**A Hybrid Approach for Detecting Cyber Bullying on Social Media Platform using Machine Learning Algorithms**

**A MAJOR PROJECT REPORT**

*Submitted by*

**Burra Sai Krishna Bharadwaj (RA1911033010109)**

**Koteswararao Gutta (RA1911003010621)**

*Under the guidance of*

**Dr. Vegesna S M Srinivasavarma**

(Research Assistant Professor)

in partial fulfilment for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**

****

**DEPARTMENT OF COMPUTING TECHNOLOGIES**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR – 603 203**

**MAY 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR – 603 203**

**BONAFIDE CERTIFICATE**

Certified that this B.Tech project report titled “**A Hybrid Approach for Detecting Cyberbullying on Social Media Platform using Machine Learning Platform**” is the bonafide work of Mr. Burra Sai Krishna Bharadwaj [Reg. No.:RA1911033010109] and Mr. koteswararao Gutta[Reg. No.: RA1911003010621] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

|  |  |  |
| --- | --- | --- |
| SIGNATURE  Dr. Vegesna S M Srinivasavarma  **GUIDE**  Assistant Professor  Dept. of CTech | SIGNATURE  Dr. Pushpalatha M  **HEAD OF Department**  Professor  Dept. of CTech | SIGNATURE  S. Gnanavel  **PANEL HEAD**  Associate Professor  Dept. of CTech |

A picture containing text, font, logo, trademark

Description automatically generated

Annexure II

Department of Computational Intelligence  
**SRM Institute of Science and Technology**  
**Own Work Declaration Form**

**Degree/ Course** **:** B.Tech in Computer Science and Engineering with a

specialization in Software Engineering

**Student Names :** Burra Sai Krishna Bharadwaj, Koteswararao Gutta

**Registration Number:** RA1911033010109, RA1911003010621

**Title of Work :**  A Hybrid Approach for Detecting Cyberbullying on Social

Media Platform using Machine Learning Platform

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc.)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| If you are working in a group, please write your registration numbers and sign with the date for every student in your group. |

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support. We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. Pushpalatha M,** Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr.Vegesna S M Srinivasavarma**, Assistant Professor, Panel Head, **Dr. S. Gnanavel**, Associate Professor in the Department of Computational Intelligence at the SRM Institute of Science and Technology, who guided us throughout the course. We are incredibly grateful to my mentor, Dr. Vegesna S M Srinivasavarma, Assistant Professor in the Department of Computing Technologies at the SRM Institute of Science and Technology, for giving me the chance to work on my project with guidance from her. She gave me the flexibility and encouragement to investigate the academic topics that I was interested in. Her zeal for finding solutions and changing the world has always been motivating. We really appreciate the assistance provided by the faculty, staff, and students of the Computing Technologies Department of the SRM Institute of Science and Technology for our study. Finally, we would like to express our gratitude to our parents, relatives, and friends for their unwavering love, ongoing support, and inspiration.

We sincerely thank the Computing Technologies Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

**Burra Sai Krishna Bharadwaj [RA1911033010109]**

**Koteswararao Gutta [RA1911003010621]**

**TABLE OF CONTENTS**

***C.NO. TITLE PAGE NO.***

[ABSTRACT viii](#_Toc134187610)

[LIST OF TABLES ix](#_Toc134187611)

[LIST OF FIGURES x](#_Toc134187612)

[LIST OF SYMBOLS AND ABBREVIATIONS xi](#_Toc134187613)

[INTRODUCTION 1](#_Toc134187614)

[1.1. General 1](#_Toc134187615)

[1.2. Purpose 2](#_Toc134187616)

[1.3. Scope 3](#_Toc134187617)

[1.4. Machine Learning 4](#_Toc134187618)

[LITERATURE REVIEW 5](#_Toc134187619)

[PROPOSED METHODOLOGY 7](#_Toc134187620)

[3.1. Data Gathering 7](#_Toc134187621)

[3.2. data cleaning 9](#_Toc134187622)

[3.3. Data Preprocessing 10](#_Toc134187623)

[3.4. Feature Extraction and Feature Selection 12](#_Toc134187624)

[3.5. Algorithms 13](#_Toc134187625)

[3.6 System Architecture 32](#_Toc134187626)

[RESULTS 33](#_Toc134187627)

[CONCLUSION 37](#_Toc134187628)

[FUTURE SCOPE 38](#_Toc134187629)

[REFERENCES 39](#_Toc134187630)

[APPENDIX 1 41](#_Toc134187631)

[APPENDIX 2 45](#_Toc134187632)

[PAPER PUBLICATION STATUS 60](#_Toc134187633)

[PLAGIARISM REPORT 62](#_Toc134187634)

# ABSTRACT

Cyberbullying is a growing concern on social media platforms, where users can hide behind anonymity and engage in harmful behavior. Mainly we study, propose a hybrid technique for identifying cyberbullying using various different algorithms. We evaluate the performance of 11 different algorithms, including Logistic Regression, MLP Classifier, SGD Classifier, K-Nearest Neighbors Classifier, Multinomial Naive Bayes, and Bagging Classifier. Our experiments show that Logistic Regression and SGD Classifier achieve the highest accuracy of 90.72% and 90.66% respectively, while Naive Bayes has the lowest accuracy of 56.63%. Our hybrid approach combines multiple machine learning algorithms to increasing precision and robustness of identifying cyberbullying. use a feature extraction technique to take the text input and extract pertinent characteristics, after which to develop ML models using these features. Our results demonstrate that machine learning algorithms can effectively detect cyberbullying on social media platforms, and our hybrid approach can provide a more comprehensive and accurate solution for identifying and preventing cyberbullying incidents.

# LIST OF TABLES

[Table 1 Results of LR, SVC DTC algos 34](#_Toc134187587)

[Table 2 Results of ADA, NB, RFC algos 34](#_Toc134187588)

[Table 3 Results of MLP, SGD, KNN algos 34](#_Toc134187589)

[Table 4 MultinomialNB, Bagging results 35](#_Toc134187590)

# LIST OF FIGURES

[Figure 1 Real life experience of cyberbullying 2](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189384)

[Figure 2 Gathered Data 8](#_Toc134189385)

[Figure 3 Gathered Data 8](#_Toc134189386)

[Figure 4 Logistic regression 15](#_Toc134189387)

[Figure 5 Support Vector Classifier 16](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189388)

[Figure 6 Decision Tree Classifier 18](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189389)

[Figure 7 Ada Boost Classifier 20](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189390)

[Figure 8 Naïve bayes classifier 21](#_Toc134189391)

[Figure 9 Multi-layer Perceptron Classifier 23](#_Toc134189392)

[Figure 10 Random Forest Classifier 25](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189393)

[Figure 11 SGD Classifier 26](#_Toc134189394)

[Figure 12 KNN Classifier 28](#_Toc134189395)

[Figure 13 Bagging Classifier 31](#_Toc134189396)

[Figure 14 System Architecture Diagram 32](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189397)

[Figure 15 Bar graph of comparing accuracies. 36](file:///C:\Users\ADMIN\Downloads\Major%20project%20final%20report.docx#_Toc134189398)

# LIST OF SYMBOLS AND ABBREVIATIONS

|  |  |
| --- | --- |
|  |  |
| **ML** | Machine Learning |
| **NLP** | Natural Language Processing |
| **NLTK** | Natural Language Toolkit |
| **GPU** | Graphics Processing Unit |
| **RAM** | Random-Access Memory |

**CHAPTER 1**

# INTRODUCTION

## 1.1. General

As social media usage continues to rise, so does the prevalence of cyberbullying - a form of online harassment that can have devastating effects on its victims. Detecting cyberbullying on such a vast platform can be a challenging task, as it requires distinguishing between derogatory, provocative, or sensitive messages from genuine conversations. ML approaches have become more popular in recent years. to automate this process, but they are not without their limitations. False detections are still a common occurrence, and it can be challenging to distinguish between general profanity and hate speech. Certainly. One real example of cyberbullying that received widespread attention is the case of Kangana Ranaut, an actress and comedian who starred in the 2016 all-female Ghostbusters film. Kangana was the target of a coordinated campaign of harassment on Twitter, which included racist and sexist insults, as well as the posting of her personal information and explicit photos.

Kangana received a barrage of abusive tweets, causing her to temporarily leave Twitter due to the overwhelming negativity. The cyberbullying campaign against her sparked widespread outrage and led to a discussion about the toxicity of social media and the need to take action to prevent cyberbullying. In this case, Kangana Ranaut is the victim of cyberbullying. The abusive tweets and messages sent to her caused her emotional distress and anxiety. The perpetrators of the cyberbullying campaign used Twitter as a platform to harass and intimidate Kangana, causing her to feel unsafe and attacked. The campaign also had a negative impact on the broader community, as it highlighted the need for greater awareness and action to prevent cyberbullying. Overall, the case of Kangana Ranaut demonstrates the harmful effects of cyberbullying and the need for individuals, venues for social media and society at large to take steps to prevent it. It is important to recognize that Cyberbullying is a significant issue that can negatively impact a victim's mental health and general wellbeing.Regardless of age, gender, or background, anyone can fall victim to cyberbullying on social media, messaging apps, or online forums. It is important for both individuals and organizations to take proactive steps to combat cyberbullying and to assist individuals who have been victimised by it. involve promoting responsible online behavior and safety, promptly reporting instances of cyberbullying, and providing resources such as counseling for those who have experienced emotional distress or anxiety as a result of it.

 In order to overcome these difficulties, we provide a hybrid method for identifying cyberbullying in this research article. Our strategy involves leveraging a manually annotated open-source dataset to apply several supervised classification techniques to user-provided inputs. We conduct a systematic assessment of the literature on cyberbullying detection methods, looking at techniques for identifying hate speech in social media while separating it

Figure Real life experience of cyberbullying

from other types of vulgarity. By utilizing different algorithms, for this purpose, we seek to establish lexical baselines. Our hybrid approach aims to increase the precision and effectiveness of cyberbullying identification and minimize the risk of false detections. By identifying and addressing the shortcomings of current approaches

## 1.2. Purpose

The rise of social media has brought about many benefits such as increased connectivity, information sharing, and community building. However, it has also given rise to new forms of harassment and bullying, known as cyberbullying. Bullying through internet interaction, harass, or intimidate is known as cyberstalking, and it can have major negative implications for one's mental health and well-being. Given the prevalence and severity of cyberbullying, It is critical to create efficient detection and measures preventing such behaviour.

Machine learning has shown great promise in detecting cyberbullying on social media platforms. However, no single approach has been found to be completely effective, and it is still a challenging problem to accurately identify and classify cyberbullying. This is due to the complex and dynamic nature of online conversations, the use of sarcasm and irony, and the difficulty in identifying subtle cues that may indicate cyberbullying.

Therefore, the purpose of this project is to develop a hybrid approach that combines multiple ML techniques in order to detect cyberbullying on social media networks. The hybrid technique proposed in this project will leverage the strengths of multiple ml techniques, such as NLP, sentiment analysis, and deep learning, to improve the accuracy and efficacy of cyberbullying detection.The project will involve collecting and analyzing social media data to training and testing hybrid approach. hybrid approach is compared with existing approaches, and the results will be evaluated using standard evaluation metrics. The project's ultimate goal is to provide an effective and efficient method for identifying cyberbullying that can be implemented into social media platforms in order to foster a safer and more inclusive online environment.

Finally, the motive of this investigation is to succeed a hybrid strategy that combines ml algorithms to identify cyberbullying on various social networking sites. By creating an effective method for detecting cyberbullying the project's goal is to contribute to the creation of a more secure and inclusive social media ecosystem. The project's successful execution could have a substantial influence on the lives of those impacted by cyberbullying and help to ongoing efforts to tackle the problem.

## 1.3. Scope

The intention of this research is to evolve and estimate a hybrid approach for identifying cyberbullying on social media sites using machine learning algorithms. The following aspects will be prioritised in the project:

1. Data Collection: The project will involve the collection of social media data from multiple sites, including Twitter, Facebook, and Instagram. APIs and web scraping techniques will be used to acquire the data.
2. Data Preprocessing: Preprocessing will be performed on the gathered data to remove noise, unnecessary information, and duplicate data. The preprocessed data will be converted into an analysis-ready format.
3. Machine Learning Algorithm Selection: The project will explore purpose for multiple ml algorithms for the identification of abuse, including nlp, sentiment analysis, and deep learning. The techniques will be selected based on their effectiveness in detecting cyberbullying and their suitability for the data.
4. Feature Extraction: The project will involve The extraction of useful information from social media data. The characteristics will include linguistic features such as the use of abusive language, sarcasm, and sentiment, as well as network features such as user interactions and network structure.
5. Model Development: The project will involve the development of a hybrid approach that combines multiple machine learning algorithms for cyberbullying detection. The approach will be designed To enhance the accuracy and efficacy of cyberbullying identification.
6. Model Evaluation: The research will assess the hybrid approach's performance using common assessment criteria accuracy, memory, and scores for F1 are a few examples. To assess technique's efficacy, its performance is going to be compared to that of existing techniques.
7. Implementation: project will involve the implementation of the hybrid approach into social media platforms to foster a more secure and inclusive online community. The implementation will involve collaboration with social media platforms to integrate the approach seamlessly.

The aim of this research is limited to the creation and testing of a hybrid strategy for detecting cyberbullying on the internet and social media platforms. project does not aim to develop solutions for preventing or addressing cyberbullying incidents. The initiative is intended to make a contribution to the development of effective methods for detecting cyberbullying and provide insights into utilise ml algorithms for cyberbullying detection.

## 1.4. Machine Learning

Machine Learning (ML), though often confused and interchangeably used with Artificial Intelligence, constitutes a subfield of computational intelligence focused on the science in creating machine-intelligent algorithms that help the machine learn from existing and past data. The algorithms achieve this by identifying and learning the various relationships between individual features of the data, looking for common patterns, responding to different situations outside of their programming restrictions and makes predictions accordingly. Machine Learning can be broadly classified into three categories (figure 1.4.1) based on the kind of data they use to learn and the type of outputs produced by them. They are:

Supervised Learning: In this type of learning, the datasets that the machine utilises to learn are entirely structured and labelled. The datasets have a set of input features and their corresponding outputs and are given to the ML algorithm as inputs for training. While training, the model learns by mapping its predictions to the true predictions present in the dataset, evaluates its performance, and automatically updates its learning curve to achieve better scores of the performance metrics. For example, classification and regression tasks such as image classification and commodity price prediction respectively.

Unsupervised Learning: In this type of learning, the datasets are not mapped with their outputs. The models are trained on a dataset consisting of several features and are expected to produce outputs by learning and identifying common patterns without conforming to a certain “correct answer”. For example, clustering algorithms used for customer segmentation, DNA pattern recognition and grouping in evolutionary biology.

**CHAPTER 2**

# LITERATURE REVIEW

Numerous methods have been developed for the purpose of detecting cyberbullying, with the majority of these methods relying on Retrieving information and processing natural language. These methods have obtained impressive accuracies.

Sambhagadi et al. [1] uses NLP approaches to try and find solutions to identify and eventually stop cyberbullying on social media. different methods applied to determine whether sacrilege are used to offensive. Crowdsourcing and lab annotations are utilized to iteratively modify the annotations used in the paper. input gathered from English-language posts on social media sites .it made possible to crawl efficiently. SVM with modifications was utilized to categorize the data, and in addition to inappropriate words that could have been missed, features like question-and-answer posts and emoticons were also taken into account. Finally, the F1-Score was determined to be 0.59 (which, while smaller than the Kaggle winner, nevertheless appears promising given that this study did not employ personalized data and a fresh and superior dataset).

Difficulties in this study included –

• With questions and answers paired together in a way that they are not in other datasets, and both questions and answers may just comprise a single thing makes challenging identify them not knowing whole thing.

• On social media, people utilize slang and casual language, it has numerous typos and abbreviations it exceedingly challenging to process them.

A Procedure is suggested by R. R. Dalvi et al. [2] using different algorithms. study, tweets are collected and datasets are created using the live Twitter API. Method contrasts using different algorithms in the collected. The output shows Naive Bayes was close to 52.75%, while final percentage got is 71.25%.Silva et al. [3] suggested a strategy for identifying cyberbullying based on psychological research; it details the design of an app called Bully Blocker, which seeks to alert the user's parents if cyberbullying is discovered. It employs conventional techniques to evaluate the user's social media data by looking classifying bullying indicators. It is designed primarily for teenagers and uses antiquated Facebook detection techniques develop serving gathering machine learning categorization may be applied.

Peter Burnap et al. [4] utilized a technique based on dictionaries to detect cyberhatred on Twitter. The approach involved creating numerical vectors using an N-gram feature engineering method that relied on a pre-established vocabulary of aggressive language. When the generated numeric vector was fed into ML F-score was 67%. Stéphan Tulkens et al [6].'s likewise used a different approach. authors employed word across a characteristic. supplied SVM classifier with generated characteristics. findings yielded an F-Score of 0.46 in online was classified an ML by Njagi Dennis et al. [5]. In 2021, Kumar et al. [7] released their research on how users behave online when interacting or submitting something publicly. This research aims to identify cyberbullying in tweets from Twitter. They have analyzed people's behaviors using machine learning methods.

Singh et al. [8] study on how recognize cyberbullying using different approaches. publication examined research on cyberbullying from 30 different researchers. They conducted research on methods for identifying and stopping cyberbullying, and they also provided users with some instructions for using platforms. A small number of researchers have employed deep learning techniques automatically identify recent instances. instance, Karthik Dinakar et al. [9] categorized delicate subjects in comments or postings on social media. They developed that construct numerical features for their study. obtained different classifiers. It is outperformed classifiers by obtaining 73% accuracy, according to their trial results. Shuhua Liu et al. [10] divided web content pages into categories of violence and hatred. In their analysis, they used trigram properties, which are symbolized by TFIDF. The authors made use of the different classifiers achieved a maximum output 69%.

**CHAPTER 3**

# PROPOSED METHODOLOGY

In this part, we will go through the dataset provided was used, as well as the recommended solution architecture for the identification of offensive behaviour and the algorithms used to detect the set of bullying involved in the twitter. To train the machine learning algorithms, a dataset containing over 24000 tweets was created. The Twitter was used to collect tweets. Twitter API and then pre-processed to eliminate any unnecessary information. NLP To clean the data and eliminate any stop words, procedures were utilised, punctuation marks, and URLs.

The next step involved feature extraction and selection. Various features such as n-grams, emoticons, and hashtags were extracted from the tweets. These features were then selected based on their relevance to the detection of cyberbullying. Selecting Feature’s is an very critical step as it reduces the dataset's depth, doing it less difficult for the algorithms to process. After feature selection, different machine learning algorithms were implemented on the existing dataset to evaluate the great possible model for identifying cyberbullying. The algorithms implemented in our research included Naive Bayes, Support Vector Machine, and other popular classifiers. These algorithms were judged based on their precision, accuracy, recall, and F1 score.

