CS 419 Course Project: Toxic Comments Classification using NLP

Team ToxBusters
Indian Institute of Technology Bombay
Nilesh (22b1541), Pranav (22b0661), Aman (22b0321) , Amit (23b2482), Jasmine (22b2727)

Abstract

This project addresses the classification of toxic comments using machine learning techniques. Labeled data from Jigsaw's Toxic Comment Classification Challenge was processed through a robust pipeline, including text cleaning and TF-IDF feature extraction. Various algorithms were evaluated, including Logistic Regression, Naive Bayes, Decision Trees, Random Forests, Support Vector Classifier (SVC), and Voting Classifier. The project incorporates visualizations like confusion matrices and ROC curves to interpret results, highlighting machine learning's potential in scalable content moderation systems.

1 Introduction

Toxic comments on online platforms harm user experience and necessitate automated detection systems. This project focuses on classifying toxic comments from Jigsaw's Toxic Comment Classification Challenge into categories like general toxicity and hate speech. Preprocessing included text cleaning, tokenization, and TF-IDF vectorization. Machine learning models such as Logistic Regression, Naive Bayes, Decision Trees, Random Forests, SVC, and a Voting Classifier ensemble were evaluated. The Voting Classifier, leveraging multiple algorithms, and SVC, known for its robustness with high-dimensional data, emerged as the most effective solutions for scalable content moderation.

2 Dataset and NLP preprocessing

Train Dataset: Contains labeled text data with categories such as toxic, severetoxic, obscene, threat, insult, and identityhate. Each comment is assigned binary labels for these categories, indicating the presence or absence of toxicity.
Test Dataset: Consists of comment texts with corresponding IDs and true toxicity category labels provided in a separate file (test_labels.csv).

Link to the Dataset
Text preprocessing is an important step in NLP to clean and prepare raw text for analysis. It involves tasks like removing special characters, converting all text to lowercase, splitting text into smaller parts (tokenization), and removing common words like "the" or "and" (stopwords). These steps make the text easier for machine learning models to understand and process, improving the accuracy of predictions.

3 Methodology

3.1 Data Preprocessing

The text data was cleaned by removing special characters, converting to lowercase, and tokenizing. Stopwords were removed, and TF-IDF vectorization was applied to convert text into numerical features.

3.2 Model Training

We trained various machine learning models on processed data which were then used to classify comments:

Logistic Regression: It performs well on high-dimensional data like text features from TF-IDF vectorization and is computationally efficient, allowing quick experimentation.

Naive Bayes: Naive Bayes is specifically suited for text data due to its probabilistic nature and ability to handle high-dimensional feature spaces. It works well with sparse data and is particularly effective for handling imbalanced datasets, a common issue in toxic comment classification.

Decision Trees: They provide an interpretable framework by breaking down decisions into simple rules. While they may overfit complex datasets, they offer insights into how the model makes predictions, which can be useful for understanding feature importance.

Random Forests: As an ensemble method, Random Forests overcome the overfitting issues of individual decision trees by averaging multiple tree predictions. They provide a balance between interpretability and performance, making them a robust choice for general-purpose classification.

Support Vector Classifiers (SVCs): A kernel-based method that finds the optimal hyperplane for classification, particularly effective for high-dimensional data.

Voting Classifier: An ensemble approach that combines the predictions of multiple models (e.g., SVC, Random Forests, Logistic Regression) to improve overall accuracy and robustness.

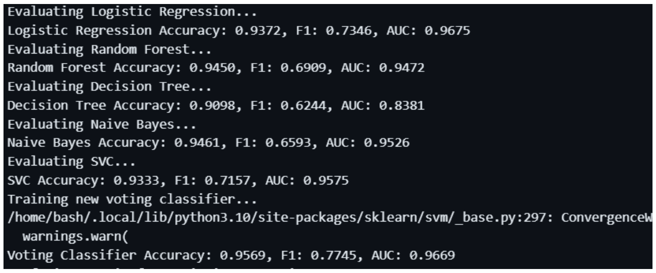
3.3 Evaluation

Each model was evaluated on the training and validation datasets using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. A confusion matrix was used to visualize true positives, false positives, true negatives, and false negatives, while ROC curves were used to assess the trade-off between sensitivity and specificity.

