MovieLens Recommendation System Report

Sreeja Vadakke Veettil

08/03/2022

Executive Summary

This report is related to the Capstone project of the HarvardX-PH125.9x:Capstone course. The aim of this project is to create a movie recommendation system using the MovieLens data set. The report starts with a general introduction to the project followed by a description of the data set. A preliminary data analysis is carried out initially to better understand the parameters to be included in the machine learning algorithm. This is followed by a description of the methods used to develop the algorithm and the results from each iteration of the algorithm, before deciding on the final algorithm. The final test using Matrix Factorization with Stochastic Gradient Descent method on the validation data set achieved an RMSE of 0.78697, with an improvement of ~26% with respect to the mean algorithm.

Contents

Introduction

Recommendation systems are among the most important and popular applications of machine learning algorithms that help users discover new products and services. It is a subclass of information filtering system whose main aim is to predict the "rating" or "preference" a user would give to an item. Based on the ratings, items are recommended to the user for which a high rating is predicted. Companies such as Amazon, Netflix and Google use these recommendation systems to better understand their customers and target them with products more effectively. In 2009, Netflix awarded a million dollars to a team of data scientists who successfully met the challenge of improving the effectiveness of its movie recommendation algorithm by 10%.

The aim of this Capstone project is to develop a movie recommendation system using one of the MovieLens data sets, collected and made available by GroupLens Research. To facilitate the algorithm development, the Movielens data set is initially split into a training set (edx) and a final hold-out test set (validation). The algorithm is developed using the edx data set and then used to predict the movie ratings in the validation data set. The Root Mean Squared Error (RMSE) is used to evaluate how close the final algorithm predictions are to the actual values in the validation data set.

Preliminary Data Analysis

The edx data set consists of 9,000,055 rows and 6 columns, with ratings provided for a total of 10,677 unique movies by a total of 69,878 unique users. Each row in the edx data set represents a rating given by one user to one movie. If each unique user provided a rating for each unique movie, the data set would have a total of 746,087,406 ratings. As the total number of rows in the edx data set is less than this number, it clearly suggests that every unique user has not rated every unique movie.

Ratings

The overall average rating in the edx data set is 3.512. The minimum and maximum rating awarded to any movie is 0.5 and 5 respectively. Figure 1 shows the distribution of the total ratings included in the edx data set. It is clear that the most common rating across all movies is 4, and that, overall, whole star ratings (79.5%) are used more often than half star ratings (20.5%).

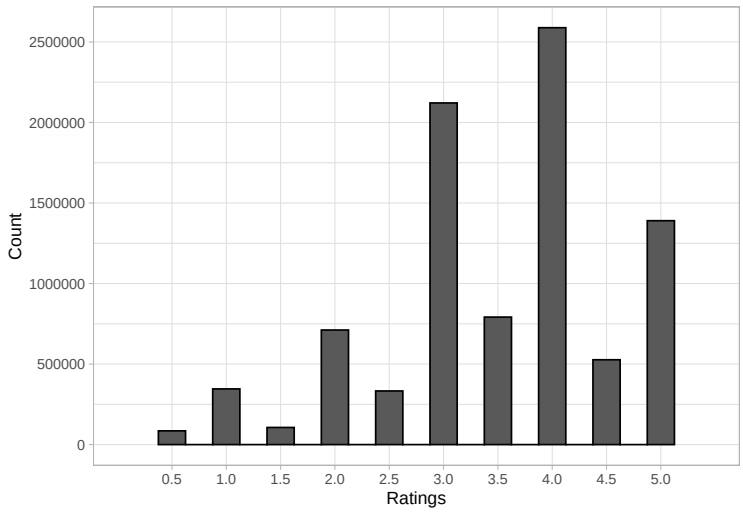

Figure 1: Overall ratings distribution

Movies

The distribution of the number of ratings by movie is shown in Figure 2. It can be observed that the distribution is right skewed, suggesting that there is a significant variation in the number of ratings received by each movie. The movie with the most ratings is Pulp Fiction, receiving a total of 31,362 ratings, whereas as many as 126 movies are rated only once. This is expected, as many movies are watched and rated only by few users, while blockbusters are more widely watched and rated. On average, a movie gets 843 ratings by the user. There are 8,553 movies (~80%) with ratings less than the average number of ratings. Prediction of actual ratings for movies with smaller number of ratings may pose a challenge.

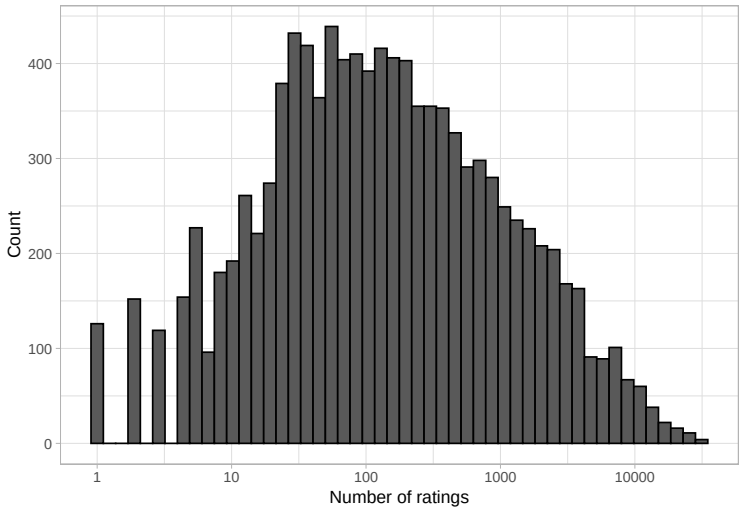

Figure 2: Number of ratings by movie

Figure 3 shows the distribution of average ratings by movie, which clearly indicates that some movies are more highly rated than the others. The distribution is near normal and slightly skewed to the left. It can be observed that ~62%(6,609) of the movies have been awarded a rating between 3 and 4. This suggests a movie effect on the ratings awarded and adjusting for this effect will be a worthwhile inclusion in the algorithm.

