# Movie Recommendation Model

*Doug MacClure*

*5/1/2019*

## Introduction

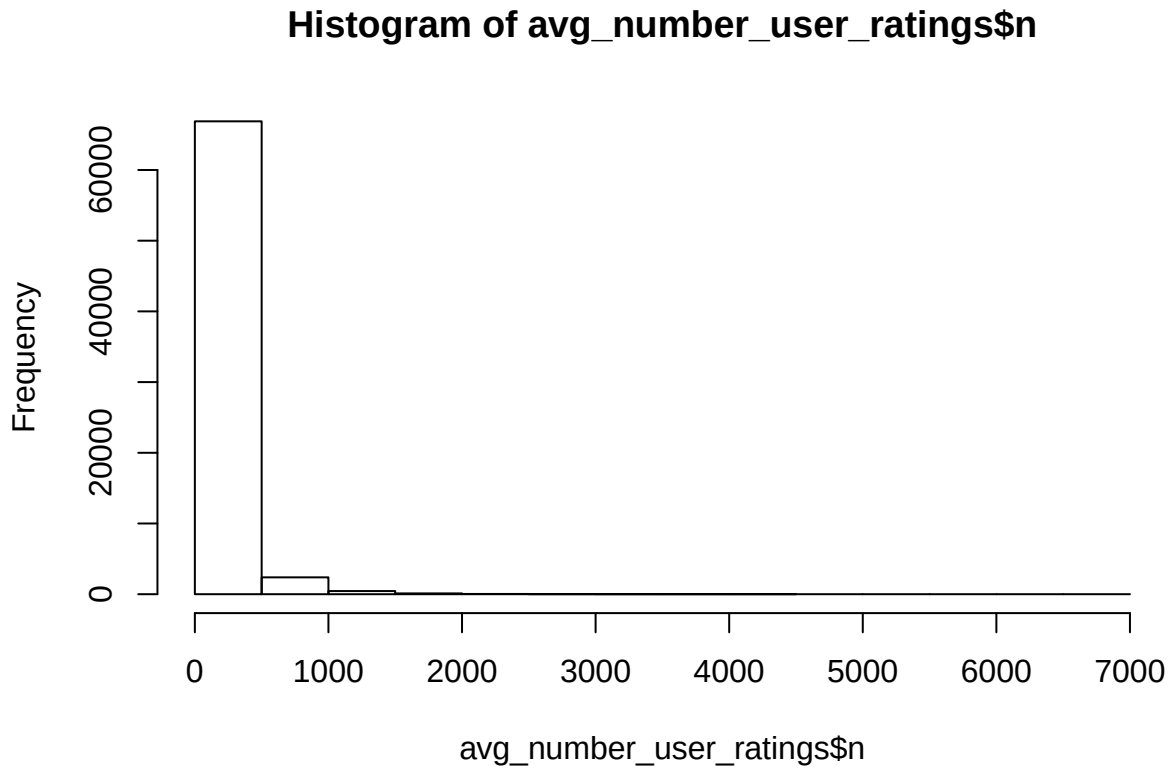
In this report, we analyze the movielens data set (found here: http://files.grouplens.org/datasets/movielens/ml-10m.zip) and construct various bias/effects-based recommendation algorithms using 90% of the movielens dataset. Other potential recommendation algorithms are discussed, and our approach using bias/effects instead is justified.

This project is done for the Data Science Capstone assignment via HarvardX. HarvardX has provided code up to creating training and validation datasets. The goal of this model is to create a recommendation model which predicts user-defined movie ratings evaluated on a validation dataset with root-mean squared error less than 0.87750.

To begin, perform the following steps: install (if necessary) and load the required packages to run the R code: caret, lubridate and tidyverse. Next, download the required data, and wrangle/coerce data for analysis. Finally, split the prepared data into training and validation datasets.

We begin our analysis of the data by noting which potential predictive variables we have to work with.

