CS545 Machine Learning Final Project Report

Classification of Comments on Social Platforms using Machine learning algorithms

Team Members and Contributions:

In this group project, all five members made equal contributions.

Lakshmi Vyshnavi Vutukuri(PSU ID: 908811297): Worked on data preprocessing, feature engineering and data visualization

Addagalla Lakshmi Sai Prasanna(PSU ID: 915156592) and Monika Kamineni(PSU ID: 920433615): Applied Logistic regression and Multinomial Naive Bayes

Deepak Lukulapu(PSU ID: 939728351) and Bhuvaneswari Kudaravalli(PSU ID: 915053514): Applied SGD Classifier and Adaboost classifier

ABSTRACT

Toxic comment classification is a crucial task in natural language processing (NLP), aimed at identifying and categorizing toxic language within textual data across online platforms. In this study, various machine learning (ML) models are employed for toxic comment classification using a dataset consisting of comments labeled with different toxicity categories. The dataset undergoes preprocessing, including text cleaning, feature engineering, and TF-IDF vectorization, to transform the raw textual data into numerical features suitable for model training. The ML models, including logistic regression, are trained on the preprocessed data to classify comments into multiple toxicity classes. Evaluation metrics such as accuracy, confusion matrices, and area under the receiver operating characteristic curve (AUC-ROC) are employed to assess the performance of the models. Additionally, cross-validation scores are utilized to evaluate the robustness and generalization ability of the models. The results demonstrate the effectiveness of ML approaches in accurately identifying toxic comments, providing insights into mitigating online toxicity and fostering healthier online discourse.

1. Introduction

In the digital age, the internet has become a battleground for freedom of speech and censorship, with numerous instances where individuals have faced legal consequences for their online expressions. Notably, a man in Thailand received a prison sentence of 35 years for making comments deemed disrespectful to the monarchy on Facebook. Similarly, in Mississippi, USA, an educator was dismissed from her position at Batesville Intermediate Primary School following a racially charged comment on her social media page targeting individuals with darker skin tones.

The phenomenon of conversational toxicity has increasingly led individuals to refrain from engaging in open dialogues online, deterred by the fear of encountering or contributing to toxic interactions. The primary objective of this research is to explore methods for identifying toxic commentary within online platforms, aiming to discourage the dissemination of potentially harmful messages. By promoting a more respectful and thoughtful exchange of ideas, the project seeks to mitigate the adverse effects of online toxicity. This study evaluates the efficacy of machine learning techniques, specifically logistic regression and neural networks, in classifying text-based content. These methodologies have gained prominence across various fields, including economics, environmental science, and healthcare, due to their ability to analyze and interpret large datasets.

2. Business Use Case

Online discussions, especially in anonymous or semi-anonymous settings, frequently devolve into hostile exchanges, marred by abuse and harassment. This toxicity undermines the very essence of community engagement online, leading to the closure of comment sections on numerous platforms. Efforts by entities such as Alphabet's Conversation AI aim to curb these negative behaviors by developing tools designed to foster healthier online interactions. Nonetheless, initial models have shown biases, mistakenly correlating certain expressions of identity with toxicity. This project is driven by the ambition to devise a model capable of discerning toxic comments with greater accuracy, including those mentioning specific identities without resorting to prejudicial associations.

This model is intended for application across various social media platforms, including Facebook, Twitter, and Instagram, to identify comments containing toxic, severely toxic, obscene, threatening, insulting, or hate speech related to identity. The project also extends to analyzing toxicity within online gaming chat rooms, employing a dataset equipped to identify various forms of toxicity through the application of six distinct classifiers. The ultimate goal is to enable the identification and moderation of toxic comments more effectively, thereby reducing cyber abuse and fostering more inclusive online environments.

3. Data

The dataset underpinning this study was sourced from Kaggle as part of the Toxic Comment Classification Challenge. It comprises seven columns, each serving a unique purpose in the analysis of online comments. The columns include the comment's unique identifier, the text of the comment, and binary indicators for various types of toxic content such as general toxicity, severe toxicity, obscenity, threats, insults, and hate speech targeting specific identities. This structure facilitates a comprehensive examination of the nature and extent of toxic content within online discourse. You can access the dataset through the provided link on the Kaggle platform, where you'll find detailed information about its structure, contents, and accompanying metadata.

Jigsaw, Toxic Comment Classification,2018. Available:
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

4. Data Pre-processing

The preparatory steps taken to refine the dataset for analysis involved several key procedures. Initially, the dataset was scrutinized for missing values, revealing no empty cells in the dataset.

Handling Missing Values:

An analysis of missing values was conducted to assess the completeness of the dataset. By identifying missing values in both the training and test datasets, appropriate strategies could be devised for handling them, ensuring the integrity of the data used for analysis.

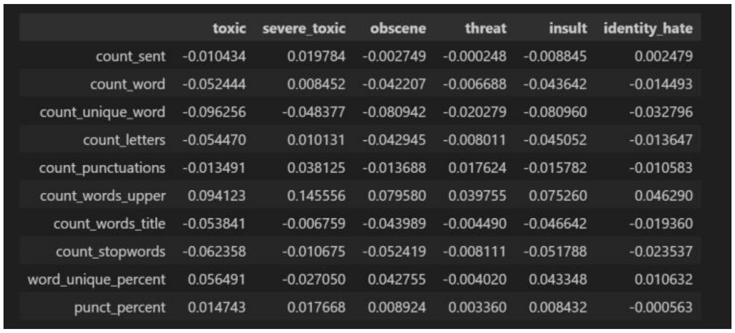
Labeling and Feature Engineering:

Comments without any associated tags were labeled as "clean" to distinguish them from comments with toxic attributes. Feature engineering techniques were employed to derive additional features from the text data, such as the count of words, sentences, punctuations, and unique words. These engineered features provided valuable insights into the linguistic characteristics of the comments.

