Song Recommendations for Users

CMPE 256 – Advance Data Mining

Spring 2023

**Group Number:** 12

**Team Name**: Rockstars

**Student Name**: Drona Jagad (016651206), Hewitt Kothari (016662230), Hardi Trivedi (016589768), Nemil Panchamia (016145272)

**Datasets Description:**

[Spotify 1.2M+ Songs | Kaggle](https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs)(345.74 MB)

**Column Description**

| **Serial No.** | **Column Name** | **Column Description** |
| --- | --- | --- |
| 1 | id | Spotify track ID |
| 2 | name | Track title |
| 3 | album | Album title |
| 4 | album\_id | Spotify album ID |
| 5 | artists | List of artist names |
| 6 | artists\_ids | List of Spotify artist IDs |
| 7 | track\_number | Track number |
| 8 | disc\_number | Disc number |
| 9 | explicit | Whether the song is explicit or not |
| 10 | danceability | How suitable a track is for dancing |
| 11 | energy | How intense and active a track is |
| 12 | key | Overall key of the track |
| 13 | loudness | Overall loudness of the track, in decibels (dB) |
| 14 | mode | Whether the track is in major mode (1) or minor (0) |
| 15 | speechiness | Proportion of spoken words in the track |
| 16 | acusticness | Confidence measure of whether a track is acoustic |
| 17 | instrumentalness | Proportion of instrumental parts in a track |
| 18 | liveness | Detects live audience in a track. Represents the probability that a track was performed live |
| 19 | valence | Measures how positive a track sounds, from 1 (extremely positive) to 0 (extremely negative) |
| 20 | tempo | Overall tempo of a track, in beats per minute (BPM) |
| 21 | duration\_ms | Duration of a track, in milliseconds (ms) |
| 22 | time\_signature | Overall time signature of a track |
| 23 | year | Release date of a track |
| 24 | release\_date | Full release date of a track, usually in YYYY-MM-DD format |

**Background:**

Music recommendation systems are becoming increasingly important as the amount of available music continues to grow. Spotify, one of the largest music streaming services in the world, has over 1.2 million songs in its catalog. Recommending songs to users based on their listening history and preferences is crucial to retaining users and increasing engagement. Additionally, playlist recommendation systems are essential for creating a personalized listening experience for users.

**Objective and Tasks:**

The main objective of this project is to explore novel approaches to solving the music recommendation and playlist recommendation problems for the Spotify 1.2M+ songs dataset. Specifically, we aim to improve the accuracy and diversity of song recommendations and create more personalized playlist recommendations for users.

We are planning to use following methodologies to solve the recommendations problem of our metadata:

1. Knowledge Graph Based Recommendation System: A knowledge graph is a graph database that represents entities, relationships, and attributes of real-world objects or concepts. In the context of music recommendation systems, a knowledge graph can be used to represent musical artists, songs, albums, genres, user listening history, and other relevant metadata. A knowledge graph-based recommendation system uses this graph database to model the relationships between different entities and generate personalized recommendations for users.One approach to implementing a knowledge graph-based recommendation system is to use graph convolutional networks (GCNs).
2. Hybrid recommendation System: Hybrid recommender systems combine multiple recommendation algorithms to generate personalized recommendations for users. In the context of music recommendation systems, hybrid systems can combine content-based filtering, collaborative filtering, and knowledge graph-based approaches to improve the accuracy and efficiency of the recommendations.
3. ANNOY in combination with Matrix Factorization: ANNOY (Approximate Nearest Neighbors Oh Yeah) is a library that provides an efficient way to search for nearest neighbors in high-dimensional data, such as music metadata. ANNOY can be used as a preprocessing step to speed up the search for nearest neighbors in the metadata space, and matrix factorization can be used to generate personalized recommendations based on the user's listening history and preferences.
4. Sequential Recommendation for Automatic Playlist Continuation: This algorithm utilizes a deep neural network with which we can model a sequential recommender system for automatic playlist continuation based on the metadata of the song itself and independent of the artist. This method will allow for new and not so famous artists to feature in a playlist based on the type of song he/she has created.