A Recommendation System for Movies

Joe King

30 June, 2020

Contents

1 Introduction

The purpose of this project is to provide a machine learning model for a movie recommendation system. The data on which this model is produced is the Movielens 10M dataset available from GroupLens research lab

The Movielens 10M dataset comprises around 10 million ratings categorised by user, movie and one or more genres. Approximately 10% of the data is reserved as a validation set, with the remainder being used to train and test the modelling of the recommendation system. Despite being a partitition of the full datset, the validation set will be treated as "unseen" as far as developing the recommendation model is concerned. It will only be used at the end of the process to validate the accuracy of the model. The allocation of the Movielens 10M dataset to training/testing and to validation is shown in Table 1.

Table 1: Summary of Movielens 10M Data

Dataset	No of Rows	No of Movies	No of Users
Full Movielens 10M	10000054	69878	10677
Training/Testing (90% of Movielens 10M)	9000055	69878	10677
Validation (10% of Movielens 10M)	999999	68534	9809

The metric for measuring the accuracy of the model produced is the root mean square error (**RMSE**) represented by the equation

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

where $y_{u,i}$ is the rating for movie i by user u, $\hat{y}_{u,i}$ is the corresponding prediction produced by the model, and N is is the number of user/movie combinations in the dataset.

The report begins with some analysis of the dataset in order to look for patterns and correlations within the data. The analysis (on the whole of the Training/Testing dataset (**edx**) informs the choices of methods used for modelling the predictions. The investigation

looks at movie and user biases (or effects). These are split into two types, namely per-movie and per-user average biases, and then time-dependent movie and user biases.

Next, the details of how each bias is modelled is set out and the recommendation system model is built. The **edx** dataset is first split 90/10 into training and testing datasets **edx_train** and **edx_test** as set out in Table 2.

Table 2: Summary of **edx** Training and Test Datasets

Dataset	No of Rows	No of Movies	No of Users
edx: Training Set	8100067	69878	10677
edx: Test Set	899988	68081	9719

A linear model is used for developing the recommendation system. Each bias modelled is added to the average rating over all movies and all users. Our estimate for the average rating (denoted $\widehat{\mu}$) is the average over all the ratings in the **edx** dataset. The predicted rating for a given row in a dataset will then be $\widehat{\mu}$ plus the sum of the biases for the parameters in that row of the dataset.

These predictions are derived from the data in the **edx_train** dataset and tested on the **edx_test** dataset.

Finally the recommendations system model is used to derive predicted ratings on the validation dataset and report the results of the **RMSE** calculation.

2 Data Analysis

2.1 Average Per-Movie Ratings

Intuitively, some movies are better than others, so it is to be expected that there is a spread of ratings between movies. By the same token, "blockbuster" movies are likely to be rated more often than say "art house" movies.

Tables 3 and 4 below show the best and worst ten movies by average rating. What is important to notice is these are generally obscure movies with just a handful of ratings, so it may be necessary to prevent these seldom-rated movies from skewing the recommendations model, suggesting regularisation of the data will be necessary.

Table 3: Best 10 Movies By Average Rating

Title	Average Rating	No of Ratings
Hellhounds on My Trail	5.00	1
Satan's Tango (Sátántangó)	5.00	2
Shadows of Forgotten Ancestors	5.00	1
Fighting Elegy (Kenka erejii)	5.00	1
Sun Alley (Sonnenallee)	5.00	1
Blue Light, The (Das Blaue Licht)	5.00	1
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva)	4.75	4
Human Condition II, The (Ningen no joken II)	4.75	4
Human Condition III, The (Ningen no joken III)	4.75	4
Constantine's Sword	4.75	2

Table 4: Worst 10 Movies By Average Rating

Title	Average Rating	No of Ratings
Relative Strangers	1.0000000	1
From Justin to Kelly	0.9020101	199
Disaster Movie	0.8593750	32
Hip Hop Witch, Da	0.8214286	14
SuperBabies: Baby Geniuses 2	0.7946429	56
Besotted	0.5000000	2
Hi-Line, The	0.5000000	1
Accused (Anklaget)	0.5000000	1
Confessions of a Superhero	0.5000000	1
War of the Worlds 2: The Next Wave	0.5000000	2

If we now look at the top five movies by their number of ratings (good, bad or indifferent), then we see very much the "usual suspects", as Table 5 shows.

