Foul Language Detection

Machine Learning Project

Pratik Patil
Data Science
DA-IICT
Pune, India
202118023@daiict.ac.in

Bhavya Chandrasala
Data Science
DA-IICT
Vadodara, India
202118028@daiict.ac.in

Jeffrey James
Data Science
DA-IICT
Gandhinagar, India
202118031@daiict.ac.in

Tanya Jagyasi
Data Science
DA-IICT
Ahmedabad, India
202118039@daiict.ac.in

***Abstract*—Because of the exponential expansion in internet usage by individuals of all civilizations and educational backgrounds, harmful online content has become a big concern in today's society. We conduct a comprehensive and thorough research work by referring to the existing results in this field and a proposed solution for the problem. We also identify the gaps present in the existing works and find a way to solve those problems. We propose an approach in this report for automatically classifying tweets into two categories: Foul Language and Not Foul Language. We also compared and contrasted various supervised algorithms. We apply the Term Frequency-Inverse Document Frequency (TF-IDF) values to several dataset machine learning models. After fine-tuning the model, the best results were obtained using Logistics Regression, Support Machine Classifier, Decision Tree, and Random Forest.**

***Index Terms*—Foul Language, modelling, classification, accuracy, detection**

I. Introduction

Living in the era of online social media and communication which has given us numerous stages to talk, comment and share opinions. Sadly, with the significance of online platform's benefits, it likewise opened doors for brutal conversations that can undoubtedly arrive at toxic levels. In recent years, abusive content on social media has become a significant source of worry. Due to the widespread popularity and use of social media sites such as Facebook, Twitter, and Instagram have resulted in numerous issues.

II. Data

This section contains all the aspects of data from collection, cleaning, and preprocessing.

A. Data Collection:

To build the models, we took the dataset file from Kaggle, which we use as the training and testing records. The attributes in the dataset are :

Field Name	**Description**
0	Serial Number
Count	Sum of Foul and Not Foul words
Foul_Lang	Number of Foul words in each Tweet
Not_Foul	Number of Non-Foul words in each Tweet
Class	Binary classification of Foul and Not Foul Tweets
Tweets	Complete Tweets as string

Unnamed: 0	Count	Foul_Lang	Not_Foul	Class	Tweets
0	3	0	3	1	!!! RT @mayasolovely: As a woman you shouldn't...
1	3	3	0	0	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2	3	3	0	0	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3	3	2	1	0	!!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4	6	6	0	0	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

Fig. 1. dataset

B. Data cleaning:

The dataset is in a CSV format. The attribute in the dataset named serial number is of no use, and hence we have removed the column.

	Number of tweets
Foul Tweets	20,620
Not Foul Tweets	4,163
Total Tweets	24783

C. Data-preprocessing:

Using the nltk library and regex expression, the following preprocessing operations were carried out:

1) Tokenization of words: A token is a single thing that provides the platform for a sentence or paragraph. Word Tokenization takes all the space as the separator and takes all the words as tokens.

2) Words that should be avoided *nltk.corpus.stopwords.words('English)* is used to retrieve a list of stopwords in the English dictionary, which is then removed. Words like 'is,' 'an',' the' etc. do not hold any meaning in this project.

3) To remove punctuation, with the help of regex expression we have removed the punctuations.

4) Stemming: - Stemming is removing a part of a word or reducing a comment to its stem or root. The stem does not have to be the same as the word's morphological root. It's similar to trimming a tree's branches down to the trunk. For example, words like laughing, laughs,laughed all are stem from to laugh word. With the help of PorterStemmer, we have stemmed to the root words.

5) Filtering out numeric stuff: We also removed any numeric content that didn't contribute to our project.

6) The following stage was to remove includes so they could be utilized with machine learning methods, which we did utilizing Python's sklearn library and the TF-IDF Transform. The TF-IDF is a statistical measure of a word's importance calculated by multiplying the number of times a word appears in a document by the inverse document frequency of the word. Rather than counting the frequency of words as CountVectorizer does, TF-IDF uses a method that reduces the weight (importance) of words that exist in many texts in common, finding them incapable of classifying the documents. Each document (row) and each word (column) are represented in the outcome matrix, together with the importance (weight) derived by tf-idf (values of the matrix). If a word in a document has a high tf-idf, it has most likely been in it and must be absent in other documents.

D. Data Visualization:

1) WordCloud: To gain a sense of the most popular terms used in tweets, a word cloud is constructed. This was done for both the Positive (Not Foul) and the Negative (Foul) categories of tweets. The larger and bolder the term, the more frequently it appears in a document and the more essential it is.

Wordcloud of Non Foul Language:

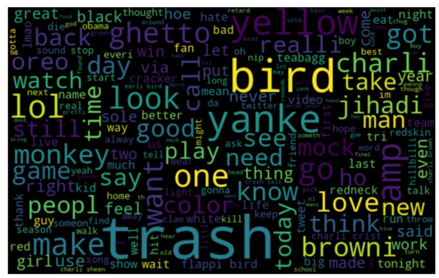

Fig. 2. Wordcloud of Not Foul Language

Wordcloud of Foul Language:

Fig. 3. Wordcloud of Foul Language

III. Model Used

A. Logistics Regression:

The most frequent binary outcome in logistic regression models is something that can take two values, such as true/false, yes/no, and in the case of this project, foul/not foul. It assures that the output probabilities add up to one and stay between zero and one, as we would predict. The logistic regression model (sometimes known as the logit model) is a linear regression variant involving the sigmoid function.

To predict a dependent data variable, a logistic regression model examines the connection between one or more existing independent variables.

$$y = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \ldots. + \beta_i x_1\right)}{1 + \exp\left(\beta_0 + \beta_1 x_1 + \ldots. + \beta_i x_1\right)} \tag{1}$$

In the above Equation,

y is the predicted output,

β_0 is the bias or intercept term

β_1 is the coefficient for the single input value (x).

Each column in the input data has an associated β coefficient that must be learned from the training data.

The *"Maximum Likelihood Estimation (MLE)"* loss function used in logistic regression is a conditional probability. If the probability is more significant than 0.5, the forecasts will be classified as class 0 (Not Foul). You will be assigned to class 1 if this does not happen (Foul).

```
Logistic Regeression:
 Report:
              precision    recall  f1-score   support

           0       0.99      0.95      0.97      3348
           1       0.95      0.99      0.97      3252

    accuracy                           0.97      6600
   macro avg       0.97      0.97      0.97      6600
weighted avg       0.97      0.97      0.97      6600

Accuracy Score: 0.9660606060606061
Precision: 0.9466076696165192
Recall: 0.9867773677736777
```

Fig. 4. Logistic Regression Results

Confusion Matrix for Logistic Regression :

In the above matrix, we can analyze the model as follows:

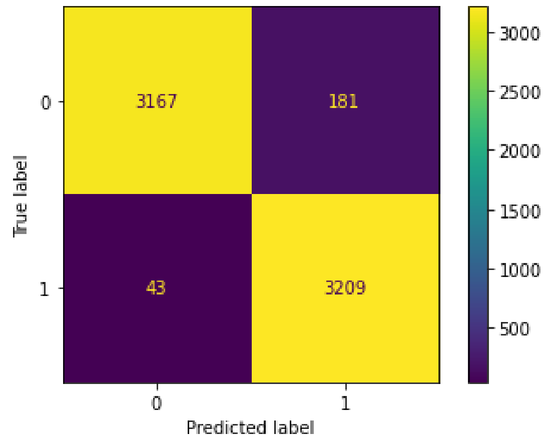

Fig. 5. Logistic Regression Confusion Matrix

True positive: The model predicted 4039 tweets from the dataset correctly.

False-positive: 83 tweets of foul were wrongly predicted as not foul by the model

False-negative: 208 tweets of Not foul were wrongly predicted as foul by the model.

True Negative: 627 tweets not foul were predicted correctly by the model.

B. Decision Tree:

The supervised learning algorithms family includes the Decision Tree algorithm. Unlike other supervised learning algorithms, the decision tree technique may be used to solve regression and classification problems. The decision to establish strategic splits significantly influences how accurate a tree is. Classification and regression trees have distinct decision criteria.

Decision trees use several ways to determine whether to split a node into two or more sub-nodes. With each generation of sub-nodes, the homogeneity of the created sub-nodes improves. Put another way, when the target variable increases, the node's purity improves. The decision tree separates the nodes into sub-nodes using all relevant factors, then selects the split that results in the most homogeneous sub-nodes. A greedy algorithm, as the name indicates, always picks the choice that appears to be the best at the moment.

Confusion Matrix for Decision tree :

In the above matrix, we can analyze the model as :

```
Decision Tree:
 Report:

              precision    recall  f1-score   support

           0       1.00      0.96      0.98      3348
           1       0.96      1.00      0.98      3252

    accuracy                           0.98      6600
   macro avg       0.98      0.98      0.98      6600
weighted avg       0.98      0.98      0.98      6600

Accuracy Score: 0.9766666666666667
Precision: 0.9561248527679623
Recall: 0.9984624846248462
```

Fig. 6. Decision Tree Results

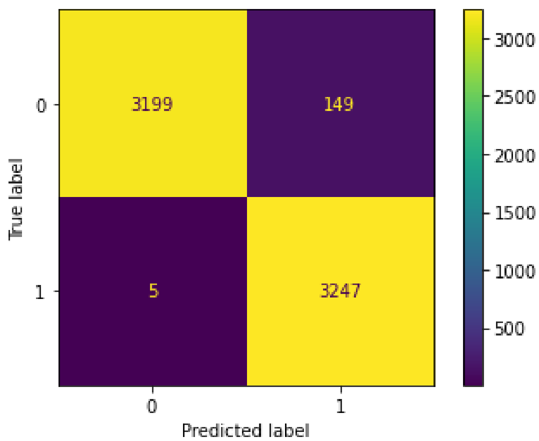

Fig. 7. Decision Tree Confusion Matrix

True positive: 3992 tweets from the dataset were predicted correctly by the model.

False-positive: 130 tweets of foul were wrongly predicted as not foul by the model

False-negative: 143 tweets of Not foul were wrongly predicted as foul by the model.

True Negative: 692 tweets not foul were predicted correctly by the model.

C. *Random Forest:*

A random forest is a machine learning approach for classifying and predicting outcomes. A method for resolving complex problems by merging many classifiers. Many decision trees make up a random forest algorithm. Bagging or bootstrap aggregation are used to train the 'forest' formed by the random forest method. Bagging Small changes to the training set can dramatically different tree structures. The data used to train decision trees is incredibly important.

This is utilized by random forest, which allows each tree to sample from the dataset at random with replacement, resulting in unique trees. This operation is known as bagging. This method determines the result based on the decision trees' predictions. The precision of the result improves as the number of trees grows. The disadvantages of a decision tree algorithm are avoided by using a random forest technique. It enhances precision and decreases dataset overfitting.

```
Random Forest:
 Report:

              precision    recall  f1-score   support

           0       1.00      0.95      0.98      3348
           1       0.95      1.00      0.98      3252

    accuracy                           0.98      6600
   macro avg       0.98      0.98      0.98      6600
weighted avg       0.98      0.98      0.98      6600

Accuracy Score: 0.9754545454545455
Precision: 0.9530791788856305
Recall: 0.9993849938499385
```

Fig. 8. Random Forest Results

Confusion Matrix for Random Forest :

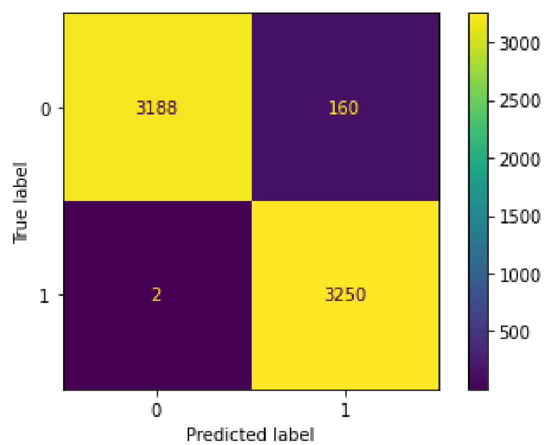

Fig. 9. Random Forest Confusion Matrix

In the above matrix, we can analyze the model as :

True positive: 3999 tweets from the dataset were predicted correctly by the model.

False-positive: 123 tweets of foul were wrongly predicted as not foul by the model

False-negative: 106 tweets of Not foul were wrongly predicted as foul by the model.

True Negative: 729 tweets not foul were predicted correctly by the model.

D. Support Vector Machine:

The Support Vector Machine, or SVM, is a Supervised Learning tool for solving classification and regression problems. However, in Machine Learning, it is mainly implemented to overcome classification problems. The SVM method's purpose is to find the optimum line or hyperplane for classifying n-dimensional space so that additional data points may be added easily in the future.

We used linear SVM in this dataset, which indicates that if a dataset can be separated into two groups using only a single straight line, it's called linearly separable data, and the classifier used was the Linear SVM classifier.

```
Support Vector Machine:
 Report:

              precision    recall  f1-score   support

           0       0.98      0.95      0.97      3348
           1       0.95      0.98      0.97      3252

    accuracy                           0.97      6600
   macro avg       0.97      0.97      0.97      6600
weighted avg       0.97      0.97      0.97      6600

Accuracy Score: 0.9675757575757575
Precision: 0.9523525908278737
Recall: 0.9833948339483395
```

Fig. 10. Support Vector Classifier Results

Confusion Matrix for Support Vector Machine :

In the above matrix, we can analyze the model as :

True positive: 4016 tweets from the dataset were predicted correctly by the model.

False-positive: 106 tweets of foul were wrongly predicted as not foul by the model

False-negative: 116 tweets of Not foul were wrongly predicted as foul by the model.

True Negative: 719 tweets not foul were predicted correctly by the model.

IV. Conclusion

Considering this work, the key messages and conclusion of this work could be summarized as follows :

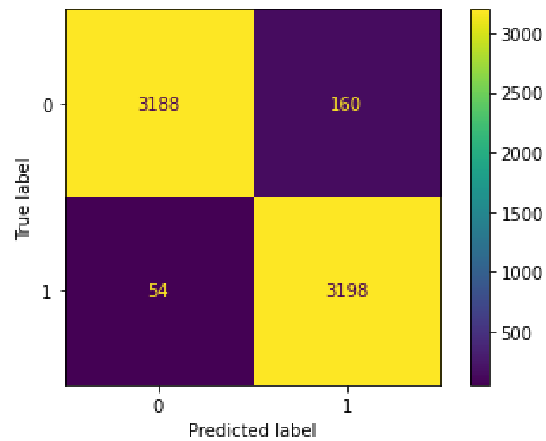

Fig. 11. SVM Confusion Matrix

a)

Model	Accuracy	Precision	Recall
Logistic Regression	0.9660	0.9466	0.9867
Decision Tree	0.9766	0.9561	0.9984
Random Forest	0.9754	0.9530	0.9993
Support Vector Machine	0.9675	0.9523	0.9833

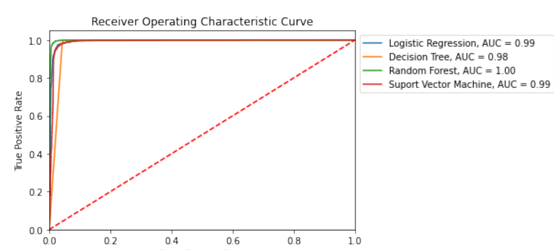

Fig. 12. Combined ROC curves for all models

In this paper,we have analysed the existing system on foul language detection using machine learning in which we use different approaches like information retrieval concepts, statistical hypothesis formulation to get more precise result. After comparing the results of the models such as Logistic Regression, Decision Tree, Random Forest, SVM on the basis of different evaluation metrics. Accuracy of 97 percent was obtained in Decision Tree and Random Forest. Hence, we can conclude that for this data, as the high-dimensional feature space data points can be categorised even when the data is not linearly separable, Decision Tree serves this purpose in the best way, so by comparing the accuracy scores we can say that Decision Tree is the best classification algorithm.

V. Bibliography:

https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset/code?resource=download

VI. References

A. S. B. Wazir et al., ”Spoken Malay Profanity Classification Using Convolutional Neural Network,” 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2021, pp. 34-38, doi: 10.1109/ICSIPA52582.2021.9576781.

https://en.wikipedia.org/wiki/Receiver$_{o}perating_{c}haracteristic$

$https://www.researchgate.net/publication/221621494_{S}upport_{V}ector_{M}achines_{T}heory_{a}nd_{A}pplications$

$Samghabadi, NiloofarSafi, etal.”Detectingnastinessinsocialmedia.”ProceedingsoftheFirstWorkshoponAbusiveLan$

$https://nerdyseal.com/wp-content/uploads/files/204/204530/the-use-of-foul-language.pdf$