CS273A Final Project Report
Toxic Comment Classification

Group 4

Abdul Kalam Syed
abdulks@uci.edu
Student ID: 75307042

Jiahao Xu
jiahx11@uci.edu
Student ID: 47632271

Abstract

This report investigates the problem of multi-label toxic comment classification using the Jigsaw Toxic Comment Challenge dataset. We conduct comprehensive data exploration, build multiple classical machine learning baselines, analyze the effects of data imbalance and downsampling, experiment with intermediate embedding-based models such as FastText, and finally evaluate transformer models, including BERT and RoBERTa. An ensemble model combining our strongest classifiers is also constructed. Our results demonstrate the impact of dataset imbalance on classical models, the benefits of downsampling for macro-F1 performance, and the strong predictive advantage of modern transformer architectures.

1 Introduction

Online platforms host millions of user-generated comments every day, a significant portion of which contain toxic content such as insults, threats, obscenity, and identity-based hate. Automatically detecting and categorizing such content is essential for maintaining healthy online communities and protecting users from harassment and abuse.

Effective toxic comment classification enables a wide range of moderation tasks, including filtering or flagging harmful content, prioritizing moderator review, suspending accounts that repeatedly violate community guidelines, and escalating credible threats to appropriate authorities. Given the scale and volume of online discussions, manual moderation alone is infeasible, making machine learning-based approaches a critical component of modern content moderation systems.

The Jigsaw Toxic Comment Classification dataset [1] provides a multi-label annotation scheme across six toxicity types, creating a challenging classification task due to significant class imbalance, linguistic variability, and multi-label dependencies.

This project aims to:

- Explore and analyze the toxic comment dataset.
- Develop several classical baselines (Naive Bayes, Logistic Regression, SVM) using TF-IDF features.
- Investigate preprocessing and downsampling strategies for imbalance mitigation.
- Evaluate FastText as a mid-level embedding baseline.
- Train and assess transformer models (BERT, RoBERTa).
- Build an ensemble combining the strongest models.

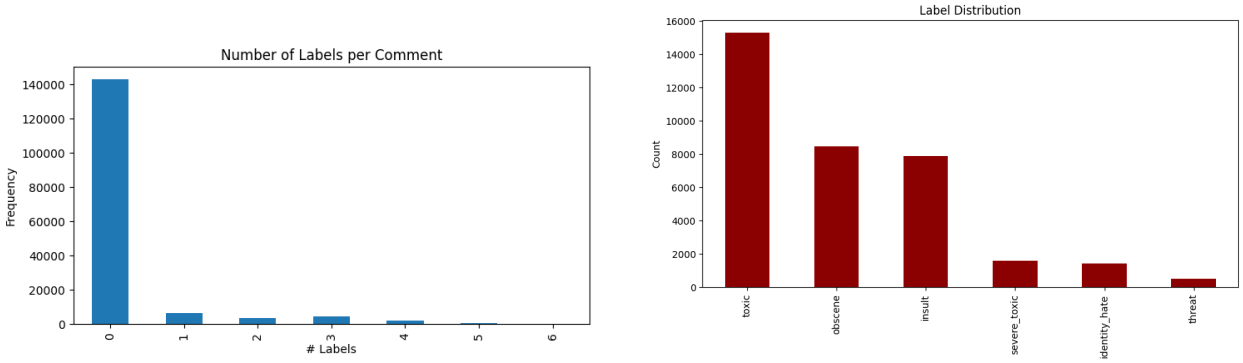
Our methodology emphasizes understanding the data, making principled modeling decisions, and comparing techniques across classical, embedding-based, and deep learning paradigms.

2 Dataset Exploration

The dataset contains approximately 160,000 comments, each annotated for six toxicity labels: `toxic`, `severe_toxic`, `obscene`, `threat`, `insult`, and `identity_hate`. A comment may exhibit multiple toxicity types.

2.1 Label Distribution

Exploratory analysis revealed extreme imbalance: over 85% of comments contain no toxicity label. Among toxic labels, `toxic` and `obscene` are the most common, while `threat` is exceedingly rare. This imbalance motivates strong regularization and downsampling. To further characterize the imbalance and multi-label structure of the dataset, we visualize both the number of toxicity labels assigned per comment and the frequency of each individual label.



(a) Number of labels per comment

(b) Per-label frequency

Figure 1: Multi-label structure and class imbalance in the Jigsaw Toxic Comment dataset.

