Data Science Internship

at Data Glacier

Project: Hate Speech Detection using Transformers

(Deep Learning)

Week 11: Deliverables

Name: Amrapali Mhaisgawali

University: -

Email: amrapali10@gmail.com

Country: United Kingdom

Specialization: Data Science

Batch code: LISUM14

Date: 15/12/2022

Submitted to: Data Glacier

Index

1. Problem Statement:

The term hate speech is understood as any type of verbal, written or behavioural 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, colour, ancestry, sex or another identity factor. In this problem, we will take you through a hate speech detection model with Machine Learning and Python.

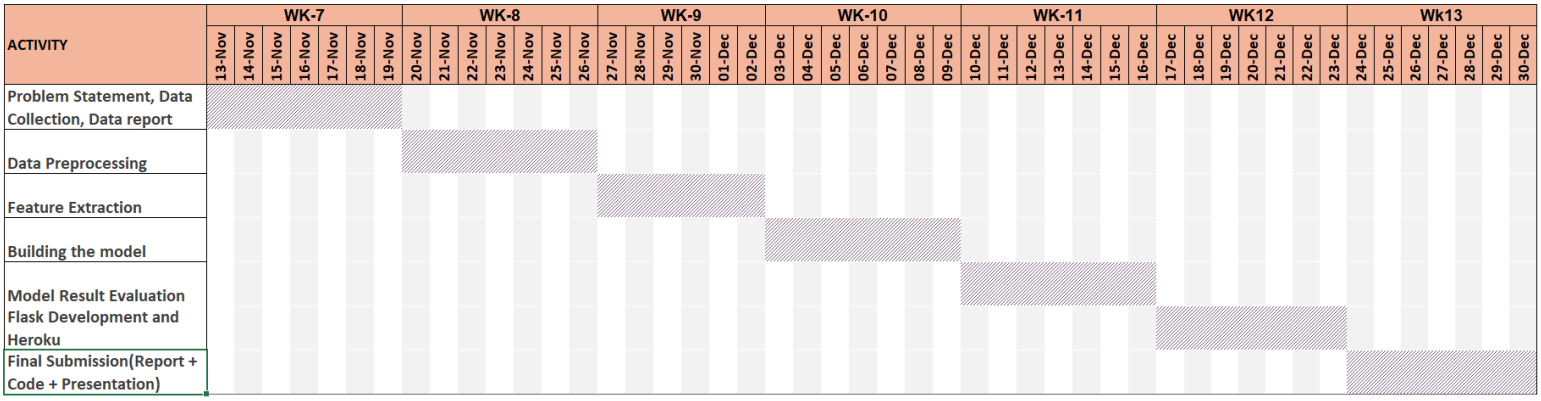
Hate Speech Detection is generally a task of sentiment classification. So, for training, a model that can classify hate speech from a certain piece of text can be achieved by training it on a data that is generally used to classify sentiments. So, for the task of hate speech detection model, we will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

Our goal is to classify tweets into two categories, hate speech or non-hate speech. Our project analyzed a dataset CSV file from Kaggle containing 31,962 tweets.

2. Business Understanding:

Social media has experienced incredible growth over the last decade, both in its scale and importance as a form of communication. The nature of social media means that anyone can post anything they desire, putting forward any position, whether it is enlightening, repugnant or anywhere between. Depending on the forum, such posts can be visible to many millions of people. Different forums have different definitions of inappropriate content and different processes for identifying it, but the scale of the medium means that automated methods are an important part of this task. Hate-speech is an important aspect of this inappropriate content.

3. Project Lifecycle



4. Data Collection

The Data 'Twitter hate Speech' is used for detection of hate speech taken from Kaggle [1] which contains the 3 features and 31962 number of observations. Data from Twitter website 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.

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

Table: Data Information

5. Data Pre-processing

Following are the steps in pre-processing

5.1 Text Cleaning

Text cleaning is the first step in data pre-processing

5.1.1 Lowercase

Data is converted to lower case. Words like User and user 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, data was converted into lower case letter.

5.1.2 Remove Punctuation

It is important to remove the Punctuation as they are not important. Punctuation are removed using regular expression.

5.1.3 Remove URLs

URLs are not important to detect the hate and free speech so they are removed from text.

5.1.4 Remove @tags

@tags are removed which basically used to mentioned someone. So, it's doesn't concern to detect hate speech therefore, they are removed by using regular expressions.

5.1.4 Remove Special Characters

Special Characters which basically don't have meaning. Therefore, they are removed. In order to remove we use python isalnum method.

5.2 Pre-processing operations

Following pre-processing operations are used for model building

5.2.1 Tokenization

Tokenization is breaking the raw text into small chunks. Text data used is in paragraph so to convert it into word tokenize nltk word_tokenize library is used. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analysing the sequence of the words.

5.2.2 Removing stop words

StopWords is basically 'a,' 'is,' 'the,' 'are' etc. These words do not have meaning and don't need to build Hate speech detection application. To remove stop words from a sentence, text is divided into words which is output of tokenization and then stop words are removed if it exits in the list of stop words provided by NLTK. To do that, StopWords is used which is imported from nltk collection.

5.2.3 Lemmatization

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed 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.

5.2.4 Lemmatization

A Worldcloud 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.

Below you see the hate speech and free speech WorldCloud:

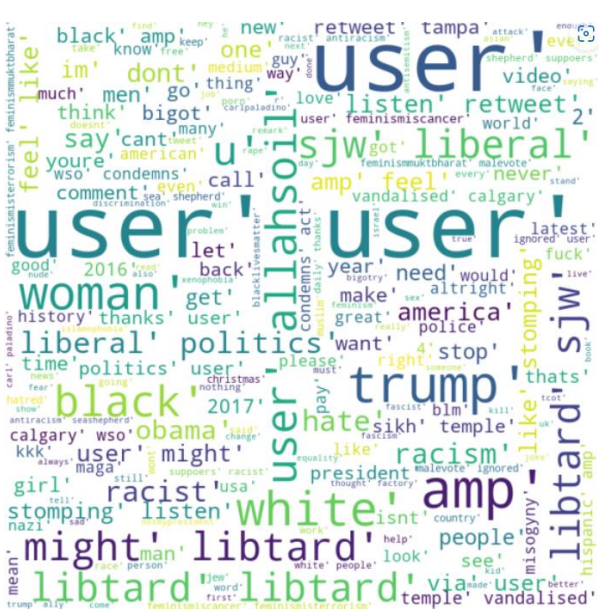


Fig 1 Hate Speech Vs Free Speech Word Cloud

6. Feature Extraction

6.1 TF-IDF Model

Once the dictionary is ready, Term Frequency-Inverse Document Frequency (TFIDF) model is applied using 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.

7. Model Building

7.1 Split the data into train-test

Data is split into training and testing, 80% data is used for training and 20% for testing. Data splitting is an important aspect of data science, particularly for creating models based on data

7.2 Build the model (RNN)

Model is built using RNN(LSTM Tensorflow 4.5.1)

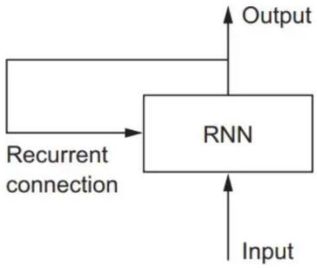


Fig 2 RNN Model

7.3 Model evaluation

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.

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

Table: Confusion Matrix

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.

Fig 3 Confusion Matrix

Below Table shows the final result that we evaluate on the basis of confusion matrix result.

Classifiers	Accuracy	Precision	TPR	FPR	F1 Score	Error Rate	Specificity
LSTM	0.985	0.974	0.99	0.02	0.985	0.01	0.973

Table: Result

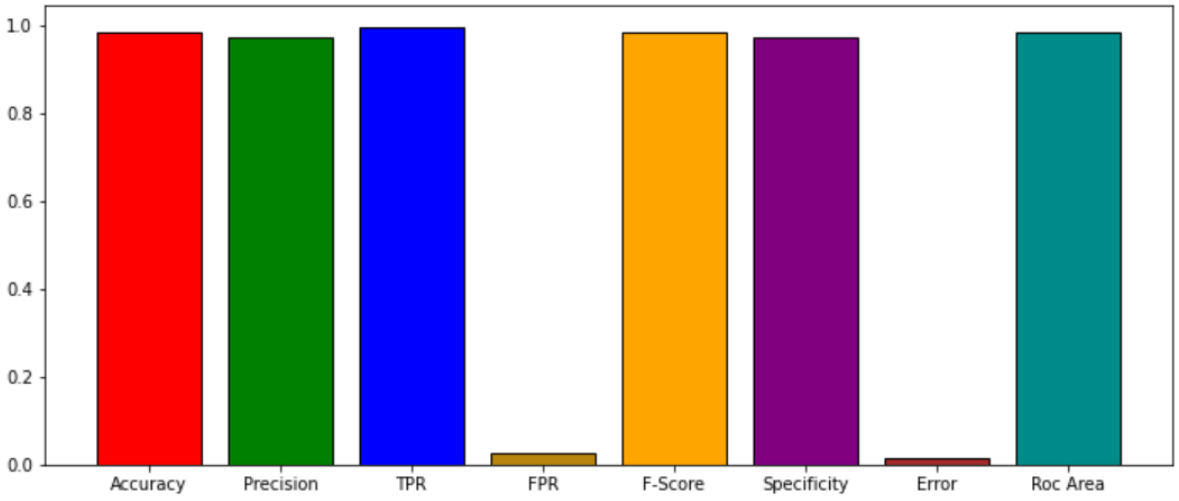


Fig 4 Visualization of Results

References

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