HarvardX: PH125.9x Data Science: Capstone Course Movie Rating Prediction Project

Andrew Hood

March 12, 2024

Contents

1 Introduction **3**

1.1 Edx dataset 3

2 Analysis **3**

2.1 Ratings ($rating) 4

2.2 Movies ($movieId) 4

2.3 Users ($userId) 4

2.4 Movie Genre ($genres) 7

2.5 Movie Title ($title) 7

2.6 Date of review ($timestamp) 10

3 Methods **10**

3.1 Splitting the edx dataset into train and test sets 10

3.2 Calculating the error loss 11

3.3 Developing the algorithm 11

3.4 Regularizing the algorithm 12

3.5 Validating the final model 13

4 Results **13**

4.1 Simple average 13

4.2 Adjusting for movie effects 14

4.3 Adjusting for user effects 14

4.4 Adjusting for genre effects 15

4.5 Adjusting for release year effects 16

4.6 Adjusting for review date effects 17

4.7 Effect of regularization 18

4.8 Final test in final_holdout_test dataset 19

5 Conclusion **19**

1 Introduction

The Movie Lens project forms part of the HarvardX: PH125.9x Data Science: Capstone course; the final course in the Harvardx Data Science Professional Certificate series. This project will use the movielens dataset (provided) with the objective of creating a machine learning algorithm that can predict movie ratings with a Root Mean Square Error (RMSE) of below 0.86490.

This report will outline the analyses and methods used in development, as well as the results and any future action items pertinent to the project.

First, we will begin by exploring the contents of the dataset, identifying any trends or biases that should be accounted for. After this, we will dive deeper into each variable in the dataset to examine the effect on movie ratings. Once we have these effects, we will account for them when developing the algorithm, and then regularize these effects to maximize the accuracy of the model. Finally, we will report our findings in the results section, and determine if we met our RMSE goal. Lastly, in the conclusion section we will talk about limitations of this project, and any future work that could further minimize the RMSE.

1.1 Edx dataset

Firstly, lets take a preliminary look at the dataset, and examine the data to identify any potential biases and see if there are any missing values.

The edx dataset contains 9,000,055 rows and 6 columns. These columns are; userId, movieId, rating, timestamp, title and genres. We can see that 69,878 unique users provided ratings for 10,677 unique movies. If every user had provided a rating for every movie the dataset would include a total of approximately 746 million ratings. Therefore, we can conclude that the dataset contains missing values since every user did not rate every movie. Given the missing values, it will be important to examine any effect that the 6 variables had on ratings given. In the analysis section of this report, we will do just this.

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

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

2 Analysis

In this section we will take a closer look at the data on a variable by variable basis to identify any patterns, or effects that each variable may present. If a variable is found to cause an effect

on a movies rating, it will be accounted for when developing the algorithm to control for bias and enhance prediction accuracy.

2.1 Ratings ($rating)

The mean overall rating in the edx dataset was 3.51. The minimum rating of any movie was 0.5 and the maximum rating was 5. The total ratings distribution included in the dataset (Figure 1) shows the most common rating across all movies was 4, and that, full-star ratings (7,156,885; 79.5%) were more common half-star ratings (1,843,170; 20.5%).

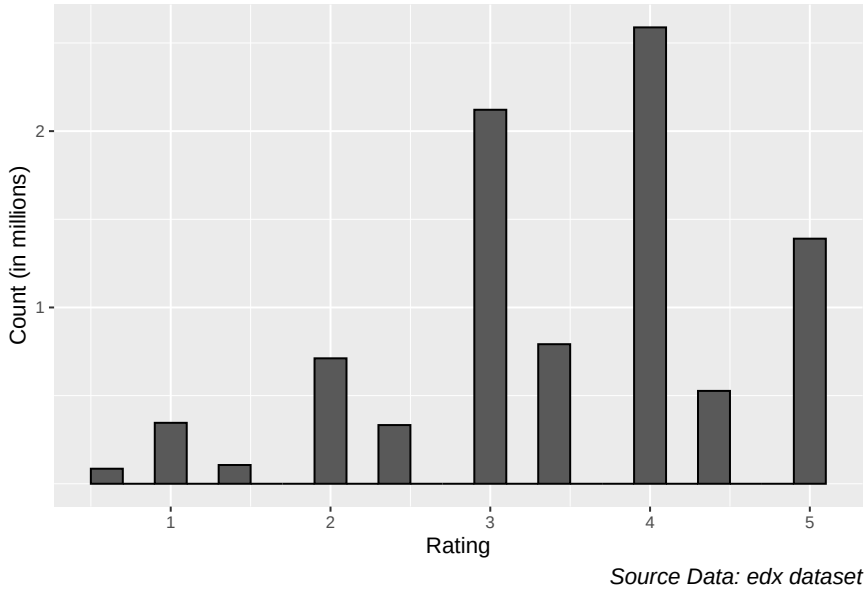

Figure 1: Overall ratings distribution

2.2 Movies ($movieId)

Initial analysis revealed human nature— some people prefer certain movies over others, leading to a variation in ratings (Figure 2). Initial analysis also reveals a wide range in the number of ratings for any given movie (Figure 3), The movie with the most ratings was, Pulp Fiction (1994), receiving a total of 31362 ratings whereas 126 movies were only rated once. This data clearly shows a movie effect present for awarded ratings. Due to this effect, it was deemed worthwhile to adjust for movie effect in the training algorithm.

