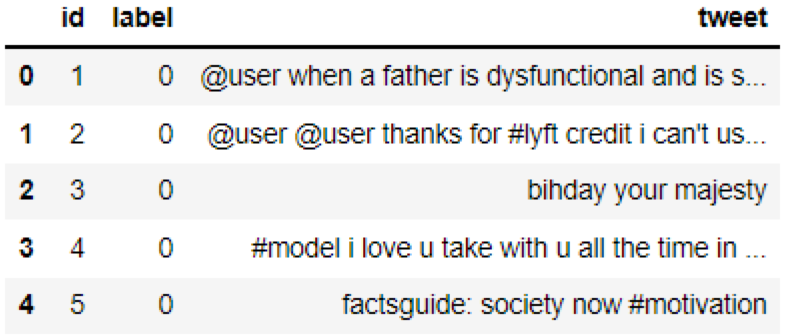
CAPSTONE PROJECT – PREDICTING HATEFUL AND RACIST TWEETS

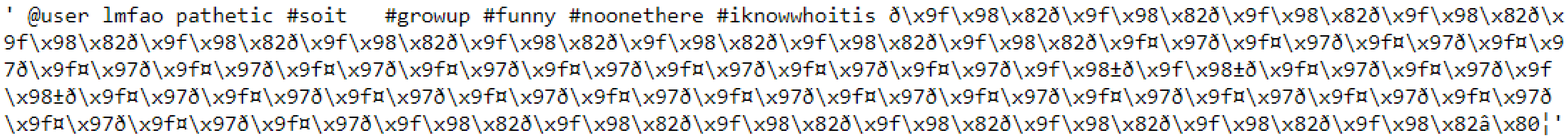
This dataset contains nearly 32K tweets which are labeled based on having racist or sexist content. This dataset is analyzed using traditional and neural networks to detect and determine offensive tweets. There is a test set of ~17000 tweets on which the best model is applied to detect negative tweets.

Part 1 – Data Wrangling

The initial data has 31962 rows X 4 columns. The columns have recordIDs of the tweets, the tweet itself and the label marking them as sexist or not., Initially there were no "null" values. A histogram plot was made that showed 93% of the tweets were labeled as un-offensive and 7% were hateful. The longest tweet had 274 characters and a word cloud was generated to understand the frequency of words. However, it seemed most of the characters were non-English (garbage) characters.

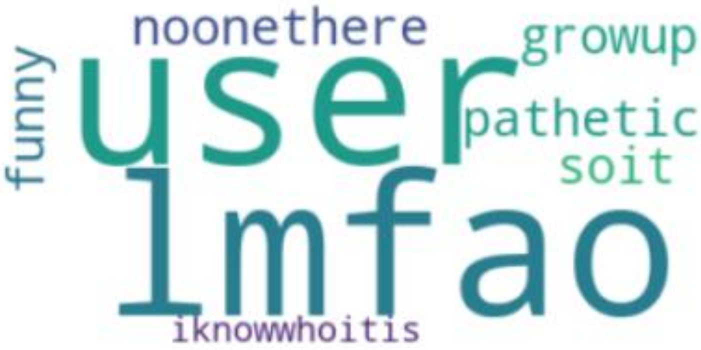


	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation



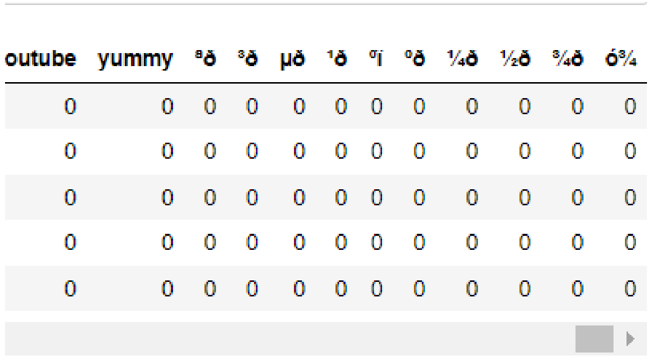
```
' @user lmfao pathetic #soit   #growup #funny #noonethere #iknowwhoitis ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x
9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x9
7ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f\x98±ð\x9f\x98±ð\x9f¤\x97ð\x9f¤\x97ð\x9f
\x98±ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð
\x9f¤\x97ð\x9f¤\x97ð\x9f¤\x97ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82ð\x9f\x98\x82â\x80¦'
```

All garbage characters need to be removed at preprocessing stage A check on duplicate values on "RecordID" revealed no duplicates.



