Toxicity Detection

Final Report

Analytics II: Machine Learning
Kayla Kim, Dongju Han, Mason Nicoletti
Dec 11, 2025

Motivation

As online platforms continue to grow, so does the amount of harmful and hateful language that appears in comment sections, which can negatively affect users and create hostile digital spaces. To better understand this issue, our project focuses on building a binary classification model that identifies whether a comment contains “toxic” speech, leading to it being rejected. Our goal is to build an effective model that can be implemented by social media platforms to flag comments for toxicity and prevent them from being posted to the platform.

Research Question: Which binary classification machine learning model can most accurately predict toxic whether a Reddit comment will be rejected for toxicity?

METHODS

Dataset

The dataset comes from the Civil Comments platform and was downloaded from Kaggle. It contains about 1.97 million observations and 46 total features. The majority of the features used in modeling are sentiment scores for a variety of reactions. We also used the column containing the text of the social media comment as well, which required separate preprocessing steps. The target variable is a binary categorical variable called rating, which shows whether the platform accepted or rejected the comment.

Workflow

We followed a consistent workflow for data preprocessing and model optimization across the four models we worked on. Each notebook began by reading in the training set, which was locally contained and standardized for the three of us. We then applied a data cleaning function, which compartmentalized in a separate folder for consistency and reusability. Because the comment text required specialized processing beyond the numerical features, we used Term Frequency-Inverse Document Frequency (TF-IDF) to transform the raw comment strings into numerical vectors. All the preprocessing steps were assembled with the model into pipelines. After defining hyperparameters, we fit a grid search on a split component of the training data to extract the best model. Finally, we validated the best model's performance, using metrics such as accuracy, precision, recall, f1 score, and creating a confusion matrix to visualize classification effectiveness. We used GridSearchCV from Sklearn with sample stratification to cross-validate and choose the best set of hyperparameters per model. Due to the computational intensity of processing the comment text features, we utilized Rivanna as computing resources to run our notebooks and get results.

Results

Findings

Random Forest was the best performing model. It excelled because it was able to produce a low false-negative count. Using F1 Score for the positive class (rejected comments) as the leading determinant of the best performing model, it can be seen that random forest achieved the highest score with the neural network performing

second best. Contrastingly, logistic regression produced poor F1 scores but scored higher in overall accuracy. This is because this model overwhelmingly predicted the more common negative class (approved comments), even for the comments that were rejected by the platform. This effect inflates the accuracy since the negative class contains far more data than the positive class.

In conclusion, we selected random forest as the best performing model because it was better able to predict the comments that were rejected. This best aligns with our goal of delivering a model that social media platforms can use to identify comments containing toxic speech.

Model Performance Table (True/Toxicity Detection)				
Model	F1 Score	Accuracy	Precision	Recall
Logistic Regression	0.15	0.94	0.50	0.09
SVM	0.25	0.93	0.44	0.18
Random Forest	0.3406	0.8582	0.2453	0.5570
Neural Network	0.2867	—	—	—

Test Set Performance

Final Model	F1 Score	Accuracy	Precision	Recall
Random Forest	0.33	0.82	0.22	0.67

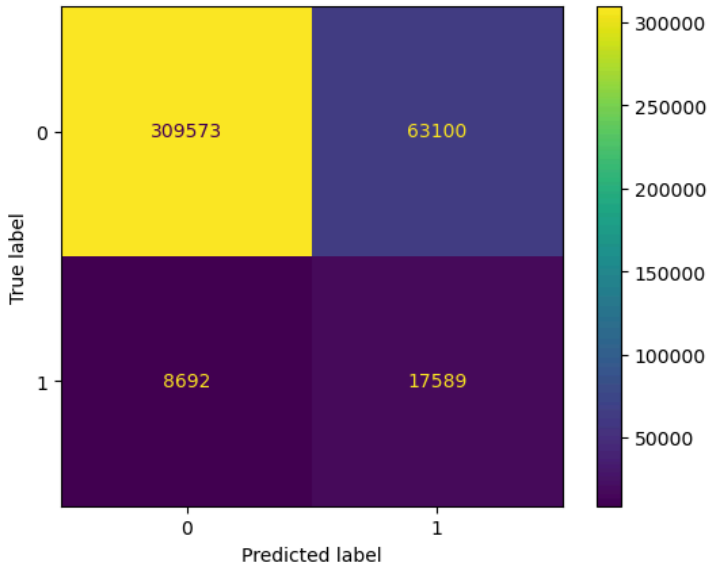

Fig. Confusion Matrix with Random Forest on the Test Set

Discussion

There were several key limitations in this project. First, the immense size of the dataset created many computational constraints. In order to run the models, even using massive computational resources such as Rivanna, it was necessary to split the data into smaller subsets. This issue risks reducing generalizability and performance consistency across the models. Another limitation was the natural class imbalance of this dataset, with approved non-toxic comments (93%) being far more common than rejected comments (7%). This contrast affected minority-class prediction in several of the models, leading to lower F1 scores. Finally, the Jigsaw dataset includes sensitive identity attributes. Without fairness metrics, it is unclear how well models treat identity-related comments, and unintentional bias remains possible.

Future work could involve implementing transformer-based models, which have the potential to perform significantly better at capturing contextual or semantic nuances inherent in toxic language.