MovieLens Report

Bertrand Jager

02/03/2020

Introduction

This dataset is a condensed version of the dataset used for the Netflix prize, which was organized to engage teams of data scientists to enhance Netflix's prediction algorithm. Prize was awarded in 2009 to the BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06%.
It is organized as a long table of 6 columns, and aproximately 10 M rows.
Each line is information about a rating provided by one Netflix user for one movie.
The columns are:
* user ID
* movie ID
* movie title
* movie genre (drama, romance, etc. . .)
* timestamp of the moment the rating was provided
* rating

The rating itself represents a number of stars (between 1 and 5), but half stars can be provided as well. Therefore it can be 1, 1.5, 2, 2.5, . . . , 4.5, 5. The ratings originate from nearly 70.000 users, and rate around 10.700 movies. After download, the dataset is split into:
* the **edx** dataset that we will use to build the machine learning algorithm
* the **validation** set that will be used to assess the accuracy of the algorithm produced

Methods

We first split the edx dataset into a training set and a test set. This test set will be used with each model produced, in order to get an idea of its accuracy, but also to determine the optimal parameters for cross validation. Once a model will have been developped and trained, it will be assessed, as a final stage, on the validation set. It is important to ensure that the validation set is never used for training, nor for optimizing parameters.
The general approach used in this study is to enhance on Professor Irizarry's approach which uses biases on movie and users as well as regularization. By using the same steps, and then by determining additional biases, we will gradually exceed the threshold of 0.86490 (maximum number of points) on the test set.
For the remainder of this document, a prediction for movie i by user u will be represented as

$$Y_{u,i}$$

Results

Let's first follow Professor Irizarry's method in the course, with 3 successive steps:
a) predict each rating to be the mean of ratings provided by all users to all movies (mu) plus some noise (epsilon). Which can be written mathematically as:

$$Y_{u,i} = \mu + \varepsilon_{u,i}$$

Once a mean has been calculated on the training set, we use it to predict the ratings of the test set. We then compare it to the actual ratings of the test set using the RMSE function:

Table 1: Just the average

Methods	results
Just the average	1.059904

b) enhance the prediction by calculating a movie effect (some movies are very much appreciated by many raters while others are consistently rated with bad scores)

$$Y_{u,i} = \mu + b_u + \varepsilon_{u,i}$$

Table 2: Movie Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429

c) enhance the prediction by adding a user effect (some users are generous, others are cranky)

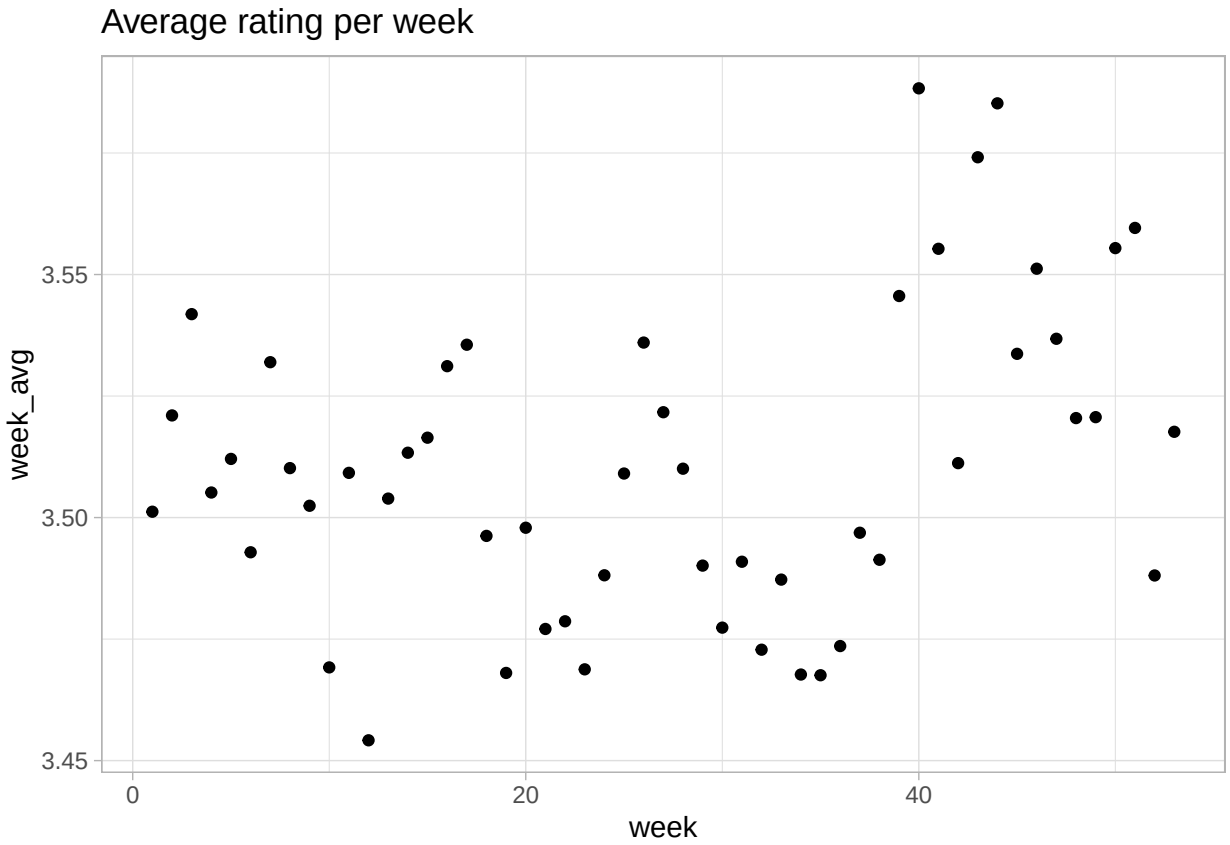
$$Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i}$$

