Introduction

Modern computing and artificial intelligence capabilities allow digital services to deploy products which understand and cater to their users. A familiar example is the movie recommendation system offered by the streaming service Netflix, which recommends users content based on their preferences and viewing habits. Netflix awarded a prize of one million dollars in 2009 to a team of data scientists who were able to improve the accuracy of their recommendation system by 10%, underscoring the great appetite for such work.

This project aims to develop a movie recommendation system using the popular MovieLens data which consists of approximately 10 million movie ratings. The data was partitioned into a training set (edx) and a final test set (final hold-out) by the course instructors. A goal was set of achieving a root mean square error (RMSE) less than 0.86490 against the movie ratings listed in the final hold-out dataset.

In this report, exploratory data analysis and visualisation are performed followed by algorithm development. Discussion of the algorithm and its components, along with conclusions, are included.

The report was compiled using R Markdown in RStudio, an integrated development environment for programming in R, a language and software environment for statistical computing.

Exploratory Analysis

The edx dataset created consists of 9,000,055 rows and 8 columns, with ratings provided by a total of 69,878 unique users for 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. Therefore, this dataset includes many missing values.

Exploratory analysis: Ratings column

In our edx dataset, the overall average rating was 3.51. The distribution of ratings included in the dataset (Figure 1) shows the most common rating across movies was 4, and that overall, whole star ratings were more common than half star ratings.

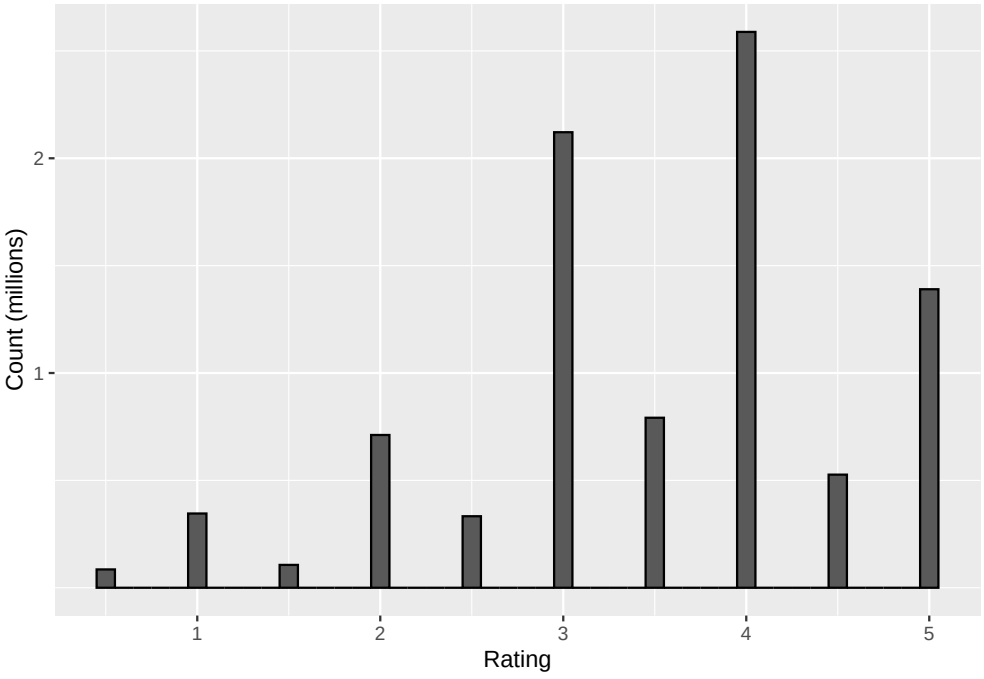

Figure 1: Overall ratings distribution

Exploratory analysis: Movies column

Due to their inherent quality or other factors, some movies received higher or lower ratings than others, regardless of user (see Figure 2). Analysis reveals there is significant variation in numbers of ratings received by each movie (Figure 3), with the movie receiving the most ratings, Pulp Fiction (1994), receiving 31362 ratings. There is a clear movie effect influencing the rating awarded, and it will be important to adjust for this in the recommendation algorithm.

Exploratory analysis: User column

User data showed an additional effect, with particular users assessing films in more or less generous fashion (Figure 4). Some users also contributed more ratings than other users (Figure 5). Clearly, there is a user effect which needs to be accounted for in the recommendation system.

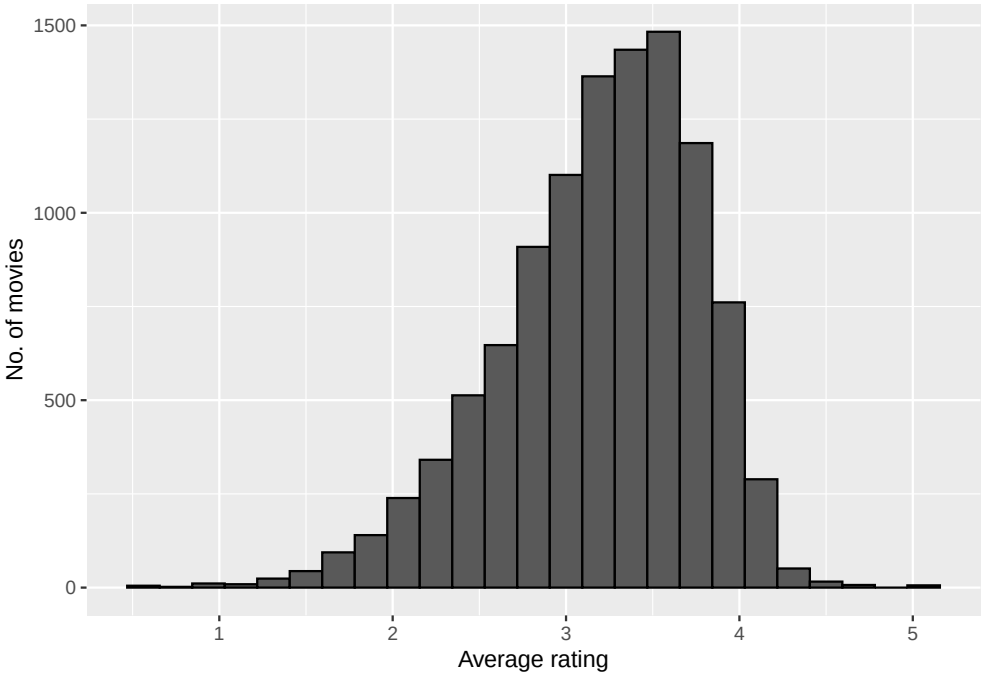

Figure 2: Movie by average rating

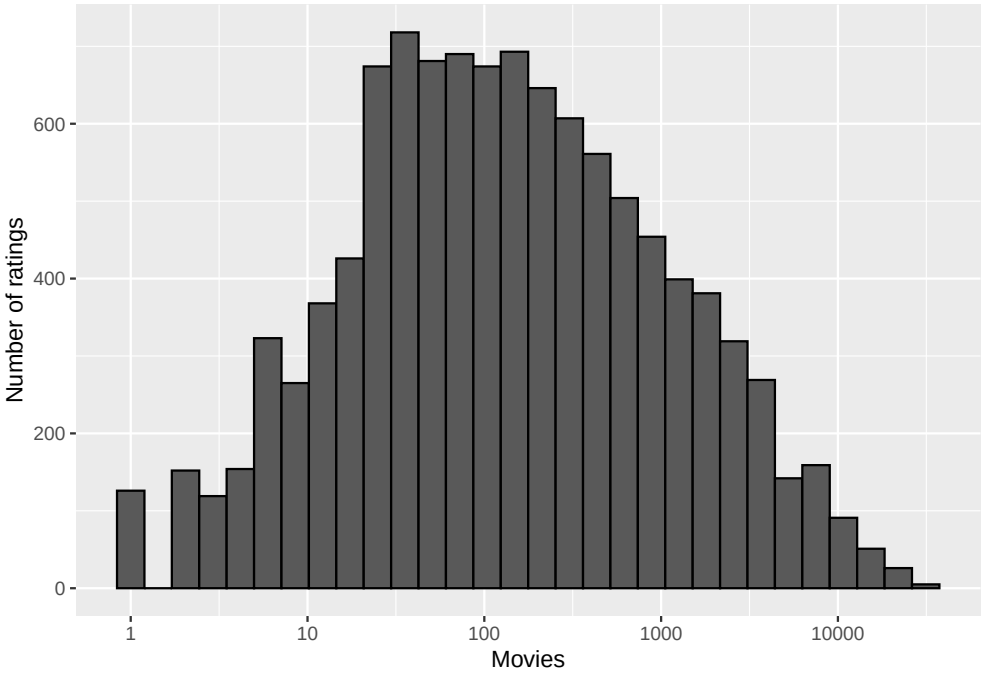

Figure 3: Number of ratings by movie

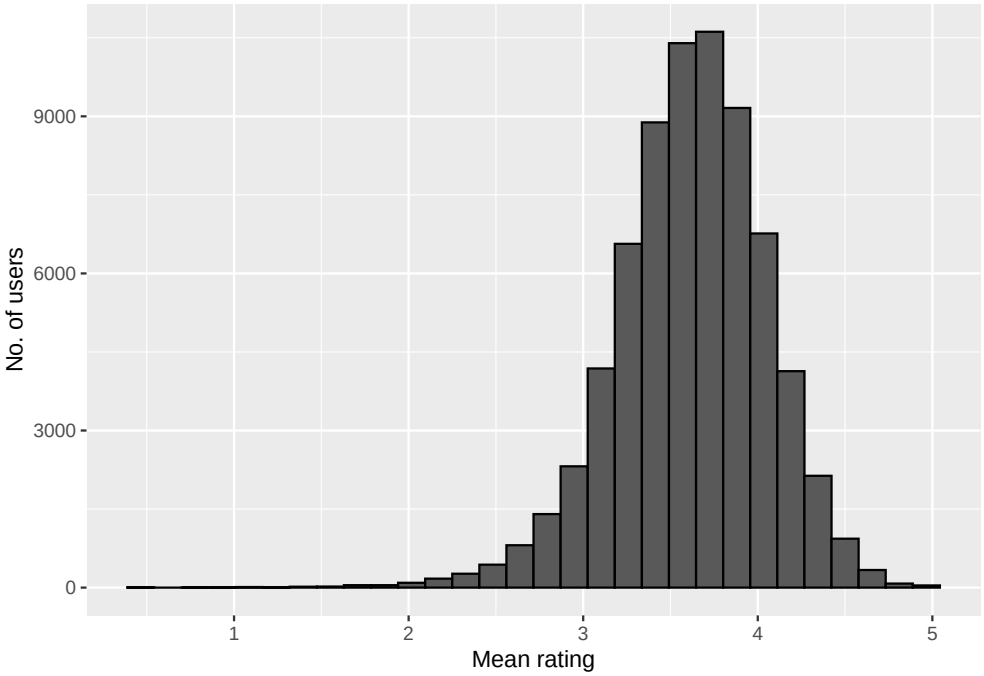

Figure 4: Mean rating by user

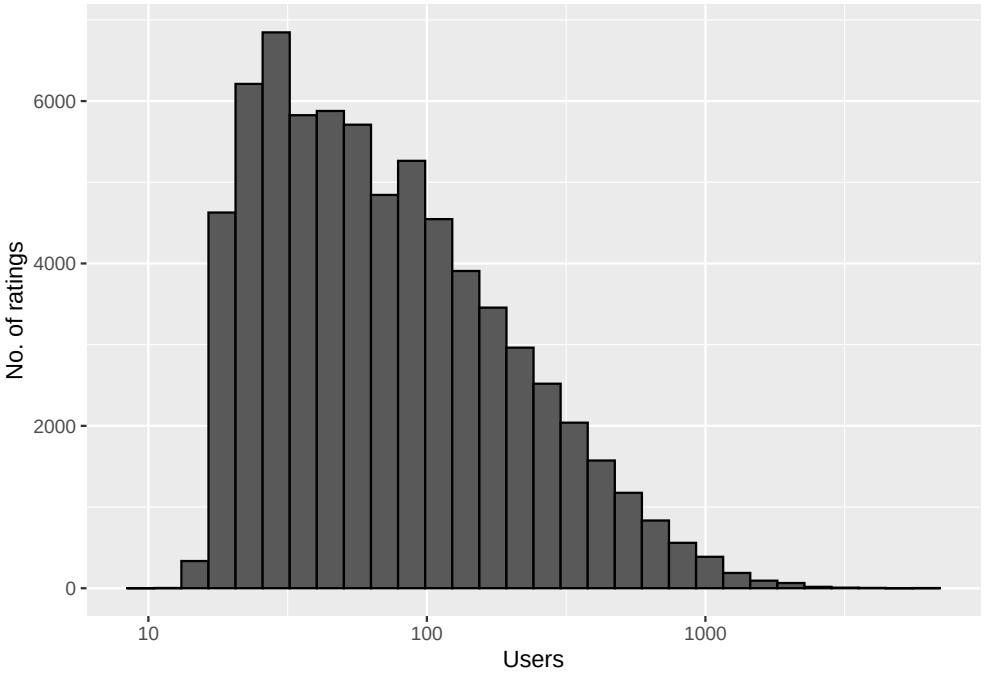

Figure 5: No. of ratings by user

Exploratory analysis: Genre column

The genre variable assigns each movie rating a set of genres with which it can be identified. Many movies were assigned to multiple genres. It was possible to identify 20 different genres and these were ranked by no. of ratings (Table 2).

Table 1: Individual genres ranked by number of ratings

Genre	No. of Ratings	Avg. 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

Comedy and drama movies received the most ratings whereas IMAX movies received the fewest. Table 2 also demonstrates there was variation in the average rating by genre. Data were grouped by genre combination and filtered for those combinations with at least 100,000 ratings. This analysis shows a clear effect of genre on rating (Figure 6). This genre effect should therefore be included as a predictor in the movie recommendation system.

Exploratory analysis: Title columns

The title variable includes both the title of the movie and the year of release, in brackets. Table 3 shows the top 10 movie titles by the number of ratings.

Exploratory analysis: Release year

Figure 7 explores for any effect of release year on mean rating. Average rating did vary by release year. There was a peak in average rating for movies released from 1940 to 1950, and average rating has declined for movies released since.

As demonstrated in Figure 8, there were few ratings assigned to movies released prior to 1970. Movies with the greatest number of ratings were released during the 1990s, peaking in 1995. Therefore, release year will be an important factor to account for in the recommendation algorithm, with the caveat that small sample sizes may impact the reliability of this effect prediction for some films.

