FinalDieudonnefIX1

Dieudonne

July 19, 2016

Introduction

This is the final project of my course Data 643 at CUNY The goal is to explore recommender system in some context as time ,as location ,gender. For this project I am 3 data set all made available by grouplense.

For MovieLense and MovieLenseMeta ,'recommenderlab' provides the data the third data could be dowloaded here https://github.com/dieudo/643Summer2016/blob/master/unifiedMLDataMulti.csv

Initially ,I planed to work and compare packages avalaible on recommendations systems ,but due to time cnstraint I am going to readjust my goal. This project will be sectioned in 3 part,the first part is comparing and building algorithms around MovieLense data and getting to know the performances associated. The second part is exploring the users that rated the movies ,can we classified them and learn something related to their age ,their occupations? The third part will be to implement a contextual time value associated to the year

```
#DATA & libraries
library(plyr)
library(RColorBrewer)
library(grid)
library("recommenderlab")
library(ggplot2)
data_package <- data(package = "recommenderlab")
data_package$results[, "Item"]
```

```
## [1] "Jester5k"                    "JesterJokes (Jester5k)"
## [3] "MSWeb"                       "MovieLense"
## [5] "MovieLenseMeta (MovieLense)"
```

```
data(MovieLense)
str(MovieLense)
```

```
## Formal class 'realRatingMatrix' [package "recommenderlab"] with 2 slots
##   ..@ data     :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. ..@ i       : int [1:99392] 0 1 4 5 9 12 14 15 16 17 ...
##   .. .. ..@ p       : int [1:1665] 0 452 583 673 882 968 994 1386 1605 1904 ...
##   .. .. ..@ Dim     : int [1:2] 943 1664
##   .. .. ..@ Dimnames:List of 2
##   .. .. .. ..$ : chr [1:943] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:1664] "Toy Story (1995)" "GoldenEye (1995)" "Four Rooms (1995)" "Get Shorty
##   .. .. ..@ x       : num [1:99392] 5 4 4 4 4 3 1 5 4 5 ...
##   .. .. ..@ factors : list()
##   ..@ normalize: NULL
```

```r
str(MovieLenseMeta)
```

```
## 'data.frame':    1664 obs. of  22 variables:
##  $ title      : chr  "Toy Story (1995)" "GoldenEye (1995)" "Four Rooms (1995)" "Get Shorty (1995)" .
##  $ year       : num  1995 1995 1995 1995 1995 ...
##  $ url        : chr  "http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)" "http://us.imdb.com/M/t
##  $ unknown    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Action     : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ Adventure  : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Animation  : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Children's : int  1 0 0 0 0 0 0 1 0 0 ...
##  $ Comedy     : int  1 0 0 1 0 0 0 1 0 0 ...
##  $ Crime      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Documentary: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Drama      : int  0 0 0 1 1 1 1 1 1 1 ...
##  $ Fantasy    : int  0 0 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 0 0 0 0 0 0 0 0 ...
##  $ Sci-Fi     : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Thriller   : int  0 1 1 0 1 0 0 0 0 0 ...
##  $ War        : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Western    : int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
class(MovieLense)
```

```
## [1] "realRatingMatrix"
## attr(,"package")
## [1] "recommenderlab"
```

```r
methods(class = class(MovieLense))
```

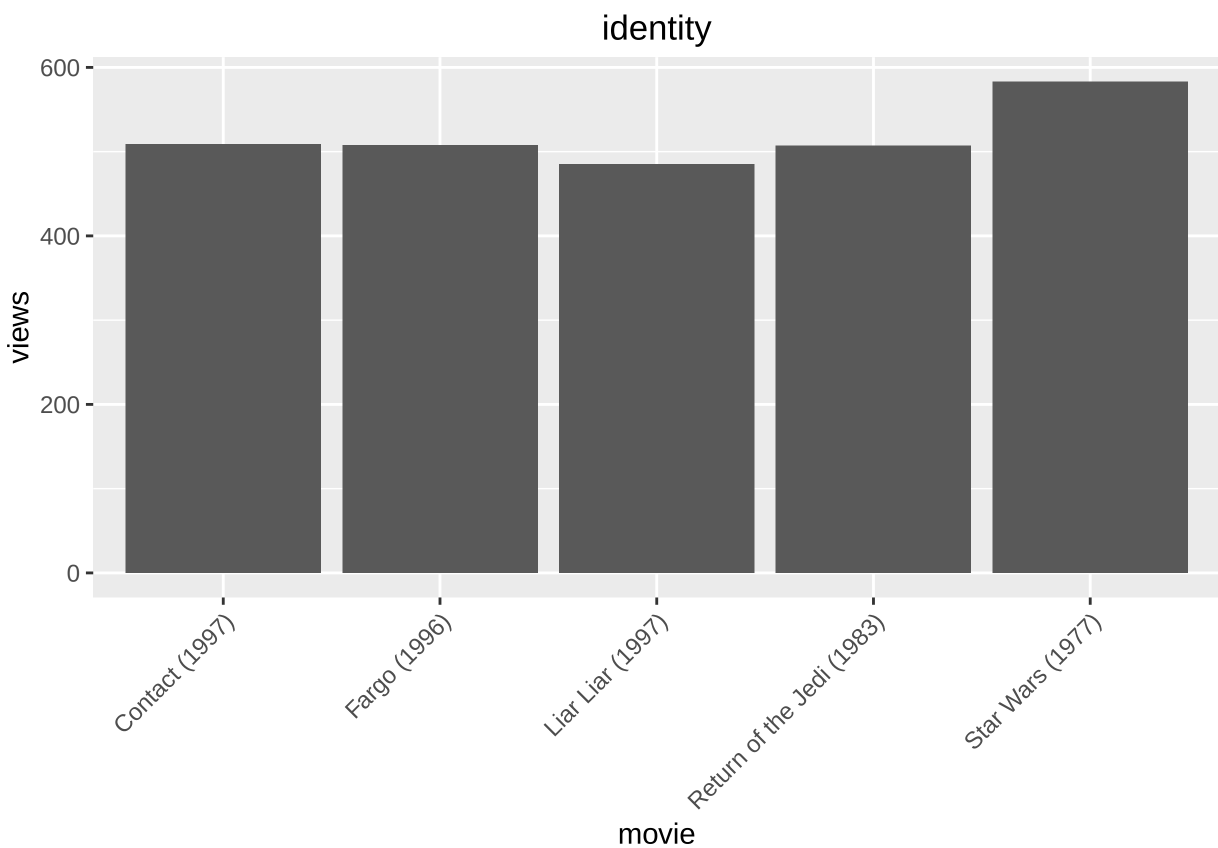
```
##  [1] [                      [<-                    binarize
##  [4] calcPredictionAccuracy coerce                 colCounts
##  [7] colMeans               colSds                 colSums
## [10] denormalize            dim                    dimnames
## [13] dimnames<-             dissimilarity          evaluationScheme
## [16] getData.frame          getList                getNormalize
## [19] getRatingMatrix        getRatings             getTopNLists
## [22] image                  normalize              nratings
## [25] Recommender            removeKnownRatings     rowCounts
## [28] rowMeans               rowSds                 rowSums
## [31] sample                 show                   similarity
## see '?methods' for accessing help and source code
```

```r
data<- read.csv("~/Downloads/unifiedMLDataMulti.csv")
```

