**Intelligent Service Recommendation Engine**

**MINI REPORT**

**ASSIGNMENT 1 MINI REPORT**

**INTRODUCTION**

The assignment given is the first stage development of a Music recommendation engine. The first stage of development involves the development of two two modules, one module is to generate artist dataset and music dataset. And the other module is to compute the similarity check from the artist or music dataset.

**PROBLEM ANALYSIS**

We need to load the data without the use of any data science library. After loading the data, in the first module we need to create an artist dataset which resembles artist features and a music dataset which resembles music features.Each of this dataset need to be stored in a dictionary. In the second module we need to calculate the similarity score between 2 datas in the dataset. Similarity scores help us to identify whether 2 data present in data are similar or not. Here we need to use any five similarity functions to calculate the similarity score. Apart from these 2 modules we also need to develop a Text based GUI which asks the user to give input and calculate the similarity. The program can only terminate only after the user is done.

**IMPLEMENTATION OF SOLUTION**

Here the first task is to load the data.csv file into program by using a function called load\_dataset\_module(). For loading the data.csv file we have imported csv library which is an inbuilt library of python. After the loading the data extract the following fields “accoustiness”, “artists”, “danceability”, “energy”, “id”, “liveness”, “loudness”, “name”, “popularity”,”speechness”, “tempo”, and “valence”.

Now the next task is to develop two dataset from the given data.csv. For implementing that I created two functions called get\_artist\_music() to load artist dataset and get\_music\_features() to load music dataset.

ALGORITHM FOR ARTIST DATA:

1. Start
2. Load the data.csv file into the program using csv library
3. Extract the needed columns from the data.csv
4. All the features of different songs of same artist will be combined and stored in unique rows.
5. Store all the data in a dictionary
6. stop

In the data.csv file there are multiple songs for some artists. So here we have combined all the different songs of each artist then combined and stored all the other fields into a unique row. The function will also assign id number to each row so that the user can easily check the similarity score between two artists. The function will return a dictionary which stores all the artist data.

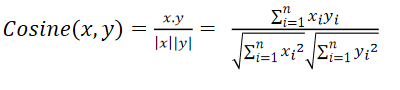
ALGORITHM FOR MUSIC DATA:

1. Start
2. Load the data.csv file
3. Using a loop iterate through each row in the data
4. Extract all the required columns and store them in a dictionary
5. stop

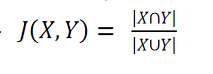
From the data.csv, we parse through each row and store all the required field from the data.csv file. The function will also assign an id number to each row data so that the user can easily check the similarity between two music tracks. The function will return a dictionary which contains all the music data

ALGORITHM FOR SIMILARITY FINDING:

1. Start
2. Declare a function to find cosine similarity. Calculate cosine similarity using mathematical formula and returns the similarity score



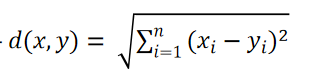
1. Declare a function to find jaccard similarity. Calculate the similarity using mathematical formula



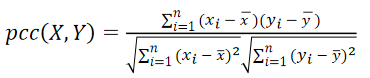
1. Declare a function to find manhattan similarity. Calculate the similarity using mathematical formula and returns the similarity score.



1. Declare a function to find euclidean distance similarity. Calculate the similarity using mathematical formula and returns the similarity score.



1. Declare a function to find pearson similarity. Calculate the similarity using mathematical formula and returns the similarity score.



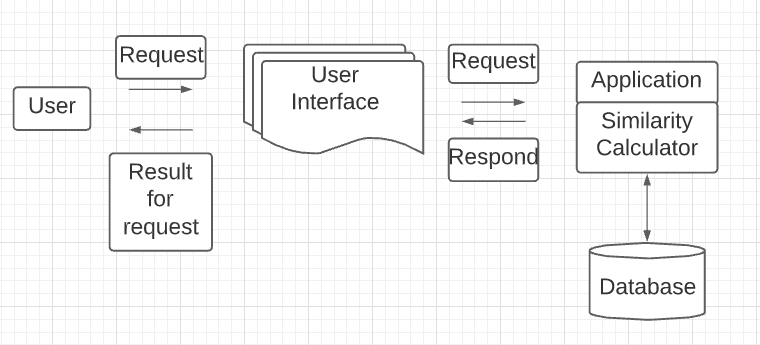
1. Stop

**PROGRAM EXECUTION**

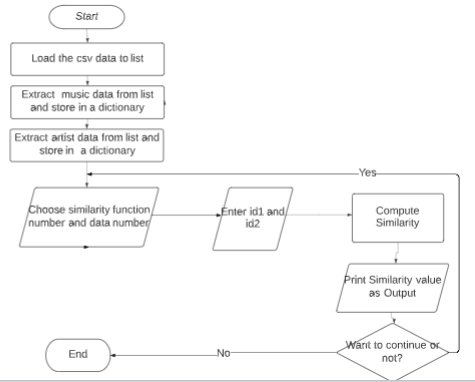
On executing the program, a simple GUI is displayed on the output screen showing details about the available functions and the dataset available. Then the program ask the user to give input for the similarity function. We give a value from 1 to 5. The value from 1 to 5 corresponds to similarity function which is displayed on screen. After receiving the input, the program again ask the user to select the dataset. We should give a value of either 1 or 2. The value 1 selects artist dataset and 2 selects music dataset. Then the user needs to give 2 id number to calculate the similarity. After receiving two id, the similarity function will be called and the similarity score will be displayed as output. Now the program will ask the user whether to continue with the program or to exit from the program. If the user selects not to exit from the program then the program again ask for the user to input similarity function, dataset number, and 2 id to compute the similarity. If the user selects to exit from the program then program will be terminated.

**PROGRAM STRUCTURE**

1. SOFTWARE ARCHITECTURE

****

2. FLOWCHART

****

**CONCLUSION**

To conclude, the implementation details of Recommendation engine are provided into some extent in this report. The program explanation, flowchart and software diagram have also been included. The next stage of the recommendation system will be developed by referring to this document.

**REFERENCES**

<https://www.kaggle.com/yamaerenay/spotify-dataset19212020-160k-tracks>

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0144059>

<https://realpython.com/>

<https://stackoverflow.com/>