Feature Engineering

The intricate process of feature engineering allowed us to expand our analytical framework from the initial single column of **comment_text** to include an additional 11 features, each offering unique insights into the textual data. Here's an elaboration of the newly engineered features:

- **Clean**: A boolean indicator confirming whether a comment is devoid of any of the toxic tags.
- **Count_sent**: Represents the total number of sentences within the comment, providing insights into its length and complexity.
- **Count_word**: The total word count in the comment, offering a measure of its verbosity.
- **Count_unique_word**: Counts the unique words present, indicating the diversity of vocabulary.
- **Count_letter**: The total character count, including spaces, giving a sense of the comment's length.
- **Count_punctuations**: Tallies the punctuation marks, shedding light on the writing style.
- **Count_words_upper**: Counts words in uppercase, which might indicate shouting or emphasis.
- **Count_words_title**: The number of words starting with a capital letter, potentially signaling proper nouns or beginnings of sentences.
- **Count_stopwords**: Quantifies the stop words used, helping identify common language patterns.
- **Mean_word_len**: Calculates the average word length, offering insights into the use of complex versus simple words.
- **Word_unique_percent**: The percentage of unique words out of the total, indicating originality.
- **Punct_percent**: The percentage of text that is punctuation, providing insights into grammatical structure.



	toxic	severe_toxic	obscene	threat	insult	identity_hate
count_sent	-0.010434	0.019784	-0.002749	-0.000248	-0.008845	0.002479
count_word	-0.052444	0.008452	-0.042207	-0.006688	-0.043642	-0.014493
count_unique_word	-0.096256	-0.048377	-0.080942	-0.020279	-0.080960	-0.032796
count_letters	-0.054470	0.010131	-0.042945	-0.008011	-0.045052	-0.013647
count_punctuations	-0.013491	0.038125	-0.013688	0.017624	-0.015782	-0.010583
count_words_upper	0.094123	0.145556	0.079580	0.039755	0.075260	0.046290
count_words_title	-0.053841	-0.006759	-0.043989	-0.004490	-0.046642	-0.019360
count_stopwords	-0.062358	-0.010675	-0.052419	-0.008111	-0.051788	-0.023537
word_unique_percent	0.056491	-0.027050	0.042755	-0.004020	0.043348	0.010632
punct_percent	0.014743	0.017668	0.008924	0.003360	0.008432	-0.000563

Fig: Feature Engineering

Text Preprocessing:

The raw text data underwent preprocessing steps to standardize and clean the textual content. Non-alphabetic characters were removed, text was converted to lowercase, and extraneous spaces were stripped. This preprocessing ensured consistency and improved the quality of the text data for subsequent analysis.

TF-IDF Vectorization:

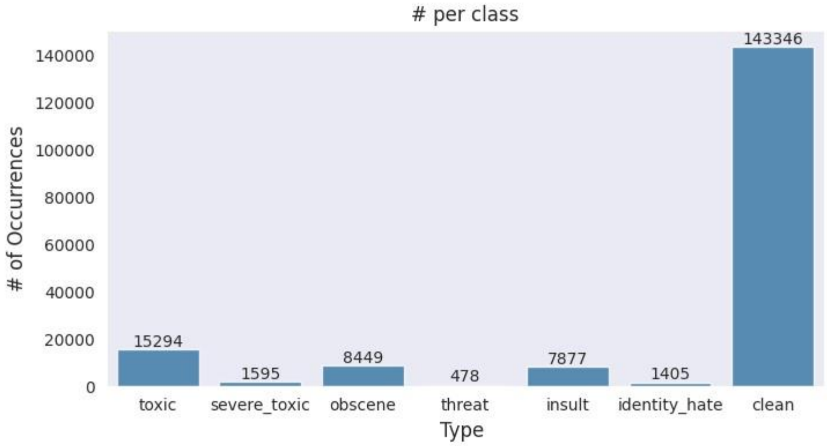
Text data was transformed into numerical feature vectors using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This process involved tokenizing the text, removing stop words, and computing TF-IDF scores for each term. TF-IDF vectorization enabled the conversion of text data into a format suitable for machine learning algorithms.

Train-Test Split:

The preprocessed data was split into training and validation sets using the train_test_split function. This division ensured that the model could be trained on a portion of the data and evaluated on an independent subset, facilitating the assessment of its performance on unseen data.

4. Data Visualization

To deepen our understanding of the dataset, we employed various data visualization techniques. These visualizations offered insightful perspectives on the distribution and characteristics of the data: Clean Feature Visualization: By introducing a new column named "clean," we were able to distinguish comments not associated with any tags. A bar graph utilizing this feature provided a clear comparison between the counts of output features and the "clean" designation, highlighting the prevalence of untagged (clean) comments.



Category Classification Visualization: Leveraging the "clean" feature, another bar graph was created to visually represent the distribution of comments across different toxicity categories such as toxic, severe_toxic, obscene, threat, insult, and identity_hate. This visualization effectively illustrated the breakdown of comments by category.