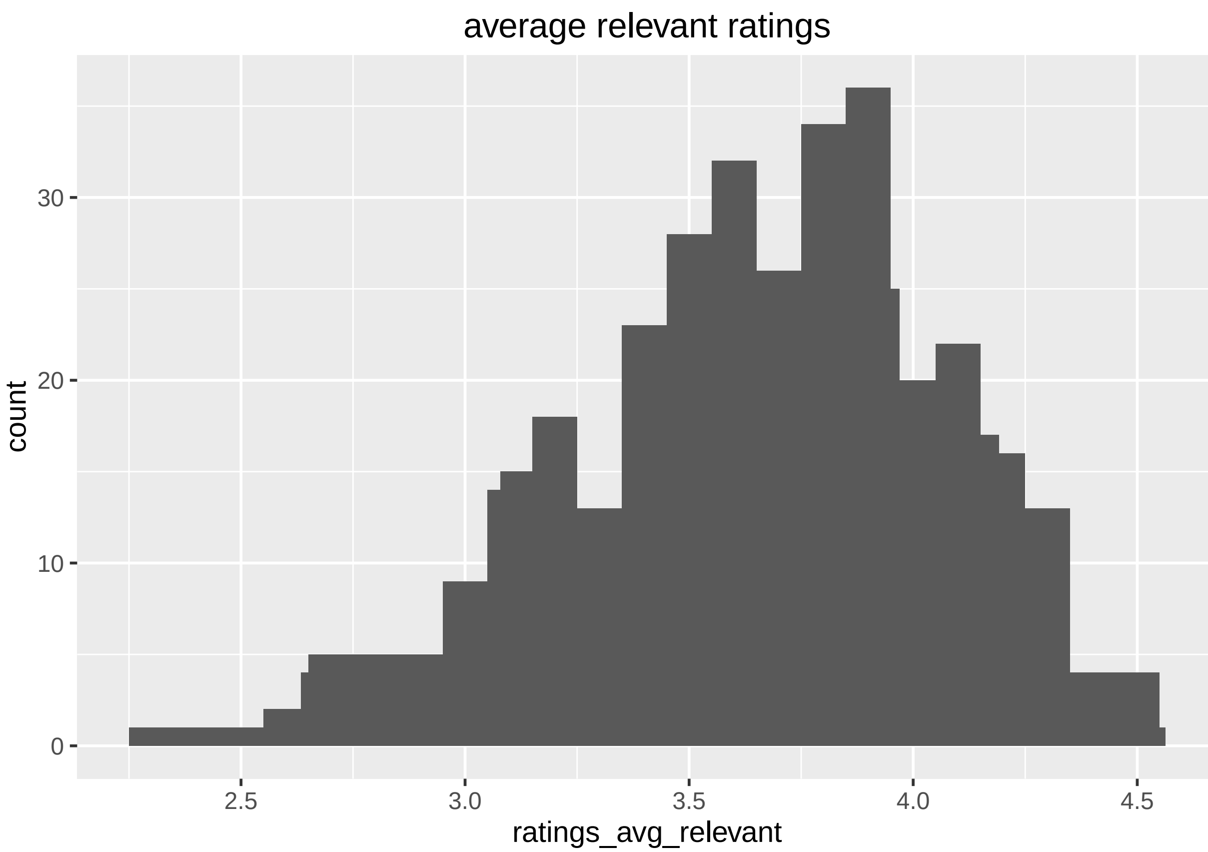
```r
views_per_movie <- colCounts(MovieLense)

views_tbl <- data.frame(
  movie = names(views_per_movie),
  views = views_per_movie
)

views_tbl <- views_tbl[order(views_tbl$views, decreasing = TRUE), ]

head(views_tbl)
```

```
##                                                 movie views
## Star Wars (1977)                     Star Wars (1977)   583
## Contact (1997)                         Contact (1997)   509
## Fargo (1996)                             Fargo (1996)   508
## Return of the Jedi (1983)     Return of the Jedi (1983)   507
## Liar Liar (1997)                     Liar Liar (1997)   485
## English Patient, The (1996) English Patient, The (1996)   481
```

```r
ggplot(views_tbl[1:5, ], aes(x = movie, y = views)) +
  geom_bar(stat="identity") + theme(axis.text.x = element_text(angle = 45, hjust = 1)) + ggtitle("identi
```



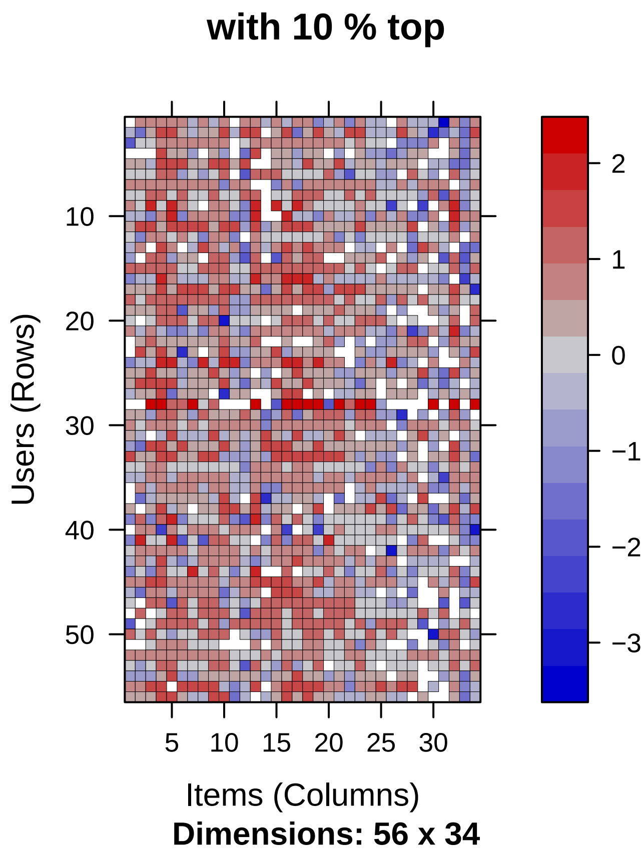
```
ratings_avg <- colMeans(MovieLense)

ratings_avg_relevant <- ratings_avg[views_per_movie > 100]

qplot(ratings_avg_relevant) + stat_bin(binwidth = 0.1) +
  ggtitle(paste("average relevant ratings"))
```




```
ratings_movies <- MovieLense[rowCounts(MovieLense) > 50,
                             colCounts(MovieLense) > 100]
ratings_movies
```

```
## 560 x 332 rating matrix of class 'realRatingMatrix' with 55298 ratings.
```

```
min_movies <- quantile(rowCounts(ratings_movies), 0.90)
min_users <- quantile(colCounts(ratings_movies), 0.90)

ratings_movies_norm <- normalize(ratings_movies)

# visualize the normalized matrix
image(ratings_movies_norm[rowCounts(ratings_movies_norm) > min_movies,
                          colCounts(ratings_movies_norm) > min_users], main = "with 10 % top")
```




```r
ratings_movies <- MovieLense[rowCounts(MovieLense) > 50,
                             colCounts(MovieLense) > 100]
ratings_movies
```

```
## 560 x 332 rating matrix of class 'realRatingMatrix' with 55298 ratings.
```

