NJIT

New Jersey Institute
of Technology

Project title: Social Media Hate Speech Detection

By:-

Dhruv Patel (dp972)

Kenil Pravinbhai Avaiya(ka683)

Siva Venkata Rahul Varma Datla(sd2336)

Yash Kamlesh Shah(yks)

Submitted to: Professor Khalid Bakhshaliyev

Abstract

This project seeks to address the critical challenge of detecting hate speech on social media platforms, given the complexity of linguistic diversity, context, and the often subtle nature of toxic content. By developing multiple advanced machine learning models and leveraging state-of-the-art natural language processing techniques, this project aims to identify and classify hate speech with high accuracy and reliability.

Project entitled **“Social Media Hate Speech Detection**” focuses on the development of a robust and adaptable framework for hate speech detection. The project integrates cutting-edge algorithms to effectively analyze diverse datasets from various languages and social media contexts.

The core objective is to create a solution that not only tackles the technical challenge of detecting hate speech but also contributes to a safer online environment. By employing advanced preprocessing pipelines, exploratory data analysis, and sentiment/emotion integration, this project uncovers nuanced patterns within toxic content. Furthermore, it addresses the issue of class imbalances by employing oversampling techniques and dynamic class-weight adjustments, ensuring reliable detection across minority categories.

This project aspires to empower platforms, moderators, and researchers with actionable insights, enhancing their ability to combat hate speech effectively. By aligning advanced analytics with real-world applicability, the project contributes to the growing field of hate speech detection while fostering healthier online discourse.

Contents

1 Problem Statement

Social media platforms, while transformative in enabling global connectivity and communication, have also become breeding grounds for hate speech, adversely affecting individuals, groups, and societal harmony. Detecting and mitigating hate speech remains a significant challenge due to the complexity of language, diverse cultural contexts, and subtle variations in toxic content. Existing detection approaches often fall short in accuracy and adaptability, particularly when addressing multilingual and imbalanced datasets.

This project, **"Social Media Hate Speech Detection"** is designed to tackle these challenges by harnessing advanced machine learning and natural language processing techniques. By incorporating sophisticated models such as BERT, enhanced preprocessing pipelines, and emotion/sentiment integration, the project aims to build a reliable system for detecting hate speech in varied linguistic and contextual settings. The goal is to create a robust solution that empowers social media platforms, researchers, and policymakers to address hate speech effectively, contributing to safer and more inclusive digital environments.

2 Objective of the Project

- The objective of the project **"Social Media Hate Speech Detection"** centers on the development of sophisticated models capable of analyzing social media text data to identify and classify hate speech with precision. The project aims to construct advanced natural language processing frameworks that leverage machine learning algorithms to address the linguistic diversity, contextual nuances, and imbalances inherent in hate speech datasets.
- Through the deployment of cutting-edge techniques, the project aspires to uncover patterns and correlations within toxic content while maintaining high accuracy across varied datasets. By surpassing conventional methods, this project seeks to empower social media platforms, moderators, and stakeholders with actionable insights to foster safer online environments. The ultimate goal is to create a reliable and adaptable hate speech detection framework that contributes to the evolving field of content moderation and online safety.

The detailed objectives/goals of this project are as follows:

- **Selection and Pre-processing of Dataset:** Identify and utilize a diverse dataset of social media comments, ensuring representation of multiple languages and contexts. Perform comprehensive data cleaning to handle inconsistencies, remove noise such as non-relevant symbols, and address issues like missing or duplicate entries to enhance the quality of the data.
- **Feature Extraction and Relevance Analysis:** Identify the most critical linguistic and contextual features that contribute to hate speech detection. This includes leveraging advanced natural language processing techniques to extract relevant features such as n-grams, sentiment polarity, and syntactic structures for improved model accuracy.
- **Algorithm Selection and Fine-tuning:** Experiment with various machine learning and deep learning models such as logistic regression, random forests, and fine-tuned BERT architectures. Optimize these algorithms through hyperparameter tuning to achieve superior performance and generalization.
- **Evaluation Metrics:** Establish comprehensive evaluation metrics, including precision, recall, F1-score, and area under the ROC curve (AUC), to effectively assess model performance, especially in detecting minority classes like hate speech.
- **Accurate Hate Speech Classification:** Develop a machine learning model capable of detecting hate speech with high precision and recall across various datasets and contexts. The focus will be on achieving robustness and adaptability, ensuring effective classification even in multilingual and imbalanced datasets.

3 Dataset Description

This dataset, sourced from Kaggle, contains labeled social media comments aimed at detecting hate speech, offensive language, and neutral content. It provides a diverse range of comments with detailed annotations, enabling a comprehensive analysis of toxic language patterns and their contextual nuances. The dataset is structured to support machine learning and natural language processing tasks for hate speech detection.

Link of the Dataset:- https://www.kaggle.com/datasets/subhajeetdas/hate-comment/data

Technical description:

- **Dataset Size:** The dataset contains 41,145 rows and 3 columns. Each row represents a unique comment, labeled for its level of toxicity.
- **Columns Description:**

  - **Comment (object):** Contains the text of the social media comment, which may include hate speech or no hate speech content.
  - **Label (int64):** Indicates the class of the comment:
    - P: Hate Speech (Toxic content with hateful intent).
    - N: Not a Hate Speech (No Toxic content).
  - **Unamed_ID (object):** A unique identifier for each comment.

Key Dataset Attributes:

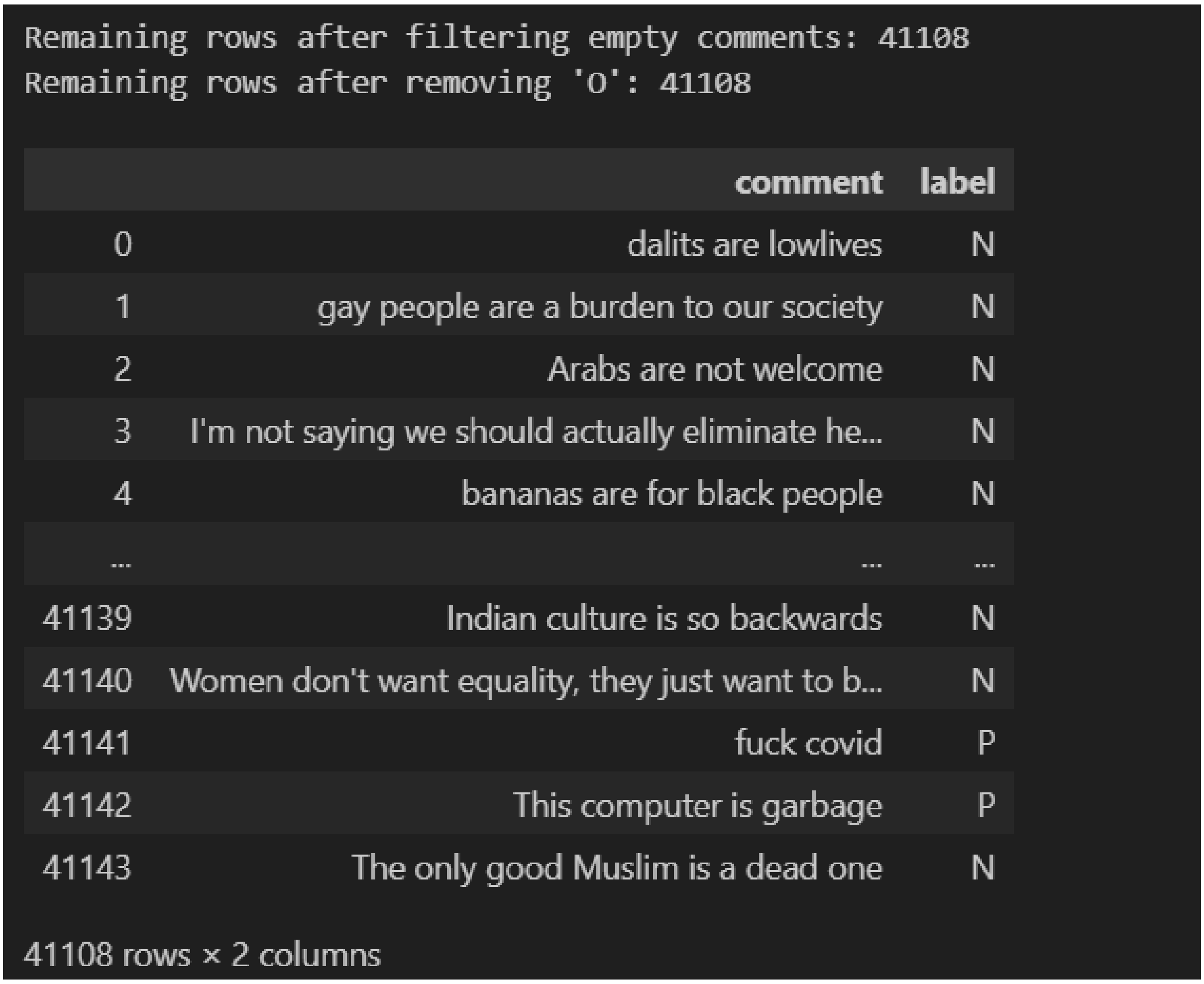
- **Balanced Representation:** While the dataset contains comments across two categories (hate speech and no hate speech), it exhibits some degree of imbalance..
- **Text Diversity:** The comment text includes a mix of formal and informal language, abbreviations, and special characters, reflecting real-world social media language.