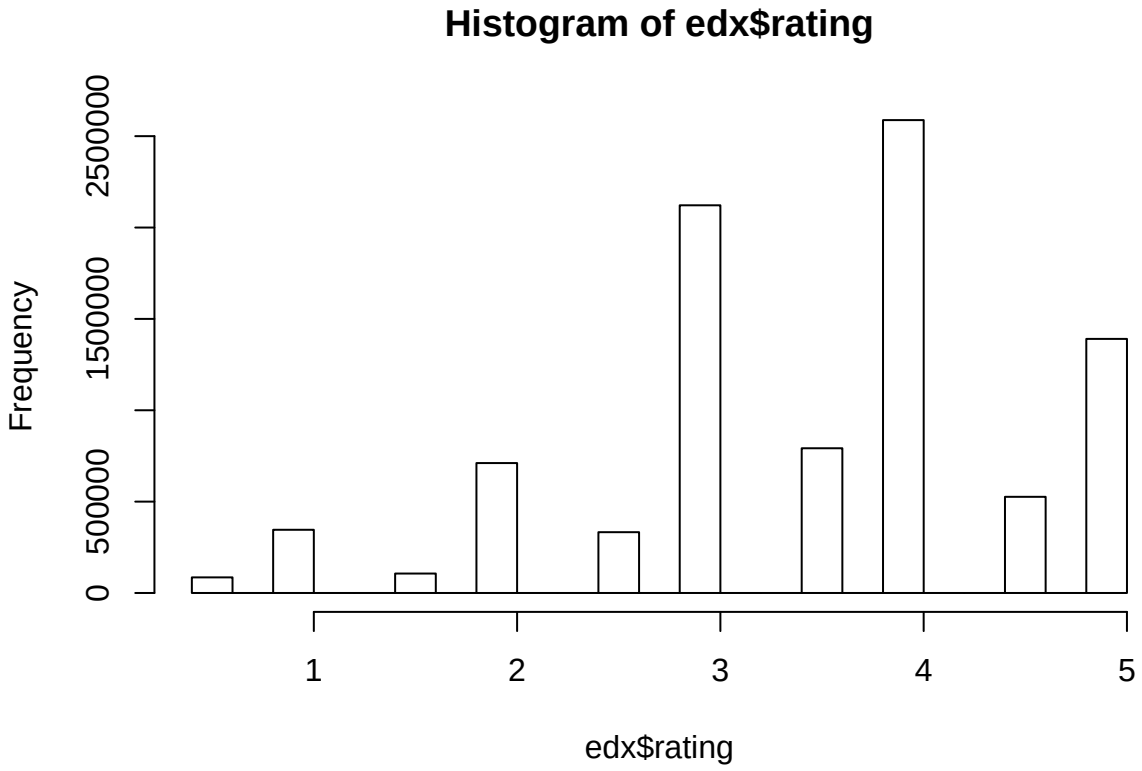
Note that the purpose of this model is to predict the rating a user would give for a given movie. Hence, we wish to build a model which predicts a movie rating given a userId and movieId. There are other predictors we can consider. Consider the following frequency histogram for number of movies rated per user.



Most users rate less than 500 movies, and some users rate many movies. However, the average number of movies rated is:

```
## [1] 128.7968
```

Hence, we have enough data and variation in the data to build a movie recommendation model, which is further verified by considering the movie-rating distribution.




## Model Construction and Analysis

Now, note that the approach discussed here will utilize a random-effects approach, where the predictors are considered random variables, which depend on userId and movieId, as opposed to fixed quantities. The author has considered regression and classification models, but the structure of the data is inappropriate for such analysis, particularly on laptops, due to memory constraints.

First, to get an idea of the baseline we are working with, consider the RMSE if we consider the constant-effects model: $Y = \mu$, where we simply assign every movie the average movie rating among all movies and all ratings.

| method | RMSE |
|---|---|
| Constant Effects Model | 1.060651 |

The way to interpret this RMSE is that the constant effects model can be expected to be within a 1-star rating of the true user rating of the movie. Can we do better?

Now, lets analyze the following potential random-effects model:

$$Y_u = \mu + b_u + b_i,$$

where $\mu$ is a constant which represents the overall mean movie rating from all users for all movies. Here, $b_i$ is the movie and $b_u$ are random variables, dependent upon a user $u$ and movie $i$. To be more specific, $b_i$ is the average difference between a movie rating for movie $i$ and the overall average movie rating for all movies (i.e., the movie effect), while $b_u$ is the average difference between movies $i$ rated by user $u$ and the sum of $\mu$ and $b_i$ (i.e., the user effect).

