Toxicity Detection From Tweets Using Machine Learning Approaches

Md. Kamrozzaman Bhuiyan	180104063
Rahat Chowdhury	180104067
Md. Rakibul Islam	180104072

Project Report

Course ID: CSE 4214

Course Name: Pattern Recognition Lab

Semester: Fall 2021

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September 2022

Toxicity Detection From Tweets Using Machine Learning Approaches

Submitted by

Md. Kamrozzaman Bhuiyan	**180104063**
Rahat Chowdhury	**180104067**
Md. Rakibul Islam	**180104072**

Submitted To

Farzad Ahmed

Sajib Kumar Saha Joy,

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 2022

ABSTRACT

Social interaction is increasing using online social media due to the peak of digitization. Cyberbullying and harassment is also increasing. Twitter is one of the famous social sites. It is used by educated and classy people compared to other social sites like Facebook. Twitter is not yet free from Cyberbullying. These incidents are drastically rising day by day. It is going to be a new trend to humiliate others by using abusive words in tweets and comments. We are going to detect toxicity from tweets. People can be aware of toxic tweets and be more sincere with their words knowing the toxicity level of their tweets. We hope to keep impact in decreasing the number of instances of toxic tweets by awaring people.

Contents

List of Figures

Chapter 1

Introduction

The internet is considered to be one of the great inventions by the people of this modern era. It opened a lot of door of possibilities that no one has ever imagined. We can now communicate with each other and share information very easily that was hectic earlier. Social media sites has grown the most with the advancement of internet. These sites are allowing people to communicate with each other both privately or publicly using internet. The most used platforms considering the number of users are Facebook having 2.449 billion users, YouTube having 2 billion users, WhatsApp having 1.6 billion users, Twitter having 340 million users. These users are from various kinds of age groups, sexuality, religion, ethnicity, culture, heritage, etc. People are allowed to share their thoughts on these platforms, but this right is sometimes abused. The amount of using hate speech has risen substantially on these platforms. It is of various forms like cyberbullying, harassment, abusive language, discrimination, racism, etc. which can cause a huge toll on the mental health of the people who is facing these. Hence, researchers have studied the patterns of hate speech and proposed possible ways to control it. We propose a technique to detect toxicity in tweets and to avail users to use a healthy social site that is free of hate.

Chapter 2

Literature Reviews

[1] used Facebook as a benchmark and considered textual contents of comments aiming at preventing the alarming spread of hate campaigns. They have proposed different types of hate categories to distinguish the kind of hate. They have used two datasets. The three-class dataset composed of 3356 documents divided into No Hate, Weak Hate and Strong Hate. The two-class dataset composed of 3575 documents divided into No Hate and Hate. The documents contain Italian texts. They have used 10-fold cross validation process to evaluate the accuracy of the two hate speech classifiers. They have used two classifiers based on Support Vector Machine (SVM) and Long Short Term Memory (LSTM) and achieved accuracy of about 72.95% and 75.23% respectively.

[2] proposed a two-step method for hate speech detection. They have used paragraph2vec for joint modeling of comments and words. Continuous BOW is used to learn the distributed representations in a joint space. This has resulted in low-dimensional text embedding and caused semantically similar comments and words reside in the same part of the space. To distinguish between hateful and clean comments, they have used the embeddings to train a binary classifier. Evaluation of their approach is done on a large-scale data set of user comments collected on Yahoo Finance website. The dataset composed of 56,280 comments containing hate speech and 895,456 clean comments. They have achieved AUC of about 0.7889 in BOW(tf), 0.6933 in BOW(tf-idf) and 0.8007 in paragraph2vec.

In [3], the authors detected profanity in chat-logs of a popular Multiplayer Online Battle Arena (MOBA) game by using a novel natural language processing framework and developed a method to classify toxic remarks. They have shown how toxicity is non-trivially linked to game success. They have shown that toxicity appears only in 60% of all matches and it is thus too infrequent to be used for predicting match outcome in general. They trained a linear support vector machine (SVM) to predict the winning team on a feature-set based on TF-IDF of each word. They introduced the parameter 't' to control the amount of chat history that is given to the classifier. They have obtained the average accuracy of

94.07% under a 10-fold cross-validation for t = 1.0.

[4] presented an approach to detect hate speech in online text, where hate speech is defined as abusive speech targeting specific group characteristics, such as ethnic origin, religion, gender, or sexual orientation. They have observed that hatred against each different group is typically characterized by the use of a small set of high frequency stereotypical words. However, they think such words may be used in either a positive or a negative sense, making our task similar to that of words sense disambiguation. They described their definition of hate speech, the collection and annotation of their hate speech corpus, and a mechanism for detecting some commonly used methods of evading common "dirty word" filters. They described pilot classification experiments in which they classify antisemitic speech reaching an accuracy 94%, precision of 68% and recall at 60%, for an F1 measure of .6375.

[5] used a supervised learning approach for detecting harassment. They used various dataset from various social platforms. In the dataset the number of negative posts are more. For the classification tool they used libSVM with the linear kernel. And they showed a comparison in performance for N-grams and TF-IDF weighted features. And said that TF-IDF is much more effective than other basic methods for detecting harassment.

