MovieLens Recommendation System

Harry

08 August, 2022

Contents

1 Introduction

Machine learning (ML) may be a sort of artificial intelligence (AI) that permits program applications to gotten to be more precise at foreseeing results without being expressly modified to do so. Machine learning calculations utilize historical information as input to foresee modern yield values. Recommendation system are a common utilize case for machine learning. Other well known employments incorporate extortion discovery, spam sifting, malware danger discovery, commerce handle robotization (BPA) and Prescient support.

Recommendation engines are application of of machine learning that typically relate with ranking or rating product / users. Loosely outlined, a recommender system may be a system that predicts ratings a user may offer to a selected item. These predictions can then be hierarchal and came back to the user.these system works on user historical interest and preferences and suggest them those producte related to their previous search.They're used by various large name companies like Google, Instagram, Spotify, Amazon, Reddit, Netflix etc. often to increase engagement with users and the platform. The MovieLens datasets have provided a popular environment for experimentation with machine learning.It contains 10 million ratings on 10,000 movies by 72,000 users. It was released in 2009.it contains the movie rating given by users on of multiple genres in the range of 0-5.

For this project our goal is to make a moving picture recommendation system using MovieLens dataset by applying machine learning algorithms and other tecniques. This dataset could be a tiny set of a far larger dataset with voluminous ratings. we have been instructed to use given code to form 2 datasets: (1) edx and (2) validation. The edx dataset is that the dataset during which we are going to train and check our models. The validation knowledge set is our 'unseen' data and can be accustomed report our final root mean square value (RMSE). The goal of this project is to achieve the RMSE price less thn 0.8649

This project implement various machine learning tehniques. First we are going to perform,data wrangling then descriptive statistis and exploratory data analysis using the movielens dataset using various visualization that contains information like users, movies, genres, and their ratings. Once information exploration is finished, we are going to preprocess our data to extrapolate options that we tend to might need for model creation. With our options in hand, we are going to produce many models and predict movie ratings. Lastly, we are going to opt for the most effective model to run against test data and report the final RMSE value.

Table 1: edx dataset: variable class and first 5 rows

userId	movieId	rating	timestamp	title	genres
numeric	numeric	numeric	numeric	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	:Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

1.1 Ratings ($rating)

The overall mean rating in the training subset was 3.51.the minimum mean rating is 0.5 and the maximum rating calculated was 5. The overall distribtuion of ratings in the dataset (Figure 1) shows that the mode of ratings across all movies was 4, and that, overall, discrete star ratings (7,156,885; 79.5%) were given more than half star ratings (1,843,170; 20.5%).

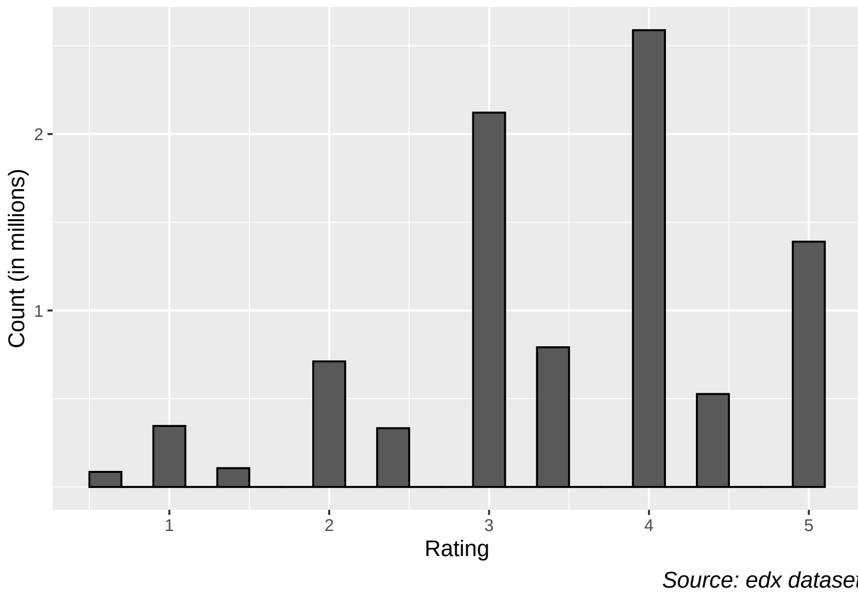

Figure 1: Overall ratings distribution

1.2 Movies ($movieId)

Undoubtedly there are some movies which are highly rated than others (see Figure 2). moreover Further analysis shows considerable variation in the number of ratings achieved by each movie (see Figure 3), with the movie with the highest number of ratings, Star Wars, achieving a total of 95038 ratings whereas as many as 126 movies were only rated once. There is noticeably a movie effect on the rating awarded and hence this effect is included in our training model to check for bias caused by this effect .

1.3 Users ($userId)

The user data statistics shows similar trend as of movie ratings with few users appearing more kind towards movies by giving higher ratings than others (see Figure 4). Some users contributed higher number of ratings than other users (Figure 5). For example, a single user gave significant total of 6616 ratings whereas as many as 1059 provided less than 10 movie ratings each.this data exploration clearly indicate a user bias effect which must be taken into account while implementing recommendation model.