4 Data Analysis and Representation

Before drawing any conclusions, we take the time to truly listen to our data. This means carefully looking at every aspect of it, from the different variables to the patterns they reveal. When we dive into understanding our data, we aim to get a broad sense of what it's all about. We want to know things like how many rows and columns it has, what kinds of values are present, the types of data they represent, and if there are any missing pieces we need to fill in.

4.1 Pre-processing

- **Insights:**
  After pre-processing (removal of empty and invalid rows), the dataset contains 41,144 rows, suggesting a clean and manageable dataset size for training machine learning models.
- **Relevance to the Project**: The removal of noisy data improves the dataset's quality, ensuring models learn from relevant and meaningful examples. However, it also emphasizes the importance of ensuring sufficient data diversity for effective generalization.



```
Remaining rows after filtering empty comments: 41108
Remaining rows after removing 'O': 41108
```

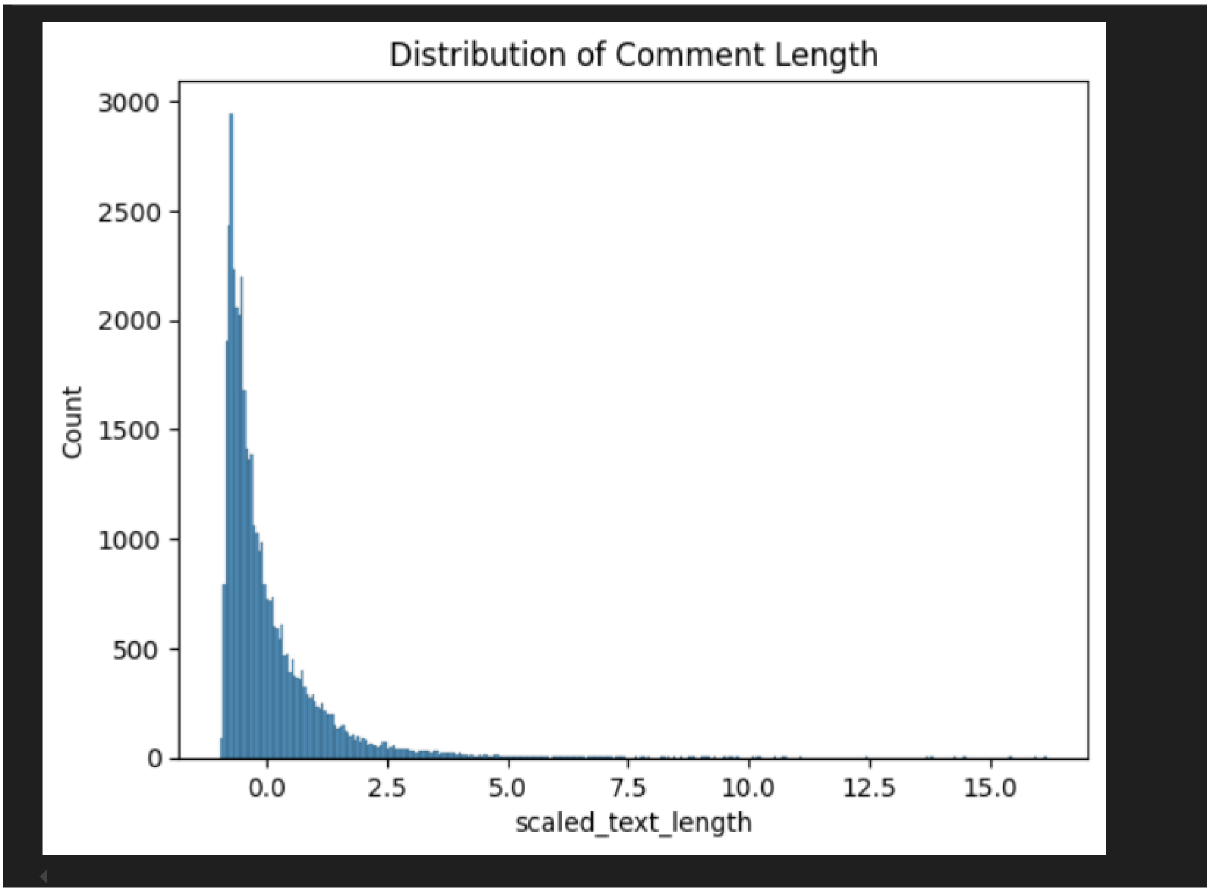
	comment	label
0	dalits are lowlives	N
1	gay people are a burden to our society	N
2	Arabs are not welcome	N
3	I'm not saying we should actually eliminate he...	N
4	bananas are for black people	N
...	...	...
41139	Indian culture is so backwards	N
41140	Women don't want equality, they just want to b...	N
41141	fuck covid	P
41142	This computer is garbage	P
41143	The only good Muslim is a dead one	N