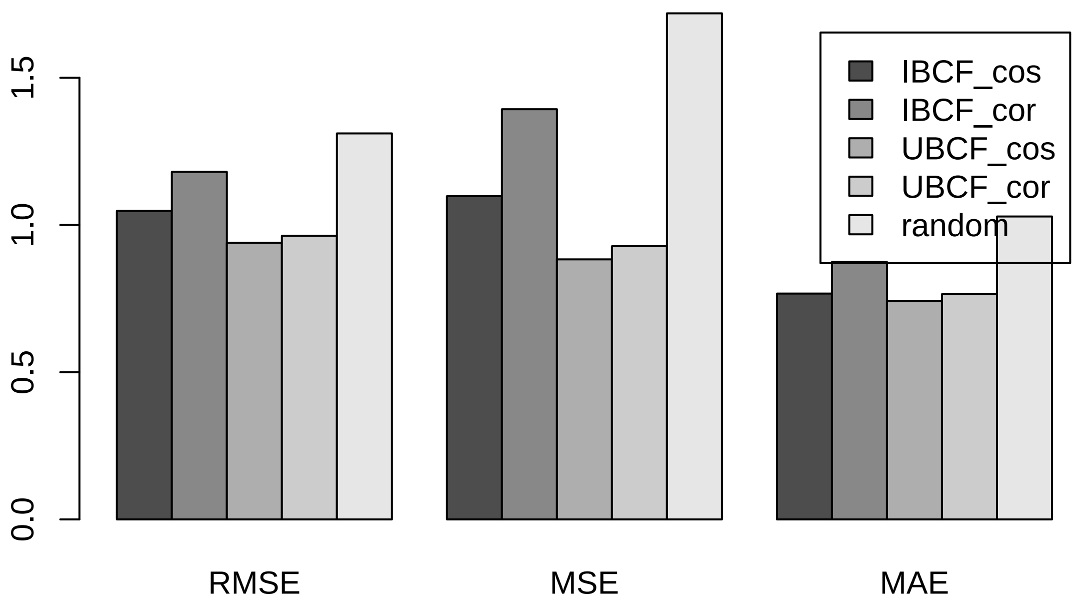
```r
percentage_training <- 0.8
items_to_keep <- 15
rating_threshold <- 3
n_eval <- 1

eval_scheme <- evaluationScheme(data = ratings_movies, method = "split",
                                train = percentage_training,
                                given = items_to_keep,
                                goodRating = rating_threshold,
                                k = n_eval)
eval_scheme
```

```
## Evaluation scheme with 15 items given
## Method: 'split' with 1 run(s).
## Training set proportion: 0.800
## Good ratings: >=3.000000
## Data set: 560 x 332 rating matrix of class 'realRatingMatrix' with 55298 ratings.
```

```r
algorithms_to_evaluate <- list(
  IBCF_cos = list(name = "IBCF", param = list(method ="cosine")),
  IBCF_cor = list(name = "IBCF", param = list(method ="pearson")),
  UBCF_cos = list(name = "UBCF", param = list(method ="cosine")),
  UBCF_cor = list(name = "UBCF", param = list(method ="pearson")),
  random = list(name = "RANDOM", param = NULL)
)

n_recommendations <- c(1, 5, seq(10, 100, 10))

results <- evaluate(eval_scheme, algorithms_to_evaluate, type = "ratings")
```

```
## IBCF run fold/sample [model time/prediction time]
##   1  [0.417sec/0.036sec]
## IBCF run fold/sample [model time/prediction time]
##   1  [0.502sec/0.021sec]
## UBCF run fold/sample [model time/prediction time]
##   1  [0.004sec/0.219sec]
## UBCF run fold/sample [model time/prediction time]
##   1  [0.005sec/0.301sec]
## RANDOM run fold/sample [model time/prediction time]
##   1  [0.001sec/0.023sec]
```

```r
plot(results)
```




```r
sapply(results, class) == "evaluationResults"
```

```
## IBCF_cos IBCF_cor UBCF_cos UBCF_cor   random
##     TRUE     TRUE     TRUE     TRUE     TRUE
```

```r
lapply(results, avg)
```

```
## $IBCF_cos
##         RMSE      MSE       MAE
## res 1.047816 1.097918 0.7671124
##
## $IBCF_cor
##        RMSE      MSE       MAE
## res 1.18038 1.393297 0.8744623
##
## $UBCF_cos
##          RMSE       MSE       MAE
## res 0.9398287 0.8832779 0.7420825
##
## $UBCF_cor
##          RMSE       MSE       MAE
## res 0.9632902 0.9279279 0.7650781
##
## $random
##         RMSE      MSE    MAE
## res 1.311106 1.718998 1.0289
```

```r
sapply(results, avg)
```

```
##       IBCF_cos  IBCF_cor  UBCF_cos  UBCF_cor   random
## [1,] 1.0478158 1.1803799 0.9398287 0.9632902 1.311106
## [2,] 1.0979180 1.3932967 0.8832779 0.9279279 1.718998
## [3,] 0.7671124 0.8744623 0.7420825 0.7650781 1.028900
```

