**Final Project Technical Report**

**AAI 501 – Applied Artificial Intelligence**

**University of San Diego**

**Toxic Comment Classification Using Machine Learning**

**Faculty Advisor:** **Azka Azka**

**Team Members:**

**Group 3**

Anugrah Rastogi

Dhrub Satyam

Mallesham D

# Table of Content

[Table of Content](#_serqozknblvz)

[Abstract](#_2uekmc39kmlu)

[1. Introduction](#_pqril5wylvem)

[2. Dataset and Preprocessing](#_m3fn1enc9h4f)

[2.1 Data Set](#_wecchjfs1oq9)

[2.2 Class Distribution Analysis](#_d1vnv9npuir)

[2.3 Text Preprocessing Pipeline](#_47xq5shojsil)

[3. EDA](#_v6gvncmf74nn)

[3.1 Label Distribution Analysis](#_5jzdp8pmwjmh)

[3.2 Multi-Label Co-Occurrence and Correlation](#_dhgftkrznr2l)

[3.3 Analysis of Label Cardinality](#_sw75w70z5sz)

[3.4 Text Length Distribution](#_jmq5arazj67h)

[3.5 Word Cloud Visualization](#_nbkz2beu4x93)

[3.6 Key Observations](#_5l296r5b9uh)

[4. Model Training and Evaluation](#_hg2jvv6z7rw3)

[4.1 Modeling Objective](#_92hhjwio4fj1)

[4.2 Data Preparation and Splitting](#_w727zqsbkn5w)

[4.3 Feature Extraction](#_5fpn0zl9kc7b)

[4.4 Model Training: One-vs-Rest Logistic Regression](#_rmh0tvu0crp1)

[4.5 Evaluation of Metrics](#_8pw9di6kijh3)

[4.6 Metric Visualization](#_4a1u6m8wo52v)

[4.7 Interpretation and Recommendations](#_2pfkwkh3ef4u)

[5. Model Selection](#_53rsc5lamja)

[5.1 Multinomial Naive Bayes](#_9gbxz9yxa0vv)

[5.2 Deep Learning Approach](#_lp5ou9j9x7zw)

[6. Discussion and Insights](#_g9xqq7x8p0x1)

[7. Conclusion](#_fb871qxjp7jp)

[Appendix (Code along with Output)](#_v0sgc06chzgx)

[Appendix A (Data Preparation and Analysis)](#_6v0rcd56c2jx)

[Appendix B EDA(Exploratory Data Analysis)](#_vjb0sgkl7pdz)

[Appendix C TF-IDF features and Logistic Regression](#_4esvab9yjwjr)

[Appendix D Model Training](#_iavqa7xswpkq)

[References](#_2xzixs4jmqze)

# 

# Abstract

With the continuous expansion of online environments, content created by users has become an essential component of modern digital interaction. However, this accessibility has also contributed to the emergence of negative behaviors such as abusive language, harassment, and hate-driven discourse. Manual moderation of such content is not only labor-intensive but also inefficient at scale. This study develops a machine learning framework to detect and categorize harmful online comments. The categories include obscenity, insults, threats, toxicity, severe toxicity, and identity-based hate. The work uses data from the publicly available Jigsaw Toxic Comment Classification Challenge. We evaluated and compared model performance by cleaning the data, performing exploratory analysis, and deploying various classification models, using accuracy, F1-score, and AUC-ROC as metrics. The outcome demonstrates the feasibility of using AI systems to support content moderation and promote healthier digital environments.

# 1. Introduction

Today’s online environment includes spaces like blogs, discussion boards, and social networking sites, which serve as common venues for conversation and content exchange. However, alongside the benefits of open communication comes the challenge of managing harmful interactions, including abusive, offensive, and threatening language. Addressing such behavior - especially at scale - requires intelligent systems capable of identifying and flagging inappropriate content. This project applies machine learning algorithms to build a classifier which can identify various types of toxic language. These include general toxicity, threats, obscenity, insults, severe toxicity, and identity-based hate. We trained the model using labeled comment data (Rastogi, Satyam, & Devasane, 2025).

# 2. Dataset and Preprocessing

This part outlines the dataset employed for the toxicity comment classification task and outlines the structured preprocessing approach implemented to ready the data for later analysis and model construction (Kaggle, 2023).

## 2.1 Data Set

The dataset utilized in this report originates from “the Jigsaw Toxic Comment Classification Challenge” and contains user-submitted remarks sourced from Wikipedia discussion pages (Jigsaw, 2018). Each comment carries one or more tags indicating harmful or offensive content, allowing multi-label classification because a single entry may exhibit multiple forms of toxicity simultaneously.

Each data record includes the following attributes:

* **toxic**: Marks general instances of toxic behavior
* **severe\_toxic**: Identifies highly aggressive or harmful language
* **obscene**: Flags profanity or vulgar content
* **threat**: Detects threatening or violent comments
* **insult**: Labels comments containing personal attacks or rudeness
* **identity\_hate**: Denotes hate speech directed toward identity-based groups

A binary column named Clean was added to separate non-toxic content. Comments without any toxic labels receive a value of 1, while those containing at least one toxic label are assigned a value of 0.

**Dataset Summary:**

* **Total comments**: 159,571
* **Clean comments**: Approximately 144,000 (~90% of the dataset)
* **Most common label**: toxic
* **Least frequent labels**: threat and identity\_hate

This dataset’s structure and labeling schema are well-suited for training multi-label classification models that must identify overlapping forms of toxicity in online content.

## 2.2 Class Distribution Analysis

An initial frequency analysis of label occurrences revealed a pronounced class imbalance. Clean comments make up the majority share of all data points. In contrast, some toxic classes - especially *threat* and *identity\_hate* - appear far less frequently. For example, clean comments make up over 89% of the data, whereas fewer than 0.5% of the records are marked as threats. This uneven distribution could skew model performance, favoring the majority class and reducing sensitivity to minority classes. Consequently, addressing this imbalance through class weighting, synthetic sampling, or other strategies becomes critical for building an equitable and reliable model.

## 2.3 Text Preprocessing Pipeline

The raw text underwent a sequence of preprocessing steps to make it ready for machine learning tasks.

* **Lowercasing:** All text was converted to lowercase form so that tokens with different cases were treated the same.
* **Removing Special Characters:** Non-letter elements, including numbers, symbols, and punctuation marks, were stripped out using regular expressions.
* **Tokenization:** The cleaned text was split into words for structured phase.
* **Stopword Elimination:** Common English words with little meaning, such as "the", "and", and "in" were removed using the NLTK stopword list (Bird, Klein, & Loper, 2009) to minimize noise.
* **Lemmatization:** Words are changed to their root form to ensure consistency. For example, "running" and "ran" were both reduced to "run."
* **Handling Missing Data:** Any entries without valid comment text were dropped to keep the dataset accurate and complete.

