Yahoo Music Recommendation

Group – 3idiots

Project Report

Under the Guidance of

Prof. Rensheng Wang

By
Seshasai Chaturvedula
Kishan Teli
Tanay Parekh

Stevens Institute of Technology

Dept. of Electrical & Computer Engineering

Data Acquisition & Processing I (EE-627A)

CERTIFICATE

This is to certify that the project report entitled “Music Recommendation” done by Harsha, Kishan Teli and Tanay Parekh is an authentic work carried out by them at Stevens Institute of Technology, Hoboken, NJ, USA under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any other degree to the best of my knowledge and belief.

ACKNOWLEDGEMENT

We extend our special thanks to **Professor Rensheng Wang,** Department of Electrical Engineering for his support and guideline to make this project. It is very wonderful experience to learn under the Professor.

TABLE OF CONTENTS

1 Introduction

1.1 Project Definition

In This project we are trying to make music recommendation system using music rating data for users and applying different algorithm and output is recommendation for songs. We use Yahoo Music Data which has large varieties data. To implement different algorithms we you python programming Language.

Figure 1 Project Flow

1.2 Motivation

In todays music world we has seeing handers thousands songs and every day more and more are coming out and to play that all songs all different companies are there but biggest question for user is which song they want to lesion because to find the correct song from this big library it is to much time consuming
But if any one can help and give some recommendation or filter out some songs so that it's very helpful for user and provider to mange this big data, we have some e-music provider like Spotify, pandora, amazon music, etc. all they have music recommendation system or filter system by different parameters.

1.3 Scope of Work

Overall Workflow:

- Observation of Music Data Set
    - Yahoo music data set is very large and contains many parameters like the Users, TracksID, AlbumID, ArtisID, GenreID and Ratings for each ID.
- Data Processing
    - Given User Data is usually not fit to use. Yahoo Music data was unsorted so we cannot use it directly. In order to solve this problem, we pre-process the data so that it can be used.
- Implement Algorithm
    - We try different-different algorithm to get batter output.
- Testing Dataset
    - Our testing dataset contains six tracks for each user which are to be tested.
- Observation
    - After the data is tested observations are made and output is predicted.

2 Data Set

Yahoo Music Dataset is very complex and it has many parameters like user id, track id, album id, artist id, genre id and rating information.

Dataset consists of approximately

- 40000 users
- 224000 songs
- 53000 albums
- 18700 artists
- 600 genres

The Track data consists of Hierarchy of

Track Id=>

Album Id=>

Artist Id =>

Genre Id

Important Dataset information is provided below with pictorial view.

2.1 Track Dataset

All Tracks are belongs to one album that track also has artist and it will also belongs to different genre.

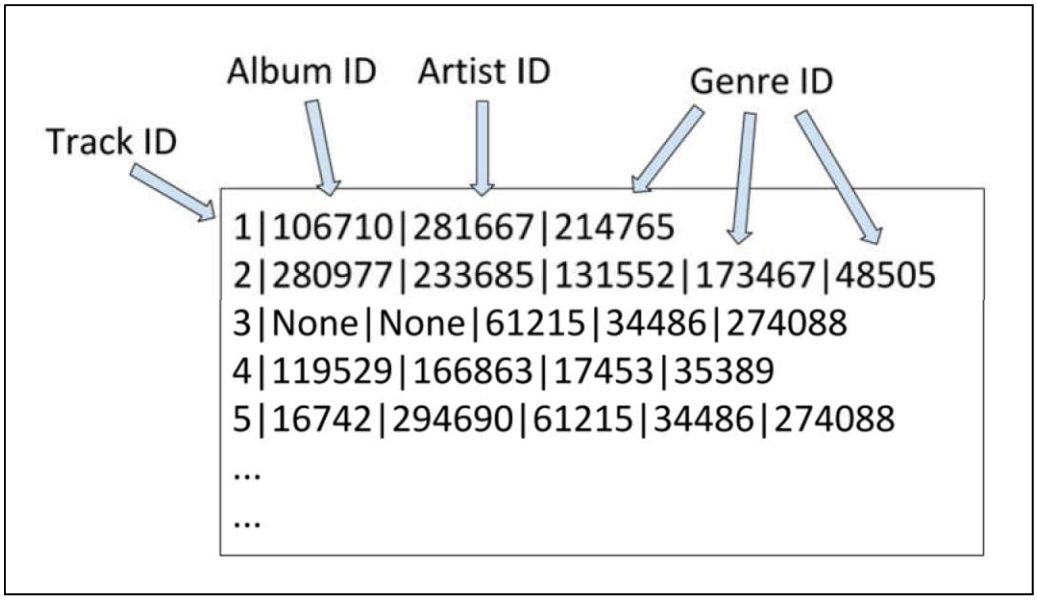

Figure 2 Track Hierarchy Information

2.2 Album Artist & Genre IDs

Data set contains separate files for album, artist and genres which represents the IDs of album, artist and genres present in dataset. This was useful while deriving a hierarchy algorithm where a track was suggested using the corresponding album, artists and genres. It also helps to check whether the item id in the training data is of which type.

