PH125.9x Capstone Movielens project

Thomas Marx

24/05/2020

Contents

1 Introduction

This report presents the first of the two projects required for the PH125.9x Capstone module of the EdX/HarvardX Data Science Professionnal Certificate.

This first project, based on the 10M-rating Movielens database, aims at training a machine learning algorithm so as to predict user ratings (from 0.5 to 5.0 stars) of movies. The quality of this algorithm will be evaluated on the basis of the Root Mean Square Error (RMSE), which will assess, on a validation dataset (constructed by code provided by EdX/HavardX), the error in terms of star rating of the predicted values provided by the various algorithms. The RMSE function will henceforth be compiled as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(\hat{y}_i - y_i)^2}$$

where i is the set of variables taken into account to minimize the RMSE. In terms of code, this will be expressed as:

```
RMSE <- function(true_rating, pred_rating){
  sqrt(mean((true_rating - pred_rating)^2))
}
```

2 Dataset analysis

The code to download the data, label it, split it into a training set (labelled *edx*) and a validation set (labelled *validation*) was provided by EdX/HarvardX.

The proper analysis of the data involves two steps:

1. Assessing the nature, characteristics and validity of the data;

2. Transforming the data provided so as to conduct additional computations

2.1 Discovering the dataset

The first step consists in displaying the header of the *edx* file:

userId	movieId	rating	title	year_release	genres
1	122	5	Boomerang	1992	Comedy|Romance
1	185	5	Net, The	1995	Action|Crime|Thriller
1	292	5	Outbreak	1995	Action|Drama|Sci-Fi|Thriller
1	316	5	Stargate	1994	Action|Adventure|Sci-Fi
1	329	5	Star Trek: Generations	1994	Action|Adventure|Drama|Sci-Fi
1	355	5	Flintstones, The	1994	Children|Comedy|Fantasy

and then some basic statistics, which confirm the range validity of the variables:

userId	movieId	rating	title	year_release	genres
Min. : 1	Min. : 1	Min. :0.500	Length:9000055	Min. :1915	Length:9000055
1st Qu.:18124	1st Qu.: 648	1st Qu.:3.000	Class :character	1st Qu.:1987	Class :character
Median :35738	Median : 1834	Median :4.000	Mode :character	Median :1994	Mode :character
Mean :35870	Mean : 4122	Mean :3.512	NA	Mean :1990	NA
3rd Qu.:53607	3rd Qu.: 3626	3rd Qu.:4.000	NA	3rd Qu.:1998	NA
Max. :71567	Max. :65133	Max. :5.000	NA	Max. :2008	NA

(these displays are based on a transformed *edx* file, with title and release date separated for modelling purposes, *cf. infra* and code)

2.2 Transforming the dataset

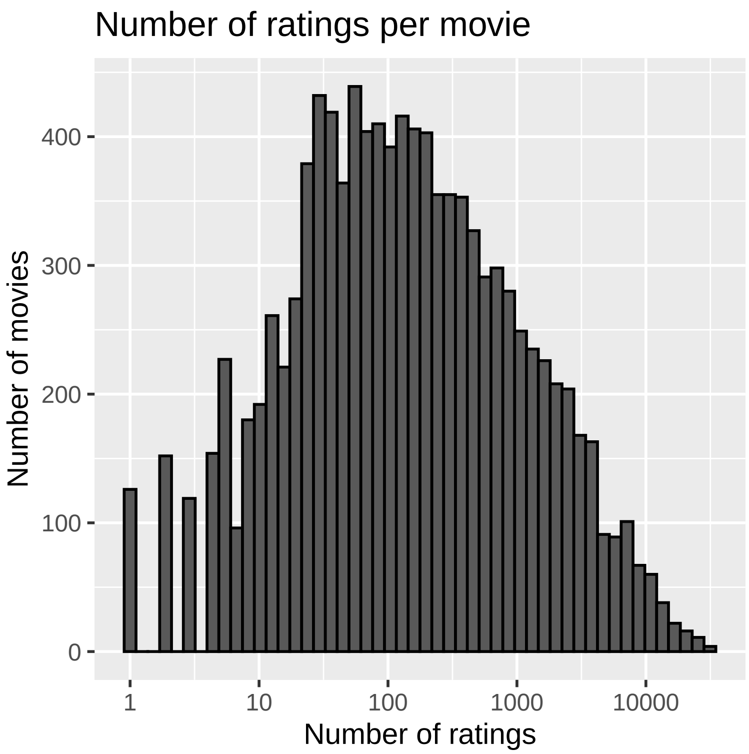
We feel that the release year of the movies could be a useful factor to take into account in our modelling. We thus isolate the proper title from the release date, using the formatting of the title as furnished (four-figure date into parentheses; the basic statistics computed above show that there are no outliers to this format, as minimum and maximum are within credible values, and the quartiles can be computed, thus confirming the absence of NAs).

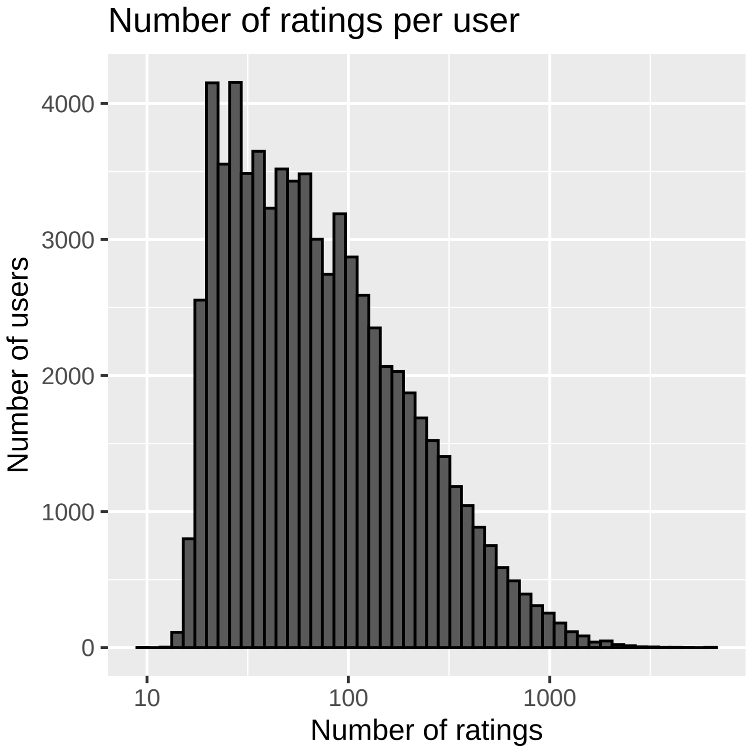
Using this release-year vector, we calculate the age of the movie at time of review, by substracting the release year from the year of the timestamp, and rounding negative values (which can occur depending on the month of review when the review year is equal to the release year) to zero.

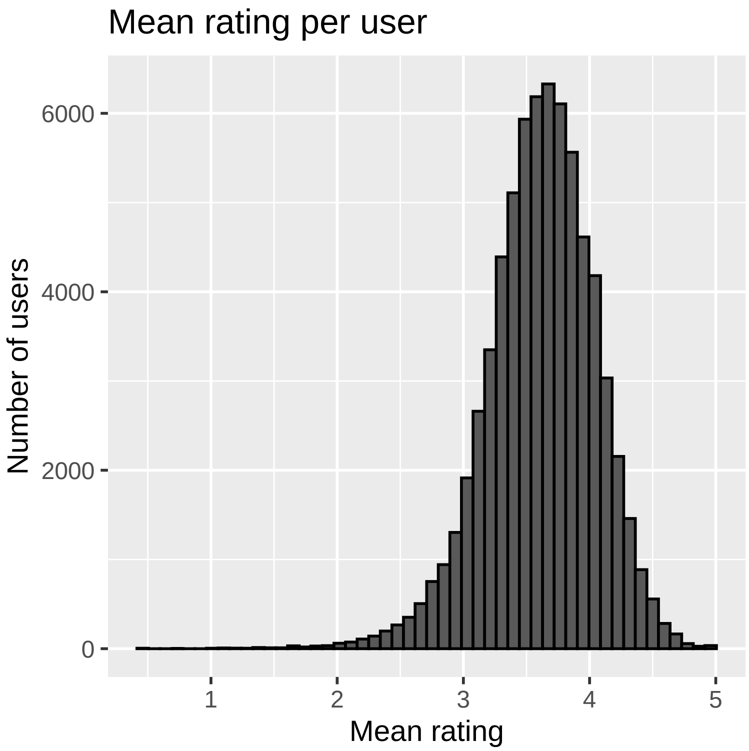
We also use the timestamp to extract the time of the day (or night) when the user rated the movie. It can indeed be a variable giving us information as to the mood he or she is in, and thus can be used to further refine our machine learning algorithm.

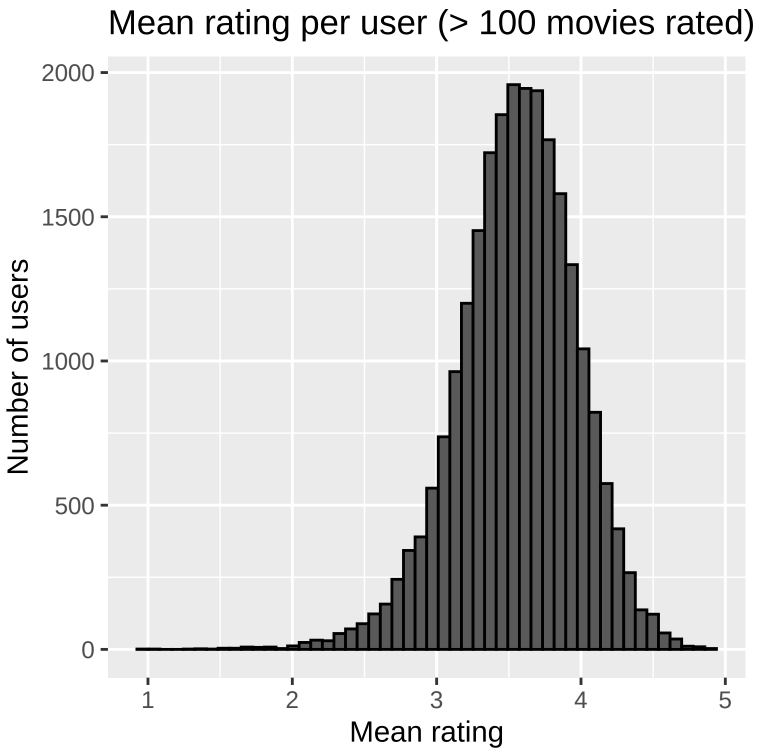
2.3 Visual approach of the dataset

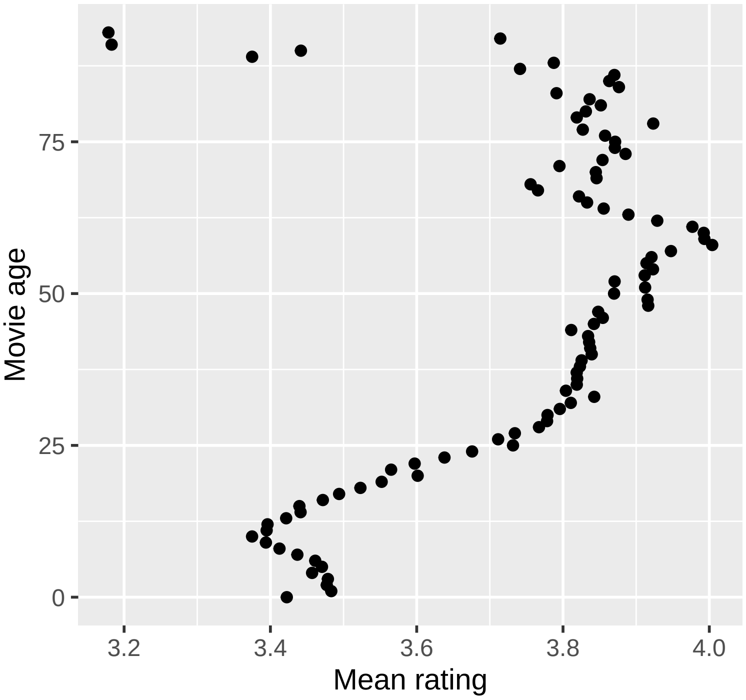
We can now visually inspect how the ratings are split:



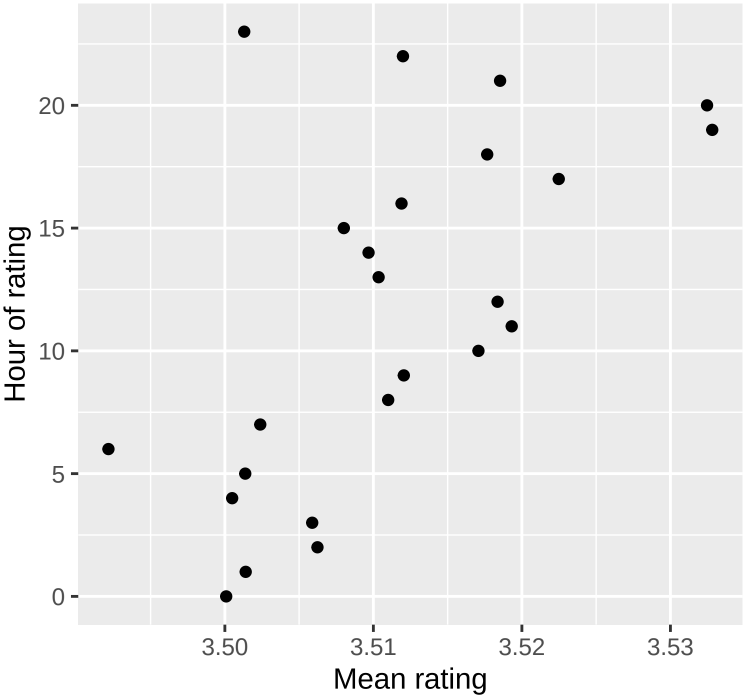



Mean rating per user
Number of users
6000
4000
2000
0
1
2
3
4
5
Mean rating


Mean rating per user (> 100 movies rated)
Number of users
2000
1500
1000
500
0
1
2
3
4
5
Mean rating



Mean rating by movie age



Mean rating by hour of rating

These graphics suggest the following behaviors:

1. Most movies are rated by 50 to 500 users in the *edx* database;

2. A large number of users only rate a few movies; users rating more than 100 movies are relatively rarer;

3. The mean rating distribution appears roughly bell-shaped, with a mean around 3.5;

4. The mean rating among heavy users (having rated more than 100 movies) appears roughly similar, suggesting the absence of a "learning curve" in the rating process;

5. While movies of the year have an about 3.5 rating, this average rating tends to lower in the 1 to 10 year-old range (less novelty among similar styles), then increases again up to 60 year-old movies (more appreciation for "classics" which speak to more generations), then tends to lower again, with movies older than 80 years at time of rating being the worst rated of all (probably due to poor ratings from younger generations towards black and white or silent movies or due their lack of special effects);

6. Movies rated during the night (between 0:00 and 8:00) tend to have a lower rating than those rated during the day, and even more so compared to those rated in the evening; this may be due to movies watched out of boredom during the night (hence with little appreciation) or of specific genres, with very critical viewers, while movies watched / rated in the early evening probably correspond to family viewings, with a higher proportion of universally appreciated movies, hence receiving higher ratings. While this phenomenon is small (the average of ratings for movies grouped by rating hour ranges from about 3.49 to 3.53), we will see later that taking this effect into account permits a (small) reduction of the RMSE.

3 Modelling approach

The modelling approach is a an iterative one. The objective is to incorporate the different effects observed above, so as to get to a model including all of them at the same time. This ultimate model including the four effects (movie, user, movie age, rating hour) will then be regularized to apply a penalty term to infrequently rated movies (the idea being that predictions on these specific movies should be considered as less reliable compared to those made on blockbusters, where an additional rating barely changes the average one).

