Item-Based Collaborative Filtering Blockchain for Secure Movie Recommendation System

Shikhar Kumar Padhy, Ashutosh Kumar Singh, and P. Vetrivelan

Abstract A movie or web show or song recommendation plays a very important role in day to day daily life in giving a common person enhanced entertainment. Such a kind of technology has the ability to recommend a group of snaps to user according their likes and dislikes, or how much a movie or films is popular. But for an efficient, accurate and precise recommender system, large volumes of data along with high quality are required. As a result role of customer's privacy comes into play. Many companies try to accommodate these concerns with various tactics but are unable to provide high decipherment security. So we are proposing a system that tends to recommend films or movies to as a replacement user using the block-chain technology that will reduce misuse of privacy concerns and will in fact entice them to share more data. It finds and search movie databases available already to cluster all the important information about the preferences and movie, such as, popularity and attractiveness, required for recommendation. It generates films flocks not only comfortable for movie makers or producer to plan a replacement movie but also useful for movie recommendations for the user of a platform.

Keywords Recommendation · Block chain replacement · Entertainment · Databases · Efficiency

1 Introduction

A recommendation system may be defined as a subclass of data filtering system that seeks to predict ratings based on user's preferences for any item primarily utilized for commercial applications. The basic idea about a recommendation system is mostly

S. K. Padhy · A. K. Singh · P. Vetrivelan (✉)
School of Electronics Engineering, Vellore Institute of Technology, Chennai, India
e-mail: vetrivelan.p@vit.ac.in

S. K. Padhy
e-mail: shikharkumar.padhy2018@vitstudent.ac.in

A. K. Singh
e-mail: ashutoshkumarsingh.2018@vitstudent.ac.in

A. Sivasubramanian et al. (eds.), *Futuristic Communication and Network Technologies*,
Lecture Notes in Electrical Engineering 792,
https://doi.org/10.1007/978-981-16-4625-6_93

software or a program which has the ability to produce a list of videos, movies, films, songs, etc. [1]. Many companies are using this technology for their products such as YouTube for recommending a video to its user or a ecommerce company like Flipkart or Amazon to suggest a particular kind of product to its user. Some of the commonly used recommendation or suggestion systems are material-based and collaborative filtering recommender systems.

1.1 Collaborative Filtering

In collaborative based filtering, the recommendation is made upon the preferences of other users. For instance, person A liked a movie that movie will be recommended to other person B because of the reason that a person which has some connections with person B has liked it [2]. The simple and effective nature of memory-based technique makes it optimum to use from the latter. They are divided into two:

User-defined collaborative based filtering: In this model, suppose user A and user B liked a movie and a new movie comes out which is liked by person A then that movie is recommended to user B.
Item-based collaborative filtering: In this model, suppose user A gives 5 star ratings to two movies X and Y then another user B when watches movies X then movie Y is also recommended to him/her based on the similar ratings criteria.

Model-based methods are made and improved using data available and digging deep into it this process is also termed as data mining, algorithms of machine learning such as clustering and classification to predict users' rating [1, 3]. For this type of system, for better accuracy, dimensionality reduction techniques from Principal Component Analysis are used. Examples of such type of systems include Latent factor, Bayesian Model, Rule-based Model.

Content-based systems use metadata like genre, director, actor, singer to suggest things like films and songs. Such a suggestion would be for instance suggesting Inception that was performed by Leonardo Di Caprio because anyone saw and rated it high. The same thing happens in the case of music industry for certain artists such as Charlie Puth because you liked his Attention cover. The basic difference between the two filtering systems is explained in Fig. 1.

1.2 Block-Chain Implementation

Today, the importance of recommender systems has increased progressively as it assists people in examining decisions in a broader perspective. Especially in the areas of e-learning, e-library, e-government and e-business services it reduces transaction costs of finding and selecting items in an online and computerized environment.

