Movie recommendation system

Vincent Pluchet

23/04/2021

Content

This reports contains four sections:

1. **Executive Summary**
2. **Methods and Analysis**
3. **Results**
4. **Conclusion**

1 - Executive Summary

Goal of the study and data used

The goal of this study was to build a movie recommendation system. The algorithm developed under this study assigns predicted ratings per movie and user. The algorithm takes into account user preferences, movie ratings by other users, and other factors described in more details below.

The algorithm was developed with the R programming language. It was trained on a set of data ("edx") containing more than **9 million** ratings. The final algorithm was then tested on a completely independent set of data ("validation"), containing around **1 million** ratings.

Results

The overall accuracy of the model was measured using a root-mean-square error (RMSE) approach, which measures the differences between the predicted ratings and the actual ratings observed in the validation set. The RMSE obtained with the final model was **0.8542**, which was around 1% better than the targeted RMSE (0.8649).

Method used

For the purpose of model development, the edx data set was partitioned in an 80% training set and a 20% testing set, until the final algorithm was defined. The final model was then re-trained on the full edx data and subject to final testing with the validation set. The validation set was not used at any other stage than the final assessment.

The approach used was to build a linear model of the type (for user u and movie i):

$$rating_{u,i} = \mu + b_i + b_u + b_g + b_w + b_{u,g} + \epsilon$$

- The constant mu represents the average rating in the training set
- b_i represents the movie effect, ie the fact that certain movies get above average ratings and others get below average ratings
- b_u represents the user effect, with certain users assigning ratings higher than average (more "lenient" users) and others lower than average ("harsher" users)
- b_g represents the genre effect, taking into account the fact that certain cinema genres get lower ratings (Horror for instance) than others (War movies tend to have high ratings)
- b_w represents the week effect, as a certain time effect was observed on ratings. Week represents the number of weeks elapsed since the first date in the data base
- $b_{u,g}$ represents the fact that users have preferred genres, to which they will give higher ratings than other genres. For instance a user who likes comedies and hates horror shows will tend to rate comedies higher and horror movies lower. Cluster analysis was used to assess this component as described further down.
- epsilon represents a random error term

Linear modeling or generalized linear modeling was not used, due to the size of the data not allowing such calculations on a laptop. Rather, model components were computed recursively, by taking their means after other effects were factored. Regularization was also introduced in order to avoid biases due to movies, users or genres having few ratings in the data base.

In order to assess the user-genre effect $b_{u,g}$, genres were clustered using kmeans clustering. Indeed, there were 797 genres in the data set. This is due to the fact that movies can be tagged to different genres, thus generating many possible combinations (for instance Action|Comedy or Action|Thriller). Using all of the resulting genres would have led to heavy calculations and over-fitting. In addition, user-genre combinations would appear in future sets without being present in the model. To avoid these problems, the 797 genres were clustered into 9 genres only, reflecting the closeness between certain genres: for instance "Action|Drama" would typically be in the same group as "Adventure|Drama". Kmeans clustering was used, on a set of user ratings per genre, in order to define "groups of genres" (or genres clusters). The final number of clusters (9) was chosen using cross-validation on the edx data set.

2- Methods and Analysis

Initial review of the data

The edx data is a significant data base, containing 9 million ratings. It contains the following fields:

- userId: a unique identifier for each user who rated a movie
- movieId: a unique identifier for each rated movie
- rating: a numerical rating given by the user for the corresponding movie
- timestamp: the timestamp identifying the date of the movie
- genres: the genre of the movie. A movie can be assigned to a unique genres, like "Comedy", or to several, in which case all genres are clubbed: "Action|Crime|Thriller".

The number of Users, Movies and Genres recorded in the edx table is as follows:

Table 1: EDX Data Count

Users	Movies	Genres
69,878	10,677	797

Data cleaning

An inspection of the table shows that there are no NA (not available) userId, movieId, ratings, genres or timestamp. This is important as these factors will be used in the model. We also checked that each movie is tagged to one genre only (taking genre combinations as specific genres in themselves). Therefore, no specific data cleaning was required.

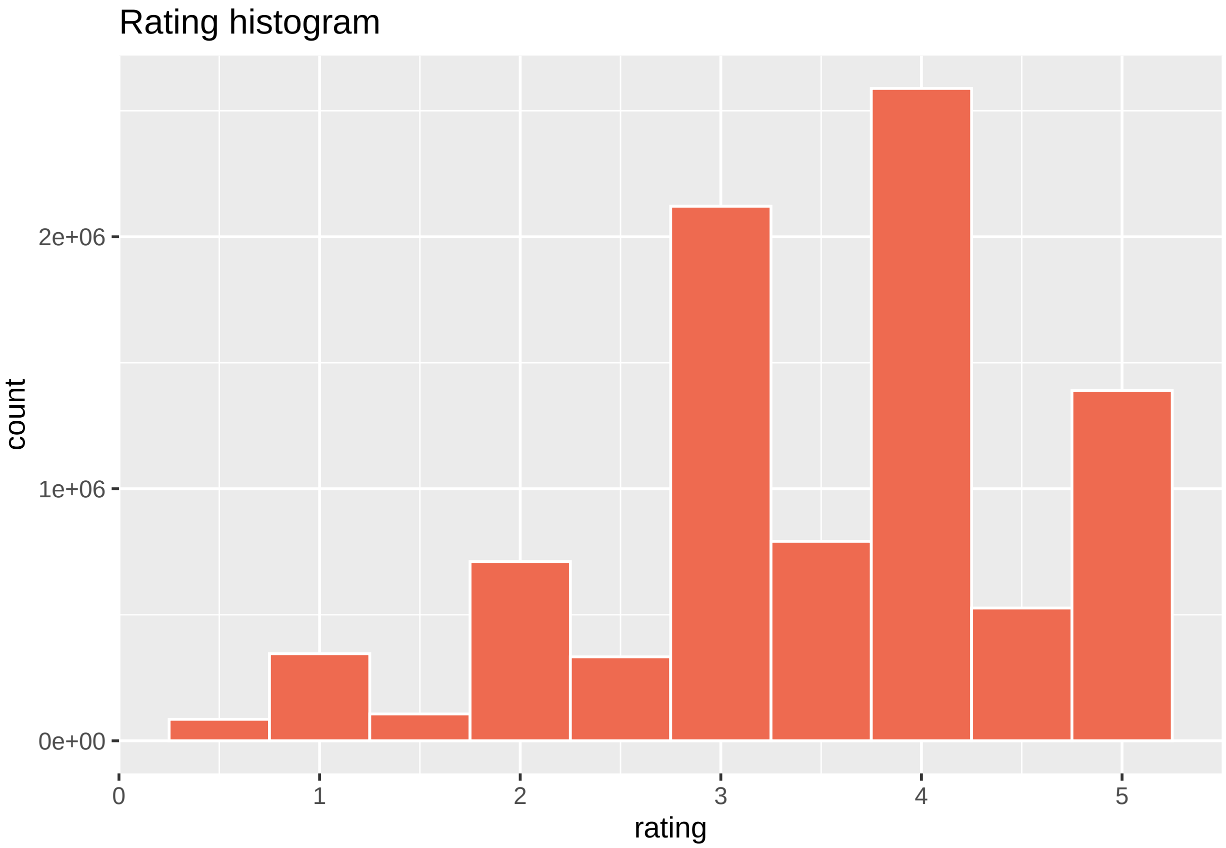
Note:

