# Toxic Comment Classification

## Project Overview

This project focuses on developing a machine learning model that classifies user comments as toxic or non-toxic. The model is intended to help a platform identify and moderate toxic comments to maintain a positive user environment.

The dataset includes comments labeled as toxic or non-toxic, and the goal is to build a model that achieves an F1 score of at least 0.75 to accurately detect toxic comments.

## Data Description

The data is provided in the file `/datasets/toxic_comments.csv` and contains the following columns:

- `text`: The text of the comment.
- `toxic`: A binary label indicating whether the comment is toxic (1) or non-toxic (0).

### Additional Features:

During the preprocessing stage, new features were created, such as the length of the original and cleaned text, which helped improve the analysis and model performance.

## Project Workflow

### 1. Data Loading and Preprocessing

- **Data Cleaning**: Converted text to lowercase, removed special characters, extra spaces, and stop words. Tokenization and lemmatization were applied.
- **Feature Engineering**: Created new features, such as the length of comments before and after cleaning.
- **Data Preparation**: Removed empty or irrelevant data.

### 2. Model Training and Evaluation

- **Vectorization**: Used TF-IDF vectorizer to convert text into numerical format.
- **Balancing**: Addressed class imbalance using oversampling techniques to increase the representation of toxic comments.
- **Models Trained**: Several machine learning models were trained and evaluated, including:
    - Logistic Regression
    - CatBoost

    - Multinomial Naive Bayes
    - Stochastic Gradient Descent (SGD)

### 3. Results

- **Stochastic Gradient Descent (SGD)** performed the best, achieving an F1 score of **0.771402** on the validation set.
- Logistic Regression followed closely with an F1 score of **0.767473**.
- CatBoost had the lowest performance with an F1 score of **0.61143**, suggesting the need for further hyperparameter tuning.

### 4. Final Model Selection

- The **SGD** model was selected for further testing and evaluation on the test set, achieving an F1 score of **0.763**.

## Conclusions

The project successfully identified toxic comments using a machine learning model with an F1 score above the target of 0.75. The Stochastic Gradient Descent (SGD) model is recommended for use in detecting toxic comments and sending them for moderation.

## Requirements

- Python 3.x
- Libraries: `pandas`, `scikit-learn`, `nltk`, `spacy`, `catboost`, `imbalanced-learn`, `matplotlib`, `tqdm`

### How to Run the Project

1. Install the required libraries using:
   bash
   ```
   pip install -r requirements.txt
   ```

2. Download the dataset and place it in the `/datasets` folder.

3. Run the provided notebook or script to preprocess the data, train the models, and evaluate the results.

## Recommendations

Based on the analysis and results, the **Stochastic Gradient Descent (SGD)** model is recommended for toxic comment classification. Further improvements can be made by experimenting with hyperparameter tuning and additional data balancing techniques.