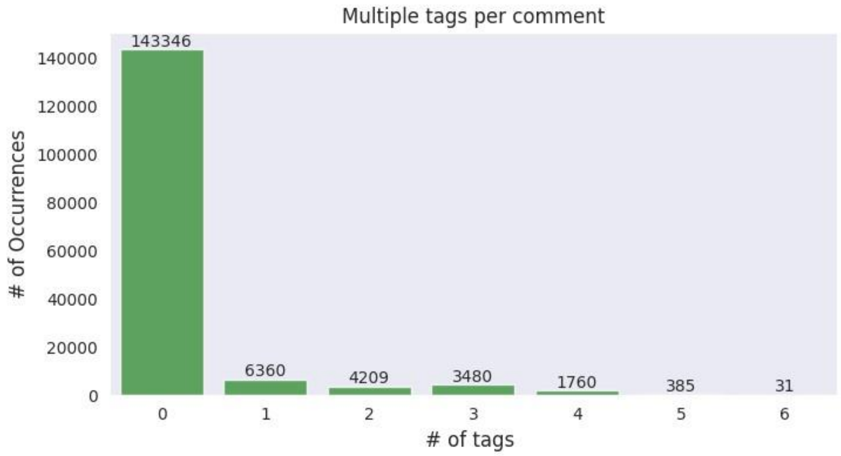


Fig: Category Classification Visualization

A correlation analysis was conducted to explore the relationship between these new features and the output variables, utilizing a heat map for visualization. This analysis revealed a lack of significant correlation between the newly added features and the output variables, leading to their exclusion from the final model. This decision underscores the nuanced challenge of identifying features that significantly impact model performance.

Output Features Correlation Heatmap: A heatmap generated from the correlation values of output features helped identify any potential relationships between them. This visualization aimed to uncover any redundancy or correlation that could inform dimensionality reduction strategies.

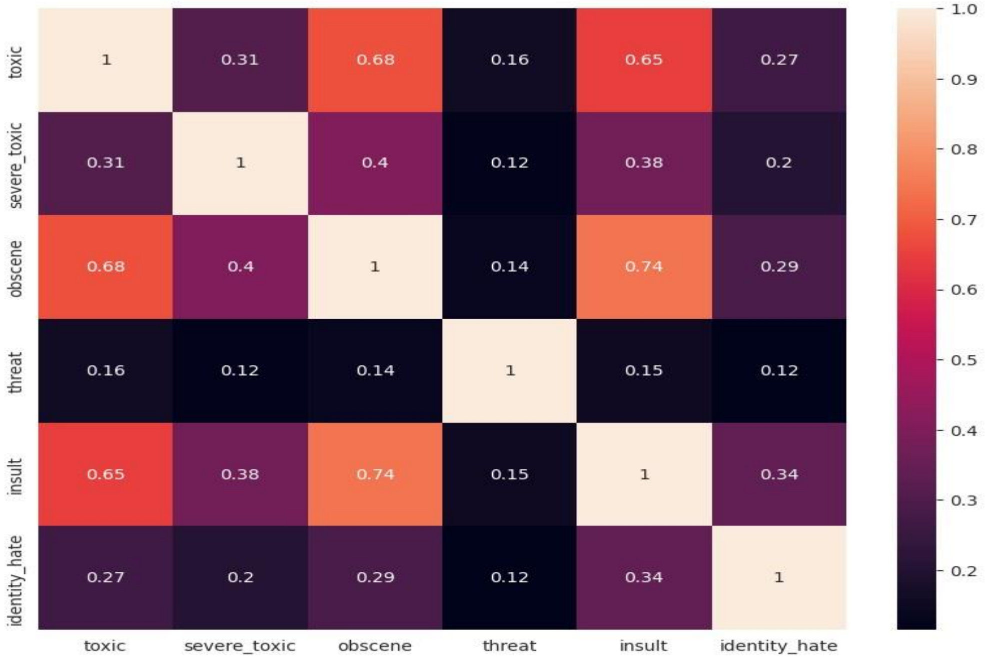
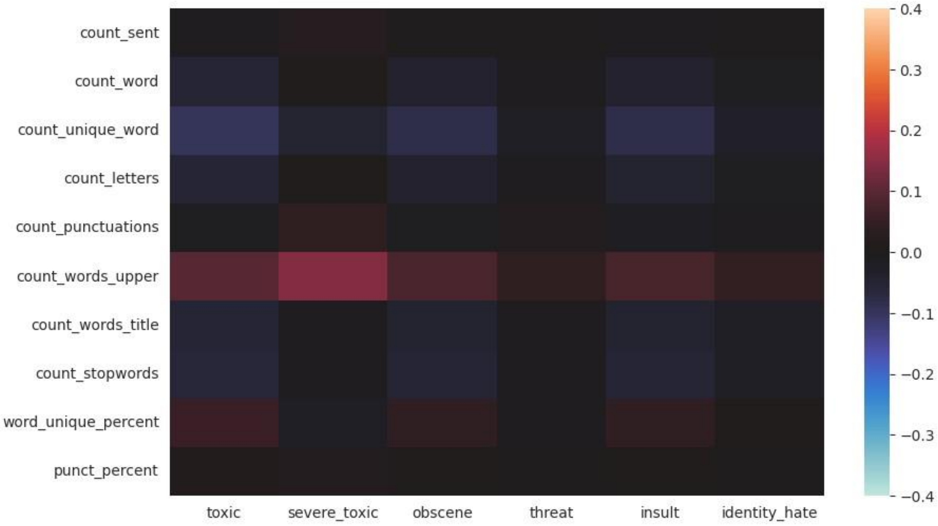


Fig: heatmap

Feature Correlation with Output Heatmap: Another heatmap was crafted to explore how the newly created features correlate with the output features. This visualization was crucial for assessing the potential of these new features to enhance the model's ability to classify text accurately.

