[Progress report] A Dynamic Music Recommendation System Based on User Feedback

Team #036: Hui Jin, Qi Liao, Zhanxu Liu, Kevin Lu, Li Xiao, Jiaan Yin

1. Introduction

In recent years, the music industry has undergone a significant digital transformation, with a shift towards online music stores and streaming services like iTunes, Spotify, and Youtube Music [1]. Consequently, the demand for automatic music recommendations has surged, facilitating music discovery tailored to individual tastes. At the heart of this lies recommendation systems, which are the crucial engines employing various algorithms to predict user preferences for songs.

2. Problem definition

Previous research has explored various methodologies of song recommendation from the most popular content-based and collaborative-based filtering [2-4] to more recent AI-driven emotion-aware recommendation approaches [5]. However, challenges persist, including user interaction, relevance, cold start, and song freshness [3].

To fill these gaps, this study seeks to develop a dynamic song recommendation system utilizing both collaborative-filtering and content-based filtering, user feedback, and an interactive interface. It aims to recommend popular and new songs aligned with user preferences, updating dynamically based on user feedback.

3. Literature Review

Content-based filtering relies on the features or attributes of items (such as movies, products, or music) to make recommendations [4, 6]. Dynamic music recommendation enhances this process by adjusting recommendations based on user feedback [7], and the visual and interactive features in music recommendation improve user acceptance and understanding of both the underlying data and the recommendations [8].

In the past decade, different machine-learning techniques have been used as content-based methods to identify song similarities. Such approaches have included ensemble learning, clustering, deep learning, and graph-based models [6].

4. Proposed Methods

4.1 Data Description and Cleaning

We obtained our data from a publicly available Kaggle competition (https://www.kaggle.com/datasets/undefinenull/million-song-dataset-spotify-lastfm), comprising two interrelated CSV files representing user listening history. These files includes more than 9.7 million listening history records, involving 50,683 songs and 962,037 users. The user listening history file records each user's play count for individual songs, while the music info file contains features such as name, artist, year, tags, genre, and other song property data. These two files are linked via a unique identifier, the ‘track_id’. For the backend, we stored the preprocessed data in SQLite database.

Figure 1. Exploratory Data Analysis

To analyze the dataset comprehensively, we conducted an Exploratory Data Analysis (EDA) using Power BI and Python shown in **Figure 1**. The correlation matrix chart (**Figure 1.a**) highlights parameters with stronger correlations, such as total play count vs number of users, energy vs loudness, and danceability vs valence. **Figure 1.b** depicts the popularity of songs over time. It suggests that songs produced in the last 20 years are played more frequently by users in comparison to older songs. Furthermore, songs produced post-1970s tend to exhibit more stable music characteristics over the years, such as acousticness, energy, and instrumentalness (**Figure 1.c**), while songs produced from the 1950s to 1970s show strong fluctuations in music characteristics, possibly due to data limitations. **Figure 1.d** illustrates the trend of songs' average characteristics by genre types, highlighting a negative correlation between acousticness (orange bar) and energy (blue bar) observed for these genre types.

4.2 Machine Learning Models/Algorithms

Our approach to modeling involves two models: a Collaborative filtering model (**Model 1** in **Figure 2**) and a Content-based filtering model (**Model 2** in **Figure 2**). Eight songs were recommended by Collaborative filtering model based on preference of similar users via the Singular Value Decomposition (SVD) algorithm. Two more songs were recommended by Content-based model through K-Nearest-Neighbors (KNN) analysis of song attributes from the "Music Info.csv" dataset. User feedback was collected through interactive interface as shown in the interaction loop in **Figure 2**. It refines future suggestions, making music discovery more personalized. Details of each methodology are explained below.