[6] used both textual and social network features to detect cyberbullying. They have used a corpus of Twitter messages, which contains 900,000 tweets from a lot of users. The dataset is categorized into two categories like Bully and Non Bully. For classification they have used some known classifier models like classification J48, Naive Bayes, SMO, Bagging and Dagging, ZeroR. And they got better performance for Dagging than other models. Also they have shown comparisons of different models. And they said classifier performance is increasing as they shift 'textual features' models to 'composite' models in terms of both ROC and TP rates.

Chapter 3

Data Collection & Processing

We have collected our dataset from Kaggle. The dataset is titled as "Toxic Tweets Dataset" [7]. The dataset contains two columns such as Tweets and Toxicity(Label). Toxicity label 0 means the tweet is not toxic and 1 means the tweet is toxic. The dataset contains about 56,744 labeled tweets from different users divided into two classes having 32,592 non-toxic tweets and 24,153 toxic tweets.

We have pre-processed our dataset by removing stop words and punctuation. Stemming is also performed as pre-processing.

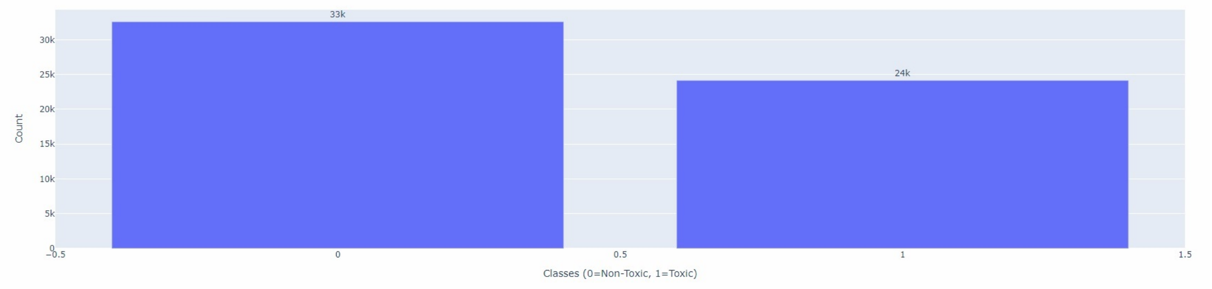


Figure 3.1: Distribution of Data Classes

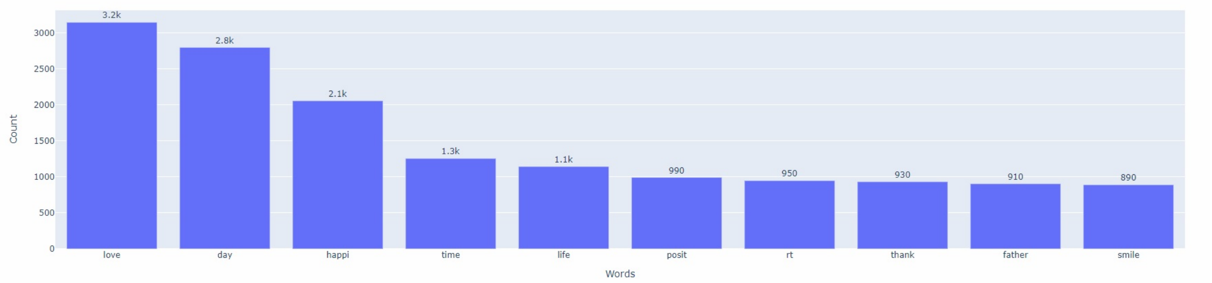


Figure 3.2: Top 10 Words in non-Toxic Tweets

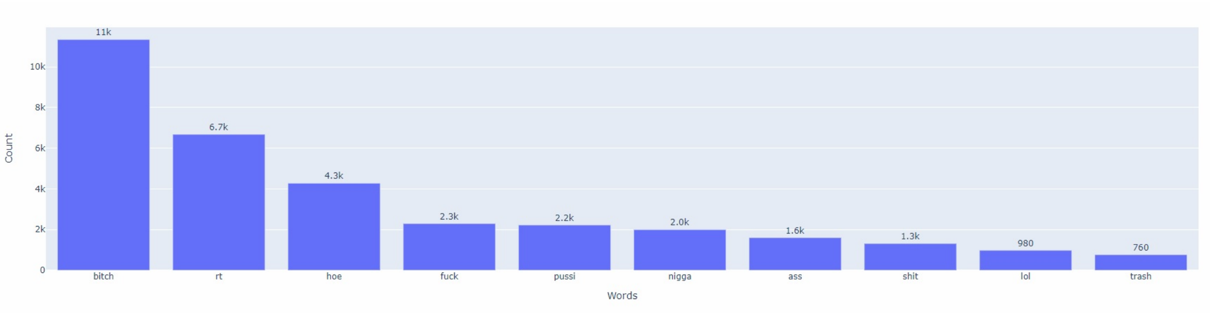


Figure 3.3: Top 10 Words in Toxic Tweets

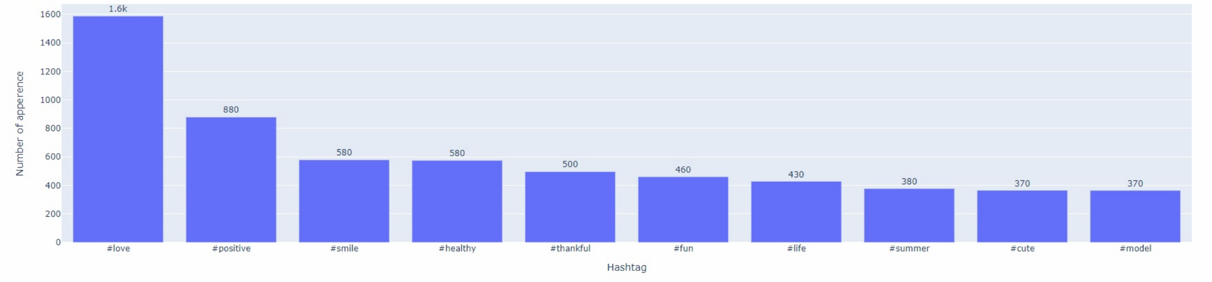


Figure 3.4: Top 10 non-Toxic Hashtags

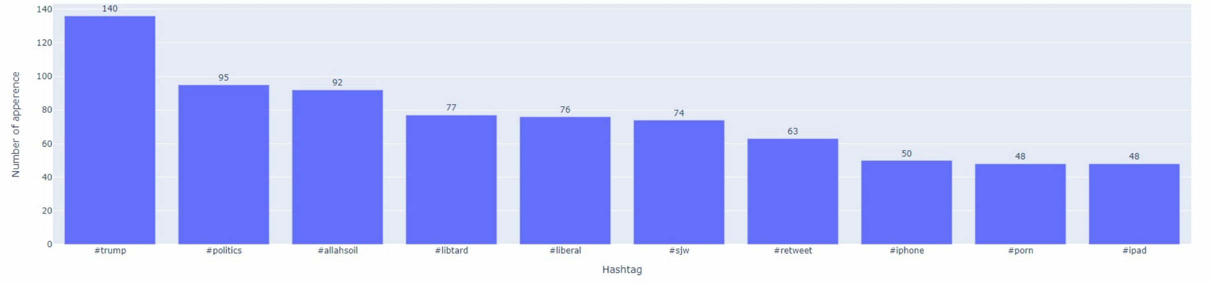


Figure 3.5: Top 10 Toxic Hashtags

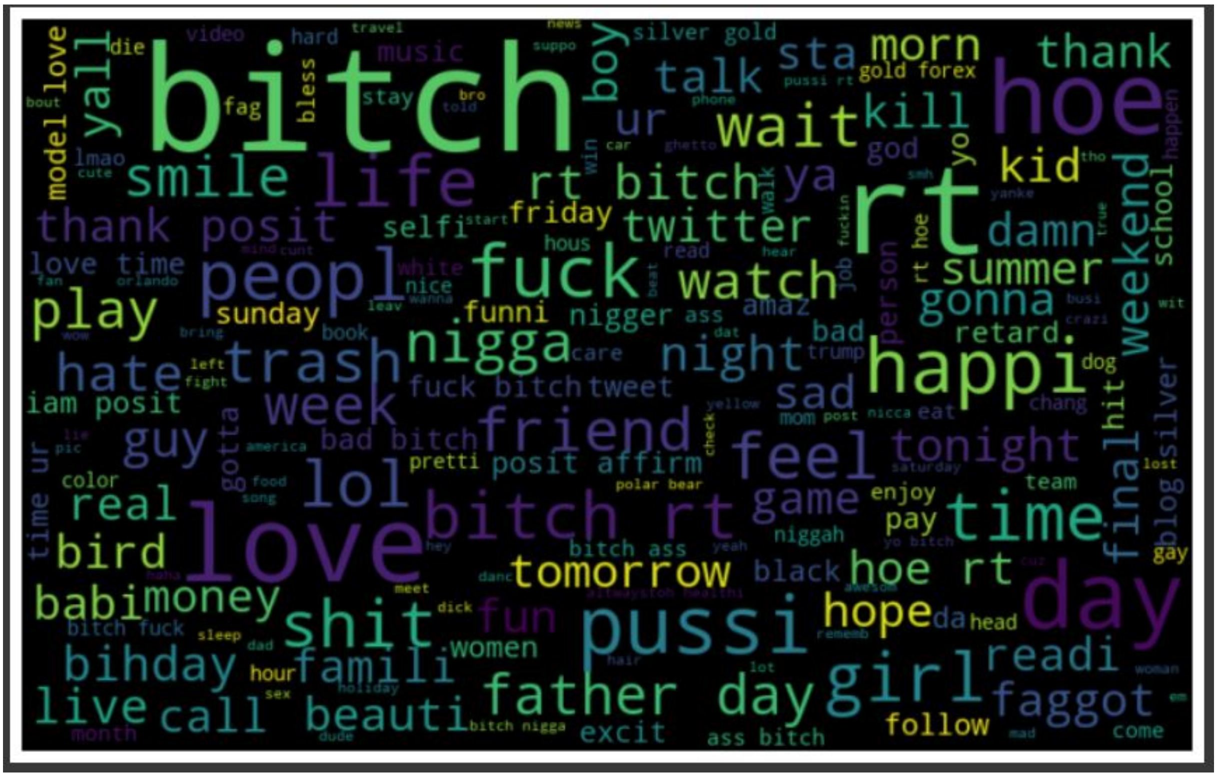


Figure 3.6: Words in Dataset

Chapter 4

