Movielens Project Report

Alvaro Ortiz

12/18/2020

Contents

Executive Summary

This report describes a Movie Recommendation System for the HarvardX Data Science Program Capstone Project. The models use the MovieLens dataset prepared in advanced in the course.

The data prepared uses the 10M version of MovieLens dataset, collected by GroupLens Research and provided by the course. We divide the MovieLens dataset in a Training (edx) and Validation Test (validation) subsamples. The `edx` dataset contains approximately 9 Millions of rows with 70.000 different users and 11.000 movies with rating score between 0.5 and 5. There is no missing values (0 or NA).

The Recommendation systems use information from ratings to products by users to make specific recommendations for users.This Big Data Set is finally used to predict the rating or satisfaction of an individual for a particular product and then recommend (or not) the specific product to the user.

Once the data is uploaded and cleaned further, we proceed with the exploratory data analysis. This will be useful to check the potential machine learning algorithms to be used in the a Recommendation System. The Rating recommendation system for Movies used in this project use the stars which users gave to the movies. The starts rank from one to five, with one star suggesting a bad movie, whereas five stars suggests it is an excellent movie.

After the first exploratory analysis of the data, we will train a machine learning algorithm using the data of the variables included in the training set. Later, we will evaluate the alternative models in the validation set.

The goal of the recommendation system isto reduce the RMSE below the one generated by a naive model. To obtain the maximum score we will have to reduce the RMSE below the threshold of 0.86490.

In the Results section we will present the alternative recommendation system models tested. The results show that all of them reduce the Root Mean Square Error (RMSE), our loss function. Besides, one of the system reduce the RMSE below the threshold established by the course of 0.8649.

The best recommendation system presented in the project is the **Regularized Movie+User & Genre Movie Model** which one we obtain RMSE of **0.8626**, clearly below the threshold to achieve maximum grades of 0.8649.

Analysis of the Data and Methodology

The data

I will be using the 10 Millions dataset obtained from the MovieLens dataset included in the dslabs package and provided by MovieLens. The database is just a subset of a larger dataset with millions of ratings to make to make the computation a little easier.

The 10 Millions dataset is divided into two dataset: `edx` for training purpose and `validation` for the validation phase. The RMSE of the models will be evaluated in the validation sample.

The `edx` dataset contains approximately 9 Millions of rows with 70.000 different users and 11.000 movies with rating score between 0.5 and 5. There is no missing values (0 or NA).The MovieLens dataset is automatically downloaded from the following links

- [MovieLens 10M dataset] https://grouplens.org/datasets/movielens/10m/
- [MovieLens 10M dataset - zip file] http://files.grouplens.org/datasets/movielens/ml-10m.zip

The Movielens Dataset is divided 2 subsets with the createDataPartition command and we will use a partition parameter p= 0.1, the training sample of "edx" (90% of the total sample) to train the algorithm while the "validation" subset (10% of the total sample) will be used to test the accuracy of our recommendation systems. Therefore, the Algorithm will be developed or trained in the "edx" (training sample) while the "validation" subset will be used to test the final algorithm.

The variables included in the "edx" database are the following:

- userID: Movielens users were selected at random for inclusion. Their ids have been anonymized.
- movieID: The identification code of the movie
- rating: Ratings are made on a 5-star scale, with half-star increments.
- timestamp: represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

- title: Movie titles, by policy, should be entered identically to those found in IMDB, including year of release. However, they are entered manually, so errors and inconsistencies may exist.
- genres: Genres are a pipe-separated list, and are selected from 18 genres. A movie can have more than a gender.

The data can be processed and cleaned further. First, we can transform the timestamp variable in a readable format. Second, we can divide the movies in the Title and in the year of release of the movie. Last, we can extract the different Genres of the movies which will be using later as a variable of the model. A sample of the transformed edx variables can check in table 1, the new variables are the following.

- We will obtain timestamp in a readable format and extract the year (**YearOfRate**) and month of rate (**MonthOfRate**)in both dataset
- We will obtain the movies tittle
- We will extract the year of release (**release**) from the movies title.

- We will extract the genres and we will add a new category to cope with remaining NAs (no genres listed)

The analysis included in the code shows that there ara not missing values in the sample.