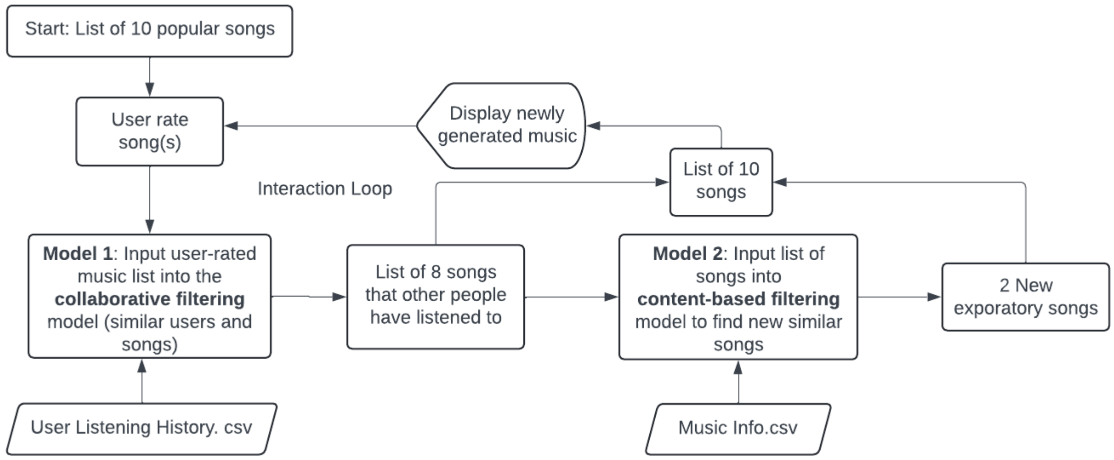

Figure 2. Flow chart of song recommendation system algorithm

4.2.1 Collaborative Filtering Model (Model 1)

In the pre-modeling data preprocessing stage, we applied outlier capping to limit 'play counts' at the 99th percentile to mitigate the influence of extreme values. For collaborative filtering modeling (**Model 1** in **Figure 2**), we utilized the **Surprise** library for its specialized focus on building and analyzing recommender systems, ease of use, and low computing requirements. The Singular Value Decomposition (**SVD**) algorithm was chosen for its effectiveness in capturing latent factors in user-item matrices. Hyperparameter tuning with **GridSearchCV** was implemented to fine-tune model.

4.2.2 Content-based Filtering Model (Model 2)

We designed a second music recommendation system that does not rely on traditional features such as playcount, artist names, or release years, but instead utilizes a K-Nearest Neighbors (**KNN**) model based on musical features to provide a novel approach to music discovery. This novel approach focuses on the attributes that describe the audio features of music: danceability, energy, acousticness, instrumentalness, speechiness, valence, and tempo. This attribute supports the discovery of lesser-known songs, broadening users' musical horizons and differentiating this system from other music recommendation platforms.

4.2.3 Continuous Feedback Loop

Simultaneously, we've implemented a continuous feedback loop, driven by a web user interface, to capture and integrate users' new feedback on recommended songs. This feature allows users to provide feedback (likes), which the system utilizes to fine-tune future recommendations. With each interaction, the user's listening history is enriched with additional features for matching similar users.

4.3 Innovative of Interface:

We built a web service with Flask-RESTful to simplify the process of creating RESTful APIs. Using front-end technologies such as HTML, CSS, and JavaScript(D3) for building user interfaces and client-side functionality. The visual of this webapp is shown in **Figure 3**.