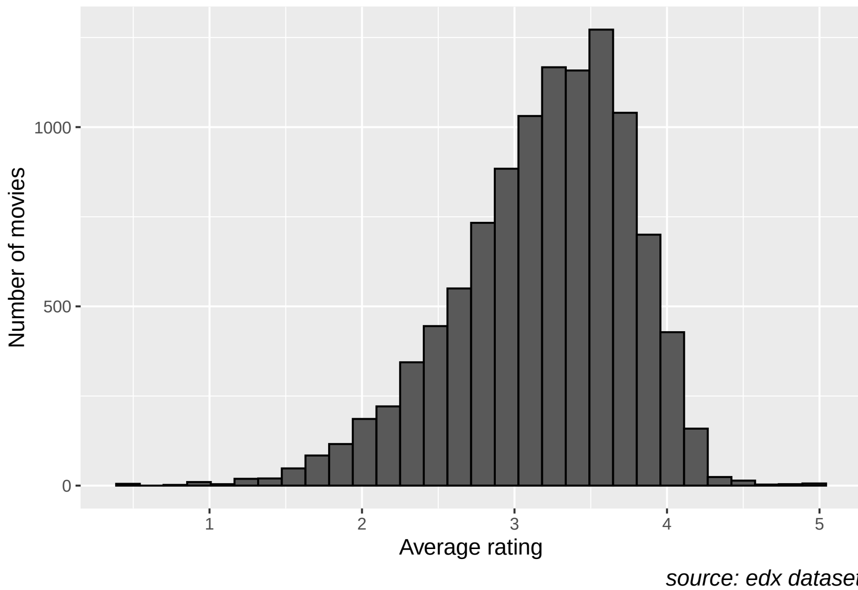

Figure 2: Movie distribution by average rating

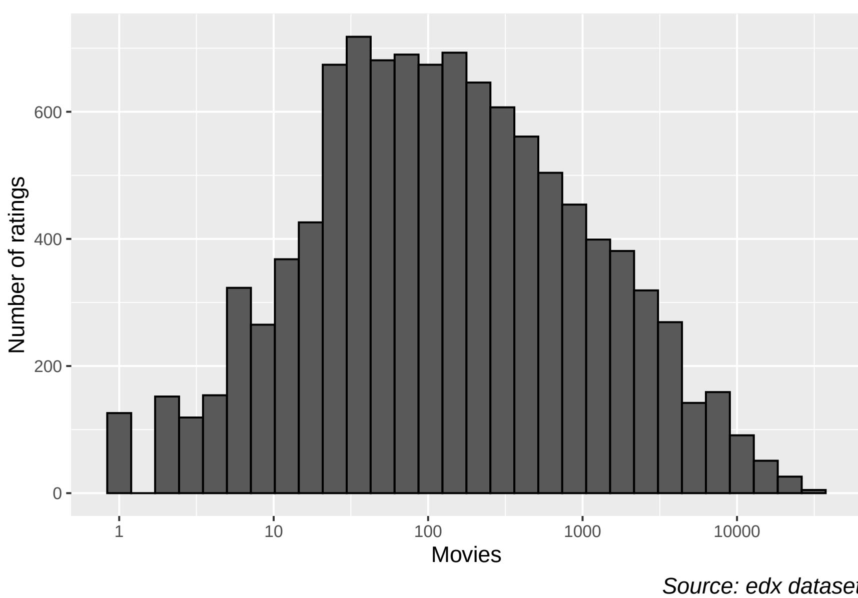

Figure 3: Number of ratings by movie

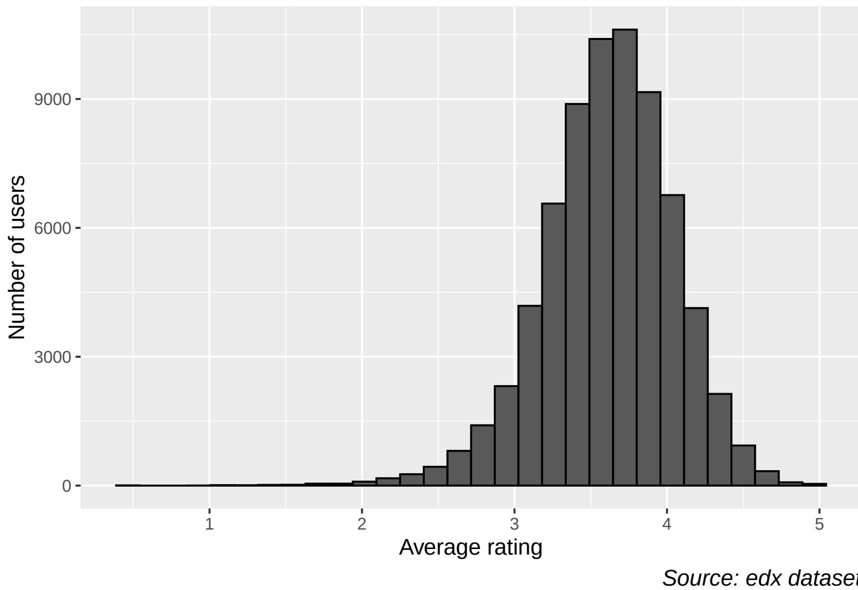

Figure 4: User distribution by average rating

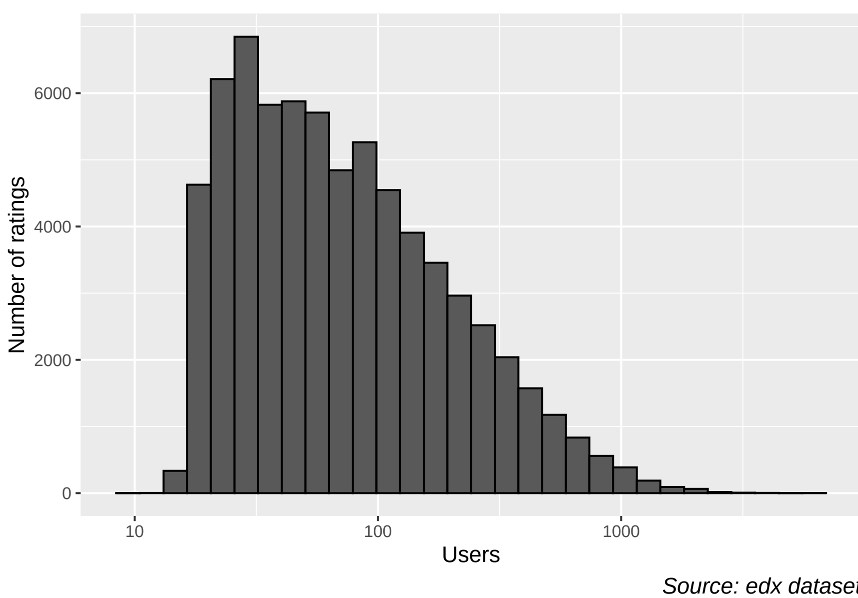

Figure 5: Number of ratings by user

1.4 Movie Genre ($genres)

As was shown in Table 1 above,the genre columns tell us the different categories related to the specified movie. For example, Net, The (1995) in the first row of the edx dataset includes Action|Crime|Thriller in the genre variable. Some movies has multiple related genre and there were a total of 971 unique genre combinations recorded in the dataset. Differentiating these combinations into rows with single genres, it was possible to identify 58 different genre categories (including "no genre listed") and to rank these by the number of ratings (Table 2).