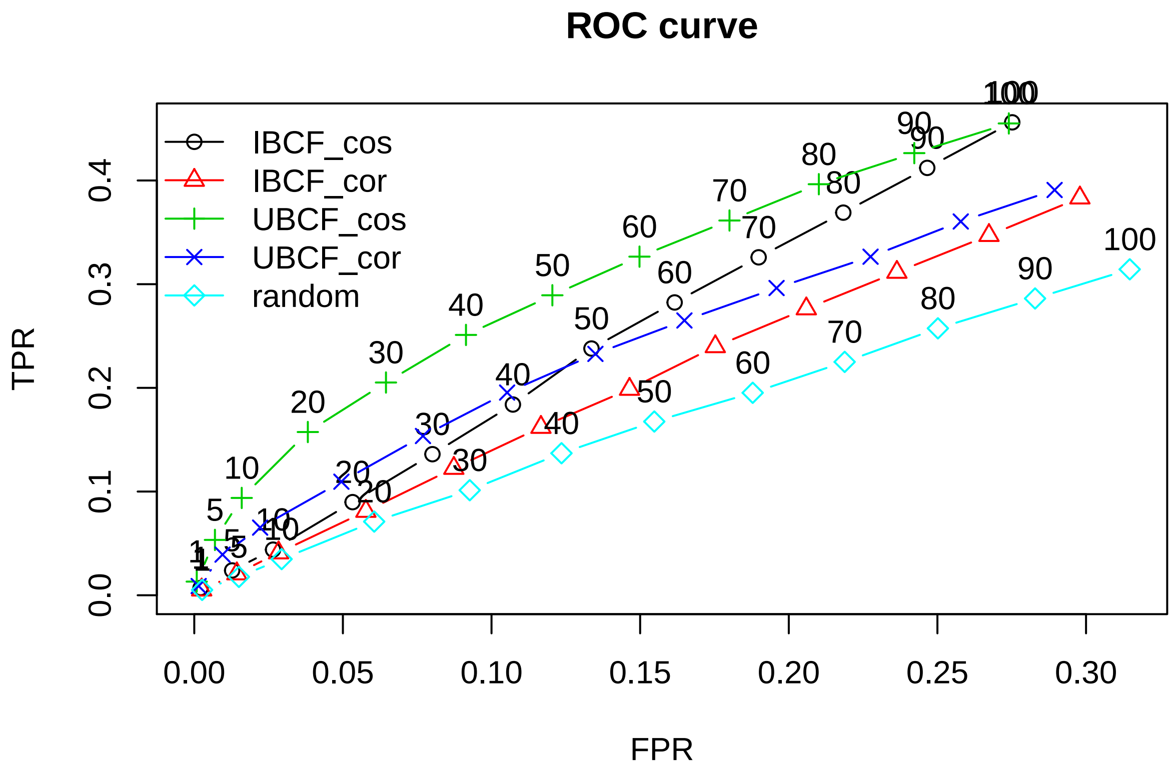
```r
recom_results <- evaluate(x = eval_scheme, method = algorithms_to_evaluate, n = n_recommendations)
```

```
## IBCF run fold/sample [model time/prediction time]
##   1  [0.383sec/0.047sec]
## IBCF run fold/sample [model time/prediction time]
##   1  [0.501sec/0.045sec]
## UBCF run fold/sample [model time/prediction time]
##   1  [0.005sec/0.248sec]
## UBCF run fold/sample [model time/prediction time]
##   1  [0.004sec/0.229sec]
## RANDOM run fold/sample [model time/prediction time]
##   1  [0.001sec/0.057sec]
```

```r
sapply(recom_results, class) == "evaluationResults"
```

```
## IBCF_cos IBCF_cor UBCF_cos UBCF_cor   random
##     TRUE     TRUE     TRUE     TRUE     TRUE
```

```r
avg_matrices <- lapply(recom_results, avg)

plot(recom_results, annotate = c(1,3,5),  legend = "topleft")
title("ROC curve")
```

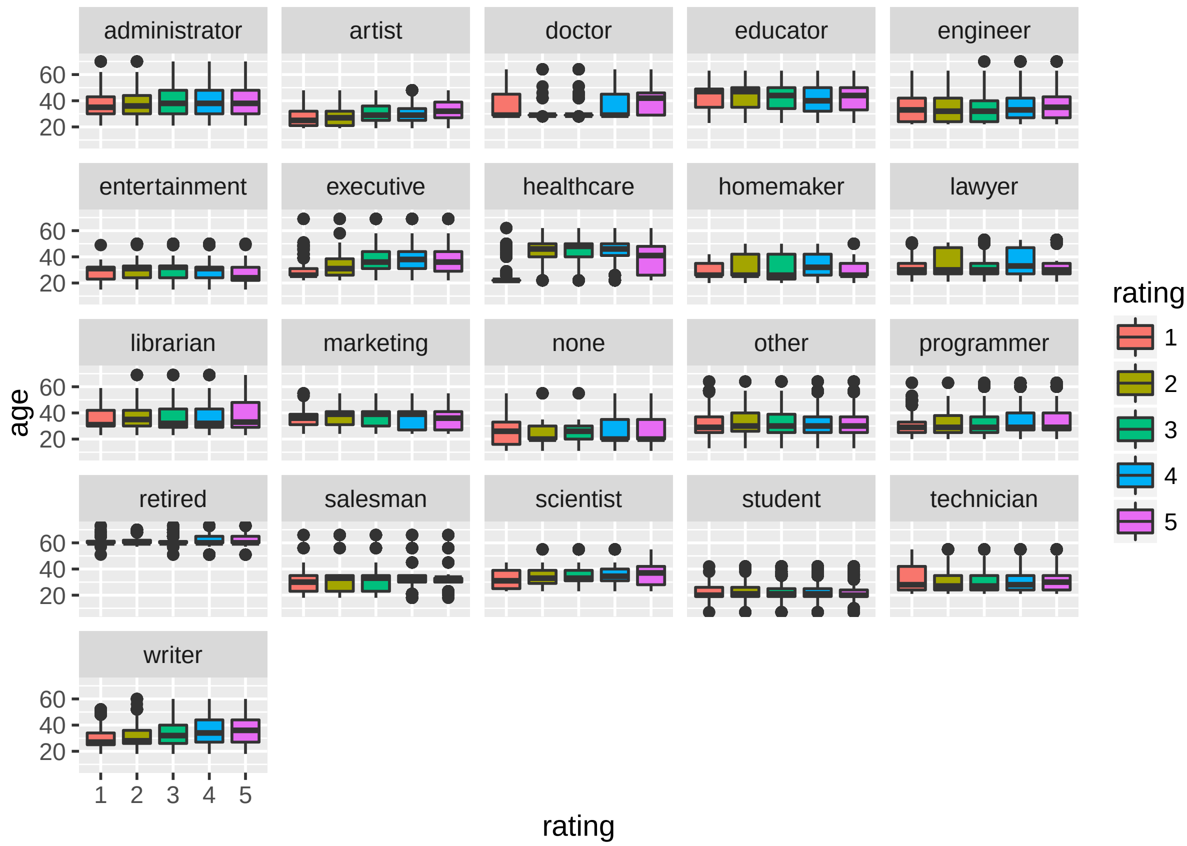



```
#UBCF_COS appeared to perform well in this dataset
```

