

# COMP9727 Recommend System Project Design

Jinghan Wang z5286124

## Abstract

This paper outlines the design of an advanced music recommendation system aimed at providing personalized music suggestions to users. The system utilizes multimodal fusion to analyze music from multiple perspectives, including but not limited to audio signals, lyrics, and user behavior data. The goal is to enhance the user experience by delivering highly relevant and enjoyable music recommendations.

Number of Words: 2135

## Scope

### Introduction

Music has always been an indispensable part of human culture and society. From ancient rituals and ceremonies to contemporary entertainment and personal enjoyment, music permeates every aspect of our lives. It serves not only as a source of joy, but also as a means of expression and communication, transcending linguistic and cultural boundaries.

In the digital age, propelled by technological advancements and the proliferation of online streaming platforms, the way we access and consume music has undergone significant changes. With the vast amount of music readily available, finding the right tracks that align with personal preferences has become a complex challenge. This is where music recommendation systems come into play.

In this design, we envision an advanced music recommendation system that utilizes multimodal fusion techniques. The goal is to create a system that not only recommends music that matches the user's taste but also continuously adapts to their evolving preferences, providing a dynamic and engaging music discovery experience.

### User Interaction

Due to the copyright restrictions on most music, this project is built on the Spotify platform. On desktop, it could be a Spotify browser extension or a custom web page with Spotify player functionalities. On mobile, it could be a custom app that integrates the Spotify player functionalities. All interfaces, in addition to standard Spotify music playback, will feature options to choose, save, or remove tracks. These interactions directly feed into the recommendation algorithm as user feedback. Users can perform normal actions such as searching, viewing play history, and listening to random tracks. They can also skip to the next recommended song based on the similarity metrics of the algorithm. These actions will dynamically influence the weight of the songs in the recommendation system.

For new users with no prior records, if their first action is to perform a search, the search results will serve as the initial basis for recommendations. If the first action is to listen to randomly recommended songs, selections will be made from Spotify's current weekly or monthly Top 10 lists. The user ratings for these songs will update the weights of the recommendation system accordingly.

Since it is impractical to access all Spotify playlists, I plan to use a foundational database from Hugging Face, containing a million songs. If a user searches for a song not in this database,

the song will be added to the database. Recommendations will be made only from the content of the database.

Additionally, the system will include advanced user interaction features such as personalized playlists and dynamic recommendation adjustments based on real-time feedback. Users will have the option to create and share playlists, and the system will track user performance and engagement with these playlists to further refine its recommendation algorithms.

### Target Audience

Since music is universal, anyone who loves music and is willing to share their music-related preferences and privacy with this project can be an audience. The system is designed to cater to a diverse user base, ensuring a personalized and satisfying music discovery experience for all.

### Advanced Concepts

Tags will be assigned to each user. If a group of users with similar tags shows a preference for a particular song, that song will be recommended to others with the same tag. However, this requires a large user base and is not applicable at the current project stage.

Furthermore, the system will explore the implementation of collaborative filtering among users with similar tags. By analyzing listening patterns and preferences within these user groups, the recommendation system can enhance its accuracy and relevance, providing a more customized music discovery experience.

### Dataset

For the music dataset, I plan to divide it into two parts. The first part is the song collection, for which I will use the Spotify Million Song Dataset([vishnupriyavr 2023](#)) from Hugging Face as the foundational song data. This dataset is comprehensive and includes a wide range of songs along with their lyrics, which can be directly used for lyric analysis.

Initially, I considered obtaining the music files for each song to extract detailed audio features such as tempo, timbre, and other acoustic characteristics. However, after performing some calculations, I realized that even with parallel processing, it would take approximately 40 days to complete the extraction of a million songs. This time frame is impractical for my project.

Therefore, to expedite the process, we will utilize the Spotify Web API. This API allows me to efficiently obtain a variety of audio features, including pitch, loudness, danceability, energy, and more. In addition, the API provides extensive metadata

about the songs, artists, albums, and related information, which is invaluable for my analysis.

For any supplementary needs that arise during the project, I may incorporate other APIs such as the Last.fm API and AcousticBrainz API. These APIs can provide additional data on aspects such as music genre, user tags, emotional content, and advanced audio analysis metrics. This approach will ensure a rich and multifaceted data set, enabling a comprehensive analysis and accurate music recommendations.

### Database limitations and solutions

As previously mentioned, the million song data set that we use was last updated at the end of 2023. Consequently, it does not include music released after this period. Additionally, there are undoubtedly numerous niche or lesser-known tracks that have not been included in this dataset. This gap presents a challenge, as the absence of recent releases and niche music could limit the accuracy and relevance of our recommendations.

To address this issue, we have implemented a dynamic update mechanism within our system. Whenever a user searches for a song that is not already present in our database, the system will automatically fetch the latest data for that song using the Spotify Web API. This data includes various audio features, metadata, artist information, album details, and more. Once the data for the new song are retrieved, it is seamlessly integrated into our existing database. This process ensures that our dataset remains current and comprehensive, effectively bridging the gap caused by the dataset's last update in 2023.

By continuously updating the database with new entries, we can provide users with recommendations that are not only personalized but also reflective of the most recent musical trends and releases. This approach ensures that our recommendation engine remains robust and capable of suggesting both mainstream hits and hidden gems, thereby catering to a wide spectrum of musical tastes and preferences.

### Method

For this model, considering the diversity of the user base and the multimodal nature of the data (lyrics, audio features, metadata), a hybrid recommender system that combines multiple methods will be the most effective approach.

