Movie Recommendation System

Jose Garcia-Garcia

21/8/2020

Introduction

Over the last decade, we have witnessed a considerable growth in the number of VOD (Video On Demand) streaming platforms and in the number of subscribers consuming those platforms.

In fact, if we check the growth of Netflix (the platfrom with more paid users around the world) we can observe that number of subscribers has growth from 25 millions in 2012 up to 167 millions at the end of 2019.

Central to its success are the recommendations algorithms, that helped to steer users towards the content that they would most enjoy. So, as recommendation algorithms are one of the AI and machine learning that are changing the world over the last few years, we have decided to implement one as main purpose of this project.

In order to achive our goal and train our algorithm, we are going to utilize one of the movie ratings datasets available for public use in the web. In concrete, we are going to manipulate the MovieLens 10M dataset. This dataset is provided by GroupLens, a research lab at University of Minnesota, specialized, amongst other things, in recommender systems.

The dataset contains 10 million ratings applied to 10681 movies by 71567 users of the online movie recommender service MovieLens. The data we are going to manipulate is contained in 2 different files:

- movies.dat
  - each line represents one movie with following format: *MovieID::Title::Genres*
  - MovieID is the real MovieLens id
  - Title includes year of release, e.g., *Braveheart (1995)*
  - Genres are a pipe-separated list, e.g, *Action|Drama|War*
- ratings.dat
  - each line represents one rating of one movie by one user, and has the following format: *UserID::MovieID::Rating::Timestamp*
  - UserID represents each individual user
  - Ratings are made on a 5-star scale, with half-star increments

Note: There is one additional file in the data set (tags.dat) which contains metadata applied to one movie by one user. However, as we are not going to use it for the purpose of our project, we are going to ignore it

The recommendation system model will be based on studying the different effects over the rating for the different features presented in the data set. Following that principle, the main steps to be executed will be:

- load the raw data set from MovieLens and transform it into a manipulable R data frame
- analyze and quantify the different effects that the features (users, movies, genres, etc. . .) have over the final rating
- implement the model based in those findings
- calculate the final accuracy of the model and present future work that could be done in order to improve that accuracy

Analysis and model design

As presented in the introduction, our recommendation system will be based on quantifying the different effects that different features have over the final rating given by a user to a movie.

That means that we need to develop an algorithm that fulfills:

$$Y_{u,i} = \mu + b_1 + b_2 + \cdots + b_n + \epsilon_{u,i}$$

where $Y_{u,i}$ represents the rating for a user and movie, μ represents the average of all ratings, each b_i term represents one different effect to be taken into account and $\epsilon_{u,i}$ is the error in our prediction.

To compare different effects introduced into our model and calculate the final model accuracy, we will use root mean squared error (RMSE) as our loss function:

$$\sqrt{\frac{1}{N}\sum_{u,i}(\hat{Y_{u,i}} - Y_{u,i})^2} = \sqrt{\frac{1}{N}\sum_{u,i}(\epsilon_{u,i})^2}$$

where $\hat{Y_{u,i}}$ is our predicted rating.

The algorithm goal consists on reducing that error $\epsilon_{u,i}$ as much as possible, hence minimize the RMSE.

In order to achieve that goal, first step consists in transform the raw data contained in the 2 Movielens files into one final dataframe that will be the base for our analysis and model implementation.

Table 1: Movielens dataframe after transforming raw data

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	231	5	838983392	Dumb & Dumber (1994)	Comedy
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

Once we have transformed our raw data into a manipulable dataframe, we are going to split the ratings into 2 different datasets:

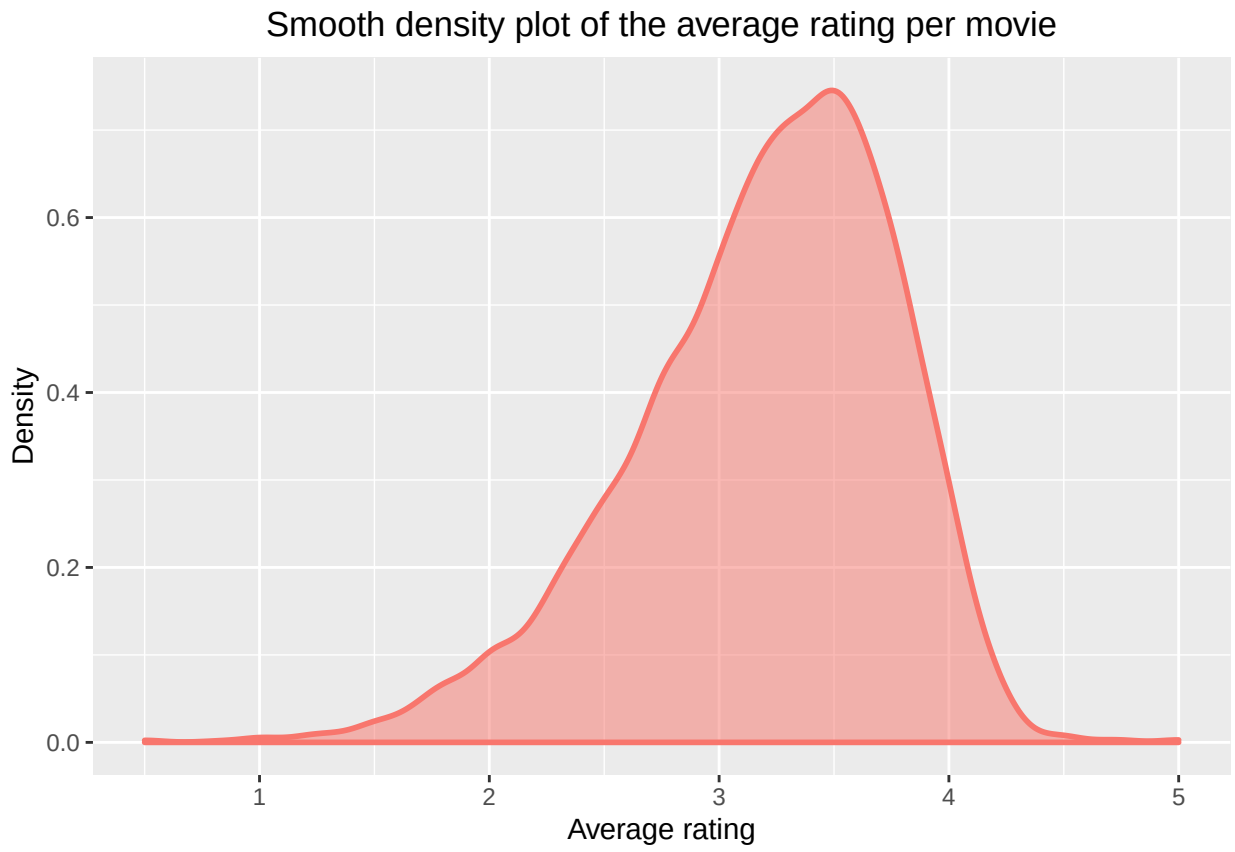
- *edx* working dataset (90% of MovieLens data), which will be used for doing the analysis and training our model
- *validation* dataset (10% of Movielens data), which will be used exclusively for the final accuracy validation of our model

Now that we have splitted the Movielens dataset into the working and validation datasets, we can start analyzing the different effects deduced from data examination and decide if it's worth it to add them into the model.

Movie effect

As the first effect we have decided to study, we have selected the movie itself as the most obvious one to start with. As we have a population of more than 10000 movies, we can predict that we are going to have very different average ratings amongst them. We have in that list Oscar winner movies (e.g. Titanic), or worldwide blockbusters (e.g. Terminator 2), which we would expect to have a much higher average rating than Razzie winner movies (e.g The Postman) or movies that were a total flop at the box-office.

If we summarize the ratings in our *edx* analysis dataset by movie, calculate the average rating for each movie and create a density plot of those average ratings, we can observe that there is a lot of variance in the avg ratings for the different movies.



We have proved that our hypothesis is correct and the average rating movie effect is quite important for our model, so we are going to include it in the model.

Before start implementing our model, we need to split our *edx* data set into 2 different datasets:

- *edx train* dataset (80%, around 7.2 millions of ratings), which will be used for training our model
- *edx test* dataset (20%, around 1.8 millions of ratings), which will be used for cross validation

First, we are going to calculate the mean of all movies μ as our first iteration of the model and use that as a baseline to compare different improvements of the model while adding different effects/terms.

