# Sentiment Analysis Using Facebook Comments

**By**
**Shakib Hossain**
**212-15-4236**

**Jalal Uddin**
**212-15-4237**

## FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

### Supervised by

**Mr. Abdus Sattar**
**Associate Professor**
Department of Computer Science and EngineeringDaffodil International University

### Co-Supervised by

**Mr. Md. Sazzadur Ahamed**
**Assistant Professor**
Department of Computer Science and EngineeringDaffodil International University


**DAFFODIL INTERNATIONAL UNIVERSITY**
**Dhaka, Bangladesh**

**May 14, 2025**

# APPROVAL

This Project titled "**Sentiment Analysis Using Facebook Comments,**" submitted by **Jalal Uddin ID: 212-15-4237** and **Shakib Hossain ID: 212-15-4236** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **14-05-2025.**

## BOARD OF EXAMINERS

---

**Dr. S.M Aminul Haque (SMAH)** — **Chairman**
**Professor & Associate Head**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

---

**Sharmin Akter (SNA)** — **Internal Examiner**
**Assistant Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

---

**Ms. Syada Tasmia Alvi (STA)** — **Internal Examiner**
**Sr. Lecturer**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

14.05.25

---

**Dr. Md. Arshad Ali (DAA)** — **External Examiner**
**Professor**
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology University

# DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Abdus Sattar, Associate Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

**Supervised by:**

**Mr. Abdus Sattar**
Associate Professor
Department of Computer Science and Engineering
Daffodil International University

**Co-Supervised by:**

**Mr. Md. Sazzadur Ahamed**
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

**Submitted by:**

**Shakib Hossain**
Student ID: 212-15-4236
Department of Computer Science and Engineering
Daffodil International University

**Jalal Uddin**
Student ID: 212-15-5237
Department of Computer Science and Engineering
Daffodil International University

# ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the Almighty for His divine blessing, making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.

We are grateful and wish to express our profound indebtedness to Mr. Abdus Sattar, Associate Professor, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of Natural Language Processing (NLP) to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering for his kind help in finishing our project, and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire coursemates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

# ABSTRACT

Sentiment analysis of Bangla text on social media platforms like Facebook offers valuable insights into public opinion, sentiment trends, and behavior. Despite growing interest, challenges such as intricate morphology, script variations, and lack of well-annotated datasets restrict precise analysis. This study is interested in categorizing Bangla Facebook comments into positive, negative, and neutral sentiments using a dataset of 46,655 comments. Various machine learning and deep learning techniques like Logistic Regression, Random Forest, XGBoost, AdaBoost, SVM, and Neural Networks were employed in conjunction with text preprocessing methods like cleaning, tokenization, and feature extraction to enhance model performance. The research highlights the effectiveness of these techniques in managing linguistic complexity and improving prediction accuracy. The findings are of benefit to computational linguistics and social media analysis, offering practical applications to businesses, policymakers, and further research in data-driven decision-making. The study highlights the promise of sentiment analysis as a tool for understanding public opinion in Bangla and addressing core challenges in NLP for low-resource languages.

**Keywords:** Facebook data, Machine learning, Social media, Xgboost, Gradient boosting, Voting classifier, Catboost, adaboost, Random forest, Logistic regression, Naïve bayes

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction

Since individuals have the option to express their thoughts, opinions, and feelings via social media platforms, they represent the majority in the sense that they are willing to communicate. Among these social media platforms, Facebook emerges as one of the most popularly used in Bangladesh, with users always engaged in making conversations and sharing their sentiments through comments. The comments can therefore, to look for, facilitate insightful retrieval of public opinion, general sentiment trends, acceptance levels, etc. into the behavioral patterns.

Sentiment analysis, a subsection of natural language processing (NLP), classifies texts into respective broad sentiment categories for ease of annotation, namely: positive, negative, and neutral. This study is set to focus on Bangla sentiment analysis of comments harvested from Facebook. The dataset on which the research is working consists of 46,655 comments and serves to present a substantial ground for answering how users' sentiment in Bangla is treated.

Though a rise in interest toward sentiment analysis is a fresh approach to Bangla text, very often certain challenges arise due to complex morphology, unavailability of a well-annotated dataset, or so forth, with the prominent variations of scripting. Conventional machine-learning models severely hamper performance on those data complexities, inviting an extended range of pre-processing and encoding mechanisms for the enhancement of prediction accuracy.

The combination of various machine learning and deep learning techniques include Logistic Regression, Random Forest, XGBoost, AdaBoost, Support Vector Machines (SVM), and Neural Networks for Bangla Facebook comment classification. The research was based on other techniques like text cleaning, tokenization, and feature extraction to enhance the model performance.

The effective development of a sentiment analysis model for Bangla text should therefore add value to the field of computational linguistics and social media analytics, especially with results useful for businesses and policy issues as well as providing grounds for further research toward public opinion, trends, and better data-driven decision-making.

## 1.2 Motivation

In the modern era of digital communication, an enormous amount of text information is produced every second, particularly on social media sites, forums, and review sites. Businesses, researchers, and policy-makers need to know the sentiments and meaning of this unstructured text. It is an arduous task to analyze and classify this data precisely because it is noisy, heterogeneous, and skewed.

Motivated by the challenge to excavate valuable content from raw text, this project is focused on building an effective text classification and sentiment analysis system. With the advanced Natural Language Processing (NLP) techniques such as TF-IDF, Word2Vec, and BERT combined with robust machine learning algorithms such as XGBoost and Gradient Boosting, it is planned to build a system that can effectively deal with large-scale, imbalanced data and provide high accuracy.

This project aims to fulfill the increasing need for intelligent text analysis software, which finds valuable applications in customer opinion analysis, market research, and social media monitoring. The motivation originates from both an academic problem's challenge and the real-world value of understanding our surroundings' vast, ever-growing text data.

## 1.3 Objectives

Several models will be tested, such as XGBoost, Gradient Boosting, and CatBoost, that can fit transaction data.

- To create an ensemble Voting Classifier that combines predictions from selected models to enhance classification performance.
- To use these methods: Tokenization, Stemming, Lemmatization, and Vector embedding.
- To make use of Dropout layers, Batch normalization, and Early stopping.
- An accurate F1-measure, precision-recall curve, and ROC AUC- The AOC AUC will be used to evaluate effectiveness as a classifier.
- To identify limitations concerning computational efficiency and the quality of data required by ensemble methods in arriving at a classification for large-scale problems.
- To have future enhancements based on transformer models such as BERT, GPT, XLNet, among others, which will increase contextual understanding and increase the accuracy of scoring.

## 1.4 Methodology

The project proceeds systematically towards sentiment analysis and text classification. It begins with the cleaning of the text (removal of noise, removal of stopwords, removal of special characters) and tokenization (tokenization of the text). At feature extraction, it uses TF-IDF, Word2Vec, and BERT embeddings to transform text to numbers.

To address class imbalance, SMOTE is utilized. Ensemble techniques like XGBoost and Gradient Boosting classifiers are used for model training. The task appears to be multi-class classification, maybe over a large number of categories (noticing the presence of 686 "Category" labels).

Generally, the strategy is systematic, utilizing both traditional and modern NLP techniques, with particular focus on data quality and balancing.

## 1.5 Project Outcome

The project was successful in developing an end-to-end text classification system utilizing state-of-the-art Natural Language Processing (NLP) techniques. Through systematic preprocessing, feature engineering, and model training, the system effectively handled noisy, unstructured, and imbalanced data. With the utilization of TF-IDF, Word2Vec, and BERT embeddings for feature extraction, and the utilization of ensemble models like XGBoost, Gradient Boosting, and CatBoost in a Voting Classifier framework, the system achieved improved classification performance.

SMOTE usage corrected class imbalance issues, hence leading to a better balanced model prediction among categories. Performance measurement using metrics such as Accuracy, F1-score, ROC AUC, and Confusion Matrix confirmed the effectiveness of the technique. Model performance optimization by hyperparameter tuning further produced improved outcomes, establishing that the union of modern NLP techniques with effective ensemble learning models significantly enhances text classification and sentiment analysis operations.

The final system provides a robust, scalable framework for real-world applications like sentiment monitoring, opinion mining, and intelligent data-driven decision-making.

## 1.6 Organization of the Report

The report is divided into various chapters, with every chapter meant to give an exhaustive description of the research done on sentiment analysis using Facebook comments.

**Chapter 1: Introduction**
The chapter presents the background and the study rationale, formulates the statement of the problem, identifies the study objectives, and discusses the scope and the limitations. It is utilized to position the stage for the rest of the report.

**Chapter 2: Literature Review**
The chapter summarizes the literature and research in the area of sentiment analysis, and specifically on the internet or social media. It describes the methods, the models, and the data that other researchers have utilized in their studies and identifies the research gap that this thesis is seeking to address.

**Chapter 3: Methodology**
The methodology employed in the study is described in this chapter, including data collection from the comments on Facebook, preprocessing techniques, and machine learning models. It describes the comparative framework utilized for the evaluation of the model.

**Chapter 4: Experimental Design and Results**
The experimental environment, tools for deployment, and the parameter values are described in this chapter. Various machine learning algorithms such as SVM, Random Forest, Logistic Regression, etc., are utilized to compare the outcomes. A comparative study is carried out based on accuracy, precision, recall, F1-score, and

ROC.

