Bangla Toxic Language Analysis using Machine Learning Approaches

CSE498R

A research submitted to the Department of CSE in partial fulfillment of the requirements for the degree of B.Sc. Engineering in CSE

By

Fahmida Sultana 181 2829 042

Under the guidance of
Dr. Sifat Momen
Associate Professor

Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
Summer 2022

Affiliated to
North South University

DECLARATION

I, hereby, declare that the work presented in this report is the outcome of my four months work performed under the supervision of Dr. Sifat Momen, Associate Professor, Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. The work was spread over a span of one of the final year courses, CSE 498R, Directed Research, in accordance with the course curriculum of the Department for the Bachelor of Science in Electrical and Electronics Engineering program.

Student's name & signature:

Fahmida

Fahmida Sultana
ID: 181 2829 042
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

APPROVAL

The research project report on ’Bangla Toxic Language Analysis using Machine Learning Approaches’ has been submitted by Fahmida Sultana (ID #1812829042), student of the Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. This report partially fulfills the requirement for the degree of Bachelor of Science in Electrical and Electronics Engineering in Summer 2022 and has been accepted as satisfactory.

Supervisor’s Signature

——————-

Dr. Sifat Momen
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

Department Chair’s Signature

——————-

Dr. Rajesh Palit
Professor & Chair
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

ACKNOWLEDGEMENT

All praises are due to Almighty Allah, Most Gracious, Most Merciful who blessed us to be here at North South University for pursuing the bachelors degree.

I am grateful and wish our profound indebtedness to **Dr. Sifat Momen**, Associate Professor, Department of Electrical and Computer Engineering (ECE), North South University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Machine Learning" and "Deep Learning" to carry out this project. His guidance, constant supervision, enthusiastic encouragement, sagacious advice and an effective surveillance throughout the entire period of the project have made it possible to complete this project.

I would like to express deepest appreciation towards **Dr. Rajesh Palit**, Professor & Chair of Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh.

Moreover, I would like to thank my friend Nusrat Islam (ID 222 5178 650), student of the Department of Electrical and Computer Engineering of North South University for helping me in this project.

I would also like to thank my family members, my parents, brothers and sisters who are the very reason of my existence. Without their unconditional love and support this research too could not be completed.

This research is dedicated to all of them.

Fahmida Sultana
181 2829 042

ABSTRACT

This research is dedicated to create a system which will fetch Bangla toxic language from online sites and identify whether the degree of toxicity of the language is medium, extreme or high. To carry out this research, dataset have been collected from Mendeley Data. The dataset contains 1958 toxic languages. This dataset contains utterances from the user-generated comments of Facebook.The data have been extracted from 8 publicly open Facebook pages. After Pre-processing such as removig whitespace, punctuation and tokenization the dataset have been splitted into 80% and 20% for training and testing the applied models respectively. Two types of feature extractors including CountVectorizer import from sklearn and another CountVectorizer has been implemented by a popular python pre-trained module known as *BnVec*. For training the dataset five types of machine learning models including Logistic Regression, Support Vector Machine, Random Forest Classifier, Decision Tree Classifier and Multinomial Naive Bayes have been applied. Accuracy of the applied was better after importing CountVectorizer from *BnVec*. 85.97%, 82.14%, 85.2%, 83.9% and 84.18% accuracy have been found from Support Vector Machine, Logistic Regression, Multinomial Naive Bayes, Random Forest Classifier and Decision Tree Classifier models respectively. So, highest accuracy obtained from Support Vector Machine.

Keywords: ML, CountVectorizer, Logistic Regression, SVM, Perceptron

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Social media playing an indispensable role in our daily life providing a public platform to share opinions including threats, spam and vulgar words often referred to as toxic comments. Toxic comments are defined as comments that are rude, disrespectful, or that tend to force users to leave the discussion. This online atmosphere creates disputable topics ranging from political propaganda, religious insanity and random hoax. Divided parties on such phenomena exchange hate comments including threats, vulgar words to attack each other personally. Such obscene words referred to toxic comments are harmful to safe user experience on the platform. Bengali stands seventh as the most spoken language in the world considering 228 million native speakers and this count is increasing rapidly due to its significance in demographic and political purposes[1]. Moreover, Bengali as a south Asian language is the national language of Bangladesh and it is also used partially in many regions of India. The increasing number of Bengali social media users post numerous statuses, comments, graphics, etc and they are instantly available for others to react. Generally, this often results in text with toxic comments and needs to be filtered out. Hence, it has become very important to analyze Bangla text data to maintain a safe and harassment-free online place.

1.2 Motivation

Toxic comments are comments that irritate people and spread hates among community. The worst of the internet comes in the form of online comments and it is getting worse each year. People often comment on a news story and it seems that, no matter what the story is, someone will find a way to connect it to politics, a personal attack or a conspiracy theory and they will have no problem posting it [2]. So it is important to identify the toxic comments. There is level of toxicity of the comments. Some comments are medium level toxic and some are very high or extreme. It is beneficial if level or degree of toxicity is also identified. In Bangladesh, many people often face harassment and bullying by strangers. In 2020, rights and legal aid NGO Ain O Salish Kendra (ASK) ran a survey in five districts of Bangladesh. They found that many young students had faced harassment online during the Covid-19 pandemic. Specifically, of the 108 children (61 girls and 47 boys) surveyed, a whopping 30% reported having faced abuse online. At least 56% of them were girls, and 88% had been harassed by strangers [3]. Huge amount of Bengali speakers use Bangla Language in online platform. So, it has been a dire need to analyze these Bangla texts from the social sites. Nevertheless, there is scarcity of Bangla text data with proper label. Most work of finding degree of toxicity has been done on english comments. So it is important to find degree of toxicity of bangla comments.

1.3 Project Goals

The main goal of this project is to identify the degree of toxicity of bangla toxic language. To keep the environment clean, there needs a regulation over online conversation. This project might help to keep the decent regulation. The dataset used in this project contains utterances from the user-generated comments of Facebook. After applying machine learning model user can analyse bangla toxic language in Facebook comments. Moreover, the toxicity level of that comments can also be able to identify.

1.4 Adopted Approach

Machine learning approach has been adopted to train the dataset of this project. The dataset has been collected from mendeley data. The data have been extracted from 8 publicly open Facebook pages. This dataset is a curated, de-duplicated, anonymized dataset that is derived from raw comments. The dataset contains 1959 rows with 08 columns and each row represents a toxic bigram with its corresponding features such as transcriptions, translation, spelling standards and degree of toxicity. Dataset will be pre- processed. Then machine learning models like Support Vector Machine, Logistic Regression, Multinomial Naive Bayes, Random Forest Classifier and Decision Tree Classifier will be applied. With this approach it will became possible to analyse the degree of toxicity of bangla language.