## 3.1. Data Gathering

Gathering data is a very important step in any research, because the accuracy and solidity of the results can be substantially influenced by the standard of the data collected for discovery. The process of collecting data involves recognizing the relevant variables and sources of information, and then gathering and organizing that information in a systematic and consistent manner. This can involve a range of techniques and methods, such as surveys, interviews, observation, and data mining. The quality of the data collected sample size, representativeness sample, and the validity and reliability of the measurement tools. Researchers must also be mindful of potential biases or errors that could impact the accuracy of the data, such as social desirability bias or measurement error. As data collection techniques and technologies continue to evolve, researchers have access to an increasing amount of data from diverse resources, like social network platforms and online forums. However, this also raises new ethical and legal concerns regarding the privacy and security of personal information. Overall, data collection is a critical component of the research process that requires careful planning, attention to detail, and ethical considerations to guarantee that the data is gathered accurately, reliable and meaningful. Tweets were streamed using the Twitter API using roughly 32 keywords related to online bullying, and these tweets make up the input dataset. These are just a handful of the terms mentioned in psychiatric literature [11], [12], and [13]: ni\*\*er, junk, donkeys, terrified, stupid, pretender, raped, fuck etc. On other side, [14] advised against using terms like "kill," "death," "dirty," "hatred,”. The initial dataset consists of 435764 records, including 130000 tweets based on terms related to racism, insults, swear words, and sexism. There are a lot of outlier tweets in this sample. As only tweets in the English language are required, tweets including other language words are eliminated, and retweets are filtered. The final dataset is created by randomly choosing roughly 10,000 tweets from the remaining tweets after these kinds of irrelevant tweets have been removed. All these procedures are carried out automatically as part of the pre-processing stage. In general, data gathering involves collecting a large amount of relevant data from various sources, including databases, APIs, or manual data entry. The data collected should be representative of the problem the model aims to solve and should be diverse enough to cover the range of scenarios that the model will be expected to handle as shown in figure 2.

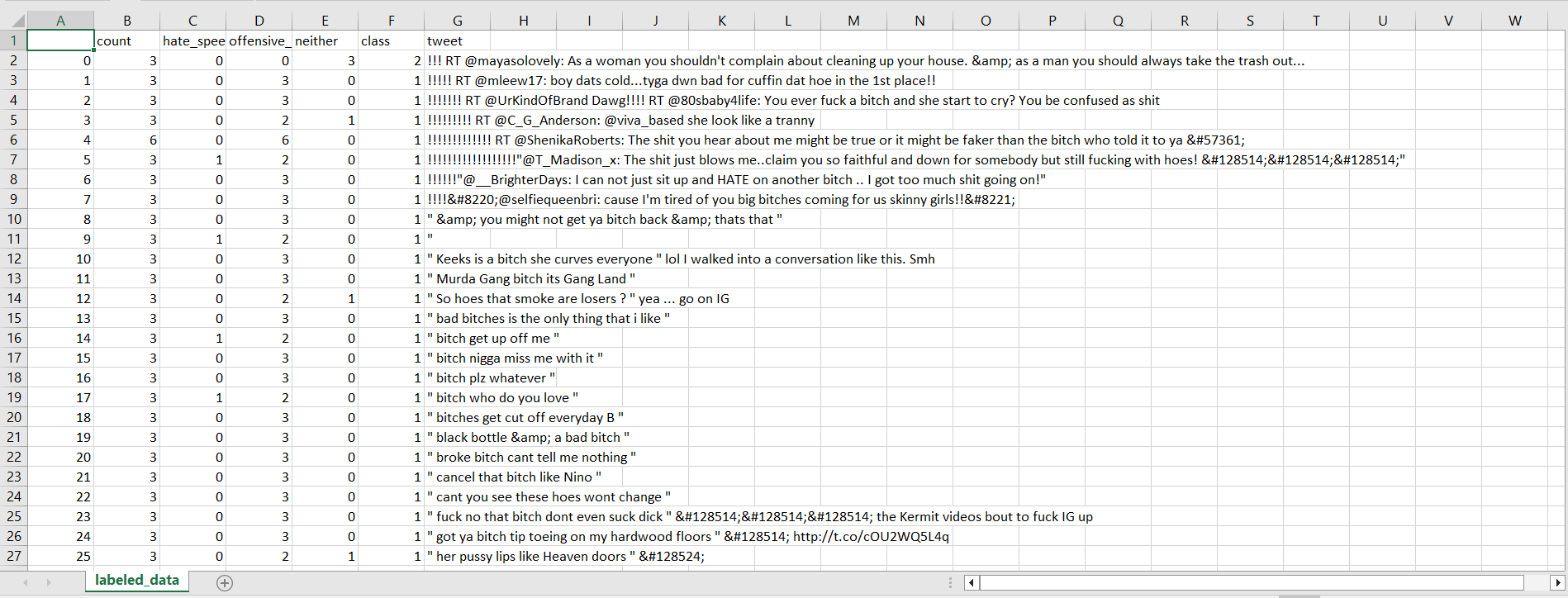


Figure Gathered Data

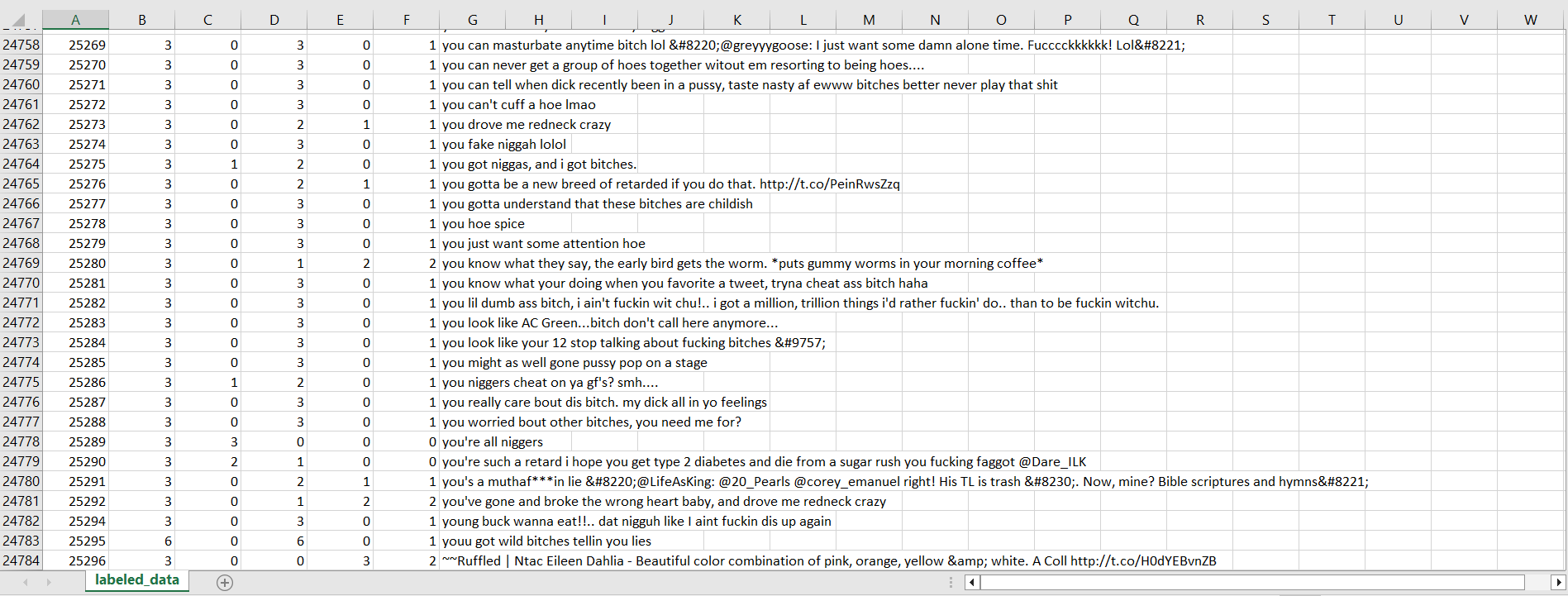


Figure Gathered Data

## 3.2. data cleaning

The rise of social media is resulted in the generation of huge amounts of information that can be used to gain insights into various aspects of society. Among the most essential sources of information in social network communication is Twitter, which enables users to send and receive brief messages known as tweets, that can be analyzed to gain insights into people's opinions, attitudes, and behaviours. However, the accuracy and quality of the data are critical to ensure that the analysis provides meaningful insights. We shall discuss the many strategies employed in this study for data cleaning of Twitter tweets datasets to ensure accuracy and quality.

Irrelevant Data: The presence of irrelevant data in Twitter tweets datasets can obscure the valuable content of tweets and make it challenging to analyze the data accurately. Examples of irrelevant data include URLs, hashtags, and user mentions. URLs are links to external web pages that can be included in tweets to provide additional information. However, they are often unrelated to the content of the tweet and can be removed without affecting the accuracy of the analysis. Hashtags are keywords preceded by the # symbol that are used to categorize tweets and make them easier to find. While hashtags can be useful, they can also be overused and irrelevant to the content of the tweet. Removing hashtags that are unrelated to the tweet can improve the accuracy of the analysis. User mentions are references to other Twitter users that can be used to direct a tweet to a specific candidate or assemblage of persons. While user mentions can be relevant to the content of the tweet, they can also be used for spamming or trolling. Removing user mentions that are not relevant to the tweet can improve the accuracy of the analysis.

Missing Data: Incomplete tweets can pose a significant challenge to the accuracy of the analysis. Imputing missing values or removing incomplete tweets altogether are common approaches used to handle missing data in Twitter tweets datasets. Missing value imputing entails guessing the missing data based on the values of other variables in the dataset. For example, if the tweet's text is missing, it can be imputed based on the hashtags or user mentions in the tweet. Removing incomplete tweets altogether can reduce the size of the dataset but can also result in the loss of valuable information.

Duplicates: Identifying and removing duplicates can help to reduce bias in the analysis and ensure that each tweet is represented only once in the dataset. This is particularly important in Twitter tweets datasets, where retweets and duplicate posts are common. Duplicate tweets can be identified based on their text, time stamp, or other variables. Removing duplicates can improve the accuracy of the analysis and make the dataset more manageable.

Normalizing Text: Normalizing text is a crucial step in data cleaning that involves standardizing slang, abbreviations, and misspellings to ensure that each tweet is represented accurately. For example, the word "cuz" can be standardized to "because," and the abbreviation "lol" can be standardized to "laugh out loud." Normalizing text can increase the reliability of the analysis and lower amount for noise in the dataset.

Stop Words: Stop words are often used terms that add little value to the analysis, such as "and," "the," and "it." Getting rid of stop words can drastically cut down on the amount of data that needs to be examined and improve the accuracy of the analysis. Stop words can be removed based on a predefined list of words or based on their frequency in the dataset.

Tokenization: Tokenization is another critical step in data cleaning that involves breaking down the text into individual words or phrases, enabling machine learning algorithms to analyze the text at the word level. By tokenizing the text, machine learning algorithms can list the most typical terms and expressions used in Twitter tweets, allowing them to analyze the sentiment and context of each tweet. Tokenization can be done using various techniques, such as whitespace token

## 3.3. Data Preprocessing

Data preparation is a critical step for machine learning along with information analysis. It entails cleaning and transforming raw data into a format that ml algorithms can understand. It improves accuracy of results by removing not important noise, errors, and outliers from the existing dataset. In this essay, we will discuss the importance of data preprocessing and its various techniques in the context of our project, which is an hybrid approach for detecting cyberbullying ml algorithms are used on communication networks. Preprocessing data is a predominant stage in ML because it helps to the correctness of the algorithms and the data quality. Raw data often contains missing values, noise, outliers, and inconsistencies that can influence the perfection of machine learning algorithms. By preprocessing the existing data, we can remove these unwanted elements and create a clean dataset that is easier for machine learning algorithms to understand and analyze.

In our project, data preprocessing is particularly important because we are dealing with social network data, which is known to be noisy and unpredictable. Social media data is often unstructured and contains a mix of different types of data, such as text, images, and videos. Preprocessing can help us to extract relevant features from this data and create a clean dataset that we can use to train and test our machine learning models.

There are several techniques used in data preprocessing. Here in the existing context, we will debate some of the most commonly used methods in the context of our project.

### Data Cleaning:

The practise of deleting or correcting erroneous, incomplete, or irrelevant data is known as data cleaning. In our project, we will use data cleaning techniques to remove irrelevant data, such as advertisements, spam, and non-English tweets, from the dataset. We will also correct any errors in the data, such as misspelled words, using text correction algorithms.

### Feature extraction:

Feature extraction includes detecting the most relevant characteristic in the dataset that can be used to detect cyberbullying. In our project, we will use feature extraction techniques to identify features such as the use of hate speech, foul language, and derogatory terms that are commonly associated with cyberbullying. We will use to extract these qualities from text data, apply natural language processing algorithms.

### Data transformation

Data transformation is the process of turning data into a format that machine learning algorithms can utilise. In our project, we will use data transformation methods like one-shot encoding and vectorization to transform the text data into a integral format that can be utilized for training and testing machine-learning models

### Data normalization

Data normalization is the process of scaling the data to ensure that all features have the same impact on the machine-learning algorithms. In our project, we will use data normalization techniques to ensure that all features are given equal consideration in the analysis. This can help to prevent bias and improve the accuracy of the results.

### Data reduction

Data reduction entails lowering the amount of a dataset in order to make it more manageable for machine learning algorithms. In our project, we will use data reduction techniques such as dimensionality reduction and sampling to minimise the amount of the dataset without sacrificing quality important information.

InConclusionData preprocessing is an major pace in machine learning and data analysis. In our project, we will use data pre-processing techniques such as data cleaning, feature extraction, data transformation, data normalization, and data reduction to create a clean dataset which can be used to train and validate machine learning models. By pre-processing the data, we can remove unwanted noise, errors, and outliers from the dataset, and improve the accuracy of the results. With the right data pre-processing techniques, we can develop accurate and effective Models of ML for identifying online bullying on social networking platforms.

## 3.4. Feature Extraction and Feature Selection

In the field of machine learning and data analysis, extraction and selection of features are important techniques used to recognize and bring out the most relevant information from a dataset. Both techniques are used to reduce the complexity of a dataset and to make machine learning algorithms' lives simpler to analyze and process the data. In this essay, we will discuss feature extraction and feature selection in detail, and explore their applications in various fields.

Feature extraction is the procedure of identifying and extracting the most important attribute or attributes from a dataset. The purpose of feature extraction is to reduce the complexity of a dataset and to identify the most relevant information that can be used for analysis. The process of translating raw data into a set of features is known as feature extraction that can be effortlessly analyzed andcultivated by machine learning algorithms.

There are several techniques used for feature extraction, including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), and others. Principal Component Analysis is a common strategy used for feature extraction in machine learning, which involves determining the linear combinations of factors responsible for the most variance in the data. LDA is another technique used for feature extraction, which includes recognizing the straight amalgamation of variables that finest separate the classes in the data. ICA is used for blind source separation, and involves identifying the independent components in a dataset.

In the context of cyberbullying detection on social network platforms, feature extraction can be used to identify the most important characteristics that may be utilised to differentiate between normal communication and cyberbullying. For example, features such as the use of foul language, hate speech, and derogatory terms can be used as indicators of cyberbullying.The process of picking a subset of characteristics from a dataset that are most relevant to the situation at hand is known as feature selection. The goal of feature selection is to minimise dataset complexity and increase the accuracy of machine learning algorithms. Feature selection involves identifying and removing irrelevant and redundant features from a dataset.

The selection will be important aspect of ML, and there are several methods employed for this purpose. These techniques include filtering techniques, wrapper and embedded methods. In filter techniques, features are classified based on their statistical significance, and the top-ranked features are chosen for analysis. On the other hand, wrapper methods select subgroups of features based on their result on a machine learning algorithm. Embedded methods, on the other hand, select features based on their contribution to the performance of a machine learning algorithm. All of these methods are useful for selecting the most relevant features for analysis, and they can help to upgrade the accuracy and efficiency of ml.

Cyberbullying detection on social network platforms, feature selection can be used to identify the most relevant attribute that can be used to differentiate among normal communication and cyberbullying. For example, features such as the use of emoticons and hashtags may not be relevant for identifying cyberbullying and can be removed from the dataset using feature selection techniques. Applications of Extraction and Selection of features are widely used in different applications like nlp, bioinformatics. In CV, the process of identifying and extracting the most essential elements from photographs, such as edges, textures, and colours, is known as feature extraction. The process of selecting the most relevant features for image classification tasks like as object identification and face detection is known as feature selection.

In NLP, feature extraction is used in finding and extract most valuble things from text data, like word frequency, sentiment, and part of speech. Selecting attributes is used to select the most relevant attribute for text classification missions, like analyzing sentiments and spam detection.

In bioinformatics, feature extraction and feature selection are help in identify, take most important things from genetic input like gene expression, single nucleotide polymorphisms (SNPs), and protein sequences. Feature selection is used to select the most relevant attributes for disease diagnosis and drug discovery.

In conclusion, feature extraction and feature selection are important methods used in machine learning and data analysis to identify and extract.

## 3.5. Algorithms

In our model we used 11 different algorithms to find the best one in identifying cyberbullying on twitter social media platform.

### 3.5.1 Logistic Regression Algorithm for identifying Cyberbullying

Logistic Regression is normally used statistical model for classifying tasks in machine learning. It is a supervised learning algorithm that is widely used for binary classification tasks. In our project on detecting cyberbullying on social media platforms using machine learning algorithms, we have utilized Logistic Regression as one of the classification algorithms.

Logistic Regression predicts the likelihood of a binary result based on input factors. It is similar to linear regression in that it predicts a probability between 0 and 1 rather than a continuous output. This possibility is then transformed to a binary output by choosing a verge value, typically 0.5. If the predicted probability is greater than the threshold value, the output is considered as 1, otherwise, it is considered as 0. Selecting a feature plays an vital part in the performance of Logistic Regression. It involves selecting the most relevant features that contribute significantly to the classification task while removing irrelevant or redundant features that may negatively impact the performance of the algorithm.

In our project, we have used feature extraction techniques such as TF-IDF to extract the most important features that can identify cyberbullying. The extracted features include the presence of hate speech, the use of foul language, and derogatory terms. These features are then used as input variables for the Logistic Regression algorithm.

The dataset will then be divided into training and testing sets. The training set is used to train the Logistic Regression model, while the testing set is used to evaluate the model's performance. For splitting the dataset, we chose a 70/30 split, with 70% of the data used for training and 30% used for testing. The Logistic Regression model is trained using the training set, where it learns the relationship between the input variables and the binary output. The model parameters, such as the coefficients and intercept, are learned using the maximum likelihood estimation technique.

The training set is used to estimate the model once it has been trained. Metrics including as accuracy, precision, recall, and F1 score are used to assess the model's performance. The accuracy statistic assesses the model's overall performance, whereas precision assesses the fraction among all good forecasts. The proportion of correct positive forecasts out of all actual positive events is calculated as recall. The F1 score is a fair assessment of the model's performance since it is the harmonic mean of accuracy and recall. In our project, we have achieved an accuracy of 86% using Logistic Regression. This signifies that 86% of the cases in the testing set were properly categorised by the model. The accuracy, recall, and F1 score were similarly high, showing that the model worked well in detecting cases of cyberbullying.

In conclusion, Logistic Regression is a powerful algorithm for binary classification tasks such as detecting cyberbullying on social media platforms. It works by guessing the possibility of a binary outcome depend on input variables and is trained using the maximum likelihood estimation technique. Feature selection plays an crucial part in the performance of Logistic Regression, and we have utilized feature extraction techniques such as TF-IDF to extract the most important features for identifying cyberbullying. The performance of the Logistic Regression model is evaluated using metrics such as accuracy, precision, recall, and F1 score, and we have achieved high values for these metrics in our project.

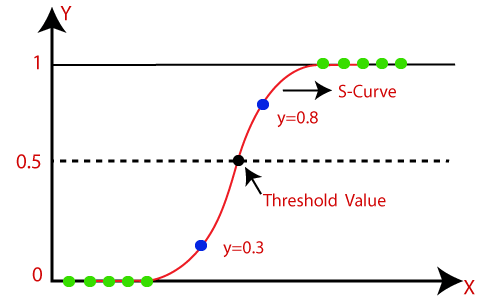


Figure Logistic regression

### 3.5.2 Support Vector Classifier Algorithm for identifying Cyberbullying

Support Vector Classifier (SVC) is a machine learning supervised learning method used for categorization problems. It operates by generating a hyperplane that divides data point classes in a high-dimensional space. The purpose of SVC is to locate the hyperplane that maximises the gap between the classes, resulting in the best separation feasible. In our project of identifying cyberbullying on social media platforms, we have implemented the SVC algorithm as one of the classifiers for our machine learning model. The SVC algorithm works well for our project because it is effective in dealing with datasets that have a huge number of attributes and can handle non-linearly separable data.