If we only include this μ term into our model, we get an RMSE of *1.0612*, which is quite bad and can be easily improved just by adding the movie effect as observed in our analysis.

Now, we are going to define the movie effect as the mean of ratings given by all users minus the overall movie mean calculated. So, we can predict the rating as the mean of ratings given for all users that rated that movie (independently of individual user, movie genre or any other effect).

After adding this new term into our model, we get an RMSE of *0.9440*, which is an interesting improvement of *0.1173* compared to just calculating all movies average. However, this still can be improved in several ways.

Movie effect regularization

In our movie rating predictor case, we can have very high or low ratings for movies with a very low number of ratings.

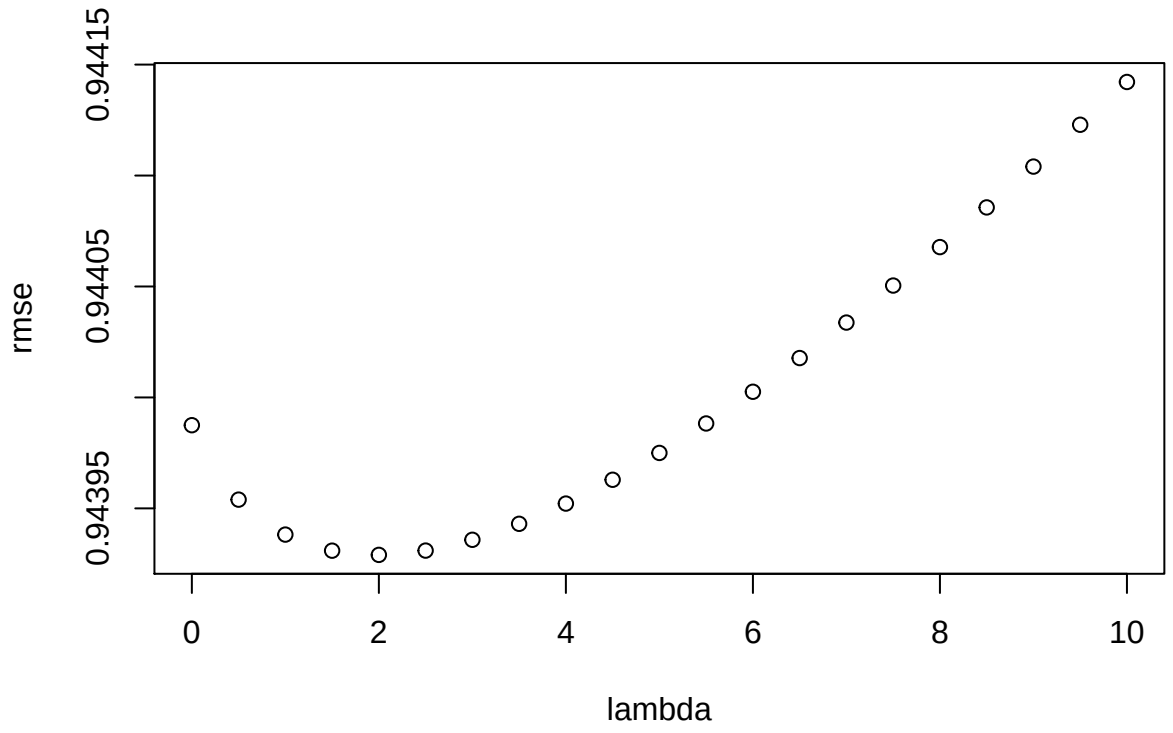
So, before start studying other possible effects to include in our model, we would like to make use of regularization for the movie effect.

Regularization adds a penalty term λ to the movie effect, so when the number of ratings for a movie is large, then λ is ignored:

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

where n_i is the number of ratings for movie i

As λ is a tunning parameter, we should run cross validation, checking which λ gives us a minor RMSE.



As per our cross-validation, the λ which produces the best RMSE for our prediction is 2. We include that λ into our model and we get an RMSE of *0.9439*

As we can observe, the improvement in the RMSE compared to non using regularization is quite small (around *0.001*). This could seem to be very strange as we expect a better improvement using this regularization but is not and have an easy explanation.

