ML Project Final Report

Topic - Hate Speech Detection

Abstract

Being regular users of social media platforms, we have noticed extensive use of racist, sexist, homophobic, or generally offensive speech on social media has increased exponentially since the no. of users for the same has increased in the past few years. So, we are creating a Hate Speech Detection model to protect new users of social media from exposing to the negativity caused due to Hate Speeches.

1. Introduction

The use of social media has increased in recent years, and this has also led to an increase in the number of users, which made the use of offensive speeches more common; thus, protecting users from Hate Speech detector is necessary. There are several Hate Speech detection models in the world right now, but they aren't providing good/required accuracy; hence, research is going on in this field to create an optimal model.

2. Literature Survey

We found and discussed two research papers related to our work. In [1], the author used the Hate Detection model to detect the offensive words spoken in a video available on YouTube, and they successfully made a different model using Multinominal Naive Bayes, Random-Forest (1024 trees), Linear SVM (Support Vector Machine), and RNN (Recurrent Neural Network). After analysis, the final result was that Random Forest Classifier provided better accuracy than any other model, with 96% accuracy. In [2] research paper focuses on TWEETEVAL, a benchmark for tweet classification. The used NLP research techniques like sentiment analysis, emotion recognition, offensive language detection, irony detection, etc., evaluation using a transformer-based model for training is used. The final result they got that RobertaBase (RoB-Bs) performs best on all the tasks, even outperforming the model -trained on Twitter-only data (RoB-Tw).

3. Dataset

3.1 **Dataset Details:**

- Count column: number of CrowdFlower users who coded each tweet (min is 3).
- hate_speech: number of CF users who judged the tweet to be hate speech.
- offensive_language: number of CF users who judged the tweet to be offensive.
- neither: number of CF users who judged the tweet to be neither offensive nor non-offensive.
- class: a class label for the type of tweet as per voting by CF users.
    - 0 - hate speech, 1 - offensive language, 2 - neither.
- Tweet column: contains the tweets

3.2 **Data Preprocessing:**

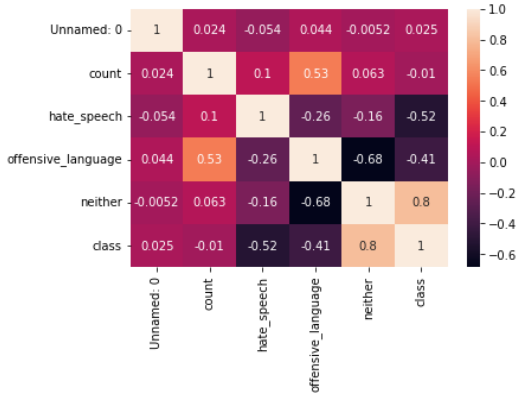
Removing unnecessary things: Hashtags, links, unwanted symbols -, <>, ? /, ';[]

Checking which words occurred most frequently in respective: hate speech, offensive speech or neither of them. (Using TfidfVectorizer)

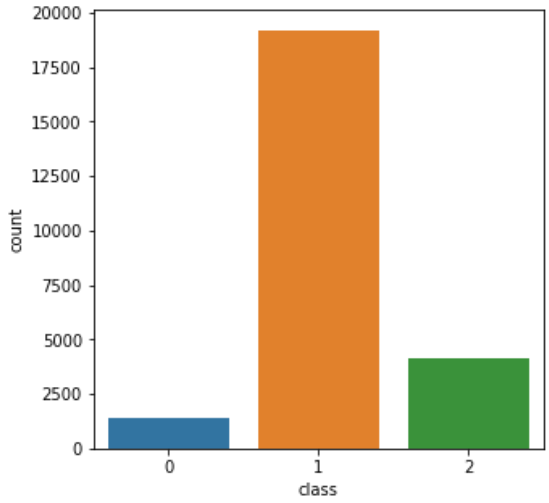
Oversampling – due to imbalance in dataset which results in uneven prediction and result by the model we are training and testing.

3.3 **Data Visualization:**

Heat-Map:



Count-Plot:

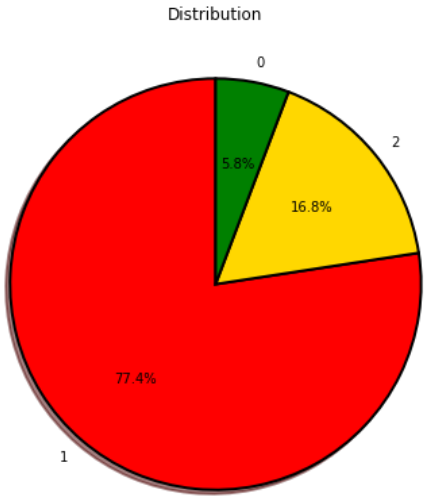


Normal tweets word-cloud:

Hate speech voted tweets word-cloud:

Offensive tweets word-cloud:

Distribution of type of tweets over data (refer class column and count-plot):



4. Methodology and Model details

TfidfVectorizer- from the sk-learn library used this feature extraction method to get required features as using the whole sentences of Tweet to predict through models isn't possible. For all three sections: hate speech, offensive, neither, and most frequently used words were stored using it and then used for feature selection.

From the sk-learn train-test split, data was split into 20% train and 80% test sets.

Oversampling was done as the data was quite imbalanced, and the performance of models was uneven and poor, hence to improve performance Oversampling was done (random oversampling).

4.1 Models Trained on above dataset:

- Random forest classifier with no. of estimators as 400.
- Default Linear regression model
- Decision tree classifier with Gini index
- SVM (Support vector Machine)

5. Result and Analysis

F1 score and accuracy without oversampling:

Logistic Regression (without oversampling) : 0.8892475287472261

	precision	recall	f1-score	support
0	0.51	0.15	0.24	320
1	0.91	0.96	0.93	3802
2	0.85	0.83	0.84	835
accuracy			0.89	4957
macro avg	0.75	0.65	0.67	4957
weighted avg	0.87	0.89	0.87	4957

Decision Tree Accuracy(without oversampling) : 0.8829937462174703

	precision	recall	f1-score	support
0	0.36	0.24	0.29	320
1	0.92	0.94	0.93	3802
2	0.84	0.86	0.85	835
accuracy			0.88	4957
macro avg	0.71	0.68	0.69	4957
weighted avg	0.87	0.88	0.88	4957

Random Forest classifier (without oversampling) : 0.8829937462174703

	precision	recall	f1-score	support
0	0.52	0.12	0.19	320
1	0.89	0.97	0.93	3802
2	0.86	0.78	0.82	835
accuracy			0.88	4957
macro avg	0.76	0.62	0.65	4957
weighted avg	0.86	0.88	0.86	4957

SVM(without oversampling) : 0.8928787573128909

	precision	recall	f1-score	support
0	0.53	0.25	0.34	320
1	0.92	0.96	0.94	3802
2	0.84	0.85	0.84	835
accuracy			0.89	4957
macro avg	0.76	0.69	0.71	4957
weighted avg	0.88	0.89	0.88	4957

With oversampling:

Logistic Regression (oversampling) : 0.8967173738991193

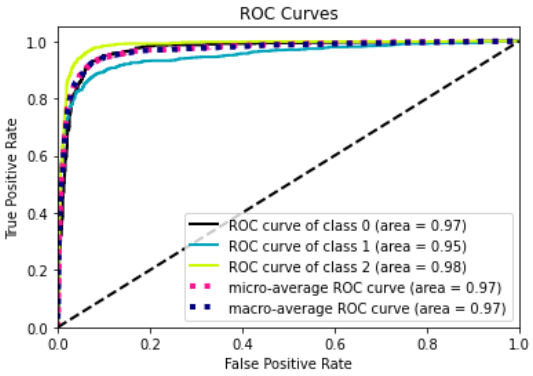
	precision	recall	f1-score	support
0	0.89	0.89	0.89	841
1	0.92	0.84	0.88	838
2	0.89	0.96	0.92	819
accuracy			0.90	2498
macro avg	0.90	0.90	0.90	2498
weighted avg	0.90	0.90	0.90	2498

```
Decision Tree Accuracy(oversampling) : 0.8987189751801441
              precision    recall  f1-score   support

           0       0.83      0.96      0.89       841
           1       0.94      0.83      0.88       838
           2       0.94      0.90      0.92       819

    accuracy                           0.90      2498
   macro avg       0.91      0.90      0.90      2498
weighted avg       0.90      0.90      0.90      2498


Random Forest classifier (oversampling) : 0.9315452361889511
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       841
           1       0.95      0.88      0.91       838
           2       0.92      0.97      0.94       819

    accuracy                           0.93      2498
   macro avg       0.93      0.93      0.93      2498
weighted avg       0.93      0.93      0.93      2498


SVM(oversampling) : 0.9151321056845476
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       841
           1       0.94      0.85      0.89       838
           2       0.92      0.95      0.93       819

    accuracy                           0.92      2498
   macro avg       0.92      0.92      0.91      2498
weighted avg       0.92      0.92      0.91      2498
```

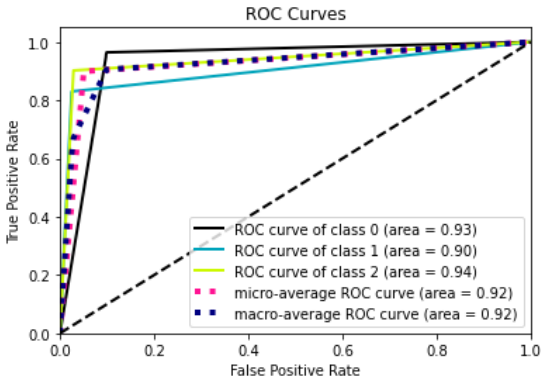
Uneven results, i.e., mainly the F1 score, were obtained without oversampling, which shows the need to make the data balanced somehow and improve the model's performance.

ROC-curve:

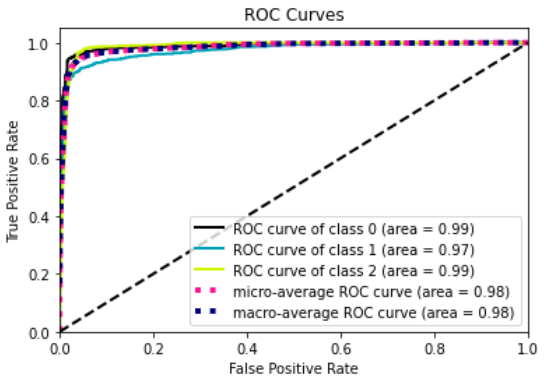
Logistic Regression:



Decision tree:



Random forest:



As it is seen from the above, accuracy score with Oversampling, the models with best Accuracy and even F1 score is SVM, followed by Random Forest Classifier and Decision Tree Classifier. Whereas, from the ROC-curve plot best one we got is Decision Tree classifier. Also, from the both classification report and ROC-curve we can analyze and say that, one of the reasons Random-Forest Classifier is giving best accuracy score is due to overfitting.
Hence, for our dataset the best models we got after analysis are SVM model and Decision Tree Classifier model.

6. Conclusion

Finally, we can say that our Decision Tree model and SVM model is doing Hate Speech Detection more accurately and is better than almost every detection model present on the internet. We used innovative use of techniques of ML, AI, etc. for the same to improve the overall efficiency of our model.

From this project we learned new and informative topics related to Machine learning like, the Multinominal Naïve Bayes, TWEETVAL, RoB-bs, Sentiment analysis, Emotion recognition model and RNN/CNN (Neural Network). We got more insights about the usage of Random Forest, Decision Tree classifier, SVM (Support vector machine), which are used in the model building and training.
Git-Hub Repo link:
https://github.com/tanay619/ML-Project_Hate-Speech-Detection

Contributions:
Tanay and Abhishek (Model Building and training, Preprocessing, Methodology)
Vijay and Rakesh (Literature Review, finding relevant data and dataset, Methodology)
Overall, all the group members worked collectively toward the Project.

References:

[1] *Ching She Wu, Unnathi Bhandary (2020) from San Jose State University San Jose, CA USA. Detection of Hate Speech in Videos Using Machine Learning*, Available at: https://american-cse.org/sites/csci2020proc/pdfs/CSCI2020-6SccvdzjqC7bKupZxFmCoA/762400a585/762400a585.pdf

[2] *Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, Luis Espinosa-Anke (2020) from Cardiff University, United Kingdom. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification*, Available at: https://arxiv.org/pdf/2010.12421.pdf

Others:

- https://osheensachdev.medium.com/finding-the-discriminative-power-of-features-to-analyse-how-different-parameters-affect-the-rating-5f405bb87cf8
- https://www.kaggle.com/code/pardhasaradhireddy/hate-speech-detection-f1-score-99
- https://www.kaggle.com/competitions/jigsaw-toxic-severity-rating/data