2.3 Users ($userId)

Analysis of user data showed an effect similar to the movie data; humans have an inherent bias when it comes to entertainment. This shows us that some users rated movies higher than other users (Figure 4). In addition to this, some users contributed more ratings than other users (Figure 5). For example, one user provided a total of 6616 ratings whereas 1059 users provided fewer than 10 movie ratings. Thus, we see a clear user bias that should be adjusted for in the training algorithm. This adjustment should improve recommendation accuracy.

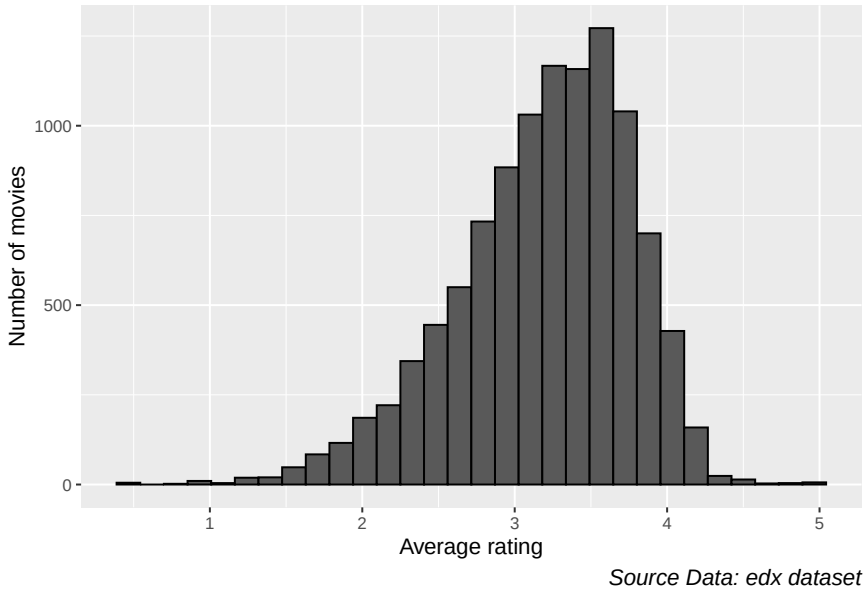

Figure 2: Movie distribution by average rating

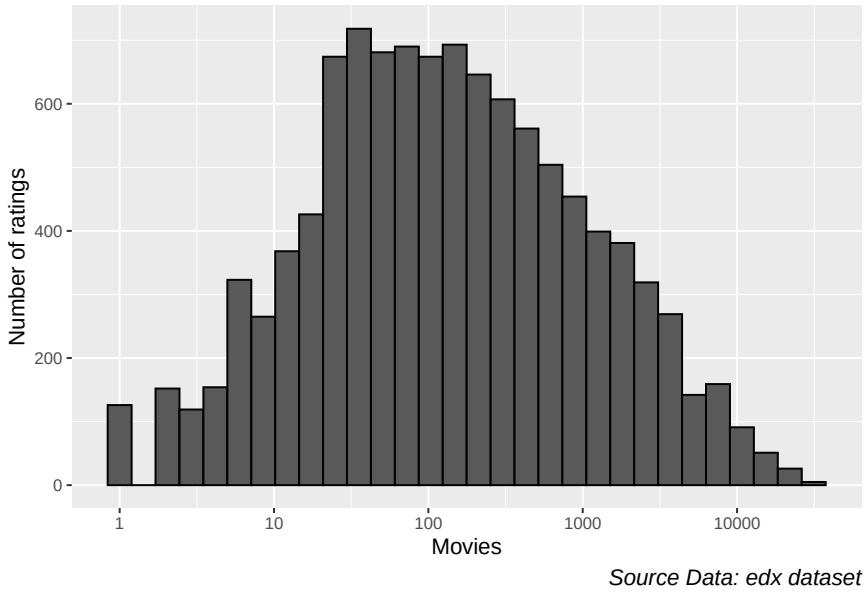

Figure 3: Number of ratings by movie

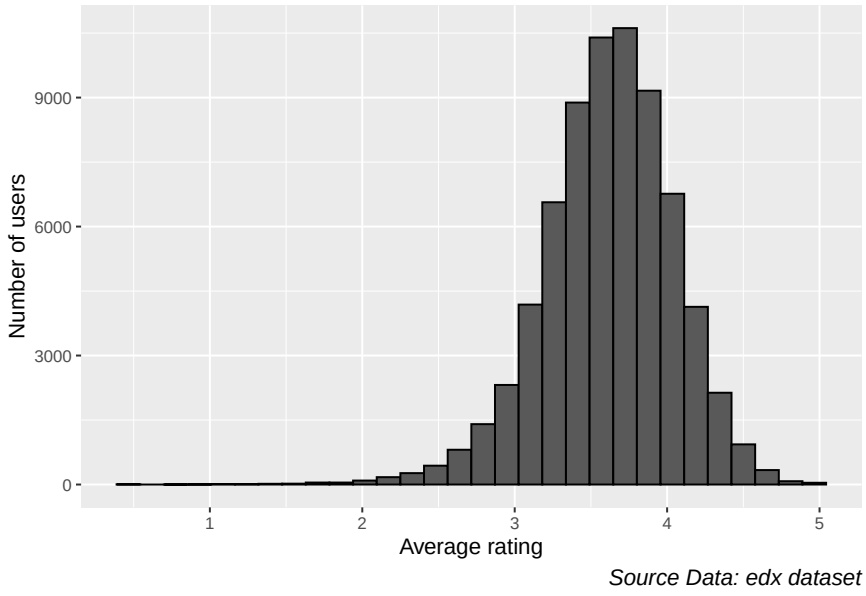

Figure 4: User distribution by average rating

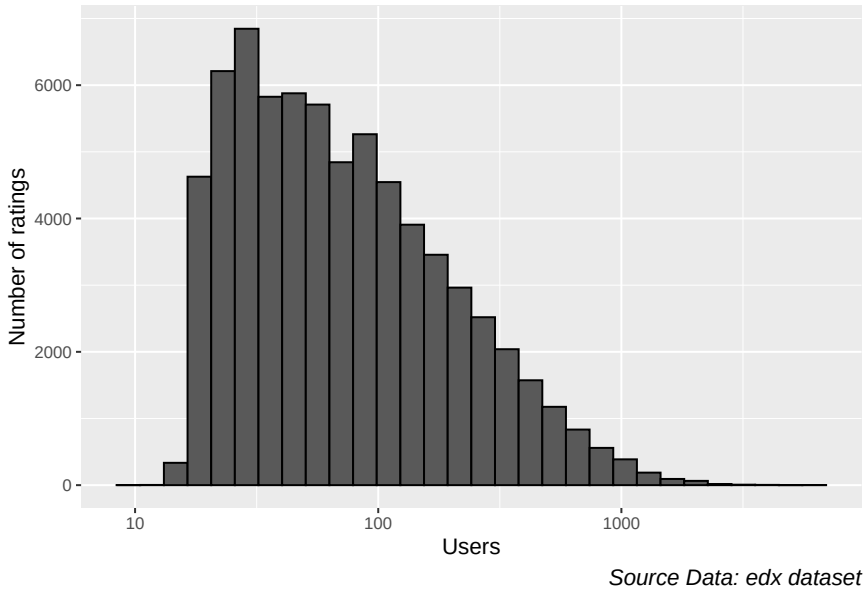

Figure 5: Number of ratings by user

2.4 Movie Genre ($genres)

As shown previously in Table 1, 'genres' identifies the category of each movie in the dataset. Some movies identified with multiple categories, leading to 797 category combinations. After breaking these combined-category observations out into rows with a single category, we identified 20 different categories including 'No Genre Listed'. Using this data, it was possible to rank these categories by the number of ratings receieved (Table 2).

Table 2: Individual genres ranked by number of ratings

Genre	No. of Ratings	Ave. Rating
Drama	3910127	3.67
Comedy	3540930	3.44
Action	2560545	3.42
Thriller	2325899	3.51
Adventure	1908892	3.49
Romance	1712100	3.55
Sci-Fi	1341183	3.40
Crime	1327715	3.67
Fantasy	925637	3.50
Children	737994	3.42
Horror	691485	3.27
