Toxic Comment Classification

AAI501 - Applied Artificial Intelligence

San Diego University
Instructor: Azka Azka

Team 7 Members:

- Anugrah Rastogi
- Dhrub Satyam
- Mallesham D

Team Members & Their Contributions

Anugrah Rastogi

- Data Preparation
- EDA
- Plot and Analysis

Dhrub Satyam

- Modeling & Deployment

Mallesham D

- Overall analysis
- Conclusions and Report

Abstract

- **Aim:**
  - Detect and classify harmful online comments into multiple categories.
- **Data:**
  - Jigsaw Toxic Comment Classification dataset from Wikipedia talk pages.
- **Methods:**
  - Data cleaning, preprocessing, TF-IDF vectorization, and classification using machine learning and deep learning models.
- **Models:**
  - Multinomial Naive Bayes, Logistic Regression, and Deep Learning.
- **Outcome:**
  - Achieved strong performance on frequent classes; rare classes need further work.
- **Suggestion:**
  - Address class imbalance, explore transformer-based models.

Introduction

Problem

Toxic language online can harm communities and individuals.

Objective

Build a classifier to detect multiple categories of toxic comments.

Dataset

Jigsaw Challenge – publicly available, labeled multi-label data.

Goal

Accurate, interpretable, and scalable detection system.

Real-World Motivation

Why This Matters:

- Toxic comments in online spaces harm mental well-being and disrupt constructive dialogue.
- Platforms like Wikipedia, Reddit, and YouTube face constant moderation challenges.
- Manual review of large volumes of user-generated content is slow, costly, and inconsistent.

Real Impact Examples:

- Social Media Moderation: Preventing harassment, cyberbullying, and hate speech.
- Online Communities: Maintaining respectful collaboration in forums and knowledge bases.
- Legal & Compliance: Helping organizations meet content moderation policies and regulatory standards.

Need for Automation:

- Rapid detection ensures harmful content is flagged before it causes damage.
- Scalable AI-driven moderation supports global platforms with millions of daily interactions.

Dataset Overview

- Total Comments: 159,571
- Labels: toxic, severe_toxic, obscene, threat, insult, identity_hate
- Additional: ‘clean’ label for non-toxic comments
- Nature: Multi-label classification (one comment can have multiple labels)

Data Preparation

- **Missing Values:** Removed empty comments
- **Lowercasing:** Standardized text to lowercase
- **Special Character Removal:** Removed punctuation, numbers, symbols
- **Tokenization:** Split into words
- **Stopword Removal:** Filtered common, low-value words
- **Lemmatization:** Reduced words to base form

Exploratory Data Analysis

- **Class distribution:** Majority clean comments (~90%)
- **Rare labels:** threat, identity_hate (<0.5%)
- **Co-occurrence:** toxic often with obscene/insult
- **Visuals:** Label frequency plots, word clouds

Model Selection

TF-IDF + Logistic Regression: Baseline, interpretable

Multinomial Naive Bayes: Efficient, strong on frequent classes

Deep Learning: Captures complex patterns, needs more for rare labels

Metric Focus: F1-score, Precision, Recall, AUC-ROC

Classification Metrics

Precision

- Of predicted defaulters, how many were correct?

Recall

- How many actual defaulters were identified?

F1 Score

- Balance between Precision & Recall.

Results

Label	Precision	Recall	F1-score
Toxic	0.90	0.62	0.73
Severe_Toxic	0.55	0.21	0.31
Obscene	0.91	0.64	0.75
Threat	0.50	0.09	0.16
Insult	0..81	0.51	0.63
Identity_Hate	0.80	0.16	0.27

Multinomial Naive Bayes

Goal: Classify text into multiple toxicity categories.

Data Representation: TF-IDF vectors (min df=3, n-gram range: 1–3).

Training Setup: 80% train, 20% test; separate binary classifiers for each label.

Strengths:

- Fast and computationally efficient.
-
- Performs well with word frequency-based features.

MNB: Results & Observations

- **Frequent classes (toxic, obscene):** Good F1-scores.
- **Rare classes (threat, identity_hate, severe_toxic):** Lower recall despite good AUC.
- **Tuning:** Best smoothing parameter α=0.01.
- **Macro F1-score:** ~0.448
- **Average AUC:** ~0.944
- **Takeaway:** Effective for high-level filtering but limited in rare class detection.

