User Personalized Music Recommendation System

- Ashwin Kumar A, Yuva Daranee P, Sailesh V

Abstract : In this project, we offer a Personalised Music Recommendation System for a satisfying musical experience. With the use of machine learning methods and methodologies, we were able to use this project to learn how to examine historical data to forecast consumer interest. This project will be useful for the expansion of music software by suggesting to users tracks that are more likely to draw new listeners.

For providing personalised music recommendations the following algorithms are used :

- ✓ ***Popularity based filtering***
- ✓ ***Item similarity based Collaborative Filtering***
- ✓ ***Singular Value Decomposition algorithm***

Three machine learning techniques are used to suggest songs that users would most likely enjoy. User listen count is used in our algorithm to generate implicit ratings, which serve as the basis for the suggestions.

The top 5 songs from each of the three methods are taken as per ranking to form a hybrid recommendation list which covers music suitable to the user in all forms thus providing a more accurate recommendation result. Even if the user is new, the login data is used to identify a similar user from the database and make recommendations that are relevant to that user.

Our Dataset is much or less based and augmented from the Spotify released Million Track Dataset that has been published and licensed to Kaggle for research purposes.

Keyword : popularity, recommendation, sound name, prediction, listen, dataset, filter, trend, Singular Value Decomposition, Item Similarity Based Collaborative Filtering.

I. INTRODUCTION

Recommendations systems are programs based on machine and deep learning algorithms that give suggestions on products for users based on their preferences and interest. They provide recommendations based on various criteria including past purchases, search history, demographic information and other factors. Whether we realize it or not, a number of recommendation systems have recently incorporated themselves into our daily lives. Popular music software's generates a new customized playlist for each subscriber which is a personalized list of 30 songs based on users' unique music taste.

In the existing system the popular songs in the trend are recommended for new users which may not be personalized for them. So, we created an implicit recommendation system for the new users. Personalized music recommendation is crucial as it creates a bond between the user and the song listening platform that all of their favourite songs are just a touch away. User's rating is not always available and missed out by the user in some cases. Therefore, using user ratings for recommendation may not result in an accurate personalization. So, we use the user-song listen count and generate an implicit rating for every song, according to which the recommendations are given to the user.

In existing papers, the models recommend popular songs at the current trend for the new users. Those systems receive information from users like favourite artists, favourite genre and then recommend the popular songs of that artist or genre. Thus, personalization is not achieved in this model. So, we propose some other methods of recommendation in addition to it for optimal user personalization.

In our recommendation approach, we identify similar users for the new user based on the data they provided, such as their age, gender, etc., and provide song recommendations based on the similar users' preferred musical genres. The similar users are already using the software and have their preference data stored in the database. The recommendations based on that will result in a better personalization.

In the existing systems the user would not have rated any songs, so providing personalized recommendations is not possible. In our model, the user-song relationship is identified via the listen count of the songs. The relationship matrix is utilized for the generation of an implicit rating for the song. Using these implicit ratings, the most preferred songs for the new user are identified and recommended.

The objective of our project is to recommend personalized songs for the new users which are going to be accurate to their personal interest and preference. This recommendation system is proposed to use the user's data like age, gender etc given by them during login to generate the recommendation for the new login. This project is to recommend songs with more accuracy to all users using many algorithms which are proposed only for personalization. To recommend without using explicit data given by the users like rating, likes and dislikes, since the users are not always willing to rate or like the songs.

Basic workflow for recommendation system:

• Collection- Data collected can be explicit (ratings and comments on products) or implicit (page views, order history, etc.).

• Storing- The type of data used to create recommendations can help you decide the kind of storage you should use- NoSQL database, object storage, or standard SQL database.

• Analyzing- The recommender system finds items with similar user engagement data after analysis.

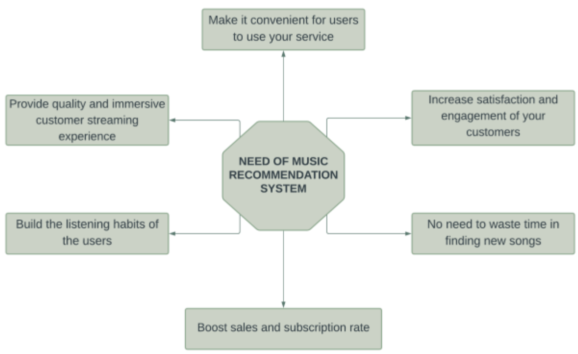
• Filtering- This is the last step where data gets filtered to access the relevant information required to provide recommendations to the user. To enable this, you will need to choose an algorithm suiting the recommendation system.

Music Recommendation System

Recommendation system is a filtering system, the purpose of which is to predict the preference that a user would give to a particular element, in our case – to a song. It is a core of huge engines that work by certain recommender algorithms and suggest a single item or a set of items to users based on such predictions.

Whether we are aware of it or not, a variety of recommendation systems have become an integral part of our daily routine since recently. Starting from accurately targeted advertising product suggestions and finishing with personalized video or music playlists compiled specifically for us – recommendation systems seem to be encompassing our everyday lives from literally every corner of digital space.

In the music industry, recommendation systems are part of a big engine of streaming apps like Spotify, YouTube Music, Deezer, Tidal, and the like.



II. RELATED WORK

The survey will contain existing recommendation models with some limitations they have and the solutions for those limitations proposed in our project.

Keshetti Sreekala, February 2020, International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 –

8958, Volume-9 Issue-3, [1] in their paper Popularity Based Recommendation System: A Recommendation engine recommends the most relevant items to the user by using different algorithms to filter the data. A Recommendation system is more useful in the context of data extraction relating to applications of big data and machine learning. As the name indicates the Popularity based recommendation system works with the current vogue. It basically uses the items which are in swing at present. This is the most basic recommendation system which provides generalised recommendation to every user depending on the popularity. Whatever is more popular among the general public that is more likely to be recommended to new customers.

In this paper I am going to use a class that we created which includes the methods to create recommendations and to recommend the item to the user. Next, I will load the data of Comma Separated Value (CSV). After that, sort the sound name based on how many users have listened to the sound name. After the collection of data code splits the dataset into training and the test dataset using 80–20 ratio. This creates an instance of popularity-based recommender's class. At last, I will use the popularity model to make the predictions.

The majority of the time, new listeners prefer whatever is more well-liked by the wider audience. Therefore, proposing well-known music may be accurate for the preferences of a new user.

Popular music recommendations are made using this method, which is not user-specific. Since there is no requirement that users just choose well-known music, Therefore, this approach lacks personalization.

Yading Song, Simon Dixon, and Marcus Pearce, 19-22 June 2012, 9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012), Queen Mary University of London, [2] in their paper A Survey of Music Recommendation Systems and Future Perspectives: Along with the rapid expansion of digital music formats, managing and searching for songs has become significant. Though music information retrieval (MIR) techniques have been made successfully in the last ten years, the development of music recommender systems is still at a very early stage. Therefore, this paper surveys a general framework and state-of-art approaches in recommending music. Two popular algorithms: collaborative filtering (CF) and content-based model (CBM), have been found to perform well.

Two user-centric approaches: context-based model and emotion-based model, have been paying increasing attention. In this paper, three key components in music recommender - user modelling, item profiling, and match algorithms are discussed. it is also time consuming to maintain the increasing metadata. Six recommendation models and four potential issues towards user experience, are explained. However, the subjective music recommendation system has not been fully investigated. To this end, we propose a motivation-based model using the empirical studies of human behaviour, sports education, music psychology.

By considering affective and social information, emotion-based model and context - based model largely, this paper improved the quality of recommendation.

Popularity bias - It recommends popular music to the new listeners which is not personalized since it's not necessary for listeners to only select well-known songs. Human effort - Since this algorithm only makes use of the rating feature, it fails when the user fails to rate the songs explicitly. Cold start – If a user newly logs into the software, we don't have access to their previously listened to songs. So, it is unable to suggest the user's preferred music without knowing any information about them.

Varsha Verma, Ninad Marathe, Parth Sanghavi, Dr. Prashant Nitnaware, from Maharashtra,India,November-December-2021, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, ISSN: 2456 – 3307, Volume-7 Issue-6, [3] in their paper Music Recommendation System Using

Machine Learning: In our project, we will be using a sample data set of songs to find correlations between users and songs so that a new song will be recommended to them based on their previous history. We will implement this project using libraries like NumPy, Pandas. We will also be using Cosine similarity along with Count Vectorizer. Along with this, a front end with flask that will show us the recommended songs when a specific song is processed.

This system will automatically search the music libraries, reducing down the time spent browsing and instantly suggesting appropriate songs to consumers. It finds correlations between users and songs so that a new song can be suggested to users based on their past listening habits.

For new users without a history, this algorithm will not suggest songs. It does this by using correlation coefficients between users and their prior history of music. This system won't suggest new genres that have a chance of being loved by people because it uses item-based filtering with music attributes.

Ramasuri Appalanidu C H, Ajay Kumar Badhan, Bhoomireddy Pushpa, Athukumsetti Jhansi Rani, Achanta Sai Dharani, Mudunuru Venkata Sai Sindhuja, July-2021, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 07, [4] in their paper Music Recommendation System with Advanced Classification: We describe a personalized music recommendation system using KNN and machine learning methods in this study. We present a collaborative filtering and content filtering recommendation algorithm to combine the output of the network with the log files to recommend music to the user in a personalized music recommendation system. The suggested system includes log files that store the previous history of the user's music playlist. The suggested music recommendation system pulls the user's history from the log file and provides music recommendations for each recommendation. Content-based approaches make suggestions based on the audio characteristics. The user's past song history is used to generate the recommendation, retaining the user's musical preferences and taste. Using the KNN algorithm to recommend songs that are similar results in recommendations that are more accurate.

As it works on the user's previous song history, it has a cold-start problem. For new users' accurate recommendations cannot be generated. As it finds similar songs for all songs in the user's logs, it has a lower time efficiency.

Karishma Mandloi, Amit Mittal, July 2018, International Journal of Computer Sciences and Engineering, Vol.-6, Issue-7, E-ISSN: 2347-2693, [5] in their paper Hybrid Music Recommendation System Using Content-based Filtering and K-Mean Clustering Algorithm: Data is recognized as an important source for knowledge generation. Sometimes users may be aware about requirements but sometimes may not. Recommender systems are software or technical facilities to provide item suggestions or predict customer preferences by using prior user information. Recommendations can help to increase sales and improve user satisfaction. Music Recommendation systems can help to explore relative music based on user preference or internal similarity. A hybrid recommender system is usually developed through the combination of multiple recommendation techniques to boost the quality of recommendations. This paper uses content-based filtering with K-mean clustering algorithm for music recommendation system which provides effective and relevant content to be suggested.

Retaining the user's musical preferences and taste, the recommendation is created using the user's prior song history. To make recommendations for songs that are related, the K means clustering algorithm produces more precise results.

As it works on K-means clustering, there are even more accurate methods available for providing recommendations with large volumes data like Singular Value

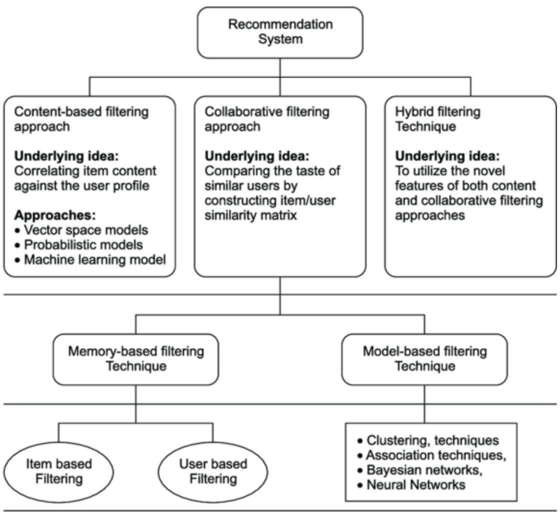
Decomposition. K means Clustering works better only on small volume data, and provides accurate results.

III. DESIGN AND WORKING

There are various types of recommendation models available:

- ➢ Collaborative Recommender system
- ➢ Content-based recommender system
- ➢ Demographic based recommender system
- ➢ Utility based recommender system
- ➢ Knowledge based recommender system
- ➢ Hybrid recommender system

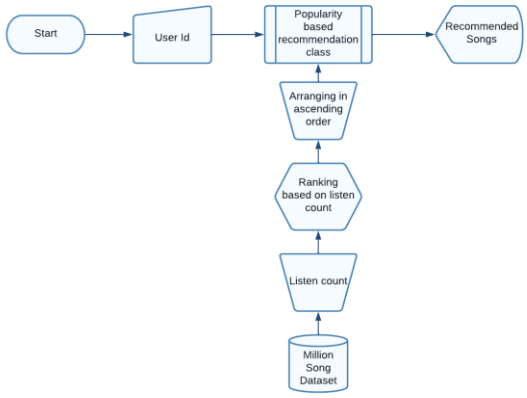
All these systems use their own efficient way of recommendation based on various criteria. As we have chosen music recommendation systems, we have applied Collaborative filtering and content-based filtering methods to form a hybrid recommendation engine.



