Music Recommender System Using Machine Learning Algorithms

Vaibhav Varshney, Vikas Yadav

***Abstract*—In the past few years, digital music service applications like Spotify, YouTube Music, Gaana, and many other such applications have gained immense popularity. The primary reason behind the success of these applications is the way these provide recommendations to users.**

Music Recommendation Systems work on various algorithms and techniques that provide personalized song recommendations on the basis of the characteristics of songs heard by the user. This functionality helps users to discover new songs of their taste and thus improving user-application interaction.

Multiple techniques and algorithms can be used to provide valuable suggestions to users. This project aims to implement, analyse, and evaluate the Popularity and Collaborative Based Filtering Models that offer recommendations based on interactions between different users and songs

Keywords—Music Recommender, Popularity Based Filtering, Collaborative Filtering, Million Song Dataset, Data Pre-processing, User-User Based, Item-Item Based, Pearson Correlation, Jaccard Similarity

I. Introduction

Designing the recommendation systems is challenging as these systems are dynamic due to the indefiniteness of predictions they offer.

Also, the number of users of digital music service applications have increased tremendously in the recent past. Extraction of data and providing personalized recommendations is time-consuming, and thus the focus on fast processing of data has become a matter of interest nowadays. The performance, scalability, and quality issues can be handled using Popularity Based Filtering and Collaborative Filtering Techniques.

Popularity based recommendation system works based on the demand for songs. It utilizes the songs which are heard by most of the users.

Collaborative filtering concerning this project is an approach of automating suggestions based on the user's past interactions with songs.

Thus, the authors have tried to create an effective recommendation system that operates on various machine learning algorithms that learn users' preferences from its history using the data available in 'Million Song Dataset' and then provide the appropriate recommendations.

II. Related Work

The recommendation systems have played a crucial role in bringing an impactful change in digital music service applications.

A fair amount of study has been conducted and published in this field of learning. Here we are going to mention a few of those works.

In [1], the authors have worked on making a hybrid model using the content and collaborative based filtering models. In collaborative filtering they've worked on improving the recommendations using the cosine based similarity.

In [2], the researchers investigated the use of collaborative filtering techniques for a music recommender system and

computed a scoring function that combines the outcome by computing similarities amongst users and amongst items. Different methodologies have been used to calculate the similarity. Some of them are Euclidean distance, cosine metric, and Pearson correlation.

In [3], the authors have worked on evaluating the recommendation systems by taking out an arbitrary subset of discovered ratings and calculating the metrics of the system in replicating such ratings.

In [4], the researchers found out the resemblances amongst a couple of songs using features of audio and lyrics. The method used in this study targets multiple features of songs. Artificial Neural

Networks (ANN) and K-Nearest-Neighbour Regression were the major machine learning techniques used.

III. Experiment Design

The 'Million Song Dataset' [5] was chosen for performing this work. It is a freely available dataset, prepared by The EchoNest. It contains an aggregation of metadata for almost a million songs, and the user listens to history. The original dataset carries a size of 280 GB, and therefore to simplify this task, the authors have used a subset of this dataset, which contains 10000 songs.

The dataset has two sections; the first consists of a collection of song_id, title, release, artist_name, and year.
The second section of the dataset holds data on the history of songs heard by users. It stores user_id, song_id, and count of the number of times that particular user has listened to that specific song with a unique song_id.

Research Variables

The collective research variables available for carrying out this work are:-

1. song_id
2. title
3. release
4. artist_name
5. year
6. user_id
7. count

Data Pre-processing

1. *Data Integration*
   The two sections of the dataset are integrated together and grouped into a single data frame by performing inner join on the song_id attribute.
   After the successful integration, the data frame now contains 20,86,946 rows and 7 columns.
2. *Data Cleaning*

a) Duplicate rows were dropped from the dataset to decrease the chances of inaccurate prediction of recommendations by the system.

b) Songs having a year of release less than 1900 were removed from the dataset as they would have been of least interest to the users. Also, many songs were present, which had a year of release as 0. So to remove such outliers as well, this process was carried out.

The completion of data pre-processing resulted in the removal of redundant and irrelevant data, and after this process, the dataset consisted of 16,77,154 rows and 7 columns.
Thus data pre-processing resulted in the removal of over 400k unsuitable rows.

IV. Methods

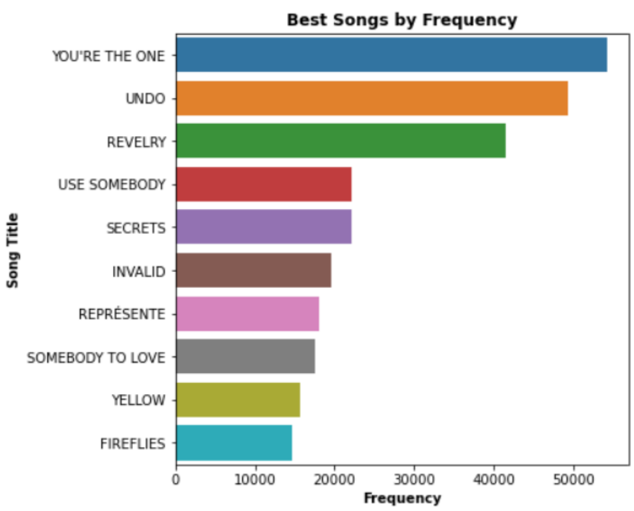
The authors have designed two models namely: Popularity Based Filtering and Collaborative Filtering Model so as to provide enhanced recommendations to users

A. Popularity Based Filtering

This kind of filtering is based on the concept of providing the most popular songs to the users. The popularity of a song is determined on the basis of various factors like the number of unique listeners of the song or the number of times that particular song has been heard.
All the popular songs are identified from the data available, and then relevant songs are suggested to the users.
For this study, the authors have classified the popularity into three types:-

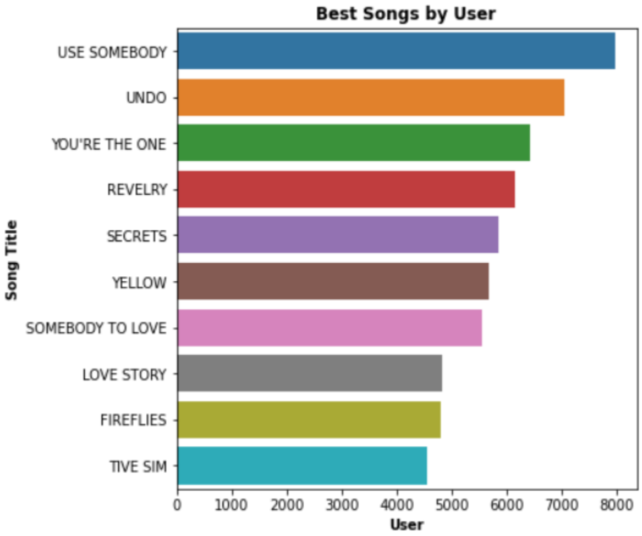
a) Frequency/Total Listen Count

In this type of popularity-based filtering, the popularity of songs has been established on account of the number of times a certain song has been heard.
The group by operation has been performed on the columns, namely song_id, title, and year to calculate the listen to count/frequency of every unique song.



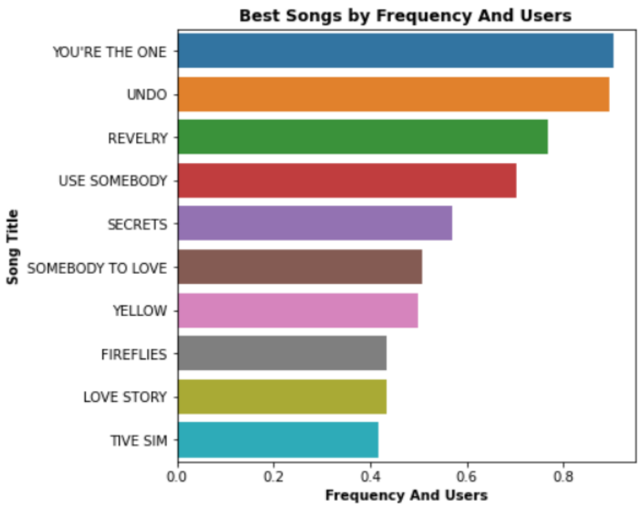
b) Unique Listeners of Song

In this type of popularity filtering, our focus is more on the unique listeners of songs rather than the listen count of songs. This factor has been taken into account so as to diversify recommendations.



c) Hybrid Popularity Filter

To ensure that suggestions offered by the filters have equal weightage for unique users and listen to count, a hybrid version of the above two popularity filters has been created. This filter considers equal preference for both the cases and offers composite suggestions to the users.



B. Collaborative Filtering

Collaborative based filtering procedures for music recommendation systems are methods that are entirely formulated upon the former interactions amongst the users and the songs. The collaborative recommenders use the historical details to predict the suggestions .These details can be fetched from the matrix (commonly known as pivot table) where the rows store the user_ids, and the columns store the song_ids.

It primarily gets the intuition of how much similar taste the users have or how much the songs are similar to each other.

There are multiple ways to find the similarity between the users or items, some of them are listed below:

1) Jaccard Similarity

The Jaccard Index, also recognized as the Jaccard similarity coefficient, is an indicator used in learning the similarities between sets of specimens. The measurement focuses majorly on finding the similarity between finite specimens, and is conventionally represented as the size of the intersection divided by the size of the union of the specimens.

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

The dilemma with this Jaccard similarity is that it neglects the values of the rating so songs. It just apprehends that how many songs in specimen A and specimen B are correlated; it doesn't take the likeness of the specimens into consideration.

2) Cosine Similarity

Cosine similarity is a method employed to estimate the similarity of the records irrespective of their capacity. Arithmetically, it calculates the cosine of the angle connecting two vectors outlined in a multi-dimensional scope.

The issue with the cosine similarity is that it treats the missing values in a vector as negative values (in terms of rating for example given 1-10 rating scale, it treats missing values as 0 which is considered highly negative).

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

3) Pearson Similarity

The Pearson correlation is nothing but the centred cosine similarity. In this, we are going to normalize values in a vector by subtracting the mean value of the vector. We have centred the values on zero which indicates the zero values as average rating and this overcomes the problem of cosine similarity, negative values indicate that it dislikes the song and positive values indicate that it likes the song. Afterwards we calculate the cosine similarity between the vectors and now it handles the easy and tough raters and also captures the intuition efficiently

$$r=\frac{\sum\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum\left(x_i-\bar{x}\right)^2\sum\left(y_i-\bar{y}\right)^2}}$$

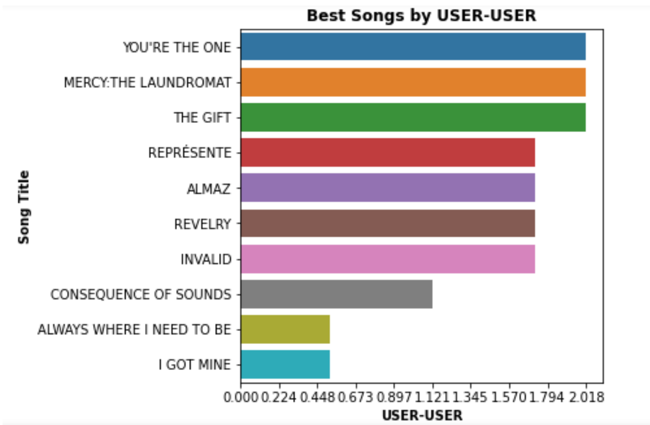
Collaborative Filtering can be divided into two categories on basis of interaction between users and songs.

Rating System
A function has been designed to compute ratings of songs on the grounds of the number of times the particular song has been heard by that user.

```python
def listen_count_to_rating(listen_count):
    if listen_count == 1:
        return 1
    elif listen_count == 2:
        return 2
    elif listen_count == 3:
        return 3
    elif listen_count == 4:
        return 4
    elif listen_count >4 and listen_count<=7:
        return 5
    elif listen_count >7 and listen_count<=10:
        return 6
    elif listen_count >10 and listen_count<=14:
        return 7
    elif listen_count >14 and listen_count<=19:
        return 8
    elif listen_count >19 and listen_count<=45:
        return 9
    elif listen_count >60:
        return 10
```

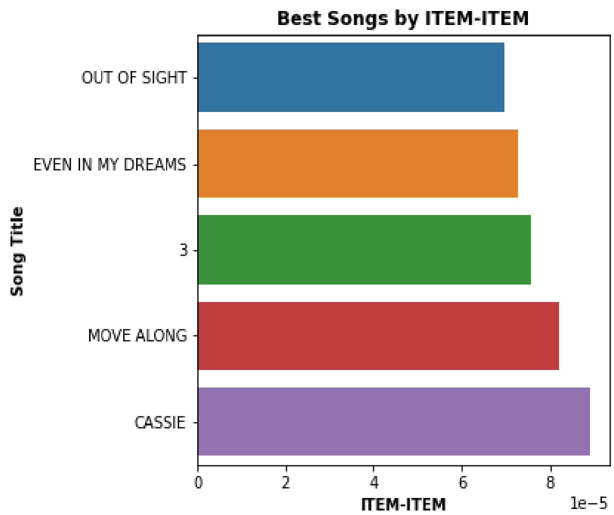
a) User-User Based
In this type of collaborative filtering, our work is to find the group of similar users whose likes and dislikes are similar to a particular user for which we want to recommend items, or we can say the group has the identical taste. This type of collaborative filtering uses similarities between users to provide recommendations. This allows for promising recommendations; that is, this collaborative filtering model can be used to recommend an item to user 'A' based on the interests of a similar user 'B'.
The authors have used Pearson correlation to compute the similarity between the users. And using K-Nearest-Neighbour algorithm they have formed the group of similar users corresponding to a user for which we need to recommend.

Recommendation to the user: db34291279818bc40973327069f41f6a022a9b86

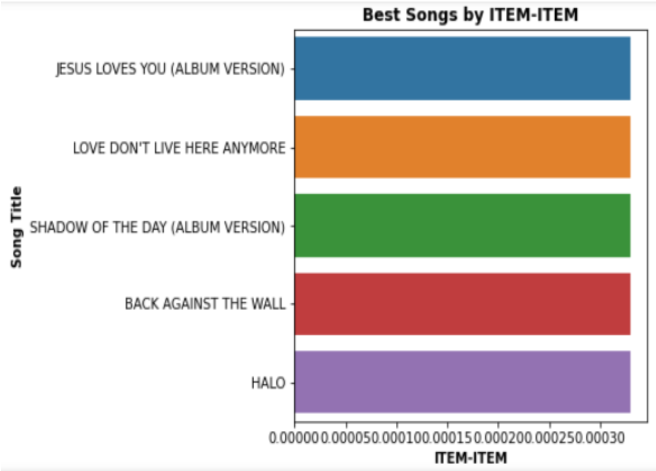


b) Item-Item Based
In this type of collaborative filtering, our work is to find a group of similar items having similar qualities/features. This type of collaborative filtering uses similarities between items to provide recommendations. This allows for promising recommendations; that is, this collaborative filtering model can recommend an item A to a user based on the similarity with item B.
The authors have used Pearson correlation and Jaccard similarity to find the similarity between the songs. And using K-Nearest-Neighbour algorithm they have formed the group of similar items corresponding to a user for which songs have to be recommended.

Recommendation to the user: 1fbdc205ddd306d96ece27248ea6f39edbd27e64 for the song_id: SOOVFYS12A81C23135 using *Pearson Relation*



Recommendation to the user 71069b5980a571ec67c6a5f4325e4ab512ff44bc for the song SOHTTNB12A8C1357E9 using *Jaccard Similarity*

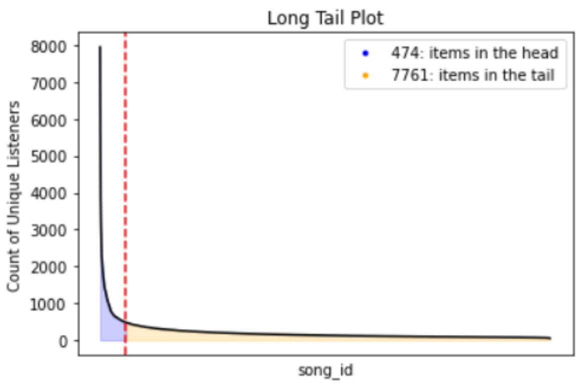


V. RESULTS AND ANALYSIS

1) Evaluation for Popularity Filtering Model

In the Popularity based model long tail curves have been plotted.

The plot depicts the numbers of unique listeners with respect to the unique songs present in the dataset. Only 474 songs are popular according to the count as depicted in the curve.



Song vs Unique Listener

2) Evaluation for Collaborative Filtering Model

In this study, authors have used four different performance measures to analyse the different models.

Root Mean Square Error:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Mean Square Error:

$$\text{MSE} = \underbrace{\frac{1}{n} \sum_{i=1}^{n}}_{\text{test set}} (\underbrace{y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i}_{\text{actual value}})^2$$

Mean Average Precision

$$mAP(u, y) = \frac{1}{N} \sum_{p=1}^{N} AP(u, y_u)$$

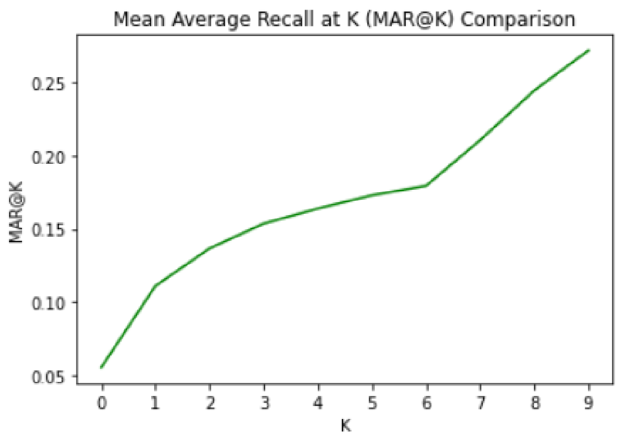
Mean Average Recall

$$mAR(u, y) = \frac{1}{N} \sum_{p=1}^{N} AR(u, y_u)$$

A) User-User Based Evaluation

The results have been displayed in the table below:

Evaluation Metrics	Value
Root Mean Square Error (RMSE)	1.45541394224723
Mean Square Error (MSE)	2.2412263204265526
Mean average Recall (mAR)	0.6488970000000001
Mean average Precision (mAP)	0.6488970000000001



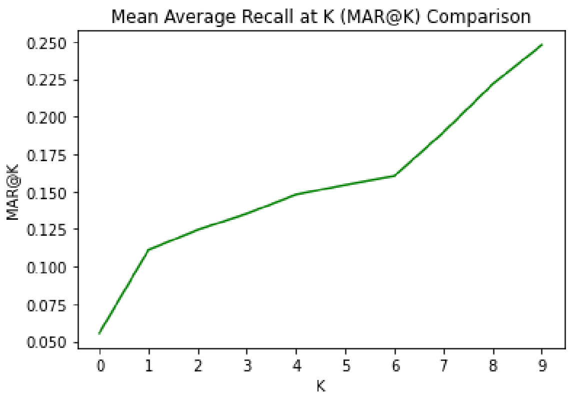
B) Item-Item Based Evaluation

Two different types of methods have been used to compute correlation and the results of these two have been discussed below.

a) Item-Item Pearson Similarity

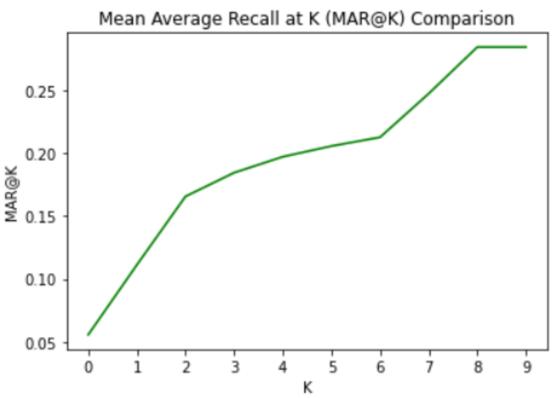
Evaluation Metrics	Value
Root Mean Square Error (RMSE)	0.53296451426650 33

Mean Square Error (MSE)	0.3105393164594553
Mean average Recall (mAR)	0.643342
Mean average Precision (mAP)	0.643342



b) Item-Item Jaccard Similarity

Evaluation Metrics	Value
Root Mean Square Error (RMSE)	0.641408136843
Mean Square Error (MSE)	0.439590501139
Mean average Recall (mAR)	0.666670000000
Mean average Precision (mAP)	0.666670000000



VI. CONCLUSION AND FUTURE WORK

The authors have implemented a Popularity based model, User-User based, and Item-Item based collaborative filtering models. The Popularity based model performed the worst with no doubt as it couldn't provide personalized recommendations. User-User based and Item-Item based models performed considerably better, with Item-Item based models performing the best. This might be due to the fact that relations amongst songs are more straightforward as compared to users. Songs have minimal sets of genres, while users may have varied tastes. In real-world contrast, item similarity is more meaningful than user similarity, as songs have some definite features/qualities. In contrast, users have indefinite features and tastes, so it's always better to find the similarities between songs. This indicates that the notion of item similarity is more meaningful than the idea of user similarity.

This project can be improvised by constructing various models using techniques like content-based filtering, collaborative filtering using Matrix Factorization, CRNN models, SVD models, Graphical models that capture the intuitions of feature mapping along with the user and the items and, Hybrid models as well.

VII. References

[1] Dupont, E., Rao, I., & Zhang, W. (2015). Offline Music Recommendation.

[2] Shakirova, E. (2017). *Collaborative filtering for music recommender system. 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).*

[3] Szu-Yu Chou, Y. Yang and Yu-Ching Lin, "Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics," 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015, pp. 1-6, doi: 10.1109/ICME.2015.7177456.

[4] Smt. Namitha S J, 2019, Music Recommendation System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 08, Issue 07 (July 2019),

[5] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere.
The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[6] Shefali Garg, Fangyan SUN. Music Recommender System, Journal of Indian Institute of Technology, Kanpur, 2014.