Offensive Comment Detection from Bangla Text

Submitted by

Shahriar Hasan Chowdhury **170104030**
Tasmiyah Binta Islam **170104033**
Sadia Tasnim **170104037**

Submitted To

Faisal Muhammad Shah, Associate Professor
Md. Tanvir Rouf Shawon, Lecturer
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September 2021

ABSTRACT

With the development of technology, everyone has access to the internet throughout the world where educated and uneducated people both can have the opportunity to run a social media account. As the users are expanding, so is cyberbullying. The Unicode system has brought out the chance for people to interact with each other in Bangla language and in various social media it is surveyed that people are getting bullied publicly in Bangla language also. There has been several researches accomplished to detect offensive texts in English language, but there are few in Bangla language comparatively. Detecting Bangla offensive texts can be a way to prevent cyberbullying and as there are few works executed on this topic, we would have a great span in this field. In this project, we have worked on classifying the texts into binary categories which are offensive and non-offensive.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, social networking has become a very important part of human life. With the help of various types of social media like Facebook, Twitter, Instagram etc. people tend to share their opinions or express their emotions in these platforms. The number of Bangla text users are rapidly increasing which totally makes sense as Bangla is the 7th most spoken language worldwide [1]. Bengali people can easily communicate in their preferable Bangla language which has made their interaction process quite easy for them. Though this system is quite advantageous, it has brought some side effects with itself. Now we can see people getting bullied in public social platforms in Bangla language which is worse as bullying in social platforms can spread very quickly to wider audiences.

In the last few years, many monitoring has been going on to expand the safety of people online, but it's quite difficult to detect all the bullying in vast social media platforms. That is why we are motivated to build an automatic system which will monitor all these offensive comments and keep a track on these. Significant works have been done on this issue in English Language where there aren't many noticeable works accomplished in the Bangla language. This clearly opens the opportunity for us to make an automatic system which will be for our native language and we hope to make this a prevention system for cyberbullying issues.

Considering the impacts of Cyberbullying, it has to be prevented before the spreading of any kind of bullying is occurred. So, our aim is to detect the offensive comments in Bangla text and we have tried to detect the offensive comments and classify them with a machine learning approach and come to an acceptable outcome.

Chapter 2

Literature Reviews

2.1 Review of Related Papers

There has been numerous research accomplished on detecting cyberbullying in various languages. We tried to collect the relevant research papers with our topic which is detecting offensive words in Bangla text and collected information which we thought were useful and necessary. There weren't many research papers found for Bangla text where satisfactory accuracies were found. Summaries of some research papers that we read are given here in the review section.

The objective of this paper [2] was to develop a cyberbullying detection model for Bangla text from social media, and to compare the performance of the algorithms. For this 4 machine learning algorithms (NB, SVM, Decision Tree and KNN) were used for detecting cyber bullying. The data used in this paper was collected from Facebook and Twitter. All of those collected data were labelled manually as either bullied or not-bullied. For vectorization, TF-IDF vectorizer was used as a text based feature. Stemming and tokenization has been applied on the data to facilitate the feature extraction. For extracting text based features a tri-gram model with tokenization of words were used. 2,400 Bangla texts with 10% bullying text were used for training the model and 10-Fold cross validation model was used for testing the performance. The authors showed that SVM performs better with 97.27% accuracy compared to other algorithms implied.

In the next paper [3] it is observed that the authors used various machine learning algorithms to detect abusive Bengali text. Multinomial Naive Bayes (MNB), Random Forest (RF) and Support Vector Machine (SVM) were used in this paper. For SVM, four SVM kernels are used which are Linear, Radial Bias Function, Polynomial, Sigmoid. For the Data, the author used Facebook comments from several Facebook celebrities' profile pages where only Bengali Unicode characters were considered.The authors showed that for Count Vectorizer, the accuracy rate of SVM linear increases with the increasing number of samples. If the test samples increase, maybe SVM will give more accurate results than MNB. At last the author stated that TFIDF Vectorizer features with SVM linear kernel performs the best.

The authors of the following paper [4] have worked on different machine learning algorithms and a model based on Deep Neural Network which were applied to detect the hateful speeches from Facebook pages. They developed an annotated dataset of 5126 Bangla comments from Facebook pages which were public. After performing the algorithms they have attained 52% accuracy on the Random Forest algorithm and 70.10% accuracy on the GRU based model. In their report they have evaluated their experiment for each configuration ten times and reported the average performance where they have used 90% data for training and 10% for testing.

By observing the next paper [5], it is noticeable that a classification model was constructed using text mining method with optimal accuracy in identifying cyberbully conversation in English text using Naive Bayes method and Support Vector Machine (SVM). A total of 12,729 data was collected from kaggle from which 11,661 data was given non-cyberbullying labeled and 1068 data was labeled cyberbully. The authors showed that the SVM method gives best performance in almost all cases. With 2 class classification SVM gives 99.4% accuracy. With four classes the accuracy for SVM is 97.81%. And for 11 classes the accuracy is 94.12%. And the highest average Accuracy is 92.11 based on the Model (SVM). The authors also showed that accuracy increases with higher n-grams. And the maximum average accuracy is 92.75% N-gram 5, and the lowest accuracy set at n-gram 1 (89.05%).

In the following paper [6] the authors have collected the data manually from the public comment sections of different social media and labeled their data by taking opinions from 50 persons. As they have processed their experiment manually, their dataset consists of only 300 data. They have proposed their own algorithm for detecting that the comment is abusive or not abusive. In their algorithm, they have measured the weight of the abusive or non-abusive words and made a processed calculation into the whole comment and found the result. This algorithm works better with increasing numbers of comments as they have shown much accuracy with 300 comments than 100 comments. They have proposed that their algorithm could be integrated with machine learning algorithms like SVM, RF, and Naïve Bayes.

