A Major Project-II Report

On

A collaborative filtering-based recommender system alleviating cold start problem

Submitted in Partial fulfilment of the Requirement for the Degree of

Master of Technology

in

Computer Science and Engineering

Submitted By

Ishan Rathi

2K17/CSE/05

Under the Guidance of

Mr. Manoj Sethi

(Associate Professor)

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahabad Daulatpur, Main Bawana Road, Delhi-110042

June 2019

CERTIFICATE

This is to certify that Project Report entitled **"A collaborative filtering-based recommender system alleviating cold start problem"** submitted by **Ishan Rathi** (2K17/CSE/05) in partial fulfilment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the original work carried out by him under my supervision.

Project Guide
Mr. Manoj Sethi
Associate Professor
Department of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the Major Project-II work entitled **"A collaborative filtering-based recommender system alleviating cold start problem"** which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master of Technology (Computer Science and Engineering) is a bona fide report of Major Project-II carried out by me. I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Ishan Rathi
2K17/CSE/05

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Mr. Manoj Sethi for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. Rajni Jindal, HOD, Computer Science & Engineering Department, DTU for her immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Ishan Rathi
Roll No – 2K17/CSE/05
M. Tech (Computer Science & Engineering)
Delhi Technological University

TABLE OF CONTENTS

Chapter 1 Introduction 10

1.1 Model for recommendation system 11

1.2 Application of recommendation system 12

1.2.1 Product recommendation 12.

1.2.2 Movie recommendation 12

1.2.3 News recommendation 12

1.3 Utility matrix 13

1.4 Populating the utility matrix 14

Chapter 2 Literature survey 15

Chapter 3 Collaborative filtering 19

3.1 Data collection 20

3.2 Pre-processing steps 20

3.2.1 Data cleaning 20

3.2.2 Generation of implicit ratings 21

3.2.3 Data integration 21

3.3 Collaborative filtering 21

3.4 Approaches to collaborative filtering 21

3.4.1 Memory based CF algorithms 22

3.4.2 Similarity metric 22

3.4.3 User based approach 25

3.4.4 Item based approach 27

Chapter 4 Challenges in recommendation systems 28

Chapter 5 Cold Start Problem 30

5.1 Popularity strategy 30

5.2 Random strategy 31

5.3 Pure entropy 31

5.4 HELF 32

5.5 Balanced strategy 32

5.6 Item based cold start problem 32

5.7 K-means clustering 36

Chapter 6 Proposed solution 38

6.1 Proposed algorithm 42

Chapter 7 Implementation and results 43

7.1 Implementation details 43

7.2 Results 44

Chapter 8 Conclusion 47

REFERENCES 48

APPENDIX A USER INTERFACE CODE 52

APPENDIX B PROGRAM OUTPUT SCREENSHOT 54

LIST OF FIGURES

FIG 1: LONG TAIL 12

FIG 2: UTILITY MATRIX 13

FIG 3: RATING OR UTILITY MATRIX 19

FIG 4: CLASSIFICATION OF CF TECHNIQUES 21

FIG 5: SAMPLE RATING MATRIX 24

FIG 6: USER BASED PEARSON CORRELATION 24

FIG 7: ITEM BASED PEARSON CORRELATION 25

FIG 8: USER BASED CF PROCESS 26

FIG 9: FUNCTION ENTROPY 31

FIG 10: IBCTAP ALGORITHM 34

FIG 11: DECISION TREE FOR NEW ITEM RECOMMENDATION 35

FIG 12: K-MEANS ALGORITHM 37

FIG 13: K-MEANS ALGORITHM ITERATION 37

FIG 14: PROPOSED SOLUTION 38

FIG 15: VISUAL REPRESENTATION AVERAGE RATINGS 40

FIG 16: K-MEANS ON AVERAGE RATINGS 41

FIG 17: K-MEANS WITH THREE CLUSTERS 41

FIG 18: INTERSECTION AMONG MULTIPLE CLUSTERS 42

FIG 19: PLOT FOR AVERAGE ROMANCE USER RATINGS 44

FIG 20: PLOT FOR AVERAGE DRAME USER RATINGS 45

FIG 21: PLOT FOR AVERAGE CRIME USER RATINGS 46

LIST OF TABLES

TABLE1: COMPARATIVE STUDY ON COLD START PROBLEM ....……16
TABLE 2:COMPARATIVE STUDY BETWEEN USER BASED AND ITEM BASED MEMORY CF MODEL .27
TABLE 3: CHALLENGES IN RECOMMENDATION SYSTEMS 29
TABLE 4: MOVIE DATASET 39
TABLE 5: USER RATINGS DATASET 39
TABLE 6: AVERAGE USER RATINGS 40
TABLE 7: MOVIE DATASET 43
TABLE 9: USER RATINGS DATASET ………………………………………………………..43

LIST OF ABBREVIATIONS

1. CF : Collaborative Filtering
2. RS : Recommendation system
3. SA : Shilling attacks

ABSTRACT

Consumers currently have a surplus of items available to purchase via online stores. Surplus of goods enables users to have huge variety but it often leads to inconvenience for users. Consumers have to spend a lot of time going through items to find goods of their preference. To automate the process of sharing relevant suggestions, recommender systems are used.

Recommender systems are making their presence felt in a number of domains, be it for ecommerce or education, social networking etc. With huge growth in number of consumers and items in recent years, recommender systems face some key challenges. These are: producing high quality recommendations and performing many recommendations per second for millions of consumers and items. New recommender system technologies are needed to scale themselves for new items as well as in new user in the system in order to get high quality recommendations.

In this thesis, we focus on collaborative approach-based recommender systems to solve the issue of cold start problem. We have compared multiple algorithms which aim to solve cold start problem and proposed a new hybrid algorithm. This new algorithm is implemented on Movie-Lens 1Million Dataset.

Keyword: Recommender systems, Hybrid algorithm, Collaborative, Movie-Lens

CHAPTER 1: INTRODUCTION

As the amount of information is increased over the web, users have many options. This is known as Information Overload.

Web Applications which predict user response based on certain parameters are called Recommendation systems. To understand Recommendation system two examples are mentioned below.

1. Offering online newspaper readers news articles based on a reader interest prediction.

2. Providing online retailer customers with recommendations about what they might want to b uy based on their past purchase history or product search.

RS can be categorized in two major categories content-based and collaborative filtering system

Content- Based Systems

These systems analyse the features of recommended items. For example, an Amazon prime video user has viewed a number of gangster movies, then "gunslinger" genre movies should be recommended to him from database in the database.

Collaborative-Filtering Systems

Based on similarity metrics between users or items, collaborative filtering systems recommend items. The items which are recommended to a user are the ones that similar users prefer for example a collaborative filtering recommendation system for Amazon prime video tastes could predict which Amazon prime show a user would like given a list user’s likes and dislikes.

1.1 Model for Recommendation Systems

The **"long-tail"** idea, which describes how web stores make more profit than conventional, brick and mortar suppliers. We will then examine the kind of applications that have been proved successful in recommendation systems.

Let's examine the long tail idea which makes recommendations systems necessary before we discuss the main applications of recommending systems.

There are limited shelf space in brick and mortar stores, which can show the customer just a small portion of the choices. On the other hand, online stores can provide the customer with everything that exists. Therefore there can be hundreds of books in a physical bookshop compared to millions of books in online book stores. An online news service provides thousands of articles per day compared to only handful provided by any physical media such as journal, magazines or newspapers.

It is quite simple to recommend in the physical world. First, customers can not personalize the store. Therefore, the choice of what is available is decided by few individuals only. In general, only the most popular books will be displayed in a book store, and only items which majority of people have interest in will be printed in the newspaper.

The difference in the physical and online worlds is known as the long-tail phenomenon. Visualization of this difference is present in Fig. 1. The vertical axis represents no. of times an item is selected. The horizontal axis represents the popularity of items in increasing order. Online Institutions provide items present on both sides of the vertical line whereas the physical institute provide items which are popular hence items left to the vertical line. Due to ample no of choices in on-line institutions each customer cannot possible browse through all the choices and will need some guidance therefore recommendation systems become necessary in such situation.

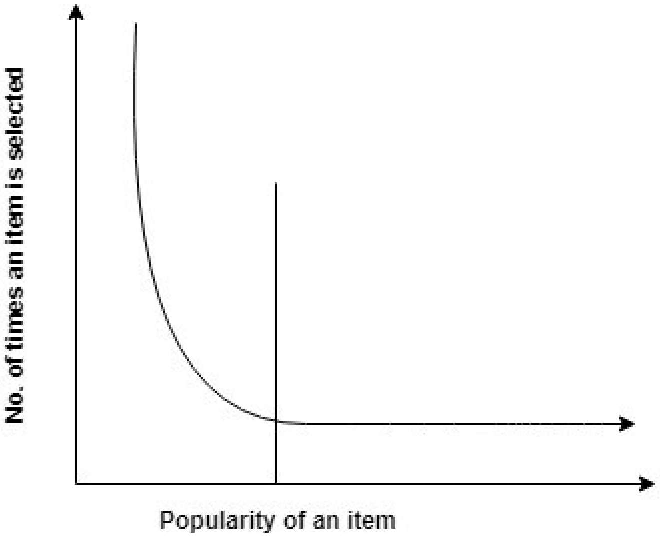


Figure 1. Long tail

1.2 Applications of Recommender systems

1.2.1 Product Recommendation

Online stores are the most important use of recommendation systems. We noted how Flipkart or similar online stores aim to offer each user some product suggestions. These suggestions are based on similar customers purchase history or some other method.

1.2.2 Movie Recommendation

Amazon prime offers recommendations for films which its customers like. These recommendations are based on user ratings, Amazon prime video has suggestions based on the web series or movie you just watched. However, it is different than how Netflix does it. It based on, "Customers Who Watched This Item Also Watched."

1.2.3 News Recommendation

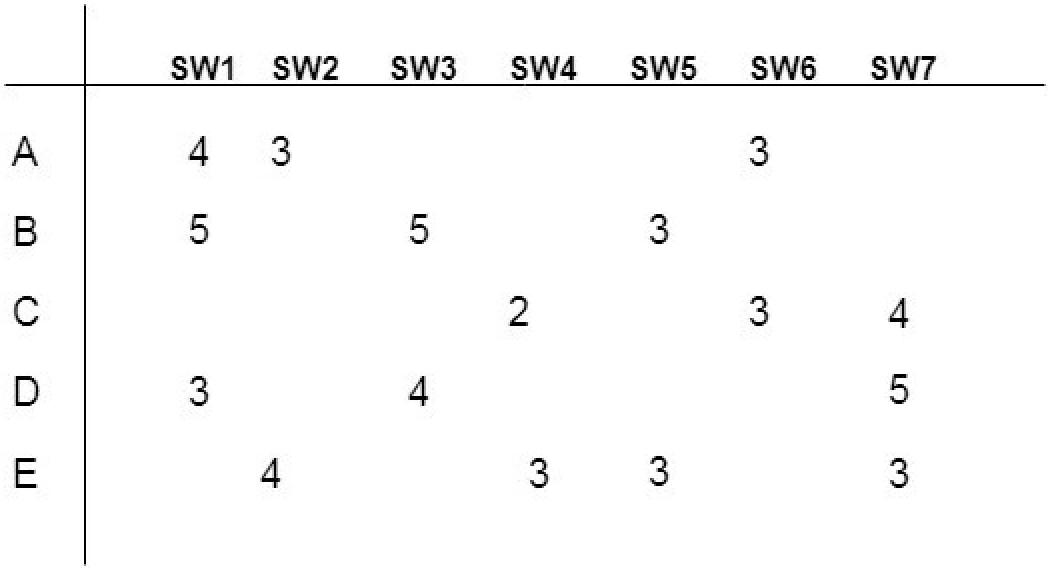
Based on the articles they had read in the past, news services have tried to identify articles of interest to readers.

The similarity could be based on the similarity of key words in papers or on articles read by people with a similar readability. The same guidelines regulate recommendation of a blog from millions of available blogs.

1.3 Utility Matrix

Utility Matrix is comprised of two users and item. For each pair of user items, a utility matrix is formed which represents what's known about the user's preference an item. Values come from an ordered set, for example, the 1–5 ratings which user has mentioned in reviews of an item. We consider matrix to be sparse, so most items are "unknown." If the rating for an item is unknown it signifies that user has not rated that item.

Fig 2 is an example of a utility matrix of ratings given by users to movies. Ratings are provided within the range 1-5 where 1 is lowest rating and 5 being the highest. Horizontal columns named SW1,SW2,SW3,SW4,SW5,SW6,SW7 are from STAR WARS movie series. The users are represented by alphabets A-E, All empty value represent that particular user has not the rated the movie so empty value will be considered as rating unknown



	SW1	SW2	SW3	SW4	SW5	SW6	SW7
A	4	3				3	
B	5		5		3		
C				2		3	4
D	3		4				5
E		4		3	3		3

Fig 2. Utility matrix

1.4 Populating the Utility Matrix

It is impossible to recommend items without a utility matrix. However, it is often difficult to acquire data to build utility matrix. There are two common methods to find user ratings. We can request users to give ratings to items. This is usually the way to obtain film ratings, and some online stores are trying to get ratings from their buyers.

Content sites like some media sites or Facebook also ask users to rate items. This approach is not good, as users generally do not want to provide answers, and it can be biased by the fact that is it coming from users who do not want to provide ratings.

Generally Utility Matrix are filled with binary values for example user bought a product can be assigned as 1 value where as I he didn’t buy an item it can be assigned 0 value. This method doesn’t provide with much valuable information. Hence many website use rating systems but still not all information can be translated into a rating system

CHAPTER 2: LITRATURE SURVEY

Recommender systems are one of the most important components of modern On-line stores. Studies have shown that useful recommendation are beneficial to both On-line stores as well as the consumer. Since past decade major research is being conducted in this stream. There are two major types of recommendation system in use which are content based **RS** and collaborative filtering RS both systems are prone to multiple types of attacks.

Content based filtering or cognitive filtering compares item with user profile to draw's result. Collaborative filtering belongs to more popular class of recommendation system and works by computing similarity between users and different items.

Collaborative systems are prone to major attacks such as sparsity, scalability and SAs. In this work collaborative systems will be in focus. These vulnerabilities cause a significant drop in accuracy as well performance of the system. Many hybrid algorithms are proposed which use multiple independent algorithm to form a solution.

For improving the accuracy and performance of recommender system techniques like forming a trust network [1] or forming clusters of users/item by using one of the standard clustering algorithms [3] [2].

A third type of RS: Context-aware RS is also becoming popular these systems treat user preferences of an item differently based on contextual factors such as time, purpose, location etc. for example " A customer may want to buy a shirt only on a certain occasion or specific reason" here that occasion and reasons are contextual factor. We can classify knowledge of RS based on contextual factors in three categories:

1. Fully Observable System: The contextual factors useful to the application as well as their structure and values are explicitly known when recommendations are made. Time, Shopping Purpose, Shopping Companion, etc. [4]

2. Partially Observable System: Only some information about contextual factors is available for example RS might know information about Time, Purpose and Location but their structure is not available [4].
3. Unobservable System: No information about the contextual factors is available to the system. RS can build its own predictive model to implicitly predict contextual factors using Markov model [4].

Out of these three RS models collaborative filtering RS will be of focus in this thesis. Following Table 1 presents a comparative study

S.NO	Author,Year	Dataset	Algorithms Used	Evaluation Metric
1	(P. Braak, N. Abdullah et al., 2009) [13]	Netflix	Clustering by genre	Time Comparision
2	(Gilda Moradi Dakhel, Mehregan Mahdavi, 2011) [14]	MovieLens	k-means clustering	Time Comparison
3	Rahul Kataria and O.P. Verma, 2016) [15]	MovieLens	Particle Swarm Optimization, K-means Clustering, Fuzzy c-means	MAE
4	(Anand Shanker Tewari, Asim Gopal Barman, 2016) [16]	Live data, trust Network of NIT Patna, India	Trust based Social Network, Association Rule Mining	Precision `

5	(Faris Alqadah, Chandan K. Reddy · Junling Hu, Hatim F. Alqadah, 2015) [17]	Paypal_big, Paypal_small, delic_bookmarks, lastfm_friends, lastfm	Bi-cluster Neighborhood Framework	Five-time Leave-One-Out cross validation (LOOCV), Hit Rate (HR), Average Reciprocal Hit-Rank (ARHR)
6	J. Ben Schafer, Joseph Konstan, John Riedl (2005) [5]	Amazon.com MovieFinder.com	Survery Paper	
7	(Dongting Sun , Zhigang Luo and Fuhai Zhang 2012)[6]	MovieLens	New Item Cold Start solution using K means Clustering	MAE
8	Soryoung Kim, Sang-Min Choi, Yo-Sub Han 2014)[7]	GroupLens,HetRec and IMDB	New Item Cold Start Solution using Item features	MAE
9	Cong Li and Li Ma 2009[8]	MovieLens	Cold Start Solution Using Tree Model	MAE
10	Reshma R, Ambikesh G and P Santhi Thilagam 2016 [9]	MovieLens	Nearest Neighbour	MAE
11	(Bushra Alhijawi, Yousef Kilani, 2016)[10]	MovieLens, Synthetic Data	Genetic Algorithm for Similarity Computation	MAE, Precision, Recall

12	(Pooyan Adibi, Behrouz Tork Ladani, 2013) [11]	MovieLens	Rating Timestamp, group membership, interest and similarity usage in CF technique	MAE

Table 1 Comparative Study on Cold Start Problem

CHAPTER 3: COLLABORATIVE FILTERING

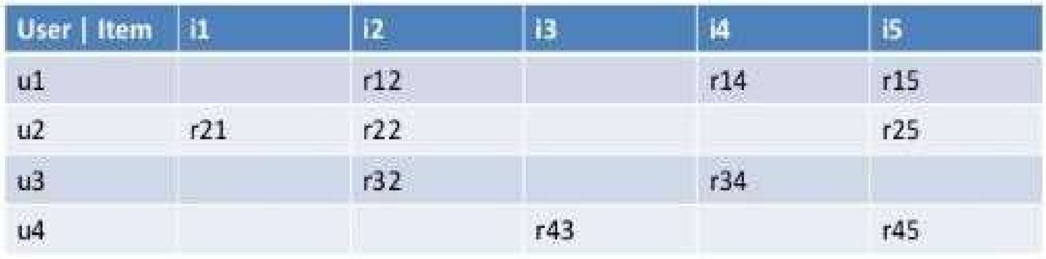
Collaborative filtering is one of the most important and popular algorithms that usually predict the rating of the particular user based on similarity between users. Algorithm works on the principal that if some user rated an item with similar rating they might rate other items with similar ratings as well. The similarity between users and/or items is obtained through common similarity measures such as cosine or adjusted cosine, Pearson correlation etc.

One of the benefits of considering like-minded users to make recommendations is that they overcome the "over-specialization" problem. Overly specialized means that the recommended items are always of the same type. Focusing on user ratings rather than content helps avoid such a problem.

The general collaborative filtering framework consists of the following three steps [18]:

1. Data Collection
2. Pre-processing
3. Collaborative

The data is collected from different sources and for its homogenization the pre-processing step is performed. After this step, we get a matrix known as the rating matrix or utility matrix with blank entries that the CF algorithm predicts. A sample rating matrix is shown in Fig. 3



User | Item	i1	i2	i3	i4	i5
u1		r12		r14	r15
u2	r21	r22			r25
u3		r32		r34	
u4			r43		r45

Figure 3 Rating or Utility Matrix

The steps of the framework are explained in detail below.

3.1Data collection

It is one of the most important activities of the entire process and the data mainly fall into the following 4 categories: -

1. Demographic data: It consists of the personal information of the users like name, telephone number, age etc. which helps businesses to construct users‟ profiles.

2. Production data: Here, classification of commodities is done based on their brands, functions etc. e.g., video tagging

3. User Behavior: e.g. playing duration of songs, book purchasing date etc.

4. User Rating: The actual ratings provided by the users

3.2 Preprocessing step

The data as collected above is in various formats due to the heterogeneity of the devices and networks; hence preprocessing is done to ensure a consistent format. There are 3 sub-steps in this step:-

3.2.1. ***Data Cleaning:***

Due to transmission errors or equipment failures, noisy data may be present in the system. Also, users may arbitrarily rate the items to save time. Hence here, we apply certain outlier detection algorithms to perform cleaning of the data.

3.2.2. ***Generation of Implicit Ratings:***

The rating matrix obtained is severely sparse leading to the data sparsity issue. We can use the user's behavior and apply machine learning techniques like neural networks to make a prediction model which can covert user behaviors into implicit ratings.

3.2.3. ***Data Integration:***

Explicit and implicit ratings are combined to form the rating matrix as shown in Fig. 3

3.3 Collaborative filtering

Finally, collaborative filtering is applied to make predictions regarding the user preferences. The detailed algorithm is explained in the next section.

3.4 Approaches to collaborative filtering

Collaborative Filtering Approaches are further classified as memory based, model based, and these techniques are explained in detail as below: -

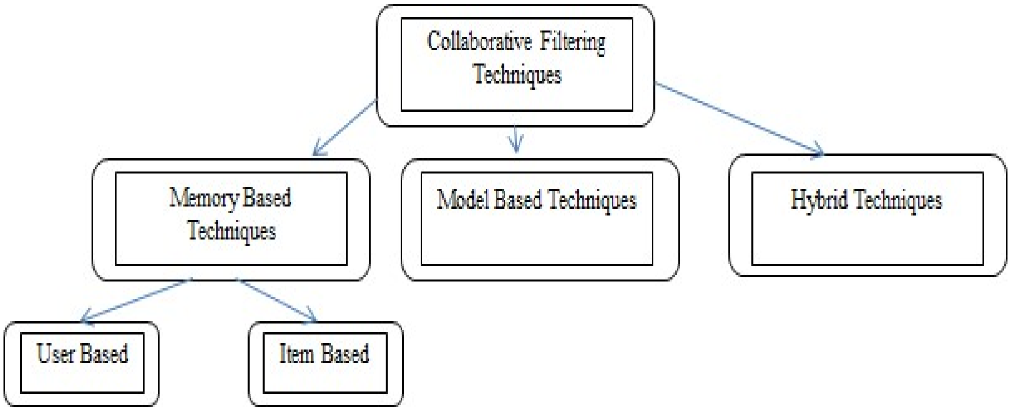


Figure 4 Classification of CF Techniques

3.4.1 Memory Based CF algorithms

These methods use the rating matrix directly for making predictions of the ratings. They are easy to implement but have large memory requirements for storing the complete rating matrix.

The general memory based model consists of the following steps [19]:-

1. Similarity Computation between users/items

2. Neighbor Selection

3. Prediction

4. Items Ranking

5. Selection of top k items

3.4.2 Similarity Metric

In order to find similarity values between users and/or items, the following similarity metrics are generally used.

1. Jaccard Similarity

   Consider two users u and v. Jaccard similarity between these users is defined as:-

   $$sim(u,v)^{Jaccard} = \frac{|I_u| \cap |I_v|}{|I_u| \cup |I_v|} \tag{1}$$

   Where I_u and I_v are the sets of items rated by user u and user v respectively.

2. Cosine Similarity:

   Here r_u and r_v respectively represent the rating vectors of users u and v. If a user has not rated a particular item, that rating is considered as zero.

   $$sim(u,v)^{COS} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \cdot \|\vec{r}_v\|} \tag{2}$$

3. Adjusted cosine similarity (ACOS)

Here, In order to remove the user bias i.e. the fact that different users give different ratings to an item even if they like it to same extent (different rating scales). P represents the set of all items. $r_{u,p}$ is the rating given by user u to item p. ru(bar) is the average rating of user u. Similar notations are followed for user v.

$$sim(u,v)^{ACOS} = \frac{\sum_{p\in P}\left(r_{u,p}-\bar{r}_u\right)\left(r_{v,p}-\bar{r}_v\right)}{\sqrt{\sum_{p\in P}\left(r_{u,p}-\bar{r}_u\right)^2}\cdot\sqrt{\sum_{p\in P}\left(r_{v,p}-\bar{r}_v\right)^2}} \qquad (3)$$

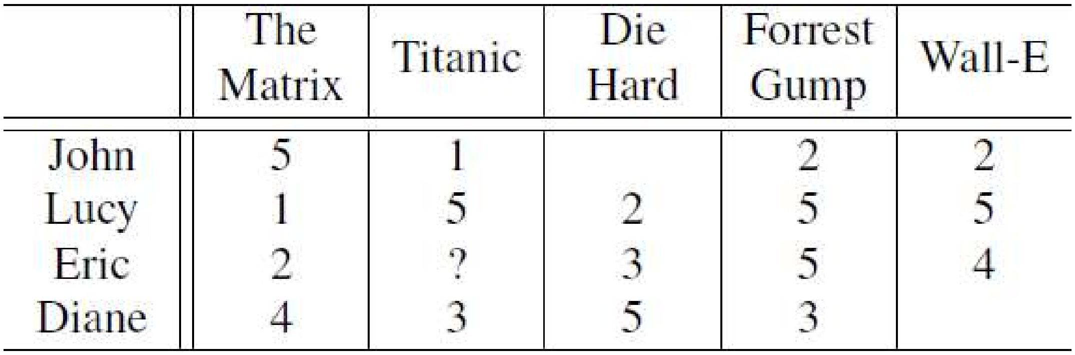
4. Pearson Correlation Coefficient (PCC)

Here, I is the set of items that are rated by both the users u and v. The difference between ACOS and PCC is that PCC uses only the co-rated items. PCC generally performs better than the other metrics.

The value of similarity metric lies between -1 and 1

$$sim(u,v)^{PCC} = \frac{\sum_{p\in I}\left(r_{u,p}-\bar{r}_u\right)\left(r_{v,p}-\bar{r}_v\right)}{\sqrt{\sum_{p\in I}\left(r_{u,p}-\bar{r}_u\right)^2}\cdot\sqrt{\sum_{p\in I}\left(r_{v,p}-\bar{r}_v\right)^2}} \qquad (4)$$

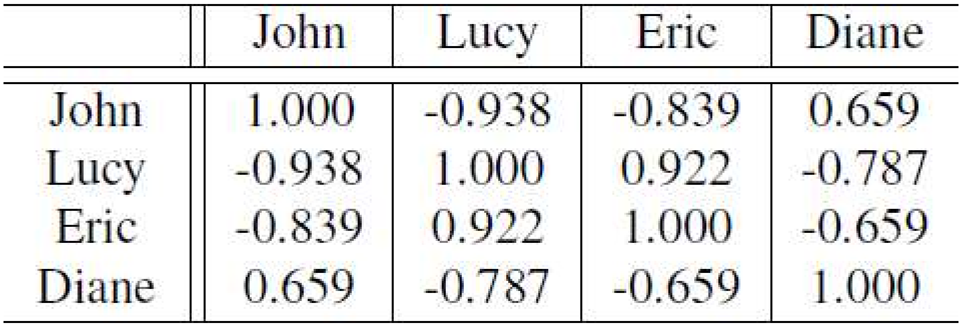
For example, consider the following rating matrix:



	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
John	5	1		2	2
Lucy	1	5	2	5	5
Eric	2	?	3	5	4
Diane	4	3	5	3	

Figure. 5 Sample rating Matrix

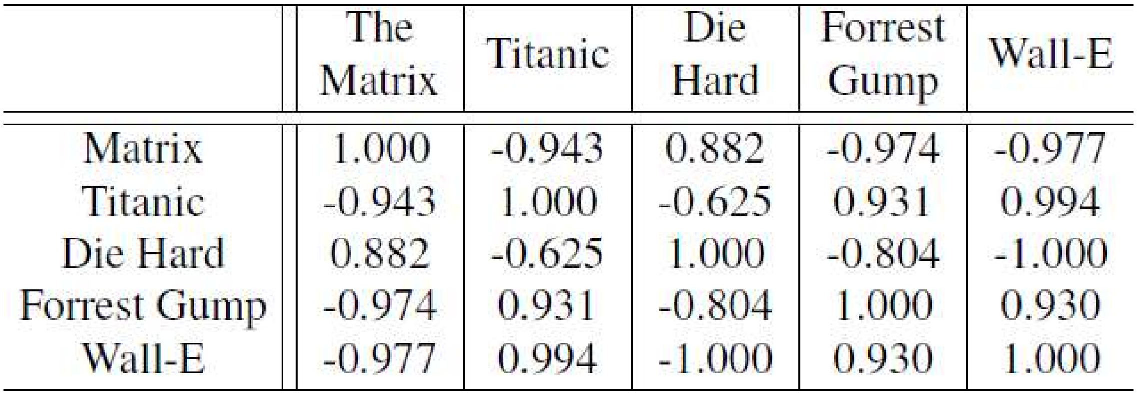
Where rows of the matrix represent users and columns are representing movies. Now, using Pearson correlation coefficient, the similarity values between each pair of users is given by the following matrix:-



	John	Lucy	Eric	Diane
John	1.000	-0.938	-0.839	0.659
Lucy	-0.938	1.000	0.922	-0.787
Eric	-0.839	0.922	1.000	-0.659
Diane	0.659	-0.787	-0.659	1.000

Figure 6 User Based Pearson Correlation

Similarly, we can use these methods for item similarity computation. For example, the Pearson correlation values for the pair of items in the above rating matrix are computes as follow q



	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
Matrix	1.000	-0.943	0.882	-0.974	-0.977
Titanic	-0.943	1.000	-0.625	0.931	0.994
Die Hard	0.882	-0.625	1.000	-0.804	-1.000
Forrest Gump	-0.974	0.931	-0.804	1.000	0.930
Wall-E	-0.977	0.994	-1.000	0.930	1.000

Figure 7 Item Based Pearson Correlation

Memory based CF Algorithms are further classified into 2 categories based on whether we calculate similarity between users or items: -

3.4.3 User based approach

This algorithm works in two phases: -

1. User neighbourhood formation phase: In this phase, we calculate the similarity between the target user u and other users using common similarity metrics as described above and select the top k neighbours who have rated the target item i. We denote these neighbours by Ni(u)

2. Recommendation phase: The predicted value of concerned rating is computed as follows: -

$$\hat{r}_{ui} = \frac{\sum\limits_{v\in\mathcal{N}_i(u)} w_{uv}\, r_{vi}}{\sum\limits_{v\in\mathcal{N}_i(u)} |w_{uv}|}. \qquad (5)$$

where w_{uv} represent the similarity value between the target user and the nearest neighbour under consideration.

In the denominator, modulus is used so that the predicted ratings are within the legitimate range of rating scale. Rating normalization with mean-clustering is performed to remove the user-bias. Therefore, the predicted rating is now computed as:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum\limits_{v \in \mathcal{N}_i(u)} w_{uv} \left(r_{vi} - \bar{r}_v\right)}{\sum\limits_{v \in \mathcal{N}_i(u)} |w_{uv}|}. \tag{6}$$

The user-based approach is pictorially depicted in the Fig. 8

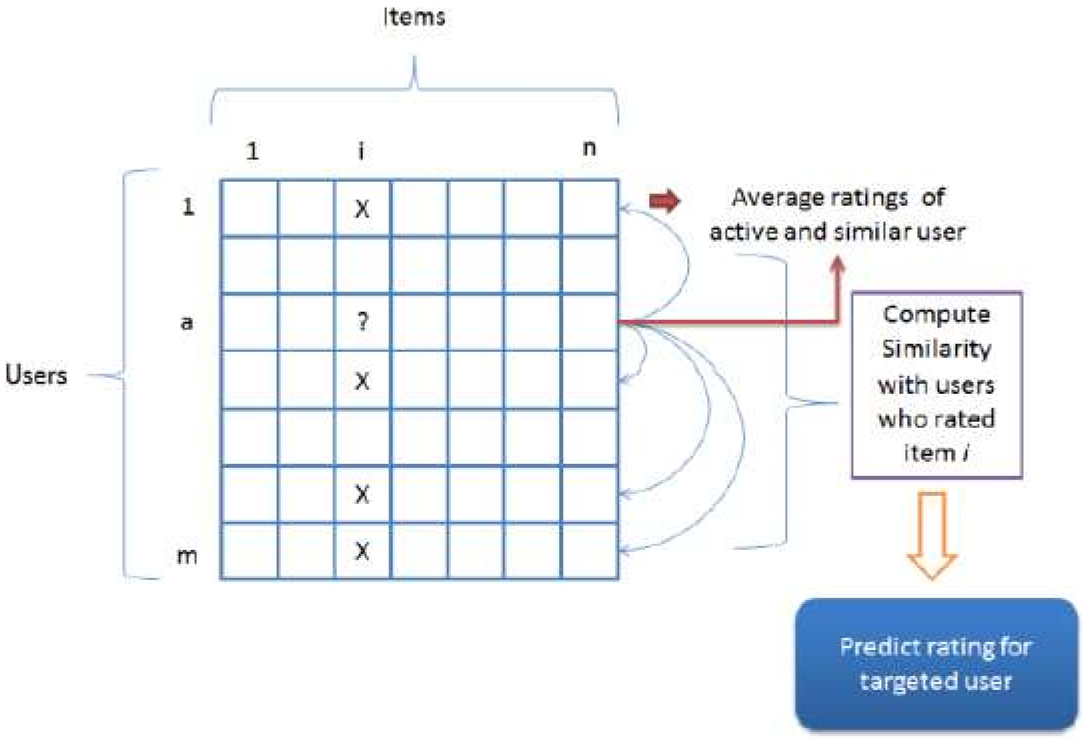

Items
1
i
n
Users
1
a
m
X
?
X
X
X
Average ratings of active and similar user
Compute Similarity with users who rated item i
Predict rating for targeted user

Figure 8 User Based CF Process

3.4.4 Item based approach

Similar to the user-based method, item based method also works in two phases: -

1. Item neighborhood formation phase: In this phase, we calculate the similarity between the target item i and other items using common similarity metrics as described above and select the top k neighbors that are rated by the target user u. We denote these neighbors by Nu(i)

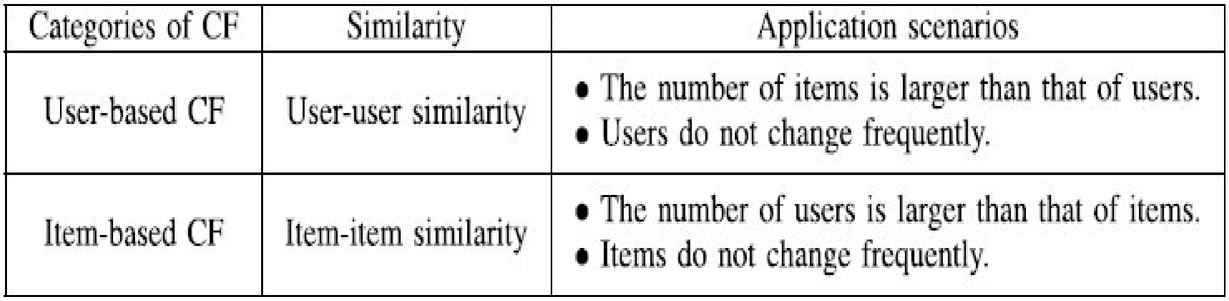
2. Recommendation phase: The predicted value of concerned rating is computed as follows: -

$$\hat{r}_{ui} = \frac{\sum\limits_{j \in \mathcal{N}_u(i)} w_{ij} r_{uj}}{\sum\limits_{j \in \mathcal{N}_u(i)} |w_{ij}|}. \qquad (7)$$

Where w_{uv} represent the similarity value between the target item and the nearest neighbour under consideration. Rating normalization with mean-clustering is performed to remove the user-bias. Therefore, the p

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum\limits_{j \in \mathcal{N}_u(i)} w_{ij} (r_{uj} - \bar{r}_j)}{\sum\limits_{j \in \mathcal{N}_u(i)} |w_{ij}|}. \qquad (8)$$

Table 2 shows a comparison between user based and item-based approaches of the memory based CF Algorithms



Categories of CF	Similarity	Application scenarios
User-based CF	User-user similarity	• The number of items is larger than that of users. • Users do not change frequently.
Item-based CF	Item-item similarity	• The number of users is larger than that of items. • Items do not change frequently.

Table 2 Comparative Study between User Based and Item Based Memory CF Model

CHAPTER 4: CHALLENGES IN RECOMMENDATION SYSTEMS

Recommendation system are not perfect and hence are prone to many types of attacks and challenges within themselves. Many of the major issues in CF systems are mentioned below

1. **Cold Start problem** – One of the major challenges in RS is to suggest item to user. To recommend item to user the RS needs to look into user's purchase history but in case of a new user no such history is present hence suggestions are inaccurate. In CF based system similarity for new users cannot be established. Some basic solution to such problem involves forcing new user to rate few items before he can start purchasing.

2. **Data sparsity Problem -** CF's principle is to aggregate like-minded user's ratings. However, because of user's absence of knowledge or incentive to rate items the reported user rating matrix is generally very empty.
   This issue will prohibit effective recommendations from being made by the CF because the preference of users is difficult to extract.

3. **SA -** In a SA user are inserted into system to provide fake or biased ratings to item, these fake ratings can make even bad products to be highly recommended by system.

4. **Changing Dataset -** The database is constantly growing leading to the problem of always changing data set.

5. **Gray sheep problem -** Some users have very different interest when it comes to item ratings and since CF works by comparing similarity of users it becomes difficult to recommend item for a Grey sheep user

6. **Long Tail Issue -** Recommendation system do not provide users with many options since all recommendations will be based user's purchase history this causes recommendations for only popular products leading to diversity in suggestions problem

SNO	CHALLENGE/ATTACK	CAUSE(S)
1.	Cold Start Problem	New users and/or items
2.	Data Sparsity Problem	Large number of missing or blank entries in the rating matrix
3.	SAs	Wrong or fake ratings provided by some users
4.	Changing Dataset	Dynamic nature of the rating matrix with constant inclusion of new users and/or items
5.	Gray Sheep Problem	Unusual interests of some users for items that is different from other users
6.	Long Tail Issue	Lack of diversity in recommendation

Table 3 Challenges in recommendation systems

CHAPTER 5: COLD START PROBLEM

It is hard to recommend new items to user as no information about his previous purchase history is available. There are multiple methods to solve this challenge but all revolve around taking some initial preferences from the new user, without these initial information useful suggestions cannot be made. This problem is commonly known as the cold start problem in recommendation system. When we talk about cold start problem regarding CF system, all suggestions are based on similarity among users/items. One of the Technique to solve cold start problem in CF RS is “ask-to-rate”. When a new user registers into the system he is asked rate few items or give a list of his preferences for items. In this way system can gain some initial information to recommend correct items to him. Another solution is, initially a new user is provided with inaccurate suggestion than according to his ratings on those suggestion his new preferences will be determined. In this way accuracy for recommendations will improve gradually.

In “ask-to-rate” method it becomes very essential to select the initial items which should to be asked to rate by the user to rate. The following mentioned methods can be used to select the initial set of items.

5.1 *Popularity Strategy:*

In this method most popular items from the system are presented to the user. Popularity of an item can be derived as the number of users who have rated the item. The item which is rated by max no. of user can be considered as most popular. It can very well be the case where even the most popular item with the most negative ratings. Popularity of an item can be defined as

$$\text{Popularity}(\tau) = |T| \qquad (9)$$

In this equation |T| refers to the number of users who have rated that item T. This strategy is easy to implement but does not give much useful information as most of user have rated the popular item, so we cannot draw a specific user profile based on most popular item especially in CF systems.

5.2 **Random Strategy:**

In this method items are selected in random manner and presented to the user for rating. This approach is not effective as the user may not have any idea about the items he is required to rate. It is one of the basic approaches for cold start problem.

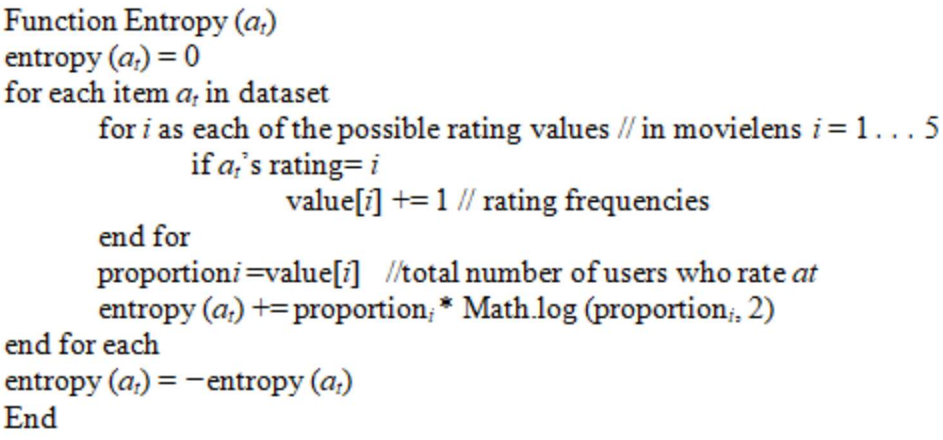
5.3 ***Pure entropy:***

Entropy is the measure of randomness in data. In this method we present users with items having mixed ratings so that system is easily able to draw user profile. The items presented are arranged in a descending order of entropy and then presented to the user in non-increasing order

Entropy is defined as:

$$\text{Entropy}(\mathrm{a}_t) = \sum_{i=1}^{5} -p_i * log p_i \qquad (11)$$

Where p_i is rating for an item. Below is mentioned pseudo code for entropy calculation



```
Function Entropy ($a_t$)
entropy ($a_t$) = 0
for each item $a_t$ in dataset
        for $i$ as each of the possible rating values // in movielens $i$ = 1 . . . 5
                if $a_t$'s rating= $i$
                        value[$i$] += 1 // rating frequencies
        end for
        proportion$i$ =value[$i$]    //total number of users who rate $at$
        entropy ($a_t$) += proportion$_i$ * Math.log (proportion$_i$, 2)
end for each
entropy ($a_t$) = −entropy ($a_t$)
End
```

Figure 9 Function Entropy

5.4HELF

Harmonic mean of entropy and Logarithm frequency. In this strategy harmonic mean of Entropy and Frequency is calculated. Normalization of Entropy and Frequency is also done so that no one factor and dominate the other

HELF value of ai is calculated as: -

$$HELF_{a_i} = \frac{2 * LF'_{a_i} * H'(a_i)}{LF'_{a_i} + H'(a_i)} \quad (12)$$

Where, LF'ai is logarithm of the frequency or popularity of ai and is normalized by a factor as well (lg(|U|)): lg(|ai|)/lg(|U|), Similarly, H'(ai) is the entropy of ai and normalized by a factor of lg (5): H(ai)/lg(5).

5.5Balanced Strategy:

In this method we use both popularity and entropy method in combination. Popularity ensures that ratings from users are high and entropy ensures that there is still randomness in items presented for survey

5.6 Item Based Cold Start Problem:

The new items added to the system are generally excluded while making recommendation and are not presented to new users for initial preference. The reason they are excluded is since these items are new and no user has any preference for it. To solve this problem a set of users can be selected and persuaded to rate these new items.

5.6.1 Market-based approach

We consider at a time t there is a set of new items $I_{t.,}$ We also associate a cost with getting a user's to rate an item. We want to maximize the reach of an item while minimizing the overall cost selecting the users this is known as the market based approach [15]. We consider at a time t there is a set of new items $I_{t.}$,also an overall budget is dedicated for all new items. Users are selected on the basis of budget and influence of a user with respect to the item.

Here an earn-per-rating (EPR) list is constructed for each user from which user can choose and provide a rating for a payment. Every new item is placed in a user's EPR list using an item rank. The rank l^t_{ak} for an item i_k in a user (u_a) EPR list. L^t_a is function of the bid price for an item b^t_k and the rate of uptake of the item ru^t_k at a time t.

Rate of uptake is :-

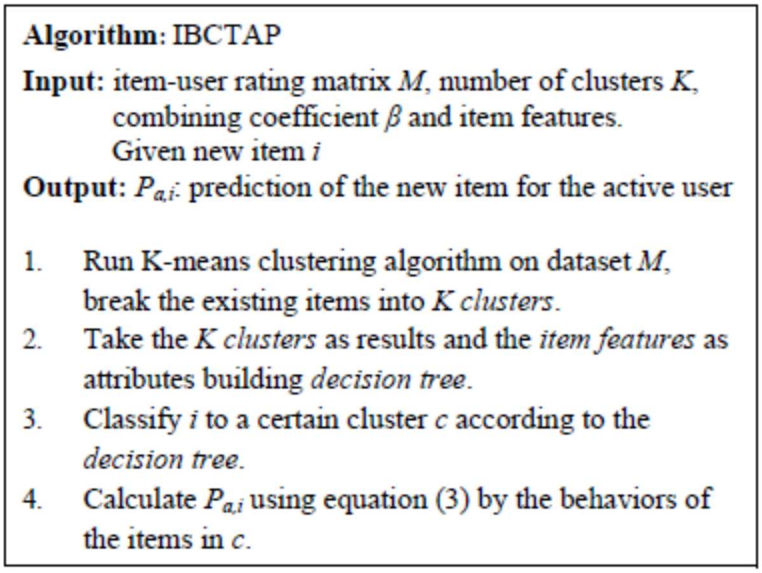
$$ru_k^t = \begin{cases} 1 & \text{if } t = t_k \\ \dfrac{\sum\limits_{u_a \in U_k^t} pu(l_{ak}^t)}{\sum\limits_{u_a \in \hat{U}_k^t} pu(l_{ak}^t)} & \text{otherwise} \end{cases} \tag{13}$$

This rate of uptake is average appeal of an item while selection by a user. This appeal of an item gives us the new item which has maximum influence. Thus, new items with high rate of uptake can be used for initial screening of a new user.

Generally, there are various features associated with items like movie can have features like actors, producers etc. One of the common approaches to select item in CF system is to find the similar items for each items. One of the important algorithm is IBCTAP algorithm [6].

In this algorithm a cluster is formed of items using user-item matrix. The item are partitioned into a group based on similar user liking and we combine this information to build a decision tree to form an association between new and old items. The algorithm IBCTAP has 4 main steps

a) Item clustering
b) Decision tree building
c) New item classifying
d) Ratings predictions



```
Algorithm: IBCTAP
Input: item-user rating matrix $M$, number of clusters $K$,
       combining coefficient $\beta$ and item features.
       Given new item $i$
Output: $P_{a,i}$: prediction of the new item for the active user

1. Run K-means clustering algorithm on dataset $M$,
   break the existing items into $K$ clusters.
2. Take the $K$ clusters as results and the item features as
   attributes building decision tree.
3. Classify $i$ to a certain cluster $c$ according to the
   decision tree.
4. Calculate $P_{a,i}$ using equation (3) by the behaviors of
   the items in $c$.
```

Figure 10 IBCTAP Algorithm

a) **Item Clustering**: - K means algorithm is used to form cluster of items which are similar it is often the case when user having a preference of item in one cluster will also have a preference of item in same cluster. Initially k centroid are initialized using K -means algorithm and uses a similarity algorithm to form clusters. Pearson correlation coefficient is used to find similarity between items. Generally, items will have high correlation if user liked both items. Using this assumption all items which are liked a by users with same taste will be same cluster.

b) **Decision Tree building:** - Standard Decision Tree building algorithms are used like ID3 and C4.5 which work on the principal of information gain. All items have set of features as described above, for database like Movie Lens 100k each movie item can have features like director, producer, actors etc. A decision tree is built for item based on these attributes the algorithm decides which feature needs to be selected for splitting up the node. The splitting on an attribute is performed only in true and false fashion. Entropy for each item is calculated followed by information gain, if information gain increased the corresponding feature then it is selected for splitting [6]. After each such splitting system proceeds further to see if nodes can be divided or not. Below is a simple example to show a decision tree for such system, where squares show available decisions and circle represents the cluster number.

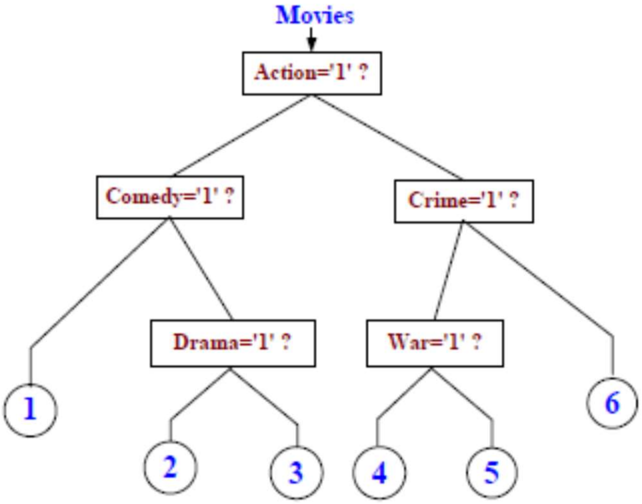


Figure 11 Decision Tree for New Item Recommendation

c) **New Item Classifying: -** Whenever a new item enters the system which no one has rated yet. IBCTAP algorithm classifies it into a cluster using the decision tree and decision algorithm. For example, if an item i arrives in the system it goes to the root of the decision tree. If the item i is a movie of comedy genre it will go to cluster 1. Therefore cluster 1 will contain all movies of comedy genre. In reality this decision tree is not so simple and contain a lot of nodes and clusters as movies can have a lot of item features and separate path for decisions. A worst situation can be when an item i goes to more than one clusters where it becomes difficult to group the new item with already present items.

d) **Rating items:** Clustering of items with similar features ensures that if a user like one item in that a cluster he will also prefer other items in the same cluster since they all have similar features. Hence the item is recommended to a user by below equation

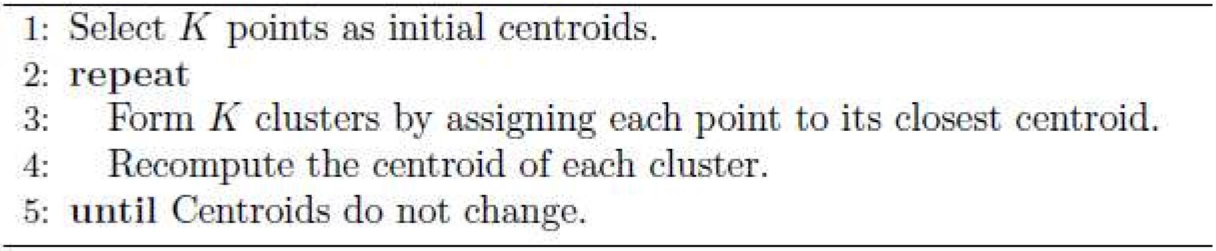
$$p_{a,i} = \beta \overline{r_i}^{c} + (1-\beta) r^{cf} \quad (14)$$

Where r_i is the mean rating of the active user "a" in that cluster for whom recommendation needs to be made and r^{cf} is the pearson correlation coefficient among ratings. Where β is the constant for balancing extremely cold start situation. The Pearson correlation coefficient can find similarity between item i and other items for an active user "a".

$$p_{a,i} = \frac{\sum_{j \in I} r_{a,j} \cdot sim(i,j)}{\sum_{j \in I} sim(i,j)} \quad (15)$$

5.7 K-Means Clustering

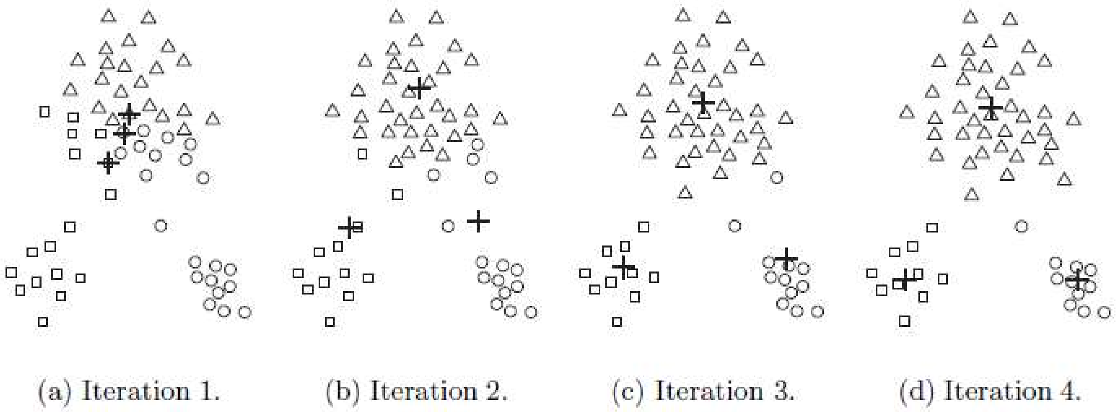
K means is a simple algorithm we start with initial set of K centroid points, This K is initial number of centroids provided by the user. Each new element is then assigned a centroid point. Centroid of each cluster is updated to after taking new element into account. This step is repeated until centroid stops changing. Below is the algorithm for K means clustering.



```
Select $K$ points as initial centroids.
repeat
    Form $K$ clusters by assigning each point to its closest centroid.
    Recompute the centroid of each cluster.
until Centroids do not change.
```

Figure.12 K-means Algorithm

These steps are performed in iterations until centroids stop changing. Below is visual representation of working of 4 iterations in K-means algorithm.



(a) Iteration 1.
(b) Iteration 2.
(c) Iteration 3.
(d) Iteration 4.

Figure 13 K-Means Algorithm Iteration

CHAPTER 6: PROPOSED SOLUTION

In the proposed framework a genre-based clustering is used where all users will be clustered based on their preference over the genre. K-means algorithm is used to perform this clustering. Clustering can be done on singular genre or group of genres, here we are forming clusters on a single genre. Dataset used for testing proposed framework is Movie-Lens 1million dataset. It is comprised of movie and movie ratings given by users. In approach used in [6] focuses on alleviating new item cold start problem using K-means on item similarity followed by a decision tree for every item. Algorithm in [6] does not work very well if new user also enters the system. Algorithm fails when items fall into more than one clusters. Proposed Framework works for both new user in system as well as new Item in the system.

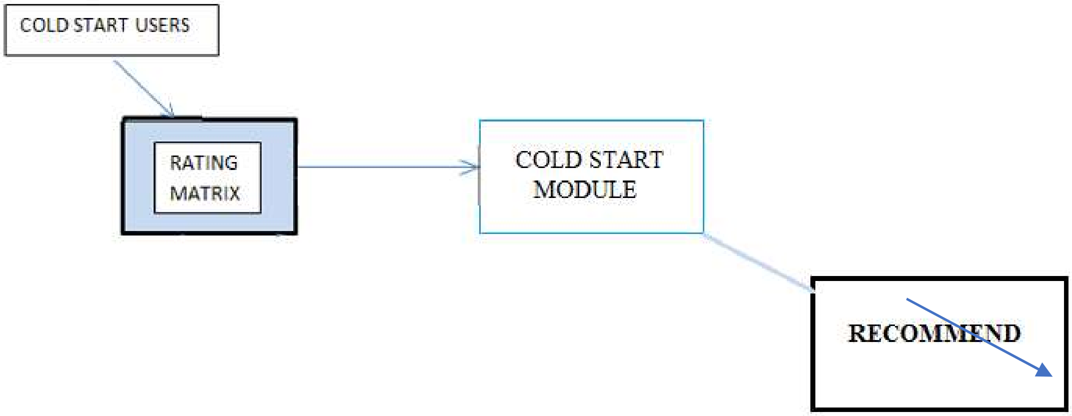


Figure 14 Proposed Solution

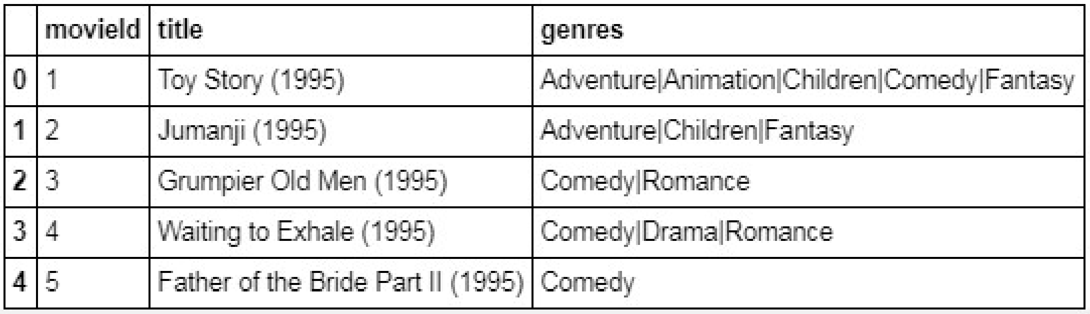
We start by taking initial user-item matrix and perform K- means upon it based on Item features which is Movie genre for this dataset after that we take new user/item preference into account and merge the clusters to obtain similar users. The list of similar users is used for recommending

1. **New User:** If algorithm is applied for a new user the resulting list of user Id corresponds to all user who have similar preference to new user. Thus, these users

rated items can be recommended to the new user.

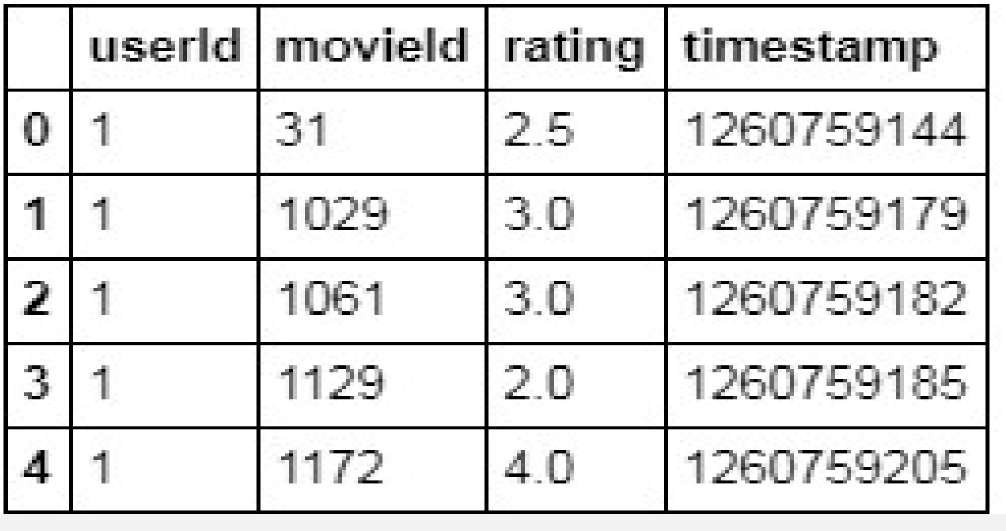
2. **New Item:** If algorithm is applied for a new item the resulting list of user id based on item features corresponds to all user to whom this item can be recommended.

Below are mentioned dataset samples from Movie-Lens to understand structure of data



	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Table 4 Movie Dataset

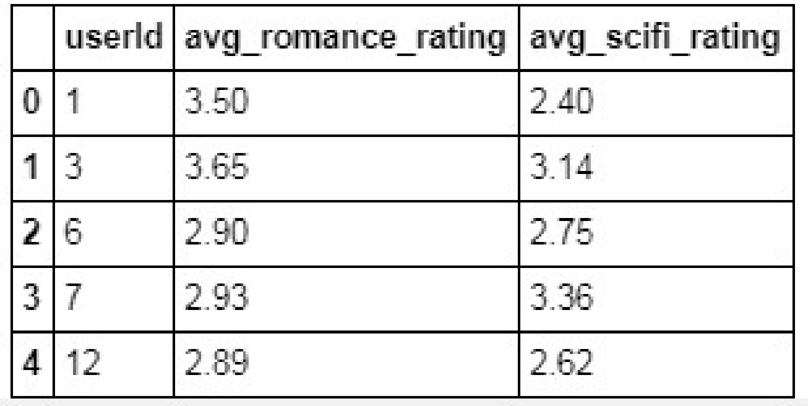


	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

Table 5 User Ratings Dataset

Example of Feature/Genre based Clustering on two genres

Let’s consider two genres Romance and Sci-fi to perform clustering of users. First, we will need to compute the average rating of each user



	userId	avg_romance_rating	avg_scifi_rating
0	1	3.50	2.40
1	3	3.65	3.14
2	6	2.90	2.75
3	7	2.93	3.36
4	12	2.89	2.62

Table 6 Average user Ratings

For more appropriate visual representation below graph represents Table 6

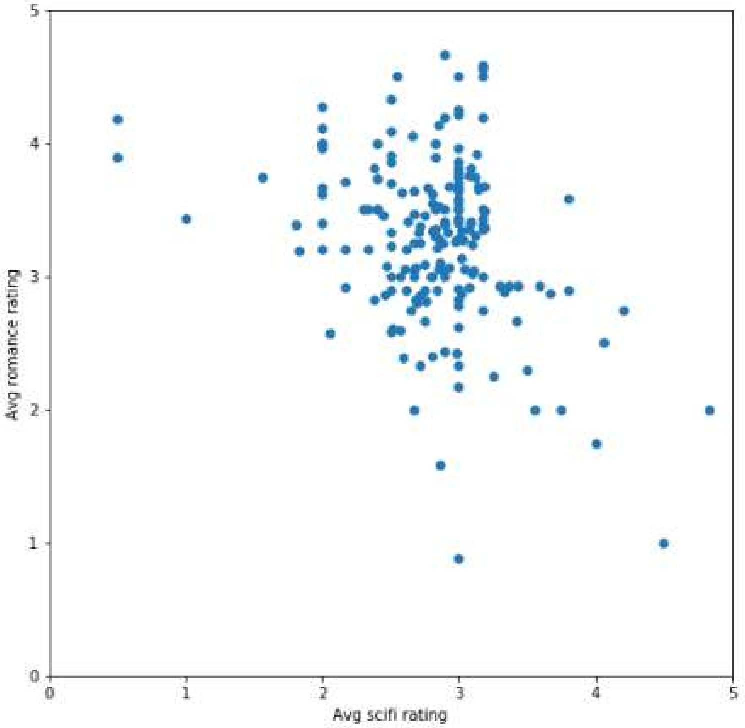


Figure 15 Visual Representation Average Ratings

The blue dots represent each user whereas the x and y axis represent average ratings in Sc-fi and Romance Genre. We can break this graph down into two categories using K-means

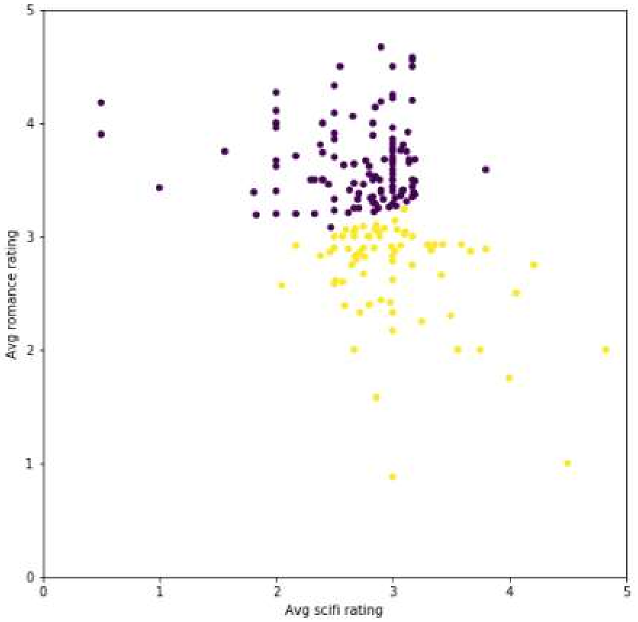


Figure 16 K-means on average ratings

We can see that the groups are mostly based on how each person rated romance movies. Similarly, we can break down the input into three clusters by setting number of initial clusters in k-means to 3

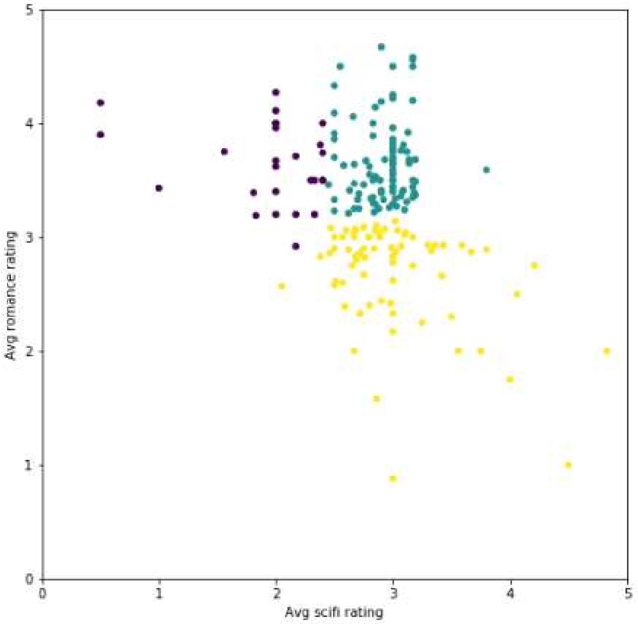


Figure 17 K-means with 3 clusters

Now the average sci-fi rating is starting to come into play. The groups are:

- people who like romance but not sci-fi (denoted by purple cluster)
- people who like sci-fi but not romance (denoted by Yellow cluster)
- people who like both sci-fi and romance (denoted by Cyan cluster)

6.1 *Proposed algorithm*

INPUT: USER-ITEM MATRIX

OUTPUT: USER LIST

STEPS:

1.) Find all Different features for an item
2.) Apply K-means Clustering on every feature
3.) Take Preference of New Item or New User
4.) Merge Clusters according to New Item/User preference
5.) Obtained User list signify Similar users in case of New user.
6.) Obtained User list signify users to whom new item can be recommended
7.) Stop

When we apply K-means on single feature it groups all user who like that feature together and user who have rating less than average in another group. Merging of cluster is done via computing Intersection function.

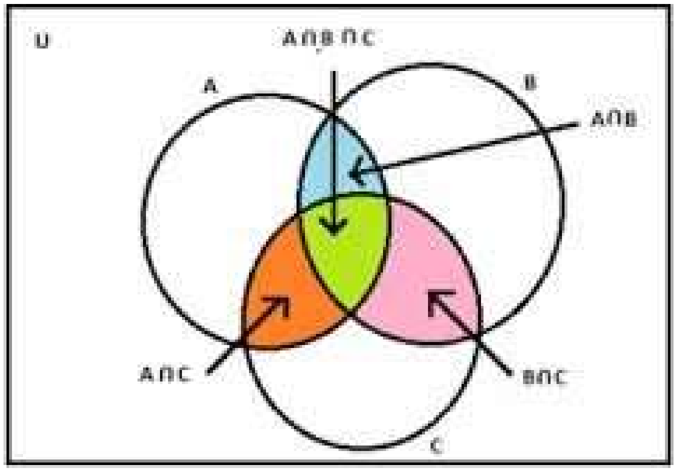
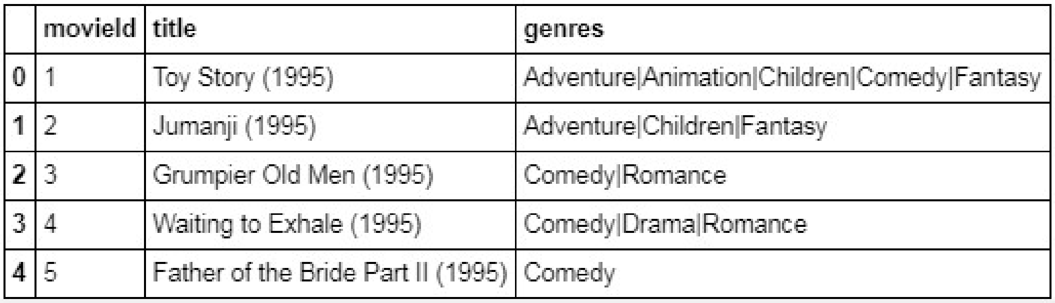


Figure 18 Intersection among multiple Clusters

CHAPTER7: IMPLEMENTATION AND RESULTS

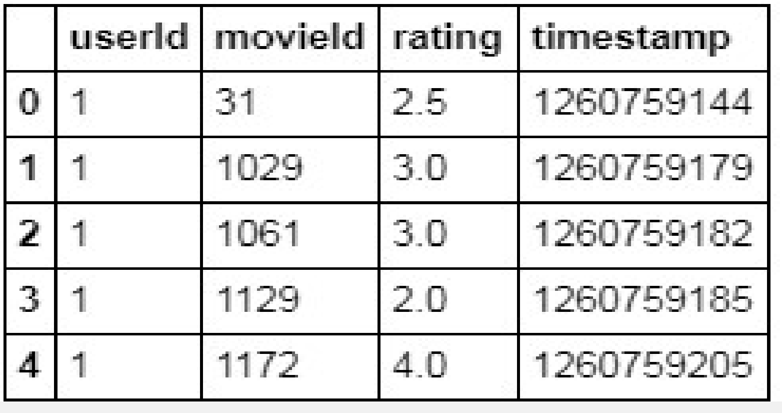
7.1 *Implementation details*

1. Python 3.7 is used for implementation of above framework with SCIKIT learn library to implement K-means algorithm.
2. Dataset used is Movie-lens 1million containing 650 user and 1900 movies which contains 1 Million edges in from of user ratings where each is movie rated from rating 1-5. Following is representation of data in dataset.



	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Table 7 Movie Dataset



	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

Table 8 User Ratings Dataset

7.2 Results

While clustering over an item feature below graph gives us a visual representation of how users are separated based on their ratings on a feature. For movie genre set to romance following is the resultant from clustering.

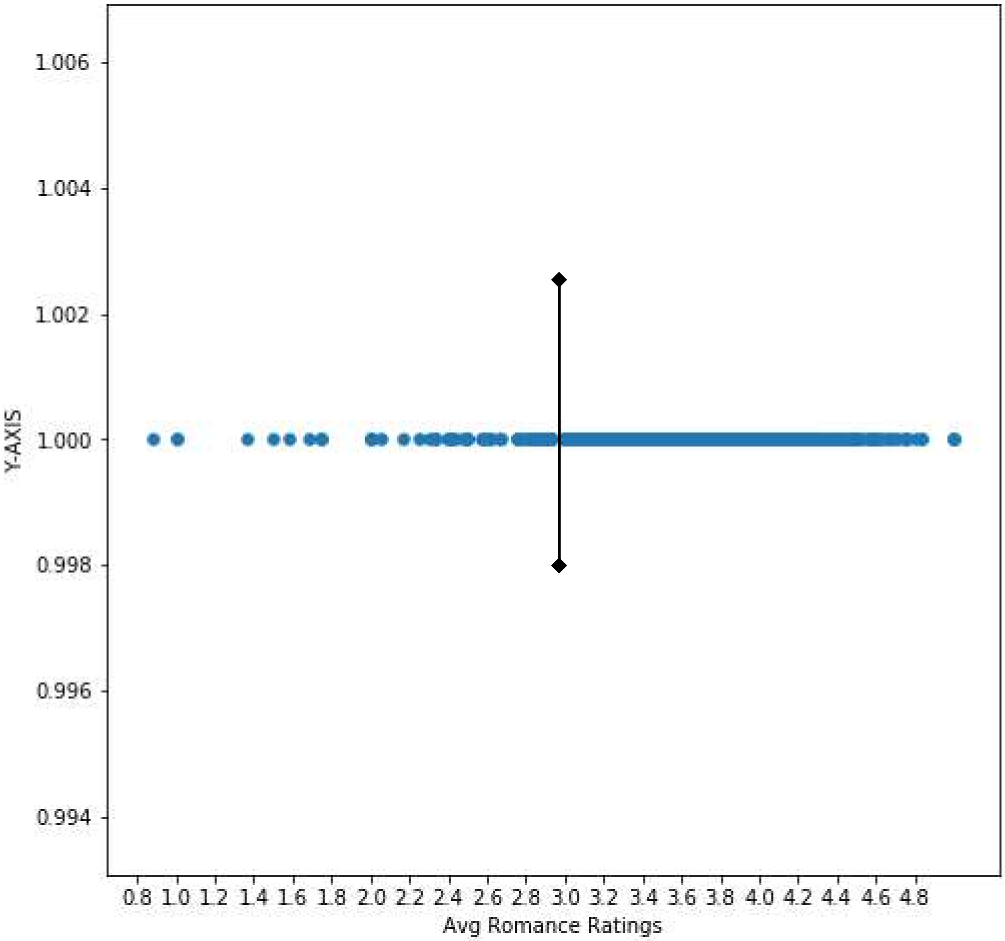


Figure 19 Plot for Avg romance user ratings

For average user ratings over romance genre it can be seen that after average rating of 3 is a crucial point where user below average rating 3 can be considered as user who do not like this genre and users above 3 are the user who prefer this genre. Similarly, for other genres

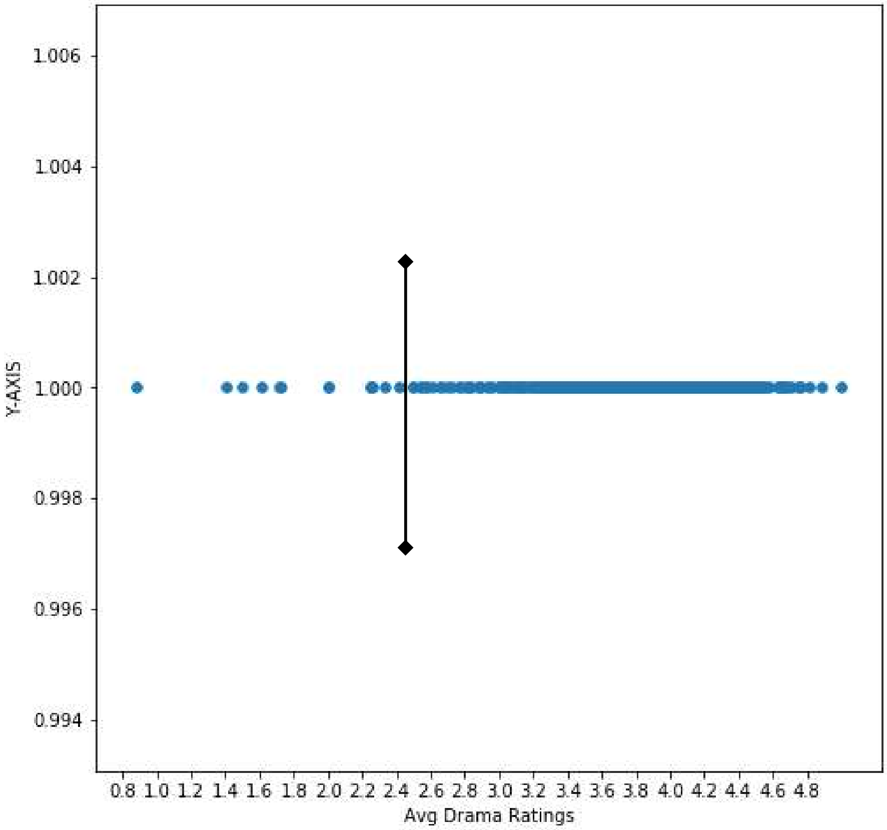


Figure 20. Plot for Avg Drama user ratings

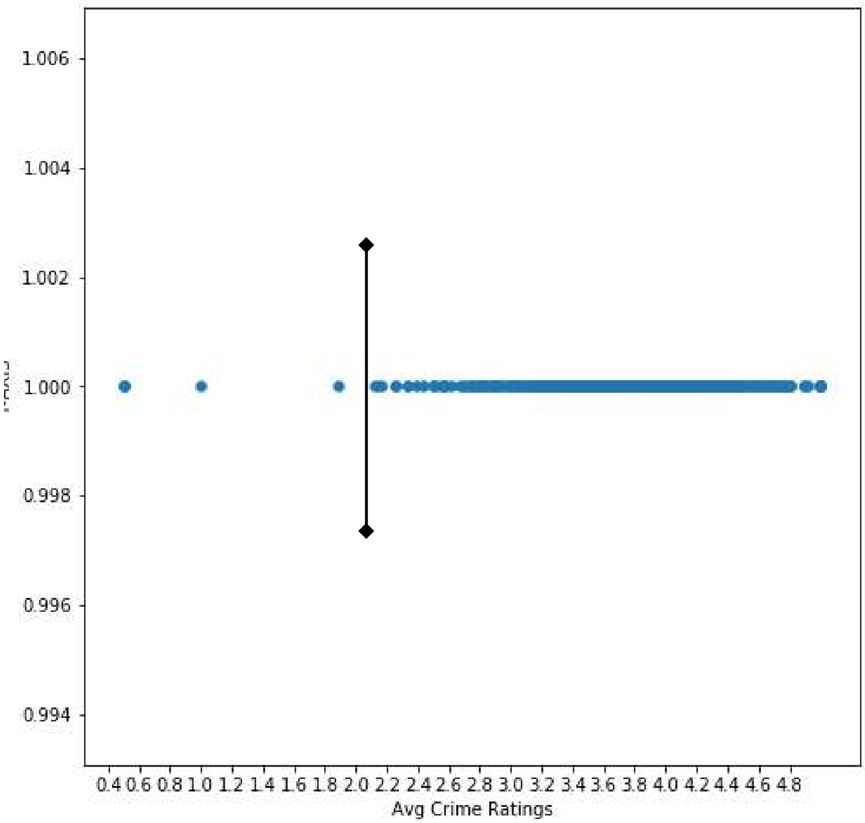


Figure 21. Plot for Avg Crime user ratings

Both Drama and Crime genre similarly show separation points at 2.4 and 2.0 average ratings. As a result, clustering on item features leads to solution for both new user and new item problem.

CHAPTER 8: CONCLUSION

Cold start problem is one of the most common challenges in CF system, most of the traditional approaches relied upon conducting initial survey from new user so that similarity can be drawn. Such approach relies majorly upon the quality of the survey conducted, if item selected for survey are not optimal it will further lead to incorrect recommendation from the system. Several approaches formed in item selection for such survey but led solution to new user problem. Another part of cold start problem is new item in the system. In recent study and research conducted mentioned in Table 1 directs toward the use of item features to resolve cold start problem along with clustering or decision tree. In this thesis, approach of item feature clustering led to solution for cold start users and items in a single algorithm. The merging of multiple clusters into one led to recommendation even over more than one item features. In conclusion K-means algorithm over item feature clustering results in cold start problem solution in collaborative recommendation system

REFERENCES

[1] Du, Y., Du, X. and Huang, L., 2016. Improve the collaborative filtering recommender system performance by trust network construction. Chinese Journal of Electronics, 25(3), pp.418-423.

[2] Leskovec, J., Rajaraman, A. and Ullman, J.D., 2014. Mining of massive datasets. Cambridge university press.

[3] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on(Vol. 3, pp. 147-150). IEEE.

[4] Mining of Massive Datasets, Ananad Rajarman, Jeffrey David Ullman, Cambridge University Press New York, NY, USA 2011.

[5] J. Ben Schafer, Joseph Konstan, John Riedl 2005, Recommender Systems in E-Commerce

[6] A Novel Approach for Collaborative Filtering to Alleviate the New Item Cold-Start Problem, Dongting Sun and Zhigang Luo

[7] Soryoung Kim, Sang-Min Choi, Yo-Sub Han 2014, Analyzing Item Features for Cold-Start Problems in Recommendation Systems, 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

[8] Cong Li and Li Ma 2009, Collaborative Filtering Cold-Start Recommendation Based on Dynamic Browsing Tree Model in E-commerce, 2009 International Conference on Web Information Systems and Mining

[9] Reshma R, Ambikesh G and P Santhi Thilagam 2016, Alleviating Data Sparsity and Cold Start in Recommender Systems using Social Behaviour, 2016 FIFTH INTERNATIONAL CONFERENCE ON RECENT TRENDS IN INFORMATION TECHNOLOGY

[10] Alhijawi, B. and Kilani, Y., 2016, June. Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. In Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on (pp. 1-6). IEEE.

[11] Adibi, P. and Ladani, B.T., 2013, May. A collaborative filtering recommender system based on user's time pattern activity. In Information and Knowledge Technology (IKT), 2013 5th Conference on (pp. 252-257). IEEE.

[12] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on(Vol. 3, pp. 147-150). IEEE

[14] Dakhel, G.M. and Mahdavi, M., 2011, December. A new collaborative filtering algorithm using K-means clustering and neighbors' voting. In Hybrid Intelligent Systems (HIS), 2011 11th International Conference on (pp. 179-184). IEEE

[15] Katarya, R. and Verma, O.P., 2016. A collaborative recommender system enhanced with particle swarm optimization technique. Multimedia Tools and Applications, 75(15), pp.9225- 9239.

[16] Tewari, A.S. and Barman, A.G., 2016, December. Collaborative book recommendation system using trust based social network and association rule mining. In Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on (pp. 85-88). IEEE.

[17] Alqadah, F., Reddy, C.K., Hu, J. and Alqadah, H.F., 2015. Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems, 44(2), pp.475-491.

[18] Yang, Z., Wu, B., Zheng, K., Wang, X. and Lei, L., 2016. A survey of collaborative filtering-based recommender systems for mobile Internet applications. IEEE Access, 4, pp.3273- 3287.

[19] Pujahari, A. and Padmanabhan, V., 2015, December. Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques. In Information Technology (ICIT), 2015 International Conference on (pp. 148-152). IEEE

[20] Liu, H., Hu, Z., Mian, A., Tian, H. and Zhu, X., 2014. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 56, pp.156-166.

[21] Mehta, B., 2007, July. Unsupervised shilling detection for collaborative filtering. In AAAI (pp. 1402-1407).

[22] Nadimi-Shahraki, M.H. and Bahadorpour, M., 2014. Cold-start problem in collaborative recommender systems: efficient methods based on ask-to-rate technique. Journal of computing and information technology, 22(2), pp.105-113.

[23] Nasiri, M. and Minaei, B., 2016. Increasing prediction accuracy in collaborative filtering with initialized factor matrices. The Journal of Supercomputing, 72(6), pp.2157-2169.

[24] Pujahari, A. and Padmanabhan, V., 2015, December. Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques. In Information Technology (ICIT), 2015 International Conference on (pp. 148-152). IEEE.

[25] Sharma, R., Gopalani, D. and Meena, Y., 2017, February. Collaborative filtering-based recommender system: Approaches and research challenges. In Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference on (pp. 1-6). IEEE.

[26] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on(Vol. 3, pp. 147-150). IEEE.

[27] Tewari, A.S. and Barman, A.G., 2016, December. Collaborative book recommendation system using trust based social network and association rule mining. In Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on (pp. 85-88). IEEE.

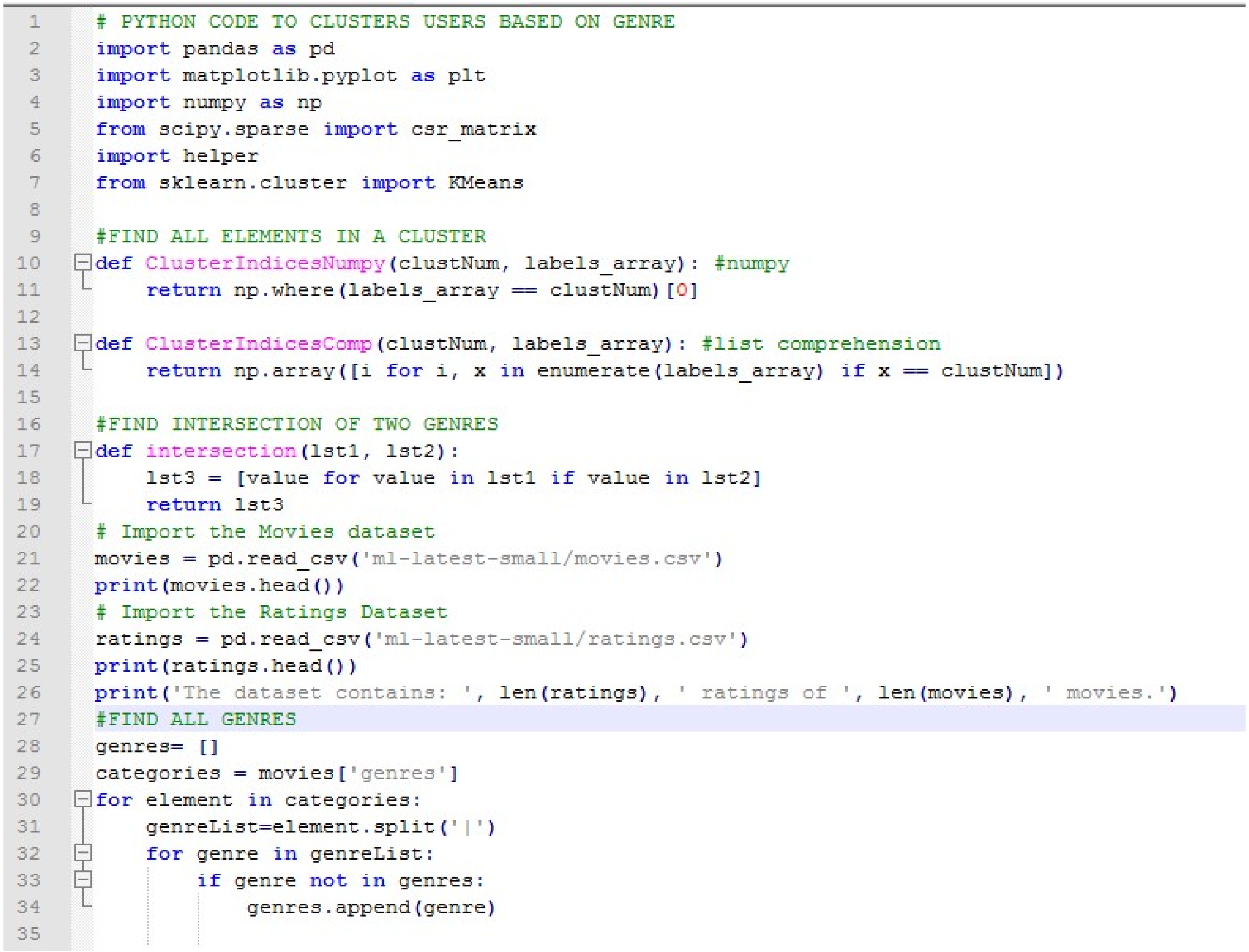
[28] Wang, P. and Ye, H., 2009, April. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. In Industrial and Information Systems, 2009. IIS'09. International Conference on (pp. 152-154). IEEE.

[29] Xie, F., Chen, Z., Shang, J., Huang, W. and Li, J., 2015, March. Item similarity learning methods for collaborative filtering recommender systems. In Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on (pp. 896-903). IEEE.

[30] Yang, Z., Wu, B., Zheng, K., Wang, X. and Lei, L., 2016. A survey of collaborative filtering-based recommender systems for mobile Internet applications. IEEE Access, 4, pp.3273- 3287.

APPENDIX A

USER INTERFACE CODE



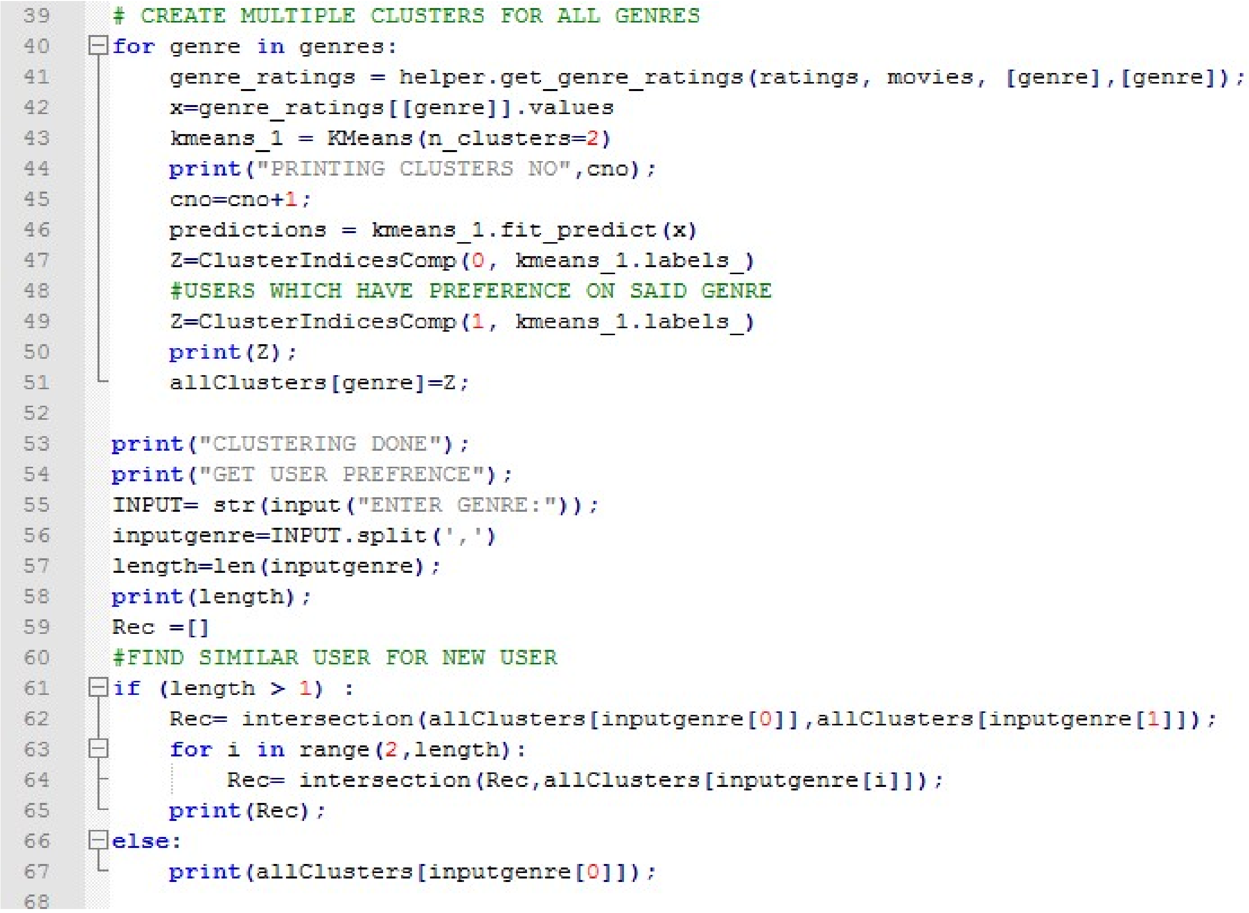
```
# PYTHON CODE TO CLUSTERS USERS BASED ON GENRE
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.sparse import csr_matrix
import helper
from sklearn.cluster import KMeans

#FIND ALL ELEMENTS IN A CLUSTER
def ClusterIndicesNumpy(clustNum, labels_array): #numpy
    return np.where(labels_array == clustNum)[0]

def ClusterIndicesComp(clustNum, labels_array): #list comprehension
    return np.array([i for i, x in enumerate(labels_array) if x == clustNum])

#FIND INTERSECTION OF TWO GENRES
def intersection(lst1, lst2):
    lst3 = [value for value in lst1 if value in lst2]
    return lst3
# Import the Movies dataset
movies = pd.read_csv('ml-latest-small/movies.csv')
print(movies.head())
# Import the Ratings Dataset
ratings = pd.read_csv('ml-latest-small/ratings.csv')
print(ratings.head())
print('The dataset contains: ', len(ratings), ' ratings of ', len(movies), ' movies.')
#FIND ALL GENRES
genres= []
categories = movies['genres']
for element in categories:
    genreList=element.split('|')
    for genre in genreList:
        if genre not in genres:
            genres.append(genre)

```

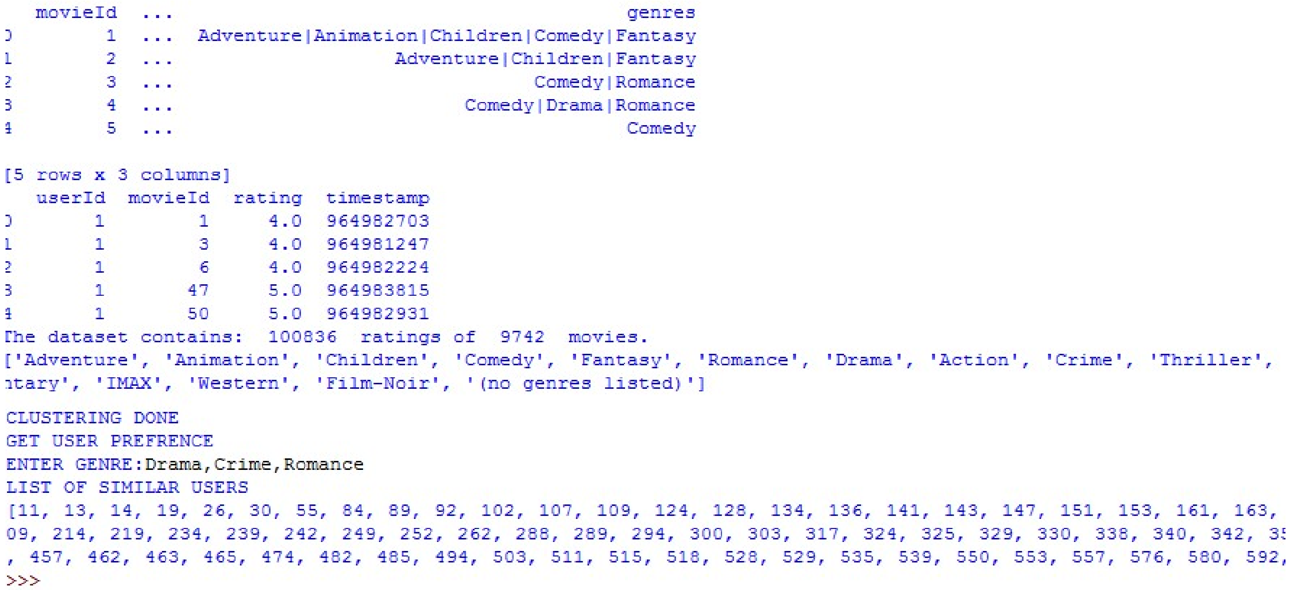


```
# CREATE MULTIPLE CLUSTERS FOR ALL GENRES
for genre in genres:
    genre_ratings = helper.get_genre_ratings(ratings, movies, [genre],[genre]);
    x=genre_ratings[[genre]].values
    kmeans_1 = KMeans(n_clusters=2)
    print("PRINTING CLUSTERS NO",cno);
    cno=cno+1;
    predictions = kmeans_1.fit_predict(x)
    Z=ClusterIndicesComp(0, kmeans_1.labels_)
    #USERS WHICH HAVE PREFERENCE ON SAID GENRE
    Z=ClusterIndicesComp(1, kmeans_1.labels_)
    print(Z);
    allClusters[genre]=Z;

print("CLUSTERING DONE");
print("GET USER PREFRENCE");
INPUT= str(input("ENTER GENRE:"));
inputgenre=INPUT.split(',')
length=len(inputgenre);
print(length);
Rec =[]
#FIND SIMILAR USER FOR NEW USER
if (length > 1) :
    Rec= intersection(allClusters[inputgenre[0]],allClusters[inputgenre[1]]);
    for i in range(2,length):
        Rec= intersection(Rec,allClusters[inputgenre[i]]);
    print(Rec);
else:
    print(allClusters[inputgenre[0]]);
```

APPENDIX B

PROGRAM OUTPUT SCREENSHOT



```
   movieId  ...                                       genres
0        1  ...  Adventure|Animation|Children|Comedy|Fantasy
1        2  ...                   Adventure|Children|Fantasy
2        3  ...                               Comedy|Romance
3        4  ...                         Comedy|Drama|Romance
4        5  ...                                       Comedy

[5 rows x 3 columns]
   userId  movieId  rating  timestamp
0       1        1     4.0  964982703
1       1        3     4.0  964981247
2       1        6     4.0  964982224
3       1       47     5.0  964983815
4       1       50     5.0  964982931
The dataset contains:  100836  ratings of  9742  movies.
['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy', 'Romance', 'Drama', 'Action', 'Crime', 'Thriller',
ntary', 'IMAX', 'Western', 'Film-Noir', '(no genres listed)']

CLUSTERING DONE
GET USER PREFRENCE
ENTER GENRE:Drama,Crime,Romance
LIST OF SIMILAR USERS
[11, 13, 14, 19, 26, 30, 55, 84, 89, 92, 102, 107, 109, 124, 128, 134, 136, 141, 143, 147, 151, 153, 161, 163,
09, 214, 219, 234, 239, 242, 249, 252, 262, 288, 289, 294, 300, 303, 317, 324, 325, 329, 330, 338, 340, 342, 3
, 457, 462, 463, 465, 474, 482, 485, 494, 503, 511, 515, 518, 528, 529, 535, 539, 550, 553, 557, 576, 580, 592,
>>>
```