title: “MovieLens Project” author: “Hadi Yolasigmaz” date: “3/10/2019” output: pdf_document: default html_document: df_print: paged

1.Introduction

This is a r project. It is prepared to dictate all steps from importing dataset to conclusion to show ourself and our online course instructors that how we would focus any project was understood .

Project

By using movilens dataset, prepare a movie recommandation system. Why can we need these kind of systems?

My pointview, HAPPINESS. If it makes happy you, products or services you likes (make you happy) can be sold easily. Buyers continue to buy their produts and services. Also cost of purchased products and services will be minimizing by continuing more sensitive data analysis.

This is an endless way but at the same time the most dangerous way for humanity.

To be a reason for losing their variety in thought because it will be concentrating the mean and small standart deviations. Outliners will be decreasing and may be zero. It is well in Healthy sector but imagination may be killed. It should be thought but we are going to focus our project for now.

the goal of the project

To get a RMSE below 0.87750 for given dataset is the goal of the project.

Use Dataset given by ‘Create Test and Validation Sets’ information updated 1/18/2019.

1.Required packages will be installed.

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

```
## Loading required package: tidyverse

## -- Attaching packages ------------------------------------------------------------------------------

## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

2.From Movielens Website, Movielens 10M.zip ,ratings.dat and movies.dat files will be downloaded and with some data prepareation r code movielens dataset is created.

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

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

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(levels(movieId))[movieId],
                                           title = as.character(title),
                                           genres = as.character(genres))

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

3.Validation set will be 10% of MovieLens data

```
set.seed(1)
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]
```

4.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")
```

5.Add rows from validation set that users and movies not in training set, edx, back into edx set to have at least a value for every user and every movie in validation set.

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

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

6.Now we only have edx and validatation tables to use as train set and test set, other temperary objects are removed.

Data is ready to analysis. How is their structure? First data should be understood.

```
str(edx)
```

```
## 'data.frame':    9000047 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ac
```

9000055 observations and 6 columns. Column names are userId, movieId, rating, timestamp, title and genres.

Analysis

The main aim is to get a RMSE below 0.87750. It will be tried to reach by applying many model. To store the results of these models, a table will be created.

RMSE is the residual mean squared error, sqrt(mean((true_ratings - predicted_ratings)^2))

Is a rating for a movie by a user known by prediction? No for human but smaller RMSE will say a better prediction is done.

It is better to view summary of edx to start.

```
summary(edx)
```

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4122   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000047     Length:9000047
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

No NA data is observed for numeric columns. Predictions will be done on rating column.

It is better to start analysis from simple

```
If we were using mean of all rating without any rules declared, what will be RMSE
```

```
average <- mean(edx$rating)
average
```

```
## [1] 3.512466
```

```
first_rmse <- RMSE(validation$rating,average)
rmse_results_table <- data_frame(method = "First - only all data average", RMSE = first_rmse)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
rmse_results_table
```

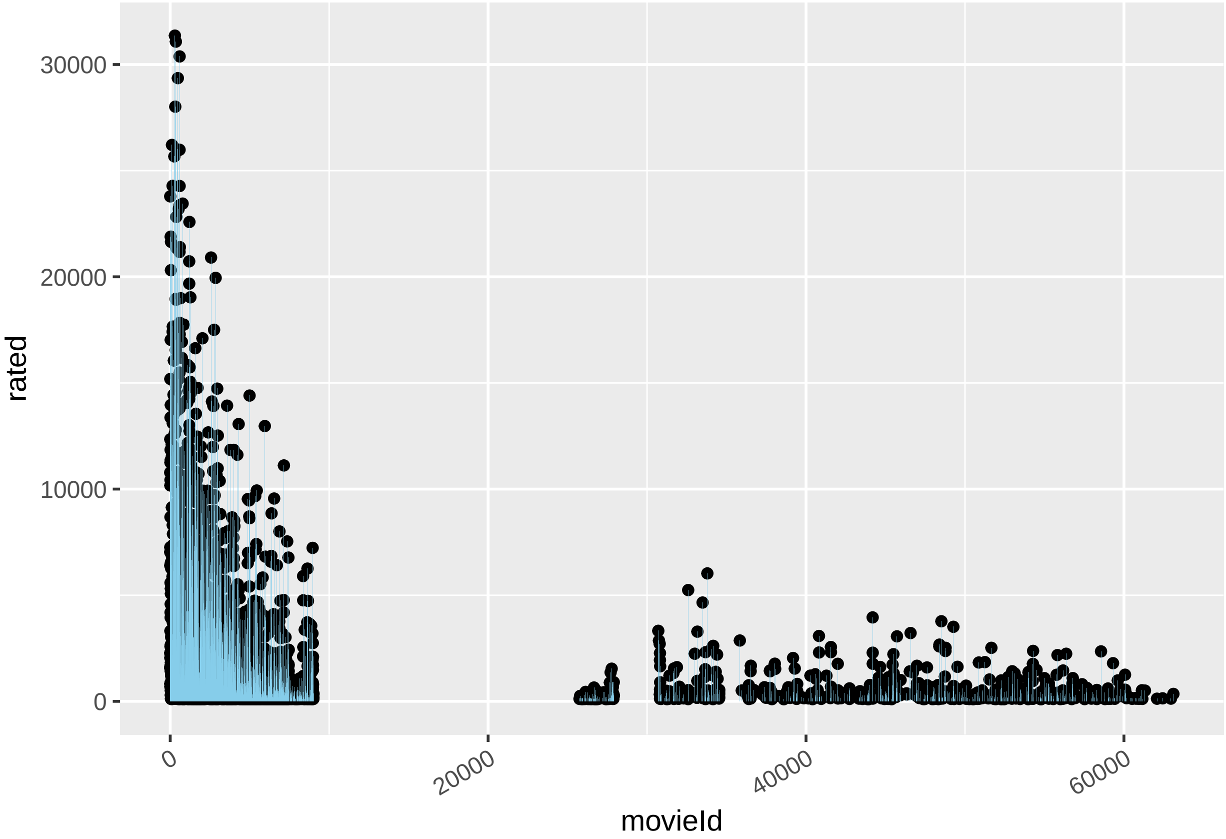
```
## # A tibble: 1 x 2
##   method                         RMSE
##   <chr>                         <dbl>
## 1 First - only all data average  1.06
```