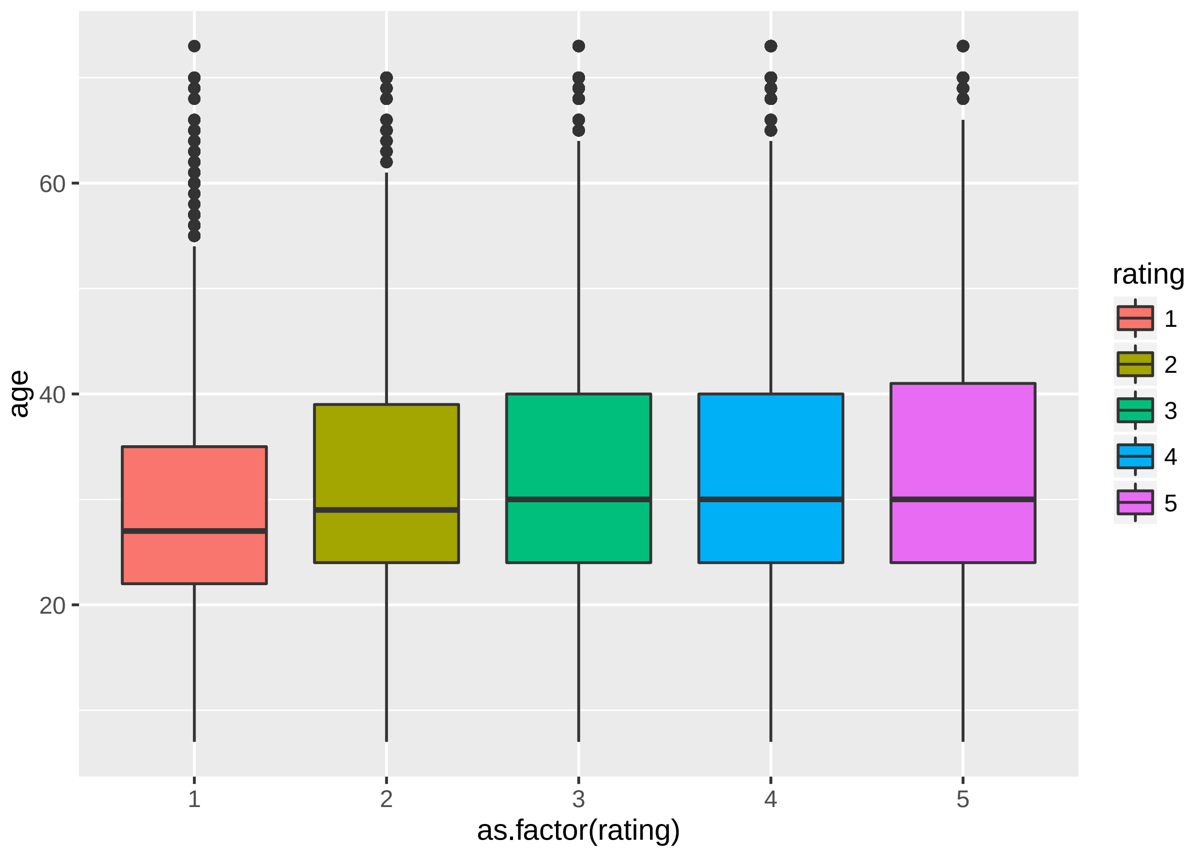
UBCF_COS appeared to perform well in this dataset

PART2 EXPLORING THE SECOND DATA

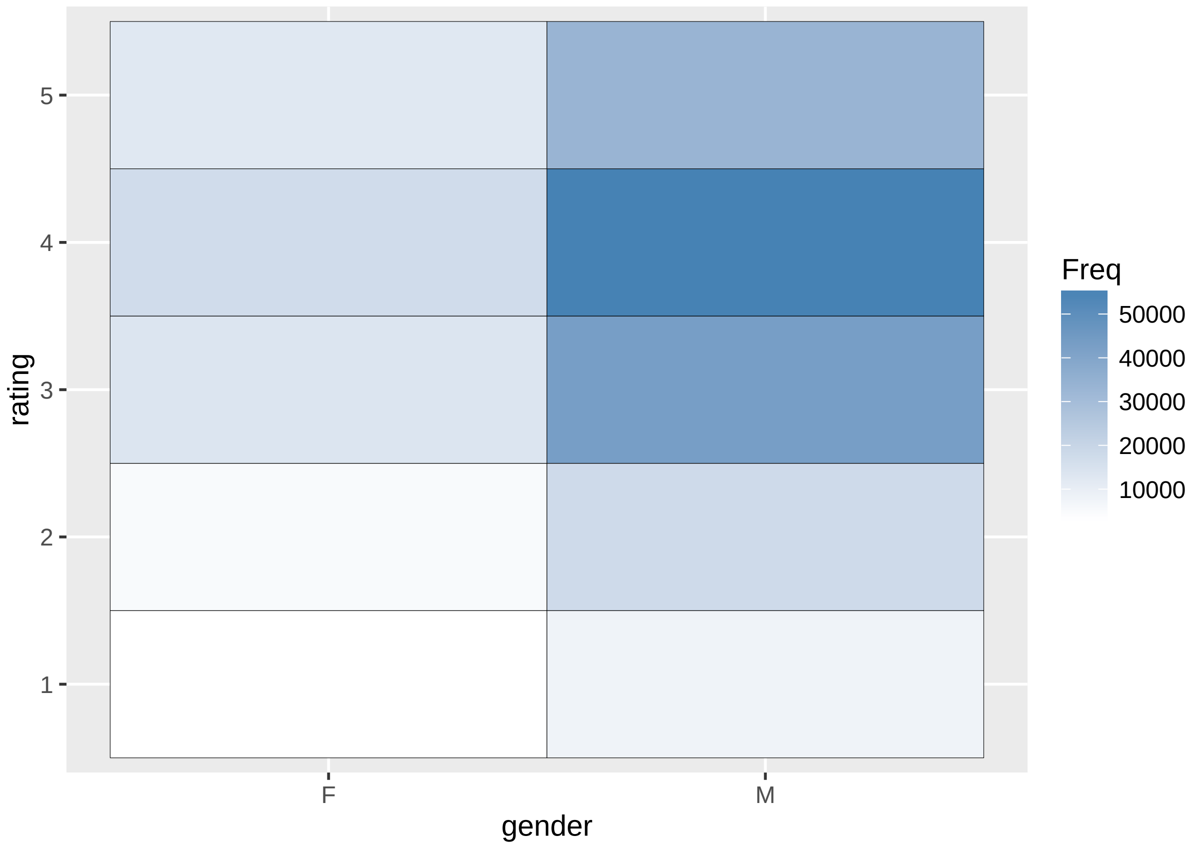
```
library(RColorBrewer)
library(grid)
library(plyr)
library(dplyr)

ggplot(data, aes(x=as.factor(rating),y=age)) +
  geom_boxplot(aes(fill=as.factor(rating))) +
  scale_fill_discrete(name="rating") +
  facet_wrap(~occupation)+xlab("rating")
```