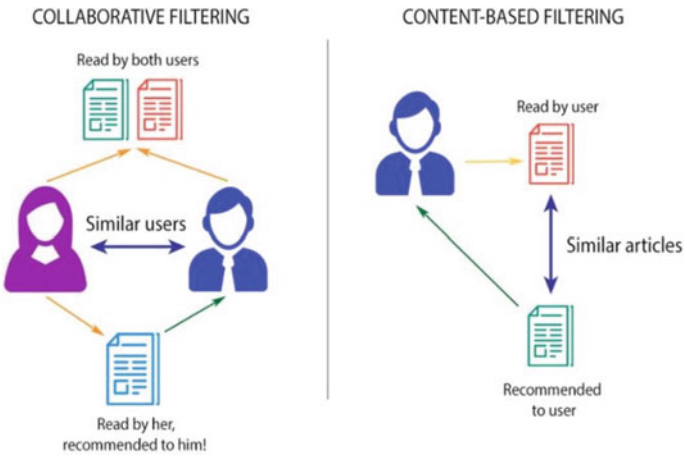
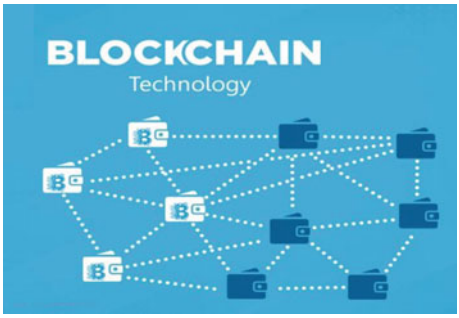


Fig. 1 Difference between collaborative and content-based Filtering

According to various marketing strategies, mass personalization helps in revenue augmentation as well as increasing customer gratification. As a result, companies tend to create a better customer interaction or behavior. Even if they know that the quality of recommendation will be increased, customers show reluctance toward sharing personal data because they fear fraud and misuse of their data. So, here we are using block-chain approach in order to store all the digital information in the form of blocks and the databases in the form of chain. Anchor aging block-chain technology enables the customers to be trustworthy of the recommender system and encourages them to fully provide more data to build it more efficient and improve its performance [4]. The contracts in the form of blocks are connected through databases in the form of chains is shown in Fig. 2.

The working is pretty simple. First the contract regarding how and what user's data are taken into account and are stored in the block-chain system by the company. Then the user can request for that contract, inspect it and allow for further sharing of his/her personal data. Finally the company can utilize that data as several input

Fig. 2 Block-chain technology



parameters to the collaborative filtering system and build the recommender engine [4]. The main advantage is that the user can stop or terminate from any stage at any time without completely disclosing the personal data [5].

2 Design and Methodology

2.1 Dataset and Importing Essential Libraries

For building our system, we have used the Movie Lens Dataset. We could use several datasets but we are using movies.csv and ratings.csv dataset from Kaggle. ***Here***'s the link to download our dataset that we have utilized in our System Project which consists of 105,339 ratings applied over 10,329 movies [1]. For this we need to install and load four packages or libraries—'reshape2', 'data.table', 'ggplot2' and 'reshape2'.

2.2 Data Retrieval

First we try to fetch our data points directly from movies.csv into movie data, data frame and ratings.csv into rating data and can make use of str, head(), summary() to display information about the movie data, data frame, view summary of the movies and first six lines of movie data, respectively. Similarly we can try it for the 'rating data' data frame [6].

2.3 Data Pre-processing

After giving head function for movie data the result we obtained is as shown in Fig. 3.

The table clearly describes about the integral values from user Id and movie Id column. Henceforth, it's important to change the genre present in the movie data data frame into a format that is more understandable by the users [6]. As there are limited number of genres so we will convert the string representing the genre into a numeric value using the one hot encoding method and we will create a matrix of it.

Now the task is to select the movies as easily as possible to do this genres that are already in the data list we will create a 'search matrix.' There are movies that have several genres, for example, Titanic, which is an American romantic film also falls under the genres of Thriller, Tragedy, Epic and Historical Drama [1].

Now our task is to pull the most different rating by creating a table and then we have to look for the movies that have been watched by most number of people in the database we have. Our first task is to count the number of views in a film

```
head(rating_data)
```

```
##   userId movieId rating  timestamp
## 1      1      16    4.0 1217897793
## 2      1      24    1.5 1217895807
## 3      1      32    4.0 1217896246
## 4      1      47    4.0 1217896556
## 5      1      50    4.0 1217896523
## 6      1     110    4.0 1217896150
```

Fig. 3 Top 6 movieId with their ratings

and then organize them in a data frame which will cluster them in descending order. Visualization of total number of views of top films could be carried out using ggplot2 function and finally visualized a heat map of the movie ratings that consisted of first 25 rows and 25 columns.

2.4 Data Preparation

Now our task is to prepare data in the given three steps.

2.5 Data Selection

From the dataset we tried to set the threshold for the minimum number of users who rated the film to 50. So it actually acts as a minimum number of views for a certain film resulting in filtering a list of watched films from the least-watched ones. And finally we will select those lists of films whose rating value is greater than the threshold value [3].

2.6 Data Normalization

There can be a case where some movies will have very high rating from the user and on the other hand some will have very low rating so in that case the movies with very high rating compared to others will have more effect on the mode so for that we have to normalize out data in a range so that a particular type of data does not have very high influence on the model compared to other data points [6, 3].

After normalizing the data points we then plot a heat map that delineates our normalized ratings which is visualized as given in Fig. 4.

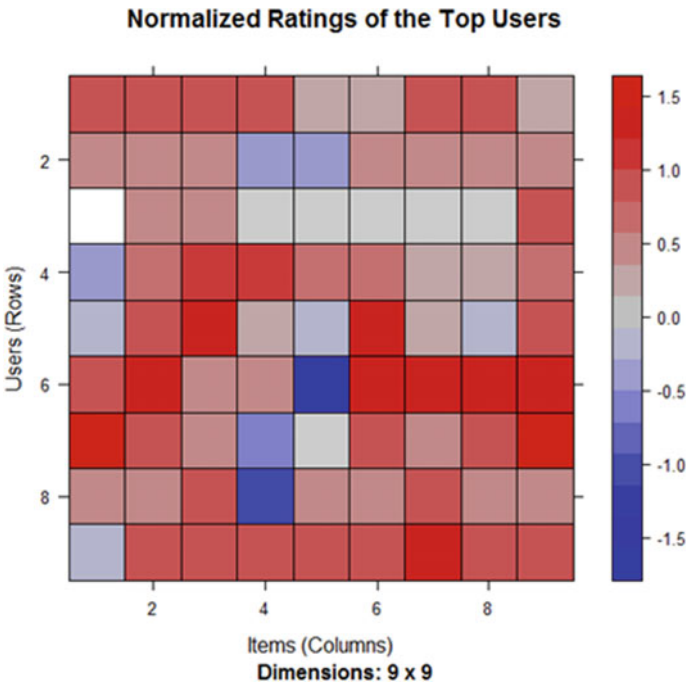


Fig. 4 Heat map normalized ratings of top users

2.7 Data Binarization

Now we will carry out with the data binarization process which basically means to convert the data in the form of discrete values 1 and 0 that will make the recommendation system more efficient. We will define a threshold value like 3 and then based on it will create a matrix that consists of 1 if the rating is above 3 otherwise it is 0.

2.8 Collaborative Filtering System

Here comes our most important part of this project in developing the recommendation engine i.e., Item-based Collaborative Filtering System. The definition of Item-Based Collaborative Filtering System is explained with the help of Fig. 5.

Basically what it does is it finds similarities between items based on people's ratings. Initially it creates a record of the customers who purchased similar items and then caters into the recommendation system. For instance, say, customer A purchases a product x and also product y[1, 2] Create a record or table of those items purchased

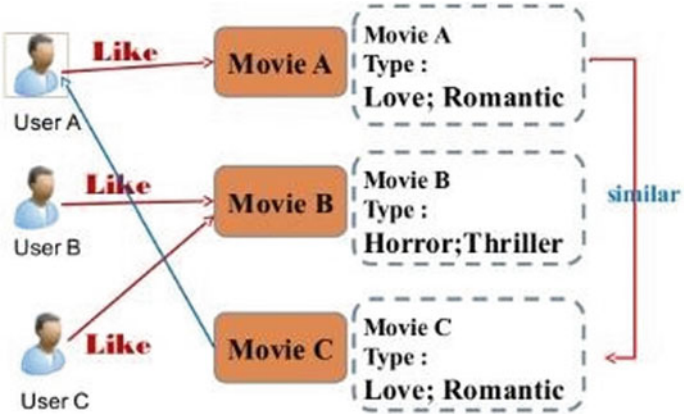


Fig. 5 Item-based collaborative filtering system

by that individual and then calculate the similarity. Figure 6 describes about the similarity between two movies watched by person A according to the Item-based collaborative filtering system.

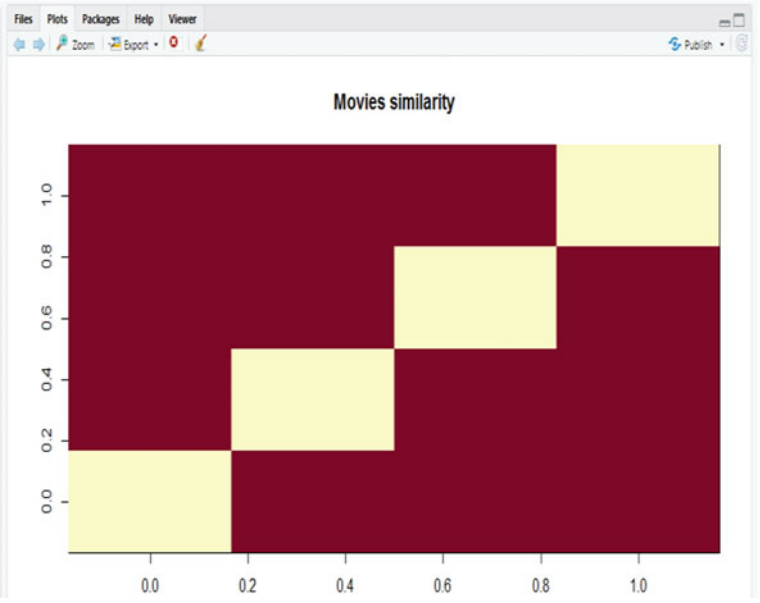


Fig. 6 Similarities between movies among different users

2.9 Building the Recommendation System Using R

Item-Based Collaborative Filter consists of parameters that are default in nature. So initially, let k denotes the number of items for computing their similarities. Here, we took k equal to 30 which primarily identifies the k most similar items and store it. There are various methods such as cosine, jaccard, pearson for the filtering process; we are going with the cosine function. Now, to retrieve the recommend model, we will use the getModel function. Then within model info we will try to find the class and dimensions of the similarity matrix and finally we will generate a heat map consisting of top 20 items and visualize the similarities shared between them. The next step is to create a variable initialized to 10 that keeps count of the movies watched by each user. To identify similar items we will use predict function and then rank them accordingly. And then each rating is multiplied with related similarities and finally adds them [1, 7].

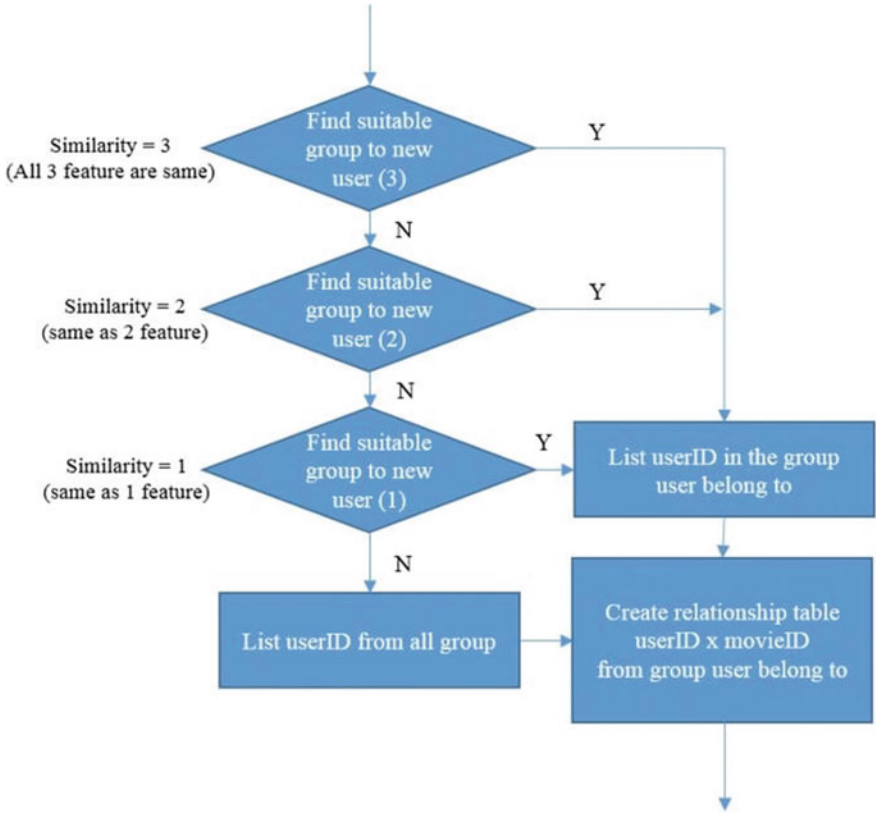


Fig. 7 Block diagram of the relationship table

Figure 7 describes the block diagram of the proposed K-NN algorithm used to create relationship table between user id and movie id from n clusters of data.

3 Steps and Implementation

This section discusses the platform and major modules used to implement the programming of the algorithm used to build the movie recommendation engine.

Processes
Step-1: Import the dataset.
Step-2: Data Exploration and Cleaning
Step-3: Merging the 2 csv files
Step-4: Working with genres column
Step-5: Working with the Cast column
Step-6: Working with directors column
Step-7: Working with Keywords column
Step-8: Similarity between movies
Step-9: Building the Recommendation engine

3.1 Algorithm for Item-Based Collaborative Filtering System

For user u, get items set N(u) that this user liked before. Recommend items which are similar to many items in N(u) to user u [2]. The necessary steps to be followed are as shown in Fig. 8.

4 Conclusion

Recommendation Engines are the foremost widespread sort of ML applications that is employed in various sectors. The evolution of these recommendation systems has drastically improved over time and has in fact assimilated several advanced ML techniques to provide the users with the content that they need [5]. In today's life, especially during this pandemic these engines have efficiently helped people to binge-watch movies according to their choices and preferences keeping them home quarantine.

However this system also has some disadvantages which obviously can be improved or upgraded by building a more advanced Memory-based cooperative filtering process where we basically divide the information database into a coaching set and a check set and then use techniques like circular function similarity to reckon

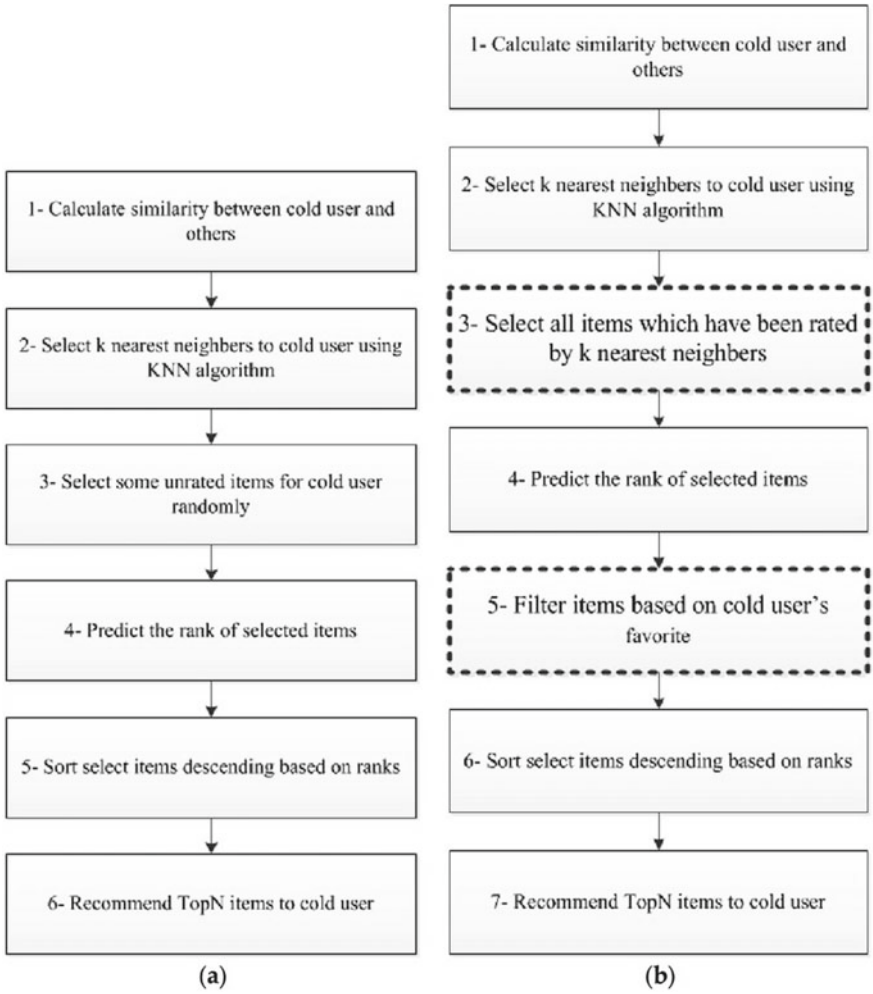


Fig. 8 Block diagram of selecting K-NN clusters of data

the similarity between the films. An alternate is to create a Model-based cooperative Filtering system. Matrix factoring is sweet at coping with quantifiability and scantiness than the previous [5].

We will then valuate your model mistreatment techniques like Root Mean square Error (RMSE).

Fig. 9 Snapshot of the final results of data analysis

5 Results and Discussion

The algorithmic program of collaborative filtering system used for filtering the films based on choices first builds a similar-items table of the purchasers who have bought them into a mixture of comparable things and the latter can then be provided to advice system[2, 7]. The similarity between single merchandise and connected merchandise will be determined with the subsequent algorithmic program–

- Every item x present in the product catalog, purchased by customer A.
- Also each item y purchased by the customer A.
- Create a record of the items x and y purchased by C.
- Calculate the similarity between i1 and i2.

Figure 9 shows our final result of Movie Recommendation Engine.

References

1. Data Flair Blogs, https://data-flair.training/blogs/data-science-r-movie-recommendation/Last. Accessed 13 September 2020
2. Yunkyoung Lee F (2005) Recommendation system using collaborative filtering. San Jose State University, pp 390–409
3. Andreas Vogl, R-Bloggers, https://www.r-bloggers.com/2020/04/movie-recommendation-with-recommenderlab/. Accessed 16 October 2020

4. Frey RM, Worner D, Alexander Illic F (2016) Collaborative filtering on the block chain: A secure recommender system for e-commerce. In: Americas Conference On Information Systems, pp 3–6
5. Lisi A, De Salve A, Mori P, Ricci LF (2019) A block chain based recommendation system. Institute Of Applied Sciences And Intelligent Systems, pp 7–9
6. RPubs, https://rpubs.com/jeknov/movieRec/last. Accessed 16 October 2020
7. Lisi A, De Salve A, Mori P, Ricci LF (2019) A smart contract based recommender system. In: Economics of grids, clouds, systems, and services. Lecture Notes in Computer Science, vol 11819, pp 1–13. Springer, Cham