PROJECT : Hate-Speech-detection-using-Transformers-Deep-Learning

Group Name: Hate Speech Detective

Members:

Name: VARSHIT MANEPALLI

Email: varshitmanepalli1810@gmail.com

Country: INDIA

College/Company: Stevens Institute of Technology

Specialization: NLP

Batch code : LISUM32

Problem Description

Hate speech detection aims to identify and classify statements that contain offensive, derogatory, or discriminatory language directed towards individuals or groups based on their identity factors such as religion, ethnicity, nationality, race, color, ancestry, sex, or other identity factors. This project involves developing a machine learning model to detect hate speech in Twitter tweets.

Business Understanding

Hate speech can have serious consequences, including perpetuating discrimination, inciting violence, and causing psychological harm. Detecting hate speech on social media platforms like Twitter is crucial for maintaining a safe and inclusive online environment. By identifying and flagging hate speech, we can help prevent the spread of harmful content and protect vulnerable individuals and communities.

Dataset Structure

Training Data

- Contains 31,962 entries and 3 columns: id, label, and tweet.
    - id: Identifier for each tweet (integer).
    - label: Binary indicator (0 or 1) indicating the presence of hate speech (integer).
    - tweet: The tweet text (string).

Test Data

- Contains 17,197 entries and 2 columns: id and tweet.
    - id: Identifier for each tweet (integer).
    - tweet: The tweet text (string).

Problems in the Data

- **Missing Values (NA values)**
  No missing values were detected in either dataset.
- **Outliers**

  Outliers are generally not applicable to text data but can be checked in the context of tweet lengths or the distribution of labels.

- **Skewed Data**
  The distribution of the label column in the training data can be checked to understand class imbalance.

Data Cleaning

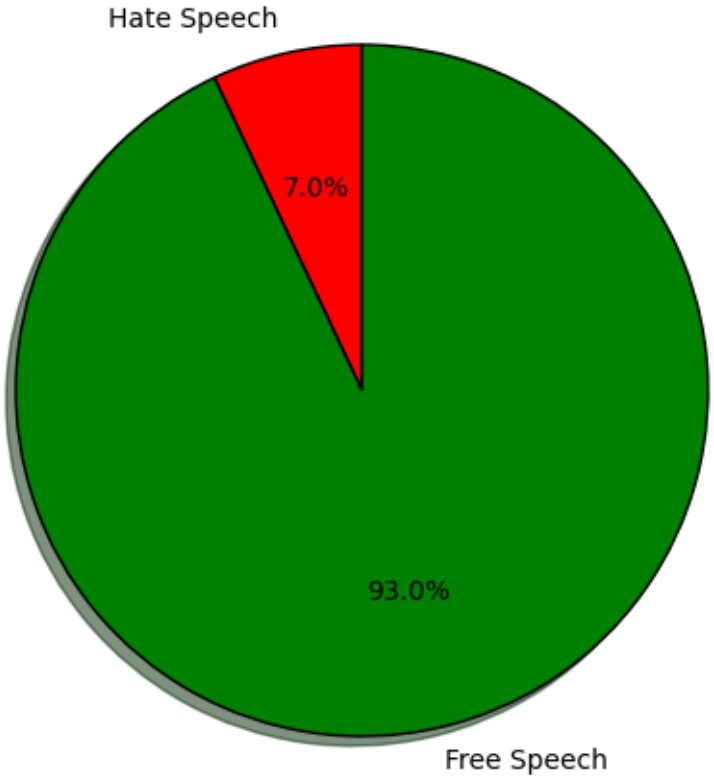
- No missing values were found, so no imputation was necessary.
- Outliers in the context of tweet lengths can be handled through filtering or transformation if required.
- Class imbalance in the label column can be addressed using resampling techniques or appropriate evaluation metrics.

