Text Analytics Assignment 1

Index No – 216030C

Objective

The objective of this assignment is to develop a robust NLP-based classifier model to differentiate negative tweets and prevent their dissemination on Twitter. This involves understanding and cleaning up the dataset, building classification models to predict tweet sentiments, and comparing the evaluation metrics of various classification algorithms.

Problem Selection

The problem selected for this assignment is to address the issue of hateful content misuse on Twitter. The significance of this problem lies in the need to create a safer and more positive online environment by effectively identifying and managing negative tweets. Text analytics plays a crucial role in this problem by enabling the development of a classifier model that can automatically detect and categorize negative tweets, allowing Twitter to take appropriate actions to prevent their dissemination.

Data Collection

For this project, the Twitter Tweets Sentiment Dataset from Kaggle was used. The dataset contains 27,481 entries with two columns: "text" and "sentiment". The "text" column includes the tweet text, while the "sentiment" column indicates the sentiment of the tweet (positive, negative, or neutral).

The dataset provides a sufficiently large and diverse collection of tweets, allowing for meaningful analysis of sentiment in Twitter data. The data was preprocessed to remove any null values and ensure the text is clean and ready for analysis.

Preprocessing Steps

Tokenization: Tokenization is the process of splitting text into individual words or tokens. This step is essential because it breaks down the text into manageable units for further analysis. In this project, the text was tokenized using a simple space-based tokenization method, where each word separated by a space was considered a token.

Lowercasing: Lowercasing involves converting all letters in the text to lowercase. This step ensures that words are treated consistently regardless of their original casing. For example, "Hello" and "hello" would be considered the same word after lowercasing.

Removal of Punctuation: Punctuation marks such as periods, commas, and exclamation marks were removed from the text. This step helps in focusing on the words themselves and removes unnecessary noise from the data.

Removal of Stopwords: Stopwords are common words that do not carry much meaning, such as "the", "is", "and", etc. These words were removed from the text to reduce the size of the vocabulary and improve the quality of the analysis.

Lemmatization or Stemming: Lemmatization and stemming are techniques used to reduce words to their base or root form. Lemmatization considers the context of the word and produces a valid base or root form, while stemming simply chops off prefixes or suffixes to reduce the word to its base form. In this project, stemming was used to simplify the text data.

Handling Missing Values: Any rows with missing values in the dataset were dropped to ensure data integrity. Missing values can lead to errors in analysis, so it's important to handle them appropriately.

Removing Duplicates: Duplicate entries, if any, were removed from the dataset. Duplicate entries can skew the analysis results, so it's important to identify and remove them.

Common Words Removal: Frequently occurring words that do not provide much information, such as "I'm", '-', '****', and '&', were removed from the text. These words were identified and removed to reduce noise in the data.

Rare Words Removal: Rare words that occur infrequently in the dataset were removed. These words can introduce noise into the analysis and removing them helps in focusing on the more relevant words in the dataset.

Stemming: Stemming is the process of reducing words to their root form. In this project, the PorterStemmer algorithm was used for stemming, which reduces words to their base form by removing suffixes. This step helps in reducing the size of the vocabulary and simplifying the text data for analysis.

These preprocessing steps were applied to the text data to clean and prepare it for further analysis, such as building classification models to predict tweet sentiments.

Analytics Techniques Used

Sentiment Analysis: Sentiment analysis was performed to classify the sentiment of tweets into categories such as positive, negative, or neutral. This analysis helps in understanding the overall sentiment of the tweets and can be useful for various applications, such as brand monitoring and customer feedback analysis.

Text Classification: Text classification was used to build models that can automatically classify tweets into sentiment categories. This involves training machine learning models on labeled data to predict the sentiment of unseen tweets.

TF-IDF Vectorization: TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was used to convert text data into numerical form, which can be used as input for machine learning models. This technique considers the frequency of words in a document relative to their frequency in the entire corpus, giving more weight to words that are more important in a particular document.

Logistic Regression: Logistic regression was used as a classification algorithm to predict the sentiment of tweets. It works well for binary classification problems like sentiment analysis and provides probabilities for each class.

Naïve Bayes Classifier: Naïve Bayes classifier was used as another classification algorithm for sentiment analysis. It is simple yet effective, especially for text classification tasks, and is based on the Bayes theorem with the assumption of independence between features.

Rationale Behind Chosen Techniques:

Sentiment Analysis: The objective of the project was to differentiate negative tweets and prevent their dissemination, which makes sentiment analysis a natural choice to categorize tweets based on their sentiment.

Text Classification: Text classification is a fundamental technique in natural language processing and was used to build models that can automatically classify tweets into sentiment categories, enabling Twitter to take appropriate actions on negative tweets.

TF-IDF Vectorization: TF-IDF vectorization was chosen to convert text data into numerical form because it considers the importance of words in a document relative to their frequency in the entire corpus, which is important for sentiment analysis.

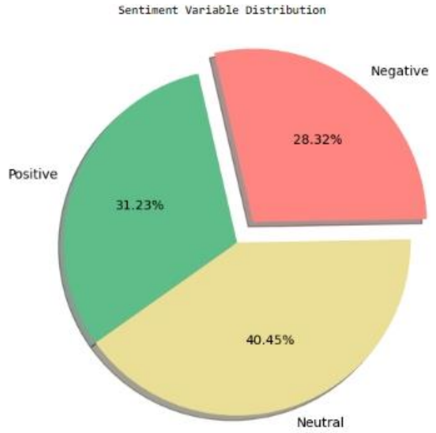
Challenges Encountered:

Computational Resources: Limited computational resources, such as memory (RAM) shortage, prevented the use of more complex models like Decision Tree, Linear Discriminant Analysis, Random Forest, and Support Vector Classifier, which require more memory and processing power. This limitation affected the ability to explore more advanced models and potentially achieve better performance.

Model Selection: Selecting the best-performing model for sentiment analysis required experimentation and evaluation of multiple algorithms. Each algorithm has its strengths and weaknesses, and finding the most suitable one for the dataset was a challenge.

Visualizations

Sentiment Distribution Analysis



The visualization under examination, entitled "Sentiment Variable Distribution," presents a comprehensive distribution of sentiments derived from an analysis of Twitter data. The graphical representation is a pie chart, segmented into three distinct categories, each denoting a different sentiment and its corresponding proportion of the total.

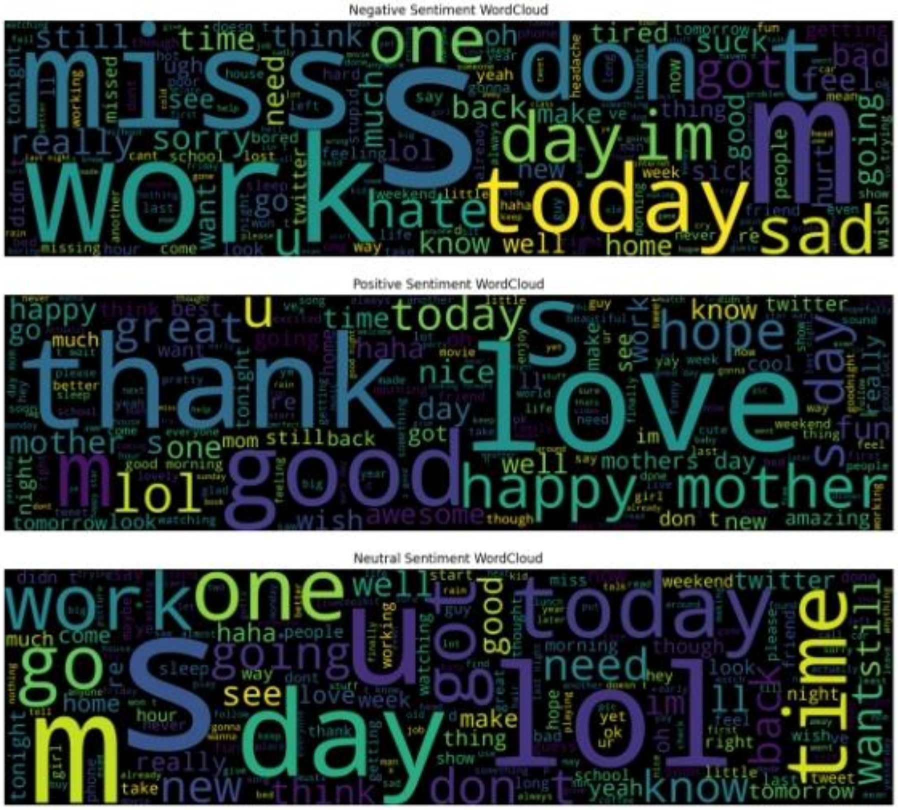
The most substantial segment, depicted in blue, signifies the 'Neutral' sentiment, constituting 40.45% of the total sentiments. This substantial proportion suggests that a significant fraction of the tweets analyzed did not distinctly express either a positive or negative sentiment, but rather conveyed a neutral or ambiguous sentiment.

The second-largest segment, depicted in green, signifies the 'Positive' sentiment, accounting for 31.23% of the total sentiments. This proportion indicates that a considerable number of tweets expressed a sentiment that can be classified as positive.

The smallest segment, depicted in red, signifies the 'Negative' sentiment, accounting for 28.32% of the total sentiments. This proportion suggests that a significant number of tweets expressed a sentiment that can be classified as negative.

This visualization provides a clear and immediate understanding of the general mood or opinions expressed in the Twitter posts analyzed. It is a valuable tool for report analysis and decision-making processes based on public opinion. The distribution of sentiments can assist in identifying trends, gauging public sentiment towards specific topics, and informing strategic decisions.

Word Cloud



Negative Sentiment WordCloud

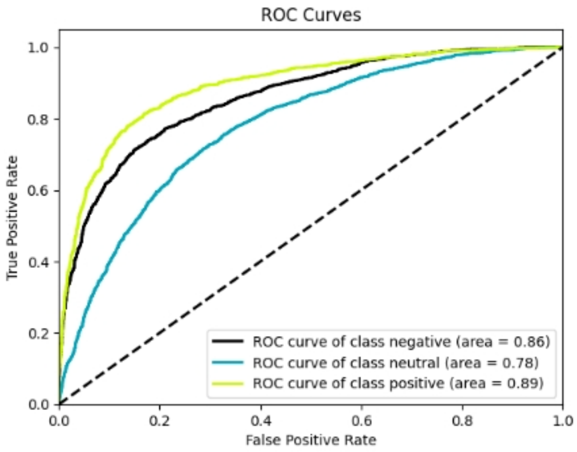
Positive Sentiment WordCloud

Neutral Sentiment WordCloud

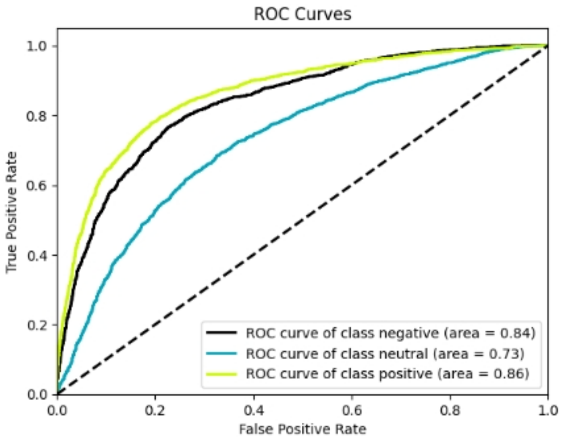
The visualizations present three word clouds generated post-cleaning from a dataset of Twitter data, each representing a different sentiment category: negative, positive, and neutral. The negative sentiment word cloud prominently displays larger words like "miss," "sad," "hate," and "sick," suggesting their frequent association with negative sentiments in the analyzed tweets. In contrast, the positive sentiment word cloud shows larger words such as "thank," "love," "good," "happy," and "mother," indicating their common occurrence in tweets expressing positive sentiment. The neutral sentiment word cloud features words like "day," "time," "work," and "one" in larger font sizes, implying their frequent appearance in tweets with neutral sentiment. The size of each word in these word clouds corresponds to its frequency of occurrence within the respective sentiment category, providing a visual representation of the most commonly associated terms for each sentiment type present in the Twitter dataset.

ROC Curve Analysis

Logistic Regression Based ROC Curves



Naïve Bayes Based ROC Curves



Logistic Regression Based Model

The visualization presents three Receiver Operating Characteristic (ROC) curves, each representing a different sentiment class derived from a dataset of Twitter data: negative (yellow), neutral (blue), and positive (green). The ROC curves display the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various classification thresholds for each sentiment class.

The Area Under the Curve (AUC) values, provided in parentheses next to the class labels, serve as a measure of the model's ability to distinguish between classes. The AUC values for the negative, neutral, and positive classes are 0.86, 0.78, and 0.89, respectively.

A higher AUC value indicates better discriminatory power of the model for a particular class. In this case, the model performs best in distinguishing the positive sentiment class, with an AUC of 0.89, followed by the negative class (AUC = 0.86), and then the neutral class (AUC = 0.78).

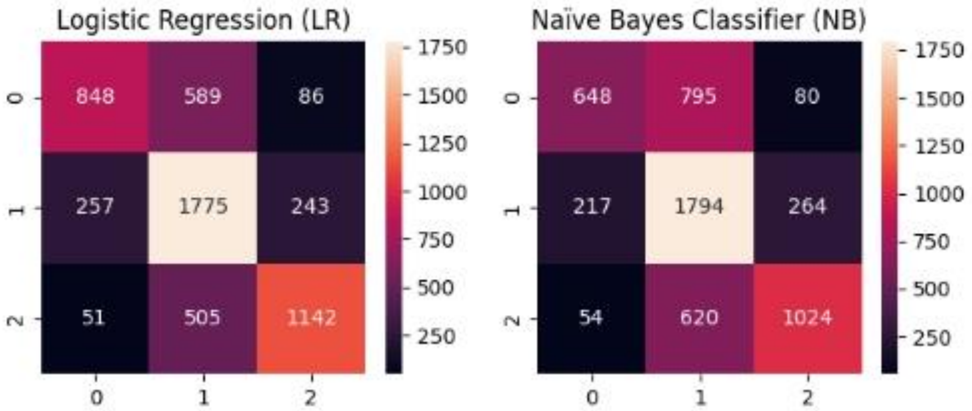
Naive Bayes Classifier Based Model

Similar to the Logistic Regression based model, the visualization displays three ROC curves for the Naive Bayes Classifier based model, each representing a different sentiment class: negative (blue), neutral (yellow), and positive (green).

The AUC values for the negative, neutral, and positive classes are 0.84, 0.73, and 0.86, respectively. Compared to the Logistic Regression model, the Naive Bayes model exhibits slightly lower performance in distinguishing the negative sentiment class (AUC = 0.84) and the neutral sentiment class (AUC = 0.73). However, it performs comparably in distinguishing the positive sentiment class, with an AUC of 0.86.

The ROC curves and AUC values provide a visual and quantitative assessment of the models' performance in classifying sentiment classes. By comparing the curves and AUC values between the two models, one can evaluate their relative strengths and weaknesses in distinguishing the different sentiment classes present in the Twitter dataset.

Heatmap



Logistic Regression Based Model (LR)

The confusion matrix for the LR model is displayed as a 3x3 heatmap, with the x-axis and y-axis representing the predicted and true classes, respectively. The diagonal elements of the matrix indicate the number of correct predictions made by the model for each class. Specifically, the model correctly predicted 848 instances of class 0, 1,775 instances of class 1, and 1,142 instances of class 2.

The off-diagonal elements of the confusion matrix represent the misclassifications made by the LR model. For instance, 58 instances of class 0 were misclassified as class 1, and 86 instances of class 0 were misclassified as class 2. Similarly, misclassifications occurred between the other class combinations, as indicated by the corresponding off-diagonal values.

Naive Bayes Classifier Based Model (NB)

The confusion matrix for the NB model is presented in a similar manner, with a 3x3 heatmap representing the predictions made on the test data. The diagonal elements show that the NB model correctly predicted 649 instances of class 0, 1,794 instances of class 1, and 1,024 instances of class 2.

Like the LR model, the NB model also exhibited misclassifications, as indicated by the off-diagonal elements of the confusion matrix. For example, 80 instances that should have been classified as class 0 were misclassified as class 2.

The color intensity of each cell in the heatmaps corresponds to the numerical value represented, with darker shades indicating higher numbers. This visual representation allows for an immediate comparison of the performance of the two models across the three classes, highlighting their strengths and weaknesses in terms of accurate predictions and misclassifications.

Conclusion

The objective of this project was to develop a robust NLP-based classifier model to differentiate negative tweets and prevent their dissemination on Twitter. The problem of identifying and managing negative or hateful content on social media platforms is critical for creating a safer and more positive online environment.

The analysis involved cleaning and preprocessing a dataset of Twitter tweets, building classification models using techniques like logistic regression and naive Bayes, and evaluating the performance of these models through visualizations such as confusion matrices and ROC curves.

One of the key challenges faced during this project was the computational resource limitation, which prevented the exploration of more complex models like decision trees, random forests, and support vector machines. Additionally, selecting the best-performing model for sentiment analysis required experimentation and evaluation of multiple algorithms.

Based on the evaluation metrics and visualizations, both the logistic regression and naive Bayes models showed promising results in distinguishing between positive, negative, and neutral sentiments. However, it is important to note that the dataset exhibited an imbalanced distribution of sentiments, with a significant proportion of neutral tweets.

In such cases, where the classes are imbalanced and accurately identifying the sentiment is crucial, the F1 score becomes a relevant metric to consider. The F1 score, which is the harmonic mean of precision and recall, provides a balanced measure of a model's performance by combining these complementary metrics into a single value.

Moving forward, it would be valuable to explore techniques for handling imbalanced datasets, such as oversampling or undersampling, and to investigate more advanced models like ensemble methods or deep learning approaches. Additionally, incorporating contextual information and handling sarcasm or irony could further improve the accuracy of sentiment analysis models.

Overall, this project demonstrates the potential of natural language processing and machine learning techniques in addressing the challenge of identifying negative content on social media platforms. By continuously refining and enhancing these models, we can contribute to creating a safer and more positive online environment for all users.