MovieLens Recommendation System Project

HarvardX Data Science Professional Certificate: PH125.9x
Data Science Initiative
The Palestinian Central Bureau of Statistics (PCBS)
Arab American University of Palestine (AAUP)

Mohammed K. Elhabbash

25 January, 2022

Contents

1.Introduction

A MovieLens recommendation system is an automated system that attempts to anticipate a user's preference for a movie to watch, depending on the previous user's rating. In general recommendation, systems are employed to suggest items. These items vary according to the field, for instance, books, news, research articles, search queries, movies, restaurants, etc. Suggested items are selected from a pre-prepared data set. The function of the recommendation system is to select item/s from this data set and recommend it/them to the user. Famous companies such as Amazon, Twitter, Facebook, Netflix and Spotify rely on recommendation systems to promote their commodity and satisfy customers desires. The following report is a response to a mandatory assignment in the ninth and final course in HarvardX's multi-part " Data Science Professional Certificate". The Project Instructions, and Project Submission is given in appendix A, and appendix B respectively.

1.1 Project methodology:

10M Dataset will be split into two portions edx, and validation dataset. **edx dataset:** 90% of 10M Dataset **validation dataset:** 10% of 10M Dataset

edx dataset will be cleaned, analysed and visualized to explore features of the dataset. Machine learning techniques and models will be employed to form a model for Movie suggestions. Models will be built rely on the edx dataset. The validation dataset then will be used to test the performance and effectiveness of the model depending on RMSE.

1.2 Dataset:

MovieLens Latest dataset is a large, continuous-updated dataset that involves the following features: **Quantitative features.**

- userId: an integer variable with a unique ID for each user.
- movieId: a numeric variable with a unique ID for each movie.
- timestamp: an integer variable represents the date and time of the ratting that was given to the movie.

Qualitative features.

- title: character variable which names the movie (not unique).
- genres: character variable associated with the movie.

Outcome:

- rating: a numeric variable lay between 0.5 and 5 express the satisfaction of the user toward a specific movie. (5 is the best value).

The 10M Dataset is the used dataset in this project, which is last updated in Sep./2018.

Table 1: *Summary of the edx data set*

rows_number	columns_number	users_number	movies_number	average_rating	genres_number	first_rating_Date	last_rating_date
9000055	6	69878	10677	3.512	797	1995-01-09	2009-01-05

1.2.1 edx dataset:

edx as a subset of the 10M MovieLens data set of 9,000,055 raws and 6 columns. 10677 movies were rated by 69878 users, where movies have 797 genres with an average rating of 3.512 The ratting process starts in 1995 and ends in 2009.

The columns represent features i.e. userId, movieId, rating, timestamp, title, and genres. Each user will take a unique Id. and each movie has also a unique Id. The user has the opportunity to rate the movies which he/she watched. The ratting is a number between 0.5 and 5 where 5 is the best. edx consist of 69,878 unique users and 10677 unique movies. The rating column is the outcome,y, which we train the model to predict.

Table 2: *The first five rows of the edx data set*

userId	movieId	rating	timestamp	title	genres
integer	numeric	numeric	integer	character	character
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

1.2.2 validation dataset:

The validation dataset has nearly the same characteristic as the edx dataset and is summarized as follows:

Table 3: *Validation data set summary*

rows_number	columns_number	users_number	movies_number	average_rating	genres_number	first_rating_Date	last_rating_date
999999	6	68534	9809	3.512	773	1995-01-09	2009-01-05

1.3.Analysing and exploration of data

A brief synopsis of each feature in the edx dataset is presented in the following.

1.3.1 Investigation of the variable "rating"

Rating, as a statistics variable, is an ordinary scale of numbers. Practically rating is an evaluation number given to the movie by the users i.e. rating: (0.5 ~5, 0.5) where 5 is the best. An illustrative figure below clarifies the portion of each rating from the total number of ratings that users give to movies, whatever the movie.

- The overall average rating in the edx dataset was 3.51
- The top 3 ratings from most to least are 4, 3, 5.
- Users desire to rate movies more positively than negatively.
- The histogram shows that the half-star ratings are less common than whole star ratings.

