NLP Hate Speech Detection

Virtual Internship

10-08-2021

By:

Manoj Kumar Thangaraj

Agenda

- Problem Statement
- Approach
- EDA Summary & Recommendations
- Featurization Technique
- Model Building
- Training & Evaluation
- Inference

Problem Statement

Overview

To detect a tweets that has hateful content and at the same time prevent prediction incorrectly. The term hate speech is understood as any type of verbal, written or behavioural communication that attacks or uses derogatory or discriminatory language .

Solution

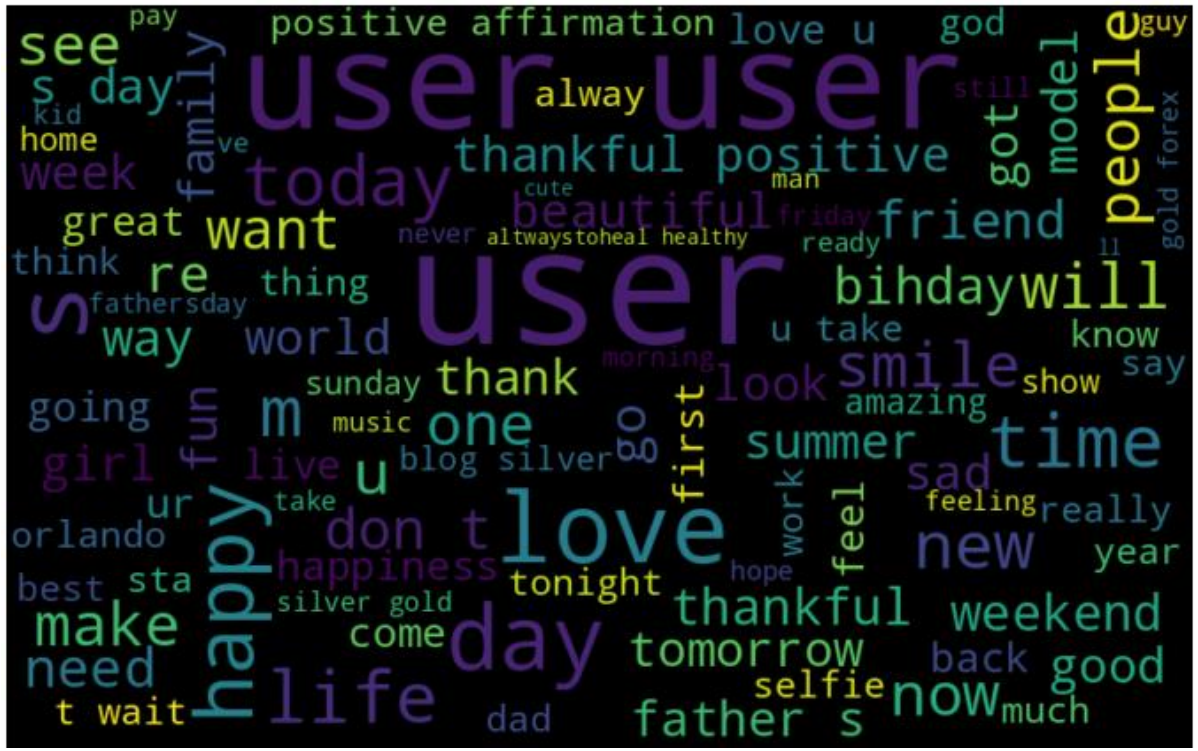
Provide them with the trained model that can accurately predict the tweets that contain hateful content.

Approach

The general approach of the project is divided into important major section in which each of the section is equally important to the project. They are outlined as follows.

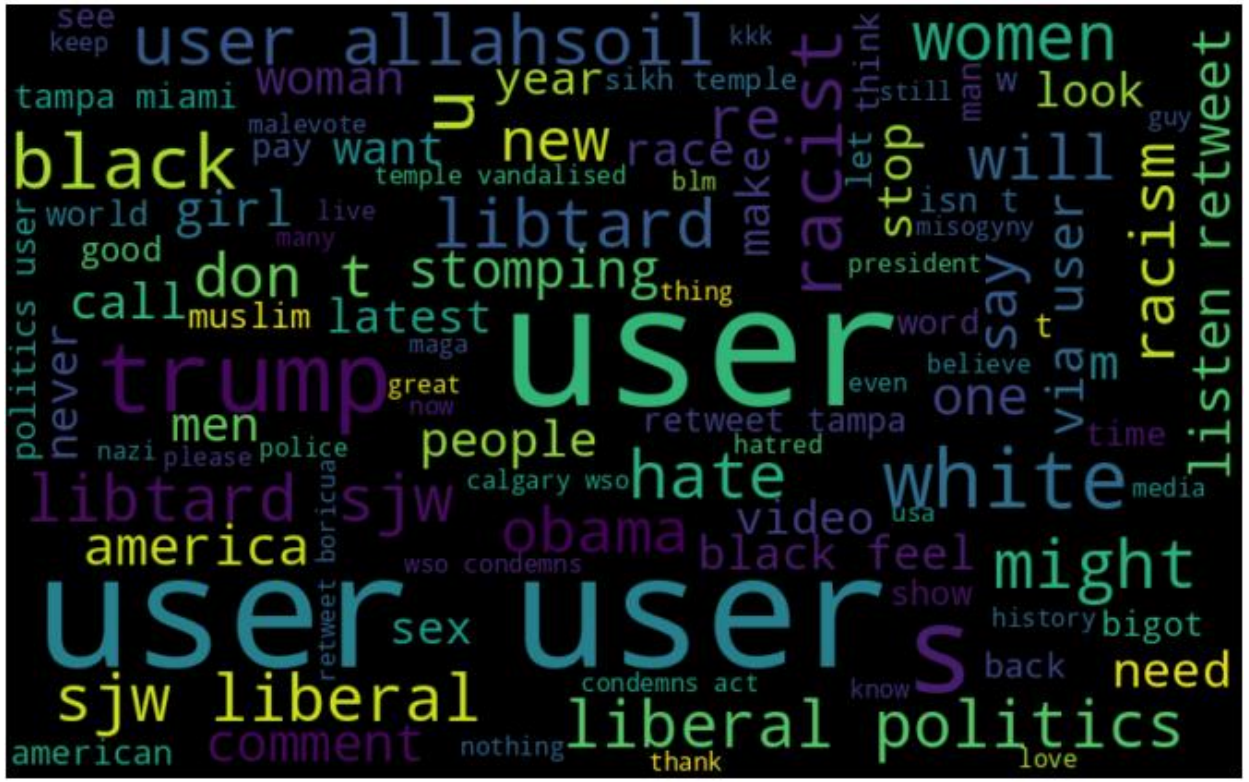
1. Downloading the data
2. Preparing and Preprocessing the data
3. Data Visualization
4. Transform the data
5. Model development
6. Training and Validation
7. Model Inference
8. Amazon SageMaker Deployment

EDA



The most often used words in the non-hateful tweets.

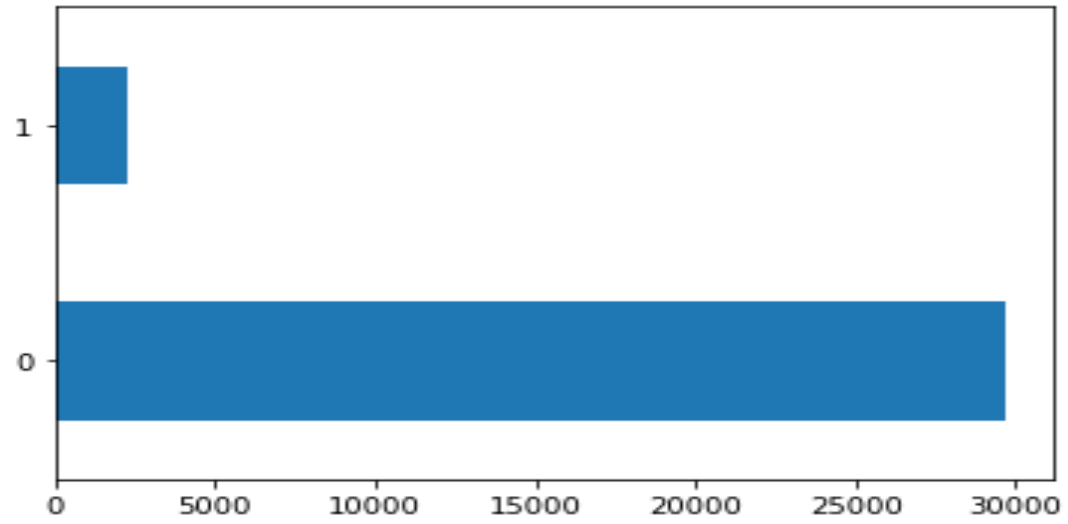
EDA



The most used words in the tweets with hateful content.

EDA

```
0    29720
1     2242
Name: label, dtype: int64
```



We observed there is a high imbalance in between the labels in the dataset.

EDA Summary & Recommendations

- The EDA has been done on the textual data to see which words used the most on each class and found that the word ‘User’ has been used much on both classes. This word is nothing, but the common word used by twitter.
- Therefore, this has no useful meaning for that word, and it might influence the model in predicting wrong class. Therefore, we can remove that word from the corpus.
- Also, we found many local slang words found in the corpus, which is not correctly spelled in English. Attempt has been made to replace those words by building a dictionary with key as the slang words and their correct word as value and replacing them across the corpus. Building such dictionary in would be useful in a long term in developing the model.
- Class Imbalance has been found on the data, so for designing the model, the under-sampling technique adopted as oversampling may overfit the model. There would be some loss in the model, but it can be compensated in a long run by acquiring more data with the other class in future

Featurization Technique

Featurization done with **Torchtext** library.

Tasks include

- Tokenization.

- Building vocabulary.

- Load them on the training and testing Iterators

Model Building

Model - Deep Learning

Architecture – PyTorch

Building **Transformer architecture** from scratch.

Training & Validation

- The training loop has 20 epochs.

- The training accuracy is 88%

- The test validation accuracy is 80%

Evaluation

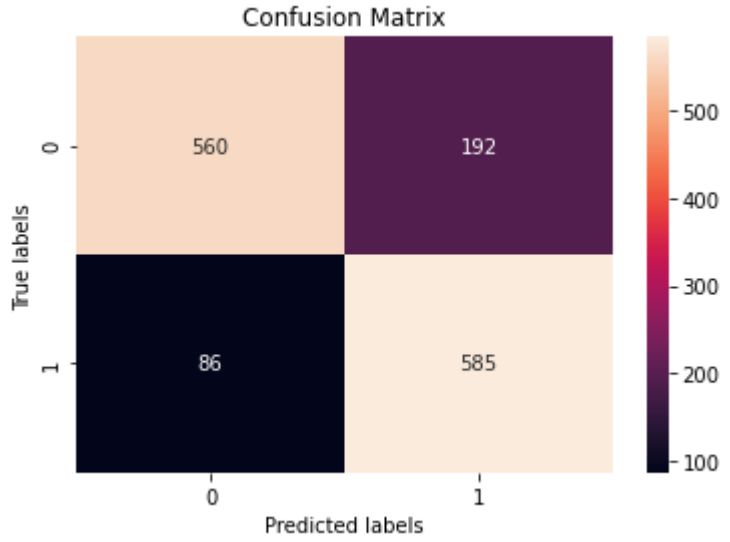
	precision	recall	f1-score	support
0	0.87	0.74	0.80	752
1	0.75	0.87	0.81	671
accuracy			0.80	1423
macro avg	0.81	0.81	0.80	1423
weighted avg	0.81	0.80	0.80	1423

- The ability to predict the hateful content over all predicted positives is 75% .

- The ability to predict the hateful content over all hateful tweets is 87% .

Evaluation

Confusion Matrix



The confusion matrix also shows the number of truly and falsely predicted contents.

Model Inference

```
#Lets use it for predicting the model

x = predict(model, "how the #altright uses  &amp; insecurity to lure men into #whitesupremacy")
print(x)
```

```
1
```

The model performs well, as it detected the hate spech with the correct label. However, the model needs to be udated every now then with the available data to make the model perform well.

The trained model was able to predict the hateful content accurately.

Thank You