Please note that the remainder of this section contains data relative to the training set (ie 80% of the edx data file) used to train the models. Conclusions remain similar when using the full edx data set.

The data shown includes regularization, which will be discussed in more details at the end of this section.

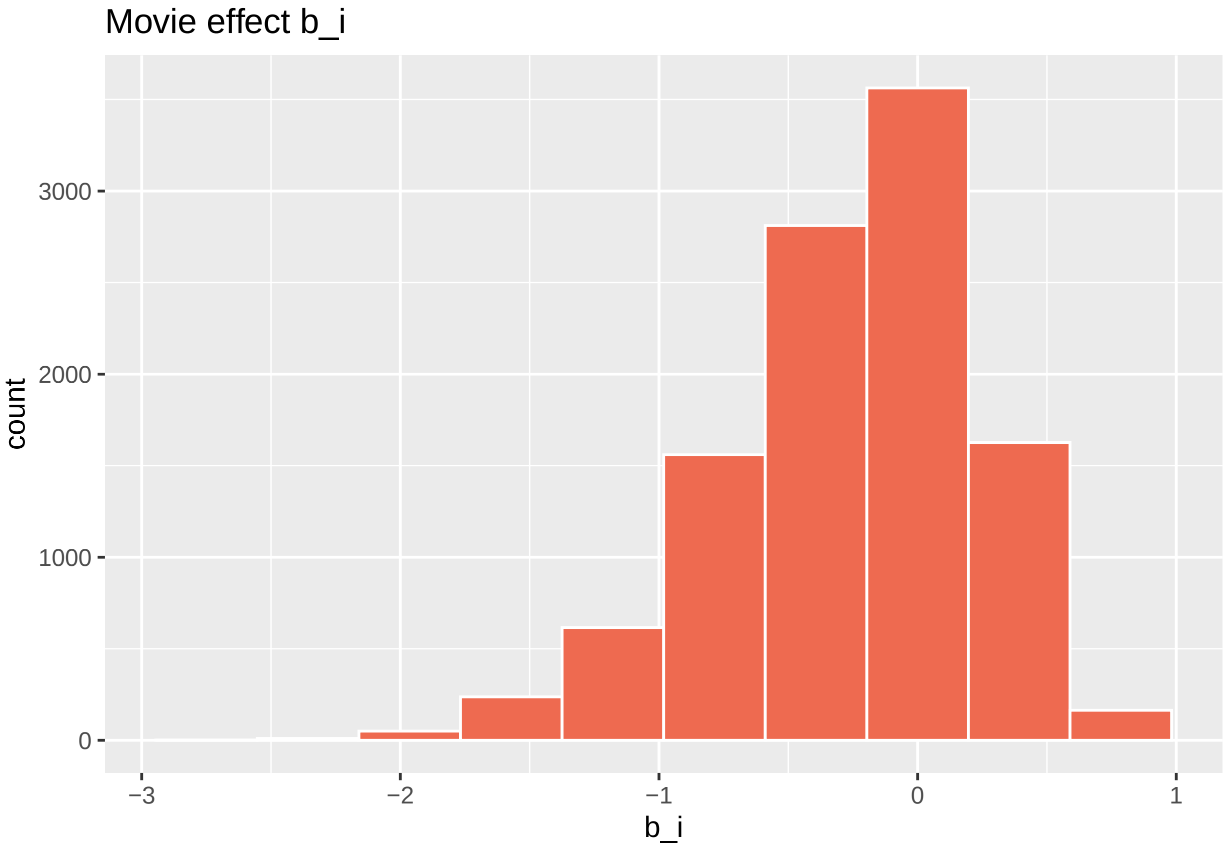
Rating histogram

The graph shows that users tend to give ratings between 3 and 5 rather than less than 3. Half ratings are less frequently used. The most frequently given rating is 4, followed by 3 then 5. In fact, the median rating is 4 and the mean rating is 3.512.