41108 rows × 2 columns

Fig 1:- After Pre-processing rows reduced from 41145 to 41108

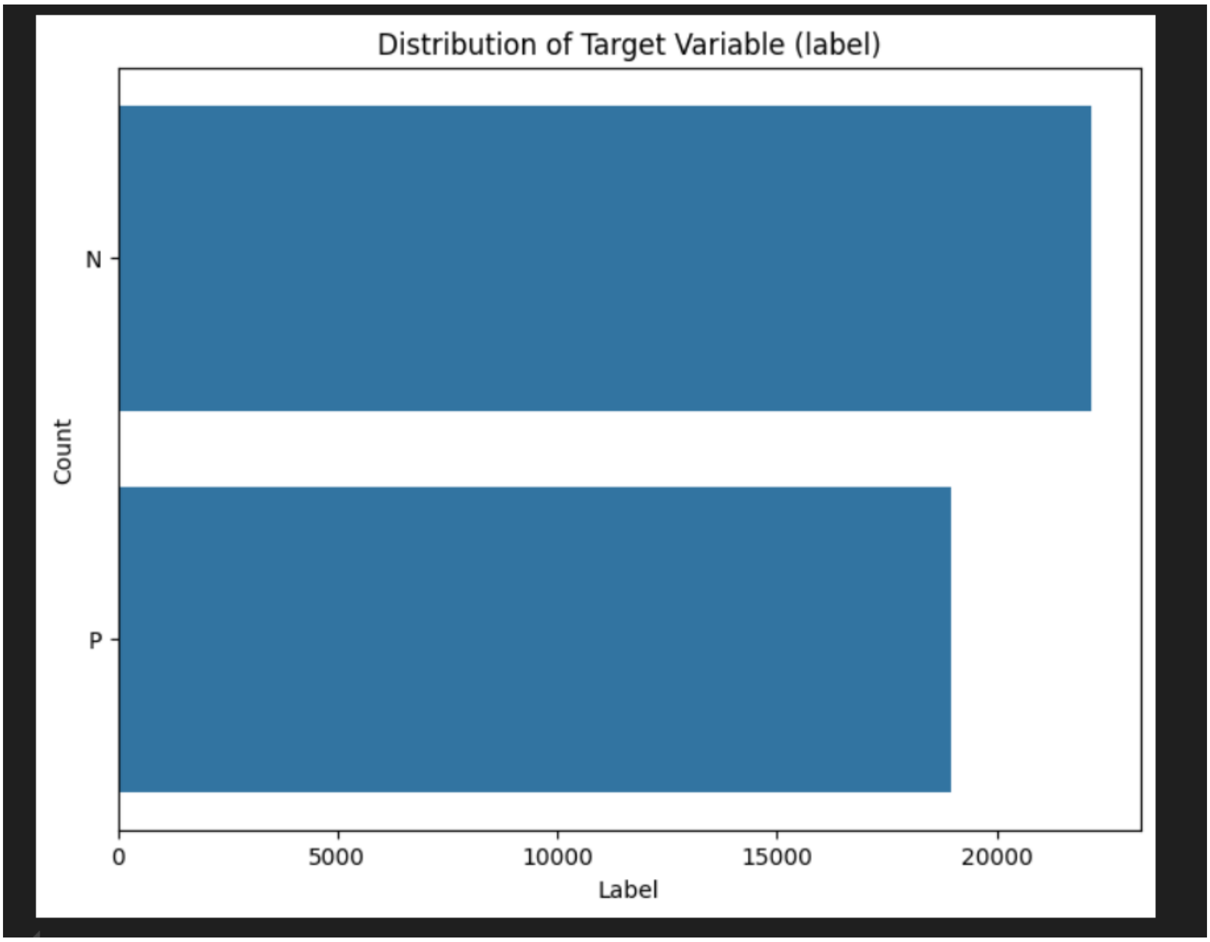
4.2 Distribution of Comment Length

- **Insights**:
  The first image illustrates the distribution of comment lengths (scaled). Most comments are short, with a steep drop-off for longer comments. This indicates that the dataset is dominated by concise text, typical of social media content.
- **Relevance:-**
  Short comments may lack context, which can pose challenges for machine learning models in detecting hate speech. Preprocessing steps such as context augmentation or external embeddings may be required to handle these short texts effectively.



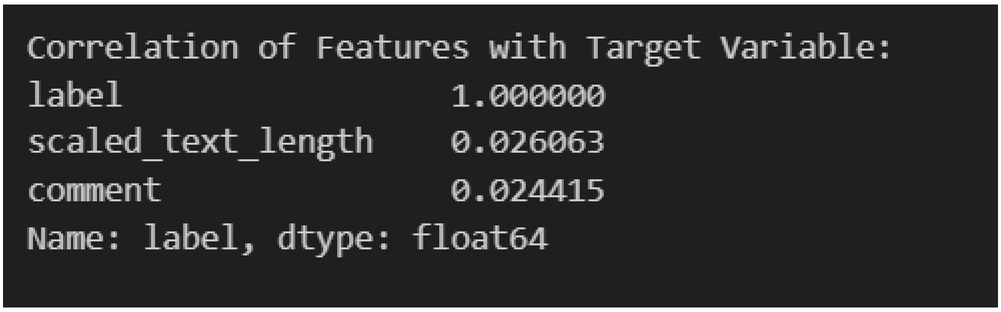
2. Distribution of Target Variable (Label)

- **Insights:**
  This bar chart shows the class distribution in the dataset. Labels "N" (neutral or Not a hate speech) dominate compared to "P" (possibly toxic), indicating a significant class imbalance.
- **Relevance:-**
  The imbalance may lead to biased model predictions favoring the majority class. Techniques such as oversampling, SMOTE (Synthetic Minority Oversampling Technique), or class-weight adjustments during training are necessary to improve model performance for the minority class.



3. Correlation of Features with Target Variable

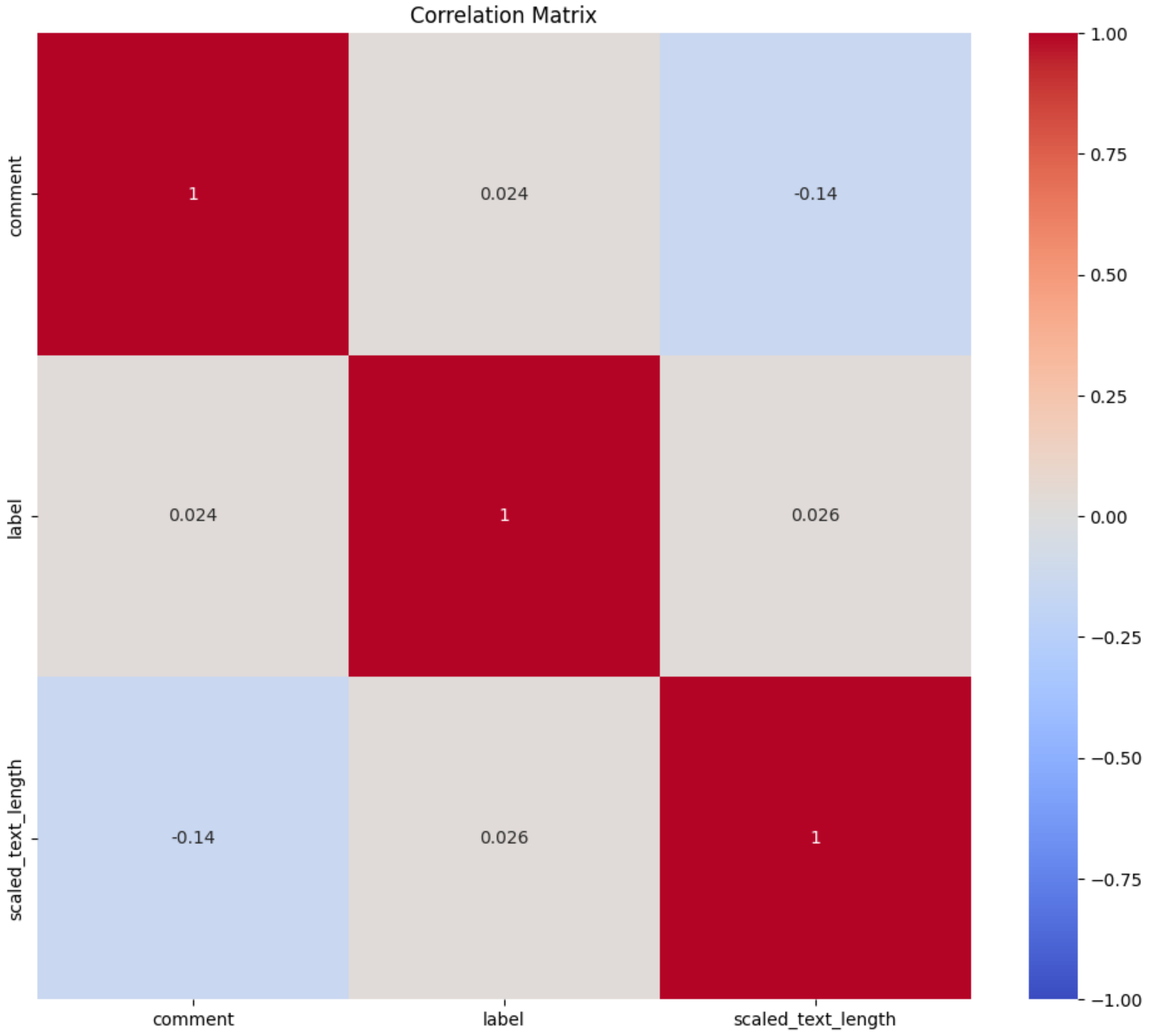
- **Insights:**
  The correlation matrix reveals weak correlations between the target variable (label) and features like scaled text length and comment. None of the features show a strong relationship with the target variable.
- **Relevance:-**
  This suggests that more complex features, such as embeddings (e.g., TF-IDF, BERT embeddings) or contextual sentiment analysis, need to be engineered to enhance predictive power. Simple statistical correlations may not suffice for accurate hate speech detection.



```
Correlation of Features with Target Variable:
label                 1.000000
scaled_text_length    0.026063
comment               0.024415
Name: label, dtype: float64
```

4. Correlation Matrix Visualization

- **Insights:**
  The heatmap visualization confirms the weak correlations between features and the target variable, with minimal inter-feature relationships.
- **Relevance:**
  This reinforces the need to create derived features or employ advanced models like transformers (e.g., BERT) that can capture non-linear and semantic relationships within the text.



4. Shap Value Visualization

SHAP (SHapley Additive exPlanations) values were used to interpret the predictions of our hate speech detection model, providing insights into feature importance and their contributions to the model's decisions.

