Hate Speech Detection

Name : M.Lakshmi Narayana Varma

Mentor : Dr. N. JaganMohan

---

1. Abstract :

Hate speech has been a problem in this digital media age. It has the potential to create a toxic online environment and can cause affects the online as well as offline social behavior of an individual [2]. The project aims to design and implement a machine learning-based hate speech detection system to help in detecting and classifying the hate speech in social media and online forums effectively.

Data preprocessing steps such as Tokenisation, Lower-casing, and Vectorisation will be executed[7] before subjecting data to feed into the machine learning classifiers, such as Logistic Regression, Support Vector Machines, Naïve Bayes ,Decision Trees, etc. These classifiers are trained, fine-tuned, and evaluated based on the type of text data that is used. The classifiers are evaluated based on different metrics, such as accuracy, precision, recall, and F1-score, to check their effectiveness.

2. Introduction :

The rapid growth of social media has made it easier for people to share their opinions. However, this has also led to an increase in hate speech, which can harm vulnerable communities and damage social unity. Detecting hate speech among the many online comments is a difficult task because it involves analyzing countless words and the subtle differences in how people express themselves[1].

This project aims to develop an automated hate speech detection system by employing Machine Learning techniques. We will apply natural language processing to analyze textual data for patterns of hate speech. Our procedure will include the collection of a labeled dataset, feature extraction through various methods, and training several different machine-learning models to ultimately achieve satisfying hate speech classification.

3. Workflow for Hate Speech Detection:

1.collecting the dataset

2. Data Preprocessing:

- Tokenize, lowercase, remove stop words, stem/lemmatize[7].
- **One-Hot Encoding**: Convert categorical data (e.g., labels) into binary vectors to ensure the model treats categories without implying a numerical order.
- Vectorize using TF-IDF, or Word Embeddings[7].

3.Model Selection:

- Test classifiers: Logistic Regression, SVM, Naïve Bayes, Decision Trees.
- Deep learning techniques, exploring deep learning models like LSTM or Transformers.

4.Evaluation Metrics:

- Measure accuracy, precision, recall, and F1-score.

5. Model Tuning:

i. Optimize parameters (grid search, cross-validation).

ii. Consider ensemble methods for improved accuracy.

6.Selecting Dataset for hate speech detection:

Dataset: Labelled Hate Speech Detection

- Content: The dataset contains 3,000 comments and posts scraped from social media platforms including Reddit, Twitter, and 4Chan in 2022 [3].
- Labels:
    - 0: Non-Hateful (comments not containing hate speech)
    - 1: Hateful (comments containing hate speech)
- Split: 80% non-hateful (2,400 comments) and 20% hateful (600 comments), providing a balanced dataset for classification tasks.

	A	B	C
1	Platform	Comment	Hateful
2	Reddit	Damn I thought they had strict gun laws in Germany	0
3	Reddit	I dont care about what it stands for or anything its connected to, I like the shields ...	0
4	Reddit	It's not a group it's an idea lol	0
5	Reddit	So it's not just America!	0
6	Reddit	The dog is a spectacular dancer considering he has two left feet.	0
7	Reddit	If ppl dont wear masks you complain... If ppl wear masks u also complain...	0
8	Reddit	We should send them All back to africa	1
9	Reddit	Checking to see if it's whataboutism from the past, yup indeed, it was whataboutism from the past	0
10	Reddit	As a european, I approve this.	0
11	Reddit	Idk which of these groups to join, they're all so cool	0

1. Choosing Between Machine Learning or Deep Learning for Hate Speech Detection:

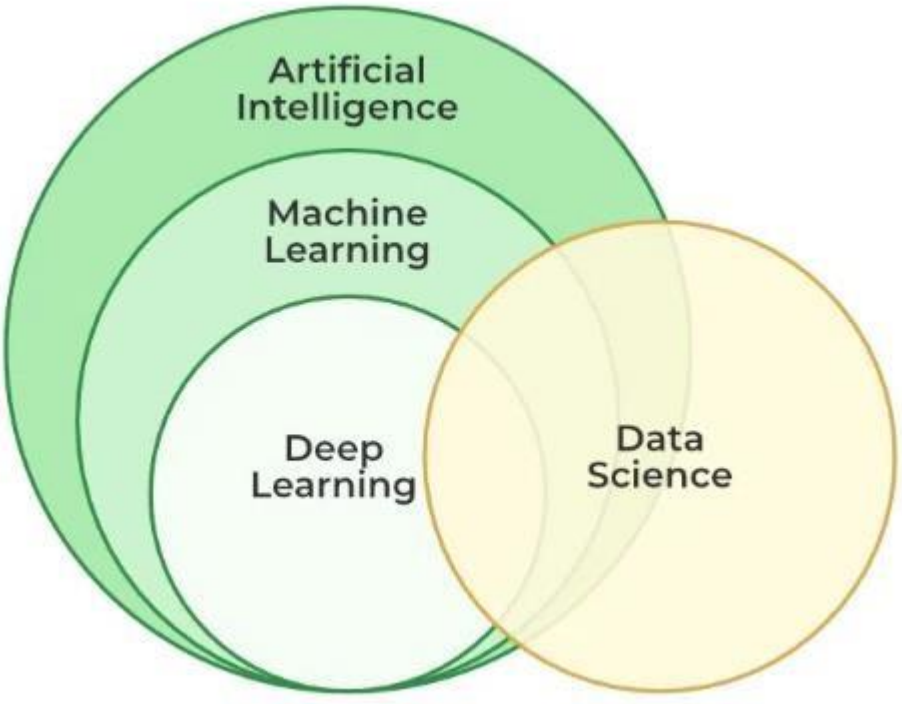
Difference Between Machine Learning and Deep Learning

1. **Machine Learning (ML)**:
    a. **Definition**: Algorithms that learn from structured data to predict outputs and discover patterns in that data[4]
    b. **Data Needs**: Works well with smaller datasets and requires less computational power.
    c. **Interpretability**: Models like Logistic Regression and Decision Trees are easier to interpret.
    d. **Algorithms**: Common ML algorithms include Logistic Regression, SVM, Naïve Bayes, and Decision Trees.

2. **Deep Learning (DL)**:

    a. **Definition**: Algorithms based on highly complex neural networks that mimic the way a human brain works to detect patterns in large unstructured data sets.
    b. **Data Needs**: Best suited for large datasets and requires significant computational resources.[4]

    c. **Performance**: Excels at handling unstructured data by capturing context and subtleties.
    d. **Models**: Examples include RNNs, LSTM, and Transformers[4].



Choice for Hate speech detection:

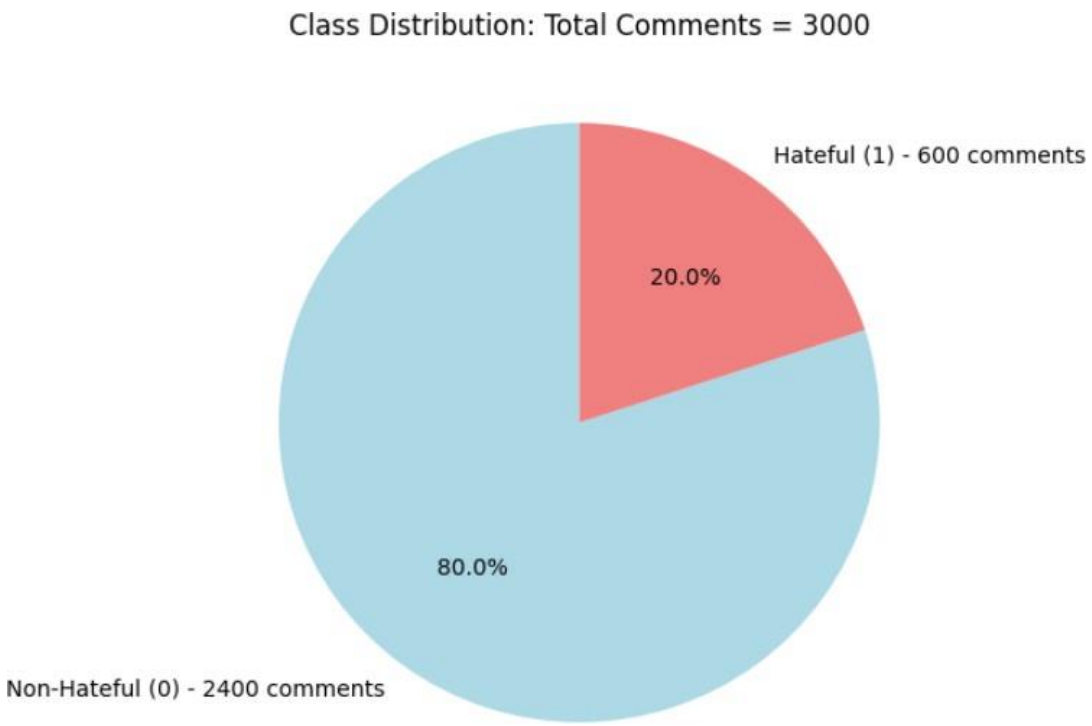
using **Machine Learning models** for this hate speech detection project because:

- **Dataset Size**: Our dataset is moderate in size, suitable for ML models. (3,000 comments)
- **Efficiency**: ML models are faster to train and tune, allowing quicker iterations

3. GRAPHICAL REPRESENTATION OF DATASET:

Class Distribution:

- **Plot Type**: Pie Chart
- **Purpose**: Show the balance of non-hateful (0) vs. hateful (1) comments.



Oversampling (Augmenting Minority Class):

HANDLING CLASS IMBALANCE VIA OVERSAMPLING

In the dataset, there was a significant class imbalance between the Hateful and Non-Hateful comments. The distribution showed 600 hateful comments (20%) and 2400 non-hateful comments (80%), which could have led to biased model performance. To address this, we employed the Random Oversampler technique[6], a widely used method to augment the minority class.

STEPS TAKEN:

Feature Vectorization:

Text data in the Comment column was transformed into numerical vectors using the TF-IDF Vectorizer. This technique converts the textual data into a matrix of numerical values while considering the importance of words in the document corpus[7].

Applying Random Oversampler technique,:

Random Oversampler technique, was used to generate synthetic samples for the minority class (**Hateful**) by interpolating between existing data points. This approach ensures that new samples are meaningful and lie within the feature space of the minority class[6].

The minority class was oversampled to match the size of the majority class, resulting in an equal distribution of 2400 samples for both classes.

Balanced Dataset:

The dataset after oversampling has the following distribution:

i. **Hateful Comments:** 2400
ii. **Non-Hateful Comments:** 2400

```
Original Class Distribution:
Hateful
0    2400
1     600
Name: count, dtype: int64
Resampled Class Distribution:
Hateful
0    2400
1    2400
Name: count, dtype: int64
```

One hot Encoding :

```
                                                Comment  Hateful  \
2995  kike shilling look like ryan done redpill peop...        1
2996                                          bait right        0
2997                                        like one lot        0
2998                         kike making money heroin new        1
2999  desecrate men making gaytrannies woman making ...        1

                                       Cleaned_Comment  Platform_4Chan  \
2995  kike shilling look like ryan done redpill peop...            True
2996                                          bait right            True
2997                                        like one lot            True
2998                         kike making money heroin new            True
2999  desecrate men making gaytrannies woman making ...            True

      Platform_Reddit  Platform_Twitter
2995            False             False
2996            False             False
2997            False             False
2998            False             False
2999            False             False
```

1. Comment: This is the original comment. For example, "kike shilling look like ryan done redpill peop...". (from the above output)

2. Hateful: A label indicating if the comment is hateful (1 means the comment is hateful, 0 means it is not).

3. Cleaned_Comment: This is the same comment, but after cleaning (like removing extra spaces, stop words, or unwanted characters). It is now simplified, e.g., "kike shilling look like ryan done redpill peop...".

4. Platform (e.g., Platform_4Chan, Platform_Reddit, Platform_Twitter): These are one-hot encoded columns, where a "True" means the comment comes from that platform, and "False" means it does not. For example, for the first row, the comment came from 4Chan, so the "Platform_4Chan" column is marked as True, and the others ("Platform_Reddit", "Platform_Twitter") are marked as False.

TEXT PREPROCESSING FOR HATE SPEECH DETECTION:

1. **Lowercasing**: Convert all text to lowercase to maintain consistency and reduce redundancy in text data[7] .

2. **Tokenization**: Split sentences into individual words or tokens for easier analysis and model training[7].

3. **Removing Punctuation**: Remove punctuation marks, which generally don't contribute to the meaning in hate speech detection[7].

4. **Removing Stop Words**: Remove common words (like "is," "and," "the") that don't carry significant meaning and can add noise[7].

5. **Stemming**: Reduce words to their root form to standardize variations of the same word.

6. **Removing Numbers**: Remove numbers, as they often don't add useful information in text classification[7].

7. **Removing Extra Whitespaces**: Remove unnecessary spaces for cleaner text formatting[7].

8. **Text Normalization**: Standardize text to handle abbreviations, misspellings, or informal language[7] .

9. **Handling Imbalanced Data**:

    a. **Resampling**: Use oversampling (e.g., SMOTE) to increase minority class samples or undersampling to reduce the majority class samples.

    b. **Class Weight Adjustment**: Adjust model class weights to pay more attention to the minority class (hate speech) during training.

    c. **Ensemble Methods**: Use techniques like bagging or boosting to improve model robustness on imbalanced dataset

MODELS USED :

1. Random Forest
2. Logistic Regression
3. Support Vector Machine
4. Gradient Boosting
5. LSTM

RandomForest Model :

A Random Forest Classifier is used to predict whether a given comment is Hateful or Non-Hateful based on the preprocessed textual data. The model is trained using a balanced dataset and evaluated on its performance using standard metrics.

SupportVectorMachine (SVM) Model:

An SVM Classifier is employed to predict the nature of comments (Hateful or Non-Hateful). Using a linear or kernel-based approach, the model finds the optimal hyperplane that separates the classes in the feature space. The performance is assessed using standard evaluation metrics.

Gradient Boosting Model:

Gradient Boosting is utilized to classify the comments into Hateful or Non-Hateful categories. The model is trained using an ensemble of weak learners (decision trees) and focuses on minimizing the classification error iteratively. The performance is evaluated using standard metrics like accuracy and F1-score.

LSTMModel:

An LSTM (Long Short-Term Memory) neural network is employed to predict whether a given comment is Hateful or Non-Hateful. By capturing the sequential dependencies in textual data, the model leverages embedding layers, spatial dropout, and recurrent layers to learn meaningful patterns. The performance is evaluated using validation and test accuracy metrics.

Libraries Used(RANDOM FOREST)

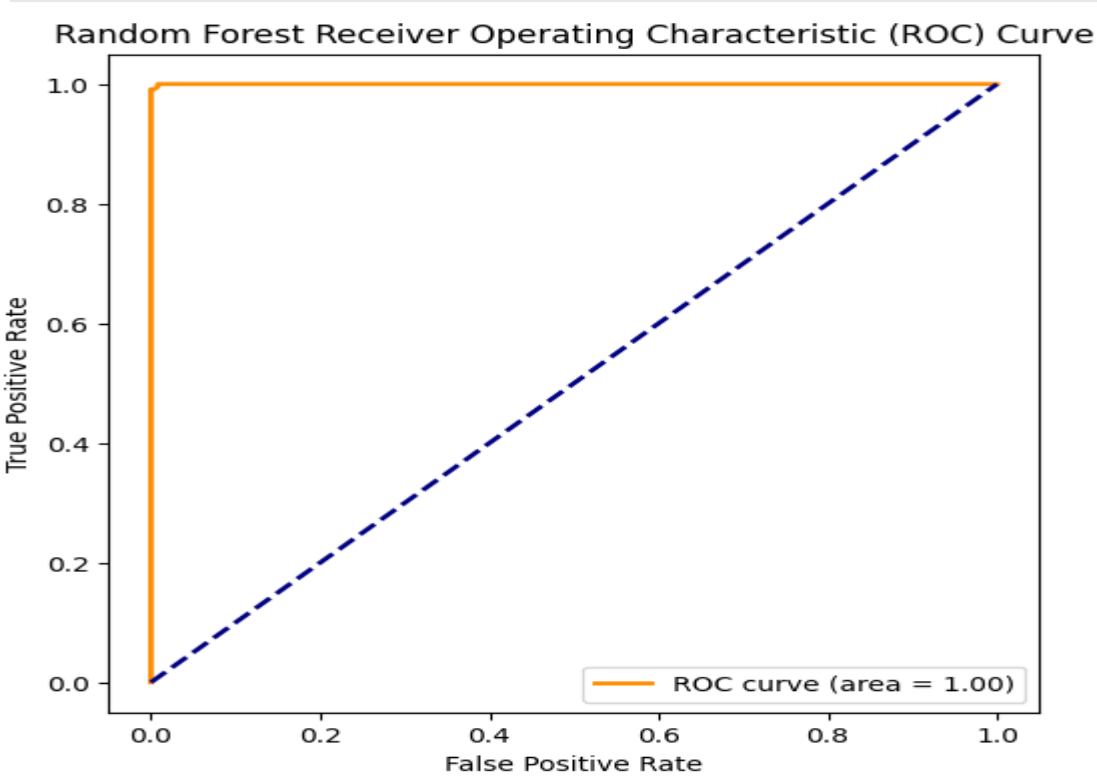
RandomForestClassifier: Implements the Random Forest algorithm for classification.

train_test_split: Used to split the dataset into training and testing sets[6].

classification_report: Provides detailed performance metrics such as precision, recall, F1-score, and support.

accuracy_score: Measures the overall accuracy of the model.

ROC Curve :



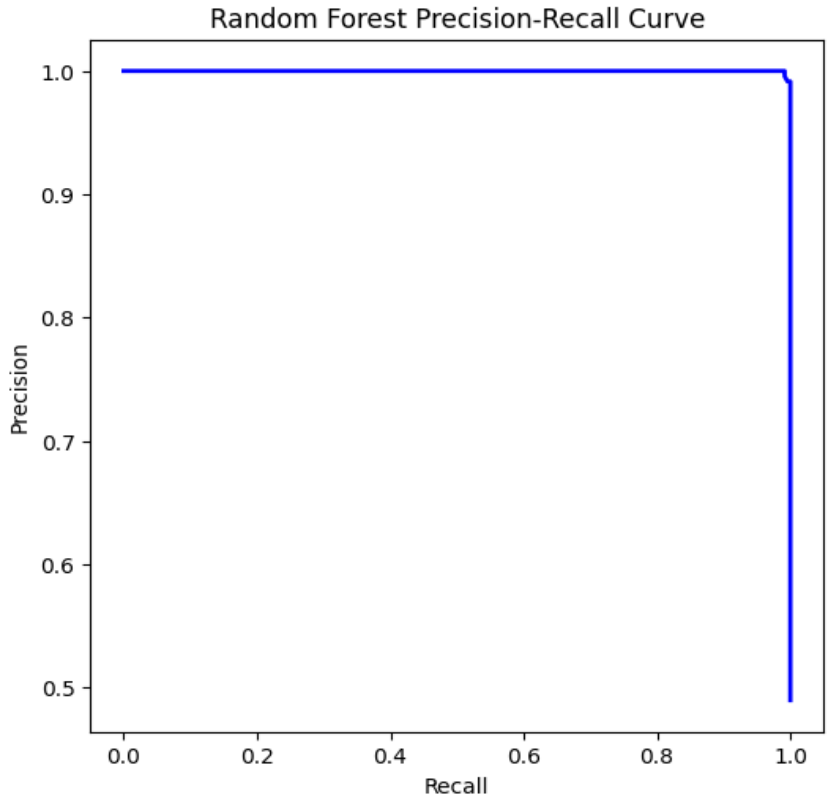
- **True Positive Rate (Recall)** is on the Y-axis:
  It tells how well your model correctly identifies hateful speech.

- **False Positive Rate** is on the X-axis:
  It shows how often the model incorrectly labels non-hateful speech as hateful.

- **Blue Line**:
  This line represents the performance of your model. The closer it is to the top-left corner, the better the model performs.

- **AUC (Area Under the Curve)**:
  The AUC value here is **0.96**.
  AUC measures how well the model can separate hateful from non-hateful content.
  An AUC of **1.0** means perfect classification, and **0.96** is very close to perfection, indicating excellent performance.

Precision-Recall Curve:



- The curve shows that the model maintains **high precision** (accuracy in identifying hate speech) across most recall levels .

- At very high recall, precision drops, which means the model includes some false positives to capture more hate speech

Classification Report by using Random Forest:

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       245
           1       0.98      1.00      0.99       235

    accuracy                           0.99       480
   macro avg       0.99      0.99      0.99       480
weighted avg       0.99      0.99      0.99       480
```

Accuracy:
The overall accuracy of the model is 99%, indicating the model correctly predicts 99% of the cases in the dataset.

Class-Level Metrics:

For class 0 (non-hate speech):

- **Precision**: 1.00 – 100% of the predictions for non-hate speech are correct.
- **Recall**: 0.98 – 98% of the actual non-hate speech examples are correctly identified.
- **F1-Score**: 0.99 – Balances precision and recall for class 0.

For class 1 (hate speech):

- **Precision**: 0.98 – 98% of the predictions for hate speech are correct.
- **Recall**: 1.00 – 100% of the actual hate speech examples are correctly identified.
- **F1-Score**: 0.99 – Balances precision and recall for class 1.

Model	Accuracy	Precision (0)	Recall (0)	F1-score (0)	Precision (1)	Recall (1)	F1-score (1)	Macro Average	Weighted Average
Random Forest	99%	1.00	0.98	0.99	0.98	1.00	0.99	0.99	0.99
Logistic Regression	92%	0.88	0.97	0.92	0.96	0.86	0.91	0.92	0.92
Support Vector Machine	100%	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00
Gradient Boosting	99%	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99

Model	Training Accuracy	Testing Accuracy
1.Gradient Boosting	0.9213541666666667	0.9166666666666666
2. Logistic Regression	0.9901041666666667	0.9854166666666667
3. Random Forest	1.0	0.9875
4. Support Vector Machine	1.0	0.9958333333333333

Results of LSTM Model :

• **Metric**	Trial ID: 0	Trial ID: 1	Trial ID: 2 (Best)
• **Embedding Dimension**	132	132	228
• **Spatial Dropout Rate**	0.3	0.2	0.4
• **LSTM Units**	192	64	128
• **LSTM Dropout**	0.3	0.2	0.2
• **Recurrent Dropout**	0.4	0.2	0.2
• **Learning Rate**	0.00057	0.00058	0.00161
• **Final Validation Accuracy**	98.02%	98.54%	98.65%
• **Test Accuracy**	98.02%	98.54%	98.54%

Best Hyperparameters:

- Embedding Dimension: 228
- Spatial Dropout Rate: 0.4
- LSTM Units: 128
- LSTM Dropout: 0.2
- Recurrent Dropout: 0.2
- Learning Rate: 0.00161

Results :

```
1/1 [==============================] - 0s 369ms/step
Text: 'I can't believe how rude that Jew is'
Prediction: Hate

1/1 [==============================] - 0s 31ms/step
Text: 'You are a black person.'
Prediction: Hate

1/1 [==============================] - 0s 15ms/step
Text: 'I will kill you.'
Prediction: Hate

1/1 [==============================] - 0s 14ms/step
Text: 'The weather is beautiful today, and I am enjoying the sunshine.'
Prediction: Non-Hate

1/1 [==============================] - 0s 16ms/step
Text: 'I love the way people help each other here; it's so inspiring.'
Prediction: Non-Hate
```

SUMMARY OF ACHIEVEMENTS:

1. **Best Model**: Support Vector Machine (SVM) with 99.58% testing accuracy and exceptional generalization capabilities.
2. **Best Precision**: Gradient Boosting with 91.67% precision, showcasing reliable classification of positive instances.
3. **Best F1 Score**: Logistic Regression with an F1 Score of 98.54%, highlighting its balanced performance between precision and recall.
4. **Overfitting**: Random Forest exhibited 100% training accuracy but slightly lower testing accuracy of 98.75%, indicating mild overfitting.
5. **Balanced Performance**: Logistic Regression achieved a well-balanced performance with 99.01% training accuracy and 98.54% testing accuracy, ensuring consistent results across datasets.

Future Scope

1. **Integration with Real-Time Systems**: The models can be deployed in real-time systems to monitor and classify data dynamically, ensuring timely and accurate predictions.

2. **Hybrid Models**: Combining multiple models, such as LSTM with Gradient Boosting, can further improve accuracy, robustness, and adaptability to complex data patterns.

CONCLUSION :

In conclusion, the study highlighted the effectiveness of Logistic Regression and SVM in achieving high accuracy and robust generalization for data classification. While Random Forest showed some overfitting, its accuracy was still notable. The results emphasize the importance of selecting the right hyperparameters and balancing datasets to optimize model performance. Future work in feature engineering and hybrid modeling may improve these results even further.

References:

1. https://www.kaggle.com/code/kirollosashraf/hate-speech-and-offensive-language- detection/input

2. https://www.kaggle.com/datasets/saurabhshahane/cyberbullying-dataset

3. data_huang_devansh.csv - Mendeley Data

4. https://figshare.com/articles/dataset/Labelled_Hate_Speech_Detection_Dataset_/19686954? file=34965762
https://www.geeksforgeeks.org/difference-between-machine-learning-and-deep-learning/

5. https://www.researchgate.net/figure/Hate-Speech-Detection-Flowchart_fig1_325414504

6. https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

7. https://www.kaggle.com/code/abdmental01/text-preprocessing-nlp-steps-to-process-text

8. https://www.geeksforgeeks.org/hate-speech-detection-using-deep-learning/

9. https://www.sciencedirect.com/science/article/pii/S0925231223003557