Movilens_Project - HarvardX: PH125.9x Data Science

Fabricio Irabuena

April - 15 - 2019

OVERVIEW

Recommendation systems use ratings that *users* have given *items* to make specific recommendations. Companies that sell many products to many customers permit these customers to rate their products. Items for which a high rating is predicted for a given user are then recommended to that user.

In this analysis we focus on Netflix, Which uses a recommendation system to predict how many *stars* a user will rate a specific movie, being 0.5 stars the minimun rate and 5 stars the maximun.

The original data set was constructed by the convination *movies* and *users*, Where each observation represents a rating given by one user to one movie. This data is divided in a *validation* set with 10% of data (it will be consider as unknown data and will only be used for the final evaluation) and the *edx* set with the 90% (considered as all the known data).

edx

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance

dimensions

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

The main goal is to predict the ratings given for the convination of user and movie within the validation set, using the *RMSE* as a performance messure, with a desireable deviation below of 0.875 stars.

With the mentioned loss function defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

Where $y_{u,i}$ as the rating for movie i by user u, the prediction as $\hat{y}_{u,i}$ with N being the number of user/movie combinations and the sum occurring over all these combinations.

The Method used for the analysis follows the steps:

1 Preparing the data.

2 Data exploration and visualization.

3 Presenting the models, calculating the variables and evaluating the results.

4 Cross validation and parameter optimization.

5 Final evaluation of the model's predictions on the *validation set.*

PREPARING DATA

Creating the *train set* and *test set* from the *edx* data, where the proportions is selected to fit with our *validation set* size.

```
## [1] "dimentions"
```

train_set

```
## [1] 8000047       6
```

test_set

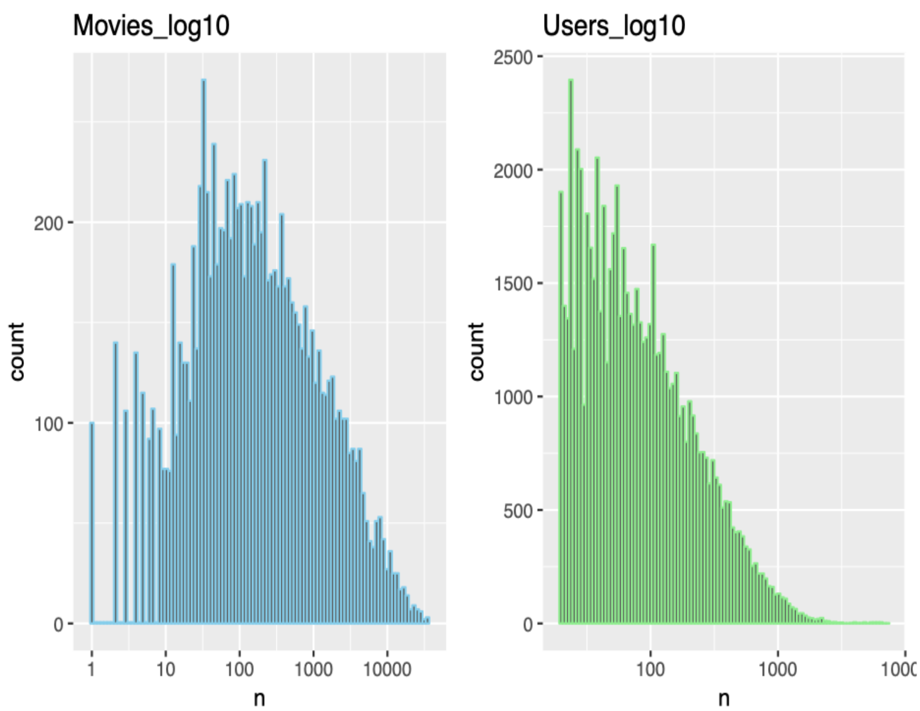
```
## [1] 1000008       6
```

DATA EXPLORATION AND VISULIZATION

We can see the number of unique *users* that provided ratings and how many unique *movies* were rated:

n_movies	n_users
10677	69878

Some *movies* get rated more than others and some *users* are more active than others as well.



Similarly, some *rates* are more frecuently given than others.