One of the key advantages of the SVC algorithm is its ability to handle non-linearly separable data. In our project, we have used the SVC algorithm to classify tweets as either cyberbullying or non-cyberbullying. Since tweets can contain complex language and use of sarcasm, it can be difficult to separate the two classes using a simple linear boundary. The SVC technique circumvents this problem by translating the data into a higher-dimensional space where it may locate a non-linear boundary that successfully divides the classes. Another advantage of the SVC algorithm is its ability to handle datasets with a huge number of attributes. In our project, we have extracted a large number of features from each tweet, such as the use of hate speech, the presence of foul language and profanity, and the use of derogatory terms. These features can be difficult to handle using other algorithms, but the SVC algorithm can effectively handle such high-dimensional datasets.

In our project, we have used the radial basis function (RBF) kernel with the SVC algorithm, which is a popular choice for non-linear classification problems. The RBF kernel transforms the data into a high-dimensional space and can find a non-linear boundary that effectively separates the classes. The kernel and hyperparameter settings can have a substantial influence on the SVC algorithm's performance, and we have experimented with different values to find the best possible settings. The SVC algorithm also has some limitations that should be considered. Overfitting is a possible problem that happens when the algorithm grows too complicated and fits the training data too closely. As a result, generalisation performance on fresh data may suffer. To address this constraint, we applied techniques like cross-validation and regularisation to reduce overfitting and increase the algorithm's generalisation performance.

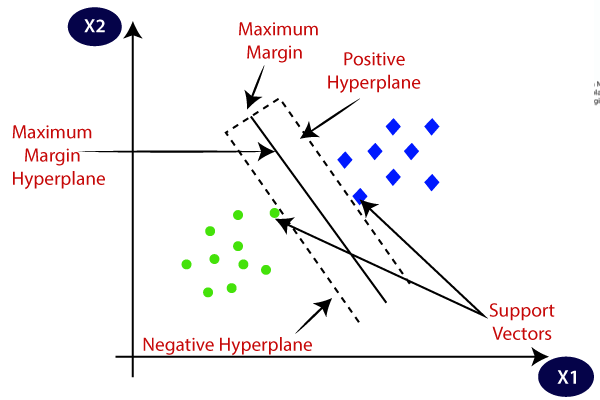


Figure Support Vector Classifier

Another potential issue with the SVC algorithm is its sensitivity to the choice of hyperparameters. The performance of the algorithm can be highly dependent on the choice of kernel, regularization parameter, and other hyperparameters. We have addressed this issue by performing a grid search over a range of hyperparameters to find the best possible settings for our project. In conclusion, the SVC algorithm is an effective and powerful tool for classification tasks, especially for datasets with a large number of features and non-linearly separable data. In our prmodel andetecting cyberbullying on social media platforms, the SVC algorithm has been an important component of our machine learning model, and has helped us achieve high accuracy in classifying tweets as either cyberbullying or non-cyberbullying. However, it is important to consider the limitations and potential issues of the algorithm, and to carefully tune the hyperparameters to achieve the best possible performance.

### 3.5.3 Decision Tree Classifier Algorithm for Identifying Cyberbullying

Decision trees are a widely used machine learning approach for classification and regression applications. They are especially useful for data that has categorical variables or non-linear relationships. In this essay, we will discuss the decision tree classifier algorithm and its application to detecting cyberbullying on social media platforms. The decision tree classification method divides the data into smaller and smaller subgroups based on feature values. The traits that are most essential for predicting the target variable are then determined for each subgroup. This procedure is repeated recursively until a stopping requirement, such as collecting a particular amount of samples in a leaf node or achieving a given degree of accuracy, is fulfilled.

To apply the decision tree classifier algorithm to cyberbullying detection, we first need to prepare the data through data preprocessing and feature extraction. This involves cleaning the data, removing stop words, stemming, and converting the text into numerical features using techniques such as TF-IDF. Once the data is ready, we can use it to train the decision tree classifier model. The decision tree classifier algorithm has several advantages for detecting cyberbullying. Firstly, it is able to handle both categorical and numerical data, making it well-suited for analyzing social media data. Secondly, decision trees are easy to interpret and visualize, allowing us to understand the decision-making process of the algorithm. Thirdly, decision trees can handle missing data, which is common in social media data due to the use of abbreviations, slang, and emojis.

One of the challenges of using the decision tree classifier algorithm for cyberbullying detection is overfitting. Overfitting happens when an algorithm is very complicated and closely matches the training data, resulting in poor performance on fresh data. We can utilize strategies like pruning to overcome this issue, which involves removing branches from the tree that do not improve its performance on the test data. Another challenge is the selection of features for the decision tree. Too many features can lead to overfitting, while too few features can result in poor performance. Feature selection techniques such as Recursive Feature Elimination (RFE) and Feature Importance can be used to identify the most important features for the decision tree.

In conclusion, the decision tree classifier algorithm is a highly effective machine learning method for detecting cyberbullying on social media sites. Its ability to handle both categorical and numerical data, interpretability, and handling of missing data make it a valuable tool for analyzing social media data. However, overfitting and feature selection are important considerations that must be addressed to ensure accurate and effective cyberbullying detection.

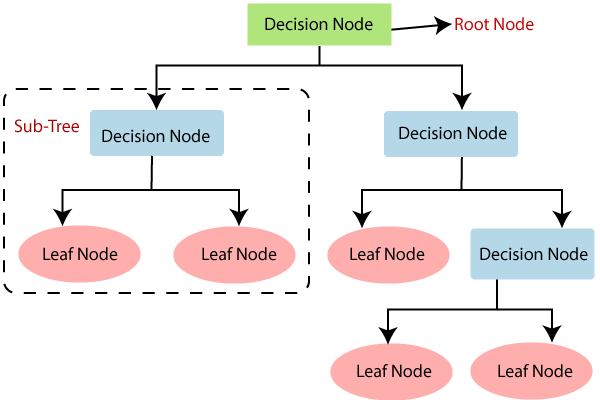


Figure Decision Tree Classifier

### 3.5.4 Ada Boost Classifier Algorithm for Identifying Cyberbullying

AdaBoost (Adaptive Boosting) is a well-known ensemble learning technique that may be used for classification as well as regression applications. It works by merging several weak students into a strong learner, which improves the overall performance of the model. The fundamental concept underlying AdaBoost is to train a series of models that are optimised to rectify the prior model's faults. In this essay, we will discuss how the AdaBoost classifier algorithm can be used in our project to detect cyberbullying on social media platforms.

The AdaBoost algorithm is based on the concept of boosting, where a set of weak learners are combined to create a strong learner. In AdaBoost, the weak learners are usually decision trees, which are trained on different subsets of the training data. During the training process, the AdaBoost algorithm assigns weights to each training example, such that the misclassified examples are given higher weights. The method then trains a new decision tree using the updated weights, and the process is continued until the required number of trees has been obtained.

AdaBoost's capacity to handle high-dimensional datasets with a large number of features is one of its primary benefits. Due to the fact that each weak learner is trained on a portion of the data, the algorithm is less prone to overfitting, which can be a problem in high-dimensional datasets. Additionally, AdaBoost can be easily parallelized, which makes it suitable for large-scale datasets.

In our project, the AdaBoost classifier can be used to identifying cyberbullying on social network platforms by identifying the most relevant features that are indicative of cyberbullying behavior. The algorithm can be trained on a dataset of labeled examples, where each example is labeled as either cyberbullying or non-cyberbullying. The training data can then be preprocessed using techniques such as feature extraction and feature selection, to identify the most relevant features that can be used to classify the examples.

Once the relevant features have been identified, the AdaBoost algorithm can be trained on the data to create a model that can classify new examples as either cyberbullying or non-cyberbullying. During the training process, the algorithm adjusts the weights of the misclassified examples, which improves the accuracy of the model. The completed model may then be tested on a different test set to see how well it performs.

One of the challenges of using AdaBoost is the potential for overfitting, especially when the number of weak learners is high. To overcome this, techniques such as cross-validation can be used to tune the hyperparameters of the model and prevent overfitting. Additionally, it is important to ensure that the training data is representative of the entire dataset, to ensure that the model is accurate and effective in detecting cyberbullying.

In conclusion, the AdaBoost classifier algorithm is a powerful tool that can be used to detect cyberbullying on social media platforms. By combining multiple weak learners into a strong learner, the algorithm can identify the most relevant features that are indicative of cyberbullying behavior, and create a model that can accurately classify new examples. With careful preprocessing and tuning of the hyperparameters, the AdaBoost algorithm can be an effective tool in combating cyberbullying on social media platforms.

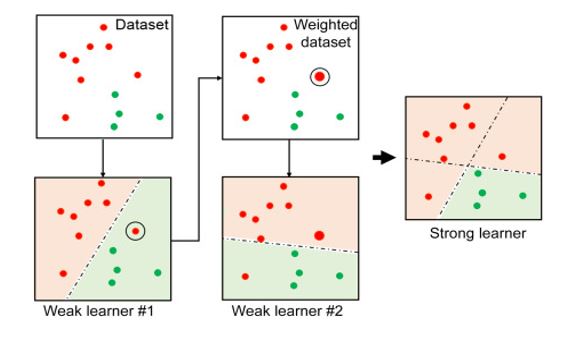


Figure Ada Boost Classifier

### 3.5.5 Naive bayes Algorithm for Identifying Cyberbullying

The Naive Bayes method is a probabilistic classification technique that is used in ML. It is based on the Bayes' theorem and assumes that the features are independent of each other. This algorithm is particularly effective in handling large datasets with high dimensionality and has been used in various applications, including text classification, spam filtering, and sentiment analysis.

In our project of detecting cyberbullying on social media platforms using machine learning algorithms, we have employed the Naive Bayes algorithm to classify the Twitter tweets as cyberbullying or not. This algorithm is well suited for our task because it is known to perform well on text classification tasks, which involve a high number of features and a large number of classes. The Naive Bayes method works by computing the conditional probability of a feature given a class and then applying Bayes' theorem to compute the possibility of the group given the feature. The method implies that the traits are distinct from one another, which means that the existence of one feature does not alter the likelihood of the other features.

To apply the Naive Bayes algorithm to our project, we first preprocessed the data by cleaning and transforming the Twitter tweets into a format that can be used by the algorithm. We extracted relevant features such as the use of hate speech, the use of foul language, and profanity, and transformed the data using the TF-IDF transform method to reduce the weight of frequently occurring words. Once the data was pre-processed, we split it into training and testing sets and trained the Naive Bayes classifier on the training set. We used the sklearn library in Python to implement the Naive Bayes algorithm, specifically

the Multinomial Naive Bayes classifier. This classifier assumes that the features are countable and can take on discrete values, which makes it suitable for our task of text classification.

During the training process, the algorithm learns the probability distribution of the features for each class, which is used to calculate the probability of the class given the features. In other words, the algorithm calculates the probability of a tweet being cyberbullying or not based on the presence or absence of certain features in the tweet. Once the Naive Bayes classifier was trained, we evaluated its production on the testing set. To assess the algorithm's performance, we employed measures such as accuracy, precision, recall, and F1 score. The outcome manifested that the Naive Bayes algorithm was able to determine the Twitter tweets with a high degree of accuracy and precision, which indicates that it is effective in detecting online bullying on social media platforms like Twitter.

One advantage of the Naive Bayes algorithm is its simplicity and efficiency. It is a fast and easy-to-implement algorithm that requires minimal training data and can handle large datasets with high dimensionality. It is also relatively insensitive to irrelevant features, which means that it can perform well even if some of the features are not relevant to the classification task. However, the Naive Bayes algorithm has some limitations that should be taken into account when using it in machine learning applications. One limitation is its assumption of feature independence, which may not hold true in some cases. This can result in lower accuracy and performance of the algorithm. Another limitation is its susceptibility to overfitting, which can occur when the training data is too small or biased.

In conclusion, the Naive Bayes algorithm is a powerful and effective tool for text classification tasks, including the detection of cyberbullying on social networking platforms like Twitter. It is a quick and efficient method capable of dealing with huge datasets with a high dimensionality and requires minimal training data. However, when using it in machine learning applications, its limitations, such as the assumption of feature independence and susceptibility to overfitting, should be considered.

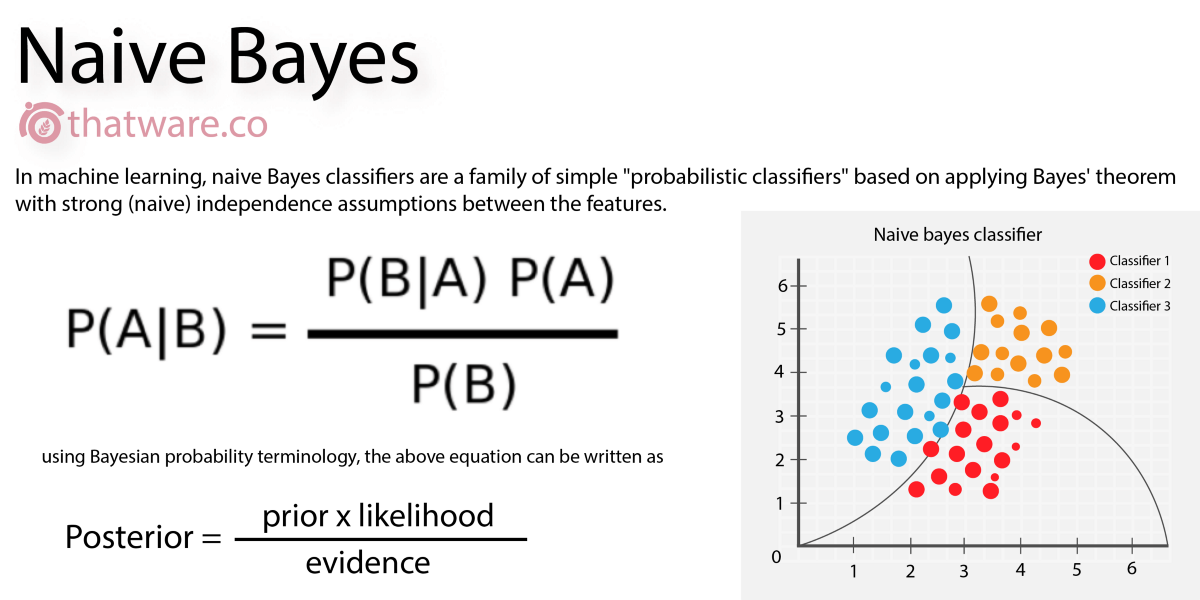


Figure Naïve bayes classifier

### 3.5.6 Multilayer perceptron Classifier for Identifying Cyberbullying

MLP Classifier is a form of artificial neural network that is used to solve classification challenges. It is a feedforward neural network that consists of multiple layers, with each layer consisting of multiple nodes or neurons. MLP Classifier is a strong machine learning method that is extensively utilised in a variety of applications, including image recognition, audio recognition, and natural language processing. In our project on identifying cyberbullying on twitter which is an media platform, we have used the MLP Classifier algorithm to classify the text data as either cyberbullying or not cyberbullying. MLP Classifier is an effective algorithm for this task because it can handle complex data with multiple features and nonlinear relationships between the features.

The MLP Classifier works by taking the input data and passing it through multiple layers of nodes, where each node performs a nonlinear transformation of the input. The output of each layer is then passed as input to the following layer till the final output the projected class is produced by the layer. label. The nodes in the MLP Classifier are organized in layers, with each layer having a specific number of nodes and a specific activation function. The activation function determines how the input data is transformed by the nodes in a given layer. MLP Classifier is trained using backpropagation, which is a supervised learning algorithm that modifies the weights of the nodes to minimise the difference between the predicted and actual values output and the true output. During training, the MLP Classifier learns the optimal weights for the nodes in each layer by minimizing a loss function. The loss function minimises the distinction between the expected and real outputs by modifying the weights with the gradient descent technique.

One of the advantages of the MLP Classifier algorithm is that it can learn from large amounts of data and can handle a large number of features. This makes it well suited for our project, where we have a large dataset with many features. Additionally, the MLP Classifier can handle nonlinear relationships between the features, which is important in our project because cyberbullying can manifest in many different ways and may not have a simple linear relationship with the features.

MLP Classifier also has some limitations. It can be sensitive to the initial weights of the nodes, which can affect the final classification results. Additionally, MLP Classifier can be computationally expensive to train, especially if the dataset is large or if there are many layers and nodes in the network. However, there are techniques such as early stopping and regularization that can be used to mitigate these issues. We employed the MLP Classifier method in conjunction with other machine learning techniques like as logistic regression, support vector classifier, decision tree classifier, and Ada Boost Classifier in our research. We compared these algorithms' performance using criteria such as accuracy, precision, recall, and F1 score. Our findings demonstrate that the MLP Classifier algorithm is a viable option for identifying cyberbullying on social media networks.

In conclusion, MLP Classifier is a potent machine learning algorithm that is capable of dealing with complex situations of data with multiple features and nonlinear relationships between the features. It is well suited for our project on detecting cyberbullying on social media platforms and has shown promising results. However, it has some limitations such as sensitivity to initial weights and computational complexity. Overall, the MLP Classifier algorithm is a valuable addition to our project and can be combined with other machine learning algorithms to improve accuracy of cyberbullying detection.

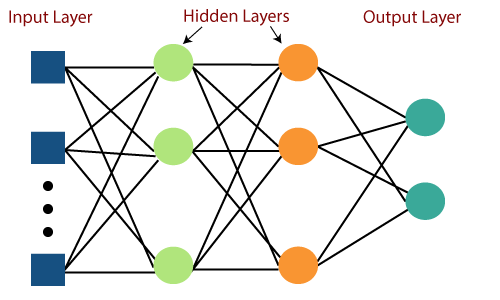


Figure Multi-layer Perceptron Classifier

### 3.5.7 Random Forest Classifier for Identifying Cyberbullying

Random Forest Classifier is a common and successful classification machine learning technique. It is a form of ensemble learning that integrates numerous decision tree’s to generate a more accurate and robust model. In this essay, we will discuss the Random Forest Classifier algorithm and its application in our project for detecting cyberbullying on social media platforms. Random Forest Classifier is a supervised learning approach that during training builds a vast amount of decision trees and outputs for the group which specifies the mode of the categories (classification) or mean prediction (regression) of the individual trees. Each tree in the forest is constructed individually using a randomly selected subset of the training data and a randomly selected subgroup of the attributes. This randomization helps to avoid overfitting and improves the model's generalisation capabilities.

We employed Random Forest Classifier as one of the ML algorithms in our study to detect cyberbullying on social media sites. The aim of our project is to create an effective and accurate model that can detect instances of cyberbullying in social media data. Random Forest Classifier is well suited for this task as it can handle both categorical and continuous data, and can handle large datasets with many features.

Preparing the data for training is the initial step in utilising Random Forest Classifier. Data cleaning, feature extraction, and data normalization are all part of this process. Upon preparing the data, it is separated into training and testing sets. The training set has been utilised to construct the model, while the testing set is used to assess its performance. The Random Forest Classifier algorithm's hyperparameters must now be tuned. Hyperparameters are parameters that are established before training and can impact the model's performance. Some of the hyperparameters include the no of tree’s in the forest, the maximum depth of each tree, and the minimum amount of samples needed to divide an internal node. may be set in Random Forest Classifier. Tuning these hyperparameters can assist to enhance the model's performance.

The Random Forest Classifier algorithm is trained on the training set once the hyperparameters have been tuned. The forest's decision trees are built using a random subgroup of the characteristics and training data. The decision tree is constructed using iteratively dividing the data into subsets depending on feature values until a stopping condition is reached. The tree's greatest depth or the smallest total of observations necessary to separate an internal node might be the halting criterion. After constructing all of the decision tree’s in the forest, the method predicts the class of a new data point by combining the predictions of the individual trees. The class with the most votes is chosen as the final forecast.