Regularization is quite useful when the number of ratings is low for a high percentage of the movies in our data set.

However, the percentage of movies with at least 5 ratings or more is *93.47%* and the total percentage of ratings in our data set related to those movies is *99.97%*. Hence, if we consider that a movie with 5 or more

ratings has a pretty accurate average and regularization doesn't make a big difference, regularization is only improving the prediction for a *0.03%* of the total population of our ratings sample.

In any case, even the improvement is quite low, we have decided to include this regularization into our model.

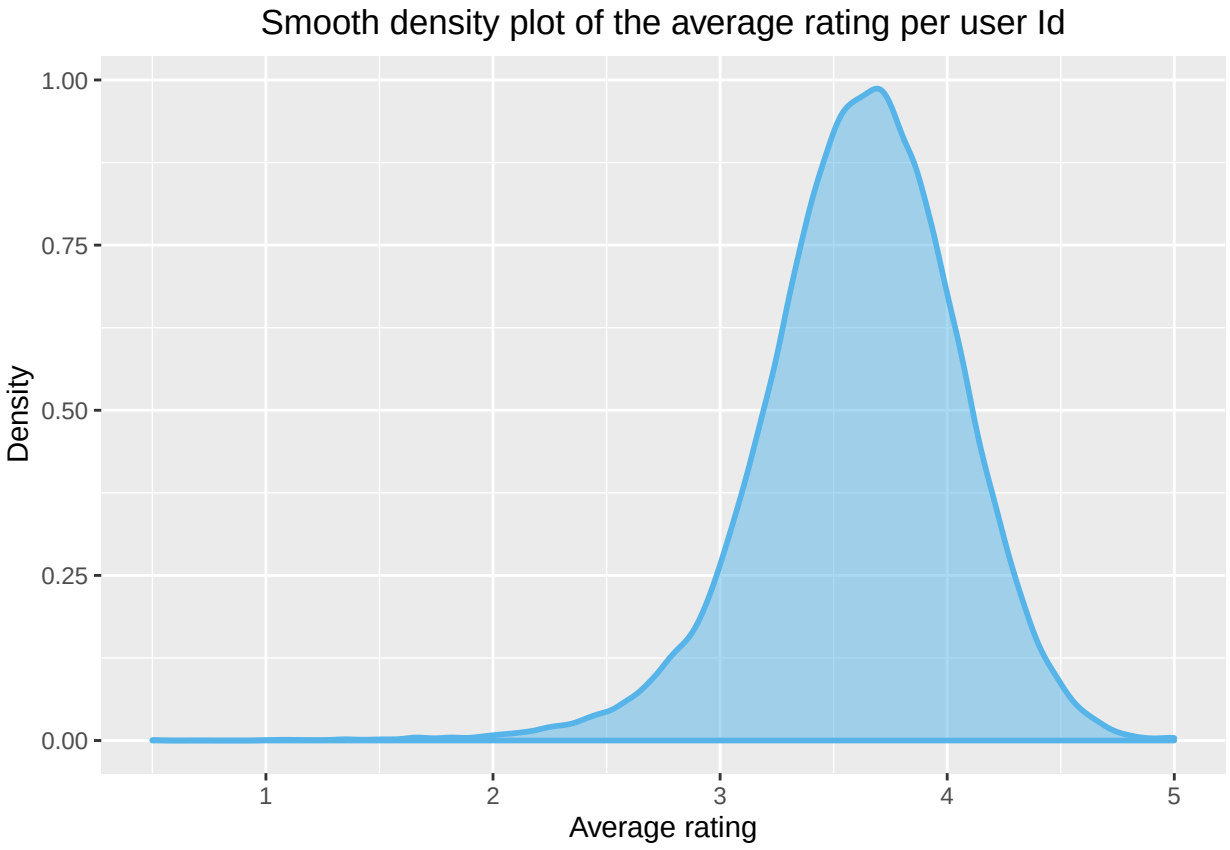
User effect

As this system is based on movie recomendations given by **users** is pretty clear than there is some kind user effect to take into account for our rating predictions.

