International Journal of Advanced Engineering Research and Science (IJAERS)
Peer-Reviewed Journal
ISSN: 2349-6495(P) | 2456-1908(O)
Vol-11, Issue-3; Mar, 2024
Journal Home Page Available: https://ijaers.com/
Article DOI: https://dx.doi.org/10.22161/ijaers.114.1

Navigating the Dark Web of Hate: Supervised Machine Learning Paradigm and NLP for Detecting Online Hate Speeches

Njideka Nkemdilim Mbeledogu and Mishael Somtochukwu Ike-Okonkwo

Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria

Received: 03 Feb 2024,

Receive in revised form: 08 Mar 2024,

Accepted: 19 Mar 2024,

Available online: 30 Mar 2024

***Keywords*— Natural Language Processing, Tokenization, Logistic Regression, Hyperparameter**

***Abstract*— Many online platform's participants are worried about hate speeches that usually trigger cyberbully attitudes that dissuades users' interest in their platforms. The study investigates hate speech in online platforms using Natural Language Processing (NLP) techniques and supervised machine learning paradigm. It specifically focused on developing a robust model capable of classifying text as 'hateful' or 'non-hateful' accurately. The approaches applied included compiling a large dataset from multiple online textual sources; preprocessing the dataset through normalization, tokenization, stop-word removal, and lemmatization; advanced feature extraction techniques such as negation handling, n-gram analysis, and Term Frequency-Inverse Document Frequency (TF-IDF) to capture the intricacies of the textual material and the model implementation phase using Logistic Regression for its efficiency in binary classification problems. The model's performance was evaluated using metrics such as accuracy, precision, recall, F1-score and confusion matrix. The baseline performance of the model with default hyperparameters achieved a test accuracy of 93%. When optimized with hyperparameter tuning and cross-validation procedures to guarantee more generalizable performance, the model achieved an accuracy of 95%. The study concluded that NLP and logistic regression technique can effectively identify hate speeches.*

I. INTRODUCTION

The growing frequency of hate speech on online communication platforms poses a major danger to the digital age's guiding principles of inclusivity, tolerance, and respectful conversation. The internet's obscurity has given people the confidence to indulge in abusive language, thereby, establishing harm as normal way of life. Popular initiatives aimed at reducing hate speech frequently rely on manual content moderation. This approach is cumbersome, time-consuming, resource-intensive and biased. Yet more, the dynamic and ever-changing character of online conversation makes it difficult to effectively recognize and respond to hate speech in real time.

Addressing the issues of hate speech needs provident approach and adequate technologies that can quickly detect it. Based on this, the research aims to use Natural Language Processing (NLP) and machine learning techniques to create a hate speech detection and sentiment analysis system that automates the detection of hate speeches based on the emotional tones to improve the safety and civility of digital communication spaces.

II. LITERATURE REVIEW

Hate speech can be said to be expressions that belittle, extricate, or support both physical and emotional violence based on any socio-attributes such as religion and ethnicity.

The UN Strategy and Plan of Action on hate speech defined it as any kind of communication in speech, writing or behavior that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, on their religion, ethnicity, nationality, race, color, descent, gender or other identity factor" (United Nations, n.d). It is characterized by expressions that demean, discriminate, or incite violence and poses significant threats to the well-being of a man and the world at large.

Hate speeches have been on increase not only among peers but also political and religious leaders. The high rate of social media and online comments have provided users with eccentric avenues to voice their opinions without any regards. This democratization of expression has necessitated this also. From targeted harassment campaigns by political elites to the least of common man, its impact on individuals and societies cannot be overemphasized.

An agreement was reached that online platforms have the responsibility to mitigate the exigencies of hate speech while upholding principles of free speech and open dialogue. In the light of this, many actions have been taken to address the occurrences of hate speeches by online platforms, pressure groups and governments, thus, creating the need for the use of supervised machine learning paradigm and natural language processing to mitigate this.

Supervised Machine Learning Paradigm

This is the learning approach of machines when under supervision whereby labeled data are used in the form of input-output pairs. The major tasks of this type of learning are regression, classification and forecasting (Kotsiantis, 2007).

Natural Language Processing (NLP)

The field of NLP is a branch of Artificial Intelligence that focuses on the interaction between humans and computers using natural language (Johnson, 2023). It leverages on computational linguistics and machine learning techniques to analyze and understand human language. By developing sophisticated algorithms and models, researchers and practitioners in NLP can automate machine translation, speech recognition, information retrieval, spam detection, text summarization, intelligent web searching, intelligent spell checking and human-computer communication.

Review of Related Work

Zhang *et al*. (2018) worked on "Detecting hate speech on Twitter using a convolution-GRU based deep neural network". The paper introduced a new method based on a deep neural network combining convolutional and gated recurrent networks. The authors conducted an extensive evaluation of the method against several baselines and state of the art on the largest collection of publicly available Twitter datasets to date. The researchers' proposed method captured both word sequence and order information in short texts.

