**MovieLens Recommender System Capstone Project -**

Dania Zhu

2021/7/1

**set working dir**

setwd("C:\\Users\\user\\Desktop\\Harvard\_R\\CapStone\\MovieLensRecommendation")

**MovieLens 10M dataset:**

[**https://grouplens.org/datasets/movielens/10m/**](https://grouplens.org/datasets/movielens/10m/)

[**http://files.grouplens.org/datasets/movielens/ml-10m.zip**](http://files.grouplens.org/datasets/movielens/ml-10m.zip)

#dl <- tempfile() #download.file(“http://files.grouplens.org/datasets/movielens/ml-10m.zip”, dl)

**creat edx, validation dataset**

**open previous saved files to save compile time**

#ratings <- read.table(text = gsub(“::”, “, readLines(unzip(dl,”ml-10M100K/ratings.dat“))), # col.names = c(”userId“,”movieId“,”rating“,”timestamp"))

**Save ratings object at my laptop, file size around 46MB**

#saveRDS(ratings, “ratings.rda”)

#movies <- str\_split\_fixed(readLines(unzip(dl, “ml-10M100K/movies.dat”)), “\::”, 3) #colnames(movies) <- c(“movieId”, “title”, “genres”) # movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(movieId), # title = as.character(title), # genres = as.character(genres))

#movielens <- left\_join(ratings, movies, by = “movieId”)

#saveRDS(movies, “movies.rda”) #saveRDS(movielens, “movielens.rda”)

**Validation set will be 10% of MovieLens data**

**set.seed(1)**

**movielens\_model<-readRDS(“movielens.rda”)**

**test\_index <- createDataPartition(y = movielens\_model$rating, times = 1, p = 0.1, list = FALSE)**

#edx <- movielens\_model[-test\_index,] #temp <- movielens\_model[test\_index,]

**Make sure userId and movieId in validation set are also in edx set**

#validation <- temp %>% # semi\_join(edx, by = “movieId”) %>% # semi\_join(edx, by = “userId”)

**Add rows removed from validation set back into edx set**

#removed <- anti\_join(temp, validation) #edx <- rbind(edx, removed)

#saveRDS(edx, “edx.rda”) #saveRDS(validation, “validation.rda”)

#rm(dl, ratings, movies, test\_index, temp, movielens, removed) #rm(temp, removed, test\_index) #gc()

**MovieLens Reccomender System Project**

edx\_dat <- readRDS("edx.rda")  
validation\_dat <-readRDS("validation.rda")

**1.Descriptive Analysis**

*# Some movies have more rating than others*  
  
*# distinct movieIDs, 10677*  
n\_distinct(edx\_dat$movieId)

## [1] 10677





*# Number of users and movies in edx set*  
edx\_dat %>%  
 summarize(n\_users = n\_distinct(userId),  
 n\_movies = n\_distinct(movieId))

## n\_users n\_movies  
## 1 69878 10677

*# Top 5 movies ratings given by top 5 frequent users*  
top5 <- edx\_dat %>%  
 dplyr::count(movieId) %>%  
 top\_n(5) %>%  
 pull(movieId)

## Selecting by n

*# Top 5 frequent users*  
top5\_u <- edx\_dat %>%  
 dplyr::count(userId) %>%  
 top\_n(5) %>%  
 pull(userId)

top5\_u

## [1] 14463 27468 59269 67385 68259

tab <- edx\_dat %>%   
 filter(movieId %in% top5) %>%   
 filter(userId %in% top5\_u) %>%  
 select(userId, title, rating) %>%   
 spread(title, rating)   
print(tab)

## userId Forrest Gump (1994) Jurassic Park (1993) Pulp Fiction (1994)  
## 1 14463 3.0 4 4.0  
## 2 27468 5.0 5 5.0  
## 3 59269 4.0 3 4.5  
## 4 67385 3.5 4 5.0  
## 5 68259 4.0 4 5.0  
## Shawshank Redemption, The (1994) Silence of the Lambs, The (1991)  
## 1 4.0 4  
## 2 5.0 5  
## 3 4.0 4  
## 4 4.5 5  
## 5 5.0 5



*# View of all unique genres*  
unique\_genres\_list <- str\_extract\_all(unique(edx\_dat$genres), "[^|]+") %>%  
 unlist() %>%unique()  
unique\_genres\_list

## [1] "Comedy" "Romance" "Action"   
## [4] "Crime" "Thriller" "Drama"   
## [7] "Sci-Fi" "Adventure" "Children"   
## [10] "Fantasy" "War" "Animation"   
## [13] "Musical" "Western" "Mystery"   
## [16] "Film-Noir" "Horror" "Documentary"   
## [19] "IMAX" "(no genres listed)"