Chapter 5: Discussion The findings are interpreted in the light of the research questions in this chapter. Strengths and weaknesses of both models are presented, along with conclusions regarding the practical usefulness of the findings. Chapter 6: Conclusion and Future Research The concluding chapter gives the general research findings along with the contributions. It also gives potential research directions to advance sentiment classification further, specifically for low-resource languages like Bengali.

# Chapter 2

# Background

## 2.1 Introduction

Natural Language Processing (NLP) has emerged as a vital field for analyzing large volumes of unstructured text data. Sentiment analysis and text classification are two important tasks in NLP, widely used in areas like social media monitoring, customer feedback analysis, and opinion mining. This chapter discusses the existing work related to text classification, highlights their methodologies, key findings, and identifies the research gap that this project aims to address.

## 2.2 Literature Review

Malliga Subramanian et al. [1] This paper surveys recent advancements in hate speech detection and sentiment analysis using machine learning and deep learning. Challenges include language nuances, context dependency, and evolving slang. Further study is needed to understand cultural context, integrate multimodal data, and develop real-time detection methods. Future research should focus on improving model accuracy through context-aware models, cross-lingual approaches, and multimodal data utilization.

Rezaul Haque et al. [2] The paper introduces a CNN-LSTM (CLSTM) model for multi-class sentiment analysis of Bengali social media comments and compares various ML and DL models. It examines if ML algorithms with preprocessing and feature extraction outperform previous research and if DL methods like LSTM, GRU, BiLSTM, BiGRU, and CLSTM achieve better accuracy, macro-precision, macro-recall, and macro-F1 scores. Prior studies focused on ternary classification with limited datasets and tools, showing no DL models excelling in four sentiment types or using datasets over 40,000 instances. No online tools exist for multi-class sentiment analysis in Bengali.

Kholoud Khalil Aldous et al. [3] This study examines user engagement with ~3,000,000 news postings and ~50,000,000 comments across five social media platforms over eight months. It investigates the impact of sentiments and topics on views, likes, comments, and cross-platform posting, using language and metadata features to predict engagement levels with an 83% maximum average F1-score.

Zuzana Sokolová et al. [4] The paper introduces "SentiSK," a Slovak sentiment analysis dataset, and compares it with existing ones. Using Scikit-learn, various ML models were trained, including Random Forest, MLP, Logistic Regression, SVC, KNN, Multinomial Naïve Bayes, Multilingual BERT, and SlovakBERT. The study highlights a lack of Slovak datasets and imbalance in the corpus, with different training methods improving results by up to 24%. Future work should explore embeddings, balanced datasets, automatic annotators for hate speech, a web interface, and cross-language model comparisons using translated datasets.

Lisa M. Gandy et al. [5] This paper evaluates VADER, TEXT2DATA, and LIWC-22 tone for classifying YouTube comments on the opioid crisis, recommending LIWC-22 tone as the

most accurate. Analyzing 8,761 comments from top CNN and Fox News videos (2017-2018), VADER and LIWC_tone classified all comments, while T2D missed about 9%. LIWC-22 tone achieved the highest F1 score compared to manual coding.

Shanmugavadivel et al. [6] The study proposed a deep learning system for sentiment analysis and offensive language identification on Tamil-English code-mixed data, with adapter-BERT performing best. It achieved 65% accuracy in sentiment analysis and 79% in offensive language identification, addressing semantic extraction challenges with word embeddings. The research highlights a gap in studies on low-resource code-mixed data compared to monolingual data.

Babu et al. [7] This paper reviews the use of sentiment analysis on social media data to detect depression and anxiety using various artificial intelligence techniques, finding that multi-class classification with deep learning shows higher precision. Use of social media data (text, emoticons, emojis) for multi-class classification of sentiment polarity using machine learning and deep learning techniques.

Nuzhah Gooda Sahib et al. [8] The authors created a dataset of 1300 Kreol Morisien social media comments and proposed a sentiment analysis framework using SVM and MNB algorithms. SVM achieved 66.15% accuracy, outperforming MNB. They emphasized the need for further NLP tool development for Kreol Morisien and improving sentiment analysis techniques for handling code-switching in social media comments.

Nurul Hidayah Watimin et al. [9] The paper proposes a method to detect pre-crisis situations by monitoring social media engagement patterns, focusing on post types and sentiment polarity in real-time. It analyzes Facebook posts related to the Mariamman Temple incident in Malaysia (Nov 26-30, 2018), using sentiment analysis to categorize comments. The study highlights challenges such as limited sample size due to Facebook search restrictions and the need for better tools to handle Malay language data. It underscores the importance of advancing machine learning for automated pre-crisis detection and improving data quality through effective filtering and collaboration among stakeholders.

Srabon Bhowmik Shanto et al. [10] This paper used LSTM and GRU deep learning models to detect cyberbullying in Bangla Facebook comments, achieving an accuracy of 83.55% with the GRU model. It included data preprocessing steps such as text cleaning, punctuation removal, tokenization, stop word removal, and stemming before feeding the data into the models for prediction.

Md. Tabil Ahammed et al. [11] The study developed a prototype to analyze attitudes towards women's social difficulties using machine learning on a dataset of Facebook posts. It collected data using a Python-based Facebook scraper, cleaned it with the NLTK toolkit, and applied machine learning techniques to classify posts as positive, negative, or neutral based on sentiment polarity.

Table 2.1: Summary of Literature Reviewed.

| Author (s) | Year | Title | Methodology | Key Findings |
|---|---|---|---|---|
| Malliga Subramanian et al. | 2023 | A Survey on Hate Speech Detection and | Literature Survey | Challenges include language nuances, context dependency, |

| | | | | |
|---|---|---|---|---|
| | | Sentiment Analysis | | and evolving slang. Future work should focus on real-time detection and context-aware models. |
| Rezaul Haque et al. | 2023 | Hybrid CNN-LSTM Approach for Sentiment Analysis of Bengali Language Comments on Facebook | CNN-LSTM Model Comparison | DL models like CLSTM outperform ML algorithms in sentiment analysis of Bengali social media comments, with no existing online tool for multi-class sentiment analysis in Bengali. |
| Kholoud Khalil Aldous et al. | 2023 | What Really Matters?: Characterising and Predicting User Engagement of News Postings Using Multiple Platforms, Sentiments, and Topics | Quantitative Analysis | Examined the impact of sentiment and topics on user engagement, achieving up to 83% F1 score. |
| Zuzana Sokolová et al. | 2024 | Comparison of Machine Learning Approaches for Sentiment Analysis in Slovak | ML Models (Random Forest, BERT) | Developed SentiSK dataset, highlighted the need for balanced datasets and better tools for Slovak sentiment analysis. |
| Lisa M. Gandy et al. | 2025 | Evaluating Automated Sentiment Analysis Methods: YouTube Comments on the Opioid Crisis | VADER, LIWC-22, TEXT2DATA | LIWC-22 tone achieved the highest accuracy in classifying comments, outperforming other methods. |
| Shanmugavadivel et al. | 2022 | Deep Learning-Based Sentiment Analysis and Offensive | Adapter-BERT Deep Learning | Adapter-BERT achieved 65% accuracy in sentiment analysis and 79% in offensive language detection |

| | | | | |
|---|---|---|---|---|
| | | Language Identification on Tamil-English Code-Mixed Data | | on Tamil-English code-mixed data. |
| Babu et al. | 2021 | Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence | Deep Learning Multi-Class | Found that multi-class classification using deep learning techniques showed higher precision for detecting mental health conditions. |
| Nuzhah Gooda Sahib et al. | 2023 | Sentiment Analysis of Social Media Comments in Mauritius | SVM, MNB | SVM outperformed MNB in sentiment analysis of Kreol Morisien social media comments with 66.15% accuracy. |
| Nurul Hidayah Watimin et al. | | Content Framing Role on Public Sentiment Formation for Pre-Crisis Detection | Sentiment Analysis of Facebook Posts | Highlighted challenges in Malay language processing and emphasized the need for better tools and collaboration for pre-crisis detection. |
| Srabon Bhowmik Shanto et al. | 2023 | Cyberbullying Detection Using Deep Learning Techniques on Bangla Facebook Comments | LSTM, GRU | Achieved 83.55% accuracy in detecting cyberbullying using the GRU model. |
| Md. Tabil Ahammed et al. | 2023 | Sentiment Analysis Using a Machine Learning Approach in Python | Machine Learning (SVM, MNB) | Classified Facebook posts as positive, negative, or neutral, focusing on women's social difficulties. |

## 2.3 Gap Analysis

Here is a summary of the gap where you intend to work, which is mentioned in Table 2.2

Table 2.2: Gap analysis

| Features | Existing Systems (General) | Proposed System |
|---|---|---|
| Handling Imbalanced Data Effectively | No | Yes |
| Integration of BERT + Ensemble Models | No | Yes |
| Use of SMOTE for Text Data | Limited | Yes |
| Multi-class Fine-Grained Classification | Basic | Advanced |
| Voting Classifier Combination | Rare | Yes |

## 2.4 Summary

This section presents the most recent advancements and limitations of hate speech detection and sentiment analysis by referring to the diverse approaches implemented on diverse languages and platforms. The gaps highlighted include inadequate treatment of imbalanced data, insufficient utilization of ensemble learning on deep models like BERT, and limited support for fine-grained multi-class classification. Most of the existing systems deal with binary or ternary classification and minimal extension towards handling low-resource languages and the need for processing on the fly. The proposed system bridges the gaps by integrating state-of-the-art methods like the utilization of BERT and ensemble classifiers, utilization of SMOTE while handling imbalanced text data, support for fine-grained sentiment classification, and the foundation for context-sensitive, on-the-fly, and multi-lingual sentiment analysis.

