Music Recommender System

1. Introduction

Currently, there are many music streaming services, like Pandora, Spotify, etc. which are working on building high-precision commercial music recommendation systems. These companies generate revenue by helping their customers discover relevant music and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good music recommendation systems.

Music recommender system is a system which learns from the users past listening history and recommends them songs which they would probably like to hear in future. I firstly implemented a popularity based model which was quite simple and intuitive. Collaborative filtering algorithms which predict (filtering) taste of a user by collecting preferences and tastes from many other users (collaborating) is also implemented.

2. Project Approach

Following algorithms were used to build an efficient music recommender system.

- Popularity based Model
- Collaborative based Model

2.1 Popularity based Model

This is one of the most naive approaches for song recommendation. In this algorithm we simply place each song in descending order of its

popularity. And then using recommender class we recommend the top 10 most popular songs to the user regardless of their listening history.

2.2 Collaborative based Model

Collaborative filtering involves collecting information from many users and then making predictions based on some similarity measures between users and between items. This can be classified into user-based and item based models. In the item-based model, it is assumed that songs that are often listened together by some users tend to be similar and are more likely to be listened together in future also by some other user. According to the user based similarity model , users who have similar listening histories, i.e., have listened to the same songs in the past and tend to have similar interests and will probably listen to the same songs in future too.

We need some similarity measure to compare between two songs or between two users. Cosine similarity weighs each of the users equally which is usually not the case. Users should be weighted less if they have shown interest in many varieties of items. Likewise, the user is weighted more if listens to a very limited set of songs.

3. Dataset

Dataset used for this project was having information about approximately 10,000 songs. "*song_df_1*" contains the song_id, user_id, listen_count. "*song_df_2*" contains the title, release year, artists name.

4. References

1. https://towardsdatascience.com/how-to-build-a-simple-song-recommender-296fcbc8c85

2. https://www.youtube.com/watch?v=PPLop4L2eGk&list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN

3. https://pandas.pydata.org/pandas-docs/stable/