Movie Recommender System

-Omkar Mutreja

-Qiamu Li

-Harsh Darji

1. Introduction
2. Objective and Business Questions
3. Data Description
4. Data Preprocessing and Preparation
5. Data Analysis
6. Challenges
7. Conclusion

1.Introduction

In today's world, every customer is faced with multiple choices. For example, If I'm looking for a book to read without any specific idea of what I want, there's a wide range of possibilities how my search might pan out. I might waste a lot of time browsing around on the internet and trawling through various sites hoping to strike gold. I might look for recommendations from other people. Also, at some point each one of us must have wondered where all the recommendations that Netflix, Amazon, Google give us, come from. We often rate products on the internet and all the preferences we express and data we share (explicitly or not), are used by recommender systems to generate, in fact, recommendations. In this project, we have developed a collaborative filtering recommender system for recommending movies. The basic idea of CFR systems is that, if two users share the same interests in the past, e.g. they liked the same book or the same movie, they will also have similar tastes in the future. If, for example, user A and user B have a similar purchase history and user A recently watched a movie that user B has not yet seen, the basic idea is to propose this movie to user B.

2.Objective and Business Questions

In this project, we develop a collaborative filtering recommender (CFR) system for recommending movies with the main objective of recommending relevant movies to the users. In the era of competitors like Amazon Prime, Netflix, Hulu, companies want maximum users to stick to their product which will result in higher profits. So, this recommender system plans to provide optimized recommendation of movies to increase viewership on our product. This project plans to give vital information to marketing department so that they can market a movie in order to make profits. Also, the PR team and Design team will use the information generated from this project to place movies efficiently on the website and target specific segment of customers.

There are two approaches develop a recommender system:

- Collaborative filtering: Collaborative filtering approaches build a model from user's past behavior (i.e. items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that user may have an interest in.

- Content-based filtering: Content-based filtering approaches uses a series of discrete characteristics of an item in order to recommend additional items with similar properties.
Content-based filtering methods are totally based on a description of the item and a profile of the user's preferences. It recommends items based on user's past preferences.

3.Data Description

The dataset used was from MovieLens and is publicly available at https://grouplens.org/datasets/movielens/latest/ . To keep the recommender simple, we used the smallest dataset available (ml-latest-small.zip), which at the time of download contaied 105339 ratings

and 6138 tag applications across 10329 movies. These data were created by 668 users between April 03, 1996 and January 09, 2016. This dataset was generated on January 11, 2016.

The data are contained in four files: links.csv, movies.csv, ratings.csv and tags.csv. We only used the files movies.csv and ratings.csv to build a recommendation system.

Movies:

The main attributes related to movies were:

- MovieId – Unique ID to identify the Movie
- Title – Describes the name of the Movie (Toy Story, Jumanji..)
- Genre – Describes the genre of the Movie (Comedy, Action..)

Summary is as follows:

```
##     movieId          title              genres
##  Min.   :     1   Length:10329       Length:10329
##  1st Qu.:  3240   Class :character   Class :character
##  Median :  7088   Mode  :character   Mode  :character
##  Mean   : 31924
##  3rd Qu.: 59900
##  Max.   :149532
```

```
##   movieId                                       title
## 1       1                           Toy Story (1995)
## 2       2                             Jumanji (1995)
## 3       3                    Grumpier Old Men (1995)
## 4       4                   Waiting to Exhale (1995)
## 5       5 Father of the Bride Part II (1995)
## 6       6                                Heat (1995)
##                                              genres
## 1 Adventure|Animation|Children|Comedy|Fantasy
## 2                  Adventure|Children|Fantasy
## 3                                Comedy|Romance
## 4                         Comedy|Drama|Romance
## 5                                       Comedy
## 6                          Action|Crime|Thriller
```

Ratings:

The main attributes related to Ratings were:

- Movie ID – Unique ID to identify the Movie
- User ID – Unique ID to identify the User
- Rating – Describes the rating of the movie (x out of 5)

Summary is as follows:

```
##      userId        movieId          rating       timestamp
##  Min.   :  1.0   Min.   :     1   Min.   :0.500   Min.   :8.286e+08
##  1st Qu.:192.0   1st Qu.:  1073   1st Qu.:3.000   1st Qu.:9.711e+08
##  Median :383.0   Median :  2497   Median :3.500   Median :1.115e+09
##  Mean   :364.9   Mean   : 13381   Mean   :3.517   Mean   :1.130e+09
##  3rd Qu.:557.0   3rd Qu.:  5991   3rd Qu.:4.000   3rd Qu.:1.275e+09
##  Max.   :668.0   Max.   :149532   Max.   :5.000   Max.   :1.452e+09
```

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

4.Data Preprocessing and Preparation

Some pre-processing was performed on the data before running the model. Firstly, most of the movies have their debut year added to their names - we want to extract this into separate columns. (EG: Skokie(1981) to Title - Skokie Year- 1981.

movieId	title	year	genres
8359	Skokie (1981)	NA	Drama
26815	Deadly Advice(1994)	NA	Comedy|Drama
40697	Babylon 5	NA	Sci-Fi
79607	Millions Game, The (Das Millionenspiel)	NA	Action|Drama|Sci-Fi|Thriller
87442	Bicycle, Spoon, Apple (Bicicleta, cullera, poma)	NA	Documentary

movieId	title	year
27914	Hijacking Catastrophe: 9/11, Fear & the Selling of American Empire	2004
72235	Between the Devil and the Deep Blue Sea	1995
88488	Summer Wishes, Winter Dreams	1973
92783	Latin Music USA	2009
93967	Keeping the Promise (Sign of the Beaver, The)	1997

Create a matrix to search for a movie by genre:

```
##   movieId                              title Action Adventure Animation
## 1       1                   Toy Story (1995)      0         1         1
## 2       2                     Jumanji (1995)      0         1         0
## 3       3            Grumpier Old Men (1995)      0         0         0
## 4       4           Waiting to Exhale (1995)      0         0         0
## 5       5 Father of the Bride Part II (1995)      0         0         0
## 6       6                          Heat (1995)      1         0         0
##   Children Comedy Crime Documentary Drama Fantasy Film-Noir Horror Musical
## 1        1      1     0           0     0       1         0      0       0
## 2        1      0     0           0     0       1         0      0       0
## 3        0      1     0           0     0       0         0      0       0
## 4        0      1     0           0     1       0         0      0       0
## 5        0      1     0           0     0       0         0      0       0
## 6        0      0     1           0     0       0         0      0       0
##   Mystery Romance Sci-Fi Thriller War Western
## 1       0       0      0        0   0       0
## 2       0       0      0        0   0       0
## 3       0       1      0        0   0       0
## 4       0       1      0        0   0       0
## 5       0       0      0        0   0       0
## 6       0       0      0        1   0       0
```

Binarizing data

Some recommendation models work on binary data, so it might be useful to binarize the data, that is, define a table containing only 0s and 1s. The 0s will be either treated as missing values or as bad ratings.

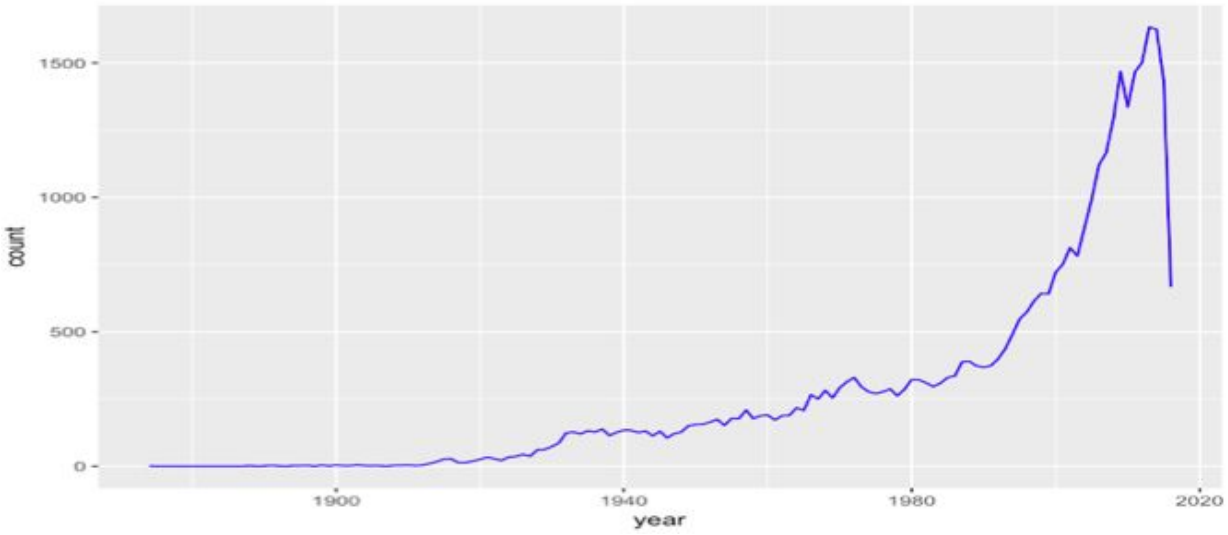
In our case, we can either:

- Define a matrix having 1 if the user rated the movie, and 0 otherwise. In this case, the information about the rating is lost.
- Define a matrix having 1 if the rating is above or equal to a definite threshold (for example, 3), and 0 otherwise. In this case, giving a bad rating to a movie is equivalent to not having rated it.

We have used the second approach where movies rated with rating greater than 3 are treated as 1 and movies rating less than 3 are treated as 0.

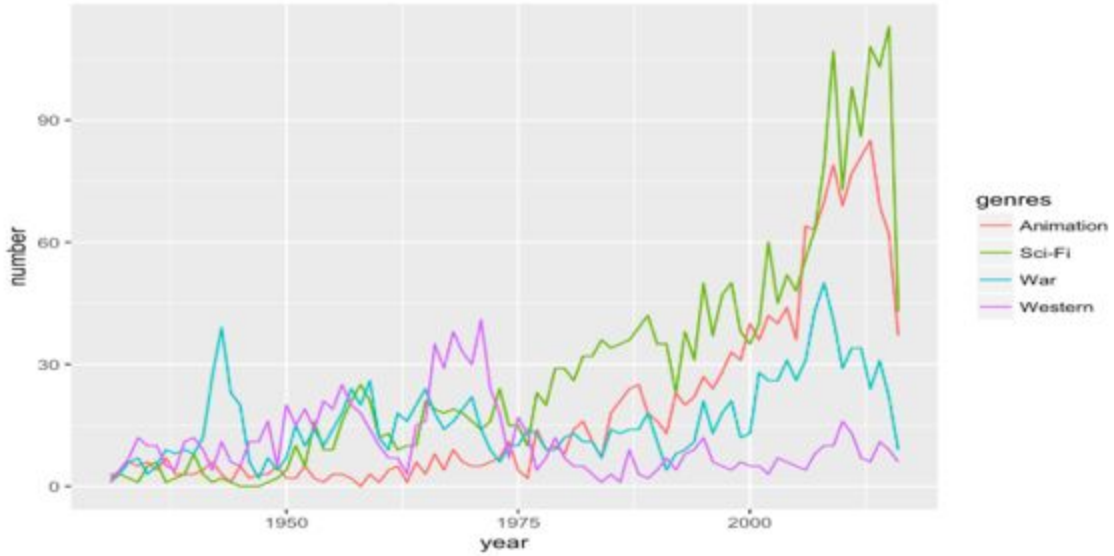
Exploratory Data Analysis:

Descriptive Statistics



We can see an exponential growth of the movie business and a sudden drop in 2016 The latter is caused by the fact that the data is collected until October 2016 so we don't have the full data on this year. Growing popularity of the Internet must have had a positive impact on the demand for movies. That is certainly something worthy of further analysis.

These movies are from different genres like Comedy, Romance, Super Hero, Sci-Fi, Animation, War and so on.



We can see that Sci-Fi movies have been increasing exponentially as compared to other genres. Animation movies have also increased in the last 20 years which means that people love

enjoying these kind of genres. War movies were popular around the time when big military conflicts occured - World War II, Vietnam War and most recently War in Afghanistan and Iraq. It's interesting to see how the world of cinematography reflected the state of the real world.

We can clearly see from the word cloud that Sci-fi, superhero, space, dystopia and social commentary are some of the most occuring genres in our dataset.

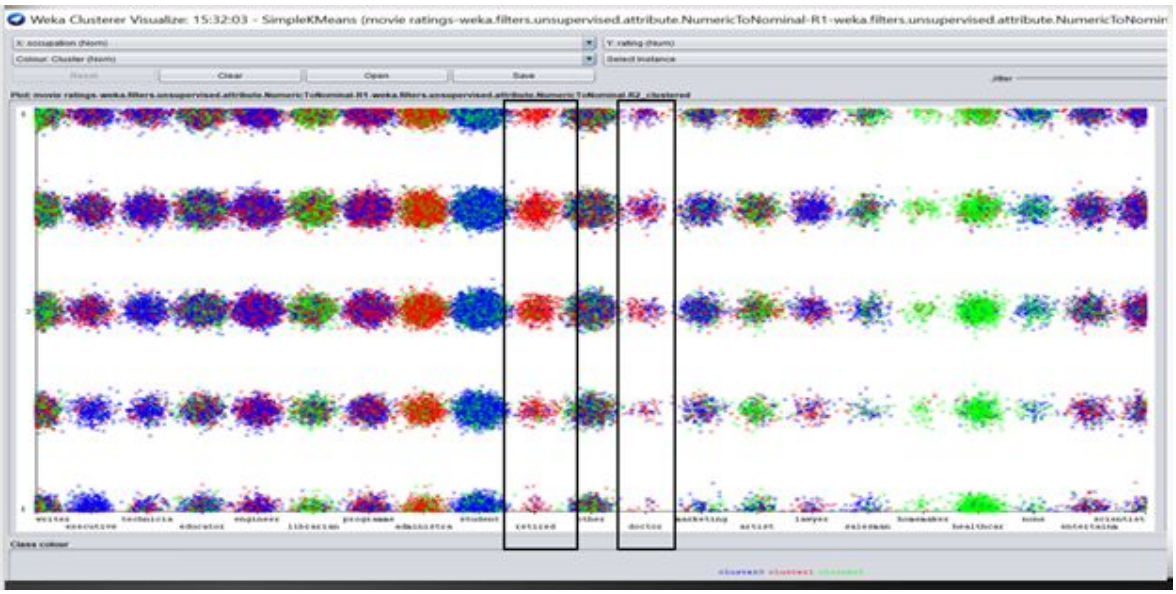
5.Data Analysis

In this part, we perform K-Means Clustering, Item-Based Collaborative Filtering and User-Based Collaborative Filtering models.

K-Means Clustering: A cluster refers to a collection of data points aggregated together because of certain similarities.

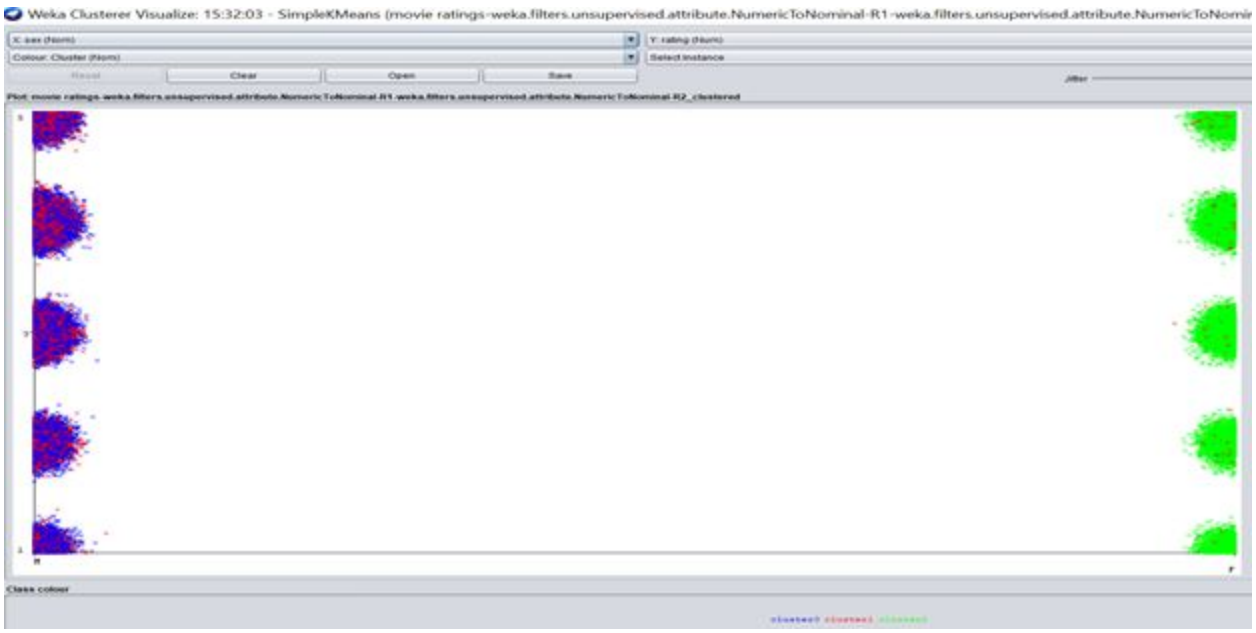
Here we perform k-means clustering to identify if there are any set of users who have similar characteristics in terms of rating a movie or depending on sex. We perform k-means based on occupation to rating ,sex to rating and age to rating.

Occupation to Rating



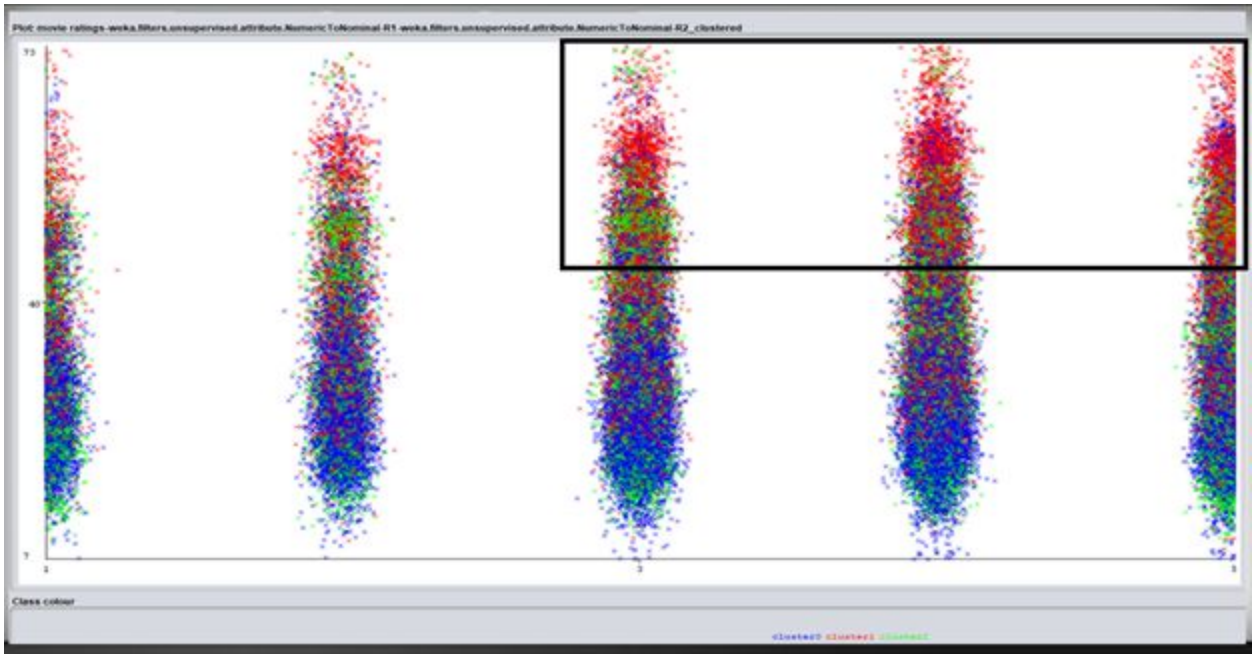
Some occupations such as retired people and doctors are more likely to give high ratings. We can see that the red circle and purple circle increases in intensity as we move up on the y-axis for retired and doctors.

Sex to Rating



Sex difference nearly makes no apparent effect on ratings which means that there are no clusters based on sex.

Age to Rating



Those people(age>40) are inclined to rate higher than those younger people.

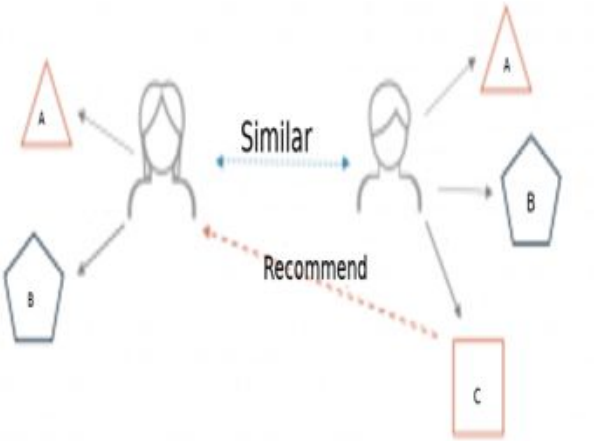
Recommendation System:

There are two kinds of models that we have implemented to build our recommendation system. Item based collaborative filtering and user based collaborative filtering. We have used Pearson and Cosine as the distance function to calculate the similarity between the items.

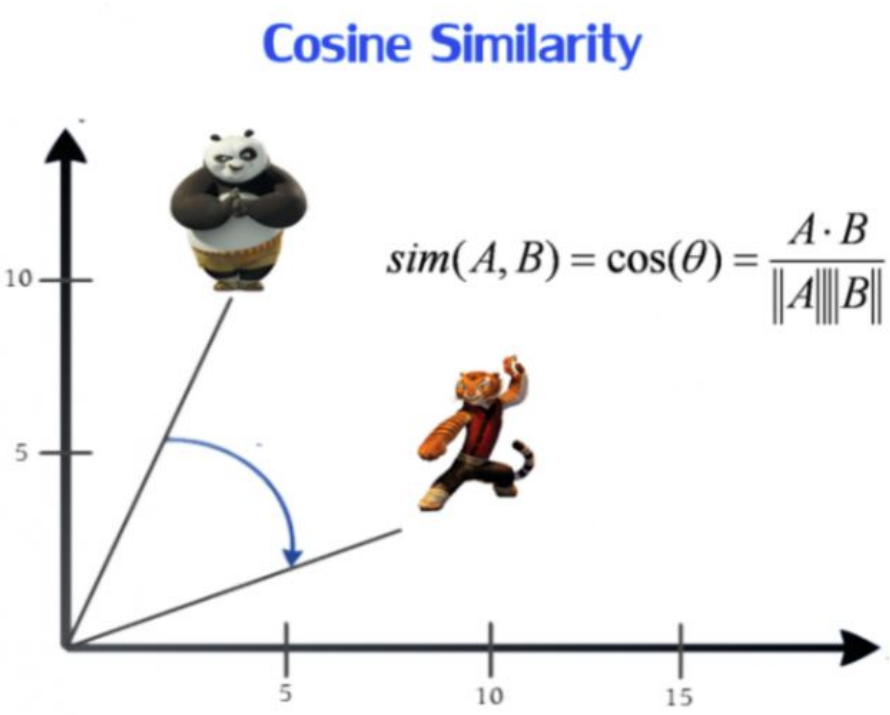
Pearson Correlation - Measures the association between the users and movies based on different aspects like ratings and genre.

The correlation is a numerical values between -1 and 1 that indicates how much two variables are related to each other. Correlation = 0 means no correlation, while >0 is positive correlation and <0 is negative correlation.

$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$



Cosine Similarity - Cosine similarity measures the similarity between two vectors by calculating the cosine of the angle between them. A simple understanding of this phenomenon:



The cosine similarity is the dot product of two vectors divided by the product of the magnitude of each vector. The reason we divide the dot product by the magnitude is because we are measuring only angle difference and the dot product is taking the angle difference and

magnitude into account. The cosine of a 0 degree angle is 1, therefore the closer to 1 the cosine similarity is the more similar the items are.

ITEM-based Collaborative Filtering Model

Like the name proposes, the Content-based Filtering approach includes examining a thing a client associated with, and giving suggestions that are comparative in substance to that thing. Content, for this situation, alludes to a lot of traits/includes that depicts your thing. For a film suggestion motor, a substance based methodology is prescribe motion pictures that are of most astounding comparability dependent on its highlights, for example, classifications, entertainers, executives, year of creation, and so on. The suspicion here is that clients have inclinations for a particular kind of item, so we endeavor to prescribe a comparable item to what the client has communicated preferring for. Likewise, the objective here is to give choices or substitutes to the thing that was seen.

Now that we have the user profiles we created in pre-processing step, we can go 2 ways from here.

1) Predict if a user likes an item based on the item descriptions (movie genres). This can be done by predicting user movie ratings.

2) Assume that users like similar items, and retrieve movies that are closest in similarity to a user's profile, which represents a user's preference for an item's feature.

Movies recommended to user 1 after applying item based filtering:

	movieId	title	genres
1564	2015	Absent-Minded Professor, The (1961)	Children|Comedy|Fantasy
3356	4291	Nine to Five (a.k.a. 9 to 5) (1980)	Comedy|Crime
7209	65585	Bride Wars (2009)	Comedy|Romance

User-Based Collaborative Filtering

The User-Based Collaborative Filtering approach groups users according to prior usage behavior or according to their preferences, and then recommends an item that a similar user in the same group viewed or liked. To put this in layman terms, if user 1 liked movie A, B and C, and if user 2 liked movie A and B, then movie C might make a good recommendation to user 2. The User-Based Collaborative Filtering approach mimics how word-of-mouth recommendations work in real life.

The User-based Collaborative Filtering recommender model was created with recommenderlab with the below parameters and the ratings matrix:

Method: UBCF

Similarity Calculation Method: Cosine Similarity

Nearest Neighbors: 30

The predicted item ratings of the user will be derived from the 5 nearest neighbors in its neighborhood. When the predicted item ratings are obtained, the top 10 most highly predicted ratings will be returned as the recommendations.

Movies recommended to user 1 after applying User based filtering:

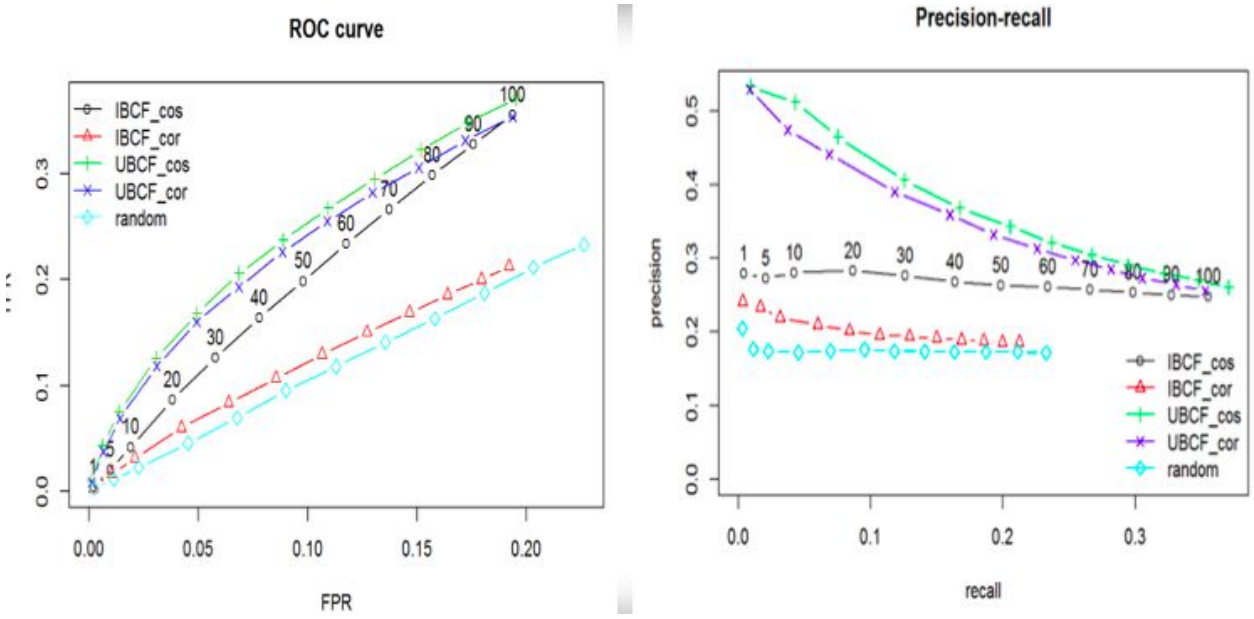
```
> recom_result
      [,1]
 [1,] "Mighty Morphin Power Rangers: The Movie (1995)"
 [2,] "Winter Guest, The (1997)"
 [3,] "Wolf (1994)"
 [4,] "Swiss Family Robinson (1960)"
 [5,] "BASEketball (1998)"
 [6,] "Player, The (1992)"
 [7,] "Fun and Fancy Free (1947)"
 [8,] "Sirens (1994)"
 [9,] "Doom Generation, The (1995)"
[10,] "Guilty as Sin (1993)"
```

Model Comparison:

We have implemented four different models:

1. Item based collaborative filtering, using cosine as the distance function
2. Item based collaborative filtering, using Pearson correlation as the distance function
3. User based collaborative filtering, using cosine as the distance function
4. User based collaborative filtering, using Pearson correlation as the distance function

We draw the ROC curves for all the models and check the Recall and Precision curve to determine the best model.



We can clearly see that User based collaborative filtering with cosine as the similarity function has the maximum area under the curve (AUC) and also the maximum precision recall value i.e. the F-score (harmonized mean) for this model is high compared to others.

6.Challenges

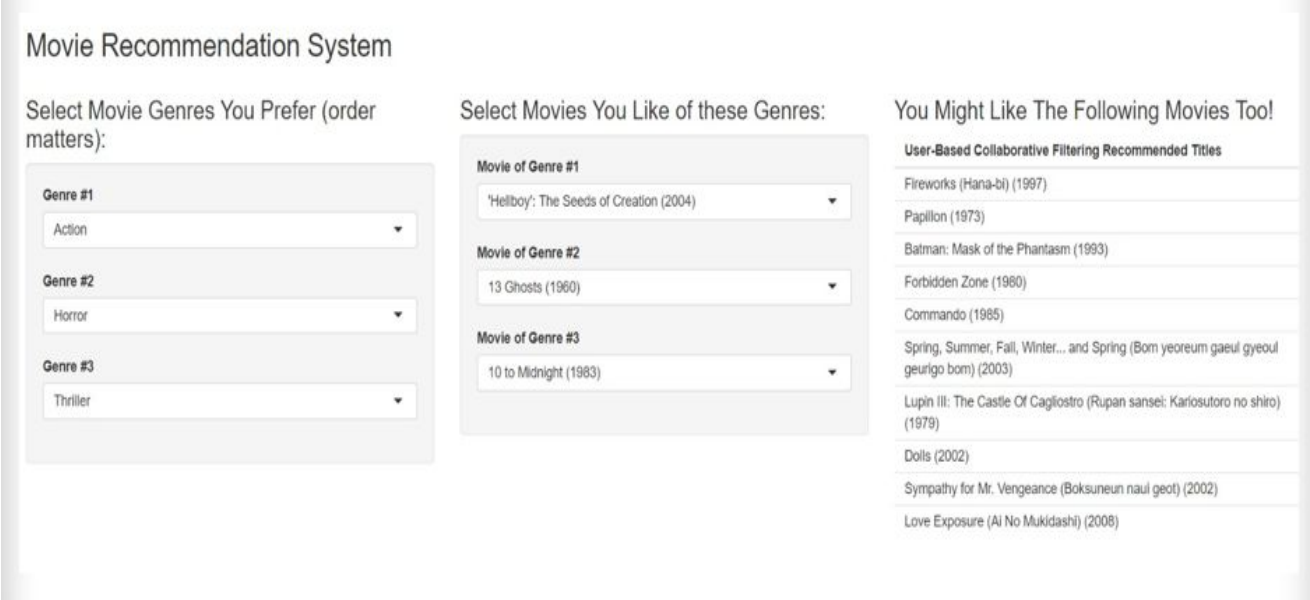
The first challenge we faced in the project is the size of the actual movie lens dataset. The size of the original dataset is 100M which is very difficult to process in R with the kind of operating system we have. Hence, we decided to work on the subset data 100K which is also provided by movie lens website. We perform our analysis and create a recommendation system on this dataset and the same analysis can be applied on the original one of 100M using an advanced computer with high processing power.

The next challenge we faced in the project is the implementation of the R-shiny application to create the user interface for our movie recommendation system. Posting on R-shiny requires a fee to be paid and hence we decided to select the free trial in which we were only allowed to give recommendations based on 1000 movies.

Logically this makes sense because User-based Collaborative Filtering gives recommendations that can be complements to the item the user was interacting with. Whereas, in item based filtering the recommendations we get will likely be direct substitutes, and not complements, of the item the user interacted with. It won't be effective to have a Item-based recommender if 80% of our movies are of same genre. Ex: if we have movies related to Comedy(same genre) in our dataset then it will recommend only comedy movies.

7.Conclusion

In this project, we have developed and evaluated a collaborative filtering recommender (CFR) system for recommending movies. An interactive app was created to demonstrate the User-based Collaborative Filtering approach for recommendation model



Here the user selected Genre- Action Movie- Hellboy,Genre- Horror Movie- 13 ghosts,Genre- Thriller Movie- 10 to Midnight and our system recommended 10 movies which are the combination of all the 3 genres the user has selected.

Item-Based Collaborative Filtering:

Strengths:

Content-based recommender systems don't require a lot of user data.
We just need item data and we are able to start giving recommendations to users.
Also, our recommendation engine does not depend on lots of user data, so it is possible to give recommendations to even our first customer as long as we have adequate data to build his user profile.

Weakness:

Our item data needs to be well distributed. It won't be effective to have a content-based recommender if 80% of our movies are of same genre.
Also, the recommendations we get will likely be direct substitutes, and not complements, of the item the user interacted with. Complements are more likely discovered through collaborative techniques.

User-Based Collaborative Filtering:

Strengths:

User-based Collaborative Filtering gives recommendations that can be complements to the item the user was interacting with.
This might be a stronger recommendation than what an item-based recommender can provide as users might not be looking for direct substitutes to a movie they had just viewed or previously watched.

Weakness:

User-based Collaborative Filtering is a type of Memory-based Collaborative Filtering that uses all user data in the database to create recommendations.
Comparing the pairwise correlation of every user in our dataset is not scalable. User-based collaborative filtering relies on past user choices to make future recommendations. The implications of this is that it assumes that a user's taste and preference remains more or less constant over time, which might not be true and makes it difficult to pre-compute user similarities offline.

Learning:

In the era of competitors like Amazon Prime, Netflix, Hulu, companies want maximum users to stick to their product which will result in higher profits. So, this recommender system plans to provide optimized recommendation of movies to increase viewership on our product. This project plans to give vital information to marketing department so that they can market a movie in order to make profits. Also, the PR team and Design team will use the information generated from this project to place movies efficiently on the website and target specific segment of customers.

In this project, we learn how companies like Amazon, Netflix suggests us products based on our purchases and how they target more customers. For ex: If I watched Avengers endgame and Thor Ragnarok on Netflix, the company will be pretty sure that I will be watching some superhero movie so they will recommend me similar type of movies and make money out of me. Understanding how these companies recommend products based on our liking and history was the biggest take away from this project.