TOWARDS SAFER DIGITAL SPACES: DETECTION OF UNSAFE CONVERSATIONS ON SOCIAL MEDIA

S.V. Patil[*1], Sanika H. Lokare[*2], Sai G. Mane[*3], Jyotirmoyee Rout[*4], Vaishnavi M. Chikhale[*5]

[*1,2.3,4,5]Computer Engineering Department, Sinhgad College Of Engineering, Pune, India.

DOI : https://www.doi.org/10.56726/IRJMETS55694

ABSTRACT

In response to the proliferation of unsafe conversations on social media platforms, this project aims to develop an automated detection system using natural language processing (NLP) and machine learning techniques. By analyzing linguistic patterns, sentiment, and contextual cues, the system seeks to distinguish between benign and harmful interactions, including harassment, hate speech, and cyberbullying. Leveraging supervised and unsupervised learning algorithms trained on labeled datasets, the system aims to identify key indicators of toxicity and aggression proactively. Integration with user-specific features and network analysis will provide insights into the dynamics of online interactions. Through collaboration with social media platforms and community stakeholders, the deployment of this detection system aims to contribute to the establishment of safer and more inclusive digital environments.

Keywords: Social Media, Unsafe Conversations, Detection, Natural Language Processing, Machine Learning, Cyberbullying, Hate Speech, Online Safety.

I. INTRODUCTION

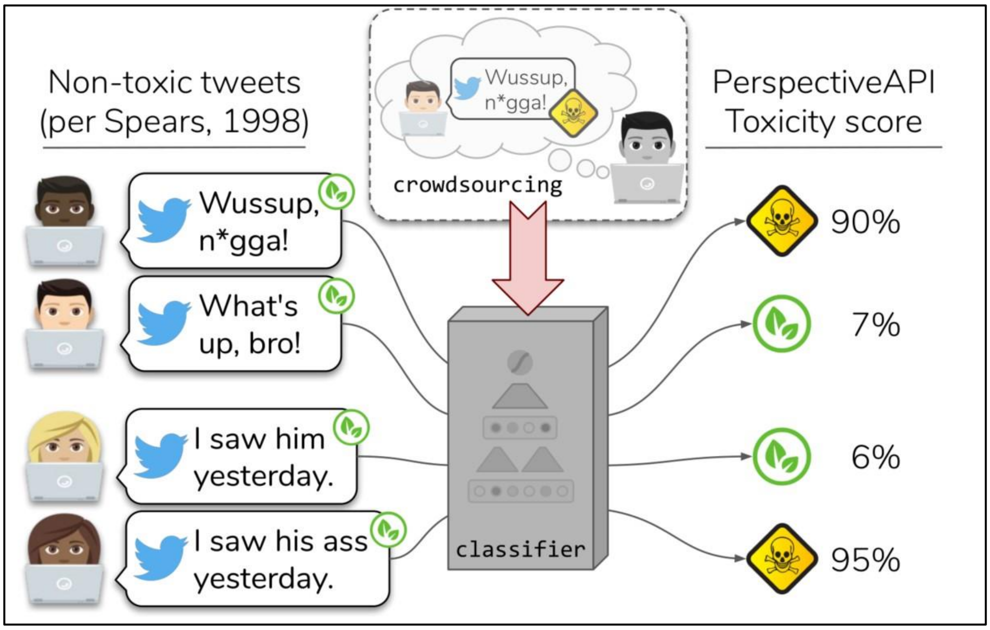
Social media platforms have become integral parts of modern communication, facilitating connections, information sharing, and community building on an unprecedented scale. With billions of users worldwide, platforms such as Facebook, Twitter, Instagram, Whatsapp, and Reddit have reshaped the way individuals interact and engage with each other, transcending geographical boundaries and cultural barriers. However, the exponential growth of social media has also brought to light significant challenges related to online safety and the prevalence of harmful behaviors within digital spaces.

One of the most concerning phenomena on social media is the proliferation of unsafe conversations, encompassing a wide range of harmful behaviors including harassment, cyberbullying, hate speech, and threats of violence. These behaviors not only undermine the well-being and mental health of individuals but also contribute to the erosion of trust and civility within online communities. Moreover, the pervasive nature of social media amplifies the impact of harmful content, potentially reaching millions of users within seconds and perpetuating cycles of toxicity and aggression.

Traditional approaches to moderation often rely on manual review processes, which are labor-intensive, time-consuming, and prone to biases. As a result, many harmful interactions go unnoticed or inadequately addressed, exacerbating the negative effects on affected individuals and the broader community.

In response to these challenges, there is a pressing need for innovative solutions that leverage technology to proactively detect and mitigate unsafe conversations on social media. Advances in natural language processing (NLP), machine learning, and computational linguistics offer promising avenues for developing automated detection systems capable of analyzing textual content and identifying linguistic markers associated with toxicity and aggression. By harnessing the power of data-driven approaches, it becomes possible to scale moderation efforts, improve response times, and enhance the overall safety of digital spaces.

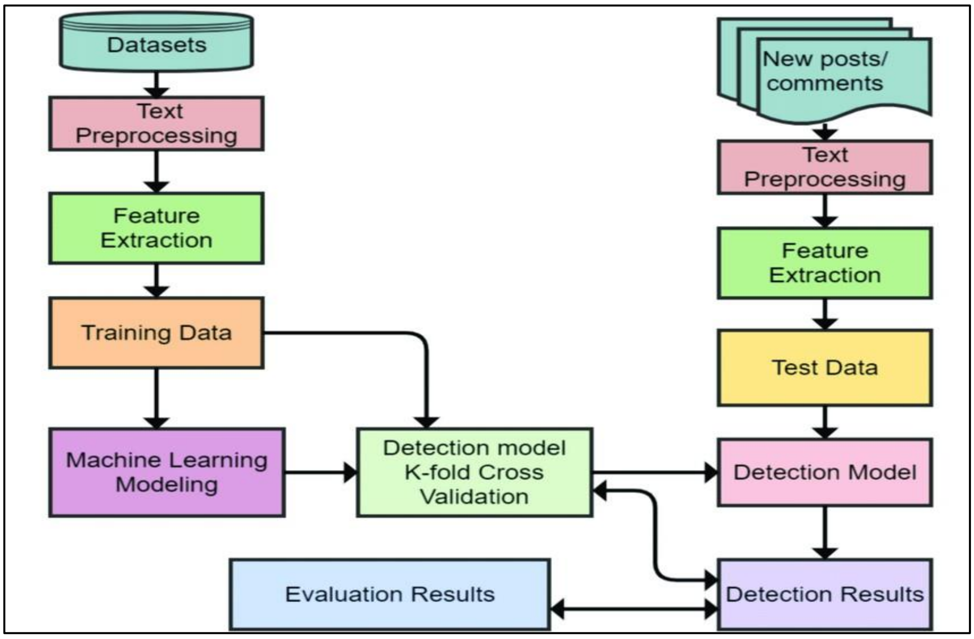
This project aims to contribute to the ongoing efforts towards creating safer digital environments by designing and implementing a robust detection system for identifying unsafe conversations on social media platforms. Through a combination of advanced NLP techniques, machine learning algorithms, and interdisciplinary insights from psychology and sociology, we seek to develop a scalable and efficient solution that can adapt to evolving online threats and user behaviors.



Motivation:

This project is driven by the urgent need to address the proliferation of harmful behaviors on social media platforms. Unsafe conversations, including harassment, cyberbullying, and hate speech, not only endanger individual well-being but also erode trust and civility within online communities. Leveraging advancements in AI and machine learning, we aim to develop automated detection systems to proactively identify and mitigate these behaviors, fostering safer and more inclusive digital spaces for all users.

II. METHODOLOGY



System Architecture

Our methodology leverages machine learning (ML) algorithms implemented in Python programming language, along with the Flask web framework, to develop a robust system for detecting unsafe conversations on social media platforms. The process involves several key steps:

1. Data Collection and Preprocessing:

- Data collection is the process of gathering raw data from various sources such as databases, APIs, files, or web scraping.

- We collect a diverse dataset of social media conversations containing both safe and unsafe content. This dataset is annotated and labeled to facilitate supervised learning.
- Data preprocessing is the cleaning and transformation of raw data into a format suitable for analysis and modeling.
- Preprocessing steps such as removal of stop words and stemming are applied to clean and standardize the text data.
- Steps used in our project:
- Lowercasing: Converts all text to lowercase to ensure uniformity in text analysis.
- Removing URLs: Eliminates URLs from the text data since they typically don't contribute to the meaning of the text.
- Removing HTML tags: Strips HTML tags, which are often present in text scraped from websites.
- Removing punctuation: Eliminates punctuation marks from the text.
- Removing newline characters: Removes newline characters, ensuring that each text is in a single line.
- Removing alphanumeric characters: Eliminates alphanumeric characters that might not carry significant meaning.
- Removing stopwords: Removes common English stopwords like 'is', 'and', 'the', etc., which occur frequently but usually don't add much value to the analysis.
- Stemming: Reduces words to their root form to normalize the text data. Stemming algorithms like the Snowball Stemmer aim to treat related words as the same (e.g., 'running' and 'runs' become 'run').

2. Feature Engineering:

- In Feature engineering, we transform raw data into features that better represent the underlying problem and improve the performance of machine learning models.
- In our project, a Count Vectorizer is used, which is a feature extraction technique used to convert a collection of text documents into a matrix of token counts.
- Count Vectorizer tokenizes the text documents into words or n-grams (contiguous sequences of n items from the text), assigns a unique integer ID to each token, and counts the occurrences of each token in each document. The resulting matrix represents the frequency of each token in each document.
- Parameters:
  - ○ lowercase: Whether to convert all text to lowercase.
  - ○ Token pattern: Regular expression defining what constitutes a token (default is words of 2 or more alphanumeric characters).
  - ○ Stop words: List of stop words to be removed from the text.
  - ○ N-gram range: Tuple specifying the range of n-grams to be extracted (e.g., (1, 1) for unigrams, (1, 2) for unigrams and bigrams).

3. Model Selection and Training:

- Model selection involves choosing an appropriate machine learning algorithm for the problem at hand, while model training refers to fitting the chosen model to the training data.
- In our project, we have chosen the Decision Tree Classifier as the machine learning algorithm.
- Decision trees are non-parametric supervised learning models used for classification and regression tasks. They learn simple decision rules inferred from the data features to predict the target variable's value.
- Working Principle: Decision trees recursively split the feature space into subsets, based on the feature that provides the best split (i.e., maximizes information gain or minimizes impurity). This process continues until certain stopping criteria are met, such as a maximum depth of the tree, minimum number of samples required to split a node or minimum decrease in impurity.
- Splitting Criteria: Common splitting criteria include:
  - ○ Gini impurity: Measures the probability of incorrectly classifying a randomly chosen element if it were randomly labeled.
  - ○ Entropy: Measures the uncertainty or randomness of a dataset's distribution. It is minimized when all samples belong to a single class.

- ○ Information Gain: Measures the reduction in entropy or impurity achieved by splitting the data on a particular feature.
- Models are trained on the labeled dataset using appropriate training-validation splits or cross-validation techniques to optimize performance metrics such as accuracy, precision, recall, and F1-score.

4. Model Evaluation and Validation:

- The trained models are evaluated on a separate test dataset to assess their performance in detecting unsafe conversations.
- Performance metrics are computed and analyzed to identify the most effective model(s) for deployment.
- In our project, the performance metrics used are –
- Prediction: The trained model is used to predict labels for the test data (X-test) using the predict() method.
- Accuracy Score: The accuracy score is calculated by comparing the predicted labels with the actual labels (y-test). It measures the proportion of correctly classified instances.
- Predicting outcomes: The model is then used to predict the label for a given input (e.g., the word "bitch" as an offensive word).
- Printing the predicted outcome: The predicted label is printed, and based on the label, a corresponding message is printed indicating whether the text is offensive or not.

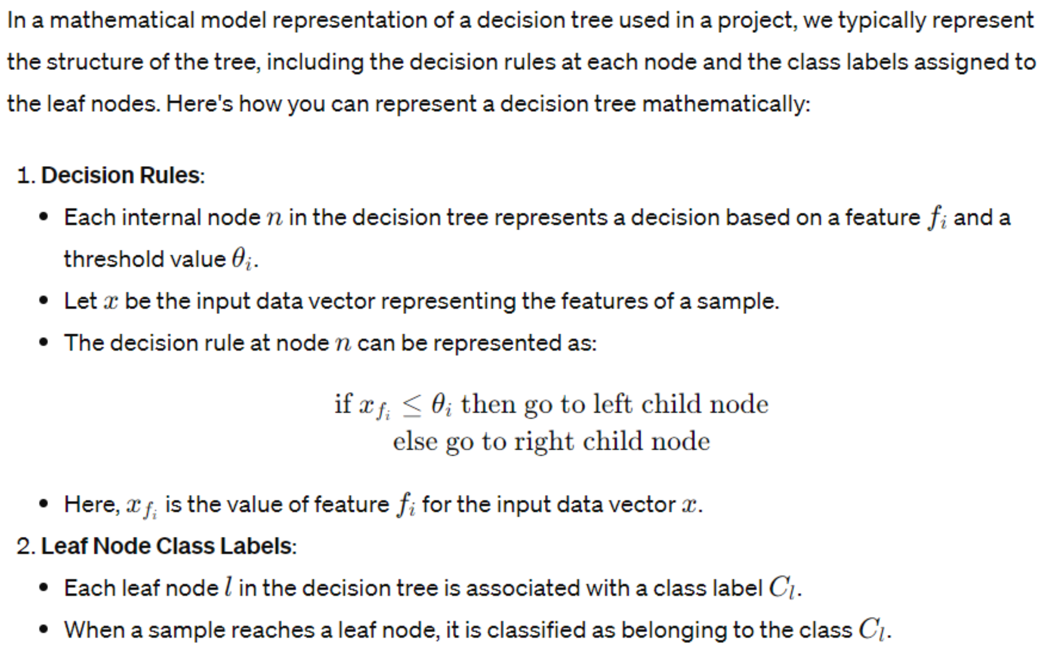
5. Integration with Flask Framework:

- The selected ML model is integrated into a web application using the Flask framework, a lightweight and extensible Python web framework.
- We design and develop a user-friendly interface where users can input social media conversations for analysis.
- Flask routes are defined to handle incoming requests, process the input data, and invoke the ML model for prediction.

6. Deployment and Testing:

- The Flask-based application is deployed on a web server to make it accessible to users.
- Comprehensive testing is conducted to ensure the reliability, scalability, and security of the system.
- By following this methodology, we aim to create an effective and scalable solution for detecting unsafe conversations on social media platforms, thereby contributing to the promotion of safer digital spaces for all users.

Mathematical Model (Decision Tree):

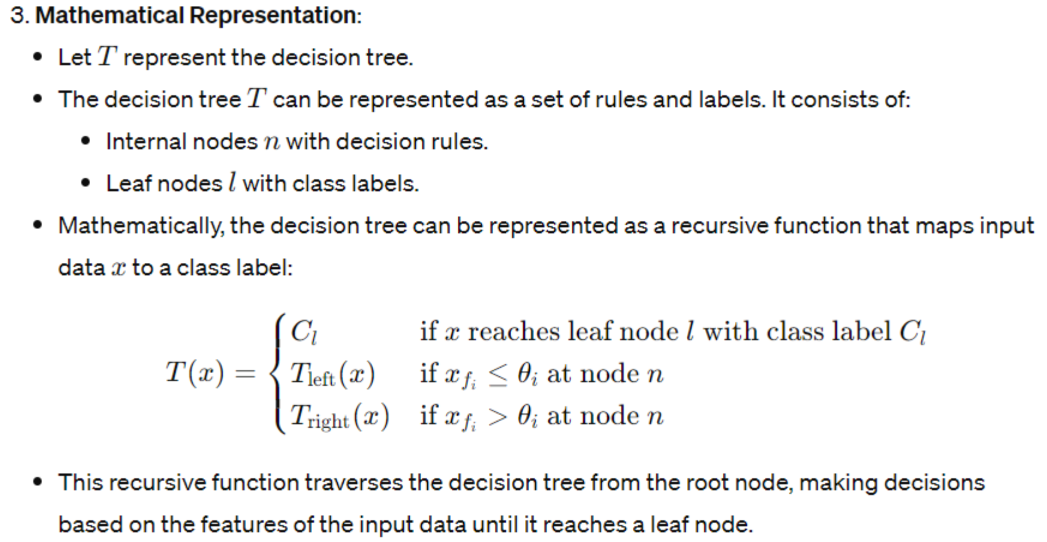


In a mathematical model representation of a decision tree used in a project, we typically represent the structure of the tree, including the decision rules at each node and the class labels assigned to the leaf nodes. Here's how you can represent a decision tree mathematically:

1. **Decision Rules**:
   - Each internal node n in the decision tree represents a decision based on a feature f_i and a threshold value θ_i.
   - Let x be the input data vector representing the features of a sample.
   - The decision rule at node n can be represented as:

$$\begin{gathered}\text{if } x_{f_i} \leq \theta_i \text{ then go to left child node}\\ \text{else go to right child node}\end{gathered}$$

   - Here, x_{f_i} is the value of feature f_i for the input data vector x.
2. **Leaf Node Class Labels**:
   - Each leaf node l in the decision tree is associated with a class label C_l.
   - When a sample reaches a leaf node, it is classified as belonging to the class C_l.



3. **Mathematical Representation**:

- Let T represent the decision tree.
- The decision tree T can be represented as a set of rules and labels. It consists of:
  - Internal nodes n with decision rules.
  - Leaf nodes l with class labels.
- Mathematically, the decision tree can be represented as a recursive function that maps input data x to a class label:

$$T(x) = \begin{cases} C_l & \text{if } x \text{ reaches leaf node } l \text{ with class label } C_l \\ T_{\text{left}}(x) & \text{if } x_{f_i} \leq \theta_i \text{ at node } n \\ T_{\text{right}}(x) & \text{if } x_{f_i} > \theta_i \text{ at node } n \end{cases}$$

- This recursive function traverses the decision tree from the root node, making decisions based on the features of the input data until it reaches a leaf node.

4. Visualization:

While the mathematical representation captures the decision rules and class labels of the decision tree, visualization techniques such as graphical representations or textual representations are often used to better understand and interpret the structure of the tree.

III. FUTURE SCOPE

While our project provides a solid foundation for detecting unsafe conversations on social media platforms, there are several avenues for future exploration and enhancement:

1. Dynamic Adaptation:

Develop mechanisms for dynamic model adaptation and fine-tuning to accommodate evolving patterns of online behavior and emerging trends in digital communication. This includes continuous monitoring of model performance and feedback loops for iterative improvement.

2. User-centric Features:

Integrate user-centric features and feedback mechanisms to personalize the detection system based on individual preferences and sensitivities. This includes user profiling, content filtering, and adjustable thresholds for sensitivity to harmful content.

3. Real-time Intervention:

Implement real-time intervention strategies not only to detect but also to mitigate the impact of unsafe conversations. This may involve automated response mechanisms, content moderation workflows, and community-driven interventions to promote positive interactions and deter harmful behavior.

4. Ethical Considerations:

Address ethical considerations related to algorithmic bias, fairness, and privacy in content moderation practices. Develop transparent and accountable mechanisms for decision-making, model interpretability, and user consent to ensure responsible deployment and usage of the detection system.

5. Collaborative Partnerships:

Foster collaborative partnerships with social media platforms, academic institutions, government agencies, and civil society organizations to share data, expertise, and best practices in combating online toxicity. This includes participation in interdisciplinary research initiatives, policy advocacy, and community engagement efforts.

IV. CONCLUSION

In conclusion, our project has successfully demonstrated the feasibility and effectiveness of leveraging machine learning algorithms within the Python programming language, alongside the Flask framework, to detect unsafe conversations on social media platforms. Through a systematic methodology encompassing data collection, preprocessing, feature engineering, model selection, integration, and deployment, we've developed a robust detection system capable of identifying harmful behaviors in real-time. The significance of this work lies in its potential to enhance the safety and well-being of social media users by proactively addressing the proliferation of toxic interactions such as harassment, cyberbullying, and hate speech. By automating the detection process and providing timely intervention, our system contributes to the creation of safer and more inclusive digital spaces where individuals can engage in meaningful dialogue without fear of intimidation or harm. Looking ahead, continual updates to the training data and model architectures can enhance performance and adaptability to evolving patterns of online behavior. Additionally, collaboration with social media platforms and community stakeholders can facilitate the integration of our detection system into existing moderation frameworks, amplifying its impact and reach across diverse user populations. Our project represents a significant step towards fostering responsible digital citizenship and promoting positive norms of interaction within online communities. By harnessing the power of machine learning and innovative technologies, we aspire to contribute to a future where social media platforms serve as vibrant, inclusive, and safe spaces for all users to connect, communicate, and thrive.

V. REFERENCES

[1] Shiza Ali, Afsaneh Razi, Seunghyun, Ashwaq Alsoubai, Chen Ling, Munmun De Choudhury, Pamela J. Wisniewski, Gianluca, “Getting Meta: A Multimodal Approach for Detecting Unsafe Conversations within Instagram Direct Messages of Youth”, Article in Proceedings of the ACM on Human-Computer Interaction · April 2023.

[2] Steve Hinton, Taissa Gladkova, David Maimon, Olga Babko-Malaya, Rebecca Cathey, “Detection of Hacking Behaviours and Communication Patterns on Social Media”, International Conference on Big Data. 2017 IEEE

[3] Shruti Shinde, Dr. Sunil B. Mane “Malicious Profile Detection on Social Media: A Survey Paper”, 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) 2020.

[4] Koosha Zarie, Reza Farahbakhsh, “Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Content”, ACM International Conference on Advances in Social Network Analysis and Mining. 2020 IEEE

[5] Gergo Hajdu, Yaclaudes Minoso, Rafael Lopez, Miguel Acosta, Abdelrahman Elleithy, “Use of Artificial Neural Networks to Identify Fake Profiles”, Department of Computer Science William Paterson University Wayne, Nj, USA. 2021