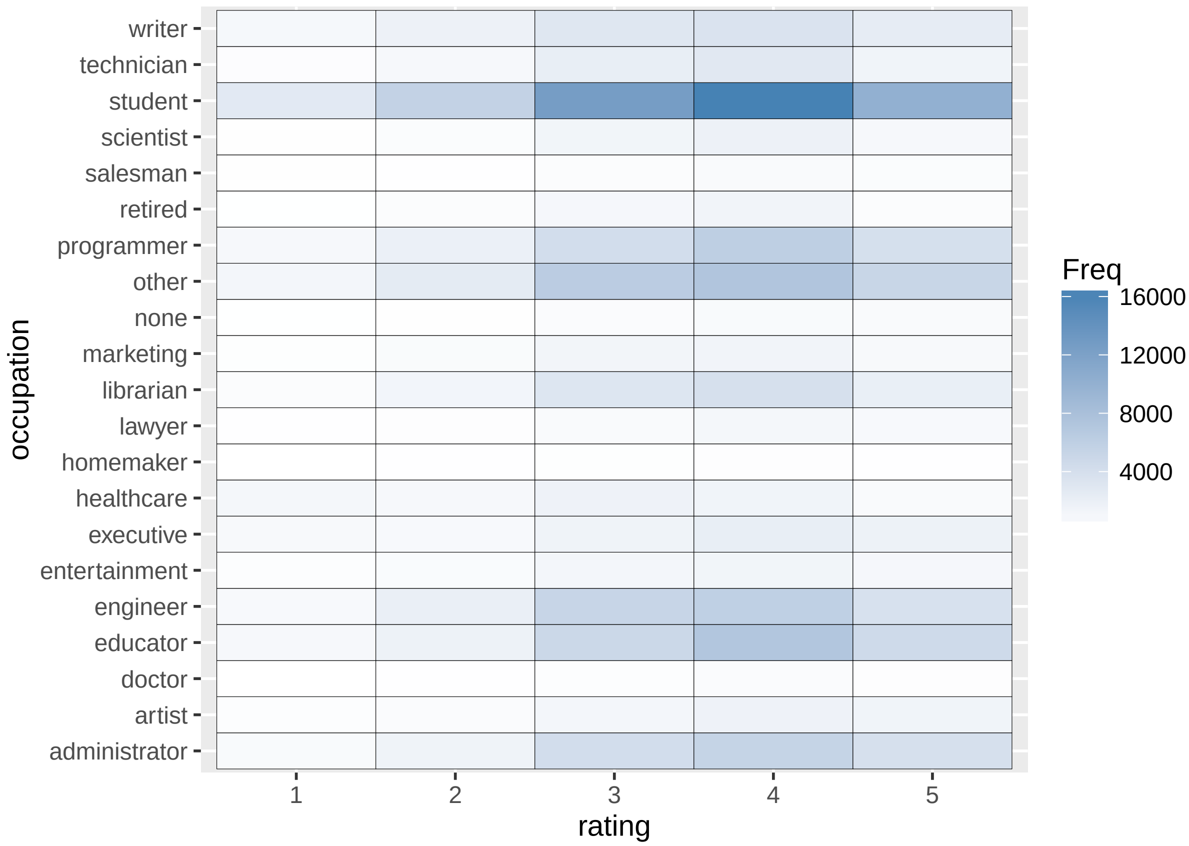

```r
# age VS rating
ggplot(data, aes(x=as.factor(rating),y=age)) +
  geom_boxplot(aes(fill=as.factor(rating))) +
  scale_fill_discrete(name="rating")
```



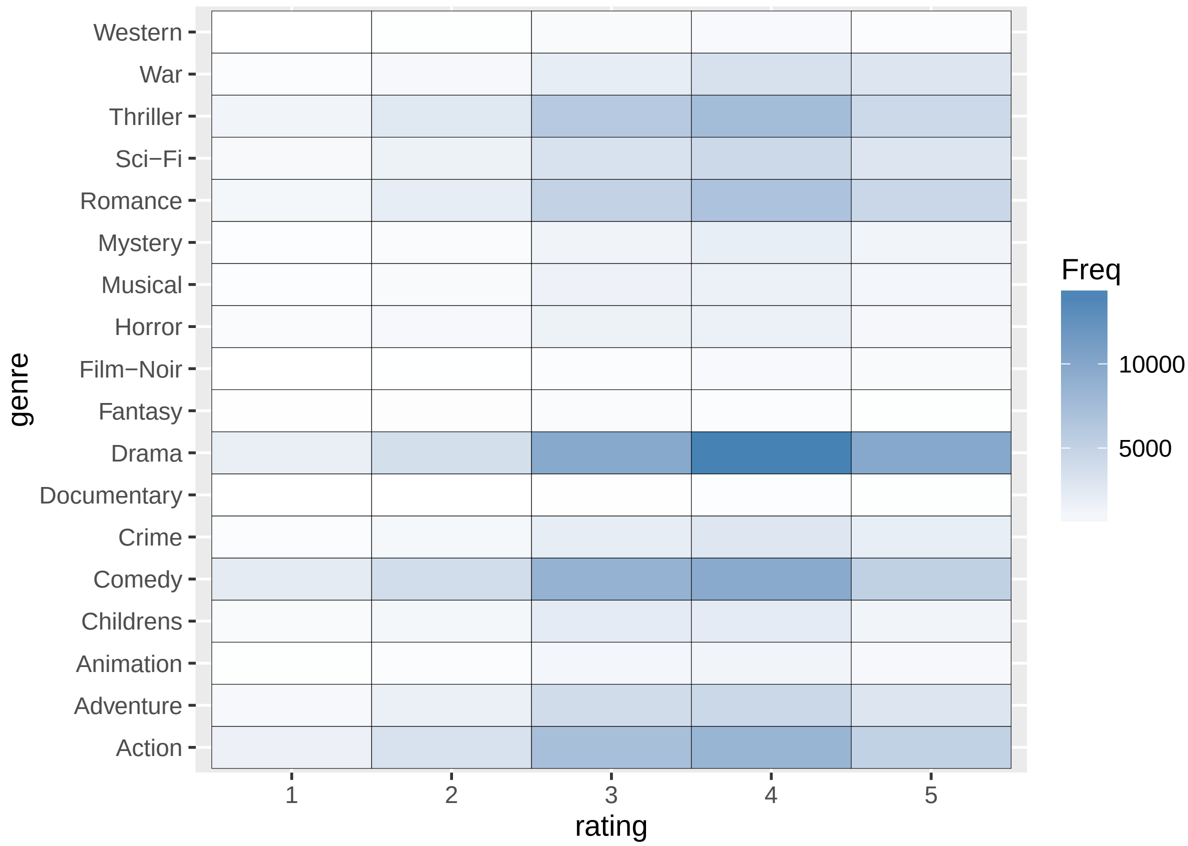

```r
# rating VS gender
gender.df <- as.data.frame(table(data$gender, data$rating))
ggplot(gender.df, aes(x=Var1, y=Var2)) +
  geom_tile(aes(fill = Freq), colour = "black") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  xlab("gender") +
  ylab("rating")
```



