COMPARATIVE STUDY OF TOXIC COMMENTS CLASSIFICATION USING MACHINE LEARNING ALGORITHMS

by

Razia Razzak
16101291
Md. Sadril
16301032
Mahmudul Hasan Shakil
16301026
Mahfuzur Rahman
16101206
Sabiha Tul Omman Taki
17101519

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

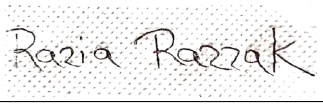
Department of Computer Science and Engineering
BRAC University
January 2021

Declaration

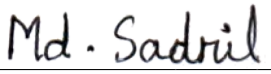
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

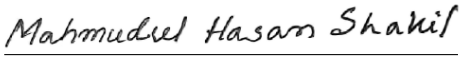
Student's Full Name & Signature:



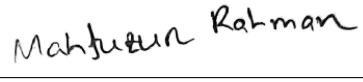
Razia Razzak
16101291



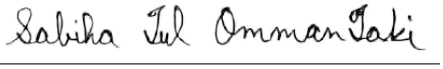
Md. Sadril
16301032



Mahmudul Hasan Shakil
16301026



Mahfuzur Rahman
16101206



Sabiha Tul Omman Taki
17101519

Approval

The thesis/project titled "Comparative Study of Toxic Comments Classification using Machine Learning Algorithms" submitted by

1. Razia Razzak (16101291)
2. Md. Sadril (16301032)]
3. Mahmudul Hasan Shakil (16301026)
4. Mahfuzur Rahman (16101206)
5. Sabiha Tul Omman Taki (17101519)

of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Computer Science and Engineering on January 8, 2021.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD
Associate Professor,
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md.Golam Robiul Alam, PhD
Associate Professor,
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor and Chairperson, Department of Computer Science and Engineering
Department of Computer Science and Engineering
Brac University

Abstract

The rapid growth of information technology and the disruptive transformation of social media have happened in recent years. Websites like Facebook, Twitter, Instagram, where people can express their thoughts or feelings by posting text, photos or videos, have become incredibly popular. But unfortunately, it has also become a place for hateful activity, abusive words, cyberbullying and anonymous threats. There are many existing works in this field but those are not fully successful yet to provide accuracy in satisfactory level. In this work, we employ natural language processing (NLP) with convolution neural networking (CNN), extreme gradient boosting (XGBoost) and support vector machine (SVM) for segmenting toxic comments at first and then classifying them in six types from a large pool of documents provided by Kaggle's regarding Wikipedia's talk page edits. Using this dataset, the hamming score of CNN model is 89% ,XGBoost model is 87% and SVM model is 84%.

Keywords: Cyberbullying; Natural Language Processing; Word Embedding; Convolutional Neural Networks; XGBoost; Support Vector Machine.

Acknowledgement

This is the work of Razia Razzak, Md. Sadril, Mahfuzur Rahman, Mahmudul Hasan Shakil and Sabiha Tul Omman Taki- students of the CSE department of BRAC University. The paper has been prepared as an attempt to consolidate the information we have acquired during these four years of education and to create a final thesis that discusses one of the most urgent problems currently terrorizing our planet in an imaginative way. All thanks to the Almighty, the creator and lord of this world, the most benevolent, beneficent and gracious, who gave us inspiration, strength and ability to complete this analysis. We are particularly grateful to our thesis supervisor, Amitabha Chakrabarty, PhD, for his immense assistance, encouragement and support in completing our research. We are also grateful to the BRAC University Faculty Staffs of the Computer Science and Engineering, who have been a light of guidance for us throughout the BRAC University study era, particularly in educating and enhancing our knowledge. Finally, we would like to express our heartfelt gratitude to our dear parents for their love and care and to our friends for their continued support and encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AdaBoost Adaptive Boosting

AUC Area Under Curve

CNN Convolutional Neural Network

FN False negative

FP False positive

GBM Gradient Boosting Machine

NLP Natural Language Processing

NLTK Natural Language Toolkit

SVM Support Vector Machine

$TF - IDF$ Term frequency-inverse document frequency

TN True negative

TP True positive

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

This chapter includes some major issues regarding our thesis topic. Specially, in the background section, we have discussed our motivation for doing the research on this topic.In research problem, we have specified the name of algorithms that we are using. In research objective section, we have discussed the main features that we want to introduced in our works.

1.1 Background

We are living in an era where technology has become one of the necessities of daily life. Over time, the pace of technological progress grows has made the internet a vast repository of intelligence that is accessed and relied on every day by millions of people. Due to the vast increase in social connections and confidentiality in online social networks, abusive activities through online means of communication have become a major problem in the world. Digitalization transfers human contact to internet channels, which has several positives, but still provides a room for internet antisocial activity such as abuse, harassment and hateful comments. Not only adults, teens and children are also being harassed on multiple sites, such as online games, YouTube etc. In the 2014 large-scale EU youngsters on-line Report [1] printed that 20% of eleven to 16-year previous had been exposed to hate messages on-line. additionally, children were 12% additional possible to be exposed to cyberbullying as compared to 2010, that clearly demonstrates that cyberbullying could be a growing drawback. The accessibility of sufficient dataset is a key problem in online harassment study, which is important to build models that characterize cyberbullying. Using the datasets that are revealed in past few years, several studies have so made their own repository from social media websites that area unit vulnerable to bullying content, like YouTube, Form Spring, Kaggle,Twitter, Instagram, MySpace and ASKfm[2]. The social media platforms have been trying to take requisite steps against hate speech and still looking for ways to automate the process. Rates of students being targeted for their religion, disability, gender, or sexual orientation were around 1 percent each[3]. In this paper, we are using four basic neural networks which are Natural language processing (NLP), Convolutional. Neural Networks (CNN), Extreme Gradient Boosting (XGBoost) and Support Vector Machine (SVM). Natural language processing (NLP) mainly for removing the toxic comments. Following data preprocessing, the proposed architecture is organized implementing procedures for data cleaning and adopting NLP methods such as tokenization, lemmatization,

stemming, and vector word translation by word embedding. Convolutional neural networks (CNN) have also been applied to subtle word embedding without any syntactic or semantic word processing, or natural language processing. In XGBoost model we have used lemmatization instead of tokenization. And in SVM model, we have used tokenization and stemming features. Our model has been tested and accuracy checks have been used to see how much efficiently it's working. According to the 2014 poll[4] of the PEW Research Institute, 73% of people on the internet have seen someone being harassed online, 45% of internet users have all been harassed and 45% were exposed to substantial harassment. More than 85% of databases are completely non-toxic and the concentrations of toxicity are totally not seen in Wikipedia. In contrast to 2010 teenagers were 12% [2] more likely to be subjected to cyberbullying, which obviously indicates negative part of social media.

1.2 Research Problem

Cyberbullying or online harassment using abusive, vulgar or hateful words has become very easy for these generation because of technological advancements. But these hateful comments can make the targeted person mentally sick, even some of them start suffering from insecurities. In This paper, our goal was to examine if there are any offensive comments in social media platforms using deep learning and to further categorize them into different categories such as toxic, severe-toxic, obscene, insult, threat and identity_hate. We also tried to measure the efficacy of the datasets of each of the algorithms. We are using Processing of Natural Language (NLP), Convolutional. Neural Networks (CNN), Extreme Gradient Boosting (XGBoost) and Support Vector Machine (SVM) for this paper. Our models have been examined and precision measurements have been used to see how well the models do.

1.3 Research Objective

The reason for this research work is to make online media free from any and all harm from a wide range of disdain and foul substance and remarks. The primary objective of this exploration is to order and distinguish any sort of disdain or offensive words or sentences from web-based media posts that can be destructive for any people or groups.This paper proposes different sets of highlight extractions including word embedding, word recurrence what is more, three different models to group scorn speech. This research proposes the word embedding(n-gram) with CNN as word implanting permits words to be spoken to in light of their implications and semantics which will permit the same importance words to have comparative portrayal. This will help in catching concealed and interesting words. XGBoost is a new calculation dependent on slope supported choice trees which are well known for speed and performance.In request to give a different measurement to this exploration XGBoost has been utilized. SVM is additionally an excellent classifier for text arrangement. As recognizing scorn discourse is moderately a new issue, there is very little related work on this topic. Besides, utilizing CNN, XGBoost and SVM for text classification exceptionally, for disdain, discourse location is the methodologies that are moderately new. This exploration expects to characterize disdain discourse

utilizing these three different models and approaches and study their presentation. The fundamental goal of this exploration work are :

- Pre-processing data applying natural learning processing (NLP)
- Finding toxic comments applying using NLP
- Reducing irregularity from data set using Data Cleaning
- Detecting and reducing words that have same meaning using stemming.
- Testing data while applying classifiers to see accuracy
- Measuring effectiveness of existing classification algorithms
- Removing toxic comments applying CNN, XGBoost and SVM
- Indicating future scopes of development

1.4 Thesis Report Outline

The remainder of the part of this exploration paper contains as following.

Chapter 2 contains the background study and literature review. It will portray the current works away at this subject and different sort of machine learning algorithms.

Chapter 3 is a methodology that depicts the structure of our complete works. It will show the entire cycle of our proposed model with appropriate outlines. This part will contain the data on our datasets, information preprocessing, and highlight extraction and classification model.

In chapter 4, the results and analysis of our model will be talked about. It will discuss the outcomes with disarray grid and classification report. This bit will show the boundaries to find out the correlation of our outcome and furthermore the finding from these outcomes. Chapter 5 will close our paper by summing up our entire work

In chapter 5, we have summarized our research work and mentioned limitations,scope and future work.

Chapter 2

Literature Review and Related Works

With the expansion of social media in last few years cyber bullying and using of toxic language has become a problem. To solve the problem not much work is done on this topic. Here are some papers that we look into:

2.1 NLP related works

In this paper [5], Text arrangement is perhaps the most broadly utilized characteristic language preparing innovations. Normal content order applications incorporate spam ID, news text grouping, data recovery, feeling investigation, and aim judgment, and so forth Conventional content classifiers dependent on AI strategies have deformities, for example, information sparsity, measurement blast and helpless speculation capacity, while classifiers dependent on profound learning network enormously improve these imperfections, stay away from awkward component extraction measure, and have solid learning capacity and higher forecast precision. For instance, convolutional neural network (CNN). This paper presents the cycle of text order and spotlights on the profound learning model utilized in content grouping.

In this paper [6], Mechanization of data extraction from qualification models will give a discovery in successful usage of data for quiet pursuit in clinical information bases. A larger part of qualification standards contains fleeting data related to ailments and occasions. This venture makes a novel natural language processing (NLP) pipeline for extraction and characterization of transient data as memorable, current, and arranged from free-text qualification models. The pipeline utilizes design learning calculations for extricating fleeting data and prepared a Random Forest classifier for arrangement.

In another paper [7], The content order is a fundamental undertaking of natural language processing (NLP), which expects to get the comparing classification marks for messages with numerous classes. These days, neural organization models have been generally utilized in the NLP field and have accomplished astounding outcomes in content grouping. In any case, because of high spatial property of text data and therefore the impressive linguistics of standard language, there square measure heretofore various territories for development within the organization structure of

text arrangement. To adapt to the above issues, this paper proposes another organization structure, which incorporates bidirectional long short-term memory (LSTM) joined with a progressive consideration system. In this organizational structure, the information is shipped off bidirectional LSTM after one-hot encoding, and the yield is exposed to progressive consideration. At long last, the SoftMax classifier is utilized to group the handled setting data. The test results show that the structure has high precision in content grouping.

In another paper [8], On account of the fast expansion in innovation and electronic interchanges, email has become a genuine specialized device. In numerous applications, for example, business correspondence, updates, scholarly notification, site page participations, email is utilized as an essential method of correspondence. In the event that we disregard spam messages, there remain many messages got each day. To decide the significance of got messages, the subject or substance of every email should be checked. In this examination, we proposed an unaided framework to characterize got messages. Gotten messages' directions are dictated by a strategy for normal language handling called as Word2Vec calculation. As per the likenesses, prepared information is assembled by k-implies calculation with a solo preparing model.

In other paper [9], Text classification is a fundamental part of the NLP, which means to foresee the classes for given writings in a specific order framework. There are numerous methods of highlight determination and characterization models. Notwithstanding, most specialists might want to utilize the epitomized techniques for outsider libraries to accomplish their objectives.

In this paper [10], Building logical frameworks is a basic issue in the field of Natural Language Processing (NLP), since most AI models give no clarifications to the forecasts. Existing methodologies for reasonable AI frameworks will in general zero in on deciphering the yields or the associations among data sources and yields. Nonetheless, the fine-grained data is regularly overlooked, and the frameworks don't expressly produce comprehensible clarifications. To more readily lighten this issue, they propose a novel generative clarification system that figures out how to settle on arrangement choices and create fine-grained clarifications simultaneously. All the more explicitly, we present the logical factor and the base danger preparing approach that figure out how to produce more sensible clarifications.

2.2 CNN related works

Convolutional neural networks (CNNs) are wide applied for classification issues in numerous fields including text classification. While doing our research, we found only few papers that was done in this topic. Amongst those few papers, one was done only for text classification using CNN. Kim et al[11] showed a series of sentence-level classification done with CNN .The authors explained how a slightest tuning and static vector shows noticeable results on multiple benchmarks. Their proposed model use task specific and static vectors. The authors also proved the importance of pre-trained and unsupervised word vectors in NLP.As internet and social media users

are increasing in a massive way, the toxicity level in social platforms are also rising substantially. Cyberbullying victimization hate comments has become a matter of significant concern, industrial and analysis communities are attempting their best to search out an economical model for on-line cyanogenetic comment prediction because of its importance in on-line interactive communications among users. In this paper, Georgakopoulos, S. V. et al.[12]used CNN model for solving toxic comment classification problem. Their research is mainly focused on the study of recent approach for text classification involving word representations and Convolutional Neural Networks. For the research, they used dataset collected Kaggle's competition regarding Wikipedia's talk page edits. The author used Word embedding and CNN to compare with the BoW approach with a set of algorithms that have proved to be very successful in text classification. The main dataset was converted in to a subset for more consistent analysis, hence used for binary classifications for filtering out toxic comments.

Zhang et al[13] proposed a new method combining convolutional neural network(CNN) and gated recurrent networks(GRU) which found to empirically improve classification accuracy. In this paper, the term 'hate speech' is interchangeably used as foul, hostile or harsh language yet they appeared that hate speech can be not the same as oppressive language Authors collected datasets from in public obtainable datasets on Twitter and created a unique one by gathering tweets discussing refugees and Muslims, that were media destined throughout the time of writing thanks to totally different recent events. The authors worked on their model by applying a bit preprocessing on tweets and then utilized CNN+GRU architecture in layers following word embedding layer, 1D convolutional layer , 1D max pooling layer ,GRU and SoftMax layer. They conducted comparative review on the datasets and showed how their proposed method outperformed baselines and gave better outcomes than the others reported results on the datasets. It determines the current standard by scoring between 1 and 13% in F1 on 6 out of 7 datasets.

The paper[14] presents a deep learning based Twitter hate speech text grouping framework. The classifier assigns every tweet to at least one of 4 predefined classifications: racism, misogyny, each (bigotry and sexism), and non-hatred expression. Four versions of the Convolutional Neural Network are ready on resp. Character 4-grams, word vectors supported linguistics knowledge made mistreatment word2vec, at random generated word vectors, and word vectors joined to character n-grams. The list of capabilities was down-sized within the organizations by max-pooling, and a softmax work accustomed characterize tweets. Tried by 10-overlay cross-validation, the model hooked in to word2vec embeddings performed best, with higher accuracy than a review, and a 78.3% F-score. Proceeds with development of long range informal communication web users, individuals day by day imparted their thoughts and insights as writings, pictures, videos, what's more, discourse. Text classification is as yet a significant issue on the grounds that these gigantic text got from the heterogeneous sources and extraordinary attitude people groups. The imparted insight is to be deficient, conflicting, uproarious and furthermore in various dialects structure. Actually, NLP, what is more, deep neural organization techniques are commonly used to address these issues. Along these lines, Word2Vec Word Implanting and Convolutional Neural Network (CNN) technologies must be modified

for viable content grouping. In this paper, the proposed model impeccably cleaned the information and generated Word vectors from a pre-prepared Word2Vec model and use CNN layer to remove better highlights for short sentences arrangement[15]. Word embeddings and CNN (Convolutional Neural Networks) engineering are the main results of the opinion inquiry. No matter what, the notion and dictionary embeddings are scarcely used, CNN is sloppy to capture the world's highlights of the phrase[16]. To this finish, linguistics embeddings, opinion embeddings, and vocabulary embeddings ar applied for messages secret writing, what is additional, 3 distinctive concerns as well as attention vector, LSTM (Long Short Term Memory) thought, and aware pooling ar incorporated with the CNN model during this paper. Moreover, a word and its setting ar investigated to elucidate the importance of the word for made input portrayal. to enhance the presentation of 3 various thought CNN models, CCR (Cross-methodology Consistent Regression) and move learning ar introduced. It's value seeing that CCR and move learning is being employed for the primary time in a very text sentiment investigation. Finally, some analyses of 2 separate datasets indicate that the projected thought of CNN models produces the simplest or the subsequent best results against existing progressive models.