Khanday *et al*. (2022) delved into detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques. The authors carried out hate speech detection using machine learning and ensemble learning techniques during COVID-19. The twitter data used were extracted using the publicly available twitter API with the help of trending hashtags during the COVID-19 pandemic. The tweets were manually annotated into two categories based on different factors. Feature extraction was performed using Term Frequency/Inverse Document Frequency (TF/IDF), Bag of Words and Tweet Length. The study found the Decision Tree Classifier to be effective when compared to other typical Machine Learning (ML) classifiers. It had 98% precision, 97% recall, 97% F1-Score, and 97% accuracy.

Rodriguez *et al*. (2022) developed a framework for detection and integration of unstructured data of hate speech on Facebook using sentiment and emotion analysis. The aim of the research was to locate and analyze the unstructured data of selected social media posts that intend to spread hate in the comment sections. To address this issue, they proposed a novel framework called FADOHS, which combines data analysis and natural language processing strategies to sensitize all social media providers to the pervasiveness of hate on social media. Specifically, they used sentiment and emotion analysis algorithms to analyze recent posts and comments on these pages. Posts suspected of containing dehumanizing words will be processed before fed to the clustering algorithm for further evaluation. According to the experimental results, the proposed FADOHS framework surpassed the state-of-the-art approach in terms of precision, recall, and F1 scores by approximately 10%.

Pamungkas *et al*. (2020) on "Do you really want to hurt me? Predicting abusive swearing in social media". They explored the phenomenon of swearing in Twitter conversations, taking the possibility of predicting the abusiveness of a swear word in a tweet context as the main investigation perspective. They developed the Twitter English corpus SWAD (Swear Words Abusiveness Dataset), where abusive swearing was manually annotated at the word level. Their collection consists of 1,511 unique swear words from 1,320 tweets. They developed models to automatically predict abusive swearing to provide an intrinsic evaluation of SWAD and confirm the robustness of the resource. They also presented the results of a glass box ablation study in order to investigate which lexical, syntactic and effective features that are more informative towards the automatic prediction of the function of swearing.

Zimmerman *et al*. (2019) researched on improving hate speech detection with deep learning ensembles. They utilized a publicly available embedding model and tested against a hate speech corpus from Twitter. To confirm the robustness of their results, they additionally tested against a popular sentiment dataset. Their method had a nearly 5 point improvement in F-measure when compared to original work on a publicly available hate speech evaluation dataset. The major difficulties they encountered was reproducibility of deep learning methods and comparison of findings from other work.

Yun *et al*. (2023) worked on BERT-Based logits ensemble model for gender bias and hate speech detection. They aimed to solve the problem on gender bias and hate speech detection, and to detect malicious comments in a Korean hate speech dataset constructed in 2020. They explored bidirectional encoder representations from transformers (BERT)-based deep learning models utilizing hyperparameter tuning, data sampling, and logits ensembles with a label distribution. They evaluated the model in Kaggle competitions for gender bias, general bias, and hate speech detection. For gender bias detection, an F1-score of 0.7711 was achieved using an ensemble of the Soongsil-BERT and KcELECTRA models. The general bias task included the gender bias task, and the ensemble model achieved the best F1-score of 0.7166.

Siino *et al*. (2021) analyzed the detection of hate speech spreaders using convolutional neural network. The authors developed a deep learning model based on a convolutional neural network (CNN) for the profiling hate speech spreaders (HSSs). Their classification (HSS or not HSS) takes advantage of the CNN based on a single convolutional layer. In this binary classification task, they performed tests using a 5-fold cross validation, in which the proposed model reached a maximum accuracy of 0.80 on the multilingual (i.e., English and Spanish) training set, and a minimum loss value of 0.51 on the same set. The trained model presented was able to reach an overall accuracy of 0.79 on the full test set.

Mozafari *et al*. (2019) worked on a BERT-Based transfer learning approach for hate speech detection in online social media. The study introduced a novel transfer learning approach based on an existing pre-trained language model called Bidirectional Encoder Representations from Transformers (BERT). The transfer learning-based fine-tuning techniques to explore BERT's capacity to detect hateful context in social media content. To evaluate the proposed approach, they made use of two publicly available datasets that have been annotated for racism, sexism, hate, or offensive content on Twitter. The results showed that their solution could obtain considerable performance on these datasets in terms of precision and recall in comparison to existing approaches. Also, their model captured some biases in data annotation and collection process and can potentially lead to a more accurate model.

III. RESEARCH METHODOLOGY

This section outlines the methodology adopted for sentiment analysis on hate speeches using a supervised learning approach. The chosen approach involves training models on labeled datasets, leveraging the rich body of research and techniques in supervised learning for sentiment classification.