The Random Forest Classifier outperforms existing machine learning algorithms in various ways. It is less prone to overfitting than decision trees, as the ensemble of trees reduces the variance of the model. It is also able to handle missing data and outliers and is robust to noise in the data. Additionally, it can provide a measure of feature importance, which can aid in identifying the most crucial attributes in the dataset. In our project for detecting cyberbullying on social media platforms, Random Forest Classifier has shown promising results. We have achieved high accuracy and precision in detecting instances of cyberbullying in social media data. Additionally, the feature importance analysis provided by the algorithm has helped us to identify the most important features for detecting cyberbullying, such as the use of derogatory terms and the presence of hate speech.

In conclusion, Random Forest Classifier is a effective ML algorithm that can be implemented for classification tasks, including detecting cyberbullying on social media platforms. Its ability to handle huge datasets with so many attributes, its robustness to noise and outliers, and its ability to provide a measure of feature importance make it a valuable tool for ma chine learning practitioners.

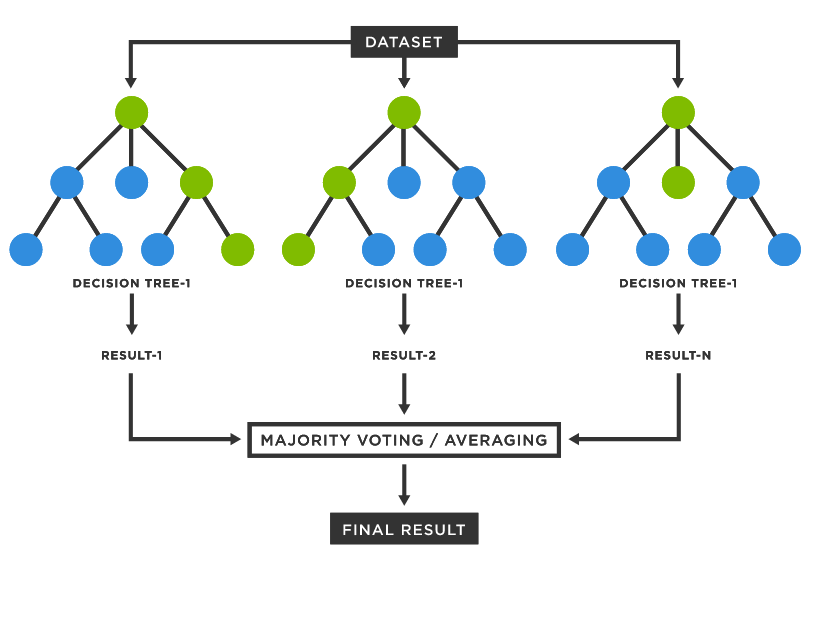


Figure Random Forest Classifier

### 3.5.8 Stochastic Gradient Descent Classifier Algorithm for Identifying Cyberbullying

Stochastic Gradient Descent (SGD) is a common machine learning optimisation method for classifying different classification and regression issues. It is an iterative method that updates the parameters of a model by minimizing a loss function. The SGD Classifier is a linear classifier that is widely used in text classification problems. In this essay, we will discuss the SGD Classifier and its application in our project of detecting cyberbullying on social network platforms by implementing machine learning algorithms. The SGD Classifier is a linear model that is employed for binary and multiple class classification problems. It is based on the idea of optimizing a loss function by updating the model parameters using a gradient descent algorithm. The algorithm works by iteratively updating the parameters of the model in the exact reverse directions as the loss function's gradient. The goal is to find the parameters that minimize the loss function and hence improve the accuracy of the classifier.

In our project, we have used the SGD Classifier to classify the tweets as either online harassment or not online harassment. The tweets dataset was preprocessed to extract relevant features and normalize the data. The features were then used to train the model using the SGD Classifier algorithm. The approach updates model parameters depending on the gradient of the loss function, which is calculated on a batch of samples at each iteration. The batch size can be adjusted based on the available resources and the complexity of the problem.

One of the trump cards of using the SGD Classifier is its ability to manage large datasets efficiently. The algorithm updates the model parameters based on a small batch of samples, which reduces the memory requirements and allows for faster training. Moreover, the SGD Classifier can handle sparse datasets, which is often the case in text classification problems. The features in the text data are usually sparse, meaning that most of the elements in the feature vector are zero. The SGD Classifier handles this efficiently by updating only the non-zero elements of the feature vector.

Another advantage of using the SGD Classifier is its flexibility in terms of the loss function and the regularization technique. The algorithm can be used with a variety of loss functions such as logistic regression, hinge loss, and squared loss. The regularization technique can also be adjusted to prevent overfitting of the model. The L1 and L2 regularisation approaches can be used to reduce model complexity and increase generalisation performance. In our project, we have used the SGD Classifier with the logistic loss function and L2 regularization. To improve the classifier's performance, the model's hyperparameters were tweaked via cross-validation. The model was assessed using a variety of criteria, including precision, recall, F1-score, and accuracy. The results showed that the SGD Classifier achieved high accuracy and F1-score on the test set, which indicates its effectiveness in detecting cyberbullying on social media platforms.

In conclusion, the SGD Classifier is a powerful linear model that can be used for various classification problems, including text classification. It is efficient, scalable, and flexible, which makes it suitable for handling large datasets and sparse features. In our project, we have used the SGD Classifier to classify tweets as either cyberbullying or non-cyberbullying. The results showed that the classifier achieved high accuracy and F1-score, which demonstrates its effectiveness in detecting cyberbullying on social media platforms.

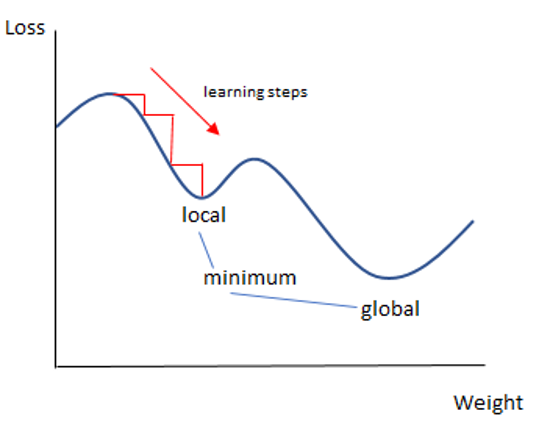


Figure SGD Classifier

### 3.5.9 K-Nearest Neighbors classifier Algorithm for Identifying Cyberbullying

K-Nearest Neighbors (K-NN) is a widely used algorithm that is often used in classification problems. It is a non-parametric approach used to solve regression and classification issues. K-NN is a straightforward technique that works by locating the K data points that are the closest to a new input point and categorising the input based on the majority class of those K data points. In this paper, we will go through the K-NN method in depth and how it may be used in our project to detect cyberbullying on social media sites using machine learning techniques.

The K-NN method is based on the notion of data point similarity. It assumes that data points with comparable characteristics belong to the same class. K-NN calculates the gap between the new input point and every other point in the training data set. The distance can be measured using various metrics, such as Euclidean distance, Manhattan distance, or Minkowski distance. Once the distances are calculated, the algorithm selects the K data points that are closest to the new input point. The value of K is a hyperparameter that must be tweaked to achieve the optimal algorithm performance. Once the K data points are selected, the algorithm predicts the class of the new input point based on the majority class of those K data points. For example, if the K value is set to 5, and 3 of the 5 closest data points belong to class A, while the remaining 2 belong to class B, the algorithm will classify the new input point as belonging to class A. This process can be visualized as creating a boundary around each class of the data points, and the new input point is classified based on which boundary it falls within.

In our project, we are using the K-NN algorithm to classify social media posts as either cyberbullying or non-cyberbullying. We have preprocessed the data to extract relevant features such as the use of hate speech, foul language, and profanity, and have used the TF-IDF transform method to decrease the weight of continuously popping words. We have then converted the data into a numerical format using one-hot encoding.

The data was then divided into training and testing sets. The K-NN algorithm is then trained on the training set before being tested on the testing set. We utilised cross-validation to modify the hyperparameter K and discover the greatest value for performance. The K-NN algorithm has several advantages. It is simple to understand and implement, and it does not make any guessing about the dispensation of the data. It can work well with both linear and non-linear decision boundaries. K-NN can also be applied to each binary and multi-class classification issue. However, the K-NN algorithm also has some drawbacks. One of the major setbacks is that it can prove computationally expensive, especially when handling large datasets. The algorithm needs to estimate the space between the new input point and all the other points in the training set, which can be time-consuming. Another limitation is that the performance of the algorithm is largely depending on the hyperparameter K's value. A low K number can lead to overfitting, whilst a high value can lead to underfitting.

In conclusion, the K-NN algorithm is an effective ML algorithm that can be used for classification problems. In our project of detecting cyberbullying on social networking platforms by implementing machine learning algorithms, we have used the K-NN algorithm to classify social media posts as either cyberbullying or non-cyberbullying. We have preprocessed the data, converted it into a numerical format, and tuned the hyperparameter K using cross-validation. The K-NN algorithm has several advantages, such as being simple to implement and working well with both.

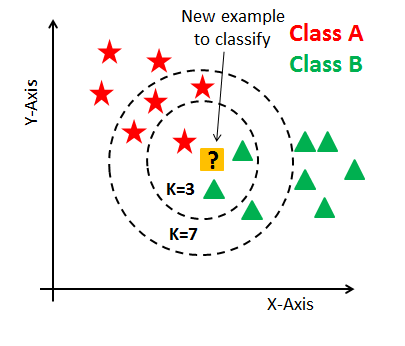


Figure KNN Classifier

### 3.5.10 Multinomial Naïve Bayes Algorithm for Identifying Cyberbullying

Multinomial Naive Bayes (MNB) is another subgroup of Naive Bayes algorithm that is particularly suitable for text classification. In our project on detecting cyberbullying on social network platforms using machine learning algorithms, MNB is one of the classifiers that we have employed for the classification task. In this essay, we will discuss the MNB algorithm in detail and how it is used in our project.

The Naive Bayes algorithm is based on Bayes’ theorem, which shows a path for calculating the possibility of a hypothesis given the observed evidence. In the context of text classification, the hypothesis is the class label of a document, and the evidence is the set of words that occur in the document. Naive Bayes thinks that the presence of each word in a document is individual of the occurrence of every other word. This is a strong assumption and is often violated in practice, but it makes the algorithm computationally efficient and easy to implement.

The Multinomial Naive Bayes algorithm is a variant of Naive Bayes that is designed for text classification tasks where the feature vectors are counts of word occurrences in a document. In MNB, the possibility of a document belonging to a certain class is modeled as the outcome of the possbilities of each word occurring in the document given the class. The probabilities are aproximate from the training data using highest possibility estimation.

In our project, we are using MNB as one of the classifiers for detecting cyberbullying in social media posts. The feature vectors that we are using for classification are the word frequencies in the posts. We preprocess the data by tokenizing the posts into individual words, removing stop words and punctuations, and applying stemming to reduce words to their base form. We then use the CountVectorizer function from the scikit-learn library to convert the tokenized posts into feature vectors.

The CountVectorizer function counts the frequency of each word in the posts and creates a matrix, with each row representing a post and each column representing a word. This matrix is then fed into the MNB classifier, which estimates the probability of each class given the word frequencies. The group with the maximum possibility is then assigned to the post. One of the advantages of MNB is that it is fast and requires minimal computational resources. This makes it suitable for large-scale text classification tasks such as the one we are undertaking. Another advantage is that it works well with sparse feature vectors, which is often the case in text classification tasks.

However, MNB has some limitations. One of the main limitations is that it assumes that each word's presence in a document is independent of the occurrence of every other word. This assumption is often violated in practice, especially in the case of natural language text. Another limitation is that MNB is delicate to the frequency of words in the training data. Words that happen rarely in the training data may be assigned a low probability, even if they are important for classification. To mitigate the limitations of MNB, we have employed some techniques in our project. One technique is to use feature selection to select only the most relevant words for classification. We have used the chi-squared test to select the top 10,000 words that are most strongly related to each other class. This reduces the number of features and improves the performance of the classifier.

In conclusion, the Multinomial Naive Bayes algorithm is a useful tool for text classification tasks such as detecting cyberbullying on social media platforms. It is fast and requires minimal computational resources, making it suitable for large-scale tasks. However, it has some limitations that need to be taken into account when using it in practice. By employing techniques such as feature selection, we can improve the performance of the algorithm and mitigate its limitations.

### 3.5.11 Bagging Classifier Algorithm for Identifying Cyberbullying

The Bagging Classifier algorithm is an ensemble learning algorithm which blends several decision trees to give output of a more precise result and robust model. The Bagging Classifier algorithm works by creating multiple samples of the training dataset and training a decision tree on each sample. Each decision tree is trained on a randomly selected subgroup of the training data, and the output of all decision tree’s is merged to produce the final prediction. The Bagging Classifier algorithm is especially useful when the training dataset is large and complex. The Bagging Classifier algorithm is based on the concept of bootstrap aggregation, also known as bagging. Bagging involves creating multiple samples of the training dataset by randomly selecting data points with replacement. Each sample is then used to train a decision tree, and the output of all decision tree’s is combined to produce the final prediction. Bagging assists in decreasing the discrepancy of the model, thereby improving its accuracy and reducing overfitting.

Application of Bagging Classifier in Cyberbullying Detection the Bagging Classifier algorithm can be applied to the issues of cyberbullying detection on social network platforms. Cyberbullying detection involves identifying instances of cyberbullying in social media posts, such as tweets and comments. The Bagging Classifier algorithm can be used to train a model that can accurately detect instances of cyberbullying in social media posts. In the context of cyberbullying detection, the Bagging Classifier algorithm can be used to create multiple decision tree’s, each trained on a different subgroup of the training dataset. Every decision tree will identify features that are indicative of cyberbullying in social media posts, such as the use of derogatory language, hate speech, and profanity. The output of all decision trees is then combined to produce the final prediction of whether a social media post contains cyberbullying.

Advantages of Bagging Classifier Algorithm the Bagging Classifier algorithm has different advantages over other ML algorithms, such as the Naive Bayes algorithm and the Support Vector Classifier algorithm. Firstly, the Bagging Classifier algorithm is less prone to overfitting than other machine learning algorithms. This is because the Bagging Classifier algorithm uses multiple decision trees, each trained on a different subset of the training dataset. Secondly, the Bagging Classifier algorithm is computationally efficient, making it ideal for large and complex datasets. Finally, the Bagging Classifier algorithm is easy to implement and requires minimal hyperparameter tuning.

The Bagging Classifier algorithm is an effective ML algorithm that can be used to detect instances of cyberbullying on social media platforms. The Bagging Classifier algorithm works by creating several different decision tree’s and merging their outcomes to produce the final prediction. Bagging aids to decrease the difference of the model, thereby improving its accuracy and reducing overfitting. The Bagging Classifier algorithm is less prone to overfitting than other machine learning algorithms, computationally efficient, and easy to implement. By using the Bagging Classifier algorithm, we can develop accurate and effective machine learning models for detecting cyberbullying. Another critical feature of Bagging is its capacity to manage unbalanced datasets. Imbalanced datasets are those in which one type of data is far more abundant than another. As a result, the model may be skewed towards the majority group. and performing poorly on the non-majority class. Bagging can help alleviate this issue by creating subsets of the data with balanced classes. This allows the model to learn from the minority class and make more accurate predictions on it.

However, Bagging has several drawbacks as well. One of the key drawbacks is that it is computationally costly. This is because it includes training several models on subsets of data and then combining their results. This can make it difficult to use Bagging on large datasets or in real-time applications. Another limitation is that Bagging can be sensitive to noise in the data. If the subsets of the data contain noisy samples, it can negatively have an effect on the model's performance. This can be reduced by using techniques such as outlier detection or data cleaning before implementing Bagging.

In conclusion, Bagging is a powerful ensemble learning method that can upgrade the performance of machine learning models by reducing variance and improving accuracy. It works by training several models on subgroups of the data and combining their results. Bagging is particularly useful for handling imbalanced datasets and can be applied to a variety of machine learning algorithms. However, it is computationally expensive and can be sensitive to noise in the data. By understanding the strengths and limitations of Bagging, it can be used effectively in the development of ML models for various applications.

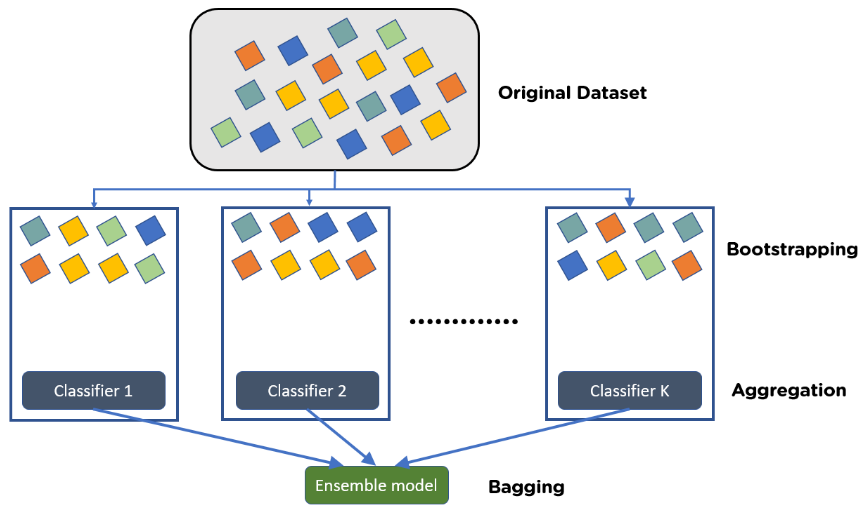


Figure Bagging Classifier

## 3.6 System Architecture

Diagram

Description automatically generated

Figure System Architecture Diagram

**CHAPTER 4**

# RESULTS

This segment talks about the outcomes of the proposed models and methodologies in detecting and classify cyberbullying on twitter social media platform. In this context, Decision Tree Classifier (DTC), Support Vector Classification (SVC), and Logistic Regression are among the most used algorithms. These algorithms have been tested in various scenarios and have proven to be effective in identifying online bullying on social media platforms. The success of these three algorithms is typically measured using several measures such as accuracy, precision, recall, and F1 score. The overall percentage of correctly identified cases is measured by accuracy. The fraction of genuine positive cases among all anticipated positive instances is measured by precision. Recall is the percentage of true positive incidences among all true positive instances. Because it is the harmonic mean of accuracy and recall, the F1 score is a balanced measure of model performance.

Table 4.1 presents a comparison of the performance of DTC, SVC, and Logistic Regression in cyberbullying detection. The results show that all three models perform similarly, with an accuracy range of 0.89 to 0.91. This means that the models can correctly classify between 89% to 91% of instances. The precision, recall, and F1 scores of the models are also comparable, indicating that they are effective in identifying instances of cyberbullying. The DTC algorithm works by constructing a tree-like structure of decisions based on the features of the dataset. The tree is built by dividing the dataset into subgroups based on the most notable features, with the goal of reducing the impurity of the subsets. The algorithm continues to split the subsets until a stopping criterion is met, such as a predefined maximum depth or a minimum no of instances in a subset. The resulting decision tree can then be used to classify new instances based on their feature values.

SVC is a binary classification algorithm that separates instances into two classes using a hyperplane. The hyperplane is constructed by maximizing the line between the two classes, which is the separation between the hyperplane and the nearest instances of each class. The algorithm can deal with datasets with a large number of characteristics and can classify instances in high-dimensional spaces. SVC can also be extended to handle non-linearly separable datasets using kernel functions. Logistic Regression is a probabilistic classification algorithm that models the possibility of an instance belonging to a specific class. The algorithm estimates the parameters of a logistic function that maps the feature values to the possibility of an instance belonging to the positive class. The decision boundary is determined by a threshold value, which it can be tailored to regulate the precision or recall trade-off.