Table 1: A Sample of the edx File

userId	movieId	rating	title	genres	release	yearOfRate	monthOfRate
1	122	5	Boomerang	Comedy|Romance	1992	1996	8
1	185	5	Net, The	Action|Crime|Thriller	1995	1996	8
1	292	5	Outbreak	Action|Drama|Sci-Fi|Thriller	1995	1996	8
1	316	5	Stargate	Action|Adventure|Sci-Fi	1994	1996	8
1	329	5	Star Trek: Generations	Action|Adventure|Drama|Sci-Fi	1994	1996	8
1	355	5	Flintstones, The	Children|Comedy|Fantasy	1994	1996	8

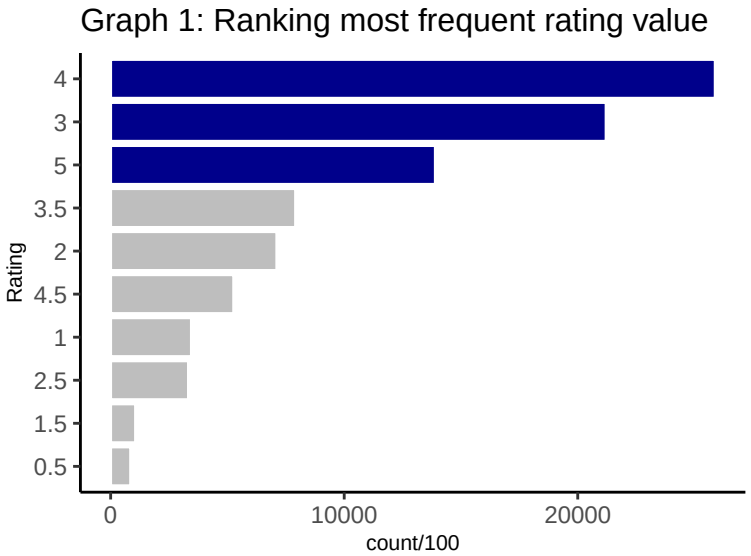
The Variables

The basis ofthe recommendation simple is to predict the rating for movie i by the user u. Thus in principle, all the other other ratings related to movie i and by user u may be used as predictors. However, different users rate different movies and a different number of movies. Furthermore, we may be able to use information from other movies that we have determined are similar to movie i or from users determined to be similar to user u.

In this sense it is important to check properties of the data to better understand how can we use them in the recommendtaion system challenges.

The Dependent Variable: Rating by User

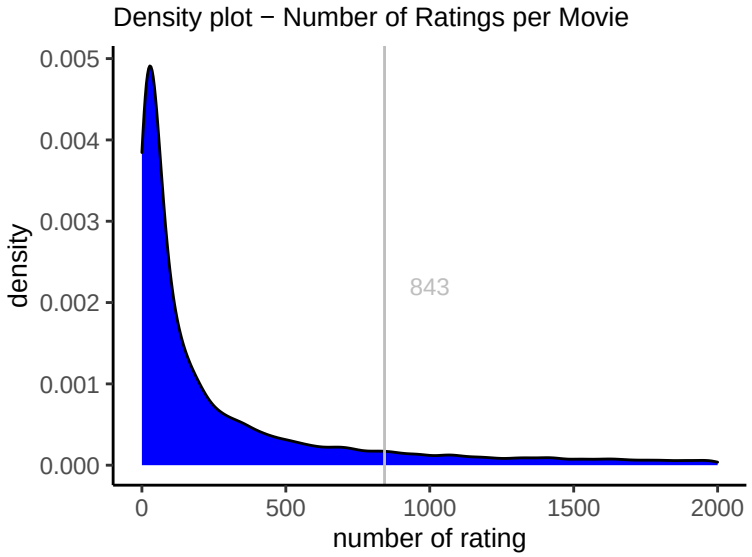
The dependent variable of the recommendation system is the rating provided by the different users. The Ratings are graded ina 5-star scale, with half-star increments. The exploratory analysis show that Users have a preference to rate movies higher rather than lower as shown by the distribution of ratings in the graph below. The data in the graph are ordered and shows that the most common rating is “4”, followed by “3” and “5”. The graph also shows that, Iin general, half rating are less common than whole star ratings. The less frequent rating is “0.5”.



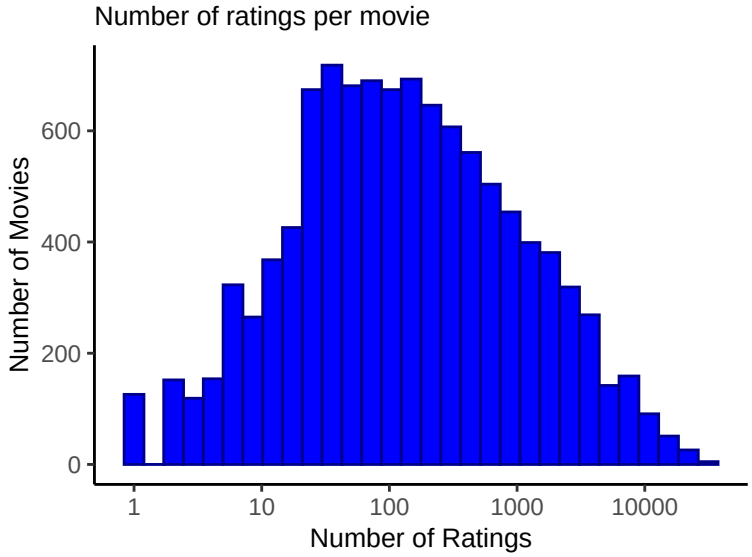
The second graph shows that some movies have been rated more often that other, while some have very few ratings and some of rthem only one rating. This will be important for our model as very low rating numbers might results in untrustworthy estimate for our predictions. The density plot of the number of the ratings