1.5 Novelty of Work

This project contain a new dataset. The dataset was pulished on 28th april'22[4]. ToxLex or Lexicon of toxic language is a dataset having the aggressive and abusive bad words used in social media. Specifically, this dataset contains utterances from the user-generated comments of Facebook. Most of the projects are done to analyse english toxic language but in this project bangla bangla toxic language will be analysed. Also, level or degree of toxicity of bangla toxic language in facebook comments can be analysed. This project has good accuracy in the applied machine learning model of Support Vector Machine.

1.6 Project Organization

This report is further organized into five chapters including Chapter 2: Literature Review, Chapter 3: Methodology, Chapter 4: Result, Chapter 5: Conclusion. The name of the second chapter is literature review where some related papers to our work is described and compared in a table. In the third chapter entitled methodology where data collection, pre-process and all the applied models has been described. Moreover, the detailed results and comparison of the applied model can be found in chapter 4, entitled 'Result'. The final chapter of this report is entitled 'Conclusion' which contains a summary of the whole project.

Chapter 2

Literature Review

In a paper [5] the reseachers have detect the toxicity on bengali social media comment using supervised models. Dataset was collected from experimented human-annotated public domain dataset available at GitHub. The dataset contains five tags for each Bengali social media comment named toxic, threat, obscene, insult, racism.In the dataset there were 10219, 4255, 5964, 528, 1, 12 and 23600 total Comments, toxic Comments, non-Toxic Comments, longest Comment length (in words), smallest Comment Length (in words), Average Comment Length (in words) - 12 Unique and words respectively.Performed punctuation and emoticons removal before doing the tokenization as they do not convey any meaning for toxicity. Long Short Term Memory (LSTM), Convolutional Neural Network (CNN) : 1D Convolutional Layer, Naive Bayes (NB), Support Vector Machines (SVM) and Logistic Regression (LR) models have been used. The accuracy of naive Bayes, Support Vector Machines, Logistic Regression, LSTM and CNN are 81.80, 84.73, 85.22, 94.13 and 95.30 respectively. CNN model provides the highest Accuracy.

In paper [6] the reseachers have detect cyberbullying using Deep Neural Network from social media comments in bangla language. The dataset that has been used in this work contains comments from the interaction section under posts given by actors, social media influencers, singers, politicians, sportsmen that can be viewed publicly on the Facebook platform . The total number of comments collected is 44001. After collecting the comments removed bad characters, punctuations, etc. from the raw data and pre-processed the collected information in order to feed it to the neural network. Pre-processing steps are in three parts: Stop words Removal, Tokenization of String and Padded sequence conversion. Calculated vectors representing each word in multidimensional space known as Word embedding. Applied model architecture are Binary Classification, Multiclass Classification, Supervised machine learning algorithms such as: Random Forest, SVM, KNN, Naïve Bayes etc. classifiers.The accuracy of Random Forest, SVM, KNN, Naive Bayes and Binary are 84, 85, 84, 79 and 87.91 respectively. Binary prodives the highest accuracy.

In paper [7] the reseachers detect bengali hate speech in public Facebook pages. Data have been collected comments from public pages by using an open-source program called FacePager. In the end, the dataset contain 10,133 comments.Pre processing have been done such as removal of bad characters, punctuation, Tokenization and stemming. Moreover, the technique of reducing inflection in words to their root forms such as mapping a group of words to the same stem, Stopwords removal, Word embedding are also done. Applied model architecture are Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K- Nearest Neighbour (KNN), Convolutional Neural Network (CNN), Multilayer Perceptron (MLP), Long Short Term Memory (LSTM), Bernoulli Na¨ıve Bayes (BNB) default parameters, Gaussian Na¨ıve Bayes (GNB) default parameters and Logistic Regression (LR) default parameters. Support Vector Machine (SVM) Kernel is ”rbf” and ”linear”, In Random Forest (RF) Tree count is 81. In Decision Tree (DT) criterion is 'entropy' and max depth is 5 In K- Nearest Neighbour (KNN) n neighbors is 14, metric is 'minkowski' p is 2. In CNN kernel size is 7, loss is binary crossentropy, activation is 'relu', optimizer is adam epoch is 22 In Multilayer Perceptron (MLP) learning rate is 0.002, optimizer is 'adam', epoch is 8, loss is binary crossentropy, batch size is 3 and activation is 'relu' Long Short Term Memory (LSTM) is epoch = 100, learning rate is 0.01, batch size is 125, loss is sparse categorical crossentropy units = 64. SVM has highset accuracy of 87%.

In paper [8] the reseachers used bangla text dataset and exploratory analysis for online harassment detection. The data that has been made accessible in this paper has been gathered and marked from the comments of people in public posts by celebrities, government officials, athletes on Facebook. The total amount of collected comments is 44001. The dataset is compiled with the aim of developing the ability of machines to differentiate whether a comment is a bully expression or not with the help of Natural Language Processing and to what extent it is improper if it is an inappropriate comment. The comments in Bengali language were filtered out from other English or mixed comments. The comments were also checked for duplicates. After getting rid of the duplicates, the comments were labelled into five categories where the bully category has four subcategories. The distribution and number of data in each category provide a good source of learning a good machine learning model to detect different kind of online harassment in Bangla language.

In paper [9] the reseachers used Machine Learning and Deep Learning approach for bangla toxic comment classification. Data was collected from Facebook. In pre processing features are divided, labeled and converted labels as array then from feature removed Punctuation. Then tokenized feature or comment text by CountVectorizer and removed Bangla stop words.Different machine learning classifiers such as CNN, SVM, LSTM, KNN, LDA and NB are used. But Back propagation Multi Label Learning gives the highest accuracy of 60%.

In paper [10] the researcher used machine learning methods for toxic comment classification. The data set is collected from Jigsaw's data set hosted on Kaggle for Toxic Comment Classification Challenge competition. Data set contains a large number of Wikipedia comments which have been labelled by human raters for toxic behaviour. The most used and effective methods are different deep neural networks, but often simpler and faster methods such as a logistic regression were used for baseline approaches.

In paper [11] the researchers used Machine Learning to Detect cyberbully. The data used for this project was collected from the website Formspring.me which is a question-and-answer formatted website that contains a high percentage of bullying content. The data was labeled using a web service, Amazon's Mechanical Turk. The data was labeled in conjunction with machine learning techniques provided by the Weka tool kit. Different algorithms are used such as J48, JRIP, IBK, AND SMO. Both decision tree learner and an instance-based learner were able to identify the true positives with 78.5% accuracy.

