Individual Software Project

Group Music Recommendation System

Student: Artur Papyan

Supervisor: Patrik Dokoupil

Introduction

A group music recommendation system is a specialized type of recommender system that is engineered to suggest music that is likely to be enjoyed by a collective group of users, as opposed to individual users. This system is more complex than individual recommendation systems because it must take into account the varied tastes and preferences of multiple users simultaneously.

The primary goal of such a system is to identify and recommend music tracks, albums, or playlists that will satisfy the group as a whole, which involves finding a balance between differing musical interests.

This is typically achieved through several steps:

Profile Aggregation: The system collects data on the musical preferences of each individual within the group. This data can be explicitly provided by the users or inferred from their listening history.

Preference Modeling: It creates a model that represents the group's combined preferences. This can be a complex process as it might require finding a middle ground between users who have very different tastes.

Recommendation Generation: Using the group preference model, the system generates recommendations that it predicts will be well-received by the entire group.

Group decision: After the system will make its suggestions, the group should decide which ones are ideal for them in later group discussion.

//Feedback Incorporation: Ideally, the system will learn from the group's reactions to its //recommendations (e.g., if certain songs are skipped or played repeatedly). (for the future)

Problem Statement

The aim of this project is to create a Group Music Recommendation System, like the one described above. The system will operate in a webpage with some specifications that will be described later in the document. This project will be used for more serious project, bachelor’s thesis, which will allow creating “user study” to analyze the behavior (the decision-making part) of group members when interacting with GRS (but that is in future for bachelor’s thesis). The bachelor thesis will be about running/conducting the user study. Now the main perspective is creating the whole system.

Functionality overview

When the user enters the webpage there appears “Log-in” or “Sign up” window. After user signs up or logs into the system, the user will see an option create a group (maybe also see the previous recommendations the system suggested from previous groups). Here the important part is that there are two types of users: “participants” and “group administrator”. The participant is someone, who is going to use the system and participate in the user study. The "administrator" corresponds to the person that creates the user study, gets the link and distributes it to other users/participants. Each will be able to create group with specified count of group members (so becoming what we call an administrator). When the group is created, the administrator will get a unique group link to distribute among other users who want to join to that recommendation group. When the group will be created the system will retrieve 20 songs from Spotify API, which should be rated by the participants. After other members search the unique link, they will get that songs to rate. When all the group members will rate the songs, on that page will appear a small chat window, and on that page the system will show the recommendations for their group (the links of the songs will appear, or I will do my best to include some play song buttons to play some short part from songs). The group of users will be able to discuss their opinions about the suggestions and pick some of them. Then when they all close the window, their chat will be closed (maybe the closing part will be other way) and group dismissed.

Used technologies

The core programming language for building the application and implementing the Machine Learning models is Python. Machine Learning library “scikit-learn” is used for building the recommendation model. Spotify API is integrated retrieving real-time music data. User authentication using Flask-Login or a similar library. Flask will be used to develop the web application. SQLite database is used to store user profiles, group information, and music data. HTML, CSS, and JavaScript for creating a user-friendly web interface.

For the other specifications please check the GitHub repository of the project(later to change to GitHub): [**https://github.com/AnOnRT/Music-Recommendation-System-for-Group#readme**](https://github.com/AnOnRT/Music-Recommendation-System-for-Group#readme)