NLP Models For Detecting Spam and Inappropriate Content On the Fly

Thesis

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Awanit Ranjan
181*CO*161
Pintu
181*CO*139
Vivek Kumar
181*CO*159

Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
January 2022

NLP Models For Detecting Spam and Inappropriate Content On the Fly

Thesis

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Awanit Ranjan
$181CO161$
Pintu
$181CO139$
Vivek Kumar
$181CO159$

Under the guidance of

Dr. Mahendra Pratap Singh
$AssistantProfessor$

Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
January 2022

DECLARATION

by the B.Tech. student

I hereby *declare* that the Thesis entitled **NLP Models For Detecting Spam and Inappropriate Content On the Fly** which is being submitted to **National Institute of Technology Karnataka, Surathkal**, in partial fulfilment for the requirements of the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** in the department of **Computer Science and Engineering**, is a *bonafide report of the work carried out by me.* The material contained in this report has not been submitted at any other University or Institution for the award of any degree.

181CO161, Awanit Ranjan
181CO139, Pintu
181CO159, Vivek Kumar

....
Place : NITK, Surathkal (Register Number, Name and Signature of the student)
Date : Department of Computer Science and Engineering

CERTIFICATE

This is to *certify* that the Thesis entitled **NLP Models For Detecting Spam and Inappropriate Content On the Fly** submitted by **SAwanit Ranjan** (Register number: 181CO161), **Pintu** (Register number: 181CO139) and **Vivek Kumar** (Register number: 181CO159) as the record of the work carried out by him, is *accepted as the P.G Major Project Thesis submission* in partial fulfilment for the requirements of the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** in the department of **Computer Science and Engineering** at **National Institute of Technology Karnataka, Surathkal** during the academic year 2021-2022.

....
Dr. Mahendra Pratap Singh
Project Guide
Department of CSE,
NITK Surathkal

....
Dr. Shashidhar G Koolagudi
Chairman DPGC
Department of CSE,
NITK Surathkal

ACKNOWLEDGEMENT

Place: Surathkal Awanit Ranjan,Pintu, Vivek Kumar
Date: 04-03-2022

Abstract

The popularity of social networks provides people with many conveniences, but their rapid climb has also attracted many attackers. In recent years, the malicious behaviour of social network spammers has seriously threatened the data security of ordinary users. To scale back this threat, many researchers have mined the behaviour characteristics of spammers and have obtained good results by applying machine learning algorithms to spot spammers in social networks. However, most of those studies overlook class imbalance situations that exist in-universe data. Usage of the internet and social media backgrounds tends within the use of sending, receiving and posting of negative, harmful, false or mean content about another individual, which ends up in a bad social media experience. It has led to a severe increase in mental state problems, especially among the young generation. It's resulted in lower self-esteem, increased suicidal ideation. Unless some measure against cyberbullying is taken, self-esteem and psychological state issues will affect a whole generation of young adults. Many of the standard machine learning models are implemented in the past for the automated detection of inappropriate content on social media. But these models haven't considered all the required features which will be wont to identify or classify an announcement or post as bullying. In this paper, we proposed AI-based model supported various features that ought to be considered while detecting spam and inappropriate content on social media.

Keywords: *AI powered, Inappropriate Content, Knowledge Base, Spam Detection*

CONTENTS

CHAPTER 1

Introduction

Millions of young people spend their time on social networking, and the sharing of information is online. Spammers appropriate bogus publicizing, erotic entertainment, phishing, and other vindictive data by means of informal communities. These malevolent practices bring about protection exposures, obliterate ordinary organization request, compromise informal community notoriety frameworks, and increment network loads, which make huge harm to innocent people. A correlation with conventional spam spread by email shows that informal community spam is more tricky, more hard to recognize, and represents a more noteworthy danger to common clients.

Besides with the context of spamming there is another threat namely Inappropriate Content spreading across multi platforms associated with growing trend of engagement in social media platforms. Cyberbullying is among the most generally recognized issues by people and networks. Due to enormous number of web-based conversation, it has been not unexpectedly seen that people discussions in some cases rapidly crash and become improper like reviling, passing impolite hateful or toxic, offensive or abusive comments and inconsiderate remarks on people. Therefore, unseemly messages or remarks are transforming into an internet based threat gradually corrupting the viability of client encounters. Consequently, programmed detection and separating of such improper language has turned into a significant issue for improving the quality of conversations with users as well as virtual agents [19].

1.1 Problem Description

Our work primarily focuses on building an NLP based Model for spam recognition and on the fly inappropriate content detection for social media. Thus, Given a tweet/message say t, we have to classify on the fly (in real-time) whether it is a spam or ham if the system with which we are dealing with is with the context of spamminess or on the other hand, we have to identify/detect whether that tweet is indeed appropriate/legitimate or Inappropriate/not legitimate for to be posted on social media platforms . This appropriateness can be with context to whether an incoming tweet is toxic, hateful, offensive , abusive, or promotes cyber-bullying.

1.2 Motivation

In the last five years an extensive research has been done by the researchers on examining the social and computational aspects of spam and inappropriate content detection. The awareness of social media and harms by online abuse are proportionally increasing .[9]. There exist many ML models for detecting spam and inappropriate contents i.e. toxic and abusive comments on different social platforms but they suffer from one or the other major drawbacks. They are trained using a specific labelled dataset which gives high training accuracy but fails to give outstanding results on the live contents. We build a new advance ML model powered with AI which detects the spam and inappropriate contents on the fly and AI embedded system learns from the user experience and feedback. Our system will retrain itself on constantly growing dataset over a specific period of time which makes it autodidacticism, unique and ineffable.

1.3 Objectives

There is mainly two objective:

- **To make this process of prediction fast and very effective in real-time.** Often, the ML model fails to deliver the desired results due to an outdated or imbalanced dataset on which the model was trained previously. Our model overcomes this problem by continuously growing the respective datasets based on users' feedback.

- **To make this tool self reliable and achieve high accuracy.** The embedded AI learner will keep on expanding its knowledge base which will be merged into the dataset once a threshold value is reached. Along with our aim is to fine tuned ML model for achieving high accuracy.

1.4 Organization of the report

In chapter-2, literature survey has been discussed for spam detection and Inappropriate content on the fly, followed by prolem statement and objectives of our work. In chapter-3, our approach is being proposed. Machine learning module and AI module are discussed. In chapter-4, experimental design and analysis is being mentioned. This includes system specification, dataset description, dataset preprocessing followed by ML models and results. In chapter-5, we conclude our work.

CHAPTER 2

Literature Survey

This section reviews the related works from the following two aspects: Spam detection approaches and second is Inappropriate Content On the fly.

2.1 Spam Detection Approaches

As of now, spam discovery is one of the main difficulties for online social network security. Supervised Machine Learning are the most widely recognized techniques utilized for spam detection. Chensu Zhao et al.[13] forces discussion on class imbalance problems for detecting spam. They had used a Heterogeneous stacking based ensemble learning framework to mitigate class imbalance on spam detection in social networks. The proposed framework consist of 2 main module one is the base module and other is the combine module. In base module they had adopted 6 different base classifiers which are SVM, Cart, GNB(Gaussian Naive Bayes), KNN, RF, LR and uses their diversity to construct new ensemble input members. On the other hand in combining module for the meta classifier they had introduced a cost sensitive learning for class imbalance problem into deep learning Neural Network training. The authors had trained their models on publicly available 6 Million spam tweets, a large GT for Timely Twitter Spam Detection. Their model produces a F1 score of 0.7 which indeed outperforms some traditional ML Supervised Learning Models.

Sahar Bosaeed et al. [10] proposes a tool for to detect spam from outgoing messages. They have built a ML based classifier NBM, SVM, NB(Naive Bayes). Their model is evaluated using 15 datasets build by inter-mixing of 5 basic datasets which are British Dataset, UCI,SMS Spam Corpous Big, SMS Spam Datasets, User Data of authors.They have applied feature based engineering for feeding input to the model. They have used accuracy for evaluation metric with an accuracy of 98.8%, also among the classifiers SVM has performed the best. Pumrapee Poomka et al. [20] proposes SMS spam detection by preparing data through word tokenization, padding data. truncating data Glove word embedding then feeding into model based on LSTM and GRU. According to authors their models outperforms SVM, NB models with giving an accuracy of 98.18% and with an error rate of 0.74% only. They had used SMS spam dataset by [6] having 5574 records with 747 as spam and 4827 as ham.

Himank Gupta et al. [5] develop a tool to detect real time spam detection on twitter. It is based on user and tweet based along with tweet text features approach to classify the tweets. It uses 4 ML CLassifer for their work which are SVM, NN, RF, Gradient Boosting. They had featured their task into various features based approach with an average accuracy of 85.5%. Chao Chen et al. [11] created a GT for spam tweets detection.They had extracted 12 lightweight features for streaming tweet spam detection and found the features discretization is important for spam detection performance. They had investigated 6 ML models to build up tweet spam detection and reports its behaviours under different experiment settings.

Rodica Potolea et al. [14] are filtering spam comments on various features such as discontinous text (incoherant comment), inadequate text, vulgar language and coherant comments such as comments not related to specific context of posting. Their architecture consist of three models. 1. Feature extraction Module, 2. Post Comment Similarity Module, 3. Topic Extraction Module and the classifiers are Naive Bayes, SVM and Decision Tree. They had used the dataset [15] comprising of 1024 comments of spam and ham. Anupama Aggarwal et al. [8] propses a tool named Phisari and browser extension for detecting spam on twitter in realtime. They had used Twitter features like tweet contents , no: of hastags, lentgh of content etc and URL based features. They had detect phising with an accuracy of 95.52%. For training the model they have collected the tweet from Twitter having URL in it and labelling it as safe or phishing. The features on which they had focused are URL based features, WHOis based features, Tweet based features and Network based features. The limitation of this tool is that it detect only phishing URL and not like spam or inappropriate tweets.

2.2 Inappropriate Content On the Fly Detection Approaches

In various social media platforms, it has been often observed that user conversations often derail and become inappropriate such as hurling abuses, passing rude and discourteous comments on individuals or certain groups/communities.

Harish Yenala et al. [17] researches on abusive behaviors on online platforms. They have surveyed existing methods and content moderation policies of online platforms for detecting abusive content on online platforms. They have characterized this as buckets of Toxicity, Profanity, Cyber-Bullying, Abusive Language, Aggressive. They also concentrated their attention in identifying inappropriate content which is target aware and target agnostic. They had used pretrained BERT, CNN, RNN, GRU and SVM as a part of ensemble. Gonzalo et al.[16] address the problem of detecting erotic/ sexual content on text documents using NLP. They have done this analysis on 12 models, three of them are text encoders BOW, TF-IDF, W2Vec and 4 are classifiers SVM, LR, KNN and Random Forest. Dataset is extracted from Reddit website. The best model is a combination of SVM with linear kernel with TF-IDF as text encoder. This is giving an accuracy of 97% and F1 score of 0.96.

