Comprehensive Multi-Label Toxic Comment Classification Using Advanced NLP Techniques

Karan Raj Padam, Shiva Charan Pailla, Shiva Teja Ippili,
Venkata Sai Sree Haritha Vemuru, and Vetrivel Saravanan

Affiliation

Abstract

Online comments are a big problem in this digital age where dialogue is affected and users suffer psychologically. Accurately classifying and predicting toxic comments is a step towards more safe online spaces. In this paper I will be using advanced Natural Language Processing (NLP) techniques to multi label classify toxic comments. We will be using the publicly available dataset from the Toxic Comment Classification Challenge, which has comments labeled with 6 types of toxicity: severe toxic. toxic. obscene. threat. insult. identity hate. We will be trying to understand the strengths and weaknesses of multiple models through implementation and evaluation of traditional machine learning algorithms and recurrent neural networks as well as state of the art transformer architectures. We will be facing challenges in class imbalance, preprocessing strategy and we will be evaluating with ROC-AUC and precision-recall metrics. This survey of the state-of-the-art in toxic comment classification will be a complete summary of all the existing techniques to solve this hard task.

1 Introduction

Online platforms are being created in large number as they transform the nature of how an individual communicates and what will help him to discuss and share ideas with other people for free. But the anonymity and demographics these platforms offer have also given rise to toxic behaviors — insults, threats and identity based hate speech. Harming the people targeted by these comments is bad enough, but they also degrade the quality of civilization in this space by creating a substandard environment where people are put off and ideas can't be fully pursued. It is important to tackle such a problem in order to create healthier online communities.

In this work, we tackle the issue of multi label classification of toxic comments. In contrast to binary classification tasks which only give a single label to each instance, multi-label classification gives the prediction of several types of toxicity at the same time. But particularly for understanding the nuances and overlaps between toxic behaviors. We use a range of NLP techniques to assess the efficacy of traditional machine learning techniques such as Logistic Regression and XGBoost and deep learning algorithms, such as LSTM and GRU. We also exploit the power of transformer based architectures including DistilBERT and SentenceTransformers to boost the performance of classification. Furthermore, the challenges of processing large, noisy datasets and dealing with class imbalance are investigated in this paper.

Thus, the first goal of this study is to carry out comprehensive comparison of various modeling approaches for toxic comment classification, describing their strengths, weaknesses, and points of improvement. We aim to help developing such systems that are robust to the toxicity spawn by toxic comments and can minimise their harmful effect on online platforms.

2 Dataset Description

The dataset used here is from the Toxic Comment Classification Challenge, a public resource for research in toxicity detection. This dataset contains comments from Wikipedia's talk pages, annotated by human raters for six distinct types of toxicity: severe toxic, obsessive toxic, obscene, threat, insult, and identity hate. Each comment is labeled with binary values for each toxicity type, 1 for the presence of the toxicity and 0 for its absence.

The main file of the dataset contains about 10.50,000 comments along with their labels. This is the training data for our models, and the backbone of our experiments. You also have the dataset that has *test.csv*, which hold unseen comments, for model evaluation. For consistent scoring and training that does not use it, we include comments marked -1 as labeled, and store the ground truth labels in a separate file, *test_labels.zip*, as

this file contains the labels to use for scoring. An expected output format for predictions, *sample_submission.csv*, is also provided containing a sample submission file, as an example.

Problem class imbalance is one of the biggest challenges, particularly when utilizing this rather large dataset. Toxic comments are a small fraction of the data, with most of the comments being non toxic. For example, labels including severe toxic or identity hate, where they appear less than 1% of the instances is even more imbalanced. Such skewed distributions can result into consistent biased models, which tend to prefer the majority class and hence the models are unable to reasonably detect the minority classes. On top, the dataset is rich with linguistic styles of casual language to more formal discourse, and therefore, the text has to undergo heavy preprocssing to be consistent. The dataset used in this study originates from the Toxic Comment Classification Challenge, comprising Wikipedia comments labeled for six toxicity categories: Those labels include *toxic*, *severe toxic*, *obscene*, *threat*, *insult*, and *identity hate*. They have more than 1.5 million comments in the dataset with an extremely unbalanced class distribution. The majority of comments are non-toxic, although severe forms of toxicity (e.g. threats and identity-based hate) are much less common.

