Hate Speech Detection

Course Details

- **Course Code and Name:** DATA602/BIOI602/MSML602: Principles of Data Science
- **Semester and Year:** Fall 2024
- **Instructor Name:** Dr. Fardina Fathmiul Alam

Group Project Information

- **Group Project Name:** Hate Speech Detection
- **URL to Final Tutorial:** https://github.com/GowthamT-1610/Hate-Speech-Detection
- **URL to Git page:** https://gowthamt-1610.github.io/Hate-Speech-Detection/
- **Group Members:**
  - Tadikamalla Gowtham Krishna (UID: 121321909)
  - Raahul Narayana Reddy Kummitha (UID: 121109521)
  - Sriyank Sagi (UID: 121302335)
  - Venakata SatySai Maruti Kameshwar Modali (UID: 121306050)
  - Dhanush Garikapati (UID: 121324924)
- **Date of Submission:** 12/14/2024

Contributions

A: Project Idea

Sriyank Sagi — Researched various project ideas, discussed with group members, and collectively decided on an NLP-based project focused on Hate Speech Detection.

B: Dataset Curation and Preprocessing

Sriyank Sagi — Removed duplicates and null values, replaced them with correct data, checked that the context contained only text and no extraneous characters. Used the ‘langdetect‘ library to ensure only English sentences were retained in the dataset.

C: Data Exploration and Summary Statistics

Raahul Narayana Reddy Kummitha — Identified critical tests for the project, performed hypothesis testing, and plotted results to evaluate the null hypothesis, which was rejected. Illustrated findings with box plots, conducted Chi-Square tests, and analyzed data distribution.

D: ML Algorithm Design/Development

Tadikamalla Gowtham Krishna — Analyzed the data thoroughly and chose two machine learning models: the DistilBERT model and the LSTM model. For DistilBERT, used the pre-trained ‘distilbert-base-uncased‘ model and split the data into 80 percent training and 20 percent testing. Performed tokenization for both datasets and created data loaders with a batch size of 16. Selected the AdamW optimizer for training. For the LSTM model, tokenized the data and selected ReLU and Sigmoid activation functions. Used the Adam optimizer to optimize model performance.
Raahul Narayana Reddy Kummitha — Naive Bayes, trained with TF-IDF vectorization and an 80/20 train-test split, efficiently leveraged feature independence for classification, with performance evaluated using precision, recall, F1-score, accuracy, confusion matrices, and visualized through bar charts.

E: ML Algorithm Training and Test Data Analysis

Tadikamalla Gowtham Krishna —

- **DistilBERT**: Completed tokenization, created data loaders, and trained the model for three epochs using the AdamW optimizer. Printed and plotted loss for each epoch.
- **LSTM**: Split data into 80 percent training and 20 percent testing. Trained for five epochs and calculated accuracy and loss for evaluation.

Raahul Narayana Reddy Kummitha —

- **Naive Bayes**: The dataset was vectorized using TF-IDF, split into 80 percent training and 20 percent testing, and trained using the Multinomial Naive Bayes algorithm, well-suited for text classification.

F: Visualization, Result Analysis, Conclusion

Venakata SatySai Maruti Kameshwar Modali — Visualized accuracy metrics and evaluation matrices such as confusion matrices and AUROC curves. Compared models and concluded that DistilBERT performed best.

G: Final Tutorial Report Creation

Dhanush Garikapati — Prepared the final project report and created the GitHub Pages tutorial. Also contributed to model selection and training.

Introduction

What is your topic?
The topic of this project is **Hate Speech Detection**. This project focuses on identifying hateful text from social media platforms using Natural Language Processing (NLP) techniques and machine learning models.

What is the main motivation for your work?
The main motivation for this project stems from the growing prevalence of hate speech on social media. Social media text often contains hateful content disguised with emoticons, slang, or modern trends, making it challenging to detect automatically. This dataset is designed to train machine learning models to identify hate speech effectively. It reflects current social media trends and aids in developing automated systems to filter out such content. The dataset is neutralized, preprocessed, and composed of text sentences categorized as hateful (”1”) or non-hateful (”0”). This work provides a benchmark dataset for hate speech detection, helping practitioners in Deep Learning (DL) and NLP while maintaining compliance with policy guidelines to mitigate cyber harm.

