Project title: Hate Speech Detection

Name: Vishesh Mahesh Jain

Email: visheshjain256@gmail.com

Batch: 02

Domain: Artificial Intelligence

Mentor: Dr. N Jagan Mohan

Email: springboardmentor891v@gmail.com

Table of Contents

Introduction

Hate speech detection has become increasingly important in today's digital age, where social media platforms are frequently used to express opinions. This project aims to build an efficient Hate Speech Detection System using a Logistic Regression model. The system processes tweets and classifies them into two categories: hate speech or non-hate speech. An interactive web interface was also developed using Gradio for real-time predictions

Objective

The primary objective of this project is to:

- Develop a machine learning model to classify tweets as hate speech or non-hate speech.
- Address challenges like data imbalance and noisy text data.
- Build a user-friendly interface to provide real-time predictions.

Dataset Description

- **Source:** The dataset used in this project is publicly available and specifically designed for hate speech detection. [1]
- **Structure:**
  - `tweet`: Contains the text of the tweet.
  - `label`: Binary labels, where 0 represents non-hate speech and 1 represents hate speech.
- **Size:** The dataset comprises 25,296 labeled tweets.
- **Description:**
  - The dataset contains tweets collected from Twitter, each labeled as hate speech or non-hate speech. The label distribution shows an imbalance, with the majority being non-hate speech. This highlights the need for techniques like SMOTE to balance the dataset during training.
  - Example entries:
    - Tweet: "I hate this so much!"
      - Label: 1 (Hate Speech)
    - Tweet: "Good morning everyone!"
      - Label: 0 (Non-Hate Speech)
- **Tweet**: The textual content of the social media post or tweet, serving as the primary input for the model.

Data Preprocessing

Preprocessing was critical to ensure clean and meaningful input data for the model. The following steps were undertaken:

- **Duplicate Removal:** Eliminated duplicate entries to ensure unique data points.
- **Text Cleaning:**
    - Removed URLs, hashtags, mentions, special characters, and emojis.
    - Converted all text to lowercase to standardize input.
- **Tokenization:** Split the text into individual words for further processing.
- **Stop Words Removal:** Removed common words like “and”, “is”, and “the” that do not contribute to sentiment analysis.
- **Lemmatization:** Reduced words to their base forms (e.g., “running” to “run”) to ensure consistency in the vocabulary.

Resulting Dataset

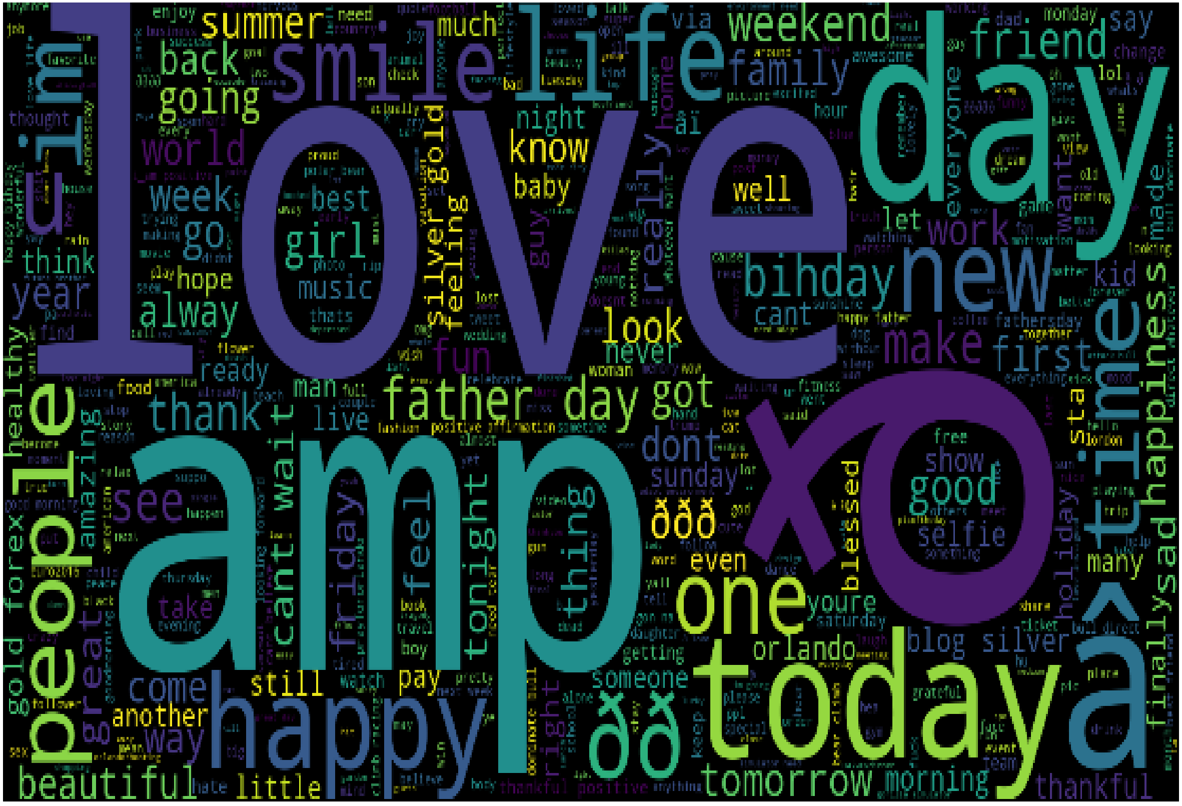
After preprocessing, the dataset was cleaned, standardized, and structured for effective analysis. By eliminating irrelevant content and ensuring uniformity, the dataset was optimized for input into machine learning algorithms, allowing for improved accuracy and consistency in hate speech detection.

