MovieLens

Dr.Narendrakumar Dasre

5/26/2024

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org") if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")

library(tidyverse) library(caret)

MovieLens 10M dataset:

https://grouplens.org/datasets/movielens/10m/

http://files.grouplens.org/datasets/movielens/ml-10m.zip

options(timeout = 120)

dl <- "ml-10M100K.zip" if(!file.exists(dl)) download.file("https://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)

ratings_file <- "ml-10M100K/ratings.dat" if(!file.exists(ratings_file)) unzip(dl, ratings_file)

movies_file <- "ml-10M100K/movies.dat" if(!file.exists(movies_file)) unzip(dl, movies_file)

ratings <- as.data.frame(str_split(read_lines(ratings_file), fixed("::"), simplify = TRUE), stringsAsFactors = FALSE)

colnames(ratings) <- c("userId", "movieId", "rating", "timestamp") ratings <- ratings %>%mutate(userId = as.integer(userId), movieId = as.integer(movieId), rating = as.numeric(rating), timestamp = as.integer(timestamp))

movies <- as.data.frame(str_split(read_lines(movies_file), fixed("::"), simplify = TRUE),stringsAsFactors = FALSE) colnames(movies) <- c("movieId", "title", "genres") movies <- movies %>%mutate(movieId = as.integer(movieId))

movielens <- left_join(ratings, movies, by = "movieId")

Final hold-out test set will be 10% of MovieLens data

set.seed(1, sample.kind="Rounding") # if using R 3.6 or later # set.seed(1) # if using R 3.5 or earlier test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE) edx <- movielens[-test_index,] temp <- movielens[test_index,]

Make sure userId and movieId in final hold-out test set are also in edx set

final_holdout_test <- temp %>% semi_join(edx, by = "movieId") %>% semi_join(edx, by = "userId")

Add rows removed from final hold-out test set back into edx set

removed <- anti_join(temp, final_holdout_test) edx <- rbind(edx, removed)

rm(dl, ratings, movies, test_index, temp, movielens, removed)

head(edx) %>% kable(caption = "Top rows of edx file") %>% kable_styling(font_size = 10, position = "center", latex_options = c("scale_down", "HOLD_position"))

#create summary table edx_summary <- data.frame(number_of_rows = nrow(edx), number_of_column = ncol(edx), number_of_users = n_distinct(edx$userId), number_of_movies = n_distinct(edx$movieId), average_rating = round(mean(edx$rating), 2), number_of_genres = n_distinct(edx$genres), the_first_rating_date = as.Date(as.POSIXct(min(edx$timestamp), origin = "1970−01−01")), the_last_rating_date = as.Date(as.POSIXct(max(edx$tim origin = "1970-01-01")))

edx_summary[,1:6] %>% kable(caption = "Summary of edx set (part 1)") %>% kable_styling(font_size = 10, position = "center", latex_options = c("scale_down","HOLD_position"))

edx_summary[,7:8] %>% kable(caption = "Summary of edx set (part 2)") %>% kable_styling(font_size = 10, position = "center", latex_options = "HOLD_position")

validation_summary <- data.frame(number_of_rows = nrow(validation), number_of_column = ncol(validation), number_of_users = n_distinct(validation$userId), number_of_movies = n_distinct(validation$movieId), average_rating = mean(validation$rating), number_of_genres = n_distinct(validation$genres), the_first_rating_date = as.Date(as.POSIXct(min(validation$timestamp), origin = "1970 − 01 − 01")), the_last_rating_date = as.Date(as.POSIXct(max(validation$timestamp), origin = "1970-01-01")))

#create summary table validation_summary[,1:6] %>% kable(caption = "Summary of validation set (part 1)", digits = 2) %>% kable_styling(font_size = 10, position = "center", latex_options = c("scale_down", "HOLD_position"))

validation_summary[,7:8] %>% kable(caption = "Summary of validation set (part 2)") %>% kable_styling(font_size = 10, position = "center", latex_options = "HOLD_position")