Methodology

We have used Naive Bayes Classifier, K-Nearest Neighbour(KNN) Classifier, Logistic Regression, Decision Tree Classifier, AdaBoost Classifier,Random Forest Classifier, Support Vector Classifier (SVC) to classify our tweet data. We have done pre-processing of our dataset like removing stop words, removing punctuation and stemming. We sliced our data into training and test sets containing 70% and 30% of total data. We have extracted features from our dataset using Count Vectorizer and TF-IDF.

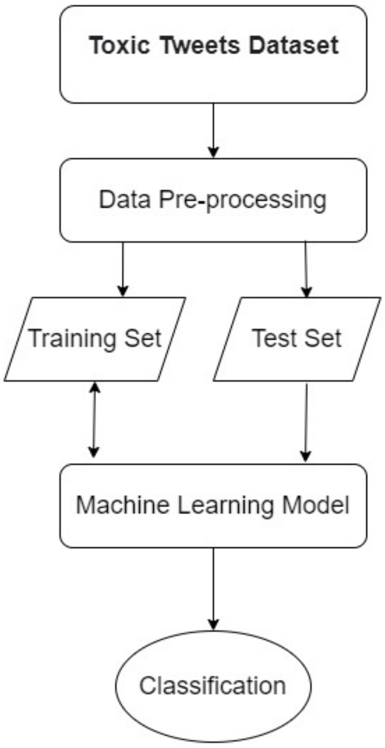


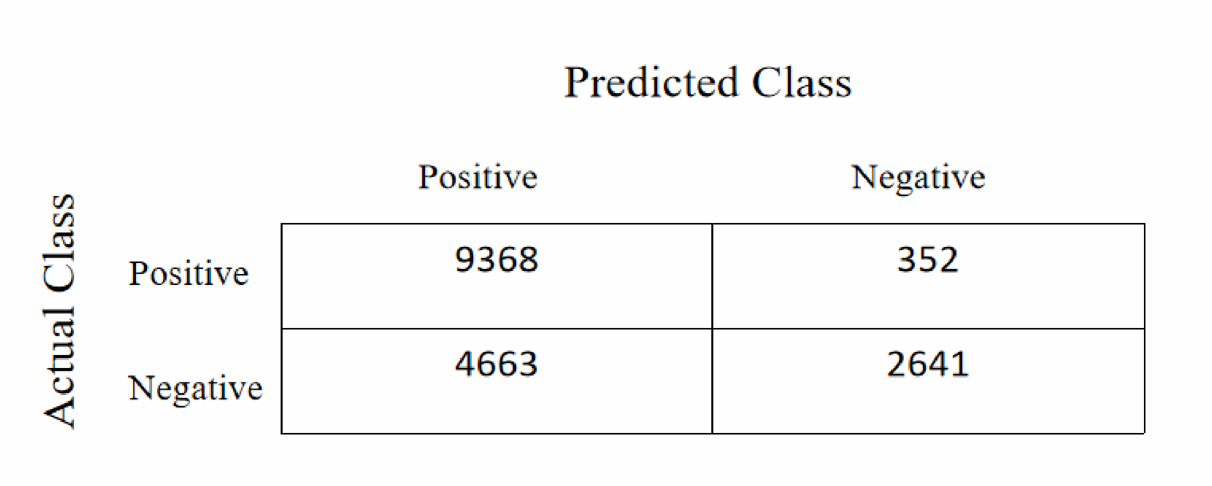
Figure 4.1: Flowchart of Methodology

Chapter 5

Experiments and Results

5.1 K-Nearest Neighbour (KNN)

We have achieved accuracy of about 71% and Mean Absolute Error is 0.29, Mean Squared Error is 0.29, Root Mean Squared Error is 0.54 and R-Squared Error is 0.64.