Semiu Salawu et al. [24] has done extensive literature and classified its review into 4 main classes which are Supervised Learning Lexicon Based, Rule Based and mixed initiative approach. They have used SVM and Naive Bayes, also they used a rule-based approach and match text to pre-defines rules to identify bullying. They had collected datasets from youtube, myspace, twitter and from mails. Sandip Modha et al. [23] present an approach to detect and visualise online aggression a special case of hate speech over social media platforms. They have covered three types of aggression which are: Overtly aggressive (OAG), Covertly aggressive (CAG),Non-aggressive (NAG). They have used SVM,DL based on CNN, Attension based model,BERT models. The authors have used standard database available for inappropriate content like TRAC for aggression detection, SemEval for offensive content in english, HASOC hate and offensiveness.

Rajesh et al.[21] developed an Automater called as Blockshame for public shamming detection in twitter. Shamming tweets are categorised into 6 types and they are: abusive,comparison, passing judgement, religions/ethinic , sarcasm/joke, whatboutery. Each tweet is classified into one of these types or as non-shamming.

They have used 6 SVM classifiers. Monirah et al.[18] have used DL approaches for cyberbullying detection. They have proposed a novel algorithms CNN-CB that eliminate the need for feature engineering and gives accuracy of 95%. CNN-CB consists of 4 layers they are: embedding, convolution , pooling and dense. Harish Yenala et al. [17] has propses a DL based approach for detcting inappropriate content in text. They have solve this issue in two scenarios. First is the Query completion suggestions in search engine and second is the user conservation in messangers. For the first scenario i.e query suggestion they have used CNN and Bi-LSTM and for conversation messages they have used LSTM and Bi-LSTM. .They also has evaluated various techniques proposed so fat for query suggestions regardingincluding standard deep learning techniques sych as CNN, LSTM, Bi-LSTM on real world dataset. They also had evaluated performance of sequential models like LSTM+BiLSTM for identifying inappropriate content in conversation. Anitigoni et al. [7] has presented a detailed view of Large scale crowd sourcing and characterization of twitter abusive behaviours. They had picked topics of interest such as spam , hateful speech, offensive language, aggression, cyberbullying, abusive i.e in short Inappropriate contents, and Normal texts.

2.3 Problem Statement and Objectives

Our work primarily focuses on building an NLP-based Model for spam recognition and on-the-fly inappropriate content detection for social media. Thus, Given a tweet/message say t, we have to classify on the fly (in real-time) whether it is a spam or ham if the system with which we are dealing with is with the context of spamminess or on the other hand, we have to identify/detect whether that tweet is indeed appropriate/legitimate or Inappropriate/not legitimate for to be posted on social media platforms. This appropriateness can be with context to whether an incoming tweet is toxic, hateful, offensive, abusive, or promotes cyber-bullying.

2.4 Summary

Plenty of work has been done in spam classification and filtering out another type of toxicity on the internet-based social interactions and personalized message services. The researchers have used datasets from all possible sources like the Reddit website, mobile SMS, emails, and standard datasets prepared by others. Some have used massive datasets containing millions of data points; others have used just a few thousand texts. From simple ML models to advance Neural Network models have been used

CHAPTER 3

Proposed Approach

NLP is a broad and multidisciplinary discipline that deals with the automatic translation of human languages. NLP is a candidate for every practical application that uses text, and breakthroughs in the machine learning discipline largely drive its success.

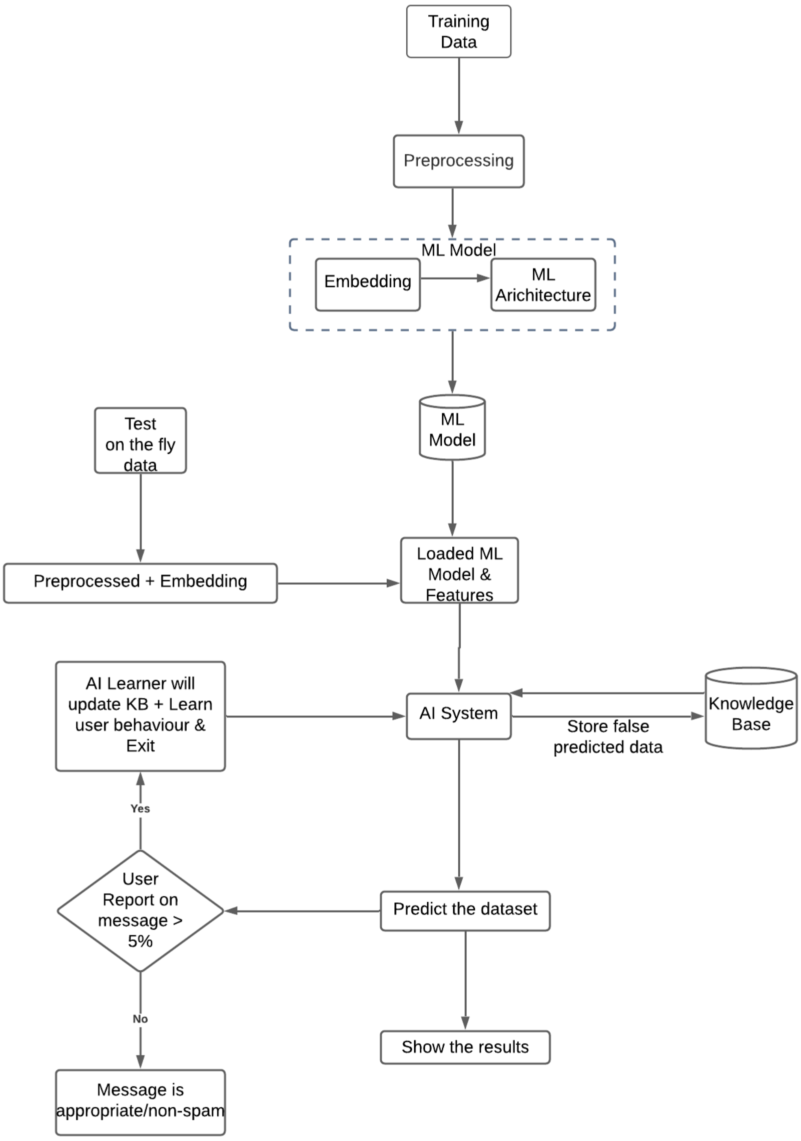
Initially, we have decided to set up some classification experiments with traditional machine learning classifiers from literature like Support Vector Machines (SVMs), Naive Bayes, etc., Logistic Regression. Some pre-proposed DL-based approaches to get a glimpse of the work. Also it helps in comparing the models w.r.t time, precision, recall, accuracy , F1 score and other performance metrics with our proposed architecture.

Now moving to our proposed architecture, To support our objective, we propose a new end-to-end ML architecture backed by AI(Artificial Intelligence). Our Deep Learning-based ML architecture is robust, unique, and free from tedious and biased feature engineering processes; on the other hand, the AI system provides autodidacticism to the machine learning architecture.

The above figure (figure 1.0) presents a holistic view of our complete architecture for detecting spam and Inappropriate Content on the fly. Before presenting a description about the complete architecture let first view its heart and brain modules i.e we have proposed our architecture in two sub-modules.

- A). **Machine Learning Module**
- B). **Artificial Intelligence Module**

Figure 3.1: Complete Proposed Architecture for Spam and Inappropriate Content detection On the fly.



3.1 Module A - Machine Learning Module

This subsection describes the proposed heterogeneous stacking-based Ensemble Learning Framework for Spam and Inappropriate Content Detection in Social Media Platforms. The existing empirical results have shown that ensemble-based learning models tend to perform better when there are significant differences among the ensemble models. The stacked model of several learning stages is the most popular ensemble learning approach. Thus, to solve the stated problem statement, we propose a novel framework with a two-level structure, i.e., a base module and a combining module.

We have proposed three different Heterogeneous Stacking-based Ensemble Learning Framework for the base module, focusing on implementation. The model that performed best will be selected.

In the First ML Framework (Fig 2.0), at the first level of ML architecture, we will be using two popular word embeddings, Glove and FastText, and three popular RNN based ML Algorithms Bi-LSTM, Bi-GRU, and Encoder-Decoder Transformer BERT, followed by Convolution layer (CNN) and Average and Max Pooling. The input, i.e., real-time tweets, will be converted into a vector by use of stated word embeddings which will go through a dropout layer for regularization into Three stated ML algorithms as shown in figure 2, the outputs from these models will be convolved and pooled by using Average and Max Pooling, and at last, all the results from pooling layer will be concatenated. This concatenated output will enter into the second level, which is our meta-classifier of ensemble network i.e. Dense layer (Fully Connected Neural Network) and at last output layer having sigmoid as an activation function for predicting 1 and 0 for spam and ham or Inappropriate and Appropriate respectively.

In the Second ML Framework (Fig 3.0), at the first level of this ML Framework again we will be using two already stated word embeddings, which are Glove and FastText, and three popular RNN based ML Algorithms Bi-LSTM, Bi-GRU, and BERT, followed by optional Convolution layer (CNN) and Pooling which we will decide by testing its performance. The input, i.e., real-time tweets, will be converted into vectors using stated word embeddings which will go through a dropout layer for regularization into Three stated ML algorithms, as shown in Figure 3. This is different from Figure 2 in the way of ensembling the outputs of two-word embeddings, i.e., instead of converging into three different models, we will rather converge into the same model; thus, the first framework takes the help of all three models from single embeddings while the second takes the help of two embeddings for producing output before concatenation.

After that, outputs are concatenated from all three models. This concatenated output will enter into the second level, which, again similar to the first framework, is our meta-classifier of ensemble network, i.e., Dense layer (Fully Connected Neural Network), and at last output, the layer having sigmoid as an activation function for predicting 1 and 0 for spam and ham or Inappropriate and appropriate respectively.

In the Third ML Framework (Fig 4.0), we can see that it is almost similar to the Second framework i.e., the first level is the same for both of the frameworks. The difference lies in the second level, which is the meta classifier. In this framework, we do not concatenate the outputs from 3 models. Instead, we pass the individual outcomes coming out from individual models through a dense layer. The output of the dense layer is given to the output box, which is a simple neuron having a sigmoid as an activation function. Outputs connect this simple neuron from the dense layer of three models by a certain weight. We will be treated as a hyperparameter whose values we will figure out by fine-tuning. This three summarized our proposed Machine Learning architecture which is the heart of the entire proposed architecture.

3.2 Module B - Artificial Intelligence Module

This module serves as the brain of the Complete proposed ML architecture because this module is responsible for making the complete system autodidactic. Here we have proposed to use the light-weighted model such as Naive Bayes and KNN for predicting the output. Our AI system will maintain an internal Knowledge Base (KB in short). Whenever the user feels the ML Model's expected outcome is incorrect, they can flag that content as inappropriate or legitimate or mark it as spam or ham, depending on the user's choice. This flagging event will trigger a request to AI learners sitting inside the AI system to update its Knowledge base with this new information. In case of tweet's legitimacy, the learner also stores the content with some extra information w.r.t to the context of tweet among Hate, Toxic, Offensive, Abusive, Cyber-Bullying or Normal . The user has a choice to mark the tweet even as multiple options among the categories of inappropriateness, i.e. Hate, Toxic, Offensive, Abusive, or Cyber-Bullying.

