UTILIZING TEXT SENTIMENT AUTOMATIC BENGALI BULLYING DETECTION THROUGH MACHINE LEARNING

NUSRAT JAHAN

MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING
NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

UTILIZING TEXT SENTIMENT AUTOMATIC BENGALI BULLYING DETECTION THROUGH MACHINE LEARNING

NUSRAT JAHAN

This Research report submitted in fulfillment of the requirements for the award of the degree of Master of Science in Information and Communication Engineering

MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING IN NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

AUGUST 2022

ACCEPTANCE

This research report is submitted to the Department of Information & Communication Engineering, Noakhali Science & Technology University, Noakhali, in partial fulfillment of the requirements for having the Master of Science (MSc) degree in Information & Communication Engineering. This research report will be evaluated under the course: Project and Thesis with course code ICE-5312.

_____

Name of Supervisor: Dr. Md Ashikur Rahman Khan
Position: Professor & Chairman
Department of Information and Communication Engineering
Noakhali Science & Technology University
Noakhali - 3814, Bangladesh

_____

Zayed Us Salehin
Assistant Professor,
Department of Information and Communication Engineering,
Noakhali Science and Technology University,
Noakhali- 3814, Bangladesh.

DECLARATION

This research report is submitted to the Department of Information and Communication Engineering of Noakhali Science & Technology University in partial fulfillment of the requirements for having the Master of Science (MSc) Engineering in Information and Communication Engineering (ICE). So, I hereby declare that this report has not been submitted elsewhere for the requirement of any kind of degree, diploma or publication.

Nusrat Jahan
Student of Information and Communication Engineering,
Noakhali Science & Technology University
ID No: BFH 1911MSc102F
Session: 2018-19
Date: 07/08/22

ABSTRACT

The number of Bengali language users on social media is rapidly increasing, and the use of code-mixing and transliteration. Multiple languages mix using code-mixing. Transliterated Bengali allows writing Bengali with Latin words. Manually reviewing and removing bullying content from social media can be time-consuming, which is undesirable in today's technologically automated world. The machine learning approach can be convenient in keeping the system updated with new types of abuser approaches. Machine learning algorithms such as a k-nearest neighbor, Naive Bayes, Support Vector Machine, Random Forest, Logistic Regression, and AdaBoost classification were applied to distinguish transliterated Bengali cyberbullying content. The dataset contained not only transliterated Bengali text but also Bengali and Code-Mixed Bengali text. Baseline features from the social media dataset were used with sentiment and personality features. The features were the number of likes and dislikes, profane words, positive words, negative words, related to posts, and sentiment. TF-IDF and n-grams technique was used for feature extraction. The performances of the algorithms were evaluated utilizing precision, recall, accuracy, F1 score, AUC, and ROC curve. Among sentiment analysis. The result shows that the linear SVM algorithm achieved the highest accuracy of 64.224% to classify sentiment for test data. In contrast, the SVM and RF outperformed other classifiers in detecting bullying comments. For testing data, the SVM and RF achieved 94.83% and 94.40% accuracy, respectively. This work will have an impact on reducing transliterated and code-mixed Bengali cyberbullying.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine learning
AI	Artificial intelligence
CNN	convolutional neural network
DP	Deep Learning
NLP	Natural Language processing
TF-IDF	Term Frequency -Inverse Document Frequency
GRU	Gated Recurrent Unit
SVM	support vector machine
RF	Random Forest,
NB	Naïve Bayes
KNN	k-nearest neighbour
LR	Logistic regression
SMO	Sequential Machine Optimizer
K-FCM	Kernel-based Fuzzy C-Means
GHSOM	Growing Hierarchical Self-Organizing Map
BOF	Bag-of- word
PCNN	Pronunciation based convolutional neural network
RBF	Radial Basis Function
MNB	Multinomial Naïve Bayes
LSTM	Long Short Term Memory
Bi-LSTM	Bi-directional Long Short Term Memory
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
BU	Bullying
Ne	Neutral
NB	Nonbullying
AUC	Area Under the Curve
ROC	Receiver Operation Characteristic

CHAPTER 1

INTRODUCTION

This research study was conducted to find a model that detects Bengali, transliterated Bengali, and English-Bangla code-mixed cyberbullying comments. This chapter discusses the impact of cyberbullying and the challenges of detecting cyberbullying. The background, research motivation, problem statement, objectives, scope, and structure of this research are covered in this chapter.

1.1 BACKGROUND STUDY

Social media has begun to dominate the world's online activities. Millions of people use it for communication and business purposes. Youth spend their time devotedly on social media and share their daily activities. There are more than three million people on social media. This website enables users to communicate with others at any time and from any location. There are some negative aspects to the positive aspects. Cyberbullying and cybercrime are also on the rise due to social media.

According to the National Crime Prevention Council (NCPC), cyberbullying happens online when someone emails or posts text, photographs, or videos intended to hurt or humiliate another person using mobile devices, video game applications, or other platforms. Because of the feature, tracking the bullies or manually deleting the post from social media is burdensome.

Many social media platforms, including Facebook, YouTube, Twitter, Snapchat, Skype, Instagram, and Wikipedia, are used for cyberbullying. Some social media platforms offer tips on how to avoid cyberbullying. Researchers focus on design models to detect and prevent abusive behavior on social media concerning the effect of cyberbullying.

Many researchers have contributed to developing methods for early detection and prevention. Several researchers have used Machine Learning (ML) with Natural Language Procession (NLP). An ML model can learn from experience, identify new types of cyberbullying, and classify them correctly.

1.2 MOTIVATION OF THE STUDY

Bullying through electronic media has become a concern as the internet and social media use for communication and information sharing increases with time. Cyberbullying has become a crucial issue as it is developed alongside technological progress and can have long-term negative impacts on its victims. Figure1.1 shows the effects of cyberbullying among victims.

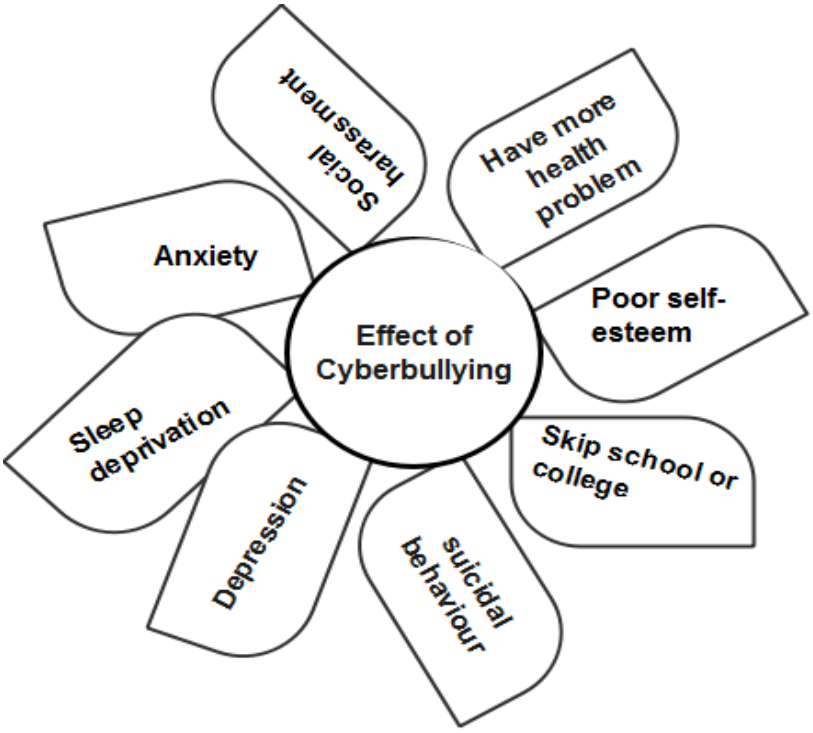


Figure 1.1 The consequences of cyberbullying.

Cyberbullying includes posting mean and hurtful things about someone on social media, threatening someone in a text message, sharing an embarrassing photo of someone without their permission, making fun of someone in an online group chat, etc. Although the majority of the time, a child's Internet use is completely safe and enjoyable, there are risks associated with cyberbullying, suicidal behavior, or pedophile grooming via social media [1].

Bullying can be limited to the schoolyard, whereas cyberbullying can reach its victims at any time by using technological devices such as mobile phones and laptop computers. In addition, online content is widely dispersed and troublesome to remove. Even if a message post once, it can be re-posted, liked, or shared multiple times, significantly amplifying the impact of an offensive or hurtful message. Action against cyberbullies takes longer because of the medium through which cyberbullying is perpetrated [2].

Some university students were perplexed about the seriousness of cyberbullying, believing it to be merely a prank rather than a crime or severe event [3]. Cyberbullying at university may be the same as cyberbullying at school. An undergraduate student, a victim of cyberbullying, emerges as an additional risk factor for the blooming of depressive symptoms, and depression can lead to suicidal ideation.

Cyberbullying has grown in prominence in recent years. According to the Florida Atlantic University survey in 2017 [4], 83% of those who cyberbullied were also bullied. 69% of those who admitted to being intimidated online have also admitted to in-person bullying. As shown in a 2018 survey conducted by Pew research center, 95% of teenagers connect to the internet, and 85% use social media [5].60% of teenagers have been victims of cyberbullying, and 87% of young people have seen cyberbullying occurring online.

In a CRC survey released in 2019, around 37% of students reported having been the victim of cyberbullying at some point [6]. In 2019, 17.4 percent of students were victims of cyberbullying, up from 16.5 percent in 2016 [7].

Bangla is the seventh spoken language worldwide, with approximately 228 million native speakers. Most Bengali native speakers use transliteration-based Bengali text on social media. Because of this, it creates trouble to detect bullying. The majority of people use hate speech during bullying. There are numerous resources for detecting English text-based cyberbullying [8]. Limited research has been performed to detect transliterated Bengali text-based bullying.

Jahan et al. utilized a small number of transliterated Bengali comments (around 200 abusive comments) in their study [9]. They take both Bangla-English codes mixed and transliterated in concern. Only three machine learning algorithms, Support Vector Machine, Random Forest, and Adaboost, were used for offensive speech detection. They imply that the

proposed method achieved the highest accuracy of 72.14% for the Random forest algorithm. One can get more precision by using other efficient algorithms.

It is clear that cyberbullying is a vital issue in today's society, and it is essential to look at how we can detect and prevent it more efficiently. Research investigating how can develop a multilingual-multi performer cyberbullying detection system. Therefore, this research focused on finding a model that can detect all Bengali-related bullying comments from social media.

1.3 PROBLEM STATEMENT

Cyberbullying is a classification problem. The algorithm's performance depends on how correctly it detects original bullying and identifies not bullying text. It is challenging to classify offensive language or bullying text. Davidson et al. distinguish between hate speech and foul language [10].

There are some countries where hate speech is a crime punishable by imprisonment. In other countries, such as the United States, where free speech is a constitutional right, removing hate speech poses a challenge for social media platforms. Most people do not consider hate speech or threats to be bullying. Hence it is difficult to filter the bullying text and standard offensive text. Table1.1 summarize multiple challenges for automatic detection of text-based cyberbullying.

Table 1.1 Challenges to detection social media text-base bullying

Challenges	Explanation
Code-switching or mixing	User use mix language. For instance, “ আপনাকে দেখতে autistic লাগছে”
Linguistic diversity	The meaning of the same word or sentence can differ based on the use.
Contextually embedded	Bullies use threat speech contextually embedded so that what in one community is perceived offensive is not so in another.
Fault detection of sentiment	The system can fail to detect bullying as users can subtly change their tone to fool the systems.
Freedom of speech	Misclassification can result in limiting individuals freedom of expression.

A lot of textual and non-textual content, as well as knowledge on aggressive behavior, may be found on social media sites. Because of the structure of social media websites, cyberbullying can occur in private, extent quickly, and last indefinitely [11]. The majority of victims and bystanders do not report cyberbullying on time. As a result, automated technologies, such as Machine Learning (ML), may play an essential role in detecting cyberbullying. Also, a model for preventing cyberbullying is of practical significance. However, preventing cyberbullying model requires significant features.

Social features can aid in the detection of cyberbullying. By analyzing the social network structure between users and deriving elements such as the number of friends, network embeddedness, and relationship centrality, the author discovers that integrating textual features with social network features significantly improves cyberbullying detection [12]. Nevertheless, there are always some challenges to detecting textual bullying comments from social media.

The transliterated text can consist of code-mixing. Code mixing allows mixing two or more languages. Most of the text on social media websites is unstructured manner. As a result, the lexicon-based approach is not so preferable [13]. The authors explained cod-switching hate speech [14], [15]. They addressed the phenomenon of code-switching at word-level or sentence level to identify hate speech in bangle text.

Using machine learning to identify and define the pattern and correlation of data with the lexicon approach improves performance. Through the assessment of the previous conducted research work, it found:

a) Have lack of transliterated cyberbullying dataset.
b) Difficult to distinguish sentiment of the comment
c) No standard system to detect multilanguage-multiplatform cyberbullying detection system.
d) Lake of a model that can detect all types of Bengali text bullying.

As a remark, most existing works consider text speech written in the Bengali alphabet. Few researchers focus on transliterated-based Bengali threatening or bullying speech detection. The main reason for the small amount of research work and accuracy is the less availability of existing datasets on transliterated Bengali-English mixed text.

1.4 RESEARCH OBJECTIVES

The purpose is to find the best performance by providing an algorithm for detecting Bengali-English code-mixing and transliterated text-based cyberbullying. Hence the main objectives of this research are:

i. To create a corpus that contain transliteration-Bengali, code-mixed Bengali and Bengali cyberbullying text.
ii. To extract sentiment from any transliterated Bengali text
iii. To see transliteration-based Bengali text cyberbullying through various machine learning algorithms.

To investigate the most effective algorithm for detecting bullying based on precision, recall, and accuracy value.

1.5 SCOPE AND LIMITATION OF THE STUDY

The study focused on detecting transliterated Bengali, code-mixed Bengali, and Bengali cyberbullying. The data was collected from social media like Facebook, Instagram, and Youtube. Web scraping toll Octoparse and Instant data scraper were used to gather data. The study was carried out using Python 3.7 programming language. To detect the language of the comment, Google Translate API was used. OneHotEncoder for categorical data, n-gram, and TF-IDF for user comments were applied for feature extraction. Considering sentiment feature increases the accuracy of the bullying. Hence, first, applied a multilingual VADER sentiment analyzer. However, it provides some error detection. So, the ML model was also used to detect bullying comments. The analysis of the model efficiency was based on precision, recall, accuracy, and f1 score.

There is some limitation to this research study. Considering the number of emojis, the type of emoji can also change the sentiment of the comment. Some comments could not be classified ideally. There Could not take into consideration which kind of sentiment it indicates. The use of more features to train the model can be helpful for models to learn and detect this type of comment.

On the other hand, only classify comments based on the presence of the word. Nevertheless, combining positive and negative words can change the classification model. Implementing of Neural network can increase the accuracy level.

1.6 WORKFLOW

Through Figure 1.2, depict workflow from start to end of the proposed research work.

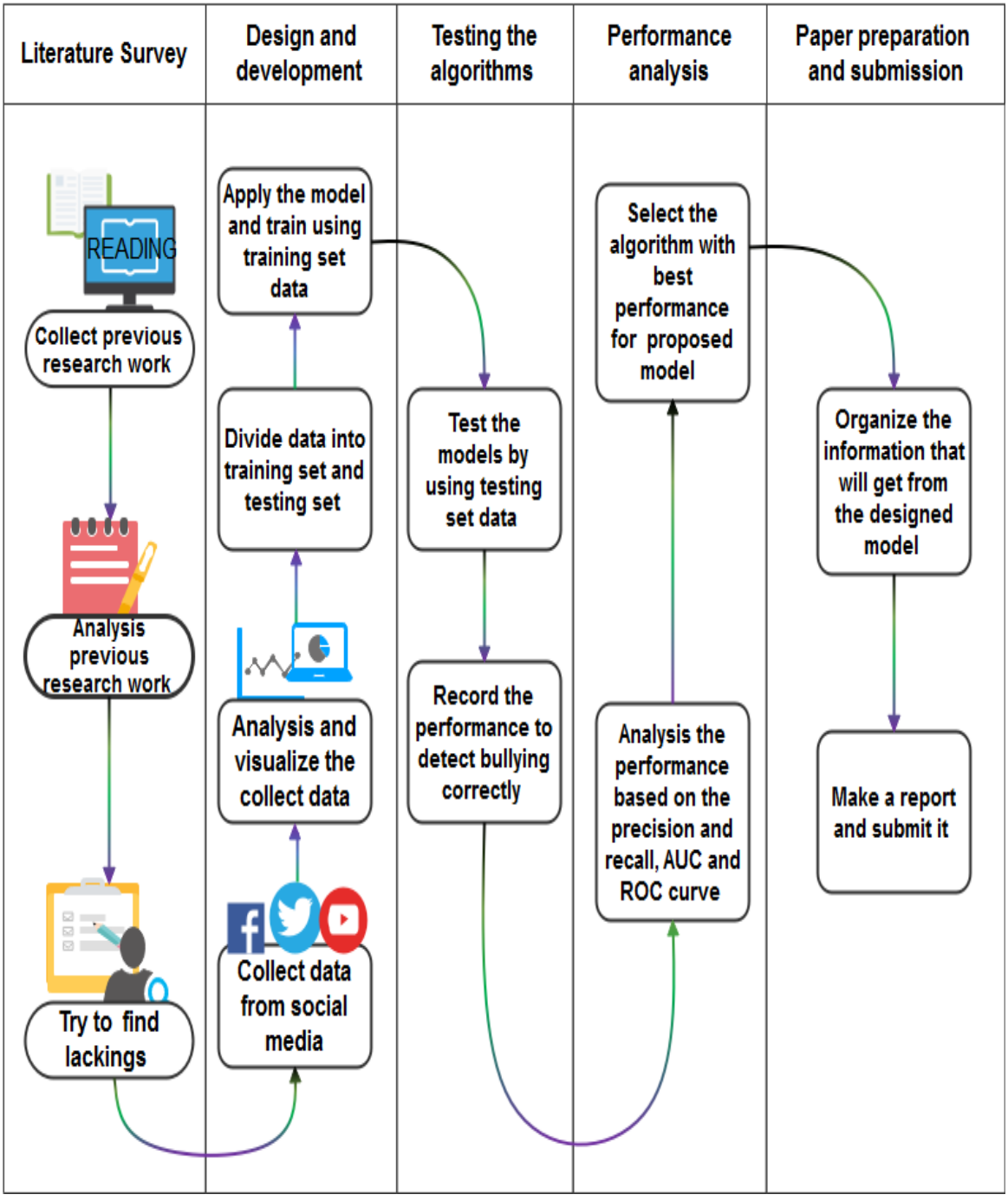


Figure 1.2 Workflow of this thesis study

The literature survey section collected previous research and attempted to analyze the gaps in those works. Data were gathered and processed to use as input machine learning

algorithms. Data was taken in two parts, training set and testing set data. Train the model in the design and development phase using the training dataset. The testing dataset was used in the testing phase to observe how well the system detects bullying and non-bullying content. Then based on the best performance with the highest precision and accuracy, select the algorithm for the prevention model. Finally, we generated a report based on the results.

1.7 THESIS OUTLINE

There are five chapters, including this Introduction chapter. Figure 1.3 shows all chapters. The introduction chapter explained the problem statement, the objectives, and the limitation of the thesis. The rest of the paper describes several related works, methodology

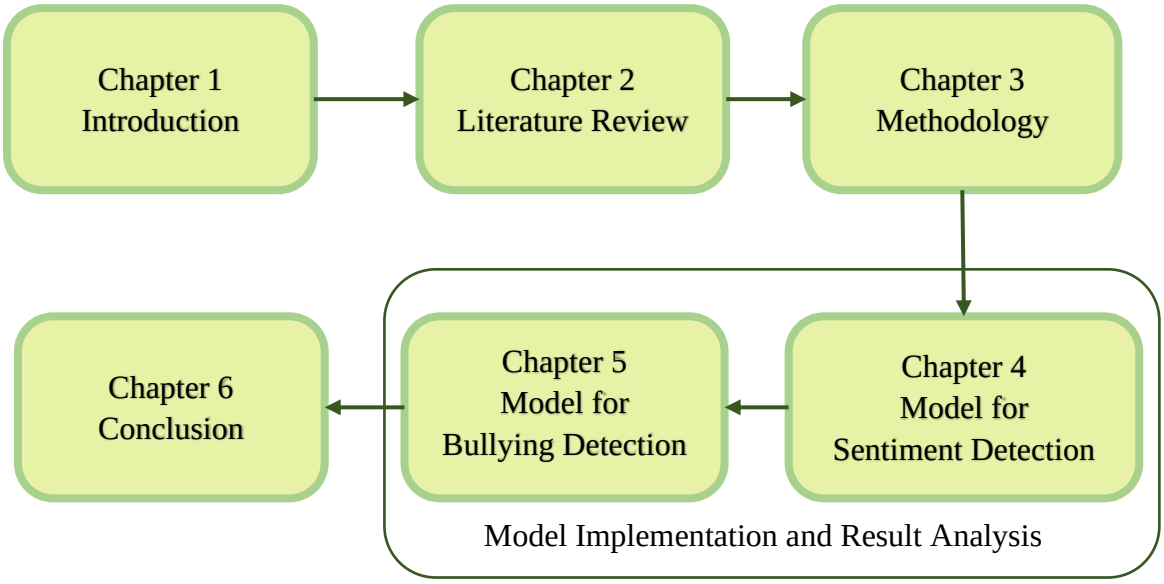


Figure 1.3 Chapters in this thesis report.

to reach the desired target, result, and conclusion. The methodology chapter describes the way of data collection and annotation, using the procedure of the algorithms and process to analyze performances. After the methodology chapter, two chapters describe the model implementation and interpret the model's outcome. The first one is 'The model for Sentiment Detection' interpret the result to detect the sentiment of the Bengali text. The second is about bullying detection employing the sentiment of the Bengali text. The final chapter, Conclusion, narrate this paper and outlines the scope for future work.

CHAPTER 2

LITERATURE REVIEW

Cyberbullying is spreading quickly and affecting a large number of people. Many countries are becoming more aware of its impact. Additionally, many researchers present an auto-detection and prevention model by using machine learning. The issue is that the majority of the work to detect cyberbullying in English. There have only been a few studies on code-mixing and transliterated Bengali cyberbullying. This literature review examines the automated cyberbullying detection and prevention techniques for different language-based bullying.

2.1 TEXT-BASED CYBERBULLYING DETECTION

The amount of hate, aggression, and bullying on social media is steadily increasing. The researcher used Machine Learning, Deep Learning, and Artificial Intelligence to detect and prevent those bullying texts. In addition, research on this subject is ongoing. Survey research was conducted to define the 'approaches to identify cyberbullying automatically [16]. They imply that researchers employ supervised learning, lexicon-based, rule-based, and mixed-initiative approaches.

Kumar et al. employ supervised ML algorithms to identify and detect the presence or absence of cyberbullying in YouTube video comments [17]. They take comments but also discussion on the reply comment into consideration. They collect data using social media API for comments. They get KNN gives the best accuracy of 83%, precision of 0.824 among KNN, Naïve Bayes, RF, and Sequential Machine Optimizer (SMO).

On the other hand, the author suggests a semi-supervised approach [18]. They proposed a fuzzy SVM classification algorithm, kernel-based Fuzzy C-Means (K-FCM) clustering algorithm for the fuzzy model. Clustering is a technique used to discover natural groping

among similar objects. Among multiple tests, they get the best precision of 55%, and F1 measure 47% for their kernel-based fuzzy SVM (K-FSVM). It is observable that their proposed algorithm performs overall best than Naïve Bayes, Random Forest, and logistic regression

.In other ways, M. D. Capua et al. discover that using the unsupervised technique yields acceptable results [16]. They use the Growing Hierarchical Self-Organizing Map (GHSOM). For a Twitter dataset, they achieve better accuracy (72%) better than the Nave Bayes.

There is also numerous research based on Convolution Neuron Network (CNN), Deep learning for detecting social media cyberbullying. Table 1 summarizes existing research on the detection of cyberbullying on popular social media platforms such as Facebook, Twitter, YouTube, and Instagram.

2.2 ANALYSIS CODE-MIXED SOCIAL MEDIA TEXT

There is a lack of a dataset for code-mixed social media text. A. Bohra et al. give a corpus for Hindi-English code-mixed social media text for hate speech detection [19]. They collect data from Twitter posts. They use Character N-Grams, Word N-Grams, punctuations, Negation words, and Lexicon features to train supervised machine learning models. With the Random Forest classifier, they get 66.7% accuracy. The accuracy is improved to 71.7% when they apply SVM classification.

In addition, Ghosh et al. analyzed Bengali-English-Hindi code-mixed social media text for sentiment identification [20]. They use a dataset from International Conference on Natural Language Processing (ICON-2015). The data was collected from Facebook pages. They consider multiple features and take them into three groups, namely, word-based features, syntactic features, and style-based features. They obtained the best accuracy of 68.5% when they merged word-based and semantic-based features.

Santosh et al. use deep learning to detect Hindi-English code-mixed hate space from a Twitter dataset [21]. Santosh et al. In light of the possibility of using out-of-vocabulary words, Long Short-Term Memory (LSTM). They looked into two methods: sub-word level LSTM models and hierarchical LSTM models with phonemic sub-word attention. SVM has the highest accuracy (70.7%). The hierarchical LSTM model has the best recall (45.1) and F1 score (48.7).

