Huzaifa Ahmed -Moeez Irfan -Hassan Hameed-Hashir Fida

Project 1:

Chapter: "Exploring Toxicity - A Tale of Words and Weights

Introduction

In the expansive world of cyberspace, where words wield considerable influence over our digital interactions, we embark on a journey to explore the intricate domain of cybersecurity. The project focuses on classifying comments from online platforms (twitter) into various categories of toxicity, such as toxic, severely toxic, obscene, threatening, insulting, and identity hate. This classification aims to identify and potentially filter harmful content in online discussions.

Methodology (Naive Bayes): Finding Balance and Text Transformation:

Data Acquisition and Preparation

The dataset comprises comments with associated toxicity labels. The dataset is divided into two parts: training and test datasets.

- Training Dataset: Contains 159571 rows 8 columns.

- Test Dataset: Contains 153164 rows and 2 columns before merging with test labels and removing rows with label '-1'.

- Test data is prepared again using merging on IDS and removing -1 rows.

The data includes columns for each category of toxicity. The merging of test data with its labels ensures the alignment of comments with their respective toxicity assessments.

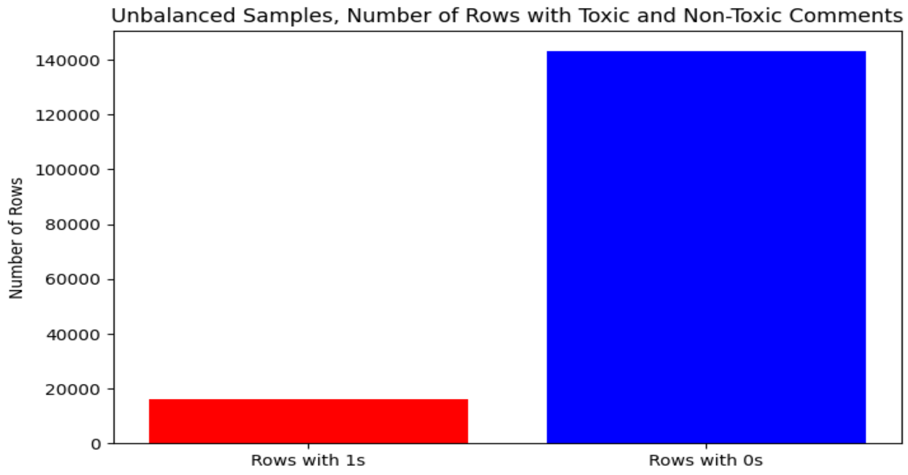
Data Preprocessing

Preprocessing steps include:

1. Text Cleaning: Lowercasing, punctuation removal, stopwords removal, lemmatization, URL removal, HTML tag removal, and handling emojis and slangs.

2. Vectorization*: Using TfidfVectorizer for both word-level and character-level features, capped at 20,000 and 40,000 features respectively.

3. Handling Imbalance in Training Data:



Addressing the class imbalance between toxic and clean data, we executed a strategic under sampling of non-toxic comments, leveling the playing field for our model.

This allowed for a more robust identification of elusive toxic instances. However, confronted with the scarcity of 1 occurrence in threat, identity_hate, and severe_toxic classes, we faced a dilemma. Balancing these classes would significantly shrink our dataset, impacting model performance. Pragmatically, we opted to maintain equilibrium solely between toxic and clean data, ensuring a judicious compromise between model effectiveness and dataset size. This process was essentially undersampling.