2.3 XGBoost related works

Not much work done with XGBoost to do text classification as it is relatively new. But we found some work like in this paper[17] the authors worked with XGBoost to detect false information that is almost similar to the type of work we do, using XGBoost LSTM, Random Forest and others to retrieve data sets, but XGBoost done much better than these. The authors used attention-based models that only use text information from various papers. In terms of accuracy, the results show that our XGBoost model improved 16.4 percent and 13.1 percent over the best baseline.

In another paper[18] the authors recognize and forestall online media hostility in both the English Hindi and Hindi-English blended datasets. The creators utilized characteristics like word vectors, forceful words, feeling scores, discourse parts and emoticons for the arrangement task. They have noticed various characterization techniques for AI, for example, XGBoost, Support Vector Machine (SVM) and Gradient Boosting Classifier (GBM). Among these, the most ideal approach to order XGBoost is. Online media, for example, Facebook, Twitter, etc. play out an essential correspondence work for the headway of creative mind, regardless of whether it is messaging, sharing pictures, video calls, or remark commitment. Notwithstanding these ideal conditions, it likewise has some negative viewpoints, which carries hate to certain territories of residents. Subsequently, such enmity, disdained by online media, ought to be perceived and dodged, which is the basic motivation behind our work. There have been banters about Hindi, English and Hindi-English blended datasets. The job is ordinarily appropriate for the Support Vector Machine (SVM). For the predominant part projecting a polling form that gives f-scores of 68.13, 54.82 and 55.31 independently for blended datasets, the yield of the three classifiers was utilized thusly.

In[19] Wikipedia English worked to classify web bullying and spam. The over-

whelming presence of online communities characterizes today's digital landscape. Web based harassing has been one of the persevering dangers to the ideal of free-streaming exchange in these societies. Wikipedia is a for example, as the danger of online provocation going from scorn discourse to individual maltreatment to spam has been looked by its wide local area of donors. As of now, to identify online badgering, Wikipedia has a human-driven component set up. They propose a structure in this paper for comprehension and distinguishing such brutality in the English Wikipedia culture. They examine the freely accessible information sources provided by Wikipedia. They find that Wikipedia's XML dumps need progressed information examination capacity to be utilized for transitory literary investigation, and, as another option, we propose a web scratching procedure to gather client level information and perform broad exploratory information examination to guarantee the highlights of clients who have been hindered for damaging acts before. We make a model of misuse identification with these information that uses regular language handling procedures, for example, character and word n-grams, characterization of assessment and arrangement of examples, and creates highlights that are utilized to anticipate Violent lead in a model dependent on AI calculations as data sources. Our best badgering location model, utilizing the XGBoost characterization model, gives us an AUC score of 84 percent.

In this paper[20] Depression is observed using XGBoost by the authors. Depression is both debilitating and widespread. It is usually undiscovered though treatable. Detached scanning of sorrow is necessary, but there are security concerns about using information from smartphones and online media. They predict that the dormancy of messaging responses would contain useful data in screening for depression, based on the recognized link between despondency and slower data preparation speed. They distinguish nine response latency-related characteristics from meta-information publicly sponsored instant message discussion in particular. They moderate the security issues by considering text metadata rather than content. We examine a number of machine learning techniques focused on head components of the inactivity highlights to predict paired screening summary ratings. Their results show that a single head segment XGBoost model achieves an F1 score of 0.67, an AUC of 0.72, and an accuracy of 0.69. In this way, they say that reacting to messaging idleness is assured as a methodology for screening despondency.

2.4 SVM related works

In the paper[21], the authors examine whether the programmed arrangement of news texts can be improved by profiltering the vocabulary to lessen the highlights utilized in the counts. At first, they compare artificial neural network and support vector machine calculations for use as news thing text classifiers. Ultimately, researchers distinguish a decrease in the arrangement of highlights that conveys improved outcomes.

The paper [22] proposes an ideal SVM algorithm for text classification by methods for different ideal techniques. The reason for the content characterization framework is to choose if the archive being referred to has a place with which of the predefined

classifications. The test results show that the proposed ideal characterization calculation delivers much preferable execution over other conventional algorithms.

In this paper[23], author rexplain the effects of a large number of experiments with different pre-processing techniques to generate successful input features. It turns out that n-grams of syllables and phonemes are particularly useful for classification.

In the paper[24], authors investigate text classification problem and equate SVM to kNN and naive Bayes on binary classification assignments. It's imperative to analyze advanced variants of these algorithms, which is the thing that we've done. Researches show that all the classifiers have accomplished equivalent execution on most issues. SVM was a fairly good overall results.

In the paper[25], to decide if SVM-based classifiers prepared in a mix of incorporation and normal avoidance articles are valuable for specialists assessing diary articles for consideration in new methodical audits. Programmed, top notch article classifiers utilizing AI may diminish the outstanding burden of specialists leading efficient surveys when subject-explicit information is scant.

The paper[26] researches the classification impact based on the SVM technique in Chinese content information and will utilize the support vector machine strategy in Chinese content for the grouping of Chinese content and for the blend of the scholarly world and experimental application.

In this review[27], the uses of the support vector machine with kernel blend (SVM-MK) for the blueprint of a text classification framework are examined. In contrast with the customary SVM, the SVM-MK utilizes a 1-norm based object function and receives raised mixes of single-function straightforward pieces. Simply a straight programming issue needs to be addressed and the cost of computing is substantially reduced.

Chapter 3

Methodology

To classify toxic comments, there have been used different classifiers such as multinomial logistic regression, Naïve Bayes, Random Forests, Decision Trees etc. Previously, two or more classifiers were combinedly used to detect hate speech in some cases which is called ensemble approach of text classifications. Generally, different classifiers give different accuracy on same dataset. So, we applied three different algorithms on our dataset to show how it varies from one another. The classifiers we used are CNN, XGBoost and SVM. In the following sections, we will be discussing our chosen methods and how they work.

3.1 Dataset

We are using dataset consist of around 1,60,000 different types of comments. There are total 9 columns in our dataset. The dataset we've got utilized in our analysis could be acquired from Kaggle that is a very talked-about publically on the market dataset named "Wikipedia speak Page Comments annotated with toxicity reasons" that content nearly one lakh sixty thousands comments with manually labeling. There are three parts containing in our dataset. Figure 3.1 shows the train dataset of our model. In our train dataset, it contains total six classes (toxic, severe_toxic, obscene, insult, threat, identity_hate). We have run this dataset in our code. Then we got some results based on train dataset.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

Figure 3.1: Train Dataset

Then we also ran our code in test dataset to see if it matches with the result of our train dataset. Figure 3.2 shows the test dataset of our model. Here, we have collected different types of comment to test our original dataset. The result of the test dataset is given below:

	id	comment_text
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	00017563c3f7919a	:If you have a look back at the source, the in...
4	00017695ad8997eb	I don't anonymously edit articles at all.

Figure 3.2: Train Dataset

3.2 Data Preprocessing with NLP

Data preprocessing could be a data processing technique that's utilised to vary the crude knowledge into a useful and effective arrangement. Certifiable knowledge is frequently inadequate, conflicting, sickly in specific practices or drifts, and is maybe attending to contain varied mistakes. knowledge preprocessing could be a incontestable technique for subsidence such problems. Data preprocessing plan crude information for additional preparing. We have included different types of data preprocessing stages. These are data cleaning, tokenization, lemmatization, stemming etc.

3.2.1 Data Cleaning

Data Cleaning assumes a significant part in the field of Data Management just as Analytics and Machine Learning. Data Cleaning implies the way toward distinguishing the off base, fragmented, incorrect, superfluous, or missing aspect of the Data and afterward adjusting, supplanting, or erasing them as indicated by the need. Data cleaning is viewed as a central component of the essential Data science. Data is the most important thing for Analytics and Machine learning. In processing or Business Data is required all over. With regards to certifiable Data, it isn't implausible that Data may contain inadequate, conflicting, or missing qualities. On the off chance that the Data is defiled, at that point it might impede the cycle or give erroneous outcomes. We should see a few instances of the significance of Data cleaning [17]. Natural language processing (NLP) is an extremely active field of research. It gives the likelihood to examine human language from the applied side, not simply hypothetically, and to attempt to explain some of the assignments thinking about human language. Python and the Natural Language Toolkit (NLTK) permit any software engineer, even an amateur, to get to know NLP errands effectively without investing

a lot of energy in contemplating or gathering assets. The point of this paper is to give significant evidence and models, which show how essential the NLTK is for the course of Computational Linguistics at the college and for analysts in the field of characteristic language handling.

3.2.2 Tokenization

Tokenization is breaking the Raw content into very little lumps. Tokenization breaks the raw content into words, sentences referred to as tokens. These tokens facilitate in understanding the distinctive circumstance or build up the model for the information processing. The tokenization helps in deciphering the importance of the content by breaking down the succession of the words. as an example, the text "God bless you" are often tokenized into 'God', 'bless', 'you'. There area unit varied techniques and libraries accessible to perform tokenization. NLTK, Genism, Keras area unit some of the libraries that may be used to attain the endeavor. Tokenization ought to be potential to either isolate words or sentences. within the event that the content is an element into words utilizing some detachment procedure it's referred to as word tokenization and an identical division accomplished for sentences is termed sentence tokenization. Stop words area unit those words within the content which does not add any importance to the sentence and their evacuation will not influence the handling of text for the characterised reason. They are eliminated from the jargon to decrease commotion and to diminish the component of the list of capabilities. We will follow some workflow in this process.

RAW TEXT → TOKENIZATION → VECTORIZATION

Raw information contains mathematical worth, accentuation, unique character, and so forth. These qualities can hamper the presentation of the model so before applying any content featurization first we have to change over Raw information into significant information which is additionally called as text preprocessing. In tokenization, we convert a gathering of sentences into a token. It is likewise called text division or lexical examination. It is fundamentally parting information into a little lump of words. The figure below shows how the sentence is broken down to a segmented form. Here, a model named FastText is used for mapping the word to a vector number.

In first step, chunk of words will be separated from a big sentence or content of information such as ["I hate toxic words"] to ["I", "hate", "toxic", "words"] and in second, the words will be embedded with some numbers to represent word vectorization. It mainly compares the group of vector words that are in vector space and finds the mathematical similarity like man to boy and woman to girl.Figure 3.3 is glimpse of tokenization.

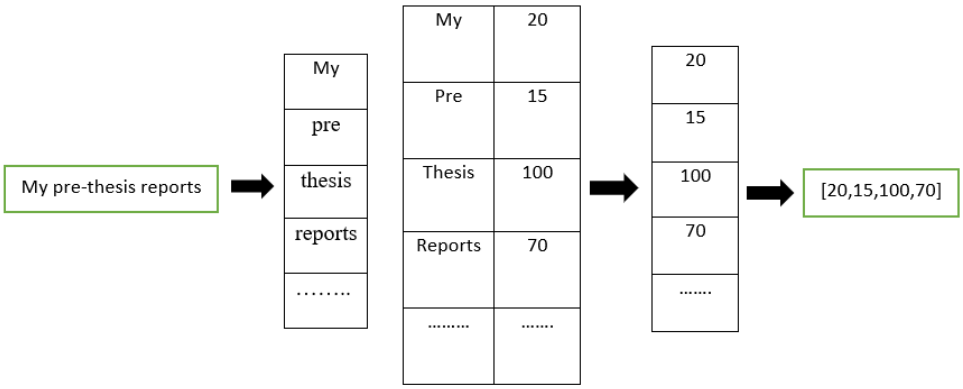


Figure 3.3: Process of Tokenization

3.2.3 Stemming

To lay out plainly, stemming is the way toward eliminating an aspect of a word, or decreasing a word to its stem or root. This may not really mean we're diminishing a word to its word reference root. We utilize a couple of calculations to conclude how to hack a word off. This is, generally, how stemming contrasts from lemmatization, which is lessening a word to its word reference root, which is more perplexing and needs an exceptionally serious extent of information on a language. We should accept we have a lot of words — swim, swam, and swimming. Each of the three words are various tenses of a similar root word swim. So, after we stem the words, we'll have quite recently the single word — swim.For example, Swim, swam, swimming = Swim. Playing, played = Play.

Be that because it might [18], the 2 words vary in their flavor. Stemming ordinarily alludes to a rough heuristic cycle that cleaves off the finishes of words within the need for accomplishing this objective accurately additional usually than not, and often incorporates the evacuation of derivational joins. Lemmatization usually alludes to doing things fitly with the use of jargon and morphological investigation of words, usually meaning to eliminate inflectional endings simply and to revive the bottom or word reference kind of a word, that is thought because the lemma. Whenever defied with the symbolic saw, stemming might restore simply s, tho' lemmatization would endeavor to come either observe or saw contingent upon whether or not the use of the token was as AN action word or a issue. the 2 might likewise distinction therein stemming most typically crumples derivationally connected words, whereas lemmatization ordinarily simply falls the distinctive inflectional kinds of a lemma. story making ready for stemming or lemmatization is frequently done by an additional module half to the ordering cycle, and varied such segments exist, each business and open supply.

3.2.4 Punctuation removal

In any kind of social media data whether it is a comment or a tweet there might be so many punctuations which are not necessary for data analysis or hatred detection [28]. We have removed all the punctuations using regular expressions.

3.2.5 Stop Words Removal

Stop words area unit the articles, auxiliaries and a few of the common words that aren't necessary for text analysis. NLTK have some predefined stop words but it also allows to add some more stop words or modify the stop word list. We have applied the NLTK stop word to remove unnecessary words from the corpus.

3.2.6 Lemmatization

Lemmatization is a process which shortens a group of similar words into the root of the words. It is the normal text pre-processing methods utilized in Natural Language Processing (NLP) and AI in general. The root word is known as a lemma in the lemmatization process. In other words, lemmatization shortens the suffix-prefix of a word and convert a word into its root form which minimizes the word space on the corpus[29]. For example, in a corpus there may exist words like able, disable, enable, ability, disability and so on. Lemmatization will shorten the suffix and prefix parts of the words disable, enable, ability and disability all the word will be referring to the root word 'able'.

3.3 Feature Extraction

As we are doing research on text classification , we need some features based on our dataset. So, the following part is a discussion based on the techniques that has been used for feature extraction.

3.3.1 Word Embedding

A word embedding is a scholarly portrayal for text where words that have a similar importance have a comparable portrayal. It is this way to deal with speaking to words and records that might be viewed as one of the key forward leaps of profound learning on testing characteristic language preparing issues. We have classified word embedding into different terms. These are FastText, Word2Vec,GloVe,TF-IDF etc.

3.3.2 FastText

FastText is another word embedding procedure that is an augmentation of the word2vec model. rather than learning vectors for words decisively, FastText ad-

dresses each word as a n-gram of characters. This aides get the significance of more limited words and allows the embeddings to get a handle on increments and prefixes. when the word has been addressed using character n-grams, a skip-gram model is prepared to accomplish capability with the embeddings. This model is seen as a bunch of words model with a window over a word in light-weight of the very truth that no inside structure of the word is considered. yet, long the characters ar inside this window, the solicitation for the n-grams doesn't assemble a qualification.

3.3.3 Word2Vec

Word2vec might be a two-layer neural web that estimates text by "vectorizing" words. Its information might be a book corpus and its yield are a lot of vectors: encapsulate vectors that discuss to words in this corpus. while Word2vec is not the slightest bit a significant neural association, it changes text into a numerical structure that significant neural associations will grasp. the clarification and accommodation of Word2vec are to gather the vectors of tantamount words in vector house. That is, ten it separates comparable qualities mathematically. Word2vec makes vectors that are dispersed numerical depictions of word features, features, for instance, the setting of individual words. It will all by itself while not human mediation.

3.3.4 GloVe

GloVe could be a solo learning calculation for obtaining vector portrayals for words. getting ready is performed on destroyed worldwide word-word co-event insights from a corpus, and therefore the succeeding portrayals feature intriguing straight foundations of the word vector house. The quantity of "settings" is, obviously, huge, since it is basically combinatorial in size. So, then we factorize this lattice to yield a lower-dimensional framework, where each column currently yields a vector portrayal for each word. All in all, this is finished by limiting a "recreation misfortune". This misfortune attempts to discover the lower-dimensional portrayals which can clarify the greater part of the fluctuation in the high-dimensional information.

3.3.5 TF-IDF

TF*IDF may be a information recovery procedure that weighs a term frequency (TF) and its inverse document frequency (IDF). every word or term that happens within the content has its separate TF and IDF score. The results of the TF and IDF variant a term is understood because the TF*IDF weight of that term. TF is used to find out the frequency of a sentence and IDF is used for finding rare words in datasets[30]. We have used this technique in both XGBoost and SVM. This is also called count vectorization technique.

$$TF = \left(\frac{Number\ of\ times\ a\ word\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \right) \tag{3.1}$$

$$IDF = log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ the\ words}\right) \tag{3.2}$$

3.4 Text Classification using CNN

Convolutional neural network(CNN) is one of the class of Deep Neural Networks which is most often applied to image processing problems.Besides this, CNN is being applied in text classification, sentiment classification,signal processing.Convolution Neural Networks are biologically inspired variants of Multilayer Neural Network [31].The reason behind the heavy use of CNN is its learning parameters.There are several applications of text classification like hate speech detection, intent classification, and organizing news articles. Text classification may be a classic topic for language process and a vital element in several applications, like internet looking out, info filtering, topic categorization and sentiment analysis.[32].