Alike Sreelakshmi et al. use the SVM and Radial Basis Function (RBF) classifier to compare the performance with SVM and Random Forest [22]. They imply that character-level features provide more information than word and document levels for code-mixed text classification.

2.3 BULLYING CONTENT ANALYSIS IN BANGLA TEXT

Puja et al. employed Multinomial Naïve Bayes (MNB), SVM, and Convolutional Neural Network (CNN) with LSTM classifiers [23]. They leveraged both emoticons and Bengali characters as input. They found that SVM with linear kernel performed best with 78% accuracy.

Awal et al. apply Naive Bayes (NB) classifier to detect abusive content in Bengali [24]. They collected text from YouTube and provided the performance of NB using 10-fold cross-validation. They collect English comments from YouTube. They use natural language processing techniques such as normalization and stemming from removing unwanted strings, like URLs, IP addresses, or other unique arrays of characters. Then create an actual Bangla text dataset from those preprocessed English comments. They obtained an accuracy of 80.57 % and an F1 score of 0.39.

Romim et al. collected 30,000 Bengali comments from YouTube and Facebook comment sections and 10,000 hate speeches [25]. Ran severanWord2Vec, FastText, and pre-trained BengFastText. Linear kernel Support Vector Machine (SVM) for baseline evaluation was used. 100 layer Long Short-Term Memory (LSTM) and 64 layers of Bi-directional Long Short-Term Memory (Bi-LSTM) are also used. According to their findings, SVM has the highest accuracy of 87.5 and an F1 score of 0.911.

Hussain et al. collected 300 comments from Facebook and an online newspaper for abusive content detection [26]. They proposed a weighted-rule-based method that utilized labeled data. Additionally, Eshan and Hasan investigated the performance of RF, MNB, and SVM classifiers for abusive language detection using unigram, bigrams, and trigram-based feature vectors [27]. They found that the SVM classifier with linear kernel and tri-gram features showed the highest accuracy.

Emon et al.use machine learning and deep learning algorithms [28]. They use linear support vector classifier (LinearSVC), logistic regression (LR), multinomial Naive Bayes (MNB),

random forest (RF), artificial neural network (ANN), and recurrent neural network (RNN) with a long short-term memory (LSTM) to detect multi-type abusive Bengali text. Among all other machine learning and deep learning algorithms, RNN with LSTM cell outperforms in detecting Bengali-abusive text. They also included a stemming rule to aid the classifier's performance.

2.4 BULLYING CONTENT ANALYSIS IN TRANSLITERATED BENGALI

Most social media users use transliterated text in the comment section or message. It is now necessary to consider how to detect transliterated Bengali text-based bullying. Jahan et al. detect abusively or hate speech using transliterated Bengali-English text [9]. They make use of SVM, Random Forest, and Adaboost. The data was gathered from popular Facebook pages. For their system, they used python's sci-kit-learn library. With accuracy and precision scores of 0.7214 and 0.8007, respectively, the Random Forest classifier outperformed the other classifiers for their proposed model. Adaboost had the highest recall with a score of 0.8131.

Sazzed presents an annotated corpus of 3000 transliterated Bengali comments divided into two categories, abusive and non-abusive, with 1500 comments for each [29]. They use SVM, Logistic Regression, BiLSTM, and Random Forest. Among all of them, SVM has the highest average of (0.865 0.15).

Additionally, Mridha et al. proposed a hybrid model to detect offensive text from social media posts [30]. They combine the AdaBoost algorithm with LSTM. Their model provides 95.11% accuracy. FastText, BERT, TF-IDF, and word2Vec feature extraction techniques were used. Initially, baseline model like linear kernel SVM, Decision Tree, Random Forest, and LSTM was applied to train and test the model for improper text classification. The LSTM model achieved the highest 94.53% accuracy for the FastText vector model.

The studies do not specify how they annotate these comments or what definitions they use to annotate for the specific class. Furthermore, individuals did not cite any feature, data engineering methodology, or detailed dataset description. They did not mention how long the sentences were or the words used. The summary of the literature on the above-discussed four types of cyberbullying is shown in Table 2.1.

Table 2.1 A survey on existing research in detecting cyberbullying

Author	**Supportive language**	**Technique**	**Evaluation / Performance**
Jahan et al. [9]	Code-mix & transliterated Bengali	SVM, Random Forest, and Adaboost	Random Forest provides the best performance. Accuracy of 0.7214 and precision of0.8007
A. Kumar et al. [17]	English	KNN, Naïve Bayes, RF and Sequential Machine Optimizer (SMO).	The best accuracy of 83% for KNN
Vinita Nahar et al [18]	English	Semi supervised approach	Precision of 55% and F1 measure 47% for their kernel-based fuzzy SVM (K-FSVM)
A. Bohra et al. [19]	Code-mix (Hindi-English)	Random Forest and SVM	66.7% accuracy for RF and 71.7% accuracy for SVM
S. Ghosh et al. [20]	Code-mix (Hindi-English)	Natural language processing (NLP)	Accuracy of 68.5%.
Santosh et al. [21]	Code-mix (Hindi-English)	Deep Learning	Proposed Hierarchical LSTM model provide highest recall (45.1) and F1 score (48.7)
Sreelakshmi et al. [22]	Code-mix (Hindi-English)	Support Vector Machine (SVM)-Radial Basis Function (RBF) classifier, Linear SVM, Random Forest	SVM-RBF give 85.81% accuracy. character-level features provide more information than word and document level for code-mixed text classification
C. Puja et al. [23]	Bangla	Multinomial Naïve Bayes (MNB), SVM, Convolutional Neural Network (CNN) with LSTM classifiers	SVM with linear kernel provide best 78% accuracy.
Awal et al. [24]	Bangla	Naive Bayes (NB)	Accuracy of 80.57 % and an F1 score of 0.39

Author	**Supportive language**	**Technique**	**Evaluation / Performance**
N. Romim et al. [25]	Bangla	SVM, 100 layer Long Short-Term Memory (LSTM) and 64 layers (Bi-LSTM)	SVM has the highest accuracy of 87.5 and an F1 score of 0.911
Hussain et al. [26]	Bangla	Weighted-rule based method	Utilized labelled data
Eshan and Hasan [27]	Bangla	RF, MNB, SVM	SVM provide highest accuracy
Emon et al. [28]	Bangla	Machine learning and deep learning	Recurrent neural network (RNN) with LSTM provide better performance
S. Sazzad [29]	Bangla	SVM, Logistic Regression, BiLSTM, and Random Forest.	SVM has the highest average of (0.865±0.15).
R. Zhao et al. [31]	English	Embeddings-enhanced Bag-of- word, Linear SVM	Precision 76.8%, Recall 79.4%, F1 score 78.0%
E.V. Altay et al. [32]	Turkish	Bayesian Logistic Regression, Random Forest, multilayer perceptron, J48 and SVM	Bayesian Logistic Regression and Random Forest provide better result
Xiang Zhang et al. [33]	English	Pronunciation based convolutional neural network (PCNN)	Accuracy 96.8% for Formspring data, 98.9% for Twitter dataset
Mangaonkar et al. [34]	English	Naive Bayes Logistic Regression SVM	Naive Bayes and the Logistic Regression algorithms are more suited for detecting cyberbullying behavior than the SVM
M. Dadvar et al. [35]	English	Hybrid approach	Best AUC 0.76
H. Hosseinmardi et al.[36]	-	Naive Bayes, linear Support Vector Machine	SVM classifier best accuracy of 0.87

According to the literature review, numerous studies are ongoing to detect text-based cyberbullying. However, the majority of the work focuses solely on the English language. In contrast, there are many countries where people do not speak English as a first language. They are using media in their native language, and cyberbullying is rising in those countries. To consider it, researchers provide a distinct method to detect language-based bullying, like Turkish, Hindi, Bengali, and Spanish. Most researchers obtained the highest accuracy for the Support Vector Machine (SVM) algorithm.

In today's social media, many people use code-mixing and transliterated text. Some effort was made to comprehend and identify code-mix bullying. However, there are only a few works on transliterated text-based bullying. Bengali native speakers, a region in the northeastern part of India and Bangladesh, use transliterated Bengali text on social media.

2.5 SUMMARY

Nevertheless, problems like online abuse and especially online abuse towards women are continuously rising in Bangladesh [35]. Just a few numbers work on transliterated Bengali text-based bullying detection. The lack of a publicly available dataset on transliterated Bengali abusive text is one of the main reasons. The following chapter will discuss the methodology to make the corpus of transliterated Bengali abusive text and how to detect transliterated Bengali text-based cyberbullying.

CHAPTER 3

METHODOLOGY

This methodology chapter describes how data is collected from several social media sites and utilized. The description of the data collection formula is given in this chapter. Furthermore, the feature extraction technique and classification models are discussed. The end of the chapter mentions the model performance analysis parameters.

3.1 DATA COLLECTION AND PROCESSING

The first step of this research work is dataset creation. There is some Bengali hate space, and the abusive comment detection public dataset is available online. However, any publicly available Bengali transliterated cyberbullying dataset is not found. So, a corpus has been created as there is a lack of an existing dataset on transliterated or Bangla-English code-mixed bullying text. This corpus contains user-generated and some manually generated transliterated Bengali text. The first step in creating a dataset is to gather information from social networking sites. Following this data collection phase, the data has to be filtered and cleaned.

3.1.1 Data Collection

According to the Statcounter Global Stats website stats, Bangladesh's most popular social media platform is Facebook, ahead of YouTube, Twitter, and Instagram [37]. So we select Facebook, Youtube, and Twitter social media sites to collect our data. Octoparse, a web scraping tool, was used to collect user comments from social media sites. For some cases, we also use the Instant Data Scraper extension to extract data. Figure3.1 illustrates how we collected the abusive and bullying comments on social media.

Comments from the popular Facebook page and celebrity IDs for Facebook and Twitter datasets were collected. On the other hand, videos on controversial topics on YouTube had been founded. This type of video is considered a rich source of objectionable and rude

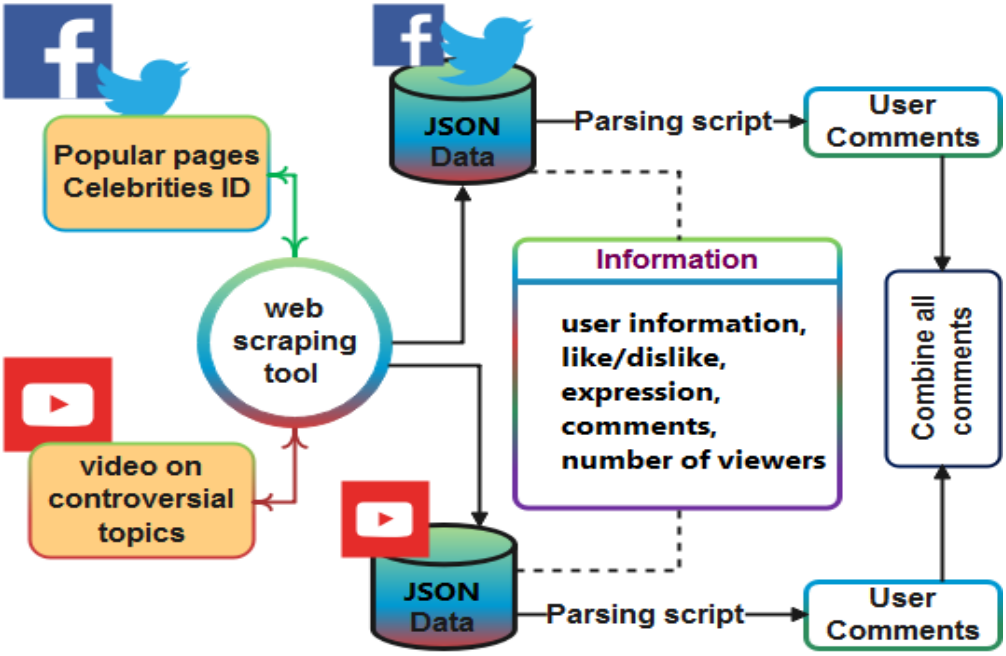


Figure 3.1 Data collection from different social media

comments. There is a big chunk of bullying comments, although social media website allows user to moderate their comments. To take local language into concern, we also choose some district-related pages. We select Noakhali and Chittagong district local languages for this work. Local language comments were collected from their district-related page and IDs.

Raw data collected through the web-scraping tool was stored in the CSV excel sheets. This raw data contains user information, likes/dislikes, expressions, comments, and the number of viewers. Initially, we gathered 2000 Facebook comments, 1000 Twitter comments, and 1300 YouTube comments. Finally, we combined the collected comments from the different social media platforms into a single corpus. Table 3.1 represents a scenario of our collected data.

The collected comments were in Bengali, English, transliterated Bengali, hindi, urdhu and code-mixing Bengali languages. Several comments only contained emojis or images. We discard unnecessary comments in the following data cleaning step as we only need text-based comments. The final courpus only contained Bengali, transliterated-Bengali and code-mixed Bengali

Table 3.1: Sample of collected data

No.	User Comment	Source	Like	Dislike
1	pAgla chuda mohila	Youtube	3	0
2	Khanki magi.. Bokachoda ranu mandal..	Youtube	1	1
3	Osm broo apni ake bare thik ...erokom selibrity nejeke ki je Mone korche .se abar orkome bike korbe	Youtube	3	2
	Dada 1 Million Hoye Gelo Congratuletion...	Youtube	5	0
4	Driverless AI NLP blog post on https://www.h2o.ai/blog/detecting-sarcasm-is-difficult-but-ai-may-have-an-answer/	Facebook	0	0
5	Banchod ayle ata kora tik na tor	Facebook	2	0
6	😆😆😆😆😆😆😂	Facebook	0	0
7	Mela bala laigche viu	Facebok	0	1
8	Thik koreche koyel . Ranu mukhe sobai mile juto Mara darkar	Youtube	1	1
9	উপস্থাপিকার ড্রেসআপ দেখে আমি মুগ্ধ 😂😂	Facebook	1	1
10	Funny guy😂😂😂😂	Instagram	0	1
11	হাইসতে হাইসতে খাদের তন হরি গেছি	Instagram	0	0
12	Khankimagi akta..salar Fasi chai	Facebook	0	1
13	hudira asole bipodjnok ader k mere fela uchit	Facebook	3	5
14	যত সহজে অন্যের বৌ কে নিজের বৌ বানায় ফেলছো !	Facebook	2	0
15	মিলু ভাইয়ের জন্য আমার খুব কষ্ট লাগে!	Youtube	3	1
16	যে যাই বলুক, তুমিই দেশ সেরা ফিল্ডার love u nasir	Facebook	1	1
17	ও আমার বাল বুঝে ….	Facebook	0	0
18	বিশ্রী সাম্প্রদায়িক জঙ্গি মানসিকতা এই মুসলমানদের	Facebook	0	2
…. …. ….	…. …. ….	…. …. ….	…. …. ….	…. …. ….
1852	Seriously mane just awesome 🤣🤣🤣🤣🤣	Youtube	0	0
1853	Haste haste pete khil dhore niyeche baba re🤣🤣🤣🤣🤣🤣🤣😂😂😁😁😁😁😁	Facebook	1	2

3.1.2 Data Processing

Data processing section includes data filtering , data annotaion, and data preprocessing. Steps involved in data processing is described below.

a) Data filtering : The user writes comments in Bengali, English, transliterated Bengali, or code-switching word. The target was to create a corpus containing transliterated Bengali, Bengali, and English text. So, we had to delete redundant data from the raw data and remove those data. Figure3.5 gives a view of the data filtering process.

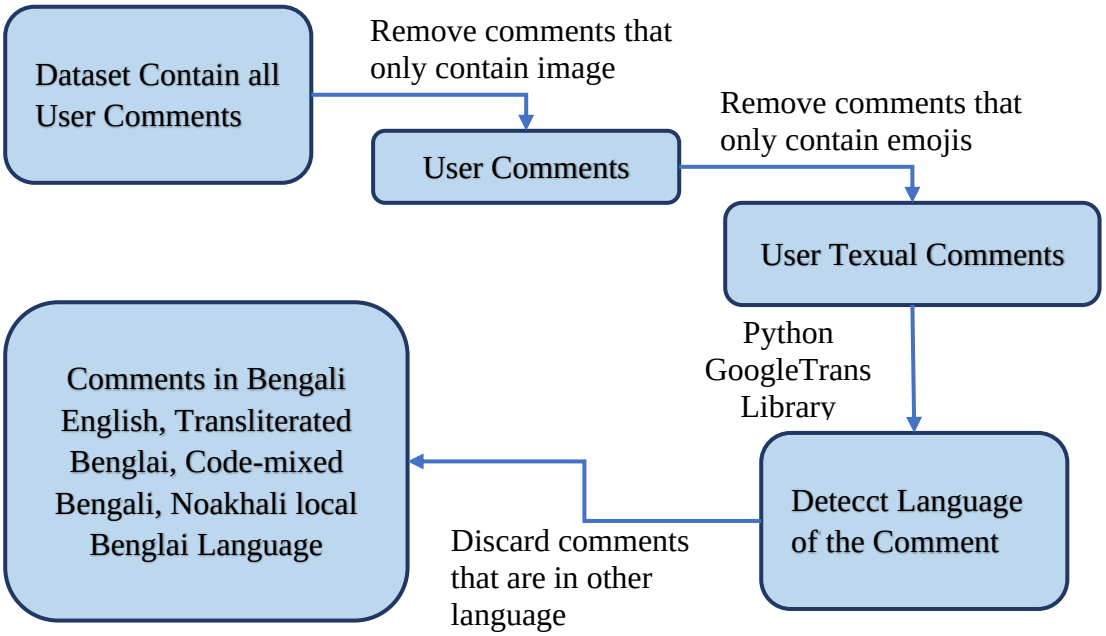


Figure 3.2 Data filtering process

Comments that only contain emojis or image was discarded from the dataset. After that, the language of the comments was detected. To detect language, we take the help of the Python Google trans library. Local language detection was also done manually as, in some cases, the GoogleTrans library could not detect Noakhali and Chittagong's local languag.

Only Bengali, English, transliterated, and code-mixed Bengali comments were taken. At last, we had 1856 comments in the dataset. Further, these comments were annotated. Among this data, bullying, neutral, and nonbullying comment was 816, 772, and 268. Also, these 1856 comments, 353, 318, 309, 279,245,148,112,92 comment was labeled as Aggressive, sexually abusive, supportive, normal_comment, sad, religiously abusive, happy, and meaningless.

b) Data Annotation: Data annotation indicates examining each comment and assigning a label. We manually labeled the dataset. We have eight individual classes for the first section to assess the sentiment of the Bengali text. The classes are supportive, aggressive, sexual harassment, religious harassment, normal comment, and meaningless. Those eight classes were again used with the baseline feature to detect three classes bullying, nonbullying and neutral. Figure 3.6 depicts the classes and their subclasses.

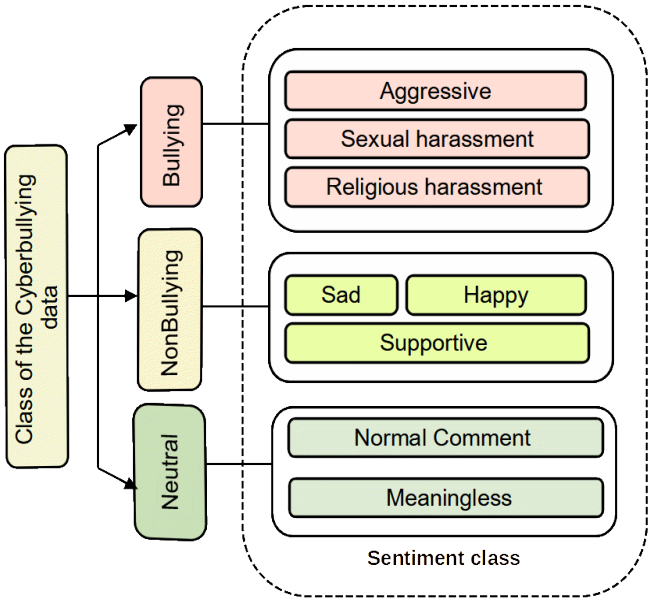


Figure 3.3 Sentiment class and Bullying class

The three classes that were assigned for comments:

i. Bullying: This class contains comments with threats and abusive words. For instance: "tui Ekta Bustard, tor life survive korbar right nai" this sentence contains a word of threats. The sentiment classes abusive, sexual harassment, and religious harassment belong to this class. Here, a comment that indicates contempt, cyberstalking, abuse, threat, rage, harassment, and disgust belongs to the abusive class.

ii. Nonbullying: The comments that did not contain abusive words or harassing content was placed in this class. For example, "apnake onek nice lagche," is no threatening word. The sentiment classes sad, happy, and supportive belong to this class. A comment is considered supportive if it shows love, excitement, advice, or positive talk about the user's post.

iii. Neutral: Comments that are not in the above two classes. The sentiment classes, normal and meaningless comment belong to this class.

Transliterated Bengali words do not have a fixed spelling, so the corpus had the same transliterated word with different spellings. We corrected those spelling in the following data preprocessing step.

c) Data Preprocessing: We need to clean or trim the data to obtain higher accuracy. Preprocessing the data required several steps, including data cleaning, stop word removal, and tokenization. The phases involved in data preprocessing are described below:

Step 1 (URLs remove): The preprocessing begins by removing URLs from the comment. Some Twitter comments contain URLs, which can decrease the model's accuracy. So we remove all URLs of comments.
Step 2 (conversion data to lower case): The comments of the dataset were converted to lowercase.
*Step 3 (Remove Emojis and Punctuation*s): The number of emojis and punctuations were counted for each comment and stored in the dataset. After storing the number of emojis and punctuation, we remove emojis and punctuation from the comments.
Step 4 (Spelling correction): We create a spelling checker list. It helps us to handle numerous variations of the same word. Table 3.2 show the example of a spelling checker list.

Table 3.2 Sample of spelling Variation available in the dataset

Word	**Variation of spelling**
Ovinoy	obinoy, obhinoy, ovhinoy, obhinoi, ovioni, ovhinoi
Kore	kre, kra, kora
Dhur	dur, dhor
Pagol	pagl, pagul, pag0l
Khelle	khalla, khella, khealla
nosto	noshto, nsshto, nsto
Vasay	bhasay, basay, vhasay, basai
Abal	abalㅣl, albal, abballl
Bejonma	bajonma, bejamana, benjonma
valo	Bhalo, balo, blo, vhalo
tui	2i, toi, ti
boka	Voka, bka, buka

We did not apply 'stop removing,' 'stemming,' and 'lemmatization' in the dataset. Removing it can change the meaning of the comment. Here we create a list of profane words and swear words. This list contains typical Bengali and local words of the Chattogram and Noakhali

districts. Using this, we count the number of profane words and swear words in the comment. Besides, a list of Bengali positive words list was prepared. The number of positive words in the comment was counted with the help of a positive word list. A negative word list has also been established. It was used to keep track of how many negative words there were.

Finally, we apply VADER rule-based model to analyze the sentiment of the comment. It provides negative, neutral, positive, and compound scores. If the compound score is more significant than zero, the comment is positive; if it is equal to zero, the comment is neutral otherwise negative. We fetched this sentiment class and stored it in the dataset.

Data preprocessing helps us reduce redundant data and increase the model's accuracy. After completing data preprocessing next step was to select features and make those features readable to the machine learning model. So, the feature extraction technique was applied in the next stage.

3.2 FEATURE EXTRACTION

Feature extraction allows us to convert the selected features to the numerical value. This numerical value of the data is then fed to the machine learning model. Here we had two steps. The first is feature selection and then applying the feature extraction technique.

3.2.1 Features Selection

Features are used to train models. The accuracy of the detection of a model depends on feature selection. V. Balakrishnan et al. demonstrate that cyberbullying detection improved when personalities and sentiments were used [38]. A similar effect was not observed for emotion. They get 91.88% accuracy when combining sentiment and personality features with baseline. Indico API and IBM Watson's Personality Insight API were used for sentiments and personality analysis, respectively. Text/content, user, and network features were the baseline features obtained from the Twitter dataset. After analyzing the previous work in this field, we came up with the following features:

i. Number of the positive words: Total number of positive words used in each comment.
ii. The number of Negative words: Number of negative words in each comment.

iii. The number of the profane words: We established a list of profane words used in transliterated Bengali comments. This list contains profane and swearing words used in Noakhali and Chattogram local language.

iv. Number of Emojis: Number of emojis used in each comment. Count emojis with the help of the python emojis library.

v. The number of punctuations: Punctuation used in the comment is also counted. Sometimes, users use punctuation to express their expressions.

vi. The number of Likes and dislikes: The number of likes and dislikes was also considered for each comment.

vii. Related to post: We also consider whether the user comment is related to the post or not. If the comment was related to the post, the assigned value was one; otherwise, 0.

viii. Sentiment: We take the sentiment score gotten by the VADER library. The sentiment of the Bengali text got through the machine learning model was also used in the second phase of Bullying detection.

Here we got baseline features from the collected dataset from different social media. Baseline features are the number of likes, dislikes, emojis, punctuations, positive words, and profane words. We selected sentiment features with baseline features to train our machine learning models. The next step was to apply the appropriate feature extraction technique to convert text data to the numerical value.