per movie shows that the average rating per movie is 843 but the distribution is biased to lower ratings, thus lower than 843 ratings are much more likely.

```
## [1] 843
```



This is more obvious from the following graph showing the distribution of the ratings per movie. The graph shows that the bulk of the ratings per movie is concentrated between 10 and 1000. We can also observe that there are 125 movies that are rated only once. This open the door to use regularization techniques in the system.

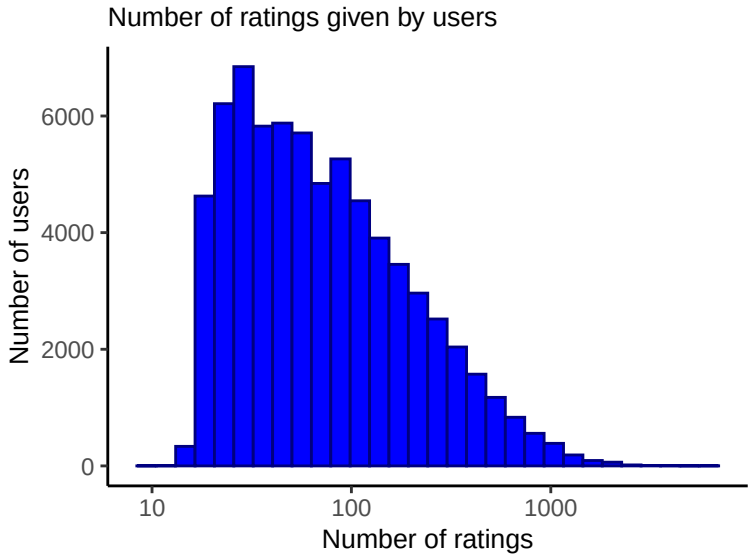


The reason to regularize is becoming obvious from the following table. As we can see in the table, 20 movies that were rated only once appear to be noisy, which can difficult the predictions of the ratings.

Table 2: 20 Movies rated only one

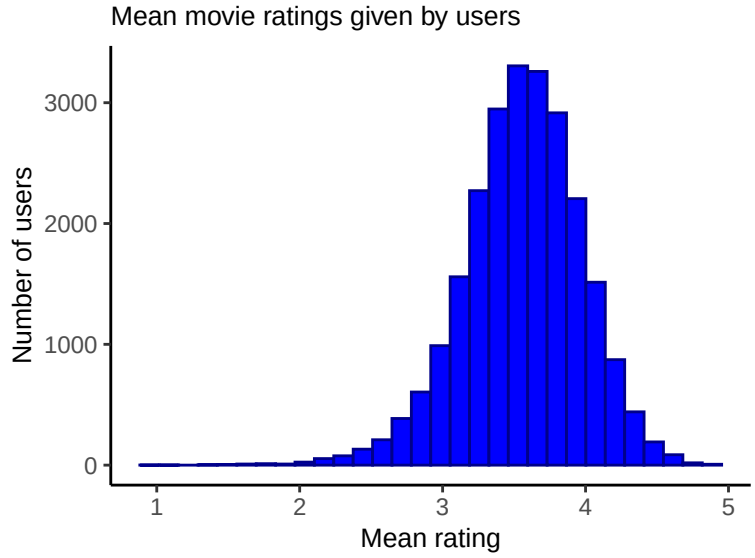
title	rating	n_rating
1, 2, 3, Sun (Un, deuz, trois, soleil)	2.0	1
100 Feet	2.0	1
4	2.5	1
Accused (Anklaget)	0.5	1
Ace of Hearts	2.0	1
Ace of Hearts, The	3.5	1
Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di...)	1.5	1
Africa addio	3.0	1
Aleksandra	3.0	1
Bad Blood (Mauvais sang)	4.5	1
Battle of Russia, The (Why We Fight, 5)	3.5	1
Bellissima	4.0	1
Big Fella	3.0	1
Black Tights (1-2-3-4 ou Les Collants noirs)	3.0	1
Blind Shaft (Mang jing)	2.5	1
Blue Light, The (Das Blaue Licht)	5.0	1
Borderline	3.0	1
Brothers of the Head	2.5	1
Chapayev	1.5	1
Cold Sweat (De la part des copains)	2.5	1

The exam of the number of ratings provided by every user provide also some insights. As we can observe, the majority of users have rated between 30 and 100 movies. However, there are users which have provided a low number of ratings which can affect also the efficency of the recommendation system.