In the next article [7], an encoder-decoder-based machine learning model was used where they detected Bangla hate speeches on social media using an attention-based recurrent neural network. They used a dataset that had 7425 Bengali comments. They collected the dataset from Facebook pages using Facebook Graph API. They classified the data into 7 categories (Hate speech, aggressive comment, religious hatred, ethnical attack, religious comment, political comment, and suicidal comment). They also did binary classification(abusive and non-abusive). They achieved 74% accuracy for LSTM decoder and GRU decoder models. After using the Attention mechanism the model was improved and got 77% accuracy. They showed that the attention mechanism was better for both binary and multi-class labeled datasets.

2.2 Overview and Comparison of the Related Papers

Since we read multiple papers to keep ourselves updated with the work done so far, we think it is the best approach to sum up a comparison table featuring the dataset, classifications, accuracy result, etc. The table is as follows:

Table 2.1: Summary of existing techniques

Reference Work	Training Data	Output Classes	Classification Algorithm	Performance/ Success
Social Media Bullying detection using machine learning on bangla text [2]	2400 Bangla text collected from Facebook and Twitter	2 Classes: Bullied, non-bullied	4 algorithms: SVM, NB, Decision Tree, KNN	Maximum accuracy: 97.27% (SVM)
An application of Machine Learning to Detect Abusive Bengali Text [3]	2500 Facebook Comments	2 classes: abusive and non-abusive	6 Algorithms: RF, MNB, SVM (Linear), SVM (Polynomial), SVM (RBF), SVM (Sigmoid)	TF-IDF Vectorizer features with SVM linear kernel performs the best

Continued on next page

Continuation of Table 2.1

Reference Work	Training Data	Output Classes	Classification Algorithm	Performance/ Success
Hateful Speech Detection in Public Facebook Pages for the Bengali Language [4]	5126 comments from Facebook pages	6 Classes: Hate Speech, Communal Attack, Inciteful, Religious Hatred, Political, Religious	6 algorithms: SVC, Linear SVC, NB, RF, Adaptive Boost, GRU Based Model	Maximum Accuracy: 70.10% with GRU based deep neural network
Cyberbullying Classification using Text Mining [5]	1600 conversation, and a total of 12,729 text data from www.kaggle.com	2 classes, 4 classes, 11 classes	5 algorithms: NB, SVM Linear, SVM Poly, SVM RBF, SVM Sigmoid	For 2 classes: SVM-poly 99.41% For 4 classes: SVM-poly 97.81% For 11 classes: SVM-poly 94.12%
An Approach to Detect Abusive Bangla Text [6]	300 comments from Facebook pages	2 classes: abusive and not abusive	Root level algorithm (proposed)	Accuracy increases with data
Bangla hate speech detection on social media using attention-based recurrent neural network [7]	7425 Bengali comments from Facebook pages	Binary class: abusive and non-abusive Multiclass: Hate speech, aggressive comment, religious hatred, ethnical attack, religious comment, political comment, and suicidal comment	Three encoder–decoder-based models: LSTM, GRU, Attention Mechanism	Maximum accuracy: 77% accuracy with attention-based decoder

Chapter 3

Data Collection & Processing

3.1 Data Collection

Collecting the proper amount of data is a very important mission as the models we would be using will be dependent on it. While we will be using machine learning models, the collection of data has to be efficient. There are not many pieces of research conducted with Bangla offensive comments, so it was quite hard to find any prepared dataset from online sources.

We have tried to find a proper dataset to use from several sources and managed to collect a Bangla abusive dataset from GitHub [8], which is a collection of 10,219 comments in total. There are 5964 comments which are non-abusive and 4354 comments which are abusive. Here is a short snap of our labeled dataset,

	Unnamed: (	comment_text	toxic	threat	obscene	insult	racism	isAbusive
0	0	এই গুলার জায়গামতো চুঁচড়া লাগিয়ে দেয়া দরকার! যত্তসব ভন্ড!"	1	0	0	1	0	1
1	1	আসিফ খানকির পোলা, তোর মারে কুত্তা দিয়ে চুদে তোরে পয়দা করসিল	1	0	1	1	0	1
2	2	শুয়োরের জ্ঞান সীমিত তাই সারাদিন গাঁজাখোরদের মত চিল্লাচিল্লি করতে থাকে।	1	0	0	1	0	1
3	3	এই মাদ্রাসার হোগুরকে ডিলডো মেরে শাস্তি দেয়া হোক।	1	0	0	1	0	1
4	4	নাস্তিকতা মানে ই মানুষিক রোগী ।	1	0	0	0	1	1
5	5	এই বানর !!আপনি কোন জংগল থেকে উদিত হইলেন ?	1	0	0	1	0	1
6	6	সাপের বাচ্চা বুইড়া খাটাস, যে নিজে ছাগু বচন করে বেড়ায়, অমানুষ বলা চলে	1	0	0	1	1	1
7	7	নেশা করে পুরাই মাতাল, নাস্তীক পারার সদস্য কিনা সে জন্যয় এ অবস্থা	1	0	0	1	1	1
8	8	হাহা ভাইকি গাঞ্জা একটু বেশি খাইছেন	1	0	0	1	0	1
9	9	ওয়া হাহা হীহী, আগে জান্তাম নাস্তীকরা সেক্স করার জন্য নাতাল, নাস্তীক ছেলে উধারতার নামে তার মা বোন	1	0	1	0	1	1
10	10	হুজুর মাদ্রাসার ছেলেরা ছাগলের সাথে সেক্স করে গরুর সাথে সেক্স করে আমার কাছে যথেষ্ট প্রমাণ আছে, অ	1	0	1	0	1	1
11	11	হা হা হা, নাস্তীস মাতালদের শেষ হাতিয়ার, মাতলামী যোক্তি আর কথখ্যন চলে!!! তাই কথা গুরিয়ে অন্যদিবে	1	0	0	0	1	1
12	12	তুই তো একটা বোকাচোদা তোকে দেখালে কি আর না দেখালে কি তুই বোকাচোদা থাকবি	1	0	0	1	0	1
13	13	ওয়াক থু,,, ছিঃছিঃ সরম! তোমরা শুধু এই কামটাই ভাল পারো, চাই সে বাব হোক বা নিজের সন্তান হোক, এছা	1	0	0	1	1	1
14	14	বলদ মুমিন	1	0	0	0	1	1
15	15	কৃষ্ণ জাউররাটার কতটা ধোন ছিল বা কত বড় ধোন ছিল যে অতগুলা মাগি চুদছে,লিলা খেলা করছে।	1	0	1	0	0	1
16	16	কিরে মাদার ফাকার,তুই নাস্তিক সেটা তোর আর তো বউ এর হোগার ভিতর ভইরা রাখ। এগুলা পাব্লিসিটি ক	1	0	1	1	1	1
17	17	নিজেরে কি ভাবিস মাদারচোদ?	1	0	1	1	0	1
18	18	তুমার পক্ষি টা আল্লাহু আকবার কইরা ফালাইয়া দিব মাদারচোদ	1	1	1	0	0	1
19	19	ঐ লোকটা একটা বোকাচোদা	1	0	0	1	0	1
20	20	ধর্মান্ধদের জ্ঞান থাকে শিশ্নের আগায়।	1	0	1	0	0	1
21	21	ক্যারে তুর মারে পাঁচবার চুদার অাগেই ভয়ে কমেন্ট ডিলেট করলি?	1	0	1	0	0	1
22	22	এরাই পরে সেক্স কন্ট্রোল করতে না পেরে ছাগল লাগাই	1	0	1	1	0	1

