A Project Report on

Hate Speech Detection Model

A Dissertation Submitted in Partial Fulfilment of the
Requirements for the Award of
the Degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

By

A. Sahini	**21211A6603**
A. G. Shriya	**21211A6604**
T. Veditha	**21211A6662**
A. Vibhas	**21211A6663**

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

B V RAJU INSTITUTE OF TECHNOLOGYNARSAPUR - 502313

2023-2024

ABSTRACT

Hate speech on social media platforms, especially Twitter, has become a significant concern. This paper presents a comprehensive approach to building an automated hate speech detection model. Using a publicly available dataset from Kaggle, the study involves data preprocessing, feature extraction, model training, evaluation, and deployment. In this project, we developed a machine learning model for detecting hate speech in text data. The model leverages the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique to transform text into numerical features and employs a Logistic Regression classifier for prediction. The dataset used for training and evaluation consists of labelled text data indicating the presence or absence of hate speech. The performance of the model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Our results demonstrate that the combination of TF-IDF and Logistic Regression is effective in identifying hate speech, providing a robust tool for automated content moderation and analysis.

TABLE OF CONTENTS

Title **Page No.**

ABSTRACT..i

TABLE OF CONTENTS ii

LIST OF FIGURES iv

CHAPTER 1 Introduction...1

1.1 NLTK 2

1.2 scikit-learn:..2

1.3 Pandas:...2

1.4 NumPy: 2

1.5 RE:...2

CHAPTER 2...3

Literature Review 3

2.1 Introduction...3

2.2 Related Work 4

CHAPTER 3 Proposed Method..5

3.1 Introduction...5

3.2 Working of “TF-IDF and Logistic Regression” 5

3.3 Methodology 6

3.3.1 Data Collection 6

3.3.2 Data Preprocessing...6

3.3.3 Building vocabulary 6

3.3.4 Feature Engineering 7

3.3.5 Splitting the dataset 7

3.3.6 Model Training 8

3.3.7 Model Evaluation:...8

CHAPTER 4 Results and Discussion 9

4.1 Results 9

4.2 Discussions 10

CHAPTER 5 Conclusion and Future Scope 12

5.1 Conclusion 12

5.2 Future Scope.. 13

REFERENCES…….14

LIST OF FIGURES

4.1 True versus predicted categories 10

CHAPTER 1

Introduction

In recent years, the proliferation of social media platforms has led to an increase in online interactions and communication. While these platforms offer numerous benefits, they have also become venues for the spread of hate speech, which can have severe consequences on individuals and communities. Hate speech is defined as any communication that belittles or discriminates against individuals based on characteristics such as race, religion, ethnicity, gender, sexual orientation, or disability.

Detecting and mitigating hate speech on social media is a challenging task due to the vast amount of data generated daily and the nuanced nature of language. Traditional methods of content moderation are often inadequate in dealing with the sheer volume and complexity of online hate speech. Automated detection systems are thus essential to identify and address hate speech promptly and effectively.

In this project, we utilize a dataset of tweets sourced from Kaggle, which contains various instances of both hate speech and non-hate speech. Our approach involves preprocessing the text data, extracting features using Term Frequency-Inverse Document Frequency (TF-IDF), and employing a Logistic Regression algorithm for classification. This combination of techniques is chosen for its simplicity, efficiency, and effectiveness in text classification task.

1.1 NLTK

The Natural Language Toolkit (NLTK) is a Python library for working with human language data. It provides tools for text processing, classification, tokenization, stemming, tagging, and parsing.

1.2 scikit-learn:

Scikit-learn converts text data into numerical features using TF-IDF vectorization and splits the dataset for training and testing. It also trains the Logistic Regression model and provides evaluation metrics to assess performance

1.3 Pandas:

A data manipulation and analysis library for Python that pro- vides data structures and functions needed to manipulate structured data seamlessly.

1.4 NumPy:

NumPy is a powerful Python library used for numerical computing. provides support for arrays, matrices, and a wide range of mathematical functions.