Chapter 3

Methodology

A comprehensive description of the process of data collection, data pre-processing and appiled models for data training has been presented in this section.

3.1 Data Collection

Data is collected from Mendeley Data [4]. ToxLex or Lexicon of toxic language is a dataset having the aggressive and abusive bad words used in social media, Specifically, this dataset contains utterances from the user-generated comments of Facebook. The texts cover the demographic and thematic distribution of Bangla's toxic language on social media. The data have been extracted from 8 publicly open Facebook pages. This dataset is a curated, de-duplicated, anonymized dataset that is derived from raw comments. The dataset contains 1959 rows with 08 columns and each row represents a toxic bigram with its corresponding features such as transcriptions, translation, spelling standards, and degree of toxicity. This dataset is single human-annotated and curated to define classifiers for toxic language detection systems. Apart from this, it is considered a wordlist having Bangla cyberbullying, hate speech, and slang.

3.2 Pre-processing

3.2.1 Dataset cleaning

Among 1959 words in the dataset,there were 3 duplicate values. After dropping that 3 dupilcate values there are 1956 words. There are no mising value present in the dataset. 6 columns such as 'ID', 'Base bigram', 'Meaning (Approx.)', 'Transcription (IPA)', 'Themetic category', 'Token Occerance', 'Unusual Spelling ', 'Degree of toxicity' are also dropped. Moreover, outliers have been checked.

3.2.2 Dataset Description

After cleaning dataset contains 1956 records in total. The distribution of the total 1956 toxic language on five categories are shown on the table 3.1.

Classes	**Toxic words**
Mid	1350
Extreme	314
High	295

Table 3.1: Categories of different Degree of Toxicity

From Table 3.1, it's seen that Extreme and High class contain almost simillar number of words while Mid contain more number of words.

For easy understanding and visualization division of class shown in figure 3.1 using a graph.

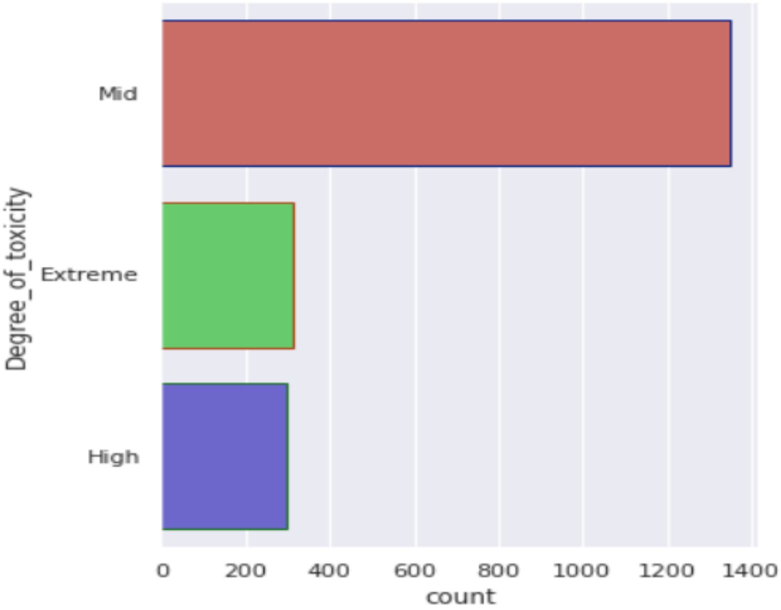


Figure 3.1: Class records in each class

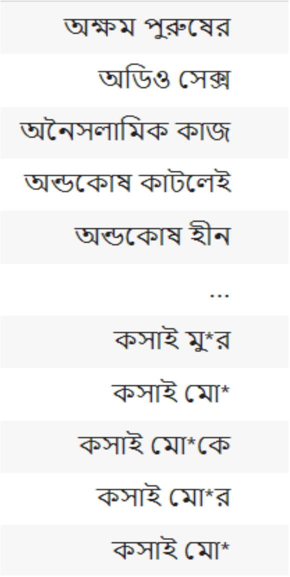
From the Figure 3.1, it is seen thatthe dataset is not that much balanced dataset since Mid class has highest amount of words but still good accuracy has been gained.

3.2.3 Data Pre-processing

Before using the data in the classification models, the raw data needs to be transformed into an understandable format. This is known as the data preprocessing techniques under natural language processing (NLP).

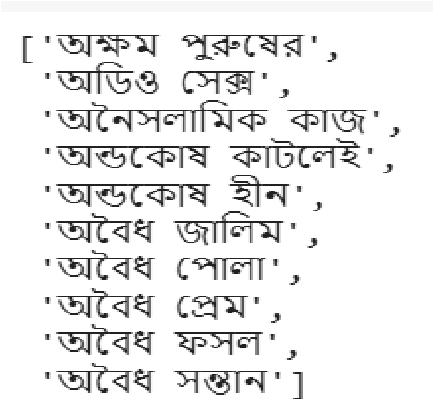
Removing punctuations, stop words removal and tokenization has been implemented on the raw data as a part of the preprocessing step.

In the beginning, the punctuations, white space, digits, and other unnecessary characters have been removed. To show the difference before and after removing those from the data some words are shown in Figure 3.2 and 3.3 respectively.



```
অক্ষম পুরুষের
অডিও সেক্স
অনৈসলামিক কাজ
অন্ডকোষ কাটলেই
অন্ডকোষ হীন
...
কসাই মু*র
কসাই মো*
কসাই মো*কে
কসাই মো*র
কসাই মো*
```

Figure 3.2: Before preprocessing the data



```
['অক্ষম পুরুষের',
 'অডিও সেক্স',
 'অনৈসলামিক কাজ',
 'অন্ডকোষ কাটলেই',
 'অন্ডকোষ হীন',
 'অবৈধ জালিম',
 'অবৈধ পোলা',
 'অবৈধ প্রেম',
 'অবৈধ ফসল',
 'অবৈধ সন্তান']
```

Figure 3.3: After preprocessing the data

From Figures 3.2 and 3.3, it's seen that punctuation and others have been removed from the data.

Then tokenization has been applied, which is one of the primary tasks in NLP. In this process, the texts are broken down into smaller units referred to as tokens. Tokenization helps to analyze each word as an independent unit.

The next step of preprocessing the data is stemming. It is a process in NLP that helps generate the root word by removing the prefixes and suffixes of the word.