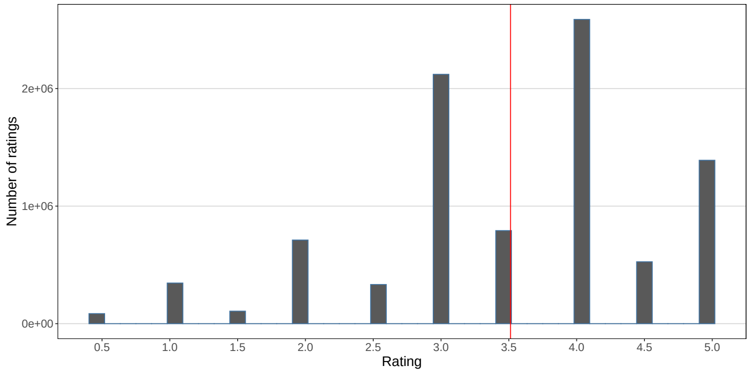


Figure 1: *Overall rating distribution in Edx dataset*

1.3.2 Investigation of Movie

The Edx dataset contains a total of 10,677 movies, each represented by a movieId.

- Some movies are rated more than other, and the average number of rating is 843.
- Approximate 20 % number of movies have a number of ratings more than the average, which represents approximately 85% of ratings.
- The average movie rating tends to increase when the number of rating increases.

The histogram below depicts the number of ratings by a movie.

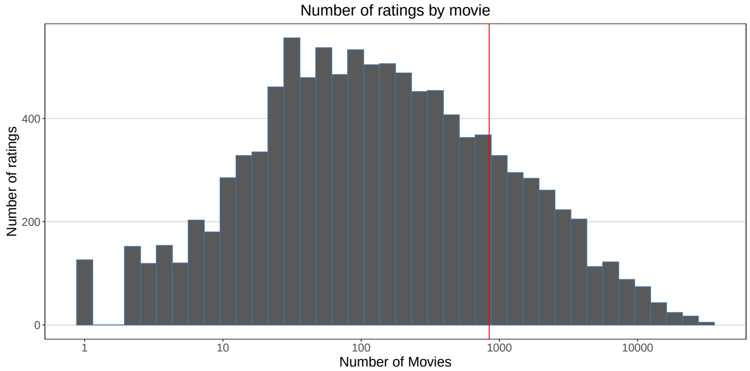


Figure 2: *Number of ratings by movies in Edx Dataset*

The histogram beneath depicts the distribution of movies based on an average rating.

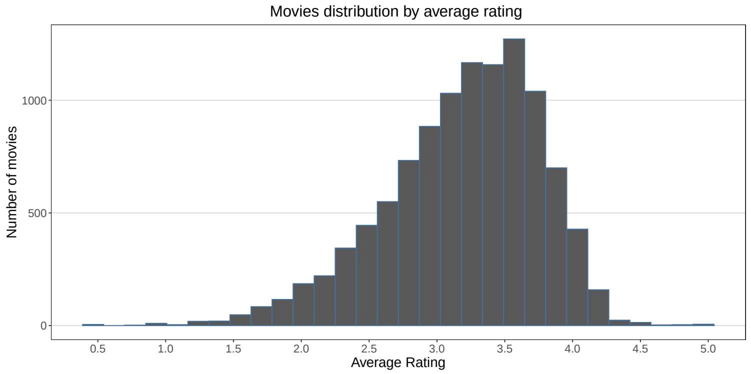


Figure 3: *Movie distribution by average rating in Edx Dataset*

1.3.3 Investigation of User

The edx dataset contains 69,878 individuals identified by userId.

- 30 % of users contribute 70 % of ratings in the whole edx dataset.
- Some users rated very few movies and their opinion may bias the prediction results.
- The average user rating tends to increase when the number of rating increases.

The number of ratings by a user is represented by the histogram below

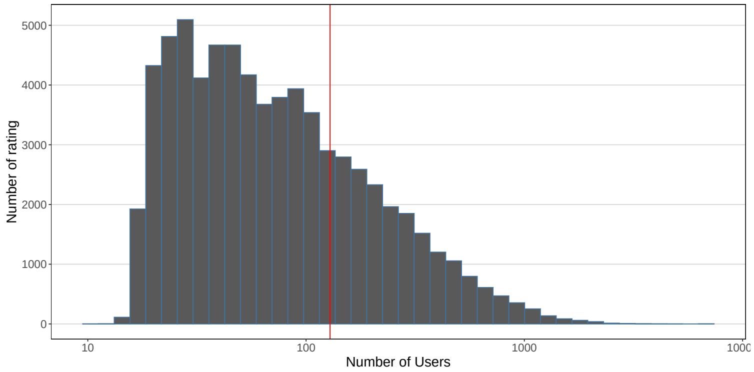


Figure 4: *Number of ratings by users in the edx dataset*

On the other hand, the following histogram represents users distribution by average rating.