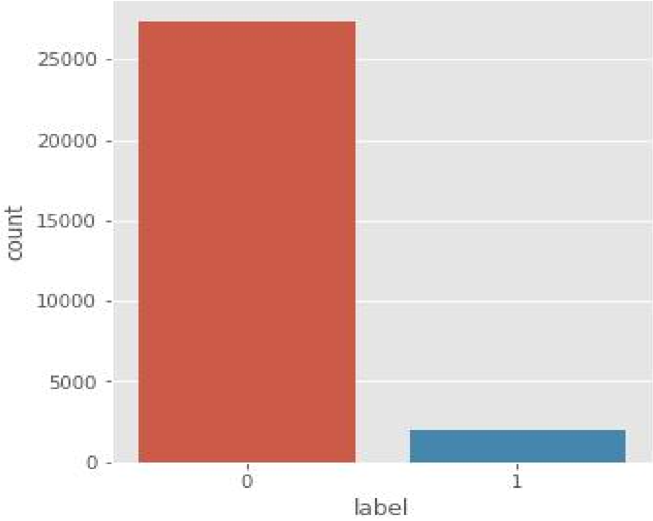
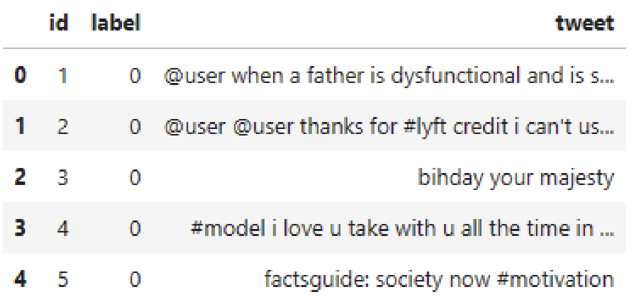


Fig.1: Graph of the data used for analysis (0-Non Hate Speech1- Hate Speech)

Data Collection

The dataset was collected from Twitter and contains a diverse set of tweets from various sources and user backgrounds, spanning over a year of data collection. The `hateDetection_train.csv` dataset utilized consists of 31964 tweets in total, with 93.2% labeled as hateful and 6.8% as non-hateful, making it an imbalanced dataset as seen in Figure 1.

To ensure transparency and reproducibility, it is crucial to provide a detailed account of the dataset's origin, size, and composition. The first step in understanding the dataset was loading it into a Pandas DataFrame. Figure 2 shows the first 5 tweets visualized from the dataset after loading it into the Pandas DataFrame.



	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

Fig.2: Raw testing data (Top 5 tweets)

Subsequently, a comprehensive exploration of the dataset was essential. This included calculating descriptive statistics, such as the mean tweet length, character distribution, and class distribution (i.e., the number of hateful and non-hateful tweets). Visualizations, such as word clouds, can also provide valuable insights into the most common words used in each category. Figure 3 shows the processes involved in hate speech detection.

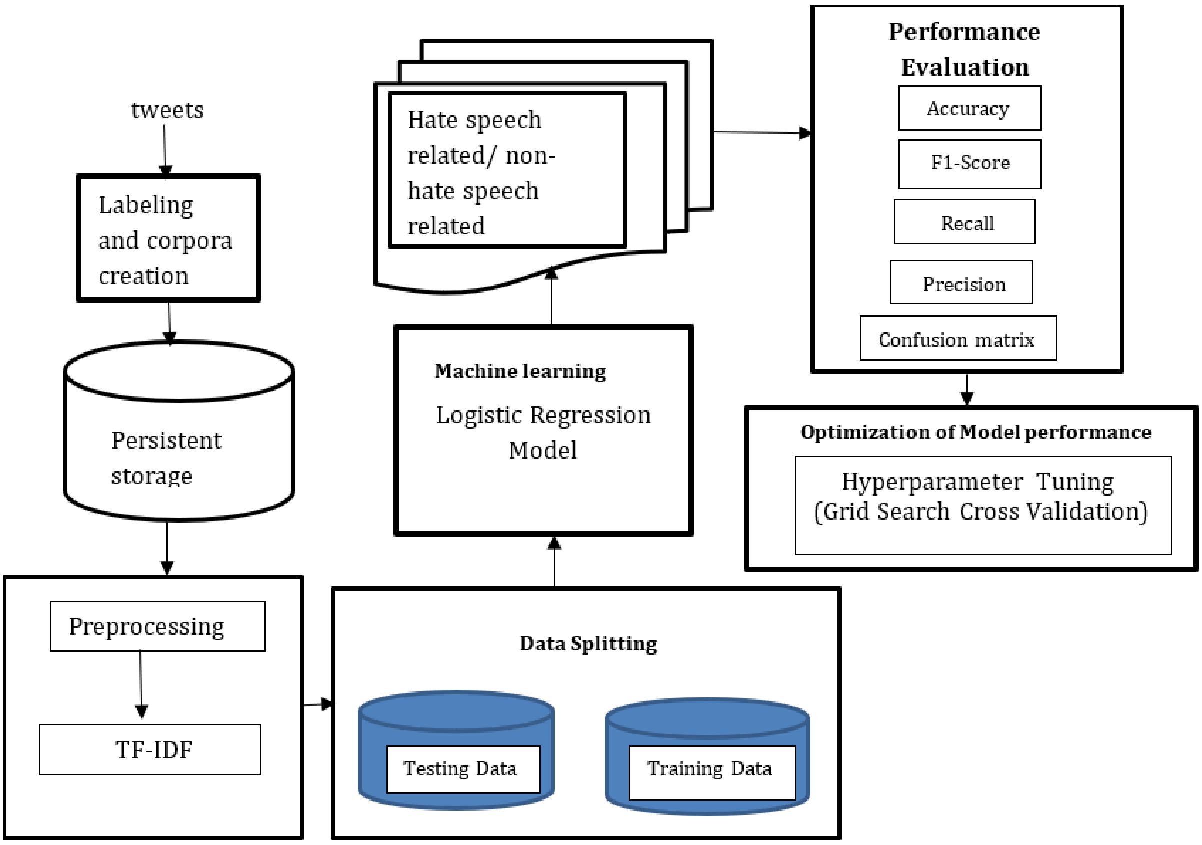

tweets
Labeling and corpora creation
Persistent storage
Preprocessing
TF-IDF
Data Splitting
Testing Data
Training Data
Machine learning
Logistic Regression Model
Hate speech related/ non-hate speech related
Performance Evaluation
Accuracy
F1-Score
Recall
Precision
Confusion matrix
Optimization of Model performance
Hyperparameter Tuning (Grid Search Cross Validation)

Fig.3: Hate Speech Detection System

To prepare the text data for modeling, the following preprocessing steps were applied. Figure 4 also shows the code block that carried out these preprocessing steps:

a. Removal of URLs, mentions, and hashtags: These elements do not carry significant semantic meaning and can be safely removed.

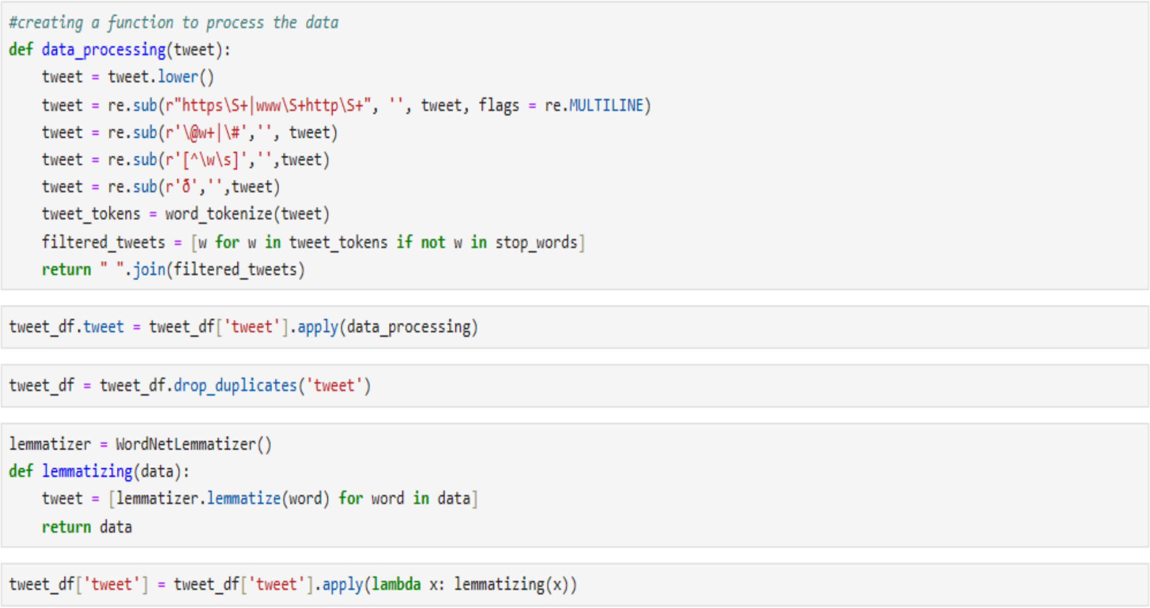
b. Conversion to lowercase: To ensure consistency in word representation and avoid treating the same word differently due to case variations.

c. Handling special characters and emojis: Special characters and emojis are retained as they may convey sentiment or context.

d. Stop word removal: Common words like "the," "and," and "in" are removed as they carry little informative value.

e. Lemmatization: Reducing words to their root forms helps in capturing the core meaning of words.

f. Duplicate tweet removal: Duplicate tweets are removed to prevent bias in the training process.



```
#creating a function to process the data
def data_processing(tweet):
    tweet = tweet.lower()
    tweet = re.sub(r"https\S+|www\S+http\S+", '', tweet, flags = re.MULTILINE)
    tweet = re.sub(r'\@w+|\#','', tweet)
    tweet = re.sub(r'[^\w\s]','',tweet)
    tweet = re.sub(r'ð','',tweet)
    tweet_tokens = word_tokenize(tweet)
    filtered_tweets = [w for w in tweet_tokens if not w in stop_words]
    return " ".join(filtered_tweets)
```

```
tweet_df.tweet = tweet_df['tweet'].apply(data_processing)
```

