Report on Movielens

Nathalie Dumont

3/11/2020

Introduction to movielens dataset

In this report, we will start by downloading the movielens dataset and cleaning the data. We will observe the influence of the different variables contained in the dataset, by making plots. After exploring the data, we will model the different parameters in order to obtain a prediction for the test set, and confront it to the validation set. As a measure of the accuracy of our prediction, we will use the root mean squared error. Afterwards we will propose some other possibilities to improve our model.

Disclaimer : This report is heavily based on the textbook of the edx Data science course. It helped me a lot to compute the graphic parts and to structure the analysis. Also, English is not my mother tongue, so please excuse my grammatical mistakes.

Datacleaning

We start this report by downloading the data and creating two sets : a test one and a validation one.

The movielens dataset is a huge dataset, as you can see :

```
str(edx)
## 'data.frame':    9000055 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 8
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|A
```

The rating consists in associating the userID with a movie, a rating, the timestamp of the rating, the title consisting in the name of the movie with its year in parenthesis, and the genres it belongs to.
It contains more than 9 million observations, which are movie ratings from 69 878 different users on 10 677 different movies.

```
n_distinct(edx$userId)
## [1] 69878
```

From this, we can assume that more than one user gives ratings for several movies. The title format is not satisfying since it combines two pieces of information : title and year. We will slice it in two columns.

```
##   userId movieId rating timestamp                 title
## 1      1     122      5 838985046              Boomerang
## 2      1     185      5 838983525               Net, The
## 3      1     292      5 838983421               Outbreak
## 4      1     316      5 838983392               Stargate
## 5      1     329      5 838983392 Star Trek: Generations
## 6      1     355      5 838984474       Flintstones, The
##                         genres year
## 1                Comedy|Romance 1992
## 2         Action|Crime|Thriller 1995
```

```
## 3  Action|Drama|Sci-Fi|Thriller 1995
## 4       Action|Adventure|Sci-Fi 1994
## 5 Action|Adventure|Drama|Sci-Fi 1994
## 6       Children|Comedy|Fantasy 1994
```

