Movie Recommendation System

HarvardX PH125.9x

Ruben Campos

25/09/22

Contents

1. Introduction

A recommendation system is an information filtering system that tries to predict the preference that a user would give to an item that want to consume in some way: be it clothes, electronics or entertainment. In the exercise that arises, it is about movies.

Similar systems, obviously much better in terms of complexity and accuracy, have been used for years in the entertainment industry. It is obvious that Netflix was practically the pioneer in this aspect, but other companies already used this system for the articles offered to the public, such as Amazon. In the end, the underlying idea of these systems is to always offer the customer what they think they will want, leaving little room for them to choose other things that are not in their interest.

We are going to work, in order to develop our own recommendation system, with the data provide in *MovieLens dataset*. It was first released in 1.998 by a team from the University of Minnesota, where collected the data that corresponds to thousands of people interactions with an online movie-recommendation system, a process in which the users are required to input their movie preferences into the system.

The success metric of our system will be the *Root Mean Square Estimate (RMSE)*. We will try to be able to determine movie ratings on a scale of 0.5 to 5 stars, measuring the RMSE on the same scale. It should be noted that a RMSE trending at 0 indicates that we are on the right track.

The target of this project is to achieve a RMSE < 0.87750

2. Project methodology

The process we have followed to develop this project is as follows:

• Download the data set.
• Split it into two different ones: edx and validation.
• Explore the data. Wrangling if necessary.
• Modeling concept.
• Evaluate the effectiveness via RMSE by splitting edx dataset into edx_train and edx_test, creating different models and cheking performance using edx_test.
• Retraining best performance model and validation.

3. Data source overview

3.1 The data sets

Movielens dataset is split in two:

• edx dataset for model training.
• validation dataset for model evaluation.

The edx dataset have a total amount of 9.000.055 and 6 variables (*userId*, *movieId*, *rating*, *timestamp*, *title*, *genres*) corresponding to each one of the columns in the data set. We have to bear in mind each record corresponds to a rating of *one movie* by *one user*. In the other hand, validation dataset consists of 999.999 records and the same variables.

As mentioned, edx dataset have six columns. Those are what must be considered the predictor variables:

• userId
• movieId
• timestamp
• title
• genres

While column rating is the outcome, as is the result that we want to predict. Following, we show a sample corresponding to edx dataset records structure:

Table 1: Sample of records structure in edx dataset

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

The edx dataset predictors *userId* and *movieId* are unique identifiers corresponding to each user and movie respectively; *timestamp* applies to the case a specific user has rated a specific movie; *title* includes each movie´s title and the year that was released; *genres* are the films genre they can be associated.