Mystery	568332	3.68
War	511147	3.78
Animation	467168	3.60
Musical	433080	3.56
Western	189394	3.56
Film-Noir	118541	4.01
Documentary	93066	3.78
IMAX	8181	3.77
(no genres listed)	7	3.64

Drama and comedy movies had the largest number of ratings while Documentary and IMAX movies had the lowest number of ratings. Seven ratings were provided for movies where no genre was listed. Table 2 also shows a variation in average rating by genre. Grouping the data by unique genre combinations and filtering to only those genre combinations with at least 100,000 ratings shows a clear genre effect with 'Comedy' movies achieving the lowest average rating while 'Crime|Drama' and 'Drama|War' films achieved the highest average rating (Figure 6). Thus, genre effect should be adjusted for in the training algorithm to improve the accuracy.

2.5 Movie Title ($title)

The title variable includes both the title of the movie and the year of release. Table 3 shows the 10 movies in the dataset with the most ratings.

In order to investigate the possible effect of release year on average rating, the title string had to be split into two separate columns, one for the title and the other for the year of release. The

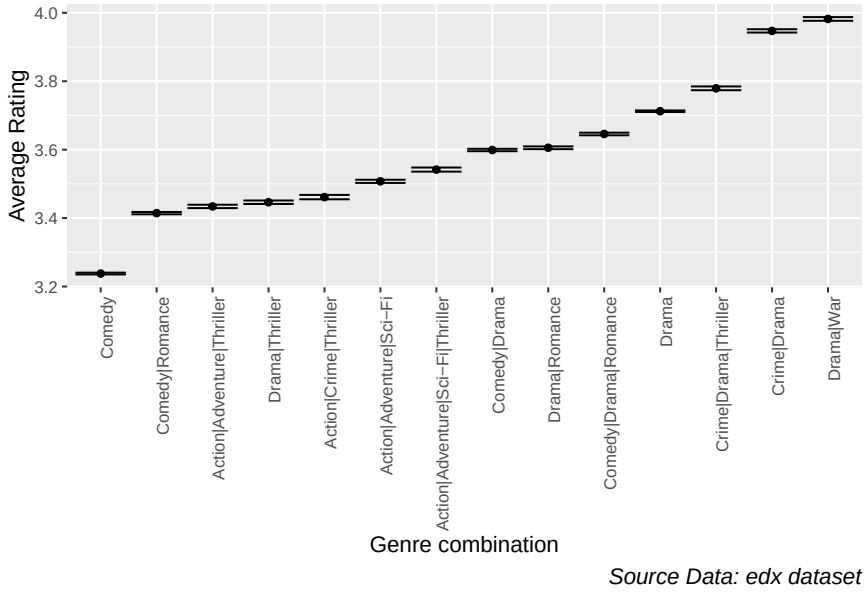

Figure 6: Average rating by genre

Table 3: Top 10 Movies by Number of Ratings

title	n
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

new dataset was then used to identify any potential release year effect on average rating. We see that average rating in fact varied by year of release (Figure 7). An interesting observation is that average rating was highest for movies released between 1940 and 1950, and has declined for movies made since that decade.

