How to Build a Trusty Movie Recommender System

Liqiang Pu, Chen Li, Ruoying Jiang

April 2, 2016

Abstract

Recommender systems or recommendation systems are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item.[9] They are an important part of the information and e-commerce ecosystem, representing a powerful method for enabling users to filter through large information and product spaces. Nowadays, the Recommender systems have impacted or even redefined our lives in many ways. [2]One example of this impact is how our online watching experience is being redefined. As we browse through various movies, the Recommender system offer recommendations of movies we might be interested in. Regardless of the perspective — business or consumer, Recommendation systems have been immensely beneficial.

In the report, we explore the Movie Lens database, which collects millions of users' rating records for the movies they have watched and their personal information. In this project, we only focus on 100K dataset from the Movie Lens database and the recommenderlab package in R. By dividing users into two parts (new user / old user), try to build a trusty Recommender system based on the characteristics and preferences of users.

1. Introduction

Recommender systems were originally defined as ones in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients" (Resnick & Varian 1997). The term now has a broader connotation, describing any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options. [11]Such a system is obviously appeal to the e-commerce time, since information is vastly sharing online. Instead of relying on recommendations from peers or experts, nowadays Computer-based systems provide the opportunity to expand the set of people from whom users can obtain recommendations. They also enable us to mine users' history and stated preferences for patterns that neither they nor their acquaintances identify, potentially providing a more finely-tuned selection experience. [4]

There is also a growing interest in problems surrounding recommendation. The first thing we should emphasize on the building of a Recommender system is the diversification of users. Our Recommender system is of no help unless we take users' experience and preference into fully consideration. A recommender system must interact with the user, both to learn the user's preferences and provide recommendations [17].

1.1 Main Procedures in Building System

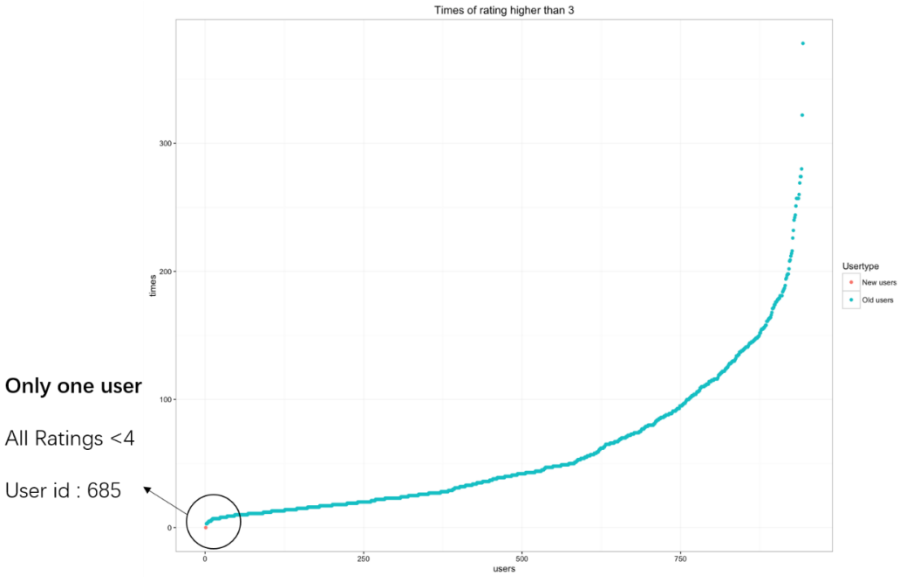
First we divide people into two parts: new users and old users. For those new users, people who have no movie records on this website, their preference on movies are unknown. However, according to other users' movie records and rating data, we can deduce those new users' preference from the sharing characteristics of people. For example, ages, genders, occupations. It is assumed that people of similar characteristics might have similar preferences. We then can find different patterns for different groups and fit models to predict new users' preference on movies and perform the recommendation. Another more intuitively way is to recommend movies by other users' historical data. Movie that has been rated highest and watched most times should be recommended to others. For those older users, people who have already rated some movies before, we'll mainly concentrate on their ratings and watching history. Here we have two methods as well: The first one a content-based recommendation, only focus on the similarity of movies—finding movies similar to the ones liked by a user using textual similarity. And to determine which movie the user actually likes, we should first consider the rating records, filter out those low-rating movies. The second one is a system designed according to the collaborative recommendation paradigm identify users whose preferences are similar to those of the given user and recommend items they have liked.[5] By dividing people into different clusters, people in the same cluster might have similar preference on movies. We'll recommend movies based on group preference. Finally, for those old users, we will combine the results of these two methods together to yield better recommendations across the board.

