Edx Capstone: Movielens Project

Luis G. Vazquez de Lara Cisneros.

12/11/2020

Contents

1 Introduction/overview/executive summary

A recommendation system may be defined as a software tool providing suggestions to users for certain items (Ricci, Rokach, and Shapira 2011). More formally, is considered a decision making strategy for users under complex information environments (Isinkaye, Folajimi, and Ojokoh 2015). RS are primarily used in E-commerce websites, such as Netflix, YouTube, Spotify, Amazon and so on, but they are of use in social media platforms as well. Recommendaton systems (RS) help to solve the problem of information overload, providing Internet users with personalized content and services (Isinkaye, Folajimi, and Ojokoh 2015). RS can be broadly classified in collaborative filtering, content based filtering or hybrid filtering (Isinkaye, Folajimi, and Ojokoh 2015), with their own strengths and weaknesses each . In collaborative filtering algorithms, the general idea is that after analyzing many rating data by many users for many items, the rating of an unknown item by certain user can be predicted (Hahsler, n.d.). This approach has some limitations, such as cold start, scalability and sparsity. RS using content-based filtering analyze user's previous behavior and recommend items based on some features; one limitation is that require considerable data to build a reliable classifier (Reddy et al. 2018). Hybrid systems combine both collaborative and content systems, aiming to reduce the limitations present in both methods. The field of RS research was boosted after the Netflix challenge. In 2006 this company offered a prize of one million dollars to the data science community. The prize could be won for those improving the current recommendation algorithm by 10% (Lohr 2009).

The present work is part of the capstone course of the *Edx Data Science Professional Certificate*, where the student is challenged to build a recommendation system using the MovieLens 10M Dataset. This data set comprises 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. To achieve this goal, I will do the following:

- Divide the MovieLens 10M dataset in two sets: a set to perform the training of the algorithms (`edx` set) and a `validation` set to validate the final model (the code was provided by the course staff).
- The validation set will be used only once at the very end of the process.
- Select the general strategy and the features that will be included in the models, giving the rationale of these decisions.
- Divide the `edx` set in a training set and a test set, using 80% and 20% of the ratings respectively. As the names imply, the train set will be used to develop the models and the test set to evaluate their performance.
- The metric I will use to test the performance of the models will be the residual mean squared error (RMSE). This loss function was used in the Netflix challenge to decide on a winner.
- Evaluate the best model with the best RMSE using the `validation` set.

The RMSE will be calculated as follows:

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

2 Methods/Analysis

The number of algorithmic approaches used to address the Netflix challenge are numerous, a summary of these techniques were reviewed by Edwin Chen (Chen, n.d.) and are summarized in table 1.

Table 1: summary of the techniques employed by researchers in the Netflix Challenge.

Strategy	Explanation
Normalization of global effects	The overall rating given to a movie is decomposed in several baseline predictors, such as the movie effect, the user effect, the interaction between the user and the movie, etc.
Neighborhood models	Prediction relies on the similarity among items, or among users.
Implicit data	Similar to the neighborhood approach, but an offset weight is added depending on implicit information like preference for a certain type of movie.
Matrix factorization	Similar to neighborhood models, but with a more global view. It decomposes users and movies into a set of latent factors.
Regression	Standard regression models either with a user-centric or movie-centric approach.
Restricted Boltzmann Machines	Generally speaking, Restricted Bolstzmann Machines perform a binary version of factor analysis, more technically, is a stochastic neural network.
Temporal effects	Ratings can be influenced by temporal effects, such as the year of appearance, time from first rating, etc. Thus, user factors are modeled in a time-dependent manner.
Regularization	Regularization allows to penalize large estimates formed by small sample sizes.
Ensemble methods	Several methods are combined and provide a single rating that takes advantage of the strengths of each model. Gradient boosted decision trees and linear regression were used to combine algorithms.

R provides a number of packages with algorithms for recommendation systems. The `recommenderlab` package is mentioned in the textbook of the course. This package, developed by Michael Hahsler, is described by the author as a framework for developing and testing recommendation algorithms. The `recosystem` package uses parallel matrix factorization. The package `rrecsys` includes several algorithms, such as Global/Item/User-Average baselines and Weighted Slope One among others. Trying to use these packages for this assignment are well beyond the scope of the course. Moreover, the size of the database and the intensity of the computations are not supported by a commodity laptop. In view of the above, I will just work with the strategy we learned in the Machine Learning course. I will develop the following models:

- Model based on the normalization of the movie and the user effects.
- Model based on the normalization of the movie and the user effects, with regularization.
- Model based on the normalization of the movie and the user effects, adding the effect of genre.
- Model based on the normalization of the movie and the user effects, adding the effect of genre, with regularization.
- Model based on the normalization of the movie and the user effects, adding the effect of ratings per year since the movie came out.
- Model based on the normalization of the movie and the user effects, adding the effect of ratings per year since the movie came out, with regularization.