3.4.1 Features of CNN

The basic structure of CNN is described below-

Input Layer

The very first thing we need to do for text classification is to give input for continuing the further steps, which we can call as Input layer. Input layer works as the initialization of the whole CNN. The input layer of a CNN is made out of counterfeit input neurons and carries the underlying information into the framework for additional preparation by ensuing layers of fake neurons. Along these lines, this layer begins the work process for the CNN classifier. The information layer might be a sentence involved linked word2vec word inserting that is trailed by a convolutional layer with numerous channels, at that point a maximum pooling layer, and at last a Softmax classifier. The data are preprocessed in the word embedding layer of the NLP before we feed its review into CNN as input. Precisely it can be said that the Embedding layer output works as input here. We slide over input data the convolution to extract features. An n-length sentence (and padding can be used according to needs/requirements) can be represented as-

$$x_1 : n = x_1 \oplus x_2 \oplus \cdots + x_n \tag{3.3}$$

Here, xi ϵ Rk is the vector of a word in the text with k as the dimension.

Convolutional Layer

As the input layer is concatenated with word embedding of NLP, in some researches it's not enclosed as a layer of CNN. on it context, main layering of CNN is started from Convolutional Layer. The Convolutional layer is sometimes the primary layer for CNN wherever we have a tendency to convolute image or data normally victimisation filters or kernels.in another word, to form a 3rd relation, it's a mathematical combination of 2 relationships that use 2 sets of data.

While adding a convolutional layer to a model, it additionally required to specify what percentage filters one desires the layer to possess. Filters square measure little units that we have a tendency to apply across the information through a window. A filter will technically consider as a comparatively little matrix that we have a tendency to decide range, the amount, the quantity of rows and therefore the number of columns that this matrix has. The depth of the filter is same as input matrix. breadth of the filter are same because the breadth of embedded matrix, whereas the filter height could vary[32].As the filter is applied over input and several other feature maps square measure generated, AN activation operate named ReLu is left out the output to supply a non-linearity for output.

Only non-linear activation functions ar used between consequent convolutional networks. If we have a tendency to simply use linear activation functions, there will not be any learning. thanks to the associativity property of convolutional, these 2 layers ar effective even as single layer. In some researches, they consider activation as a single layer. Some on the other hand, refers it as a part of convolutional layer. Activation is not necessarily executed after convolution. Most of the papers follow the sequence like convolution→ activation→pooling. This is not strictly the case as:

$$relU(MaxPool(Conv(M))) = MaxPool(reLu(Conv(M))) \tag{3.4}$$

The feature map generated by the convolutional layer is taken by activation function to generate the output activation map. We can represent a feature map which as-

$$c = [c1, c2, c3, ...c_n - h + 1] \tag{3.5}$$

If a layer output is processed as an input for next layer, it is necessary to propagate the output of the previous layer through an activation function to use an extreme value of the output. The length of input and output is maintained by padding. [33] The output of ReLU is clipped to zero on condition that convolution output is negative.

If xi:i+j is used for showing the concatenation of words xi,xi+1, . . . ,xi+j and wϵRhk, is used as a h words window that generates new feature, then a generated feature ci can be represented like this-

$$xi : i + h - 1byci = f(wxi : i + h - 1 + b) \tag{3.6}$$

Here bϵR is an inclination term and f is a non-linear capacity, for example, the hyperbolic tangent. This channel is applied to every conceivable window of words in the sentence to create a component map.

$$\mathrm{c_i} = [\mathrm{c1}, \mathrm{c2}, \mathrm{c3} \ldots\ldots\ldots, \mathrm{c_n} - \mathrm{h} + 1] \tag{3.7}$$

with c $\in Rn - h + 1$

3.4.2 Pooling Layer

The output of the convolution layer is then used because the input of pooling layer. it's used when every convolution layer. Pooling involves a down sampling of options. Typically, there area unit 2 hyper parameters within the pooling layer. the primary parameter is that the dimensions of the spatial extent that is especially reducing the spatiality of feature map and therefore the second layer is stride that is what percentage options the window skips on the dimension and height. goop pool layer uses 2*2 max filter with the stride of two that may be a non-overlapping filter. A max filter returns the most price that area unit the options within the regions. Average filters that returns the common of options may also be used however the max pooling works higher in observe. For this reason, max pooling is employed principally. Since pooling is applied through each layer within the 3D volume, the feature map's depth won't amendment when pooling. Max pooling can be represented as -

$$c' = \max[c] \mid \quad] \tag{3.8}$$

where ci is feature map$c_i = [c1, c2, c3 \ldots\ldots\ldots, c_n - h + 1]$

3.4.3 Fully Connected layer

The last layer of the CNN classifier is the fully connected layer. The output of the pooling layer that is 3D feature map, is the input for this layer. But that input is a one-dimensional feature vector. The depth of 3D feature map is high and the reason of this increased depth is the increased number of kernels that are used in the previous layers. To convert this into one dimension, the output width and height should be made to 1 using flattening. Flattening means converting the 3D matrix into a 1D vector. This activation function is used for characterizing the generated features of the input into different classes based on the training dataset. At the end of this layer is the Softmax and Logistic layer. For binary classification, logistics is used, and Softmax is for multi-classification.Figure 3.4 shows the basic architecture of CNN model.

3.5 Text Classification using XGBoost

XGBoost stands for extreme gradient boosting. XGBoost is a library boosted by an optimized distributed gradient. It is highly effective, scalable and portable. Under the Gradient Boosting paradigm, it applies machine learning algorithms. XGBoost offers a parallel tree boost (also known as GBDT, GBM) that easily and reliably addresses several data science issues. For supervised learning issues, XGBoost is used, where we use the xixi training data (with multiple features) to predict a yiyi goal variable. It is based on a decision tree. It can be used for regression, categorization, ranking and prediction specified by the consumer. For small to medium size data, XGBoost is best.

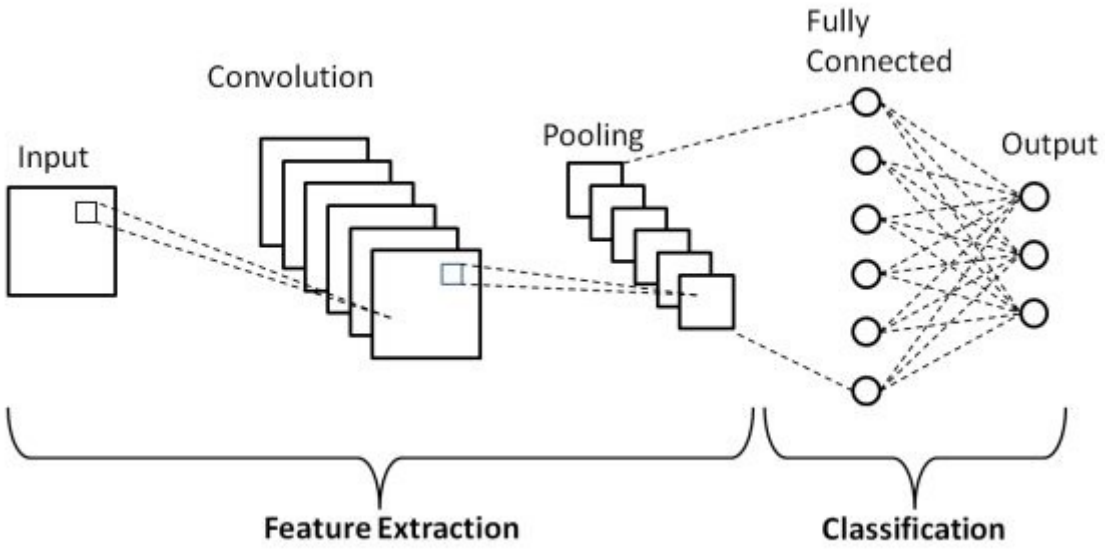


Figure 3.4: Basic architecture of CNN

3.5.1 XGBoost features

As there are few frills, the library is laser-focused on computational speed and model efficiency. Nonetheless, it does deliver a range of sophisticated functions.

3.5.2 Model Features

The implementation of the system promotes the functionality of the applications of scikit-learn and R, with new additions such as regularization. There is support for three major types of gradient boosting:

1. The Gradient Boosting algorithm, including the learning rate, is also called the gradient boosting machine.
2. Stochastic Gradient Boosting with row, column and column sub-sampling per split step.
3. For both L1 and L2 regularization, regularized Gradient Boosting.

3.5.3 System Features

The library offers a framework to be used in a number of computing environments, not least:

1. . Tree construction parallelization and use all of your CPU cores during preparation.
2. . For training very large models using a cluster of computers, Distributed Computing.
3. For rather huge datasets that do not fit into memory, Out-of-Core Computing.
4. Cache Optimization of data and algorithm structures to allow the best use of hardware.

3.5.4 Algorithm Features

The implementation of the algorithm has been optimized for compute time and memory space efficiency. One design aim was to make the best use of the resources available to train the model. Such main features for implementing algorithms include:

- Implementation of Sparse Aware with automated handling of missing data values.
- Block Structure to enable the development of tree parallelization.
- Continued training so that on new data you can further improve an already fitted model.

XGBoost is free open source software that can be used under an Apache-2 permissive license.

3.5.5 Background of XGBoost

To understand and use XGBoost properly we must know how XGBoost developed. XGBoost is built after the gradient boosting it is far better in accuracy and timing. For this it has recently gained popularity for its effectiveness. So, two things need to understand before XGBoost. Boosting and gradient boosting machine.

Boosting

For the most part, the advancement of an application for AI comprised of taking a solitary student, for example, a logistic regressor, a decision tree, a support vector machine, or a fake neural organization, taking care of its information and showing it through this data to play out an undertaking. At that point, outfit techniques were conceived, requiring the utilization of a few students to independently help the presentation of any of them. (Overall, accomplish just marginally preferred outcomes over an arbitrary model) together. As a rule, outfit procedures, as we can see beneath, are created by gathering variations of individual Decision Trees. Inside this group of troupe techniques, Boosting models fall. Boosting, initially called Hypothesis Boosting, comprises the idea of separating or gauging the information used to prepare our frail understudy group, so each new understudy gives more weight or is just prepared with discoveries that have been inadequately ordered by past understudies.
Our model group figures out how to make precise expectations on a wide range of information by doing this, not simply on the most well-known or straightforward discoveries. Boosting ought not to be mistaken for Bagging, the other significant group of outfit strategies: while poor people students are prepared in sacking utilizing arbitrariness in equal, the students are consecutively prepared in boosting to have the option to execute the information task.

Gradient boosting machine

Through first executing the AdaBoost Algorithm, the gradient boosting calculation (gbm) can be clarified all the more promptly. The AdaBoost Algorithm begins with the preparation of a choice tree in which an equivalent weight is allotted to every perception. We raise the loads of those discoveries that are difficult to sort and lower the loads for each one of the individuals who are anything but difficult to arrange, in the wake of analyzing the primary tree. On this weighted information, the subsequent tree is then developed. Here, the idea is to work with the main tree's forecasts. Thus, our present worldview is Tree 1 + Tree 2. From this refreshed 2-tree group model, we at that point measure the order mistake and create a third tree to assess the updated residuals. For a given number of cycles, we duplicate this cycle. The resulting trees permit one to recognize perceptions that the past trees don't group well. Accordingly, the weighted amount of the expectations made by the past tree models is the forecasts of the last gathering model.

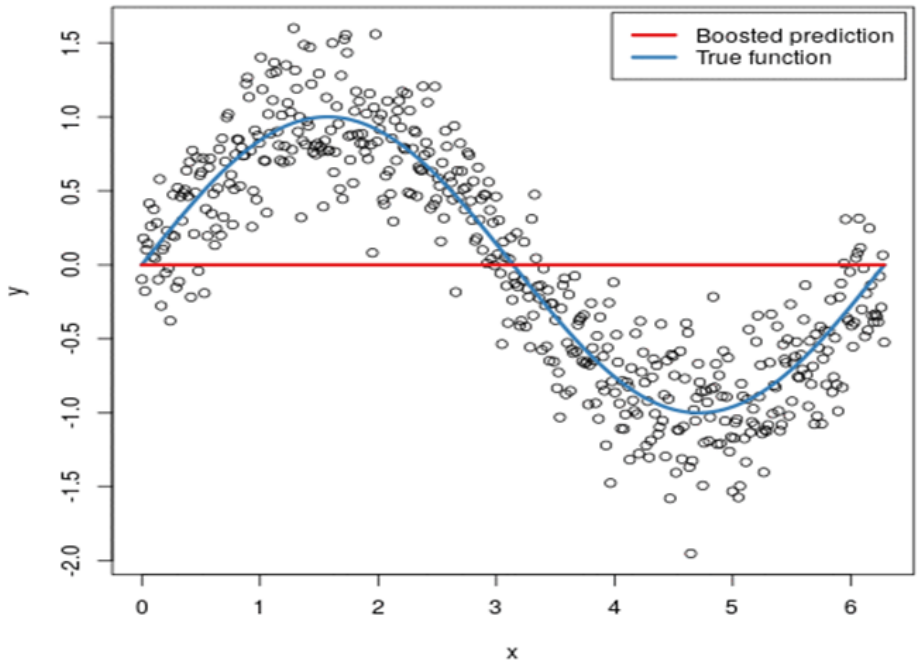


Figure 3.5: Prediction using XGBoost

Figure 3.5 indicates the prediction of XGBoost algorithm.Gradient Boosting, in an incremental, additive and sequential way, trains multiple models. How the two calculations characterize the weaknesses of weak students is the vital qualification among AdaBoost and the Gradient Boosting Algorithm (e.g. choice trees). While the AdaBoost model characterizes the inefficiencies by utilizing high weight information focuses, by utilizing angles in the misfortune work, slope boosting plays out the equivalent (y=ax+b+e, e requires a superior note as it is the blunder term). The misfortune work is a measurement indicating how successful the coefficients of

the model are at fitting the hidden outcomes. A reasonable familiarity with the job of misfortune will rely upon what we are endeavoring to augment. For instance, on the off chance that we need to utilize relapse to gauge deal esteems, the misfortune capacity will be founded on the mistake among genuine and expected house costs. Moreover, in the event that our point is to group credit defaults, at that point measurement of how compelling our prescient model is at characterizing awful advances will be the misfortune include. One of the principal factors for utilizing slope boosting is that it causes one to expand a cost work characterized by the client, instead of a misfortune work that commonly has less influence and doesn't really adjust to applications in reality.

3.6 Text Classification using SVM

Support Vector Machine (SVM) a machine learning algorithm which is largely implemented in classification problems although it can be implemented in both classification and regression problems. In an SVM algorithm, data items are plotted as points in the n-dimensional space, with the values of particular co-ordinates as the values of features. Then, by detecting the hyper-place that distinguishes the two classes, classification is performed. The objective of SVM, as stated before, is detecting a hyperplane in the n- dimensional space (where nis the number of features) that distinguishes the points of data. SVM is suitable for big sample sets of classification, by and large for text classification. The algorithm of SVM is grounded on structural risk minimization theory, which works by first compressing the original data in order to support the vector set, then by learning to use the subset, new information is obtained. The regulations, made distinct by support vector is also given. The SVM therefore, is a good classifier, which has better performance and applications to an extensive degree.Figure 3.6 shows SVM classifier.

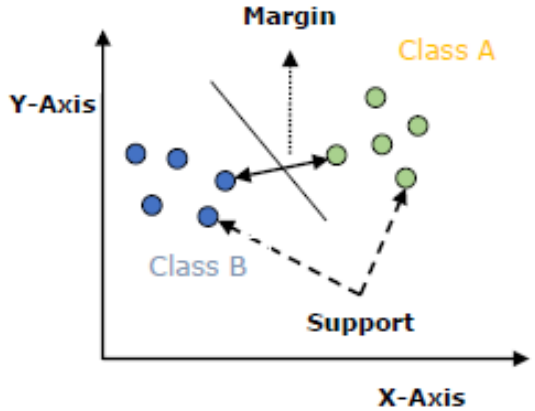


Figure 3.6: SVM classifier

Some of the important notions of SVM are mentioned below: Support Vectors: The datapoints are necessary to construct the SVM. The points most adjacent to the

hyperplane are addressed as support vectors which help in separating lines. The support vectors effect the orientation and position of the hyperplane. The margin of the classifier can be maximized with the help of the support vectors and deleting them results in the alteration of the position of the hyperplane.

Hyperplane

As seen from the diagram presented above, the space or decision plane gets split up between a set of objects belonging to different classes. The decision boundaries that aid in distinguishing the data points are called hyperplanes. Different classes are assigned to the data points that fall on either sides of the hyperplane. The number of features decides the dimension of the hyperplane. For example, the hyperplane is a two-dimensional plane if the number of input feature is 3. However, in case the number of input feature surpasses 3, the dimension of the hyperplane becomes difficult to imagine.

Margin

Margin is characterized as the hole that exists between the two lines of the closest information focuses that have a place with various classes and is determined by estimating the opposite distance between the line and the help vectors. The Margins that are large are viewed as acceptable edges while the terrible Margins are the edges that are little. In strategic relapse, the yield estimation of the direct capacity is taken and compacted with the reach [0,1] with the assistance of the sigmoid capacity. The compacted esteem is allowed with levels. In the event that the worth surpass a limit esteem (0.5), level 1 is allocated. On the off chance that it is under a limit esteem (0.5), level 0 is appointed. In SVM, the yield estimation of the straight capacity is taken and on the off chance that the worth surpasses 1, one class is relegated and on the off chance that the yield is - 1, another class is appointed. As in SVM, the limit esteems are changed to 1, and - 1, the fortification scope of qualities is [-1,1]. It goes about as Margin. Figure 3.7 shows the SVM scattered diagram.

