Movie Lens Recommendation Model

Vasileios Plessas

25 February, 2022

Contents

1 Introduction

Recommendation systems are popular applications of machine learning utilised extensively by digital companies [1]. Netflix is an example of a company which uses these systems to understand their customers better and to target them with media content more effectively [2]. In 2009, Netflix awarded a $1M prize to the team of data scientists who had successfully met the challenge of improving their movie recommendation algorithm by 10% [3].

This analysis is part of HarvardX's Capstone Project for the Data Science Professional Certificate. The objective was to develop a recommendation system using the MovieLens dataset which consists of 10 million movie ratings. The goals were for the final algorithm to:

a) Improve predictions by reducing the the root mean square error (RMSE) by 10% or more over the naive algorithm (Just the Average) and
b) Predict ratings with a root mean square error (RMSE) of less 0.8712 (Winning Score of the Netflix challenge) versus the actual ratings included in the validation set.

To facilitate this work, the dataset was split into a training set (edx) and a final hold-out test set (validation) using code provided by the course organisers. As instructed we have not used the validation dataset until the very end of our analysis where we used it to calculate our final RMSE against that hold-out set. Furthermore we've partitioned the edx dataset between train and test sets (edx_train and edx_test accordingly) to allow us to build our algorithm and test our progress as we proceeded.

This report starts by presenting the exploratory analysis used to understand the edx dataset and explore the interactions and distributions of the variables present. It proceeds with presenting the methodology used to develop and test the algorithm and discusses the findings after each iteration of the development process. It concludes with presenting and discusisng the final results as well as any limitations identified and recommendations for future work to be carried out.

2 Exploratory Analysis

The structure of the data set is shown below. The edx dataset is a data.table, data.frame consisting of 9,000,055 rows and 6 columns, with ratings provided by a total of 69,878 unique users for a total of 10,677 unique movies. If each unique user had provided a rating for each unique rating the dataset would include a total of approximately 746 million ratings. Clearly, therefore, this dataset includes many missing values, i.e. every user has not rated every movie.

```
## Classes ’data.table’ and ’data.frame’:   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)"
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "
##  - attr(*, ".internal.selfref")=<externalptr>
```

In order to explore temporal effects later in our analysis such as Release Year and Year Rated, we proceeded with extracting these dimensions from the timestamp field present in our dataset. The new data structure is now as follows:

```