The User Effect

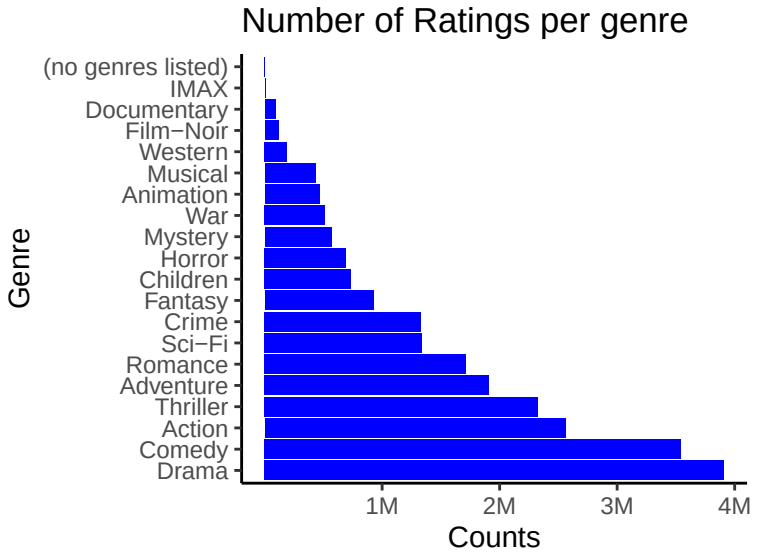
Different users can differ in how they rate the movies. While some users tend to give much lower star ratings other users tend to give higher star ratings than average. The visualization below includes only users that have rated at least 100 movies. As observed, the average rating is between 3.5 and 3.8 which shows that on average users rate movies with a positive bias.

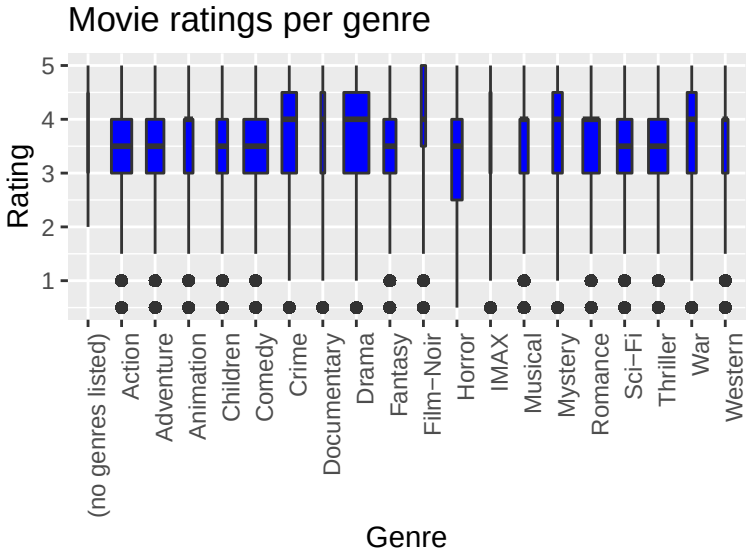


The Gender Effect

Let's explore the effect of genres on movie ratings. It looks that we can exploit some patterns or bias. First, the different genres looks rated differently. The first graph shows the number of movies in each genres and we can appreciate that the Drama genres is rated the most while the IMAX genres is the least rated. However, this doesn't necessarily mean people prefer to rate the Drama movies over other types, because this could simply reflects there are more movies in the Drama genres.

The second graph shows that some genres tend to have higher ratings than the average (such as Film-Noir) and some tend to have lower ratings (such as Horror). However, the genres effect seems to be less relevant.





Regularization

As we can observed there are aome reasons to introduce regularization in the system. The supposed "best" and "worst" movies were rated by very few users, in some cases just 1, and these movies were mostly strangeones. This is because ratings with just a few users, have more uncertainty. Therefore, larger estimates of the biases , negative or positive, are more likely.

These are noisy estimates that we should not trust, especially when it comes to prediction as Large errors can increase our RMSE, so we would rather be conservative when unsure.

Thus regularization and a penalty term will be applied to the models in this project. Regularizations are techniques used to reduce the error by fitting a function appropriately on the given training set and avoid overfitting. Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. The additional term controls the excessively fluctuating function such that the coefficients don't take extreme values.

The Modelling Approach

The aim in this project is to train a machine learning algorithm that predicts user ratings (from 0.5 to 5 stars) using the inputs of a provided subset (edx dataset provided by the staff) to predict movie ratings in a provided validation set.