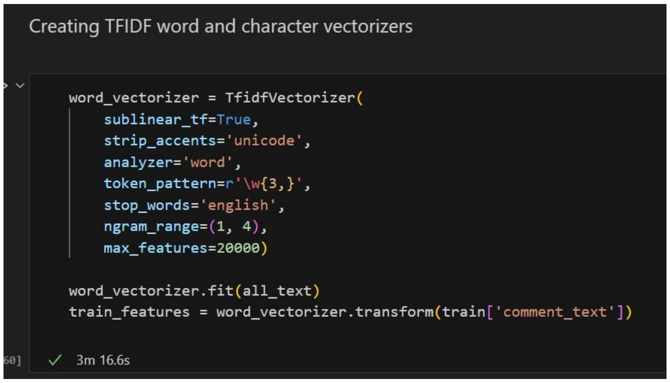


These visualizations not only facilitated a more comprehensive understanding of the dataset but also informed the modeling process by highlighting significant patterns and relationships within the data.

5. Models Evaluation

In this research, we employed a variety of datasets to train and evaluate the performance of different models, including the Logistic Regression model, Adaboost Classifier, Multinomial NB Classifier, and SGD Classifier. Unlike conventional multi-class classification, the output features in our study are distinctive, allowing for a single comment to be categorized into multiple classes. For instance, a derogatory term could simultaneously fall under the categories of toxic, obscene, and insulting. Therefore, we iterated the training process for each model multiple times, each with a different set of output features, to adapt to this complexity.

To prepare our classifiers, we utilized TF-IDF scores, derived from the TF-IDF vectorizer in the sci-kit learn library. The configuration for this process included several parameters: sublinear_tf for sublinear scaling, an analyzer to treat words as tokens, a token_pattern to identify tokens based on a specified pattern, stop_words to exclude stopwords, ngram_range to create unigrams, bigrams, and trigrams, and max_features to limit the analysis to a predefined number of top features. The dataset was partitioned using a random seed of 2018, allocating 30% for validation purposes. The outcomes of the predictions were then compiled into a data frame for further analysis, such as computing the confusion matrix and accuracy metrics.



Creating TFIDF word and character vectorizers

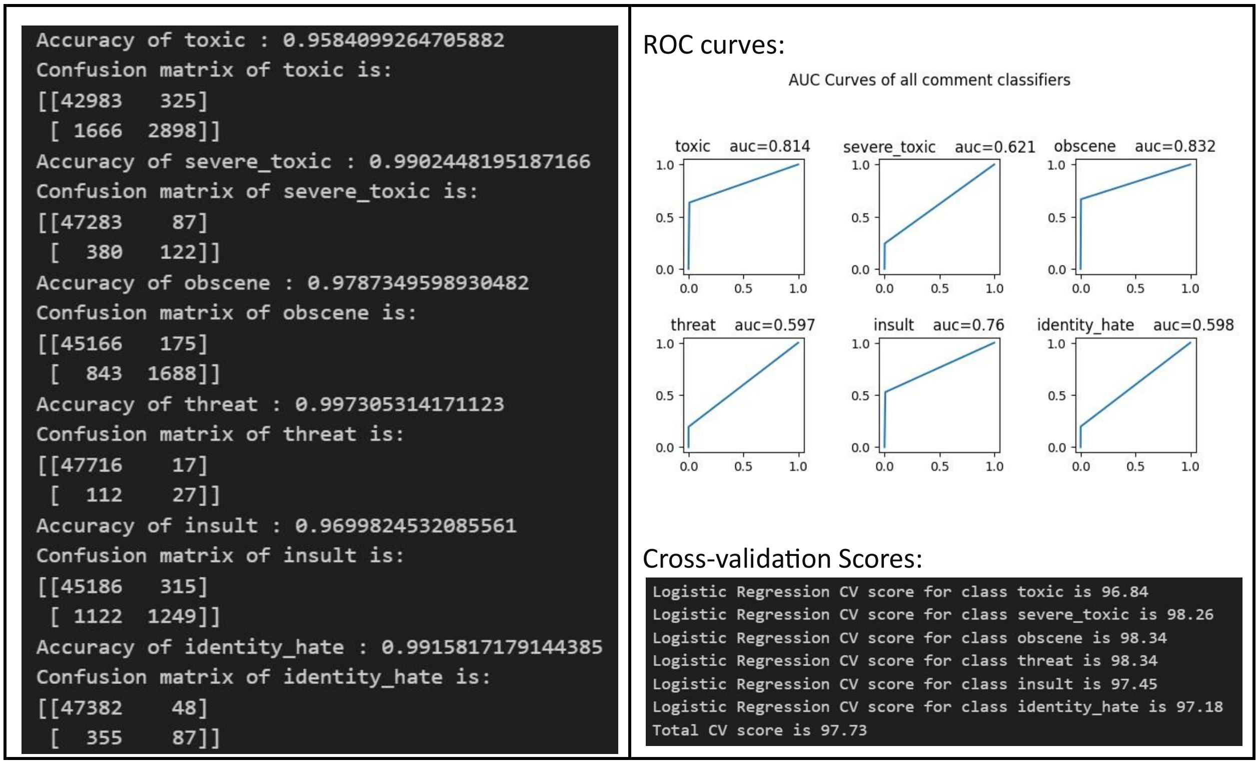
```
word_vectorizer = TfidfVectorizer(
    sublinear_tf=True,
    strip_accents='unicode',
    analyzer='word',
    token_pattern=r'\w{3,}',
    stop_words='english',
    ngram_range=(1, 4),
    max_features=20000)

word_vectorizer.fit(all_text)
train_features = word_vectorizer.transform(train['comment_text'])
```

60] ✓ 3m 16.6s

LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability of a binary outcome based on input features. In the context of toxic comment classification, logistic regression proves to be an effective tool. By leveraging the textual content of comments as input features, logistic regression models can learn to distinguish between toxic and non-toxic comments. Through the sigmoid function, logistic regression computes the probability that a given comment belongs to the toxic class, allowing for nuanced predictions. The interpretability of logistic regression coefficients further aids in understanding which words or phrases contribute most strongly to toxicity prediction. Moreover, logistic regression's simplicity and efficiency make it particularly suitable for processing large volumes of text data commonly encountered in online platforms. Overall, logistic regression serves as a powerful and interpretable technique for toxic comment classification, enabling platforms to identify and moderate harmful content effectively.

For the **Logistic Regression model**, the validation phase yielded specific accuracies and confusion matrix data to find the correct classification.



```
Accuracy of toxic : 0.9584099264705882
Confusion matrix of toxic is:
[[42983   325]
 [ 1666  2898]]
Accuracy of severe_toxic : 0.9902448195187166
Confusion matrix of severe_toxic is:
[[47283    87]
 [  380   122]]
Accuracy of obscene : 0.9787349598930482
Confusion matrix of obscene is:
[[45166   175]
 [  843  1688]]
Accuracy of threat : 0.997305314171123
Confusion matrix of threat is:
[[47716    17]
 [  112    27]]
Accuracy of insult : 0.9699824532085561
Confusion matrix of insult is:
[[45186   315]
 [ 1122  1249]]
Accuracy of identity_hate : 0.9915817179144385
Confusion matrix of identity_hate is:
[[47382    48]
 [  355    87]]
```

ROC curves:

Cross-validation Scores:

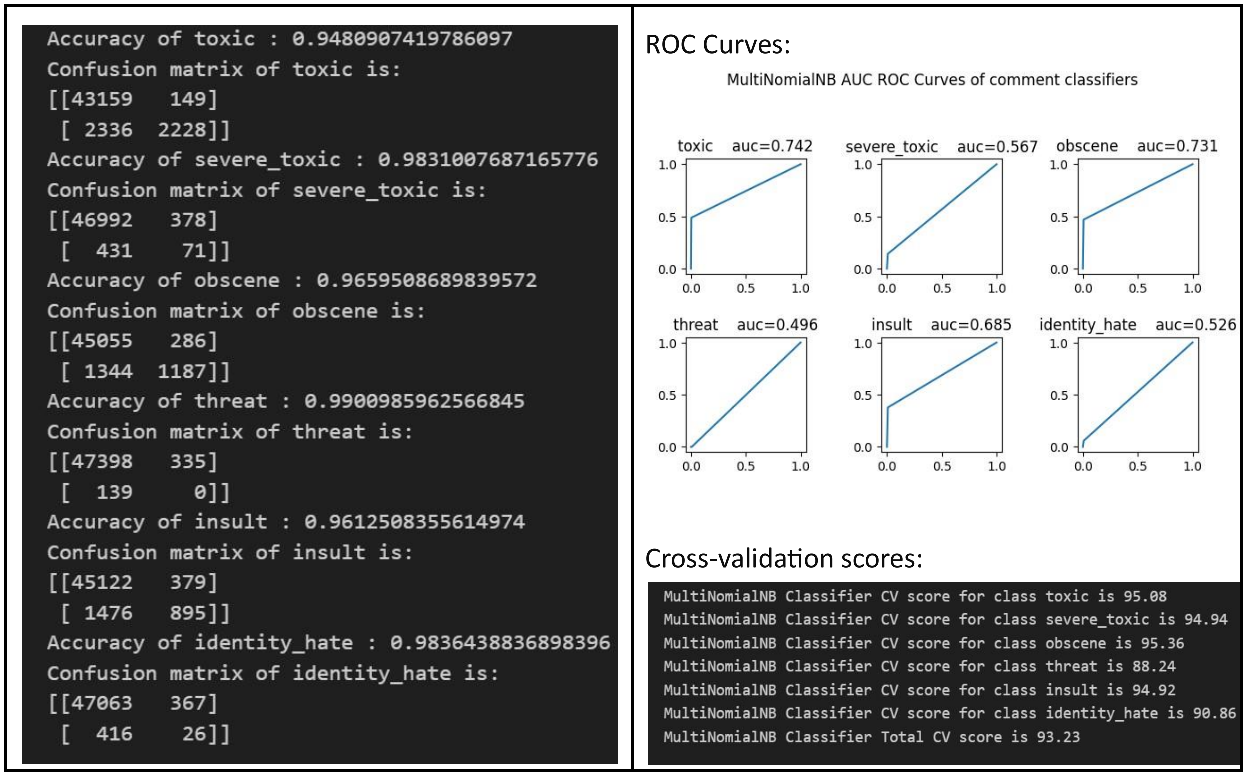
```
Logistic Regression CV score for class toxic is 96.84
Logistic Regression CV score for class severe_toxic is 98.26
Logistic Regression CV score for class obscene is 98.34
Logistic Regression CV score for class threat is 98.34
Logistic Regression CV score for class insult is 97.45
Logistic Regression CV score for class identity_hate is 97.18
Total CV score is 97.73
```