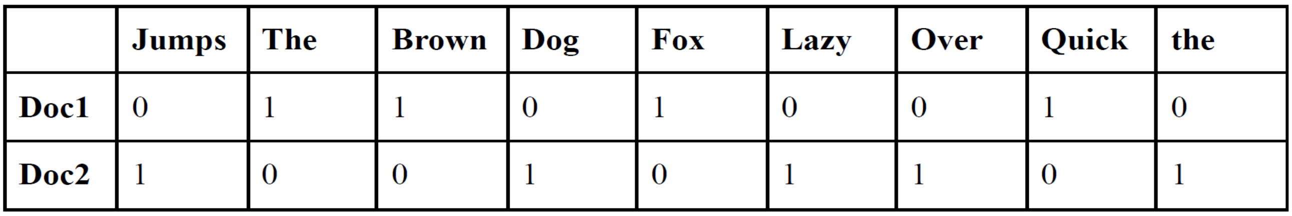
Finally, stop words removal is the procedure where the essential words are extracted by removing the irrelevant words. There are many Bengali words that fall under the category of stop words.

3.2.4 Feature Extraction

After the preprocessing of data, the text needs to be encoded into a matrix or vector of features to be used for modeling. This is known as feature extraction in NLP. Machine learning algorithms cannot process raw text data directly; hence, one of the well-known feature extraction techniques called CountVectorizer has been applied to the preprocessed data before fitting into the machine learning models.

CountVectorizer

CountVectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. The value of each cell is nothing but the count of the word in that particular text sample. A sample matrix has shown in Figure 3.4 exhaustively.



	Jumps	The	Brown	Dog	Fox	Lazy	Over	Quick	the
Doc1	0	1	1	0	1	0	0	1	0
Doc2	1	0	0	1	0	1	1	0	1

Figure 3.4: CountVectorizer

CountVectorizer has been used for both bangla and english language respectively. CountVectorizer for english language has been implemented from sklearn. Again, CountVectorizer for bangla language has been implemented from from a popular python pre-trained module known as BnVec. It is an open-source library for the Bangla word extraction system.

3.3 Applied Models

After the preprocessing of data, then the data were used for training and classification. Datasets have been splitted into 80 percent for training and 20 percent for testing the model. Machine learning models are selected for dataset training. All our implementation has been carried out using the Google Colab in Python language. For each classification model, the results were analyzed and evaluated to compare the performance of different classifiers. It has been discussed further in the Result Analysis section. Finally, the model prediction on the dataset and manual testing was made using the best classification model.

The five machine learning classifiers that have been implemented as follows:

1. Decision Tree
2. Logistic Regression
3. Support Vector Machine
4. Multinomial Naive Bayes
5. Random Forest

3.3.1 Decision Tree

Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). Types of decision trees are based on the type of target variable we have. We have used a Categorical Variable Decision Tree which has a categorical target variable.

The decision of making strategic splits heavily affects a tree's accuracy. A tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches. Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes. The splitting process results in fully grown trees until the stopping criteria are reached. The steps of Decision Tree Classifier:

1. Using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG) (reduction in uncertainty towards the final decision).
2. In an iterative process, we can then repeat this splitting procedure at each child node until the leaves are pure. This means that the samples at each leaf node all belong to the same class.

3. In practice, we may set a limit on the depth of the tree to prevent overfitting. We compromise on purity here somewhat as the final leaves may still have some impurity.

We have fitted the training data into the Decision Tree model and analyzed its performance.

3.3.2 Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. Logistic Regression is one of the most common and useful classifications in machine learning. It belongs to the group of linear classifiers and is somewhat similar to polynomial and linear regression. Logistic regression is fast and relatively uncomplicated. Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification. In a classification task, the outputs of an algorithm fall into one of various pre-chosen categories. The classification model attempts to predict the output value when given several input variables, placing the example into the correct category. When using logistic regression, a threshold is usually specified that indicates at what value the example will be put into one class vs. the other class.

In our dataset, logistic regression has been applied like the other five machine learning models to evaluate the performance.

3.3.3 Support Vector Machine

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. However, most of the time it is used for solving binary classification problems. SVM can handle missing data's. SVM algorithm follows a certain technique to predict the final result. This is where the terms such as hyperplane, margin and kernel comes in. Therefore, the first task in applying the SVM technique is to plot all the data points on an n-dimensional graph where n is the total number of features and the value of each feature will be a specific co-ordinate in the graph. The main task of SVM is to find the optimal hyperplane that will separate the two class from each other.

In our study, we have used a linear kernel, scale as the value of the parameter gamma and the regularization parameter (C) as 1.0. Then we have fitted the training data into the SVM model and analyzed its performance.

3.3.4 Multinomial Naive Bayes

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 of email or newspaper article. Naïve Bayes works in four steps:

1. Convert the data set into a frequency table
2. Create Likelihood table by finding the probabilities
3. Now, use the Naive Bayesian equation to calculate the posterior probability for each class.
4. The class with the highest posterior probability is the outcome of prediction

Multinomial Naive Bayes' theorem is stated mathematically as the following Equation 3.1.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{3.1}$$

Where:
A,B = events
$P(A|B)$ = probability of A given B is true
$P(B|A)$ = probability of B given A is true
P(A), P(B) = the independent probabilities of A and B

3.3.5 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. It adds additional randomness to the model, while growing the trees. It searches for the best feature among a random subset of features, while splitting a node. It is an example of ensemble learning, in which we combine multiple machine learning algorithms to obtain better predictive performance. A technique known as bagging is used to create an ensemble of trees where multiple training sets are generated with replacement. In the bagging technique, a data set is divided into N samples using randomized sampling. Then, using a single learning algorithm a model is built on all samples. Later, the resultant predictions are combined using voting or averaging in parallel. If there is a high non-linearity complex relationship between dependent independent variables, a tree model will outperform a classical regression method. Random Forest works in four steps:

1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.

In our case, we used random_state=42 and n_estimators=40. Then the training data is fitted into the Random Forest Classification model, and its performance on the dataset has been evaluated.

Chapter 4

Results and Analysis

4.1 Performance Measurement Metric

Confusion matrix, precision, recall and F1-score to measure the performance of our models. Precision and recall are two numbers which together are used to evaluate the performance of classification or information retrieval systems. A perfect classifier has precision and recall both equal to 1. Precision and recall should always be reported together.

4.1.1 Precision

Precision is defined as the fraction of relevant instances among all retrieved instances. The formula to find out the precision, is given in Equation 4.1.

$$Precision = \frac{TP}{TP+FP} \tag{4.1}$$

Where,
TP = The true positive rate, that is the number of instances which are relevant and which the model correctly identified as relevant.
FP = The false positive rate, that is the number of instances which are not relevant but which the model incorrectly identified as relevant.