What question(s) are you trying to answer with your analysis?
This project aims to address the following key questions:

1. Can we effectively classify social media text as hateful or non-hateful using modern machine learning techniques such as DistilBERT, LSTM and Naive Bayes models?
2. What are the most reliable features or characteristics in social media text that help distinguish hate speech from non-hateful content?
3. How do modern NLP models perform in identifying hate speech across diverse text inputs, including emoticons, slang, and mixed sentiments?

Why is answering those questions important?
Answering these questions is crucial for several reasons:

- **Social Impact:** Hate speech on social media poses a significant threat to societal harmony, leading to online abuse, cyberbullying, and mental health issues. Identifying and filtering such content contributes to a safer digital space.
- **Policy Compliance:** Social media platforms are required to comply with anti-cyberbullying and hate speech regulations. Automated hate speech detection systems can help platforms ensure compliance effectively.
- **Improved Moderation:** Automated systems can assist human moderators by reducing workload and providing consistent and unbiased moderation.
- **Advancing NLP Research:** This dataset and analysis contribute to advancements in NLP and machine learning by serving as benchmarks and encouraging further improvements in text classification techniques.

Data Curation

What is the source of your data?

The dataset used for this project was sourced from a curated collection of social media comments provided on open-access platforms(kaggle). The dataset consists of text sentences categorized as either hateful ("1") or non-hateful ("0"). These categories reflect the sentiment and nature of the social media text, making it ideal for training machine learning models for hate speech detection.

What is your dataset, and what does it contain?

The dataset is a collection of text comments extracted from social media platforms. Each comment is labeled as either hateful ("1"), which includes derogatory, offensive, or abusive language, or non-hateful ("0"), which contains no form of hate speech. The dataset also incorporates diverse text samples, including emoticons, slang, and modern communication trends, making it representative of real-world social media challenges.

How was the data prepared for analysis?

The dataset underwent several preprocessing steps to ensure its readiness for analysis. Duplicate and null values were identified and removed to eliminate redundancy and inconsistencies. Text cleaning was performed to remove extraneous characters, such as special symbols or numbers, leaving only meaningful text. Using the 'langdetect' library, only English comments were retained, aligning with the project's focus. The dataset labels ("1" for hateful and "0" for non-hateful) were also cross-checked to ensure their correctness for accurate classification.

How was the data transformed for analysis?

The cleaned dataset was organized into a pandas DataFrame with two columns: "Text," containing the cleaned social media comments, and "Label," indicating whether the comment is hateful ("1") or non-hateful ("0"). The text data was tokenized into sequences of integers, where each integer represents a unique word in the vocabulary. To ensure uniformity in sequence length, all sequences were padded or truncated to a maximum length of 100 tokens. Finally, the dataset was split into 80

How was the data stored and queried?

The processed dataset was stored as a CSV file, making it accessible for manipulation and sharing. During analysis, the dataset was loaded into a pandas DataFrame, allowing efficient querying, transformation, and integration into machine learning workflows.

Why was this data preparation necessary?

These preprocessing and transformation steps were crucial to ensure the dataset was clean, consistent, and suitable for analysis. They addressed common issues such as missing or duplicate data, irrelevant characters, and inconsistent labeling, which could have negatively impacted the performance of the machine learning models.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) was performed to gain insights into the dataset and understand its structure, distribution, and key characteristics. The following steps summarize the EDA process and findings:

Label Distribution:

The dataset consists of text labeled as either hateful (”1”) or non-hateful (”0”). During EDA, the label distribution was analyzed, revealing that the dataset was imbalanced, with more non-hateful comments compared to hateful ones. This imbalance was addressed during the machine learning process.

Statistical Tests and Hypothesis Testing:

Various statistical tests were conducted to identify key attributes of the dataset:

- Hypothesis testing was performed to examine whether there was a significant difference in the frequency of hateful and non-hateful comments. The null hypothesis, stating no significant difference, was rejected based on the results.
- A Chi-Square test was performed to analyze the relationship between specific words and their association with hateful or non-hateful labels, confirming the statistical significance of certain terms in identifying hate speech.