## Classes ’data.table’ and ’data.frame’:   9000055 obs. of  8 variables:
##  $ userId      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId     : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating      : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp   : int  838985046 838983525 838983421 838983392 838983392 838984474 838983653
##  $ title       : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994
##  $ genres      : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller
##  $ release_year: num  1992 1995 1995 1994 1994 ...
##  $ year_rated  : num  1996 1996 1996 1996 1996 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

2.1 Overall Ratings

Figure 1 shows the distribution of the ratings, with the mean rating 3.51 indicated by the blue dashed line. We can see that users tend to rate movies more positively than negatively. They also prefer to give whole star ratings than half star ones.

2.2 Movies

Some movies are naturally more highly rated than others (see Figure 2). Further analysis indicates significant variation in the number of ratings received by each movie (see Figure 3). With a certain number of movies receives the majotiry of ratings while others being rated only a few time. There’s clearly a movie effect present in the data and it’s the first effect we’ll try and capture into our model.

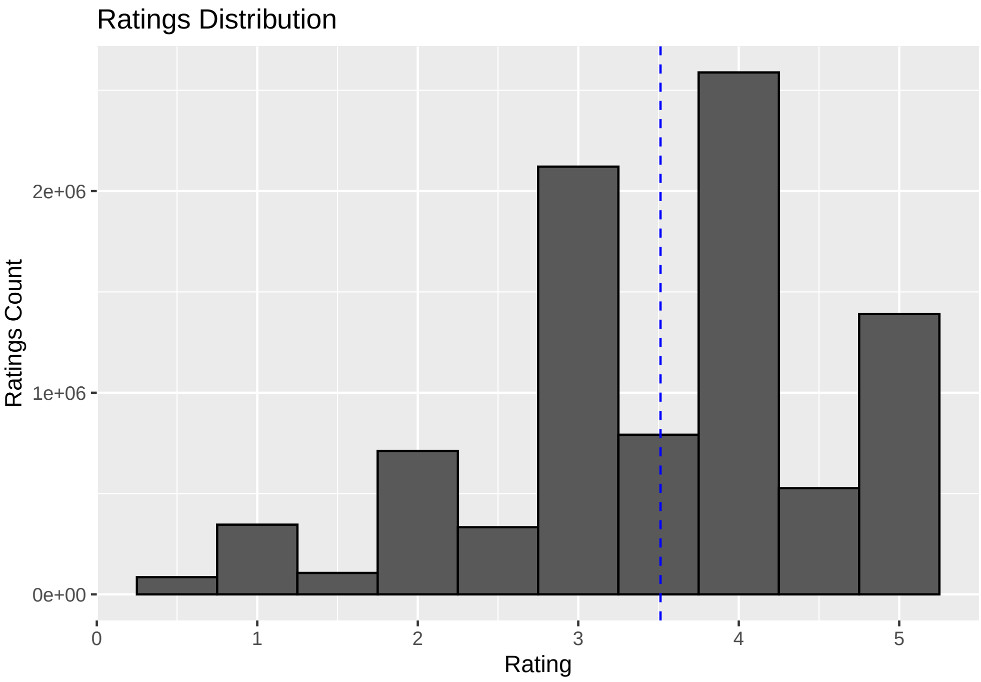

Figure 1: Overall Ratings Distribution

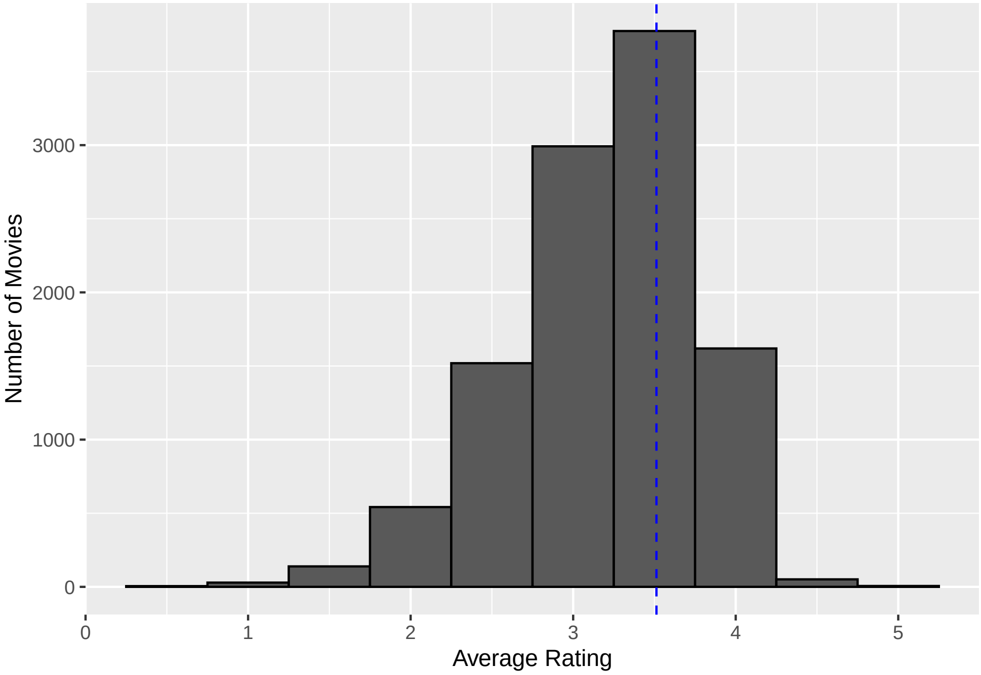

Figure 2: Average Rating by Movie

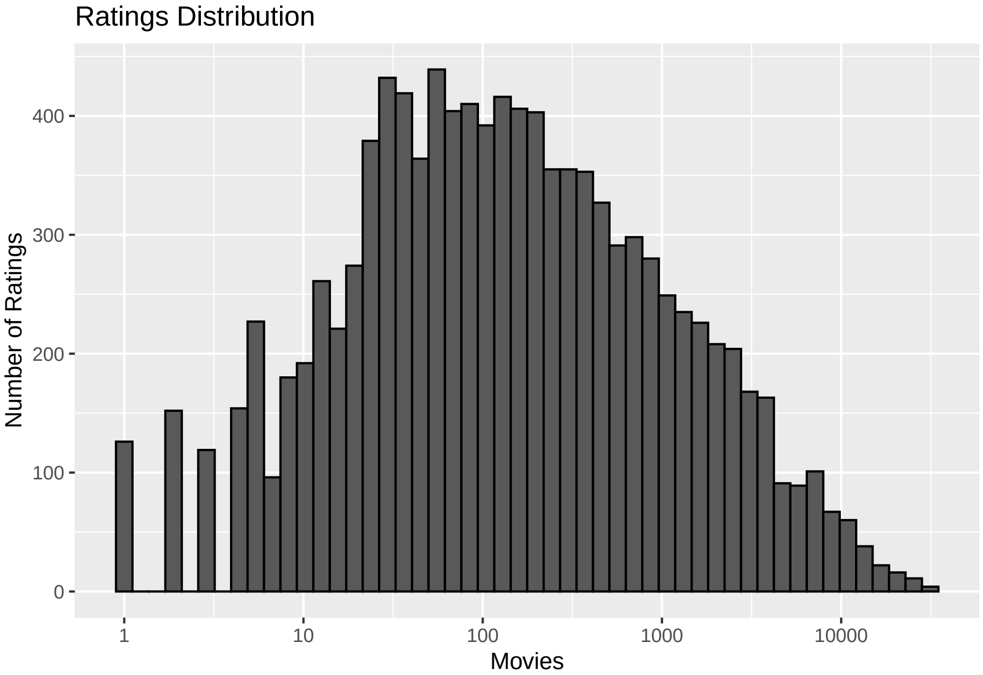

Figure 3: Number of ratings by Movie

2.3 Users

User data shows a pattern of some users being more generous in the way they assessed movies (see Figure 4). Some users contributed many more ratings than other users (Figure 5). For example, one user provided a total of 6616 ratings whereas as many as 1059 provided fewer than 10 movie ratings each. This indicates a clear user effect which, if adjusted for, may further improve the accuracy of a movie recommendation system.

2.4 Release Year

The year the movie was released reveals an interesting trend (see Figure 6). There is a noticeable increase in ratings for movies released between the 30s and 50s. Perhaps even more important is the wide spread of data points away from the distribution's mean up until the 1970s. This effect doesn't continue in the years after where we see the majority of the datapoints converging towards the mean and fall within the 95% confidence interval. For that reason we will make the assumption that the Release year has an effect on user ratings which we will try to capture in our model.

2.5 Year of Review

The year each movie was reviewed (see Figure 7) does not exhibit the strong seasonal effect which we observed in the release year graph. However it shows a 10 year downwards trend starting from 1995 and stabilising around 2005 and afer that it's pretty much flat. We believe that the year of review would not have a large effect on the user rating prediction algorithm, however we will capture it in our model and measure it's impact on the RMSE.