1.3.4 Investigation of the effect of Time on the average rating

Time is recorded as the UNIX timestamp, which is just several seconds between a certain date and the Unix Epoch. This count begins on January 1st, 1970, at UTC, with the Unix Epoch.

There is some evidence about the time effect on rating average, but this effect is not strong. The figure below shows the average movie ratings by month

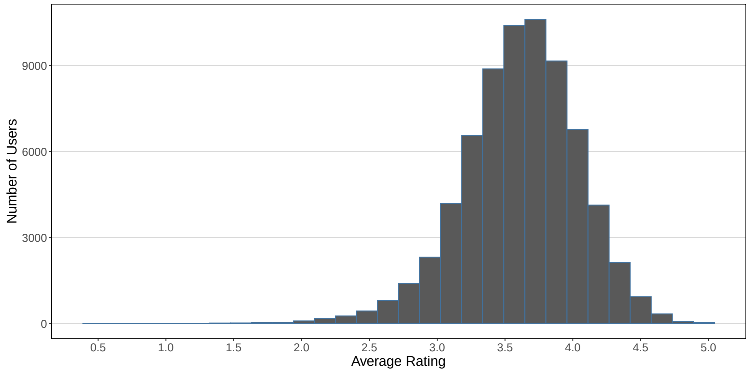


Figure 5: *Users distribution by average rating in the edx dataset*

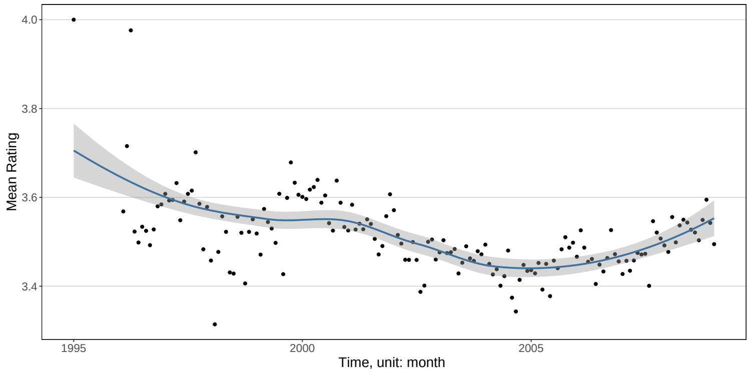


Figure 6: *Average ratings by time/month in Edx Dataset*

1.3.5 Genres Exploration:

A movie could be classified into one or more genres; there are 20 levels of genre.

- The number of ratings varies per genre.
- The rating average for genres are Converging, although the number of ratings varies.
- The genres only slightly affect movie ratings.

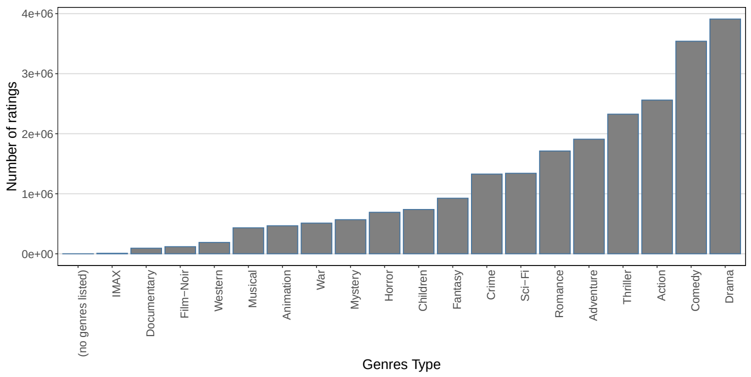


Figure 7: *Number of ratings by genre in the edx dataset*

Figure 7 illustrates the number of movies per genre. On the other hand, we can see the average of ratings per genre as shown in figure 8.

