Toxic Comment Detection - Project Report

In today's digital world, online platforms have become primary spaces for communication and expression. However, they are also prone to harmful content such as toxic comments, which can negatively impact users and communities. This project focuses on building a binary classification system to detect toxic comments in text, helping moderators and platforms maintain a safe and respectful environment.

Problem Definition

Many existing systems detect only overtly abusive or offensive language, often missing subtle forms of harm. Our goal is to create an NLP-based solution that classifies user-generated text into two categories: Toxic and Non-Toxic. This system uses machine learning to analyze the text and identify harmful comments in real-time or from stored datasets.

System Overview

The system uses Natural Language Processing (NLP) techniques to process and analyze text data. It applies text cleaning, tokenization, and vectorization using the TF-IDF method. A Logistic Regression model is trained on a labeled dataset to classify comments. The trained model and TF-IDF vectorizer are saved for later use, enabling quick predictions without retraining.

Modules

1. Data Preprocessing: Cleaning text by removing punctuation, numbers, and special characters, converting to lowercase, and removing extra spaces. 2. Feature Extraction: Using TF-IDF vectorization to convert text into numerical features. 3. Model Training: Applying Logistic Regression with balanced class weights for better performance on imbalanced data. 4. Evaluation: Using metrics like Precision, Recall, F1-score, and Accuracy to assess model performance. 5. Prediction: Loading the saved model and vectorizer to classify new comments.

Implementation & Results

The dataset used is derived from the Jigsaw Toxic Comment Classification Challenge on Kaggle. For this project, a binary label was used (Toxic or Non-Toxic). The Logistic Regression model with TF-IDF features achieved strong performance, providing reliable classification results. The model and vectorizer are stored as `.pkl` files for deployment.

Conclusion & Future Work

This project demonstrates the effectiveness of combining TF-IDF with Logistic Regression for toxic comment detection. Future enhancements could include: - Multi-class classification for different types of toxicity. - Using deep learning models like BERT for better context understanding. - Adding multilingual support for broader applicability.

References

1. Kaggle: Jigsaw Toxic Comment Classification Challenge. 2. Scikit-learn Documentation - Logistic Regression & TF-IDF Vectorizer. 3. Python Pandas & Joblib Libraries.