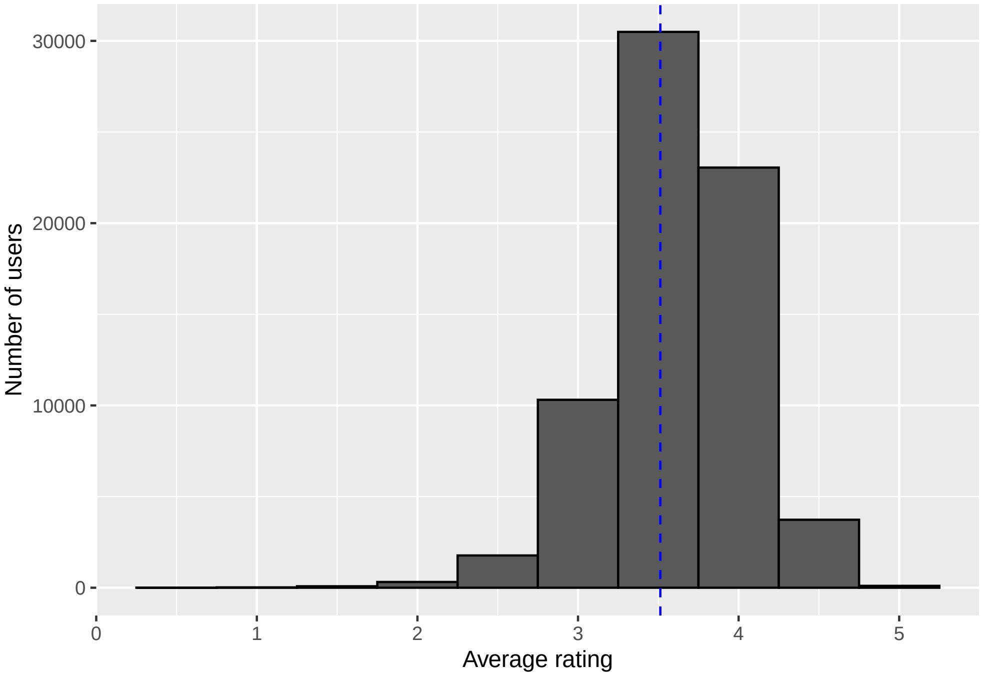

Figure 4: Average Rating by User Distribution

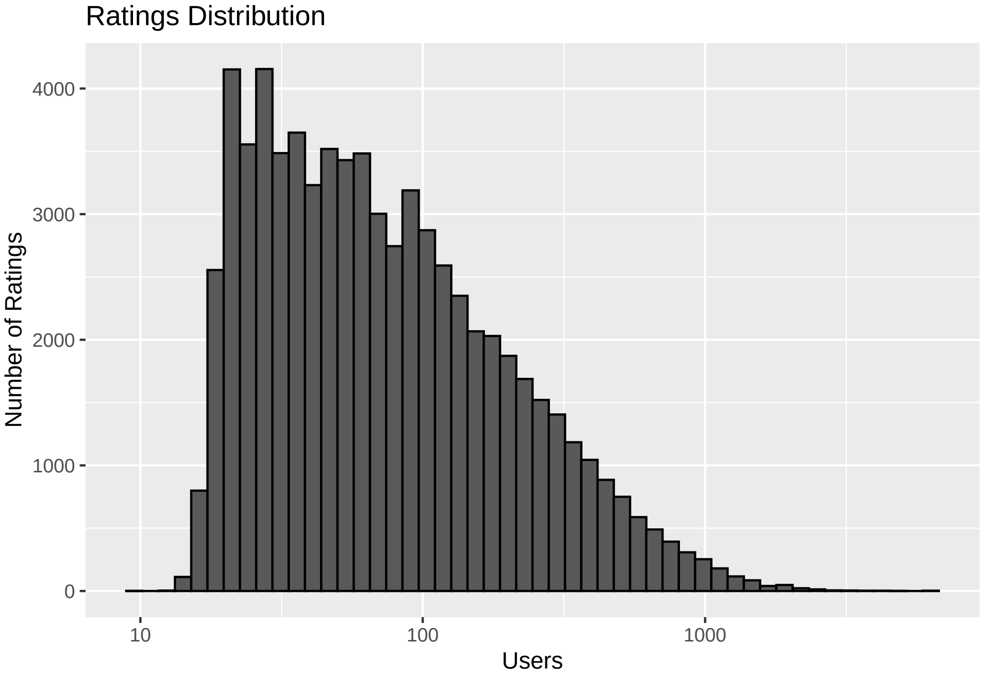

Figure 5: Number of ratings by User

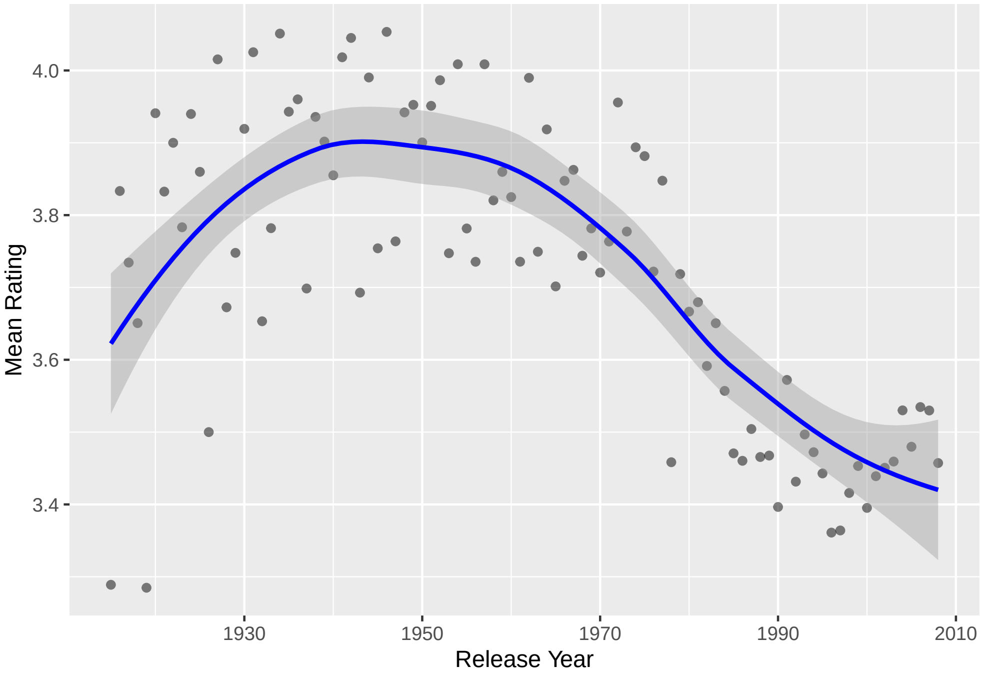

Figure 6: Average Rating Curve based on Year of Release

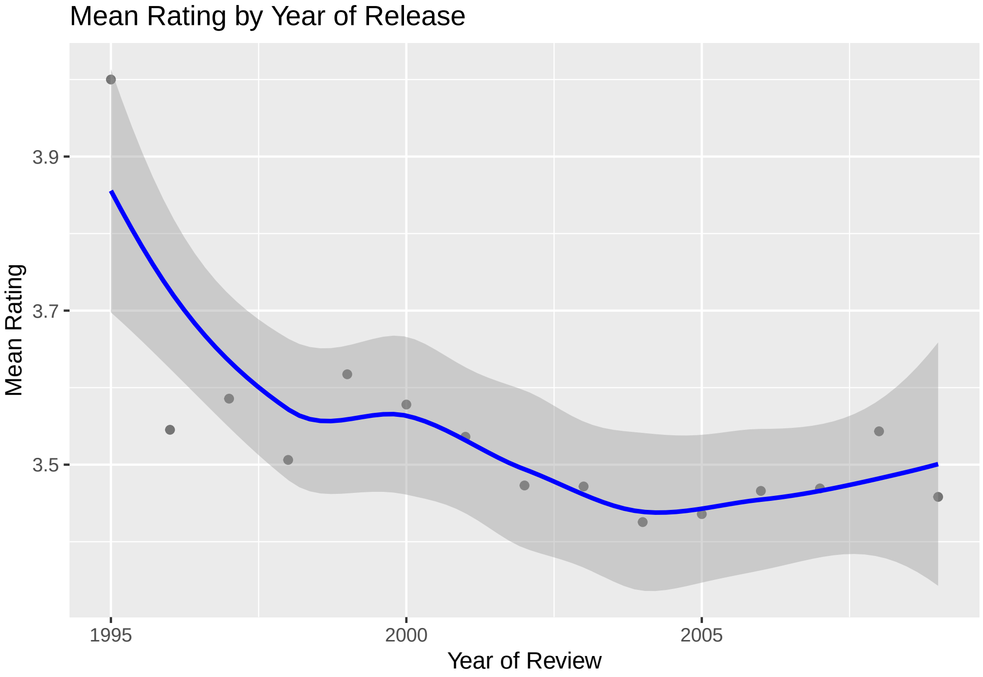

Figure 7: Average Rating Curve based on Year of Review

2.6 Movie Genre

The edx dataset contains a "genres" field which indicates the genre each movie belongs to. The majority of the movie to genre relationships are one to one, meaning that each movie belongs to a single genre. However there many instances where one movie belongs to multiple genres.There is a total of 797 unique combinations in the dataset. To explore the ratings distributions by genre in our dataset, we seperated those multi-genre combinations into individual rows with a single genre. Due to the size of our data set this operation was not possible to complete with processing power of a normal laptop. Hence we've taken a random sample of 1 million rows to analyse.

To ensure we review genres with significant number of ratings, we've further filtered our sample dataset to genres with over 10,000 ratings. We can infer from this chart that there's an indication of a trend in how