2.1 Movie and user effects

Figure 1, gives us insight on the variability in the number of times a movie gets rated and on the number of ratings the users provide. Some movies get rated more than others, and some users are more active than others. These observations give the rationale to use them as baseline predictors.

2.1.1 Model with normalization of the movie and the user effects

In view of the above, I developed a first model based on the normalization of the movie and the user effects:

$$Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i} \tag{2}$$

The estimates of b_i will be approximated with the average of:

$$Y_{u,i} - \hat{\mu} \tag{3}$$

for each movie i.

The estimates of b_u will be approximated with the average of:

$$Y_{u,i} - \hat{\mu} - \hat{b}_i \tag{4}$$

for each user u.

2.1.2 Normalization of the movie and the user effects with regularization

The next step is to create the model of equation (2), but with a regularization technique. The BellKor solution used L2 regularization, also called ridge regression (Koren, n.d.). The general idea is to add a penalization to avoid overfitting when there are very few items to estimate the coefficient. Instead of minimizing the least squares equation, we minimize an equation that adds a penalty:

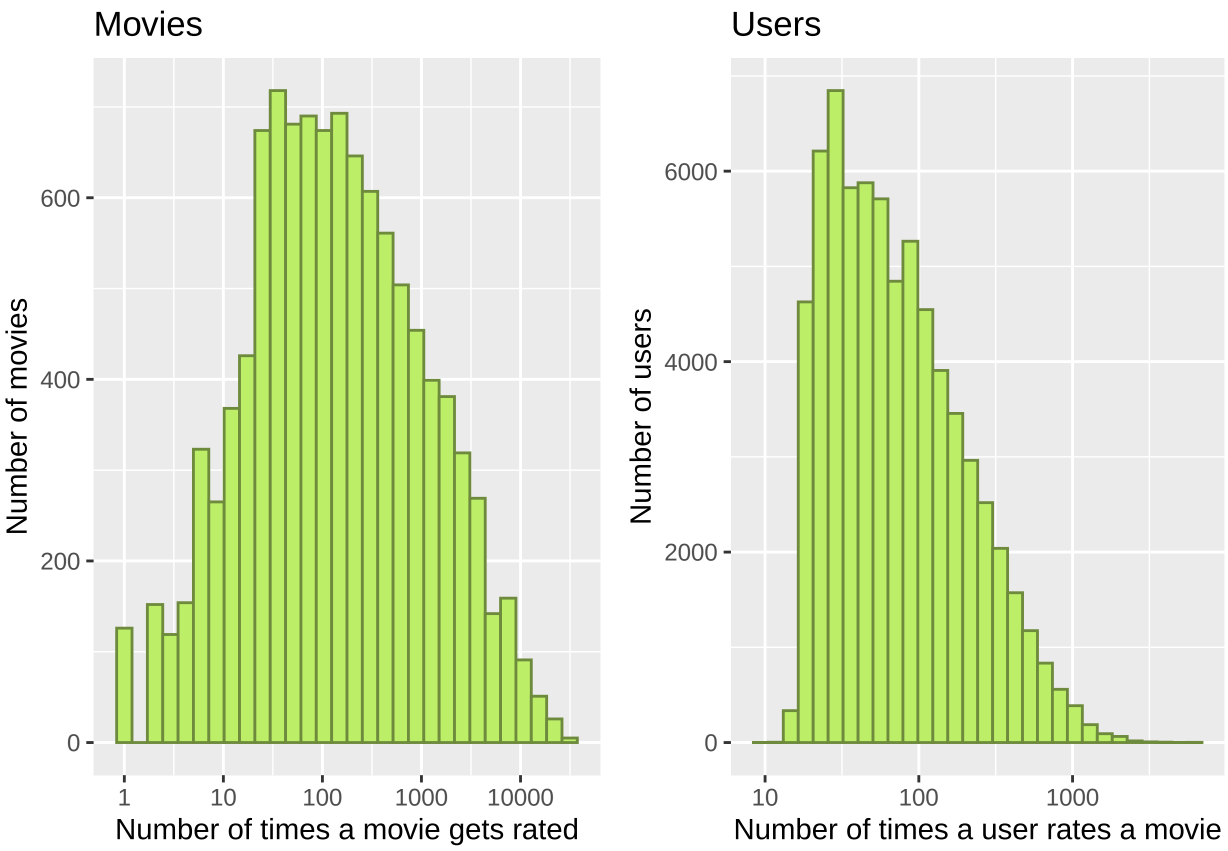

Figure 1: variability of movie and user ratings.

