International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 01 - 07, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1001

Machine Learning Approach to Determine the Impact of Hate Speech Based on Public Opinions

Obilikwu Patrick
Benue State University
Makurdi, Nigeria

Charles Obekpa
Benue State University
Makurdi, Nigeria

Aamo Iorliam
Benue State University
Makurdi, Nigeria

Abstract: In an era dominated by online communication, the prevalence of hate speech has emerged as a significant societal concern. This paper presents a comprehensive study utilizing machine learning techniques to assess the detrimental effects of hate speech on public opinions. Utilizing a varied dataset encompassing public sentiments, we utilize advanced machine learning algorithms to categorize the adverse effects of hate speech. This study employs a combination of quantitative, qualitative, and experimental research methodologies. The experimentation phase involved training and testing the models using Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF). The resulting model demonstrates high accuracy and precision, with LR achieving (0.96, 0.97), DT showing (0.99, 100), and RF exhibiting (100, 100). Our findings reveal compelling insights into the negative impact of hate speech. By classifying the negative repercussions, this research not only advances our understanding of online discourse but also provides a valuable foundation for the development of strategies to combat hate speech and cultivate a more inclusive digital environment.

Keywords: Hate Speech, predictive modeling, opinion mining, machine learning

1. INTRODUCTION

In an era characterized by unprecedented connectivity and information dissemination, the rise of hate speech has emerged as a pressing concern for both online. The pervasive nature of digital platforms has provided a fertile ground for the proliferation of harmful and discriminatory language, amplifying its potential to inflict profound societal harm [1]. Addressing this complex challenge requires a multifaceted approach, with machine learning presenting a promising avenue for automated detection and analysis [2].

This paper delves into the intersection of technology and societal well-being, aiming to harness the power of machine learning algorithms to classify the negative impact of hate speech. By leveraging machine learning techniques, we endeavor to shed light on the impact of hate speech on the victims and to equally decode the predominant hate rhetoric and sentiment in a given location. The objective of this study is to develop robust machine learning models capable of classifying hate speech [3]. Through these endeavors, we hope to contribute to the development of informed, data-driven strategies for mitigating the adverse effects of hate speech.

In this pursuit, we navigate a landscape where linguistic nuance, context, and evolving cultural norms play pivotal roles. This necessitates a nuanced approach to algorithmic design, one that balances accuracy with adaptability in the face of ever-changing forms of expression. By harnessing the power of natural language processing, we endeavor to equip our models with the capacity to decipher the intricacies of language, discerning between legitimate discourse and expressions laden with hate [4].

In the following sections, we will delve into the foundational concepts, methodologies, and empirical findings that underpin our approach. Through a synthesis of cutting-edge machine learning techniques and a deep dive into the intricate landscape of hate speech, we endeavor to illuminate the path forward in our collective pursuit of a safer, more inclusive digital sphere.

2. LITERATURE REVIEW

Study by [5], shows compelling evidence of online hate speech which has assume different dimensions. People and organization perpetrate hate speech which target individuals, groups, community or race for several reasons such as; intimidating, shaming, discrediting, and incite violence etc. [6] supported the above assertions and that online hate speech have become pervasive due to the presence of internet which has brought about the opportunities for disseminating hate speech. And defined Hate speech as bias-motivated, hostile, malicious speech aimed at a person or a group of people because of some of their actual or perceived innate characteristics and that the main motivation is to expresses discriminatory, intimidating, disapproving, antagonistic, prejudicial attitudes toward the targets. The study advocates the need for collective responsibility to tackle online hate speech.

With the rise of hate speech phenomena in the Twittersphere and social media in particular, significant research efforts have been undertaken in order to provide automatic solutions for detecting hate speech, varying from simple machine learning models to more complex deep neural network models [7].

The study by [8] developed a model using linear support vector machine and Naïve bayes to identify offensive language in social media tweet, the study obtains the following results for accuracy and recall for Naïve Bayes (92%, 95%), and Linear SVM (90%, 92%).