4.1.2 Recall

Recall, sometimes referred to as ‘sensitivity, is the fraction of retrieved instances among all relevant instances. The formula to find out the recall is given in Equation 4.2.

$$Recall = \frac{TP}{TP+FN} \tag{4.2}$$

Where,
TP = The true positive rate, that is the number of instances which are relevant and which the model correctly identified as relevant.
FN = The false negative rate, that is the number of instances which are relevant and which the model incorrectly identified as not relevant.

4.1.3 F-score

Precision and recall are sometimes combined together into the F-score, if a single numerical measurement of a system's performance is required. The formula to find out the F1-score is given in Equation 4.3.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.3}$$

4.2 Model analysis

This project is completed using machine learning models based on NLP. Natural Language Processing (NLP) is a subfield of machine learning that makes it possible for computers to understand, analyze, manipulate and generate human language. Total five machine learning classifiers that have been implemented shown in figure 4.1

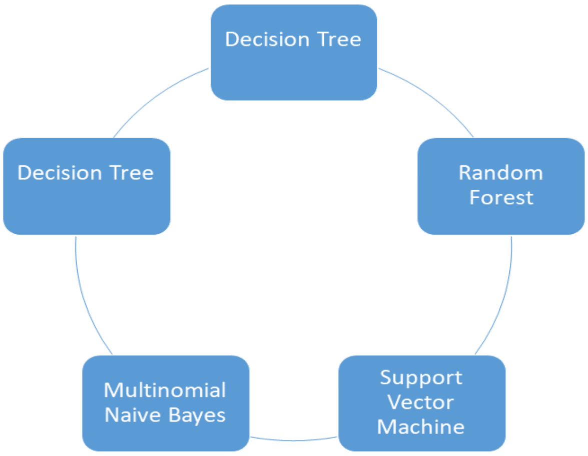
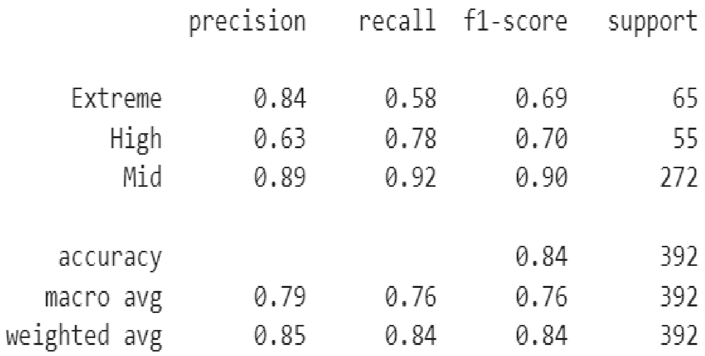


Figure 4.1: Implemented Models

4.2.1 Decision Tree(DT)

Accuracy obtained from Decision Tree model is 84.18% when run with BnVec CountVectorizer and 67.86% when run with sklearn CountVectorizer. It doesn't performs well then some other models. It's classification report is shown in Figure 4.2



```
              precision    recall  f1-score   support

     Extreme       0.84      0.58      0.69        65
        High       0.63      0.78      0.70        55
         Mid       0.89      0.92      0.90       272

    accuracy                           0.84       392
   macro avg       0.79      0.76      0.76       392
weighted avg       0.85      0.84      0.84       392
```

Figure 4.2: Classification Report of Decision Tree

Confusion matrix graph has been plotted for clear idea. A confusion matrix is a table that is used to define the performance of a classification algorithm. Figure 4.3 illustrates the confusion matrix of decision tree algorithm where x axis denotes the predicted class and y axis denotes the true class.It shows that, mild class predicted 249 record correctly and over here number of mistakes or error is much lesser than to other two classes.As a result, higher accuracy is obtained due to less mistakes. Conversely, high and extreme class correctly predicted 43 and 38 records, respectively which is considerately higher. Here, number of mistakes is higher.

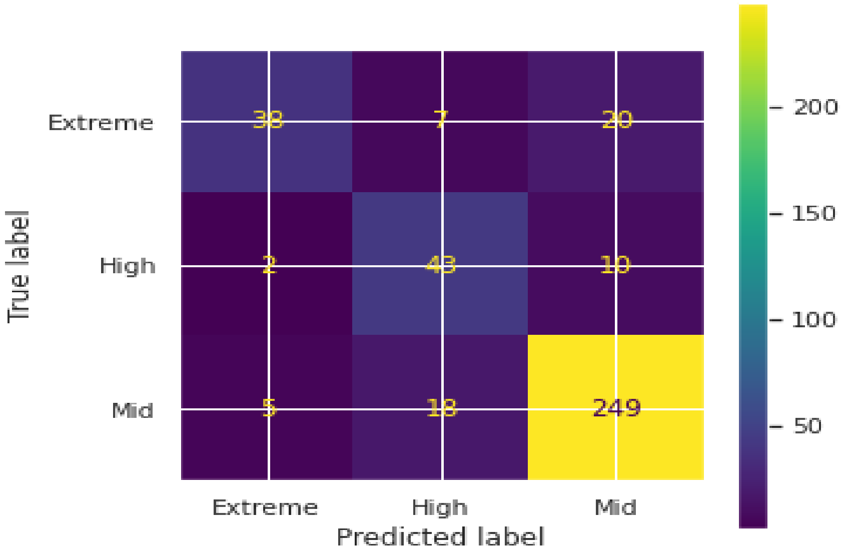
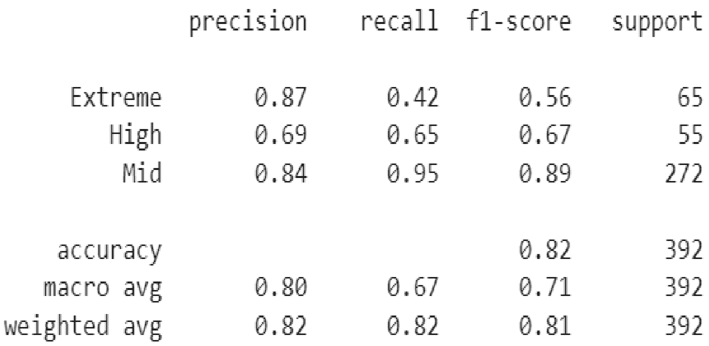


Figure 4.3: Confusion Matrix of Decision Tree

4.2.2 Logistic Regression(LR)

Accuracy obtained from Logistic Regression model is 82.14% when run with BnVec CountVectorizer and 71.68% when run with sklearn CountVectorizer. It performs better then some other models. It's classification report is shown in Figure 4.4



```
              precision    recall  f1-score   support

     Extreme       0.87      0.42      0.56        65
        High       0.69      0.65      0.67        55
         Mid       0.84      0.95      0.89       272

    accuracy                           0.82       392
   macro avg       0.80      0.67      0.71       392
weighted avg       0.82      0.82      0.81       392
```

Figure 4.4: Classification Report of Logistic Regression

Figure 4.5 illustrates the confusion matrix of Logistic Regression(LG) algorithm where x axis denotes the predicted class and y axis denotes the true class. Like decision tree algorithm, mild class of logistic regression algorithm predicted the highest records. It predicted 259 record correctly and over here number of mistakes or error is much lesser than to other two classes. As a result, higher accuracy is obtained due to less mistakes along with higher precision. Conversely, high and extreme class correctly predicted 36 and 27 records, respectively which is considerately higher. Here, number of mistakes is higher.

