# Recommendation System

*Mathieu Klimczak*

*12/06/2019*

## Contents



### Introduction

First, let's import the datas and check their structures. Both datasets are the ones that can be downloaded from the following link

https://drive.google.com/drive/folders/1IZcBBX0OmL9wu9AdzMBFUG8GoPbGQ38D

```
edx <- readRDS("Data/edx.rds")
validation <- readRDS("Data/validation.rds")

dim(edx)
```

```
## [1] 9000055       6
```

```
dim(validation)
```

```
## [1] 999999      6
```

```
head(edx)
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
## 7      1     355      5 838984474       Flintstones, The (1994)
##                           genres
## 1                 Comedy|Romance
## 2          Action|Crime|Thriller
## 4   Action|Drama|Sci-Fi|Thriller
## 5        Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
## 7       Children|Comedy|Fantasy
```

```
head(validation)
```

```
##   userId movieId rating timestamp
## 1      1     231      5 838983392
## 2      1     480      5 838983653
## 3      1     586      5 838984068
## 4      2     151      3 868246450
## 5      2     858      2 868245645
```

```
## 6      2    1544      3 868245920
##                                                        title
## 1                                      Dumb & Dumber (1994)
## 2                                      Jurassic Park (1993)
## 3                                         Home Alone (1990)
## 4                                            Rob Roy (1995)
## 5                                     Godfather, The (1972)
## 6 Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                       genres
## 1                                     Comedy
## 2        Action|Adventure|Sci-Fi|Thriller
## 3                           Children|Comedy
## 4                  Action|Drama|Romance|War
## 5                               Crime|Drama
## 6 Action|Adventure|Horror|Sci-Fi|Thriller
```

```
# str(edx)
```

### Construction of the recommendation System

The goal here is to build a recommendation system. That is, given the edx dataset which will serve as a train set, being able to construct a model which will rate movies based on different assumptions.

The metric we'll use to validate our recommendation system will be the Residual Mean Squared Errors metric given the following code.

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

We choose here to build a linear model. We'll first consider a "Naive approach" by adding biases one after another, then we'll consider regularization.

We will concentrate ourselves to the users, movie, genres, and week (which will be constructed later) predictors to create this model.

## Naive approach

In a naive way of thinking, we'll consider first that our system gives the same note for any movies. That is, the mean of all the already known notes.

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

```
## [1] 3.512465
```

The prediction being the same for every movies, it's given by a column vector of dimension nrow(validation) with entires mu_hat.

```
predictions <- rep(mu_hat, nrow(validation))

naive_rmse <- RMSE(validation$rating, predictions)
```

This approach isn't the best. We are missing 1 point in the rating of the movies.

```
rmse_results <-
  tibble(method = "Just the average", RMSE = naive_rmse)

rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.061202 |

### Adding the Movie bias

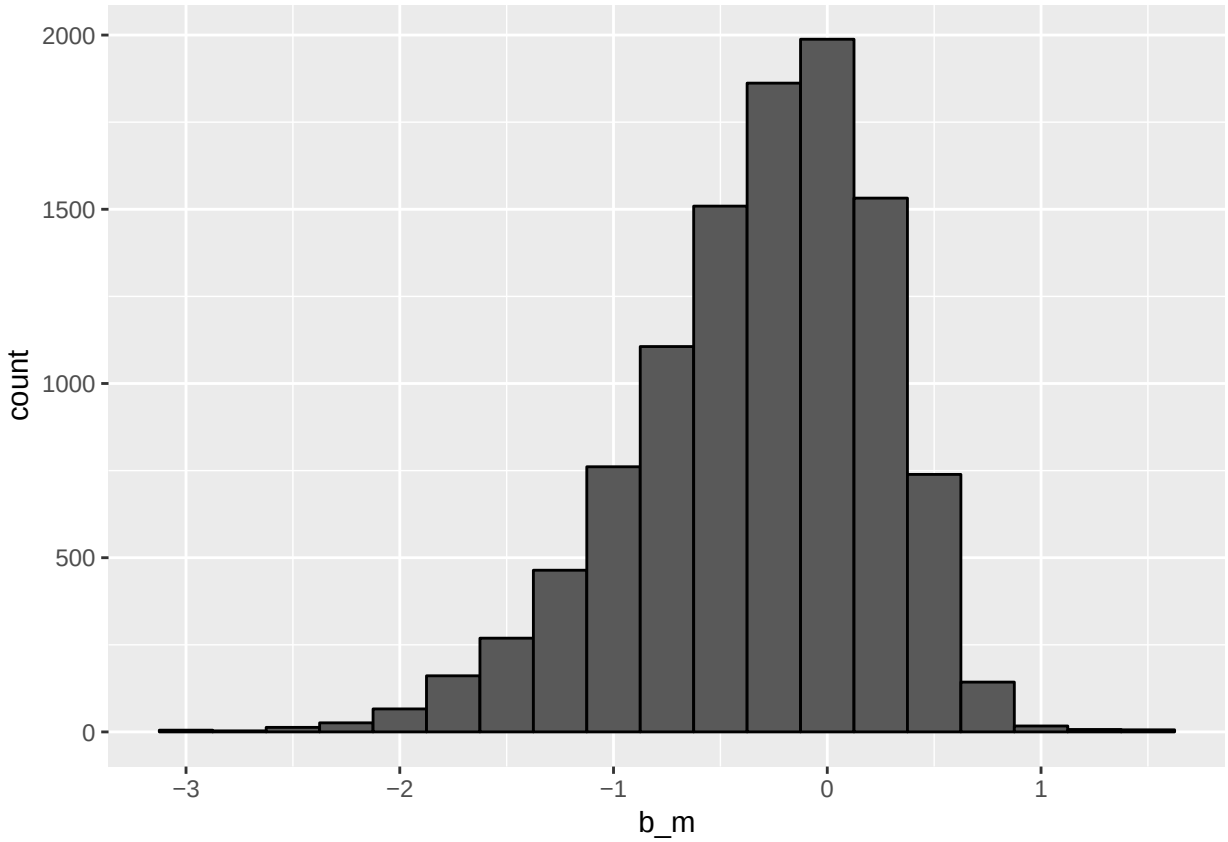
Let's consider there's a movie bias, some movies are considered good and will have a positive bias, others will a negative one. First let's see if this assumption is reflected on the train set.

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

#Let's compute the average bias for each movies
movie_avgs <- edx %>%
  group_by(movieId) %>%
  summarize(b_m = mean(rating - mu))
```

```
head(movie_avgs)
```

```
## # A tibble: 6 x 2
##   movieId    b_m
##     <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
```

```
movie_avgs %>% ggplot(aes(x = b_m)) + geom_histogram(binwidth = 0.25, color = "black")
```




