A Hybrid Deep Learning Model and Explainable AI-based Bengali Hate Speech Multi-label Classification and Interpretation

by

Mahmudul Hasan Shakil
21166034

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

Department of Computer Science and Engineering
BRAC University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/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.
5. I would like to request the embargo of my thesis for 6M from the submission date due to a journal publication of my research.

Student's Full Name & Signature:

Mahmudul Hasan Shakil

21166034

Approval

The thesis titled "A Hybrid Deep Learning Model and Explainable AI-based Bengali Hate Speech Multi-label Classification and Interpretation" submitted by

1. Mahmudul Hasan Shakil (21166034)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on September 29, 2022.

Examining Committee:

External Expert Examiner:
(Member)

Dr. Ashis Talukder, Ph.D.

Associate Professor
University of Dhaka

Internal Expert Examiner:
(Member)

Dr. Md Khalilur Rhaman

Associate Professor
Department of Computer Science and Engineering
BRAC University

Internal Expert Examiner:
(Member)

Dr. Md. Ashraful Alam

Assistant Professor
Department of Computer Science and Engineering
BRAC University

Supervisor:
(Member)

Md.Golam Robiul Alam, Ph.D.

Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Amitabha Chakrabarty, Ph.D.

Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D.

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Data innovation has moved quickly in recent years, and various unfavorable alterations have been made to the network medium. Social media platforms like Facebook, Twitter, and Instagram are becoming more and more popular because they allow users to express their opinions through messages, photographs, and notes. In particular, in Bangladesh and other locations where the Bengali language is spoken. In any case, it has regrettably turned into a space with toxic remarks, cyberbullying, and unidentified hazards. Numerous studies have been conducted in this area, but none have produced accurate results. Some effective pre-trained transformer models have been introduced. To identify Bengali malicious and non-malicious text at an early stage using simple Natural Language Processing (NLP). This study suggests a Convolutional Neural Network with Bi-Directional Long Short-Term Memory (CNN-BiLSTM) hybrid strategy. This model can also classify any Bengali text data into six levels. Additionally, the transformed dataset is subjected to several conventional Machine Learning methods using an estimator, and Explainable AI interprets these techniques (XAI). In the last stage, Stacking Classifier which is superior to any prior activity is used to ensemble all classifiers and the estimator.

Keywords: Cyberbully; Natural Language Processing; Transformer; CNN; Bi-LSTM; Machine Learning; Explainable AI.

Acknowledgement

This is the work of Mahmudul Hasan Shakil – a student of the CSE department of BRAC University. The paper has been prepared as an attempt to consolidate the information I have acquired during these two 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 me inspiration, strength, and ability to complete this analysis.
I am particularly grateful to my thesis supervisor, Md. Golam Robiul Alam, Ph.D., for his immense assistance, encouragement, and support in completing my research.
I am also grateful to the BRAC University Faculty Staff of Computer Science and Engineering, who have been a light of guidance for me throughout the BRAC University study era, particularly in educating and enhancing my knowledge.
Finally, I would like to express my heartfelt gratitude to my dear parents for their love and care and our friends' 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

AI Artificial Intelligence

AUC Area Under Curve

$BERT$ Bidirectional Encoder Representations from Transformers

$Bi-LSTM$ Bi-Directional Long Short-Term Memory

CNN Convolutional Neural Network

FN False-negative

FP False positive

GBM Gradient Boosting Machine

$LIME$ Local Interpretable Model-agnostic Explanations

$LSTM$ Long Short-Term Memory

ML Machine Learning

NLP Natural Language Processing

ROC Receiver Operating Characteristics Curve

$SHAP$ Shapley Additive Explanations

SVM Support Vector Machine

$TFIDF$ Term frequency-inverse document frequency

TN True negative

TP True positive

XAI Explainable Artificial Intelligence

$XGBoost$ Extreme Gradient Boosting

Chapter 1

Introduction

This chapter contains some key information about the topic of my thesis. I've discussed my motivations for conducting this study, especially in the background section. The names of the algorithms I'm using are included in the research subject. In the section on the research objective, I discussed the essential components I want to incorporate in my writing.

1.1 Background Research

Cyberbullying is a form of bullying or intimidation that uses electronic media to target victims. Cyberbullying has become widespread among younger generations in recent years due to the rapid expansion of the digital world. Like domestic violence, cyberbullying against women has also increased during the COVID-19 pandemic. Bangladesh has laws and regulations to prevent this and the penalties are very strict. However, the process is often too long or difficult. In addition, lack of awareness of the issue and slow processes and adjudication remain major challenges. This fact was mentioned at the "International Fortnight to Resist Violence against Women" round table held in Dhaka in November 2021. Technology has risen to the status of a basic necessity in our current age. The internet has evolved into a massive knowledge base that millions of people access and rely on every day as the pace of technological advancement quickens. Abuse through online means of communication has grown to be a significant global issue as a result of the enormous growth in social connections and confidentiality in online social networks. Digitalization shifts human interaction to online platforms, which has certain advantages but also leaves the potential for harmful online behavior including abuse, harassment, and hate speech. On several websites, including online gaming, YouTube, and others, harassment of children and teenagers as well as adults is prevalent. According to the 2014 large-scale EU youth internet report [2], 20% of 11 to 16-year-olds have access to hateful content online. In addition, children were 12% more likely in 2014 than in 2010 to be exposed to cyberbullying, indicating that the problem is only getting worse. Building models that define cyberbullying is crucial because one of the main issues with online harassment studies is the lack of sufficient dataset accessibility. Several researchers have created their repositories from social media platforms including YouTube, Form Spring, Kaggle, Twitter, Instagram, MySpace, and ASKfm that are susceptible to bullying content using the datasets that have been made public in recent years [10]. The necessary measures against hate speech have been attempted

by social media sites, and they are still looking for ways to automate the process. Around 1% of students reported being singled out for discrimination based on their sexual orientation, gender, or race [4].

1.2 Research Problem

The rapid growth of Internet usage in Bangladesh has been a steady trend over the past decade. However, harassment of women is believed to be on the rise due to widespread patriarchal attitudes and norms and the lack of adequate legal protections. According to a survey [61] sponsored by ActionAid Bangladesh, 50% of Bangladeshi women have complained of online harassment. Over 62% of the victims are under 25 years old. Interestingly, victims cited Facebook as the platform under the most pressure. About 25% of these women victims did not take action against the perpetrators. 76% of women experienced psychological problems such as depression and anxiety as a result of these problems. Some 48% of victims felt it was not worth filing a complaint, and 52% reported it to the government's Cybercrime Investigation Department. 30% of women did not know where to apply. Because of technology improvements, cyberbullying or online harassment using derogatory, offensive, or cruel language has become quite simple for this generation. However, these offensive remarks he can potentially make the target mentally ill, and some y even begin to have anxieties. In this study, I used deep learning to look for offensive remarks on social media sites and then categorized them into Not-bully, Religious, Sexual, Threat, Troll, and Slang categories. Additionally, I made an effort to gauge how effective each algorithm's datasets were.

1.3 Research Objective

Modern technology has made it much too simple to engage in online harassment or provocation using repressive, bigoted, or hurtful rhetoric. Online media has developed into a hub of information and photos about the area as a result of its availability and accessibility. Everyone was motivated to become involved before the others, thanks to it. Unfortunately, while still placing them at risk for injury, this stage hates and scrutinizes people who develop a range of sexual identities, paths, and orientations. A variety of customers may have actual worries as a result of online bullying and provocative behavior, which may also result in major psychological issues including melancholy and self-destruction. During the pandemic, students' online activity has increased several times. Fake accounts are often created by collecting photos of victims from social networks. Many accounts are also hacked and offensive material is uploaded for public viewing. However, most students do not know how to stay safe online and have little or no understanding of cyberbully or online bullying. Orientation and training sessions on these issues for all stakeholders in relevant institutions can go a long way in reducing these online threats. Parental monitoring and supervision can play a positive role in preventing or correcting online bullying, so parents and guardians, in addition to students, should be involved in this process.

This research aims to make sure that online media does not contain any objectionable content or remarks. The goal of this study is to apply comprehensive infor-

mation to identify the presence of hostile remarks on a web-based media network. It shouldn't be labeled as poisonous, very toxic, profane, insulting, threatening, or hateful of anyone's identity. I also made an effort to assess the accuracy of the data set used in each computation. In the beginning, I tried to apply some pre-trained Bengali transformers mechanisms and got some predicted outcomes. After that, I am using Natural Language Processing (NLP) and Convolutional Neural Networks (CNN) with Bi-directional Long Short-Term Memory (Bi-LSTM). To create the recommended architecture, data preprocessing techniques were combined with NLP techniques like tokenization and stemming to turn words into vectors. This architecture is capable of identifying speech and categorizing it into five groups. Then, using a baseline estimate for binary classification in the following phase, I applied conventional classifiers to the collection of processed data from the initial phase. I used a variety of explainable AIs to understand these classifiers. I used several Machine Learning Classifiers to categorize the produced dataset. As a baseline estimator, I utilized logistic regression. Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) are the two terms I picked to explain. The accuracy I obtained after combining such machine learning classifiers with Stacking classifiers is remarkable to any earlier study.

1.4 Research Contributions

In this work, we have introduced a Hybrid model based on Convolutional Neural Network (CNN) and Bi directional Long Short-Term Memory (Bi-LSTM) with Natural Language Processing (NLP) for Bengali hate speech classification. Later on, we have implemented various Machine Learning Models along with Explainabilty. Specifically, the main contributions of the paper are summarized as follows:

- We modified the data set to multi-label data from single-label data. So, it can classify more accurately according to the comment. A sentence may express both the Sexual and the Religious sentiment, by multi-label classification we can detect both of the sentiments.
- We added a new class and named it as 'Slang'. Before this, it was not possible to detect slang type comment. Now, slang type comment detection can be possible.
- Our hybrid model performs better than most of the pre-trained transformer based models. Pre-trained models add bias from the pre training to the model. As a result, our trained model performs better.
- After model fitting, we got a processed data set and applied it through multiple machine learning models which addressing a fusion in Bengali Machine Learning studies.
- We have applied Explainable AI's to these machine learning models and evaluated classification reports. It is a revolutionary work for Bengali data set explanation.

- We made an ensemble model after stacking all the Machine Learning Models. There is no previous work in the field of Bengali Machine Learning where anyone addressed this kind of ensemble learning model.

1.5 Thesis Report Outline

The remaining sections of this research study are as follows.

The background analysis and literature review are both included in Chapter 2. The various machine learning techniques and ongoing research on these topics will be shown.

The technique in Chapter 3 shows the overall format of our efforts. It will display the whole cycle of my suggested model with the proper outlines. This section will provide details about the datasets, information preprocessing, and a model for highlight extraction and categorization.

The model's analysis and outcomes will be discussed in chapter 4. It will go over the results with a categorization report and a disorganized grid. This section will outline the parameters for determining the connection between our results and the conclusions drawn from them.

Chapter 5 will conclude the paper by summarizing all we've done. We have outlined our research in chapter 5 and discussed its scope, limits, and future directions.

Chapter 2

Literature Review and Related Works

The proliferation of online media, the misuse of internet activity, and the usage of obscene language have recently become issues. To resolve this issue, no significant actions have been performed. Here is a sample of the documents I am looking at:

2.1 Bengali Speech related works

This study [72] mostly focuses on BNLP subfields such as sentiment analysis, speech recognition, optical character recognition, and text summarization. A thorough analysis of the most modern BNLP tools and techniques appears to be lacking in resources. Because of this, the authors analyze 75 BNLP research papers in-depth in this paper and group them into 11 categories: information extraction, machine translation, named entity recognition, parsing, parts of speech tagging, question answering system, sentiment analysis, spam, and fake detection, text summarization, word sense disambiguation, and speech processing and recognition.

A deep learning-based implementation for speech emotion identification is presented by the authors in this work [73]. The system combines a time-distributed flatten (TDF) layer with a deep convolutional neural network (DCNN) and a bidirectional long-short-term memory (BLSTM) network. The recently constructed audio-only Bangla emotional speech corpus SUBESCO has been used to test the suggested model. All the models mentioned in this study were examined in a series of experiments for baseline, cross-lingual, and multilingual training-testing setups. The experimental results show that the model with a TDF layer outperforms other cutting-edge CNN-based SER models that are capable of representing emotions both temporally and sequentially.

The goal of this project [71] is to gather data on Bangla comments posted on social media sites to build a classifier model that can rapidly and accurately determine whether a remark is social or anti-social. From Facebook and YouTube, two well-known social media channels, 2000 comments were collected. To discriminate between rude and polite remarks, their study used supervised machine learning classifiers including Logistic Regression (LR), Random Forest (RF), Multinomial Naive Bayes (MNB), and Support Vector Machine (SVM) as well as artificial neural network models like Gated Recurrent Unit (GRU). Finally, in their study, language

models such as unigrams, bigrams, and trigrams have been used.

This article [28] uses a convolutional neural network to create a Bangla number recognition system from a spoken stream (CNN). The proposed method uses Mel Frequency Kepstrum Coefficient (MFCC) analysis to extract relevant characteristics from speech signals. The speech dataset, which comprises 6000 utterances of 10 isolated Bangla digits (5 utterances for every 120 speakers), was produced. After that, CNN is trained using the voice signal's characteristics as input.

In another study [12], several Deep Neural Network (DNN-HMM) and Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) based models for speech recognition in Bangla were examined to construct a voice search module for the Pipilika 1 search engine. For this effort, a small corpus of 9 hours of voice recordings from 49 distinct speakers with a vocabulary of 500 unique words was assembled.

This article [44] describes how to identify Bangla broadcast speech using support vector machines (SVM). This approach is based on the MATLAB platform and linear kernel SVM. In this experiment, four different forms of noisy broadcast voice samples are employed, and SVM produces a decent result.

This study [25] shows a Bangla corpus that has been open-source licensed and has been designed with sentiment analysis in mind1. Over 10,000 texts have been gathered and carefully annotated with sentiment polarity. They then switched to the Word domain and added sentiment polarity annotations to more than 15,000 words that were produced from these phrases. To guarantee the quality, at least two and occasionally three annotators have cross-annotated every entry in the corpus. They had to develop a secondary corpus for Bangla word stemming as a need for producing a high-quality sentiment analysis corpus, and for cross-validation by at least two and occasionally three annotators to guarantee quality.

They [17] need the Bangla datasets to complete the work. The Bangla dataset, however, is not accessible. They thus gathered information from Facebook. It takes a lot of work to get data from social networks. The information includes grammatical errors and many languages. So, they formed a team to gather the information. Processing the data fell to a different crew. The data was then classified as hate speech or not. The group's members knew enough about hate speech. They had no bias towards the information. Their data includes hate speech directed against women, the disabled, the community, culture, ethnicity, race, and sex. For their task, a machine learning method is perfect. For the study, they utilized the SVM and Nave Bayes methods.

They provide [14] an extensive collection of methods to recognize sentiment and extract emotions from Bangla texts in this article. To categorize a Bangla phrase with a three-class (positive, negative, neutral) and a five-class (strongly positive, positive, neutral, negative, highly negative) sentiment label, they develop deep learning-based models. Additionally, they create models that can identify any one of the six fundamental emotions in a Bangla text (anger, disgust, fear, joy, sadness, and surprise). They assess the effectiveness of their model using a fresh dataset of comments from various YouTube videos in Bangla, English, and Romanized Bangla.

In this study [52], they put forth DeepHateExplainer, a comprehensible method for

hate speech identification from the under-resourced Bengali language. They use a neural ensemble method using transformer-based neural architectures (i.e., monolingual Bangla BERT-base, multilingual BERT-cased/uncased, and XLM-Roberta) to classify Bengali writings into political, personal, geopolitical, and religious dislikes after thoroughly preprocessing them. Then, using sensitivity analysis and layer-wise relevance propagation, important (most and least) phrases are determined (LRP). To assess the quality of explanations about fidelity, they finally compute comprehensiveness and sufficiency ratings.

Since Bengali text hasn't been studied in this specific field, the authors of this research [64] intend to develop a model that can evaluate the Bengali text and determine if the reviews are positive or negative. Surveys, various social media evaluations, ratings, and other methods are used to gather the data, which is then labeled. The data were trained in DL (Deep Learning) and ML (Machine Learning) models after cleaning and extracting several characteristics. By comparison, the Long-Term Short-Term (LSTM) model outperforms other models, it is discovered in the conclusion.

In this project [67], the team wants to develop a machine learning model to assess the reviews' sentiment. Internet users are growing daily in Bangladesh. So, they decided to create the Bangla language model. They couldn't find any meal review datasets in Bangla that they could utilize for their project. Then they gathered and classified more than a thousand evaluations of Bangla cuisine from several websites, like Food panda, Hungrynaki, Shohoz food, Pathao food, etc. They have taken different characteristics from cleaned data after the appropriate preprocessing, and they have utilized these features to train and evaluate machine learning and deep learning models.

2.2 Hate Speech related works

The author of this article [24] blends CNN and LSTM, or a modified version of it. a text classification model known as NA-CNN-LSTM or NA-CNN-COIF-LSTM without activation function is made available to CNN. The suggested model outperforms conventional CNNs or LSTMs, according to experimental results on subjective data sets and text categorization goals.

A DC (twin) technique that is much superior to CNN-LSTM is presented in this article [39]. Two channels are employed in this DC system to gain input at the word and character levels, respectively. It is advised to combine the current state of the unit of time and the current state of the spending hours for hybrid attention before focusing on the weight calculation. Weighted conclusions are produced after determining the probability distribution of the input weights for each time step. The input data from each time step is then trained to increase the generality of the model learning.

A CNN-LSTM hybrid model-based text categorization technique was put forth by the author [13]. Words are represented as vectors by the algorithm using Word2vec's Skip-Gram and CBOW templates, local text features are shown by CNNs, and context snippets are produced by LSTMs, which store historical data. Using the

SoftMax classifier for text dependence and classification, CNN replaces the output of opportunity vectors as an LSTM input. A test on the algorithm's ability to increase text classification accuracy was done on the Chinese news site Sogou.com.

In a different paper [69], the authors developed a stacking classifier in which they combined six existing classifiers—the Support Vector Machine, Artificial Neural Network, Logistic Regression, Decision Tree, Random Forest, and Gaussian Naive Bayes classifiers—into a single model that represents the logistic regression classification used for the six fundamental models tuned by hyperparameters.

According to the compatibility of the section, the authors of this research [62] provided a strategy that modifies the classification results using the stack classifier model. The inside is also described as being difficult to navigate for regular meetings. In the end, this strategy provides the desired classification outcomes straightforwardly and affordably.

Naive Bayes, SkLearn, Support Vector Machine, MultinominalNB, GaussianNB, and BernoulliNB were among the classification techniques employed by the authors [37] in this study. They pick diverse qualities to comprehend the precise position of various algorithms (numbers or digits). To select the algorithm with the highest accuracy, they tested the proposed algorithm and utilized ensemble learning to merge multiple categorization algorithms.

The effectiveness of the categorization of managed learning models in Turkish texts with various parameters was examined in this context [9]. These examples were used to test the system's ability to categorize novelties into five specified classes after the system had been trained using a variety of training materials. This is done by comparing and elucidating the classification performances of the Multinomial Naive Bayes, Bernoulli Naive Bayes, Support Vector Machine, K-Nearest Neighbor, and Decision Trees algorithms in Turkish news texts with various settings.

With the use of the audio from YouTube videos, they created [51] their dataset and evaluated it using several training models in this work. They explored certain deep learning strategies as well as machine learning models. They concluded their trials that the deep learning GRU model and logistic regression produce the greatest accuracy on the available dataset.

Due to people's disrespect for proper spelling, grammar, and punctuation, while posting comments online, the suggested solution requires a significant amount of preprocessing. They gathered [20] feedback and likes from public Facebook pages to create the dataset. They employed Unigrams, Bigrams, the number of likes, the categories of emojis, emotion scores, the number of abusive terms in each remark, as well as the offensive and threatening words that were used in the comments and were discovered using their suggested method. Profanity can also be detected using the method outlined above.

It is simple to locate the mistakes in conventional neural network-based short text classification algorithms for sentiment categorization. The Word Vector Model (Word2vec), Bidirectional Long-Term and Short-Term Memory networks (BiLSTM), and convolutional neural network (CNN) are coupled to provide a solution to this issue. The experiment [46] demonstrates that the Word2vec word embedding-

associated CNN-BiLSTM model's accuracy was 91.48%. This demonstrates that in brief text, the hybrid network model outperforms the single-structure neural network.

Based on enhanced BiLSTM-CNN, this research [38] suggests a sentiment categorization method. First, the user's words are turned into vectors by word frequency in the text, and then BiLSTM is used to extract the text's sequence characteristics. The data is then subjected to further feature extraction using CNN so that higher dimensional user information can be utilized for categorization. The experimental findings demonstrate that the model built using this strategy efficiently optimizes conventional algorithms while accounting for the link between various elements in the text. The performance is somewhat better when compared to similar conventional network models for text sentiment categorization.

A model of feature fusion[43] using convolutional neural networks (CNN) and bidirectional long short-term memories (BiLSTM) is presented to address the issue that the neural network structure utilized in the current text sentiment analysis job cannot extract the key characteristics of the text. The local features of the text vector are extracted using CNN, while the global features of the text context are extracted using BiLSTM. The issue of gradient disappearance or gradient diffusion in conventional recurrent neural networks (RNN) is effectively avoided, and the single CNN model no longer overlooks the semantic and grammatical information of words in the context. After the corpus has been preprocessed, the text is represented as a two-dimensional word vector matrix, and the CNN is then used to extract the local information features. The BiLSTM is then utilized as the input to understand the connection between the order of words and sentences. Then an attention mechanism is presented to draw attention to critical information. emotion's impact on text. The suggested feature fusion model effectively increases the accuracy of text categorization, according to experimental data.

In this research[56], they provide a novel approach to sentiment analysis that applies text categorization using the CNN-BiLSTM framework to the Bangla dataset. The text is categorized in this categorization scheme as either positive or negative. To create a deep learning model, they employed a dataset on Bangla news comments, which was composed of Bangla postings. They have prepossessed their dataset, which includes language translation, stop word deletion, tokenization, and high-frequency word extraction, to obtain accurate analytical findings. Then, layers of CNN and LSTM are created in such a way that they have accomplished their high accuracy to build the sentiment analysis model.

The Bert model is used in this structure as the text information extractor[74]. Once the Bert model has produced its output features, multi-head attention and the TEXT-CNN model are used to extract additional feature information. Finally, low-dimensional feature vectors with more dense semantic information are generated and combined to increase the information entropy of text vectors. Lastly, the loss function with flooding mechanism is used to perform backpropagation to further prevent the model's over-fitting problem on smaller data sets and improve the model's generalizability. The BiLSTM with self-attention is used to extract the information of various words in text information and then classify the text information.

This study suggests[59]a sentiment classification model for barrage text that includes the attention mechanism, the BiLSTM, and the ALBERT pre-training language model. Utilize the BiLSTM network to extract the text context relationship features, the ALBERT pre-training language model to obtain the dynamic feature representation of the barrage text, the attention mechanism to dynamically adjust the feature weights, and the SoftMax classifier to determine the sentiment category of the barrage text. Studies on the barrage text data set collected by the crawler reveal that the W2V-Att-CNN and W2V-Att-LSTM models with attention mechanism are compared by the ALBERT-ATT-BiLSTM model.

This study examines[49] the application of Bayesian and Random Forest to automatically optimize BiLSTM-CNN systems. They developed the ASR using the BiLSTM-CNN model and altered its hyperparameter values to take into account their low-hardware requirements when optimizing it. Additionally, they acquired 1,000 voice data snippets from different movies and categorized them into mood and stress groups. They employed the bigram textual feature and transcription of their voice data to achieve contextual-level comprehension in their ASR.

In this article[11], text placement is perhaps the most widely used sign language to prepare for innovation. Typical applications of content curation include spam identification, news text clustering, data retrieval, sentiment investigation, target discovery, and more. it includes. Thanks to the deep learning network, these deficiencies are greatly improved, avoiding difficult component extraction measures, and having strong learning ability and high prediction accuracy. An example is the Convolutional Neural Network (CNN). This article introduces text ordering cycles and demonstrates a deep learning model used for content clustering.

The mechanization of data mining from the skill models in this article[5] provides findings that successfully use the data for automated retrieval in clinical databases. Most eligibility standards contain variable data related to terms and conditions. The company is developing a new natural language processing (NLP) pipeline that extracts and describes ad-hoc data in order from current, royalty-free, in-memory models of text.

In another article[21], content sequencing is the main initiative in natural language processing (NLP) that is expected to gain relative classification capabilities. A message with many classes. Today, neural tissue models are widely used in the field of NLP and have produced surprising results in content clustering. In any case, due to the large spatiality of textual data, standard languages have impressive linguistic, square dimensions. Different directions of development in the organizational structure of text placement so far. To accommodate the aforementioned problem, this paper proposes a different organizational structure that includes bidirectional long-term memory (LSTM) combined with a progressive balance system. in the organizational structure. The information is passed into bidirectional LSTMs after one-hot encoding, and the output is subject to incremental revision. Finally, a SoftMax classifier is used to cluster the processed parametric data. Test results show a high degree of accuracy in structure when grouping content.

In another article[27], the innovation and rapid expansion of electronic exchanges have made e-mail a truly specialized device. In many applications, such as business

correspondence, updates, scientific notices, website page engagement, and email, email is an important communication method. Ignoring spam messages leaves a lot of messages every day. You must examine the subject or content of each email to determine the meaning of the message you received. In this study, we proposed an unassisted framework that describes the received message. The direction of an incoming message is determined by a common language processing strategy called Word2Vec computation. e.g., trained information refers to calculations using individually trained k-models.

In another article[32], text classification is a key part of NLP. This means that the classes for a particular record must be placed in a particular sequence frame. There are several ways to identify highlights and feature patterns. However, most experts prefer to use methods implemented in external libraries to achieve their goals.

In this work[78], the generation of logical structures is a key problem in natural language processing (NLP), because most AI models cannot provide explanations for prediction. Existing methodologies for intelligent AI frameworks typically deal with revenue or the relationship between data sources and revenue. However, detailed data is often overlooked and frameworks do not provide clear explanations. To alleviate this problem more easily, they propose a new generative enhancement system that determines how to resolve layout choices while generating detailed interpretations. They would clarify the basic approach to logical factors and risk preparation, which will determine how more important interpretations can be made.

These are just a few of the articles written on this topic. One of these few articles was created for text classification using CNNs. Kim et al.[1] Show sentence-level classification performed using CNN. The authors describe how minimal tuning and static vectorization can produce observable results across multiple benchmarks suggested by them. This model is load dependent and uses static vectors. The authors also demonstrated the importance of pre-trained and unsupervised word vectors in NLP. As internet users and social networks proliferate, so does the level of toxicity of social platforms. Hateful comments about victims of cyberbullying have become a major concern, and the industry and analytics community are working hard to find economic models to predict cyberbullying comments online due to the importance of interactive online communication between users.

Here in this paper, Georgakopoulos, S. V, and others[8] solved the problem of classifying toxic comments using a CNN model. Their research focuses mainly on recent research. An approach to word representation and text classification using convolutional neural networks. For their study, they used a dataset collected from a Kaggle competition for editing Wikipedia talk pages. The authors used word embedding and CNNs to compare the BoW approach with a set of highly successful text classification algorithms. The main dataset was transformed into subsets for more consistent analysis and thus used for binary classification to filter out toxic comments.

Zhang et al[15] proposed a novel method that combines a convolutional neural network (CNN) and a gated recurrent network (GRU) to improve classification accuracy empirically. In this article, the term "hate" is replaced by a word that means insult, hostility, or abuse, but it seems that hate speech cannot be equated with oppressive

language. The authors created their dataset by collecting datasets from the publicly available Twitter dataset as well as tweets about refugees and Muslims. This tweet has been designated for media use thanks to Total at the time of writing. Some recent events. The author applied preprocessing to the tweets and worked the model using CNN + GRU architecture in the subsequent layers, word embedding layers, 1D convolutional layers, 1D max-pooling layers, GRU, and SoftMax layers. They performed a comparative study on the dataset and showed how the proposed method outperformed the benchmark and produced better results than other dataset reporting methods. Set the current standard as follows: Scores from 1 to 13% in F1 for 6 out of 7 datasets.

