Final Document on HADOOP MUSIC RECOMMENDATION SYSTEM

Website link : webpages.uncc.edu/sgummad2/

**Introduction**

**Recommender systems are systems that, based on information about a user's past patterns and consumption patterns in general, recommend new items to the user.**

**Our Project**

Music Recommendation System selects the most interesting song for a user based on ratings given by them. In this system, we consider that the rating of a song is influenced by likes and dislikes of users. This system’s recommendation list works by considering user’s interests, making the suggestions more accurate and personalized. Songs are chosen based on similarity of songs which user has listened to or liked previously, and a prediction measure is calculated for the user and similar songs. The project uses item-based collaborative filtering to recommend a list of top 10 songs to the user.

**Approach**

**Item Based Collaborative Recommendation:**

**Item-based collaborative filtering is a model based** algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of several similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset. The parameter considered in our project is different users rating.

* We grouped all the songs each user has rated, for all the users. Now we have ordered pairs of interests for songs each user has.

UserId [songId1=rating1, songId2=rating2, ...]

* Then we calculated similarity between every song with every other song which are rated by the same user.

SongId [songId1=similarity1, songId2=similarity2, ...]

* Then we predict the user-song pair rating using weighted average.

UserId SongId ActualRating PredictedRating

* Based on the above predicted ratings, we recommend top 15 songs to a user.

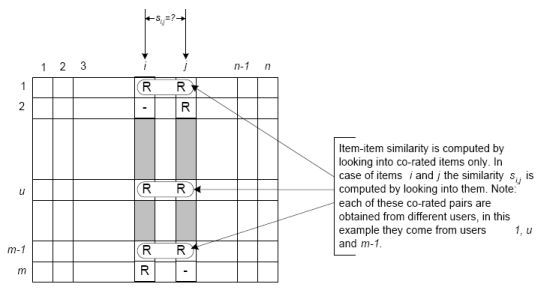
UserId [songId1, songId2, .............., songId15]

* We have implemented this using Hadoop Map Reduce and the programming language used is java.

**Tasks**

* **Similarity Computation:**

The similarity values between items are measured by observing **all the users who have rated both the items**. As shown in the diagram below, the similarity between two items is dependent upon the ratings given to the items by users who have rated both of them:

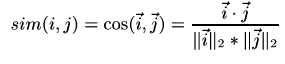


**Similarity Measures:**

There are a number of different mathematical formulations that can be used to calculate the similarity between two items. As can be seen in the formulae below, each formula includes terms summed over the set of common users U.

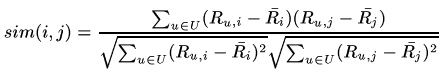
Cosine based Similarity:

It is also known as **vector-based similarity**, this formulation views two items and their ratings **as vectors**, and defines the similarity between them as the angle between these vectors:



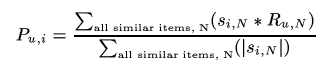
Pearson Correlation based similarity

This similarity measure is based on how much the ratings by common users for a pair of items deviate from average ratings for those items:



Prediction Calculation:

Once we make a model using one of the similarity measures described above, we can predict the rating for any user-item pair by using the idea of **weighted sum**. First we take all the items similar to our target item, and from those similar items, we pick items which the active user has rated. We weight the user's rating for each of these items by the similarity between that and the target item. Finally, we scale the prediction by the sum of similarities to get a reasonable value for the predicted rating:



**Recommendation Computation:**

Retrieve songs similar to each song that the user has rated and calculate predicted rating for each similar item. Remove songs that are already rated by the user, compare rating for remaining similar items, sort the list and output top 10 recommended songs.

**Dataset:**

Project is implemented using the Yahoo Music user ratings of songs with song attributes, version 1.0 (R2) dataset from the Web scope Datasets which was provided on canvas and it contains huge data. Data set can be found in the URL: http:// webscope. sandbox. yahoo.com/myrequests.php . Each song in this data set is accompanied by artists, album and genre attributes. The users, songs, artists and albums are represented by randomly assigned numeric ids, so that no identifying information is revealed.

Tab is used as a delimiter for all data files.

This dataset consists of twenty files:

1. 10 Train files: ydata-ymusic-user-song-ratings-meta-v1\_0/train-n.txt (n is 0-9)

2. 10 Test files: ydata-ymusic-user-song-ratings-meta-v1\_0/test-n.txt (n is 0-9)

Each row of the test & train file is of the form:

“UserID SongID rating”

Snippet:

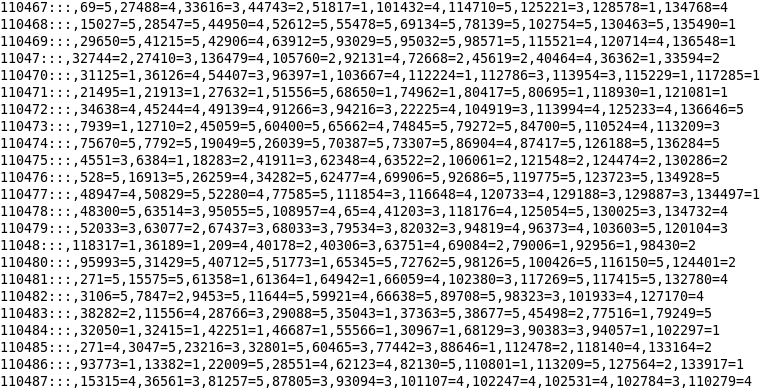
49 2169 5

49 2169 2

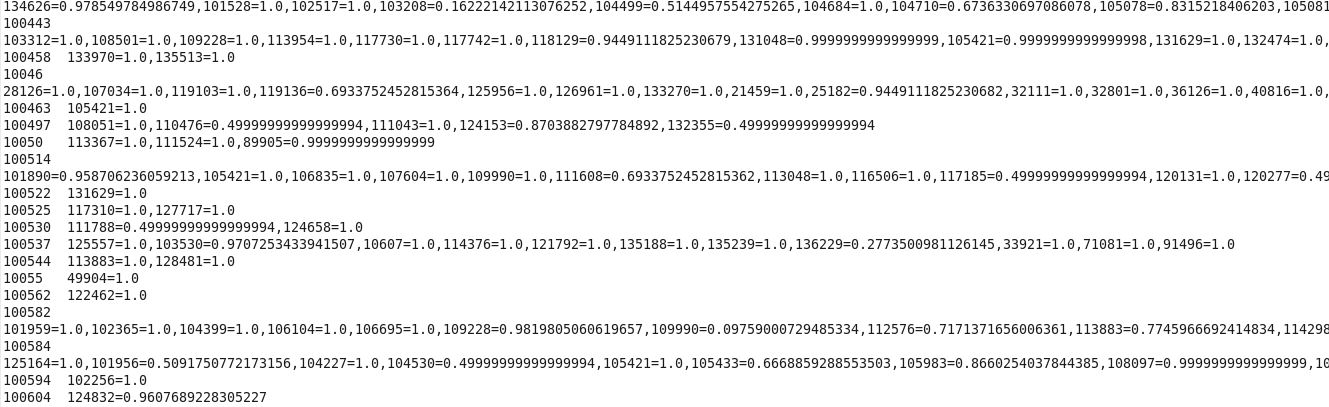
50 311 5

**Output Screens:**

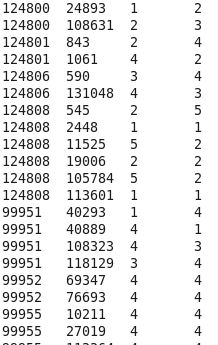
**1.Preprocessing of data**

****

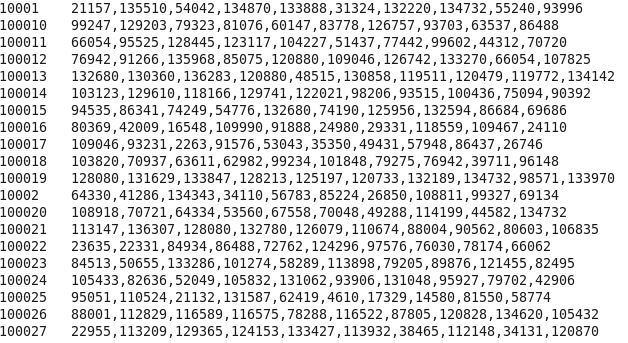
**2.Similarity List**

****

**3.Predicted Rating List**

****

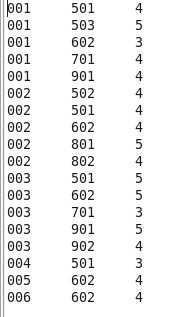
**4.Final Recommendation List**

****

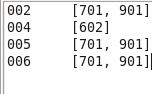
**Motivation:**

We tried to implement our music recommendation system by giving our own input data collected from our friends. The suggested recommendations are aligned to our friends interests based on song ratings he/she has provided.

Collected Data:



Recommendations:



**Performance Evaluation:**

The similarity techniques used for our recommendation system are Pearson Coefficient and Cosine Similarity. The time taken to run the project using these techniques are 95.5684 and 140.2354 respectively. Prediction Accuracy is measured using Mean Absolute Error method(MAE). Accuracies are found to be 2.5 and 2.42 respectively for the above mentioned techniques.

**Task Accomplishment**

* **Definitely**: As mentioned in our project proposal, we have successfully implemented the definitely part. Final outcome will now give a user top 10 music recommendation list based on Item Based Collaborative Filtering technique, using song ratings from different users. By calculating the similarity between each pair of song in the input using similarity techniques, we have successfully implemented the Collaborative system.
* Likely: We tried to implement User Based Collaborative Filtering Technique. But, due to time constraint we could not design it completely.
* Ideally: In future, we will try to implement the system in PySpark .

**Roles and Responsibilities:**

* Requirement Analysis and Data Gathering- Nithya, Manasa , Vichithra
* Coding: Preprocessing – Manasa

Cosine Similarity Calculation – Nithya

Pearson Coefficient Calculation – Nithya

Prediction Analysis – Vichithra

Recommendation – Vichithra, Manasa

* Documentation and Webpage – Nithya, Manasa , Vichithra

**References:**

<https://www.scienceez.com/build-recommender-system/>

<https://www.youtube.com/watch?v=DNP9romqWMI>

<https://en.wikipedia.org/wiki/Recommender_system>

<https://thesai.org/Downloads/Paper_37/Item_based_Multi_Criteria_Collaborative_Filtering_Algorithm.pdf>