MovieLens Project Report

William Fang

10/06/2020

Contents

1 Introduction

The goal of this project is to create a movie recommendation system using the MovieLens data set. Using ideas introduced in **PH125.8x: Data Science: Machine Learning**, the aim is to develop a ML algorithm on one subset of data, and use it to predict movie ratings against a validation set.

The data set is the 10M version of the MovieLens data set, with the following structure:

```
'data.frame':   9000055 obs. of  6 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653..
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (19"..
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thrill"..
```

Each row is a single rating of a movie by a user.

The models introduced in the course are implemented and evaluated against the data to establish a baseline accuracy which will be improved upon. Then the features of the data that have not been incorporated in the course models are explored and analysed to determine if they can be used to improve the prediction model. Finally the models will be tested against a validation set (that was not used during development) to establish their accuracy, which is calculated using Root Mean Square Error.

2 Analysis

The `edx` training set is split into a `train_set` and `test_set` for the purposes of analyzing and developing models. As a starting point, the course models from **PH125.8x: Data Science: Machine Learning** are implemented and evaluated against these sets.

2.1 Course Models

2.1.1 Just The Average

The first model was simply to predict a rating value as the average of all ratings.

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

$Y_{u,i}$ is the predicted rating of movie i by user u, μ is the average of all ratings, and $\epsilon_{u,i}$ represents independent errors sampled from the same distribution centered at 0.

The RMSE value obtained from this model is 1.0600537.

2.1.2 Movie Effect

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

Extension of the first model, where a movie effect term is added. b_i is the bias for movie i, which is calculated as the average of the difference from μ across all ratings for the movie.

The RMSE value obtained from this model is 0.9429615.

2.1.3 Movie and User Effect

The movie effect model has another term added to it, that represents user bias b_u, and is calculated in a similar manner as for movie effect.

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

The RMSE value obtained from this model is 0.8646843.

2.1.4 Regularized Movie Effect Model.

The movie effect is regularized by providing a term that essentially penalizes movies with few ratings.

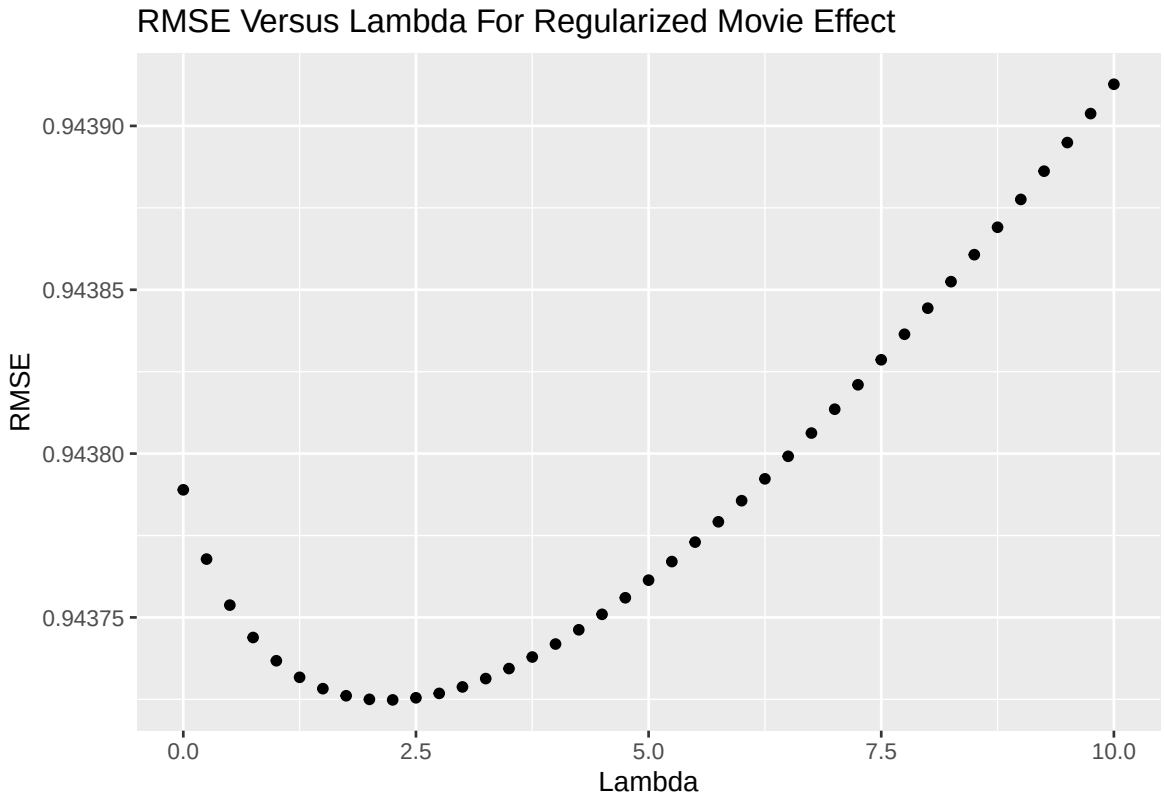
$$Y_{u,i} = \mu + b_i(\lambda) + \varepsilon_{u,i}$$

where b_i is approximated by:

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

n_i is the number of ratings of movie i. The optimal value for λ is obtained using cross validation.

The RMSE values versus lambda are shown in the plot below:

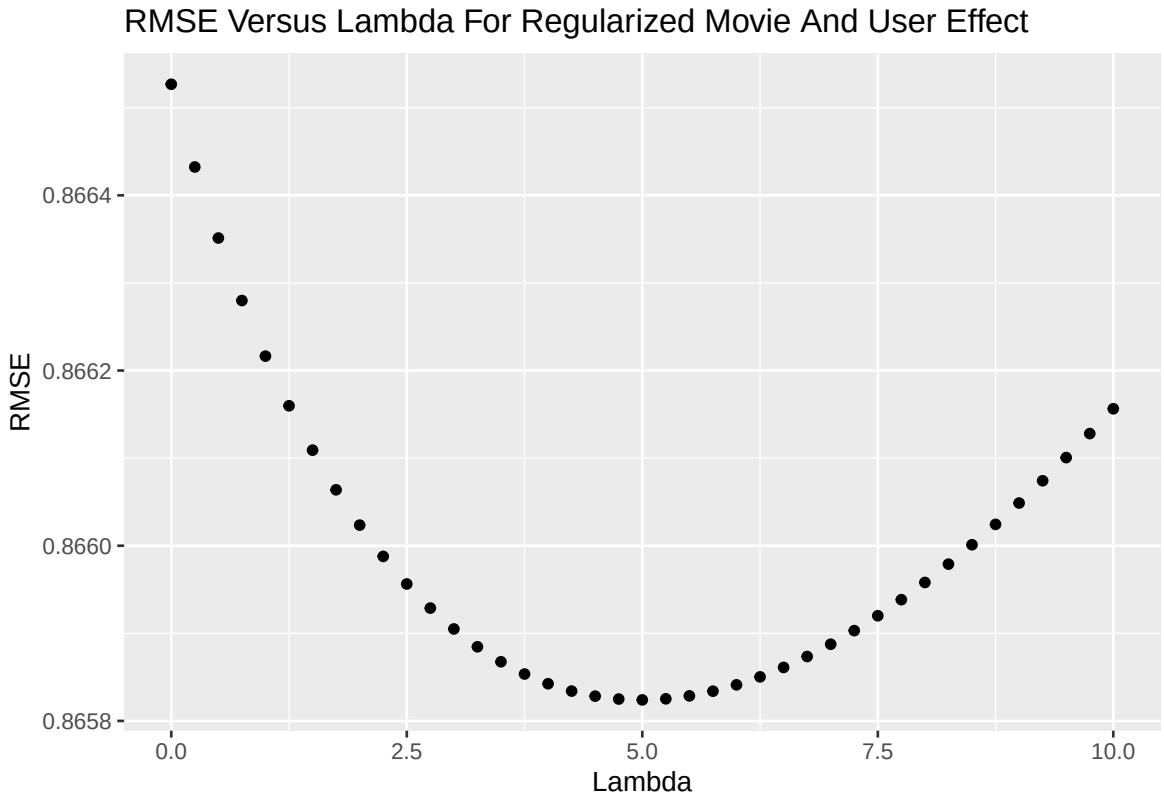


The optimal value for λ is 2.25. Using that for training on the `train_set`, and evaluating against the `test_set`, the RMSE value obtained from this model is 0.9429389.

2.1.5 Regularized Movie and User Effects Model

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

In a similar manner to the regularized movie effect model, the user bias can also be regularized.



The above plot shows RMSE values versus lambda, and the optimal value for λ is now 5. Using that for training on the `train_set`, and evaluating against the `test_set`, the RMSE value obtained from this model is 0.9429389.

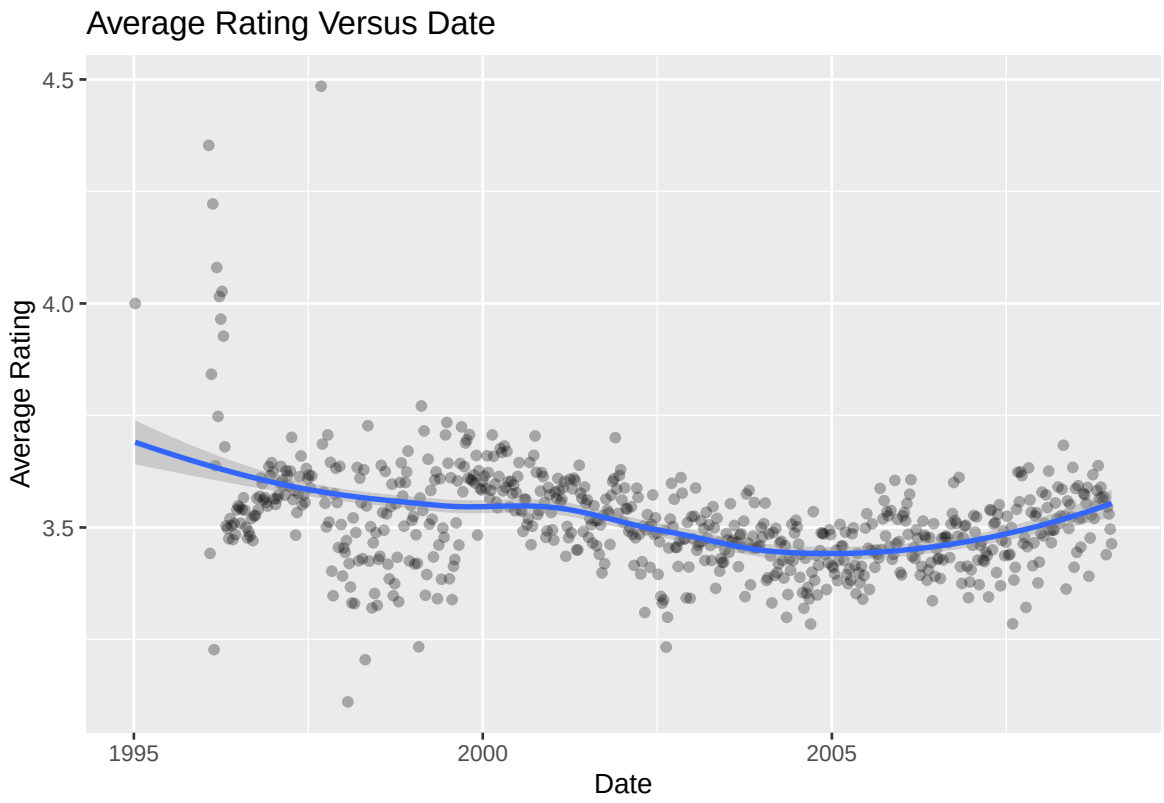
From the data, there are 3 elements that are yet to be incorporated into any of these models, movie title, timestamp of the rating, and movie genres. Using the regularized movie and user effects model as a base line model, these features are examined and new terms are developed to be added to the base.

2.2 Movie Title

It is not obvious that there would be any predictive power in the title of a movie, distinct from that of the movie effects value b_i, so no time has been spent exploring this feature of the data set.