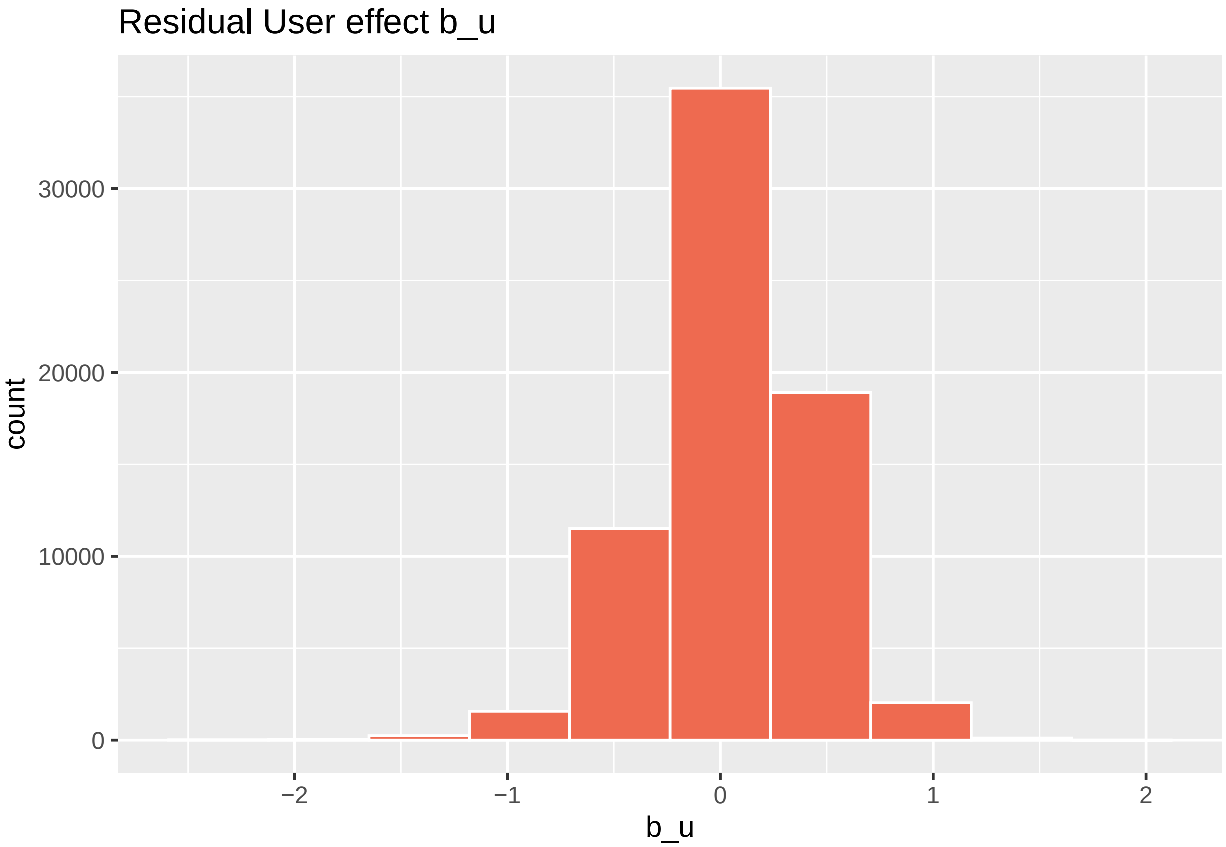
Viewing the Movie Effect

The plot shows how movies ratings deviate from the overall mean rating. Some movies rate better than average, others are rated below average. This is not surprising: movies can be well received (positive movie effect b_i) or badly received (negative movie effect b_i). This effect should definitely be captured in the model.



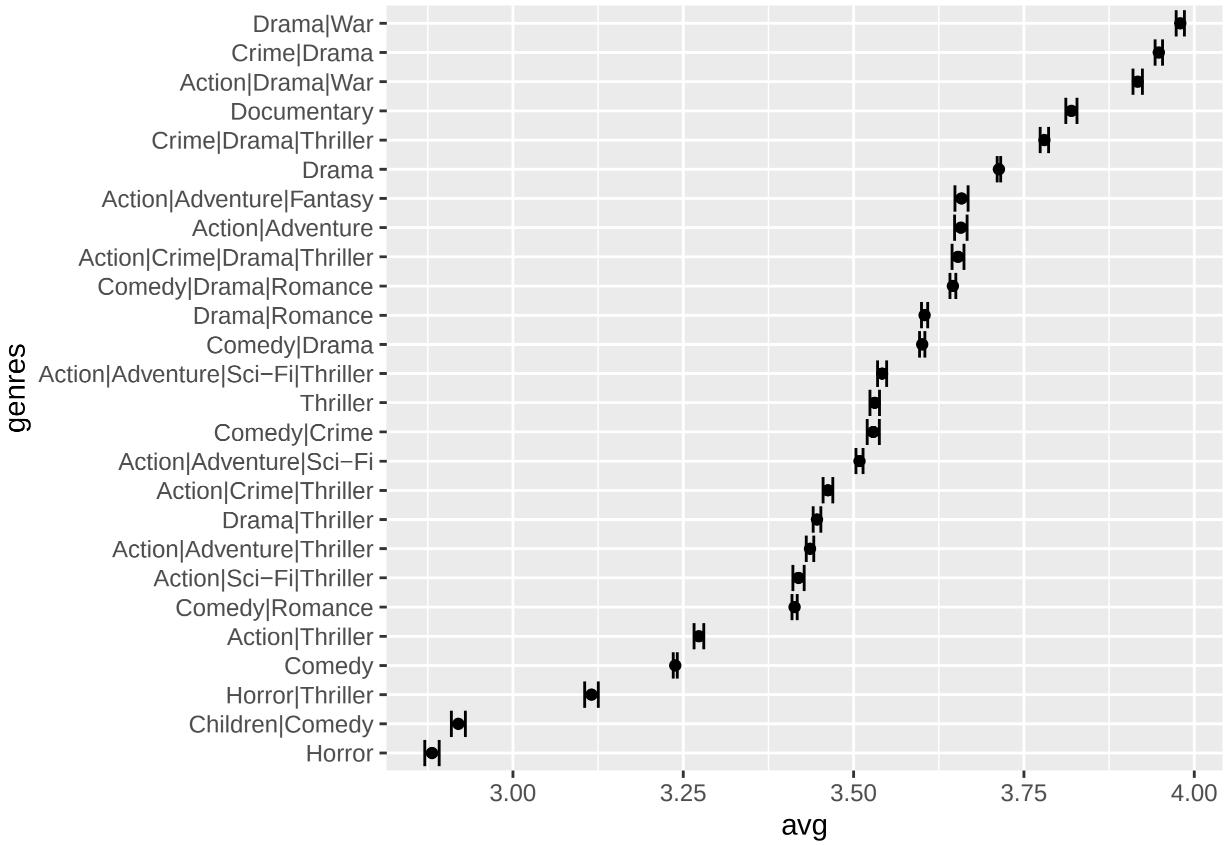
Viewing the User effect

Once the movie effect is factored, a user effect is measured ("residual" user effect). The plot shows that some users tend to give higher ratings (positive effect), whereas others give lower ratings (negative effect). This effect should therefore also be captured in the model.