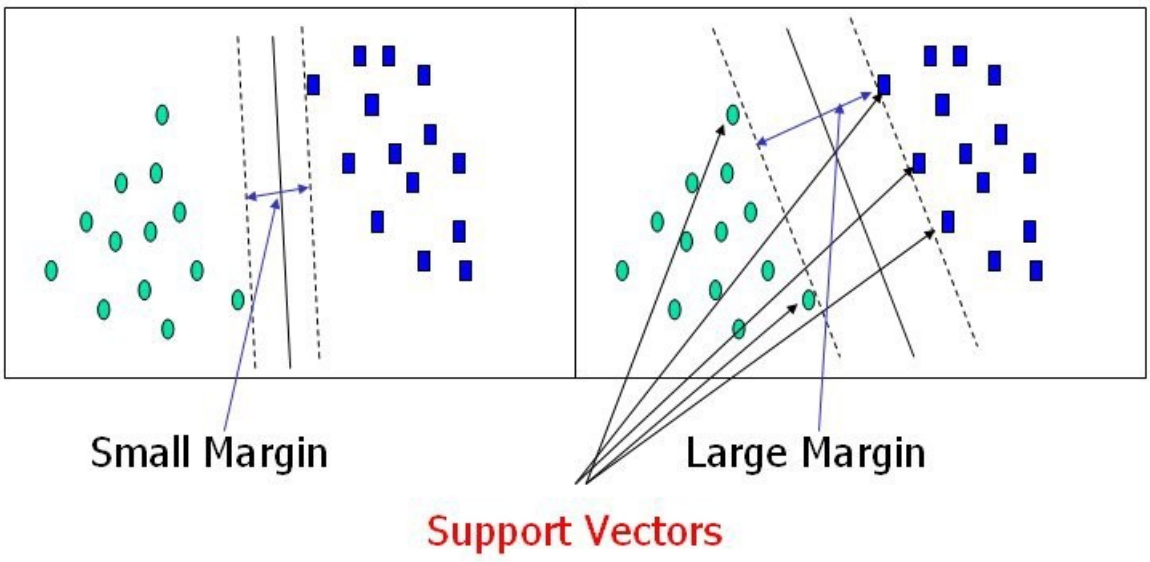


Figure 3.7: SVM scattered diagram

3.6.1 SVM Kernel

SVM algorithm is applied with the portion that changes over an input data space into the required structure. In SVM, the part admissions a low dimensional space and converts it into a higher-dimensional space, the strategy for which is known as the portion stunt. Part joins more extra measurements and converts non-detachable issues into divisible issues, which makes SVM more precise, adaptable, and amazing. A portion of the sorts of pieces utilized by SVM are referenced underneath:

3.6.2 Linear Kernel

Linear kernel is used as the dot product of any two observations. The formula is given below-

$$K(x, xi) = sum(x \times xi)K(x, xi) = sum(x \times xi) \tag{3.9}$$

It can be seen from the formula above that the product of two vectors is the sum of the multiplication of each pair of values that were given as input.

Polynomial Kernel
Polynomial kernel is a nonexclusive type of linear kernel and differentiates curved or nonlinear input space. The formula is given below-

$$k(X, Xi) = 1 + sum(X \times Xi)^d k(X, Xi) = 1 + sum(X \times Xi)^d \tag{3.10}$$

in the formula given above, d is the degree of polynomial which is needed to be stated manually in the learning algorithm.

Radial Basis Function (RBF) Kernel
RBF kernel charts the input space in unlimited dimensional space. SVM classification mostly utilizes this kernel. The formula is given below-

$$K(x, xi) = exp(-\gamma \times sum(x - xi^2))K(x, xi) = exp(-\gamma \times sum(x - xi^2)) \tag{3.11}$$

In the formula given above, gamma is the range [0,1], which is needed to be stated manually in the learning algorithm. A suitable default gamma value is 0.1. The primary objectives of SVM can be summed up in two points: (1) It is used to observe for the linearly separable case. As for the linearly inseparable case, a nonlinearity mapping is required to convert the low dimensional sample space that is inseparable, to higher dimensional feature space, which is separable. (2) SVM is also grounded on structural risk minimization theory, which, from the feature space, locates the optimal separating hyperplane. Therefore, the learning machine gets global optimization and the estimated danger of the whole sample space meets a definite upper limit with a probability.

3.7 Our Proposed Model

In this work, we employ natural language processing (NLP) with convolution neural networking (CNN), extreme gradient boosting (XGBoost) and support vector machine (SVM) for segmenting toxic comments at first and then classifying them in six

types from a large pool of documents provided by Kaggle's regarding Wikipedia's talk page edits.

Convolutional Neural Network (CNN):
At first, we have collected dataset containing six different classes. This dataset has almost 1,60,000 different types of comment. Using NLP algorithm, we have cleaned our dataset with tokenization and steaming feature. We make a confusion matrix based on CNN algorithm. Then, train and test data set has been called. After that, we tokenized every comment of our data set and also assigned some values to those tokenized words which is called vectorization. Then, we have labelled our data with the help of word embedding feature that contains FastText, Word2Vec and GloVe.Then, We use the CNN classifier on dataset. There are different layers in CNN architecture that find the toxicity of the labelled dataset. In CNN architecture, at first we called embedding layer. Then, we called convolutional layer that included MaxPooling, GlobalMaxPooling and BatchNormalization function. Then, fully connected layer came in the architecture. The following figure 3.8 shows the layer for CNN architecture.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 371, 300)	63101400
conv1d (Conv1D)	(None, 371, 128)	192128
max_pooling1d (MaxPooling1D)	(None, 123, 128)	0
global_max_pooling1d (Global	(None, 128)	0
batch_normalization (BatchNo	(None, 128)	512
dense (Dense)	(None, 50)	6450
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 6)	306

Figure 3.8: Layer of CNN architecture

Then, we try to find out the AUC score to evaluate how our algorithm specifically identify the words correctly. We take the batch size as 64 and epoch as 2 to compute AUC graph properly. We use ADAM optimizer for optimization and binary crossentropy as lose function. Figure 3.9 shows our proposed CNN model.

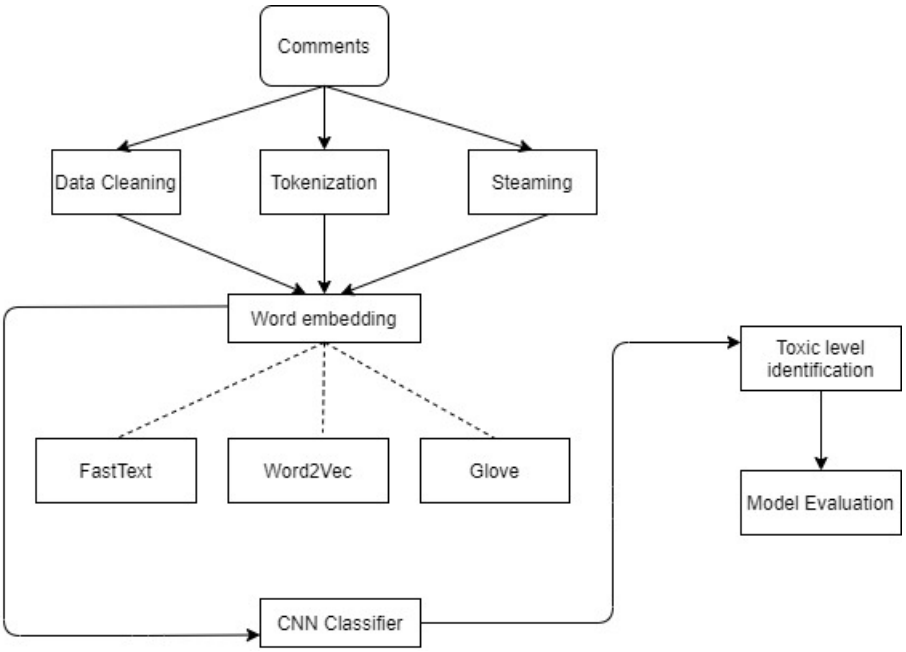


Figure 3.9: Our proposed CNN model

Extreme Gradient Boosting (XGBoost)

We have collected data set containing six different classes. This data set has almost 1,60,000 different types of comment. Using NLP algorithm, we have cleaned our dataset with steaming, lemmatization, stop word removal and punctuation removal features. We call WordNetLemmatizer library for lemmatizing our data. Then, we have separated the train and test features. After that, we make a bag of words and find the TF-IDF score of that words.For TF-IDF calculation, we call tfidfvectorizer function. Then we split our train data and test data.Then we have used XGBoost classifier. In XGBoost classifier, we prepared AUC score for comparing between the true values and predicted values.Then, we call a match function to ensure the matching true values and predicted values. Finally,we evaluate our model.Figure 3.10 shows our proposed XGBoost model.

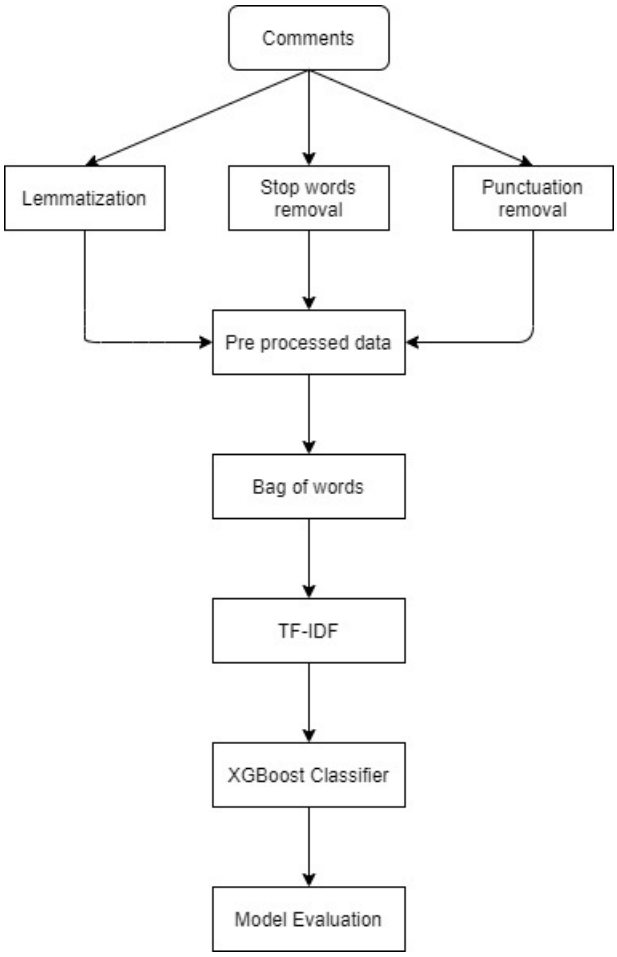


Figure 3.10: Our proposed XGBoost model

Support Vector Machine (SVM):
We have collected data set containing six different classes. This data set has almost 1,60,000 different types of comment. Using NLP algorithm, we have cleaned our dataset with tokenization, steaming, lemmatization, stop word removal and punctuation removal features. We call WordNetLemmatizer library for lemmatizing our data and tokenizer to tokenize our data. Then, we have separated the train and test features. After that, we make a bag of words and find the TF-IDF score of that words.For TF-IDF calculation, we call tfidfvectorizer function. Then we split our train data and test data.Then we have used linear SVM classifier. In SVM classifier, we prepared AUC score for comparing between the true values and predicted values.We have used GridSearchCV (Cross Validation) for differentiate train data and test data. Finally,we evaluate our model. Figure 3.11 shows our proposed SVM model.

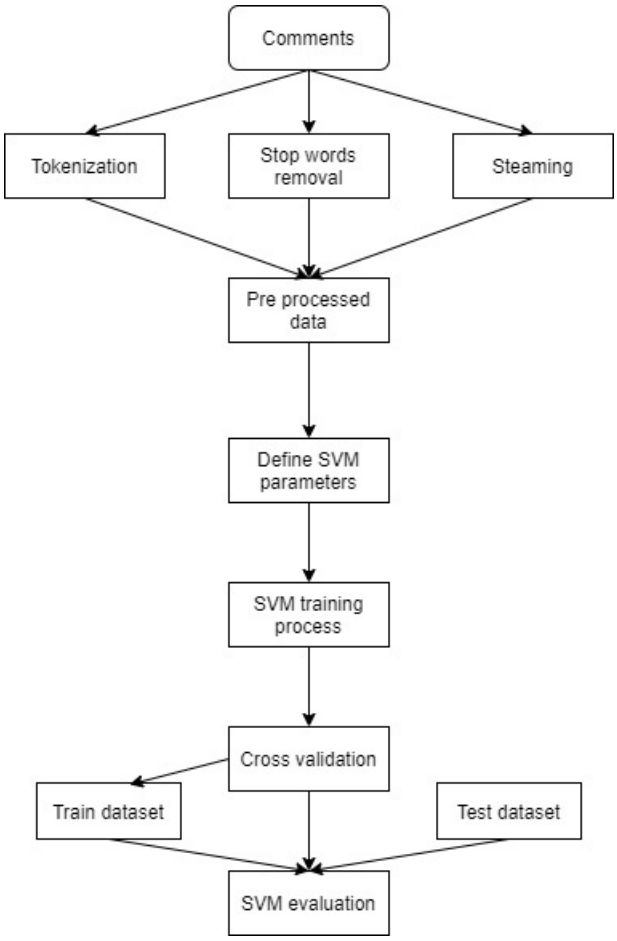


Figure 3.11: Our proposed XGBoost model

And finally, we have calculated the hamming score of all three models and also made an overall performance of precision, recall and f1 score function.

Chapter 4

Result and Analysis

In this part, the result of our proposed models and analysis will be talked about. Here the outcome segment will contain the general execution of the proposed model with some presentation boundary. The analysis part will have a comparison of our results, and evaluation of our results, and an elaborate discussion on our overall performance and model.

4.1 Result

In any analysis or project, result's the foremost very important portion because it shows the end result or findings of any analysis or model. during this portion, we are going to assess the results through some metrics of performance and show them mistreatment graphical illustration. The major metrics that are used here are:

- Data Cleaning
- Data Representation
- Classification Report

4.1.1 Data cleaning

Data Cleaning implies the way toward distinguishing the off base, fragmented, incorrect, superfluous, or missing aspect of the Data and afterward adjusting, supplanting, or erasing them as indicated by the need [34]. As a part of data processing, we also did data cleaning in our data set. The following bar-chart in figure 4.1 shows that, not all the comments are toxic. It fetches those comments and cleaned out those non-toxic comments, and remaining toxic comments are divided into six classes.

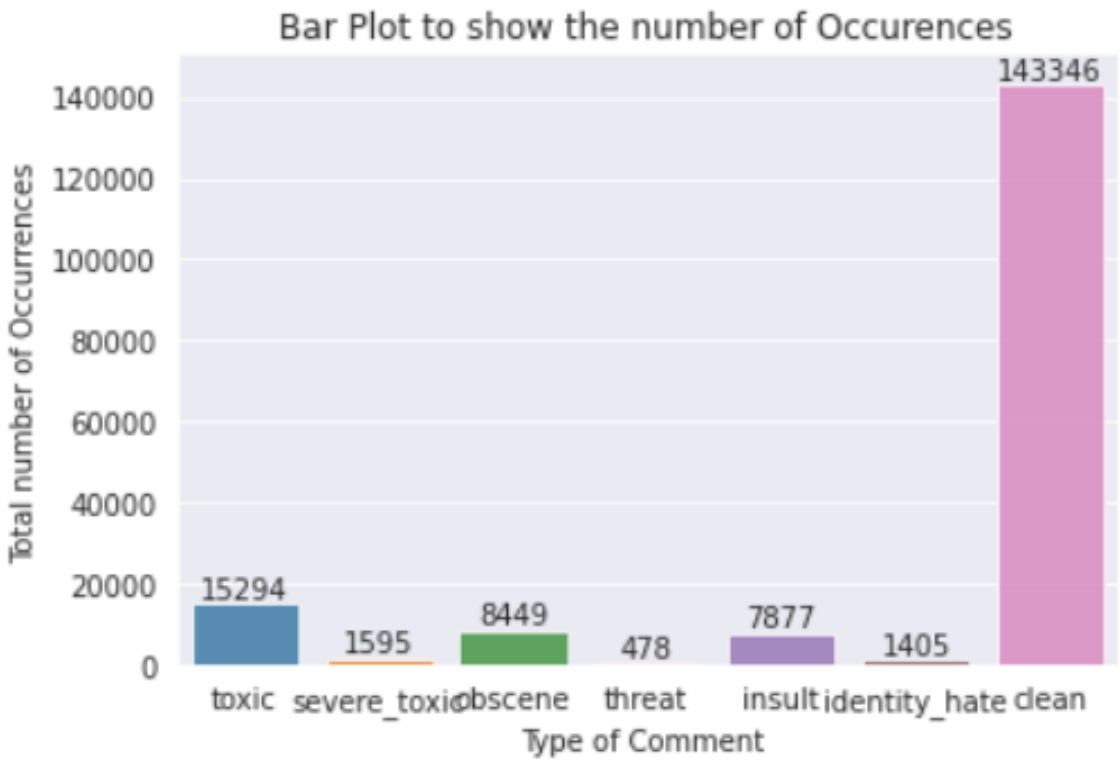


Figure 4.1: Number of comment occurrences

Natural language processing (NLP) is an extremely active field of research. It gives the likelihood to examine human language from the applied side, not simply hypothetically, and to attempt to explain some of the assignments thinking about human language. Python and the Natural Language Toolkit (NLTK) permit any software engineer, even an amateur, to get to know NLP errands effectively without investing a lot of energy in contemplating or gathering assets. The point of this paper is to give significant evidence and models, which show how essential the NLTK is for the course of Computational Linguistics at the college and for analysts in the field of characteristic language handling[35]. A comment can have a place with these classes or a subset of these classifications, which makes it a multi-label order issue. The following figure 4.2 will show multiple tags per comment.