Table 5: Best 5 Movies By No of Ratings

Title	Average Rating	No of Ratings
Pulp Fiction	4.154789	31362
Forrest Gump	4.012822	31079
Silence of the Lambs, The	4.204101	30382
Jurassic Park	3.663522	29360
Shawshank Redemption, The	4.455131	28015

It is interesting to see a scatter plot of the average movie ratings by the number of times they are rated. Figure 1 shows this. The best and worst ten movies by rating are highlighted in red and our the top five movies by number of ratings are highlighted in blue.

The pattern shows how as the number of ratings increase, the range of average ratings narrows. Our best and worst movies by

rating are outliers in this plot, the top 5 movies by ratings count fall in the same range as other movies. Another observation is that there is a weak positive correlation between the average rating and the number of ratings (ρ = 0.2114161).

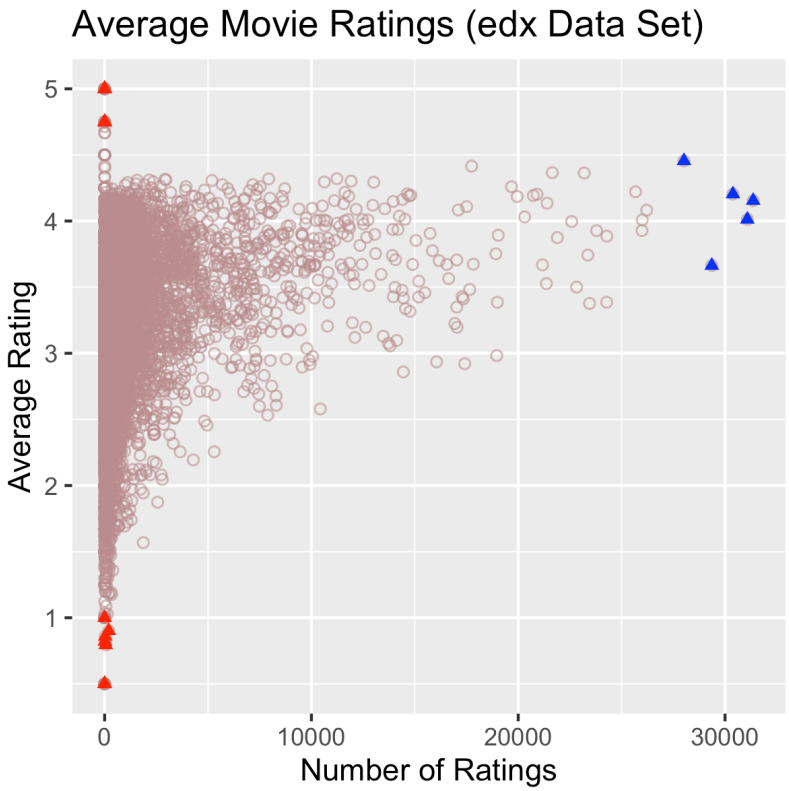

Figure 1: Average Movie Ratings

2.2 Average Per-User Ratings

As is the case for movies, it is to be expected that there is a spread of average ratings between users. Similarly, users who are avid movie watchers are likely to rate movies more often than other users. If user tastes are thrown into the mix, then intuitively the variability between users is going to be greater than for movies.

Tables 6 and 7 below show the ten users who gave the highest and lowest movie ratings respectively. Table 8 shows the five most prolific raters of movies. As for movies, it's important to notice that generally those high and low ratings are from infrequent raters.

Again, it's important to note that we may need to regularise the data to ensure that the infrequent raters do not skew the model.