3.2.2 Feature Extraction Technique

The raw text had to convert into a vector space. We use the feature extraction technique by G. Rounak et al. [39]. They use Term Frequency Inverse Document Frequency (TF-IDF) and Bag of n-grams (BONF) for feature extraction. The TF–IFD value increases proportionally to the number of times a word appears in the document and decreases with the number of documents in the corpus that contain the word [40]. A bag-of-n-grams model represents a text document as an unordered collection of n-grams (bigram if n=2).

For categorical features' Dummy variable' technique is used. There is only one feature, Bengali sentiment, which is the categorical feature. Other features are numerical; hence no necessity to encode.

3.3 MODELS FOR CLASSIFICATION

The research used k-nearest neighbor (KNN), Naïve Bayes, Support Vector Machine, Random forest (RF), and Logistic Regression classification algorithm to detect cyberbullying. Naïve Bayes, Support Vector Machine, and Random forest (RF) was used to detect the sentiment of the transliterated Bangla text. Here, defines the logic for using those algorithms.

a) **K-nearest neighbor (KNN):** Supervised learning algorithm KNN is an instance-based classification algorithm. This classifier algorithm predicts the target label by locating the nearest neighbor class. The closest class to the to-be-classified point is determined using Euclidean distance or cosine similarity [41]. The KNN algorithm is one of the simplest classification algorithms, but it can produce competitive results [42]. The procedure to predict labels by the KNN model is illustrated in Figure 3.4.

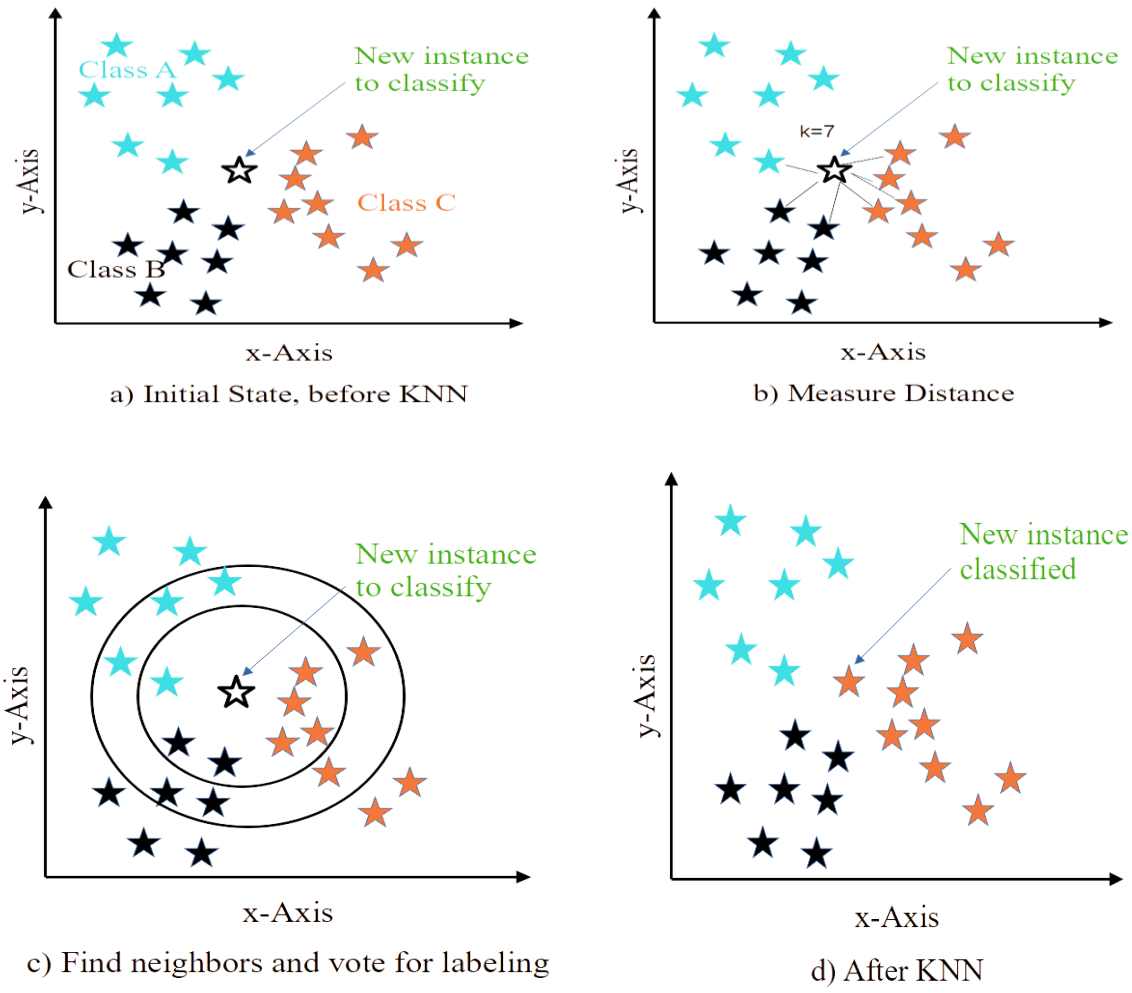


Figure 3.4 The KNN algorithm procedure to decide neighbours.

For instance, there are three colored stars. The KNN model first finds the distance for k's number of neighbors. If k's value is seven, it will measure seven neared star neighbor distance. After that, vote for labeling. In Figure 3.4, the new star instance gets the vote for black star class two, blue star class one, and orange star class four. So, the new start belongs to the orange class.

The KNN works well in terms of accuracy and efficiency for big data. KNN was used to build cyberbullying prediction models [43]. The author uses the Twitter dataset and gets the highest accuracy, 61.06, and the lowest, 59.79, for K's values 2 and 10, respectively.

b) **Naïve Bayes**: Naïve Bayes is a probabilistic algorithm that works on the Bayes theorem with strong-independent assumptions [44]. Figure 3.5 defines the procedure of the Naïve Bayes classifier. For text classification, researchers use Bayesian learning frequently. This model assumes a parametric model generates text and uses training data to calculate Bayes-optimal estimates of the model parameters.

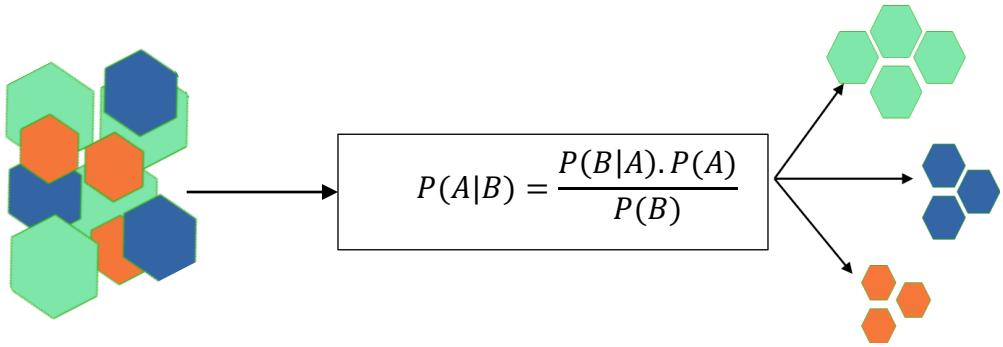
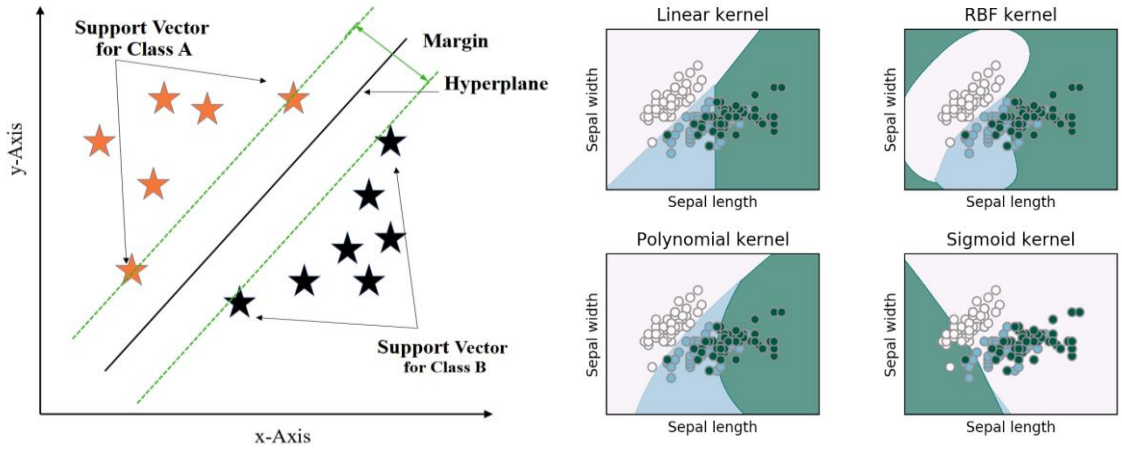


Figure 3.5 The procedure of Naïve Bayes classifier

H Zhang describes the working procedure method of Naïve Bayes in detail [45]. Authors get 33.91% accuracy in detecting cyberbullying using the Naïve Bayes algorithm [43]. To detect text-based emotion on the Twitter dataset, use Naïve Bayes in their model and get an accuracy of 83% [46].

c) **Support Vector Machine (SVM):** SVM is a statistic-based algorithm that creates a line or a hyperplane to separate the data into classes [47]. Four types of SVM based on the kernel type, namely Linear, polynomial, RBF, and sigmoid, are used in this study. The procedure of these four types of SVM is depicted in Figure 3.6.

For instance, there are two types of stars: a black and an orange star. The SVM first creates an optimal decision boundary called hyperplane to classify those two types of stars. After that, find the support vectors. Support vectors are the closest point of each



a) The basic procedure of the SVM

b) Four types of SVM
source: dulnvxiers.tk

Figure 3.6 The SVM classifier.

class from the hyperplane. The distance between support vectors hyperplane is called margin. The SVM classifier always finds the hyperplane with the maximum margin. There is less chance of future misclassification for maximum margin. Sometimes narrow margins lead to misclassification for future new data.

The researchers discovered that the SVM-based cyberbullying model effectively detected cyberbullying. Mangaonkar et al. constructed an SVM-based cyberbullying detection model for YouTube [34]. Authors, to detect Indonesian twitter cyberbullying using text classification using KNN and SVM [48]. They state that SVM gains a higher F1 score of 67% than the KNN algorithm. Amgad Muneer et al. compare different ML algorithms and find that SVM provides better recall (1.00) than the others [49].

d) **Random forest (RF):** RF consists of many decision trees randomly selecting feature variables for the classifier input [50]. An example of an RF classifier is shown in Figure 3.7. For instance, there is a dataset with distinct types of shapes. When this dataset is given to the RF model, the dataset is divided into subsets. Each subset is then given to separate decision trees. Those decision trees produce prediction results based on the

training data. For a new instance, the majority vote was taken into consideration to label this new instance. In Figure 3.7, the new instance gets a majority vote for class A. Hence it belongs to class A.

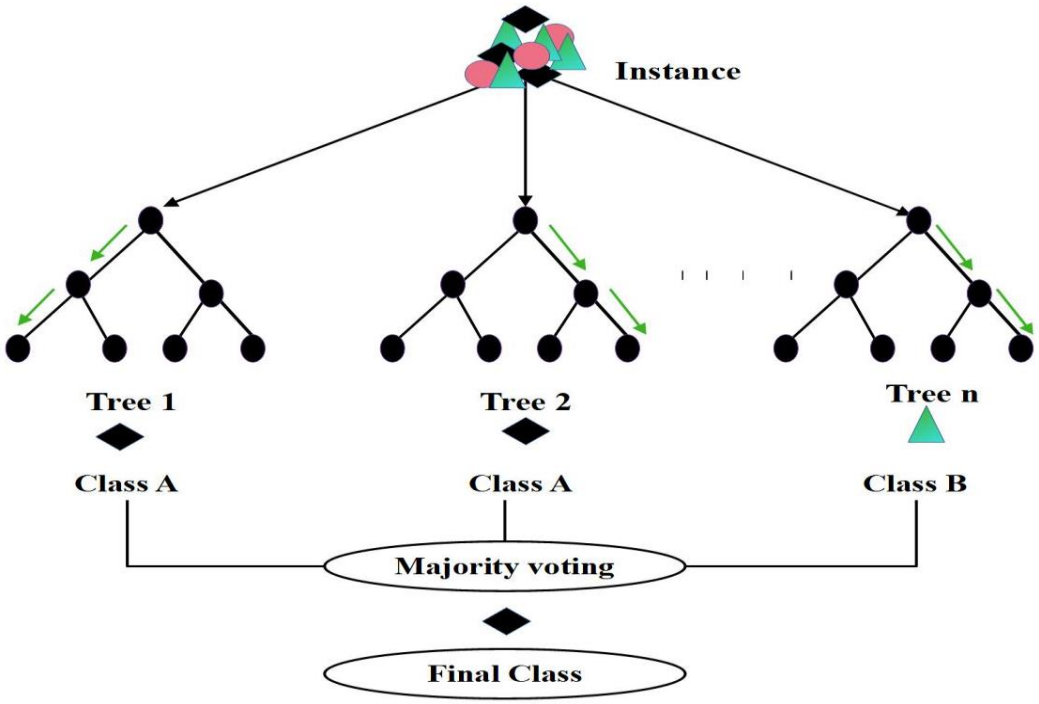


Figure 3.7 The Random Forest classifer.

The author used the RF algorithm to detect abusive comments in Bangla-English code-mixed and transliterated text and got 72.14% accuracy [9]. C.Van Hee et al. also use RF to construct of cyberbullying prediction model [51]. Another research work done by the authors finds that RF has achieved an accuracy of 89.84% and a precision of 0.9338 to detect bullying [47].

e) **AdaBoost:** AdaBoost or Adaptive Boosting is one of the ensemble boosting classifiers. It combines multiple classifiers to increase the accuracy of classifiers. It is an iterative approach [52]. AdaBoost classifier builds a robust classifier by combining multiple poorly performing classifiers so that we will get higher accuracy. Each training observation is initially assigned equal weight. The definitive boundaries obtained during several iterations are combined using several lows. Thus, the accuracy of the overall iteration is enhanced. The accuracy of 89.30% and recall of 0.8756 is obtained by using

AdaBoost [49]. They find that the accuracy value is greater than the SVM. The authors get 69.03% accuracy for AdaBoost [9].

f) **Logistic Regression (LR):** Logistic regression uses the sigmoid function. This LR model can be used to classify Bullying and Sentiment. The author gets the highest accuracy of 90.57%, with less prediction time of 0.0015, by using LF to detect bullying [49]. Figure 3.8 illustrate the function of the Logistic Regression model.

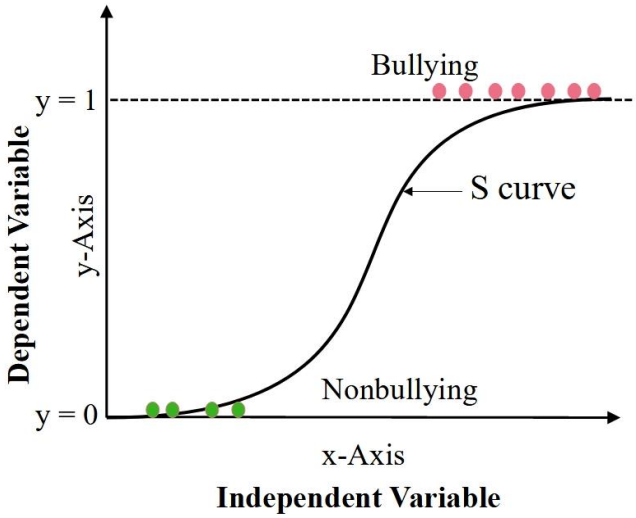


Figure 3.8 The Logistic Regression classifier.

The logistic regression algorithm takes features (inputs) and generates a forecast based on the likelihood of a class matching the input. For instance, if the likelihood is ≥0.5, the instance classification will be a positive class; otherwise, the prediction will be in the negative class [53].

The logistic regression works depending on the sigmoid function with an S-shape curve. This sigmoid function maps each output value between 0 and 1. The function is:

$$P(b) = \frac{1}{1 + e^{(-b)}} \tag{3.1}$$

In Figure 3.8, the comments are classified depending on the threshold value. The comment with a prediction value greater than the threshold value is labeled as "1", called the Bullying class. The NonBullying comment class is labeled as "0" as they have a prediction value less than the threshold value. Figure 3.8 show an example of binary classification, whereas in

this study, the classification in multiclass classification problem. Because of this multinomial LR model was implemented to classify sentiment and detect bullying.

The algorithm that provides the highest accuracy and precision to detect the sentiment of the transliterated Bengali text was selected for the next phase. Then will merge the output of the best sentiment analysis algorithm with the baseline feature to detect cyberbullying. Here, analyze performance based on metrics to find the best-performed algorithm.

3.4 PERFORMANCE ANALYSIS

This study utilizes six ML algorithms: KNN, Naïve Bayes, SVM, LR, RF, and AdaBoost. It is essential to review standard metrics to understand the performance of the conflicted models. This research will compare performance by using various concise metrics. It will determine how well the model can distinguish between cyberbullying and non-cyber bullying content and classify sentiment.

The commonly used matrices for performance analysis of ML algorithms are precision and recall, accuracy, F-measure and AUC-ROC curve. Here briefly describe the metrics that will take into consideration to analyze performance.

a) Precision: Precision value is the proportion of correct positive classifications from cases that predict as positive. It is the accuracy of the optimistic prediction.

$$Precision = \frac{True\ positive\ (TP)}{True\ Positive + False\ positive\ (FP)} \tag{3.2}$$

For cyberbullying detection, TP indicates the number of accurately detected cyberbullying comments in bully class. The number of cyberbullying comments that are not bullying but selected as bullying is mentioned by FP. For cyberbullying classification, precision is as follows:

$$Precision = \frac{Total\ number\ of\ accurately\ predicted\ actual\ bullying\ comments}{Total\ number\ of\ predicted\ bullying\ comments} \tag{3.3}$$

The higher precison value indicate better performance. If precison value 70% for the Bullying class that means overall 30% comment was wrongly predicted as Bullying.

b) Recall: Recall is the proportion of correct positive classification (True positive) from positive cases.

$$recall = \frac{True\ positive\ (TP)}{True\ Positive(TP) + False\ Negative(FN)} \tag{3.4}$$

Here FN is the number of comments in bully class, but the algorithm predicted them as non-bullying. For cyberbullying, and sentiment classification recall is as follows:

$$recall = \frac{\begin{matrix} Total\ number\ of\ accurately\ predicted \\ actual\ bullying\ comments \end{matrix}}{Total\ number\ of\ actually\ bullying\ comments} \tag{3.5}$$

$$recall = \frac{\begin{matrix} Total\ number\ of\ accurately\ predicted \\ actual\ 'Class\ A'sentiment\ comments \end{matrix}}{Total\ number\ of\ actual\ \ 'ClassA'comments} \tag{3.6}$$

c) F-measure: F-measure combines both precision and recall. It also known as F1 score.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{3.7}$$

d) Accuracy: The ratio of actually detected cases to overall cases is accuracy.

$$accuracy = \frac{True\ positive + True\ Negative\ (TN)}{Total\ number\ of\ \ comments} \tag{3.8}$$

TN define the number of comments that are not bullying comments and detected in non-bullying class.

$$accuracy = \frac{\begin{matrix} The\ number\ of\ \ comments \\ that\ are\ predecte\ accurately \end{matrix}}{Total\ number\ of\ \ comments} \tag{3.9}$$

Comparison between models is made based on performance metrics. A model with high accuracy and precision is preferable. If accuracy is low, but precision is high, then there can be a systematic error. The unacceptable model is with low precision and accuracy. On the other side, if the precision value is high for one model and recalls for another, then f1-score has to be used to determine the best-performing model.

3.5 FLOWCHART

Today it is also concerning to prevent bullying as it can already create damage before it deletes or erases social media. So, this study aims to evaluate a prevention model based on the best-performed algorithm. Prevention techniques will be helpful for both the user and social media website. Figure 3.9 shows the detail of the methodology to identify and prevent transliterated cyberbullying.

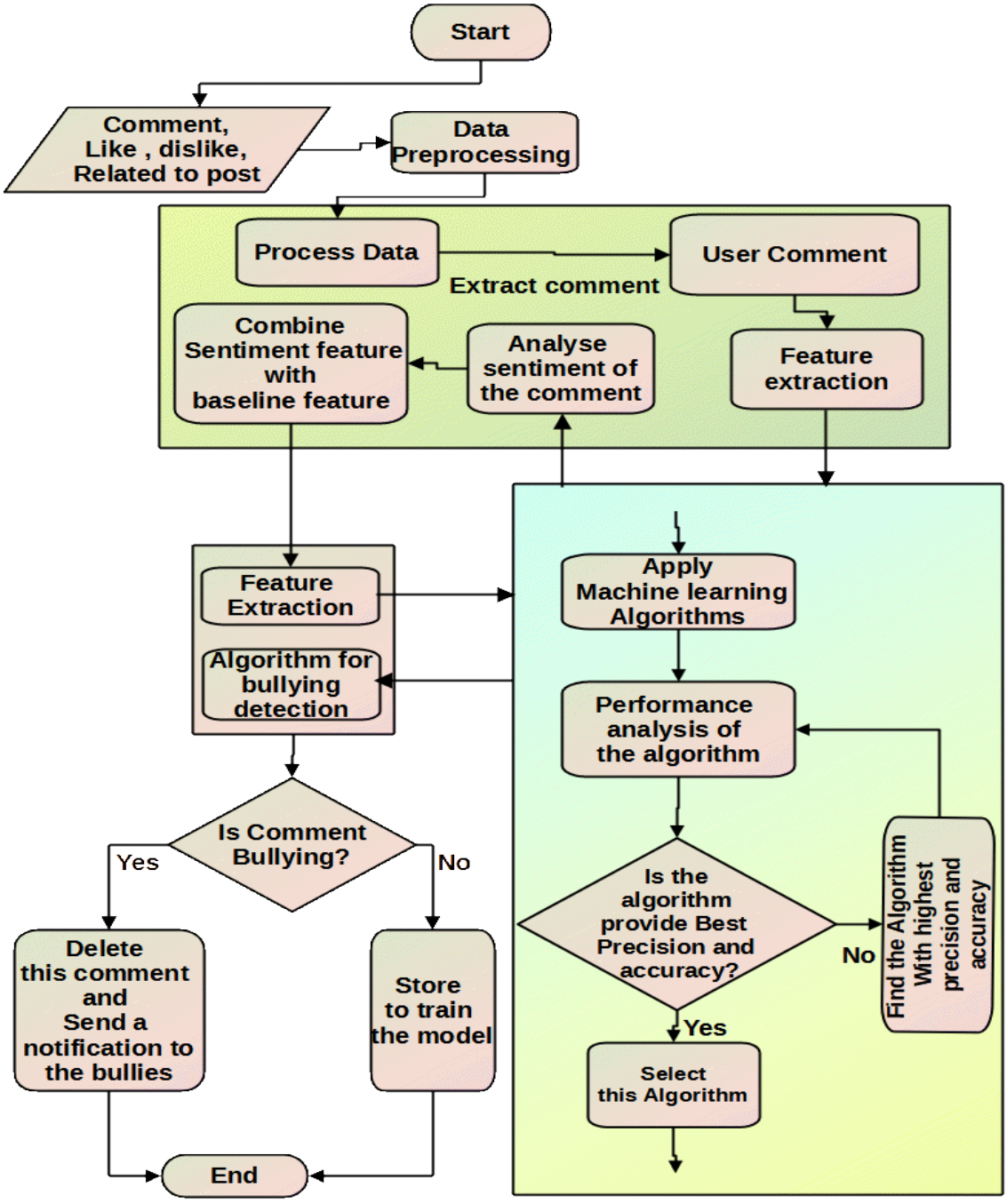


Figure 3.9 Flowchart of the cyberbullying detection and prevention model.

The workflow involves five steps: dataset creation, data preprocessing, feature extraction, classification algorithm application, and model selection. Figure 3.9 show that after getting the user comment, remove URLs from the comment and convert the comment to lowercase. Each comment's number of emojis, punctuations, and profane, positive, and negative words is counted. After that, a dataset includes user comments, likes and dislikes, emojis, punctuations, profane words, and positive and negative words. This created dataset was used to detect the sentiment of the text.

Here, TF-IDF and n-gram feature extraction techniques make text data usable. The selected feature is taken as independent data to detect the dependent sentiment class. 70% of the data was used to train the model, while 25% was for testing.

After that, the Python sentiment analysis model VADER was used to detect sentiment. As it only provides three classes of sentiment, ML models have to apply to detect the sentiment of transliterated Bengali text. The best-performed algorithm is selected and used for the next phase. The final model to detect cyberbullying has been trained with baseline and sentiment features.

Different machine learning algorithms to process data have been used. This work uses KNN, NB, SVM , RF, LR, and AdaBoost to detect cyberbullying. As the concern is classification, the focus must be on the algorithms' precision value and accuracy. An algorithm with the highest precision value indicates how correctly detect original bullying. We select the algorithm with the highest precision and accuracy for the final cyberbullying detection and prevention model.

3.6 SUMMARY

It is possible to increase the accuracy of detecting transliterated Bengali text-based bullying. The model's accuracy depends on the data collection process and feature selection to train the model. In this study, the sentiment of the text is used with baseline features to increase accuracy.