Figure 3.1: Screenshot of labeled data

3.2 Data Preprocessing

Data preprocessing is needed to transform the raw data into a machine-understandable logical format. Unlike the English language, the Bangla language has diversity, variation, and complexity. For example, the words ‘he’, ’you’ etc can be expresses in several meanings in Bangla. So natural language processing for Bangla text takes many steps to properly process the data, which can improve the performance of the model.
For processing our data, we have used tag, link and mention removal, stop words and punctuation removal, and for vectorization we have used TF-IDF vectorizer.

- **Tag, Link and Mention Removal:**
  The comments we collected contained many tags, hyperlinks, URLs, mentions, etc. which do not have any relevance for text classification. So, the comments that consist of these properties, needed to be processed without these tags or mentions.

- **Stop Word and Punctuation Removal:**
  Stop words are the most common words in any natural language and it has to be removed for the processing of language in the model. The stop words do not add much value to the dataset as it just increases the length of the sentence and unnecessary for classifying the text. By removing these, the dataset will be focused on the important words to train the model precisely.

- **Tokenization:**
  Tokenization is a very simple but very important task in natural language processing. It is the process of splitting the sentence into some pieces which are called tokens. A Token can be a word, phrase, or sentence. The pattern of a sentence can be easily classified by simply analyzing the word present in the text. Tokenization is needed for converting text into numeric features and also for removing stop words. For our dataset, we have used the python bnlp module to tokenize the comments into words.

- **Text to Feature Vectorization (TF-IDF):**
  Text to feature vectorization is a procedure where the text is processed to feature vector, as the machine does not understand the texts directly. For the machine, the texts need to be in a numerical value which will be understandable by the machine to run the model.
  For our dataset, we used TF-IDF vectorizer to measure the relevance of a word, like how many times a word is in the document and how important that word is. This process is done by multiplication of two metrics, which are, measurement of the appearance of a word in a document and the inverse document of the frequency of a word in the document.

TF-IDF is combined into two parts which are term frequency, and inverse document frequency.

$$tf(t,d) = \frac{f_{t,d}}{\Sigma_{t' \epsilon d} f_{t',d}} \tag{3.1}$$

Here, tf(t, d) stands for term frequency and this represents the frequency of term t.

$$idf(t,d) = log \frac{N}{|d \epsilon D : t \epsilon d|} \tag{3.2}$$

Here, N stands for the total number of documents and $d\epsilon D : t\epsilon d$ is for the number of documents where the term t appears.

Then tf-idf is calculated as,

$$tfidf(t,d,D) = tf(t,d) * idf(t,d) \tag{3.3}$$

We used this for defining a word in a phrase in our dataset.

Chapter 4

Methodology

Machine learning has a huge variety of processes to recognize patterns within natural languages. But unlike English language, Bangla language has its own different patterns, features, natures, representations, and implementations of numerous varieties. Hence, classifying Bangla text could be more challenging to figure out than many Unicode based languages. For our project, we planned a workflow that helped us to go through a proper way to complete our proposed target.
The flow chart is given below.