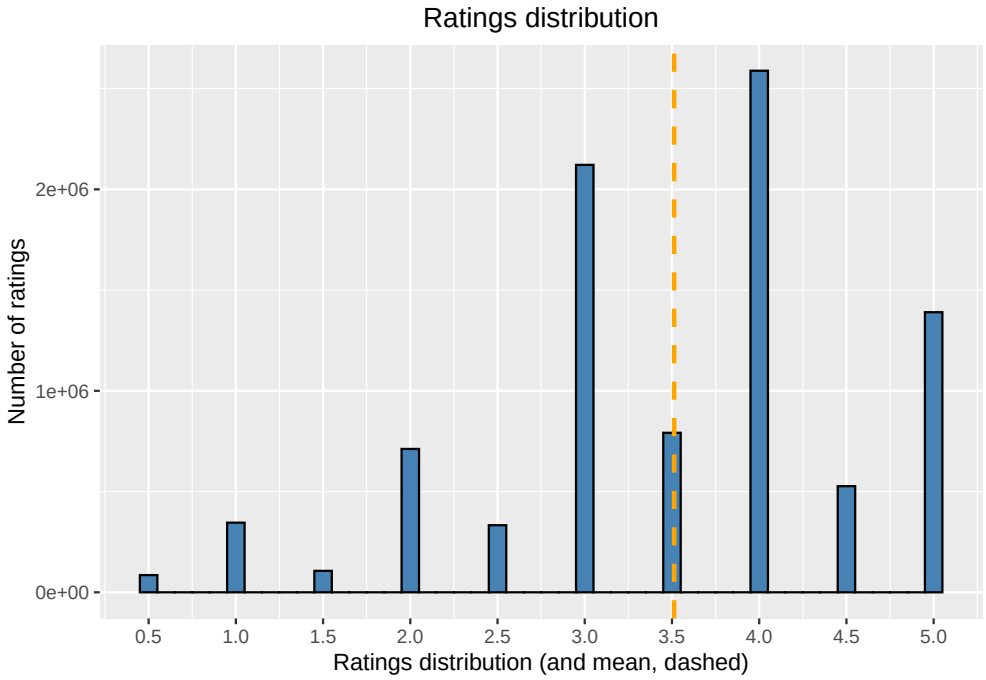
3.2 Data exploration

3.2.1 Movies & Ratings

We consider that movies and ratings should be analyzed jointly, since these parameters are intimately related and are completely dependent on. And although significant data is obtained separately, we have not seen that they are conclusive, contrary to what we are going to see next, when doing it together.

The edx dataset has a total 10.677 different movies, while validation dataset has 9.809.

The rating is a number provided by users and is between 0.5 (minimum, worst rating) and 5 (maximum, best rating). First step is to see the distribution of ratings (and mean rating, dashed) in order to define the mean value of the valuation on the ratings, allowing us to have a first perspective on the positivity or negativity of ratings:

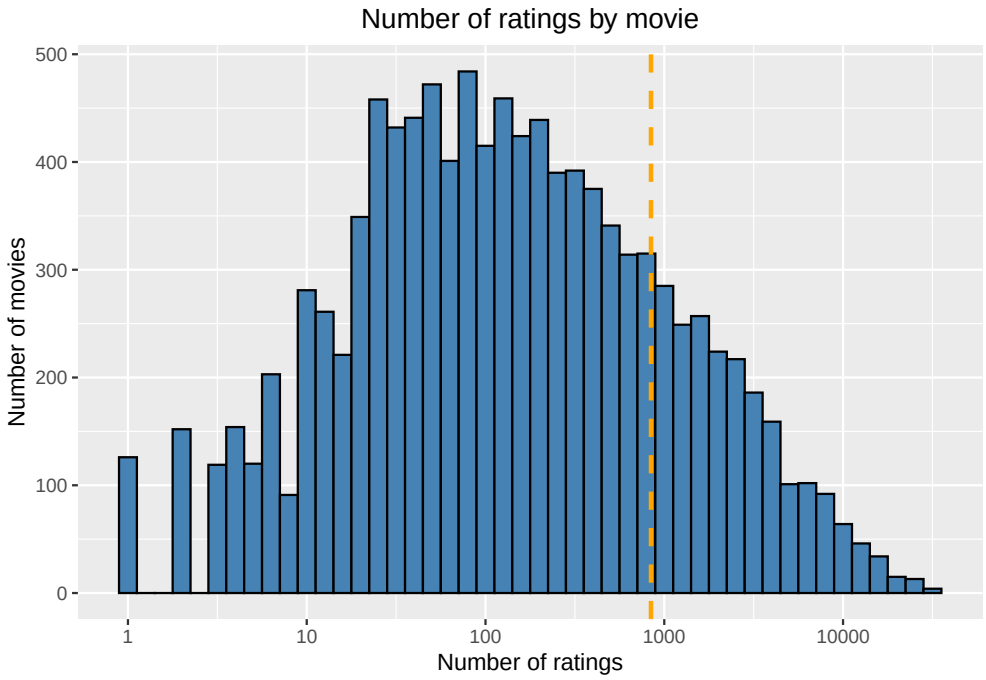


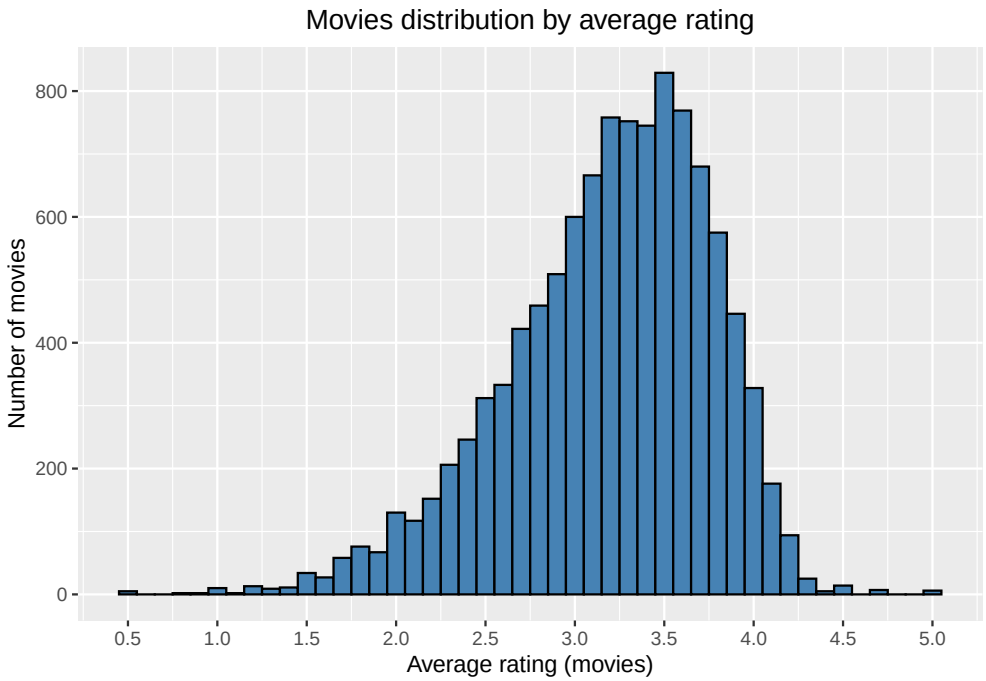
Our first conclusions by analyzing the info provided from this figure, are:

- Overall average rating at edx dataset is 3.51 It can be considered a positive rating value.
- Top 3 ratings from users are (from smallest amount of ratings to largest) : 5, 4, 3. All three values are positive.
- Bearing in mind both points above, we assert users tend to value positively the films.

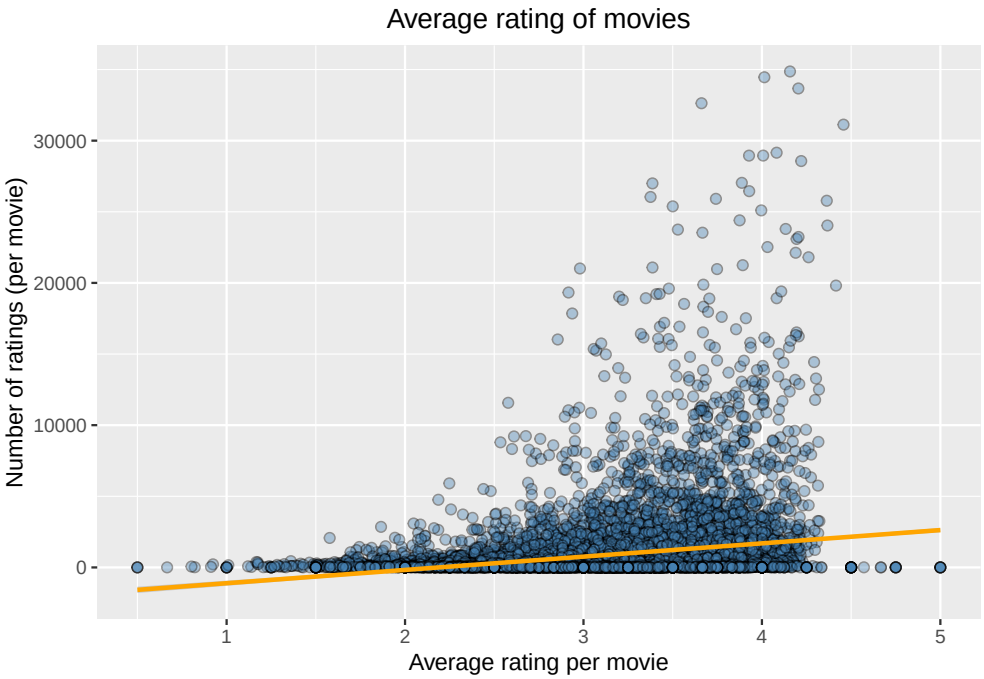
Now we know the number of ratings and mean rating for all movies at edx dataset. But, what are the ratings the users given to all movies?. It is important to know the size of the data sample, in order to have an unbiased view of it.

We can obtain the answer with the following figures:





Now we´re able to say, that there´s a clear fact: some movies are more often rated than others, as we can see by the distribution of the counts of ratings, reinforcing part of what we affirmed in the analysis of the ratings distribution. But furthermore, means that movies with a higher rating average, are also rated more often:



By analyzing the info provided, we conclude in relation with Movies & Ratings:

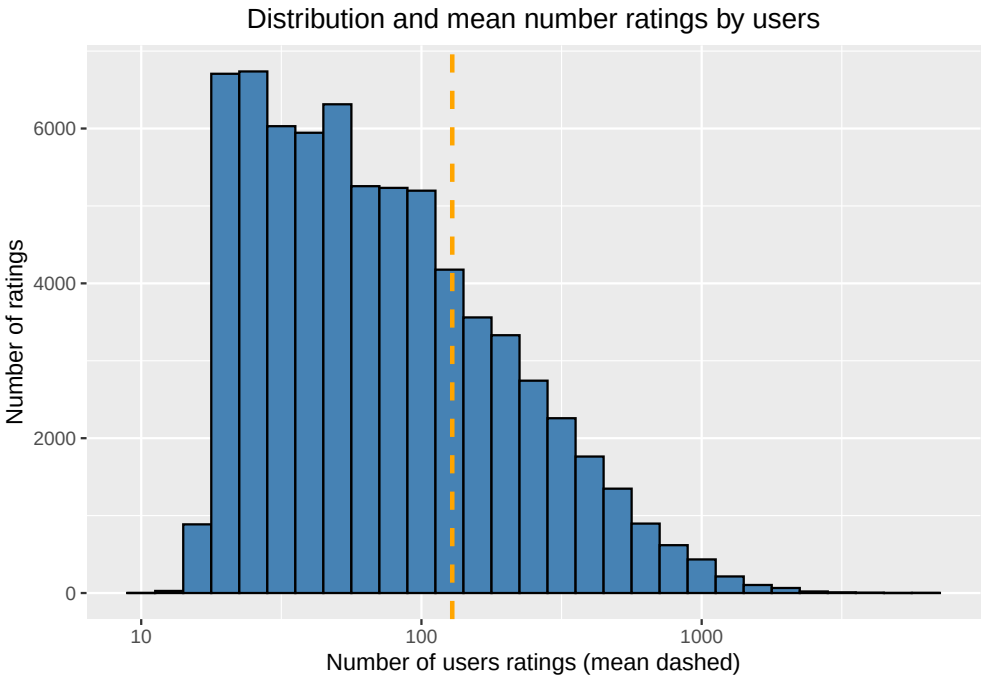
• There is a clear tendency to value positively. We believe that it is due to a replica effect: users tend to assess according to what others have previously done.
• This trend could be due substantially to the fact that MovieLens system offers higher rated movies more often, so users tend to rate them higher accordingly, something that may make sense if we take into account that users tend to look for well-rated movies to entertain themselves.
• Approximate 20% number of movies have a number of ratings higher than the average, which represents about 85% of ratings.

3.2.2 Users

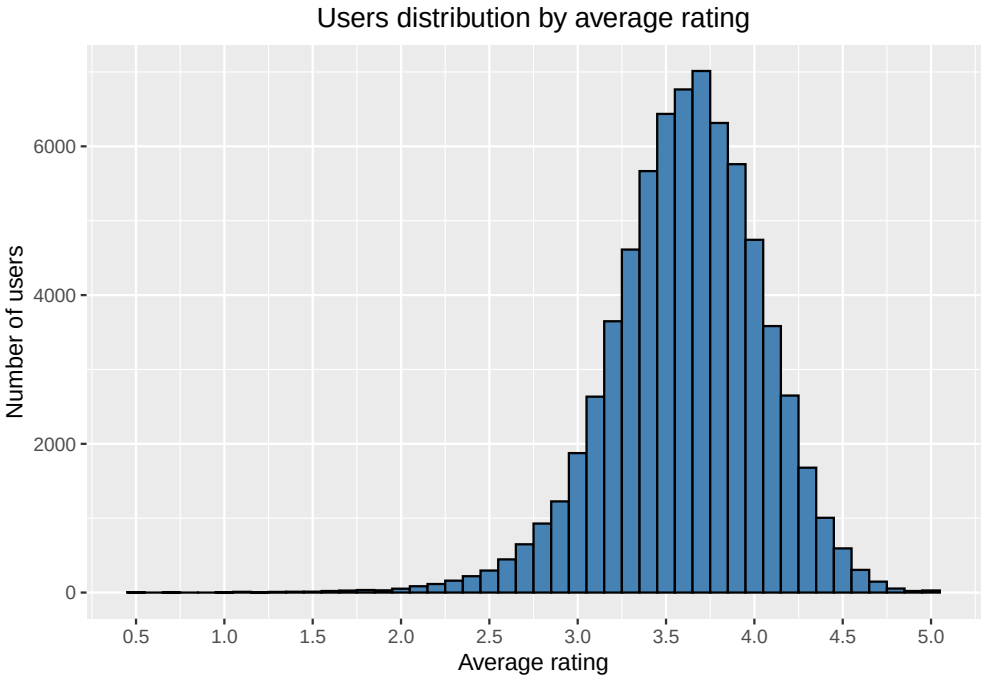
The total number of unique users in the edx dataset is 69.878 , while in validation dataset is 68.534 users represented by *userid*. By checking other data related with users in our datasets, we find out:
• in *edx dataset* the minimum number of ratings by a single user is 10; the maximum number is 6.616.
• in the *validation dataset* the minimum number of ratings by a single user is 1, while the maximum number is 743.

But the mean rating each user provides to a film, is identical in both data sets: 3.61

The following histogram represents the distribution and mean of number ratings by user:



We need to know how users rate (low, high…). We need to know the distribution by average rating:



Two things stand out as a result of what we can see in the figures:

• the fact that the ratings of the users to the films can be considered high (their average).
• There are user groups that rate thousands of movies and there are those that rate very few movies. These groups can bias our analysis
• that the majority of users have given a positive or very positive evaluation of the films they have seen. And there´s a few group of them in the opposite side.

From our point of view, it does not seem that the evaluations made by the users have been correctly carried out (with criteria) or at least substantiated. We do not know if it is part of the process of generating recommendations, thus biasing the information that we are analyzing, or if the users really have a criterion that is sufficiently based on an adequate critical spirit. In any case, we must take both possibilities into account when building prediction models.

3.2.3 Genres

The different genres we can find on both data sets are *Action*, *Adventure*, *Animation*, *Children*, *Comedy*, *Crime*, *Documentary*, *Drama*, *Fantasy*, *Film-Noir*, *Horror*, *IMAX*, *Musical*, *Mystery*, *Romance*, *Sci-Fi*, *Thriller*, *War*, *Western*

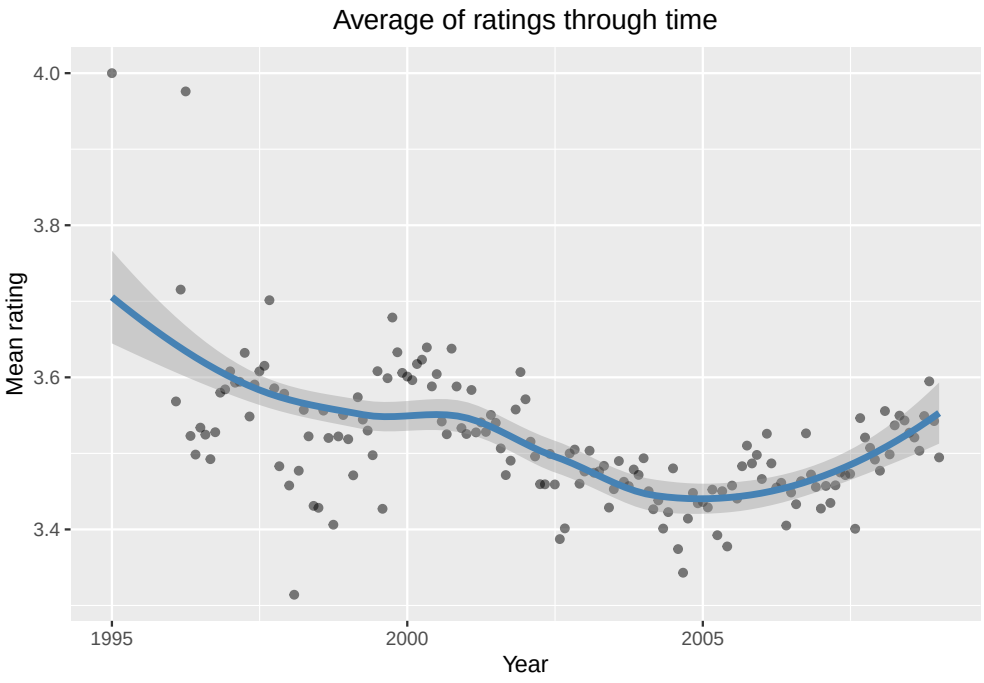
However, there's an exception on edx dataset as we can find also entries named as "*(no genres listed)*".

First, we are going to determine the preferences (via scoring) that users have about the genres of movies they watch:

Looking at the information provided by the previous figure, we can see that users prefer *Noir-type* movies while at the other end of their preferences are *Horror* movies. That means that average rating fluctuates significantly.

3.2.4 Time

The last parameter to finish the data exploration, corresponds to the way the time may affect our predictions. The following figure represent the average ratings of movies by year:



We may conclude that time, although has some effect on ratings, it is not significant on ratings.

4. Modeling

The evaluation metric utilized in the data modelling of this research is the Root Mean Square Error. As mentioned at the begining of this document, our this project was to achieve a RMSE lower than 0.86490.

We have defined the following models:

• Base Line (Naive). • Movie effect.
• User effect.
• Regularized Movie and user effect model.
• Matrix Factorization (based on the residuals of the best model).

4.1 Model performance

The way we are going to compare the models, has Root Mean Squared Error (RMSE) as our loss function, as is the standard deviation of the residuals (prediction errors) that are a measure of how far from the regression line data points are. In other words, tells us the average distance between the predicted values from the model and the actual values in the dataset: the lower the RMSE is, the better the model fits.

In the formula shown below, $y_{u,i}$ is defined as the actual rating provided by user i for movie u, $\hat{y}_{u,i}$ is the predicted rating for the same, and N is the total number of user/movie combinations.

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

4.2 Model criteria

Taking into account the information analyzed throughout this document in relation to the predictor variables, both Genres and Time are not significant as contributors of relevant information. We have splitted into training and testing datasets at 80%:20% ratios respectively. A total of five models have been created in an incremental fashion.

To first build a baseline prediction model, we will consider movie and user effect. The simplest algorithm for predicting ratings is to apply the same rating to all movies. Here, the actual rating for movie i by user u, $Y_{u,i}$, is the sum of this "true" rating, μ, plus $\epsilon_{u,i}$, the independent errors sampled for the same distribution.

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

• Movie effects

We know that movies are rated differently, some higher than others. As there is a high number of movies and ratings, we believe that this point will improve the accuracy of the prediction model: any further improvement to our model, may be to take into account the effect of the movies on the rating b_i.

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

The least squares estimate of the movie effects, $\widehat{b}_i$, can be derived from the average of $Y_{u,i} - \hat{\mu}$ for each movie i and, thus, the following formula was used to take account of movie effects:

$$\hat{b}_i = mean\left(\hat{y}_{u,i} - \hat{\mu}\right)$$

• Movie and user effects

Of course, some users are more active than others at rating movies, so further refinements have been made to the algorithm in order to adjust for user effects (b_u).

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

In this case too, rather than fitting linear regression models, the least square estimates of the user effect, $\hat{b}_u$ has been calculated using:

$$\hat{b}_u = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i\right)$$

4.3 Regularization

Regularization is a form of regression technique that shrinks or regularizes or constraints the coefficient estimates towards 0 (or zero). In this technique, a penalty is added to the various parameters of the model in order to reduce the freedom of the given model. In our case, will allow us to penalize large estimates that come from small sample sizes.

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

where the first term $\frac{1}{N}\sum_{u,i}\left(y_{u,i} - \mu - b_i - b_u\right)^2$, strives to find b_u's and b_i's that fit the given ratings. The regularizing term, $\lambda\left(\sum_i b_i^2 + \sum_u b_u^2\right)$, avoids over fitting by penalizing the magnitudes of the parameters. This least square problem can be solved via the matrix factorization.

We used cross validation to pick the best λ and using calculus we can show that the values of b_i and b_u that minimize this equation are :

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

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

4.4 Matrix factorization

Matrix Factorization is a collaborative filtering solution for recommendations. The fundamental assumption behind collaborative filtering technique is that similar user preferences over the items, could be exploited to recommend those items to a user who has not seen or used it before. In simpler terms, we assume that users who agreed in the past (e.g. purchased the same product or viewed the same movie) will agree in the future.

We will apply Matrix Factorization with parallel stochastic gradient descent. In order get the results desired, we have used *Recosystem package* that is typically used to approximate an incomplete matrix using the product of two matrix in a latent space.

5. Results

5.1 Model 1: BaseLine

Model calculates the average rating. The RMSE related to this model is the one that we will use as a reference to improve successively.

The average rating is mu = 3.51, and the RMSE is 1.06.

Model	RMSE
Just the Average	**1.059904**

5.2 Model 2: Movie Effects

It is obvious that the characteristics of the movies affect the ratings provided by users. That induces us to apply b_i for each movie our model. The average rating for a movie is sure to have a difference from the overall average rating for all movies.

The estimate of movie effect (b_i) changes significantly over all of the movies included in the train dataset. We can see adding the movie effect into the algorithm improves the accuracy of the model by 11.00%, yielding an RMSE of 0.94, but unfortunately its over our target.

Model	RMSE
Just the Average	**1.059904**
Movie Effect Model	**0.943700**

5.3 Model 3: Movies and Users effects

Due the fact users can provide very low ratings for all the movies they have seen, distorting what other users have objectively rated, we will try to reduce that noise by adding the user bias b_u to the movie effect model previously to calculate the RMSE for this model.

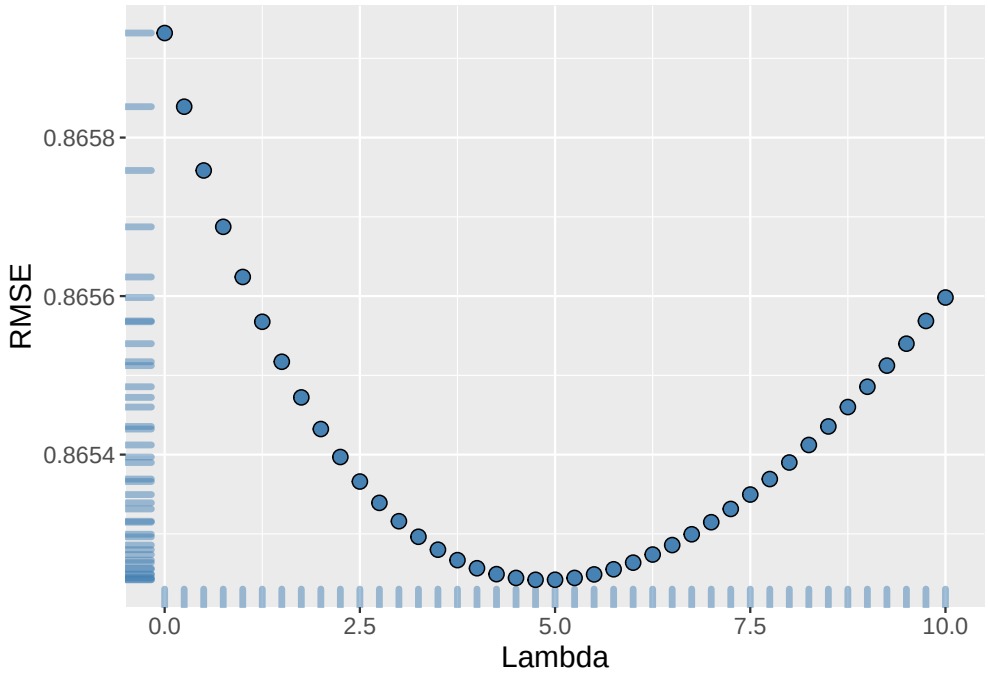
Adjusting for both movie and user effects has allowed us to improve RMSE by 18.00% versus the Baseline model.

Model	RMSE
Just the Average	**1.059904**
Movie Effect Model	**0.943700**
Movie and User Effect model	**0.865900**

5.4 Model 4: Movie and user regularization

In order to avoid overfitted or under fitted, we use regularization to reduce the chance of overfitting and help us get an optimal model. In this case, we will take into account on the movie and user effects, by adding a larger penalty to estimates from smaller samples via the parameter λ.

The reason is that as we´ve seen before, there are users who have valued a multitude of movies, while others have hardly intervened; In the same way, there are films with many ratings, while others have hardly been rated. This supposes the distortion of the sample data.



In the previous figure we can see the RMSE that has been assigned to each of the lambda values that we have tested. The process tells us that the optimal value for λ was 4.75 since it reduced the RMSE to 0.87, which is an improvement of 18.00% in the precision of the baseline and enough to exceed the RMSE that we had set as the project objective.

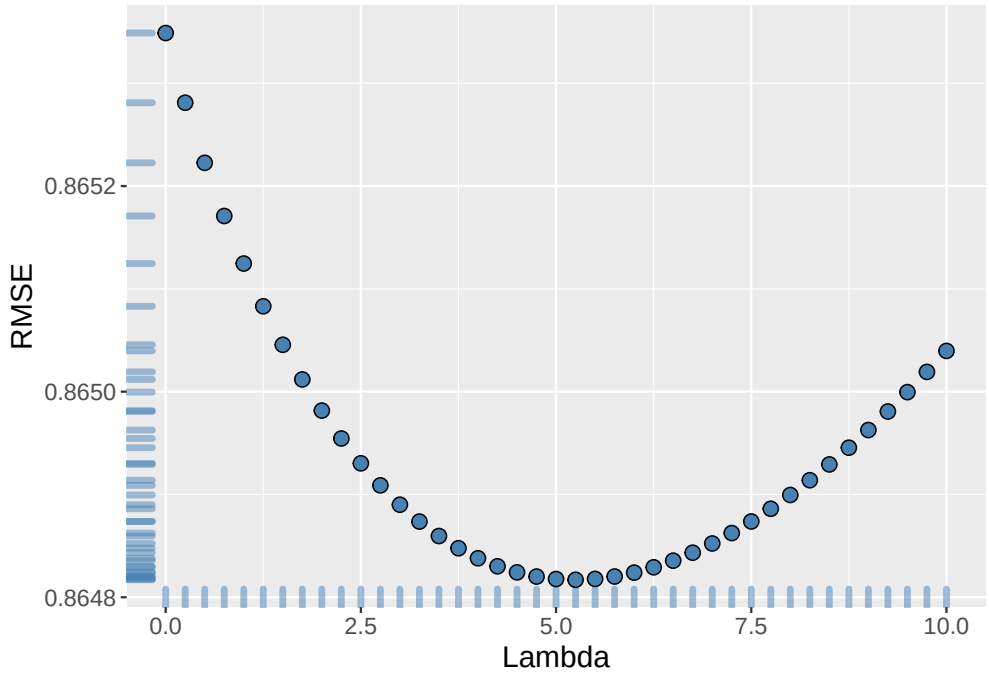
Model	RMSE
Just the Average	**1.059904**
Movie Effect Model	**0.943700**
Movie and User Effect model	**0.865900**
Regularized Movie and User Effect Model	**0.865200**

5.5 Model 5: Matrix Factorization

Applying the Matrix factorization method, we obtain an RMSE equal to 0.8 This represents a reduction of 25.00% relative to the baseline model.

Model	RMSE
Just the Average	**1.059904**
Movie Effect Model	**0.943700**
Movie and User Effect model	**0.865900**
Regularized Movie and User Effect Model	**0.865200**
Matrix Factorization	**0.796500**

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




```
## iter      tr_rmse          obj
##    0       0.8581   6.9613e+06
##    1       0.8335   6.4341e+06
##    2       0.8161   6.2626e+06
##    3       0.7987   6.0970e+06
##    4       0.7834   5.9534e+06
##    5       0.7709   5.8391e+06
##    6       0.7608   5.7497e+06
##    7       0.7524   5.6792e+06
##    8       0.7453   5.6208e+06
##    9       0.7392   5.5746e+06
##   10       0.7337   5.5298e+06
##   11       0.7290   5.4946e+06
##   12       0.7247   5.4610e+06
##   13       0.7209   5.4339e+06
##   14       0.7174   5.4081e+06
##   15       0.7142   5.3860e+06
##   16       0.7114   5.3655e+06
##   17       0.7087   5.3469e+06
##   18       0.7063   5.3296e+06
##   19       0.7041   5.3144e+06
```

```
## prediction output generated at C:\Users\ruben\AppData\Local\Temp\RtmpOAEeZL\file5428661cd20
```

5.6 Final Model

As we are able to see in the previous sections, we´ve been building the models to be evaluated through edx_test dataset. The best performing model for this subset is Matrix Factorization which produces an RMSE of 0.8.

Bearing this result in mind, we consider the final model must adopt the same methodology being the difference it has to be constructed over the entire data set, but expecting at least the same results, if not better ones.

After running the model, the final test on the validation dataset achieves an RMSE of 0.79, an improvement of 26.00% compared to the baseline model:

Model	RMSE
Best Model: Matrix Factorization	**0.7867**

6. Recapitulation

In this project, a recommender system was created for the 10 million Movielens dataset with a RMSE of 0.86396 when tested on a hold-out dataset, which is 10% of the original dataset.

Through five different models developed, we have reached a final model, Matrix Factorization, which yields an RMSE of 0.79 when trained on edx and tested on validation being an improvement over the first model of a 26.00%.