However, it is important to note that movies made before 1970 tend to have a lower number of ratings. The 1990's saw the creation of movies with the highest number of ratings. Specifically, 1995 which accounted for roughly9% of the total number of ratings present in the dataset. Therefore, release year should be adjusted for when creating a training algorithm. However, it is important to note the uncertainty caused by small sample sizes for certain years in the dataset.

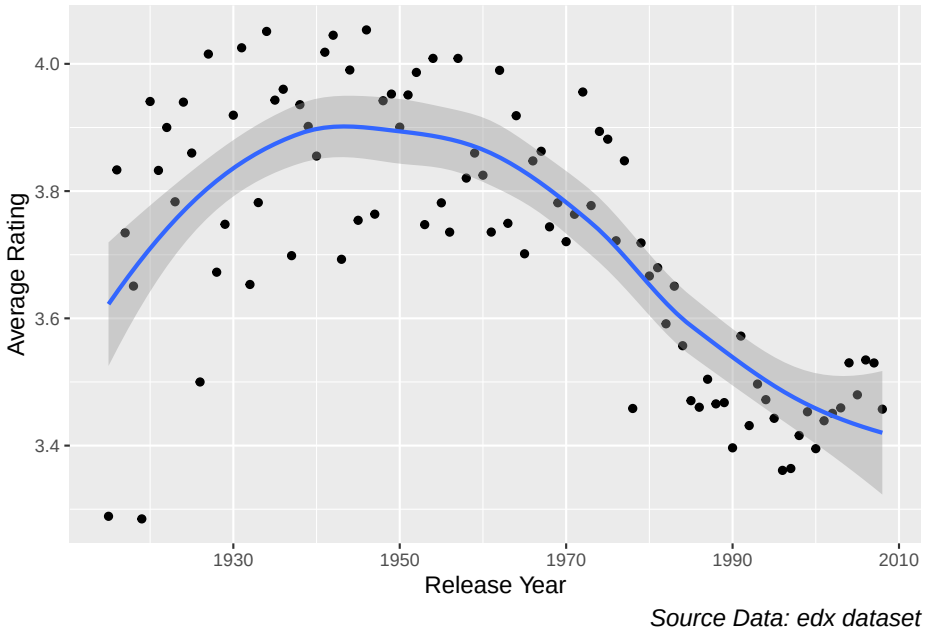

Figure 7: Average rating by year of release

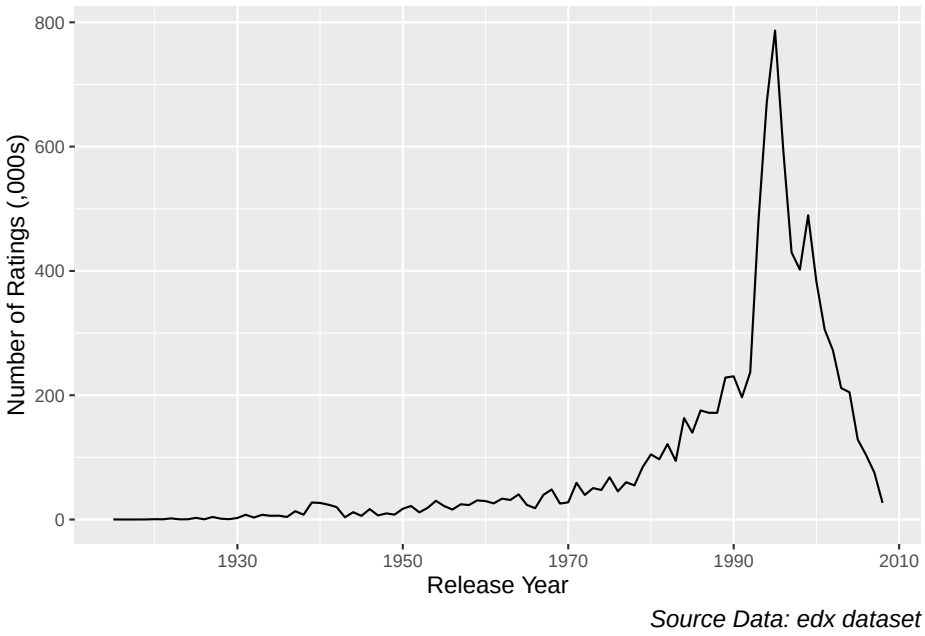

Figure 8: Number of ratings by year of release

2.6 Date of review ($timestamp)

The timestamp is a convenient way of digitally recording both date (yymmdd) and time (hhmmss) information, based on an epoch (time zero) of midnight on 1 January 1970. To aid analysis of the effect of review date on ratings, the timestamp data was transformed into date format, omitting time data and rounding to the nearest week.

The earliest review included in the dataset was given in 1995. This rating was given when the average rating was highest, prior to the gradual decline mentioned earlier, before the eventual increase in average rating in 2005. Review data has a small effect relative to the movie and user effect, as shown over time in Figure 9, however, it is still valuable to account for in the algorithm in the pursuit of maximum accuracy.