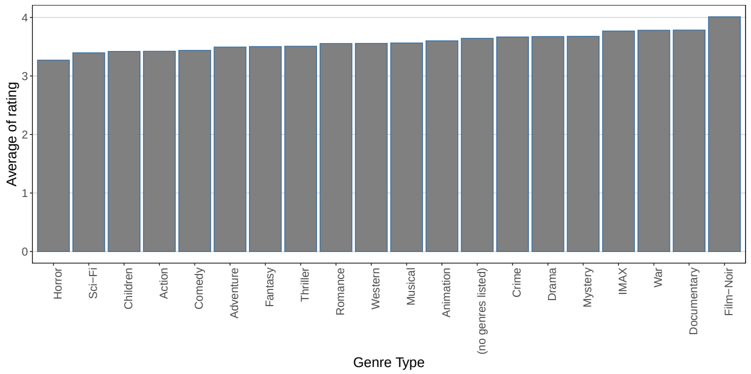


Figure 8: *Distribution of rating average per genre in the edx dataset*

We can see the relationship between numbers of ratings and rating average as shown in figure9.

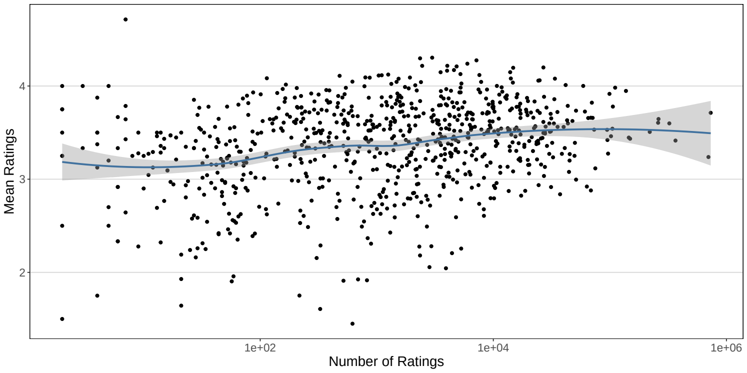


Figure 9: *Relation between Number of Rating vs Mean Rating for Genre*

2. Analysis and modelling approach

We'll discuss the modeling technique we used to build our models in this section, as well as the metric we used to evaluate model performance.

2.1 Model performance evaluation

We'll use root mean squared error (RMSE) as our loss function to compare the performance of our models. The root mean square error (RMSE) is the difference between the anticipated ratings generated from the model and the actual ratings in the test set.
$y_{u,i}$ is the actual rating supplied by user i for movie u, $\hat{y}_{u,i}$ is the projected rating for the same, and N is the total number of user/movie pairings in the formula below.

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

2.2 Constructing and developing baseline Model

Genres and Time do not contribute much information to the data, according to the analysis mentioned above. As a result, I'll examine the movie impact and the user effect to start building a baseline prediction model.

Applying the same rating to all movies is the easiest for forecasting ratings. The real user u rating for movie i, $Y_{u,i}$, is the sum of this "true" rating, μ, and the independent errors sampled for the same distribution, $\epsilon_{u,i}$.

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

2.3 Movie effects

Building on this effect will therefore increase the precision of the prediction as it is known that some movies are normally higher than others. This means perhaps a further enhancement to our model by taking the effect of the film on the rating of b_i into account.

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

The $\hat{b}_i$, the lowest film estimate, may come from an average of $Y_{u,i} - \hat{\mu}$ for every i movie and, consequently, the following format was utilized in order to take movie effects into account

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

2.4 Movie and user effects

Some users are more active in rating films than others hence the algorithm was further refined for the purpose of adjusting for the user impacts (b_u).

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

The least square estimates of the user effect, $\hat{b}_u$, were obtained using the following formulas instead of fitting linear regression models.

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

2.5 Regularization

Regularization allows for high estimates from small sample sizes to be imposed. The Bayesian method, which shrank predictions, has certain commonalities. The overall concept is to apply a penalty to the square equation we reduce for big b_i values. Therefore it is more difficult to reduce with numerous big b_i or b_u.

By addressing the less square problem, a more accurate calculation of bu and bi will treat them symmetrically

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

If $\frac{1}{N}\sum_{u,i}\left(y_{u,i} - \mu - b_i - b_u\right)^2$, tries to discover the ratings of both b_u's and b_i's. The $\lambda\left(\sum_i b_i^2 + \sum_u b_u^2\right)$ regulation term prevents overfitting by penalizing parameter magnitudes. The method of the stochastic gradient descent, which is an item of the matrix recommender engine, may handle this least-square issue very rapidly.

We utilized cross-validation to get the optimal λ, and we can prove that with calculus the b_i or b_u variables that minimize this equation are

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

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

2.6 Matrix factorization

In recommender systems, matrix factorization is a type of collaborative filtering technique. The user-item interaction matrix is decomposed into the product of two smaller dimensionality rectangular matrices by matric factorization procedures. Because of its success, this family of algorithms became well-known during the Netflix prize challenge, as recounted by Simon Funk in a 2006 blog post, in which he shared his findings with the scientific community.