Feature Importance (Bar Plot)

The bar plot ranks the top features by their average SHAP values, representing their impact on predictions. Key observations:

- Feature 4129 is the most influential, followed by 821, 5432, and 4677.
- Lower-ranked features like 7152 and 3707 have smaller but meaningful contributions.

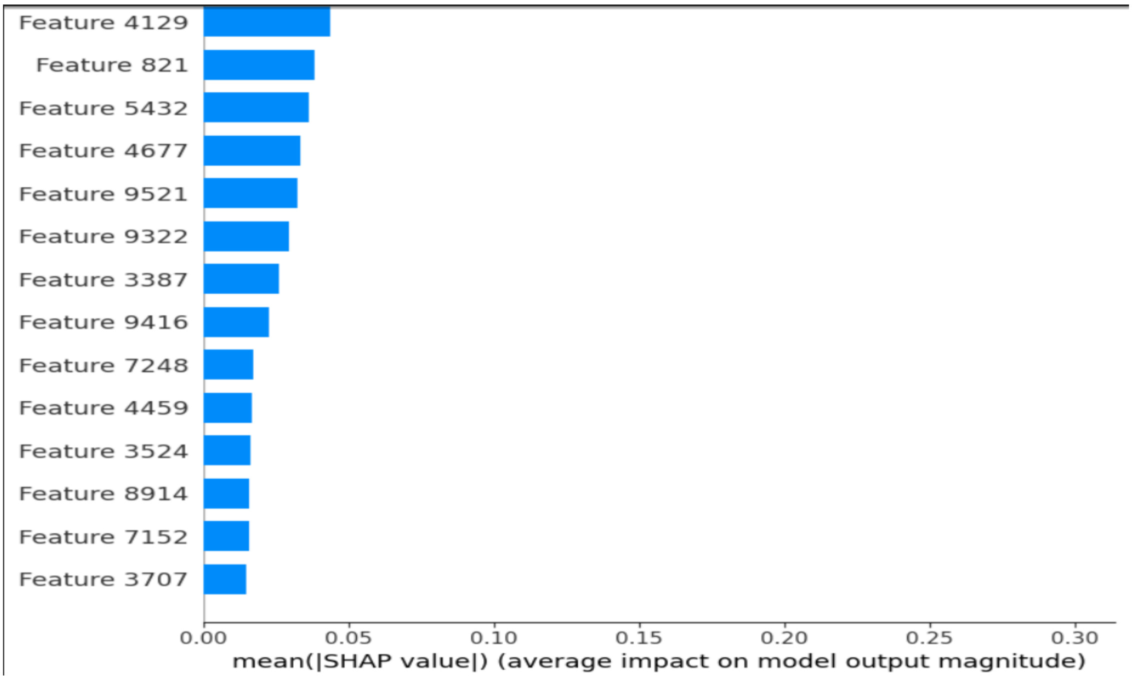
SHAP Summary Plot

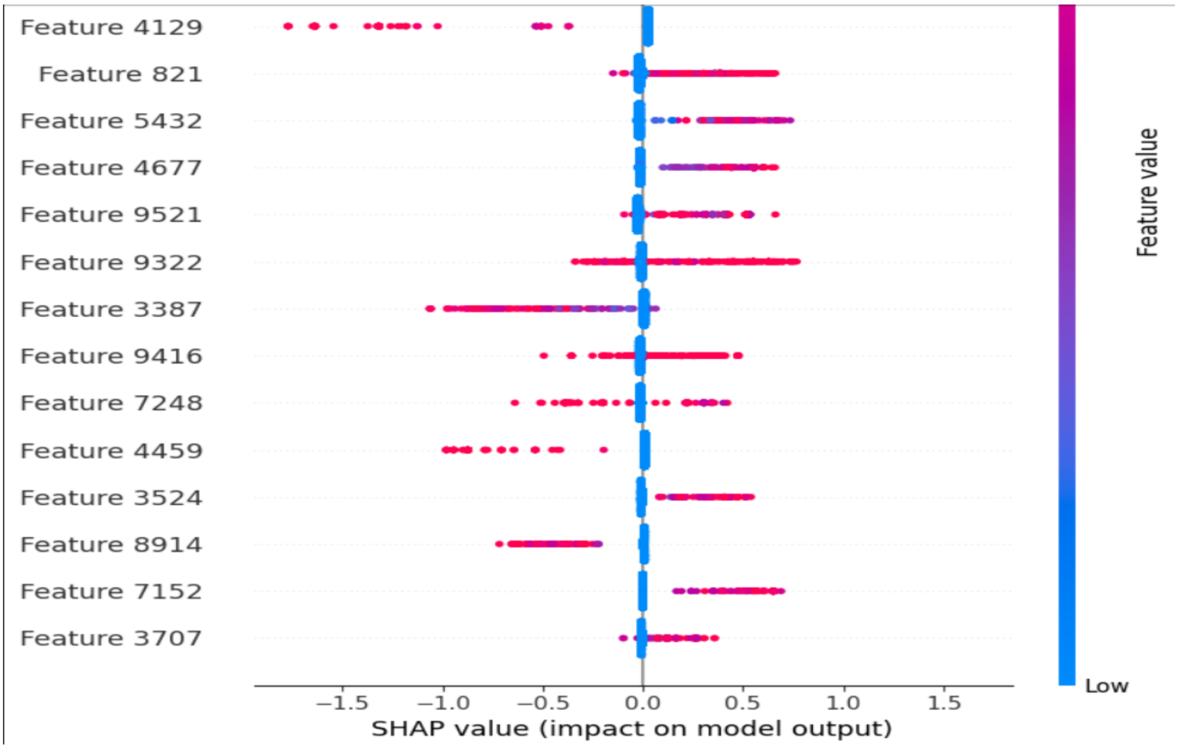
The summary plot shows the direction and magnitude of feature impacts:

- Positive SHAP values increase hate speech likelihood, while negative values decrease it.
- Color indicates feature value (red = high, blue = low). For example, higher values of Feature 4129 and 821 strongly push predictions toward hate speech.

Significance of SHAP Analysis

- **Interpretability**: Explains how features influence model decisions.
- **Bias Detection**: Helps identify potential biases in features or data.
- **Feature Engineering**: Guides refinement of impactful features.
- **Model Validation**: Ensures predictions align with domain expectations.





These analyses highlight the challenges and opportunities for your project, including the need for sophisticated text preprocessing, handling class imbalance, and engineering features capable of capturing context and nuance.

5 Literature Review

Detecting hate speech on social media has been the focus of numerous research projects due to its societal and ethical implications. Various methodologies have been proposed to address challenges such as data imbalance, contextual understanding, and linguistic diversity. Two notable approaches serve as exemplary contributions to the field:

Twitter Hate Speech Classification

Prakhar Prasad's project on Twitter hate speech classification demonstrates the effective use of BERT fine-tuning, which enhances the contextual understanding of text data—a critical requirement for accurately detecting hate speech. To address the common challenge of class imbalance in hate speech datasets, this project employs oversampling techniques, ensuring that minority classes (such as hate speech) are adequately represented during training. Additionally, the project incorporates a comprehensive text preprocessing pipeline, which includes:

- Text normalization to handle diacritics and ensure uniformity.
- Removal of irrelevant elements such as user handles, URLs, and digits.
- Advanced tokenization using NLTK's TweetTokenizer, which preserves hashtags and contractions for better semantic interpretation.

This project also highlights the importance of exploratory data analysis (EDA), particularly in analyzing class imbalances and identifying key terms associated with hate speech. In modeling, weighted logistic regression is used with randomized grid search for hyperparameter tuning, significantly improving detection accuracy for minority classes.

Toxic Comment Classification

Shishir Kumar’s toxic comment classification project takes a broader approach by categorizing comments into multiple toxic labels, such as "obscene," "insult," and "identity hate." This comprehensive method leverages insightful visualizations and word distributions to highlight overlaps between categories and guide feature selection. The project employs a simpler logistic regression model with a tf-idf vectorizer for text representation. While less advanced than Prasad's use of BERT, the project excels in providing actionable insights through detailed EDA and category-specific analyses.

These studies underscore the importance of advanced preprocessing, thoughtful handling of imbalanced datasets, and the strategic use of machine learning techniques in creating robust hate speech detection models. By incorporating these proven methodologies, this project aims to advance the field further while addressing the challenges unique to hate speech detection in multilingual and dynamic social media contexts.