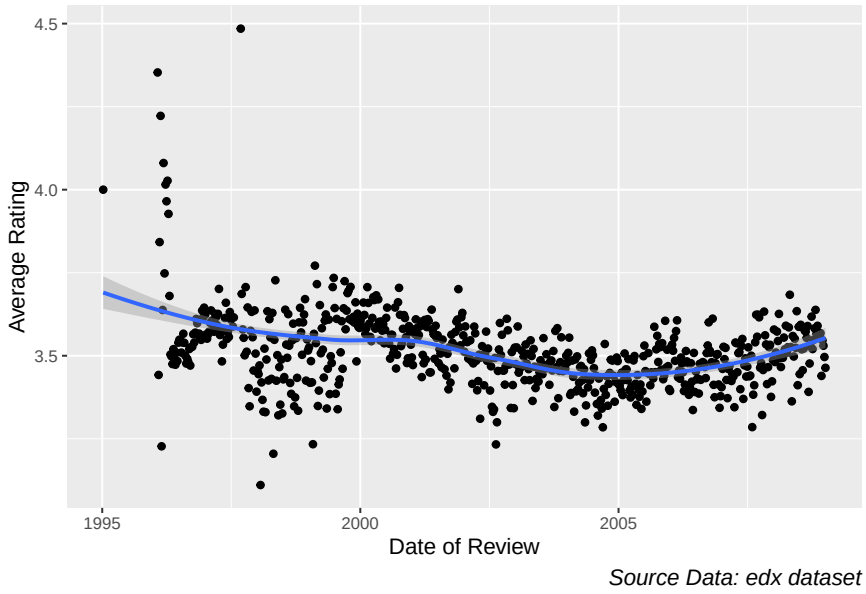

Figure 9: Average rating by date of review

3 Methods

3.1 Splitting the edx dataset into train and test sets

Since the final_holdout_test dataset had to be reserved for final RMSE verification, the edx dataset had to be used for both training and testing the algorithm. This allows the benefit of cross-validation for the model, while minimizing he risk of over-training the algorithm.

For splitting the edx dataset, a similar approach was used as before on the movielens dataset, where the caret function 'createDataPartition' was used to divide the edx dataset into 2 smaller datasets, 80% for train, and 20% for test. Next, the same as in the original approach, the dplyr functions 'semi_join' and 'anti_join' were used to ensure the validity of the train and test datasets for their respective purposes.

3.2 Calculating the error loss

The root mean square error (RMSE) is the standard deviation of the residuals. The residuals are the distance of a data point from the regression line. Therefore, RMSE essentially measures the spread of the data, or the concentration of data points around the regression line. The RMSE was calculated to represent the error loss between the predicted ratings derived from applying the algorithm and actual 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, and N is the total number of user/movie combinations.

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

The goal of the project was to develop an algorithm that achieved an RMSE below 0.86490 as set out below. A simple table was created to capture the project goal as well as the results obtained during development within the edx dataset and in the final_holdout_test dataset (Section 4: Results).

Method	RMSE	Difference
Project objective	0.86490	-

3.3 Developing the algorithm

The simplest algorithm for predicting ratings is to apply 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 average of all ratings is the estimate of μ that minimizes the RMSE. Thus, $\hat{\mu}$ = mean(train_set$rating) was the simple formula used to train the first algorithm.

The analysis detailed in the prior section showed that ratings were not equal across all movies in the dataset. This means some movies received a higher average rating Sthan others and accounting for the effects behind this variation will improve the accuracy of the prediction. Thus, the training algorithm was further refined 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 Analysis section also showed a variation in how users rated movies so further refinements were made to the algorithm to adjust for user effects (b_u). As previously, rather than fitting linear regression models, the least square estimate of the user effect, $\hat{b}_u$ was calculated using the formulas shown below.

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

Movie ratings were also dependent on genre, with some genres receiving higher average ratings than others. This effect was observed even when movies were associated with multiple genres, as they were in the original movielens dataset. Thus, the rating for each movie and user was further refined by adjusting for genre effect, b_g, and the least squares estimate of the genre effect, $\hat{b}_g$ was 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 release year of the movie. The Analysis section revealed an effect of the release year, b_y, on the movies rating. The least squares estimate of the year effect, $\hat{b}_y$ was calculated using the formula shown below, further reducing bias in the algorithm.

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

Next, there was a small effect of the date of review (b_r) on the average rating given for each movie. This was incorporated into the model by applying a smooth function to the movie's release date for each rating given by movie and user. Next, by rounding the date of review to the nearest week, the data was effectively smoothed. The least squares estimate taking review date effect, $\hat{b}_r$ into account was calculated using the formula shown below.

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

3.4 Regularizing the algorithm

Lastly, Analysis section revealed that average rating was affected by movie, user, genre, release year and review date. However, it was also noted that the number of ratings for each movie and by each user varied rather significantly. The affect of these variations (b) is the increased uncertainty due to the smaller relative sample sizes of the affected categories.

Regularization is a useful tool in ensuring the accuracy of an algorithm. The regularization process allows us to prevent over-fitting the model against random patterns in the data. In essence, regularization 'simplifies' the final result. The penalty term , λ, is a parameter used for tuning that is

chosen during the cross-validation process with the edx dataset. Lastly, the movie effect, b_i can be used to regulate this effect as shown below:

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

Based on the above, the least squares estimate for the regularized effect of movies can be calculated as below, where n_i is the number of ratings recieved for movie i. The effect of $\frac{1}{\lambda+n_i}$ is such that when the sample size is large, i.e. n_i is a large number, λ has little impact on the estimate, $\hat{b}_i(\lambda)$. On the other hand, where the sample size is small, i.e. n_i is small, the impact 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 regularization model was developed to adjust for all of the effects previously described, as shown below. A range of values for λ (range: 4-6, with increments of 0.1) was applied in order to tune the model to minimize the RMSE value. As before, all tuning was completed within the edx dataset using the train and test datasets, to avoid over-training the model in the final_holdout_test dataset.