Besides, the AI Learner will use the KNN and Naive Bayes to validate the prediction from the model. Thus, a weighted score will be calculated if there is a conflict of prediction from our AI Learner and ML model. These weights will

Figure 3.2: The First ML architecture proposed Framework

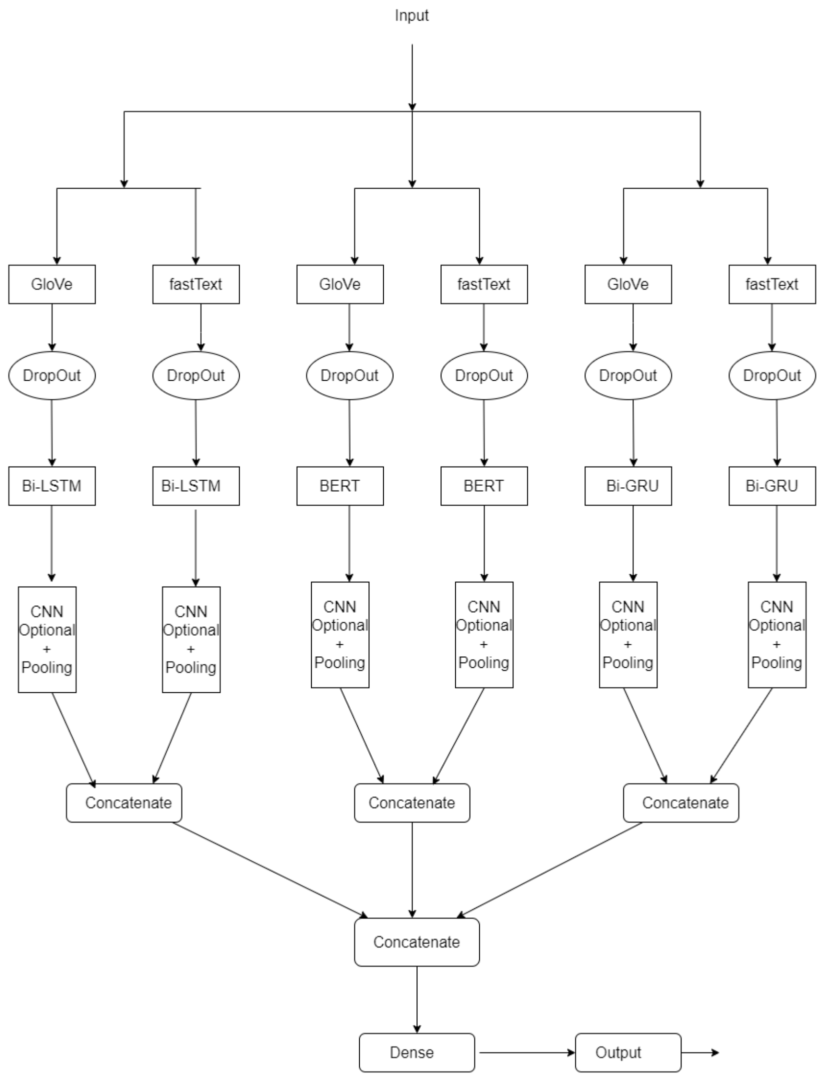
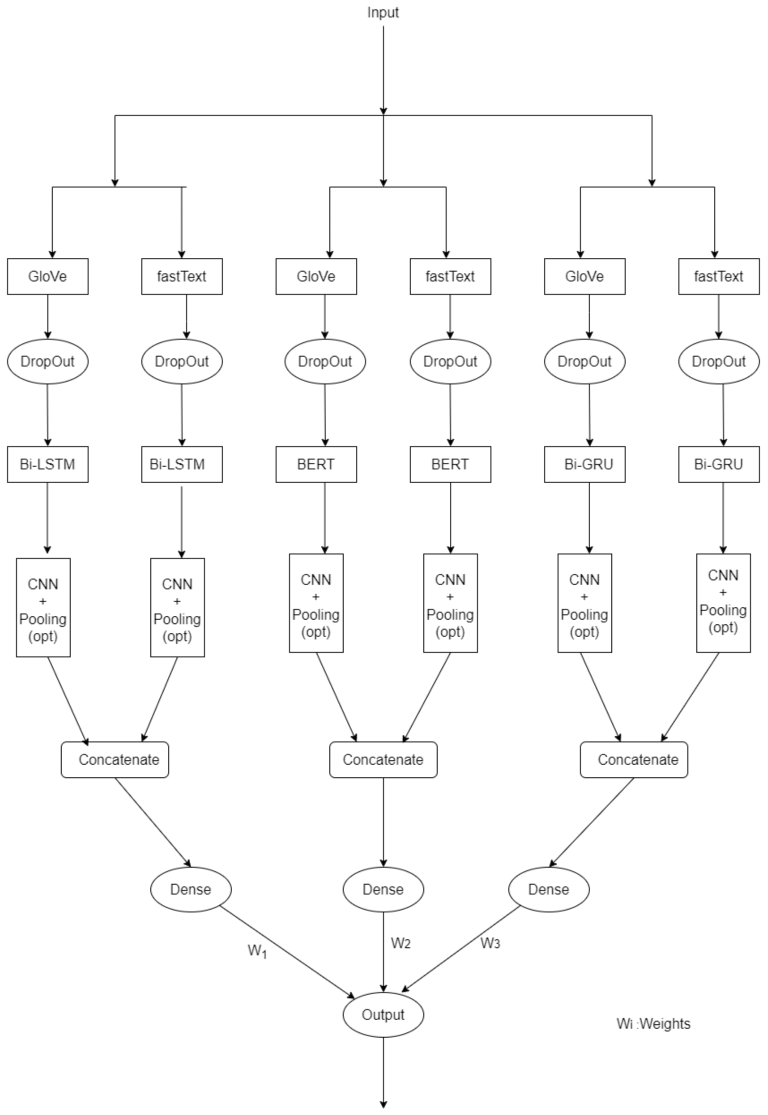
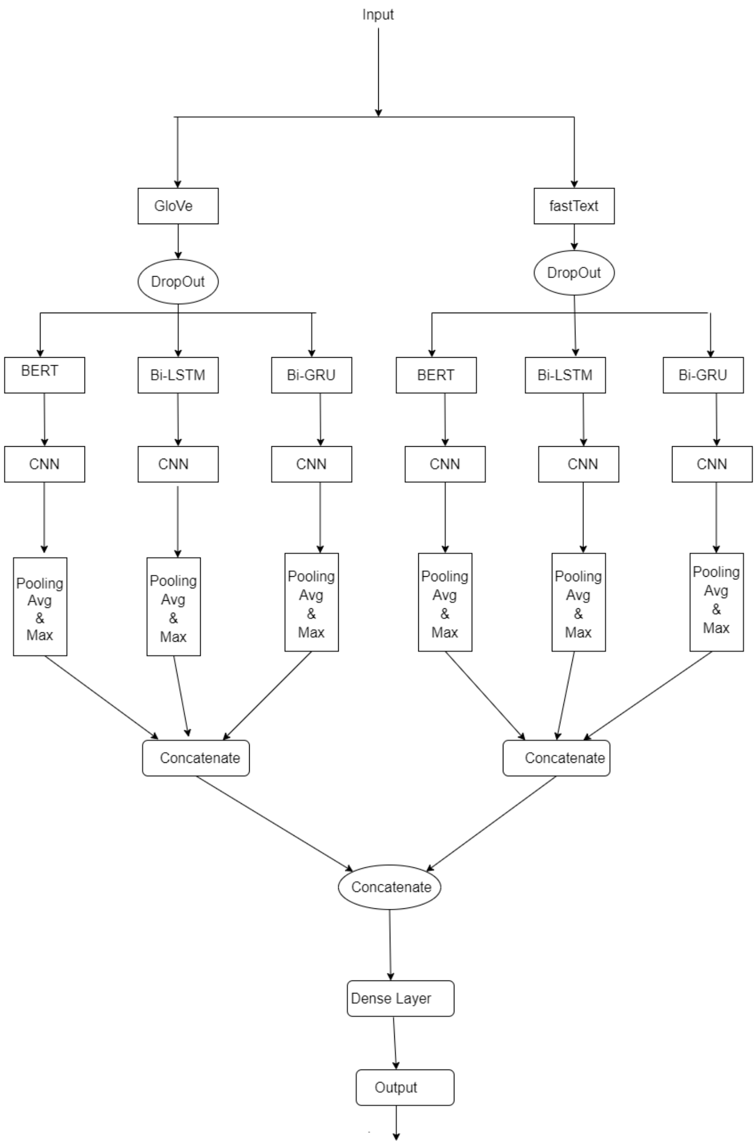


Figure 3.3: The Second ML architecture proposed Framework



keep on changing depending on user feedback. Thus, this flexibility and lightweightedness allow the model to perform and validate the predictions in real-time. Besides, our system can adjust to the population's sentiment by auto-learning. One more feature this AI system possesses is that upon filling KB to a specific limit, its data will be pushed to the respected training set. Our ML model will re-train on this new pumped training data set. This procedure will update the weights of the model parameters and allow the system for a short of feedback for continuous learning. The 5% percent mentioned in the diamond box in figure 1.0 states that once the 5% of the people seen that tweet will report/flag that tweet, then only that flagged tweet will become a candidate for pumping into the training set. Also, our KNN or Naive Bayes will use this KB (Knowledge Base) for

Figure 3.4: The Third ML architecture proposed Framework



validating the predictions from Module A, i.e., ML Model.

After growing through the heart and brain of the complete system, it gets easy to explain the whole architecture. First, we will prepare the ML Model with the help of a training dataset. The trained model will be loaded in the memory to do On the Fly predictions. Once any tweet pops in real-time, it will be converted into the vector and fed into the trained ML Model for prediction; also, that tweet in the form of a generated vector will be passed in parallel to the AI Learner. After receiving the predictions from ML Model and AI Learner, the AI system will check for the possibility of any conflict between the predicted values. In case of no conflict, the necessary actions will be taken depending on the output. In any case of conflict, the predicted output will be the weighted average of both the predictions, which will be adjusted dynamically. Besides, necessary actions will be triggered if the user flags or reports sudden tweets, as explained in section 5.2 (Module B- Artificial Intelligence Module).

3.3 Summary

The final model will be comprised of a heterogeneous ensemble stack of ML models. AI will be embedded into it to make the model self-dependent for future upgradation and improve classification accuracy. Different types of word embeddings have been used at different stages in the different model architecture. Advance Neural Networks i.e. BERT, Bi-LSTM and Bi-GRU are used for yielding high accuracy.