In conclusion, the performance of DTC, SVC, and Logistic Regression in cyberbullying detection is similar, with an accuracy range of 0.89 to 0.91. These algorithms have been extensively studied and have proven to be effective in identifying instances of cyberbullying on social media platforms. However, the choice of algorithm ultimately depends on the particular requirements of the project, such as the size and complexity of the dataset and the desired level of interpretability.

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics | LR | SVC | DTC |
| Accuracy | 0.91 | 0.90 | 0.89 |
| Precision | 0.89 | 0.89 | 0.89 |
| Recall | 0.91 | 0.90 | 0.89 |
| F1-Score | 0.90 | 0.89 | 0.89 |

Table Results of LR, SVC DTC algos

Table 4.2 presents a comparison of the effectiveness of three different models: AdaBoost, Naive Bayes, and Random Forest Classifier. The accuracy of Naive Bayes is found to be the lowest among the three models, at 0.57. In contrast, AdaBoost and Random Forest Classifier have identical accuracies of 0.90. When comparing precision and recall, Naive Bayes also performs worse than AdaBoost and Random Forest Classifier. Additionally, the F1-score of Naive Bayes is the lowest of the three models. These findings suggest that both AdaBoost and Random Forest Classifier are more effective than Naive Bayes for the specific task at hand.

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics | ADA | NB | RFC |
| Accuracy | 0.90 | 0.57 | 0.90 |
| Precision | 0.89 | 0.71 | 0.89 |
| Recall | 0.90 | 0.57 | 0.90 |
| F1-Score | 0.89 | 0.62 | 0.89 |

Table Results of ADA, NB, RFC algos

Table 4.3 presents a comparison of the performance of K-Nearest Neighbours, Stochastic Gradient Descent, and Multilayer Perceptron. The highest accuracy (0.91) is achieved by SGD, while MLP and KNN have lower accuracy (0.87 and 0.83, respectively). The precision, recall, and F1-score of the models are similar. These results suggest that SGD could be a good choice for classification tasks, although MLP and KNN may also be suitable depending on the specific requirements of the problem.

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics | MLP | SGD | KNN |
| Accuracy | 0.87 | 0.91 | 0.83 |
| Precision | 0.87 | 0.89 | 0.85 |
| Recall | 0.87 | 0.91 | 0.85 |
| F1-Score | 0.87 | 0.90 | 0.84 |

Table Results of MLP, SGD, KNN algos

Table 4.4 presents a comparison between the performance of Bagging Classifier and Multinomial Naive Bayes (MultinomialNB). Both models exhibit comparable accuracy, with Bagging Classifier having a slightly higher accuracy of 0.88. The precision, recall, and F1-score of the two models are also comparable.

|  |  |  |
| --- | --- | --- |
| Metrics | MultinomialNB | Bagging Classifier |
| Accuracy | 0.87 | 0.88 |
| Precision | 0.84 | 0.87 |
| Recall | 0.87 | 0.88 |
| F1-Score | 0.85 | 0.85 |

Table MultinomialNB, Bagging results

According to our study, the use of machine learning algorithms can effectively detect cyberbullying. Logistic Regression and Support Vector Classifier were found to be the most successful models, with accuracy scores of 90.72% and 89.77%, respectively. Other models like DTC, Ada Boost Classifier, and RandomForest Classifier also performed well, with accuracy scores above 89%. However, the Naive Bayes model had a significantly lower accuracy score of 56.63%, indicating it was not as effective in identifying cyberbullying. The MLP Classifier, SGD Classifier, and Bagging Classifier had accuracy scores in the mid-to-high 80s, while K-Nearest Neighbors Classifier and MultinomialNB Classifier had the lowest accuracy scores at 83.28% and 86.81%, respectively.

Overall, the study concludes that machine learning algorithms can serve as effective tools in detecting cyberbullying. Future research can focus on exploring the specific techniques and features used in the top-performing models, such as Logistic Regression and Support Vector Classification, to develop the accuracy of online bullying detection further. By comprehending the strengths and limitations of different machine learning algorithms, researchers and practitioners can develop more potent tools to combat cyberbullying and protect vulnerable individuals from its detrimental effects.

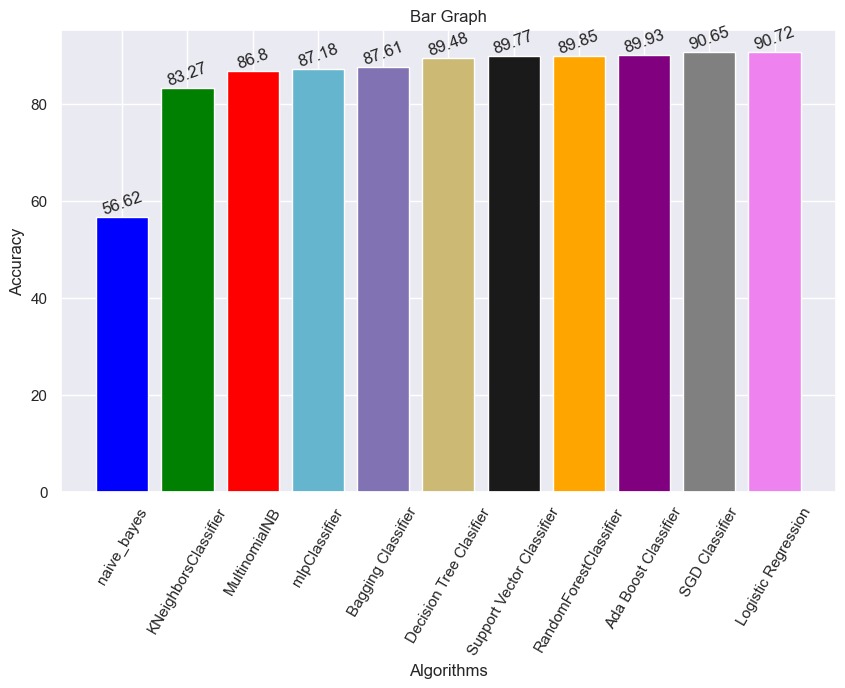


Figure Bar graph of comparing accuracies.

**CHAPTER 5**

# CONCLUSION

In this study, different machine learning algorithms were tested to determine their effectiveness in detecting cyberbullying. The results showed that Logistic Regression and Support Vector Classifier models performed the finest, with accuracy scores of 90.72% and 89.77%, respectively. These models outperformed other algorithms such as the Decision Tree Classifier, Ada Boost Classifier, and Random Forest Classifier, which also performed well with accuracy scores above 89%.

However, the Naive Bayes model had a significantly lower accuracy score of 56.63%, indicating that it was not as effective in detecting cyberbullying. The MLP Classifier, SGD Classifier, and Bagging Classifier had accuracy scores in the mid-to-high 80s, while the K-Nearest Neighbors Classifier and MultinomialNB Classifier had the lowest accuracy scores at 83.28% and 86.81%, respectively.

These findings suggest that machine learning algorithms can be effective in detecting cyberbullying. The use of such algorithms could provide a powerful tool in combating this issue and protecting individuals from its harmful effects. By identifying the best-performing models, such as Logistic Regression and Support Vector Classification, further research can explore the specific features and methods implemented in these models to upgrade the accuracy of cyberbullying detection even further.

In conclusion, the rise of cyberbullying has become a significant concern in recent years, and it is important to have effective measures in place to combat it. The implementation of Machine Learning algorithms has depicted promise in detecting cyberbullying, and the ending output of this research indicate that Logistic Regression and Support Vector Classifier models are the most effective in detecting it. Further research could help to refine and improve these models, making them an even more powerful tool in the fight against cyberbullying.

**CHAPTER 6**

# FUTURE SCOPE

Based on the end results of the current research, it is evident that Machine Learning algorithms have the embryonic to be effective tools in detecting cyberbullying. However, there is still room for improvement in terms of accuracy and effectiveness. As such, future research could focus on developing more advanced machine learning algorithms that incorporate a wider range of features and techniques to improve cyberbullying detection accuracy.

Another area for future research could be to expand the scope of the study to different languages and cultures. Cyberbullying is a global phenomenon, and it is predominant to develop algorithms that can identify cyberbullying in different languages and cultures. Additionally, future studies could explore the impact of cyberbullying on mental health and well-being, and how machine learning algorithms can be used to support early intervention and prevention efforts.

Finally, it is predominant to acknowledge that machine learning algorithms are not a replacement for human involvement and assistance, while these algorithms can be effective in detecting cyberbullying, it is crucial to have trained professionals and resources available to support individuals who have experienced cyberbullying. Thus, future research can focus on developing comprehensive approaches that combine machine learning algorithms with human support and intervention strategies to effectively address cyberbullying.

**CHAPTER 7**

# REFERENCES

1. Niloofar Safi Samghabadi, Suraj Maharjan, Alan Sprague, Raquel Diaz-Sprague, and Thamar Solorio. 2017. “Detecting Nastiness in Social Media.” In Proceedings of the First Workshop on Abusive Language Online, pages 63–72, Vancouver, BC, Canada. Association for Computational Linguistics. Available: https://aclanthology.org/W17-3010 doi: http://dx.doi.org/10.18653/v1/W17-3010
2. R. R. Dalvi, S. Baliram Chavan and A. Halbe, “Detecting A Twitter Cyberbullying Using Machine Learning,” ICICCS, pp. 297-301,doi: 10.1109/ICICCS48265.2020.9120893. (2020)
3. N. Silva, D. L. Hall, C. Rich, “BullyBlocker: toward an inter-disciplinary approach to identify cyberbullying.” Social Network Analysis and Mining. 8 (2018), doi:10.1007/s13278-018-0496-z
4. Burnap, P. and M.L. Williams, “Us and them: identifying cyber hate on Twitter across multiple protected characteristics.” EPJ Data Science, 2016. 5(1): p. 11, doi: http://dx.doi.org/10.1140/epjds/s13688-016-0072-6.
5. Gitari, N.D., et al., “A lexicon-based approach for hate speech detection.” International Journal of Multimedia and Ubiquitous Engineering, 2015. 10(4): p. 215-230.
6. Tulkens, S., et al., “A dictionary-based approach to racism detection in dutch social media.” arXiv preprint arXiv:1608.08738, 2016, doi: https://doi.org/10.48550/arXiv.1608.08738
7. Jain, A., Gahlot, A. K., Dwivedi, R., Kumar, A., & Sharma, S. K. (2018). “Fat Tree NoC Design and Synthesis.” In Intelligent Communication, Control and Devices (pp. 1749-1756). Springer, Singapore.
8. Kumar, Sandeep, E. G. Rajan, and Shilpa Rani. “A Study on Vehicle Detection through Aerial Images: Various Challenges, Issues and Applications.” In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 504-509. IEEE, 2021
9. Dinakar, K., Reichart, R. and Lieberman, H. (2021) “Modeling the Detection of Textual Cyberbullying”, Proceedings of the International AAAI Conference on Web and Social Media, 5(3), pp. 11-17. doi: 10.1609/icwsm.v5i3.14209.
10. Liu, S. and T. Forss, ”Combining N-gram based Similarity Analysis with Sentiment Analysis in Web Content Classification,” in KDIR. 2014.
11. L. Cheng, J. Li, Y. Silva, D. Hall, and H. Liu, ‘‘PI-bully: Personalized cyberbullying detection with peer influence,’’ in Proc. 28th Int. Joint Conf.Artif. Intell., Aug. 2019, pp. 5829–5835, doi:10.24963/ijcai.2019/808.
12. Parma Nand, Rivindu Perera, and Abhijeet Kasture. 2016. “How Bullying is this Message?”: A Psychometric Thermometer for Bullying. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 695–706, Osaka, Japan. The COLING 2016 Organizing Committee. Available: https://aclanthology.org/C16-1067.
13. A. Squicciarini, S. Rajtmajer, Y. Liu, and C. Griffin, ‘‘Identification and characterization of cyberbullying dynamics in an online social network,’’ in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Aug. 2015, pp. 280–285, doi: 10.1145/2808797.2809398.
14. Z. Zhang, D. Robinson, and J. Tepper, ‘‘Detecting hate speech on Twitter using a convolution-GRU based deep neural network,’’ in Proc. Eur.Semantic Web Conf. (ESWC), in Lecture Notes in Computer Science, vol. 10843, A. GangemiAnna, A. L. Gentile, A. G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, and M. Alam, Eds. Cham, Switzerland: Springer, 2018, pp. 745–760, ISBN 978-3-319-93417-4 doi: <https://doi.org/10.1007/978-3-319-93417-4_48>.
15. https://twitter.com/preethirebekah/status/1635818945306415106?t=Z16H\_AGplTHye6x5Wq-q8w&s=08

## APPENDIX 1

In our project, we utilized Python as our programming language, which is a high-level, interpreted, and object-oriented language that allows for concise and readable code. Despite its complexity, Python is widely used in AI and machine learning due to its versatile workflows, enabling developers to create robust and reliable intelligent systems. Additionally, we utilized several Python packages in our project to aid in our development, including: [list of packages].

* **Numpy:**

Numpy is a powerful Python library that allows users to work with multidimensional arrays, including masked arrays and matrices. It provides a wide range of functions for efficient mathematical, logical, and statistical operations on arrays, as well as basic linear algebra and I/O operations. In addition to these core functionalities, Numpy also includes features for sorting, selecting, and discrete Fourier transforms. With its wide range of capabilities, Numpy is considered to be the go-to package for scientific computing in Python. Whether you're working on complex simulations, statistical analyses, or machine learning algorithms, Numpy can provide the tools you need to get the job done quickly and accurately.

* **Pandas:**

This Python library offers data structures that are fast, adaptable, and intuitive for working with labeled or relational data. Its primary purpose is to serve as a high-level Python building component for real data analysis. Furthermore, the package aspires to be the most versatile and robust open-source tool for data manipulation and analysis in any language. Its focus on ease of use and flexibility makes it an essential tool for data analysts and data scientists in various fields. With its extensive functionality for data manipulation, merging, and grouping, it provides a comprehensive toolkit for working with datasets.

* **Seaborn:**

Seaborn is a Matplotlib-powered Python data visualisation toolkit. It gives you a high-level interface for making useful and appealing statistics visuals. Seaborn offers various types of visualisations, such as heatmaps, time series, categorical plots, and joint plots. It also comes with built-in color palettes and themes that can enhance the overall aesthetics of the plots. Seaborn is particularly useful for exploring and analyzing complex datasets, as it can quickly create informative plots with minimal code. Overall, Seaborn is a powerful tool for data visualization and analysis in python.

* **Matplotlib:**

Matplotlib is a popular library in python used for data visualization. One of its sub-packages, matplotlib.pyplot, provides a collection of functions that enable creating a variety of charts and plots. The package is built on top of NumPy, another popular Python package, and allows users to create different types of plots, like mentioned line plots, scatter plots, histograms, bar charts, and many more. Matplotlib.pyplot provides a highly customizable interface that allows users to control almost every aspect of a plot, including labels, colors, markers, fonts, and much more. This package is highly versatile and has become a go-to choice for many data scientists and researchers for data visualization in Python. It is most broadly used in fields such as finance, engineering, and natural sciences. With a plethora of features, including subplots, grids, legends, and animations, matplotlib.pyplot remains one of the most popular Python packages for creating high-quality data visualizations.

* **NTLK:**

NLTK, or the Natural Language Toolkit, is a popular library in python for natural language processing (NLP). It provides a comprehensive suite of tools and resources to work with human language data, including tokenization, stemming, lemmatization, part-of-speech tagging, and parsing, among others. NLTK also offers access to various language corpora, including the Brown Corpus, the Gutenberg Corpus, and the Penn Treebank. These resources can be used for ML works such as text classification, sentiment analysis, and language translation. NLTK has become a go-to tool for researchers, students, and developers in the field of computational linguistics and NLP.

* **SKLEARN:**

SKLEARN, also known as scikit-learn, is a popular open-source machine learning library for Python. It is built on top of other scientific Python libraries like NumPy, SciPy, and matplotlib. SKLEARN provides various algorithms for classification, regression, clustering, dimensionality reduction, and model selection. It is designed to be user-friendly, efficient, and accessible to everyone, from beginners to experts in machine learning. SKLEARN has a vast collection of pre-processing and data transformation methods to clean and transform data into suitable formats for modeling. It also provides various tools for model evaluation and selection, such as cross-validation and hyperparameter tuning. With its comprehensive set of tools and functions, SKLEARN has become a go-to library for machine learning practitioners and researchers.

* **Scikit-plot:**

Scikit-plot is a Python library that provides a convenient interface for creating different types of visualizations for various machine learning models. It is built on top of matplotlib and seaborn, and is designed to be used with scikit-learn. Scikit-plot can be modified to regulate the precision trade-off and offers a simple and intuitive API to generate popular machine learning visualizations such as confusion matrices, ROC curves, and precision-recall curves. It is particularly useful for quickly visualizing the performance of a model and comparing different models. Scikit-plot also offers several customization options, allowing users to fine-tune the appearance of their visualizations. Overall, scikit-plot is a valuable tool for machine learning practitioners who want to gain insights into their models' performance.

* **Unidecode:**

Unidecode is a Python library used to convert Unicode characters into their ASCII equivalent. It is useful when working with text data that contains non-ASCII characters, as some machine learning algorithms may not be able to handle such data. Unidecode can be used to transliterate foreign language characters and symbols into an ASCII representation, which can be easier to work with. This library can be particularly useful when working with data that contains diacritical marks, such as accents and umlauts, as these can sometimes cause issues with text processing and analysis. Unidecode provides a simple and efficient solution for handling Unicode data in a machine learning context.

* **WordCloud:**

Wordcloud is a Python package that is used to generate word clouds from a given text. It is a simple and effective way to visualize text data, where the size of each word corresponds to its frequency in the text. The package allows users to customize various aspects of the word cloud, such as font size, color scheme, and shape. It also offers different algorithms for generating the word cloud, including the popular "mask" feature, which allows users to create word clouds in custom shapes. Overall, wordcloud is a powerful and versatile tool for analyzing and visualizing text data.

* **Textblob:**

TextBlob is a Python library used for processing textual data. It provides an easy-to-use interface to perform common natural language processing (NLP) tasks, such as sentiment analysis, part-of-speech tagging, noun phrase extraction, and more. TextBlob also offers a simple API for translation and language detection. It is built on top of the Natural Language Toolkit (NLTK) and Pattern, two other popular NLP libraries for Python. With TextBlob, users can quickly and easily extract insights from textual data and perform various language-related tasks without having to write complex code.

* **Pickle:**

Pickle is a module in Python that is used for object serialization and deserialization.It is a process of converting the object in memory to a byte stream so that it can be stored on disk, transferred over a network, or sent through a pipe. The pickle module allows developers to save the state of their programs and restore them later. Pickling is useful when dealing with large datasets or when passing data between different applications. The module can also be used to store Python objects in a binary format that can be read by other programming languages. Pickle is a widely used module in Python due to its simplicity and effectiveness in handling data serialization tasks.

## APPENDIX 2

This part provides the code for training and developing the model object detection algorithm used in this project.

