ANA 515 Assignment 4 Data Analytics Project

Prashanth Paisa

2022-10-15

## Business Goal

The goal of performing analysis of MovieLens Dataset is to create user/movie profile by using their movie watch history. Understanding of the users mindset and according to that recommending them with a movie genre is the important goal of this analysis. Apart from that, this analysis will also help movie industry about the way to release a movie and the genre of the movies from the knowledge extracted from this data analysis.

## Dataset Retrieval

This MovieLens Dataset is retrieved from the following link:

<https://drive.google.com/file/d/1Dn1BZD3YxgBQJSIjbfNnmCFlDW2jdQGD/view>

## Intializing required libraries

library(recommenderlab)

## Warning: package 'recommenderlab' was built under R version 4.2.1

## 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

## Warning: package 'proxy' was built under R version 4.2.1

##   
## 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

library(ggplot2)  
library(data.table)  
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.2.1

##   
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':  
##   
## dcast, melt

library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.2.1

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## Loading Data

movie\_data <- read.csv("IMDB/movies.csv",stringsAsFactors=FALSE)  
rating\_data <- read.csv("IMDB/ratings.csv")  
str(movie\_data)

## '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)" "Waiting to Exhale (1995)" ...  
## $ genres : chr "Adventure|Animation|Children|Comedy|Fantasy" "Adventure|Children|Fantasy" "Comedy|Romance" "Comedy|Drama|Romance" ...

summary(movie\_data)

## 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

summary(rating\_data)

## 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

## Describing Data

describe(movie\_data)

## movie\_data   
##   
## 3 Variables 10329 Observations  
## --------------------------------------------------------------------------------  
## movieId   
## n missing distinct Info Mean Gmd .05 .10   
## 10329 0 10329 1 31924 39140 581.8 1273.8   
## .25 .50 .75 .90 .95   
## 3240.0 7088.0 59900.0 93202.8 105803.0   
##   
## lowest : 1 2 3 4 5, highest: 146684 146878 148238 148626 149532  
## --------------------------------------------------------------------------------  
## title   
## n missing distinct   
## 10329 0 10327   
##   
## lowest : '71 (2014) 'burbs, The (1989) 'Hellboy': The Seeds of Creation (2004) 'night Mother (1986) 'Round Midnight (1986)   
## highest: Zorba the Greek (Alexis Zorbas) (1964) Zorro, the Gay Blade (1981) Zozo (2005) Zu: Warriors from the Magic Mountain (Xin shu shan jian ke) (1983) Zulu (1964)   
## --------------------------------------------------------------------------------  
## genres   
## n missing distinct   
## 10329 0 938   
##   
## lowest : (no genres listed) Action Action|Adventure Action|Adventure|Animation Action|Adventure|Animation|Children  
## highest: Sci-Fi|Thriller|IMAX Thriller Thriller|War War Western   
## --------------------------------------------------------------------------------

describe(rating\_data)

## rating\_data   
##   
## 4 Variables 105339 Observations  
## --------------------------------------------------------------------------------  
## userId   
## n missing distinct Info Mean Gmd .05 .10   
## 105339 0 668 1 364.9 227.7 60 88   
## .25 .50 .75 .90 .95   
## 192 383 557 625 668   
##   
## lowest : 1 2 3 4 5, highest: 664 665 666 667 668  
## --------------------------------------------------------------------------------  
## movieId   
## n missing distinct Info Mean Gmd .05 .10   
## 105339 0 10325 1 13381 20484 161 344   
## .25 .50 .75 .90 .95   
## 1073 2497 5991 56146 79242   
##   
## lowest : 1 2 3 4 5, highest: 146684 146878 148238 148626 149532  
## --------------------------------------------------------------------------------  
## rating   
## n missing distinct Info Mean Gmd .05 .10   
## 105339 0 10 0.965 3.517 1.151 1.5 2.0   
## .25 .50 .75 .90 .95   
## 3.0 3.5 4.0 5.0 5.0   
##   
## lowest : 0.5 1.0 1.5 2.0 2.5, highest: 3.0 3.5 4.0 4.5 5.0  
##   
## Value 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0  
## Frequency 1198 3258 1567 7943 5484 21729 12237 28880 8187 14856  
## Proportion 0.011 0.031 0.015 0.075 0.052 0.206 0.116 0.274 0.078 0.141  
## --------------------------------------------------------------------------------  
## timestamp   
## n missing distinct Info Mean Gmd .05 .10   
## 105339 0 84686 1 1.13e+09 207443703 8.484e+08 8.917e+08   
## .25 .50 .75 .90 .95   
## 9.711e+08 1.115e+09 1.275e+09 1.406e+09 1.441e+09   
##   
## lowest : 828564954 828564955 828564956 828565403 828803826  
## highest: 1452287529 1452347804 1452370333 1452370659 1452404919  
## --------------------------------------------------------------------------------