CHAPTER 4

Experimental Design and Analysis

4.1 System Specification

We have wrote, trained and tested our model on 2 platform which are follwoing :

- **Google Coolab Platform** : Colab, or 'Colaboratory', it allows user to write and execute Python in the browser, with

  - 1. Zero configuration required,
  - 2. Free access to GPUs and
  - 3. Easy sharing

  It's specifications includes :

  - Disk : 107.72 GB
  - RAM : 12.69 GB
  - CPU : Intel(R) Xeon(R) CPU @ 2.20GHz (1 core, 2 threads)
  - GPU : 1xTesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM

- **Kaggle Platform** : Kaggle Notebooks run in a remote computational environment. They provide the hardware which user need so user only need to worry about the code. At time of writing the code and testing, each Notebook editing session is provided with the following resources:

  - 12 hours execution time for CPU and GPU notebook sessions and 9 hours for TPU notebook sessions
  - 20 Gigabytes of auto-saved disk space (/kaggle/working)
  - Additional scratchpad disk space (outside /kaggle/working) that will not be saved outside of the current session
  - CPU Specifications : 4 CPU cores with 16 Gigabytes of RAM

– GPU Specifications : 2 CPU cores with 13 Gigabytes of RAM

– TPU Specifications : 4 CPU cores with 16 Gigabytes of RAM

4.2 Dataset Description

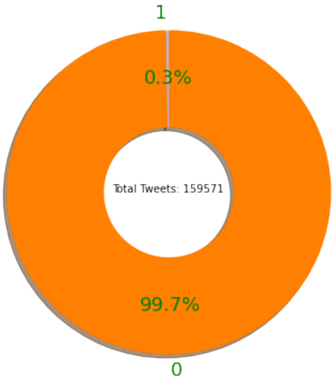
The second step in our work after literature survey was to collect the datasets for both Spam Detection and Inappropriate Content Detection. Thus for our work we have collected following dataset :

4.2.1 Inappropriate Content Detection

For the Inappropriate Content Classification we are using dataset : jigsaw-toxic-comment-classification-challenge Datset hosted on Kaggle platform from Toxic Comment Classification Challenge, This dataset is a multi-headed in nature containing **6 different types of of inappropriateness** which are following with its respective sitribution within the training dataset :

- **Threat** : Threat is defined as a statement of an intention to inflict pain, injury, damage, or other hostile action" against a target [2]. This includes death threats, threats of physical violence, and, for women, often threats of sexual violence.

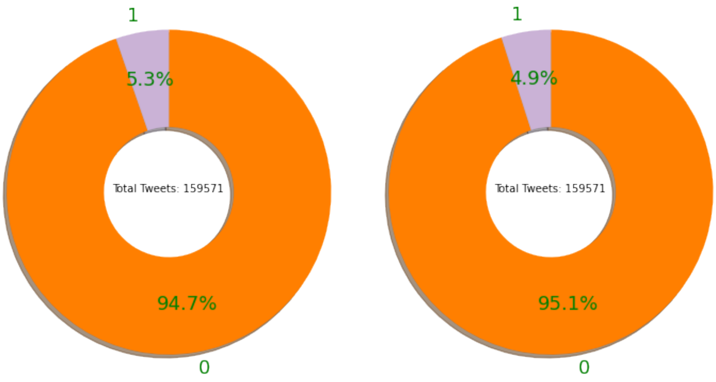
Figure 4.1: Distribution of threat comments



- **Obscenity** Obscenity refers to a narrow category of pornography that violates contemporary community standards and has no serious literary, artistic, political or scientific value. [4]

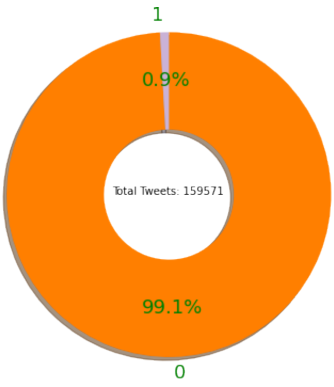
- **Insults** : An insult is an expression or statement (or sometimes behavior) which is disrespectful or scornful. Insults may be intentional or accidental. An insult may be factual, but at the same time pejorative, such as the word "inbred". [1]

Figure 4.2: Distribution of Obscene and Insults comments espectively from left



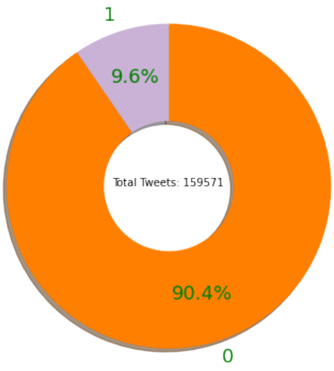
- **Identity-Hate**: Online hate or identity - hate can be defined as any hateful posts about a person or group based on their race, religion, ethnicity, sexual orientation, disability or gender. [3]

Figure 4.3: Distribution of Identity-Hate comments



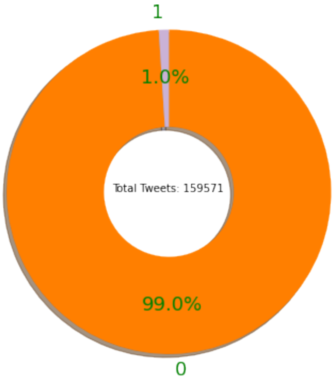
- **Toxic** : A toxic comment is defined as a rude, dis- respectful, or unreasonable comment that is likely to make other users leave a discussion. [22]

Figure 4.4: Distribution of Toxic comments



- **Severe-Toxic/Aggressive**: Severe-toxic or Aggression is defined as any act of aggression, or a behavior intended to harm another person who does not wish to be harmed. [12]

Figure 4.5: Distribution of threaSevere-Toxic/Aggressive comments



Though there are many other datasets for Inappropriate Content but we are going with this dataset because it is nicely structured, high availability and the most important it provides various classes of Inappropriateness thus not restricting the model to yield a just a binary result of appropriate or Inappropriate. Also, since we have noticed the skew-ness of dataset so, we will be taking F1-Score too for performance evaluation along with accuracy.

4.2.2 Dataset For Spam Detection

For the Spam Detection we are using 2 datasets i.e combining of emails and text spam :

- **Spam or Not Spam Dataset** : A collection of emails taken from spamassassin.apache.org i.e. Apache SpamAssassin's public datasets. There are 2500 ham and 500 spam emails in the dataset. In this dataset all the numbers and URLs were converted to strings as NUMBER and URL respectively. This is the simplified spam and ham dataset.This dataset is also available on kaggle.

- **Spam Text Message Classification** : A structured data of SMS messages in CSV format which has 2 variables- category and message available on kaggle.

4.3 Dataset Preprocessing

A series of steps are performed before model is feeded into the ML Model and which are following:

Remove punctuation: Tweets contains many punctuation i.e. comma, apostrophe, quotation, question, exclamation, brackets, braces, parenthesis, dash, hyphen, ellipsis, colon and semicolon. In this step, we remove all these characters from each tweet.

To Lower: Text follows certain language specific syntax. One of the mostly used norm is to keep first letter of the sentence capital. The glove file contains all the words in lower format. This step converts each and every alphabetical character into lower case.

Tokenization: Still each tweet is single string. Stopwords can't be removed from this string. In this step, we divide this entire text string into words. Each word will be a separate string now.

Remove stopwords: English language contains many extra words which help in framing the sentences. These words don't have significant meaning when standalone. These are called stopwords i.e. ”a”, “the”, “is”, “are” and etc. In this step, we remove all the stopwords present in each tweet.

Stemming: This step is very crucial to retrive the dense vector from the glove file. Glove file contains the root words only. Stemming reduces the inflected words to their root words. For example: swimming is reduced to swim.

Lemmatizing: Lemma simply means the root words, but unlike stemming it gives the root word based on the context of the sentence. ”good” is the lemma of ”better”, stemming misses this point. On many words, lemma and stem might give same result. For example, the word ”walking” has lemma and stem word ”walk”. Lemmatisation attempts to select the correct lemma depending on the context, unlike stemming.

4.4 Machine Learning Models & Results

4.4.1 ML Models Used

We have tried different ML models and DL models for achieving our goal of detecting spam and inappropriate content before starting to work on our proposed model.

- **Gaussian NB:** Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. The model has given accuracy of 35 percent and F1 score of 0.51.

- **KNN:** A k-nearest-neighbor algorithm is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it. The model has given accuracy of 75 percent and F1 score of 0.42.

- **Decision Tree Classifier:** Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.The model has given accuracy of 81.6 percent and F1 score of 0.67.
- **SGD Classifier:** Stochastic Gradient Descent is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. The model has given accuracy of 88 percent and F1 score of 0.82.
- **Logistic Regression:** Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.The model has given accuracy of 88 percent and F1 score of 0.82.
- **RandomForestClassifier:** A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The model has given accuracy of 88.6 percent and F1 score of 0.83.
- **GradientBoostingClassifier:** Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. The model has given accuracy of 89.2 percent and F1 score of 0.80.
- **SVC:** Support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. The model has given accuracy of 89.8 percent and F1 score of 0.83.

4.4.2 Framework & Libraries Used

In order to bulid the above stated machine learning models we have used following libraries/frameworks :

- **Tensorflow** as backend for DL Method while for traditional ML Models we have used.

- **sklearn** : This for Model implementation, feature extraction, splitting the dataset, calculating precision,recall, accuracy and generating classification summary.
- **numpy** : This is for matrix/array operations
- **pandas** : This is for dataframe management and operations related to it.
- **matplotli**b : This is for plotting of graph.

4.4.3 Results

Results for Spam Detection

For Spam detection tasks the above models yields following results :

Result of Stated Model for Spam Detection in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	GaussianNB	0.34	1.00	0.51	35%
2	KNN	0.96	0.27	0.42	75%
3	Decision Tree Classifier	0.68	0.66	0.67	81.6%
4	SGD Classifier	0.82	0.83	0.82	88%
5	Logistic Regression	0.84	0.80	0.82	88%
6	RandomForestClassifier	0.82	0.83	0.83	88.6%
7	GradientBoostingClassifier	0.85	0.76	0.80	89.2%
8	SVC	0.86	0.80	0.83	89.8%

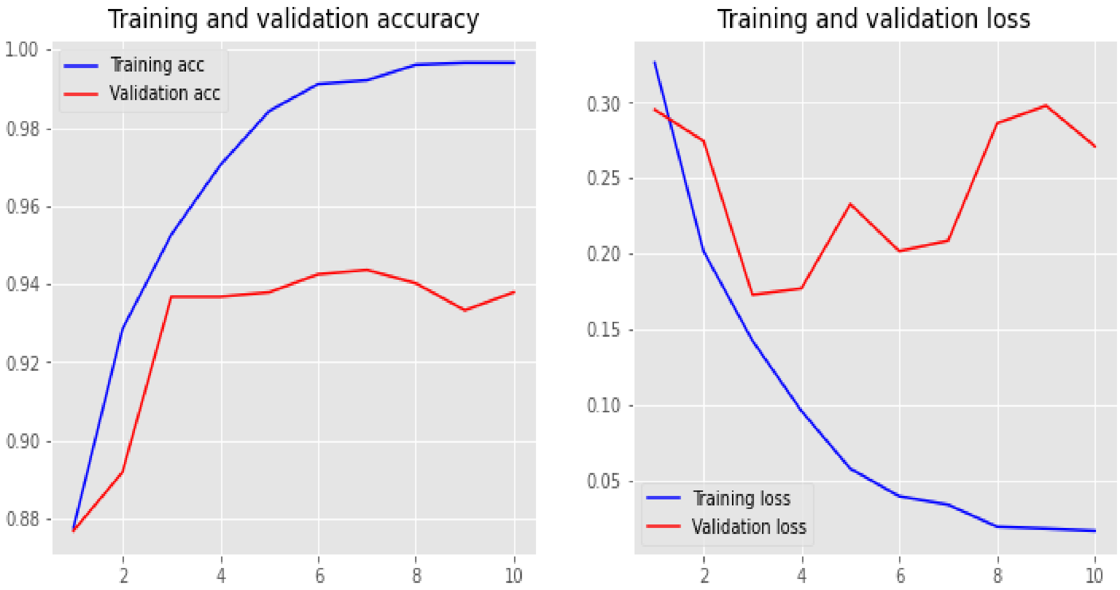
Among ML(Non-DL) approaches the model which perform the best was SVC with an accuracy of 89.8% and F1 Score of 0.83. On the other hand the maximum accuracy achieved by DL model after performing parameter tuning is 94%. The corresponding graphs of accuracy and loss of training and validation is below.