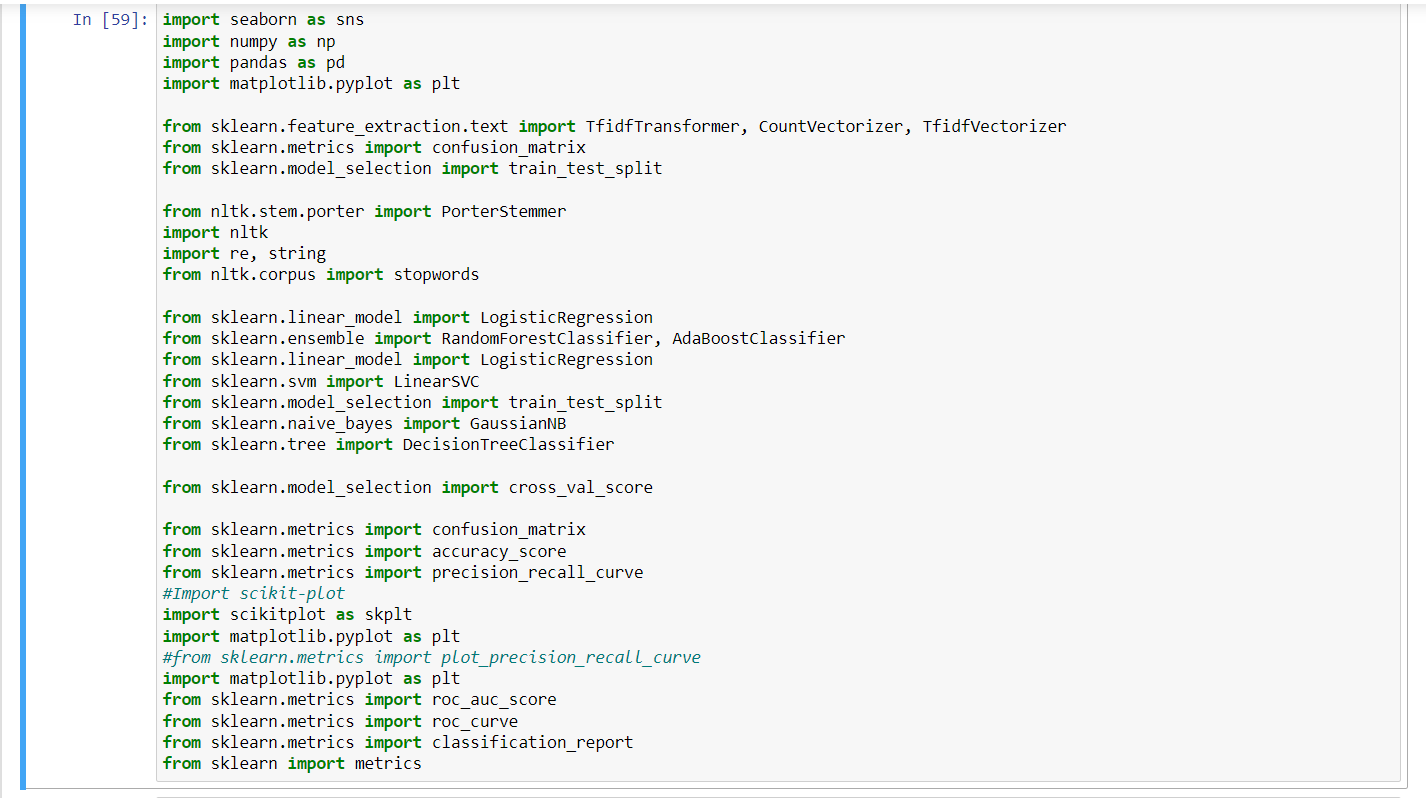


Figure: Importing required Lib

Importing necessary libraries and packages such as seaborn, numpy, pandas, and matplotlib.pyplot for data visualization and manipulation, as well as various machine learning algorithms from scikit-learn library such as Logistic Regression, Random Forest Classifier, AdaBoost Classifier, LinearSVC, Naive Bayes, Decision Tree Classifier, etc.

The code also imports other modules such as PorterStemmer from the Natural Language Toolkit (nltk), and scikit-plot for visualization of machine learning models. Additionally, the code imports confusion matrix, accuracy score, precision\_recall\_curve, classification\_report, roc\_curve, and other metrics from scikit-learn library for model evaluation. This code sets up the necessary environment for building and evaluating various machine learning models for a given dataset, by importing the necessary packages and modules.



Figure: Displaying Data

This code imports the 'warnings' module and uses it to suppress warning messages during program execution. Then, it reads in a CSV file 'labeled\_data.csv' located at the specified file path using the 'pd.read\_csv()' function from the pandas library and assigns it to the variable 'data'. Finally, it displays the first 5 rows of the 'data' using the 'head()' function.

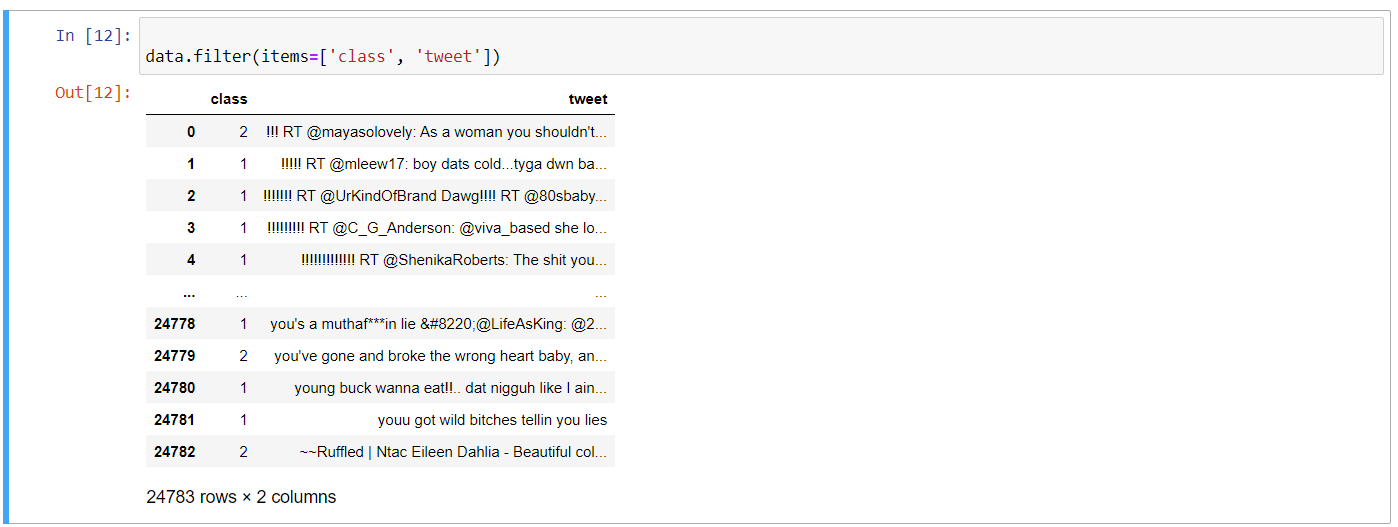


Figure: Filtering data

This code returns a new DataFrame that contains only the columns 'class' and 'tweet' from the original DataFrame 'data'.

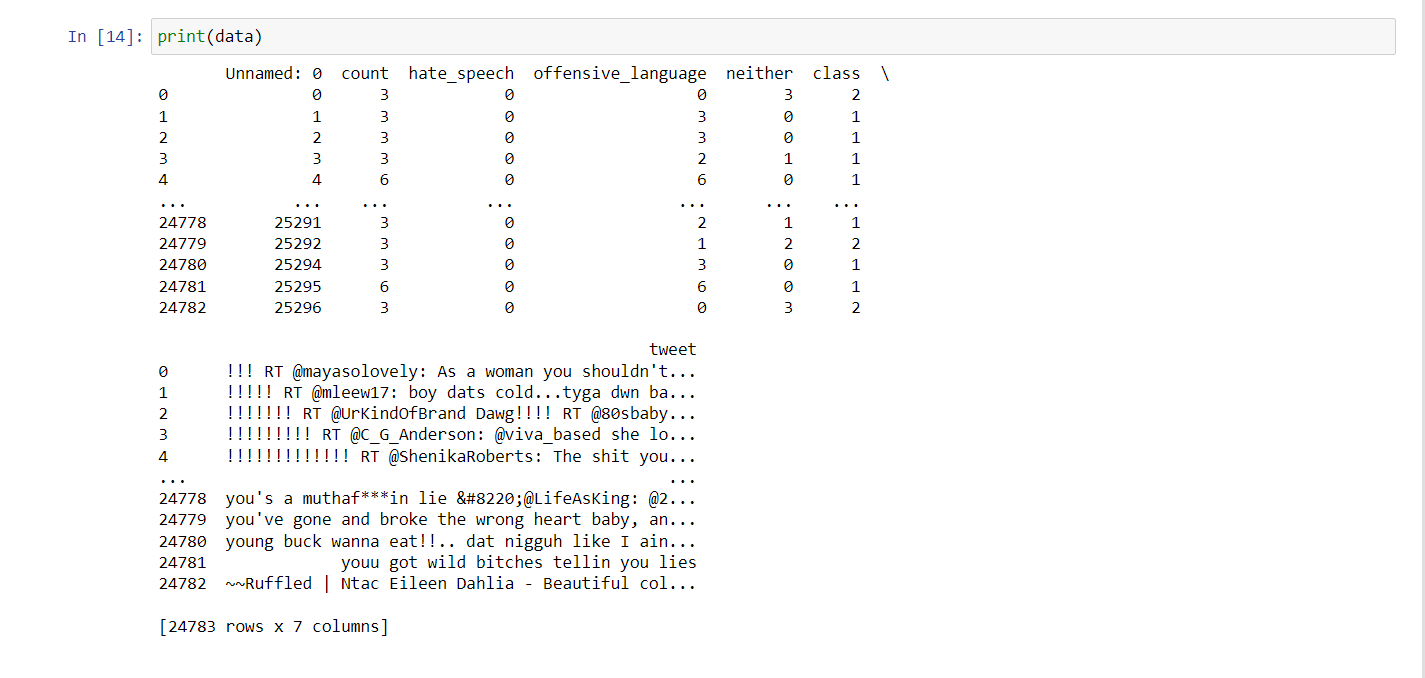


Figure: printing data

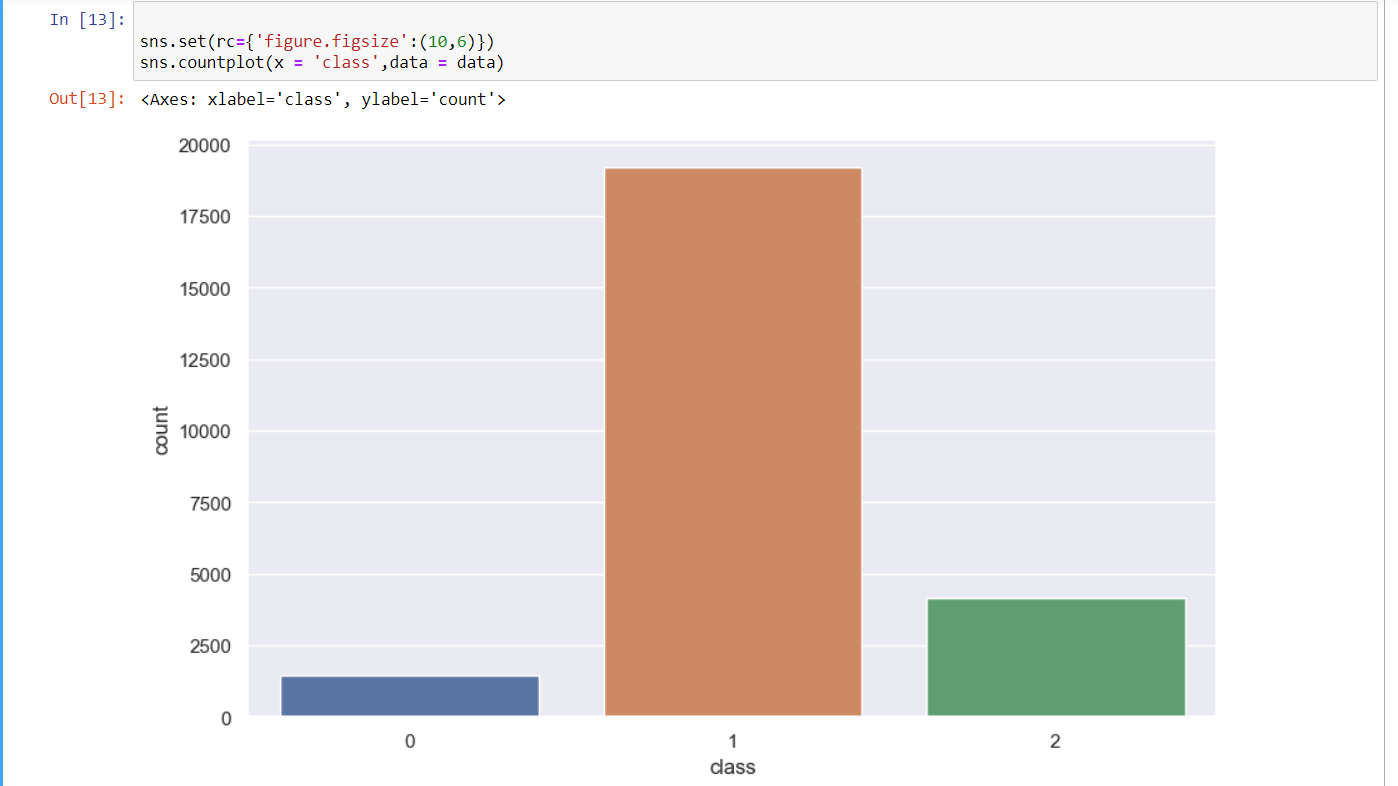


Figure: Data Framing

**Text Preprocessing Functions using Regular Expressions:**

The code provides three text preprocessing functions for cleaning text data using regular expressions:

**case\_convert():** This function converts all the text to lower case. It iterates over all the tweets in the dataset and replaces them with their lowercase equivalents using a list comprehension.

**remove\_specials():** This function removes all special characters and numbers from the text using regular expressions. It iterates over all the tweets in the dataset and replaces them with only alphabets using **re.sub()** method.

**remove\_shorthands():** This function expands common contractions in the text data. It uses a dictionary of contraction mappings to replace them with their full form. It also iterates over all the tweets in the dataset and replaces them with their expanded form.

All the functions make changes to the 'data' object which presumably contains a pandas DataFrame of text data. These functions can be used as a part of a larger preprocessing pipeline for cleaning and preparing text data before applying machine learning models.



Figure: Text pre-processing

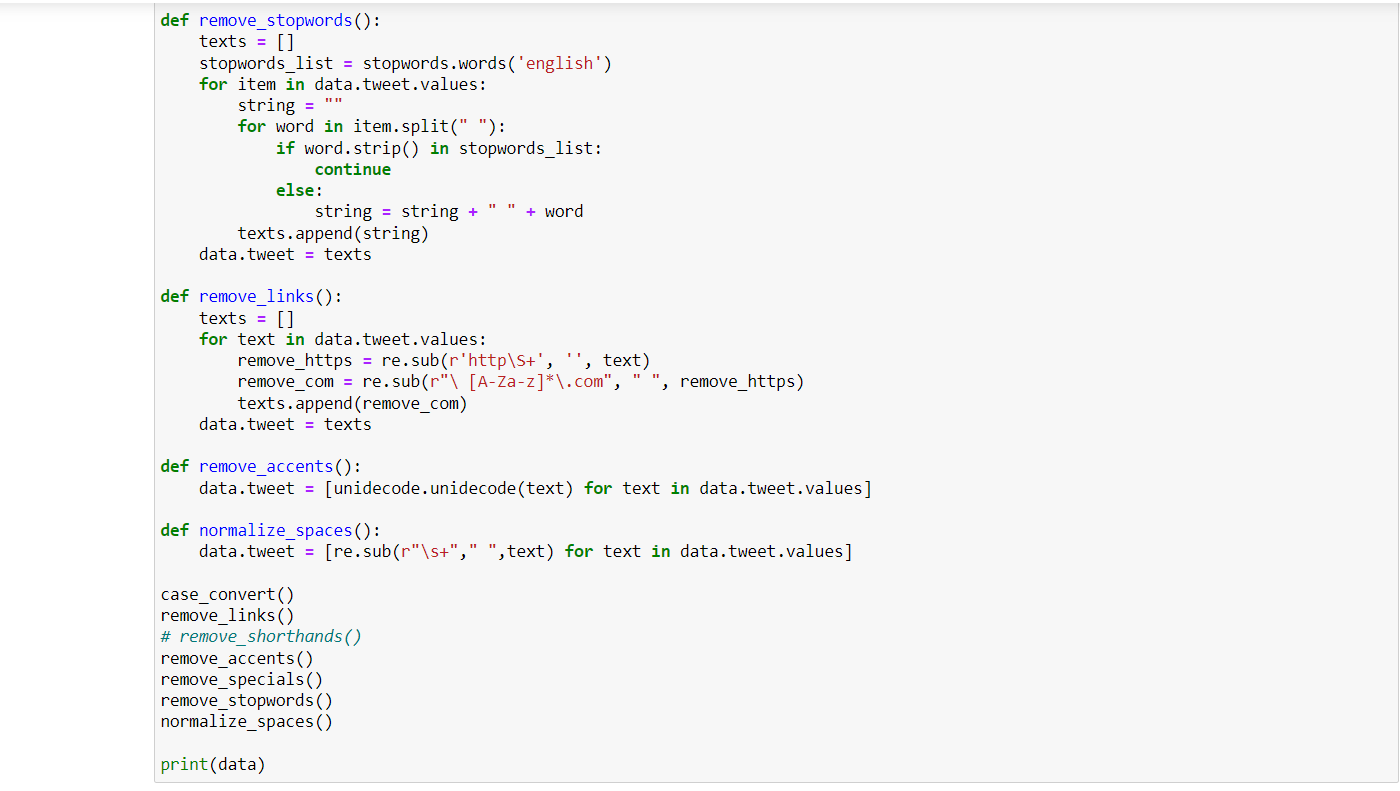


Figure: Text pre-processing



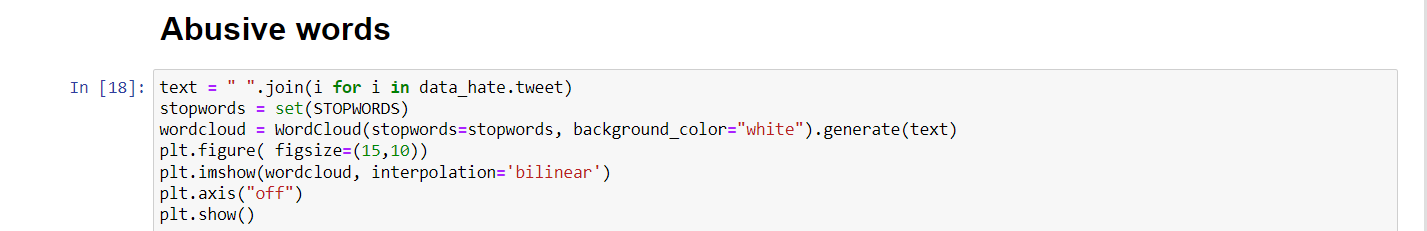


Figure: Implementation of visualizing abusive words



Figure: word cloud visualization of abusive language

Generating a word cloud visualization of the tweets in the data\_hate DataFrame.

First, the tweets are joined together as a single string using a list comprehension and the .join() method. A WordCloud object is then created, passing in the set of stopwords and setting the background color to white.

Finally, the word cloud is displayed using Matplotlib. The figure size is set to 15x10 inches, the image is displayed using bilinear interpolation, and the axes are turned off. The resulting visualization will show the most frequent words used in the data\_hate tweets, with larger words representing more frequently used terms.



