DOCUMENTATION

Title: HATE SPEECH DETECTION

Author: SANJAY VENKATESHWARAN

Mentor: Dr N JAGAN MOHAN

Introduction:

Hate speech detection is a vital application of natural language processing (NLP) and Machine learning to identify and mitigate harmful content across digital platforms. As social media and online communities have become central to modern communication, the prevalence of hate speech expressions that incite violence, discrimination, or hostility towards individuals or groups based on attributes like race, gender, religion, or sexual orientation has become a significant issue. The goal of hate speech detection is to automatically flag or filter out such toxic language, fostering safer and more inclusive online spaces.

Objectives:

- Detect Hate Speech: Classify text data into two categories: Hate Speech and Normal Speech.
- Enhance Accuracy: Implement preprocessing techniques and machine learning models to achieve high classification accuracy.
- Create a Scalable Pipeline: Build an automated and reusable pipeline for text preprocessing and model training.

Database Preparation:

- Kaggle’s Toxic Comment Classification Dataset Contains labeled comments from Wikipedia’s talk pages, with categories like "toxic", "severe toxic", "neither", "threat", "count", “normal” and "identity hate".

- This dataset is versatile, with different types of toxicity beyond just hate speech, making it ideal for multi-label classification models.

Further Classification:

Re - labeling and Verification:

- Review your dataset labels to ensure consistency, especially if the data was collected from different sources.
- Consider adding a manual or semi-automated check to verify that each instance aligns well with "offensive," "normal," or "neither," as this can impact the accuracy of your model.

Dataset Description:

Source: Kaggle's Toxic Comment Classification Dataset.

Training Data:

- **Size:** 31,962 entries.
- **Columns:** id, label, tweet.

Testing Data:

- **Size:** 17,197 entries.
- **Columns:** id, tweet.

Preprocessing of Datasets:

Lowercasing:

- Remove unnecessary characters, such as punctuation, special symbols, and extra whitespaces.
- Normalize text by converting all characters to lowercase to ensure case consistency.

Tokenization:

- Split sentences into individual words or tokens, which allows models to analyze the text at the word level.

- For deep learning models, you might want to use tokenizers specific to libraries, like the BERT tokenizer for transformers.

Stop Words Removal and Punctuation Removal:

- Remove common but uninformative words like "is," "the," and "and" to reduce noise in the data.
- Be mindful of context, as stop words may carry meaning in specific types of hate speech (e.g., "we" and "they" could be relevant in detecting group-targeted hate speech).

Stemming and Lemmatization:

- Convert words to their base or root forms to reduce vocabulary size and improve generalization. For example, "running" and "runs" would be reduced to "run."
- Stemming is quicker but can be less precise, while lemmatization is more contextually accurate.

Removing Extra White Spaces:

- **Trim Whitespace**: Remove leading and trailing white spaces in each text sample. This prevents redundant spaces from interfering with tokenization and feature extraction.
- **Remove Extra Spaces**: Replace multiple spaces between words with a single space to standardize the text format.

Removing Numbers:

- Remove Irrelevant Numbers: Many numbers (like dates, IDs, and random digits) may not carry meaningful information for hate speech detection. Removing or replacing them with a generic token (NUM) can reduce noise.

Text normalization:

- It is a preprocessing step in natural language processing (NLP) that involves transforming text into a consistent, standard format to ensure that variations in language do not hinder analysis.

- It prepares textual data for easier processing by algorithms and typically includes several techniques.

Handling Imbalanced Data:

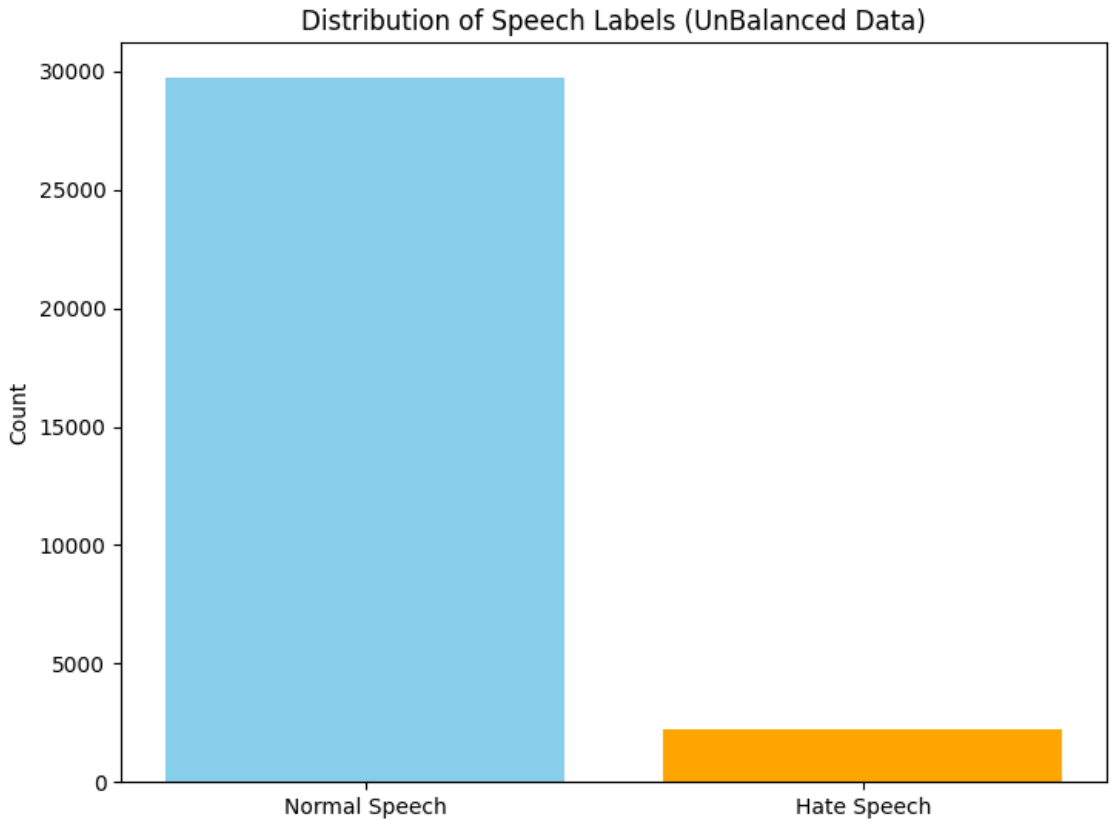
- Imbalanced data is common in hate speech detection, where "offensive" or "neither" classes might be underrepresented compared to "normal" content. This can be addressed by:
- **Oversampling**: To balance the dataset, duplicate samples from the minority classes (“offensive” or “neither”) are used.

Distribution (Unbalanced Data)

- Normal Speech: 29,720
- Hate Speech: 2,242

Visualization:

- Bar chart highlighting the skewed class distribution.



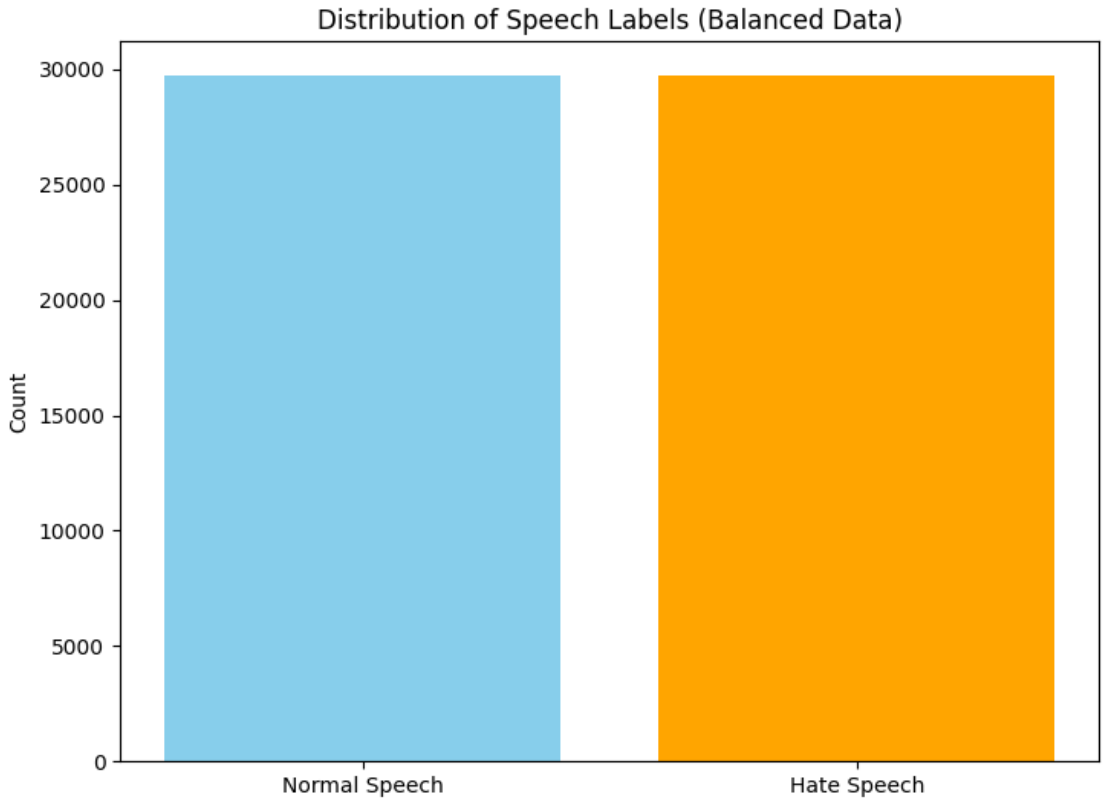
Resulting Dataset:

Balanced Distribution:

- Normal Speech: 29,720 entries.
- Hate Speech: 29,720 entries.