#create a summary table grouping by rating

rating_sum <- edx %>% group_by(rating) %>% summarize(count = n())

gg <- rating_sum %>% mutate(rating = factor(rating)) %>% ggplot(aes(rating, count)) + geom_col(fill = "steel blue", color = "white") + theme_classic() + labs(x = "Rating", y = "Count", title = "Number of rating", caption = "Figure 1 - Rating in edx dataset") ggplotly(gg)

gg <- rating_sum %>% mutate(rating = factor(rating)) %>% ggplot(aes(rating, count)) + geom_col(fill = "steel blue", color = "white") + theme_classic() + labs(x = "Rating", y = "Count", title = "Number of rating", caption = "Figure 1 - Rating in edx dataset") ggplotly(gg)

#create summary table grouping by movieId

movie_sum <- edx %>% group_by(movieId) %>% summarize(n_rating_of_movie = n(), mu_movie = mean(rating), sd_movie = sd(rating))

#create figure of number of rating

gg <- movie_sum %>% ggplot(aes(n_rating_of_movie)) + geom_density(fill = "gold1") + labs(title = "Density plot - number of rating", x = "number of rating", y = "density", caption = "Figure 3 - The long tail number of rating") + geom_vline(aes(xintercept = mean(movie_sum$rating_of_movie)), color = "red") + annotate("text", x = 2000, y = 0.0022, label = print(round(mean(movie_sum$n_rating_of_movie),0)), color = "red", size = 3) + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none") ggplotly(gg)

gg <- movie_sum %>% ggplot(aes(n_rating_of_movie, mu_movie)) + geom_point(color = "steel blue", alpha = 0.3) + geom_smooth()+ geom_vline(aes(xintercept = mean(movie_sum$n_rating_of_movie)), color = "red") + annotate("text", x = 2000, y = 5, label = print(round(mean(movie_sum$n_rating_of_movie),0)),

color = “red”, size = 3) + theme_classic() + labs(title = “Scatter plot - Average rating vs number of rating”, x = “Number of rating / movie”, y = “Average rating”, caption = “Figure 4”) + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12))

ggplotly(gg)

subplot(ggplotly(movie_sum %>% ggplot(aes(mu_movie)) + geom_histogram(fill = “steel blue”, color = “black”, binwidth = 0.5) + labs(title = “Distribution of movie’s average rating”, x = “rating”, y = “count”, caption = “Figure 5”) + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12))),

ggplotly(rating_sum %>% ggplot(aes(x = rating, y = count)) + geom_col(fill = “grey”, color = “black”) + labs(title = “Distribution of true rating”, x = “rating”, y = “count”, caption = “Figure 6”) + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12))),

nrows = 1)

gg <- movie_sum %>% mutate(group = cut(n_rating_of_movie, breaks = c(-Inf, mean(n_rating_of_movie),Inf), labels = c(“n < 843”, “n > 843”))) %>% ggplot(aes(sd_movie, fill = group)) + geom_density(alpha = 0.5) + labs(title = “Standard deviation of rating”, x = “Standard deviation”, y = “count”, caption = “Figure 7 - N < 843 number of rating less than average, N > 843 number of rating greater than average”) + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12))

ggplotly(gg)

#create summary table grouping by userId

user_sum <- edx %>% group_by(userId) %>% summarize(n_user_rated = n(), mu_user = mean(rating), sd_user = sd(rating))

#create figure of number of rating

gg <- user_sum %>% ggplot(aes(n_user_rated)) + geom_density(fill = “steel blue”, alpha = 0.8) + labs(title = “Density plot - number of user rated”, x = “number of rating”, y = “density”, caption = “Figure 8”) + geom_vline(aes(xintercept = mean(user_sumn_user_rated), color = "red") + annotate("text", x = 400, y = 0.009, label = print(round(mean(user_sumn_user_rated),0)), color = “red”, size = 3) + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = “none”)

ggplotly(gg)

#top 10 users have highest number of rating

user_sum %>% arrange(desc(n_user_rated)) %>% head(10) %>% kable(caption = “Top 10 users with highest number of rating given”, digits = 2) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

#top 10 users have lowest number of rating

user_sum %>% arrange(n_user_rated) %>% head(10) %>% kable(caption = “Top 10 users with lowest number of rating given”, digits = 2) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

gg <- user_sum %>% ggplot(aes(n_user_rated, mu_user)) + geom_point(color = “steel blue”, alpha = 0.3) + geom_smooth() + theme_classic() + labs(title = “Scatter plot - number of rating user given vs average rating”, x = “number of rating user given”, y = “average rating”, caption = “Figure 9”) + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = “none”)

ggplotly(gg)

gg <- user_sum %>% mutate(group = cut(n_user_rated, breaks = c(-Inf, mean(n_user_rated), Inf), label = c("< 129", ">129"))) %>% ggplot(aes(sd_user, fill = group)) + geom_density(alpha = 0.5) + labs(title = "Standard deviation of rating by user", x = "Standard deviation", y = "count", caption = "Figure 10") + theme_classic() + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none")

ggplotly(gg)

edx <- edx %>% mutate(rating_time = as.Date(as.POSIXct(timestamp, origin = "1970-01-01"))) %>% mutate(rating_year = year(rating_time))

edx <- edx %>% mutate(release_year = as.integer(substr(title, str_length(title) - 4, str_length(title) - 1))) release_year_sum <- edx %>% group_by(release_year) %>% summarize(n = n(), average_rating = mean(rating))

subplot(ggplot(release_year_sum, aes(release_year, n)) + geom_point(color = "steel blue", alpha = 0.6) + geom_line(color = "steel blue") + theme_classic() + labs(title = "number of movies by release year", caption = "Figure 11") + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none"),

release_year_sum %>% ggplot(aes(release_year, average_rating)) + geom_point(color = "steel blue", alpha = 0.6) + theme_classic() + geom_smooth() + labs(title = "average rating by release year", caption = "Figure 12") + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none"),

nrows = 1)

fit_lm <- lm(average_rating ~ I(release_year^3) + I(release_year^2) + I(release_year), data = release_year_sum) summary(fit_lm)

#calculate the first rating time of each movie movie_sum <- edx %>% group_by(movieId) %>% summarize(n_rating_of_movie = n(), mu_movie = mean(rating), first_rating_time = min(timestamp))

#calculate the aging time edx <- edx %>% left_join(movie_sum, by = "movieId") edx <- edx %>% mutate(aging_time = round((timestamp - first_rating_time)/60/60/24/30,0))

#create a summary table grouping by aging time aging_time_sum <- edx %>% group_by(aging_time) %>% summarize(n_aging_time = n(), average_rating = mean(rating))

#visualize by ggplot subplot(ggplot(aging_time_sum, aes(aging_time, n_aging_time)) + geom_point(color = "steel blue") + geom_line(color = "steel blue") + theme_classic() + labs(title = "number of rating per aging time", x = "aging time (month)", y = "count", caption = "Figure 13") + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none"),

```
          ggplot(aging_time_sum, aes(aging_time, average_rating)) +
           geom_point(color = "steel blue") +
           geom_line(color = "steel blue") +
           theme_classic() +
           labs(title = "average rating per aging time",
                x = "aging time (month)",
                y = "average rating",
                caption = "Figure 14") +
    theme(axis.title.x = element_text(size = 10),
  axis.title.y = element_text(size = 10),
  plot.title = element_text(size = 12),
  legend.position = "none"),

          nrows = 2)
```

#create a vector of genres genres <- str_replace(edx$genres,"\|.*","") genres <- genres[!duplicated(genres)] genres

#calculate the number of movies per each genres n_genres <- sapply(genres, function(ge){ index <- str_which(edx$genres, ge)length(edx$rating[index])

})

#calculate the average rating of each genres genres_rating <- sapply(genres, function(ge){ index <- str_which(edx$genres, ge)mean(edx$rating[index], na.rm = T) })

#create a summary data by genres genres_sum <- data.frame(genres = genres, n_genres = n_genres, average_rating = genres_rating)