different genres are rated (see Figure 8). While the majority of genres tend to gravitate towards the edx dataset's average rating 3.51 , others tend to be closer to the opposing ends of the distribution. Horror movies seem to be rated poorly by users while Film-Noir, Documentaries and War movies tend to receive quite high ratings. Perhaps it is worth mentioning the size of the error bars for those high rated genres which seems to be larger when compared to the bulk of the genres converging closer to the mean. This could indicated a higher variability in the star ratings for these genres. Regardless we want to examine the genre effect and it's impact on our predictions so we will capture it in our model.

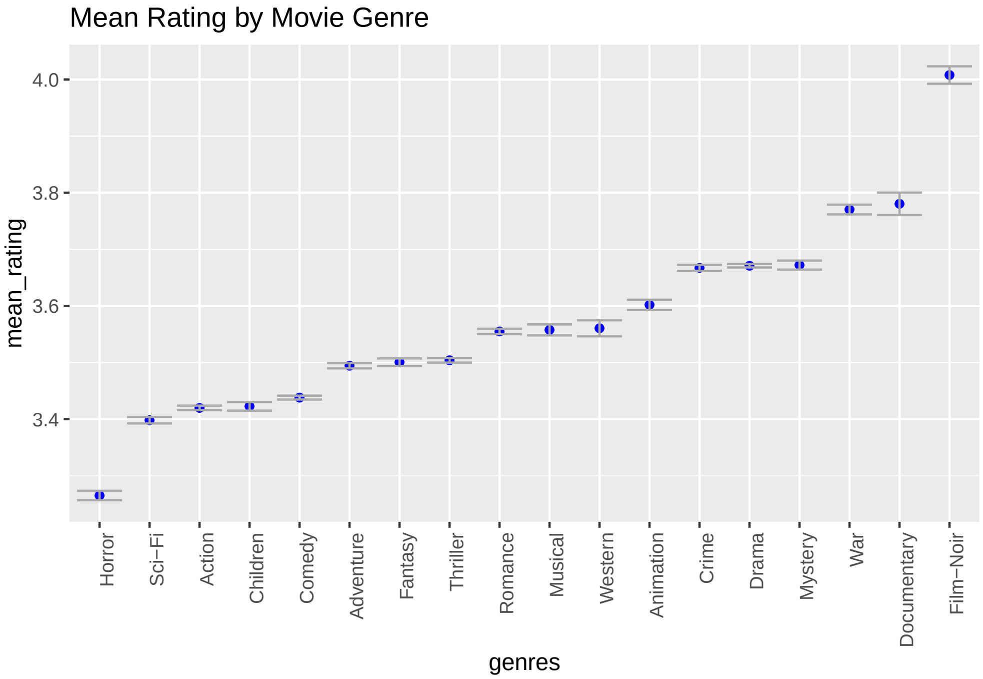

Figure 8: Average Rating by Genre

3 Methodology

As mentioned already in the introduction, the validation dataset was reserved for the final hold-out test, the edx dataset was split into train (90%) and test (10%) sets which were used to train and test the algorithm in development. This is important to allow for cross-validation and refinement of the final model without the risk of over-training. Other methods for cross-validation include K-fold cross validation and bootstrapping but were not utilised here [4].