These preprocessing steps helped standardize and sanitize the input text, ensuring it was noise-free, well-organized, and ready for subsequent feature extraction and model training.

After preprocessing, each comment had a standardized cleaned version (clean\_text) and a binary flag indicating whether it was completely non-toxic. This structured and noise-free dataset served as the input for the feature extraction and classification steps that followed.

Preprocessing the dataset offers several important benefits. It helps remove inconsistencies and unwanted characters from the text, ensuring cleaner input data. This step also improves the quality of word representations, making them more meaningful for model training. The final cleaned dataset is organized to work efficiently with both machine learning and deep learning models.

# 3. EDA

Exploratory Data Analysis (EDA) serves as a critical early phase in any data-driven modeling pipeline. It provides a deep understanding of the dataset's characteristics, uncovers patterns and anomalies, and guides the formulation of preprocessing and modeling strategies. For the toxicity comment classification problem, the dataset comprises user-generated comments labeled with one or more forms of toxic behavior, including *toxic, threat, insult, obscene,* *severe toxic,* and *identity hate*. As each comment can belong to several categories or none, the task is treated as a multi-label classification scenario.

## 3.1 Label Distribution Analysis

The first stage of the examination involved counting the number of occurrences for each toxicity category in the dataset. Among them, Toxic appeared the most, with ‘*obscene’* and ‘*insult’* following in frequency. In contrast, **'threat'** and **'identity hate'** were observed much less frequently. This revealed a pronounced **class imbalance**, where a majority of comments were either non-toxic or labeled with only one toxic category.

A bar chart visualization (Figure 1) illustrated the skewed distribution of labels. This type of imbalance can lead to biased model predictions favoring majority classes unless handled appropriately during training.

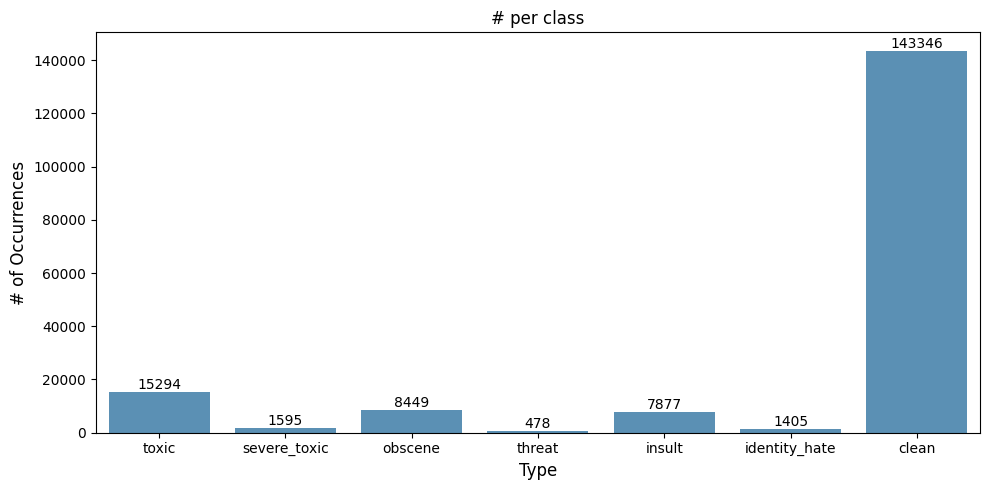


Figure 1:Distribution of Toxic Labels

## 3.2 Multi-Label Co-Occurrence and Correlation

To explore relationships among different toxic categories, a co-occurrence matrix and correlation heatmap were generated. The analysis revealed that some labels frequently appeared together. For instance, comments labeled as *severe toxic* were always accompanied by the *toxic* label, suggesting a strong dependency.

The correlation heatmap (Figure 2) using **Pearson correlation** provided a quick glance at linear associations between labels. However, as Pearson's method assumes continuous variables, its application to binary data may yield misleading results.



Figure 2: Heatmap of Label Correlations

To provide a more appropriate measure for binary categorical data, **Cramér’s V** statistic was computed for selected label pairs (Kaggle, 2023). This metric is derived from the chi-square test and quantifies the strength of association between two binary variables. It was observed, for example, that the relationship between *Toxic* and *Severe toxic* had a moderate Cramér’s V value, indicating a meaningful overlap.

## 3.3 Analysis of Label Cardinality

The count of toxicity-related labels attributed to each comment- referred to as *label cardinality* - was analyzed. Most comments had zero or only one toxic label. A minority of comments carried multiple toxicity labels, with very few examples assigned to all six.

Figure 3 visualizes the distribution of label counts per comment. This plot helps in understanding the complexity of the multi-label classification problem and highlights the need for models that can effectively handle multiple, sometimes overlapping, toxic categories.



Figure 3: Number of Toxic Tags per Comment

## 3.4 Text Length Distribution

The number of characters in each comment was plotted to assess the text length distribution. Most comments were relatively short, with a character length of under 500. A few exceptional entries contained lengths stretching into the range of several thousand characters. Understanding this distribution is vital for selecting appropriate sequence lengths in natural language processing (NLP) models.

## 3.5 Word Cloud Visualization

To explore textual content patterns, **word clouds** were generated separately for clean (non-toxic) comments and toxic ones. This provided qualitative insight into frequently used words in each group. Toxic comments often contained aggressive or profane language, while clean comments exhibited more neutral or conversational vocabulary.

These visualizations support the hypothesis that vocabulary differs significantly between toxicity and non-toxicity comments, justifying the use of word-level and character-level features for classification.

## 3.6 Key Observations

* A significant imbalance exists within the dataset, as most comments do not exhibit toxic behavior.
* Labels like *threat* and *identity hate* are severely underrepresented.
* Strong correlations exist between certain label pairs (e.g., *toxic* and *severe toxic*).
* Many comments carry only one toxic label, suggesting the need for independent binary classifiers.
* Text lengths are mostly short, enabling standard vectorization techniques without truncation.

# 4. Model Training and Evaluation

## 4.1 Modeling Objective

This phase is dedicated to constructing a preliminary model framework aimed at managing the multi-label categorization of toxic content. Each comment in the dataset may be tagged with one or more toxicity labels. Given the text of a comment, the goal is to predict all applicable labels using a machine learning approach.

The model built here utilizes **Term Frequency-Inverse Document Frequency** features for vectorizing text and applies a **Logistic Regression classifier** using the One-vs-Rest strategy.

## 4.2 Data Preparation and Splitting

The cleaned dataset, previously preprocessed and saved as “cleaned\_comments\_df”, is loaded into memory. The comment\_text field serves as the primary input variable, whereas the six toxicity categories act as the multiple target labels in the classification task.

## 4.3 Feature Extraction

TF-IDF converts raw text into numerical feature vectors, giving greater weight to distinctive terms while minimizing the influence of frequently happening words. Bi-grams are also included to capture multi-word expressions.

## 4.4 Model Training: One-vs-Rest Logistic Regression

A logistic regression algorithm is trained using the **One-vs-Rest (OvR)** strategy, which fits one binary classifier per label. This approach works well for independent multi-label classification problems.

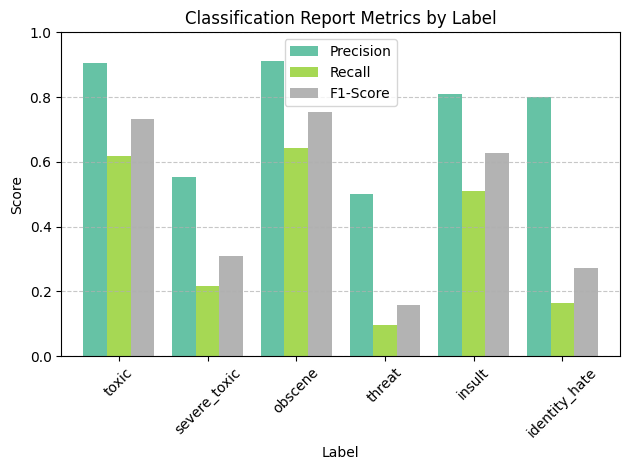
## 4.5 Evaluation of Metrics

Each label is calculated separately using accuracy, F1-score, recall, and precision. The results for all six categories are mentioned in the following table.



## 4.6 Metric Visualization

A bar chart was plotted to visualize the F1-score, recall, and precision for each class. This visual representation helps identify performance imbalances across labels.



## 4.7 Interpretation and Recommendations

The TF-IDF and Logistic Regression model serves as a strong baseline but exposes several challenges.

* **Imbalanced data**:
  + Rare labels, such as threat and identity\_hate, appear only in limited instances, which leads to reduced recall performance.
* **Simple model limitations**:
  + Linear models may struggle with complex language patterns found in toxic speech.

# 5. Model Selection

The model selection process involved evaluating multiple techniques for multilabel text classification, with a focus on balancing predictive accuracy across both frequent and rare toxicity categories.

## 5.1 Multinomial Naive Bayes

**Dataset & Feature Representation**

The dataset comprised 159,571 labeled comments, where each entry could be marked with more toxic types - Toxic, Obscene, Severe\_toxic, Insult, Threat, and Identity\_hate - plus an extra “clean” label to denote non-toxic comments. The textual content was converted into numerical feature representations through TF-IDF vectorization, applying a minimum document frequency threshold of three and considering n-grams ranging from single words to sequences of three words. This approach enabled the model to capture both individual terms and meaningful short phrases.

**Training Configuration**The data is categorized into **80% training** and **20% testing**. Multinomial Naive Bayes (MNB) was applied independently to each label, enabling separate binary classifiers to address the multi-label nature of the problem (Kaggle, 2023).

**Initial Observations**

* **Frequent classes** such as *Toxic* and *Obscene* achieved relatively strong predictive performance.
* **Less common categories -***threat*, *severe\_toxic*, and *identity\_hate*, showed reduced F1-scores, reflecting challenges in accurately detecting these instances even though their AUC values remained relatively strong.

**Hyperparameter Optimization**A grid search over smoothing parameters identified **α = 0.01** as optimal for all labels within the tested range. This adjustment improved stability and performance, particularly for the minority classes.

**Final Model Metrics**The tuned MNB models achieved:

* **Macro F1-score** (average across labels): ~0.448
* **Average AUC**: ~0.944  
   These results confirmed that the model was well-suited for high-level filtering of toxic content, though rare class detection remained a challenge.

**Deployment Context**The final configuration was capable of classifying new, unseen comments by providing independent predictions for each toxicity label, making it suitable for automated content moderation workflows.

### 

## 5.2 Deep Learning Approach

**Evaluation Summary**A deep learning approach was additionally investigated for this classification problem, with its effectiveness evaluated based on recall, F1-score, and precision metrics for each individual label.

**Key Findings from Classification Report**

* Some categories displayed high precision but lower recall, suggesting the model was conservative in labeling certain types of toxicity.
* Minority classes showed reduced F1-scores compared to frequent categories, consistent with the trends seen in the MNB model.
* The disparity between performance on common versus rare classes highlighted the dataset’s imbalance.

**Training Dynamics**Analysis of the training history revealed patterns in accuracy and loss progression.

* If accuracy plateaued while loss continued to decrease, the model showed signs of overfitting.
* Rapid convergence in early epochs suggested the model quickly captured dominant patterns, but may have underrepresented low-frequency label structures.

**Effectiveness and Limitations**The deep learning method demonstrated competitive accuracy and robust handling of common labels, but performance gains over MNB were limited for rare categories. This reinforces the need for **class imbalance strategies** (e.g., focal loss, re-weighting) in future iterations.

# 

# 6. Discussion and Insights

The comparative evaluation of the implemented models provides several key observations about the nature of the dataset, the challenges posed by multi-label classification, and the strengths and limitations of each modeling approach.

**1. Dataset Characteristics**

The dataset’s imbalance was one of the most influential factors affecting performance. High-frequency categories such as *toxic* and *obscene* consistently achieved stronger precision and recall across models, while rare categories (*threat* and *identity\_hate*) lagged behind in predictive accuracy. This imbalance encouraged models to favor the majority class, often at the expense of minority class recall.

**2. Multinomial Naive Bayes Findings**

The MNB approach delivered competitive performance for a lightweight model, with particularly strong results in frequent labels. However, the smoothing parameter α had a clear impact on performance. The optimal value of 0.01 improved stability across labels, but rare-class performance still remained below desirable thresholds. Despite this, the model’s high average AUC (~0.944) demonstrated that it was capable of ranking toxic versus non-toxic cases effectively, even if exact label predictions for minority classes were less reliable.

**3. Deep Learning Observations**

The deep learning model exhibited slightly better adaptability to feature complexity, producing competitive results for common categories. Precision remained high for certain classes, but recall suffered in low-frequency labels. Training history indicated that while the model captured frequent patterns quickly, it lacked sufficient minority-class exposure, which may have contributed to underfitting in those areas.

**4. Practical Implications**

From a deployment perspective, MNB offers faster training and inference, making it suitable for large-scale, real-time moderation systems. Deep learning models, while more resource-intensive, provide the potential for richer feature extraction—particularly if paired with contextual embeddings such as BERT.

**5. Lessons Learned and Future Directions**

The results reinforce that **class imbalance handling is essential** for any production-ready toxic comment classifier. In the future, the project can be improved in several ways. One approach is to adjust class weights during training to handle the imbalance between common and rare labels.

Another option is to use data augmentation or synthetic sampling methods to increase the number of examples for underrepresented categories. Additionally, transformer-based models could be applied to capture deeper context and subtle language patterns, leading to more accurate predictions.

Overall, the experiments demonstrate that while baseline models can perform well for dominant toxicity categories, achieving balanced performance across all labels requires both algorithmic adjustments and targeted preprocessing strategies.

# 7. Conclusion

The toxic comment classification project successfully demonstrated the use of machine learning algorithms to detect and categorize harmful online content into multiple toxicity-related labels. Using the Jigsaw Toxic Comment Classification dataset, we implemented a complete pipeline—starting from data preprocessing and exploratory analysis to feature engineering, model training, and performance evaluation.

The results show that basic models, like Multinomial Naive Bayes paired with TF-IDF features, perform well on commonly occurring classes. Deep learning models can better capture complex patterns, but they may continue to face challenges with rare categories unless specific strategies are applied. Across all approaches, class imbalance emerged as the primary challenge, leading to reduced recall for rare labels such as *threat* and *identity\_hate*.

From a practical perspective, the developed models provide a foundation for automated moderation systems that can be integrated into social media platforms, discussion forums, or other user-generated content environments. By reducing reliance on manual moderation, these models have the potential to improve online safety and reduce the spread of harmful speech.

Future work will focus on several key areas. First, the robustness of the model will be enhanced by incorporating contextual embeddings, such as BERT or RoBERTa, to better capture nuanced meanings in text. Second, class imbalance will be addressed through techniques like oversampling, synthetic data generation, or cost-sensitive learning. Third, the dataset will be expanded to include multilingual comments and domain-specific toxic expressions. Fourth, real-time deployment strategies will be implemented to ensure scalability and low-latency predictions for large-scale online platforms. Finally, explainable AI methods will be explored to improve transparency in toxicity detection, enabling moderators and end-users to understand why a comment is flagged.

Overall, the study provides important insights into both the technical and practical aspects of multi-label text classification for toxic content detection and lays the groundwork for building more sophisticated and fair moderation tools in the future.

# Appendix (Code along with Output)

## Appendix A (Data Preparation and Analysis)

**Appendix A.1 Data Preprocessing**

**Code:**

**

**Output:**

**(159571, 8)**

## Appendix B EDA(Exploratory Data Analysis)

**Appendix B.1 Labeling Clean Comments**

**Code:**

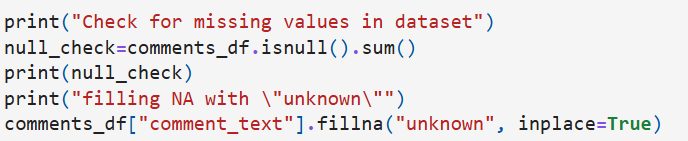
****

**Output:**

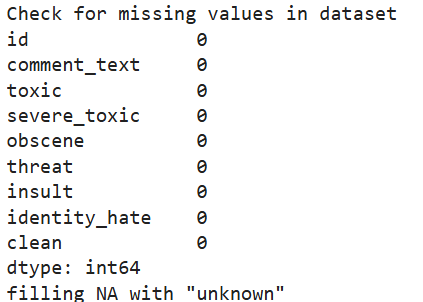
****

**Appendix B.2 Checking null Comment**

**Code:**

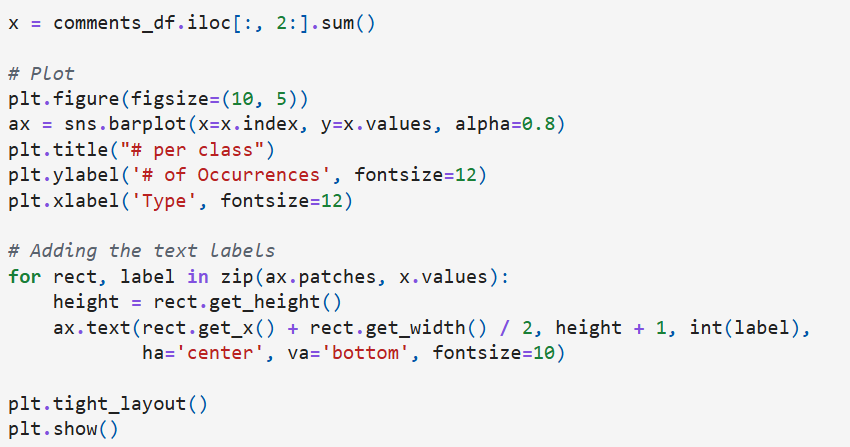
****

**Output:**

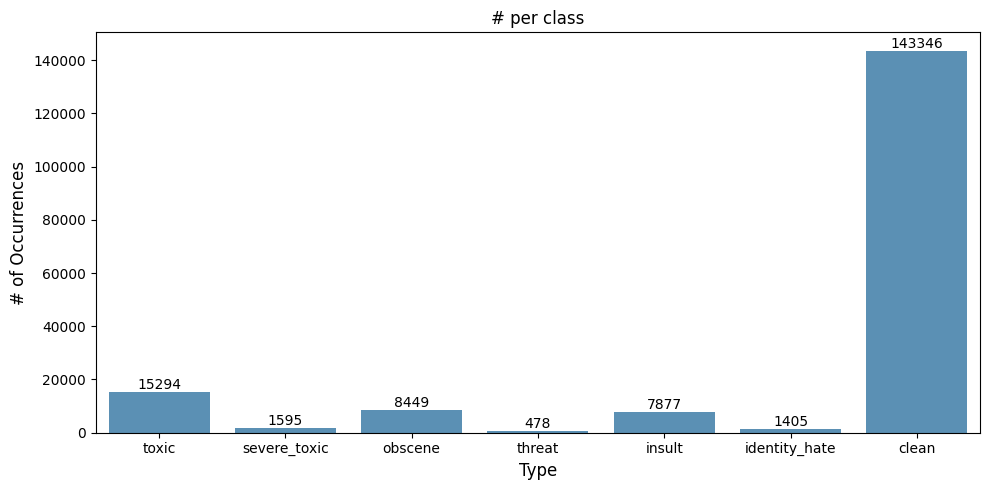
****

**Appendix B.2 Toxicity per class**

**Code:**

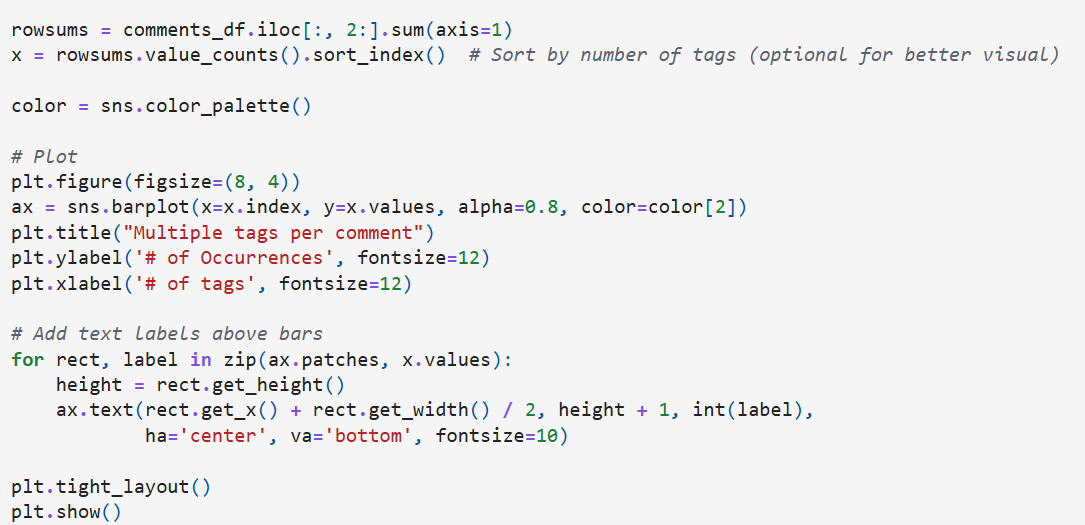
****

**Output:**

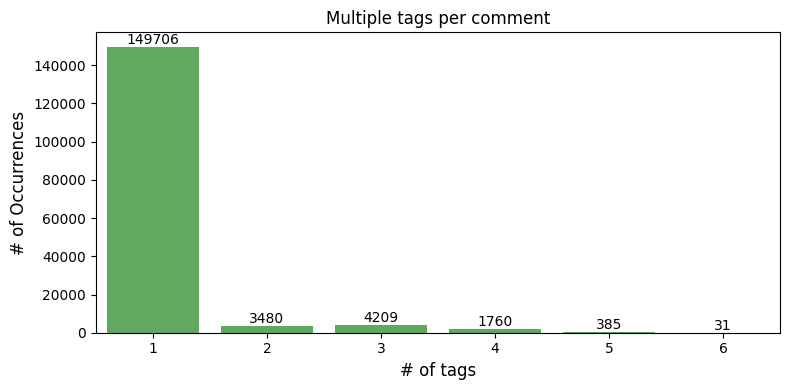
****

**Appendix B.3 Class Imbalance Analysis**

**Code:**

****

**Output:**

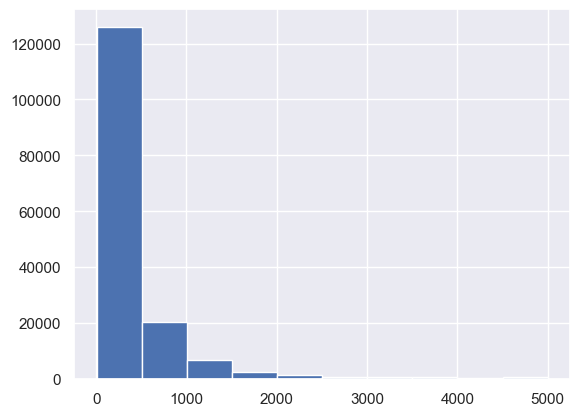
****

**Appendix B.4 Histogram plot for text length**

**Code:**

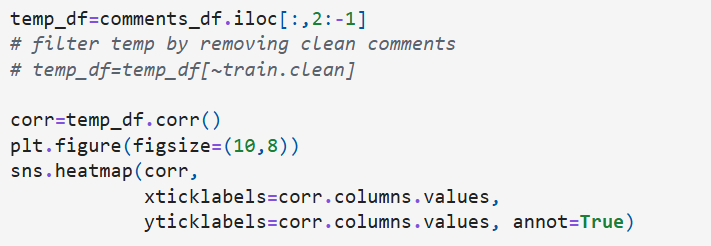
****

**Output:**

****

**Appendix B.5 correlation plot**

**Code:**

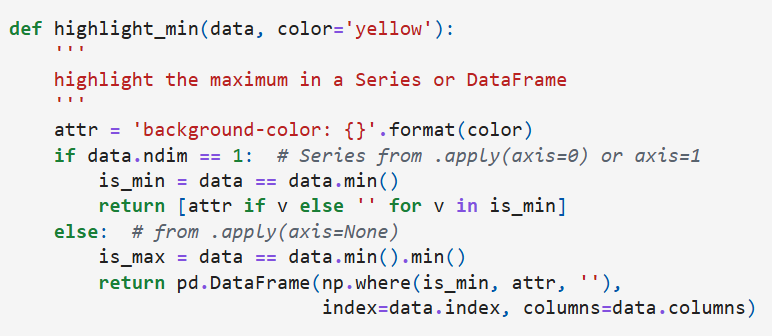
****

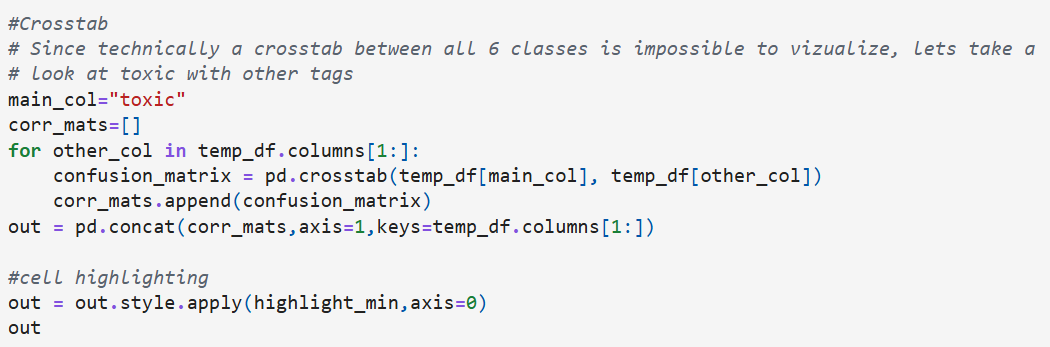
**Output:**

****

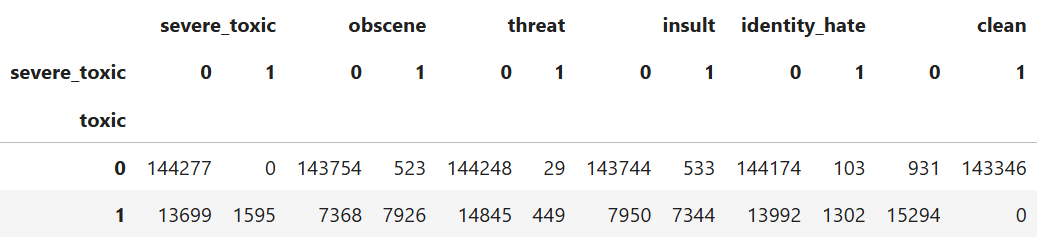
**Appendix B.6 Crosstab of Toxic comments with the other classes**