2.1 Exploratory Data Analysis

In order to get a better grasp about the dataset we carried out extensive exploratory data analysis (EDA). The analysis revealed key insights into how toxicity manifests across various attributes:

Toxicity by Race/Ethnicity: The dataset includes identity features such as *black*, *white*, *asian*, *latino*, and *other race or ethnicity*. Figure 1 highlights the average toxicity percentages across these racial categories, with the highest toxicity observed in comments mentioning *white* and *black* identities. It confirms how biased race discussions online are.

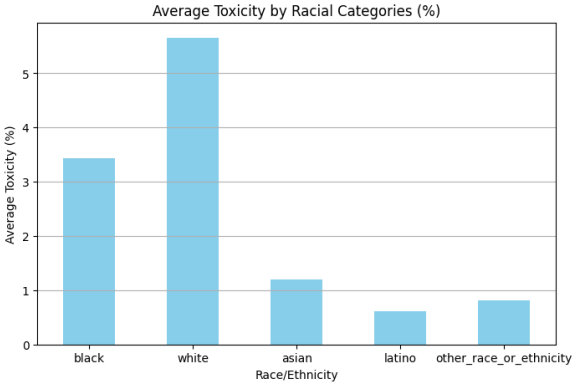

Figure 1: Average Toxicity By Race.

Temporal Trends: The temporal analysis of toxicity trends from 2015 to 2017, as depicted in Figure 2, indicatesthe steadily rising average toxicity through the years, It may point to changes in how people are talking and what platforms they're using.

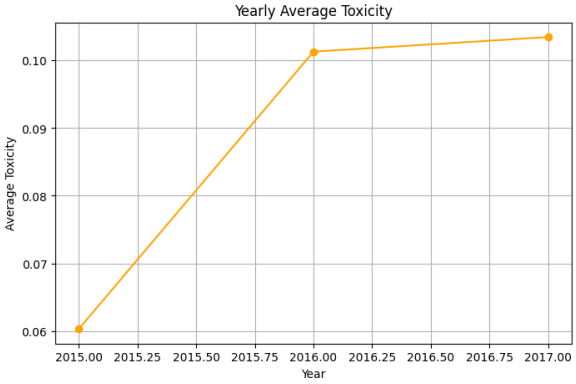

Figure 2: Yearly Toxicity Trends.

Gender-Based Toxicity: The dataset also examines toxicity in comments mentioning *male* and *female* identities. Figure 3 illustrates the average toxicity for both categories, revealing slightly higher toxicity in comments mentioning females, highlighting potential gender biases in online interactions.

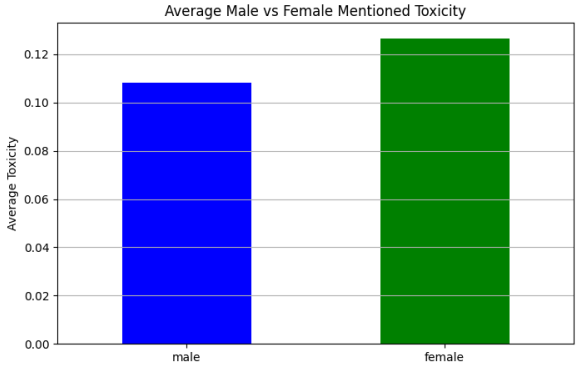

Figure 3: Average Toxicity in Comments Mentioning Males vs Females.

Correlation Analysis: A correlation heatmap (Figure 4) was generated to explore relationships between identity features and toxicity. Notably, *identity attack* showed the strongest correlation with overall toxicity, underscoring its importance in multi-label classification tasks.