Table 6: Users with the 10 Highest Average Ratings

User ID	Average Rating	No of Ratings
1	5	19
7984	5	17
11884	5	18
13027	5	29
13513	5	17
13524	5	20
15575	5	29
18965	5	49
22045	5	18
26308	5	17

Table 7: Users with the 10 Lowest Average Ratings

User ID	Average Rating	No of Ratings
28416	1.0384615	26
3457	1.0000000	19
24176	1.0000000	131
24490	1.0000000	17
6322	0.7058824	17
13496	0.5000000	17
48146	0.5000000	25
49862	0.5000000	17
62815	0.5000000	20
63381	0.5000000	18

Table 8: 5 Most Frequent Rating Users

User ID	Average Rating	No of Ratings
59269	3.264586	6616
67385	3.197720	6360
14463	2.403615	4648
68259	3.576933	4036
27468	3.826871	4023

A scatter plot of the average user ratings by the number of times they rate is shown in Figure 2. The highest and lowest users by their average rating are highlighted in red and the top five most prolific users are highlighted in blue.

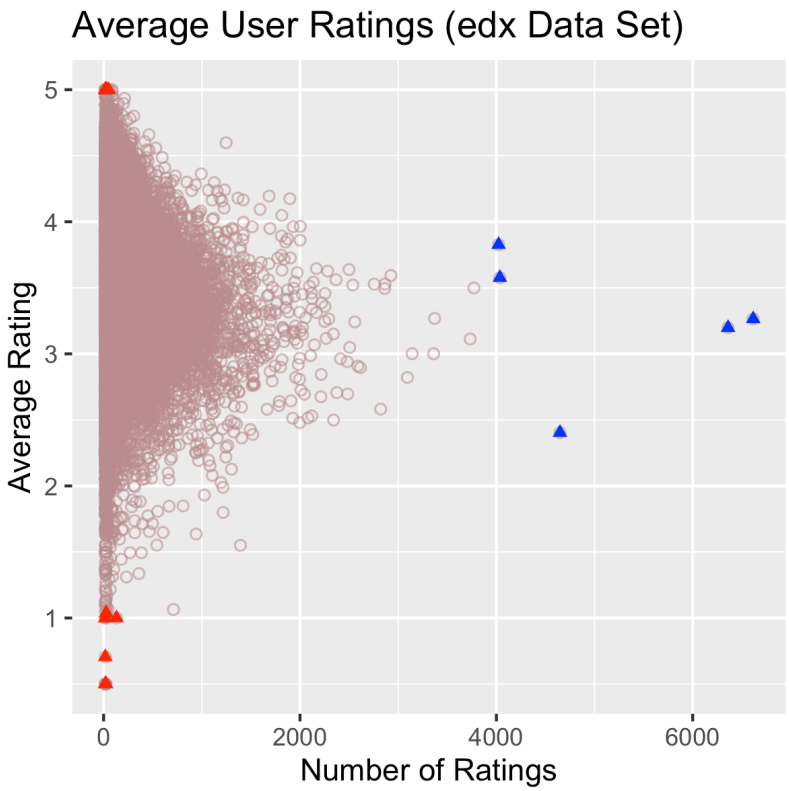

Figure 2: Average User Ratings

As for movies, the plot show the highest and lowest raters by average rating are outliers. The scatter pattern also shows how as

the number of ratings increase, the range of ratings narrows. It's interesting that our most prolific users are quite different in their average rating, suggesting that they have a bias in one direction or the other from the mean user rating.

It is also evident from the plot that the correlation between number of ratings and average rating is very weak. In fact there is a weak negative correlation (ρ = -0.1550551), suggesting users become slightly more critical as they rate more movies.

2.3 Time-dependent Variations

It is logical to think that the popularity of a movie changes over time. It might be a "hit" that starts off with good ratings when it is released, then as time goes on, opinions change as to how good it was. Or it might be a "slow burner" that improves with age. It's likely different movies go in and out of fashion over time.