Filtering approaches for recommendation system

3.1 Popularity-Based Recommendation Engine

The simplest recommendation engine is the Popularity-Based, that basically standing, if some item is liked by a vast majority of our user base, then it is a good idea to recommend that item to users who have not interacted with that item. The code to develop this kind of recommendation is extremely easy and is effectively just a summarization procedure that determines which items of the content have the most users and then that will become our standard recommendation set for each user. The recommendation list will be the same for all users. Since this is the naive approach, the recommendation may not be personalized.



3.2 Item Similarity Based Recommendation Engine:

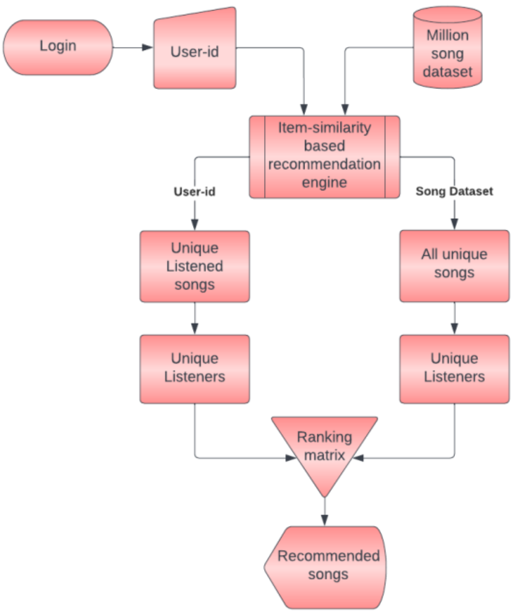
To provide a more personal recommendation to the user we need to apply a recommendation engine that considers some kind of similarities between users and the songs they have listened to. In other words, it is a recommendation engine based on calculating similarities between a user's items and the other items in our dataset. Usually, to define similarity among a set of items, we need a feature set on the basis of which both items can be described. In our case it will mean features of the songs on the basis of which one song can be differentiated from another.

Since our dataset doesn't have this data, we can use the Jaccard index to do an implicit similarity, based on common users, in terms of the users who listen to these songs. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

The basic idea remains that if two songs are being listened to by a large fraction of common users out of the total listeners, the two songs can be said to be similar to each other.

So, we need to calculate the similarity of each song in the user's list to those in our dataset, using the similarity metric defined previously. To make the computation more feasible, probably you use a cluster or, in our case, we limited our items to the most popular 5,000 songs so it is quite unlikely that we would miss out on any important recommendations.



3.3 Matrix factorization using Singular Value Decomposition algorithm:

Matrix factorization are methods that reduce a matrix into constituent parts, such that when these matrices are multiplied, we get the original matrix. It makes it easier to calculate more complex matrix operations. Matrix factorization methods, also called matrix decompositions methods, are a foundation of linear algebra in computers, even for basic operations such as solving systems of linear equations, calculating the inverse, and calculating the determinant of a matrix.

Matrix factorization can be used to discover latent features between two different kinds of entities. For example, we can try to explain a song in mathematical terms by measuring its beats, tempo, and other such features and then define similar features in terms of the user. Once we have consciously defined such "features", we can use them to find matches for a user based on some similarity criteria. You can use matrix factorization to discover these latent features and they seem to work great.

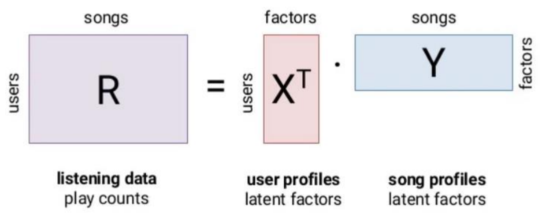
The starting point of any matrix factorization-based method is the utility matrix, a matrix of user Vs item dimension. No, this is a sparse matrix, since not all items are used by the user. The process of matrix factorization means finding out a low rank approximation of the utility matrix. So, we want to break down the utility matrix U into two low rank matrices so that we can recreate the matrix U by multiplying those two matrices:

Assuming the process helps us identify latent factors/features, meaning as K, our aim is to find two matrices X and Y such that their product (matrix multiplication) approximates R.