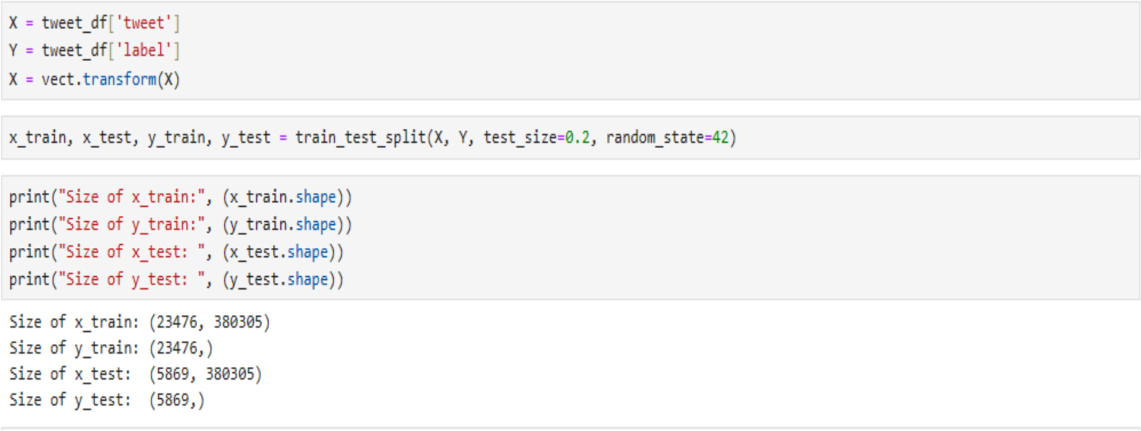
```
tweet_df = tweet_df.drop_duplicates('tweet')
```

```
lemmatizer = WordNetLemmatizer()
def lemmatizing(data):
    tweet = [lemmatizer.lemmatize(word) for word in data]
    return data
```

```
tweet_df['tweet'] = tweet_df['tweet'].apply(lambda x: lemmatizing(x))
```

Fig.4: Code block for Data preprocessing

Data Splitting: To evaluate model performance effectively, the dataset was split into training (80%) and testing (20%) sets using a random split with a fixed random state. This ensures reproducibility and allows me to assess the model's generalization ability on unseen data. Figure 5 shows the code block used in splitting the dataset.



```
X = tweet_df['tweet']
Y = tweet_df['label']
X = vect.transform(X)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

```
print("Size of x_train:", (x_train.shape))
print("Size of y_train:", (y_train.shape))
print("Size of x_test: ", (x_test.shape))
print("Size of y_test: ", (y_test.shape))
```