When we talk about user effects, we are talking about effects like:

- each person is different, hence has a different measure scale for everything in life, and that includes movies
- each person likes different types of movies, i.e. some people prefer comedies, other prefer action movies

So, first of all, we need to check if our hypothesis is correct. If we summarize the ratings in our *edx* analysis dataset by user, calculate the average rating given by each user and create a density plot of those average ratings, we can observe that there is a lot of variance in the average ratings given by each user.



As a first step, we are going to introduce the average user effect into our model. This effect is not taken into account genre preferences, just the different scale of measure for each individual (there are people with tendency to grant higher ratings to the movies and others with the opposite tendency).

In this case, we are going to follow next steps:

- calculate the user effect for each one of the movies that given user has rated, using the training set, and as the result of the rating minus the mean for all movies and the movie effect regularized
- calculate the mean user effect as the average of those individual user effects per movie

- apply that to the model and predict our ratings using the test dataset and as result of adding the mean for all movies, plus the movie effect plus the average user effect

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

where b_u represents the user average effect for user u.

After adding this new term into our model, now the RMSE has decrease to *0.8665*. As expected, this user effect is quite important and we have been able to get an improvement of *0.0765* compared to the one obtained only using the movie effect with regularization.

However, as we mentioned before, each person likes different types/genres of movies: people that like comedies tend to grant a better rating to comedies, people that like action movies will do the same for action movies, etc.

So we think that this user effect can be improved if we expand it including the different genres of the rated movies by each user. As many movies have more than one genre, the final rating will include the average of the user effects for all the genres in the movie.

In order to do that, we are going to follow next steps:

1. Identify and split different genres for each movie, adding a new column for each different genre to each movie (value will be 1 if genre is present for that movie, 0 otherwise)

Table 2: Different genres for each movie (matrix of 1s and 0s added as columns to dataframe)

movieId	genres	Action	Adventure	Animation	Children	Comedy	Crime
122	Comedy|Romance	0	0	0	0	1	0
185	Action|Crime|Thriller	1	0	0	0	0	1
292	Action|Drama|Sci-Fi|Thriller	1	0	0	0	0	0
316	Action|Adventure|Sci-Fi	1	1	0	0	0	0
329	Action|Adventure|Drama|Sci-Fi	1	1	0	0	0	0
356	Comedy|Drama|Romance|War	0	0	0	0	1	0

2. Calculate the user effect average for each kind of genre:

i) Calculate the average for each genre that user has rated at least one film

ii) In case that there are genres for which user has not viewed/rated any movie, we apply the overall user effect calculated in previous step

Table 3: User effect per genre. Note: Only 8 rows shown for visualization purposes

userId	Action	Adventure	Animation	Children	Comedy	Crime	Documentary
1	1.6232954	1.5490116	1.4478378	1.4708830	1.5936133	1.8741882	1.5599683
2	-0.0930314	-0.0437468	-0.1356088	-0.7097269	-0.2209038	-0.1356088	-0.1356088
3	-0.0816022	-0.1466091	-0.0889455	-0.0889455	-0.0741386	-0.2528600	0.1662890
4	0.5545785	0.7282053	1.3237621	1.6312732	0.3817759	1.2540226	0.5682402
5	-1.3822021	-0.6592688	-2.9259999	-0.5469279	-0.0455948	0.6061207	0.1257224
6	0.2488597	0.0835768	-0.1932721	0.4129898	0.4806232	0.4706540	0.4129898

3. Calculate the prediction using our *edx test* dataset:

i) Calculate the user effect for each prediction as the mean of the user effect for the movie different genres

ii) Take into account that there is a small number of movies without any genre defined, so we are going to apply for them the user average from previous step