#print out the summary table by genres genres_sum %>% arrange(desc(n_genres)) %>% head %>% kable(caption = "Summary table by genres", digits = 2) %>% kable_styling(font_size = 10, position = "center", latex_options = "HOLD_position")

subplot(

#ranking genres by number of each appear in the edx data set genres_sum %>% mutate(top5 = ifelse(genres %in% c("Comedy","Drama","Action","Thriller","Adventure"), "top5","non")) %>% ggplot(aes(x = reorder(genres, n_genres), n_genres, fill = top5)) + geom_col(color = "white") + theme_classic() + coord_flip() + labs(title = "number of movie by genres", y = "number of rating", x = "genres", caption = "Figure 15") + scale_fill_manual(values = c("grey","steel blue")) + theme(legend.position = "none"),

comparing average rating of each genres in edx data set

ggplot(genres_sum, aes(x = reorder(genres, average_rating), average_rating)) + geom_col(fill = "steel blue", color = "white") + theme_classic() + coord_flip() + labs(title = "average rating by genres", y = "average rating", x = "genres", caption = "Figure 16"),

nrows = 2)

gg <- edx %>% group_by(genres) %>% summarize(count = n(), rating = mean(rating)) %>% ggplot(aes(x = reorder(genres, rating), rating)) + geom_col(fill = "steel blue") + theme(axis.ticks.x = element_blank(), axis.text.x = element_blank(), axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12)) + labs(x = "multiple genres", y = "average rating", title = "average rating by genres in edx data", caption = "Figure 17")

ggplotly(gg)

#create list average users avg_user_list <- user_sum %>% filter(n_user_rated >= round(mean(n_user_rated),2)-1, n_user_rated <= round(mean(n_user_rated),2)+1) %>% select(userId, mu_user)

#select randomly 4 users set.seed(1, sample.kind = "Rounding")

avg_user_list <- sample(avg_user_list$userId, 10) avg_user_list

#create figure of rating change by genres of average users gg <- edx %>% filter(userId %in% avg_user_list) %>% group_by(genres) %>% summarize(rating = mean(rating), count = n()) %>% ggplot(aes(reorder(genres, rating), rating))+ geom_col(fill = "steel blue") + theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) + labs(x = "genres", title = "Rating change by genres", caption = "Figure 18") + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12), legend.position = "none")

ggplotly(gg)

model_1_movie <- RMSE(edxmu_movie, edxrating)

rmse_results <- data_frame(method="Only baseline is movie average",
RMSE = model_1_movie)

rmse_results %>% kable(caption = “RMSE by method”, digits = 4) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

b_j_sum <- edx %>% mutate(yhat = rating - mu_movie) %>% group_by(userId) %>% summarize(n_user_rated = n(), b_j = mean(yhat))

subplot(b_j_sum %>% ggplot(aes(b_j)) + geom_histogram(fill = “steel blue”, color = “white”) + theme_classic() + labs(title = “Distribution of user bias”, caption = “Figure 19”) + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12)),

b_j_sum %>% ggplot(aes(n_user_rated, b_j)) + geom_point(color = “steel blue”, alpha = 0.5) + theme_classic() + labs(title = “Scatter plot of user bias and number of user rated”, caption = “Figure 20”) + theme(axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10), plot.title = element_text(size = 12)),

nrows = 1)

edx <- edx %>% left_join(b_j_sum, by = “userId”) %>% mutate(mu_movie_user = mu_movie + b_j)

model_2_movie_user <- RMSE(edxmu_movie_user, edxrating)

rmse_results <- bind_rows(rmse_results, data_frame(method=“Add Specific-effect of user”,
RMSE = model_2_movie_user))

rmse_results %>% kable(caption = “RMSE by method”, digits = 4) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

b_time_sum <- edx %>% mutate(error = rating - mu_movie_user) %>% group_by(aging_time) %>% summarize(b_time = mean(error))

subplot(edx %>% ggplot(aes(mu_movie_user))+ geom_histogram(fill = “steel blue”, color = “white”) + theme_classic() + labs(title = “predicted rating, movie - user effect”, caption = “Figure 20”),

b_time_sum %>% ggplot(aes(b_time)) + geom_histogram(fill = “steel blue”, color = “white”) + theme_classic() + labs(title = “time effect parameter”, caption = “Figure 21”),

nrows = 1)

