HarvardX Capstone Project - Recommendation System

Frank Valdivia

2025-05-22

1. INTRODUCTION

This goal of this project is to build a recommendation system for Netflix. This recommendation system can predict which movies users would like to watch. Based on the history of movie ratings by users, the system can suggest movies the users might be most interested in.

In this project, the MovieLens dataset from Netflix will be used to create the recommendation system.

The dataset is pulled from the dslabs package and has millions of ratings. A subset of 10 million ratings will be used to reduce computing time.

Machine learning methods will be used, which means that two sub datasets will be created during this process:

A train set will be generated with the name: edx; this dataset will be used to train the model.

A test set will be generated with the name: final_holdout_test; this dataset will be used to test the model trained using the train dataset.

The Root Mean Squared Error (RMSE) will be used to calculate the error between predictions and true values in the test set (final_holdout_test set)

During this process, several sub dataset will be created and are explained in the Methods section.

2. ANALYSIS AND METHODS

the overall process are as follows:

2.1. loading libraries

2.2. loading files from server and generating datasets

2.3. Analysis and methods

2.1 loading libraries

To run the report the following R libraries need to be previously installed:

- library(tidyverse)
- library(caret)
- library(ggplot2)
- library(dplyr)
- library(gridExtra)

In this step above libraries are loaded

2.2 loading files from server and generating datasets

The files are downloaded, unzipped and used to create data sets.

Movies Ratings and Movies are pulled from:

http://files.grouplens.org/datasets/movielens/ml-10m.zip

The following datasets are created and will be used in training the different models.

- movielens: Movie ratings including Movie information
- ratings: Movie ratings with MovieId and UserId
- movies: Movie information
- edx:Train dataset to be used to train every model
- final_holdout_test: Test dataset to be used to test the model

Following are the structure and number of rows that each of these datasets have:

```
##
## Movielens Dataset: Head
```

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	231	5	838983392	Dumb & Dumber (1994)	Comedy
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi

```
##
## Movielens Dataset: Number of Records
```

dim(movielens)
10000054

```
##
## Ratings Dataset: Head
```

userId	movieId	rating	timestamp
1	122	5	838985046
1	185	5	838983525
1	231	5	838983392
1	292	5	838983421
1	316	5	838983392
1	329	5	838983392

```
##
## Ratings Dataset: Number of Records
```

dim(ratings)
10000054

```
##
## Movies Dataset: Head
```

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action|Crime|Thriller

```
##
## Movies Dataset: Number of Records
```

dim(movies)
10681

```
##
## edx (train set) Dataset: Head
```

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

```
##
## edx (train set) Dataset: Number of Records
```

dim(edx)
9000055

```
##
## final_holdout_test (test set) Dataset: Head
```

userId	movieId	rating	timestamp	title	genres
1	231	5	838983392	Dumb & Dumber (1994)	Comedy
1	480	5	838983653	Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller
1	586	5	838984068	Home Alone (1990)	Children|Comedy
2	151	3	868246450	Rob Roy (1995)	Action|Drama|Romance|War
2	858	2	868245645	Godfather, The (1972)	Crime|Drama