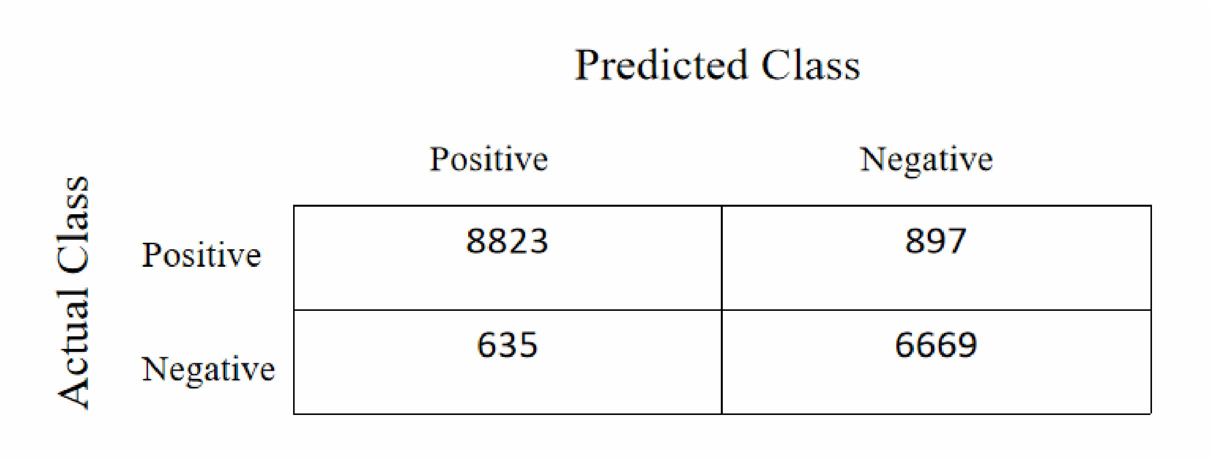


		Predicted Class	
		Positive	Negative
Actual Class	Positive	9368	352
	Negative	4663	2641

Figure 5.1: Confusion Matrix of KNN

5.2 Naive Bayes

We have achieved accuracy of about 91% and Mean Absolute Error is 0.89, Mean Squared Error is 0.89, Root Mean Squared Error is 0.29 and R-Squared Error is 0.88.

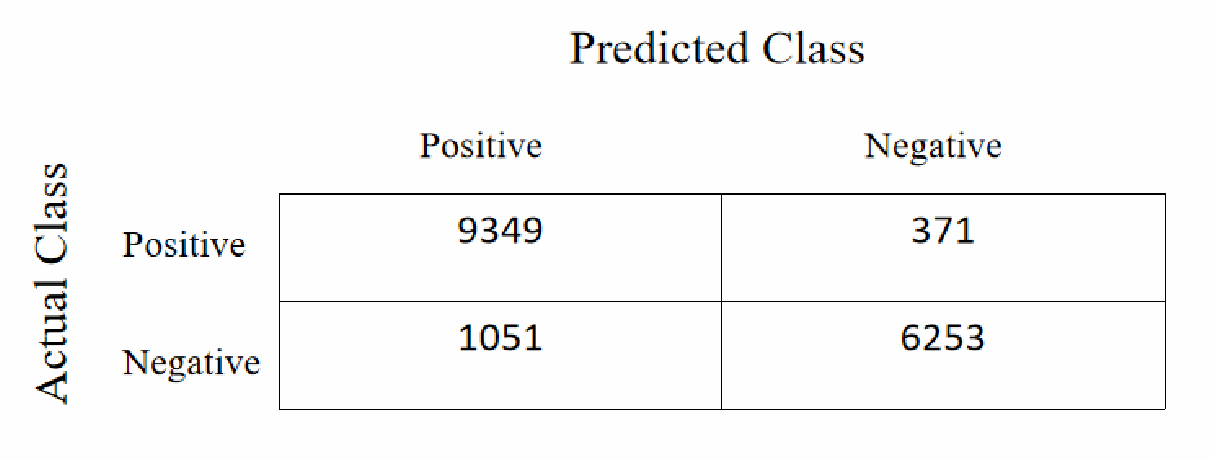


Actual Class	Predicted Class: Positive	Predicted Class: Negative
Positive	8823	897
Negative	635	6669

Figure 5.2: Confusion Matrix of Naive Bayes Classifier

5.3 AdaBoost

We have achieved accuracy of about 92% and Mean Absolute Error is 0.08, Mean Squared Error is 0.08, Root Mean Squared Error is 0.29 and R-Squared Error is 0.89.

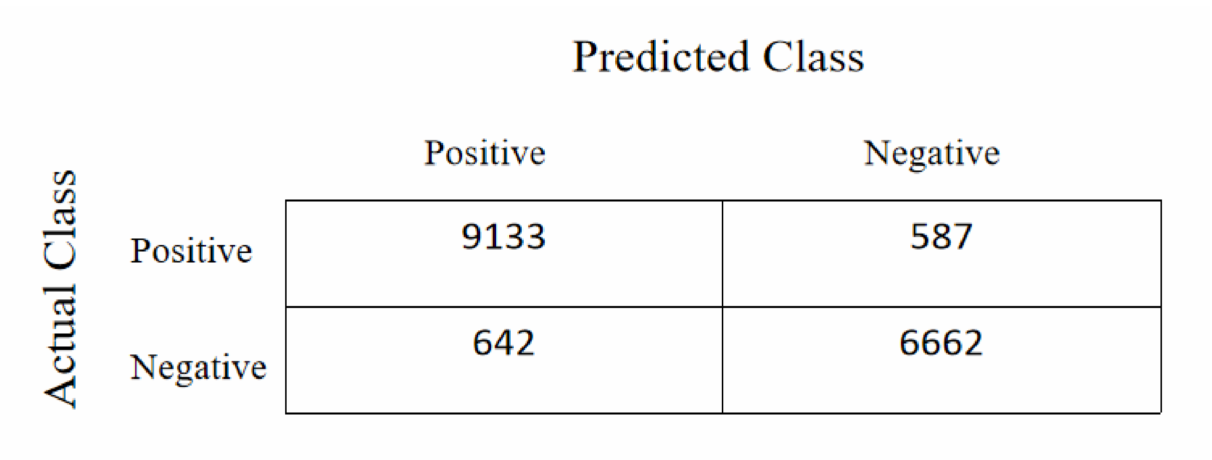


Actual Class	Predicted Class: Positive	Predicted Class: Negative
Positive	9349	371
Negative	1051	6253

Figure 5.3: Confusion Matrix of AdaBoost

5.4 Decision Tree Classifier

We have achieved accuracy of about 93% and Mean Absolute Error is 0.07, Mean Squared Error is 0.07, Root Mean Squared Error is 0.27 and R-Squared Error is 0.91.