After the exploratory analysys we will test for several potential Recommendation Systems or algorithms. The different alternatives tested are the following:

- A Naive model (Average) will be estimated as an starting point.
- We will check for the existence of Movie content Effects bias or the fact that some movies are rated better or worse than others in the model (Movie Based)
- The fact that User can have different tastes will be also tested (User Based).
- We will test if different Genres effect bias. In particular we will test Genre Movie and Genres User effects.
- Finally, we will test whether the introduction of Regularization can improve the performance of the models

The value used to evaluate algorithm performance, our loss fuction, is the Root Mean Square Error (RMSE) which is one of the most used measure of the differences between values predicted by a model and the values observed. The RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset, a lower RMSE is better than a higher one. The effect of each error on RMSE is

proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers.

The function that computes the RMSE for vectors of ratings and their corresponding predictors will be the following:

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

with N being the number of user/movie combinations and the sum occurring over all these combinations.

The models that will be developed will be compared using their resulting RMSE in order to assess their quality. The evaluation criteria for this algorithm is a RMSE expected to be lower of a naive model and and RMSE of 0.8649 to obtain the high score for the project.

Results

This section describes the models and results in terms of forecasting accuracy of the different recommendation systems tested in this project. The last section introduces a summary of the main results in terms of the RMSE.

The Naive Mean Baseline Model or Mean

The simplest possible recommendation system is to predict the same rating for all movies regardless of user.A model that assumes the same rating for all movies and users with all the differences explained by random variation would look like this:

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

We know that the estimate that minimizes the RMSE is the least squares estimate of $\hat{\mu}$ is the mean and $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0.

The RMSE on the `validation` dataset for the naive model is **1.06** and still very far for the target RMSE (below 0.8649) and that indicates poor performance for the model.

Movie-Based Model, a Movie Content-based Approach

The first Non-Naive Model takes into account the content of the movies. We know from experience that some movies are just generally rated higher than others. This intuition should beconfirmed by data. We can test this assumption by augment our previous model by adding the term b_i to represent average ranking for movie i. The formula used is:

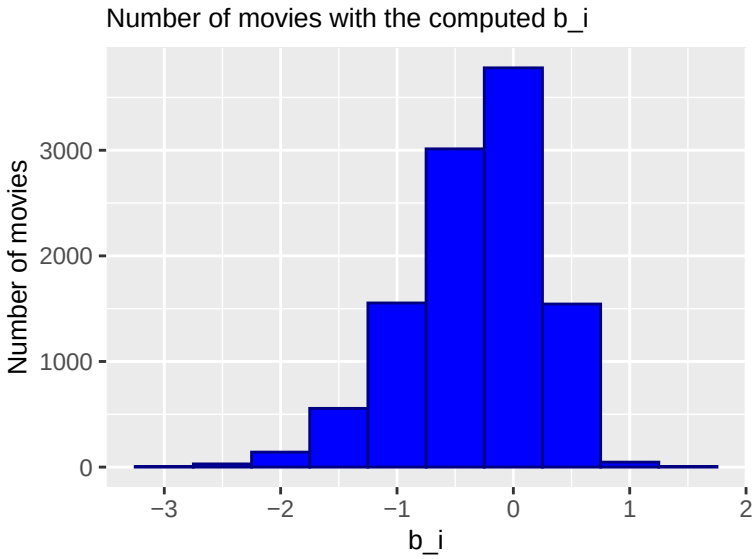
$$Y_{u,i} = \hat{\mu} + b_i + \epsilon_{u,i}$$

In this case, we know that the least squares estimate b_i is just the average of

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

. We can see in the graph that the bias of movie i can still change substantially.

With $\hat{\mu}$ is the mean and $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the popularity of movie i, i.e. the bias of movie i.



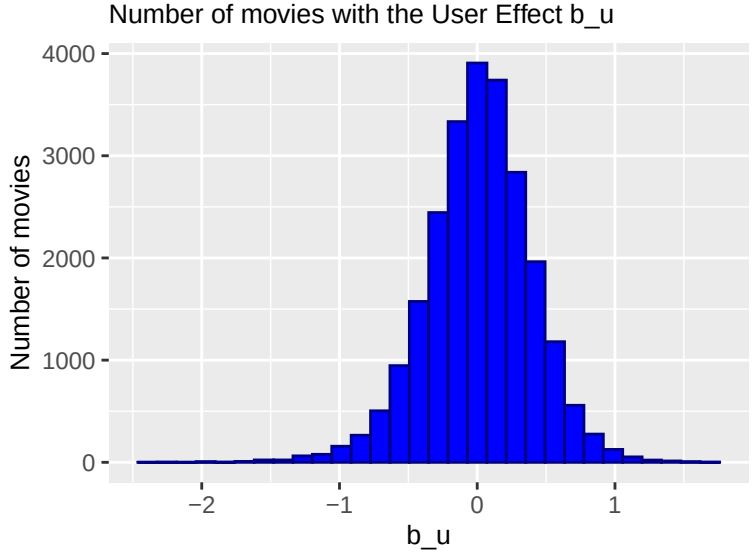
The RMSE on the `validation` dataset is **0.943** so the intuition that the content of the movie matters is confirmed by the model. The model os better than the Naive Mean-Baseline Model, but it is also very far from the target RMSE (below 0.87) indicating that the model can be improved further.