First, consider the incremental improvement in RMSE when using the user-effects model versus the movie and user effects model.

| method | RMSE |
|---|---|
| Constant Effects Model | 1.0606506 |
| User Effects Model | 0.9945538 |
| Movie + User Effects Model | 0.8655329 |

Using R version 3.6, we have a RMSE score of 0.8655329 in using both user and movie effects. Now, note that we can do better than this. Consider the following information.

| movie_rating_stdev |
|---|
| 0.4861842 |

| title | b_i |
|---|---|
| Besotted (2001) | -3.012464 |
| Grief (1993) | -3.012464 |
| Besotted (2001) | -3.012464 |
| Hi-Line, The (1999) | -3.012464 |
| Accused (Anklaget) (2005) | -3.012464 |
| War of the Worlds 2: The Next Wave (2008) | -2.762464 |
| War of the Worlds 2: The Next Wave (2008) | -2.762464 |
| SuperBabies: Baby Geniuses 2 (2004) | -2.698905 |
| SuperBabies: Baby Geniuses 2 (2004) | -2.698905 |
| SuperBabies: Baby Geniuses 2 (2004) | -2.698905 |

| title | b_i |
|---|---|
| Constantine's Sword (2007) | 1.487536 |
| Sun Alley (Sonnenallee) (1999) | 1.487536 |
| Fighting Elegy (Kenka erejii) (1966) | 1.487536 |
| Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994) | 1.487536 |
| Shadows of Forgotten Ancestors (1964) | 1.487536 |
| Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994) | 1.487536 |
| Hellhounds on My Trail (1999) | 1.487536 |
| Blue Light, The (Das Blaue Licht) (1932) | 1.487536 |
| Human Condition II, The (Ningen no joken II) (1959) | 1.320869 |
| Human Condition II, The (Ningen no joken II) (1959) | 1.320869 |

The variance in movie effects is quite high, despite the relatively small improvement in incorporating movie effects into our model. Also, as we can see from the above table, most of the worst and best rated movies are relatively unknown, and have very few ratings associated with them. This is one source of the relatively large variance. Since the error equation for a model is given by

$$Bias^2 + Variance + \epsilon$$

where $\epsilon$ is irreducible error that cannot be reduced by reducing bias and variance.

We will reduce the variance in this model by considering **ridge regresssion**. In ridge regression, we reduce the variance a variable includes into a model by including a penalty term in the residual sum of squares equation which gets larger when $b_i^2$ and $b_u^2$ get large:

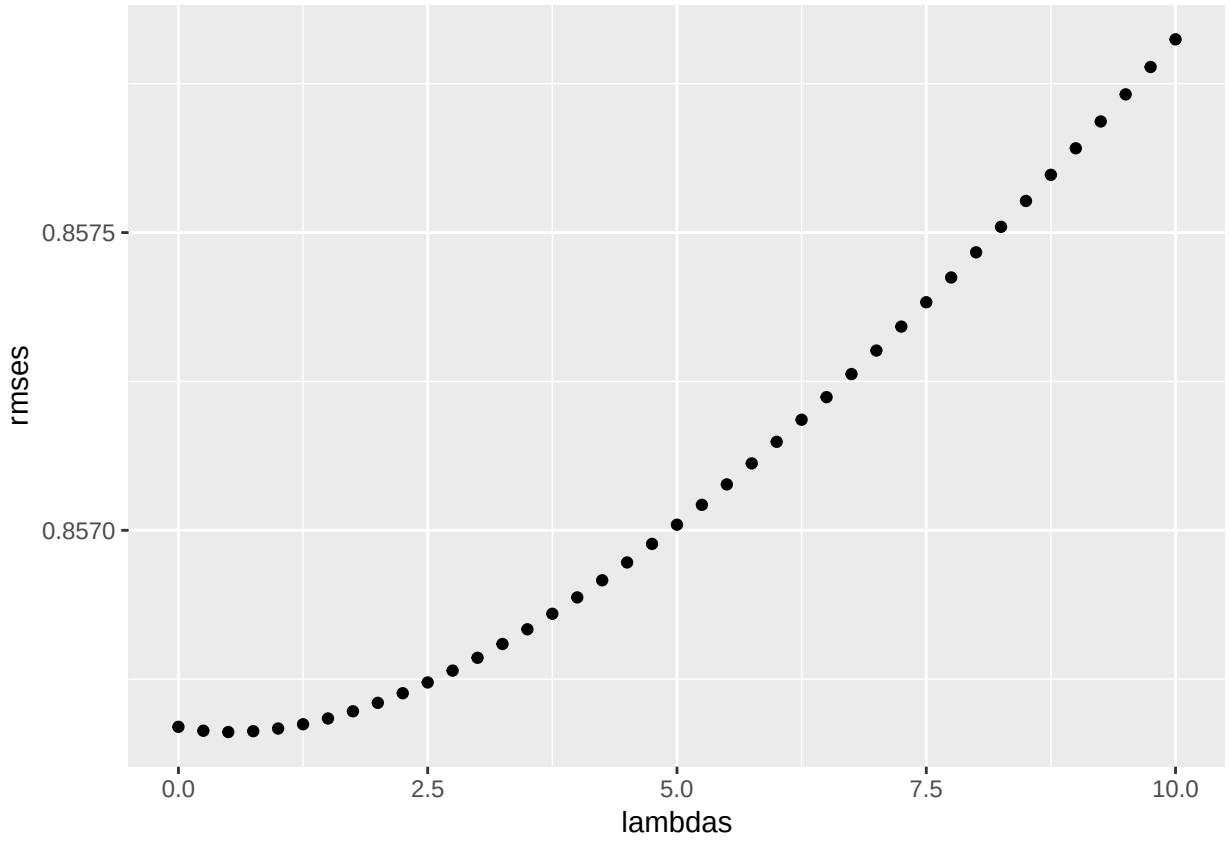
$$\frac{1}{N}\sum_{u,i}(y_{u,i} - \mu - b_i - b_u)^2 + \lambda\left(\sum_i b_i^2 + \sum_u b_u^2\right).$$

Now, we wish to find the optimal $\lambda$ such that the above is minimized. We can do this by taking partial derivatives with respect to $b_i$ and $b_u$, and set them to zero. This reduces to a standard Calculus III problem, except $b_i$ and $b_u$ both depend on the paramter $\lambda$. To be more specific, we have that

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

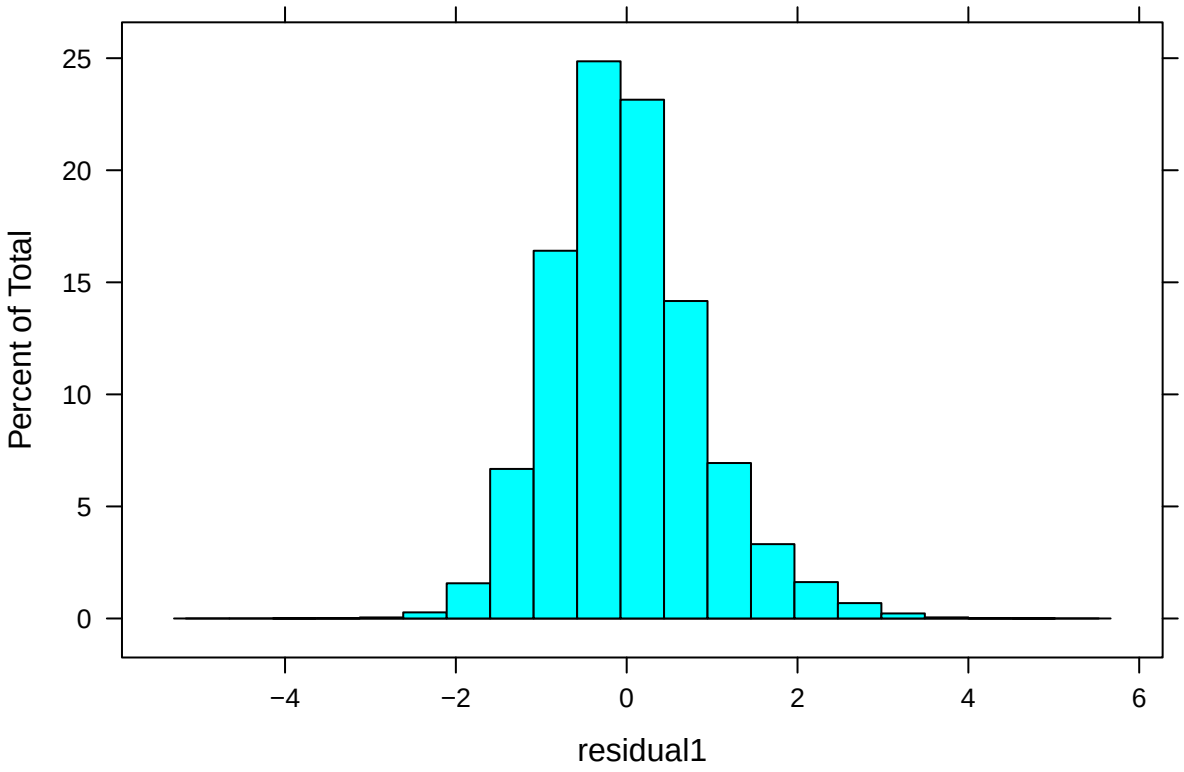
$$\hat{b}_u(\lambda) = \frac{1}{n_u + \lambda}\sum_{u=1}^{n_u}(Y_{u,i} - \mu - \hat{b_i}(\lambda))$$