The first plot in figure 3 shows how the average rating of our top 5 movies by the number of ratings has varied. It indicates a relatively long term variation in a movie's average rating over time.

When it comes to users who rate movies, then human nature comes into the mix. Rating then depends on all sorts of factors dependent on, by example, mood, fashion, or who they a watch with. The second plot in figure 3 shows how the ratings of the top 5 most prolific users vary over time. In indicates a lot of variability between users and a shorter term volatility in their ratings.

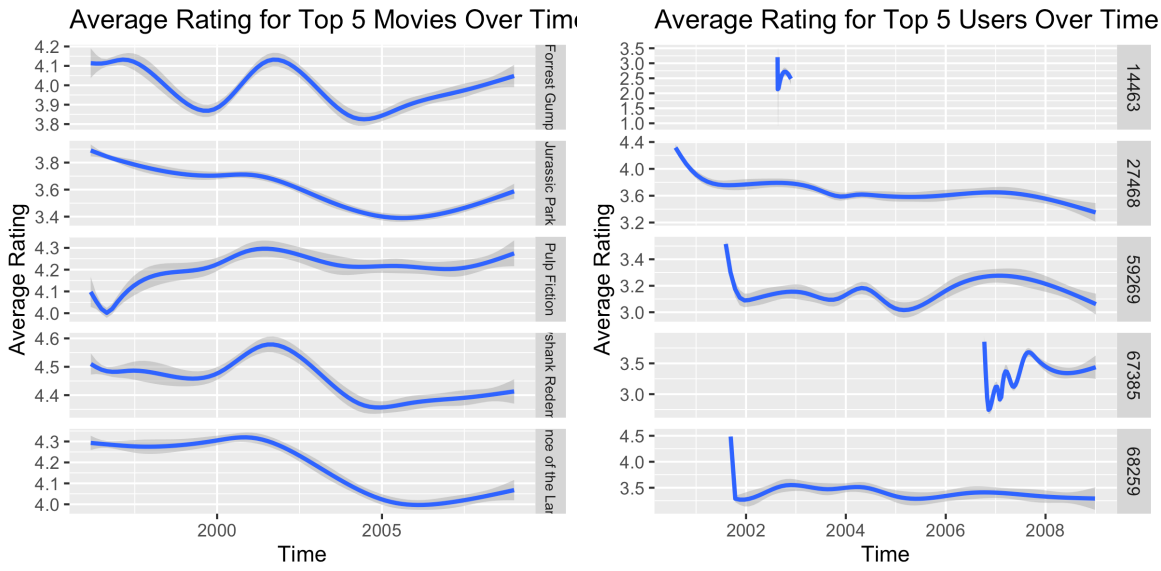

Figure 3: Time-Dependent Rating Variation

In modelling these two time-dependent biases, it makes sense to treat them separately with different time scales over which to measure the variation.

3 Building the Recommendation System

This section sets out how the recommendation system was developed. As noted in the Introduction, the Movielens 10M dataset was divided into the **edx** dataset for training and testing the model, and a validation dataset reserved for a final check on the accuracy of the model using the **RMSE** metric.

As also previously stated, the **edx** dataset is divided into **edx_train** and **edx_test** datasets.

3.1 Naive Average Model

The simplest model we can consider assumes the rating is the same for all users and movies, and that the variation from this is random. We can express this as:

A model that assumes the same rating for all movies and users with all the differences explained by random variation would look like this:

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

where μ is the average rating and $\epsilon_{u,i}$ denotes the random variation from μ for movie i and user u.

The best estimate available for μ is the average over all the ratings in the training dataset (**edx_train**). Our estimate is

$\hat{\mu}$ = 3.5125096

When we calculate **RMSE** on the edx_test dataset, we find

RMSE $= 1.060969$

Not unexpectedly, this naive average model really does not give us an accurate estimate of $Y_{u,i}$.