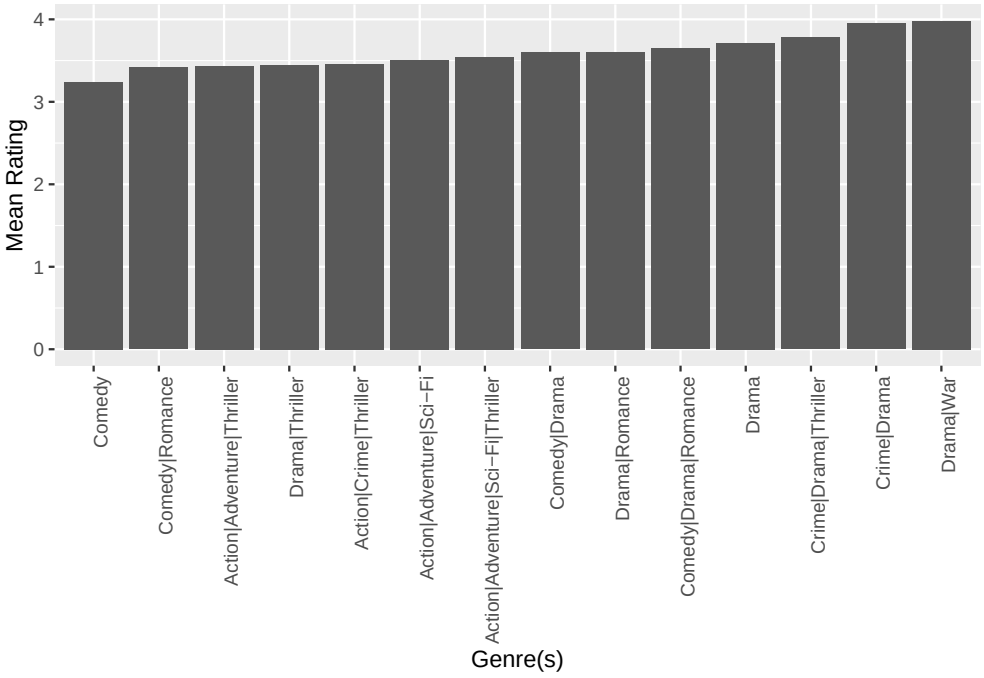

Figure 6: Average rating by genre

Table 2: 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

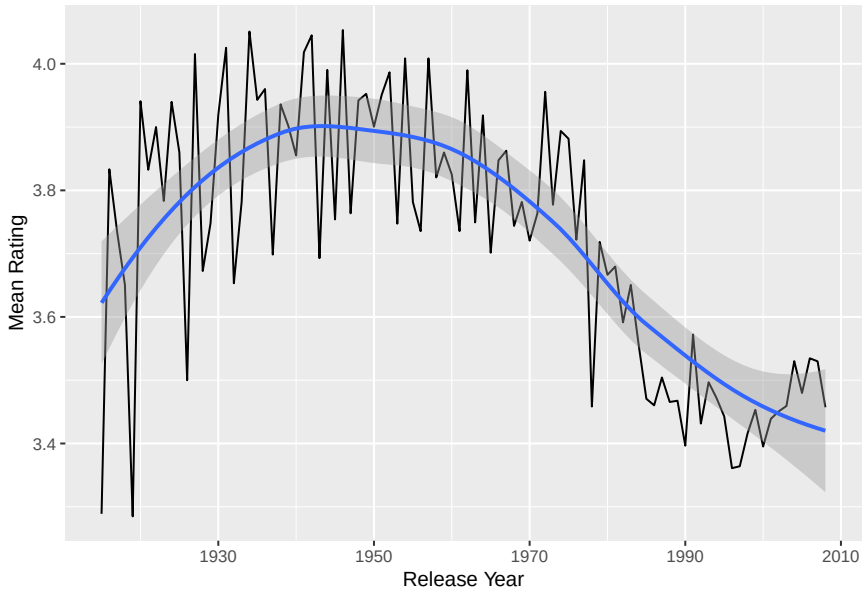

Figure 7: Mean rating by year of release

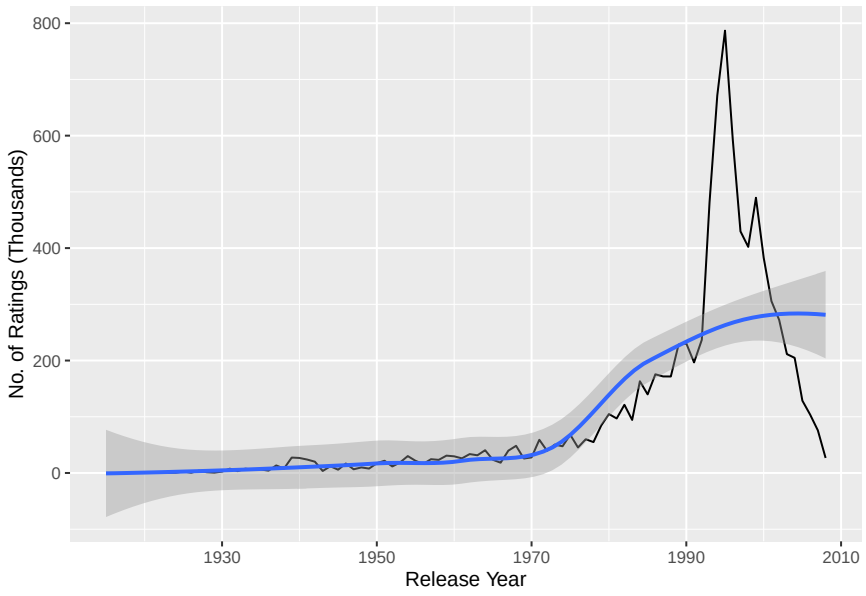

Figure 8: No. of ratings by release year

Exploratory analysis: Review date

To analyse the effect of date of review and rating, timestamp data was mutated into standard date format, omitting time and rounding to nearest week (so as to smooth the data.

Ratings in the dataset were highest in 1995, when the first review was submitted. There has been a gradual decline in rating since, until 2005 when average rating began once more to increase. The observation of this trend justified its inclusion in the recommendation algorithm.

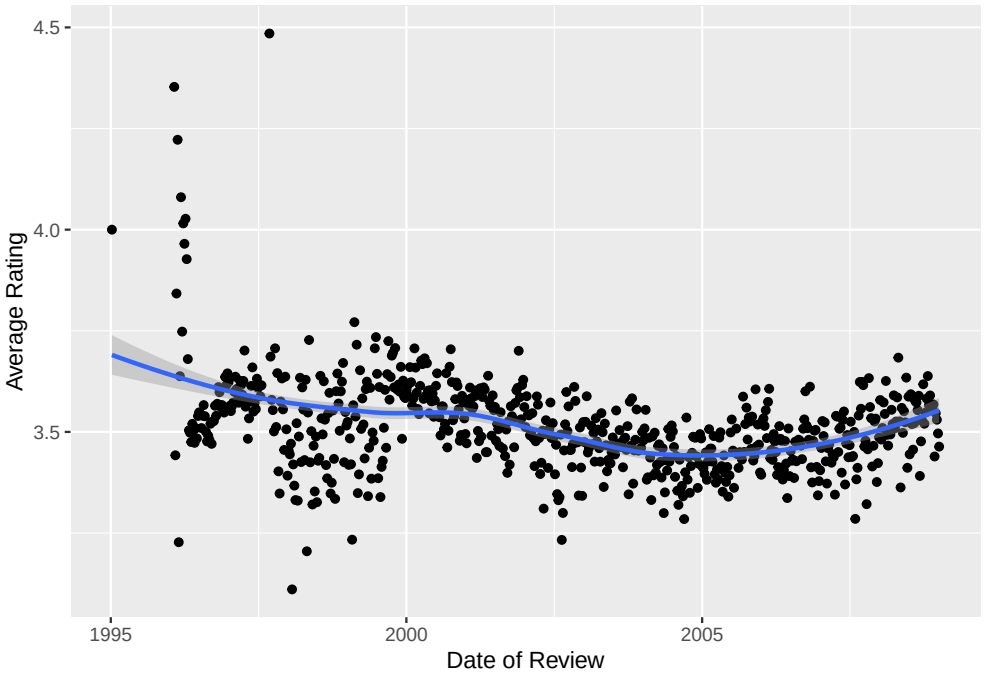

Figure 9: Average rating by date of review

Methods

Splitting edx out into train and test sets

Since the final hold-out data was to be withheld for final tesing of the model, the edx dataset was used both for training and testing of the model during its development. Partitioning this data during development would be crucial to allow for refinement and to prevent overtraining.

The caret package and the 'createDataPartition' function was used to partition the edx data into a train and test set. Dplyr table join functions 'semi-join' and 'anti-join' were further used to ensure that the test set and train set include the same users and movies.

Calculating error loss

Residual mean square error (RMSE) was used to represent error loss between ratings predicted by the algorithm and actual ratings derived from the test set. RMSE is defined as the standard deviation of prediction errors, where these are a measure of the spread of data points from the regression line. This project's objective was to develop an algorithm that would achieve an RMSE less than 0.86490.

Method	RMSE	Difference
Project objective	0.8649	-

Algorithm development

A simple algorithm to predict movie rating would apply the same rating to all films. The rating for the movie i by the user u ($Y_{u,i}$) was the sum of the true rating μ plus $\epsilon_{u,i}$, being the independent errors sampled across this distribution.

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

It is logical that the mean of all ratings in the dataset would minimise the RMSE. Therefore, $\hat{\mu}$ = mean(train_set$rating) was the formula that was used to train the first implementation of the algorithm.

In the previous exploratory analysis section, it was established that ratings were not evenly assigned across all movies. This movie effect b_i should be incorporated into the algorithm to improve the accuracy of the prediction.

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

The least squares estimate of the movie effect $\hat{b}_i$ was derived from the average $Y_{u,i} - \hat{\mu}$ of each movie i. The following formula was thus incorporated into the prediction algorithm:

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

Exploratory analysis showed a clear user effect, ie. that different users tended toward patterns of different ratings. The least squares estimate of user effect $\hat{b}_u$ was calculated as below:

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

A clear genre effect b_g was also observed in exploratory analysis performed, where some genres tended to receive higher or lower ratings than others. The least squares estimate of this genre effect $\hat{b}_g$, was calculated as below and incorporated into the model:

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

Release year effect b_y was an additional factor influencing the average rating received by movies in our exploratory analysis. The least squares estimate of this effect $\hat{b}_y$ was incorporated into the model as below:

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

An effect of review date b_r on rating was also observed in exploratory analysis. To incorporate this effect into the model, smoothing by rounding date to the nearest week was required. The least squares estimate of this effect was incorporated into the model as below:

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

Regularising the algorithm

A final factor to account for in the algorithm is the variance in the number of ratings given to films for different effects. Some films received more or less ratings, with the variance depending as well on release year and review date. Estimates of the effect b were therefore subject to differing levels of certainty depending on the number of ratings received. Regularisation was applied to the algorithm in order to weight effect estimates by their sample size. A penalty term λ was defined by cross validation using the edx dataset. Rather than the ridge regularisation discussed in the course material, lasso regularisation was used. This was applied to the movie effect b_i per the below formula:

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

Validating the final model

With the algorithm developed, the final holdout data was then used to validate he recommendation system. Dplyr functions were first required to ensure that the review date and release year variables were present in the final holdout dataset.

The final regularised model with optimised λ was then applied to predict ratings from the final holdout dataset and an RMSE was calculated.

Results

Algorithm 1: Simple average

The simple approach of predicting an average rating from the train set for each entry in the testing set resulted in an RMSE of 1.060799. This RMSE did not meet the project objectives, indicating that further refinement was required.

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899

Algorithm 2: Adjustment for movie effects

There was a large amount of variation in movie effect between different movies in the dataset, as demonstrated in Figure 10. Adding the movie effect to the algorithm yielded an improved RMSE of 0.944578, still higher than the objective for this project.

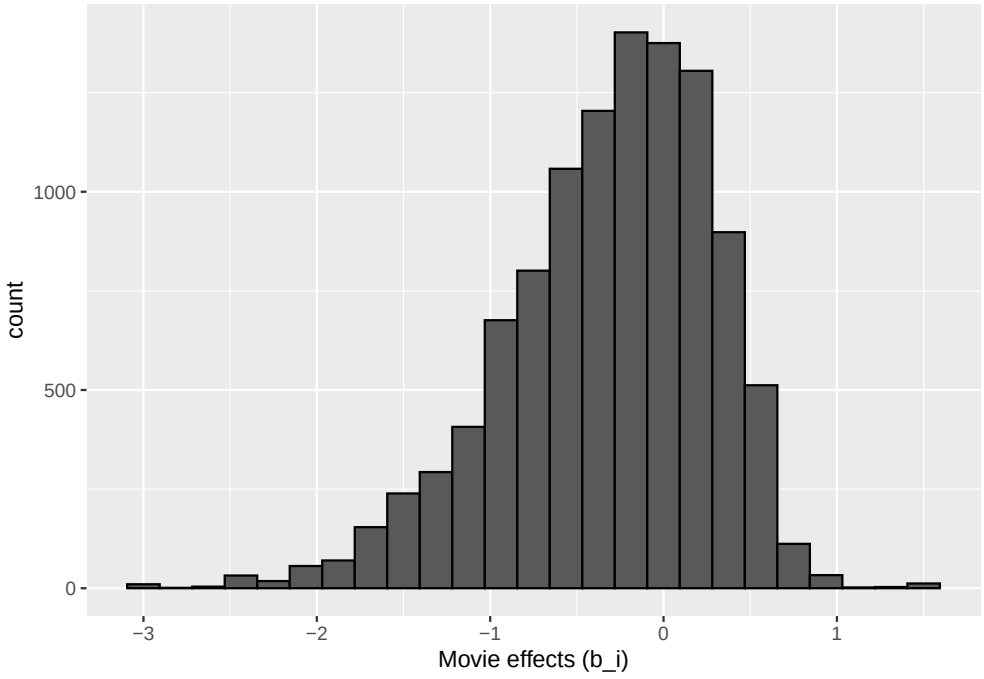

Figure 10: Movie effect distribution

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678

Algorithm 3: Adjustment for user effects

User effect was an additional source of variability accounted for in the model, as demonstrated in Figure 11. Adjusting for both the user and movie effect improved the RMSE of the algorithm by 18.01%, thereby indicating that these two factors contributed a large part of overall variability to the dataset.

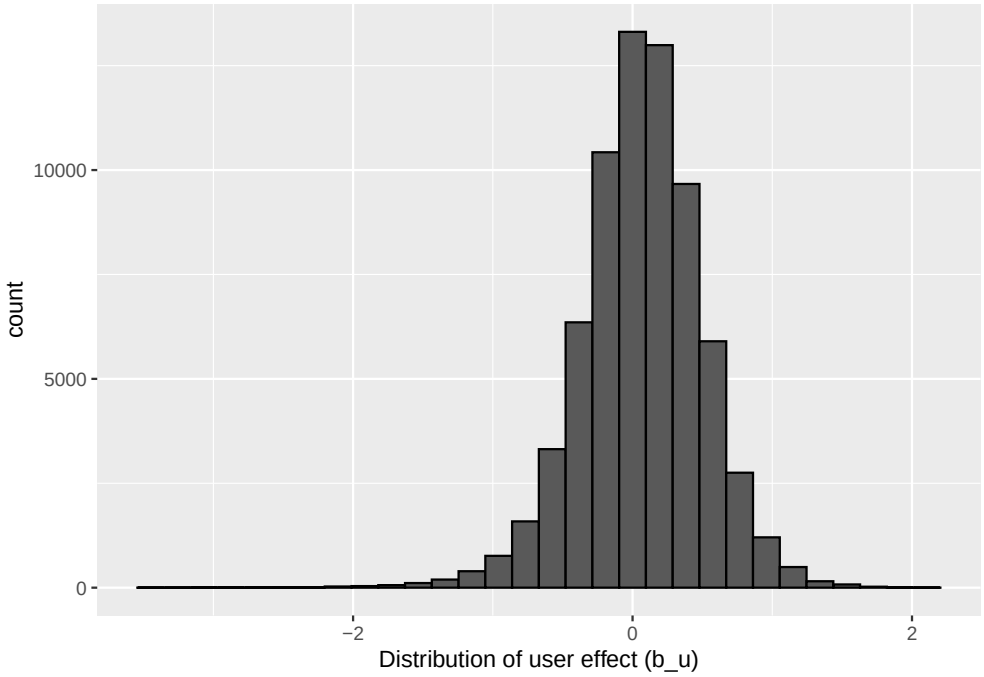

Figure 11: Distribution of user effects

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678
Movie + User effects (b_u)	0.869761	0.004861

Algorithm 4: Adjusting for genre effects

As can be seen in Figure 12, genre was an additional source of variability on the data. When this factor was adjusted for in the algorithm with the inclusion of b_g, in addition to user and movie effect, an RMSE of 0.869394 was achieved. Though a comparatively modest improvement, this does bring the algorithm closer to meeting the project objective RMSE.

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678
Movie + User effects (b_u)	0.869761	0.004861
Genre, movie, and user effect (b_g)	0.869394	0.004494

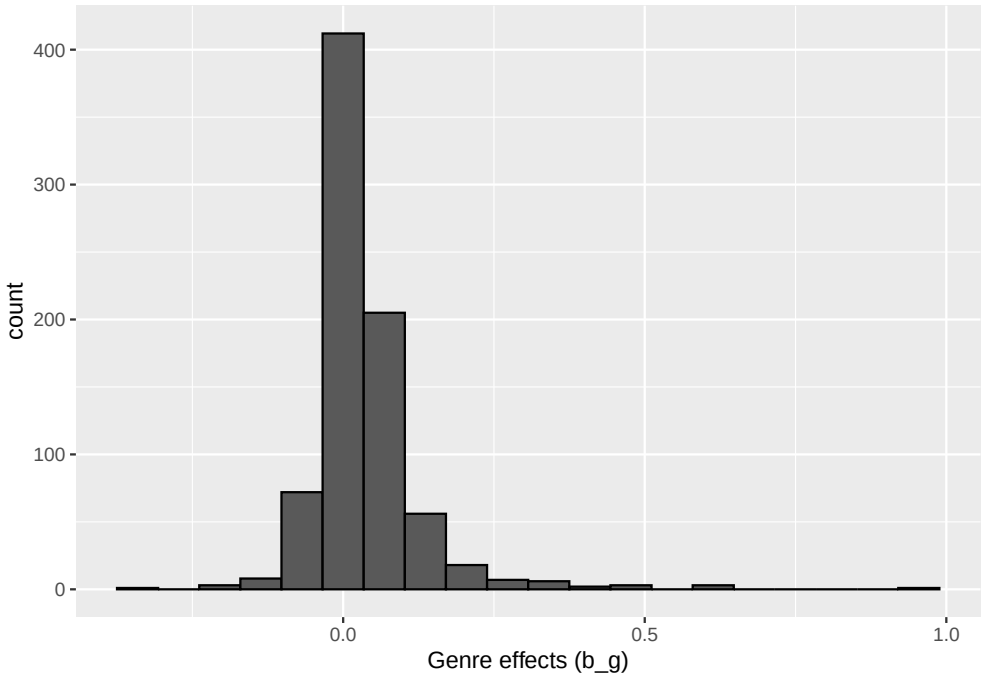

Figure 12: Distribution of genre effects

Algorithm 5: Adjustment for release year effects

Exploratory analysis showed that the year of movie release was an additional source of modest variability to the dataset (Figure 13). Compensating for this effect in the prediction algorithm yielded a small improvement in RMSE, 0.869241. This figure was close to the project objective.

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678
Movie + User effects (b_u)	0.869761	0.004861
Genre, movie, and user effect (b_g)	0.869394	0.004494
Movie, User, Genre and Year effects (b_y)	0.869241	0.004341

Algorithm 6: Adjustment for review date effect

Exploratory analyis showed that the date of review was an additional factor influencing the rating given by a user (Figure 14). When this effect was incorporated into the predicition algorithm, an improvement of 18.08% was observed in the RMSE, yielding an RMSE of 0.869059. This result was still short of the project objective.

Algorithm 7: Regularisation

Exploratory analysis showed that there was a degree of variability in the number of ratings each movie received in the dataset. This variability of sample size added instability to the accuracy of the algorithm.

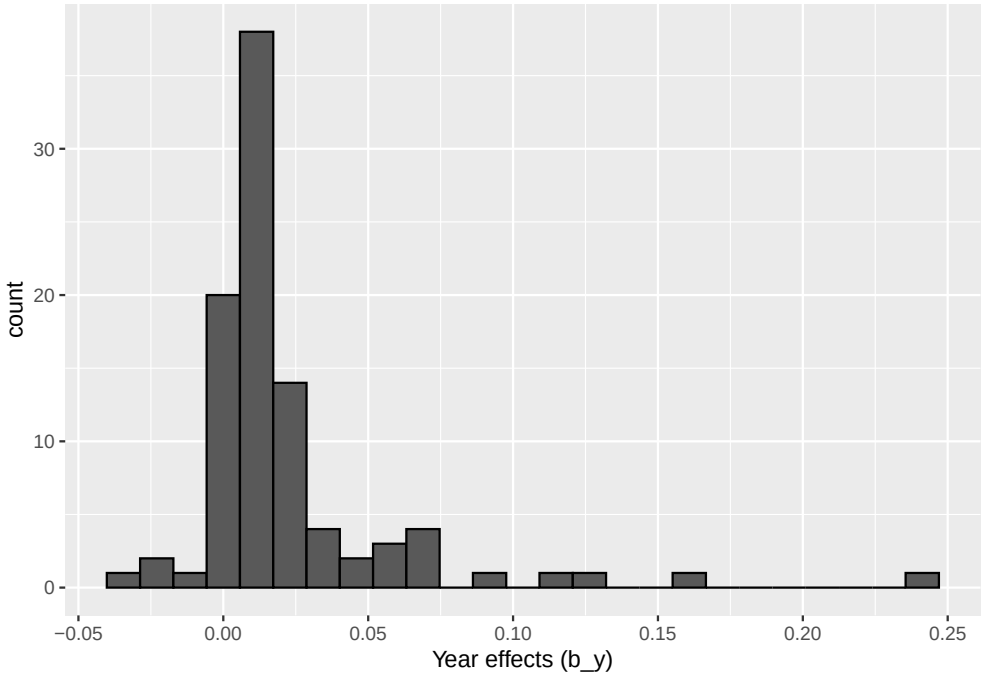

Figure 13: Distribution of release year effect

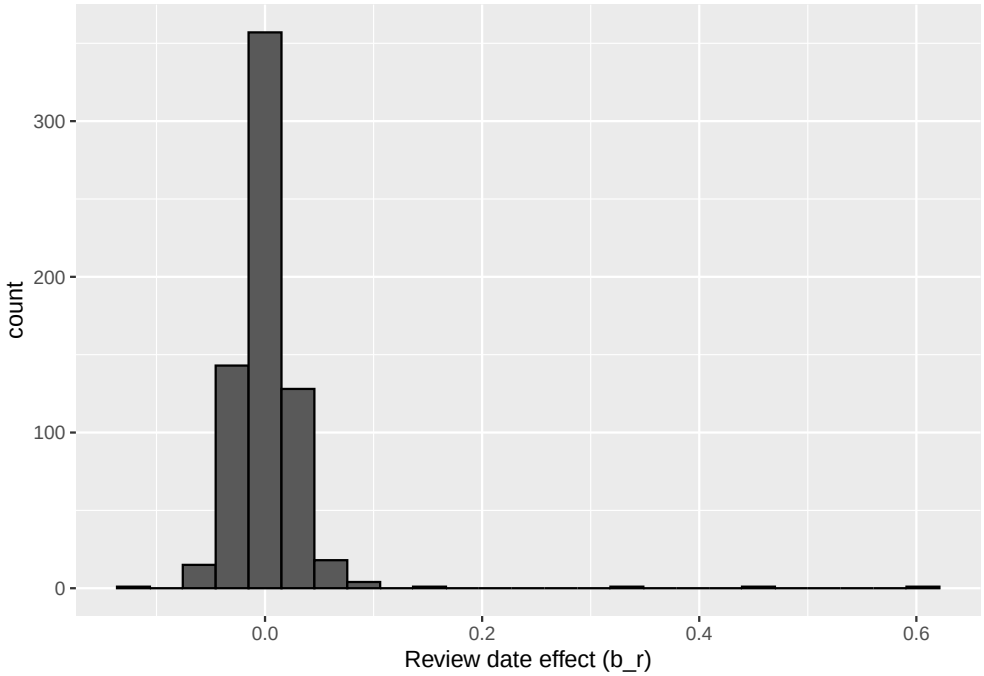

Figure 14: Distribution of review date effects

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678
Movie + User effects (b_u)	0.869761	0.004861
Genre, movie, and user effect (b_g)	0.869394	0.004494
Movie, User, Genre and Year effects (b_y)	0.869241	0.004341
Movie, User, Genre, Year and Review Date effects (b_r)	0.869059	0.004159

Regularisation was applied, tuned with numerous values of the parameter λ to minimise RMSE (Figure 15). The RMSE achieved with this regularised model was 0.86744.

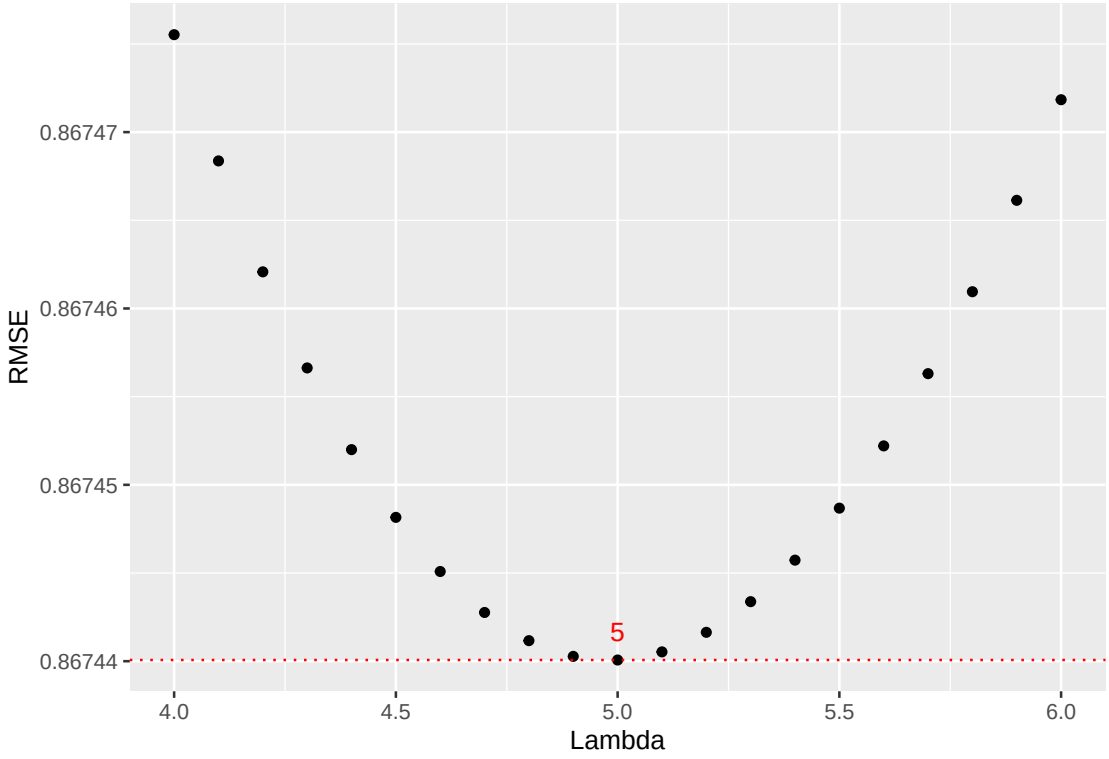

Figure 15: Tuning parameters for lambda

Final hold-out test dataset

To confirm the validity of the algorithm developed here, it was used to predict outcomes in the final hold-out dataset prepared at the beginning of the project. An RMSE of 0.864054 was achieved.

Method	RMSE	Difference
Project objective	0.8649	-
Simple average	1.060799	0.195899
Movie effects (b_i)	0.944578	0.079678
Movie + User effects (b_u)	0.869761	0.004861
Genre, movie, and user effect (b_g)	0.869394	0.004494
Movie, User, Genre and Year effects (b_y)	0.869241	0.004341
Movie, User, Genre, Year and Review Date effects (b_r)	0.869059	0.004159
Regularised Movie, User, Genre, Year and Review Date effect	0.86744	0.00254

Method	RMSE	Difference
Project objective	0.86490	-
Final model holdout	0.864054	-0.000846

Conclusion

This project aimed to build a recommendation system that could predict movie ratings in the MovieLens database. RMSE was used to measure the accuracy of the prediction algorithm when measured against actual ratings from a final holdout dataset, and a target RMSE of less than 0.8490 was set.

Exploratory data analysis identified effects (or biases) in the dataset introduced by different movies, users, genres, release years, and review dates. These biases were accounted for in the prediction algorithm and lasso regularisation applied to reduce variability introdcued by small sample size effects. A final RMSE of 0.86744 was achieved against the test data, and validated against the final holdout test set with an RMSE of 0.864054.

Though the algorithm developed here did meet the stated project objective, there are numerous avenues suggesting further work to improve its functionality. Matrix factorisation is one such approach, which would allow multifactorial filtering and weighting of data in response to patterns observed in various effects. Due to its relatively low processing demands, matrix factorisation would likely be the next step in improving this algorithm.

Experimenting with unsupervised machine learning techniques, such as Principal Component Analysis (PCA) or Random Forest (RF) would be a further step toward optimising the performance of the algorithm. These techniques were not attempted here due to their relatively high processing and memory requirements, but could yield substantial improvements to RMSE obtained.