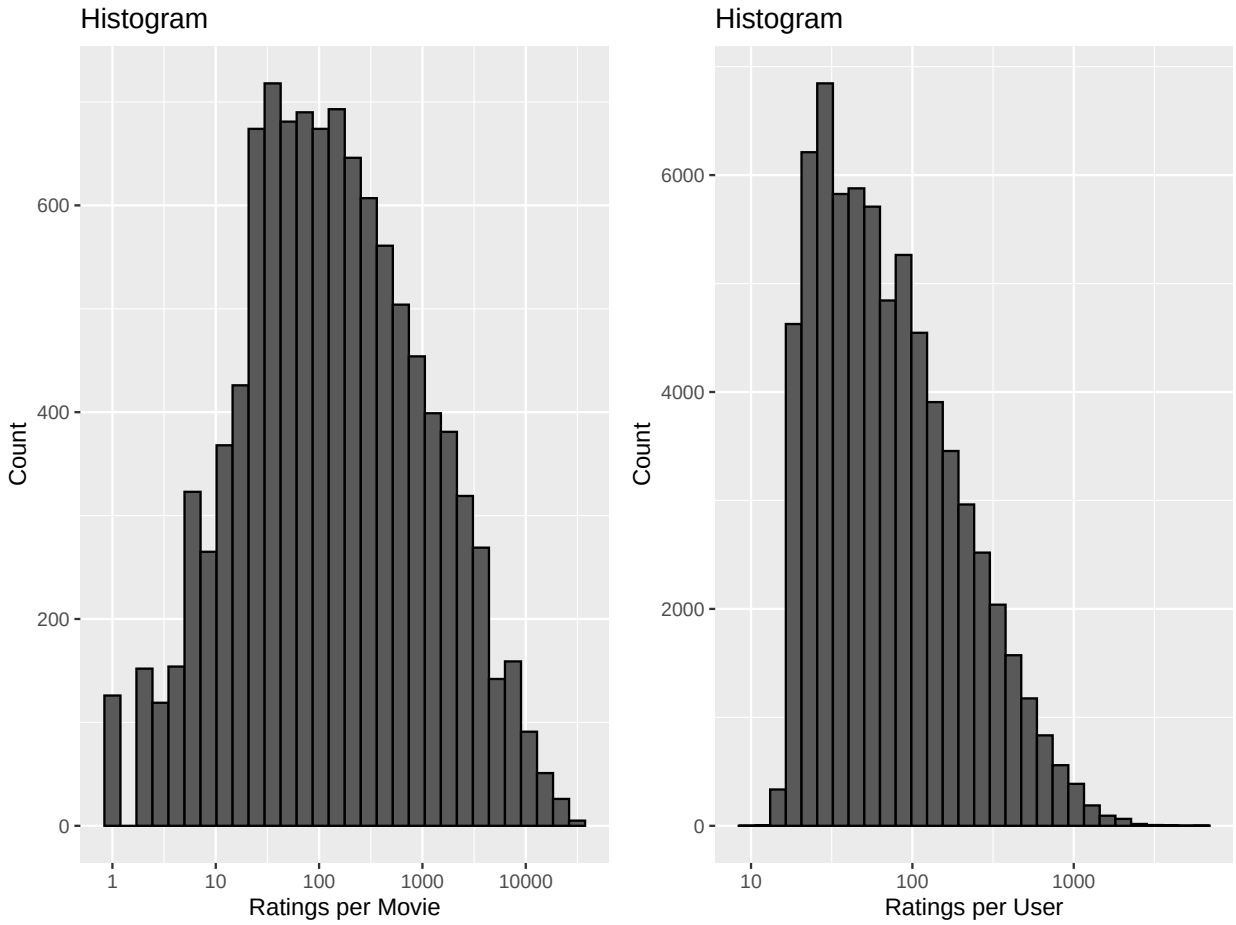
userId	movieId	rating	timestamp	title	genres
2	1544	3	868245920	Lost World: Jurassic Park, The (Jurassic Park 2) (1997)	Action|Adventure|Horror|Sci-Fi|Thriller

```
##
## final_holdout_test (test set) Dataset: Number of Records
```

dim(final_holdout_test)
999999

2.3 Analysis and methods

Following are the Histograms of Ratings counted by Movie and by User taken from the Train dataset (edx)



Some general characteristics of the data:

- They seem to be normally distributed.
- Some movies have more ratings than others.
- Some users rate more movies than others.
- There is no significant presence of outliers.

The models will use ratings from movies, users, and movies & users.

2.3.1 Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE), which is the square root of the mean squared error (MSE), will be used to estimate the error of every method.

Six models will be built and each model assessed against its value of RMSE.

