# Lyrics Only Genre Classification
# TF-IDF Baseline vs MiniLM Embeddings, with Diagnostics

Hill Zhang

**Abstract**

This project predicts a song's genre using lyrics text only, with no audio features. I built a clean, reproducible text classification pipeline and compared three lyric representations under one fixed classifier, multinomial logistic regression: (1) TF-IDF bag of words, (2) MiniLM sentence embeddings, and (3) MiniLM embeddings compressed through a small autoencoder. The key result is that TF-IDF remains a strong baseline under macro-F1. The diagnostic plots explain why performance plateaus across representations and why minority genres are hardest.

## Project summary

- **Goal:** predict one of five genres (Folk, Jazz, Metal, Pop, Rock) from lyrics text only.
- **Pipeline:** cleaning, stratified split, feature extraction, model training, validation tuning, held out test evaluation.
- **Best model (macro-F1):** TF-IDF (20k vocabulary) + logistic regression achieved **0.6096** test accuracy and **0.4732** test macro-F1.
- **Main takeaway:** MiniLM matched accuracy (0.6092) but dropped macro-F1 (0.4426), and compression dropped further (0.5950, 0.3970).
- **Diagnostics:** confusion matrix and PCA show heavy overlap, Rock absorbs mistakes, which explains the macro-F1 gap.

## 1 Problem

Given a song's lyrics, predict one of five genres: Folk, Jazz, Metal, Pop, or Rock. The point is to test how much genre signal exists in lyrics alone, and whether modern embeddings beat a strong TF-IDF baseline under a fixed evaluation setup.

## 2 Data and preprocessing

I started from the Kaggle *Multi-Lingual Lyrics for Genre Classification* dataset (used locally, not included in the repository). I filtered to five target genres, removed rows with missing lyrics or labels, then downsampled to 50,000 songs for fast iteration while preserving genre proportions.

## 3 Model and representations

### 3.1 Classifier (constant across runs)

I used multinomial logistic regression for all representations. I tuned the regularization strength $C \in \{0.1, 1, 10\}$ on the validation set using macro-F1, then retrained on train plus validation with the chosen $C$ and reported test metrics.

Table 1: Train, validation, test split (fixed across all experiments).

| Split | Purpose | Size | Notes |
|---|---|---|---|
| Train | Fit models | 30,000 | Used to learn parameters |
| Validation | Tune regularization $C$ | 10,000 | Selected by macro-F1 |
| Test | Final evaluation | 10,000 | Held out, reported once |

### 3.2 Representations (what changes)

- **TF-IDF (20k):** sparse bag of words features, capped at a 20,000 word vocabulary.
- **MiniLM (384d):** dense sentence transformer embedding per song (all-MiniLM-L6-v2), then standardized features.
- **MiniLM + autoencoder (64d):** compress 384d embeddings to a 64d latent space, then classify using the latent vector.

Table 2: Autoencoder architecture used to compress MiniLM embeddings.

| Layer | Input dim | Output dim |
|---|---|---|
| Encoder Linear 1 | 384 | 128 |
| Encoder Linear 2 | 128 | 64 (latent) |
| Decoder Linear 1 | 64 | 128 |
| Decoder Linear 2 | 128 | 384 |

## 4 Results

Table 3 reports held out test performance. Accuracy is similar for TF-IDF and MiniLM, but macro-F1 is higher for TF-IDF, which indicates better performance on smaller genres. The autoencoder bottleneck drops further, consistent with information loss from compression.

Table 3: Held out test performance (higher is better).

| Representation | Test accuracy | Test macro-F1 |
|---|---|---|
| TF-IDF (20k) | 0.6096 | 0.4732 |
| MiniLM (384d) | 0.6092 | 0.4426 |
| MiniLM + AE (64d) | 0.5950 | 0.3970 |

## 5 Diagnostics (what the plots explain)

### 5.1 Confusion matrix: why macro-F1 drops

Figure 1 is the normalized confusion matrix for the best macro-F1 model (TF-IDF + logistic regression). Three patterns matter:

- **Rock is the magnet class:** many errors from other genres land in Rock.

- **Pop vs Rock is the most intuitive confusion:** the boundary is fuzzy for lyrics only models.
- **Folk is hardest:** its correct rate is low and a large fraction is predicted as Rock, which pulls down macro-F1 even when accuracy looks acceptable.

