SENG474 Progress Report: Hybrid Music Recommendation System

Yule Wang, Weiting Ye, Daehwan (David) Kim, Adithyakrishna Arunkumar, Abhay Cheruthottathil

March 10, 2025

1. The Problem

Problem Description:
Our project is focused on developing a hybrid music recommendation system that generates ranked song lists for a given input—whether it is a single track or an entire playlist. We are merging two key datasets to create a unified, normalized dataset. This dataset combines the audio features from the Spotify Million Playlist Dataset (including track ID, albumID, danceability, energy, key, loudness, mode, speechness, acousticness, instrumentalness, liveness, valence, tempo, etc.) with the popularity metric (the frequency count of each track) from the Spotify 1.2M Songs Dataset.

Changes from the Formal Proposal:

- We now directly integrate the popularity metric into our training process.
- We have decided to use a masking approach for collaborative filtering on the playlist dataset.
- Our focus has shifted toward a single unified model that leverages both intrinsic audio features and playlist popularity to produce ranked recommendations.

2. Goals

Primary Objectives:

- **Model Performance:** We aim to produce a ranked list in which highly relevant songs appear at the top. We will quantify success using metrics such as Mean Average Precision (MAP), precision@k, and Normalized Discounted Cumulative Gain (NDCG).
- **Validation via Masking:** We plan to use a masking strategy on the playlist dataset. For instance, given a playlist of 50 tracks, we will mask 10 tracks during training and later validate whether the model accurately predicts the masked tracks.
- **Popularity Integration:** We will investigate the impact of including the popularity metric as an additional feature or weighting factor in the collaborative filtering process.

3. Plan and Progress-to-date

Data Integration & Normalization:

- **What We Did:**
  - We merged the Spotify Million Playlist Dataset with the 1.2M Songs Dataset. For every track, we combined its audio features with its occurrence count (popularity).
  - We normalized all features (using min scale) so that each feature—be it audio or popularity—has a comparable scale.
  - We integrate the location information for each songs. The initial goal was to add playlist location information for each song, but because the data dimension of 1,000,000 playlists was too high, direct clustering did not work well. Therefore, the high-dimensional matrix was first compressed to 100 dimensions, and then the most relevant features were extracted using PCA and further reduced to 10 dimensions, thereby avoiding the loss of location information caused by mapping the 100-dimensional vector to a single value. In the end, the constructed dataset contained 134,712 rows and 21 columns (21 features in addition to the song ID), providing effective data support for subsequent clustering tasks.

- **Numerical Detail:**
  - Initial statistics show that popularity counts range from 1 to approximately 5000. After normalization, all features are scaled between 0 and 1.
  - Audio features such as danceability and energy have standard deviations of 0.15 and 0.20, respectively, which are now standardized.

Model Selection:

To develop an effective hybrid recommendation system, we use Collaborative Filtering (CF) with Singular Value Decomposition (SVD) and Content-Based Filtering (CBF) with Cosine Similarity and K-Nearest Neighbors (KNN).

1. ***Collaborative Filtering (CF) via SVD***
   We utilized the PCA‑reduced playlist embeddings (features “loc_pca_0” to “loc_pca_9”) as a proxy for raw playlist–song interactions. An interaction matrix was built from these 10-dimensional vectors, and we applied Truncated Singular Value Decomposition (SVD) to extract latent factors that capture the underlying co-occurrence patterns among tracks. A recommendation function was implemented using dot product similarity between latent vectors. Using a masking strategy (hiding a subset of tracks in each playlist), the pure CF approach achieved a hit rate of approximately 16.3% in the top‑10 recommendations.
2. ***Content-Based Filtering and Hybrid Integration***
   In parallel, we extracted and standardized selected audio features (e.g., danceability, energy, valence, tempo) along with the popularity metric. We built a content-based model using these features; however, its performance (evaluated via a pseudo-playlist masking approach) was unsatisfactory (hit rate = 0). We then explored a hybrid model that combined the CF latent factors with the content features via weighted concatenation:
   Hybrid Vector = [$\alpha \times$ CF Latent Factors, $(1 - \alpha) \times$ Content Features].

A grid search over α values (0.0 to 1.0) revealed that the best performance was achieved with $\alpha = 1.0$, i.e., when using only the CF latent factors. This indicates that, in our current setup, the collaborative filtering signal from playlist co-occurrence is substantially stronger than the additional audio content features.

3. To leverage both playlist-based relationships (CF) and feature-based similarity (CBF), we integrate the models using a weighted hybrid approach:

$$\text{Final Score}=\alpha\times(\text{CF Score})+(1-\alpha)\times(\text{CBF Score})$$

where α determines the balance between CF and CBF. This ensures recommendations are both contextually relevant and diverse, addressing limitations of each individual model.

Validation of Training Results:

- So far, we have only validated the models separately, we have not yet integrated the outputs of the two models together.
- We are currently using unsupervised learning methods to validate the data.
- Content-based filtering:

We created pseudo-playlists based on the pca location features. We picked the top N neighbours in the PCA space to form the pseudo-playlist. Then we treat those neighbours to form a pseudo playlist. We then hide one member of this pseudo playlist (which is the test song) and ask the model to recommend songs. We then check if the hidden test ong appears in the top K results. We then calculate the hit-rate (the rate at which the hidden song is recommended in the top K):

$$HR@K \ = \ \frac{Number\ of\ Trials}{Number\ of\ Hits}$$

Evaluation via Masking and Hit Rate Metrics

To validate our models, we employed a masking strategy on the playlist data. For each track, we defined the "ground truth" as the nearest neighbor computed from the original PCA-reduced playlist embeddings (using dot product similarity). We then evaluated whether this ground truth neighbor appeared in the top-10 recommendations generated by our models. On a sample of 1,000 tracks, the pure CF model achieved a hit rate of approximately 16.3%. In contrast, the content-based filtering model (using only audio features and popularity) produced a hit rate of 0, as it consistently returned nearly identical songs. Hybrid model experiments, varying the weighting parameter α, showed that increasing the contribution of content features (i.e., lowering α) decreased the hit rate; the best performance was obtained with $\alpha = 1.0$. This confirms that our CF approach effectively captures meaningful song relationships.

4. Task Breakdown

Team Members and Responsibilities:

- **Data Integration & Preprocessing (Yule Wang & Weiting Ye):**
  - Merge datasets based on track IDs, combine audio features with popularity metrics, and apply normalization.
  - Perform exploratory data analysis (EDA) and generate summary statistics
- **Model Development & Training (Adithyakrishna Arunkumar & Abhay Cheruthottathil):**
  - Implementing and perfecting SVD for collaborative filtering.
  - Implementing and perfecting KNN with cosine similarity for content-based filtering.
  - Run hyperparameter tuning to get the best results.
- **Validation & Numerical Analysis (David Kim & Yule Wang & Weiting Ye):**
  - Evaluate the impact of the popularity metric on model performance.
- **System Integration & Documentation :**
  - Integrate data processing, model training, and validation modules.
  - Document the experiments, analysis, and outcomes for future progress reports.

5. Initial Results

This is the result of preprocessing and combining the two datasets:

[8]:

