MovieLens

Rand

5/21/2020

Introduction

This document is for the submission of MovieLens project which aim to predict different movies' ratings given by different users.

The MovieLens data set is a set of 10 milion rows with the following columns: Movie Id User Id Movie Title Timestamp, which represents the time of the rating Genres Rating. Each row represent a rating for an unique MovieId and UserId combination (a rating given by one user to one movie).

The dataset has already been partioined into an edx dataset for building the model, and a validation set to test it. The edx part has been further divided into train set and test set to avoid over-fitting. The validation set has only been use at the end to test the choosen model.

The analysis used differenet biases introduced to the mean of the dataset: Movie average rating User average rating Release year group average rating (tow groups: before 1978, after 1978) Difference between release and rating year average rating Genres average rating.

These biases, in addition to the mean of the whole rating in the edx dataset, were used to estimate linear regression model (predict furture ratings) as the sum of dataset rating mean + biases for eah observation. In addition, regularization was used and tested for different lambda values, choosing the optimum (defined as the lambda that minimizes RMSE) for the final model. The aim of regulariztion is to penalize average ratings given by small number of observations.

This model is built upon the model used in the course material and book which already demonstrated the importane of adding movie bias and user bias.

The following analysis is done from this starting point looking for additional biases to improve the results.

Analysis

To start the analysis, first I downloaded the required libraries and loaded the edx dataframe.

```
library(tidyverse)
library(caret)
library(dplyr)
library(lubridate)
load("rda/edx.rda")
```

Then, I started exploratory data analysis with the following steps:

1- examining different genres groups and their average ratings by group_by function. Top rated genres:

```
edx %>% group_by(genres) %>% summarize(count= n(), avg = mean(rating))%>%
  arrange(desc(avg)) %>% head()
```

```
## # A tibble: 6 x 3
##   genres                           count   avg
##   <chr>                            <int> <dbl>
## 1 Animation|IMAX|Sci-Fi                7  4.71
## 2 Drama|Film-Noir|Romance           2989  4.30
## 3 Action|Crime|Drama|IMAX           2353  4.30
## 4 Animation|Children|Comedy|Crime   7167  4.28
## 5 Film-Noir|Mystery                 5988  4.24
## 6 Crime|Film-Noir|Mystery           4029  4.22
```

Least rated genres:

```
edx %>% group_by(genres) %>% summarize(count= n(), avg = mean(rating))%>%
   arrange(desc(avg)) %>% tail()
```

```
## # A tibble: 6 x 3
##   genres                                 count   avg
##   <chr>                                  <int> <dbl>
## 1 Adventure|Drama|Horror|Sci-Fi|Thriller   217  1.75
## 2 Action|Drama|Horror|Sci-Fi                 4  1.75
## 3 Comedy|Film-Noir|Thriller                 21  1.64
## 4 Action|Horror|Mystery|Thriller           327  1.61
## 5 Action|Animation|Comedy|Horror             2  1.5
## 6 Documentary|Horror                       619  1.45
```

By examining the head and the tail, we notice that some genres are significantly rated more than the others so I decided to include genres bias in the final model.

2- I retrieved the release year form the titel, the rating year from the timestamp, and computed the difference between them.Then, I added a column for each of them to the edx data set (relyear, ratyear, yeardif respectively):

```
edx <- edx%>%
  mutate(relyear=as.numeric(str_match(title, "\\((\\d{4})\\)$")[,2]),
         ratyear=year(as.Date(as.POSIXct(timestamp,origin="1970-01-01"))),
         yeardif = ratyear-relyear)
```

3- I plotted the release year against its average rating to examine the relationship between them keeping only years with more than 1000 rating to reduce noise.

From the plot, it appears that there are two groups: films that have been released before 1978 has slightly higher ratings than those released after. but there is no pattern inside each group. Based on that I added a column to differentiate films released before 1978 and after it to include in in the final model. (year group bias).