Bag of words of the longest tweet.

Initially, CountVectorizer was used to vectorize the tweets. Checking the columns, it was Observed that there were several garbage characters in al tweets:



outube	yummy	ªð	³ð	µð	¹ð	ºï	ºð	¼ð	½ð	¾ð	ó¾
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Garbage characters.

Part 2 – Pre-processing

Initially, the length of each tweet was obtained using a word tokenizer. Only alphanumeric words were taken. The aim was to introduce a new feature column that will display the length of each tweet. A TfidfVectorizer was then used to vectorize the tweets. The reason to use TfidfVectorizer is that it tries to highlight words in each tweet as opposed to "across all tweets" using the inverse density function. Only about 200 words were chosen for developing the models keeping in mind the time constraint; >6 hours was required to tune gradient boost model using 500 feature words. Also, monogram and bigram words were chosen in the model. The garbage characters were added to the list of English stop words and used to separate from the tweets.

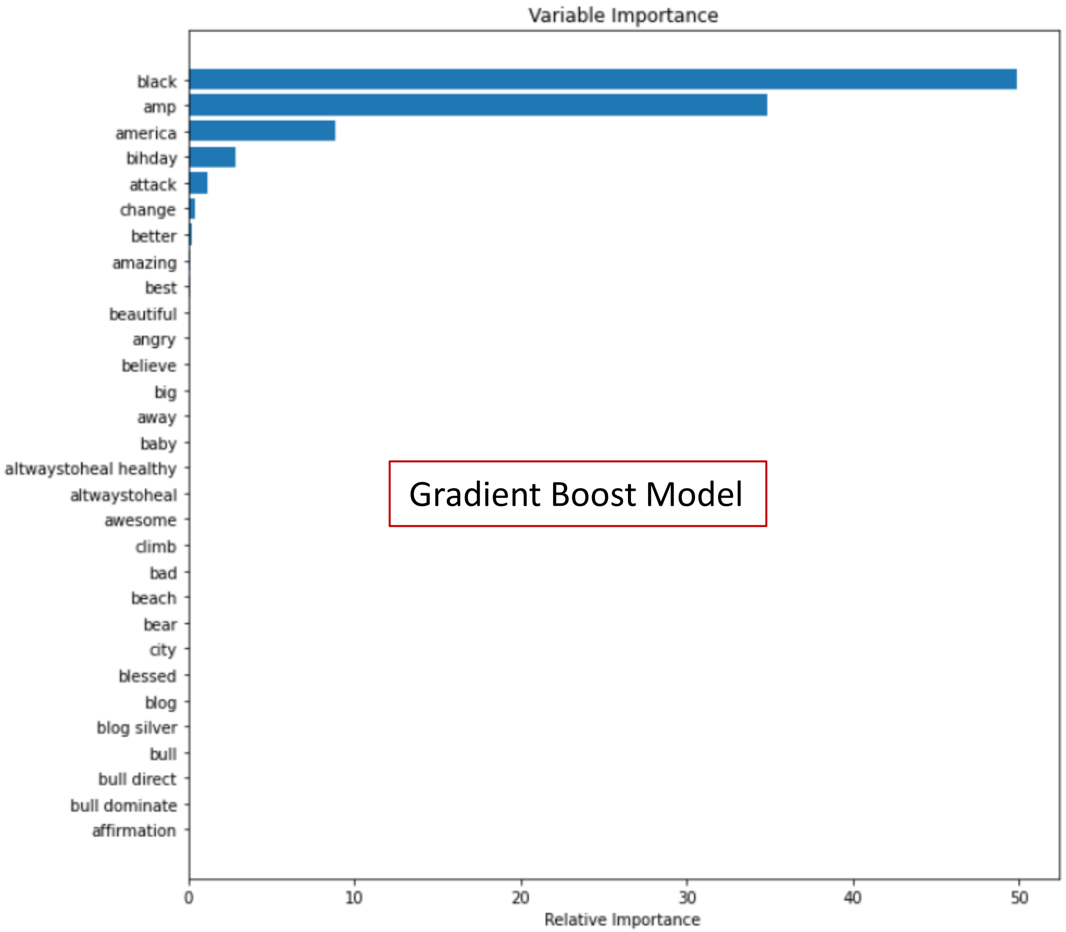
Part 3 - Modeling

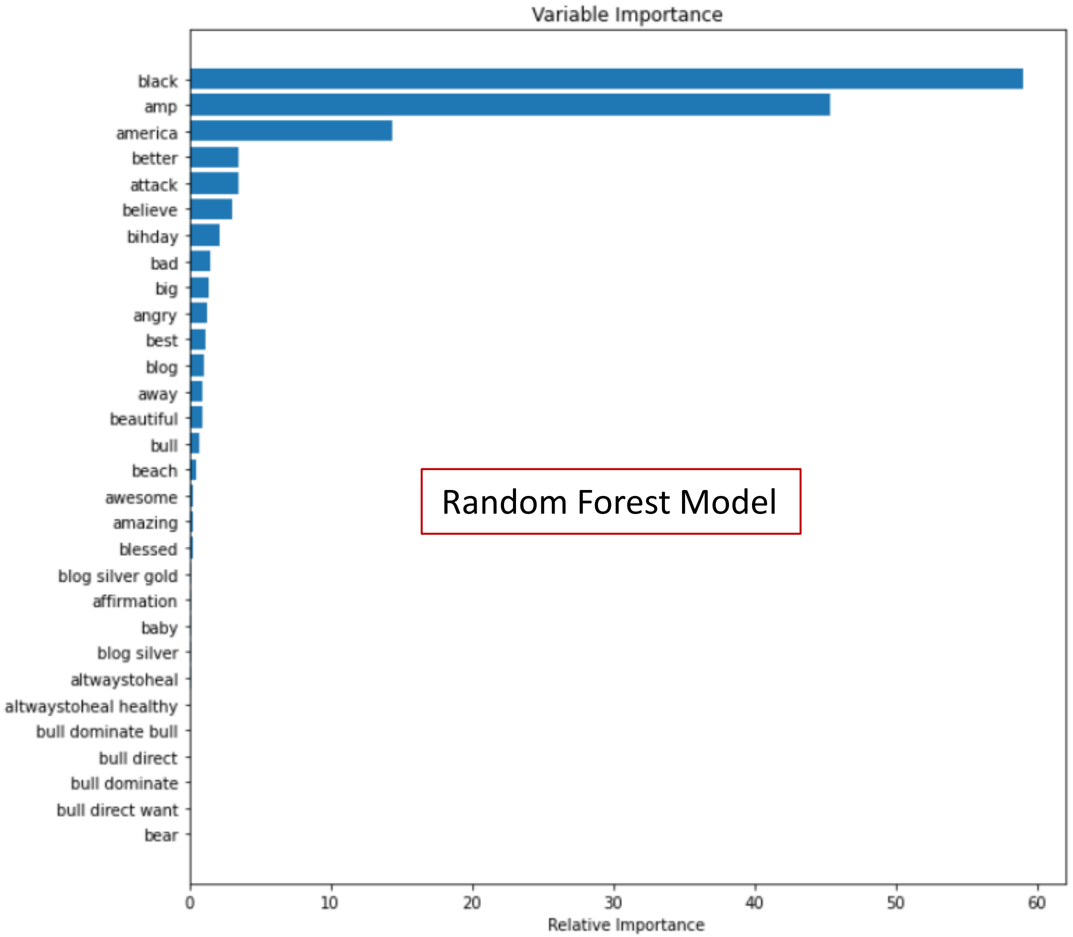
For comparison, logistic regression, random forest, gradient boost model and neural network models were used to get the best Accuracy. Hyperparameter tuning was carried out for the models and the best parameters were selected which were used to predict the test values.

Model	Accuracy (before tuning)	Accuracy (after tuning)
Logistic Regression	*0.932*	*0.932*
Random Forest	*0.933*	*0.933*
Gradient Boost Model	*0.928*	*0.933*
Neural Network	*0.932*	*0.935*

The neural network model could not be optimized across all activation function, optimizers, batch-size, epoch, learning-rate, momentum due to time constraint. Hence, a procedure has been shown using SGD optimizer. Early stopping and drop-outs were included in the model.
A gridSearchCv was used during hyperparameter tuning.
It was shown that by increasing the maximum feature size to 500; increased the accuracy of the neural network to 0.948; however, the model was not able to classify the offensive tweets properly. It seemed there was overfitting of the data with too many features.

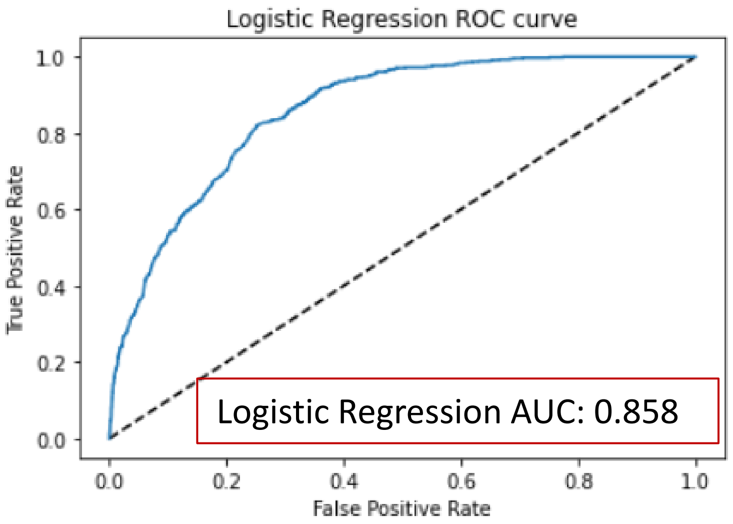
The feature importances were displayed for gradient boost classifier and random forest model:


Gradient Boost Model

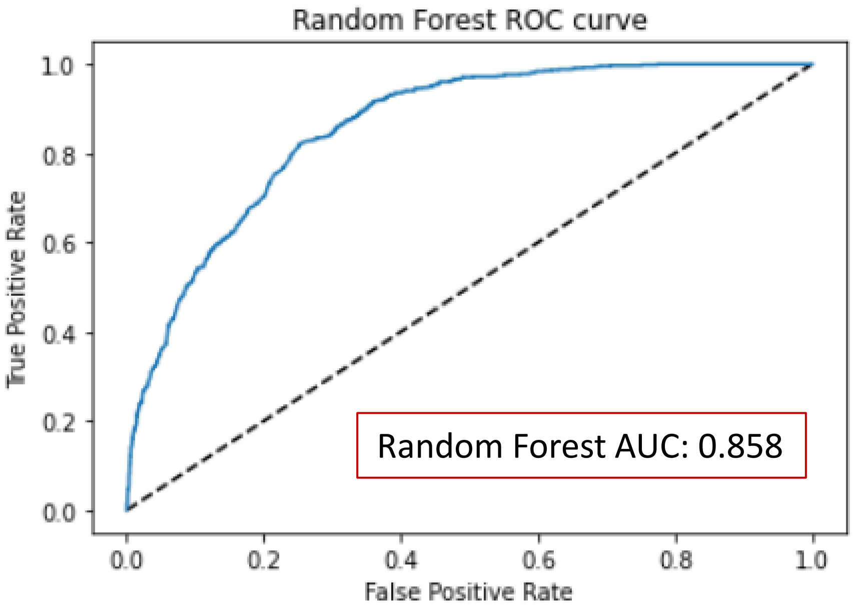

Random Forest Model

It seems like the feature importances from the random classifier model to an extent match with the gradient boost model. Both models seem to agree that features like "black", "amp", "america", "attack" are important.

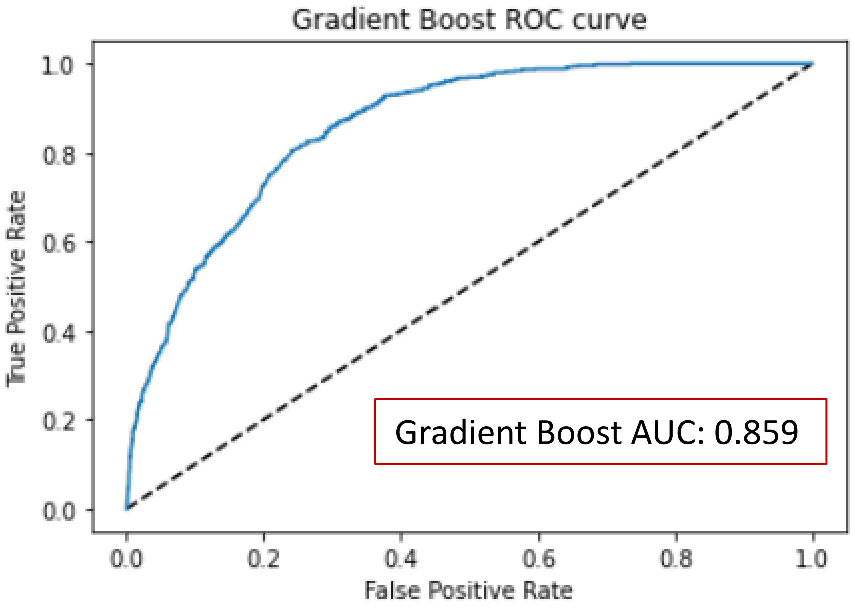
The ROC and AUC plots were calculated for all four models. As observed, the neural network model gave the highest AUC


Logistic Regression ROC curve

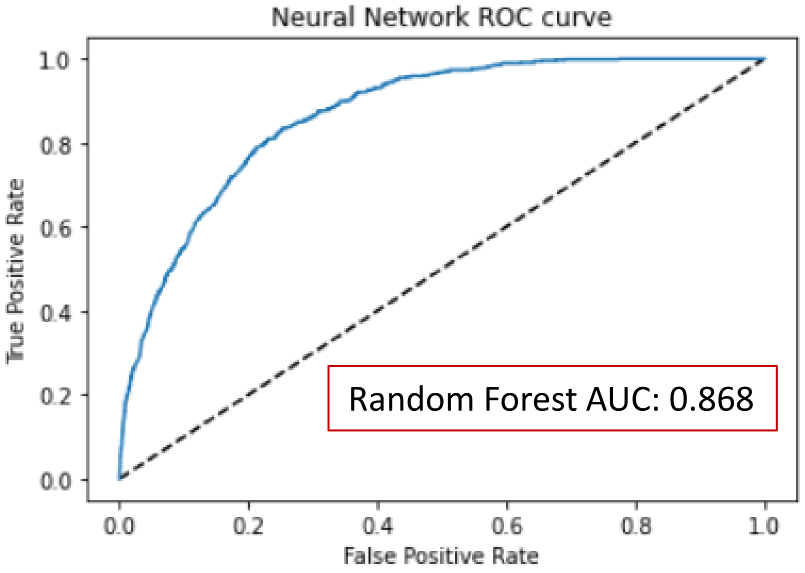
Logistic Regression AUC: 0.858


Random Forest ROC curve

Random Forest AUC: 0.858


Gradient Boost ROC curve

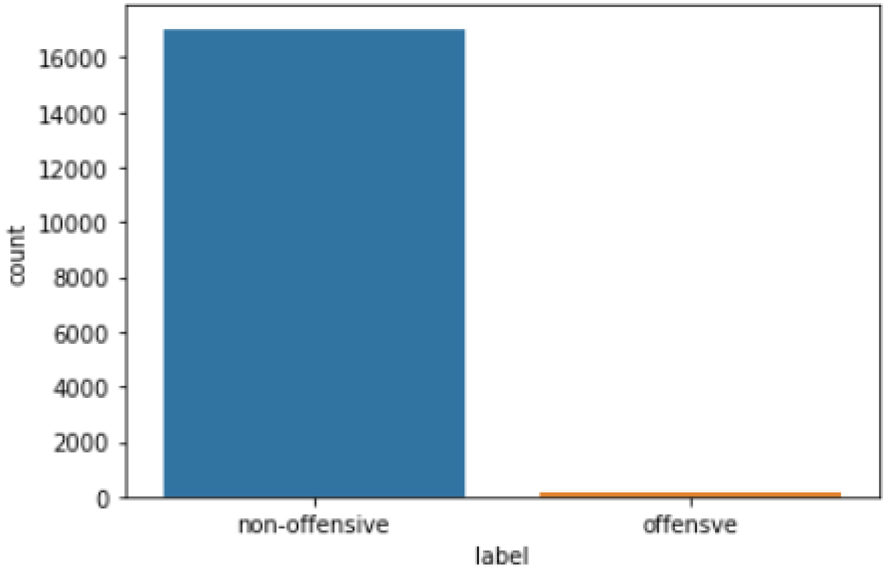
Gradient Boost AUC: 0.859


Neural Network ROC curve

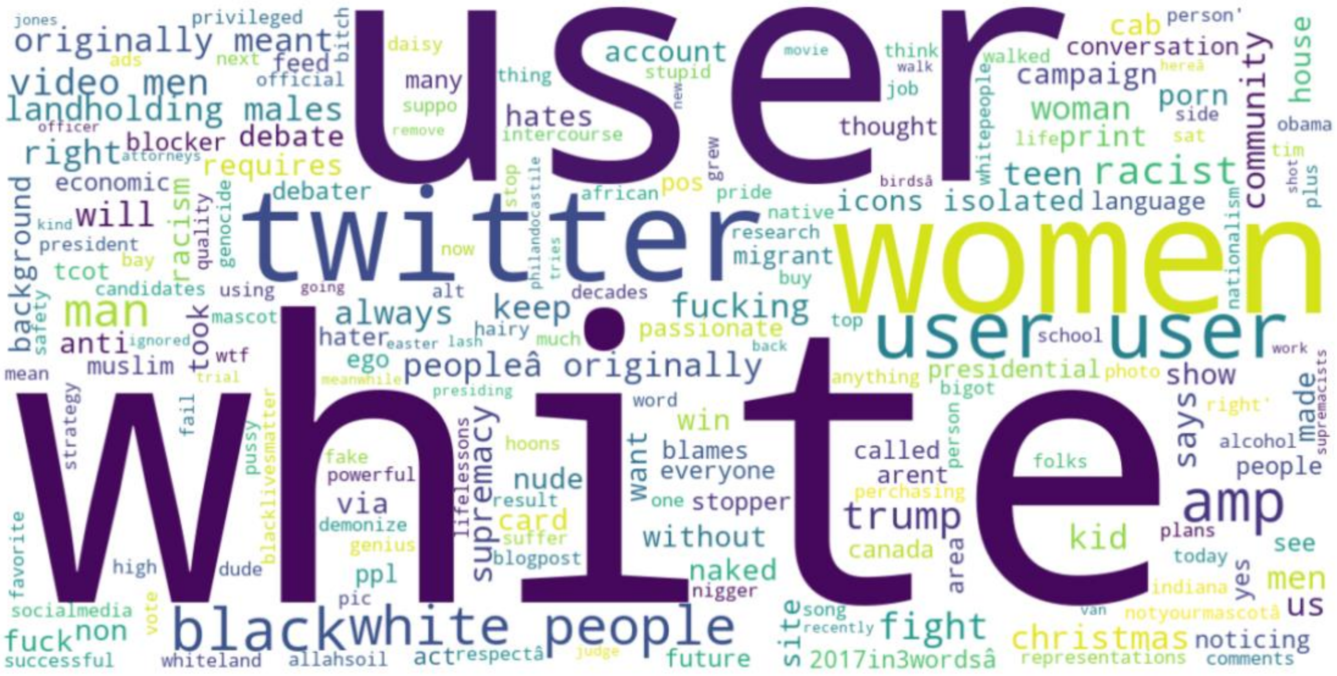
Random Forest AUC: 0.868

Part 4- Model testing and Visualization

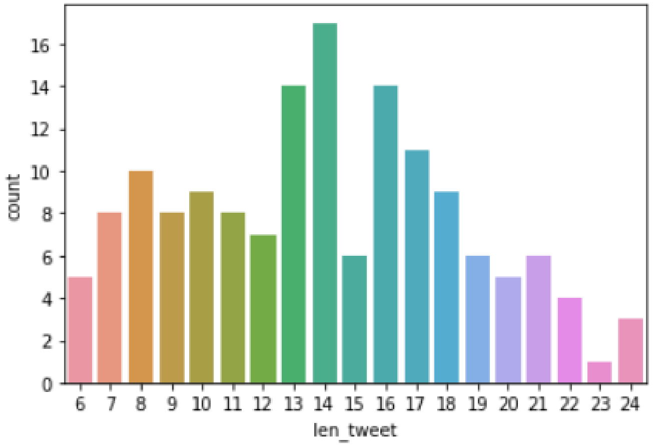
The neural network model with 3 hidden layers and each with 302 nodes were used to predict racist/hateful tweets on the test data. As seen, in the histogram, 151 tweets were marked as offensive whereas 17046 tweets were marked as non-offensive. A word cloud of the all 151 offensive tweets was plotted as well as the histogram distribution of the hateful tweets with word length was generated. The length vs tweet count showed a Gaussian distribution.



Around 0.8% of tweets were labeled as offensive on test data



A word cloud of all offensive tweets in test data



A histogram of count vs length of all offensive tweets