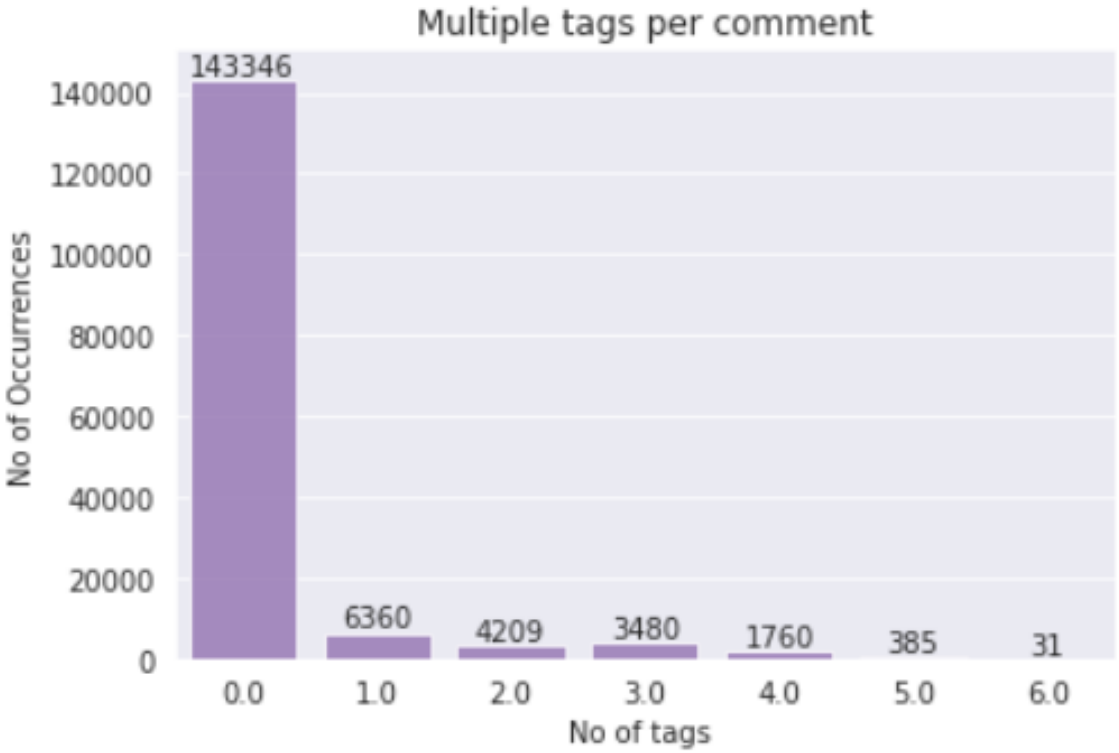


Figure 4.2: Multiple tags per comment

In data recovery, TF-IDF, short for term repetition converse report repetition, could be a mathematical measure that's projected to replicate however important a word is to a record in an assortment or corpus. it's often utilised as a weight think about hunts of knowledge recovery, text mining, and consumer displaying. Based on TF-IDF score, we have got some bar-charts for all the classes included in our data set.

Figure 4.3 shows the TF-IDF score of some words in 'toxic' class. Here we can see the word 'fuck' has 0.08 TF-IDF score, 'shit' has 0.04 TF-IDF score, 'ass' has 0.03 TF-IDF score in terms of toxic class.

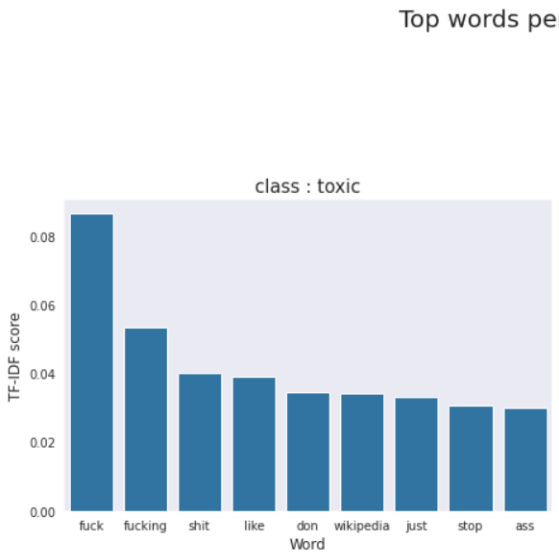


Figure 4.3: TF-IDF score (toxic)

Figure 4.4 shows the TF-IDF score of some words in 'severe_toxic' class. Here we can see the word 'fuck' has 0.25 TF-IDF score, 'shit' has 0.07 TF-IDF score, 'ass' has 0.05 TF-IDF score in terms of severe_toxic class.

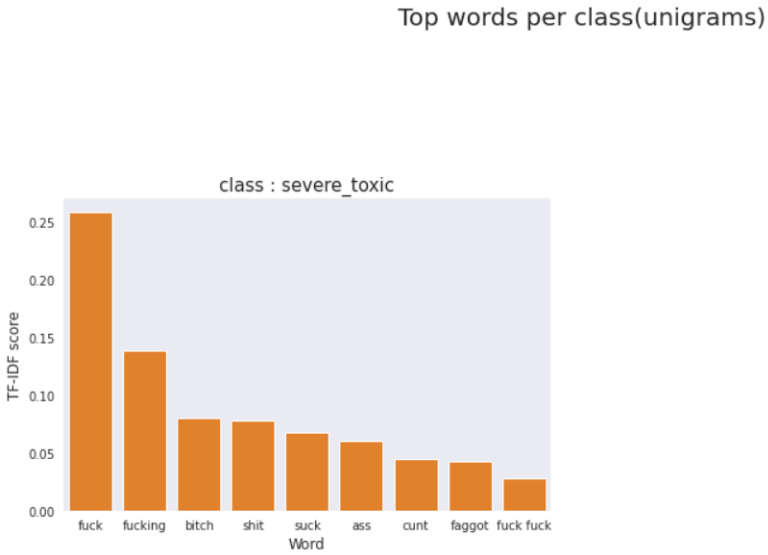


Figure 4.4: TF-IDF score (severe_toxic)

Figure 4.5 shows the TF-IDF score of some words in 'obscene' class. Here we can see the word 'fuck' has 0.15 TF-IDF score, 'shit' has 0.06 TF-IDF score, 'ass' has 0.05 TF-IDF score in terms of obscene class.

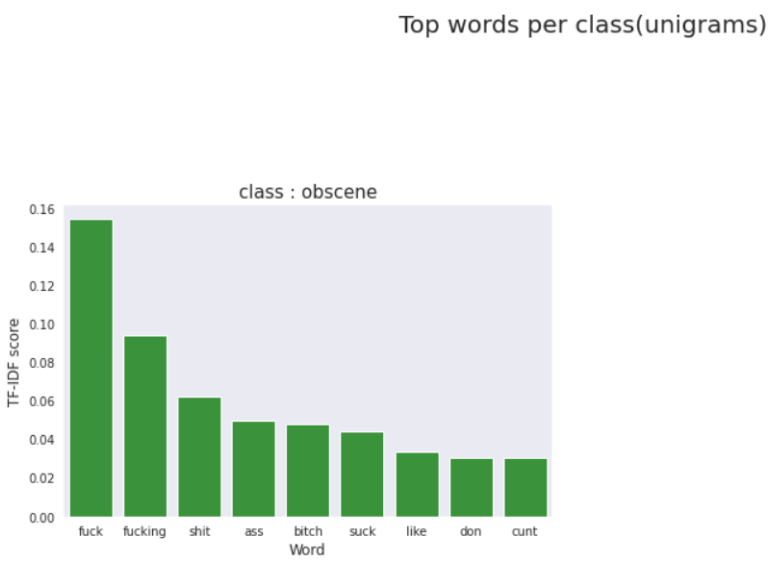


Figure 4.5: TF-IDF score (obscene)

Figure 4.6 shows the TF-IDF score of some words in 'threat' class. Here we can see the word 'fuck' has 0.08 TF-IDF score, 'shit' has 0.05 TF-IDF score, 'ass' has 0.06 TF-IDF score in terms of threat class.

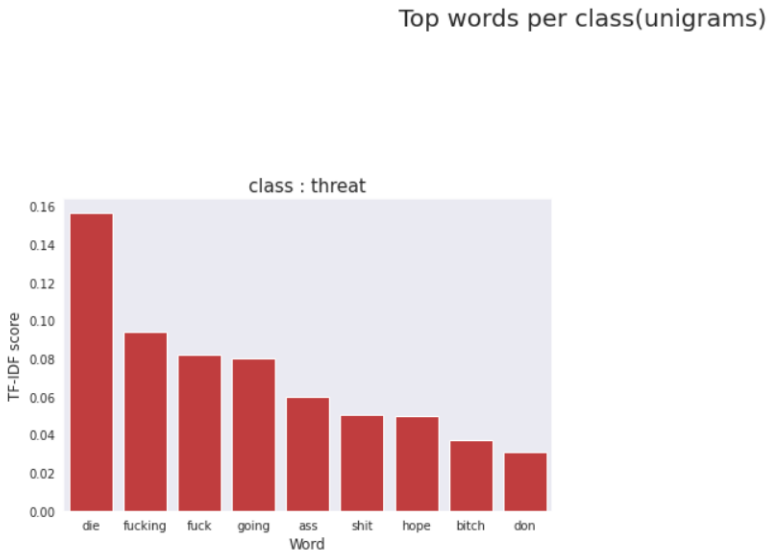


Figure 4.6: TF-IDF score (threat)

Figure 4.7 shows the TF-IDF score of some words in ‘insult’ class. Here we can see the word ‘fuck’ has 0.13 TF-IDF score, ‘shit’ has 0.05 TF-IDF score, ‘ass’ has 0.04 TF-IDF score in terms of insult class.

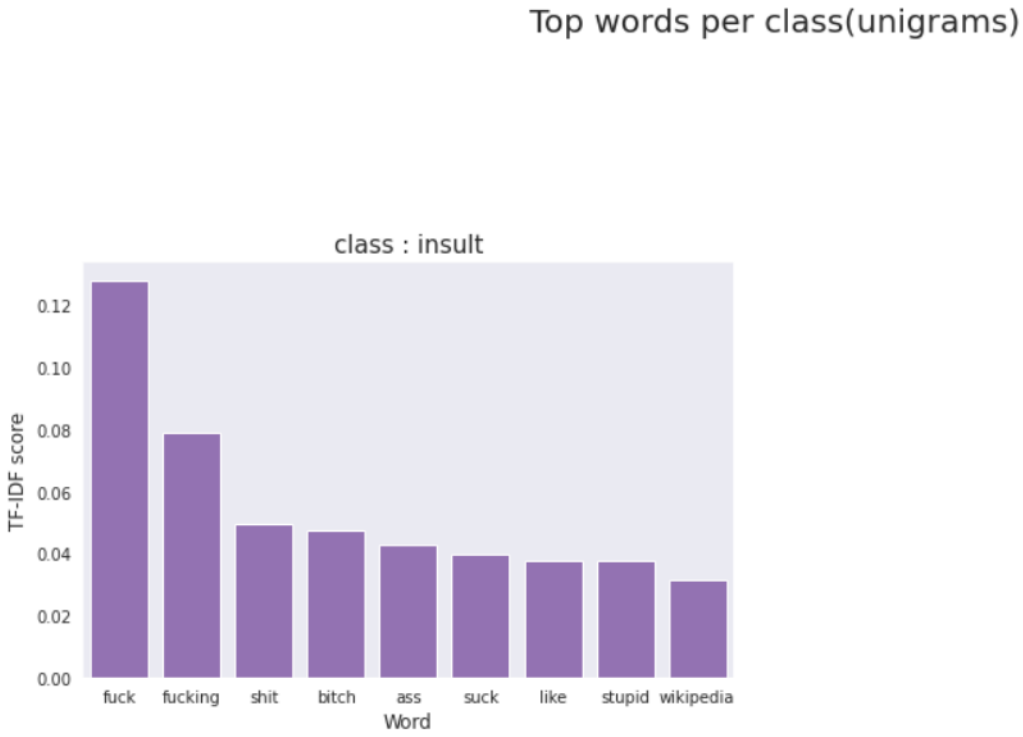


Figure 4.7: TF-IDF score (insult)

Figure 4.8 shows the TF-IDF score of some words in ‘identity_hate’ class. Here we can see the word ‘fuck’ has 0.11 TF-IDF score, ‘shit’ has 0.05 TF-IDF score, ‘ass’ has 0.04 TF-IDF score in terms of identity_hate class.

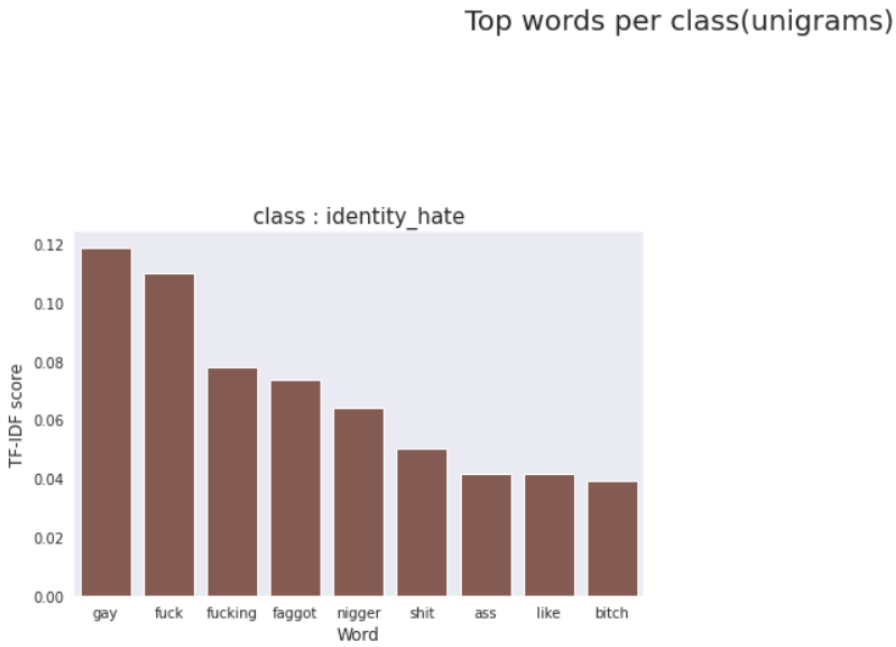


Figure 4.8: TF-IDF score (identity_hate)

Based on TF-IDF score,we have showed some bar-charts for all the classes included in our dataset.

4.1.2 Data representation

The dataset contains total six classes (toxic, severe_toxic, obscene, insult, threat, identity_hate) which are described down below in Figure 4.9:

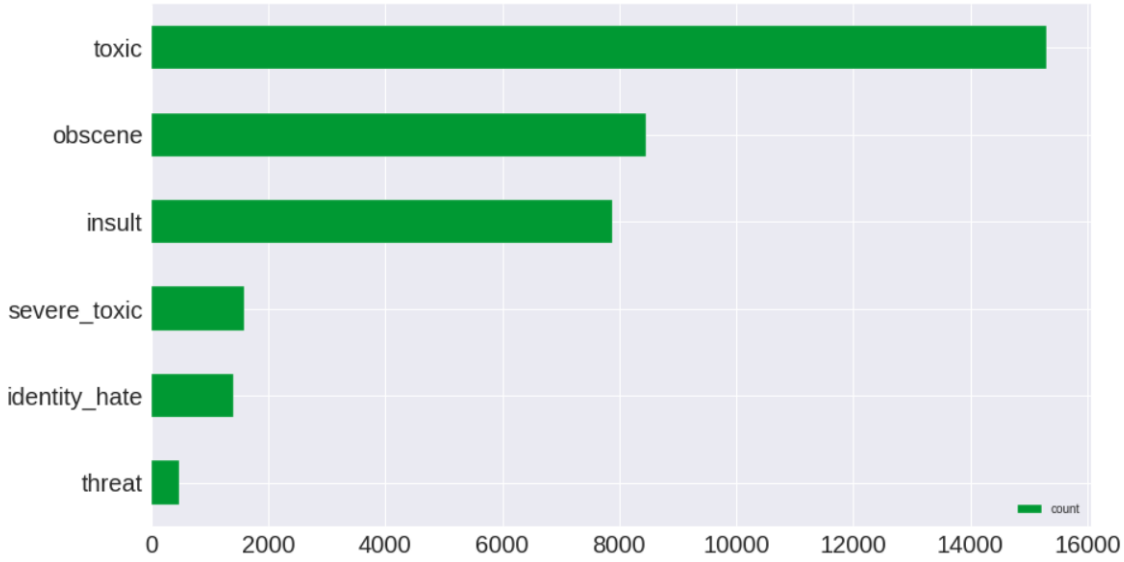


Figure 4.9: Data representation in Bar-Chart

The correlation matrices in Figure 4.10 shows that 'Toxic, comments are most strongly correlated with 'insult' and 'obscene' class. Moreover 'Toxic' and 'Thread' have the only weak correlation. Further, there is very weak correlation between 'Obscene' and 'insult' comments are also highly correlated, which makes perfect sense. It also shows the class 'threat' has the weakest correlation with all classes.