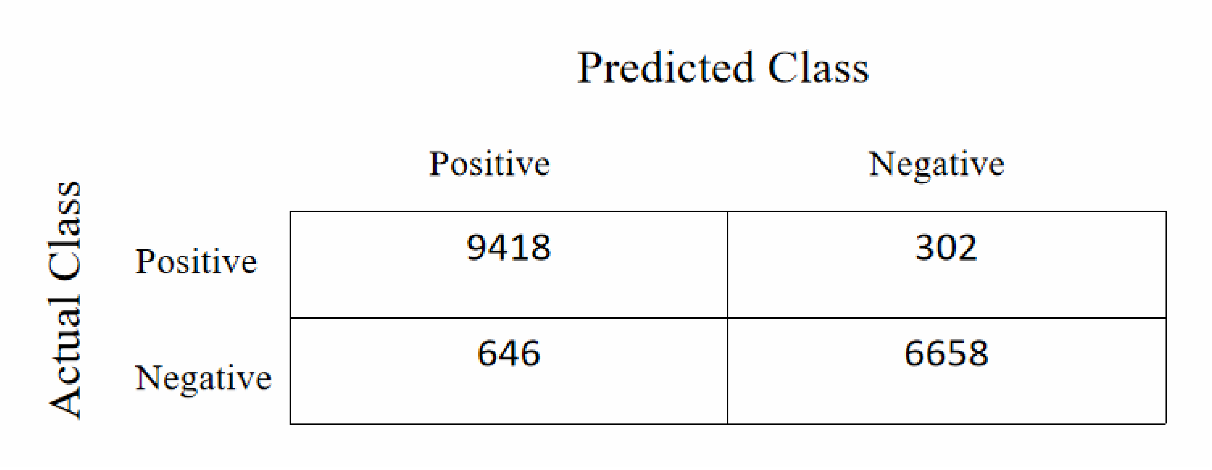
Predicted Class

Actual Class		Positive	Negative
	Positive	9133	587
	Negative	642	6662

Figure 5.4: Confusion Matrix of Decision Tree Classifier

5.5 Support Vector Classifier (SVC)

We have achieved accuracy of about 94% and Mean Absolute Error is 0.06, Mean Squared Error is 0.06, Root Mean Squared Error is 0.24 and R-Squared Error is 0.93.



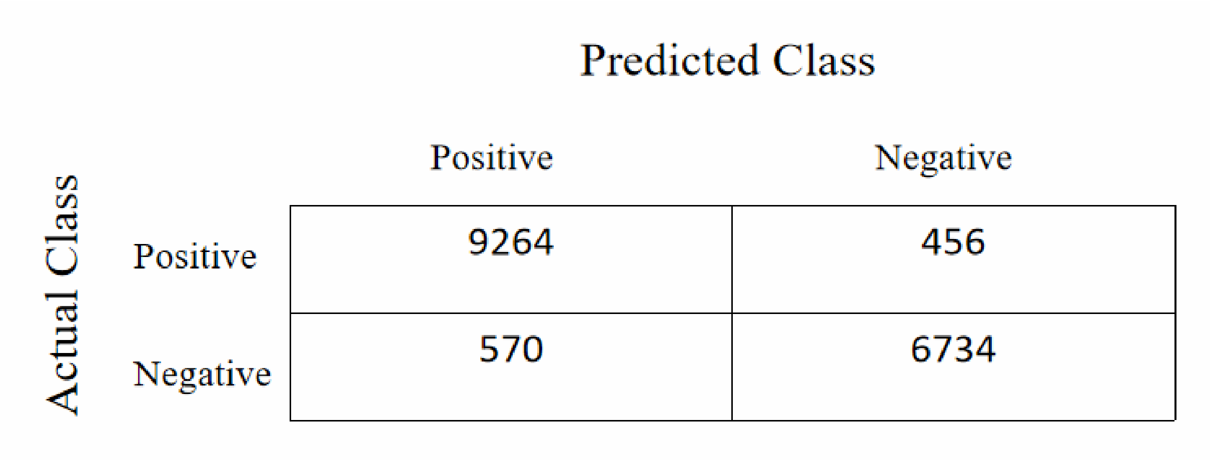
Predicted Class

Actual Class		Positive	Negative
	Positive	9418	302
	Negative	646	6658

Figure 5.5: Confusion Matrix of Support Vector Classifier (SVC)

5.6 Random Forest Classifier

We have achieved accuracy of about 94% and Mean Absolute Error is 0.06, Mean Squared Error is 0.06, Root Mean Squared Error is 0.25 and R-Squared Error is 0.93.