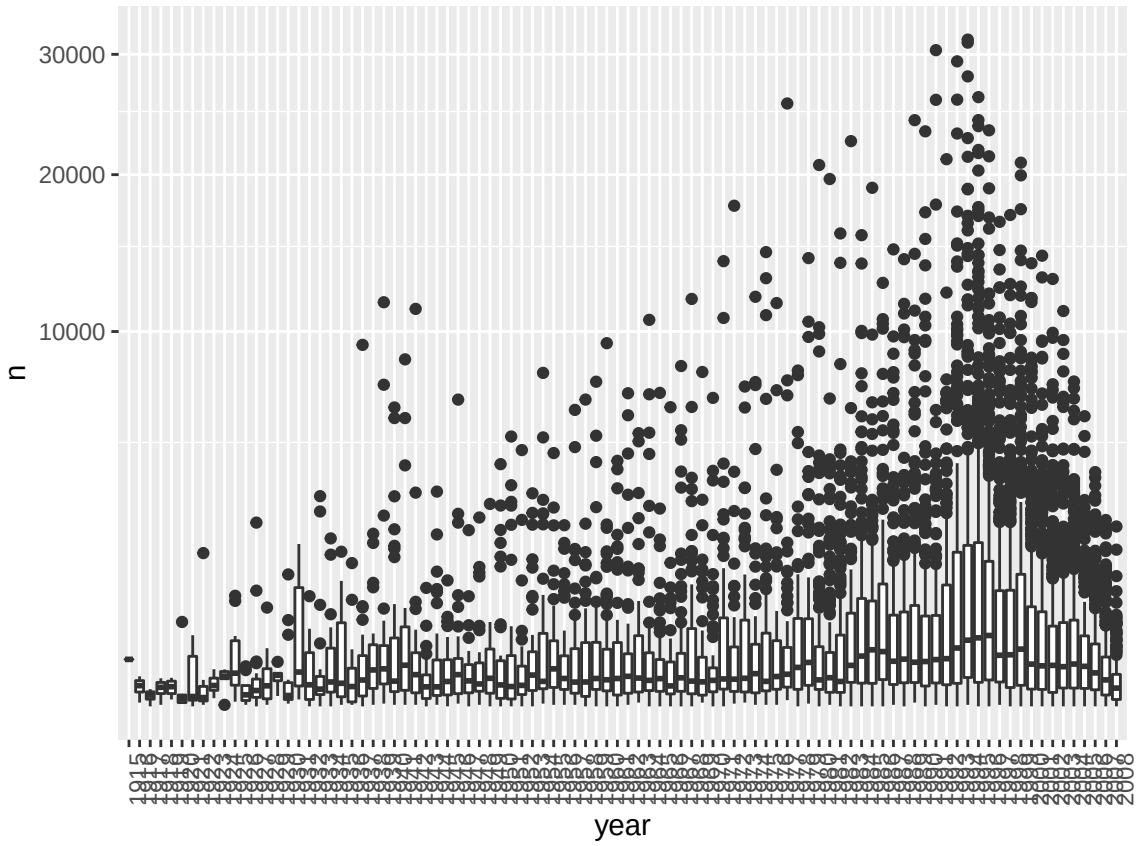
We are also going to change the timestamp to something more readable: the week the rating happened.

```
##   userId movieId rating timestamp                 title
## 1      1     122      5 838985046              Boomerang
## 2      1     185      5 838983525               Net, The
## 3      1     292      5 838983421               Outbreak
## 4      1     316      5 838983392               Stargate
## 5      1     329      5 838983392 Star Trek: Generations
## 6      1     355      5 838984474       Flintstones, The
##                          genres year       date
## 1                Comedy|Romance 1992 1996-08-04
## 2          Action|Crime|Thriller 1995 1996-08-04
## 3  Action|Drama|Sci-Fi|Thriller 1995 1996-08-04
## 4       Action|Adventure|Sci-Fi 1994 1996-08-04
## 5 Action|Adventure|Drama|Sci-Fi 1994 1996-08-04
## 6       Children|Comedy|Fantasy 1994 1996-08-04
```

Influence of the various parameters

Influence of year

Let's observe the influence of year on the movie ratings :