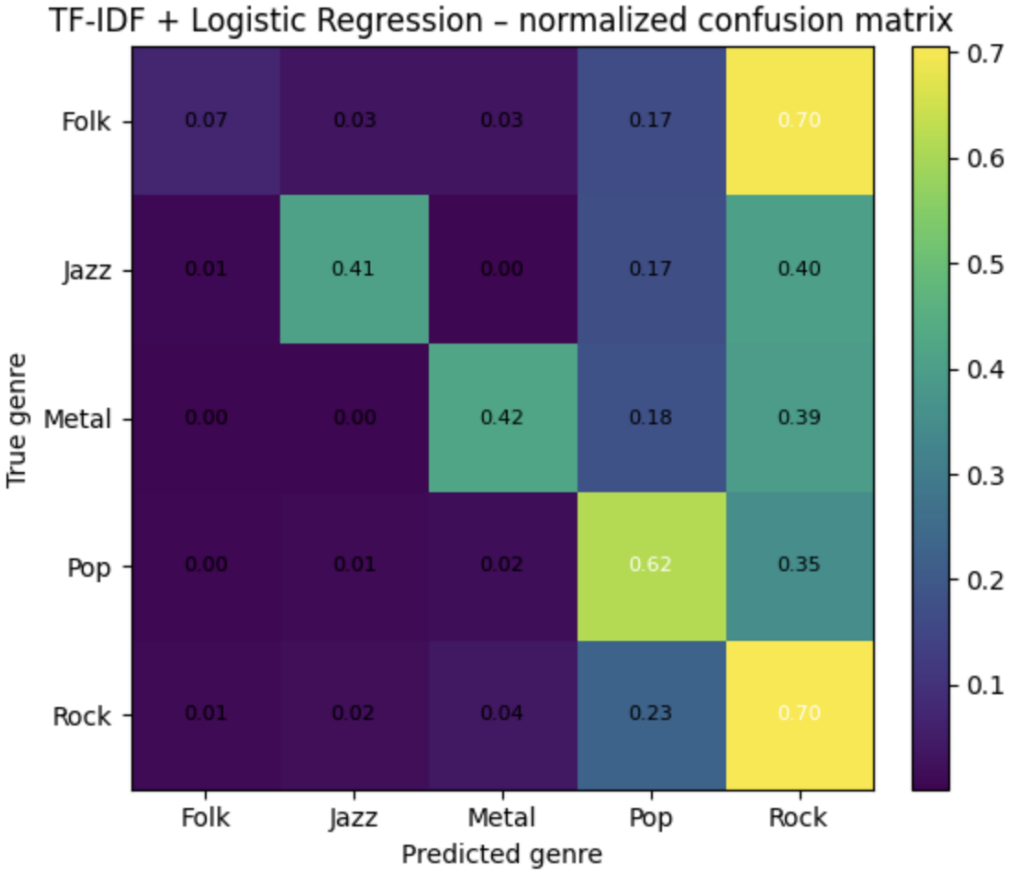

Figure 1: Normalized confusion matrix for TF-IDF + logistic regression (rows sum to 1).

### 5.2 PCA of MiniLM embeddings: why separation is limited

Figure 2 is a PCA projection of the 384 dimensional MiniLM embeddings. The point is not that PCA proves the task is impossible. It is a sanity check: if a strong pretrained embedding space already looks heavily mixed in a simple 2D view, it makes sense that a linear classifier will not get clean separation. This matches the observed plateau across representations, especially for Pop and Rock.

### 5.3 Interpretability check: top TF-IDF tokens

Figure 3 shows the words with the highest positive logistic regression weights for the Folk class under TF-IDF. These are not the most frequent words, they are the most discriminative relative to other genres. This makes the TF-IDF baseline debuggable, and it helps explain why TF-IDF can compete with, and even beat, embeddings on this task.

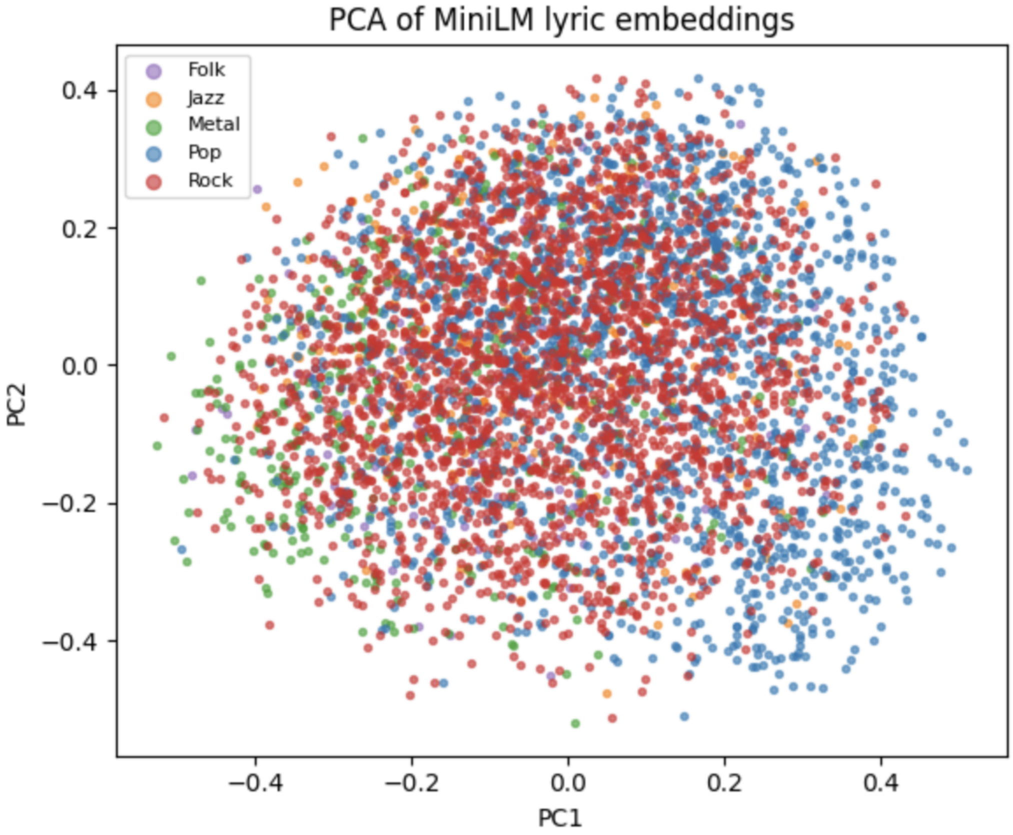

Figure 2: PCA projection of MiniLM lyric embeddings (PC1 vs PC2). Colors overlap heavily.

## 6 Engineering and reproducibility

- **Fixed split discipline:** the same train, validation, test split is used across all representations for a fair comparison.
- **Metric choice:** macro-F1 is emphasized because the dataset is imbalanced, accuracy can look fine even if small genres perform poorly.
- **Reproducible run:** the notebook runs cleanly under Restart Kernel plus Run All with no hidden state dependence.

**To reproduce locally (suggested):**

- Place the Kaggle CSV at `data/lyrics_train.csv` (not tracked in git).
- Run `lyrics_genre_project.ipynb` to reproduce preprocessing, feature extraction, tuning, and evaluation.
- Core dependencies: pandas, numpy, scikit-learn, matplotlib, sentence-transformers, torch.

## 7 Limitations and next steps

- **Lyrics only ambiguity:** some genres are inherently hard to separate without audio features.

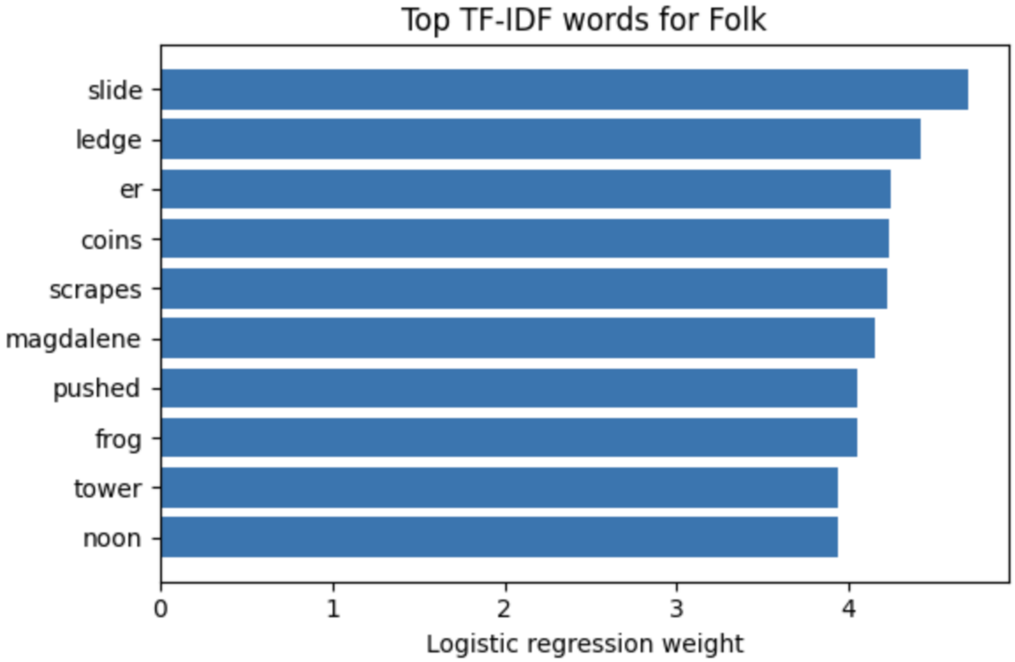

Figure 3: Example interpretability check: top positive logistic regression weights for the Folk class under TF-IDF.

- **Structured confusions:** Rock absorbs mistakes, future work could try class weighting, calibration checks per class, or adding n grams.
- **Embedding direction:** try finetuning, alternative pooling, or nonlinear classifiers, while keeping the same discipline of fixed splits and macro-F1 reporting.

**Data note:** this repository should not include the raw Kaggle dataset due to size and licensing constraints. The report and figures are derived outputs.