Viewing the Genre effect

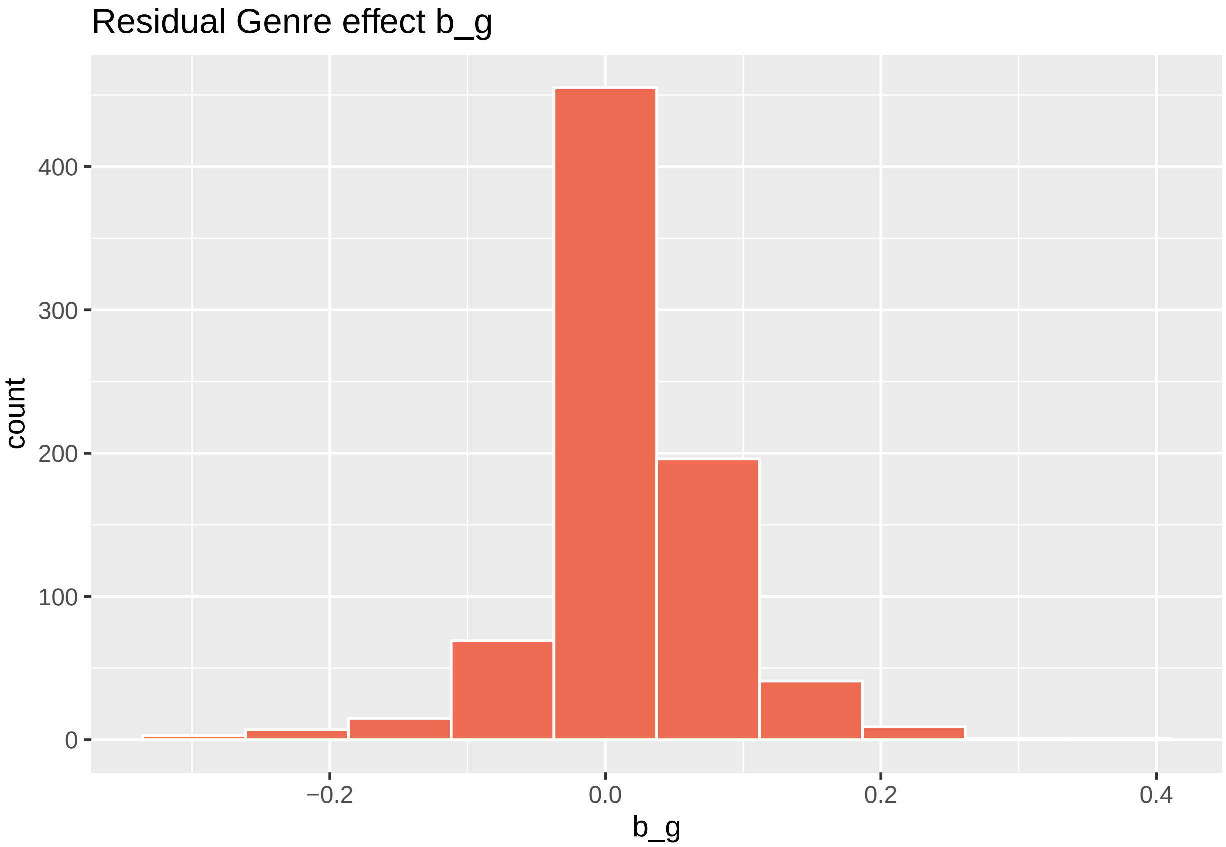
An error-plot of the most rated genres shows a clear genre effect, before taking into account other factors. This confirms that we should try to add a genre effect to the model.



As explained earlier, movies can be tagged to different genres, thus generating many possible combinations (for instance Comedy|Romance or Comedy|Crime). We chose not to split the genres into individual components. The reason is two-fold:

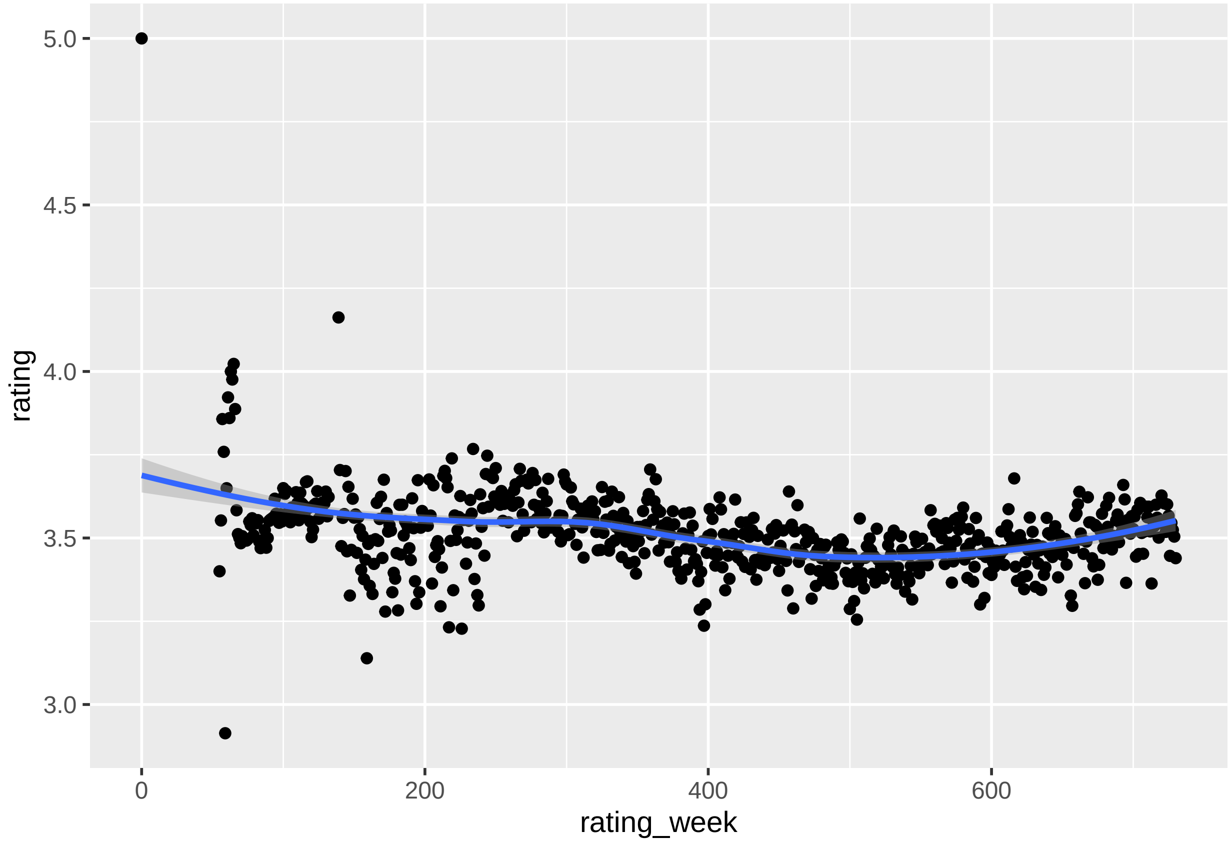
- "Comedy" may not have the same impact when it is tagged along Romance or along Crime. Therefore there is perhaps not a "Comedy" alone effect.
- We will use clustering to group Genres when assessing the User-Genre effect, therefore splitting genres is not really necessary: if two genres are similar in terms of how users rate them, say Thriller and Action|Adventure|Sci-Fi|Thriller, then they will end up in the same cluster.

Computing the residual genre effect (net of movie and user effects) displays the following graph, which confirms that a genre component should be used in the model:

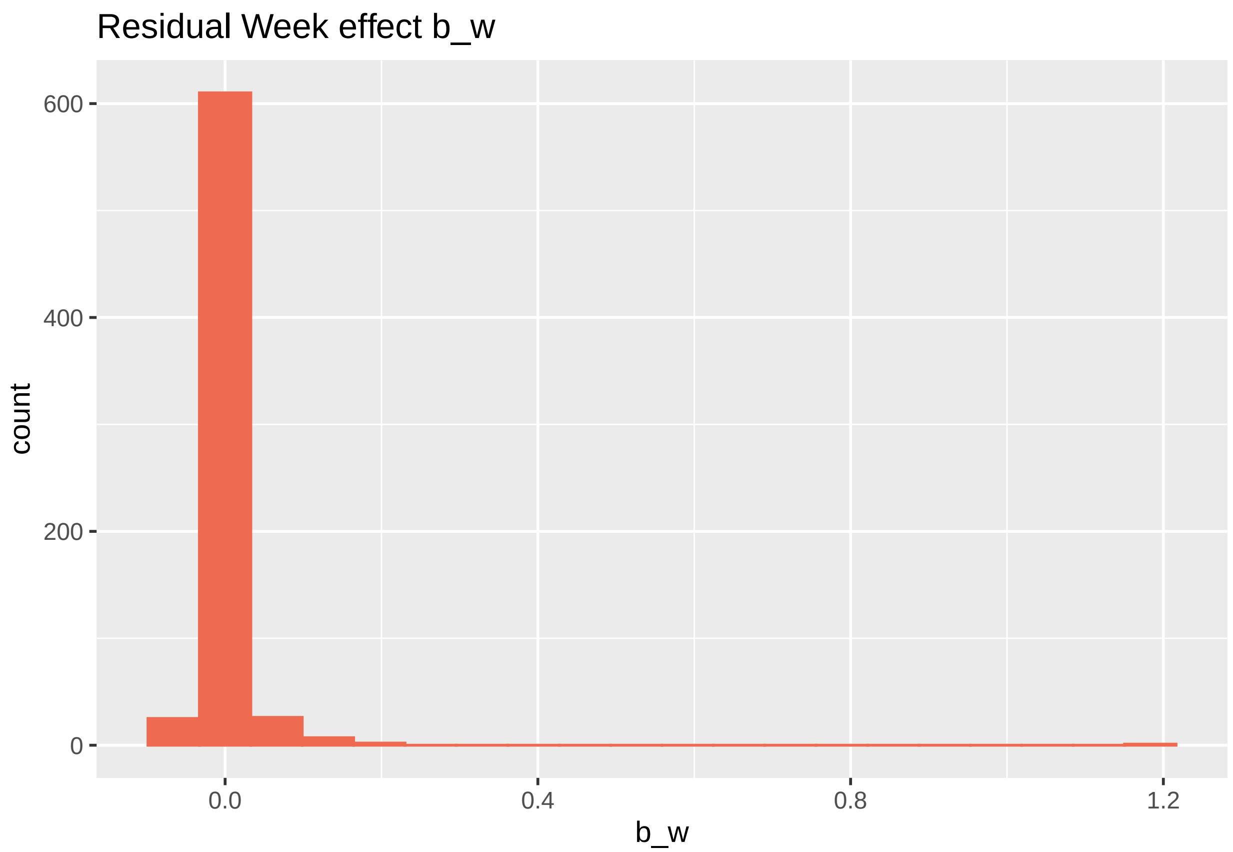


Viewing the Time effect

The number of weeks elapsed since the first movie date in the data set seems to have an impact on the ratings, as can be seen on the below plot showing average rating per week. The impact is not very strong but should probably not be ignored:



As a consequence, a residual week-effect (after factoring the user, movie and genre effects) was computed. The plot shows that it is indeed not insignificant:


Residual Week effect b_w
count
600
400
200
0
0.0
0.4
0.8
1.2
b_w

Viewing the User-Genre effect

Using a simple model with distinct movie, user, genre and time effects would look like this:

$$rating_{u,i} = \mu + b_i + b_u + b_g + b_w + \epsilon$$

This does not feel satisfactory however. As stated in the Executive Summary, users have preferences for certain genres and dislike other genres. This effect must be factored in the model, because otherwise we could end up recommending a Romantic Comedy movie to someone who only likes Thrillers.

