NATURAL LANGUAGE PROCESSING

**TWITTER HATE SPEECH IDENTIFICATION**

# 1.0 INTRODUCTION

Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI, concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. Natural Language Processing (NLP) has numerous applications across various fields. According to Sarvandani (2023), some of the top applications of NLP include Sentiment Analysis, Language Translation, Text Summarization, Named Entity Recognition (NER), Question Answering**,** Chatbots and Virtual Assistants, Information Extraction,Text Classification, Text Generation, Speech Recognition and Voice Assistants e.t.c.

This study focuses on applying natural language processing techniques on a hate speech identification dataset, distinguishing sentiments expressed in tweets as either hate speech, offensive language or neither. By leveraging NLP methods, the goal is to build a robust classifier capable of accurately identifying and categorizing tweets of different sentiments. The insights gained from this analysis can provide valuable information for social media platforms, policymakers and other stakeholders involved in addressing online hate speech. Understanding the patterns and occurrence of hate speech on Twitter can help make informed strategies which can be developed to reduce its impact and protect users from harmful content

## 1.1 OBJECTIVES

The specific objectives of this project include:

* Applying natural language processing (NLP) algorithms to tweets expressing sentiments classified as hate speech, offensive language, or neither.
* Developing a robust machine learning classifier that can reliably recognize and classify tweets according to their attitudes.
* Learn about the trends and instances of hate speech in Twitter data, which can be helpful for a variety of stakeholders, including social media companies, legislators, and groups that deal with hate speech online.

## 1.2 DATASET DESCRIPTION

This is the content of the dataset for the Hate speech identification problem.

* count = number of CrowdFlower users who coded each tweet (min is 3, sometimes more users coded a tweet when judgments were determined to be unreliable by CF).
* hate\_speech = number of CF users who judged the tweet to be hate speech.
* offensive\_language = number of CF users who judged the tweet to be offensive.
* neither = number of CF users who judged the tweet to be neither offensive nor non-offensive.
* class = class label for majority of CF users. 0 - hate speech 1 - offensive language 2 - neither
* tweet = raw tweet text

# 2.0 METHODOLOGY

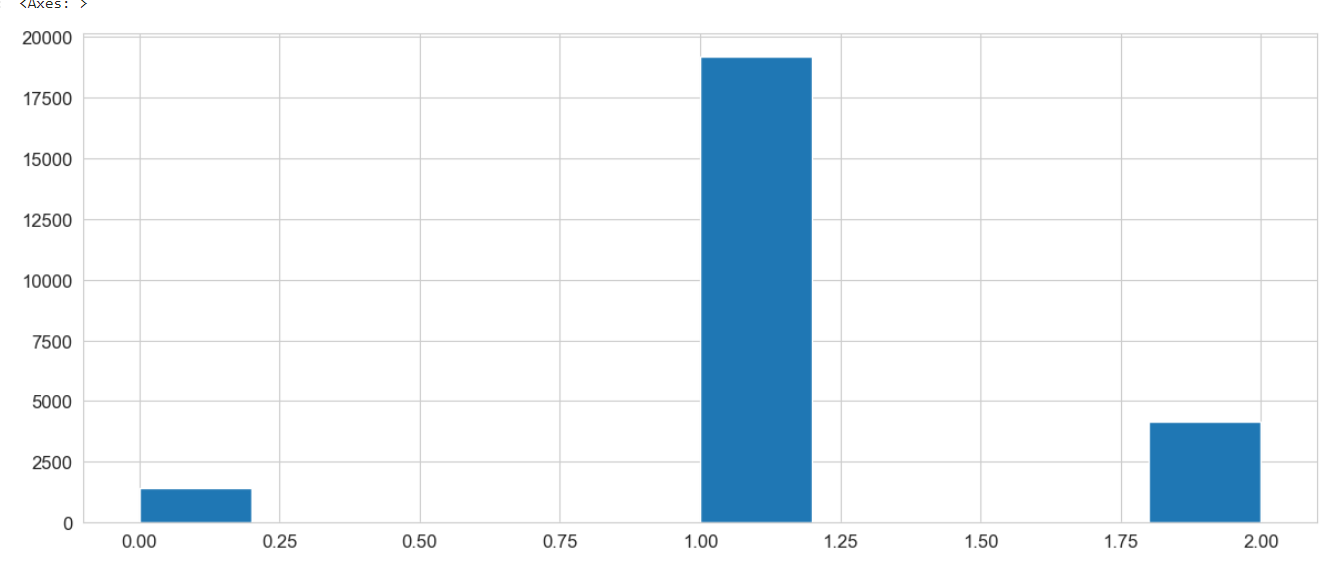
The methodology for the project involved data preprocessing, exploratory data analysis, feature engineering, model selection and evaluation.

## 2.1 DATA PREPROCESSING

The first step was preparing the raw data, which included things like figuring out how it was distributed and filling in any blanks. To be more precise, a thorough cleaning process was applied to the text data in the tweet column using bespoke routines. These routines were created to remove a variety of items, including URLs (web addresses), stopwords (frequent words with no semantic meaning), diacritical marks (accented characters), multiple spaces, tags, and digits (numerical characters). Text normalization techniques were also used to normalize the text data and guarantee its uniformity and cleanliness. These included word reductions to their base or dictionary form (lemmatization), word reductions to their root form (stemming), and text tokenization (Tweetokenizer) into individual words or tokens designed for tweets. All together, these preprocessing procedures sought to ensure that the data was consistent, free of unnecessary components, and able to yield valuable insights in later phases of the project, readying it for analysis.

## 2.2 EXPLORATORY DATA ANALYSIS

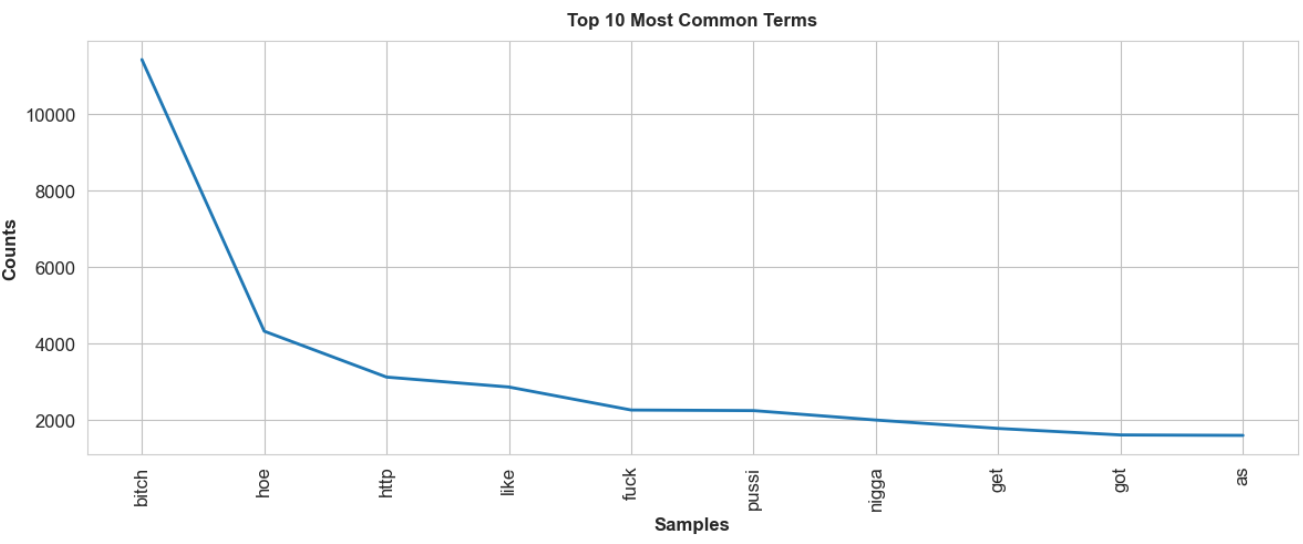
Following the first round of data preprocessing, a thorough exploratory data analysis (EDA) was carried out. This required a careful analysis of the dataset's many components in order to understand its fundamental properties. Initially, the dataset's class distribution was examined to see how well-balanced or unbalanced the various categories and labels were. To further understand the typical length of tweets in the dataset—which may be a sign of the type of content or user behavior—the distribution of tweet lengths was also carefully examined. In addition, a word cloud was created to show the frequency of words in the dataset visually, emphasizing the terms that are most frequently used both generally and within each class or group.



*Fig. 1-Distribution of Classes*



*Fig. 2-Most Commonly Used words*



*Fig. 3-Distribution of Common words*

## 2.3 FEATURE ENGINEERING

Feature engineering was an essential part of the methodology, coming after exploratory data analysis and data pretreatment. During this phase, a number of cutting-edge approaches were applied in order to improve the dataset and extract useful information from the tweets' textual content. SentimentIntensityAnalyzer, Word2Vec, and LDA were the three main algorithms that were used to complete different tasks.

First, sentiment analysis was done on the text data using SentimentIntensityAnalyzer, which made it possible to extract sentiment-related features. The objective of this research was to extract the sentiment or emotional tone from the tweets, adding important sentiment-related data to the dataset.

Furthermore, the widely used word embedding method Word2Vec was utilized to generate numerical depictions of the words included in the dataset. By converting words into dense vectors of real numbers, this method captures the semantic connections between words according to how they are used in context. Word embeddings that convey semantic interpretations were added to the dataset by integrating Word2Vec, allowing for a more in-depth examination of the textual content.

Finally, a probabilistic topic modeling method called Latent Dirichlet Allocation (LDA) was used to find latent topics in the twitter data. Using patterns of word co-occurrence, LDA finds underlying subjects within a set of documents and allocates words to these topics. The methodology's use of LDA was intended to uncover any underlying themes or subjects within the tweets, adding more depth to the research and interpretation process.

When combined, these feature engineering techniques added additional features to the dataset and yielded a more in-depth understanding of the tweets' textual content. The strategy aimed to improve the dataset's representational strength and identify significant patterns and insights that could aid in the ensuing stages of analysis and modeling by utilizing these cutting-edge methodologies.

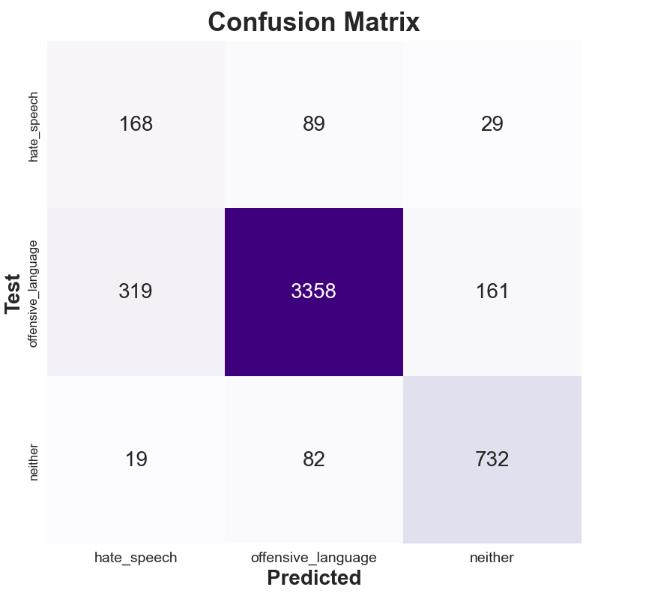
## 2.4 MODEL SELECTION AND EVALUATION

The methodology concentrated on selecting suitable base models and resolving the imbalanced nature of the dataset throughout the model selection and evaluation phase. In particular, because they work well for classification tasks, Gradient Boosting Classifier and Logistic Regression were chosen as the foundational models. The Synthetic Minority Over-sampling Technique (SMOTE) and RandomUnderSampler were used in a resampling method to assist balance the distribution of classes in the dataset and lessen the impact of class imbalance. The text data was transformed into numerical vectors for tweet preprocessing using TF-IDF (Term Frequency-Inverse Document Frequency). These methods converted the textual material into a format that machine learning algorithms could understand. Tokenization and preprocessing procedures were carried out in order to guarantee that the input data was correctly formatted and prepared for analysis before model training.

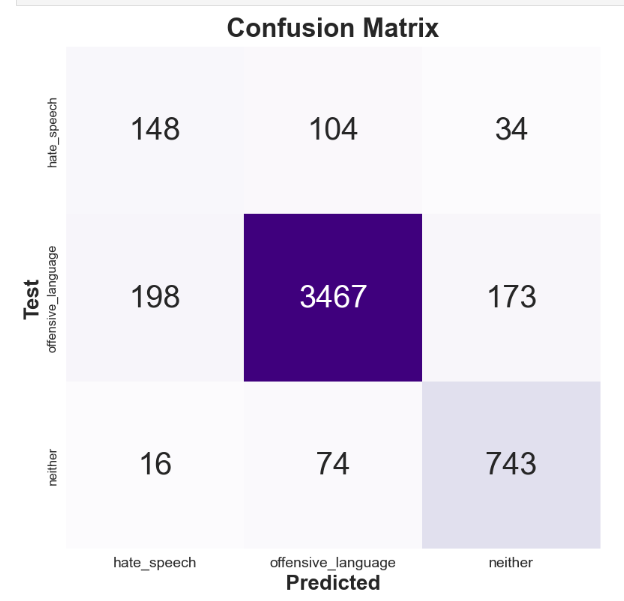
The approach looked at applying advanced deep learning techniques in addition to conventional machine learning models. Using the preprocessed data, an LSTM (Long Short-Term Memory) model was trained to take advantage of its capacity to identify long-range dependencies in sequential data.

This methodology took a thorough approach to addressing the difficulties involved in modeling, analyzing, and preparing the data in order to identify hate speech. The systematic workflow meant that each stage contributed to the larger aim of constructing a successful classifier, while simultaneously offering useful insights into the features of the data and the effectiveness of different modeling strategies. The goal of this method was to produce a thorough and knowledgeable analysis of the identification of hate speech in Twitter data.

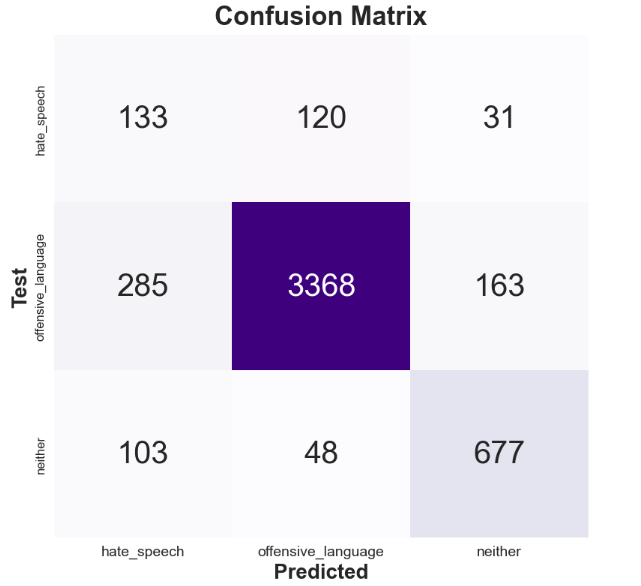
# 3.0 RESULTS

This shows the distribution of predicted classes using the Logistic Regression model.****

This is the distribution of predicted classes of the test set using Gradient Boosting Classifier.



This is the distribution of predicted classes of the test set using the LSTM model.



# 4.0 DISCUSSION

**Logistic Regression Model**

The precision scores show how well the model can identify instances of each class. With scores of 95% and 79% for offensive language (class 1) and neither (class 2), respectively, the model demonstrated good precision; however, its precision for hate speech (class 0) was only 33%. This shows that when it came to neutral and offensive tweets, the model performed better than when it came to hate speech. The recall ratings show how well the model can identify examples of each class. For all classifications, the model showed a rather good recall rate, catching 59% of hate speech cases, 87% of offensive language cases, and 88% of neutral cases. This shows that, despite significantly worse performance, the model was successful in finding instances across all classes, with a slightly lower performance in predicting hate speech instances compared to other classes. The F1-scores offer a fair evaluation of the model's performance because they take precision and recall into account. Even though hate speech (class 0) had a lower F1-score of 0.42 than offensive language (class 1), which had a higher score of 0.91, and neither class 2, which had a higher score of 0.83, it still shows that the precision and recall for hate speech classification are reasonably balanced. The model's overall accuracy of 86% shows that it can accurately categorize tweets into the appropriate categories. To enhance its performance in predicting the imbalanced class, the model would need hyperparameter tuning, as indicated by the lower precision and F1-score for hate speech.

**Gradient Boosting Classifier Model**

The GradientBoosting Classifier model's precision scores matched those of the Logistic Regression model; it scored highly for neither (class 2) nor offensive language (class 1), at 95% and 78%, respectively, but it scored worse for hate speech (class 0), at 41%. This suggests a similar pattern, where the model was more successful in recognizing neutral and offensive tweets than hate speech. The model captured 52% of hate speech cases, 90% of offensive language instances, and 89% of neutral instances, according to recall ratings that mirrored the Logistic Regression model. This indicates that the model performed reasonably well in capturing instances of all classes, with the exception of hate speech instances. The GradientBoosting Classifier model's F1-scores, which were 0.46 for hate speech (class 0), 0.90 for offensive language (class 1), and 0.83 for neither (class 2), demonstrated the proper balance between precision and recall. For every class, these results show a fair mix between recall and precision, with foul language and neither receiving especially high marks. The model's total accuracy of 88% shows that it can correctly categorize tweets into the appropriate categories. But similarly with the Logistic Regression model, the poorer precision for hate speech indicates that this particular class needs work.

**LSTM Model**

In comparison to the Logistic Regression and Random Forest Classifier models, the LSTM model's precision scores were significantly lower in all classes: 26% for hate speech (class 0), 95% for offensive language (class 1), and 78% for neither (class 2). This suggests that in comparison to the other classes, the LSTM model was less successful in detecting instances of hate speech. Similar trends were seen in the LSTM model's recall scores, which showed that the model correctly identified 47% of hate speech instances, 88% of offensive language instances, and 82% of neutral language instances. The memory for hate speech was noticeably lower than the recall for offensive language, suggesting that the algorithm had trouble capturing instances of hate speech effectively. The model's F1-scores, which were 0.33 for class 0 and 0.92 for class 1 and 0.80 for class 2, demonstrated the poorer precision and recall for hate speech. These results show a notable disparity in the model's performance between the various classes, with the model performing worse overall in identifying hate speech. This model's ability to classify tweets into their appropriate categories was demonstrated by its 85% overall accuracy. To improve the model's performance in this particular class, use of pretrained models can help learn from the features better, as indicated by the lower precision, recall, and F1-score for hate speech statistics.

# 5.0 CONCLUSION

In summary, using a dataset of labeled tweets, this project attempted to solve the challenge of categorizing tweets into groups of hate speech, offensive language, or neither. Through the application of machine learning algorithms and natural language processing (NLP) techniques, the research aimed to create a strong classifier that could reliably recognize and classify tweets according to their moods. In order to educate measures to lessen its impact and shield users from offensive content, the initiative also sought to shed light on the trends and occurrences of hate speech on Twitter.

The methodology used a step-by-step process, starting with data pretreatment to clean and get the text data ready for analysis. Next, in order to learn more about the properties of the dataset, exploratory data analysis (EDA) and feature engineering were performed to extract pertinent features for modeling. To determine which machine learning model performed best for the task, a number of models were trained and assessed, including Random Forest Classifier, Logistic Regression, and LSTM.

The models' performance was evaluated for each class, and the findings showed differences in accuracy, precision, recall, and F1-score. No single model scored consistently well across all classes, despite the fact that each model showed strengths in specific domains, such as the Random Forest Classifier's excellent precision for offensive language. Each model's advantages and disadvantages were emphasized during the conversation, which also shed light on how well each worked and possible areas for development.

In general, this effort advances the knowledge of how to categorize hate speech in social media data. The created models can be useful instruments for locating and responding to hate speech on Twitter, and the knowledge gathered from this analysis can help shape tactics to stop hate speech online and shield users from its negative impacts. To enhance the models' functionality and ability to be applied to larger datasets and social media platforms, more investigation and model improvement might be required.

# 6.0 REFERENCES

1. Sarvandani, M. (2023) 'Top 15 applications of NLP + Python codes', Medium. Available at: <https://medium.com/@mohamadhasan.sarvandani/top-applications-of-nlp-54834ca9c8ca> (Accessed: [14th January 2024])
2. IBM (2023) 'What is natural language processing?', [Online]. Available at: <https://www.ibm.com/topics/natural-language-processing> (Accessed: [14th January 2024])
3. Mazumder, S. (2023) '5 Techniques to Handle Imbalanced Data For a Classification Problem', Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/> (Accessed: [15th January 2024])
4. Williams, L. (2020) 'Sentiment Analysis: Aspect-Based Opinion Mining', Towards Data Science. Available at: <https://towardsdatascience.com/%EF%B8%8F-sentiment-analysis-aspect-based-opinion-mining-72a75e8c8a6d> (Accessed: 15th January 2024)
5. Tate, A. (2023) 'A Comprehensive Guide to Natural Language Processing Algorithms', Data. Available at: <https://hex.tech/blog/comprehensive-guide-to-natural-language-algorithms/> (Accessed: 15th January 2024)