```
Size of x_train: (23476, 380305)
Size of y_train: (23476,)
Size of x_test:  (5869, 380305)
Size of y_test:  (5869,)
```

Fig.5: Data Splitting Code Block Module

Feature Extraction

Feature extraction is a crucial aspect of natural language processing tasks. In the research, the textual data was represented as numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. It measures the importance of words in a document relative to the entire corpus. The experiment was carried out with different n-gram ranges (1-2 and 1-3) to investigate the impact of capturing word sequences on model performance as seen in Figure 7. This experiment is vital as it helped to identify which textual features are most informative for hate speech detection. Figures 6 and 7 show the word cloud of the most frequent words in hate speech tweets and the feature extraction code block respectively.

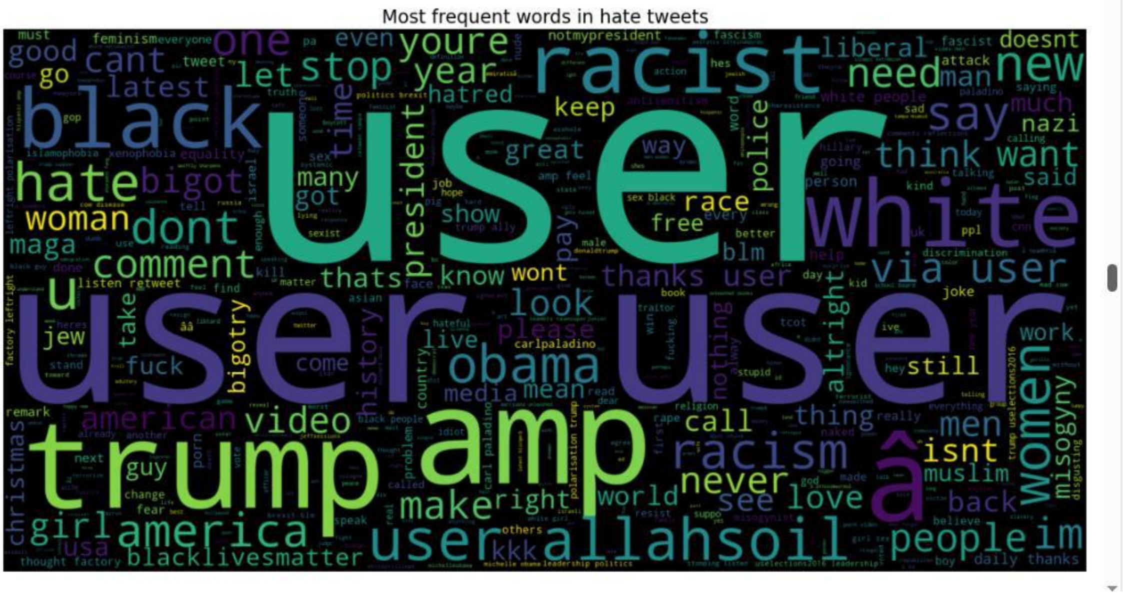
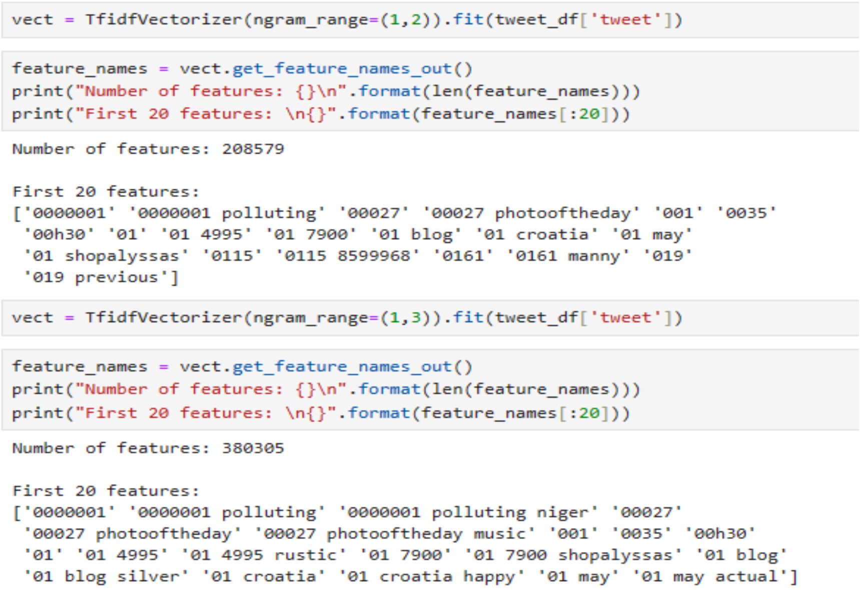


Fig.6: Word Cloud of the hate speech detected tweets

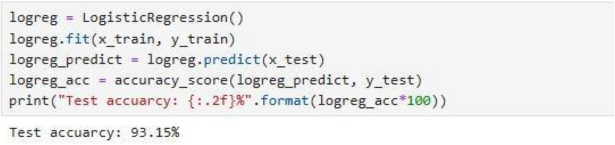


```
vect = TfidfVectorizer(ngram_range=(1,2)).fit(tweet_df['tweet'])

feature_names = vect.get_feature_names_out()
print("Number of features: {}\n".format(len(feature_names)))
print("First 20 features: \n{}".format(feature_names[:20]))

Number of features: 208579

First 20 features:
['0000001' '0000001 polluting' '00027' '00027 photooftheday' '001' '0035'
 '00h30' '01' '01 4995' '01 7900' '01 blog' '01 croatia' '01 may'
 '01 shopalyssas' '0115' '0115 8599968' '0161' '0161 manny' '019'
 '019 previous']

vect = TfidfVectorizer(ngram_range=(1,3)).fit(tweet_df['tweet'])

feature_names = vect.get_feature_names_out()
print("Number of features: {}\n".format(len(feature_names)))
print("First 20 features: \n{}".format(feature_names[:20]))

Number of features: 380305

First 20 features:
['0000001' '0000001 polluting' '0000001 polluting niger' '00027'
 '00027 photooftheday' '00027 photooftheday music' '001' '0035' '00h30'
 '01' '01 4995' '01 4995 rustic' '01 7900' '01 7900 shopalyssas' '01 blog'
 '01 blog silver' '01 croatia' '01 croatia happy' '01 may' '01 may actual']
```

Fig.7: Feature Extraction Code Block

Model Selection and Training

After extensive experimentation with various machine learning algorithms, Logistic Regression was selected as the most suitable model for the binary classification task of hate speech detection. Logistic Regression is well-suited for this task due to its simplicity, interpretability and effectiveness in handling textual data as seen in Figure 8.



```
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_predict = logreg.predict(x_test)
logreg_acc = accuracy_score(logreg_predict, y_test)
print("Test accuarcy: {:.2f}%".format(logreg_acc*100))

Test accuarcy: 93.15%
```

Fig.8: Code block of Logistic Regression

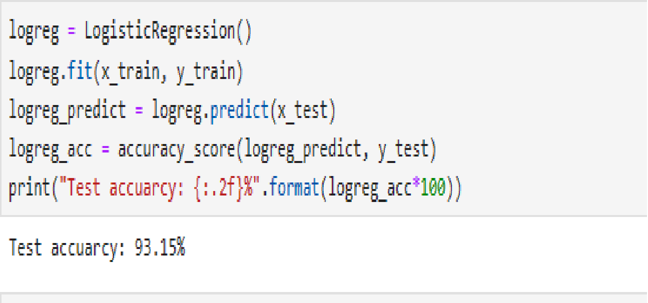
The performance of the Logistic Regression model heavily depends on its hyperparameters. To optimize these hyperparameters, Grid Search Cross-Validation was employed. The grid search explored different combinations of hyperparameters (C and solver) and selected the ones that can maximize the model's performance on the validation set. This process enhances the model's ability to discriminate between hateful and non-hateful tweets.

Performance Evaluation

Five performance metrics were used – Accuracy, Confusion matrix, Precision, Recall and F1-Score.

Accuracy: This is the primary evaluation metric used in the research. It measures the proportion of correctly classified tweets. While accuracy provides an overall assessment of model performance, it may not be sufficient for imbalanced datasets. Figure 9 depicts the code block for determining accuracy. It is calculated as:

$$Accuracy = \frac{Number\ of\ correctly\ classified\ tweets}{Total\ number\ of\ tweets} \quad (1)$$

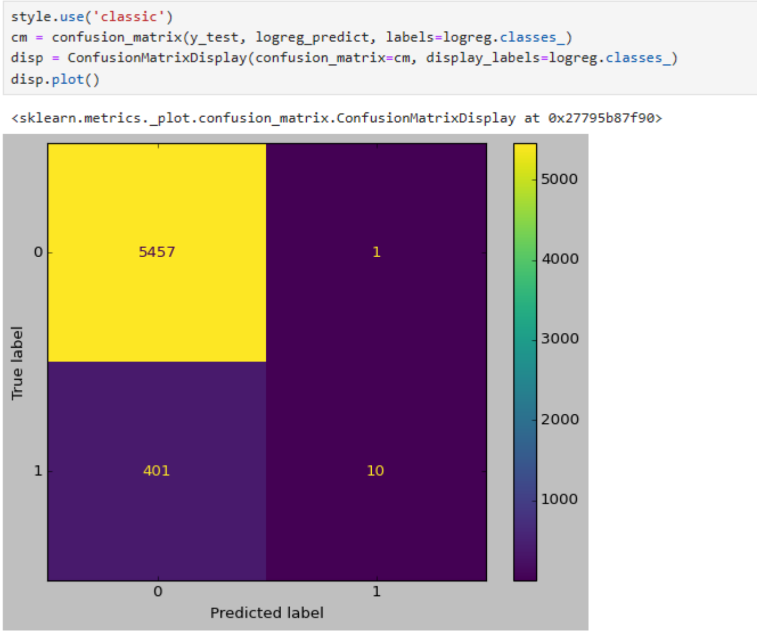


```
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_predict = logreg.predict(x_test)
logreg_acc = accuracy_score(logreg_predict, y_test)
print("Test accuarcy: {:.2f}%".format(logreg_acc*100))
```

```
Test accuarcy: 93.15%
```

Fig.9: Code block for calculating the accuracy of the model

Confusion Matrix: To gain deeper insight into model performance, a confusion matrix that visualizes the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) was employed as shown in Figure 10. This information helped to identify specific patterns of errors made by the model, such as whether it tends to have more false positives or false negatives.



```
style.use('classic')
cm = confusion_matrix(y_test, logreg_predict, labels=logreg.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=logreg.classes_)
disp.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27795b87f90>
```

Fig.10: Confusion Matrix Display

Precision: This is used to determine the exactness of the measurement. It measures the accuracy of positive predictions. It is calculated as:

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \quad (2)$$

Recall: This measures the completeness of positive predictions, that is, measure of how well a model correctly identifies True Positives. Equ. (3) shows its calculation:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (3)$$

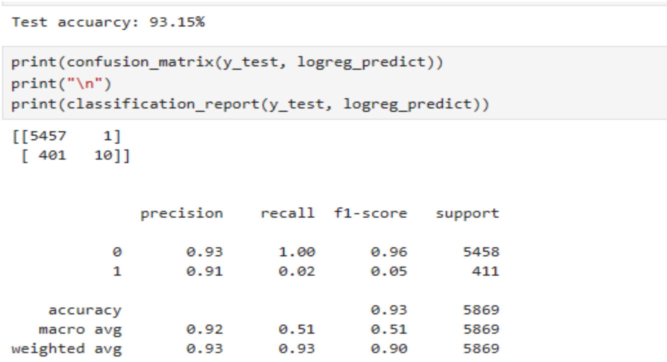
F1-Score: A measure of a model's accuracy on a dataset. It is a harmonic means of both precision and recall of the model. It is determined as:

$$F1_{Score} = 2\ X\ \frac{(precision\ x\ recall)}{(precision + recall)} \quad (4)$$

IV. RESULTS AND DISCUSSIONS

The performance metrics analysis provides insights into the model's effectiveness in distinguishing 'hateful' from 'non-hateful' texts and discusses potential factors influencing misclassifications.

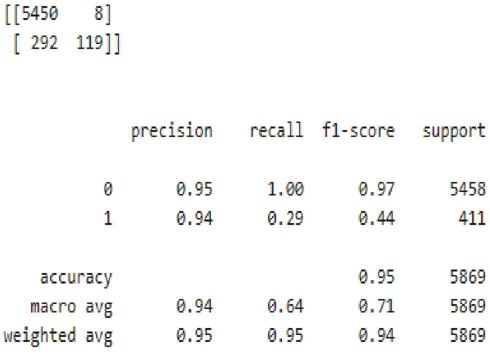
The baseline performance of the model with default hyperparameters, achieved a test accuracy of 93% as shown in Figure 11. This baseline performance serves as a reference point for evaluating the effectiveness of subsequent improvements.