The paper[7] presents a deep learning framework for clustering text on Twitter. The rating assigns each Tweet one or more of four predefined ratings: racist, misogynistic, hateful (fantasy and sexist), or non-hate speech. Four variants of convolutional neural networks are ready to answer. 4-gram symbols, word vectors support linguistically broken word2vec, randomly generated word vectors, and word vectors related to n-gram symbols. The feature list was narrowed down using the SoftMax operation used by organizations to account for maximum aggregation and tweets. The cross-validated model with 10 overlays added to the Word2vec embedding performed best in our review, with high accuracy and an F-score of 78.3%. With the successful development of Internet users for informal long-distance communication, individuals speak their minds every day. Concepts like notes, photos, videos, and speeches. Classifying the text was a huge challenge because these large texts came from different sources and different perspectives from different groups of people. As a result, the understanding is wrong, contradictory, noisy, and even in different dialects. salvage. Indeed, NLP et al. Deep neural tissue techniques are often used to address these issues. In this regard, Word2Vec Word Implantation and Convolutional Neural Network (CNN) techniques should be applied to the clustered content. The model presented in this paper perfectly cleans information from pretrained Word2Vec models, generates word vectors, and uses CNN layers to better remove short sentence markers [41]. Engineering and Implementation of Convolutional Neural Networks (CNN). The main conclusions of the study. However, related terms and vocabulary are rarely used, and CNN fails to capture general aspects of sentences[16]. To do this, we use verbal input, comment input, and lexical input to encode the message, three special attention, attention vectors, long-short-term memory (LSTM) thinking, and mind integration with a CNN model. This article. We also examine words and locations to highlight the importance of words that represent revenue generated. To improve the performance of three different CNN models, constant cross-regression (CCR) and continuous learning techniques are introduced. It is important to see the first uses of CCR and mobile learning in high-text sensory research. Finally, analysis of two separate datasets showed that the CNN model performed the simplest or best compared to the progressive model using predictive inference.

Since XGBoost is relatively new, not much work has been done on text classification. However, we find in this article[22] that the author works with XGBoost to detect false information which is almost similar to the kind of work we do, and XGBoost uses LSTM, Random Forest and others to retrieve data sets, but it works like XGBoost finder. The authors used an attention-based model that uses only

textual information from other articles. The results show that the XGBoost model improved by 16.4% and 13.1% over the best baseline in terms of accuracy.

In another paper[29], the authors recognize and prevent online media hostility in mixed English-Indian and Hindi-English datasets. The developers used features like word vectors, power words, mood scores, parts of speech, and emoticons to work with the layout. They mentioned several ways to describe AI such as XGBoost, Support Vector Machine (SVM), and Gradient Boosting Classifier (GBM). Among them, is the best way to order XGBoost. Facebook, Twitter, etc. Send messages, share photos, make video calls or save appointments. Despite these ideals, he has a negative attitude that leads to hatred of a certain residential area. Therefore, it is necessary to sense and avoid this hostility, which is a major cause of unsympathetic online media behavior. In this paper, we have discussed Hindi, English and Hindi-English mixed datasets. This operation is usually suitable for Support Vector Machines (SVM). The performance of the three classifiers in the multiple-prediction survey format yielded independent f-scores of 68.13, 54.82, and 55.31 for the mixed data set.

English Wikipedia[26] has worked on cyberbullying and spam classification. The dominance of online communities is a hallmark of today's digital landscape. Cyberbullying is one of the constant threats to the ideals of free-flowing sharing in this society. From derogatory comments to personal attacks to spam, contributors analyzed the challenge online. Here, Wikipedia astronauts are protected by people to identify online behavior. They offer a structure for understanding English abuses and English in this article. They explore resources available for free on Wikipedia. They believe that Wikipedia's XML dumps need advanced information research capabilities to be used for ad hoc literature research, and as an alternative, they propose a web scraping procedure to gather consumer-level information and provide a wide range of research information resources. Investigate to get the highlights. Percentage of clients previously blocked for malicious activity. We use this information to generate misuse identification models that use conventional language processing procedures such as letter and word n-grams, rating descriptions, and pattern rankings, and to generate used highlights. Predict violence in models based on AI calculations as data sources. Best badger appearance model using XGBoost feature model, AUC score of 84%

In this paper[45], the authors observed depression using XGBoost. Depression is debilitating and widespread. It can be treated, but usually goes undiagnosed. Remote pain scans are desirable, but there are security concerns about using information from smartphones and online media. They suggest that message response lag may provide useful data in depression screening based on the recognized association between hopelessness and low data processing speed. They distinguish nine latency-related characteristics of meta-information, namely public sponsored chat messages. Address security vulnerabilities by checking text metadata, not content. We review several machine learning techniques that focus on key areas of inactivity to predict relevant screening summary ratings. Their results show that the XGBoost model with a key segment achieves an F1 score of 0.67, an AUC of 0.72, and an accuracy of 0.69. Thus, they argue that responses to blank messages are offered as a way to sort out despair.

Chapter 3

Methodology

Previously, I applied Hate Speech Classification Implementing NLP and CNN with Machine Learning Algorithm Through Interpretable Explainable AI. After that, I applied Toxic Voice Classification Implementing CNN-LSTM & Employing Supervised Machine Learning Algorithms Through Explainable AI-SHAP. In both cases, I tried English hate speech classification. Finally, I tried Bengali hate speech classification and evaluate this with machine learning explainability.

To classify Bengali hate speech, I have used a hybrid deep learning model that has been combined with Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (Bi-LSTM). Primarily, I have collected the dataset from the research gate. After collecting the dataset, I divided the dataset into multilabel with existing data annotation. Secondly, I have added more data with proper annotation to the existing dataset. For comparison, I have used multiple pre-trained transformer-based models. After getting an improved accuracy, I used the processed dataset through multiple machine learning algorithms. I tried to show the Explainability of these machine learning models. Finally, I stacked all the machine learning classifiers and made an ensemble model with an outstanding accuracy rate. In the following sections, I will be discussing my chosen methods and how they work.

3.1 Initial Architecture

The main period of the proposed approach's work process is pictured in Figure. 3.1 flowchart shows that subsequent to bringing in the dataset[79] and parting it into preparing and testing sets, the work process is shown. Following that, the dataset goes through a preprocessing stage that incorporates information purging, tokenization, stemming, and word implanting. We tried three notable word inserting procedures and observed that the outfit in addition to the CNN classifier was the most reliable. The CNN design utilizes parallel grouping to recognize regardless of whether remarks are poisonous and afterward predicts subclass poisonous levels assuming that the remarks are destructive.

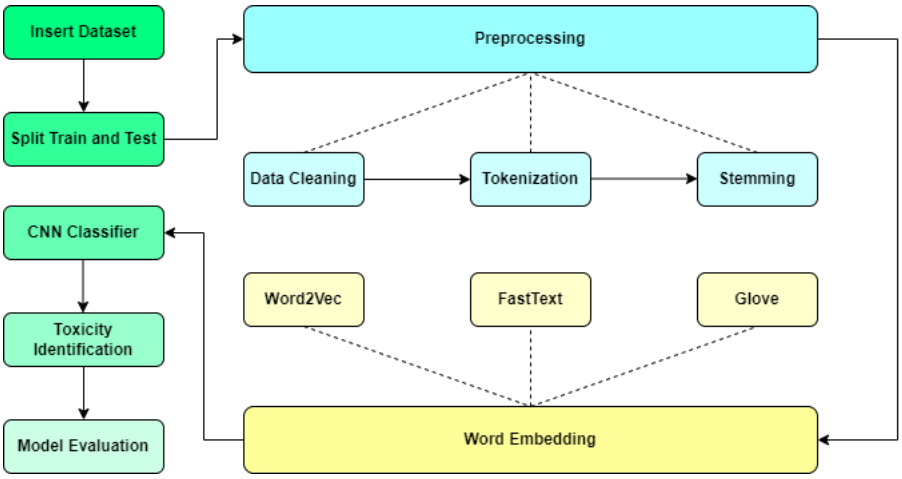


Figure 3.1: The first phase of the workflow

3.1.1 Data Preprocessing

Data Cleaning

Data cleaning is fundamental for producing better outcomes and quicker handling by eliminating abnormality from the dataset. Stop words are taken out, accentuations are taken out, all words are changed to bring down the case, copy words are taken out, URLs, emoticons, or shortcodes of emoticons are eliminated, numerals are taken out, one person based word is taken out, and images are taken out. Normal Language Tool stash (NLTK) is a python library for an assortment of dialects that we utilized in our model to further develop classification and precision.

Tokenization

It is the most common and vital part of NLP, where a sentence brimming with words is isolated or broken into individual words, every one of which is alluded to as a token. For making an interpretation of the word to a vector number, a model called FastText is used.

Stemming

Stemming is a method for planning words by erasing or bringing down the emphasis types of words like playing, played, and perky to find the root, otherwise called a lemma. There are additions, prefixes, tenses, sexual orientations, and other linguistic structures in these words. Moreover, assuming we inspect a gathering of words and observe a root that isn't of a similar sort, we think of it as a different class of that word, known as a lemma. For a better result, we apply the lemma approach in our model.

3.1.2 Word Embedding

The portrayal of a vector constructed utilizing brain networks is learned through word installation. It's for the most part used to control word vector portrayals in

a significant other option. Word implanting changes vector portrayals for mathematical words by making an interpretation of semantic information to an inserted space.

3.1.3 CNN Architecture for Classification

Considering its innate ability to utilize two factual characteristics known as 'local stationarity' and 'compositional structure,' Convolutional Neural Network, or CNN, has been broadly used to tackle picture order issues. The main rule is to execute CNN for harmful remark order, sentences should be encoded prior to being taken care of to CNN engineering. To work on the situation, the methodology of involving jargon in a media of record containing words, which has sets of texts planned into number lengths going from 0 to 1, was utilized. From that point forward, the cushioning approach is utilized to fill the archive network with zeros to accomplish the most extreme length, as CNN engineering requires contact input dimensionality. The subsequent stage is to change over the encoded reports into networks, with each column compared to a solitary term. The inserting layer, in which a thick vector changes any word (column) into a portrayal of low aspects, moves the frameworks created. Our CNN configuration is contained a 50 unit completely connected (thick) layer with a part size of five out of 128 channels for five-word embeddings. The inserting strategy utilizes fixed thick word vectors created with programs like FastText, word2vec, and GloVe, which were talked about in the past segment. We utilize the ADAM analyzer and twofold cross-entropy misfortune to prepare our model, which we assess with paired precision in the principal stage prior to continuing to multi-class arrangement for a dangerous evening out. We utilize four ages for high register power, skewering the preparation informational index into small clusters of 64 examples, with 70% of the information being utilized for preparing and 30% for testing.

3.1.4 Applying Classifiers

After getting the processed data set that is collected through the CNN Classifiers, we deploy this on five different classifiers. We have used Extreme Gradient Boosting (XGBoost) classifier, Random Forest classifier, Decision Trees classifier, AdaBoost classifier, Gradient Boosting classifier. We dropped the one-of-a-kind id esteem and took just the worth of six classes. Then, at that point, we split the informational index into train and test where 60% of the information is utilized for preparing and 40% for testing. Furthermore, we take the worth of an irregular state as 7. From that point forward, we change the informational collection into the test and train the informational index that is prepared to apply through classifiers. Figure 3.2 is showing the second period of our exploration work.

3.1.5 Generate Explainable AI

Artificial Intelligence (AI), a huge topic, has exploded in popularity in recent years. AI models have begun to surpass human intelligence at a rate no one could have imagined, as more and more complex models are released each year. Explainable AI refers to a set of approaches or strategies for explaining the decision-making process

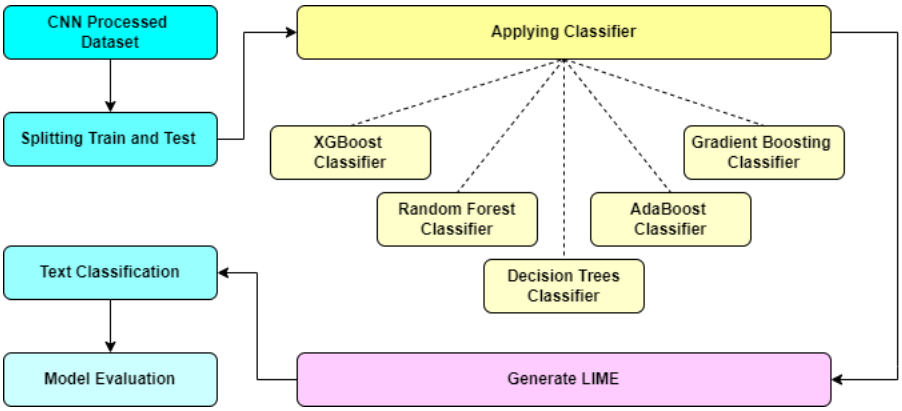


Figure 3.2: The second phase of the workflow

of a particular AI model. With more and more advanced strategies emerging each year, this relatively new branch of AI has shown immense promise. We used one Explainable AI (XAI) method for the result evaluation of these classifiers. We choose the LIME model for applying the XAI method. Local Interpretable Model-Agnostic Explanations (LIME) is an acronym for Local Interpretable Model-Agnostic Explanations [19]. We take the positive comment as well as a negative comment that was previously trained through all five classifiers. And apply those comments through the LIME model and get different results.

3.1.6 Appending Classifiers

In the last stage, we affix this large number of classifiers into one rundown and count the F1-weighted score for every classifier. Then, at that point, we utilize K-fold cross-validation where n split is 10 and random state is 12. Cross-validation is a resampling strategy for assessing AI models on a little example of information. That is, little example size is utilized to assess how the model would act overall when it is utilized to create forecasts on information that was not utilized during preparation. It's a well-known technique since it's not difficult to handle and creates a less slanted or hopeful gauge of model skill than different methodologies, like a basic train/test split.

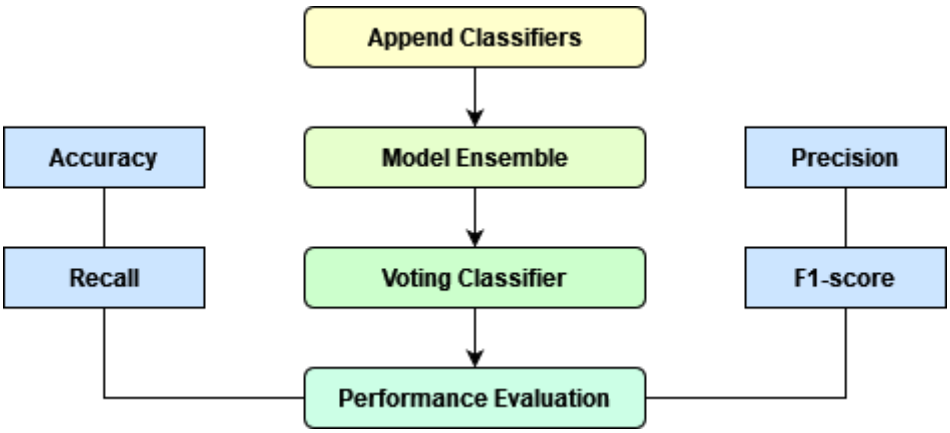


Figure 3.3: The final phase of the workflow

3.1.7 Ensemble Model

From that point onward, we present outfit realization which is displayed in Figure 3.3. Gathering learning is a far-reaching nonexclusive method for AI that joins the forecasts from various models to work on prescient execution. Group approaches are models that are made in products and afterward consolidated to come by better outcomes. Much of the time, gathering approaches give more exact outcomes than a solitary model. In various AI rivalries, the triumphant arrangements utilized troupe draws near. We utilized a Voting Classifier to outfit every one of the classifiers. A Voting Classifier is an AI model that gains from an outfit of many models and predicts a result (class) in view of the greatest likelihood of the result being the picked class. Rather than developing separate devoted models and deciding their precision, we make a solitary model that is prepared by various models and predicts yield in view of their total larger part of decisions in favor of each result class. We utilize delicate deciding in favor of our outfit model and get various measurements of exactness, review, f1 score, and accuracy.

3.2 Developed Architecture

Figure 3.4 shows the main process time for the proposed approach. This process shows the work after collecting the database and dividing it into a train and test. The dataset then goes through pre-processing steps, including data cleaning, tokenization, and stemming. After trying three important word-inserting methods, we found that the elements and ratings of CNN are very reliable. CNN uses parallel aggregation to determine the toxicity of observations and predict the toxicity of segmentation, which are destructive observations.

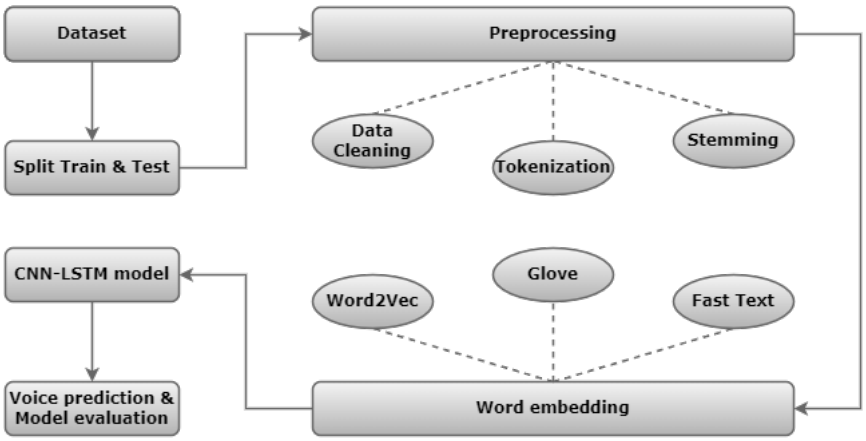


Figure 3.4: First Stage of the workflow

3.2.1 Data Preprocessing

Data cleaning is the basis for optimal results and fast management by eliminating deviations from the database. Remove pause words, highlight, replace all lowercase letters, repeat words, delete URLs, emojis, or short emoji codes, remove numbers, and photograph people talking. This is the most common and important part of

NLP, in which the whole sentence is divided into isolated or individual words, each of which is called a token. A template called FastText is used to define word vector numbers. Stemming is a method for planning words by erasing or bringing down the emphasis types of words like playing, played, and playful to find the root, otherwise called a lemma. There are additions, prefixes, tenses, sexual orientations, and other linguistic structures in these words.

3.2.2 Word Embedding

FastText is a word embedding method developed based on the word2vec model. Instead of directly studying word vectors, fastText calls each word an n-gram character. Helps to understand the meaning of short words and helps to insert to understand suffixes and prefixes. Once the word is identified by the letter n-gram, the grammar model is arranged for online learning. This model is a model of words with a sliding window at the top of the word because the internal structure of the word is not considered. While there are characters in this window, the n-gram table is not a bar. FastText works well with rare words. So even if a word is not found in the textbook, it can be divided into n-grams to get the results. Both Word2vec and GloVe cannot display words that are not in the sample dictionary. This is the biggest advantage of this method.

3.2.3 CNN with LSTM Architecture for Classification

The proposed architecture is a combination of a convolutional neural network (CNN) and a long-short-term memory (LSTM) network, briefly described below.

Convolutional Neural Network

A special type of multilayer perceptron is a CNN, but a simple neural network cannot learn complex features despite its deep learning structure. CNNs have proven effective in many applications such as image classification, object recognition, and clinical image analysis. The basic idea of CNN is to extract local features from the input layer of the upper layer and transform them into lower layers for complex features. CNNs consist of convolutional, pooling, and fully connected layers.

Long Short-Term Memory

Long short-term memory is the long-term evolution of recurrent neural networks (RNN). LSTM provides memory modules instead of regular RNN modules. Adds state to a cell and keeps the state for a long time. The main difference is RNN. LSTM networks can store historical information and combine it with received data. The LSTM is connected to three valves: intake, exhaust, and exhaust. x¬t represents the current input. Ct and Ct-1 represent the new and previous cell status, and ht and ht-1 represent the current and previous versions, respectively.

CNN-LSTM Network

In this hybrid model, CNN is the basic rule for malicious reference sequences, sentences must be encrypted before being watched by CNN Engineering. The embedding method uses standard dense word vectors created by programs such as FastText,

word2vec, and GloVe, which were discussed in the previous section. In our CNN configuration, there is a fully connected layer of 32 layers, which is 4 epochs for every 64-batch size for word embedding. In the convolutional layer, we used kernel size as 3 and used 'Relu' as activation. We use the 'Adam analyzer and 'binary cross entropy to prepare our model, which we evaluate using a previous parsing in the initial phase before performing a multi-class classification in a hazardous era. After the max pooling layer, it performs through the LSTM layer. And finally, we used 'SoftMax' as activation in the dense layer.

3.2.4 Supervised Machine Learning Algorithms

Figure 3.5 shows the next process for the proposed approach. Supervised learning allows you to collect data and generate data from previous experience. This helps to improve performance through experience.

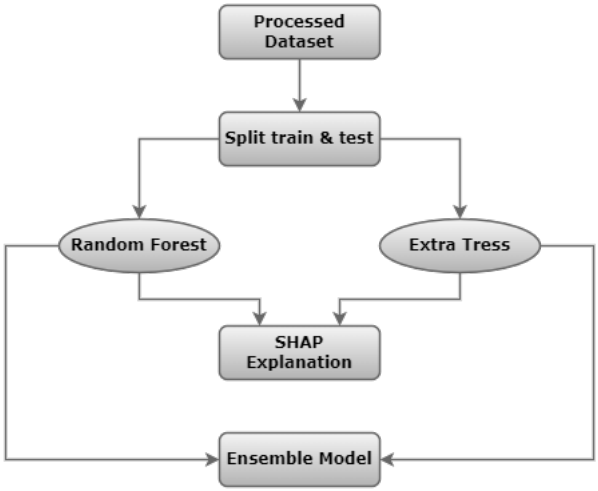


Figure 3.5: Second Stage of the workflow

Machine learning under inspection can help solve various computer problems in the real world. We used two supervised machine learning algorithms.

Random Forest

Random forest is a supervised machine learning algorithm widely used in classification and regression tasks. The decision tree is built on different models and averages the best votes in the case of classification and regression. One of the most important features of the Random Forest algorithm is that it can handle datasets with continuous variables. This provides better results for classification problems.

Extremely Randomized Trees

Extra Trees Classifier is an ensemble learning method that combines the results of several unrelated decision trees from the "forest" to obtain a classification result. Conceptually, a decision tree is very similar to a random forest classification, except that it is a forest structure. Additional Trees Each final forest tree consists of the original training sample. Each tree in each experimental node receives a random

sample of Q features from a set of features selected for the best diagnosis according to some mathematical criteria. This arbitrary function model produces many connected finite trees.

3.2.5 Explainable AI - SHAP

SHAP stands for SHapley Additive exPlanations. The purpose of this method is to calculate the provided/received forecast by calculating the contribution of each feature to the forecast. SHAP has many visualization tools for explaining the model, but we covered a few, each of which has its own specificities. We have shown a summary plot, decision plot, force plot, and waterfall plot for the random forest classifier and extra trees classifier.

3.2.6 Ensemble Model

Stack classification or generalization is a group method of machine learning. Uses machine learning algorithms to learn how best to combine predictions from two or more basic machine learning algorithms. The advantage of aggregation is that it utilizes models that work well in classification or regression problems and makes predictions that work better than any other model in the set. We used a stacking classifier to ensemble our model.

3.3 Final Architecture

In this section, we are discussing about dataset, data cleaning, CNN architecture, Bi-LSTM architecture, our proposed model, an architecture comparison with transformer model, some supervised machine learning algorithms, a details visualization of these machine learning models, machine learning model explanation with XAI, and finally an ensemble learning model is introduced in the following sections. We have taken eight machine learning models for explainability. We took XGBoost cllassifer, Random Forest classifier, Decision Trees classifier, AdaBoost classifier, Gradient Boosting classifier, Extra Trees classifier, Light GBM classifier, and CatBoost classifier. For explanation we took local interpretable model-agnostic explanations (LIME) and Shapely Additive Explanations (SHAP). For ensembling all the classifiers, we choose a stacking model. We used voting classifier for stacking all the machine learning classifiers. The figure 3.6 indicates the architecture of our proposed workflow. All components are explained in the following sections:

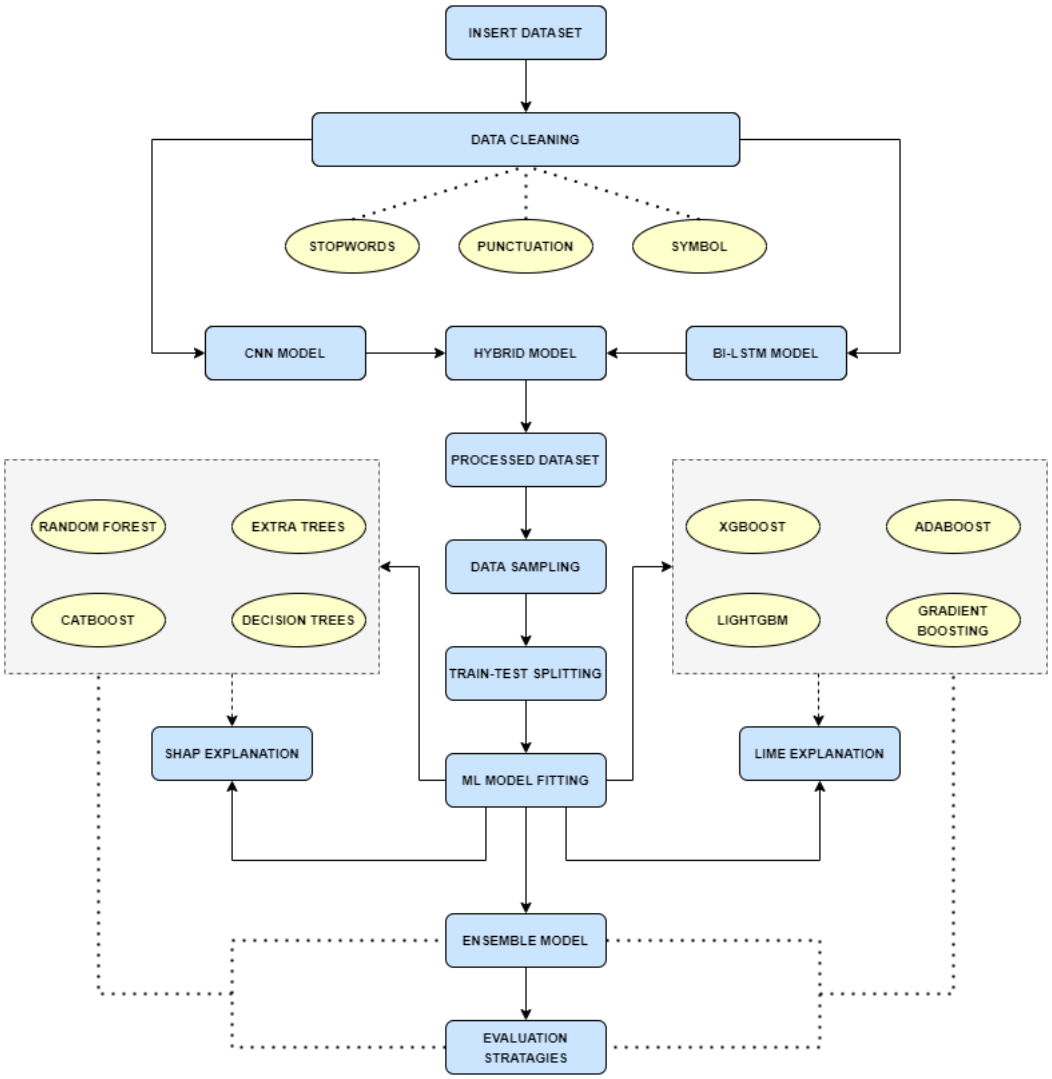


Figure 3.6: Top Level Overview of the Proposed Model

3.4 Dataset

I am using a dataset consisting of around 45,000 different types of Bengali comments. Initially, I collected the dataset from the research gate[76]. I have separated the comments for the multi-label classification from the existing annotation. There are five classes in that dataset. I collected some extra annotated data and inserted it into the existing dataset with an additional class called ”Slang”. So, there are a total of 7 columns in this dataset. The first is the comment column and the rest of the six is a label for a particular comment. Figure 2 shows the model training dataset. The dataset contains a total of six classes (Not Bully, Religious, Sexual, Threat, Troll, Slang). I ran this dataset in my code. Then I got results based on the training dataset.

Comments	Not_Bully	Religious	Sexual	Threat	Troll	Slang
জাতিয় ভাবে এদের নাস্তিক, মুখ চিনে রাখ।।মুরতাদ শয়তান।	0	1	0	0	0	0
তামাশার শেষ কই	1	0	0	0	0	0
সাপা তুমি বৎস্য ধরিয়া মৎস্য, বেচিয়া গঞ্জে, অকূল কুঞ্জে না পাইয়া তাজা,কেন ভক্ষণ করিয়াছ মেয়াদ উত্তীর্ণ গাজা!!!	0	0	0	0	1	0
নাম,জশ, খ্যাতি অর্জন করতে বছরের পর বছর সময় লাগে কিন্তু সেই জশ,খ্যাতি নষ্ট করতে ২ সেকেন্ড ও লাগেনা	0	0	0	0	1	0
অ মার্গি	0	0	1	0	0	1
১৯৪৩ সালের দুর্ভিক্ষে অনাহার ক্লিষ্ট হাড্ডিসার এক মার্গীর স্থির চিত্র	0	0	0	0	1	0
বেশ্যা, খানকি, চুক্কি মার্গি	0	0	1	0	0	1
এবার সুন্দর লাগছে আপু। অভিনন্দন	1	0	0	0	0	0
আলহামদুলিল্লাহ অসাধারণ	1	0	0	0	0	0
বাংলার জমিনে কোন নাস্তিকের জায়গা হবে না,.....	0	1	0	0	0	0
অসাধারণ ভাইজান এটা	1	0	0	0	0	0
ওকে বের করে দাও।এই দেশ থেকে নাস্তিক	0	1	0	0	0	0
পাছা দেখাইওনাতোমাকে দেখলে এমনিতেই হাত ঘাড়ি চালানা পারতা হ্যাঁয়	0	0	1	0	0	1
আল্লাহ তুমি এই মেয়ের কথা মাপ করার মালিক, আল্লাহ তুমি ওর মাপ করে দেয়ে হেদায়েত দান করো, আমিন	0	0	1	0	0	1
কোন শালা ভিডিও ও রেকোড বানিয়েছে সে শালা তো মনে হয় জারোক ছেলে তার মা তো পতিতার সরদার	0	0	0	0	1	0
ওরে দে লারা কনডম ছাড়া.!!নাস্তিক	0	1	0	0	0	0
তার বাড়ি কোথায়	1	0	0	0	0	0

Figure 3.7: A portion of Dataset

In Figure 3.7, you can see that all the comments are labeled with binary no 0 and 1. Where 0 represents no and 1 represents yes.

3.5 Data Cleaning

Data cleaning is a data processing technique used to transform raw knowledge into useful and effective controls. Empirical knowledge is often insufficient, inconsistent, not defined in specific procedures or programs, and may contain various errors. Data processing can be an undeniable technique to solve such problems[23]. Data preprocessing prepares the raw data for further processing. I have included several types of data processing steps. These include data cleaning, stop words removal, punctuation removal, symbol removal, missing data management, and more.

3.5.1 Stop words Removal

Stop words are words of any language that do not add much meaning to the sentence. You can safely remove them without damaging the meaning of the sentence. For some search engines, these are the most common short service words, such as at, which, etc. This can cause problems when searching for sentences with stop words. If we have a task of text classification or sentiment analysis, we must remove stop words because they do not provide any information to our model, that is, they do not introduce unnecessary words into our set, but if the task is language translation. We have to stop talking. effective, they must be translated into other words.
There are no hard and fast rules about when to remove stop words. But if you need to do something related to language classification, spam filtering, topic generation, automatic tagging, sentiment analysis, or text classification, I recommend removing the stop words. I have used a list of Bengali stop words that have been frequently used in Bengali literature. For example, 'হইবে', 'হৈলে', 'হইয়া', 'হচ্ছে', 'হত', 'হতে', 'হতেই', 'হবে', 'হবেন', 'হয়েছিল', 'হয়েছে', 'হয়েছেন', 'হয়ে', 'হয়নি', 'হয়', 'হয়েই', 'হয়তো', 'হল', 'হলে', 'হলেই', 'হলেও', 'হলো', 'হিসাবে', 'হওয়া', 'হওয়ার', 'হওয়ায়', 'হন', 'হোক'

3.5.2 Punctuation Removal

A second commonly used text processing method is to remove punctuation from text data. The process of removing punctuation makes all text look the same. For

example, data and words! After the deletion process, punctuation marks are treated in the same way.
You have to be careful with the text when you remove the punctuation marks because abstract words don't make sense if you remove them[3]. Depending on what you set in the parameter, it will change to "shouldn't" or "can't", like "don't". You should also be careful when choosing a list of punctuation marks to remove based on data usage. There are a total of 32 basic punctuation marks. The string module can be used directly with regular expressions to replace punctuation marks in text with an empty string. The 32 points that this road module gives us.

3.5.3 Symbol Removal

Sometimes, depending on your use case, the number may not contain important information from the text. Therefore, it is better to get rid of them than to keep them. For example, if you're doing sentiment analysis and the numbers don't represent your data, but if you're doing NER (Name Entity Recognition) or POS (Part of Speech tagging), use the number removal method carefully.
You can see the data, but it is better to remove the extra space because it will not save extra memory. With the increase in the number of users on social media platforms, the use of emojis in our daily life has increased significantly. Removing emoticons when parsing text is sometimes the right thing to do. Because sometimes it contains no information. If you do text analysis of Twitter and Instagram data, you will often see these emojis, but there are very few texts without them today.

3.6 Feature Extraction

As I am researching Bengali text classification, I need some features based on dataset. Therefore, the following section is a discussion based on the techniques used for feature extraction.

3.6.1 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 [77].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.[18]. 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-

$$x_1 : n = x_1 \oplus x_2 \oplus \cdots + x_n \tag{3.1}$$

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[18].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:

$$relU(MaxPool(Conv(M))) = MaxPool(reLu(Conv(M))) \tag{3.2}$$

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-

$$c = [c1, c2, c3, ...c_n - h + 1] \tag{3.3}$$

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. [56] 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-

$$xi : i + h - 1byci = f(wxi : i + h - 1 + b) \tag{3.4}$$

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.

$$\mathrm{c_i} = [\mathrm{c1}, \mathrm{c2}, \mathrm{c3} \ldots\ldots\ldots, \mathrm{c_n} - \mathrm{h} + 1] \tag{3.5}$$

with

$$c \in Rn - h + 1 \tag{3.6}$$

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 -

$$\mathrm{c'} = \max[\mathrm{c}] \mid \quad] \tag{3.7}$$

where ci is feature map

$$\mathrm{c_i} = [\mathrm{c1}, \mathrm{c2}, \mathrm{c3} \ldots\ldots\ldots, \mathrm{c_n} - \mathrm{h} + 1] \tag{3.8}$$

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

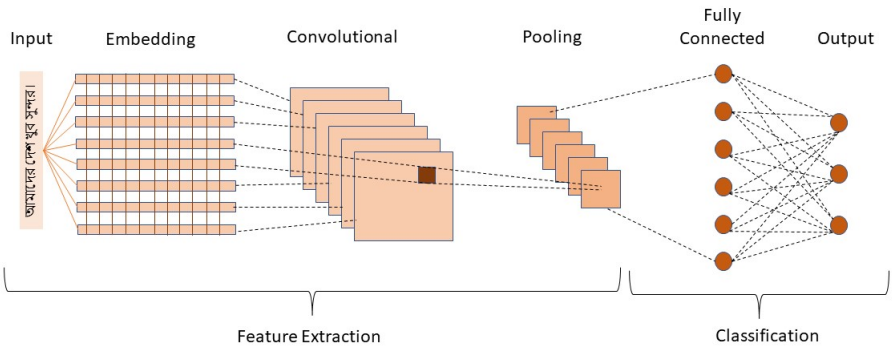


Figure 3.8: CNN Architecture

of this layer is the Soft max and Logistic layer. For binary classification, logistics is used, and Soft max is for multi-classification.Figure 3.8 shows the basic architecture of CNN model.

3.6.2 Text Classification using Bi-LSTM

Bi-LSTM is the process of constructing neural networks so that sequential information can be stored either backward (from future to past) or forward (from past to future).

In both directions, the input flows in both directions, which is different from a typical bidirectional LSTM. Generic LSTMs allow you to make the input flow one-way forward and backward. However, the two-way approach allows input to flow in both directions, preserving future and historical information. Let's take an example to explain better.

You cannot fill in the blanks in the sentence "Boys...". But in the future, if you have the phrase "the kids are coming home from school," it's easy to predict what you want to do with an empty space pattern from the past, and bidirectional LSTMs allow neural networks to do just that.

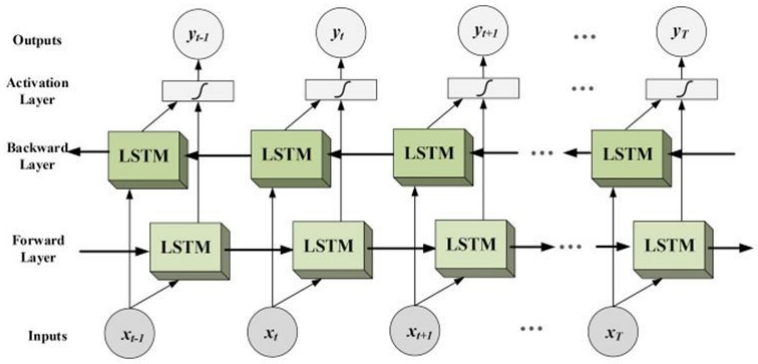


Figure 3.9: Bi-LSTM Architecture

In the diagram in figure 3.9, we can see the flow of information in the reverse and forward layers. BI-LSTMs are commonly used when sequencing operations are required. These networks can be used for text classification, speech recognition and predictive models.

3.7 Proposed CNN-BiLSTM based Hybrid Model

In this work, I employed a hybrid deep learning model with Convolutional Neural Network (CNN) and Bi directional Long Short-Term Memory (Bi-LSTM). Figure 3.6 shows the hybrid architecture that I made by CNN along with Bi-LSTM architecture.

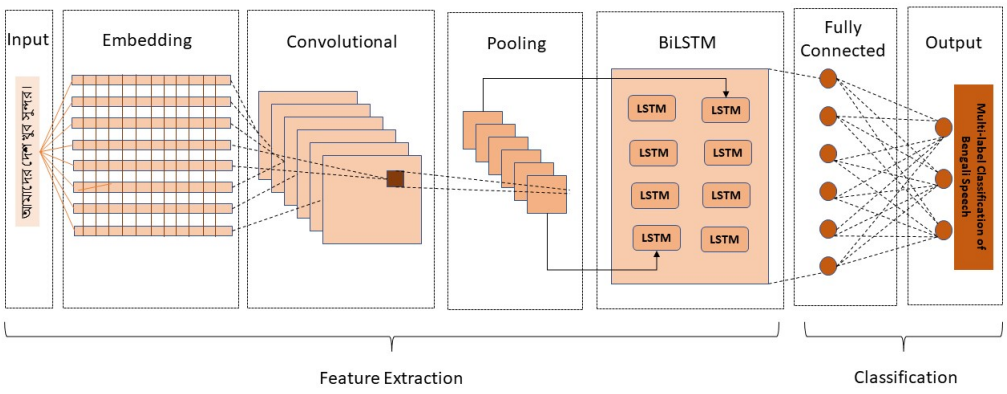


Figure 3.10: Proposed Architecture

3.7.1 Internal Architecture

At first, we defined a sequential model for our desired model. Then, we added an embedding layer from CNN architecture. We found the length of vocab size is 74438, the longest comment size is 200, the average comment size is 11.348549410698096,

the standard deviation of comment size is 15.254202939669483, and the max comment size is 57 from our dataset. We used pad sequence from keras. Here, we took maxlen = 57, padding = 'post', truncating = 'post'. In the embedding layer, we took embedding size = 74438+1, embedding dim = 600, input length = 57, trainable = true. After that, we added a convolutional layer from CNN architecture. As it is an 1D matrix, we took Conv1D where filter size = 128, kernel size = 3, padding = 'same', activation = 'relu'. Then, we added Maxpooling1D size = 3. Then, we added Batch Normalization. In the second step, we added bi -directional LSTM layer where the input gate size is 100. After that, we added fully connected layers. We added drop out layer and the value are 0.5. Finally, we added a dense layer where dense size is 6 because we have total 6 classes in our dataset. Here, we took 'sigmoid' as an activation. In the end, we summarized our model with summary function.Figure 3.11 shows the overall model architecture of our proposed model.

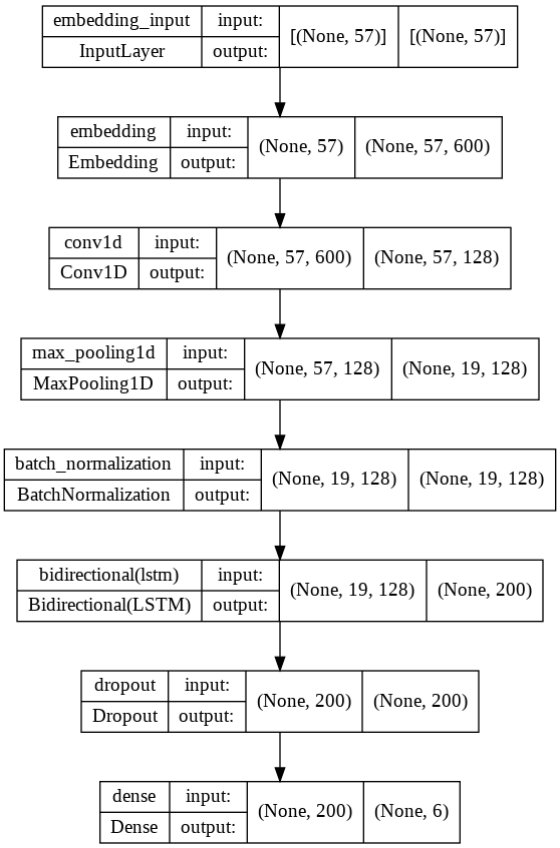
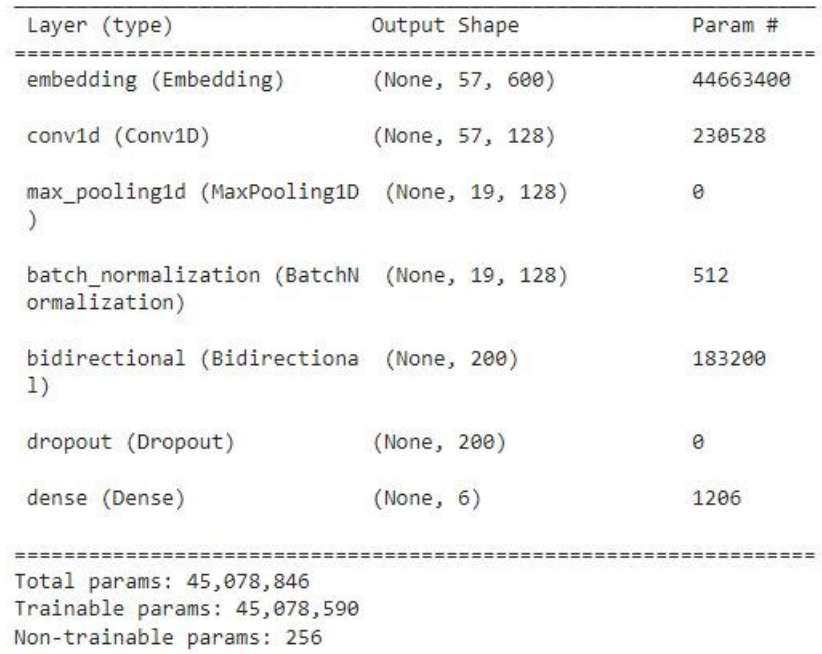


Figure 3.11: Internal Architecture

3.7.2 Model Compilation

We compiled our model where we took 'binary cross entropy' as a loss function, 'Adam' as an optimizer, metrics = 'accuracy'. Then, we fitted our model. We took a batch size of 32. We did total 100 epochs for fitting our model. We used early stopping function where in max mode, we monitored roc auc value of our model. Figure 3.12 shows the layer and output shape of our proposed model.
This model can predict any Bengali comment and classify them into six classes. It can detect whether the comment is Religious, or Sexual, or Threat, or Troll, or



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 57, 600)	44663400
conv1d (Conv1D)	(None, 57, 128)	230528
max_pooling1d (MaxPooling1D)	(None, 19, 128)	0
batch_normalization (BatchNormalization)	(None, 19, 128)	512
bidirectional (Bidirectional)	(None, 200)	183200
dropout (Dropout)	(None, 200)	0
dense (Dense)	(None, 6)	1206

Total params: 45,078,846
Trainable params: 45,078,590
Non-trainable params: 256

Figure 3.12: Layer and Output Shape

Slang, or Not Bully. And it gives a predicted percentage value of any comment according to these six classified sections. In our result part, we would show some output results of our model prediction.

3.7.3 Multi-label Classification

After fitting our model, we predicted the value of every comment in our dataset with prediction function. As we are performing a multi-label classification[34], we got total of 6 multi class values. Certain classification tasks require predicting more than one class label. This implies that classes are not mutually exclusive. These tasks are known as multi-label classification or multiple-label classification for short. In multi-label classification, zero or more labels are required as output for each input sample. The outputs are required to be mutually exclusive. The assumption is that the output labels are dependent on the input. After that, we made this to binary classification. We took the value of every class and got cross validated values for every comment. We marked 1 as a Bully comment and 0 as a Not Bully comment. And exported the dataset in data frame using panadas library.

3.8 Pre-Trained Transformer Based Models

A transformer model is a neural network that learns context and thus derives meaning by following relationships in sequence data (such as the words in this sentence). Transformer models increasingly use a mathematical technique called attention or self-awareness[68] to detect even subtle ways in which data elements are placed. Nowhere in the chain of influence and interdependence. Transfiguration translates text and speech in near real-time, opening up meetings and classes to many hearing-impaired participants. Transformers are the most advanced type of model available

today for sequence processing. Perhaps the most important application of these models is in word processing tasks, the most important of which is a machine translation. Indeed, transformers and their conceptual descendants have permeated every criterion of Natural Language Processing (NLP), from question-answering to grammatical correctness. In many ways, the transformer architecture is undergoing an evolution similar to the Convolutional Neural Network (CNN) we saw after the ImageNet competition in 2012, both for better and for worse. A transformer can be understood from its three components:

- The encoder encodes the input string into a state vector.
- The attention mechanism allows our transformer model to focus on finer aspects of sequential input currents. It is used several times by encoders and decoders to help them contextualize the input data.
- The decoder decodes the state representation vector to produce the target output sequence.

I tried multiple pre-trained BERT transformer models in our dataset to compare the performance evaluation with our proposed model. BERT stands for Bi-Directional Encoder Representation from Transformers. Figure 3.13 shows the BERT architecture for our dataset. It is intended to jointly adapt the left and right contexts to pre-train two-dimensional representations from unlabeled text. Thus, state-of-the-art models for a variety of NLP applications may be created by simply adding an extra output layer to a pre-trained BERT model.

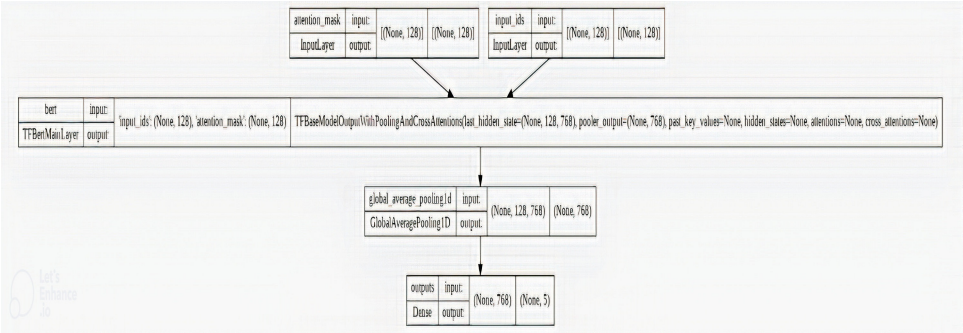


Figure 3.13: BERT Model

First off, BERT stands for the term "Bidirectional Encoder Representation from Transformers," which is easily understood. Each word in this sentence has a specific meaning, which we shall explore in this essay. At the moment, this family's key benefit is that BERT is built on the Transformer design.
Second, all of Wikipedia (which has 2,500 million words!) and the Book Corpus are included in the enormous untagged corpus on which BERT is already trained (800 million words). The pre-training stage is half of BERT's success. In fact, as we train a model on a huge corpus, our model starts to comprehend the language more and more deeply.
Third, the BERT model is "deeply two-dimensional." Bidirectional refers to the fact that during the training phase, BERT gathers knowledge from both the left and right sides of the token context.

3.9 Data Sampling

After that, we exported the predicted processed dataset. And we applied it through different machine learning algorithms. In machine learning, class imbalance is a frequent issue, particularly in classification issues[42]. The accuracy of the model can be greatly impacted by unbalanced data. The majority of machine learning algorithms work best when there are nearly equal numbers of examples in each class. This is so because accuracy maximisation and error minimization are the two main considerations while designing algorithms. In contrast, if the data set is unbalanced, you may predict the majority class with a decent amount of confidence. The dominant class will be captured, but the minority class—which is frequently the objective of the model—will not be captured. Resampling is a method frequently used to handle datasets that are severely imbalanced. Removing samples from the majority class is necessary.
Before applying, we performed dataset resampling. We used SMOTE technique for data resampling. The synthetic minority over-sampling technique, or SMOTE for short, is a preprocessing technique used to address a class imbalance in a dataset. In the real world, we often find ourselves trying to train a model on a dataset with few examples of a given class[47], which results in poor performance. Due to the sensitive nature of the data, occurrences are so uncommon I want to talk to you about something serious., it's not always realistic to go out and acquire more. One way to resolve this issue is to under-sample the majority class. We would exclude rows corresponding to the majority class such that there is roughly the same number of rows for both the majority and minority classes. However, in doing so, we would lose out on a lot of data that could be used to train our model and improve its accuracy. Another alternative is to oversimple the minority class. We artificially increase the number of observations from the minority class by randomly duplicating observations. The issue with this approach is that it leads to over-fitting because the model learns from the same examples. This is where SMOTE comes into play. The SMOTE algorithm can be described as follows:

- Take the difference between the sample and its nearest neighbor.
- A random integer between 0 and 1 should be multiplied by the difference between the two values.
- You can add this difference to the sample to generate a new synthetic example in the feature space.
- Continue with the next nearest neighbor up to a user-defined number.

After performing SMOTE technique, we got a balanced dataset. Previously, original dataset shape Counter is '1': 28581, '0': 15539. Now, resample dataset shape Counter is '1': 28581, '0': 28581

3.10 Applying Machine Learning Algorithms

After data balancing, we split the processed dataset into training and testing. We took a test size of 0.4 and random state = 7. We got a processed dataset which indicates binary classification. Figure 3.14 shows the processed dataset that we got.

Comments	Religious	Sexual	Threat	Troll	Slang	Status
কুত্তার মত চেহারা	1.994862e-11	1.413871e-10	3.631158e-10	1.000000e+00	1.729250e-14	1
ওহে সাফা কবির পরকাল বিশ্বাস করো নাতোমার কথা যা...	1.000000e+00	1.876250e-10	2.226771e-11	6.885733e-10	2.602646e-13	1
শালী নাস্তিকের বাচ্চা	1.000000e+00	9.776600e-10	2.656588e-12	6.534697e-09	2.990229e-12	1
আক্কেলেই আখখেতেই কোনোটিই ব্যাপক অর্থবহ উপমা ব...	2.421095e-12	2.792055e-10	1.746222e-12	5.306298e-10	9.178857e-17	0
হিরু আলম চিনে পাগল হিসেবে জায়েদ খান সবাই চিনে...	8.302291e-11	3.198568e-10	9.846891e-10	1.000000e+00	1.647999e-14	1

Figure 3.14: Processed Dataset

We have applied XGBoost Classifier, Random Forest Classifier, Decision Trees Classifier, AdaBoost Classifier, Gradient Boosting Classifier, Extra Trees Classifier, Light GBM Classifier and Cat Boost Classifier along with an estimator named logistic regression through this processed dataset. Now, we are discussing about these machine learning models.

3.10.1 XGBoost Classifier

A scalable and distributed Gradient Boosted Decision Tree (GBDT) machine learning package is called XGBoost, or Extreme Gradient Boosting. It uses tree alignment and is the leading machine learning library for regression, classification and ranking problems. To understand XGBoost[36], it is important to first understand the machine learning concepts and algorithms that make up XGBoost: supervised machine learning, decision trees, ensemble learning, and gradient boosting. Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features. In the classifier, we considered max depth=2, random state=1, n estimators=100 as parameters.

3.10.2 Random Forest Classifier

A popular supervised machine learning technique for classification and regression applications is random forest[40]. It creates decision trees based on several models and, in the case of regression, relies on a majority vote for categorization and averaging. The Random Forest algorithm's ability to handle datasets with continuous variables is one of its key characteristics. Better outcomes for categorization issues are produced by doing this. We used the parameters max depth = 2, random state = 1, and n estimators = 100 in the classifier.

3.10.3 Decision Trees Classifier

A decision tree is a supervised learning method that may be used to classification and regression issues, while classification issues are frequently favoured[19]. It is a tree-like classifier in which core nodes stand in for dataset properties, branches for decision rules, and leaf nodes for individual decisions. A decision node and a termination node are the two nodes in a decision tree. While leaf nodes are the outcomes of those decisions and have no other branches, decision nodes are utilised to make decisions and have numerous branches. Based on the qualities of a particular collection of data, a conclusion or experiment is drawn. We used the parameters max depth = 2 and random state = 1 in the classifier.

3.10.4 AdaBoost Classifier

The AdaBoost algorithm, also known as adaptive boosting, is a boosting approach used in machine learning as an ensemble method[54]. This is called adaptive scaling because weights are reassigned to each sample and misclassified samples have higher weights. Boosting is used to reduce the bias and variance of supervised learning. It works on the principle that students develop constantly. With the exception of the first, each subsequent student develops from the previous one. Simply put, weak students become strong. The AdaBoost algorithm works on the same principle as Boost, with a slight difference. In the classifier, we considered random state=1, n estimators=100 as parameters.

3.10.5 Gradient Boosting Classifier

Gradient boosting is a type of ensemble method where you build several weak models (often decision trees) and combine them to get better overall performance[35]. In gradient boosting, each predictor tries to improve its previous version by minimizing errors. But the interesting idea behind gradient boosting is that instead of fitting a predictor to the data at each iteration, it actually fits a new predictor to the residual errors of the previous predictor. In the classifier, we considered max depth=2, random state=1, n estimators=100 as parameters.

3.10.6 Extra Trees Classifier

An ensemble learning approach known as a highly randomised tree classifier (also known as a redundant tree classifier) combines the classification outcomes of several unconnected decision trees gathered in a "forest" to obtain a final result[33]. It differs mainly in how the decision trees in the forest are built, conceptually speaking, from a random forest classifier. further trees The first training sample is used to build each decision tree in the forest. The decision tree must then decide which feature to use to partition the data according to some mathematical criterion after receiving a random sample of k features from the feature set at each test node (usually the Gini index). This arbitrary feature selection results in many associated decision trees. In the classifier, we considered max depth=2, random state=1, n estimators=100 as parameters.

3.10.7 Light GBM Classifier

LightGBM is an ensemble gradient boosting method implemented with the Auto ML tool in training and based on decision trees[30]. LightGBM uses a histogram-based method where the data is grouped using a distribution histogram. Instead of each data point, containers are used to copy data, calculate percentages and distribute. This method can also be extended to spatial datasets. Another feature of LightGBM is its proprietary function, in which the algorithm combines proprietary functions to reduce dimensionality, making it faster and more efficient. In the classifier, we considered max depth=2, random state=1, n estimators=100 as parameters.

3.10.8 Cat Boost Classifier

Gradient-based decision trees are the foundation of CatBoost. A succession of decision trees are built one after the other during training[75]. Compared to earlier trees, each tree is formed with less loss. The output parameters determine how many trees are planted. Use an additional detector to avoid matches. If provoked, the trees will refuse to build. The CatBoost algorithm is a powerful and greedy new gradient gain implementation. This feature allows CatBoost to learn faster and make predictions 13-16 times faster than other algorithms. In the classifier, we considered max depth=2, random state=1, n estimators=100 as parameters.

3.10.9 Logistic Regression

Logistic regression is a model for predictive modeling of binary classification[31]. The parameters of a logistic regression model can be estimated using a probability system called maximum likelihood estimation. According to this framework, a probability distribution (class label) for the target variable must be assumed, and then a probability function must be defined that calculates the probability of observing the result given the input data and the model. This function can be optimized to find the parameter set that gives the largest sum in the training dataset.

3.11 Model Explanation with XAI

A collection of procedures and methods known as Explainable Artificial Intelligence (XAI) enables consumers to comprehend and believe the findings and conclusions reached by machine learning algorithms. In descriptive AI[53], the AI model, its anticipated effects, and any potential biases are all described. It helps model accuracy, fairness, transparency and outcomes in AI-based decision-making. Interpretable AI is critical for an organization to build trust and confidence when building AI models. Understanding AI allows an organization to approach AI development responsibly.

As artificial intelligence becomes more sophisticated, humans are challenged to understand and rethink how algorithms produce results. The entire computer process is often referred to as a ”black box” that is almost impossible to interpret. These black box models are generated directly from the data. Furthermore, even the engineers or data scientists who create the algorithm cannot understand or explain exactly what is happening inside them or how the AI algorithm arrived at a certain result. We have used LIME and SHAP both XAI methods for a model explanation.

3.11.1 Locally Interpreted Model-agnostic Explanations

LIME, or locally interpreted model-agnostic explanations, is a method that reliably interprets the predictions of any classifier or regression by approximating them locally with an explanatory model. The model changes the data by changing the attribute values and observes the effect on the result[57]. It plays the role of an ”explainer” to explain the predictions of each data model. The LIME output is a set of descriptors representing the contribution of each feature to the model prediction,

which is a form of the local descriptor. In the result section, the detailed LIME explanation is shown for every machine learning algorithm.

3.11.2 SHapley Additive exPlanations

SHAP (Shapley Additive Explanations) is a method for interpreting individual predictions based on the optimal Shapley value in game theory[63]. The Shapley value is a widely used method in cooperative game theory and has desirable properties. Data instance eigenvalues act as federation members. The Shapley value is the average minimum contribution of the eigenvalues across all possible coalitions. In the result section, the detailed SHAP explanation is shown for every machine learning algorithm.

3.12 Ensemble Model

Ensemble modeling is the process of running two or more related but separate analytical models and then combining the results into a single estimate or extension to improve the accuracy of predictive analytic and data mining applications. In predictive modeling and other types of data analysis, individual models based on data models may suffer from biases, high variability, or inaccuracies that affect the reliability of the analysis results[65]. Similar disadvantages can occur when special modeling methods are used. By combining different models or analyzing multiple models, data scientists and other data analysts can reduce the impact of these limitations and better inform business decision makers. Integrated modeling is gaining popularity as more organizations use the computing resources and advanced analytic software required to run these models. In addition, advances in Hadoop and other big data technologies have enabled companies to store and analyze large amounts of data, giving them more opportunities to run analytical models on different types of data.

We used voting classifier for ensemble all the machine learning classifiers with an estimator[55]. We chose voting = 'soft' in the voting classifier and cross-validated all the models where we took k fold as cross-validator.A voting classifier is a machine learning model that predicts an output (class) based on the likelihood that the class will be chosen as the output. It is trained on an ensemble of models. The output class is simply predicted based on the voting classifier's largest majority of votes after the results of each classifier that was provided to it have been aggregated. The goal is to create a model that is trained using those models and predicts the outcome based on their majority votes for each output class rather than creating individual-specific models and determining each model's correctness. The voting classifier accepts two different voting methods.

- **Hard Voting:** The class with the biggest majority, or the class with the highest probability predicted by each classifier, is the projected exit class in hard voting. If three classifiers are used, the majority of them will predict A as the output in this case (A, A, B). A will thus be the last forecast.
- **Soft voting:** The output class is a forecast based on the typical value of that class's probability. The three models' projected probabilities for classes A =

(0.30, 0.47, 0.53) and B = (0.30, 0.47, 0.53) are based on certain input data (0.20, 0.32, 0.40). Class A is the winner because it has the highest average probability, with a mean of 0.4333 compared to class B's average of 0.3067.

Finally, we measured performance metrics and evaluated the detailed results of this ensemble learning model.

Chapter 4

Result and Analysis

In this section, the results of our proposed models and analyzes will be discussed. Here, the resulting segment will contain the general implementation of the proposed model with a presentation boundary. The analysis will include comparing our results and evaluating our results, as well as an in-depth discussion of our overall performance and model.

4.1 Dataset Visualization

In our dataset, there are almost 44,120 rows and 7 columns are specified. First row contains the title of each column. Rest of the rows indicate different comments along with labeling. First column is 'Comments' column. Rest of the 6 columns are 'Not Bully', 'Religious', 'Sexual', 'Threat', 'Troll', and 'Slang'. Figure 4.1 shows the Comment length of the overall dataset.

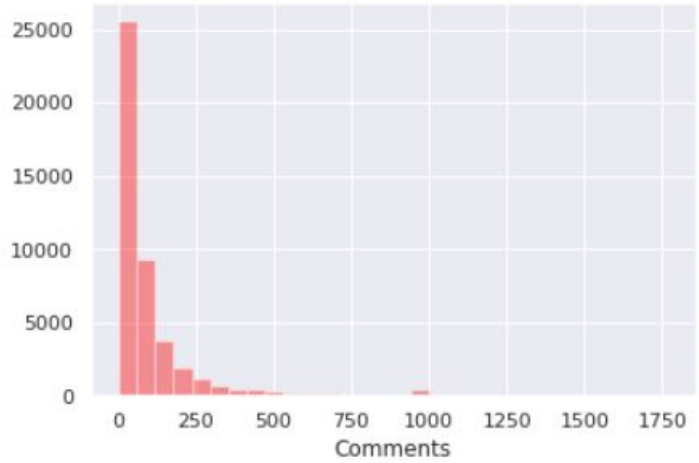


Figure 4.1: Length of Comments

As we are doing multi-label classification, each class is identical with binary value of 0 and 1. Where 1 depicts that the comment belongs to particular of that class and 0 depicts that the comment does not belong to particular of that class.

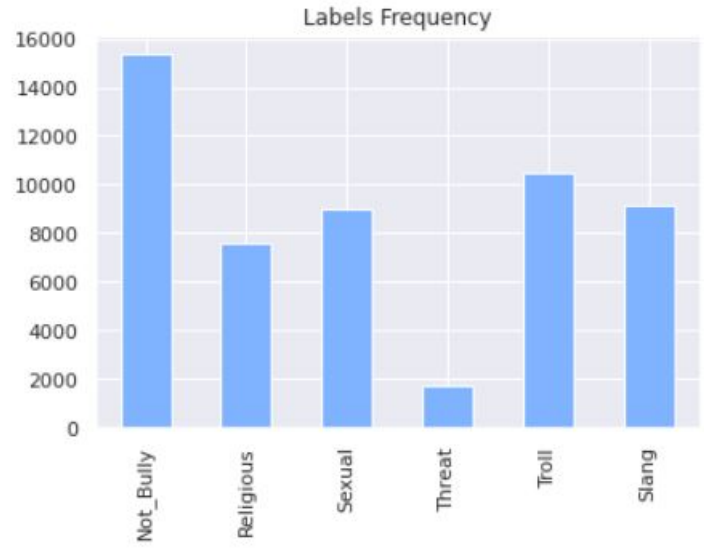


Figure 4.2: Frequency of Labels

Figure 4.2 shows label frequency of each class where the label value is 1. On the other side, figure 4.3 shows both label frequency of each class where the label value is 1 and 0.

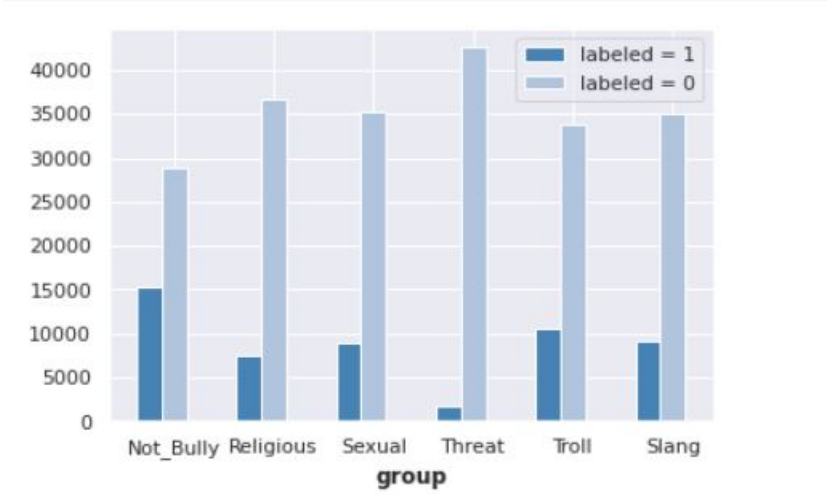


Figure 4.3: Frequency of both Labels

The correlation coefficients between variables are displayed in a table called a correlation matrix. The association between two variables is displayed in each cell of the table. Data are compiled using the correlation matrix, which is also utilised as an input for advanced analysis and as a diagnostic for that analysis. Figure 4.4 shows that 'Sexual', comments are most strongly correlated with 'Threat' and 'Religious' class. Moreover 'Not Bully' and all other classes have weak correlation. It also shows the class 'Slang' has the weakest correlation with all classes.

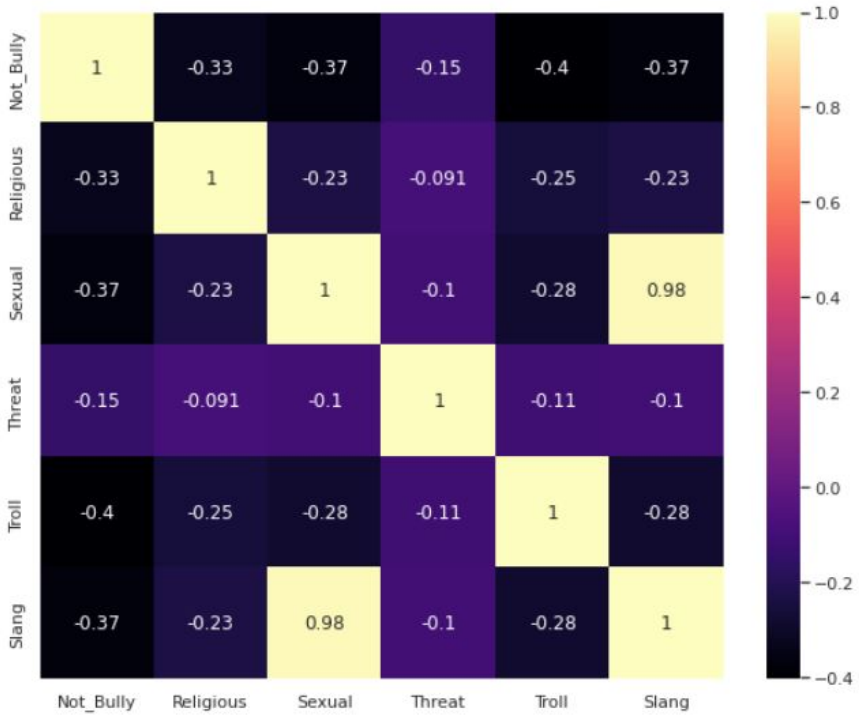


Figure 4.4: Visual representation of correlation between classes

There are lots of comments containing in our data set. These comments contain some common religious words also. Because of this re occurrence of words, we have tried to show them in Word Cloud.

Figure 4.5: Bangla Word cloud

These Word Cloud images will show those common Bengali religious words that are using in day-to-day life. Figure 4.5 is an example of Word Cloud of our Bengali dataset.

4.2 Model Tuning

Visualizing the performance of any machine learning model is an easy way to understand the data generated by the model and make informed decisions about pa-

rameters that affect the machine learning model or changes that need to be made to higher parameters.

Figure 4.6: Model Loss

The original data set is divided so that 40% of the total data is designated as the test set and the rest is left as the training set. The trainer is redistributed so that 40% of the trainer is assigned as the validation set and the rest is used for training purposes.

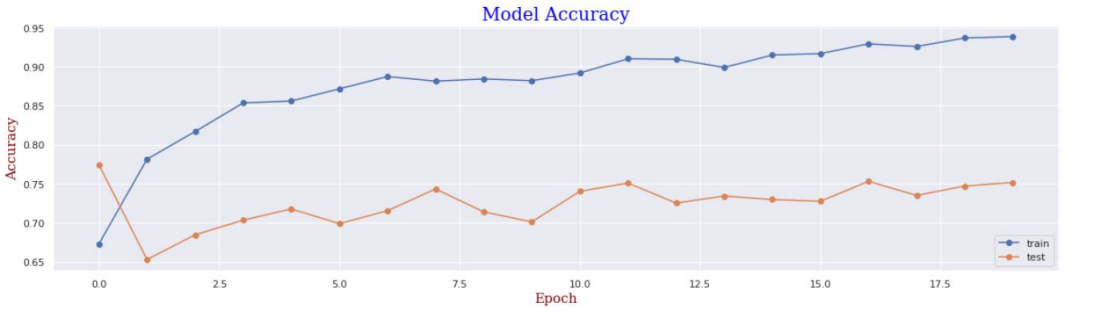


Figure 4.7: Model Accuracy

Of the total dataset, 60% is considered as the training set and 40% as the validation set. Model accuracy and loss data for each period is stored in the history object. Figure 4.6 plots the graph of the training loss vs. validation loss over the number of 25 epochs. On the other hand, Figure 4.7 plots the graph of the training accuracy vs. validation accuracy over the number of 25 epochs.

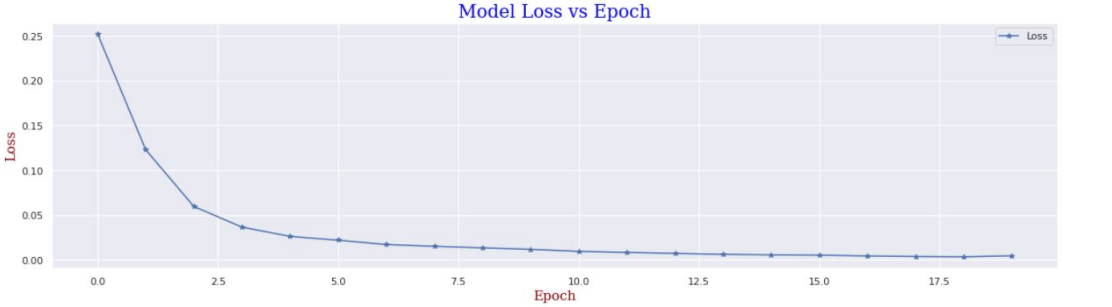


Figure 4.8: Model Loss vs Epoch

As we set 25 epochs on our model fitting, figure 4.8 shows the graph of the training loss over the number of 25 epochs.

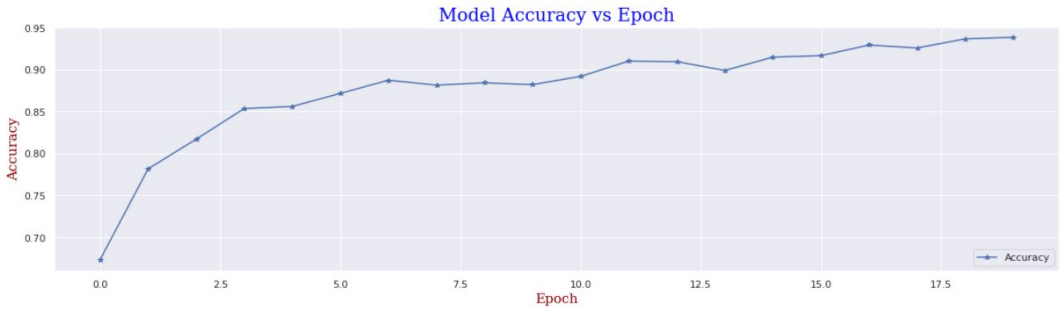


Figure 4.9: Model Accuracy vs Epoch

Here again, figure 4.9 shows the graph of the training accuracy over the number of 25 epochs.

4.3 Comparison with Transformer Models

As we discussed in the methodology section that we have used our dataset on multiple pre-trained transformer models[66]. We have used a couple of same sentences to predict the result. Here, we stacked all the model results along with our hybrid deep learning model to see the comparison of the prediction. Table 4.1 depicts the prediction of the neural space reverie Indic transformers bn RoBERTa pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	1%	93%	1%	3%	1%	0%
ডানা কাটা পরি	1%	93%	1%	3%	1%	0%
জবাই কইরা দিবো শালারে	0%	0%	4%	1%	95%	0%
শালী নাস্তিকের বাচ্চা	1%	99%	1%	1%	1%	0%
শালির ফাসি চাই	7%	4%	2%	92%	3%	0%
আমাদের দেশ সুন্দর	99%	0%	0%	0%	1%	0%
দিচ ইচ চুদা কবির	1%	1%	97%	1%	3%	22%
আপনার ডাক নাম নাকি প্রিয়া	45%	1%	1%	0%	43%	0%

Table 4.1: Neural Space RoBERTa

This is a RoBERTa language model pre-trained on a single language training set of 6 GB. The pre-training data is mostly taken from OSCAR. This model can be configured for various low-level tasks such as text classification, POS tagging, question answering, etc. Embedding this model can also be used for feature-based learning. Table 4.2 depicts the prediction of the neural space reverie Indic transformers bn BERT pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	3%	47%	1%	1%	59%	0%
ডানা কাটা পরি	40%	0%	16%	1%	40%	0%
জবাই কইরা দিবো শালারে	4%	3%	2%	93%	8%	0%
শালী নাস্তিকের বাচ্চা	3%	70%	10%	1%	15%	0%
শালির ফাসি চাই	2%	12%	1%	89%	6%	0%
আমাদের দেশ সুন্দর	95%	3%	1%	3%	1%	0%
দিচ ইচ চুদা কবির	7%	2%	58%	0%	30%	16%
আপনার ডাক নাম নাকি প্রিয়া	9%	27%	5%	0%	14%	3%

Table 4.2: Neural Space BERT

This is a BERT language model pre-trained on a single language training set of 3 GB. The pre-training data is mostly taken from OSCAR. This model can be configured for various low-level tasks such as text classification, POS tagging, question answering, etc. Embedding this model can also be used for feature-based learning. Table 4.3 depicts the prediction of the monsoon NLP Bangla ELECTRA pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	4%	62%	16%	7%	6%	0%
ডানা কাটা পরি	8%	14%	8%	6%	63%	0%
জবাই কইরা দিবো শালারে	13%	7%	11%	10%	80%	0%
শালী নাস্তিকের বাচ্চা	8%	83%	8%	11%	8%	3%
শালির ফাসি চাই	4%	9%	19%	8%	53%	4%
আমাদের দেশ সুন্দর	92%	6%	6%	4%	7%	0%
দিচ ইচ চুদা কবির	12%	7%	81%	10%	15%	31%
আপনার ডাক নাম নাকি প্রিয়া	25%	4%	9%	4%	67%	0%

Table 4.3: Bangla ELECTRA

This is the second trial of a Bengali/Bengali language model trained by Google Research's ELECTRA. The pre-training data is mostly taken from OSCAR. This model can be configured for various low-level tasks such as text classification, POS tagging, question answering, etc. Embedding this model can also be used for feature-based learning. Table 4.4 depicts the prediction of the Sagor Sarkar Bangla BERT Base pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	1%	98%	1%	0%	1%	0%
ডানা কাটা পরি	35%	0%	9%	1%	43%	0%
জবাই কইরা দিবো শালারে	4%	2%	1%	95%	2%	0%
শালী নাস্তিকের বাচ্চা	1%	99%	1%	1%	1%	0%
শালির ফাসি চাই	3%	2%	2%	94%	2%	4%
আমাদের দেশ সুন্দর	99%	0%	1%	1%	1%	0%
দিচ ইচ চুদা কবির	0%	1%	97%	0%	2%	1%
আপনার ডাক নাম নাকি প্রিয়া	58%	0%	2%	0%	45%	0%

Table 4.4: Bangla BERT base

The Sagor Sarkar Bangla Part Base Model is a natural language processing (NLP) model implemented in the Transformer library, typically using the Python programming language. Bangla-BERT-BASE is a pre-trained Bengali language model that uses the masked language model described in BERT. Table 4.5 depicts the prediction of the Indic BERT pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	1%	6%	6%	26%	65%	0%
ডানা কাটা পরি	25%	0%	8%	1%	61%	0%
জবাই কইরা দিবো শালারে	5%	0%	60%	1%	34%	0%
শালী নাস্তিকের বাচ্চা	2%	4%	1%	96%	2%	0%
শালির ফাসি চাই	1%	1%	16%	4%	74%	0%
আমাদের দেশ সুন্দর	98%	1%	1%	1%	1%	0%
দিচ ইচ চুদা কবির	5%	0%	88%	1%	8%	25%
আপনার ডাক নাম নাকি প্রিয়া	68%	1%	3%	2%	26%	0%

Table 4.5: Indic BERT

Assamese, Bengali, English, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, and Telugu are just a few of the main Indian languages that the multilingual Albert model, known as Indic BERT, has been taught on. Despite having a lot less parameters than other generic models like mBERT and XLM-R, Indic BERT can handle many jobs better. The prediction made by the BERT multilingual base pre-trained model is shown in Table 4.6.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	2%	71%	7%	22%	5%	0%
ডানা কাটা পরি	4%	1%	6%	2%	92%	0%
জবাই কইরা দিবো শালারে	10%	6%	4%	85%	12%	0%
শালী নাস্তিকের বাচ্চা	8%	6%	7%	82%	1%	0%
শালির ফাসি চাই	13%	9%	5%	89%	8%	0%
আমাদের দেশ সুন্দর	97%	2%	1%	1%	2%	0%
দিচ ইচ চুদা কবির	12%	20%	29%	0%	20%	2%
আপনার ডাক নাম নাকি প্রিয়া	12%	12%	14%	1%	59%	0%

Table 4.6: BERT multilingual base

Pre-trained model on the top 102 languages with largest Wikipedia using Masked Language Modeling (MLM) target. BERT is a transform model that is pre-trained on a large set of multilingual data in a self-supervised manner. This means that it only pre-trains on the source texts and does not label them in any way (which is why it can use a lot of public data) through an automatic process. Create records and tags from these texts. Table 4.7 depicts the prediction of the neural space reverie Indic transformers bn DistilBERT pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	4%	90%	3%	3%	4%	0%
ডানা কাটা পরি	30%	0%	4%	0%	66%	0%
জবাই কইরা দিবো শালারে	2%	2%	3%	81%	18%	0%
শালী নাস্তিকের বাচ্চা	2%	75%	12%	81%	8%	0%
শালির ফাসি চাই	7%	3%	5%	89%	5%	0%
আমাদের দেশ সুন্দর	96%	2%	1%	2%	2%	0%
দিচ ইচ চুদা কবির	15%	4%	56%	2%	37%	5%
আপনার ডাক নাম নাকি প্রিয়া	26%	3%	3%	0%	55%	0%

Table 4.7: Neural Space DistilBERT

This is a DistilBERT language model pre-trained on a single language training set of 6 GB. The pre-training data is mostly taken from OSCAR. This model can be configured for various low-level tasks such as text classification, POS tagging, question answering, etc. Embedding this model can also be used for feature-based learning. Table 4.8 depicts the prediction of the neural space reverie Indic transformers bn XLM-RoBERTa pre-trained model.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	0%	97%	0%	1%	1%	0%
ডানা কাটা পরি	2%	0%	95%	1%	2%	0%
জবাই কইরা দিবো শালারে	7%	2%	1%	94%	1%	0%
শালী নাস্তিকের বাচ্চা	1%	99%	1%	1%	1%	0%
শালির ফাসি চাই	6%	2%	1%	93%	1%	0%
আমাদের দেশ সুন্দর	99%	1%	0%	1%	1%	0%
দিচ ইচ চুদা কবির	0%	1%	96%	0%	4%	29%
আপনার ডাক নাম নাকি প্রিয়া	25%	0%	1%	0%	75%	0%

Table 4.8: Neural Space XLM-RoBERTa

This is an XLM-RoBERTa language model pre-trained on a single language training set of 6 GB. The pre-training data is mostly taken from OSCAR. This model can be configured for various low-level tasks such as text classification, POS tagging, question answering, etc. Embedding this model can also be used for feature-based learning.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	0%	100%	0%	0%	0%	22%
ডানা কাটা পরি	72%	0%	0%	0%	54%	0%
জবাই কইরা দিবো শালারে	0%	0%	0%	100%	0%	0%
শালী নাস্তিকের বাচ্চা	0%	100%	0%	0%	0%	0%
শালির ফাসি চাই	0%	0%	0%	100%	0%	0%
আমাদের দেশ সুন্দর	100%	0%	0%	0%	0%	0%
দিচ ইচ চুদা কবির	0%	0%	61%	0%	93%	51%
আপনার ডাক নাম নাকি প্রিয়া	100%	0%	1%	0%	0%	0%

Table 4.9: Hybrid Deep Learning

Table 4.9 indicates the prediction of our hybrid deep learning model. Here, we can see that this model can precisely predict the value of each sentence.

Comments	Not Bully	Religious	Sexual	Threat	Troll	Slang
সব মালাউন খারাপ	NO	YES	NO	NO	NO	NO
ডানা কাটা পরি	YES	NO	NO	NO	YES	NO
জবাই কইরা দিবো শালারে	NO	NO	NO	YES	NO	NO
শালী নাস্তিকের বাচ্চা	NO	YES	NO	NO	NO	NO
শালির ফাসি চাই	NO	NO	NO	YES	NO	NO
আমাদের দেশ সুন্দর	YES	NO	NO	NO	NO	NO
দিচ ইচ চুদা কবির	NO	NO	YES	NO	YES	YES
আপনার ডাক নাম নাকি প্রিয়া	YES	NO	NO	NO	NO	NO

Table 4.10: Hybrid Deep Learning (Binary)

Table 4.10 depicts the binary prediction of our model where 0 indicates as NO and 1 indicates as YES. And this prediction is based on previous table where a value of each class is less than 50% is depicts as NO and otherwise it marked as YES.

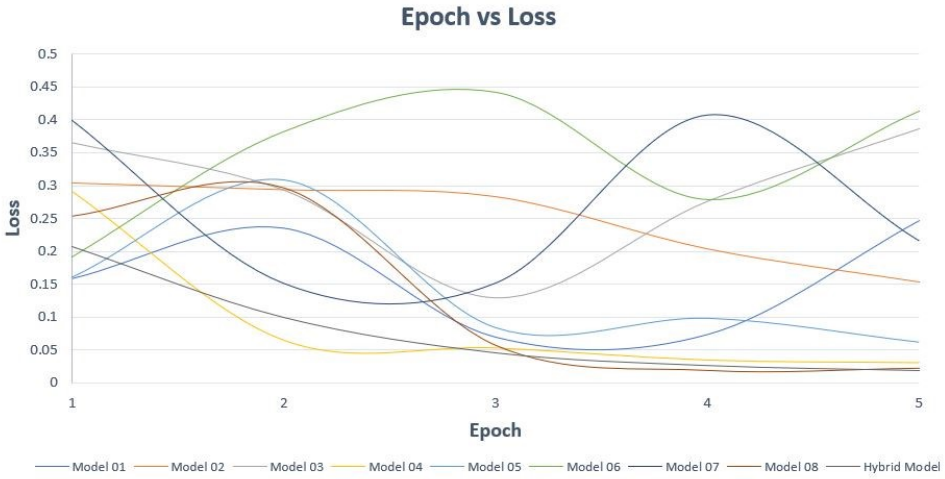


Figure 4.10: Epoch vs Loss (All Models)

Figure 4.10 shows that the total epochs and loss of each pre-trained transformer model including our hybrid ones. Here, we took only first 5 epochs for all the models.

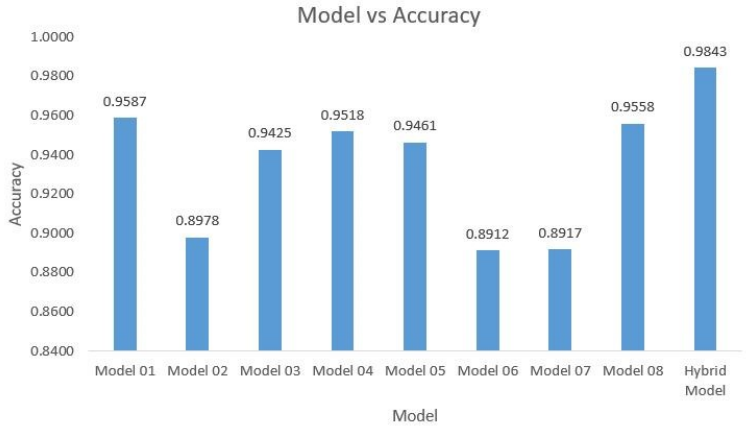


Figure 4.11: Model vs Accuracy (All Models)

Figure 4.11 shows that the accuracy of all the pre-trained models including our hybrid deep learning model. Here, we can see that the accuracy of our model is slightly better than other pre-trained models.

4.4 Applying Machine Learning Algorithms

As we discussed before that we are using eight machine learning algorithms. Here is the result and the visualization for each machine learning algorithms. In this section, we are showing feature importance, confusion matrix, validation score, etc. As well as we are showing SHAP and LIME explanation in this section.

4.4.1 Feature Analysis

In this section we are showing the result of feature importance, confusion matrix, cross validation score, recursive feature elimination, validation curve and learning

curve for particular machine learning algorithms.

Feature importance

Feature importance indicates which contexts had the greatest impact on each prediction made by classification or regression analysis[70]. Each attribute importance value has a value and a direction (positive or negative) that indicate how each field (or data point attribute) affects a particular prediction. It assigns a score to the input characteristics based on their importance in predicting the outcome. The more features are responsible for predicting the outcome, the higher their rating. It can be applied to classification and regression problems.

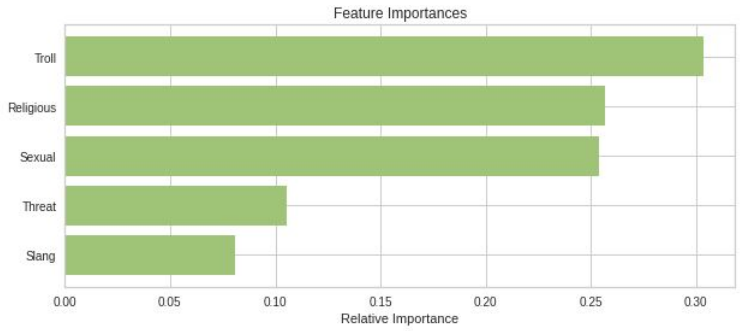


Figure 4.12: Feature Importance of XGBoost

Figure 4.12 shows the feature importance of XGBoost classifier. Here we can see that, XGBoost classifier takes mostly Troll, Sexual, and Religious class as feature importance.

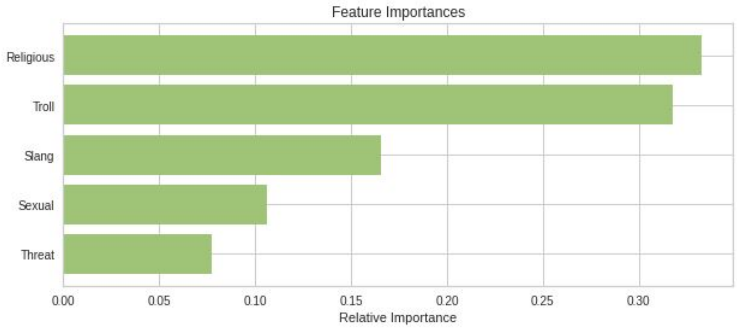


Figure 4.13: Feature Importance of Random Forest

Figure 4.13 shows the feature importance of Random Forest classifier. Here we can see that, Random Forest classifier takes mostly Troll, Sexual, and Religious class as feature importance.

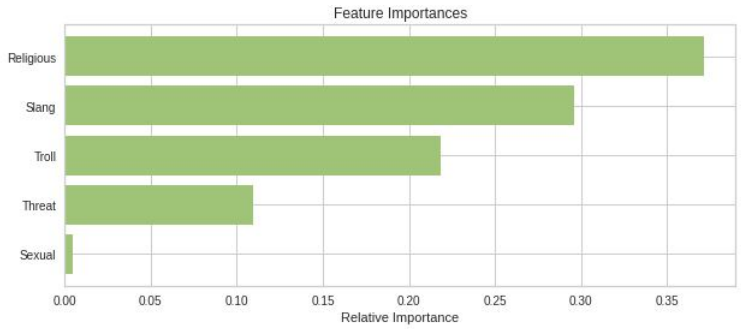


Figure 4.14: Feature Importance of Decision Trees

Figure 4.14 shows the feature importance of Decision Trees classifier. Here we can see that, Decision Trees classifier takes mostly Troll, and Sexual class as feature importance.

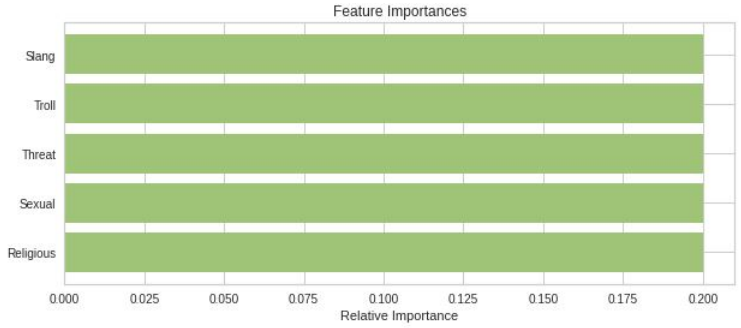


Figure 4.15: Feature Importance of AdaBoost

Figure 4.15 shows the feature importance of AdaBoost classifier. Here we can see that, AdaBoost classifier takes mostly Troll, Threat, Sexual, and Religious class as feature importance.

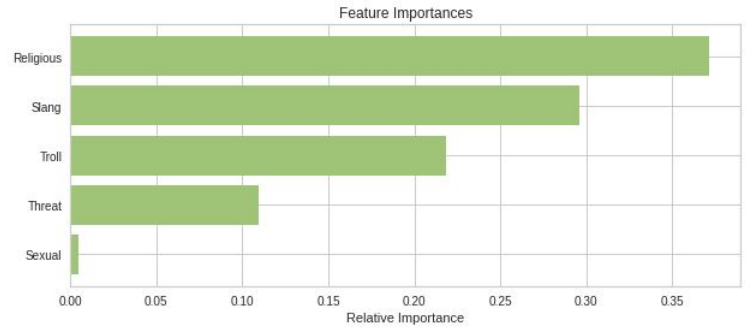


Figure 4.16: Feature Importance of Gradient Boosting

Figure 4.16 shows the feature importance of Gradient Boosting classifier. Here we can see that, Gradient Boosting classifier takes mostly Troll, Sexual, and Religious class as feature importance.

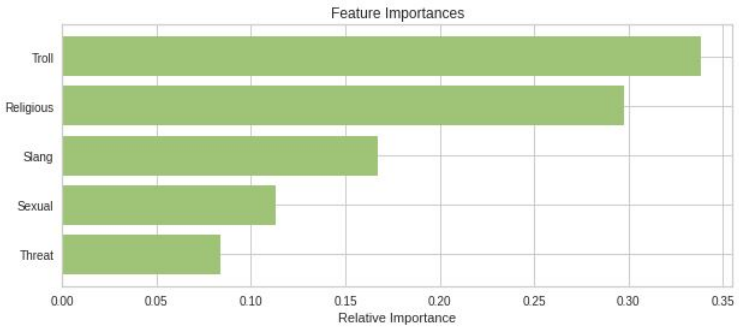


Figure 4.17: Feature Importance of Extra Trees

Figure 4.17 shows the feature importance of Extra Trees classifier. Here we can see that, Extra Trees classifier takes mostly Troll, Sexual, and Religious class as feature importance.

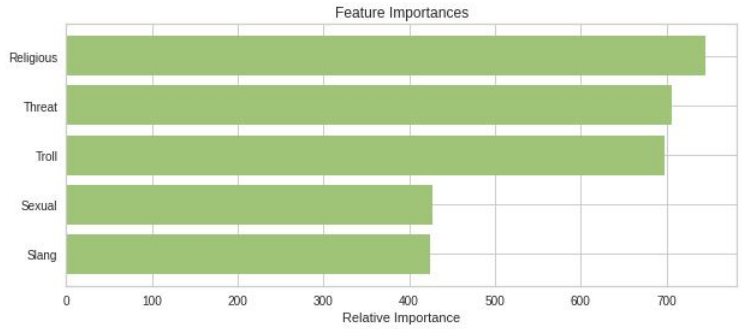


Figure 4.18: Feature Importance of Light GBM

Figure 4.18 shows the feature importance of Light GBM classifier. Here we can see that, Light GBM classifier takes mostly Troll, Sexual, and Religious class as feature importance.

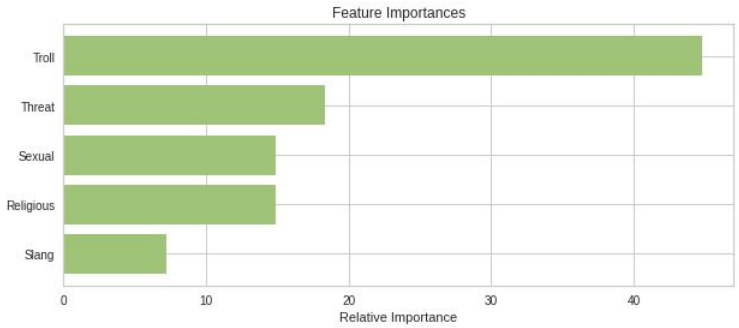


Figure 4.19: Feature Importance of Cat Boost

Figure 4.19 shows the feature importance of Cat Boost classifier. Here we can see that, Cat Boost classifier takes mostly Troll, Sexual, and Religious class as feature importance.

Confusion Matrix

Confusion matrices show the number of predicted and actual values[48]. The "TN" output stands for True Negative, indicating the number of accurately classified negative samples. The term "TP" stands for true positive, which denotes the quantity of positively identified samples that are correctly categorised. The abbreviation "FP" stands for "false positive," which refers to the number of genuine negative samples that were mistakenly labelled as positive. and "FN" is the number of real positive samples that were mistakenly labelled as negative. Accuracy is one of the most popular categorization standards.

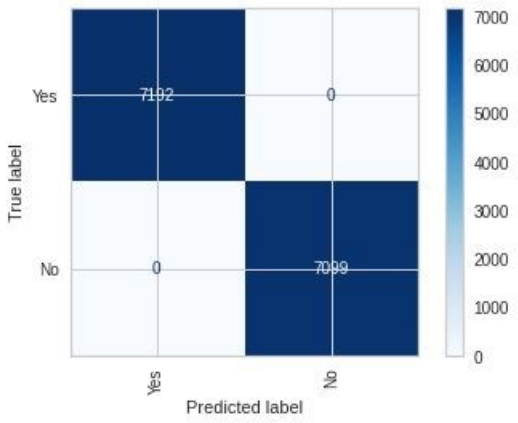


Figure 4.20: Confusion Matrix of XGBoost

Figure 4.20 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 7192 as our true positive which indicates at 7192 cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

FP (false positive): Here we get 0 as our false positive which indicates at 0 cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

TN (true negative): Here we get 7099 as our true negative which indicates at 7099 cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

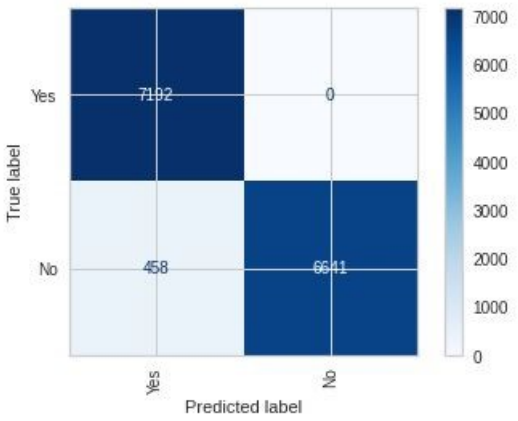


Figure 4.21: Confusion Matrix of Random Forest

Figure 4.21 shows the confusion matrix for the model using Random Forest 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 7192 as our true positive which indicates at 7192 cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

FP (false positive): Here we get 458 as our false positive which indicates at 458 cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

TN (true negative): Here we get 6641 as our true negative which indicates at 6641 cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

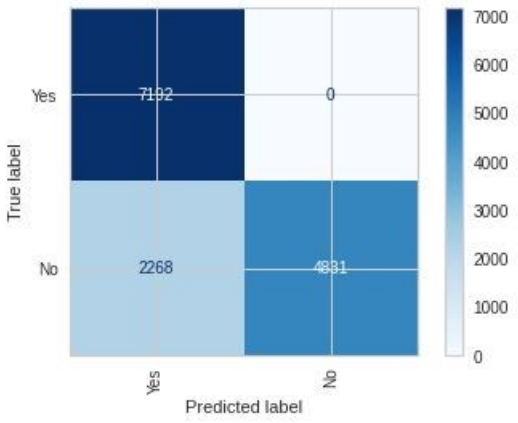


Figure 4.22: Confusion Matrix of Decision Trees

Figure 4.22 shows the confusion matrix for the model using Decision Trees 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 7192 as our true positive which indicates at 7192 cases the classifier predicted as ‘toxic’ and actually the text was ‘toxic’.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘toxic’.

FP (false positive): Here we get 2268 as our false positive which indicates at 2268 cases the classifier predicted as ‘toxic’ and actually the text was ‘non-toxic’.

TN (true negative): Here we get 4881 as our true negative which indicates at 4881 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘non-toxic’.

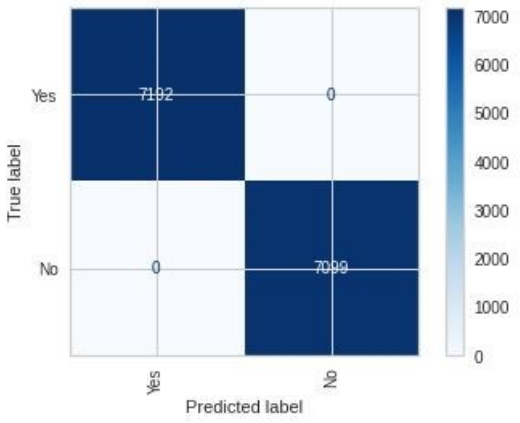


Figure 4.23: Confusion Matrix of AdaBoost

Figure 4.23 shows the confusion matrix for the model using AdaBoost 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 7192 as our true positive which indicates at 7192 cases the classifier predicted as ‘toxic’ and actually the text was ‘toxic’.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘toxic’.

FP (false positive): Here we get 0 as our false positive which indicates at 0 cases the classifier predicted as ‘toxic’ and actually the text was ‘non-toxic’.

TN (true negative): Here we get 7099 as our true negative which indicates at 7099 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘non-toxic’.

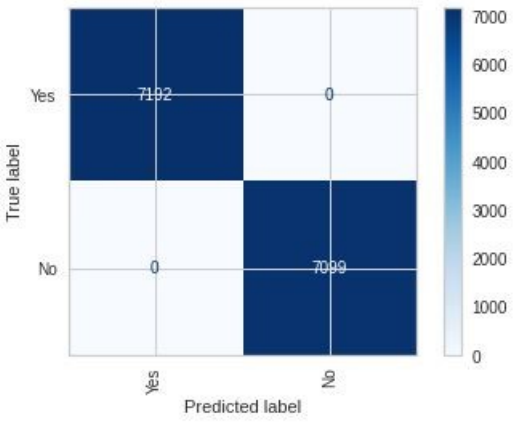


Figure 4.24: Confusion Matrix of Gradient Boosting

Figure 4.24 shows the confusion matrix for the model using Gradient Boosting 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 7192 as our true positive which indicates at 7192 cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

FP (false positive): Here we get 0 as our false positive which indicates at 0 cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

TN (true negative): Here we get 7099 as our true negative which indicates at 7099 cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

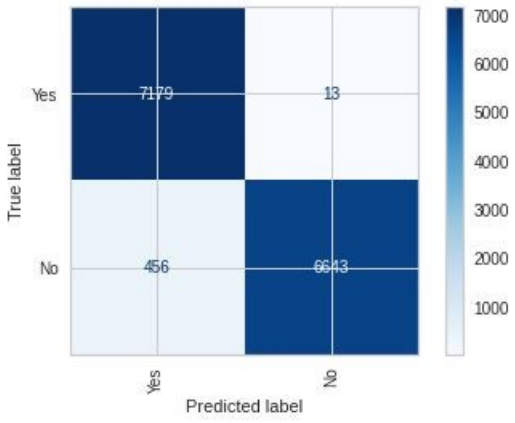


Figure 4.25: Confusion Matrix of Extra Trees

Figure 4.25 shows the confusion matrix for the model using Extra Trees 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 7179 as our true positive which indicates at 7179 cases the classifier predicted as ‘toxic’ and actually the text was ‘toxic’.

FN (false negative): Here we get 13 as our true negative which indicates at 13 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘toxic’.

FP (false positive): Here we get 456 as our false positive which indicates at 456 cases the classifier predicted as ‘toxic’ and actually the text was ‘non-toxic’.

TN (true negative): Here we get 6643 as our true negative which indicates at 6643 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘non-toxic’.

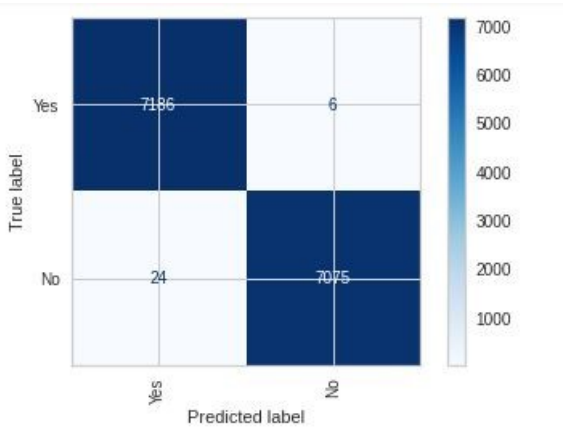


Figure 4.26: Confusion Matrix of Light GBM

Figure 4.26 shows the confusion matrix for the model using Light GBM 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 7186 as our true positive which indicates at 7186 cases the classifier predicted as ‘toxic’ and actually the text was ‘toxic’.

FN (false negative): Here we get 6 as our true negative which indicates at 6 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘toxic’.

FP (false positive): Here we get 24 as our false positive which indicates at 24 cases the classifier predicted as ‘toxic’ and actually the text was ‘non-toxic’.

TN (true negative): Here we get 7075 as our true negative which indicates at 7075 cases the classifier predicted as ‘non-toxic’ and actually the text was ‘non-toxic’.

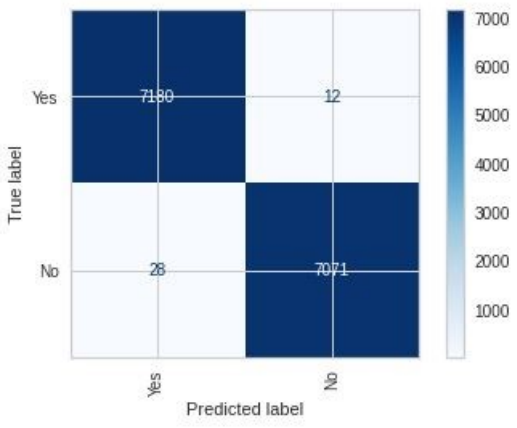


Figure 4.27: Confusion Matrix of Cat Boost

Figure 4.27 shows the confusion matrix for the model using Cat Boost 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 7180 as our true positive which indicates at 7180 cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

FN (false negative): Here we get 12 as our true negative which indicates at 12 cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

FP (false positive): Here we get 28 as our false positive which indicates at 28 cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

TN (true negative): Here we get 7071 as our true negative which indicates at 7071 cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

Recursive Feature Elimination

RFE is popular because it is easy to construct and use[60], and it is useful for selecting those features (columns) in the training dataset that are most or most related to the prediction of the target variable. There are two important tuning parameters when using RFE: the number of features to select and the choice of feature selection algorithm. Both of these hyperparameters can be tested, although the effectiveness of the method does not depend on the good structure of these hyperparameters.

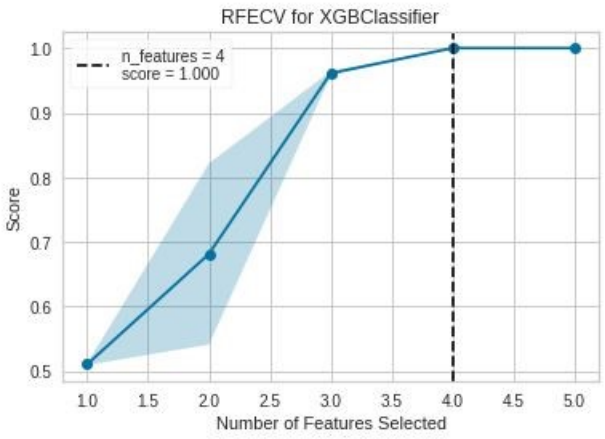


Figure 4.28: Recursive Feature Elimination of XGBoost

Figure 4.28 shows the recursive feature elimination of cross validation for XGBoost classifier. Here, it takes 4 no of features for recursive feature elimination and obtains a score of 1.000.

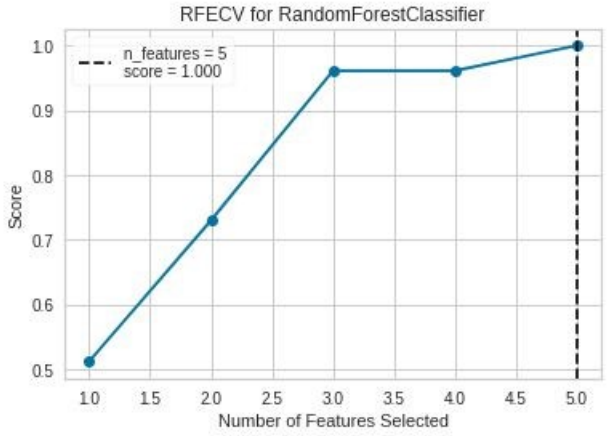


Figure 4.29: Recursive Feature Elimination of Random Forest

Figure 4.29 shows the recursive feature elimination of cross validation for Random Forest classifier. Here, it takes 5 no of features for recursive feature elimination and obtains a score of 1.000.

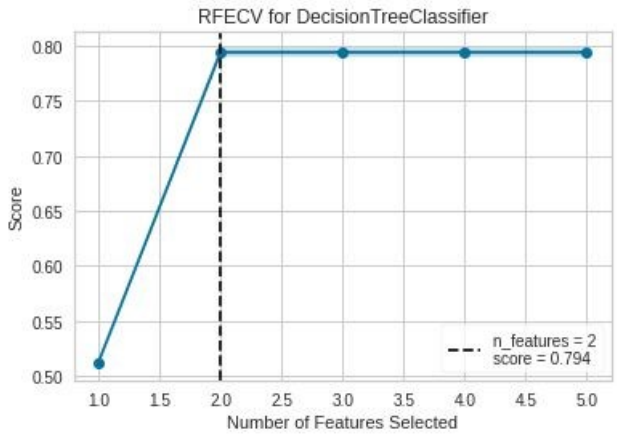


Figure 4.30: Recursive Feature Elimination of Decision Tree

Figure 4.30 shows the recursive feature elimination of cross validation for Decision Tree classifier. Here, it takes 2 no of features for recursive feature elimination and obtains a score of 0.794.

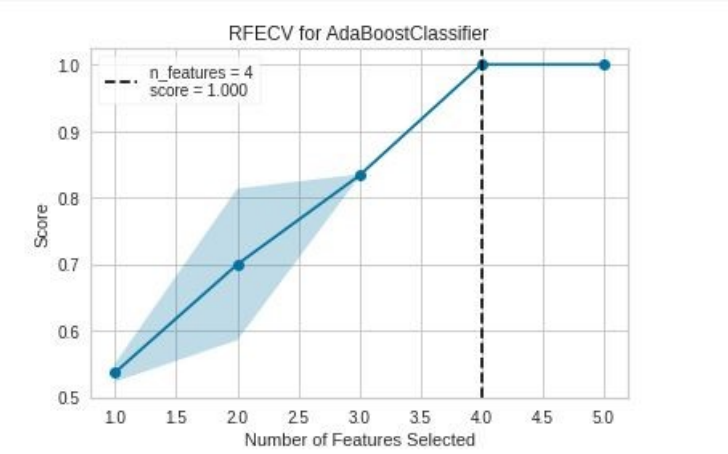


Figure 4.31: Recursive Feature Elimination of AdaBoost

Figure 4.31 shows the recursive feature elimination of cross validation for AdaBoost classifier. Here, it takes 4 no of features for recursive feature elimination and obtains a score of 1.000.

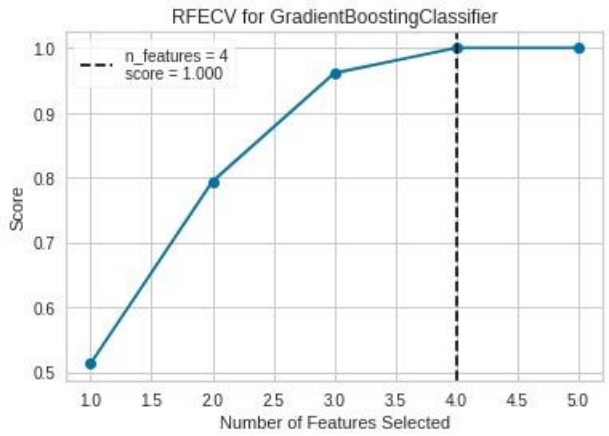


Figure 4.32: Recursive Feature Elimination of Gradient Boosting

Figure 4.32 shows the recursive feature elimination of cross validation for Gradient Boosting classifier. Here, it takes 4 no of features for recursive feature elimination and obtains a score of 1.000.

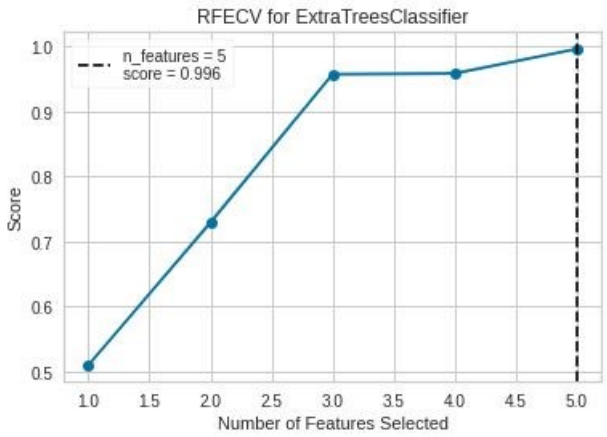


Figure 4.33: Recursive Feature Elimination of Extra Trees

Figure 4.33 shows the recursive feature elimination of cross validation for Extra Trees classifier. Here, it takes 5 no of features for recursive feature elimination and obtains a score of 0.996.

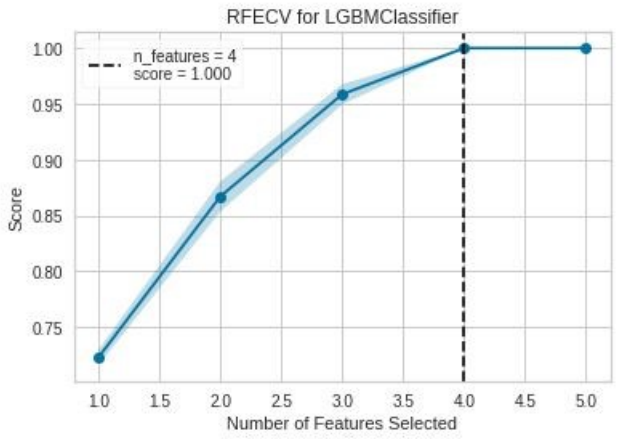


Figure 4.34: Recursive Feature Elimination of Light GBM

Figure 4.34 shows the recursive feature elimination of cross validation for Light GBM classifier. Here, it takes 4 no of features for recursive feature elimination and obtains a score of 1.000.

Cross Validation Score

CV means Cross Validation. It is a method of using existing training data to inform your model[6] and using that data to predict how well the model can predict new data results. This is useful when you have a small data set and want to increase the size of the training set after you have enough validation sets to make good predictions about the quality of the model.

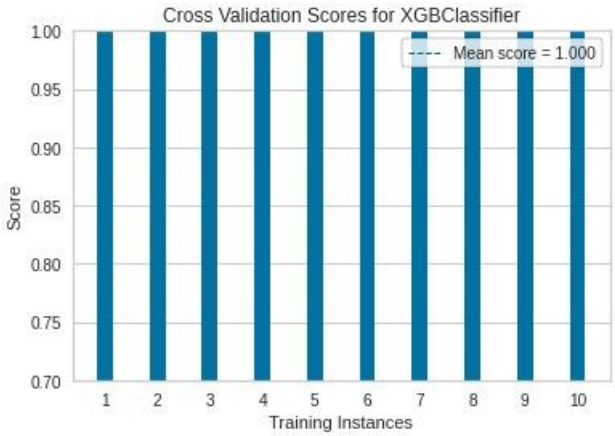


Figure 4.35: Cross-validation scores of XGBoost

Figure 4.35 shows the cross-validation score of XGBoost classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used

f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0.

