Artificial Intelligence

***Lab Project Submission :***

***Project Title:***

***AI based Music Recommendation System Using Machine Learning Algorithms.***

***Under Guidance of : Dr JHILIK BHATTACHARYA***

***Team Members:***

|  |  |
| --- | --- |
| ***Name:*** | ***Roll Number:*** |
| Prem Jadwani | 102103215 |
| Kavish Mudai | 102103230 |
| Krishna Khanduja | 102153007 |
| Jappan Sharma | 102103228 |

INTRODUCTION

Rapid development of mobile devices and internet has made possible for us to access different music resources freely. The number of songs available exceeds the listening capacity of 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 costumers to discover music by giving quality recommendation. Thus, there is a strong need of a good recommendation system.

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 recommenda- tion 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 fu- ture. We have implemented various algorithms to try to build an effective recommender system. We firstly im- plemented 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. We have also done experiments on content based models, based on latent factors and metadata.

PROBLEM STATEMENT:

The problem statement for an AI-based music recommendation system is to develop a solution that can provide personalized music recommendations to users based on their listening history, preferences, and behavior. The challenge lies in designing an algorithm that can accurately predict a user's music taste and suggest relevant songs, artists, or playlists, while also taking into account the latest music trends and the user's mood, location, and time of day. Additionally, the system needs to ensure the recommendations are diverse, and not simply repetitive, to avoid boredom and to keep users engaged. Overall, the goal is to create a music recommendation system that delivers an enhanced listening experience, ultimately leading to increased user engagement and retention.

METHODOLOGY

**PROPOSED SYSTEM**

The working of the system starts with the collection of data and selecting the

important attributes. Then the required data is preprocessed into the required format.

The data is then divided into two parts training and testing data. The algorithms are

applied and the model is trained using the training data. The accuracy of the system is

obtained by testing the system using the testing data. This system is implemented

using the following modules.

1.) Collection of Dataset

2.) Selection of attributes

3.) Data Pre-Processing

4.) Disease Prediction

**Collection of dataset :**

Initially, we collect a dataset for our music recommendation system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 80% of training data is used and 20% of data is used for testing.

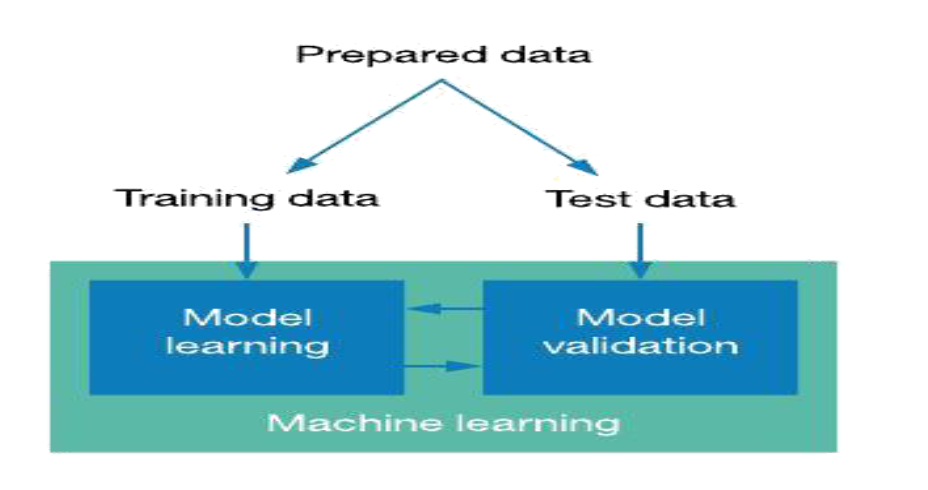
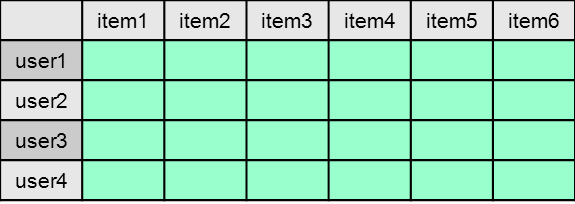


Figure: Collection of Data

**Selection of attributes :**

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system.



.

**Pre-processing of Data**

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Pre-processing of data is required for improving the accuracy of the model.

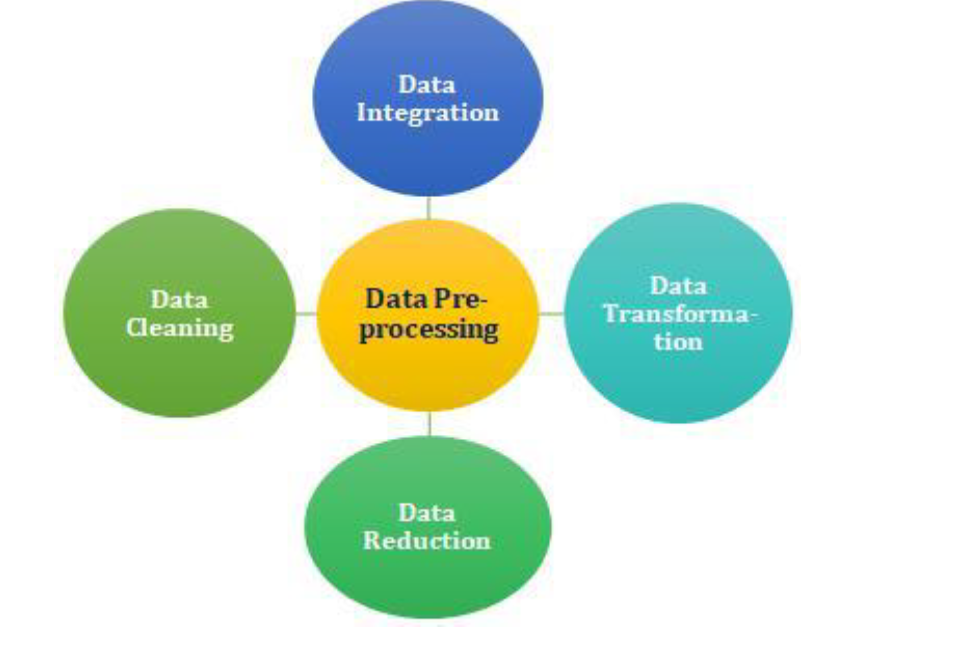
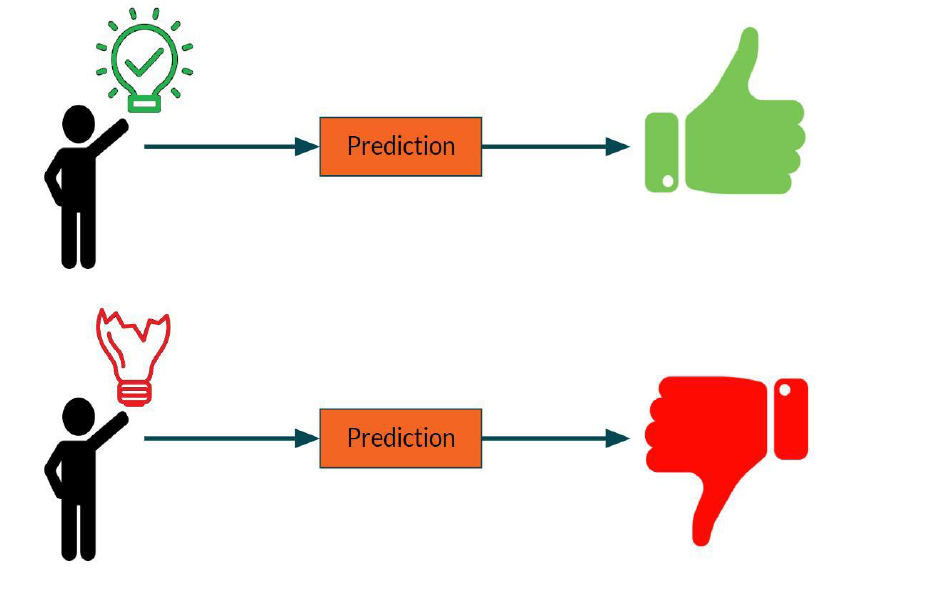


Figure: Data Pre-processing

**Prediction of Model:**

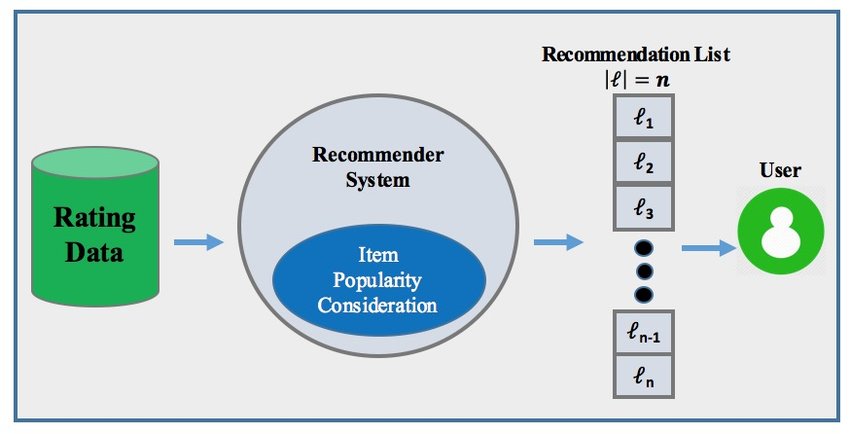
Various machine learning algorithms like BIRCH, KNN, Collaborative filtering and SVD are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the best generated recommendations for our playlists



**ALGORITHMS:**

[1] **POPULARITY BASED MODEL**

It is the most basic and simple algorithm. We find the popularity of each song by looking into the training set and calculating the number of users who had listened to this song. Songs are then sorted in the descending order of their popularity. For each user, we recommend top most popular songs except those already in his profile. This method involves no personalization and some songs may never be listened in future.



**[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.

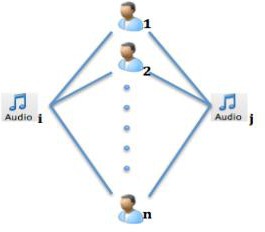
According to user based similarity model [Figure2][8], users who have similar listening histories, i.e., have lis- tened to the same songs in the past tend to have similar interests and will probably listen to the same songs in future too.

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. User should be weighted less if he has shown interests to many variety of items (it shows that either she does not discern between songs based on their quality, or just likes to explore). Likewise, user is weighted more if lis- tens to very limited set of songs. The similarity measure, *wij* = *P* ( *i* ), also has drawbacks that some songs which are listened more by users have higher similarity values not because they are similar and listened together but be- cause they are more popular.

*j*

We have used conditional probability based model of similarity[2] between users and between items:

*Wu,v* = *P* (*v/u*)*αP* (*v/u*)1−*α, αϵ*(0*,* 1)



**[3] MINI BATCH K MEANS**

A different approach is the use of mini batch k means whose main idea is to use small random batches of data of a fixed size, so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters and this is repeated until convergence. Each mini batch updates the clusters using a convex combination of the values of the prototypes and the data, applying a learning rate that decreases with the number of iterations. This learning rate is the inverse of the number of data assigned to a cluster during the process. As the number of iterations increases, the effect of new data is reduced, so convergence can be detected when no changes in the clusters occur in several consecutive iterations. The empirical results suggest that it can obtain a substantial saving of computational time at the expense of some loss of cluster quality, but not extensive study of the algorithm has been done to measure how the characteristics of the datasets, such as the number of clusters or its size, affect the partition quality.

Mini-batch K-means is a variation of the traditional K-means clustering algorithm that is designed to handle large datasets. In traditional K-means, the algorithm processes the entire dataset in each iteration, which can be computationally expensive for large datasets.

**[4]KNN MODEL**

K-Nearest Neighbors (KNN) algorithm can also be used in an AI-based music recommendation system to provide personalized music recommendations to users based on their listening history and preferences.

The KNN algorithm works by finding the K closest data points (in this case, songs) to a target data point (a song that the user has listened to and liked) based on their similarities. In the context of music recommendation systems, the similarity between two songs can be measured using various features such as genre, artist, tempo, and so on.

Once the K nearest songs are identified, the KNN model can use their ratings and preferences to predict how much the user might enjoy listening to the target song. This prediction can then be used to recommend the song to the user, who is likely to enjoy it based on their previous listening history.

KNN-based music recommendation systems are advantageous because they do not require a training process, and they can easily incorporate new data as users continue to listen to music. However, they can be computationally expensive when the data set is large, and they may not be effective when dealing with highly sparse data.

Overall, a KNN-based music recommendation system can provide personalized recommendations to users, increasing their engagement and satisfaction with the music streaming service.

**[5]BIRCH MODEL**

**[4]KNN MODEL**

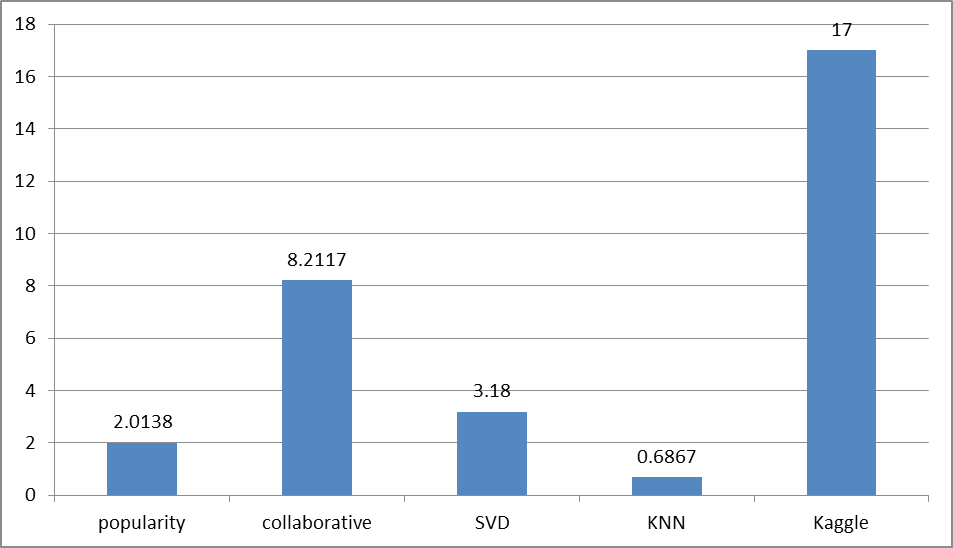
BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm can also be used in an AI-based music recommendation system to cluster similar songs together and provide recommendations based on these clusters.

BIRCH is a clustering algorithm that can create a hierarchical clustering of the data. In the context of music recommendation systems, BIRCH can group similar songs together based on their features such as genre, artist, tempo, and so on. By grouping songs into clusters, BIRCH can identify similarities between songs that may not be apparent otherwise.

Once the clusters are identified, BIRCH can recommend songs to users based on their listening history and the preferences of other users who have listened to songs within the same cluster. This can lead to more personalized recommendations for the user, as the recommendations are based on the user's individual listening history as well as the preferences of other users who have similar listening patterns.

BIRCH-based music recommendation systems are advantageous because they can easily handle large datasets and are computationally efficient. However, they may not be as effective in identifying more subtle patterns in the data, such as user preferences that may not be explicitly represented in the song features.

Overall, a BIRCH-based music recommendation system can provide personalized recommendations to users, increasing their engagement and satisfaction with the music streaming service.



Top of Form

**DATASET DETAILS:**

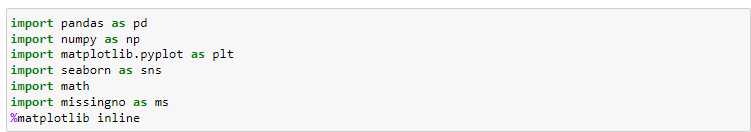
**(LINK :-** https://os.unil.cloud.switch.ch/fma/fma\_metadata.zip**)**

We used data provided by Free Music Archive(FMA) hosted by Kaggle. The data is open; meta-data, audio content analysis, etc. are available for all the songs. It is also very large and contains around 48 million (userid, songid, play count) triplets collected from histories of over one million users and metadata (280 GB) of millions of songs[7]. But the users are anonymous here and thus information about their demography and timestamps of listening events is not available. The feedback is implicit as play-count is given instead of explicit ratings. The con- test was to predict one half of the listening histories of 11,000 users by training their other half and full listening history of other one million users.

Since, processing of such a large dataset is highly mem- ory and CPU-intensive, we used validation set as our main data. It consists of 10,000,000 triplets of 10000 users. We used metadata of only 10,000 songs (around 3GB). From the huge amount of song metadata, we focus only on features that seem to be most relevant in characterizing a song. We decided that information like year, duration, hotness, danceability, etc. may distinguish a song most from other songs. To increase processing speed, we con- verted user and song ids from strings to integer numbers.

**CODE EXPLANATION( of our finalised model) :**

[1]:



Here we are importing different important libraries that are being used in the model:

Brief description of the imported modules:

1. **Pandas:** Pandas is a powerful open-source data manipulation and analysis library for Python. It provides easy-to-use data structures such as DataFrame and Series, which allow for efficient data handling and analysis tasks

2. **Numpy:** NumPy contains a multi-dimensional array and matrix data structures. It can be

utilised to perform a number of mathematical operations on arrays

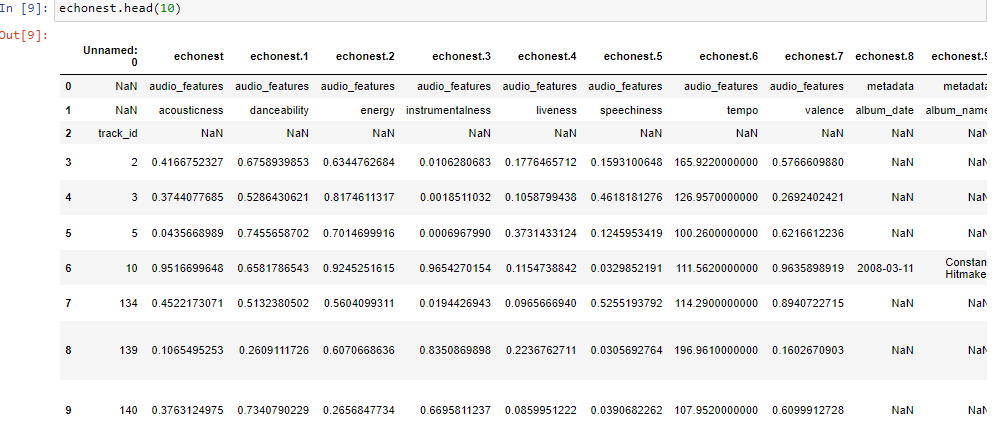
3. **Seaborn:** Seaborn is a Python data visualization library based on Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics. Seaborn is designed to work well with Pandas, another popular data manipulation library in Python, making it a powerful tool for data visualization and analysis.

4. **matplotlib:** Matplotlib is a multi-platform data visualization library built on NumPy

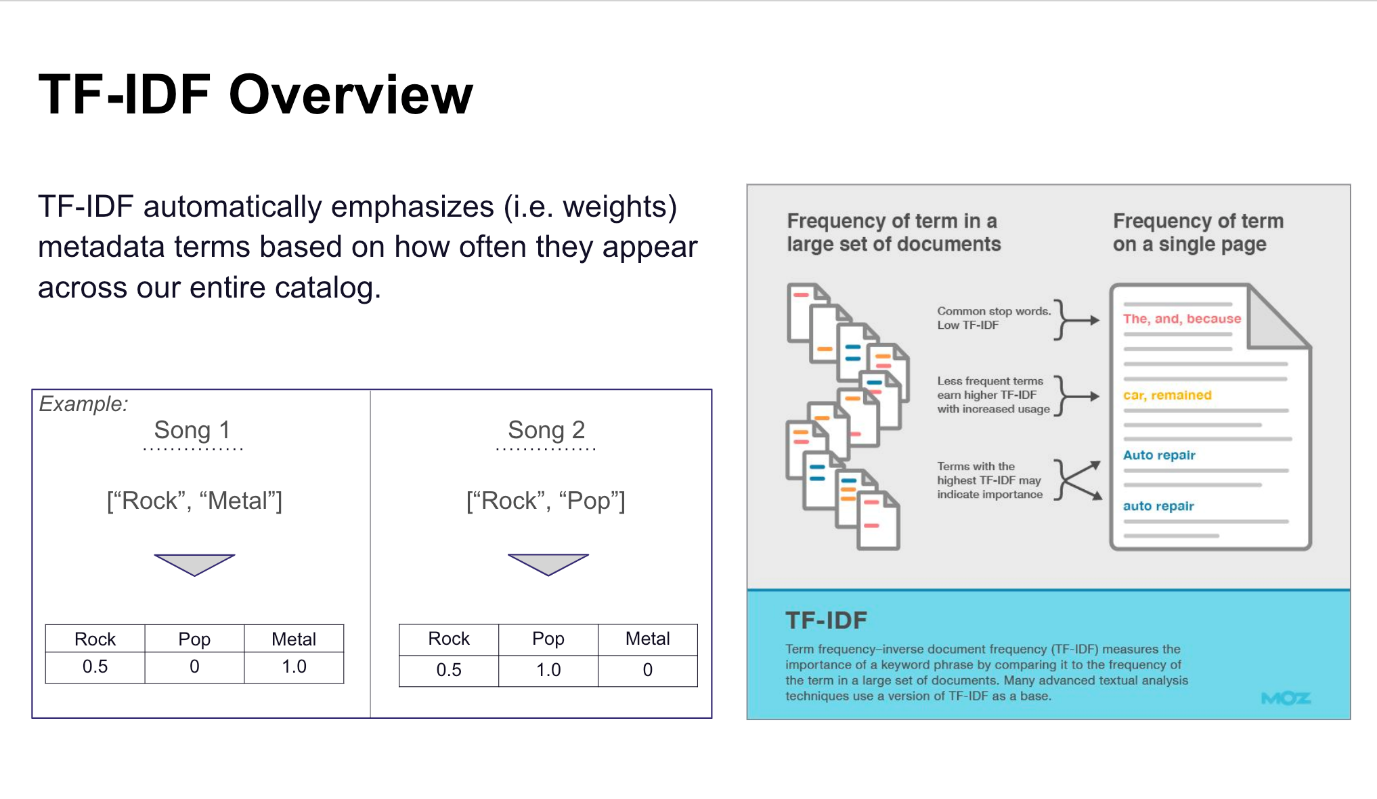
arrays. It allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

5. **missingno:** missingno is a Python library for visualizing missing data in datasets. It provides a variety of useful visualizations for identifying patterns of missing values and understanding the distribution and completeness of the data.

[2]:



Now we load our given dataset, using panda library on basis of genre,tracks,features and echoset.

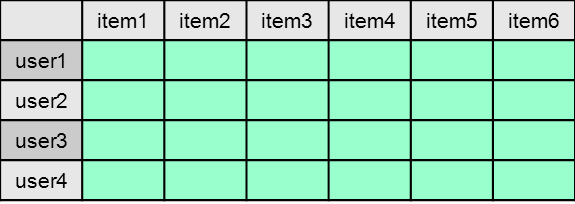




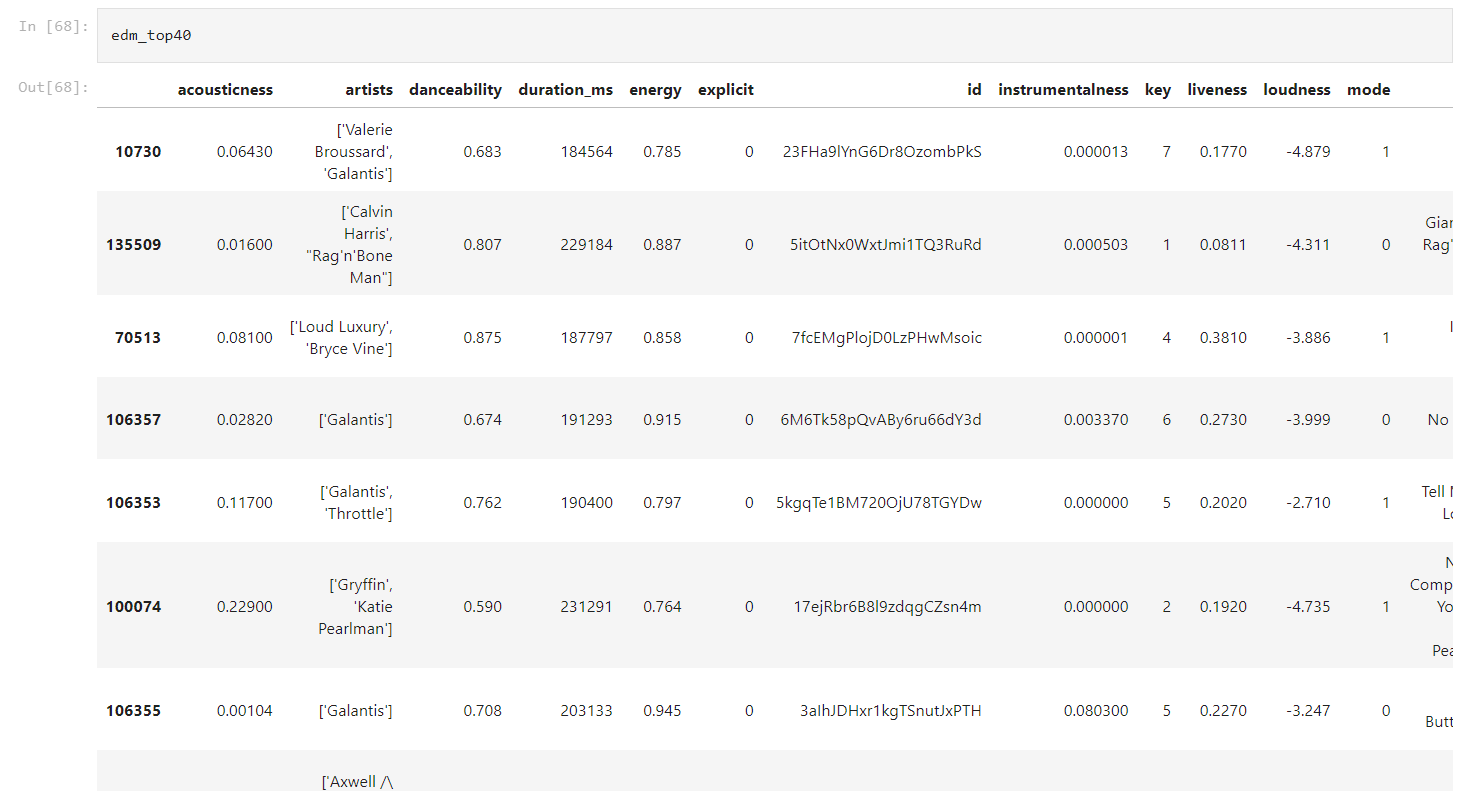
Collecting all data from different basis on genres , tracks , features and echoset and combining them all into a final dataset to produce a single data set for further analysis to produce recommendations based on machine learning algorithms.

**PERFORMANCE ANALYSIS:**

Correlation Matrix: The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.



**GENERATED RECOMMENDATIONS: -**

****

**RESULT:**

We got best results for memory based collaborative filtering algorithm. Our BIRCH based latent factor model gives better results than popularity based model. It lags behind collaborative filtering algorithm because the matrix was too sparse which prevented objective functions to converge to global optimum. Our K-NN model did not work well and performs worse than even the popularity model. The reason behind this is that we have the features of only 10,000 songs, which is less than 3 % of the whole dataset so only some of these 10,000 songs could be rec- ommended. The huge lack of information leads to the bad performance of this method

**CONCLUSION:**

This is a project of our Artificial Intelligence course. We find it is very good as we got a chance to practice theories that we have learnt in the course, to do some implemen- tation and to try to get a better understanding of a real artificial intelligence problem: Music Recommender System. There are many different approaches to this problem and we get to know some algorithms in detail and espe- cially the four models that weve explained in the paper. By manipulating the dataset, changing the learning set and testing set, changing some parameters of the problem and analyzing the result, we earn a lot practicing skills. We have faced a lot of problem 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 lot of efforts, we have overcame all of these.

The best part of this project is the teamwork. Both of us come from different countries and thus have different cultures and ways of working. We took a bit of time to get to know each other, to adjust ourselves and to perform like a team. We become much more efficient by the time the team spirit is formed and we also enjoy more. We both find this project a nice experience and all the effort put is worthy. We have learnt a lot from this project.

In terms of research, we still have a lot to do to make our studies a better one. Music Recommender System is such a wide, open and complicated subject that we can take some initiatives and do a lot more tests in future.

We also got to realize that building a recommender sys- tem is not a trivial task. The fact that its large scale dataset makes it difficult in many aspects. Firstly, recommending 500 correct songs out of 380 million for different users is not an easy task to get a high precision. Thats why we didnt get any result better than 10 %. Even the Kag- gle winner has only got 17 %. Secondly, the metadata includes huge information and when exploring it, it is dif- ficult to extract relevant features for song. Thirdly, techni- cally speaking, processing such a huge dataset is memory and CPU intensive.

All these difficulties due to the data and to the system itself make it more challenging and also more attractive. We hope that we will get other opportunities in the future to work in the domain of artificial intelligence. We are certain that we can do a better job.

. **FUTURE WORK**

Run the algorithms on a distributed system, like Hadoop or Condor, to parallelize the computation, decrease the runtime and leverage distributed mem- ory to run the complete MSD.

*•*

Combine different methods and learn the weightage for each method according to the dataset

Automatically generate relevant features

*•*

Develop more recommendation algorithms based on different data (e.g. the how the user is feeling, social recommendation, etc)

*•*