Results for Inappropriate Detection

For Inappropriate detection tasks we have applied ML approaches(Non-DL) for each classes of Inappropriateness.

- For Toxic (Inappropriate) Content results are following:

Figure 4.6: Graph depicting Loss/Accuracy for Training & validation of DL model for Spam Detection .



Result of Stated Model for Toxic(9.6%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	KNeighborsClassifier	0.83	0.30	0.44	92.8%
2	DecisionTreeClassifier	0.66	0.67	0.67	93.7%
3	GradientBoostingClassifier	0.92	0.39	0.55	93.9%
4	SVC	0.94	0.41	0.58	94.2%
5	MultinomialNB	0.81	0.60	0.69	94.9%
6	RandomForestClassifier	0.86	0.59	0.70	95.3%
7	SGDClassifier	0.87	0.62	0.73	95.6%

In Classification Algorithms summary we can see the for Toxic Class the best model having highest accuracy & F1 Score is SGD Classifier with an accuracy and F1 Score of 95.7% and 73.2% respectively. The model performing best w.r.t precision is SVC with precision of 94.7%. The best recall Model is DT with an recall of 67.4%.

In case of time complexity analysis we have best training time for KNN with Training Time of 0.014 sec and Worst Training Time of 1626.626 sec for SVC. The best prediction is given by SGD model with prediction time of 0.018 sec while the worst is for SVC with prediction time of 835.36 sec.

Figure 4.7: Classification Summary for Toxic.

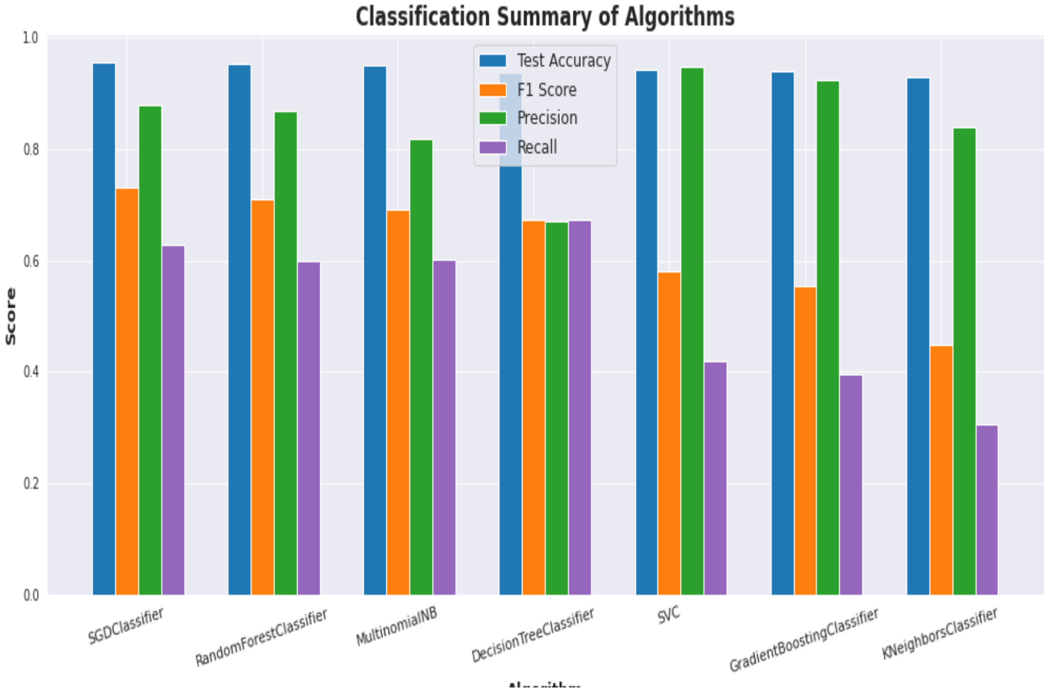
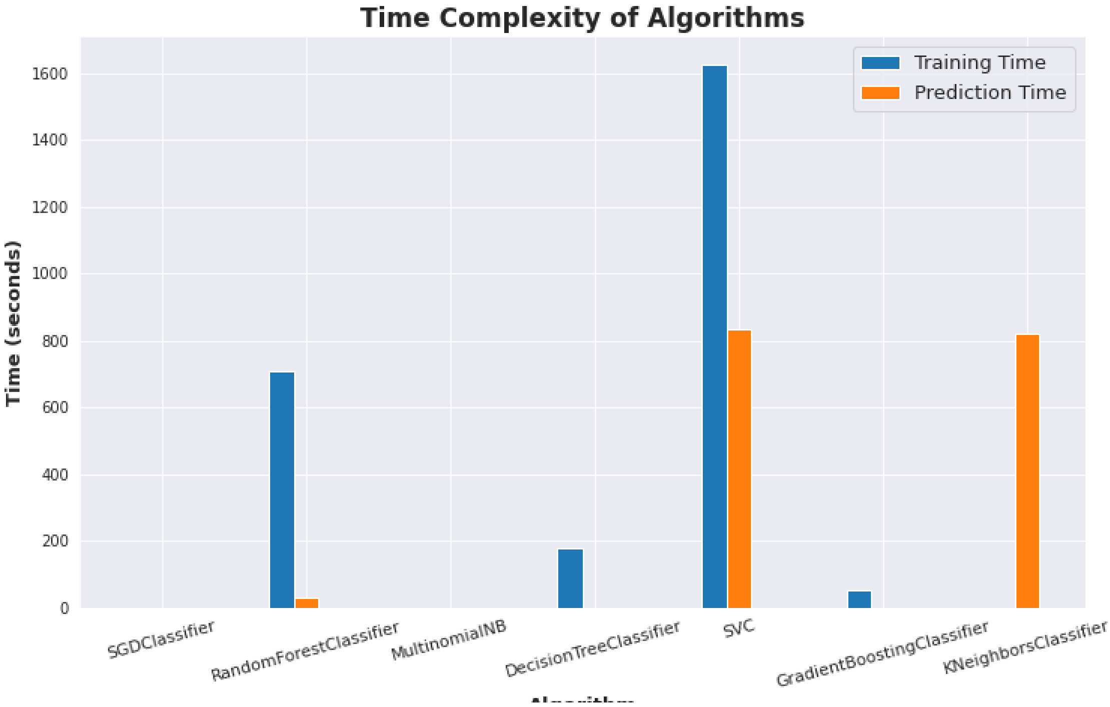


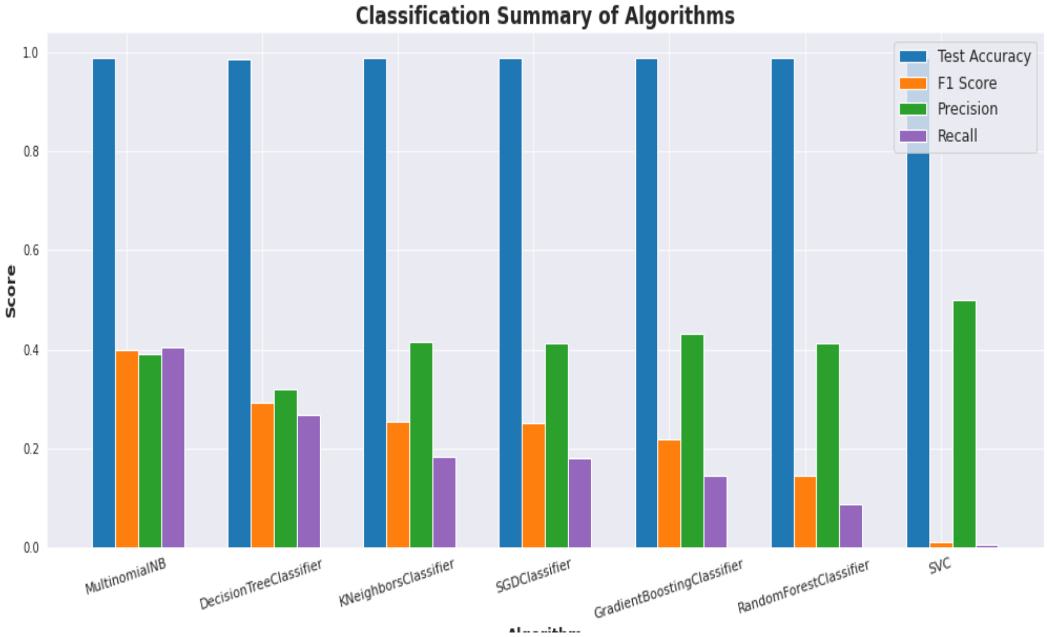
Figure 4.8: Bar Chart showing Time Evaluation of different algorithms for Toxic.



