Cyberbullying Detection Challenge – Solution Documentation

By: Ishika Chittora
Email: ishikachittora947@gmail.com

---

1. Model Overview

For the cyberbullying detection challenge, we implemented and evaluated multiple classification models, combining both traditional machine learning and deep learning techniques. The models developed include:

- **Logistic Regression**: A simple linear model used for binary classification.
- **Support Vector Classifier (SVC)**: Effective for high-dimensional data, particularly text.
- **Random Forest Classifier**: An ensemble-based decision tree model.
- **Voting Classifier**: An ensemble of Logistic Regression, SVC, and Random Forest using soft voting.
- **LSTM (Long Short-Term Memory Network)**: A deep learning model capable of learning temporal dependencies in text.
- **XGBoost Classifier**: A high-performance gradient boosting model known for its robustness with imbalanced datasets and structured data.

These models were evaluated on a cleaned, labeled text dataset where the labels indicated whether a given text sample was cyberbullying (1) or not (0).

---

2. Data Preprocessing and Feature Engineering

The dataset was initially noisy and required the following preprocessing steps:

- **Cleaning**: Removed special characters, punctuation, numbers, and extra whitespace.
- **Lowercasing**: Standardized text to lowercase for consistency.
- **Stopword Removal & Lemmatization**: Using NLTK to reduce noise and normalize words.

For feature engineering:

- **TF-IDF Vectorization** was applied for traditional ML models (Logistic, SVC, RF, Voting).
- **Tokenization and Padding** were performed for the LSTM model using Keras' Tokenizer and pad_sequences with a max length of 100.

---

3. Training Methodology

- **Train-Test Split**: All models used an 80/20 train-test split, with stratification to preserve label distribution.
- **Evaluation Metrics**: Accuracy, Precision, Recall, and F1 Score were used to assess performance.

- **Model Training**:
    - Logistic, SVC, XgBoost and RF were trained using scikit-learn with balanced class weights.
    - LSTM was trained using TensorFlow with an embedding layer, LSTM layer, and two dense layers.
    - VotingClassifier combined the predictions from Logistic, SVC, and RF for improved robustness.

---

4. Model Performance Summary

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	~84.4%	~38.9%	~41.2%	~40.0%
SVC	~86%	~41.6%	~29.4%	~34.4%
Random Forest	~88.1%	~55.5%	~29.4%	~38.4%
Voting Classifier	~87.4	~50.0%	~11.7%	~19.0%
LSTM	~87.4	~0	~0	~0
XgBoost	~85.9	~40.0%	~23.5	~29.6%

Analysis:

- **Logistic Regression** showed balanced metrics and was easy to interpret, but performance was limited due to its linear nature.
- **SVC** slightly improved accuracy and precision over logistic regression, but recall was still low, showing it missed many positive (bullying) cases.
- **Random Forest** achieved the highest accuracy and precision but still suffered from low recall. This suggests it was very selective, capturing fewer but more certain bullying cases.
- **Voting Classifier** surprisingly showed a sharp drop in recall and F1 score despite its accuracy. This is likely due to ensemble members (especially RF) dominating the voting process and being too conservative in detecting positives.
- **LSTM** severely underperformed with 0 precision, recall, and F1. This may indicate overfitting, a broken data pipeline (such as all zeros in labels), or ineffective training.

- **XGBoost** offered competitive balance, improving recall over SVC and Random Forest, making it a strong candidate in handling class imbalance better than most others.

---

5. Insights and Future Work

Key Insights:

- Class imbalance significantly affects traditional models, causing them to bias toward majority class.
- TF-IDF is a reliable feature representation for simple models, but deep learning offers richer contextual understanding.
- Ensemble methods (Voting) can help overcome individual model weaknesses.