Figure: Implementation of visualizing offensive words



Figure: Word visualization of offensive language



Figure: Implementation of visualizing offensive words



Figure: Word visualization of Normal language

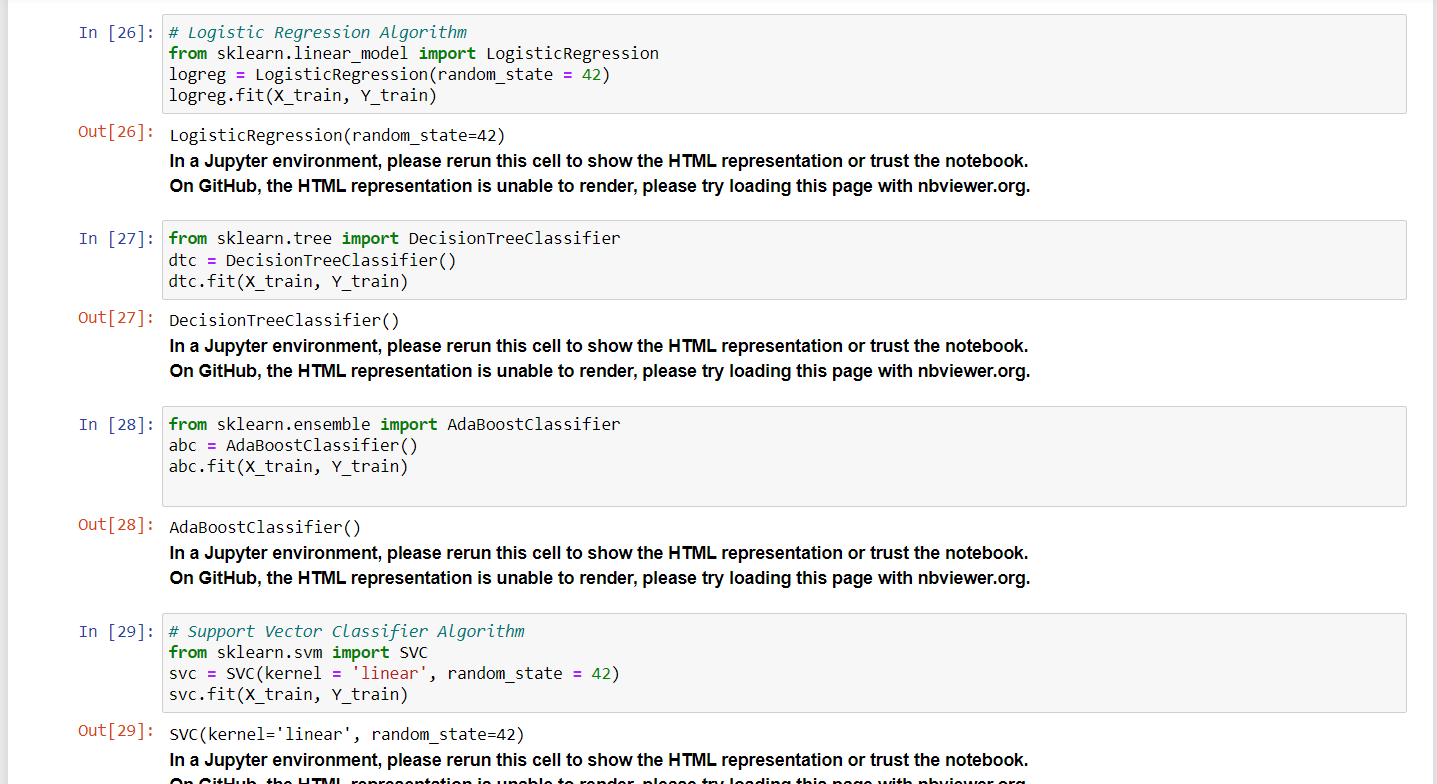


Figure: Training algorithm models

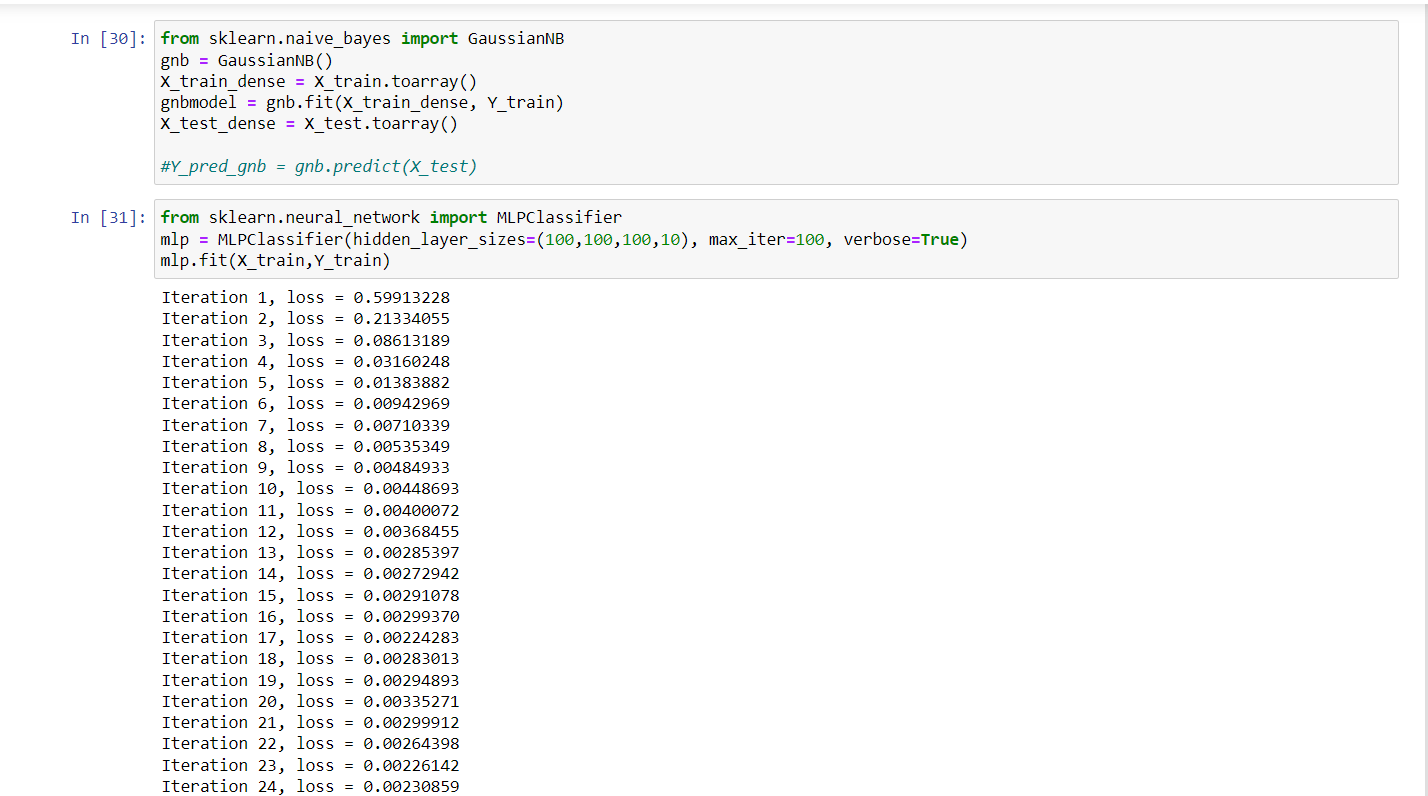


Figure: Training Algorithm Models



Figure: Training Algorithm Models

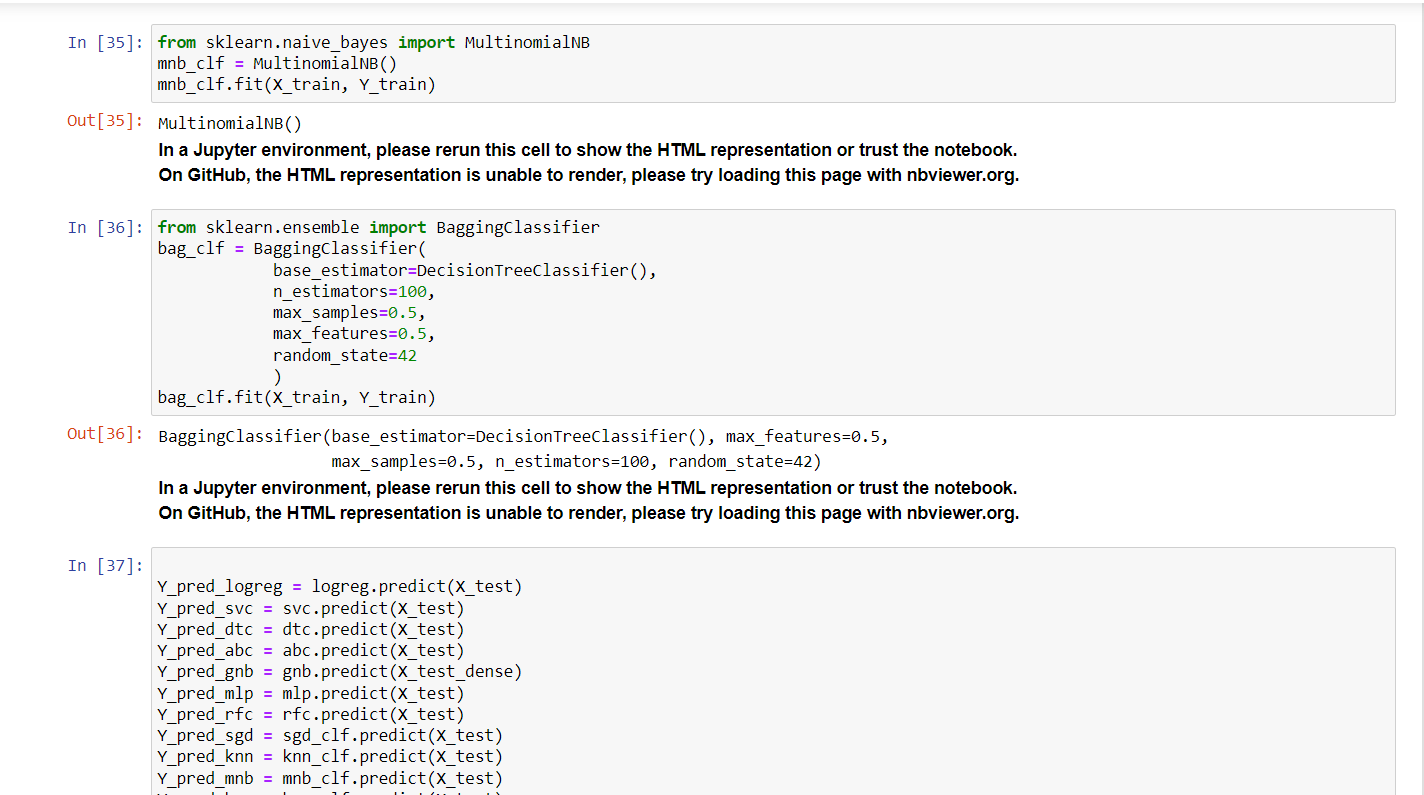


Figure: Training Algorithm Models

Training models on the training data X train with corresponding labels Y train. The LogisticRegression class is transferred from the sklearn.linear\_model module, and an glimpse of this class is created with random\_state set to 42. random\_state is a parameter that determines the random number generation for shuffling the data during training.

The fit technique is then called on the log reg object with X train and Y train as arguments. This trains the logistic regression model on the training data.