2.2 Impact of Imbalance on Model Choice

Extreme class imbalance is known to disproportionately harm recall for minority labels in multi-label text classification tasks. In toxicity detection, rare categories such as `threat` and `identity_hate` often rely on contextual cues rather than frequent surface-level keywords, making them difficult to capture using purely lexical models [9, 8].

Prior work has shown that transformer-based architectures, which model contextualized token representations via self-attention, are more robust to sparse and semantically diverse toxic expressions than traditional bag-of-words approaches [7, 8]. This motivates our later evaluation of BERT and RoBERTa models, particularly with respect to minority label recall.

2.3 Evaluation Metrics

Given the extreme class imbalance in the Jigsaw Toxic Comment dataset, accuracy alone is not a reliable performance metric. Because over 85% of comments contain no toxic labels, a classifier that predicts all samples as non-toxic would achieve high accuracy while failing to detect toxic content.

We therefore prioritize F1-based metrics, which balance precision and recall and provide a more informative assessment of performance on minority classes. In particular, macro F1 treats each toxicity label equally and is sensitive to improvements on rare labels, while micro F1 reflects overall performance across all label instances. Accuracy is reported for completeness but is not used as the primary model selection criterion.

2.4 Comment Length and Linguistic Patterns

Clean comments are generally shorter and contain more standard English vocabulary, while toxic comments often include:

- Profanity and slur variants (lexical diversity).
- Misspellings and obfuscated forms (e.g., “id1ot”, “f”).
- Higher n-gram specificity, benefiting TF-IDF linear models.

2.5 Motivation for Downsampling

Given the dominance of clean comments, we created a downsampled version of the dataset by selecting 20,000 clean comments and including *all* toxic ones. Preliminary experiments showed that downsampling improved macro-F1 for classical models without harming interpretability.

2.6 Additional Quantitative Analysis

We performed a more detailed quantitative analysis of the labels and comment statistics. Using the training split of 159,571 comments, we computed the frequency of each toxicity label and an auxiliary indicator *any_toxic*, which is 1 if any of the six labels is positive for a given comment. The vast majority of comments are non-toxic under this definition, confirming the strong imbalance suggested by the raw label counts. The most frequent labels are *toxic*, *obscene*, and *insult*, while *severe_toxic*, *threat*, and *identity_hate* occur in only a few hundred examples each in our validation split. We also measured comment length in words and characters. The median comment length is on the order of a few dozen words, with most comments shorter than roughly 50 words but a long tail of very long comments. This heavy-tailed length distribution motivated us to use bag-of-words style TF–IDF features with a capped vocabulary size. Finally, we examined label co-occurrence by computing a 6×6 co-occurrence matrix. As expected, *severe_toxic* almost always appears together with *toxic*, and there is substantial overlap between *obscene* and *insult*. These dependencies suggest that the labels are not independent, which is important when interpreting per-label performance and choosing appropriate evaluation metrics.

3 Classical Baselines Using TF-IDF

We implemented three classical machine learning models trained on TF-IDF features:

1. Logistic Regression (LR)
2. Linear Support Vector Machine (SVM)
3. Multinomial Naive Bayes (NB)

All models were wrapped using One-vs-Rest classification to accommodate the multi-label setting.

3.1 TF-IDF Feature Representation

We used a TF-IDF representation with up to 100,000 features, including uni- and bi-grams:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot \log\left(\frac{N}{\text{df}(t)}\right)$$

This representation is well-suited for toxicity detection because many toxic expressions are highly lexical and n-gram specific. In particular, short phrases, profanity variants, and character-level obfuscations often serve as strong indicators of toxic intent. TF–IDF emphasizes such discriminative terms by down-weighting common words while preserving rare but informative n-grams, making it an effective baseline for sparse text classification tasks [3, 9].

3.2 Downsampling Effects

Downsampling clean comments improved both macro and micro F1 for LR and SVM, as illustrated in Figure 2. The improvement reflects higher recall on minority toxic classes, consistent with theoretical expectations for imbalanced multi-label datasets.

We observed:

- Accuracy decreases slightly (expected due to fewer clean examples).
- Macro-F1 improves (better balance across rare labels).
- Micro-F1 remains competitive.