**2. Data Analysis**

options(digits = 4)  
*# The RMSE function that will be used in this project is:*  
RMSE <- **function**(true\_ratings = NULL, predicted\_ratings = NULL) {  
 sqrt(mean((true\_ratings - predicted\_ratings)^2))  
}

**— 1. Native Mean Avg Model —**

*# Calculate the average of all movies*  
mu\_hat <- mean(edx\_dat$rating)  
mu\_hat *#3.512*

## [1] 3.512

*# Predict the RMSE on the validation set*  
rmse\_mean <- RMSE(validation\_dat$rating, mu\_hat) *#1.06065*  
  
*# Creating a results dataframe*  
results <- data.frame(model="Naive Mean Avg Model", RMSE=rmse\_mean)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.061

**— 2. Movie Avg Model —**

*# Calculate the average by movie*  
movie\_avgs <- edx\_dat %>%  
 group\_by(movieId) %>%  
 summarize(b\_i = mean(rating - mu\_hat))  
  
*# Compute the predicted ratings on validation dataset*  
rmse\_movie\_model <- validation\_dat %>%  
 left\_join(movie\_avgs, by='movieId') %>%  
 mutate(pred = mu\_hat + b\_i) %>%  
 pull(pred)  
  
rmse\_movie\_model\_result <- RMSE(validation\_dat$rating, rmse\_movie\_model)  
  
*# Adding row to the results*  
results <- results %>% add\_row(model="Movie-Based Model", RMSE=rmse\_movie\_model\_result)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.0607  
## 2 Movie-Based Model 0.9437

**— 3. Movie User Avg Model —**

*# Calculate the average by user*  
user\_avgs <- edx\_dat %>%  
 left\_join(movie\_avgs, by='movieId') %>%  
 group\_by(userId) %>%  
 summarize(b\_u = mean(rating - mu\_hat - b\_i))  
  
*# Compute the predicted ratings on validation dataset*  
rmse\_movie\_user\_model <- validation\_dat %>%  
 left\_join(movie\_avgs, by='movieId') %>%  
 left\_join(user\_avgs, by='userId') %>%  
 mutate(pred = mu\_hat + b\_i + b\_u) %>%  
 pull(pred)  
  
rmse\_movie\_user\_model\_result <- RMSE(validation\_dat$rating, rmse\_movie\_user\_model)  
  
*# Adding row to the results*  
results <- results %>% add\_row(model="Movie+User Based Model", RMSE=rmse\_movie\_user\_model\_result)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.0607  
## 2 Movie-Based Model 0.9437  
## 3 Movie+User Based Model 0.8655

**— 4 Movie User Genre Avg Model —**

*# calculate genre bias*  
genre\_avgs <- edx\_dat %>%  
 left\_join(movie\_avgs, by='movieId') %>%  
 left\_join(user\_avgs, by='userId') %>%  
 group\_by(genres) %>%  
 summarize(b\_u\_g = mean(rating - mu\_hat - b\_i - b\_u))  
  
  
*# Compute the predicted ratings on validation dataset*  
rmse\_movie\_user\_genre\_model <- validation\_dat %>%  
 left\_join(movie\_avgs, by='movieId') %>%  
 left\_join(user\_avgs, by='userId') %>%  
 left\_join(genre\_avgs, by='genres') %>%  
 mutate(pred = mu\_hat + b\_i + b\_u + b\_u\_g) %>%  
 pull(pred)  
  
rmse\_movie\_user\_genre\_model\_result <- RMSE(validation\_dat$rating,rmse\_movie\_user\_genre\_model)  
  
*# Adding row to the results*  
results <- results %>% add\_row(model="Movie+User+Genre Based Model", RMSE=rmse\_movie\_user\_genre\_model\_result)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.0607  
## 2 Movie-Based Model 0.9437  
## 3 Movie+User Based Model 0.8655  
## 4 Movie+User+Genre Based Model 0.8652

**— 5 Regularized Movie Model —**

set.seed(1)  
lambdas <- seq(0, 10, 0.1)  
  
*# Compute the predicted ratings on validation dataset using different lambda*  
rmses <- sapply(lambdas, **function**(lambda) {  
   
 *# Calculate the average by user*  
 b\_i <- edx\_dat %>%  
 group\_by(movieId) %>%  
 summarize(b\_i = sum(rating - mu\_hat) / (n() + lambda))  
   
 *# Compute the predicted ratings on validation dataset*  
 predicted\_ratings <- validation\_dat %>%  
 left\_join(b\_i, by='movieId') %>%  
 mutate(pred = mu\_hat + b\_i) %>%  
 pull(pred)  
   
 *# Predict the RMSE on the validation set*  
 return(RMSE(validation\_dat$rating, predicted\_ratings))  
})  
  
*# Get the lambda value that minimize the RMSE*  
min\_lambda <- lambdas[which.min(rmses)]  
  
*# RMSE*  
rmse\_regularized\_movie\_model <- min(rmses)  
  
*# Adding row to the results*  
results <- results %>% add\_row(model="Regularized Movie-Based Model", RMSE=rmse\_regularized\_movie\_model)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.0607  
## 2 Movie-Based Model 0.9437  
## 3 Movie+User Based Model 0.8655  
## 4 Movie+User+Genre Based Model 0.8652  
## 5 Regularized Movie-Based Model 0.9437

**— 6 Regularized Movie + User Model —**

**This step takes some minutes to run at my laptop 4G windows 10**

set.seed(1)  
  
rmses <- sapply(lambdas, **function**(lambda) {  
  
 *# Calculate the average by movie*  
 b\_i <- edx\_dat %>%  
 group\_by(movieId) %>%  
 summarize(b\_i = sum(rating - mu\_hat) / (n() + lambda))  
   
 *# Calculate the average by user*  
 b\_u <- edx\_dat %>%  
 left\_join(b\_i, by='movieId') %>%  
 group\_by(userId) %>%  
 summarize(b\_u = sum(rating - b\_i - mu\_hat) / (n() + lambda))  
   
 *# Compute the predicted ratings on validation dataset*  
 predicted\_ratings <- validation\_dat %>%  
 left\_join(b\_i, by='movieId') %>%  
 left\_join(b\_u, by='userId') %>%  
 mutate(pred = mu\_hat + b\_i + b\_u) %>%  
 pull(pred)  
   
 *# Predict the RMSE on the validation set*  
 return(RMSE(validation\_dat$rating, predicted\_ratings))  
})  
  
*# Get the lambda value that minimize the RMSE*  
min\_lambda <- lambdas[which.min(rmses)]  
  
*# Predict the RMSE on the validation set*  
rmse\_regularized\_movie\_user\_model <- min(rmses)  
  
*# Adding the results to the results dataset*  
results <- results %>% add\_row(model="Regularized Movie+User Based Model", RMSE=rmse\_regularized\_movie\_user\_model)  
print.data.frame(results)

## model RMSE  
## 1 Naive Mean Avg Model 1.0607  
## 2 Movie-Based Model 0.9437  
## 3 Movie+User Based Model 0.8655  
## 4 Movie+User+Genre Based Model 0.8652  
## 5 Regularized Movie-Based Model 0.9437  
## 6 Regularized Movie+User Based Model 0.8650

**saveRDS(results, “capstone\_results.rda”)**

**# — 7 Regularized Movie User Genres Model —**

**set.seed(1)**

**lambdas <- seq(0, 10, 0.1)**

**# Compute the predicted ratings on validation dataset using different values of lambda**

**rmses <- sapply(lambdas, function(lambda) {**

**# Calculate the average by movie**

**b\_i <- edx\_dat %>%**

**group\_by(movieId) %>%**

**summarize(b\_i = sum(rating - mu\_hat) / (n() + lambda))**

**# Calculate the average by user**

**b\_u <- edx\_dat %>%**

**left\_join(b\_i, by=‘movieId’) %>%**

**group\_by(userId) %>%**

**summarize(b\_u = sum(rating - b\_i - mu\_hat) / (n() + lambda))**

**# Calculate genre bias**

**b\_g <- edx\_dat %>%**

**left\_join(b\_i, by=‘movieId’) %>%**

**left\_join(b\_u, by=‘userId’) %>%**

**group\_by(genres) %>%**

**summarize(b\_g = sum(rating - b\_i - mu\_hat - b\_u) / (n() + lambda))**

**# Compute the predicted ratings on validation dataset**

**predicted\_ratings <- validation\_dat s%>%**

**left\_join(b\_i, by=‘movieId’) %>%**

**left\_join(b\_u, by=‘userId’) %>%**

**left\_join(b\_g, by=‘genres’) %>%**

**mutate(pred = mu\_hat + b\_i + b\_u + b\_g) %>%**

**pull(pred)**

**# Predict the RMSE on the validation set**

**return(RMSE(validation\_dat$rating, predicted\_ratings))**

**})**

**# Get the lambda value that minimize the RMSE**

**min\_lambda <- lambdas[which.min(rmses)]**

**# Predict the RMSE on the validation set**

**rmse\_regularized\_movie\_user\_genre\_model <- min(rmses)**

**# Adding the results to the results dataset**

**results <- results %>% add\_row(model=“Regularized Movie+User+Genre Based Model”, RMSE=rmse\_regularized\_movie\_user\_genre\_model)**

**print.data.frame(results)**