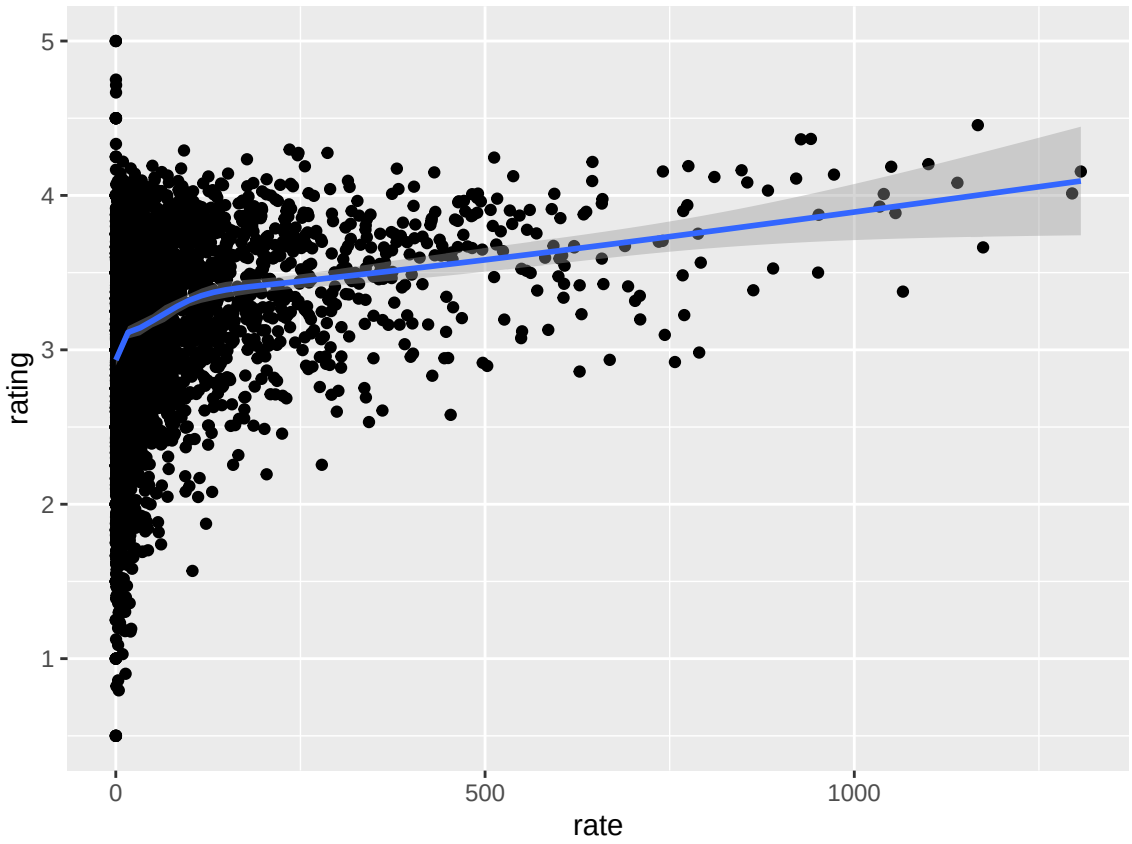


We can see the number of ratings improving in the 90's, and decreasing afterwards. So years seem to have a strong impact on the number of ratings. What about the most rated movies ?

Influence of the number of ratings

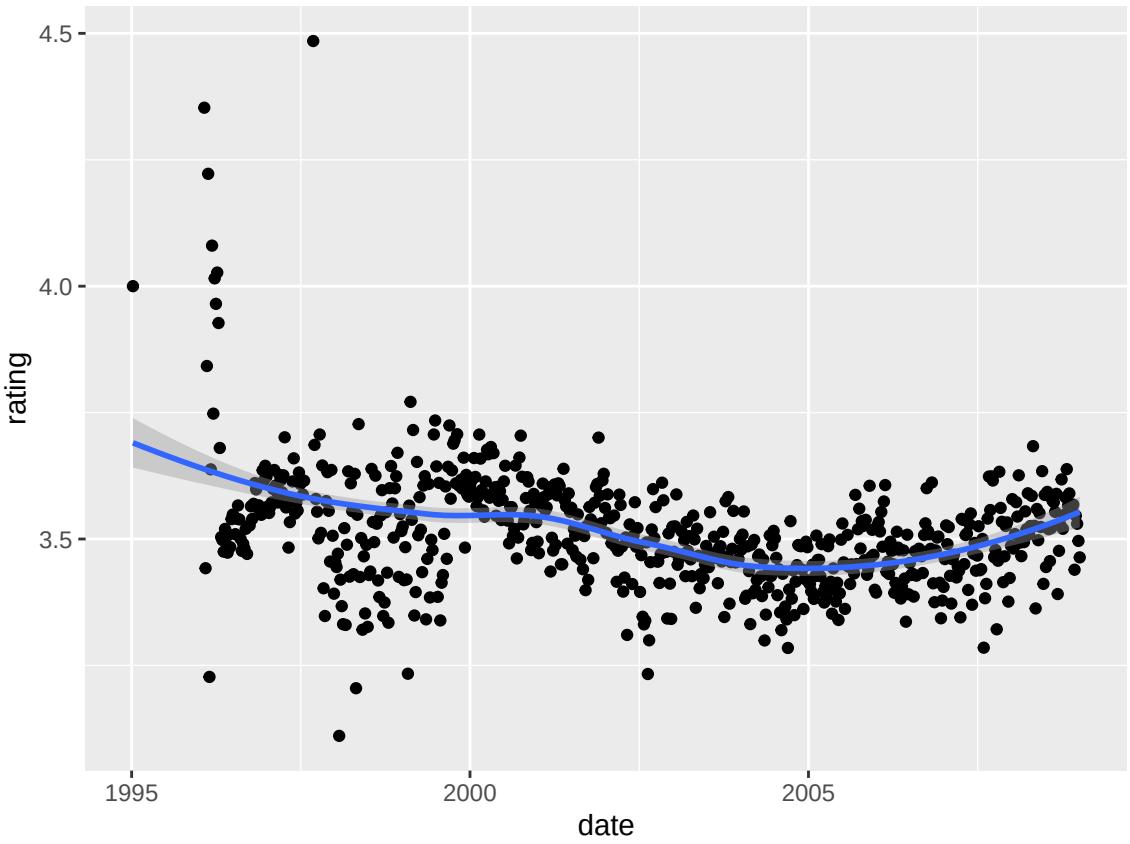
```
## # A tibble: 25 x 6
##    movieId     n years title                          rating  rate
##      <dbl> <int> <dbl> <chr>                           <dbl> <dbl>
##  1     110 26212    25 Braveheart                       4.08 1048.
##  2    2571 20908    21 Matrix, The                      4.20  996.
##  3     780 23449    24 Independence Day (a.k.a. ID4)    3.38  977.
##  4     150 24284    25 Apollo 13                        3.89  971.
##  5       1 23790    25 Toy Story                        3.93  952.
##  6    2858 19950    21 American Beauty                  4.19  950
##  7     608 21395    24 Fargo                            4.13  891.
##  8      32 21891    25 12 Monkeys (Twelve Monkeys)      3.87  876.
##  9      50 21648    25 Usual Suspects, The              4.37  866.
## 10    2762 17504    21 Sixth Sense, The                 4.11  834.
## # ... with 15 more rows
```