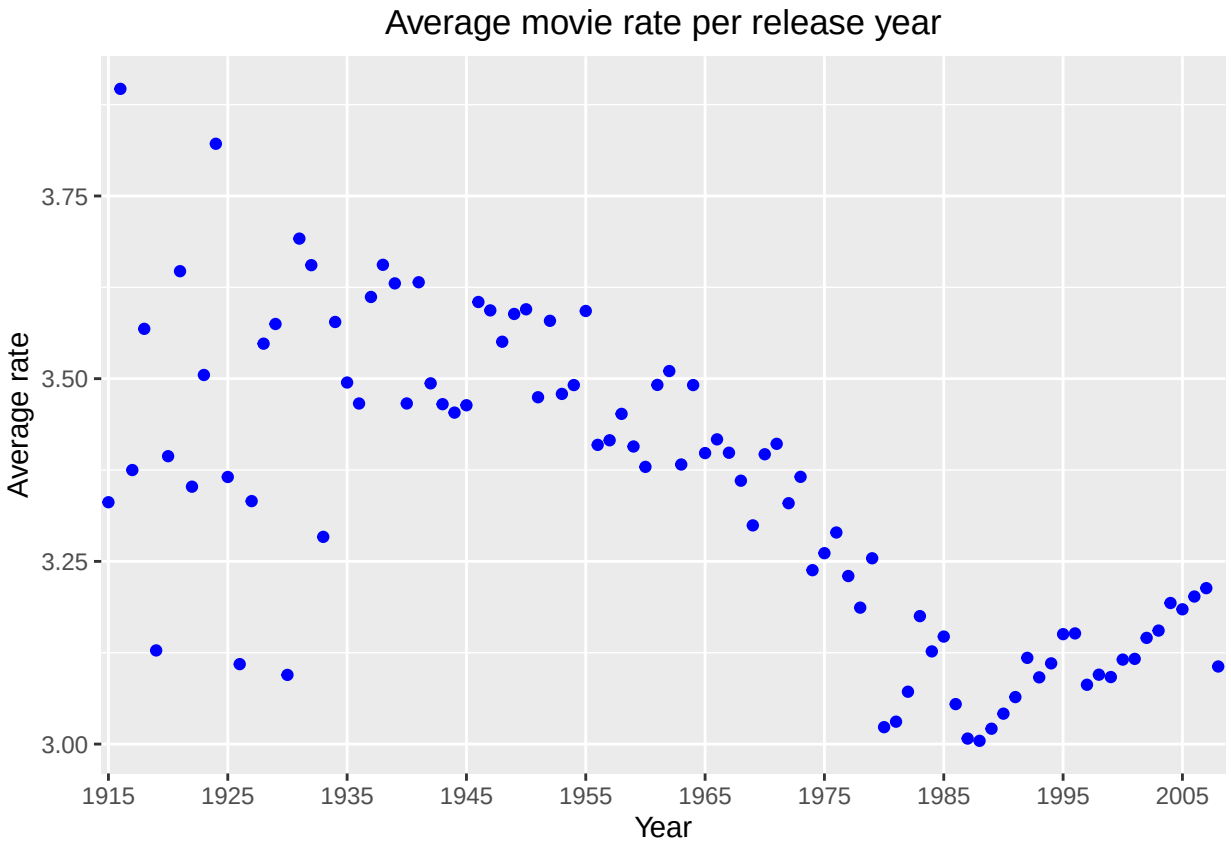
After adding this new term into our model, now the RMSE has decreased to *0.8527*. As expected, doing the breakdown of the user effect for each genre, has improved the accuracy of our model.

Movie release date

In order to improve our model, is possible to include additional terms. In this case, our hypothesis is that in modern times only classical movies that were very successful are watched and rated by users. Hence, we expect that ratings for classical movies are in general higher than for modern movies.

So, as a first step, we are going to find out if this theory is correct:

- extract release year from the title. E.g. *1995* from *Toy Story (1995)*
- plot the movie average per release year



In the graph we can observe that classical movies have a higher average rating (usually between 3.3-3.7 in years 1915 to 1970) than new ones, around (3-3.2 from 1985 to 2005), so we are going to apply this effect into our model.

After adding the release year effect into our model, the RMSE has slightly improved to *0.8523*.

Out of range values regularization

As a final improvement for our model, we are going to consider that predictions may have values higher or lower than maximum or lowest expected ratings, as a result of applying our model.

For example, if we observe the behaviour of user with Id equals to 1:

- user is rating every movie in the train set with 5 stars (maximum value allowed)
- as a result, the user effect for every genre and user 1 is quite high (e.g. 1.63 for Action, 1.55 for Adventure, etc...)
- when we apply this user effect for making a prediction for a movie with a low average rating, we get a predicted rating closer to 5 for that particular movie and user combination, which is probably pretty accurate
- however, when we apply it to a movie with a high average rating of 3.5 or more, the predicted rating will be for sure higher than 5, which is not expected as the max allowed value is 5

In order to regularize these outliers, we are going to look for the predicted values higher than 5 and regularize them to the maximum value of 5. Regarding the minimum expected value, after checking all the ratings in the train dataset, we observe that the mininum is 0.5, so we are going to do proceed in the same way.