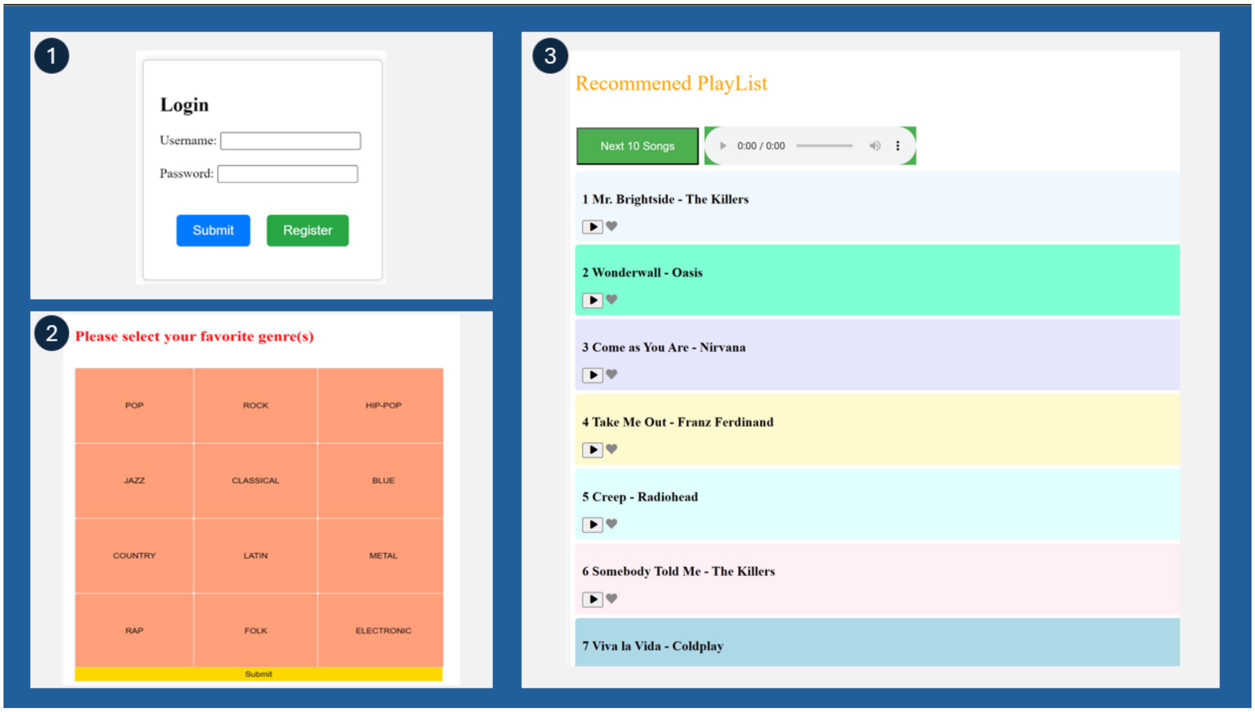

Figure 3. Innovative Interface: Dynamic song recommendation and Feedback integration (user **(1)** login page, **(2)** a cold start page for new users to choose favorite genre(s), and **(3)** a customized user dashboard containing a dynamic recommended song list with music playing and feedback function)

The webapp is designed to realize the computational function of **dynamic song recommendation** and **Feedback integration.** It includes a user **(1) Webpage 1:** login page, **(2) Webpage 2:** a cold start page for new users, and **(3) Webpage 3:** a customized user dashboard containing a recommended song list and a music player. A snapshot of each of the pages is shown above in **Figure 3**. The login page allows users to log in so they can get customized recommended songs. Each user will be assigned a user ID that matches the "user_id" in the "User Listening History" data file. Once users log in to the system, they will be identified as new users or existing users with a listening history. If the user is new, the system will bring him/her to webpage 2, which shows a collection of popular music genres in orange tiles. New users can pick multiple genres that they like and get a recommendation list after the user hits the "Submit" button. Once the tile is picked, it will turn green. If the user is an existing user, he/she will be brought to webpage 3, which is the customized user dashboard that includes a dynamic recommended song list generated by our innovative recommendation algorithm. More predicted songs can be loaded if the user hits the "Next 10 songs" button. Each song includes two feedback buttons under the name of the song – play icon ▷ and heart icon ♡. The play icon allows users to "play" or "pause" the audio which will automatically update the "User Listening History", while the heart icon turns red if the user provides positive feedback which will add weight to that music type in the next round of recommendation.

5. Evaluation

5.1 Evaluation of Collaborative Filtering Model (Model 1)

The dataset is partitioned into training and testing sets, with a split ratio of 85% for training and 15% for testing. We employed cross-validation for model evaluation. The prediction on the test dataset gives an RMSE value of 2.9347 and a MAE value of 1.6458. After evaluating algorithm SVD on three splits of test dataset, it gives a mean RMSE value of 2.9748, with std of 0.0018 and a man MAE value of 1.6794, with std of 0.0002. The consistent and lower values of RMSE and MAE signify superior performance, establishing them as standard benchmarks for recommendation systems.