X = |U| x K matrix (A matrix with dimensions of num_users * factors)

Y = |P| x K matrix (A matrix with dimensions of factors * num_songs)

Model listening data as a product of latent features:



To make a recommendation to the user, we can multiply the corresponding user's row from the first matrix by the item matrix and determine the items from the row with maximum ratings. That will become our recommendations for the user. The first matrix represents the association between the users and the latent features, while the second matrix takes care of the associations between items (songs in our case) and the latent features.

Matrix Factorization and Singular Value Decomposition (SVD)

There are multiple algorithms available for determining factorization of any matrix. We use one of the simplest algorithms, which is the singular value decomposition or SVD. You can follow these steps to determine the factorization of a matrix using the output of the SVD function.

Factorize the matrix to obtain U, S, and V matrices.

Reduce the matrix S to first k components. (The function we are using will only provide k dimensions, so we can skip this step.)

Compute the square root of reduced matrix Sk to obtain the matrix Sk1/2.

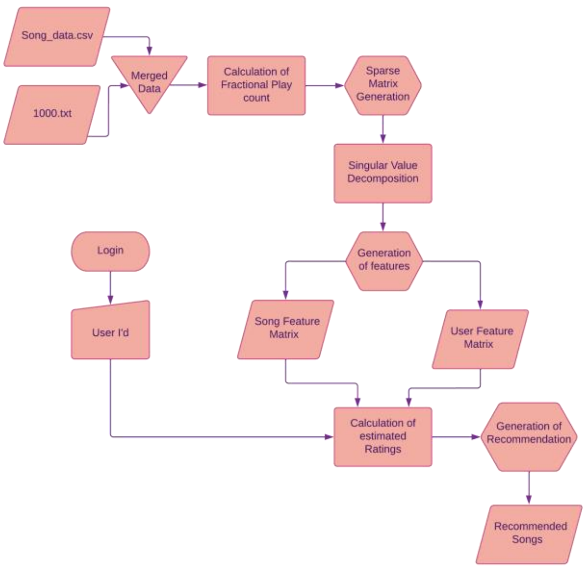
Compute the two resultant matrix U*Sk1/2 and Sk1/2*V as these will serve as our two factorized matrices

We can then generate the prediction of user i for product j by taking the dot product of the ith row of the first matrix with the jth column of the second matrix.

Let's create tree functions to help us on it:

compute_svd: It uses the svds function provided by the scipy library to break down our utility matrix into three different matrices, and proceed with the other three steps above.

compute_estimated_matrix: use the decomposed matrices by SVD and provide the predictions.



IV. EXPERIMENTAL SETUP AND RESULTS

In order to carry out the desired functionality I have used a system of i7 10th gen or above, 16Gb minimum, 5Gb of free space, VS code or Jupyter notebook, python 3.10.8 or below (some packages are not supported in latest versions of python) and packages Pandas, NumPy, Matplotlib.pyplot, Seaborn, Scipy, Random, Datetime, math.

SYSTEM ARCHITECTURE:

The following algorithms are used :

- Popularity based filtering
- Item similarity based Collaborative Filtering
- Singular Value Decomposition algorithm

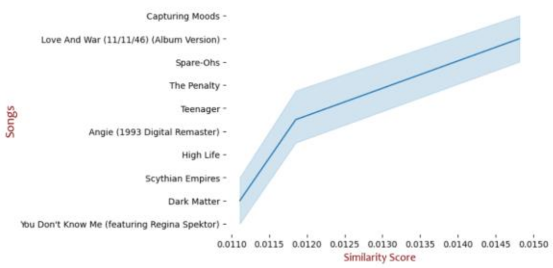
Three machine learning techniques are used to suggest songs that users would most likely enjoy. User listen count is used in our algorithm to generate implicit ratings, which serve as the basis for the suggestions.

The top 5 songs from each of the three methods are taken as per ranking to form a hybrid recommendation list which covers music suitable to the user in all forms thus providing a more accurate recommendation result. Even if the user is new, the login data is used to identify a similar user from the database and make recommendations that are relevant to that user.

OVERVIEW : Initially the users are asked for their date of birth for the identification of their age and their gender. This data is used for the classification of users based on their age and gender criteria. From our database the users similar to the new user's criteria are identified as similar users. A random user is selected from the similar users and the user's id is passed into the system for the recommendation of songs to the new user based on the data of the existing similar user.

Firstly, the user's I'd is passed into the initial model of recommendation, popularity-based filtering model. The model initially evaluates the popularity of songs based on their listen count in the system. Thus, the trending and most listened songs are identified and given as a recommended list. As we suppose the new user may like the song in the latest trend.

Secondly the user's I'd is passed into the second model of recommendation, item based collaborative filtering. To recommend the song suitable to the user, the matrix is formed between the songs listened by the similar user and the users who had listened to each and every song. The recommendation of songs for the new user is given based on the analysis of relationship matrix formed above.



Finally, the user's I'd is passed into the final model, matrix factorization using Singular Value Decomposition algorithm. It considers the user and the song interaction to form a sparse matrix. It decomposes the matrix into factor matrices for the identification using the Singular Value Decomposition method based on the latent features. Here the user feature matrix is multiplied with the song matrix for the generation of implicit ratings for the songs that the user has never rated.

Using the factor matrices, recommendations are provided. The songs are ranked according to those ratings and the top 10 songs are recommended.

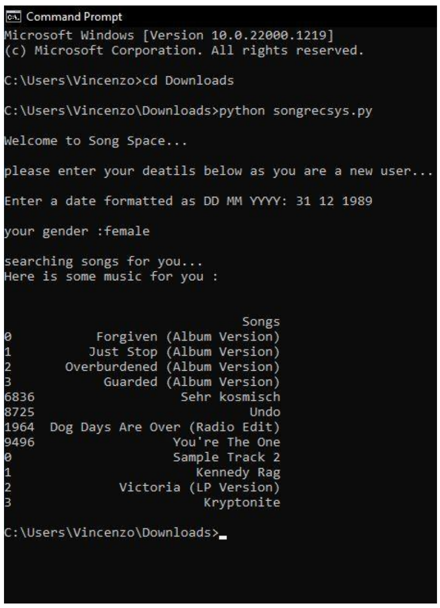
Now the recommendations of all the 3 models are taken into consideration. The top 3 songs of recommendations provided by each model are taken for the final recommendation list. So that the songs under the following 3 criteria are covered with high implicit ratings for the user personalized recommendations.

Popular songs in trend

Similar songs from comparable users

Songs predicted from implicit user rating

Thus, these nine songs provided to the user as recommendation works on the best of three methods. So, it is supposed to have advantages and improved accuracy over other existing models.



V. RESULT AND ANALYSIS

10.1 DATASET:

Our music recommendation engine uses the Million Song Dataset, a publicly accessible database of audio characteristics and metadata for one million recordings of current popular music. The Million Song Dataset (MSD) is a sizable dataset that includes the personal data of 2 million users. The songs are specimens of contemporary western pop music.

The reason for choosing this dataset is due to its vast song data of one million songs and diverse properties of about every aspect of a song. It contains numerous relationships among various means of comparison.

The subsets of the dataset song_data.csv and 1000.txt are used as input for our program model.

10.2 CONCLUSION:

This model uses the best of three models for providing the user with popularity, similarity and predicted based songs which are most probably preferred by the users. It can provide recommendations for both new users and existing users due to implementation of various recommendation systems by predicting the similar users using the login data. So, it outperforms any model that works on a single algorithm for yielding results due to its hybrid nature.

REFERENCES

[1]. Popularity Based Recommendation System - Keshetti Sreekala, International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-9 Issue-3, February, 2020

[2]. A Survey of Music Recommendation Systems and Future Perspectives - Yading Song, Simon Dixon, and Marcus Pearce, 9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012), 19-22 June 2012, Queen Mary University of London

[3].Music Recommendation System

-Smt. Namitha SJ, International Journal of Engineering Research & Technology, ISSN: 2278-0181 IJERTV8IS070267 Published by : www.ijert.org Vol. 8 Issue 07, July-2019

[4].Personality Based Music Recommendation System -Abhishek Paudel, Brihat Ratna Bajracharya, Miran Ghimire, Nabin Bhattarai

[5]. Music Recommendation System With Advanced Classification - Ramasuri Appalanidu C H, Ajay Kumar Badhan, Bhoomireddy Pushpa, Athukumsetti Jhansi Rani, Achanta Sai Dharani, Mudunuru Venkata Sai Sindhuja

[6]. Hybrid Music Recommendation System Using Content-based Filtering and K-Mean Clustering Algorithm - Karishma Mandloi, Amit Mittal

[7]. How to build a simple song recommender system - Eric Le

[8]. Music Recommendation System Project using Python - Neelesh Biswas

[9]. Million Song - Recommendation Engines - Marcelo Marques