$$\sum_{u,i}(y_{u,i}-\mu-b_i-b_u)^2+\lambda\left(\sum_i b_i^2+\sum_u b_u^2\right) \tag{5}$$

Solving for $\hat{b}_i(\lambda)$ using calculus we get:

$$\hat{b}_i(\lambda)=\frac{1}{\lambda+n_i}\sum_{u=1}^{n_i}(Y_{u,i}-\hat{\mu}) \tag{6}$$

Solving for $\hat{b}_u(\lambda)$ using calculus we get:

$$\hat{b}_u(\lambda)=\frac{1}{\lambda+n_i}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}-\hat{b}_i\right) \tag{7}$$

By cross-validation, we can find the value of λ that minimizes the RMSE.

2.2 Genre effect

Genre is also an important variable. The `edx` database has a column with the genre of the movies. A movie has more than one genre in general. Figure 2 shows the relationship between the genres (grouped as in the database) and the mean rating.

The number of genre combinations are 797. For modeling purposes, I splitted the combinations into their individual genres. In table 2, I show the genres used. Seven movies had no genre listed, the genre *Film-Noir* had the highest average rating (4.01), the genre most frequently found was *Drama*, 3,910,127 films fell in this genre. On the contrary, the genre *Horror* had de lowest rating (3.27), the genre with less movies was *IMAX*, with only 8,181 pictures classified in this genre.

2.2.1 Normalization of the movie and the user effects, adding the effect of genre.

This approach falls in the category of *content-based filtering*. One of the mathematical approaches described in the literature is to create a rating matrix and calculate the Ecuclidian distance between current users and other users (Reddy et al. 2018). The package `recommenderlab` has implemented this algorithm, but due to the size of `edx` this cannot be done in a commodity laptop. Instead, we will normalize the genre effect, summing up the contribution of each genre, as follows:

$$Y_{u,i}=\mu+b_i+b_u+\sum_{k=1}^{K}x_{u,i}\beta_k+\varepsilon_{u,i} \tag{8}$$

with $x_{u,i}^k=1$ if $g_{u,i}$ is genre k, and K the number of genres of *ith* movie.

I will estimate β_k, the contribution of the genre to the rating as the average of:

$$Y_{u,i}-\hat{\mu}-\hat{b}_i-\hat{b}_u$$

for each movie genre k.

