Syracuse University

CIS 600: Applied Natural Language Processing

“Multi-Model Framework for Toxic Comment Classification”

Submitted By:

1. Darsh Shah	(SUID: 547308192)
2. Rajvi Gandhi	(SUID: 334427464)
3. Ramya Rajan	(SUID: 636547507)
4. Saurabh Borsiwala	(SUID: 357095572)
5. Sudharshan Krishnamurthy	(SUID: 925615183)

Table of Contents

1. Introduction

Within today's globalized digital environment, online platforms are essential channels for expression, cooperation, and communication. The safety, inclusivity, and decency of online communities are severely threatened by the pervasiveness of poisonous comments within the enormous sphere of digital speech. In addition to interfering with meaningful discourse, toxic comments reinforce harmful habits and spread harm. They are typified by their aggressive, provocative, or inflammatory nature.

Innovative approaches that effectively identify and reduce harmful content using Natural Language Processing (NLP) are needed to address the problem of toxic comment classification. The detection of poisonous comments in traditional ways frequently relies on lone NLP models, which may find it difficult to discern the complex linguistic clues and contextual nuances present in toxic language. Thus, complex frameworks that can thoroughly analyze and classify harmful remarks in a variety of linguistic settings and cultural quirks are desperately needed.

In order to improve the precision, resilience, and effectiveness of toxic comment detection systems, this study presents a novel Multi-Model Framework for Toxic Comment Classification. Through the integration of a wide range of NLP models, including neural network topologies, machine learning algorithms, and language heuristics, our framework aims to leverage the synergy of many modalities for maximum efficiency.

The overarching objectives of this project are twofold:

1. **Comprehensive Toxicity Evaluation:** Our approach aims to offer a comprehensive evaluation of comment toxicity by combining several NLP models, each skilled at capturing different aspects of language complexity. Our goal is to reduce false positives and false negatives by increasing the sensitivity and specificity of harmful comment classification using extensive feature extraction and ensemble learning approaches.
2. **Generalizability and Adaptability:** Our approach places a high priority on generalizability and adaptability because it acknowledges the dynamic character of online discourse and the ongoing growth of language norms. Through the utilization of transfer learning techniques and ongoing model updating processes, our goal is to develop a solution that can be tailored to changing linguistic environments and new types of toxicity in diverse digital platforms and cultural settings.

This introduction sets the stage for the report's later sections, which will examine our Multi-Model Framework for Toxic Comment Classification's theoretical foundations, experimental design, performance assessment, and practical applications. By this project, we hope to support the development of more inclusive, safe, and healthy online communities that value respectful communication among members of the community.

2. Problem Statement and Approach

a. Problem Statement

An intricate problem that has significant effects on digital discourse, community well-being, and platform integrity is the spread of harmful remarks on social media sites. Existing methods for classifying toxic comments frequently fail to correctly detect and effectively manage instances of toxicity, despite significant efforts to minimize toxic conduct. Among the main flaws in the approaches used currently are:

- **Linguistic Complexity:** Sarcasm, irony, and cultural allusions are just a few of the many language nuances that toxic comments display. These subtleties make it difficult for conventional NLP models to identify and understand contextually relevant cues.
- **Domain Adaptability:** Numerous models for detecting toxic comments are trained on particular datasets or domains, which restricts their ability to handle the variety of language patterns and cultural contexts seen in online communities and platforms.
- **Real-time Detection:** Maintaining the integrity and safety of online environments requires the prompt identification and moderation of harmful comments. But current algorithms might not be as efficient and scalable as needed for large-scale, real-time harmful comment detection.

To tackle these issues, a novel strategy that integrates the best features of several NLP models into a coherent framework that can classify harmful comments in real time and with resilience and adaptability must be developed.

b. Approach

We used a methodical, iterative procedure that includes data collecting, cleaning, preprocessing, exploratory data analysis (EDA), model development, evaluation, and result interpretation to create the Multi-Model Framework for Toxic Comment Classification. By taking these stages in order, we want to leverage Natural Language Processing (NLP) approaches to build a flexible and strong framework that can effectively identify and categorize harmful remarks in a variety of online venues. The method's phases are all painstakingly planned to guarantee the framework's efficacy, integrity, and scalability, which helps to build more secure and welcoming online communities.

1. Data Collection: As part of our methodology, we started by compiling a broad collection of comments from other websites. We gathered a vast corpus of text data covering several areas and linguistic settings from publically accessible datasets.

2. Data Cleaning and Loading: After receiving the raw data, we went through a rigorous data cleaning process to get rid of duplicate entries, non-textual components, and unnecessary metadata. After cleaning, datasets were put into the proper data structures for additional processing and examination.

3. Data Preprocessing: We used a number of text processing methods throughout the data pretreatment phase to standardize the textual data and get it ready for analysis. To put the raw text into a format appropriate for further modeling stages, this includes tokenization, lowercasing, punctuation removal, stop word removal, and lemmatization.

4. Exploratory Data Analysis (EDA): Understanding the properties and distribution of the comment data was made possible in large part via exploratory data analysis, or EDA. Key parameters such comment length, poisonous comment frequency, linguistic feature distribution, and variable relationships were investigated using statistical analysis, visualization, and qualitative investigation.

5. Feature Engineering: we conducted feature engineering to enhance the representation of text data. We employed TF-IDF features to measure word importance, integrated sentiment polarity scores to capture affective information, and combined TF-IDF with Doc2Vec vectors for semantic context. This multi-faceted approach aimed to create a comprehensive feature space, incorporating both linguistic features and deeper semantic structures for improved toxic comment classification.

6. Model Building: Our strategy was centered on creating a Multi-Model Framework for Toxic Comment Classification. In order to capture many aspects of linguistic complexity and context, we used a wide range of models such as Logistic Regression, Naive Bayes (Multinomial), Sequential NN model, Support Vector Machine

7. Evaluation: We carried out thorough evaluation processes after model training to evaluate the effectiveness of each individual model as well as the collective framework. Evaluation measures were calculated to quantify classification performance across several toxicity classes and datasets, including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). To further guarantee the models' robustness and generalizability, we used strategies like holdout validation and cross-validation.

8. Interpretation of Results: Our approach's last phase entailed analyzing the model evaluation data and deriving practical conclusions. Key language characteristics influencing classification judgments were found, misclassified cases were investigated, and model predictions were assessed. Future iterations of the Multi-Model Framework for Toxic Comment Classification will be informed by the insights gained from the interpretation of the results, which were used to optimize feature selection procedures and refine model architectures.

c. Significance

In the context of online communication, community regulation, and social welfare, toxic comment classification is extremely important. There are various benefits and concerns associated with developing a strong Multi-Model Framework for Toxic Comment Classification.

1. Enhanced Online Safety: The framework helps to create safer and more inclusive online environments by properly recognizing and filtering toxic remarks in real-time. By preventing the transmission of bad content, users can be shielded from psychological injury, harassment, and cyberbullying and can enjoy a more positive online experience.
2. Encouragement of Positive Discourse: Negative remarks obstruct positive discourse and prevent the sharing of ideas in online groups. The framework fosters a healthier discourse by assisting in the removal or moderation of harmful content. This allows users to express a variety of viewpoints and participate in important conversations without fear of hatred or intimidation.
3. Efficiency of Community Moderation: Large amounts of user-generated content are difficult for online platforms to filter successfully. By automatically identifying and categorizing harmful remarks, the Multi-Model Framework optimizes the moderation process, relieving the workload of human moderators and facilitating more effective content moderation on a large scale.
4. Protection Against Online Abuse: Hate speech, discrimination, and online abuse can all be sustained by toxic comments that single out people for being that person's color, gender, sexual orientation, religion, or other characteristics. The framework fights online abuse and upholds the values of equality, respect, and dignity for all users by quickly identifying and resolving toxic situations.
5. Gained insights on user behavior, community dynamics, and developing patterns in online discourse are obtained through the analysis of hazardous comment data and classification results. These insights can be used to improve the platform. In order to prevent toxicity and improve user experience, platforms can use these findings to guide policy creation, intervention tactics, and platform design.
6. Contribution to Research and Development: The creation of a Multi-Model Framework for Toxic Comment Classification advances machine learning and natural language processing (NLP) approaches. This research broadens the toolkit available to tackle intricate language problems in digital communication by investigating innovative methods for toxicity detection and classification.

In conclusion, the Multi-Model Framework is important for more than just platform moderation; it has wider implications for community well-being, technical innovation, online safety, and societal advancement. This framework is a proactive step toward fostering healthier and more polite online interactions by utilizing the power of natural language processing and machine intelligence.

3. Data

a. Data and Data Extraction

- Searching Kaggle for datasets related to the identification of harmful remarks and toxic comments including cyberbullying was the main method used to get data for this project. The main resource was the over 400,000 posts, tweets, and comments from various social media networks that made up the Kaggle datasets. The offensiveness—or lack thereof—of these datasets was noted in the annotations. Labels 0 and 1, which indicated whether a comment was considered toxic or not, were added to the resulting dataset.
- Preprocessing and data refinement were considered necessary because the original datasets were too big and showed inherent class disparities. In addition to presenting computing difficulties, the magnitude of the dataset prompted questions about the effectiveness of training models on unbalanced data, which could result in skewed or biased results.
- Furthermore, measures were implemented to guarantee representativeness and diversity throughout the data collection stage. User-generated content spanning a wide range of social media sites was utilized to source tweets and comments. The goal of this method was to convey the subtleties and complexity present in online debates.
- The dataset that was gathered was intended to cover a range of toxic comments, offensive language, and incidents of cyberbullying that are common on social media sites. Furthermore, the data's labels or annotations functioned as the foundation for supervised learning algorithms, which made it possible to create models that could discriminate between offensive and non-offensive information.
- A painstaking process of collecting, selecting, and honing datasets was carried out in order to provide a corpus appropriate for training, validating, and testing toxic comment classification models. The next phases of the project's workflow, such as data preparation, feature engineering, and model building, are built around this improved dataset.

b. Data Preprocessing

- Data preprocessing stands as a pivotal phase in natural language processing (NLP) and text analysis. Raw text data necessitates cleaning and transformation to ready it for subsequent analysis. This study outlines the procedures employed to clean and tokenize text data, thereby enhancing its quality and rendering it primed for further analytical procedures.

Data Cleaning

1. Convert to lowercase

All characters were changed to lowercase to start the text normalization process. By doing this, case differences that can complicate analysis are removed and uniformity in text data is guaranteed.

2. Remove URLs

In order to keep the text's content front and center, all hyperlinks (URLs) were eliminated. The accuracy of the analysis that follows is improved by this removal of superfluous material.

3. Remove user mentions

It was forbidden to mention other users in the post, even by handles or other references. This is an essential step in preventing user-specific references and keeping the focus on the text itself.

