Mini Project Report on

HATE SPEECH DETECTION

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by:

Student Name:- SURYANSH BAJPAI **University Roll No.:-** 2019166

Under the Mentorship of

Ms. Garima Sharma
Designation:-Assistant Professor

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
January 2023

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Hate Speech Detection"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms. Garima Sharma, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name:-**Suryansh Bajpai** **University Roll no.**:-**2019166**

Table of Contents

Chapter 1: -

Introduction

Any speech that disparages a group of people because of their race , religion, nationality, sexual orientation, or gender identity is considered hate speech.

In common language, “hate speech” refers to offensive discourse targeting a group or an individual based on inherent characteristics such as race, religion or gender and that may threaten social peace.

Hate speech is frequently used to promote bigotry and hatred. Additionally, it can be applied to threaten and intimidate others. People may experience loneliness, anxiety, and fear as a result.

Online hate speech may be distributed quickly, cheaply, and anonymously in contrast to traditional media. It might be able to instantly connect with a large and varied audience. Additionally troubling is the relative persistence of nasty internet content, which can reappear and acquire popularity over time.

To develop new responses, it is essential to track and comprehend hate speech across many online networks and platforms.

Introduction 1.1 : -

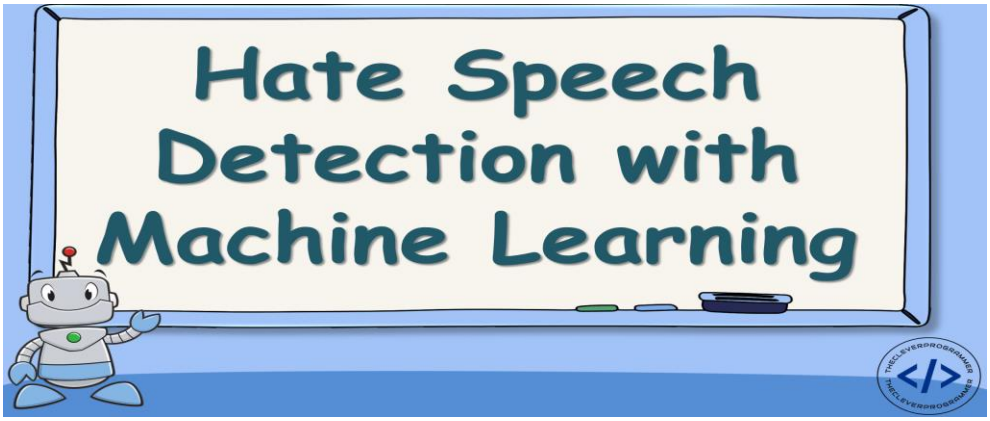
What is Machine Learning?

Machine learning (ML) is a field of study focused on comprehending and developing "learning" methods, or methods that use data to enhance performance on a particular set of tasks. It is considered to be a component of artificial intelligence.

Without being explicitly programmed to do so, machine learning algorithms create a model from sample data, also referred to as training data, in order to make predictions or decisions.

Machine learning algorithms are used in a wide range of applications, including computer vision, speech recognition, email filtering, medicine, and agriculture, where it is challenging or impractical to create conventional algorithms that can perform the required tasks.

How can machine learning be used to detect hate speech?



A sort of artificial intelligence called machine learning can be used to learn from data. Data patterns can be discovered using it.

To identify hate speech, machine learning algorithms can be utilised. These algorithms are capable of text analysis and hate speech detection. They can also be used to gauge a text's tone. This can be used to spot hate speech that poses as a joke or a sarcastic comment. Automated hate speech identification is a crucial weapon in the fight against hate speech's propagation, especially on social media.

Traditional classifiers, deep learning and transfer learning-based classifiers, or a combination of both types of classifiers are methods for detecting hate speech using machine learning. A machine learning technique called deep learning can be used to learn from data.

Data patterns can be discovered using it. A kind of machine learning called transfer learning can be used to learn from data that has already been subjected to learning by another machine learning algorithm.

The advancement of numerous machine learning and natural language processing fields, including hate speech detection, has been greatly influenced by trained methods. Hate speech can be identified using these methods.

Naive Bayes, Support Vector Machines (SVM), Extreme Gradient Boosting (XGBoost), Multi-Layer Perception (MLP), and Long Short-Term Memory Networks are some of the machine learning algorithms that can be used to detect hate speech (LSTM).

Introduction 1.2 : -

This report will provide an overview on a machine learning model which will detect hate speech in a text.

My model uses various machine learning algorithms and natural language processing techniques to determine whether a text is hate speech, offensive speech, or normal speech.

My model is trained and tested on a supervised structured data set in which class 0 means Hate speech, class 1 means Offensive speech and class 2 means No hate speech.

The code first reads a CSV file with the labeled text using the pandas library before training a model using the Naive Bayes approach using the sklearn package. It used wordcloud to give a visual representation of the most frequent words in hate speech. The accuracy of this model is calculated with logistic regression.

The code also takes input from the user in the form of text and predicts whether the inputted text is Hate speech, offensive speech, or normal speech.

Chapter 2: -

Literature Survey

↳ Literature on hate speech: -

1. The article "A" Hate Speech Detection using Pre-trained Language Models" by Javid Dadashkarimi and Arash Einolghozati, published in Transactions of the Association for Computational Linguistics (2021), proposes a method for identifying hate speech in text using pre-trained language models and demonstrates how fine-tuning these models on a hate speech detection task can enhance their performance.

2. Fatemeh Torabi Asr and Mohammad Taher Pilehvar's article, "Cross-Lingual Hate Speech Detection via Transfer Learning," was published in Transactions of the Association for Computational Linguistics in 2021. In order to improve the performance of pre-trained models on a hate speech detection task in a different language, this research provides a method for cross-lingual hate speech detection via transfer learning.

3. Pascale Fung and Xingdi Yuan's article, "Adversarial Training for Robust Hate Speech Detection," was published in the Proceedings of the Conference on Empirical Methods in Natural Language Processing in 2021. In order to increase the resilience of hate speech detection algorithms against adversarial cases, this research suggests an adversarial training strategy.

4. ACM Computing Surveys (2020), "A Survey of Hate Speech Detection Using Natural Language Processing" by David Silva, Isabela Albuquerque, and Fabio Souza - This study presents a thorough overview of the state-of-the-art ways for identifying hate speech in text, including multimodal approaches, classical machine learning techniques, and deep learning techniques.

↳ Famous articles on hate speech detection: -

1. S. Weber and C. Potthast's article, "Automated Hate Speech Detection and the Problem of Offensive Language," was included in the 2017 Proceedings of the Eleventh International Conference on Web and Social Media. The difficulties of identifying hate speech in text are covered in

this article, along with the issues of offensive language and the limitations of the available machine-learning techniques.

2. The Conversation (2019), "AI-powered hate speech detection: the need for a critical approach," by Z. Zannettou and S. Blackburn - This article highlights the shortcomings of the available AI-powered hate speech detection techniques and the requirement for a more critical means of identifying and combating online hate speech.

3. R. David, T. K. Anitha, and S. Rajeswari's "Hate Speech Detection: A Review," which was released in the International Journal of Advanced Research in Computer Science and Software Engineering in 2019 - This article offers a thorough analysis of the most recent methodologies for identifying hate speech in text, including multimodal approaches, classical machine learning techniques, and deep learning techniques.

↳ Famous researchers on hate speech detection: -

1. Prof. Nello Cristianini is a well-known researcher in the area of machine learning and has significantly influenced the creation of algorithms for identifying hate speech in text.

2. Dr. Mona Diab, a specialist in natural language processing, has written a number of publications on the subject of detecting hate speech. Her work has focused on creating machine learning models that can recognise hate speech in Arabic and other languages.

3. Dr. Kai-Wei Chang is a machine learning and natural language processing expert who has made important contributions to the creation of deep learning models for the detection of hate speech.

4. Dr. Joachim Wagner is a researcher in the field of natural language processing who has written extensively on the topic of detecting hate speech and the moral issues that surround it.

5.

These researchers are among the most well-known in the discipline of identifying hate speech.

Their work has improved the accuracy and robustness of hate speech detection models, and they have published numerous papers in reputable publications and conferences.

Chapter 3: -

Methodology

↳ Important libraries of the code: -

1. **Panda**:- Pandas is a data analysis and manipulation software package created for the Python programming language. It includes specific data structures and procedures for working with time series and mathematical tables.

2. **Numpy: -** The Python package NumPy is used to manipulate arrays. Additionally, it has matrices and linear algebra-related functions. You may use it without restriction as it is an open-source project. Numerical Python is referred to as NumPy.

3. **CounterVectorizer: -** CountVectorizer is a wonderful tool provided by the Python scikit-learn module. It is used to turn a text into a vector depending on how frequently (count) each word appears across the entire text.

4. **Train test split: -** The train test split() method is used to divide our data into train and test sets. Our data must first be divided into features (X) and labels (y). The data frame is divided into X train, X test, Y train, and Y test. Using the X train and Y train sets, the model is trained and fitted.

5. **Decision tree; -** Decision Tree, a supervised machine learning algorithm, bases its conclusions on a set of rules, much like how people do.

6. **Wordcloud:** - Word clouds can be created using the wordcloud python library. A word cloud is a graphic depiction of the most frequent terms in a text or collection of texts, with each word's size according to how frequently it appears in the text.

7. **NLTK:** - The Python package for natural language processing is called Natural Language Toolkit (nltk). It offers a set of tools for processing linguistic data, including as parsing and semantic reasoning, as well as tools for working with human language data, such as tokenization, stemming, and tagging.

8. **Matplotlib:**- For the Python programming language and its NumPy numerical mathematics extension, Matplotlib is a graphing library. For integrating charts into programmes utilising all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK, it offers an object-oriented API.
   Line plots, scatter plots, bar plots, histograms, 3D plots, and many more types of graphs can be made with Matplotlib.

9. **Logistic regression:-** A statistical technique for categorization issues is logistic regression. In logistic regression, the probability of a binary result is predicted using a linear combination of input features (0 or 1). The class or label of the input is then predicted using this probability.

⤷ the methodology for the code: -

1. The first step is to import the necessary libraries such as pandas, numpy, nltk, matplotlib, Wordcloud and CountVectorizer ,train_test_split, decision tree from sklearn.

2. Next, we use pandas library to read a CSV file from a local file and convert it into a pandas Data Frame. This dataset contains supervised text which will be used to train the model.

3. Preprocessing the text data, which can involve actions like lowercasing, tokenization, and removing stop words, is the next step. The data must be prepared for further analysis in this step.

4. The training set is then used to create a vocabulary of words using the CountVectorizer, and the text is then transformed into numbers using the fit transform method.

5. We divided the dataset into training and test sets after loading it. The test set is used to assess the model's performance after it has been trained using the training set. Data is used 65% for testing and 35% for training.

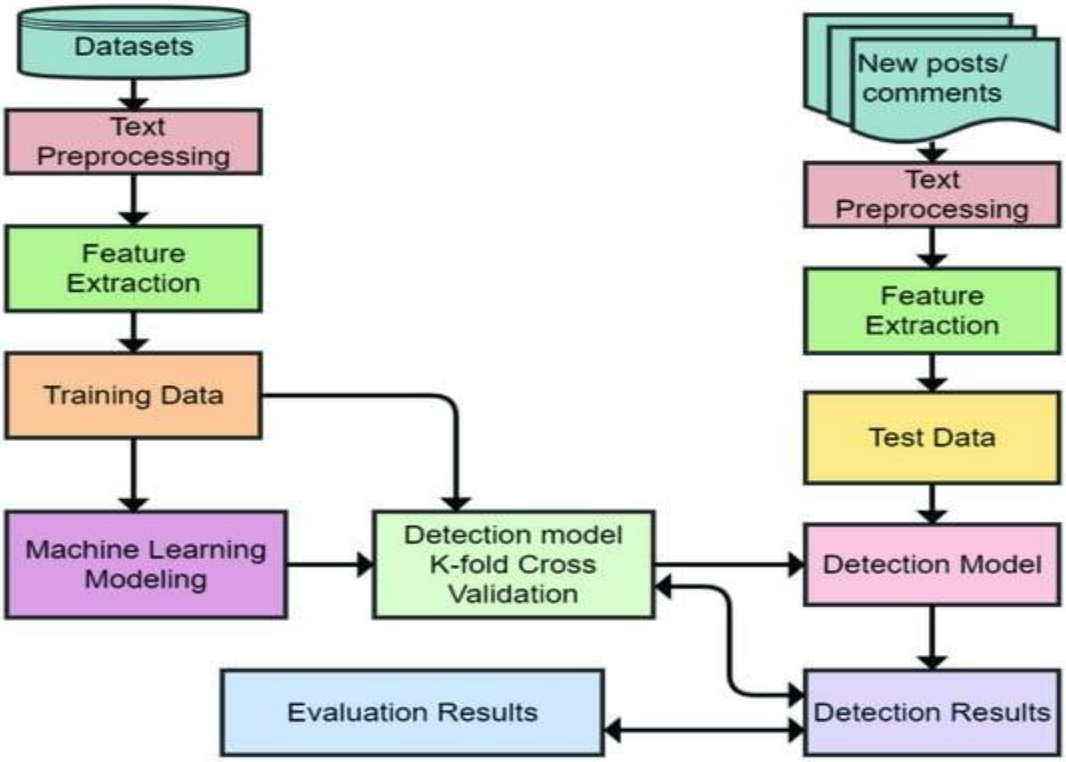
6. We train the model using the decision tree and fit method and passing the vectorized text and label of the text.

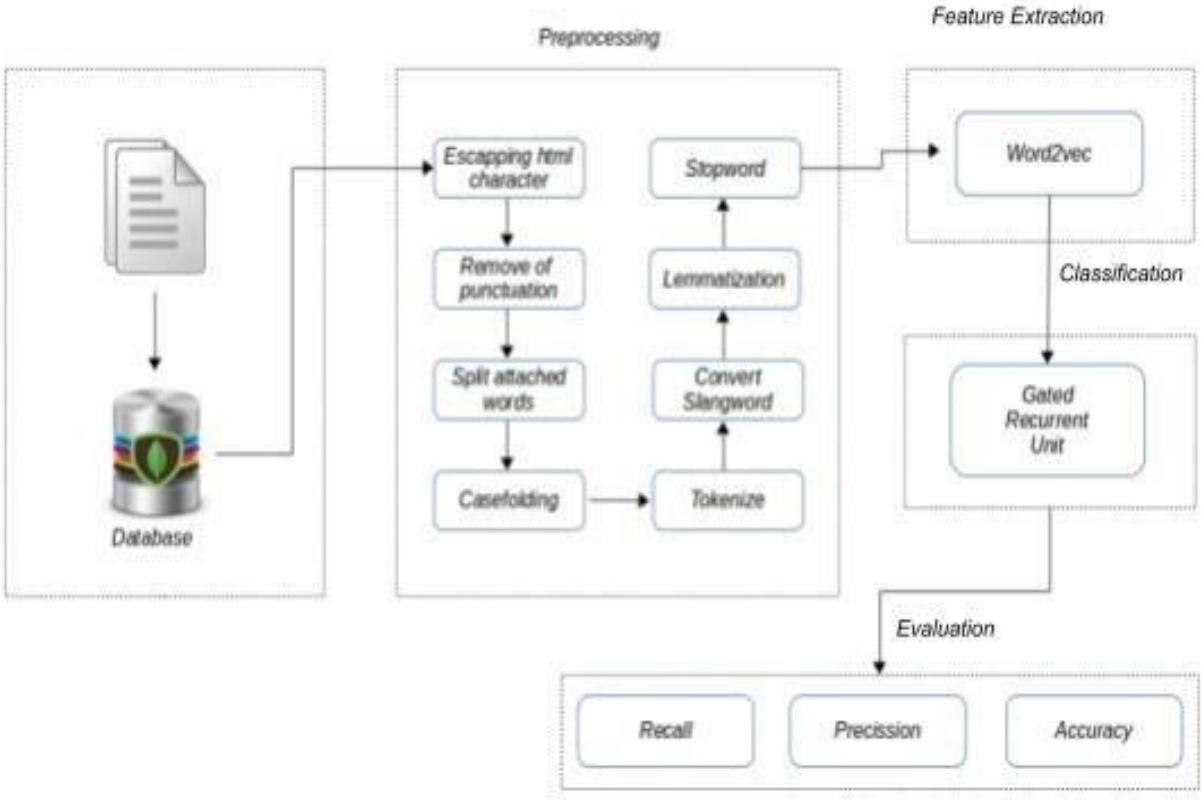
7. We use wordcloud to get a visualisation of most frequent hate speech words, offensive words and normal words.

8. After that we use accuracy_score and logistic regression from sklearn to get accuracy of the dataset.

9. In order to enhance the model's performance, the Byne method also includes a step called fine-tuning, in which the model is retrained using a new dataset or with modified parameters.

10. We use the predict method to make predictions on the input text and pass the vectorized text as input.


Datasets
Text Preprocessing
Feature Extraction
Training Data
Machine Learning Modeling
Detection model K-fold Cross Validation
Evaluation Results
New posts/ comments
Text Preprocessing
Feature Extraction
Test Data
Detection Model
Detection Results


Preprocessing
Feature Extraction
Database
Escapping html character
Remove of punctuation
Split attached words
Casefolding
Tokenize
Convert Slangword
Lemmatization
Stopword
Word2vec
Classification
Gated Recurrent Unit
Evaluation
Recall
Precission
Accuracy

Chapter 4:

Result And Discussion

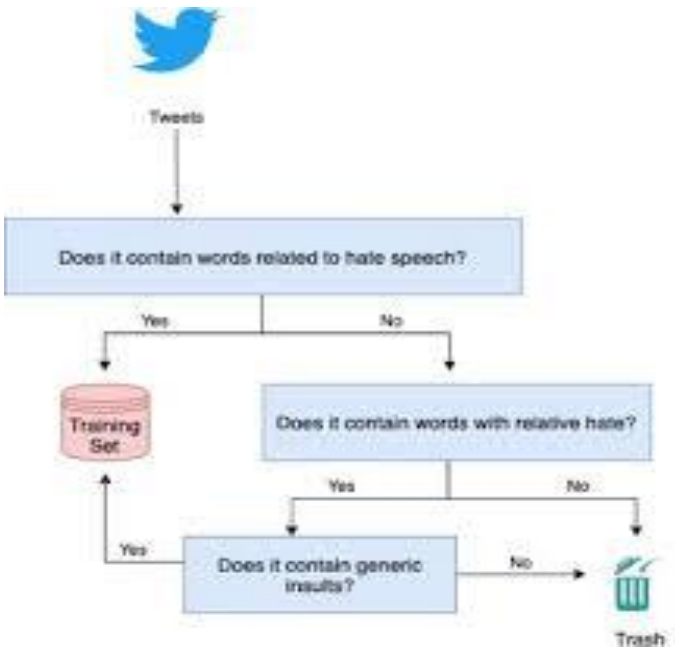
The specific dataset and the mode's performance would determine the results and discussion for the code.

⤷ Result:-

If the input text contains hate speech, offensive language, or non-hate speech, the code will output that information.

The accuracy of the model will be evaluated on the test set using the accuracy_score method.

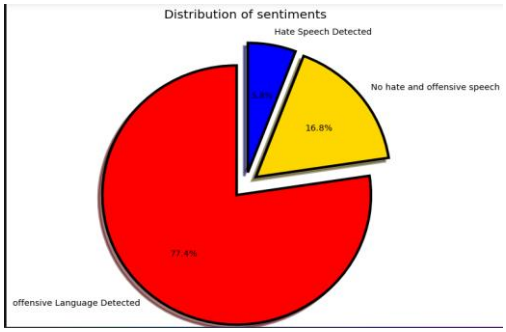
We also used classification_report to get the precision, recall and f1-score. We obtained a confusion matrix and visualization of most frequent hate speech words using sklearn and word cloud.



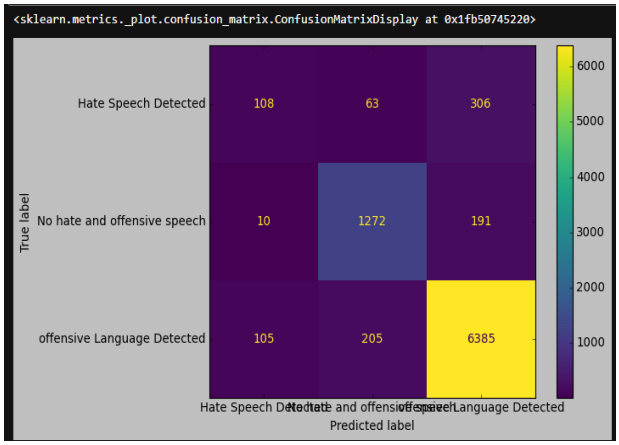
⤷ Discussion:-

On the basis of the outcomes, the model's performance would be examined during the discussion. If the accuracy is high, the model can correctly identify whether the text is hate speech, offensive speech, or otherwise. Low accuracy indicates that the model cannot accurately classify the text.

Pi-chart of different classes in data set:-



Confusion-Matrix after training and testing:-



Classification report of the data set:-

	precision	recall	f1-score	support
Hate Speech Detected	0.48	0.23	0.31	477
No hate and offensive speech	0.83	0.86	0.84	1473
offensive Language Detected	0.93	0.95	0.94	6695
accuracy			0.90	8645
macro avg	0.75	0.68	0.70	8645
weighted avg	0.89	0.90	0.89	8645

Accuracy:-

```
logreg_acc = accuracy_score(y_pred, y_test)
print("Test accuracy: {:.2f}%".format(logreg_acc*100))
```

```
Test accuracy: 89.82%
```

Chapter 5: -

Conclusion And Future Work

↳ Conclusion:-

To find hate speech in text, the code combines machine learning algorithms with natural language processing techniques. The model can categorise new text as hate speech, offensive speech, or non-hate speech after being trained on a dataset of supervised text. The accuracy of the model can be assessed using the accuracy score method, and the performance of the model depends on the particular dataset. The programme uses the user's input text to predict whether the output will be hate speech or not while also displaying the accuracy score.

The model's and the training dataset's limitations must also be taken into account. The quality of the data, the size of the dataset, and the difficulty of the problem may have an impact on the model's performance.

↳ Future work:-

We can improve the efficiency of our model by:-

1. Use a larger and more varied dataset: The quality and diversity of the training data are two of the most critical aspects of a hate speech detector's performance. The accuracy and robustness of the detector can be enhanced by employing a larger and more varied dataset that contains a variety of different types of hate speech.

2. Use cutting-edge machine learning methods: Deep learning and transfer learning are two methods that can help a hate speech detector perform better. These methods allow the model to extract from the data more intricate features, which enhances its capacity to generalise to new examples.

3. Integrate context-based data: Context-based data, such as the social network structure and the user's history, can be used to enhance hate speech detection. It can provide more context for the text and aid in better understanding the hate speech.

4. Human evaluation: Conducting a human evaluation is a crucial step in creating a hate speech detector. It is crucial to have a diverse group of human reviewers who can assess the detector's performance and give feedback on its robustness and accuracy.

5. Ensemble models: To enhance the performance of the hate speech detector, ensemble models can be used to combine the predictions of various models.

References

[1] Pete Burnap and Matthew L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making", Policy & Internet, vol. 7, no. 2, pp. 223-242, 2015.

[2]"Deep Learning for Hate Speech Detection: A Survey" by F. Hashemi and M. E. Newman, published in IEEE Access (2020)

[3] Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques https://ieeexplore.ieee.org/document/9062549

[4] Using tools like Twitter sentiment analysis and dealing with text data (e.g. https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data predictive-python/)

[5] Full tweet texts are provided with their labels for training data.

https://drive.google.com/drive/folders/1uQiyJ_mDlOCcecMw7C-JYUs9bGnVJ_j8