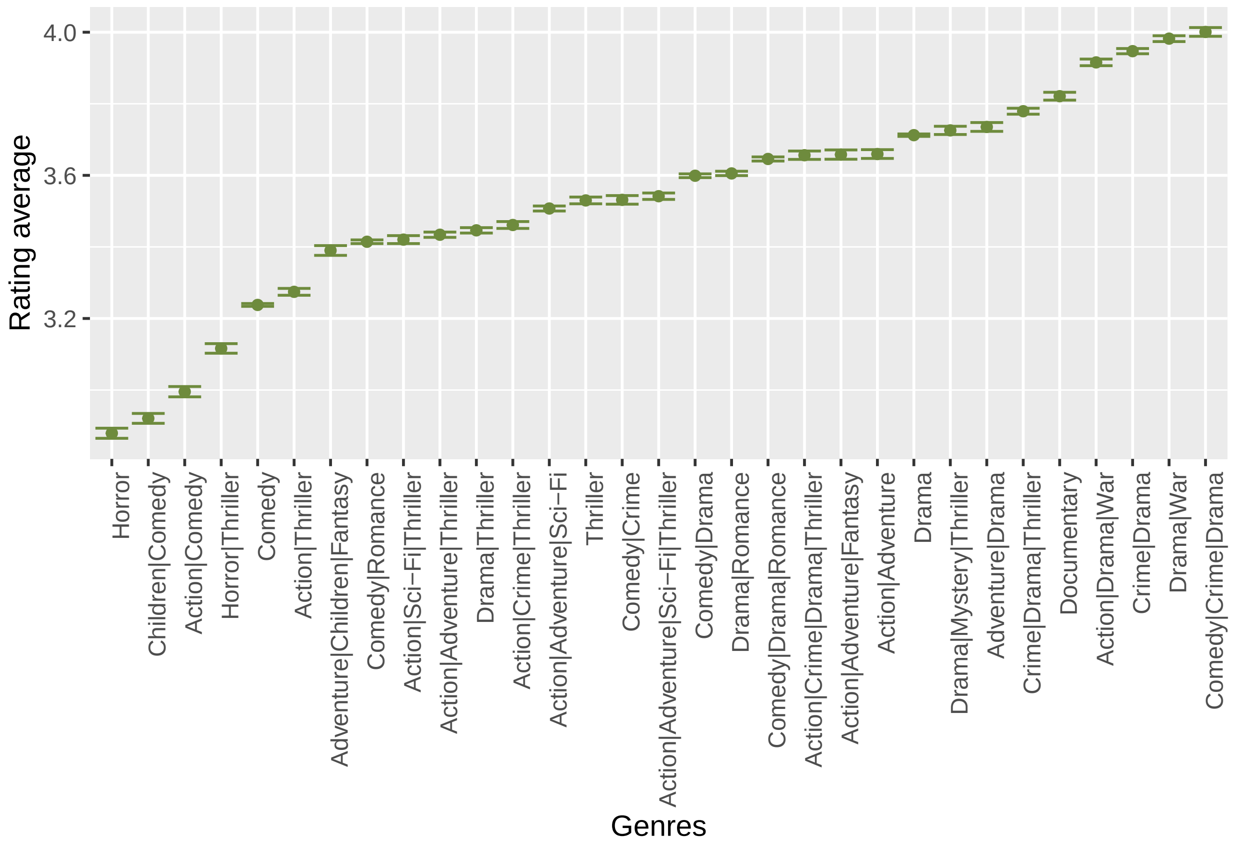

Figure 2: effect of genre on rating. Only movies with more than 50000 ratings are shown. Bars represent three standard errors.

Table 2: genres used in the edx dataset.

	Average rating	Number of movies
(no genres listed)	3.64	7
Action	3.42	2,560,545
Adventure	3.49	1,908,892
Animation	3.60	467,168
Children	3.42	737,994
Comedy	3.44	3,540,930
Crime	3.67	1,327,715
Documentary	3.78	93,066
Drama	3.67	3,910,127
Fantasy	3.50	925,637
Film-Noir	4.01	118,541
Horror	3.27	691,485
IMAX	3.77	8,181
Musical	3.56	433,080
Mystery	3.68	568,332
Romance	3.55	1,712,100
Sci-Fi	3.40	1,341,183
Thriller	3.51	2,325,899
War	3.78	511,147
Western	3.56	189,394

2.2.2 Normalization of the movie and the user effects, adding the effect of genre, with regularization.

I will use the the model of equation (8), but the estimation of $\hat{b}_k$ will be done with the equation (9). The values of $\hat{b}_u$ and $\hat{b}_i$ will be estimated using the equations (6) and (7) respectively.

$$\hat{b}_k(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} \left(Y_{u,i} - \hat{\mu} - \hat{b}_u - \hat{b}_i \right) \tag{9}$$

For each genre k. Due to computational limitations, the value of λ calculated in the regularized model with the movie and user effects will be used.

2.3 Ratings per year effect.

Most rated movies tend to have better ratings. To give evidence of this, I stratified the movies by ratings per year and calculate their average ratings. Figure 3 depicts the trend of this quantity and the average rating. We can see that as a movie gets more rated, the average rating increases.

To create the predictive model using this variable, I will transform the ratings per year to a categorical variable, creating intervals. Figure 4 shows that the trend is mantained.

2.3.1 Model based on the normalization of the movie and the user effect, adding the effect of ratings per year.

The exploration data analysis provides graphic evidence that there is a relationship between the ratings per year and the rating. I will use a very naive approach: I will transform the ratio into a categorical variable, and the model will be:

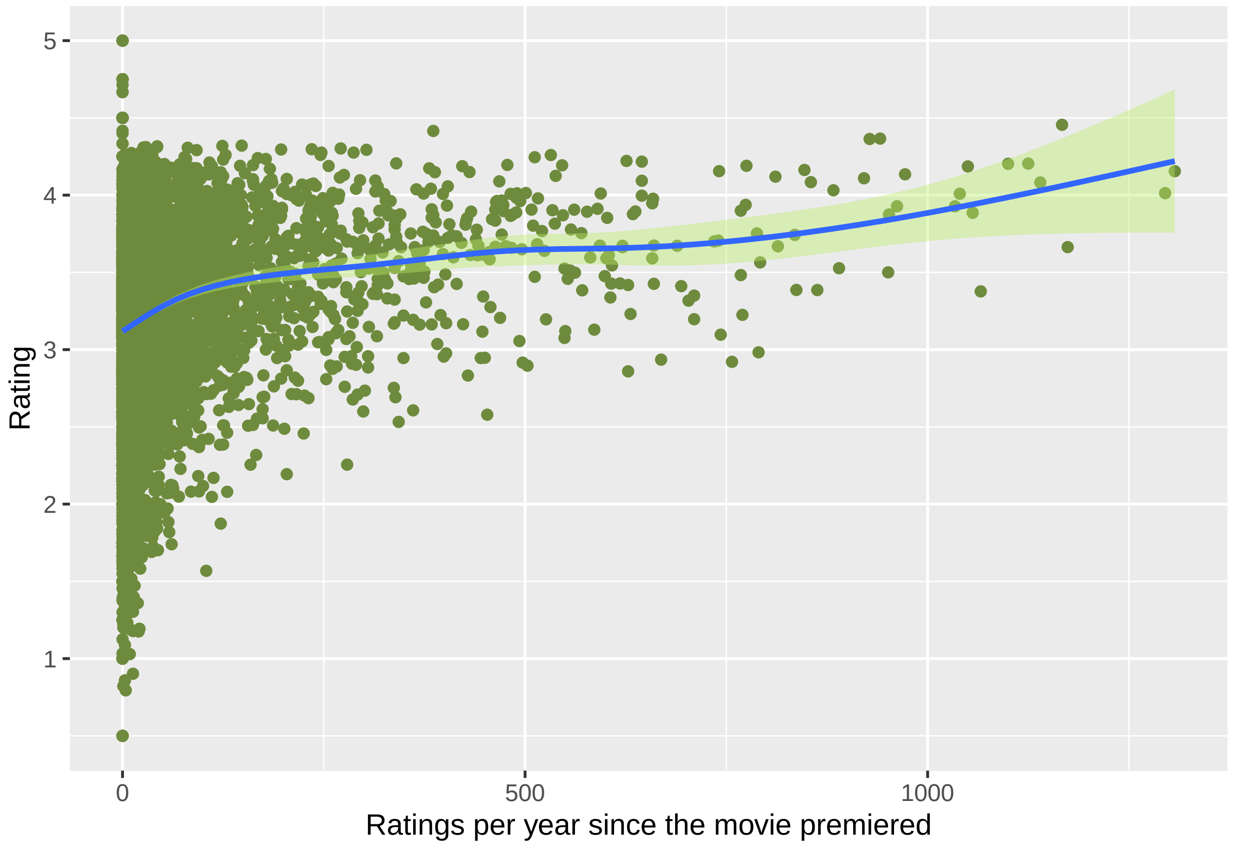

Figure 3: Relationship between ratings per year and rating.

