CSE 4027-Data Analytics

Team 6

Slot: F

R PROJECT ON
MOVIE RECOMMENDATION SYSTEM

Under the esteemed guidance of

Dr.N.Senthil Murugan

Assistant Professor, VIT-AP University

Submitted by

S.V.N.Sai Varun-19BCE7092

TABLE OF CONTENTS

AIM

The main goal of this project is to build a recommendation engine that recommends movies to users by building an item based Collaborative Filter.

ABSTARCT

Now-a-days recommender systems are used widely. These recommendation systems are used to help users find what they want based on their preferences and previous interactions.Such recommendation systems are beneficial for organizations that collect data from large amounts of customers, and wish to effectively provide the best suggestions possible.

The popularity of recommendations systems have gradually increased and are recently implemented in almost all online platforms that people use. The content of such system differs from films, podcasts, books and videos, to colleagues and stories on social media, to commodities on e-commerce websites, to people on commercial websites. Often, these systems are able to retrieve and filter data about a user's preferences, and can use this intel to advance their suggestions in the upcoming period.

INTRODUCTION

A recommendation system is a type of information filtering system which challenges to assume the priorities of a user, and make recommendations on the basis of user's priorities. Recommendation systems plays an important role in e-commerce and online streaming services, such as Netflix, YouTube and Amazon prime. These systems estimate the most likely product that consumers will buy or will be interested in. A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. This not only helps them show relevant products/interests to users but also allows them to stand out in front of competitors.Companies competing for customer loyalty invest on systems that capture and analyses the user's preferences, and offer products or services with higher likelihood of purchase.

There are two types of recommendation systems – Content-Based Recommendation System and Collaborative Filtering Recommendation.In this project of recommendation system ,ITEM based collaborative recommendation system is developed. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset. How IBCF works is that it suggests an item based on items the user has previously consumed. It looks for the items the user has consumed then it finds other items similar to consumed items and recommends accordingly.

ABOUT THE DATASET

There are two datasets-movies.csv and ratings.csv

```
> movies
   movieId                                                        title                                        genres
1        1                                             Toy Story (1995)   Adventure|Animation|Children|Comedy|Fantasy
2        2                                               Jumanji (1995)                    Adventure|Children|Fantasy
3        3                                      Grumpier Old Men (1995)                                Comedy|Romance
4        4                                     Waiting to Exhale (1995)                          Comedy|Drama|Romance
5        5                           Father of the Bride Part II (1995)                                        Comedy
6        6                                                  Heat (1995)                         Action|Crime|Thriller
7        7                                               Sabrina (1995)                                Comedy|Romance
8        8                                          Tom and Huck (1995)                            Adventure|Children
9        9                                          Sudden Death (1995)                                        Action
10      10                                             GoldenEye (1995)                     Action|Adventure|Thriller
11      11                               American President, The (1995)                          Comedy|Drama|Romance
12      12                           Dracula: Dead and Loving It (1995)                                 Comedy|Horror
13      13                                                 Balto (1995)                  Adventure|Animation|Children
14      14                                                 Nixon (1995)                                         Drama
15      15                                      Cutthroat Island (1995)                      Action|Adventure|Romance
16      16                                                Casino (1995)                                   Crime|Drama
17      17                                 Sense and Sensibility (1995)                                 Drama|Romance
18      18                                            Four Rooms (1995)                                        Comedy
19      19                        Ace Ventura: When Nature Calls (1995)                                        Comedy
20      20                                           Money Train (1995)            Action|Comedy|Crime|Drama|Thriller
21      21                                            Get Shorty (1995)                         Comedy|Crime|Thriller
22      22                                               Copycat (1995)           Crime|Drama|Horror|Mystery|Thriller
23      23                                             Assassins (1995)                         Action|Crime|Thriller
24      24                                                Powder (1995)                                  Drama|Sci-Fi
25      25                                     Leaving Las Vegas (1995)                                 Drama|Romance
26      26                                               Othello (1995)                                         Drama
27      27                                          Now and Then (1995)                                Children|Drama
28      28                                            Persuasion (1995)                                 Drama|Romance
29      29    City of Lost Children, The (CitÃ© des enfants perdus, La) (1995)  Adventure|Drama|Fantasy|Mystery|Sci-Fi
30      30                Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)                            Crime|Drama
31      31                                       Dangerous Minds (1995)                                         Drama
32      32                       Twelve Monkeys (a.k.a. 12 Monkeys) (1995)                        Mystery|Sci-Fi|Thriller
```

movies.csv

There are 10329 rows and 3 columns in movies.csv dataset with attributes being movieId,title,genres.

```
> ratings
   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
7       1     150    3.0 1217895940
8       1     161    4.0 1217897864
9       1     165    3.0 1217897135
10      1     204    0.5 1217895786
11      1     223    4.0 1217897795
12      1     256    0.5 1217895764
13      1     260    4.5 1217895864
14      1     261    1.5 1217895750
15      1     277    0.5 1217895772
16      1     296    4.0 1217896125
17      1     318    4.0 1217895860
18      1     349    4.5 1217897058
19      1     356    3.0 1217896231
20      1     377    2.5 1217896373
21      1     380    3.0 1217896030
22      1     457    4.0 1217896015
23      1     480    3.5 1217895972
24      1     527    4.5 1217896341
25      1     589    3.5 1217896078
26      1     590    3.5 1217896038
27      1     592    2.5 1217896043
28      1     593    5.0 1217895932
29      1     597    3.0 1217897176
30      1     608    3.5 1217896319
31      1     648    3.5 1217896397
32      1     719    0.5 1217895799
```

ratings.csv

There are 105339 rows and 4 columns in ratings.csv dataset with attributes being userId,movieId,rating,timestamp

IMPLEMENTATION

Importing Essential Libraries :

install.packages('recommenderlab')

```
> install.packages('recommenderlab')
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Cherukuri/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependencies 'float', 'RcppProgress', 'arules', 'proxy', 'registry', 'irlba', 'recosystem'


  There is a binary version available but the source version is later:
       binary source needs_compilation
arules  1.7-1  1.7-2              TRUE

  Binaries will be installed
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/float_0.2-6.zip'
Content type 'application/zip' length 3964479 bytes (3.8 MB)
downloaded 3.8 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/RcppProgress_0.4.2.zip'
Content type 'application/zip' length 34020 bytes (33 KB)
downloaded 33 KB
```

```
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/arules_1.7-1.zip'
Content type 'application/zip' length 2607106 bytes (2.5 MB)
downloaded 2.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/proxy_0.4-26.zip'
Content type 'application/zip' length 245358 bytes (239 KB)
downloaded 239 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/registry_0.5-1.zip'
Content type 'application/zip' length 197421 bytes (192 KB)
downloaded 192 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/irlba_2.3.5.zip'
Content type 'application/zip' length 314432 bytes (307 KB)
downloaded 307 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/recosystem_0.5.zip'
Content type 'application/zip' length 1321141 bytes (1.3 MB)
downloaded 1.3 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/recommenderlab_0.2-7.zip'
Content type 'application/zip' length 2067564 bytes (2.0 MB)
downloaded 2.0 MB

package 'float' successfully unpacked and MD5 sums checked
package 'RcppProgress' successfully unpacked and MD5 sums checked
package 'arules' successfully unpacked and MD5 sums checked
package 'proxy' successfully unpacked and MD5 sums checked
package 'registry' successfully unpacked and MD5 sums checked
package 'irlba' successfully unpacked and MD5 sums checked
package 'recosystem' successfully unpacked and MD5 sums checked
package 'recommenderlab' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Cherukuri\AppData\Local\Temp\Rtmps3h8x3\downloaded_packages
```

library('recommenderlab')

```
> library('recommenderlab')
Loading required package: Matrix
Loading required package: arules

Attaching package: ‘arules’

The following objects are masked from ‘package:base’:

    abbreviate, write

Loading required package: proxy

Attaching package: ‘proxy’

The following object is masked from ‘package:Matrix’:

    as.matrix

The following objects are masked from ‘package:stats’:

    as.dist, dist

The following object is masked from ‘package:base’:

    as.matrix

Loading required package: registry
Registered S3 methods overwritten by 'registry':
  method               from
  print.registry_field proxy
  print.registry_entry proxy
> 
```

install.packages('ggplot2') library(ggplot2)

install.packages('data.table')

library(data.table)

install.packages('reshape2')

library(reshape2)

```
> install.packages('ggplot2')
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the
 appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Cherukuri/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependencies 'colorspace', 'farver', 'labeling', 'munsell', 'RColorBrewer', 'viridisLite',
 'digest', 'gtable', 'isoband', 'scales'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/colorspace_2.0-2.zip'
Content type 'application/zip' length 2645307 bytes (2.5 MB)
downloaded 2.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/farver_2.1.0.zip'
Content type 'application/zip' length 1752621 bytes (1.7 MB)
downloaded 1.7 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/labeling_0.4.2.zip'
Content type 'application/zip' length 62679 bytes (61 KB)
downloaded 61 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/munsell_0.5.0.zip'
Content type 'application/zip' length 245486 bytes (239 KB)
downloaded 239 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/RColorBrewer_1.1-2.zip'
Content type 'application/zip' length 55707 bytes (54 KB)
downloaded 54 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/viridisLite_0.4.0.zip'
Content type 'application/zip' length 1299504 bytes (1.2 MB)
downloaded 1.2 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/digest_0.6.29.zip'
Content type 'application/zip' length 266591 bytes (260 KB)
downloaded 260 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/gtable_0.3.0.zip'
Content type 'application/zip' length 434327 bytes (424 KB)
downloaded 424 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/isoband_0.2.5.zip'
Content type 'application/zip' length 2726764 bytes (2.6 MB)
downloaded 2.6 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/scales_1.1.1.zip'
Content type 'application/zip' length 558895 bytes (545 KB)
downloaded 545 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/ggplot2_3.3.5.zip'
Content type 'application/zip' length 4130301 bytes (3.9 MB)
downloaded 3.9 MB
```

```
package 'colorspace' successfully unpacked and MD5 sums checked
package 'farver' successfully unpacked and MD5 sums checked
package 'labeling' successfully unpacked and MD5 sums checked
package 'munsell' successfully unpacked and MD5 sums checked
package 'RColorBrewer' successfully unpacked and MD5 sums checked
package 'viridisLite' successfully unpacked and MD5 sums checked
package 'digest' successfully unpacked and MD5 sums checked
package 'gtable' successfully unpacked and MD5 sums checked
package 'isoband' successfully unpacked and MD5 sums checked
package 'scales' successfully unpacked and MD5 sums checked
package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Cherukuri\AppData\Local\Temp\Rtmps3h8x3\downloaded_packages
> library(ggplot2)
> install.packages('data.table')
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the
 appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Cherukuri/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/data.table_1.14.2.zip'
Content type 'application/zip' length 2600846 bytes (2.5 MB)
downloaded 2.5 MB

package 'data.table' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Cherukuri\AppData\Local\Temp\Rtmps3h8x3\downloaded_packages
> library(data.table)
data.table 1.14.2 using 4 threads (see ?getDTthreads).  Latest news: r-datatable.com
> install.packages('reshape2')
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the
 appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Cherukuri/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependency 'plyr'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/plyr_1.8.6.zip'
Content type 'application/zip' length 1498474 bytes (1.4 MB)
downloaded 1.4 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/reshape2_1.4.4.zip'
Content type 'application/zip' length 817985 bytes (798 KB)
downloaded 798 KB

package 'plyr' successfully unpacked and MD5 sums checked
package 'reshape2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Cherukuri\AppData\Local\Temp\Rtmps3h8x3\downloaded_packages
```

```
> library(reshape2)

Attaching package: 'reshape2'

The following objects are masked from 'package:data.table':

    dcast, melt
```

