Hate Speech Detection using Transformers (Deep Learning)

Team Member Details

Group Name: NLP Pioneers

Name: Duc Duong

Email: duongduc@grinnell.edu

Country: United States

College: Grinnell College

Specialization: Data Science & NLP

Problem Description

Hate speech refers to written, verbal, or behavioral communication that attacks or discriminates against individuals or groups based on their race, ethnicity, religion, nationality, gender, or other identity factors. This project aims to build a hate speech detection model using transformer-based deep learning approaches to classify tweets as hate speech or non-hate speech.

Data Understanding

The dataset is sourced from Kaggle and contains approximately 49,200 tweets across two CSV files (train and test). Each row corresponds to one tweet, identified by an ID, with associated features including the text content of the tweet and its label. The dataset includes five main columns: ID, tweet text, and label information.

Type of Data

The dataset is text-based, consisting of short social media posts (tweets). It is a structured dataset stored in CSV format, where each record represents a single tweet. Features are both categorical (labels) and textual (tweet content).

Problems in the Data

- Duplicate entries: There are ~49,200 rows but only ~16,130 unique tweets.
- Missing values: Potential NA values may exist in tweet texts or labels.
- Noise: Tweets may include URLs, hashtags, emojis, and mentions that do not contribute to

classification.
- Class imbalance: Hate speech occurrences are fewer compared to non-hate speech tweets.
- Outliers: Extremely long or irrelevant tweets may exist.
- Skewness: Label distribution may be skewed heavily towards non-hate speech.

Approaches to Handle Data Issues

- **Deduplication**: Remove duplicate tweets based on ID and text to ensure unique samples.
- **Handling Missing Values**: Drop records with NA values in essential fields (tweet text or label).
- **Text Cleaning**: Normalize text by removing special characters, URLs, hashtags, and user mentions.
- **Outlier Handling**: Filter tweets with excessively short or long length to avoid noise.
- **Class Imbalance**: Apply SMOTE, oversampling, or use class-weighted loss functions to balance labels.
- **Tokenization**: Apply transformer-based tokenizers (BERT, RoBERTa) for consistent feature extraction.