It seems that the most rated movies have great ratings. This plot confirms it.



Influence of date

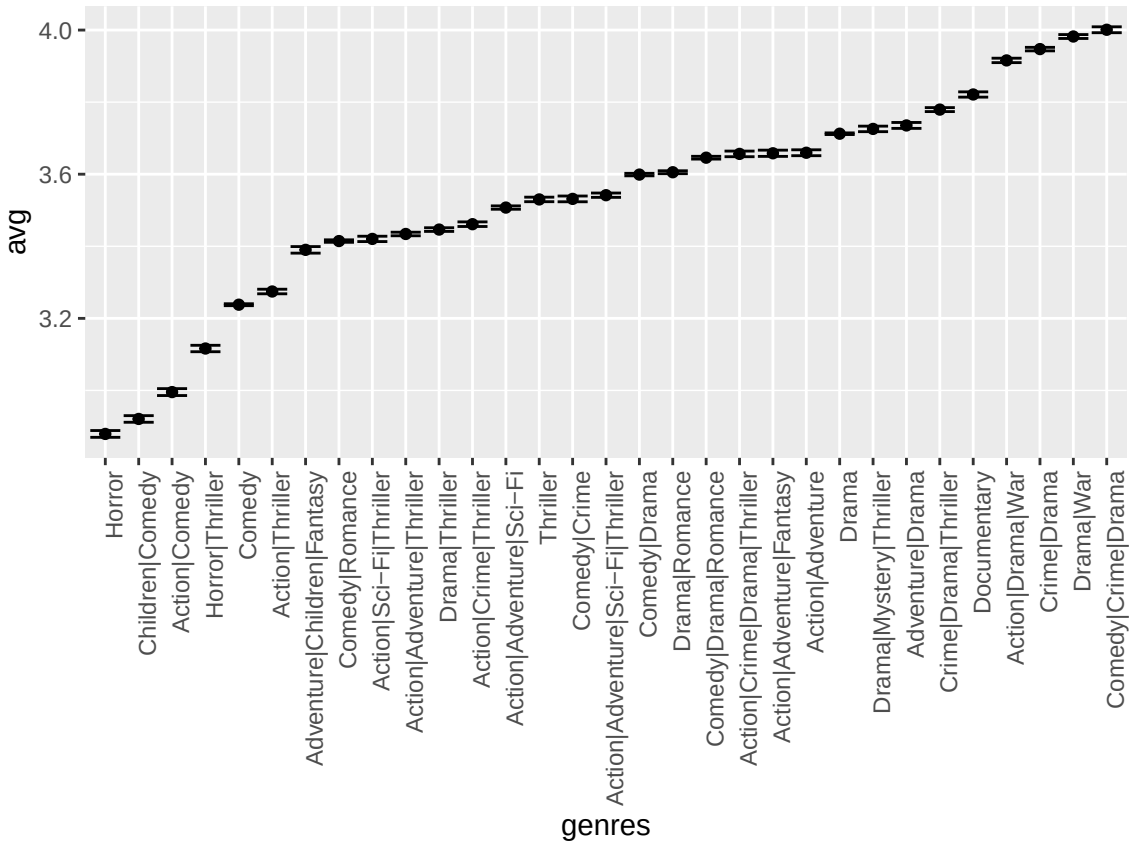
Does the timestamp (or date as we have renamed it) has an influence on ratings ?