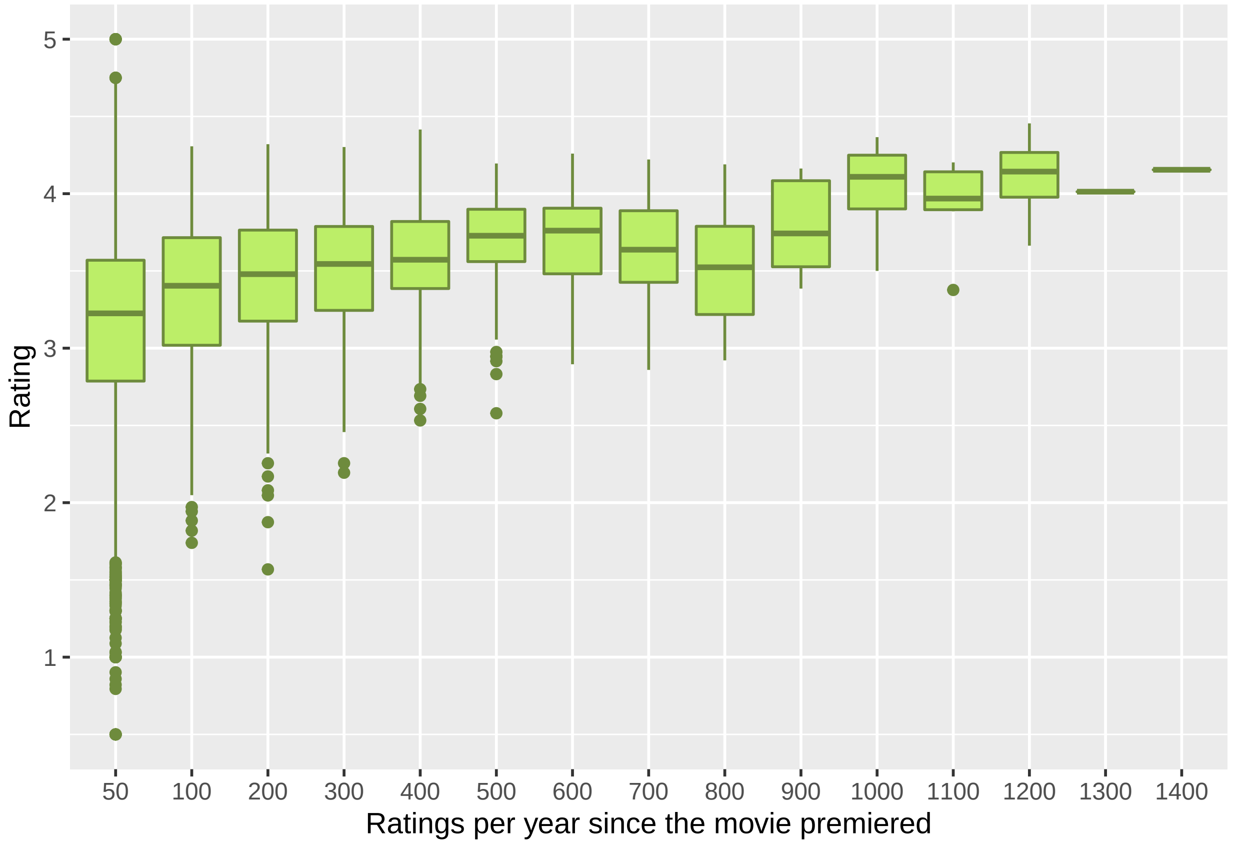

Figure 4: Relationship between ratings per year and rating, creating intervals on the number of ratings per year since de movie premiered.

$$Y_{u,i} = \mu + b_i + b_u + b_{yr} + \varepsilon_{u,i} \tag{10}$$

Where b_{yr} will be calculated as the average of:

$$Y_{u,i} - \mu - b_i - b_u$$

for each yr_i, where:

$$yr_i = n/(2018 - year)$$

year is the year the movie premiered.

2.3.2 Model based on the normalization of the movie and the user effect, adding the effect of the ratio: ratings per year/number of years since the movie came out, with regularization.

The same model as in equation (10), but b_{yr} will be estimated as follows:

$$\hat{b}_{yr}(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} \left(Y_{u,i} - \hat{\mu} - \hat{b}_u - \hat{b}_i \right) \tag{11}$$

for each yr_i. The values of $\hat{b}_u$ and $\hat{b}_i$ will be estimated using the equations (6) and (7) respectively. The best value of λ which minimizes RMSE will be calculated by cross-validation.

3 Results

The code used to create the models is not shown in this document, but it can be seen in the corresponding .rmd file from the github repository. I splitted the `edx` dataset in a `train_set` and a `test_set` using the `caret` package. The training set comprised 80% of the `edx` database (7,200,043 ratings), thus the test set was formed by the remaining 20% (1,799,964 ratings).

The RMSE of the model which only took into account the user and the movie effects [see equation (2)] was 0.8664993. For the model with regularization, the best λ for $\hat{b}_i$ and $\hat{b}_u$ calculated with the equations (6) and (7) was found with cross-validation. Figure 5 shows that the value of λ which minimizes the RMSE is 4.81. The regularization improved the RMSE to 0.8658014.

The model which included the genre [see equation (8)] reported a RMSE of 0.8664361 when tested against the test set. When the regularization was used, the RMSE did not change (0.866436).

The next model I tested included the ratings per year, using the equation (10). The RMSE obtained with this approach was 0.8662466. For the model with regularization, the coefficients for the ratings-per-year effect were calculated as in equation (11), $\hat{b}_i$ and $\hat{b}_u$ were calculated with the equations (6) and (7) respectively. The best λ was found using cross-validation. Figure 6 shows the relation between several values of λ and the RMSE obtained. The value of λ which renders the best RMSE is 4.6. Surprisingly, the RMSE with the regularized model which included the ratings per year dropped down to 0.6024742.

Table 3 summarises the results obtained with the models tested, ordered according to the RMSE obtained. According to this loss function, the models with regularization tended to perform better. The RMSE improvement obtained in the model which added the ratings-per-year effect with regularization must be seen with caution, as the Netflix challenge considered worthy improvements of 10%. Furthermore, considering that the mathematical approach is right, the variable *ratings-per-year* is of little use, as new movies have less chance to be rated.