In addition to accuracy metrics, we employ Precision and Recall at K to evaluate the model's efficacy in recommending relevant items within the top-K suggestions. These metrics play a pivotal role in assessing the practical effectiveness of the recommendation system from the user's perspective, providing insights into the system's ability to generate personalized and pertinent recommendations. The excellent avg. precision score of 0.89 and average recall value of 0.79 truly demonstrate good practical effectiveness of this recommendation system from the user's perspective.

5.2 Evaluation of Content-based model (Model 2)

Evaluating a KNN-based music recommendation system focused on musical features is inherently challenging due to the subjective nature of music preference and the emphasis on discovery. Success in this aspect might not be fully captured by traditional metrics like accuracy or precision.

Without a live feedback loop with real users, it's challenging to gather data on the effectiveness of the model. Small scale testing within the group was done to evaluate the recommended exploratory songs. But our own bias would interfere with true feedback. Therefore, we recommend the following evaluation for future exploration:

- User Studies and Surveys: Conduct detailed user studies or surveys to gather qualitative and quantitative feedback on user satisfaction and discovery within the recommended music.
- Long-term Studies: Track user engagement over time to understand how the model influences long-term music exploration and listening habits.

5.3 Evaluation of Interface

Our evaluation strategy includes a structured online survey aimed at understanding the user experience and the practical utility of the tool for our intended audience. This survey, targeting a representative sample of 20 users, is designed to get both qualitative and quantitative feedback around the following Human Computer Interaction design principles: **Simplicity** (no irrelevant information), **Affordance** (the design tells you how to use the tool), **Flexibility** (multiple ways to do the same thing), and **Overall satisfaction**.

Detailed questionnaire is included in **Appendix**. Responses are analyzed to pinpoint areas of success and opportunities for optimization. Here is the summary of survey results:

- The interface was highly rated for its simplicity, affordance, flexibility, and overall experience. Specifically, cold start page with clickable tiles of different genre is very easy to navigate and use. In the user dashboard, the two-music player design is favored by users. The individual playing function under each recommended song makes it easy to play or pause each song without scrolling all the way to the top or bottom of the webpage. The player on the top makes it easy to know whether there is some song playing and easy to pause it without searching through the list.

- Users also mentioned that the “like” button (heart icon) is a simple and intuitive way to provide feedback.

Here are some recommendations for future development:

- Currently, the music cannot be played automatically as a playlist. Some users like to use it as a music player. And the song connected with the link is a short version of the full song. Thus, it is helpful if Youtube link or Google search can be provided as a link for each recommended song in the user dashboard.
- It is not very clear to the users which songs are exploratory songs. To make it clear, the exploratory songs can be put in a separate section or be separated with a different visual. The benefit of this is the user can choose between listening to preferred songs or focusing on exploring new songs.

6. Discussion and Conclusions

This study developed a dynamic song recommendation system utilizing both collaborative-filtering and content-based filtering, user feedback, and interactive interface. It aims to recommend popular and new songs aligned with user preferences, updating dynamically based on user feedback. Unlike other song recommenders, such as Spotify, Youtube music, etc., which focus on streaming of songs, our system aims at exploring new music and finding out the ones user like in a short period of time. To do that, we used the million song dataset from a publicly available Kaggle competition (https://www.kaggle.com/datasets/undefinenull/million-song-dataset-spotify-lastfm). We used a collaborative filtering model using SVD to identify similar users based on their listening history while 2 exploratory songs were recommended based on musical characteristics using KNN.

To make it user-friendly and enjoyable to use this system, we made a dynamic interactive webapp including three webpages. The interface was highly rated for its simplicity, affordance, flexibility, and overall experience.

There are several recommendations for future development: **(1)** Include Youtube link or Google search as a link for each recommended song in the user dashboard. **(2)** Put exploratory songs in a separate section or be separated with a different visual to allow users to choose whether they want to hear exploratory songs. **(3)** To better evaluate user satisfaction of the recommendation of exploratory songs (**Model 2**), we recommend conducting detailed user studies or surveys to gather qualitative feedback on user satisfaction and discovery within the recommended music and track user engagement over time to understand how the model influences long-term music exploration and listening habits.