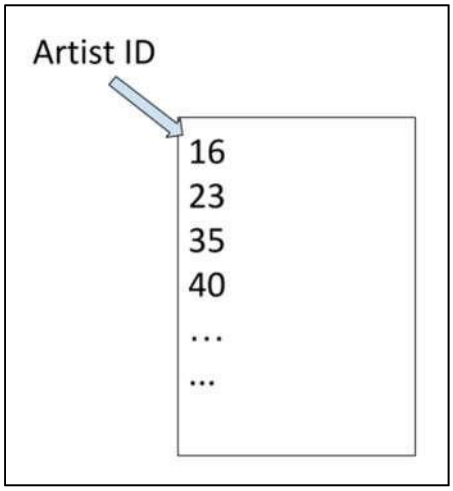

Figure 3Album Artist & Genre IDs

2.3 Training Data

Training data has the user id and number of Id the user had rated. The rating could be for a track id, album id, artist id or genre id. This Training data will be used to Train the Algorithm and predict the songs for the user.

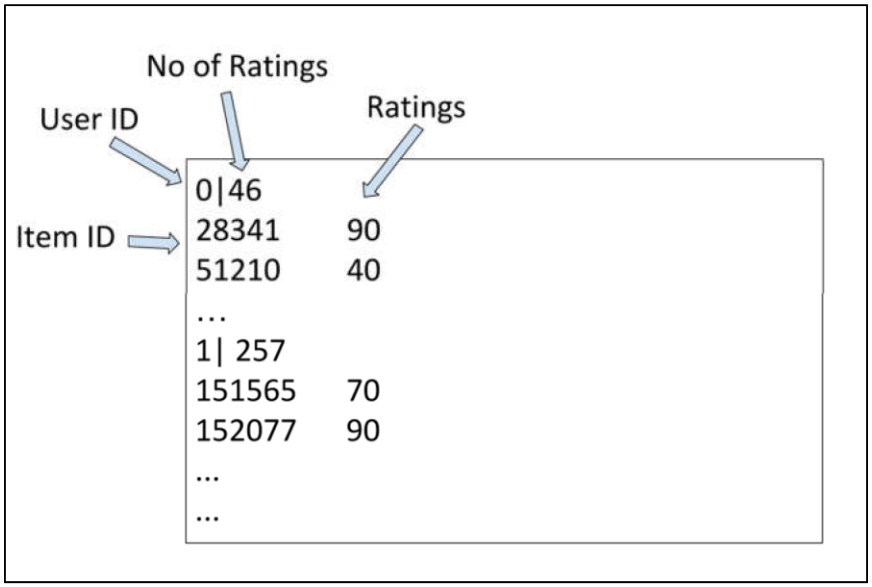

Figure 4 Training Data

2.4 Testing Data

From the training data, algorithm system created below test data file which has the IDs of a song to predict. system will predicts the ratings for the 6 tracks and sort them according to ratings value. The 3 highest rating tracks will be recommended for user and will be given 1 as result. Other 3 tracks will be given 0 value as result.

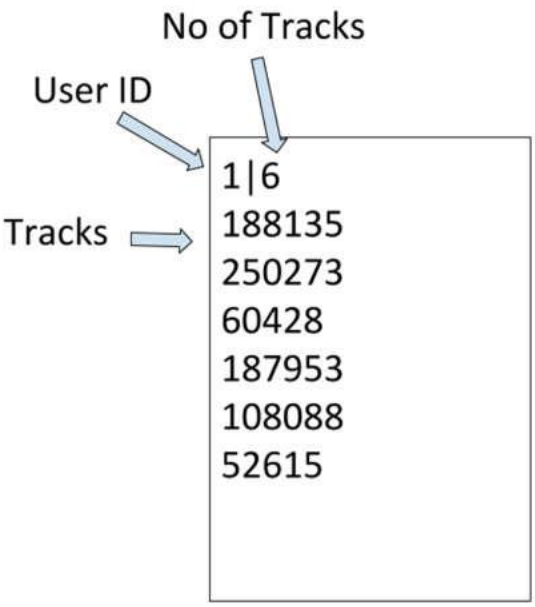

Figure 5 Testing Data

2.5 Data Pre-Processing

Raw data which we get is not always useful for training. We need to process the given data and create new data for Machine Learning.

We get below data from raw data for use in Algorithms

- Track => Album => Artist => Genre Hierarchy for each Testing Data
- List of Tracks presents in Each Album (Album => Tracks)
- List of Albums of each Artist (Artist => Album)
- List of Artist of various Genres (Genre => Artist)

3 Training Algorithm

Most important part of any Machine Learning Project is to selection of the Algorithm for Data Processing & Training. We use total four and We first use Weight Method.

This section provides insights of the various algorithms used in the project.

3.1 Method 1– Album + Artist + Genre Ratings with weights

- Album is more related to Songs compared to Genre
- Album ratings will have more impact on recommendation
- So we use the weights for each information
- The weights were changed to get the best possible result.
- So here changing the weights gives the change in correction rate
- Album (1.2), Artist (0.6), Genre (0.3) – 0.87

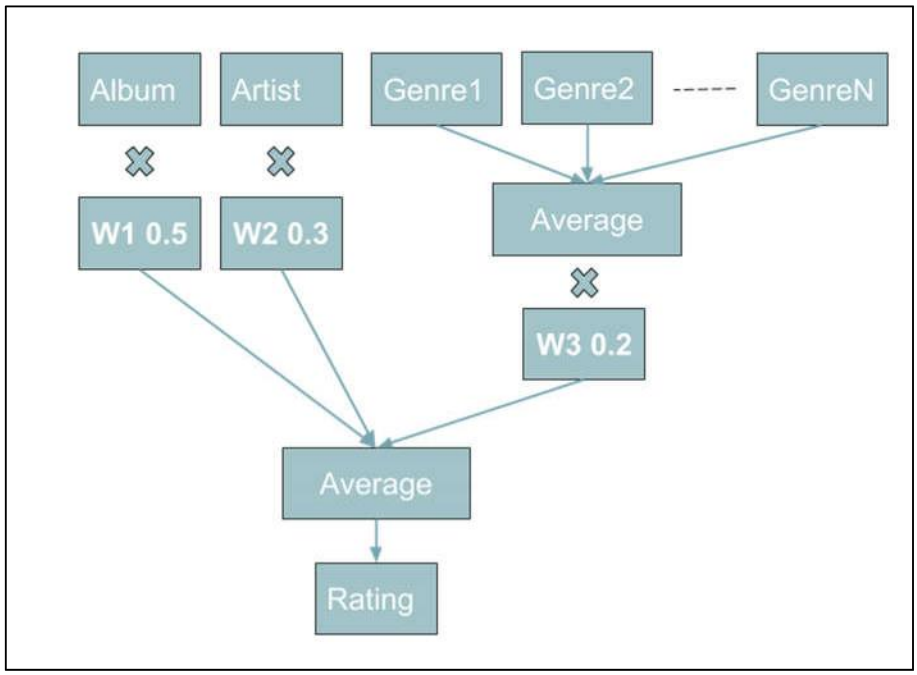

Figure 6 Album Artist and Genre with Weight Algorithm

3.2 Method 2 – Alternating Least Squares Algorithm

- ALS attempts to estimate the ratings matrix R as the product of two lower-rank matrices, X and Y, i.e. X * Yt = R. Typically these approximations are called 'factor' matrices. The general approach is iterative. During each iteration, one of the factor matrices is held constant, while the other is solved for using least squares. The newly-solved factor matrix is then held constant while solving for the other factor matrix.
- So, we see that in data there are empty data in factor matrix r because so all user did not rated all songs so we get none over there.
- We can't say that user doesn't like that song or liked that song so to get nearest value we use this method
- After applying this method we get score for all Id for all user.
- Then we process the information obtained and get an output.
  Accuracy – 0.78

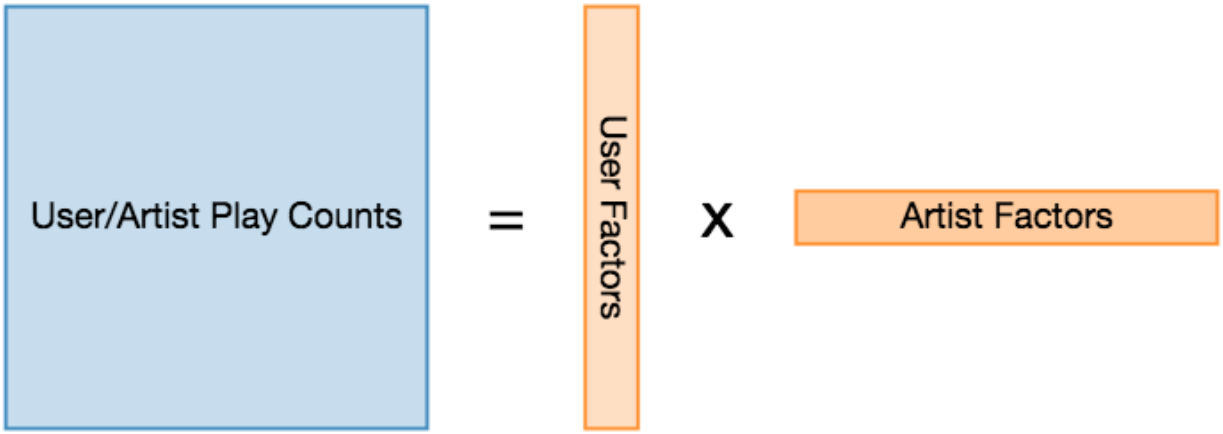

Figure 7 Alternating Least Squares Algorithm

3.3 Method 3– Keras Model

- We have also used keras model of tensorflow to predict the ratings.
  Configuration of the Keras model are:
- It is a sequential model of Convolutional Neural Network.
- First all the ratings are converted to vectors of int type.
- UserID and the itemID (including TrackID, Album ID, Artist ID and Genre ID) are mapped based on UserID and the features data is compressed and feed as input.
- The model has a layer size of 10 and a density of 1.
- The model is time complex as the input data is huge. It took us around 8 hours to train the model with 5 epochs and a batch size of 100.
- Final Accuracy we got with the model is 0.81.

3.4 Method 4 – Ensemble Algorithm

- Once we have observation from various algorithms, we can ensemble the results of various algorithm and generate the new prediction.
- This method is helpful because various algorithms have their own approaches and Pros and cons. Therefore, sometimes a method can have corner cases which it cannot predict properly.
- In this scenario, if we ensemble different outputs then one method can eliminate the corner cases of other methods and can help each other to improve the result.
- Here, we tried the prediction results from the various previous results and tried combine two to up to ten prediction files results using Ensemble Algorithm
- We got prediction results from 0.78 to 0.878 for different cases.
- The Highest prediction result that we got is 0.878

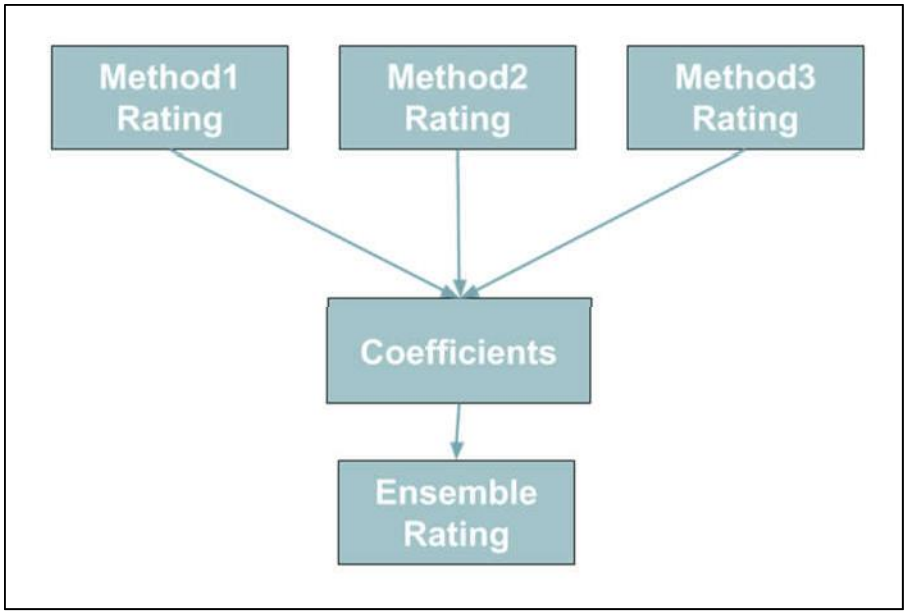

Figure 8 Ensemble Algorithm

4 Result

Observation

- We got a highest score of 0.878 in Kaggle by using Method 4.
- Using Keras and the ALS method of the spark library got scores that are less than the weighted method and Ensembling method.

5 Conclusion

Working on this project was a great learning experience as well as helped us to get an opportunity to work on the practical example of Machine Learning Application. We made our hands dirty with Data Pre-processing, find different approaches to improve the correction rate and wrote completely new algorithm from scratch. Leader Board also helped us to have healthy competition among our class groups which motivated us to achieve higher scores.

6 Files Disclosed

The following files are disclosed with our submission.

Weights Model:

1. Score_generator.py: Add weights to the ratings and give the final ratings.
2. Prediction_maker.py: After weighted ratings file is generated, this script we make predictions on the test data.

Spark ALS Model:

1. Spark_final.py: Complete training and prediction in single file.

Keras Model:

1. Utils.py: For data preprocessing and keras model building.
2. Recommend.py: For training the model and saving the model.
3. Predict_scores.py: To make predictions from the trained model.

Ensembling learning:

1. Ensembling.py: To ensemble all the previous submissions and make an Ensembling model.

All the methods give predcitions.txt file as an output. It is in the format
UserID|TrackID|Rating

Txt_to_csv.py: Script to change prediction text files to CSV file which can be submitted in Kaggle.