2.3 Timestamp

In the course exercises the average rating per week was plotted:

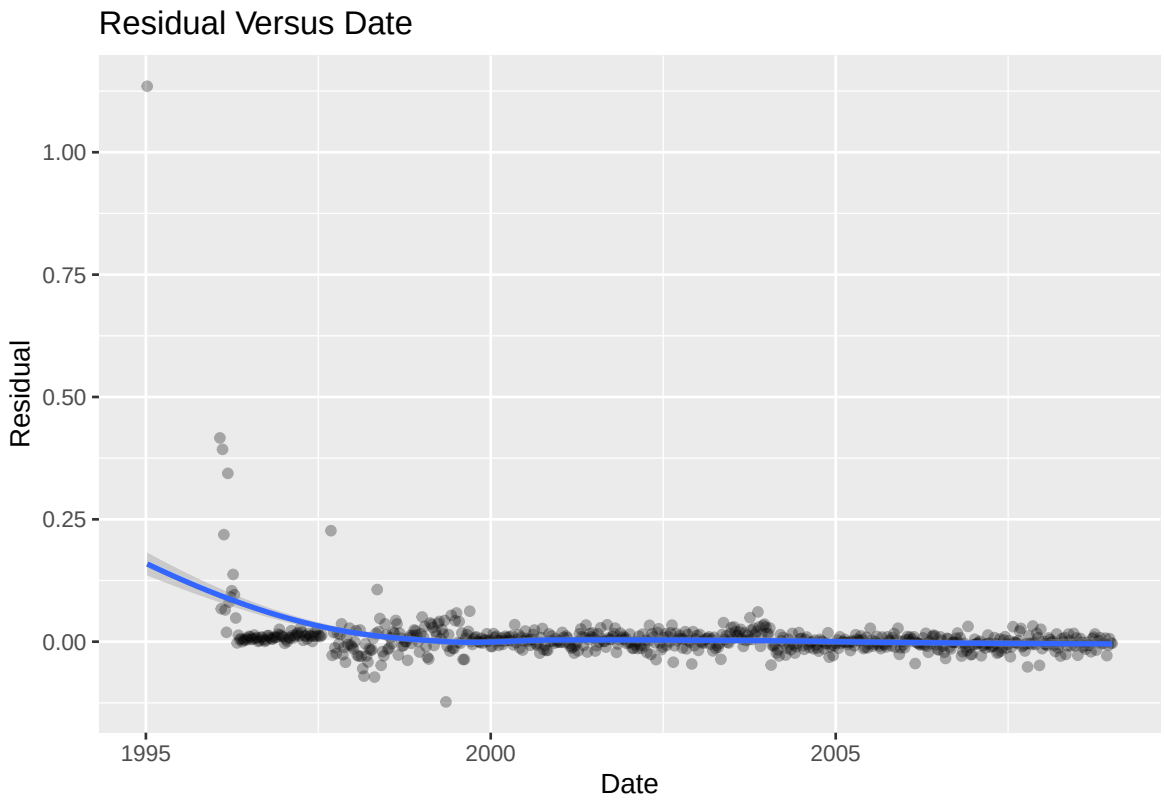


The plot shows that there is slight variation of the average rating over time, so there may be an increase in prediction accuracy if we add another term to the prediction model that accounts for this. The formula is (based on the course exercise):

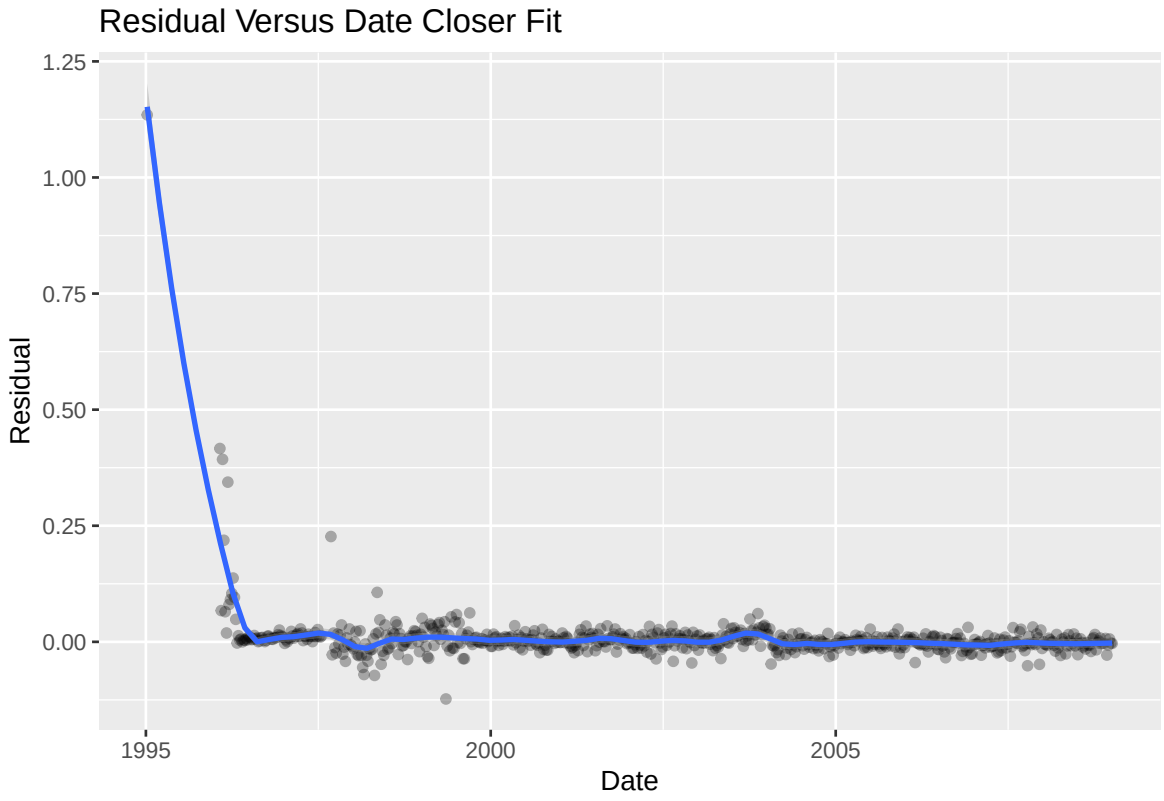
$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f(d_{u,i}) + \varepsilon_{u,i}$$

$d_{u,i}$ is the week that user u rated movie i and f is a smooth function.