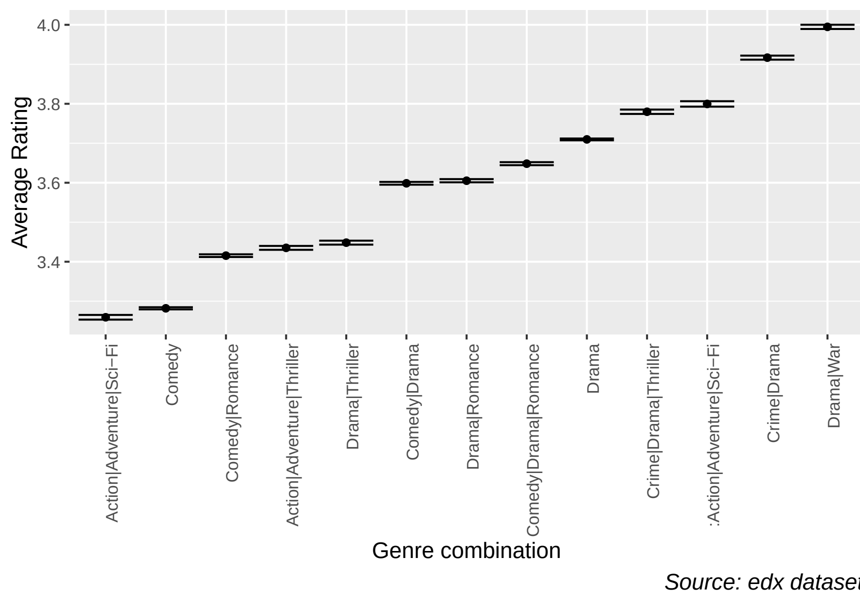

Figure 6: Average rating by genre

Movies related with drama and comedy genres had the most ratings whereas Documentary and IMAX movies had the low number of ratings. Seven ratings were provided for movies for which no genre was recorded. Table 2 also shows a distribution of mean rating by genre.To avoid the complexity of analysis the data was grouped by the movies by unique genre combinations and then filtering only those combinations with minimum 100,000 ratings , shows a noticeable effect of genre with 'Comedy' movies recorded as the lowest mean rating whereas 'Crime|Drama' and 'Drama|War' movies recorded the highest mean rating (Figure 6). hence,this significant effect is included in training the algorithm for the movie recommendation system.

1.5 Movie Title ($title)

The title variable shows movies title along with their year of release. Table 3 shows the top 10 movie titles by the number of ratings.

In order to check the effect, if any, of the year of release information on mean rating, the title column was split into two separate columns, one for the title and the other for the year of release. The new column year of releasr was then used to check the effect of release year on mean rating and, as can be seen in Figure 7, mean rating varied by year of release. furthermore, the mean rating was highest for movies released between 1940 and 1950 and lowest for movies released since that period.

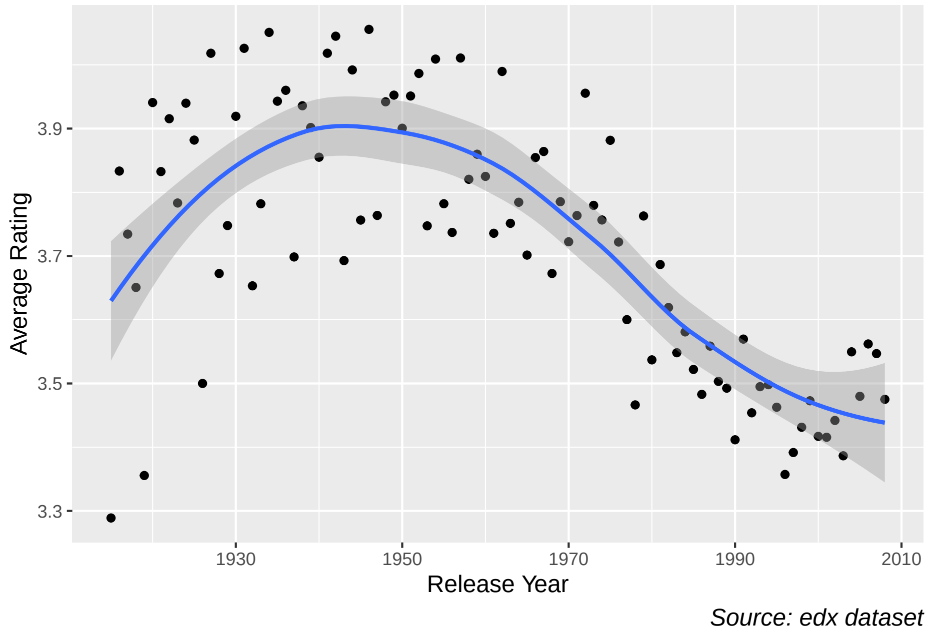

Figure 7: Average rating by year of release

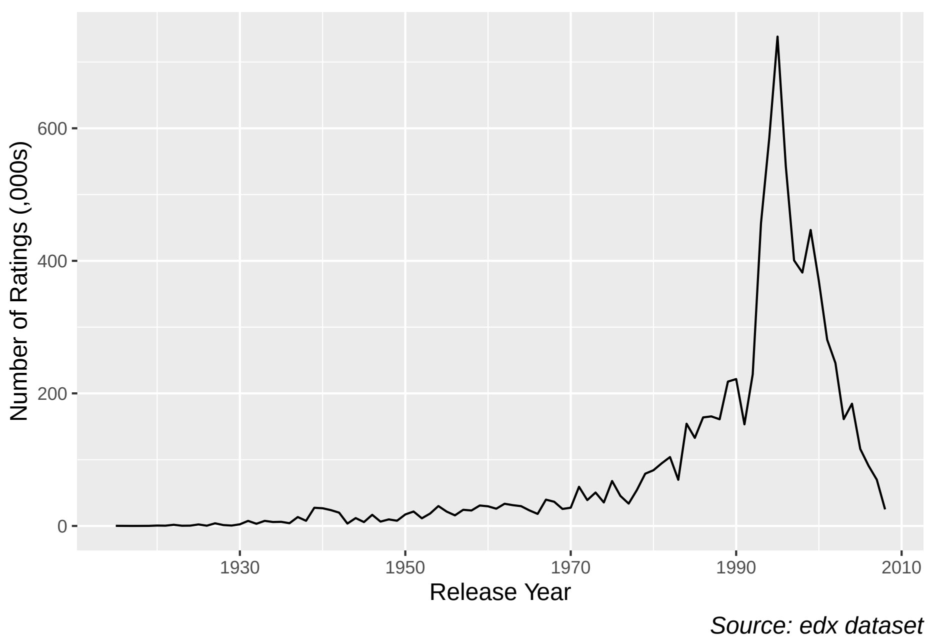

Figure 8: Number of ratings by year of release

However, as shown in Figure 8, the number of ratings was less within the data assigned to movies released before 1970. The movies with the mode number of ratings were released during the 1990s, peaking in 1995 with approximately 8% of the total number of ratings recorded in the edx data. hence year of release can also be taken into account as the effect of release year should improve the accuracy of the training model

1.6 Date of review ($timestamp)

A timestamp is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day. To ease the analysis of the effect of review date on ratings, the timestamp data was converted into date format, dropping time data and rounding to the nearest week..

The oldest review included in the dataset was completed in 1995. This was also when the mean rating was highest, with a slight decline in ratings observed until around 2005 after which mean ratings began to increase again. The effect of review date on the mean rating was moderate compared to that observed for movies and users but there was still some changes over time (see Figure 9), justifying its importance to be included in the implementation of the recommendation model.

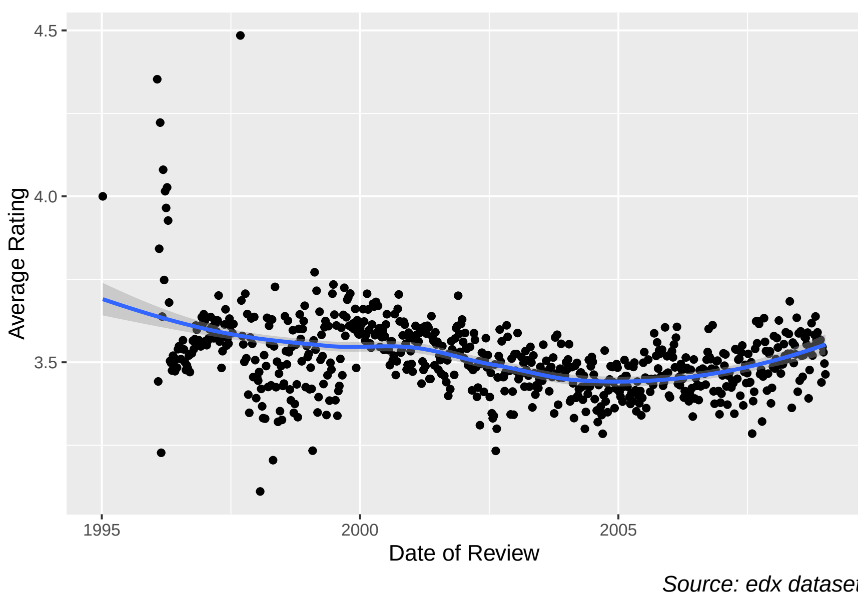

Figure 9: Average rating by date of review

2 Methods

2.1 Splitting edx out into train and test sets

As the validation dataset was instructed to be only used for the final hold-out test, the edx dataset needed to be used both to train and test the recommendation model in development. This is important to allow for cross-validation and improvment of the final model avoiding risk of over-fitting.

Here, the same technique was applied as with the original movielens dataset, using the caret library we used 'createDataPartition' to divide the edx dataset into training (80%) and testing (20%) sets. As before, the dplyr functions 'semi_join' and 'anti_join' were used, firstly to ensure that the test set only included users and movies variables that are present in the training set and, secondly to add the excluded data to the training set in order to maximise the data available for training purposes.

2.2 Calculating the error loss

The root-mean-square deviation or root-mean-square error is a mostly used measure of the differences between predicted values by a model or an estimator and the original values recorded. The RMSE was calculated to check for the error loss between the predicted movie ratings calcuated after training the algorithm and actual recorded ratings in the test set. In the formula shown below, $y_{u,i}$ is defined as the actual rating provided by user i for movie u, $\hat{y}_{u,i}$ is the predicted rating for the same movies after training the model, and N is the total number of user over movie combinations.

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

The objective of the project was to develop a model that achieved an RMSE value less than 0.86490 as set out below. A simple table was designed to record the project objective as well as the results obtained during development within the edx dataset and in the final hold-out test in the validation dataset (see Section 4: Results).

2.3 Developing the algorithm

The easiest algorithm for predicting ratings is to set the same rating to all movies. Here, the actual rating for movie i by user u, $Y_{u,i}$, is the sum of this "true" rating, μ, plus $\epsilon_{u,i}$, the independent errors sampled for the same distribution.

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

The mean of all ratings is the estimate of μ that lowers the RMSE. Thus, $\hat{\mu}$ = mean(train_set$rating) was the simple formula used to train the intial model.

The exploratory data analysis explained in the previous section showed that rating distibution across all movies was uneven. That is, few movies got a higher mean rating than others and, and taking this effect into consideration will therefore improve the accuracy of the prediction. Thus, the training model was further improvised by taking into account the effect of movie on rating, b_i.