All team members have contributed a similar amount of effort.

Reference

1. Van den Oord, A., S. Dieleman, and B. Schrauwen, *Deep content-based music recommendation.* Advances in neural information processing systems, 2013. **26**.
2. Yajie Hu, M.O., *NEXTONE PLAYER: A MUSIC RECOMMENDATION SYSTEM BASED ON USER BEHAVIOR*, in *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. 2011: Miami,Florida (USA) p. 103-108.
3. Serhii Ripenko, N.H. *Music recommendation system: all you need to know*. 2022; Available from: https://www.eliftech.com/insights/all-you-need-to-know-about-a-music-recommendation-system-with-a-step-by-step-guide-to-creating-it/.
4. Kathavate, S. *Music Recommendation System using Content and Collaborative Filtering Methods*. 2021.
5. Hieu Tran, T.L., Anh Do, Tram Vu, Steven Bogaerts, Brian Howard. *Emotion-Aware Music Recommendation*. in *Conference on Artificial Intelligence*. 2023.
6. Deldjoo, Y., M. Schedl, and P. Knees, *Content-driven music recommendation: Evolution, state of the art, and challenges.* Computer Science Review, 2024. **51**: p. 100618.
7. Pampalk, E., T. Pohle, and G. Widmer, *Dynamic Playlist Generation Based on Skipping Behavior*. 2005. 634-637.
8. Andjelkovic, I., D. Parra, and J. O'Donovan, *Moodplay: Interactive music recommendation based on Artists' mood similarity.* International Journal of Human-Computer Studies, 2019. **121**: p. 142-159.

Appendix

Survey Questionnaire

1. On a scale of 1-5, rate the **Simplicity** (no irrelevant information) of the **Webpage 2 (Cold Start)**?
   1 - Not Simple At All
   2 - Slightly Simple
   3 - Moderately Simple
   4 - Very Simple
   5 - Extremely Simple
2. On a scale of 1-5, rate the **Affordance** (the design tells you how to select your favorite genres) of the **Webpage 2 (Cold Start)**?
   1 - Not Intuitive At All
   2 - Slightly Intuitive
   3 - Moderately Intuitive
   4 - Very Intuitive
   5 - Extremely Intuitive
3. On a scale of 1-5, rate the **Simplicity** (no irrelevant information) of the **Webpage 3 (Dashboard)**?
   1 - Not Simple At All
   2 - Slightly Simple
   3 - Moderately Simple
   4 - Very Simple
   5 - Extremely Simple
4. On a scale of 1-5, rate the **Affordance** (the design tells you how to play and rate recommended songs) of the **Webpage 3 (Dashboard)**?
   1 - Not Intuitive At All
   2 - Slightly Intuitive
   3 - Moderately Intuitive
   4 - Very Intuitive
   5 - Extremely Intuitive
5. [**Flexibility**] On a scale of 1-5, rate the usefulness of the design of two music players (one underneath each song and one at the top) of the **Webpage 3 (Dashboard)**.
   1 - Not Useful At All
   2 - Slightly Useful
   3 - Moderately Useful
   4 - Very Useful
   5 - Extremely Useful
6. **[Affordance]** Was the function of the feedback button (heart icon ♡) clear and understandable?
   1 - Not Clear At All
   2 - Slightly Clear
   3 - Moderately Clear
   4 - Very Clear
   5 - Extremely
7. Clear Rate the overall user-friendliness of the interface.
   1 - Not User-Friendly At All
   2 - Slightly User-Friendly
   3 - Moderately User-Friendly
   4 - Very User-Friendly
   5 - Extremely User-Friendly

8. What was your overall experience using the interface?
    1 - Very Poor
    2 - Poor
    3 - Average
    4 - Good
    5 - Excellent
9. Any other recommendation or suggestions on interface or functionality improvement?