The residuals, after removing μ and movie and user effects, look like this:



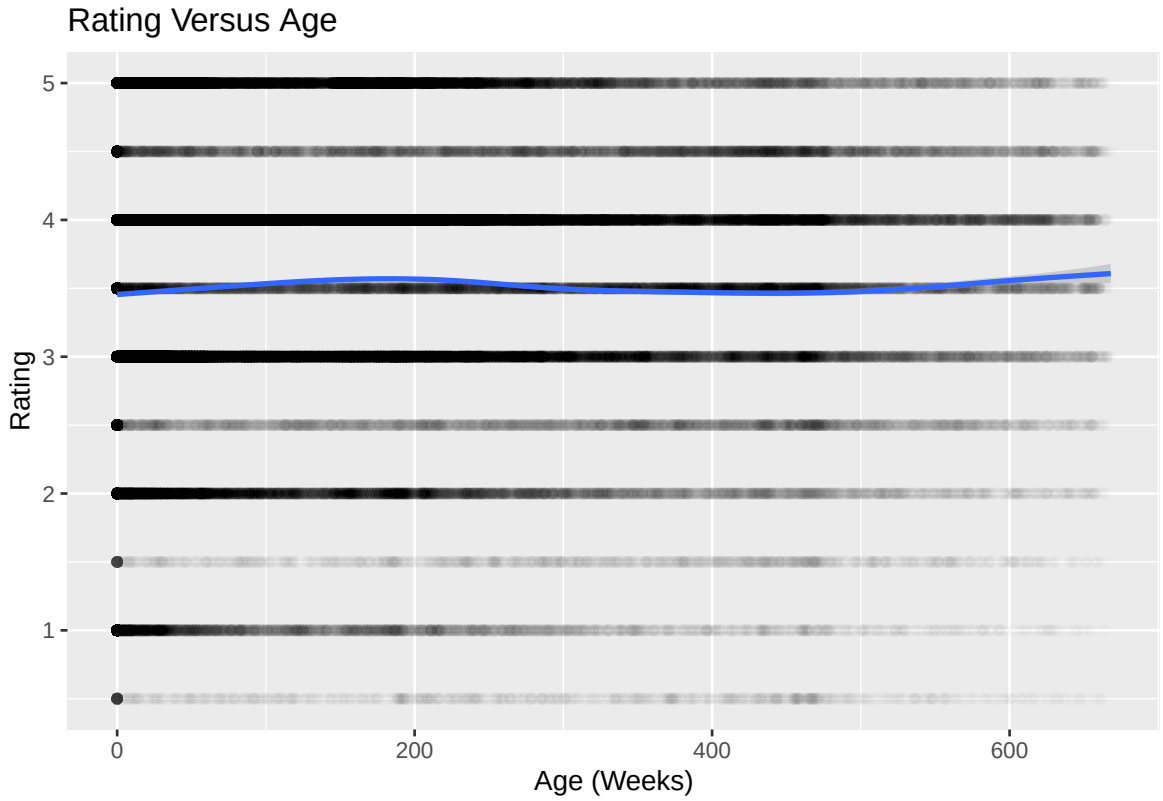
After removal of the movie and user effects, it can be seen that over time there is not that much variation and only small improvement is expected from the f term. The line is produced by `geom_smooth()`, which is using loess with a span value of 0.75, and doesn't fit the data very well. Using a span of 0.1 and degree of 2 produces the following:



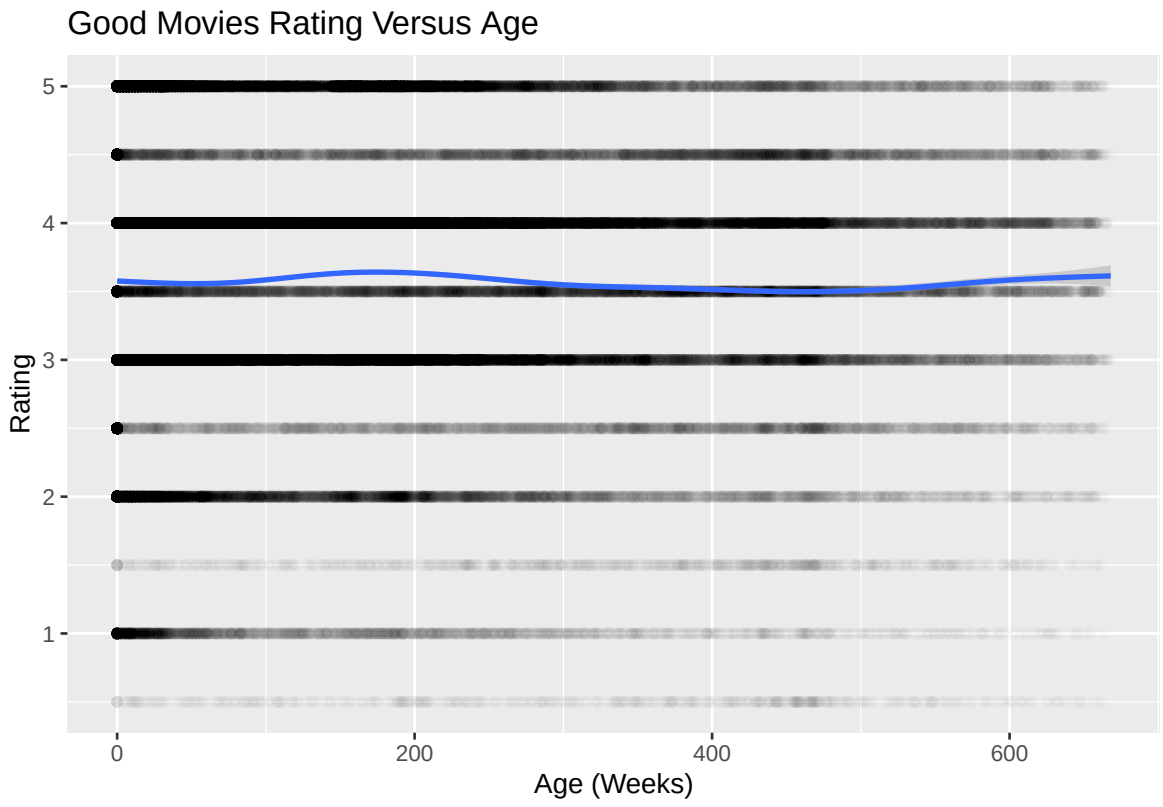
Using loess for f, the resulting RMSE value is 0.8641235, which is negligible improvement on the base regularized movie and user effect model with RMSE 0.8641362.