We'll use Matrix Factorization in conjunction with stochastic gradient descent in parallel. It develops a Recommender System by Using Parallel Matrix Factorization with the help of the "recosystem" package, which is a R wrapper of the LIBMF library. The main task of the recommender system i.e. recosystem is to predict unknown entries in the rating matrix based on observed values.

3.Result

The primary goal is to create a recommendation system that will minimize the RMSE to less than 0.86490. Because the validation dataset was held aside for the final hold-out test, edx has been divided into two datasets: edx train (80%) and edx test (20%). edx train is used to build a variety of models, while edx test is used to evaluate their performance.

3.1 Model 1: BaseLine Model

It's just a model that ignores all the feathers and calculates the average rating. This model serves as a baseline against which we will aim to improve RMSE.
The RMSE is 1.0599 and the average rating is μ= 3.5125.

Model	RMSE
Just the Average	**1.059904**

3.2 Model 2: Movie effects Model

We'll add the bias of movies b_i for each movie to the model because the features of a film can influence its ratings. The average rating for that particular film will differ from the general average rating for all films. We'll determine the RMSE of this model by plotting the movie bias distribution.

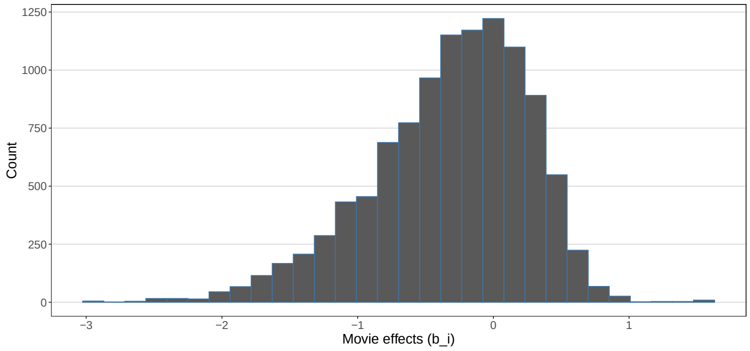


Figure 10: Distribution of movie effect (b_i) edx_train dataset

Figure 10 depicts how the estimate of movie effect $\hat{b}_i$ differs significantly across all of the movies in the training dataset. Including the movie effect in the algorithm increased the model's accuracy by 10.96%, producing an RMSE of 0.9437, which is still higher than the target.

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

3.3 Model 3: Movies and Users effects Model

In a manner analogous to the movie effect, a user's characteristics can influence a movie's rating. A user's overall rating for all movies watched could be lower than what other users have rated.
The user bias b_u will be added to the movie effect model, and the RMSE will be calculated.

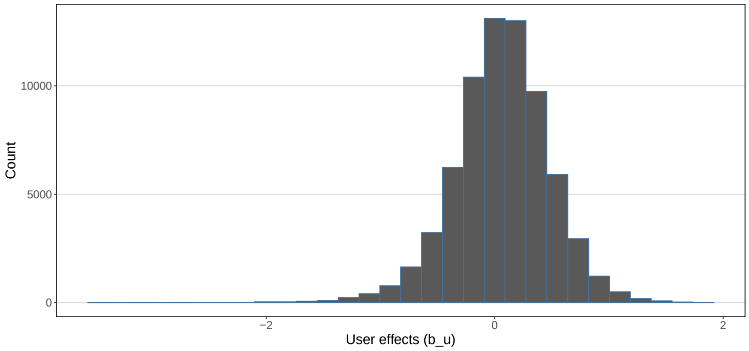


Figure 11: Distribution of movie effect (b_u) in edx_train dataset

The predicted effect of user $\hat{b}_u$ building on the movie effects model is shown in Figure 11. It was clear that correcting for user effects improved the algorithm's accuracy. Adjusting for both the movie and user impacts reduces the RMSE by 18.30% when compared to the baseline model.

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

3.4 Model 4: Regularization of movie and user effects

The approach of regularization was being applied with regard to movie and user effects by applying a higher penalty to smaller sample estimations. So we're going to employ λ parameter.

During our data analysis, we discovered that some people are more engaged in movie reviews than others. Some people have only rated a few films. Some films, on the other hand, have received extremely few ratings. We should not put our faith in these erratic estimates. Furthermore, RMSE is susceptible to huge mistakes. As a result, we must include a penalty word to devalue such an effect.