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

A linear regression model would take some time to run given the large dataset involved. Instead, the least squares estimate of the movie effects, $\hat{b}_i$, can be derived from the average of $Y_{u,i} - \hat{\mu}$ for each movie i and, thus, the following formula was used to take account of movie effects within the training algorithm.

$$\hat{y}_{u,i} = \hat{\mu} + \hat{b}_i$$

The exploratory data analysis also showed that distribution of movies rating by user is also uneven so further improvisation were made to the model to adjust for user effects (b_u). As previously, rather than fitting linear regression models, the least square estimates of the user effect, $\hat{b}_u$ was calculated using the formulas shown below.

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

Genre has also an effect on movie ratings, with some genres achieving higher mean ratings than others. This effect was observed even when movies were allocated to multiple genres, as in the original dataset. Therefore, the each movie rating and user was further improvised by adjusting for genre effect, b_g, and the least squares estimate of the genre effect, $\hat{b}_g$ calculated using the formula shown below.

$$Y_{u,i} = \mu + b_i + b_u + b_g + \epsilon_{u,i}$$
$$\hat{b}_g = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u\right)$$

The fourth bias to adjust for within the model was the movie release year varaible. The exploratory analysis in the previous section showed an effect of the release year, b_y,on movie rating and the least squares estimate of the year effect, $\hat{b}_y$ calculated using the formula shown below, building on the algorithm developed already.

$$Y_{u,i} = \mu + b_i + b_u + b_g + b_y + \epsilon_{u,i}$$
$$\hat{b}_y = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u - \hat{b}_g\right)$$

There was also a small effect of the time stamp of review (b_r)on the mean rating acheived by each movie and user. The most appropriate way to take this effect into the model would be to use a smooth function to the day of release for each rating by movie and user. Rounding the date of review to the nearest week provided significant smoothness of the data. The least squares estimate, taking into account the date of review effect, $\hat{b}_r$ was calculated using the formula shown below.

$$Y_{u,i} = \mu + b_i + b_u + b_g + b_y + b_r + \epsilon_{u,i}$$
$$\hat{b}_r = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u - \hat{b}_g - \hat{b}_y\right)$$

2.4 Regularising the algorithm

Finally, the exploratory data analysis showed that not only is the mean rating affected by the movie, user, genre, year of release and date of review, but that the number of ratings also cahnges. Thus, for example, some movies and genres of movie received less number of ratings than others while some users provided less number of ratings than others. Similarly, the number of ratings also changes by year of release and date of review. In each of these cases, the aftermath of these cahnges is that the estimates of the effect (b) will have been subject to greater error when based on a smaller number of ratings.

Regularisation is an effective method for penalising large effect estimates that are based on small sample sizes [@irizarry_2020]. The penalty term, λ, is a tuning parameter chosen using cross-validation within the edx dataset. Thus, the movie effect, b_i can be regularised to penalise these large effects as follows.

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

Based on the above, the least squares estimate for the regularised effect of movies can be calculated as below, where n_i is the number of ratings made for movie i. The effect of $\frac{1}{\lambda+n_i}$ is such that when the sample size is large enough, i.e. n_i is a big number, λ has weak effect on the estimate, $\hat{b}_i(\lambda)$. meanwhile, where the sample size is small, i.e. n_i is small, the effect of λ increases and the estimate shrinks towards zero.

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

Here, the regularisation algorithm was developed to take all of the effects previously described into account, as shown below. A range of values for λ (range: 4-6, with increments of 0.1) was applied in order to improve the model to minimise the RMSE value. As before, all tuning was completed within the edx dataset, using the train and test sets, so as to avoid over-fitting the model in the validation set.

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i-b_u-b_g-b_y-b_r\right)^2+\lambda\left(\sum_{i}b_i^2+\sum_{u}b_u^2+\sum_{g}b_g^2+\sum_{y}b_y^2+\sum_{r}b_r^2\right)$$

2.5 Validating the final model

Having improved the model algorithm within the train and test sets created by partitioning edx, the final task of the project was to train the algorithm using the whole edx dataset and then to predict ratings within the validation dataset. Prior to doing this, it was necessary to include the date of review and year of release variables in the validation set using the mutate function using the dplyr package.

The final model, adjusting for all discussed effect biases introduced by movie, user, genre, release year, review date, and collectively regularised using the favourable value for λ, was used to predict ratings in the validation dataset, and to calculate the final validation root mean square value.

3 Results

3.1 Simple average

Predicting the mean rating from the train set (3.51) for every row in the test set resulted in a RMSE of 1.06, which is greater than the project objective. Moreover, an RMSE of 1.06 means that predicted ratings are more than 1 star away from the actual rating, an siginifcant error loss for a movie recommendation system which is unacceptable.

3.2 Adjusting for movie effects

Figure 10 shows that the estimate of movie effect (b_i) changes significantly across all of the movies included in the train set. Adding this effect into the model, in order to adjust for the movie effect, improved the accuracy of the model by 11.02%, yielding an RMSE of 0.94, albeit still well greater than the target.