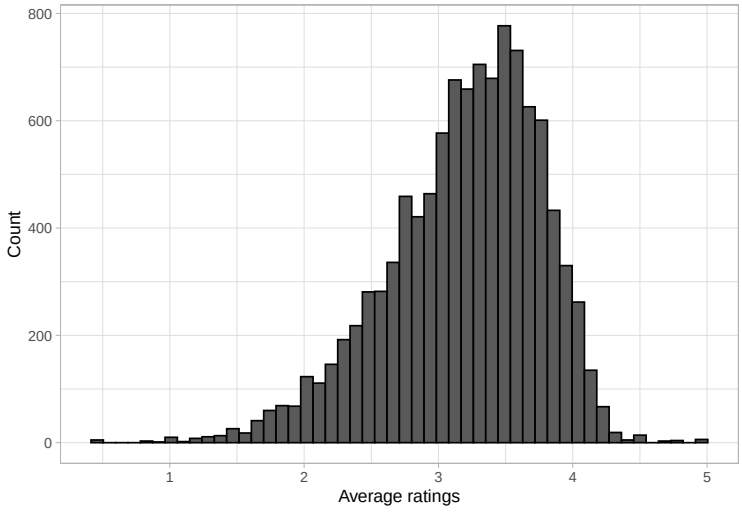

Figure 3: Movie distribution by average rating

Users

The distribution of number of ratings by user is shown in Figure 4. It can be observed that the distribution is skewed to the right, i.e. not every user is equally active and some users contributed with ratings more than others. The average number of ratings a user gives is 129, while the highest number of ratings is 6,616 and as many as 1,059 users provided fewer than 10 ratings each. Users with the number of ratings higher than the average of all users is only ~27%. The majority of users (61%) have rated only between 20 and 100 movies.

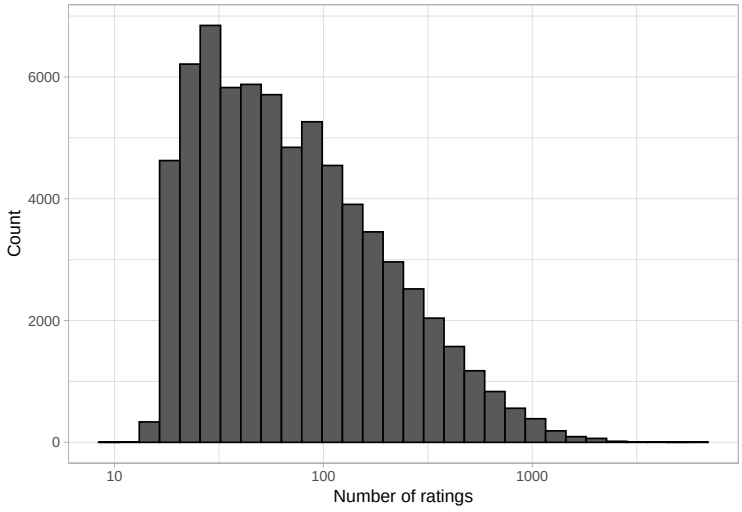

Figure 4: Number of ratings by user

Figure 5 shows the distribution of average ratings by user. The distribution is near normal with a pattern similar to that observed for movies, with some users providing higher ratings than others. From Figure 5, it can be observed that ~74%(51,751) of the users awarded a movie rating between 3 and 4. This suggests an user effect, which adjusted for may improve the accuracy of the algorithm.

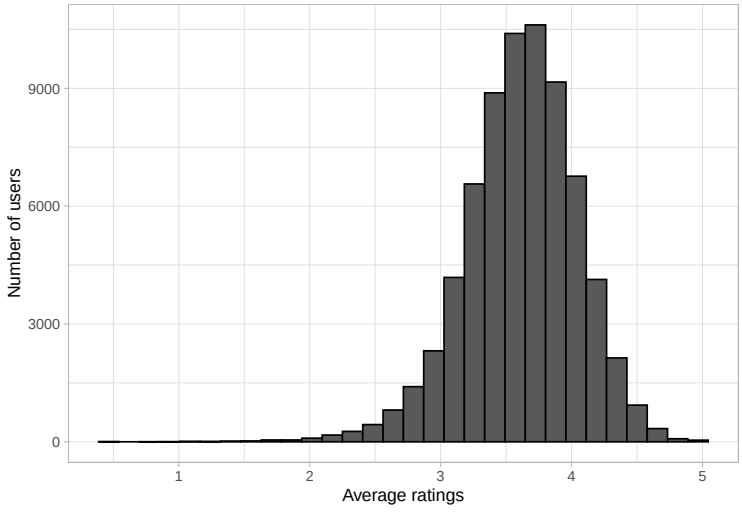

Figure 5: User distribution by average rating

Movie Genres

The information on the movie genres is provided in the MovieLens data set as a combination of different classifications. For example, Boomerang in the first row of the edx data set includes Comedy|Romance in the 'genres' variable. Some movies are assigned to more than one genre and there are a total of 797 genre combinations in the data set. The distribution of the number of ratings by genre combinations is shown in Figure 6, which clearly suggests that some genre combinations are getting more ratings than others. To simplify the analysis, grouping the data by genre combinations and filtering those with at least 50,000 ratings, the average ratings by genre combinations is shown in Figure 7.

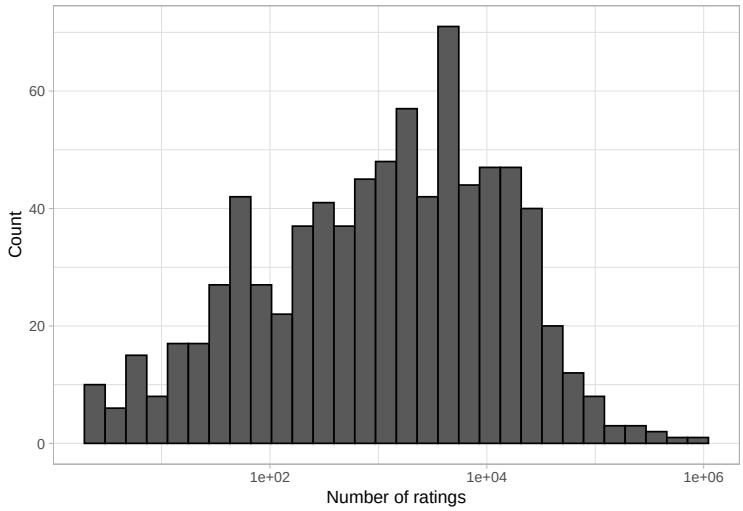

Figure 6: Number of ratings by genre

A genre effect can be observed from Figure 7, with 'Horror' and 'Children|Comedy' movies achieving the lowest average ratings whereas 'Comedy|Crime|Drama' and 'Drama|War' movies achieving the highest ratings. However it is difficult to infer a distinct effect of the genres on the average ratings, as each movie can have a unique combination of different genres and each genre can be associated with different movies. For example, movies involving 'Comedy' genre have average ratings of 2.9, 3.2, 3.5 and 4. This suggests that the 'genres' variable in the current format may not be suitable to be incorporated in the algorithm, as it might lead to overfitting. A possible solution would be to separate the genre combinations.