Distribution Analysis:

The length of comments (in terms of word count) was analyzed, showing that hateful comments tend to be shorter than non-hateful ones. A box plot was created to visualize this distribution, highlighting the difference in comment lengths between the two classes.

Key Visualizations:

EDA included the generation of several visualizations to summarize the dataset’s characteristics:

- Bar plots showing the frequency of hateful vs. non-hateful labels.
- A word cloud to visualize the most common words in each category (hateful and non-hateful comments), providing a clear representation of the textual trends.
- A box plot illustrating the distribution of comment lengths across the two labels.

Findings and Insights:

The EDA revealed that hateful comments often include specific keywords or slang, while non-hateful comments tend to be longer and more formal. Additionally, the dataset’s imbalance necessitated further preprocessing to ensure fair model training. These insights guided the selection of machine learning models and preprocessing techniques to improve performance and accuracy.

Conclusion:

The EDA provided critical insights into the dataset, allowing for informed decisions in the subsequent stages of the project. By understanding the dataset’s characteristics, we ensured that the preprocessing and modeling approaches were tailored to effectively address the challenges of hate speech detection.

Primary/Machine Learning Analysis

Based on the results of our Exploratory Data Analysis (EDA), we selected three machine learning techniques: **DistilBERT**, **LSTM (Long Short-Term Memory)**, and **Naive Bayes** models. These models were chosen for their proven ability to process and analyze textual data, particularly in the context of Natural Language Processing (NLP) tasks. All three techniques align with the project's objective of effectively classifying social media text as hateful or non-hateful.

Why were these techniques chosen?
DistilBERT is a smaller and faster variant of the BERT (Bidirectional Encoder Representations from Transformers) model. It is pre-trained on large corpora and excels at capturing the context of words in a sentence. Its ability to leverage transfer learning makes it an ideal choice for this project, as it can quickly adapt to the task of hate speech detection with minimal computational resources. LSTM, on the other hand, is a type of Recurrent Neural Network (RNN) that is well-suited for processing sequential data such as text. It is effective at capturing long-term dependencies in textual data, which is critical for understanding the context of words and phrases in sentences. Naive Bayes, being computationally efficient and straightforward, was chosen as a baseline model for comparison. It applies Bayes' theorem with an independence assumption between features, making it suitable for quick text analysis.

How do these techniques help answer the key questions?
Using the pre-trained DistilBERT model, we tokenized and transformed text data into embeddings that capture word relationships and context. The mechanism in DistilBERT enables the model to focus on critical parts of the input text, ensuring accurate classification of hateful versus non-hateful comments. The LSTM model processes the tokenized data to identify sequential patterns within the text. By analyzing relationships between words over time, LSTM detects subtle cues in language that differentiate hate speech from non-hateful content. Naive Bayes provided quick training and prediction times, enabling efficient initial exploration of text data distributions and feature importance through TF-IDF.

Implementation Details:
For DistilBERT, the pre-trained 'distilbert-base-uncased' model was used. The dataset was tokenized using the DistilBERT tokenizer, and data loaders were created with a batch size of 16. The AdamW optimizer was chosen for training, and the model was fine-tuned for three epochs. For LSTM, the text data was tokenized and padded to ensure uniform input length. The model used ReLU and Sigmoid activation functions and the Adam optimizer. The LSTM model was trained for five epochs, with accuracy and loss monitored at each step. For Naive Bayes, the dataset was vectorized using the TF-IDF approach, split into 80 percent training and 20 percent testing, and trained using the Multinomial Naive Bayes variant. While it performed well for initial exploration, it struggled with complex patterns compared to DistilBERT and LSTM.

Conclusion:
DistilBERT and LSTM were highly effective in addressing the project's objectives. DistilBERT provided excellent results due to its advanced contextual understanding, while LSTM offered a complementary perspective by analyzing sequential relationships. Naive Bayes, although limited in handling nuanced language patterns, served as an efficient baseline for understanding feature importance and text distributions.

Visualization

In this section, we present the key visualizations that provide insights into the analysis, preprocessing, and performance of the hate speech detection models.