```
Test accuarcy: 93.15%

print(confusion_matrix(y_test, logreg_predict))
print("\n")
print(classification_report(y_test, logreg_predict))

[[5457    1]
 [ 401   10]]

              precision    recall  f1-score   support

           0       0.93      1.00      0.96      5458
           1       0.91      0.02      0.05       411

    accuracy                           0.93      5869
   macro avg       0.92      0.51      0.51      5869
weighted avg       0.93      0.93      0.90      5869
```

Fig.11: Baseline performance without hyperparameter tuning

Tuning hyperparameter enhances the performance of a model. Through the grid search cross-validation, the hyperparameters of the logistic regression model was optimized, resulting in an improved accuracy of 95% as seen in Figure 12. The optimal hyperparameters are determined to be C = 0.1 and solver = newton-cg.



```
[[5450    8]
 [ 292  119]]

              precision    recall  f1-score   support

           0       0.95      1.00      0.97      5458
           1       0.94      0.29      0.44       411

    accuracy                           0.95      5869
   macro avg       0.94      0.64      0.71      5869
weighted avg       0.95      0.95      0.94      5869
```

Fig. 12: Performance Evaluation after hyperparameter tuning

V. CONCLUSION

The research focused on creating a machine learning model for detecting hate speech in online textual content using NLP techniques. Based on the performance evaluation, Logistic Regression model showed reliable results in classifying text as either a hate speech or non-hate speech.

REFERENCES

[1] United Nations (n.d). What is hate speech? Retrieved from https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech

[2] Johnson, .A. (2023). NLP Vs Computational Linguistics: Understanding the Differences. Retrieved from https://medium.com/@andrew_johnson_4/nlp-vs-computational-linguistics-understanding-the-differences-57044aa41ad2.

[3] Khanday, A. M. U. D., Rabani, S. T., Khan, Q. R., and Malik, S. H. (2022). Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques. International Journal of Information Management Data Insights, *2*(2), Pgs. 100-120.

[4] Kotsiantis, S.B. (2007). Supervised Machine Learning: A Review of Classification Techniques, Informatica 31, Pgs. 249-268.

[5] Mozafari, M., Farahbakhsh, R. and Crespi, N. (2020). A BERT-based transfer learning approach for hate speech detection in online social media. In Complex Networks and Their Applications VIII: Volume 1 Proceedings of the 8th International Conference on Complex Networks and Their Applications, COMPLEX NETWORKS 2019 Vol.8, Pgs. 928-940. Springer International Publishing.

[6] Pamungkas, E. W., Basile, V. and Patti, V. (2020, May). Do you really want to hurt me? Predicting abusive swearing in social media. In Proceedings of the 12th Language Resources and Evaluation Conference, Pgs. 6237-6246.

[7] Rodriguez, A., Chen, Y. L. and Argueta, .C. (2022). FADOHS: framework for detection and integration of unstructured data of hate speech on facebook using sentiment and emotion analysis, IEEE Access, 10, Pgs. 22400-22419.

[8] Siino, M., Di Nuovo, E., Tinnirello, I. and La Cascia, M. (2021). Detection of hate speech spreaders using convolutional neural networks. In *CLEF (*Working Notes*)*, Pgs. 2126-2136.

[9] Wang, Z. and Cha, Y. J. (2021). Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. Structural Health Monitoring, 20 (1), Pgs. 406-425.

[10] Yun, S., Kang, S. and Kim, H. (2023). BERT-Based Logits Ensemble Model for Gender Bias and Hate Speech Detection. Journal of Information Processing Systems, 19 (5).

[11] Zhang, D., Mao, R., Song, X., Wang, D., Zhang, H., Xia, H., and Gao, Y. (2023). Humidity sensing properties and respiratory behavior detection based on chitosan halloysite nanotubes film coated QCM sensor combined with support vector machine. Sensors and Actuators B: Chemical, 374, 132824.

[12] Zhang, Z., Robinson, D. and Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. In The Semantic Web: 15th International Conference Proceedings, ESWC 2018, Heraklion, Crete, Greece, June 3–7, Springer International Publishing, Pgs.745-760.

[13] Zimmerman, S., Kruschwitz, U. and Fox, C. (2018). Improving hate speech detection with deep learning ensembles. In Proceedings of the 11th international conference on language resources and evaluation (LREC 2018).