Feature Extraction: The TF-IDF Symphony

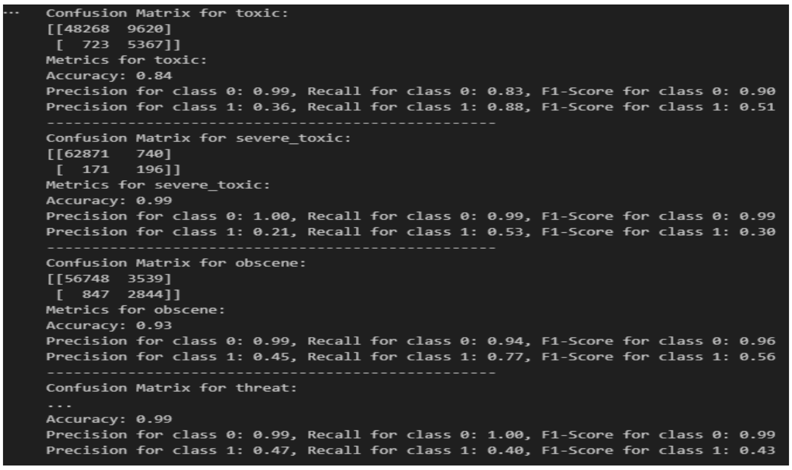
Our feature extraction methodology centers on TF-IDF (Term Frequency-Inverse Document Frequency), chosen for its efficiency and interpretability. This method quantifies the importance of words in comments, considering both frequency within a comment (TF) and rarity across the entire dataset (IDF). Its scalability and transparency make TF-IDF an efficient conductor in capturing the nuances of language, providing our model with a clear lens to distinguish toxic from benign expressions.

Motivation behind Naive Bayes:

The choice of the Naive Bayes classifier, specifically the Multinomial Naive Bayes model, is driven by its suitability for text classification, computational efficiency, and alignment with the independence assumption in text data. Its simplicity makes it an ideal baseline model for multi-label classification, providing interpretability and serving as a benchmark for more complex models.

Findings and Evaluation:

In our analysis, toxic comments are illuminated through metrics, showcasing nuanced performance. Notably, the "toxic" label achieves 84% accuracy with a balanced interplay of precision, recall, and an F1-score of 0.51. The rarer "severe_toxic" label dances with a challenging F1-score of 0.30. "Obscene" achieves a delicate balance at 93% accuracy and an impressive F1-score of 0.56. Yet, in the "threat" domain, rarity challenges the model, resulting in an F1-score of 0.05. "Insult" harmonizes with 93% accuracy and an F1-score of 0.52. "Identity_hate" presents a nuanced narrative, grappling with precision-recall intricacies for an F1-score of 0.43, emphasizing challenges in rare instances. Our exploration underscores the impact of label distribution imbalance, guiding future strategies for nuanced toxicity assessment(like RNN).



```
···   Confusion Matrix for toxic:
      [[48268  9620]
       [  723  5367]]
      Metrics for toxic:
      Accuracy: 0.84
      Precision for class 0: 0.99, Recall for class 0: 0.83, F1-Score for class 0: 0.90
      Precision for class 1: 0.36, Recall for class 1: 0.88, F1-Score for class 1: 0.51
      --------------------------------------------------
      Confusion Matrix for severe_toxic:
      [[62871   740]
       [  171   196]]
      Metrics for severe_toxic:
      Accuracy: 0.99
      Precision for class 0: 1.00, Recall for class 0: 0.99, F1-Score for class 0: 0.99
      Precision for class 1: 0.21, Recall for class 1: 0.53, F1-Score for class 1: 0.30
      --------------------------------------------------
      Confusion Matrix for obscene:
      [[56748  3539]
       [  847  2844]]
      Metrics for obscene:
      Accuracy: 0.93
      Precision for class 0: 0.99, Recall for class 0: 0.94, F1-Score for class 0: 0.96
      Precision for class 1: 0.45, Recall for class 1: 0.77, F1-Score for class 1: 0.56
      --------------------------------------------------
      Confusion Matrix for threat:
      ...
      Accuracy: 0.99
      Precision for class 0: 0.99, Recall for class 0: 1.00, F1-Score for class 0: 0.99
      Precision for class 1: 0.47, Recall for class 1: 0.40, F1-Score for class 1: 0.43
```

Sequence Model RNNs:

Motivation for using RNN with LSTM:

Choosing an RNN with LSTM layers for toxic comment classification is driven by its effectiveness in capturing the sequential nature of text, crucial for discerning long-range dependencies and semantic representations. The architecture, particularly with LSTM layers, proves well-suited for the complexities of toxic comment classification, enhancing accuracy by leveraging sequential information and nuanced semantic understanding.