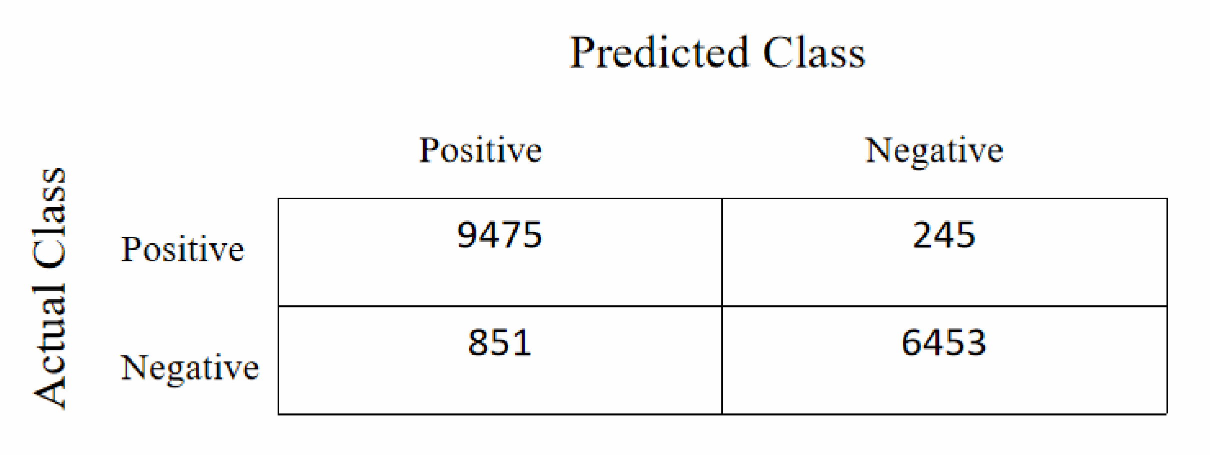


		Predicted Class	
		Positive	Negative
Actual Class	Positive	9264	456
	Negative	570	6734

Figure 5.6: Confusion Matrix of Random Forest Classifier

5.7 Logistic Regression

We have achieved accuracy of about 94% and Mean Absolute Error is 0.06, Mean Squared Error is 0.06, Root Mean Squared Error is 0.25 and R-Squared Error is 0.92.

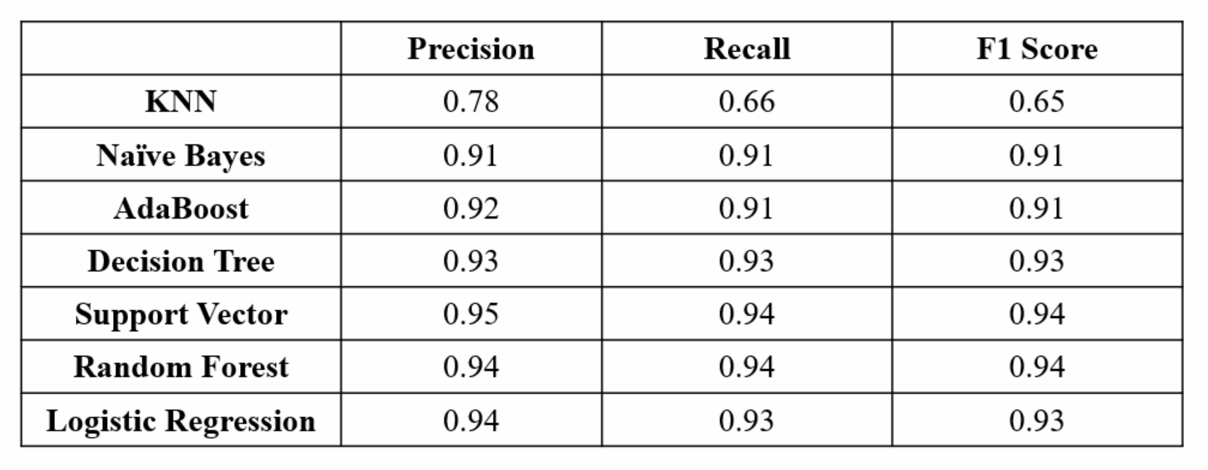


		Predicted Class	
		Positive	Negative
Actual Class	Positive	9475	245
	Negative	851	6453

Figure 5.7: Confusion Matrix of Logistic Regression

Figure 5.8 shows the Precision, Recall and F1 Score of our used models.

Figure 5.9 shows the ROC curve that shows the trade-off between specificity and sensitivity. Curves that are closer to the top-left corner denotes better performance. There are two parameters of ROC curve. These are False Positive Rate and True Positive Rate plotted in the X-axis and Y-axis respectively.