The distribution of this bias looks like a skewed Gaussian, in particular it's not constant, thus certain movies tend to be considered as good movies and have a positive bias, while others won't. We now implement this bias as a new variable of our model.

```
#The RHS of the following code is of the form real_number + Matrix, where
#the matrix is given by the code
#validation %>%   left_join(movie_avgs, by='movieId') %>%  .$b_m

predicted_ratings <- mu + validation %>%
  left_join(movie_avgs, by = 'movieId') %>%
  .$b_m
```

The left join here is used to be sure that the validation dataset has the predictor b_m.

Adding the movie bias as the b_m variable to our model improves the RMSE score, as we can see from the following computations.

```
model_1_rmse <- RMSE(validation$rating, predicted_ratings)

rmse_results <- bind_rows(rmse_results,
                          tibble(method = "Movie Effect Model",
                                 RMSE = model_1_rmse))
rmse_results %>% knitr::kable()
```

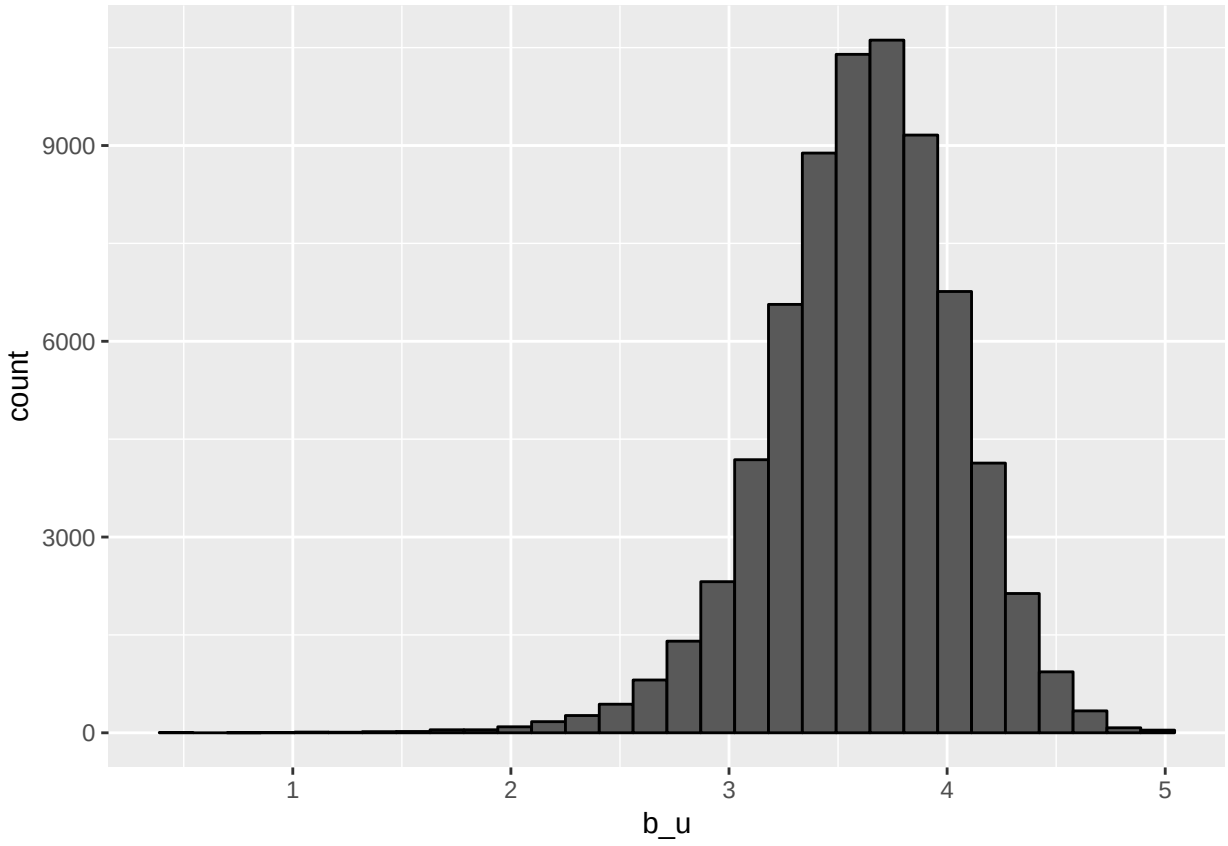
| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |

### Adding the User bias

To further improves our model, we can now consider the user bias. Some users will like action movies and give them a higher rate, while others will prefer comedies, thus giving a smaller score on action movies than the first ones.

This assumption about the user bias seems to be confirmed by the following plot, showing the distribution of the notes for users that have rated at least a hundred movies.

```
edx %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating)) %>%
  filter(n() >= 100) %>%
  ggplot(aes(b_u)) +
  geom_histogram(bins = 30, color = "black")
```




Let's extract the user bias from our train set and implement it in our model as the b_u variable.

```
user_avgs <- edx %>%
  left_join(movie_avgs, by = 'movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_m))
```

Once again, adding this variable to our model has improved our RMSE score.

```
predicted_ratings <- validation %>% left_join(movie_avgs, by = "movieId") %>%
    left_join(user_avgs, by = "userId") %>% mutate(pred = mu + b_m + b_u) %>%
    .$pred
```

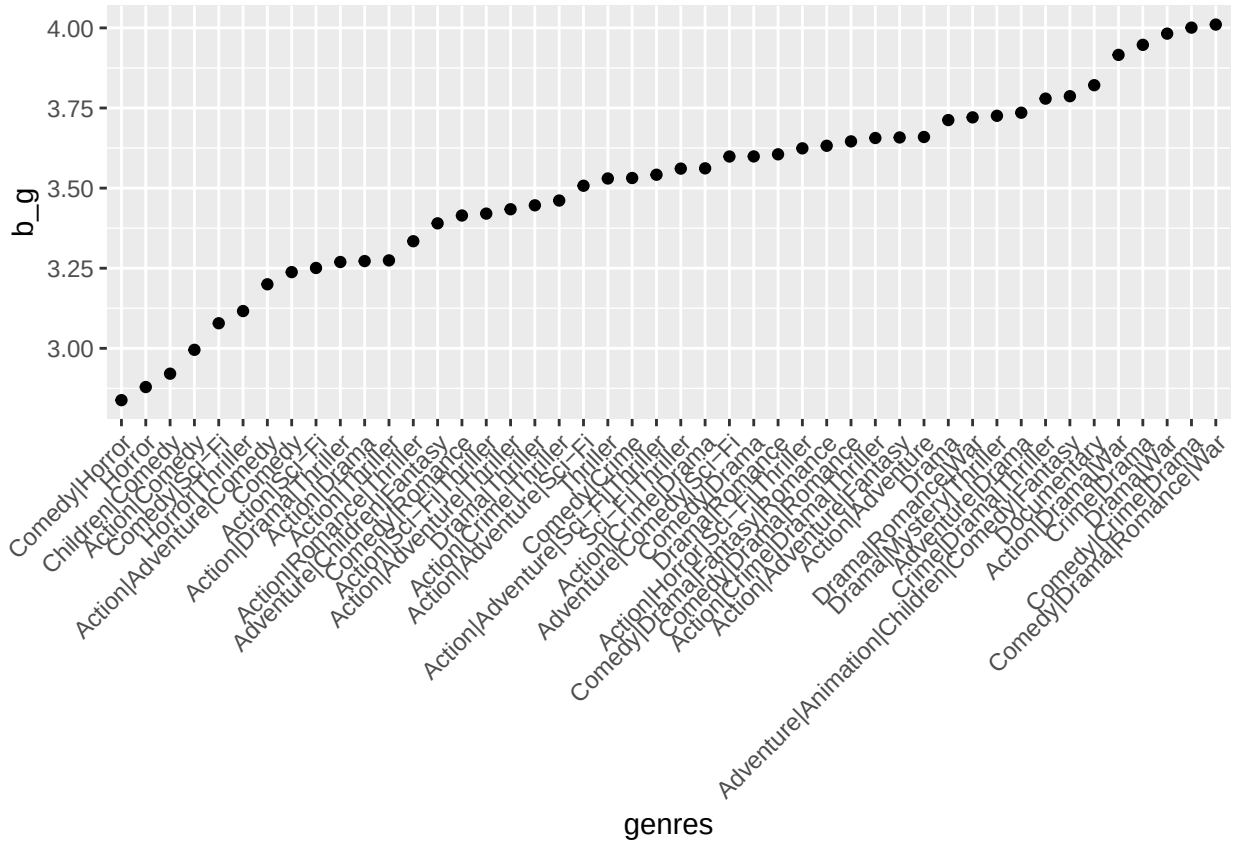
```
model_2_rmse <- RMSE(validation$rating, predicted_ratings)

rmse_results <- bind_rows(rmse_results, tibble(method = "Movie + User Effects Model",
    RMSE = model_2_rmse))

rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |

### Adding the Genres bias

As some genres might be more popular than other, they might tend to have better ratings. Let's just see here the average ratings of the genres having more than 35000 ratings.

```
edx %>% group_by(genres) %>%
  filter(n()>35000) %>%
  summarize(b_g = mean(rating)) %>%
  mutate(genres = reorder(genres, b_g)) %>%
  ggplot(aes(x=genres,y=b_g)) +
  geom_point()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




Let's compute this genre bias.

```
genre_avgs <- edx %>%
  left_join(movie_avgs, by = 'movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  group_by(genres) %>%
  summarize(b_g = mean(rating - mu - b_m - b_u))
```

Although the other assumptions clearly improved the RMSE, adding gender bias did not improve the RMSE that much.

```
predicted_ratings <- validation %>%
  left_join(movie_avgs, by = 'movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  left_join(genre_avgs, by = 'genres') %>%
  mutate(pred = mu + b_m + b_u + b_g) %>%
  .$pred


model_3_rmse <- RMSE(validation$rating, predicted_ratings)

rmse_results <- bind_rows(rmse_results,
                          tibble(method = "Movie + User + Genres Effects Model",
                                 RMSE = model_3_rmse))

rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |
| Movie + User + Genres Effects Model | 0.8649469 |

### Adding the week bias

We'll study here the effect of the week on the ratings of the movies. Before doing that, we had to our datasets, the required predictor.

```
edx <- edx %>% mutate(date = as_datetime(timestamp),
                      week = round_date(date, "week"))

validation <- validation %>% mutate(date = as_datetime(timestamp),
                                    week = round_date(date, "week"))
```

```
week_avgs <- edx %>%
  left_join(movie_avgs, by = 'movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  left_join(genre_avgs, by = 'genres') %>%
  group_by(week) %>%
  summarize(b_w = mean(rating - mu - b_m - b_u - b_g))
```

Adding the week bias in our model gives us the following RMSE score.

```
predicted_ratings <- validation %>%
  left_join(movie_avgs, by = 'movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  left_join(genre_avgs, by = 'genres') %>%
```

```
  left_join(week_avgs, by = 'week') %>%
  mutate(pred = mu + b_m + b_u + b_g + b_w) %>%
  .$pred


model_4_rmse <- RMSE(validation$rating, predicted_ratings)

rmse_results <- bind_rows(
  rmse_results,
  tibble(method = "Movie + User + Genres + Week Effects Model",
         RMSE = model_4_rmse)
)

rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |
| Movie + User + Genres Effects Model | 0.8649469 |
| Movie + User + Genres + Week Effects Model | 0.8648392 |

## Regularization

Up to now, we have considered every observations in the same way, whether it is a niche movie with few but excellent ratings, resulting in a high movie bias, or a Hollywood blockbuster movie with hundreds of ratings, some goods some bads, and thus with a mitigated movie bias.

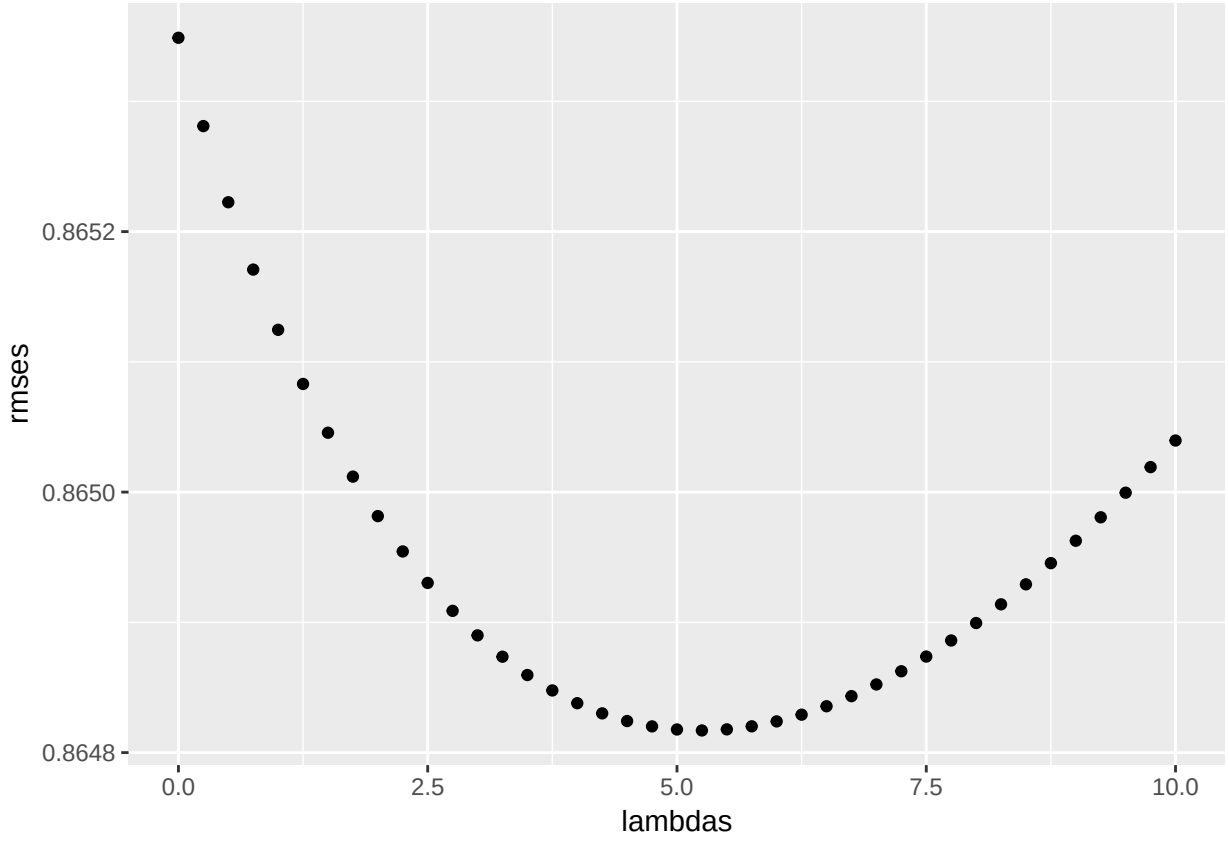
We'll try to overcome this problem by using regularization. Rather than taking the mean in our computations, we'll add a coefficient lambda in our denominators to try to the effect cause by the number of observations.

We'll try several regularizations.

### Movie + User Regularization

```
lambdas <- seq(0, 10, 0.25)


rmses <- sapply(lambdas, function(l) {
  mu <- mean(edx$rating)
  #Regul movie bias
  Rmovie_bias <- edx %>%
    group_by(movieId) %>%
    summarize(b_m = sum(rating - mu) / (n() + l))
  #Regul user bias
  Ruser_bias <- edx %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_m - mu) / (n() + l))

  #apply predictions
  predicted_ratings <-
    validation %>%
```

```
    left_join(Rmovie_bias, by = "movieId") %>%
    left_join(Ruser_bias, by = "userId") %>%
    mutate(pred = mu + b_m + b_u) %>%
    .$pred
  return(RMSE(predicted_ratings, validation$rating))
})

qplot(lambdas, rmses)
```




```
#the lambda giving the minimal RMSE is
lambda <- lambdas[which.min(rmses)]

lambda
```

```
## [1] 5.25
```