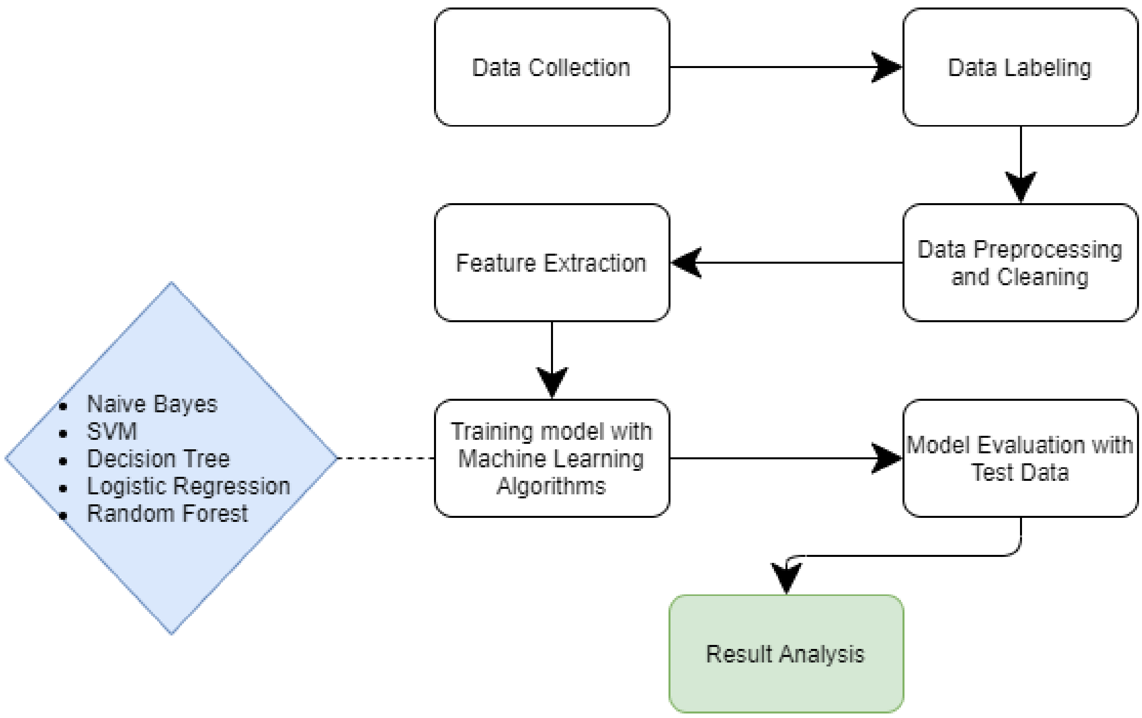

Figure 4.1: Work Flow Diagram

Chapter 5

Experiments and Results

After processing the data, we prepared the dataset for training and testing. After that, we used some machine learning algorithms to train the model and measured its performances by testing data. The machine learning models that we used are SVM, Naïve Bayes, Decision Tree, Random Forest, and Logistic regression.

5.1 Performance of Different Models

We measured Accuracy, Precision, Recall, and F1-score to understand the performance of each model we built with different machine learning algorithms. Also, confusion matrix is shown to describe the performance of the classification models.

5.1.1 Support Vector Machine

We used SVM RBF kernel to train our model and measured the performance for the binary class. The percentage of the performance for SVM is expressed by a table below,

Table 5.1: Performance of SVC-RBF

	Accuracy	**Precision**	**Recall**	**F1-Score**
Performance(%)	83.099	96.776	83.099	88.997

In our model, we plotted the measurements in a graph for better visualization of the performance which is given below,

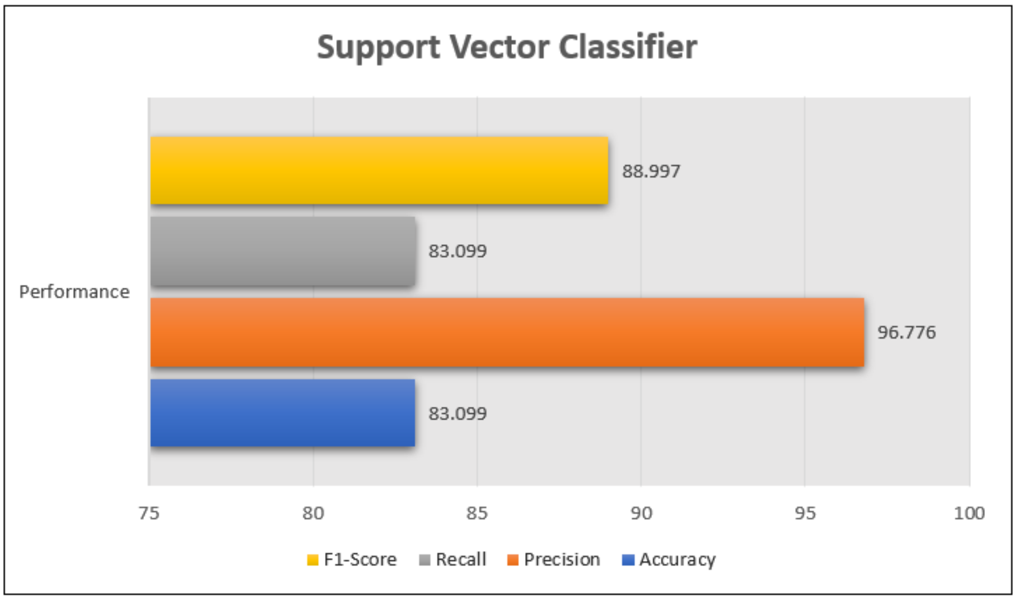

Figure 5.1: Graphical representation of 5.1

From the above figure we can see that SVC with RBf kernel gives a satisfactory result, where its accuracy is 83.099% and f1-score is 88.997%.