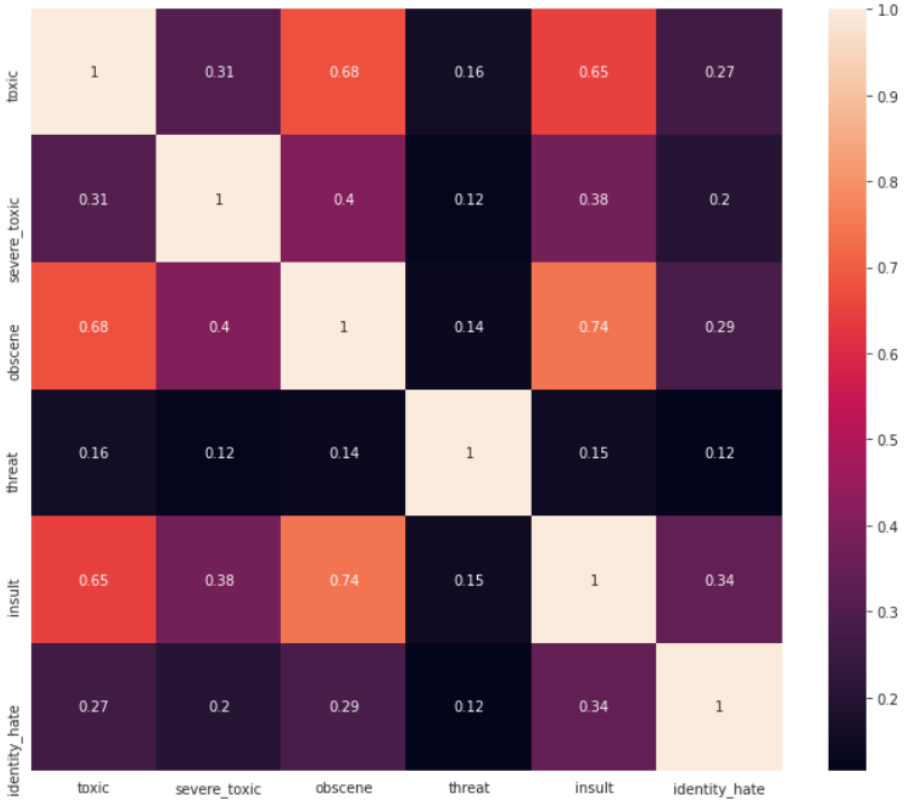


Figure 4.10: Visual representation of correlation between classes

There are lots of comments containing in our data set. These comments contain some common toxic words also. Because of this re occurrence of words, we have tried to show them in Word Cloud. These Word Cloud images will show those common toxic words that are using in day to day life. Figure 4.11 is an example of WordCloud of our dataset.

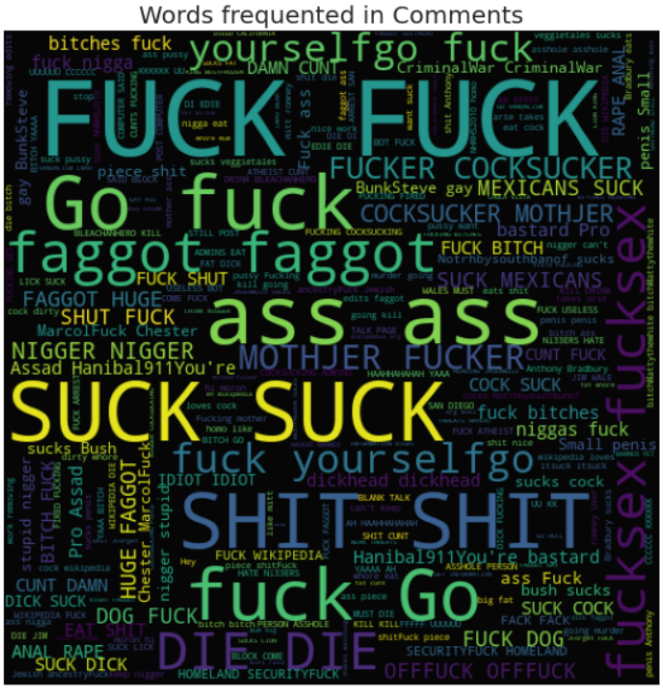


Figure 4.11: Word Cloud

4.1.3 Classification Report

In classification problem, confusion matrix indicates performance measurement. It shows the confused condition of a model during prediction. The output can be two or more than two depending on the problem statement.

In table 4.1 of confusion matrix, some general terms like True positive (TP), False positive (FP), True negative (TN) and False negative (FN) are used[36]. As confusion matrix shows the overall prediction of a machine learning model, this is a very important parameter to measure performance of the model as accuracy alone is not the best measurement of performance for a model. Here we have showed three confusion matrices for our data set using the three proposed models.

	Predicted Values		
		Negative(0)	Positive(1)
Actual Values	Negative(0)	TN	FP
	Positive(1)	FN	TP

Table 4.1: Confusion Matrix

Figure 4.12 shows the confusion matrix for the model using CNN where we have used our dataset. This dataset contains only toxic and non-toxic labelled data. In the matrix, we have true values on Y axis and predicted values on the X axis. The values from the matrix represents as below-

TP (true positive): Here we get 0.71 or 71% as our true positive which indicates at 71% cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

TN (true negative): Here we get 0.96 or 96% as our true positive which indicates at 96% cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

FP (false positive): Here we get 0.04 or 4% as our true positive which indicates at 4% cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

FN (false negative): Here we get 0.29 or 29% as our true positive which indicates at 29% cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

So, the proposed CNN model can detect toxic 71% correctly from the text and for the non-toxic one it is 96%.

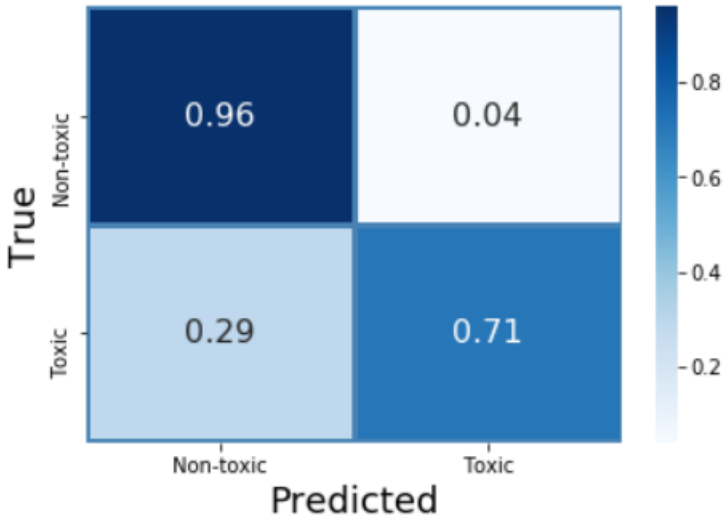


Figure 4.12: Confusion Matrix for CNN model

Figure 4.13 shows the confusion matrix for the model using XGBoost where we have used our dataset. This dataset contains only toxic and non-toxic labelled data. In the matrix, we have true values on Y axis and predicted values on the X axis. The values from the matrix represents as below-

TP (true positive): Here we get 0.53 or 53% as our true positive which indicates at 53% cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

TN (true negative): Here we get 0.98 or 98% as our true positive which indicates at 98% cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

FP (false positive): Here we get 0.02 or 2% as our true positive which indicates at 2% cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

FN (false negative): Here we get 0.47 or 47% as our true positive which indicates at 47% cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

So, the proposed XGBoost model can detect toxic 53% correctly from the text and for the non-toxic one it is 98%.

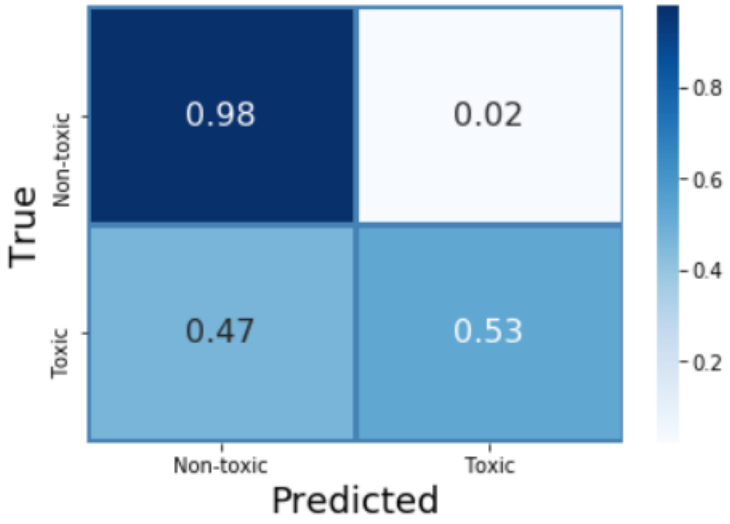


Figure 4.13: Confusion Matrix for XGBoost model

Figure 4.14 shows the confusion matrix for the model using XGBoost where we have used our dataset. This dataset contains only toxic and non-toxic labelled data. In the matrix, we have true values on Y axis and predicted values on the X axis. The values from the matrix represents as below-

TP (true positive): Here we get 0.78 or 78% as our true positive which indicates at 78% cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

TN (true negative): Here we get 0.94 or 94% as our true positive which indicates at 94% cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

FP (false positive): Here we get 0.06 or 6% as our true positive which indicates at 6% cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

FN (false negative): Here we get 0.22 or 22% as our true positive which indicates at 22% cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

So, the proposed SVM model can detect toxic 78% correctly from the text and for the non-toxic one it is 94%.

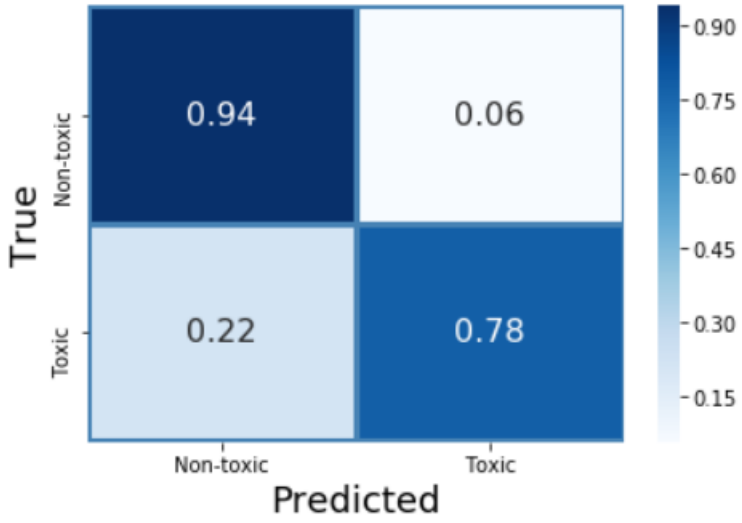


Figure 4.14: Confusion Matrix for SVM model

Classification report shows the quality of predictions through some calculations. More precisely, it calculates the true positive, true negative, false positive and false negative values from the confusion matrix to make classification report. In this classification report Precision, recall and f-1 score are the main metrics. Precision shows the percentage of being right in prediction. Where recall shows the total relevant results that classified correctly by the model or algorithm.

Precision: Precision is a popular parameter to measure the performance of any model. It shows that how much the result of a model is relevant. We can get precision by following simple equation which uses the data from confusion matrix. The equation looks like below:

$$Precision = \left(\frac{TP}{TP + FP}\right) \tag{4.1}$$

Using CNN, XGBoost and SVM we got precision for each class of our dataset. They will be discussed below their respective Figures.

Recall: Recall shows how much relevant results are correctly classified by a model. Recall metric calculates what percentage of the particular Positives are captured by labeling it as Positive (True Positive) by the model. Like accuracy and precision, recall is also assumed to be an important parameter to analysis the performance of any model. This value come from confusion matrix and the equation looks like below:

$$Recall = \left(\frac{TP}{TP + FN}\right) \tag{4.2}$$

We have obtained recall values for each class of the dataset on the proposed models using CNN (convolutional neural network), XGBoost and SVM. They will be discussed individually below their respective figure.

F1-score:F1 score is the function of precision and recall which measures the weighted average of precision and recall. F1 score calculates precision and recall at the same time by using harmonic mean instead of arithmetic mean. So, this is another important index for performance evaluation:

$$F1 - score = \left(\frac{2 * Recall * Precision}{Recall + Precision} \right) \tag{4.3}$$

We have calculated f1-score for each class of the dataset on the proposed models using CNN (convolutional neural network), XGBoost and SVM. They will be discussed individually below their respective figure.

Figure 4.15 shows the classification report with precision, recall and f1 score using CNN as classification model. Here precision, recall and f1 score for toxic is 0.938, 0.935 and 0.937 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.993 and 0.992 respectively, then precision, recall and f1 score for 'obscene' is 0.963, 0.965 and 0.964 respectively, then precision, recall and f1 score for 'threat' is 0.995, 0.996 and 0.996 respectively, then precision, recall and f1 score for 'insult' is 0.960, 0.964 and 0.961 respectively and then precision, recall and f1 score for 'identity_hate' is 0.988, 0.990 and 0.988 respectively.

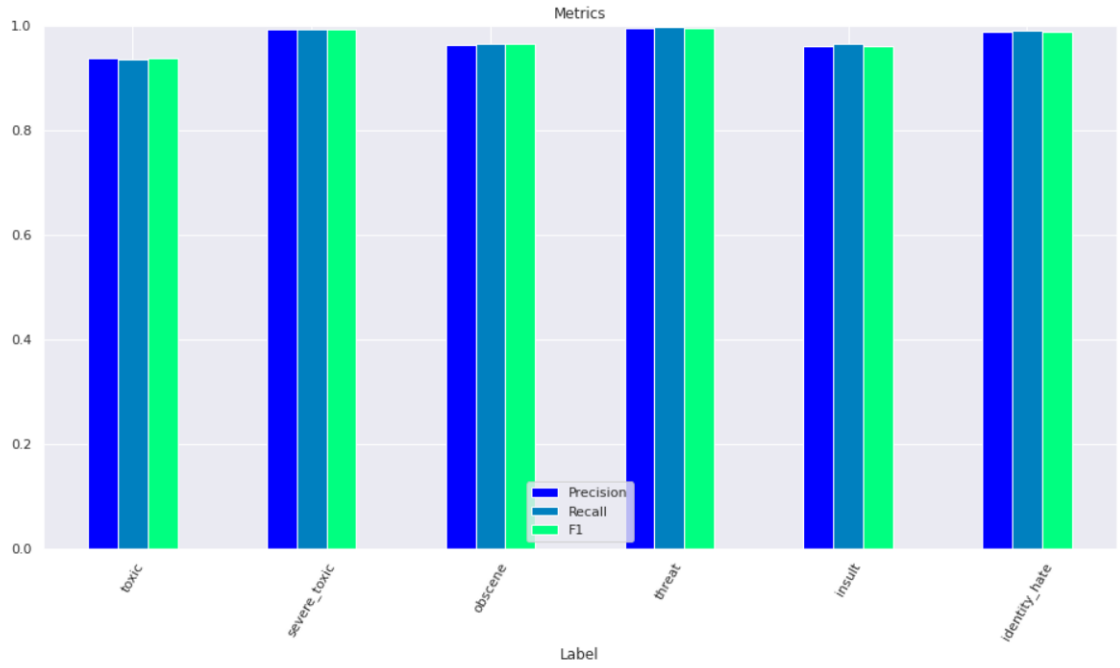


Figure 4.15: Classification report with precision, recall and f1 score on CNN model

Figure 4.16 shows the classification report with precision, recall and f1 score using XGBoost as classification model. Here precision, recall and f1 score for toxic is 0.929, 0.935 and 0.930 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.994 and 0.992 respectively, then precision, recall and f1 score for 'obscene' is 0.959, 0.963 and 0.958 respectively, then precision, recall and f1 score for 'threat' is 0.993, 0.996 and 0.995 respectively, then precision, recall and f1 score for 'insult' is 0.954, 0.960 and 0.953 respectively and then precision, recall and f1 score for 'identity_hate' is 0.985, 0.988 and 0.983 respectively.