The following two chapters analyze the model implementation and sentiment and bullying detection results. Chapter four, 'Model for Sentiment Detection,' depicts the model's performance in detecting the sentiment of the Bengali text. The sentiment result has been used to detect bullying comments. The model's performance in detecting Bengali bullying comments is explained in chapter 5.

CHAPTER 4

MODEL FOR SENTIMENT DETECTION

In this chapter the performance of the ML model in detecting sentiment of the Bangla text is analyzed. This chapter mainly has two sub-sections: performance analysis of each model and comparative analysis between the models.

4.1 PEERFORMANCE ANALYSIS OF THE MODEL

The first dataset that only includes user comments was used for sentiment analysis. After preprocessing the dataset, the Python GoogleTrans module to convert those comments into Bengali and English was used. The following section describes the outcome gained through the package and model used for sentiment detection.

4.1.1 VADER

The Python sentiment analysis module, VADER, was applied to those converted texts. VADER is specifically attuned to sentiments expressed on social media so that it can understand slang words. The sentiment detected through the VADER was fetched and stored on the database. Table 4.1 shows the sample dataset after sentiment analysis using VADER.

The comments were classified into positive, negative, and neutral. VADER uses a dictionary to analyze sentiment [54]. This dictionary maps lexical features to sentiment scores. This sentiment score indicates how positive or negative the sentence is.

The sentence is negative if the sentiment score is less than zero. A sentence is positive if its sentiment score is greater than zero. If the sentiment score is zero, the sentence is classified as neutral. The sentiment class obtained by the VADER module is not enough to understand the exact emotion of the user. So, set out to find a machine learning algorithm capable of classifying sentiment into more than three categories.

Dataset with transliterated Bengali text was labeled eight individual classes for each comment. The number of positive and negative words in the text to analyze sentiment were also considered. The dataset was converted into two segments, 75% data for training and 25% for testing. The following sub-section describes the performance of the ML models for sentiment detection.

4.1.2 KNN algorithm

For the KNN algorithm, K=1 to K=6 was tried to find the best precision value. The accuracy of the model changes with K's value. The performance of KNN model for testing and training data is depicts below.

KNN Model Performance for Testing Data

The highest accuracy of 49.138% for K=1 was achieved for testing data. The detailed of KNNs evaluation are shown in Table 4.2.

Table 4.2 Accuracy of KNN model for distinct k values to detect sentiment.

Value of K	1	2	3	4	5	6
Accuracy	49.14%	41.37%	32.81%	32.81%	32.10%	32.11%

For k=2, the KNN model provides 41.367% accuracy. After that, the accuracy of k's values 3 to 6 was 32%. The value remains the same, so the value of k's was not increased. However, it is not enough to choose the best model for sentiment analysis depending on accuracy. Hence, other performance matrices were also analyzed. Figure 4.1 depicts the confusion matrix of the KNN model for testing data.

Here, the number of supportive comments that were predicted accurately was 81. There was a conflict between aggressive and sexually_abusive comments. Fifty-six aggressive comments were detected as sexually abusive by the KNN model. Probably the reason behind this was the similarity between the tone of both types of comments.

On the other side, the KNN model mispredicted most of the meaningless comments as sexually_abusive comments. 39 religiously abusive comment was also detected as

sexuallya_abusive comment. However, 37 and 12 comments of sad and happy sentiments, respectively, were predicted accurately.

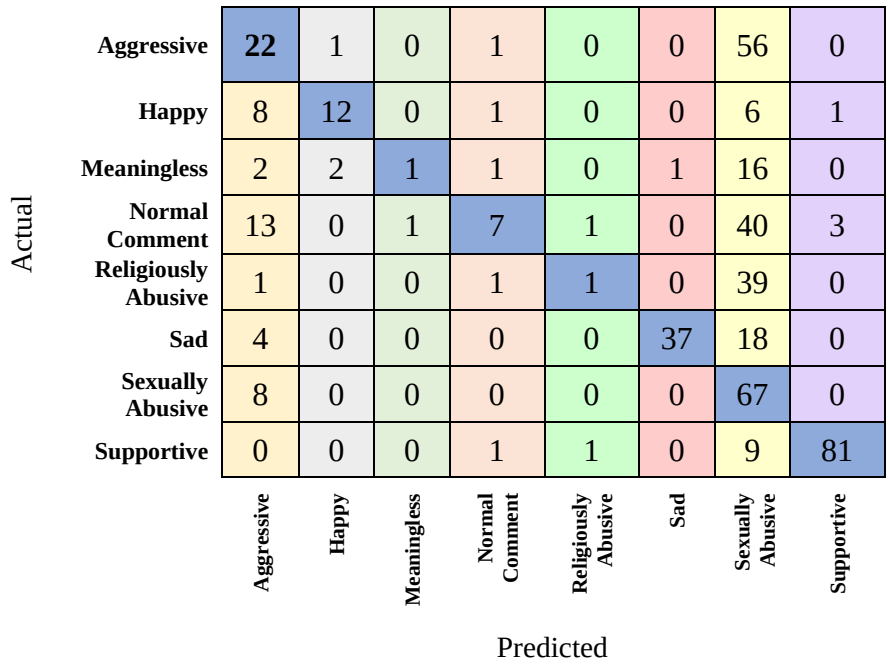


Figure 4.1 Confusion matrix of KNN model to detect sentiment of the test data.

The precision, recall, and f1 score of the KNN model to detect the sentiment of the comment is shown in Figure 4.2.

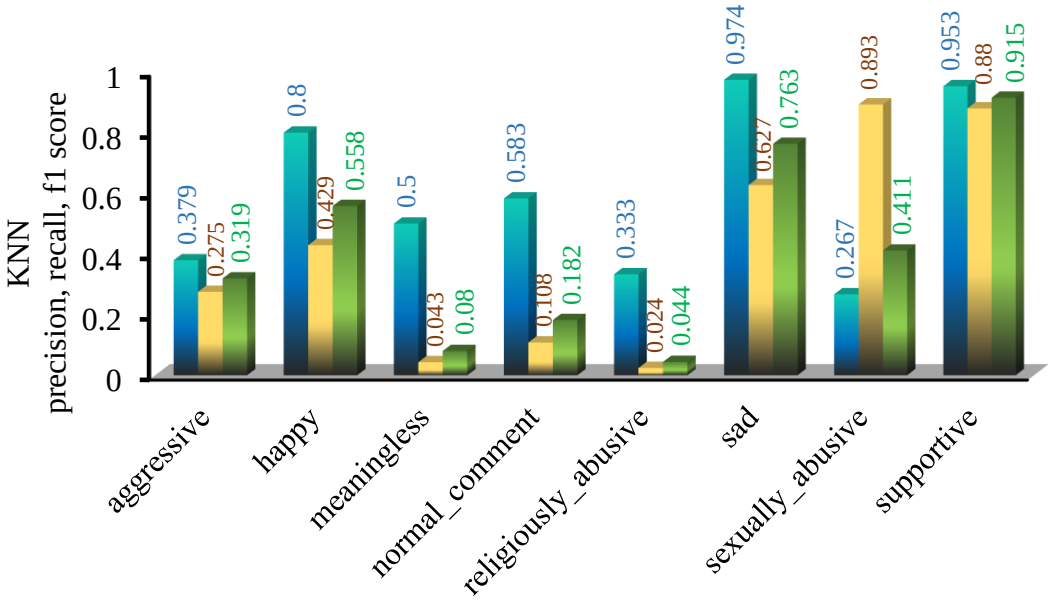


Figure 4.2 Precision, Recall and f1 score of KNN model for k=1

The highest precision value of 0.974 was achieved for the sad sentiment. 0.88 and 0.915 were the highest recall and f1 score values for supportive comments. However, the KNN model could not detect most meaningless, normal, and religiously abusive comments.

Here, it was observable that most of the meaningless comment was mispredicted. For instance, 'djhfkaureo hdo ie j h' is a meaningless comment detected as sexually abusive. A religiously abusive comment like 'শালা মূর্তি পূজারী শিবের ধনেরর পূজারি' was also detected as a sexually abusive comment. However, among 464 test comments, 228 comments were accurately predicted by the KNN model.

KNN Model Performance for Training Data

KNN model with K's value one was examined for training data. For training data, the KNN model achieved an accuracy of 97.99%. There is a vast difference between testing and training data accuracy. The confusion matrix of the KNN model is illustrated in Figure 4.3.

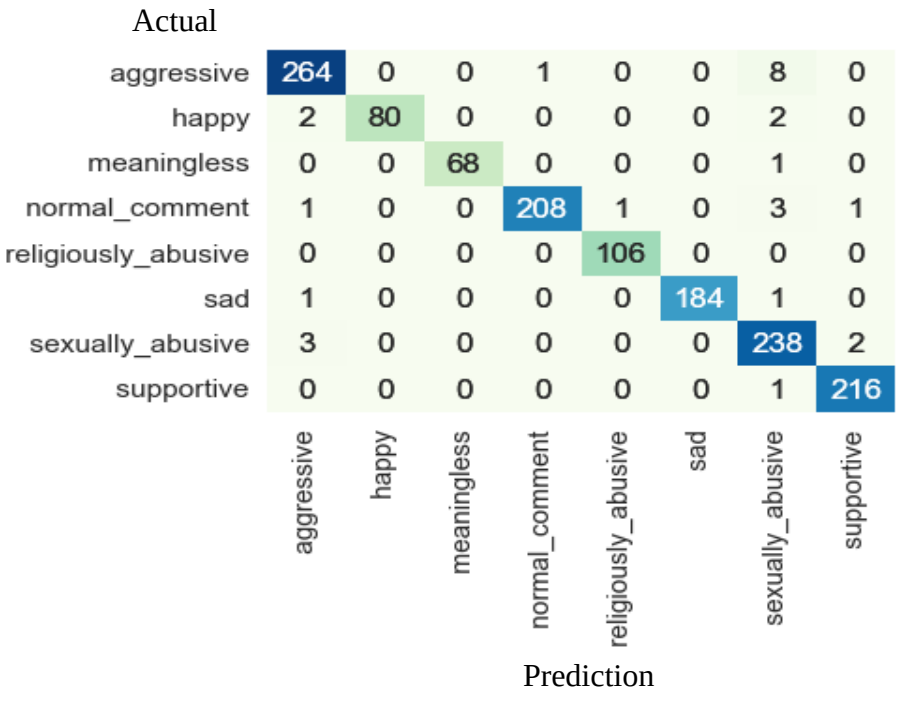


Figure 4.3 Confusion matrix of KNN model to detect sentiment of the training data.

The highest TP value of 264 is for the Aggressive class. The FP and FN value of this Aggressive class are 7 and 9. That means, among 273's aggressive comment, only 9's comment was not classified as aggressive. In contrast, 7's comments were predicted to an aggressive comment that was not aggressive. Similarly, TP, FP, and FN values can be

calculated for other classes. With the help of TP, FP, and FN value, the precision, recall, and f1 score was measured for the KNN model, shown in Table 4.3.

Table 4.3 For training data Precison, Recall and f1-score of the KNN model to detect sentiment of the Bengali text**.**

Sentiment	Precision	Recall	F1-score
Aggressive	0.97	0.97	0.97
Happy	1.00	0.95	0.98
Meaningless	1.00	0.99	0.99
Normal Commen	1.00	0.97	0.98
Religiously Abusive	0.99	1.00	1.00
Sad	1.00	0.99	0.99
Abusive	0.94	0.98	0.96
Supportive	0.99	1.00	0.99

The highest precision value of 1 was for the Happy, Meaningless, Normal, and Sad class. Its state, the accurate prediction for those classes was 100%. From the confusion matrix, only one comment of the 'normal_comment' class was mispredicted. The 'Religiously Abusive' and 'Supportive classes had the highest precision value. Hence the wrong classification of these two classes was 0%. Also, the 'Religiously Abusive' class had the highest f-1 score.

The training and testing data show that the KNN model accurately predicted 1592 comments in 1856. There was 464's comment in the training data, and 228's comment was predicted accurately. 1364's comment of the test data among 1392's was predicted accurately. So, the average accuracy rate of the KNN model for detecting the sentiment of Bengali text is 85.78%.

4.1.3 Naïve Bayes Algorithm

Multinomial NB was used as it is suitable for text classification where data are encoded with the TF-IDF technique. The default value 1 of parameter alpha was taken, which indicates Laplace smoothing.

NB Model Performance for Testing Data

For test data, 53.448% accuracy was obtained by implementing multinomial NB. The Multinomial NB confusion matrix, with alpha = 1, is shown in Figure 4.4. Out of 464

comments, NB correctly predicts 245 comments. The highest 72 comments were appropriately anticipated; this group was supportive. There was a categorization contradiction among aggressive and sexually abusive comments, similar to the KNN model.

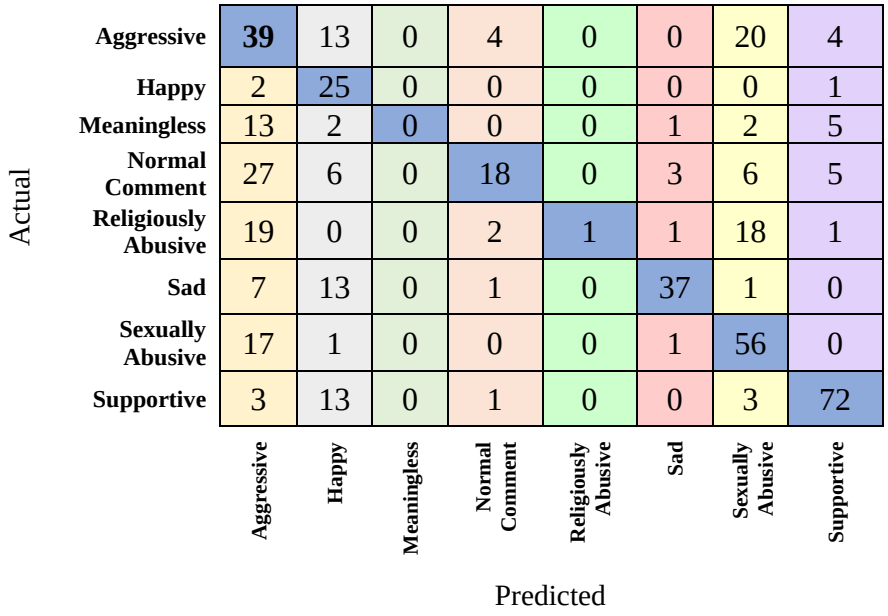


Figure 4.4 Confusion matrix of NB model to detect sentiment of the test data.

Twenty aggressive comments were detected as sexually_abusive comments. 39 aggressive comment was predicted as aggressive. Nevertheless, 19 and 27 religiously_abusive and normal comments were classified as aggressive. The highest type-1 error value of 88 was for the aggressive class. By utilizing the TP, FN, and FP value, the precision, recall, and f1-score were evaluated. The performance of the Multinomial NB is presented in Table 4.4.

Table 4.4 Performance matrices of Naïve Bayes algorithm

Sentiment	**Precision**	**Recall**	**F1-score**
Aggressive	0.308	0.488	0.377
Happy	0.343	**0.893**	0.495
Meaningless	0	0	0
Normal Commen	0.692	0.277	0.396
Religiously Abusive	**1**	0.024	0.0465
Sad	0.860	0.627	0.725
Abusive	0.528	0.747	0.619
Supportive	0.818	0.783	**0.8**

Table 4.4 shows that NB provides the highest precision, recall, and f1 scores for the religiously_abusive, happy, and supportive classes, respectively. However, this model was unable to detect meaningless comments. The meaningless class's precision, recall, and f1 score was zero. The comparison of precision values is shown in Figure 4.5.

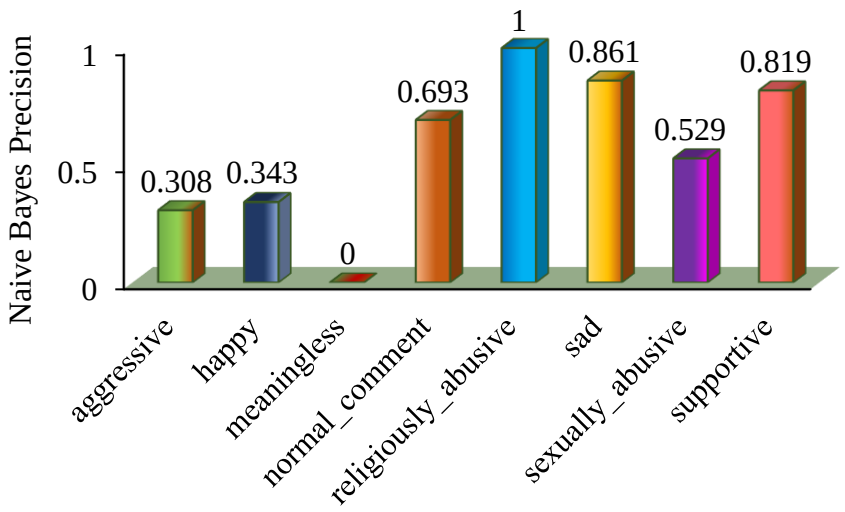


Figure 4.5 Precision value of Naïve Bayes algorithm to detect sentiment.

The highest recall value of 0.893 was achieved for the happy class, but its precision value was 0.307. On another side, 1.00 was the highest precision value for the Religiously_abusive class, which had a 0.02381 recall value. It is observable that NB provides balanced precision, recall, and f1 score for sad and sexually_abusive classes.

The precision value indicates the proportion of correct classification of a specific class from cases that are predicted as this specific class. For instance, consider the 'Happy' class to calculate precision values. The happy class's precision value is the ration TP and the summation of TP and FP for this class. For happy class, the formula of precision is:

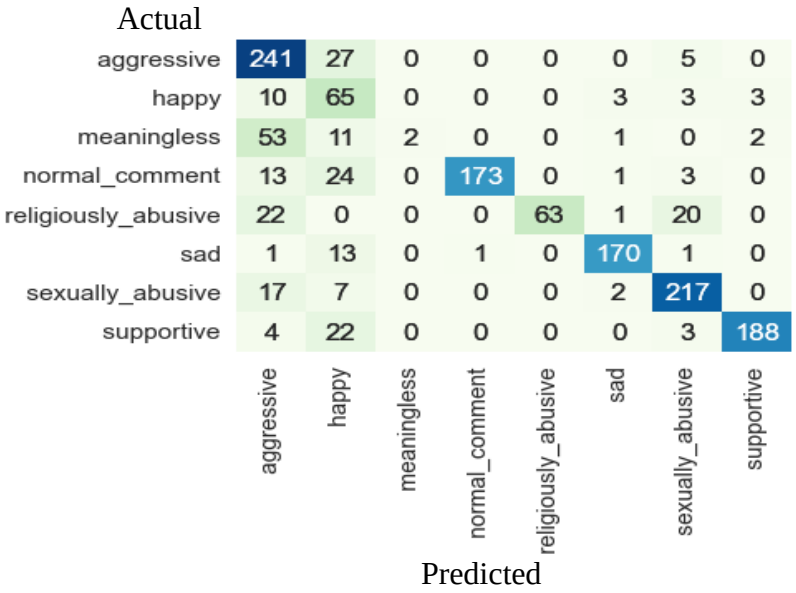
$$Precision_{happy} = \frac{TP_{happy}}{TP_{happy} + FP_{happy}} = \frac{25}{25 + 48} = 0.343 \quad (4.1)$$

Similar to the "Happy" class, the precision value of additional classes can be computed. The precision value for the 'sad' and 'supportive' classes was nearly 80%. Less than 35% of the aggressive and happy classes had precision values. Sexually abusive comments received 52.9% precision, while normal_comments received 69.3%.

For testing data, the NB model provided better performance than the KNN algorithm in terms of accuracy. However, the NB model could not detect comments of the 'meaningless' class. Regarding the precision value, KNN showed better performance detecting all types of comments than the NB model.

NB Model Performance for Training Data

For traning data, the NB model provide 80.388% accuracy. Other performance metirces of NB model is showed in Figure 4.6.



a) Confusion matrix of NB model to detect sentiment of the training data.

Sentiment	Precision	Recall	F1-score
Aggressive	0.67	0.88	0.76
Happy	0.38	0.77	0.51
Meaningless	1	0.03	0.06
Normal Commen	0.99	0.81	0.89
Religiously Abusive	1	0.59	0.75
Sad	0.96	0.91	0.93
Abusive	0.86	0.89	0.88
Supportive	0.97	0.87	0.92

b) Precision, recall and f1-score of NB model

Figure 4.6 Performance metrices of NB model to detect sentiment for training data

The NB model could predict 1119's comments correctly among 1392's. The highest value of Type-1 error is 120, shown for the 'Aggressive' class. Type-1 error is zero percent for three classes. Those three classes are Meaningless, Normal_comment, and Religiously_Abusive. Nevertheless, these three classes showed a higher value of Type 2 error. 67, 41, and 43 is the type-2 error value for the 'Meaningless', 'Normal_comment, and 'Religiously_abusive' class, respectively.

Here, the NB model predicted 1364 comments for the overall dataset correctly. So, the accuracy of this model is 73.49% for the total dataset. This value is less than the previously discussed KNN model.

4.1.4 SVM Algorithm

For the SVM classifier, four kernels were examined. The kernels were RBF, Sigmoid, Polynomial, and Linear. The parameter C was altered to determine the highest accuracy and best F1 scoring. For the polynomial kernel, the degree value was three.

SVM model performance for testing data

The Liner and RBF kernels' performance is higher than the Polynomial and Sigmoid kernels. The SVM's performance for these four kernels with C=5 is presented in Table 4.5. Among the four, the SVM classifier with RBF kernel performed best, and the polynomial kernel performed worse concerning accuracy. 64.224% accuracy was achieved with RBF and Linear kernel.

In comparison, 25.216% accuracy was for SVM with the polynomial kernel where the degree was three. The value of the degree was examined from 3 to 5. The accuracy value of the polynomial kernel for the distinct value of C with degree 3 is shown in Figure 4.7.

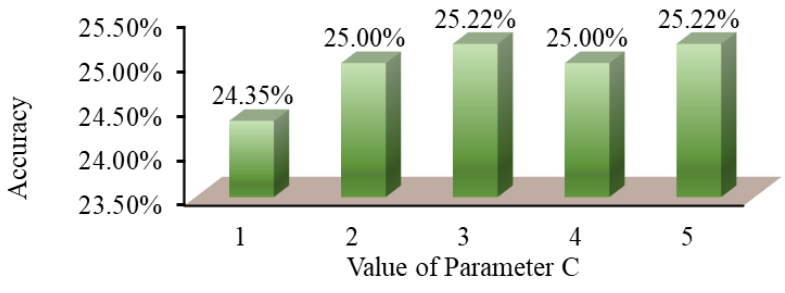


Figure 4.7 The accuracy of SVM classifier with polynomial kernel

Rather than other polynomial kernels, the polynomial kernel with degree three performed better. The polynomial kernel provides 24.784% accuracy for degree values four and five.

SVM with sigmoid kernel obtained 50% accuracy. The highest precision, recall, and f1 scores were for religiously_abusive, sexually_abusive, and supportive classes, respectively. However, this was unable to distinguish meaningless comments. The precision, recall, and f1 score for the SVM classifier with a sigmoid kernel was zero for the meaningless class. Although it was not as effective as RBF and linear kernels, sigmoid kernels outperformed polynomial kernels.

Both RBF and linear kernels produced accuracy above around 64%. It shows that both linear and non-linear functions can fit the dataset. For the meaningless class, the linear kernel provides the highest precision value of 1.00. At the same time, RBF had the highest precision value of 1.00 for the meaningless and religiously_abusive class. The RBF kernel's supportive class's highest recall value was 91.3 percent. Both were tested in the parameter c's 1 to 5 range. The SVM's accuracy for various c values while employing RBF and a linear kernel is shown in Table 4.6.

Table 4.6 Accuracy of SVM for distinct C values and kernels.

C value	Liner Kernel Accuracy	RBF Kernel Accuracy
1	63.147%	32.328%
2	64.224%	53.664%
3	64.224%	59.052%
4	64.009%	63.147%
5	64.009%	64.224%

When C values were 2 and 3, the liner kernel's accuracy value was 64.224%. The RBF kernel, however, got the same precision at C=5. Concerning the RBF kernel, an increase in C's values led to greater accuracy. On the other hand, the accuracy of the linear kernel first increased before beginning to decline. The confusion matrix of linear SVM is shown in Figure 4.6.

Among 464 test comments linear SVM predict 293 comment correctly. If consider the Sad class then TP for this class was 49, FN was 10, FP was 4 and TN was 401. That means that

49 comments of the 'Sad' class were predicted correctly; ten sad comments were not predicted as the sad comment were sad comments.

On the other side, 4 Sad comments were predicted as sad was false, and 401 comments were predicted as accurate but for the other class. Similarly, TP, FP, TN, and FN can be calculated for other classes.

