IDENTIFY THE SENTIMENTS

Divyakumar Prajapati

OVERVIEW

Problem statement

The objective of this project is perform Sentiment Analysis to detect racist or sexist tweet. So task is to classify racist and sexist tweets from other tweets.

What is Sentiment Analysis?

Sentiment analysis (or opinion mining) uses natural language processing and machine learning to interpret and classify emotions in subjective data.

What we did?

1. Text Preprocessing.
2. Data Exploration.
3. Feature Extraction.
4. Model Building.

Text Inspection

Let's check out a few non racist/sexist tweets.

```
train[train['label'] == 0].head(10)
```

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ð□□¦ð□□¦ð□□¦
4	5	0	factsguide: society now #motivation
5	6	0	[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get there. #allshowandnogo
6	7	0	@user camping tomorrow @user @user @user @user @user @user @user dannyâ□¦
7	8	0	the next school year is the year for exams.ð□□¯ can't think about that ð□□ #school #exams #hate #imagine #actorslife #revolutionschool #girl
8	9	0	we won!!! love the land!!! #allin #cavs #champions #cleveland #clevelandcavaliers â□¦
9	10	0	@user @user welcome here ! i'm it's so #gr8 !

Now check out a few racist/sexist tweets.

```
train[train['label'] == 1].head(10)
```

	id	label	tweet
13	14	1	@user #cnn calls #michigan middle school 'build the wall' chant " #tcot
14	15	1	no comment! in #australia #opkillingbay #seashepherd #helpcovedolphins #thecove #helpcovedolphins
17	18	1	retweet if you agree!
23	24	1	@user @user lumpy says i am a . prove it lumpy.
34	35	1	it's unbelievable that in the 21st century we'd need something like this. again. #neverump #xenophobia
56	57	1	@user lets fight against #love #peace
68	69	1	ð□□©the white establishment can't have blk folx running around loving themselves and promoting our greatness
77	78	1	@user hey, white people: you can call people 'white' by @user #race #identity #medâ□¦
82	83	1	how the #altright uses & insecurity to lure men into #whitesupremacy

ut; double click to hide output

Text Inspection

Let's check dimensions of the train and test dataset.

```
train.shape, test.shape
```

```
((31962, 3), (17197, 2))
```

Train set has 31,962 tweets and test set has 17,197 tweets.

Let's have a glimpse at label-distribution in the train dataset.

```
train["label"].value_counts()
```

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

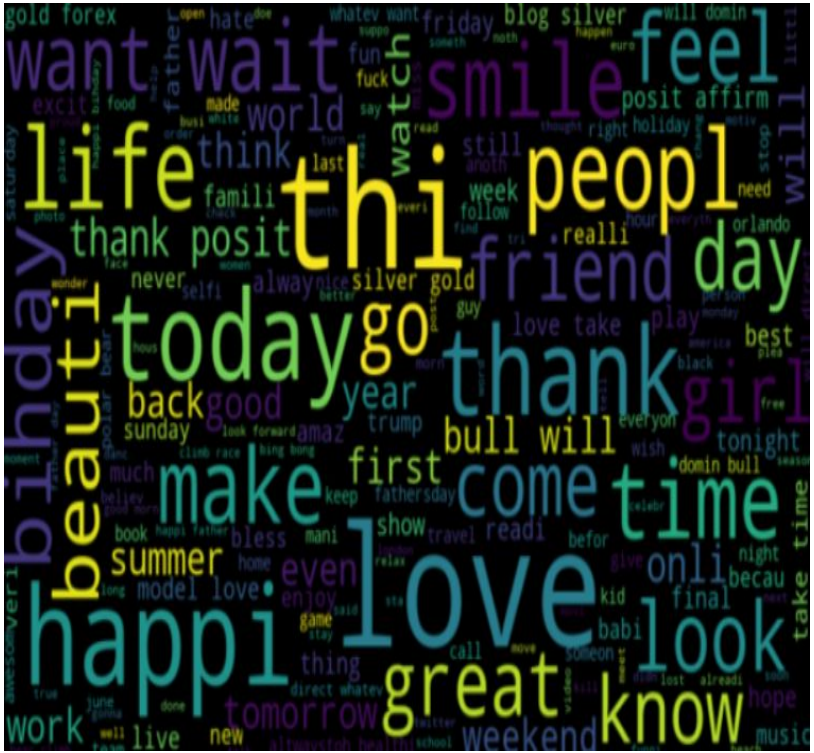
In the train dataset, we have 2,242 (~7%) tweets labeled as racist or sexist, and 29,720 (~93%) tweets labeled as non racist/sexist. So, it is an imbalanced classification challenge.