2.3.2 First model: Y = mu_hat

The predictor of the first model is just the mean (mu_hat) of ratings in the train set (edx)

y = mu_hat

RMSE can be calculated using the test set (final_holdout_test) and mu_hat as predictor.

```
## mu_hat:  3.512465
```

```
## mu_hat_rmse:  1.061202
```

Any number other than mu_hat would result in a higher RMSE. A prediction and RMSE can be calculated using mu_hat_0.1, which is equal to mu_hat + 0.1

```
## mu_hat+0.1:  3.612465
```

```
## mu_hat+0.1 _rmse:  1.065944
```

As expected, mu_hat+0.1 _rmse is higher than mu_hat_rmse

The first model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.061202	

2.3.3 Second model: Y = mu_hat + Movie_bias

Every Movie has its own Rating mean, which might be different from the overall mean or different from other Movie Rating mean. That is due to the bias (or effect) of every movie on the ratings.

In the second model, the Movie bias or effect will be incorporated.

The Bias will be calculated for every Movie as the difference between the Mean Rating of that movie and the Mean Rating of the whole dataset (mu_hat).

b_i (or bi) will be the Bias per Movie.

b_i is then added to mu_hat for every Movie and that is the Predictor for that Movie.

Y = mu_hat + b_i

Adding the Movie bias or b_i should improve the prediction and generate a smaller RMSE

b_i is calculated for every movie and stored into the movie_avgs dataset.

```
##
## movie_avgs Dataset: Head
```

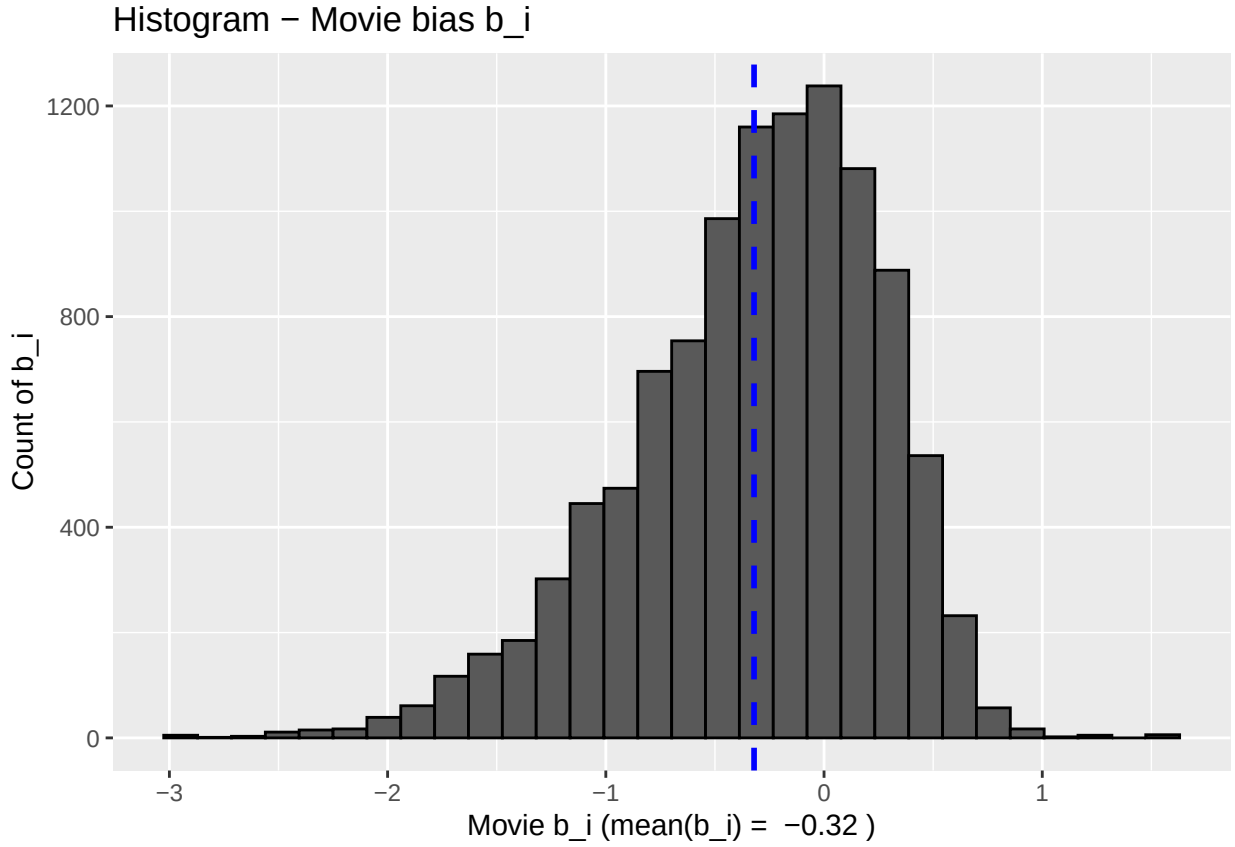
movieId	b_i
1	0.4151725
2	-0.3070658
3	-0.3654817
4	-0.6481659
5	-0.4437933

movieId	b_i
6	0.3028191

```
##
## movie_avgs Dataset: Number of Records
```

dim(movie_avgs)
10677

In the following plot, the distribution of Movie bias shows that there is variation around the value of 0 and most bias numbers go from -3 to 1



When adding Movie bias to the previous model, the prediction is Y = mu_hat + Movie_bi where Movie_bi = Movie bias

RMSE can be calculated using the test set (final_holdout_test) and mu_hat + Movie_bi as predictor.

```
## movie_bi_rmse:  0.9439087
```

The second model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	

When adding Movie bias, the RMSE is lower than the first model's RMSE

2.3.4 Third model: Y = mu_hat + Movie_bias + User_bias

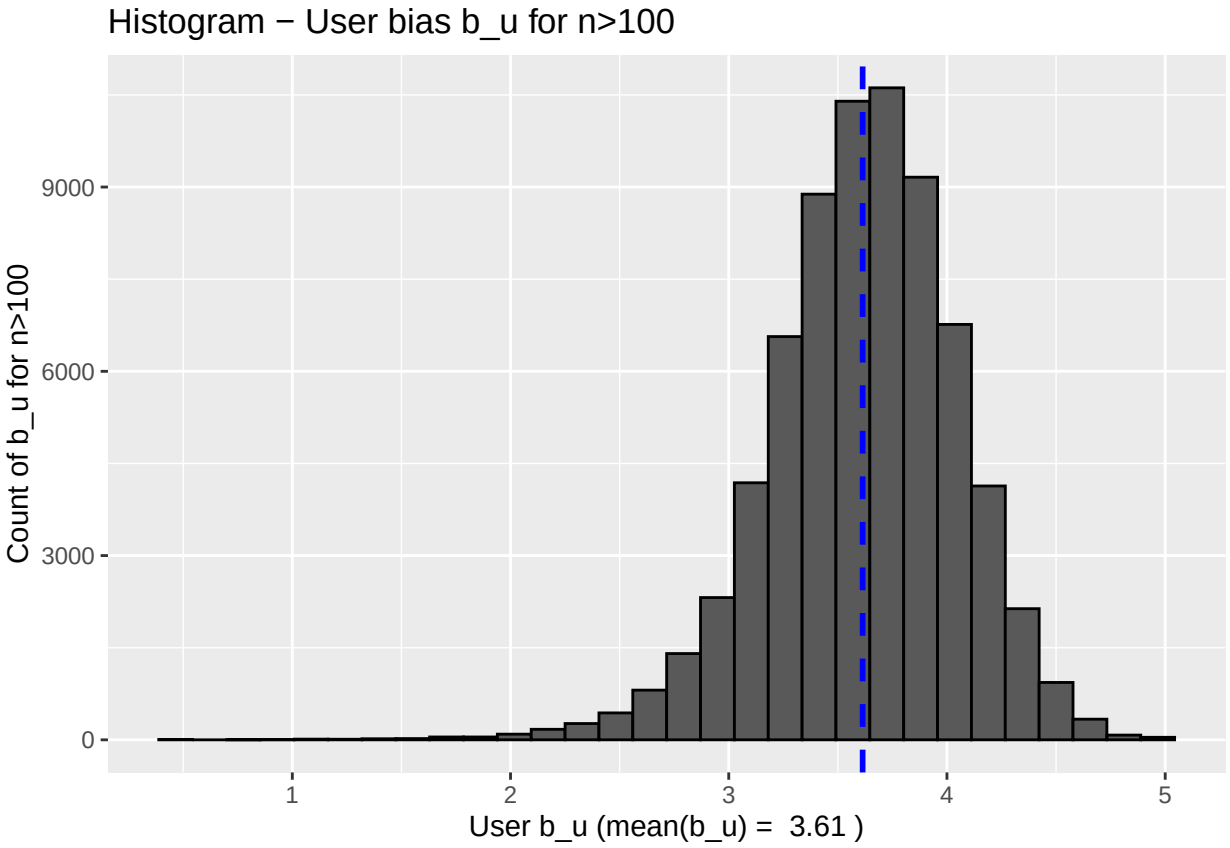
In the third model, User bias (effect) will be added to the previous model, Movie bias.

This updated model will have both Movie and User biases.

bu (or b_u) will be the Bias per User. bu is added to mu_hat and movie_bi for every Movie and User and that is the Predictor.

Adding the User bias to Movie bias should improve the prediction and generate a smaller RMSE

In the following plot, only Users that have rated 100 or more movies are included.



The distribution of Ratings shows that there is variation around the value of 4 and most ratings go from 1 to 5 This variation means that some users rate higher than other users do.

This variability also suggests that incorporating User bias would indeed improve the model

The third model will be Y = U + bi + bu

Here the prediction is Y = U + Movie_bi + User_bu

User bias will be calculated by subtracting mu_hat and movie_bias from Y (Rating)

User_bias = bu = Y - u - bi

User_bias will be averaged per user and stored in a new dataset: user_avgs

```
##
## user_avgs Dataset: Head
```

userId	b_u
1	1.6792347
2	-0.2364086
3	0.2643303
4	0.6520781
5	0.0852677
6	0.3462454

```
##
## user_avgs Dataset: Number of Records
```

dim(user_avgs)
69878

The prediction is Y = mu_hat + Movie_bi + User_bu where Movie_bi = Movie bias; User_bu = User bias.

RMSE can be calculated using the test set (final_holdout_test) and mu_hat + Movie_bi + User_bu as predictor.

```
## movie_bi_user_bu_rmse:  0.8653488
```

The third model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	
3: Predictor = mu_hat + Movie_bi + User_bu	0.8653488	

When adding User bias to Movie bias, the RMSE is lower than the previous model's RMSE

2.3.5 Fourth model: Y = mu_hat + Movie_Regularized_bias (Regularization)

The fourth model will apply Regularization.

RMSE was reduced when movie bias and user bias were incorporated but does not take into account when there are very few ratings (small sample size) for a movie or user.

The following table shows the 15 largest mistakes predicted by the second model (Y = mu_hat + Movie_bias). Residuals are calcualted subtracting (mu_hat + bi) from Y from the test set:

Ratings with highest absolute residuals	Residual
Pokémon Heroes (2003)	3.970803
Shawshank Redemption, The (1994)	-3.955131
Shawshank Redemption, The (1994)	-3.955131
Shawshank Redemption, The (1994)	-3.955131
Godfather, The (1972)	-3.915366
Godfather, The (1972)	-3.915366
Godfather, The (1972)	-3.915366
Usual Suspects, The (1995)	-3.865854
Usual Suspects, The (1995)	-3.865854
Usual Suspects, The (1995)	-3.865854
Schindler's List (1993)	-3.863493

Ratings with highest absolute residuals	Residual
Schindler's List (1993)	-3.863493
Schindler's List (1993)	-3.863493
Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie) (2002)	3.821782
Casablanca (1942)	-3.820424

```
##
## Following is the list of the 15 best movies based just on Movie_bias.
```

Movie title (Best)
Hellhounds on My Trail (1999)
Satan's Tango (Sátántangó) (1994)
Shadows of Forgotten Ancestors (1964)
Fighting Elegy (Kenka erejii) (1966)
Sun Alley (Sonnenallee) (1999)
Blue Light, The (Das Blaue Licht) (1932)
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)
Human Condition II, The (Ningen no joken II) (1959)
Human Condition III, The (Ningen no joken III) (1961)
Constantine's Sword (2007)
More (1998)
I'm Starting From Three (Ricomincio da Tre) (1981)
Class, The (Entre les Murs) (2008)
Mickey (2003)
Demon Lover Diary (1980)

```
##
## Most of the movies above are not well known.

##
## Following is the number of ratings of the 15 best movies based just on Movie_bias.

##  [1] 1 2 1 1 1 1 4 4 4 2 7 3 3 1 1

##
## These unknown movies have just one or very few ratings (very small sample size),

## which does not provide a good predictor. Regularization corrects that.

##
## Following is the list of the 15 worst movies based just on Movie_bias.
```

Movie title (Worst)
Besotted (2001)
Hi-Line, The (1999)
Accused (Anklaget) (2005)
Confessions of a Superhero (2007)
War of the Worlds 2: The Next Wave (2008)
SuperBabies: Baby Geniuses 2 (2004)
Hip Hop Witch, Da (2000)
Disaster Movie (2008)
From Justin to Kelly (2003)
Criminals (1996)
Mountain Eagle, The (1926)

Movie title (Worst)
Stacy's Knights (1982)
Dog Run (1996)
Monkey's Tale, A (Les Château des singes) (1999)
When Time Ran Out... (a.k.a. The Day the World Ended) (1980)

```
##
## Most of the movies above are not well known.

##
## Following is the number of ratings of the 15 worst movies based just on Movie_bias.

##  [1]   2   1   1   1   2  56  14  32 199   2   2   1   1   1   1

##
## These unknown movies have just one or very few ratings (very small sample size),

## which does not provide a good predictor. Regularization corrects that.
```

Good sample size of Ratings per Movie is desired to have a good predictor. Very few ratings tells us that they have very small sample sizes.

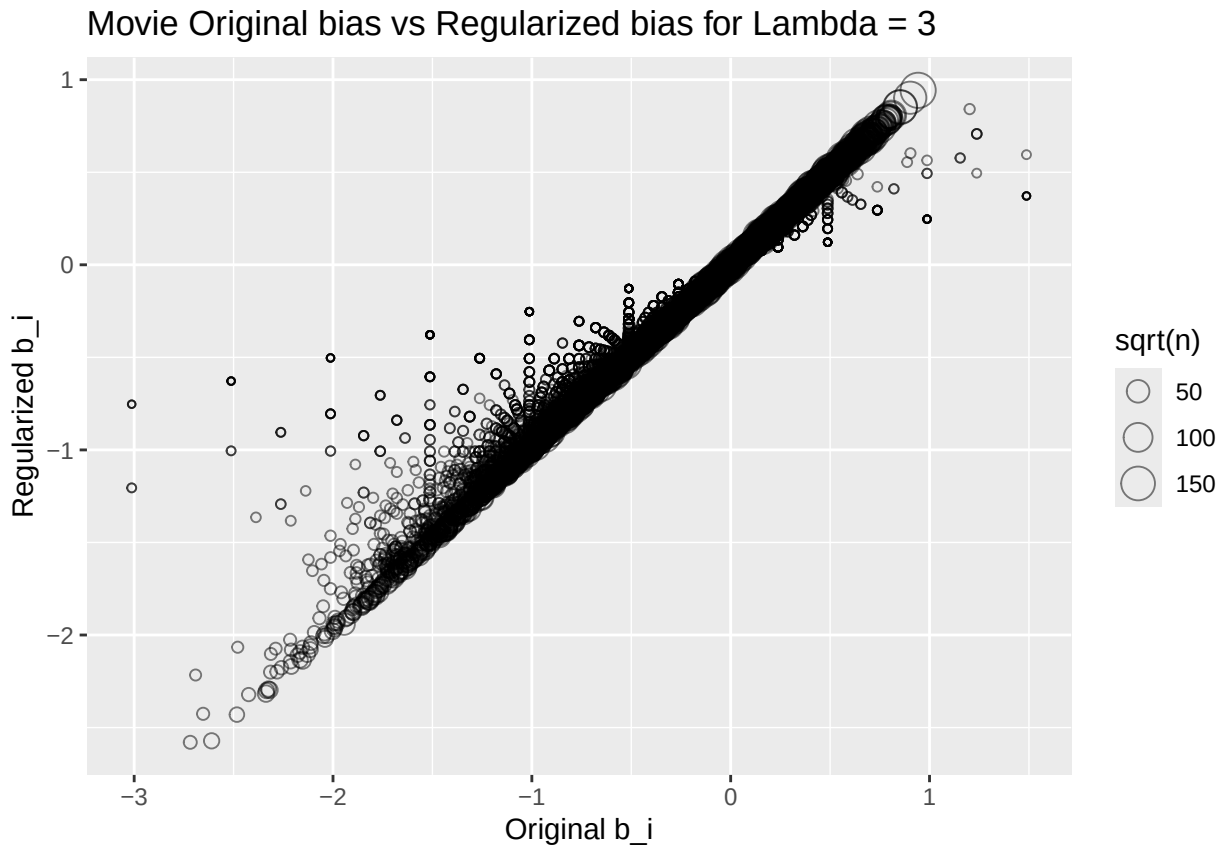
Having less ratings (smaller sample sizes) increases bias and generates higher residuals. Higher residuals increase the RMSE.

Regularization will be used to reduce the impact of large residuals that are coming from movies with very few ratings (or small sample sizes).

Using Regularization, the small sample size bias is penalized by adding a Lambda to "n", where n is the number of ratings for movie i.

- The higher the "n", the lower the impact of Lambda, thus, less penalized.
- The lower the "n", the higher the impact of Lambda, thus, more penalized.

Lambda = 3 will be used to generate Movie_Regularized_bias (Regularized_b_ias_i), and compare it to Movie_bias (original_b_i) in a plot.



The resulting plot shows how Regularized_bias and Original_bias differ more for small values of n.

The lists of the top 15 best and worst movies are generated again but using Movie_regularized_bias instead of Movie_bias.

```
##
## Following is the list of the 15 best movies based on Movie_regularized_bias.
```

Movie title (Best)
Shawshank Redemption, The (1994)
Godfather, The (1972)
Usual Suspects, The (1995)
Schindler's List (1993)
More (1998)
Casablanca (1942)
Rear Window (1954)
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)
Third Man, The (1949)
Double Indemnity (1944)
Paths of Glory (1957)
Seven Samurai (Shichinin no samurai) (1954)
Godfather: Part II, The (1974)
Dark Knight, The (2008)
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)

```
##
## Following is the number of ratings of the 15 best movies based just on Movie_bias.
```

```
##  [1] 28015 17747 21648 23193     7 11232  7935  2922  2967  2154  1571  5190
## [13] 11920  2353 10627
```

```
##
## These movies have mostly thousands of ratings, which provides better predictors.
```

```
##
## Following is the list of the 15 worst movies based on Movie_regularized_bias.
```

Movie title (Worst)
SuperBabies: Baby Geniuses 2 (2004)
From Justin to Kelly (2003)
Pokémon Heroes (2003)
Disaster Movie (2008)
Carnosaur 3: Primal Species (1996)
Glitter (2001)
Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie) (2002)
Gigli (2003)
Barney's Great Adventure (1998)
Hip Hop Witch, