**Thesis On**

**Abusive Bangla comments detection on public Facebook Pages using Machine Learning Algorithm**

**Submitted by**

**Asadullah Al Noman**

Roll: ASH1811048M

Session: 2017-2018

**Thesis Supervisor**

**Tanvir Zaman Khan**

Assistant Professor

Department of Information & Communication Engineering



Noakhali Science & Technology University

Noakhali – 3814

Bangladesh

**Date: 5 March 2023**

# 

# DECLARATION

This thesis report is submitted to the Department of Information & Communication Engineering, Noakhali Science & Technology University, Sonapur, Noakhali-3814, in partial fulfilment of the requirements for having a B.Sc degree in Information & Communication Engineering. So, I hereby declare that this thesis report has not been submitted elsewhere for the requirement of any kind of degree, diploma, or publication.

|  |
| --- |
|  |

**Asadullah Al Noman**

Student of Information & Communication Engineering

Noakhali Science & Technology University

Roll No: ASH1811048M

Session: 2017-18

# ACCEPTANCE

This thesis report is submitted to the Department of Information & Communication Engineering(ICE), Noakhali Science & Technology University, Sonapur, Noakhali-3814, in partial fulfilment of the requirements for having a B.Sc degree in Information & Communication Engineering.

**Tanvir Zaman Khan**

Assistant Professor

Department of Information & Communication Engineering

Noakhali Science & Technology University

Noakhali – 3814, Bangladesh

# ACKNOWLEDGEMENTS

First, I would like to express my gratitude to Almighty Allah who has given me the strength and golden opportunity to do this thesis work –" **Abusive Bangla comments detection in public Facebook pages using Machine Learning algorithm.**” I would like to thank my thesis supervisor, Tanvir Zaman Khan Sir for their unwavering support, guidance, and patience throughout this research work. Their insightful feedback and constructive criticism have been invaluable in shaping the content and direction of this thesis.

I wish to thank and convey my respect to all of my honourable teachers at the Department of Information and Communication Engineering, Noakhali Science and Technology University for their valuable suggestions and academic help during the period of consultancies concerning the research.

Asadullah Al Noman

# ABSTRACT

In the era of social networking platforms, user-generated content is overflowing every second on online social media platforms like Facebook. So observing and identifying many contents, including threats and sexual harassment, are more accessible than traditional media. Online content with extreme harmfulness can lead to online harassment, profanity, personal attacks, and bullying acts. As Bangla is the seventh most spoken language worldwide, the utilization of the Bangla language on Facebook has raised current times. The usage of abusive comments on Facebook with Bangla also has increased terrifyingly. To classify abusive comments swiftly and precisely. Therefore, we proposed a model to detect abusive comments using Machine Learning. We used several machine Learning algorithms e.g. SVC= Support Vector Classifier, MNB= Multinomial Naive Bayes, LR= Logistic Regression, RF=Random Forest Classifier, DT= Decision Tree Classifier, KNN= KneighborsClassifier, and SGC= Stochastic Gradient Descent Classifier have been tested to detect multi-type abusive Bangla comments. The performances of the algorithms were evaluated utilizing precision, recall, accuracy, and F1-score. The result shows that the SGC model gives the best accuracy for our dataset. Bangla abusive comments detection can help to prevent online harassment, which is on the rise.

CONTENTS

[DECLARATION ii](#_Toc128627317)

[ACCEPTANCE iii](#_Toc128627318)

[ACKNOWLEDGEMENTS iv](#_Toc128627319)

[ABSTRACT v](#_Toc128627320)

[LIST OF FIGURES viii](#_Toc128627321)

[LIST OF TABLES x](#_Toc128627322)

[CHAPTER I 1](#_Toc128627323)

[INTRODUCTION 1](#_Toc128627324)

[1.1 Background Information 1](#_Toc128627325)

[1.2 Problem Statement 2](#_Toc128627326)

[1.3 Research Objectives 2](#_Toc128627327)

[CHAPTER II 3](#_Toc128627328)

[LITERATURE REVIEW 3](#_Toc128627329)

[2.1 Introduction 3](#_Toc128627330)

[2.2 Existing Research Works 4](#_Toc128627331)

[CHAPTER III 6](#_Toc128627332)

[DESIGN AND METHODOLOGY 6](#_Toc128627333)

[3.1 Workflow 6](#_Toc128627334)

[3.2 Steps in method 8](#_Toc128627335)

[3.2.1 Raw data collection 9](#_Toc128627336)

[3.2.2 Data Preprocessing 10](#_Toc128627337)

[3.2.4 Vectorization 13](#_Toc128627338)

[3.2.5 Dataset split: 14](#_Toc128627339)

[3.2.6 Building Models 15](#_Toc128627340)

[3.3 Performance Analysis 19](#_Toc128627341)

[3.4 Conclusion 21](#_Toc128627342)

[CHAPTER IV 22](#_Toc128627343)

[RESULT ANALYSIS AND DISCUSSION 22](#_Toc128627344)

[4.2 Results Analysis 22](#_Toc128627345)

[4.2.2 Performance of Machine Learning Algorithms 22](#_Toc128627346)

[4.2.3 Machine Learning Algorithms Comparison 29](#_Toc128627347)

[4.3 Single Comment Check: 32](#_Toc128627348)

[4.4 Discussion 33](#_Toc128627349)

[CHAPTER V 34](#_Toc128627350)

[CONCLUSION AND FUTURE WORK 34](#_Toc128627351)

[5.1 Conclusion 34](#_Toc128627352)

[5.2 Future Work 34](#_Toc128627353)

[References 35](#_Toc128627354)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Title** | **Page No** |
| 3.1 | [Workflow diagram](#fig3_1) | 6 |
| 3.2 | [Steps in method](#fig3_2) | 6 |
| 3.3 | [Comments count by class\_label](#fig3_3) | 8 |
| 3.4 | [Some unprocessed dataset](#fig3_4) | 8 |
| 3.5 | [Steps in preprocessing](#fig3_5) | 9 |
| 3.6 | [Before punctuation removing](#fig3_6) | 10 |
| 3.7 | [After punctuation removing](#fig3_7) | 10 |
| 3.8 | [After Stop word removing](#fig3_8) | 11 |
| 3.9 | [After stemming](#fig3_9) | 12 |
| 3.10 | [Data count after preprocessing](#fig3_10) | 12 |
| 3.11 | [Train and Test data split](#fig3_11) | 13 |
| 3.12 | [Possible Hper lines in SVC](#fig3_12) | 14 |
| 3.13 | [Logistic Regression function](#fig3_13) | 15 |
| 3.14 | [Decision tree visualization](#fig3_14) | 16 |
| 3.15 | [Random Forest Visualization](#fig3_15) | 17 |
| 3.16 | [KNeighborsClassifier Visualization](#fig3_16) | 18 |
| 3.17 | [Stochastic Gradient Descent Classifier Visualization](#fig3_17) | 19 |
| 3.18 | [Confusion matrix](#fig3_18) | 20 |
| 4.1 | [Classification report of Support Vector Classifier](#fig4_1) | 22 |
| 4.2 | [Confusion matrix ofSupport Vector Classifier](#fig4_2) | 23 |
| 4.3 | [Classification report of Multinomial Naïve Bayes](#fig4_3) | 23 |

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page No** |
| 4.4 | [Confusion matrix of Multinomial Naïve Bayes](#fig4_3) | 24 |
| 4.5 | [Classification report of Logistic Regression](#fig4_4) | 24 |
| 4.6 | [Confusion matrix of Logistic Regression](#fig4_5) | 25 |
| 4.7 | [Classification report of Random Forest Classifier](#fig4_6) | 25 |
| 4.8 | [Confusion matrix of Random Forest Classifier](#fig4_7) | 26 |
| 4.9 | [Classification report of Decision Tree Classifier](#fig4_8) | 26 |
| 4.10 | [Confusion matrix of Decision Tree Classifier](#fig4_9) | 27 |
| 4.11 | [Classification Report of KneighborsClassifier](#fig4_10) | 27 |
| 4.12 | [Confusion matrix of KneighborsClassifier](#fig4_11) | 28 |
| 4.13 | [Classification Report of Stochastic Gradient Descent Classifier](#fig4_12) | 28 |
| 4.14 | [Confusion matrix of Stochastic Gradient Descent Classifier](#fig4_13) | 29 |
| 4.15 | [Performance Measure comaprisons of Algorithms(TF-IDF)](#fig4_14) | 31 |
| 4.16 | [Two positive comments](#fig4_15) | 32 |
| 4.17 | [Two Negative comments](#fig4_16) | 32 |

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No** |
| 2.2 | [Existing Research Works](#table2_1) | 4 |
| 4.2.3.1 | [Comparison between Precision](#table2_3) | 29 |
| 4.2.3.2 | [Comparison between Recall](#table2_3) | 30 |
| 4.2.3.3 | [Comparison between F1-Score](#table2_4) | 30 |
| 4.2.3.4 | [Comparison between Accuracy](#table2_5) | 30 |

# LIST OF TABLES

# CHAPTER I

# INTRODUCTION

## 1.1 Background Information

Social Networking Sites (SNS) such as Facebook, Twitter, etc., have facilitated modern communications with a handful of services. On average with 2.27 billion 1 active users, Facebook is the most preferable medium of interaction with people. People talk, argue, and express opinions with people or communities by enjoying the different features of Facebook [1]. However, arguments or debates often turn towards intolerance of others' opinions, which often leads to abuse or hate speech towards a person, community, culture, or religion. Dhaka, the capital of Bangladesh, has the second-highest number of active Facebook users which is about 1.1% of total monthly active SNS users all around the world. People engage in different groups and often create pages for promoting products and ideas. For example, the Facebook page of Sakib Al Hasan, 3 one of the topmost celebrity players has about 1 million likes and followers [2]. Besides, actors, players, and political, cultural, or social parties maintain Facebook pages for different purposes. Different studies indicate that these pages are potential sources for spreading hateful speech on political, religious, and cultural aspects. According to the Counter-Terrorism and Transnational Crime (CTTC) cyber unit of Bangladesh, about 2,500 Facebook pages are responsible for spreading communal hatred. Moreover, Facebook pages are public and anyone can see the posts. Therefore, public pages have become honeypots to spread hateful speech and create unfavorable situations [3].

Hate speech or abusive comments can be defined as the willing and frequent harm imposed through electronic media. It has become a challenge to stop abusive comments on social platforms. One possible solution can be detecting the texts containing abusive words and preventing them from being displayed [4].

With the rise of Machine Learning algorithms and Natural Language processing techniques, we now have a perfect chance to deploy them to detect abusive comments. Some of the tasks that the Machine Learning algorithms being capable of are manifested below.

## 1.2 Problem Statement

The uses of social media are widespread in recent years. The general public expresses their opinion via comments about national and international Facebook pages like politics, celebrities, culture, scholars,s, etc. These responses from the public play a vital role in how things may change in the future. For this reason, it becomes an emerging problem to know which one is an abusive comment and which one is a non-abusive comment. if the situation is misled it can turn into a huge national problem like a civil war. we build a model to solve this problem to show a way to detect abusive comments on Facebook pages. By using machine learning algorithms we will detect these abusive and non-abusive comments.

## 1.3 Research Objectives

The focus is to find the best performance by providing an algorithm for detecting abusive and non-abusive Bangla comments. Hence the main objectives of this research are :

i. To detect abusive and non-abusive Bangla comments through the use of various machine learning algorithms.

ii. To provide the comparative performance among distinct machine learning algorithms.

iii. To investigate the most effective algorithm for detecting abusive comments on precision, recall, and accuracy values.

iv. To improve the existing system in terms of accuracy and other performance parameters.

# CHAPTER II

# LITERATURE REVIEW

## 2.1 Introduction

Sentiment analysis has become one of the most popular research topics currently and it is used for identifying positive and negative opinions and emotions like happy, sad, and neutral. Sentiment Analysis is the computational investigation of human opinions, attitudes, and feelings [5]. 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 detects abusive comments in English. There have only been a few studies on abusive Bangla comment detection. This literature review examines the abusive Bangla comments detection and prevention techniques for the Bangla language.

Much work has been completed on detecting hate speech using deep learning models [7], [10]. There have also been efforts to grow the accuracy of predicting hate speech specifically by extracting unique semantic features of hate speech comments [6]. Researchers have also utilized fastText to build models that can be trained on billions of words in less than ten minutes and classify millions of sentences among hundreds of classes [14].

There is also numerous research, based on Convolution Neural Networks (CNN), Deep Learning to detect hate speech on social media. Table 1 summarizes existing research on the detection of abusive text on social media platforms such as Facebook and Twitter.

## 2.2 Existing Research Works

|  |  |  |  |
| --- | --- | --- | --- |
| **Index No** | **Author** | **How it is used (technique)** | **Result Analysis** |
| 1 | Lucky et al.[3] | Naive Bayes, Random Forest, Decision Tree, K-Nearest Neighbor, SVM, Logistic Regression, RNN. | MNB gives the highest accuracy among machine learning algorithms. |
| 2 | Tanvir et al.[8] | MNB, MLP, SVM, decision tree, random forest, stochastic gradient descent (SGD), ridge, perceptron, and k-nearest neighbors (k-NN). | SVM with the full dataset obtained the highest accuracy of 88%. |
| 3 | Remon et al.[7] | SVM, Random Forest (RF), K- Nearest Neighbour (KNN), CNN, MLP, BNB, Logistic Regression (LR). | The SVM has fared the best. MLP has the best accuracy in deep learning, while CNN has also done brilliantly. |
| 4 | Saiful et al.[1] | Support Vector Machine (SVM), Convolutional Neural Networks (CNN), Bi-directional Long Short Term Memory (Bi-LSTM). | Existing word embedding models trained with informal texts perform better than those trained with formal text. |
| 5 | Amit et al.[17] | Linear Support Vector Classifier (LinearSVC), Logistic Regression, MNB, Random Forest (RF), ANN, RNN with LSTM. | RNN with LSTM cell performs best. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Index No** | **Author and Year** | **How it is used (technique)** | **Result Analysis** |
| 7 | William et al.[19] | Naive Bayes, Random Forest, Decision Tree, SVM, XGBoost. | MNB gives the highest accuracy among machine learning algorithms. |
| 8 | Ahmed et al.[12] | MNB, MLP, SVM, decision tree, random forest. | MNB with the full dataset obtained the highest accuracy of 84%. |
| 9 | Shafayet et al.[13] | MNB, MLP, SVM, XGBoost,  Logistic regression. | The SVM has fared the best. MNB with the full dataset obtained the highest accuracy of 88%. |
| 10 | Adil et al.[10] | Random Forest, MNB, SVM,  Logistic regression. | SVM gives the highest accuracy with 91%. |
| 11 | Ahmed et al.[6] | Support Vector Machine (SVM), NB. | NB gave the best accuracy with 94%. |
| 12 | Celestine et al.[9] | LSTM,BLSTM,RNN,GRU. | RNN with LSTM cell performs best. |
| 13 | Rahul et al.[5] | Support Vector Machine (SVM), NB. | SVM was the best with 71.1% accuracy. |
| 14 | Romim et al.[14] | SVC, Linear SVC, NB, RF,  Adaboost, GRU. | GRU achieved the best result with 70.10% accuracy. |
| 15 | Shahnoor et al.[15] | Random Forest, MNB, SVM. | SVM achieved the best result with accuracy. |

# CHAPTER III

# DESIGN AND METHODOLOGY

## 3.1 Workflow

Here is a workflow for Abusive Bangla comments detection on public Facebook Pages using Machine Learning Algorithm.

1. **Data collection:** Collect data from public Facebook pages with Bangla comments. This data will be used to train and test the machine-learning algorithms.
2. **Data preprocessing:** Preprocess the data by cleaning, tokenizing, and transforming the Bangla text into a numerical format that can be used for training the machine learning algorithms.
3. **Feature engineering:** Extract relevant features from the preprocessed Bangla text data, such as n-grams, word embeddings, or other linguistic features.
4. **Data splitting:** Split the preprocessed data into training and testing sets.
5. **Model building:** Select multiple machine learning algorithms for the task of classifying Bangla comments as abusive or non-abusive.
6. **Model training:** Train each selected machine learning algorithm on the preprocessed data and extracted features to learn to classify Bangla comments as abusive or non-abusive.
7. **Model evaluation:** Evaluate the performance of the trained model using various performance matrices, such as accuracy, precision, recall, F1 score, and confusion matrix.
8. **Model comparison:** Compare the performance of each trained model using various performance matrices, such as accuracy, precision, recall, F1 score, and confusion matrix.
9. **Comment classification:** Use the best-performing model to classify Bangla comments on public Facebook pages as abusive or non-abusive.

The proposed methodology is shown in Fig, 3.1

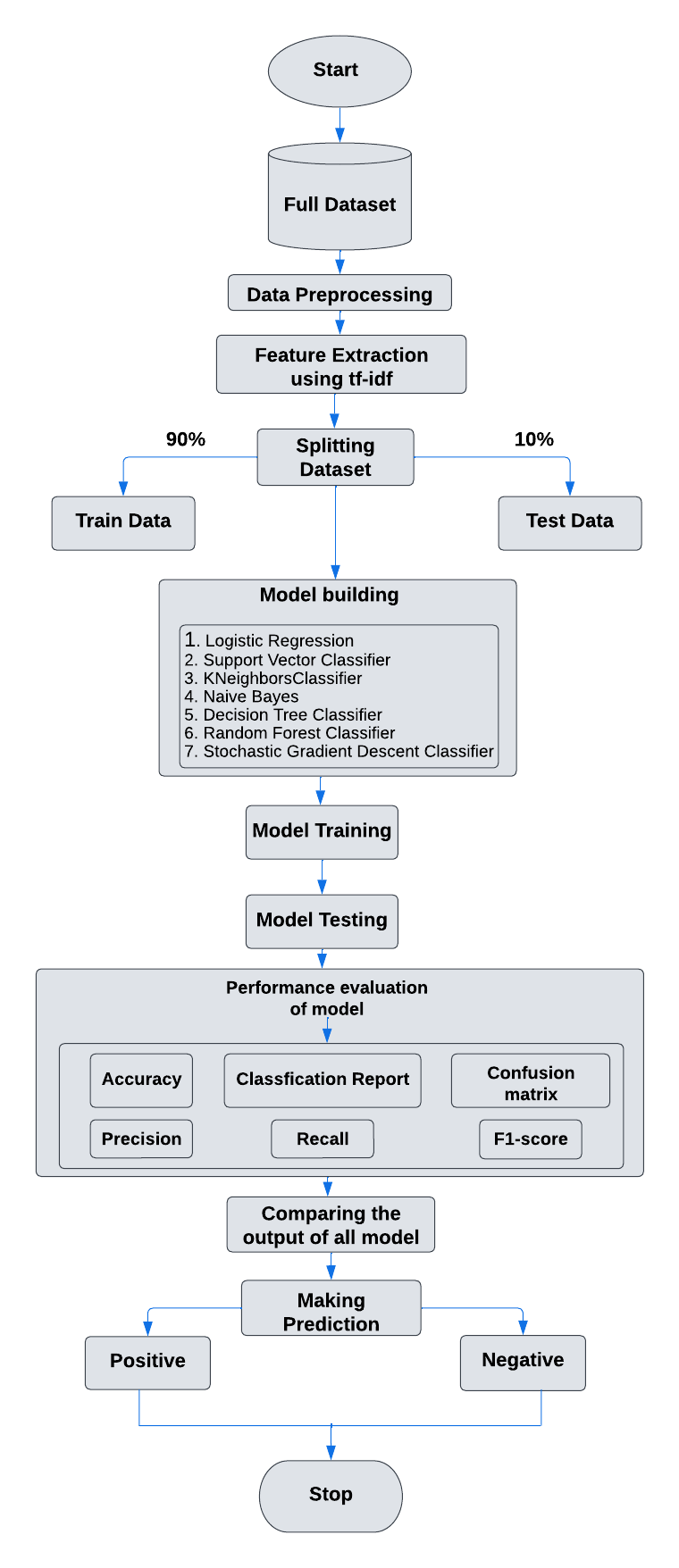


Fig 3.1: Workflow diagram

## 3.2 Steps in method

Our method can be shown in 4 parts.

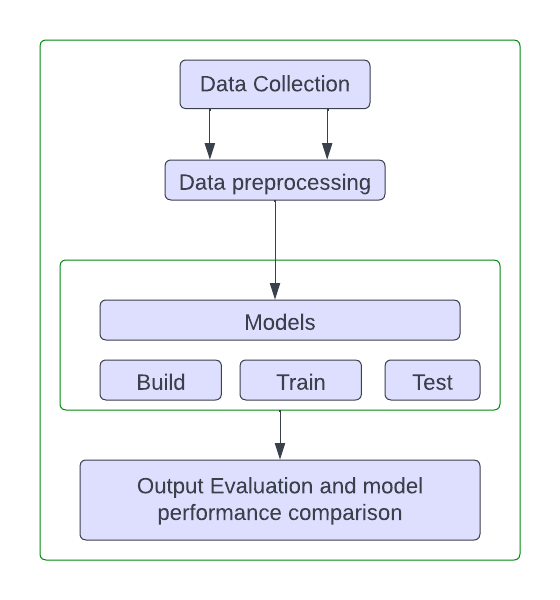
1. Raw data collection
2. Data preprocessing
3. Models building, Training, and Test
4. Performance Measures

Fig 3.2: Steps in the method

### 3.2.1 Raw data collection

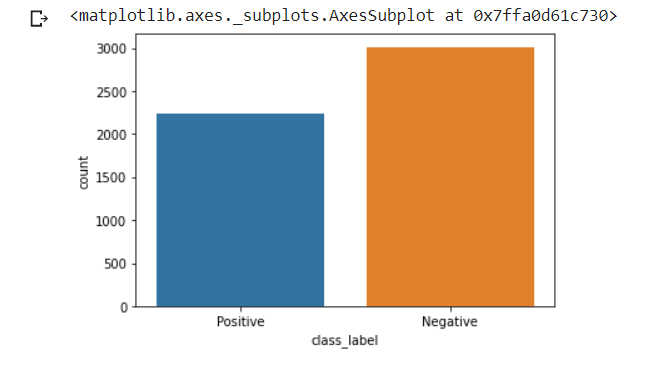
The first step of this research work is dataset creation. There is some Bengali hate space, and the abusive comment detection public dataset is available online. For the research purpose, we collected a total of 5252 comments from GitHub [24], which consisted of 3 columns id, text, and class\_label. It is an open-source platform for downloading any dataset. In Our initial stage, our non-abusive data are labelled as class\_label positive, and abusive data are labelled as class\_label negative. The rest of the project works on that. The count of different sentiment label comments is given below by a bar plot

Fig 3.3: Comments count by class\_label

Our dataset name is “comments". Here we show some unprocessed comments.

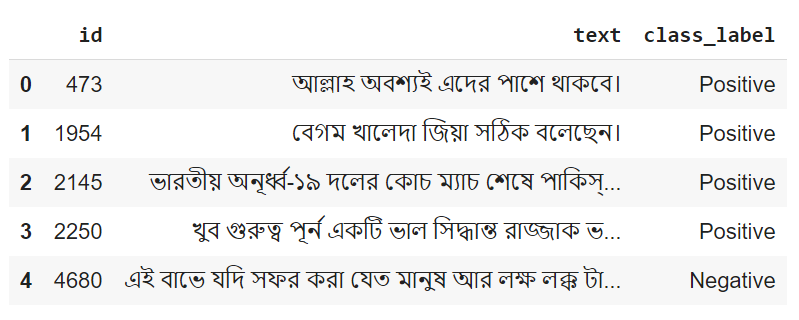


Fig 3.4: Some unprocessed dataset

### 3.2.2 Data Preprocessing

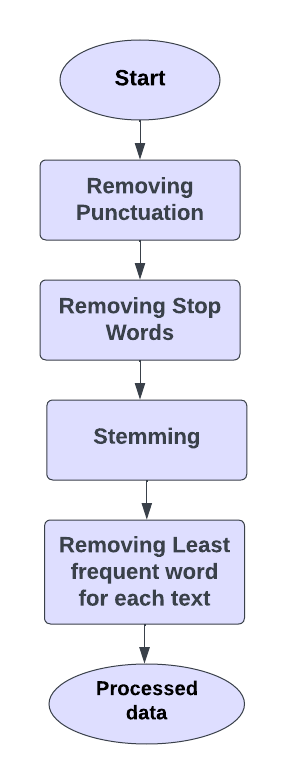
For data preprocessing we need to remove all the unnecessary words, commas, stop words, and other symbols. We can do these processes with the following steps

Fig 3.5 Steps in preprocessing

**Removing Punctuation:** Our First task is to remove stop words and punctuation marks from the text content. Punctuation marks, on the other hand, are symbols that are used to indicate the structure and meaning of sentences but are not typically analyzed as part of the text data [4]. Examples of Bangla punctuation marks include "।", "একটি বিন্দু", and "বাংলা এমনকি".

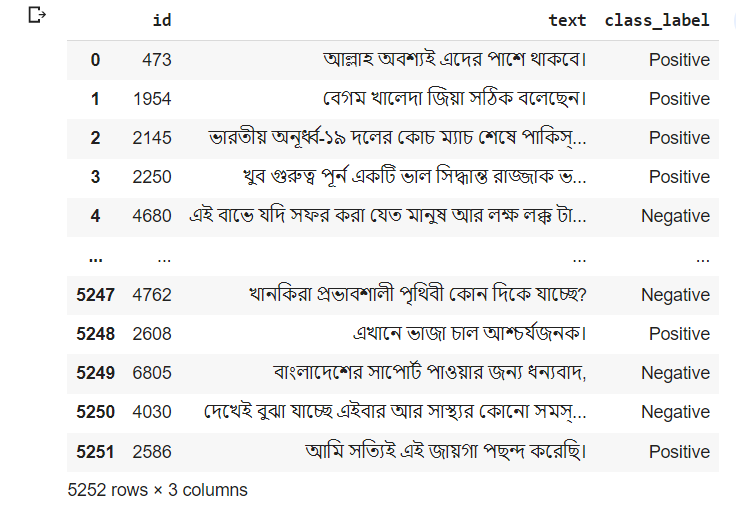


Fig 3.6: Before punctuation removing

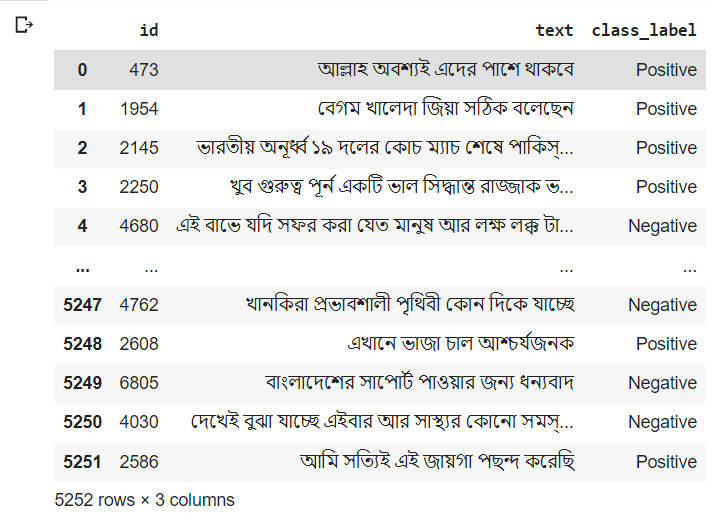


Fig 3.7: After punctuation removing

**Removing Stop Words:** Stop words are words that occur frequently in a language but do not carry any significant meaning in the context of text analysis, such as articles, conjunctions, and prepositions [7]. Examples of Bangla stop words include "এবং", "একটি", and "যেমন".

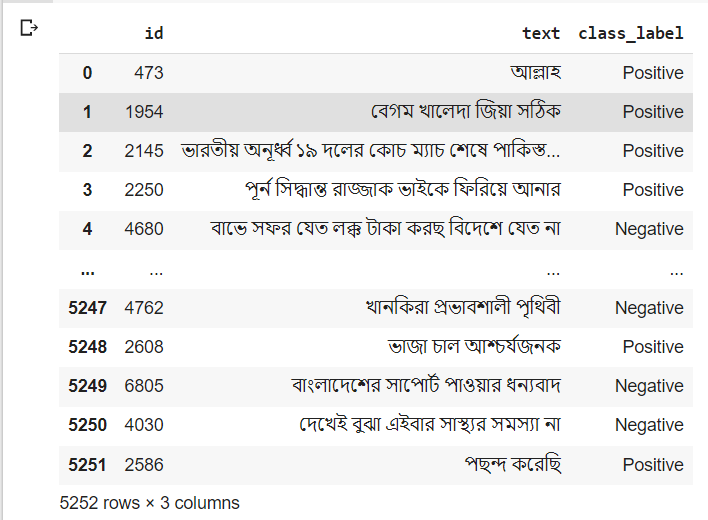
****

Fig 3.8: After Stop word removing

**Removing the least frequent words for each class\_label:** Some of the words doesn't contribute much to training the models. For this purpose, we first separate the datasets based on their class label. Then we count the frequency of every word. Then remove the words which have the least frequency, frequency=1.

**Stemming:** Bangla stemming is the process of reducing Bangla words to their base or root form, which is also known as the lemma. This process involves removing the affixes (prefixes and suffixes) from the words so that variations of the same word can be grouped as a single term. For this purpose, we use the "Bangla-stemmer" library to stem the Bangla words [11].

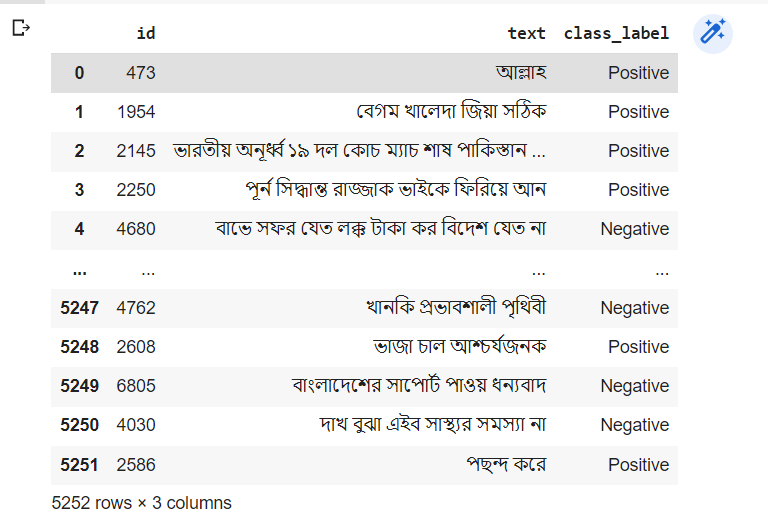
****

Fig 3.9: After stemming

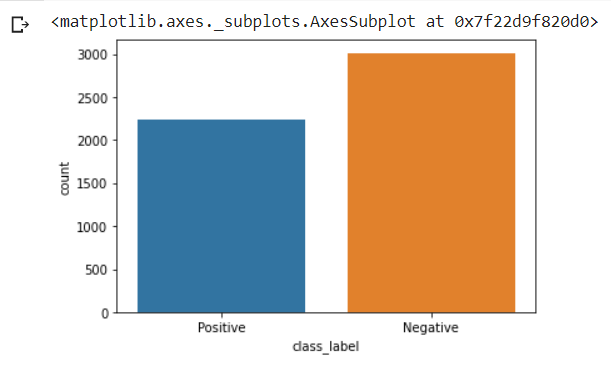
After completing all the preprocessing steps we have 5221 comments available. In the below bar plot, we can see the actual dataset.

Fig 3.10: Data count after preprocessing

### 3.2.4 Vectorization

The computer doesn't understand any human language. For predicting or detecting any content to its class or label, we have to train them in their understandable language. Which is nothing but 0 and 1.

**Vectorization:** It is the process of converting text data into numerical vectors that can be used as input to machine learning algorithms. The goal of vectorization is to represent text data in a way that can be easily processed by machine learning models, which typically require numerical input data.

In this research, we used only one vectorizer TF-IDF. These are briefly discussed below.

**TF-IDF Vectorizer:**

Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical measure of the importance of a word in a document or corpus. It is calculated by taking two factors into account: how frequently a word appears in a document (TF) and how uncommon the word is in the corpus (IDF) [16].

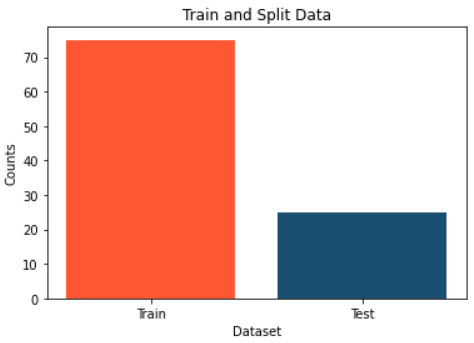
The frequency of a word in a document is measured by TF, while the frequency of a word in the entire corpus is measured by IDF. The TF-IDF score is the product of these two factors, and it gives more weight to words that are frequent in a document but rare across the corpus [17].

TF-IDF can be calculated by multiplication of TF with IDF,

= \*

**N-Gram:** It specifies how many continuous words or token is countable for a specific feature extraction algorithm like TF-IDF vectorizer. We use the N-gram range (1-3) for the vectorizer. An example of the N-gram word is given below.

3.2.5 Dataset split: For performing the prediction we need to split the whole data into two different sets.

**Train set:** For training the models. We consider 90% of the data to be in the train.

**Test set:** For testing the models. We consider 10% of the data to be a test set.

Fig 3.11: Train and Test data split

### 3.2.6 Building Models

We implemented several Machine Learning algorithms to test and detect the specific class of a single comment.

We total built seven Machine Learning models. There are:

1. Support Vector Classifier
2. Multinomial Naive Bayes
3. Logistic Regression
4. Random Forest Classifier
5. Decision Tree Classifier
6. KNeighborsClassifier
7. Stochastic Gradient Descent Classifier

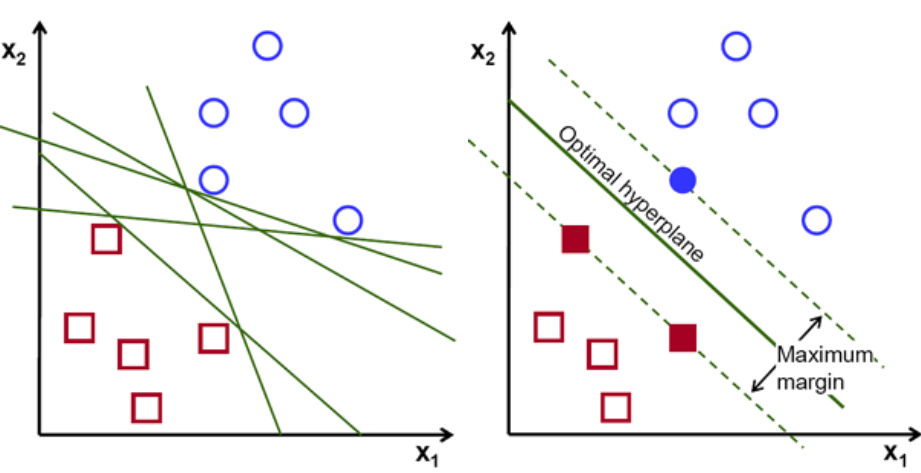
**Support vector Classifier:** A Support Vector Classifier, or SVC, is a popular Supervised Learning algorithm that is used for both classification and regression problems. The SVM algorithm's goal is to find the best line or decision boundary for categorizing n-dimensional space so that we can easily place new data points in the correct category in the future. A hyperplane is the best decision boundary [20].

Fig 3.12: Possible Hper lines in SVC[20]

**Multinomial Naive Bayes Classifier:** Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. The probability of P(c|x) where c is the class of the possible outcomes and x is the given instance is calculated using equation (1) [7].

P(c|x) = P(x|c) \* P(c) / P(x) ---------------------(1)

**Logistic Regression**

After using a transformation function, linear regression predictions are continuous values (rainfall in cm), while logistic regression predictions are discrete values (i.e, whether a student passed/failed). The logistic function h(x)=1/(1+ex) is the name of the transformation function used in logistic regression. This results in an S-curve [9].

The result of logistic regression is a set of probabilities for the default class. Because it's a probability, the result is between 0 and 1.

The y-value is generated by using the logistic function h(x)=1/(1+e-x) to log transform the x-value. The probability is then forced into a binary categorization using a threshold. The logistic function is depicted in the graph below:

Fig 3.13:Logistic Regression function[25]

**Decision Tree Classifier**

A machine learning algorithm used for classification tasks is Decision Tree Classifier. It works by building a tree-like model of decisions and their possible outcomes. The algorithm takes a dataset and divides it into subsets based on the values of input features recursively. As a result, a tree of decision nodes and leaf nodes is formed, with each decision node representing a test on an input feature and each leaf node representing a class label [16].

Fig 3.14: Decision tree visualization[26]

**Random Forest Classifier:**

Random Forest Classifier is a classification algorithm that uses supervised learning. It is an ensemble learning method that, during training, builds multiple decision trees and outputs the class which is the mode of the classes predicted by the individual trees. For the given training dataset with targets and features, the decision tree algorithm will have a set of rules. A random forest, unlike decision trees, does not require calculating information gain to find the root node. It predicts the outcome based on the rules of each randomly generated decision tree and saves the result. As a result, the prediction with the most votes is the random forest algorithm's final forecast [12].

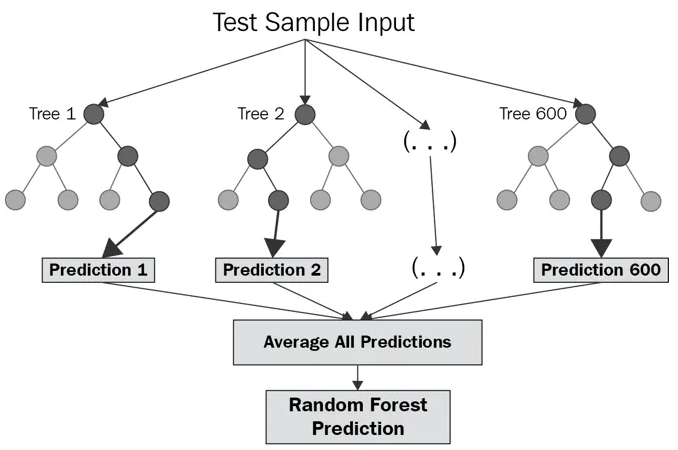


Fig 3.15 Random Forest Visualization[21]

**KNeighborsClassifier**

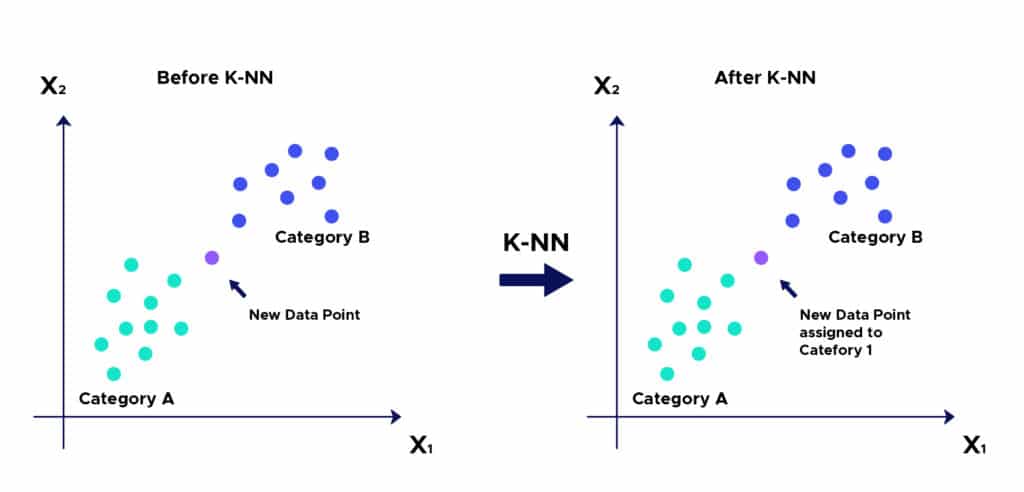
KNeighborsClassifier is a classification machine learning algorithm. It is a type of instance-based learning algorithm that works by locating the k nearest neighbors of a new data point and predicting the class label of the new data point using the class labels of those neighbors. The user specifies the value of k, which represents the number of neighbors to consider. However, it is sensitive to the choice of k and can be slow for large datasets, so caution should be exercised when using it for these types of problems [11].

Fig 3.16: KNeighborsClassifier Visualization[23]

**Stochastic Gradient Descent Classifier**

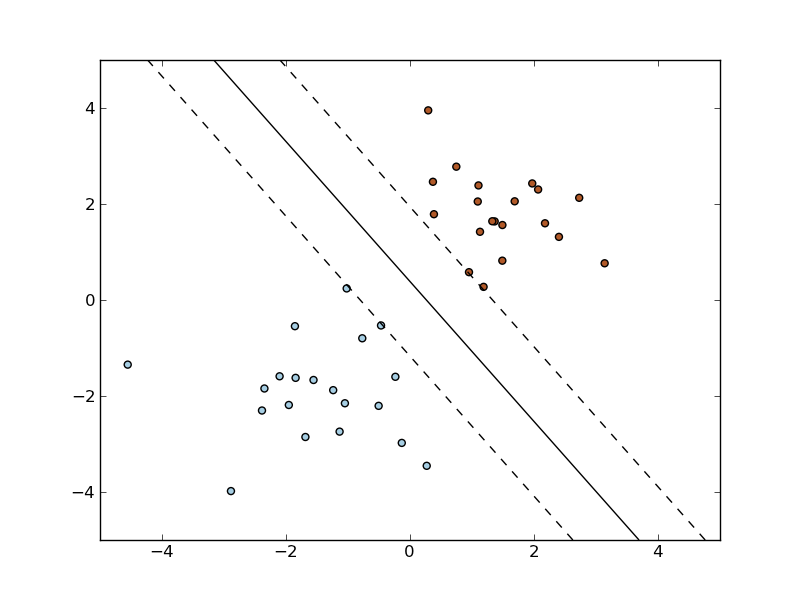
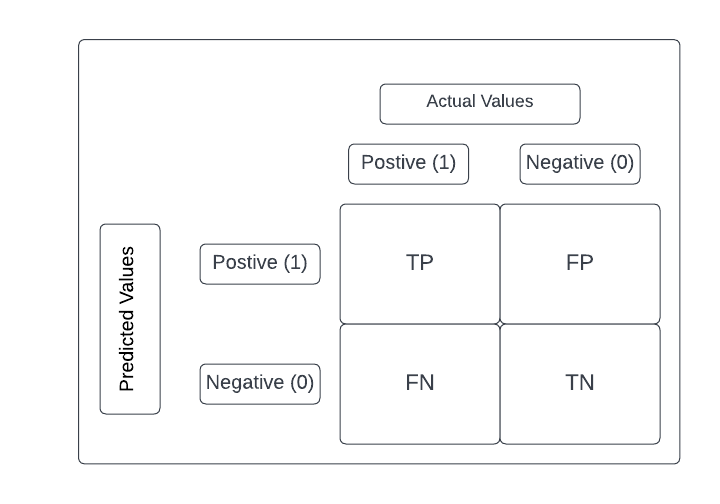
SGD Classifier is a linear model for classification tasks that are trained using the Stochastic Gradient Descent optimization algorithm. It is a straightforward and efficient algorithm that works well with large datasets. For each iteration of Stochastic Gradient Descent, a few samples are chosen at random rather than the entire data set. Gradient Descent uses the term "batch" to refer to the total number of samples from a dataset that are used to calculate the gradient for each iteration [10].

Fig 3.17: Stochastic Gradient Descent Classifier Visualization[22]

## 3.3 Performance Analysis

This research will compare performance by using various evaluation matrices. It will get how well the model can distinguish between abusive and non-abusive comments. This study will use four ML algorithms, namely SVC, MNB, LR, RF, DT, KNN, and SGC. It is essential to review the standard matrix to understand the performance of the conflicted models. The commonly used confusion matrix for performance analysis of ML algorithms is accuracy, precision, recall, f1-score, and roc area. Here briefly describe the matrix that will take into consideration to analyze the performance.

Fig 3.18: Confusion matrix

Here TP, FP, FN, and TN refer to True Positive, False Positive, False Negative, and True Negative respectively. The Confusion Matrix is extremely important for measuring Accuracy, Precision, and Recall. Equation 2, 3, 4, and 5 shows the calculation of accuracy, precision, recall, and f1-score.

**Accuracy:** It calculates the ratio of correctly identified records from the total number of records.

Accuracy = **----------------------(2)**

**Precision:** It estimates the ratio of the correctly identified records to the number of all identified records.

 Precision = **------------------------------------(3)**

**Recall:** It estimates the ratio of the correctly classified records to the total number of records.

 Recall = **------------------------------------- (4)**

**F1-Score:** It is the harmonic mean of Precision and Recall.

 F1-score = 2 ∗ **-------------------(5)**

## 3.4 Conclusion

It is possible to increase the accuracy to detect abusive Bangla comments on Facebook. The accuracy of the model depends on the data collection process and feature selection to train the model. Early detection and prevention of to spread of abusive Bangla comments can reduce the impact on the victim. The following chapter will mention our expected results and the impact of this proposed research study.

# CHAPTER IV

# RESULT ANALYSIS AND DISCUSSION

In the previous chapter, We showed the process of data collection, processing, and building models. All of those were the foundation for this chapter. In this chapter, We discussed the results that we found after implementing the algorithms and analyzed them.

## 4.2 Results Analysis

In this section, we will discuss the performance parameters of different Machine Learning algorithms we have applied. For performance analysis, we have taken accuracy, precision, recall, and f1 score as parameters.

Later in this chapter, we are going to show how our models respond in the case of a single comment. We take a comment from the Facebook page which does not belong to our dataset. Then for seven Machine Learning algorithms, we will check which algorithm detects which comments are abusive or non-abusive more correctly than others.

### 4.2.2 Performance of Machine Learning Algorithms

**Support Vector Classifier**

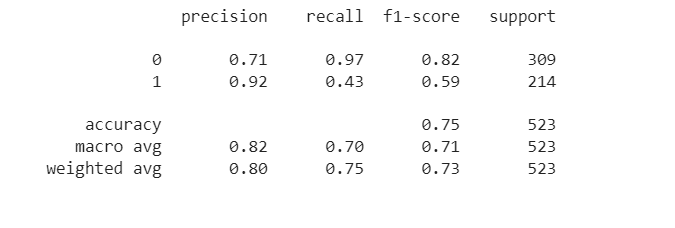
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.1: Classification report of Support Vector Classifier

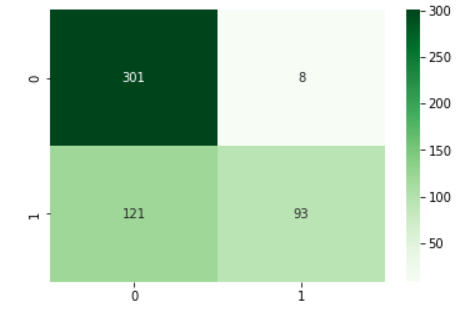
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.2: Confusion matrix ofSupport Vector Classifier

Accuracy for Support Vector Classifier: 75.33%

**Multinomial Naive Bayes**

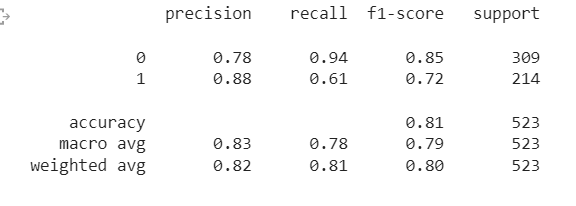
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.3: Classification report of Multinomial Naïve Bayes

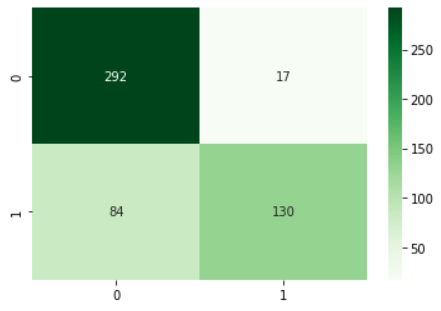
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.4: Confusion matrix of Multinomial Naïve Bayes

Accuracy for Multinomial Naive Bayes: 80.68%

**Logistic Regression**

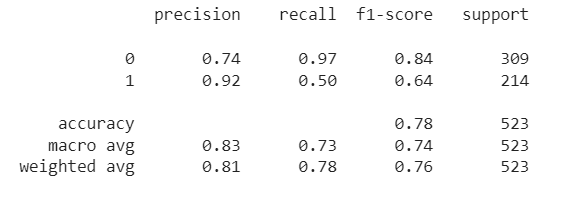
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.5: Classification report of Logistic Regression

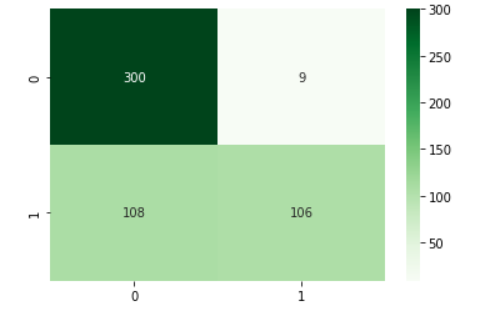
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.6:Confusion matrix of Logistic Regression

Accuracy for Logistic Regression: 77.62%

**Random Forest Classifier**

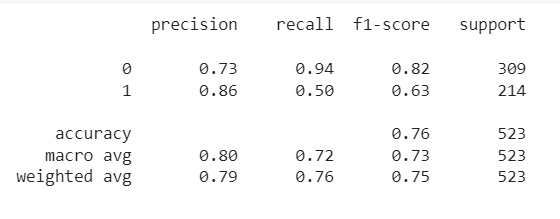
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.7: Classification report of Random Forest Classifier

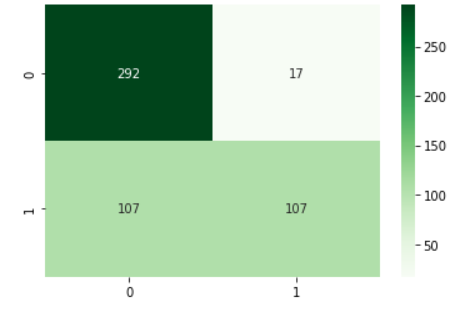
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.8: Confusion matrix of Random Forest Classifier

Accuracy for Random Forest Classifier:77.05%

**Decision Tree Classifier**

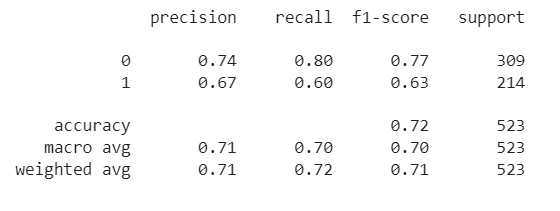
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.9: Classification report of Decision Tree Classifier

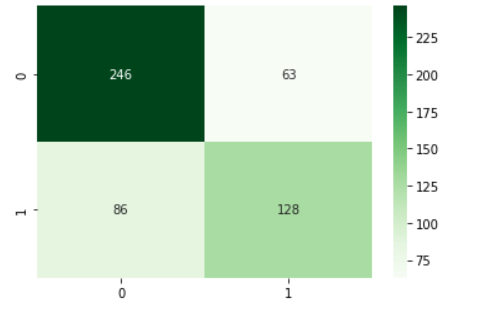
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.10: Confusion matrix of Decision Tree Classifier

Accuracy for Decision Tree Classifier: 70.12%

**KneighborsClassifier**

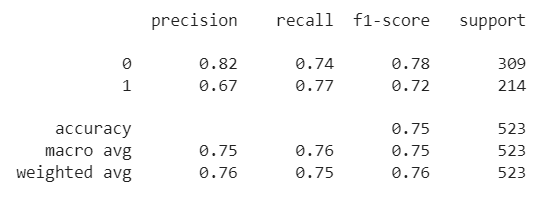
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.11: Classification Report of KneighborsClassifier

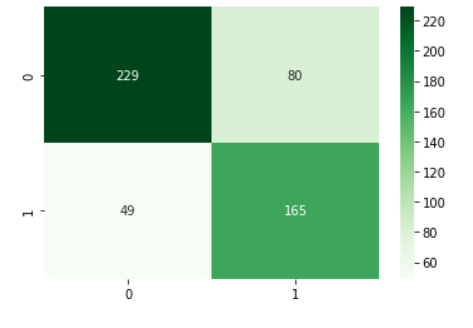
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.12: Confusion matrix of KneighborsClassifier

Accuracy for KneighborsClassifier: 75.33%

**Stochastic Gradient Descent Classifier**

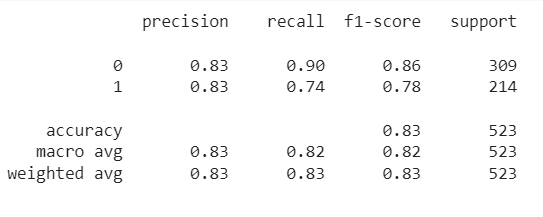
**Classification Report:** In this section, we will describe the classification report of the Support Vector Classifier. To plot the classification report we used the sklearn library.

Fig 4.13: Classification Report of Stochastic Gradient Descent Classifier

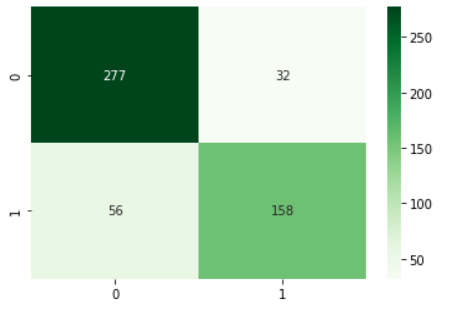
**Confusion matrix:** After obtaining our result and classification report of the Support Vector Classifier, we constructed a confusion matrix for the Support Vector Classifier. The confusion matrix is shown below:

Fig 4.14: Confusion matrix of Stochastic Gradient Descent Classifier

Accuracy for Stochastic Gradient Descent Classifier: 83.17%

### 4.2.3 Machine Learning Algorithms Comparison

Where,

SVC= Support Vector Classifier, MNB= Multinomial Naive Bayes, LR= Logistic Regression, RF=Random Forest Classifier, DT= Decision Tree Classifier, KNN= KneighborsClassifier, SGC= Stochastic Gradient Descent Classifier

**4.2.3.1 Comparison between Precision**

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Support Vector Classifier | 0.80 |
| Multinomial Naïve Bayes | 0.82 |
| Logistic Regression | 0.81 |
| Random Forest Classifier | 0.79 |
| Decision Tree Classifier | 0.70 |
| KneighborsClassifier | 0.76 |
| Stochastic Gradient Descent Classifier | 0.83 |

**4.2.3.2 Comparison between Recall**

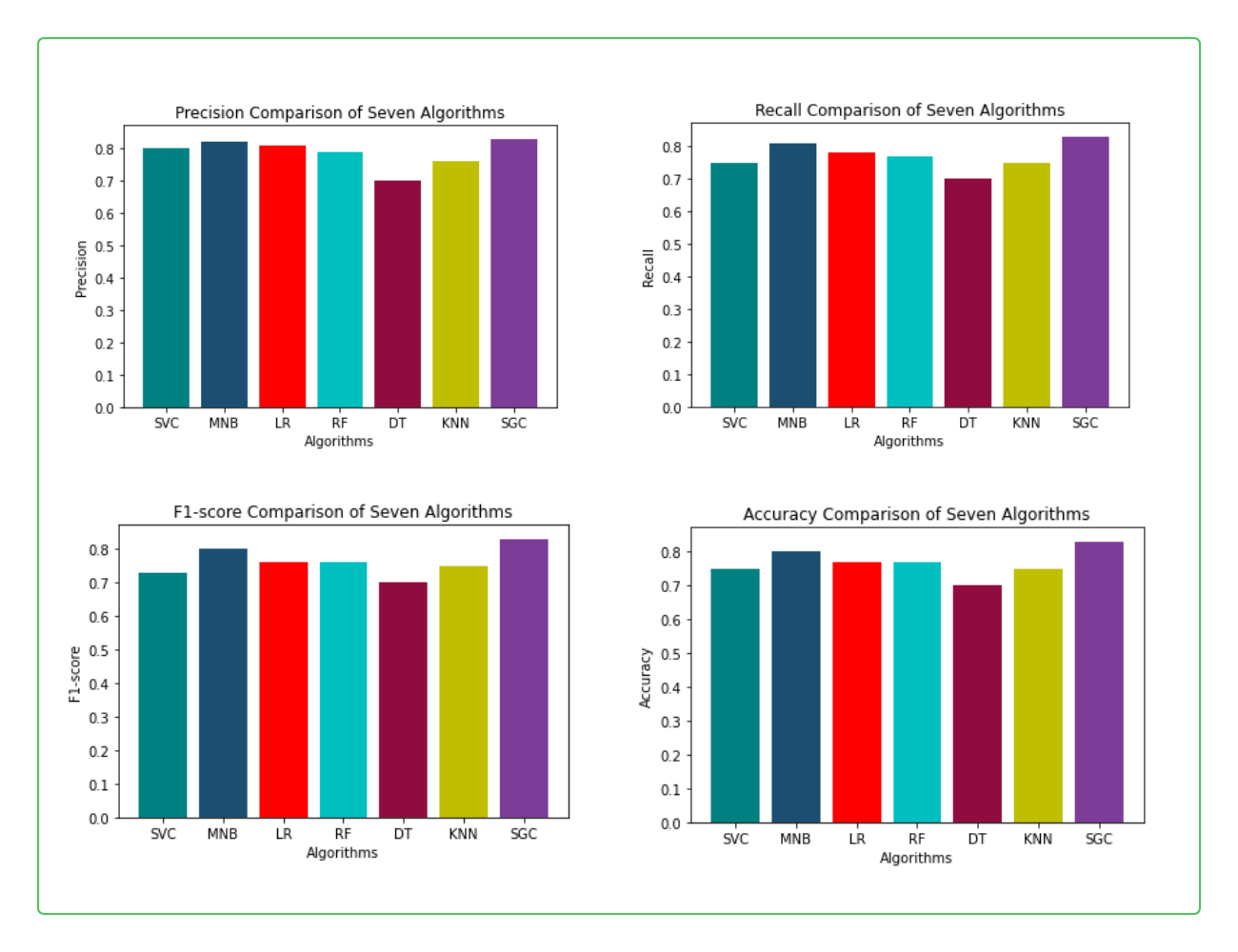
|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Support Vector Classifier | 0.75 |
| Multinomial Naïve Bayes | 0.81 |
| Logistic Regression | 0.78 |
| Random Forest Classifier | 0.77 |
| Decision Tree Classifier | 0.70 |
| KneighborsClassifier | 0.75 |
| Stochastic Gradient Descent Classifier | 0.83 |

**4.2.3.3 Comparison between F1-Score**

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Support Vector Classifier | 0.73 |
| Multinomial Naïve Bayes | 0.80 |
| Logistic Regression | 0.76 |
| Random Forest Classifier | 0.76 |
| Decision Tree Classifier | 0.70 |
| KneighborsClassifier | 0.75 |
| Stochastic Gradient Descent Classifier | 0.83 |

**4.2.3.4 Comparison between Accuracy**

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Support Vector Classifier | 75.33% |
| Multinomial Naïve Bayes | 80.68% |
| Logistic Regression | 77.62% |
| Random Forest Classifier | 77.05% |
| Decision Tree Classifier | 70.17% |
| KneighborsClassifier | 75.33% |
| Stochastic Gradient Descent Classifier | 83.17% |

Fig 4.15: Performance Measure comparisons of Algorithms(TF-IDF)

Overall, We get

**Best Accuracy:** Stochastic Gradient Descent Classifier, 83.17%

**Best Precision:** Stochastic Gradient Descent Classifier, 83%

**Best Recall:** Stochastic Gradient Descent Classifier, 83%

**Best F1-Score:** Stochastic Gradient Descent Classifier, 83%

## 4.3 Single Comment Check:

In this section, we detect abusive and non-abusive Bangla comments from a given textual comment. All of the testing comments are unseen for the model. Some examples are given below.

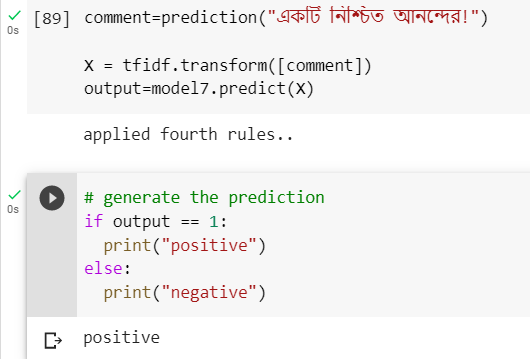
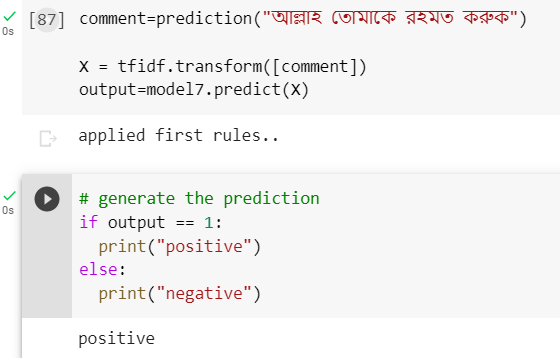


Fig 4.16: Two positive comments

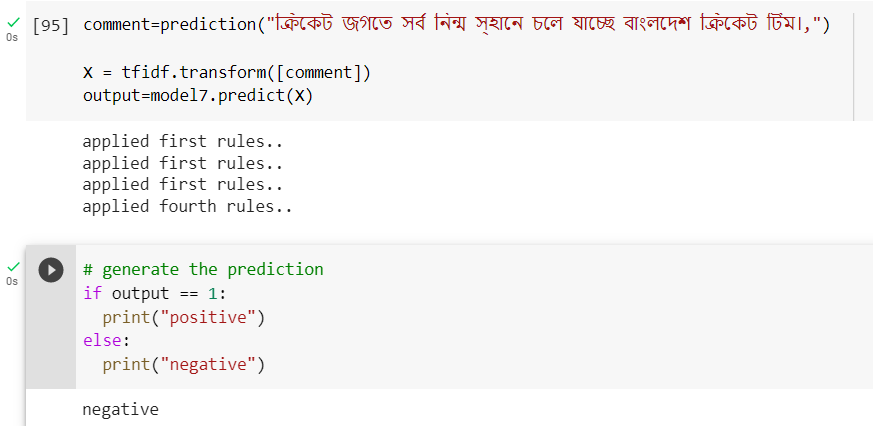
In this above comment, the sentiment behind this positive. So we can say these two comments are positive.

Fig 4.17: Negative comment

In this above comment, the sentiment behind this is Negative. So we can say these comment are Negative comments.

## 4.4 Discussion

From all the analyses we made above, we see that the Stochastic Gradient Descent Classifier performed better than all seven Machine Learning algorithms. It gives about 83.31% accuracy to detect abusive and non-abusive comments from textual data. Multinomial Naive Bayes gives the second-best accuracy which is 80%. Other algorithms perform below average.

Hence to detect abusive and non-abusive Bangla comments SGC can be a better choice.

The performance of algorithms varies on the size of the data. If our datasets consist of more data, then it surely has more to train than if may give better accuracy. It also depends on data preprocessing, if we don't consider some of the preprocessing steps then the accuracy becomes lesser.

# CHAPTER V

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

Abusive Bangla comments detection on public Facebook Pages using Machine Learning Algorithm is a challenging task, but it can be achieved with appropriate data preprocessing, feature engineering, and algorithm selection. Machine learning algorithms like SVC, MNB, LR, RF, DT, KNN, and SGC models can be used for this task.

To achieve the best performance, a thorough evaluation of the trained models is necessary, including the use of performance matrices like accuracy, precision, recall, F1 score, and confusion matrix. Model testing is also essential to evaluate the real-world performance of the best-performing model.

By predicting abusive and non-abusive comments, we can filter out and flag abusive comments, which can help prevent the spread of hate speech and maintain a safe and respectful environment on public Facebook pages. Ongoing monitoring of the performance of the machine learning algorithms is necessary to ensure their effectiveness and make necessary adjustments as needed.

## 5.2 Future Work

Since collecting a large amount of labelled data for the Bangla language is challenging, data augmentation techniques like data synthesis, translation, and rotation can be used to expand the dataset and improve the performance of the models. Ensemble learning techniques, such as bagging, boosting, and stacking, can be used to combine the outputs of multiple machine learning algorithms and improve the overall performance of the system.

Overall, these potential areas for future work can help improve the performance and accuracy of the Abusive Bangla comments detection system and make it more effective in promoting a safe and respectful online environment.

# References

[1] E. A. Emon, S. Rahman, J. Banerjee, A. K. Das, and T. Mittra, "A Deep Learning Approach to Detect Abusive Bengali Text," 2019 7th Int. Conf. Smart Comput. Commun. ICSCC 2019, no. September, pp. 1–5, 2019, DOI: 10.1109/ICSCC.2019.8843606.

[2] S. C. Eshan and M. S. Hasan, “An application of machine learning to detect abusive Bengali text,” 20th Int. Conf. Comput. Inf. Technol. ICCIT 2017, vol. 2018-Janua, pp. 1–6, 2018, DOI: 10.1109/ICCITECHN.2017.8281787.

[3] R. R. Dalvi, S. Baliram Chavan, and A. Halbe, “Detecting A Twitter Cyberbullying Using Machine Learning,” Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020, no. Iciccs, pp. 297–301, 2020, DOI: 10.1109/ICICCS48265.2020.9120893.

[4] N. Romim, M. Ahmed, H. Talukder, and M. Saiful Islam, “Hate Speech Detection in the Bengali Language: A Dataset and Its Baseline Evaluation,” pp. 457–468, 2021, DOI: 10.1007/978-981-16-0586-4\_37.

[5] J. Mehta, “Cyber Bullying Detection using Machine Learning,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 9, no. 9, pp. 144–151, 2021, DOI: 10.22214/ijraset.2021.37934.

[6] E. Ijmtst, “Imbalanced Text Classification Features for Toxic Comments,” no. January 2022, DOI: 10.46501/IJMTST0801054.

[7] N. Romim, M. Ahmed, M. S. Islam, A. Sen Sharma, H. Talukder, and M. R. Amin, “HS-BAN: A Benchmark Dataset of Social Media Comments for Hate Speech Detection in Bangla,” pp. 1–8, 2021, [Online]. Available: http://arxiv.org/abs/2112.01902

[8] M. T. Ahmed, M. Rahman, S. Nur, A. Islam, and D. Das, “Deployment of machine learning and deep learning algorithms in detecting cyberbullying in Bangla and romanized Bangla text: A comparative study,” Proc. 2021 1st Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2021, no. February 2022, 2021, DOI: 10.1109/ICAECT49130.2021.9392608.

[9] T. Islam, N. Ahmed, and S. Latif, “An evolutionary approach to comparative analysis of detecting bangla abusive text,” Bull. Electr. Eng. Informatics, vol. 10, no. 4, pp. 2163–2169, 2021, DOI: 10.11591/EEI.V10I4.3107.

[10] E. A. E. Lucky, M. M. H. Sany, M. Keya, S. A. Khushbu, and S. R. H. Noori, “an Attention on Sentiment Analysis of Child Abusive Public Comments Towards Bangla Text and Ml,” 2021 12th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2021, no. July 2021, DOI: 10.1109/ICCCNT51525.2021.9580154.

[11] M. S. A. Sanoussi, C. Xiaohua, G. K. Agordzo, M. L. Guindo, A. M. M. A. Al Omari, and B. M. Issa, “Detection of Hate Speech Texts Using Machine Learning Algorithm,” 2022 IEEE 12th Annu. Comput. Commun. Work. Conf. CCWC 2022, pp. 266–273, 2022, DOI: 10.1109/CCWC54503.2022.9720792.

[12] A. Isaac, R. Kumar, and A. Bhat, “Hate Speech Detection Using Machine Learning Techniques,” Lect. Notes Electr. Eng., vol. 840, no. Iciccs, pp. 125–135, 2022, DOI: 10.1007/978-981-16-9012-9\_11.

[13] M. T. Ahmed, M. Rahman, S. Nur, A. Z. M. T. Islam, and D. Das, “Natural language processing and machine learning based cyberbullying detection for Bangla and Romanized Bangla texts,” Telkomnika (Telecommunication Comput. Electron. Control., vol. 20, no. 1, pp. 89–97, 2022, DOI: 10.12928/TELKOMNIKA.v20i1.18630.

[14] A. M. Ishmam and S. Sharmin, “Hateful speech detection in public Facebook pages for the Bengali language," Proc. - 18th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019, no. December, pp. 555–560, 2019, DOI: 10.1109/ICMLA.2019.00104.

[15] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, “Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier,” Int. J. Inf. Eng. Electron. Bus., vol. 8, no. 4, pp. 54–62, 2016, DOI: 10.5815/ijieeb.2016.04.07.

[16] P. William, R. Gade, R. E. Chaudhari, A. B. Pawar, and M. A. Jawale, “Machine Learning based Automatic Hate Speech Recognition System,” Int. Conf. Sustain. Comput. Data Commun. Syst. ICSCDS 2022 - Proc., pp. 315–318, 2022, DOI: 10.1109/ICSCDS53736.2022.9760959.

[17] B. Haidar, M. Chamoun, and A. Serhrouchni, “A multilingual system for cyberbullying detection: Arabic content detection using machine learning,” Adv. Sci. Technol. Eng. Syst., vol. 2, no. 6, pp. 275–284, 2017, DOI: 10.25046/aj020634.

[18] N. I. Remon, N. H. Tuli, and R. D. Akash, “Bengali Hate Speech Detection in Public Facebook Pages,” 2022 Int. Conf. Innov. Sci. Eng. Technol. ICISET 2022, no. February, pp. 169–173, 2022, DOI: 10.1109/ICISET54810.2022.9775900.

[19] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, “Cyberbullying detection solutions based on deep learning architectures,” Multimed. Syst., 2020, DOI: 10.1007/s00530-020-00701-5.

[20] Support Vector Machine (SVM) Algorithm - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[21] Machine Learning Random Forest Algorithm - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/machine-learning-random-forest-algorithm

[22] GeeksforGeeks. (2023, March 1). ML Stochastic Gradient Descent SGD. https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/

[23] Team, D. S. (2022, September 28). KNN Algorithm in Machine Learning using Python. Resources for Data Science. https://datamites.com/blog/k-nearest-neighbor-knn-algorithm

[24] B. (n.d.). bnlp-resources/sentiment/youtube\_sentiment at main · banglanlp/bnlp-resources.GitHub. <https://github.com/banglanlp/bnlp>

[25] Logistic Regression in Machine Learning - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/logistic-regression-in-machine-learning

[26] Decision Tree Algorithm in Machine Learning - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm