A Comparative Analysis of Supervised Machine Learning Algorithms for Comment Toxicity Classification

Anonymous

1. Introduction

People can express their ideas and beliefs freely on the internet, but abusing this freedom can have a detrimental effect on the community. Reliable machine learning models and algorithms are needed to identify and assess such communication over the expanse of the Internet (Sheth et al., 2021).

The Conversation AI team, a research project started by Jigsaw and Google, has provided the dataset for Comment Toxicity. The identity and target labels were transformed from numerical values to binary values using a threshold of 0.5 (*Jigsaw Unintended Bias in Toxicity Classification*, n.d.).

Exploring the effectiveness of supervised machine learning algorithms in comment toxicity classification is my study's primary objective. Multinomial Naive Bayes, Linear Support Vector Classifier (SVC), Logistic Regression, and Decision Trees are the algorithms used in this study.

The top performing models were determined to be the Logistic Regression and the Linear SVC using various evaluation metrics, but the Logistic Regression had a significantly lower training cost than the Linear SVC.

Furthermore, if the decision threshold is optimized for Decision Trees and Naive Bayes they can perform better and provide more accurate predictions.

2. Literature Review

In supervised learning, the machine learning algorithm generates a classifier using a set of training documents for every class that have been (usually manually) classified.

A study compared the accuracy results on multiple benchmark datasets for text classification, Multinomial Naive Bayes outperformed many state-of-the-art Naive Bayes text classifiers (Wang et al., 2015). However, the study only tests these classifiers for multi class labels and does not take into consideration the classifier's performance for binary text classification tasks.

Another study compared Naive Bayes, KNN and SVC for binary text classification, and found that SVC performed better. However, the research also highlighted several disadvantages, such as the significant cost of training the SVM model on a large dataset (Colas & Brazdil, 2006) .

Furthermore, a study proposed that the kernel function and error parameter C selections have a significant impact on SVM results. As a result of this study , how to promote SVM for text classification is a challenge because researchers used a priori knowledge to select the kernel function and parameter (Liu et al., 2010).

One study used logistic regression and the Naive Bayes classifier to perform sentiment analysis. The performance is evaluated by accuracy and precision. In comparison to Naive Bayes classifier, the analysis using logistic regression works better by being 10.1% more accurate and 4.34% more precise whilst consuming approximately a fifth of the implementation time (Prabhat & Khullar, 2017).

It was also found that Decision Trees outperformed other algorithms like Naive Bayes and Vector Space Models in a study on the classification of Arabic text. The classifiers were evaluated on two different corpora, and the Decision Tree classifier performed better (Harrag et al., 2009). However, it should be mentioned that the analysis was performed on Arabic texts, and it's possible that the Decision Tree classifier won't perform better than the other classifiers for the given toxicity classification problem in English.

My work focuses on analyzing the four binary text classifiers, an area where there hasn't been much prior research.

3. Methodology

3.1. Analyzing the Dataset

The dataset (*Jigsaw Unintended Bias in Toxicity Classification*, n.d.) comprises 27 columns, but only two of them—Comment and Toxicity—are necessary for binary text classification. To make the dataset easier to understand and to make the subsequent processing steps faster, the other attributes have been discarded. In the train and test datasets, there are 1,40,000 and 15,000 instances, respectively.

3.2. Pipeline for processing textual data

The most textual information that serve to differentiate between text-categories are identified during the preprocessing stage of the analysis, which transforms the original textual information into a data-analysis-ready structure (Srividhya & Anitha, 2010). The pipeline for processing the textual data is depicted in Figure 1.

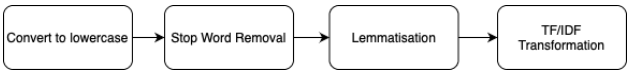

Figure 1. Pipeline for Processing Textual Data

3.2.1 Stop Word Removal

In machine learning, a number of the most used English words are irrelevant. They are referred to as "Stop words." Stop-words are frequently used words that provide no value, such as *pronouns, prepositions, and conjunctions.*

3.2.2 Lemmatisation

Lemmatization aims to reduce each word's inflectional forms to a single base or root word. It looks at each word's morphological analysis in order to determine the correct root. To provide that kind of analysis, dictionary for the language is necessary.

3.2.3 TF/IDF Transformation

In the field of text classification, the TF/IDF transformation is extensively used, and almost every other transformation technique is a variation of it. While TF reflects a word's frequency, IDF expresses each word's significance within the corpus (Srividhya & Anitha, 2010).

3.3. Classifier Fitting

Linear SVC, Logistic Regression, Decision Trees, and Multinomial Naïve Bayes are the four classifiers that were trained using the TF/IDF vectorized training dataset. Then, using the test data's vectorized comments, predictions are generated and further evaluated.

3.4. Evaluation Metrics

The efficiency of a model is explained by evaluation metrics. The ability of evaluation metrics to distinguish between different model outcomes is a crucial feature. The type of model, how it is implemented, and the problem being addressed each determine the metric selection. The metrics used to evaluate the classifiers used in this study are described below.

3.4.1 Confusion Matrix

By computing statistical metrics such as the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives, one may determine the accuracy of document classification systems (FN). Accuracy , F1 score, and many more metrics can be detected using Confusion Matrix (Navin J R & R, 2016).

3.4.2 F1 Score

The classes in the dataset (*Jigsaw Unintended Bias in Toxicity Classification*, n.d.) are unbalanced because, when the train and test datasets are merged, there are 1,29,679 non-toxic comments and 25,321 toxic comments as depicted in Figure 2 .

All instances may be classified in the larger class since there are two classes and one is significantly smaller than the other, enabling the classifier to reach high accuracy (Novaković et al., 2017).

The classifiers are evaluated using the macro averaged F1 score because of imbalance of class labels in dataset as depicted in Figure 2. Since macro averaging treats both classes equally and accords no class to greater significance, macro averaged F1 score is utlised.

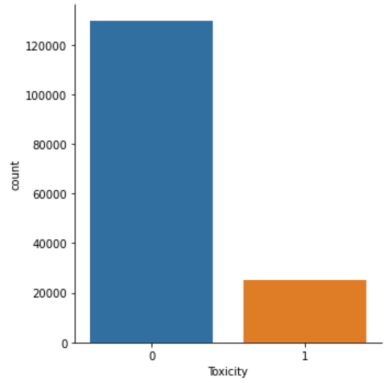

Figure 2. Class Distribution of Dataset

3.4.3 AUC-ROC Curve

An indicator of performance for classification tasks at varied threshold levels is the AUC-ROC curve. AUC (Area Under Curve) represents for the level or measurement of separability, while ROC (Receiver Operating Characteristic) is a probability curve. It illustrates how effectively the model can differentiate among classes. The higher the AUC, the better the model performs in classification tasks.

4. Results

A brief overview of the classifiers' performance in regards to various evaluation metrics is presented in this section. The following subsections provide additional detail on the evaluation measures that were employed, which are Confusion Matrix, F1 Score and AUC-ROC Curve.

4.1. Confusion Matrix

Figure 3 illustrates the confusion matrices for the classifiers, and Table 1 presents the FP, TP, TN, and FN values.

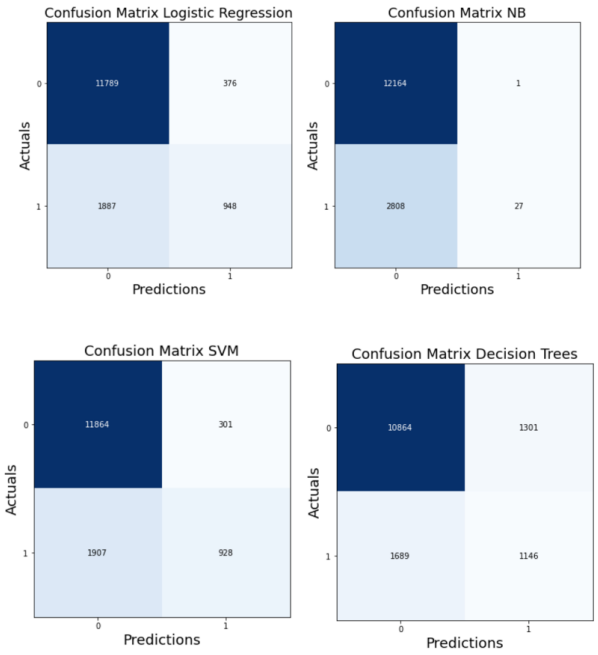

Figure 3. Confusion Matrices for Algorithms

The true positives (TP) and true negatives (TN) for each algorithm are higher for the classifiers, and they accurately predict in majority of cases irrespective of the class distribution. Performance measures like F1 Score, Specificity, and Sensitivity are evaluated with the use of the Confusion Matrix. The below table presents the results of all the classifiers of all the categories in the matrix.

Model	False Positive (FP)	False Negative (FN)	True Positive (TP)	True Negative (FN)
Linear SVC	301	1907	11864	928
Logistic Regression	376	1887	11789	948
Multinomial NB	1	2808	12164	27
Decision Trees	1301	1689	10864	1146

Table 1. Confusion Matrix Results Summary

4.2. F1 Score

Table 2 gives an overview of the study's findings, and Figure 4 compares the F1 scores of the classifiers.

Model	Macro F1 Score
Linear SVC	0.69
Logistic Regression	0.68
Multinomial NB	0.46
Decision Trees	0.66

Table 2. Macro Averaged F1 Scores of Algorithms

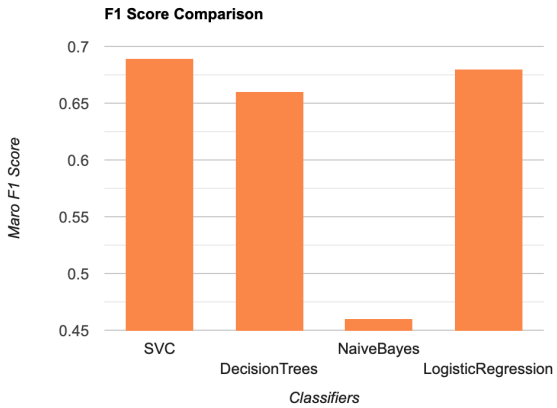

Figure 4. Comparison of F1 Scores

With an F1 score of 0.69, Linear SVC gets the highest score, closely followed by Logistic Regression's 0.68. With an F1 score of 0.66, Decision Trees algorithm comes in third place to the other two. However, Naive Bayes demonstrates a sharp decline in F1 score, with an F1 score of just 0.46.

4.3. AUC-ROC Curve

The ROC-AUC Curve for the four classifiers in comparison to the random prediction baseline is presented in the figure below. A straight line that randomly selects any class is the Random Prediction Baseline. It is depicted as the dotted line in figure 5.

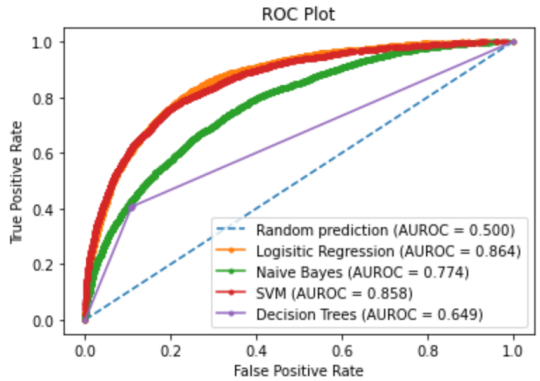

Figure 5. ROC Plot for Algorithms

Table 2 presents the ROC scores of the classifiers on the test dataset.

Model	ROC Score
Linear SVC	0.858
Logistic Regression	0.864
Multinomial NB	0.774
Decision Trees	0.649

Table 3 ROC Scores of Algorithms

The classifiers' Area Under Curve (AUC) provides a good indication of how well algorithms predict the labels.

5. Discussion

The macro averaged F1 Score is the benchmark used to evaluate the classifiers due to the imbalanced classes in the dataset.

The best classifier for predicting class labels is the Linear Support Vector Classifier (SVC), which has the greatest F1 Score of any classifier analyzed. Additionally, Linear SVC has the second highest ROC score and Area Under Curve (AUC), further demonstrating its effectiveness as a classifier among the other alternatives.

SVCs have the ability to manage large feature spaces because they employ overfitting protection, which is independent of the number of features. Since the majority of text classification issues can be separated linearly, the linear kernel is used for SVC, using alternative kernels produced worse results.

Typically, binary classification tasks are solved using logistic regression, which extracts features from instances to predict the class label. Given that the Comment Toxicity problem is a binary classification problem, Logistic Regression works very well and requires far less training time than Linear SVC. The ROC curve for logistic regression overlaps with the SVM ROC curve for the majority of the data and has the greatest ROC Score of 0.864 among all the classifiers, and it has an F1 score of 0.68. It should be emphasized that the Logistic Regression Algorithm does poorly with multiclass labels.

The simplicity and interpretability of decision trees are excellent, but their capacity to learn complex rules and scale to huge amounts of data is more limited. With an F1 score of 0.66 but a weak

ROC score of 0.44, Decision Trees had the lowest area under the curve of all the classifiers. The classifier performs a respectable job of classifying at the present threshold but will perform badly for alternative threshold values, according to the low ROC Score and high F1 score.

Lastly, the Naive Bayes classifier, has a F1 score of 0.46 and a ROC score of 0.774. The classifier performs similarly to Decision Trees but not as well as Linear SVC and Logistic Regression.

The Naive Bayes classifier currently performs poorly due to the disparity in ROC and F1 scores, however an ideal threshold can benefit the classifier in producing more accurate predictions. It should be emphasized, that even with training on the ideal threshold, the ROC Curve does not indicate that the Naive Bayes classifier will outperform Logistic Regression or Linear SVC.

6. Conclusion

Overall, these results indicate that, for our dataset, Logistic Regression worked as well as Linear SVC and required considerably less training time. The most effective classifier, despite the longest calculation time, was linear SVC. Despite their poor performance, Decision Trees and Naive Bayes could both be improved to perform much better by adjusting the threshold to improve their ROC and F1 scores.

References

Colas, F., & Brazdil, P. (2006). Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In M. Bramer (Ed.), *Artificial Intelligence in Theory and Practice* (Vol. 217, pp. 169–178). Springer US. https://doi.org/10.1007/978-0-387-34747-9_18

Harrag, F., El-Qawasmeh, E., & Pichappan, P. (2009). Improving arabic text categorization using decision trees. *2009 First International Conference on Networked Digital Technologies*, 110–115. https://doi.org/10.1109/NDT.2009.5272214

Jigsaw Unintended Bias in Toxicity Classification. (n.d.). Retrieved September 28, 2022, from https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification

Liu, Z., Lv, X., Liu, K., & Shi, S. (2010). Study on SVM Compared with the other Text Classification Methods. *2010 Second International Workshop on Education Technology and Computer Science*, 219–222. https://doi.org/10.1109/ETCS.2010.248

Navin J R, M., & R, P. (2016). Performance Analysis of Text Classification Algorithms using Confusion Matrix. *International Journal of Engineering and Technical Research (IJETR)*, *6*(4), 75–78.

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., & Milica, T. (2017). Evaluation of Classification Models in Machine Learning. *Theory and Applications of Mathematics & Computer Science*, *7*(1), Article 1.

Prabhat, A., & Khullar, V. (2017). Sentiment classification on big data using Naïve bayes and logistic regression. *2017 International Conference on Computer Communication and Informatics (ICCCI)*, 1–5. https://doi.org/10.1109/ICCCI.2017.8117734

Sheth, A., Shalin, V. L., & Kursuncu, U. (2021). *Defining and Detecting Toxicity on Social Media: Context and Knowledge are Key* (arXiv:2104.10788). arXiv. http://arxiv.org/abs/2104.10788

Srividhya, V., & Anitha, R. (2010). *Evaluating Preprocessing Techniques in Text Categor ization*. *2010*, 3.

Wang, S., Jiang, L., & Li, C. (2015). Adapting naive Bayes tree for text classification. *Knowledge and Information Systems*, *44*(1), 77–89. https://doi.org/10.1007/s10115-014-0746-y