The goals of this objective is two-fold: a) Improve predictions by reducing the the root mean square error (RMSE) by 10% or more over the naive algorithm (Just the Average) and b) Predict ratings with a root mean square error (RMSE) of less 0.8712 (Winning Score of the netflix challenge) versus the actual ratings included in the validation set.

3.1 Calculating the error loss

The residual mean square error (RMSE) is defined as the standard deviation of the residuals (prediction errors) where residuals are a measure of spread of data points from the regression line [5]. 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.

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

3.2 Algorithm Development

3.2.1 Just the Average

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

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

Predicting the average rating from the train set (3.51) for every entry in the test set resulted in a RMSE of 1.06, substantially above the project objective. Moreover, an RMSE of 1.06 means that predicted ratings are more than 1 star away from the actual rating, an unacceptable error loss for a movie recommendation system. Additionally it is quite far away from the project's second objective to reduce RMSE below 0.8712.

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005

3.2.2 Movie Bias

The next step in our algorithm development is to calculate and adjust for movie bias. As not all movies have received the same rating, by accounting for this effect b_m should improve the accuracy of our algorithm. We can define the models as :

$$Y_{u,m} = \mu + b_m + \epsilon_{u,m}$$

Due to the size of the dataset we cannot use a linear model to explain this relationship. Instead we'll use the least squares estimate of the movie effects $\hat{b}_m$ which can be derived from the average of $Y_{u,m} - \hat{\mu}$ for each movie m .

$$\hat{y}_{u,m} = \hat{\mu} + \hat{b}_m$$

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296

Figure 9 shows that the estimate of movie effect (b_m) varies considerably across all of the movies included in the train set. Adding this effect into the algorithm, in order to adjust for the movie effect, has indeed improved the accuracy of the model, yet still well above the target.

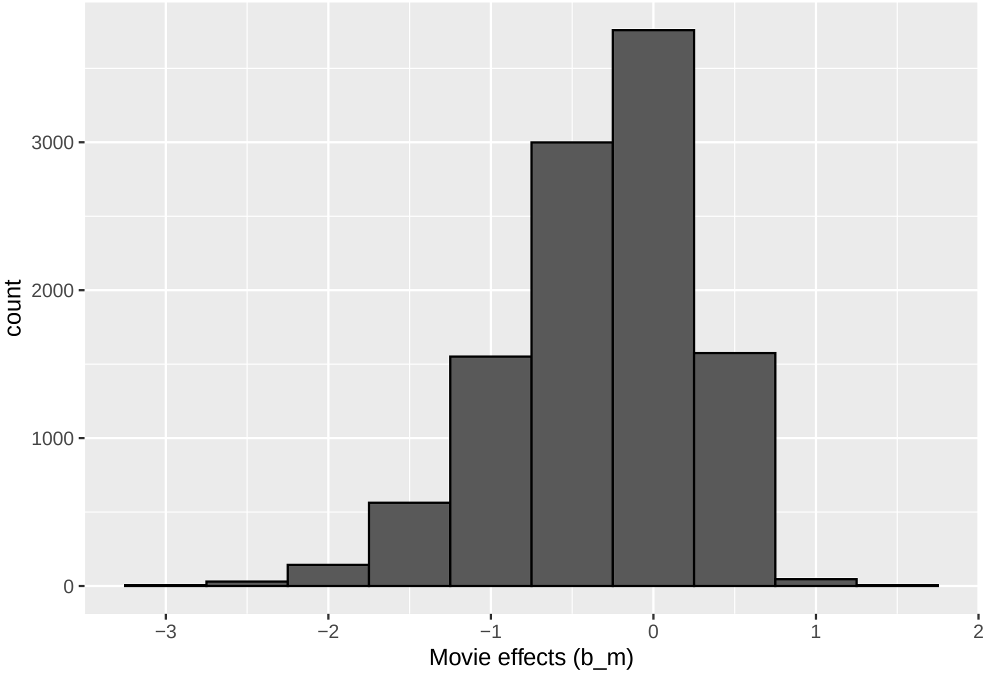

Figure 9: Movie Effect Distribution

3.2.3 User Bias

The exploratory analysis also showed that different users rated movies differently so further refinements were made to the algorithm to adjust for user effects (b_u). The least square estimates of the user effect, $\hat{b}_u$ was calculated using the formulas shown below.

$$Y_{u,m} = \mu + b_m + b_u + \epsilon_{u,m}$$

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

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296
Users Effect	0.86448

Figure 10 shows the estimated effect of user (b_u) building on the movie effects model above. Whilst b_u showed less variability than was observed with b_m, it was evident that adjusting for user effects enhanced the accuracy of the algorithm. Indeed, adjusting for user effects resulted in reaching both of the projects objectives. Thus, adjusting for both movie and user effects demonstrated the strong bias introduced by each of these variables on ratings. But can we do better?