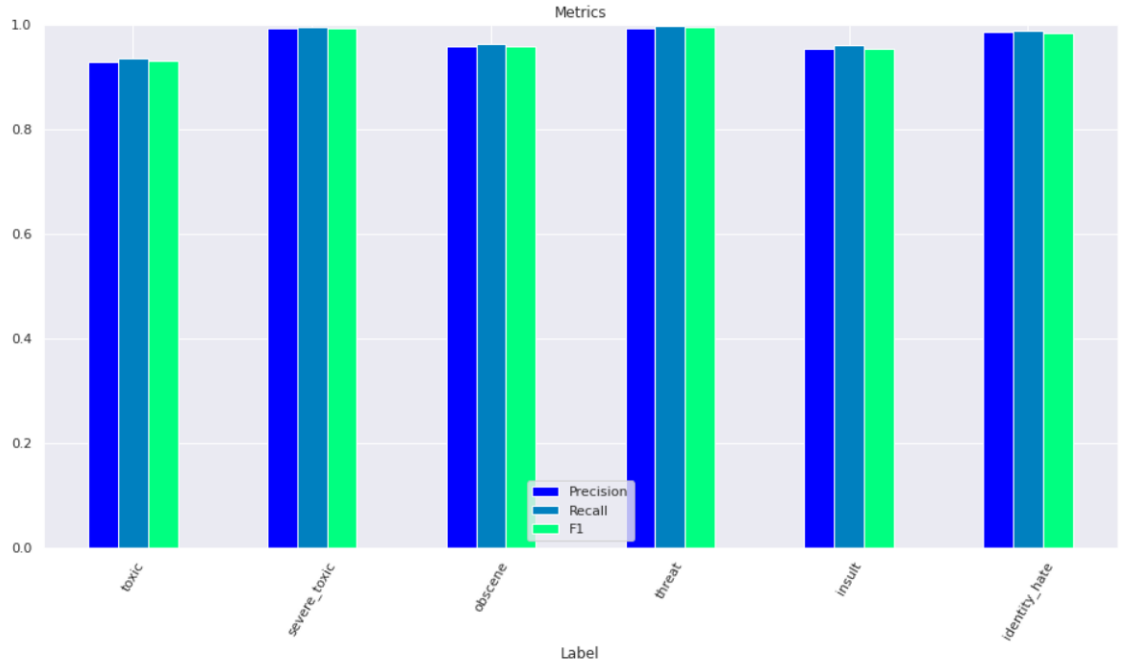


Figure 4.16: Classification report with precision, recall and f1 score on XGBoost model

Figure 4.17 shows the classification report with precision, recall and f1 score using SVM as classification model. Here precision, recall and f1 score for toxic is 0.938, 0.925 and 0.929 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.993 and 0.993 respectively, then precision, recall and f1 score for 'obscene' is 0.962, 0.963 and 0.962 respectively, then precision, recall and f1 score for 'threat' is 0.995, 0.996 and 0.995 respectively, then precision, recall and f1 score for 'insult' is 0.958, 0.961 and 0.959 respectively and then precision, recall and f1 score for 'identity_hate' is 0.988, 0.990 and 0.989 respectively.

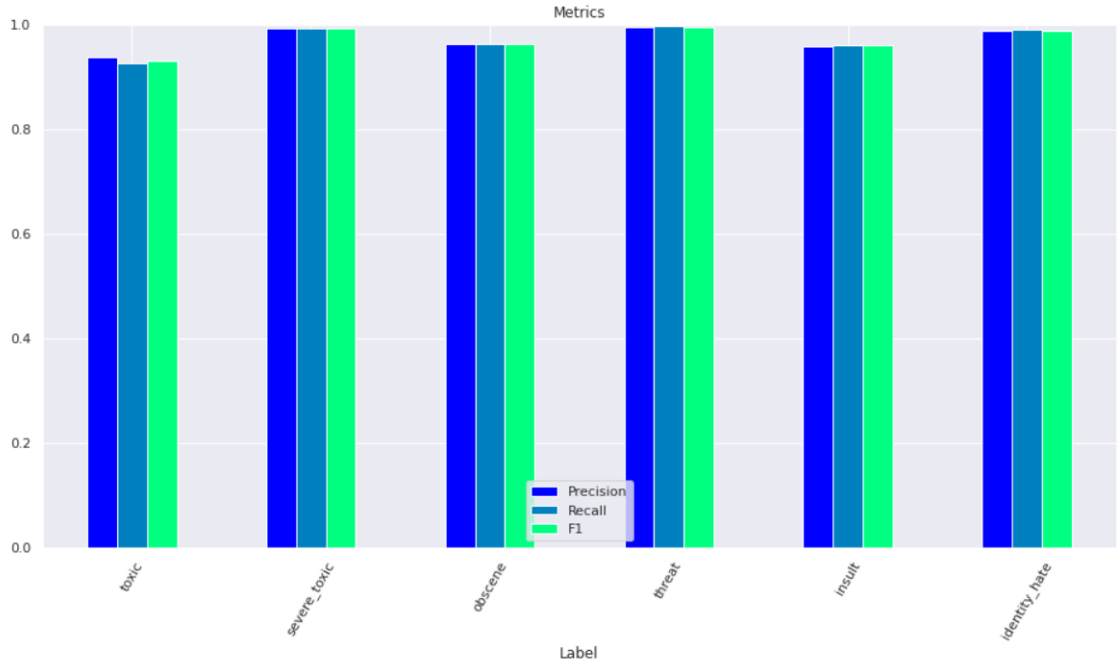


Figure 4.17: Classification report with precision, recall and f1 score on SVM model

We know that, AUC measures values from zero to one. A model whose forecasts are 100% wrong has associate AUC score is 0; one whose expectations are 100% right has associate AUC score is 1.0. We have tried to calculate the AUC score of three different algorithms separately along with six different classes that contains our dataset.

The below figure 4.18 shows the graph of TPR (true positive rate) vs FPR (false positive rate) of each class that has been labeled in our dataset. The result of this classes is, toxic AUC = 0.93, severe_toxic AUC = 0.98, obscene AUC = 0.96, threat AUC = 0.96, insult AUC = 0.95, identity_hate AUC = 0.95. And finally, we got the mean AUC of CNN model is 0.96.

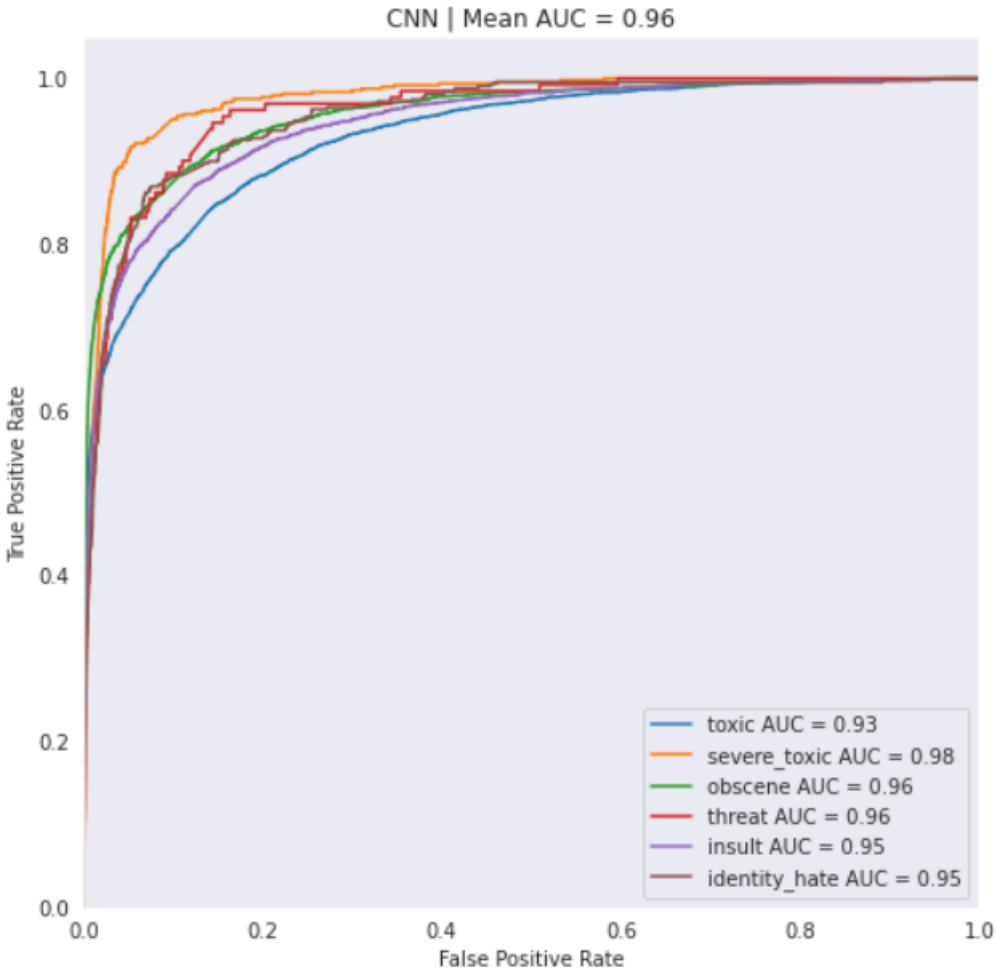


Figure 4.18: AUC graph for CNN model

The below figure 4.19 shows the graph of TPR (true positive rate) vs FPR (false positive rate) of each class that has been labeled in our dataset. The result of this classes is, toxic AUC = 0.89, severe_toxic AUC = 0.95, obscene AUC = 0.93, threat AUC = 0.88, insult AUC = 0.91, identity_hate AUC = 0.91. And finally, we got the mean AUC of XGBoost model is 0.91.

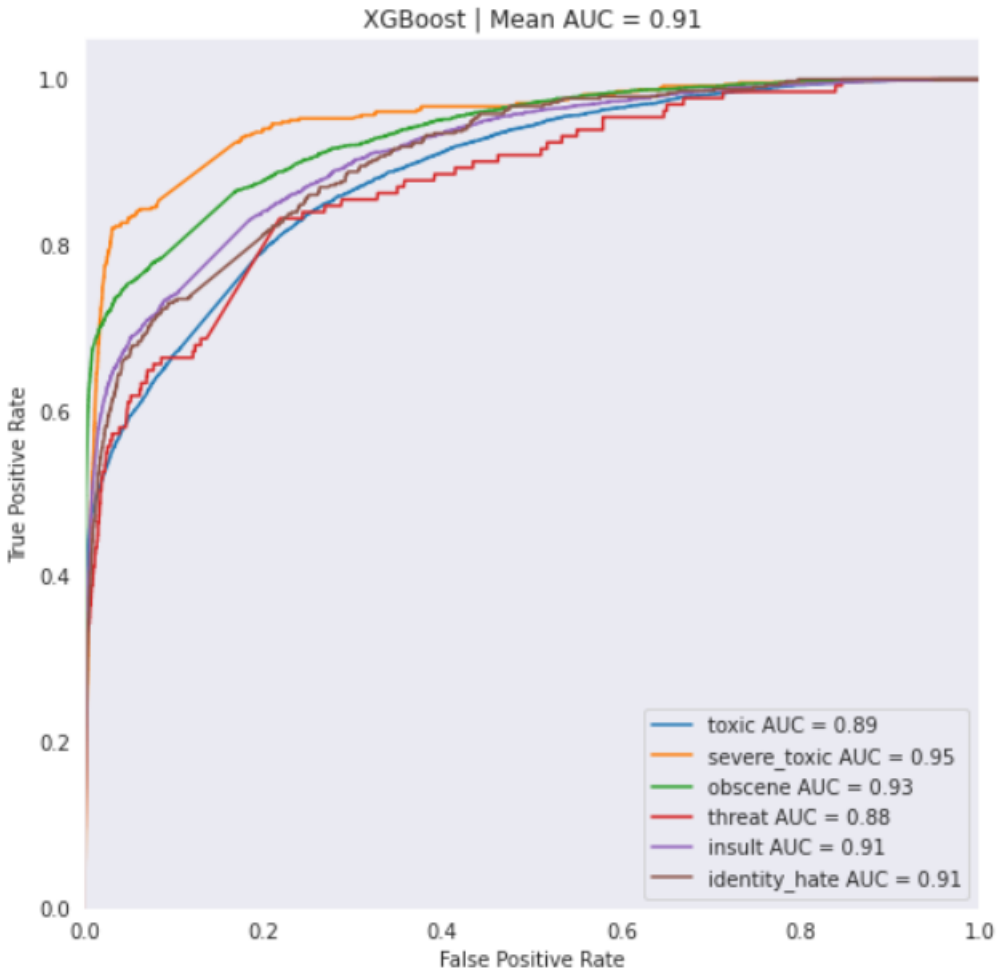


Figure 4.19: AUC graph for XGBoost model

The below figure 4.20 shows the graph of TPR (true positive rate) vs FPR (false positive rate) of each class that has been labeled in our dataset. The result of this classes is, toxic AUC = 0.90, severe_toxic AUC = 0.97, obscene AUC = 0.94, threat AUC = 0.91, insult AUC = 0.91, identity_hate AUC = 0.91. And finally, we got the mean AUC of SVM model is 0.92.

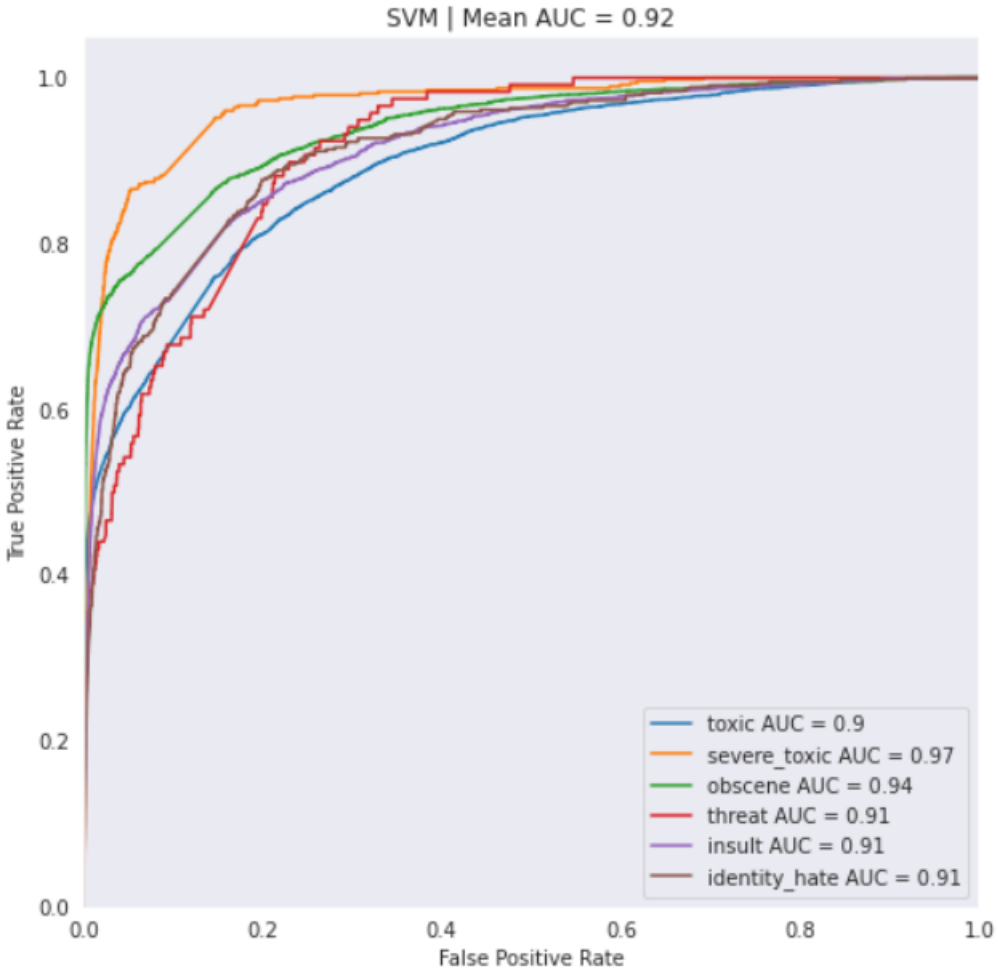


Figure 4.20: AUC graph for SVM model

4.2 Analysis

We proposed three models for toxic word classification from social media post or comments. One is using CNN (convolutional neural network), second one is by using XGBoost and last one is SVM. Then we took a set of data to evaluate our model. Here comparative performance tables have been given which shows a comparative overview of model performance:

Table 4.2 shows the classification report with precision, recall and f1 score using CNN as classification model. Here precision, recall and f1 score for toxic is 0.938, 0.935 and 0.937 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.993 and 0.992 respectively, then precision, recall and f1 score for 'obscene' is 0.963, 0.965 and 0.964 respectively, then precision, recall and f1 score for 'threat' is 0.995, 0.996 and 0.996 respectively, then precision, recall and f1 score for 'insult' is 0.960, 0.964 and 0.961 respectively and then precision, recall and f1 score for 'identity_hate' is 0.988, 0.990 and 0.988 respectively.

CNN	Toxic	Severe_toxic	Obscene	Threat	Insult	Identity_hate
Precision	0.938	0.992	0.963	0.995	0.960	0.988
Recall	0.935	0.993	0.965	0.996	0.964	0.990
F1-score	0.937	0.992	0.964	0.995	0.961	0.988

Table 4.2: CNN Analysis table

Table 4.3 shows the classification report with precision, recall and f1 score using XGBoost as classification model. Here precision, recall and f1 score for toxic is 0.929, 0.935 and 0.930 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.994 and 0.992 respectively, then precision, recall and f1 score for 'obscene' is 0.959, 0.963 and 0.958 respectively, then precision, recall and f1 score for 'threat' is 0.993, 0.996 and 0.995 respectively, then precision, recall and f1 score for 'insult' is 0.954, 0.960 and 0.953 respectively and then precision, recall and f1 score for 'identity_hate' is 0.985, 0.988 and 0.983 respectively.