Table 3: Movie+User Effect

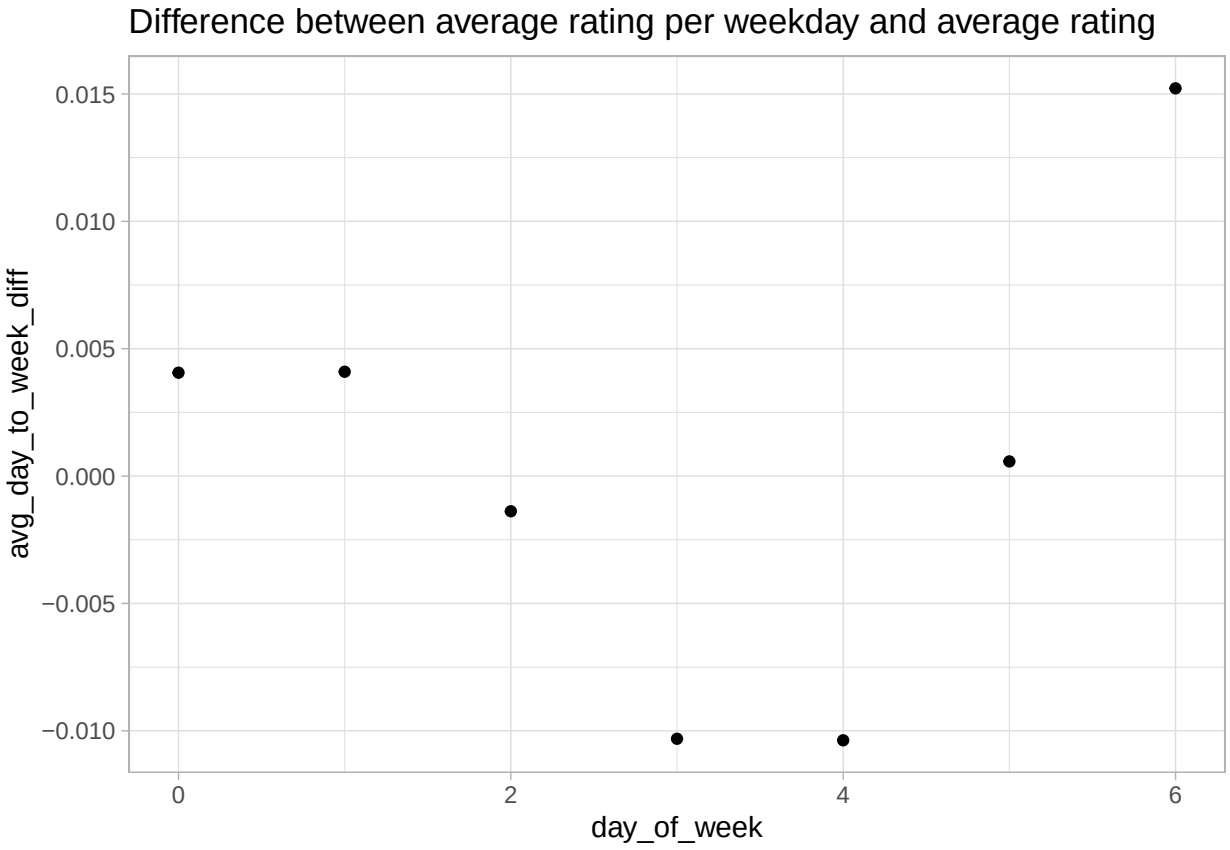
Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319

