



Data Science Internship at Data Glacier

Project: Hate Speech Detection using Transformers
(Deep Learning)

Week 12: Deliverables

Name: Amrapali Mhaigawali

University: -

Email: amrapali10@gmail.com

Country: United Kingdom

Specialization: Data Science

Batch code: LISUM14

Date: 22/12/2022

Submitted to: Data Glacier

Index

1.	Problem Statement	3
2.	Business Understanding	3
3.	Project Lifecycle	3
4.	Data Collection	4
5.	Data Pre-processing	4
	5.1 Text Cleaning.....	4
	5.1.1 Lower case.....	4
	5.1.2 Remove Punctuation.....	4
	5.1.3 Remove URL.....	4
	5.1.4 Remove @ tags.....	4
	5.1.5 Remove special characters	4
	5.2 Pre-processing Operations.....	5
	5.2.1 Tokenization	5
	5.2.2 Remove stop words	5
	5.2.3 Lemmatization	5
	5.2.4 WordCloud	5
6.	Feature Extraction.....	6
7.	Model Building.....	6
	7.1 Train Test Split.....	6
	7.2 Build the model (RNN).....	6
	7.3 Model Evaluation	7
8.	Application Design	9
	8.1 Running ML model into Flask framework	9
	8.2 Procedure for running the application	9
9.	Conclusion	10
10.	References	11

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

ACTIVITY	WK-7						WK-8						WK-9					WK-10					WK-11					WK12					Wk13															
	13-Nov	14-Nov	15-Nov	16-Nov	17-Nov	18-Nov	19-Nov	20-Nov	21-Nov	22-Nov	23-Nov	24-Nov	25-Nov	26-Nov	27-Nov	28-Nov	29-Nov	30-Nov	01-Dec	02-Dec	03-Dec	04-Dec	05-Dec	06-Dec	07-Dec	08-Dec	09-Dec	10-Dec	11-Dec	12-Dec	13-Dec	14-Dec	15-Dec	16-Dec	17-Dec	18-Dec	19-Dec	20-Dec	21-Dec	22-Dec	23-Dec	24-Dec	25-Dec	26-Dec	27-Dec	28-Dec	29-Dec	30-Dec
Problem Statement, Data Collection, Data report																																																
Data Preprocessing																																																
Feature Extraction																																																
Building the model																																																
Model Result Evaluation Flask Development and Heroku																																																
Final Submission(Report + Code + Presentation)																																																

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 exists 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:

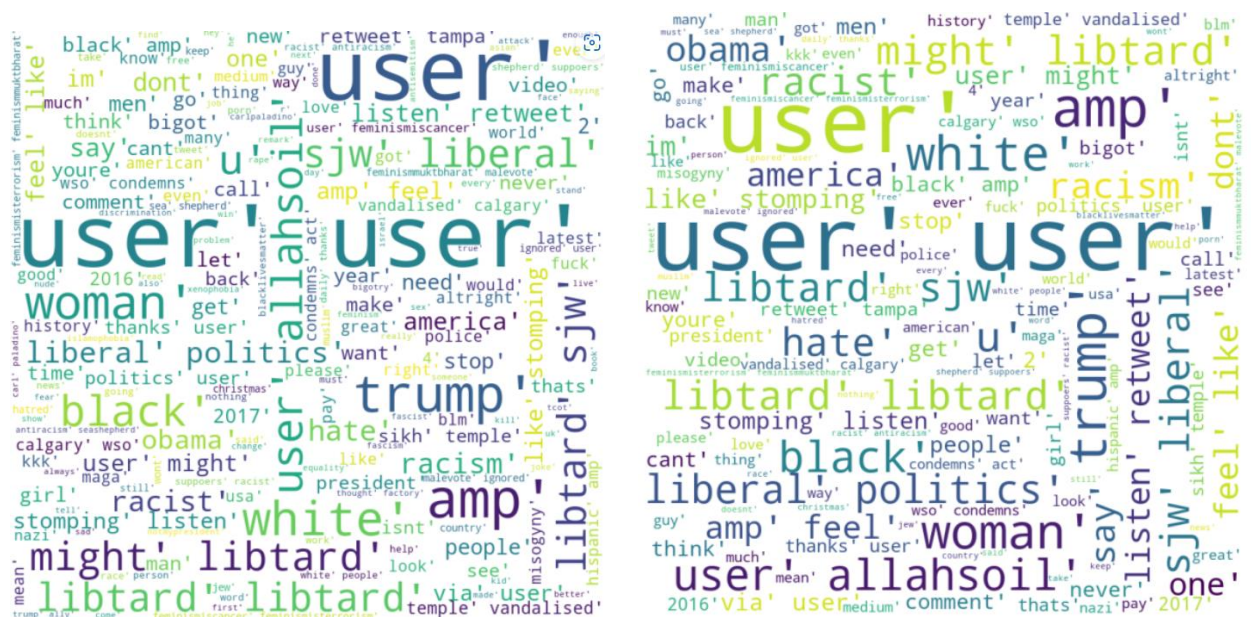


Fig 1 Hate Speech Vs Free Speech Word Cloud

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)

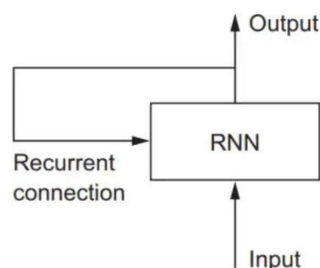


Fig 2 RNN Model

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

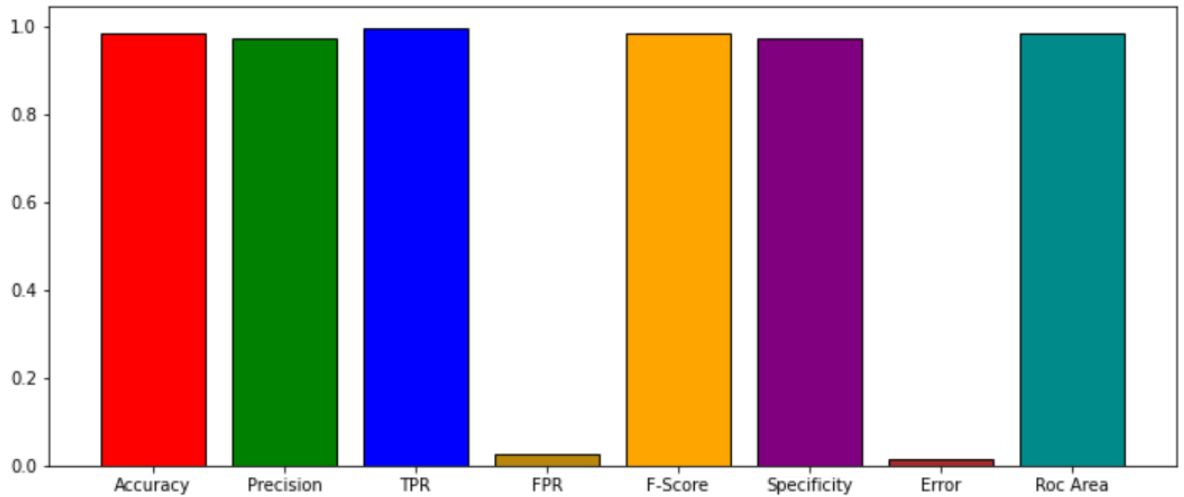


Fig 4 Visualization of Results

8. Application Design

8.1 Running ML model into Flask framework

The twitter hate speech dataset was used to build a prediction model that will accurately classify hate speech. LSTM algorithm is used to build the model. After training the model, it is desirable to have a way to persist the model for future use without having to retrain. To persist pickle file of model was created. Web application was developed that consists of a simple web page with a form field to enter a message. After submitting the message to the web application, it will render it on a new page which gives the result of hate speech/free speech.

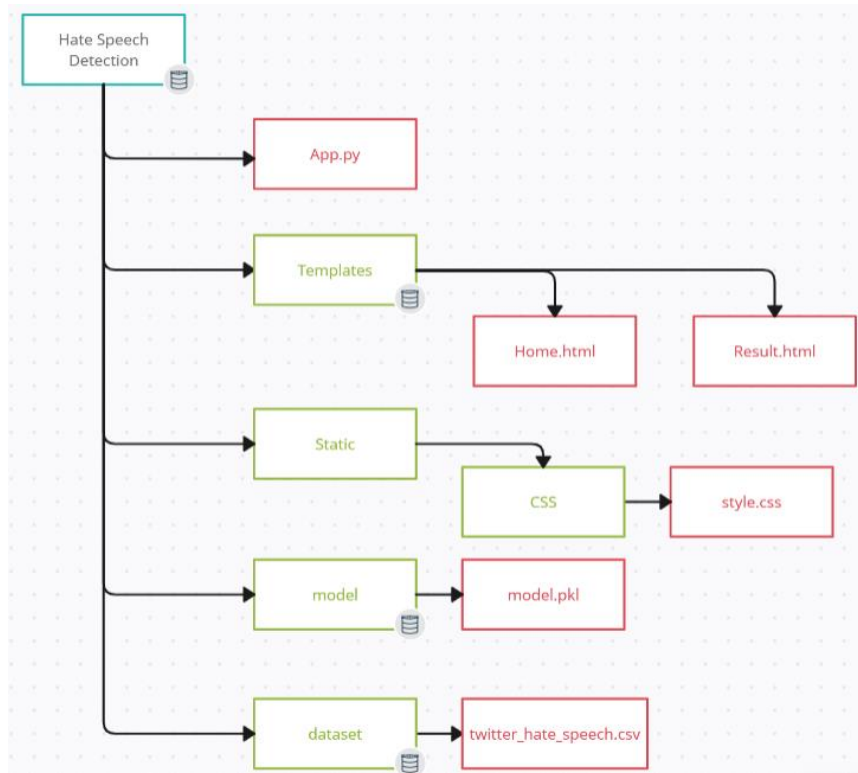


Fig 5 Directory Structure

Above directory structure is used for application.

8.2 Procedure for running the application

Run the application using command `python app.py`

```
C:\Windows\System32\cmd.exe - python app.py

C:\Users\AMRAPALI\Documents\internship\Week12\project>.\app\scripts\activate

(app) C:\Users\AMRAPALI\Documents\internship\Week12\project>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 112-323-031
```

Now open a web browser and navigate to `http://127.0.0.1:5000/`, it should open a simple website.

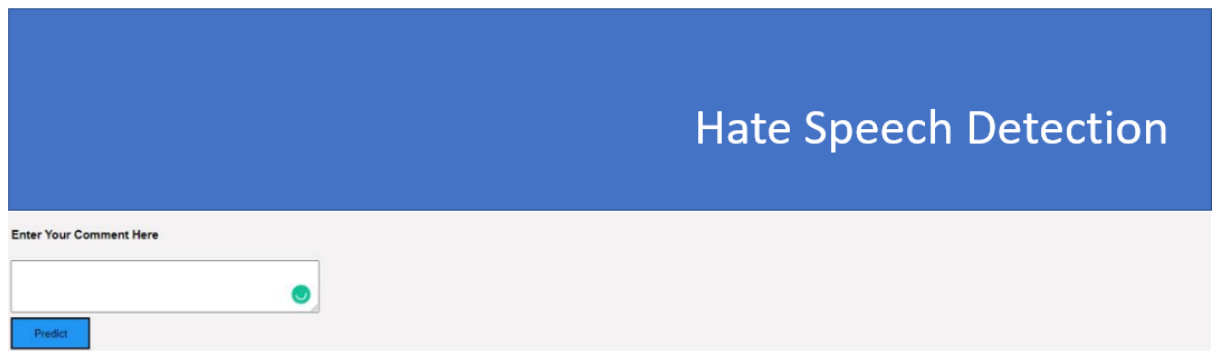


Fig 6: Twitter Hate Speech application

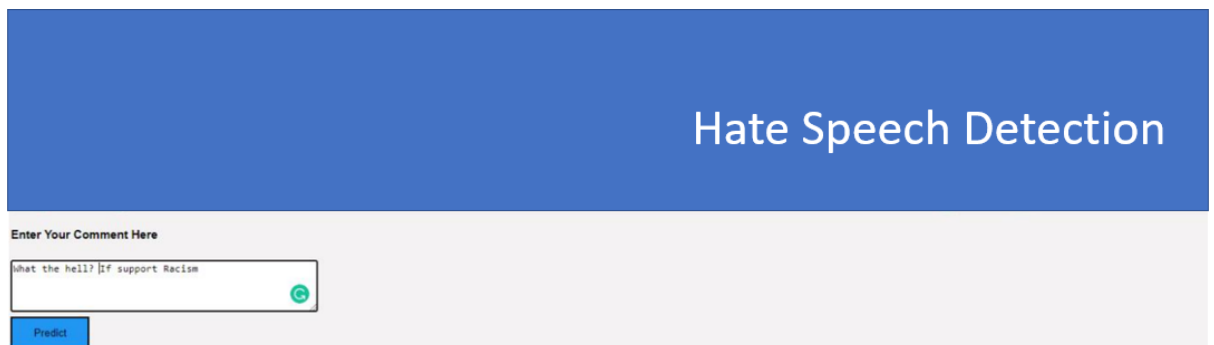


Fig 7: Input to Twitter Hate Speech application

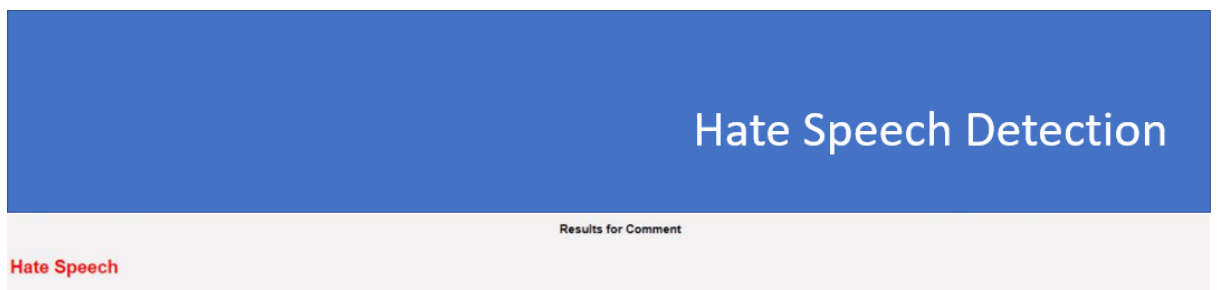


Fig 8: Output of Twitter Hate Speech application

9. Conclusion

The goal of this project was to find capable methods and settings that could be used to help the detection of Hate and Free Speech of twitter. The error rate of the model is not zero, so still, some incorrect can be classified as true by the model.

References

- [1] https://www.kaggle.com/datasets/vkrahul/twitter-hate-peech?select=train_E6oV3IV.csv