Movie + User Model, a User-based approach

From our exploratory data analysis we know that there is substantial variability across users as well: some users are very negative and others love every movie. This implies that a further improvement to our previous model is to account for the fact that the users have different tastes and rate differently.The formula used is:

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

With $\hat{\mu}$ is the mean and $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the popularity of movie i, i.e. the bias of movie i. The b_u is a measure for the mildness of user u, i.e. the bias of user u is a user-specific effect. Now if a cranky user (negative b_u) rates a great movie (positive b_i), so the effects counter each other and we may be able to correctly predict that this user gave this great movie a 3 rather than a 5. As we can see in the graph the number of movies with the computed b_u is now more concentrated.



The RMSE on the `validation` dataset is **0.865** and this is very good. The Movie+User Based Model reaches the desidered performance but applying the regularization techniques, can improve the performance just a little.

Movie + User + Genres Movie Content Model

From the data exploratory analysis we know that there are some variability in genres. First, it looks that some Genre movies are rated higher than others. This intuition should be confirmed by data. We can test this assumption by augment our previous model by adding the term

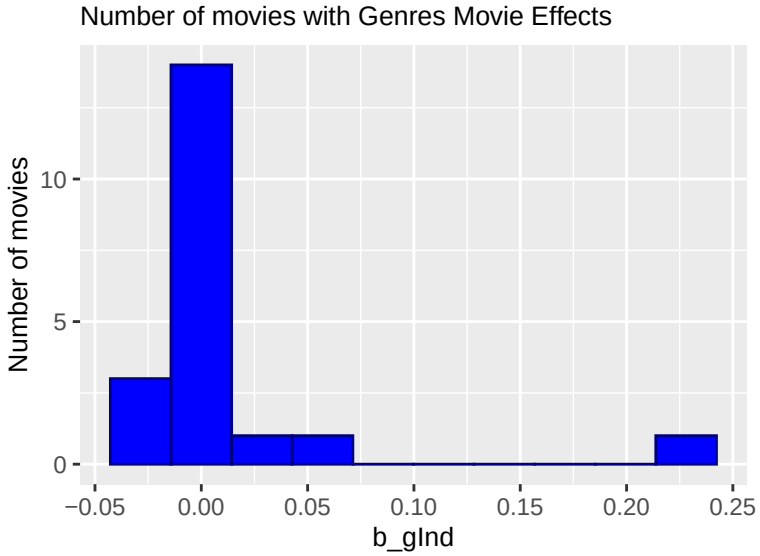
$$b_{i,g}$$

to represent this bias. The formula used is:

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

With $\hat{\mu}$ is the mean and $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the popularity of movie i, i.e. the bias of movie i. The b_u is a measure for the mildness of user u, i.e. the bias of user u. The $b_{i,g}$ is a measure of how movies i of different genres g can be rated differently.

As we can see from the graph of Number of Movies with Genre Movie Effcets the dispersion is much lower now which make the results somehow promising.



The RMSE on the `validation` dataset is **0.8631** and this is very good. The Movie+User+Genres Individual Based Model reaches the desired performance and adding the `genre` predictor improve slightly the model's performance. Later we will check if Applying the regularization techniques, can improve the performance.

Movie + User + Genres Movies + Genres User Model

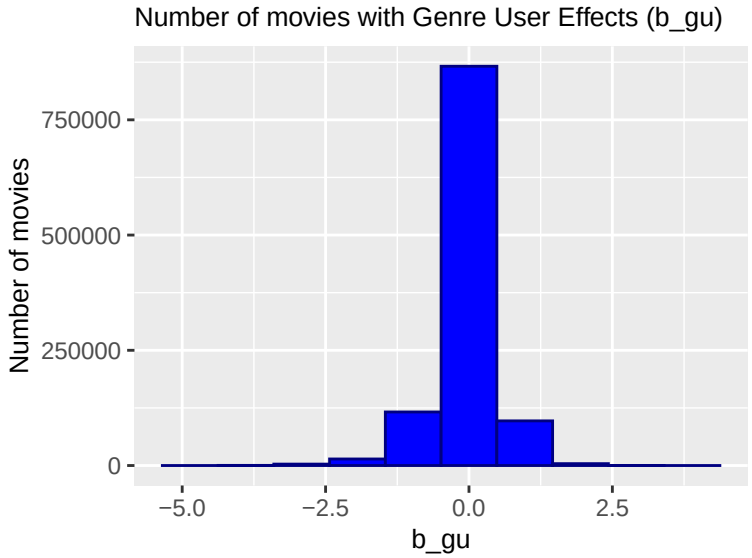
Beyond the fact that some Genre movies are rated higher than others some users can have different tastes and rate different the movies of different genres. We can test this assumption by augmenting again our previous model by adding the term