First rmse is 1.06. It is far from 0.87750.

Movie Recommandation System should start by analysing the movie

First movie and how many times they were rated graph.

```
edx %>% select(movieId,rating) %>%
  mutate(x=1) %>%
  group_by(movieId) %>%
  summarize( rated=sum(x)) %>%
  filter (rated>100 )%>%
  ggplot (aes(x=movieId, y=rated)) +
  geom_point() +
  geom_bar(stat="identity", fill="skyblue", alpha=0.7) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



It is showing that some ranges for movies are selected for this analysis. There are some movies rated more even more than 30.000 times.

A new column, gb_movie, group by movie will be added by calculating average rating of this movie and saved.

If we use this new average as an additive to predicting rating. What will be last RMSE.

```
gb_movie_avgs <- edx %>% group_by(movieId) %>% summarize(gb_movie=mean(rating-average))
gb_movie_avgs
```

```
## # A tibble: 10,669 x 2
##    movieId gb_movie
##      <dbl>    <dbl>
##  1       1   0.415
##  2       2  -0.307
##  3       3  -0.365
##  4       4  -0.648
##  5       5  -0.444
##  6       6   0.303
##  7       7  -0.154
##  8       8  -0.378
##  9       9  -0.515
## 10      10  -0.0866
## # ... with 10,659 more rows
```

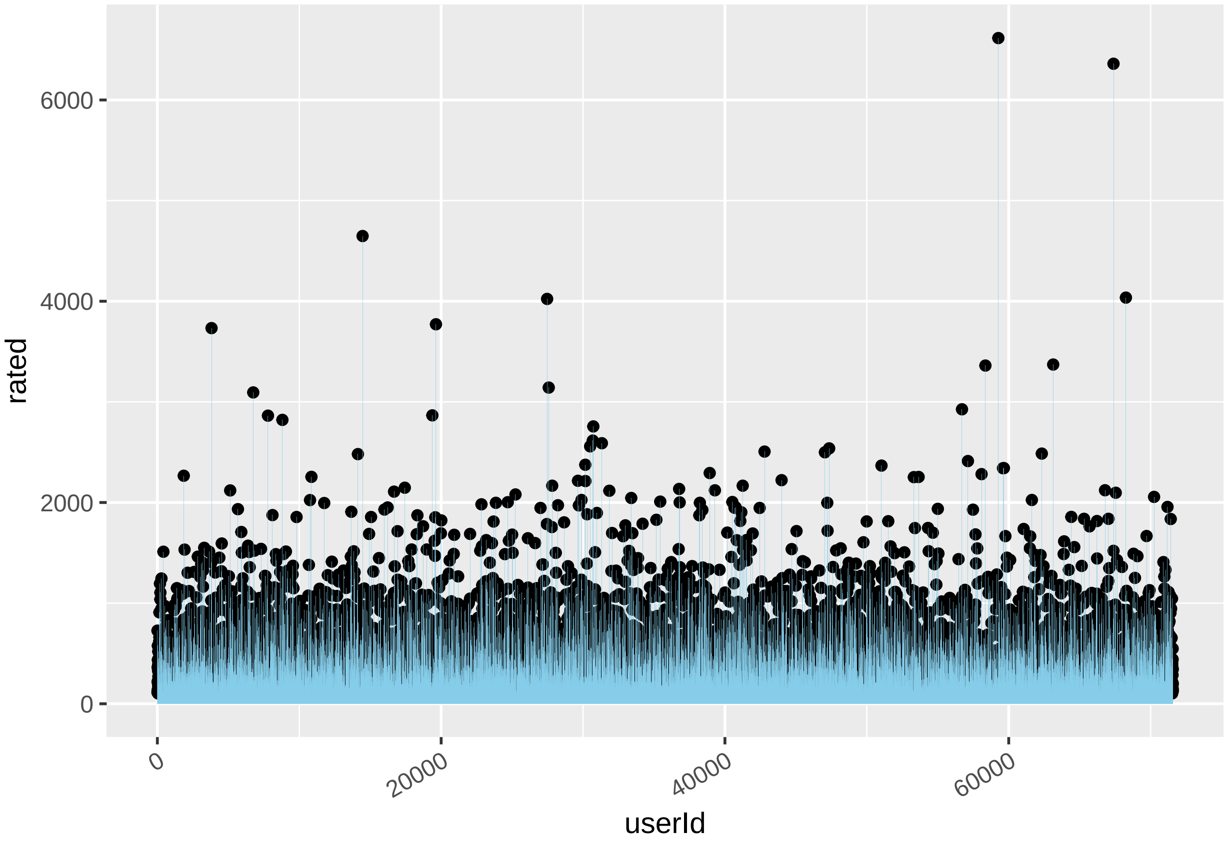
```
#predict
movie_predicted_ratings <- average + validation %>%
  left_join(gb_movie_avgs, by='movieId') %>%
  pull(gb_movie)
#predicted_ratings
gb_movie_rmse <- RMSE(movie_predicted_ratings, validation$rating)
rmse_results_table <- bind_rows(rmse_results_table, data_frame(method="Group by Movie",    RMSE = gb_mo
rmse_results_table
```