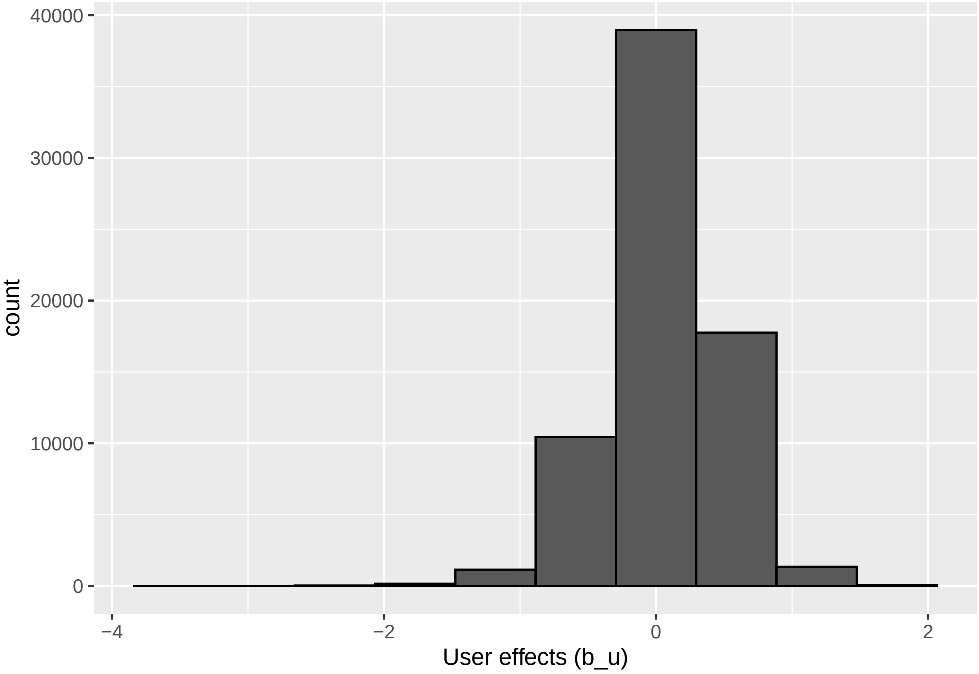

Figure 10: User Effect Distribution

3.2.4 Genre Bias

Movie ratings were also dependent on genre, with some genres achieving higher average ratings than others. Therefore, 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, $\hat{b}_g$ calculated using the formula shown below.

$$Y_{u,m} = \mu + b_m + b_u + b_g + \epsilon_{u,m}$$

$$\hat{b}_g = mean\left(\hat{y}_{u,m} - \hat{\mu} - \hat{b}_m - \hat{b}_u\right)$$

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296
Users Effect	0.86448
Genre Effect	0.86411

Figure 11 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 adjusting for genre, in addition to movie and user bias, was an RMSE of 0.86411. Thus adding genre effects into the model only pimproved thevaccuracy of the algorithm by very little, versus the previous model. Regardless, any incremental improvement is acceptable.

3.2.5 Release Year Bias

The exploratory analysis has shown a strong seasonal pattern between the release year of the movie and the number of ratings. The least squares estimate of the year effect, $\hat{b}_y$ calculated using the formula shown below, building on the algorithm developed already.

$$Y_{u,m} = \mu + b_m + b_u + b_g + b_y + \epsilon_{u,m}$$

$$\hat{b}_y = mean\left(\hat{y}_{u,m} - \hat{\mu} - \hat{b}_m - \hat{b}_u - \hat{b}_g\right)$$

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296
Users Effect	0.86448
Genre Effect	0.86411

Method	RMSE
Release Year Effect	0.86392

The year of movie release adds some additional variability to the average rating in the train set as shown in Figure 12. Indeed, incorporating this into the training algorithm yielded a RMSE of 0.86392 which is a modest improvement over the previous model.

3.2.6 Review Year Bias

In our exploratory analysis, we did identify a slight downwards trend during a 10yr period (1995 - 1005), which made us consider it as an effect that we would like our model to capture.

$$Y_{u,m} = \mu + b_m + b_u + b_g + b_y + b_r + \epsilon_{u,m}$$

$$\hat{b}_r = mean\left(\hat{y}_{u,m} - \hat{\mu} - \hat{b}_m - \hat{b}_u - \hat{b}_g - \hat{b}_y\right)$$

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296
Users Effect	0.86448
Genre Effect	0.86411
Release Year Effect	0.86392
Year Rated Effect	0.86383

As expected, the rating year had a small impact on ratings and this was confirmed by visualising the distribution of b_r in Figure 13.

3.3 Regularisation

The exploratory analysis showed that not only is the average rating affected by the movie, user, genre, year of release and date of review, but that the number of ratings also varies. Thus, for example, some movies and genres of movie received fewer ratings than others while some users provided fewer ratings than others. Similarly, the number of ratings varied by year of release and date of review. In each of these cases, the consequence of this variation is that the estimates of the effect (b) will have been subject to greater uncertainty when based on a smaller number of ratings.

Regularised regression is a machine learning algorithm which penalises parameter estimates which come from small sample sizes and are deemed to be somewhat unreliable [4].