By separating the genre combinations into rows with individual genres, it is possible to identify 20 different genre categories (including 'no genre listed') which are listed in Table 1 by the number of ratings and number of movies. It is to be noted that as one movie might belong to more than one individual genre, the movies in Table 1 are not distinct, the total number of movies in Table 1 (21,557) is greater than total number of unique movies (10,677) in the edx data set. So if one row is considered as one record, the above splitting of the genre combinations actually duplicated each record into multiple ones, depending on the genre combinations for each movie.

It can be noticed from Table 1 that 'Drama' and 'Comedy' movies have the most number of ratings whereas 'Documentary' and 'IMAX' movies have the fewest ratings. However, this doesn't necessarily mean that users prefer to rate the 'Drama' and 'Comedy' movies over other types, rather this reflects the fact that there are more movies in these genres, as evident from Table 1, compared to others. The last genre type '(no genres listed)' listed in Table 1 is not really a genre, but that for 7 ratings of one single movie, the genres information is not provided. The top 5 popular genres 'Drama', 'Comedy', 'Thriller', 'Action' and 'Adventure' appear 61% in the edx data set.

The distribution of the average ratings by individual genres with at least 50,000 ratings is shown in Figure 8. A clear effect of genres can be observed with 'Horror' movies achieving the lowest average rating and

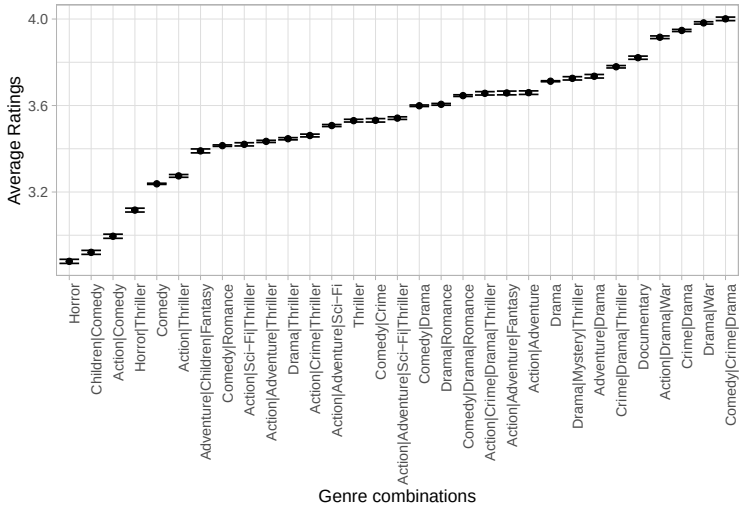

Figure 7: Average ratings by genre

Table 1: Individual genres ranked by number of ratings and movies

Genre	No. of ratings	No. of movies
Drama	3910127	5336
Comedy	3540930	3703
Action	2560545	1473
Thriller	2325899	1705
Adventure	1908892	1025
Romance	1712100	1685
Sci-Fi	1341183	754
Crime	1327715	1117
Fantasy	925637	543
Children	737994	528
Horror	691485	1013
Mystery	568332	509
War	511147	510
Animation	467168	286
Musical	433080	436
Western	189394	275
Film-Noir	118541	148
Documentary	93066	481
IMAX	8181	29
(no genres listed)	7	1

'Film-Noir', 'Documentary' and 'War' movies achieving the highest rating. On comparing the two top rated genres from Table 1, namely 'Drama' and 'Comedy', it can be observed that the former has a higher average rating as compared to the latter. There are differences in the average ratings between the individual genres and the standard deviation within each genre appears to be small. This suggests that the movie genre effect is also an important parameter and addressing this effect will improve the accuracy of the algorithm. However because most of the genres have an average rating within 1 star of each other, this parameter is not expected to affect the algorithm as much as the movie or user effects. Moreover, it is also important to take into account the greater uncertainty in point estimates created by small sample sizes for some genres.

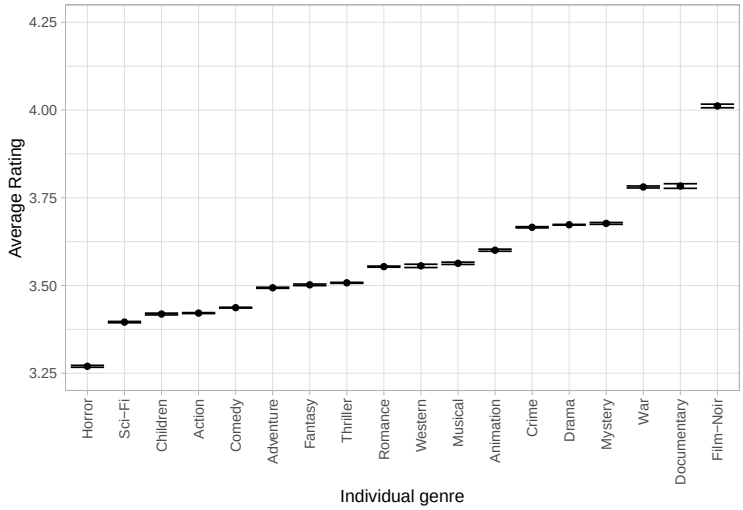

Figure 8: Average ratings by individual genre

Movie Release year

In order to explore the effect, if any, of the movie release year on ratings, the 'title' string in the edx data set is split into two columns, namely the 'title' and the 'release year'. Figure 9 shows the number of ratings by release year. The earliest and latest movie release year included in the edx data set is 1915 and 2008 respectively. It can be observed from Figure 9 that from 1950 there has been an exponential growth in the number of released movies. It is also clear that very few ratings are assigned to movies released prior to 1970. The movies with the most ratings are released during the 1990s, peaking in 1995 with ~9% of the total number of ratings included in the edx data set.

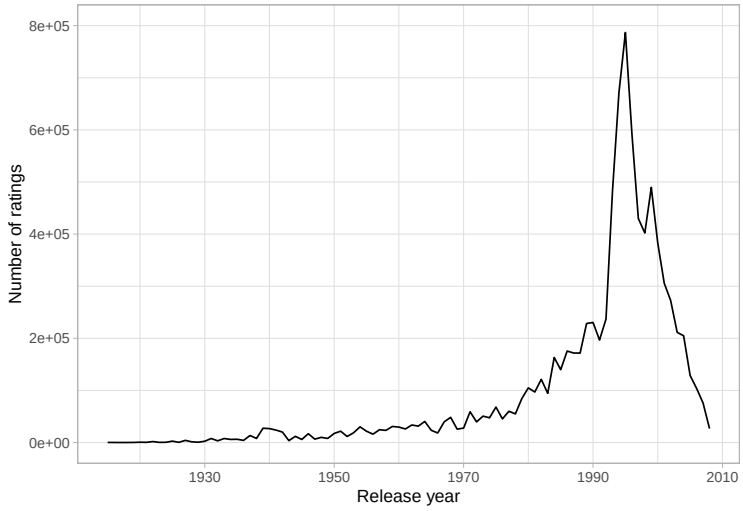

Figure 9: Number of ratings by release year

The distribution of the average ratings by movie release year is shown in Figure 10. It can be observed that movies released prior to 1970 appear to get higher average rating as compared to those released since that period. The average rating for a movie released between 1930s and 1960s is ~3.889, compared to a rating of ~3.489 for movies released beyond 1980. This could be attributed to the fact that perhaps the reviewers are viewing older movies that are more critically acclaimed and considered as classics. The movie release year

does seem to have an effect on the ratings awarded and adjusting for this effect could improve the accuracy of the algorithm. However as with movie genres, taking into account the greater uncertainty in point estimates created by the small sample sizes in some years would also be important.

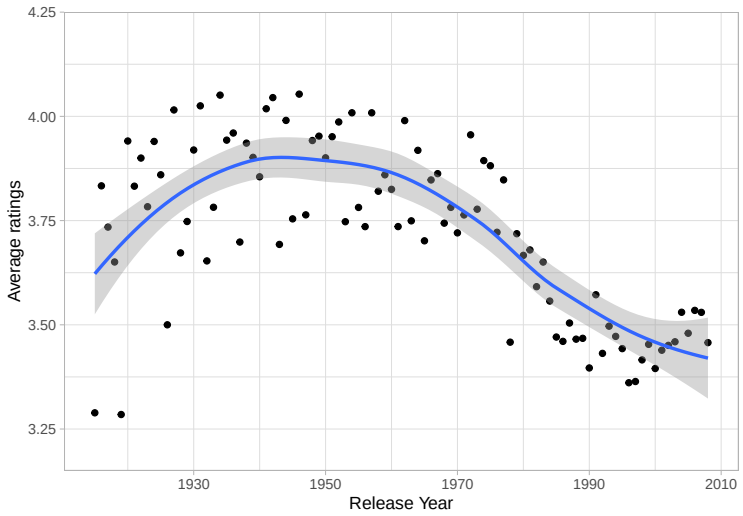

Figure 10: Average rating by release year

Date and time of review

The 'timestamp' variable in the edx data set records both the date (yymmdd) and time (hhmmss) information, based on an epoch (time zero) starting midnight on 1 January 1970. To analyse the effect, if any, of review date and time on ratings, the 'timestamp' variable is mutated into date and time format, and rounding to the nearest week. The earliest and latest review included in the data set was completed in 1995 and 2009 respectively. Figure 11 shows the distribution of the average ratings by review date. It can be noticed that the average ratings are highest (~3.567) during 1995-1998, beyond which there is only a minor variation. The effect of review date on the average ratings is small relative to that observed for movies, users, genres and release year, suggesting that including this parameter may only slightly improve the algorithm accuracy.

The distribution of the average ratings by review time is shown in Figure 12. It can be observed that the average ratings are highest (~3.533) during the post midnight hours (0-1 hrs) beyond which there is a very minor variation. The effect of the review time on the average ratings is negligible compared to that observed for movies, users, genres and release year, thus suggesting that including this variable may not largely improve the accuracy of the algorithm.

Method and Analysis

Splitting edx data set into train and test sets

The edx data set is used to both train and test the algorithm under development, as the validation data set is used only for the final test. By following the technique as applied to the Movielens data set to create the edx and validation data sets, the edx data set is split into train (80%) and test (20%) sets. It is ensured that the test set only includes userIds and movieIds that are present in the train set.

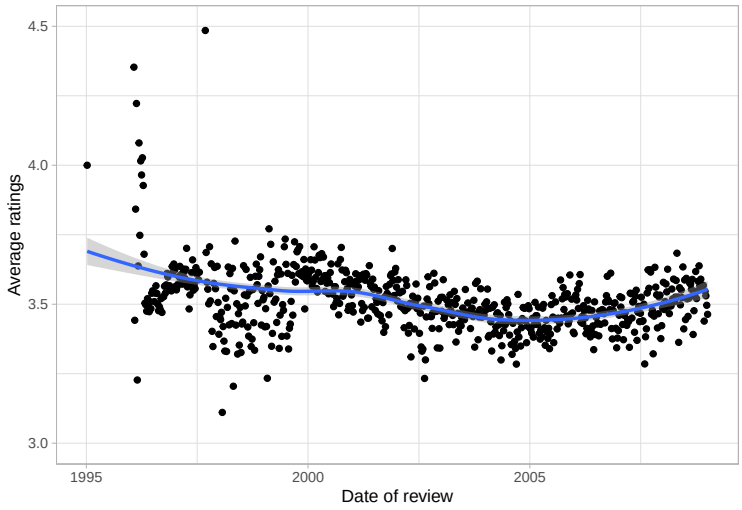

Figure 11: Average rating by date of review

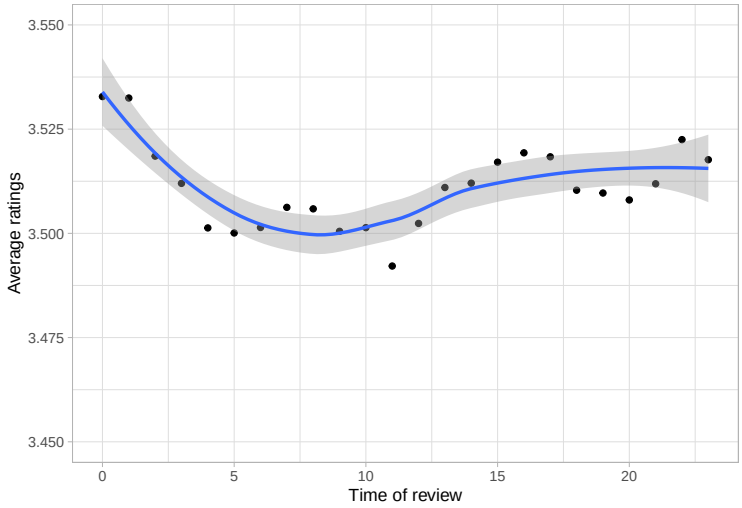

Figure 12: Average rating by time of review

Algorithm accuracy estimation

The goodness of the developed machine learning algorithm is evaluated using the Root Mean Squared Error (RMSE), defined as the standard deviation of the residuals, where residuals are a measure of how far the data points are from the regression line. The RMSE is a measure of the differences between the actual ratings in the test set and the predicted ratings from applying the algorithm. In the formula shown below, the actual and predicted ratings provided by the user u for a movie i are defined as $y_{u,i}$ and $\hat{y}_{u,i}$ respectively, and N is the total number of user/movie combinations.

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

The objective of this project is to develop an algorithm to predict movie ratings as close as possible to the actual values in the validation data set, i.e. achieve an RMSE as small as possible.

Algorithm development using edx data set

Mean algorithm

The most simplest form of an algorithm for predicting the movie ratings is to apply the same rating to all the movies, assuming that the movie to movie variation is the randomly distributed error. The actual rating by a user u for a movie i, $Y_{u,i}$, is the sum of this "actual" rating, μ, plus $\epsilon_{u,i}$, defined as the independent errors sampled for the same distribution centered at 0.

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

The least squares estimate of μ, which is the mean of all ratings is the estimate that minimizes the RMSE. Any value other than the mean increases the RMSE, so this is a good initial estimation. Thus, $\hat{\mu}$ = mean(train_set$rating) is the most simplest form of the algorithm.

Method	RMSE
Mean algorithm	1.0599

Using the mean rating from the train set (3.512) for every entry in the test set resulted in an RMSE of 1.0599. As the RMSE value is larger than 1, it suggests that the predicted ratings are more than 1 star away from the actual ratings.

Movie effects

The preliminary analysis in the previous section highlighted that ratings are not assigned equally across all the movies, i.e. some movies are generally rated higher than the others. Therefore accounting for the movie effects will improve the accuracy of the algorithm. The mean algorithm is refined by adding the term b_i, to represent the average rating for a movie i and the least squares estimate of the movie effects, $\hat{b}_i$ is calculated as the mean of the difference between the observed rating and the mean as shown in the below formula.

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

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

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374

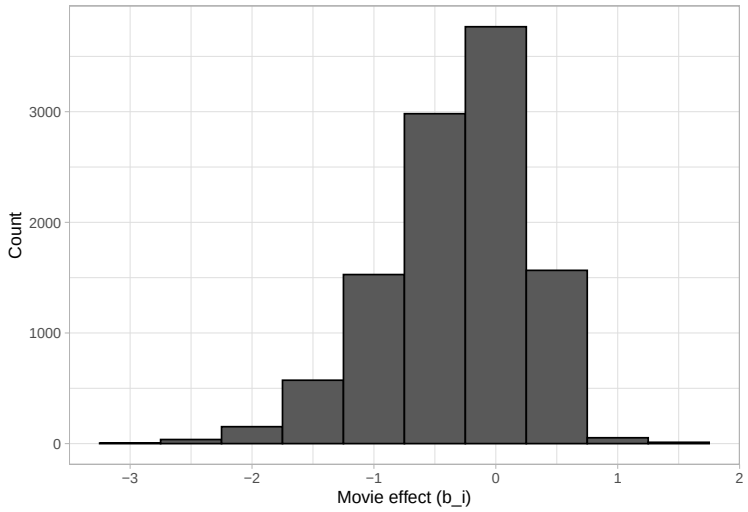

Figure 13: Movie effects distribution

Figure 13 shows the distribution of the estimate of movie effects (b_i). It is clear that the distribution is left skewed towards negative rating effect and that this estimate varies substantially across all of the movies included in the train set. Incorporating this into the algorithm yields an RMSE of 0.94374, thus improving the accuracy of the mean algorithm by 10.96%.

User effects

Similar to the movie effects, there is a substantial variability across users as well, i.e. different users rated movies differently. The algorithm is further refined by adjusting for the user effects (b_u) and as with the movie effects, the least square estimates of the user effect, $\hat{b}_u$ is calculated using the formula shown below:

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

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

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593

The distribution of the estimated user effects (b_u) building on the movie effects algorithm is shown in Figure 14. It is evident that the distribution is more symmetric and that b_u shows less variability as compared to b_i. Adjusting for both the movie and user effects resulted in an RMSE of 0.86593, with an improvement of 18.30% as compared to the mean algorithm. The improvement in RMSE demonstrates the strong effects introduced by each of these parameters on ratings.

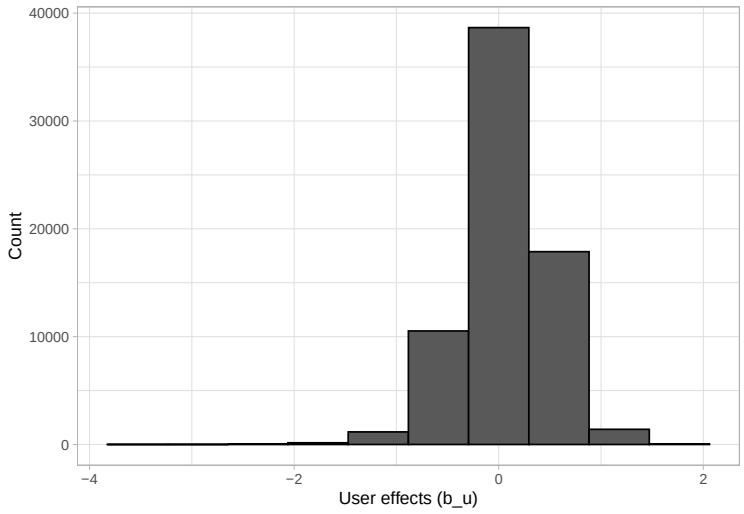

Figure 14: User effects distribution

Movie genre effects

The preliminary analysis also revealed the dependence of movie ratings on genres, with some genres achieving higher average ratings than others. This effect was observed even when movies were allocated to multiple genres. Therefore, the rating for each movie and user was further refined by adjusting for the combined genre effects, b_{gc}, and the least squares estimate of the genre effects, $\hat{b}_{gc}$ is calculated using the below formula.

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

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

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559

Figure 15 shows the distribution of estimates of the combined genre effects, b_{gc} in the train set. The distribution is right skewed and the variation across different genre combinations is quite evident from this figure. The algorithm RMSE after adjusting for the combined genre effects, in addition to the movie and user effects, is 0.86559. Thus including the effects of combined genres as a parameter in the algorithm only provided a moderate improvement in the accuracy of the algorithm, reducing the RMSE by 0.04% versus the previous algorithm and 18.33% versus the mean algorithm.

The preliminary analysis revealed a clear effect of the individual genres on the average ratings, so the genre combinations under the 'genres' variable in the test and train sets are separated into rows with individual genres. The effect of individual genres on the movie ratings is evaluated by following an approach similar to that applied with genre combinations. The rating for each movie and user is refined by adjusting for the individual genre effects, b_g, and the least squares estimate of $\hat{b}_g$ is calculated as shown below.

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

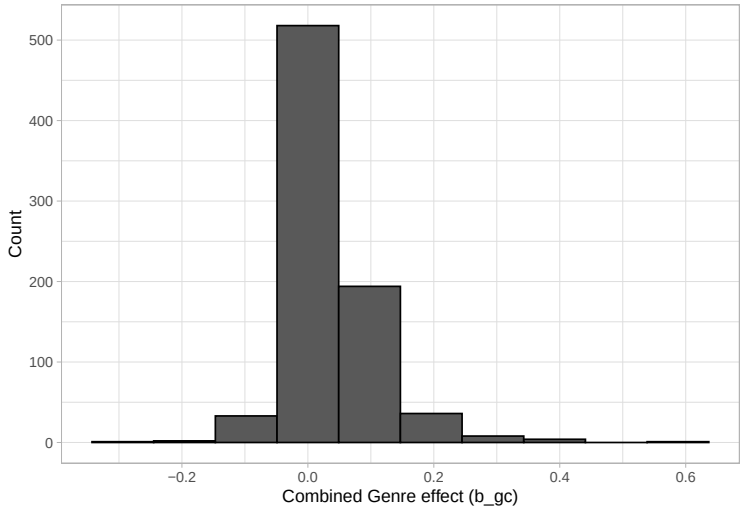

Figure 15: Combined Genre effects distribution

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

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419

The distribution of estimates of the individual genre effects in the train set is shown in Figure 16. Similar to the combined genres, the distribution is right skewed and the variation across different individual genres is evident from this figure. However, the estimates of individual genre effects show less variability compared to those of combined genre effects. The algorithm after adjusting for the individual genre effects in addition to the movie and user effects results in an RMSE of 0.86419. Thus including the effects of individual genres as a parameter in the algorithm provided an improvement in the algorithm accuracy, reducing the RMSE by 18.47% versus the mean algorithm. Also, it is evident that including the effects of individual genres as a parameter improves the accuracy of the algorithm by 0.16% versus the algorithm incorporating the combined genres. Therefore, the algorithm with the individual genre effects is used for the analysis presented below.

Release year effects

The preliminary analysis revealed a moderate effect of the movie release year on the ratings. The algorithm is refined by taking into account the effect of the release year, b_y, and the least squares estimate of $\hat{b}_y$ is calculated as shown below.

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

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

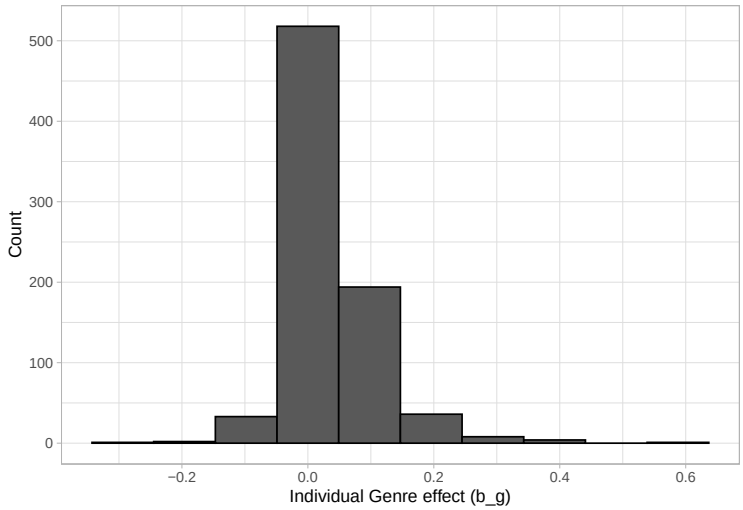

Figure 16: Individual Genre effects distribution

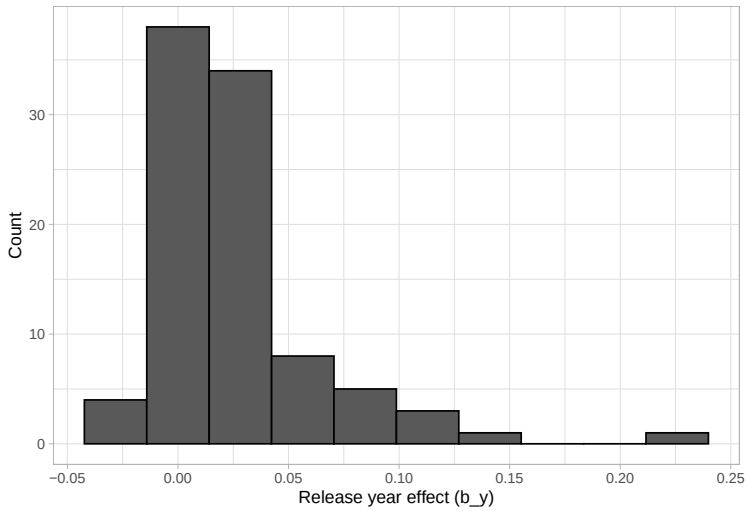

Figure 17: Movie release year effects distribution

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388

The distribution of the movie release year effects (b_y), shown in Figure 17, is right skewed and the year of movie release adds only moderate additional variability to the average rating in the train set. Incorporating this parameter resulted in a modest incremental improvement of 0.04% in the accuracy of the algorithm.

Review date effects

As noted in the previous section, there was a minor effect of the date of review (b_r) on the average ratings. The most appropriate way to incorporate this parameter into the algorithm would be to apply a smooth function. This data smoothing is achieved by rounding the review date to the nearest week. The refined algorithm and the least squares estimate of the review date effect $\hat{b}_r$ is calculated as shown below.

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

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

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388
Movie+User+Individual Genre+Release year+Review date Effects	0.86368

Figure 18 shows the distribution of the estimates of the review date b_r in the train set, suggesting that the review date adds only a minor variability to the average rating. Incorporating this parameter into the algorithm resulted in an RMSE of 0.86368, and an improvement of 18.51% versus the mean algorithm.