Confusion Matrix:

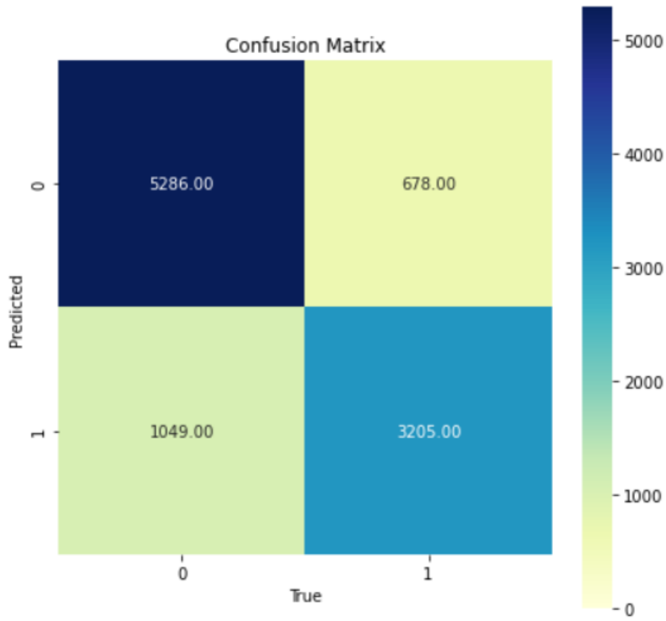

Figure 5.2: Confusion Matrix of SVM Model

5.1.2 Naive Bayes

Naive Bayes is a commonly used algorithm for text classification and we used naive Bayes algorithm to train and test our model. The performance measurement is given in the following table,

Table 5.2: Performance of Naive Bayes

	Accuracy	Precision	Recall	F1-Score
Performance(%)	76.112	96.049	76.112	82.793

The following graph is a visual representation of the performance table which is plotted in our model,

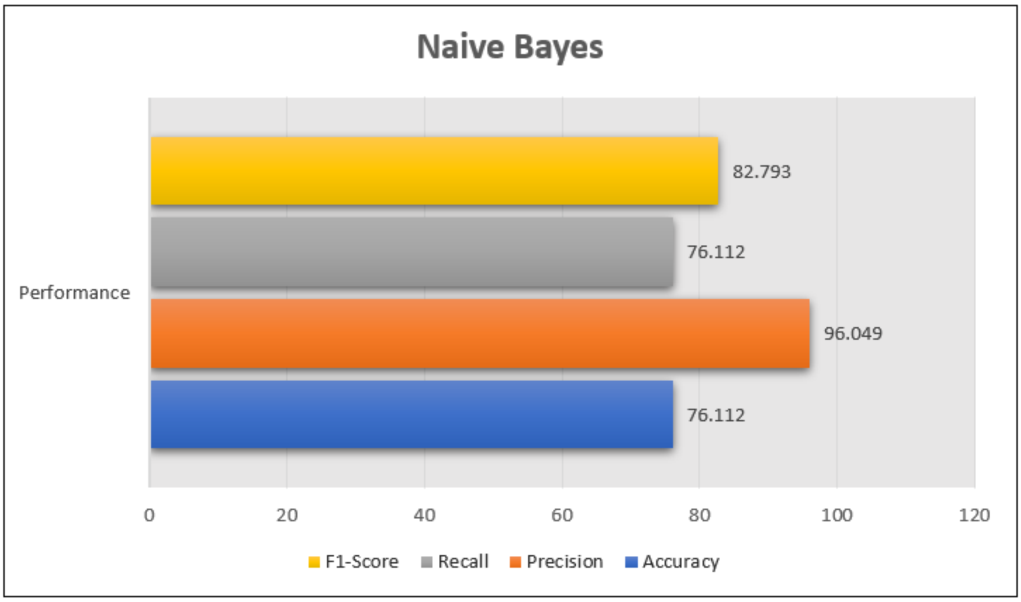

Figure 5.3: Graphical representation of 5.2

Here, we can see the accuracy Naive Bayes algorithm is 76.112%, and f1-score is 82.793%.

Confusion Matrix:

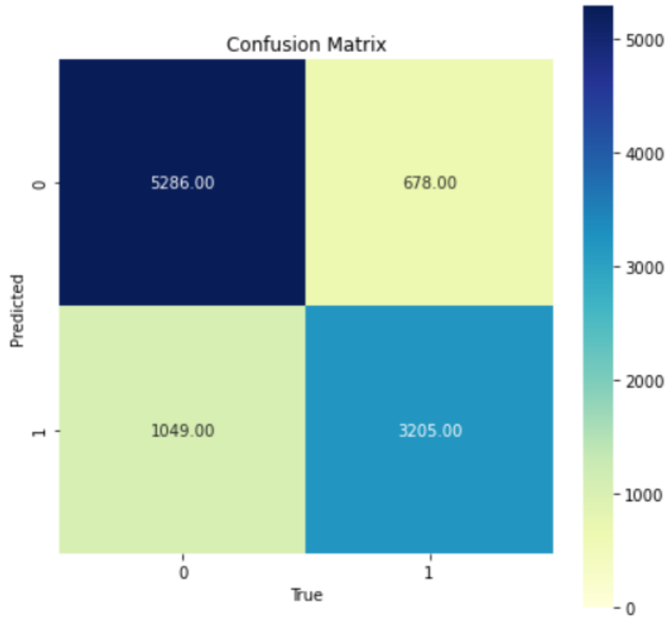

Figure 5.4: Confusion Matrix of Naive Bayes Model

5.1.3 Logistic Regression

Logistic Regression can be a very effective algorithm for binary text classification. We used this algorithm to train and test our model and the outcome as performance is presented in the table given below,

Table 5.3: Performance of Logistic Regression

	Accuracy	Precision	Recall	F1-Score
Performance(%)	79.135	96.51	79.135	86.463

In the following graph representation, we can see the performance measurement portrayed for the binary classification, where the accuracy is 79.135% and f1-score is 86.463%.