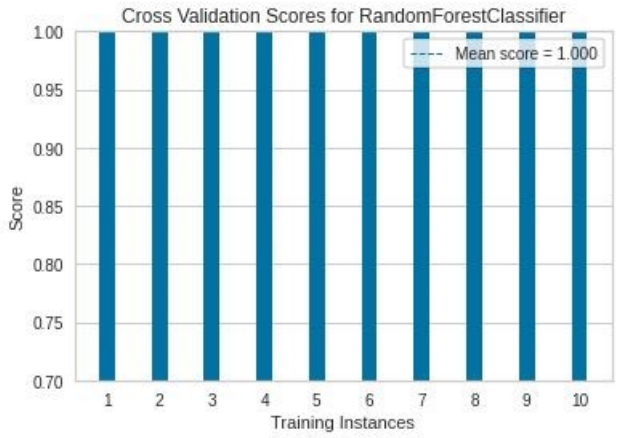


Figure 4.36: Cross-validation scores of Random Forest

Figure 4.36 shows the cross-validation score of Random Forest classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0.

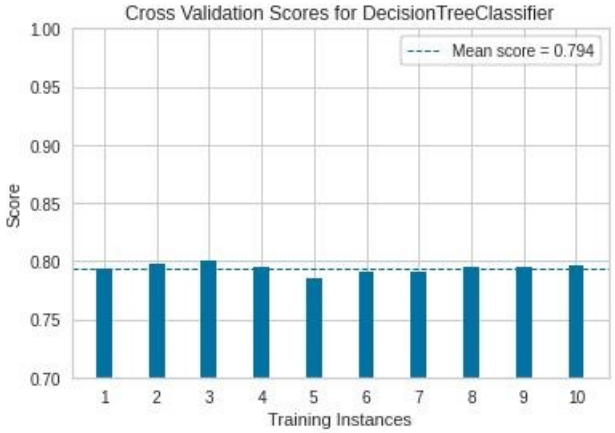


Figure 4.37: Cross-validation scores of Decision Trees

Figure 4.37 shows the cross-validation score of Random Decision Tree classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 0.794.

Figure 4.38: Cross-validation scores of AdaBoost

Figure 4.38 shows the cross-validation score of AdaBoost classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0.

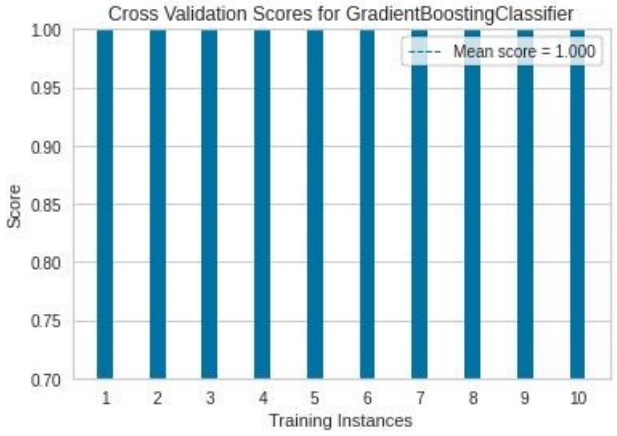


Figure 4.39: Cross-validation scores of Gradient Boosting

Figure 4.39 shows the cross-validation score of Gradient Boosting classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0

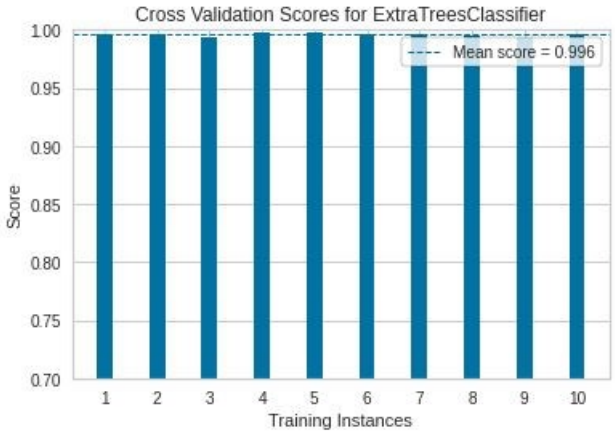


Figure 4.40: Cross-validation scores of Extra Trees

Figure 4.40 shows the cross-validation score of Extra Tree classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 0.996.

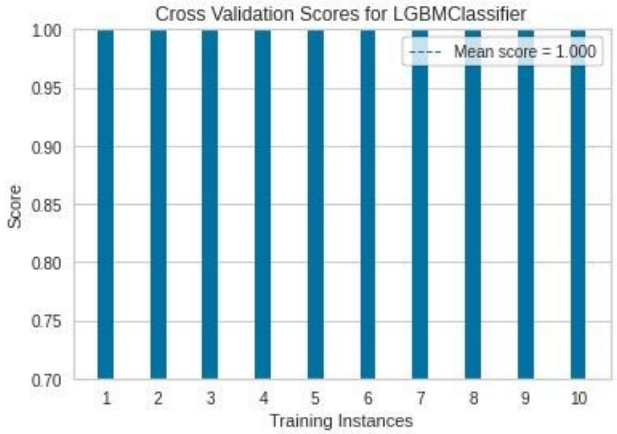


Figure 4.41: Cross-validation scores of Light GBM

Figure 4.41 shows the cross-validation score of Light GBM classifier. For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0.

Validation Curve

A validation curve is an important diagnostic tool that shows the sensitivity between changes in the accuracy of a machine learning model and changes in some model parameters[50]. A validation curve is usually drawn between some model parameters and the model estimate. The validation curve has two curves - one for training set estimation and one for cross-validation estimation.

The validation curve is used to evaluate an existing model based on the above parameters, not to modify the model. This is because if we fit a model based on a validation score, the model may be biased with respect to the specific data it is fitted to. This is a poor estimate of the generalizability of the model.

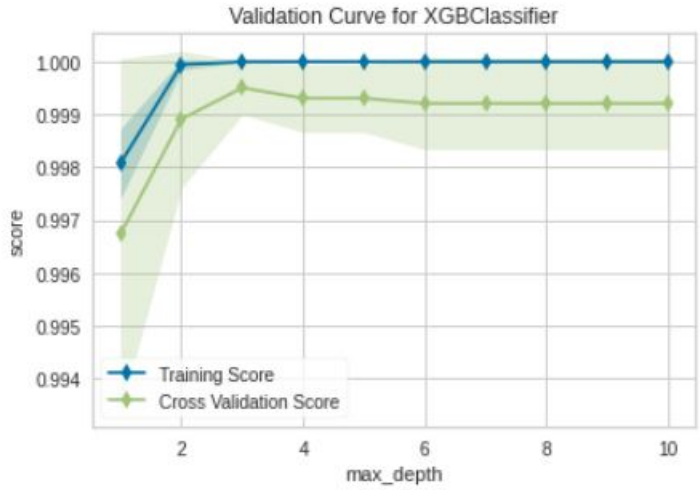


Figure 4.42: Validation-curve for XGBoost

Figure 4.42 shows the validation-curve for XGBoost classifier. Here it shows that the training score and validation score are in the same line. Here, we take 10 iterations as max depth to validate the curve.

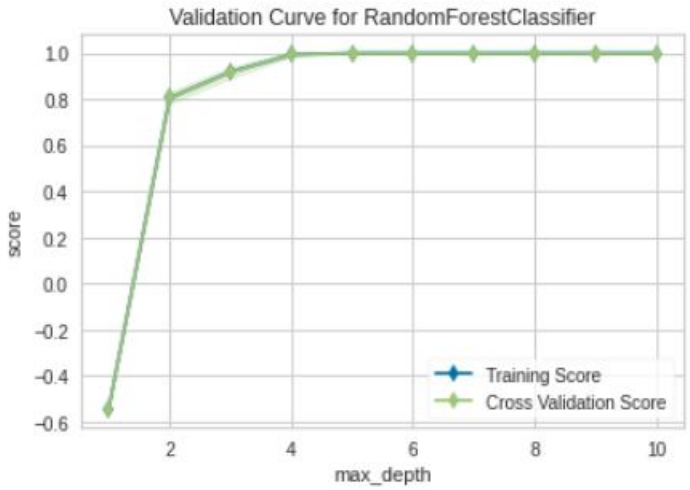


Figure 4.43: Validation-curve for Random Forest

Figure 4.43 shows the validation-curve for Random Forest classifier. Here it shows that the training score and validation score are in the same line. Here, we take 10 iterations as max depth to validate the curve.

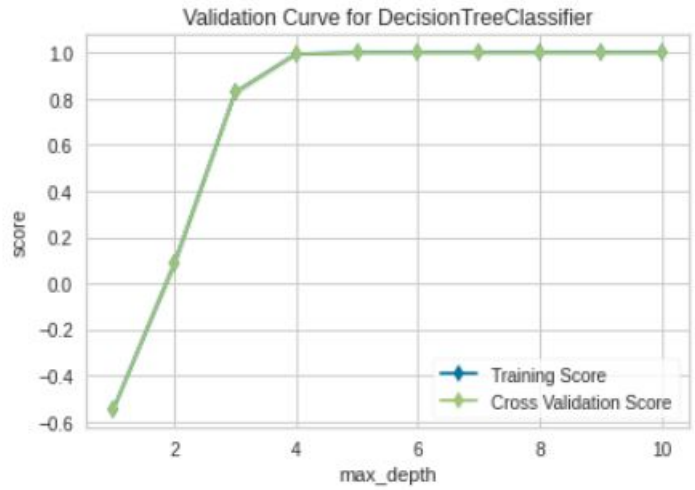


Figure 4.44: Validation-curve for Decision Trees

Figure 4.44 shows the validation-curve for Decision Tree classifier. Here it shows that the training score and validation score are in the same line. Here, we take 10 iterations as max depth to validate the curve.

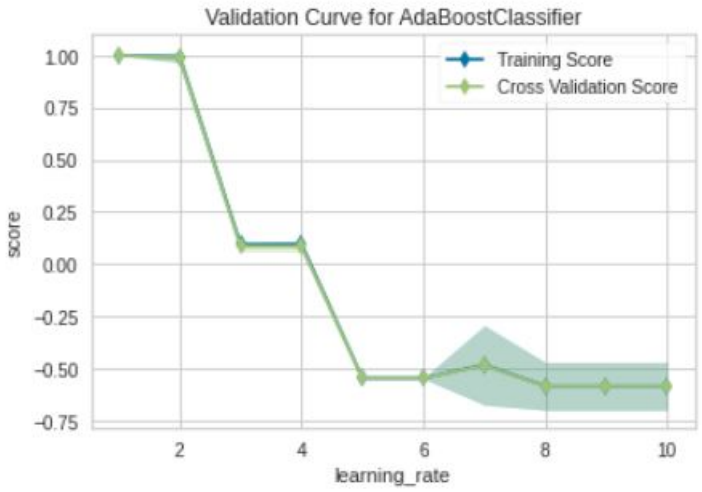


Figure 4.45: Validation-curve for AdaBoost

Figure 4.45 shows the validation-curve for AdaBoost classifier. Here it shows that the training score and validation score are in the same line. But gradually the curve line is going downward. Here, we take 10 iterations as max depth to validate the curve.

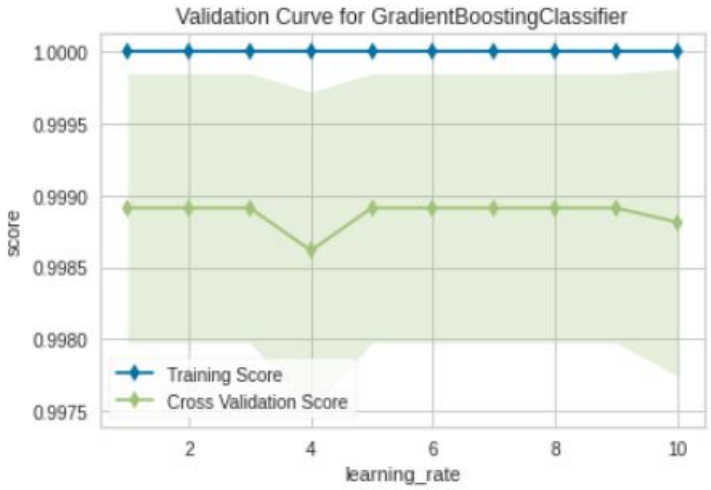


Figure 4.46: Validation-curve for Gradient Boosting

Figure 4.46 shows the validation-curve for Gradient Boosting classifier. Here it shows that the training score and validation score are almost in the same line. Here, we take 10 iterations as max depth to validate the curve.

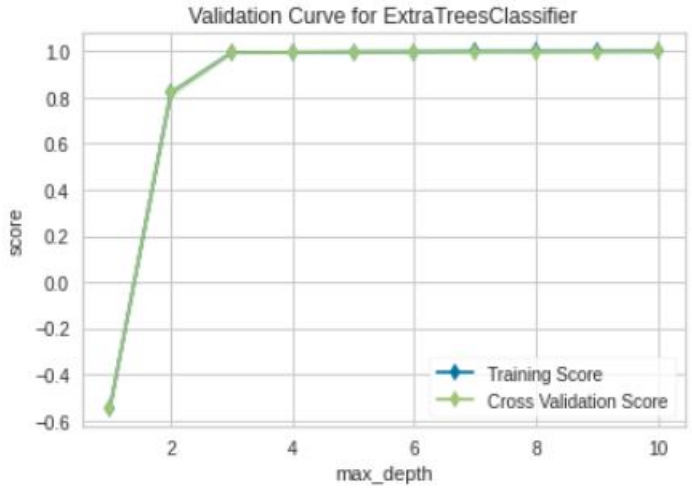


Figure 4.47: Validation-curve for Extra Trees

Figure 4.47 shows the validation-curve for Extra Tree classifier. Here it shows that the training score and validation score are in the same line. Here, we take 10 iterations as max depth to validate the curve.

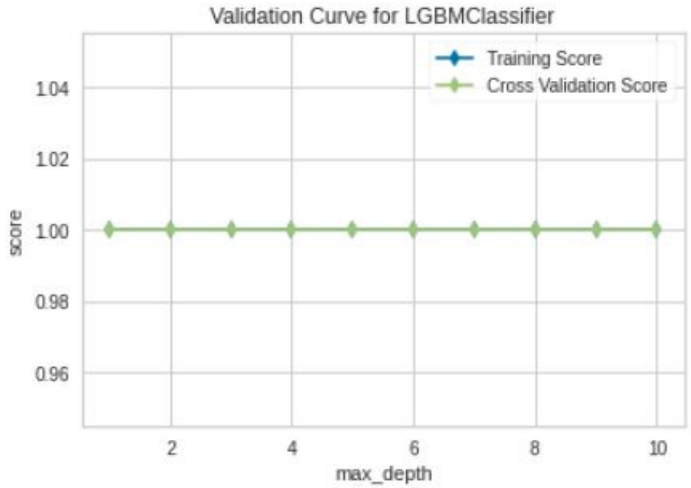


Figure 4.48: Validation-curve for Light GBM

Figure 4.48 shows the validation-curve for Light GBM classifier. Here it shows that the training score and validation score are in the same line. Here, we take 10 iterations as max depth to validate the curve.

Learning Curves

For algorithms that gradually learn (optimise their internal parameters over time), such as deep learning neural networks, learning curves are frequently employed in machine learning[58]. A maximum criteria may be used to measure learning, in which case larger scores indicate greater learning. Accuracy in classification is one instance.

It is more typical to utilise a lower score, such as losses or errors, where better scores (smaller numbers) denote more training and a value of 0.0 denotes that the training dataset was successfully trained with no mistakes. Every stage of the machine learning training procedure allows for the evaluation of the model's present state. On the basis of training data, the model may then be assessed to see how effectively it "learns." A dataset used for in-service validation that is separate from the training dataset can also be used to assess it. The validation dataset evaluation provides insight into how effectively the model "generalises."

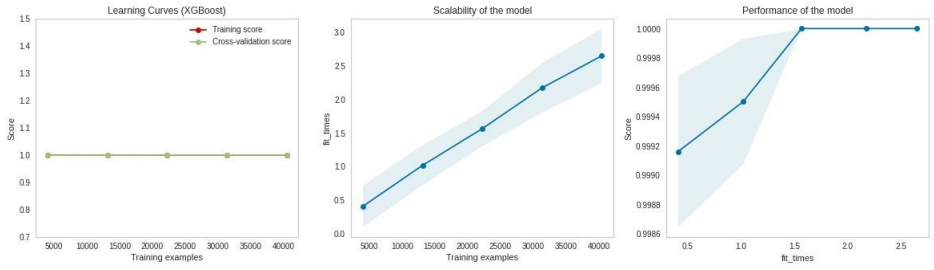


Figure 4.49: Learning curves for XGBoost

Figure 4.49 shows, Learning curves for XGBoost. Keep in mind that both the

training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models.The performance of this model is initially increasing then remain same to end.

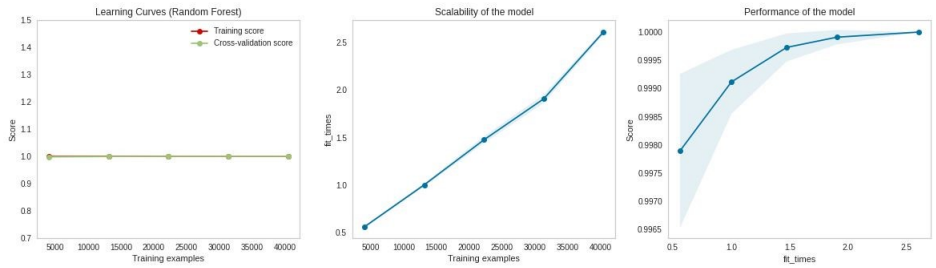


Figure 4.50: Learning curves for Random Forest

Figure 4.50 shows, Learning curves for Random Forest. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models. The performance of this model is gradually increasing.

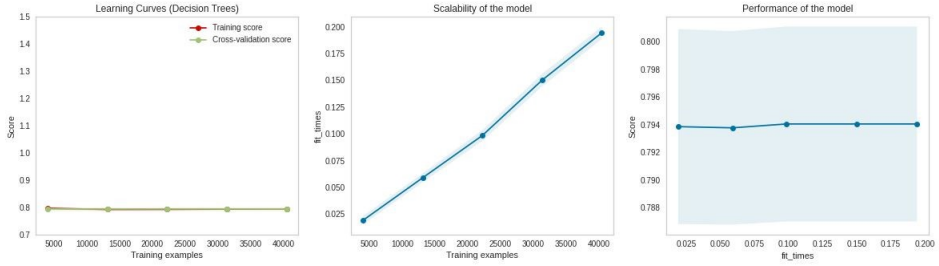


Figure 4.51: Learning curves for Decision Trees

Figure 4.51 shows, Learning curves for Decision Trees. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models. The performance of this model is same from start to end.

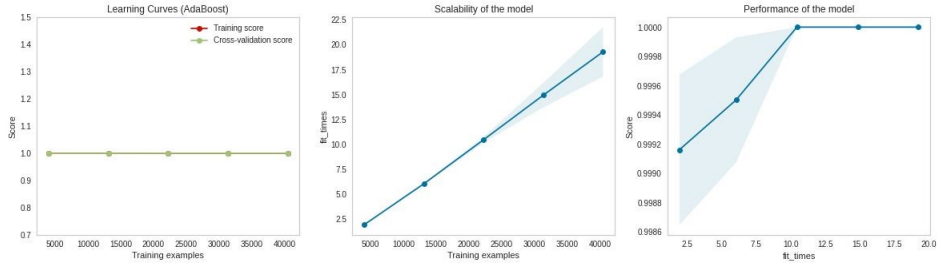


Figure 4.52: Learning curves for AdaBoost

Figure 4.52 shows, Learning curves for AdaBoost. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models.The performance of this model is initially increasing then remain same to end.

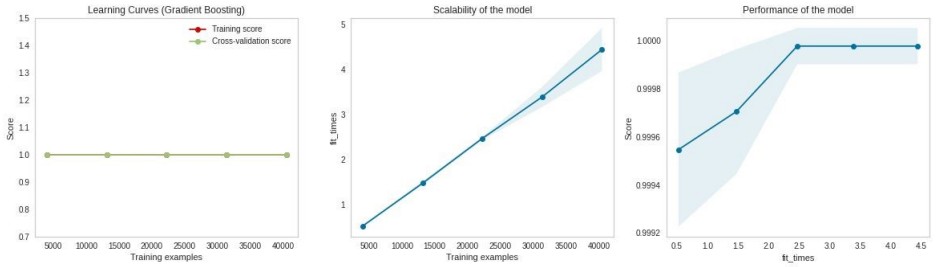


Figure 4.53: Learning curves for Gradient Boosting

Figure 4.53 shows, Learning curves for Gradient Boosting. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models.The performance of this model is initially increasing then remain same to end.

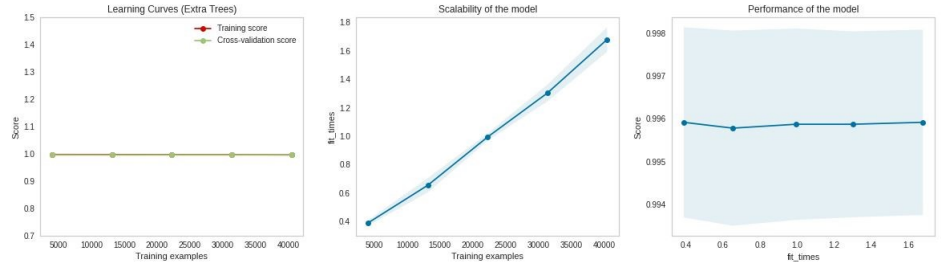


Figure 4.54: Learning curves for Extra Trees

Figure 4.54 shows, Learning curves for Extra Trees. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models. The performance of this model is same from start to end.

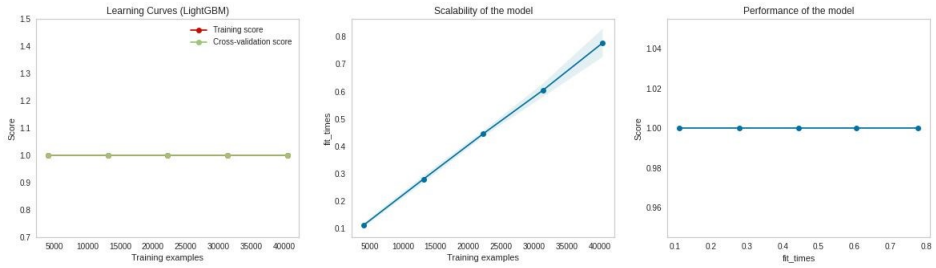


Figure 4.55: Learning curves for Light GBM

Figure 4.55 shows, Learning curves for Light GBM. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models. The performance of this model is exactly same from start to end.

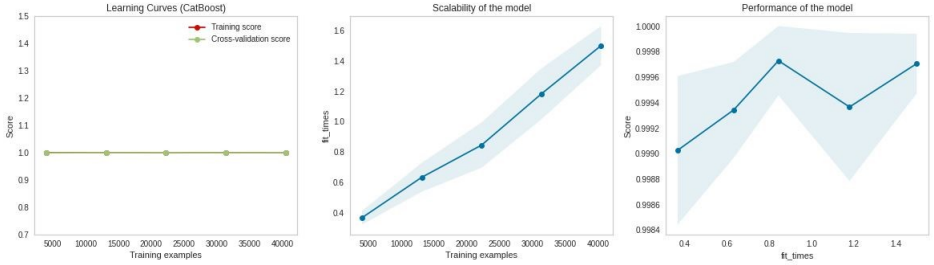


Figure 4.56: Learning curves for Cat Boost

Figure 4.56 shows, Learning curves for Cat Boost. Keep in mind that both the training score and the cross-validation score are excellent from beginning to end and nearly identical. We can plainly see that the training score is still close to the maximum and that adding more training samples will raise the validation score. The second figure displays the lengths of time the models need to learn with varied training dataset sizes. The final plot displays how long each training size took to train the models. The performance of this model is initially increasing then decreasing and finally again increasing.

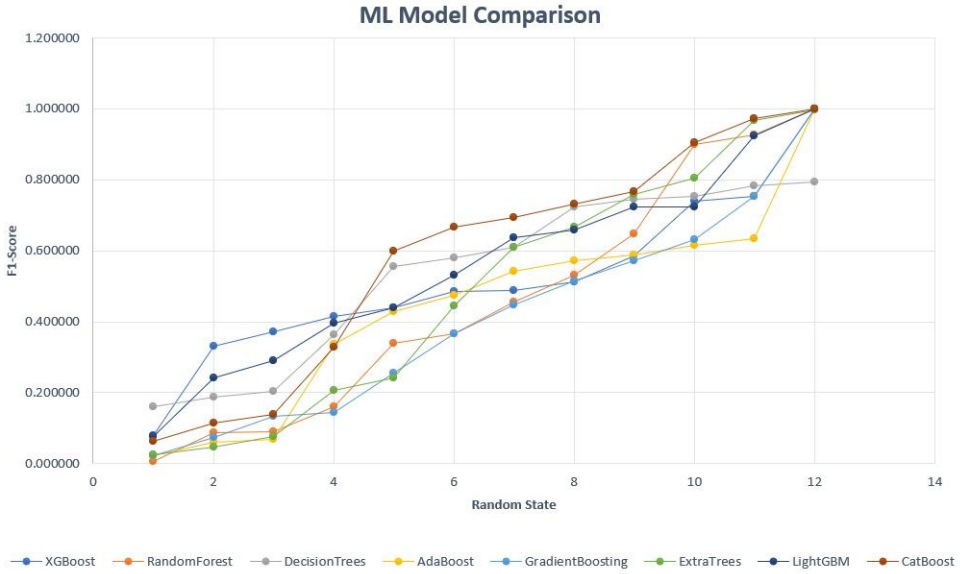


Figure 4.57: Machine Learning Model Comparison

Figure 4.57 shows the Machine Learning model comparison for all the classifiers. Here, we measured the F1-score for all the models along with 14 random state values.

4.4.2 Explainability of ML Models

We have used both LIME and SHAP for model explanation. In this section, we are showing the outcomes for our chosen machine learning models.

LIME

LIME provides local model descriptions. LIME modifies the data model by changing attribute values and observing the effect on the output. This is often due to the fact that people are interested in seeing the output of the model. We took one good comment and one bad comment to see how each of the classifier performs.

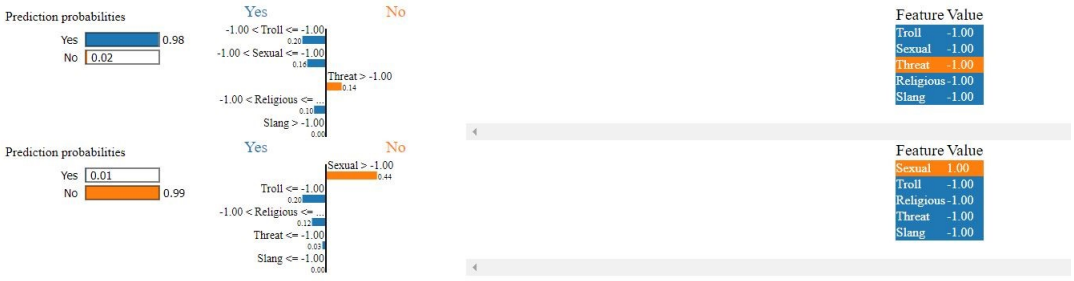


Figure 4.58: LIME explanation of XGBoost

In figure 4.58, the model is 98% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 99% of the time, and the class values raise that probability.

Figure 4.59: LIME explanation of Random Forest

In figure 4.59, the model is 56% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 78% of the time, and the class values raise that probability.

Figure 4.60: LIME explanation of Decision Trees

In figure 4.60, the model is 63% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 100% of the time, and the class values raise that probability.

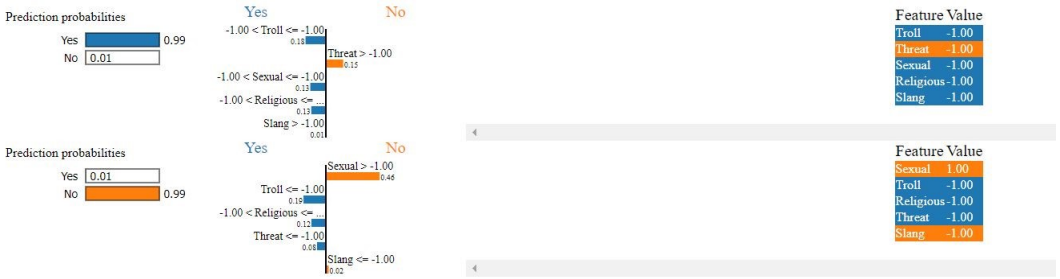


Figure 4.61: LIME explanation of AdaBoost

In figure 4.61, the model is 99% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 99% of the time, and the class values raise that probability.

Figure 4.62: LIME explanation of Gradient Boosting

In figure 4.62, the model is 98% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 99% of the time, and the class values raise that probability.

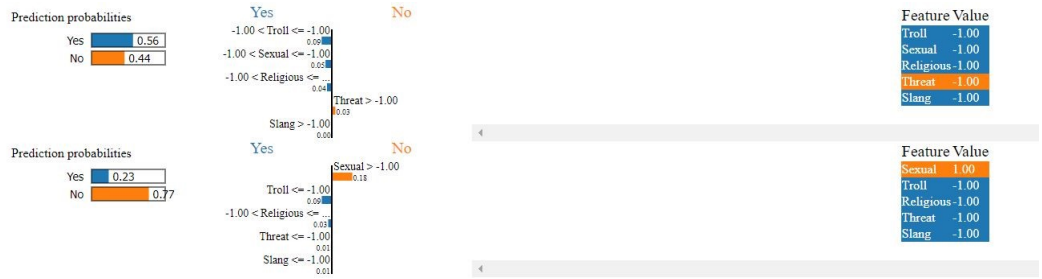


Figure 4.63: LIME explanation of Extra Trees

In figure 4.63, the model is 56% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 77% of the time, and the class values raise that probability.

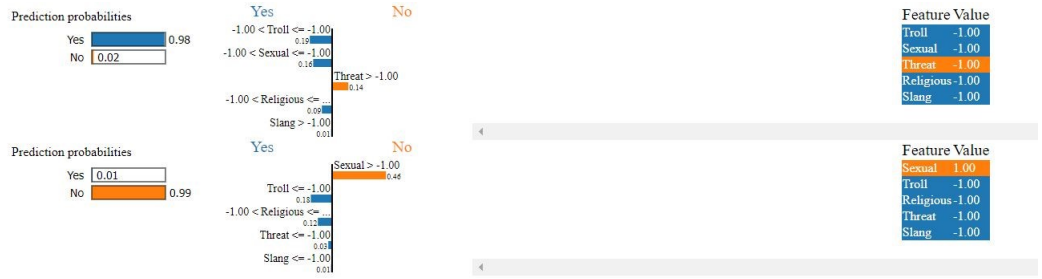


Figure 4.64: LIME explanation of Light GBM

In figure 4.64, the model is 98% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 99% of the time, and the class values raise that probability.

Figure 4.65: LIME explanation of Cat Boost

In figure 4.65, the model is 100% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 100% of the time, and the class values raise that probability.

SHAP

A summary plot combines the importance of a function with its effects. Each point in the abstract diagram is a Shapley value for a function and an event. The position on the y-axis is determined by the characteristic and the Shapley value on the x-axis. The color indicates the value of the attribute from low to high. The overlap points are moved along the y-axis, so we get a distribution of Shapley values for each function. Features are ranked by their importance.

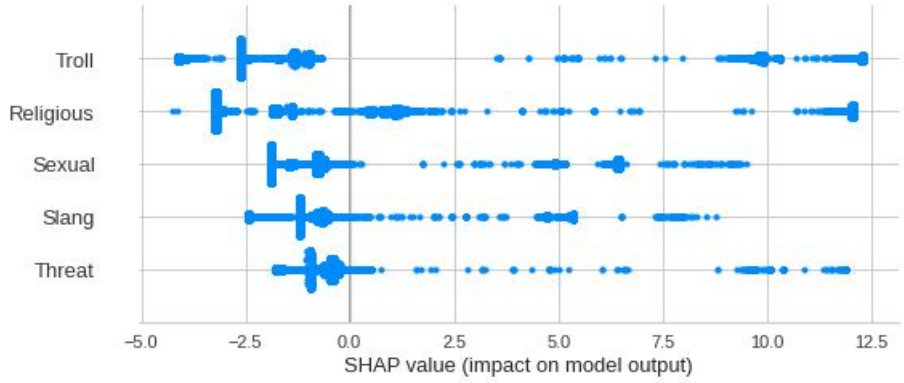


Figure 4.66: SHAP summary plot of XGBoost

Figure 4.66 shows, SHAP summary plot of XGBoost. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

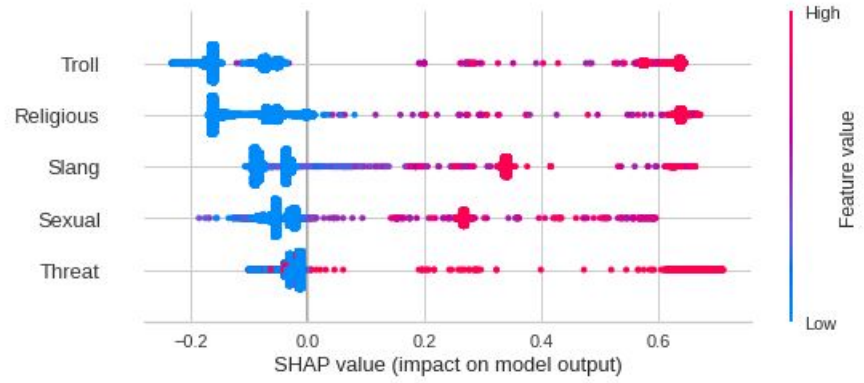


Figure 4.67: SHAP summary plot of Random Forest

Figure 4.67 shows, SHAP summary plot of Random Forest. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

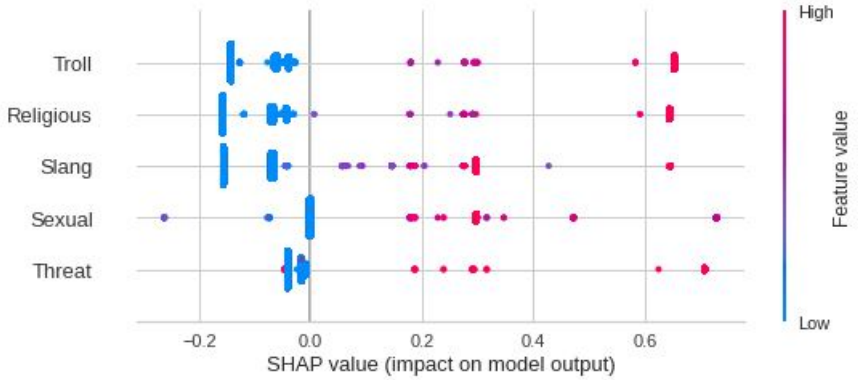


Figure 4.68: SHAP summary plot of Decision Trees

Figure 4.68 shows, SHAP summary plot of Decision Trees. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

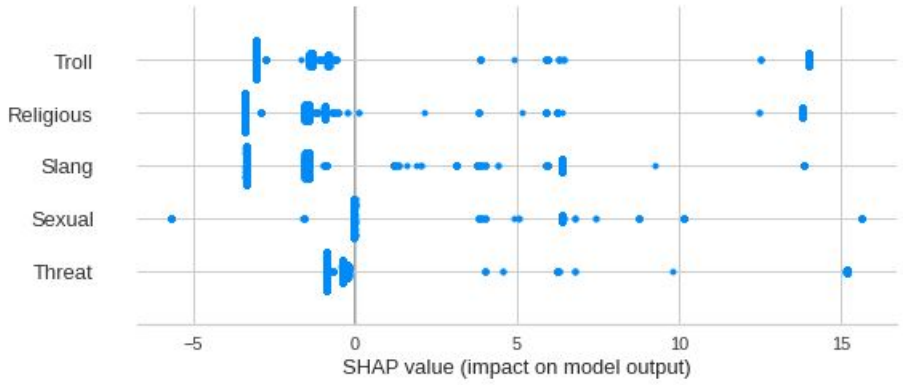


Figure 4.69: SHAP summary plot of Gradient Boosting

Figure 4.69 shows, SHAP summary plot of Gradient Boosting. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

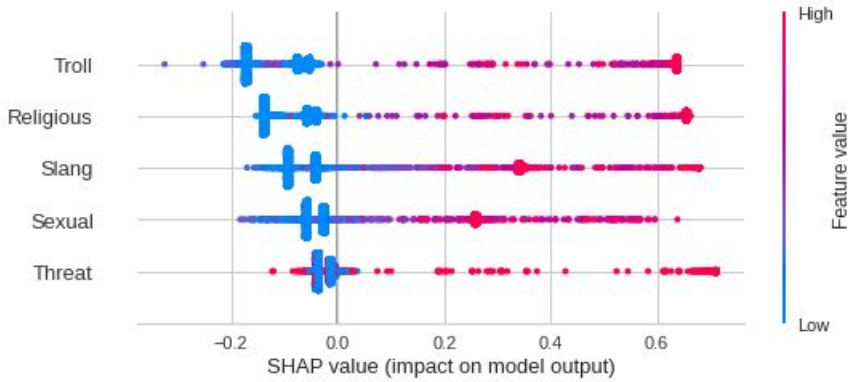


Figure 4.70: SHAP summary plot of Extra Trees

Figure 4.70 shows, SHAP summary plot of Extra Trees. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

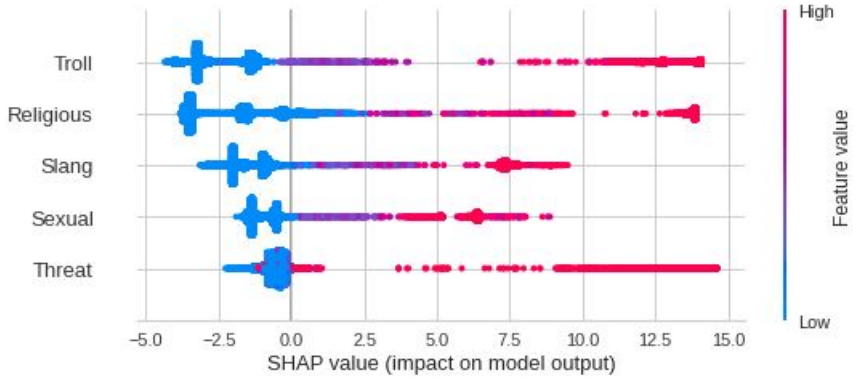


Figure 4.71: SHAP summary plot of Light GBM

Figure 4.71 shows, SHAP summary plot of Light GBM. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

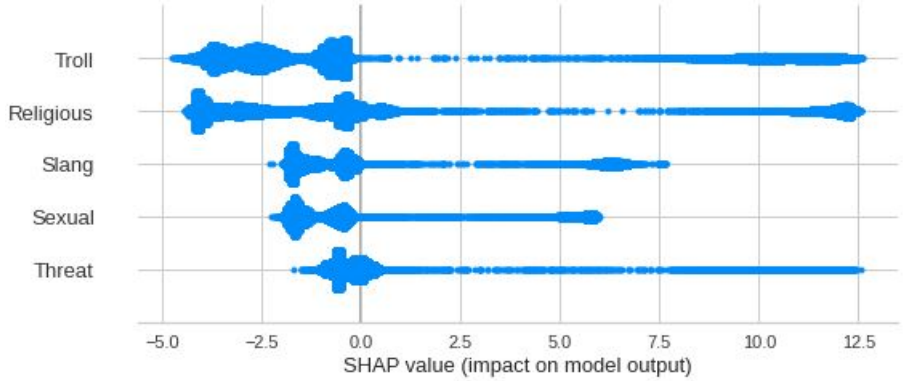


Figure 4.72: SHAP summary plot of Cat Boost

Figure 4.72 shows, SHAP summary plot of Cat Boost. The summary plot gives you a first hint of the relationship between feature values and their impact on predictions.

You may determine which characteristics had the biggest impact on the model's forecast for a single observation using the SHAP force plot. This is especially helpful if you ever need to justify something to your supervisor, like why your model told you to reject a certain person's loan application.

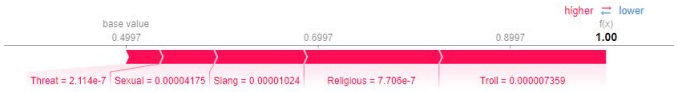


Figure 4.73: SHAP force plot of Random Forest

Figure 4.73 shows, SHAP force plot of Random Forest. In this case, the goal is to determine if the statement is harmful or not. The model's score for this observation is indicated by the bold 1.0 in the plot above. Lower scores cause the model to

predict 0, whereas higher values cause it to predict 1. The elements that were crucial to predicting this observation are depicted in red and blue, with red denoting features that increased the model score and blue denoting traits that decreased the score. The closer the feature is to the line separating red from blue, the more of an influence it had on the score, and the size of the bar indicates how much of an impact it had. So this particular comment is ultimately classified as toxic, because they were pushed higher by all the factors shown in red.

Figure 4.74: SHAP force plot of Decision Trees

Figure 4.74 shows, SHAP force plot of Decision Trees. In this case, the goal is to determine if the statement is harmful or not. The model's score for this observation is indicated by the bold 1.0 in the plot above. Lower scores cause the model to predict 0, whereas higher values cause it to predict 1. The elements that were crucial to predicting this observation are depicted in red and blue, with red denoting features that increased the model score and blue denoting traits that decreased the score. The closer the feature is to the line separating red from blue, the more of an influence it had on the score, and the size of the bar indicates how much of an impact it had. So this particular comment is ultimately classified as toxic, because they were pushed higher by all the factors shown in red. Figure 4.75 shows, SHAP

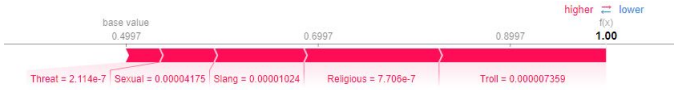


Figure 4.75: SHAP force plot of Extra Trees

force plot of Extra Trees. In this case, the goal is to determine if the statement is harmful or not. The model's score for this observation is indicated by the bold 1.0 in the plot above. Lower scores cause the model to predict 0, whereas higher values cause it to predict 1. The elements that were crucial to predicting this observation are depicted in red and blue, with red denoting features that increased the model score and blue denoting traits that decreased the score. The closer the feature is to the line separating red from blue, the more of an influence it had on the score, and the size of the bar indicates how much of an impact it had. So this particular comment is ultimately classified as toxic, because they were pushed higher by all the factors shown in red.

Figure 4.76: SHAP force plot of Light GBM

Figure 4.76 shows, SHAP force plot of Light GBM. In this case, the goal is to determine if the statement is harmful or not. The model's score for this observation is indicated by the bold 10.61 in the plot above. Lower scores cause the model to predict 0, whereas higher values cause it to predict 1. The elements that were crucial to predicting this observation are depicted in red and blue, with red denoting features that increased the model score and blue denoting traits that decreased the score. The closer the feature is to the line separating red from blue, the more of an influence it had on the score, and the size of the bar indicates how much of an impact it had. So this particular comment is ultimately classified as toxic, because they were pushed higher by all the factors shown in red.

The influence of a feature's evidence on the model's output is shown by the feature's SHAP value. The waterfall plot is made to show the evolution of the SHAP values graphically. Unfortunately, we lack this knowledge. Given the evidence of all the features, each feature shifts the model output from our previous expectation under the background data distribution to the final model prediction. The features are sorted by the magnitude of their SHAP values when the number of features in the models exceeds the maximum display parameter, with the features with the smallest magnitude clustered together at the bottom of the plot.

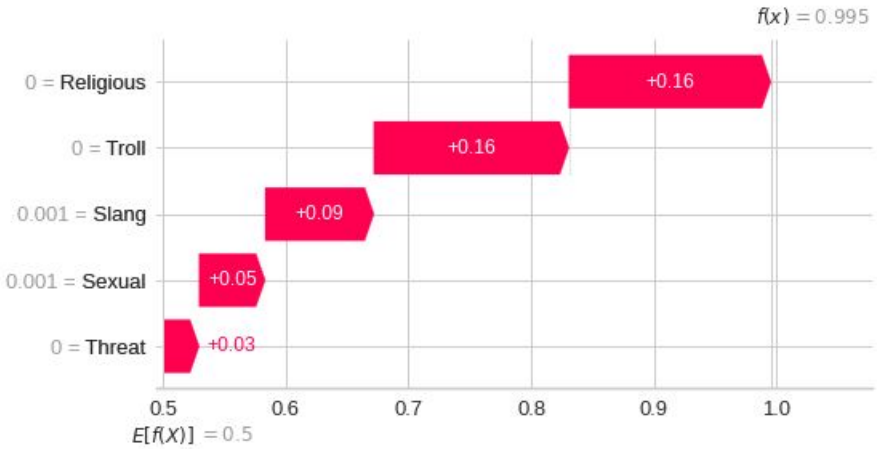


Figure 4.77: SHAP waterfall plot of Random Forest

Figure 4.77 shows, SHAP waterfall plot of Random Forest. Here, we use a waterfall plot to illustrate an explanation of a single forecast. The effect of the evidence a feature provides on the model's output is indicated by the feature's SHAP value. With the help of the waterfall plot, we can see clearly how the SHAP values (evidence) of individual features change the model output from what we had first anticipated based on the distribution of the background data to what the model predicts as a whole when all the features are present.

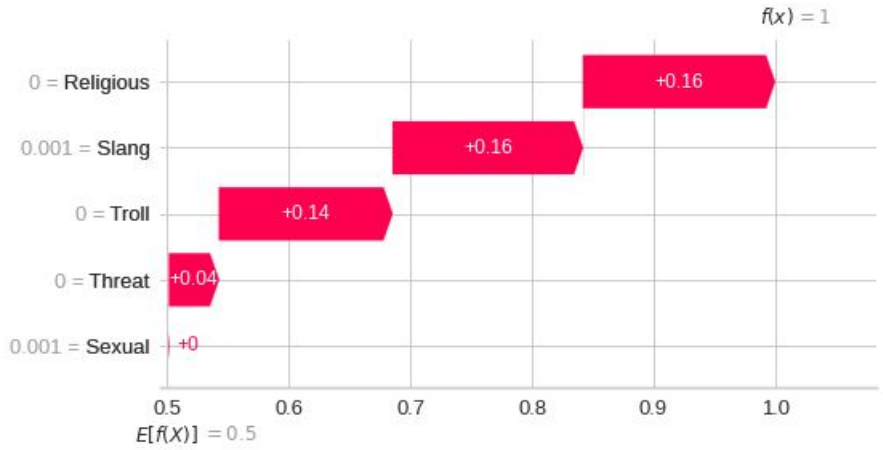


Figure 4.78: SHAP waterfall plot of Decision Trees

Figure 4.78 shows, SHAP waterfall plot of Decision Trees. Here, we use a waterfall plot to illustrate an explanation of a single forecast. The effect of the evidence a feature provides on the model's output is indicated by the feature's SHAP value. With the help of the waterfall plot, we can see clearly how the SHAP values (evidence) of individual features change the model output from what we had first anticipated based on the distribution of the background data to what the model predicts as a whole when all the features are present.

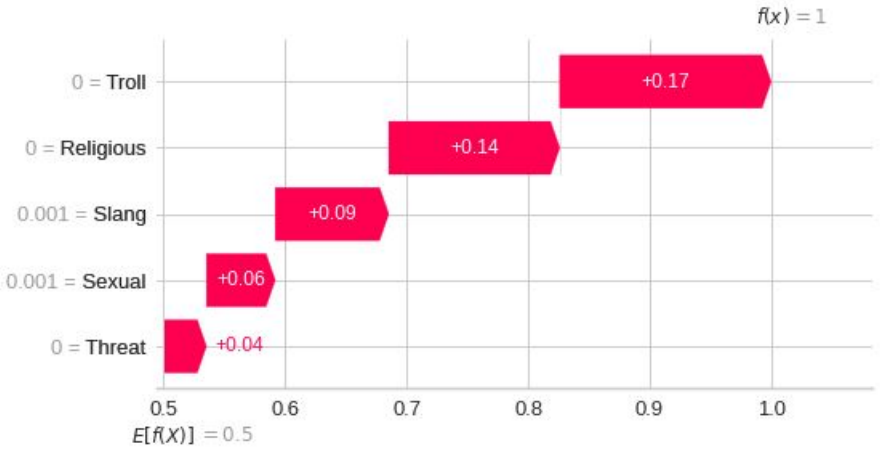


Figure 4.79: SHAP waterfall plot of Extra Trees

Figure 4.79 shows, SHAP waterfall plot of Extra Trees. Here, we use a waterfall plot to illustrate an explanation of a single forecast. The effect of the evidence a feature provides on the model's output is indicated by the feature's SHAP value. With the help of the waterfall plot, we can see clearly how the SHAP values (evidence) of individual features change the model output from what we had first anticipated based on the distribution of the background data to what the model predicts as a whole when all the features are present.

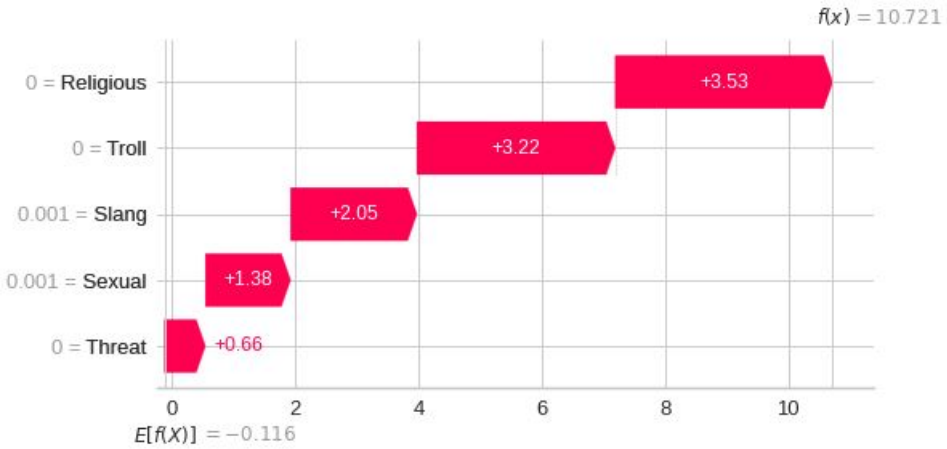


Figure 4.80: SHAP waterfall plot of Light GBM

Figure 4.80 shows, SHAP waterfall plot of Light GBM. Here, we use a waterfall plot to illustrate an explanation of a single forecast. The effect of the evidence a feature provides on the model's output is indicated by the feature's SHAP value. With the help of the waterfall plot, we can see clearly how the SHAP values (evidence) of individual features change the model output from what we had first anticipated based on the distribution of the background data to what the model predicts as a whole when all the features are present.

Another useful plot for interpreting the results is the dependence plot. This function makes a simple dependence plot with feature values on the x-axis and SHAP values on the y-axis. It is optional to color by another feature. By default, this function makes a simple dependence plot with feature values on the x-axis and SHAP values on the y-axis. You are welcome to use a different variable for the SHAP values on the y-axis, and color the points by the value of a designated variable.

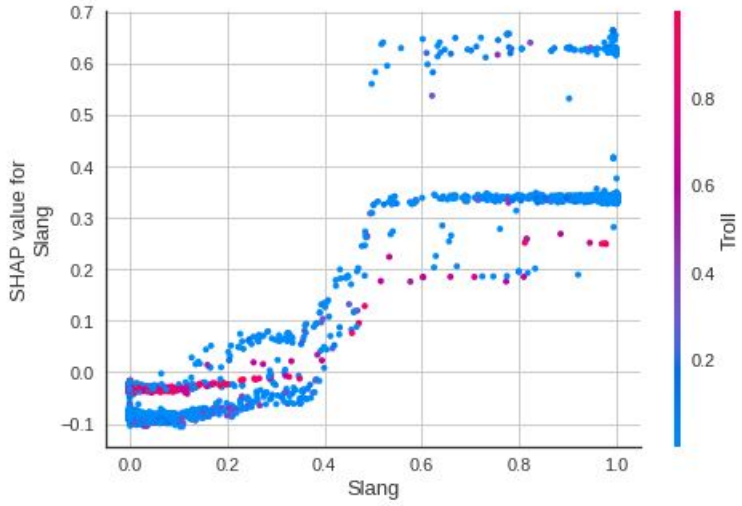


Figure 4.81: SHAP dependence plots of Random Forest

Figure 4.81 shows, SHAP dependence plots of Random Forest. One prediction (row)

from the dataset is represented by each dot. The value of the characteristic is on the x-axis (from the X matrix). The feature's SHAP value, which shows how much the output of the model for the prediction of that sample changes when the value of that feature is known, is plotted on the y-axis. The units in this model are the prediction of toxicity in each comment. The color is associated with a secondary feature that might interact with the feature we are graphing. A distinct vertical pattern of coloring will appear if there is an interaction impact between this other variable and the feature we are plotting. This implies an interaction impact between slang and troll in the scenario.

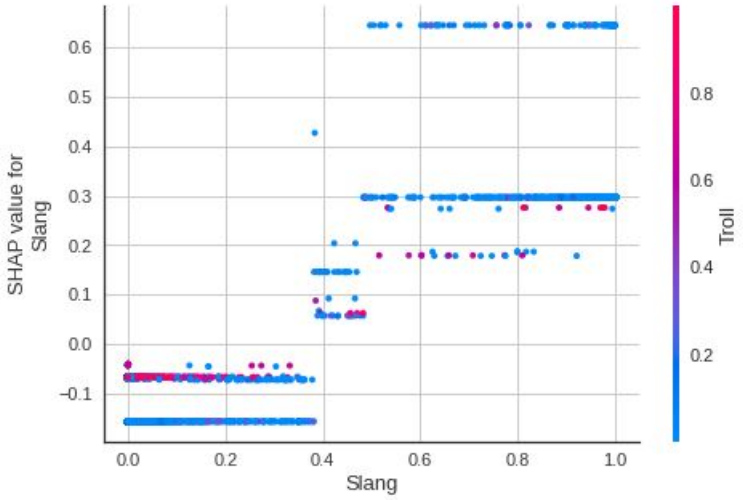


Figure 4.82: SHAP dependence plots of Decision Trees

Figure 4.82 shows, SHAP dependence plots of Decision Trees. One prediction (row) from the dataset is represented by each dot. The value of the characteristic is on the x-axis (from the X matrix). The feature's SHAP value, which shows how much the output of the model for the prediction of that sample changes when the value of that feature is known, is plotted on the y-axis. The units in this model are the prediction of toxicity in each comment. The color is associated with a secondary feature that might interact with the feature we are graphing. A distinct vertical pattern of coloring will appear if there is an interaction impact between this other variable and the feature we are plotting. This implies an interaction impact between slang and troll in the scenario.

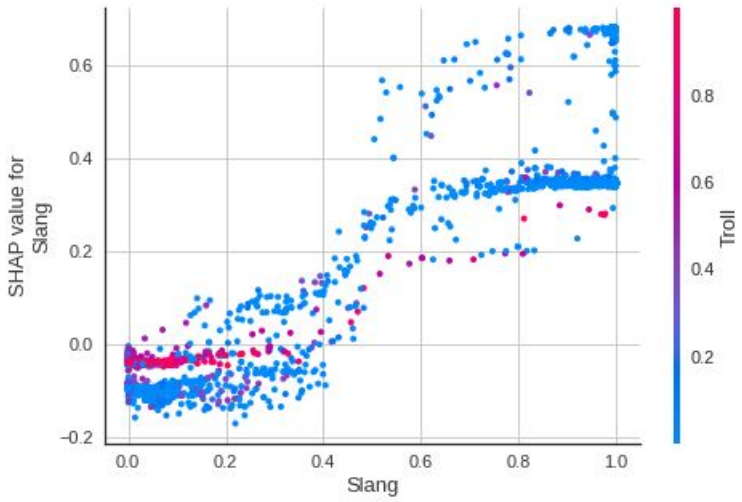


Figure 4.83: SHAP dependence plots of Extra Trees

Figure 4.83 shows, SHAP dependence plots of Extra Trees. One prediction (row) from the dataset is represented by each dot. The value of the characteristic is on the x-axis (from the X matrix). The feature's SHAP value, which shows how much the output of the model for the prediction of that sample changes when the value of that feature is known, is plotted on the y-axis. The units in this model are the prediction of toxicity in each comment. The color is associated with a secondary feature that might interact with the feature we are graphing. A distinct vertical pattern of coloring will appear if there is an interaction impact between this other variable and the feature we are plotting. This implies an interaction impact between slang and troll in the scenario.

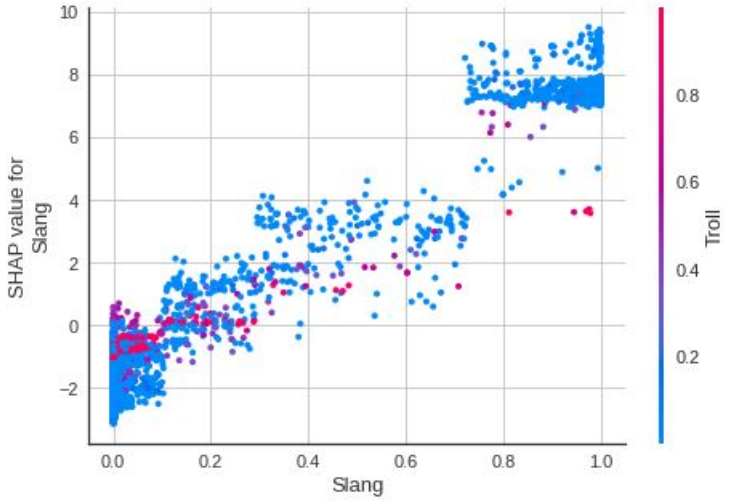


Figure 4.84: SHAP dependence plots of Light GBM

Figure 4.84 shows, SHAP dependence plots of Light GBM. One prediction (row) from the dataset is represented by each dot. The value of the characteristic is on the x-axis (from the X matrix). The feature's SHAP value, which shows how much the output of the model for the prediction of that sample changes when the value of that feature is known, is plotted on the y-axis. The units in this model are the

prediction of toxicity in each comment. The color is associated with a secondary feature that might interact with the feature we are graphing. A distinct vertical pattern of coloring will appear if there is an interaction impact between this other variable and the feature we are plotting. This implies an interaction impact between slang and troll in the scenario.

SHAP decision plots demonstrate how complex models arrive at their predictions. The decision plot supports SHAP interaction values - first-order interactions estimated from tree-based models. A dependency diagram shows a single interaction for many predictions, whereas a decision diagram shows all major effects and interactions together.

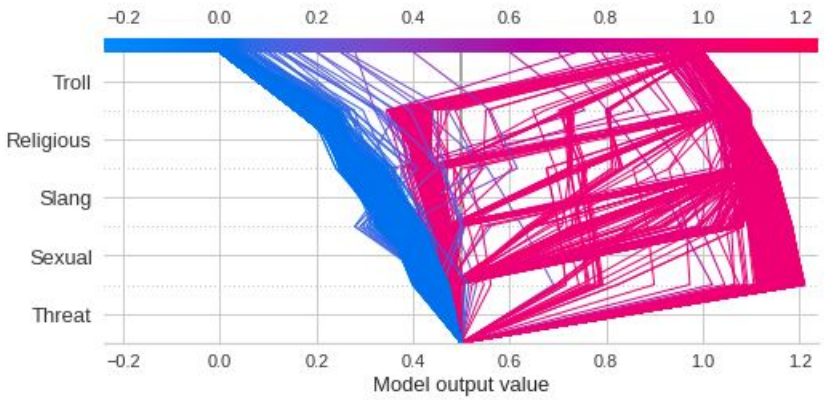


Figure 4.85: SHAP decision plots of Random Forest

Figure 4.85 shows, SHAP decision plots of Random Forest. It provides a broad overview of the contribution to prediction as the summary plot. In order to calculate the output values, shap values are gradually added to the model's base value from bottom to top of the decision plot. One can see that some of the blue-colored strings produced a final class value of 0, while the remaining red-colored strings produced a final class value of 1.

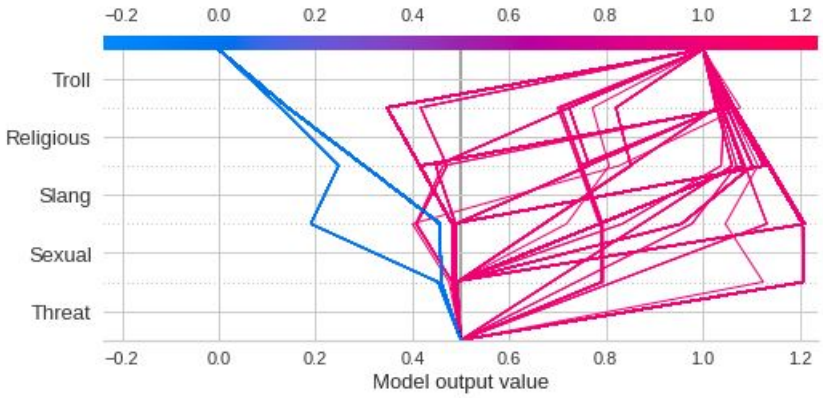


Figure 4.86: SHAP decision plots of Decision Trees

Figure 4.86 shows, SHAP decision plots of Decision Trees. It provides a broad overview of the contribution to prediction as the summary plot. In order to calculate

the output values, shap values are gradually added to the model's base value from bottom to top of the decision plot. One can see that some of the blue-colored strings produced a final class value of 0, while the remaining red-colored strings produced a final class value of 1.

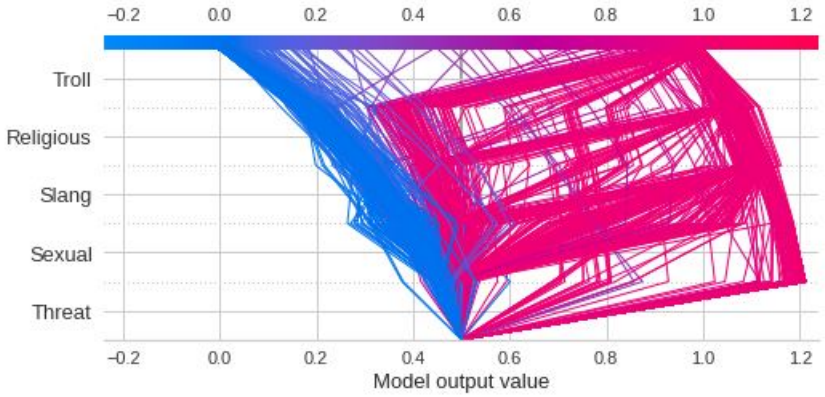


Figure 4.87: SHAP decision plots of Extra Trees

Figure 4.87 showws, SHAP decision plots of Extra Trees. It provides a broad overview of the contribution to prediction as the summary plot. In order to calculate the output values, shap values are gradually added to the model's base value from bottom to top of the decision plot. One can see that some of the blue-colored strings produced a final class value of 0, while the remaining red-colored strings produced a final class value of 1.

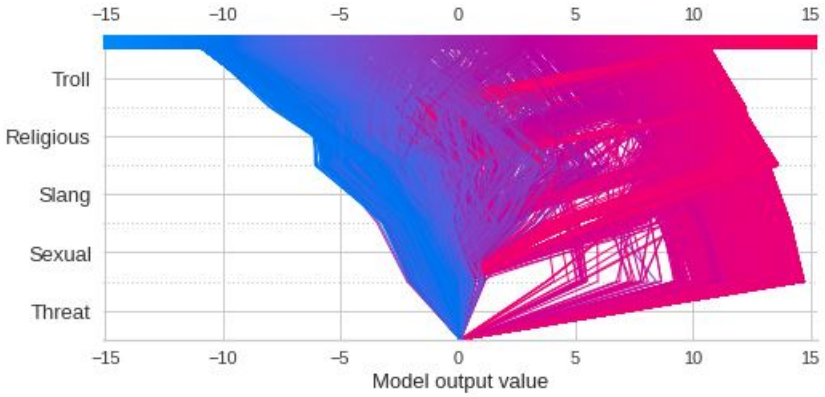


Figure 4.88: SHAP decision plots of Light GBM

Figure 4.88 shows, SHAP decision plots of Light GBM. It provides a broad overview of the contribution to prediction as the summary plot. In order to calculate the output values, shap values are gradually added to the model's base value from bottom to top of the decision plot. One can see that some of the blue-colored strings produced a final class value of 0, while the remaining red-colored strings produced a final class value of 1.

4.5 Ensemble Modeling

In this section, we stacked all the classifiers in a list and made an ensemble model. We found an outstanding performance of this model. We used voting classifier to stack all these classifiers. The output result has been shown in below figures:

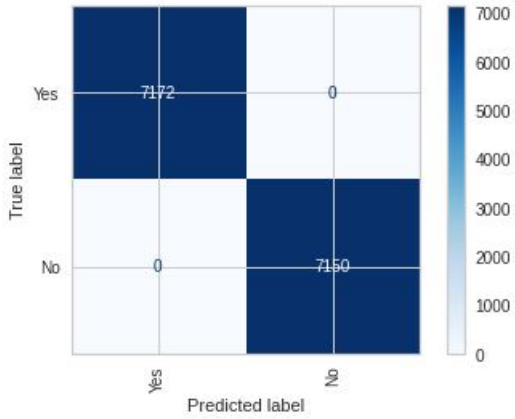


Figure 4.89: Confusion Matrix of Voting Classifier

Figure 4.89 shows the confusion matrix of voting classifiers. 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 7172 as our true positive which indicates at 7172 cases the classifier predicted as 'toxic' and actually the text was 'toxic'.

FN (false negative): Here we get 0 as our true negative which indicates at 0 cases the classifier predicted as 'non-toxic' and actually the text was 'toxic'.

FP (false positive): Here we get 0 as our false positive which indicates at 0 cases the classifier predicted as 'toxic' and actually the text was 'non-toxic'.

TN (true negative): Here we get 7150 as our true negative which indicates at 7150 cases the classifier predicted as 'non-toxic' and actually the text was 'non-toxic'.

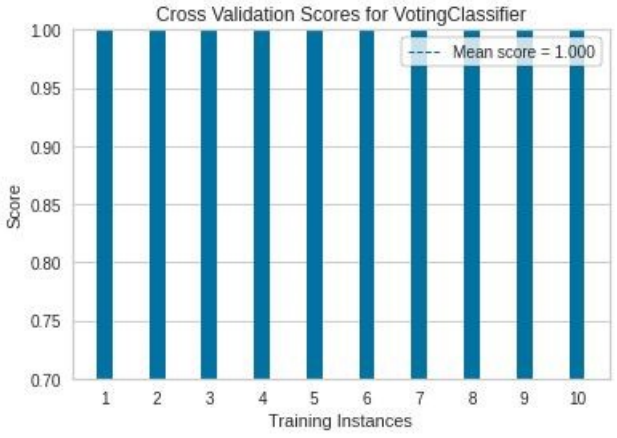


Figure 4.90: Cross-validation scores of Voting Classifier

For cross validation, we used a k-fold approach. In this approach, the data set is divided into 10 (k=10) number of subsets, or "folds," before training is carried out on each subset, but only one (k=1) subset is used to evaluate the trained model. This approach involves 10 iterations, with a new subset set aside for testing each time. Figure 4.90 shows the cross-validation score of voting classifier. We used f1-weighted for measuring scores. After performing 10 iterations, we got the mean score of 1.0.

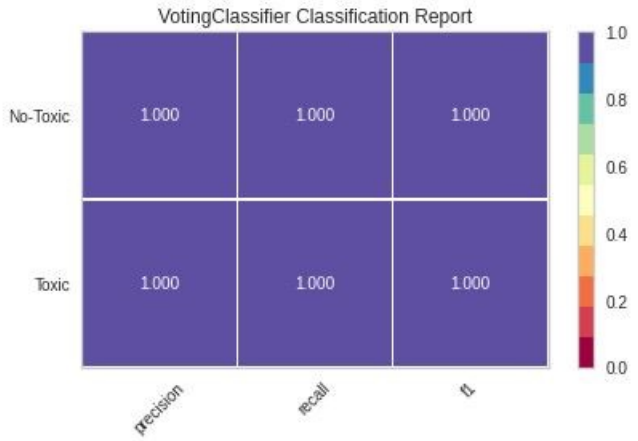


Figure 4.91: Classification report of Voting Classifier

Figure 4.91 shows the overall classification report of voting classifier. Here we measured that,the accuracy is 0.998, the precision is 0.997, the recall is 1.000 and the f1-score is 0.991.

Figure 4.92: LIME explanation of Voting Classifier

At last, we generate the LIME model as the XAI method in our ensemble model. We randomly select one toxic comment and one non-toxic comment. In figure 4.92, the model is 100% certain that the first comment is a good one. The percentage of positive comments is increased by the class values. Next, let's examine a negative comment. The model predicts that the comment is unfavorable 100% of the time, and the class values raise that probability.

Chapter 5

Conclusion

In this chapter, we summarize the entire report of our paper and also mention some limitations and future work in light of our work. That way, in the future, we can add new things or use other algorithms to get more accurate results.

5.1 Conclusion

In this study, we depict the classification of Bengali hate speech, which is an important matter, but it is further complicated by the difficulty of using climbing, network integration to intimidate operators, and wrong or harmful thoughts. However, by adding CNNs after natural language processing, including data cleaning, tokenization, and disassembly word removal, we aim to achieve the main accuracy of other current works. We combine CNN with Bi-LSTM to obtain a hybrid model. We used the hosted CNN-BiLSTM dataset for eight different machine learning classifiers and explained SHAP and LIME. We also show an in-depth comparison of our proposed model with other previously prepared transformer models. Finally, we combine these machine learning algorithms with voice classifiers to achieve pinpoint accuracy. We tried to convey a different calculation, different from others, which makes our work a classic. Being involved in online media production is our greatest asset, and access to education is not too far behind, as both are vulnerable to mass hate and abusive comments. It is also in our interest to apply the system in the near future to chat boxes on social media and educational platforms, which are subject to a flood of negative 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 large Bangla dataset and detecting toxic comment.
- Creating a large dictionary of Bengali hate word.
- A word tokenizer which will be effective for tokenizing comments that contain hidden profanity.
- We also intend to apply fusion model which may evolve transformer model with deep learning model for more precise result.

5.1.2 Scope and Limitations

Through this article, we have tried to establish a system that can categorize malicious comments in Bengali from any post on the web. The intensity of online media is so ultimately that it can make anything go viral in the blink of an eye. Certainly, one can find different types of individuals based on race, identity, sexual orientation, religion, etc. So, it is possible to effectively visualize the spread of any hateful or malicious substance through these steps. These can lead to major events such as uproar, attempts at self-destruction, even psychological warfare. Either way, all of this can be prevented at its root by depicting and acknowledging malicious or hateful content on the web. The number of customers on the web media is gradually increasing, and with the increasing number of customers, the number of tweets and messages is also increasing in the same way. Controlling and observing this huge number of comments is a colossal task. Besides the complexity of natural language and the new method of using scorn and malicious words, this type of testing is even more difficult.

Bibliography

[1] Y. Kim, "Convolutional neural networks for sentence classification," *arXivpreprint*, 2014. DOI: arXiv:1408.5882.

[2] S. Livingstone, L. Haddon, J. Vincent, G. Mascheroni, and K. Olafsson, "Children go mobile," *The UK report*, 2014.

[3] 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.

[4] T. Mowen, J. Brent, and A. Kupchik, "The handbook of measurement issues in criminology and criminal justice," *School crime and safety*, vol. 434, no. 2, 2016.

[5] 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*, pp. 120–121, 2016. DOI: 10.1109/i-Society.2016.7854192.

[6] Y. Sanjay and S. Sanyam, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 78–83. DOI: 10.1109/IACC.2016.25.

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

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

[9] F. Gurcan, "Multi-class classification of turkish texts with machine learning algorithms," *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies*, pp. 1–5, 2018. DOI: 10.1109/ISMSIT.2018.8567307.

[10] C. V. Hee, G. Jacobs, C. Emmery, *et al.*, "Automatic detection of cyberbullying in a social media text," vol. 13, no. 10, 2018.

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

[12] J. R. Saurav, S. Amin, S. Kibria, and M. S. Rahman, "Bangla speech recognition for voice search," *2018 International Conference on Bangla Speech and Language Processing*, pp. 1–4, 2018. DOI: 10.1109/ICBSLP.2018.8554944.

[13] X. She and D. Zhang, "Text classification based on hybrid cnn-lstm hybrid model," *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 185–189, 2018. DOI: 10.1109/ISCID.2018.10144.

[14] N. I. T. to and M. E. Ali, "Detecting multilabel sentiment and emotions from bangla youtube comments," *2018 International Conference on Bangla Speech and Language Processing*, pp. 1–6, 2018. DOI: 10.1109/ICBSLP.2018.8554875.

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

[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] S. Ahammed, M. Rahman, M. H. Niloy, and S. M. M. H. Chowdhury, "Implementation of machine learning to detect hate speech in bangla language," *2019 8th International Conference System Modeling and Advancement in Research Trends*, pp. 317–320, 2019. DOI: 10.1109/SMART46866.2019.9117214.

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

[19] Y. Feng-Jen, "An extended idea about decision trees," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 349–354. DOI: 10.1109/CSCI49370.2019.00068.

[20] M. Jahan, I. Ahamed, M. R. Bishwas, and S. Shatabda, "Abusive comments detection in bangla-english code-mixed and transliterated text," *2019 2nd International Conference on Innovation in Engineering and Technology*, pp. 1–6, 2019. DOI: 10.1109/ICIET48527.2019.9290630.

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

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

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

[24] Y. Luan and S. Lin, "Research on text classification based on cnn and lstm," *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 352–355, 2019. DOI: 10.1109/ICAICA.2019.8873454.

[25] F. Rahman, "An annotated bangla sentiment analysis corpus," *2019 International Conference on Bangla Speech and Language Processing*, pp. 1–5, 2019. DOI: 10.1109/ICBSLP47725.2019.201474.

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

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

[28] M. I. R. Shuvo, S. A. Shahriyar, and M. A. H. Akhand, "Bangla numeral recognition from speech signal using convolutional neural network," *2019 International Conference on Bangla Speech and Language Processing*, pp. 1–4, 2019. DOI: 10.1109/ICBSLP47725.2019.201540.

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

[30] C. Soufiane and T. Kouhyar, "Early prediction of sepsis from clinical data using single light-gbm model," in *2019 Computing in Cardiology (CinC)*, 2019, Page 1–Page 4. DOI: 10.23919/CinC49843.2019.9005718.

[31] Z. Xiaonan, H. Yong, T. Zhewen, and S. Kaiyuan, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139. DOI: 10.1109/ICCSNT47585.2019.8962457.

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

[33] A. Y. Ahmad, A. V. Elijah, B. A. Oluwagbemiga, and A. A. Kareem, "Ai meta-learners and extra-trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 8, pp. 142 532–142 542, 2020. DOI: 10.1109/ACCESS.2020.3013699.

[34] S. Enhui, S. Lin, X. Jiucheng, and Z. Shiguang, "Multilabel feature selection using mutual information and ml-relieff for multilabel classification," *IEEE Access*, vol. 8, pp. 145 381–145 400, 2020. DOI: 10.1109/ACCESS.2020.3014916.

[35] D. J, K. Y. Woon, and D. Dalia, "Comparison of gradient boosting and extreme boosting ensemble methods for webpage classification," in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2020, pp. 77–82. DOI: 10.1109/ICRCICN50933.2020.9296176.

[36] B. Junchen, "Multi-features based arrhythmia diagnosis algorithm using xgboost," in *2020 International Conference on Computing and Data Science (CDS)*, 2020, pp. 454–457. DOI: 10.1109/CDS49703.2020.00095.

[37] R. Parveen, N. Shrivastava, and P. Tripathi, "Sentiment classification of movie reviews by supervised machine learning approach using ensemble learning & voted algorithm," *2nd International Conference on Data, Engineering and Applications*, pp. 1–6, 2020. DOI: 10.1109/IDEA49133.2020.9170684.

[38] S. T. R. Ma and Z. Fu, "Text sentiment classification based on improved bilstm-cnn," *2020 Asia-Pacific Conference on Image Processing, Electronics, and Computers*, pp. 1–4, 2020. DOI: 10.1109/IPEC49694.2020.9115147.

[39] B. Z. Y. Z. S. Liang, S. Cheng, and J. Jin, "A double channel cnn-lstm model for text classification," *2020 IEEE 22nd International Conference on High-Performance Computing and Communications*, pp. 1316–1321, 2020. DOI: 10.1109/HPCC-SmartCity-DSS50907.2020.00169.

[40] K. Sajib, M. Raihan, A. Nasif, *et al.*, "Breast cancer risk prediction using xgboost and random forest algorithm," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–4. DOI: 10.1109/ICCCNT49239.2020.9225451.

[41] 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.

[42] M. M. Sultan, H. J. Zhexue, S. Salman, E. T. Z., and S. Kuanishbay, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85–101, 2020. DOI: 10.26599/BDMA.2019.9020015.

[43] F. Sun and N. Chu, "Text sentiment analysis based on cnn-bilstm-attention model," *2020 International Conference on Robots & Intelligent System*, pp. 749–752, 2020. DOI: 10.1109/ICRIS52159.2020.00186.

[44] R. N. Swarna, "Bangla broadcast speech recognition using support vector machine," *2020 Emerging Technology in Computing, Communication, and Electronics*, pp. 1–6, 2020. DOI: 10.1109/ETCCE51779.2020.9350865.

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

[46] W. Yue and L. Li, "Sentiment analysis using word2vec-cnn-bilstm classification," *2020 Seventh International Conference on Social Networks Analysis, Management and Security*, pp. 1–5, 2020. DOI: 10.1109/SNAMS52053.2020.9336549.

[47] S. Chutimet and K. Sivakorn, "Application of natural neighbor-based algorithm on oversampling smote algorithms," in *2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, 2021, pp. 217–220. DOI: 10.1109/ICEAST52143.2021.9426310.

[48] B. Erik, A. Vakil, L. Jia, and E. Robert, "Multimodal data fusion using canonical variates analysis confusion matrix fusion," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–10. DOI: 10.1109/AERO50100.2021.9438445.

[49] A. B. Gumelar, E. M. Yuniarno, D. P. Adi, A. G. Sooai, I. Sugiarto, and M. H. Purnomo, "Bilstm-cnn hyperparameter optimization for speech emotion and stress recognition," *2021 International Electronics Symposium*, pp. 156–161, 2021. DOI: 10.1109/IES53407.2021.9594024.

[50] W. Hao and Z. Xuping, "Modal dynamic modelling and experimental validation of a curved extensible continuum manipulator," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1958–1965. DOI: 10.1109/ICRA48506.2021.9561914.

[51] M. I. H. Junaid, F. Hossain, and R. M. Rahman, "Bangla hate speech detection in videos using machine learning," *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*, pp. 0347–0351, 2021. DOI: 10.1109/UEMCON53757.2021.9666550.

[52] M. R. Karim, "Deephateexplainer: Explainable hate speech detection in underresourced bengali language," *2021 IEEE 8th International Conference on Data Science and Advanced Analytics*, pp. 1–10, 2021. DOI: 10.1109/DSAA53316.2021.9564230.

[53] T. C. Kla and J. Jirayus, "Explainable ai for software engineering," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 1–2. DOI: 10.1109/ASE51524.2021.9678580.

[54] Y. I. Nurma, P. Erick, S. Asep, and N. Dessy, "Adaboost support vector machine method for human activity recognition," in *2021 International Conference on Artificial Intelligence and Big Data Analytics*, 2021, pp. 1–4. DOI: 10.1109/ICAIBDA53487.2021.9689713.

[55] Puneet, Deepika, S. Pritpal, B. Rahul, and S. Saket, "Coronary heart disease prediction using voting classifier ensemble learning," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 181–185. DOI: 10.1109/ICAC3N53548.2021.9725705.

[56] Sarkar, Ahamed, and Khan, "An experimental framework of bangla text classification for analyzing sentiment applying cnn & bilstm," *2021 2nd International Conference for Emerging Technology*, pp. 1–6, 2021. DOI: 10.1109/INCET51464.9456393.

[57] S. Siddharth, O. Nikita, and S. K. K., "An approach to identify captioning keywords in an image using lime," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 648–651. DOI: 10.1109/ICCCIS51004.2021.9397159.

[58] verma Vivek Kumar and K. Vinod, "Optimization of regression algorithms using learning curve in wsn," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, pp. 379–382. DOI: 10.1109/ICACITE51222.2021.9404686.

[59] Y. Wu and J. He, "Sentiment analysis of barrage text based on albert-att-bilstm model," *2021 4th International Conference on Pattern Recognition and Artificial Intelligence*, pp. 152–156, 2021. DOI: 10.1109/PRAI53619.2021.9551040.

[60] M. A. Zaenul, A. Sumarni, P. Yoga, and A. Yuli, "The effect of recursive feature elimination with cross-validation (rfecv) feature selection algorithm toward classifier performance on credit card fraud detection," in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, 2021, pp. 270–275. DOI: 10.1109/ICAICST53116.2021.9497842.

[61] H. U. A., "Tire changes, fresh air, and yellow flags: Challenges in predictive analytics for professional racing," *The Financial Express: Combating cyber-bullying of women*, 2022. [Online]. Available: https://thefinancialexpress.com.bd/views/combating%20-cyber-bullying-of-women-1650466631.

[62] S. S. Ahmadi and H. Khotanlou, "A hybrid of inference and stacked classifiers to indoor scenes classification of rgb-d images," pp. 1–6, 2022. DOI: 10.1109/MVIP53647.2022.9738755.

[63] M. G. R. Alam, "Shapley-additive-explanations-based factor analysis for dengue severity prediction using machine learning," *Journal of Imaging*, no. 8, p. 229, 2022. DOI: 10.3390/jimaging8090229.

[64] J. J. Bonny, N. J. Haque, M. R. Ulla, P. Kanungoe, Z. H. Ome, and M. I. H. Junaid, "Deep learning approach for sentimental analysis of hotel review on bengali text," *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies*, pp. 1–7, 2022. DOI: 10.1109/ICAECT54875.2022.9808001.

[65] D. Dhruti, C. Ling, and A.-M. A. M., "Long short term memory water quality predictive model discrepancy mitigation through genetic algorithm optimisation and ensemble modeling," *IEEE Access*, vol. 10, pp. 24 638–24 658, 2022. DOI: 10.1109/ACCESS.2022.3152818.

[66] "Hugging face – the ai community building the future.," 2022. [Online]. Available: https://huggingface.co/.

[67] M. I. H. Junaid, F. Hossain, U. S. Upal, A. Tameem, A. Kashim, and A. Fahmin, "Bangla food review sentimental analysis using machine learning," *2022 IEEE 12th Annual Computing and Communication Workshop and Conference*, pp. 0347–0353, 2022. DOI: 10.1109/CCWC54503.2022.9720761.

[68] M. Kazuki, K. Yuta, and N. Takayuki, "Simple and effective multimodal learning based on pre-trained transformer models," *IEEE Access*, vol. 10, pp. 29 821–29 833, 2022. DOI: 10.1109/ACCESS.2022.3159346.

[69] V. O. Khilwani, V. Gondaliya, S. Patel, J. Hemnani, B. Gandhi, and S. K. Bharti, "Diabetes prediction, using stacking classifier," *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pp. 1–6, 2022. DOI: 10.1109/AIMV53313.2021.9670920.

[70] R. G. Krishnan, W. Shaowei, O. G. A., K. Yasutaka, and H. A. E., "The impact of feature importance methods on the interpretation of defect classifiers," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2245–2261, 2022. DOI: 10.1109/TSE.2021.3056941.

[71] M. Sarker, M. F. Hossain, F. R. Liza, S. N. Sakib, and A. A. Farooq, "A machine learning approach to classify anti-social bengali comments on social media," *2022 International Conference on Advancement in Electrical and Electronic Engineering*, pp. 1–6, 2022. DOI: 10.1109/ICAEEE54957.2022.9836407.

[72] O. Sen, "Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, pp. 38 999–39 044, 2022. DOI: 10.1109/ACCESS.2022.3165563.

[73] S. Sultana, M. Z. Iqbal, M. R. Selim, M. M. Rashid, and M. S. Rahman, "Bangla speech emotion recognition and cross-lingual study using deep cnn and blstm networks," *IEEE Access*, vol. 10, pp. 564–578, 2022. DOI: 10.1109/ACCESS.2021.3136251.

[74] E. Xu, D. Qin, J. Huang, and J. Zhang, "Multi text classification model based on bret-cnn-bilstm," *2022 IEEE 5th International Conference on Big Data and Artificial Intelligence*, pp. 184–189, 2022. DOI: 10.1109/BDAI56143.2022.9862653.

[75] W. Yuchen, "Personality type prediction using decision tree, gbdt, and cat boost," in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, 2022, pp. 552–558. DOI: 10.1109/BDICN55575.2022.00107.

[76] M. F. Ahmed, "Bangla online comments dataset," *Mendeley Data*, vol. 1, DOI: 10.17632/9xjx8twk8p.

[77] 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,

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

[79] *Toxic comment classification challenge*. [Online]. Available: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data.