## Data Preparation

movie\_genre <- as.data.frame(movie\_data$genres, stringsAsFactors=FALSE)  
movie\_genre2 <- as.data.frame(tstrsplit(movie\_genre[,1], '[|]',   
 type.convert=TRUE),   
 stringsAsFactors=FALSE)  
colnames(movie\_genre2) <- c(1:10)  
  
list\_genre <- c("Action", "Adventure", "Animation", "Children",   
 "Comedy", "Crime","Documentary", "Drama", "Fantasy",  
 "Film-Noir", "Horror", "Musical", "Mystery","Romance",  
 "Sci-Fi", "Thriller", "War", "Western")  
genre\_mat1 <- matrix(0,10330,18)  
genre\_mat1[1,] <- list\_genre  
colnames(genre\_mat1) <- list\_genre  
  
for (index in 1:nrow(movie\_genre2)) {  
 for (col in 1:ncol(movie\_genre2)) {  
 gen\_col = which(genre\_mat1[1,] == movie\_genre2[index,col])  
 genre\_mat1[index+1,gen\_col] <- 1  
}  
}  
genre\_mat2 <- as.data.frame(genre\_mat1[-1,], stringsAsFactors=FALSE) #remove first row, which was the genre list  
for (col in 1:ncol(genre\_mat2)) {  
 genre\_mat2[,col] <- as.integer(genre\_mat2[,col]) #convert from characters to integers  
}   
str(genre\_mat2)

## '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 ...

## Model Implementation

SearchMatrix <- cbind(movie\_data[,1:2], genre\_mat2[])  
ratingMatrix <- dcast(rating\_data, userId~movieId, value.var = "rating", na.rm=FALSE)  
ratingMatrix <- as.matrix(ratingMatrix[,-1]) #remove userIds  
#Convert rating matrix into a recommenderlab sparse matrix  
ratingMatrix <- as(ratingMatrix, "realRatingMatrix")  
recommendation\_model <- recommenderRegistry$get\_entries(dataType = "realRatingMatrix")  
lapply(recommendation\_model, "[[", "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."

recommendation\_model$IBCF\_realRatingMatrix$parameters

## $k  
## [1] 30  
##   
## $method  
## [1] "cosine"  
##   
## $normalize  
## [1] "center"  
##   
## $normalize\_sim\_matrix  
## [1] FALSE  
##   
## $alpha  
## [1] 0.5  
##   
## $na\_as\_zero  
## [1] FALSE

## Data Visualization

Here a bar plot has been developed with Trips by Hour and Month.

similarity\_mat <- similarity(ratingMatrix[1:4, ],  
 method = "cosine",  
 which = "users")  
movie\_views <- colCounts(ratingMatrix) # count views for each movie  
table\_views <- data.frame(movie = names(movie\_views),  
 views = movie\_views) # create dataframe of views  
table\_views <- table\_views[order(table\_views$views,  
 decreasing = TRUE), ] # sort by number of views  
table\_views$title <- NA  
for (index in 1:10325){  
 table\_views[index,3] <- as.character(subset(movie\_data,  
 movie\_data$movieId == table\_views[index,1])$title)  
}  
table\_views[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)

ggplot(table\_views[1:6, ], aes(x = title, y = views)) +  
 geom\_bar(stat="identity", fill = 'steelblue') +  
 geom\_text(aes(label=views), vjust=-0.3, size=3.5) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 ggtitle("Total Views of the Top Films")