**Code:**

****

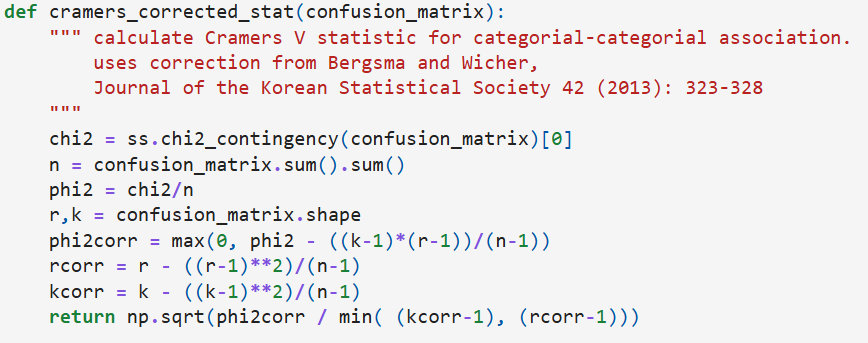
****

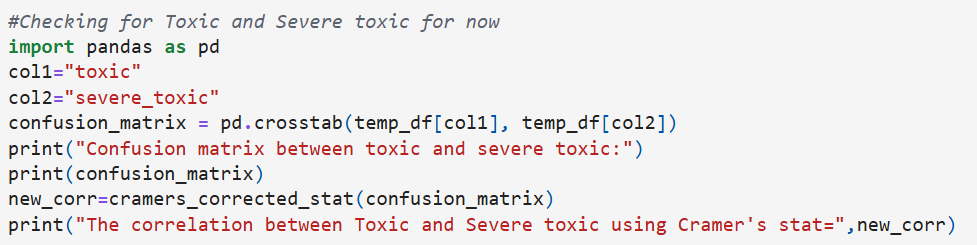
**Output:**

****

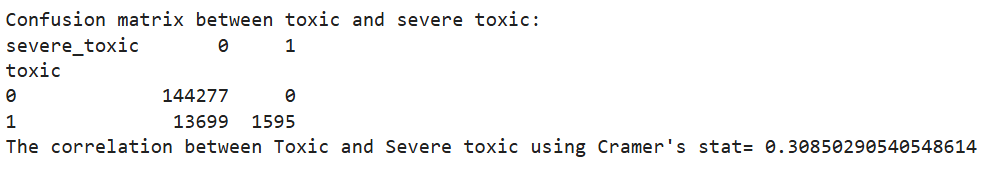
**Appendix B.7 Confusion matrix between toxic and severe toxic**