On the other hand, there is no point trying to measure the user-genre effect for genres which are seldom used. The data shows indeed a significant disparity between genres which are used very often and others, which are seldom used.

Certain genres are used very often:

Table 2: Most used Genres

genres	count
Drama	586,784
Comedy	560,998
Comedy|Romance	292,545
Comedy|Drama	259,016
Comedy|Drama|Romance	208,890
Drama|Romance	207,101
Action|Adventure|Sci-Fi	176,134
Action|Adventure|Thriller	119,294
Drama|Thriller	116,102
Crime|Drama	109,915

Certain genres are seldom used:

Table 3: Least used Genres

genres	count
Action|Animation|Comedy|Horror	1
Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi	1
Adventure|Mystery	1
Action|Adventure|Animation|Comedy|Sci-Fi	2
Action|War|Western	2
Animation|Documentary|War	2
Crime|Drama|Horror|Sci-Fi	2
Documentary|Romance	2
Drama|Horror|Mystery|Sci-Fi|Thriller	2
Drama|Musical|Thriller	2
Fantasy|Mystery|Sci-Fi|War	2

This confirms that genre clustering would be useful.

In order to build these clusters, we proceed as follows:

- A table with average ratings per genre per user is created. The table is filtered in order to contain only users with more than 200 ratings. This is done to reduce table size but also to build genre clusters without distortion from users with a limited number of ratings. We do check, however, that the table contains all genres.
- This table is converted to a matrix with genres as rows and userId as predictors (columns)
- kmeans clustering is then used to define the clusters. As explained in the Executive Summary, we opted for 9 clusters, after performing a cross-validation exercise with various numbers of clusters. Please see regularization section further down for more details.
- Once the clusters are defined, the information is brought into the training set and the user-clusters combinations are added. The residual effect is then computed by computing the mean rating net of all the other effects previously mentioned. A specific regularization factor is used, as discussed in the regularization section.

Description of the clusters:

All genres are clustered (total of 797 allocated to 9 clusters). There is one very large group, whereas the other groups are of more comparable size. In spite of having one group much larger than the others, the share of within-variance is relatively well spread. This tends to indicate that there is a bulk of genres which are not highly differentiated in the way users rate them, and a number of other genres which are rated in different ways by users.

Table 4: Group Size and Variance

Group	Size	Variance_Within	Share_of_Variance
1	53	913,833	12 %
2	30	655,265	8 %
3	26	591,775	8 %
4	32	636,677	8 %
5	473	1,535,901	20 %
6	36	791,543	10 %
7	55	883,706	11 %
8	49	988,417	13 %
9	43	876,793	11 %

The variance analysis below shows that the Between variance is relatively small, confirming that the clusters are generally well defined:

Table 5: Variance analysis

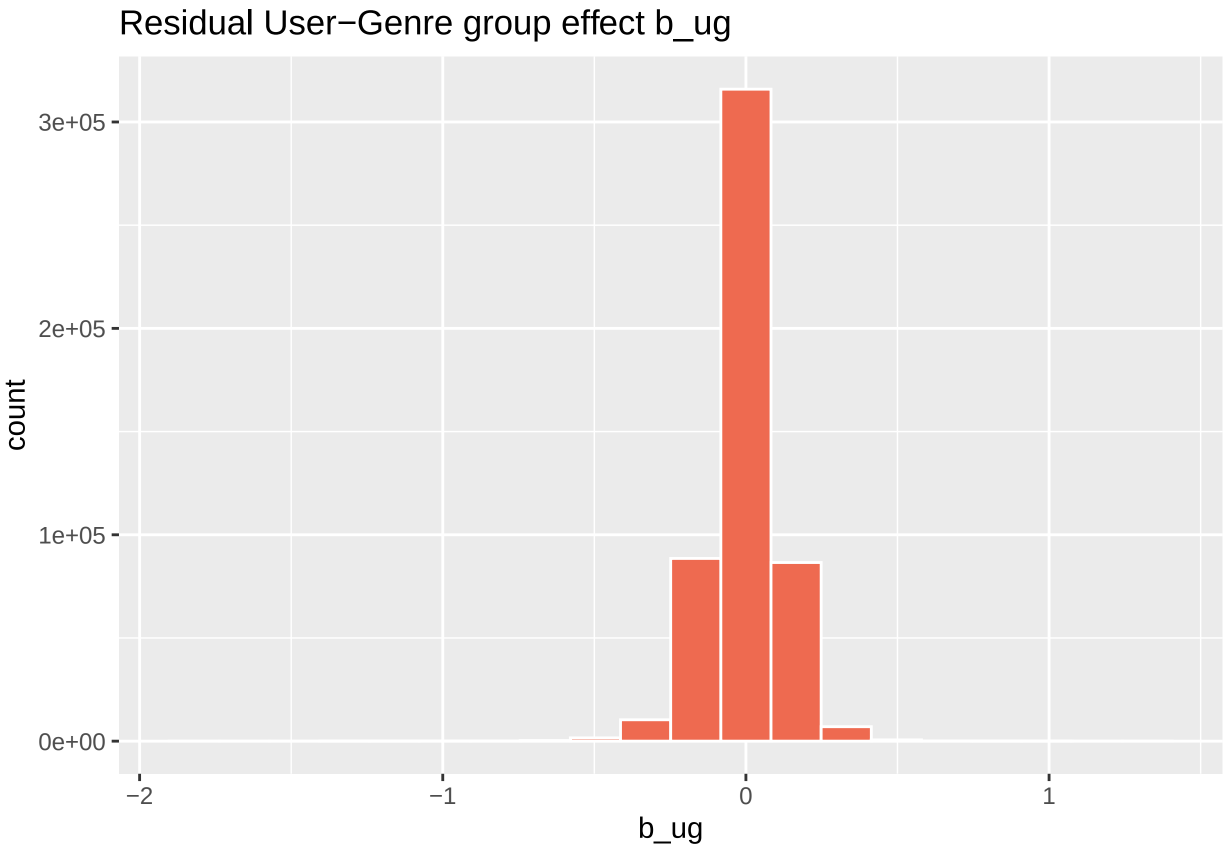
Variance_Type	Variance	Share_of_Variance
Within	7,873,910	89 %
Between	987,184	11 %
Total	8,861,094	100 %

As an example, we provide below the top genres listed in the second group. As the table shows, the movies relate mostly to family movies with children, comedy and romance genres:

Table 6: Group 2 - Genres with highest count

genre_group	genres	count
2	Children|Comedy	50,839
2	Comedy|Sci-Fi	35,667
2	Adventure|Comedy|Sci-Fi	34,979
2	Comedy|Drama|Fantasy|Romance	34,530
2	Adventure|Comedy	25,726
2	Comedy|Musical	24,753
2	Comedy|Fantasy|Romance	22,424
2	Comedy|Fantasy	15,862
2	Children|Drama|Sci-Fi	12,696
2	Adventure|Comedy|Drama	12,617

Once the clusters are defined, the user-genre effect is computed, net of other effects (movie, user, genres, time) and using regularization. The graph below shows the user-genre effect and confirms that it is not negligible.



Regularization

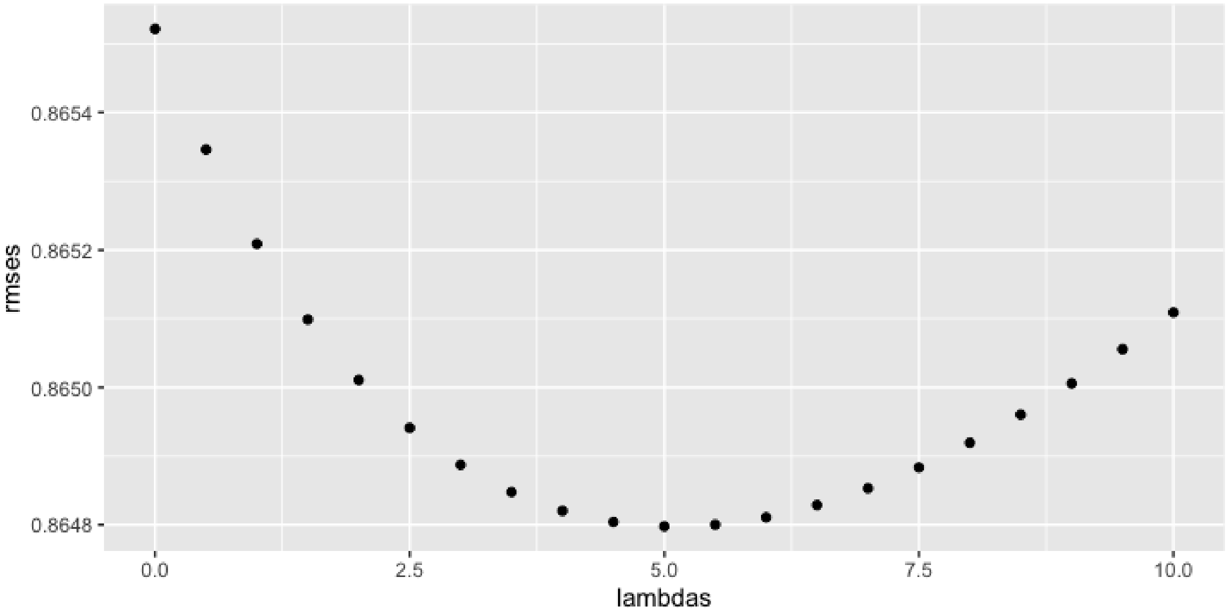
Regularization factors are introduced to remove biases from users, movies or genres with very few ratings. Regularization factors are added to observation counts when computing averages, thus decreasing the impact of small counts, whereas for larger counts, adding the factor does not have a material impact. For simplicity, two regularization factors are computed:

- One for the movies, user and genre effect
- One for the user-group effect

Cross-validation on the training set (80% of edx data) and test set (20% of edx data) was used to determine the regularization factors. Cross-validation was also used to determine the number of clusters to be used for genres. As there is a potential link between the user_group regularization factor and the number of clusters, both were evaluated recursively.

Regularization for the movies, user and genre effects:

The below graph shows that the overall RMSE reduction versus nil regularization is not highly significant for movies, user and genre effect, however there is a small effect and the optimal regularization factor is 5 :



Number of clusters:

The number of clusters was carefully considered. Using too few clusters may lead to an over-simplification of the user-genre effect. On the other hand, too many clusters could lead to over-fitting. Many user-cluster combinations may also appear in future data sets which were not captured in the training set, in which case the user-effect could not be used.

Cross-validation was used to determine an acceptable number of clusters. The NbClust package was also tried, as it provides information on the optimal number of clusters, but the running time was too long for practical use.

Examples of the results obtained are shown in the below graph (other ranges were used but only a selection is shown here for simplicity). Whilst there is not an obvious trend, the optimal number of cluster in the graph is 9.