Visualization:

- Bar chart showing the equal distribution of Normal and Hate Speech.



Machine Learning (ml):

- Refers to algorithms that allow computers to learn patterns from data and make decisions without being explicitly programmed for every possible scenario. Traditional ML methods rely on structured data and often require feature engineering.

Deep Learning (dl):

- A subset of ML that uses neural networks with multiple layers (deep neural networks) to automatically learn hierarchical patterns from large volumes of data. DL models are highly effective at capturing complex patterns but typically require more data and computational power.

Feature Extraction:

TF-IDF Vectorization:

- Converts text into numerical features based on term frequency-inverse document frequency (TF-IDF).
- **Vocabulary Size:** Limited to the 5,000 most frequent terms.
- **Tool Used:** TfidfVectorizer from sklearn.

Model Building:

Pipeline Design:

- **Vectorization**: CountVectorizer + TfidfTransformer.
- **Classifier**: Stochastic Gradient Descent (SGD).
- **Workflow**: Automated preprocessing and training using sklearn's Pipeline.

Train-Test Split:

- **Training Data:** 80% of the dataset.
- **Testing Data:** 20% of the dataset.

Model Evaluation:

Metrics:

- **Accuracy**: 97.03%.
- **F1 Score**: 0.97.

Classification Report:

Class	Precision	Recall	F1-Score
Normal Speech	0.98	0.96	0.97
Hate Speech	0.96	0.98	0.97

Training vs Testing Accuracy:

- **Training Accuracy**: 97.12%.
- **Testing Accuracy**: 97.03%.

These metrics highlight that the model generalizes well without significant overfitting. User Interaction

Real-Time Classification:

Users can input text to be classified as either Normal Speech or Hate Speech.

Example Predictions

Input: "You're so annoying!" → **Output:** Hate Speech.

Input: "Have a great day!" → **Output:** Normal Speech.

Challenges and Limitations:

Challenges:

Data Imbalance:

- Addressed using upsampling techniques to balance the dataset.

Noisy Data:

- Handled through robust preprocessing steps, including removal of mentions, URLs, and special characters.

Limited Diversity:

- The dataset is primarily English-based, limiting the model's applicability to multilingual scenarios.

Limitations:

- **Model Bias**: Despite balancing, subtle biases may persist due to the dataset's inherent nature.
- **Context Understanding**: Models like SGD rely on surface-level text features and may miss nuanced meanings.

Future Work:

Multilingual Support:

- Extend the model to detect hate speech in multiple languages.

Advanced Models:

- Incorporate transformer-based architectures like BERT for better context understanding.

Real-Time Integration:

- Develop APIs to deploy the model in real-time moderation systems for social media platforms.

Enhanced Datasets:

- Collect and preprocess larger, diverse datasets for better generalization.

Conclusion:

The Hate Speech Detection project successfully applies machine learning and natural language processing techniques to address a critical societal issue. By implementing robust data preprocessing, handling class imbalance, and leveraging effective modeling strategies, the project achieved high accuracy and generalization capabilities.

References:

- S. Fortuna and S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," ACM Computing Surveys, vol. 55, no. 2, pp. 1-37, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3485210
- Schmidt and M. Wiegand, "A Survey on Hate Speech Detection Using Natural Language Processing," in Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media (SocialNLP), Valencia, Spain, Apr. 2017, pp. 1–10. [Online]. Available: https://aclanthology.org/W17-1101
- Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," IEEE Transactions on Big Data, vol. 7, no. 1, pp. 99-112, Mar. 2021, doi: 10.1109/TBDATA.2020.3030897.
- R. Mishra, P. Yannakoudakis, and E. Shutova, "Tackling Online Abuse: A Survey of Automated Abuse Detection in Content Moderation," IEEE Access, vol. 9, pp. 90913–90933, 2021, doi: 10.1109/ACCESS.2021.3086846.
- Waseem, Z. Hovy, and D. Davidson, "Understanding Abuse: A Typology of Abusive Language Detection Subtasks," in Proceedings of the First Workshop on Abusive Language Online, Vancouver, Canada, Aug. 2017, pp. 78-84. [Online]. Available: https://aclanthology.org/W17-3012