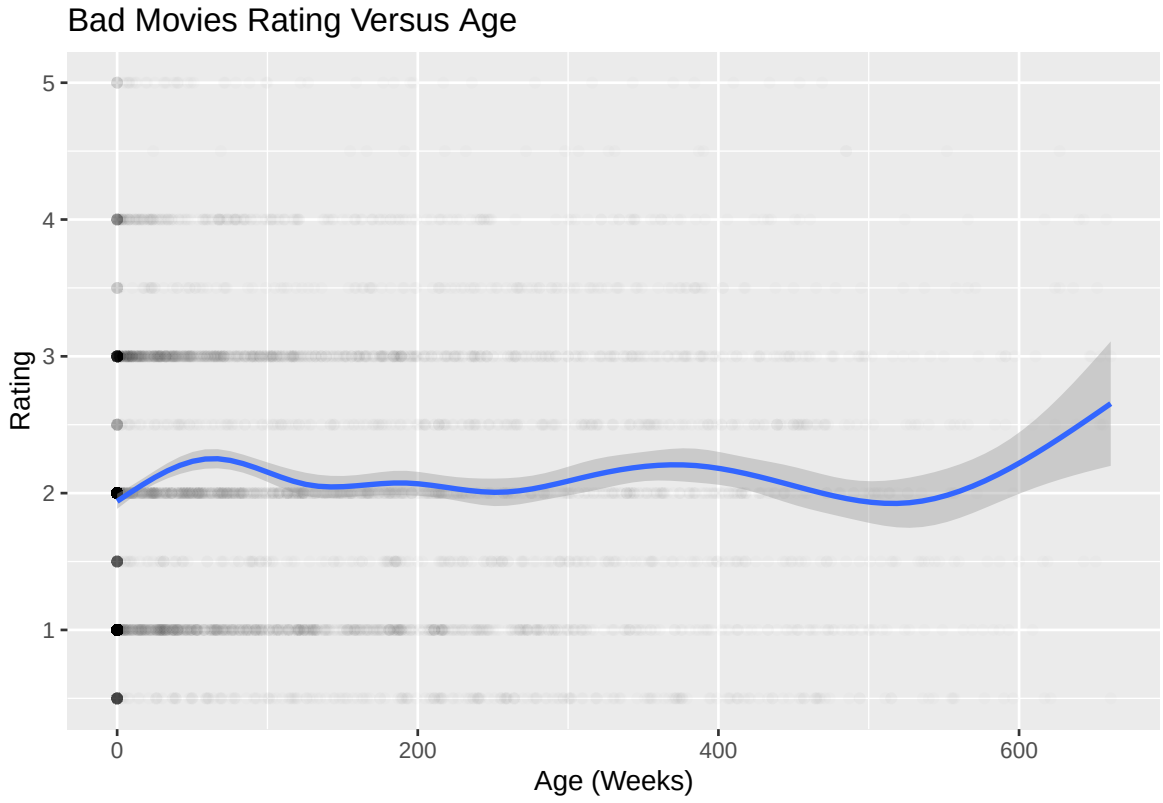
2.3.1 Movie Time Effect

Instead of examining rating against date, an alternative is to explore the relationship between age of a movie and the ratings. Approximate the release of a movie with the date it was first rated, then the age of a movie is the elapsed time between the timestamp of the rating, and the first rating timestamp. To speed up plotting, a sub sample of 10^5 rows of the `edx` data set is used in the following.



From the plot it can be seen that the lower right quadrant is not as densely occupied, indicating that the older a movie is, the less low rating values it receives and there is a slight upward trend in rating. This could be explained by people being willing to watch older movies that are considered good, but there is very little interest in watching older movies that are regarded as bad.





The above two plots split the data into two sets. One where the average rating of a movie is equal to or greater than 2.5, and the other below 2.5. The plot for the above average movies is similar to the first plot of all movies, while for the below average movies it is different. This indicates that more accurate predictions will likely be obtained by modelling per movie, rather than having a single function applied

to all movies. The overall model is then:

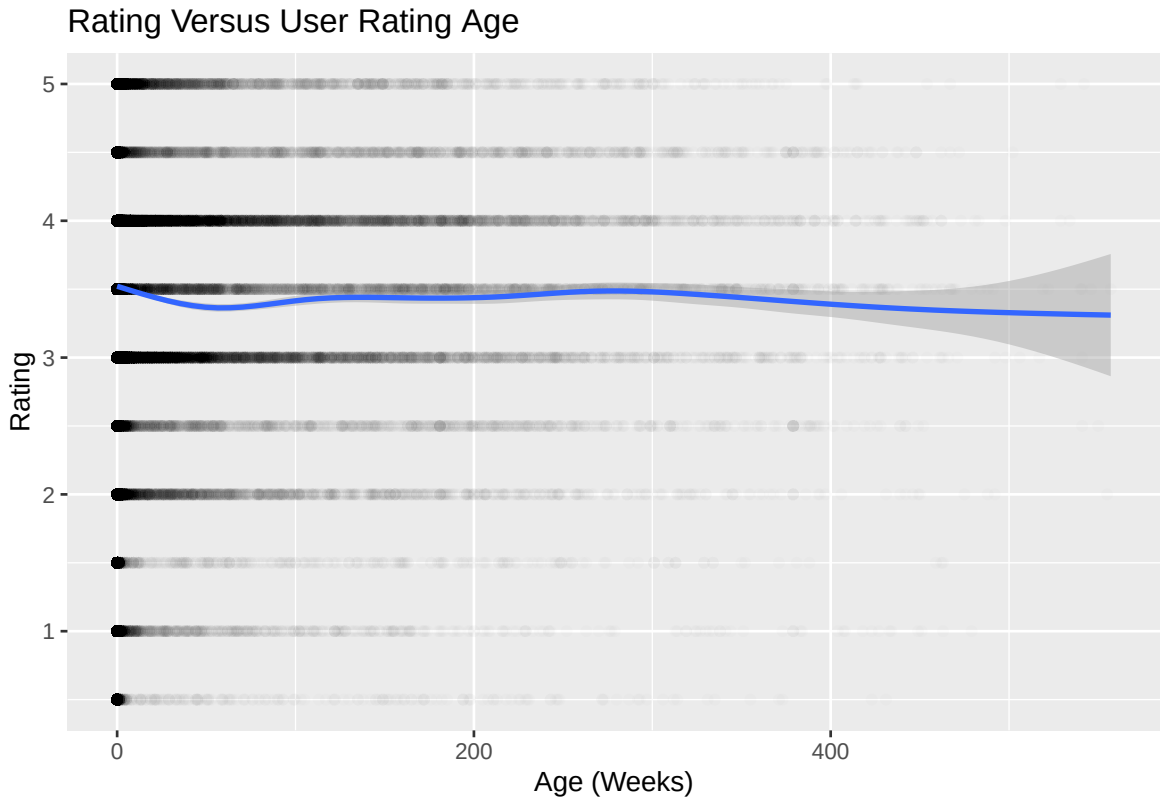
$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f_i(d_{u,i}) + \varepsilon_{u,i}$$

The steps involve converting the training data timestamp to `datetime` type, rounded to week resolution. Then computation of the residuals (rating value minus mu and movie and user effects), and loess was fitted on a per movie basis of rating against date. Note that for movies with too few reviews (arbitrarily chosen as 100), no loess model is fitted and those movies will have no movie time effect.