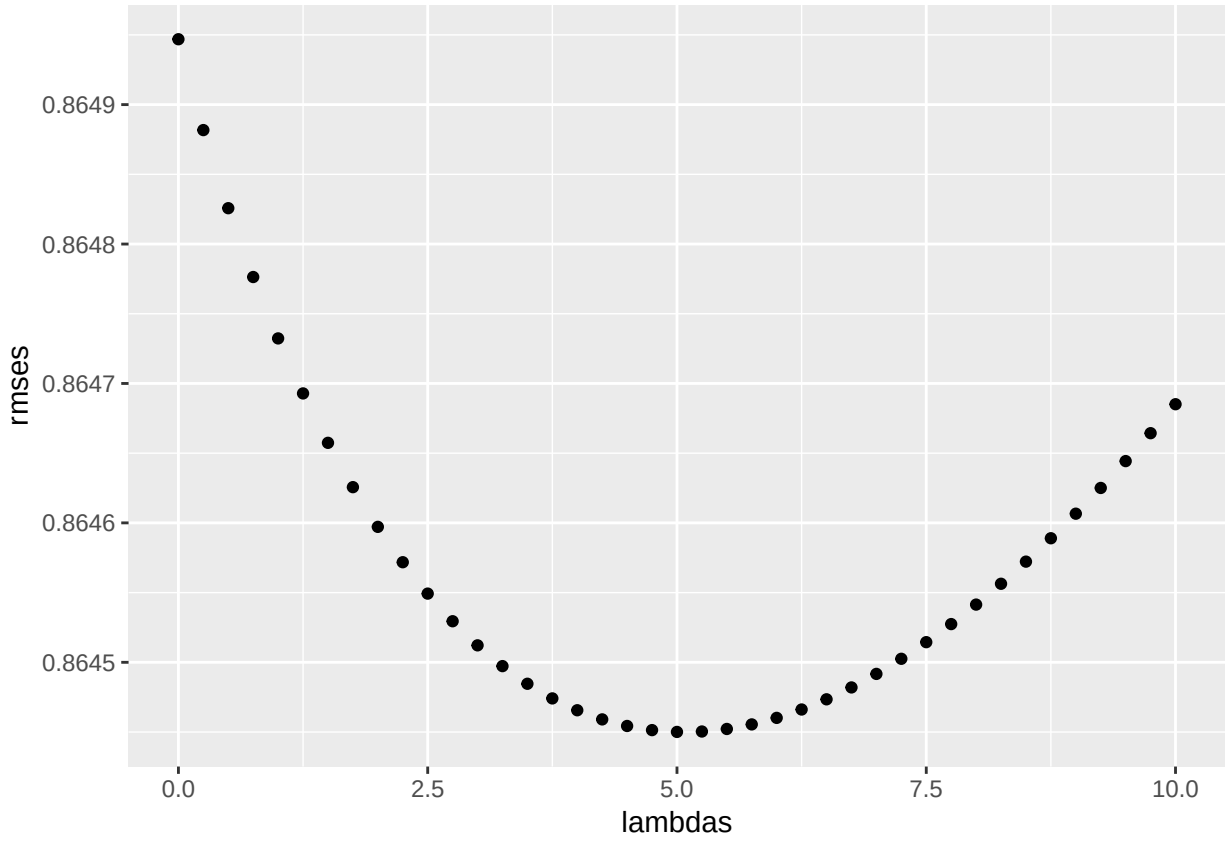
```
rmse_results <- bind_rows(rmse_results,
                          tibble(method = "Regularized Movie + User Effect Model",
                                 RMSE = min(rmses)))
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |
| Movie + User + Genres Effects Model | 0.8649469 |
| Movie + User + Genres + Week Effects Model | 0.8648392 |

| method | RMSE |
|---|---|
| Regularized Movie + User Effect Model | 0.8648170 |

**Movie + User + Genres Regularization**

```
rmses <- sapply(lambdas, function(l) {
  mu <- mean(edx$rating)
  #Regul movie bias
  Rmovie_bias <- edx %>%
    group_by(movieId) %>%
    summarize(b_m = sum(rating - mu) / (n() + l))
  #Regul user bias
  Ruser_bias <- edx %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_m - mu) / (n() + l))
  #regul genres bias
  Rgenre_bias <- edx %>%
    left_join(Rmovie_bias, by = 'movieId') %>%
    left_join(Ruser_bias, by = 'userId') %>%
    group_by(genres) %>%
    summarize(b_g = sum(rating - b_m - b_u - mu) / (n() + l))

  #apply predictions
  predicted_ratings <-
    validation %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    left_join(Ruser_bias, by = "userId") %>%
    left_join(Rgenre_bias, by = 'genres') %>%
    mutate(pred = mu + b_m + b_u + b_g) %>%
    .$pred
  return(RMSE(predicted_ratings, validation$rating))
})

qplot(lambdas, rmses)
```




```
#the lambda giving the minimal RMSE is
lambda <- lambdas[which.min(rmses)]