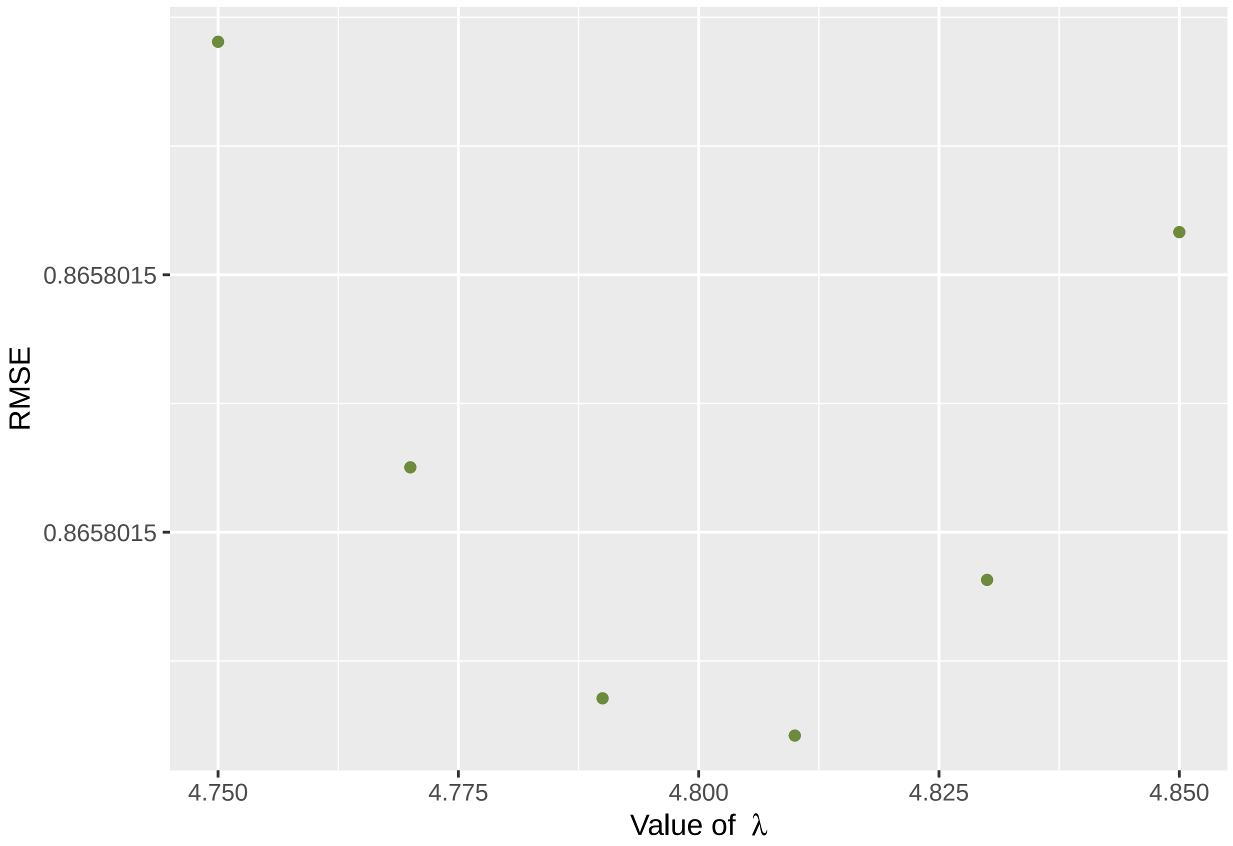

Figure 5: effect of differents values of λ on the RMSE of the model with the user and movie effects.

Table 3: Summary of the performance of the different algorithms tested.

MODELS	RMSE
Movie and user effects, adding the ratings-per-year effect with regularization	0.6024742
Normalization of movie and user effects, with regularization	0.8658014
Movie and user effects, adding the ratings-per-year effect	0.8662466
Movie and user effects, adding the effect of genre with regularization	0.8664360
Movie and user effects, adding the effect of genre	0.8664361
Normalization of movie and user effects	0.8664993