4. Remove special characters, numbers, and punctuation

Terminology, non-alphabetic letters, and digits were eliminated to improve text cleanliness. At this stage, only significant words and phrases are kept for further examination.

5. Replace multiple white spaces with a single space

One space had to be inserted between consecutive white spaces in order to provide uniform spacing. This keeps the content readable and makes the tokenization process easier later on.

Tokenizing

1. Removal of stop words

Stop words were common phrases that were eliminated to improve the quality of the analysis. Readers are able to concentrate more on the text's primary concepts as a result of the words that barely add to its meaning being removed.

2. Stemming

Using stemming, words were reduced to their most basic form. Through the reduction of word form variations, this technique guarantees that words that are similar are treated as identical for more efficient analysis.

3. Lemmatization

Condensing words into their dictionary or basic form was done by lemmatization. In doing so, you can improve text normalization and make analysis more reliable by guaranteeing that different grammatical variations of a term are consistently represented.

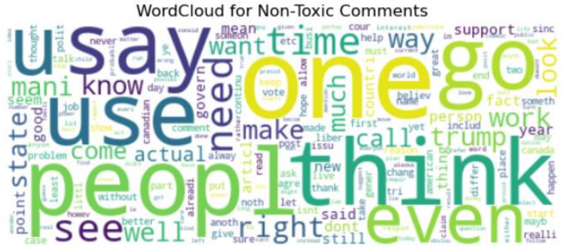
Two essential preparation stages for text analysis are data cleansing and tokenization. The text data is prepared for additional investigation and analysis by employing the methods outlined to clean and standardize it. These stages provide advantages for later natural language processing tasks, such as increased precision and productivity.

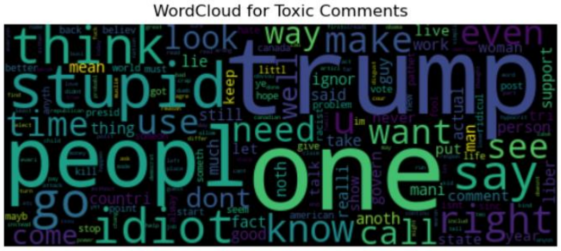
4. Exploratory Data Analysis

Exploratory Data Analysis (EDA) stands as a pivotal initial phase for comprehending the essence of data pertinent to toxic comments identification. EDA furnished us with insights into data distribution, patterns, and subtleties, thereby facilitating a more insightful approach to model construction. Our preliminary analyses encompassed scrutinizing the prevalence of toxic comments tags, investigating comment length distributions, and pinpointing potential class imbalances. Utilizing visualization methods like word clouds and frequency plots offered glimpses into the prevalent words and phrases linked with detrimental content. Furthermore, sentiment analysis unveiled the emotional undertones of the comments. EDA not only guided subsequent feature engineering choices but also illuminated potential hurdles, such as ambiguous cases, that models needed to overcome. This exhaustive exploration laid the groundwork for a more nuanced comprehension of the dataset, steering the formulation of strategies for proficient hate speech and toxic comments detection.

a. Word Cloud for Non-Toxic and Toxic Comments

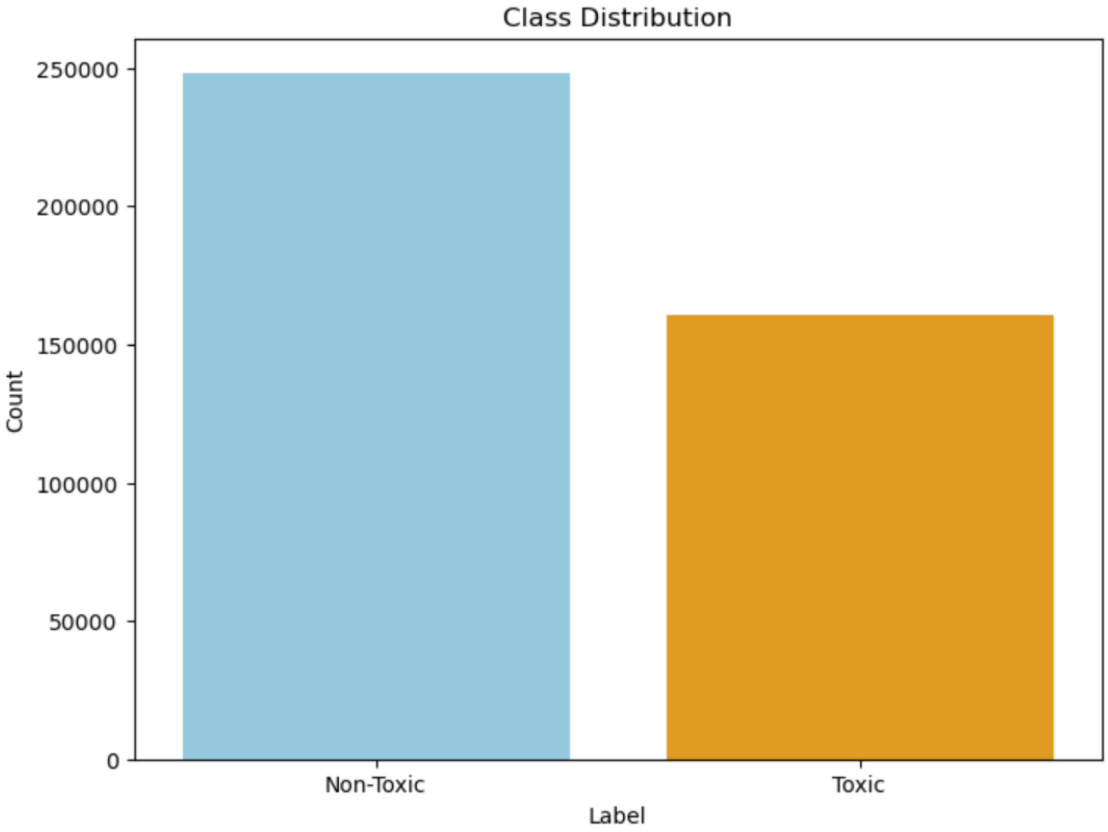
Word clouds were created to graphically represent the most frequently recurring terms in both toxic and non-toxic remarks during the exploratory data analysis (EDA) stage. A wide range of neutral and positive phrases were abundant in the word cloud that represented non-toxic comments, indicating a harmless content spectrum. The toxic comment word cloud, on the other hand, highlighted the frequency of disparaging and discriminatory words and provided a visual representation of the distinctive linguistic features linked to harmful expressions in the dataset.


WordCloud for Non-Toxic Comments


WordCloud for Toxic Comments

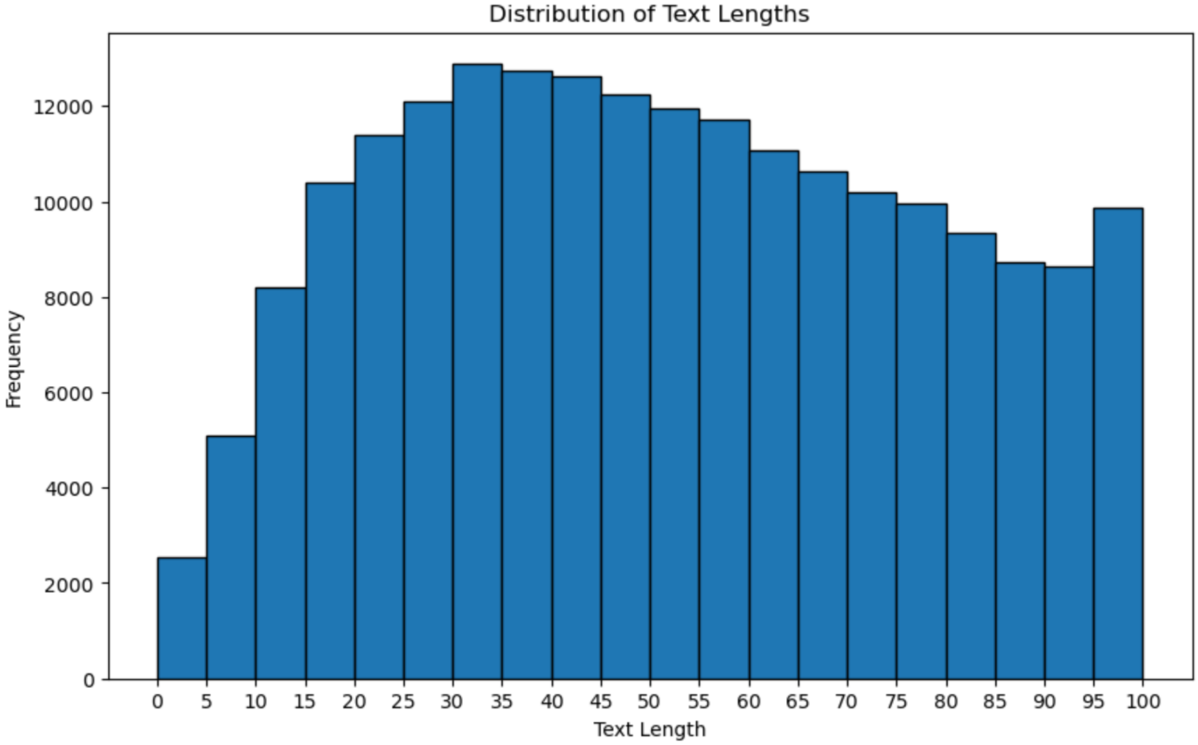
b. Class Distribution for Non-Toxic and Toxic Comments

During exploratory data analysis (EDA), we examined the distribution of classes between toxic and non-toxic remarks and discovered a considerable majority of non-toxic cases over hazardous ones. This mismatch emphasizes the difficulties of discovering harmful comments in the dataset, which are quite infrequent. Recognizing this distribution is critical for developing models that can effectively address the inherent class imbalance, ensuring robust performance in accurately identifying toxic comments while handling the larger volume of non-toxic comments.