Date seems to have only some influence on ratings.

Influence of genre

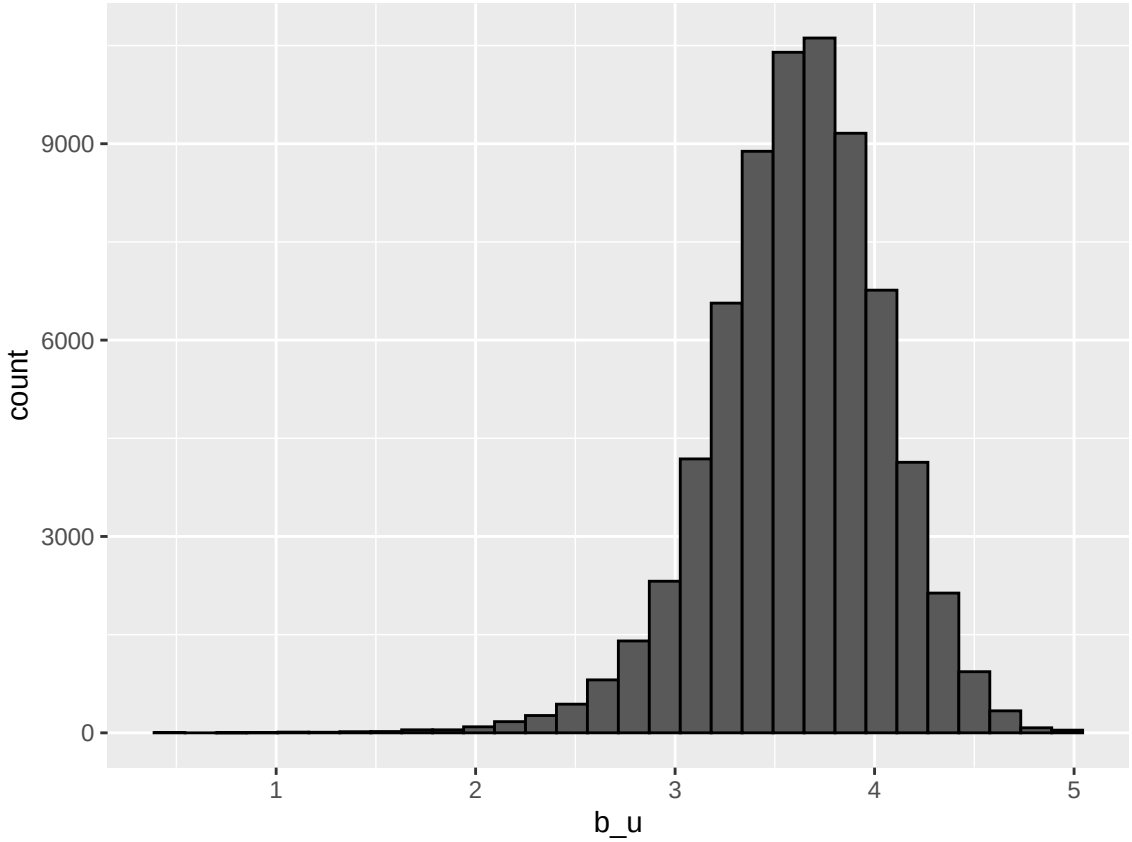
We will plot the average rating by genre :