There are 6411 predicted ratings higher than 5 and 492 lower than 0.5 after applying our model to the 1.8 million of rows in our edx test dataset. After doing this regularization process, the RMSE has improved a little bit until *0.8519*.

Results

As a result of analysing effects presented in previous section, an improvement in the RMSE after adding each one of these terms is observed:

Table 4: RMSE after adding different effects, calculated using the cross validation test dataset

Method	RMSE
Average of all movies	1.0611729
Average for each different movie	0.9439875
Average for each different movie with regularization	0.9439290
Regularized movie + user average effect	0.8664654
Regularized movie + user avg effect per genre	0.8526527
Regularized movie + user effect per genre + movie release year	0.8523260
Regul. movie + user per genre + release year + round max/min	0.8519521

As we can deduce, using average for all ratings (μ) as the base model, improvement in the RMSE is higher when we add effects like movie individual average (around *0.12*) or user effect per genre (additional *0.09*), than other effects like release year of the movie or using techniques like regularization.

In any case, as commented previosly, as long as they add improvements into our RMSE, we are adding them into the model in order to get the lowest possible error.

The resulting model will be represented in the following way:

$$Y_{u,i} = \mu + b_i(\lambda) + \sum_{g_i,u} b_u + b_{ry,i} + b_{reg} + \epsilon_{u,i}$$

where:

- $Y_{u,i}$ represents the rating for a user and movie
- μ represents the average of all ratings,
- $b_i(\lambda)$ term represents movie i average effect using regularization, being our calculated (λ) equals to 2
- $\sum_{g_i,u} b_u$ represents the summatory of the user u effects for each different genre g_i of the movie i

- $b_{ry,i}$ represents the release year effect for movie i
- b_{reg} represents the regularization term where the predicted rating, as result of applying the rest of the terms, is higher than 5 or lower than 0.5
- $\epsilon_{u,i}$ is the error in our prediction.

Once that we have decided the form of our final model, we are going to calculate the final RMSE using the validation dataset, which results to be **0.8521**.

Conclusion

We have been able to provide a model for prediction of ratings given by user for any movie. As shown in previous sections, our model takes into account:

- different effects that are involved in the given rating: movie popularity and acclamation, user personal preferences and movie release date
- different techniques like regularization or keeping the predictions between the expected range doing some normalization tasks

As a result of calculating the performance of our model using the root mean squared error (RMSE), we have proven that the accuracy is pretty good, being the validation RMSE equal to **0.8521**. This result is better than the maximum expected one, *0.86490*, that was asked in the project requirements.

However, as part of a possible future work, we can study if this performance would be improved using additiona data or different prediction algorithms. Some of these lines of investigation could be:

- studying effects that could be deduced from manipulating other data already present in the Movielens dataset. Examples:
  - studying the effect that time of the day (or day of the week) when the score was given by user could have over that rating. I.e., mood of the user could change depending on hour of the day, day of the week, etc...
  - studying other data presented in the *tags* file ignored for this study. I.e., users that rate higher scores to movies with specific metadata are likely to grant higher scores as well to movies with same related metadata given by other users
- collecting additional data useful for deducing additional effects. Examples:
  - collecting list of main actors/actresses for the movies. It's quite probable that users that like some specific actors/actresses, will grant a higher rating to movies starring them
  - classifying films that are part of popular franchises like Star Wars, Marvel, Harry Potter, etc... It's probable that users that like movies from a specific franchise will like the rest of them
- studying performance of different prediction algorithms instead of using the model we have defined. I.e., this problem is a good candidate for applying a nearest neighbour algorithm (knn):
  - expected output classes will be the half point ratings between 0 and 5 (0.5,1,1.5,...,5)
  - we will predict the one with higher probability after applying the model
  - we will combine different input features (userId, movieId, genres, etc...) and use different parameters (i.e. number of neighbours to include) in order to find out the combination that produces the lowest RMSE