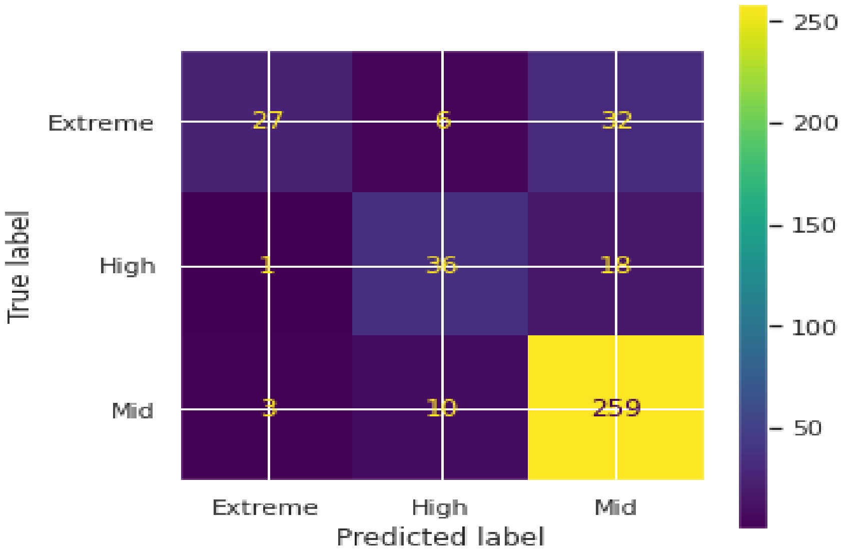
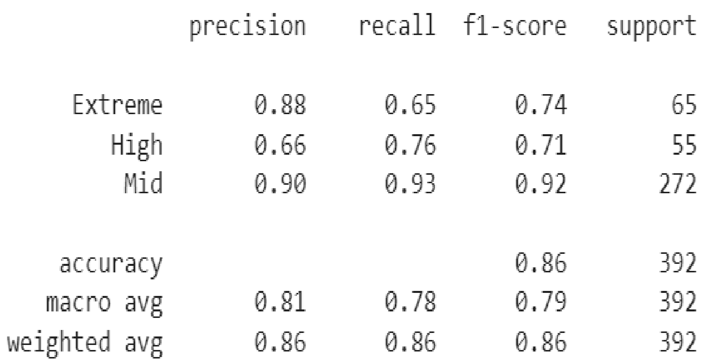


Figure 4.5: Confusion Matrix of Logistic Regression

4.2.3 Support Vector Machine(SVM)

Accuracy obtained from Support Vector Machine model is 85.97% when run with BnVec CountVectorizer and 70.41% when run with sklearn CountVectorizer. It performs best then other models. It's classification report is shown in Figure 4.6



```
              precision    recall  f1-score   support

     Extreme       0.88      0.65      0.74        65
        High       0.66      0.76      0.71        55
         Mid       0.90      0.93      0.92       272

    accuracy                           0.86       392
   macro avg       0.81      0.78      0.79       392
weighted avg       0.86      0.86      0.86       392
```

Figure 4.6: Classification Report of Support Vector Machine

Figure 4.7 illustrates the confusion matrix of Support Vector Machine(SVM) algorithm where x axis denotes the predicted class and y axis denotes the true class. Here, mild class of logistic regression algorithm predicted the highest records. It predicted 253 record correctly and over here number of mistakes or error is much lesser than to other two classes. As a result, higher accuracy is obtained due to less mistakes along with higher precision. Conversely, high and extreme class correctly predicted 42 and 42 records, respectively which is considerately higher. Here, number of mistakes is higher.

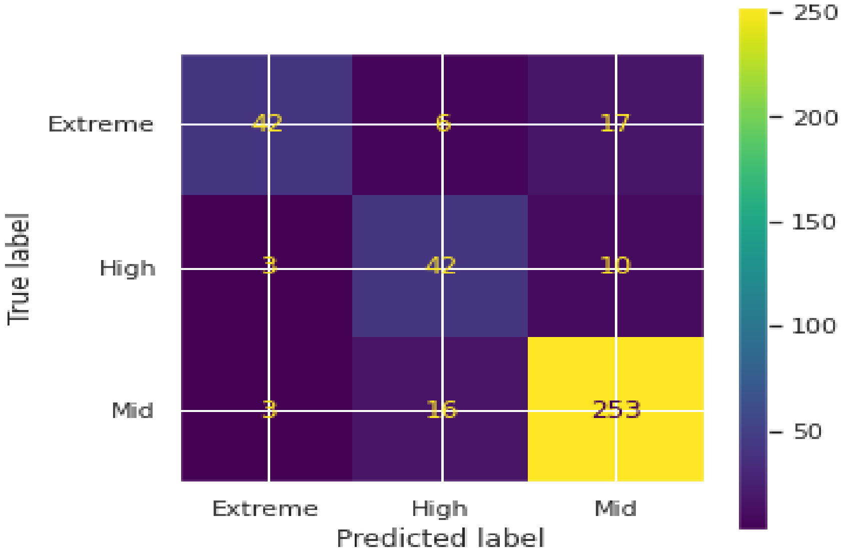
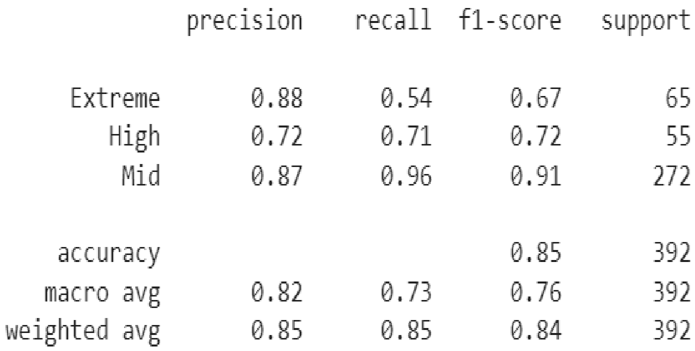