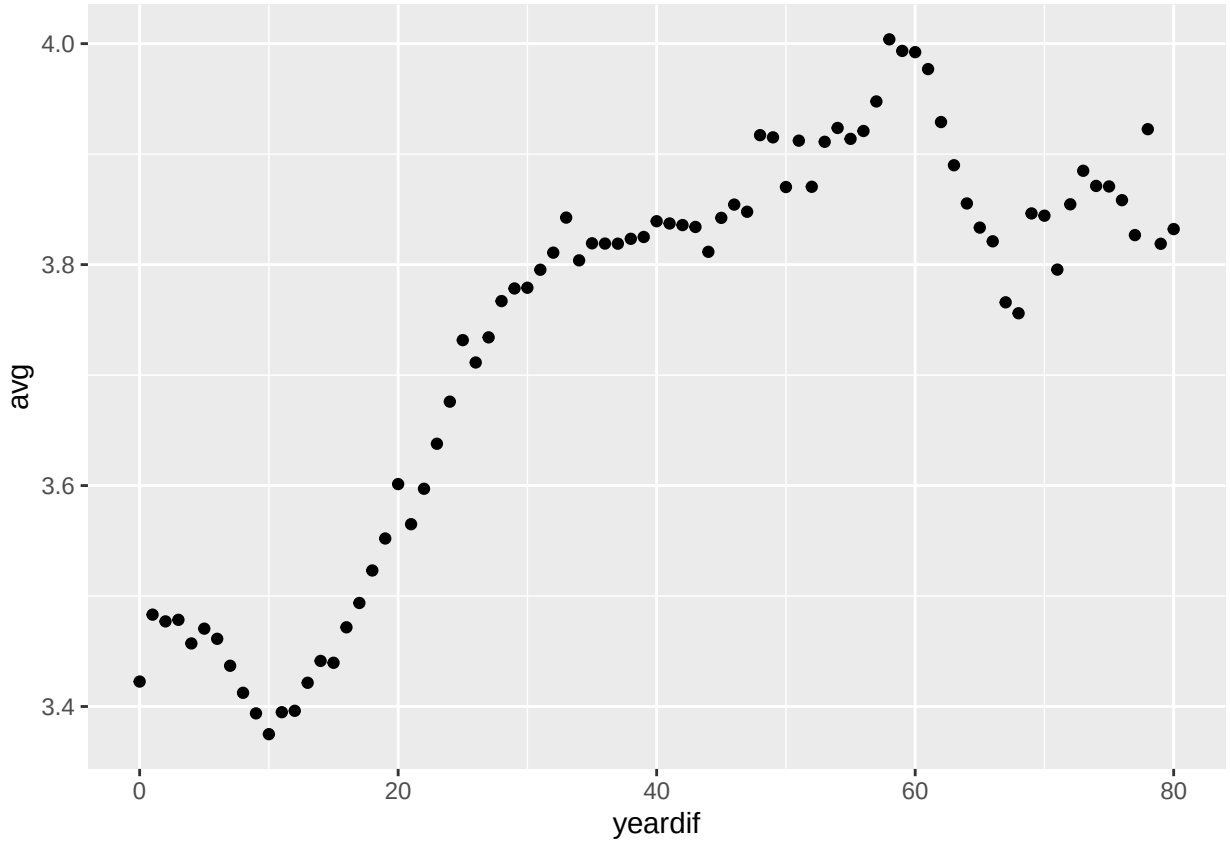
```
edx <-edx %>%  mutate(ygroup = ifelse(relyear<1978, 1, 0))
```

Claculating the mean of each group to approve the conclusion

```
edx %>% group_by(ygroup) %>% summarize(avg = mean(rating))
```

```
## # A tibble: 2 x 2
##   ygroup   avg
##    <dbl> <dbl>
## 1      0  3.46
## 2      1  3.85
```

4- I plotted difference beween release and rating years against their average ratings, keeping only groups with more than 1000 rating to reduce noise.



Different patterns appear in the plot, before 10 there was a negative relationship. After 10 there is a positive relationship. Given that the relationship is not random I decided to include year difference bias in the analysis.