$$b_{u,g}$$

to represent the user genres bias.The formula used is:

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

With $\hat{\mu}$ is the mean and $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the popularity of movie i, i.e. the bias of movie i. The b_u is a measure for the mildness of user u, i.e. the bias of user u. The $b_{i,g}$ is a measure for how different genres movies i are rated differently and the $b_{u,g}$ is a measure for how much a individual i likes the genre g



The RMSE on the `validation` dataset is **0.8657** the result is good but not better than the movie genre model.

Regularization

Regularization techniques are used to reduce the error by fitting a function appropriately on the given training set and avoid overfitting. Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. This additional term controls the excessively fluctuating function such that the coefficients don't take extreme values.

Regularized Movie-Based Model

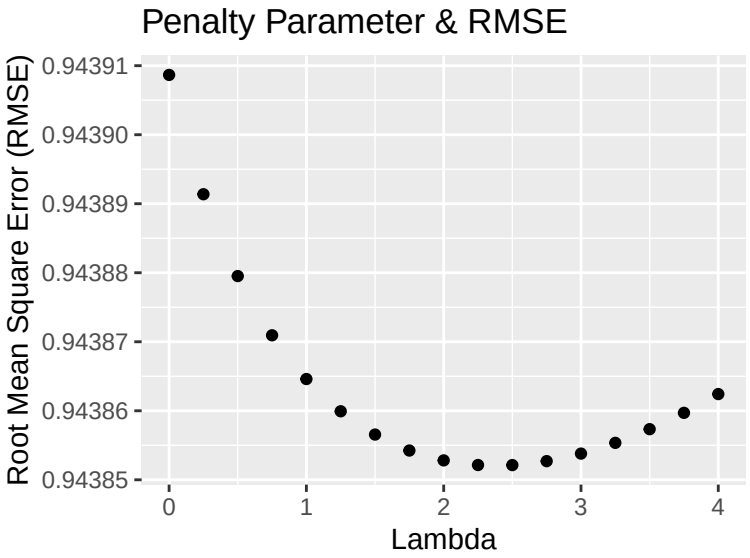
The regularization method allows us to add a penalty λ (lambda) to penalizes movies with large estimates from a small sample size. For our simple model to test (the Movie Content Model) we can optimize b_i by introducin a penalty term in this equation to account for excess fluctuation as follows

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

the new estimated parameter b_i can be expressed as follows:

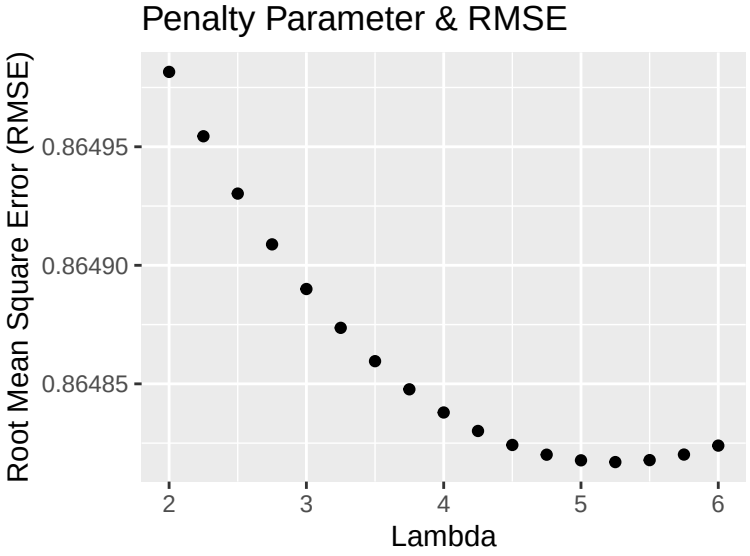
$$\hat{b_i}(\lambda)=\frac{1}{\lambda+n_i}\sum_{u=1}^{n_i}(Y_{u,i}-\hat{\mu})$$

The penalty parameter have to be infered by estimating the lambda parameter which reduces the RMSE in the validation sample. In the following paragraphs we show the graphs of lambdas for the different computed models.