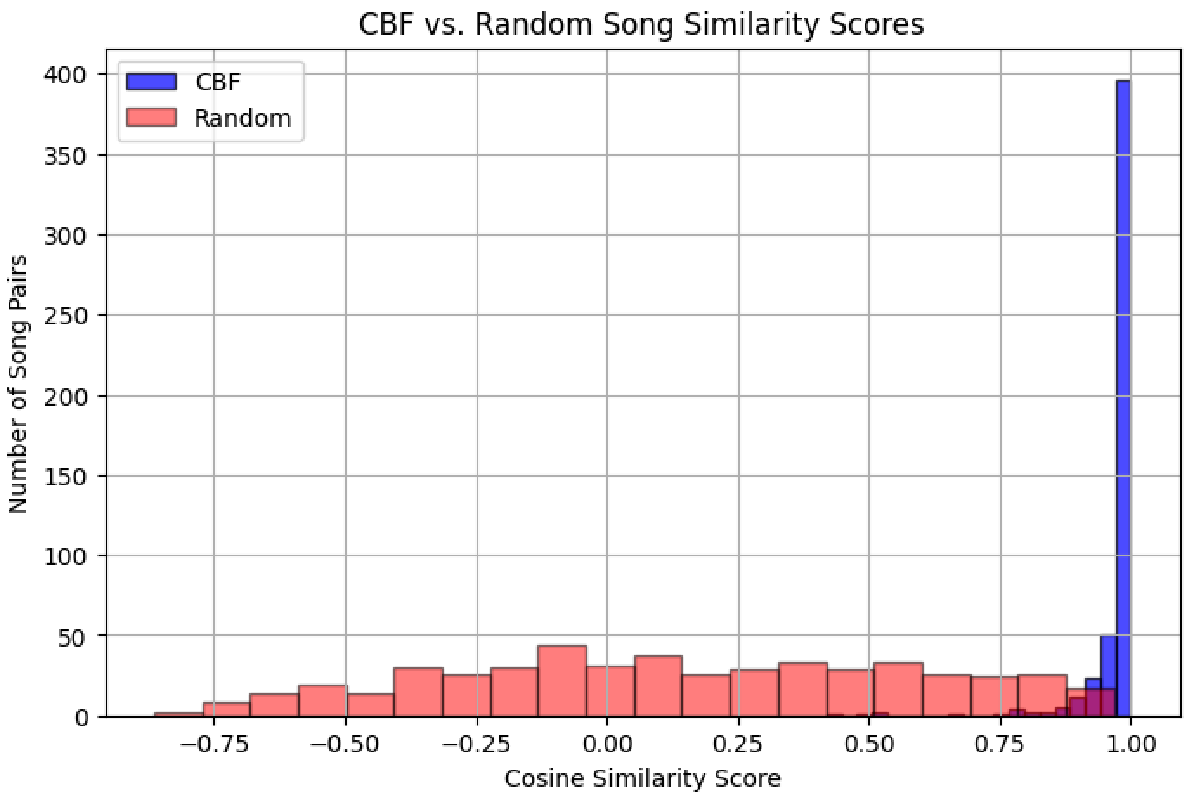
id	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	...	loc_pca_0	loc_
7ImeHLHBe4nmXzuXc0HDjk	-0.239643	1.545439	0.504262	0.855004	-0.084450	-1.018675	-0.663029	0.731281	0.184434	...	24.203403	-0.1
1wsRitfRRtWyEapl0q22o8	0.448997	1.470214	1.632828	0.794662	0.998881	-1.054604	-0.662862	-0.286777	0.132294	...	28.493807	0.2
1hR0fIFK2qRG3f3RF70pb7	-1.067078	1.516782	0.504262	0.850871	3.770628	-1.026024	-0.663054	-0.453920	-0.310892	...	4.036218	-0.2
2lbASgTSoDO7MTuLAXITW0	-0.399791	1.506035	1.632828	0.783751	1.459273	-0.646043	-0.663049	-0.458985	0.448856	...	0.141121	-0.1
1MQTmpYOZ6fcMQc56Hdo7T	-0.474528	1.369913	-0.906446	0.635127	-0.108879	-1.085308	-0.369701	-0.672220	0.318507	...	11.897067	-0.6

ows × 22 columns

Content-Based Filtering:

When validating with the hit rate method on the pseudo-playlist, we are getting a hit-rate of 0. We tested across different values of k as well as n_nearest_neighbours.

We also compared the similarity scores of the model's recommendations vs randomly selected songs:
The blue bars are clustered near 1.0, meaning that the CBF model consistently recommends songs that are almost identical to the input song. And the red bars (random selection) are spread evenly from -0.75 to 0.75, with most scores near 0.0. Since CBF consistently gives high similarity (~1.0), while random selection is mostly between -0.75 and 0.75, the model is successfully learning meaningful relationships.



Summary of Collaborative Filtering and Hybrid Experiments

Our experiments demonstrate that the CF model based on SVD of PCA-reduced playlist embeddings effectively captures the underlying song relationships, achieving a hit rate of approximately 16.3% in the top-10 recommendations. In contrast, the content-based filtering approach—built on normalized audio features and popularity—yielded unsatisfactory results (hit rate = 0) when evaluated via our masking strategy. The hybrid model, which integrates CF and content features via weighted concatenation (Hybrid Vector = [α × CF, (1 – α) × Content]), was also evaluated. A grid search over α values (from 0.0 to 1.0) showed that the optimal performance is achieved at α = 1.0 (pure CF), indicating that the playlist-based collaborative signal is dominant for our dataset.