3.1 Naive, average rating model

To get a baseline, we can start by a naive model where we would predict the rating of each movie as the average of all ratings:

$$Y_i = \mu$$

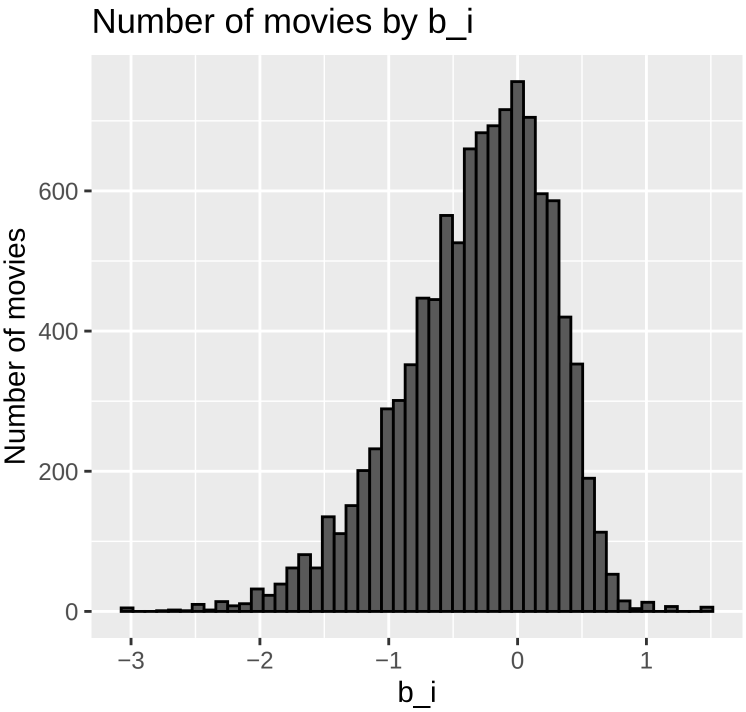
Calculating the RMSE based on this prediction and the validation set, we get:

method	RMSE
Naive, average rating model	1.061202

From the RMSE grading table provided by EdX/HarvardX, we can see that this RMSE is still quite far from the objective.

3.2 Movie effect

The first effect acknowledges the idea that not all movies are equal, and that the rating process tends to be left skewed (more movies are rated lower than the average than higher). We plot the number of movies by b_i, which represents the difference between the movie rating and the average rating :



As a consequence, our first non-naive model can be expressed as:

$$Y_i = \mu + b_i + \epsilon_i$$

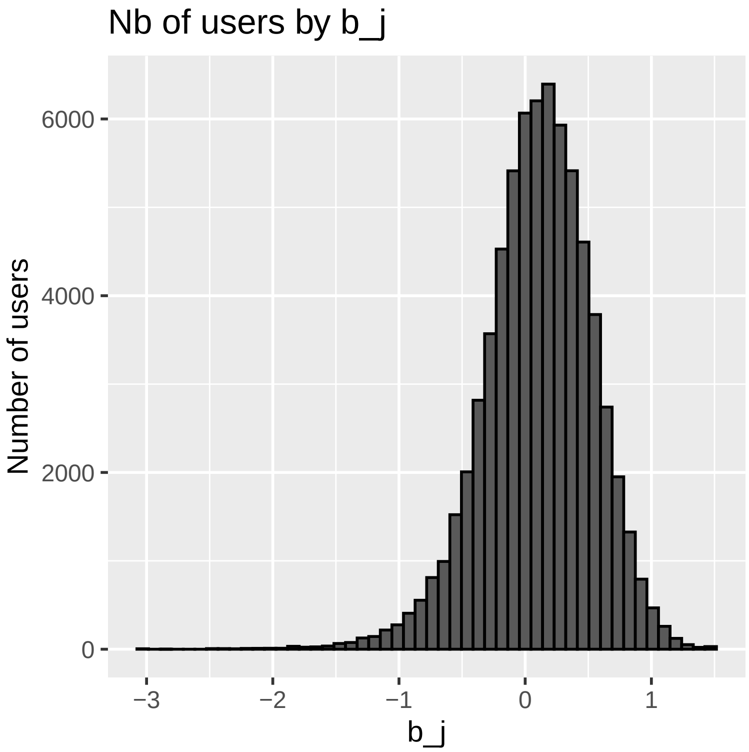
Inputing this new prediction in the RMSE function, we get:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087

which shows a clear enhancement compared to the naive model. Yet, there is still some room for improvement.

3.3 User effect

A similar approach can be followed regarding users, as some viewers tend to giver consistently higher-than-average ratings. Though not as important as the left-one for movies, there appears to be a slight right skew for users.



We thus compute a similarly expressed prediction, this time with b_j instead of b_i:

$$Y_j = \mu + b_j + \epsilon_j$$

and adding the computation result with this new prediction to our RMSE comparison table, we get:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087
User effect model	0.9783360

We observe that this user effect, while still enabling a significantly lower RMSE compared to the naive model, is not as strong as the movie one.

3.4 Combined movie and user effect model

We can now assess the impact on the RMSE of including both effects in the model. The idea of including a joint effect is that some viewers will rate poorly (below the average) most movies, including some blockbusters or acclaimed movies which would normally be rated highly.

$$Y_{i,j} = \mu + b_i + b_j + \epsilon_{i,j}$$

The RMSE comparison table with the additional line related to this combined model predictions now looks like this:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087
User effect model	0.9783360
Combined movie and user effect model	0.8850398

As expected, the combined movie and user effect does bring an improvement over the single effect models (either movie or user). We can now add more effects to try and improve the RMSE.

3.5 Combined movie, user and age effect model

As observed in the dataset analysis part, there is an inverted S-shaped relation between the averating rating of movies grouped by age at time of rating and their age. We can therefore add to your iterative model an age effect, reflecting the age of the movie at time of rating (zero when rated the same year as it was released):

$$Y_{i,j,k} = \mu + b_i + b_j + b_k + \epsilon_{i,j,k}$$

We then obtain the following cumulative RMSE table:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087
User effect model	0.9783360
Combined movie and user effect model	0.8850398
Combined movie, user and age effect model	0.8845048

We observe that incorporating this additional effect, while still lowering the RMSE, only has a marginal enhancement effect.

3.6 Combined movie, user, age and rating hour effect model

We now incorporate a ultimate effect, related to the time of the day at time of rating. As discussed in the dataset analysis part, the idea is that the "viewing mood" will be different depending on whether the movie is rated during the night or in the early evening, with more critical viewers in the first case. The model now appears as:

$$Y_{i,j,k,l} = \mu + b_i + b_j + b_k + b_l + \epsilon_{i,j,k,l}$$

and the cumulative RMSE table now looks like this:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087
User effect model	0.9783360
Combined movie and user effect model	0.8850398
Combined movie, user and age effect model	0.8845048
Combined movie, user, movie age and rating hour effect model	0.8844970

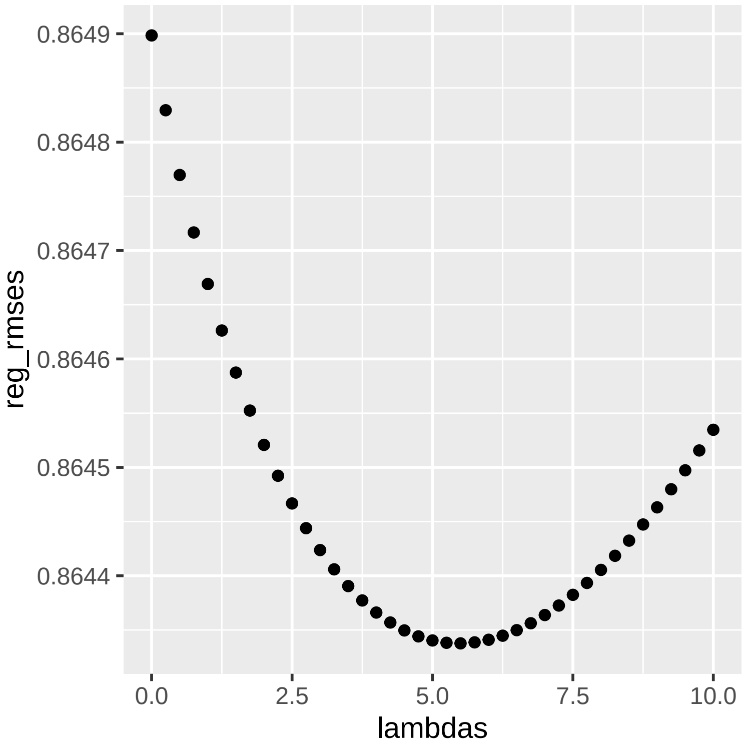
The gain on the RMSE from an additional factor now becomes really marginal. We have to use other techniques to lower the RMSE more drastically.

3.7 Regularizing the four-effect model

We now account for the different weights among the effects; some movies are very rarely rated, some users only rated a few movies, some movie ages are very infrequent and some rating hours are much less likely to occur than others. We thus write a function that takes different values of the regularizing factor and computes the respective RMSEs.

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

reg_rmses <- sapply(lambdas, function(r){

  mu <- mean(edx$rating)

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

  b_j <- edx %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_j = sum(rating - b_i - mu) / (n() + r))

  b_k <- edx %>%
    left_join(b_i, by="movieId") %>%
    left_join(b_j, by="userId") %>%
    group_by(movie_age) %>%
    summarize(b_k = sum(rating - b_i - b_j - mu) / (n() + r))

  b_l <- edx %>%
    left_join(b_i, by="movieId") %>%
    left_join(b_j, by="userId") %>%
    left_join(b_k, by="movie_age") %>%
    group_by(rate_hour) %>%
    summarize(b_l = sum(rating - b_i - b_j - b_k - mu) / (n() + r))

  pred_mvusagehour_reg <-
    validation %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_j, by = "userId") %>%
    left_join(b_k, by = "movie_age") %>%
    left_join(b_l, by = "rate_hour") %>%
    mutate(pred = mu + b_i + b_j + b_k + b_l) %>%
    .$pred

  return(RMSE(pred_mvusagehour_reg, validation$rating))
})
```

The plot relating the different values of lambdas to their respective RMSEs looks like this:



The minimum is reached for lambda at :

```
lambdas[which.min(reg_rmses)]
```

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

We can now add the minimum value of these regularized RMSEs in our iterative table:

method	RMSE
Naive, average rating model	1.0612018
Movie effect model	0.9439087
User effect model	0.9783360
Combined movie and user effect model	0.8850398
Combined movie, user and age effect model	0.8845048
Combined movie, user, movie age and rating hour effect model	0.8844970
Regularized (lambda = 5.5) combined movie, user, movie age and rating hour effect model	0.8643376

According to the grading scale for the RMSE provided by EdX / HarvardX, we have now reached our objective of an RMSE lower than 0.86490.

4 Conclusion and perspectives

We have opted for constructing an iterative model, incorporating an increasing number of factors, rather than using various, distinct methods. The objective was to observe the marginal gain obtained by adding each additional factor. We saw that these gains mainly came from the movie and user effects, and that regularizing the complete, four-effect model enabled another significant gain, reaching our tarhet of an RMSE lower than 0.86490.

We have deliberately opted not to use the movie categories vector, for various reasons:

1. It involves some treatment (separating the different genres, accounting for potential movies without genre, considering the hierarchy between the multiple genres of a movie) which may carry some subjectivity;

2. Each movie category may itself involve a subjecive dimension: some rather wide categories such as comedy, action or drama may refer to rather different actual types of movies;

3. These categories may also depend on the eye of the person responsible for assigning categories: a comedy movie for an appraiser may be rather considered as a romance one for another, and as both for a third one;

4. There may be some colinearity among genres: Toy Story" is registered under five different categories, but the sole “Animation” one may be considered as sufficient to correctly label the movie genre;

5. These movie categories may also not be stable throughout time: a movie of the 1950s **today** labelled as drama may not necessarily have been considered as such at that time.

Obviously, adding this effect in our model could further improve the RMSE, but as discussed, we felt that the methodological uncertainties outweighed the potential RMSE gain, considering we had already reached our quantitative target in that matter.

Another approach could have been a K-nearest neighbours one, as the data types would fit well with this kind of approach. We however feel that, given the “curse of dimensionality” associated with this method, reaching our objective of an RMSE lower than 0.86490 would have involved a significantly higher computing time.

5 Technical annex

Computing environment:

```
## [1] "CPU: Intel64 Family 6 Model 142 Stepping 10, GenuineIntel"
```

```
## [1] "OS: Windows_NT"
```

```
## [1] "R version: 4.0.0"
```