The penalty lambda which reduces the RMSE error is near 2.5 .The RMSE on the `validation` dataset is **0.9438**. This improves marginally the RMSE obtained in the normal Movie Based Model but is far from our best models.

Regularized Movie + User Model

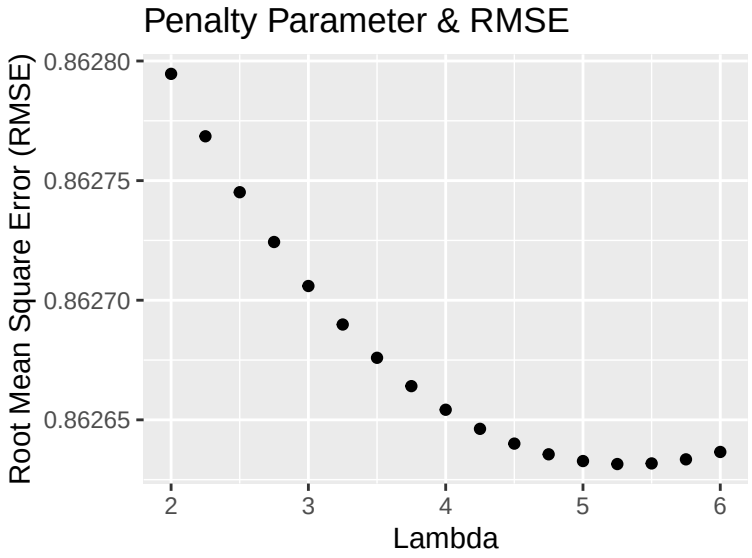


The penalty lambda which reduces the RMSE error is near 5.25 and the RMSE on the `validation` dataset obtained is **0.8648**. The Regularized Movie+User Based Model improves marginally the result of the Movie and User Non-Regularized Model of 0.8653.

Table 3: RMSE of Alternative Models

Method	RMSE
Average Movie Rating Model (Naive)	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Movie + User + Genres Movie Effects Model	0.8631334
Movie + User + Genre Movie + Genre User Effects Model	0.8657017
Regularized Movie Effect Model	0.9438521
Regularized Movie and User Effect Model	0.8648170
Regularized Movie + User + Genre Movie Effects Mode	0.8626315

Regularized Movie + User and Genre Movie Model



The RMSE on the `validation` dataset is **0.8626** and this is the best result of the builted models. The Regularized Movie+User+Genre Based Model improves the result of the Non-Regularized Model of 0.8631. With a RMSE of 0.8626 this is our best model and the RMSE is below the threshold (0.8649) requested to obtain the maximum score.

Summary of the Results

The results of the different recommendation systems can be observed in Table 3. The table shows the RMSE of the alternative recommendation systems trained in the edx sample and tested in the validation sample.

The results show that both the Movie and User content outperform significantly the Naive (Average Model) model. Particularly, the Movie and User Effect Model reduces the RMSE to 0.865 from the 1.06 RMSE of the naive model. The Genre Movie effects models also add some gains in terms of forecast accuracy but marginally. By adding the Genre Movie effects of the Movie and User effect model, the RMSE is reduced further to 0.8631, which is already below the best benchmark requested of 0.8649

There are additional gains coming from accounting from the introduction of regularization. Particularly, introducing penalties to our models marginally reduce some of the models. The best model is the Regularized Movie and User adding Genre Movie Effects. This model reduce the RMSE further to 0.8626 which clearly outperform the best requested RMSE of 0.8649 This is the summary results for all the model builted, trained on `edx` dataset and validated on the `validation` dataset. (target 0.8649)

Conclusion

In this project we have analyzed the Movielens data set to propose a recommendation System for Movies. After analyzing the statistics and graph properties of the data we decide to account for some of the information included in the data base. Particularly, the Movie and User biases can be used to improve notably a recommendation system. Beside, adding the Genre bias by Movies to the former effects can be used to improve the properties of the algorithm.

In the project we also show how the introduction of regularization through penalties in the error term can improved further the accuracy of the model.

While the biases and regularization were enough to reduce the RMSE error of some of our models below the lower error requested there is margin to check for further improvements. First, we can add some extra information as potential time bias (checking for changes over time of the Movies and User effects). Second, we can explore for some other techniques exploiting the correlation between movies or users. Particularly we can check for Matrix Factorization techniques (PCA, SVD) to improve further the properties of the recommendation system for MovieLens data.

After training different models, it's very clear that `movieId` and `userId` contribute more than the `genre` predictor. Without regularization, the model can archieves and overtakes the desidered peformance, but the best is the enemy of the good and applying regularization and adding the `genre` predictor, it make possible to reach a RSME of **0.8626** that is the best result for the trained models.