Data Visualization

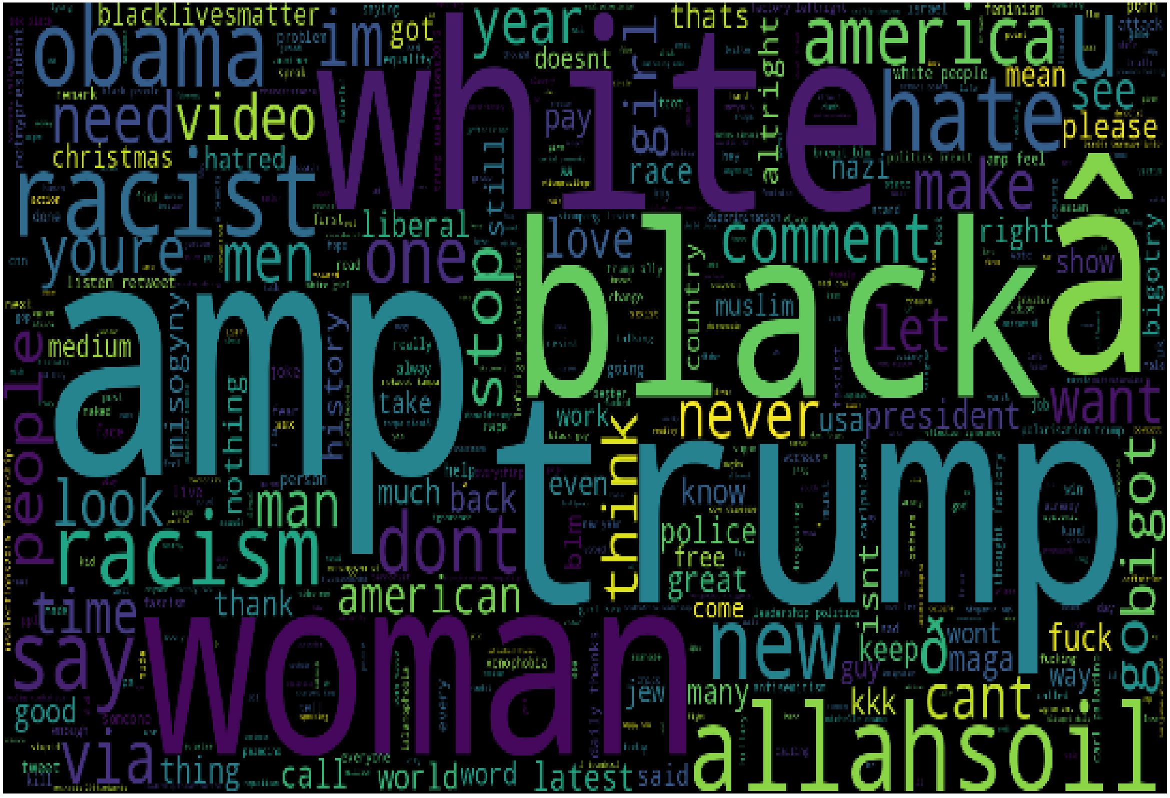
To gain insights into the dataset, several visualizations were created:

- **Label Distribution:**
    - A bar chart and pie chart showed that approximately 85% of the tweets were non-hate speech, while 15% were hate speech.
- **Word Clouds:**
    - Visualized the most frequently occurring words in hate speech and non-hate speech categories. For instance, words like "hate" and "kill" were prominent in hate speech tweets, while neutral or positive terms were common in non-hate speech. [2]

Most Frequent Words in Non-Hate Tweets



Most Frequent Words in Hate Tweets



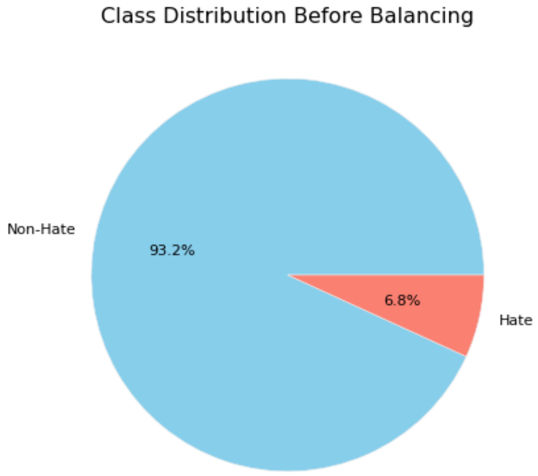
Feature Extraction

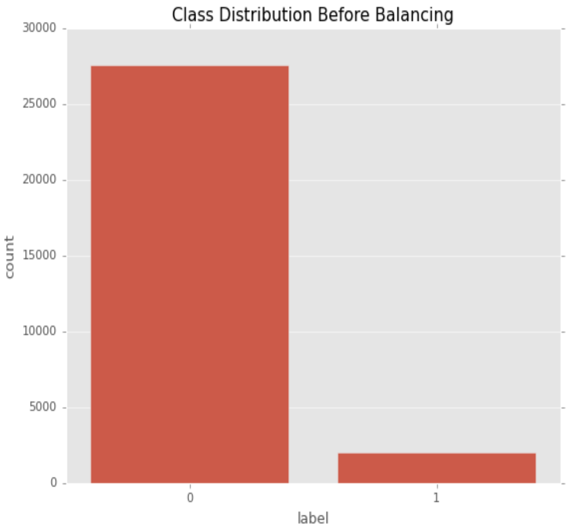
Text data was converted into numerical format for machine learning:

- **TF-IDF Vectorization:**
    - Transformed text into numerical vectors based on Term Frequency-Inverse Document Frequency. [3]
    - Supported n-grams (1-gram, 2-gram, and 3-gram combinations) to capture context and word associations effectively.