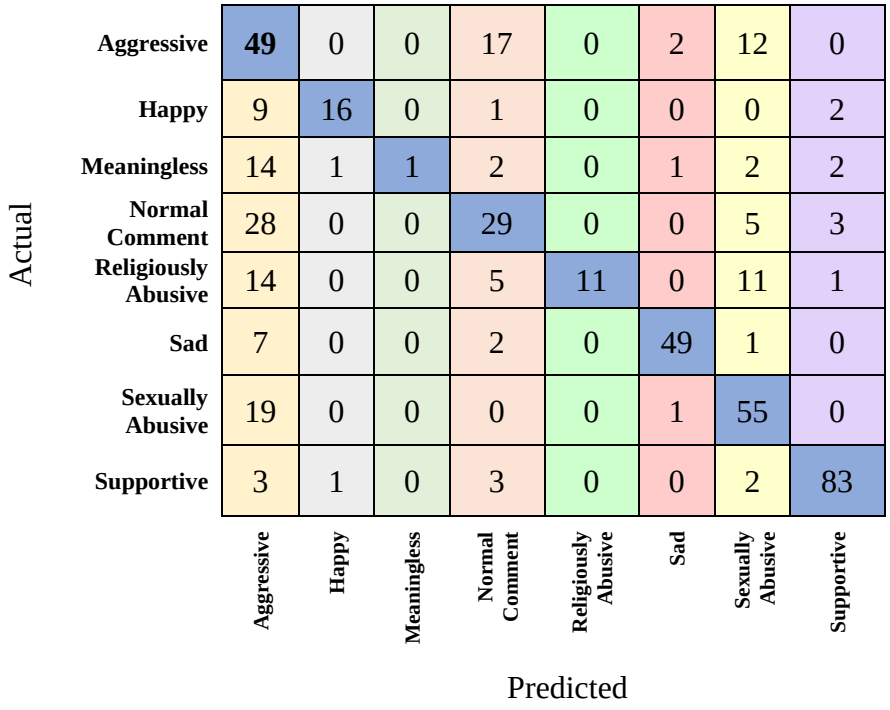


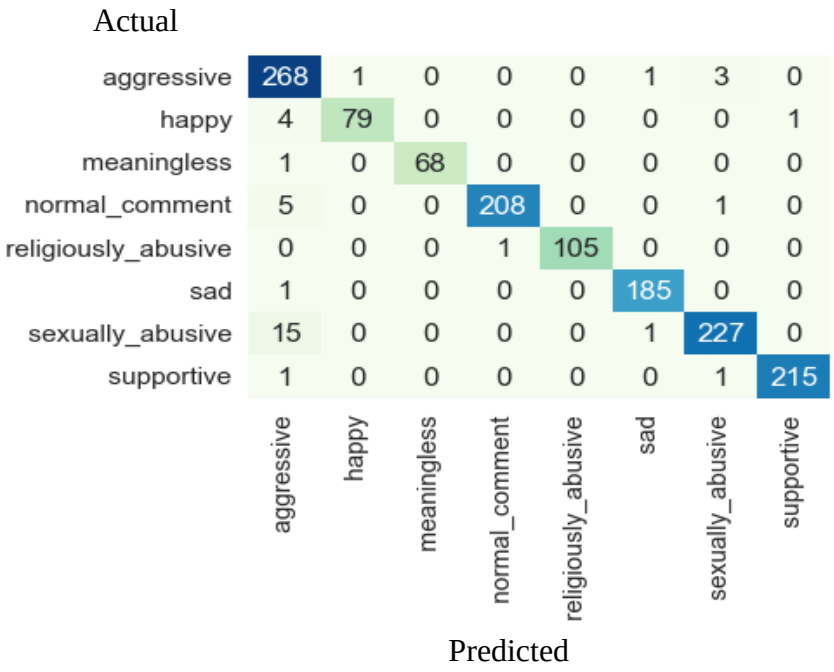
Figure 4.8 Confusion matrix of SVM model to detect sentiment of the test data.

The 'Aggressive' class had the highest FP of 95, and the 'Supportive' class had the maximum TP of 83. The religiously abusive class had the lowest FP, and the meaningless class had the lowest TP. It can summarize that Linear SVM classifier performance can be considered to predict the sentiment of the user comment.

SVM model performance for Training data

SVM model with linear kernel and C's value 2 achieved 97. 34% accuracy for the training dataset. From the confusion matrix shown in Figure 4.9, the linear SVM correctly predicted 1355 comments. There is no mispredicted comment for the 'Meaningless,' 'Normal_comment,' and 'Religiously_abusive' classes. That means the FP value for those three classes is zero. The NB model also had the same. However, the optimism is that, unlike the NB model, the FN value for the SVM model is not high for these three classes.

The precision value for those three classes is one, which defines the prediction accuracy for those classes as 100%. 99% sensitivity was for the 'Meaningless' and 'Religiously_abusive' classes. The 'Normal_comment' class had 97% sensitivity, and the FN value was 6.



a) Confusion matrix

Sentiment	Precision	Recall	F1-score
Aggressive	0.91	0.98	0.94
Happy	0.99	0.94	0.96
Meaningless	1	0.99	0.99
Normal Commen	1	0.97	0.98
Religiously Abusive	1	0.99	1
Sad	0.99	0.99	0.99
Abusive	0.98	0.93	0.96
Supportive	1	0.99	0.99

b) Precision, recall and f1-score of SVM model

Figure 4.9 Performanc matirces of linear SVM model to detect sentiment of the training data.

The f1-score of one was the highest for the 'Religiously_abusive' class. Also, other classes had an f1-score above 90%, indicating model stability. The lowest f1-score was for the

'Aggressive' class. The reason was that its precision value was less than the other classes. The precision value for the 'Aggressive' class was 91%. Because of the high Type-1 error or FP value, this precision value decreases. The FP value for this 'Aggressive' class was 27, higher than others. Here, the linear SVM mispredicted 9% of aggressive comments, and only 2% of actual aggressive comments were classified as other classes. Similarly, sensitivity and rate of misclassification can be calculated for other classes.

The linear SVM accurately predicts 1648 comments in a total of 1856 comments. Hence, the accuracy of the linear SVM is 88.79%. Also, this model can predict all the sentiment class comments.

4.1.5 Random Forest Algorithm

The RF algorithm consists of an ensemble of decision trees, with the results of each tree being combined to produce a more accurate result. It is known as the "bagging" method to combine the results. The number of trees is not fixed. One can run as many trees as one wants to cause the RF model does not overfit. However, it is suggested to train the model with a large number of trees.

RF model performance for testing data

The number of trees in this study started at 100 and was subsequently increased to 170 for the RF model. The accuracy for a specific number of trees is shown in Table 4.7.

Table 4.7 Accuracy of RF algorithms for different number of trees.

Number of Trees	Accuracy
100,	59.698%
110	57.759%
120	57.112%
130	57.759%
140	58.621%
150	59.052%
160	57.759%
170	59.483%

Accuracy fluctuated with the number of trees, sometimes rising and sometimes falling. The accuracy percentage for trees 100, 110, and 160 was 57.759%. The lowest accuracy was for

120 trees, and the highest was for 170 trees. As the RF model achieved the highest accuracy with the minimum number of trees of 100, the other performance metrics were analyzed for this number of trees. The confusion matrix for the RF model is shown in Figure 4.10.

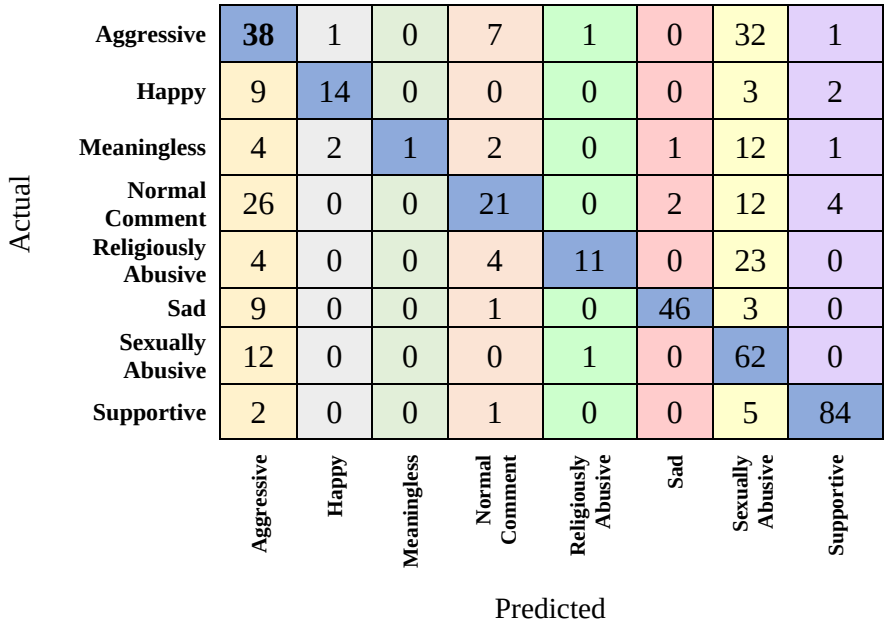


Figure 4.10 Confusion matrix of RF model to detect sentiment of the test data.

The "sexually abusive" class's RB model TP value was 62, FP value was 93, and FN value was 13. The 'meaningless' class had the lowest TP value, and the 'supportive' class had the highest. The TP value indicates that among 92 supportive comments, the RF model detects 84 comments accurately. The highest TN value of 433 was for the 'happy' class, and the lowest value of 296 was for the 'sexually_abusive' class. The TP. TN, FP, FN values of the stntiment class obtained by the RF model is shown in Table 4.8.

Table 4.8 TP, TN, FN, FN value of the RF model to detect sentiment

Value / Sentiment	FP	FN	TP	TN
Aggressive	56	42	38	328
Happy	3	14	14	433
Meaningless	0	21	1	441
Normal_comment	15	44	21	384
Religioulsy_abusive	2	31	11	420
Sad	3	13	46	402
Sexually_abusive	93	13	62	296
Supportive	8	8	84	364

From the value of FP, FN, TP, and TN, the precision, recall, and f1 score can be calculated for each class. . The precision, recall, and f1 score for the RF model is shown in Figure 4.11. If we consider the 'Sad' class, precision indicates 94% correct was the identification of the

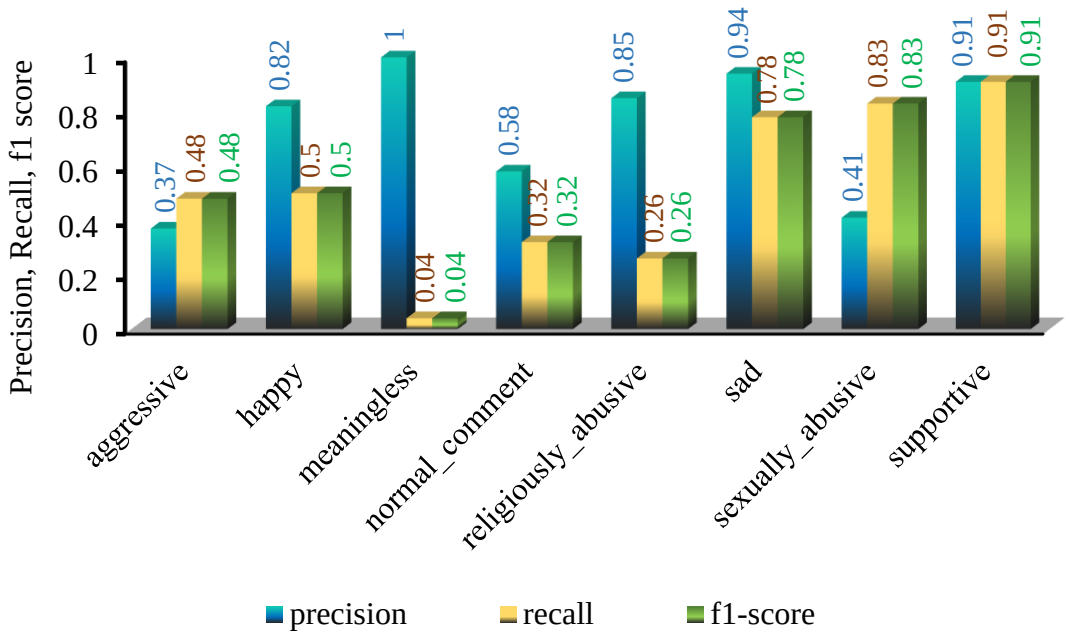


Figure 4.11 Precision, Recall and f1 score of RF algorithm to detect sentiment

'Sad' comment. 0.78% of actual 'Sad' comment was identified correctly. The RF mode with 100 trees perfectly classifies 78% of observations into the correct 'Sad' class.

The highest precision was for the 'meaningless' class, but the recall and f1 score of this class was lower than the precision value. That means the RF model classified a few 'meaningless' comments, but most were correct. This condition was also true for the 'happy', 'normal_comment', 'religiously_abusive' and 'sad' classes. The opposite condition was observed for the 'aggressive,' 'sexually_abusive,' and 'supportive' classes.

The 'aggressive' and 'sexually_abusive' classes had a lower precision value than the recall and f1 scores. In this case, the RF classifier returned many results, but most predictions were incorrect. Additionally, the supportive class's precision, recall, and f1-score were 91%. That means the RF classification model with random states of 20 and 100 trees provided the best result to detect the 'supportive' comments.

Like the KNN and Naïve Bayes model, the RF model shows most of the misclassification for the 'aggressive' and 'sexually_abusive' classes. The reason is that most Bengali slang

words are used in both types of comments. For instance, 'ও আমার বাল বুঝে …' is an aggressive comment in which slang word is 'বাল.' This slang word is also found in the 'sexually_abusive' and 'religioulsy_ abusive' words.32 'aggressive' comments and 23 'religiously_abusive' comments were detected as 'sexually_abusive' comments by the RF model.

RF Model Performance for Training Data

The estimator value of the RF model varied, but the accuracy value remained the same at 98.060%. The confusion matrix for the RF model with the estimator value of 100 is shown in Figure 4.12. From the confusion matrix, it is seen that the total FP and FN value is 27, and the TP value is 1365. Hence the error rate is identical for type1 and type2 errors.

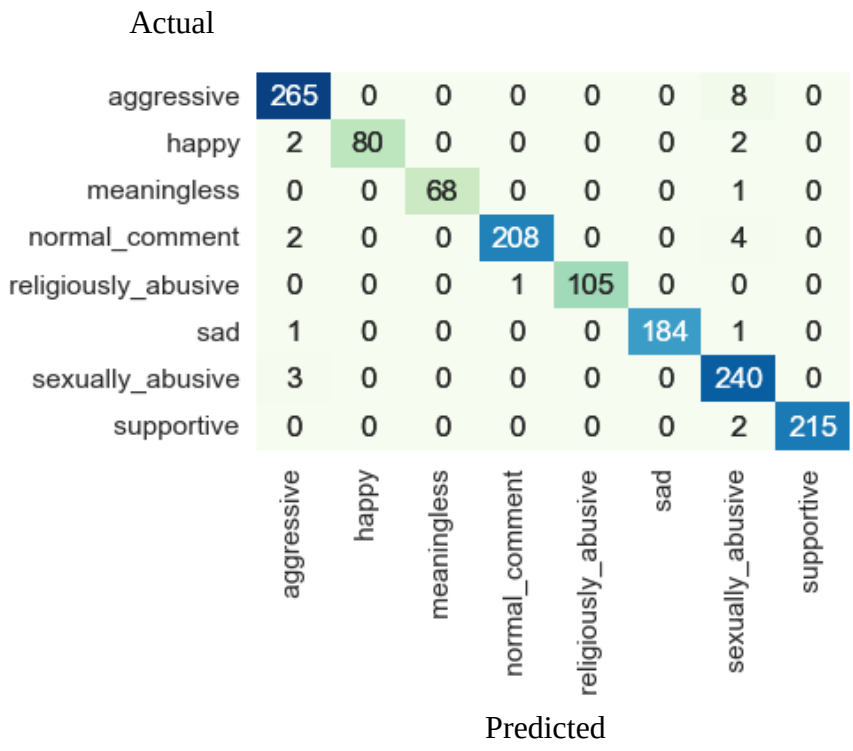


Figure 4.12 Confusion matrix of the RF model to detect sentiment of the training data.

The highest type1 error, FP value, was for the 'Sexually abusive' class; 18's sexually abusive comment was mispredicted. 8 was the aggressive class's highest type2 error or FN value. The precision, recall, and f1 score for each class was estimated using the FP, FN, and TP values, as shown in Table 4.9.

The average precision, recall, and f1 score was 99%, 98%, and 98%, respectively. That showed that the RF model accurately predicted and classified most comments. If there are 1000 comment then 999 commetn will be predicted accurately.

Table 4.9 The precision, recall and f1 score of the RF model to detect sentiment of the training dataset.

Sentiment	Precision	Recall	F1-score
Aggressive	0.97	0.97	0.97
Happy	1	0.95	0.98
Meaningless	1	0.99	0.99
Normal Commen	1	0.97	0.98
Religiously Abusive	1	0.99	1
Sad	1	0.99	0.99
Abusive	0.93	0.99	0.96
Supportive	1	0.99	1

Out of the 1856 comments, the RF correctly predicted 1642, giving it an accuracy of 88.469%. In conclusion, it can say that the RF model performance was better than the KNN and Naïve Bayes model as it was able to anticipate all types of sentiment labels. Also, the accuracy of the RF model was 10.56% and 6.25% higher than the KNN and NB models. However, the performance was approximately similar to the SVM model in terms of accuracy and precision value.

4.1.6 Logistic Regression

A linear classifier that uses the sigmoid function as its foundation is the logistic regression algorithm. In this study, logistic regression was implemented using the scikit-learn Python package. The LR model's parameters were adjusted following the dataset. As the sentiment classification was multiclass, the multinomial option was specified for the multiclass parameter, and for the 'solver' parameter, 'lbfgs' was taken. The inverse of regularization strength C's value was 1.0. This value of C indicates a light penalty.

LR model performance for testing data

The confusion matrix of the LR model is shown in Figure 4.13. The diagonal value of the LR model confusion matrix is the correct value predicted by the LR model. This value belongs

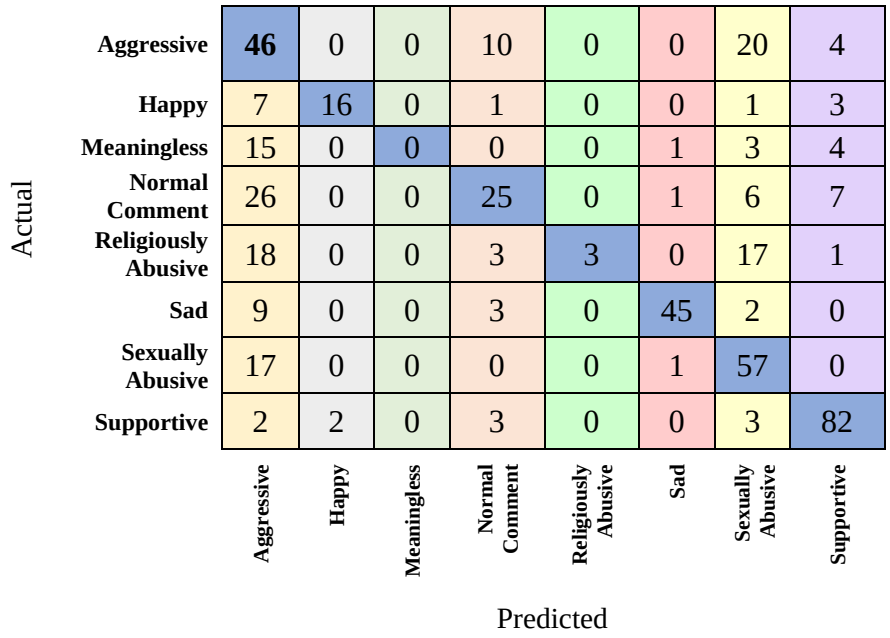


Figure 4.13 Confusion matrix of LR model to detect sentiment of the test data.

to the TP type. The rest of the values of the matrix are, respectively, the number of errors. The most elevated error value of 26 was for the 'normal_comment' class. This value belongs to the FP value. The other error values, FP and FN, for each class are listed in Table 4.10.

Table 4.10 FP and FN value for each sentiment class by the LR model.

Sentiment	**FP**	**FN**
Aggressive	94	34
Happy	2	12
Meaningless	0	23
Normal_comment	20	40
Religioulsy_abusive	0	39
Sad	3	14
Sexually_abusive	52	18
Supportive	19	10

The FP is known as a type-1 error, and FN is a type-2 error. Both errors indicate misclassification. For the 'aggressive' class 94 comment was incorrectly classified that was predicted as aggressive even though it was not. 34's aggressive comments were not predicted as aggressive though they were. Similarly, the FP and FN values indicate misclassified comments for other classes.

The classes "meaningless" and "religiously abusive" had FP values of 0. This FP value of the LR model indicated that no other class comments were categorized as "meaningless" and "religiously abusive." The LR model could not identify any "meaningless" comments, but it correctly predicted 3 "religiously abusive" remarks.

The FN value for the 'meaningless' class was 23, and 39 was for the 'religiously_abusive' class. Most of the 'religiously_abusive' comments were predicted as 'aggressive' and 'sexually_abuisve' comments. Using the TP, TN, FP, and FN values of each sentiment class, each class's precision, recall, and f1 score was calculated and shown in Figure 4.14.

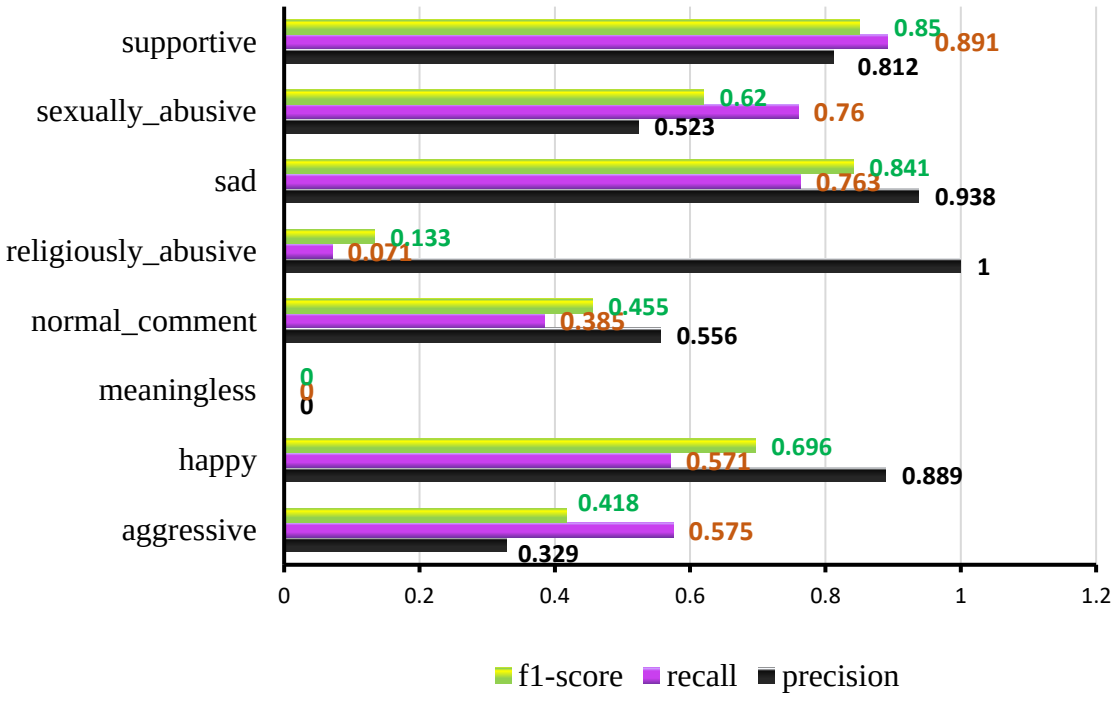


Figure 4.14 Precision, Recall and f1 score of LR algorithm to detect sentiment.

The highest precision value of 1 was achieved for the 'religiously_abusive' class, and the lowest 0 was for the meaningless class. That means 100% of the 'religiously_abusive'

comment that was positive was correctly classified, and for the 'meaningless,' this was zero percent. The formula to calculate the precision value for each class was:

$$Precision_{ra} = \frac{TP_{ra}}{TP_{ra} + FP_{ra}} = \frac{3}{3+0} = 1 \quad (4.2)$$

$$Precision_{meaningless} = \frac{TP_{meaningless}}{TP_{meaningless} + FP_{meaningless}} = \frac{0}{0+0} = 0 \quad (4.3)$$

The highest recall value or sensitivity of 89.1% was for the supportive class. That means that among 92 supportive comments, 89.1% of comments were correctly identified. The f1 score was 85% for this class. Only 3's supportive comment was misclassified as sexually_abusive. 2's comment was for each class of aggressive and happy. The formula to calculate the recall value for the 'supportive' class is:

$$Recall_{supportive} = \frac{TP_{supportive}}{TP_{supportive} + FN_{supportive}} = \frac{92}{92+10} = 90\% \quad (4.4)$$

It was observed that the LR model identified a more supportive class than the other sentiment class.

The LR model acquired 59.052% accuracy, which was better than the KNN and NB models. Like KNN and NB, this model also misclassified most of the aggressive and sexually_abuisve comments. However, this model was unable to detect meaningless comments for testing data. So, the performance of training data also has to observe. If the model cannot identify one of the sentiment classes, it cannot be considered for sentiment detection.

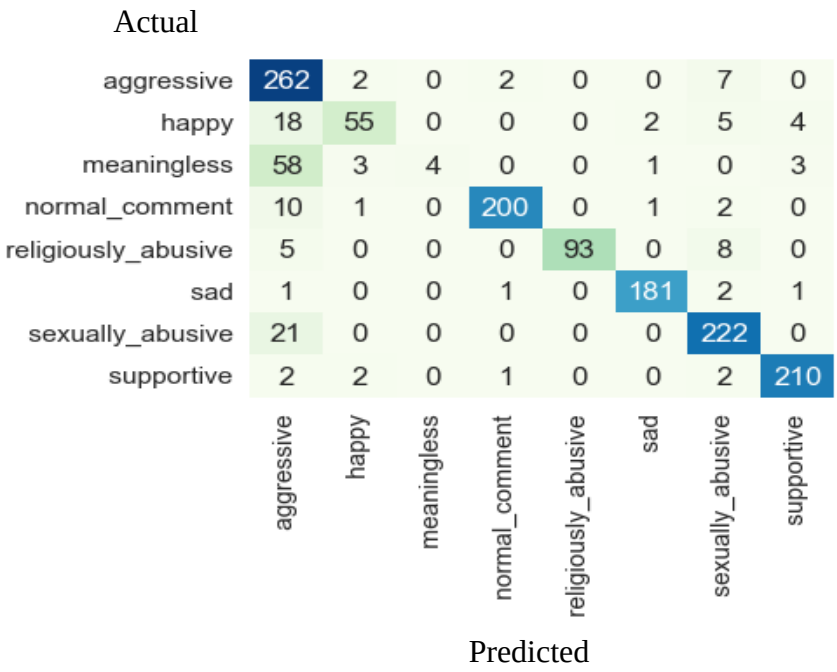
LR model performance for training data

The performance of the LR model is shown in Figure 4.15. The cumulative TP value for the training dataset was 1227, and FP and FN values were 165.

The highest type1 error value was for the 'Aggressive' class. The highest 58 comments were predicted to an aggressive comments, but they were meaningless comments. This class's actual positive prediction ratio was 69%, and sensitivity was 96%.

The highest type2 error value of 65 was seen for the meaningless class as most of the meaningless comment was predicted to an aggressive comment. For this class, the FP value

was zero; hence the ratio of accurate optimistic prediction was 100%. However, by the LR model, the misclassification ratio was 94% for meaningless classes. Instead of this class, the other class's f1 score was above 80%.



a) Confusion matrix of the LR model to detect sentiment.

Sentiment	Precision	Recall	F1-score
Aggressive	0.69	0.96	0.81
Happy	0.87	0.65	0.75
Meaningless	1.00	0.06	0.11
Normal Commen	0.98	0.93	0.96
Religiously Abusive	1.00	0.88	0.93
Sad	0.98	0.97	0.98
Abusive	0.90	0.91	0.9
Supportive	0.96	0.97	0.97

b) Precision, recall and f1-score for the LR model

Figure 4. 15 The Performance meritces for the LR model to detect Bengali sentiment fo training data

The average accuracy of the LR model is 80.87%, as it detects 1501 comments accurately among 1856. However, for newly seen data, its performance was not acceptable.

4.1.7 AdaBoost

While using the Adaptive Boosting classifier, the number of estimators varied with learning rate 1.0 and random state 1. The number of estimators experimented from 50 to 100. The accuracy of the AdaBoost model for each estimator is listed in Table 4.11.

Table 4.11 Accuracy of AdaBoost model for distinct estimator value.

Estimator	50	60	70	75	80	90	100
Accuracy	32.974%	34.914%	35.991%	38.793%	**39.009%**	38.362%	37.931%

The accuracy value fluctuated. In the beginning, the accuracy value increased with the estimator value. The highest accuracy value of 39.009% accuracy value was achieved with 80 estimators. After that, the accuracy value begins to decrease. So, the AdaBoost model with 80 estimators was determined for performance analysis. The confusion matrix for this AdaBoost model is shown in Figure 4.16.

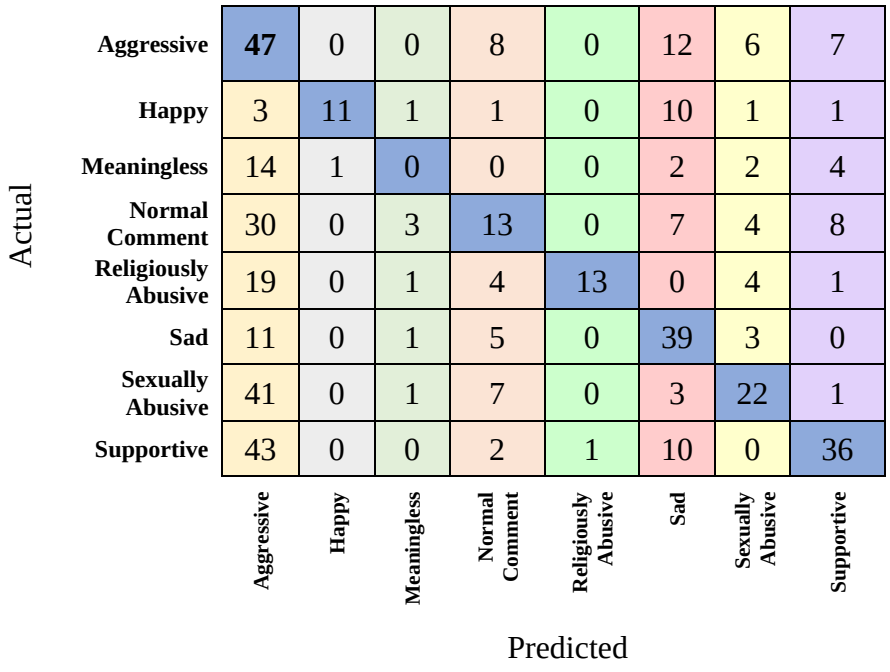


Figure 4.16 Confusion matrix of AdaBoost model to detect sentiment of the test data.

The accuracy value alone is insufficient to choose the categorization model. One can view the precision, recall, and recall performance matrices from the confusion matrix. According to the AdaBoost confusion matrix, the TP values for the aggressive, happy, meaningless,

normal_comment, religiously_abusive, sad, sexually_abusive, and supportive classes are 47,11,0,13,13,39,22, and 36, respectively. This TP value is kept in the matrix's diagonal.

The FN value for each class is the total number of comments of each actual class row, except the TP number. Each predicted class column's total number of comments except the TP value is the FP value for each class. For instance, for the sad class, the TP value was 39, FP was 44, and FN was 20. Using the TN, FP, TP, and FN value, the precision, recall, and f1 score for each class was calculated. The precision and recall values are shown in Figure 4.17 and Figure 4.18 respectively

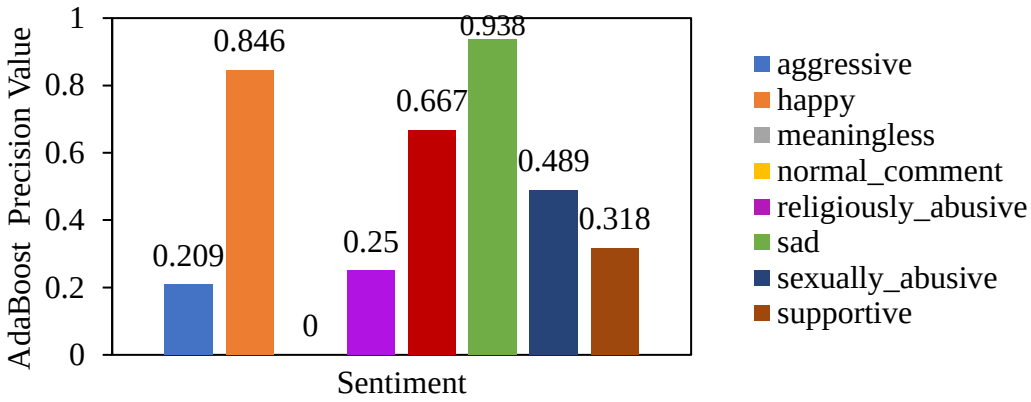


Figure 4.17 The precision value of AdaBoost algorithm for sentiment detection.

The highest precision values of 93.8% and the lowest of 0% were for the sad and meaningless classes, respectively. This precision value of the sad class indicates that among the total predicted 83 sad comments, 93.8% was the sad comment, and 6.2% of prediction for this sad class was wrong.

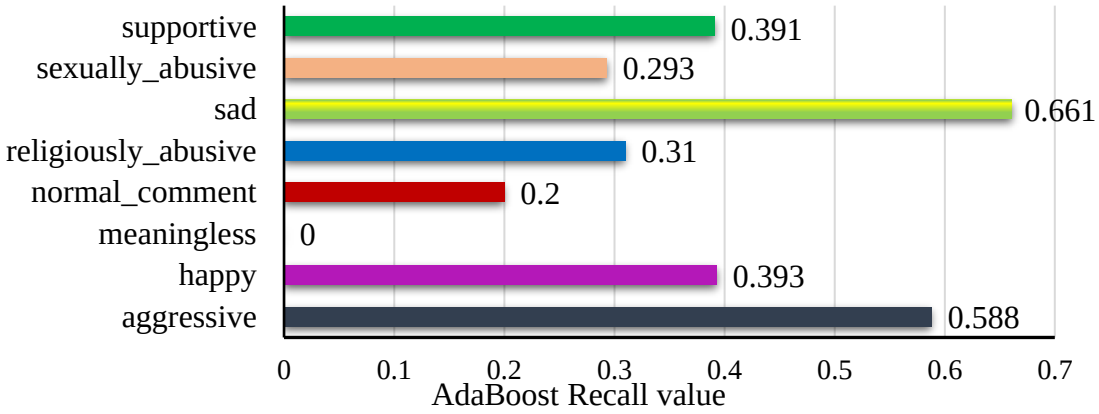


Figure 4.18 The recall value of AdaBoost algorithm for sentiment detection

Similarly, the percentage of the inaccurate prediction was 79.1%, 15.4%, 100%, 75%, 33.3%, 51.1%, and 68.2% for the aggressive, happy, meaningless, normal_comment, religiously_abusive, sexually_abusive, and supportive class, respectively. This wrong prediction for each class is based on the total comment predicted for each class.

The recall value achieved by the AdaBoost classifier is shown in Figure 4.13. The recall value of 66.1% was the highest for the sad class. This recall value defines that 66.1% comment as being classified as the sad comment among 59 actual sad comment. Among 59 actual sad comment 33.9% was predicted wrongly. The formula to calculate recall value for the sad class is:

$$Recall_{\ sad} = \frac{TP_{sad}}{TP_{sad} + FN_{sad}} = \frac{39}{39 + 20} = 66.1\% \qquad (4.4)$$

Similarly, for the supportive, sexually_abusive, religiously_abusive, normal_comment, meaningless, happy, aggressive the incorrect classification was 60.9%, 70.7%, 69%, 80%, 100%, 60.7% and 41.2%, respectively. This AdaBoost model was unable to predict the meaningless comment accurately.

Each class's recall and precision value were acceptable except for the meaningless class. Among used 6 ML classifiers to classify sentiment, the AdaBoost had the lowest accuracy. The accuracy value of AdaBoost had a 10.129% difference with the second lowest accurate KNN model.

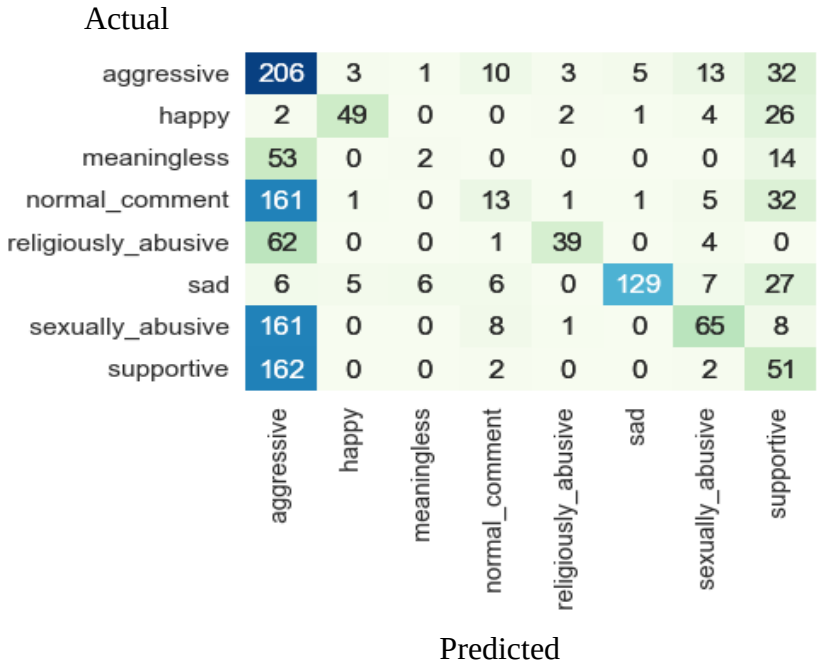
AdaBoost Classifier Performance for Training Data

The AdaBoost classifier could predict 554 comments accurate among 1392 comments. So, 39.799% is the AdaBoost classifier's accuracy detecting the training data's sentiment. The formula of accuracy:

$$Accuracy_{AdaBoost} = \frac{\begin{matrix} Accurately \\ predicted\ comments \end{matrix}}{Total\ comments} = \frac{554}{1392} = 0.39799 \qquad (4.5)$$

The performance matrices of the Adaboost algorithm for sentiment detection are shown in Figure 4.19. The highest misprediction was shown for the 'Aggressive' class. Most of the 'Sexually_abusive,' 'Supportive,' and 'Normal_comment' was predicted to an aggressive comment. The highest 201 comments of the 'Normal_comment' class were misclassified.

From the precision value, it is seen that this model misclassified most of the meaningless comments. The ratio of misclassification was 97% for meaningless comments. Only 22% prediction was accurate among the total predicted meaningless comment of 9.



a) Confusion matrix of the LR model

Sentiment	Precision	Recall	F1-score
Aggressive	0.25	0.75	0.38
Happy	0.84	0.58	0.69
Meaningless	0.22	0.03	0.05
Normal Commen	0.33	0.06	0.1
Religiously Abusive	0.85	0.37	0.51
Sad	0.95	0.69	0.8
Abusive	0.65	0.27	0.38
Supportive	0.27	0.24	0.25

b) Precision, Recall and f1- score of AdaBoost classifier

Figure 4.19 Performance metrices of AdaBoost model to detect sentiment for training data

Here, the AdaBoost classifier was worse than the other classifier detecting Bengali sentiment. However, there are fewer differences between training and testing data performance.

4.2 Comparative Analysis of Algorithms

Among six examined ML algorithms, three, namely KNN, SVM, and RF, could detect all eight types of the sentiment of the comment if comment is newley seen. The precision and recall value of those three algorithms for each sentiment class is listed in Table 4.12. The other three algorithms, namely NB, LR, and AdaBoost, cannot detect only meaningless comment classes. The sample of sentiment predictin of Bengali and transliterated Bengali text by ML classifers is shown in Table 4.13.

Table 4.12 The value of performanc metrices of the KNN, SVM and RF algorithm for testing data

Sentiment	**Precision**			**Recall**			**F1 score**		
	KNN	**SVM**	**RF**	**KNN**	**SVM**	**RF**	**KNN**	**SVM**	**RF**
Aggressive	0.379	0.357	0.37	0.275	0.588	0.48	0.319	0.443	0.48
Happy	0.8	0.889	0.82	0.429	0.571	0.5	0.558	0.696	0.5
Meaningless	0.5	1	1	0.043	0.13	0.04	0.08	0.231	0.04
Normal comment	0.583	0.544	0.58	0.108	0.477	0.32	0.182	0.508	0.32
Religioulsy abusive	0.333	0.934	0.85	0.024	0.333	0.26	0.044	0.491	0.26
Sad	0.974	0.893	0.94	0.627	0.847	0.78	0.763	0.87	0.78
Sexually abusive	0.267	0.598	0.41	0.893	0.733	0.83	0.411	0.659	0.83
Supportive	0.953	0.902	0.91	0.88	0.891	0.91	0.915	0.896	0.91

The precision value of RF and SVM was higher than the KNN, but they are near each other. The f1 score for each classifier was utilized to compare the performance of the two algorithms because they both offer higher precision and recall.

For testing data, the RF classifier's f1 score for detecting sentiment was 53.07 %, compared to the SVM's f1 score of 59.92 %. The accuracy of the SVM model was also better than the RF model. 5.127% was the accuracy difference between the SVM and the RF algorithm. Hence, in terms of f1 score and accuracy, the SVM model is better than the other five examined ML algorithms to detect Bengali text sentiment.

The RF and the SVM model had an accuracy above 97% for training data. The difference is only 0.718%. However, for training data, it was observable that the accuracy value of the

SVM classifier fluctuated with the C's value. The RF model's accuracy was not changed for a different number of estimators. The TP value for the RF and the SVM is shown in Table 4.13.

Table 4.13 The number of accurately predicted comments by the RF and SVM model.

Classifier	TP		Total TP	Accuracy
	Test Data	Train Data		
SVM	293	1355	1648	88.79%
RF	277	1365	1642	88.469%

Table 4.13 shows that the performance of both models was better than the other model. The RF model accurately predicted the highest 1365 comments in the training dataset. The SVM predicts more accurately than the RF model for new data to the model.

4.3 CONCLUSION

Early detection and prevention to spread transliterated Bengali cyberbullying can reduce impact of the victim. This Bullying detection can be done accurately with the help of sentiment score. To detect sentiment of all types of Bengali comment like code-mixed, transliterated and local, the SVM and the RF model can be preferable. The sentiment detect from this model can be used to as a input feature to detect bullying comment from the social media. In Chapter 5 the performance of the model to detect Bengali bullying utilizing the sentment of the text is presented.

CHAPTER 5

MODEL FOR BULLYING DETECTION

In this chapter, the model's performance in detecting bullying utilizing the sentiment of the Bangla text achieved in the previous chapter is analyzed. Like the previous chapter, this chapter has two sub-sections: performance analysis and comparative examination of the models.

5.1 PEERFORMANCE ANALYSIS OF THE MODEL

The dataset used for bullying identification includes user comments, the number of likes and dislikes, emojis, punctuation, abusive words, positive words, and emotion.TF-IDF feature extraction was applied to the user comments and dummies variable technique to the sentiment features. After that, the ML classifiers were applied to this processed dataset. The performance of the ML classifiers is demonstrated below.

5.1.1 KNN Algorithm

The KNN model works based on the votes from the K number neighbors. For each comment of the test dataset, the algorithm will look for K's comments that are closest to the new comment. To find the highest accuracy, the value of K's varies from 1 to 8. To measure the distance, "Manhattan Distance" was used with the parameter p=1. The accuracy values for distinct K values with parameter p=1 are listed in Table 5.1.

Table 5.1 Accuracy of KNN model for distinct k values to detect bullying comment.

K	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Accuracy	87.93%	88.58%	87.72%	89.87%	88.15%	89.23%	88.58%	87.77%

There was accuracy between 87% to 90% for K's values from 1 to 8. The value was increased at the beginning from K's value 1 to 4. After that, the accuracy value fluctuated. The accuracy value stated to decrease after the value of K=6. The accuracy of KNN with K=4 and K=6 provides accuracy above 89%. However, in term of precision and accuracy, the KNN with K=4 perform better than K=6. The precision, recall, and f1 score for KNN with K=4 are documented in Table 5.2.

Table 5.2 The performance matrix of KNN to detect bullying.

Performancce Matrix	**Bullying**	**Neutral**	**NonBullying**
Precision	92.30%	74.30%	93.10%
Recall	99.00%	74.30%	85.60%
f1-score	95.60%	74.30%	89.20%

92.3% out of the total projected bullying comments were correctly predicted. Also, for the Bullying comment, 1% prediction was wrong among the total actual bullying comment. 25.7% prediction was wrong among total predicted neutral comments. Among predicted NonBullying comments, 6.9% were misclassified. Here, the KNN model's highest precision and recall were for the Bullying class.

5.1.2 Naïve Bayes Algorithm

Multinomial NB with parameter alpha=1.0 provided 83.836% accuracy. The NB algorithm work based on the Bayes theorem. If any of the feature-based probabilities become zero, it creates a problem. In order to eradicate this problem, Laplace smoothing was used. The confusion matrix of the NB model to detect bullying comments is pictured in Figure 5.1.

The TP values for the classes of Bullying, Neutral, and Nonbullying were 202, 18, and 169, respectively. Bullying, Neutral, and Non-Bullying had an FN value of 4, 52, 19. The FP values for the NonBullying, Neutral, and Bullying classes were 47, 6, and 22, respectively. The above value was obtained based on support comments of 206, 70, and 188 for the Bullying, Neutral, and NonBullying class.

The highest 43's Neutral comment was detected as a NonBullying comment. This 43's Was the highest FP value of the NonBullying comment. Defines the misprediction of the NB model for the NonBullying comment was more than the other two classes. The neutral comment had the highest FN value, clarifying that most of the actual Neutral comments were classified erroneously. Also, the lowest TP was for the Neutral class, but the support comment was also the lowest for the Neutral class. From the confusion matrix, it has been

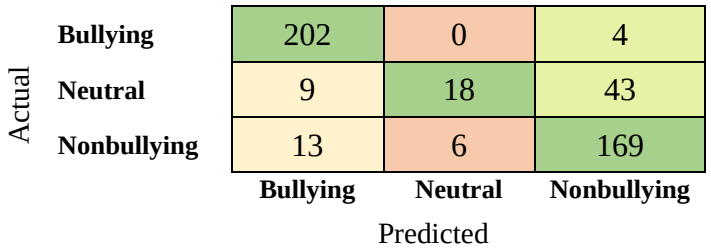


Figure 5.1 Confusion matrix of Naïve Bayes to detect bullying.

observed that there was a conflict between the Neutral and NonBullying classes. This mention is alleviated with the analysis of the precision and recall value. The precision, recall, and f1 score for the NB model are shown in Figure 5.2.

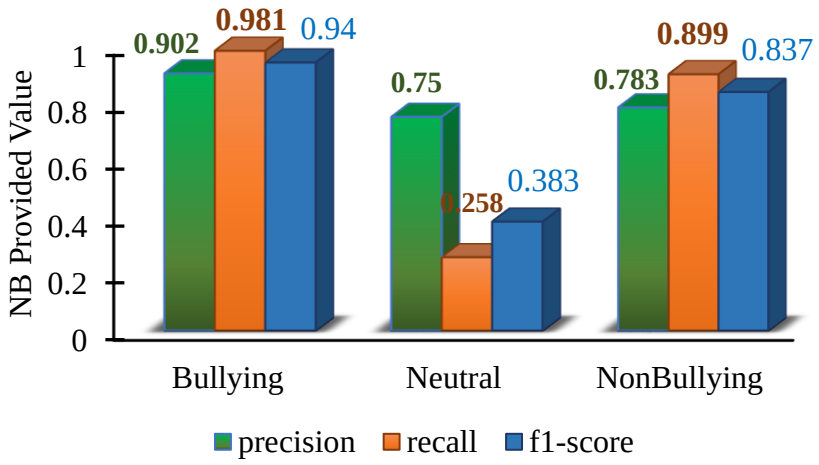


Figure 5.2 The performance matrix of NB model to detect bullying.

The highest precision and recall value were for the Bullying class. As seen, 9.8% of the comment was mispredicted as Bullying comment. The actual Bullying comment was misclassified at 1.9%. The Bullying class had the highest precision and recall, so the f1 score for this class was higher than others.

On the other side, between Neutral and NonBullying classes, Nonbullying class performance was better regarding precision and recall. There was a small amount of diffidence between the precision values, but for the Recall and f1 score, it was not the same. The NonBullying class had 64.1% and 45.4% higher recall and f1 scores than the Neutral class. Hence, the NB model could predict the Bullying class better than the other classes.

5.1.3 SVM Algorithm

The linear SVM classifier was implemented to detect bullying. The parameter c's value was altered and checked accuracy value. Accuracy values achieved for each c value are enumerated in Table 5.3.

Table 5.3 The accuracy of liner SVM for distinct c's value.

C's value	1	2	3	**4**	5	6	7
Accuracy	93.750%	93.319%	93.750%	**94.828%**	94.612%	94.612%	94.397%

The linear SVM with c=4 achieved the highest accuracy. The accuracy value first improved along with the value of c. After reaching the highest accuracy of 94.83%, the value dropped. Additionally, the polynomial and the 'RBF' kernel were checked. Both kernel polynomial and RBF provide the highest accuracy of 93.164% and 91.164% for c=4, respectively. The confusion matrix for liner kernel-based SVM is shown in Figure 5.3.

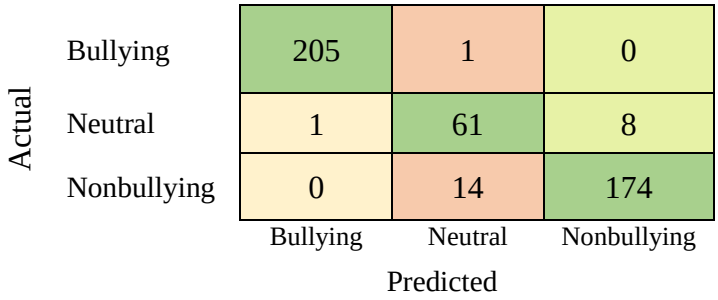


Figure 5.3 Confusion matrix for liner SVM to detect bullying comment.

This linear SVM shows the conflict between the Neutral and NonBullying classes. The TP values were 205, 61, and 174 for the Bullying, Neutral, and NonBullying class. The lowest FP and FN value was for the Bullying class. For the Neutral class, the FP and FN value were

15 and 9, respectively. 8 and 14 were the FP and FN values for the NonBullying class. Using those TP, FP, TN, and FN values, precision, recall, and f1 scores were calculated. For instance, find the precision and recall value for the Neutral class.

$$Precision_{Neutral} = \frac{TP}{TP + FP} = \frac{61}{61 + 15} = 0.803 \tag{4.5}$$

$$Recall_{Neutral} = \frac{TP}{TP + FN} = \frac{61}{61 + 9} = 0.871 \tag{4.6}$$

$$f1\ score_{Neutral} = 2 * \frac{Precision * Recall}{Precision + Recall} = 2 * \frac{(0.803 * 0.871)}{(0.803 + 0.871)} = 0.836 \tag{4.7}$$

Besides, the precision, recall, and f1 score for other classes were also calculated and shown in Figure 5.4.

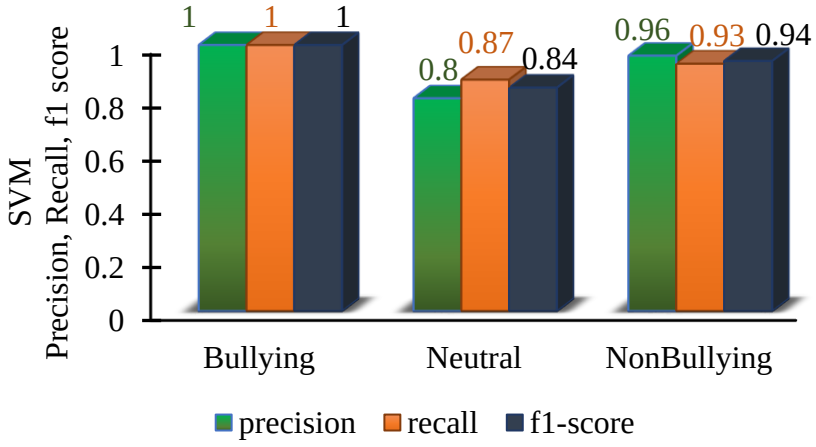


Figure 5.4 The value of performance matrices of SVM to detect bullying.

The highest precision, recall, and f1 score of 1 was shown for the Bullying class. 13% of actual Neutral comments were misclassified, and 20% were wrongly predicted. For the NonBullying class, the wrong prediction was 4% among all predicted NonBullying comments. 7% NonBullying comment was misclassified. The NonBullying class achieved a 10% more prominent f1 score than the Neutral class. From the precision, recall, and f1 score, it is observed that the liner SVM performs better than the previously discussed KNN and NB models.

5.1.4 Random Forest Algorithm

The performance of the RF algorithm was examined with random state values 3. The number of estimators gradually increased from 100 to 250. The accuracy value obtained from the RF model is indicated in Table 5.4.

Table 5.4 Accuracy value of Rf model to detect bullying for the specific number of trees.

N estimators	100	150	170	200	250
Accuracy	93.966%	93.996%	94.397%	94.397%	94.181%

The accuracy value was maximum for the RF model with 170 and 200 estimators. The accuracy percentage held steady at 93.996% for 100 to 150. The accuracy rate was 94% between 170 and 200. After that, the accuracy value started to decline. The performance of the RF model with 170 estimators is examined as it retained the highest accuracy. The confusion matrix for the RF model is illustrated in Figure 5.5.

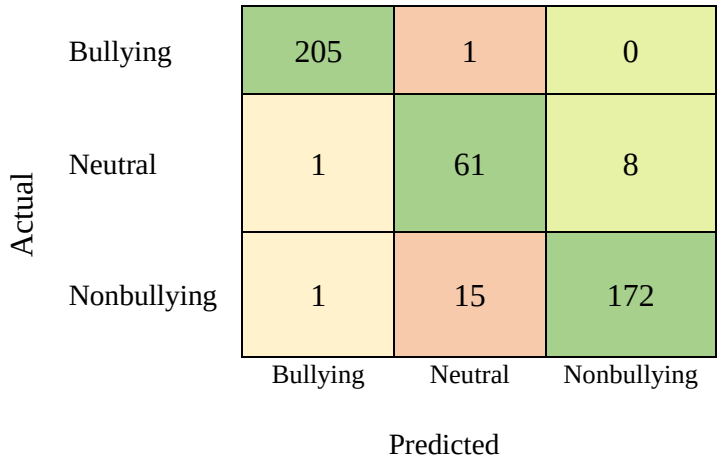


Figure 5.5 The confusion matrix of RF model to detect bullying.

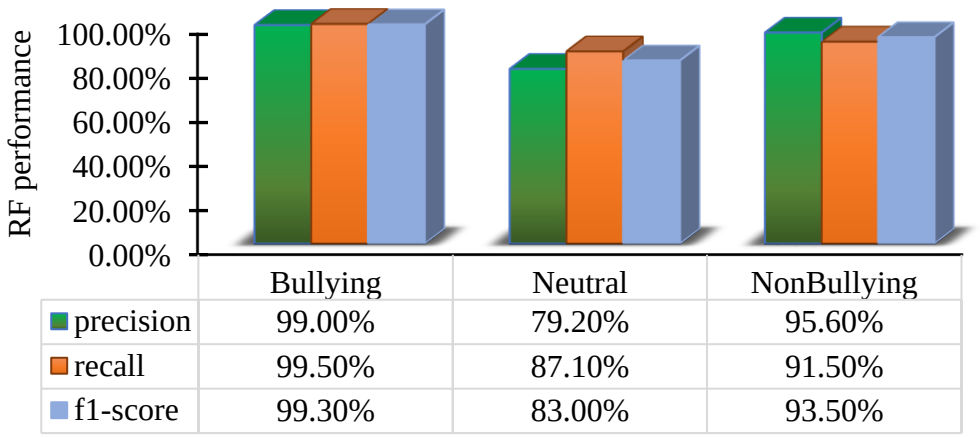
Like the SVM model, the RF model could also predict 205 comments accurately. This 205 is the TP value of the Bullying class. The FN and FP values for this class were 1 and 2 and had a TN score of 256. The TP, TN, FP, and FN values were likewise calculated for the other two classes. The TP, TN, FP, and FN value for each class found by the RF model is demonstrated in Table 5.5.

Among 206 actual bullying comments, one comment was misclassified, and two comments were wrongly predicted as bullying comments. 9's comment on the Neutral class was not classified as Neutral. The number of comments was 16, not in the Neutral class but predicted

Table 5.5 The TP, FP, FN and TN value of the RF model to detect bullying.

Label	TP	FP	FN	TN
Bullying	205	2	1	256
Neutral	61	16	9	378
NonBullying	172	8	16	268

as Neutral. The conflict of the class prediction was shown between the Neutral and NonBullying classes. 8's actual Neutral comment was detected as a NonBullying comment, and 15's NonBullying comments were detected as Neutral comments. The precision, recall, and f1 score from the confusion matrix of the RF model is demonstrated in Figure 5.6.



	Bullying	Neutral	NonBullying
precision	99.00%	79.20%	95.60%
recall	99.50%	87.10%	91.50%
f1-score	99.30%	83.00%	93.50%

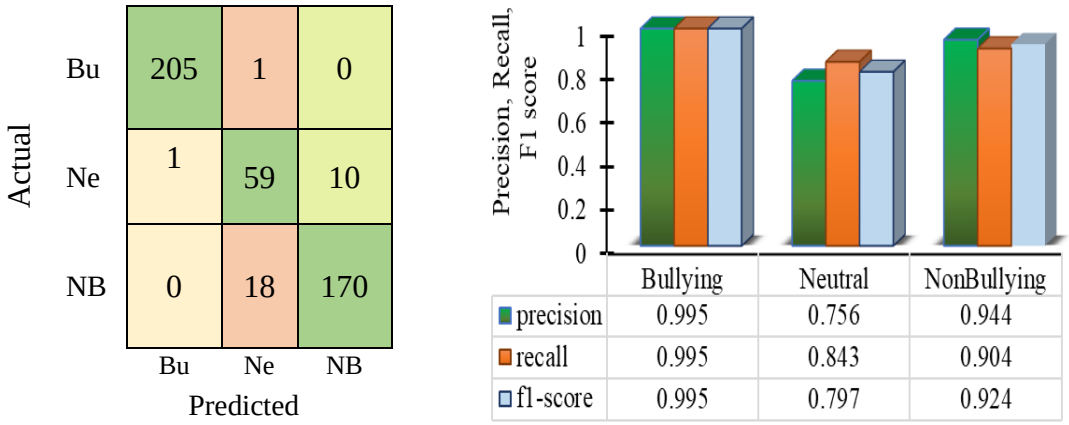
Figure 5.6 The precision, recall and f1-score of the RF model to detect Bullying.

From the precision and recall value, it was found that the RF model provides the highest misprediction of 20.8% for the neutral class. The lowest misprediction of 1% was for the Bullying class. It also showed less misclassification for the Bullying class. The distinction of the wrong prediction was 3.4% between the Bullying and the NonBullying class.

The RF model performs better than the KNN and the NB model in detecting bullying comments. However, the performance of the SVM and the RF model is near each other. As the RF model can detect all three types of comments and provide better precision value, this model can be regarded as detecting Bengali bullying comments.

5.1.5 Logistic Regression

The Bengali text bullying detection is the multiclass classification. For the multiclass classification, the multinomial LR model is the best choice. Hence, the multinomial LR with the maximum iteration of 400 was examined. The parameter solver was specified to 'lbfgs,' and multi_class set to multinomial for the LR model. The confusion matrix and performanc matrices for the LR model is illustrated in Figure 5.7.



a) Confusion Matrix b) Performance metrices

Figure 5.7 Confusion matrix, Precision, Recall and f1-score of the LR model to detect bullying.

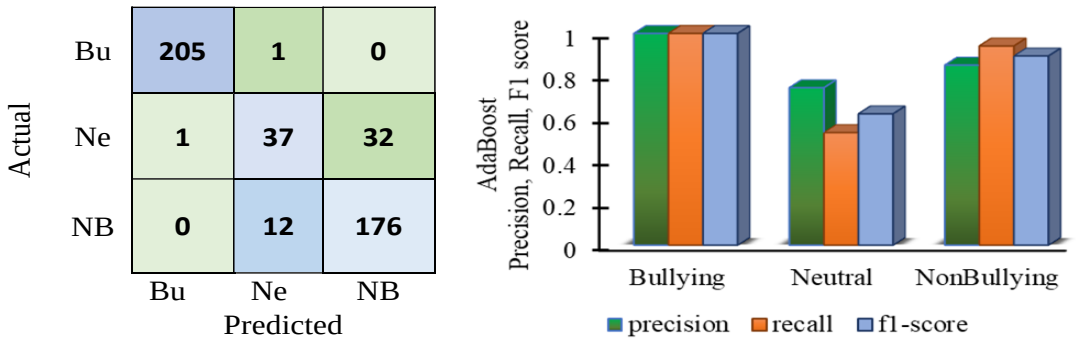
The diagonal value of the confusion matrix defines the number of accurately predicted comments. For instance, if a new bullying comment is classified as Bullying, it will be TP. If a bullying comment is not predicted as Bullying, it will be the FP or FN value.The TP value for the Bullying, Neutral, and NonBullying class was 205, 59, and 170, respectively. 1, 19, and 10 was the type-1 error, namely the FP value for the Bullying, Neutral, and NonBullying class. For Bullying, Neutral, and NonBullying Type-2 error values were 1, 11, and 18.

The precision, recall, and f1 score for the Bullying class was 100%. The Neutral class had a 75.6% precision value. Hence, the LR model correctly detected 75.6% of the total predicted Bullying comments. The sensitivity for the Neutral class was 84.3%. That means the LR model misclassified 15.7% of comments among 77 neutral comments. 5.6% NonBullying comment was also wrongly predicted as NonBullying comment. The amount of wrong misclassification was 9.6% for this class by the LR model.

The LR model provides an accuracy of 93.534% and average precision and recall of 91%. The recall and precision values were more significant than the KNN and NB model. However, the LR model with iteration 400 can also be considered to detect Bengali text bullying. The LR model's accuracy was above 90% and could detect all classes.

5.1.6 AdaBoost

The Decision tree classifier with maximum depth one was used for the AdaBoost algorithm. The learning rate was 1.0, and the random state was 4. To find the best accuracy, the number of estimators was varied and studied the result. This number of estimators defines the maximum number of estimators at which the boosting has to terminate. In this study, parameter n_estimators were tested from 100 to 200. With the estimator's values, the accuracy value of 90.086% remains the same. The confusion matrix achieved from the AdaBoost algorithm based on the test data is pictured in Figure 5.8.



a) Confusion Matrix b) Performance Metrices

Figure 5.8 Confusion matrix, Precision, Recall and f1-score of the LR model to detect bullying.

The TP values were 205, 32, and 176 for the Bullying, Neutral, and NonBullying class. That means a total of 413 comments were predicted correctly among 464 comments. FP values were 1, 13, and 32, and FN values were 1, 33, and 12 for the Bullying, Neutral and NonBullying classes, respectively. AdaBoost classifies 32 neutral comments as nonbullying comments and 12 nonbullying comments as neutral. However, the positive side is that any Bullying comment was not detected as a NonBullying comment and vice versa.

5.2 Comparative Analysis of Algorithms

The performance matrices precision, recall, f1-score, and accuracy were used to detect the best-performed algorithm. As the concern was to detect the bullying comment, the first target was to discover the algorithm with the highest precision and recall. The accuracy and precision value for training and testing data per ML classifier is demonstrated in Table 5.6.

Table 5.6 The performance of ML models to detect Bengali text bullying comments.

Performance Matrix		**NB**	**SVM**	**KNN**	**LR**	**AdaBoost**	**RF**
Training Accuracy		90.30%	99.64%	92.53%	97.70%	90.59%	99.86%
Testing Accuracy		83.84%	94.83%	89.87%	93.54%	90.09%	94.40%
Training Precision	Bullying	0.93	1.00	0.95	1.00	1.00	1.00
	Neutral	0.89	0.98	0.75	0.91	0.73	0.99
	NonBullying	0.88	0.88	0.97	0.98	0.86	1.00
Testing Precision	Bullying	0.90	1.00	0.92	1.00	1.00	0.99
	Neutral	0.75	0.80	0.74	0.76	0.74	0.79
	NonBullying	0.78	0.96	0.93	0.94	0.85	0.96

The KNN and NB models showed less than 90% accuracy for the test data. The NB classifier's accuracy differed by 6.46 percent between the training and test data sets. The difference in the accuracy of KNN between training and test data was 2.66%. The AdaBoost classifier had an average of 90.09% accuracy for testing data and 91% for training data.

The SVM, LR, and RF had an accuracy above 90%. The accuracy difference between SVM and LR was 1.29%, and between the SVM and RF model was 0.43% for testing data. For the testing data, the LR and RF models had a 0.86% accuracy difference and a 2.16% difference for training data.

The Table 5.7 demonstrated the bullying prediction of distinct ML classifier with actual level. The TP, FP, TN, and FN values were observed as the accuracy values were near each other. The TP, FP, TN, and FN value for training data is listed in Table 5.8. The LR model's type-1 and type-2 errors were higher than that of the SVM and RF. TP value of bullying class was the same as 610 for the three ML classifier. From the TP, FP, and FN,

Table 5.8 The TP, FP, and FN values of LR, SVM, and RF to classify bullying comments.

Label		LR	SVM	RF
TP	Bullying	610	610	610
	Neutral	185	197	198
	Nonbullying	565	581	582
FP	Bullying	3	0	0
	Neutral	18	3	0
	Nonbullying	11	1	2
FN	Bullying	0	0	0
	Neutral	13	1	0
	Nonbullying	19	3	2

it can be stated that three algorithms, namely LR, SVM, and RF, can be an ideal choice to predict Bengali text-based cyberbullying. As the error rate was less than the TP value, those models can accurately classify most bullying comments. For better investigation, the recall and f1-score of the SVM and RF algorithm were analyzed, listed in Table 5.9.

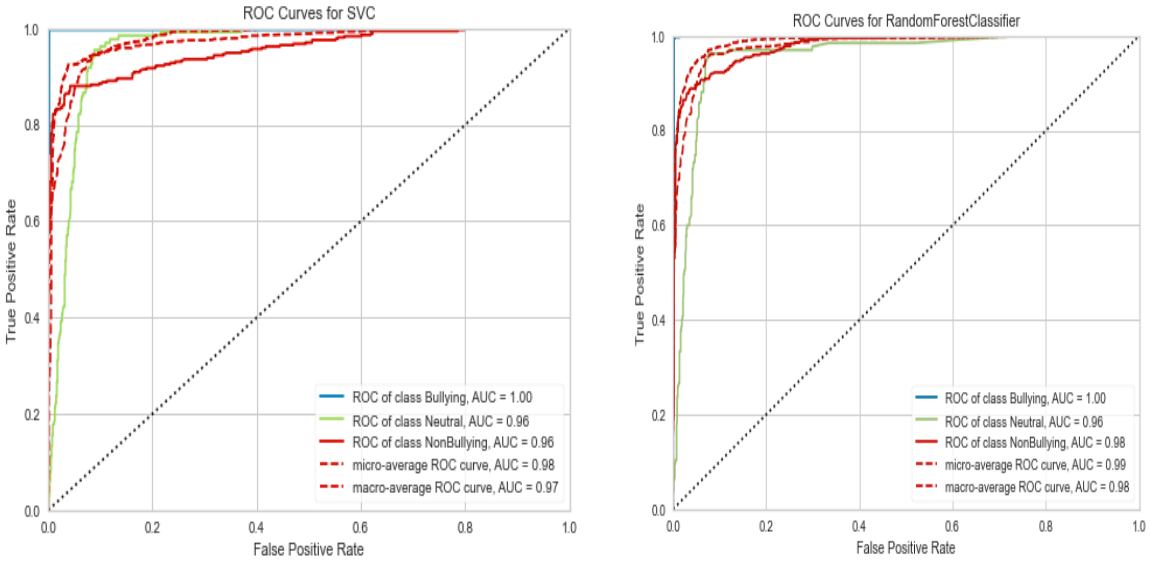
Table 5.9 The recall and f1-score value of the SVM and RF algorithm to detect bullying comments.

Label			SVM	RF
Recall	Training	Bullying	1.00	1.00
		Neutral	0.87	0.87
		NonBullying	0.93	0.91
	Testing	Bullying	1.00	1.00
		Neutral	0.99	1.00
		NonBullying	0.99	1.00
F1 score	Training	Bullying	1.00	0.99
		Neutral	0.84	0.83
		NonBullying	0.94	0.93
	Testing	Bullying	1.00	1.00
		Neutral	0.99	0.99
		NonBullying	1.00	1.00

From the f1-score, it is observed that the RF is a better model for detecting Bengali and transliterated Bengali bullying comments. The SVM model can be a better alternative to avoid misclassification.

However, as the dataset was imbalanced, the AUC-ROC curve was also observed for the RF and SVM classifier, shown in Figure 5.9. The Receiver Operation Characteristic (ROC)

curve plots the true positive (recall) against the false positive ratio. Area Under the Curve (AUC) matrices are used to calculate the overall performance of a classification model based on the area under the ROC curve.



a) ROC curve for the SM b) ROC curve for the RF

Figure 5.9 The ROC of the RF and SVM classifier.

The AUC value of the Bullying class is one for both the RF and SVM model. That is, the highest AUC value indicates the most accurate prediction. Here, it is shown that the difference between the AUC value for the Nonbullying class is 0.1, and no difference for the Neutral class. Hence, the RF and the SVM classifier can be a better choice to detect transliterated Bengali bullying comments.

5.3 SUMMARY

The SVM and RF algorithm performed better than other utilized ML classifiers to detect the sentiment of Bangla text. Similarly, both the SVM and RF perform better than others in terms of accuracy, precision, and recall. However, for the f1-score, the RF algorithm showed better accuracy. Also, the RF algorithm is better for large datasets between SVM and RF algorithms. Hence to detect the sentiment of Bengali and transliterated Bengali text SVM classifier can be a better choice. The social media features and The Bengali text sentiment obtained from the SVM algorithm were fed to the ML model to detect bullying comments. The RF and SVM classifiers can be considered for Bengali text bullying detection as they outperformed other classifiers.

CHAPTER 6

CONCLUSION

This fact indicates that cyberbullying has just become a curse for this generation. This research study tried to find an ML classifier that performs better to detect Bengali and transliterated Bengali text cyberbullying comments. This chapter includes an overview of the research work, the impact of this research, and future work scope.

6.1 SUMMARY

For transliterated Bengali words, there is no standard spelling. Social user does not maintain any grammatical rules. As a result, employing the Lexical procedure to detect offensive transliterated Bengali text is less effective. This study tried to find Machine Learning algorithms to increase accuracy. The contribution of this study can be summarized as follows:

i. There is a lack of a dataset on transliterated Bengali abusive social media text. A corpus of Bengali, code-mixed Bengali, and transliterated Bengali abusive text was created. Data was collected from different social media, like Facebook, Instagram, YouTube, etc.

ii. Using the VADER sentiment analysis package of python sentiment score of Bangla and transliterated Bangle text was checked. Most of the transliterated Bangla text was misclassified.

iii. Train the ML models to classify the sentiment of the Bangal and transliterated Bangla text. The linear SVM classifier with parameter c's value 2 and RF model with 100 estimators outperformed.

iv. The sentiment class and social media features were provided to the ML models. The SVM and RF model had the highest performance in terms of accuracy, precision, recall, and f1-score.

In conclusion, it can be stated that the SVM and RF both algorithms can be preferable for detecting Bengali text-based bullying comments. The SVM can be more fitting for new data, and the RF model can be preferable for large datasets.

6.2 IMPACT OF PROPOSED RESEARCH

Bangladesh had 36.00 million social media users in January 2021 [53]. The number of social media users in Bangladesh increased by 3.0 million (+9.1%) between April 2019 and January 2020. Social media penetration in Bangladesh was 22% in January 2020. Most users use code mixed and transliterated Bengali for posts, comments, or messages. There is no fixed spell for transliterated Bengali words, which is critical to identify bullying content through Natural language processing. Machine Learning algorithms can solve this use.

Cyberbullying poses a threat to adolescents' health and well-being [55]. 49% of Bangladeshi school pupils face cyberbullying [56]. Victims of cyberbullying are more intense to commit suicide than non-victims. In Bangladesh, cybercrime and cyberbullying are increasing. Transliterated Bengali bullying text detection will reduce the impact of cyberbullying. Both the social media platform and Bengali text users will benefit from it. For native Bengali users, the social media platform will be safer.

6.3 FUTURE WORK SCOPE

The corpus created for transliterated Bengali bullying text can be used for additional research purposes. The more accurate and precise model can also be used for future work. Here are some examples of future scope to consider:

i. Detecting transliterated Bengali text-based bullying in streaming data and real-time
ii. Detect the victim's emotional state after a cyberbullying incident
iii. To use the neural network or deep learning to improve classifier performance.
iv. Transliterated Bengali word representation learning

Cyberbullying disrupts the victim's life. It subjected the victim to social, mental, and physical torment. A person's life can be saved through early detection and prevention of the dissemination of hazardous content.

REFERENCES

[1] Dan Olweus, "School bullying: Development and some important challenges," Annual review of clinical psychology, 9, pp. 751-780, 2013.

[2] Dooley, Julian J., Jacek Pyżalski, and Donna Cross. "Cyberbullying versus face-to-face bullying: A theoretical and conceptual review," Journal of Psychology , 217(4), pp. 182-188, 2009.

[3] Myers, Carrie-Anne, and Helen Cowie. "Bullying at university: The social and legal contexts of cyberbullying among university students," Journal of Cross-Cultural Psychology, 48(8), pp. 1172-1182, 2017.

[4] Florida Atlantic University. Nationwide teen bullying and cyberbullying study reveals significant issues impacting youth.
https://www.sciencedaily.com/releases/2017/02/170221102036.htm, Accessed:2 August, 2021.

[5] Pew research center. A Majority of Teens Have Experienced Some Form of Cyberbullying. (Online). https://www.pewresearch.org/internet/2018/09/27/a-majority-of-teens-have-experienced-some-form-of-cyberbullying/, Accessed: 2 August, 2021

[6] Cyberbullying Research Center. 2019-cyberbullying-data. 2019. (Online). https://cyberbullying.org/2019-cyberbullying-data, (Accessed: 19-August-2021)

[7]Cyberbullying Research Center. School Bullying Rates Increase by 35% from 2016 to 2019', 2019. (Online). https://cyberbullying.org/school-bullying-rates-increase-by-35-from-2016-to-2019, Accessed: 19-August-2021

[8] Semiu Salawu; Yulan He; Joanna Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," IEEE Transactions on Affective Computing, 10 October 2017.

[9] Maliha Jahan, Istiak Ahamed, Md Rayanuzzaman Bishwas, and Swakkhar Shatabda, "Abusive comments detection in bangla-english code-mixed and transliterated text," 2nd International Conference on Innovation in Engineering and Technology (ICIET), IEEE, pp.1–6, 2019.

[10] Davidson, T., Warmsley, D., Macy, M., & Weber, I., "Automated Hate Speech Detection and the Problem of Offensive Language," Proceedings of the International AAAI Conference on Web and Social Media, 11(1), pp. 512-515, 2017.

[11] Q. Li, "New bottle but old wine: A research of cyberbullying in schools," Comput. Hum. Behav, 2007.

[12] Qianjia Huang, Vivek K. Singh, Pradeep K. Atrey, "Cyber Bullying Detection Using Social and Textual Analysis," ACM Multimedia, 2014.