To find the approximately optimal $\lambda$, it is important that we **DO NOT** determine the optimal $\lambda$ using the test set! Doing so is considered a fatal error in machine learning. The reason for this is because we are not approximating how the optimized model would run on live data. We've already tuned the model to both the training AND test sets. Hence, there is no final validation of the model. Hence, we tune $\lambda$ using the training set and determine the performance of the model on the test/validation set.




As we can see, optimally $\lambda \approx 0.5$. Now, consider the improvement when considering regularization.

The author has observed other RMSE scores by running the code with R version 3.5.2. We can also determine how the model is performing by considering the residual distribution. Note that the mean is centered at zero and the distribution is approximately normal. This is a good indication that the model is neither over-estimating nor under-estimating on average.




Next, we consider incorporating genres-effects into our random-effects model. Do we see an improvement in RMSE? Our new model is:

$$Y_{u,i} = \mu + \hat{b_i}(\lambda) + \hat{b_u}(\lambda) + g_{u,i}.$$

Observe the new residual distribution is essentially the same as before. Hence, we are starting to see diminishing returns in including more random-effects variables into our model.

Despite diminishing returns, we do see a decrease in RMSE of $\approx 0.0004$. To finish, we incorporate time effects. However, before we do so, we need to note that timestamp is a number representing the number of minute since 01-01-1970 00:00. Note that we need to reduce the granularity of such a variable, since each particular timestamp will be associated with a small number of ratings, which we observe after computing the number of distinct timestamps and the total numer of observations we are considering.

| number_of_distinct_timestamps | number_of_observations |
|---:|---:|
| 6520453 | 9000061 |

Hence, the granularity of the time effect is by week. Further, observe that the new residual histogram is again essentially the same as before.

## Results

Finally, we compare the RMSE of each random-effects model to determine the best model, and if this model is satisfactory.

| method | RMSE |
|---|---|
| Constant Effects Model | 1.0606506 |
| User Effects Model | 0.9945538 |
| Movie + User Effects Model | 0.8655329 |
| Regularized Movie + User Effects Model | 0.8653901 |
| Genre + Regulraized Movie + User Effects Model | 0.8650573 |
| Movie + User + Genre + Time Effects Model | 0.8649572 |

We see that for $d_{u,i}$ denoting the random variable associated with the average rating of movie $u$ by user $i$ during a given week, the best performing model of the three is model 3:

$$Y_{u,i} \approx \mu + \hat{b_u}(\lambda) + \hat{b_i}(\lambda) + g_{u,i} + d_{u,i}.$$

Here, the RMSE is 0.86495722., which is well below the target RMSE of 0.87750.

Now, note that the author has considered generative model techniques as well. However, using Naive Bayes thus far has been unable to yield a model with RMSE below 1.1 and accuracy greater than 35%.

## Conclusion

In conclusion, we have shown that a random-effects modeling approach is most appropriate given the data we have to work with. We have generated a model which meets the initial goal of having the predictive power where the RMSE is below 0.8775.

There is room for further improvement in this model. The author would like to eventually try ensemble models, where regression and is done on smaller and more computable subsets.