A Mini Project Synopsis on

Music Player Recommendation System

T.E. - I.T Engineering

Submitted By

Shridhar Joshi	**19104034**
Bhimraj Parihar	**19104004**
Akshada Warik	**19104042**

Under The Guidance Of

Prof. Anagha Aher

Prof. Roshni Singh

DEPARTMENT OF INFORMATION TECHNOLOGY

A.P. SHAH INSTITUTE OF TECHNOLOGY
G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic year: 2021-22

CERTIFICATE

This to certify that the Mini Project report on Music Player Recommendation System has been submitted by Shridhar Joshi (19104034), Akshada Warik (19104042) and Bhimraj Parihar (19104004) who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **Information Technology**, during the academic year **2021-2022** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Prof. Anagha Aher
Guide

Prof. Roshni Singh
Co-guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)
1.
2.

Place: A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

This project would not have come to fruition without the invaluable help of our guide Prof. Anagha Aher and co-guide Prof. Roshni Singh. Expressing gratitude towards our HOD, Prof. Kiran Deshpande, and the Department of Information Technology for providing us with the opportunity as well as support required to pursue this project. We would also like to thank our teacher Prof. Nahid Shaikh who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

Chapter 1

Introduction

Rapid development of mobile devices and the internet has made it possible for us to access different music resources freely. The number of songs available exceeds the listening capacity of a single individual. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their customers to discover music by giving quality recommendations.

Research in music recommender systems (MRSs) has recently experienced a substantial gain in interest both in academia and in industry. Thanks to music streaming services like Spotify, Pandora, or Apple Music, music aficionados are nowadays given access to tens of millions music pieces. By filtering this abundance of music items, thereby limiting choice overload, MRSs are often very successful to suggest songs that fit their users' preferences. This is partly because of the fact that users' tastes and musical needs are highly dependent on a multitude of factors, which are not considered in sufficient depth in current MRS approaches, which are typically centered on the core concept of user–item interactions, or sometimes content-based item descriptors.

In contrast, we argue that satisfying the users' musical entertainment needs requires taking into account intrinsic, extrinsic, and contextual aspects of the listeners, as well as more decent interaction information. For instance, personality and emotional state of the listeners (intrinsic) as well as their activity (extrinsic) are known to influence musical tastes and needs. So are user's contextual factors including weather conditions, social surrounding, or places of interest.

1.1 Purpose

With the rise of digital content distribution, we need to have access to a huge music collection. With millions of songs to choose from, we sometimes feel overwhelmed. Thus, an efficient music recommender system is necessary in the interest of both music service providers and customers.

There is a need for large-scale and personalized recommendation system, which learns from user's listening history and features songs and predicts songs that a user would like to listen to.

We listen music when waking up, while in transit at work, and with our friends. For many, music is like a constant companion. It can bring us joy and motivate us, accompany us through difficult times and alleviate our worries. Hence music is much more than mere entertainment, but as stated earlier, growth in the amount of digital information have created a potential challenge of information overload where a recommendation engine plays a very crucial role in filtering the vital fragment out of large amount of dynamically generated information according to user’s preferences, interest or observed behaviour about item. Hence, with this project, we attempt to devise a method to improve the collaborative filtering engine in order to compute the similar users for the recommendation of music as it is believed, person with similar personality has similar taste in music.

1.2 Objectives

- To create a web-based music player that allows users to listen to their music libraries.
- To create search options for the ease of access.
- To create a user friendly and attractive layout.
- To recommend songs that a user is more likely to be interested in listening to.
- To learn more about user’s preferences and constraints.
- To help users become more aware of their preferences.
- To enhance the usage of the main music streaming application.
- To provide recommendations to new users for whom the system has very little information about their likes and dislikes.
- To provide users a free system this brings the opportunity of discovering new music.
- To develop a working system capable of making the most of free online services and save user’s time.
- To find out if the personality of the user can be a crucial factor in the music recommendation system.
- To find out if recommendation engines can be enhanced.

1.3 Scope:

- To be useful for people who want ad-free experience using search and sort options.
- To serve as a free platform for music lovers.

- To provide more features and functionalities, this will make the music player more attractive.
- To predict what the user could like without knowing the user.
- To provide an advanced yet handy tool with an easy-to-use interface.
- To shorten the time between data collection and data analysis.
- To utilize resources in an efficient manner by increasing their productivity through recommendation.
- To make a music player that contributes hugely in saving time and enhances data quality.

- To discover if the personality traits of anindividual can be used for the enhancement of the recommendation engine in order to providethe more personalized content to the user as a recommendation.

Chapter 2

Problem Definition

Whilst easy access is definitely given in today's music provision, the issue is of finding good quality music. It's difficult for people to choose from millions of songs. Most of the time is gone in searching the songs for that particular moment of time. Therefore, our Music recommender system is a system which learns from the user’s past listening history and recommends them songs which they would probably like to hear in future.

One of the major problems of recommender systems in general and music recommender systems in particular is the *cold start* problem, i.e., when a new user registers to the system or a new item is added to the catalogue and the system does not have sufficient data associated with these items/users. In such a case, the system cannot properly recommend existing items to a new user (*new user* problem) or recommend a new item to the existing users (*new item* problem).

Chapter 3

Proposed System

The aim of the project is to develop a system of improved facilities. The system provides proper recommendation and reduces the manual work.

The main objective of this work is to develop an application for music recommendations. The application allows users to select and listen to the songs available in the device. Whenever a user listens to a particular song, a log is created. In order to suggest songs to the users, we use various strategies to implement recommendation engine. The main motive of this Proposed System is extending the capabilities of the traditional recommendation System.

The system is being built to:

- Minimize manual data entry.
- Minimum time needed for the various processes.
- Greater Efficiency.
- Better service.

3.1 Feature and Functionality

The following features and functionality will provide the user a better and smooth experience:

Search option and easy filters:

- We have provided search options and filters for the user which makes it easy to find the required music using the search option.

User-friendly GUI and easy to understand functions:

- Pause/Play button- control the flow of the song.
- Timeline Slider: instantly play the song instance that sounds pleasing to your ears.

- Next/Previous: Enables the user to traverse to the next/previous song.
- Playlist: A list of all the songs will be displayed here. The user can collapse/expand the playlist according to his/her choice.
- Looping: Loop the playlist, put your favourite song on repeat or party shuffle your playlist. All of this on click of a button.

Recommendation Engine:

- The engine recommends songs to user based on popularity and song history.
- We have built two types of recommendation engines: Popularity-based recommendation and Similarity-based recommendation.

Chapter 4

Project Outcomes

This Final Project aimed to create a practical and useable product, which can be considered as software and even an educational tool.

- To discover music by giving quality recommendations.
- Implementation of machine learning algorithms.
- Users can listen to music 24/7 at the tip of their hands.
- Users can listen to music without any interruptions.
- Easy Interface & Easy to Manage.
- Can identify cross-genre items.
- Domain knowledge about item is not needed.
- Adaptive: quality improves over time.

Chapter 5

Software Requirement

- Browser – Chrome or any modern browser
- HTML/CSS – up-to-date browsers for HTML5 & CSS3
- JavaScript – ECMAScript 2015 or above
- Python 3+
- Python libraries -
  - Pandas
  - NumPy
- JavaScript libraries –
  - music-list.js
  - script.js

Chapter 6

Project Design

The design of our project i.e., the overview of the system, flowchart, algorithms, components and ideas used to build the project are explained below in brief:

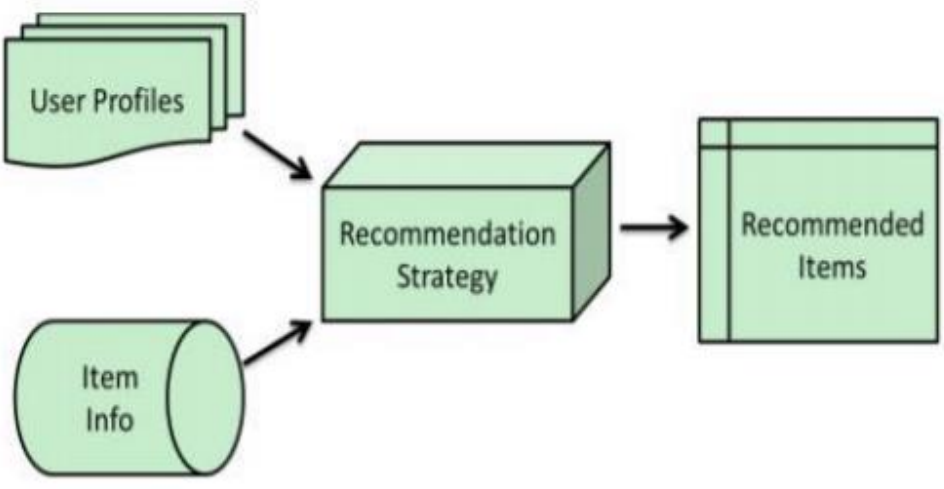


Figure 6.1 Overview of the system

The dataset that we have chosen for this particular project is a subset of “Million songs Dataset”. The dataset is divided into 2 parts: one contains the song information, while the other maps the user id to the songs and the number of times a particular song is being played. All of this information is further passed on to the recommendation engine which applies certain algorithms like Popularity based recommendation, Content-Based Filtering and Collaborative Filtering to recommend the songs which might be suitable for that particular user.

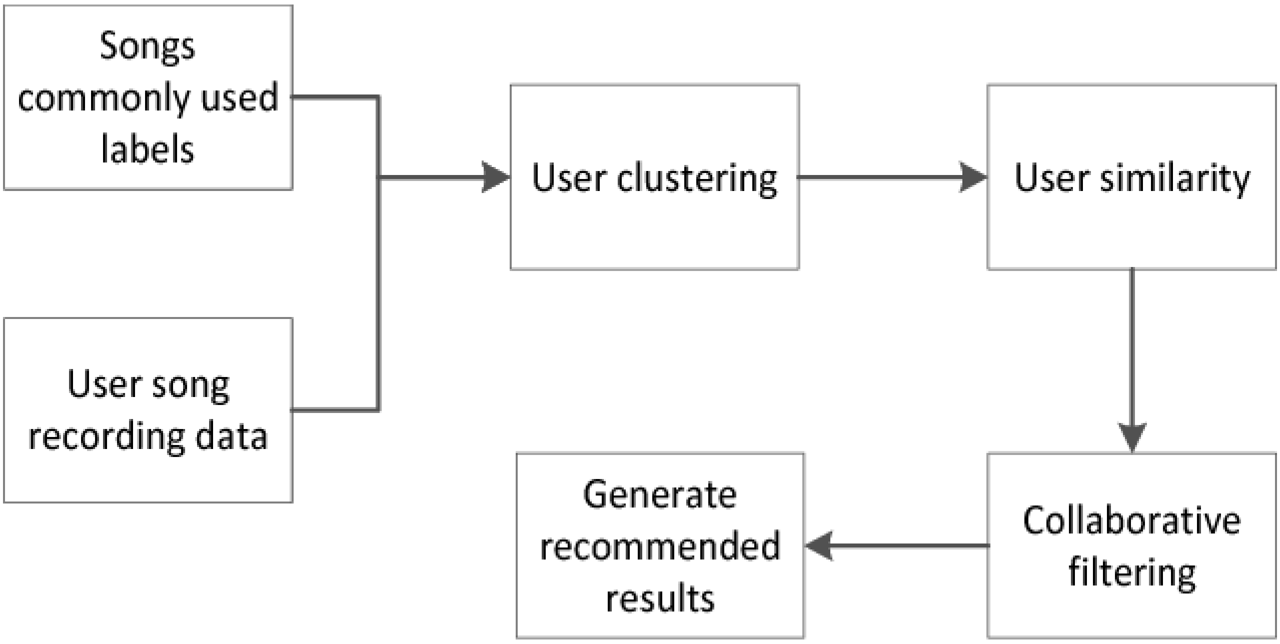
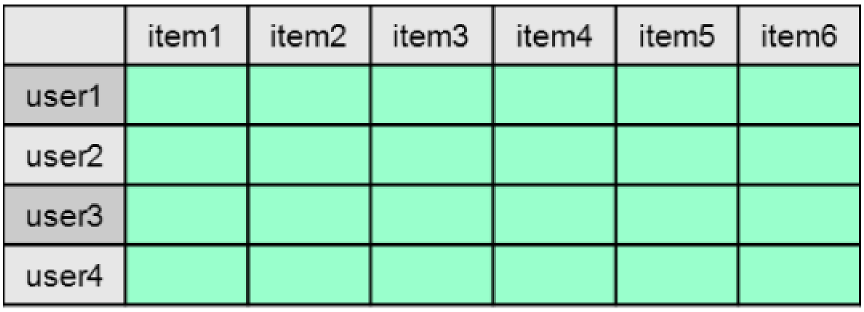


Figure 6.2 Flowchart of Algorithm

The “song_data.csv” file contains metadata of each song like the song_id, title, release, artist_name and year. On the other hand, the file “triplets_file.csv” contains attributes like user_id, song_id and listen_count. Both of these data are combined and preprocessed. Each song is assigned a score value and a rank which proves to be helpful for Popularity-based and Similarity-based recommendations.



	item1	item2	item3	item4	item5	item6
user1						
user2						
user3						
user4						

Figure 6.3 Co-occurrence Matrix

The co-occurrence matrix is the resultant matrix which contains the information regarding the number of unique songs per user as well as the number of unique users per song. This information is crucial for allocating the rank and the recommendation score for each song on the basis of Similarity-based recommendation.

This determines how individual scoring components affects the overall scoring between two items. The similar things are emphasized more while less similar one's contribution drop down to zero.

Collaborative Filtering:

Collaborative filtering is the predictive process behind recommendation engines. Recommendation engines analyse information about users with similar tastes to assess the probability that a target individual will enjoy something, such as a video, a book or a product. Collaborative filtering is also known as social filtering.

Collaborative filtering uses algorithms to filter data from user reviews to make personalized recommendations for users with similar preferences. Collaborative filtering is also used to select content and advertising for individuals on social media.

Three types of collaborative filtering commonly used in recommendation systems are neighbour-based, item-to-item and classification- based.

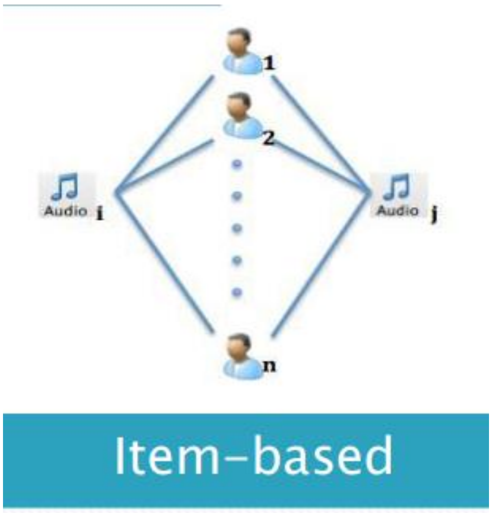


Figure 6.4 Item Based Model

In 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 user-based similarity model, users who have similar listening histories, i.e., have listened to the same songs in the past tend to have similar interests and will probably listen to the same songs in future too.

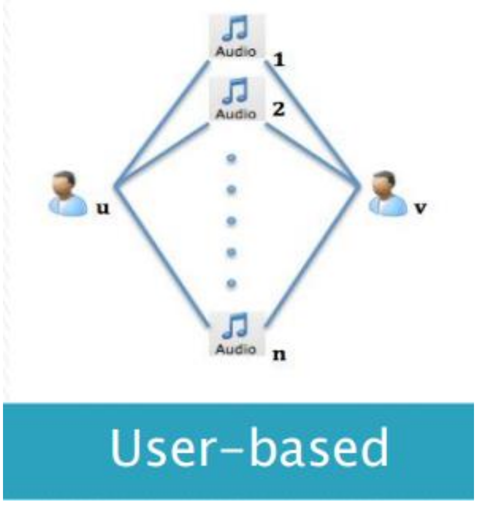


Figure 6.5 User Based Model

Users who listen to the same songs in the past tend to have similar interests and will probably listen to the same songs in future.

Content-Based Filtering:

Content-based filtering is a type of recommender system that attempts to guess what a user may like based on that user's activity.

Content-based filtering makes recommendations by using keywords and attributes assigned to objects in a database (e.g., items in an online marketplace) and matching them to a user profile. The user profile is created based on data derived from a user's actions, such as purchases, ratings (likes and dislikes), downloads, items searched for on a website and/or placed in a cart, and clicks on product links.

The content-based recommendation consists of the analysis of the content of the item's candidates for recommendation. This approach aims to infer the user's preferences in order to recommend items that are similar in content to items they have previously liked. This method does not need any feedback of the listener, it is only based on sound similarity which is deduced from the features extracted from the previous listened songs. This method is based on the similarities between the different items. To estimate similarities, it is a matter of extracting features to best describe the music. The Machine Learning algorithm then recommends the closest item to those that the user already likes.

Chapter 7

Project Scheduling

Group Member	Time	Work done
Everyone	2nd week of February	Literature Review, reading papers, Designing UI and choosing approach.
Bhimraj Parihar	4th week of February	Creating the front-end with required options of pause, play, next etc.
Akshada Warik	1st week of March	Choosing the Dataset and implementation of pre-processing.
Shridhar Joshi	3rd week of March	Implementation of algorithms and creating the recommendation engine
Everyone	4th week of March	Connecting the front-end to the recommendation engine.
Everyone	2nd week of April	Testing and fixing errors and making required changes. Creation of Report and Presentation.

Chapter 8

Implementation

The following was the output after the successful implementation of the project. The following figures show and explain the GUI, Popularity-based recommendation and Similarity-based recommendation in brief:

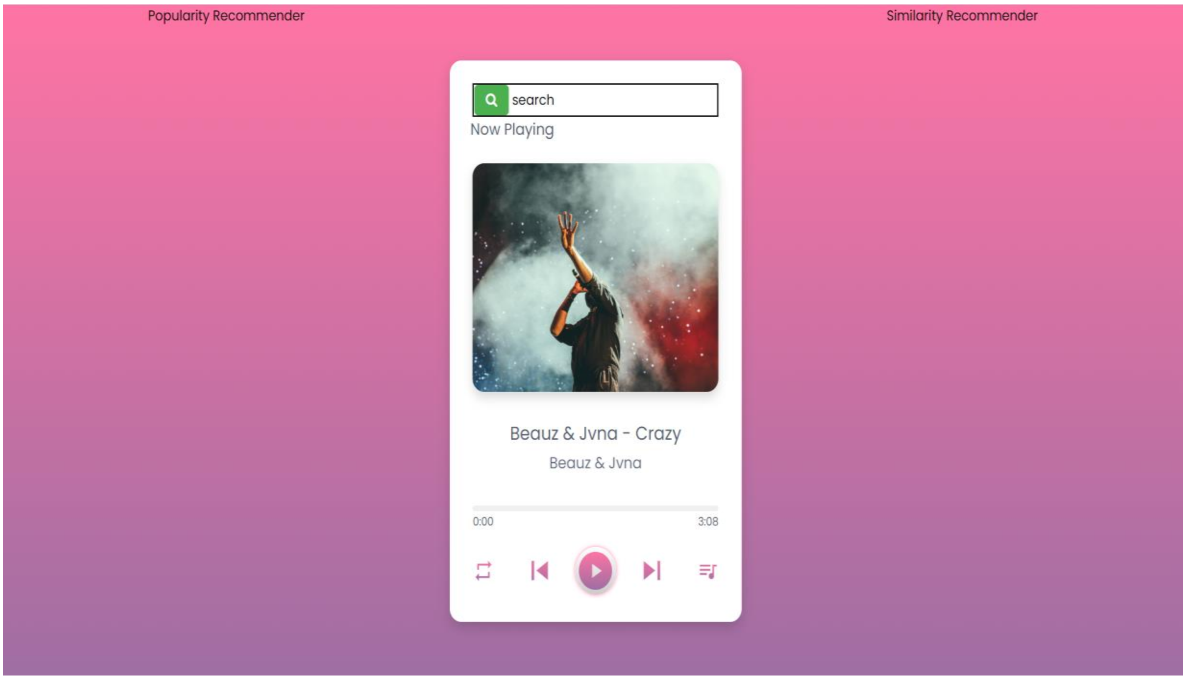


Figure 8.1 GUI

Since this is a single page application, the home page itself acts as the GUI with the music player aligned at the center of the screen. On the left-hand side of the player, you will be able to see popularity-based recommendations for the user. On the other hand, i.e., right-hand side of the player, you will get similarity-based recommendations.

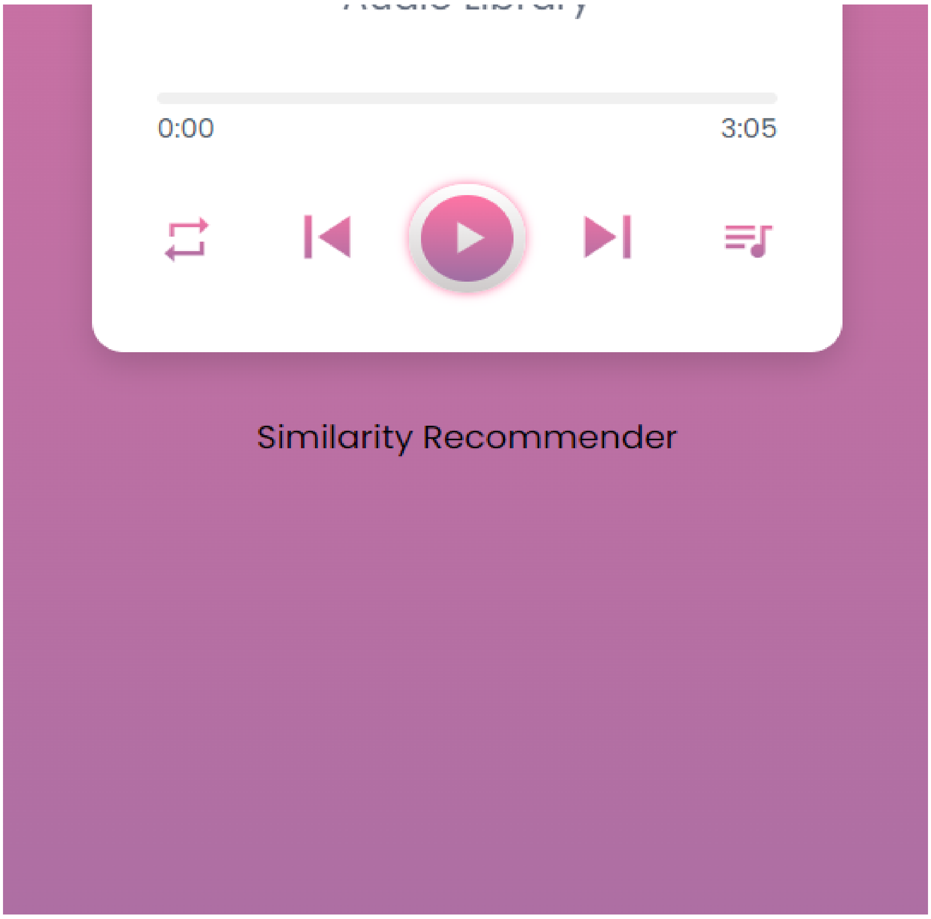
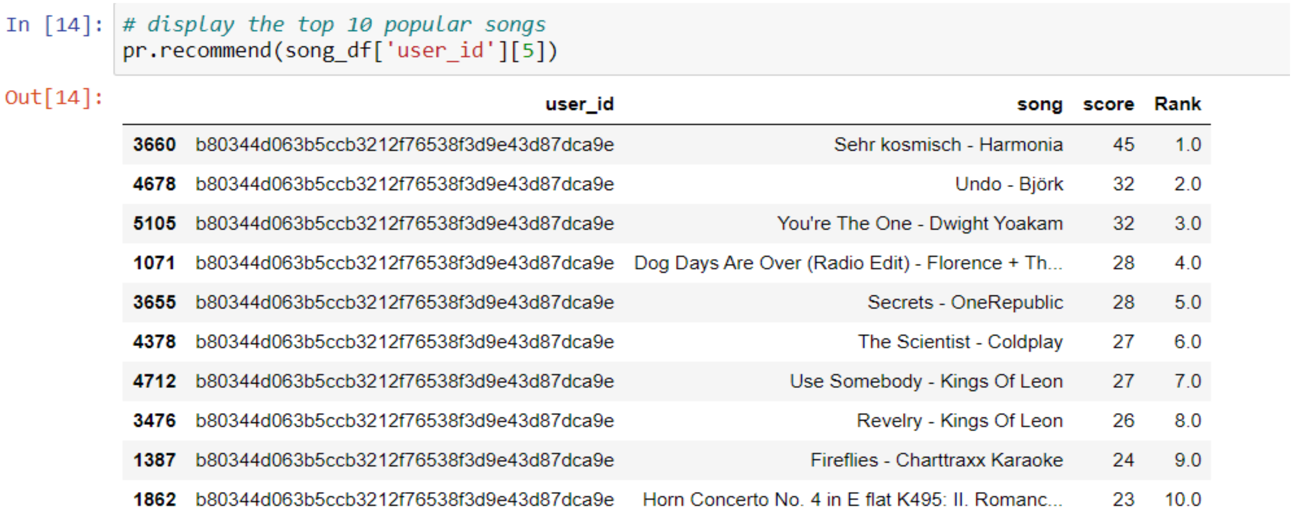


Figure 8.2 Responsiveness of Website

Also, the website is responsive, so if the screen size is reduced below a certain value, the Popularity-based recommendations, the music player and the similarity-based recommendations get stacked upon each other.



```
In [14]: # display the top 10 popular songs
         pr.recommend(song_df['user_id'][5])
```

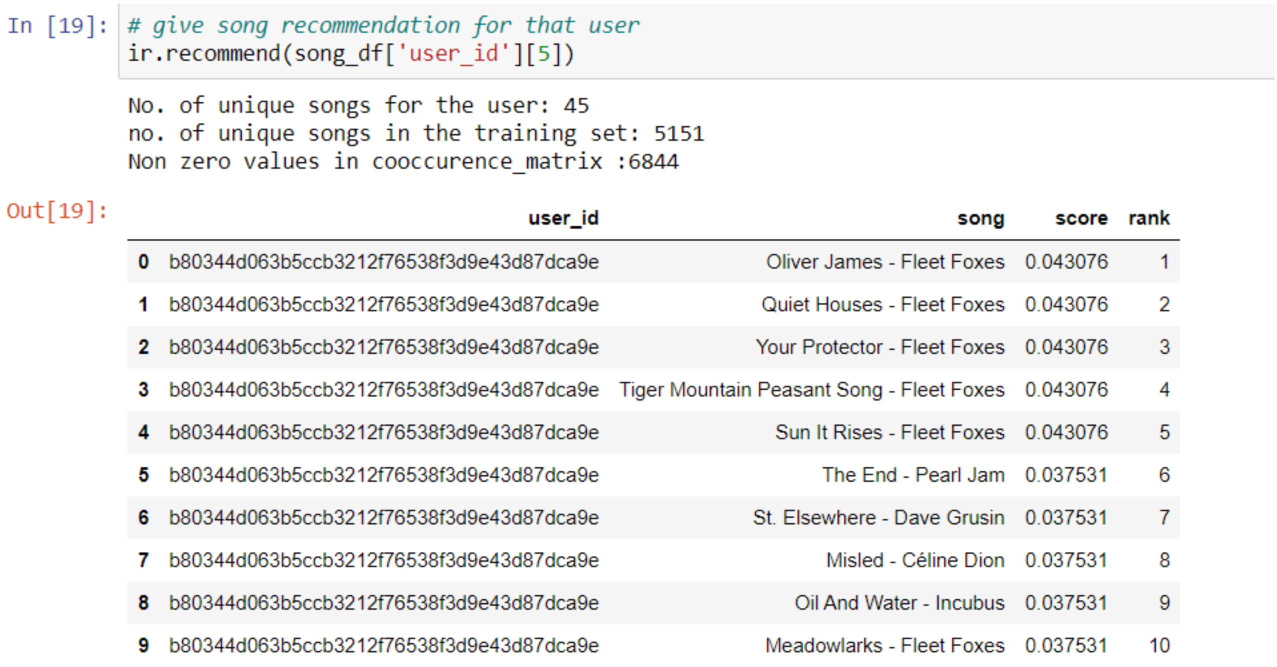
Out[14]:

	user_id	song	score	Rank
3660	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Sehr kosmisch - Harmonia	45	1.0
4678	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Undo - Björk	32	2.0
5105	b80344d063b5ccb3212f76538f3d9e43d87dca9e	You're The One - Dwight Yoakam	32	3.0
1071	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Dog Days Are Over (Radio Edit) - Florence + Th...	28	4.0
3655	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Secrets - OneRepublic	28	5.0
4378	b80344d063b5ccb3212f76538f3d9e43d87dca9e	The Scientist - Coldplay	27	6.0
4712	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Use Somebody - Kings Of Leon	27	7.0
3476	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Revelry - Kings Of Leon	26	8.0
1387	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Fireflies - Charttraxx Karaoke	24	9.0
1862	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Horn Concerto No. 4 in E flat K495: II. Romanc...	23	10.0