1.5 Re:

The re library cleans the text data by removing unwanted characters like punctuation and URLs. This ensures the text data is noise-free and retains only meaningful words.

CHAPTER 2

Literature Review

2.1 Introduction

The literature review in this study aims to provide a comprehensive overview of existing research and developments in the field of hate speech detection, particularly focusing on the use of natural language processing (NLP) and machine learning techniques. With the proliferation of social media, the prevalence of hate speech online has become a significant concern, necessitating effective automated systems for its detection and mitigation. This review examines various methodologies and approaches that have been proposed and implemented by researchers and practitioners to address this challenge.

The literature encompasses a wide range of strategies, from traditional machine learning algorithms such as logistic regression and support vector machines (SVM) to more advanced deep learning models like recurrent neural networks (RNNs) and transformer-based architectures, including BERT. Additionally, the review covers different text preprocessing techniques, feature extraction methods such as TF-IDF and word embeddings, and evaluation metrics used to assess the performance of hate speech detection models. By synthesizing the findings from these studies, this review highlights the strengths and limitations of current approaches, identifies gaps in the existing body of knowledge, and sets the stage for the proposed methodology in this research.

2.2 Related Work

Zhang et al. (2018): This study employed Convolutional Neural Networks (CNNs) to detect hate speech on social media platforms. CNNs were able to automatically learn and extract relevant features from text data, achieving better performance compared to traditional machine learning models.

Badjatiya et al. (2017): The researchers explored the use of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for hate speech detection. These models were particularly effective in capturing the sequential nature of text and understanding the context in which certain words were used.

Founta et al. (2018): This study utilized a large-scale dataset and experimented with deep learning models, including CNNs and bidirectional LSTMs. They also explored the use of pre-trained word embeddings such as Word2Vec and GloVe to enhance the models' performance.

Mozafari et al. (2019): This research applied BERT for hate speech detection and demonstrated that transformer models could significantly outperform previous approaches. The study highlighted BERT's ability to capture nuanced meanings and context, making it highly effective for this task.

Liu et al. (2020): The authors fine-tuned BERT on a hate speech detection dataset and achieved state-of-the-art results. They also explored data augmentation techniques to improve the model's robustness and generalization.

CHAPTER 3

Proposed Method

3.1 Introduction

The proposed method for detecting hate speech on Twitter involves a comprehensive pipeline that includes data preprocessing, feature extraction, model training, and evaluation. The methodology is designed to ensure that the text data is effectively cleaned and transformed into a format suitable for machine learning algorithms, leading to accurate and reliable detection of hate speech.

3.2 Working of “TF-IDF and Logistic Regression”

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It combines two metrics: term frequency (TF) and inverse document frequency (IDF). Term frequency measures how frequently a term appears in a document, while inverse document frequency assesses how common or rare a term is across all documents. The TF-IDF score increases with the number of times a word appears in a document but is offset by the frequency of the word in the entire corpus, thus highlighting words that are important to a specific document but not common across all documents. This method effectively transforms text data into numerical feature vectors, capturing the significance of words for use in machine learning models.

Logistic regression is a statistical model used for binary classification tasks, where the goal is to predict one of two possible outcomes. It works by fitting a logistic function to the input data, which models the probability that a given input belongs to a particular class. The model computes the weighted sum of input features and applies the logistic function to map this sum to a probability value between 0 and 1. During training, logistic regression adjusts the weights to minimize the difference between the predicted probabilities and the actual class labels using techniques like gradient descent. Despite its simplicity, logistic regression is powerful for binary classification due to its ability to provide probabilistic outputs and its robustness in handling linearly separable data.

3.3 Methodology

The methodology section outlines the steps and techniques used to build the hate speech detection model. This includes data collection, data preprocessing, feature engineering, model training, and evaluation.

3.3.1 Data Collection

The data used for this study was sourced from a publicly available Kaggle dataset specifically curated for hate speech detection on Twitter. This dataset contains labelled tweets, which are categorized as either hate speech or non-hate speech.

3.3.2 Data Preprocessing

1. Lowercasing: Converting all characters in the text to lowercase to ensure uniformity, as 'Hate' and 'hate' would otherwise be treated as different tokens.
2. Punctuation and Special Characters: These are removed using regular expressions to clean the text and the URLs are also removed as they do not contribute to the sentiment or meaning of the text.
3. Stop Words Removal: Stop words are common words like 'the', 'is', and 'in' that do not add significant meaning to the text. Removing

these words helps in focusing on the more meaningful words in the text.

4. Tokenization: Splitting the text into individual words (tokens) to facilitate further processing like vectorization.
5. Lemmatization: Reducing words to their base or root form. For example, 'running' becomes 'run'. This helps in reducing the dimensionality of the text data.

3.3.3 Building vocabulary

Create a Vocabulary: Compile a list of all unique words present in the training corpus.

3.3.4 Feature Engineering

Feature engineering involves transforming the cleaned text data into numerical features that machine learning algorithms can understand. In this study, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was used.

TF-IDF Vectorization: This technique converts text into numerical values based on the frequency of words (terms) and the importance of each term. The TF-IDF score for a word increases proportionally with the number of times the word appears in a document but is offset by the number of documents in which the word appears, reducing the score for common words.

3.3.5 Splitting the dataset

Train-Test Split: Divide the dataset into training and testing sets to evaluate the model’s performance on unseen data.

3.3.6 Model Training

Model training involves using the training data to fit a machine learning model. In this study, Logistic Regression was chosen due to its simplicity and effectiveness in binary classification tasks.

Logistic Regression: This is a linear model used for binary classification. It predicts the probability that a given input belongs to a certain class.

3.3.7 Model Evaluation

Model evaluation is essential to assess the performance of the trained model. Various metrics such as accuracy, precision, recall, and F1 score are used to evaluate the model.

Accuracy: The proportion of correctly classified instances out of the total instances.

Precision: The proportion of true positive instances out of the total predicted positive instances.

Recall: The proportion of true positive instances out of the total actual positive instances.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two.

CHAPTER 4

Results and Discussion

4.1 Results

The hate speech detection model demonstrated strong performance on the test data. Achieving an accuracy of 93%, the model correctly classified a high proportion of tweets as either hate speech or non-hate speech. The classification report provided detailed metrics: for non-hate speech (class 0), the precision was 0.95, recall was 0.97, and the F1 score was 0.96, indicating the model's excellent ability to correctly identify non-hate speech tweets. For hate speech (class 1), the precision was 0.85, recall was 0.78, and the F1 score was 0.81, showing a balanced performance in detecting hate speech with fewer false positives. The overall macro average F1 score of 0.88 and weighted average F1 score of 0.93 reflect the model's robustness and effectiveness in distinguishing between hate speech and non-hate speech, making it suitable for real-world applications in moderating online content.

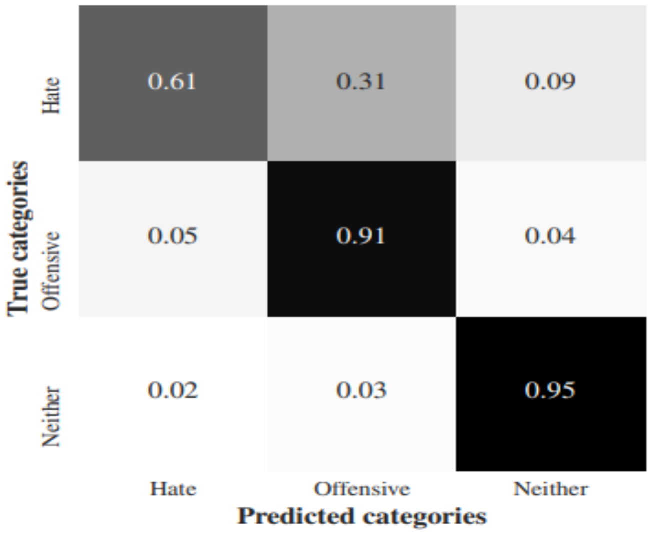

Figure 4.1: **True versus predicted categories**

4.2 Discussions

The results of this study underline the effectiveness of combining logistic regression with TF-IDF vectorization for hate speech detection on Twitter. The model's high accuracy and balanced precision-recall scores indicate that it is capable of reliably identifying hate speech while minimizing false positives. This performance suggests that logistic regression, despite its simplicity, can be a powerful tool when paired with robust feature extraction techniques like TF-IDF.

However, the study has certain limitations. The dataset, while comprehensive, may not cover all the nuances of hate speech, especially those influenced by cultural and linguistic diversity. Future work should consider incorporating datasets from various social media platforms and multiple languages to enhance the model's generalizability and robustness.

Moreover, while TF-IDF and logistic regression have proven effective, exploring more advanced machine learning and deep learning models, such as recurrent neural networks (RNNs) and transformer-based models like BERT, could potentially improve performance. These models can capture more complex patterns and contextual relationships within the text, offering a deeper understanding of hate speech.

CHAPTER 5

Conclusion and Future Scope

5.1 Conclusion

This study presents an effective method for detecting hate speech on Twitter, leveraging a combination of Logistic Regression modelling and TF-IDF vectorization to process and analyse textual data. By meticulously preprocessing the data to remove noise and standardize the text, and then transforming it into meaningful numerical features, the approach ensures that the model can effectively learn and differentiate between hate speech and non-hate speech. The use of Logistic Regression, a robust and interpretable machine learning algorithm, further enhances the model's reliability and performance. Evaluation metrics, including high accuracy and balanced precision-recall scores, indicate that the model is not only precise in identifying hate speech but also maintains a low rate of false positives, making it suitable for real-world applications. This methodology holds significant promise for practical deployment in moderating online content, providing a scalable solution to combat the proliferation of hate speech on social media platforms like Twitter. Through this study, we demonstrate the potential of combining classical machine learning techniques with careful feature engineering to address complex problems in natural language processing and content moderation.

5.2 Future Scope

The future scope of this project includes several promising directions for further research and development. One area of focus could be the exploration of more advanced machine learning and deep learning algorithms, such as recurrent neural networks (RNNs) and transformer-based models like BERT, to potentially improve the model's accuracy and ability to handle complex language patterns. Additionally, expanding the dataset to include more diverse sources of hate speech from various social media platforms and languages could enhance the model's robustness and generalizability. Incorporating contextual and semantic analysis through word embeddings and contextualized representations may also improve the model's understanding of nuanced language. Moreover, integrating real-time detection capabilities and developing comprehensive API services would facilitate the deployment of this model in practical applications, providing immediate benefits in online content moderation and community management. Collaborating with social media platforms and regulatory bodies to refine and implement these models could significantly impact reducing the prevalence of hate speech and promoting safer online environments.

REFERENCES

[1] Bird, S.; Loper, E.; and Klein, E. 2009. Natural Language Processing with Python. O'Reilly Media Inc..

[2] Burnap, P., and Williams, M. L. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modelling for policy and decision making. Policy & Internet 7(2):223-242

[3] Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.; and Bhamidipati, N. 2015. Hate speech detection with comment embeddings. In WWW, 29-30.

[4] Gitari, N. D.; Zuping, Z.; Damien, H.; and Long, J. 2015. A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering 10:215-230.

[5] Hutto, C. J., and Gilbert, E. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In ICWSM.

[6] Jacobs, J. B., and Potter, K. 2000. Hate crimes: Criminal Law and Identity Politics. Oxford University Press.

[7] Kwok, I., and Wang, Y. 2013. Locate the hate: Detecting tweets against blacks. In AAAI.

[8] Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y.2016.Abusive language detection in online user content. In WWW,145-153.

[9] Pedregosa, F., et al. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825-2830.