Offensive Language Detection Models

Author: Maria Fernanda Castillo Castro

Abstract

To answer the research question on how to detect offensive language in text data, a decision tree classifier and a random forest to do a binary classification (OFF or NOT) of offensive language were created. These algorithms were created using the OLID datasets, which contain information from 14100 tweets divided into two categories: 13240 for the training and 860 for the test set .These datasets were originally created for the OffensEval-2019 competition.

1 Materials

- OLID dataset

2 Model Selection

The Python libraries scikit-learn and matplotlib were used to create and visualize the decision tree model. This model was based on the "Decision Trees and Random Forests: The Titanic Disaster" model by Mario Gutiérrez-Roig and Lisa Voigt, Lecturers in Data Science and Statistics at the University of Essex. However, the model was modified to analyse text data.(Roig)

2.1 Summary of 2 selected Models

A decision tree with depth three was selected due to its ease to build, test, and interpret, especially because a visualization of it can show how the key features selected affect the classification process node by node. However, decision trees tend to overfit, meaning that the model memorises the data instead of generalising. (Patil, 2021)

The use of a random forest model was used to reduce overfitting. In this project, we will compare the performance of a single decision tree with random forests of 10 and 100 trees.

2.2 Critical discussion and justification of model selection

In this exercise we will compare two different models: Decision Tree and Random Forest.

These models were chosen because it might be interesting to see exactly which words would increase the likelihood of a tweet being offensive. Also, the question was raised whether the accuracy would improve when using many trees, thus the selection of a random forest.

Even though this might offer interesting results, both models have limitations. For instance, they do not take into account the context in which each word was used and tend to become overly complicated and difficult to explain the deeper the trees are.

- In Figure 1 the pipeline of both models can be observed.

3 Design and implementation of Classifiers

Data collection and preprocessing: The dataset used for this algorithm is the Offensive Language Identification Dataset. Tokens were extracted and changed to lowercase; only alphabetic data was kept; and stopwords were removed from the tweets column.

Dataset	Total	% OFF	% NOT
Train	12313	33.23	66.76
Valid	927	33.22	66.77
Test	860	27.90	72.09

Table 1: Dataset Details

Feature selection: A count vectorizer was set up to select the features for the decision tree and random forest models; the most frequent tokens are the ones important to represent each label, "offensive" or "not offensive."(Heidenreich, 2018)

Model evaluation: To evaluate the performance, a max-depth function was used to estimate which depth is optimal for the decision tree.

In figure 2, we can observe that the optimal depth for this model is around 38 since higher depths

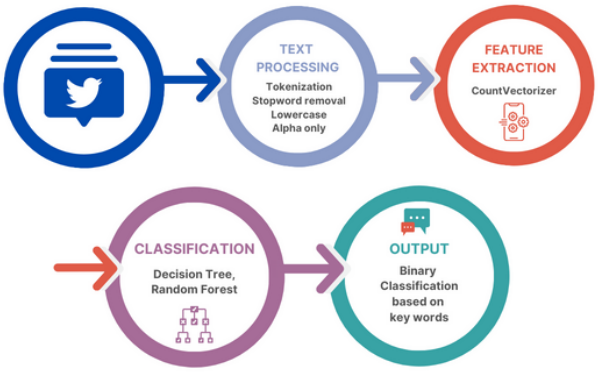

Figure 1: Diagram explaining the pipeline and decision tree and random forest models.

increase the difference between the training and test sets, meaning that our model tends to overfit the data.

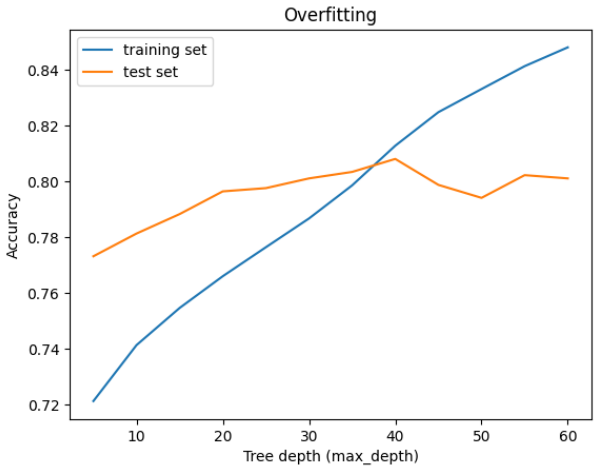

Figure 2: Accuracy measured by max-depth function

Comparison with other models:

After measuring the model performance of both models showed in Table 2, it was found that a random forest with 100 trees performs significantly better.

Model	F1 Score
Model 1	0.566
Model 2	0.734

Table 2: Model Performance

In figure 3, we can observe that the performance of the decision tree model with depth three was compared with random forests of 10 and 100 trees of depth ranging from 5 to 60.

Mean accuracy peaks at different tree depths depending on the number of trees in the random forest. A decision tree peaks around the depth of 20, a random forest with 10 trees peaks around the depth of 60, and a random forest with 100 trees peaks at a depth of 50.

Single-decision trees tend to perform better with low depths, while random forests do better with high depths.

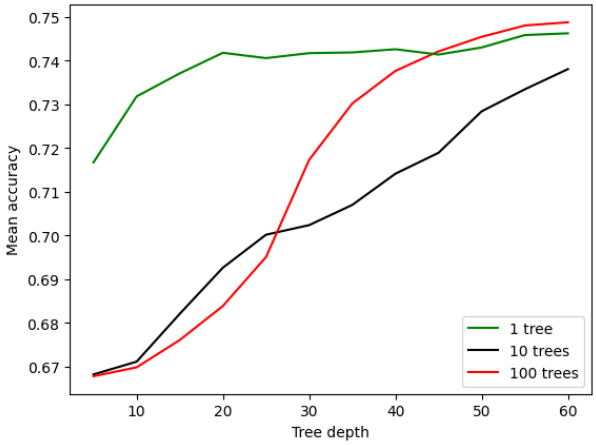

Figure 3: Comparison of Models based on Different three depth and number of trees.

Model interpretation: The decision tree model found the most important features to be able to identify offensive language based on key words.

Figure 4 represents model number 1. In this model, we can see that the decision rule starts by classifying tweets by the number of times the word "shit" was used (shit<=0.5),leaving 8221 tweets on the left child node and classifying them in the "Offensive" category, and 4092 tweets on the right child node and classifying them in the "Not Offensive" category. The Gini impurity of the root node, the probability of incorrectly classifying a datapoint, is 0.444.(Galarnyk)

Then, for the first right child node, our tree separates tweets again by the number of times the

word "nice" is used (nice<=0.5) creating two extra child nodes, 32 tweets with this word present are classified as "not offensive" and sent to the left second child node, and 301 tweets without this word present are classified as "Offensive". The Gini impurity of the first right-side child node is 0.174, lower than the root node Gini. After this, our tree creates four extra nodes, two for each child node.

After this, the tree continues classifying tweets depending on the words used.

In general, the deeper the tree, the more features it can use to increase its accuracy. However, it can become too complicated to understand and visualize. For this reason, model 1 is a decision tree with a maximum depth of 3.

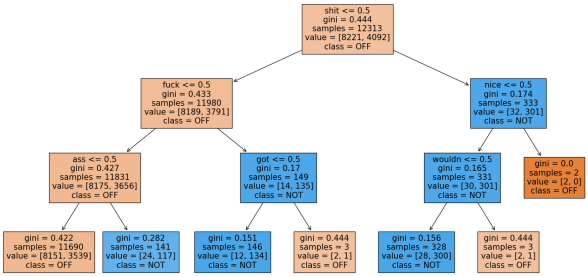

Figure 4: Decision Tree model

4 Data Size Effect

Data %	Total	% OFF	% NOT
25%	3078	32.71	67.28
50%	6156	33.15	66.84
75%	9235	33.40	66.59
100%	12313	33.23	66.76

Table 3: Train Dataset Statistics of Different Size

In the results obtained after comparing both models using a 5-fold cross validation in Table 4, it was found that the average accuracy for the decision tree is 0.74, while its test set accuracy score is 0.80. Meanwhile, the average accuracy for the random forest is 0.75, and its test accuracy score is 0.79.

In figure 5, we can observe that, on average, the random forest performs better than the decision tree for offensive language classification.

Taking into account Table 4 and Figure 6, we can observe that for the decision tree model, the size of the test data sets does not have a significant impact on the average performance across all metrics.

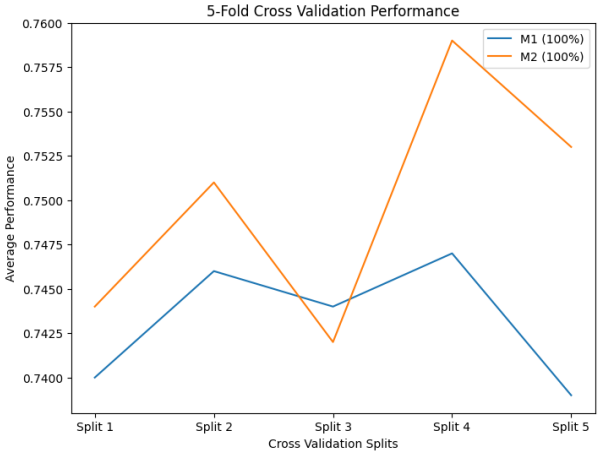

Figure 5: Cross Validation

Figure 6: Decision Tree model

When analyzing different test data sizes for Random Forest (Table 6 and Figure 7), there is a significant increase in average performance when using 75 and 100 percent of the data compared to using 25 and 50 percent across all metrics.

5 Summary

5.1 Discussion of work carried out

Overall, the F1-score of Model 1, Decision Tree Classifier with a depth of 3, is 0.566 with an accuracy of 0.761 on the test set.

Next we used a random forest classifier of 100 trees to compare it with our original tree of depth 3 and found that the random forest obtained an F1-score of 0.734 and an accuracy of 0.81 on the test set.

Both overcame the accuracy of 0.67. Meaning that both models outperformed the most basic model of labeling all tweets as "Not Offensive".

5-fold cv	M1(100%)	M2(100%)
Split 1	0.740	0.744
Split 2	0.746	0.751
Split 3	0.744	0.742
Split 4	0.747	0.759
Split 5	0.739	0.753

Table 4: Comparing two Model's using 100% training data: 5-fold cross validation.

Performance on test set	M1(25%)	M1(50%)	M1(75%)	M1(100%)
F1 score	0.539	0.542	0.565	0.566
Accuracy	0.752	0.753	0.760	0.761
Precision	0.816	0.818	0.820	0.830
Recall	0.561	0.563	0.577	0.578

Table 5: Comparing Model Size: Model performance metrics using Model 1 with different Data Size Training Data

Performance on test set	M2(25%)	M2(50%)	M2(75%)	M2(100%)
F1 score	0.683	0.703	0.734	0.734
Accuracy	0.790	0.791	0.813	0.810
Precision	0.773	0.756	0.795	0.783
Recall	0.663	0.684	0.711	0.714

Table 6: Comparing Model Size: Model performance metrics using Model 2 with different Data Size Training Data

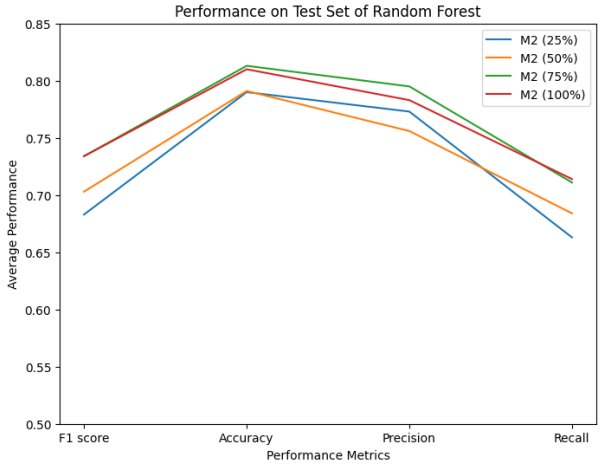

Figure 7: Decision Tree model

The random forest algorithm is reasonably good at predicting offensive language in tweets.

5.2 Lessons Learned

Decision trees are not the best tool to detect offensive language, but they provide full transparency in the classification process. Also, the most important step when building these models is the preprocessing stage, specifically tokenization. Stopword removal, lowercase, and keeping only alphabetical data increased the overall performance metrics.

The first time the models were executed with tokenization but without the other steps, the performance was lower than the accuracy of the most basic model of labelling all tweets as "not offensive".

References

Michael Galarnyk. Understanding Decision Trees for Classification (Python) — towardsdatascience.com.

Hunter Heidenreich. 2018. Natural Language Processing: Count Vectorization with scikit-learn — towardsdatascience.com.

Aakanksha Patil. 2021. Detecting Fake News using Supervised Learning — medium.com.

Mario Gutierrez Roig. Decision Trees and Random Forests: The Titanic Disaster — moodle.essex.ac.uk.