Another study by [9] developed machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, Multinomial Naïve Bayes, Stochastic Gradient Descent to classify suspicious text in Bengali text documents, feature extraction and selection techniques such as TF-IDF and BOW were used in the work. The five classifiers were trained using the feature extracted datasets for TF-IDF and BOW and gave

different results. For the BOW feature extraction techniques Random Forest gave the highest accuracy of 83.21%, while Stochastic Gradient Descent gave the highest precision with 83.79%. Whereas for the TF-IDF feature extracted datasets, the Stochastic Gradient Descent gave accuracy of 84.57% and a precision of 83.78%. Comparison of the two feature extracted techniques shows TF-IDF outperforms the BOW techniques in terms of the overall results obtained in the study.

In a similar study by [10] to detect offensive content in code-mixed dataset of Dravidian languages, machine learning models were equally trained using different features such as n-gram, character n-gram, combined word and custom word embedding. Using the TF-IDF weights of word and character n-gram features they trained Machine learning classifiers. The model for Malayalam language got the official rank of 2nd and obtained an F1-score of 0.77, while Tamil language Model got the official rank of 3rd and F1 score of 0.87 from the results.

A study by [11] developed eight Machine Learning models with three selected feature engineering techniques to detect hate speech in text. The experiment shows that bigram features and TF-IDF gave a better result compared to word2Vec and Doc2Vec feature engineering techniques. Amongst the models trained SVM (79%) gave better results compared to the rest of the models while KNN gave the least result.

Another study by [7] developed deep learning model using CNN, GRU, CNN + GRU, and BERT to classify hate speech into hate and non-hate. They used annotated data to train the models and out-domain (secondary) data for testing to see how well the model generalize to unseen datasets. From the results of the experiment, both the annotated and an out-domain dataset showed that the CNN model gave the best performance, with an F1-score of 0.79 and area under the receiver operating characteristic curve (AUROC) of 0.89.

From the literature review by [12] various approaches are already in used to detect hate speech ranging from rule based, Machine Learning and deep learning and Hybrid techniques which lend credence to the fact that several studies have been adduce to detect hate speech from existing literature.

Most of the literature review shows extensive progress in detecting the presence of hate speech and also classifying hate speech from non-hate using machine learning and deep learning techniques. However, gap still exist in the literature as to the negative impact of hate speech on the victims and predominant hate rhetoric in a given geolocation. This work intends to close the above gap by classifying hate speech based on its impact on the victims and also identify the predominant hate-based rhetoric in a given geolocation.

3. METHODOLOGY

This research employed quantitative, qualitative, and experimental research methods. The study involved conducting experiments to train and test supervised machine learning models, utilizing Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF). The model with the highest accuracy was chosen for deployment, and algorithm performance was assessed using Confusion Matrix and classification reports.

3.1 Multi-class Logistic Regression

Multi-class logistic regression, also known as multinomial logistic regression or SoftMax regression, is a classification algorithm used to model the relationship between a set of input features and multiple classes. It extends binary logistic regression to handle problems where there are more than two classes. The goal is to estimate the probability of each category for a given set of predictor values [13].

The mathematical theory behind multi-class logistic regression involves several key concepts:

3.1.2 Linear Combination of Features

In multi-class logistic regression, each class has its set of weights associated with the input features. For a given class j (where j ranges from 1 to C, where C is the number of classes) the linear combination of features X can be represented as:

$$\mathbf{Z_j} = \boldsymbol{\theta_{0j}} + \boldsymbol{\theta_{1j}\chi_1} + \boldsymbol{\theta_{2j}\chi_2} + \ldots + \boldsymbol{\theta_{nj}\chi_n} \quad (1)$$

Here, Z_j is the linear combination for class j, xi represents the i-th feature, and θ_(0j) is the weight associated with i-th feature for class j.

3.1.3 SoftMax Function

In multi-class logistic regression, the SoftMax function is used to transform the linear combinations Z_j into class probabilities. The SoftMax function for class j is defined as:

$$\mathrm{P}(\mathrm{Y} = j|\mathrm{X}) = \frac{e^{zj}}{\sum_{k=1}^{C} e^{zk}} \quad (2)$$

Here, P (Y= j | X) is the probability of the input belonging to the class j, and C is the total number of classes. The exponential function e^{zj} ensures that the probabilities are non-negative, and the denominator sums over all classes to normalize the probabilities to add up to 1.

3.1.4 Multi-class Loss Function

The loss function used in multi-class logistic regression is typically the cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the true class labels. The loss for a single training example is defined as:

$$L(\mathrm{Y}, \mathrm{P}) = -\sum_{j=1}^{C} y_j \; log(P(y = j \mid X)) \quad (3)$$

Here, L (Y, P) is the loss for the training example, yj is an indicator variable (1 if the true class is j, 0 otherwise), and

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 01 - 07, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1001

P (γ = j | X) is the predicted probability for class j.

3.1.5 Optimization

The goal of training in multi-class logistic regression is to minimize the overall loss across all training examples. This is typically done using optimization techniques like gradient descent.

The gradients of the loss with respect to the model parameters θ (0j) are computed, and the parameters are updated iteratively to minimize the loss.

3.1.6 Algorithm

1. Input
   Training dataset D with features X and corresponding labels Y
   Learning rate (α), regularization parameter (λ), number of iterations
2. Data Preprocessing
   Encode categorical variables (if any) and normalize/standardize the feature values.
3. One-hot Encoding of Labels
   Convert the multi-class labels (Y) into a one-hot encoded matrix $Y_{one-hot}$ with C columns, where C is the number of classes
4. Initialize Parameters
   Initialize weight matrix W and bias vector b with random or zero values.
5. Training
   Function Train_Multinomial_Logistic_Regression (X, Y, one_hot, α, λ, num_iterations):
   For i in range num_iterations:
   Calculate the linear scores $Z = XW + b$
   Calculate the SoftMax probabilities P for each class:
   $P = softmax(Z)$
   Calculate the loss using the cross-entropy loss function:

$$J = -\frac{1}{M}\sum_{j=1}^{C}\gamma_{ij}\, log(P_{ij}) + \frac{\lambda}{2m}\sum_{k=1}^{n}\sum_{j=1}^{C}W^2\ k_j$$

   Calculate the gradient of the loss w.r.t W and b

$$dW = \frac{1}{m}X^T.(P - Y_{one-hot}) + \frac{\lambda}{m}W$$

$$db = \frac{1}{m}\sum_{i=1}^{m}(P - Y_{one-hot})$$

   Return W and b
6. Prediction
   Function Predict_Multinomial_Logistic_Regression (X, W, b):
   Calculate the linear scores $Z = XW + b$
   Calculate the SoftMax probabilities P for each class:
   $P = soft\max(Z)$
   Return the class with the highest probability for each sample.

3.2 Decision Tree

Decision tree algorithms comprise trees that categorize data by looking at feature values. Every node in a decision tree depicts a feature that has to be classified, and the branches depict values that are considered by such a node, [14]. A decision tree is a popular machine learning algorithm used for both classification and regression tasks. It works by recursively partitioning the dataset into subsets based on the values of input features, ultimately leading to a tree-like structure where each leaf node represents a class label (in classification) or a numerical value (in regression).

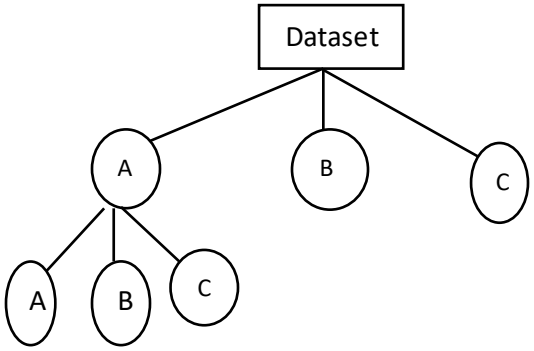

Figure 1: Multiclass Decision Tree

3.2.2 Entropy

Entropy is a measure of disorder or impurity in a set of data. In the context of decision trees, it's used to quantify the impurity of a dataset before a split. Mathematically, entropy is defined as:

$$H(S) = -\sum_{i=1}^{c} pi\ log_2(pi) \quad (4)$$

Where c is the number of classes and pi is the probability of an instance belonging to class i.

3.2.3 Information Gain

Information Gain is the reduction in entropy or impurity achieved by partitioning a dataset based on a specific attribute (feature). It helps in deciding which attribute to split on. Mathematically, Information Gain is calculated as:

$$IG(S,A) = H(S) - \sum_{v\epsilon Values(A)}\frac{|S_v|}{|S|}.H(S_v) \quad (5)$$

Where S is the dataset, A is the attribute being considered for the split, Values (A) are the possible values of attributes A, S_v is the subset of S where attributes A takes the value v.

3.2.4 Gini Impurity

Gini Impurity is an alternative to entropy for measuring impurity. It quantifies the probability of a randomly chosen element being misclassified. For a dataset S with c classes, Gini Impurity is calculated as:

$$Gini\ (S) = 1 - \sum_{i=1}^{c}(P_i)^2 \quad (6)$$

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 01 - 07, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1001

3.3 Random Forest Classifier

A Random Forest classifier is an ensemble learning method that combines the predictions of multiple decision trees to improve the accuracy and robustness of a classification task. It is based on the idea that a collection of weak learners (individual decision trees) can come together to form a strong learner. Random Forest is a supervised machine learning algorithm that uses a group of decision tree models for classification and making predictions, [15]. As the name indicates, a forest will be created from a group of decision trees and then will be randomized.

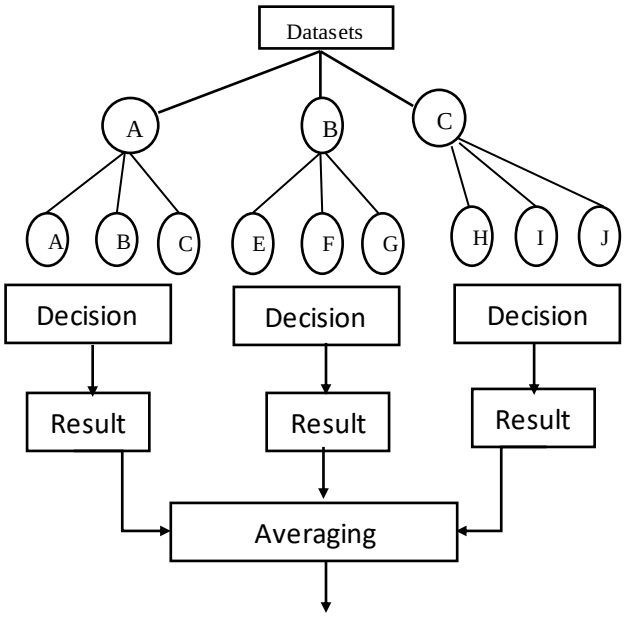

Figure 2: Random Forest

3.4 Data Pre-processing

To construct a comprehensive dataset for training and evaluation. We sourced data from social media platforms such as Facebook [16]. Prior to training, we subjected the dataset to extensive preprocessing [17]. This involved tasks such as lowercasing, punctuation removal, and tokenization.

To facilitate supervised learning, we engaged a team of human annotators proficient in understanding and categorizing hate speech [8]. Annotations were carried out using a multi-class labeling system, distinguishing between non-offensive language, offensive language, and hate speech to ensure inter-annotator agreement [18]. The dataset was split into training and test sets with the training set accounting for 70% while the test set 30% of the data.

The input variables are the hate motivated public comments express on social media platform. Because of freedom of expression, social media users are allowed to post and express their opinions freely. The data labeling was done by the annotators.

Table 1. Annotated data with labels

Labels	Comments
Violence	their people have ruined the country
discrimination	make sure you don't admit them in our institution
positive	are you coming for the wedding?
harrassment	you don't belong here

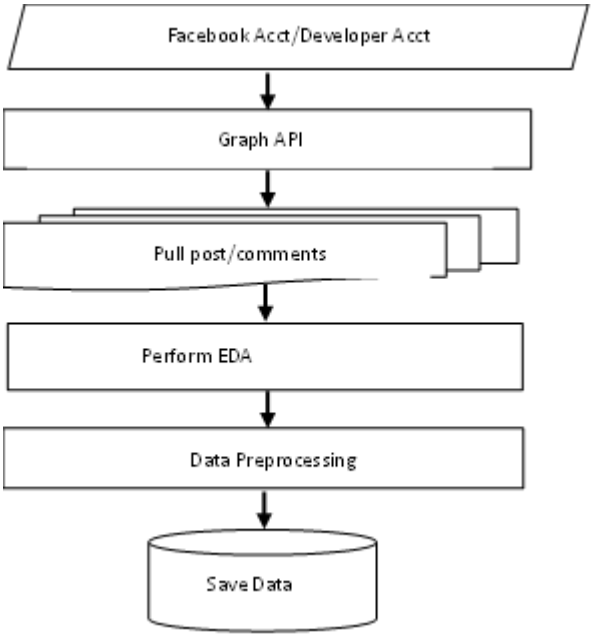

Figure 3: Data Collection Techniques

The target variable or the labels are the negative impact of a given comments or post from the public.

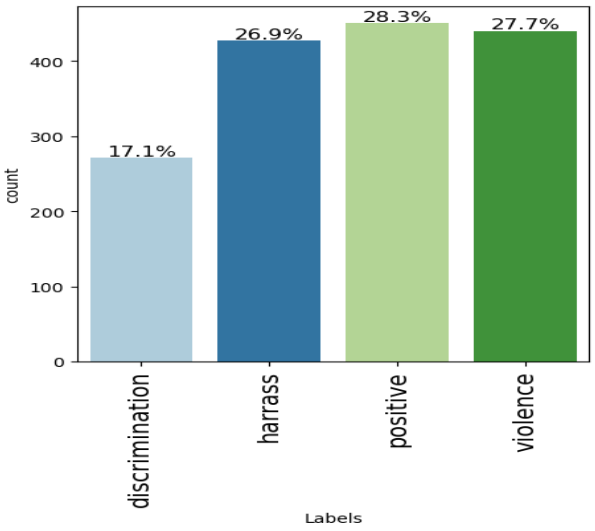

Figure 4: Distributions of labels in the datasets

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 01 - 07, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1001

We experimented with a range of machine learning models such as Logistic Regression, Decision Tree and Random Forest. To assess the efficacy of our models, we conducted rigorous evaluation using standard metrics such as precision, recall, F1-score, and accuracy.