Fig: Results of Logistic Regression

MULTINOMIAL NAIVE BAYES MODEL

The Multinomial Naive Bayes (MultinomialNB) model is a probabilistic classifier commonly used in text classification tasks, including toxic comment classification. For toxic comment classification, MultinomialNB classifiers are utilized to predict the probability of comments belonging to toxic or non-toxic classes based on textual features. Unlike logistic regression, MultinomialNB assumes that features follow a multinomial distribution, making it suitable for text data represented as word frequency counts or TF-IDF features. By fitting MultinomialNB models to training data and making predictions, the model learns to distinguish between toxic and non-toxic comments. Through evaluation metrics such as accuracy and confusion matrices, the effectiveness of the MultinomialNB classifier in classifying toxic comments can be assessed. Additionally, the use of ROC curves and area under the curve (AUC) scores provides insights into the classifier's performance across different thresholds. Overall, MultinomialNB serves as a valuable tool for toxic comment classification, offering simplicity, efficiency, and effectiveness in identifying and moderating harmful content on online platforms.

The Multinomial Naive Bayes model, upon validation with the same dataset, also demonstrated specific accuracies and confusion matrix which gives us the correct classification.



```
Accuracy of toxic : 0.9480907419786097
Confusion matrix of toxic is:
[[43159   149]
 [ 2336  2228]]
Accuracy of severe_toxic : 0.9831007687165776
Confusion matrix of severe_toxic is:
[[46992   378]
 [  431    71]]
Accuracy of obscene : 0.9659508689839572
Confusion matrix of obscene is:
[[45055   286]
 [ 1344  1187]]
Accuracy of threat : 0.9900985962566845
Confusion matrix of threat is:
[[47398   335]
 [  139     0]]
Accuracy of insult : 0.9612508355614974
Confusion matrix of insult is:
[[45122   379]
 [ 1476   895]]
Accuracy of identity_hate : 0.9836438836898396
Confusion matrix of identity_hate is:
[[47063   367]
 [  416    26]]
```

ROC Curves:

Cross-validation scores:

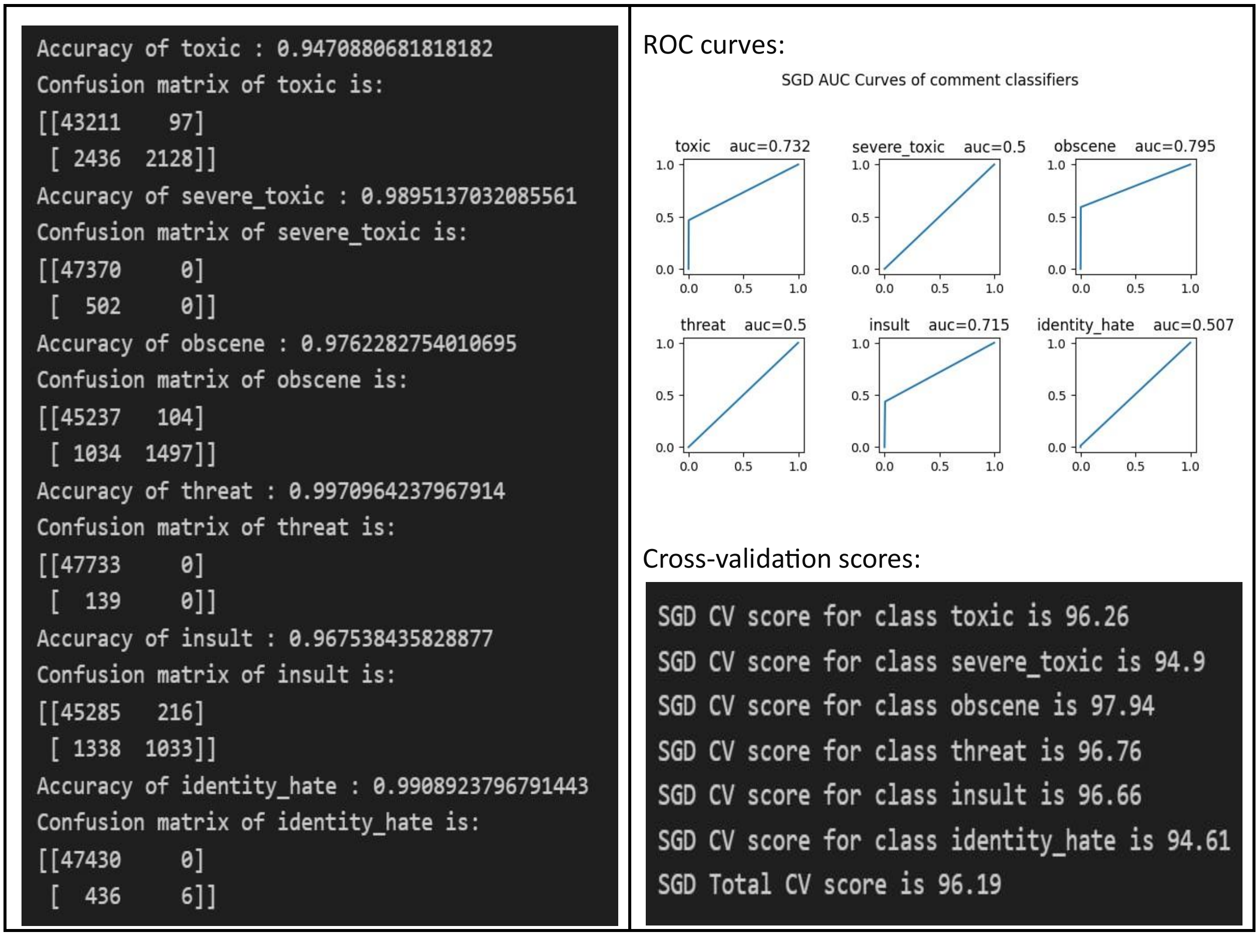
```
MultiNomialNB Classifier CV score for class toxic is 95.08
MultiNomialNB Classifier CV score for class severe_toxic is 94.94
MultiNomialNB Classifier CV score for class obscene is 95.36
MultiNomialNB Classifier CV score for class threat is 88.24
MultiNomialNB Classifier CV score for class insult is 94.92
MultiNomialNB Classifier CV score for class identity_hate is 90.86
MultiNomialNB Classifier Total CV score is 93.23
```