Training the model: 1- First I selected the chosen columns (movieId, userId, ygroup, yeardif, genres, rating).

```
edx <- edx %>% select(movieId, userId, ygroup, yeardif, genres, rating)
```

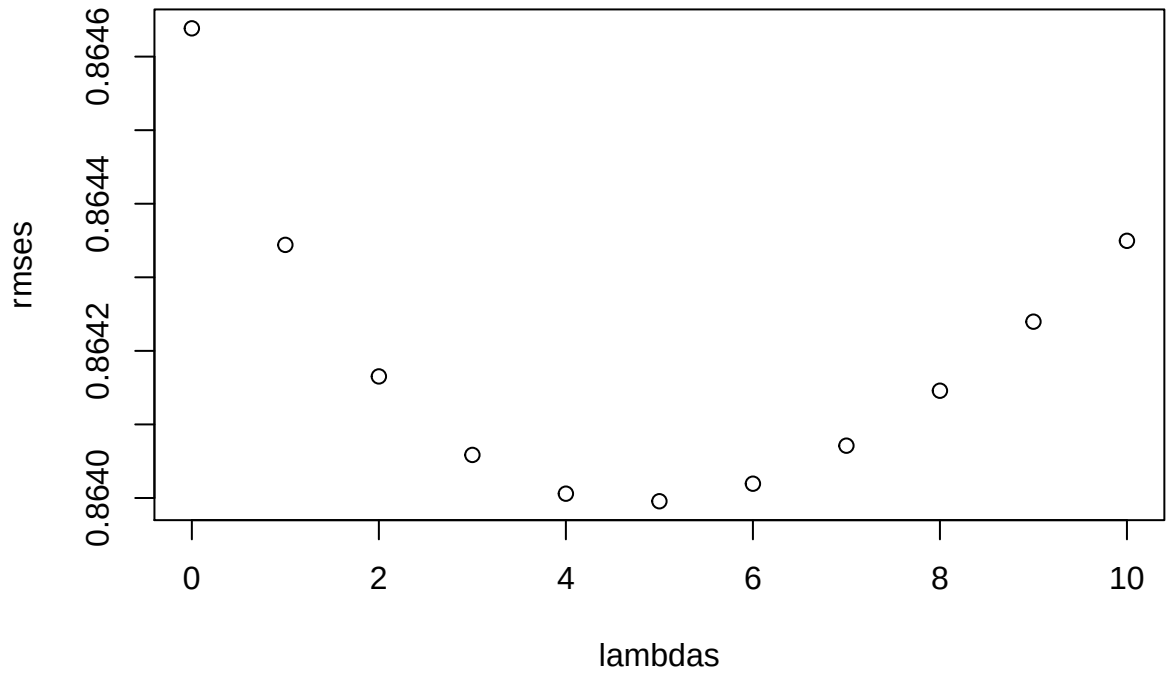
2- Then I set the seed (1999) to make te results reproducable, and partiotioned the edx dataset into training set (80%) and test set (20%), making sure to not have userIds or movieIds in the test set that are not present in the training set.

```
suppressWarnings(set.seed(1999, sample.kind = "Rounding"))
test_index <- createDataPartition(y = edx$rating, times = 1, p = 0.2, list = FALSE)
train_set <- edx[-test_index,]
test_set <- edx[test_index,]
test_set <- test_set %>%
  semi_join(train_set, by = "movieId") %>%
  semi_join(train_set, by = "userId")
```