Figure 4.7: Confusion Matrix of Support Vector Machine

4.2.4 Multinomial Naive Bayes(NB)

Accuracy obtained from Multinominal Naive Bayes model is 85.2% when run with BnVec CountVectorizer and 70.41% when run with sklearn CountVectorizer. It performs better then other models. It's classification report is shown in Figure 4.8



```
              precision    recall  f1-score   support

     Extreme       0.88      0.54      0.67        65
        High       0.72      0.71      0.72        55
         Mid       0.87      0.96      0.91       272

    accuracy                           0.85       392
   macro avg       0.82      0.73      0.76       392
weighted avg       0.85      0.85      0.84       392
```

Figure 4.8: Classification Report of Multinomial Naive Bayes

Figure 4.9 illustrates the confusion matrix of Multinomial Naive Bayes(NB) algorithm where x axis denotes the predicted class and y axis denotes the true class. Here, mild class of logistic regression algorithm predicted the highest records. It predicted 260 record correctly and over here number of mistakes or error is much lesser than to other two classes. As a result, higher accuracy is obtained due to less mistakes along with higher precision. Conversely, high and extreme class correctly predicted 39 and 35 records, respectively which is considerately higher. Here, number of mistakes is higher.

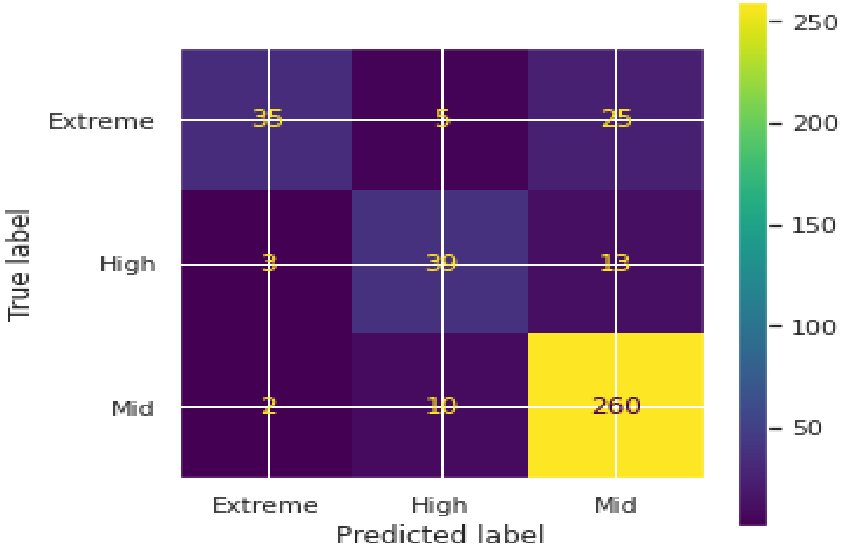
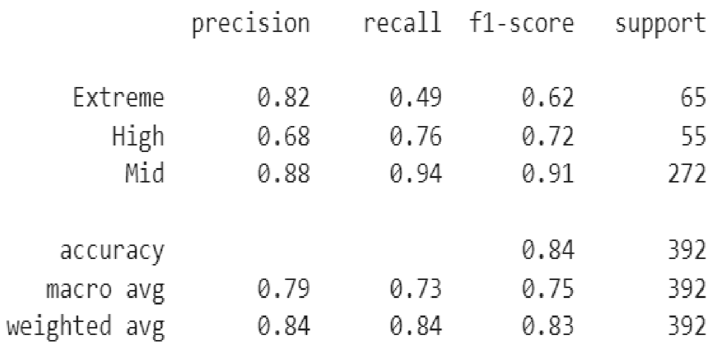


Figure 4.9: Confusion Matrix of Multinomial Naive Bayes

4.2.5 Random Forest(RF)

Accuracy obtained from Multinominal Naive Bayes model is 83.9% when run with BnVec CountVectorizer and 70.41% when run with sklearn CountVectorizer. It performs better then other models. It's classification report is shown in Figure 4.10



```
              precision    recall  f1-score   support

     Extreme       0.82      0.49      0.62        65
        High       0.68      0.76      0.72        55
         Mid       0.88      0.94      0.91       272

    accuracy                           0.84       392
   macro avg       0.79      0.73      0.75       392
weighted avg       0.84      0.84      0.83       392
```

Figure 4.10: Classification Report of Random Forest

Figure 4.11 illustrates the confusion matrix of Random Forest(RF) algorithm where x axis denotes the predicted class and y axis denotes the true class. Here, mild class of logistic regression algorithm predicted the highest records. It predicted 255 records correctly and over here number of mistakes or error is much lesser than to other two classes. As a result, higher accuracy is obtained due to less mistakes along with higher precision. Conversely, high and extreme class correctly predicted 42 and 32 records, respectively which is considerately higher. Here, number of mistakes is higher.

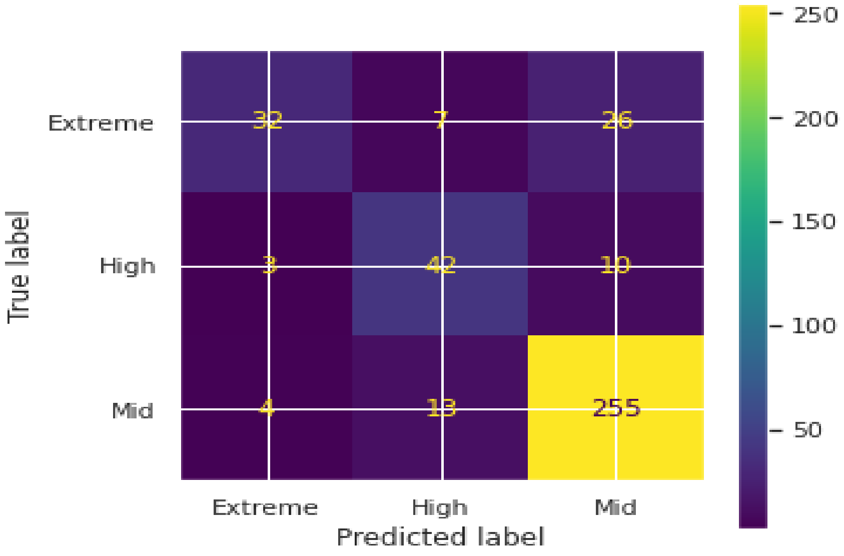


Figure 4.11: Confusion Matrix of Random Forest

4.3 CountVectorizer

After preprocessing, Countvectorizer have been applied. Then, training data is divided into the five types of machine learning classifier to evaluate the result of different classifier. The the results of those classifiers are described and analysed exhaustively in this section. CountVectorizer has been used for both Bangla and English Language. For bangla BnVec has been used and for english sklearn has been used. Models providing different accuracy with CountVectorizer import from sklearn and BnVec has been shown in chart 4.12