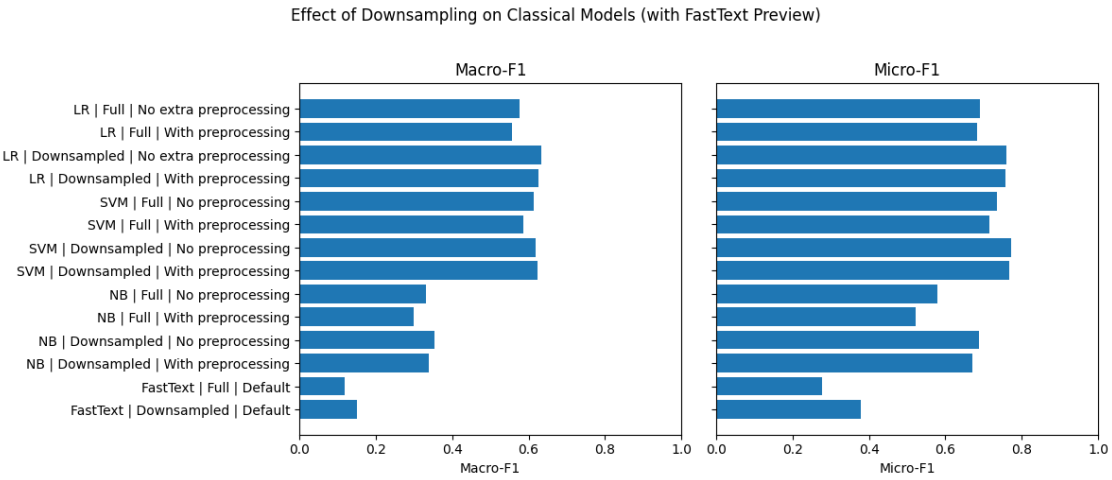


Figure 2: Effect of downsampling on macro and micro F1 for classical models.

3.3 Hyperparameter Tuning

We tuned the following hyperparameters:

- Logistic Regression and SVM regularization parameter $C \in \{0.1, 0.5, 1.0, 3.0, 10.0\}$
- Naive Bayes smoothing parameter $\alpha \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$

Hyperparameter tuning was performed on the *full dataset* to avoid instability introduced by downsampling. The selected parameters were then applied consistently to both the full and downsampled training conditions. Across models, $C = 1.0$ and $\alpha = 0.1$ provided strong overall performance in terms of both micro and macro F1, serving as stable reference settings for subsequent experiments.

3.4 Implemented TF–IDF Models and Results

In our implementation, we focused on TF–IDF features with unigrams and bigrams, a maximum vocabulary size of 100,000, English stopword removal, and a minimum document frequency of three. On top of this representation, we trained one-vs-rest logistic regression and linear SVM classifiers using the scikit-learn library with `class_weight="balanced"` to compensate for label imbalance.

For logistic regression, we systematically tuned the regularization parameter $C \in \{0.1, 0.5, 1.0, 3.0, 10.0\}$ on an 80/20 train–validation split (stratified on the `any_toxic` indicator). Table 1 summarizes the mean ROC–AUC and macro F1-score across the six labels on the validation set:

C	Mean ROC–AUC	Macro F1
0.1	0.9753	0.5002
0.5	0.9776	0.5324
1.0	0.9777	0.5500
3.0	0.9763	0.5741
10.0	0.9722	0.5878

Table 1: Validation performance of our logistic regression model for different values of C.

As C increases from 0.1 to 1.0, both mean ROC–AUC and macro F1 improve, indicating that the model is initially underfitting. The highest mean ROC–AUC (around 0.978) is obtained near $C = 0.5$–1.0. When we increase C further to 3.0 and 10.0, the mean ROC–AUC slightly decreases, suggesting mild overfitting, but the macro F1-score continues to increase up to approximately 0.59. This reflects a better precision–recall trade-off at the fixed decision threshold of 0.5, especially for rare labels, and we therefore select $C = 10$ as our final logistic regression model.

With $C = 10$, the logistic regression model achieves a macro F1-score of about 0.59 on the validation set. The frequent labels (`toxic`, `obscene`, `insult`) obtain F1-scores around 0.75, 0.78, and 0.69, respectively, while the rare labels (`severe_toxic`, `threat`, `identity_hate`) are more challenging, with F1-scores between approximately 0.40 and 0.47.

We also trained a one-vs-rest linear SVM on the same TF–IDF features. This model achieved a macro F1-score of approximately 0.585 on the validation set, which is very close to the logistic regression model with $C = 10$. In our experiments, logistic regression slightly outperformed the SVM, confirming that well-regularized linear models are strong baselines for this dataset.

3.5 Naive Bayes Baseline

We also evaluated a Multinomial Naive Bayes (NB) classifier as a lightweight probabilistic baseline using the same TF–IDF feature representation. Naive Bayes is commonly used in text classification due to its computational efficiency and robustness in high-dimensional sparse feature spaces [6].

Despite these advantages, the conditional independence assumption made by Naive Bayes is poorly suited to toxic comment classification, where meaningful phrases and contextual dependencies between words are often critical. As a result, while NB achieved high accuracy on the full dataset by correctly classifying the majority of non-toxic comments, its macro F1-score was substantially lower than those of Logistic Regression and SVM (as illustrated in Figure 2), indicating weak performance on minority toxicity labels.

Downsampling improved NB's macro F1 slightly by increasing exposure to toxic examples, but overall performance remained limited compared to discriminative linear models. These results confirm that while Naive Bayes provides a useful computational baseline, it is insufficient for capturing the nuanced lexical and contextual patterns required for effective multi-label toxicity detection.

4 Additional Non-linear Model: SVD + Random Forest

To explore a non-linear ensemble method in addition to linear models, we combined TF–IDF features with Truncated Singular Value Decomposition (SVD) and a Random Forest classifier. The goal of this experiment is to investigate whether adding non-linearity on top of a low-dimensional embedding can improve performance on toxic comment classification.

4.1 Truncated SVD Representation

Directly training a Random Forest on the full 100,000-dimensional sparse TF–IDF vectors would be computationally expensive and potentially ineffective. Instead, we first applied Truncated SVD (Latent Semantic Analysis) to reduce the TF–IDF features to 100 dense components. This low-rank representation captures coarse semantic structure in the comments while making the feature space more manageable for tree-based models.

4.2 Random Forest Classifier and Results

On top of the 100-dimensional SVD features, we trained a one-vs-rest Random Forest classifier with 200 trees. On the validation set, this SVD + Random Forest pipeline achieved a mean ROC–AUC of approximately 0.91 and a macro F1-score of about 0.40. For frequent labels such as `toxic`, `obscene`, and `insult`, the model attains F1-scores in the range 0.59–0.71, but for rare labels like `threat` and `identity_hate`, the F1-scores drop to around 0.10–0.17.

Compared to our best logistic regression model (macro F1 $\approx$ 0.59, mean ROC–AUC $\approx$ 0.98) and the linear SVM (macro F1 $\approx$ 0.585), the SVD + Random Forest model performs substantially worse, especially on minority classes. This suggests that, for high-dimensional sparse text data, linear classifiers with TF–IDF features and appropriate regularization are better aligned with the structure of the data than tree-based ensembles, even when the latter are combined with dimensionality reduction.

5 FastText Baseline

FastText [4] offers a mid-level baseline between bag-of-words models and deep neural models. We constructed FastText-compatible training files with multi-label prefixes and trained using 100-dimensional embeddings and word n-grams.

5.1 Challenges and Observations

- Multi-label FastText requires label prefix engineering (e.g., `__label__toxic`).
- The overwhelming majority of clean labels caused the model to predict "clean" too often.
- Downsampling improved performance but still lagged significantly behind TF-IDF + SVM/LR.

FastText failed to sufficiently capture the fine-grained lexical structure that TF-IDF models exploit, leading to substantially lower macro-F1 scores.

6 Transformer Models

6.1 BERT

We fine-tuned `bert-base-uncased` [2] for multi-label classification using a sigmoid output layer and binary cross-entropy loss. We evaluated BERT under both full and downsampled training conditions, and compared batch sizes of 16 and 256. In both settings, a batch size of 256 yielded more stable training and stronger overall performance, and was therefore selected for reporting.

For the full dataset, BERT was trained for three epochs, which was sufficient to reduce the training loss below 0.04. Under this configuration, the model achieved a macro F1-score of 0.643 and a micro F1-score of 0.794, with an exact-match accuracy of 0.929. Performance was strongest for frequent labels such as `toxic`, `obscene`, and `insult`, while rarer labels such as `threat` and `identity_hate` remained more challenging.

We also trained BERT on the downsampled dataset to assess the effect of reducing class imbalance. Using the same batch size of 256, the model was trained for six epochs to ensure convergence. The best downsampled configuration achieved a macro F1-score of 0.636 and a micro F1-score of 0.752, with an exact-match accuracy of 0.903. While downsampling improved minority-label exposure, BERT performed slightly better overall when trained on the full dataset, suggesting that the model benefits from larger-scale contextual diversity. Table 2 summarizes the effect of batch size and training data size on BERT performance.

Training Setting	Batch Size	Macro-F1	Micro-F1
Full Dataset	16	0.667	0.789
Full Dataset	256	0.643	0.794
Downsampled Dataset	256	0.636	0.752

Table 2: Effect of batch size and training data size on BERT performance.

6.2 RoBERTa

We fine-tuned `roberta-base` [5] for multi-label classification using the same architecture and loss formulation as BERT. Based on prior batch-size experiments with BERT, all RoBERTa experiments were conducted using a batch size of 256.

On the full dataset, RoBERTa was trained for three epochs, reaching a final training loss comparable to BERT. Under this setting, the model achieved a macro F1-score of 0.635 and a micro F1-score of 0.783, with an exact-match accuracy of 0.922. Overall performance was slightly lower than BERT on the same dataset, particularly in terms of micro F1.

In contrast to BERT, RoBERTa benefited more from downsampling. When trained on the downsampled dataset for three epochs, RoBERTa achieved a higher macro F1-score of 0.667, indicating improved balance across minority toxicity labels. However, this came at the cost of reduced micro F1 and exact-match accuracy compared to full-dataset training.

These results suggest that RoBERTa is more sensitive to class imbalance than BERT, exhibiting improved minority-label performance under downsampling but weaker overall performance when trained on the full dataset. As a result, BERT remains the stronger standalone transformer model in our experiments.

7 Ensemble Model

To leverage the complementary strengths of classical and transformer-based models, we constructed an ensemble combining TF–IDF Logistic Regression, TF–IDF Linear SVM, and BERT. Logistic Regression and SVM were trained on the downsampled dataset to improve minority-label recall, while BERT was trained on the full dataset (with a batch size of 256) to exploit richer contextual information. All models produced per-label probability estimates, which were combined using uniform probability averaging.

Individually, Logistic Regression and SVM achieved macro F1-scores of 0.580 and 0.558, respectively, while BERT achieved a substantially higher macro F1-score of 0.647. The ensemble improved upon all individual models in terms of macro F1, achieving a score of 0.667. This improvement indicates better balance across rare and frequent toxicity labels.

While the ensemble slightly reduced micro F1 and exact-match accuracy compared to BERT alone, it maintained competitive performance on these metrics, suggesting that the ensemble trades a small amount of majority-label precision for improved minority-label recall. Overall, the ensemble provides the strongest balanced performance across evaluation metrics and represents the best-performing model in this study. Table 3 summarizes the performance gains achieved by the ensemble over individual models.

Model	Macro-F1	Micro-F1	Exact Match	Per-label Acc
TF–IDF + Logistic Regression	0.580	0.657	0.860	0.969
TF–IDF + Linear SVM	0.558	0.701	0.894	0.978
BERT (Full Dataset)	0.647	0.788	0.924	0.984
Ensemble (LR + SVM + BERT)	**0.667**	0.779	0.919	0.983

Table 3: Performance comparison of individual models and the final ensemble on the validation set.

8 Conclusion

We summarize findings across classical, embedding-based, and transformer models, highlighting lessons about feature representation, class imbalance, and multi-label learning.

9 Work Distribution

Abdul Kalam Syed - Data exploration, SVM, Naive Bayes, FastText, BERT, RoBERTa, Ensemble Model, and Report.
Jiahao Xu - Data exploration, Linear Logistic Regression, SVM, SVD, Random Forest, Ensemble Model, and Report.

References

[1] Toxic comment classification challenge. `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`, 2018.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[3] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, 1998.

[4] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *EACL*, 2017.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[6] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[7] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. Hate speech detection using deep learning: A comparative study. *Applied Sciences*, 10(17):5931, 2020.

[8] Bertie Vidgen and Taha Yasseri. Detecting abusive language using contextualized word representations. In *Proceedings of the 8th International Workshop on Natural Language Processing for Social Media*, 2020.

[9] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International World Wide Web Conference*, 2017.