Note:-Milestone2 contains detailed Literature review

Link:-

https://drive.google.com/file/d/1eDZCdAzen-wAxJL8FoK0rhZBUaPN9CN2/view?usp=drivesdk

6 Methodology

1) Data Collection:

The project begins with the collection of a comprehensive social media dataset containing user comments labeled as "hate speech," or "non a hate speech." The dataset, sourced from platforms like Kaggle, includes multilingual text data to capture the diverse linguistic nature of social media. Ensuring data quality is critical, with steps taken to address issues such as missing values, duplicates, and noisy data.

2) Exploratory Data Analysis (EDA):

Following data collection, EDA is conducted to understand the dataset's structure and identify key patterns. Key tasks include:

- Analyzing class distributions to address imbalances.
- Visualizing text length, word frequency, and key term distributions.
- Identifying correlations between features such as sentiment scores and hate speech labels.
- Detecting anomalies and inconsistencies to guide further preprocessing.

3) Feature Engineering:

After EDA, features are engineered to improve model performance. Techniques include:

- Tokenizing and vectorizing text data using methods like TF-IDF and word embeddings.
- Extracting linguistic features such as sentiment polarity, part-of-speech tags, and n-grams.
- Addressing data imbalance using resampling techniques like oversampling or SMOTE.

4) Model Selection:

A variety of machine learning models are employed and evaluated for their effectiveness in hate speech detection:

- **RandomForest:**
  An ensemble learning method that builds multiple decision trees and aggregates their predictions. Random Forest is highly effective in handling imbalanced datasets and identifying key features contributing to predictions. It provides feature importance scores to guide model refinement.

- **NeuralNetwork:-(MLP)**
  A multi-layer perceptron (MLP) is implemented to learn complex patterns in text data. By using dense layers and activation functions, MLPs can capture intricate relationships between features, making them suitable for nuanced tasks like hate speech detection.
- **NaiveBayes(Multinomial):**
  A probabilistic algorithm that excels in text classification tasks. It calculates the probability of each class based on word frequencies, making it a simple yet effective model for initial benchmarking.
- **Logistic-Regression:**
  A baseline model that predicts the probability of hate speech based on a linear combination of features. While simple and interpretable, logistic regression serves as a benchmark to compare the performance of more complex models.
- **GradientBoosting:**
  An advanced ensemble learning technique that builds a sequence of decision trees, optimizing for classification accuracy. Gradient boosting is particularly effective in capturing non-linear relationships and is robust to overfitting.
- **XGBoost**

  XGBoost (Extreme Gradient Boosting) is a powerful ensemble learning algorithm that builds a series of decision trees, where each tree corrects the errors of the previous ones. It is highly efficient, scalable, and effective for structured data tasks. XGBoost uses gradient boosting techniques to minimize errors and optimize performance.

5) Training and Testing:
The dataset is divided into training and testing sets, ensuring robust model training and evaluation. Cross-validation is applied to mitigate overfitting and ensure consistent performance across different data splits.

6) Model Evaluation:
Models are evaluated using metrics such as:

- **Accuracy:** Measures overall correctness.

- **Precision, Recall, and F1-Score:** Crucial for assessing model performance on imbalanced datasets.
- **ROC-AUC:** Evaluates the model's ability to distinguish between classes.

7) Hyperparameter Tuning:

Grid search and randomized search techniques are employed to optimize model parameters, enhancing predictive accuracy and reducing overfitting.

8) Prediction:

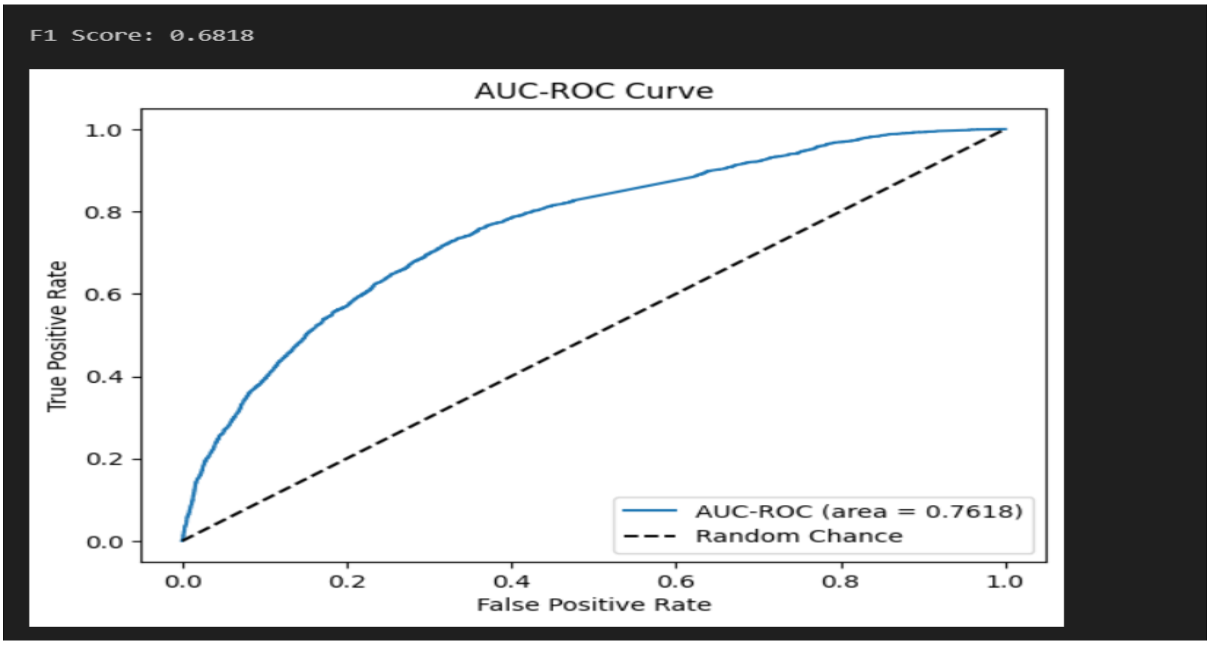
Once trained and evaluated, the model is deployed to classify new, unseen social media comments. By leveraging historical patterns and engineered features, the model generates classifications, enabling the identification of hate speech with high precision and recall. These predictions empower platforms and stakeholders to foster healthier online interactions and mitigate the spread of hate speech.