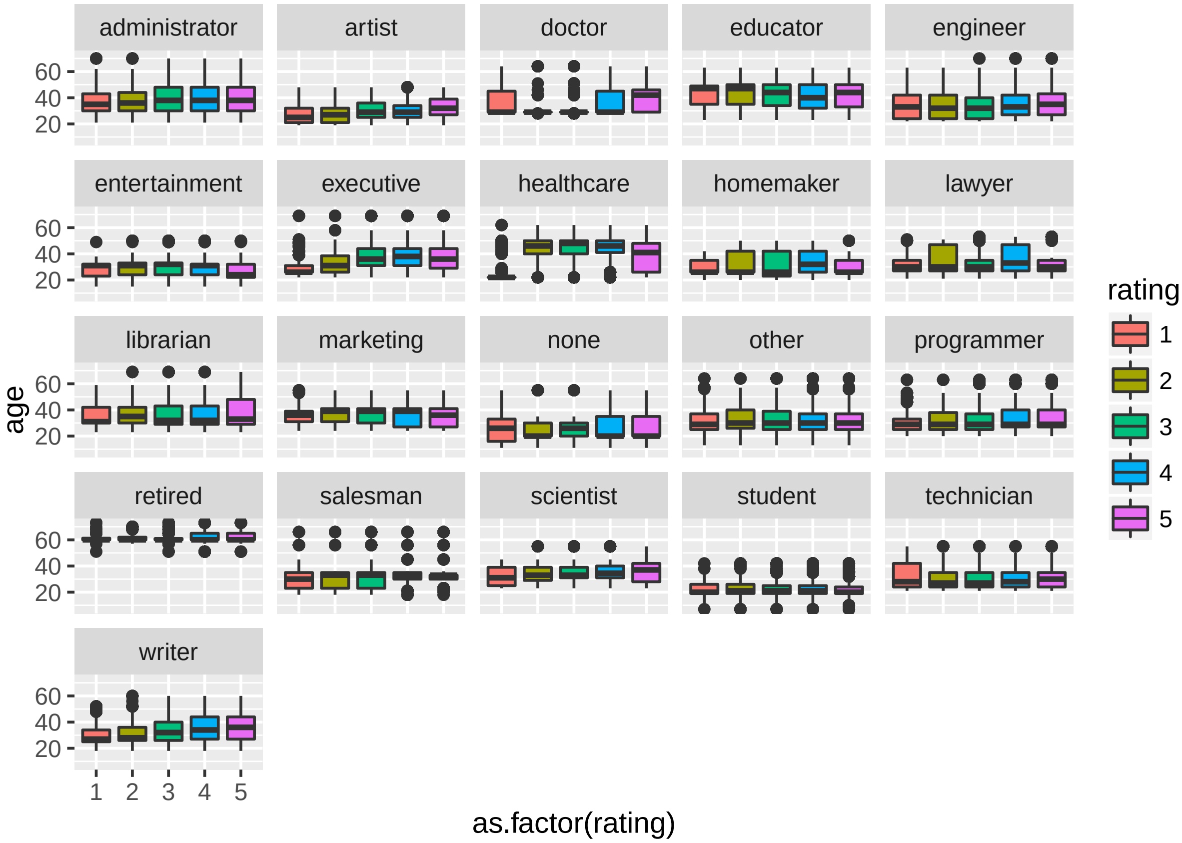

```r
# rating VS occupation
occupation.df <- as.data.frame(table(data$occupation, data$rating))
ggplot(occupation.df, aes(x=Var1, y=Var2)) +
  geom_tile(aes(fill = Freq), colour = "black") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  xlab("occupation") +
  ylab("rating")+coord_flip()
```




```r
# rating VS genre
genre.df <- as.data.frame(table(data$genre, data$rating))
ggplot(genre.df, aes(x=Var1, y=Var2)) +
  geom_tile(aes(fill = Freq), colour = "black") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  xlab("genre") +
  ylab("rating")+coord_flip()
```




```r
# age VS rating VS occupation
ggplot(data, aes(x=as.factor(rating),y=age)) +
  geom_boxplot(aes(fill=as.factor(rating))) +
  scale_fill_discrete(name="rating") +
  facet_wrap(~occupation)
```



Age component

young people rate lower ,rating seems to be positively related to age .

Gender

Men rate more than women and they rates at 4 most

Occupation

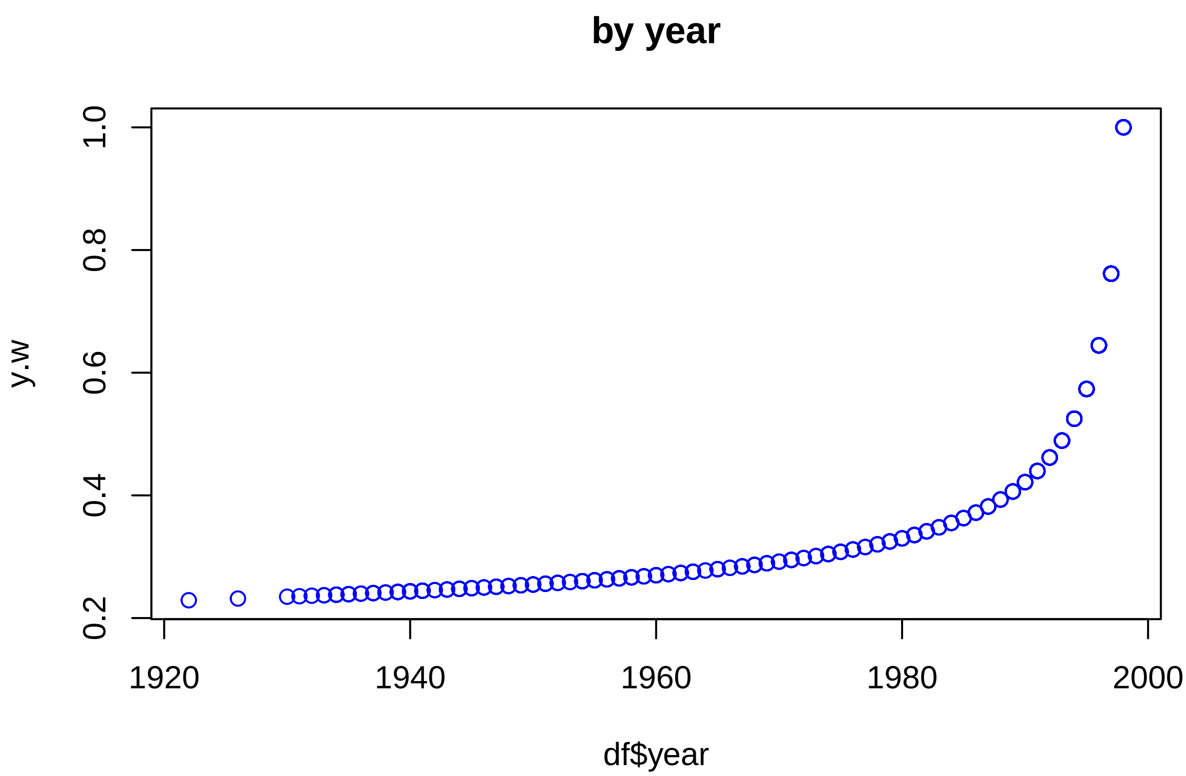
Comparing with other occupations, the number of students who rate is the largest; and students rates at 4 most.