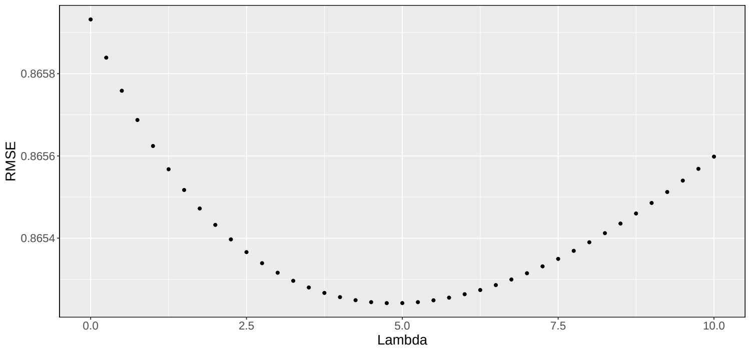


Figure 12: Selecting the tuning parameter in edx_train dataset

The RMSE delivered across each of the lambda values evaluated is shown in the graph above. The ideal value for *lambda* was 4.75, which reduced the RMSE to 0.8652, which was just enough to beat the project's target RMSE. This resulted in a total improvement in baseline accuracy of 18.37%.

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**

```
## iter      tr_rmse          obj
##    0       0.8600   5.5621e+06
##    1       0.8380   5.1814e+06
##    2       0.8212   5.0401e+06
##    3       0.8043   4.9102e+06
##    4       0.7888   4.7920e+06
##    5       0.7752   4.6920e+06
##    6       0.7634   4.6111e+06
##    7       0.7530   4.5393e+06
##    8       0.7439   4.4804e+06
##    9       0.7360   4.4281e+06
##   10       0.7290   4.3836e+06
##   11       0.7228   4.3454e+06
##   12       0.7173   4.3111e+06
##   13       0.7124   4.2809e+06
##   14       0.7081   4.2554e+06
##   15       0.7042   4.2324e+06
##   16       0.7006   4.2107e+06
##   17       0.6974   4.1921e+06
##   18       0.6945   4.1749e+06
##   19       0.6919   4.1602e+06
```

```
## prediction output generated at C:\Users\mhabbash\AppData\Local\Temp\RtmpoXGxQ5\file10b478c7683f
```

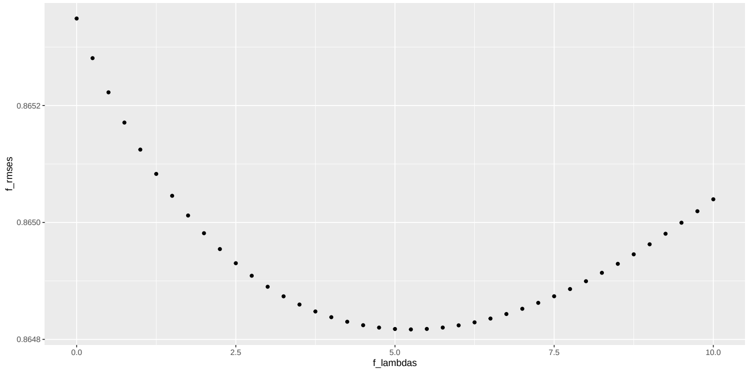
3.5 Model 5: Matrix Factorization

The following results may be achieved by following the necessary process for recosystem library provided on the site. We get RMSE equal to 0.7965 by applying the Matrix factorization method. By calculating the

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**

decrease in the percentage of RMSE we can observe a drop of more than 24.85% % in the matrix factorization technique in comparison with the basic model.

```
## [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\mhabbash\AppData\Local\Temp\RtmpoXGxQ5\file10b415d35315
```

3.6 Final Model

The above sections create models that are trained using edx_test, a subset of the edx dataset. This results in that Matrix factorization with an RMSE of 0.7965 is the top-ranked model. The final model, therefore, follows the Matrix factorization approach. The ultimate model is based on "edx_dataset" and evaluated using the "validation" dataset. RMSE of 0.7867 which is an improvement of 25.78% compared to a baseline model was achieved by the last hold-out test in the validation dataset. As may be seen in the following table.