- For Severe-Toxic/Aggresive (Inappropriate) Content results are following:

Result of Stated Model for severe˙toxic(1%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	DecisionTreeClassifier	0.32	0.26	0.29	98.7%
2	MultinomialNB	0.39	0.40	0.39	98.77%
3	SGDClassifier	0.41	0.18	0.25	98.927%
4	KNeighborsClassifier	0.41	0.18	0.25	98.929%
5	GradientBoostingClassifier	0.43	0.14	0.21	98.95%
6	RandomForestClassifier	0.41	0.08	0.14	98.96%
7	SVC	0.50	0.005	0.009	99%

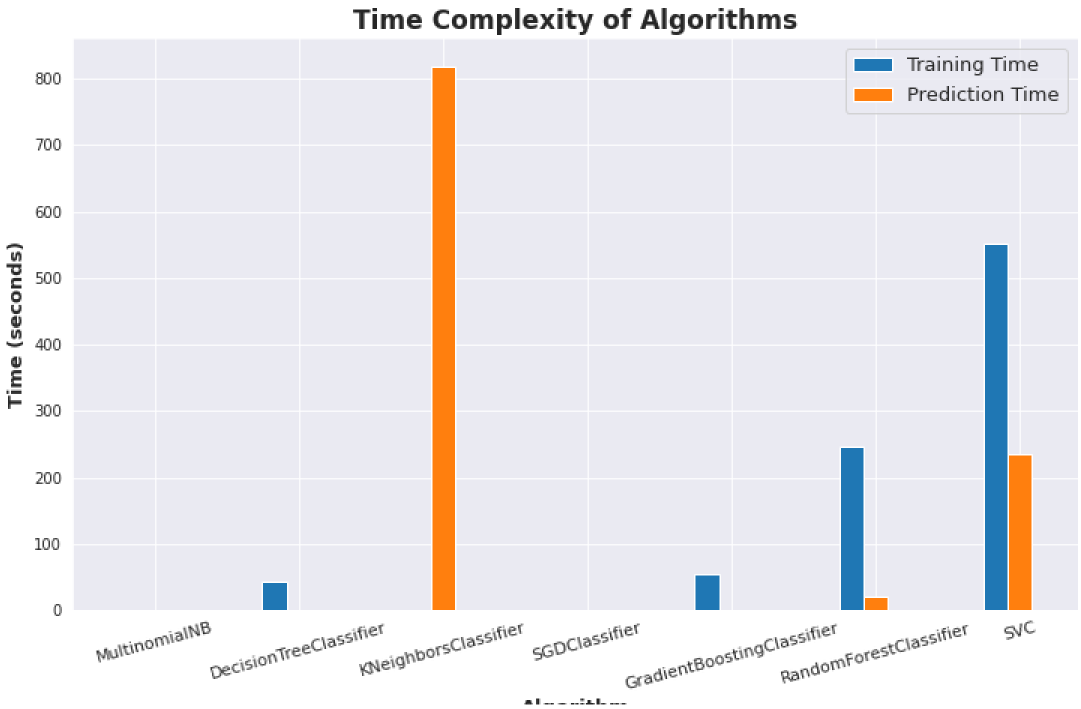
Figure 4.9: Classification Summary for Severe-Toxic.



In Classification Algorithms summary we can see the for Severe-Toxic Class the best model having highest accuracy & F1 Score is SVC with an accuracy of 99.0% and Multinomial NB with F1 Score of 39.8% respectively. The model performing best w.r.t precision is SVC with precision of 50.0%. The best recall Model is Multinomial NB with an recall of 40.6%. In case of

time complexity analysis we have best training time for KNN with Training Time of 0.013 sec and Worst Training Time of 552.606 sec for SVC. The best prediction is given by SGD model with prediction time of 0.019 sec while the worst is for KNN with prediction time of 819.139 sec.

Figure 4.10: Bar Chart showing Time Evaluation of different algorithms for Severe-Toxic.



- For Obscene (Inappropriate) Content results are following:

Result of Stated Model for Obscene(5.3%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	KNeighborsClassifier	0.90	0.38	0.53	96.5%
2	MultinomialNB	0.78	0.61	0.69	97.05%
3	GradientBoostingClassifier	0.89	0.54	0.67	97.20%
4	SVC	0.91	0.52	0.66	97.21%
5	DecisionTreeClassifier	0.75	0.75	0.75	97.39%
6	SGDClassifier	0.85	0.69	0.76	97.74%
7	RandomForestClassifier	0.87	0.69	0.77	97.88%

In Classification Algorithms summary we can see the for Obscene Class the best model having highest accuracy & F1 Score is Random Forest Classifier with an accuracy of 97.9% and F1 Score of 77.8% respectively. The model performing best w.r.t precision is SVC with precision of 91.7%. The best recall Model is DT with an recall of 75.8%.

Figure 4.11: Classification Summary for Obscene.

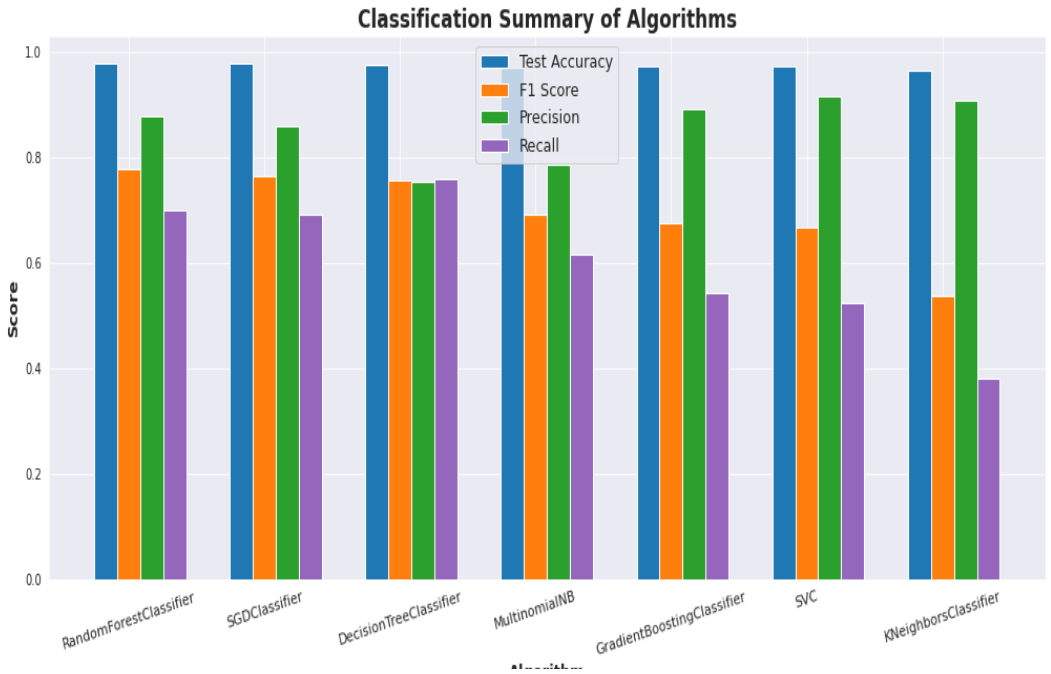
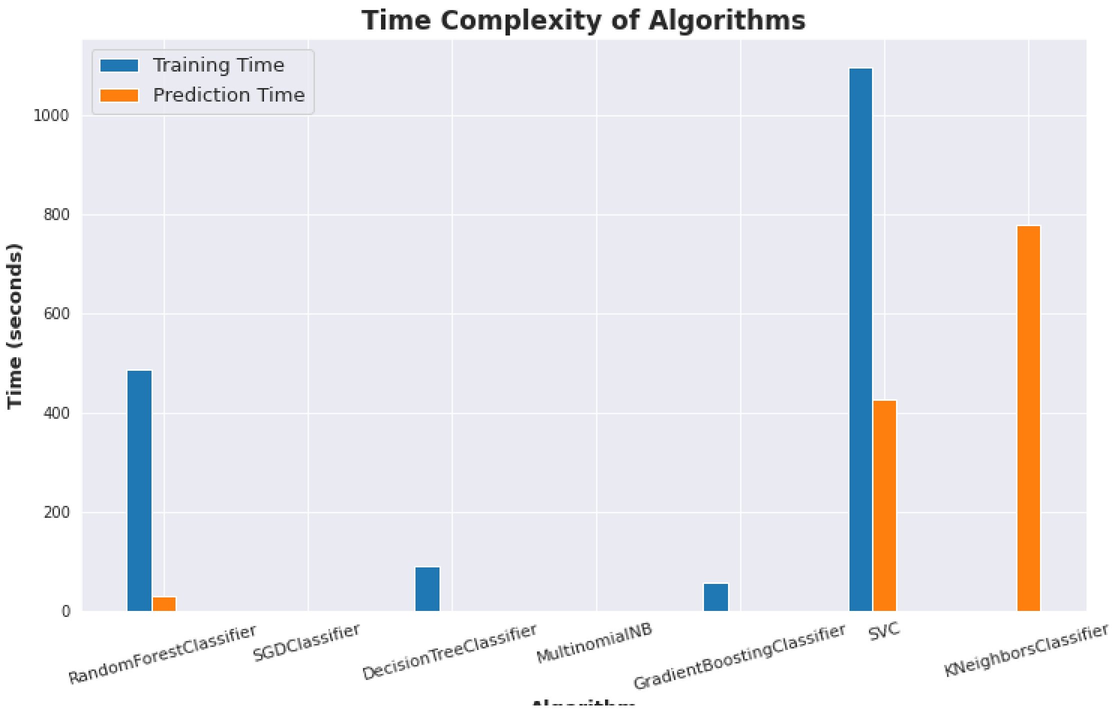


Figure 4.12: Bar Chart showing Time Evaluation of different algorithms for Obscene.



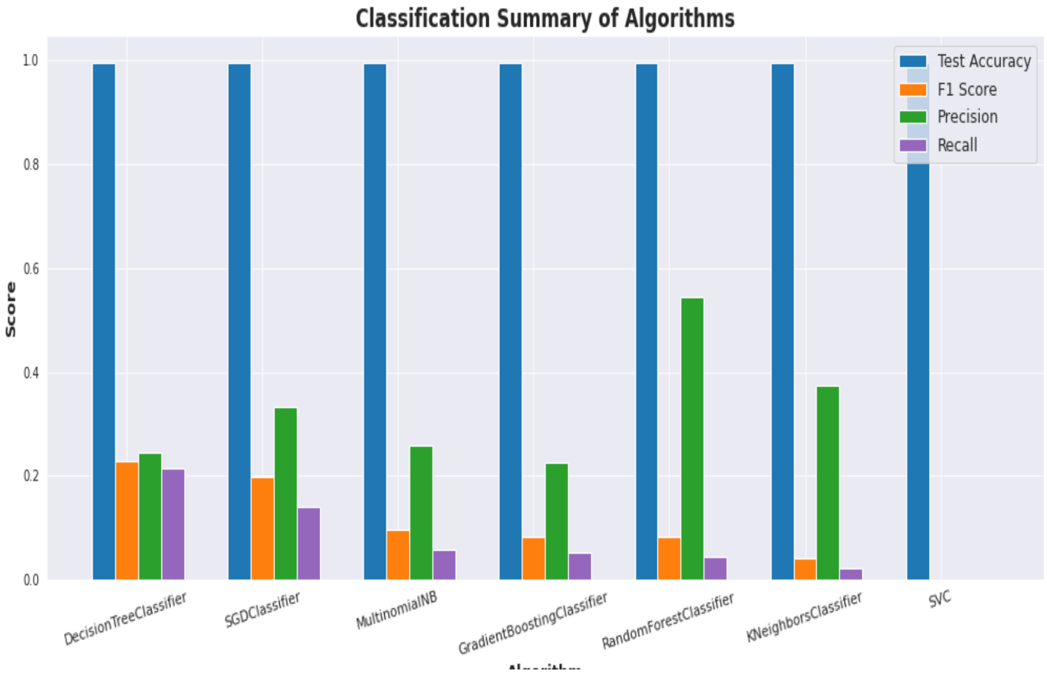
In case of time complexity analysis we have best training time for KNN with Training Time of 0.015 sec and Worst Training Time of 1099.801 sec for SVC. The best prediction is given by SGD model with prediction time of

0.017 sec while the worst is for KNN with prediction time of 781.1 sec.