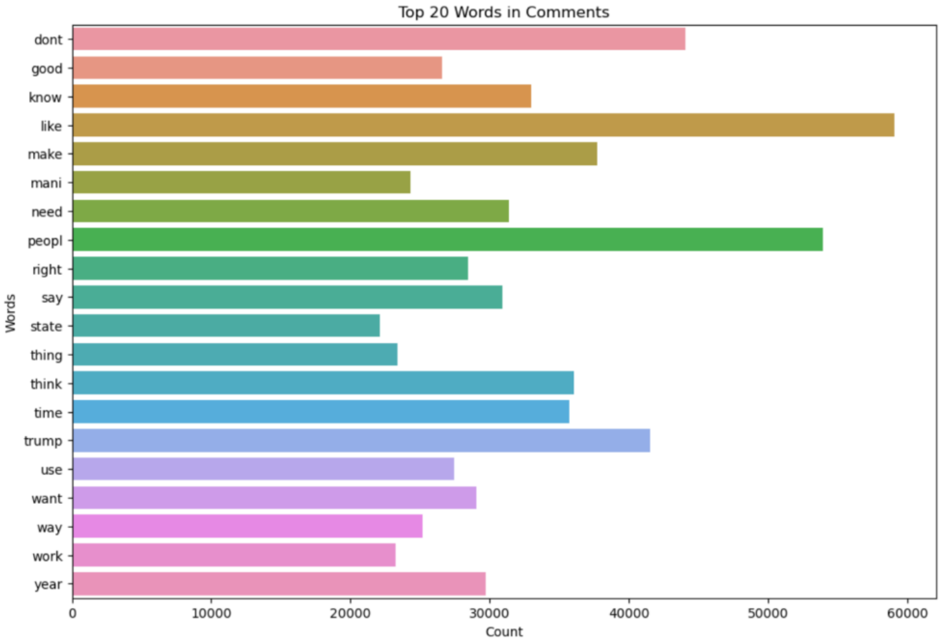
c. Distribution of Text Lengths

In the exploratory data analysis, we looked at the distribution of text lengths in the comments and discovered an average length of 25 to 45 characters. This insight gives a quantitative understanding of the dataset's comment lengths, allowing for the selection of appropriate text processing techniques and ensuring the model's adaptability to varied comment lengths. Analyzing this distribution is critical for increasing the model's sensitivity to both short and long expressions while keeping efficacy in toxic comments identification.

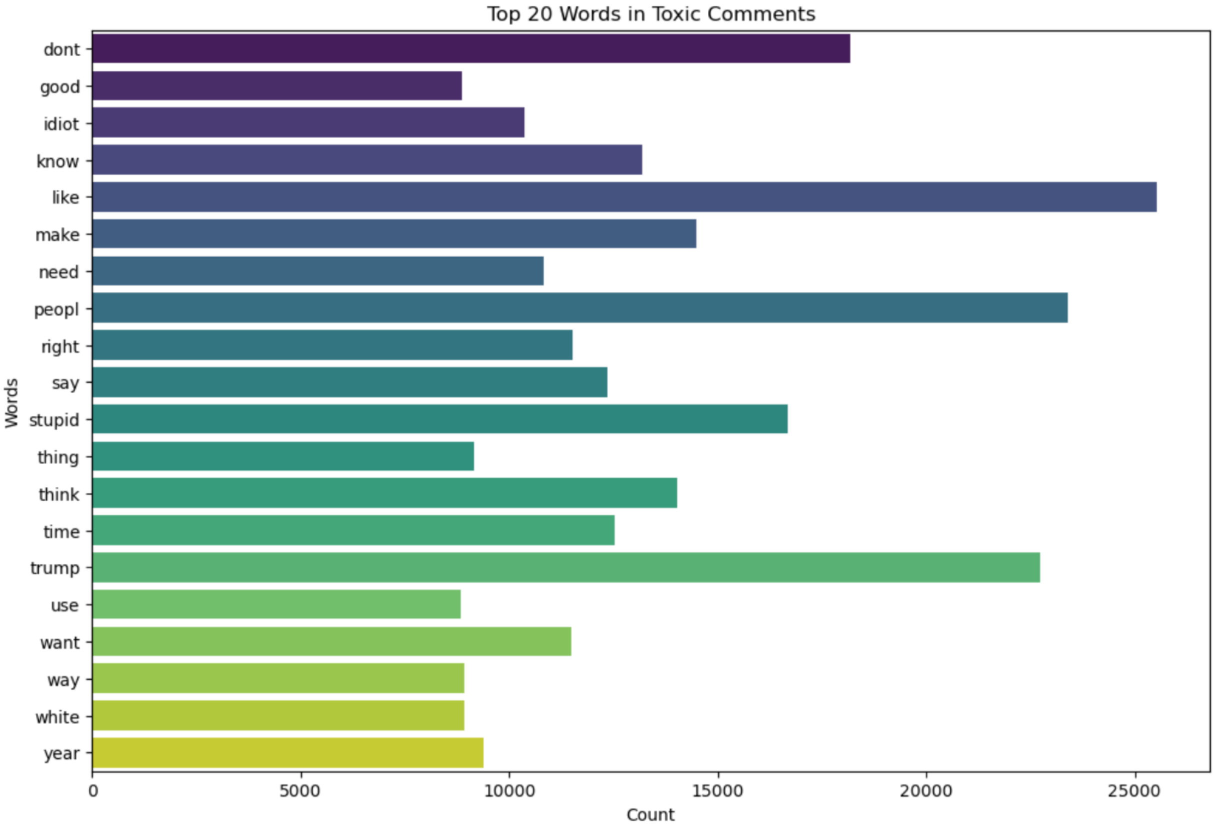


d. Horizontal Bar Graphs for Top 20 Words

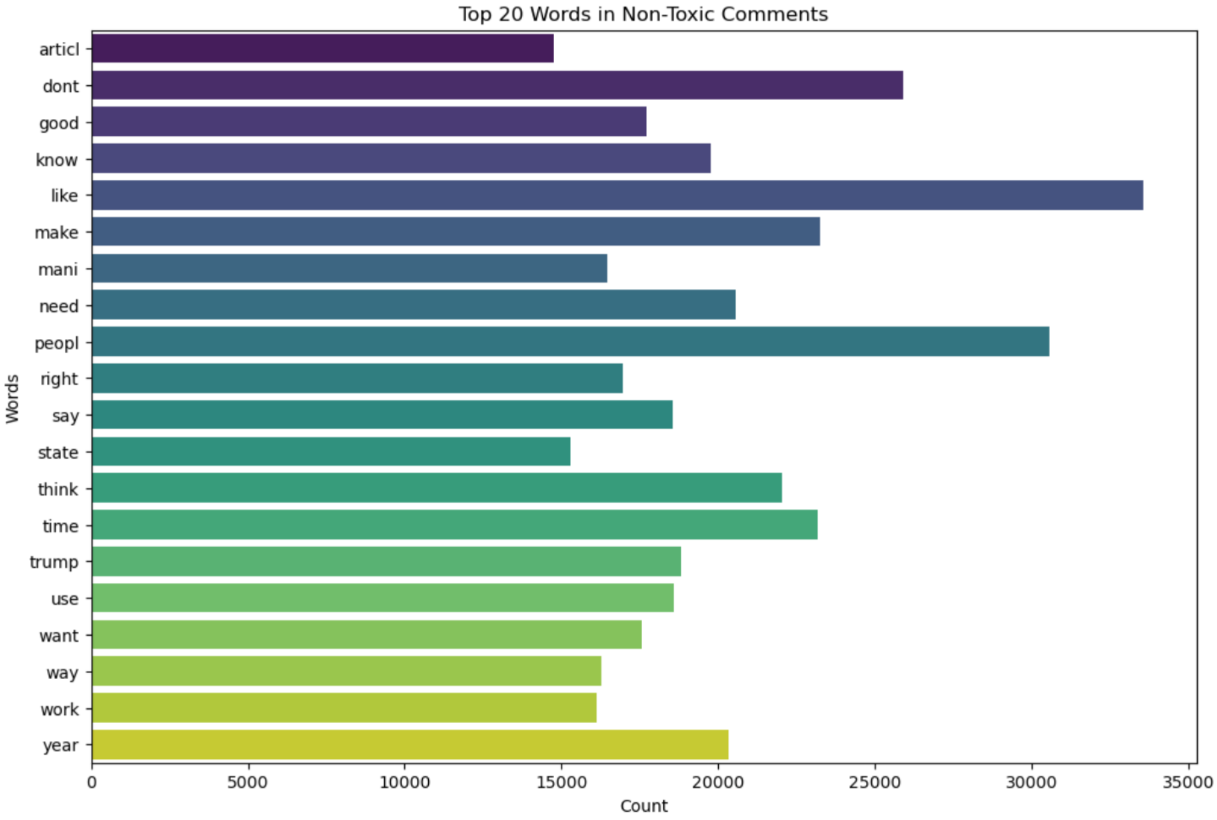
In our exploratory data analysis, we used horizontal bar graphs to graphically emphasize the top 20 most often used words in tweets and comments. These graphs provide an understandable picture of the language landscape, highlighting the most common terms in the data. This research helps to discover essential themes and language patterns, which guide further feature engineering selections for toxic comments and cyberbullying detection models.



In our exploratory data analysis, we employed horizontal bar graphs to illustrate the top 20 most frequent words encountered in toxic comments. These graphical representations offer a focused view of the particular vocabulary linked with toxic expressions, shedding light on the prevalent phrases and linguistic structures within this subset. Examining these top terms aids in comprehending the unique linguistic characteristics of toxic comments, thereby assisting in feature selection and model development for effective detection.

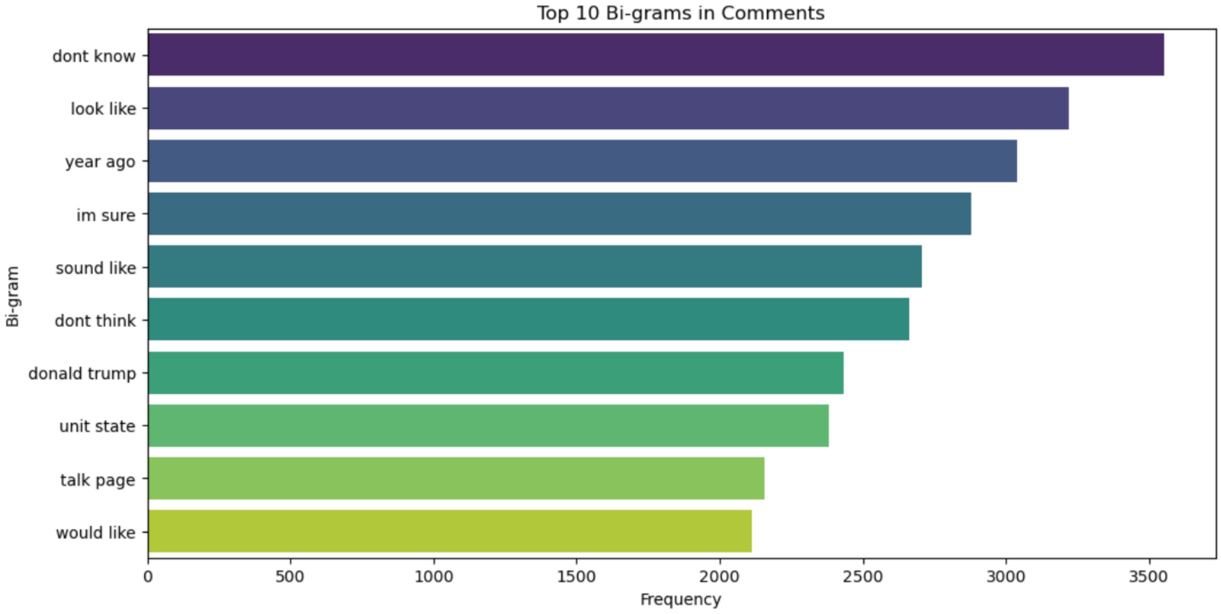


In our exploratory data analysis, we employed horizontal bar graphs to display the top 20 most common words in non-toxic comments. These visualizations offer a thorough depiction of the prevailing terms in harmless content, granting insights into the positive and neutral language frequently observed in non-toxic expressions. Grasping the distinctive vocabulary present in non-toxic instances is essential for developing models capable of accurately distinguishing between harmful and non-harmful content, thereby enhancing the effectiveness of hate speech and toxic comment detection.