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

4. Conclusion

This study goes through a few approaches for building recommendation systems with a residual mean square error of less than 0.86490. When the model trained on edx and evaluated on validation, the highest performing model is matrix factorization, which has an RMSE of 0.7867.This improves the RMSE's model by around 25.78% on the first model. While this algorithm built here fulfilled the goal of the study, there is still a significant error loss suggesting that the accuracy of the recommendation systems may still be improved. Finally, a particularly strong method for the recommendation systems seems to be matrix factorization. Those interested in an enhanced way to constructing systems that suggest the recosystem package should consult. It helps with the development of matrix factorisation in huge quantities of data.

5. References

[1] “Introduction to Data Science - Data Analysis and Prediction Algorithms with R”, Dr. Rafael A. Irizarry link

[2] “R Markdown: The Definitive Guide”, Yihui Xie, J. J. Allaire, Garrett Grolemund, 2019-06-03 link

[3] “Recommender System Using Parallel Matrix Factorization”, Yixuan Qiu, 2021-01-09. link

[4] “E. Winning the Netflix Prize: A Summary”, Chen, 2020/10/15. link

6.Appendices

Appendix A:

A.1 MovieLens Project Instructions

The submission for the MovieLens project will be three files: a report in the form of an Rmd file, a report in the form of a PDF document knit from your Rmd file, and an R script that generates your predicted movie ratings and calculates RMSE. The R script should contain all of the code and comments for your project. Your grade for the project will be based on two factors:

1.Your report and script (75%)
2.The RMSE returned by testing your algorithm on the validation set (the final hold-out test set) (25%)

Note that to receive full marks on this project, you may not simply copy code from other courses in the course series and be done with your analysis. Your work on this project needs to build on code that is already provided. Please note that once you submit your project, you will not be able to make changes to your submission.

A.1.1 Report and Script (75%)

Your report and script will be graded by your peers, based on a rubric defined by the course staff. Each submission will be graded by three peers and the median grade will be awarded. To receive your grade, you must review and grade the submissions of five of your fellow learners after submitting your own. This will give you the chance to learn from your peers.

Please pay attention to the due dates listed! The project submission is due before the end of the course to allow time for peer grading. Also note that you must grade the reports of your peers by the course close date in order to receive your grade.

A.1.2 RMSE (25%)

Your movie rating predictions will be compared to the true ratings in the validation set (the final hold-out test set) using RMSE. Be sure that your report includes the RMSE and that your R script outputs the RMSE.

Note that to receive full marks on this project, you may not simply copy code from other courses in the course series and be done with your analysis. Your work on this project needs to build on code that is already provided.

IMPORTANT: Make sure you do NOT use the validation set (the final hold-out test set) to train your algorithm. The the final hold-out test set should ONLY be used to test your final algorithm. The final hold-out test set should only be used at the end of your project with your final model. It may not be used to test the RMSE of multiple models during model development. You should split the edx data into a training and test set or use cross-validation.

A.1.3 Honor Code

You are welcome to discuss your project with others, but all submitted work must be your own. Your participation in this course is governed by the terms of the edX Honor Code. If your project is found to violate the terms of the honor code, you will receive a zero on the project, may be unenrolled from the course, and will be ineligible for a certificate.

A.1.4 Project Due Date

Submissions for the Movielens project are due one week before course close, on March 9, 2022, at 23:59 UTC. This allows time for peer grading to occur! Peer grades are due at course close, on March 16, 2022, at 23:59 UTC.

A.1.5 Peer Feedback

You are strongly encouraged to give your peers thoughtful, specific written feedback in addition to the numerical grades in the rubic. Think about the type of feedback that would help you improve your work and offer that type of feedback to your fellow learners.

If you feel your report was not fairly graded by your peers, you may report it in the discussion forum to ask for staff review of the report.

Appendix B:

B.1 MovieLens Project Submission

B.2.1 Your Response

due Jul 29, 2021 02:59 EEST (in 2 weeks, 3 days)

Enter your response to the prompt. You can save your progress and return to complete your response at any time before the due date (Thursday, Jul 29, 2021 02:59 EEST). After you submit your response, you cannot edit it.

B.2.2 The prompt for this section

Your submission for this project is three files:

1.Your report in Rmd format
2.Your report in PDF format (knit from your Rmd file)
3.A script in R format that generates your predicted movie ratings and RMSE score (should contain all code and comments for your project)

You may upload the three files directly to the edX platform or submit a GitHub link in the text response box below.

To upload and submit your files press the "Choose Files" button, select three files at once (using the control key on a Windows machine or command key on a Mac) and press "Choose," type a description for each (PDF, Rmd, R), and then press the "Upload files" button. If uploading files, we recommend also providing a link to a GitHub repository containing the three files above in case there is a problem with the upload process.

