Hate Speech Detection in Social Media

Omkar Vaidya
ov5232@rit.edu

ABSTRACT

In today's scenario with the increase in the use of social media platforms, people from different backgrounds share their opinions on different aspects of life every day on social media. This has resulted in conflicts among people and the use of hate speech which eventually becomes a very serious problem on social media. Detecting hate speech manually from social media posts is very tedious and nearly impossible given the large scale of different social media platforms. Therefore there needs to be an efficient hate-speech classifier. The key challenge in automatic hate-speech detection for a given social media post is the misclassification of a post as hate speech due to the presence of offensive language. The project implements a multi-class classifier to identify differences between hate speech and offensive language and efficiently classify a social media post. The dataset used for the project focuses on Twitter data and aims to work on detecting hate speech in tweets posted by different users. The predictions are made into three categories viz. hate speech, offensive language, and neither. The analysis of the predictions from different classifier models shows that the implemented models can successfully distinguish hate speech and offensive language in social media posts thereby reducing the misclassification of posts as hate speech.

1. INTRODUCTION

With an increase in the use of the internet and different social media platforms over the last few decades, social media platforms have become a popular medium for online communication. People from all over the world express their views and opinions about a variety of subjects in the form of social media posts on platforms such as Facebook, Twitter, etc. People from different backgrounds interact with each other online in the form of text such as messages and comments. But sometimes communications between people turn abusive due to differences in opinions. Thus, different social media activities face the problem of hate speech and can take various forms. Defining hate speech is difficult as there's no one such definition that would incorporate all aspects of it. In a generalized context, hate speech can be defined as the language used towards an individual or a group to express hate or is insulting or humiliating towards that individual or group. Detection of hateful speech is important for analyzing public sentiments towards an individual or group in order to inhibit associated wrongful activities.

Using hate speech on social media platforms is a pressing issue as it might lead to harassment, bullying, trolling, hurt sentiments of people, etc., and therefore should be detected in order to curb it on social media platforms. Therefore this project aims to develop a hate speech classifier that tries to classify the speech as hate speech, offensive, or neither. The project aims to distinguish offensive language and hate speech as previous works have led to misclassification of social media posts as hate speech due to the presence of offensive language. The aim would be to efficiently classify hate speech by identifying subtle linguistic differences between offensive words and hate speech and also leverage syntactic features to better identify hate speech. This process would involve doing preprocessing tasks, studying and using appropriate feature selection methods (TF-IDF, POS Tags-unigrams, bigrams, and trigrams), and applying machine learning models such as L1 Regularizer Logistic Regression, Naive Bayes, Random Forests, Linear SVM's to perform the classification task [3]. The project would be useful in identifying tweets with hate speech in order to filter them and prevent hate speech tweet recommendations to other users and flagging posts that spread hate speech on social media platforms. The contributions of the project are as follows:

- Efficiently classify hate speech by identifying subtle linguistic differences between offensive words and hate speech and leveraging syntactic features of tweets to better identify hate speech.
- Reduce the misclassification of offensive language as hate speech.
- Classifier models that classify social media posts (tweets) into three classes - hate speech, offensive language, and neither.
- Improve upon human annotators in classifying tweets in the right category.

2. DATASET

The process of creating the dataset, as per the dataset source [5] consists of collecting a hate speech lexicon containing different words and phrases identified by people on the

Copyright 2020 ACM X-XXXXX-XX-X/XX/XX ...$10.00.

internet as hate speech which is compiled by Hatebase.org. The dataset consists of around 25k tweets chosen randomly from a large corpus of tweets which consist of terms from the lexicon and manually coded by the Crowdflower users. The tweets are labeled into three categories as hate speech, offensive but not hate speech, and neither hate speech nor offensive. The columns of the dataset are - count, hate_speech, offensive_language, neither, class and tweet. The column 'count' consists of values that represent the number of Crowdflower users who coded each tweet. The minimum number of users who coded is three for all tweets with some tweets having up to a maximum of nine people who coded the tweet in either of the three classes. The columns 'hate_speech', 'offensive_language', and 'neither' contain values of the number of Crowdflower users who judged the tweet to be in their respective categories. (hate_speech or offensive_language or neither). The column 'class' consists of the majority class label based on the majority class category for a specific tweet coded by the Crowdflower users. The column 'class' consists of a set of values - 0,1,2 belonging to classes - hate speech, offensive, and neither respectively. The column 'tweet' consists of the actual tweet as posted by different users on Twitter. A quick look at the dataset can be seen in Figure [1].

	count	hate_speech	offensive_language	neither	class	tweet
0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't...
1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4	6	0	6	0	1	!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

Figure 1: Dataset

3. DATA PREPROCESSING

The different tweets that comprise our dataset, may contain fruitless, redundant, and inconsistent data which is not useful in determining hate speech. Therefore, passing the data the way it is would degrade the performance of our classifier model. Therefore, it is very essential to perform some preprocessing before we pass our data to the classifier model. The data being in the form of text, we perform some text preprocessing as the first step in NLP in the process of building a model [4]. The tweets in the dataset consist of Twitter handles and mentions(@...), some URL references(https...), and extra spaces. Therefore we remove all the Twitter handles, URLs and remove extra spaces and keep just one space between words in the tweet. Punctuations, numbers, and special characters are also removed. Tweets are converted to lower case, stemmed, and tokenized. The tweets are filtered from stopwords as they do not add much meaning to sentences. The preprocessing steps are implemented by using regular expression(regex) and Python's nltk [1] library. The preprocessing steps can be demonstrated with an example tweet as follows:

Original Sentence: "!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. amp; as a man you should always take the trash out..."

Removing Twitter handles, URLS, spaces: !!! RT : As a woman you shouldn't complain about cleaning up your house. amp; as a man you should always take the trash out...

Removing special characters, lowercase, stopword removal and tokenization: ['woman', 'complain', 'cleaning', 'house', 'amp', 'man', 'always', 'take', 'trash']

4. FEATURE GENERATION

One of the most important steps that will have an impact on the performance of the machine learning model is the features that we provide as an input to the model. Feature generation is important in order to convert the raw text (tweets) into a matrix of features for the machine learning model. For the task of hate speech detection, after the preprocessing steps are done we create unigram, bigram, and trigram features each weighted by its TF-IDF. The TF-IDF vectorizer from Python's scikit-learn [2] library is used to generate the n-grams for each tweets. Additionally, we also construct Part-of-Speech(POS) tag unigrams, bigrams, and trigrams weighted by its TF-IDF to capture information about the syntactic structure. A quick look at the combined feature generated matrix (tfidf+pos tags) for each tweet can be seen in Figure [2].

	0	1	2	3	4	5	6	7	8	9	...	7583	7584	7585	7586	7587	7588	7589	7590	7591	7592
24778	2.583261	3.907834	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.958858	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24779	3.874892	3.907834	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24780	1.291631	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24781	1.291631	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24782	2.583261	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 2: Feature Matrix for Tweets

5. MODEL

In order to classify the tweets into one of the three categories (hate, offensive, neither) we implement different machine learning models such as - L1 Logistic Regression, Random Forest, Naive Bayes, and Support Vector Machine(SVM). After the features are generated for each tweets, these features act as the input to the classifier models. The data is split into training and testing datasets in a ratio of 80%:20%. The training data consists of around 20k records and the testing data consists of around 5k records. The models are trained on the training dataset and using the test dataset the prediction is made to predict the class label for each tweet of test data.

6. EVALUATION

Four classifier models are implemented viz. SVM, L1-Logistic Regression, Random Forest and Naive Bayes to efficiently perform the task of hate speech detection. The different performance evaluation metrics such as accuracy, precision score, recall score, f1-score, confusion matrix and roc-auc curve are used to evaluate the performance of the models. The evaluation results for all the models can be seen in the figures [3, 4, 5, 6, 7, 8, 9]. Table 1 shows the different performance evaluation measures such as - Accuracy, Precision, Recall, and F1-score for each implemented model.

7. RESULTS AND DISCUSSION

For the L1 Logistic Regression model, we achieve the highest accuracy of about 82% and from the confusion matrix we observe that about 189 tweets that were previously marked as hate speech are classified as offensive. The Random Forest

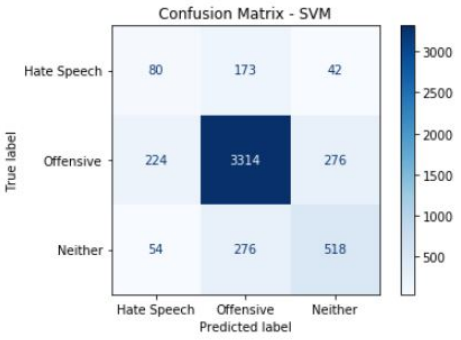

Figure 3: SVM Confusion Matrix

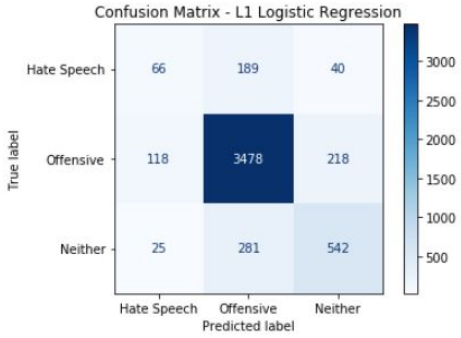

Figure 4: L1-Logistic Regression Confusion Matrix

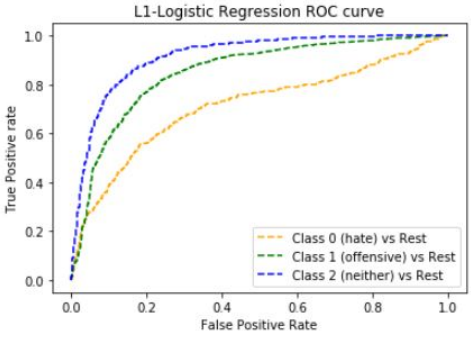

Figure 5: L1-Logistic Regression ROC Curve

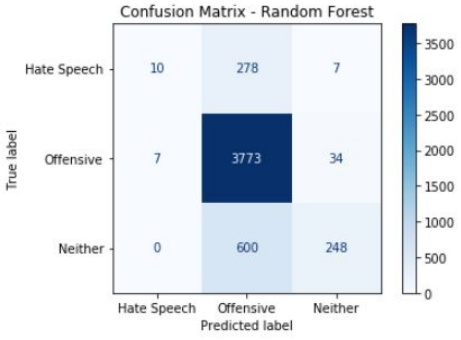

Figure 6: Random Forest Confusion Matrix

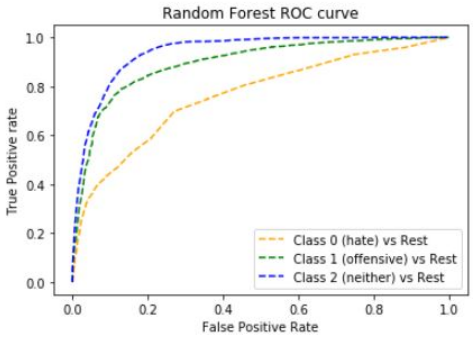

Figure 7: Random Forest ROC Curve

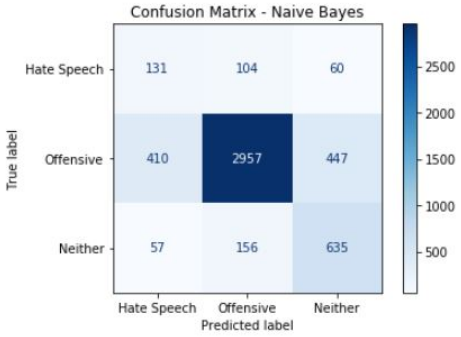

Figure 8: Naive Bayes Confusion Matrix

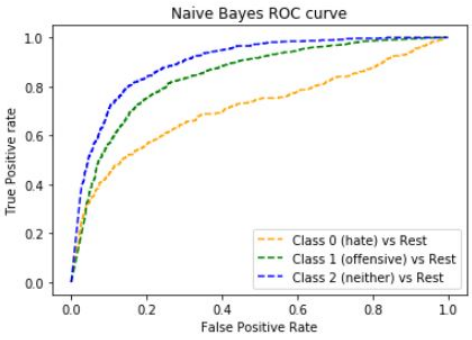

Figure 9: Naive Bayes ROC Curve

classifier achieves an accuracy of 81% and about 278 tweets that were previously marked as hate speech are classified as offensive. For the SVM classifier model, we get an accuracy of around 79% and we observe that about 173 tweets that were previously marked as hate speech are classified as offensive. The Naive Bayes classifier obtains the lowest accuracy of about 75% and we see that about 104 tweets that were previously marked as hate speech are classified as offensive.

From table 1 we observe that the precision, recall, and F1-scores are pretty consistent with the accuracy of the models which shows that the models output relevant information. Having a look at the Receiver Operating Characteristic(ROC) curve plots of the implemented models, we observe that the Area Under Curve(AUC) is good enough for each class and thus can be said that the implemented mod-

	LR	SVM	RF	NB
Accuracy	82.43%	78.91%	81.32%	75.11%
Precision	81.25%	79.69%	80.59%	81.53%
Recall	82.33%	78.91%	81.31%	75.10%
F1 Score	81.76%	79.28%	76.43%	77.38%

Table 1: Results of Classification Models

els do good in classifying the tweets. Therefore based on the classification results of the models it can be said that the implemented approach does efficiently classify posts as hate speech and reduces the misclassification of posts as hate speech due to the presence of offensive language.

8. CONCLUSION AND FUTURE WORK

If we happen to combine offensive language and hate speech then we would erroneously consider most people to be hate speakers. In the context of social media, the same confusion can lead to the assumption of certain social media posts as hate speech when it doesn't actually promote it. Given the serious implications of hate speech (moral and legal), it is important that we efficiently and accurately identify the offensive language and hate speech. The implemented approach in project - to identify subtle linguistic differences using unigram, bigram, and trigram features weighted by TF-IDF for tweets and also combining it with POS tags to generate features by understanding the syntactic structure of the tweet is seen to efficiently classify hate speech. The project results show efficient classification of tweets as hate speech, offensive language and neither. In the area of developing an efficient automatic hate speech classifier for social media, some future work can be to understand the social context in social media posts to better identify hate speech. Additionally, research can be done to understand the individual characteristics of people promoting hate speech on social media, identify social situations of hate speech, and develop a model that would consider the aforementioned factors to better identify hate speech.

9. REFERENCES

[1] Natural language toolkit. `https://www.nltk.org/`. [December 7, 2020].

[2] Scikit-learn. `https://scikit-learn.org/`. [December 7, 2020].

[3] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. Deep learning for hate speech detection in tweets. *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*, 2017.

[4] G. Koushik, K. Rajeswari, and S. K. Muthusamy. Automated hate speech detection on twitter. *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 2019.

[5] T-Davidson. t-davidson/hate-speech-and-offensive-language.