7 Implementation

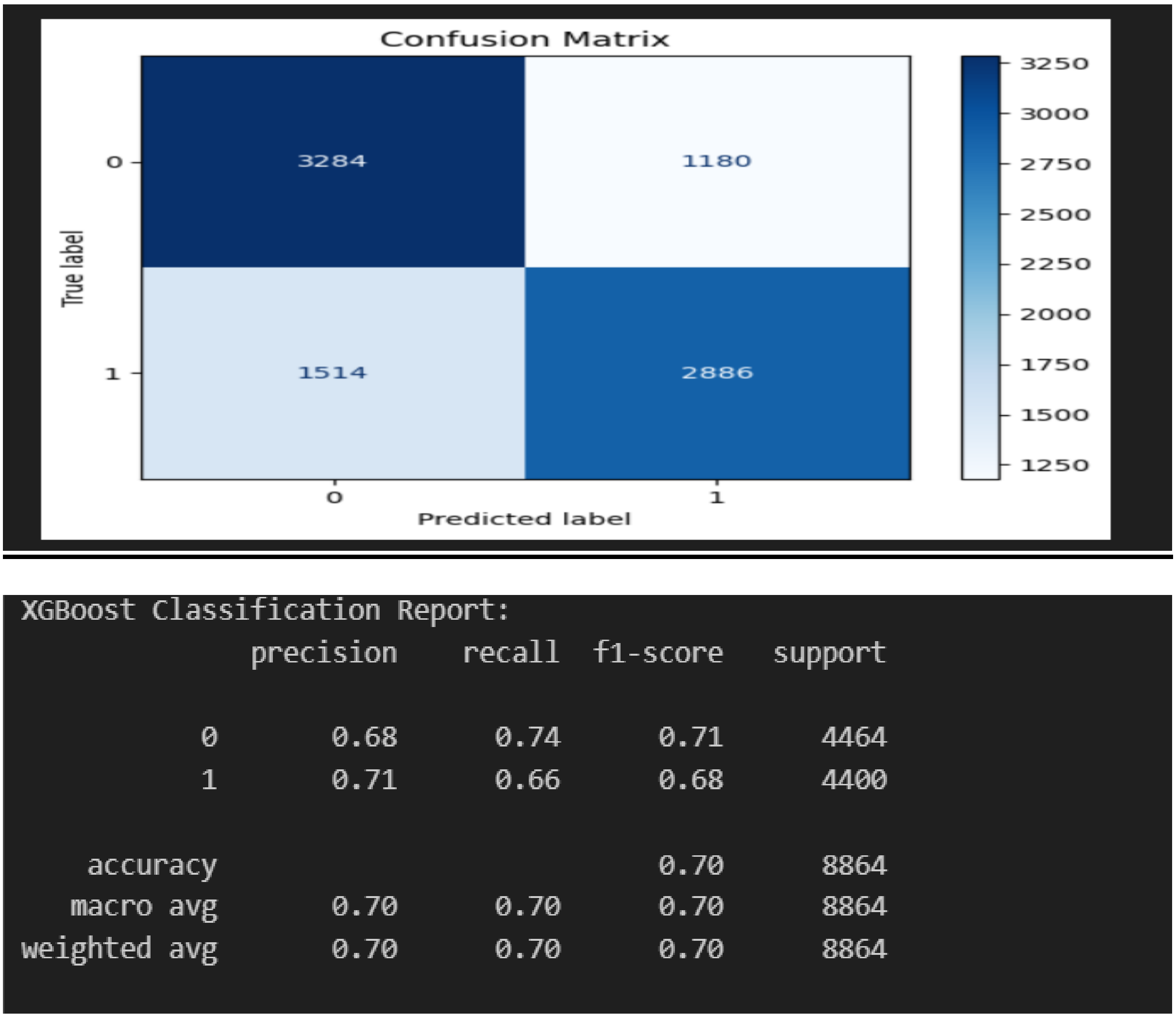
Link to the code:-

https://drive.google.com/file/d/15XSfI4dWFON94IUuyYUQU-_oazs0wi03/view?usp=drive_link

8 Results

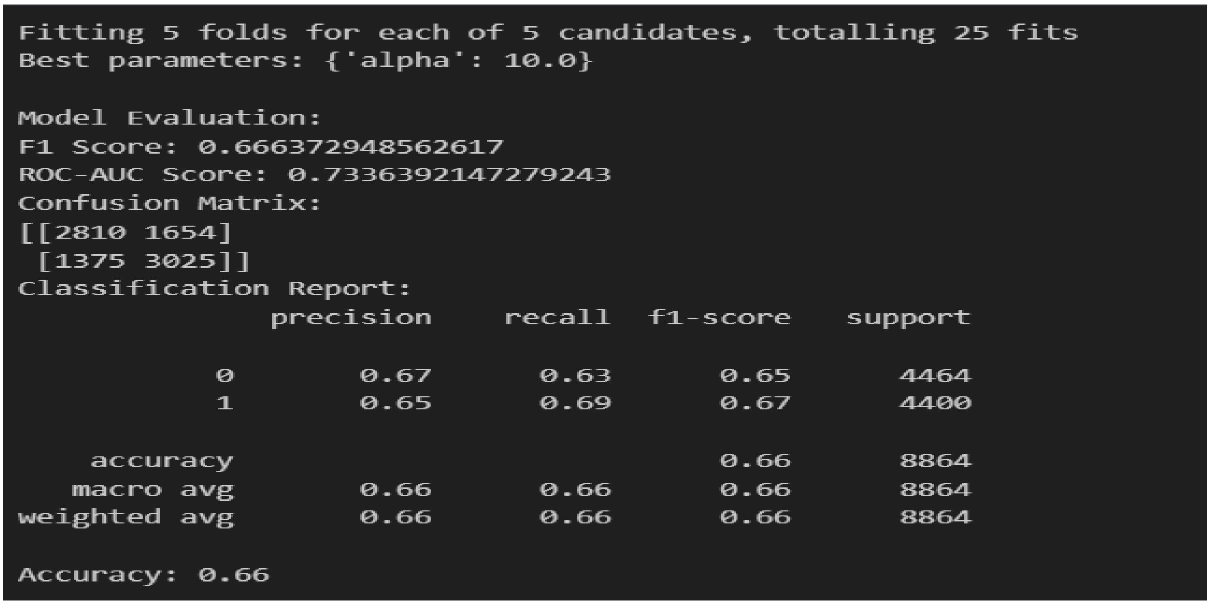


```
F1 Score: 0.6818
```



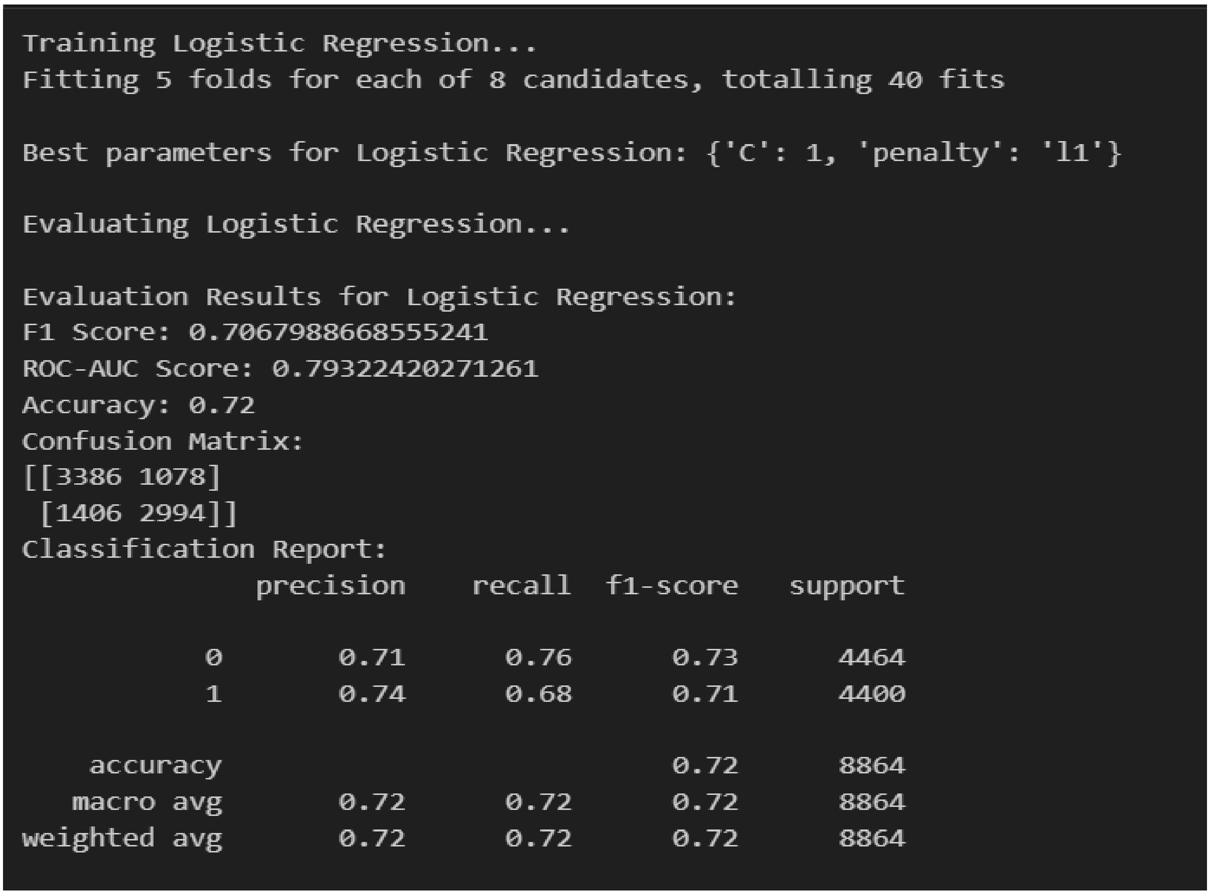
```
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.74      0.71      4464
           1       0.71      0.66      0.68      4400

    accuracy                           0.70      8864
   macro avg       0.70      0.70      0.70      8864
weighted avg       0.70      0.70      0.70      8864
```

Fig 8.1 Xboost classification Report



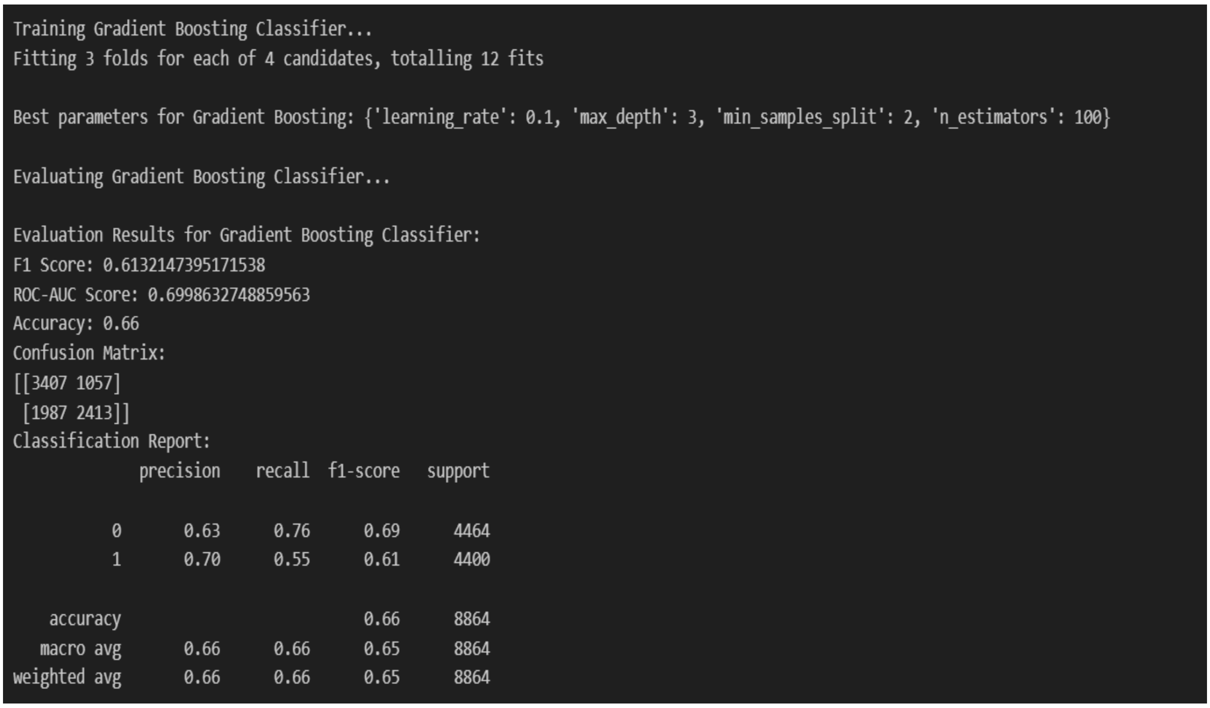
```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
Best parameters: {'alpha': 10.0}

Model Evaluation:
F1 Score: 0.666372948562617
ROC-AUC Score: 0.7336392147279243
Confusion Matrix:
[[2810 1654]
 [1375 3025]]
Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.63      0.65      4464
           1       0.65      0.69      0.67      4400

    accuracy                           0.66      8864
   macro avg       0.66      0.66      0.66      8864
weighted avg       0.66      0.66      0.66      8864

Accuracy: 0.66
```

Fig 8.2 Naïve_Bayes classification Report



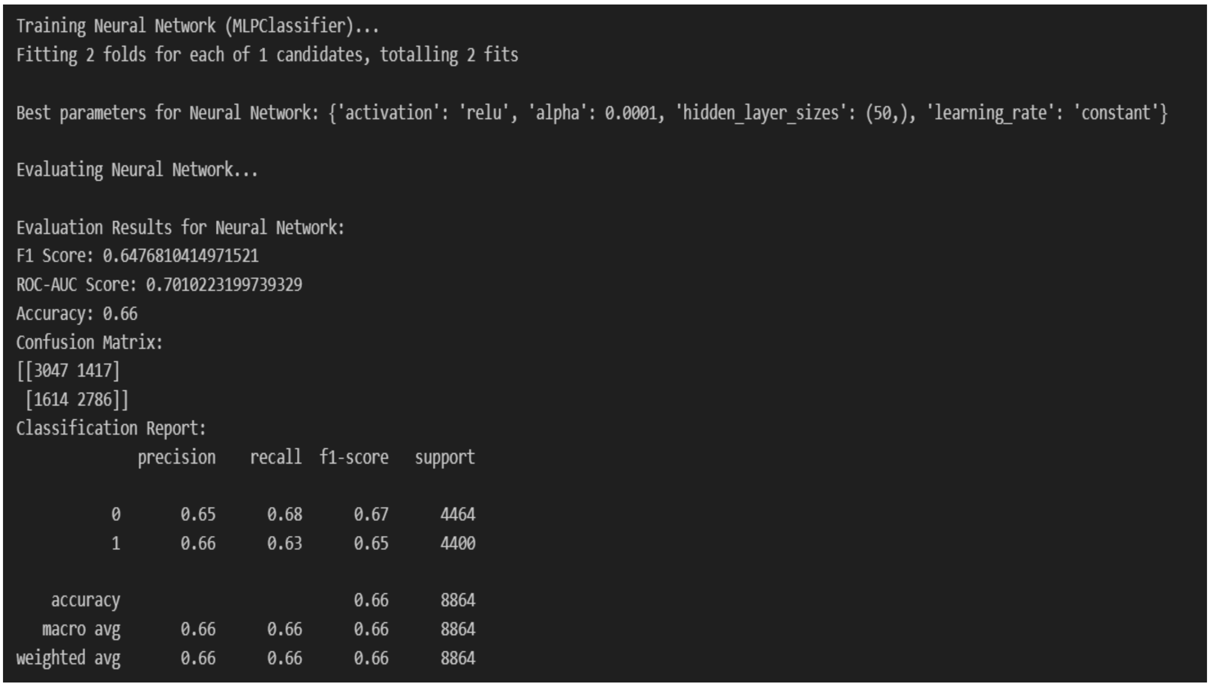
```
Training Logistic Regression...
Fitting 5 folds for each of 8 candidates, totalling 40 fits

Best parameters for Logistic Regression: {'C': 1, 'penalty': 'l1'}

Evaluating Logistic Regression...

Evaluation Results for Logistic Regression:
F1 Score: 0.7067988668555241
ROC-AUC Score: 0.79322420271261
Accuracy: 0.72
Confusion Matrix:
[[3386 1078]
 [1406 2994]]
Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.76      0.73      4464
           1       0.74      0.68      0.71      4400

    accuracy                           0.72      8864
   macro avg       0.72      0.72      0.72      8864
weighted avg       0.72      0.72      0.72      8864
```