- For Threat (Inappropriate) Content results are following:

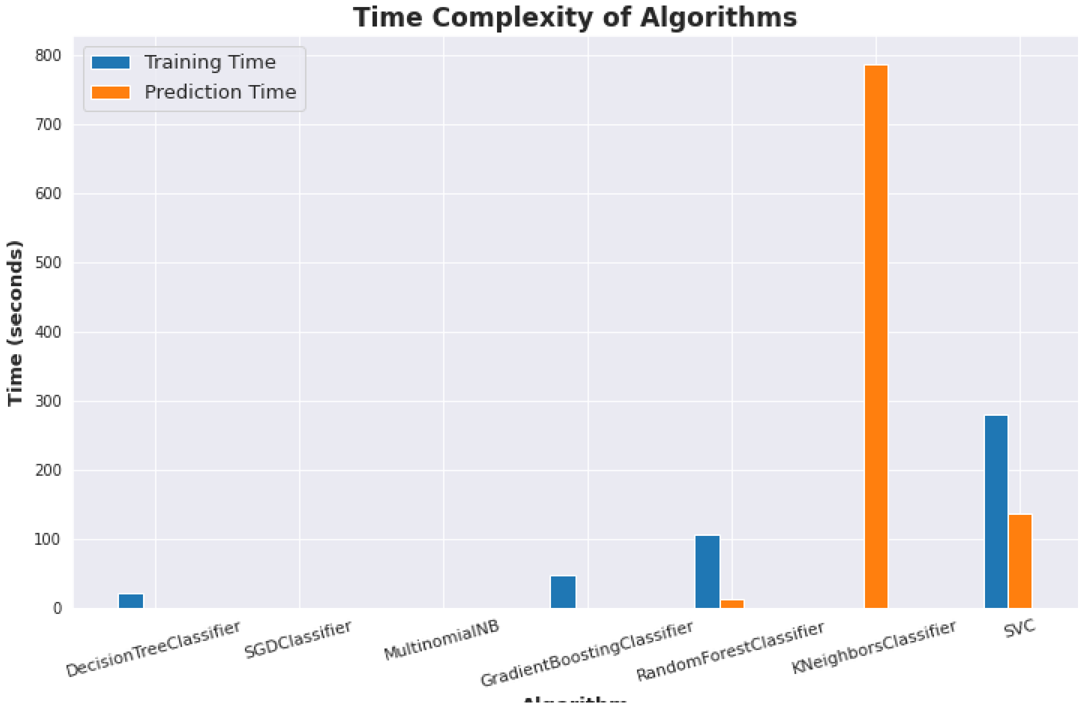
Result of Stated Model for Threat(0.3%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	DecisionTreeClassifier	0.24	0.21	0.22	99.50%
2	SGDClassifier	0.33	0.13	0.19	99.61%
3	MultinomialNB	0.25	0.05	0.09	99.62%
4	GradientBoostingClassifier	0.22	0.05	0.08	99.62%
5	RandomForestClassifier	0.54	0.04	0.08	99.66%
6	KNeighborsClassifier	0.37	0.022	0.041	99.65%
7	SVC	0.00	0.00	0.00	99.65%

Figure 4.13: Classification Summary for Threat.



In Classification Algorithms summary we can see the for Threat Class the best model having highest accuracy & F1 Score is Random Forest Classifier with an accuracy of 99.7% and DT with F1 Score of 22.7% respectively. The model performing best w.r.t precision is Random Forest Classifier with precision of 54.5%. The best recall Model is DT with an recall of 21.3%.

Figure 4.14: Bar Chart showing Time Evaluation of different algorithms for Threat.



In case of time complexity analysis we have best training time for KNN with Training Time of 0.013 sec and Worst Training Time of 281.285 sec for SVC. The best prediction is given by SGD model with prediction time of 0.018 sec while the worst is for KNN with prediction time of 787.83 sec.

- For Insult (Inappropriate) Content results are following:

Result of Stated Model for Insult(4.9%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	KNeighborsClassifier	0.79	0.29	0.42	96.14%
2	GradientBoostingClassifier	0.82	0.35	0.50	96.48%
3	SVC	0.84	0.39	0.53	96.65%
4	DecisionTreeClassifier	0.61	0.60	0.61	96.22%
5	MultinomialNB	0.70	0.55	0.62	99.68%
6	RandomForestClassifier	0.77	0.51	0.62	99.9%
7	SGDClassifier	0.78	0.53	0.63	96.99%

In Classification Algorithms summary we can see the for Threat Class the

Figure 4.15: Classification Summary for Insult.

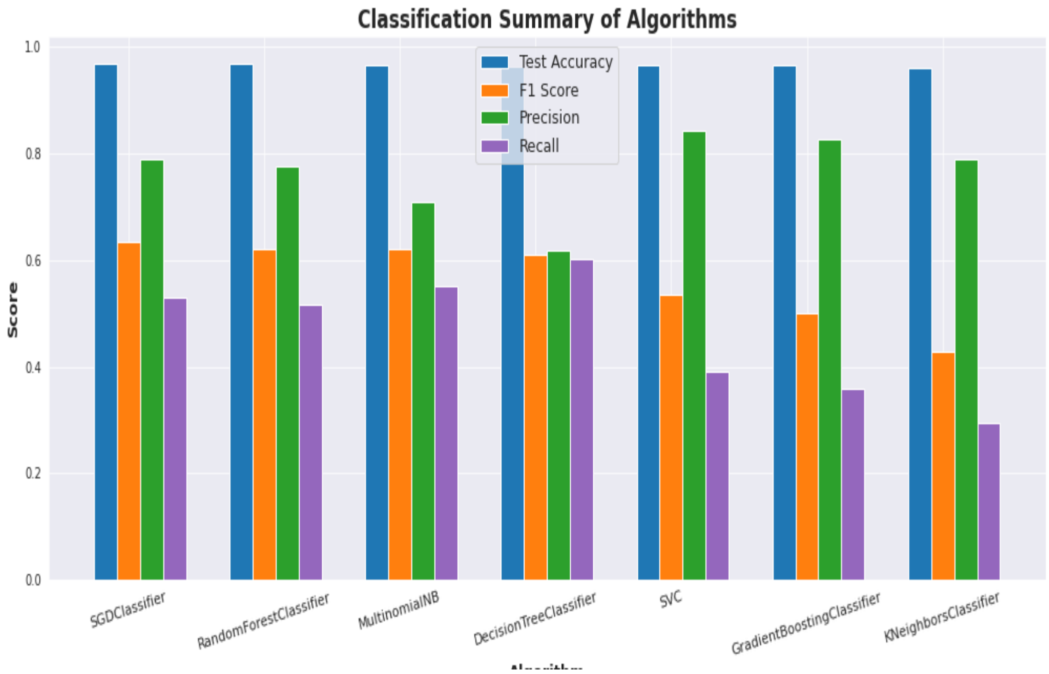
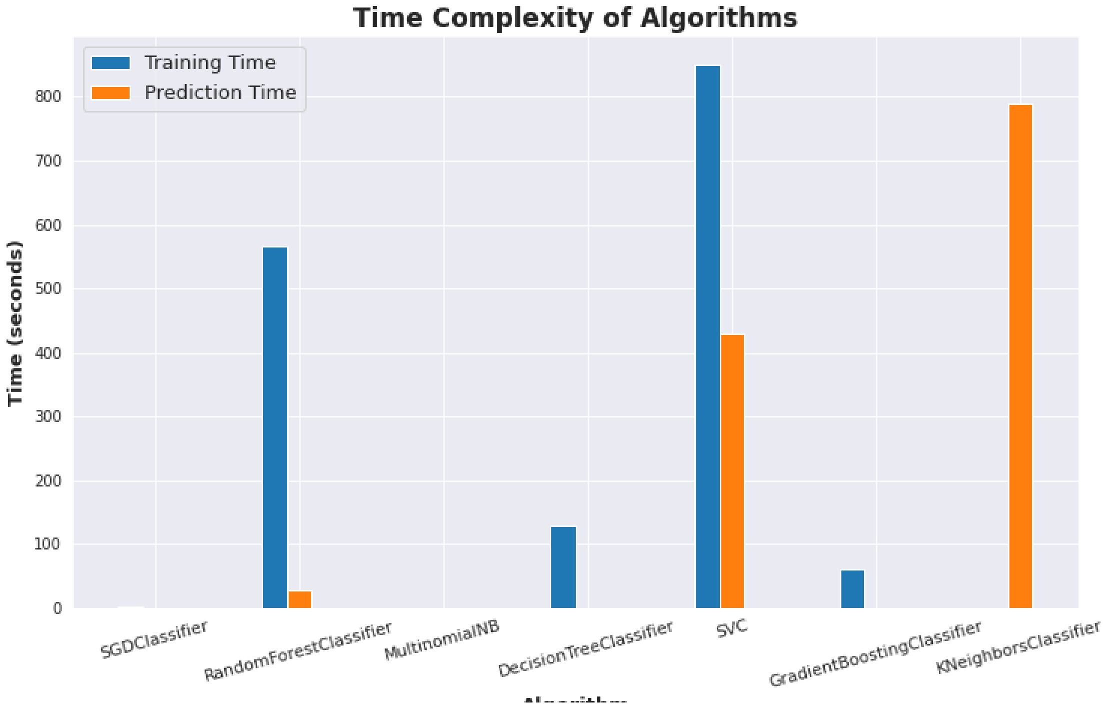


Figure 4.16: Bar Chart showing Time Evaluation of different algorithms for Insult.



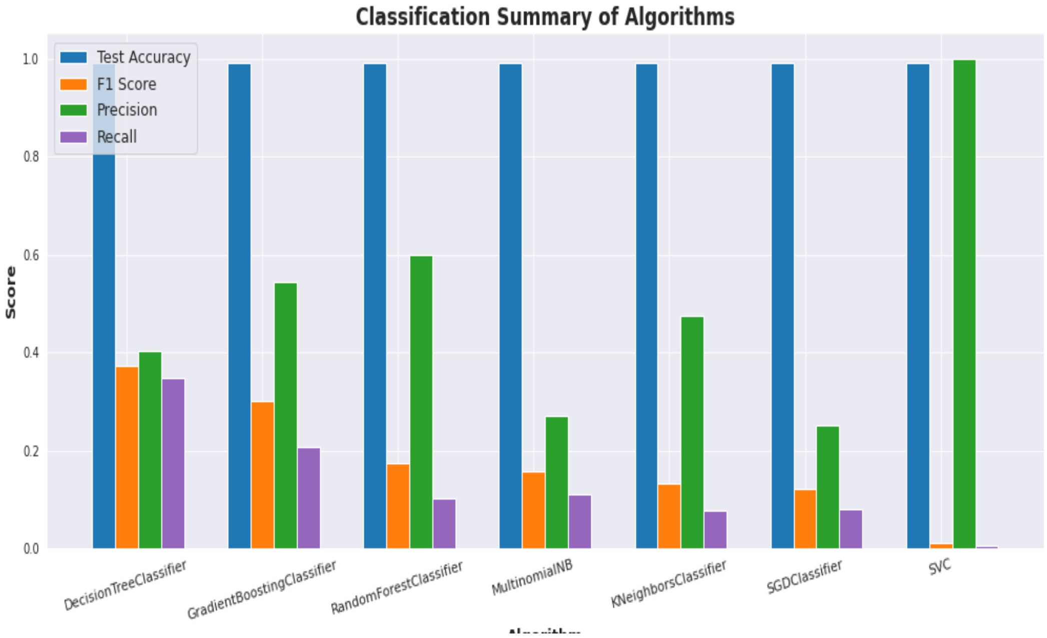
best model having highest accuracy & F1 Score is SGD Classifier with an accuracy of 97.0% and with F1 Score of 63.4% respectively. The model performing best w.r.t precision is SVC with precision of 84.3%. The best recall Model is DT with an recall of 60.2%.