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

3.5 Validating the final model

After refining the algorithm with the edx train and test datasets, the final stage of the study was to train the model using the entire edx dataset. Then, using this trained algorithm, to predict movie ratings for the final_holdout_test dataset. Before involving the final_holdout_test dataset, we must incorporate the release year and review data columns into the data using the dplyr package.

The final model adjusted to account for movie, user, genre, release year, and review date bias was regularized using the optimal λ, and then used to predict movie ratings in the final_holdout_test dataset, and lastly, calculate the final RMSE.

4 Results

4.1 Simple average

Predicting the average rating from the train set (3.51) for every entry in the test set resulted in a RMSE of 1.06, which was above the stated goal for the study. Additionally, an RMSE of 1.06 means that predicted ratings are greater than 1 star away from the actual rating, resulting in a rather inaccurate movie recommendation system.

Method	RMSE	Difference
Project objective	0.86490	-
Simple average	1.06057	0.19567

4.2 Adjusting for movie effects

Figure 10 shows that the estimate of movie effect (b_i) varies considerably across all movies in the train set. Adding this effect into the algorithm, in order to adjust for the movie effect, improved the accuracy of the model by 11.02%, yielding an RMSE of 0.94, better than before, but still greater than our goal.

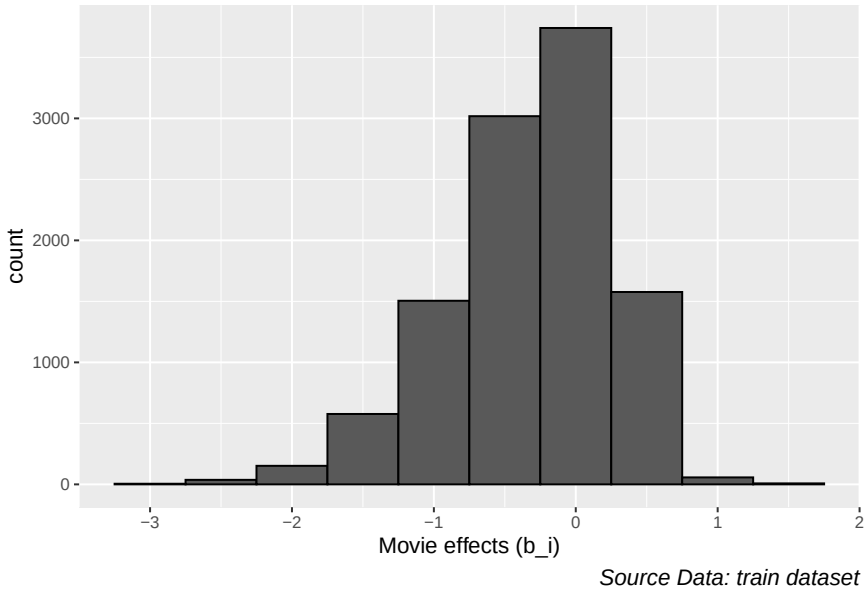

Figure 10: Distribution of movie effects

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

4.3 Adjusting for user effects

Figure 11 shows the estimated user effect (b_u) building on the movie effect model above. While b_u showed less variability than was observed with b_i, it was obvious that adjusting for user effect further enhanced the accuracy of the algorithm. After, adjusting for user effects, the algorithm returned an RMSE of 0.86617, proving this theory. Thus, adjusting for both movie and user effect improved the RMSE by 18.33% relative to the simple model.

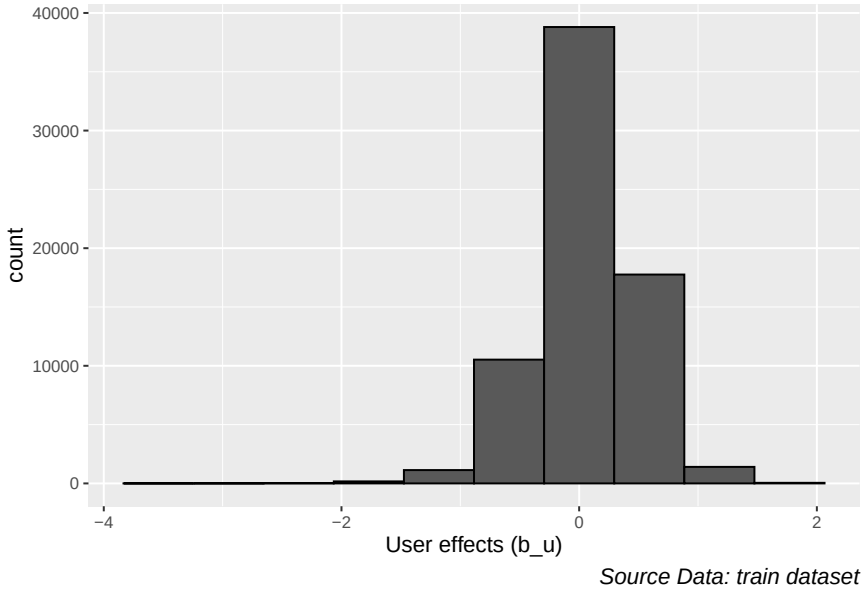

Figure 11: Distribution of user effects

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

4.4 Adjusting for genre effects

Figure 12 shows the distribution of estimated genre effect, b_g in the train set. Which again shows variation across different category combinations.