Tokenization and Padding

We tokenized and padded the comments using Keras's Tokenizer with a vocabulary size of 20,000. The maximum length of sequences was set to 200, and sequences were padded or truncated as necessary. We split the balanced training data into training and validation sets (80-20 split) for model training and evaluation.

Model Architecture

The RNN model consists of the following layers:

- An Embedding layer to represent words in a 128-dimensional vector space.

- Two LSTM layers with 64 and 32 units, respectively. The first LSTM layer returns sequences, while the second one does not.

- A Dense layer with 64 units and ReLU activation function.

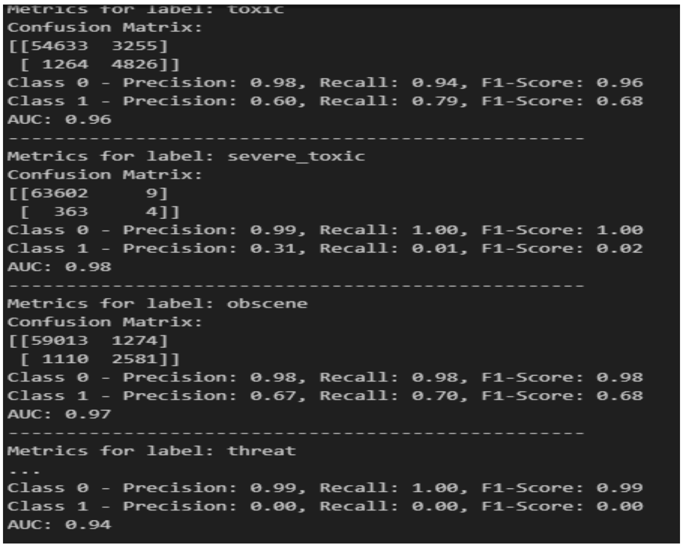
- A Dropout layer with a rate of 0.5 to prevent overfitting.

- An output Dense layer with 6 units (corresponding to the six labels) and sigmoid activation.

Compilation and Training

We compiled the model using binary cross-entropy loss and the Adam optimizer, with accuracy as the metric. The model was trained over three epochs on the training data and validated on the validation set.

Prediction and Evaluation



```
Metrics for label: toxic
Confusion Matrix:
[[54633  3255]
 [ 1264  4826]]
Class 0 - Precision: 0.98, Recall: 0.94, F1-Score: 0.96
Class 1 - Precision: 0.60, Recall: 0.79, F1-Score: 0.68
AUC: 0.96
--------------------------------------------------
Metrics for label: severe_toxic
Confusion Matrix:
[[63602     9]
 [  363     4]]
Class 0 - Precision: 0.99, Recall: 1.00, F1-Score: 1.00
Class 1 - Precision: 0.31, Recall: 0.01, F1-Score: 0.02
AUC: 0.98
--------------------------------------------------
Metrics for label: obscene
Confusion Matrix:
[[59013  1274]
 [ 1110  2581]]
Class 0 - Precision: 0.98, Recall: 0.98, F1-Score: 0.98
Class 1 - Precision: 0.67, Recall: 0.70, F1-Score: 0.68
AUC: 0.97
--------------------------------------------------
Metrics for label: threat
...
Class 0 - Precision: 0.99, Recall: 1.00, F1-Score: 0.99
Class 1 - Precision: 0.00, Recall: 0.00, F1-Score: 0.00
AUC: 0.94
```

The model excels in identifying non-toxic and non-threatening comments, displaying high precision (0.98-0.99) and recall (0.94-1.00) across these categories. However, challenges arise in classifying severe toxic and threatening comments, resulting in lower precision (0.00-0.31) and recall (0.00-0.79). The model maintains a good overall balance, achieving an AUC of 0.96-0.98. In summary, while effectively distinguishing benign comments, it faces difficulties with nuanced toxic and threatening content, emphasizing the trade-off between precision and recall. Achieving a robust balance is essential for versatile model application.