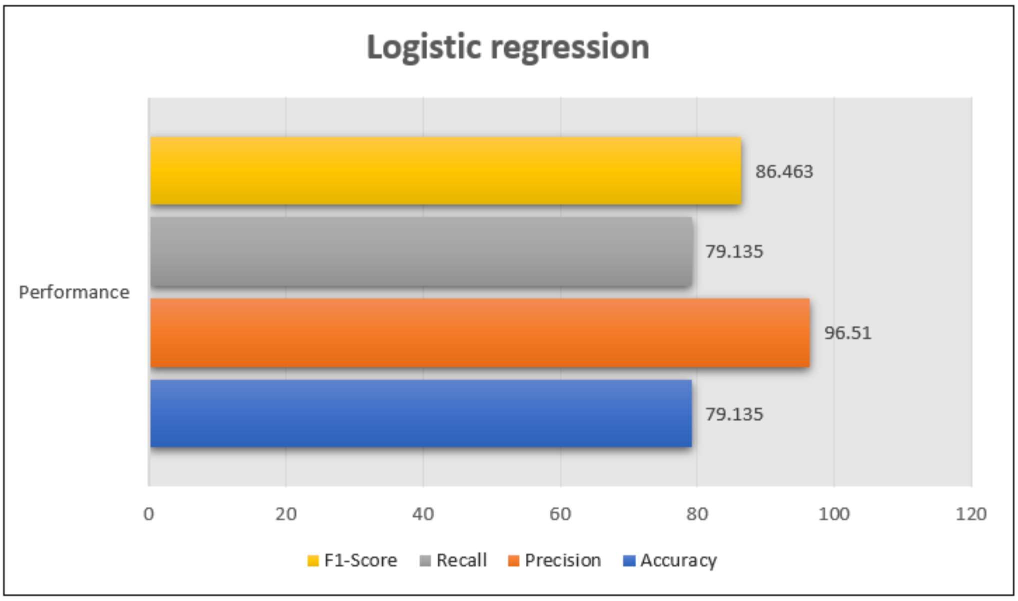

Figure 5.5: Graphical representation of 5.3

Confusion Matrix:

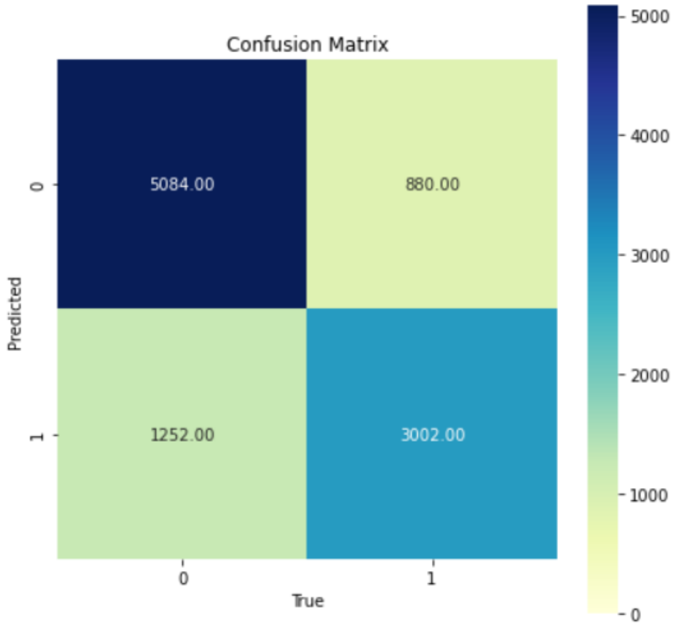

Figure 5.6: Confusion Matrix of Logistic Regression Model

5.1.4 Decision Tree

Decision tree algorithm can be used for both classification and regression. We used this algorithm to classify our data and measured the performance which we can notice from the given table,

Table 5.4: Performance of Decision Tree

	Accuracy	**Precision**	**Recall**	**F1-Score**
Performance(%)	75.005	95.416	75.005	83.489

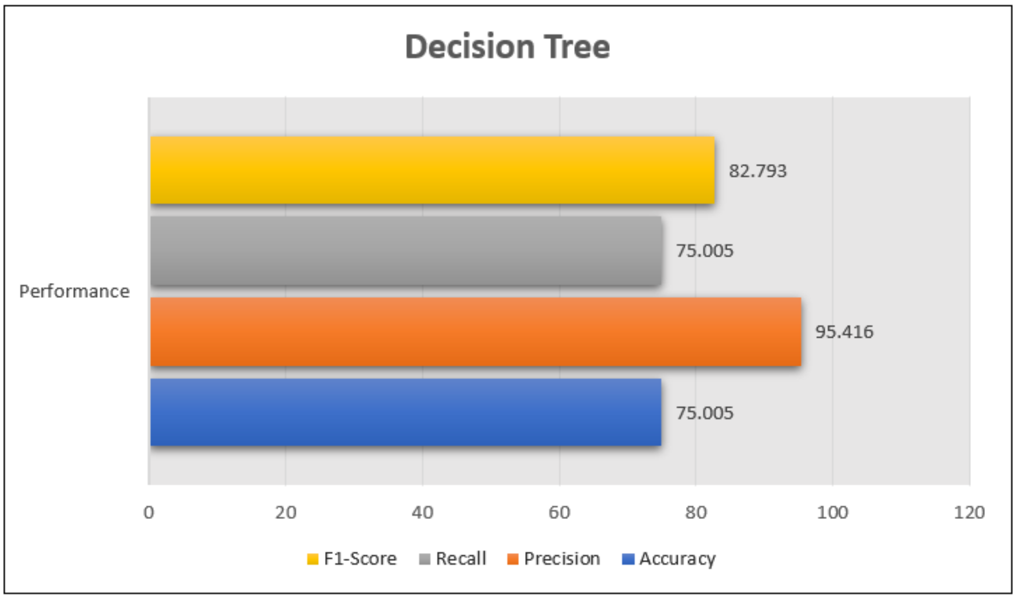

Figure 5.7: Graphical representation of 5.4

By the visual representation of the performance graph above, we can spot the accuracy for binary classification is 75.005% and f1-score is 83.489%.