Figure: printing accuracies

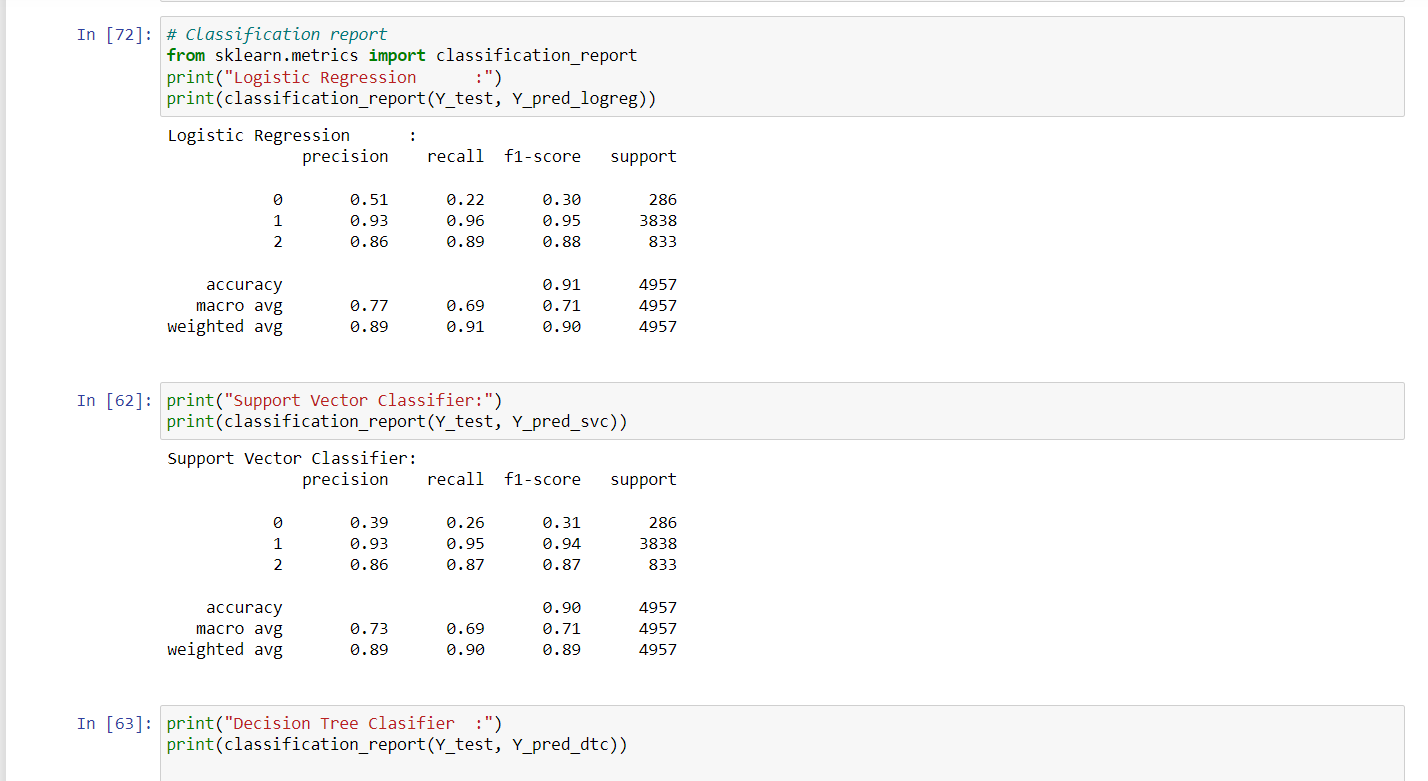


Figure: Scores of Logistic and SVM

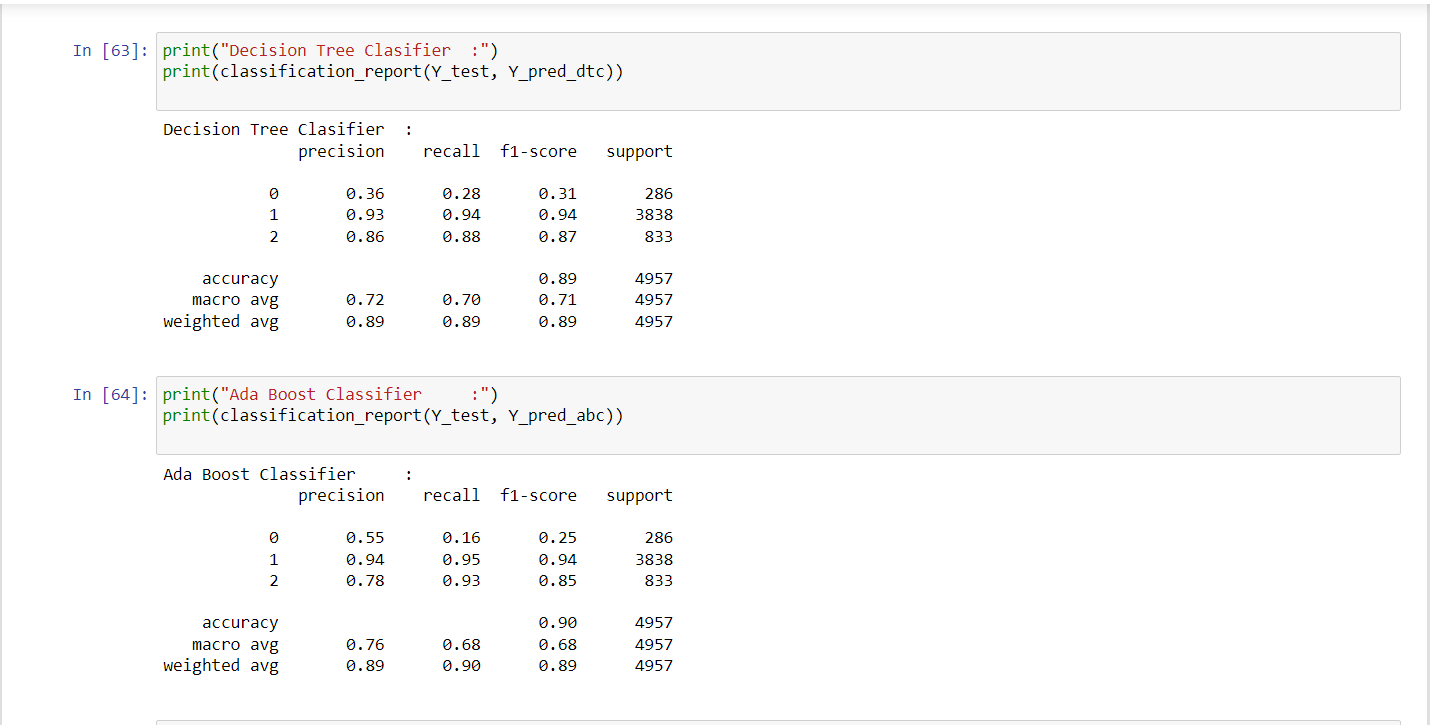


Figure: Scores of DTC and Ada Boost

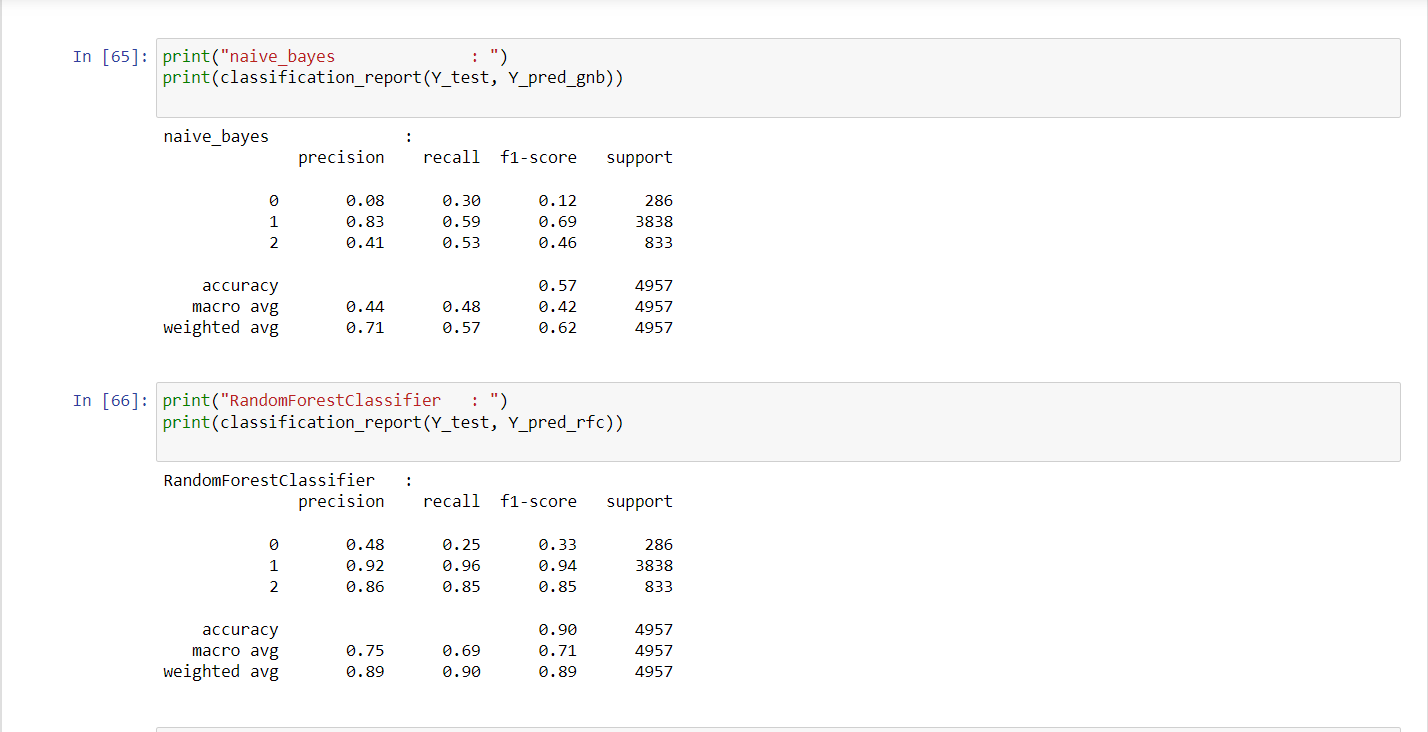


Figure: Scores of Naïve bayes and Random forest

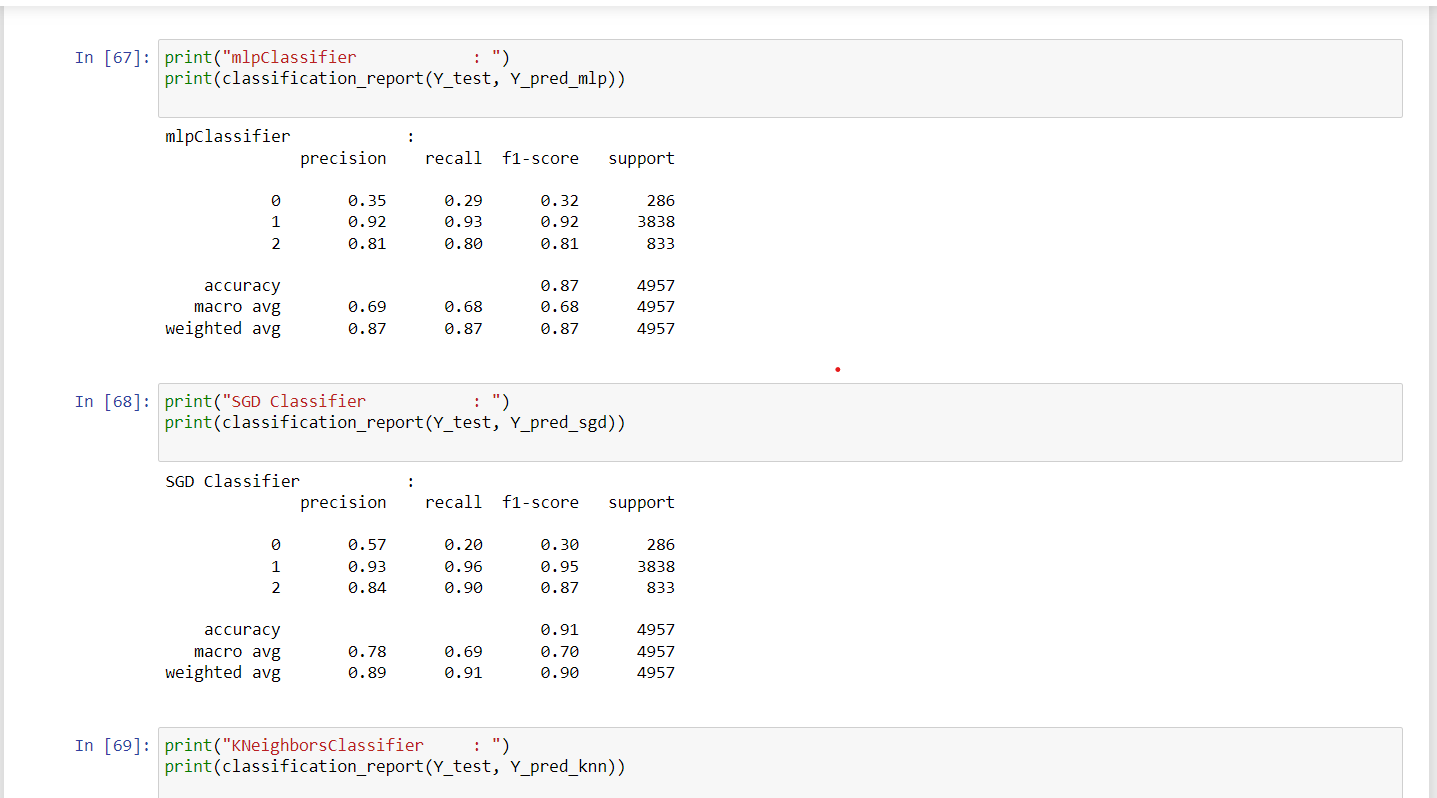


Figure: Scores of MLP and SGD

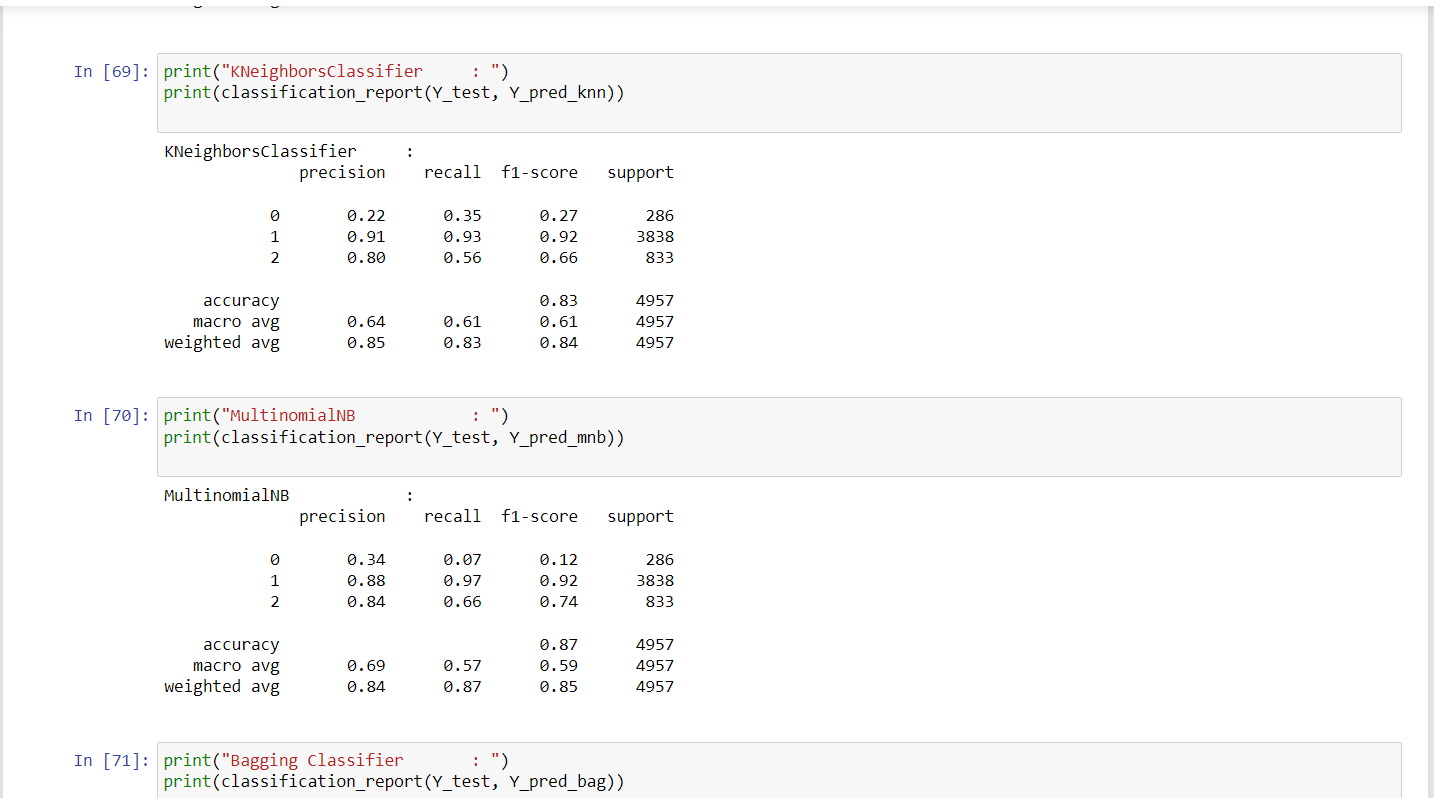


Figure: Scores of algorithms

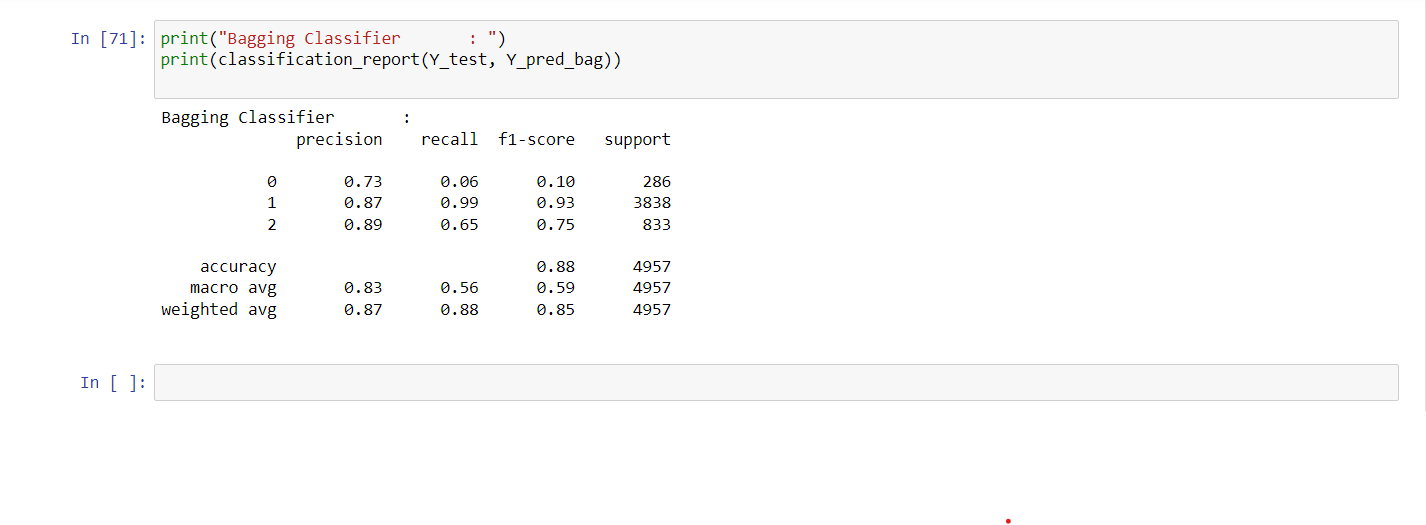


Figure: Scores of algorithm

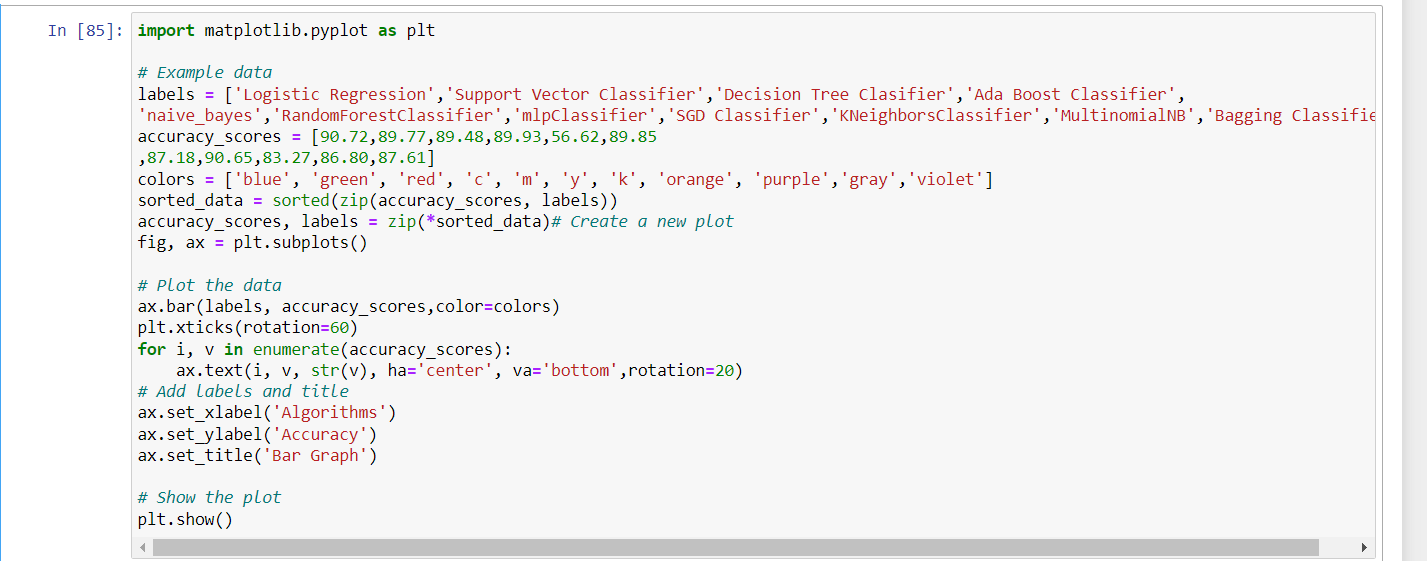


Figure: Bar graph representing

This code creates a bar graph comparing the accuracy scores of different ML algorithms. The x-axis depicts the algorithms and the y-axis depicts the accuracy scores. Each bar represents an algorithm and its corresponding accuracy score. The height of each bar represents the accuracy score, and the color of the bar corresponds to a different color in the "colors" list. The "sorted\_data" variable sorts the accuracy scores and labels in ascending order, which is then used to plot the bar graph. The "plt.xticks(rotation=60)" line rotates the x-axis labels by 60 degrees to avoid overlap. The "ax.text" line adds text labels to the top of each bar indicating the accuracy score. The final "plt.show()" line displays the bar graph.

****

Figure: Saving the model



Figure: identifying abusive as 1

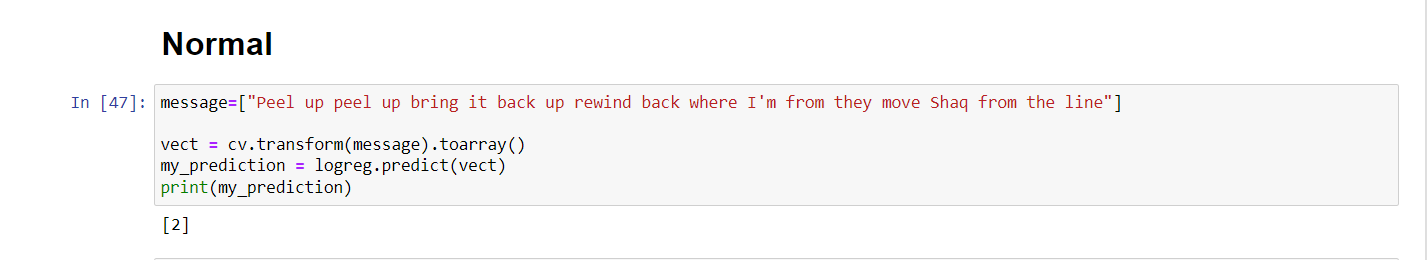


Figure: Identifying normal as 2

## PAPER PUBLICATION STATUS

Our study report was presented at the ICIOT 2023 conference, which was hosted at SRM. Our article was approved as paper id:426.

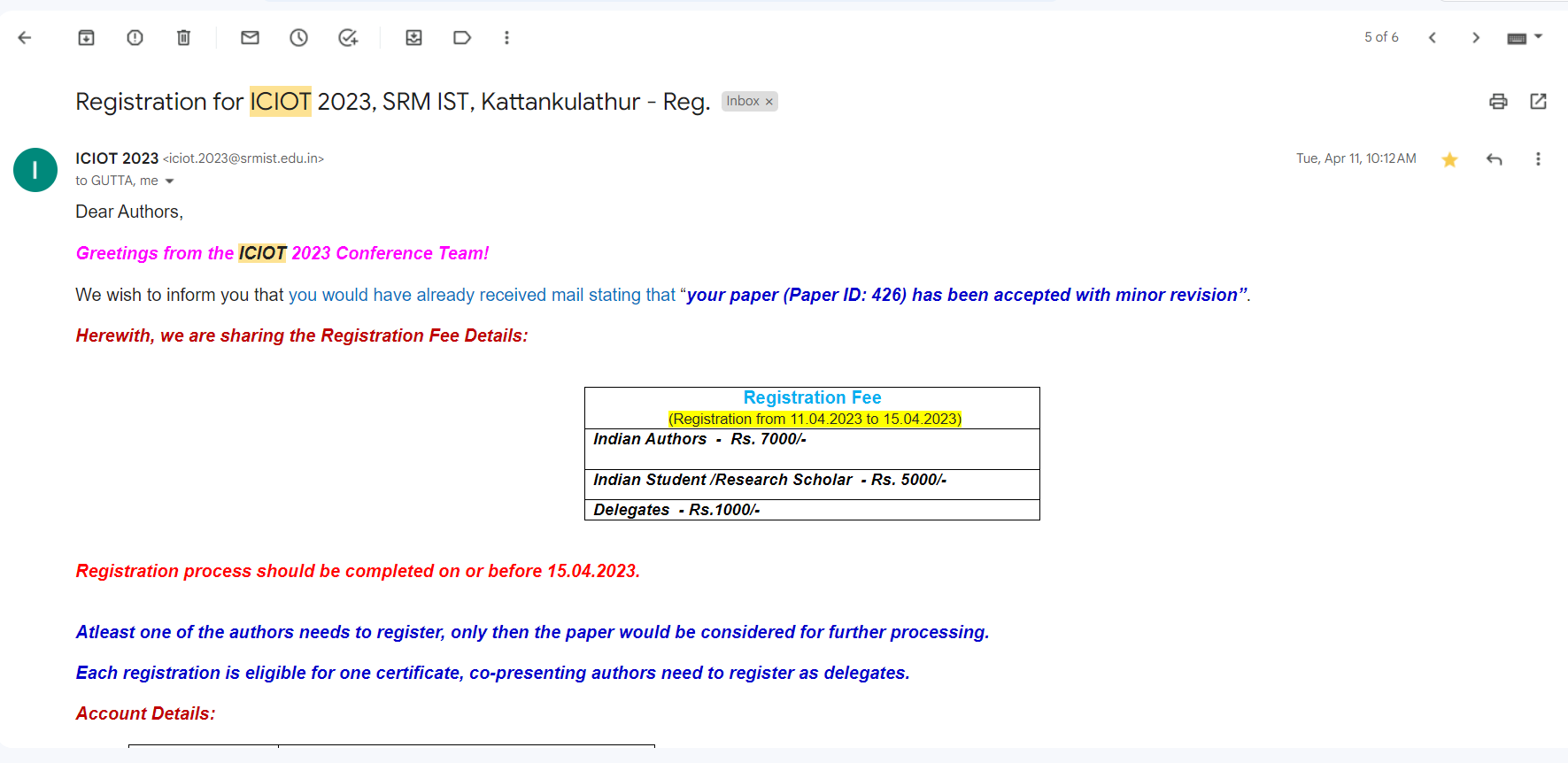


Figure A1: Paper Acceptance

## PLAGIARISM REPORT