Fig8.3 Logistic Regression Classification Report



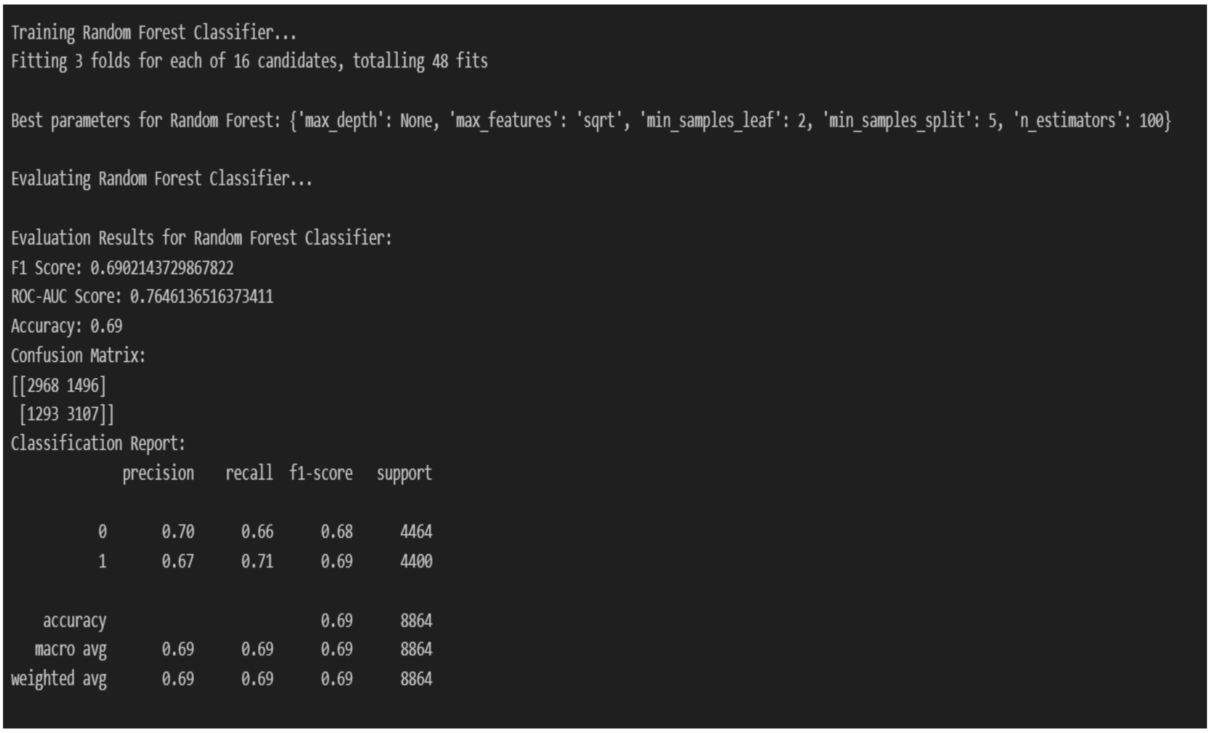
```
Training Gradient Boosting Classifier...
Fitting 3 folds for each of 4 candidates, totalling 12 fits

Best parameters for Gradient Boosting: {'learning_rate': 0.1, 'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 100}

Evaluating Gradient Boosting Classifier...

Evaluation Results for Gradient Boosting Classifier:
F1 Score: 0.6132147395171538
ROC-AUC Score: 0.6998632748859563
Accuracy: 0.66
Confusion Matrix:
[[3407 1057]
 [1987 2413]]
Classification Report:
              precision    recall  f1-score   support

           0       0.63      0.76      0.69      4464
           1       0.70      0.55      0.61      4400

    accuracy                           0.66      8864
   macro avg       0.66      0.66      0.65      8864
weighted avg       0.66      0.66      0.65      8864
```

Fig 8.4 Gradient_boosting Classification Report



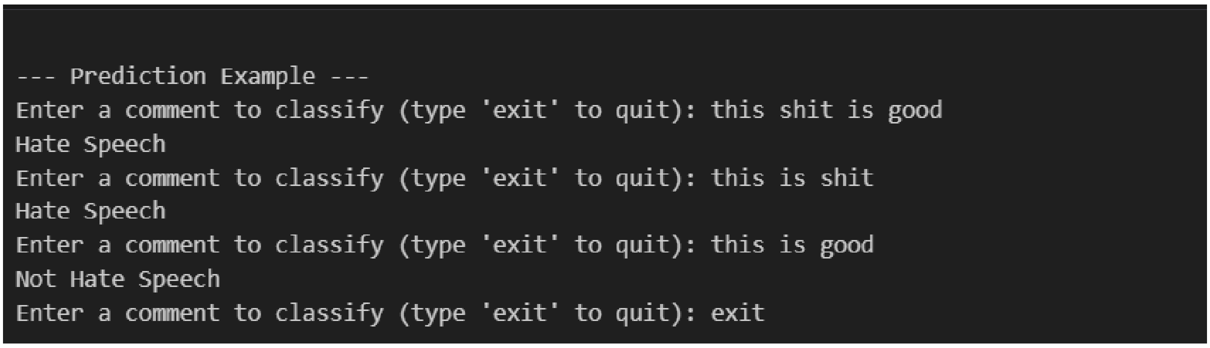
```
Training Neural Network (MLPClassifier)...
Fitting 2 folds for each of 1 candidates, totalling 2 fits

Best parameters for Neural Network: {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (50,), 'learning_rate': 'constant'}

Evaluating Neural Network...

Evaluation Results for Neural Network:
F1 Score: 0.6476810414971521
ROC-AUC Score: 0.7010223199739329
Accuracy: 0.66
Confusion Matrix:
[[3047 1417]
 [1614 2786]]
Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.68      0.67      4464
           1       0.66      0.63      0.65      4400

    accuracy                           0.66      8864
   macro avg       0.66      0.66      0.66      8864
weighted avg       0.66      0.66      0.66      8864
```

Fig 8.5 Neural Network(MLP) Classification Report



```
Training Random Forest Classifier...
Fitting 3 folds for each of 16 candidates, totalling 48 fits

Best parameters for Random Forest: {'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}

Evaluating Random Forest Classifier...

Evaluation Results for Random Forest Classifier:
F1 Score: 0.6902143729867822
ROC-AUC Score: 0.7646136516373411
Accuracy: 0.69
Confusion Matrix:
[[2968 1496]
 [1293 3107]]
Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.66      0.68      4464
           1       0.67      0.71      0.69      4400

    accuracy                           0.69      8864
   macro avg       0.69      0.69      0.69      8864
weighted avg       0.69      0.69      0.69      8864
```

Fig 8.6 Random Forest Classification Report



```
--- Prediction Example ---
Enter a comment to classify (type 'exit' to quit): this shit is good
Hate Speech
Enter a comment to classify (type 'exit' to quit): this is shit
Hate Speech
Enter a comment to classify (type 'exit' to quit): this is good
Not Hate Speech
Enter a comment to classify (type 'exit' to quit): exit
```

Fig 8.7 Testing Model's Classification

9 Conclusion

- This project focused on evaluating the performance of various machine learning models, including Random Forest, MLP (Multi-Layer Perceptron), Naive Bayes, Logistic Regression, Gradient Boosting, and SVM, for the detection of hate speech in social media comments. The study began with the collection and preprocessing of a labeled dataset, incorporating advanced text cleaning and feature engineering techniques to ensure high-quality data for model training and evaluation.
- Our analysis revealed the distinct advantages and limitations of each model in hate speech detection. Logistic Regression provided a simple and interpretable baseline, though its performance was constrained by its inability to capture complex, non-linear patterns in the data. Random Forest and Gradient Boosting demonstrated strong performance in handling imbalanced datasets and identifying key features, with Gradient Boosting excelling in capturing subtle relationships through iterative learning. MLP, leveraging its neural network-based structure, showed potential in identifying nuanced patterns in text data but required significant computational resources and careful tuning for optimal results. Naive Bayes offered efficiency and simplicity for text classification tasks but was less effective in capturing contextual nuances compared to more advanced methods. SVM, with its ability to handle high-dimensional data and non-linear relationships, achieved competitive results, particularly when paired with optimized kernel functions and hyperparameters.
- The findings emphasized the absence of a universal model for hate speech detection. The choice of the best-performing model depends on various factors, including the characteristics of the dataset, computational constraints, the importance of interpretability, and the trade-off between model complexity and accuracy. Future work could explore ensemble methods or hybrid models that combine the strengths of multiple approaches to improve detection accuracy. Additionally, incorporating

domain-specific features or external data sources, such as sentiment analysis or cultural context, could further enhance model robustness and generalization.

- Despite challenges such as data imbalance and the intricacies of social media language, this project contributes meaningful insights to the ongoing efforts in hate speech detection. It underscores the importance of a multi-faceted approach and opens avenues for continued research and development to foster safer online communities.

10 References

1. https://www.kaggle.com/
2. https://ieeexplore.ieee.org/Xplore/home.jsp
3. https://chatgpt.com/
4. https://www.kaggle.com/code/prakharprasad/twitter-hate-speech-classification
5. https://www.youtube.com
6. https://www.kaggle.com/code/jarvis11/toxic-comment-classification-eda/comments

Link to the Video Presentation:-

https://drive.google.com/file/d/1e0IORD5DXYFQvv9f3fVFgnGdxW_-hsBV/view?usp=drivesdk

Contribution of Team Members in the Project

1. **Dhruv Patel**: Responsible for data preprocessing, building and training the Multinomial Naïve Bayes model, and preparing the project report.
2. **Kenil Aviya**: Focused on developing and training the Random Forest, Neural Network (MLP), and Gradient Boosting models.
3. **Yash Shah**: Specialized in building and optimizing Gradient Boosting and XGBoost models.
4. **Rahul**: Conducted exploratory data analysis, developed and trained the Logistic Regression model, and created the PowerPoint presentation