Pretrained Encoder-Transformers BERT:

Motivation:

Pretrained encoder-transformers like BERT offer effectiveness in toxicity classification through capturing contextual information, enabling transfer learning for accelerated task-specific adaptation. BERT's contextual embeddings enhance semantic understanding, crucial for discerning nuances in text. Their efficiency lies in reducing the need for extensive training on task-specific data, resulting in significant computational resource savings.

Data Preparation:

Data Sources: The data was sourced from three CSV files: train.csv, test.csv, and test_labels.csv.

Data Merging and Filtering: The test data was merged with the test labels and filtered to exclude rows with '-1' labels, indicating missing or invalid data.

Balancing the Dataset: The training dataset was balanced by ensuring equal representation of toxic and non-toxic comments. This was crucial to prevent model bias towards the more prevalent class. Essentially same as for Naive bayes (Using under sampling)

Dataset Splitting: The balanced data was split into training, testing, and validation sets, with 25% reserved for testing and validation.

Data Preprocessing was left for Bert Pre-Encoded Model, for example, by using lowercase=True

Model Training and Evaluation

BERT Tokenization: Text data was tokenized using the BERT tokenizer to convert comments into a format suitable for the BERT model.

BERT Model Initialization: The BERT model was initialized with 6 output labels corresponding to different toxicity types.

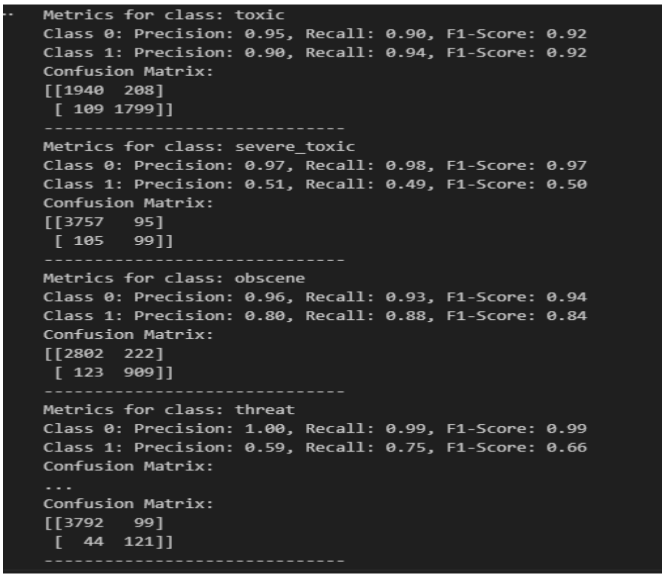
DataLoader Creation: DataLoaders for training, testing, and validation datasets were prepared to facilitate efficient batch processing.

Optimizer Setup: The AdamW optimizer with a learning rate of 2e-5 was used.

Training Process: The model was trained over three epochs, with the training loss and validation loss monitored to gauge model performance.