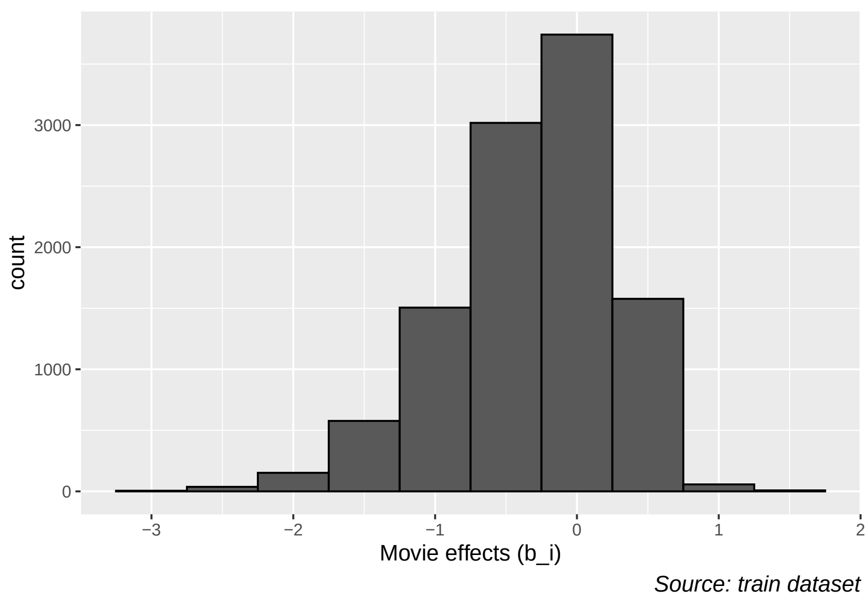

Figure 10: Distribution of movie effects

3.3 Adjusting for user effects

Figure 11 shows the estimated impact of user (b_u) building on the movie effects model above. Whilst b_u showed less variability than was observed with b_i, it was clearly indication that taking user effect into account has enhanced the accuracy of the algorithm. Indeed, adjusting for user effects resulted in an RMSE of 0.86617. Thus, considering both user and movie effects lowered the RMSE value by 18.33% versus the simple model, demonstrating the strong bias producted by each of these variables on movie ratings.

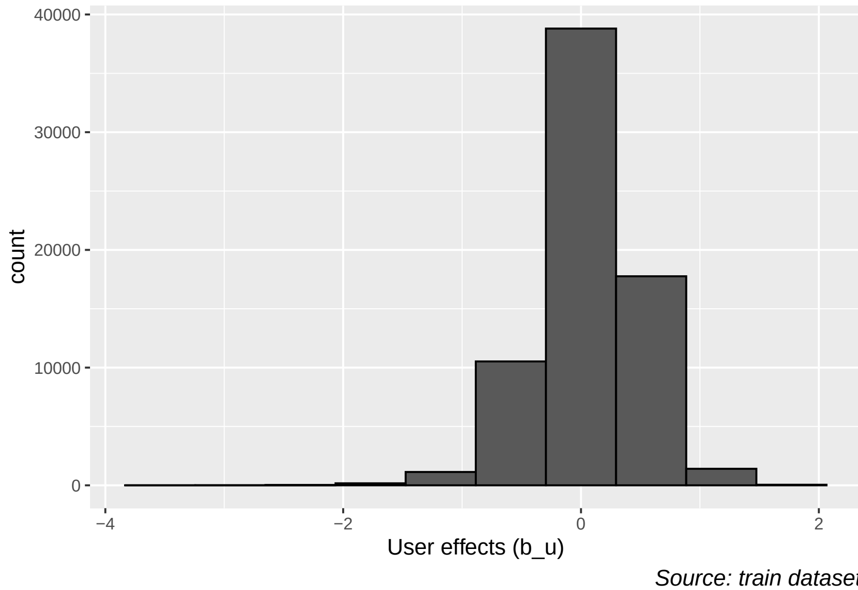

Figure 11: Distribution of user effects

3.4 Adjusting for genre effects

Figure 12 shows the distribution of estimate genre effects, b_g in the train set, once again showing some variation across different genre combinations.

The output from the model when taking genre effect into consideration, in addition to movie and user bias, was an RMSE of 0.8658. Thus including the ffect of genre on rating into the model only provided a moderate improvement in the accuracy of the algorithm, lowering the RMSE by 0.04% versus the previous model and 18.37% versus the original model. This improvement did bring the model very close to meeting the project objective rmse, reducing the difference to only 0.0009.

3.5 Adjusting for release year effects

The year of movie release adds moderate additional changeability to the average rating in the train set as shown in Figure 13. Indeed, taking this effect into account in the training algorithm yielded a modest incremental improvement of 0.02% in the accuracy of ratings prediction bringing the RMSE little closer to meeting the project objective at 0.86566.

3.6 Adjusting for review date effects

lastly the date of review bias effect was taken into account. The exploratory analysis had shown that this had a small efect on ratings and this was confirmed by visualizing the distribution of b_r in Figure 14.

Adding the effect of review date into the algorithm delivered an RMSE of 0.86549, an improvement of 18.39% versus the original model but still not quite enough to meet the project objective.