3.2 Movie Bias Model

The earlier analysis of the data in **edx** suggests that we should allow for a per-movie bias in the rating. Intuitively, this makes sense: some movies are better than others!

The movie bias model assumes that each movie varies from the average by a fixed bias amount and looks like this:

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

where the new item b_i represents the bias from the average of movie i.

The estimate $\hat{b}_i$ of b_i is given by

$$\hat{b}_i = \frac{\sum_{i \in A}(Y_{u,i} - \hat{\mu})}{N_{i \in A}}$$

where $i \in A$ represents all movies i in the training set A (**edx_train**) and $N_{i \in A}$ is the number of ratings in A for movie i.

Figure 4 shows the distribution of $\hat{b}_i$.

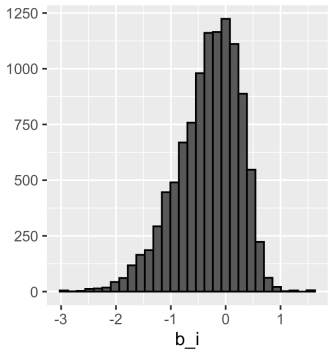

Figure 4: Movie Bias Distribution

Using this model to make predictions $Y_{u,i}$ on **edx_test** yields

RMSE $= 0.9438836$

This certainly improves our estimates over the naive average model.

3.3 Movie and User Bias Model

The visualisation of the data in **edx** in the Data Analysis section above shows that there is likely to be a per-user bias in ratings.

It makes sense therefore to add in a user bias to the model. The model now looks like this:

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

where the new item b_u represents the bias from the average of movie i over and above the movie bias b_i.

The estimate $\hat{b}$ of b_u is given by

$$\hat{b}_u = \frac{\sum_{u \in A}(Y_{u,i} - \hat{\mu} - \hat{b}_i)}{N_{u \in A}}$$

where $N_{u \in A}$ is the number of ratings in the training data set A (**edx_train**) for user u.

The distribution of $\hat{b}_u$ is shown in figure 5.

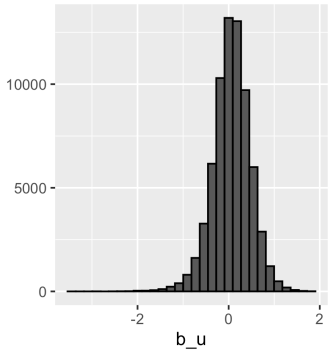

Figure 5: User Bias Distribution

Using this model to make predictions $Y_{u,i}$ on **edx_test** yields

RMSE $= 0.8660993$

Another improvement in the **RMSE** showing this is moving in the right direction.

3.4 Regularised Movie and User Bias Model

The inspection of the **edx** dataset revealed that movies with few ratings and users who have made few ratings could potentially skew the estimates of the movie and user biases. In order to deal with this problem, an approach designed to take less account of the ratings for the least rated movies and the least active users is used.

Specifically, a penalty is added to the least squares calculation:

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

The aim is to estimate λ to minimise this expression. This is done by iteration to find $\hat{\lambda}$, the estimate of λ that minimises

$$(\frac{1}{N_B}) \sum_{u,i \in B} (Y_{u,i} - \frac{\hat{b}_i \times N_{i \in A}}{(\lambda + N_{i \in A})} - \frac{\hat{b}_u \times N_{u \in A}}{(\lambda + N_{u \in A})})^2$$

where N_B is the number of ratings in the test dataset B (**edx_test**).

(Note that $\hat{b}_i \times N_{i \in A}$ and $\hat{b}_u \times N_{u \in A}$ are equivalent to $\sum_{i \in A} b_i$ and $\sum_{u \in A} b_u$ respectively).

A range of possible values from 0 to 10 in 0.25 increments was used to estimate λ. Figure 6 shows a plot of **RMSE** by λ.

