Final project Step 2

Yograj Karki

5/21/2021

In this step, I've started to import libraries and data. I'll be looking at the data with str() function and also will be seeing the first few rows by head() function. Later on, I'll be creating the matrices of genres of movies so that the recommenderlab could work with the data.

Importing libraries

Importing the dataset

```
setwd("~/MSDS/DSC520/dsc520/Final_project")
movie_data <- read.csv("IMDB-Dataset/movies.csv", stringsAsFactors=FALSE)
rating_data <- read.csv("IMDB-Dataset/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 (19
##  $ genres : chr  "Adventure|Animation|Children|Comedy|Fantasy" "Adventure|Children|Fantasy" "Comedy|R
```

Glimpse of the data

```
# Movies data
head(movie_data)
```

```
##   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 data
head(rating_data)
```

```
##   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
```

```
# extracting the genres as a dataframe
movie_genre <- as.data.frame(movie_data$genres, stringsAsFactors=FALSE)

# Splitting the collective genres into individual ones
movie_genre2 <- as.data.frame(tstrsplit(movie_genre[,1], '[|]', type.convert=TRUE), stringsAsFactors=FA

# Assigning column names as serial numbers assuming each movie may have maximum of 10 genres
colnames(movie_genre2) <- c(1:10)
```

```
# List of all the genres

list_genre <- unique(movie_genre2[c("1")])[1:18,] # there was 19 values but last one was without genre

#list_genre <- c("Action", "Adventure", "Animation", "Children", "Comedy",
    #            "Crime","Documentary","Drama", "Fantasy", "Film-Noir",
      #          "Horror","Musical", "Mystery","Romance","Sci-Fi", "Thriller", "War", "Western")

# Initializing a matrix
genre_mat1 <- matrix(0,10330,18)

genre_mat1[1,] <- list_genre

# Assigning genres  as column names
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) #removing first row, which was the

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:
##  $ Adventure  : int  1 1 0 0 0 0 0 1 0 1 ...
##  $ Comedy     : int  1 0 1 1 1 0 1 0 0 0 ...
##  $ Action     : int  0 0 0 0 0 1 0 0 1 1 ...
##  $ Drama      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Crime      : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Children   : int  1 1 0 0 0 0 0 1 0 0 ...
##  $ Mystery    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Animation  : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Documentary: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Thriller   : int  0 0 0 0 0 1 0 0 0 1 ...
##  $ Horror     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Fantasy    : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ Western    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Film-Noir  : 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 ...
##  $ Musical    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ War        : int  0 0 0 0 0 0 0 0 0 0 ...
```

Combining the genre matrix with the movies data resulting in a search matrix

```
SearchMatrix <- cbind(movie_data[,1:2], genre_mat2[])
head(SearchMatrix)
```

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

For the movie recommendation system to make sense of the ratings through recommenderlabs, we have to convert our matrix into a sparse matrix one. This new matrix is of the class 'realRatingMatrix'.

```r
ratingMatrix <- reshape2::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")
ratingMatrix
```

```
## 668 x 10325 rating matrix of class 'realRatingMatrix' with 105339 ratings.
```

Exploring recommendation model options

```r
recommendation_model <- recommenderRegistry$get_entries(dataType = "realRatingMatrix")
names(recommendation_model)
```

```
##  [1] "HYBRID_realRatingMatrix"       "ALS_realRatingMatrix"
##  [3] "ALS_implicit_realRatingMatrix" "IBCF_realRatingMatrix"
##  [5] "LIBMF_realRatingMatrix"        "POPULAR_realRatingMatrix"
##  [7] "RANDOM_realRatingMatrix"       "RERECOMMEND_realRatingMatrix"
##  [9] "SVD_realRatingMatrix"          "SVDF_realRatingMatrix"
## [11] "UBCF_realRatingMatrix"
```

Since we're interested to create the model based on IBCF algorithm or Item Based Collaborative filtering, let's look at the parameters for that.

```r
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
```

So far, I have reached at this step, Next steps will be: - Creating visualizations of ratings and most watched movies etc.. I'll be exploring data as much as possible. - Then, I'll work on the recommender system based on Collaborative Filtering System. For that, I'll split the data in 80:20 ratio for - - training and testing purposes. - Train the model __ Make some predictions/recommendations - Validate the model