Table 3: Table continues below

rating	4.0	3.0	5.0	3.5	2.0	4.5
num_rates	2588430	2121240	1390114	791624	711422	526736
rates_share	28.8%	23.6%	15.4%	8.80%	7.90%	5.85%

rating	1.0	2.5	1.5	0.5
num_rates	345679	333010	106426	85374
rates_share	3.84%	3.70%	1.18%	0.949%

Ratings Summary

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.5	3	4	3.512	4	5

MODELING APPROACH

MEAN MODEL

A model that assumes the global mean *mu* as the only rating for all *movies* and *users* with all the differences explained by random variation:

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

```
## [1] "mu= 3.5126"
```

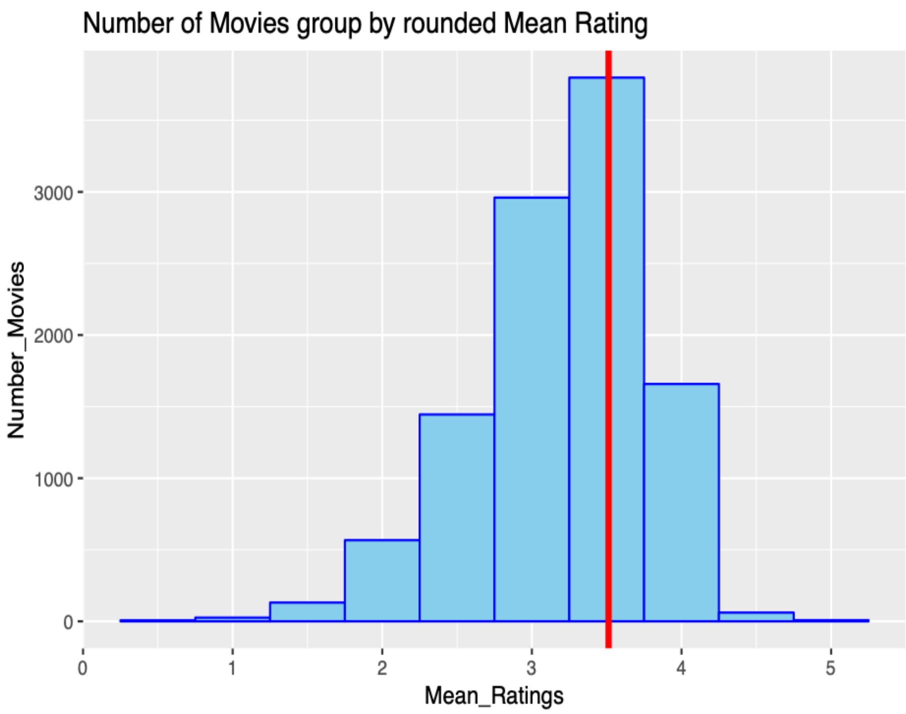
Method	RMSE
Simple-Mean Model	1.0613

MOVIE EFFECT MODEL

In addition, different movies are rated differently and each *movie* has its own mean. We can have an idea of how they are distributed grouping them by a rounded mean with a range +/- 0.25 from its center:

```
##  [1] "[0.5-0.75)"   "[0.75-1.25)"  "[1.25-1.75)"  "[1.75-2.25)"
##  [5] "[2.25-2.75)"  "[2.75-3.25)"  "[3.25,-3.75)" "[3.75,-4.25)"
##  [9] "[4.25-4.75)"  "[4.75-5)"
```

mean_rate	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
num_rates	7	26	131	567	1445	2960	3798	1658	61	8



We can extend the simple mean model by adding the term b_i to represent average ranking for each *movie* i:

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

Where:

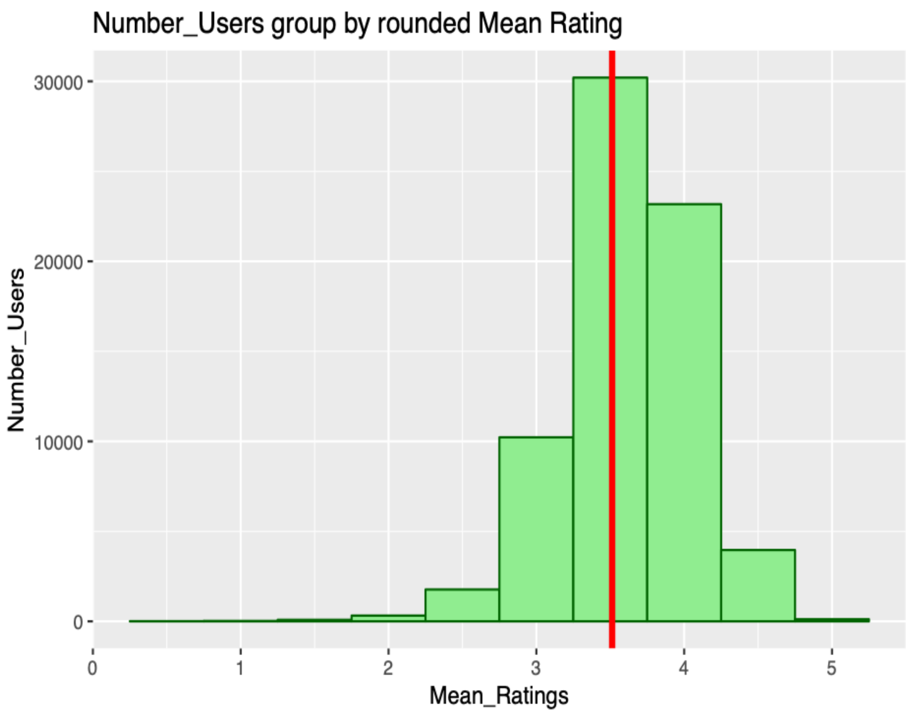
$$b_i = \frac{1}{n_i} \sum_{i=1}^{n_i} (y_{u,i} - \hat{\mu})$$

Method	RMSE
Simple-Mean Model	1.0613
Movie-Effect Model	0.94415

MOVIE-USER EFFECT MODEL

Following the same aproach as for *movies*, there is substantial variability across *users* as well.

mean_rate	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
num_rates	6	20	83	320	1770	10223	30203	23174	3963	116



Its possible to add the *user* effect term b_u to previous model.

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

Where:

$$b_u = \frac{1}{m_u} \sum_{u,i}^{m_u} (y_{u,i} - \hat{\mu} - b_i)$$

Method	RMSE
Simple-Mean Model	1.0613
Movie-Effect Model	0.94415
Movie-User-Effects Model	0.86613

REGULARIZED MOVIE USER EFFECT MODEL

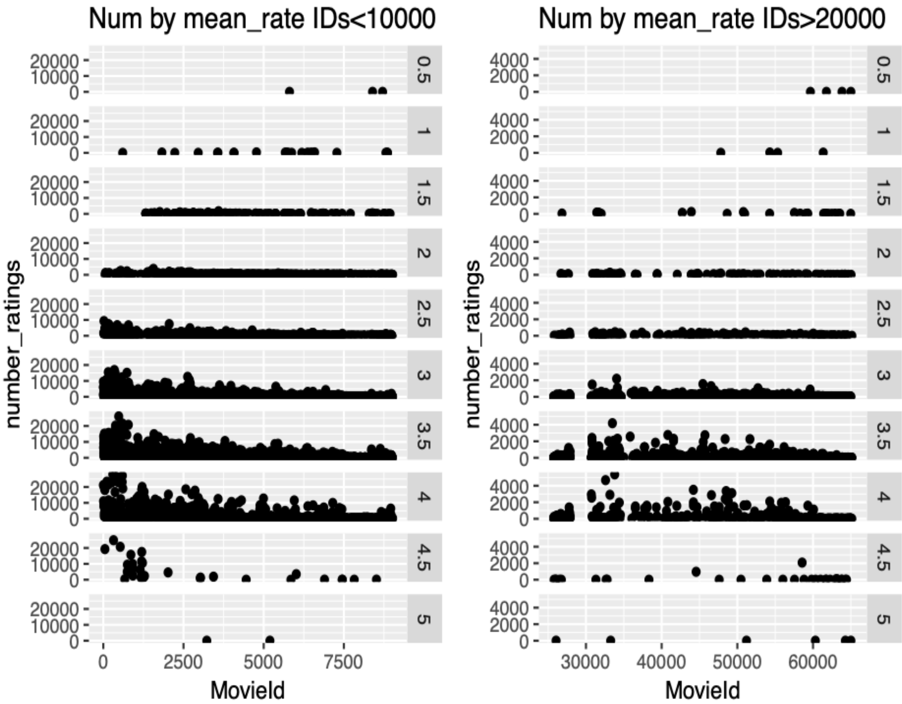
In the scenario that few ratings per user or movies are given, it is better to be conservatives and consider the *user* or *movie* effect just partialy, because it may be over or under rating be them.

Here we have some examples, where the minimum and maximum values of the *mean* by *movies* are often given by a few number of *rates*

movieId	title	mean_rate	num_rates
3226	Hellhounds on My Trail (1999)	5	1
51209	Fighting Elegy (Kenka ereji…	5	1

movieId	title	mean_rate	num_rates
60309	Along Came Jones (1945)	5	1

movieId	title	mean_rate	num_rates
5805	Besotted (2001)	0.5	1
8394	Hi-Line, The (1999)	0.5	1
8707	Grief (1993)	0.5	1

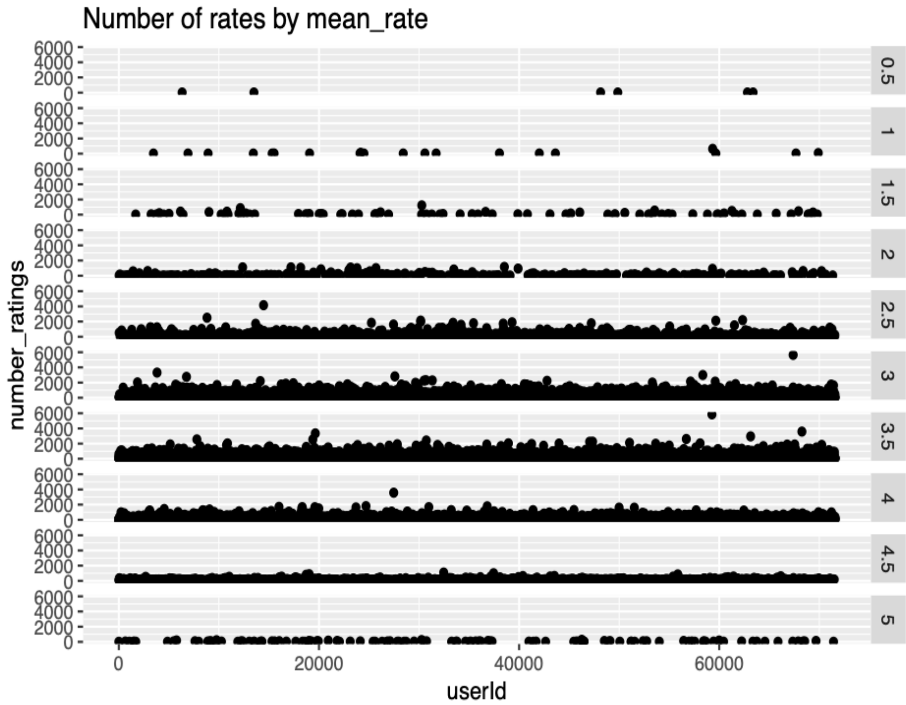


Note: *the data was splited just for visualization purposes, and there are not movies' IDs in between codes 10000 and 20000*

Similarly, on the *users* side for those who gave few *rates* For instance:

userId	mean_rate	num_rates
26308	5	14
52674	5	14
7984	5	15

userId	mean_rate	num_rates
49862	0.5	16
13496	0.5	17
63381	0.5	17



The goal of penalized regression is to control the total variability of the movie effects: $\sum_{i=1}^{n} b_i^2$. Specifically, instead of minimizing the least square equation, we minimize an equation that adds a penalty:

$$\frac{1}{N}\sum_{u,i}(y_{u,i}-\mu-b_i)^2+\lambda\sum_i b_i^2$$

The first term is just least squares and the second is a penalty that gets larger when many b_i are large.

$$\hat{b}_i(\lambda)=\frac{1}{\lambda+n_i}\sum_{i=1}^{n_i}(Y_{u,i}-\hat{\mu})$$

where n_i is the number of *ratings* made for *movie i*. This approach will have our desired effect: when our sample size n_i is very large, a case which will give us a stable estimate, then the penalty λ is effectively ignored since $n_i+\lambda \approx n_i$. However, when the n_i is small, then the estimate $\hat{b}_i(\lambda)$ is shrunken towards 0. The larger λ, the more we shrink.

We can use regularization for the estimate user effects as well. We are minimizing:

$$\hat{b}_u(\lambda)=\frac{1}{\lambda+m_u}\sum_{u=1}^{m_u}\left(Y_{u,i}-\hat{\mu}-\hat{b}_i\right)$$

where m_u is the number of *ratings* made for *user u*. This approach will have our desired effect: when our sample size m_u is very large, a case which will give us a stable estimate, then the penalty λ is effectively ignored since $m_u+\lambda \approx m_u$. However, when the m_u is small, then the estimate $\hat{b}_u(\lambda)$ is shrunken towards 0. The larger λ, the more we shrink.

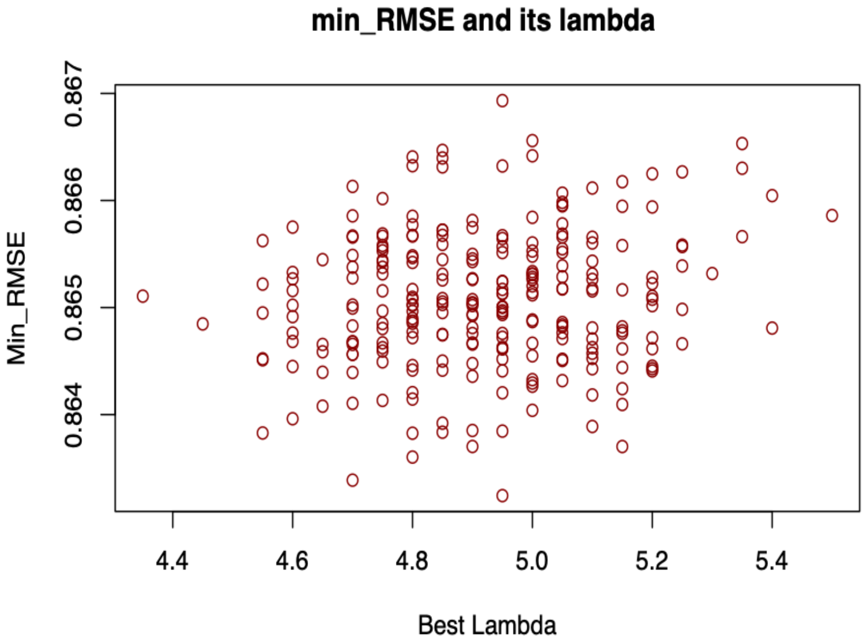
$$\frac{1}{N}\sum_{u,i}(y_{u,i}-\mu-b_i-b_u)^2+\lambda\left(\sum_i b_i^2+\sum_u b_u^2\right)$$

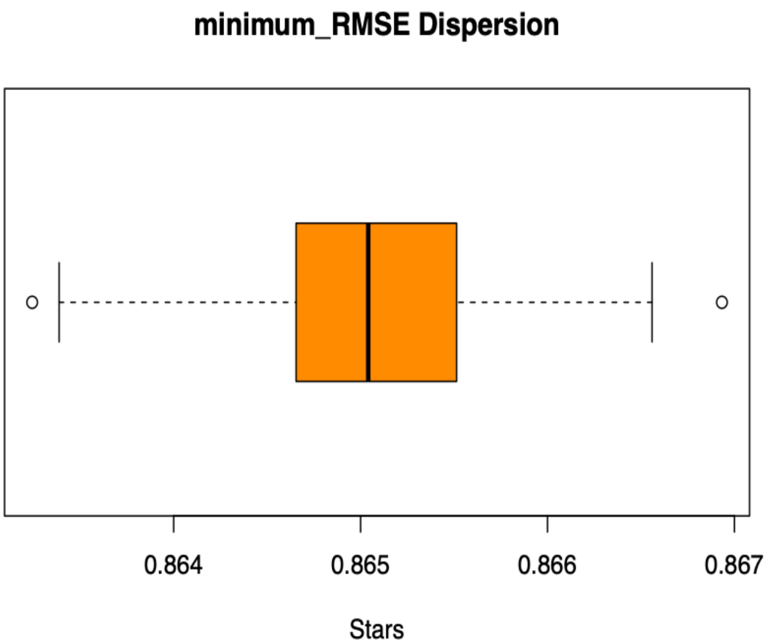
In the next section, we will be implementing full cross validation just on the *train set*, without using the *validation set* until the final assessment.

CROSS VALIDATION AND PARAMETER OPTIMIZATION

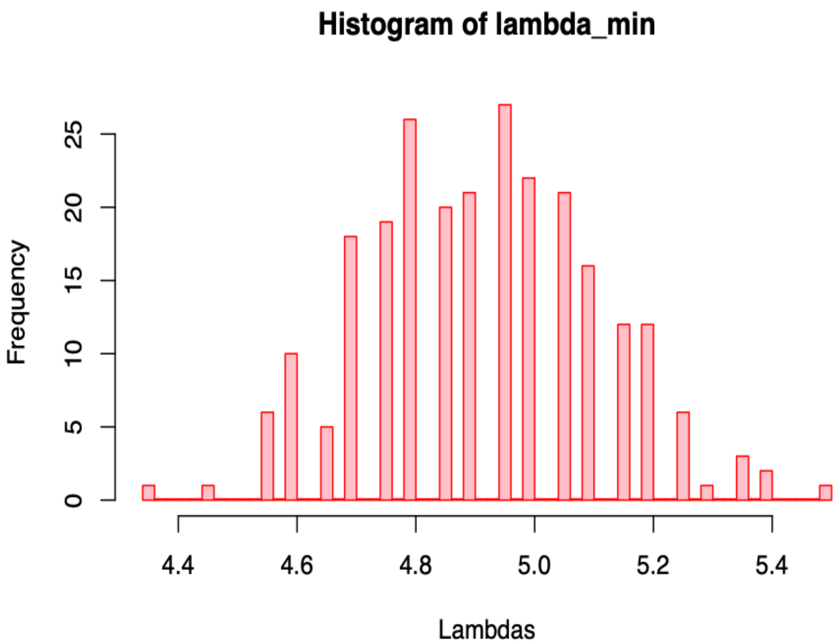
We are taking many new *samples* of the *edx set* to pick the values of λ which minimize the *RMSE* on each sample, the *size* is the same as in original setup. Within this case 250 new *samples* were taken.

Choosing the penalty terms λ





Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.8632	0.8647	0.865	0.8651	0.8655	0.8669



```
## [1] "Optimized lambdas"
```

Table 16: Table continues below

4.35	4.45	4.55	4.6	4.65	4.7	4.75	4.8	4.85	4.9	4.95	5
1	1	6	10	5	18	19	26	20	21	27	22

5.05	5.1	5.15	5.2	5.25	5.3	5.35	5.4	5.5
21	16	12	12	6	1	3	2	1

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.35	4.8	4.9	4.918	5.05	5.5

Testing the model results:

Now we can test the *Regularized-movie-user-effect-model* agaist a the original *test set* using the optimized value found for lambda $\lambda = 4.9$.

Method	RMSE
Simple-Mean Model	1.0613
Movie-Effect Model	0.94415
Movie-User-Effects Model	0.86613
Regularized-Movie-User Effect Model	0.86552

3. *RESULTS*

Finally, we are able to evaluate our final *Regularized-Movie-User Effect Model* using the *edx set* with our *validation set*, where the last one remains yet as unknown data, and reveals the results under a possible real scenario.

Method	RMSE
Final Model vs validation set	0.86482

CONLUSION

Following the steps we learned in the course, we have improved the base line of a *Mean-Model* for predicting ratings, going through the *user* and *movies effects* and finally arriving to the *Regularized-Movie-User Effect Model*, which was optimized for $\lambda = 4.9$ and reflected a *RMSE* ≈ 0.865 on the *validation set*, in other words, a deviation of 0.86 *stars* from the predictions.

Please follow this link to have access to the project files: https://github.com/AmadoLabX/Data-Science-HX