BUILDING A RECOMMENDER SYSTEM USING MOVIELENS DATASET

INTRODUCTION

- Recommender system – Information filtering technique
- Widely used in ecommerce, streaming websites, email campaigns, loyalty programs etc.
- Types of recommendations:
  - Personalized
  - Non-personalized
- Recommendation Techniques:
  - Content Based Systems
  - Collaborative Filtering Techniques

Why Use Recommender Systems?

- The use of such systems is very important because of the following reasons:
  - To address the problem of Information Overload
  - To improve user experience
  - To increase revenue

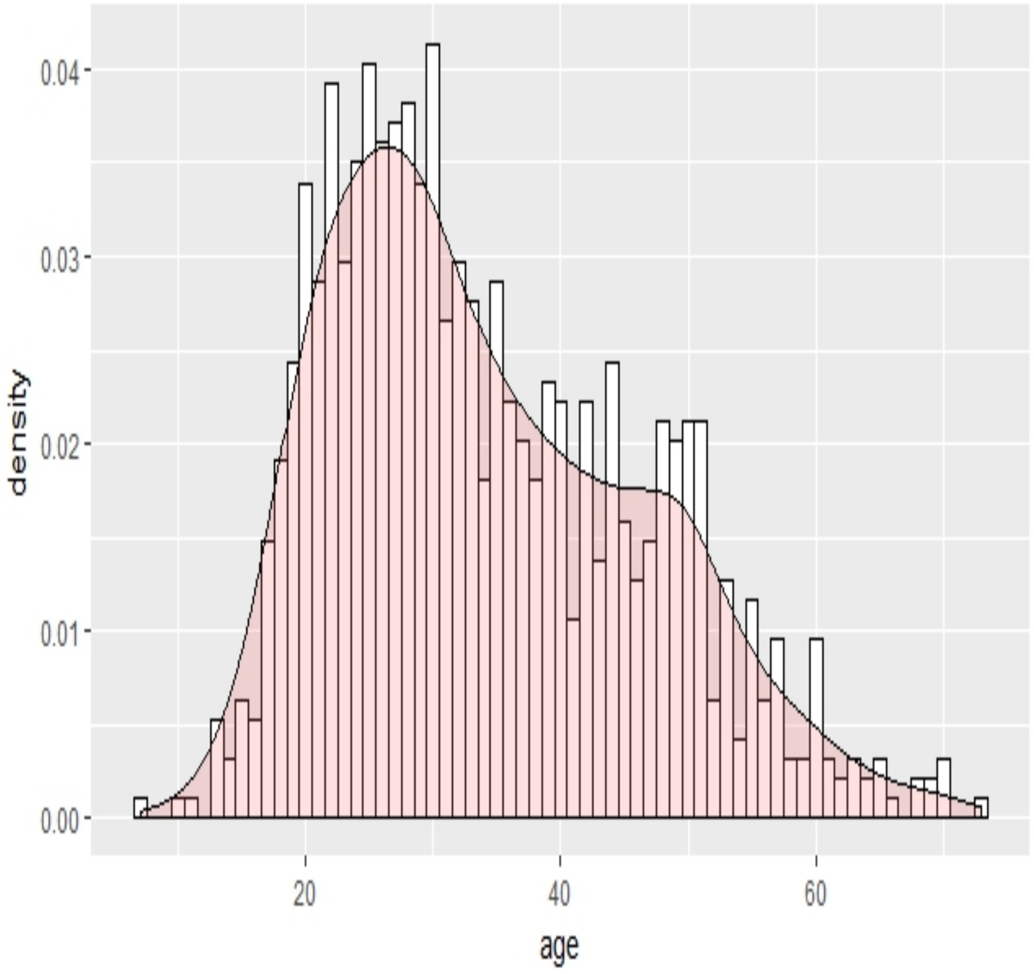
DATASET

- The dataset was obtained from Group Lens website.
- The data consisted of movie ratings for the past 20 years.
- It consisted of 9126 movies across 18 genres.
- It consisted of more than 100000 ratings by 671 users.
- The users were represented by an ID and information about their age, gender and occupation was also included.

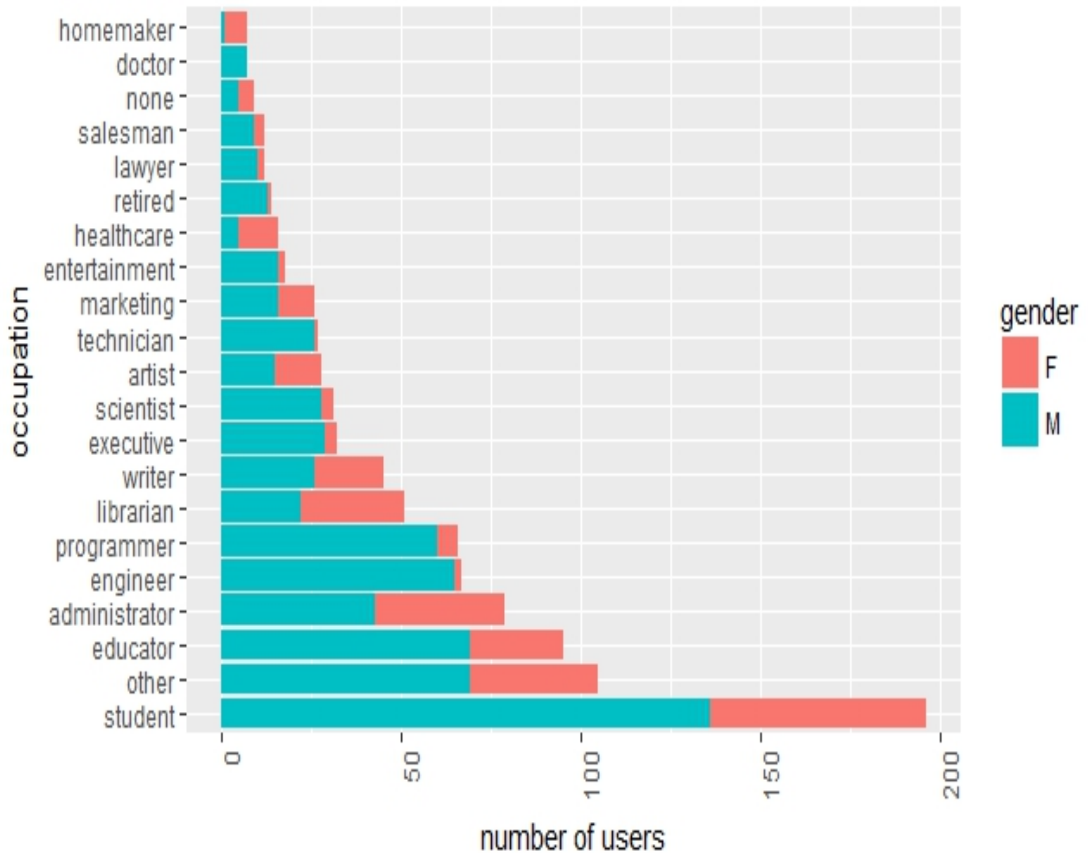
OBJECTIVE

- The following are our objectives:
  - To perform exploratory data analysis of the Movielens dataset
  - To develop a recommender system in order to provide personalized recommendations
  - To compare across Item Based and User Based Collaborative Filtering models

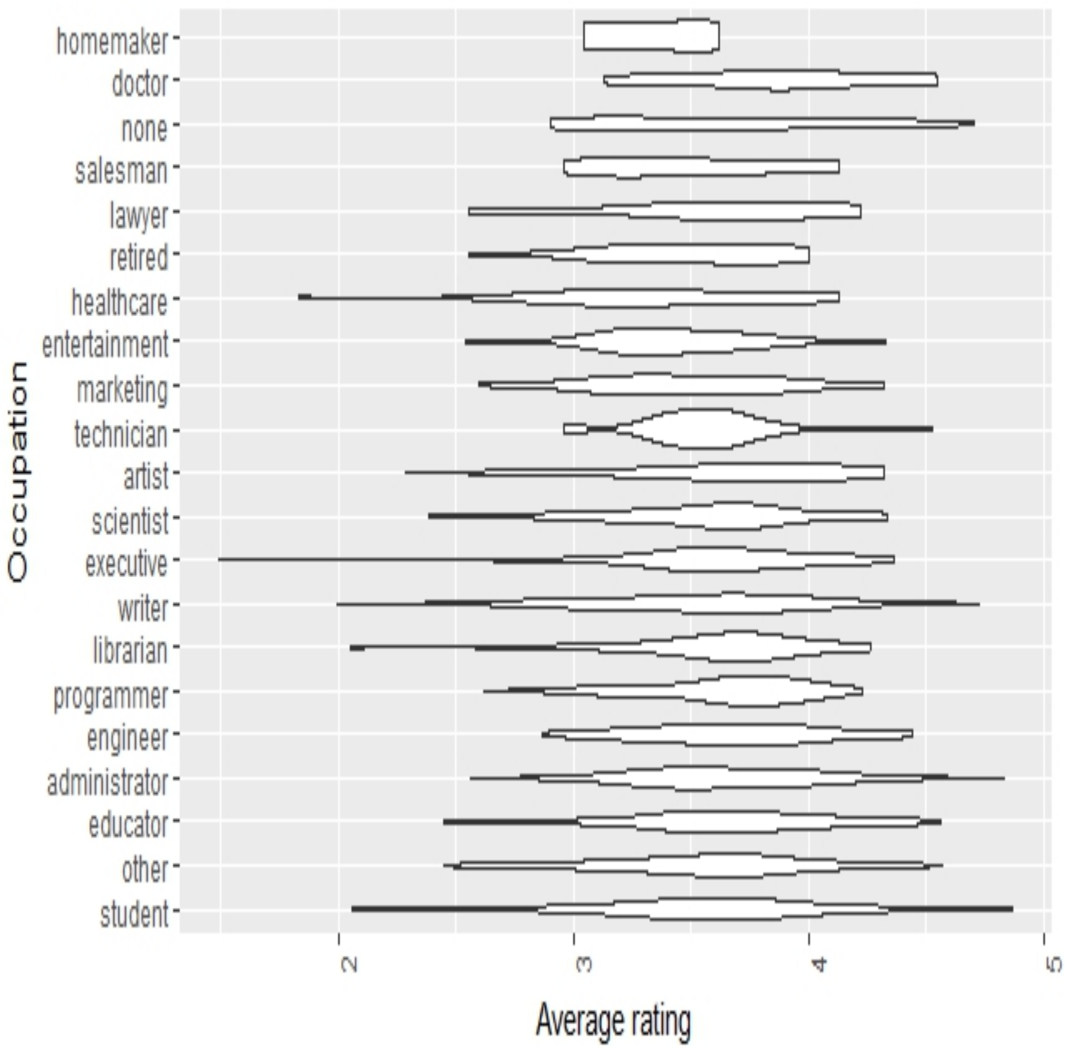
EXPLORATORY DATA ANALYSIS



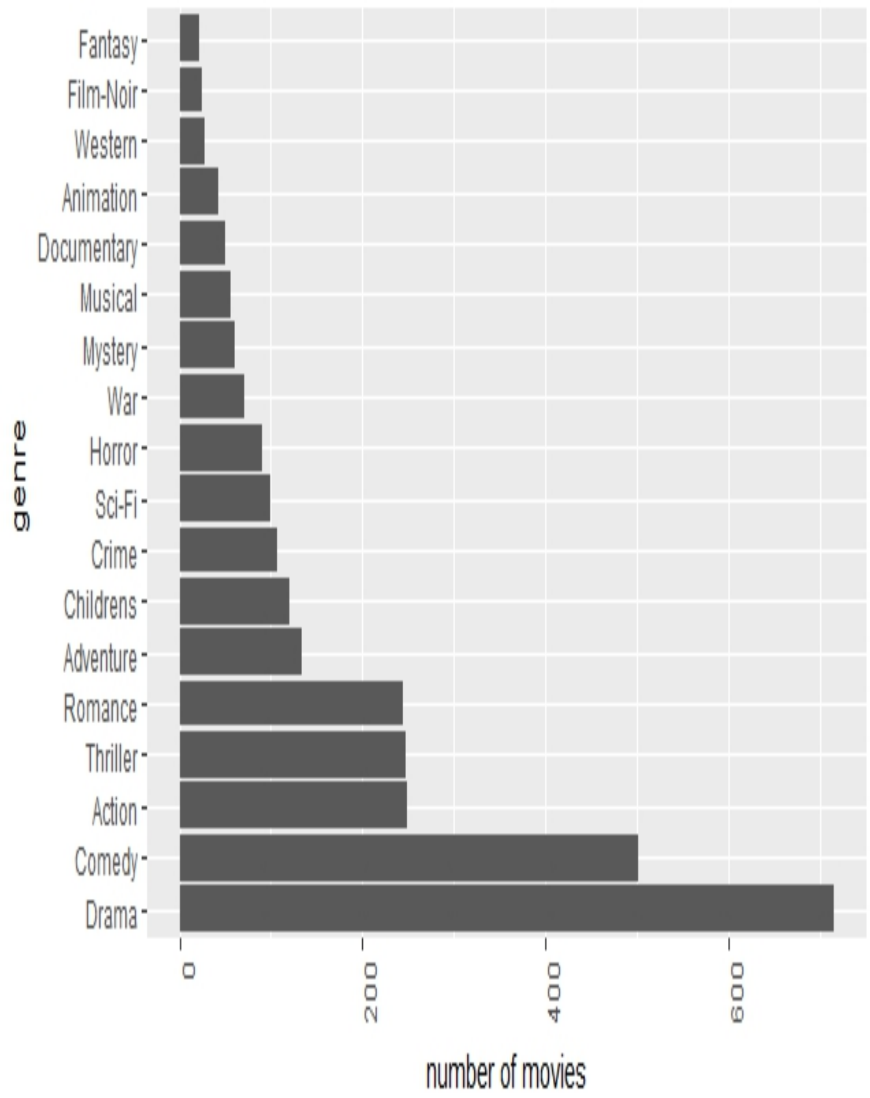
- Graph represents Age Vs Density
- Common age group seems to be late teens & mid thirties
- Also, there seems to be a small peak occurring in late forties.



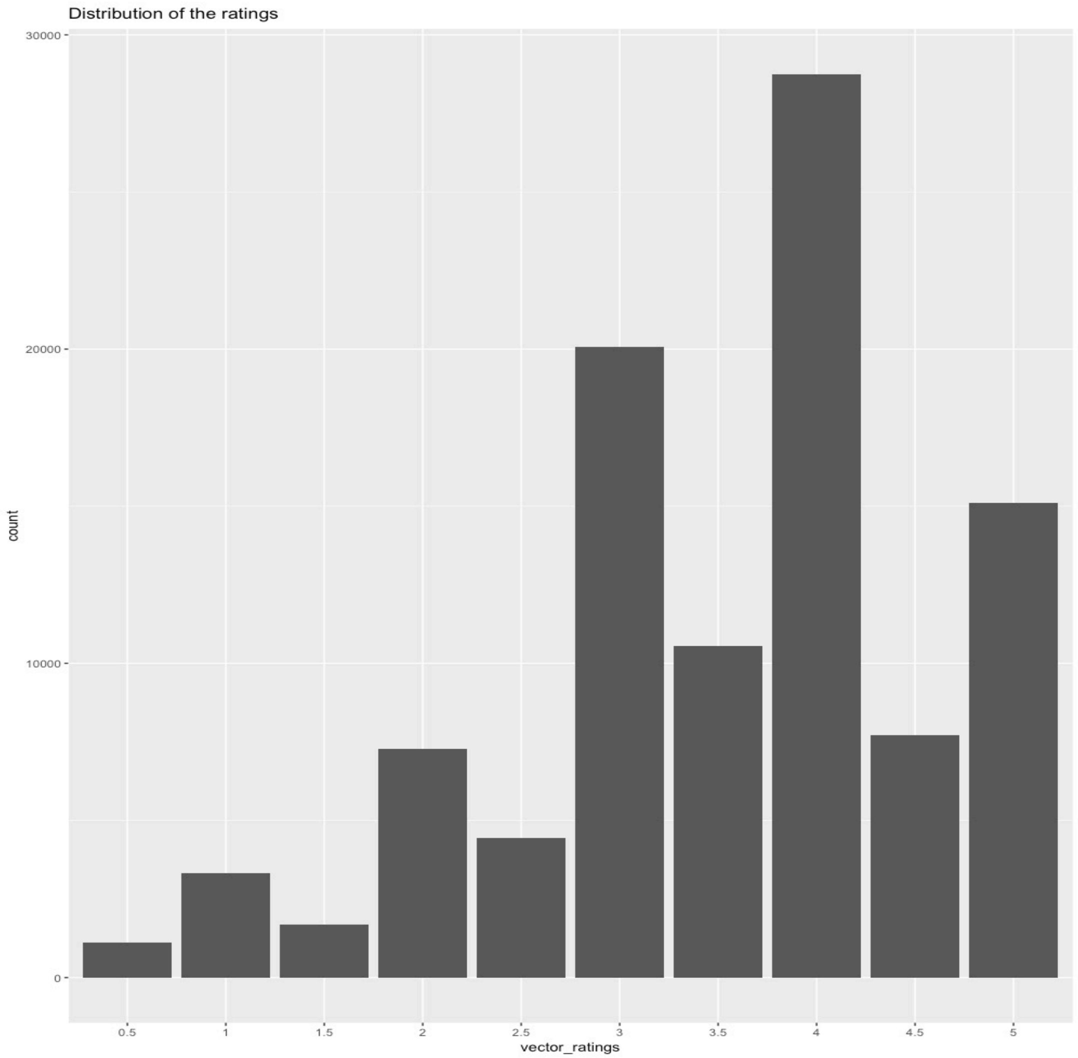
- Graph represents Number of Users Vs Occupation
- We can see that userbase is dominated by students while doctors and homemakers contribute the least to the userbase
- Overall more male users than female users



- Graph represents Average rating Vs Occupation
- It is evident that majority of ratings seem to be above average
- Executive and Healthcare workers tend to give a lower rating while students, administrators and doctors tend to give higher ratings



- Graph represents Genre Vs number of Movies
- Drama and comedy make up most of the movies in the dataset
- The least number of movies belong to the genres of fantasy, noir and western



- Graph represents Ratings Vs Count
- Majority of rating were 4 on a scale of 5
- Many users gave ratings of 3 & 5
- 0.5 and 1.5 received the least amount of ratings from users

COLLABORATIVE FILTERING

- Branch of recommendation that takes into account the information about various users
- ”Collaborative” refers to the fact that users collaborate with each other to recommend items.
- The basic assumptions of collaborative filtering are as follows:
  - Users with similar interests have common preferences
  - Large number of user preferences are available
- The two main approaches are:
  - Item Based Collaborative Filtering
  - User Based Collaborative Filtering

ITEM BASED COLLABORATIVE FILTERING

- Based on similarity between items
- The core algorithm is based on three important steps:
  - Measuring similarity between two items
  - Identifying k most similar items
  - Identifying items most similar to user’s purchases
- Data was divided into training and test set in 80:20 split

BUILDING THE RECOMMENDATION MODEL

```
recc_model <- Recommender(data = recc_data_train,
                          method = "IBCF",
                          parameter = list(k = 30))
```

- The model is built using Recommender function from recommmenderlab package
- The model extracts k similar items for every given movie rated by user
- The model was built on training set using parameters like cosine similarity and k = 30

APPLYING THE RECOMMENDER SYSTEM

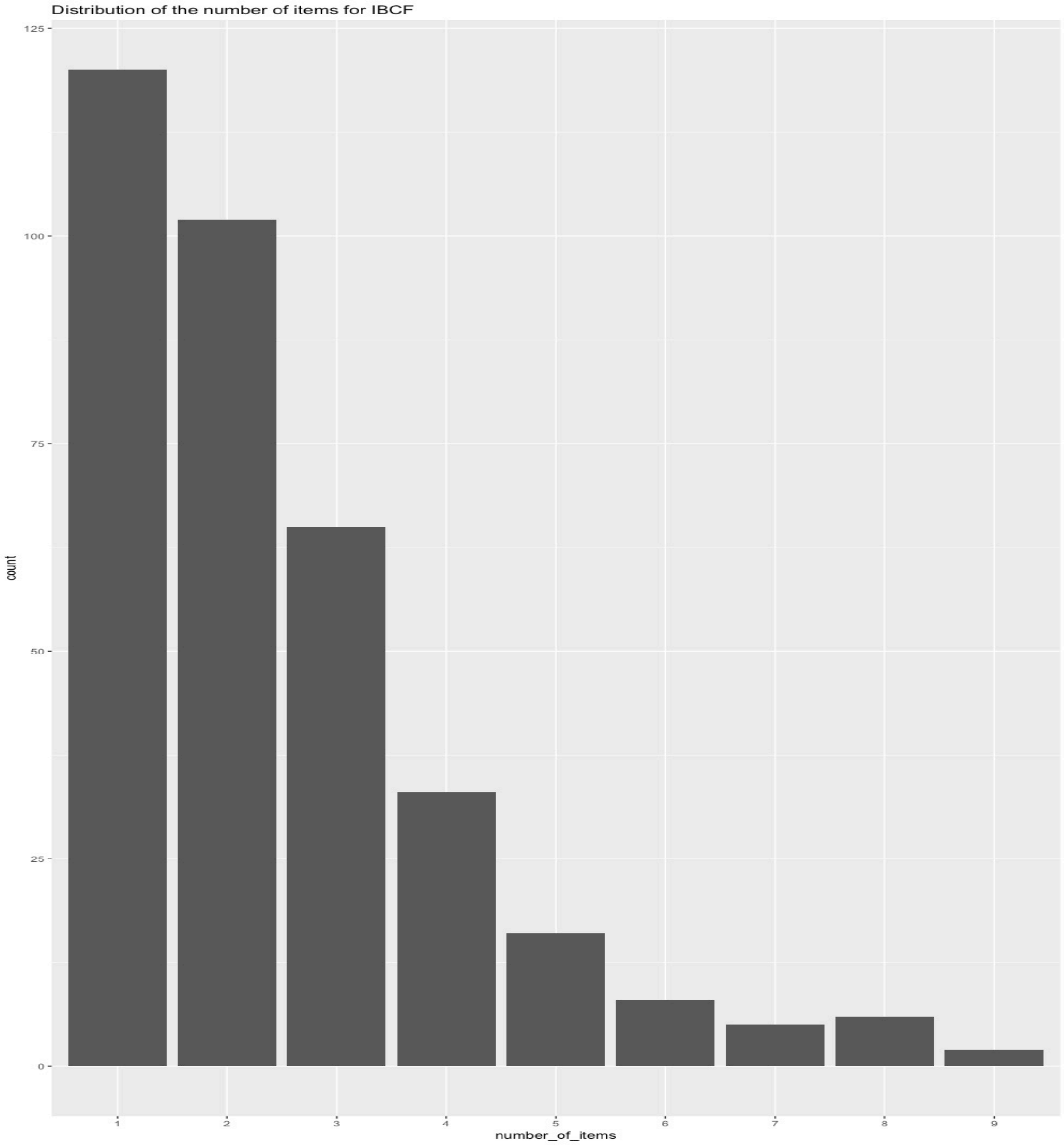
- The recommender system was applied on test set
- Algorithm extracts all movies rated by user and identifies similar items
- Algorithm ranks similar items in the following way:
  - Extract user rating of each purchase which is used as a weight
  - Extract similarity of the item with its associated purchases
  - Multiply each weight with related similarity
  - Sum everything up

ITEM BASED COLLABORATIVE FILTERING

```
 [1] "Toy Story (1995)"
 [2] "Casino (1995)"
 [3] "Sense and Sensibility (1995)"
 [4] "Leaving Las Vegas (1995)"
 [5] "Twelve Monkeys (a.k.a. 12 Monkeys) (1995)"
 [6] "Dead Man Walking (1995)"
 [7] "Seven (a.k.a. Se7en) (1995)"
 [8] "Usual Suspects, The (1995)"
 [9] "Taxi Driver (1976)"
[10] "Crimson Tide (1995)"
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	1	2	440	708	1204	36	110	223	111	34
[2,]	16	141	553	1676	1394	786	265	596	337	261
[3,]	17	708	914	2001	1201	1527	292	1682	778	661
[4,]	25	1183	1517	2406	1307	2321	1136	7147	903	1291
[5,]	32	1391	1527	2770	300	2424	1285	7438	904	1394
[6,]	36	1517	1580	3176	1259	4886	1302	33794	908	1641
[7,]	47	2011	1954	3793	1207	5218	1961	40815	912	1748
[8,]	50	912	2000	4896	1961	5816	1968	44191	913	2005
[9,]	111	509	2080	8665	6502	6377	2194	46578	924	2291
[10,]	161	337	2683	3994	2395	6378	2329	51662	1097	2355

- The image on the left shows the movies recommended to the User 1
- The image on the right shows the movies recommended to first 10 users in terms of movieID
- For example, movieID 2 represents “Jumanji” and movieID 34 represents “Babe”



	Movie title	No of items
745	Wallace & Gromit: A Close Shave (1995)	9
1148	Wallace & Gromit: The Wrong Trousers (1993)	9
50	Usual Suspects, The (1995)	8
246	Hoop Dreams (1994)	8

- The x-axis in the graph shows the number of movies recommended to a user based on his/her rating.
- The y-axis represents the number of times users were recommended certain amount of movies
- For example, If a user rated Usual Suspects, then he/she was probably recommended 8 similar movies

USER BASED COLLABORATIVE FILTERING

- Given a user, similar users are first identified and top rated items by similar users are recommended
- Algorithm follows the following steps:
  - Measure how similar a user is to a new one
  - Identify most similar users
  - Rate movies rated by most similar users
  - Picks the top rated movies

BUILDING THE RECOMMENDATION MODEL

- The model is built using Recommender function from recommmenderlab package
- The model extracts k similar items for every given movie rated by user
- The model was built on training set using parameters like cosine similarity and k = 30

APPLYING THE RECOMMENDATION MODEL

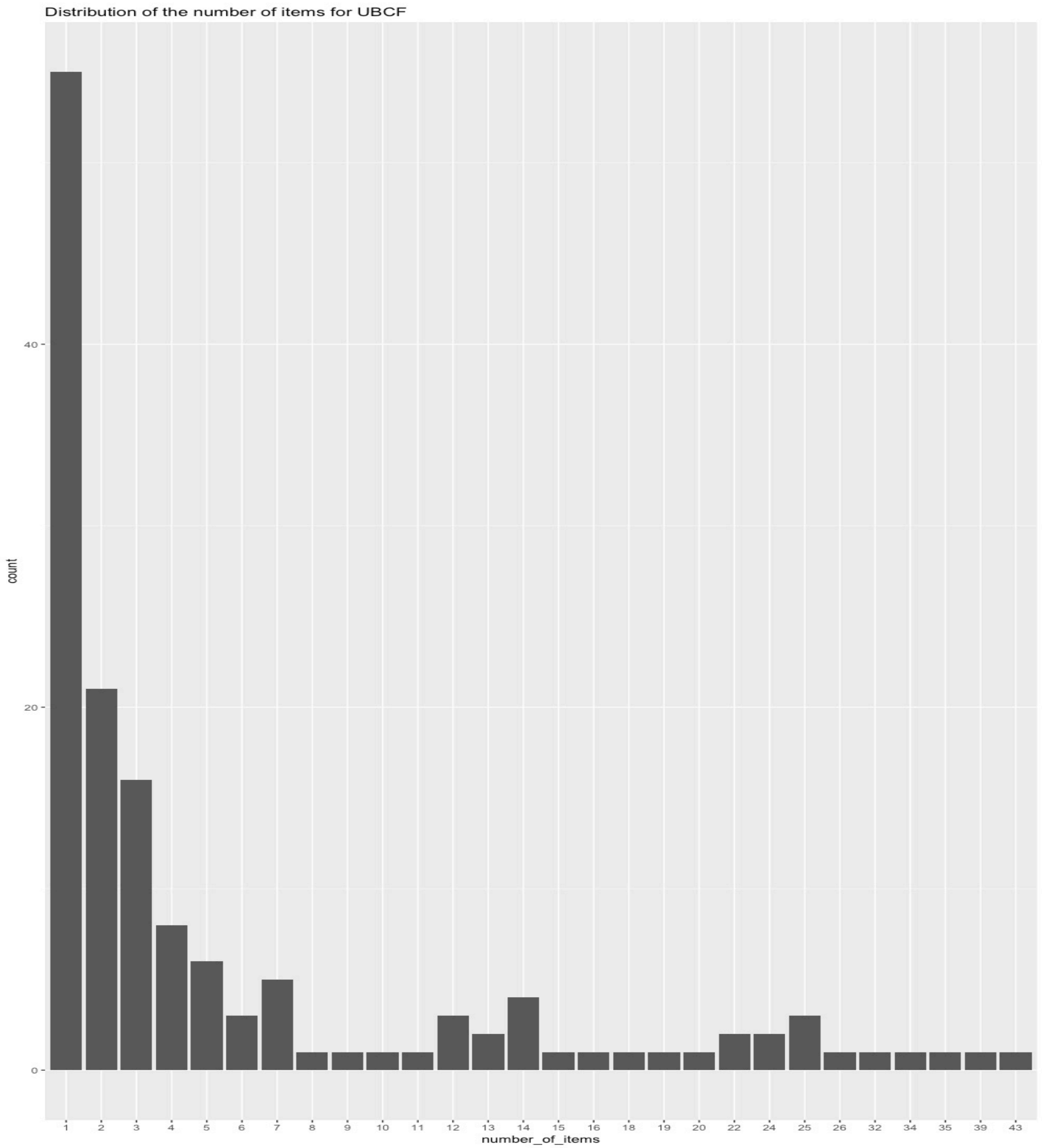
- The recommender system was applied on test set
- Algorithm extracts all movies rated by user and identifies similar items
- Algorithm ranks similar items in the following way:
  - Extract user rating of each purchase which is used as a weight
  - Extract similarity of the item with its associated purchases
  - Multiply each weight with related similarity
  - Sum everything up

USER BASED COLLABORATIVE FILTERING

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	318	608	2762	1197	318	296	296	1196	50	858
[2,]	2959	778	858	5952	1221	593	1213	1210	150	296
[3,]	50	541	527	4306	2028	50	1193	296	608	1208
[4,]	47	1213	4993	539	2858	111	318	58559	5952	2019
[5,]	527	1206	2571	58559	1196	2858	778	50	1291	541
[6,]	593	1230	7153	50	50	920	2858	2028	260	1230
[7,]	2571	1252	3996	1356	1213	1247	1196	47	7153	111
[8,]	4993	111	111	54286	47	750	47	858	1036	1221
[9,]	293	4878	1196	4886	1197	246	1288	1291	5349	1244
[10,]	223	912	1198	587	260	1213	50	1197	2571	1206

- The image on the right shows the movies recommended to first 10 users in terms of movieID
- For example, movieID 50 represents “Usual Suspects” and movieID 318 represents “Shawshank Redemption”

	Movie title	No of items
50	Usual Suspects, The (1995)	43
296	Pulp Fiction (1994)	39
318	Shawshank Redemption, The (1994)	35
858	Godfather, The (1972)	34



- The x-axis in the graph shows the number of movies recommended to a user based on his/her rating.
- The y-axis represents the number of times users were recommended certain amount of movies
- For example, If a user rated Pulp Fiction, then he/she was probably recommended 39 similar movies

COMPARISON BETWEEN MODELS

- Defined the different models as a list using the cosine as distance function for both IBCF and UBCF.

- Used random recommendations in order to have a base line

- The below tables shows the performance evaluation matrix

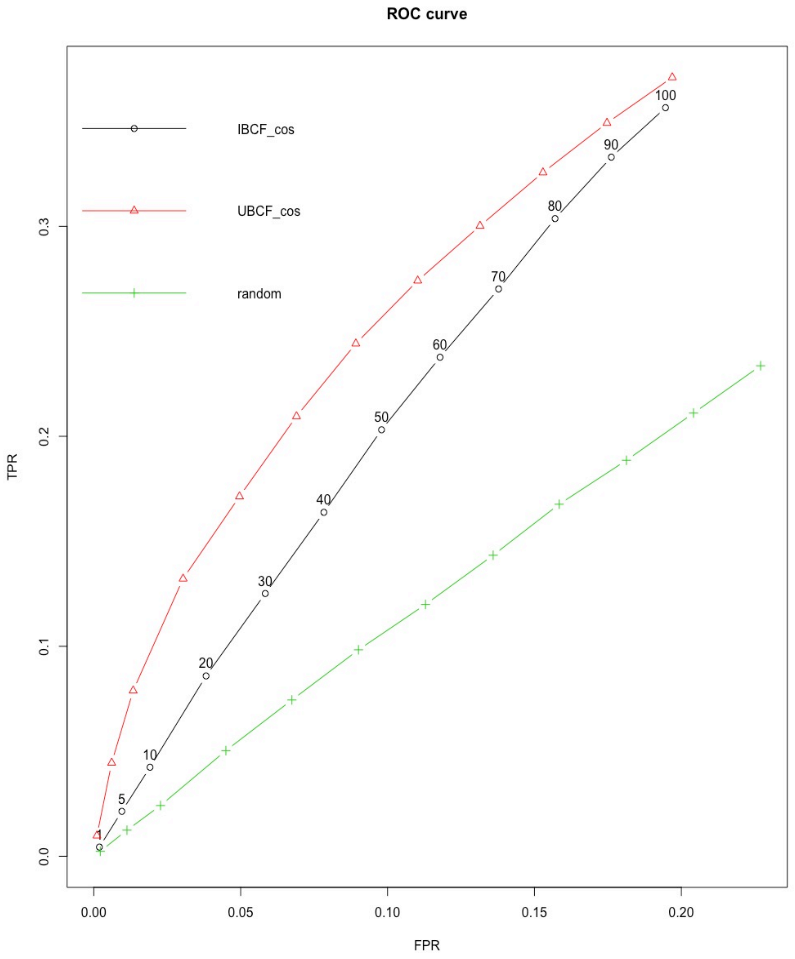
IBCF

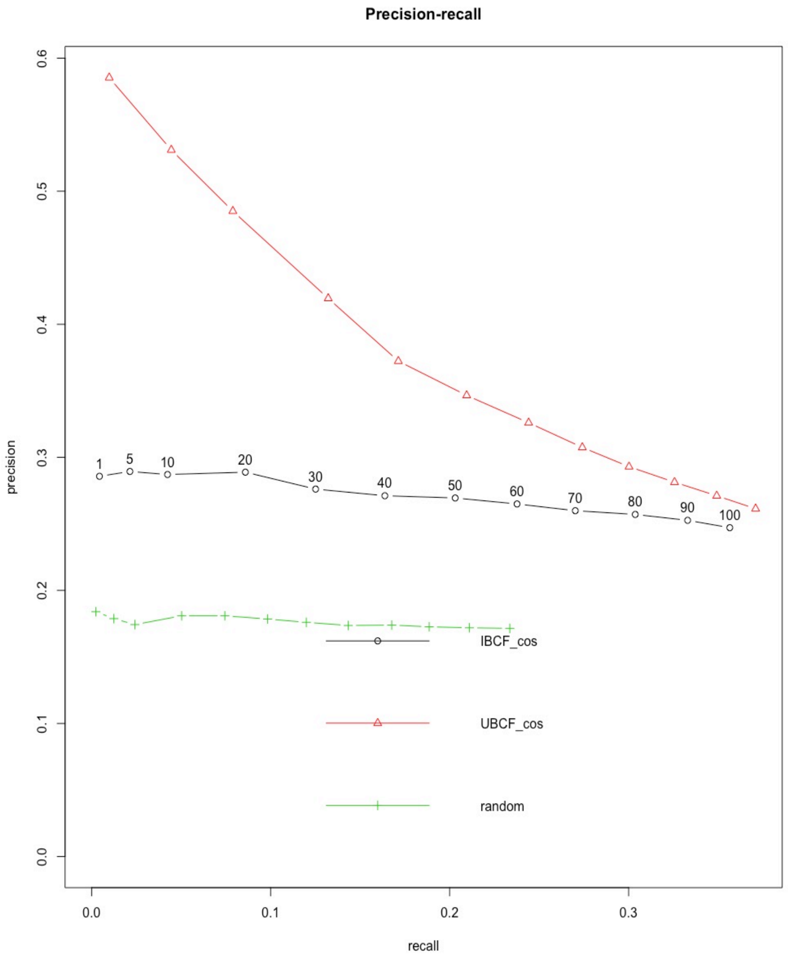
	precision	recall	TPR	FPR
1	0.2857544	0.004376754	0.004376754	0.001916081
5	0.2893674	0.021390406	0.021390406	0.009553096
10	0.2870825	0.042395607	0.042395607	0.019143837
20	0.2888419	0.085880516	0.085880516	0.038214141
30	0.2761230	0.125128732	0.125128732	0.058365477
40	0.2711260	0.163801778	0.163801778	0.078317304
50	0.2694407	0.203105016	0.203105016	0.097951799
60	0.2649466	0.237659616	0.237659616	0.117863727
70	0.2598896	0.270193784	0.270193784	0.137791395
80	0.2570686	0.303685870	0.303685870	0.157011794
90	0.2526504	0.332952498	0.332952498	0.176166322
100	0.2471263	0.356502672	0.356502672	0.194608982

UBCF

	precision	recall	TPR	FPR
1	0.5853388	0.009818739	0.009818739	0.001077148
5	0.5310120	0.044503998	0.044503998	0.006095563
10	0.4850403	0.078888547	0.078888547	0.013417315
20	0.4194372	0.132176765	0.132176765	0.030399001
30	0.3722894	0.171345282	0.171345282	0.049603696
40	0.3465191	0.209521525	0.209521525	0.069002228
50	0.3259922	0.244137342	0.244137342	0.089200730
60	0.3074520	0.274128717	0.274128717	0.110267207
70	0.2929706	0.300211465	0.300211465	0.131467707
80	0.2813107	0.325633323	0.325633323	0.152895421
90	0.2710997	0.349208392	0.349208392	0.174680710
100	0.2613692	0.370931376	0.370931376	0.196905273

The graphs show the ROC curve & Precision/Recall curves





CONCLUSION

- Two recommendation systems were developed and we found that User based system is better than item based due to the following reasons:
  - Accuracy of User based was higher compared to Item based system
  - Provides stronger recommendation as users might not be looking for direct substitutes for a movie that they previously watched
- Although user based performed better than item based, it was computationally time consuming.
- Future work can include the implementation of some form of dimensionality reduction like PCA in order to reduce computational time.

THANK YOU