Final Project

Music Recommendation Engine

Northeastern University: College of Professional Studies

EAI6010: Applications of Artificial Intelligence

Professor: Sergiy Shevchenko

Team: Husky 1

Ayush Anand

Anjal Mohammed

Mohit

Saurav Dibakar

Shashikant Nallan Chakravartula

Sangayya Hiremath

June 24th, 2025

Why This Technology?

Music platforms normally rely on collaborative filtering that needs listening history. In "cold-start" situations such as brand-new users, newly released songs, or privacy-sensitive settings, that data is missing. A **content-based k-Nearest Neighbors (k-NN)** model built only on Spotify audio features (danceability, energy, tempo, valence, popularity, and others) fills the gap. Cosine distance in a 14-dimensional feature space gives an intuitive measure of sonic similarity while needing zero prior user data and just minutes to set up.

Benefits

- **Cold-start friendly** – works instantly for new songs or users.
- **Transparent and interpretable** – similarity is an explicit function of familiar audio attributes.
- **Fast prototype** – scikit-learn k-NN plus FastAPI returns results in under 100 ms for roughly 232,000 tracks.
- **Reproducible pipeline** – StandardScaler and saved model artifacts let classmates replicate results easily.

Drawbacks and Challenges

- **No personalization** – every user receives the same list for the same seed tracks.
- **Scalability** – exact k-NN slows once the catalog grows into millions; approximate ANN libraries such as FAISS or Annoy are planned for future work.
- **Feature bias** – adding popularity can push results toward mainstream hits even if they are less sonically similar.

Working Example (Demo)

1. **Endpoint**: `POST /recommendations` on a live FastAPI server (QR link in slides).
2. **Input**: JSON body containing one or more Spotify `track_id` values and the desired `top_n`.
3. **Process**: map IDs to scaled feature vectors, perform cosine-distance search, convert to similarity = 100 × (1 − distance).
4. **Output**: JSON array of the top-N similar songs, each with title, artist, genre, and percent similarity. A companion `GET /songs?page&limit` endpoint lets users browse the full catalog.
5. Github Repository: https://github.com/AnjalMohammed/MusicRecommendationEngine