MovieLens Capstone

CT

14/03/2021

Introduction

This project will create a movie recommendation system using the MovieLens dataset, and a machine learning algorithm.

About the dataset

The dataset used is the 10M version of the MovieLens dataset. The dataset contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service MovieLens. Users were selected at random for inclusion. All users selected had rated at least 20 movies. Each user is represented by an id, and no other personal identifiable information is provided. More information on the dataset can be found here: https://grouplens.org/datasets/movielens/10m/

Goal

The machine learning algorithm will use the inputs in one subset (Training Set) to predict movie ratings in another set (Validation Set) A number of algorithms will be trialed to determine the optimum mixture of features to obtain a movie rating prediction with the lowest Root Mean Square Estimate (RMSE). **The goal RSME is <0.86490**.

Note: The final validation dataset will only be used at the end of the project to assess the final model. It will not be used to test the RMSE of multiple models during model development.

Key Steps

Key Steps in the project:

1. Investigate the structure of the training data set, generate plots to visualise data where required.
2. Pre-process the data
   - Clean/Wrangle data
   - Standardise or transforming predictors
   - Remove predictors that are not useful, are highly correlated with others, have very few non-unique values, or have close to zero variation.
3. Split the Training (edx) data into Training and Test set in order to evaluate algorithms
4. Build a base algorithm, and include additional features/biases and/or regularisation to optimise
5. Summarise data, choosing optimal algorithm based on training/test data
6. Apply algorithm to validation dataset

Method/Analysis

Library

The following packages/versions have been used, this includes dependencies:

```
package packages
  caret   6.0.86
```

```
 data.table   1.13.6
      dplyr    1.0.4
    forcats    0.5.1
    ggplot2    3.3.3
      knitr     1.31
    lattice  0.20.41
  lubridate  1.7.9.2
      purrr    0.3.4
      readr    1.4.0
    stringr    1.4.0
     tibble    3.0.6
      tidyr    1.1.2
  tidyverse    1.3.0
```

Initial Data Load

Data downloaded from the 10m Dataset has two files: Ratings and Movies. The initial data wrangle extracts these files, and stores them into a data-frame. This data-frame is then split into two: 'edx' (Model Data) and 'validation' (Final Validation Data). The validation set is set at 10% of the MovieLens data. The Model Data is further split into two: Train Data (used for training the algorithms) and Test Data (for verifying the models), with the Test data set at 10% of the Train data.

Data Structure

The structure of the data is shown below. Its worth noting that the year the movie was released is not a separate data field, and is a part of the Movie Title. As an outcome of this investigation the movie release year has been split into its own column so further analysis can be performed to determine if this should be a predictor. The rated year (timestamp) needs to be converted into a Year format (YYYY) to complete further analysis.

In addition, the genres are split by a pipe delimiter (|) - at this stage no manipulation of this data-field will be performed.

```
Classes 'data.table' and '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 838984885 8389
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Adve
```

The data-set contains the following number of rows and columns:

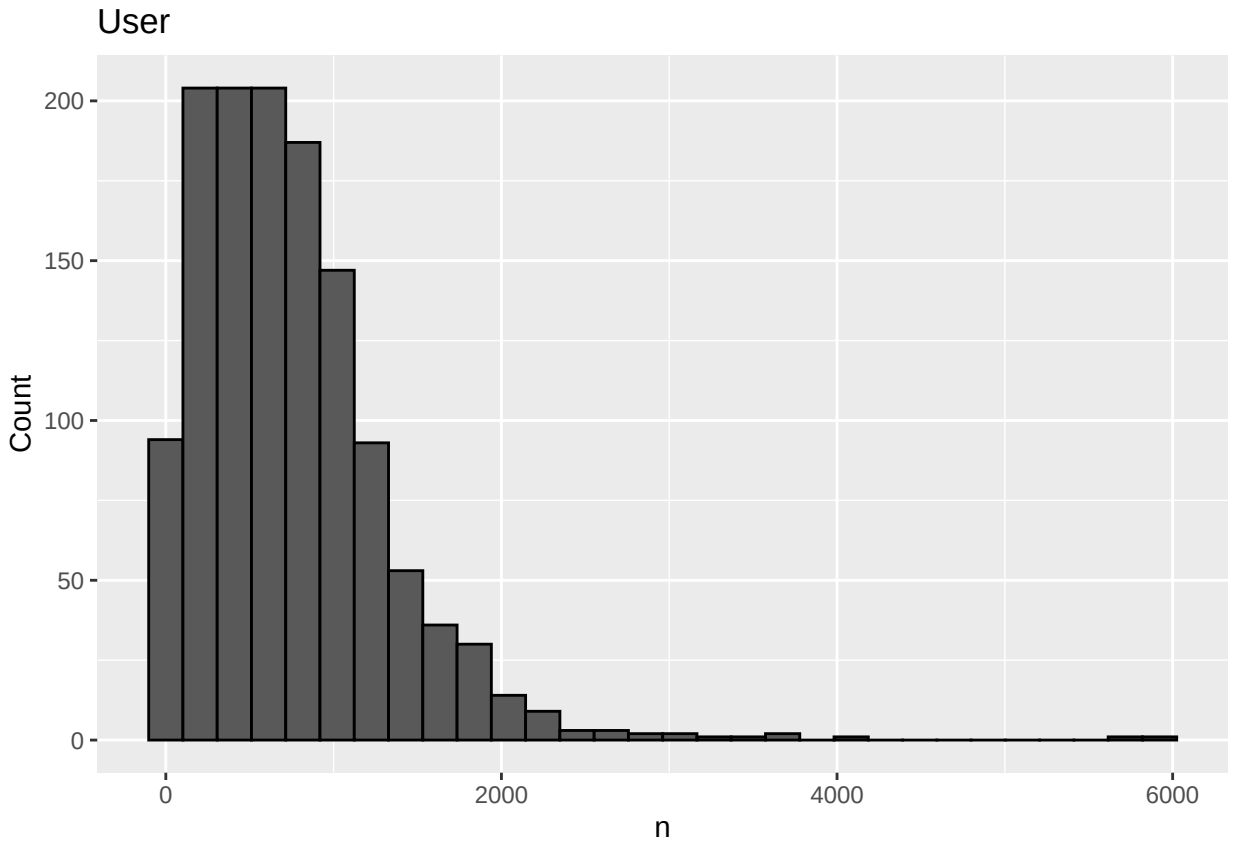
x
9000055
6

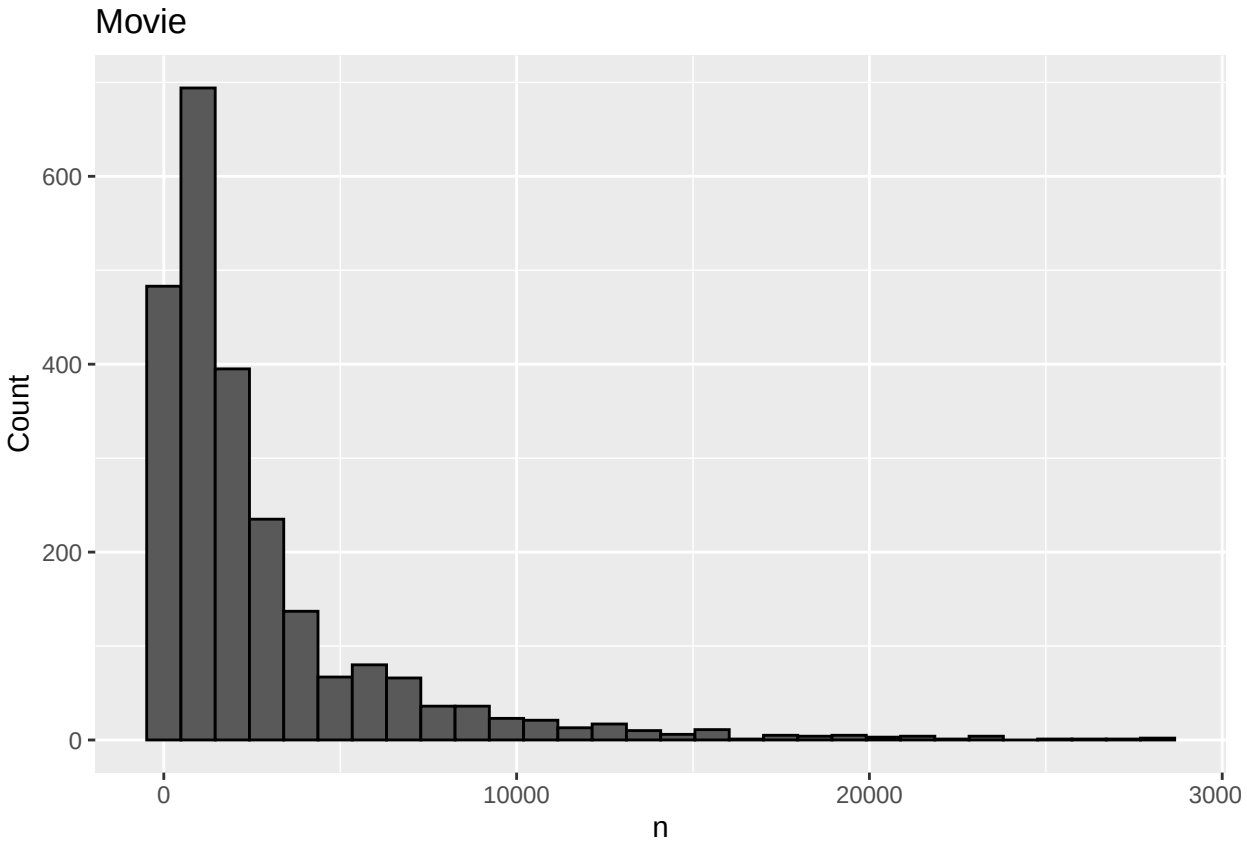
There are the following number of unique rows for the predictors:

n_users	n_movies	n_genre	n_year_rel	n_year_rate
69878	10677	797	94	15

The following data shows the distribution of users and movies. We can see some movies are rated a lot more

than others, and also some users rate more movies than other. This indicates that regularisation might be required when using these as predictors. Regularisation can optimise the model by penalising large estimates that are formed using small sample sizes.

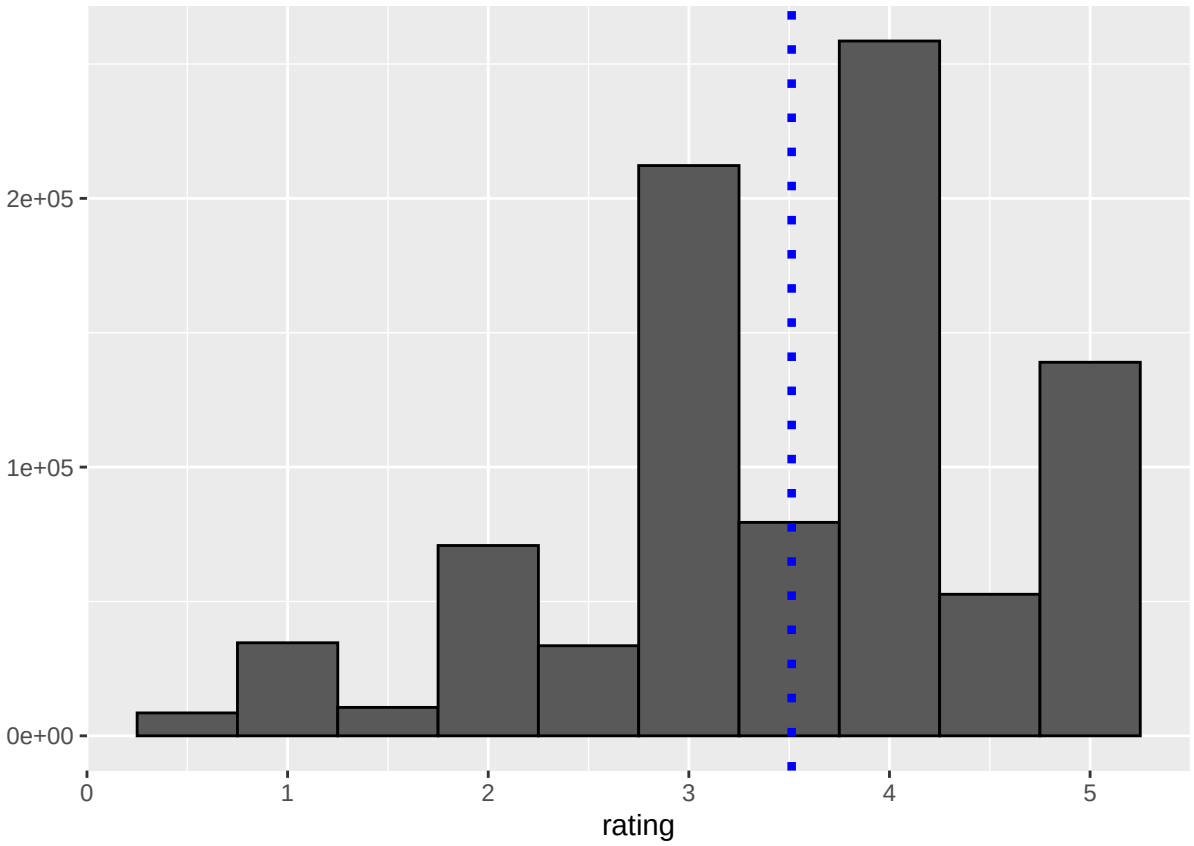




Results

First Model

In the first model we predict the same rating for all movies regardless of user. The predicted rating is the average rating across all rows. The chart below shows the distribution of ratings, the average (blue dotted line).

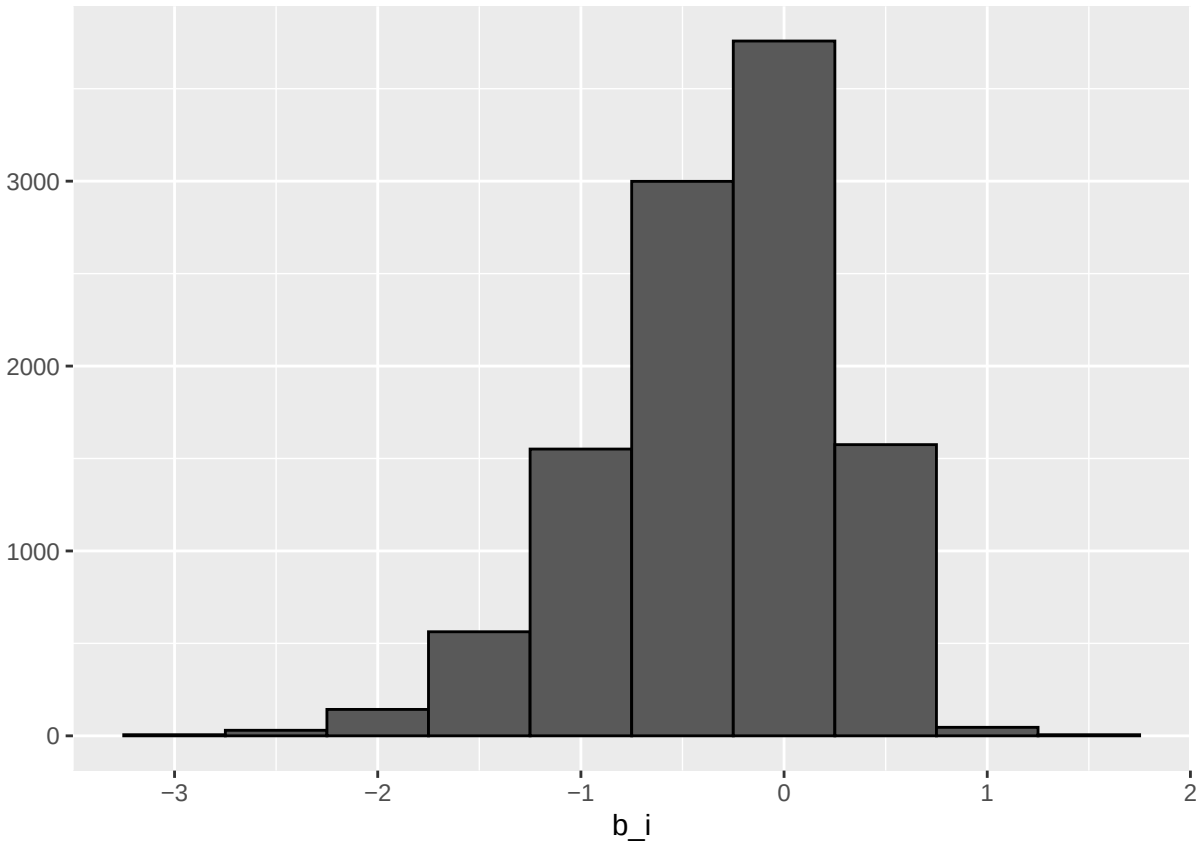


method	RMSE
Target	<0.86490
Average	1.06005

This RSME is in this first model is above the target of 0.86490, and so further optimisation is required.

First Bias (Movie ID)

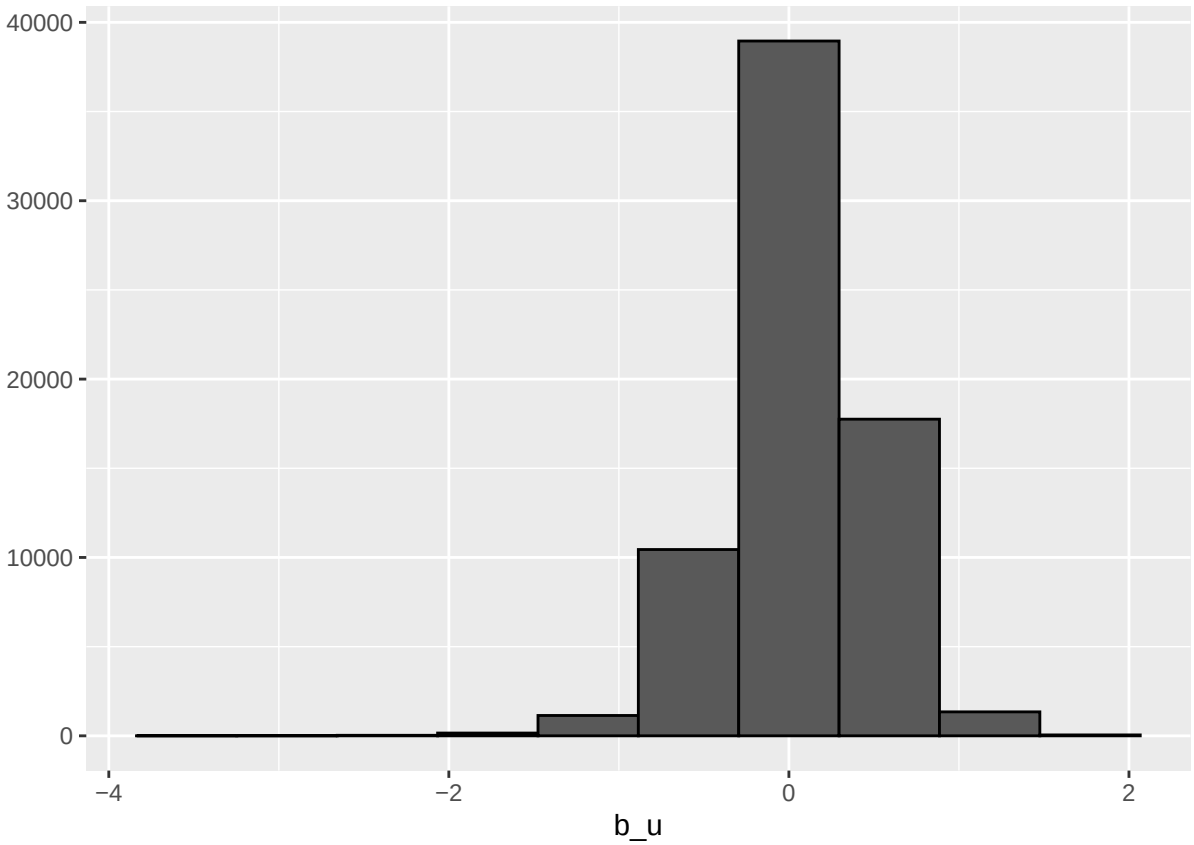
Next we add in our first bias: the Movie ID. This bias shifts the estimated user rating based on the average rating of each movie (based on all users). From the plot we can see how this distributes ratings.



method	RMSE
Target	<0.86490
Average+Movie	0.94296

Second Bias (User ID)

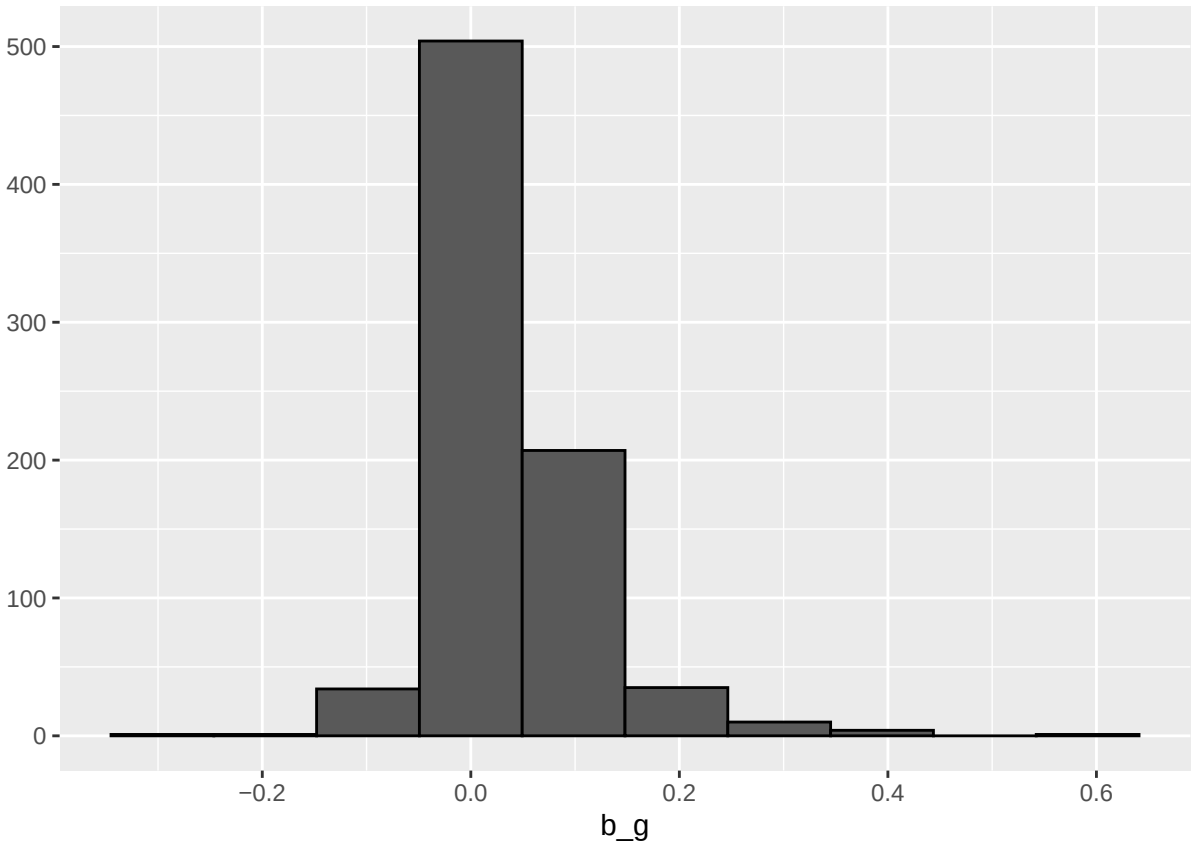
Now we add in our second bias: the User ID. This will shift the estimated rating based on the average rating from each users. From the plot we can see how this shifts the ratings.



method	RMSE
Target	<0.86490
Average+Movie+User	0.86468

Third Bias (Genre)

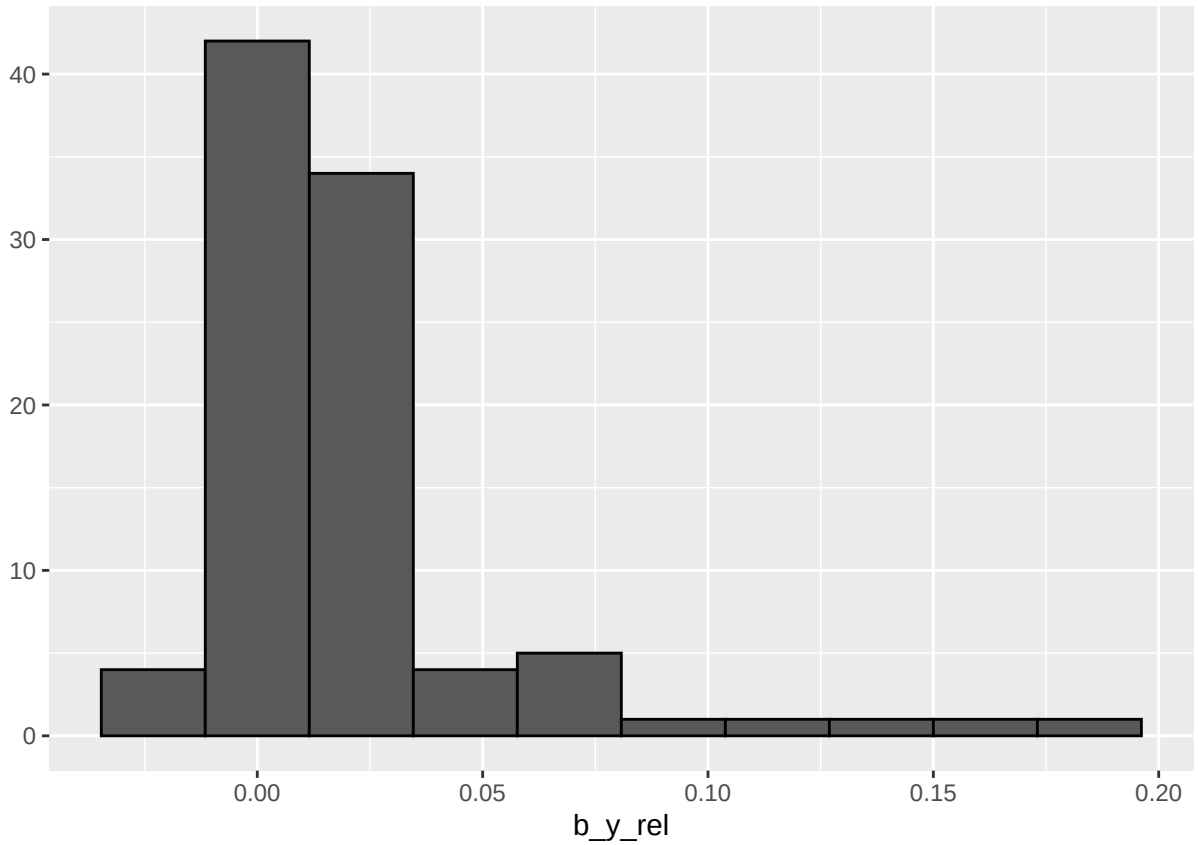
Now we add in our third bias: the Genre. This will shift the estimated rating based on the average rating for each genre. From the plot we can see this further shits the ratings.



method	RMSE
Target	<0.86490
Average+Movie+User+Genre	0.86432

Fourth Bias (Release Year)

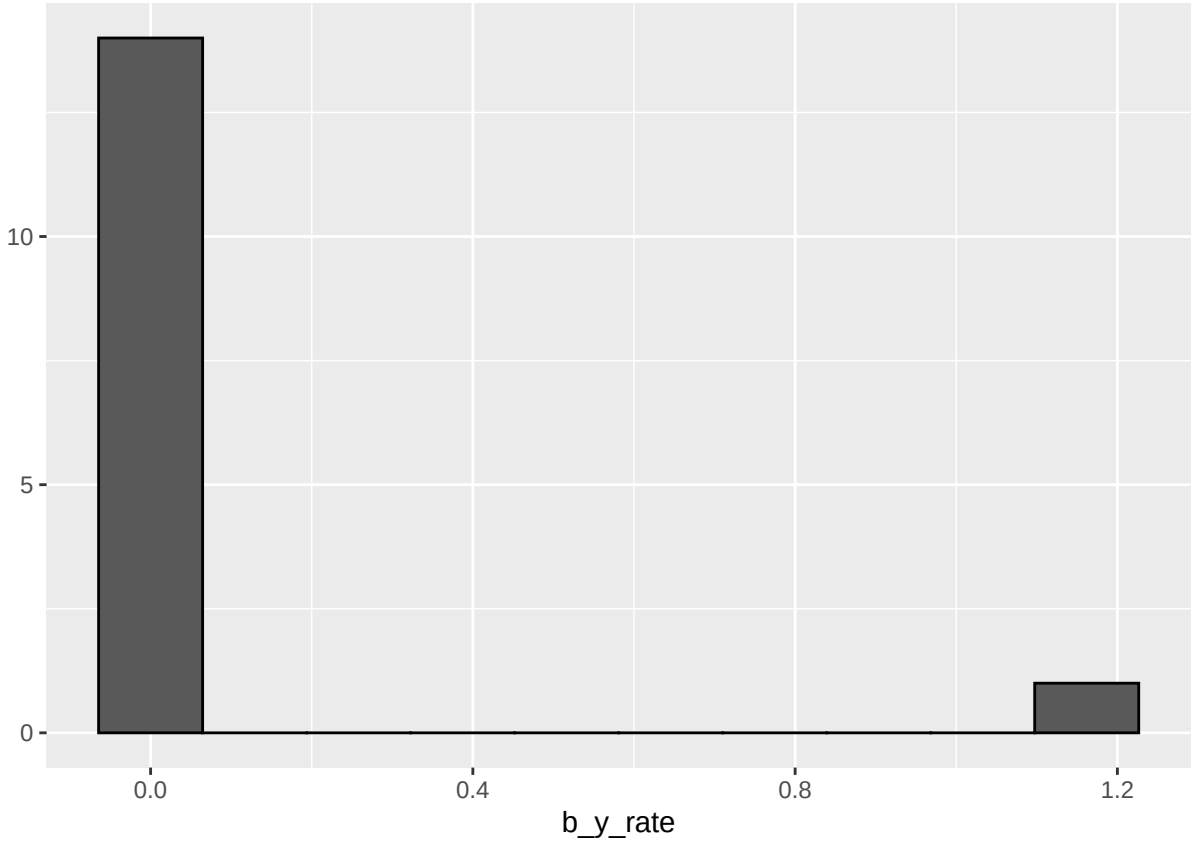
Now we add in our fourth bias: the Year Released. This will shift the estimated rating based on the year the movie was released. From the plot we can see this shifts the ratings.



method	RMSE
Target	<0.86490
Average+Movie+User+Genre+Year Released	0.86413

Fifth Bias (Year Rated)

Now we add in our fifth bias: the Year Rated This will shift the estimated rating based on the year the movie was rated. From the plot we can see this improves shifts the ratings.



method	RMSE
Target	<0.86490
Average+Movie+User+Genre+Year Rel+Year Rated	0.86407

Summary of Model Biases

Summary of all biases shown below. Summary shows how each bias improves the RMSE, reaching a level below the target.

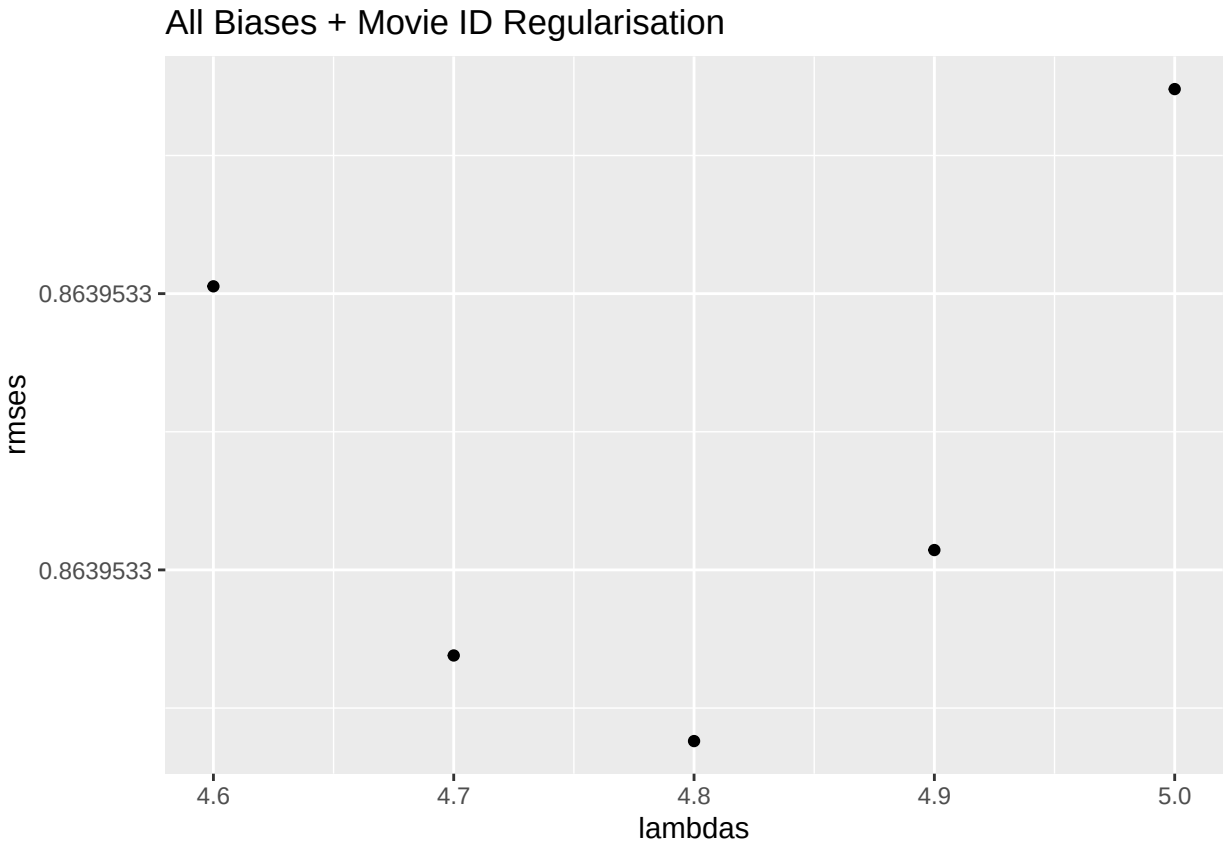
method	RMSE
Target	<0.86490
Average	1.06005
Average+Movie	0.94296
Average+Movie+User	0.86468
Average+Movie+User+Genre	0.86432
Average+Movie+User+Genre+Year Released	0.86413
Average+Movie+User+Genre+Year Rel+Year Rated	0.86407

Regularisation Optimisation

Regularisation can optimise the model by penalising large estimates that are formed using small sample sizes. The following explores how regularisation impacts the final RSME value. Regularisation is considered a tuning parameter, with small changes applied to this parameter, until the lowest RSME is identified.

Movie ID Regularisation

The chart shows how Movie ID regularisation improves the final RSME. The optimal tuning parameter is identified as the minimum value on the chart.

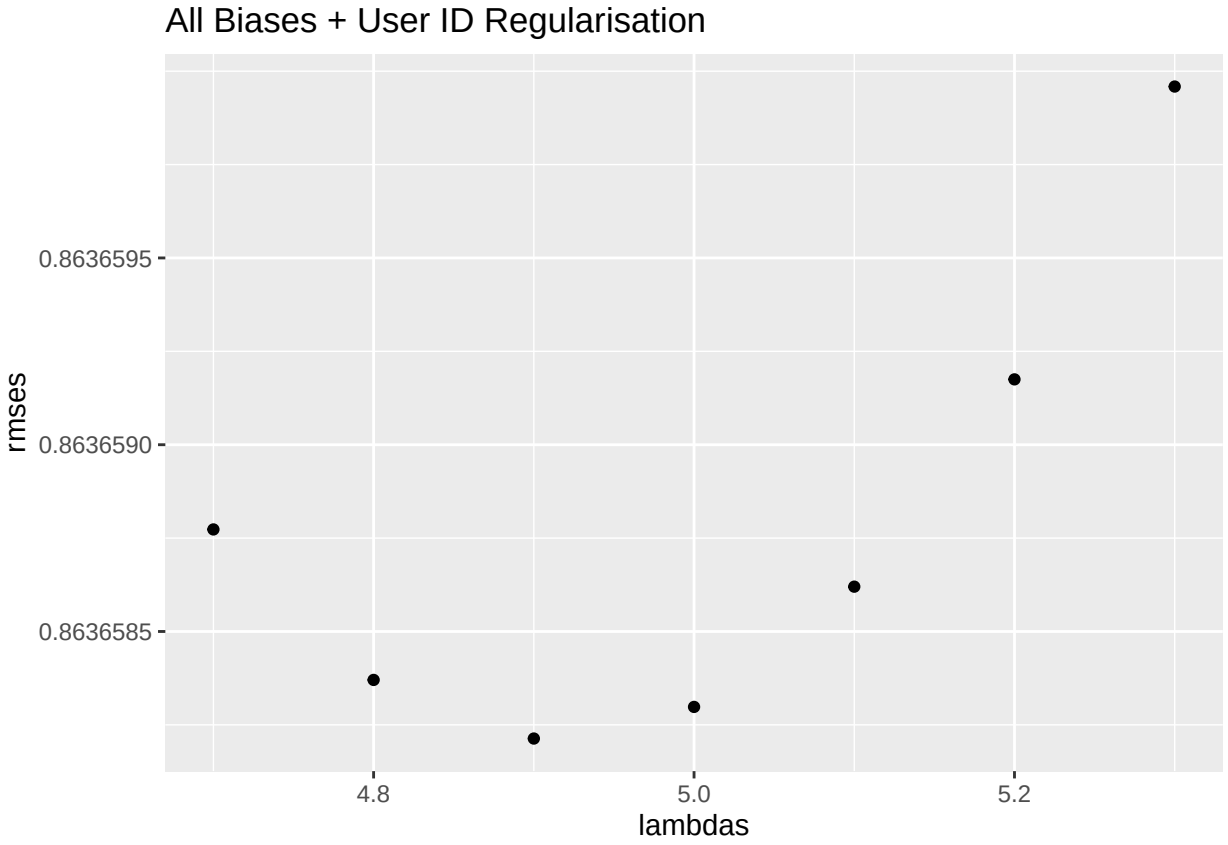


method	RMSE
Target	<0.86490
All Biases + Movie ID Regularisation	0.86395

This tuning parameter used for the Movie ID regularisation is:

x
4.8

User ID Regularisation

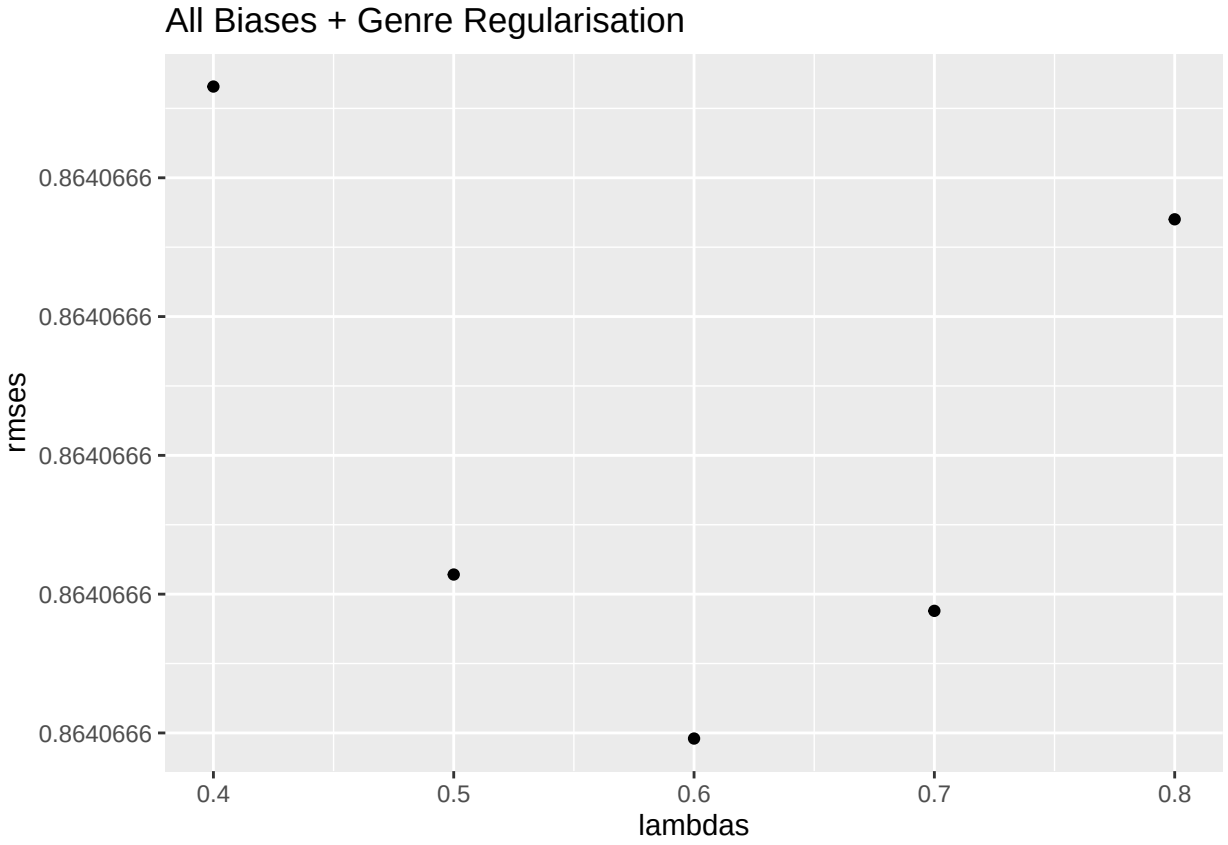


method	RMSE
Target	<0.86490
All Biases + User ID Regularisation	0.86366

This tuning parameter used for the User ID regularisation is:

x
4.9

Genre Regularisation



method	RMSE
Target	<0.86490
All Biases + Genre Regularisation	0.86407

This tuning parameter used for the Genre ID regularisation is:

x
0.6

Combined User ID and Movie ID Regularisation

Now check with the values combined

method	RMSE
Target	<0.86490
All Biases + All Regularisation	0.86354

Based on the initial data investigation we determined that no values were below 0.5, and no values were above 5. Based on this information we should also adjust the final dataset to ensure no values are above or below this values. Where a result is less than or equal to 0, then it is changed to 0.5. Similarly, results above 5 are changed to 5.

method	RMSE
Target	<0.86490
All Biases + All Regularisation + Out of Range Adjustment	0.86343

Summary of all models

The below table summarises the RSME for all models that have been evaluated.

method	RMSE
Target	<0.86490
Average	1.06005
Average+Movie	0.94296
Average+Movie+User	0.86468
Average+Movie+User+Genre	0.86432
Average+Movie+User+Genre+Year Released	0.86413
Average+Movie+User+Genre+Year Rel+Year Rated	0.86407
All Biases + Movie ID Regularisation	0.86395
All Biases + User ID Regularisation	0.86366
All Biases + Genre Regularisation	0.86407
All Biases + All Regularisation	0.86354
All Biases + All Regularisation + Out of Range Adjustment	0.86343

Final Evaluation Using Validation Dataset

Based on the optimisations above, we now use a model that applies all biases, optimises using the regularisation of Movie ID and User ID, and remove the out of range data. As the final step in the project this model is applied to the validation data set to evaluate the final RSME.

method	RMSE
Target	<0.86490
Validation Data	0.86448

Conclusion

The final RSME was:

method	RMSE
Target	<0.86490
Validation Data	0.86448

This is below the target of 0.86490, and so our model is working as per the initial scope of the project. This model uses biases, regularisation and a range cut-off to optimise the model.

Future work could explore the use of a matrix factorisation technique, or other machine learning algorithms such as Collaborative Filtering or Neural Networks.

Run time was a limitation with this model, with compile time exceeding 30 minutes even using caching techniques, making it difficult to quickly iterate and trial different models. Updated hardware or a reduced data-set is necessary to further investigate different modeling techniques.

Table 21: Code Running Time (min)

x
25.8