setwd("C:/Users/Cherukuri/Desktop/movierecommendation")

movies <- read.csv("movies.csv",stringsAsFactors=FALSE)

ratings <- read.csv("ratings.csv") str(movies) str(ratings)

```
> setwd("C:/Users/Cherukuri/Desktop/movierecommendation")
> movies <- read.csv("movies.csv",stringsAsFactors=FALSE)
> ratings <- read.csv("ratings.csv")
> str(movies)
'data.frame':   10329 obs. of  3 variables:
 $ movieId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ title  : chr  "Toy Story (1995)" "Jumanji (1995)" "Grumpier Old Men (1995)" "Wai
ting to Exhale (1995)" ...
 $ genres : chr  "Adventure|Animation|Children|Comedy|Fantasy" "Adventure|Children|
Fantasy" "Comedy|Romance" "Comedy|Drama|Romance" ...
> str(ratings)
'data.frame':   105339 obs. of  4 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : int  16 24 32 47 50 110 150 161 165 204 ...
 $ rating   : num  4 1.5 4 4 4 4 3 4 3 0.5 ...
 $ timestamp: int  1217897793 1217895807 1217896246 1217896556 1217896523 121789615
0 1217895940 1217897864 1217897135 1217895786 ...
> |
```

movies

```
> movies
   movieId                                                     title                                      genres
1        1                                          Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
2        2                                            Jumanji (1995)                  Adventure|Children|Fantasy
3        3                                   Grumpier Old Men (1995)                              Comedy|Romance
4        4                                  Waiting to Exhale (1995)                        Comedy|Drama|Romance
5        5                        Father of the Bride Part II (1995)                                      Comedy
6        6                                               Heat (1995)                       Action|Crime|Thriller
7        7                                            Sabrina (1995)                              Comedy|Romance
8        8                                       Tom and Huck (1995)                          Adventure|Children
9        9                                       Sudden Death (1995)                                      Action
10      10                                          GoldenEye (1995)                   Action|Adventure|Thriller
11      11                            American President, The (1995)                        Comedy|Drama|Romance
12      12                        Dracula: Dead and Loving It (1995)                               Comedy|Horror
13      13                                              Balto (1995)                Adventure|Animation|Children
14      14                                              Nixon (1995)                                       Drama
15      15                                   Cutthroat Island (1995)                    Action|Adventure|Romance
16      16                                             Casino (1995)                                 Crime|Drama
17      17                              Sense and Sensibility (1995)                               Drama|Romance
18      18                                         Four Rooms (1995)                                      Comedy
19      19                     Ace Ventura: When Nature Calls (1995)                                      Comedy
20      20                                        Money Train (1995)            Action|Comedy|Crime|Drama|Thriller
21      21                                         Get Shorty (1995)                       Comedy|Crime|Thriller
22      22                                            Copycat (1995)            Crime|Drama|Horror|Mystery|Thriller
23      23                                          Assassins (1995)                       Action|Crime|Thriller
24      24                                             Powder (1995)                                Drama|Sci-Fi
25      25                                  Leaving Las Vegas (1995)                               Drama|Romance
26      26                                            Othello (1995)                                       Drama
27      27                                      Now and Then (1995)                              Children|Drama
28      28                                         Persuasion (1995)                               Drama|Romance
29      29 City of Lost Children, The (CitÃ© des enfants perdus, La) (1995)  Adventure|Drama|Fantasy|Mystery|Sci-Fi
30      30         Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)                                 Crime|Drama
31      31                                    Dangerous Minds (1995)                                       Drama
32      32                          Twelve Monkeys (a.k.a. 12 Monkeys) (1995)                      Mystery|Sci-Fi|Thriller
```

```
> ratings
   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
7       1     150    3.0 1217895940
8       1     161    4.0 1217897864
9       1     165    3.0 1217897135
10      1     204    0.5 1217895786
11      1     223    4.0 1217897795
12      1     256    0.5 1217895764
13      1     260    4.5 1217895864
14      1     261    1.5 1217895750
15      1     277    0.5 1217895772
16      1     296    4.0 1217896125
17      1     318    4.0 1217895860
18      1     349    4.5 1217897058
19      1     356    3.0 1217896231
20      1     377    2.5 1217896373
21      1     380    3.0 1217896030
22      1     457    4.0 1217896015
23      1     480    3.5 1217895972
24      1     527    4.5 1217896341
25      1     589    3.5 1217896078
26      1     590    3.5 1217896038
27      1     592    2.5 1217896043
28      1     593    5.0 1217895932
29      1     597    3.0 1217897176
30      1     608    3.5 1217896319
31      1     648    3.5 1217896397
32      1     719    0.5 1217895799
```

names(movies)

names(ratings)

dim(movies) dim(ratings)

```
> names(movies)
[1] "movieId" "title"   "genres"
> names(ratings)
[1] "userId"    "movieId"   "rating"    "timestamp"
> dim(movies)
[1] 10329     3
> dim(ratings)
[1] 105339      4
```

summary(movies) head(movies)

summary(ratings) head(ratings)

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

tail(movies)
tail(ratings)

```
> tail(movies)
      movieId                                title                      genres
10324  146656                         Creed (2015)                       Drama
10325  146684       Cosmic Scrat-tastrophe (2015) Animation|Children|Comedy
10326  146878          Le Grand Restaurant (1966)                      Comedy
10327  148238       A Very Murray Christmas (2015)                      Comedy
10328  148626                The Big Short (2015)                       Drama
10329  149532 Marco Polo: One Hundred Eyes (2015)          (no genres listed)
> tail(ratings)
       userId movieId rating  timestamp
105334    668  141472    2.5 1442679119
105335    668  142488    4.0 1451535844
105336    668  142507    3.5 1451535889
105337    668  143385    4.0 1446388585
105338    668  144976    2.5 1448656898
105339    668  148626    4.5 1451148148
```

DATA PRE-PROCESSING:

any(is.na(movies)) any(is.na(ratings))

```
> any(is.na(movies))
[1] FALSE
> any(is.na(ratings))
[1] FALSE
```

UserId column, as well as the movieId column, consist of integers. We need to convert the genres present in the movies dataframe into a more usable format by the users. So here we will firstcreate a matrix that comprises of corresponding genres for each of the films.

```
genre1 <- as.data.frame(movies$genres, stringsAsFactors=FALSE) library(data.table)

genre2 <- as.data.frame(tstrsplit(genre1[,1], '[|]', type.convert=TRUE),

stringsAsFactors=FALSE)  colnames(genre2) <- c(1:10) g_list<-

c("Action","Adventure","Animation","Children","Comedy","Crime","Documentary","Dra
ma","Fantasy","Film-

Noir","Horror","Musical","Mystery","Romance","SciFi","Thriller","War","

Western") g_matrix1 <- matrix(0,10330,18) g_matrix1[1,] <- g_list

colnames(g_matrix1) <- g_list for (i in 1:nrow(genre2)) {   for (j in

1:ncol(genre2)) {

gen_col = which(g_matrix1[1,] == genre2[i,j])

g_matrix1[i+1,gen_col] <- 1

  }

}

g_matrix2 <- as.data.frame(g_matrix1[-1,], stringsAsFactors=FALSE)  for

(j in 1:ncol(g_matrix2)) {

 g_matrix2[,j] <- as.integer(g_matrix2[,j])

}

str(g_matrix2)
```

```
> library(data.table)
> genre1 <- as.data.frame(movies$genres, stringsAsFactors=FALSE)
> library(data.table)
> genre2 <- as.data.frame(tstrsplit(genre1[,1], '[|]', type.convert=TRUE), stringsAsFactor
s=FALSE)
> colnames(genre2) <- c(1:10)
> g_list <- c("Action","Adventure","Animation","Children","Comedy","Crime","Documentar
y","Drama","Fantasy","Film-Noir","Horror","Musical","Mystery","Romance","Sci-Fi","Thrille
r","War","Western")
> g_matrix1 <- matrix(0,10330,18)
> g_matrix1[1,] <- g_list
> colnames(g_matrix1) <- g_list
> for (i in 1:nrow(genre2)) {
+   for (j in 1:ncol(genre2)) {
+     gen_col = which(g_matrix1[1,] == genre2[i,j])
+     g_matrix1[i+1,gen_col] <- 1
+   }
+ }
> g_matrix2 <- as.data.frame(g_matrix1[-1,], stringsAsFactors=FALSE)
> for (j in 1:ncol(g_matrix2)) {
+   g_matrix2[,j] <- as.integer(g_matrix2[,j])
+ }
> str(g_matrix2)
'data.frame':   10329 obs. of  18 variables:
 $ Action     : int  0 0 0 0 0 1 0 0 1 1 ...
 $ Adventure  : int  1 1 0 0 0 0 0 1 0 1 ...
 $ Animation  : int  1 0 0 0 0 0 0 0 0 0 ...
 $ Children   : int  1 1 0 0 0 0 0 1 0 0 ...
 $ Comedy     : int  1 0 1 1 1 0 1 0 0 0 ...
 $ Crime      : int  0 0 0 0 0 1 0 0 0 0 ...
 $ Documentary: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Drama      : int  0 0 0 1 0 0 0 0 0 0 ...
 $ Fantasy    : int  1 1 0 0 0 0 0 0 0 0 ...
 $ Film-Noir  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Horror     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Musical    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Mystery    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Romance    : int  0 0 1 1 0 0 1 0 0 0 ...
 $ Sci-Fi     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Thriller   : int  0 0 0 0 0 1 0 0 0 1 ...
 $ War        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Western    : int  0 0 0 0 0 0 0 0 0 0 ...
> 
```

We will create a ‘search matrix’ that will allow us to perform an easy search of the films by specifying the genre present in our list.

