PROJECT ABSTRACT

Music Recommendation using Spotify Data

Playlist-continuation recommender trained on the Spotify Million Playlist Dataset (1M playlists, 2M+ tracks), framed on the RecSys Challenge 2018 task, delivered as a built-to-order student project with notebook, demo app, report, PPT and viva kit.

01. Title

Music Recommendation using Spotify Data

02. Introduction / Background

Playlists are how most people organize music, and 'what should I add next?' is a recommendation problem Spotify turned into the RecSys Challenge 2018 by releasing the Million Playlist Dataset: 1,000,000 user playlists, 2M+ unique tracks, ~300K artists. The challenge task — automatic playlist continuation — asks for up to 500 tracks that extend a partial playlist, evaluated with R-precision, NDCG and the clicks metric across 10 scenarios (title-only, first track, first 5, and more). This project rebuilds that task at student scale: track embeddings from playlist co-occurrence blended with audio-feature similarity and title matching.

03. Problem Statement

Students hear about the famous Spotify challenge but the full 1M-playlist corpus is too heavy to download and train on casually, and most tutorials stop at toy collaborative filtering that ignores the playlist-continuation framing entirely. The project delivers a faithful, runnable slice: the real MPD schema, the real 10-scenario evaluation protocol, the real challenge metrics — on a documented subset — plus a playlist-builder demo that shows why each continued track was chosen.

04. Objectives

- Implement the playlist-continuation pipeline on a documented MPD subset: track embeddings from playlist co-occurrence (Word2Vec-style), audio-feature similarity, and playlist-title matching.
- Evaluate on the 10 RecSys 2018 continuation scenarios with R-precision, NDCG and recommended-songs clicks (design targets ~0.18 / ~0.35 / ~1.9; final numbers from the build).
- Build an interactive demo: seed-track playlist builder, ranked continuations with reason chips, audio-feature explorer and evaluation dashboard.
- Deliver the complete student kit: source code, notebook, report, PPT and viva Q&A.

05. Proposed Methodology

The pipeline has three stages. (1) Data: the MPD JSON slices are parsed into playlist-track pairs; a documented subset (fixed sampling seed) keeps training feasible on a student PC. (2) Model: track embeddings are learned from co-occurrence with a Word2Vec-style objective; at inference the continuation score blends three signals — embedding/co-occurrence similarity to seed tracks (0.5), audio-feature similarity on danceability/energy/valence (0.3), and playlist-title keyword matching (0.2). (3) Evaluation: each test playlist is truncated per the 10 challenge scenarios, 500 tracks are recommended, and R-precision, NDCG and clicks are computed exactly as in the challenge.

06. System Architecture / Working

The user adds seed tracks to a playlist, names it, and presses 'Continue playlist'. The backend scores every catalog track with the three-signal blend, ranks them, and renders the top 5 with score bars and reason chips ('co-occurs in N playlists', 'same vibe: Synthwave', 'fits title'). The track explorer shows any track's audio-feature profile. The evaluation view documents the 10 scenarios, the three challenge metrics with design targets, and the dataset's honest limitations.

07. Hardware / Software Requirements

- Hardware: any 64-bit PC with 8 GB+ RAM; no GPU needed (subset-scale training).
- Python 3.10, gensim (track embeddings), pandas, NumPy, scikit-learn.
- Matplotlib (metric and scenario charts).
- Flask demo web app (playlist builder, track explorer, evaluation views).
- Spotify Million Playlist Dataset subset (documented sampling; full 1M optional).

08. Expected Outcomes

- A playlist-continuation recommender evaluated on the authentic 10-scenario protocol with challenge metrics logged by the notebook.
- A documented subset pipeline any examiner can re-run, with the sampling seed fixed.
- An interactive demo where every continued track carries an explanation.
- Report, PPT and viva Q&A covering playlist continuation, the three challenge metrics, and the MPD's sampling caveats.

09. Applications

- Playlist-builder features for student music apps.
- Teaching reference for the RecSys 2018 challenge setup and its metrics.
- Template for sequential/session-based recommendation coursework.

- Baseline for experimenting with transformer-based playlist models.

10. Future Scope

- Transformer playlist model (BERT4Rec-style) trained on full track sequences.
- Audio-content embeddings from raw waveforms for cold tracks.
- Playlist-title generation (the inverse task) with a language model.
- Full-corpus training on cloud infrastructure with the complete MPD.