Hate Speech Detection on Tweets

Multiclass classification of tweets using different modelling techniques

Shika Shyam – 002194543
Deepika Balasubramaniam – 002194564

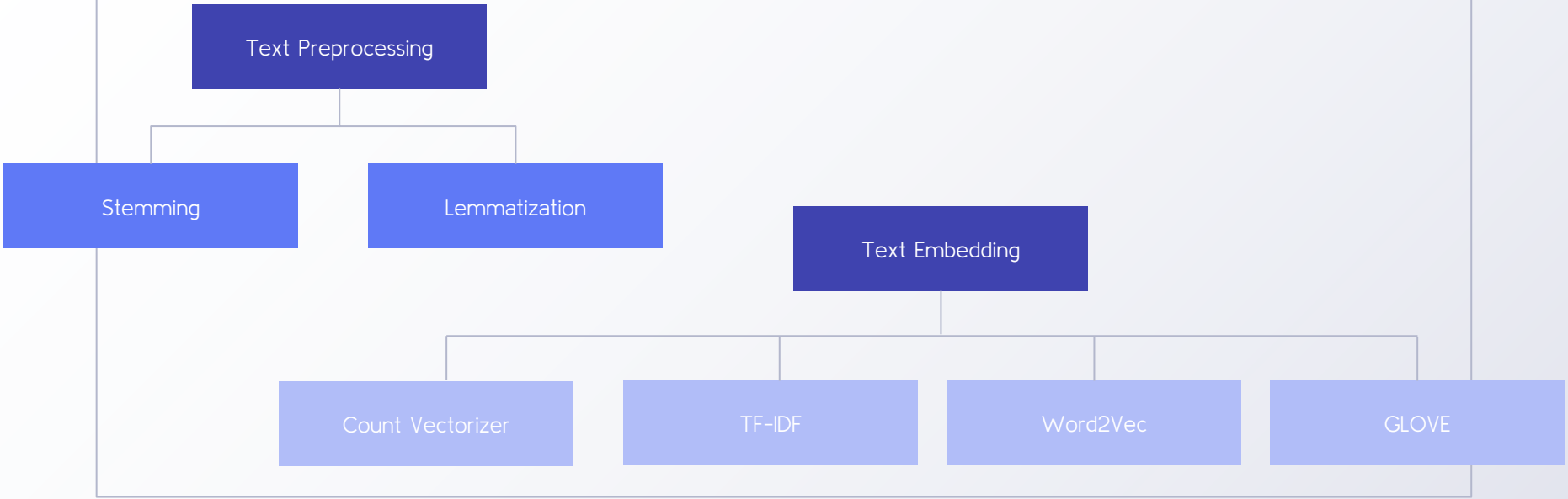
Goal

In this project our goal is to classify tweets as hate speech or related and non-hate speech. In a more specific sense, we will be classifying tweets into one of four categories - Normal, Spam, Abusive and Hateful.

How did we achieve it?

- We will be cleaning the tweets by removing special characters and non-alphanumeric characters, and then using NLP Text preprocessing techniques.
- For text preprocessing we will be using Stemming or Lemmatization, and compare them
- We will be making use of 4 different techniques to do feature engineering - i.e. converting the tweets into feature sets using embedding techniques, namely - Count Vectorizer, Term Frequency - Inverse Document Frequency (TF-IDF), Word to Vector (W2V), Global Vectors for Word Representation (GLOVE).
- We will then be training Gaussian Naive Bayes, Multinomial Naive Bayes, Support Vector Machine and Long Short-Term Memory (LSTM) models on each of these ablation settings.
- We will choose the best performing combination of preprocessing method + embedding method, and then use that setting on a hyperparameter tuned version of NB, SVM and LSTM to better our model performance even further.
- Since ultimately, this is a classification task, we will look at how each of these settings and each of these models performed with respect to three main metrics - Precision, Recall and F-1 Score.
- We will compare each of our models (with each ablation setting) 27 different settings in total and identify the best models for our project goal of detecting and classifying hate speech.

Methods we used



Packages Used

Sklearn
Feature Extraction
Metrics
Model Selection
Naïve Bayes
SVM

Gensim
KeyedVectors

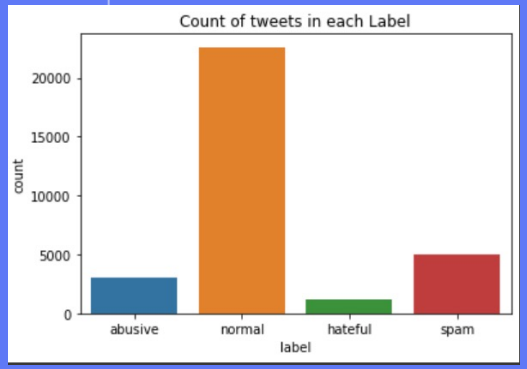
TensorFlow
Keras
Tokenizer
Embedding, Dense, Dropout
Preprocessing
Sequential

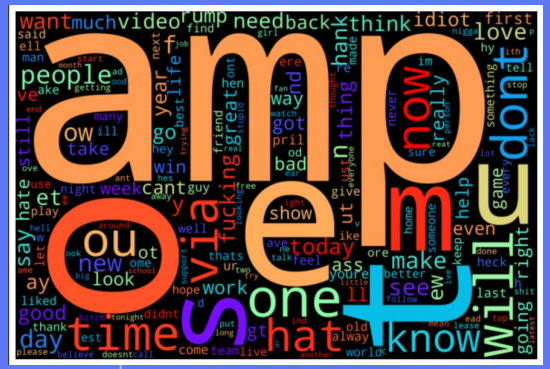
Plotting Libraries
Seaborn
Matplotlib
Plotly
Wordcloud
PIL

NLTK
Tokenize
Stem
StopWords

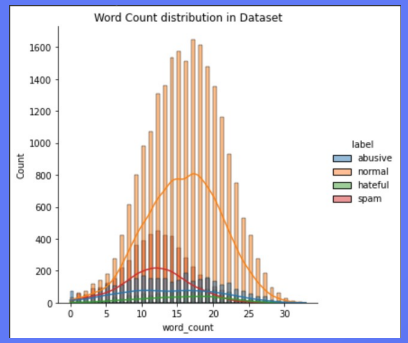
General Libraries
Pandas
Numpy
Pickle
String
re

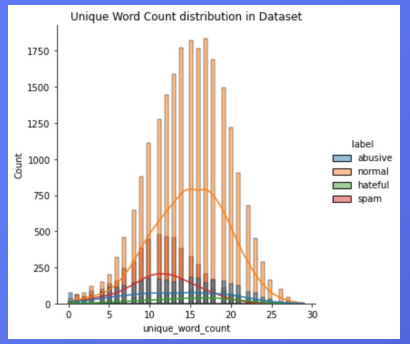
Exploratory Data Analysis

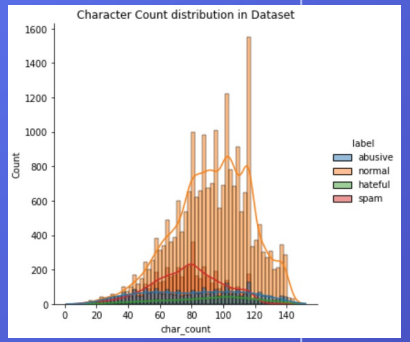


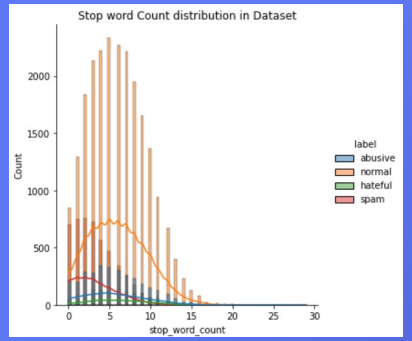


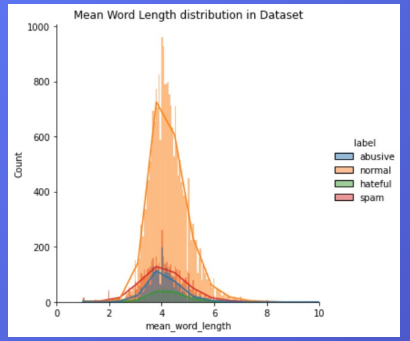
Meta features







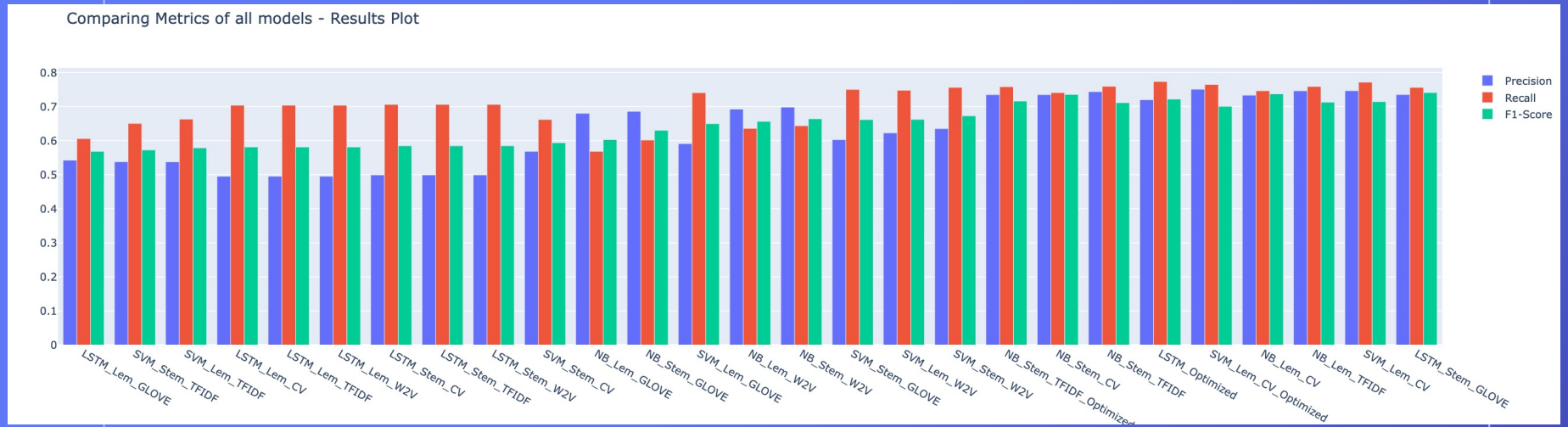




Comparing all models and ablation settings

Model	Text Preprocessing	Embedding technique	Precision	Recall	F-1 Score
Naïve Bayes	Stemming	CV	0.73	0.74	0.74
Naïve Bayes	Lemmatization	CV	0.73	0.75	0.74
Naïve Bayes	Stemming	TFIDF	0.74	0.76	0.71
Naïve Bayes	Lemmatization	TFIDF	0.75	0.76	0.71
Naïve Bayes	Stemming	W2V	0.69	0.64	0.66
Naïve Bayes	Lemmatization	W2V	0.7	0.64	0.66
Naïve Bayes	Stemming	GLOVE	0.69	0.6	0.63
Naïve Bayes	Lemmatization	GLOVE	0.68	0.57	0.6
Support Vector Machine	Lemmatization	CV	0.75	0.77	0.71
Support Vector Machine	Stemming	CV	0.57	0.66	0.59
Support Vector Machine	Lemmatization	TFIDF	0.54	0.66	0.58
Support Vector Machine	Stemming	TFIDF	0.54	0.65	0.57
Support Vector Machine	Lemmatization	W2V	0.62	0.75	0.66
Support Vector Machine	Stemming	W2V	0.64	0.76	0.67
Support Vector Machine	Lemmatization	GLOVE	0.59	0.74	0.65
Support Vector Machine	Stemming	GLOVE	0.6	0.75	0.66
LSTM	Lemmatization	CV	0.5	0.7	0.58
LSTM	Stemming	CV	0.5	0.71	0.58
LSTM	Stemming	TFIDF	0.5	0.71	0.58
LSTM	Lemmatization	TFIDF	0.5	0.7	0.58
LSTM	Stemming	GLOVE	0.73	0.76	0.74
LSTM	Lemmatization	GLOVE	0.54	0.61	0.57
LSTM	Stemming	W2V	0.5	0.71	0.58
LSTM	Lemmatization	W2V	0.5	0.7	0.58
Naïve Bayes Optimized	Stemming	TFIDF	0.73	0.76	0.72
LSTM Optimized	Lemmatization	W2V	0.72	0.77	0.72
SVM Optimized	Lemmatization	CV	0.75	0.77	0.7

Comparing all models and ablation settings

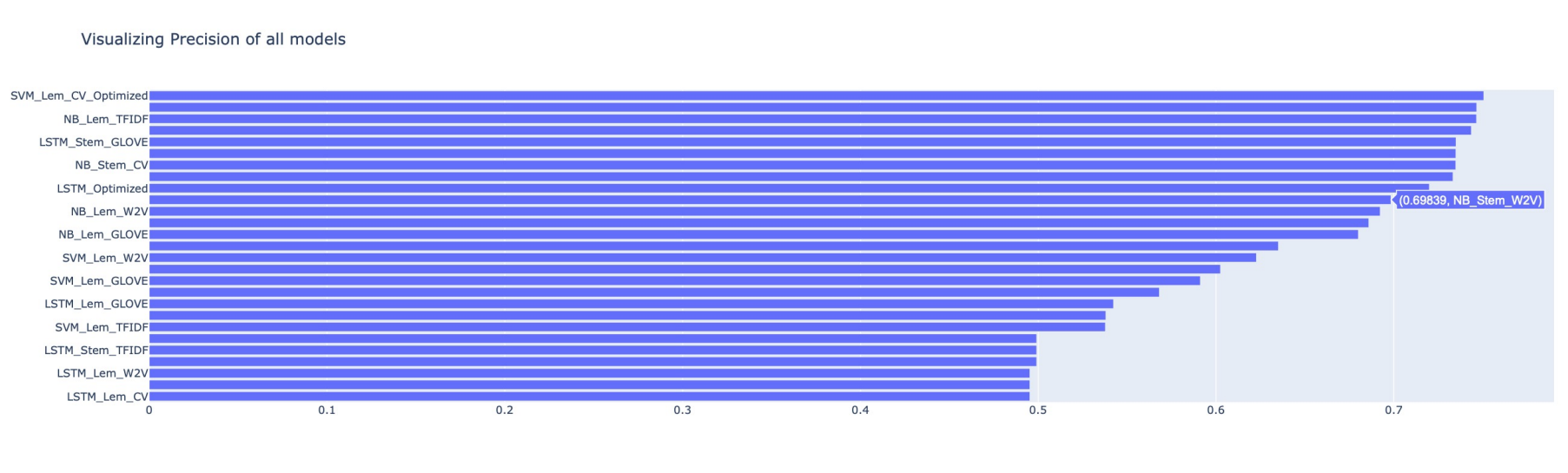


Evaluating Model Performance

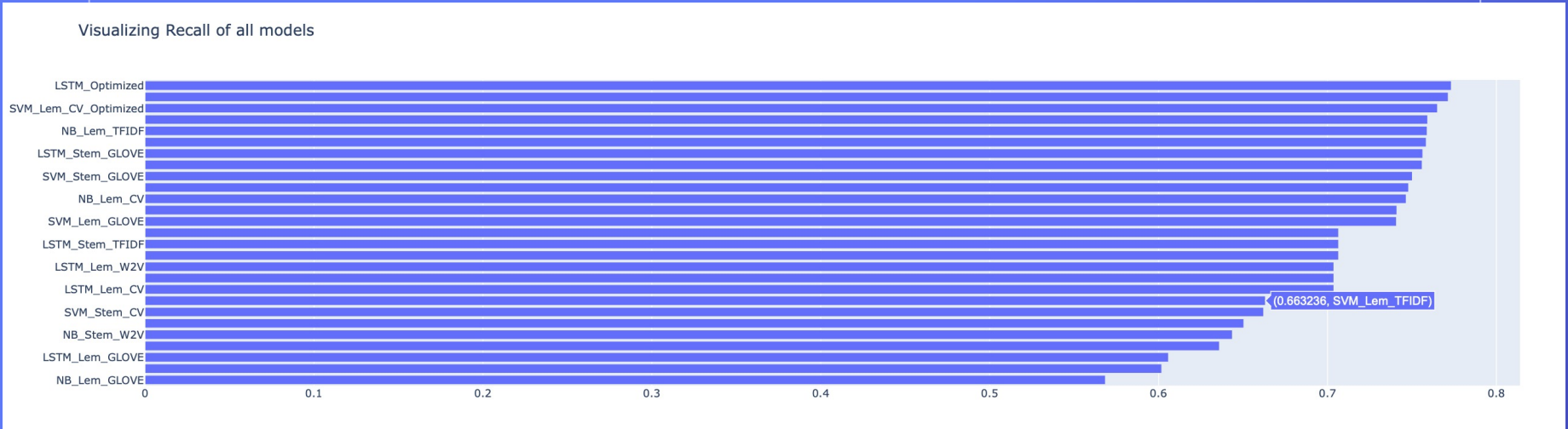
We have used three metrics to determine our best model

1. Precision
2. Recall
3. F-1 Score

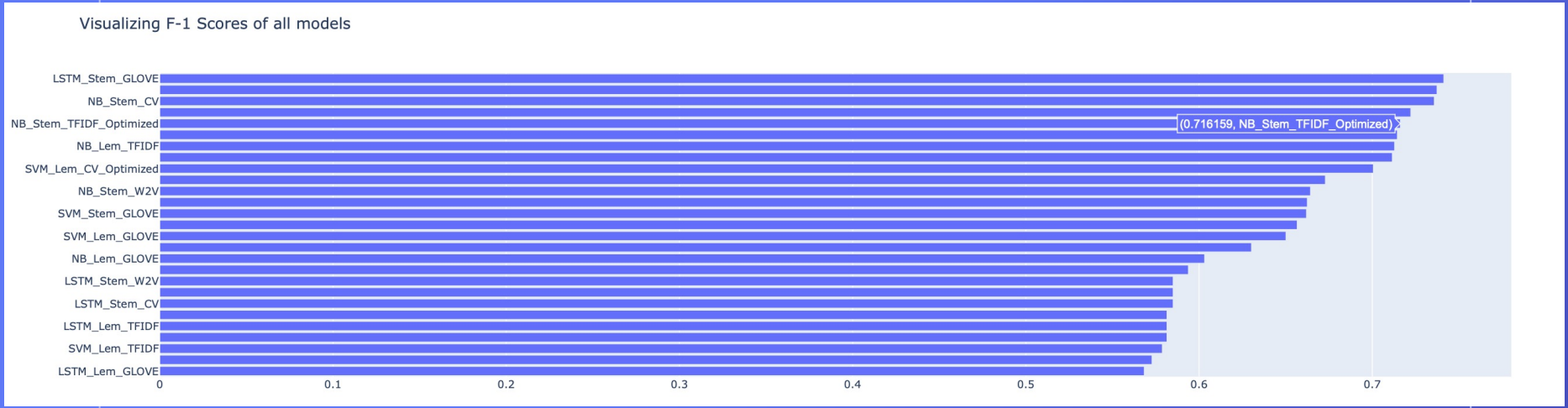
Plotting Precision



Plotting Recall



Plotting F-1 Score



Conclusion

In this project, we have successfully done the following:

- Explored the features and meta-features of a twitter tweets dataset that is classified into one of four classes - Normal, Spam, Abusive and Hateful
- We found patterns in the meta-features which lead us to believe that abusive and hateful tweets generally contain lesser words and rarely uses up the full 140 character limit of tweets.
- We then used two different preprocessing methods, Stemming and Lemmatization in combination with four different embedding methods - Count Vectorizer, TF-IDF, GLOVE and Word2Vec
- We tried all 8 combinations of the above with 3 different model types - Naive Bayes, SVM and LSTM.
- We compared the performance of the models using 3 major metrics used for any classification task - Precision, Recall and F1 score.
- We then picked the best performing ablation setting for each of the models, and used GridSearchCV for hyperparameter tuning Naive Bayes model, RandomSearchCV for hyperparameter tuning the SVM model and finally, Keras tuner for hyperparameter tuning the LSTM.
- Comparing all the metrics of all the optimized and default models against each ablation setting (27 in total), we saw that the Optimized SVM model with Lemmatization and Count Vectorization stood out as the best performing model.
- Since the task at hand is to classify tweets as abusive, hateful, normal or spam, we prefer a model that has high recall, and can tolerate low precision and F1 score. The reason being, we would rather have our model correctly classify an abusive tweet as abusive, and incorrectly classify a normal tweet as abusive THAN miss classifying an abusive tweet altogether.

Thanks!