XGBoost	Toxic	Severe_toxic	Obscene	Threat	Insult	Identity_hate
Precision	0.929	0.992	0.959	0.993	0.954	0.985
Recall	0.935	0.994	0.962	0.996	0.960	0.988
F1-score	0.930	0.992	0.957	0.995	0.953	0.983

Table 4.3: XGBoost Analysis table

Table 4.4 shows the classification report with precision, recall and f1 score using SVM as classification model. Here precision, recall and f1 score for toxic is 0.938, 0.925 and 0.929 respectively, then precision, recall and f1 score for 'severe_toxic' is 0.992, 0.993 and 0.993 respectively, then precision, recall and f1 score for 'obscene' is 0.962, 0.963 and 0.962 respectively, then precision, recall and f1 score for 'threat' is 0.995, 0.996 and 0.995 respectively, then precision, recall and f1 score for 'insult' is 0.958, 0.961 and 0.959 respectively and then precision, recall and f1 score for 'identity_hate' is 0.988, 0.990 and 0.989 respectively.

XGBoost	Toxic	Severe_toxic	Obscene	Threat	Insult	Identity_hate
Precision	0.938	0.992	0.962	0.995	0.958	0.988
Recall	0.925	0.992	0.963	0.996	0.961	0.990
F1-score	0.929	0.992	0.962	0.995	0.959	0.989

Table 4.4: SVM Analysis table

Hamming score is the measurement of relevant words in train and test dataset. On the other hand, hamming loss is the opposite of hamming score. Because it shows the measurement of mismatch of words in dataset. We have calculated hamming score separately for CNN, XGBoost and SVM model. In table 4.5, it shows the differences of hamming score and hamming loss among CNN, XGBoost and SVM.

Model	Hamming Score	Hamming Loss
CNN	0.89	0.11
XGBoost	0.87	0.13
SVM	0.84	0.16

Table 4.5: Hamming score and loss analysis

We also created a bar-chart as shown in figure 4.21 to show the overall differences of hamming score and hamming loss between these three algorithms.

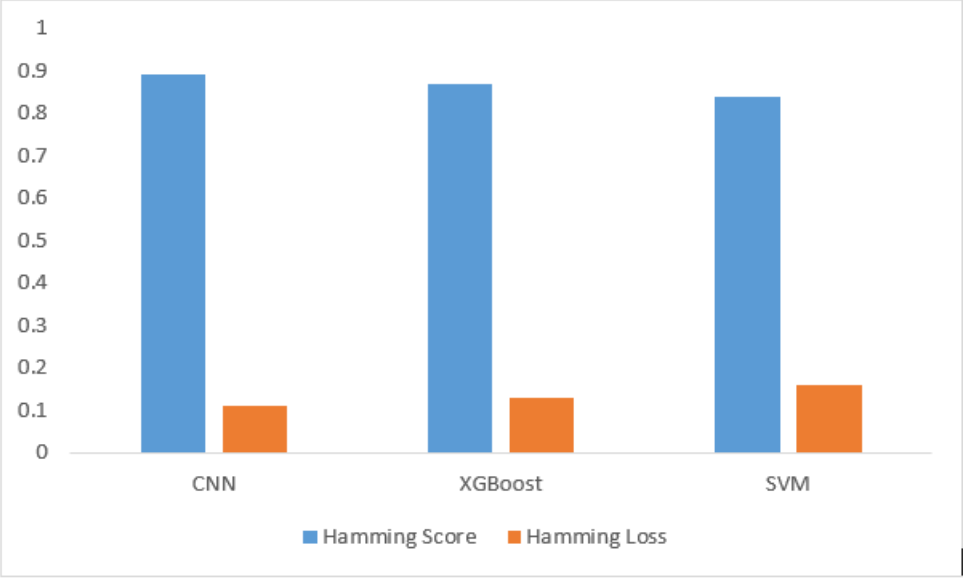


Figure 4.21: Hamming score and Hamming loss of our proposed model

We know that, AUC measures values from zero to one. A model whose forecasts are 100% wrong has associate AUC score is 0; one whose expectations are 100% right has associate AUC score is 1.0. We have tried to calculate the AUC score of three different algorithms separately along with six different classes that contains our dataset. Finally, we have calculated the mean AUC of all three models. Table 4.6 shows the AUC score of all three algorithms that are given below:

Model	AUC
CNN	0.96
XGBoost	0.91
SVM	0.92

Table 4.6: Model vs AUC analysis

We also created a bar-chart of AUC score vs Model in figure 4.22. The graphical representation of AUC score for all the three algorithms are given below:

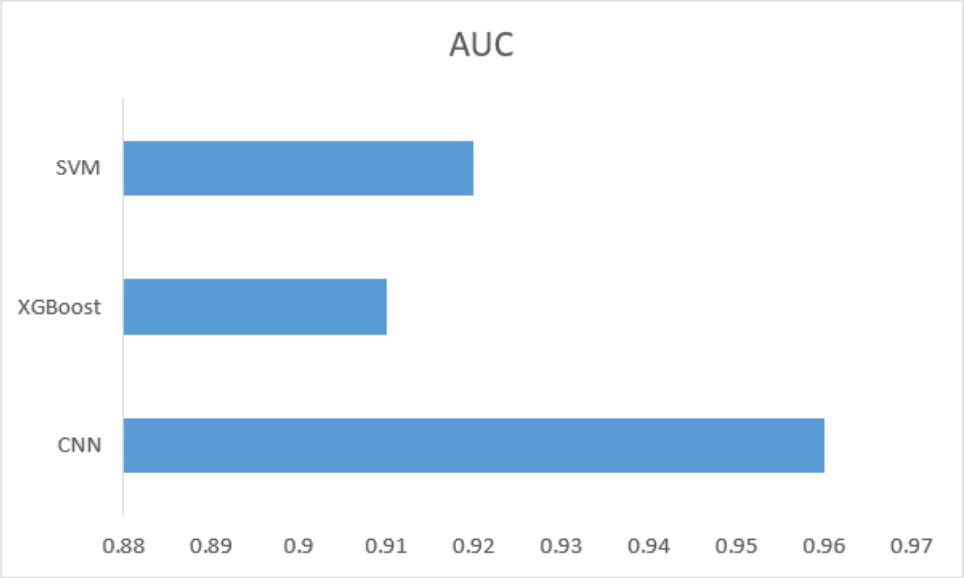


Figure 4.22: Model vs AUC representation

From our analysis, we have collected the average of precision, recall and f1-score for each algorithm. In table 4.7, the comparison of all the values are shown:

Model	Precision	Recall	F1-score
CNN	0.973	0.974	0.973
XGBoost	0.969	0.973	0.968
SVM	0.972	0.971	0.971

Table 4.7: Comparison of precision, recall, f1-score

We also created a bar-chart of precision, recall and f1-score for every algorithm in figure 4.23. The graphical representation of precision, recall and f1-score for all the three algorithms are given below:

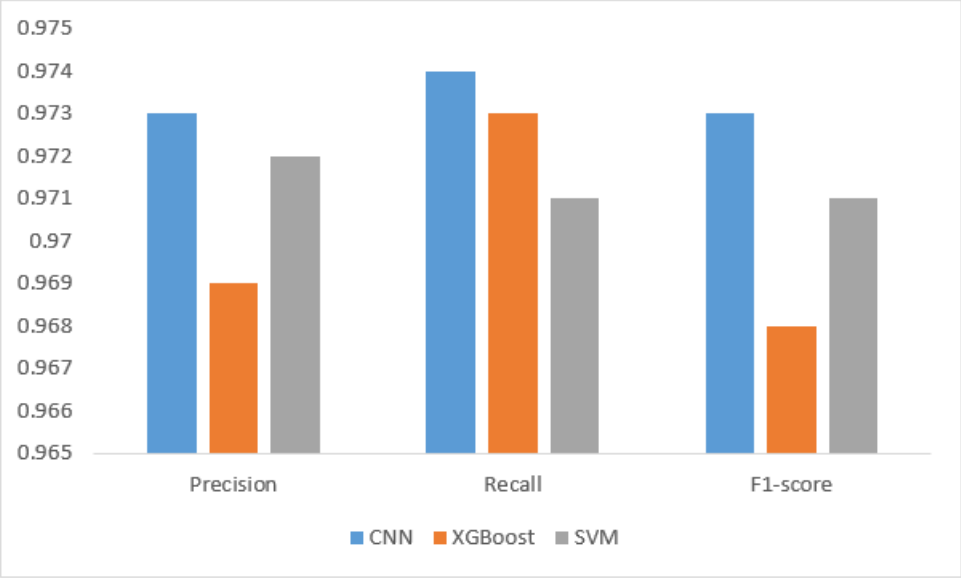


Figure 4.23: Comparison of precision, recall, f1-score on CNN, XGBoost and SVM

In figure 4.23, indicate the differences in terms of precision, recall and f1-score of CNN, XGBoost and SVM algorithm. Here, blue bar indicates CNN model, orange bar indicates XGBoost model and grey bar indicates SVM model.

Chapter 5

Conclusion

In this chapter, we have summarized our whole thesis report and also mentions some limitations and future works considering our work. So that, in future we can add more updated thing or can use other algorithms to get more precise result

5.1 Conclusion

In this paper, we have presented a toxic comment classification system which is an essential tool for social media sites. With the ever-expanding popularity and use of social media platforms, the numbers of vulgar and negative comments are also increasing. The system is also imperative to preclude the cyber bullying, toxic or offensive comments as addressing these issues are still grueling. We will be continuing our work to identify toxic comments by implementing CNN with FastText word embedding technique, XGBoost with XGBoost classifier and SVM with vector gradient classifier. Till now, to identify toxic comments with CNN, XGBoost and SVM we have prepared the data using natural language processing including data cleaning, tokenization, lemmatization, punctual removal, stop words removal and stemming. After that, we will be using CNN for two purposes. Firstly, we will classify that comments are toxic or non-toxic. On the second place, we will label the toxic classified comments into various subclasses. In the near future, it is also in our utmost interest to employ the system in social medias and teaching platform chat box as these two platforms are prone to encountering massive amounts of negativity and toxic comments.

5.1.1 Future work

For the future work, we will consider the following things in order to expand the work as there is still some scope.
- Creating a Bangla dataset and detecting toxic comment.
- Creating a large dictionary of hate word.
- A word tokenizer which will be effective for tokenizing comments that contain hidden profanity.
- We also intend to apply linear regression, LSTM, Random forest for more precise classification.

5.1.2 Scope and Limitations

Through this paper we have attempted to build up a system which can classify toxic comments from any web-based media posts. The intensity of online media is definitive to the point that it can make anything viral in a flicker of an eye. One can undoubtedly find various kinds of individuals regarding their race, identity, sexual orientation, religion and so forth So spreading any hate or toxic substance can be conceivable effectively through these stage. These can lead on to large occurrences like uproars, self-destructive endeavors even psychological warfare. In any case, all of these thing can be halted from the root by characterizing and recognizing the toxic or hate related content from web-based media. The quantity of clients in web-based media are expanding step by step and with the expanding number of clients the quantity of tweets and posts are similarly expanding. Controlling and observing this gigantic number of comments is a colossal work. Adjacent to the intricacy of natural language and the new method of utilizing contempt and toxic word make this kind of examination all the more testing.

Bibliography

[1] S. Livingstone, L. Haddon, J. Vincent, G. Mascheroni, and K. Ólafsson, "Net children go mobile: The uk report," 2014.

[2] C. Van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever, B. Verhoeven, G. De Pauw, W. Daelemans, and V. Hoste, "Automatic detection of cyberbullying in social media text," *PloS one*, vol. 13, no. 10, e0203794, 2018.

[3] T. Mowen, J. Brent, and A. Kupchik, "School crime and safety," *The handbook of measurement issues in criminology and criminal justice*, vol. 434, 2016.

[4] M. Ibrahim, M. Torki, and N. El-Makky, "Imbalanced toxic comments classification using data augmentation and deep learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2018, pp. 875–878.

[5] W. L. J. Cai J. Li and J. Wang, "Deep learning model used in text classification," *15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2018*, pp. 123–126, 2018. DOI: 10.1109/ICCWAMTIP.2018.8632592..

[6] A. O. G. Parthasarathy and P. Anderson, "Natural language processing pipeline for temporal information extraction and classification from free text eligibility criteria," *International Conference on Information Society (i-Society), Dublin, 2016*, pp. 120–121, 2016. DOI: 10.1109/i-Society.2016.7854192..

[7] Y. X. J. Li and H. Shi, "Bidirectional lstm with hierarchical attention for text classification," *IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019*, pp. 456–459, 2019. DOI: 10.1109/IAEAC47372.2019.8997969..

[8] S. Sel and D. Hanbay, "E-mail classification using natural language processing," *27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019*, pp. 1–4, 2019. DOI: 10.1109/SIU.2019.8806593..

[9] Y. Zheng, "An exploration on text classification with classical machine learning algorithm," *International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 2019*, pp. 81–85, 2019. DOI: 10.1109/MLBDBI48998.2019.00023.

[10] H. Liu, "Towards explainable nlp: A generative explanation framework for text classification," [Online]. Available: arXiv.org.

[11] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[12] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–6.

[13] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *European semantic web conference*, Springer, 2018, pp. 745–760.

[14] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the first workshop on abusive language online*, 2017, pp. 85–90.

[15] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental short sentences classification by using cnn deep learning model with fine tuned word2vec," *Procedia Computer Science*, vol. 167, pp. 1139–1147, 2020.

[16] Z. Zhang, Y. Zou, and C. Gan, "Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression," *Neurocomputing*, vol. 275, pp. 1407–1415, 2018.

[17] J. Lin, G. Tremblay-Taylor, G. Mou, D. You, and K. Lee, "Detecting fake news articles," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 3021–3025.

[18] S. Si, A. Datta, S. Banerjee, and S. K. Naskar, "Aggression detection on multilingual social media text," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2019, pp. 1–5.

[19] C. Rawat, A. Sarkar, S. Singh, R. Alvarado, and L. Rasberry, "Automatic detection of online abuse and analysis of problematic users in wikipedia," in *2019 Systems and Information Engineering Design Symposium (SIEDS)*, IEEE, 2019, pp. 1–6.

[20] M. Tlachac and E. A. Rundensteiner, "Depression screening from text message reply latency," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2020, pp. 5490–5493.

[21] A. Basu, C. Walters, and M. Shepherd, "Support vector machines for text categorization," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, IEEE, 2003, 7–pp.

[22] Z.-Q. Wang, X. Sun, D.-X. Zhang, and X. Li, "An optimal svm-based text classification algorithm," in *2006 International Conference on Machine Learning and Cybernetics*, IEEE, 2006, pp. 1378–1381.

[23] G. Paaß, J. Kindermann, and E. Leopold, "Text classification of news articles with support vector machines," in *Text Mining and its Applications*, Springer, 2004, pp. 53–64.

[24] F. Colas and P. Brazdil, "On the behavior of svm and some older algorithms in binary text classification tasks," in *International Conference on Text, Speech and Dialogue*, Springer, 2006, pp. 45–52.

[25] S. Kim and J. Choi, "An svm-based high-quality article classifier for systematic reviews," *Journal of biomedical informatics*, vol. 47, pp. 153–159, 2014.

[26] Y. Lin, H. Yu, F. Wan, and T. Xu, "Research on classification of chinese text data based on svm," in *IOP Conference Series: Materials Science and Engineering*, 2017, pp. 1–5.

[27] L. Wei, B. Wei, and B. Wang, "Text classification using support vector machine with mixture of kernel," *Journal of Software Engineering and Applications*, vol. 5, p. 55, 2012.

[28] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making", 2015.," 2015.

[29] S. Frenda and B. Somnath, "Deep analysis in aggressive mexican tweets, 2018.," 2018. [Online]. Available: http://hdl.handle.net/2318/1676282..

[30] H. B. S. A. Rahmah and Z. A. Hasibuan, "Exploring technology-enhanced learning key terms using tf-idf weighting," *Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia*, pp. 1–4, 2019. DOI: 10.1109/ICIC47613.2019.8985776.

[31] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959.

[32] B. Amrutha and K. Bindu, "Detecting hate speech in tweets using different deep neural network architectures," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 923–926.

[33] ——, "Detecting hate speech in tweets using different deep neural network architectures," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 923–926.

[34] V. G. A. D. Sarma, "Data cleaning: A practical perspective , morgan claypool," 2013. DOI: 10.2200/S00523ED1V01Y201307DTM036.

[35] S. H. J. O. Contreras and Z. B. Abubakar, "Automated essay scoring with ontology based on text mining and nltk tools," *International Conference on Smart Computing and Electronic Enterprise (ICSCEE), Shah Alam, 2018*, pp. 1–6, 2018. DOI: 10.1109/ICSCEE.2018.8538399.

[36] E. K. M. A. Wibowo Haryanto and Muljono, "Influence of word normalization and chi-squared feature selection on support vector machine (svm) text classification," *International Seminar on Application for Technology of Information and Communication, Semarang, 2018*, pp. 229–233, 2018. DOI: 10.1109/ISEMANTIC.2018.8549748.