	Precision	Recall	F1 Score
KNN	0.78	0.66	0.65
Naïve Bayes	0.91	0.91	0.91
AdaBoost	0.92	0.91	0.91
Decision Tree	0.93	0.93	0.93
Support Vector	0.95	0.94	0.94
Random Forest	0.94	0.94	0.94
Logistic Regression	0.94	0.93	0.93

Figure 5.8: Precision, Recall and F1 Score

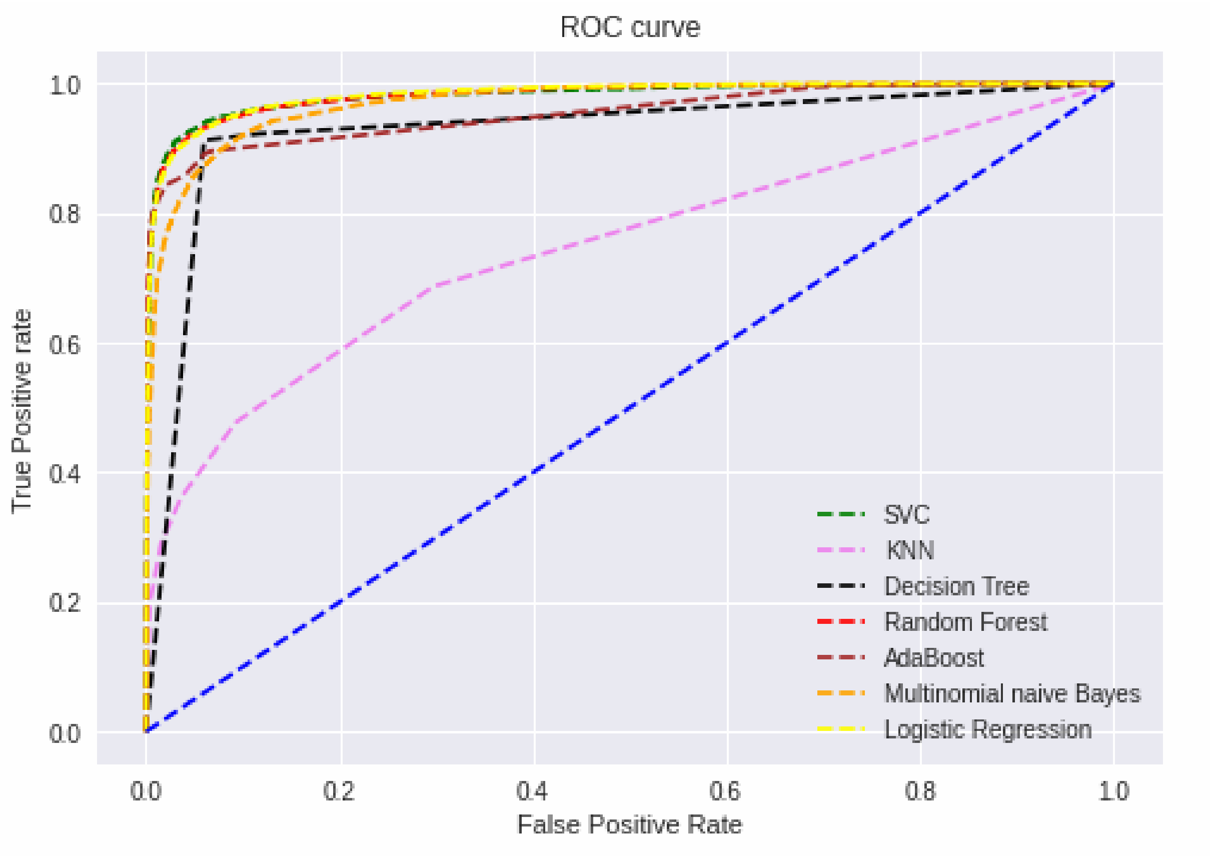


Figure 5.9: ROC Curve

Chapter 6

Future Work and Conclusion

We have used Naive Bayes Classifier, K-Nearest Neighbour(KNN) Classifier, Logistic Regression, Decision Tree Classifier, AdaBoost Classifier,Random Forest Classifier, Support Vector Classifier (SVC) to our data. Among these models Logistic Regression performed better in our data. We have achieved 94% accuracy using Logistic Regression and F1 score is 0.93.

We are keeping platform independent toxicity detection as our future work. We will work with a more robust dataset for this purpose. Moreover, we will try to generate toxicity score against the comments.

References

[1] Vigna, F.D., Cimino, A., Dell'Orletta, F., Petrocchi, M., Tesconi, M. (2017). Hate Me, Hate Me Not: Hate Speech Detection on Facebook. ITASEC.

[2] Nemanja, Jing, Robin, Mihajlo, Vladan and Narayan, "Hate Speech Detection with Comment Embeddings"

[3] M. Märtens, S. Shen, A. Iosup and F. Kuipers, "Toxicity detection in multiplayer online games," 2015 International Workshop on Network and Systems Support for Games (NetGames), 2015, pp. 1-6, doi: 10.1109/NetGames.2015.7382991.

[4] William Warner and Julia Hirschberg. 2012. Detecting Hate Speech on the World Wide Web. In Proceedings of the Second Workshop on Language in Social Media, pages 19–26, Montréal, Canada. Association for Computational Linguistics.

[5] Dawei, Zhenzhen, Liangjie, Brian D.,April Kontostathis, Lynne Edwards - Detection of Harassment on Web 2.0, 2009.

[6] Qianjia, Vivek, Pradeep - Cyber Bullying Detection Using Social and Textual Analysis, 2014.

[7] https://www.kaggle.com/datasets/ashwiniyer176/toxic-tweets-dataset

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This project report was generated on Saturday 3rd September, 2022 at 7:31pm.