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

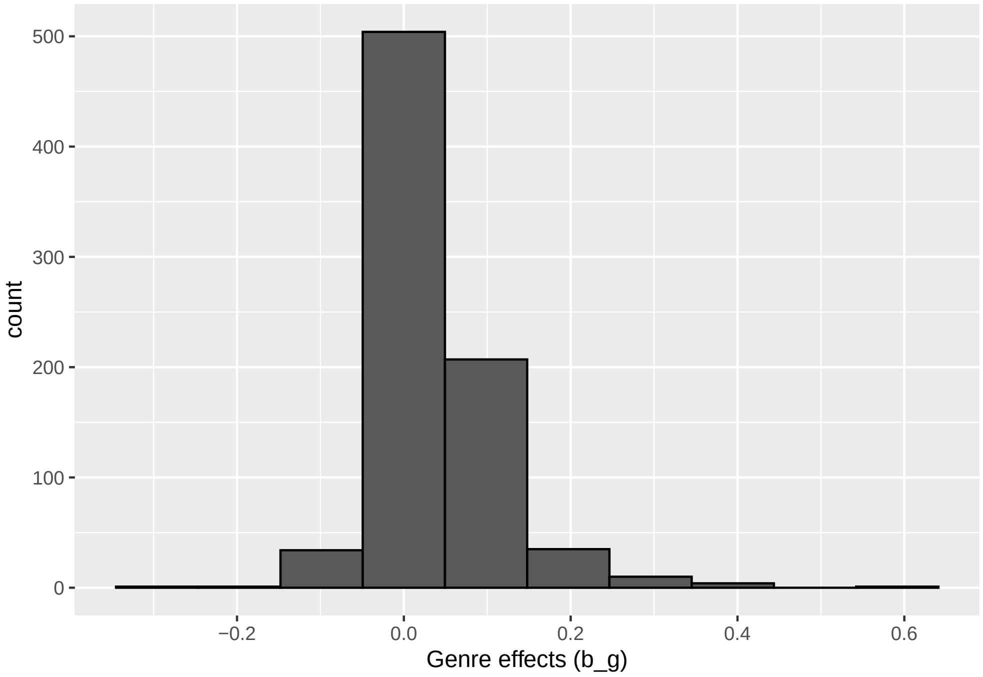

Figure 11: Genre Effect Distribution

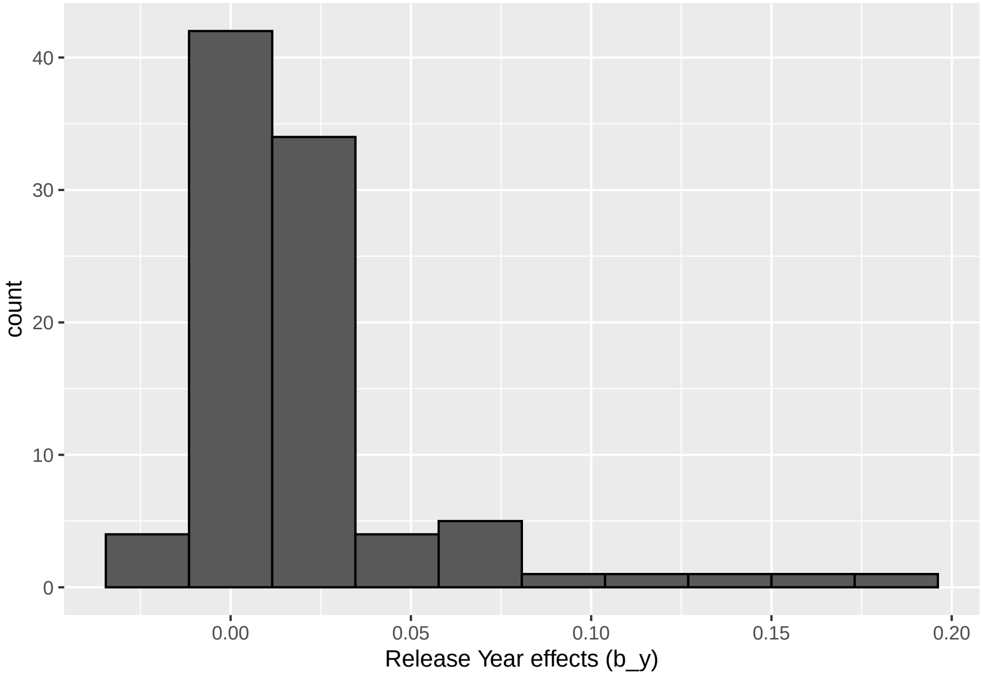

Figure 12: Release Year Effect Distribution

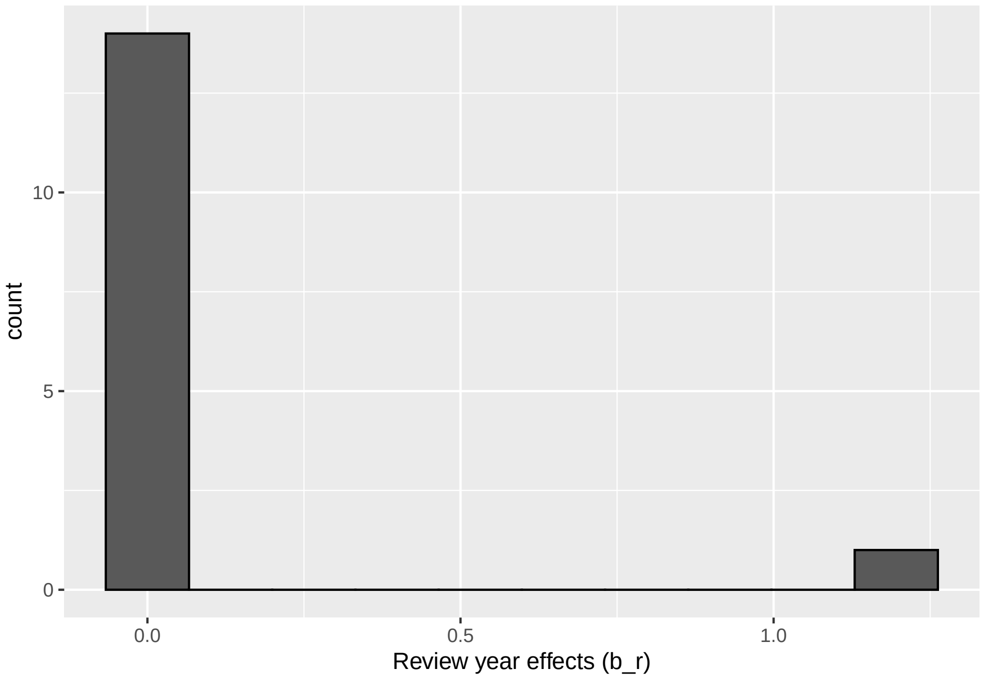

Figure 13: Year Rated Effect Distribution

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, i.e. n_i is a big 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.

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

The regularisation model we developed to adjust for all of the effects previously described, as shown below. A range of values for λ (4 - 6) with increments of 0.01 was applied in order to tune 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-training the model in the validation set.

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

Figure 14 shows the RMSE delivered across each of the λ tested. The optimum value for λ was 4.9 which minimised RMSE to 0.86354.