Confusion Matrix:

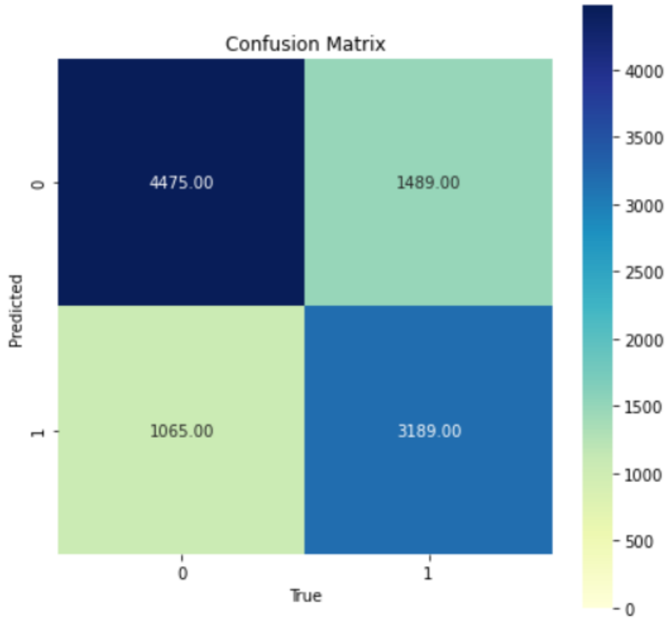

Figure 5.8: Confusion Matrix of Decision Tree Model

5.1.5 Random Forest

To train our model for classifying offensive texts, we used Random Forest algorithm. By using random forest, we got some measurements value of accuracy, precision, f1-score, and recall which is presented in the given table,

Table 5.5: Performance of Random Forest

	Accuracy	**Precision**	**Recall**	**F1-Score**
Performance(%)	79.086	96.013	79.086	86.154

We plotted the values of performance measurement and got a better visualization which is shown in the following,

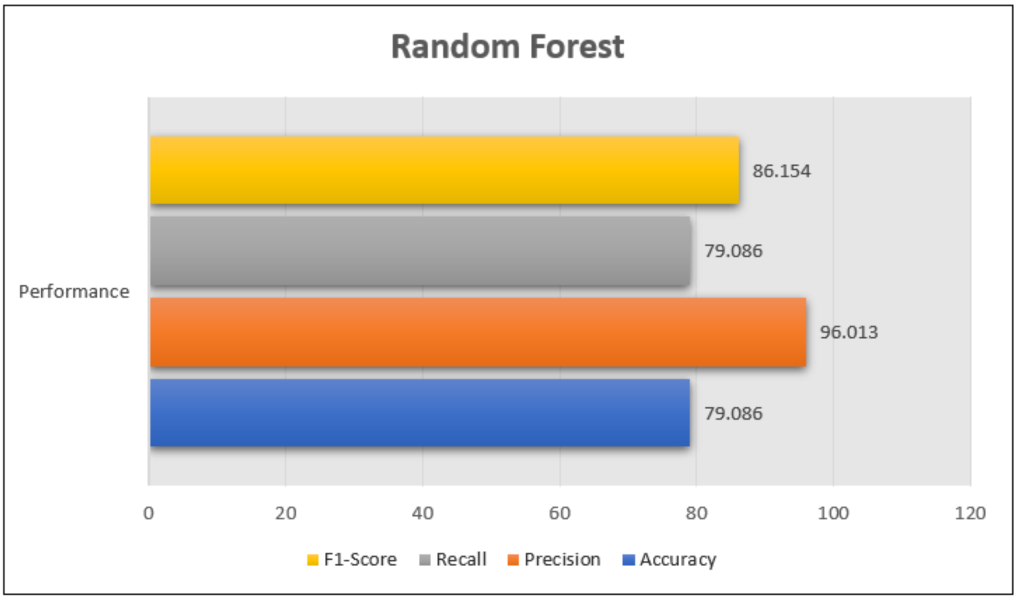

Figure 5.9: Graphical representation of 5.5

From the above graph, we can see the accuracy for binary classification is quite good. So, for our dataset and model, this algorithm gives an almost satisfactory result as the accuracy is 79.086% and f1-score is 86.154%.

Confusion Matrix:

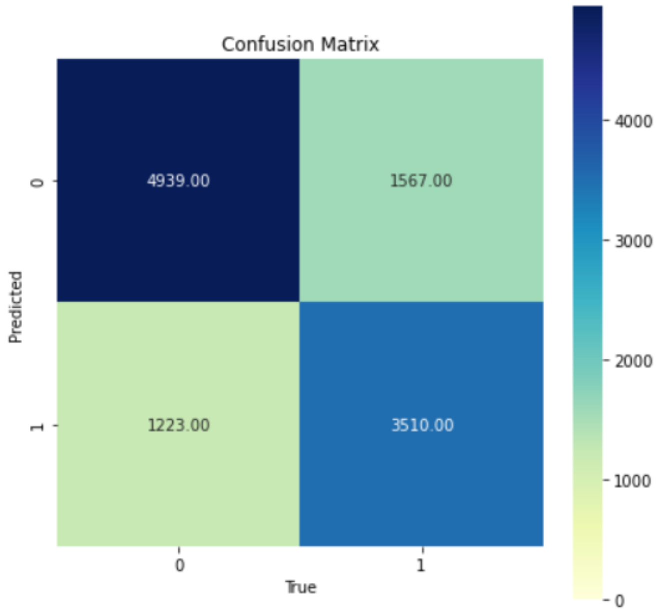

Figure 5.10: Confusion Matrix of Random Forest Model

5.2 Comparison of Different Models

Accuracy Comparison of Machine Learning Models:

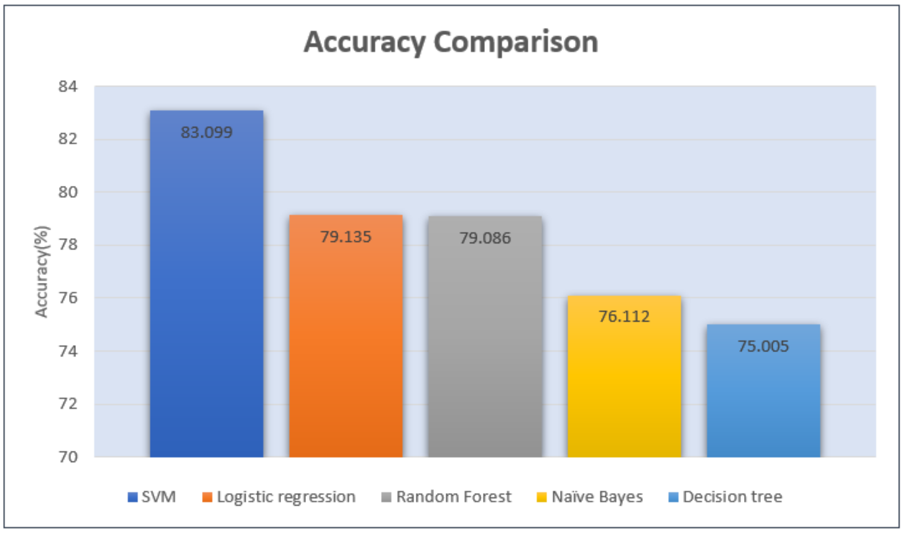

Figure 5.11: Accuracy of several ML models

In the above figure of graph representation, we have exhibited the accuracy of each machine learning model we used for binary classification. We came to a result that the highest accuracy we get for our model is from support vector machine algorithm which is 83.099%.

F1-score Comparison of Machine Learning Models:

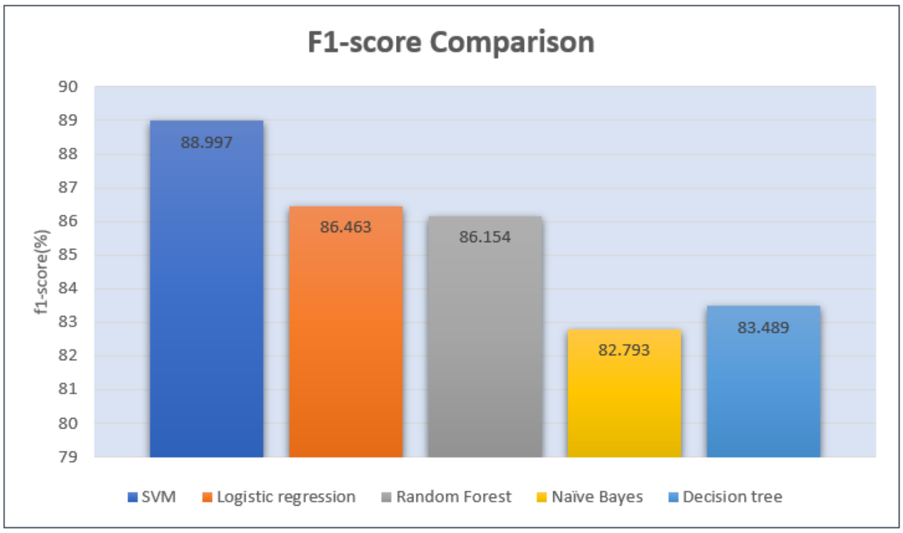

Figure 5.12: F1-score of several ML models

In the above figure of graph representation, we have exhibited the f1-score of each machine learning model we used for binary classification. We came to a result that the highest f1-score we get for our model is from support vector machine algorithm which is 88.997%.

Chapter 6

Future Work and Conclusion

Classification of offensive Bangla comments can help to categorize the bullying occurrence and monitor the process to prevent cyberbullying. Here, we collected a reasonable amount of data to train the models and classified the comments with binary classification. We used some machine learning models to find out the accuracy of each model and come to a decision on which model works better for classifying offensive texts in Bangla. Overall, among the machine learning models, SVM gave the best accuracy compared to others.

We have classified the texts into 2 classes. But a comment can be threatful, religious, toxic, and so on. So, we can do multiclass classification in the future. Also sometimes a comment can be both religious and threatful, and by just classifying the comments we will get the result as religious or threatful or just abusive. Nevertheless, the comment that is both religious and threatening, is much weighted than the comment being only religiously abusive. So, multiclass classification and severity measurement for these comments could be our next approach.

References

[1] "Ranked: The 100 Most Spoken Languages Around the World." https://www.visualcapitalist.com/100-most-spoken-languages/. Accessed: Feb 22, 2021.

[2] S. Akhter *et al.*, "Social media bullying detection using machine learning on bangla text," in *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*, pp. 385–388, IEEE, 2018.

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

[4] A. M. Ishmam and S. Sharmin, "Hateful speech detection in public facebook pages for the bengali language," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 555–560, IEEE, 2019.

[5] S. M. Isa, L. Ashianti, *et al.*, "Cyberbullying classification using text mining," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 241–246, IEEE, 2017.

[6] M. G. Hussain, T. Al Mahmud, and W. Akthar, "An approach to detect abusive bangla text," in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pp. 1–5, IEEE, 2018.

[7] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, "Bangla hate speech detection on social media using attention-based recurrent neural network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, 2021.

[8] "GitHub - aimansnigdha/Bangla-Abusive-Comment-Dataset." https://github.com/aimansnigdha/Bangla-Abusive-Comment-Dataset. Accessed: Feb 22, 2021.