SearchMat<- cbind(movies[,1:2],g_matrix2[])

head(SearchMat)

```
> SearchMat <- cbind(movies[,1:2],g_matrix2[])
> head(SearchMat)
  movieId                              title Action Adventure Animation Children Comedy Crime Documentary Drama Fantasy Film-Noir Horror Musical
1       1                   Toy Story (1995)      0         1         1        1      1     0           0     0       1         0      0       0
2       2                     Jumanji (1995)      0         1         0        1      0     0           0     0       1         0      0       0
3       3            Grumpier Old Men (1995)      0         0         0        0      1     0           0     0       0         0      0       0
4       4           Waiting to Exhale (1995)      0         0         0        0      1     0           0     1       0         0      0       0
5       5 Father of the Bride Part II (1995)      0         0         0        0      1     0           0     0       0         0      0       0
6       6                        Heat (1995)      1         0         0        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
> 
```

There are movies that have several genres.This applies to the majority of the films.

We have to convert our matrix into a sparse matrix one. This new matrix is of the class ‘realRatingMatrix’.

ratings_mat<- dcast(ratings,userId~movieId,value.var="rating",na.rm=FALSE)

ratings_mat<- as.matrix(ratings_mat[,-1])

ratings_mat<- as(ratings_mat,"realRatingMatrix")

ratings_mat

```
> ratings_mat <- dcast(ratings,userId~movieId,value.var="rating",na.rm=FALSE)
> ratings_mat <- as.matrix(ratings_mat[,-1])
> ratings_mat <- as(ratings_mat,"realRatingMatrix")
> ratings_mat
668 x 10325 rating matrix of class ‘realRatingMatrix’ with 105339 ratings.
> 
```

Let us now overview some of the important parameters that provide us various options for building recommendation systems for movies

recommend <- recommenderRegistry$get_entries(dataType="realRatingMatrix")

names(recommend)

```
> recommend <- recommenderRegistry$get_entries(dataType="realRatingMatrix")
> names(recommend)
 [1] "HYBRID_realRatingMatrix"      "ALS_realRatingMatrix"      "ALS_implicit_realRatingMatrix" "IBCF_realRatingMatrix"
 [5] "LIBMF_realRatingMatrix"       "POPULAR_realRatingMatrix"  "RANDOM_realRatingMatrix"       "RERECOMMEND_realRatingMatrix"
 [9] "SVD_realRatingMatrix"         "SVDF_realRatingMatrix"     "UBCF_realRatingMatrix"
```

lapply(recommend,"[[","description")

```
> lapply(recommend,"[[","description")
$HYBRID_realRatingMatrix
[1] "Hybrid recommender that aggegates several recommendation strategies using weighted averages."

$ALS_realRatingMatrix
[1] "Recommender for explicit ratings based on latent factors, calculated by alternating least squares algorithm."

$ALS_implicit_realRatingMatrix
[1] "Recommender for implicit data based on latent factors, calculated by alternating least squares algorithm."

$IBCF_realRatingMatrix
[1] "Recommender based on item-based collaborative filtering."

$LIBMF_realRatingMatrix
[1] "Matrix factorization with LIBMF via package recosystem (https://cran.r-project.org/web/packages/recosystem/vignettes/introduction.html)."

$POPULAR_realRatingMatrix
[1] "Recommender based on item popularity."

$RANDOM_realRatingMatrix
[1] "Produce random recommendations (real ratings)."

$RERECOMMEND_realRatingMatrix
[1] "Re-recommends highly rated items (real ratings)."

$SVD_realRatingMatrix
[1] "Recommender based on SVD approximation with column-mean imputation."

$SVDF_realRatingMatrix
[1] "Recommender based on Funk SVD with gradient descend (https://sifter.org/~simon/journal/20061211.html)."

$UBCF_realRatingMatrix
[1] "Recommender based on user-based collaborative filtering."
```

recommend$IBCF_realRatingMatrix$parameters

```
> recommend$IBCF_realRatingMatrix$parameters
$k
[1] 30

$method
[1] "Cosine"

$normalize
[1] "center"

$normalize_sim_matrix
[1] FALSE

$alpha
[1] 0.5

$na_as_zero
[1] FALSE
```

EXPLORING SIMILAR DATA:

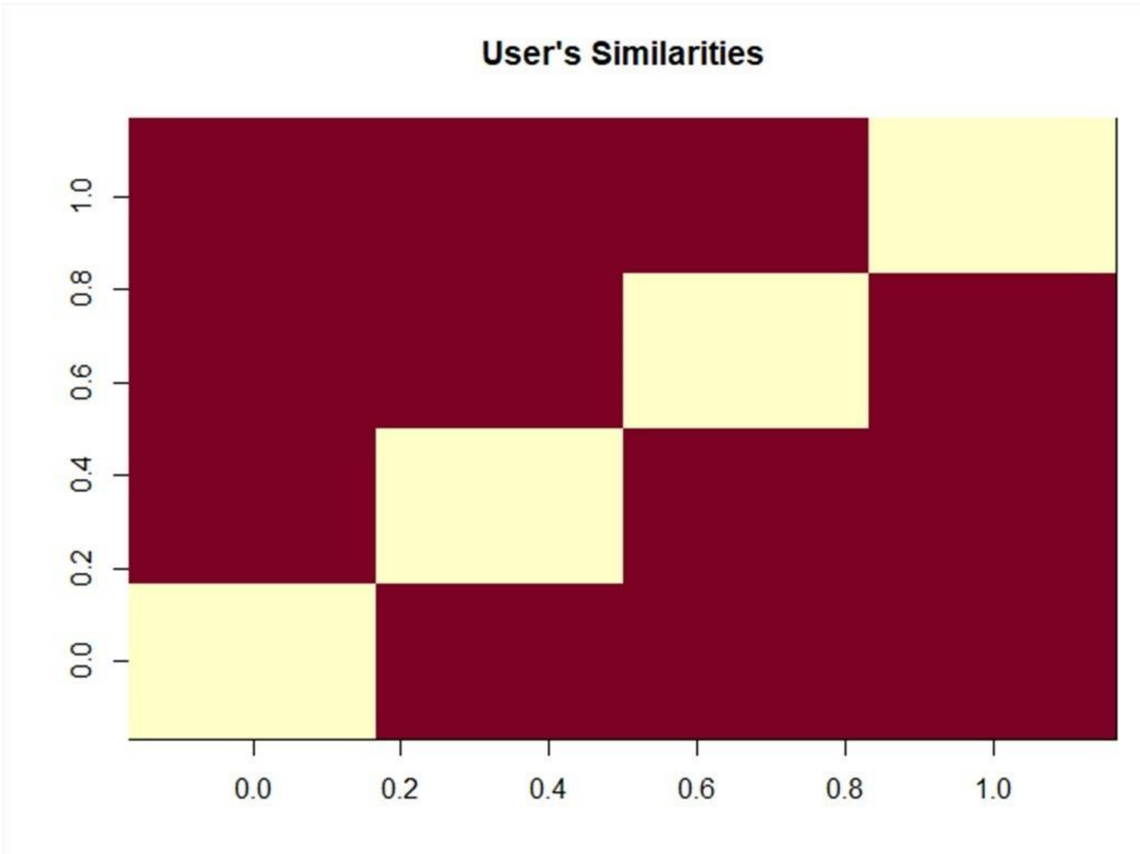
Collaborative Filtering involves suggesting movies to the users that are based on collecting preferences from many other users.Recommending movies is dependent on creating a relationship of similarity between the two users. With the help of recommenderlab, we can compute similarities using various operators like cosine, pearson as well as jaccard.

```
sim_matrix<- similarity(ratings_mat[1:4, ],method = "cosine",which = "users")

as.matrix(sim_matrix)

image(as.matrix(sim_matrix),main="User's Similarities")
```

In the above matrix, each row and column represents a user. We have taken four users and each cell in this matrix represents the similarity that is shared between the two users.