Data Cleaning and Preprocessing

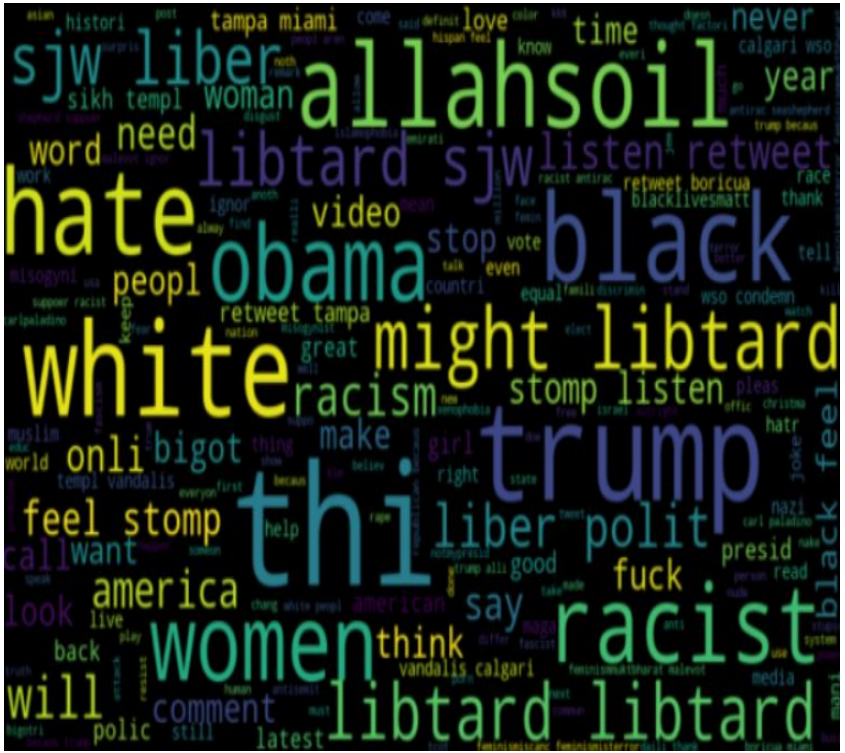
1. Removing Twitter Handle: Example @handles
2. Removing Punctuations, Special character, Number: Example, !$#%^&*
3. Removing short words (length > 3): Because they are very common and contains less information
4. Lowercasing words
5. Spelling correction
6. Text Normalisation: Stemming, Lemmatization. To remove same word coming with different representation.
7. Object Standardization: Remove Slang words.
8. Stopwords Removal
9. Converting Vulgar Words: In Tweets some vulgar words were represented as particularly ‘$#%^&’, so we this representation to some vulgar words to get better features.

Data Exploration and Visualisation

Common words in Non Racist/ sexist tweets



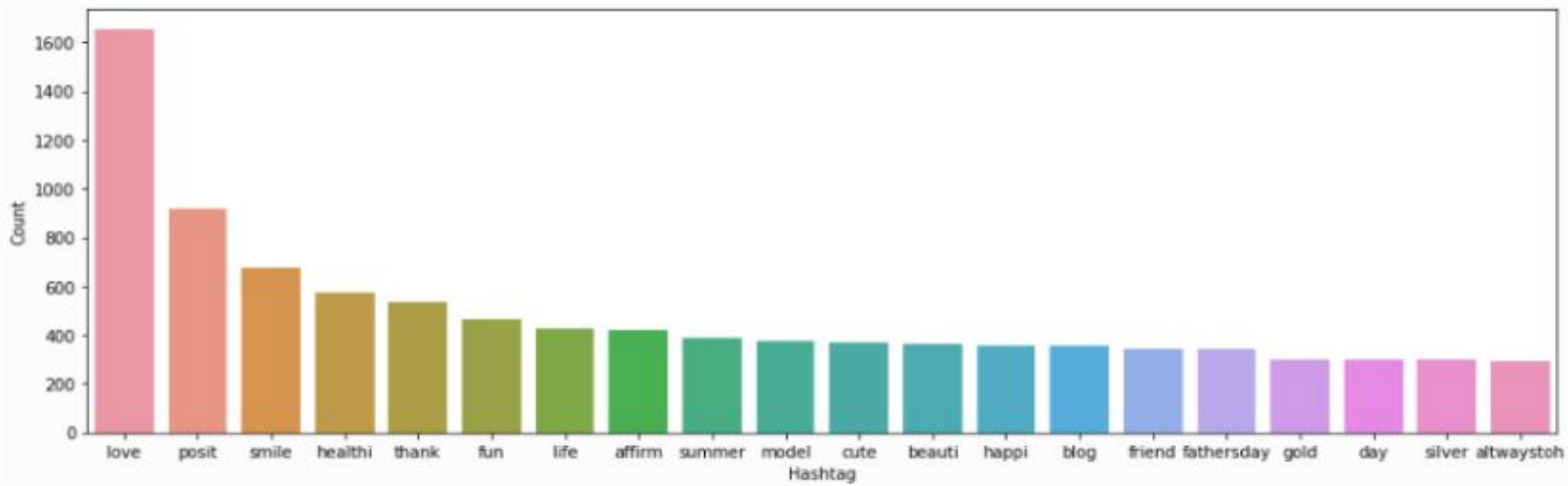
Common words in Racist/ sexist tweets



Data Exploration and Visualisation

20 most common HasHtag in Non- Racist/ Sexist tweets

```
plt.figure(figsize=(16,5)) ax = sns.barplot(data=d, x= "Hashtag", y = "Count") ax.set(ylabel
= 'Count') plt.show()
```

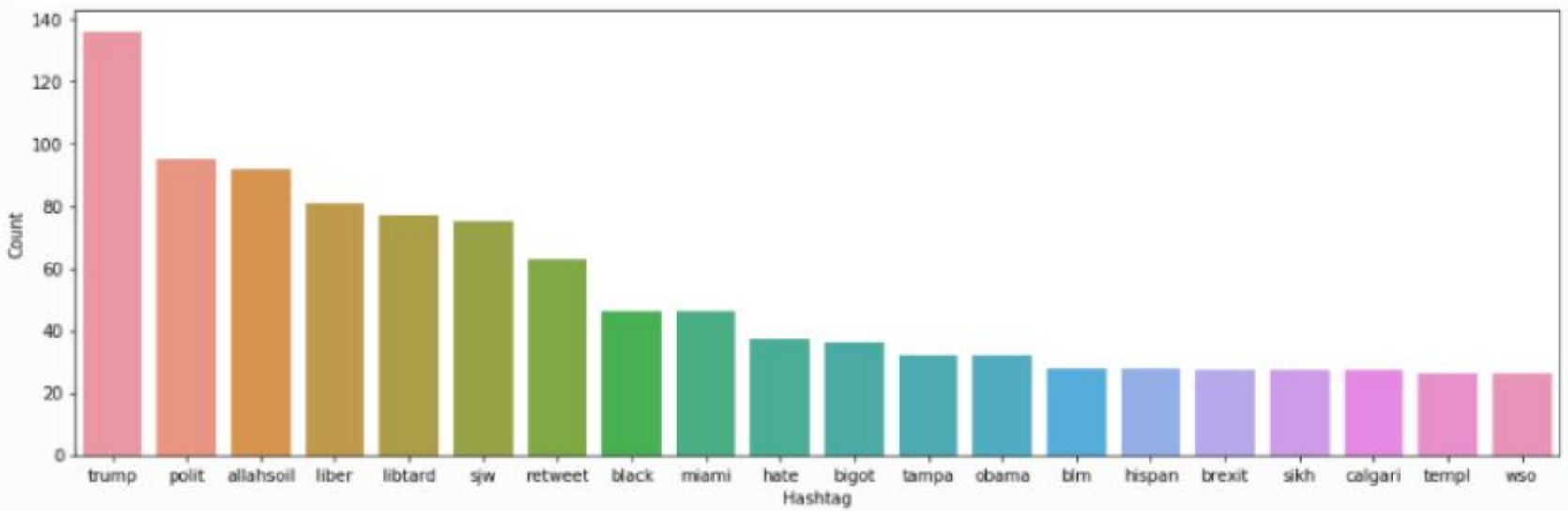


All words in this are positive. Now lets see about racist tweets.

Data Exploration and Visualisation

20 most common hashtag in Racist/ Sexist tweets

```
plt.figure(figsize=(16,5)) ax = sns.barplot(data=e, x= "Hashtag", y = "Count")
```



As expected the words are mostly negative. It would be good idea to include this hashtag in features.

Feature Construction

1. **Bag of words**: Bag of Words model is used to preprocess the text by converting it into a *bag of words*, which keeps a count of the total occurrences of most frequently used words.
   Sklearn library: sklearn.feature_extraction.test.CountVectorizer
2. **TF-IDF:** Scikit learn **Tfidftransformer** and **Tfidfvectorizer** aim to do the same thing, which is to convert a collection of raw documents to a matrix of TF-IDF features.
   Sklearn library: sklearn.feature_extraction.test.CountVectorizer.
3. **Word2Vec**: The word embeddings is the representation of text into vectors. The Idea here is similar words will have minimum distance between there vectors.

Model Building

Algorithm Used:

1. Linear Regression
2. Naive Bayes
3. Random forest
4. XGBoost
5. ANN

Metrics Used:

F1 score: It is weighted average of Precision and Recall. There it takes false positive and False Negative into account. Therefore it is suitable for unbalanced training set.

True Positive(TP): Actual positive cases which also got predicted right.

True Negative(TN): Actual Negative cases which also got Negative right.

False Positive(FP): Actual positive cases which got predicted negative.

False Negative(FN): Actual negative cases which got predicted positive.

Recall = TP/(TP+FN)

Precision = TP/(TP+FP)

ANN Model

Layer (type)	Output Shape	Param #
input_char (InputLayer)	(None, 100)	0
embedding_7 (Embedding)	(None, 100, 100)	25600
dropout_14 (Dropout)	(None, 100, 100)	0
conv1d_7 (Conv1D)	(None, 98, 250)	75250
global_max_pooling1d_7 (Glob	(None, 250)	0
dense_7 (Dense)	(None, 100)	25100
dropout_15 (Dropout)	(None, 100)	0
activation_7 (Activation)	(None, 100)	0
main_output (Dense)	(None, 4)	404

Total params: 126,354
Trainable params: 126,354
Non-trainable params: 0

None

Model Building

Best performance was giving by ANN with TF-IDF feature matrix.
Accuracy: 0.8823

Thank you