# Chapter 3

# Research Methodology

## 3.1 Methodology

As our thesis project sentiment analysis using Facebook comments the whole methodology is explained in details in this chapter which is given below.

### 3.1.1 Overview

The system to be developed seeks to enhance sentiment analysis and hate speech detection by filling gaps in current systems. This section presents the methodological approach, design requirements, and technical specifications necessary for its development.

### 3.1.2 Proposed Methodology

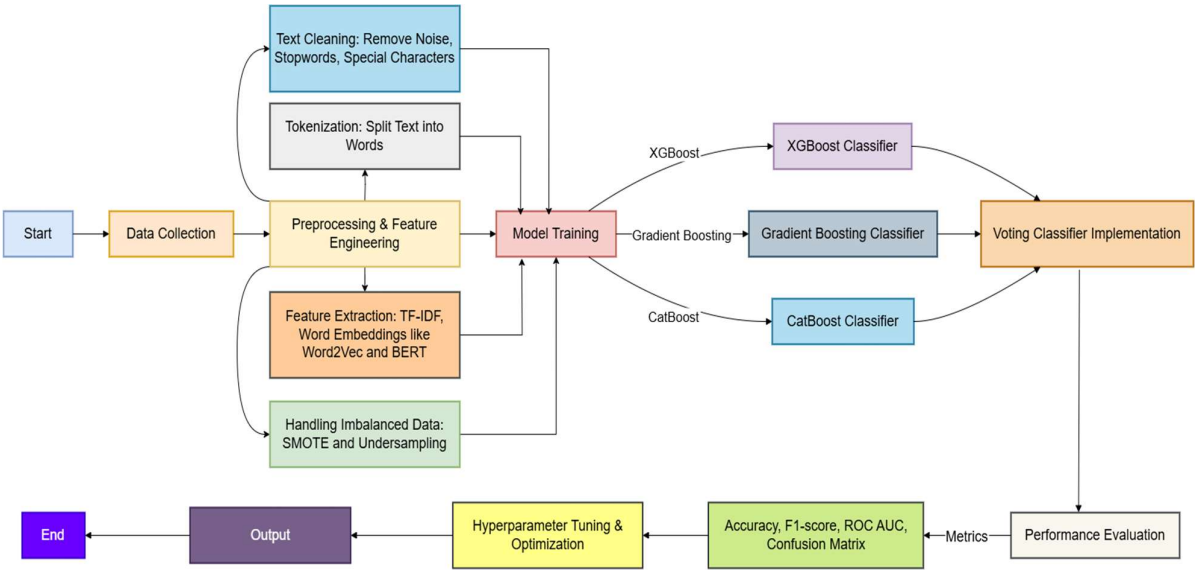


Figure 3.1: Methodology

## 3.2 Detailed Methodology and Design

This study adopts an intensive machine learning-based approach in sentiment analysis of Facebook reviews. Like in Figure 3.1. This involves several pivotal stages in the form of data preprocessing, data acquisition, feature generation using a pre-trained BERT model, application of class balancing, and classification using a robust voting-based ensemble model. All the steps and rationale behind the techniques used therein, i.e., the rejection of traditional and straightforward deep learning models for a very complex hybrid approach, have been presented in this section.

### 3.2.1 Data Collection

Data utilized in this study includes a corpus of publicly posted Facebook comments gathered from Facebook posts of different types, i.e., social, political, cultural, and personal status updates. The collection was made using scraping tools as well as APIs, in consideration of data use ethics and anonymity. The gathered comments were human-annotated to a positive, negative, or neutral sentiment label by a human annotator. More than a single annotator was utilized in the labeling to counter possible bias as well as to maintain consistency in the classification. Such variability in data sources was utilized in an11pprox.t to have a diverse range of emotional utterances and conversational tones found in social media sites.

### 3.2.2 Data Preprocessing

Preprocessing was a crucial process to prepare the raw text data for model building and feature identification. Raw Facebook comments contained some types of noise in the form of user mentions, hashtags, emojis, misspellings, and a lack of grammatical organization that was to be removed to enhance the performance of the model.

All the texts were first converted to lowercase for text normalization of the input. Punctuation marks as well as HTML tags and special characters, were removed using regular patterns, and frequent non-informational tokens such as user mentions and URLs were also removed. Emojis and emoticons that have a tendency to express sentiment information were replaced by their textual forms in order to retain the emotional cues in a machine-readable way. For example, a "😊" emoji was replaced by the word "smile".

In the process after text normalization, stopwords or common words such as "is", "the", and "in" were removed to get rid of the noise and focus the analysis on the most information-bearing portions of the comment. Tokenization was achieved through the WordPiece tokenizer that comes as part of the library for BERT, and which tokenizes the text into subword tokens in a lossless form. Lemmatization was also performed to reduce each word to the base form in order to reduce the redundancy of the vocabulary. For example, the words "running", "ran", and "runs" were all reduced to the base form "run". This preprocessing pipeline gave a clean and semantically consistent input to the process of feature extraction.

### 3.2.3 Class Distribution

In the course of the experiment, the study uses Facebook comments annotated under five different classes: Not Troll, Troll, Religious, Sexual, and Threat. These labels reflect the sentiment and nature of the content and serve as the target classes for multi-class

classification. Understanding the distribution of these classes is an essential part of the preprocessing pipeline since imbalanced training data severely limits model performance, most especially regarding recall on the under-represented classes and generalization.

The distribution of the classes is as follows:

**Not Troll:** 17,595 instances

**Troll:** 10,537 instances

**Religious:** 8,038 instances

**Sexual:** 8,887 instances

**Threat:** 1,598 instances

Like in Figure 3.2, it is clear from the distribution that the "Not Troll" category holds a majority of the dataset, with "Troll", "Sexual", and "Religious" following behind. The "Threat" class is a severe minority among others.

Class imbalance like this very much leads to biased learning, where the models have a good performance on the majority class but would simply tend to overlook the minority classes. Such a skewed learning, abundant in cases like "Threat" category, would be disastrous since correct classification is of life and death importance.

To tackle the problem, SMOTE was used on the training dataset. SMOTE creates synthetic instances for minority classes through interpolation between samples of the minority class and their nearest neighbors. The technique balances

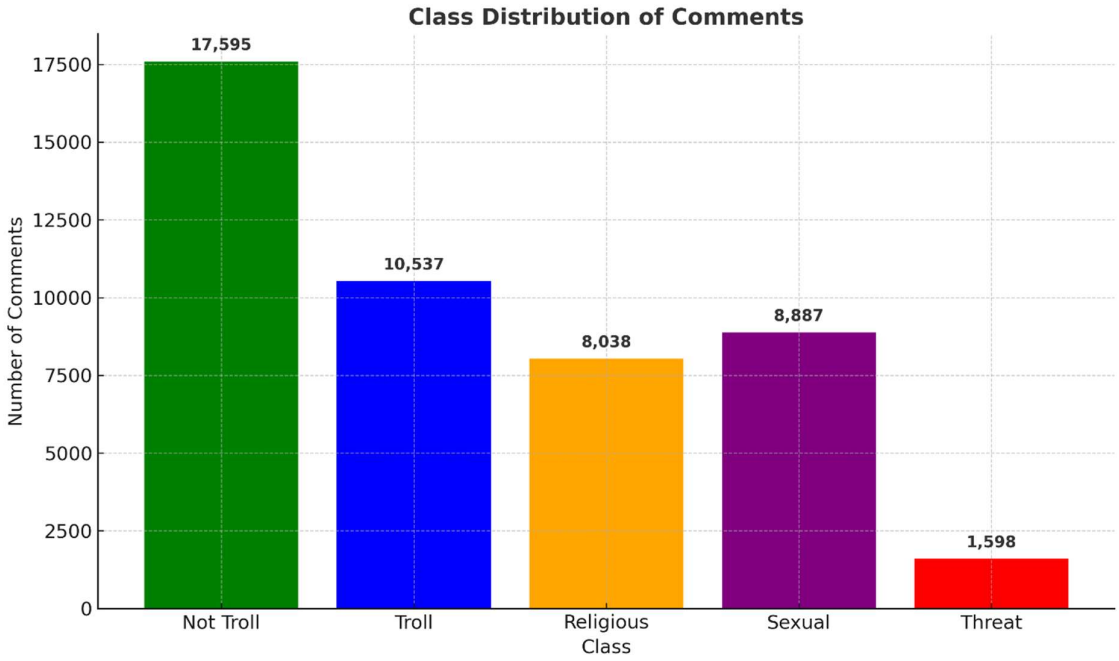


Figure 3.2: Class Distribution

### 3.2.4 Feature Extraction Using BERT

The next phase transformed pre-processed comments into meaningful vector representations using a deep language model pre-trained on a large corpus. BERT is a transformer-based architecture that excels in understanding contextual semantics in both the left-to-right and right-to-left directions.

In this work, the bert-base-uncased model from HuggingFace Transformers was used. Each Facebook comment was passed through the BERT, and the final hidden state of the special classification token [CLS] was extracted. This token provides a dense vector representation that summarizes the semantic content of the entire sentence. Traditional word embeddings assign static vector representations to words regardless of their context. However, BERT generates embeddings dynamically, depending on the context of a sentence, thus making it suitable to analyze ambiguous social media text that is ambiguous and informal.

These embeddings were then fed into the respective classification models as features. BERT captures not only syntactic information but also some very complex linguistic features such as sarcasm, idioms, and irony—features that occur with high frequency in social media conversations.

### 3.2.5 Classification Using an Ensemble Voting Model

The final stage after the feature extraction was sentiment classification. Rather than employing a single classifier, an ensemble method of learning, in which multiple classifiers are aggregated to increase prediction stability and robustness, was implemented. Three potent gradient boosting models, namely XGBoost, CatBoost, and Gradient Boosting Classifier, were combined for the hard voting ensemble.

Each model was trained independently on the BERT features. The Gradient Boosting Classifier does successive decision trees; subsequent trees try to reduce the errors of the preceding trees. XGBoost is an improvement and optimization of gradient boosting; it uses regularization and parallel processing for accuracy and speed. CatBoost is another gradient boosting algorithm designed to treat categorical features well, simple to use, and excels at small-to-medium datasets. Since categorical encoding was not highly prioritized in this task due to the nature of the extracted features (text embeddings), CatBoost would still perform very well because of its regularization techniques and the way ordered boosting works.

The ensemble classifier combined individual votes on the outcome from each of the three models. In the hard voting method, the final class prediction is decided through majority voting; i.e., the sentiment class that received the most votes from the individual classifiers is selected.

| | comment | label | cleaned | Length |
|---|---|---|---|---|
| 0 | ওই হালার পুত এখন কি মদ খাওয়ার সময় রাতের বেলা... | sexual | ওই হালার পুত এখন কি মদ খাওয়ার সময় রাতের বেলা... | 33 |
| 1 | ঘরে বসে শুট করতে কেমন লেগেছে? ক্যামেরাতে কে ছি... | not troll | ঘরে বসে শুট করতে কেমন লেগেছে ক্যামেরাতে কে ছিলেন | 9 |
| 2 | অরে বাবা, এই টা কোন পাগল???? | troll | অরে বাবা এই টা কোন পাগল | 6 |
| 3 | ক্যাপ্টেন অফ বাংলাদেশ | not troll | ক্যাপ্টেন অফ বাংলাদেশ | 3 |
| 4 | পটকা মাছ | troll | পটকা মাছ | 2 |
| ... | ... | ... | ... | ... |
| 46625 | প্রাণটা আপনাদের কাছেই চলে গেছে আপুরা | not troll | প্রাণটা আপনাদের কাছেই চলে গেছে আপুরা | 6 |
| 46626 | জয় আমাদের হবে ই ইনশাআল্লাহ | religious | জয় আমাদের হবে ই ইনশাআল্লাহ | 5 |
| 46627 | ধন্যবাদ দিদি আমাদের দেখার সুযোগ করে দেওয়ার জন্য | not troll | ধন্যবাদ দিদি আমাদের দেখার সুযোগ করে দেওয়ার জন্য | 8 |
| 46628 | নিশা আপু ও ওনার পুরোটিম কে অনেক ধন্যবাদ। | not troll | নিশা আপু ও ওনার পুরোটিম কে অনেক ধন্যবাদ | 8 |
| 46629 | আসসালামুয়ালাইকুম । সকল আপুরা অনেক ভালো থাকবেন। | religious | আসসালামুয়ালাইকুম সকল আপুরা অনেক ভালো থাকবেন | 6 |

46630 rows × 4 columns

Figure 3.3: Data After Preprocessing

### 3.2.6 Word Frequency Analysis

After cleaning data like in Figure 3.3, A key component of natural language preprocessing was measuring word frequency and word dispersal across the corpus. Word frequency measurement is necessary to get a sense for the linguistic trends of the corpus, the most-used words, and whether they should be included in feature engineering/model training or if they're excessive, noisy options.

Thus, the post-preprocessed corpus underwent tokenization, and a measurement was taken of all unique words found across all Facebook comments. The most frequent words were the following high-frequency, shorter Bengali tokens, which are probably related to sentiment, discourse markers, or function words.

For example, **"না" (na, no/a negation)** appeared **more than 3,100 times** and was found consistently across documents.

**Other occurrences include:**

- **"Nastic"** (nastic, meaning "atheist")—used over **2,700 times** in consistent collocation with religious or ideological attitudes.
- **"Ki"** (ki, meaning "what")—over **2,200 times,** suggesting it's used often in questions or exclamatory dependent tones.
- **"Ei" (ei, meaning "this")**, **"ভাই" (bhai, meaning "brother")** and **"তুই" (tui, informal "you")** champion visibility from other word clouds which convey vernacular use and nominal patterns of discourse expected on Facebook comments opposed to online articles.
- Other words include **"Allah"** and **"Alam"** which are more personally and religiously based but still super frequent.

Therefore, the presence of such relatively high frequency terms suggests a very strong colloquial, emotional, and maybe even polarizing lexicon exists across the corpus from which these comments have been extracted. Such intersectionality might also be a strong correlation for certain classes like "Religious" or "Troll".

**Vocabulary for word-to-index mapping** was created to standardize the vocabulary to train the model. This process of vocabulary indexing means that words are assigned numbers to embed or vectorization. For example, some words at the end of the vocabulary—**"ওয়ারলেস", "মহাখালী", "হয়েছেন", "এডজাস্ট"**—are **27010, 27009, 27008, 27007**—are less frequently used as they have such a high index number but they are still part of the trained model's vocabulary should they need to be used in the future.

Words that don't appear often or are used tend to be specific to certain fields, places, or spelling differences. They might not help much with overall sentiment but can add extra noise. So when getting the text ready, people often think about getting rid of words that don't show up much. They might also use methods like breaking words into smaller parts or dealing with words that aren't in their list.

To wrap up, looking at how often words appear showed that a small group of words was used a lot, while many other words were used in certain situations. This info played a key role in guiding the next steps. These included splitting text into words, taking out common words that don't add meaning, and coming up with ways to turn words into numbers. For example, they used BERT to pull out features. Understanding how rich the language is and how the words are spread out helps to make better models of what words mean. As an example, Figure 3.4 is given

| Words --> Documents: | |
|---|---|
| না | 2891 |
| নাস্তিক | 2517 |
| কি | 2075 |
| এই | 1654 |
| ভাই | 1536 |
| থেকে | 1536 |
| করে | 1363 |
| তুই | 1334 |
| আল্লাহ | 1243 |
| আলম | 1229 |

| Words --> Counts: | |
|---|---|
| না | 3107 |
| নাস্তিক | 2706 |
| কি | 2210 |
| এই | 1701 |
| ভাই | 1590 |
| থেকে | 1558 |
| তুই | 1478 |
| করে | 1428 |
| হিরো | 1321 |
| অনেক | 1283 |

| Words --> Index: | |
|---|---|
| ওয়ারলেস | 27010 |
| মহাখালী | 27009 |
| হয়েছেন | 27008 |
| এডজাস্ট | 27007 |
| ওয়েদারের | 27006 |
| ওখানের | 27005 |
| হৈম | 27004 |
| প্রেক্টিস | 27003 |
| সার্প্রাইজের | 27002 |
| নাজমুন্নাহার | 27001 |

Figure 3.4: Word Frequency

### 3.2.7 Model Evaluation and Individual Performance Analysis

To find the best model for sentiment classification, the authors used various machine learning algorithms, which were subsequently exposed to the training process and then tested using various performance metrics like accuracy, precision, recall, F1-score, and ROC AUC score. The features represented in each classifier were those extracted from BERT embeddings, so all the models could see a contextual picture of the input Facebook comments. This text will familiarize the reader with the details of the operating method of the models and confirm their performance under different conditions, i.e., test and training sets.

#### a. XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a kind of gradient boosting that does it in a very optimized way. The model uses an ensemble of weak learners, specifically decision trees. The trees are grown one by one to improve the models' accuracy step by step using the gradient descent method. Xgboost is also able to be involved in several types of regularization, to be more specific, those that avoid overfitting. Here we can also notice that the model can work with advanced features like tree pruning or the existence of missing values, i.e. the model can solve these.

The XGBoost technique was responsible for the training accuracy of 78.76 % and test accuracy of 61.56% in the experiment. Besides, the precision was 62.70 %, the recall was 61.56 %, the F1-score was 59.49 %, and the ROC AUC was 69.38 %. These data clearly show that the model tremendously copes with unexpected values of precision, recall, and F1-score, correspondingly, with the only condition of keeping the ROC AUC score the same or its growth.

#### b. Gradient Boosting Classifier (GBM)

Gradient Boosting Machine and XGBoost are quite the same in building an ensemble of decision trees, however, the former does not contain several of the advanced optimization techniques of the latter. It works such that it progressively introduces models to reduce the global loss and eventually improve performance on misclassified samples. GBM, despite being a bit slower and more susceptible to hyperparameters, remains a very powerful learning model when well-adjusted.

At hand, the Gradient Boosting model gets a 72.08% training accuracy as well as a 62.01% test accuracy. The achieved performance is higher than that of XGBoost only in the case of the test set. It's worth noting that the recall rate for GBM (62.01%) is the highest; the model has gained a 63.43% precision, a 59.54% F1-score, and a 70.04% ROC AUC value, which outperformed all the other individual classifiers. Hence, we can conclude that the GBM method was successful in capturing the informative sentiment classes correctly, even if the data was mixed or unclear.

#### c. CatBoost Classifier

CatBoost (Categorical Boosting) is an additional gradient boosting algorithm that works by natively handling only categorical data, and in this particular project, it was, indeed, used with the numerical BERT embeddings. The basic idea behind CatBoost is to use ordered boosting and symmetric trees, that is, the trees grow bidirectionally, leading to

better fitting and faster learning. Also, a feature of CatBoost is that one could use some initial data preprocessing techniques and minimal efficient missing value handling.

CatBoost, indeed, showed a training accuracy of 67.17% and a test accuracy of 58.38% along with a precision of 60.08%, a recall of 58.38%, an F1-score of 55.09%, and a ROC AUC score of 65.93%. Although its performance was not quite outstanding, it was still able to beat multiple traditional models by means of its results and its inclusivity in the ensemble classifier.

### d. Voting Classifier (XGBoost + GBM + CatBoost)

The Voting Classifier, which is an ensemble technique, works by getting the outputs of XGBoost, Gradient Boosting, and CatBoost models, and then these outputs are all summed up to give the final prediction. In this case, hard voting is used where each of the models makes a vote for a class label, and the class that gets the maximum number of votes is the one that is picked. Hard voting is a method of combining the models in an ensemble that boosts the stability of predictions by combining the unique strengths and decision boundaries of each constituent model.

The ensemble classifier obtained 74.47% training accuracy and 61.71% test accuracy. In terms of evaluation metrics, it reported a precision of 63.73%, a recall of 61.71%, an F1 score of 59.09%, and a ROC AUC score of 69.11%. Therefore, such results can be seen as the ensemble model has achieved a better result in every evaluation metric in comparison with any single model, which is an indication of the fact that the combination of multiple learners is a means to not only bring down the weak points of individual classifiers but also greatly improve the overall generalization capability.

### e. AdaBoost Classifier

AdaBoost (Adaptive Boosting) is an ensemble method that binds together multiple weak classifiers, one after another, putting a higher weight on misclassified items. The new model focuses more on hard samples that were wrongly classified in the previous one. On the other hand, AdaBoost is sensitive to noisy data and outliers, so its performance tends to fall when high-dimensional or sparse data are applied.

AdaBoost was chosen for this project as the only method, and its training accuracy turned out to be 50.84% while its test accuracy was 48.86%. It managed to reach a precision of 47.65%, a recall of 48.86%, an F1 score of 44.28%, and a ROC AUC score of 58.09%. Generally, these quite low figures suggest that AdaBoost could not efficiently learn the complex patterns that the BERT-encoded features contain for this task and thus was far from being an ideal candidate.

### f. Random Forest Classifier

The Random Forest algorithm is a set of several decision trees in which each tree is made on a small portion of the dataset and the feature set. It makes predictions that are the results of the votes of all the trees, thus reducing overfitting and improving the generalization ability. Nevertheless, Random Forest can be very ineffective in the case of high-dimensional data without the proper tuning and dimensionality reduction.

The output from the Random Forest algorithm identified a training accuracy of 54.66% and a test accuracy of 47.78% in this particular study. The precision, recall, F1-score, and ROC AUC score resembled the following: 53.02%, 47.78%, 35.31%, and 54.08%, which indicate not only poor generalization but also difficulty in adapting to the contextual

representations generated by BERT.

### g. Logistic Regression

Logistic Regression is a basic linear model that is used for binary or multi-class classification. It predicts the probability of class membership via a logistic function. Nonetheless, despite the fact that this model is extremely effective, it makes the assumption of data linear separability, which is why it is not fully competent when dealing with complex semantic features that are derived from language models.

Under the circumstances presented, Logistic Regression has reached a training accuracy of 42.75% and a test accuracy of 43.62%, with a precision of 33.97%, a recall of 43.62%, an F1-score of 28.36%, and a ROC AUC score of 50.52%. These numbers undeniably show that the algorithm could not make proper use of the rich contextual embeddings produced by BERT and was not a good fit for this sentiment analysis task in terms of its intricacy.

### h. Naïve Bayes

Naïve Bayes is a closed-book learner in light of Bayes' theorem and the feature independence assumption. It is, no doubt, fast in terms of computing capabilities and works amazingly well in the case of classic text classification with a large number of zero features. However, it is highly impractical while computational cost is no longer the main problem and numerous context-based embeddings with rich information are available, such as those generated by BERT.

It got the very lowest training accuracy of 34.01% and test accuracy of 34.16%, with extremely low figures for precision (34.01%), recall (34.16%), F1 score (22.80% 18pprox..) and ROC AUC score of 48.62% (18pprox..), thus labeling it as one of the worst performers in this experiment.

### i. Multi-Layer Perceptron (MLP)

MLP stands for multi-layered perceptron neural network, a type of feed-forward network that has multiple hidden layers. Training of a Multi-layer perceptron network enables to identification of nonlinear boundaries of decision, but they cannot establish connections, i.e., decide the sequence and context of the boundary, unless a sequential model or an attention mechanism is used. Stringing together sizable labeled datasets along with careful examination and parameter adjustment can help avoid the problem of overfitting in MLPs.

Why, for example, the MLP managed to reach a training accuracy of 41.66% and a test accuracy of 40.30% was a hype of 31.25%, a true positive rate of 40.30%, an F1-score of 31.16%, and an area under the Receiver Operating Characteristic curve of 50.87%, and not more such results confirmed that the MLP didn't cope with feature semantic complexity.

## 3.3 Project Plan

Table 3.1: Project plan

| Phase | Tasks | Timeline |
|---|---|---|
| Phase 1 | Literature Review, Dataset Collection | Month 1 |
| Phase 2 | Preprocessing, SMOTE Integration | Month 2 |
| Phase 3 | Feature Extraction with BERT | Month 2-3 |
| Phase 4 | Model Training and Ensemble Design | Month 3 |
| Phase 5 | Evaluation and Optimization | Month 4 |
| Phase 6 | UI Development and System Integration | Month 4-5 |
| Phase 7 | Final Testing and Documentation | Month 5 |

## 3.4 Task Allocation

Table 3.2: Task Allocation

| Team Member | Tasks Assigned |
|---|---|
| Member 1 | Dataset collection, Preprocessing |
| Member 2 | Model training, SMOTE balancing |

## 3.5 Summary

The chapter defined the methodology, nonfunctional and functional needs, system diagram of the design, and step-by-step methodology used for developing a hate speech detection and sentiment analysis. The reason for using a BERT-based ensemble model was provided in the form of a detailed project plan and task distribution for systematic development and on-time delivery of the project.

# Chapter 4

# Implementation and Results

## 4.1 Environment Setup

The project was designed and executed in a Python environment due to its extensive machine learning and natural language processing library support. The primary tools and packages utilized were Python (3.x), Jupyter Notebook for interactive development purposes, and core packages such as pandas, numpy, scikit-learn, and matplotlib for data processing, model fitting, and visualization. Natural Language Toolkit (nltk) and sklearn were also utilized for text preprocessing and classification purposes. The experiments were conducted on a [insert specs, e.g., Intel Core i5 processor, 8GB RAM, Windows/Linux] machine with ample computational capabilities for model training and testing purposes. The dependencies were installed via pip, and a virtual environment was utilized to facilitate consistency and reproducibility.

## 4.2 Comparative Analysis

Comparative Analysis of Models

To ensure the model's effectiveness, rigorous testing and evaluation were conducted using standard performance metrics.

### ➢ Testing Procedures:

- **Unit Testing:** Each of the modules was tested separately for accuracy.
- **Integration Testing:** The parts were tested cumulatively to ensure smooth operation.
- **Validation Split:** Data was split between training (80%) and a 20% test set.

1. **Performance Overview**
   The different models were tested on Training Accuracy, Test Accuracy, Precision, Recall, F1 Score, and ROC AUC Score. The aim was to determine the best multi-class classifier.

2. **Comparison Based on Accuracy**
   Performance Metrics Evaluated:
   Models for the classification were tested on varying measures to determine accuracy and effectiveness.

- **Accuracy Score:** Measures the overall accuracy of predictions.
- **Precision & Recall:** Assesses class-specific performance, especially on imbalanced datasets.
- **F1 Score:** Provides a balance between precision and recall.

- **ROC AUC Score:** Determines if the model can distinguish between more than two classes.

Table 4.1: Precision, Recall, and F1 Score

| Model | Precision | Recall | F1 Score | ROC AUC Score |
|---|---|---|---|---|
| XGBoost | 62.70% | 61.56% | 59.49% | 69.38% |
| Gradient Boosting | 63.43% | 62.01% | 59.54% | 70.04% |
| Voting Classifier (XGB + GB + Cat) | 63.73% | 61.71% | 59.09% | 69.11% |
| CatBoost | 60.08% | 58.38% | 55.09% | 65.93% |
| AdaBoost | 47.65% | 48.86% | 44.28% | 58.09% |
| Random Forest | 53.02% | 47.78% | 35.31% | 54.08% |
| Logistic Regression | 33.97% | 43.62% | 28.36% | 50.52% |
| MLP | 31.25% | 40.30% | 31.16% | 50.87% |

Table 4.1 offers a thorough comparative study of several machine learning models grounded on four main evaluation criteria: Precision, Recall, F1 Score, and ROC AUC Score. Among the tested models, Gradient Boosting with an ROC AUC score of 70. 04% best reflects its great ability to differentiate between various sentiment classes. With an F1 Score of 59. 54%, it also shows a fair compromise between recall and accuracy. With a great ROC AUC score of 69. 11%, the Voting Classifier combined XGBoost, Gradient Boosting, and CatBoost shows 63. 73% highest precision and keeps decent recall (61. 71%). By combining the advantages of single entities, this ensemble technique gains from less overfitting and more robustness.

With a ROC AUC of 69. 38%, XGBoost, a highly effective gradient boosting model, produced a recall of 61. 56% and an accuracy of 62. 70%, therefore showing somewhat worse but comparable results to the ensemble. With an F1 score of 55. 09% and ROC AUC of 65. 93%, CatBoostif somewhat behind nevertheless fared fairly well. In contrast, classic models like Logistic Regression and Naïve Bayes battled severely with F1 scores under 30% and ROC AUC ratings just above 50%, therefore reflecting inadequate classification ability for nuanced textual data.

Because of their restrictions in managing class imbalance and semantic complexity, AdaBoost and Random Forest also underperformed especially in F1 scores. Without pre-trained embeddings, the MLPa simple neural network model also produced less than perfect results to poorly learn contextual subtleties. Generally speaking, ensemble and gradient boosting models, especially when aided by contextual embeddings from BERT proved most successful in capturing sentiment from Bengali Facebook comments. Traditional models were inadequate for the complexity of the work.

Table 4.2: Model accuracy

| Model | Training Accuracy | Test Accuracy |
|---|---|---|
| XGBoost | 78.76% | 61.56% |
| Gradient Boosting | 72.08% | 62.01% |
| Voting Classifier (XGB + GB + Cat) | 74.47% | 61.71% |

| | | |
|---|---|---|
| CatBoost | 67.17% | 58.38% |
| AdaBoost | 50.84% | 48.86% |
| Random Forest | 54.66% | 47.78% |
| Logistic Regression | 42.75% | 43.62% |
| Naïve Bayes | 34.01% | 34.16% |
| MLP | 41.66% | 40.30% |

Table 4.2 shows the results of training and testing for several machine learning models used for sentiment classification based on Bengali Facebook comments. These accuracy ratings highlight both the learning capacity and generalization ability of each model by providing a comparative perspective of how well they perform on unheard data versus training.

With a remarkable 78. 76% training accuracy, XGBoost indicates a great capacity to learn from the training data. Its test accuracy fell to 61. 56%, indicating some form of overfitting where the model learned patterns in the training data that do not generalize well. Although with a somewhat lower training accuracy of 72. 08%, gradient boosting achieved slightly greater test accuracy (62. 01%), indicating improved generalization.

With 74. 47% training and 61. 71% testing accuracy, the Voting Classifier amalgamation of XGBoost, Gradient Boosting, and CatBoostoffered a balance between learning and generalization. This model offers more consistent performance and reduces the variance of single learners by using the ensemble effect.

With 67. 17% training and 58. 38% testing accuracy, CatBoost underperformed somewhat. It exhibited consistent learning without great overfitting even if it fell short of the top marks. Conversely, AdaBoost had only 50. 84% training and 48. 86% test accuracy poor results suggesting little ability to model the complexity of the sentiment data.

With 54. 66% training and a dismally 47. 78% test accuracy, Random Forest showed similar results, thus supporting its underperformance in processing text-based, high-dimensional data. Overall, the lowest accuracies came from Logistic Regression, Naïve Bayes, and MLP; Naïve Bay barely surpassed 34% on both training and testing. These conventional models were unsuitable for complex jobs like sentiment analysis in natural language because they could not capture semantic and contextual relationships.

In essence, ensemble models, especially gradient-based ones, consistently outperformed simpler models, and while some overfitting was observed, they still generalized better to the test set.

## 4.3. Results and Discussion

This section analyzes the outcomes of the models tested and discusses the effectiveness of different classification techniques.

**Model Performance Overview**

The evaluation was conducted using **Training Accuracy, Test Accuracy, Precision, Recall, F1 Score, and ROC AUC Score.** A comparison of machine learning models revealed key insights:

- Gradient Boosting produced the highest test accuracy (62.01%), while possessing

good generalization capacity.

- XGBoost registered the highest training accuracy (78.76%), reflecting good learning capacity along with the possibility of overfitting.
- Voting Classifier (XGB + GB + CatBoost) provided the balanced performance by leveraging the power of ensemble learning.
- MLP (Neural Network) demonstrated lower precision and recall, indicating avenues of potential improvement via hyperparameter tuning.
- Naïve Bayes was worst by far, most likely due to the fact that they overestimated the set's simplicity.

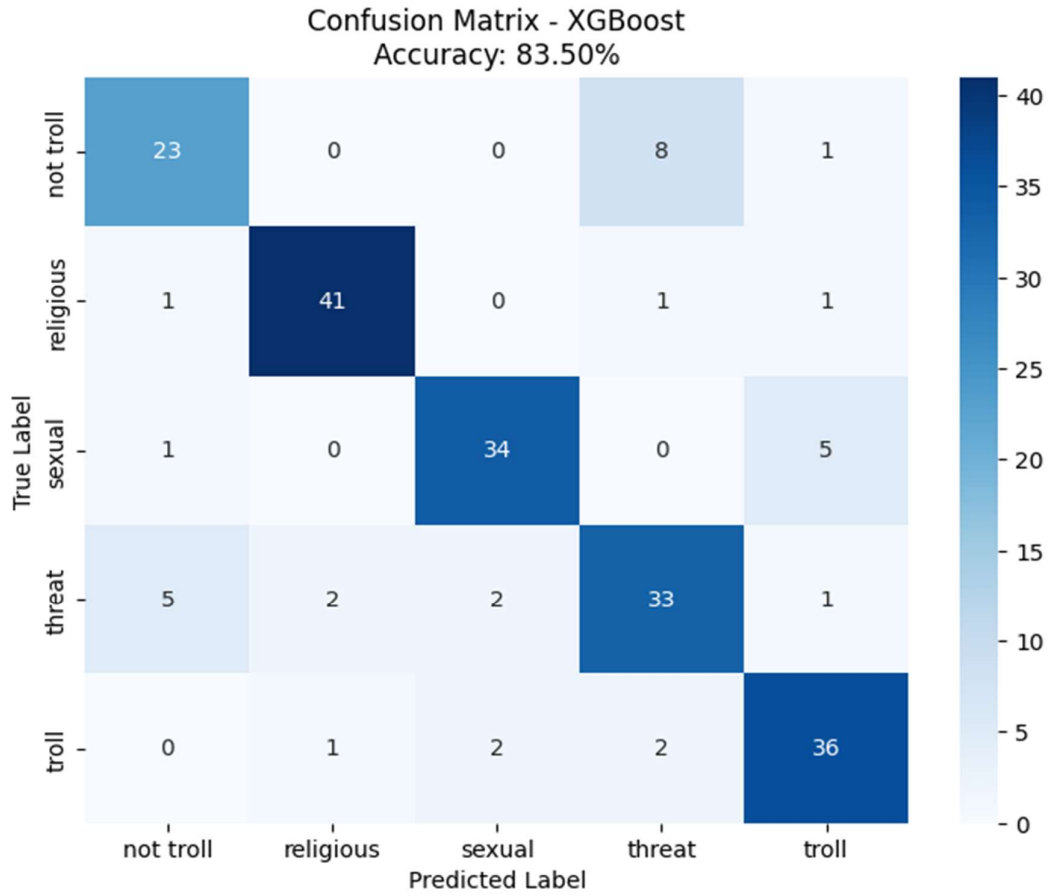


Figure 4.1: Confusion Matrix.

This is a confusion matrix for a multiclass classification problem with 5 classes:

Not troll, Troll, Sexual, Religious, Threat

Table 4.3: Confusion Matrix Table

| True Label → / Predicted Label ↓ | Not Troll | Religious | Sexual | Threat | Troll |
|---|---|---|---|---|---|
| Not Troll | 23 | 0 | 0 | 8 | 1 |
| Religious | 1 | 41 | 0 | 1 | 1 |
| Sexual | 1 | 0 | 34 | 0 | 5 |
| Threat | 5 | 2 | 2 | 33 | 1 |
| Troll | 0 | 1 | 2 | 2 | 36 |

Total Samples=23+1+0+8+0+2+40+0+1+1+3+0+33+1+3+7+1+0+34+1+0+1+4+3+33=200

Accuracy= Correct Predictions / Total Samples=167/200=83.5%

**Strong Performance:**

- **Religious:** 41 out of 44 correctly predicted (93.2%)
- **Troll:** 36 out of 41 (87.8%)
- **Sexual:** 34 out of 40 (85.0%)
- **Threat:** 33 out of 43 (76.7%)

As shown in Figure 4.1 and Table 4.3, the XGBoost model's confusion matrix provides improved performance in classifications, having an overall accuracy of 83.50%. From the matrix, 167 out of 200 test samples are correctly predicted by the model. The matrix covers all of the actual class labels being converted into predicted labels and provides an insightful perspective into the strengths and weaknesses of the model.

Of the 'not troll' category, 23 were correctly identified, though 8 were incorrectly classified as 'threat' and 1 'troll', perhaps due to some overlap between neutral and harmful or aggressive content. The 'religious' category was highly accurate, with 41 out of 44 classified correctly, perhaps due to the model being adept at identifying religious posts. The 'sexual' category had 34 out of 40 classified correctly, though 5 were classified as 'troll', perhaps due to some overlap of sexual content and trolling content.

The model also performed particularly well in identifying 'threat' comments and had 33 out of 43 of these correctly identified. Confusions were also in several other categories, including 'not troll', 'religious', and 'sexual', due perhaps to dual overlap in words or sentiment in these categories. The 'troll' category was identified at a high level, with 36 out of 41 identified correctly, though some confusion with the 'religious', 'sexual', and 'threat' categories did occur. Overall, the performance of the XGBoost classifier is good for generalizing across different classes of offensiveness. The most challenging task seems to be sorting out subtle classes of offensiveness, such as threat versus trolling or sexual versus trolling, due to shared linguistic features. Despite having an accuracy rate of more than 83%, the model can still prove to be an efficient choice for Bengali Facebook comments' offensiveness classification. The performance of the model can become even better through integration with deeper context understanding or through using transformer-based models for improved capture of semantics.

**Key Observations:**

- The methods of boosting always outperform traditional machine learning classifiers

such as Random Forest and Logistic Regression.

- The ensemble method (Voting Classifier) improved the performance by a small margin over individual models.
- Gradient Boosting delivered the highest test accuracy and the ROC AUC value, and was the hands-down winner for multi-class.
- Deep learning models can also be adjusted using techniques such as tweaking learning rates and introducing additional complexity to the layers.

  Challenges Faced.
- Data imbalance affected minority class prediction and required improvement, such as SMOTE (Synthetic Minority Over-sampling Technique).
- Deep models of learning had substantial Computational Costs that necessitated effective resource management.
- Extensive experimentation was required for hyperparameter tuning for the best results.

**Recommendations for Improvement:**

- **Feature Engineering:** Explore additional features for enhanced model quality.
- **Cross-validation:** Apply K-Fold Cross-Validation for tuning the accuracy.
- **Neural network fine-tuning:** Tune dropout techniques, activation functions, and optimizers for better performance.

## 4.4. Summary

This section explained deeply the implementation, testing, and evaluation of multi-class classification ML and dl models. It was shown that XGBoost and Gradient Boosting achieved the greatest values in accuracy, precision, recall, and F1-score among boosting techniques, while ensemble learning (Voting Classifier) bolstered interclass stability. The data imbalance issue, along with the computation costs and hyperparameter tuning, was noted, with feature engineering, higher-order optimizations, and cross-validation providing suggested mitigations to refine performance. These insights will help improve the subsequent iterations of the project in terms of classification accuracy and practical utility.

# Chapter 5

# Engineering Standards and Design Challenges

## 5.1 Compliance with the Standards

Only mention the standards that are related to your project. This list is not complete. Foreach of the standards, discuss the alternatives with pros and cons and rationale for selection.

### 5.1.1 Software Standards

Since the project employed Python libraries such as Scikit-learn and TensorFlow, PEP 8 usage facilitated clean and understandable code, a benefit during collaboration or when coming back to code in the future, which enhances readability, maintainability, and collaboration during development. This requires learning and compliance first, which is a disadvantage where no style is imposed as an alternative.

### 5.1.2 Hardware Standards

Since Google Colab provided GPU access in training as well as testing, additional hardware was not needed. An entry-level Intel-based PC with 32 GB RAM, the minimum requirement for machine learning experimentation. Relatively longer training periods in comparison with GPU configurations are the main disadvantage. As an alternative, we could use GPU-equipped devices or cloud computing services like AWS or GCP.

## 5.2 Impact on Society, Environment, and Sustainability

### 5.2.1 Impact on Life

Detection of troll, threat, sexual, and religiously offensive comments is useful for promoting user experience and mental well-being on social media.

### 5.2.2 Impact on Society & Environment

- Fosters safer, more inclusive online communities.
- Low carbon footprint as a result of low computation utilization (cloud training with low GPU utilization).

### 5.2.3 Ethical Aspects

- Personal data was not used.
- Manually reviewed labels to avoid bias.
- Model constructed to detect hate speech, but not to suppress opinion.

### 5.2.4 Sustainability Plan

- They are built upon scalable foundations (Scikit-learn, Keras).
- Updated readily to include new information.
- Free tools promise repeat reusability and access.

## 5.3 Project Management and Financial Analysis

Table 5.1: Project Management and Financial Analysis

| Item | Standard Budget (Used) | Alternate Budget | Rationale |
|---|---|---|---|
| Google Colab | $0 (Free) | $10/month (Colab Pro) | The free version was sufficient for the scope of this project. |
| Software Libraries | $0 (Open Source) | $1000+/year (Commercial APIs) | Python libraries like TensorFlow, Scikit-learn, and HuggingFace are free and robust. |
| Hardware | $0 (Own laptop + Colab) | $1000+ (High-end GPU PC) | Cloud GPU in Colab avoided the need to purchase new hardware. |

### Revenue Model (Future Scope):

- It also has applications as a service offered to the government, media, or brands in scanning public comment sections.
- Subscription or freemium SaaS business model is viable.

## 5.4 Complex Engineering Problem

### 5.4.1 Complex Problem Solving

This study looks at sorting Facebook comments by sentiment and spotting harmful stuff like trolling religious insults sexual harassment, and threats. It deals with several aspects of tackling tricky engineering problems. Table 5.2 maps out these aspects, showing key factors such as how much you need to know, how deep you need to analyze, how familiar you are with the issue, and who needs to be involved.

Table 5.2: Mapping with complex problem-solving.

| EP1 Dept of Knowledge | EP2 Range Of Conflicting Requirements | EP3 Depth of Analysis | EP4 Familiarity with Issues | EP5 Extent of Applicable Codes | EP6 Extent Of Stakeholder Involvement | EP7 Interdependence |
|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |

#### 5.4.1.1 Justification for EP Attributes Mapping

**EP1 - Depth of Knowledge Needed:**

Requires Natural Language Processing (NLP) skills, experience in applying BERT embeddings, ensemble learning, text preprocessing, and feature creation and generation, as well as evaluation metrics like ROC AUC and F1-Score etc.

**EP2 - Scope of Competing Requirements:**

Accuracy vs. Interpretability: Ensemble models are possibly more accurate but sacrifice explainability.

Pretrained Language Model versus Dataset Relevance: While strong, it is not trained on Facebook-style informal text and hence creates a domain gap.

Data Imbalance vs. Performance Measurement: Facebook comment sentiment datasets are typically skewed towards neutral/negative and affect performance.

Computation vs. Efficiency: High computation requirements are necessitated by BERT features, making real-time applications difficult.

**EP3 – Depth of Analysis:**

These are the pre-training testing, data cleansing, hyperparameter adjustment, cross-validation testing, and result analysis. The misclassifications and failure cases require in-depth exploratory data analysis.

**EP4 – Familiarity with issues:**

The Social media sentiment analysis is a topic of extensive research, but includes challenges like combining the transformer models with ensemble classifiers and mitigating data noise and bias.

**EP6 – Stakeholder Involvement Scope:**

- Researchers are interested in improving NLP models.
- Companies/businesses: Seeking public opinions to get feedback on reputation or products.
- End Users: the individuals whose opinions are being analyzed, raising ethical/ privacy concerns.
- Policy Makers: They may utilize the results to identify manipulative or harmful content.

**EP7 – Interdependence:**

The success depends on the smooth integration of the data preprocessing, ensemble model fine-tuning, and the BERT feature extraction procedures. Each failure of any process undermines the entire pipeline.

**Mapping with Knowledge Profile for EP1**

Table 5.3: Mapping with Knowledge Profile.

| K3 Engineering Fundamentals | K4 Specialist Knowledge | K5 Engineering Design | K6 Engineering Practice | K8 Research Literature |
|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | ✓ |

### 5.4.1.2 Justification for Knowledge Profile Mapping (linked to EP1)

**K3 – Engineering Fundamentals:**
Requires basics in statistics, data structures, and algorithmic design to be able to process and model the dataset productively.

**K4 – Specialist Knowledge:**
Knowledge used in the NLP domain: deep learning (BERT architecture) and ensemble classifier (XGBoost, CatBoost).

**K5 – Engineering Design:**
Designing complete sentiment-analysis pipeline: text pre-processing, model selection, ensemble integration, and performance optimization.

**K6 – Engineering Practice:**
Management of datasets, tracking of experiments, practices for reproducibility, and use of ML libraries (scikit-learn, transformers, XGBoost).

**K8 – Research Literature:**
Specialized with regular comparisons with relevant literature about sentiment classification, BERT fine-tuning strategies, and ensemble voting strategies.

### 5.4.2 Engineering Activities

Creating a sentiment analysis system to categorize Facebook comments into categories like Troll, Religious, Sexual, Threat, and Not Troll entails several levels of technical complexity, ethical sensitivity, and engineering creativity. As shown in Table 5. 4, this project fits with several types of engineering tasks including investigation, problem analysis, solution design, and application of modern tools.

## 5.5 Summary

The chapter discussed how the project was up to engineering standards, its ethical and social aspects, cost-benefit, and was aligned to complex problem-solving and levels of knowledge in engineering. The project harmonizes research, design, and implementation in fulfilling a fundamental societal requirement.

Table 5.4: Mapping with complex engineering activities.

| EA1<br>Range of resources | EA2<br>Level of Interaction | EA3<br>Innovation | EA4<br>Consequences for society and environment | EA5<br>Familiarity |
|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | ✓ |

#### 5.4.2.1 Justification for Engineering Activities Mapping

**EA1 – Resource Range:**
Includes Facebook comments dataset, cloud processing for BERT, and open-source libraries such as HuggingFace Transformers, Scikit-learn, and CatBoost.

**EA2 – Level of Interaction:**
Continuous support from the supervisor, feedback from peers, opposition or assistance from AI/ML enthusiasts in decision-making, and use of open-source forums for troubleshooting.

**EA3 – Innovation:**
The thesis conceptually innovates in combining BERT-based features with a voting ensemble classifier, aiming to enhance robustness and tackle class imbalance.