The RMSE calculation obtained after building the model against the training set, and using it to predict against the test set is 0.8600744, which is an improvement over the base regulated movie and user effects model.

2.3.2 User Time Effect

In a similar manner to the movie effect varying with time, there may be a time effect applicable to the user effect.



In the above plot of rating versus user age (age being the time since the user first rated a movie, not the user's actual age), there is a slight dip after the start, before moving back to the average. As per the movie time effect, it is likely to be more accurate to model per user, rather than a trend across all users.

Adding a term to the base model for this:

$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f_u(d_{u,i}) + \varepsilon_{u,i}$$

$f_u(d_{u,i})$ is a time effect per user. However there are 69878 users, and trying to fit loess models for them was beyond the available memory of the desktop PC used.

2.4 Genres

The genres column contains character string values, each value represents the genres the movie belongs to (the genres are separated by the "|" character). The genres are ordered alphabetically, so, for example if a movie belongs to the `Comedy` and `Romance` genres, it will be have the value `Comedy|Romance` and not `Romance|Comedy`.

Table 1: Movie Genres Sample

	title	genres
1	Boomerang (1992)	Comedy|Romance
2	Net, The (1995)	Action|Crime|Thriller
4	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	Stargate (1994)	Action|Adventure|Sci-Fi
6	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	Flintstones, The (1994)	Children|Comedy|Fantasy

These are the unique genre values:

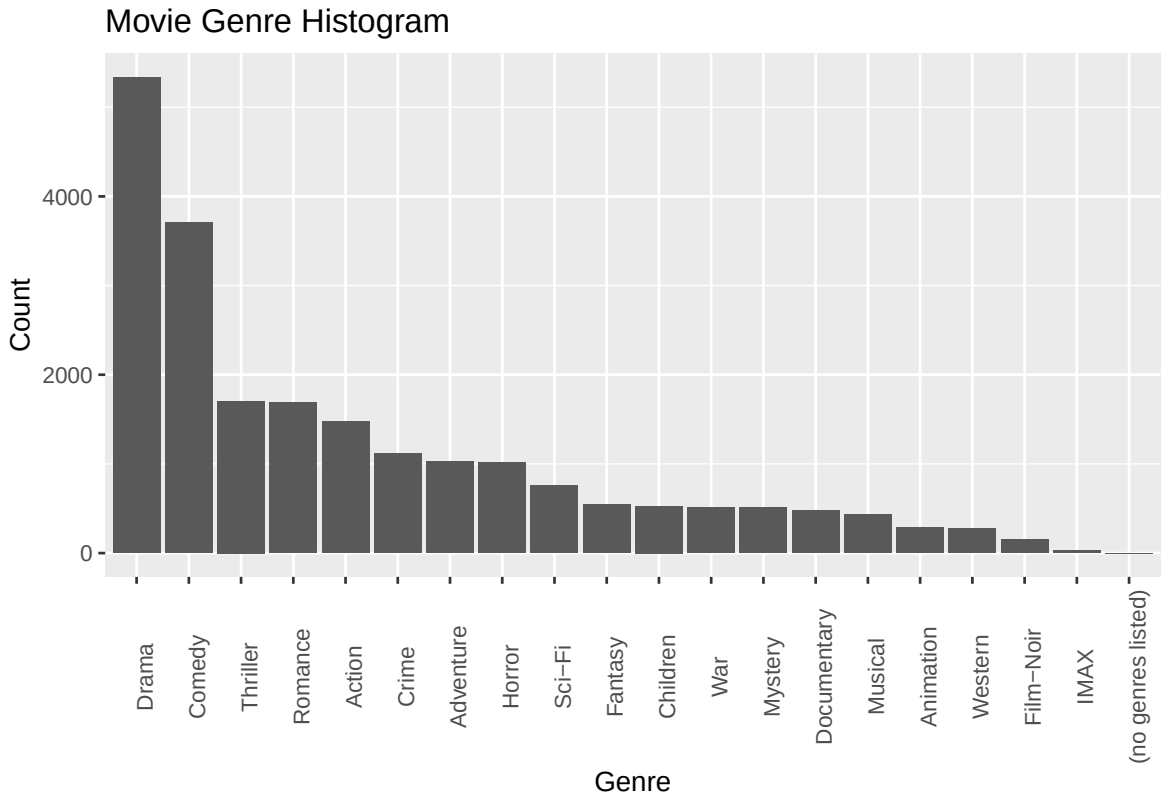
```
 [1] "Comedy"             "Romance"            "Action"
 [4] "Crime"              "Thriller"           "Drama"
 [7] "Sci-Fi"             "Adventure"          "Children"
[10] "Fantasy"            "War"                "Animation"
[13] "Musical"            "Western"            "Mystery"
[16] "Film-Noir"          "Horror"             "Documentary"
[19] "IMAX"               "(no genres listed)"
```

There is a special value `(no genres listed)`, which indicates a movie has no genres. There is only one movie with this value:

Table 2: Movies With No Genres

movieId	title	genres
8606	Pull My Daisy (1958)	(no genres listed)