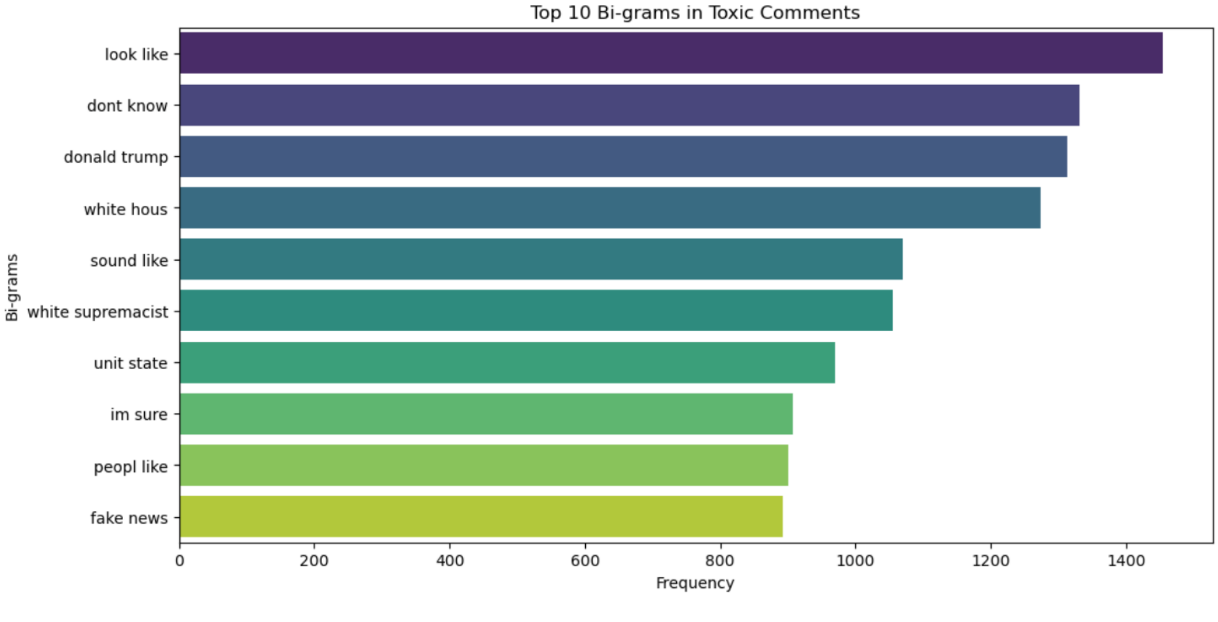


e. Horizontal Bar Graphs for Top 10 Bi-grams

In our exploratory data analysis, we utilized horizontal bar graphs to showcase the top 10 most common bi-grams in all comments. These visualizations provide a nuanced insight into paired word combinations, unveiling prevalent linguistic patterns within the dataset. The analysis of bi-grams plays a crucial role in capturing contextual information, thereby improving the model's capability to detect subtle nuances in language, leading to more effective identification of toxic comments.

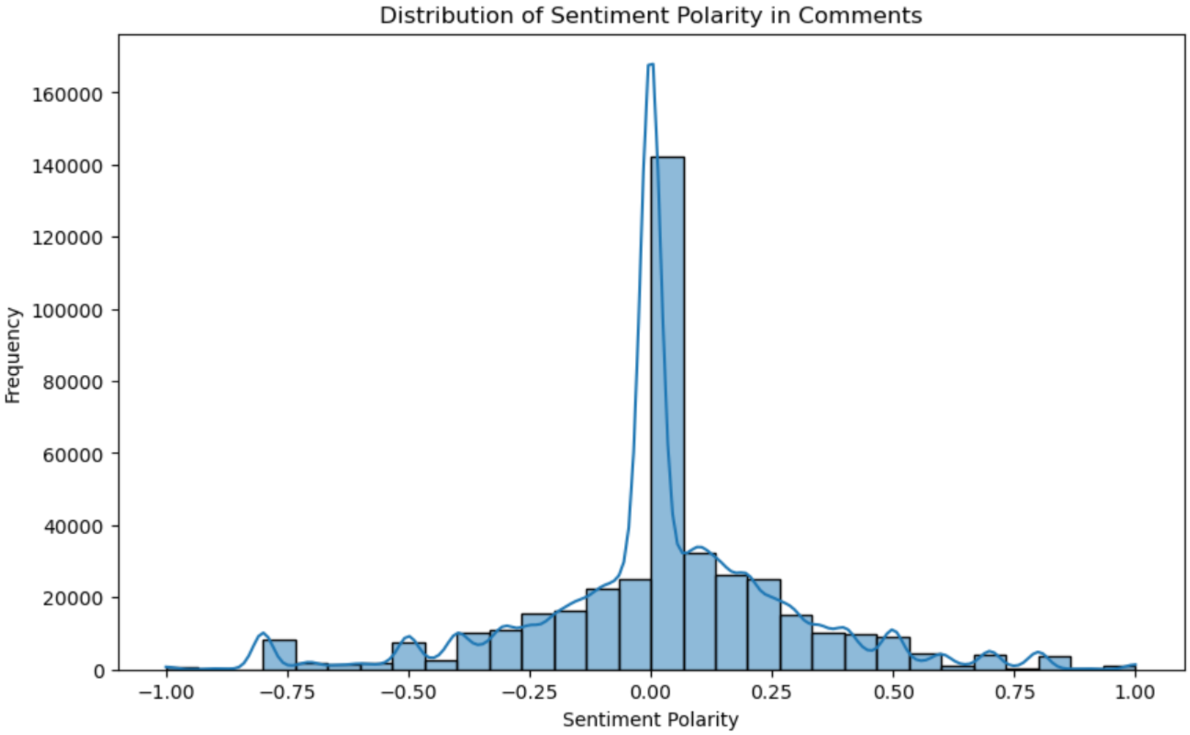


During our exploratory data analysis, we utilized horizontal bar graphs to highlight the top 10 most common bi-grams found in toxic comments. These visualizations reveal pairs of words that frequently appear together in harmful expressions, offering deeper insights into the contextual nuances of toxic comments. Examining these bi-grams enriches our comprehension of the specific language patterns linked with harmful content, facilitating the development of more nuanced and context-aware models for detecting toxic comments.



f. Sentiment Polarity

Sentiment polarity analysis involves determining the emotional tone expressed in text and assigning a number score to represent the sentiment. Using techniques such as TextBlob, sentiment polarity scores range from negative to positive, showing the intensity of negativity or positivity in the language employed. This analysis is critical for understanding the emotional context of comments, as it provides vital elements for toxic comments and cyberbullying detection models to evaluate the sentiment conveyed in social media content. The results show that many terms in the dataset were neutral, and the proportion of positive and negative attitudes was almost equal.



5. Multi-Model Approaches

a. Naive Bayes:

Naive Bayes is a probabilistic classifier based on Bayes' theorem, with the "naive" assumption of independence between features. Despite its simplicity, Naive Bayes can be effective for text classification tasks like toxic comment classification. It's computationally efficient and often serves as a baseline model for comparison with more complex algorithms.

In the context of toxic comment classification, Naive Bayes works by calculating the probability that a comment belongs to a particular toxicity class based on the presence of specific words or features in the comment. It's particularly useful when dealing with large datasets and can handle a high number of features efficiently.

b. Logistic Regression:

Logistic Regression is another popular algorithm for binary classification tasks, where the goal is to predict the probability that a given input belongs to a particular class. Despite its name, it's a linear model for classification rather than regression.

Implementation details:

Used default sigmoid logistic function.

Split the dataset into training and testing sets (80% training, 20% testing).

Scaled the TF-IDF features using StandardScaler.

Trained the model for 1000 iterations on the scaled features.

Made predictions on the test set using the trained model (logistic_model.predict).

Leveraged its simplicity and interpretability for understanding feature importance and model decisions.

c. Support Vector Machines (SVM):

SVM is a powerful algorithm for classification tasks, especially in high-dimensional spaces. It works by finding the hyperplane that best separates the different classes in the feature space.

In the context of toxic comment classification, SVM can be used to find the hyperplane that best separates toxic comments from non-toxic ones based on the features extracted from the text. SVMs are effective for tasks with a clear margin of separation between classes and can handle non-linear relationships through the use of kernel functions.

d. Sequential Neural Network (NN) Model:

Sequential NN models, such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs), are deep learning models that can capture complex relationships in sequential data like text. They are particularly effective for tasks like toxic comment classification, where the order of words and context are important.

Implementation Details:

Utilized dense layers, starting with 256 neurons and ReLU activation.

Employed a Dropout layer to prevent overfitting.

Followed by a dense layer with 128 neurons and ReLU activation.

Used another Dropout layer.

Concluded with a dense layer with 1 neuron and sigmoid activation for binary classification.

Compiled using the Adam optimizer with a learning rate of 0.001 for efficient optimization.

These models, when used in combination as part of a multi-model framework, can complement each other's strengths and improve the overall accuracy and efficiency of toxic comment classification.

6. Feature Engineering

- Feature engineering is an important step in developing models used for toxic comment classification. This stage improves both the interpretability and performance of the model. In our study, we used a range of feature engineering strategies to capture distinct characteristics of toxicity-related online comments.

  In our analysis, we had implemented the following features in our model:

  1. TF-IDF Vectorization:
     - → TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was used to determine the importance of terms/words within each comment.
     - → This technique allows for the modeling of comments as vectors, emphasizing words that are both frequent in a comment and unusual across the entire dataset.
  2. Sentiment Analysis using TextBlob:
     - → Sentiment polarity scores were extracted from the comments using TextBlob's sentiment analysis functionality.
     - → By using this strategy, we were able to include the sentimental tone of the comments in our feature set, which gave us important insight into the general sentiment that was voiced.
  3. Doc2Vec for Comment Embedding:
     - → Doc2Vec, a technique for embedding documents into vectors, was used to extract the semantic meaning from the comments.
     - → This allowed for a more nuanced interpretation of the material by displaying comments in a continuous vector space.

Integration of Features:

- The TF-IDF features, sentiment scores, and Doc2Vec vectors were all combined to form a comprehensive and consistent feature set.
- Integrating TF-IDF features, sentiment scores, and Doc2Vec vectors into a cohesive feature set is a clever technique for increasing the model's information richness. Combining these two types of characteristics allows the model to use both the lexical and semantic elements of the comments, resulting in a more sophisticated comprehension of the content.
- This holistic approach is intended to improve the models' discriminatory power, making it easier to distinguish between toxic and non-toxic comments.
- The two-step integration procedure, which first combines TF-IDF and sentiment polarity characteristics and then incorporates Doc2Vec vectors, enables an incremental analysis of feature interactions. This iterative approach allows for an evaluation of how each new collection of features contributes to the model's overall performance.

TF-IDF and Sentiment Polarity Combination:

- The initial phase is to combine TF-IDF features, which measure the value of words inside comments, with sentiment polarity scores, which indicate the emotional tone of the text.
- This combination seeks to give a balance of content-specific information and overall mood indicated in the comments.

Incorporating Doc2Vec Vectors:

- Building on the TF-IDF and sentiment features, Doc2Vec vectors add to the feature set by embedding comments in a continuous vector space that captures semantic links between words and phrases.
- This phase improves the model's ability to recognize subtle differences in meaning and context, resulting in better discrimination between toxic and non-toxic comments.
- The study addresses the potential problems of combining different types of information by employing a tiered approach to feature integration. It also detects any interactions or redundancy between characteristics that could affect model performance.
- Each step of model training provides insight into how feature combinations affect the accuracy and effectiveness of toxic comment classification. The findings show that adding all three qualities yields no significant gain, motivating more research into the dynamics of feature interactions and their implications for model performance.
- The training of models at each stage offers information on the effect of feature combinations on the accuracy and effectiveness of toxic comment classification. The absence of significant improvement when incorporating all three characteristics, as mentioned in the findings, calls for more investigation into the dynamics of feature interactions and their consequences for model performance.

Performance Evaluation:

- Despite the comprehensive feature set, combining two features or incorporating all features did not yield a significant difference in model performance.
- Logistic Regression, SVM, and Sequential Neural Network demonstrated consistent performance across various feature combinations, suggesting robustness in handling diverse types of features.
- Naive Bayes exhibited comparable performance for TF-IDF and the combination of TF-IDF with sentiment polarity scores, scoring the highest accuracy of all. However, the inclusion of all features led to a significant drop in accuracy to 61%, indicating a potential challenge in handling the increased feature dimensionality. It could also be due to the fact that MultinomialNB model could not handle negative values hence MinMaxScalar() was used to scale the values of sentiment polarity and doc2vec vector.

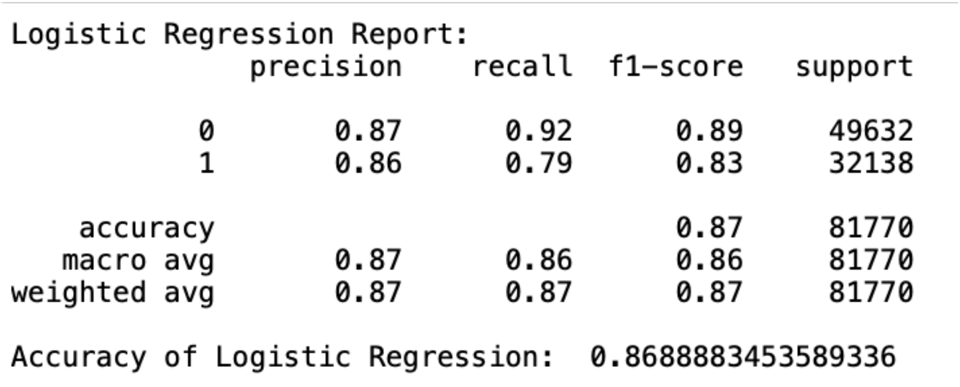
Implications and Recommendations:

- The lack of considerable performance increase with feature combinations emphasizes the need for additional research into the nature of features and their interactions.
- The disparity in Naive Bayes performance with all features implies that the algorithm may have difficulties in handling complicated feature connections.
- Future research may look into other feature engineering methodologies, model topologies, or hyperparameter tuning to achieve even greater performance gains.
- In conclusion, combining TF-IDF features, sentiment scores, and Doc2Vec vectors into a cohesive feature set demonstrates a methodical approach to capturing both lexical and semantic information. This approach, while subtle and extensive, emphasizes the ongoing problem of optimizing feature combinations for harmful comment classification and the necessity for additional research into more sophisticated modeling techniques.

7. Evaluation & Results

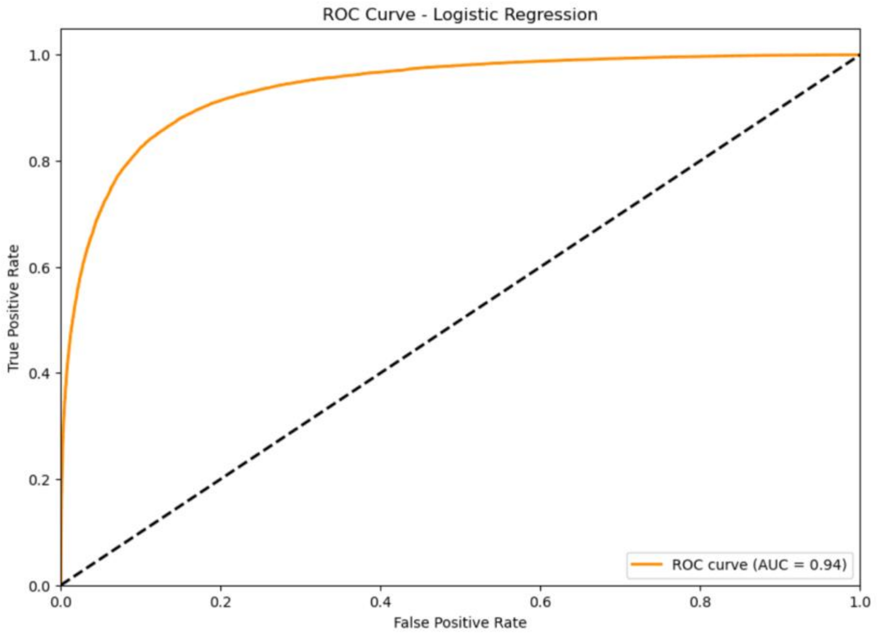
a. Logistic Regression Model Results

- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the logistic regression model is run on our preprocessed dataset.

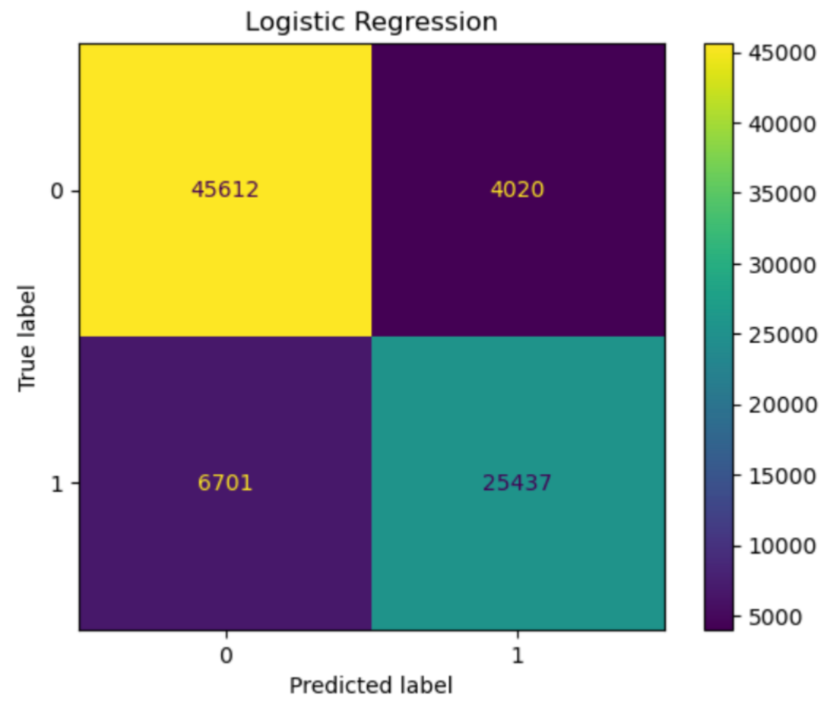


```
Logistic Regression Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.89     49632
           1       0.86      0.79      0.83     32138

    accuracy                           0.87     81770
   macro avg       0.87      0.86      0.86     81770
weighted avg       0.87      0.87      0.87     81770

Accuracy of Logistic Regression:  0.8688883453589336
```

- The following ROC (Receiver Operator Characteristics) curve shows False Positive rate vs the True Positive rate on the output of the Logistic Regression Model.

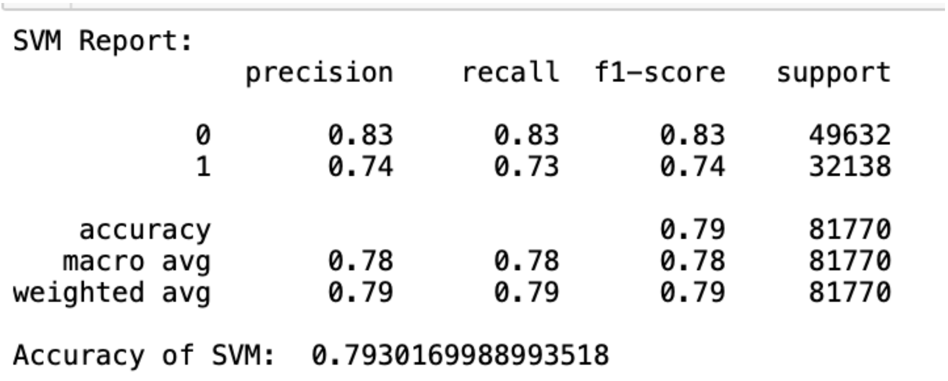


- The following Confusion Matrix shows Values between True Labels and Predicted Labels on the output of the Logistic Regression Model.



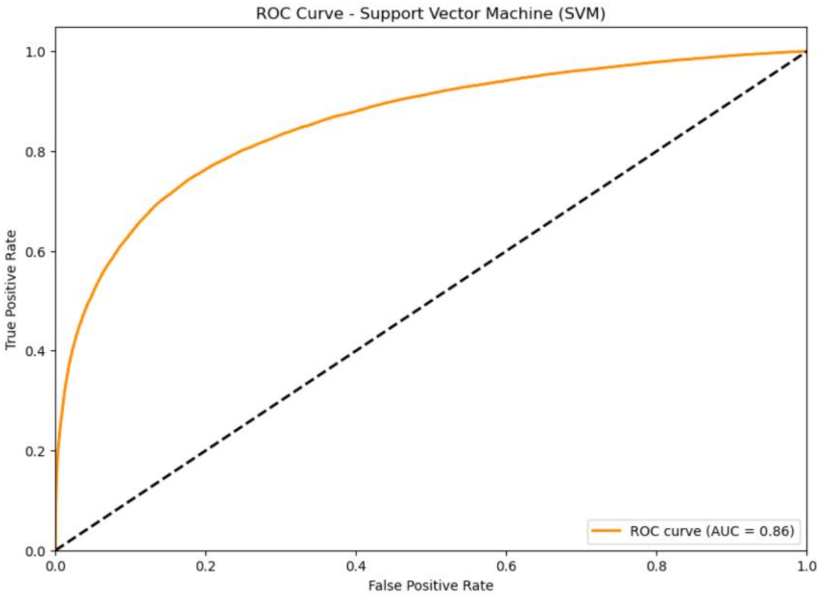
b. Support Vector Machine (SVM) Model Results

- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Support Vector Machines (SVM) model is run on our preprocessed dataset.

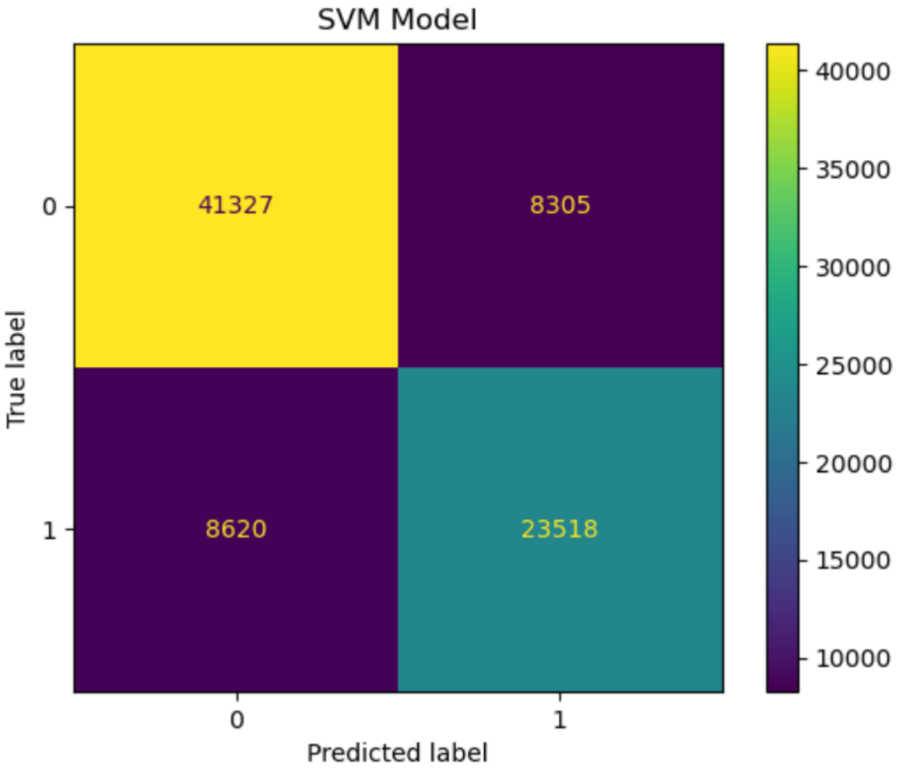


```
SVM Report:
              precision    recall  f1-score   support

           0       0.83      0.83      0.83     49632
           1       0.74      0.73      0.74     32138

    accuracy                           0.79     81770
   macro avg       0.78      0.78      0.78     81770
weighted avg       0.79      0.79      0.79     81770

Accuracy of SVM:  0.7930169988993518
```

- The following ROC (Receiver Operator Characteristics) curve shows False Positive rate vs the True Positive rate on the output of the Support Vector Machines (SVM) Model.

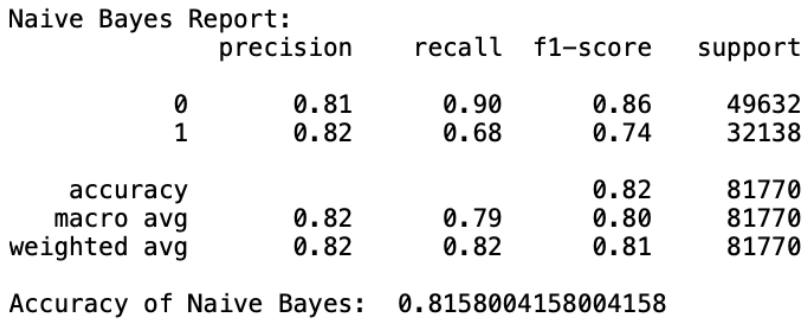


- The following Confusion Matrix shows Values between True Labels and Predicted Labels on the output of Support Vector Machines (SVM) Model.



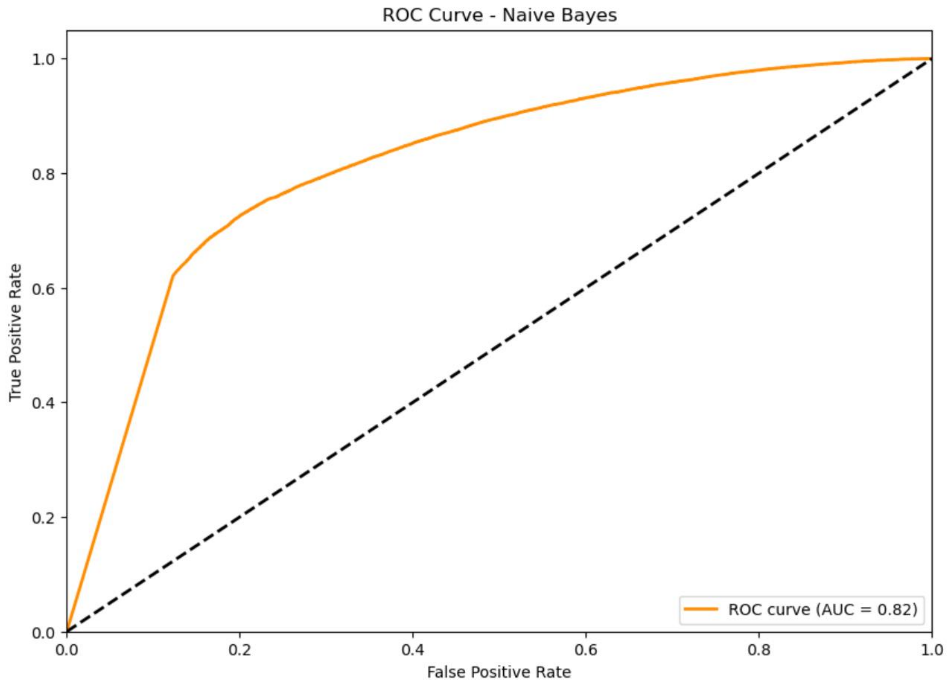
c. Naïve Bayes Model Results

- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Naive Bayes model is run on our preprocessed dataset.

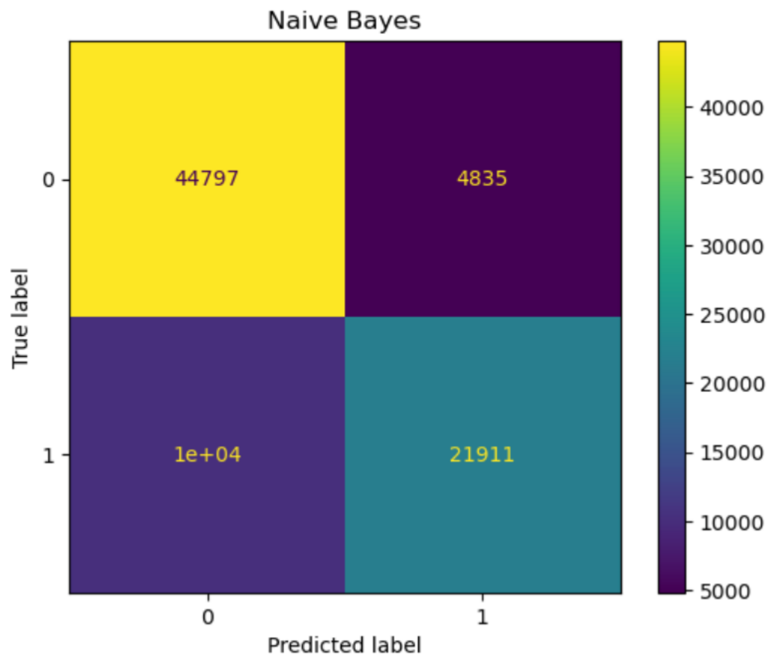


```
Naive Bayes Report:
              precision    recall  f1-score   support

           0       0.81      0.90      0.86     49632
           1       0.82      0.68      0.74     32138

    accuracy                           0.82     81770
   macro avg       0.82      0.79      0.80     81770
weighted avg       0.82      0.82      0.81     81770

Accuracy of Naive Bayes:  0.8158004158004158
```

- The following ROC (Receiver Operator Characteristics) curve shows False Positive rate vs the True Positive rate on the output of the Naive Bayes Model.

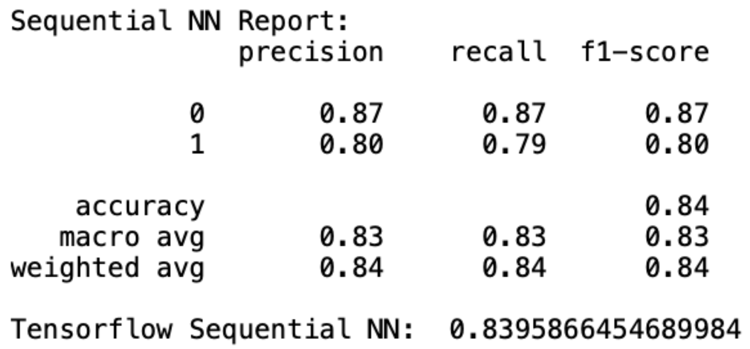


- The following Confusion Matrix shows Values between True Labels and Predicted Labels on the output of Naive Bayes Model



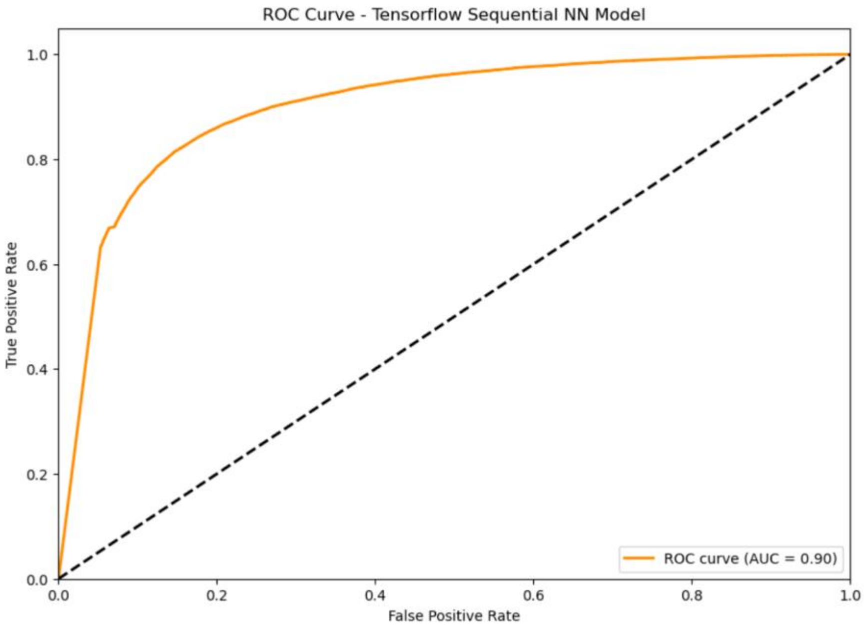
d. Sequential Neural Network (NN) Model Results:

- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Sequential Neural Network model is run on our preprocessed dataset.

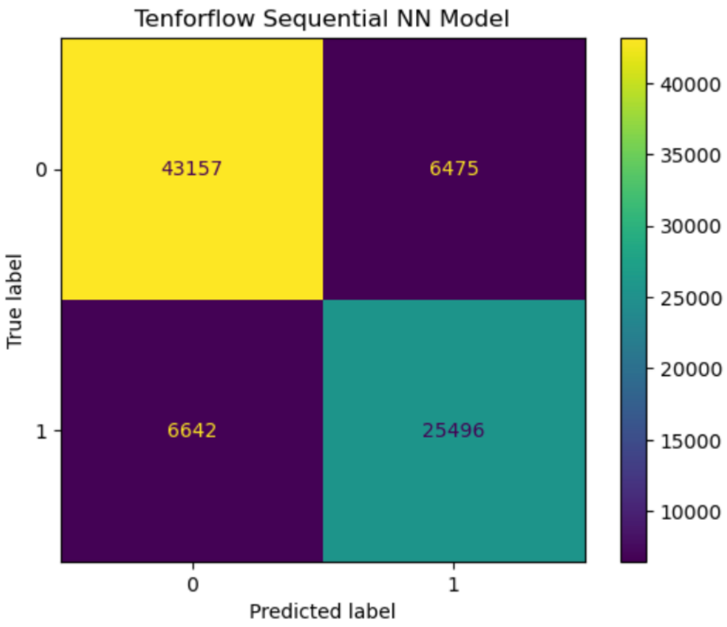


```
Sequential NN Report:
              precision    recall  f1-score   support

           0       0.87      0.87      0.87     49632
           1       0.80      0.79      0.80     32138

    accuracy                           0.84     81770
   macro avg       0.83      0.83      0.83     81770
weighted avg       0.84      0.84      0.84     81770

Tensorflow Sequential NN:  0.8395866454689984
```

- The following ROC (Receiver Operator Characteristics) curve shows False Positive rate vs the True Positive rate on the output of the Sequential Neural Network Model.

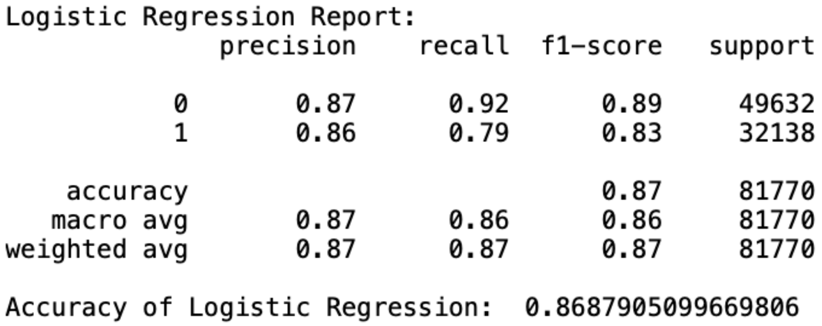


- The following Confusion Matrix shows Values between True Labels and Predicted Labels on the output of Sequential Neural Network Model



e. Logistic Regression Model Results with TF-IDF and Sentimental Polarity Features

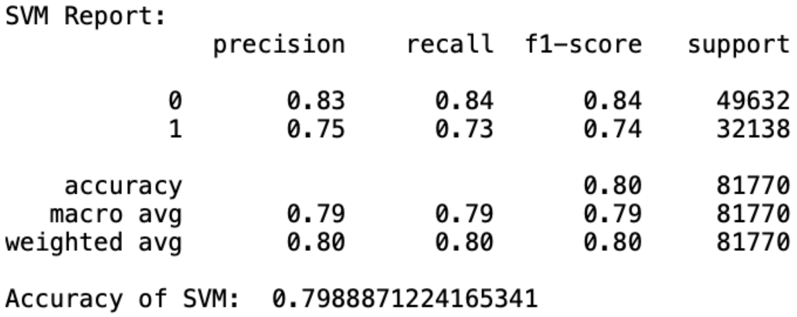
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the logistic regression model is run on the dataset with Extra 2 preprocessed features (TF-IDF) vectors and Sentimental Polarity.



```
Logistic Regression Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.89     49632
           1       0.86      0.79      0.83     32138

    accuracy                           0.87     81770
   macro avg       0.87      0.86      0.86     81770
weighted avg       0.87      0.87      0.87     81770

Accuracy of Logistic Regression:  0.8687905099669806
```

f. Support Vector Machines (SVM) Results with TF-IDF and Sentimental Polarity Features

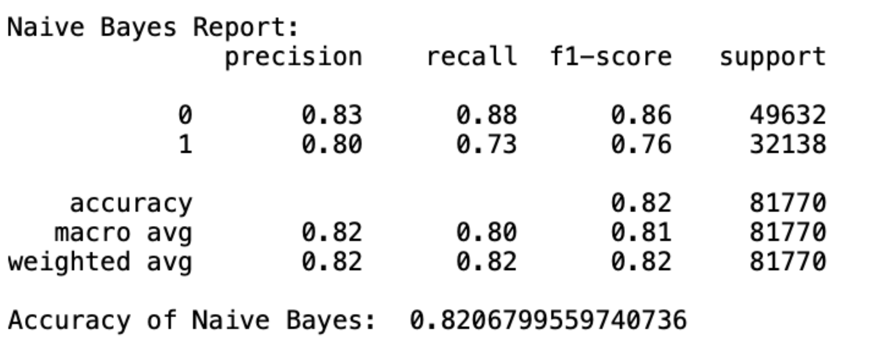
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Support Vector Machines (SVM) model is run on the dataset with Extra 2 preprocessed features (TF-IDF) vectors and Sentimental Polarity.



```
SVM Report:
              precision    recall  f1-score   support

           0       0.83      0.84      0.84     49632
           1       0.75      0.73      0.74     32138

    accuracy                           0.80     81770
   macro avg       0.79      0.79      0.79     81770
weighted avg       0.80      0.80      0.80     81770

Accuracy of SVM:  0.7988871224165341
```

g. Naive Bayes Model Results with TF-IDF and Sentimental Polarity Features

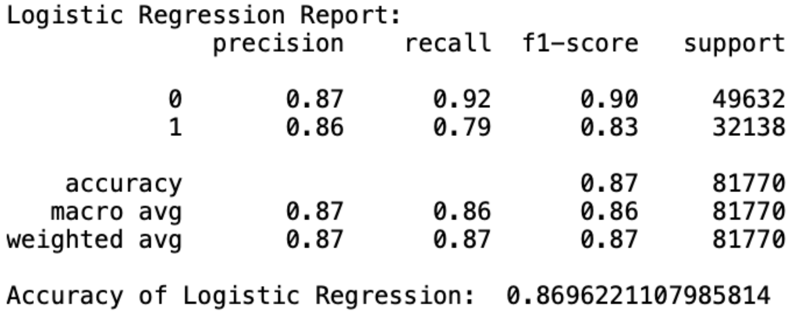
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Naive Bayes model is run on the dataset with Extra 2 preprocessed features (TF-IDF) vectors and Sentimental Polarity.



```
Naive Bayes Report:
              precision    recall  f1-score   support

           0       0.83      0.88      0.86     49632
           1       0.80      0.73      0.76     32138

    accuracy                           0.82     81770
   macro avg       0.82      0.80      0.81     81770
weighted avg       0.82      0.82      0.82     81770

Accuracy of Naive Bayes:  0.8206799559740736
```

h. Logistic Regression Model Results with all the Features combined

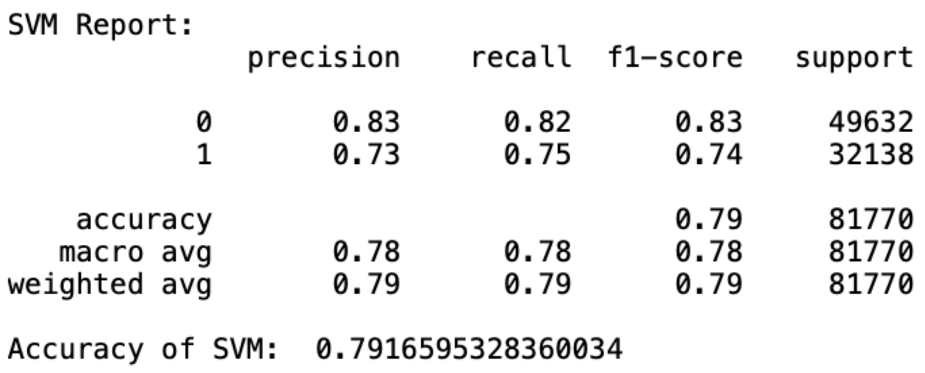
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Logistic Regression model is run on the dataset with all the features (TF-IDF) vectors, Sentimental Polarity and Doc2Vec combined.



```
Logistic Regression Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.90     49632
           1       0.86      0.79      0.83     32138

    accuracy                           0.87     81770
   macro avg       0.87      0.86      0.86     81770
weighted avg       0.87      0.87      0.87     81770

Accuracy of Logistic Regression:  0.8696221107985814
```

i. Support Vector Machines Results with all the Features combined

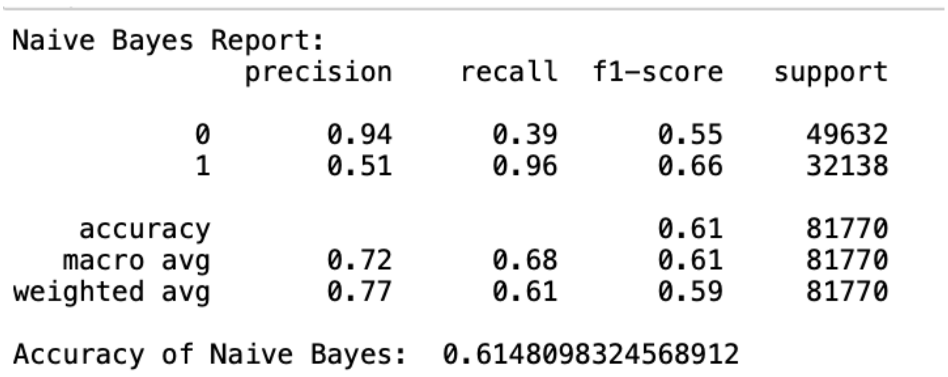
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Support Vector Machine is run on the dataset with all the features (TF-IDF) vectors, Sentimental Polarity and Doc2Vec combined.



```
SVM Report:
              precision    recall  f1-score   support

           0       0.83      0.82      0.83     49632
           1       0.73      0.75      0.74     32138

    accuracy                           0.79     81770
   macro avg       0.78      0.78      0.78     81770
weighted avg       0.79      0.79      0.79     81770

Accuracy of SVM:  0.7916595328360034
```

j. Naive Bayes Model Results with all the Features combined

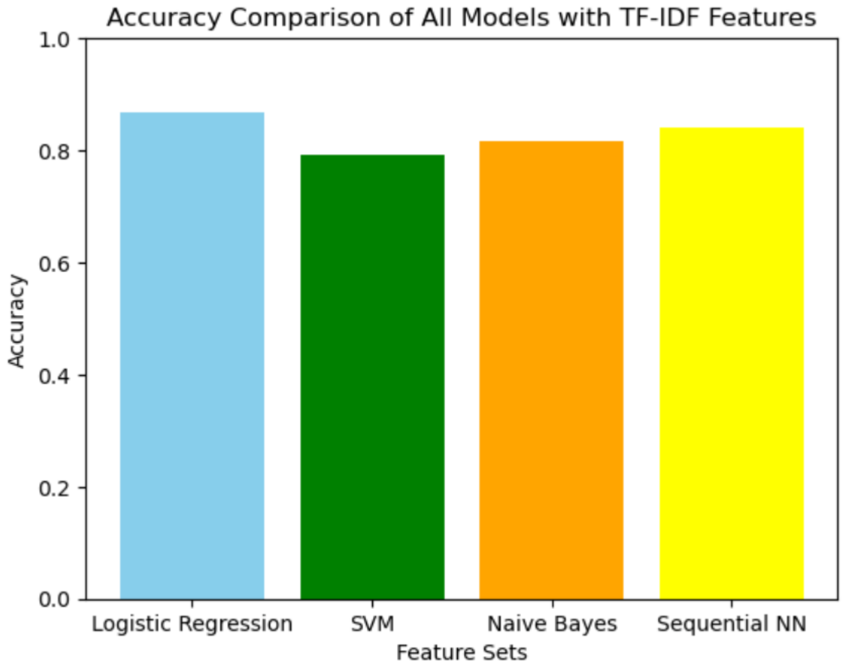
- The following accuracy plot shows the precision, recall, f-1 score and support for accuracy, macro avg and weighted avg when the Naive Bayes model is run on the dataset with all the features (TF-IDF) vectors, Sentimental Polarity and Doc2Vec combined.



```
Naive Bayes Report:
              precision    recall  f1-score   support

           0       0.94      0.39      0.55     49632
           1       0.51      0.96      0.66     32138

    accuracy                           0.61     81770
   macro avg       0.72      0.68      0.61     81770
weighted avg       0.77      0.61      0.59     81770

Accuracy of Naive Bayes:  0.6148098324568912
```

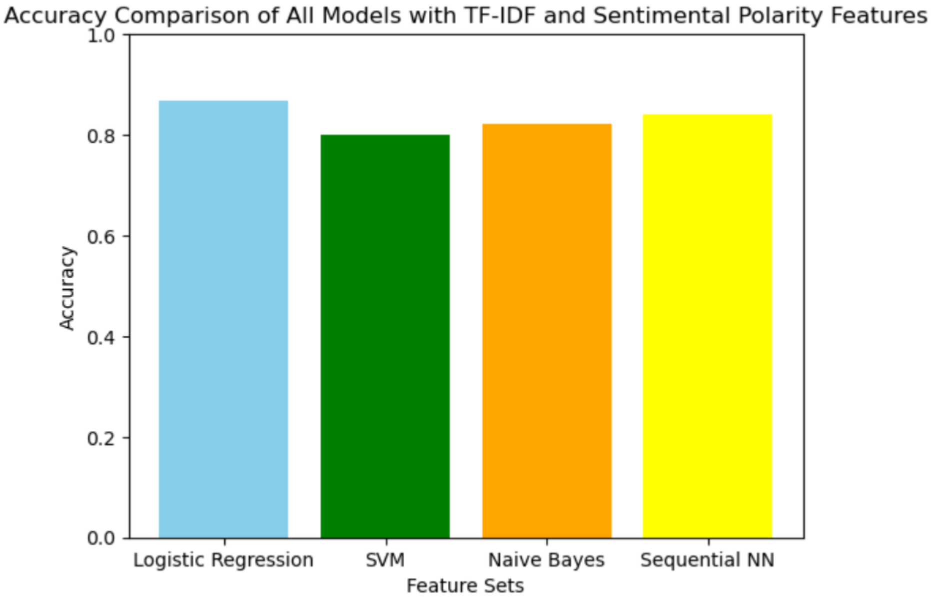
k. Plotting of all models with TF-IDF Features

- Comparison of all the models with TF-IDF vector features and all of them give very similar accuracy but logistic regression gives the most accuracy of 88%.



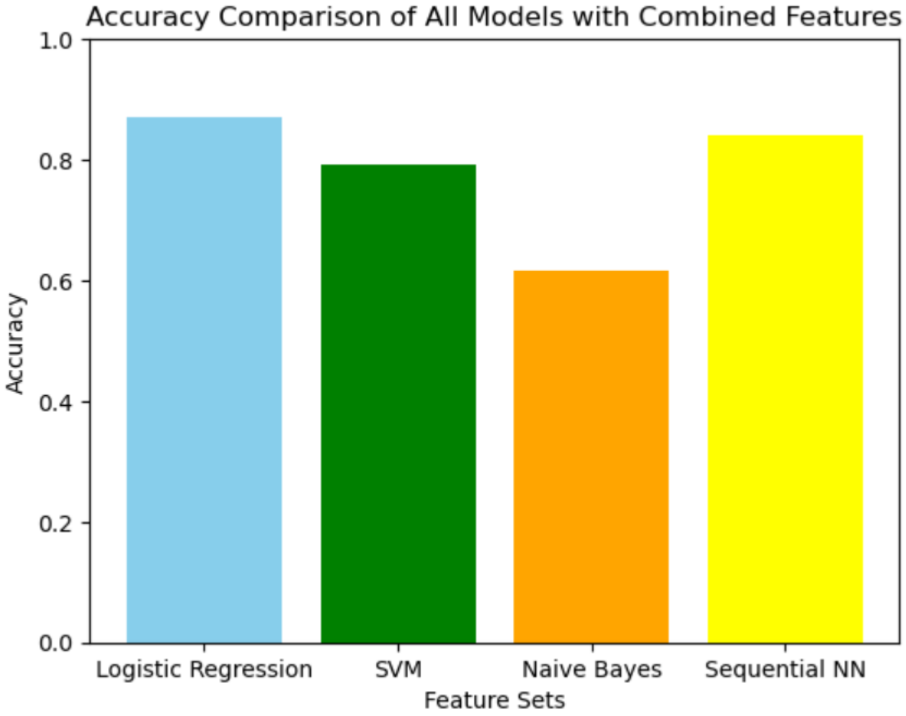
l. Plotting of all models with TF-IDF and sentimental polarity Features

- Comparison of all the models with TF-IDF vector and sentimental polarity features and all of them give very similar accuracy but logistic regression gives the most accuracy of 88%.



m. Plotting of all models with TF-IDF Features

- Comparison of all the models with all the features combined and all of them give very similar accuracy but Naive Bayes performs significantly poorly because it cannot classify negative data in Doc2Vec Vector features.



8. Conclusion

- In conclusion, after testing a range of features, we found that TF-IDF with Sentimental Polarity was the most useful feature set for all models, proving its resilience and adaptability in capturing subtleties of toxicity. Furthermore, the Sequential Neural Network Model performed better when combining different features, demonstrating its capacity to use a variety of linguistic and contextual data for precise categorization.

- When comparing individual models, Logistic Regression was the clear winner because it constantly achieved superior scores for precision, recall, and accuracy. This helped to reduce false positives and successfully identify hazardous remarks. Furthermore, the examination of Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) demonstrated the Logistic Regression superior discrimination capability. With an AUC of 0.94, this network is adept at differentiating between toxic and non-toxic comments.

- By the time this project comes to an end, it is clear that our Multi-Model Framework has advanced the state-of-the-art in harmful comment classification. Still, there is much space for more research and development, especially in terms of improving model designs, adding more contextual information, and tackling moral issues like prejudice reduction. We're determined to keep up the research and cooperation in order to build safer, more welcoming online communities in the future.

9. Future Work

- Our data point to multiple areas for future development. Growing the labeled dataset is essential to enhancing the model's versatility in terms of linguistic and cultural backgrounds. Moreover, adding characteristics pertaining to user behavior and network context could yield useful extra data and improve the accuracy of the model. We also discussed the benefits of ensemble modeling, which combines several techniques to lower errors and boost overall efficacy.

- Implementing the created model for real-time content moderation is a viable avenue for further study and use. A safer online environment would result from the quick detection and elimination of dangerous content made possible by this. Our study lays the groundwork for the creation of cutting-edge, dependable solutions that can help create a more welcoming and positive online environment as social media platforms continue to struggle with the problems of toxic comments and speeches. Toxic comments are widespread problems in online communities that can be addressed more effectively and scalably by utilizing the knowledge gathered from this study and iteratively improving the model.

- To get even more performance increases, future research might look into different feature engineering techniques, model topologies, or hyperparameter tuning.

References

- Ashish, A. Rani and H. Shyan, "A Comparative Study and Analysis on Toxic Comment Classification," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 783-787, doi: 10.1109/ICSCSS57650.2023.10169771.
- Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. 2021. A Comparative Study of Using Pre-trained Language Models for Toxic Comment Classification. In Companion Proceedings of the Web Conference 2021 (WWW '21). Association for Computing Machinery, New York, NY, USA, 500–507. https://doi.org/10.1145/3442442.3452313
- Carta, Salvatore, et al. "A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification." KDIR. 2019.
- " Jigsaw Unintended Bias in Toxicity Classification." Kaggle, 2019. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data?select=train.csv
- “Toxic Comment Classification Challenge.” Kaggle, 2018. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data?select=train.csv.zip