At the end we tested our model on a test dataset and saved the results of predictions as toxic_predictions.csv

4 Results

Results after training are as follows:



```
Evaluating Logistic Regression...
Logistic Regression Accuracy: 0.9372, F1: 0.7346, AUC: 0.9675
Evaluating Random Forest...
Random Forest Accuracy: 0.9450, F1: 0.6909, AUC: 0.9472
Evaluating Decision Tree...
Decision Tree Accuracy: 0.9098, F1: 0.6244, AUC: 0.8381
Evaluating Naive Bayes...
Naive Bayes Accuracy: 0.9461, F1: 0.6593, AUC: 0.9526
Evaluating SVC...
SVC Accuracy: 0.9333, F1: 0.7157, AUC: 0.9575
Training new voting classifier...
/home/bash/.local/lib/python3.10/site-packages/sklearn/svm/_base.py:297: ConvergenceW
  warnings.warn(
Voting Classifier Accuracy: 0.9569, F1: 0.7745, AUC: 0.9669
```

Figure 1: Accuracy, F-1 score, AUC

As we can seen, the Voting Classifier is the best-performing model in this setup, delivering superior accuracy and F1-Score while maintaining a strong AUC. For standalone models, SVC is a close second, excelling in balanced performance across all metrics. Decision Trees, on the other hand, perform poorly and are less suitable for this task.

While the Voting Classifier provides the best results, it requires more computational resources compared to simpler models like Logistic Regression or Naive Bayes.

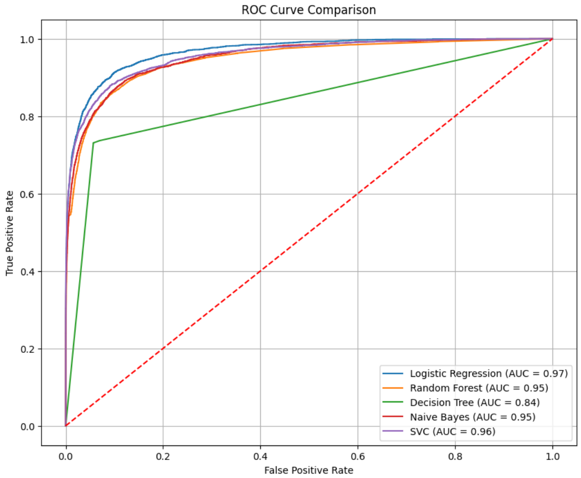


Figure 2: ROC curves comparison

Conclusion

This project successfully classified toxic comments using various machine learning models, with the Voting Classifier achieving the best performance, including an accuracy of 95.69%, F1-Score of 0.7745, and AUC of 0.9669. Among individual models, SVC performed strongly with an accuracy of 93.33% and an AUC of 0.9575, showcasing its robustness in handling high-dimensional text data. Simpler models like Logistic Regression and Naive Bayes provided competitive results, with accuracies of 93.72% and 94.61%, respectively, while Decision Trees lagged with an accuracy of 90.98%. These results highlight the importance of ensemble methods like the Voting Classifier for balancing precision, recall, and overall generalization, making them highly effective for complex classification tasks like toxic comment detection.

References

[1]

[Udaykiran *et al.*, 2021] Udaykiran Goud Nallabolu *Classification of Online Toxic Comments Using Machine Learning Algorithms* . B. Tech, Vignan Institute of Technology and Science, 2021

[2] Group Project for MSDS621 Machine Learning at University of San Francisco

[3] YouTube Playlist followed for learning NLP basics