Music Recommendation System Based on NetEase Cloud Music

1. Abstract

Recommender systems, as one of the most successful web intelligence implementation, has a great impact on our daily life.[1] The whole procedure of Recommender System is Data Acquisition, Data Processing, Data Analysis, and Data as well as Outcome Visualization. One of the applications of Recommender Systems is when shopping online on Amazon or other shopping websites, their Recommender Systems will automatically push the items information to users, which they are interested most based on the huge items and users' information.[2] So, the main objective of the project is to build a Recommender System to recommend music to our users by applying the methods learning from this course, such as Content-based and Collaborative Filtering Recommender System. Therefore, we will separate this project into six parts, namely, Problem Identification, Data Acquisition, Data Processing, Data Analysis, Building a Recommendation System, and Visualization of Results.

Keywords: Recommender Systems, Content-Based, Collaborative Filtering.

2. Introduction

Nowadays, people are more likely to listen to musics by apps. If they want to listen to different kinds of music but have no idea what songs they want to, a Recommender System recommending songs to them will be a better choice.[3] Since songs of the type they listen to is similar to the outcome of Recommender Systems. For example, if a user listens to classic frequently, which means the user likes this type of music, and Recommender System will recommend classic music to the user to meet the user's requirement.

3. Motivation

The amount of music data is getting larger and there is not necessarily a relationship between the data. So what we need to do is to extract the data and analyze the relationships among massive data and do a recommendation to our users.

For example, Everyone's historical listening record is different, some people like classical and some like pop. If we can extract these records, then we can label each type of person, i.e. what type of music they like. Meanwhile, we should also label every song and match the users and songs. Having done this, we recommend songs to every user and need to do feedback. If users like the songs we recommend, they will listen to them many times so our recommendation is accurate. But if they only listen one time and do not listen later, it means we need to reconsider our algorithms to obtain a more accurate result.

4. Data Acquisition

The module of data acquisition is mainly composed of two parts. One part is to obtain user information from NetEase Cloud Music's website, that is, website crawler. The other part is to store the captured information in the database and do data connection to ensure that the algorithm module can call-related data conveniently and quickly.

4.1 Flow Chart

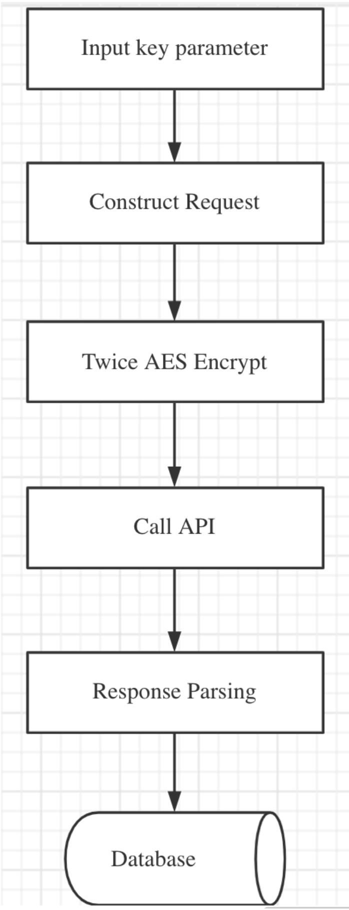

Figure 1 Data Acquisition Flow Chart

4.2 Design ideas and procedure of Data Acquisition

1) Use the user's ID as a parameter to construct the request.

2) Encrypt the request twice with AES.[4]

3) Send encrypted information to the NetEase server.

4) Parse the received response.

5) Store in the database.

4.3 Database Storage and Its Design diagram

To meet the atomicity of data storage, use the one-to-one relationship design principle mode as much as possible, or reduce the one-to-many feature columns as much as possible. In this process, try to avoid large data and data miscellaneous phenomena, otherwise, it will not only affect the software development progress but also increase the difficulty of the work and affect the quality of its products. This is also convenient for us to maintain and expand the existing database. At the same time, fully understand the inevitable connection between entities, and then achieve the goal of information and data dispersion, and on this basis, improve the overall work efficiency of the staff, improve the reliability, science, security, and performance of software applications (figure 2).

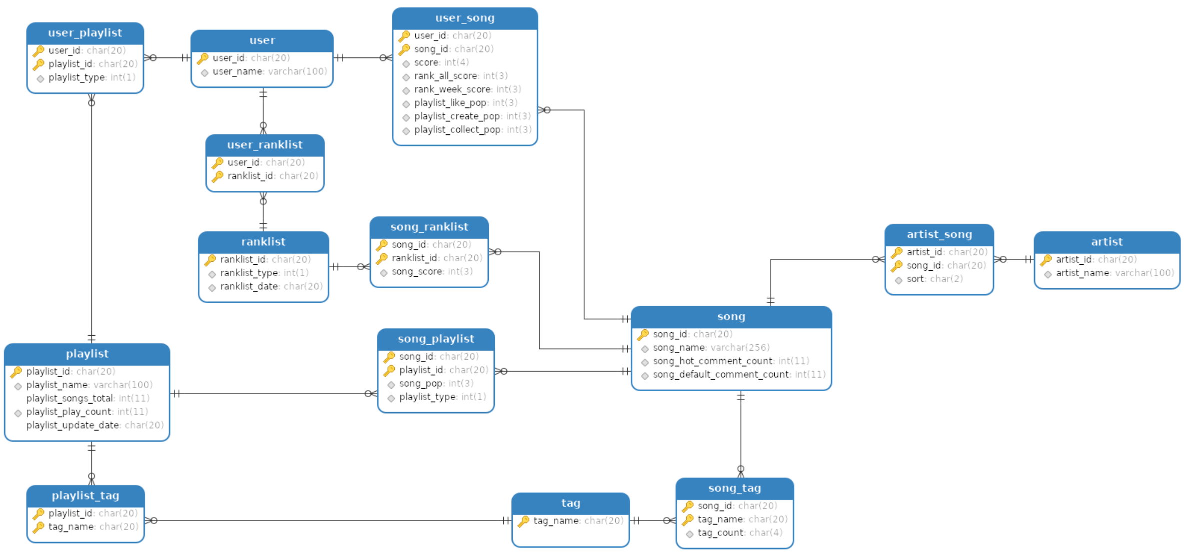

Figure 2 Database Design Diagram[5]

5. Recommender Systems

The Recommender System made by this project is divided into two parts. One is based on the user's listening history and favorite songs i.e. a content-based Recommender System.[6] The other one is to use clustering and collaborative filtering to recommend songs without the user's listening data i.e. based on user collaborative filtering.[7]

5.1 Content-based Recommender System

5.1.1 Flow Chart

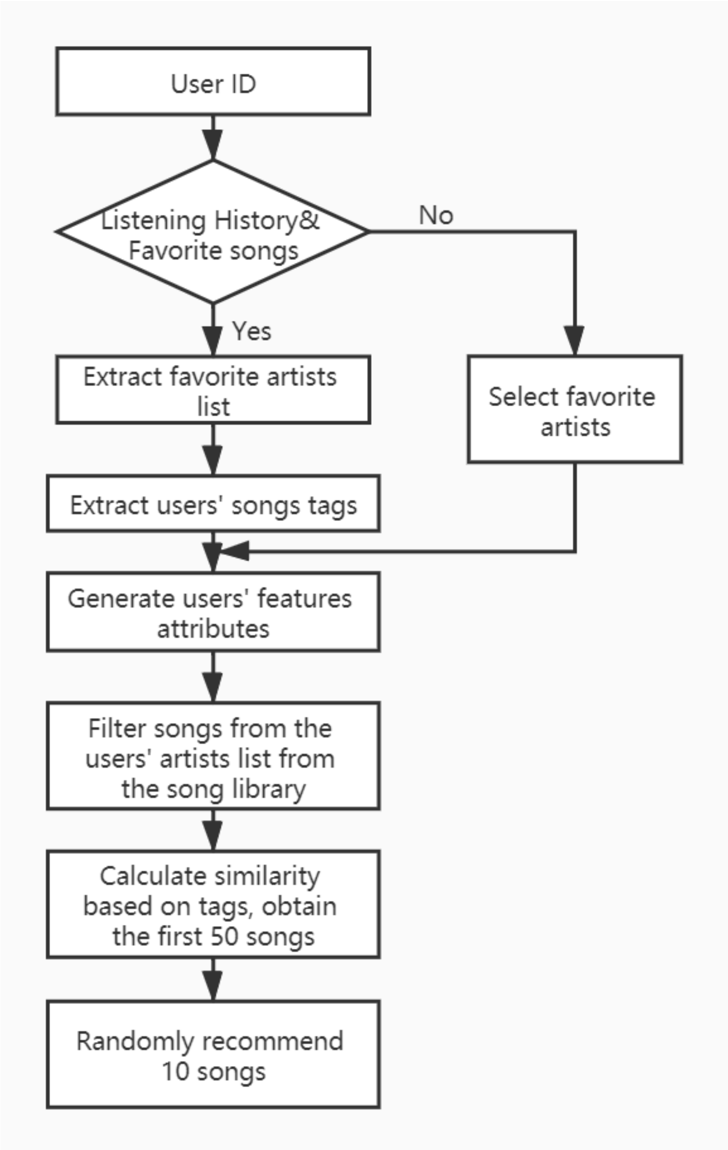

Figure 3 Flow Chart of Content-based

5.1.2 Design ideas and procedure of Content-based

1) If the user has not listened to the song previously and favorite song lists, let the user select the favorite artists and song tags.

2) If the user has a history of listening to songs and favorite song lists, extract a list of favorite artists and tags of favorite songs.

3) Combine the above data into user characteristic attributes, including the user’s artist list and user’s tag list.

4) According to the user's artist list, filter out the songs of these artists in the song library.

5) According to the user’s tag list, calculate the similarity of each song, and sort it, finally take the first 50 songs.

6) Randomly recommend songs to users from these 50 songs.

5.1.3 Operation results using Content-based

Input user's id 558900071 and the operation result is shown in figure 4.

```
---  Recommend Songs ---
[29535483 'Immortals (End Credit Version) [\\From \\"Big Hero 6”]"'
 'Fall Out Boy']
[551816010 '我们' '陈奕迅']
[108242 '她说' '林俊杰']
[30431364 '光' '陈粒']
[108493 '我还想她' '林俊杰']
[1400256289 '你的答案' '阿冗']
[451703096 'Shape of You' 'Ed Sheeran']
[543987451 'Way Back' 'Vicetone']
[190449 '吻别' '张学友']
[115502 '红日' '李克勤']
```

Figure 4 Content-based Operation Result

The first item is the id of song, the second is the name of the song, and the third is an artist.

5.1.4 The advantages and disadvantages of Content-based

Advantages:

1) Even if there is no historical data of the user listening to the song, the user also can be recommended, according to the artist and song tag selected by the user.

2) Increased randomness, in line with the daily listening habits. If the system recommends a song list every time, there will be randomness and will not always recommend the same song list to the user.

3) Recommends songs of artists that the user likes, which also fits most people 's preferences.

Disadvantages:

1) Recommended songs may have been listened to by the user previously.

2) The recommendations are popular songs. For relatively unpopular songs, due to a small number of tags, it is difficult for them to be recommended. For those who like to listen to unpopular songs, the system performs not particularly friendly.

5.2 Collaborative Filtering Recommender Systems

5.2.1 Flow Chart

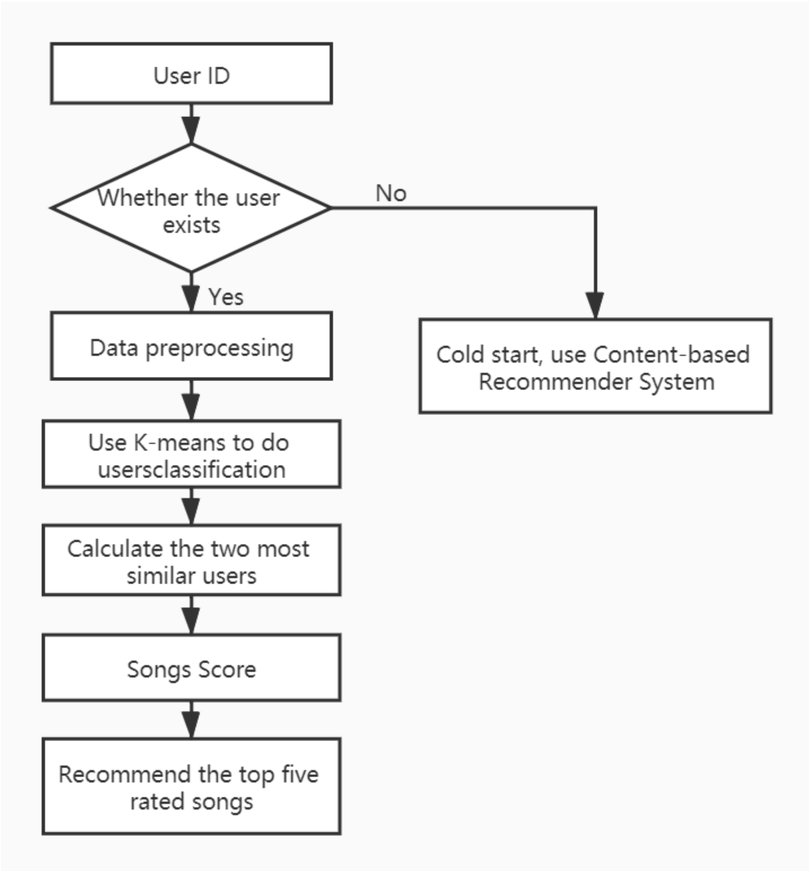

Figure 5 Collaborative Filtering

5.2.2 Design ideas and process of Collaborative Filtering

The Collaborative Filtering Recommendation System mainly uses user-based system filtering. User-based collaborative filtering mainly considers the similarity between users and users. As long as you find the items that similar users like and predict the target user's rating of the corresponding items, you can find the highest rated ones. Items are recommended to users.

For the data of NetEase cloud music songs, the K-means algorithm is used to modify the obtained data.

1) Data Processing

By reading the data from the CSV file, load the history data of each user that has listened to songs from NetEase Cloud Music captured on the network. Second, extract the users' id and songs information, such as songs' id, songs' name, artists, users' rating, and other information. Third, generate corresponding users-songs matrix for these data.

2) Data Modification

Since the historical data of users listening to songs is very huge, it will cause a considerable performance impact on the calculation of related user ratings afterward. Therefore, we use the K-means algorithm to initially process the song data.[8] When users are divided into one category by the K-means algorithm, we only perform collaborative filtering on this similar category of users to score songs.

3) User selection

For the target users of the recommended songs, by using the K-means algorithm, we select all users with the same label, calculate the similarity, and then pick the two most similar users. The user's similarity calculation method selects cosine similarity.[9]

4) Songs Scores

For the scores and rating of the final songs, we use user-based collaborative filtering, since we need to consider that the recommended songs that users have not been touched before. So we filter the songs that users have listened before and select only the songs that the users have not listened to before.

5.2.3 Operation Results using Collaborative Filtering

Input user's id 1999616317, and the operation result is shown in figure 6.

```
[31654747, '问', '简弘亦']
[584155, 'いつも何度でも', '伊藤サチコ']
[443794, 'あの夏へ', '久石譲']
[409650851, '霜雪千年 (Cover 洛天依 / 乐正绫) ', '双笙']
```

Figure 6 CF Operation Result

The first item is the song's id, second is the song's name and the third is an artist.

5.2.4 The advantages and disadvantages of Collaborative Filtering

Advantages:

1. The recommended results are more accurate because they are all recommendations from the same group.

2. The model is versatile.

Disadvantages:

1. If the user has new behaviors, such as listening to another tag ’s song recently, the system needs to be updated frequently, which requires high efficiency of the algorithm.

2. Cannot handle the cold start problem, so we can use Content-based Collaborative Filtering to solve it.

6. Results Visualization

We use a chatbot to allow users to query their own recommended music. The procedure is that firstly, if the user is a new subscriber, then the user can choose five favorite artists, and the Recommender System will recommend songs to the user. If the user is already in the users' library, then the user should input the user's ID and receive the recommended songs (Figure 7). The system is very intelligent and the recommendation is accurate. Combining the two algorithms, the recommendation process is improved reasonably. Using content-based solves the cold start problem of collaborative filtering.

Figure 7 Result Visualization

7. Conclusion

To summarize, The project we developed is a Recommender System to recommend songs to users based on NetEase Cloud Music. Reasonably combine Data Acquisition, Data Processing, Data Analysis using Content-Based as well as Collaborative Filtering algorithms, and Result Visualization. These steps have been completed, and more accurate recommendation results have been obtained.

8. Reference

[1] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56-58.

[2] Smith, B., & Linden, G. (2017). Two decades of recommender systems at amazon. com. *Ieee internet computing*, *21*(3), 12-18.

[3] Schedl, M., Knees, P., McFee, B., Bogdanov, D., & Kaminskas, M. (2015). Music recommender systems. In *Recommender systems handbook* (pp. 453-492). Springer, Boston, MA.

[4] Chodowiec, P., & Gaj, K. (2003, September). Very compact FPGA implementation of the AES algorithm. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 319-333). Springer, Berlin, Heidelberg.

[5] Soler, J., Boada, I., Prados, F., Poch, J., & Fabregat, R. (2010). A formative assessment tool for conceptual database design using UML class diagram. *International Journal of Emerging Technologies in Learning (iJET)*, *5*(3), 27-33.

[6] Aggarwal, C. C. (2016). Content-based recommender systems. In *Recommender Systems* (pp. 139-166). Springer, Cham.

[7] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

[8] Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In Icml (Vol. 1, pp. 577-584).

[9] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, *18*(3), 491-504.