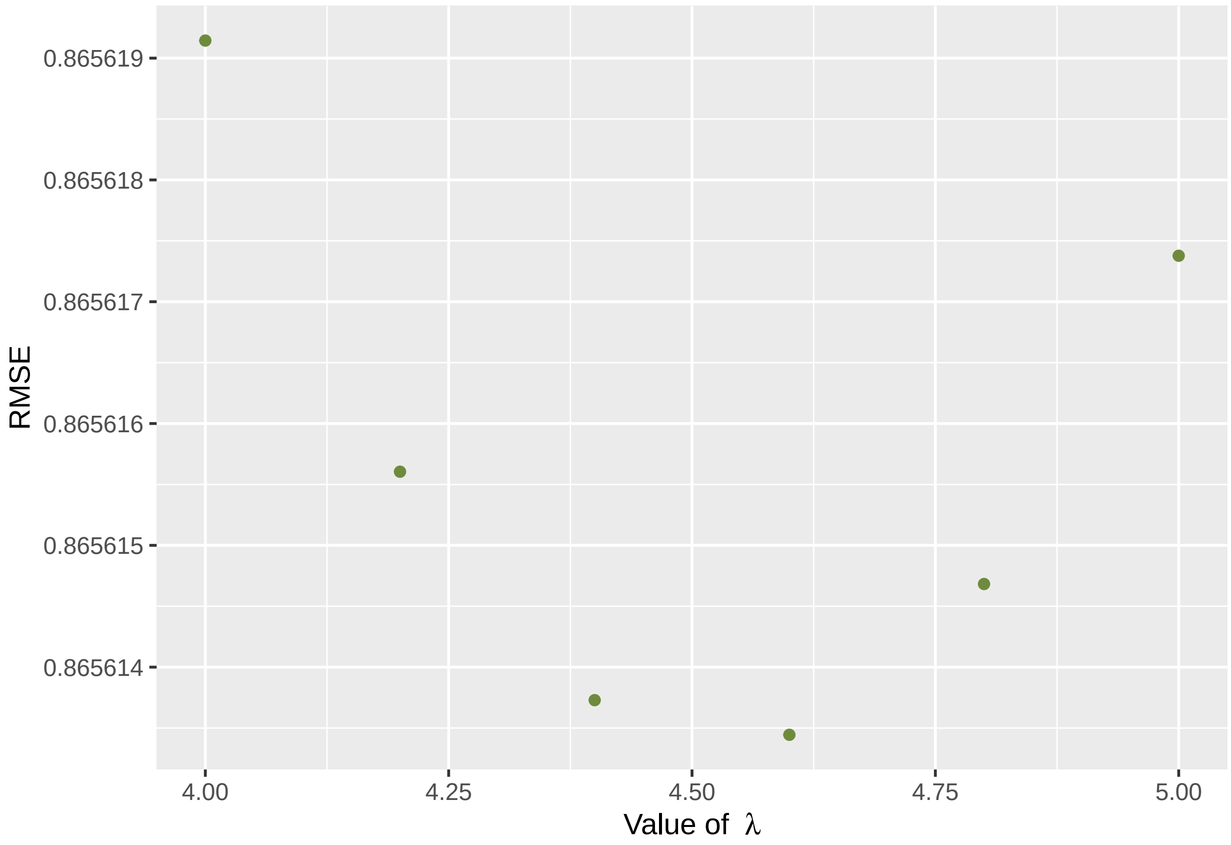

Figure 6: effect of differents values of λ on the RMSE of the model with the ratings-per-year effect.

3.1 Performance of the best model with the validation set

The model that considered the user, movie and the ratings-per-year effects with regularization was tested in the `validation` set. Table 4 shows the value of RMSE. It is not very different from the one obtained when the test set was used[1].

Table 4: Performance of the model with the best RMSE using the validation set

MODEL	RMSE
Movie and user effects, adding the ratings-per-year effect with regularization	0.6043639

4 Conclusion

The developement of recommendation systems is a promising area of research. Better mathematical models will impact other areas beyond the entertainment industry, as they can be applied across the fields of science, commerce and politics. In this work, the models employed are rather modest, but only are intended to demonstrate the skills learned in the *Edx Data Science Certificate* courses. As mentioned in the Introduction section, some R packages are available, but the size of the `edx` database severely limits their use in a commodity laptop. With these limitations in mind, we can conclude that modeling a few baseline predictors such as those used in this work, can capture the main effects in the data. Future work may include datasets with behavioral and demographic data, such as location, age, gender and previously chosen movies, among others.

References

Chen, Edwin. n.d. "Winning the Netflix Prize: A Summary." http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/.

Hahsler, Michael. n.d. "Recommenderlab: A Framework for Developing and Testing Recommendation Algorithms," 40.

Isinkaye, F. O., Y. O. Folajimi, and B. A. Ojokoh. 2015. "Recommendation Systems: Principles, Methods and Evaluation." *Egyptian Informatics Journal* 16 (3): 261–73. https://doi.org/https://doi.org/10.1016/j.eij.2015.06.005.

Koren, Yehuda. n.d. "The BellKor Solution to the Netflix Grand Prize," 10.

Lohr, Steve. 2009. "Netflix Awards $1 Million Prize and Starts a New Contest." https://bits.blogs.nytimes.com/2009/09/21/netflix-awards-1-million-prize-and-starts-a-new-contest/.

Reddy, Srs, Sravani Nalluri, Subramanyam Kunisetti, S Ashok, and Venkatesh Bachu. 2018. "Content-Based Movie Recommendation System Using Genre Correlation." In.

Ricci, Francesco, Lior Rokach, and Bracha Shapira. 2011. "Introduction to Recommender Systems Handbook." In, edited by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, 1–35. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-85820-3_1.

[1] For some reason, some movies in the training set did not appeared in the validation set, thus `NAs` were omitted to calculate the RMSE.