Review time effects

The final parameter to be included in the algorithm is the effect of the review time, which is shown in the previous section to have a negligible effect on the average rating. The algorithm is refined by adjusting for the effects of the time of review b_t and the least squares estimate of the review time effects, $\hat{b}_t$ is calculated using the below formula.

$$Y_{u,i} = \mu + b_i + b_u + b_g + b_y + b_r + +b_t + \epsilon_{u,i}$$

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

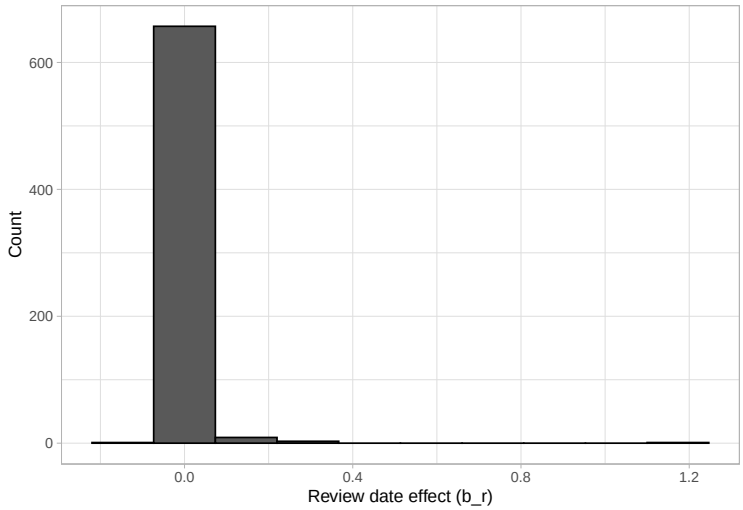

Figure 18: Movie review date effects distribution

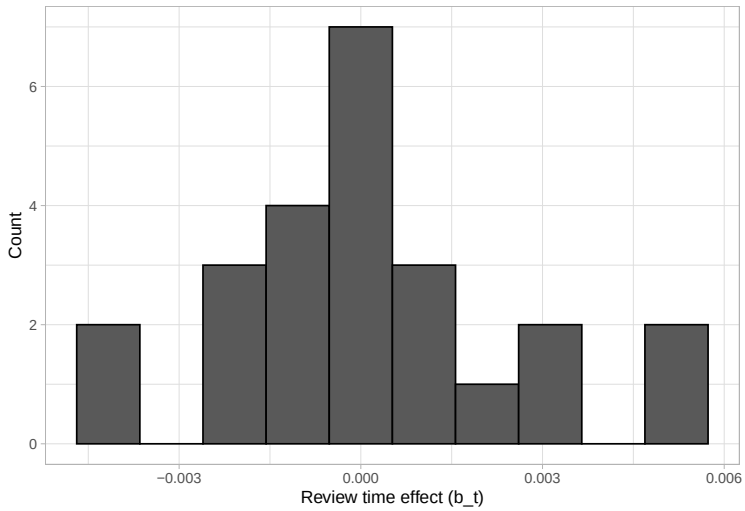

Figure 19: Movie Review time effects distribution

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388
Movie+User+Individual Genre+Release year+Review date Effects	0.86368
Movie+User+Individual Genre+Release year+Review date+Review time Effects	0.86368

The distribution of the estimates of the review time b_t in the train set, shown in Figure 19, reveals negligible additional variability. There is no improvement in the algorithm accuracy after including this parameter compared to the previous algorithm, thus suggesting that this parameter is not a worthwhile addition and therefore is neglected for further analysis.

Regularization

It is clear from the preliminary analysis that not only is the average ratings affected by movie, user, genre, year of release and review date, but that the number of ratings also varies across each parameter. Some users are more actively involved in movie reviewing, while there are also users who have rated very few movies (less than 20 movies). Some movies and movie genres, both combined and individual, received fewer ratings than others. Similarly, the number of ratings varied by year of release and review date. In each of these cases, the consequence of this variation is that the estimates of the effects (b) are subject to greater uncertainty when based on a smaller number of ratings.

Regularization is a technique used to penalize large estimates formed using small sample sizes, i.e. to constrain the total variability of the sample sizes effects. This is achieved by adding an additional penalty term in the error function and avoid over fitting. The penalty term, λ, controls the excessively fluctuating function such that the algorithm coefficients do not take extreme values. As an example, the movie effect estimates, b_i can be regularized to penalize the large effects as follows.

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

In the above equation, the first term is the least squares estimate and the second term is the penalty term that gets bigger when many b_i are large. The least squares estimate for the regularized effect of movies can be calculated as below, where n_i is the number of ratings for movie i.

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

The effect of $\frac{1}{\lambda+n_i}$ is such that when the sample size n_i is large, $\lambda+n_i\approx n_i$, and the term λ is effectively ignored resulting in a stable estimate. On the other hand, when the sample size n_i is small, the effect of λ increases and the estimate b_i shrinks towards zero. As movies and users are the dominant parameters affecting the ratings, the algorithm incorporating the movie and user effects is refined by implementing regularization as shown below.

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

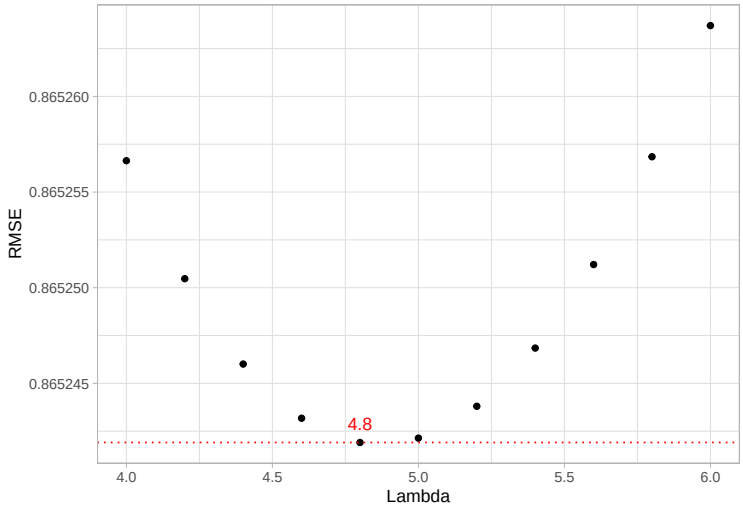

Figure 20: Select optimal penalty term

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388
Movie+User+Individual Genre+Release year+Review date Effects	0.86368
Movie+User+Individual Genre+Release year+Review date+Review time Effects	0.86368
Regularized Movie+User Effects	0.86524

Figure 20 shows the values of the RMSE for each of the tested λ values (range: 4-6, with increments of 0.2). It is clear that the optimal value of λ is 4.8, with the RMSE minimized to 0.86524, and achieving an improvement of 18.37% with respect to the mean algorithm.

The algorithm adjusted for all of the effects (movie, user, genre, year of release and review date) described previously is further refined by applying regularization and has the following form. It is to be noted that regularization is applied on both the algorithms with combined and individual genres.