### Lyrics Matching

Similar to movie recommendations, each song's lyrics can be converted into TF-IDF (Term Frequency-Inverse Document Frequency) vectors. This involves calculating the frequency of the term for each word in the lyrics and adjusting it to the frequency with which the word appears in other songs. By representing the lyrics in this vectorized form, we can calculate the cosine similarity between the lyrics of songs the user likes and other songs. This similarity measure can then be used to recommend songs with similar lyrical content, thus enhancing the relevance of the recommendations based on textual analysis.

### Audio Feature Matching

Utilizing the audio features provided by the Spotify API, such as danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentality, liveness, valence and tempo, we can compute the similarity between songs based on these attributes. Each characteristic provides a different aspect of the characteristics of the music, and by calculating the similarity between these

multiple dimensions, we can recommend songs with similar audio profiles. Alternatively, converting the audio signal into spectrograms and using convolutional neural networks (CNNs) to extract features can provide a more nuanced understanding of the audio. These features can then be compared to identify similar songs, leveraging deep learning techniques for more sophisticated audio analysis.

### Artist and Album Information

By leveraging metadata such as genre, artist, release year, and album, we can create rules to recommend songs. For instance, if a user has shown a preference for a specific genre or artist, we can recommend other songs that fit these criteria. This rule-based approach uses explicit knowledge about the music to provide recommendations that align with the user's known preferences. Additionally, understanding the context of the music, such as its cultural and temporal significance, can further refine the recommendations, ensuring they are relevant and engaging for the user.

### Listening History and Time of Play

Using recurrent neural networks (RNNs) or transformers to model the sequence of songs a user listens to allows us to recommend the next song in the sequence. This approach captures the temporal dynamics of listening behavior, understanding how users' preferences change over time. By incorporating contextual information, such as the time of day, user activity, or user demographics, we can further tailor recommendations to fit the user's current situation. For example, analyzing listening patterns might reveal that users prefer different types of music in the morning compared to the evening, or that certain activities are associated with specific genres.

### User Tag-Based Promotion

Utilizing user tags and preferences, we can implement user-based collaborative filtering using K-Nearest Neighbors (KNN). By calculating the similarity between users based on their tags and preferences, we can identify users with similar musical tastes. Recommending songs that these similar users like can provide personalized recommendations that go beyond the user's immediate listening history, introducing them to new music that aligns with their tastes.

### Hybrid Recommendation System

Combining all the individual models described above into a hybrid recommender system can provide a comprehensive and robust recommendation engine. Using a weighted hybrid approach, we can integrate content-based, collaborative filtering, knowledge-based, sequential, and context-aware recommendations. By assigning appropriate weights to each method, the system can balance the influence of various factors, ensuring that the final recommendations are well-rounded and tailored to the user's diverse preferences.

**Weighted Hybrid Approach:** Assign weights to different recommendation methods based on their relevance and effectiveness. For example, we might give more weight to content-based recommendations for users with well-defined lyrical or audio preferences, and more weight to collaborative filtering for users with rich interaction histories. **Adaptive Weighting:** Dynamically adjust weights based on user feedback and behavior. If

a user frequently interacts with recommendations from a specific method, increase the weight for that method to enhance personalization.

By leveraging the strengths of multiple recommendation techniques, the hybrid system can provide high-quality, personalized recommendations that cater to the diverse and dynamic nature of user preferences. This comprehensive approach ensures that users receive relevant and engaging music suggestions, enhancing their overall experience with the platform.

## Evaluation

For evaluating the model, we consider multiple dimensions to ensure a comprehensive assessment. Our evaluation strategy includes the following components:

1. **User Satisfaction Scores**

We implement options for users to express their preferences by liking, favoriting, or disliking songs. Additionally, songs that users do not interact with are treated as neutral (neither liked nor disliked). This metric provides direct feedback on user satisfaction and helps us understand user preferences more accurately.

2. **Click-Through Rate (CTR)**

This metric helps measure the effectiveness of the recommendation in capturing user interest. A higher CTR indicates that the recommendations are relevant and appealing to users.

3. **Repeat Listen Rate**

The frequency with which users repeatedly listen to a specific song gauges the user's fondness for that song. A high repeat listen rate suggests that the song is highly favored by users.

4. **Dwell Time**

The duration of time a user spends listening to a recommended song before moving to the next one indicates engagement and satisfaction. Longer dwell times suggest higher user satisfaction, while shorter dwell times followed by quick skips suggest dissatisfaction.

5. **Skip Rate**

The rate at which users skip a song after listening to it for a short period can indicate dissatisfaction with the recommendation, signaling the need to adjust the recommendation algorithm.

6. **Temporal Listening Patterns**

By analyzing listening habits across different times of the day, we can tailor recommendations to better suit user preferences at different times. For example, users might prefer upbeat music in the morning and relaxing tunes in the evening.

## Model Training and Evaluation Process

1. **Pre-Trained Models**

We utilize several pre-trained models to generate initial recommendations, fine-tuning them with our specific dataset to better capture the nuances of user preferences. Each model is tested under various conditions to evaluate its performance in recommending songs that users are likely to enjoy.

2. **Dynamic Weight Adjustment**

Based on user interactions and feedback, we dynamically adjust the weights assigned to each model, allowing the system to adapt to changing user preferences and improve

the relevance of recommendations over time. Algorithms that can learn from real-time data continuously optimize the weightage of different models.

3. **Precision @ N and Recall @ N**

These metrics are crucial for tuning model parameters. Precision @ N measures the proportion of relevant recommendations within the top-N recommended items, while Recall @ N measures the proportion of all relevant items included in the top-N recommendations. By balancing precision and recall, we ensure that the recommendations are both accurate and comprehensive.

## Reference List

vishnupriyavr. 2023. spotify-million-song-dataset.  
<https://huggingface.co/datasets/vishnupriyavr/spotify-million-song-dataset>.