**Code:**

****

****

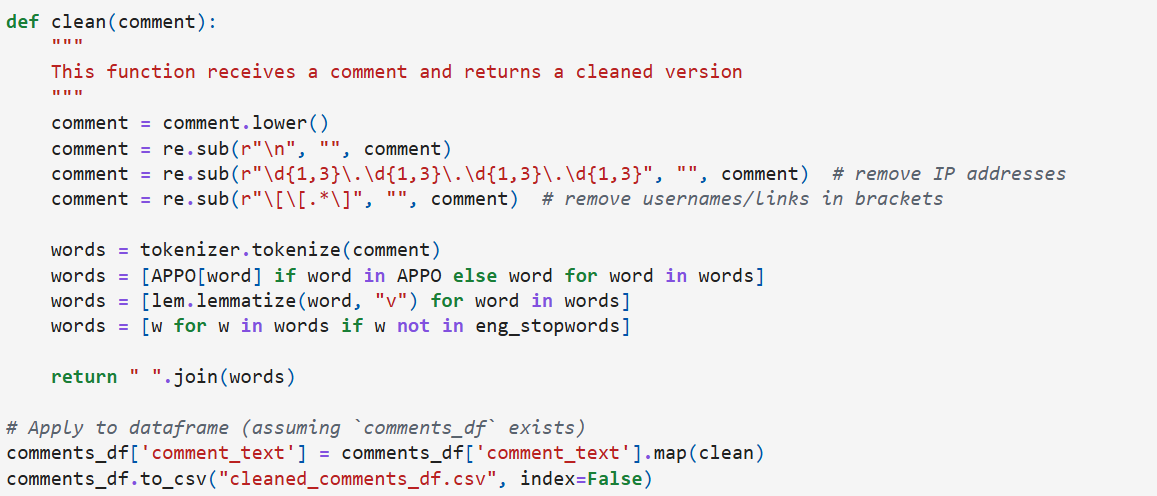
**Output:**

****

**Appendix B.8 Clean up the comment text**

**Code:**

****

****

**Output:**

**Appendix B.9 Wordclouds - Frequent words**

**Code:**

****

****

**Output:**

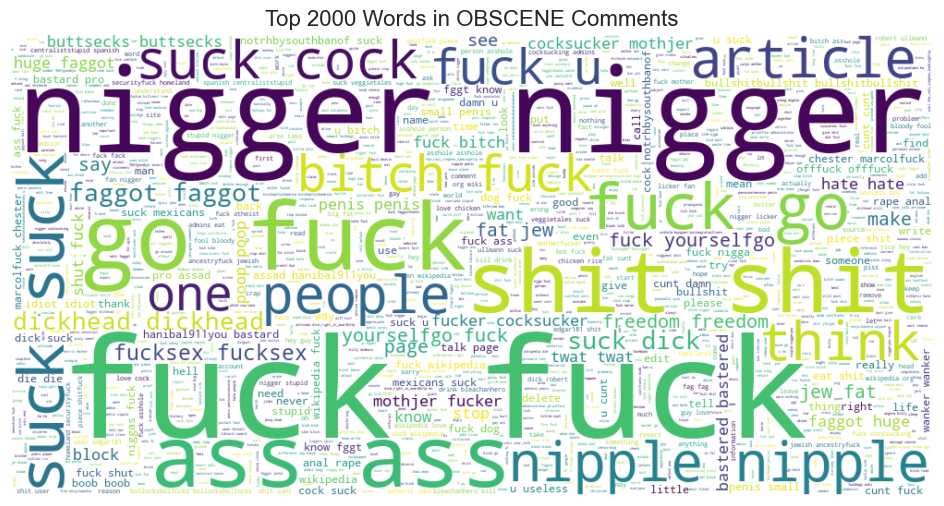
**🔍 Generating WordCloud for: toxic**

****

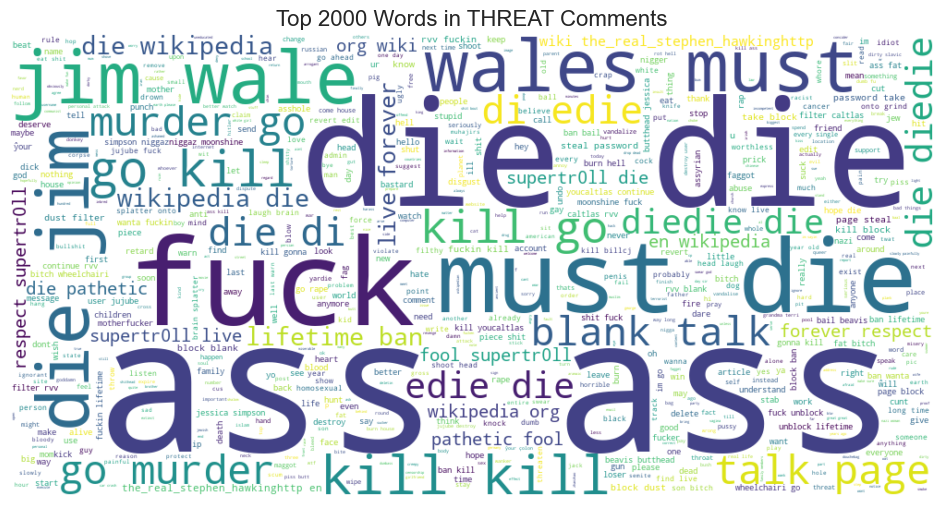
**🔍 Generating WordCloud for: severe\_toxic**

****

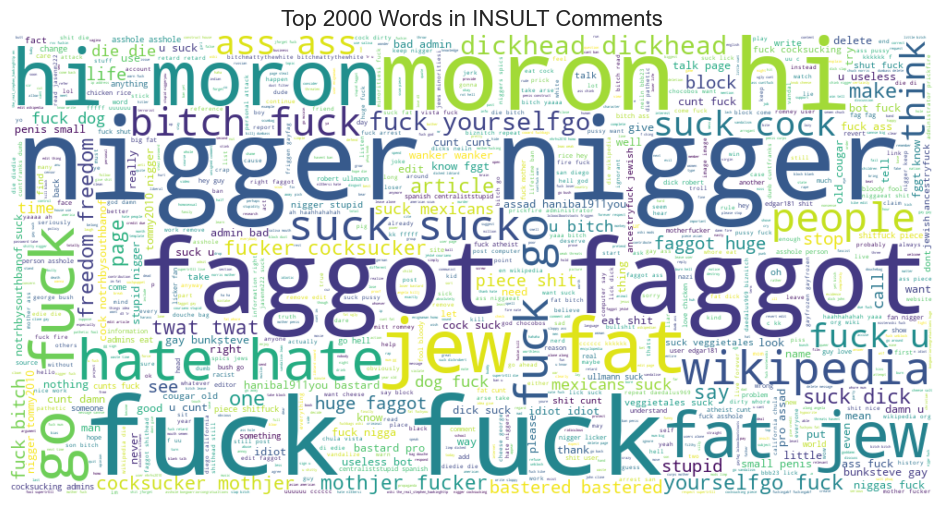
**🔍 Generating WordCloud for: obscene**

****

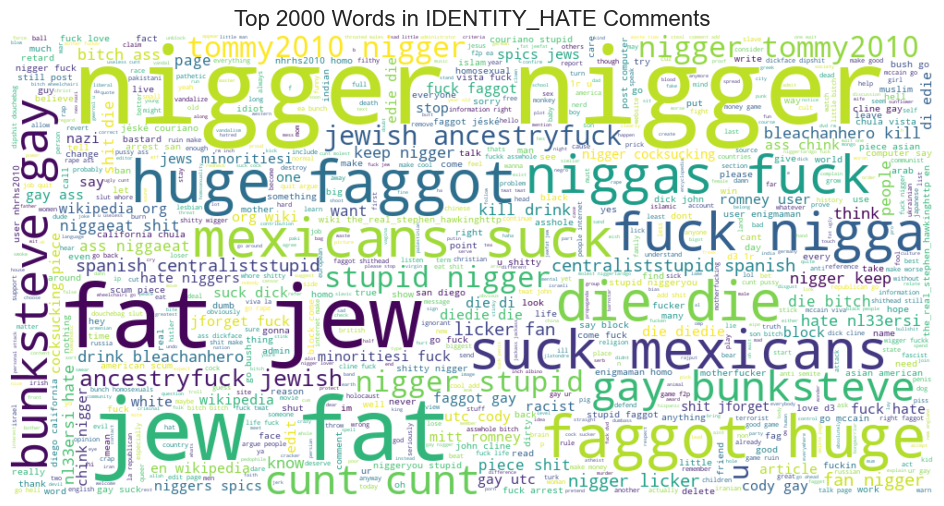
**🔍 Generating WordCloud for: threat**

****

**🔍 Generating WordCloud for: insult**

****

**🔍 Generating WordCloud for: identity\_hate**

****

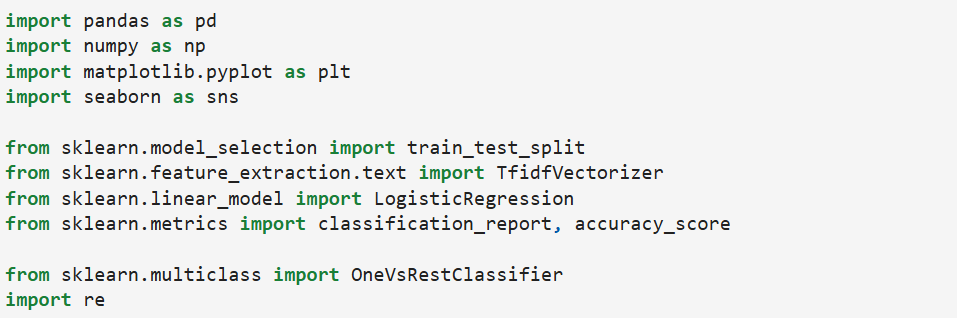
**🔍 Generating WordCloud for: CLEAN comments**