Now, we delineate the similarity that is shared between the films

```
> sim_matrix <- similarity(ratings_mat[1:4, ],method = "cosine",which = "users")
> as.matrix(sim_matrix)
          1         2         3         4
1 0.0000000 0.9760860 0.9641723 0.9914398
2 0.9760860 0.0000000 0.9925732 0.9374253
3 0.9641723 0.9925732 0.0000000 0.9888968
4 0.9914398 0.9374253 0.9888968 0.0000000
> image(as.matrix(sim_matrix),main="User's Similarities")
> |
```



movie_sim<- similarity(ratings_mat[, 1:4],method ="cosine",which = "items")

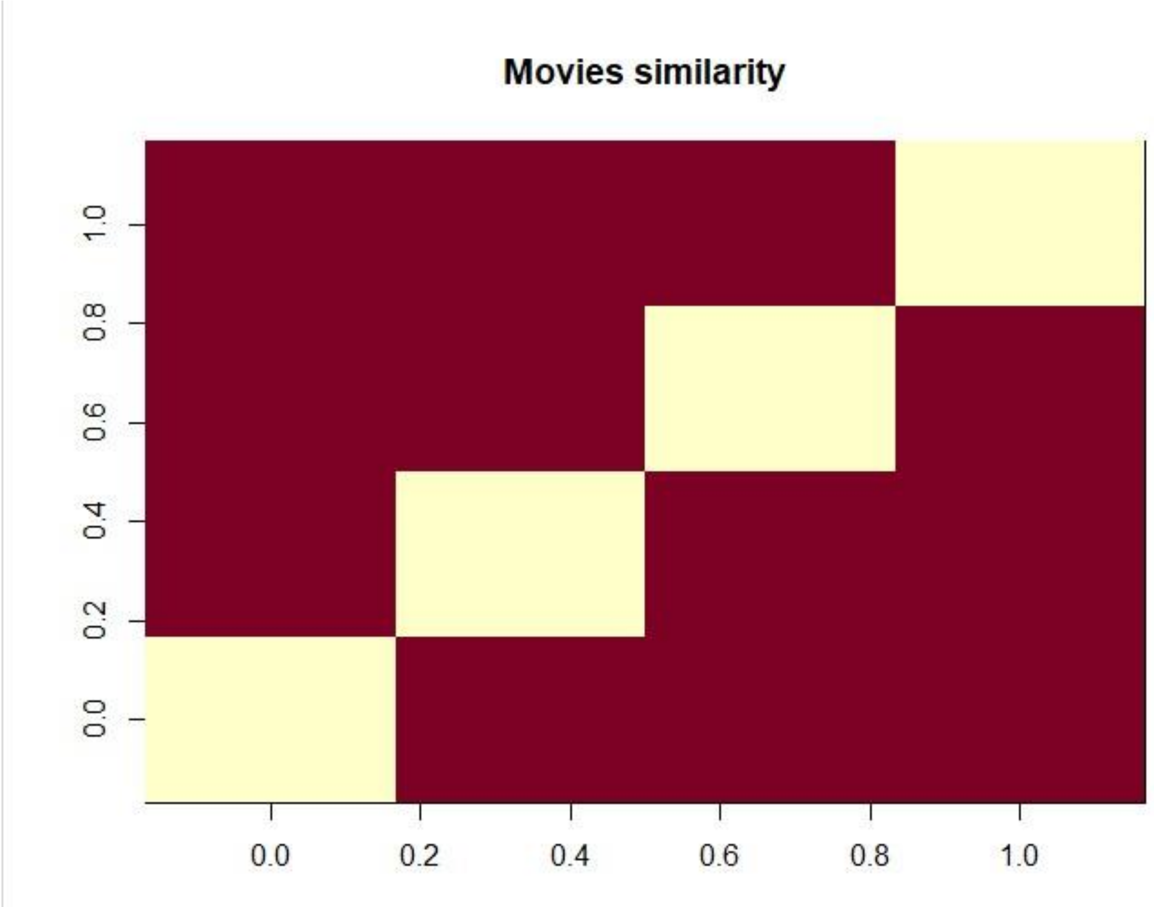
as.matrix(movie_sim)

image(as.matrix(movie_sim),main = "Movies similarity")

```
> movie_sim <- similarity(ratings_mat[, 1:4],method ="cosine",which = "items")
> as.matrix(movie_sim)
          1         2         3         4
1 0.0000000 0.9669732 0.9559341 0.9101276
2 0.9669732 0.0000000 0.9658757 0.9412416
3 0.9559341 0.9658757 0.0000000 0.9864877
4 0.9101276 0.9412416 0.9864877 0.0000000
> image(as.matrix(movie_sim),main = "Movies similarity")
> |
```

Let us now extract the most unique ratings –

Now, we will create a table of ratings that will display the most unique ratings.



rating_unique<- as.vector(ratings_mat@data) unique(rating_unique)

```
> rating_unique <- as.vector(ratings_mat@data)
> unique(rating_unique)
 [1] 0.0 5.0 4.0 3.0 4.5 1.5 2.0 3.5 1.0 2.5 0.5
> |
```

ratings_table<- table(rating_unique) ratings_table

```
> ratings_table <- table(rating_unique)
> ratings_table
rating_unique
      0     0.5       1     1.5       2     2.5       3     3.5       4     4.5       5
6791761    1198    3258    1567    7943    5484   21729   12237   28880    8187   14856
> |
```

MOST VIEWED MOVIES VISUALIZATION

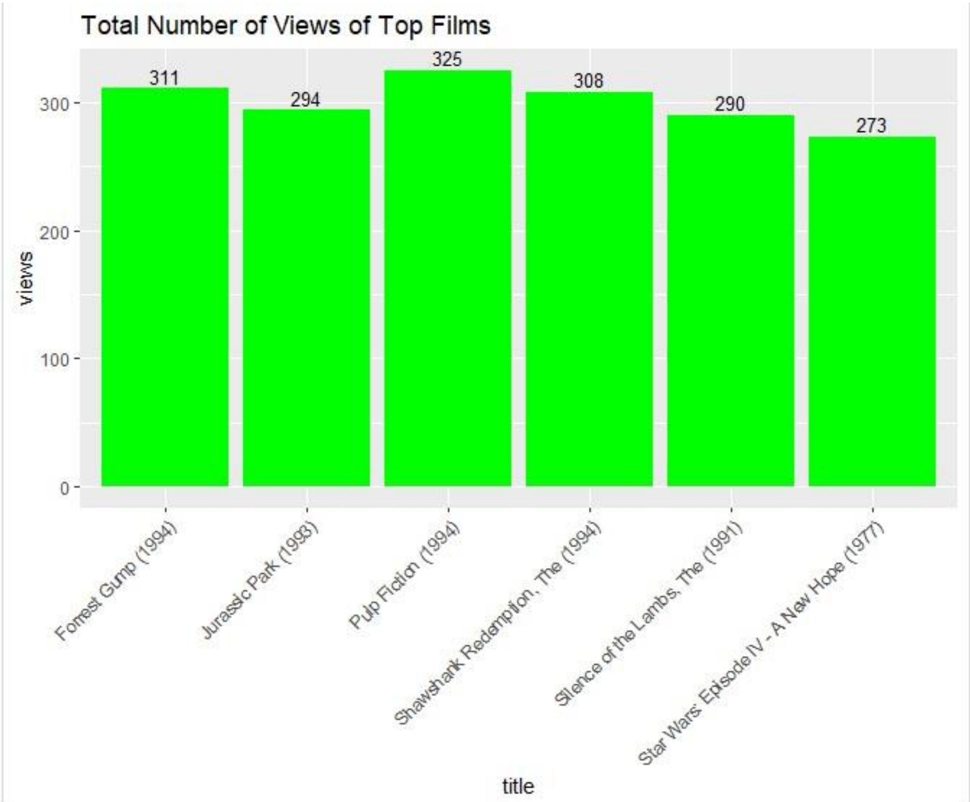
We will explore the most viewed movies in our dataset. We will first count the number of views in a film and then organize them in a table that would group them in descending order.

```
library(ggplot2) m <-
colCounts(ratings_mat)
t <- data.frame(movie=names(m),views=m)
t <- t[order(t$views,decreasing=TRUE), ]
t$title<- NA for (index in 1:10325){
  t[index,3] <- as.character(subset(movies,movies$movieId==t[index,1])$title)
} t[1:6,]
```

```
> library(ggplot2)
> m <- colCounts(ratings_mat)
> t <- data.frame(movie=names(m),views=m)
> t <- t[order(t$views,decreasing=TRUE), ]
> t$title <- NA
> for (index in 1:10325){
+   t[index,3] <- as.character(subset(movies,movies$movieId==t[index,1])$title)
+ }
> t[1:6,]
    movie views                                 title
296   296   325                   Pulp Fiction (1994)
356   356   311                   Forrest Gump (1994)
318   318   308      Shawshank Redemption, The (1994)
480   480   294                  Jurassic Park (1993)
593   593   290      Silence of the Lambs, The (1991)
260   260   273 Star Wars: Episode IV - A New Hope (1977)
>
```

Now, we will visualize a bar plot for the total number of views of the top films. We will carry this out using ggplot2.

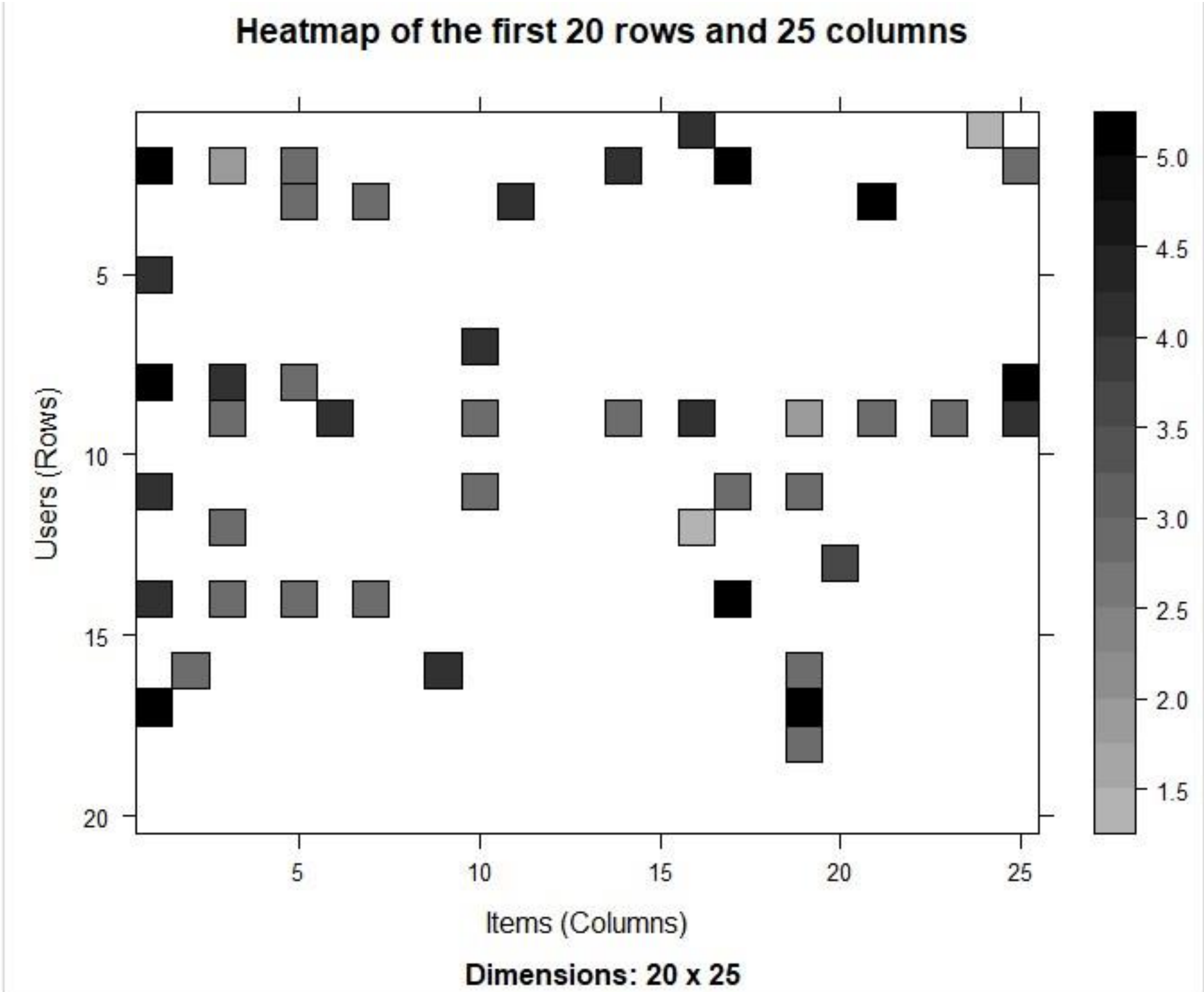
```
ggplot(t[1:6, ],aes(x=title,y=views)) +
geom_bar(stat="identity",fill='green') +
geom_text(aes(label=views),vjust=-0.3,size=3.5) +
theme(axis.text.x=element_text(angle=45,hjust = 1)) +
ggtitle("Total Number of Views of Top Films")
```