We can observe that genre has a great influence upon rating.

Influence of user

Let's see the mean of the ratings for each user :



We can see on this histogram that some users are prone to good rating whereas others are more critical. So it is likely that the user will influence the rating.

Modeling approach

Mean

If every movie had the same rating, we could predict the mean rating for any movie. The estimate that minimizes RMSE is the average of all ratings.

```
mu_hat<-mean(edx_mod$rating)
mu_hat
## [1] 3.512465
```

We can build an RMSE function to evaluate our prediction

```
RMSE<-function(true_ratings,predicted_ratings){
  sqrt(mean((true_ratings-predicted_ratings)^2))
}
```

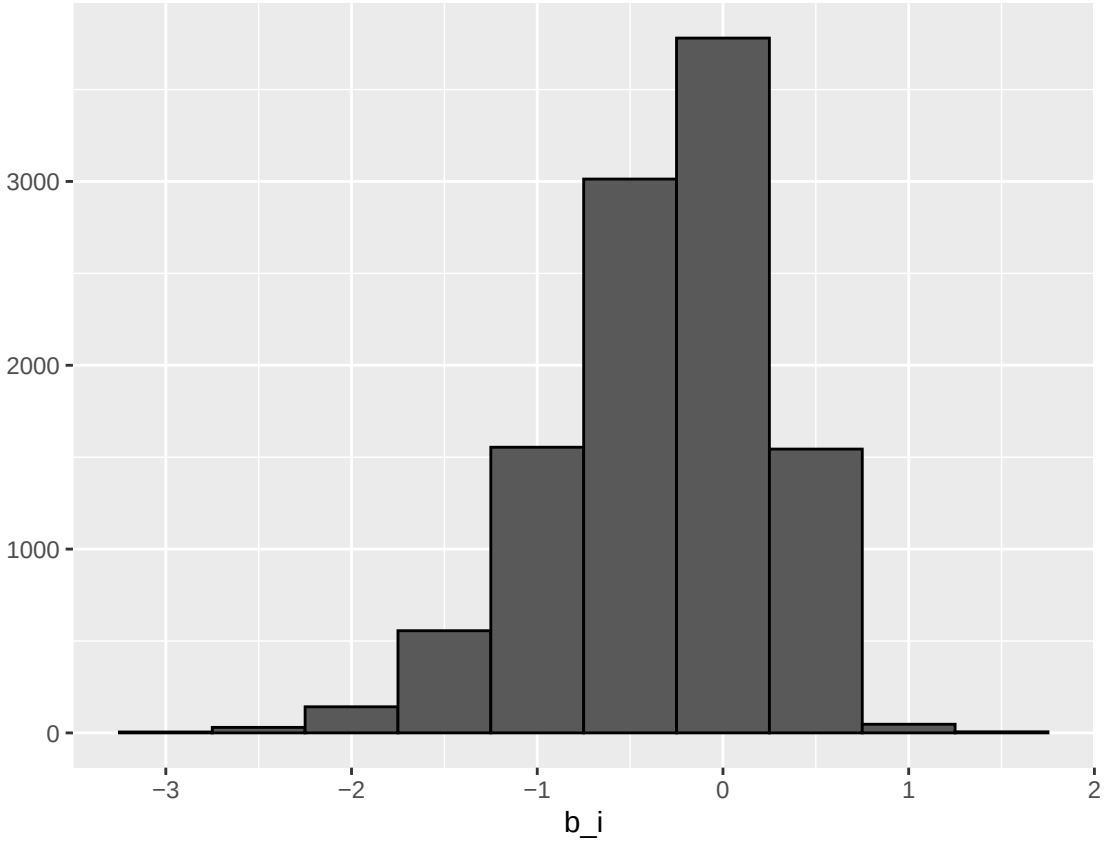
The simplest model would predict the mean of ratings for unrated movie. We compare the actual rating versus the predicted rating, which in our case is the mean, mu_hat.

```
RMSE(edx_mod$rating,mu_hat)
## [1] 1.060331
```