3- I built the regression model with regularization, trained it using train set and tested it on test set from edx with different lambda values. Calculating the RMSE for each lambda. Biases used are: MovieId (b_i), UserId (b_u), Genres (b_g), Difference between release and rating year (b_yd), Release year group (b_yg). The model start by calulating the mean of the ratings in the train data set (mu), then calculate biases subsequently by the following equation: sum(rating- mu-(sum of previously introduced bias))/ (number of observation in each group +lambda). Finally, calculating predictions as (mu + b_i + b_u+b_g+b_yd+b_yg).

```
lambdas <- seq(0, 10, 1)

rmses <- sapply(lambdas, function(l){

  mu <- mean(train_set$rating)

  b_i <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  b_u <- train_set %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+l))


  b_g <- train_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by="userId") %>%
    group_by(genres) %>%
    summarize(b_g = sum(rating - b_i -b_u - mu)/(n()+l))

  b_yd <- train_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    left_join(b_g, by="genres") %>%
    group_by(yeardif) %>%
    summarize(b_yd = sum(rating - b_i -b_u-b_g - mu)/(n()+l))
  b_yg <- train_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    left_join(b_g, by = "genres") %>%
    left_join(b_yd, by="yeardif") %>%
    group_by(ygroup) %>%
    summarize(b_yg = sum(rating - b_i -b_u-b_g -b_yd- mu)/(n()+l))
  predicted_ratings <-
    test_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    left_join(b_g, by = "genres") %>%
    left_join(b_yd, by = "yeardif")%>%
    left_join(b_yg, by="ygroup") %>%
    mutate(pred = mu + b_i + b_u+b_g+b_yd+b_yg) %>%
    pull(pred)

  return(RMSE(predicted_ratings, test_set$rating))
})
```

4- I plotted lambda values against RMSEs.



Results

The lambda that minimized the RMSE is 5 with RMSE = 0.8639

```
min(rmses)
```

```
## [1] 0.8639956
```

```
bestl = lambdas[which.min(rmses)]
bestl
```

```
## [1] 5
```

Now I retrained the final model using the whole edx data. I calculated the rating mean of the whole edx dataset and recalculated the biases using this mean and lambda 5.

```
bestl <- 5
mu <- mean(edx$rating)
b_i <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+bestl))

b_u <- edx %>%
```

```
  left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu)/(n()+bestl))


b_g <- edx %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by="userId") %>%
  group_by(genres) %>%
  summarize(b_g = sum(rating - b_i -b_u - mu)/(n()+bestl))

b_yd <- edx %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  left_join(b_g, by="genres") %>%
  group_by(yeardif) %>%
  summarize(b_yd = sum(rating - b_i -b_u-b_g - mu)/(n()+bestl))
b_yg <- edx %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  left_join(b_g, by = "genres") %>%
  left_join(b_yd, by="yeardif") %>%
  group_by(ygroup) %>%
  summarize(b_yg = sum(rating - b_i -b_u-b_g -b_yd- mu)/(n()+bestl))
```

Here, I loaded the validation set, added the additional columns (relyear, ratyear, ydif, ygroup) and tested the model on it.

```
load("rda/validation.rda")
validation <-validation%>% mutate(relyear=as.numeric(str_match(title,"\\((\\d{4})\\)$")[,2]),
                                  ratyear = year(as.Date(as.POSIXct(timestamp, origin="1970-01-01"))),
                                  yeardif = ratyear-relyear)
validation <-validation %>%  mutate(ygroup = ifelse(relyear<1978, 1, 0))
validation <- validation %>% select(movieId, userId, ygroup, yeardif, genres, rating)
predicted_ratings <-
  validation %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  left_join(b_g, by = "genres") %>%
  left_join(b_yd, by = "yeardif")%>%
  left_join(b_yg, by="ygroup") %>%
  mutate(pred = mu + b_i + b_u+b_g+b_yd+b_yg) %>%
  pull(pred)
```

The RMSE is 0.86401

```
RMSE(predicted_ratings, validation$rating)
```

```
## [1] 0.8640157
```

The model produced the required RMSE (<0.8649) in a reasonable amount of time.