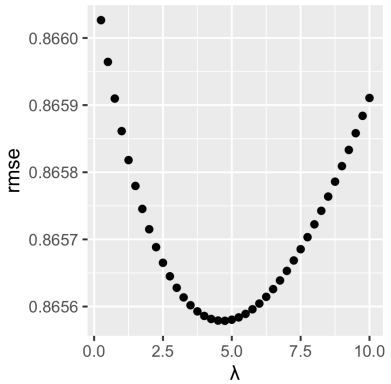

Figure 6: Regularising Movie and User Biases λ v **RMSE**

The estimate of λ for this model is $\hat{\lambda} = 4.75$ and this yields

RMSE $= 0.8655788$

3.5 Regularised Movie and User Biases with Added Time Dependent Movie Bias Model

The next bias to consider is the time-dependent movie bias. In other words, the bias relative to changes in average movie ratings over time.

The model now looks like this:

$$Y_{u,i} = \mu + b_i + b_u + f(d_i) + \epsilon_{u,i}$$

where d_i is the bias for movie i on day d and f is a smooth function of d_i.

In reality, it is not possible to accurately define function $f(d_i)$. However, we can approximate it by allocating ratings to time "bins", and work out the average bias within each of these bins. It makes sense to also continue to use a "penalised least squares approach" and at this stage the simplest choice is to use the value of $\hat{\lambda}$ already calculated. When all biases have been modelled, we will return to see if there is a better estimate to minimise **RMSE**.

So the residual bias being modelled is

$$\hat{d}_i(t_j) = \frac{1}{N_{i \in t_j}} \sum_{i \in t_j} (Y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u)$$

where t_j is time bin j and $N_{i \in t_j}$ is the number of ratings for movie i in time bin j.

Intuitively, since movies are released on a weekly basis, it makes sense to use bins based on weeks. Of course, it is necessary to pick an appropriate time bin size. From the data visualisation in the analysis section of the report, it looks like there is a relatively long term variation in the time dependent movie bias.

Continuing with the penalized least squares approach with the previously calculated $\hat{\lambda}$, the aim here is to pick a bin size to minimise this expression

$$(\frac{1}{N_B}) \sum_{u,i \in B} (Y_{u,i} - \frac{\hat{b}_i \times N_{i \in A}}{(\lambda + N_{i \in A})} - \frac{\hat{b}_u \times N_{u \in A}}{(\lambda + N_{u \in A})} - \frac{\hat{d}_i(t_j) \times N_{i \in t_j, A}}{(\lambda + N_{i \in t_j, A})})^2$$

where $N_{i \in t_j, A}$ is the number of ratings for movie i in dataset A (**edx_train**) that fall in time bin t_j.

For any particular movie, the time bins, (always counted from a Monday) were calculated from the first time it was given a rating with the training dataset (**edx_train**). Where a movie was released within the period covered by the data, this means that bins were counted from the initial release date of the movie. A range of bin sizes were considered from 1 week to 52 weeks. The optimisation showed that the optimal bin size was 52.

It was found that the RMSE started to flatten out after about 20 weeks but was still decreasing all the way through to 52 weeks (Figure 7).

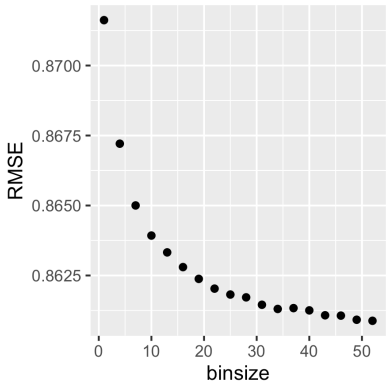

Figure 7: Time-Dependent Movie Bias Binsize

Using a bin size of 52 weeks yields

RMSE $= 0.8608815$

3.6 Regularised Static and Time-Dependent Movie and User Biases Model

The final bias to consider is the time-dependent user bias. In other words, the bias relative to changes in a user ratings over time.

The model now looks like this:

$$Y_{u,i} = \mu + b_i + b_u + f(d_i) + f(d_u) + \epsilon_{u,i}$$

where d_u is the bias for user un day d and f is a smooth function of d_i.

The approach here is the same as for movies, but it's likely that the time-dependent user bias is rather more complicated. However, for simplicity, it's a good starting point. More complicated approaches are beyond the scope of this report.

The residual bias being modelled is

$$\hat{d}_u(t_v) = \frac{1}{N_{u \in t_v}} \sum_{i \in t_j} (Y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u - \hat{d}_i(t_j))$$

where t_v is time bin v and $N_{u \in t_v}$ is the number of ratings for user u in time bin v.

So that all users are treated in a similar way, the bins are calculated from the Monday on or before a user's first rating. From the data visualisation in the analysis section of the report, there is rather more variability between users and volatility within a user's ratings. Not surprising as human nature is involved!

Again using the penalized least squares approach with the previously calculated $\hat{\lambda}$, the aim here is to pick a bin size to minimise the following expression

$$(\frac{1}{N_B}) \sum_{u,i \in B} (Y_{u,i} - \frac{\hat{b}_i \times N_{i \in A}}{(\lambda + N_{i \in A})} - \frac{\hat{b}_u \times N_{u \in A}}{(\lambda + N_{u \in A})} - \frac{\hat{d}_i(t_j) \times N_{i \in t_j, A}}{(\lambda + N_{i \in t_j, A})} - \frac{\hat{d}_u(t_v) \times N_{u \in t_v, A}}{(\lambda + N_{u \in t_v, A})})^2$$

where $N_{u \in t_v, A}$ is the number of ratings for movie i in dataset A (**edx_train**) that fall in time bin t_v.

For this bias, a shorter bin size is likely to be more appropriate. A range of bin sizes were considered from 1 week to 4 weeks. The optimisation showed that the optimal bin size was 1.

It was found that the RMSE increased as the bin size increased (Figure 8). This indicates that this is not the end of the story when it comes to time-dependent user bias, although it does give a significant reduction in the **RMSE**.

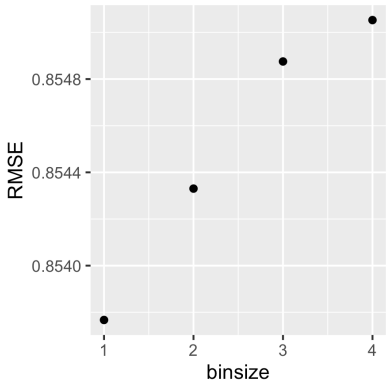

Figure 8: Time-Dependent User Bias Binsize

Using a bin size of 1 week yields

RMSE $= 0.8537669$

3.7 Final Regularised Static and Time-Dependent Movie and User Biases Model

At this stage our model uses a penalty constant $\hat{\lambda}$ = 4.75 derived before the time-dependent biases were added. A last step therefore is to see if another value of $\hat{\lambda}$ will improve the estimates still further.

The expression (as in previous section) to be optimised for λ is

$$(\frac{1}{N_B}) \sum_{u,i \in B} (Y_{u,i} - \frac{\hat{b}_i \times N_{i \in A}}{(\lambda + N_{i \in A})} - \frac{\hat{b}_u \times N_{u \in A}}{(\lambda + N_{u \in A})} - \frac{\hat{d}_i(t_j) \times N_{i \in t_j, A}}{(\lambda + N_{i \in t_j, A})} - \frac{\hat{d}_u(t_v) \times N_{u \in t_v, A}}{(\lambda + N_{u \in t_v, A})})^2$$

Again, a range of values for λ between 0 and 10 in increments of 0.25 was chosen for optimisation. Figure 9 shows the plot of λ v **RMSE**.

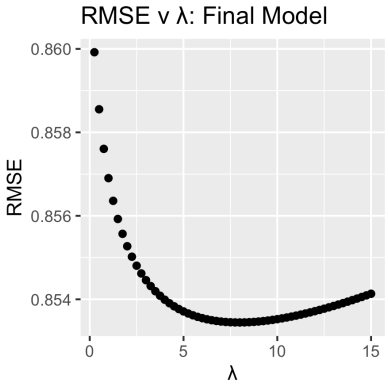

Figure 9: Optimising λ for Final Model

The final value for $\hat{\lambda}$ = 8 and this gives

RMSE $= 0.8534453$

3.8 Validating the Movie Recommendations System and Calculating the Final RMSE

The developed model for making Movie Recommendations predicts the rating a user will give a movie on a given date.

Each prediction is the sum of these five elements:

3.8.1 Estimated Average Rating

$\hat{\mu} = \frac{1}{N_A} \sum_{i,j \in A} Y_{i,j} =$ 3.5125096

3.8.2 Penalised Movie Bias

$$\hat{b}_i^* = \frac{\hat{b}_i \times N_{i \in A}}{(\hat{\lambda} + N_{i \in A})}$$

where $\hat{b}_i$ is the average over the training dataset A (**edx_train**) of all ratings for movie i, $N_{i \in A}$ is the corresponding number of ratings and $\hat{\lambda} = 8$

3.8.3 Penalised User Bias

$$\hat{b}_u^* = \frac{\hat{b}_u \times N_{u \in A}}{(\hat{\lambda} + N_{u \in A})}$$

where $\hat{b}_u$ is the average over the training dataset A (**edx_train**) of all ratings for movie i and $N_{i \in A}$ is the corresponding number of ratings

3.8.4 Penalised Time-Dependent Movie Bias

$$\hat{d}_i^*(t_j) = \frac{\hat{d}_i(t_j) \times N_{i \in t_j, A}}{(\lambda + N_{i \in t_j, A})}$$

where $\hat{d}_i(t_j)$ is the average over the training dataset A (**edx_train**) for all ratings for movie i that fall in time bin t_j, and $N_{i \in t_j, A}$ is the corresponding number of ratings.

The time bins t_j are counted from the Monday before the first rating of movie i in training dataset A in 52-week periods.

3.8.5 Penalised Time-Dependent User Bias

$$\widehat{d}_u^*(t_v) = \frac{\hat{d}_u(t_v) \times N_{u \in t_v, A}}{(\lambda + N_{u \in t_v, A})}$$

where $\hat{d}_u(t_v)$ is the average over the training dataset A (**edx_train**) for all ratings for user u that fall in time bin t_v, and $N_{u \in t_v, A}$ is the corresponding number of ratings.

The time bins t_v are counted from the Monday before the first rating by user i in training dataset A in 1-week periods.

Calculating the predicted ratings over the validation data set (**validation**) using this model yields the following final result for the Root Mean Square Error

RMSE $= 0.8529406$

4 Summary and Further Work

As can be seen from the previous section, the root mean square estimate for the validation data set is 0.8529406.

The model formula for deriving predictions covers four biases, two each for movie and user, both static and time-dependent.

The analysis suggested that both user and movie biases existed in the ratings data so it made a deal of sense to model for these. The RMSE figure for the validation set turned out lower than that for the test dataset. This shows that it the model developed presents a reasonable approach to the problem.

While it was entirely appropriate to fit a linear model to the static movie and user biases, the modelling of the time-dependent biases was perhaps a little simplistic. Using time bins allowed a linear model to be fitted within each time bin. Further work could explore a more sophisticated method for fitting a non-linear, continuous model to these time-dependent biases, especially for the user bias where the volatility suggests there are other effects to investigate.

The use of penalised least squares in deriving the biases certainly improved the accuracy of the modelling and helped to reduce the influence of less rated movies and less frequent users from the predictions.

The datasets also included information regarding the genre of movies. Biases associated with movie genres have not been investigated here, but it is likely that a Principal Component Analysis of genre versus movie and genre versus user is likely to reveal some further insights into improvements in the modelling of the recommendation system.

Other influences on ratings might include principal actor and director biases, although this is beyond the scope of the data here.