lambda
```

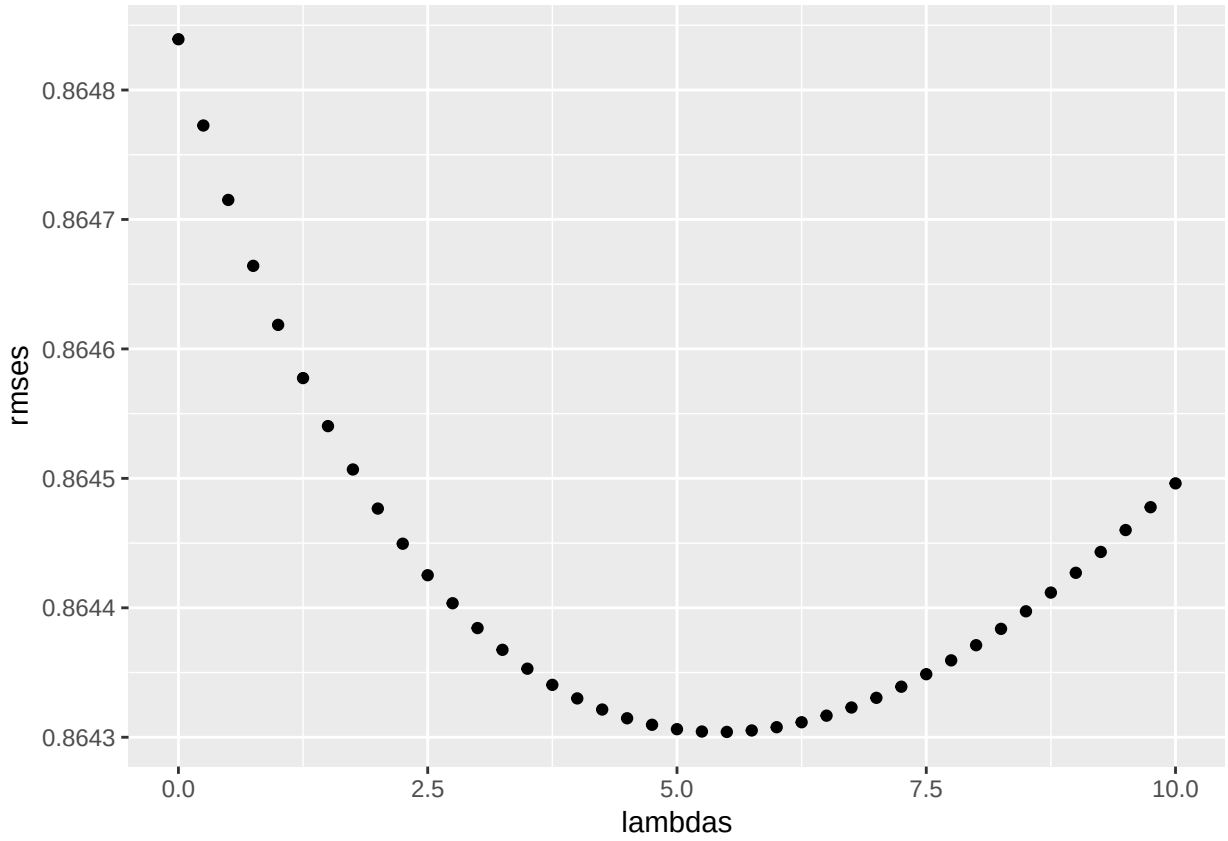
```
## [1] 5
```

```
rmse_results <- bind_rows(
  rmse_results,
  tibble(method = "Regularized Movie + User + Genres Effect Model",
             RMSE = min(rmses))
)
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |
| Movie + User + Genres Effects Model | 0.8649469 |
| Movie + User + Genres + Week Effects Model | 0.8648392 |
| Regularized Movie + User Effect Model | 0.8648170 |
| Regularized Movie + User + Genres Effect Model | 0.8644501 |

**Movie + User + Genres + Week Regularization**

```
rmses <- sapply(lambdas, function(l) {
  mu <- mean(edx$rating)
  #Regul movie bias
  Rmovie_bias <- edx %>%
    group_by(movieId) %>%
    summarize(b_m = sum(rating - mu) / (n() + l))
  #Regul user bias
  Ruser_bias <- edx %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_m - mu) / (n() + l))
  #regul genres bias
  Rgenre_bias <- edx %>%
    left_join(Rmovie_bias, by = 'movieId') %>%
    left_join(Ruser_bias, by = 'userId') %>%
    group_by(genres) %>%
    summarize(b_g = sum(rating - b_m - b_u - mu) / (n() + l))
  #regul week bias
  Rweek_bias <- edx %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    left_join(Ruser_bias, by = "userId") %>%
    left_join(Rgenre_bias, by = 'genres') %>%
    group_by(week) %>%
    summarize(b_w = sum(rating - b_g - b_m - b_u - mu) / (n() + l))

  #apply predictions
  predicted_ratings <-
    validation %>%
    left_join(Rmovie_bias, by = "movieId") %>%
    left_join(Ruser_bias, by = "userId") %>%
    left_join(Rgenre_bias, by = 'genres') %>%
    left_join(Rweek_bias, by = 'week') %>%
    mutate(pred = mu + b_m + b_u + b_g + b_w) %>%
    .$pred
  return(RMSE(predicted_ratings, validation$rating))
})

qplot(lambdas, rmses)
```




```
#the lambda giving the minimal RMSE is
lambda <- lambdas[which.min(rmses)]

lambda
```

```
## [1] 5.5
```

```
rmse_results <- bind_rows(
  rmse_results,
  tibble(method = "Regularized Movie + User + Genres + Week Effect Model",
             RMSE = min(rmses))
)
```

