Hate Speech Detection Project

Individual Project
Name: Riya Gaur
Email: riyagaur1299@gmail.com
Country: USA
Specialization: NLP (Natural Language Processing)

Problem Description

Hate speech is any form of communication, whether verbal, written, or behavioral, that attacks or discriminates against individuals or groups based on attributes like religion, ethnicity, nationality, race, color, sex, or other identity factors. The aim of this project is to develop an NLP-based hate speech detection model that can classify tweets as either containing hate speech (1) or not (0). This model will assist in content moderation, promoting safer and more inclusive digital spaces.

Data Understanding

Type of Data

- **Source:** Twitter dataset containing labeled tweets
- **Attributes:**
  - `id` : Unique identifier for each tweet.
  - `label` : Binary classification (0 for non-hate speech, 1 for hate speech).
  - `tweet` : The actual text content of the tweet.
- **Dataset Size:**
  - Training Set: 31,962 tweets
  - Test Set: 17,197 tweets

Problems in the Data

- **Class Imbalance:**
  - **Non-hate speech (label=0):** 92.98%
  - **Hate speech (label=1):** 7.02%
  - Issue: The dataset is heavily skewed towards non-hate speech, potentially affecting model performance.
- **Missing Values:**

    - No missing values detected in the dataset.
- **Noisy Text:**
    - Tweets contain user mentions (`@user`), URLs, hashtags, emojis, and special characters.
- **Outliers:**
    - Certain tweets contain excessive punctuation or repetitive characters.
- **Short Text Lengths:**
    - Tweets are limited in characters, affecting NLP feature extraction.

Approaches to Overcome Data Issues

Handling Class Imbalance

- **Oversampling:** Using **Synthetic Minority Over-sampling Technique (SMOTE)** to generate synthetic hate speech samples.
- **Undersampling:** Randomly reducing non-hate speech samples to balance the dataset.
- **Cost-sensitive Learning:** Assigning higher class weights to hate speech samples during training.

Data Preprocessing

- **Text Cleaning:**
    - Remove special characters, hashtags, URLs, and user mentions (`@user`).
    - Convert text to lowercase.
    - Tokenization and stopword removal.
    - Apply lemmatization and stemming.
- **Feature Engineering:**
    - **TF-IDF Vectorization** for traditional ML models.
    - **Word Embeddings** (Word2Vec, GloVe, or BERT) for deep learning models.
    - **N-grams** for capturing contextual information.

Modeling Approaches

- **Baseline Models:**
    - Logistic Regression, Support Vector Machines (SVM), and Random Forest using TF-IDF features.
- **Advanced Models:**
    - **Transformer-based Models (BERT, RoBERTa, DistilBERT)** for deep contextual learning.
    - Fine-tuning pre-trained transformers on the dataset

Project Lifecycle and Deadline

1. **Problem Understanding**
2. **Data Collection and Cleaning**
3. **Data Representation and Exploration**
4. **Model Building and Training**
5. **Performance Evaluation**
6. **Deployment and Inference**

Deadline: Feb 28