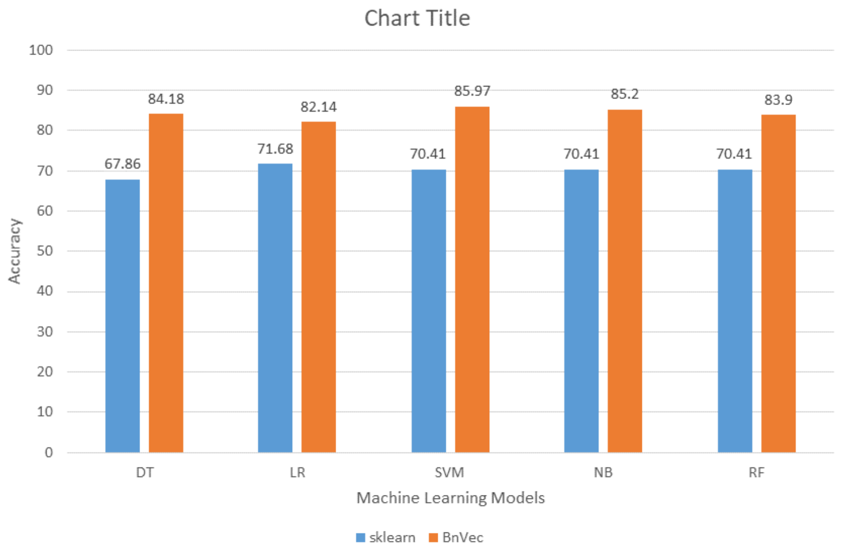


Figure 4.12: Difference of Accuracy between Models implemented with ConuntVectorizer import from sklear and BnVec

In figure 4.12 comparison among all the accuracy of the applied models run with CountVectorizer import from sklearn and BnVec has been shown. Blue bar is CountVectorizer import from sklearn and orange bar is CountVectorizer import from BnVec. Orange bar is taller than blue bar. So BnVec provides better accuracy than sklearn. Moreover, it is seen that Support Vector Machine(SVM) has the higest accuracy and Logistic Regression(LR) has the lowest accuracy when CountVectorizer is import from BnVec. On the other hand, Logistic Regression(LR) has the higest accuracy and Decision Tree(DT) has the lowest accuracy when CountVectorizer is import from sklearn.

4.4 Comparison of Applied Models with Related Works

In this project degree or level of toxicity of bangla words is detected. In paper [5] also toxicity on bangla language is detected using supervised model. In that paper amount of data is 23600 and in this project number of data is 1959. Comparison among the accuracy of some models applied in this project and in paper [5] is shown in the table 4.1

Model Name	Applied Model Accuracy	Related Works Model Accuracy
Naive Bayes	85.2%	81.80%
SVM	85.97%	84.73%
Logistic Regression	82.14%	85.22%

Table 4.1: Comparison of Applied Models with Related Works

From the table 4.1 it is seen that accuracy of Naive Bayes and SVM(Support Vector Machine) models in this project are better than the other paper or related works.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

A system is build to analyse bangla toxic language using machine learning. It is going to find out the degree or level of toxicity of bangla toxic comments on Facebook or online sites.

Dataset collected from Mendeley Data contain 1958 toxic comment which have been extracted from 8 publicly open Facebook pages. Dataset classified into 3 class mid, high and extreme. Countvectorizer has been imported from sklearn and BnVec vectorizer for extracting features have been used. Accuracy of the applied models are better for BnVec vectorizer. After prepocessing such as removing whitespace, punctuation and Tokenization different machine learning models are applied. The machine learning models are Decision Tree, Logistic Regression, Support Vector Machine, Multinominal Naive Bayes and Random Forest. Highest accuracy is 85.97% which is obtained from Support Vector Machine with CountVectorizer import from BnVec.

5.2 Future work

In future, model can be more accurate.Other feature extractors e.g. word2vec, GloVe and fastText and deep learning model e.g. ANN, CNN can be applied to compare how well it performs with these new approach. The model can be deployed for better understanding of the users. Thus, this project can be given a good interface.

References

[1] Babbel.com and L. N. GmbH, “The 10 most spoken languages in the world.” [Online]. Available: https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world

[2] “How toxic online comments are unhealthy and cyclical,” Apr 2021. [Online]. Available: https://www.govtech.com/opinion/how-toxic-online-comments-are-unhealthy-and-cyclical.html

[3] A. Jahin, “The real and intangible threat of online child harassment,” Mar 2021. [Online]. Available: https://www.thedailystar.net/opinion/news/the-real-and-intangible-threat-online-child-harassment-2059949

[4] M. M. O. Rashid, “Toxlex_bn: A curated dataset of bangla toxic language derived from facebook comment,” *Data in Brief*, vol. 43, p. 108416, 2022.

[5] N. Banik and M. H. H. Rahman, “Toxicity detection on bengali social media comments using supervised models,” in *2019 2nd International Conference on Innovation in Engineering and Technology (ICIET)*, 2019, pp. 1–5.

[6] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, “Cyberbullying detection using deep neural network from social media comments in bangla language,” 2021. [Online]. Available: https://arxiv.org/abs/2106.04506

[7] N. I. Remon, N. H. Tuli, and R. D. Akash, “Bengali hate speech detection in public facebook pages,” in *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, 2022, pp. 169–173.

[8] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, “Bangla text dataset and exploratory analysis for online harassment detection,” 2021. [Online]. Available: https://arxiv.org/abs/2102.02478

[9] A. Jubaer, A. Sayem, and M. A. Rahman, “Bangla toxic comment classification (machine learning and deep learning approach),” in *2019 8th International*

Conference System Modeling and Advancement in Research Trends (SMART), 2019, pp. 62–66.

[10] D. Androcec, “Machine learning methods for toxic comment classification: a systematic review,” *Acta Universitatis Sapientiae, Informatica*, vol. 12, pp. 205–216, 12 2020.

[11] K. Reynolds, A. Kontostathis, and L. Edwards, “Using machine learning to detect cyberbullying,” in *2011 10th International Conference on Machine Learning and Applications and Workshops*, vol. 2, 2011, pp. 241–244.