Fig: Results of Multinomial Naive Bayes Model

STOCHASTIC GRADIENT DESCENT MODEL

The Stochastic Gradient Descent (SGD) Classifier model, implemented through the SGD Classifier class in scikit-learn, is a versatile algorithm commonly employed in text classification tasks, including toxic comment classification. In this, SGD classifiers are utilized to predict the probability of comments belonging to toxic or non-toxic classes based on textual features. The loss parameter is set to "hinge", indicating the use of the hinge loss function for linear SVM classification. By fitting SGD Classifier models to training data and making predictions, the model learns to distinguish between toxic and non-toxic comments efficiently. Through evaluation metrics such as accuracy and confusion matrices, the effectiveness of the SGD Classifier in classifying toxic comments can be assessed. Additionally, the use of ROC curves and area under the curve (AUC) scores provides insights into the classifier's performance across different thresholds. Overall, the SGD Classifier serves as a valuable tool for toxic comment classification, offering efficiency, scalability, and effectiveness in identifying and moderating harmful content on online platforms.

When validating the **SGD Classifier**, we observed certain accuracies and confusion matrix results which identically finds the toxic comments.



```
Accuracy of toxic : 0.9470880681818182
Confusion matrix of toxic is:
[[43211    97]
 [ 2436  2128]]
Accuracy of severe_toxic : 0.9895137032085561
Confusion matrix of severe_toxic is:
[[47370     0]
 [  502     0]]
Accuracy of obscene : 0.9762282754010695
Confusion matrix of obscene is:
[[45237   104]
 [ 1034  1497]]
Accuracy of threat : 0.9970964237967914
Confusion matrix of threat is:
[[47733     0]
 [  139     0]]
Accuracy of insult : 0.967538435828877
Confusion matrix of insult is:
[[45285   216]
 [ 1338  1033]]
Accuracy of identity_hate : 0.9908923796791443
Confusion matrix of identity_hate is:
[[47430     0]
 [  436     6]]
```

ROC curves:

Cross-validation scores:

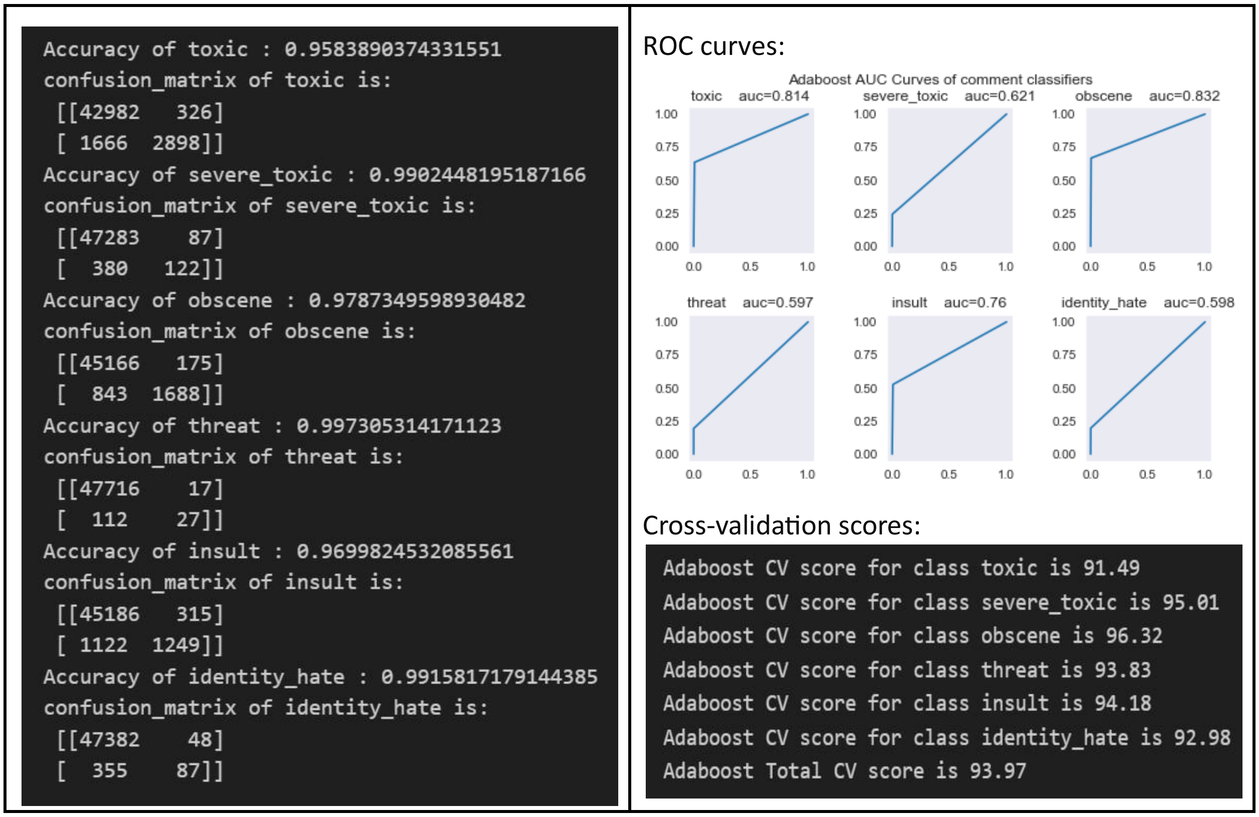
```
SGD CV score for class toxic is 96.26
SGD CV score for class severe_toxic is 94.9
SGD CV score for class obscene is 97.94
SGD CV score for class threat is 96.76
SGD CV score for class insult is 96.66
SGD CV score for class identity_hate is 94.61
SGD Total CV score is 96.19
```