Method	RMSE
Target Objective	0.8712
Just the Average	1.06005
Movie Effect	0.94296

Method	RMSE
Users Effect	0.86448
Genre Effect	0.86411
Release Year Effect	0.86392
Year Rated Effect	0.86383
Regularised RMSE	0.86354

3.4 Final Validation

Now that our algorithm development has been completed, the final step is to train the algorithm using the entire edx dataset and then to predict ratings using the validation dataset which we will be using for the first time in our analysis. We will use the the optimal $\lambda = 4.9$ which we calculated during the regularisation step of the process and model all effects over the full edx data.

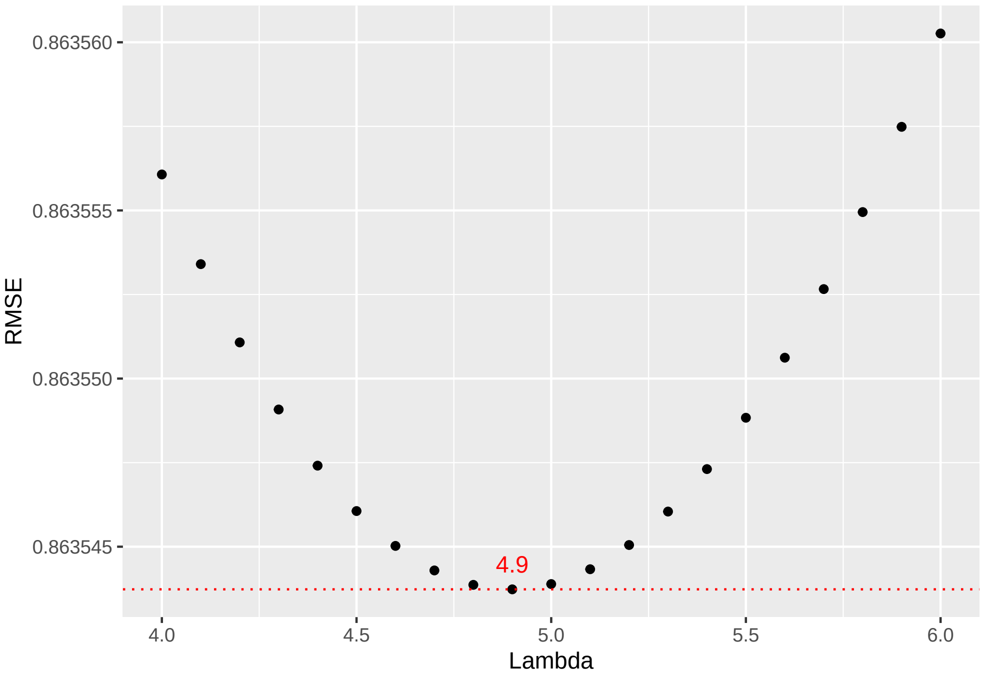

Figure 14: Lamda Optimisation

4 Results

The algorithm we've developed, trained and tested has achieved the two goals of our objective against the final hold-out (validation) dataset. More specifically, we have achieved a RMSE of 0.86416 which is a -0.19589 improvement over the naive algorithm "Just the Average".

4.1 Imrovement over the Naive Algorithm

Method	RMSE	Difference
Just the Average	1.06120181029262	-
RMSE Validated	0.86416	-0.19704

and we have achieved an improvement of format(round(RMSE_validated - target_rmse, 5), scientific = F) over the winning score of the Netflix challenge.

4.2 Improvement over the winning score of the competition

Method	RMSE	Difference
Target RMSE	0.8712	-
RMSE Validated	0.86416	-0.00704

5 Conclusion

The objective of this project was to develop a recommendation system using the MovieLens 10M dataset that predicted ratings with a residual mean square error of less than 0.8712 and an improvemnt of minimum 10% over the naive algorithm.. Adjusting for a number of estimated biases introduced by the movie, user, genre, release year and review date, and then regularising these in order to constrain the variability of effect sizes, met the project objective goals yielding a model with an RMSE of 0.86354. This was confirmed in a final test using the previously unused validation dataset, with an RMSE of 0.86416.

Although the algorithm developed here met the project objective goals it still includes a sizeable error loss, not all of which may be considered truly independent, something that's justified by slightly worse performance of the algorithm against the final validation test vs the test set used during its development. We conclude that there is still room for accuracy improvement of the recommendation system with techniques that can account for some of this non-independent error. One such approach is matrix factorisation, a powerful technique for user or item-based collaborative filtering based machine learning which can be used to quantify residuals within this error loss based on patterns observed between groups of movies or groups of users such that the residual error in predictions can be further reduced [4].

6 References

[1] Schrage, 2017, title= Great Digital Companies Build Great Recommendation Engines , url= https://hbr.org/2017/08/great-digital-companies-build-great-recommendation-engines , journal= Harvard Business Review , publisher= Harvard Business School Publishing , author= Schrage, M. , year= 2017 , month= Aug

[2] Schrage, 2018, title= How Marketers Can Get More Value from Their Recommendation Engines , url= https://hbr.org/2018/06/how-marketers-can-get-more-value-from-their-recommendation-engines , journal= Harvard Business Review , publisher= Harvard Business School Publishing , author= Schrage, M. , year= 2018 , month= Jun

[3] Lohr, 2009, title= Netflix Awards $1 Million Prize and Starts a New Contest , url= https://bits.blogs.nytimes.com/2009/09/21/netflix-awards-1-million-prize-and-starts-a-new-contest , journal= The New York Times , publisher= The New York Times , author= Lohr, S. , year= 2009 , month= Sep

[4] Irizarry, 2020, title= Introduction to data science: data analysis and prediction algorithms with R , url= https://www.crcpress.com/Introduction-to-Data-Science-Data-Analysis-and-Prediction-Algorithms-with/Irizarry/p/book/9780367357986 , publisher= CRC Press , author= Irizarry, Rafael A. , year= 2020

[5] glen_2020, title= RMSE: Root Mean Square Error , url= https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/ , journal= StatisticsHowTo.com: Elementary Statistics for the rest of us! , author= Glen, Stephanie , year= 2020 , month= Jul