From the above bar-plot, we observe that Pulp Fiction is the most-watched film followed by Forrest Gump.

Now, We will visualize a heatmap of the movie ratings. This heatmap will contain first 20 rows and 25 columns as follows

```
image(ratings_mat[1:20, 1:25],axes=FALSE,main="Heatmap of the first 20 rows and 25
columns")
```



Selecting useful data

For finding useful data in our dataset, we have set the threshold for the minimum number of users who have rated a film as 50. This is also same for minimum number of views that are per film. This way, we have filtered a list of watched films from leastwatched ones.

r <- ratings_mat[rowCounts(ratings_mat)>50,colCounts(ratings_mat)>50]

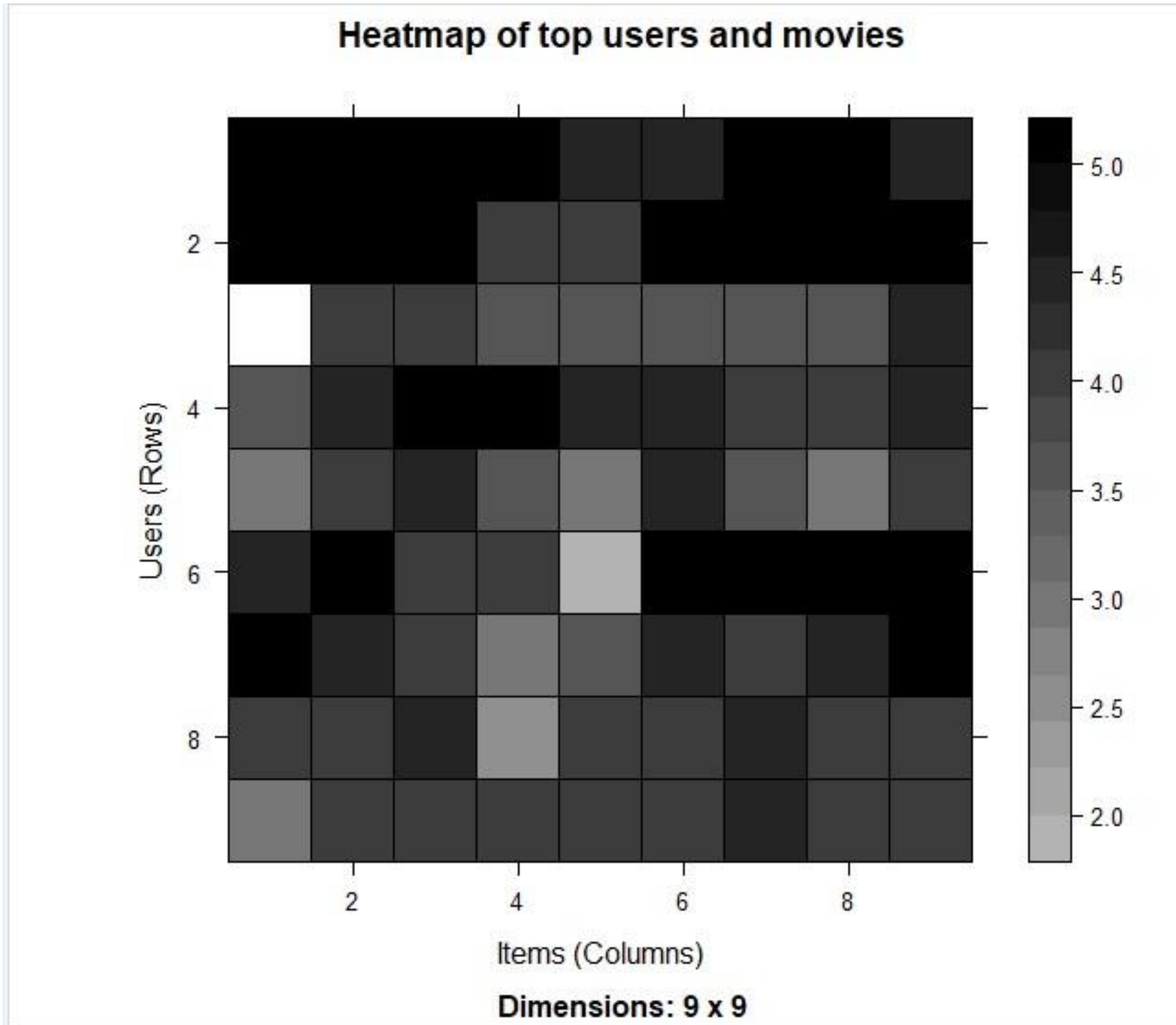
r

```
> r <- ratings_mat[rowCounts(ratings_mat)>50,colCounts(ratings_mat)>50]
> r
420 x 447 rating matrix of class 'realRatingMatrix' with 38341 ratings.
>
```

minfilms<- quantile(rowCounts(r), 0.98) minusers<-

quantile(colCounts(r), 0.98)

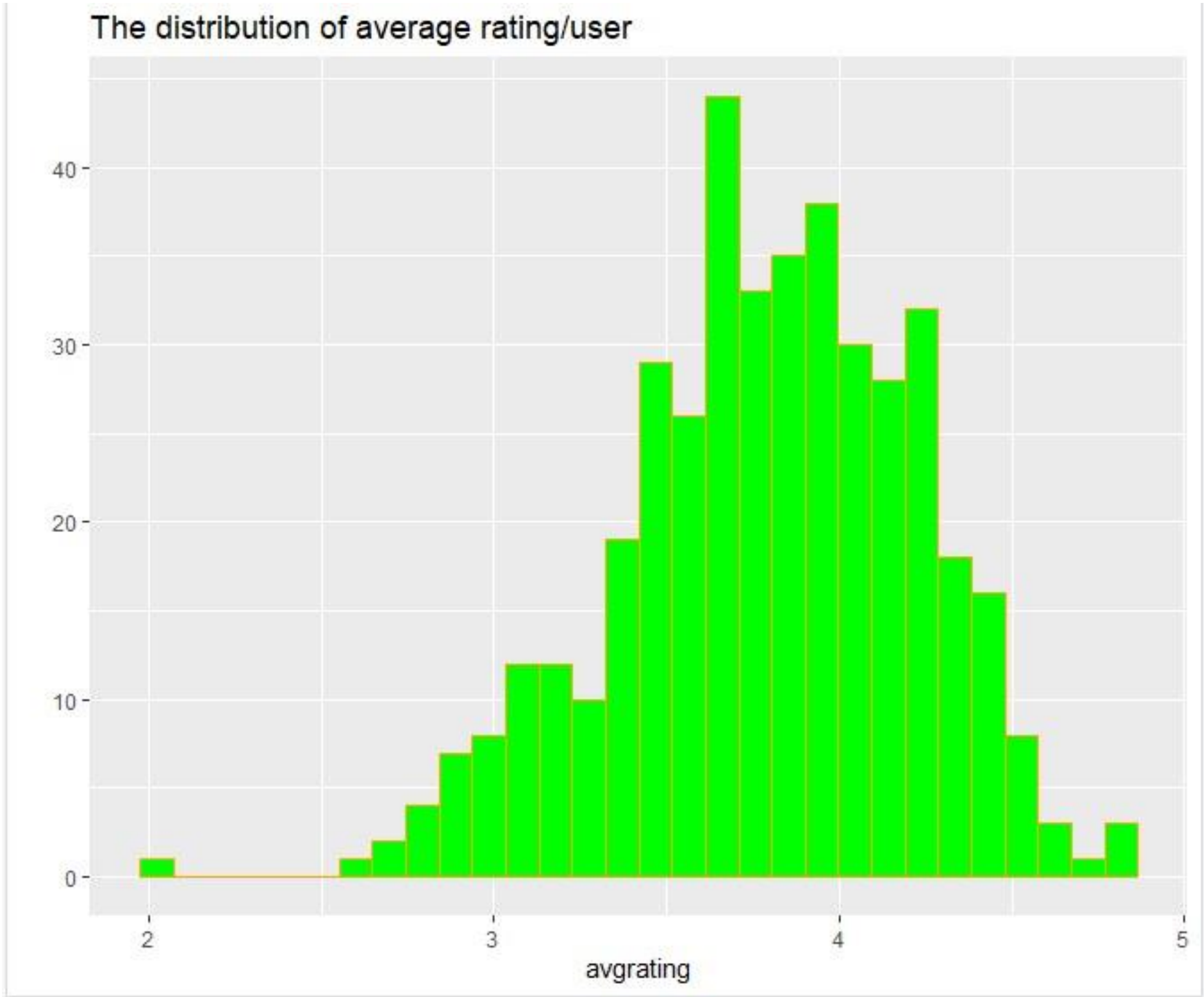
image(r[rowCounts(r)>minfilms,colCounts(r)>minusers],main="Heatmap of top users and movies")



avgrating<- rowMeans(r)

qplot(avgrating, fill=I("green"), col=I("orange")) + ggtitle("The

distribution of average rating/user")