Type of movie

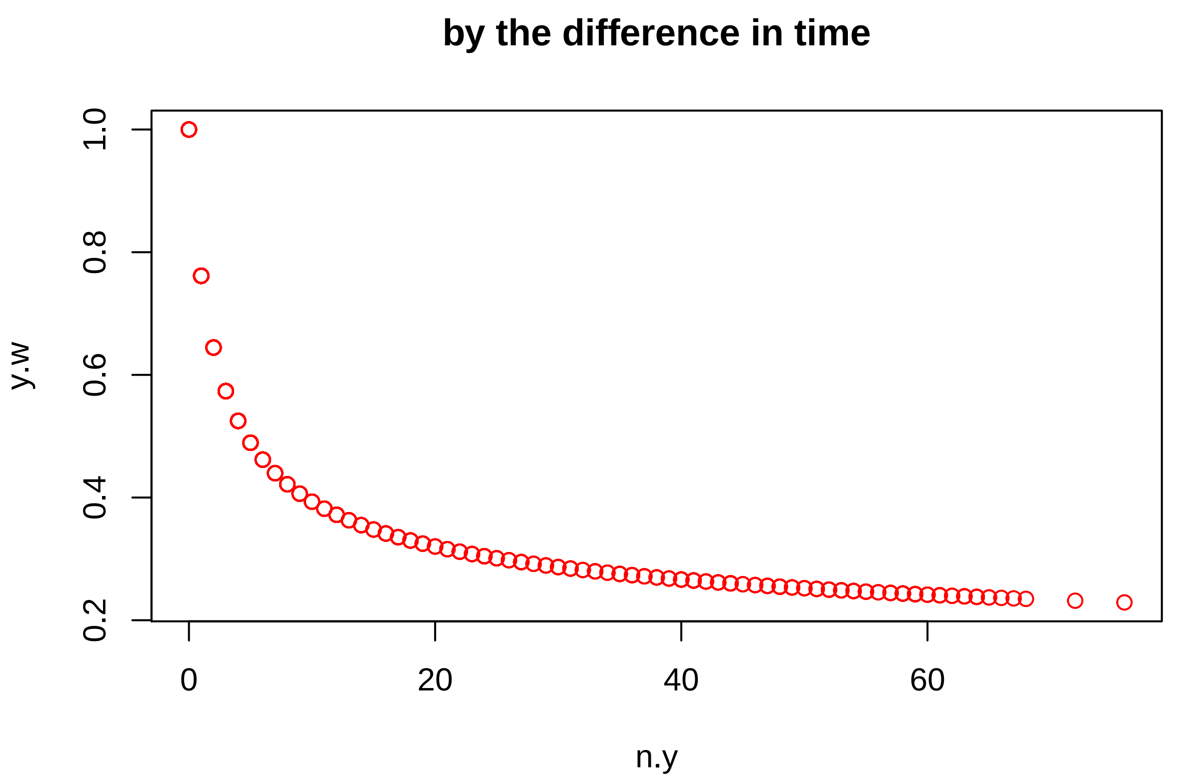
Drama, comedy, action have more rates

PART3 TIME CONTEXT IN TERM OF THE YEAR RELEASE

```r
#summary(MovieLenseMeta)
#str(MovieLense)
df=MovieLenseMeta
#We will only consider users  that views more than 100 movies and who rate more than 50 movies
ratings_movies <- MovieLense[rowCounts(MovieLense) > 50, colCounts(MovieLense) > 100]



set.seed(0)
test <- sample(x = 1:5,
                    size = nrow(ratings_movies),
                    replace = TRUE)
for(i in 1:5) {
  train <- test == i
  Rtrain <- ratings_movies[train, ]
  Rtest <- ratings_movies[!train, ]
}

model<- Recommender(data = Rtrain, method = "UBCF")

predictions <- predict(model, newdata = Rtest, n = 15)

#Choose the biggest year as a benchmark
m <- max(MovieLenseMeta$year, na.rm = TRUE)
#Check the number of years between

n.y <- m- df$year

yrs <- as.numeric(levels(as.factor(n.y)))
wts <- 1 / log(yrs + exp(1))

#year weight
y.w <- 1 / log(n.y + exp(1))
#Exponential decay
plot(df$year,y.w,col="blue",main="by year")
```




```
plot(n.y,y.w,col="red",main="by the difference in time")
```



```r
#Remove na
y.w[is.na(y.w)] <- 0
weights <- data.frame(title = df$title, wt = y.w, stringsAsFactors = FALSE)

Rdf <- data.frame(user = sort(rep(1:length(predictions@items),predictions@n)), rating = unlist(predicti
Rdf$title <- predictions@itemLabels[Rdf$index]
library(dplyr)
Rwt <- inner_join(Rdf, weights, by = "title")

Rwt <- Rwt%>% mutate(wt_rating = rating * wt) %>% group_by(user) %>% arrange(desc(wt_rating)) %>% selec
head(Rwt, 13)
```

```
## Source: local data frame [13 x 3]
## Groups: user [13]
##
##      user              title wt_rating
##     <int>              <chr>     <dbl>
## 1     433 Apt Pupil (1998)  4.935953
## 2     161 Apt Pupil (1998)  4.819775
## 3     273 Apt Pupil (1998)  4.553560
## 4      47 Apt Pupil (1998)  4.510672
## 5       4 Apt Pupil (1998)  4.426098
## 6     401 Apt Pupil (1998)  4.333340
## 7      28 Apt Pupil (1998)  4.256444
## 8     259 Apt Pupil (1998)  4.253136
## 9      25 Apt Pupil (1998)  4.220872
## 10    143 Apt Pupil (1998)  4.217124
## 11    222 Apt Pupil (1998)  4.197089
## 12     11 Apt Pupil (1998)  4.159134
## 13    314 Apt Pupil (1998)  4.052962
```

```r
Rdf2 <- Rdf %>% group_by(user) %>% arrange(desc(rating)) %>% select(user, title, rating) %>% top_n(5, ra

head(Rdf2, 10)
```

```
## Source: local data frame [10 x 3]
## Groups: user [4]
##
##      user                        title   rating
##     <int>                        <chr>    <dbl>
## 1      62 Usual Suspects, The (1995) 5.429503
## 2      62      Godfather, The (1972) 5.383490
## 3      62           Star Wars (1977) 5.340693
## 4     341               Fargo (1996) 5.292231
## 5     341          Casablanca (1942) 5.257697
## 6      62        Blade Runner (1982) 5.248608
## 7      62 Princess Bride, The (1987) 5.203360
## 8     265      Godfather, The (1972) 5.195706
## 9     265             Vertigo (1958) 5.166715
## 10    271           Star Wars (1977) 5.150736
```

CONCLUSION AND FUTURE WORK

From this work ,we can see that the best recommender algorithm depend on the data we have in hand ,the context of the business will determine which algorithm to pick.One must dive deeper to understand the paterns of the data as well to get insights ,it is also important to explore all types of data that could get involved in the business . It is also important to take into consideration other factors such as time ,location and connections to other intities. For future work ,I would to extend this work on computing similarity related to the age ,sex,occupation and give recommendations based on those factors.

Reference : Suresh K. Gorakala, Michele Usuelli **Time weights** Building a Recommendation System with R https://www.packtpub.com/big-data-and-business-intelligence/building-recommendation-system-r

unifiedMLDataMulti.csv Justin Chu : cleaning the grouplense data and puting into :unifiedMLDataMulti.csv

https://github.com/JustinChu?tab=repositories