The root squared mean error is quite high. As we get something around 1, we are 1 point away from the actual rating when we are predicting.

Best and worst movies

It is likely that the highest rated movies will get higher ranking, and worst movies on the contrary will get lower ranking. We will add a term to the mean rating in our model, which represents the average rating for each movie. It will be the average of the true ratings for the movie minus the average.



This estimate - a biais for bad/good movie, variates a lot. It seems a good idea to take it in consideration for our model. Let's see how it improves our RMSE :

```
predicted_ratings<-mu_hat+edx_mod %>%
  left_join(movie_avgs,by='movieId') %>%
  pull(b_i)
RMSE(predicted_ratings,edx_mod$rating)
## [1] 0.9423475
```

Modeling user effect

We are going to compute the estimate of the user effect, and add it to our model to predict the ratings. The RMSE improves.

```
user_avgs <- edx_mod2 %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu_hat - b_i))
predicted_ratings <- edx_mod2 %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
```

```
  mutate(pred = mu_hat + b_i + b_u) %>%
  pull(pred)
RMSE(predicted_ratings, edx_mod$rating)
## [1] 0.8567039
```

Modeling time effect

As we have seen that time has a slight effect on ratings, we model the effect of time (or date) via a smooth function.

```
time_effect <- edx_mod2 %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(date=round_date(date,unit='week'))%>%
  group_by(date)%>%
  summarize(b_t = smooth(mean(rating - mu_hat - b_i - b_u)))
edx_mod2<-edx_mod2%>%mutate(date=round_date(date,unit='week'))
```

Then we can view how the RMSE has improved

```
## [1] 0.8566027
```

Just a little, as expected.

Modeling the genre effect

Let's model the genre effect :

```
genre<-edx_mod2%>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(time_effect,by='date') %>%
  group_by(genres)%>%
  summarize(b_g = mean(rating - mu_hat - b_i - b_u - b_t))

predicted_ratings <- edx_mod2 %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(time_effect, by='date')%>%
  left_join(genre, by='genres') %>%
  mutate(pred = mu_hat + b_i + b_u + b_t + b_g) %>%
  pull(pred)

RMSE(predicted_ratings, edx_mod2$rating)
## [1] 0.8562585
```

The results

We have now a model which includes the influence of multiple factors : general rating of the movie by other users, date, genre. We will now calculate the RMSE over the validation set. We start by splitting the title column and reconstructing the date.

```
validation_2<-validation %>%
  mutate("title"=str_match(validation$title,"(.*) \\((.*)\\)")[,2], "year"=as.numeric(str_match(validati
validation_2 <-validation_2 %>%
  mutate(date = as_datetime(timestamp)) %>%
  mutate(date = round_date(date, unit = "week"))
predicted_ratings <- validation_2 %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(time_effect, by='date')%>%
  left_join(genre, by='genres') %>%
  mutate(pred = mu_hat + b_i + b_u + b_t + b_g) %>%
  pull(pred)

RMSE(predicted_ratings, validation_2$rating)
## [1] 0.8648488
```

Conclusion

We could improve our results several ways. As proposed in the course's textbook, we could regularize estimates. We could also do a PCA, principal components analysis to identify correlation between factors.