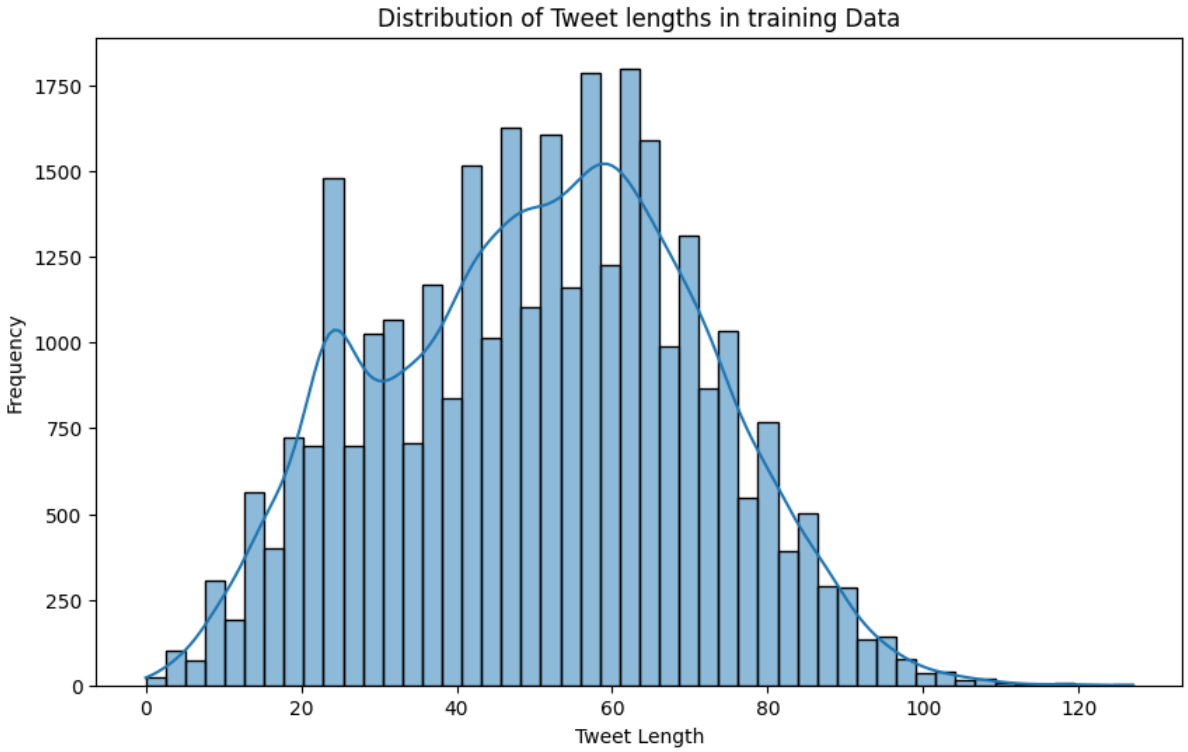
Data Preprocessing

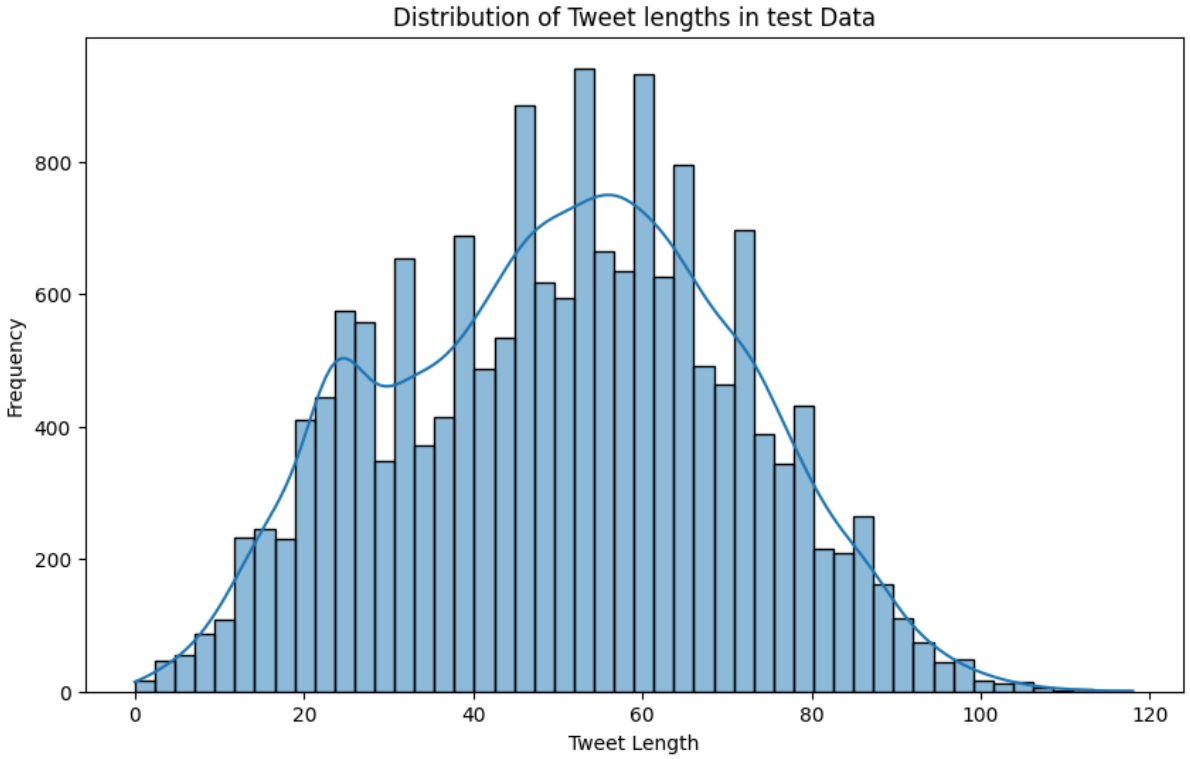
- **Tokenization**
  - Converting text into individual words or tokens.
  - Example: Splitting sentences into words.
- **Removing Stop Words**
  - Removing common words that do not contribute much to the meaning (e.g., "and", "the", "is").
  - This helps in reducing the dimensionality of the text data.
- **Stemming/Lemmatization**
  - Reducing words to their root form.
  - Example: Converting "running" to "run".
- **Handling Special Characters and Punctuation**
  - Removing or replacing special characters and punctuation marks to clean the text.
  - Example: Removing hashtags, mentions, and URLs.
- **Lowercasing**
  - Converting all text to lowercase to ensure uniformity.
  - Example: Converting "Hate" and "hate" to "hate".
- **Generating Word Clouds**
  - Visualizing the most frequent terms in hate speech and non-hate speech tweets to understand common patterns.

Data Visualization

These are the visualizations I have done based on the data

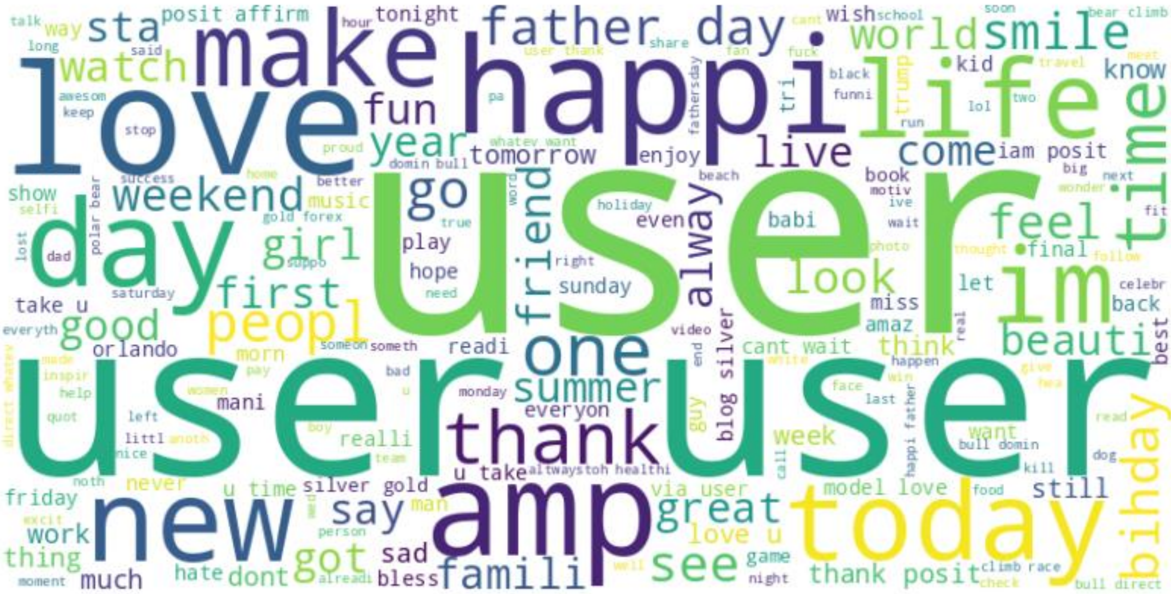



Distribution of Tweet lengths in training Data
Frequency
Tweet Length
0
250
500
750
1000
1250
1500
1750
0
20
40
60
80
100
120


Distribution of Tweet lengths in test Data
Frequency
Tweet Length
0
200
400
600
800
0
20
40
60
80
100
120

Word Cloud of Training Data

Word Cloud of Test Data



Common Words in class 0 (Non-Negative Tweets)

Common Words in class 1 (Negative Tweets)