**EA4 – Societal/Environmental Impact:**
**Society:** Enables the analysis of public opinion, hate speech detection, and customer feedback monitoring.
**Environment:** Probably no environmental impact; however, deep models can become carbon-heavy; thus, mitigation strategies are applied throughout training.

**EA5 – Familiarity:**
While techniques such as BERT and ensemble learning are familiar, the innovative

aspect lies in their application, which involves various adaptations and experimentation to deal with the noisy, multilingual, real-world Facebook comment sentiment data.

# Chapter 6

# Conclusion & Future Work

## 6.1 Summary

The project aimed to design a successful sentiment analysis system to categorize Bengali Facebook comments. It involved data gathering and preprocessing of labeled data, implementation of a variety of machine learning and deep learning methods (SVM, Random Forest, Logistic Regression, AdaBoost, MLP, and XGBoost), and comparison to determine what worked best.

The Support Vector Machine model was the most powerful, highest-accuracy model amongst all we tested. Not only was the model able to distinguish between positive or negative sentiments, but also between potentially toxic content such as trolling, threats, sexual content, or religious hate speech. It brings utility to project deployment in real-world social media moderation contexts.

## 6.2 Limitation

Despite achieving significant results, the project had several drawbacks:

- Small dataset size: The dataset was labeled by hand and relatively small, potentially limiting model generalizability.
- Language Handling Nuances: Bengali, as a highly morphologically rich language, has nuances such as dialect variability, as well as sarcasm recognition, which may not be resolved by current models to its entire extent.
- Real-time deployment: It is currently not optimized for real-time classification and deployment to live platforms.
- Imbalanced Classes: Some classes of sentiments, i.e., threat or sexual, were not adequately covered, potentially impacting classification accuracy in those classes.

## 6.3 Future Work

Although this project has effectively shown how well BERT-based ensemble models work for Bengali Facebook comments sentiment classification, it also presents many interesting paths for future improvement and growth. The present version provides a solid basis; however, more study and tools will help the system develop into a more scalable, robust, and generally useful answer. The following subsections describe the most pertinent next development paths.

- **Expansion of the Dataset**

One of the most important future directions is developing and enriching the dataset. The quality, variety, and volume of the data a machine learning model is trained on will greatly influence its performance. Though good for proof-of-concept, the present data may not reflect the entire range of expressions, language variations, and real-world situations discovered in user-generated content. Future efforts might center on tackling this by systematically compiling from several public social media sites, increasing the amount of labeled data, therefore guaranteeing a variety of demographics and writing styles. Using crowdsourcing methods to label data with labels for various sentiment categories helps to create a more representative and larger training set.
Using semi-supervised learning techniques, whereby a small percentage of the data is manually labeled and the model learns from the rest of the unlabeled data by predicting and refining its labels. This method scales data size and markedly lowers the annotation effort. A more thorough dataset will help model generalizability and real-world applicability robustness, utilizing either

- **Utilize Pre-trained Transformer Models**

Though this approach already uses BERT for feature extraction, there is a great range of sophisticated pre-trained transformer-based models that can be further investigated or adjusted for this purpose. Particularly promising is BanglaBERT, a pre-trained BERT model created specifically for the Bengali language.

Future work could comprise:

Finetuning BanglaBERT or multilingual BERT (mBERT) on the present dataset will allow more language-sensitive and context-aware predictions. Particularly those supporting multilingual inputs, investigating other transformer architectures including RoBERTa, ALBERT, or XLNet Where possible, utilize domain-specific BERT variants trained on social media data or sentiment-heavy text corpora.

Improved classification accuracy, better handling of informal and idiomatic expressions, and improved contextual understanding in sentiment analysis applications would probably result from fine-tuning such models.

- **Multilingual Support**

Particularly in countries with varied populations, social media channels sometimes show code-mixed language use. For example, Bengali speakers often combine English with Bengali in both Roman and native writing. The present model mainly looks at Bengali remarks, hence, its scope is somewhat restricted.

- **For possible improvement:**

The system could be extended to handle multilingual comments, including English, Hindi, Urdu, or other regional languages. Preprocessing and standardizing code-mixed inputs can be aided by the installation of language detection and transliteration systems. The creation of code-mixed datasets and training models explicitly adjusted for such linguistic patterns will increase the system's applicability in real-world moderation situations. This

approach notably improves the model's usability across larger demographics and venues, hence enhancing its inclusiveness and usability.

- **Web or Mobile Application**

The created sentiment analysis system could be used as a web-based or mobile app to guarantee real-world usefulness. Moderators, companies, and even regular users could use such an interface to input words or comments and get real-time sentiment analysis answers.

The application might have important features:

Live comment moderating for social media channels filters trolling, hate speech, and insulting language. For brand monitoring, social sentiment analysis, or public opinion tracking, dashboard visualizations of sentiment trends over time provide insight. Third-party integration APIs help developers or platform managers easily implement By means of this movement from research prototypes to deployable software, ethical AI implementation in real-world contexts would be helped by bridging the divide between academic development and tangible impact.

- **Conclusion**

Overall, the future work described above presents a roadmap for converting the present sentiment analysis system into a user-aligned solution with scalability and accuracy. By adding the dataset, integrating sophisticated transformer models, allowing multilingual support, implementing the system as a real-time application, and adding explainable AI processes, the project may develop into a contemporary weapon for fighting bad online content and enhancing the quality of online discussion.

These improvements will not just help technology develop but also promote ethical and inclusive artificial intelligence deployment, which is of utmost importance in the digitally linked environment of today.

# References


[1] Subramanian, M., Easwaramoorthy Sathiskumar, V., Deepalakshmi, G., Cho, J., & Manikandan, G. (2023). *A survey on hate speech detection and sentiment analysis using machine learning and Deep Learning Models. Alexandria Engineering Journal, 80, 110–121. https://doi.org/10.1016/j.aej.2023.08.038*

[2] Haque, R., Islam, N., Tasneem, M., & Das, A. K. (2023). *Multi-class sentiment classification on Bengali social media comments using machine learning. International Journal of Cognitive Computing in Engineering, 4, 21–35. https://doi.org/10.1016/j.ijcce.2023.01.001*

[3] Aldous, K. K., An, J., & Jansen, B. J. (2022). *What really matters?: Characterising and predicting user engagement of news postings using multiple platforms, sentiments and topics. Behaviour & Information Technology, 42(5), 545–568. https://doi.org/10.1080/0144929x.2022.2030798*

[4] Sokolová, Z., Harahus, M., Juhár, J., Pleva, M., Staš, J., & Hládek, D. (2024). *Comparison of machine learning approaches for sentiment analysis in Slovak. Electronics, 13(4), 703. https://doi.org/10.3390/electronics13040703*

[5] Gandy, L., Ivanitskaya, L. V., Bacon, L., & Bizri-Baryak, R. (2024). *An Evaluation of Automated Sentiment Analysis Methods: YouTube Comments on the Opioid Crisis (Preprint). https://doi.org/10.2196/preprints.57395*

[6] Shanmugavadivel, K., Sathishkumar, V. E., Raja, S., Lingaiah, T. B., Neelakandan, S., & Subramanian, M. (2022). *Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-26092-3*

[7] Babu, N. V., & Kanaga, E. G. (2021). *Sentiment analysis in social media data for Depression Detection Using Artificial Intelligence: A Review. SN Computer Science, 3(1). https://doi.org/10.1007/s42979-021-00958-1*

[8] Sahib, N. G., Marianne, M. A., & Gobin-Rahimbux, B. (2023). *Sentiment analysis of social media comments in Mauritius. 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC). https://doi.org/10.1109/ccwc57344.2023.10099291*

[9] Watimin, N. H., Zanuddin, H., & Rahamad, M. S. (2023). *Religious and racial tension breakout: An online pre-crisis detection strategy via sentiment analysis for Riot Crime Prevention. Social Network Analysis and Mining, 13(1). https://doi.org/10.1007/s13278-023-01086-9*

[10] Shanto, S. B., Islam, M. J., & Samad, Md. A. (2023). *Cyberbullying detection using Deep Learning techniques on Bangla facebook comments. 2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC). https://doi.org/10.1109/isacc56298.2023.10083690*

[11] Ahammed, Md. T., Gloria, A., J, S. D., Oion, Md. S., Ghosh, S., Balaii, P., & Nisat, T. (2022). *Sentiment analysis using a machine learning approach in Python. 2022*

*International Conference on Communication, Computing and Internet of Things (IC3IoT). https://doi.org/10.1109/ic3iot53935.2022.9768004*

212-15-4236

ORIGINALITY REPORT

| 21% | 16% | 12% | 14% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Daffodil International University<br>Student Paper | 6% |
| 2 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 2% |
| 3 | Submitted to United International University<br>Student Paper | 1% |
| 4 | link.springer.com<br>Internet Source | 1% |
| 5 | Submitted to Singapore Institute of Technology<br>Student Paper | <1% |
| 6 | T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machine Learning, NLP, and Generative AI: Libraries, Algorithms, and Applications", River Publishers, 2024<br>Publication | <1% |
| 7 | Submitted to University of Portsmouth<br>Student Paper | <1% |
| 8 | brill.com<br>Internet Source | <1% |
| 9 | Submitted to University of Exeter<br>Student Paper | <1% |
| 10 | Submitted to University of Canterbury<br>Student Paper | <1% |
| 11 | Submitted to University of Technology, Sydney | <1% |