After following these steps, we get an accuracy of 0.865932 on the test set. Which is not enough. Therefore, to enhance the prediction, we have to look for additional biases that could be used to enhance the prediction.

Time is an obvious candidate for predictor. It seems reasonable that people have different moods throughout the year (for example: more generous during the end of year celebration, more agressive after paying taxes, more relaxed during the summer break, etc. . .). For this reason it makes sense to study the average rating given each week:



There is a clear indication that the moment of the year matters. But its pattern is confused, and therefore difficult to exploit. That is why I decided to leave it appart for the time being and come back to it if no simpler technique could be found.
An other influence time has on rating is through weekdays. Because people are more relaxed during the weekend, fresh on Monday, but exhausted on Friday, the weekday the rating is given can also have influence. To assess its influence, let's plot the difference between the average rating given to a movie for each day, and its annual rating. The result shows a clear difference per day, suggesting that people provide significantly lower ratings on Wednesday, and more generous ratings on Saturday for example.



This leads to the following model:

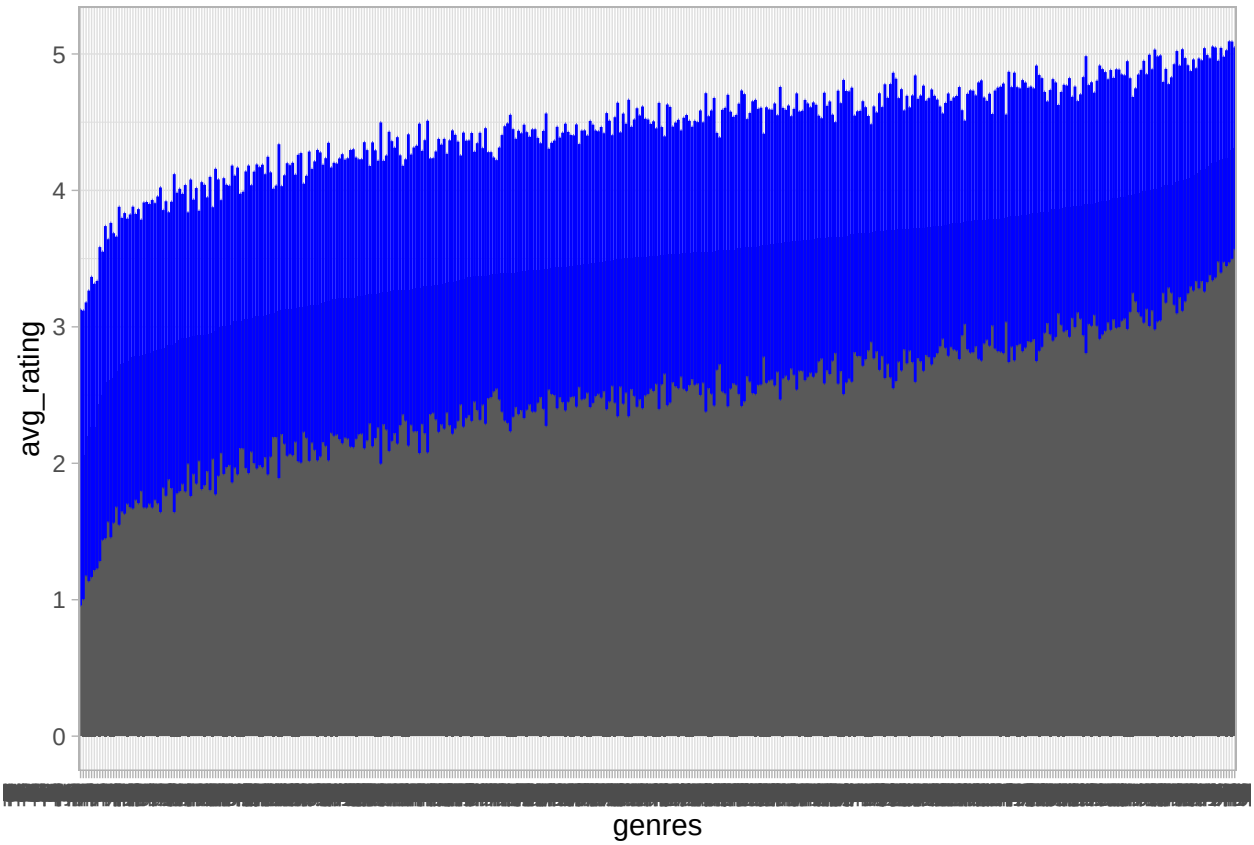
$$Y_{u,i} = \mu + b_i + b_u + d_u + \varepsilon_{u,i}$$

We implement this model, and again test it on the test set.
As the model is linear, it produces predictions that are below 1 and others taht are above 5. Which means no sense for ratings. Therefore all predicted ratings that are below 1 are "rectified" to 1, and predicted ratings above 5 are "rectified" to 5.

Table 4: Movie + User + Weekday Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319
Movie + User + Weekday Effect	0.8656712

The result obtained (0.865671) is worth 10 points for the capstone project. Let's try to do better.
To achieve that, let's examine information on the genres, because it is plausible that certain genres induce a more positive rating than others. Does the data support that view?
If we define a category as any existing combination of genres in the data frame, and restrict this analysis to categories that have at least 1000 elements. What are the average and standard errors for each category?
A bar chart plot of mean ratings per category (in gray) with standard errors (in blue) makes it quite obvious that the genre has influence on the rating:



We can implement it, leading to the following model:

$$Y_{u,i} = \mu + b_i + b_u + d_u + g_{u,i} + \varepsilon_{u,i}$$

As previously, let's train the model on the training set.
Once done, it is interesting to assess its potential influence by calculating the maximum influence a category has on ratings:

```
## [1] 0.1119886
```

It is a material influence. Much stronger than the day of the week. Looks promising...

Let's assess it on the testing set. We start by calculating the predictions for the test_set. As mentioned for the day of the week, the range of outcomes is wider than 1 to 5:

```
## [1] -0.7239339  6.0895018
```

It is easy to rectify this by affecting 1 to predicted values below 1, and 5 to predicted values above 5. With these rectified values, one can assess the model's accuracy on the test set.

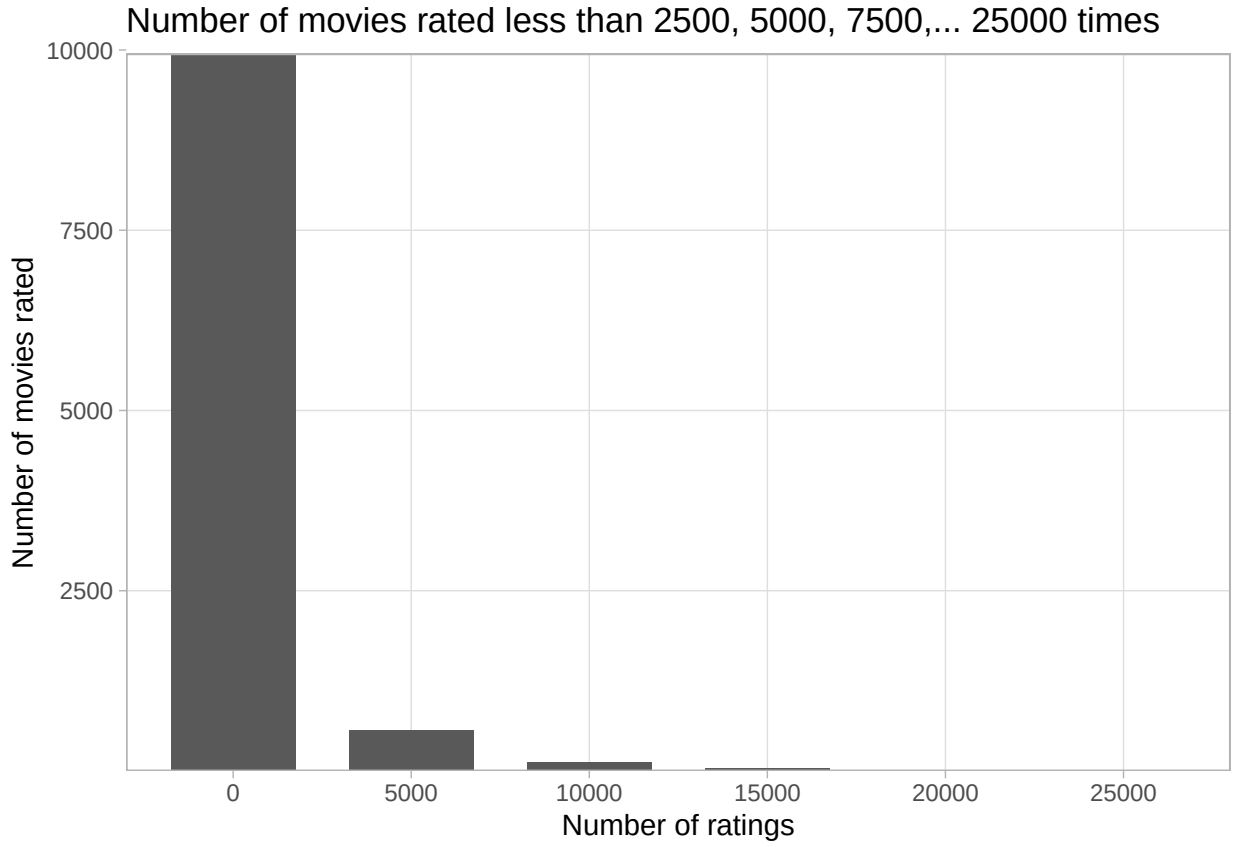
Table 5: Movie + User + Weekday + Category Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319
Movie + User + Weekday Effect	0.8656712
Movie + User + Weekday + Category Effect	0.8653638

The accuracy of our model on the test set is therefore 0.865364, which is not yet good enough.
An effective way to enhance it is to understand that by calculating a "movie bias" by averaging ratings given in the training set, it accounts for the same degree of confidence for movies that have been rated by a great number of users, as to movies that have only been rated by a handful or even a single one. Which is wrong since we can't be sure that a handful will rate consistently with the taste of others. Therefore the model should provide less confidence in movie biases that have been calculated based on few ratings. For this we can use regularisation, to penalize these ratings. We can easily verify that the range in the number of ratings provided to movies is very large:

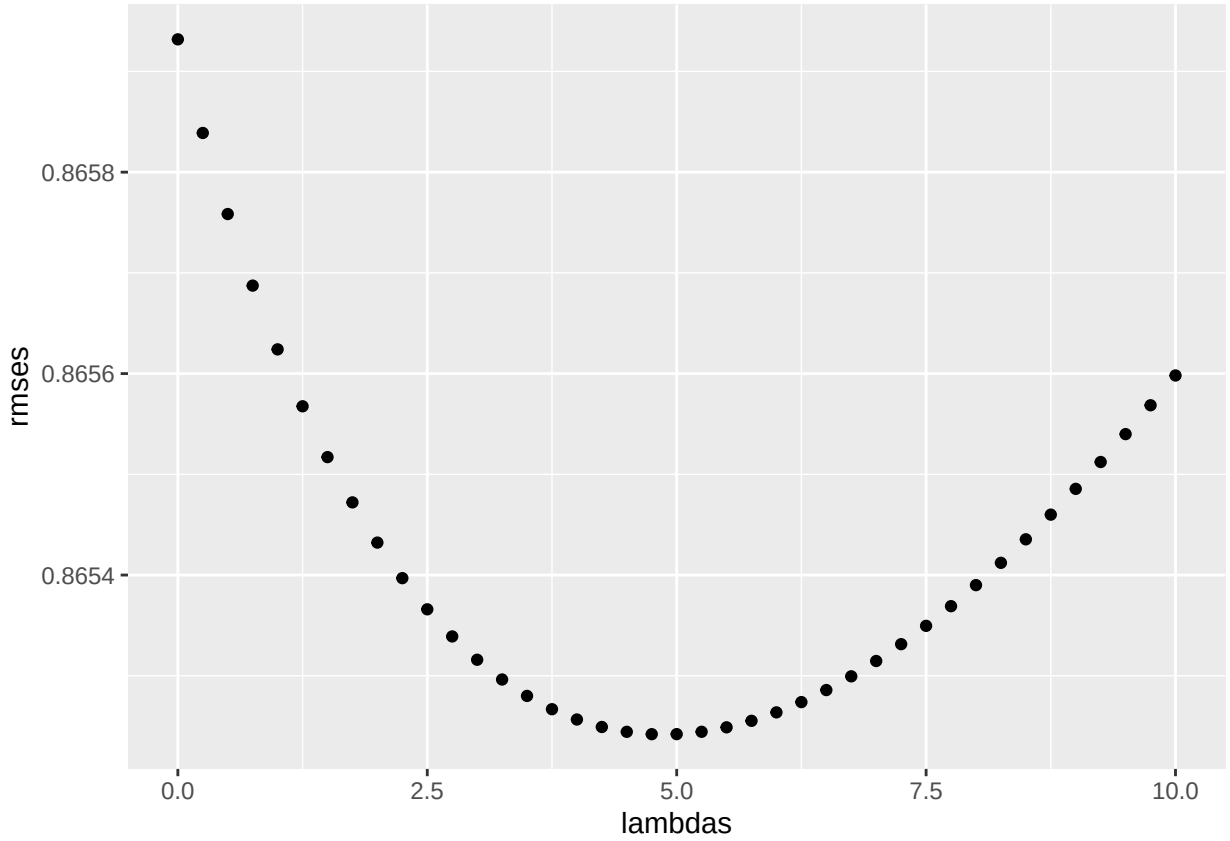
```
## [1]     1 25115
```

The distribution of these ratings is also telling. Let's have a look at it: how many movies have been rated between 0 and 5000 times, how many have been rated between 5000 and 10.000, etc...



The vast majority have been rated between 0 and 2500 times.
It must be noted that regarding categories, only categories for which there was more than 1000 ratings provided were considered in the previous model.

For regularization, we need to choose a certain lambda value. To optimize this value, we train the model on the training set, with different values of lambda, ranging from 0 to 10. Then we evaluate the accuracy of these models on the test set. The lambda value providing the best accuracy will be used going forward.



The plot shows an optimal lambda at 4.75 leading to an accuracy of 0.865242 on the test set, good enough for our objectives.

It must be noted that since the prediction model used here is linear, some of the values predicted are out of bound (i.e. outside the 1 to 5 star netflix rating scheme). I have "rectified" these values, by replacing any rating above 5 by 5, and any rating below 1 to 1. This technique provided better results on the test set.

Table 6: Regularized Movie + User + Weekday + Category Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319
Movie + User + Weekday Effect	0.8656712
Movie + User + Weekday + Category Effect	0.8653638
Regularized Movie + User + Weekday + Category Effect	0.8652420

This technique having been developped on the training set and providing results above target, it was then possible to validate it on the validation set. But before that, an enhancement of the predictive power of our ML algorithm was obtained by training the algorithm on the entire edx set and not only on the training set. This does not break the golden rule of not using the same data to train and to test, as the edx set is now used for training, while the validation set remains unused until the validation phase.

To increase the volume of training material, we can use the entire edx set, including the test set. This will NOT break the golden rule of not training on the validation set.

Table 7: Validation for: Regularized Movie+User+Weekday+Category Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319
Movie + User + Weekday Effect	0.8656712
Movie + User + Weekday + Category Effect	0.8653638
Regularized Movie + User + Weekday + Category Effect	0.8652420
Validation for: Regularized Movie+User+Weekday+Category Effect	0.8648201

Accuracy on the validation set is 0.86482, which is already below the target of 0.86490 to get 25 points.

But as previously, we can observe that some of the predictions can easily be enhanced since they are below 1 or above 5, which is the range of possible ratings: -0.4105674, 5.9996051 We can therefore, as previously, reassign predictions that are below 1 to 1 and those that are above 5 to 5.

Table 8: Validation for: Regularized then Rectified Movie+User+day+Category Effect

Methods	results
Just the average	1.0599043
Movie Effect	0.9437429
Movie+User Effect	0.8659319
Movie + User + Weekday Effect	0.8656712
Movie + User + Weekday + Category Effect	0.8653638
Regularized Movie + User + Weekday + Category Effect	0.8652420
Validation for: Regularized Movie+User+Weekday+Category Effect	0.8648201
Validation for: Regularized then Rectified Movie+User+Day+Category Effect	0.8646934

Conclusion

The objective to create a machine learning algorithm going below RMSE = 0.86490 has been reached, by extending the 2 methods used in the course (modeling with biases and regularization). We had to add 2 more biases (category effect and weekday effect). Just as in the course, it was necessary to regularize the results with a certain lambda that ensured to penalize the prediction power of movies unfrequently rated.

Other techniques could have been used, for example Principal Component Analysis.