#calculate predicted rating edx <- edx %>% left_join(b_time_sum, by = “aging_time”) edx$b_time[is.na(edx$b_time)] <- 0

edx <- edx %>% mutate(mu_movie_user_time = mu_movie_user + b_time)

model_3_movie_user_time <- RMSE(edx$mu_movie_user_time, edx$rating)

rmse_results <- bind_rows(rmse_results, data_frame(method=“Movie - User - Time Effect Model”,
RMSE = model_3_movie_user_time))

rmse_results %>% kable(caption = “RMSE by method”, digits = 4) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

#calculate the genres effect bias b_genres_sum <- edx %>% mutate(error = rating - mu_movie_user_time) %>% group_by(genres) %>% summarize(b_genres = mean(error))

edx <- edx %>% left_join(b_genres_sum, by = “genres”)

edx <- edx %>% mutate(mu_movie_user_time_genres = mu_movie_user_time + b_genres)

model_4_movie_user_time_genres <- RMSE(edx$mu_movie_user_time_genres, edx$rating)

rmse_results <- bind_rows(rmse_results, data_frame(method=“Movie - User - Time - Genres Effect Model”,
RMSE = model_4_movie_user_time_genres))

data.frame(rmse_results) %>% kable(caption = “RMSE by different method”, digits = 4) %>% kable_styling(font_size = 10, position = “center”, latex_options = “HOLD_position”)

#calculate the rating time validation <- validation %>% mutate(rating_time = as.Date(as.POSIXct(timestamp, origin = "1970-01-01"))) %>% mutate(rating_year = year(rating_time))

#calculate the aging time validation <- validation %>% left_join(movie_sum, by = "movieId") validation <- validation %>% mutate(aging_time = round((timestamp - first_rating_time)/60/60/24/30,0))

validation <- validation %>% left_join(b_j_sum, by = "userId") %>% left_join(b_time_sum, by = "aging_time") %>% left_join(b_genres_sum, by = "genres")

kable(data.frame(n_NA = colSums(is.na(validation[,14:16]))), caption = "NA value check in validation set", digits = 0) %>% kable_styling(font_size = 10, position = "center", latex_options = "HOLD_position")

validation$b_time[is.na(validation$b_time)] <- mean(validation$b_time, na.rm = T)

validation <- validation %>% mutate(predicted_rating = mu_movie + b_j + b_time + b_genres)

RMSE(validation$rating, validation$predicted_rating)

subplot(validation %>% ggplot(aes(rating)) + geom_histogram(fill = "grey", color = "black") + theme_classic() + labs(title = "true rating", caption = "Figure 22"),

validation %>% ggplot(aes(predicted_rating)) + geom_histogram(binwidth = 0.5, fill = "steel blue", color = "white") + theme_classic() + labs(title = "predicted rating", caption = "Figure 23"),

validation %>% ggplot() + geom_density(aes(rating), fill = "grey", alpha = 0.7) + geom_density(aes(predicted_rating), fill = "steel blue", alpha = 0.7) + theme_classic() + labs(title = "density plot of true rating and predicted rating", caption = "Figure 24"),

nrows = 2)

gg <- validation %>% group_by(rating) %>% summarize(n = n(), rmse = round(RMSE(rating, predicted_rating),4)) %>% ggplot(aes(rating, rmse)) + geom_line(color = "grey") + geom_point(aes(size = n/10000),color = "steel blue") + theme_classic() + labs(title = "RMSE compared to true rating scale", caption = "Figure 25")

ggplotly(gg)

validation %>% mutate(group_movie = ifelse(n_rating_of_movie > 842.5,"m > 842","m < 842"), group_user = ifelse(n_user_rated > 128.5, "u > 128","u < 128")) %>% group_by(group_user, group_movie) %>% summarize(rmse = RMSE(rating, predicted_rating), count = n(), percent = 100*round(n()/nrow(validation),3)) %>% kable(caption = "RMSE by group of movies and users",digits = 4) %>% kable_styling(font_size = 10, position = "center", latex_options = "HOLD_position") %>% footnote(general = c("u = 128: average number of rating given by a user", "m = 842: average number of rating per movie"), number = c("u < 128: number of user rated less than 128", "u > 128: number of user rated greater than 128", "m < 843: number of rating per movie less than 843", "m > 843: number of rating per movie greater than 843"))