```
## # A tibble: 2 x 2
##   method                         RMSE
##   <chr>                         <dbl>
## 1 First - only all data average 1.06
## 2 Group by Movie                0.944
```

0.944 is good, it is more near to 0.87750. But above .90 is only 5 points that not enough, a new rating should be tried to reach to aim.

This time, we focus User column. What will be new rating if we include user.

```
edx %>% select(userId,rating) %>%
  mutate(x=1) %>%
  group_by(userId) %>%
  summarize( rated=sum(x)) %>%
  filter (rated>100 )%>%
  ggplot (aes(x=userId, y=rated)) +
  geom_point() +
  geom_bar(stat="identity", fill="skyblue", alpha=0.7) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Users are rating movies that helping us prepare a well defined movie recommandation system.

What will new average be?

```
gb_user_avgs <- edx %>% group_by(userId) %>% summarize(gb_user=mean(rating-average))

user_predicted_ratings <- average + validation %>%
  left_join(gb_user_avgs, by='userId') %>%
  pull(gb_user)
#predicted_ratings
gb_user_rmse <- RMSE(user_predicted_ratings, validation$rating)
rmse_results_table <- bind_rows(rmse_results_table, data_frame(method="Group by User",    RMSE = gb_user
rmse_results_table
```

```
## # A tibble: 3 x 2
##   method                          RMSE
##   <chr>                          <dbl>
## 1 First - only all data average 1.06
## 2 Group by Movie                0.944
## 3 Group by User                 0.978
```

It is not made any improvement better than movieID

what happen if we continue analysing by combination of movieID and userID

```
gb_user_avgs <- edx %>%
  left_join(gb_movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(gb_user = mean(rating - average - gb_movie))


gb_predicted_ratings <- validation %>%
  left_join(gb_movie_avgs, by='movieId') %>%
  left_join(gb_user_avgs, by='userId') %>%
  mutate(pred = average + gb_movie + gb_user) %>%
  pull(pred)


gb_movie_user_RMSE <- RMSE(gb_predicted_ratings, validation$rating)
rmse_results_table <- bind_rows(rmse_results_table,
                                data_frame(method="Combination of Movie and User",  RMSE = gb_movie_use
rmse_results_table
```

```
## # A tibble: 4 x 2
##   method                         RMSE
##   <chr>                         <dbl>
## 1 First - only all data average 1.06
## 2 Group by Movie                0.944
## 3 Group by User                 0.978
## 4 Combination of Movie and User 0.865
```

0.8653488 is below 0.87750.

It is done

3.Results

```
rmse_results_table
```

```
## # A tibble: 4 x 2
##   method                         RMSE
##   <chr>                         <dbl>
## 1 First - only all data average 1.06
## 2 Group by Movie                0.944
## 3 Group by User                 0.978
## 4 Combination of Movie and User 0.865
```

Result table showing that it is done in 4 calculations but many times it is not as easy as this. Here it is studied on known conditions.

4.Conclusions.

Recommanding the movies to users by this way, users will be more satisfied means more happy. This is the reason why it is done.

For prepare a recommandation system or any system. The aim should be known. If what is needed known, focusing that problem and deciding data analysis methods and selecting the best fitting machine learning models, obtained results can be concluded which help decision makers to give their decisions.