Deep Learning Approach

- **Goal:** Capture complex patterns in toxic language.
- **Architecture:** Sequential deep learning model with embedding & dense layers.
- **Training:** Used same preprocessed TF-IDF/embedding data split as MNB.
- **Metrics:** Evaluated per-label precision, recall, F1-score.

Deep Learning: Results & Insights

Strengths: Better adaptability to feature complexity than MNB.

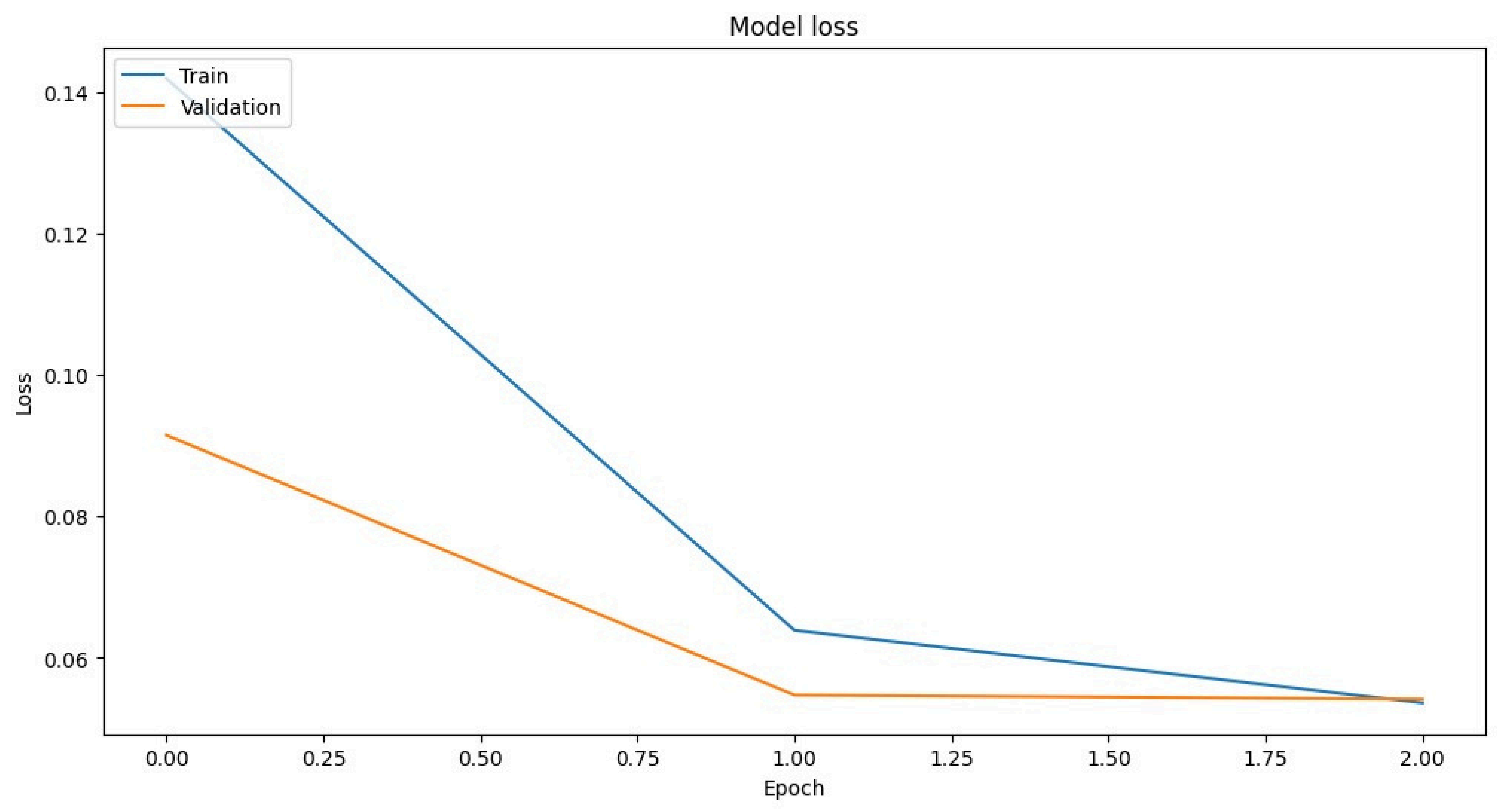
Challenges:

- Minority classes still underperform due to imbalance.
- Precision high for common classes, recall low for rare ones.

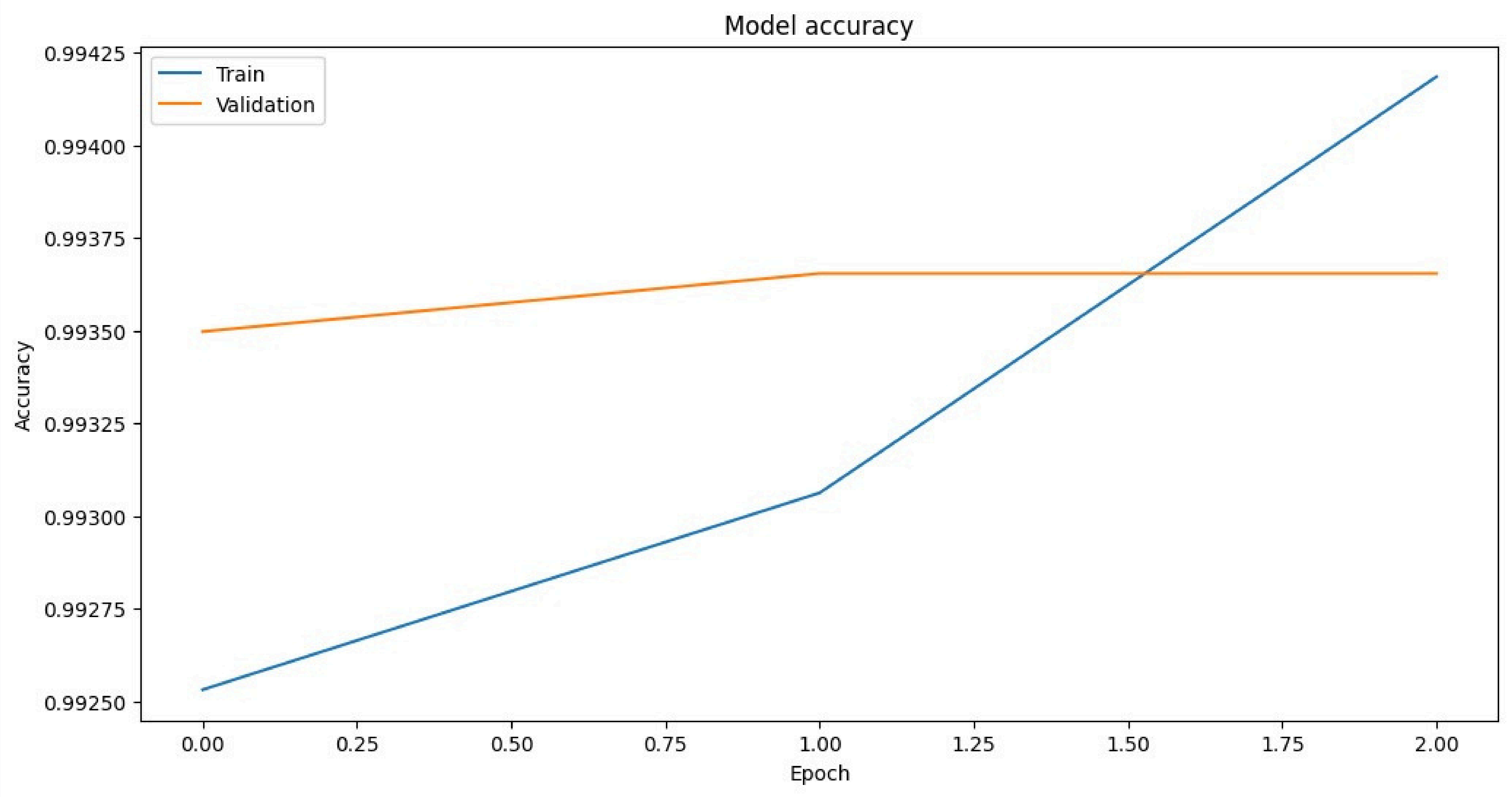
Training Behavior: Quick convergence; risk of overfitting.

Future Improvements: Use class imbalance strategies (focal loss, re-weighting).

Deep Learning: Results & Insights



Deep Learning: Results & Insights



Insights

- Frequent labels achieve higher recall & precision
- Rare labels suffer from low recall despite good AUC
- Logistic Regression + TF-IDF works well for baseline
- Naive Bayes provides quick, interpretable results
- Deep Learning has potential but needs imbalance handling

Conclusion

- Multi-model approach effective for frequent classes
- Rare class prediction remains challenging

Future Work

- Apply class weights, SMOTE, or data augmentation
- Explore BERT/transformer architectures
- Extend to multilingual datasets
- Develop explainable AI tools for moderator trust