Fig: Results of Stochastic Gradient Descent Model

ADABOOST CLASSIFIER MODEL

The AdaBoost Classifier model, implemented through the AdaBoost Classifier class in scikit-learn, is a powerful ensemble learning algorithm widely used in various classification tasks, including toxic comment classification. In this, AdaBoost classifiers are utilized to predict the probability of comments belonging to toxic or non-toxic classes based on textual features. With its ability to combine multiple weak classifiers to form a strong classifier, AdaBoost improves classification accuracy by iteratively focusing on instances that are hard to classify. By fitting AdaBoost Classifier models to training data and making predictions, the model effectively distinguishes between toxic and non-toxic comments. Through evaluation metrics such as accuracy and confusion matrices, the effectiveness of the AdaBoost Classifier in classifying toxic comments can be assessed. Additionally, the use of ROC curves and area under the curve (AUC) scores provides insights into the classifier's performance across different thresholds. Overall, the AdaBoost Classifier serves as a valuable tool for toxic comment classification, offering high accuracy, robustness, and the ability to handle complex classification tasks efficiently.

Lastly, the **Adaboost Classifier**, employing the identical dataset split for validation, produced certain accuracies and confusion matrix statistics.



```
Accuracy of toxic : 0.9583890374331551
confusion_matrix of toxic is:
 [[42982   326]
 [ 1666  2898]]
Accuracy of severe_toxic : 0.9902448195187166
confusion_matrix of severe_toxic is:
 [[47283    87]
 [  380   122]]
Accuracy of obscene : 0.9787349598930482
confusion_matrix of obscene is:
 [[45166   175]
 [  843  1688]]
Accuracy of threat : 0.997305314171123
confusion_matrix of threat is:
 [[47716    17]
 [  112    27]]
Accuracy of insult : 0.9699824532085561
confusion_matrix of insult is:
 [[45186   315]
 [ 1122  1249]]
Accuracy of identity_hate : 0.9915817179144385
confusion_matrix of identity_hate is:
 [[47382    48]
 [  355    87]]
```

ROC curves:

Cross-validation scores:

```
Adaboost CV score for class toxic is 91.49
Adaboost CV score for class severe_toxic is 95.01
Adaboost CV score for class obscene is 96.32
Adaboost CV score for class threat is 93.83
Adaboost CV score for class insult is 94.18
Adaboost CV score for class identity_hate is 92.98
Adaboost Total CV score is 93.97
```

Fig: Results of Adaboost classifier model

This structured approach enabled us to meticulously train and validate each model, leveraging the strengths of each classifier to address the unique challenges presented by our multi-category classification problem.

Future Work

The exploration undertaken in this project reveals promising directions for future work. Despite achieving commendable performance with Logistic Regression and AdaBoost, the journey doesn't end here. Our analysis indicated that the newly created features, while insightful, did not significantly impact the model's predictive power. This realization opens the door to further investigation into data sourcing and feature extraction techniques that could unveil more predictive elements, enhancing the model's accuracy.

A pivotal area for future exploration is the incorporation of response context into our dataset. The current limitation, marked by a high misclassification rate, could be mitigated by understanding the conversational context in which comments are made. This approach, potentially enriched by Topic Modeling, promises to refine our classification strategies by identifying trigger words and thematic elements critical for accurate toxicity detection.

Conclusion

This project embarked on a mission to navigate the complexities of classifying toxic comments within the vast expanse of online discourse. Through the application of machine learning models, we delved into the intricacies of data pre-processing, feature engineering, and model evaluation, illuminating the path to distinguish between toxic and clean comments effectively. Our journey revealed the strengths of Logistic Regression, Multinomial Naive Bayes, SGD Classifier and AdaBoost models in their ability to navigate the nuanced landscape of toxic comment classification. By transforming textual data into a format ready for machine learning analysis and rigorously testing various models, we have laid the groundwork for future advancements in this critical field. The project underscores the importance of continuous exploration and innovation in leveraging machine learning to foster a safer and more respectful online environment.

References

1. Wu, Lin and Weng, “Probability estimates for multi-class classification by pairwise coupling”, JMLR 5:975-1005, 2004.
2. Baskar, S.S, L. Arockiam and S. Charles, 2013. A systematic approach on data pre-processing in data mining. Compusoft.
3. Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In ICML (Vol. 3, pp. 616-623).
4. Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and scikit -learn. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorize r.html.
5. Mark Schmidt, Nicolas Le Roux, and Francis Bach: Minimizing Finite Sums with the Stochastic Gradient Descent.
6. Fan, Rong-En, et al., “LIBLINEAR: A library for large linear classification.”, Journal of machine learning research 9.Aug (2008): 1871-1874.
7. Y. Freund, and R. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, 1997.
8. Jigsaw, Toxic Comment Classification,2018. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.