Data Science Intern at Data Glacier

Hate Speech Detection using Transformers (Deep Learning)

Team Members:

Farheen Fatima
Mohamed Derbeli

Table of contents:

Project Plan

Week	Date	Plan
Week 7	Sep 19, 2022	Problem statement, Data collection and Data report
Week 8	Sep 26, 2022	Data preprocessing
Week 9	Oct 2, 2022	Data preprocessing
Week 10	Oct 9, 2022	Building the model
Week 11	Oct 16, 2022	Building the model
Week 12	Oct 23, 2022	Result
Week 13	Oct 30, 2022	Report and presentation

Problem Statement

The term hate speech is understood as any type of verbal, written or behavioral communication that attacks or uses derogatory or discriminatory language against a person or group based on what they are. In other words, based on their religion, ethinicity, natiionality, race, color, ancestry, sex or other identity factor, In the problem, we will take you through a hate speech detection model with Machine Learning, Deep Learning and Python.

Hate Speech Detection is generally a task of sentiment classification. So, for training a model that can classify hate speech from a certain piece of text can be achieved by training it on data that is generally used to classify sentiments. So, for the task of hate speech detection model, we will use the Twitter tweets to identity tweets containing Hate Speech.

Dataset

For this project, the data is taken from Kaggle which has three features and 31962 instances. The features are Id, Label and Tweets. Labels are classified as: 1 represents Hate speech and 0 represents Free-speech.

Number of instances	31962
Number of Files	1
Number of Features	3
File Format	csv
Data_size	2.95MB

Table1: Data Information

Data PreProcessing

We performed the following data preprocessing steps on the text data.

1. **Check for Null values** - We checked for the null values in the data and there were no null values found.

2. **Check for imbalance data** - In this dataset, we found that the target column has imbalance data. To make the data balanced, we used oversampling technique.

3. **Remove Punctuation** - We removed the punctuation from the text data because it is not helpful in the sentiment analysis. We have used Regular Expression to do this.

4. **Remove URLs** - we removed the URLs because for hate speech detection we need only the text and not the URLs.

5. **Remove Tags and Special characters** - we removed @tags and symbols like [!";$()&,.:'/@?] using regular expressions.

6. **Convert to Lower case** - we converted all the letters to lowercase because NLP is case sensitive. Hence it will be converted as two different words even if the words are the same.

7. **Tokenization** - Tokenization is breaking the sentence into a list of words. This helps in understanding the context and interpreting the meaning of the text by analyzing the sequence of the words.

8. **Removing stopwords** - Stopwords like a, the, is, are, of, etc do not have meaning and hence we removed the stop words.

9. **Lemmatize** - Lemmatization is the process of grouping together the different forms of a word so that they can be analyzed as a single item. It brings context to the words, links words with similar meanings to one word. Basically, it converts the word to its root form.

10. **Word Cloud** - A word cloud is the visual representation of the text data, which is often used to depict keyword metadata on websites or to visualize free form Text.

Feature Extraction

After preprocessing the data, we padded the sequence of the words and then we created embedded matrix using the Glove word embedding and then we splitted the data into train and test using scikit learn's train_test_split method. We have used 80% of the data for training and 20% for the testing.

Build the Model:

Logistic Regression: Since the problem is a classification problem, first of all we have tested the LR using the provided data.

Logistic Regression, Random Forest, Naive Bays, SVM: We have noticed that the dataset is not balanced since 93% of the data is for the free speech. Therfore we have balanced the data and applied several classification algorithms.

NN model: We created a neural network model using Glove embedded in the first layer of the model. We trained the model using 'Adam' optimizer and 'Sigmoid' activation function with 100 neuron counts.

LSTM Model: We created a Long Short Term Memory (LSTM) model using Glove embedded in the first layer of the model. We trained the model using ‘Relu’ and ‘sigmoid’ activation and ‘adam’ optimizer.

Results

We evaluated the results using accuracy, precision, recall, F1 score and the confusion matrix.

1. **Unbalanced dataset:**

	precision	recall	f1-score	support
0	0.96	1.00	0.98	2984
1	0.95	0.35	0.51	213
accuracy			0.96	3197
macro avg	0.95	0.67	0.74	3197
weighted avg	0.95	0.96	0.95	3197

Logistic Regression, Accuracy Score: 0.955270566155771

2. **Balanced dataset:**

	precision	recall	f1-score	support
0	0.99	0.95	0.97	2948
1	0.95	0.99	0.97	2996
accuracy			0.97	5944
macro avg	0.97	0.97	0.97	5944
weighted avg	0.97	0.97	0.97	5944

Logistic Regression, Accuracy Score: 0.9703903095558546

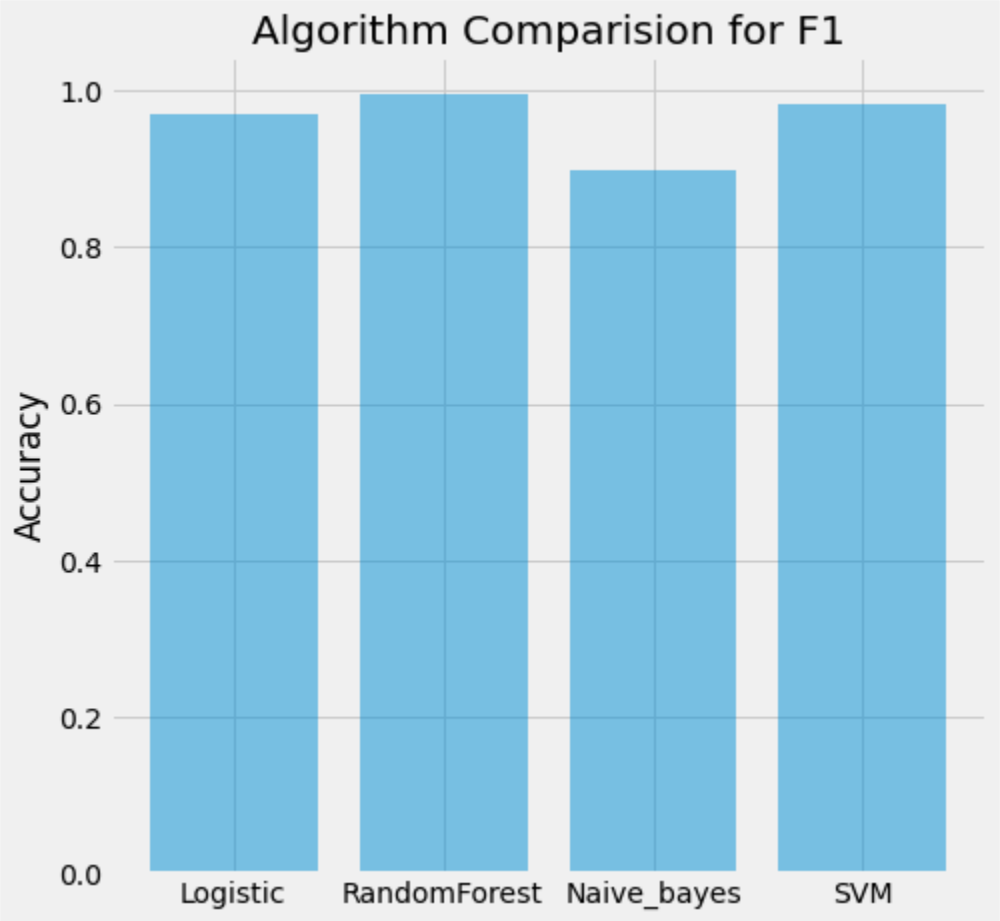
	precision	recall	f1-score	support
0	0.84	0.99	0.91	2948
1	0.98	0.81	0.89	2996
accuracy			0.90	5944
macro avg	0.91	0.90	0.90	5944
weighted avg	0.91	0.90	0.90	5944

Naive Bayes, Accuracy Score: 0.8977119784656796

	precision	recall	f1-score	support
0	1.00	0.99	0.99	2948
1	0.99	1.00	0.99	2996
accuracy			0.99	5944
macro avg	0.99	0.99	0.99	5944
weighted avg	0.99	0.99	0.99	5944

Random Forest, Accuracy Score: 0.9944481830417228

A comparison among the above algorithms is detailed by the following diagram:



	precision	recall	f1-score	support
0	0.96	0.97	0.96	5937
1	0.53	0.43	0.47	456
accuracy			0.93	6393
macro avg	0.74	0.70	0.72	6393
eighted avg	0.93	0.93	0.93	6393

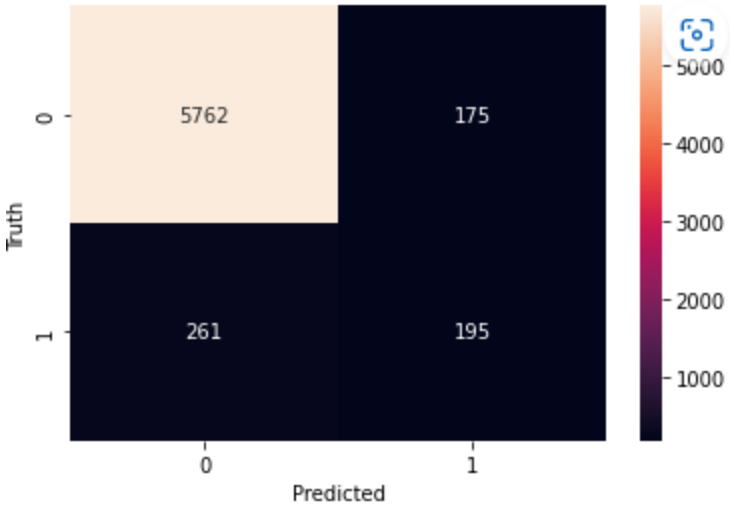


Table 2: Evaluation metrics using Neural Network model

	precision	recall	f1-score	support
0	0.97	0.98	0.97	5937
1	0.68	0.57	0.62	456
accuracy			0.95	6393
macro avg	0.83	0.77	0.80	6393
weighted avg	0.95	0.95	0.95	6393

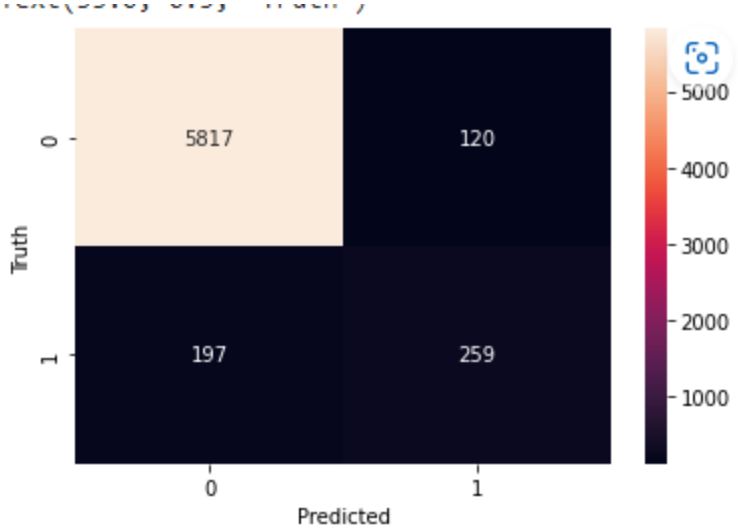


Table 3: Results using LSTM Model

References:

1. https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech?select=train_E6oV3lV.csv
2. https://analyticsindiamag.com/hands-on-guide-to-word-embeddings-using-glove/
3. https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/