Data Science Intern at Data Glacier

Project: Hate Speech Detection using Transformers (Deep Learning)

Week 11: Deliverables

Name: Ishwarya Arulvel

Email: ish351994@gmail.com

Country: Canada

Specialization: Data Analyst

Batch Code: LISUM14

Date: 14 Dec 2022

Submitted to: Data Glacier

Table of Contents:

1. Project Plan

Weeks	Date	plan
Weeks 07	Nov 16, 2022	Problem Statement, Data Collection, Data Report
Weeks 08	Nov 23, 2022	Data Preprocessing (Text Cleaning)
Weeks 09	Nov 30, 2022	Data Preprocessing (Preprocessing Operation)
Weeks 10	Dec 7, 2022	Building the Model
Weeks 11	Dec 14, 2022	Model Result Evaluation
Weeks 12	Dec 21, 2022	Flask Development + Heroku
Weeks 13	Dec 28, 2022	Final Submission (Report + Code + Presentation)

2. 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, ethnicity, nationality, race, color, ancestry, sex or another identity factor. In this problem, we will take you through a hate speech detection model with Machine Learning and Python.

Hate Speech Detection is 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 used to classify sentiments. So, for the task of hate speech detection model, we will use the Twitter tweets to identify tweets containing Hate speech.

3. Data Collection

The Data is about Twitter hate Speech taken from Kaggle [1] which contains the 3 number of features and 31962 number of observations. Dataset using Twitter data, it was used to research hate-speech detection. The text is classified as: hate-speech, offensive language, and neither. Due to the nature of the study, it is important to note that this dataset contains text that can be considered racist, sexist, homophobic, or offensive.

Table 1: Data Information

Total number of observations	31962
Total number of files	1
Total number of features	3
Base format of the file	csv
Size of the data	2.95 MB

4. Data Preprocessing

In part, we explain the data preprocessing approach that we apply in the text data.

4.1 Text Cleaning

First, we cleaned our text because it was so messy data.

4.1.1 Lowercase

Converting a word to lower case (NLP -> nlp). Words like Racism and racism mean the same but when not converted to the lower case those two are represented as two different words in the vector space model (resulting in more dimensions). Therefore, we convert all text words into lower case letters.

4.1.2 Remove Punctuation

It is important to remove the Punctuation because is not important. Therefore, we remove that Punctuation to do that we use regular expression.

4.1.3 Remove URLs

In this part, we remove URLs because we are working on hate speech application which detect the hate and free speech and to get the output, we need to give only text not URLs therefore, we remove the URLs because we need only clean text input.

4.1.4 Remove @tags

In this part, we remove @tags which basically used when we mentioned someone So, it's doesn't concern to our application therefore, we remove @tags by using regular expressions.

4.1.5 Remove Special Characters

Remove Special Characters is essentially the following set of symbols [!"#$%&'()*+,-./:;<=>?@[]^_`{|}~] which basically doesn't have meaning. Therefore, we remove that kind of symbol because we don't need that. To remove it we use the python isalnum method.

4.2 Preprocessing Operation

In this part, we implement the preprocessing operation

4.2.1 Tokenization

Tokenization is breaking the raw text into small chunks. Our text data is into paragraphs so to convert into work tokenize we use nltk work tokenize library. These tokens help in

understanding the context or developing the model for the NLP. Tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.

4.2.2 Removing Stop Words

Stop Words are basically ‘a,’ ‘is,’ ‘the,’ ‘are’ etc. If we look at our dictionary, then these words do not have meaning and don’t need that to build Hate speech detection application. To remove stop words from a sentence, we divide text into words which we did above in tokenization and then remove the word if it exits in the list of stop words provided by NLTK. To do that, we first import the Stop Words collection from the nltk.

4.2.3Lemmatization

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is like stemming but it brings context to the words. So, it links words with similar meanings to one word. Like the word Intelligently, intelligence, convert into root form intelligent.

4.2.4 Word Cloud

A World cloud is a visual representation of text data, which is often used to depict keyword metadata on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color.

5. Exploratory Data Analysis (EDA)

5.1 TF-IDF Model

Once the dictionary is ready, we apply Term Frequency-Inverse Document Frequency (TF-IDF) model, and we take 2000 most frequent words from dictionaries for each Hate/Free Speech of the whole dataset. Each word count vector contains the frequency of 2000 words in the whole dataset file.

5.1.1 Split the Data into Train into Test

In this part, we split the data into Train. And we split 80% for training and 20% for tests. Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model. Data splitting is an important aspect of data science, particularly for creating models based on data.

5.2 Build the Model

5.2.1 Logistic Regression with TF-IDF on N-Grams

This process explains how to run a classification algorithm and more specifically a logistic regression of a "The hate and free speech detection on twitter" using as features the TD-IDF of unigrams, bi-grams, and trigrams. We can easily apply any classification, like Random Forest, Support Vector Machines etc. Finally, it finds whether the text is hated speech or free speech. logistic regression:

Logistic regression is a supervised machine learning method which is like linear regression but instead of using a linear equation it uses a sigmoid function which makes the output value in specific range which is used for text classification also.... Hate Speech Detection in Twitter using Natural Language Processing.

6.Result Evaluation and Discussion

In this chapter, we evaluate our Result and define the evaluation criteria to calculate the performances of our best classification model.

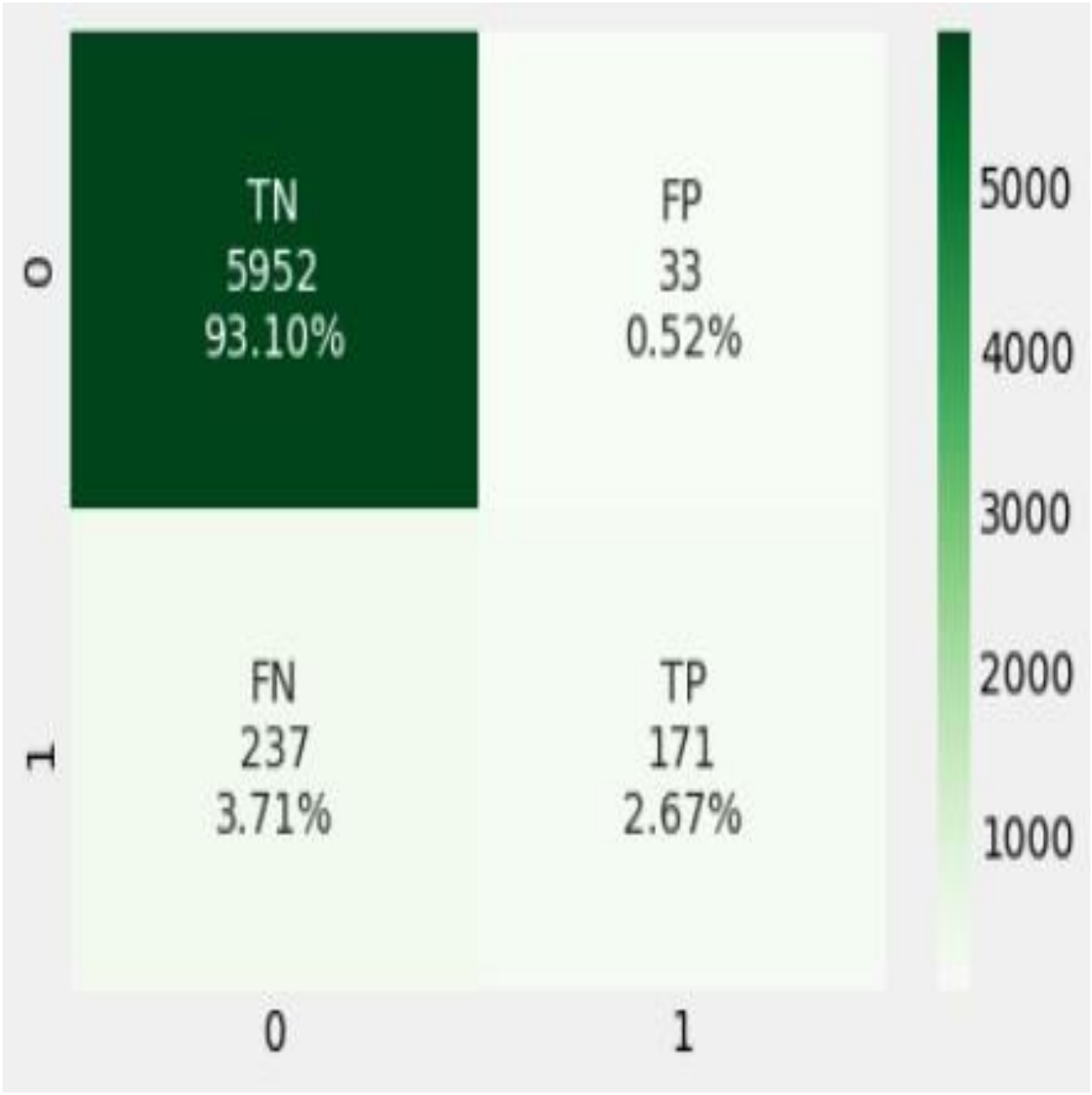
6.1Evaluation Criteria

The confusion matrix was used to evaluate the classification models throughout the training process. The confusion matrix is a table that compares predicted and actual outcomes. It is frequently used to describe a classification model's performance on a set of test data.

Table 1: Confusion Matrix

Class	**Predicted Negative**	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

Important metrics were constructed from the confusion matrix in order to evaluate the classification models. In addition to the accurate classification rate or accuracy, other metrics for evaluation included True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Precision, F1 score, and Misclassification rate.



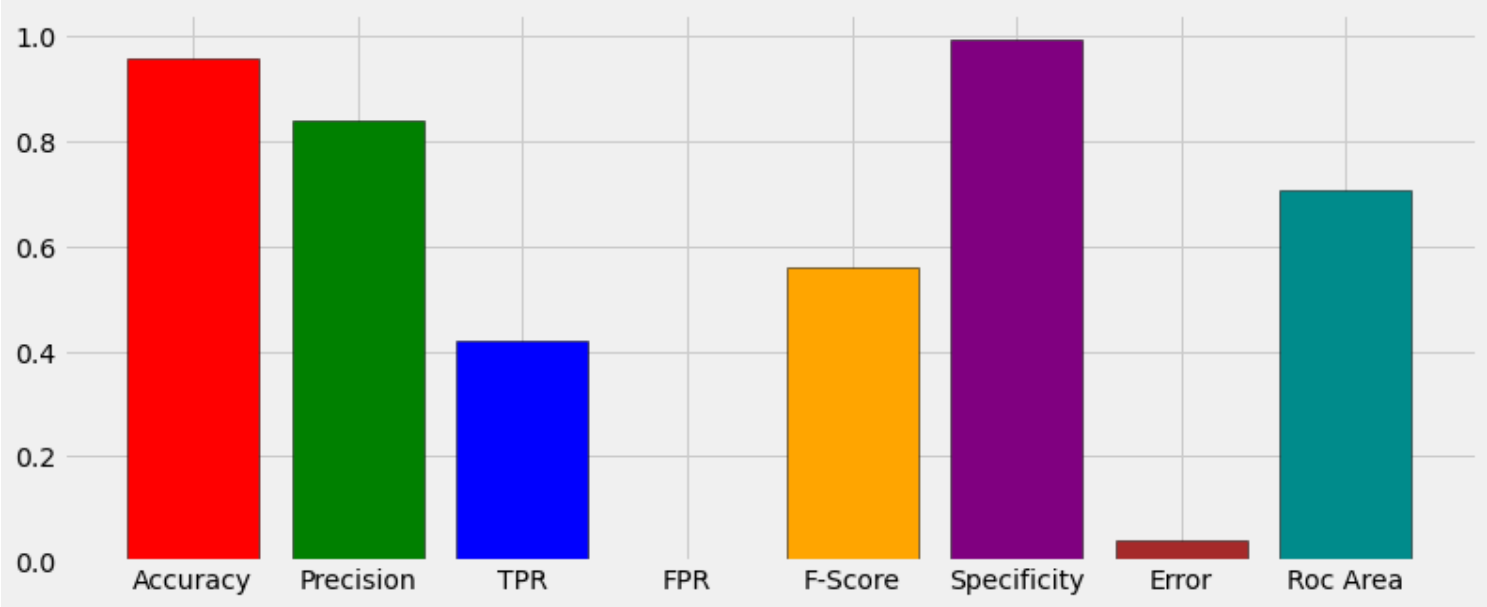
Confusion Matrix

Below table shows the result that we evaluate based on confusion matrix result

Table : Results

Classifiers	Accuracy	Precision	TPR	FPR	F1 Score	Error Rate	Specificity
CNN with LSTM	0.9577	0.8382	0.4191	0.0055	0.5588	0.0422	0.9944

Below you can see the visualization result of above table as well.



Reference

[1] https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech?select=train_E6oV3lV.csv