BOOTCAMP REPORT

Toxic Comment Classification Challenge

By

John Ndolo: jndolo@aimsammi.org **Yvan Pimi:** ypimi@aimsammi.org

Albert Agisha: aagisha@aimsammi.org **Mohammed Abukalam :** mabukalam@aimsammi.org

African Institute of Mathematical Sciences (AIMS-Senegal)

Group Project 6 - Public Score: 0.96999 - Private Score: 0.96857

Problem statement

Discussing things you care about can be difficult. Many people stop expressing their minds online due to the fear of abuse and harassment, and give up on seeking different opinions. Platforms struggle to facilitate conversations effectively, which forces many communities to restrict or disable user comments. In this context, the majority of platforms and parties have developed methods to identify and categorise different types of talks according to their level of toxicity. The study of harmful online behaviours, such as toxic remarks, is one area of emphasis (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). Different models have so far been developed to manage the degree of toxicity in talks. The existing models, however, continue to have flaws and don't let consumers choose the forms of toxicity in which they're most interested in finding(e.g. some platforms may be fine with profanity, but not with other types of toxic content).

Objectives

This challenge aims to build a multi-headed model that is capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate better than previous models. The model's goal is to predict a probability of each type of toxicity for each comment.

Data

The data used includes a large number of Wikipedia comments which have been labelled by human raters for toxic behaviour. The target toxicity level is divided into toxic, severe toxic, obscene, threat, insult, and identity hate for various comments in the data. The task is to model the data so that we can categorise each comment and assign a probability to each level of toxicity for the particular comment. The data is divided into training and test data. While the test data predicts the toxicity probability for these comments, the training data comprises comments with their binary labels. Some remarks in the test set are not scored in order to deter hand labelling and are assigned a value -1 to differentiate them.

Data Exploration and Methodology

The dataset used in this work is text data from kaggle. Because the data is sourced from web pages, there may be a lot of noise. As a result, before proceeding with the analysis and modelling, we cleaned and prepared the data so that it can be used by our models. Some of the preprocessings done to our data included removing punctuations (links, stopwords, html, non ascii, non printable, special characters, numbers, pattern), tokenization, stopwords, stemming, lemmatization (words and verbs) and Vectorizing data using term frequency inverse document frequency (Tf-idf). The preprocessing is clearly illustrated in the notebook file attached to this report. After data preprocessing, we choose two models; Logistic regression and Multinomial naive bayes which are described below. With the two models, our core objective was to assign probabilities of toxicity of comments classified into six classes (targets), namely, toxic, severe toxic, obscene, threat, insult, identity hate.

Models and Evaluations

A. Logistic regression

Logistic regression is a supervised learning algorithm used to predict a dependent categorical target variable. In essence, if you have a large set of data that you want to categorise, logistic regression may be able to help. For binary and linear classification problems, logistic regression is a simpler and more efficient method. It is a classification model that is simple to implement and achieves good results with linearly separable classes.

$$Logistic\, function\ = \frac{1}{1+e^{x}}$$

Where x is the input variable.

To complete the classification task in this project, we used logistic regression with regularisation. It was used in the case of multiple classes. For this model we did our implementation using the ScikitLearn library. For the best score, we had the regularisation hyperparameter C=1 achieving an accuracy score of 0.96999 as our public score and 0.96857 as the private score. We did three trials on the same algorithm by varying the hyperparameter “C” which demoted the regularisation term.

B. Multinomial Naive Bayes

This probabilistic algorithm applies the bayesian theorem to assign tags of texts or comments. For instance, putting tags on emails or newspaper stories. It calculates the tag’s likelihood for each text or document and outputs the tag with highest probability. Normally, the algorithm is best used in multiclass classification. The Bayes formula is given as;

$$P(A|B)\ = \frac{P(B|A) * P(A)}{P(B)}$$

In this study, we also employed the multinomial Naive Bayes implementing it from scratch and used Sklearn to compare our results. Both implementations gave good scores with implementation from scratch recording an accuracy score of 0.82867 and the Sklearn implementation recording a score of 0.94234 (public score) and 0.93954 (private score). To control overfitting we added a laplace smoothing parameter, alpha ($\alpha\ =\ 2$).

Results

Model	hyperparameter	Accuracy Public	Accuracy Private
Logistic Regression	C=1	0.96999	0.96857
Multinomial Naive Bayes	α =2	0.94234	0.93954
From Scratch (MNB)	α =2	0.84634	0.82867

Challenge

Fine tuning the hyperparameters to record an optimal score was our major challenge since it was based on trial and error. In Logistic Regression, the model can overfit if you have multiple highly-correlated inputs. As sourced from web pages and social media posts, the data was very noisy and that required meticulous preprocessing.

Conclusion

Through this exercise, which was designed to address social media firms' needs for the classification and rating of toxic comments, we explored one of the practical uses of machine learning, especially natural language processing. We managed to classify and assign probabilities on various classes of toxicity applied to a specific comment. A notable accuracy of 0.96857 was achieved with the use of the Logistic regression model and 0.93954 by use of multinomial naive bayes algorithm. Consequently, we concluded that the Logistic regression model was better than multinomial naive bayes algorithm for our research study. Additionally, we chose the two models because of their deceptive simplicity and ease of implementation, which provided adequate accuracy to aid in decision-making. Ultimately, we propose use of more robust natural processing algorithms such as LSTM, BERT and neural networks among others to improve on the accuracy of classifications in future works.

References

(1) Application of Logistic Regression in Natural Language Processing. Available from: https://www.researchgate.net/publication/342075482_Application_of_Logistic_Regression_in_Natural_Language_Processing [accessed Aug 13 2022].

(2) Xu, S., Li, Y., & Wang, Z. (2017). *Bayesian multinomial Naïve Bayes classifier to text classification*. In Advanced multimedia and ubiquitous engineering (pp. 347-352). Springer, Singapore. Available from: https://link.springer.com/chapter/10.1007/978-981-10-5041-1_57