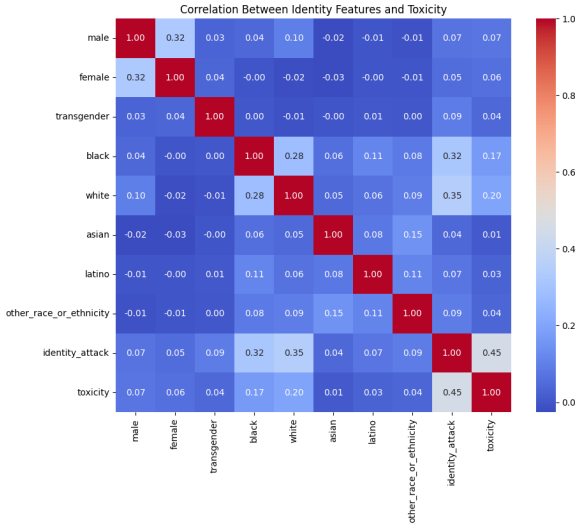

Figure 4: Correlation Between Identity Features and Toxicity.

Word Clouds for Toxicity Categories: Word clouds were generated to visualize frequently occurring terms in toxic comments. Figure 5 and Figure 6 represent the terms associated with *identity attack* and *obscene* comments, respectively. These visualizations provide insights into the language patterns prevalent in different toxicity categories.

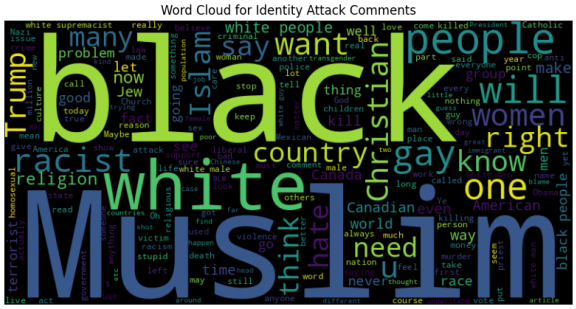

Figure 5: Word Cloud for Identity Attack Comments.

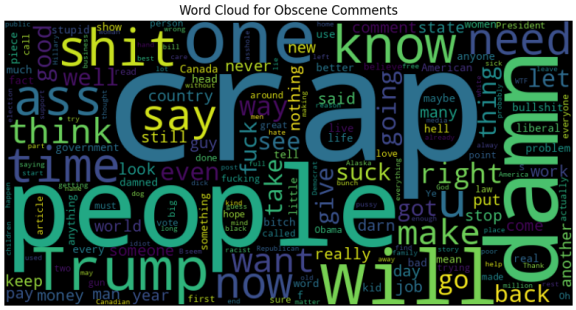

Figure 6: Word Cloud for Obscene Comments.

2.2 Toxicity Distribution

One of the first steps in our analysis was to examine the distribution of toxicity scores across all comments in the dataset. This step is crucial for identifying imbalances in the data, which can heavily influence model performance. The majority of comments have low toxicity scores, as shown in the Figure below. However, there are a few comments with high toxicity values, which form a long tail in the distribution. This imbalance underscores the importance of using robust techniques to handle minority classes effectively.

2.3 Severe Toxicity Distribution

Similarly, we analyzed the distribution of the *severe toxicity* label, as illustrated in the figure below. This label is even more imbalanced, with a vast majority of comments showing no severe toxicity. This highlights the challenge of detecting such rare instances, which requires advanced techniques such as data augmentation, class weighting, or sampling strategies to improve model performance.