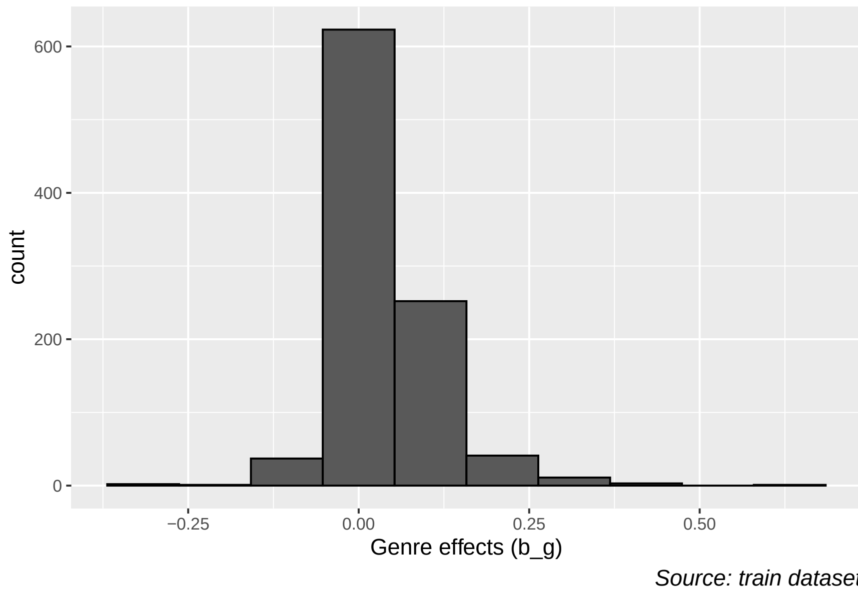

Figure 12: Distribution of genre effects

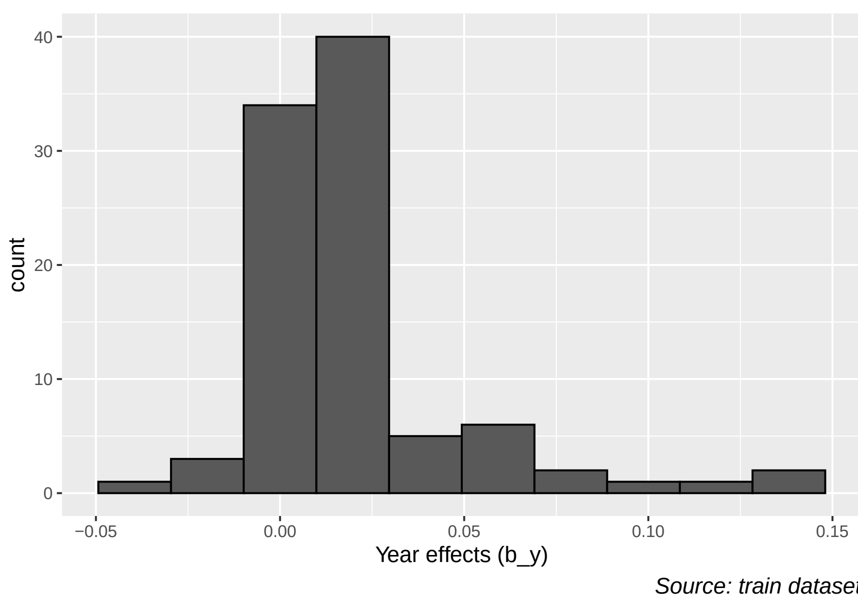

Figure 13: Distribution of release year effects

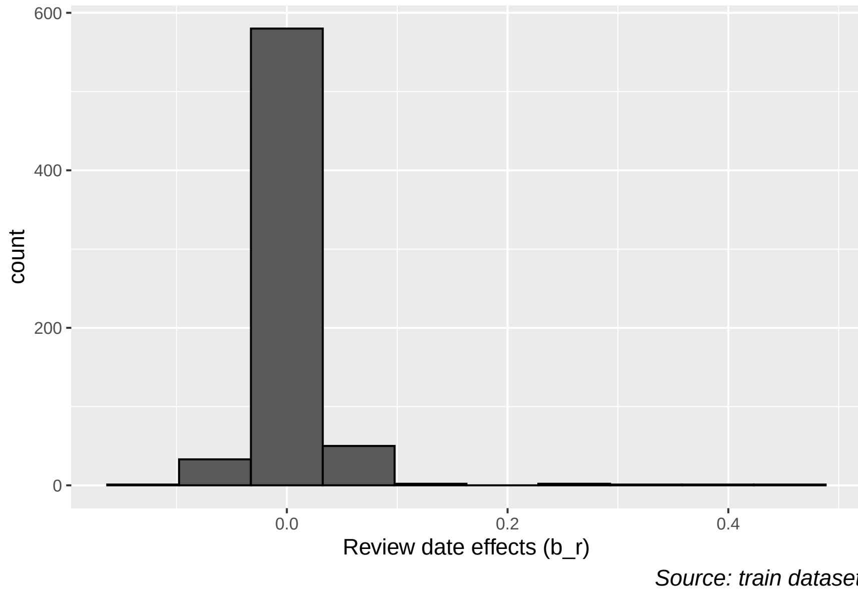

Figure 14: Distribution of review date effects

3.7 Effect of regularisation

The final step in developing the model was to apply regularization. Figure 15 shows the RMSE calculated across each of the values for λ tested. The optimum value for λ was 5.1 which lowered RMSE to 0.86482, which was just enough to surpass the target RMSE set as the project objective. This represented a total improvement of 18.46% in the accuracy of the model by adjusting for movie, user, genre, year of release and review date effects and applying regularization to the combination of these effects.

3.8 Final hold-out test in validation dataset

The final hold-out test in the validation dataset calculated an RMSE of 0.86406, an improvement of 18.53% versus the origina model based on the overall mean rating and 0.00084 below the target RMSE value set as project objective.