Note that when downloading files for peer assessments, R and Rmd files will be downloaded as txt files by default.

B.2.3 MovieLens Grading Rubric

The following is the grading rubric your peers will be using to evaluate your project. There are also opportunities for your peers to provide written feedback as well (required for some categories and optional for others). You are encouraged to give thoughtful, specific written feedback to your peers whenever possible (i.e., more than just "good job" or "not enough detail").

Note that to receive full marks on this project, you may not simply copy code from other courses in the course series and be done with your analysis. Your work on this project needs to build on code that is already provided.

After you submit your project, please check immediately after submitting to make sure that all files were correctly uploaded. Occasionally, there are file upload failures, and it's easiest to fix if these are caught early.

B.2.3.a Files (10 points possible)

The appropriate files are submitted in the correct formats: a report in both PDF and Rmd format and an R script in R format.

- 0 points: No files provided AND/OR the files provided appear to violate the edX Honor Code.
- 3 points: Multiple requested files are missing and/or not in the correct formats.
- 5 points: One file is missing and/or not in the correct format.
- 10 points: All 3 files were submitted in the requested formats.

B.2.3.b Report (40 points possible)

The report documents the analysis and presents the findings, along with supporting statistics and figures. The report must be written in English and uploaded. The report should be written assuming that the reader is not familiar with the project or the data. The report must include the RMSE generated. The report must include at least the following sections:
1. an introduction/overview/executive summary section that describes the dataset and summarizes the goal of the project and key steps that were performed.
2. a methods/analysis section that explains the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and your modeling approach.
3. a results section that presents the modeling results and discusses the model performance.
4. a conclusion section that gives a brief summary of the report, its limitations and future work.

- 0 points: The report is either not uploaded or contains very minimal information AND/OR the report appears to violate the edX Honor Code.

- 10 points: Multiple required sections of the report are missing.
- 15 points: The methods/analysis or the results section of the report is missing or missing significant supporting details. Other sections of the report are present.

- 20 points: The introduction/overview or the conclusion section of the report is missing, not well-presented or not consistent with the content.

- 20 points: The report includes all required sections, but the report is significantly difficult to follow or missing supporting detail in multiple sections.

- 25 points: The report includes all required sections, but the report is difficult to follow or missing supporting detail in one section.

- 30 points: The report includes all required sections and is well-drafted and easy to follow, but with minor flaws in multiple sections.

- 35 points: The report includes all required sections and is easy to follow, but with minor flaws in one section.

- 40 points: The report includes all required sections, is easy to follow with good supporting detail throughout, and is insightful and innovative.

B.2.3.c Code (25 points)

The code in the R script should should be well-commented and easy to follow. The code provided in the R script should contain all of the code and comments for your project. You are not required to run the code provided (although you may if you wish), but you should visually inspect it.

- 0 points: No code provided AND/OR the code appears to violate the edX Honor Code.

- 10 points: Code appears that it would not run/is very difficult to follow or interpret AND/OR is not consistent with the report.

- 15 points: Code appears that it would run without throwing errors, can be followed, is at least mostly consistent with the report, but has no comments or explanation.

- 15 points: Code is simply a copy of code provided in previous courses in the series without expanding on it, but is otherwise well-commented.

- 20 points: Code appears that it would run without throwing errors, can be followed, is largely consistent with the report, but without sufficient comments or explanations.

- 25 points: Code is easy to follow, is consistent with the report, and is well-commented.

B.2.4 RMSE (25 points)

Provide the appropriate score given the reported RMSE. Please be sure not to use the validation set (the final hold-out test set) for training or regularization - you should create an additional partition of training and test sets from the provided edx dataset to experiment with multiple parameters or use cross-validation.

- 0 points: No RMSE reported AND/OR code used to generate the RMSE appears to violate the edX Honor Code.
- 5 points: RMSE >= 0.90000 AND/OR the reported RMSE is the result of overtraining (validation set - the final hold-out test set - ratings used for anything except reporting the final RMSE value) AND/OR the reported RMSE is the result of simply copying and running code provided in previous courses in the series.

- 10 points: 0.86550 <= RMSE <= 0.89999

- 15 points: 0.86500 <= RMSE <= 0.86549

- 20 points: 0.86490 <= RMSE <= 0.86499

- 25 points: RMSE < 0.86490