Report: Toxic Comment Classification

Anastasia Vysotskaya

1 Project Goal and Code Structure

This project is an NLP text classification task based on the Wikipedia Comments dataset. The objective is to predict whether a user comment is toxic or non-toxic. It applies traditional machine learning models on TF-IDF and linguistic features to detect toxicity in the text.

My personal goal for this project was to build a clean and reproducible machine learning pipeline for toxic comment classification. The main focus was repository structure and reproducibility, not optimisation of model performance.

Code organisation:

- `src/preprocess.py` – CLI entry point; builds cached features.
- `src/features.py` – text normalisation, lemmatisation, TF–IDF, meta features.
- `src/caching.py` – caching and loading of feature matrices, manifests.
- `src/train.py` – trains classifiers, performs BayesSearchCV if enabled, evaluates on validation set.
- `src/evaluate.py` – evaluates on test set, saves metrics and plots.
- `src/dict_baseline.py` – dictionary-based model using an external toxic lexicon.
- `src/interpret.py` – saves small sets of false positives/negatives.
- `scripts/run.sh` / `run.bat` – full pipeline execution.

Outputs are stored under `outputs/<model>/` (metrics, figures, errors) and models are saved under `models/<model>/`.

2 Preprocessing

Steps applied:

- Text normalisation, lemmatisation, TF–IDF vectorisation.
- Addition of meta features (caps ratio, punctuation counts, profanity ratio, etc.).
- Feature selection with `SelectKBest` → 150k features.

Skipped step: stemming. Lemmatisation provides cleaner, dictionary-based forms and avoids noise. Stopwords removal was not implemented, `SelectKBest` makes sure noisy words do not make it into the features.

3 Feature Engineering

Representations:

- Word n-grams (1-2) with TF–IDF.
- Character n-grams (3-4) with TF–IDF.

Character n-grams are especially useful in toxicity detection. They capture subword patterns, obfuscations, and unusual spellings, which word-level features often miss.

4 Modelling

Models trained:

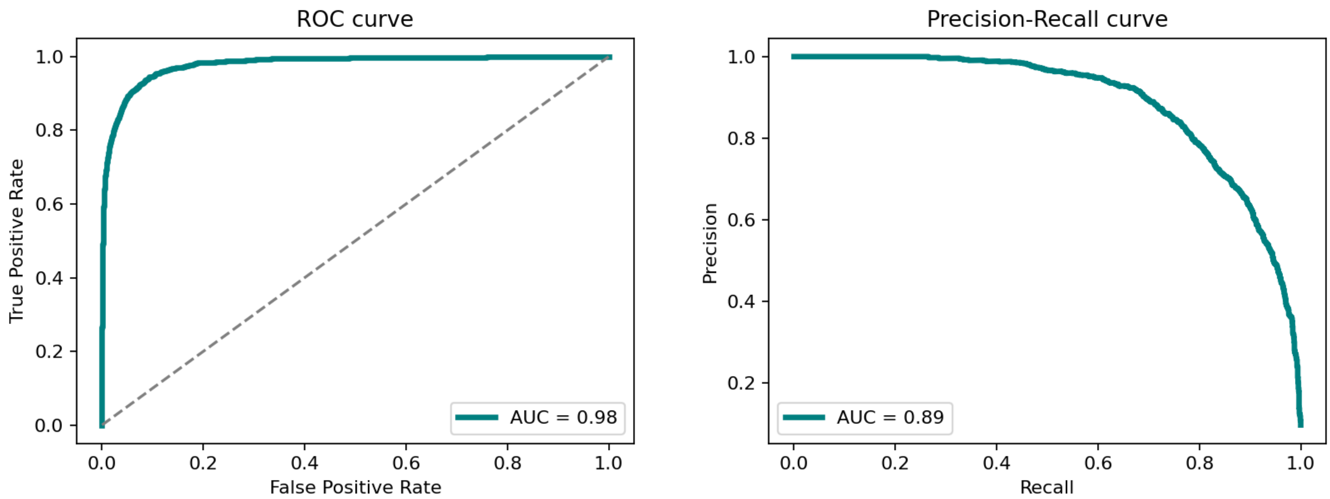
- Logistic Regression (default for linear text classification).
- Linear SVM (maximises margin for better separation of noisy classes).
- Passive Aggressive Classifier (chosen for speed and suitability for large-scale linear text problems).

Dictionary approach was selected as a baseline. It predicts toxic if any word from an external profanity list is found. Hyperparameters were fixed after BayesSearchCV for Logistic Regression and Passive Aggressive Classifier; Linear SVM used defaults due to cost and time constraints.

5 Evaluation

Model	Precision(1)	Recall(1)	macro F1	ROC–AUC	PR–AUC	Hyperparameters
Baseline	0.3383	0.7312	**0.6833**	0.7898	**0.2731**	profanity lexicon
LogReg	0.8596	0.7371	**0.8868**	0.9786	**0.8872**	C=4.7, penalty=l1, class_weight=None, tol=2.7×10^{-6}
Linear SVM	0.8851	0.7103	**0.8840**	0.9767	**0.8868**	defaults
PAC	0.8761	0.7122	**0.8827**	0.9752	**0.8835**	C=0.04, loss=hinge, average=50, tol=2.8×10^{-5}

I consider macro F1 and PR-AUC the most meaningful metrics under class imbalance. ROC and PR curves for Logistic Regression are presented on Figure 1.



(a) ROC curve

(b) Precision-Recall curve

Figure 1: Logistic Regression metrics on the test set.

6 Reflection

- **Metric choice:** Accuracy and ROC-AUC are dominated by the majority non-toxic class. F1 and PR-AUC directly measure the trade-off between precision and recall for the minority toxic class.
- **Error consequences:**
  - False positives: benign users flagged; harms commenters, increases moderator load.
  - False negatives: toxic content slips through; harms readers, damages community trust.
- **Context:** Decision threshold can be adjusted:
  - Lower → more recall, useful for automated pre-filtering.
  - Higher → more precision, suitable for human moderator support.

  This trade-off can be illustrated with a PR curve (Figure 1b).
- **COVID Analogy:** Like COVID testing, trade-off depends on use case. Screening tests favour recall; confirmatory moderation tools favour precision.