The output from the model when adjusting for genre, in addition to movie and user bias, returned an RMSE of 0.86582. We can see adding genre effects into the model provided a smaller improvement in overall accuracy of the algorithm, reducing the RMSE by 0.04% versus the previous model and 18.36% versus the original model. This improvement did bring the model very close to meeting the stated goal, reducing the difference to only 0.00092.

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)	0.86582	0.00092

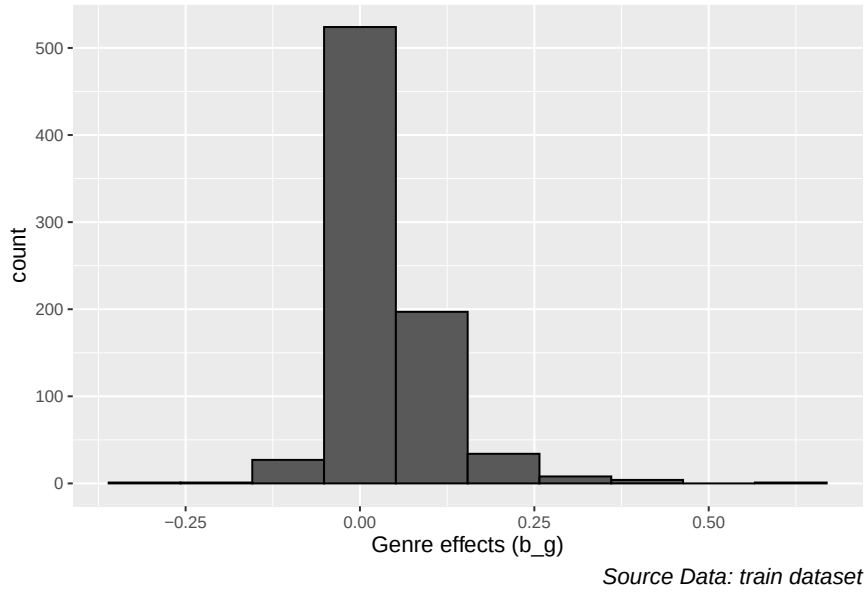

Figure 12: Distribution of genre effects

4.5 Adjusting for release year effects

The year of movie release adds slight variability to the average rating in the train set as shown in Figure 13. Incorporating this effect into the algorithm provided a slight improvement in model accuracy of 0.02% in the accuracy of ratings prediction bringing the RMSE slightly closer to meeting the goal at 0.86567.

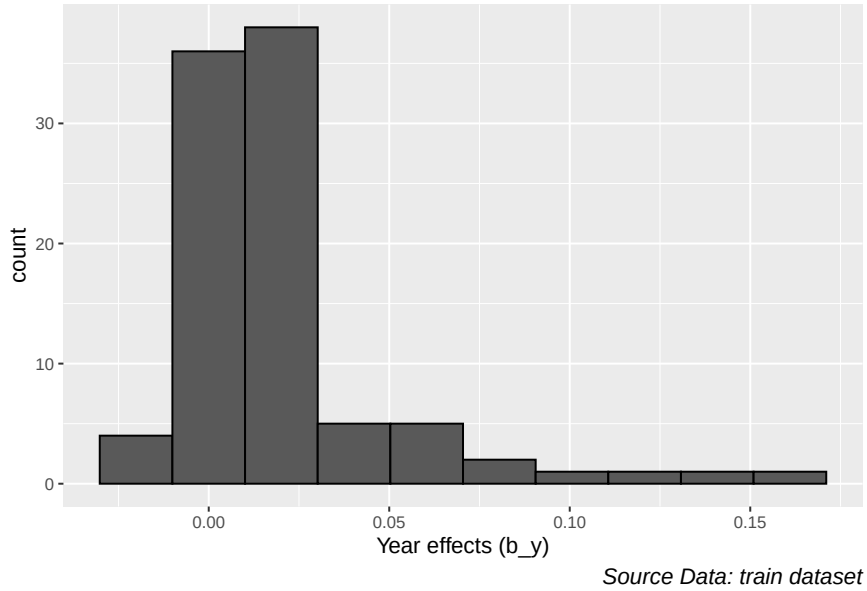

Figure 13: Distribution of release year effects

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)	0.86582	0.00092
Movie, User, Genre and Year effects (b_y)	0.86567	0.00077

4.6 Adjusting for review date effects

The final effect to account for was review date. The Analysis section showed this had a minor impact on ratings which was confirmed by showing the distribution of b_r in Figure 14.