## Results

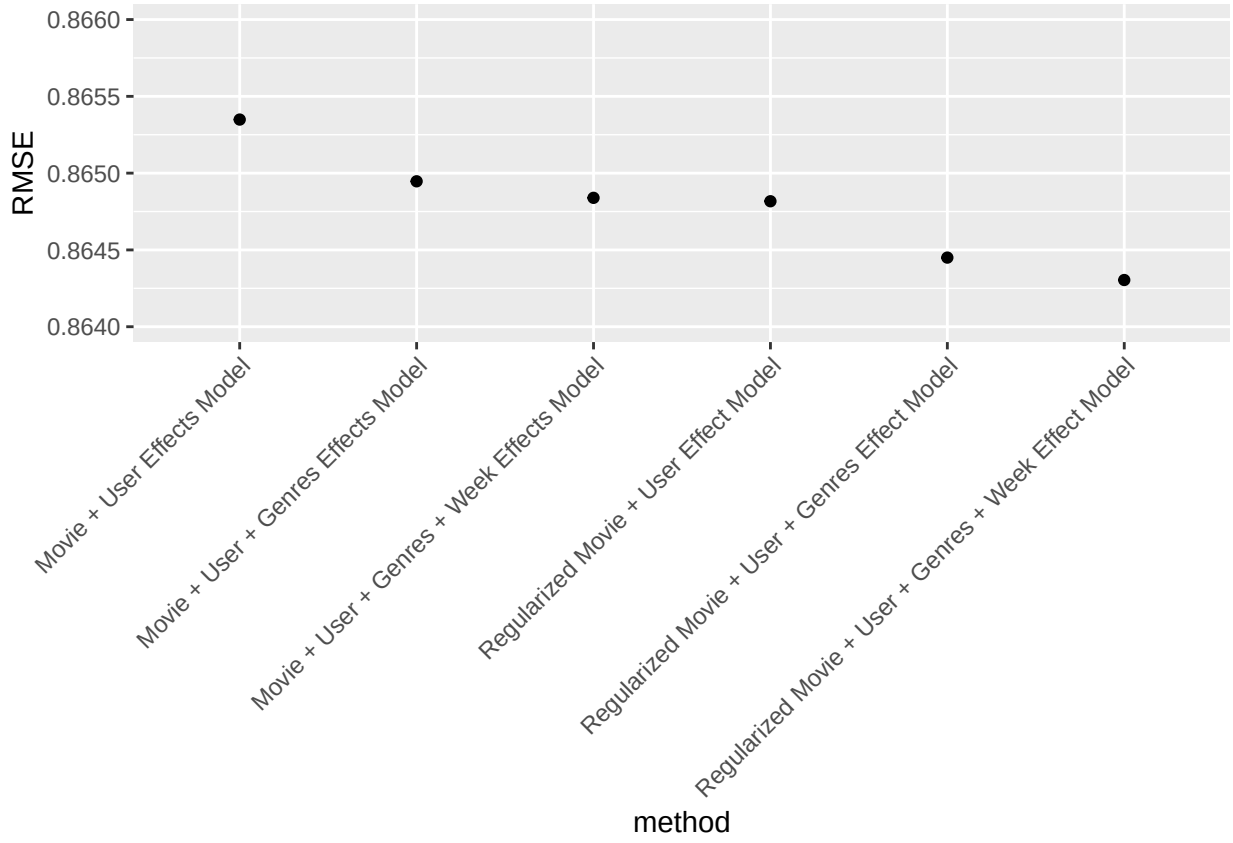
Here's the summarized results about the RMSE socres of the differents models.

```
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| Just the average | 1.0612018 |
| Movie Effect Model | 0.9439087 |
| Movie + User Effects Model | 0.8653488 |
| Movie + User + Genres Effects Model | 0.8649469 |
| Movie + User + Genres + Week Effects Model | 0.8648392 |
| Regularized Movie + User Effect Model | 0.8648170 |
| Regularized Movie + User + Genres Effect Model | 0.8644501 |

| method | RMSE |
|---|---|
| Regularized Movie + User + Genres + Week Effect Model | 0.8643041 |

```
rmse_results %>% filter(RMSE <=0.9) %>%
  mutate(method=reorder(method,desc(RMSE))) %>%
  ggplot(aes(x=method,y=RMSE, ymin=0.864, ymax=0.866)) +
  geom_point()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




## Conclusion

While the first two biases, movie and user, had an significant impact on the improvement of the RMSE, the genres and week bias, as they are right now, don't.

Using regularization on the Movie+User model is at the same level that the model of all four biases. Thus we might discard the genres and week bias, and stick to the regularized Movie+User model.

The last two regularied models did make an improvement of the RMSE score, but they are also more time and memory consuming regularized Movie+User model.

A good trade-off choice would then be the regularized Movie+User model.