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

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

For the algorithm including combined genre effects, the range of λ values applied to refine the algorithm is 4-6, with increments of 0.2. This algorithm resulted in an optimal value of 5 for λ and minimized RMSE value of 0.86454, achieving an improvement of 18.43% with respect to the mean algorithm. The range of applied λ values for the algorithm with individual genres is 13-15, with increments of 0.2. The optimal value of λ and the minimized RMSE are respectively 14 and 0.86312. This RMSE represents the total highest improvement of 18.57% with respect to the mean algorithm.

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388
Movie+User+Individual Genre+Release year+Review date Effects	0.86368
Movie+User+Individual Genre+Release year+Review date+Review time Effects	0.86368
Regularized Movie+User Effects	0.86524
Regularized Movie+User+Combined Genres+Release year+Review date Effects	0.86454
Regularized Movie+User+Individual Genre+Release year+Review date Effects	0.86312

Matrix Factorization

Recommender systems can take different approaches, such as the commonly used content based and collaborative filtering, to achieve comparable results. Content-based filtering compare item characteristics and make recommendations by finding items that are similar to what a user previously liked, whereas collaborative filtering tries to find users that have similar taste and predict based on how these users interacted with different items. Both these approaches have their strengths and weaknesses.

Matrix factorization (MF) is a type of collaborative filtering algorithm that utilises the user-item relations to make predictions for the user, based on the assumption that users who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. These algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices and have been proven to be effective for implementing recommender systems, most notably in the Netflix prize competition.

The data is converted into a matrix such that each user is in a row, each movie is in a column and the rating is in the cell. The MF approach reduces the dimensions of the rating matrix R (of size $|U|$x$|I|$), by factorizing it into a product of two latent factor matrices, P (of size $|U|$x$|K|$) for the users and Q (of size $|I|$x$|K|$) for the movies, where K is the number of latent features. The matrix product approximates $|R|$ and is given by the below equation, where each row of P and Q respectively represent the strength of the associations between a user and the features and between a movie and the features.

$$R \approx PQ^T = \hat{R}$$

To get the prediction of the rating for a movie m_i by a user u_u, the dot product of their vectors is calculated as below:

$$\hat{r_{u,i}} = p_u q_i^T = \sum_{k=1}^{K} p_{u,k} q_{k,i}$$

There are multiple ways of factorizing a matrix into multiple components, but most methods do not work with missing values in the matrix. One approach is to impute the missing values, but doing so could distort the data. Another approach is to factorize by only using the values for the observed ratings and try to minimize the squared error, but this could result in overfitting the training data. A regularization term is added to the squared error to prevent overfitting and the impact of regularization is controlled by constant β as shown below:

$$min \sum_{u,i} (r_{u,i} - p_u q_i^T)^2 + \beta((||p_u||)^2 + (||q_i||)^2)$$

The stochastic gradient descent (SGD) algorithm solves the optimization equation shown above by looping through each rating in the training data, trying to predict the rating and calculating the prediction error. It then updates the vectors q_i and p_u by a factor proportional to a constant α, known as the learning rate. Algorithm iteration over the training data continues, calculating the error and updating q_i and p_u, until convergence is found or a satisfying approximation of the original matrix is achieved. The 'recosystem' package is utilized to perform matrix factorization on the regularized movie+user algorithm. The following are the parameter values with the minimum cross validated loss: the number of latent factors (dim) is 30, the L1 regularization for $|P|$ and $|Q|$ factors (costp_l1, costq_l1) is 0, the L2 regularization for $|P|$ and $|Q|$ factors (costp_l2, costq_l2) is 0.1 and 0.01 respectively, convergence is controlled by a number of iterations, niter = 10 and learning rate, the step size in gradient descent, lrate is 0.1. The number of threads for parallel computing, nthread is 1.

Method	RMSE
Mean algorithm	1.0599
Movie Effects	0.94374
Movie+User Effects	0.86593
Movie+User+Combined Genre Effects	0.86559
Movie+User+Individual Genre Effects	0.86419
Movie+User+Individual Genre+Release year Effects	0.86388
Movie+User+Individual Genre+Release year+Review date Effects	0.86368
Movie+User+Individual Genre+Release year+Review date+Review time Effects	0.86368
Regularized Movie+User Effects	0.86524
Regularized Movie+User+Combined Genres+Release year+Review date Effects	0.86454
Regularized Movie+User+Individual Genre+Release year+Review date Effects	0.86312
Matrix Factorization	0.79646

The RMSE after applying MF with SGD on the test data is 0.79646, with an improvement of 24.86% versus the mean algorithm. On comparing the values of the RMSE from the different algorithm iterations, it is clear that MF with SGD largely improves the accuracy of the prediction.

Results

Final algorithm

The mean algorithm is refined using the train and test sets, created by partitioning the edx data set, to develop the algorithm to yield the lowest RMSE. It is clear from the analysis presented in the previous section that using the MF with SGD method on regularized movie+user algorithm achieves the best accuracy with the lowest RMSE. Therefore this algorithm is trained using the full edx data set to predict ratings within the validation data set.

Final test using the validation dataset

Method	RMSE
Mean algorithm	1.0612

The test using the MF with SGD method on the validation data set achieved an RMSE of 0.78697, with an improvement of 25.84% versus the algorithm based on the overall mean rating.

Method	RMSE
Mean algorithm	1.0612
Matrix Factorization	0.78697

Conclusions

A machine learning algorithm to predict the ratings from the MovieLens data set is developed. The Movielens data set is initially split into a training set (edx) and a test set (validation). The edx data set is further used to both train and test the algorithm under development before deciding on the final algorithm. The RMSE is used to evaluate the accuracy of the developed algorithm. The mean algorithm gives an RMSE of 1.0599 on the test data set. The best linear model (incorporating the effects of movie, user, individual genre, release year and review date, and then regularizing these parameters in order to constrain the variability of sample sizes effect) largely improved it to 0.86312. Furthermore, MF with SGD method on regularized movie+user algorithm greatly brought the RMSE down to 0.79646, thus achieving the best accuracy with the lowest RMSE. The testing of the final algorithm incorporating the MF with SGD method on the validation data set achieved an RMSE of 0.78697. This suggests that matrix factorization is a very powerful technique for recommendation system. The effects of individual genre and release year could be further explored to improve the performance of the algorithm. The Ensemble, Random Forest and the Slope One methods could also be considered in the future to enhance the overall performance of prediction. The techniques used for the algorithm development in this report are limited due to the limitations of using other powerful tools to train such a large data set on a personal computer.