1. **Text Length Distribution by Label:**

This plot shows the distribution of the number of words in texts labeled as Hate Speech and Non-Hate Speech, indicating that Non-Hate Speech tends to have longer text lengths.

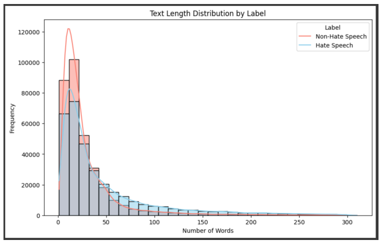


Figure 1: Text Length Distribution by Label

2. **Cumulative Distribution of Text Lengths by Label:**

This plot illustrates the cumulative percentage of text lengths, showing that shorter text lengths dominate in both categories.

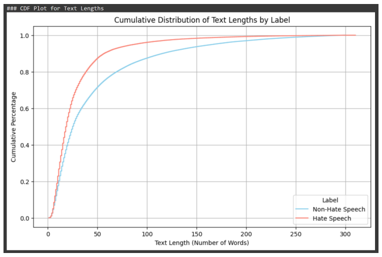


Figure 2: Cumulative Distribution of Text Lengths by Label

3. **Mean Sentiment Score by Speech Type:**

This plot compares the mean sentiment scores for Hate Speech and Non-Hate Speech, with error bars indicating statistical significance.

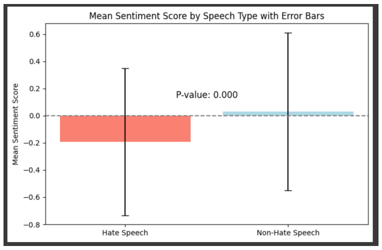
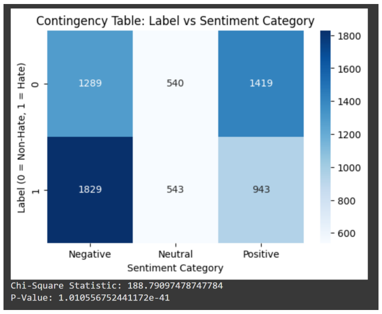


Figure 3: Mean Sentiment Score by Speech Type with Error Bars

4. Contingency Table: Label vs Sentiment Category:

This heatmap displays the relationship between text labels and sentiment categories, highlighting significant differences across categories.



```
Chi-Square Statistic: 188.79097478747784
P-Value: 1.010556752441172e-41
```

Figure 4: Contingency Table: Label vs Sentiment Category

5. Distribution of Sentiment Scores:

This histogram shows the distribution of sentiment scores for both Hate Speech and Non-Hate Speech, emphasizing patterns in sentiment polarity.

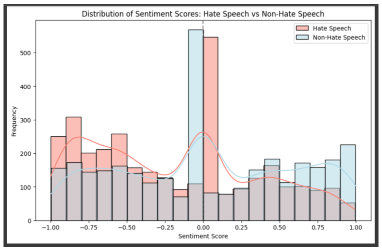
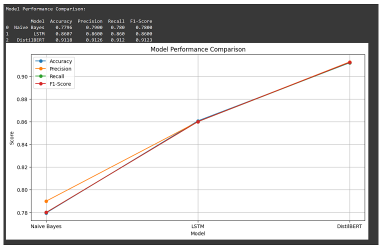


Figure 5: Distribution of Sentiment Scores: Hate Speech vs Non-Hate Speech

6. Model Performance Comparison:

This plot compares the performance metrics (Accuracy, Precision, Recall, F1-Score) of Naive Bayes, LSTM, and DistilBERT, with DistilBERT emerging as the best model.



```
Model Performance Comparison:

         Model  Accuracy  Precision  Recall  F1-Score
0  Naive Bayes    0.7796     0.7900   0.780    0.7800
1         LSTM    0.8607     0.8600   0.860    0.8600
2   DistilBERT    0.9118     0.9126   0.912    0.9123
```

Figure 6: Model Performance Comparison

Insights and Conclusions

This project focused on hate speech detection using advanced Natural Language Processing (NLP) techniques, specifically DistilBERT and LSTM and Naive Bayes models. By combining modern machine learning methodologies with robust data preprocessing and analysis, the project successfully addressed the challenges posed by real-world social media text. These efforts demonstrate how automation can play a vital role in moderating online content and promoting safer digital spaces.