[13] Al-Garadi, Mohammed Ali, et al. "Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges." IEEE Access, 7, pp. 70701-70718, 2019.

[14] Chanda, Arunavha, Dipankar Das, and Chandan Mazumdar., "Unraveling the English-Bengali code-mixing phenomenon," Proceedings of the Second Workshop on Computational Approaches to Code Switching, 2016.

[15] Kumar, Ritesh, Bornini Lahiri, and Atul Kr Ojha. "Aggressive and offensive language identification in hindi, bangla, and english: A comparative study." SN Computer Science, 2(1), pp. 1-20, 2021.

[16] Semiu Salawu; Yulan He; Joanna Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," IEEE Transactions on Affective Computing, 10 October 2017.

[17] Akshi Kumar, Shashwat Nayak, Navya Chandra, "Empirical analysis of supervised machine learning techniques for Cyberbullying detection," International Conference on Innovative Computing and Communications, Springer, Singapore, pp. 223-230, 2019.

[18] Vinita Nahar1, Sanad Al-Maskari1, Xue Li1 and Chaoyi Pang, "Semi-supervised learning for cyberbullying detection in social networks," Australasian Database Conference, Springer, Cham, pp. 160-171, 2014.

[19] Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed S. Akhtar, Manish Shrivastava, "A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection," In Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media, pp. 36-41. 2018.

[20] Souvick Ghosh, Satanu Ghosh, Dipankar Das, "Sentiment Identification in Code-Mixed Social Media Text," Cornel University, arXiv preprint arXiv:1707.01184, 2017.

[21] Santosh TY, Aravind KV, "Hate speech detection in hindi-english code-mixed social media text," Proceedings of the ACM India Joint International Conference on Data Science and Management, pp. 310-313, 3 January 2019.

[22] Sreelakshmi ka, Premjith Ba, Soman K.Pa, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," Procedia Computer Science, 171, pp.737-744., 2020

[23] Puja Chakraborty and Md Hanif Seddiqui., "Threat and abusive language detection on social media in bengali language," In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, pp. 1–6, 2019.

[24] Md Abdul Awal, Md Shamimur Rahman, and Jakaria Rabbi, "Detecting abusive comments in discussion threads using na¨ıve bayes," International Conference on Innovations in Science, Engineering and Technology (ICISET), IEEE, pp.163–167, 2018.

[25] Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam., "Hate speech detection in the bengali language: A dataset and its baseline evaluation," Proceedings of International Joint Conference on Advances in Computational Intelligence, Springer, Singapore pp 457-468, 2020.

[26] Md Gulzar Hussain, Tamim Al Mahmud, and Waheda Akthar., " An approach to detect abusive bangla text," International Conference on Innovation in Engineering and Technology (ICIET), IEEE, pp. 1–5, 2018. 31

[27] Shahnoor C Eshan and Mohammad S Hasan, "An application of machine learning to detect abusive bengali text," In 2017 20th International Conference of Computer and Information Technology (ICCIT), IEEE, pp.1–6, 2017.

[28] Estiak Ahmed Emon, Shihab Rahman, Joti Banarjee, Amit Kumar Das, and Tanni Mittra, "A deep learning approach to detect abusive Bengali text," 7th International Conference on Smart Computing & Communications (ICSCC), IEEE, pp. 1–5,2019.

[29] Salim Sazzed, "Abusive content detection in transliterated Bengali-English social media corpus." In Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, pp. 125-130. 2021.

[30] M. F. Mridha, Md. A. H. Wadud et al. "L-Boost: Identifying Offensive Texts From Social Media Post in Bengali." *Ieee Access*,9, pp: 164681-164699. December 2021

[31] Rui Zhao , Anna Zhou , Kezhi Mao, "Automatic detection of cyberbullying on social networks based on bullying features," International Conference of Distributed Computing and Networking, pp. 43 , 2016. 30

[32] Elif Varol Altay , Bilal Alataş, "Detection of Cyberbullying in Social Networks Using Machine Learning Methods," International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018.

[33] ZhangX, Tong J,VishwamitraN,Whittaker E, Mazer J-P,Kowalski R, Hu H, Luo F,Macbeth J, Dillon E., "Cyberbullying detection with a pronunciation based convolutional neural network," 15th IEEE international conference on machine learning and application, IEEE, 2016.

[34] A. Mangaonkar, A. Hayrapetian, and R. Raje, "Collaborative detection of cyberbullying behavior in Twitter data," Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT), Dekalb, IL, USA, , pp. 611-616, May 2015.

[35] Maral Dadvar, Dolf Trieschnigg, and Franciska de Jong, "Experts and Machines Against Bullies: A Hybrid Approach to Detect Cyberbullies," Canadian conference on artificial intelligence. Springer,Advances in Artificial Intelligence, pp 275-281, 2014.

[36] Hosseinmardi H, Mattson S-A, Rafiq R-I, Han R, Lv Q, Mishra S, "Detection of cyberbullying incidents on the instagram social network," Technical report by Association

for the advancement of artificial intelligence, pp 1–9, 2015.

[37] Statcounter Global Stats, https://gs.statcounter.com/social-media-stats/all/bangladesh, Accessed: 17 February 2022

[38] Vimala Balakrishnan, Shahzaib Khan, Hamid R. Arabnia, “Improving Cyberbullying Detection using Twitter Users’ Psychological Features and Machine Learning,” Computers and Security,90, pp. 101710,2020

[39] Rounak Ghosh, Siddhartha Nowal , Dr. G. Manju, “Social Media Cyberbullying Detection using Machine Learning in Bengali Language,” International Journal of Engineering Research & Technology (IJERT), 10, 2021.

[40] Niranjan B Subramanian. Converting Raw Text to Numerical Vectors using Bag of Words, N_Grams and TF-IDF. https://aiaspirant.com/bag-of-words/, (2nd September 2021)

[41] Rohit Dwivedi., “How Does K-nearest Neighbor Works In Machine Learning Classification Problem?, (online). https://www.analyticssteps.com/blogs/how-does-k-nearest-neighbor-works-machine-learning-classification-problem. Access: 5 August, 2021

[42] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, “Ef_cient kNN classifcation algorithm for big data,” Neurocomputing, 195, pp. 143-148, Jun. 2016.

[43] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, “Supervised machine learning for the detection of troll pro_les in Twitter social network: Application to a real case of cyberbullying," Proc. Int. Joint Conf. SOCO-CISIS-ICEUTE. Cham, Switzerland: Springer, pp. 419-428,2014.

[44] Selva Prabhakaran. How Naive Bayes Algorithm Works? (with example and full code). https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-a"nd-full-code/”, Access: 2 september, 2021

[45] H. Zhang, “The optimality of naive Bayes," Tech. Rep., 2004.

[46] Liza Wikarsa and Sherly Novianti Thahir. "A text mining application of emotion classifications of Twitter's users using Naive Bayes method." 2015 1st International Conference on Wireless and Telematics (ICWT). IEEE, 2015.

[47] Rushikesh Pupale. Support Vector Machines(SVM) — An Overview. https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989, Acceseds: 2 september, 2021

[48] Hani Nurrahmi, and Dade Nurjanah. "Indonesian twitter cyberbullying detection using text classification and user credibility." 2018 International Conference on Information and Communications Technology (ICOIACT). IEEE, 2018.

[49] Amgad Muneer, Suliman Mohamed Fati, “A comparative analysis of machine learning

techniques for cyberbullying detection on twitter," Future Internet, 12(11), pp. 187, 2020.

[50] Tony Yiu, Understanding Random Forest.
https://towardsdatascience.com/understanding-random-forest-58381e0602d2,
Accessed:7 August 2021.

[51] C.Van Hee, E. Lefever, B.Verhoeven, J. Mennes, B. Desmet, G. De Pauw, and V. Hoste, "Detection and fine-grained classification of cyberbullying events," International conference recent advances in natural language processing (RANLP). 2015.

[52] Avinash Navlani, AdaBoost Classifier in Python.
https://www.datacamp.com/community/tutorials/adaboost-classifier-pytho,
Accessed:7 August 2021.

[53] Saishruthi Swaminatha. Logistic Regression — Detailed Overview,
https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.
Accessed:7 August 2021.

[54] Pio Calderon, VADER Sentiment Analysis Explained
https://medium.com/@piocalderon/vader-sentiment-analysis-explainedf1c4f9101cd9
Accessed:5 March 2022.

[55] C. L. Nixon, "Current perspectives: the impact of cyberbullying on adolescent health," Adolescent health, medicine and therapeutics, vol. 5, p. 143, 2014.

[56] S. Mahmood , S. Islam " Bullying at High Schools: Scenario of Dhaka City," Journal of Advanced Laboratory Research in Biology, 8(4), pp.79-84,2017.

Appendix A

Code

1. Sentiment Analysis

Packages

```
import pandas as pd
import numpy as np
import re
import sklearn
import scipy as sp
import string
import matplotlib.pyplot as plt
import nltk
import emoji
import emojis
from string import punctuation
from collections import Counter
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.compose import ColumnTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

Data Processing

```
spcolumn=["process_comments","label","Bsentiment"]
corpus=pd.read_csv("bengali sentiment score.csv",usecols= spcolumn)
corpus['process_comments'] = corpus['process_comments'].astype(str)
print(corpus['label'].value_counts())
print (corpus['Bsentiment'].value_counts())


def data_preprocessing(comment):
    comment_lower= str(comment.lower())
    #emoji_count=emojis.count(comment_lower)
    url = re.compile(r'https?://\S+|www\.\S+') #remove web link from string python
    comment_extract_ursl = url.sub(r'', comment_lower)
    return comment_extract_ursl

def emoji_count(comment):
    emo=emojis.count(comment)
    return emo
```

```
def punctuation_number(comment):
    punctuation_counts = 0
    for i in comment:
        if i in string.punctuation:
            punctuation_counts=punctuation_counts+1
    return punctuation_counts


def remove_punc(comments):
    comment=str(comments)
    punctuations = '''()-[]{};:'",<>./?@#$%^&*_~\n\r'''
    no_punc = ""
    for char in comment:
        if char not in punctuations:
            no_punc = no_punc + char
    return no_punc


#spell correction

def dictionary():
    myfile = open ("bangla spelling correction dectionary.txt")
    d = { }
    for line in myfile:
        x = line.strip().split("\t")
        key, values = x[0], x[2:]
        d.setdefault(key, []).extend(values)
    return d

dictionary_tb= dictionary()
listOfItems = list(dictionary_tb.items())

#creat a list of the dictionary values
dictt_values=list(dictionary_tb.values())
listn=[]
for i in range (len(dictt_values)):
    for j in range (len(dictt_values[i])):
        listn.append(dictt_values[i][j])

def find_in_list(word):
    if word in listn:
        return True
    else:
        return False

def getKeysByValue(valueToFind):
    z=find_in_list(valueToFind)
    if z== True:
        for item  in listOfItems:
            for items in item:
                for i in range (len(items)):
```

```
                if items[i] == valueToFind:
                    return item[0]
    else:
        return valueToFind

def pass_comment(comments):
    comment= comments
    for word in comment:
        word=str(word)
        listOfKeys = str(getKeysByValue(word))
        for i in range(len(comment)):
            if comment[i]==word:
                comment[i]=listOfKeys
    return ' '.join(map(str, comment))      #map(function, iteration)
    #print (comment)


corpus['token1']=(corpus['not_punc_comments'].apply(lambda comment:
comment.strip().split(" ")))
corpus['spell_correct_with_emo'] = corpus['token1'].apply(lambda comment:
pass_comment(comment))


#abusive word count from spelling corrected line

#covert abusive word text file to list
with open('bengali positive word.txt','r', encoding="utf8") as f:
    listl=[]
    for line in f:
        strip_lines=line.strip()
        listli=strip_lines.split("\t")

        for i in range (len(listli)):
            m=listl.append(listli[i])

#count abusive word
def positive_word_number(comment):
    comment=comment.split(' ') #convert string to list
    save=[]
    for w in listl:
        for word in comment:
            if w==word:
                save.append(word)
    return len(save)

corpus['positive_word_number']=corpus['spell_correct_with_emo'].apply(lambda
comment: positive_word_number(comment))
print(corpus.positive_word_number)

# Translate to Bengali
def translateBengali(comment):
```

```
    translator= Translator()
    translated = (translator.translate(comment, src='bn', dest='bn').text)
    return translated

corpus["translatedBengali"]= corpus["spell_correct_without_emo"].apply(lambda
comment: translateBengali(comment))
print(corpus["translatedBengali"])
```

Sentiment Detection

```
#apply Vader to analyze sentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
sentiment= SentimentIntensityAnalyzer()
corpus['senti_score']= corpus["translatedBengali"].apply(lambda review:
                                        sentiment.polarity_scores(review))

corpus['compound_value']= corpus['senti_score'].apply(lambda score_dict:
score_dict['compound'])
print(corpus['compound_value'])
corpus['sentiment']= corpus['compound_value'].apply(lambda c: 'positive' if c>0 else
('negative' if c<0 else 'neutral'))
corpus.head()

#created new csv file that we will use for bullying and bengali sentiment detection
columns_name=['process_comments','compound_value',
                    'sentiment','translatedBengali','url_extracted',
                    'spell_correct_with_emo','spell_correct_without_emo','likes',
                    'Related_to_post','punc_number','emoji_number',
                      'abusive_word_number', 'positive_word_number',
                    'sentiment', 'Bsentiment','label']
corpus.to_csv('1vader sentiment score paper add.csv', index=False,
columns=columns_name

# call the data preprocessing function
corpus['url_extracted']= corpus["process_comments"].apply(lambda comment:
  data_preprocessing(comment))
corpus['emo_count']=corpus['url_extracted'].apply(lambda comment:
emoji_count(comment))
corpus['without_punc']=corpus['url_extracted'].apply(lambda comment:
remove_punc(comment))
corpus.head()

# create a new corpus
new_train_corpus= corpus[['process_comments',
      'without_punc','emo_count','Bsentiment']].  copy()
new_train_corpus['Bsentiment'] = new_train_corpus['Bsentiment'].astype(str)
new_train_corpus.head()


#Train and test data splitting
x=new_train_corpus.drop(['Bsentiment','process_comments'],axis='columns')
```

```
y=new_train_corpus.Bsentiment
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.25, random_state=3)


#Apply features extraction
transformer1 = ColumnTransformer([
                ('vectorizer', TfidfVectorizer(ngram_range=(1, 3)), 'without_punc')],
                remainder='passthrough')

vectorizer = TfidfVectorizer()
x_train_tf = transformer1.fit_transform(x_train)
x_test_tf=  transformer1.transform(x_test)

#confusion matrics
import seaborn as sns
def plot (y_test, y_predict):
   labels=unique_labels(y_test)
   colum=[f'Predicted  { label}' for label in labels]
   indc=[f'Actual  { label}' for label in labels]

   table=pd.DataFrame(metrics.confusion_matrix(y_test, y_predict),
         columns= colum, index=indc)
   sns.set(font_scale=1.4)
   return sns.heatmap(table, annot=True, fmt='d', cbar=True, cmap=plt.cm.GnBu)

#accuracy and classification report show
def accuracy_class_report(y_test, y_predict, file_name):
   accuracynb = (metrics.accuracy_score(y_test, y_predict))*100
   print("accuracy:   %0.3f%%" % accuracynb)

   report_dict = metrics.classification_report(y_test, y_predict,
   output_dict=True,zero_division=0)
   classification_report=pd.DataFrame(report_dict)
   classification_report.to_csv(file_name)
   print(metrics.classification_report(y_test, y_predict,zero_division=0))
   plot(y_test, y_predict)

#applying classificatin algorithm naive bayes
naive=MultinomialNB().fit(x_train_tf, y_train)
predictnaive= naive.predict(x_test_tf)
accuracynb = (metrics.accuracy_score(y_test, predictnaive))*100
output_dict=True,zero_division=0)
plot(y_test, predictnaive)
accuracy_class_report(y_test,predictnaive, 'naive report.xlsx')

#Apply liner SVM
svmmodel = SVC(kernel='linear', C=3) # for c=1,2,3,4 accuracy 64.009%  f1 score is same
svmmodel.fit(x_train_tf, y_train)
predictsvmlinear= svmmodel .predict(x_test_tf)
scoresvm = (metrics.accuracy_score(y_test, predictsvmlinear))*100
accuracy_class_report(y_test, predictsvmlinear, 'class svm linear.xlsx')
```

```
#Apply polynomial SVM
svmmodel = SVC(kernel='poly', degree=3, C=5) #degree 3= 25.216% , 4=23.70% 5=
24.784%
svmmodel.fit(x_train_tf, y_train)
predictsvmpoly = svmmodel .predict(x_test_tf)
scoresvm = (metrics.accuracy_score(y_test, predictsvmpoly))*100
accuracy_class_report(y_test, predictsvmpoly, 'class svm poly.xlsx')

# Apply RBF SVM
svmmodel = SVC(kernel='rbf', C=5) #degree 1=32.3283,  2= 53.664%, 3=59.052% ,
4=63.147%, c=5 64.224%
svmmodel.fit(x_train_tf, y_train)
predictsvmrbf = svmmodel .predict(x_test_tf)
scoresvm = (metrics.accuracy_score(y_test, predictsvmrbf))*100
accuracy_class_report(y_test, predictsvmrbf, 'class svm rbf.xlsx')

#Apply sigmoid SVM
svmmodel = SVC(kernel='sigmoid', C=5) #degree 1=32.3283,  2= 53.664%, 3=59.052% ,
4=63.147%, c=5 64.224%
svmmodel.fit(x_train_tf, y_train)
predictsvmsigmoid = svmmodel .predict(x_test_tf)
scoresvm = (metrics.accuracy_score(y_test, predictsvmsigmoid))*100
accuracy_class_report(y_test, predictsvmsigmoid, 'class svm sigmoid.xlsx')

#Apply KNN classifier
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train_tf, y_train)
predictknn=knn.predict(x_test_tf)
plot(y_test, predictknn)
accuracy_class_report(y_test, predictknn, 'knn report.xlsx')

#Apply RF classifier
rfclassifier = RandomForestClassifier(n_estimators = 100, random_state=20)
rfclassifier.fit(x_train_tf, y_train)
predictrf=rfclassifier.predict(x_test_tf)
accuracy_class_report(y_test,predictrf,'rf report.xlsx')
plot(y_test, predictrf)

rfclassifier = RandomForestClassifier(n_estimators = 170,random_state=20)
rfclassifier.fit(x_train_tf, y_train)
predictrf170=rfclassifier.predict(x_test_tf)
accuracy_class_report(y_test,predictrf170,'rf report_170.xlsx')
plot(y_test, predictrf170)

#Apply LR classifier
logicregression= LogisticRegression(multi_class='multinomial',solver='lbfgs',
                          C=1.0, max_iter=100) #Make an instance of the Model
logicregression.fit(x_train_tf, y_train)
predictlr=logicregression.predict(x_test_tf)
accuracy_class_report(y_test, predictlr, 'lr report.xlsx')
```

```
#Apply AdaBoost classifier
adaboost= AdaBoostClassifier(n_estimators=80, learning_rate=1.0, random_state=1)
adaboost.fit(x_train_tf, y_train)
predictada80=adaboost.predict(x_test_tf)
plot(y_test, predictada80)
accuracy_class_report(y_test, predictada80, 'adaboost report_80.xlsx')
#print (predictada)

#Store data for next phase
BengaliDataframe = pd.DataFrame({'comment':x_test1.process_comments,'Actual': y_test,
                'Predict_nb': predictnaive,
                 'Predict_svm_linear':predictsvmlinear,'predict_svm_rbf':predictsvmrbf,
                 'Predict_knn':predictknn,
                 'Predict_rf_100':predictrf, 'Predict_rf_170':predictrf170,
                'Predict_lr':predictlr, 'predict_ad80':predictada80,'Predict_ab':predictada})
BengaliDataframe.to_csv('Bengali_sentiment_prediction.csv')
```

2. Bullying Detection

```
#call the dataset used in the previous section
columns_name=['spell_correct_without_emo','likes',
            'Related_to_post','punc_number','emoji_number',
            'abusive_word_number', 'positive_word_number',
            'sentiment', 'Bsentiment','label']
corpus=pd.read_csv('vader sentiment score.csv',usecols=columns_name)
corpus['spell_correct_without_emo'] = corpus['spell_correct_without_emo'].astype(str)

#Apply Dummy variable
Bsentiment_dummies=pd.get_dummies(corpus.Bsentiment)
Bsentiment_dummies
merged_corpus= pd.concat([corpus,Bsentiment_dummies],axis='columns')
merged_corpus
final_corpus= merged_corpus.drop(['Bsentiment','supportive'],axis='columns')

#Train test splitting
x=final_corpus.drop(['label', 'sentiment'], axis='columns')
y=final_corpus.label
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.25, random_state=0)

#Apply TF-IDF
from sklearn.compose import ColumnTransformer
transformer = ColumnTransformer([
                    ('vectorizer', TfidfVectorizer(ngram_range=(1, 1)),
                    'spell_correct_without_emo')], remainder='passthrough')

vectorizer = TfidfVectorizer()
x_train_tf = transformer.fit_transform(x_train)
x_test_tf=  transformer.transform(x_test)
```

```
#applying classificatin algorithm naive bayes
naive=MultinomialNB().fit(x_train_tf, y_train)
predictnaive= naive.predict(x_test_tf)
accuracynb = (metrics.accuracy_score(y_test, predictnaive))*100
plot(y_test, predictnaive)
accuracy_class_report(y_test,predictnaive, 'naive report bullying.xlsx')

#applying classificatin algorithm naive bayes
from sklearn.naive_bayes import MultinomialNB
naive=MultinomialNB().fit(x_train_tf, y_train)
predictnaivetrain= naive.predict(x_train_tf)
#calculating acuracy and illustrate testing result
accuracynb = (metrics.accuracy_score(y_train, predictnaivetrain))*100
output_dict=True,zero_division=0)
plot(y_train, predictnaivetrain)
accuracy_class_report(y_train,predictnaivetrain, '1 naive report bullying train.xlsx')

#SVM for testing data
svmmodel = SVC(kernel='linear', degree=1, C=4)
svmmodel.fit(x_train_tf, y_train)
predictsvm = svmmodel .predict(x_test_tf)
scoresvm = (metrics.accuracy_score(y_test, predictsvm))*100
accuracy_class_report(y_test, predictsvm, 'class svm bull.xlsx')

#SVM for training data
svmmodel = SVC(kernel='linear', degree=1, C=6)
svmmodel.fit(x_train_tf, y_train)
predictsvmtrain = svmmodel .predict(x_train_tf)
scoresvm = (metrics.accuracy_score(y_train, predictsvmtrain))*100
accuracy_class_report(y_train, predictsvmtrain, '2 class svm bull train.xlsx')

#Apply KNN for bullying detection
knn = KNeighborsClassifier(n_neighbors=4, p=1)
knn.fit(x_train_tf, y_train)
predictknn=knn.predict(x_test_tf)
plot(y_test, predictknn)
accuracy_class_report(y_test, predictknn, 'knn report bull.xlsx')

#KNN for trining data
knn = KNeighborsClassifier(n_neighbors=4, p=1)
knn.fit(x_train_tf, y_train)
predictknntrain=knn.predict(x_train_tf)
plot(y_train, predictknntrain)
accuracy_class_report(y_train, predictknntrain, '3 knn report bulltrain.xlsx')

#Apply LR
logicregression= LogisticRegression(multi_class='multinomial', solver='lbfgs',
max_iter=400, random_state=2) #Make an instance of the Model
logicregression.fit(x_train_tf, y_train)
predictlr=logicregression.predict(x_test_tf)
accuracy_class_report(y_test, predictlr, 'lr report bull.xlsx')
```

```
predictlrtrain=logicregression.predict(x_train_tf)
accuracy_class_report(y_train, predictlrtrain, '1 lr report bull train.xlsx')

#Apply AdaBoost
adaboost= AdaBoostClassifier(n_estimators=100, learning_rate=1.0, random_state=4)
adaboost.fit(x_train_tf, y_train)
predictada=adaboost.predict(x_test_tf)
plot(y_test, predictada)
accuracy_class_report(y_test, predictada, 'adaboost report bull.xlsx')
predictadatrain=adaboost.predict(x_train_tf)
plot(y_train, predictadatrain)
accuracy_class_report(y_train, predictadatrain, '6 adaboost report bull train.xlsx')
#print (predictada)

#Apply RF
rfclassifier = RandomForestClassifier(n_estimators = 170, random_state=3)
model=rfclassifier.fit(x_train_tf, y_train)
predictrf=rfclassifier.predict(x_test_tf)
accuracy_class_report(y_test,predictrf,'rf report bull.xlsx')
plot(y_test, predictrf)
predictrftrain=rfclassifier.predict(x_train_tf)
accuracy_class_report(y_train,predictrftrain,'4 rf report bull train .xlsx')
plot(y_train, predictrftrain)


#from yellowbrick.classifier import ROCAUC
from yellowbrick.classifier import ROCAUC
visualizer = ROCAUC(model, classes=["Bullying", "Neutral", "NonBullying"])
visualizer.fit(x_train_tf, y_train)        # Fit the training data to the visualizer
visualizer.score(x_test_tf, y_test)        # Evaluate the model on the test data
visualizer.show()

visualizer = ROCAUC(svmmodel, classes=["Bullying", "Neutral", "NonBullying"])
visualizer.fit(x_train_tf, y_train)
visualizer.score(x_test_tf, y_test)
visualizer.show()
```