Music Recommendation System

Anant Bhaskar
IIIT Delhi
anant21516@iiitd.ac.in

Hrishav Basu
IIIT Delhi
hrishav20067@iiitd.ac.in

Priyansh
IIIT Delhi
priyansh21183@iiitd.ac.in

November 23, 2023

1 Introduction

In this project proposal, we outline our plan to develop a Music Recommendation System. The goal of our system is to provide users with personalized music recommendations based on their preferences and listening history. With the increasing amount of music available online, users often struggle to discover new music that aligns with their tastes. Our proposed system aims to address this issue and enhance users' music discovery experience.

1.1 Motivation

The motivation behind this project comes from the ever-growing demand for personalized content recommendation. With the popularity of streaming platforms, users have access to vast music libraries, but they often find it overwhelming to navigate and find music that suits their preferences. We believe that building an intelligent music recommendation system will enhance user engagement and satisfaction by providing them with relevant music choices.

1.2 Related Work

We have identified the following three recent works in the field of music recommendation:

1. "Deep Learning for Music Recommendation: Challenges and Opportunities" by Li et al. (2021) [**?**]: This work provides an overview of the challenges and opportunities of using deep learning techniques for music recommendation. It discusses various approaches and highlights the potential benefits of incorporating deep learning models.
2. "Music Recommendation System using Collaborative Filtering and Deep Learning" by Wang et al. (2020) [**?**]: This paper presents a hybrid recommendation system that combines collaborative filtering and deep learning to provide accurate music recommendations. The approach considers both user preferences and item characteristics.
3. "Item-based Variational Auto-encoder for Fair Music Recommendation" by Jinhyeok Park1,† , Dain Kim1,† and Dongwoo Kim1,*(2022) [**?**]: The focus of the paper is on addressing fairness concerns in music recommendation systems. The proposed model leverages item-based collaborative filtering and VAEs to provide personalized music recommendations while mitigating potential biases and ensuring fairness in the recommendations.

2 Timeline

Our project timeline is as follows:

- **Weeks 1-2**: Literature review and familiarization with existing music recommendation techniques.
- **Weeks 3-4**: Data collection and preprocessing, including acquiring user listening histories and music metadata.

- **Weeks 5-6**: Implementation of collaborative filtering and deep learning models for music recommendation.
- **Weeks 7-8**: Model evaluation and fine-tuning for performance improvement.
- **Weeks 9-10**: User interface development and integration with the recommendation system.
- **Weeks 11-12**: Testing, debugging, and finalizing the system for presentation.

3 Individual Tasks

To ensure effective project management and distribution of responsibilities, we have outlined individual tasks for each team member:

- **First Author**: Data collection and preprocessing, collaborative filtering model implementation.
- **Second Author**: Machine learning model implementation, model evaluation and fine-tuning.
- **third Author**: Data collection and fine-tuning of the model.

The success of the project will be assessed based on the completion of these tasks and the overall system performance.

4 Final Outcome

We aim to create a functional music recommendation system that provides users with accurate and relevant music suggestions. By leveraging collaborative filtering and deep learning techniques, we intend to enhance the quality of recommendations and improve the music discovery experience for users.

5 Dataset details with data preprocessing techniques

Creating Music Recommendation system involves working with music related datasets that are listed below:

5.1 Dataset

- **Spotify Tracks Dataset**:This is a collection of Spotify music from 125 different genres. Each track has its own set of audio features. The data is in tabular CSV format.Easy to build a Recommendation System based on some user input or preference.
- **Additional Contextual Data**: Depending on the recommendation system's complexity, contextual data such as location, time of day, and user mood may be included.

5.2 Data Preprocessing Techniques

- **Data Cleaning**: In this technique, we handle the missing values, duplicates, and inconsistencies in the dataset. For instance, we might want to remove duplicate data and fill missing values in the audio feature with some default values.
- **Data Integration**: In this technique, if the data is coming from multiple sources, we integrate them as one dataset. This technique may merge user-artist interactions with music metadata, ensuring identifier consistency.
- **Normalization and Scaling**: In this technique, when working with some numerical values, we need to normalize or scale them to ensure that they have similar ranges. This will help algorithms to perform better and prevent any feature from dominating any recommendation.
- **Text Processing (for textual data)**: In this technique If our dataset includes textual information, such as song titles or artist names, we can apply techniques like tokenization, stemming, and vectorization (e.g., TF-IDF or word embeddings) to convert text into a numerical format that recommendation algorithms can use.
- **Handling Imbalanced Data**: In some cases, user-item interactions may be imbalanced, with

some items being top-rated and others less so. Techniques like oversampling, under-sampling, or different loss functions can help address this imbalance.

- **Data Splitting**: To evaluate recommendation algorithms, it's essential to split your dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to fine-tune hyperparameters, and the test set is used to evaluate the model's performance.

6 Methodology

6.1 Collaborative Filtering

In order to give consumers individualized suggestions based on their past interactions and preferences as well as the preferences of other users, collaborative filtering is a prominent technique used in recommendation systems.Important elements:

- **User-Item Interaction Matrix**:It is a data structure that depicts user interactions with things in a tabular style, with rows denoting people and columns denoting items, and cells denoting details of those interactions. This matrix is used to record and arrange user preferences and behavior, which aids in creating tailored suggestions.

- **Similarity Metrics**: Indicate the metrics for measuring similarity between songs. It is performed by using Cosine similarity, Pearson correlation, and Jaccard similarity.

- **Item-Based Collaborative Filtering**: Item-based collaborative filtering is a recommendation technique that suggests items to users based on the similarity between items themselves rather than relying on user similarity. it is done by creating a item similarity matrix.

- **Matrix factorization**: Matrix factorization is a technique used in recommendation systems to uncover latent patterns or features in a user-item interaction matrix. It's particularly popular for collaborative filtering, where it helps in making personalized recommendations based on these latent factors.

6.2 Content-based Recommender Systems

A sort of recommendation system called a content-based recommender system makes recommendations to users based on the traits or content of the items and the preferences the users have expressed. These systems are especially helpful when the items in the system have access to specific information, such as textual descriptions, tags, or characteristics. How content-based recommender systems operate is as follows:

- **Item Representation**:Each component of the system is represented by a collection of characteristics or features. These attributes, such as genre, keywords, artists (for music), etc.

- **User Profile**: Using the user's behaviors and choices, the system builds a user profile. The user's prior preferences are reflected in this profile, which frequently makes use of the same characteristics as the products.

- **Feature Extraction**: The system pulls pertinent features for both the user profile and the objects. This may employ feature engineering, dimensionality reduction, or natural language processing (NLP) to handle textual data.

- **Calculating Similarity**: The system determines how closely each item in the catalog resembles the user profile. Cosine similarity, Jaccard similarity, or Pearson correlation are examples of common similarity measures.

- **Ranking and Recommendation**: The method ranks the things according to similarity scores and suggests to the user the top-N items with the highest similarity scores.

6.3 Hybrid Recommender Systems

Hybrid recommender systems integrate two or more distinct recommendation approaches to provide suggestions that are more precise and efficient. These methods take use of each technique's positive aspects while minimizing its negative ones. In recommendation systems, hybrid techniques are frequently used to address issues including the cold start problem, data scarcity, and the demand for personalisation. There are several ways to mash together recommendation methods:

- **Weighted Hybrid**:In this method, suggestions are created separately using various recommendation algorithms (such as collaborative filtering and content-based filtering), and the weighting of each recommendation's relevance is determined by how well it performs. These distinct recommendations are combined into a final recommendation using a weighting system. Based on user behavior or the quality of a suggestion, weights can be fixed or dynamically changed.

- **Feature Combination**: In content-based recommendation systems, features from many sources or recommendation methods can be integrated to provide a user or object profile that is more in-depth. Genre and artist-based content-based features can be integrated with user-item interactions-based collaborative filtering features in a music recommendation system to provide a more robust user-item matrix.

- **Switching Hybrid**: This method dynamically chooses the recommendation methodology depending on certain user or item attributes. For new users who have little interaction history, the system could initially choose for a content-based approach; but, when the user's interaction history develops, it might convert to collaborative filtering.

- **Cascade Hybrid**: The recommendations from one approach are used as input in cascade hybrid systems. An initial set of suggestions may be provided using a collaborative filtering model, and these recommendations might subsequently be filtered or improved using a content-based approach. This strategy can aid in ensuring that recommendations are pertinent and tailored.

- **Model-Based Hybrid**: Some hybrid systems provide a cohesive framework from a single model that combines many recommendation algorithms. This model is capable of capturing intricate connections between consumers, products, and recommendation engines. Machine learning strategies, such as deep learning or ensemble approaches, can be used to generate model-based hybrids.

6.4 Proposed Method

Develop a music recommendation system that offers personalized song recommendations based on user-provided input, such as a song name and genre. The system utilizes a hybrid approach, combining content-based and collaborative filtering techniques, along with feature-based matching, to enhance the accuracy and relevance of recommendations.

- **User Input**:Allow users to input a song name and genre as their preferences for recommendations. These inputs serve as initial filters for the recommendation process.

- **Content-Based Filtering**: Employ content-based filtering to narrow down the dataset to songs that match the user's specified genre. This initial filtering ensures that recommendations align with the user's genre preference.

- **Collaborative Filtering**: Implement collaborative filtering techniques to enhance recommendations. Utilize user-item interaction data (if available) to identify songs and artists that are similar to the user's preferences, based on the behavior of users with similar tastes.

- **Feature-Based Recommendations**: Calculate average feature values (e.g., danceability, energy) for the songs within the specified genre. These averages provide reference points for recommending songs with similar feature characteristics to the user's input.

- **Content-Based Recommendations**: Utilize the calculated average feature values to identify songs within the same genre that closely match the user's input in terms of feature characteristics. Similarity metrics are applied to identify these matches.
- **Hybrid Recommendations**: Combine the results from content-based and collaborative filtering approaches to generate a final set of recommendations. This hybrid approach ensures that recommendations are both genre-specific and personalized based on user behavior.
- **Presentation**: Present the recommended songs to the user, including song name, artist, popularity, and links to listen or explore further.
- **New Song Recommendations**: For newly introduced songs, extract their features and calculate the average feature values within their respective genres. Recommend songs from the same genre with feature characteristics similar to the new song.

/

7 Conclusion

the Hybrid Music Recommendation System with Genre-Based Feature Matching offers a comprehensive solution to the challenges of music recommendation. By combining content-based, collaborative filtering, and feature-based approaches, the system provides users with a dynamic and engaging music discovery experience that caters to their preferences and encourages exploration within their favorite genres. As music continues to evolve, this system remains adaptable and poised to deliver high-quality recommendations to music enthusiasts.