Figure 8.3 Popularity Based Recommendation

The popularity-based recommendation engine considers the aggregate of the count for each song. This helps us decide which song is played the most and hence, most popular. The song with the highest count gets rank 1 and so on the top 10 popular songs are displayed.

The recommendation engine sorts songs by popularity in a decreasing order. For each user, it recommends the songs in order of popularity, except those already in the user's profile



```
In [19]: # give song recommendation for that user
         ir.recommend(song_df['user_id'][5])

No. of unique songs for the user: 45
no. of unique songs in the training set: 5151
Non zero values in cooccurence_matrix :6844
```

Out[19]:

	user_id	song	score	rank
0	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Oliver James - Fleet Foxes	0.043076	1
1	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Quiet Houses - Fleet Foxes	0.043076	2
2	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Your Protector - Fleet Foxes	0.043076	3
3	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Tiger Mountain Peasant Song - Fleet Foxes	0.043076	4
4	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Sun It Rises - Fleet Foxes	0.043076	5
5	b80344d063b5ccb3212f76538f3d9e43d87dca9e	The End - Pearl Jam	0.037531	6
6	b80344d063b5ccb3212f76538f3d9e43d87dca9e	St. Elsewhere - Dave Grusin	0.037531	7
7	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Misled - Céline Dion	0.037531	8
8	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Oil And Water - Incubus	0.037531	9
9	b80344d063b5ccb3212f76538f3d9e43d87dca9e	Meadowlarks - Fleet Foxes	0.037531	10

Figure 6.4 Similarity Based Recommendation

For the Similarity-based recommendation engine, a score and a rank are allocated similar to Popularity-based recommendation engine. After that a cooccurrence matrix is generated which tell us regarding the number of unique songs per user as well as the number of unique users per song. Based on this information, a similarity-based recommendation is generated which is unique for each user based on his/her song history.

Chapter 9

Conclusion

During the course of this project, we were able to make a music recommendation system using a hybrid approach of collaborative and content filtering using Popularity based and Item Similarity based recommendation system. In our project, we have tried to briefly describe the various types of recommendation techniques. We also discuss the feedback techniques for recommender system. In the future, we will try to add a greater number of artists and languages which will make the recommendation stronger giving even better playlists for the users. This is a project of our Machine Learning course. There are many different approaches to this problem and we get to know some algorithms in detail. By manipulating the dataset, changing the learning set and testing set, changing some parameters of the problem and analysing the result, we learn a lot of practicing skills. We've faced a lot of problems in dealing with this huge dataset, how to explore it in a better way and we also had difficulties in some programming details. However, with a lot of effort, we have overcome all of these.

In the future, we will try to add a greater number of artists and languages which will make the recommendation stronger giving even better playlists for the users. We can try the system with other machine learning models as well to compare the results and look for better results. When there are millions of songs out there, our motive was to give the users their preference of songs which they want to listen to and we are satisfied after getting one step closer to it. For future applications, an emotional detector system that will recommend the songs by recognizing our facial emotion can be developed.

References

[1] Kathavate, Sheela. "Music Recommendation System using Content and Collaborative Filtering Methods." *International Journal of Engineering Research & Technology (IJERT)* 10.02 (2021): 167-171.

[2] Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

[3] Garg, Shefali, and S. U. N. Fangyan. "Music Recommender System CS365: Artificial Intelligence." (2014): 1-6.

[4] Spotify 2006, accessed 9th February 2022,

< https://open.spotify.com>

[5] YouTube Music 2015, accessed 10th February 2022,

< https://music.youtube.com>

[6] Amazon Music 2007, accessed 10th February 2022,

< https://music.amazon.in/>

[7] Chemeque Rabel, Marine. "Content-based music recommendation system: A comparison of supervised Machine Learning models and music features." (2020).