For an uninformed reader:
This project provides a clear understanding of the problem of hate speech detection and the steps involved in solving it. An uninformed reader gains insights into:

- The significance of identifying and addressing hate speech to combat online abuse and foster inclusive communication.
- How datasets are prepared for analysis, including data cleaning, filtering, and preprocessing to ensure quality inputs for machine learning models.
- The importance of choosing effective machine learning models, such as DistilBERT, LSTM and Naive Bayes, to handle the complexities of textual data.

Through detailed explanations and visualizations, the report equips an uninformed reader with foundational knowledge about hate speech detection using machine learning.

For a reader familiar with the topic:
A knowledgeable reader gains valuable insights from this project, particularly in comparing the effectiveness of different NLP models. Key takeaways include:

- The comparative analysis of DistilBERT, LSTM and Naive Bayes models, highlighting their respective strengths in handling text classification tasks.
- Strategies for addressing challenges such as data imbalance, ensuring fair and unbiased model training.
- The detailed evaluation of models using performance metrics like accuracy, loss trends, confusion matrices, and AUROC curves, which provide a nuanced understanding of their capabilities.

These insights offer advanced readers a practical perspective on implementing and evaluating machine learning models for hate speech detection.

Key Conclusions:
DistilBERT proved to be the most effective model, demonstrating superior contextual understanding and accuracy compared to LSTM and Naive Bayes. The incorporation of advanced preprocessing techniques, such as language filtering and sequence padding, further enhanced model performance. This project emphasizes that combining state-of-the-art NLP models with rigorous preprocessing is essential for tackling hate speech detection challenges.

Final Thoughts:

This project bridges theory and practice in hate speech detection, offering a machine learning framework that addresses social media challenges while fostering safer, more inclusive digital spaces.

Data Science Ethics

The ethical considerations in this project focused on ensuring that the hate speech detection system was fair, unbiased, and transparent. Given the sensitive nature of hate speech detection, it was imperative to address potential ethical issues in data collection, preprocessing, and model development.

Potential Biases in Data Collection:
The dataset used for this project was sourced from open-access platforms and consisted of social media text labeled as hateful or non-hateful. A primary concern was the potential for biases in the labeling process, as subjective human judgments could lead to inconsistencies. Additionally, the dataset could reflect societal biases, such as the overrepresentation of certain demographics or linguistic styles in hateful content.

Mitigation Strategies for Bias:
To address these concerns, several steps were taken:

- The dataset was reviewed for balance between hateful ("1") and non-hateful ("0") labels to avoid skewed model training.
- Preprocessing steps, such as removing duplicates and ensuring consistent language filtering using the ‘langdetect‘ library, were employed to eliminate noise and irrelevant data.
- Special care was taken to retain the diversity of text inputs, including slang, emoticons, and modern linguistic trends, to ensure the model could generalize effectively.

Fairness in Model Development:
The selection of machine learning models, DistilBERT, LSTM and Naive Bayes, was guided by their ability to process diverse textual data and handle imbalanced datasets effectively. The training and testing processes were carefully designed to avoid overfitting or underfitting, ensuring unbiased predictions across all data subsets. Regular monitoring of model performance metrics, such as accuracy, loss, and confusion matrices, provided transparency and helped identify any potential bias in predictions.

Transparency in Analysis:
To ensure transparency, all preprocessing steps, model parameters, and evaluation criteria were documented comprehensively in the report and GitHub repository. Visualizations, such as word clouds and AUROC curves, were included to provide clear and interpretable results. This approach ensures that the methodology can be reproduced and scrutinized by others.

Conclusion:
By addressing biases in data and ensuring fairness and transparency in model development, this project demonstrates a commitment to ethical principles in data science. The steps taken to mitigate ethical concerns not only enhance the credibility of the hate speech detection system but also contribute to creating more equitable and trustworthy AI solutions for addressing societal challenges.