In case of time complexity analysis we have best training time for KNN with Training Time of 0.015 sec and Worst Training Time of 851.040 sec for SVC. The best prediction is given by SGD model with prediction time of 0.018 sec while the worst is for KNN with prediction time of 790.03 sec.

- For Identity-Hate (Inappropriate) Content results are following:

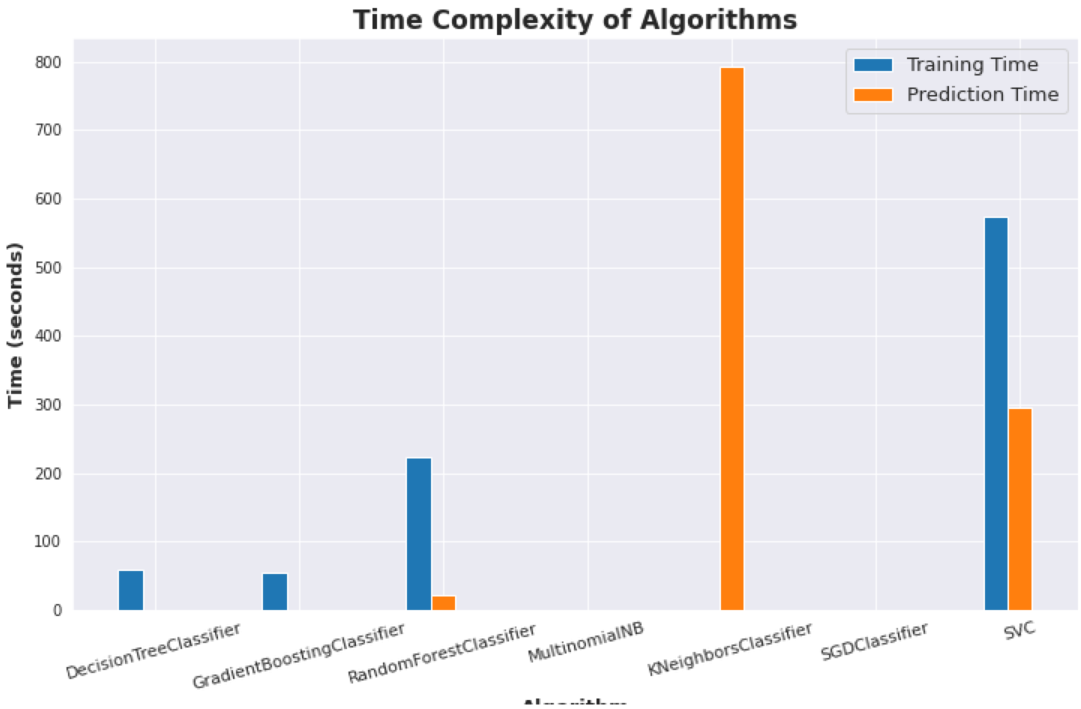
Result of Stated Model for identity˙hate(0.9%) content in detail					
S.No.	ML Model	Precision	Recall	F1 score	Accuracy
1	MultinomialNB	0.27	0.11	0.15	98.94%
2	DecisionTreeClassifier	0.40	0.34	0.37	98.96%
3	SGDClassifier	0.25	0.079	0.12	98.97%
4	KNeighborsClassifier	0.47	0.07	0.13	98.10%
5	SVC	1.0	0.0056	0.011	99.12%
6	RandomForestClassifier	0.60	0.101	0.17	99.14%
7	GradientBoostingClassifier	0.54	0.20	0.29	99.14%

Figure 4.17: Classification Summary for Identity-Hate.



In Classification Algorithms summary we can see the for Identity Hate Class the best model having highest accuracy & F1 Score is Gradient Boosting Classifier with an accuracy of 99.1% and DT with F1 Score of 37.3% respec-

Figure 4.18: Bar Chart showing Time Evaluation of different algorithms for Identity-Hate.



tively. The model performing best w.r.t precision is SVC with precision of 100.0%. The best recall Model is DT with an recall of 34.8%.

In case of time complexity analysis we have best training time for KNN with Training Time of 0.013 sec and Worst Training Time of 572.827 sec for SVC. The best prediction is given by SGD model with prediction time of 0.018 sec while the worst is for KNN with prediction time of 793.77 sec.

4.5 Summary

For spam classification, SVC is the best model with enhanced accuracy of 90%. The category of inappropriate content comprises six distinct sub-categories. Toxic content is best predicted by the SGD model with 95.6% accuracy. Severe toxic tweets are best predicted by the SVC model with 99% accuracy. Obscene tweets are best filtered out by the Random Forest model with an accuracy of 97.88%. Threat content is best predicted by the SVC model with an outstanding accuracy of 99.65%. Insult tweets are best detected by the SGD model with 96.99% accuracy. Identity hate tweets are best filtered out by the Gradient Boosting model with 99.14%. None of the categories is best predicted by either the Decision Tree model or Multinomial NB model.

CHAPTER 5

Conclusion and Future Work

Every single second, an enormous amount of data is being generated on different social media platforms through messages, videos and images. It is an impossible task to detect inappropriate content in the vast amount of data generated by users. An algorithm is needed to automate the task of finding inappropriate content on the fly. In this work, a machine learning models re addressed to detect spam and inappropriate content on social media.

The aim of this study is to classify messages in different classification like threat, obscenity, insults,identity-Hate, toxic and aggressive for differet class of inappropriateness if the message is inappropriate. The dataset used to train and test the model is taken from Kaggle. Our model has predicted the best result with an accuracy of 90 in spam detection and an average accuracy of 97 for predicting inappropriate content on the fly. Our model can be used to detect real spam and inappropriate message present on social media.

Bibliography

[1] Insult. *https://en.wikipedia.org/wiki/Insult.*

[2] Online harassment field manual. *https://onlineharassmentfieldmanual.pen.org/defining-online-harassment-a-glossary-of-terms/.*

[3] Online hate. *https://www.esafety.gov.au/young-people/online-hate.*

[4] Punishment for publishing or transmitting obscene material in electronic form. 2017.

[5] A framework for real-time spam detection in twitter. *IEEE 2018*, 2018.

[6] T. A. Almeida and G. Hidalgo. p J., M. Sms spam detection dataset. *Availabe at. http://www.dt.fee.unicamp.br/ tiago/smsspamcollection/*, 2011.

[7] Despoina Chatzakou Ilias Leontiadis Jeremy Blackburn Gianluca Stringhini Athena Vakali Michael Sirivianos Nicolas Kourtellis Antigoni-Maria Founta, Constantinos Djouvas. Large scale crowdsourcing and characterization of twitter abusive behavior. *roceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)*, 2018.

[8] Ponnurangam Kumaraguru † † Indraprastha Institute of Information Technology India Arizona State University USA anupamaa@iiitd.ac.in arajades@asu.edu pk@iiitd.ac.in Anupama Aggarwal †, Ashwin Rajadesingan . Phishari: Automatic realtime phishing detection on twitter. *IEEE*, 2012.

[9] Leon Derczynski Bertie Vidgen. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *The Alan Turing Institute, London, United Kingdom, Department of Computer Science, IT University of Copenhagen, Copenhagen, Denmark*, 2020.

[10] Sahar Bosaeed, Iyad Katib, and Rashid Mehmood. A fog-augmented machine learning based sms spam detection and classification system. *Conference on Signal and Image Processing (ICSIP)*, 2020.

[11] Member IEEE Yi Xie Yang Xiang Senior Member IEEE Wanlei Zhou Senior Member IEEE Mohammad Mehedi Hassan Abdulhameed AlElaiwi Chao Chen, Jun Zhang and Majed Alrubaian. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE transactions on computational social systems, vol. 2, no. 3, september 2015 65 a performance evaluation*, 2015.

[12] Amy Bellmore Chelsea Olson. Online aggression. *Elseiver*, 2013.

[13] Xuefeng Li Yixian Yang Chensu Zhao, Yang Xin and Yuling Chen. A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data. *MDPI applied science*, 2020.

[14] Mihaela Dinsoreanu Cristina Rădulescu and Rodica Potolea. Identification of spam comments using natural language processing techniques. *IEEE*, 2014.

[15] David Carmel Gilad Mishne and Ronny Lempel. Blocking blog spam with language model disagreement. *In Proceedings of the First International Workshop on Adversarial Information Web (AIRWeb), Chiba, Japan, May 2005, pp. 1-6.*, 2005.

[16] Víctor González-Castro Andrew C. Parnell Gonzalo Molpeceres Barrientos, Rocío Alaiz-Rodríguez. Machine learning techniques for the detection of inappropriate erotic content in text. *International Journal of Computational Intelligence Systems (2020)*, 2020.

[17] Manoj K. Chinnakotla Harish Yenala, Ashish Jhanwar and Jay Goyal. Deep learning for detecting inappropriate content in text. *International Journal of Data Science and Analytics (2018)*, 2018.

[18] Mourad Ykhlef Monirah Abdullah Al-Ajlan. Deep learning algorithm for cyberbullying detection. *International Journal of Advanced Computer Science and Applications (IJCSA)*, 2018.

[19] Kyle Dent Ameya Bhatawdekar Sheikh Muhammad Sarwar Momchil Hardalov Yoan Dinkov Dimitrina Zlatkova Guillaume Bouchard Isabelle Augenstein Preslav Nakov, Vibha Nayak. Detecting abusive language on online platforms: A critical analysis. *Cornell University*, 2021.

[20] Nittaya Kerdprasop Pumrapee Poomka, Wattana Pongsena and Kittisak Kerdprasop. Sms spam detection based on long short-term memory and gated recurrent unit. *International Journal of Future Computer and Communication, Vol. 8, No. 1, March 2019*, 2017.

[21] Senior Member IEEE Niloy Ganguly Rajesh Basak, Shamik Sural and IEEE Soumya K. Ghosh, Member. Online public shaming on twitter: Detection, analysis, and mitigation. *IEEE transactions on computational social systems, vol. 6, no. 2, april 2019*, 2019.

[22] Julian Risch and Ralf Krestel. Toxic comment detection in online discussions.

[23] Thomas Mandl Chintak Mandalia Sandip Modha, Prasenjit Majumder. Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance. *Expert Systems with Applications ELSEVIER (2020)*, 2020.

[24] Yulan He Semiu Salawu and Joanna Lumsden. Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on affective computing, vol. 11, no. 1, january-march 2020*, 2020.