Findings

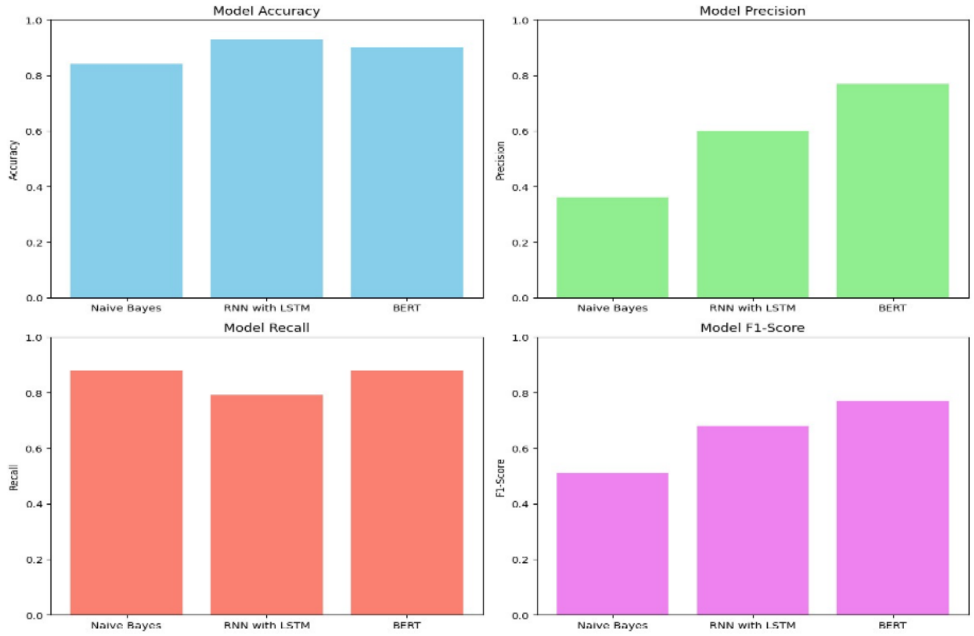
The model excels in toxicity classification with 92% accuracy for toxic comments and an impressive F1-Score of 0.92. While achieving 97% accuracy in severe toxic comments, it faces a trade-off with a moderate F1-Score of 0.50 due to lower precision (0.51) and recall (0.49). For obscene comments, the model performs well with 94% accuracy and an F1-Score of 0.84. In identifying threatening comments, it achieves 99% accuracy, displaying perfect precision (1.00) for non-threatening comments and a balanced F1-Score of 0.66 for threatening ones. Overall, the model demonstrates effective performance, emphasizing the need for a balanced precision-recall trade-off in diverse scenarios.



```
    Metrics for class: toxic
    Class 0: Precision: 0.95, Recall: 0.90, F1-Score: 0.92
    Class 1: Precision: 0.90, Recall: 0.94, F1-Score: 0.92
    Confusion Matrix:
    [[1940  208]
     [ 109 1799]]
    ------------------------------
    Metrics for class: severe_toxic
    Class 0: Precision: 0.97, Recall: 0.98, F1-Score: 0.97
    Class 1: Precision: 0.51, Recall: 0.49, F1-Score: 0.50
    Confusion Matrix:
    [[3757   95]
     [ 105   99]]
    ------------------------------
    Metrics for class: obscene
    Class 0: Precision: 0.96, Recall: 0.93, F1-Score: 0.94
    Class 1: Precision: 0.80, Recall: 0.88, F1-Score: 0.84
    Confusion Matrix:
    [[2802  222]
     [ 123  909]]
    ------------------------------
    Metrics for class: threat
    Class 0: Precision: 1.00, Recall: 0.99, F1-Score: 0.99
    Class 1: Precision: 0.59, Recall: 0.75, F1-Score: 0.66
    Confusion Matrix:
    ...
    Confusion Matrix:
    [[3792   99]
     [  44  121]]
    ------------------------------
```

COMPARISON OF ALL THREE:

The comparison of Naive Bayes, RNN with LSTM, and BERT reveals distinctive performance characteristics. Naive Bayes excels in accuracy and recall for non-toxic comments but struggles with precision for toxic ones. RNN with LSTM shows balanced performance, leveraging contextual understanding to effectively distinguish toxic and non-toxic comments. BERT outperforms in precision and F1-score, showcasing advanced contextual understanding crucial for accurate toxicity identification. In general, all models exhibit high accuracy, with BERT slightly leading. BERT is identified as the best for general use, offering superior balance in precision, recall, and F1-score. Naive Bayes is prioritized for high recall, while



BERT is recommended for scenarios demanding high precision. In conclusion, BERT emerges as the most effective model for toxic comment classification, combining advanced contextual understanding with balanced metrics.