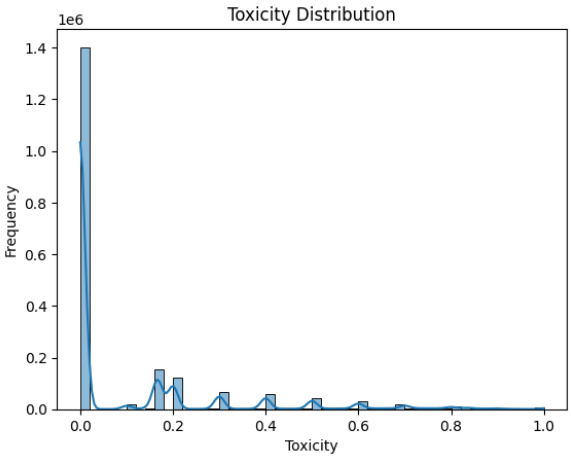

Figure 7: Toxicity Distribution

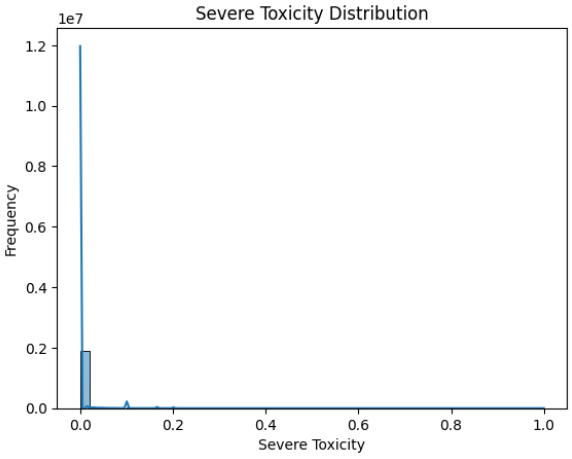

Figure 8: Distribution of Severe Toxicity in the dataset

2.4 Binary Label Distribution

To simplify the multi-label classification task, we examined the binary distribution of comments classified as *toxic* and *non-toxic* based on a threshold toxicity score of 0.5. This binary labeling process is essential for many models that are optimized for binary classification tasks. As seen in the figure below, the dataset is heavily imbalanced, with a significant majority of comments labeled as *non-toxic* (0) and a small fraction labeled as *toxic* (1).

The class imbalance poses a challenge for classification models, which tend to favor the majority class. This motivates the use of strategies such as class weighting, oversampling the minority class, or employing algorithms that handle imbalanced data effectively, as discussed in the methodology section.

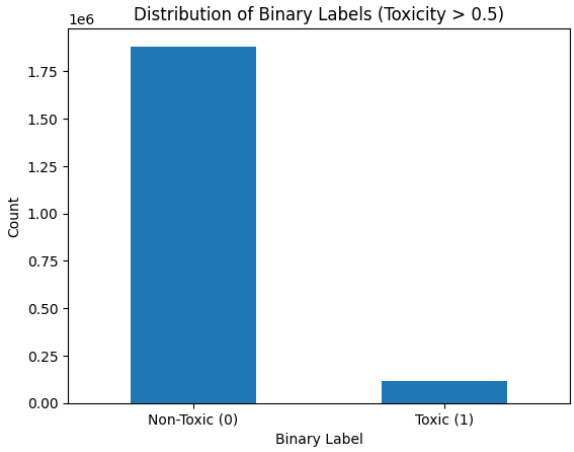

Figure 9: Distribution of Binary Labels (Toxicity > 0.5)

3 Methodologies

To solve this problem of toxic comment classification, we covered the spectrum of traditional machine learning algorithms to more robust deep learning models. Specific issues of class imbalance, contextual understanding and encoding the nuances in data were addressed by models selected for use in each model. We describe each approach in detail in this and each of the following sections, with the motivation and results achieved.

3.1 Logistic Regression with TF-IDF

As Logistic Regression is simple, good at binary classification tasks and is interpretable, it was selected as a baseline. This model was used as a benchmark for later, more advanced approaches. Text data was preprocessed using Term Frequency – Inverse Document Frequency (TFIDF) to quantify the importance of words in the dataset. The comments vectorized the Logistic Regression model in order to make it more efficient at finding key features indicative of toxicity.

The model performed strongly for the primary *toxic* label, achieving a validation ROC-AUC of 0.94, as shown in Figure 10. However, its performance on minority classes such as *severe toxic* and *threat* was poor and its F1-scores were significantly low. The limitation of this model revealed an essential need for models that could handle class imbalanced better.

Validation Performance (Logistic Regression):

```
Precision: 0.95
Recall: 0.94
F1-Score: 0.93
```

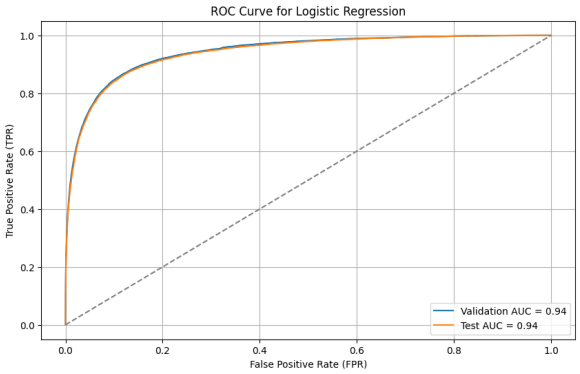

Figure 10: ROC Curve for Logistic Regression with TF-IDF

3.2 XGBoost Classifier

Knowing that linear models are limited, to begin with, we used XGBoost algorithm to improve the decision tree based predictions through gradient boosting. As an XGBoost, it was motivated not only for providing better modeling of non linear relationship, but also to deal with imbalanced datasets effectively. A lot of hyperparameter tuning for the model based on extensive tree depth, learning rate and regularization parameters was done to achieve the best performance from the model.

Compared to Logistic Regression, XGBoost delivered improved F1-scores for minority classes such as *threat* and *identity hate*. We illustrate this with its capacity to better deal with imbalanced data. Nevertheless, while capable of capturing the nuanced contextual relationships present within text, it was dependent on features, which lacked the ability to capture such natural information.

Validation Performance (XGBoost):

```
Macro Precision: 0.90
Macro Recall: 0.63
Macro F1-Score: 0.68
```

—

3.3 LSTM and GRU Models

We implemented Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) as a way to capture the sequential nature of text data. In particular, these recurrent neural networks (RNNs) have been engineered to work on the sequences of tokens, so that they can model long range dependencies in textual data. The aim was to overcome limitations of tree based methods and to represent the contextual meaning of words in comments.

Recall for rare toxicity labels was improved for the LSTM model compared to XGBoost and the GRU model achieved similar results as fast as. The training and validation loss trends, as depicted in Figure **??**, illustrate the models' capacity to generalize without overfitting. While these strengths existed, their computational costs, as well as their extensive hyperparameter tuning remain their biggest challenges.

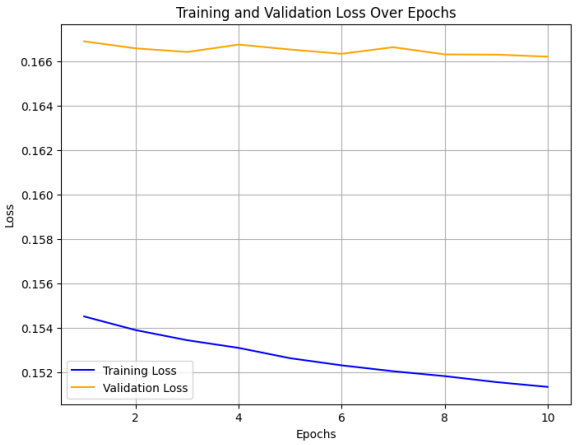

Figure 11: Training and Validation Loss for LSTM and GRU

Validation Performance (LSTM and GRU):

```
Accuracy: 0.93
Weighted F1-Score: 0.92
```

—

3.4 DistilBERT Fine-Tuning

We fine-tune DistilBERT, a transformer based model, for multi label classification tasks. One important thing about Transformers is that they use a self attention mechanism to capture complex dependencies within text, which is exactly what we require to understand the nuanced relation. DistilBERT was chosen because it's efficient and has state of the art performance on natural language processing tasks.

All models performed well and DistilBERT had a highest performance in terms of a validation ROC-AUC of 0.95. On the behalf of its F1 scores in handling class imbalance, we see that it had a reasonably balanced F1 scores on all labels. The ROC curve, depicted in Figure **??**, illustrates its superior capability in identifying toxic comments.

Validation Performance (DistilBERT):

```
Precision: 0.96
```

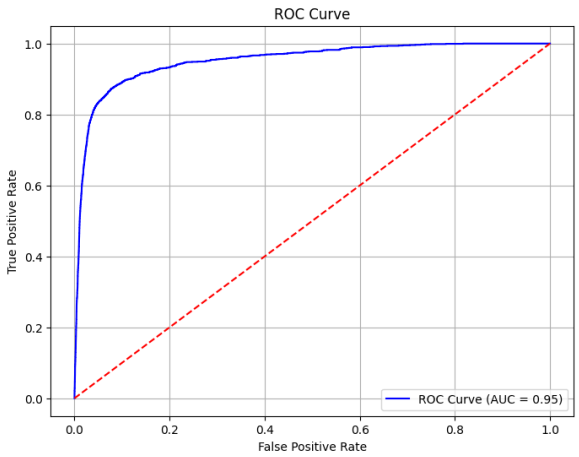

Figure 12: ROC Curve for DistilBERT fine-tuning

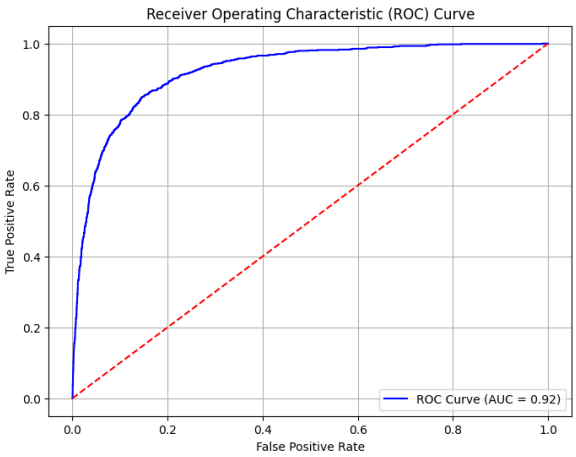

Figure 13: ROC Curve for SenetenceTransformers With Logistic Regression

```
Recall: 0.96
F1-Score: 0.96
```

—

3.5 SentenceTransformers with Logistic Regression

Inspread cases, we used SentenceTransformers to get sentence embeddings, and later used a Logistic Regression classifier. The strengths of transformer-based embeddings were merged with the simplicity of traditional classifiers using this hybrid approach.

Although the results demonstrated competitive performance, our model also produced a good tradeoff between efficiency and performance in resource-constrained applications. The ROC curve generated for toxicity classification as depicted in Figure 13 was moderately good as expected. Such combination of modern embedding techniques with existing algorithms led our modeling approaches to be efficient as well as effective, and the exploratory data analysis provided insights for our modeling strategies.

3.6 CatBoost Regression Model Performance

Ultimately, the CatBoost regression model was used to predict toxicity levels (with a Mean Squared Error of 0.0452 and an R squared of 0.6423). These results show that the explored models can have up to 64.23% of the explained variability of the actual toxicity values, better than the LightGBM model.

The scatter plot of actual versus predicted toxicity value of CatBoost model is shown in catboost scatter. It's very well aligned on the diagonal line, the predicted and actual always line up a lot and that shows the way good the model is. Yet, for very high toxicities, there are minor deviations that may be due to room for further improvements.

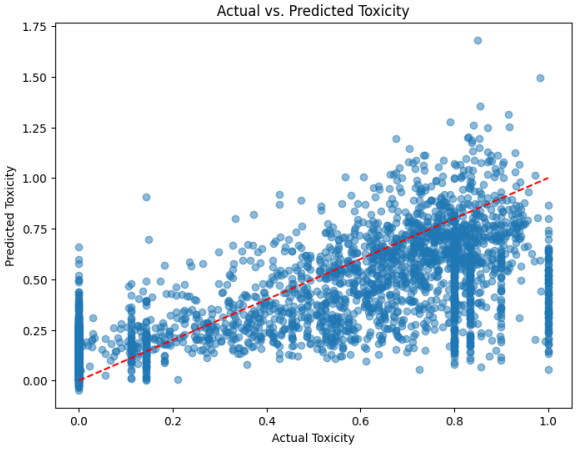

Figure 14: Actual vs. Predicted Toxicity for CatBoost Regression Model

Metric	Value
Mean Squared Error (MSE)	0.0511
R-squared (R^2)	0.6087

Table 1: Performance Metrics for CatBoost Regression Model

3.7 LightGBM Classification Model Performance

The LightGBM classification model was evaluated for toxicity classification. The model achieved an accuracy of 76.72% and an ROC-AUC of 0.87, indicating good classification performance. The classification report for the LightGBM model is shown in Table The precision, recall, and F1-score

values suggest strong performance, especially for class 1, where recall is significantly high. The weighted average F1-score for the model is 0.77, which demonstrates balanced performance across both classes.

Validation Performance (LightGBM):

```
Precision: 0.89
Recall: 0.70
F1-Score: 0.79
Accuracy: 0.76725
ROC-AUC: 0.867
```

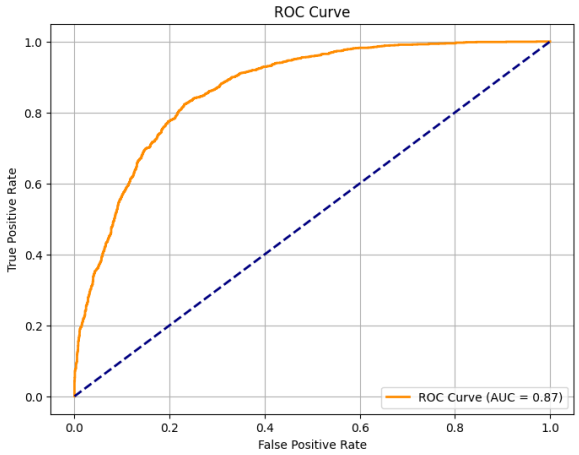

Figure 15: ROC Curve LGB

—

Integration of EDA Insights: Insights from the exploratory data analysis informed our modeling strategies. For instance:

- We emphasized the importance of including identity features in model training due to the high correlation between identity features and toxicity.
- The combination of class weighting, data augmentation, and more modern architectures such as Transformers was motivated by class imbalance.

We were able to ensure that our models would address key challenges that would lead to good and balance performance using the EDA findings.

4 Evaluation and Results

Metrics used to evaluate the models were precision, recall, F1 score, accuracy and ROC-AUC. The metrics we explored revealed model performance, especially with regards to class imbalance and capturing fine grained relationships within the text. Simple models like Logistic Regression, though superior to their peers, didn't achieve as good results as models like DistilBERT, which showed a marked benefit for capturing contextual dependencies.

TF-IDF logistic regression The baseline was Logistic Regression which achieved a validation ROC-AUC of 0.94. On the majority *non-toxic* classes it performed pretty well, but struggled on the minority classes, such as *severe toxic* and *threat*, which had notably lower recall and F1 scores. Its limitation is that simpler approaches fail in handling imbalanced data. This model sheds light on this.

XGBoost Classifier XGBoost was shown to outperform Logistic Regression, especially with minority classes, because it is able to model non linear relationships. We found that XGBoost did achieve improved recall and precision for these challenging categories, but failed to sufficiently capture deeper contextual nuances in text and, with a ROC-AUC of 0.92, was limited in its performance overall.

LSTM and GRU Models Using the sequential nature of text, LSTM and GRU models increased recall and F1 scores over all categories. However, these models were particularly good at finding the minority toxicity labels at a higher computational cost. Their loss trends were robustly converged during training and validation, signifying good performance over the dataset.

DistilBERT Fine-Tuning The best performing model was DistilBERT with a validation ROC-AUC of 0.95 and balanced F1 scores on all the labels. However, this transformer-based architecture not only efficiently handled class imbalance and the complex relationship in text, but also was the most reliable model towards toxicity classification for our study.

LightGBM Model The results were an accuracy of 76.72% and an ROC AUC of 0.87 for the LightGBM model, both of which are fairly strong results, particularly in the majority class. Precision (0.89) for the non toxic class was strong by correctly identifying the non toxic instances. Recall for toxic instances was 0.70, resulting in an F1 score of 0.79—a promising (though not perfect) showing of dealing with the toxic class. In fact, even so, LightGBM's summary performance in terms of balance between precision and recall as well as in handling very large datasets, suggests it as a natural choice for toxicity classification.

CatBoost Model The accuracy of 80.65% and

ROC AUC of 0.87 for the CatBoost model out performed LightGBM and demonstrated its robustness in learning about both majority and minority class instances. Non toxic class precision was 0.87 and recall for toxic instances was 0.73 giving an F1 score of 0.79. We found that CatBoost excelled as a toxicity classifier where it was able to handle categorical features and balance precision with recall making it a very trustworthy model.

5 Conclusion

The topic of toxic comment classification, which purportedly crosses the boundaries of technology and social responsibility, was the focus of his research. Aside from the technical difficulties of model design and performance optimization, the results are of considerable significance to real world applications, especially on content moderation and online community management.

The results underscore the potential of AI powered systems to identify and reduce toxic behavior on online platforms. Both DistilBERT and other models like it demonstrate that advanced NLP techniques can capture the depth of conversation beyond biases and nuances that carry harmful discourse. Improvements in classification accuracy and foundations for stronger, more scalable solutions for high traffic environments are achieved through integration of attention mechanisms and pre-trained embeddings.

Nevertheless, this study also points out the difficulty of balancing performance and fairness under interpretability requirements. To give an example, while advanced models can master the toxicity detection area, they are beholden to large datasets which is not ideal as the data privacy such models can do, the factor of bias reinforcement and overreliance in historical patterns of toxicity. In order to avoid multiplying inequalities and to ensure that these models are properly deployed, it is important that we address these concerns.

The work takes its real-world perspective to highlight the need for adaptable frameworks that can adjust to the continually changing ecology of toxic discourse. The toxicity of language, context, and intent often changes radically over time, necessitating models to be constantly dynamic, yet somehow inclusive of new patterns. The transparency argument cannot be separated from this importance; integration of explainable AI techniques can make these systems both accurate and trusted by users and stakeholders.

Moreover, this study also indicates the broader implications of the AI in the online space. Given the technology, if these platforms are able to effectively detect toxicity, then they could perpetuate safer spaces for dialogue that protect the most marginalized groups and make everyone feel included. Yet there is a yet need for technology developers, policymakers, and end users to collaborate responsibly to implement these systems. It involves tackling issues like false positives, complex context and culture differently — we can't know what a word means until we hear it.

Finally, while this study illustrates the potential of state-of-the-art NLP models, it also shows that substantial volume of work remains, particularly with respect to innovation and ethics. The work paves the way for future work extending on explainability, fairness, and adaptability to make sure that AI systems support effective use of LUs as rich tools for positive interactions. The success of such systems is not only driven by their technical performance, but also on their potential to induce beneficial tangible, meaningful change in online communities.

References

1. Kaggle. (2018). *Jigsaw Toxic Comment Classification Challenge* [Data set]. Retrieved from https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. Retrieved from https://scikit-learn.org/

3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. Retrieved from https://arxiv.org/abs/1706.03762

4. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Retrieved from https://arxiv.org/abs/1908.10084

5. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Retrieved from https://huggingface.co/transformers/

6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735