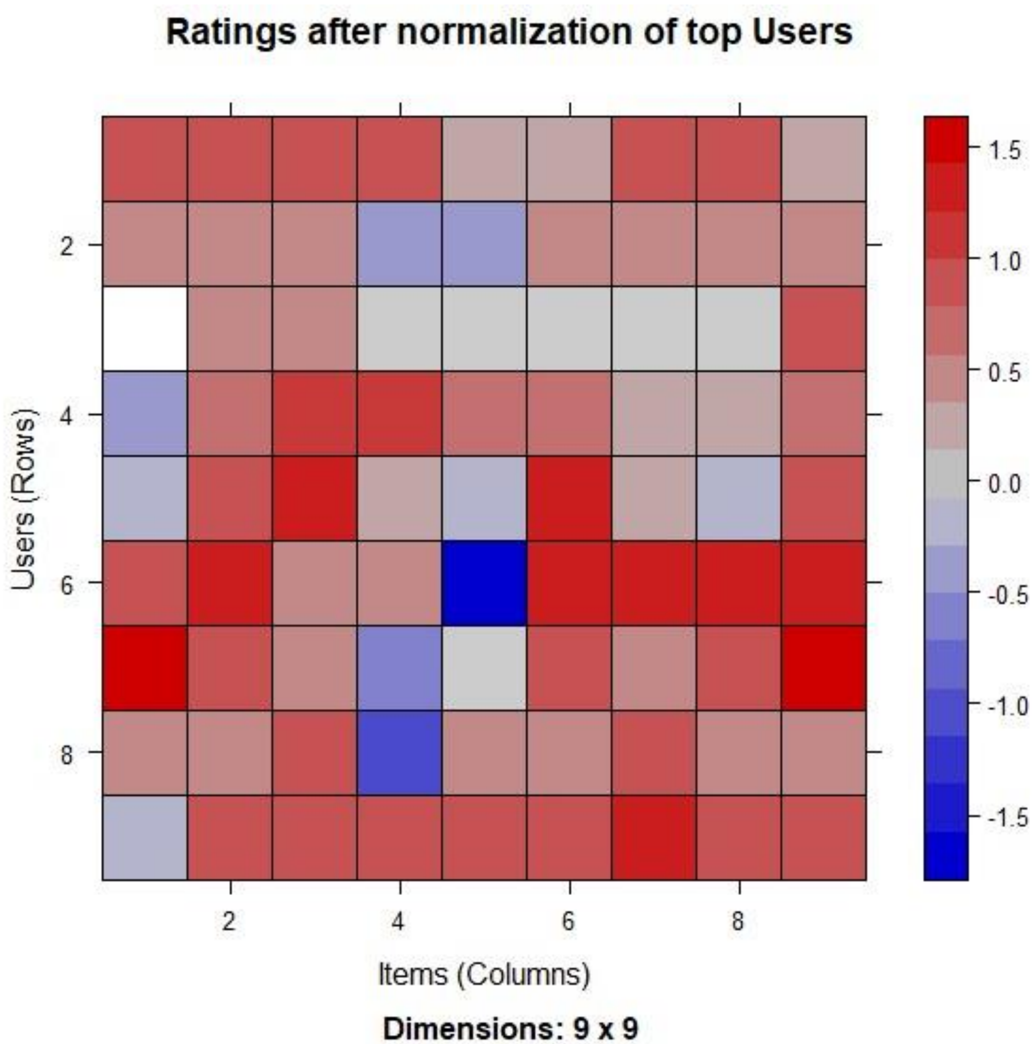
DATA NORMALIZATION :

In the case of some users, there can be high ratings or low ratings provided to all of the watched films. This will act as a bias while implementing our model. In order to remove this, we normalize our data. Normalization is a data preparation procedure to standardize the numerical values in a column to a common scale value. This is done in such a way that there is no distortion in the range of values. Normalization transforms the average value of our ratings column to 0. We then plot a heatmap that delineates our normalized ratings.


```
normr<- normalize(r) sum(rowMeans(normr)>0.00001)
```

image(normr[rowCounts(normr)>minfilms,colCounts(normr)>minusers],main="Ratings after normalization of top Users")

```
> normr <- normalize(r)
> sum(rowMeans(normr)>0.00001)
[1] 0

> image(normr[rowCounts(normr)>minfilms,colCounts(normr)>minusers],main="Ratings after normalization of top Users")
> |
```



DATA BINARIZATION :

In the final step we will binarize our data. Binarizing the data means that we have two discrete values 1 and 0, which will allow our recommendation systems to work more efficiently. We will define a matrix that will consist of 1 if the rating is above 3 and otherwise it will be 0.

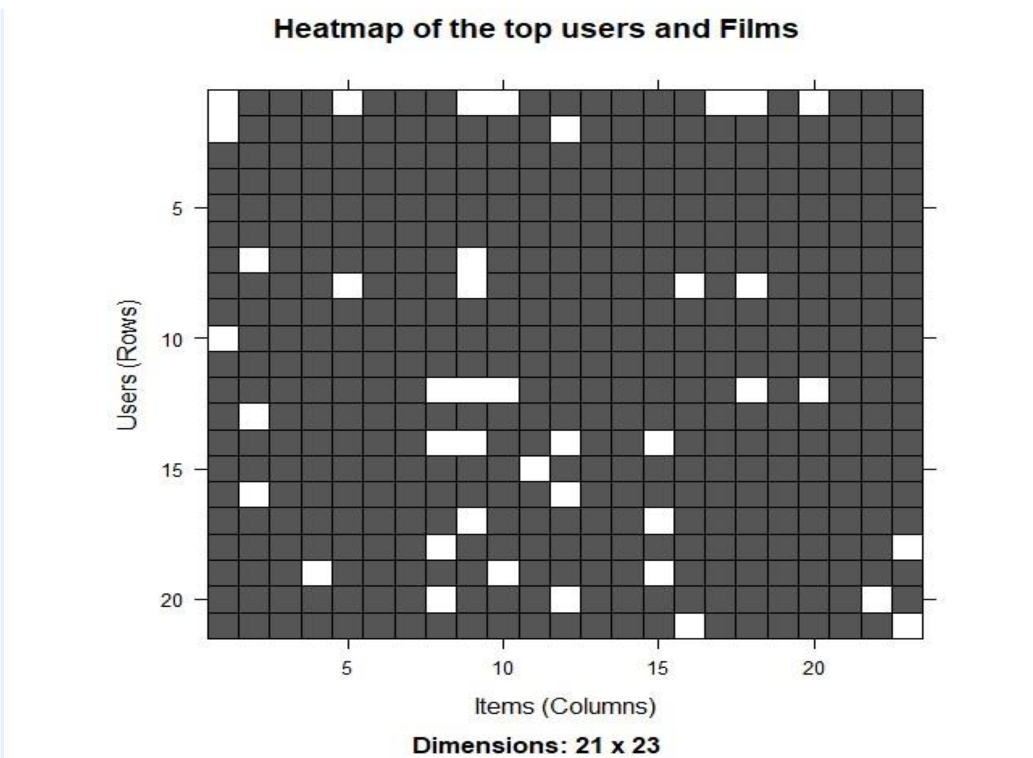
bmin_movies<- quantile(rowCounts(r),0.95) bmin_users<-

quantile(colCounts(r),0.95) toprated_films<-

binarize(r,minRating=3)

image(toprated_films[rowCounts(r)>bmin_movies,colCounts(r)>bmin_users],main="Heatmap of the top users and Films")

```
> image(normr[rowCounts(normr)>minfilms,colCounts(normr)>minusers],main="Ratings after normalization of top Users")
> bmin_movies <- quantile(rowCounts(r),0.95)
> bmin_users <- quantile(colCounts(r),0.95)
> toprated_films <- binarize(r,minRating=3)
> image(toprated_films[rowCounts(r)>bmin_movies,colCounts(r)>bmin_users],main="Heatmap of the top users and Films")
> |
```



COLLABORATIVE FILTERING SYSTEM :

Movie Recommendation System

We will develop our very own Item Based Collaborative Filtering System. This type of collaborative filtering finds similarity in the items based on the people's ratings of them. The algorithm first builds a similar-items table of the customers who have purchased them into a combination of similar items. This is then fed into the recommendation system.

We will build this filtering system by splitting the dataset into 80% training set and 20% test set.

sample_data<-sample(x=c(TRUE,FALSE),size=nrow(r),replace=TRUE,prob=c(0.8, 0.2))

train <- r[sample_data,] test <- r[!sample_data,]

```
> sample_data<-sample(x=c(TRUE,FALSE),size=nrow(r),replace=TRUE,prob=c(0.8, 0.2))
> train <- r[sample_data, ]
> test <- r[!sample_data, ]
> 
```

BUILDING THE RECOMMENDATION SYSTEM :

We will now explore the various parameters of our Item Based Collaborative Filter. These parameters are default in nature. In the first step, k denotes the number of items for computing their similarities. Here, k is equal to 30. Therefore, the algorithm will now identify the k most similar items and store their number. We use the cosine method which is the default one but you can also use pearson method.

recom<- recommenderRegistry$get_entries(dataType ="realRatingMatrix")

recom$IBCF_realRatingMatrix$ recom_system$IBCF_realRatingMatrix$parameters

```
> recom<- recommenderRegistry$get_entries(dataType ="realRatingMatrix")
> recom$IBCF_realRatingMatrix$parameters
$k
[1] 30

$method
[1] "Cosine"

$normalize
[1] "center"

$normalize_sim_matrix
[1] FALSE

$alpha
[1] 0.5

$na_as_zero
[1] FALSE

> |
```

rec_model<- Recommender(data=train,method="IBCF",parameter=list(k = 30))

rec_model

class(rec_model)

```
> rec_model <- Recommender(data=train,method="IBCF",parameter=list(k = 30))
> rec_model
Recommender of type 'IBCF' for 'realRatingMatrix'
learned using 342 users.
> class(rec_model)
[1] "Recommender"
attr(,"package")
[1] "recommenderlab"
> |
```

Using the getModel() function, we will retrieve the recommen_model. We will then find the class and dimensions of our similarity matrix that is contained within model_info. Finally, we will generate a heatmap, that will contain the top 20 items and visualize the similarity shared between them.

get_info_of_model<- getModel(rec_model)

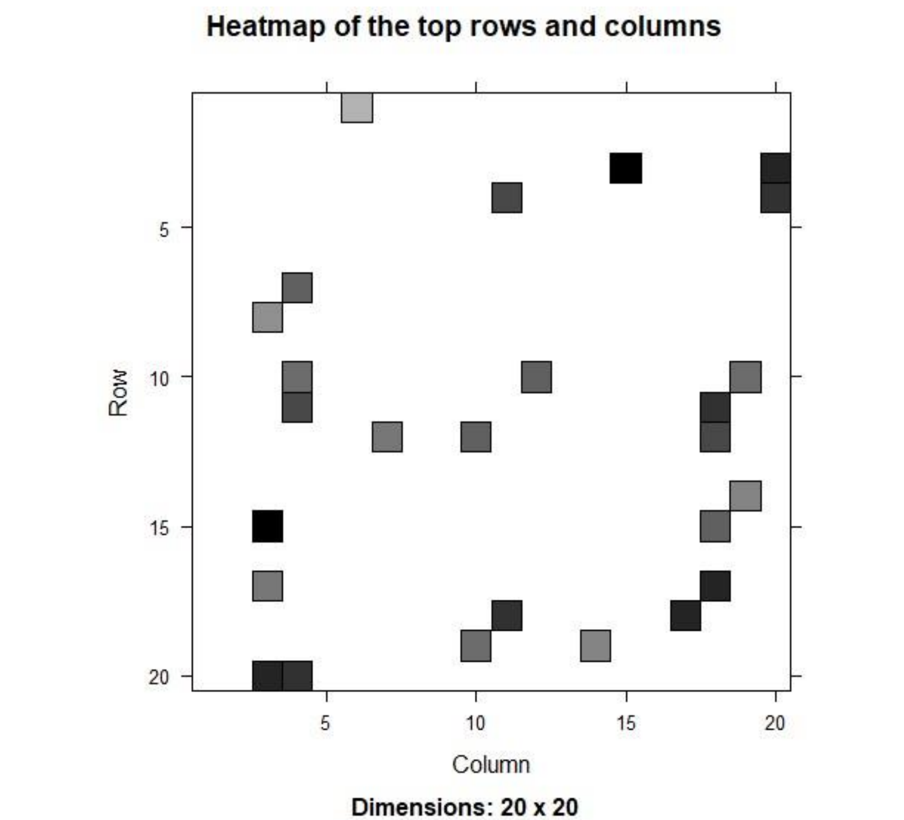
class(get_info_of_model$sim)

dim(get_info_of_model$sim) s <- 20

image(get_info_of_model$sim[1:s,1:s],mai

n="Heatmap of the top rows and columns")

```
> get_info_of_model <- getModel(rec_model)
> class(get_info_of_model$sim)
[1] "dgCMatrix"
attr(,"package")
[1] "Matrix"
> dim(get_info_of_model$sim)
[1] 447 447
> s <- 20
> image(get_info_of_model$sim[1:s,1:s],main="Heatmap of the top rows and columns")
> |
```



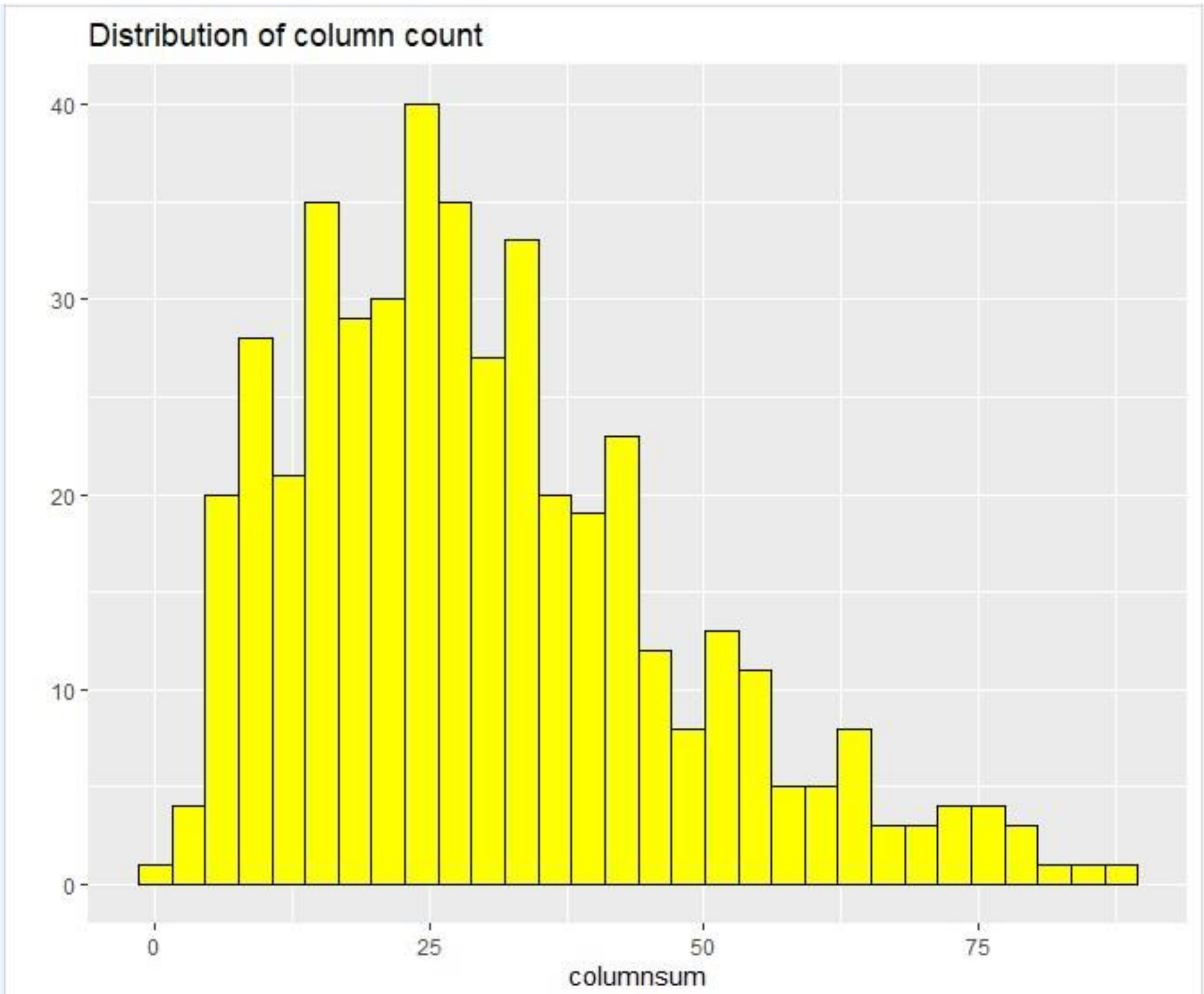
We will carry out the sum of rows and columns with the similarity of the objects above 0. We will visualize the sum of columns through a distribution as follows

rtotal<- rowSums(get_info_of_model$sim>0) table(rtotal)

columnsum<- colSums(get_info_of_model$sim>0)

qplot(columnsum, fill=I("yellow"), col=I("black"))+ ggtitle("Distribution of column count")

```
> rtotal <- rowSums(get_info_of_model$sim>0)
> table(rtotal)
rtotal
 30
447
> columnsum <- colSums(get_info_of_model$sim>0)
> qplot(columnsum, fill=I("yellow"), col=I("black"))+ ggtitle("Distribution of column count")
```



To build Recommender System:

We will create a top_recommendations variable which will be initialized to 10, specifying the number of films to each user. We will then use the predict() function that will identify similar items and will rank them appropriately. Here, each rating is used as a weight. Each weight is multiplied with related similarities. Finally, everything is added in the end.

x <- 10

predicted_recommendations<- predict(object = rec_model,newdata = test,n = x)

predicted_recommendations

```
> x <- 10
> predicted_recommendations <- predict(object = rec_model,newdata = test,n = x)
> predicted_recommendations
Recommendations as 'topNList' with n = 10 for 78 users.
> |
```

u1 <- predicted_recommendations@items[[1]] # recommendation for the first user mu1

<- predicted_recommendations@itemLabels[u1] mu2 <- mu1

for (i in 1:10){

 mu2[i] <- as.character(subset(movies,movies$movieId == mu1[i])$title)

}
mu2

```
> u1 <- predicted_recommendations@items[[1]] # recommendation for the first user
> mu1 <- predicted_recommendations@itemLabels[u1]
> mu2 <- mu1
> for (i in 1:10){
+   mu2[i] <- as.character(subset(movies,movies$movieId == mu1[i])$title)
+ }
> mu2
 [1] "Sabrina (1995)"                                   "Taxi Driver (1976)"
 [3] "Interview with the Vampire: The Vampire Chronicles (1994)"  "Like Water for Chocolate (Como agua para chocolate) (1992)"
 [5] "Much Ado About Nothing (1993)"                    "Schindler's List (1993)"
 [7] "Searching for Bobby Fischer (1993)"               "Nightmare Before Christmas, The (1993)"
 [9] "Wallace & Gromit: A Close Shave (1995)"           "North by Northwest (1959)"
> |
```

recomm_matrix<- sapply(predicted_recommendations@items,function(x){
as.integer(colnames(r)[x]) }) # matrix with the recommendations for each user
recomm_matrix[,1:4]

```
> recomm_matrix <- sapply(predicted_recommendations@items,function(x){ as.integer(colnames(r)[x]) }) # matrix with the recommendations for each user
> recomm_matrix[,1:4]
      [,1] [,2] [,3]  [,4]
 [1,]    7 2997    1  1204
 [2,]  111 2019   62  3147
 [3,]  253 7438  145  1394
 [4,]  265  594  260 60069
 [5,]  497  904  261   858
 [6,]  527  953  661  2858
 [7,]  529 1036  783  8360
 [8,]  551 1210  852   265
 [9,]  745 1214 1047  1258
[10,]  908 1220 1302  5418
```

```
no_of_items<- factor(table(recomm_matrix))
```

chart_title<- "Distribution of the Number of Items for Item based Collaborative filtering"

qplot(no_of_items, fill=I("violet"), col=I("black")) + ggtitle(chart_title) sorted_items<-

sort(no_of_items, decreasing = TRUE) n <- head(sorted_items, n = 4) df <-

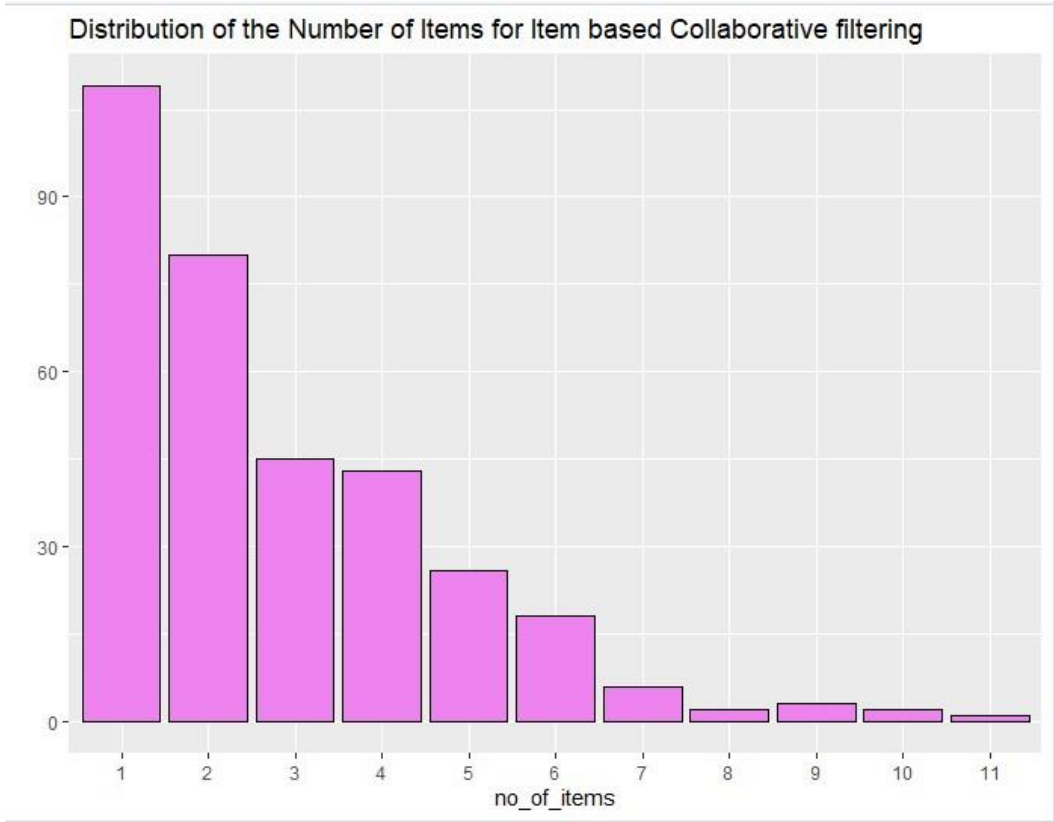
data.frame(as.integer(names(n)),n) for(i in 1:4) {

 df[i,1] <- as.character(subset(movies,movies$movieId == df[i,1])$title)

}

colnames(df) <- c("Movie Title", "No. of Items") head(df)

```
> no_of_items <- factor(table(recomm_matrix))
> chart_title <- "Distribution of the Number of Items for Item based Collaborati
ve filtering"
> qplot(no_of_items, fill=I("violet"), col=I("black")) + ggtitle(chart_title)
> sorted_items <- sort(no_of_items, decreasing = TRUE)
> n <- head(sorted_items, n = 4)
> df <- data.frame(as.integer(names(n)),n)
> for(i in 1:4) {
+   df[i,1] <- as.character(subset(movies,movies$movieId == df[i,1])$title)
+ }
> colnames(df) <- c("Movie Title", "No. of Items")
> head(df)
                         Movie Title No. of Items
529 Searching for Bobby Fischer (1993)           11
508                Philadelphia (1993)           10
919           Wizard of Oz, The (1939)           10
111                 Taxi Driver (1976)            9
> |
```


Distribution of the Number of Items for Item based Collaborative filtering
90
60
30
0
1
2
3
4
5
6
7
8
9
10
11
no_of_items

RESULTS:

Recommendations for user:

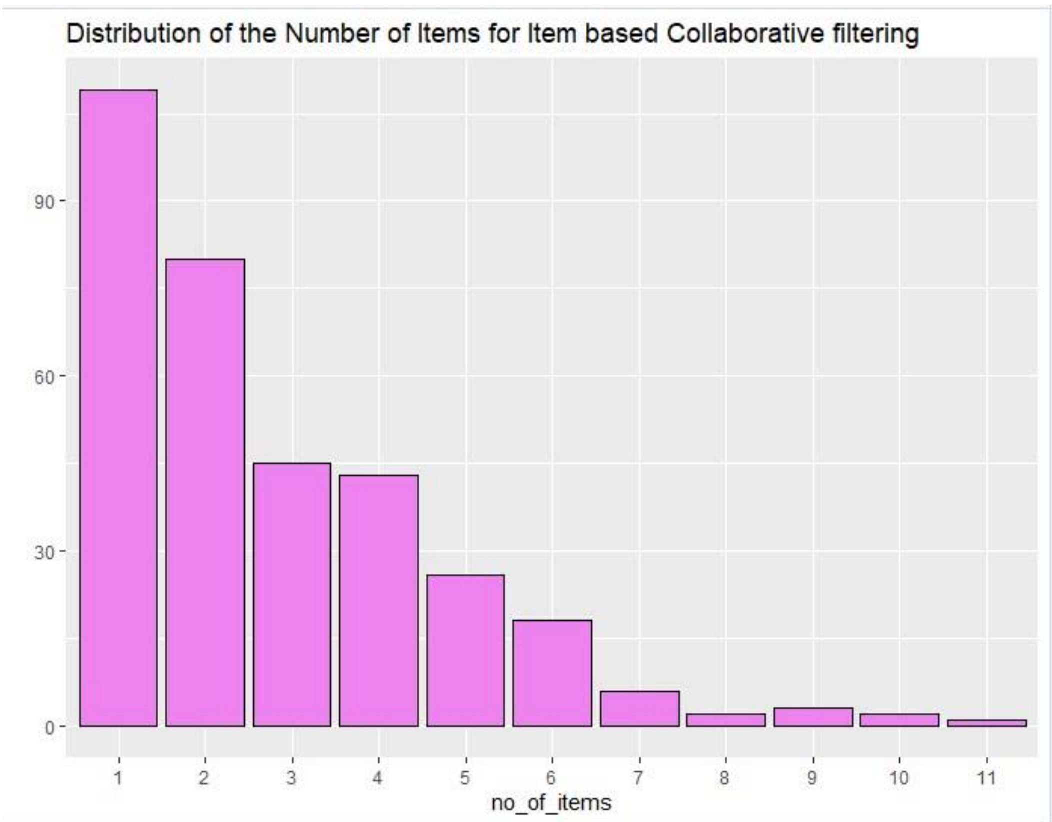
```
> u1 <- predicted_recommendations@items[[1]] # recommendation for the first user
> mu1 <- predicted_recommendations@itemLabels[u1]
> mu2 <- mu1
> for (i in 1:10){
+   mu2[i] <- as.character(subset(movies,movies$movieId == mu1[i])$title)
+ }
> mu2
 [1] "Sabrina (1995)"                                     "Taxi Driver (1976)"
 [3] "Interview with the Vampire: The Vampire Chronicles (1994)"  "Like Water for Chocolate (Como agua para chocolate) (1992)"
 [5] "Much Ado About Nothing (1993)"                      "Schindler's List (1993)"
 [7] "Searching for Bobby Fischer (1993)"                 "Nightmare Before Christmas, The (1993)"
 [9] "Wallace & Gromit: A Close Shave (1995)"             "North by Northwest (1959)"
>
```

Matrix with 10 Recommendations for each user

```
> recomm_matrix <- sapply(predicted_recommendations@items,function(x){ as.integer(colnames(r)[x]) }) # matrix with the recommendations for each user
> recomm_matrix[,1:4]
      [,1] [,2] [,3]  [,4]
 [1,]    7 2997    1  1204
 [2,]  111 2019   62  3147
 [3,]  253 7438  145  1394
 [4,]  265  594  260 60069
 [5,]  497  904  261   858
 [6,]  527  953  661  2858
 [7,]  529 1036  783  8360
 [8,]  551 1210  852   265
 [9,]  745 1214 1047  1258
[10,]  908 1220 1302  5418
```

Distribution of number of users for item based collaborative filtering

```
> no_of_items <- factor(table(recomm_matrix))
> chart_title <- "Distribution of the Number of Items for Item based Collaborati
ve filtering"
> qplot(no_of_items, fill=I("violet"), col=I("black")) + ggtitle(chart_title)
> sorted_items <- sort(no_of_items, decreasing = TRUE)
> n <- head(sorted_items, n = 4)
> df <- data.frame(as.integer(names(n)),n)
> for(i in 1:4) {
+   df[i,1] <- as.character(subset(movies,movies$movieId == df[i,1])$title)
+ }
> colnames(df) <- c("Movie Title", "No. of Items")
> head(df)
                          Movie Title No. of Items
529 Searching for Bobby Fischer (1993)          11
508                Philadelphia (1993)          10
919           Wizard of Oz, The (1939)          10
111                Taxi Driver (1976)            9
> 
```



Movie Recommendation System

CONCLUSION:

Now a days the recommender systems are used in a variety of areas including music, movies, books, news, search queries, and commercial products .These Movie Recommender systems are a powerful tools for extracting additional value for a business from its user databases. This benefit users by enabling them to find movies they like. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web.

New technologies are needed that can dramatically improve the scalability of recommender systems.These kind of movie recommender systems should be given utmost importance for increasing the amount of customers utilizing the services.