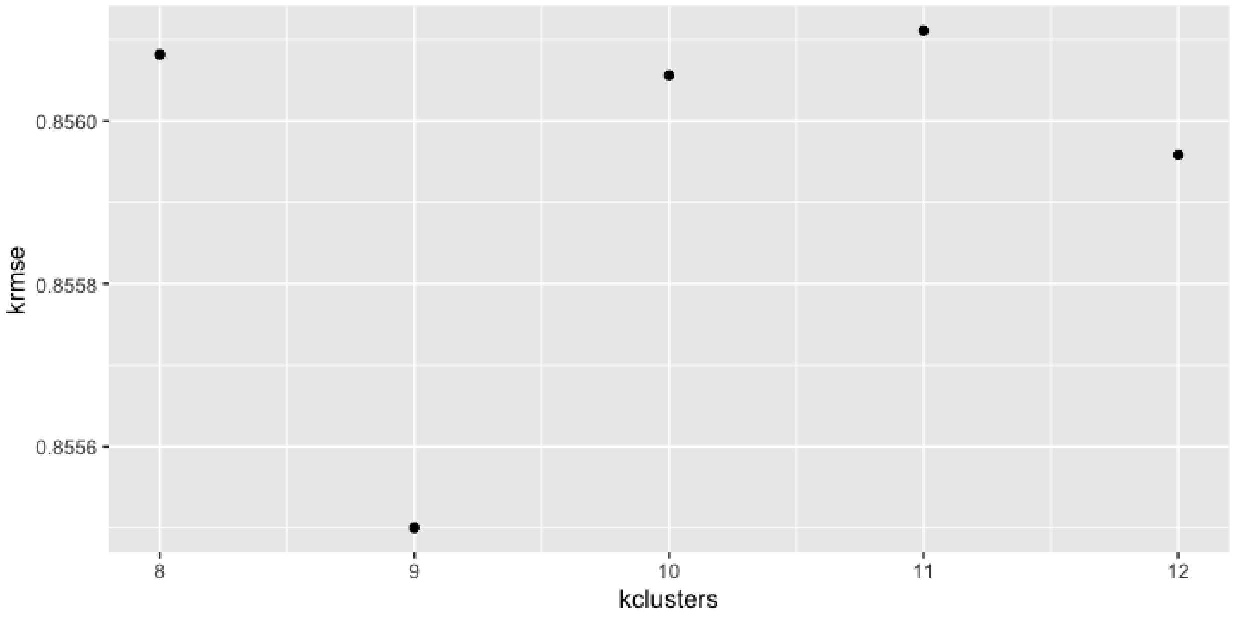


Figure 1: RMSE versus various number of clusters

Regularization for the user-genre effect:

The optimal regularization factor for user-genre effect is 15 as shown in the below graph:

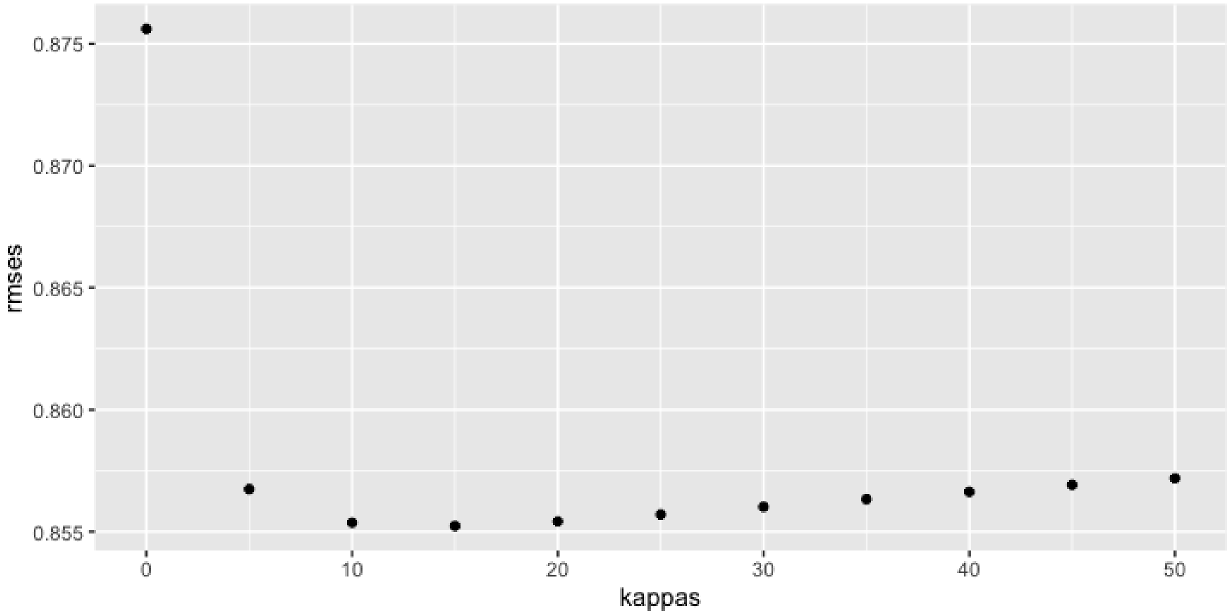


Figure 2: RMSE versus various regularization factors

The result of the cross-validation exercise is therefore as follows:

- Regularization factor for users, movies and genres: 5
- Number of clusters for genres: 9
- Regularization factor for users-genres effect: 15

As described in the Executive Summary, the final model takes the following form, for user u and movie i:

$$rating_{u,i} = \mu + b_i + b_u + b_g + b_w + b_{u,g} + \epsilon$$

The definition of each component is provided in the Executive Summary.

3 - Results

The models fitted on the training set (80% of edx) are evaluated against the test set (20% of edx).

Results show that:

- The first two components (movie and user effects) have a very significant impact on the model accuracy (this could be however because they are introduced first)
- Adding the genre and week/time effect adds further improvements but more marginally
- The user-genres effect adds a definite final improvement to the model. This improvement is around 1%

Test set RMSEs for each model:

(training set = 80% of edx, test set = 20% of edx)

Table 7: Models tested on test_edx set

method	RMSE
Simple average model	1.0599
Movie Effect Model	0.9437
Movie and User effect	0.8652
Movie+User+Genre effect	0.8649
Adding week effect	0.8648
With User Group effect	0.8552

For information, every model was also tested against the Validation test, after the model was fitted on the full edx set. Results are as follows and consistent with the results on the test set:

Table 8: Models tested on Validation set

method	RMSE
Simple average model	1.0612
Movie Effect Model	0.9439
Movie and User effect	0.8648
Movie+User+Genre effect	0.8645
Adding week effect	0.8643
FINAL with User Group effect	0.8542

4 - Conclusion and recommendations

The final developed algorithm allows to predict movie ratings with an RMSE lower than targeted. Several improvements could be considered in future studies:

- The order in which factors are added may influence the overall result. For instance, should the user-genre effect be introduced first, would the result be different? The impact of measuring the factors in a different order could be assessed.
- Generalized Linear Modeling could be tested, if computing power permits. This would allow to de-correlate the various effects more efficiently.
- More work could be done on Genres or Users clustering. Other techniques such as matrix factorization could be envisaged.
- The definition of "genres" in the data itself should probably be reviewed to be made more systematic. Is there a real difference, for instance, between Thriller and Drama|Thriller? A more compact list of genres in the data itself may allow more precise predictions.

END OF REPORT