The main difference between old users and new users is that we can get the information of movie preference directly by using the ratings, which means we need to make sure that every old user provides enough information. As for the Movie Lens data, we have a filter for classification.

(i) The user has rated over 10 movies and these ratings contain 4 or 5.

(ii) The user provides any information about age, gender and occupation.

The first condition above ensures that the user has enough rating information including preferred movies, which is needed to apply the method for old users. When (i) isn't satisfied, we may consider the target user as a new user even if he has rating information (ratings are all below 4, contains no preference information). As for the new users, if (ii) is satisfied, then we'll use new user method with personal information, if not, we'll just recommend the most popular movies.



Only one user

All Ratings <4

User id : 685

Figure 1

Times of ratings that are higher than 3 for all the users

In the 100K data, among 943 users, only one user has no rating over 3 and we'll treat him as "new user" even he has enough rating information. By the way, we may also user the information of users' dislike, namely, ratings below 3 to avoid recommend "bad" movies. We'll discuss that in the challenging part.

2. Recommendations for New Users

Most Recommendation algorithms use input about a customer's interests to generate a list of recommended items. [7] Thus, The user has to rate a sufficient number of items before an information-based recommender system can really understand the user's preferences and present the user with reliable recommendations. As a consequence, a new user, having no or very few ratings, would not be able to get accurate recommendations. Therefore, our group first wants to solve this problem, knows as "cold-start" problem, by recommending movies for new users.

2.1 New Users with no personal information

We assume people in this cluster have no personal information (i.e. age, gender, occupation) and rating histories in the rating dataset. Thus, the method we employed for them are based on the popularity and average ratings of all the other old users. Here, the popularity means movies with at least 10 % users (for our data, the value is 95) in the rating dataset have rated. The common thread is to recommend the "popular" movies with highest average ratings.

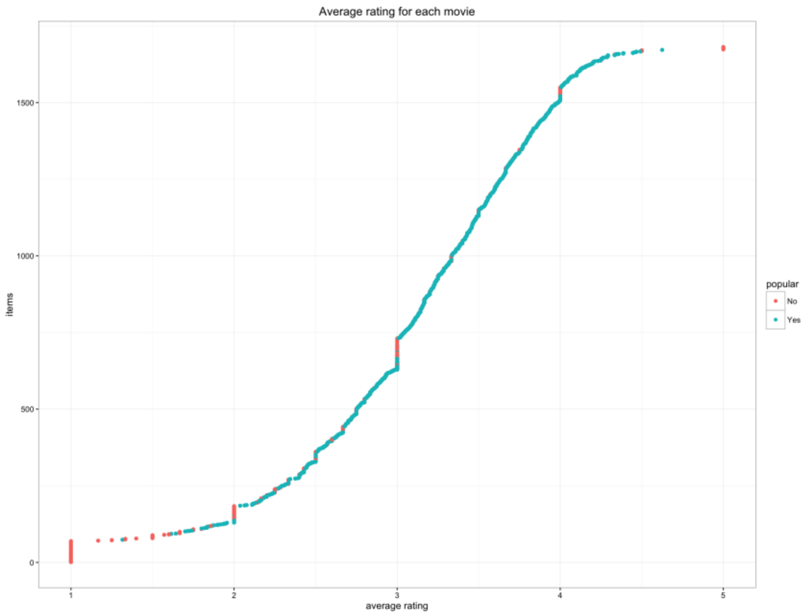


Figure 2

The average ratings for all the movies in the dataset, defined as popular or unpopular.

According to figure 2, even though the highest average rating score of movies in the dataset is close to 5, our method would not recommend it since it is not a “popular” movie. Instead, we should recommend the other high-rating movies without this movie.

Through this method, our recommending movies for all the new users with no personal information and rating records are as follows.

Recommend list for No information users:

- Citizen Kane (1941)
- 12 Angry Men (1957)
- Star Wars (1977)
- Usual Suspects, The (1995)
- Rear Window (1954)
- Shawshank Redemption, The (1994)
- Casablanca (1942)
- Wrong Trousers, The (1993)
- Schindler's List (1993)
- Close Shave, A (1995)

2.2 New Users with personal information

We assume people in this cluster have filled in some personal information (age, gender, occupation) online, but still have no rating histories for movies.

Since now we have some user information, the Recommender system can do more to guide users in a personalized way to interesting or useful objects in a large space of possible options. Therefore, we should recommend movies based on the popularity and rating histories of other old users that share the identical characteristics with those new users.

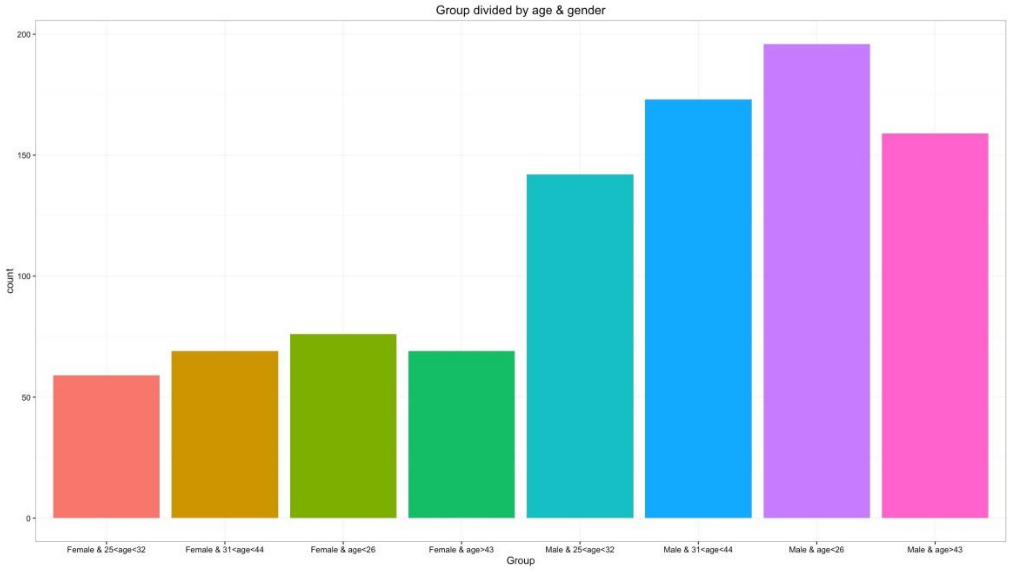


Figure 3

Divide new users with personal information into 8 groups by age and gender

The figure 3 presents an intuitively grouping by dividing the old users into 8 clusters according to their age (Quantiles) and gender (Male & Female). It makes sense because we believe that age and gender would have big influence on the recommend list. For example, young age people would give higher ratings on Animation movies while Drama may not be appeal to this age group. Also female may prefer Romance movies to Fiction movies. Though occupation may have some effects on the recommend list, two people may have similar preference with same age and gender even their occupation are not the same.

Therefore, if we know new user’s age and gender, we will recommend movies based on the ratings of old users that are in the same group. Find out those movies that are popular and high-rating for this group, make recommendations to the new users.

Similarly, if we only know the new user’s age (gender), then we will recommend movies based on all the group members that have the same age (gender). If we only know new user’s occupation, we will recommend movies according to all the old users in the rating dataset with same occupation.

Here, we use an example to explain the second method.

Assuming a new user, who has age =20, gender =Female, occupation = student, and we know all/part information of her.

Personal Information			Recommending movies’ ID									
age	gender	occupation	movie1	movie2	movie3	movie4	movie5	movie6	movie7	movie8	movie9	movie10
20	F	student	5	276	1104	1217	1283	1319	1400	1509	1575	1655
20	F	NA	5	276	1104	1217	1283	1319	1400	1509	1575	1655
NA	F	student	276	306	1104	1217	1283	1319	1331	1400	1509	1575
20	NA	student	5	276	306	321	1217	1283	1319	1400	1575	1655
NA	NA	student	276	306	321	1217	1283	1319	1400	1509	1575	1655

Table 1

The test results for one user with age=20, gender=Female, occupation=student, while we know all or partly characteristics of her.

According to table 1, the recommend lists will be identical for users of same age and gender, no matter what occupation it is. And there is slightly difference when we only know one of the characteristics, i.e. age, gender and occupation.
Using R, the Recommend list for new users with age =20 & gender =Female are:

- One Flew Over the Cuckoo's Nest (1975)
- 12 Angry Men (1957)
- Usual Suspects, The (1995)
- Shawshank Redemption, The (1994)
- Wrong Trousers, The (1993)
- To Kill a Mockingbird (1962)
- Star Wars (1977)
- Rear Window (1954)
- Casablanca (1942)
- Schindler's List (1993)

The providing of user information is a complement to the first method. And it helps to solve the “Rating Sparsity” problem. For example, in the movie recommendation system, there may be some users whose tastes are unusual compared to the rest of the population, there will not be any other users who are particularly similar, leading to poor recommendations. [13]

3. Old Users

According to the filter we define above, only the users that have rated enough movies and show their preference from the ratings are regarded as old users. Compared to user’s information, it is easier for us to find people’s preference on movies based on the ratings. Here we’ll use the collaborative method. The fundamental assumption behind this method is that other users’ opinions can be selected and aggregated in such a way as to provide a reasonable prediction of the active user’s preference. Intuitively, we may assume that, if some users agree about the quality or relevance of some items, then they will likely agree about other items. Here, UBCF and MBCF are applied to find the recommend movies. The two approaches have the same basic idea: similarity. When using UBCF, we suppose that people have similar ratings on movies should have similar tastes. The MBCF method is mainly focusing on the similarity of movies. Besides using the rating information, the movie type data can also be used to make the results better.

3.1 UBCF (user-based collaborative filtering)

User-based collaborative filtering predicts a test user's interest in a test item based on rating information from similar users, namely, find other users whose past rating behavior is similar to that of the current user and use their ratings on other items to predict what the current user will like. Now we are faced with an important question, how to define two users are similar to each other. It is necessary for us to measure the similarity between users. After learning about the similarity functions, we decide to user the Cosine similarity. It is simple and easy to understand. Being different from other approaches, the rating information of users is represented vectors and similarity is measured by the cosine distance between two rating vectors. We have the formula, where $r_{u,i}$ represents the rating of user u on the ith movie.

$$s(u,v) = \frac{\mathbf{r}_u \cdot \mathbf{r}_v}{\|\mathbf{r}_u\|_2 \|\mathbf{r}_v\|_2} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2}\sqrt{\sum_i r_{v,i}^2}}$$

For the Movie Lens data, we first construct a rating matrix whose rows are user IDs and columns are movie IDs. We notice that there are some NAs in this matrix and we replace these ratings by 0 which means unrated movie. Suppose now we are recommending movies for the kth user, then we can use the ratings of this user which is the kth row of the rating matrix to do matrix multiplication with the whole matrix to get the similarity with all the users (include himself). As for the problem of finding the similar users based on similarity, we have two solutions. One is using threshold and another one is selecting the top-N users. However, if we use the top-N method, we may mistakenly select the users that are not "similar" enough when all the similarity values are low, this may give us bad recommendations. Thus it's more reliable to use threshold.

3.2 MBCF (movie-based collaborative filtering)

Rather than using similarities between users' rating behavior to predict preferences, movie CF uses similarities between the rating of movies. If two movies tend to have the same users like and dislike them, then they are similar and users are expected to have similar preferences for similar items. For every movie, we need to find its "neighbors", which are the movies have similar ratings. Again, we use cosine distance to measure the similarity between movies and use threshold to find the movies that are supposed to recommend to our target user.

3.3 Movie Type Selecting

As mentioned above, we may use the movie types data to find the preferred movie types of the target user. Here we use the rating matrix and the movie-type matrix to do the matrix multiplication and find the average rating for each movie type. Then we can find this user’s favorite movie types based on the result.

Example:

We here provide an example. Suppose we’d like to recommend No.300 user some movies. Because this user has rated many movies and shows his or her preference through the ratings, we regard this user as old user. First use the UBCF method with threshold equals to 0.25 and we find the “similar” users whose IDs are

35 126 155 166 170 304 341 451 482 510 511 644 725 783 810 812 816 8
32

Then we pick out the rating information of these users and try to find the most commonly preferred movies by this group (we also need to make sure that the results do not contain the movies that have already been rated by No.300 user). Here is the result.

- Antonia's Line (1995)
- Fargo (1996)
- Independence Day (ID4) (1996)
- Jerry Maguire (1996)
- Men in Black (1997)
- Starship Troopers (1997)
- Face/Off (1997)
- Air Force One (1997)
- In & Out (1997)
- Devil's Advocate, The (1997)
- Titanic (1997)
- Deceiver (1997)
- Mad City (1997)
- Boogie Nights (1997)
- Man Who Knew Too Little, The (1997)
- Wag the Dog (1997)
- Deep Rising (1998)
- Big Lebowski, The (1998)
- Shadow Conspiracy (1997)
- Hugo Pool (1997)

As for the MBCF method, we first find the movies that No.300 user like, here we define ratings over 3 as “like”. These are the preferred movie IDs,

243 257 288 300 322 409 456 833 872 876 881 948 1012 1094

Then we apply MBCF to find the similar movies of these movies. And choose most popular ones in this cluster.

- Star Wars (1977)
- Rock, The (1996)
- Independence Day (ID4) (1996)
- Mission: Impossible (1996)
- Broken Arrow (1996)
- Ransom (1996)
- Scream (1996)
- Jungle2Jungle (1997)
- Men in Black (1997)
- Contact (1997)
- Liar Liar (1997)
- Air Force One (1997)
- Murder at 1600 (1997)
- Conspiracy Theory (1997)
- Saint, The (1997)
- Money Talks (1997)
- Booty Call (1997)

Combine the two results above, we get our final recommend movies:

- Independence Day (ID4) (1996)

- Men in Black (1997)
- Air Force One (1997)

We can recommend more movies by reducing the threshold of the CF method. Now let’s look at the favorite movie types of this user. By using the method mentioned above, we can get the following plot,

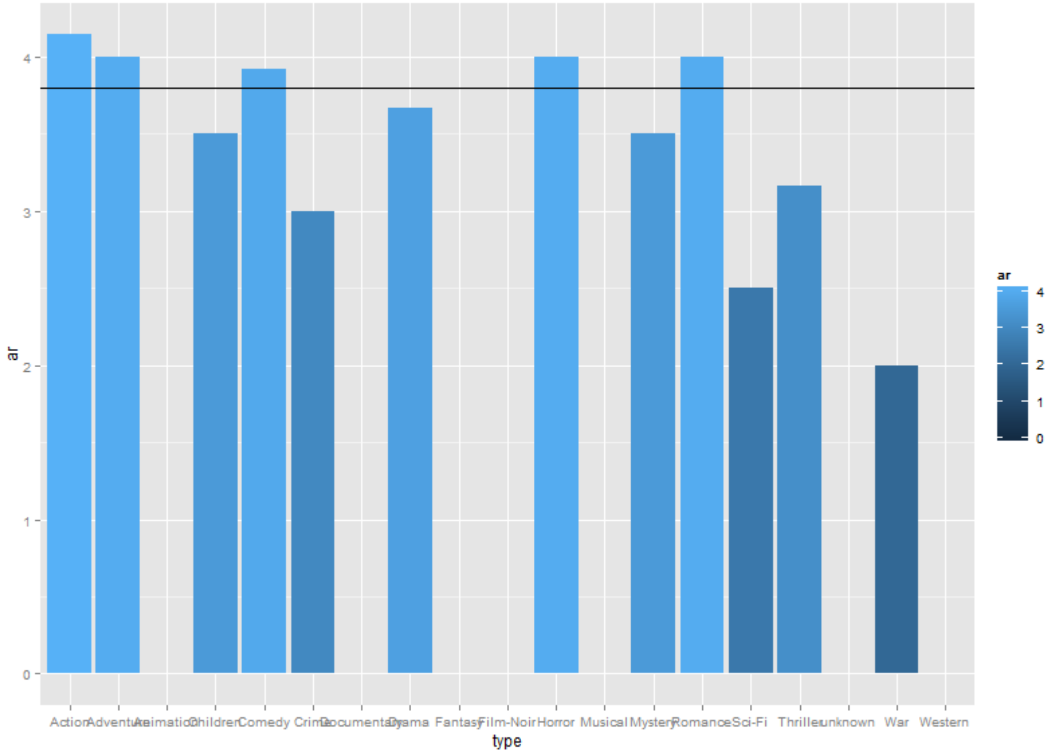


Figure 4

By setting the threshold to be 3.8 (the Horizontal line.), we get the following types which are preferred by No.300 user,

Action Adventure Crime War

As for the movies we recommend above, we can check their movie type info rmation:

Movie	Name	Action	Adventure	Crime	War
121	Independence Day (ID4) (1996)	1	0	0	1
257	Men in Black (1997)	1	1	0	0
300	Air Force One (1997)	1	0	0	0

Table 2

As we can see, these three movies all belongs to action movies and some of them have the war and adventure features. Thus it is quite reliable to recommend these movies to No.300 user.

4. Conclusions and Future Work

There has been much research done on recommendation technologies over the past several years that have used a broad range of statistical, machine learning, information retrieval, and other techniques that have significantly advanced the Recommender systems. What we have done is just a small work based on the 100k dataset of MovieLens, therefore the accuracy and representativeness cannot be secured. Specifically speaking, in the part of recommending for new users, we mainly rely on the item popularity, user personalization, and combination of these strategies to determine the best movies for a new user. The two methods we mentioned above will generate better results since the problem of “rating sparsity” has been overcome by utilizing user profile information when calculating user similarity. And according to our definition of popularity, for a responding group, at least 10 % of people in this group should have rated on the recommend movies, in the situation that we have some ideas about the users’ characteristics. Whereas, to some extreme extent, which the group size is small (e.g. size=21), it’s possible that we would recommend none of the movies in the dataset since all of them are not “popular”. Or if the group size is too small (e.g. size= 5), all the rating movies in this group will be “popular”. Hence, it will lead to incorrect recommendations if the group members in the dataset are insufficient. There, for our dataset, which only includes 943 users, we require the minimum group size to be 30.

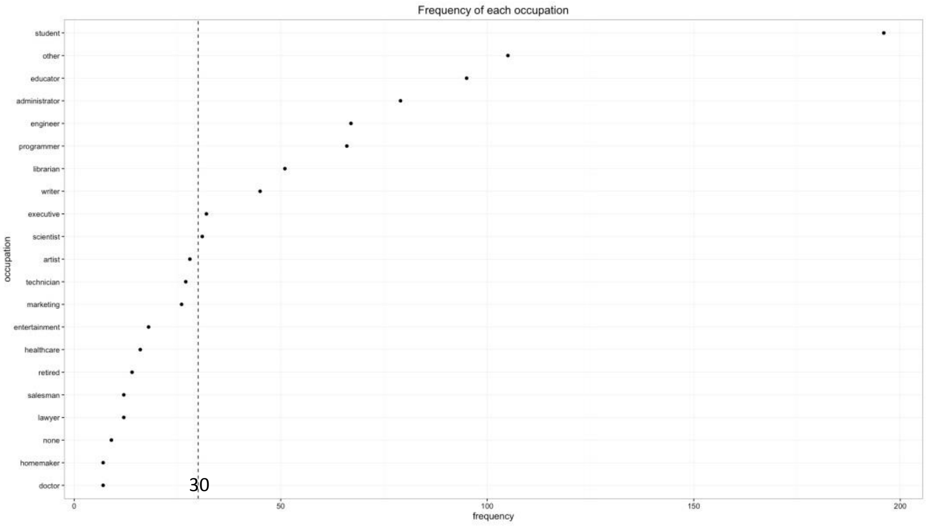


Figure 5

From the figure 5, if we only know new user is a doctor, we can’t recommend movies according to doctor group due to the small group size, and we should use the new user method with no personal information.

For further research, in order to make more accurate recommendations, the system should obtain more information about the users to deduce the interests and preference, we might use the CHTC (Center for High Throughout Computing) services to expand our analysis.

Reference

[1] Linden G, Smith B, York J.Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE 2003.p. 76-80.

[2] Kaushik Pal, TechAlpine, “How big data is used in Recommender Systems to change our lives”.

[3] Jyoti, Dhawan S, Singh K. Analyzing User Ratings for Classifying Online Movie Data using Various Classifiers to Generate Recommendations. In proc. ABLAZE, IEEE 2015.p. 295-300

[4] J.A. Konstan, J. Riedl, A. Borchers, and J.L. Herlocker, “Recommender Systems: A GroupLens Perspective,” Proc. Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08, 1998.

[5] Balabanovic, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. Communications of the ACM 40(3), 66–72 (1997).

[6] Paul Resnick and Hal R. Varian, “Recommender systems. (Special Section: Recommender Systems)” Communications of the ACM, March 1997 v40 n3 p56(3)

[7] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro, “Content-based Recommender Systems: State of the Art and Trends” © Springer Science+Business Media, LLC 2011.

[8] Sanjeev Dhawana , Kulvinder Singhb , Jyotic, “High Rating Recent Preferences Based Recommendation System”, Procedia Computer Science 70 (2015) 259 – 264.

[9] Francesco Ricci, Lior Rokach and Bracha Shapira, “Introduction to Recommender Systems Handbook”, Accessed from web on April 2, 2016.

[10] Jyoti, Dev P, Dhawan S, Singh K. Automotive Tools for Making Effective Recommendations for E-commerce Websites: An In-depth Comparative Study. IOSR-JCE 2015. p. 53-58.

[11] Robin Burke, California State University, Fullerton Department of Information Systems and Decision Sciences, “Hybrid Recommender Systems: Survey and Experiments”.

[12] Chuang H, Wang L, Pan C.A Study on the Comparison between Content-based and Preference-based Recommendation Systems. In proc. SKG, IEEE 2008.p. 477-480.

[13] M. Balabanovic and Y. Shoham, “Fab: Content-Based, Collaborative Recommendation,” Comm. ACM, vol. 40, no. 3, pp. 66-72, 1997.

[14] L. Si and R. Jin, “Flexible Mixture Model for Collaborative Filtering,” Proc. 20th Int’l Conf. Machine Learning, Aug. 2003.

[15] R.J. Mooney and L. Roy, “Content-Based Book Recommending Using Learning for Text Categorization,” Proc. ACM SIGIR ’99 Workshop Recommender Systems: Algorithms and Evaluation, 1999.

[16] D. Billsus and M. Pazzani, “Learning Collaborative Information Filters,” Proc. Int’l Conf. Machine Learning, 1998.

[17] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, “Collaborative Filtering Recommender Systems”.

[18] Jun Wang, Arjen P. de Vries , Marcel J.T. Reinders,Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion, Information and Communication Theory Group