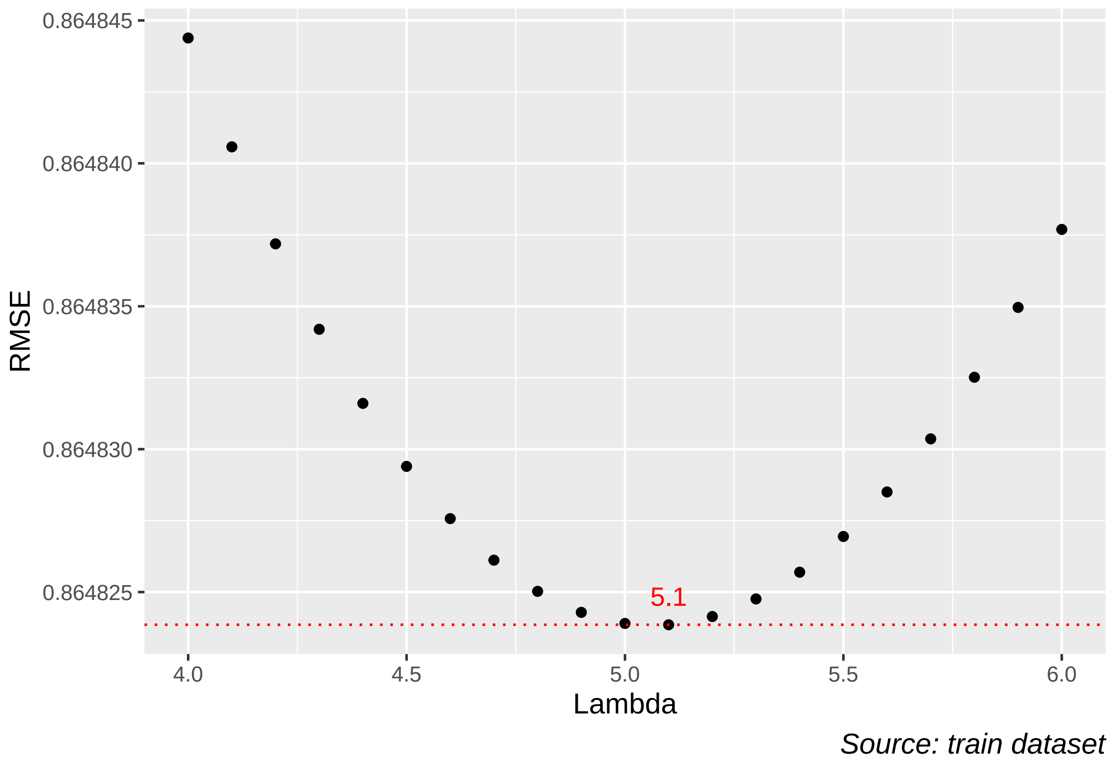

Figure 15: Selecting the tuning parameter

4 Conclusions

The main aim of this project was to build a recommendation system using the MovieLens 10M dataset that predict movie ratings with a root mean square error of less than 0.86490. taking all estimated biases into account like effect by the movie, user, genre, release year and review date, and then regularizing these in order to restrict the variability of effect sizes, met the project objective yielding a model with an RMSE of 0.86482. This was validated in a final test using the previously unused validation dataset, with an RMSE of 0.86406.

Although the algorithm developed here met the project objective it still includes a considerable error loss ,uncertainity, not all of which may be considered truly independent, which suggests that there is still much needed study to improve the accuracy and efficiency of the recommendation system with methods that can be used to minimize these dependent errors. One such useful technique is matrix factorization, a useful method for item-based or user-based collaborative filtering which can be used to express residuals within this error loss based on trends observed between groups of users or groups of movies such that the residual error in predictions can be further minimized.

The techniques,functions implemented in this project were limited with less functionality due to the impracticality of using some powerful tools to train such a large dataset on a personal computer. One of the key advantages of matrix factorisation which has contributed to its popularity in recommendation systems is that it is both scalable and compact which makes it memory efficient and compatible with use on personal computers [@koren_2009b]. Thus, further work on the recommendation system developed here would focus on the use of matrix factorisation and also to run these large dataset on high resources machines to train dataset efficiently.

Table 2: Individual genres ranked by number of ratings

Genre	No. of Ratings	Ave. Rating
Drama	3877927	3.67
Comedy	3443953	3.45
Thriller	2325849	3.51
Action	2140992	3.40
Adventure	1844966	3.50
Romance	1711484	3.55
Sci-Fi	1338439	3.40
Crime	1300261	3.66
Fantasy	921492	3.50
Children	729300	3.43
Horror	669599	3.30
Mystery	568035	3.68
War	511147	3.78
Animation	438536	3.59
Musical	432967	3.56
:Action	412989	3.53
Western	189297	3.56
Film-Noir	118541	4.01
:Comedy	93178	3.03
Documentary	79560	3.79
:Adventure	63926	3.36
:Animation	28632	3.77
:Crime	27454	4.07
:Drama	25240	3.43
:Horror	21884	2.31
:Documentary	13392	3.73
:Children	8694	2.64
IMAX	8181	3.77
:Fantasy	4145	2.99
Rouge) (1994)::Drama	3648	4.14
Bleu) (1993)::Drama	3312	4.01
:Sci-Fi	2729	3.31
Blanc) (1994)::Comedy	2688	3.92
The Final Dimension) (1994)::Action	1921	2.46
Jian dan ren wu) (1996)::Action	1910	3.33
Chao ji jing cha) (1992)::Action	1458	3.45
Miami Beach (1988)::Comedy	1111	1.78
Tengoku no Tobira) (2001)::Action	1033	3.81
:Romance	616	2.96
:Mystery	297	3.22
:Musical	113	2.15
:Western	97	2.89
mi ricordo, sì, io mi ricordo) (1997)::Documentary	72	3.57
:Thriller	50	1.72
La prophétie) (2003)::Action	50	2.97
Ichijôji no kettô) (1955)::Action	36	3.47
kettô Ganryûjima) (1956)::Action	35	3.76
Die Sinfonie der Großstadt) (1927)::Documentary	31	3.45
Kowokashi udekashi tsukamatsuru) (1972)::Action	29	3.97

Table 3: Top 10 Movies by Number of Ratings

title	n
Star Wars	95038
Lord of the Rings	38488
Star Trek	34677
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Ace Ventura	29389
Jurassic Park (1993)	29360
Mission	28134
Shawshank Redemption, The (1994)	28015

Method	RMSE	Difference
Project objective	0.86490	-

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881
Movie + User effects (b_u)	0.86617	0.00127

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881
Movie + User effects (b_u)	0.86617	0.00127
Movie, User and Genre effects (b_g)	0.8658	9e-04

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881
Movie + User effects (b_u)	0.86617	0.00127
Movie, User and Genre effects (b_g)	0.8658	9e-04
Movie, User, Genre and Year effects (b_y)	0.86566	0.00076

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881
Movie + User effects (b_u)	0.86617	0.00127
Movie, User and Genre effects (b_g)	0.8658	9e-04
Movie, User, Genre and Year effects (b_y)	0.86566	0.00076
Movie, User, Genre, Year and Review Date effects (b_r)	0.86549	0.00059

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567
Movie effects (b_i)	0.94371	0.07881
Movie + User effects (b_u)	0.86617	0.00127
Movie, User and Genre effects (b_g)	0.8658	9e-04
Movie, User, Genre and Year effects (b_y)	0.86566	0.00076
Movie, User, Genre, Year and Review Date effects (b_r)	0.86549	0.00059
Regularised Movie, User, Genre, Year and Review Date effects	0.86482	-0.00008

Method	RMSE	Difference
Project objective	0.86490	-
Validation of Final Model	0.86406	-0.00084