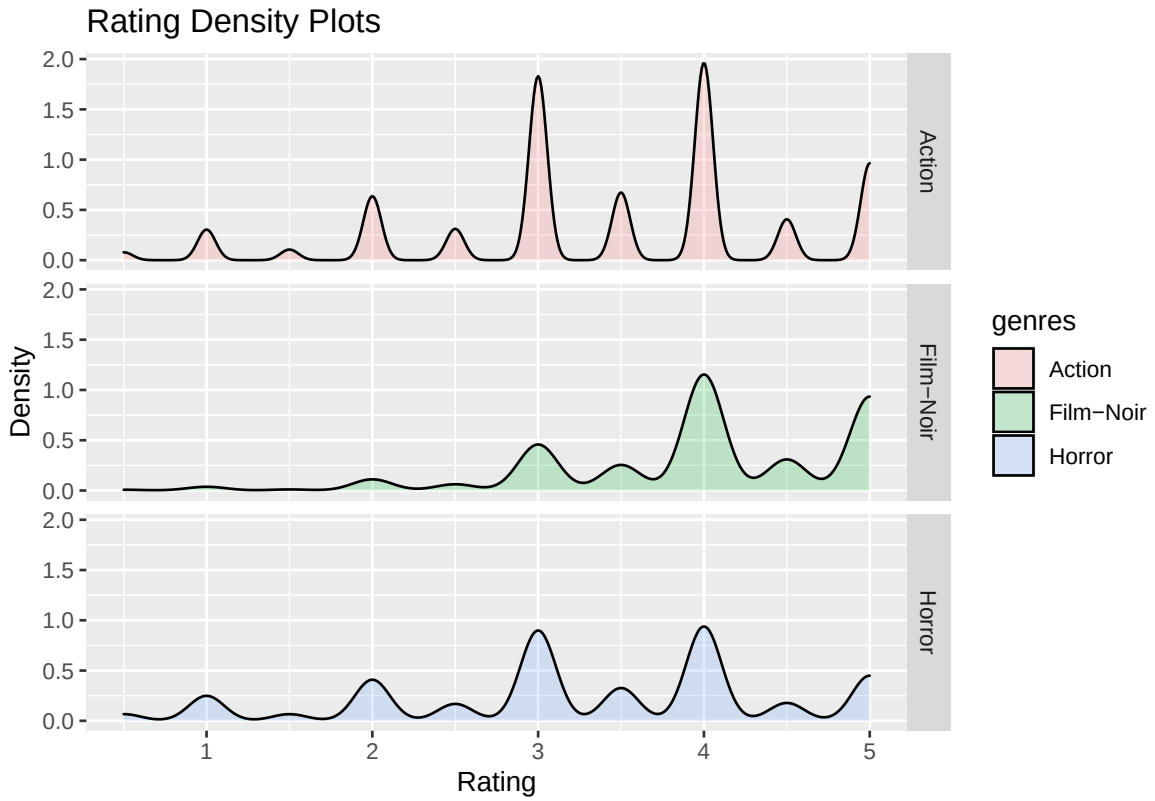
A histogram plot of genres shows comedies and dramas as the most popular genres.



Is there any relationship between genre and rating? The following box plot is based on a sub sample of 1 million rows from the `edx` data set.

From the graph it can be seen that there is a relationship between genre and rating. Film-Noir movies tend to be rated highly, while horror movies receive more lower ratings than other genres. Unlike other genres, ratings of 0.5 or 1 aren't considered outliers for horror movies.

As an alternative visualization, density plots of three genres, Horror, Action, and Film-Noir are created:



From the density plots, Action has high proportions of ratings of 3 and 4, while Film-Noir has high ratings and very little low ratings. Horror also has ratings of 3 and 4 as the highest density, however they are not as pronounced as with the Action genre.

The above plots show effects of genre aggregated across all users, however individual users have different preferences and accuracy is likely to be improved by modelling a per user genre effect. The formula for adding a user genre effect to the regularized movie and user model is (based on, but differing from the formula in the course exercise):

$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + \sum_{k=1}^{K} x_{i,k}\beta_{u,k} + \varepsilon_{u,i}$$

K is the set of genres, $\beta_{u,k}$ is the bias for user u and genre k, and $x_{i,k}$ is 1 if movie i belongs to genre k, otherwise 0. In other words, we are adding a user's genre bias to a movie only if the movie belongs to that genre.

Below the residual (difference between predicted value of the regularized movie and user model) is shown for a single row for user 1 and the movie Boomerang.

Table 3: Residual Movie Example

userId	movieId	title	genres	residual
1	122	Boomerang (1992)	Comedy|Romance	0.8373439

The movie Boomerang belongs to 2 genres, and the residual needs to be apportioned between the two. One way would be to simply divide the residual equally between the 2 genres. However if the user has rated a lot of comedies, but not many romances, then it would be better to apply a weight to each genre that reflects this.

Denote the number of times a user has rated a movie with genre k as $n_{u,k}$, then the weight to apply to

genre k for movie i is:

$$w_{i,u,k} = n_{u,k} / \sum_{k=1}^{K} x_{i,k} n_{u,k}$$

$x_{i,k}$ is defined as previously. The residual for a movie i, rated by user u can be expressed as:

$$r_{i,u} = \sum_{k=1}^{K} w_{i,u,k} \beta_{i,u,k}$$

The term $\beta_{i,u,k}$ represents the bias for genre k of user u for movie i. Then the user's bias for each genre, $\beta_{u,k}$ can be taken as the average of the values of $\beta_{i,u,k}$.

Table 4: Distribution of Residual Per Genre

title	genres	count	residual	bias
Boomerang (1992)	Comedy	9	0.8373439	0.5382925
Boomerang (1992)	Romance	5	0.8373439	0.2990514
Outbreak (1995)	Action	8	0.2774263	0.1056862
Outbreak (1995)	Drama	6	0.2774263	0.0792647
Outbreak (1995)	Sci-Fi	4	0.2774263	0.0528431
Outbreak (1995)	Thriller	3	0.2774263	0.0396323
Stargate (1994)	Action	8	0.3432932	0.1615498
Stargate (1994)	Adventure	5	0.3432932	0.1009686
Stargate (1994)	Sci-Fi	4	0.3432932	0.0807749
Star Trek: Generations (1994)	Action	8	0.3534137	0.1229265

The above table shows a portion of the results of calculation results for user 1. User 1 has rated 9 movies in the Comedy genre, and 5 in the Romance genre, so the residual of 0.837 for the movie Boomerang is apportioned into biases where Comedy has almost twice as much as Romance. The genre biases for user 1 are then calculated as the average bias per genre.

