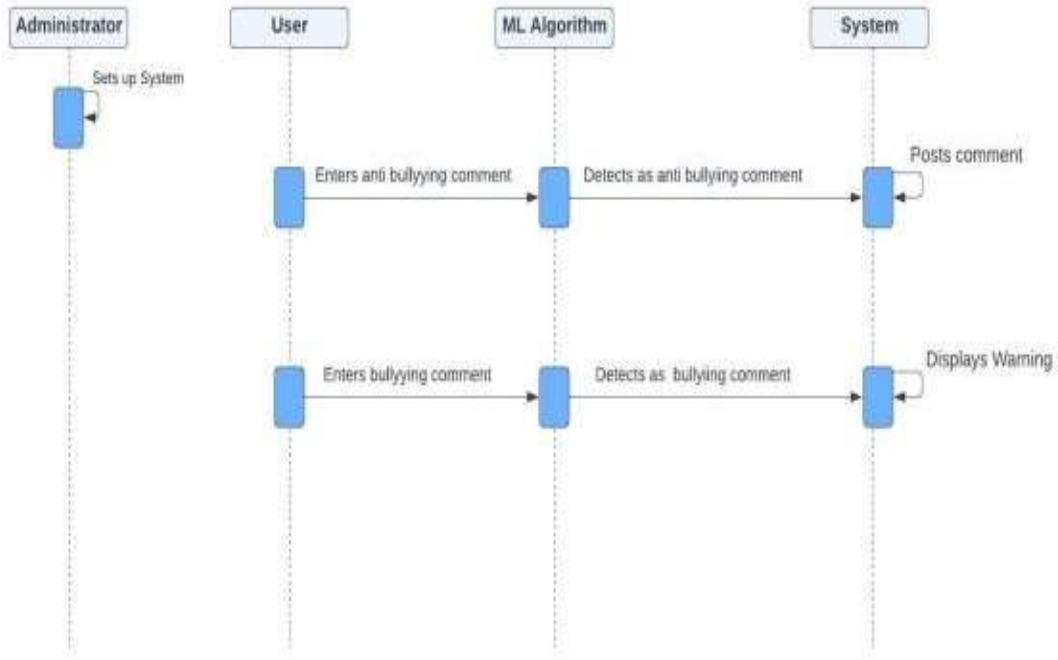
4.1.3 Activity Diagram

The activity diagram is yet another essential UML behavioural diagram that is used to depict the dynamic components of the system. An activity diagram is utilised whenever there is a need to model the progression of work from one task to the next.



4.1.4 Sequence Diagram

The flow of messages through the system is represented by a sequence diagram, which is also referred to as an event diagram in some contexts. It can be put to use in the creation of a diverse range of dynamic environments.



CHAPTER-5

IMPLEMENTATION

5.IMPLEMENTATION

5.1 PROCEDURE

The development of the project is based on the Dataset considered and effective tuning of parameters of Machine Learning Algorithms. The system consists of basically 4 phases:

1. Data Gathering

2. Data processing

3. Training Phase

4. Testing Phase

1. Data Gathering: The dataset presented here consists of a set of tweets that were gathered using the Twitter API. More than 1000 tweets from various time periods were entered into the database. The datasets showing Text Labels are represented in the ensuing photos.

2. Data Processing: Prior to using the regression model, the raw data must be adjusted to meet our needs. Since raw data is frequently inconsistent, fragmentary, lacking in specific behaviours or features, and possibly even containing noise. Therefore, we must eliminate all of these anomalies and transform the dataset into something that machine learning algorithms can use. As a result, we analyzed data from internet sources every day to produce meaningful data metrics linked to profanity in the output that could be utilized to train our models. We downloaded some comment data in xlsx format. Therefore, we had to convert the xlsx file format to the common CSV format for machine learning model training. Additionally, data can occasionally contain a variety of inconsistencies, including noisy data that a model cannot comprehend and value dominances of one variable over another, which can make a model inconsistent in its ability to make accurate predictions.

3. Training Phase: We first import a certain algorithm class or module and construct an instance of it in order to train the model. We then fitted the model to the training data using that specific instance. Then, we verify it by evaluating its accuracy score and adjusting its parameters till the desired results are obtained.

4. Testing Phase: To test the model, we compare test data with anticipated values from the training phase. After that, enter a new value to see if the forecast was accurate. Fine-tune the algorithmic parameters and re-fit the model if it didn't predict at that time.

5.2. MODULES

- Numpy
- Pandas
- Scikit-learn
- Nltk
- TF-IDF Vectorizer
- imblearn.pipeline
- Random Forest Classifier
- SMOTE
- XGBoost
- Pickle

5.3. MODULE DESCRIPTION

Numpy:

Numpy is a library for the Python programming language that allows users to manipulate data arrays. Arrays of data in various dimensions can be worked with in a quick and flexible manner because to the object that it provides.This module is the fundamental building block for Python's ability to perform scientific computing. The following is a list of some of its more notable characteristics:

one that is successful in N dimensions despite difficult (radio) operations

Skills in the Fourier transform, random number generation, and linear algebra are all put to use in the development of devices for combining C/C++ and Fortran codes.

In addition to the obvious uses it has in the scientific community, Numpy may also be used as a powerful multidimensional data container. Numpy's ability to easily and quickly communicate with a wide variety of database management systems is made possible by its versatility in configuring any data type.

Pandas:

Cleaning, filling, normalising, merging, and visualising the data are some of the laborious and time-consuming chores that are made easier by using Pandas, which is a tool for working with data.

• Statistical investigation

• Data inspection • Data loading and archiving

We utilized it in the current project to load data into the form. Pandas csv:

Scikit-learn:

Scikit-learn is a free package that may be used for machine learning and is compatible with the Python programming language. It includes support vector machines, random forests, gradient boosting, k-means, and DBSCAN among its many classification, regression, and clustering methods. In addition, support is provided for the scientific and numerical Python libraries known as NumPy and SciPy.

The sklearn.preprocessing package includes a wide number of helpful utility methods and transformer classes. These methods and classes are used to convert raw feature vectors into a format that is more suitable for the estimators that come later. Encoding target labels with values ranging from 0 to n_classes-1 must be done since this requirement must be met.

Labels can be normalized using Label Encoder.

Additionally, it can be used to change non-numerical labels.

Nltk:

The Python framework Natural Language Tool Kit is used to create NLP applications. Natural Language Processing (NLP) tasks include:

document classification based on 'Topics' looking up words in documents

Recognition of voice identification of handwriting

examining handwritten text for grammatical and spelling issues evaluating handwritten exam answers from universities or schools processing general documents and texts

import nltk.corpus PorterStemmer: Stemmers remove morphological affixes (such plural words etc.) from words, leaving only the word stem. stop words: eliminates stop words from a piece of text.

TF-IDF Vectorizer:

At first appearance, determining the total number of words in a text appears to be a simple task. The frequency with which articles such as "the," "an," "your," etc. appear in written texts makes word counts alone insufficient for text processing. The length of the passages does not matter for the purpose of the text analysis. Stop words can be efficiently removed from a text by the utilisation of TF-IDF.

The term "Term Frequency - Inverse Document Frequency" refers to this technique. A numerical statistic called the TF- IDF rates the significance of each word in a document.

Term Frequency: The frequency with which a word appears in text.

Inverse Document Frequency: Determine if a word is frequently used or infrequently used in a document.

imblearn.pipeline: The pipeline's goal is to put together a number of phases that can be cross-validated while using various parameters. With their names and the parameter name separated by a", it makes it possible to set the parameters of the numerous steps.

The utility function "make pipeline" is a shortcut for creating pipelines. It receives a pipeline by

automatically filling the names with a variable amount of guesses.

Random Forest Classifier:

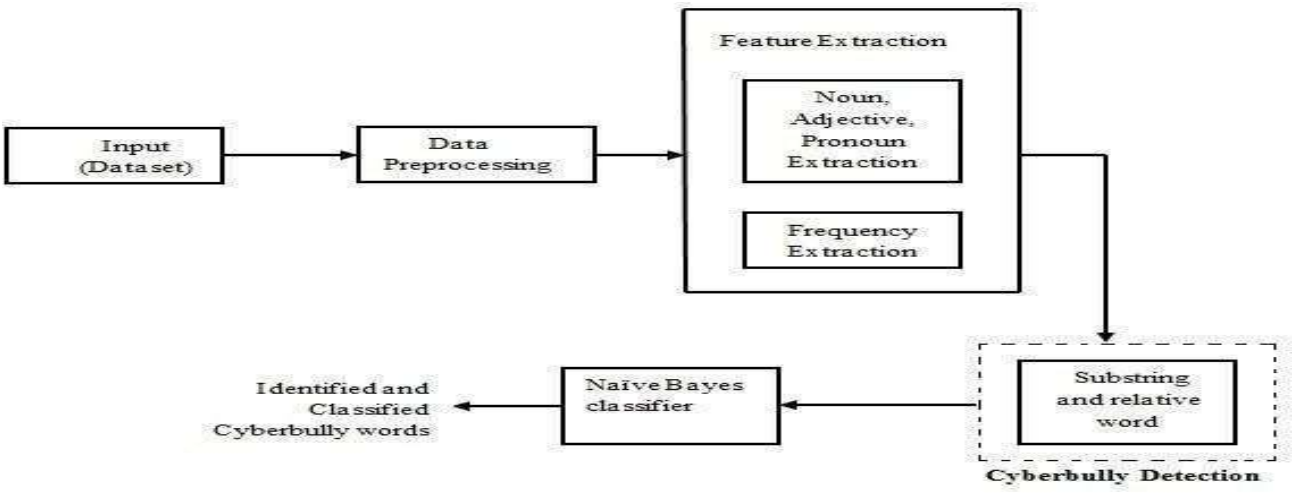
The Random Forest classifier is made up of numerous distinct decision trees that are used individually. Each tree offers a unique prediction for the class that is being considered. The final result that we have is the maximum size of the class that was projected. This classifier generates an output that is a composite of several decision trees, which enables it to function as a supervised learning model that generates accurate results. The random forest approach doesn't rely on just one decision tree like other approaches do; rather, it takes into account all of the different forecasts and bases its ultimate conclusion on those. The Random Forest classifier is made up of numerous distinct decision trees that are used individually. Each tree offers a unique prediction for the class that is being considered. The final result that we have is the maximum size of the class that was projected. This classifier generates an output that is a composite of several decision trees, which enables it to function as a supervised learning model that generates accurate results. The random forest approach doesn't rely on just one decision tree like other approaches do; rather, it takes into account all of the different forecasts and bases its ultimate conclusion on those.

SMOTE:

The Synthetic Minority Oversampling Technique involves oversampling the minority class in order to address imbalanced datasets. Duplicating examples from the minority class is the simplest method, but these examples don't provide any new insight into the model. Instead, fresh examples can be created by synthesizing the current ones.

XG Boost:

A gradient boosting package for Python is called Extreme Gradient Boosting. Gradient-boosted decision trees serve as its foundation. It is a boosting algorithm that is used to increase the model's robustness and accuracy in various competitions like Kaggle. For small to medium-sized structured or tabular data, decision tree-based algorithms are thought to be the most effective. Using XGBoost to create models is a simple process. But increasing its effectiveness is quite challenging. In XGBoost, there are a variety of parameters that need to be tuned.



Benefits of XGBoost:

Highly adaptable: we may create unique optimisation goals and assessment standards. Additionally, Hadoop implementation is supported.

Parallel Processing: It uses parallel processing to its fullest advantage and is quicker than gradient boosting.

Regularization :Standard Gradient Boosting is susceptible to over-fitting since it lacks regularisation. But regularisation is supported by XGBoost. It's also referred to as "Regularised Boosting."

Missing Values: It contains techniques built-in to manage missing values.

Tree Pruning: When using Gradient Boosting, the algorithm halts whenever it experiences a split with a negative loss. In contrast, in the instance of XGBoost, it increases to max_depth and begins to prune the splits with no net gain.

Built-in Cross Validation: Unlike Gradient Boosting, which employs GridSearch for these operations, it enables the user to execute a cross-validation after each iteration.

Keep going Current Mode: We can keep using the current model in XGBoost

Pickle:

A Python object structure can be serialised and deserialized with the help of the pickle module, which is part of Python. Any Python object can be saved to disc using a technique called pickling. Before saving the item as a file, Pickle performs a process known as "serialisation." Pickling is an operation in Python that transforms a Python object, such as a list or dict, into a string consisting of character values. This character stream ought to have all of the information required to recreate the object within another Python function.

5.4. INTRODUCTION TO TECHNOLOGIES USED

What is Python?

Python is a programming language that is utilised in the creation of software. A few Python-related nuggets that are noteworthy are presented in the following text.

Python is, by a significant margin, the most frequently selected option among high-level languages. Python is a versatile programming language that may be utilised in either an Object-Oriented or Procedural setting. The size of comparable Python code is often less than that of comparable Java or C++ code.

Programmers have a reduced need to type, and the language's emphasis on indentation ensures that their code will always be straightforward to read.

Python is used by a large number of the most well-known technology businesses in the world, such as Google, Amazon, Facebook, Instagram, Dropbox, and Uber, to name just a few.

The adaptability of Python's enormous standard library is widely regarded as one of the programming language's most appealing qualities.

• Software with a User Interface (GUI)

• machine learning

• • The Django framework was developed as free software. Image manipulation

- Scraping from the internet
- Multimedia
- Test frameworks

Python's merits:

1. Extensive Libraries

Python's vast library includes a wide variety of useful tools, some of which are described below: regular expressions; the generation of documentation; unit testing; web browsers; concurrent access to data; common gateway interfaces; electronic mail; image processing; and so on. Because of this, we won't have to manually compose the entirety of the code.

2. Extensible

We have shown that Python can be adjusted to facilitate translation between a variety of different languages. It's possible that some of your job will require you to use programming languages such as C++ and C. In the field of project management, this is a very valuable asset.

3. Embeddable

The capability of integrating Python is another factor that contributes to the adaptability of the language. Python code can be incorporated into source code written in C++ or Java. Because of this, we are able to implement scripting capabilities directly into the source code of our other languages.

4. Improved Productivity

Programmers are able to be more productive when using this language due to its ease of use and its comprehensive library, in comparison to languages such as Java and C++. It is necessary to complete additional work while spending less time writing.

5. IOT Opportunities

Python, a programming language that is utilised by emerging platforms such as the Raspberry Pi, forecasts that the Internet of Things will have a prosperous future. This is one method for eliminating the barriers that exist between various languages and the outside world.

6. Simple and Easy

When working with Java, you can find yourself in a situation where you need to design a class simply to print the phrase "Hello World." However, all that is necessary for Python is a print statement. In addition to this, it is simple to learn, comprehend, and include into one's coding. One of the reasons why Python developers frequently suffer when transitioning to other languages like Java is because of this.

7. Readable

It is much simpler to read code written in Python than in English due to Python's less verbose nature. As a result of this, it is extremely user-friendly and approachable for programmers of all experience levels. Indentation is essential for defining blocks, although curly brackets are not required under any circumstances. This makes it far simpler to comprehend the code.

Compared to Other Languages, Python Has Many Benefits

1. Reduced Coding

When compared to other programming languages, Python requires far less code to do almost any given task. Because the standard libraries that come with Python are so well-supported, you won't need to install any additional packages in order to accomplish what you set out to do. Python is frequently advised to those who are just beginning their careers as computer programmers for this and other reasons.

2. Affordable

Because Python is both open-source and free, organisations of all sizes, from sole proprietorships to global conglomerates, can reap the benefits of the language's capabilities. Because Python is so widely used, you should have an easier time finding people who are willing to assist you.

3. Python is for Everyone

Python applications can be run on any of the most popular operating systems, including Windows, Mac OS X, and Linux. To remain competitive in the industry, developers need to be fluent in multiple languages; yet, Python's growing popularity can be attributed to the language's adaptabilit

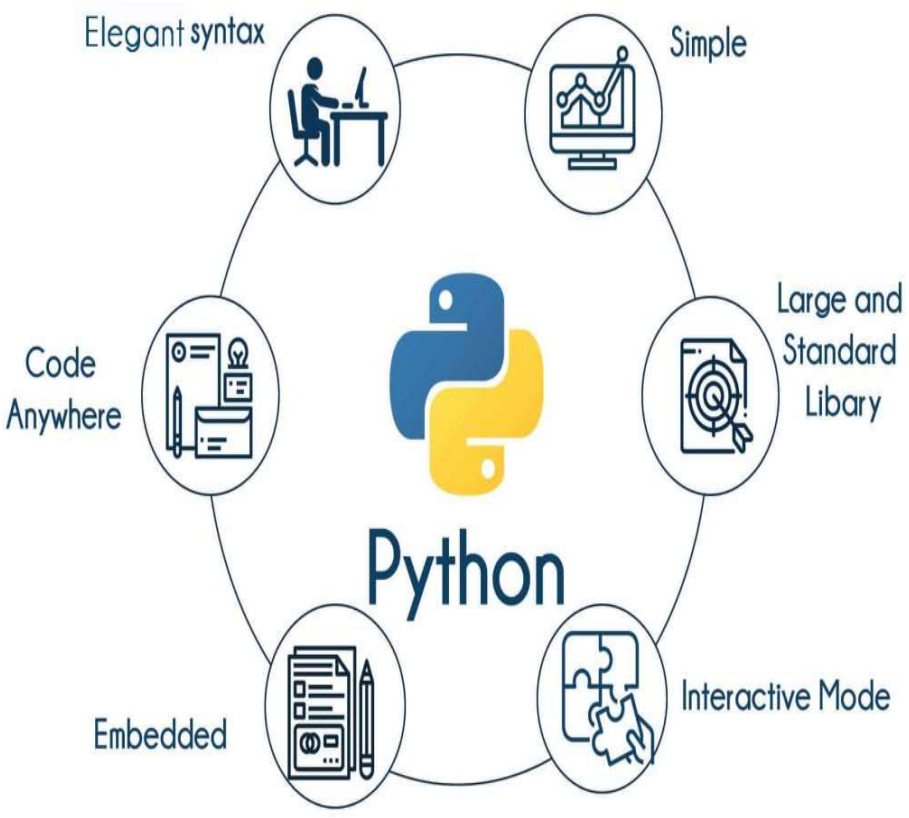


Fig. 5.3.1: Python Features

Jupyter

Jupyter notebook is a free and open-source web application that allows users to write and share documents that contain interactive computations, as well as code, photos, and other features. Users are able to generate dynamic "stories" by combining code, data, and various visualisations within a single notebook. In addition to Python code, notebooks may also include paragraphs, markdown, figures, links, and other textual components. Jupyter notebooks have an intuitive user interface that makes it easy to create, edit, and execute notebooks. Additionally, these notebooks are frequently utilised and extensively documented. A online interface known as "Dashboard" or "control panel" that displays local files allows users to see notebook pages and run code snippets. The format of the results, as well as their presentation on the browser, is designed to be aesthetically pleasant.

5.5. SAMPLE CODE

```
import numpy as np
import pandas as pd
data = pd.read_csv("cyberbullying_tweets.csv")
data.head()
data.isnull().sum()
data.shape
data.dropna(inplace=True)
data.isnull().sum()
data.shape
data.head()
from sklearn import preprocessing


# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()


# Encode labels in column 'species'.
data['cyberbullying_type']= label_encoder.fit_transform(data['cyberbullying_type'])


data['cyberbullying_type'].unique()
X = data['tweet_text'].values
y = data['cyberbullying_type'].values
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
nltk.download("stopwords")
ps = PorterStemmer()
corpus = []
for i in range(len(X)):
```

```
print(i) news = re.sub('[^a-zA-Z]', ' ', X[i])
news = news.lower()
news = news.split()

news = [ps.stem(word) for word in news if word not in stopwords.words('english')]
news = ' '.join(news)

corpus.append(news) X[0], corpus[0]

!pip install imblearn

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from imblearn.pipeline import Pipeline, make_pipeline

X_train, X_test, y_train, y_test = train_test_split(np.asarray(corpus), y, test_size = 0.2,
random_state = 24)
X_train.shape, X_test.shape
X_train[0],          y_train[0]
X_test[0], y_test[0]
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
results=[]
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
tfidf = TfidfVectorizer()
model = RandomForestClassifier()
x_train = tfidf.fit_transform(X_train)
x_test = tfidf.transform(X_test)
model = RandomForestClassifier(bootstrap=False, class_weight='balanced',
                criterion='gini',     max_depth=None,     max_features='auto',
                max_leaf_nodes=None,
                min_impurity_decrease=0.0,          min_impurity_split=None,
                min_samples_leaf=1,                      min_samples_split=2,
                min_weight_fraction_leaf=0.0,               n_estimators=250,
                n_jobs=None, oob_score=False, random_state=42, verbose=0,
                warm_start=False)

from imblearn.over_sampling import SMOTE
pipe_line = Pipeline([
                ('tfidf', tfidf),
```

```
                ('smote',
                SMOTE(random_state=12)), ('model',
                model)
])

pipe_line.fit(X_train,  y_train)
pred = pipe_line.predict(X_test)
print(accuracy_score(y_test, pred))
acc=accuracy_score(y_test, pred)

results.append(acc)
import pickle
!pip install xgboost==1.1.1
import xgboost
from xgboost import XGBClassifier
xgb_model  =  XGBClassifier()
xgb_pipe_line = Pipeline([
                ('tfidf', tfidf),
                ('smote', SMOTE(random_state=12)),
                ('model', xgb_model)
])

xgb_pipe_line.fit(X_train,  y_train)
pred = xgb_pipe_line.predict(X_test)
acc=accuracy_score(y_test,  pred)
results.append(acc)

print(accuracy_score(y_test,                pred))
pickle.dump(xgb_pipe_line, open("xgb.pkl", 'wb'))
xgboost.__version__
import numpy as np

import matplotlib.pyplot as plt
import json
```

```
# creating the dataset

data = {'Random Forest':results[0],'Xgboost':results[1]}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))
colors = ['red', 'green']
# creating the bar plot
plt.bar(courses, values, color =colors,
        width = 0.4)

for index,value in emumerate(values):
   plt.text(value,index,str(value))
plt.xlabel("Algorithm used")
plt.ylabel("Accuracy")
plt.show()
```

CHAPTER-6

TEST CASES

6-TEST CASES

TEST CASE	OUTPUT	STATUS
Comment1: u truly love nature	Identified as anti-bullying	Your comment is posted Successfully
Comment2: Not a great picture Asshole	Identified as bullying	Warning! Your comment is filtered
Comment3: Beautiful pic	Identified as anti-bullying	Your comment is posted Successfully
Comment4: Bullshit	Identified as bullying	Warning! Your comment is filtered
Comment5: looks awesome	Identified as anti-bullying	Your comment is posted Successfully

CHAPTER-7

SCREENSHOTS

7- SCREENSHOTS

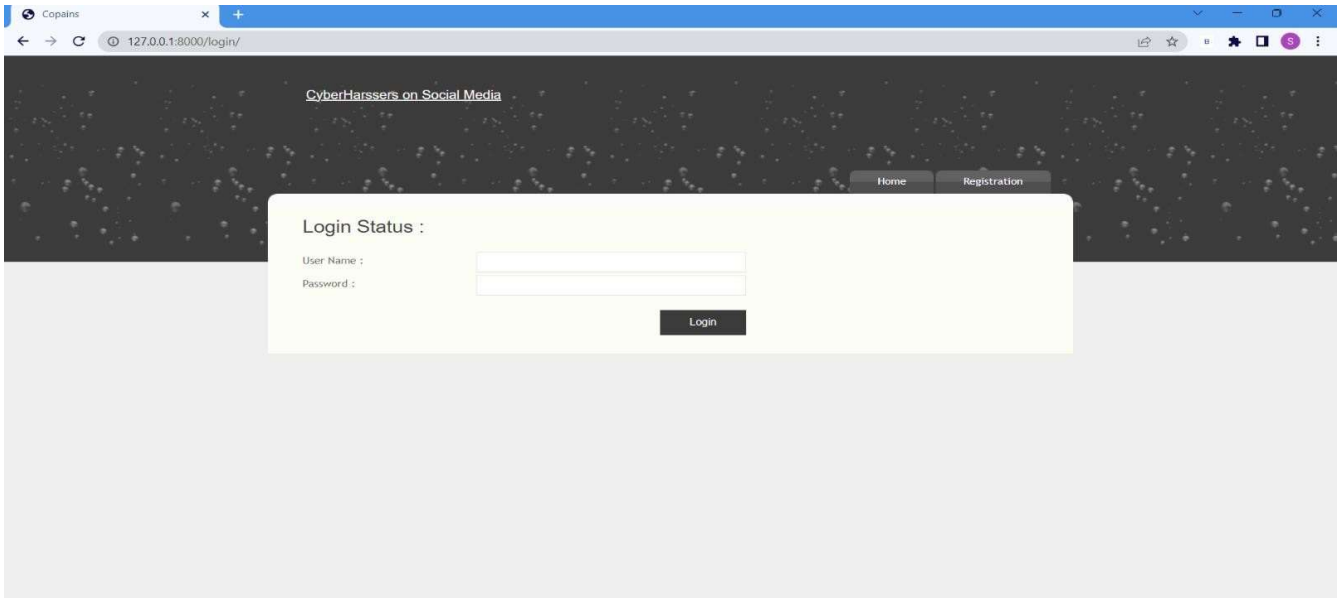


Fig. 7.1: This is the login page to the platform

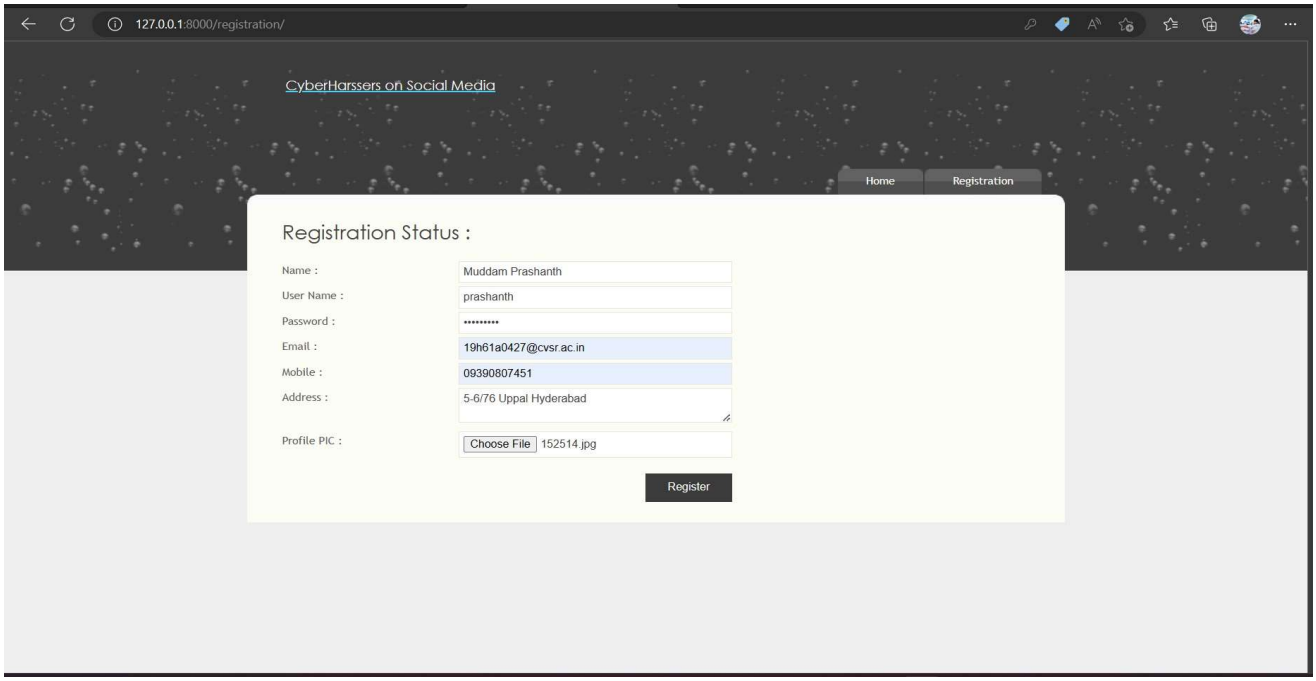


Fig 7.2: It shows the new user registration process

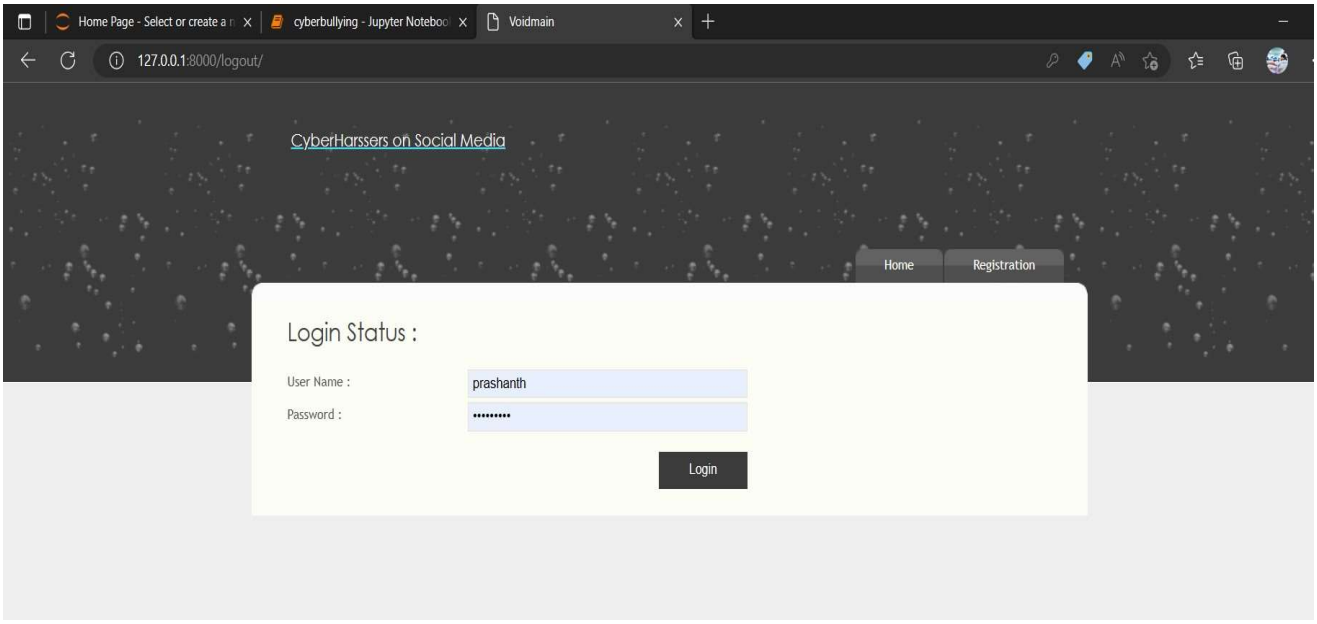


Fig 7.3: This shows that one can login with their credentials

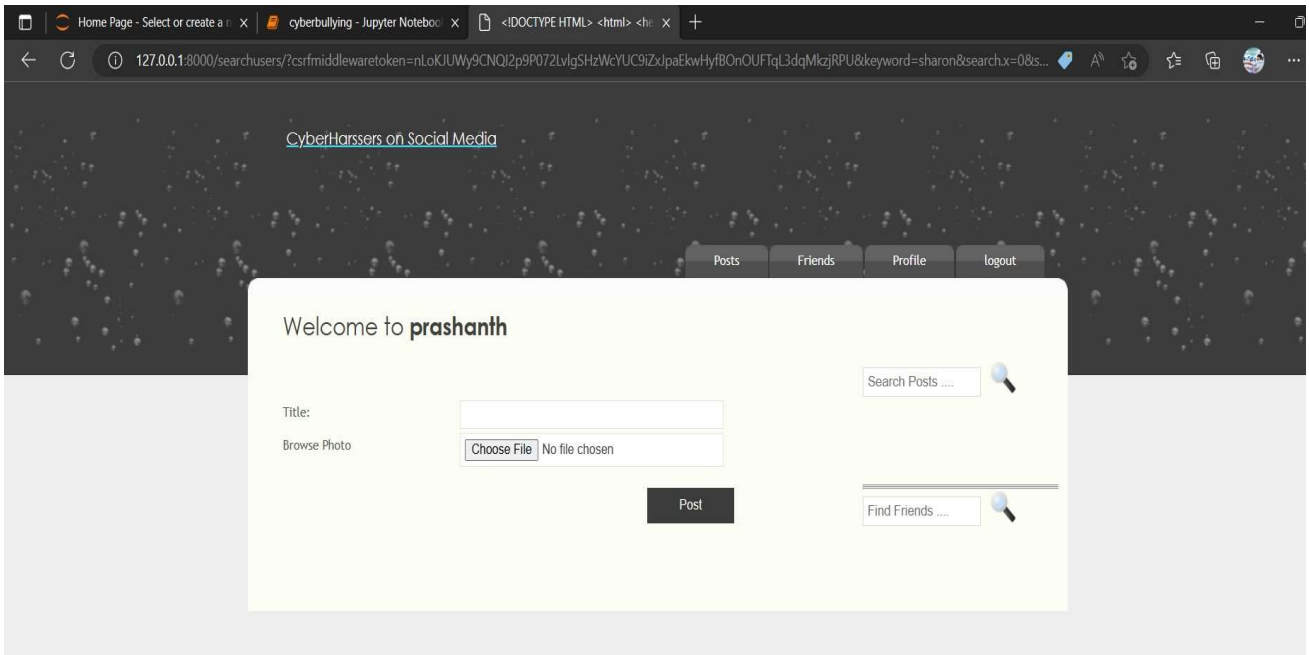


Fig 7.4: This shows the users home page just after logging in

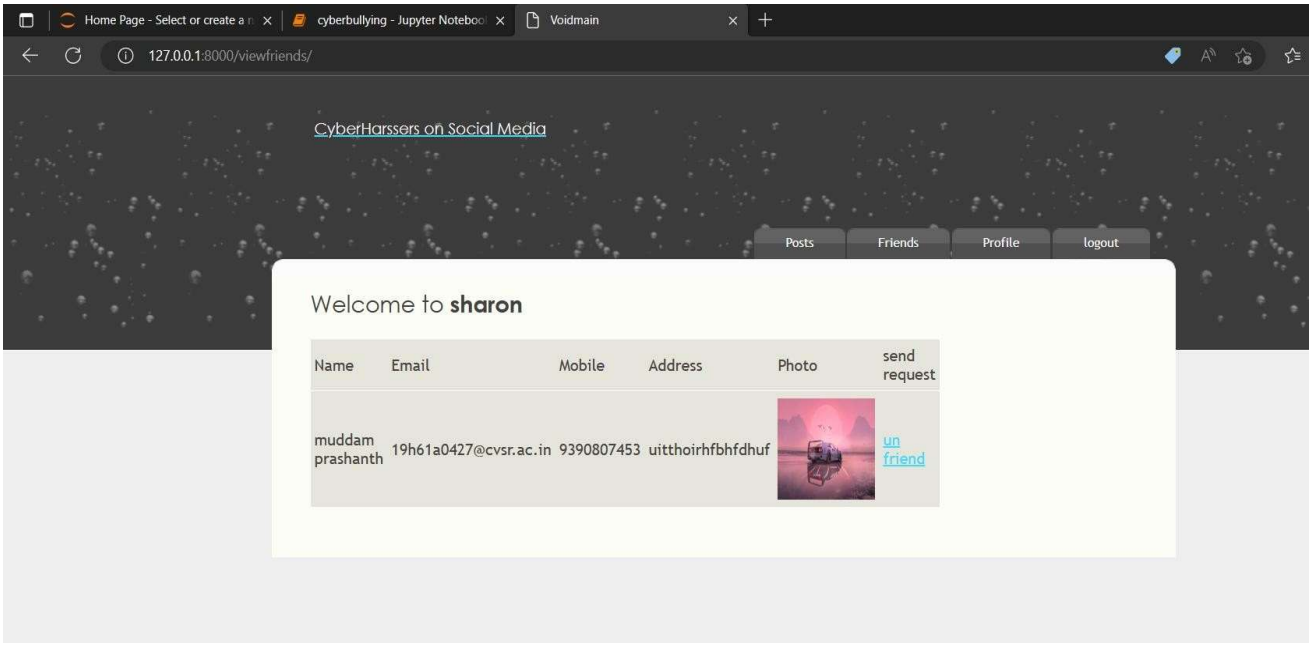


Fig 7.5: This shows the other users web page where one can accept their friend requests

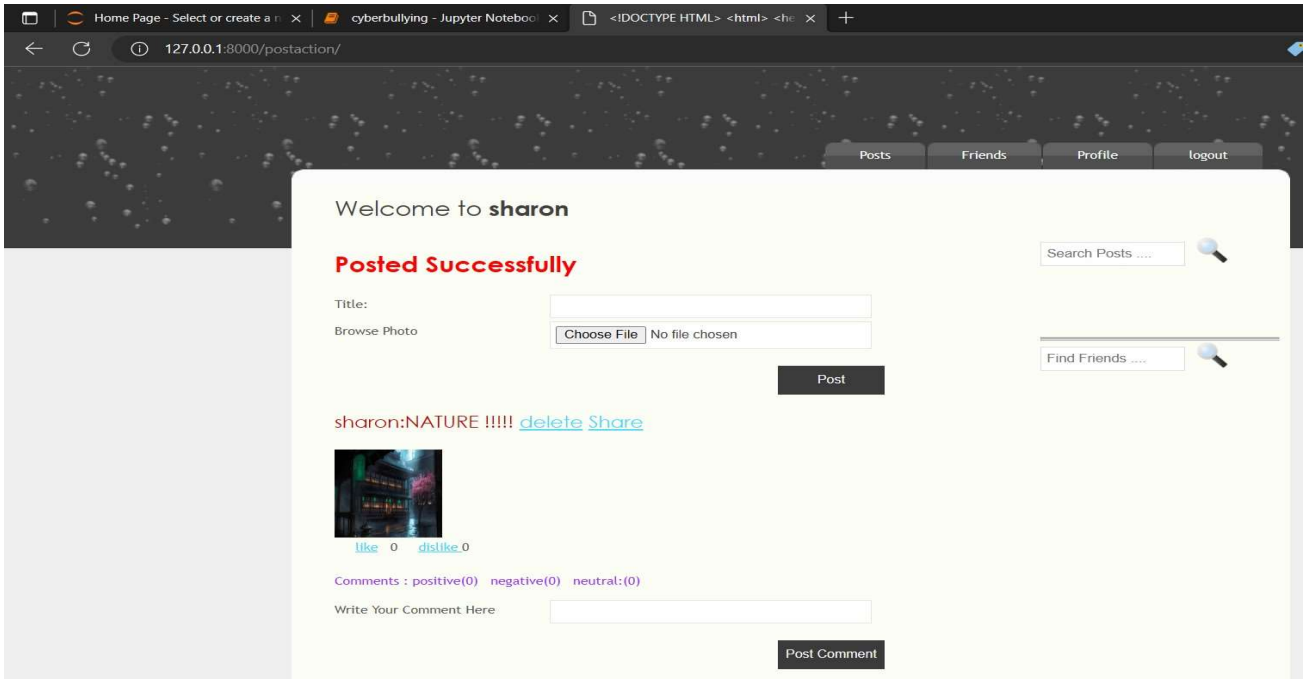


Fig 7.6: This shows the post that a user can upload. Here, user named Sharon has posted a picture with the description Nature!

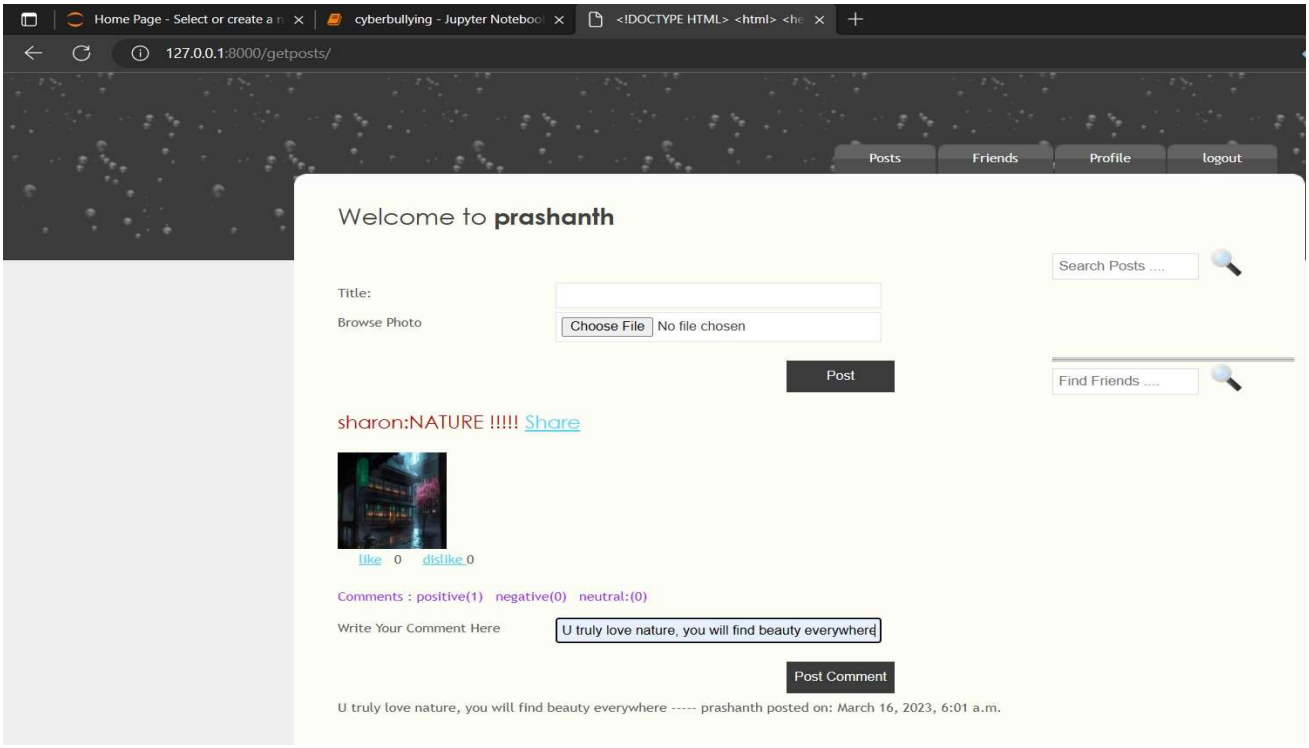


Fig 7.7: This shows the anti-bullying comment that is posted by Prashanth on Sharon's post.

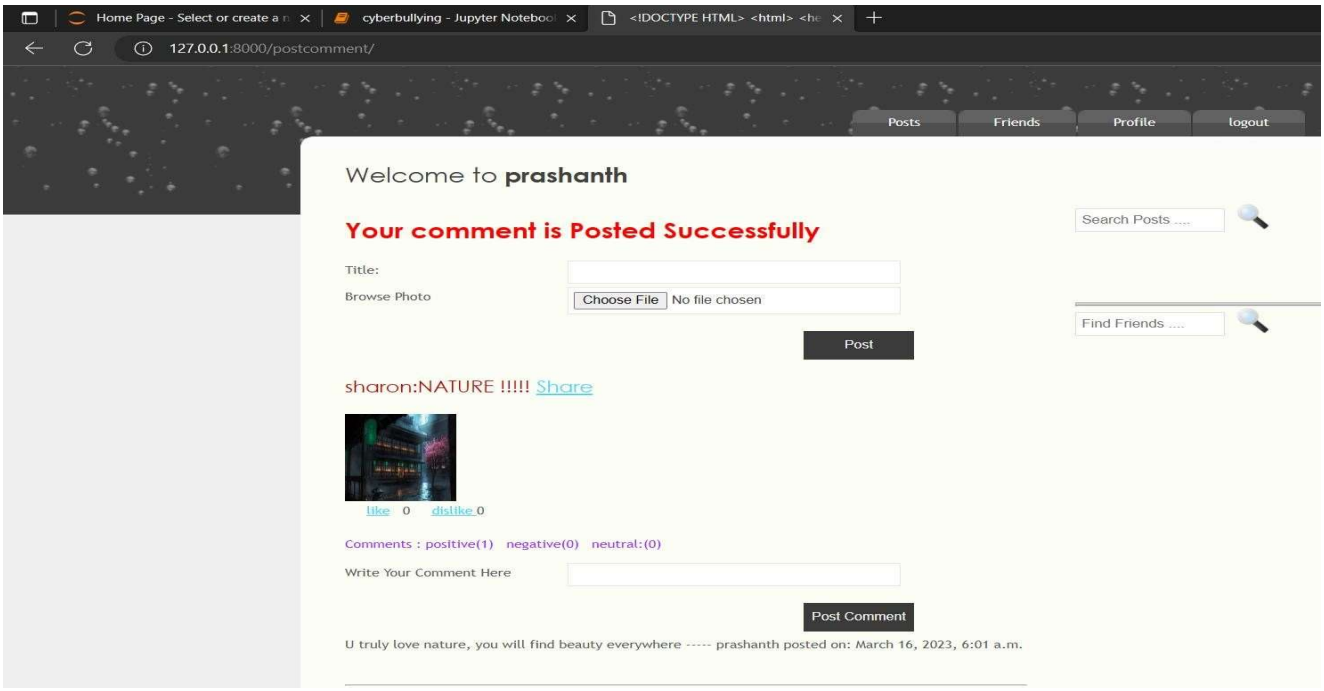


Fig 7.8: This shows that the anti-bullying comment is posted successfully.

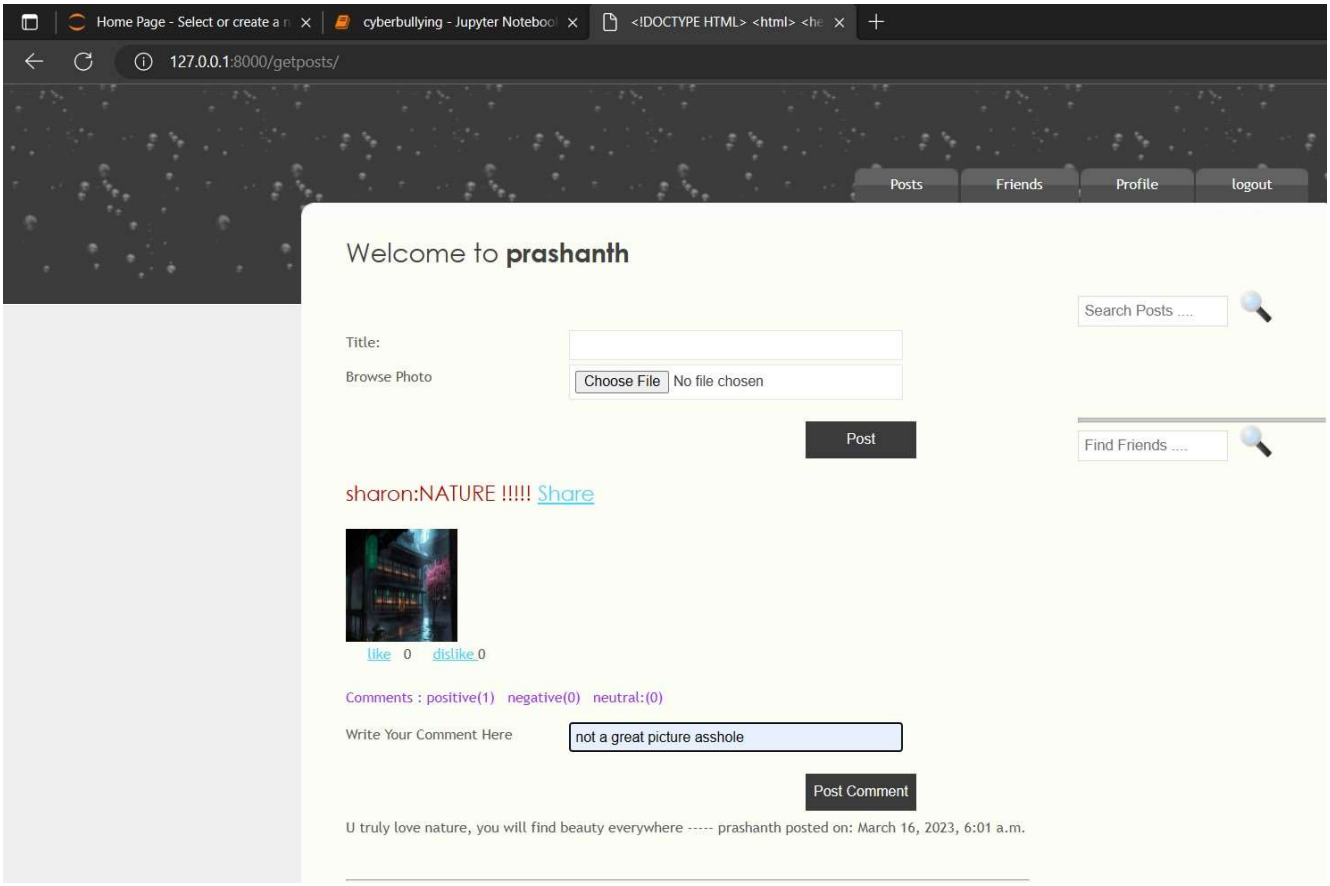


Fig 7.9: This shows that the comment is filtered when a user tries to post a bullying comment.

CHAPTER-8

CONCLUSION

8-CONCLUSION

Alongside the advent of social media platforms and the growing popularity of teenage usage of social media, a new phenomenon known as cyberbullying has developed as a severe societal concern. This problem is still in its early stages of development. In order to reduce the negative impacts of cyberbullying, the development of a programme that can identify instances of cyberbullying automatically is required. Due to the serious nature of the problem, we looked at the possibility of employing BoW and TF-IDF in order to automatically locate posts on social media related to cyberbullying. One of the four different methods of machine learning that are utilised to identify bullying content is called SVM for both BoW and TF-IDF. We have high hopes that in the not too distant future, we will be able to apply deep learning methods to the development of a system that will be able to recognise and classify instances of cyberbullying in written Bengali materials automatically.

CHAPTER-9

FUTURE ENHANCEMENT

9- FUTURE ENHANCEMENT

Now, the prediction is limited to text and can be enhanced further to predict the Cyberbullying from images, videos and emojis. The extent of the prediction can be enhanced to even include more languages.

CHAPTER-10

REFERENCES

10- REFERENCES

[1] Balakrishnan, Vimala & Khan, Shahzaib & Arabnia, Hamid. (2020). Improving Cyberbullying Detection using Twitter Users' Psychological Features and Machine Learning. Computers & Security. 90. 101710. 10.1016/j.cose.2019.101710.

[2] J. Yadav, D. Kumar and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 1096-1100, doi: 10.1109/ICESC48915.2020.9155700.

[3] R. R. Dalvi, S. Baliram Chavan and A. Halbe,"Detecting A Twitter Cyberbullying Using Machine Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 297-301, doi: 10.1109/ICICCS48265.2020.9120893

[4] John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad, Eslam Amer and Ammar Mohammed, "Social Media Cyberbullying Detection using Machine Learning" International Journal of Advanced Computer Science and Applications (IJACSA),10(5),2019.

[5] Monirah Abdullah Al-Ajlan and Mourad Ykhlef, "Deep Learning Algorithm for Cyberbullying Detection" International Journal of Advanced Computer Science and Applications (IJACSA), 9(9), 2018. http://dx.doi.org/10.14569/IJACSA.2018.090927

[6] M. A. Al-Ajlan and M. Ykhlef, "Optimized Twitter Cyberbullying Detection based on Deep Learning," 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, 2018, pp. 1-5, doi: 10.1109/NCG.2018.8593146.

[7] Y. J. Foong and M. Oussalah, "Cyberbullying System Detection and Analysis," 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, 2017, pp. 40-46, doi: 10.1109/EISIC.2017.43.

[8] X. Zhang et al., "Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 740-745, doi: 10.1109/ICMLA.2016.0132.

[9] M. Di Capua, E. Di Nardo and A. Petrosino, "Unsupervised cyber bullying detection in social networks," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 432-437, doi: 10.1109/ICPR.2016.7899672.

[10] S. Parime and V. Suri, "Cyberbullying detection and prevention: Data mining and psychological perspective," 2014 International Conference on Circuits, Power and Computing Technologies

[ICCPCT2014], Nagercoil, 2014, pp. 1541-1547, doi: 10.1109/ICCPCT.2014.7054943.