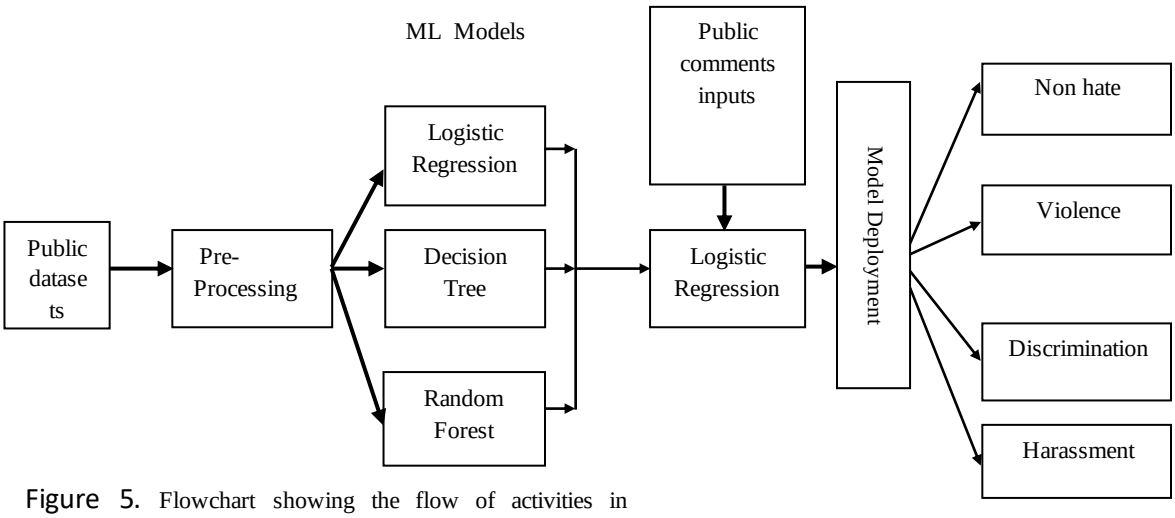

Figure 5. Flowchart showing the flow of activities in implementing the machine learning model for hate speech classification

To determine the impact of hate speech on public sentiment, we conducted sentiment analysis on the annotated dataset. We employed established sentiment lexicons and machine learning-based approaches to classify the emotional trajectory of the text.

3.5 Model Training and Evaluations

The three models were trained using the above datasets. The data was split using the 7:3 ratio i.e., 70% of the data were used to train the model while the remaining 30% for testing. The accuracy of the models was evaluated and the model with the highest accuracy was used for deployment. The codes were written using Python programming Language and the Visual Studio Code Integrated Development Environment (IDE) bundled with Anaconda.

A confusion matrix is a table used in machine learning to evaluate the performance of a classification algorithm. It summarizes the predictions of a model on a set of data for each class and compares them with actual labels. The matrix typically includes four values: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values are used to calculate various performance metrics such as accuracy, precision, recall, and F1 score, which help assess the model's classification performance.

Accuracy: In machine learning, accuracy is a metric that measures the correctness of the predictions made by a model. It is defined as the ratio of correctly predicted instances to the total instances in the dataset.

Accuracy = (TP +TN) / (TP + TN + FP + FN)

Precision: Precision is a metric in machine learning that assesses the accuracy of the positive predictions made by a model. It is defined as the ratio of true positive predictions to the total number of positive predictions

Precision = TP / (TP + FP)

Recall: Recall, also known as sensitivity or true positive rate, is a metric in machine learning that measures the ability of a model to capture and correctly identify all the relevant instances of a particular class, out of all the instances that truly belong to that class in the dataset.

Recall = TP / (TP + FN)

F1 score: The F1 score is a metric in machine learning that combines both precision and recall into a single value. It is particularly useful when there is an imbalance between the classes or when both false positives and false negatives need to be considered.

F1-score = 2 * (Precision * Recall) / (Precision + Recall)

The model with the highest accuracy will be save for deployment. Joblib library was used to save the model. Joblib is a library that help to save and export machine learning models to production environment.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 01 - 07, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1001

4. RESULTS

The following table and figures below show the performance evaluation of the models.

Table 2: Model performance and results

Trained model	Test Accuracy	Precision	Recall	f1-score
Logistic Regression	0.9657	0.97	0.96	0.96
Decision Tree	0.9981	1	1	1
Random Forest	1	1	1	1