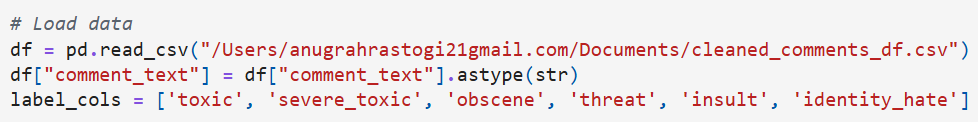
****

## Appendix C TF-IDF features and Logistic Regression

**Appendix AC.1 Load the Cleaned Data**

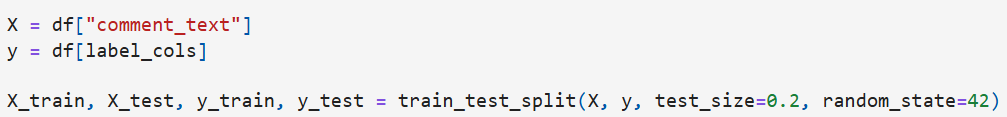
**Code Snippet:**

****

****

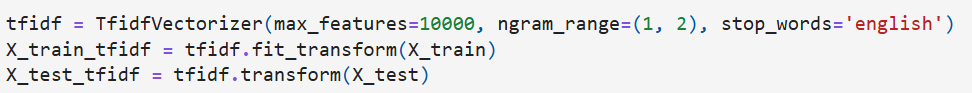
**Appendix C.2 Split the Data**

**Code Snippet:**

****

**Appendix C.3 TF-IDF Vectorization**

**Code Snippet:**

****

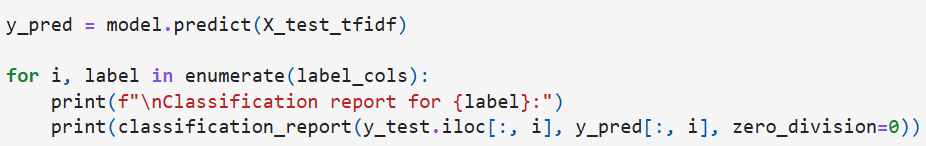
**Appendix C.4 Train One-vs-Rest Logistic Regression**

**Code Snippet:**

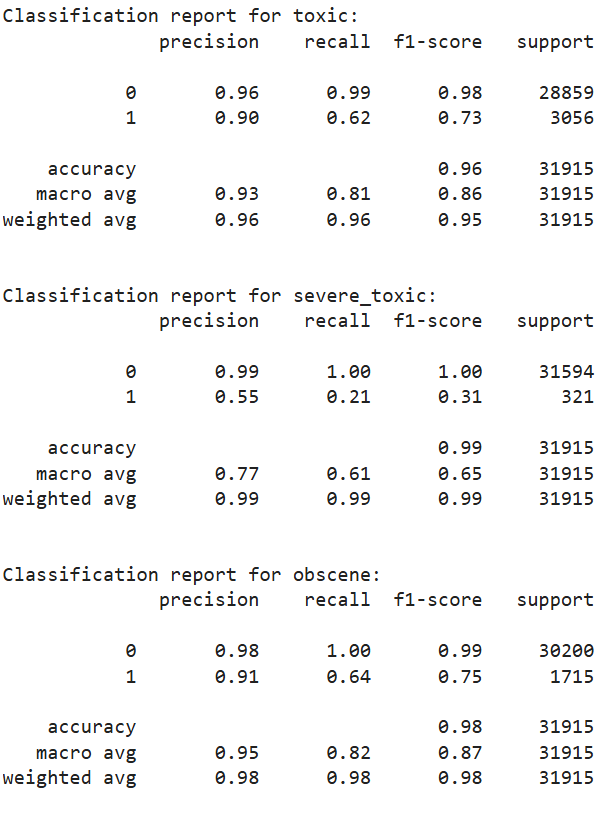
****

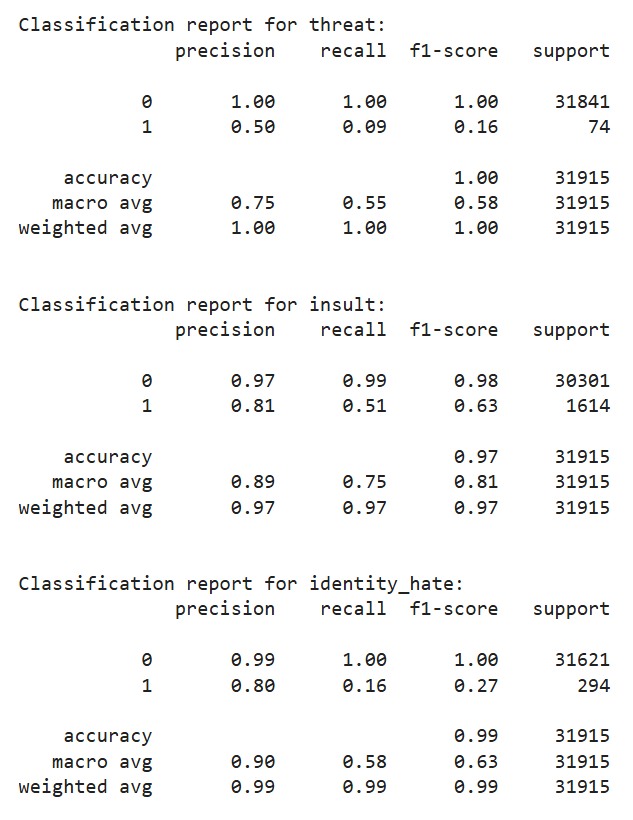
**Appendix C.5 Model Evaluation**

**Code Snippet:**

****

**Output:**

****

****

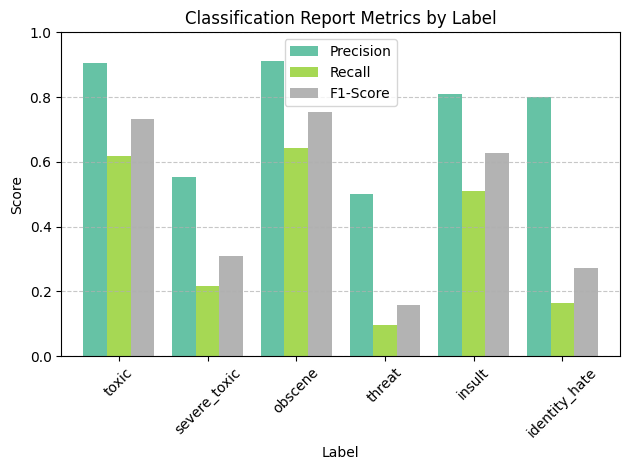
**Appendix C.6 Barchart for Model Classification**

**Code Snippet:**

****

****

**Output:**

****

# 

# References

1. Jigsaw. (2018). *Toxic Comment Classification Challenge* [Data set]. Kaggle. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
2. Rastogi, A., Satyam, D., & Devasane, M. (2025). *Toxic comment classification* (Source code). GitHub.<https://github.com/Anugrahrastogi/toxic-comment-classification>
3. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the Natural Language Toolkit*. O’Reilly Media.

**Team Contributions:**

**Submission Details:**