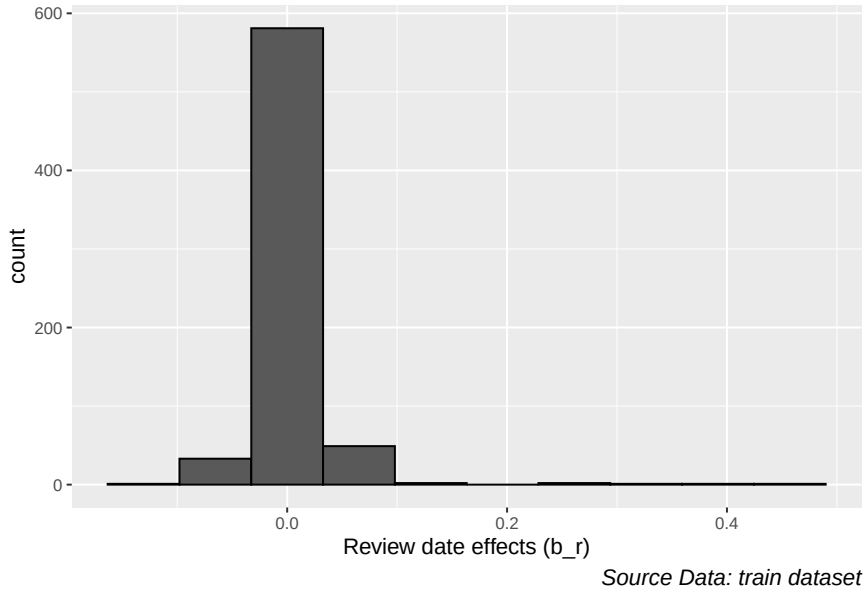

Figure 14: Distribution of review date effects

Accounting for review date effect delivered an RMSE of 0.86549, an improvement of 18.39% versus the original model but still not quite as low as we needed.

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)	0.86582	0.00092
Movie, User, Genre and Year effects (b_y)	0.86567	0.00077
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)	0.86582	0.00092
Movie, User, Genre and Year effects (b_y)	0.86567	0.00077
Movie, User, Genre, Year and Review Date effects (b_r)	0.86549	0.00059
Regularized Movie, User, Genre, Year and Review Date effects	0.86483	-0.00007

4.7 Effect of regularization

The final step in developing and enhancing the model was to apply regularization. Figure 15 shows the RMSE delivered across each of the values for λ tested. The optimal value for λ was 5.1 which reduced the RMSE to 0.86483, which was low enough to accomplish the goal RMSE. This represented a total improvement of 18.46% in the accuracy of the model by adjusting for movie, user, genre, release year and review date effects and applying regularization to the combination of these effects.

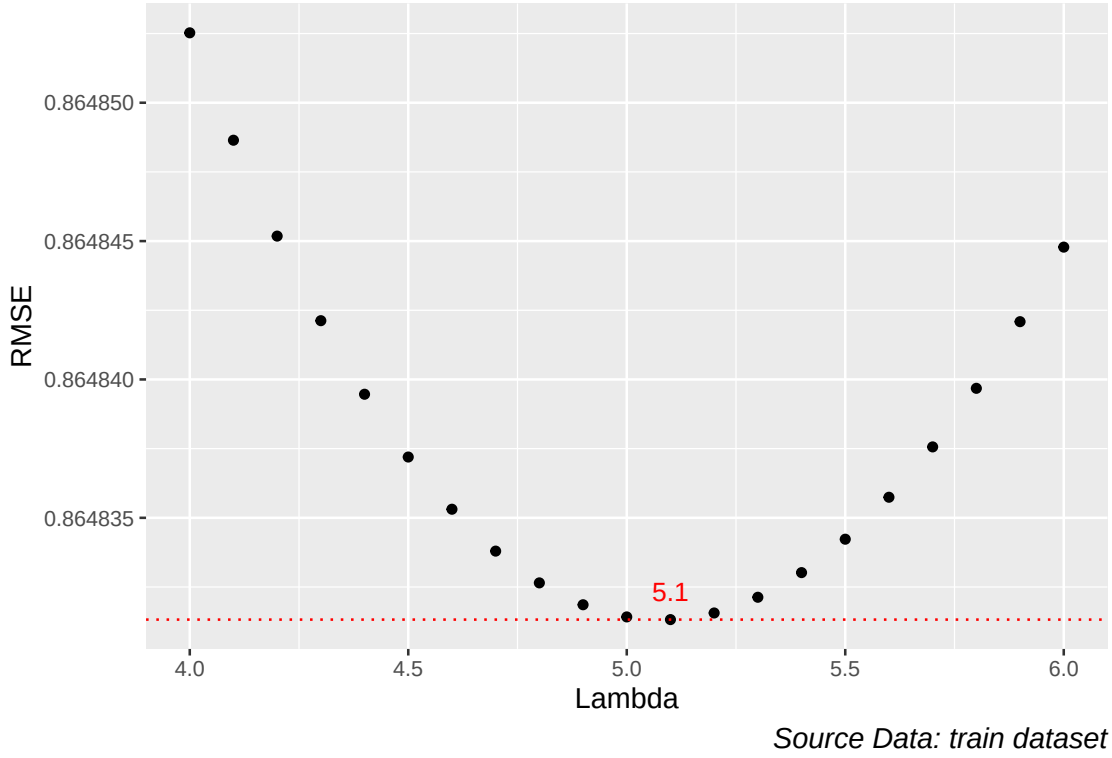

Figure 15: Selecting the tuning parameter

Method	RMSE
Project Goal:	0.86490
Final Model:	0.86405

4.8 Final test in final_holdout_test dataset

The model's final test using the final_holdout_test dataset achieved an RMSE of 0.86405, an improvement of 18.53% versus the simple model based on the overall average rating and 0.00085 below the RMSE goal.

5 Conclusion

Using the movielens 10M dataset, the goal of this study was to create a movie reccommendation system that could achieve an RMSE value below 0.86490. After adjusting for multiple biases inherent in the data, and regularizing the combinations of these biases, the final model achieved an RMSE of 0.86405, surpassing the stated RMSE goal.

While the model in this report achieved the goal, more work could be done to improve recommendation accuracy. To further account for non-independent error in the model, matrix factorization could be utilized to identify patterns in the data and reduce the residuals of these patterns, thus, yielding an even lower RMSE.