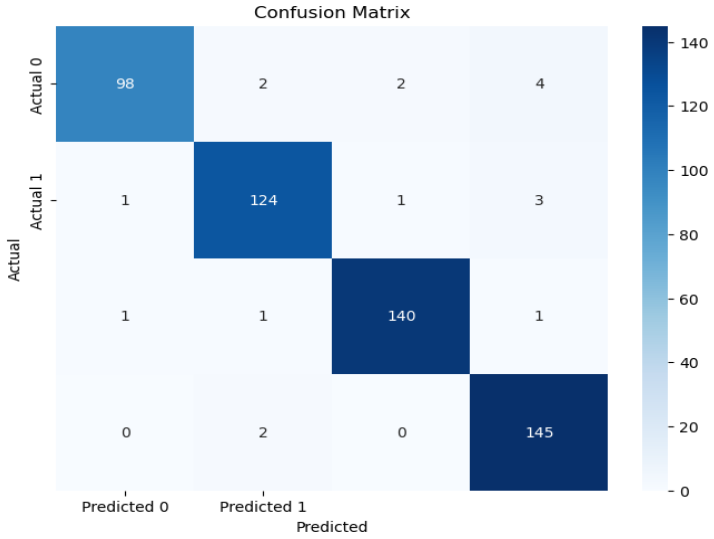

Figure 6: Logistic regression confusion matrix

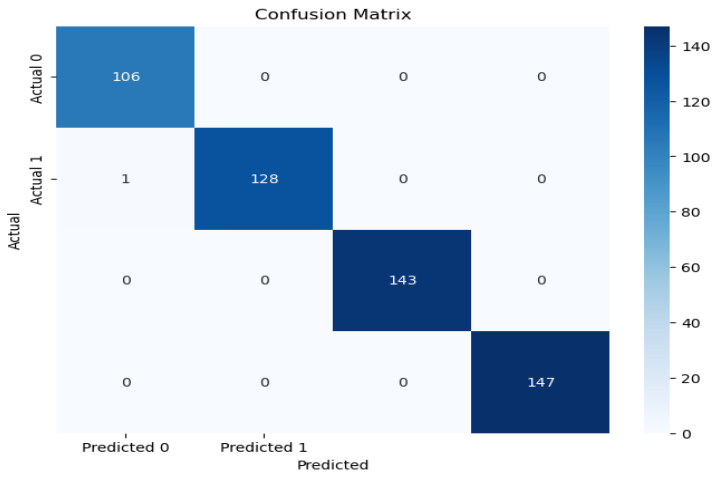

Figure 7: Decision Tree Confusion Matrix

5. DEPLOYMENT

After deploying the model register users can access the user interface in the form of html text field and type their comments and then submit using the submit button as is no Deployment ensures our model is made available for public use; the selected model (Logistic regression) was deployed locally using the open source Streamlit Python library. Streamlit is an open source, light weight Python library that is used to deploy machine learning models to production [19]. It lets you create and share amazing data apps with fewer efforts and lines of code.

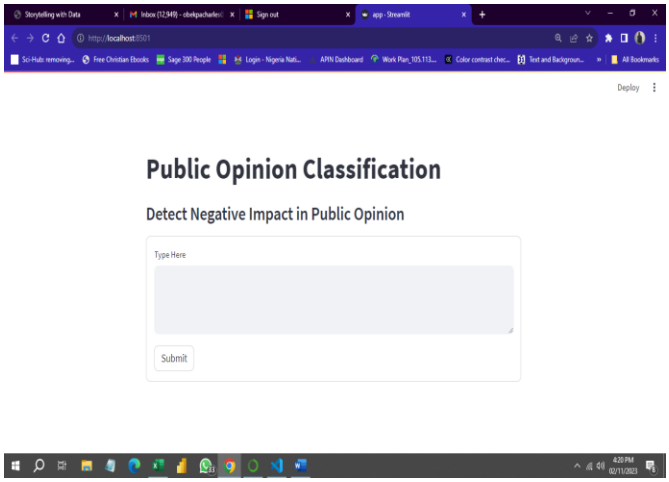

Figure 8: User interface to the deployed model

Figure 9: Final prediction from the model

The above experiments demonstrated promising results in classifying hate speech. Decision Tree and Random Forest classifiers tends to overfit the data while the Logistic regression classifier provided a nuanced and improved results during testing informing the decision to use the classifier for deployment.

6. CONCLUSIONS

In the face of an increasingly interconnected digital landscape, the identification and mitigation of hate speech emerges as a critical imperative for the preservation of inclusive and constructive public discourse. This study has embarked on a journey at the intersection of technology and societal well-being, leveraging machine learning approach to dissect and classify the detrimental impact of hate speech on public sentiment.

Through an extensive empirical investigation, we have demonstrated the efficacy of our machine learning models in

discerning and classifying the impact of hate speech and also isolating the predominant hate rhetoric in an area.