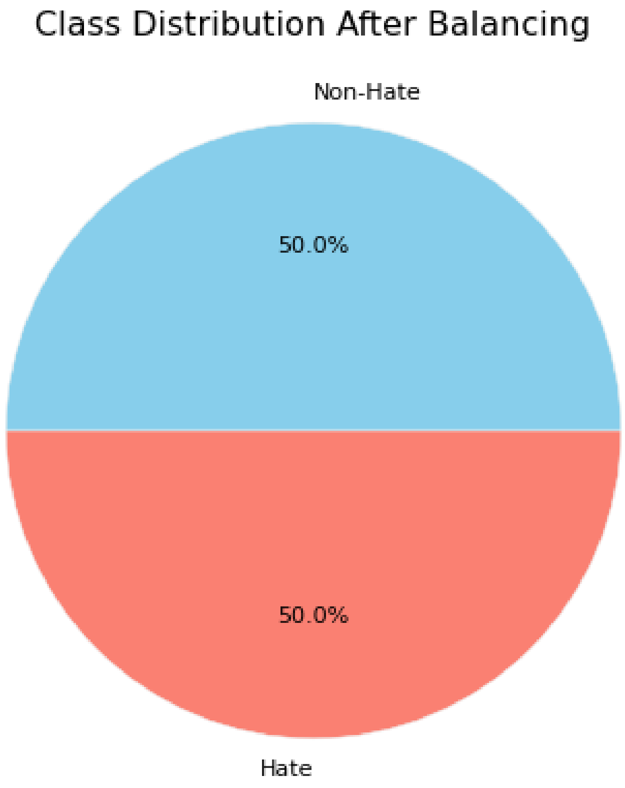
Dataset Balancing

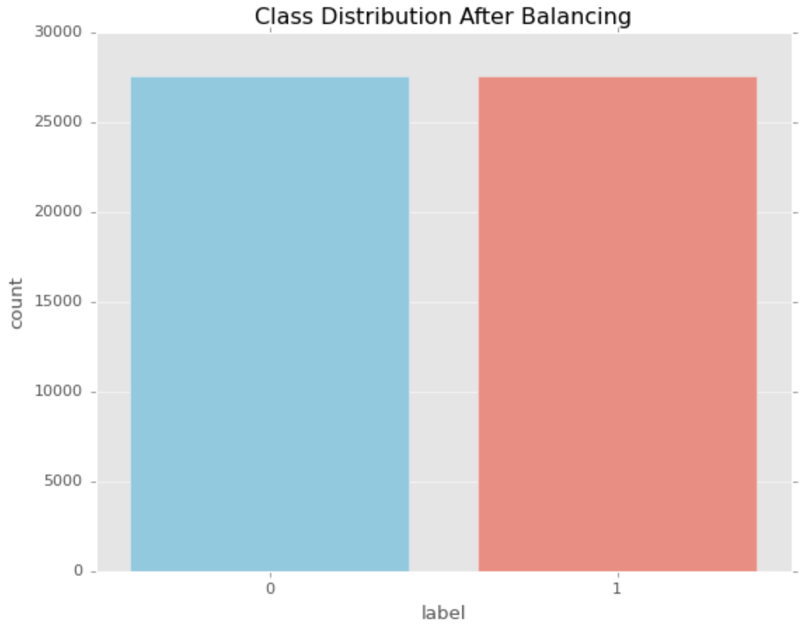
Problem: The dataset was imbalanced, with significantly more non-hate speech tweets.





Solution: SMOTE (Synthetic Minority Oversampling Technique) was used to oversample the minority class, ensuring better model performance by creating synthetic samples of hate speech tweets. [4]





Model Building

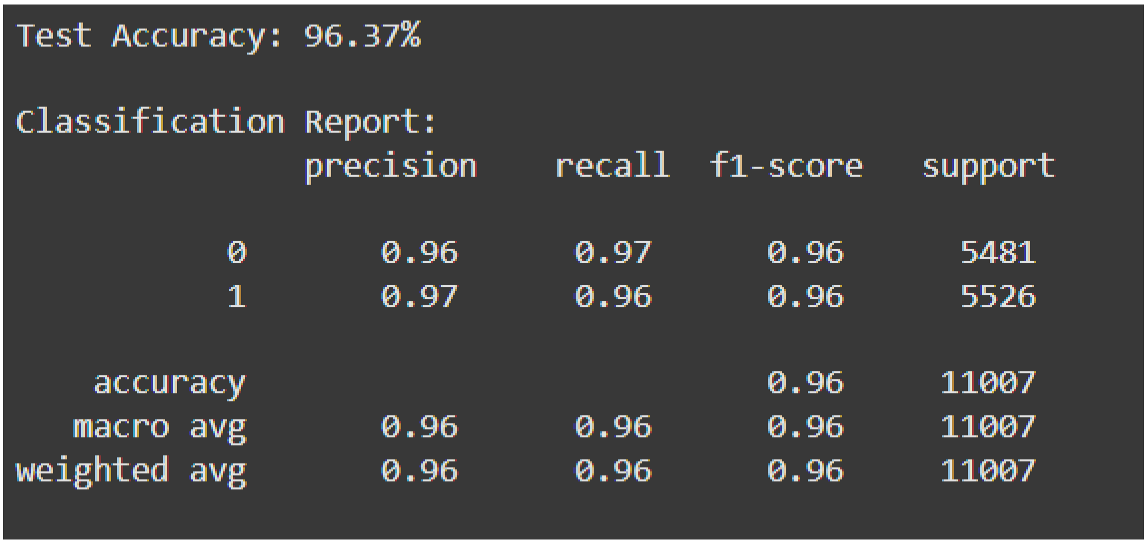
- **Algorithm Used:** Logistic Regression, chosen for its simplicity and efficiency in binary classification tasks.
- **Hyperparameter Tuning:**
  - GridSearchCV was used to find the optimal model parameters, such as regularization strength (C value) and solver selection. [5]
- **Training:**
  - The dataset was split into training and testing sets (80:20 split).
  - Cross-validation was employed to evaluate model performance and prevent overfitting.

Model Evaluation

- **Metrics:**

- **Accuracy:** The model achieved a test accuracy of **96.37%**, demonstrating high performance in classifying tweets.
- **Confusion Matrix:** Displayed true positives, true negatives, false positives, and false negatives, providing insights into model predictions.
- **Classification Report:** Detailed performance metrics for each class:

Classification Report



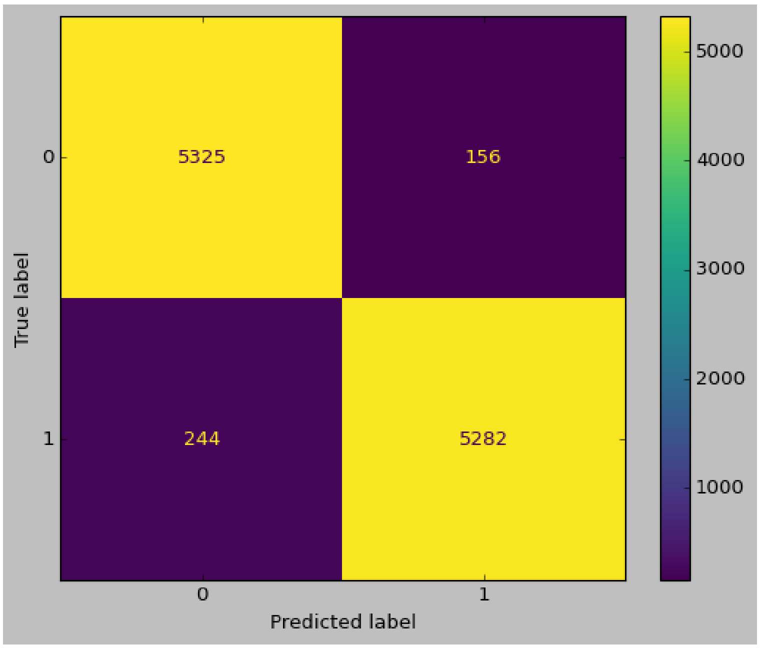
```
Test Accuracy: 96.37%

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96      5481
           1       0.97      0.96      0.96      5526

    accuracy                           0.96     11007
   macro avg       0.96      0.96      0.96     11007
weighted avg       0.96      0.96      0.96     11007
```

Confusion Matrix



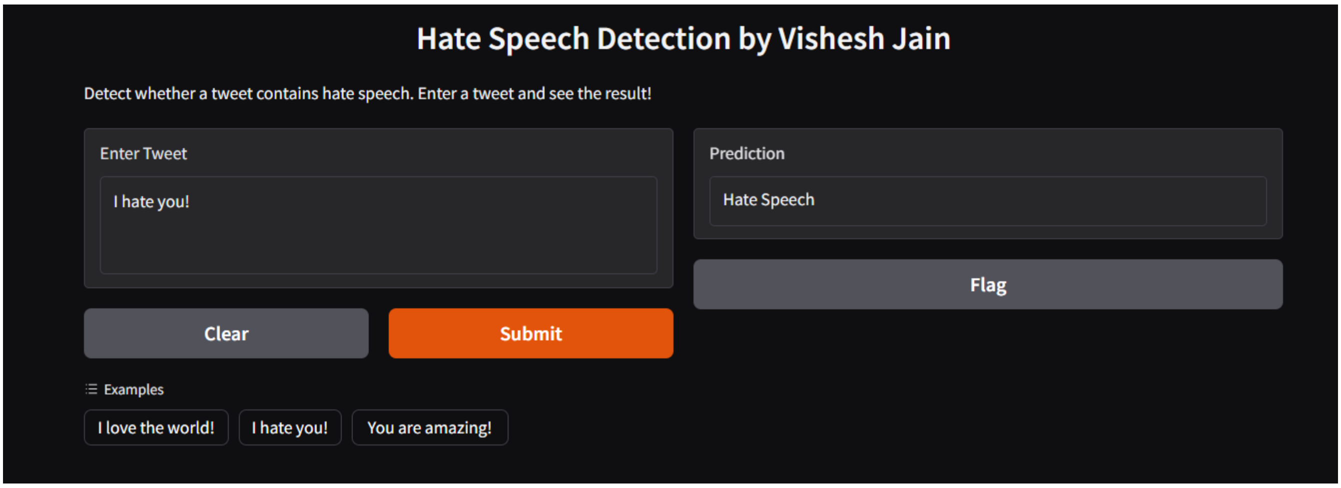
Deployment

- **Serialization:**
  - The trained model and TF-IDF vectorizer were saved as `.pkl` files using the `pickle` library, allowing for seamless deployment.
- **Gradio Interface:**
  - An interactive web interface was developed to allow users to input tweets and classify them in real-time.

Interactive Web Interface

- **Functionality:**
  - Users input a tweet and receive a prediction: “Hate Speech” or “Non-Hate Speech.”
- **Technology Used:** Gradio, for a user-friendly and interactive experience.[6]
- **Example:**
  - Input: "I hate you"

- ○ Output: "Hate Speech"



Hate Speech Detection by Vishesh Jain

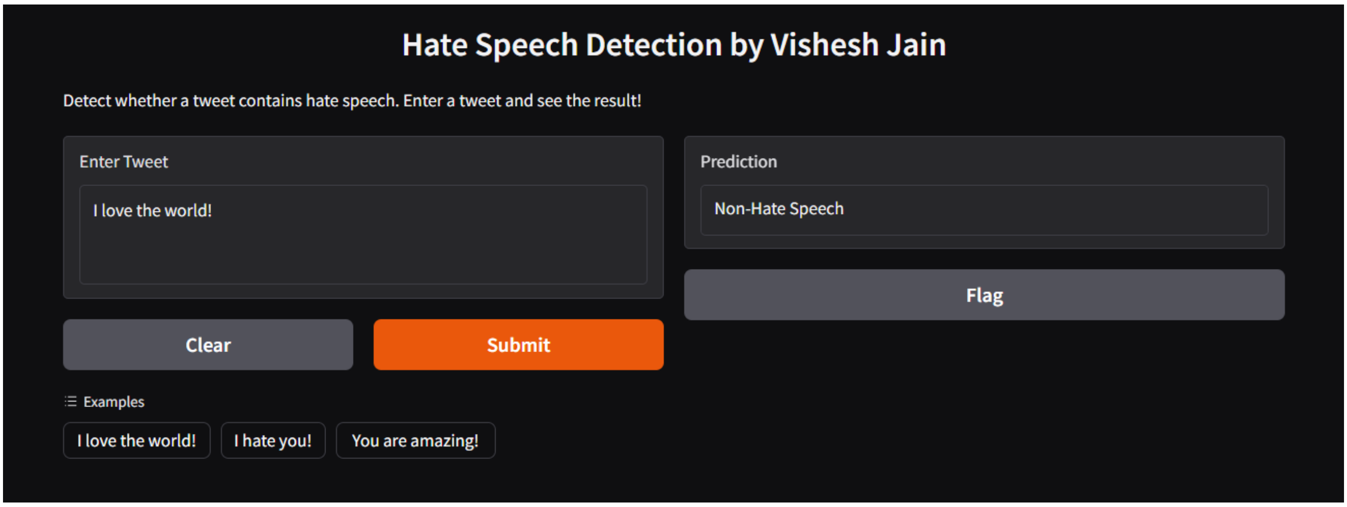
Detect whether a tweet contains hate speech. Enter a tweet and see the result!

Enter Tweet

I hate you!

Prediction

Hate Speech



Hate Speech Detection by Vishesh Jain

Detect whether a tweet contains hate speech. Enter a tweet and see the result!

Enter Tweet

I love the world!

Prediction

Non-Hate Speech

Conclusion

- Successfully built a Hate Speech Detection System using Logistic Regression.
- Developed a Gradio-based interface for real-time predictions.
- The project addresses a critical societal issue by identifying and flagging harmful content on social media.

References

[1]https://www.kaggle.com/datasets/arkhoshghalb/twitter-sentiment-analysis-hatred-speech?select=train.csv

[2]https://pypi.org/project/wordcloud/

[3]https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/

[4]https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[5]https://scikit-learn.org/1.5/modules/generated/sklearn.model_selection.GridSearchCV.html

[6]https://www.gradio.app/docs/python-client/introduction