Table 5: User 1 Comedy Bias

title	genres	count	residual	bias
Boomerang (1992)	Comedy	9	0.8373439	0.5382925
Flintstones, The (1994)	Comedy	9	1.2051670	0.6380296
Forrest Gump (1994)	Comedy	9	-0.3188639	-0.1304443
Naked Gun 33 1/3: The Final Insult (1994)	Comedy	9	0.7409981	0.3922931
Beverly Hills Cop III (1994)	Comedy	9	0.9358819	0.4010922
Hot Shots! Part Deux (1993)	Comedy	9	0.7910298	0.3746983
Robin Hood: Men in Tights (1993)	Comedy	9	0.6783806	0.6783806
Sleepless in Seattle (1993)	Comedy	9	0.1497982	0.0674092
Aladdin (1992)	Comedy	9	0.0280106	0.0093369

The above table shows the biases for the Comedy genre for the Comedy movies rated by user 1. User one's bias for the genre Comedy is then the average of these values (0.3298987).

Note that computing these genre biases per user involves splitting each row into multiple rows, one per genre contained in the `genres` column, this leads to increased memory usage that is beyond the available memory of the desktop computer this report was produced on. To work around this, the training set is broken into smaller chunks for processing, rather than attempting the whole set at once.

The RMSE of the regularized movie and user effect and genre model is 0.8493713.

2.5 Full Model

$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f_i(d_{u,i}) + \sum_{k=1}^{K} x_{i,k}\beta_{u,k} + \varepsilon_{u,i}$$

The full model is based on the regularized movie and user effects, with a term added for movie effect over time, and user genre effects.

The RMSE for the full model is 0.8454856.

Summary table of all RMSE model values trained on `train_set` and evaluated against `test_set`:

Table 6: RMSE Results

method	RMSE
Just the average	1.06005
Movie effects	0.94296
Movie and User effects	0.86468
Regularized Movie Effect Model	0.94294
Regularized Movie + User Effect Model	0.86414
Regularized Movie + User Effect + Week Time Effect Model	0.86412
Regularized Movie + User Effect + Movie Time Effect Model	0.86007
Regularized Movie + User Effect + Genre Effect Model	0.84937
Regularized Movie + User Effect + Movie Time Effect + Genre Effect Model	0.84549

The full model that will be trained on the `edx` set and evaluated against the `validation` set is:

$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f_i(d_{u,i}) + \sum_{k=1}^{K} x_{i,k}\beta_{u,k} + \varepsilon_{u,i}$.

In addition, the course defined models, the movie time effect model, and the user genre effect models will also be trained and evaluated for comparison purposes.

3 Results

The table below shows the results of training the various models against the full `edx` set, and then predicting against the `validation` set. The results are similar to those obtained against the training and test sets when developing the models, indicating the models were not over fitted and generalized to data not seen (the `validation` set).

Table 7: Validation RMSE Results

method	RMSE
Just the average	1.06120
Movie effects	0.94391
Movie and User effects	0.86535
Regularized Movie Effect Model	0.94385
Regularized Movie + User Effect Model	0.86482
Regularized Movie + User Effect + Week Time Effect Model	0.86481
Regularized Movie + User Effect + Movie Time Effect Model	0.86044
Regularized Movie + User Effect + Genre Effect Model	0.84941
Regularized Movie + User Effect + Movie Time Effect + Genre Effect Model	0.84518

3.1 Course Models

3.1.1 Just The Average

Worst performance with a RMSE value of 1.0612018. The average rating across all movies is 3.5124652 which indicates a user bias towards viewing and rating movies that they will like.

3.1.2 Movie Effect

RMSE of 0.9439087 performs better than just the average of all movies. It represents the idea that some movies are widely regarded as "good", "bad" etc.

3.1.3 Movie And User Effect

Improvement on the movie effect model with a RMSE of 0.8653488, representing the idea that users have different standards when rating. Some users may be very critical and consistently give low ratings, others may be the opposite.

3.1.4 Regularized Movie Effect Model.

The regularization penalty term was calculated using k-fold cross validation with a value of 2.25, which lead to only a slight improvement over the non-regularized version, 0.9438521 versus 0.9439087.

3.1.5 Regularized Movie and User Effects Model

Using cross validation, the best λ value was 5. Regularization has somewhat of an improvement over the non-regularized version, 0.8648177 versus 0.8653488. But it is not as large of an improvement obtained by adding another bias term to the model.

3.2 New Models

The regularized movie and user effects model is the best performing course model, the new models developed used this as a basis to build upon.

3.2.1 Week Time Effect Model

A bias was added for modeling the effect of rating date (at a resolution of a week), and there was no significant change from the base model results. This indicates the date of a review, by itself, has no predictive power.

3.2.2 Movie Time Effect Model.

A term representing the movie effect over time was added, by fitting loess models per movie. An improved RMSE value of 0.8604436 was obtained compared to the base model value of 0.8648177. Generally, "good" movies will receive positive ratings long after they are first released as users will be attracted to watching older "good" movies, while "bad" movies will not attract any ratings the older they are.

3.2.3 Genres Model

The bias of individual users towards different genres was modeled, and the resulting RMSE of 0.849413 was a significant improvement over the base model value of 0.8648177. Movie genre has strong predictive power on an individual user basis.

3.2.4 Full Model

The full model (base model with genre and movie time effects) gave the best RMSE value of 0.8451831. The largest components contributing to this are the regularized movie and user effects. These present an improvement of approximately 0.196 over just taking the average rating. The user genre effect is the next largest contributing roughly 0.015 improvement, followed by the movie time effect 0.004.

4 Conclusion

In this report, the movie recommendation system introduced in **PH125.8x: Data Science: Machine Learning** was improved upon by adding further terms that incorporated more features available in the MovieLens data set. After exploring the data, two further terms were added to form a new model:

$$Y_{u,i} = \mu + b_i(\lambda) + b_u(\lambda) + f_i(d_{u,i}) + \sum_{k=1}^{K} x_{i,k}\beta_{u,k} + \varepsilon_{u,i}$$

One term represents the effect of time on movie bias, and was modeled using loess. The other term was for representing individual user biases towards movie genres. A final RMSE value of 0.8451831 was obtained by training on the `edx` set and evaluating against the `validation` set.

An attempt was made to model the effect of time on user bias, however the memory requirements were beyond the available memory of the desktop PC used to develop this report. With more powerful hardware, this could possibly be explored. Further techniques could also be tried with better hardware, such as matrix factorization.

In fitting loess models for movie bias over time, an arbitrary value of 100 was used as a cut off point. Movies with less than this number of data points didn't have a loess model fitted. This is a value that could be tuned to obtain an optimal value. Models other than loess could also be explored for modelling biases over time.