Furthermore, our analysis reveals the profound impact of hate speech on public sentiment. This not only highlights the emotional toll borne by affected individuals and communities but also underscores the potential for polarization and division within society.

However, our work is not without its limitations. Challenges persist in discerning subtle linguistic constructs, and the ever-evolving nature of hate speech demands continued vigilance and refinement in model training. Furthermore, the dynamic nature of online discourse necessitates ongoing efforts to adapt and refine our approach.

In summary, this study advances a multifaceted approach to addressing the impact of hate speech on public opinion and identify the hate sentiment in a given location. By harnessing the power of machine learning, we have endeavored to illuminate the intricate dynamics of hate speech, contributing to a safer, more inclusive digital sphere. As we look to the future, our hope is that this research serves as a catalyst for continued innovation and collaboration in the pursuit of a more empathetic and understanding online community.

7. ACKNOWLEDGMENTS

I express my profound and sincere gratitude to the post graduate coordinator Dr. Adeyelu Adekunle, Ekle Francis Adoba and my supervisors for tremendous support in this research and to my family for the financial and moral support accorded me.

8. REFERENCES

[1] Lutfiye S, M, A. 2019. Identification of Offensive Language in social media.

[2] Zafer A, Amr, T. (2019).p *Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach.* Springer-Verlag GmbH Austria.

[3] Dewa, A, N, T, Ketut, G, D, P. 2021. Hate Speech Classification in Indonesian Language Tweets by Using Convolutional Neural Network.

[4] Sattam. A. (2018). *A lexicon based method to search for extreme opinions*. PLOS | ONE.

[5] Steiger, K. R. (2020). Online hate: Introduction into motivational causes, effects and regulatory contexts.

[6] Raphael, C. (2011). Fighting Hate and Bigotry on the Internet. Policy & Internet.

[7] Raghad, A, Hend, A. (2020). A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. Applied Sciences.

[8] Gabriel, A, De, S, Marjory, Da, C. (2019). Automatic offensive language detection from Twitter data using machine learning and feature selection of metadata. http://alt.qcri.org/semeval2019.

[9] Omar, S, Mohammed M, H, Kayes, R, N, Iqbal, H, S. (2020). Detecting Suspicious Texts Using Machine Learning Techniques. Applied Sciences.

[10] Varsha, P, Manish, J, Prasad, A, Joshi, M, M, Tanmay J. (2020). Using Machine Learning for Detection of Hate Speech and Offensive Code-Mixed Social Media text.

[11] Sindhu, A, Sarang, S, H, K, Zafar, A, Sajid, K, Ghulam, M. (2020). Automatic Hate Speech Detection using Machine Learning: A Comparative Study. International Journal of Advanced Computer Science and Applications.

[12] Mohiyaddeen, S, S. (2021). Automatic Hate Speech Detection: A Literature Review. International Journal of Engineering and Management Research.

[13] Williams, R. (2021). Multinomial Logit Models - Overview.

[14] Lior Rokach, Oded, M. (2015). Data Mining with Decision Trees Theory and Applications.

[15] Jonathan, K, A, Kassim T, Wilhemina, A, P, Sandra, A, Harriet, A, D, Emmanuel, O, O, Samuel, A, A, John, E. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. doi: https://doi.org/10.1016/j.dajour.2023.100163

[16] Taherdoost, H. (2021). Data Collection Methods and Tools for Research; A Step-by-Step Guide to Choose Data Collection Technique for Academic and Business Research Projects. International Journal of Academic Research in Management.

[17] Anubha Parashar, Apoorva, P, Weiping, D, Mohammad, S, Imad, R. (2023). Data Preprocessing and Feature Selection Techniques in Gait Recognition: A Comparative Study of Machine Learning and Deep Learning Approaches. Elsevier.

[18] Teodor, F, David, I, M, J, Helena, H. (2021). Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. DOI: 10.1007/978-3-030-64148-1_13.

[19] Arul S, Muskaan, D, Kowsigan, M. (2022). Indian Crop Production: Prediction And Model Deployment Using Ml And Streamlit .

[20] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender