A Hate Speech Detection Model for Snapchat.

GROUP 6

GAURAV MISHRA

VISHAL VENKATESH

1. Business Problem

Hate speech on platforms like Snapchat can have serious social consequences, including incitement to violence, discrimination, and mental health issues. Detecting and mitigating hate speech is crucial for maintaining a safe and inclusive online environment.

The goal of this project is to develop a robust system for detecting hate speech in text-based content on Snapchat.

2.Proposed Solution

1. Data Collection
2. Data Preprocessing
3. Feature Extraction
4. Model Development
5. Model Training

Dataset Description

Dataset Description

Contains user comments collected from different platforms, with labels indicating whether the comment is hateful or not.

Consists of three columns.
-**Platform**: The platform from which the comment was collected.
-**Comment**: The text of the user comment.
-**Hateful**: Label indicating whether the comment is hateful (1) or not (0).
-Source : Kaggle

Data Visualization

Below is an example of how the data is structured:

Platform	**Comment**	**Hateful**
Reddit	Damn I thought they had strict gun laws in Ger...	0
Reddit	I don't care about what it stands for or anythi...	0
Reddit	It's not a group it's an idea lol	0
Reddit	So it's not just America!	0
Reddit	The dog is a spectacular dancer considering he...	0

Data Preprocessing Steps:

Importing necessary libraries: Loading essential libraries like pandas, numpy, sklearn, etc.

Loading the dataset: Importing the dataset into a dataframe for analysis.

Checking for missing values: Identifying and handling any missing values in the data.

Splitting training and testing set: Dividing the data into training and testing sets to evaluate the model's performance.

Using TF-IDF Vectorizer: Converting text data into numerical format using TF-IDF for feature extraction.

Tokenization and Embedding Techniques

In Machine Learning Model

The embedding technique used is TF-IDF (Term Frequency-Inverse Document Frequency) Vectorization. TF-IDF helps in representing text data in a numerical form which is important for machine learning algorithms to process and learn from the data.

In Deep Learning Model

The embedding technique used is Word Embeddings. Word embeddings are a type of word representation that allows words to be represented as vectors in a continuous vector space. This technique captures the semantic meaning of words, making it suitable for deep learning models.

Modeling

Various models that I have used in the machine learning model:

- **Logistic Regression**: A simple yet powerful model for binary classification problems. It is easy to implement and interpret.
- **Support Vector Machine (SVM):** Effective in high-dimensional spaces and works well with a clear margin of separation.
- **Random Forest**: An ensemble method that uses multiple decision trees to improve the model's accuracy and prevent overfitting.

Modeling

Selected Model: Logistic Regression

Reason: Logistic Regression was chosen due to its simplicity, efficiency, and good performance on the dataset. It is also less computationally intensive compared to more complex models.

Modeling

Deep Learning Models

Various models that I have used in the deep learning model:

CNN (Convolutional Neural Network): Effective for text classification by capturing local features and patterns in the text.

Combination of CNN and Bidirectional LSTM (Long Short-Term Memory): Utilizes both convolutional layers for feature extraction and LSTM layers for capturing temporal dependencies in text data.

Modeling

Selected Model: CNN

Reason: CNN was chosen due to its superior performance in handling text data and its ability to extract meaningful features from the text.

Evaluation Metrics:

Logistic Regression Accuracy:

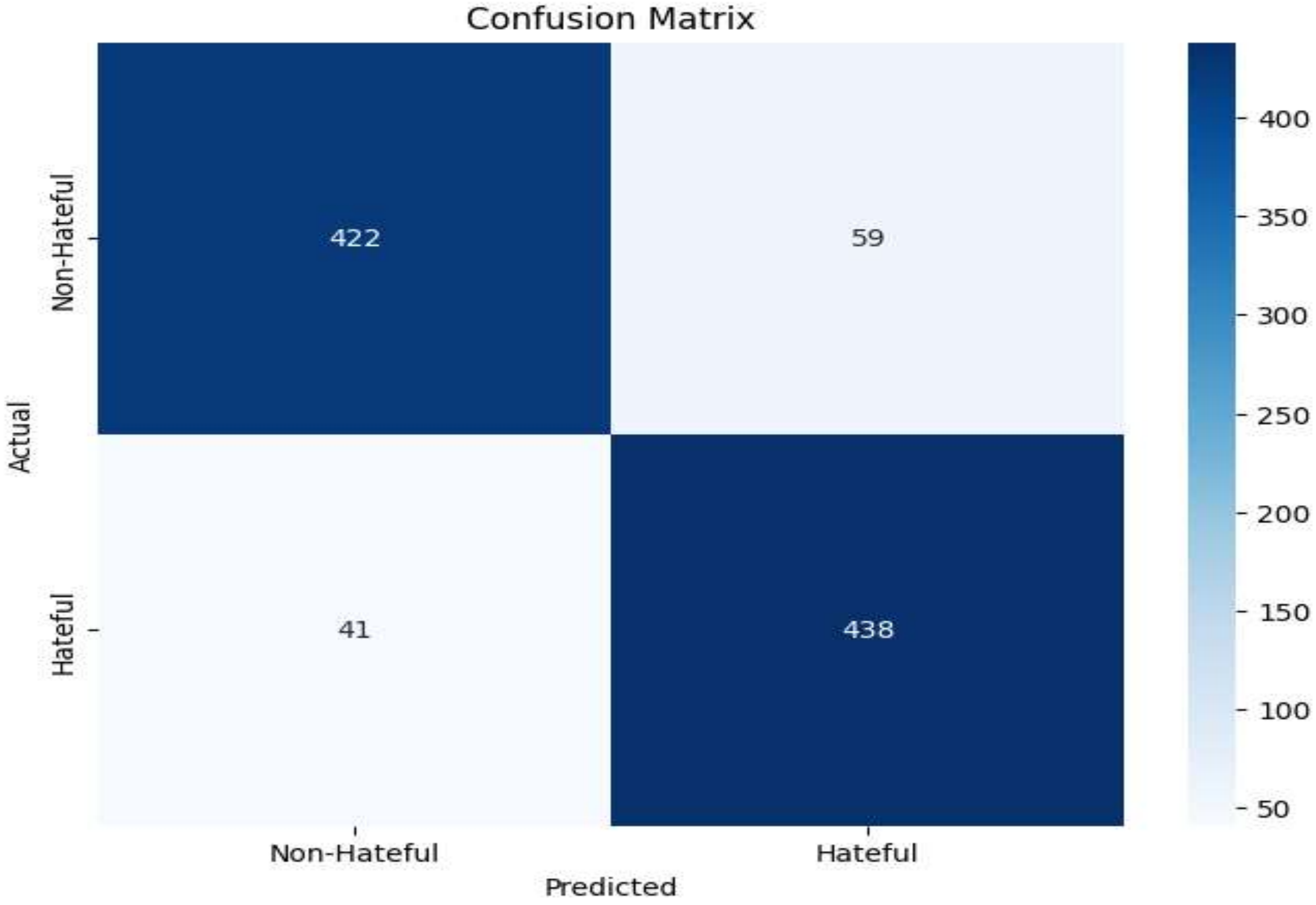
- Test Accuracy: 0.8958

	Precision	Recall	F1-Score	Support
0	0.91	0.88	0.89	481
1	0.88	0.91	0.90	479
Accuracy			0.90	960
Macro Avg	0.90	0.90	0.90	960
Weighted Avg	0.90	0.90	0.90	960

Evaluation Metrics:

Logistic Regression Accuracy:

•Test Accuracy: 0.8958



Evaluation Metrics:

Deep Learning Model:

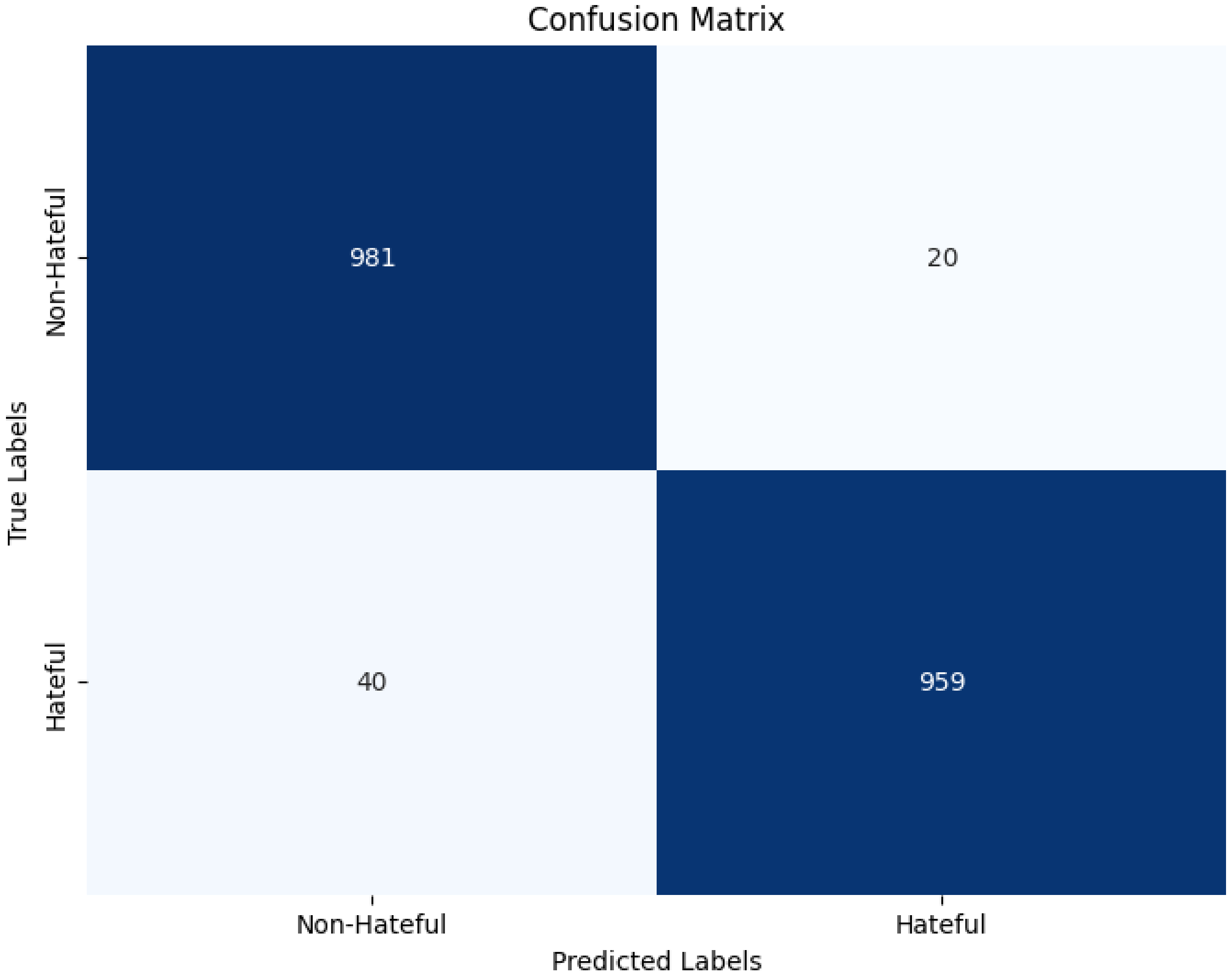
- Test Accuracy: 0.8980

	Precision	Recall	F1-Score	Support
0	0.88	0.92	0.90	1001
1	0.91	0.88	0.90	999
Accuracy			0.90	2000
Macro Avg	0.90	0.90	0.90	2000
Weighted Avg	0.90	0.90	0.90	2000

Evaluation Metrics:

Deep Learning Model:

•Test Accuracy: 0.8980



Thankyou