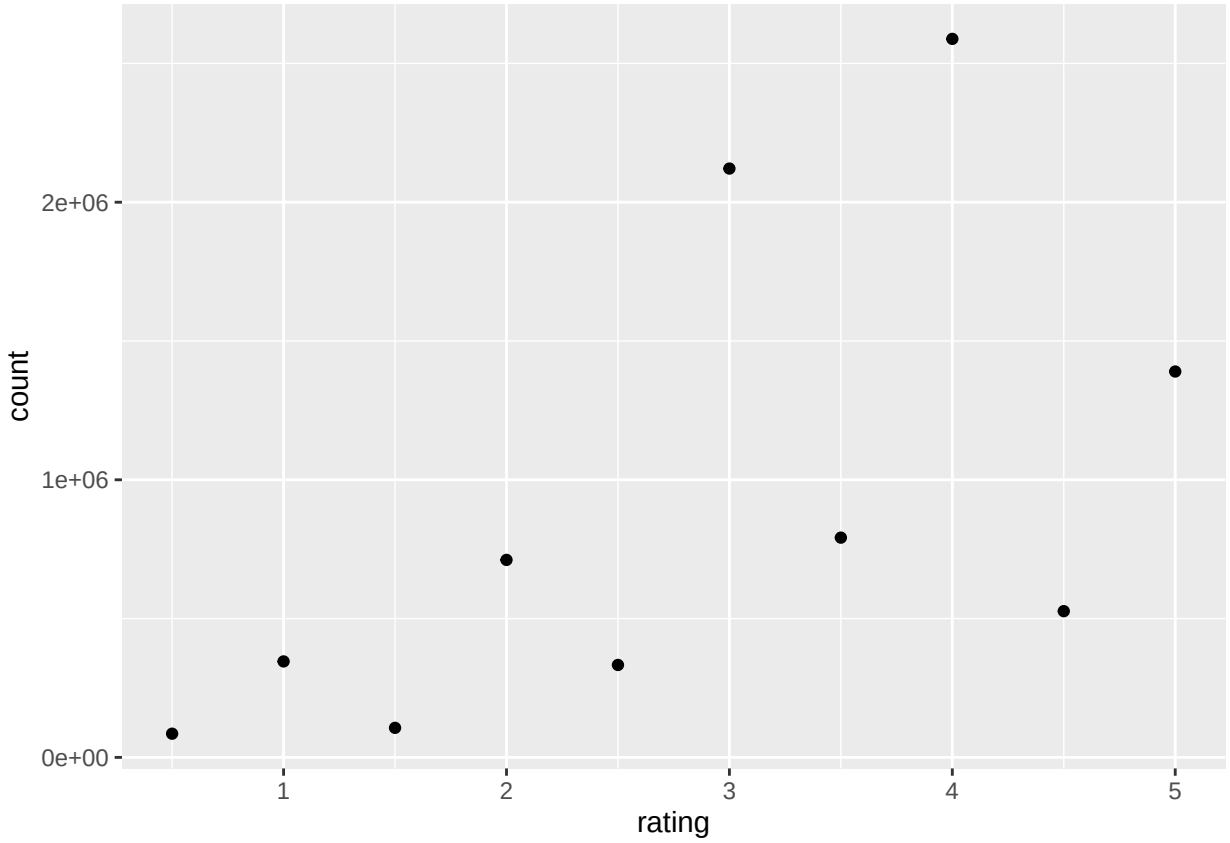
Conclusion

To summarize, this reported introduced the MovieLens dataset and built a model to predict movies' ratings by users, using all-ratings mean modified by different biases, namely: Movie average rating, User average rating, Release year group, average rating (tow groups: before 1978, after 1978), Difference between release and rating year average rating, Genres average rating. Regularizing by lambda = 5.

The model produced the required RMSE (<0.8649) in a resonable amount of time but It can be further improved by:

1- Exploring and examining addittional possible biases: Month and day of the week of the ratings. And analysing others more, like genres combinations, and find similarities and group them in more general groups.

2- Taking into account rating probabilities. As examined in questions 7 & 8 in the quiz, full numbers (1,2,3,4,5) are more common than halfs (0.5, 1.5, 2.5, 3.5, 4.5). Also, 4 and 3 are the most common ratings.



3- Trying other models like knn, decision tree ... etc, to see if they perform better. Although, they will take long time.

4- Ensambling different models' results to get more precise predictions.