Reflection: Lessons Learned from the Deezer Skip Prediction Project

Vimerlin Govender[1], Meleknur Özgü[1] and Xinmeng Song[1]

[1]*Lucerne University of Applied Sciences and Arts (HSLU)*

Scientific Knowledge

This project deepened our understanding of how recommender systems operate in practice versus theory. In coursework, collaborative filtering and matrix factorisation are presented as the core techniques, but we discovered that for the skip prediction task—framed as binary classification—gradient-boosted trees with carefully engineered features significantly outperformed traditional recommendation approaches. Our item-based collaborative filtering baseline achieved a modest Hit@10 rate, while XGBoost with 47 features reached 0.9341 AUC on the same data.

We also learned the importance of distinguishing between *offline evaluation metrics* and *real-world recommendation quality*. AUC measures discriminative ability on a fixed test set, but tells us nothing about diversity, novelty, or long-term user satisfaction. Writing the production system section of our report forced us to engage with this gap concretely, rather than treating it as an abstract limitation.

Technical Skills

The project provided substantial hands-on experience with the full machine learning pipeline:

- **Feature engineering at scale**: Working with 7.5 million interactions taught us that feature computation must be designed for efficiency. Target encoding with smoothing, log-transformed counts, and leakage-free user statistics required careful implementation to avoid subtle data leakage bugs.
- **Model training and comparison**: We gained practical experience with XGBoost, LightGBM, and PyTorch, learning their respective strengths. The finding that tree models outperform neural networks on tabular data—while theoretically known—was instructive to observe first-hand.
- **Reproducibility and collaboration**: Maintaining clean code, version control with Git, and reproducible pipelines across three team members taught us that software engineering discipline is as important as modelling skill in data science projects.

Practical Recommender System Insights

Three insights stand out. First, **user–item interaction features dominate**: the most important features were not content-based metadata (genre, duration) but behavioural signals (user–artist affinity, smoothed listen rates). This suggests that collaborative signals, even when encoded as tabular features, capture more predictive information than content alone.

Second, **simple ensembles hit diminishing returns quickly**: our XGBoost–LightGBM ensemble matched but did not exceed the best individual model, indicating high prediction correlation. In hindsight, diversity in model families (e.g., adding a sequence-based model) would likely yield more ensemble benefit than combining two tree-based methods.

Recommender Systems Course Project, February 2026
author1@university.edu (A. One); author2@university.edu (A. Two); author3@university.edu (A. Three)

Third, **the gap between competition and production is larger than expected**. Our competition pipeline assumes a fixed catalogue, known users, and batch prediction. A real system must handle millions of new tracks weekly, users with no history, and sub-100ms latency constraints. This project gave us a concrete appreciation for why production recommender systems are complex distributed systems, not just well-tuned classifiers.