Music Minimal Skip Optimizer

STAT 4830 Slides Draft

Problem Statement

Skipping Songs

Skip behavior is core engagement sign for music platforms.

Skips usually mean:

- Shorter listening sessions
- Lower satisfaction
- Reduced retention

Objective

Given a fixed set of tracks, find an ordering that minimizes expected skip events

Minimize total predicted skip probability across playlist positions

Impact

Ordering is a real product decision (Daily Mix, Radio, Discover Weekly)

Constraints and Risks

Constraints

- No online A/B testing
- No counterfactual user exposure
- Historical skips logged under original orderings
- Combinatorial optimization over permutations

Risks

- Position bias may dominate preference
- Skip probabilities may be miscalibrated
- Offline metrics may overestimate real-world gains

Dataset

Spotify Sequential Skip Prediction (WSDM Cup 2018)

What it contains

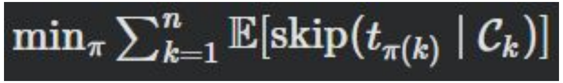
- Ordered track sequences within sessions
- Skip labels / listening duration
- Track metadata (genre, artist)
- Session boundaries

Why it’s suitable

- Designed for sequential skip modeling
- Supports offline evaluation of reordering policies

Mathematical Formulation

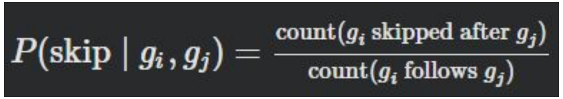
Let a playlist consist of tracks {t1,…,tn} . We seek a permutation π that minimizes expected skip events:



$$\min_{\pi} \sum_{k=1}^{n} \mathbb{E}[\text{skip}(t_{\pi(k)} \mid C_k)]$$

where C_k denotes session context, including previous tracks and position.

We begin with a frequency-based probabilistic baseline, motivated by interpretability and robustness:



$$P(\text{skip} \mid g_i, g_j) = \frac{\text{count}(g_i \text{ skipped after } g_j)}{\text{count}(g_i \text{ follows } g_j)}$$

where *gi* is the genre of the current track and g_j is the genre of the immediately preceding track.

Overview: Two-Model Pipeline

Skip Prediction Model

Estimate the probability that a track will be skipped given its context.

- Input: track features + session context
- Output: P(skip | track, context) for each candidate track
- Current implementation: Empirical genre-transition probabilities
- Planned extension: Logistic regression with position bias and richer context features

Playlist Reordering Model

Use predicted skip probabilities to reorder tracks to minimize expected skips.

- Input: Set of tracks + predicted skip probabilities
- Output: Minimize expected total skips across playlist positions
- Current strategy: Greedy sort by ascending predicted skip probability

Playlist Reordering Strategy

Greedy ordering:

Predict skip probability for each track

Sort tracks by ascending predicted skip probability

Compare reordered playlist to original ordering

Advantages: Simple, fast, strong baseline

Next Steps:

Transition-aware costs

Frank–Wolfe relaxation over permutations

Evaluation and Validation

Skip prediction: AUC, Log loss

Playlist ordering: NDCG, Skip@k, Reduction in predicted skip probability vs original ordering

Important note

- All results are offline and non-causal
- Standard practice in recommender systems
- Limitations explicitly documented

Limitation Observed

No position bias correction yet

No user personalization

Reordering ignores transition smoothness

Sparse statistics for rare genres

Noisy skip signals for short tracks

Next Steps

Immediate:

Add explicit position bias features

Implement logistic regression skip predictor in PyTorch

Evaluate probability calibration

Longer-term:

Transition-aware ordering

Pairwise ranking losses

Session-level survival modeling

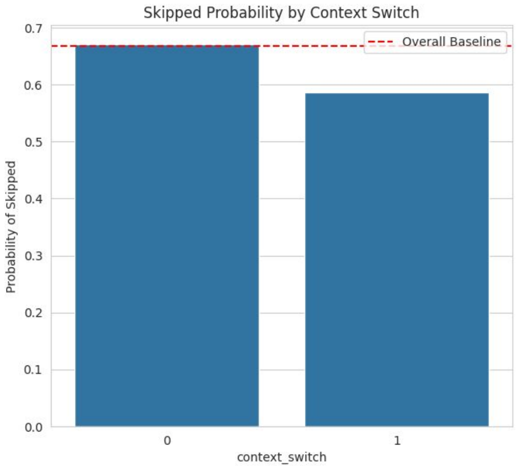
Key takeaway:

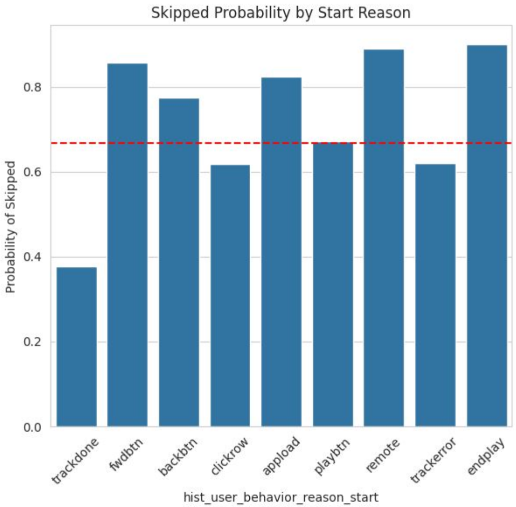
Simple baselines are strong

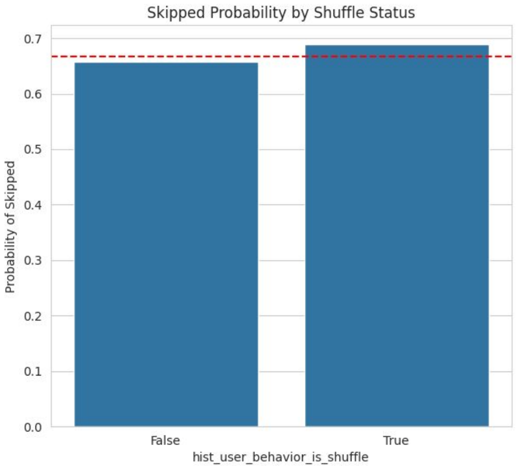
Validation design matters as much as model complexity

Separating prediction from optimization clarifies the system

Baseline Skip Prediction Model







Context Switch

- Skip rate lower with context switch (~34%)
- Skip rate higher without switch (~52%)

Interpretation: switching playlists often signals intentional listening

Start Reason (Strongest Signal)

High skip likelihood

- fwdbtn: ~76%
- backbtn: ~69% → User actively searching / skimming

Low skip likelihood

- trackdone: ~16% → Passive, continuous listening

Shuffle Mode

- Shuffle ON: ~55% skip rate
- Shuffle OFF: ~50% skip rate

Small but consistent increase in skipping under shuffle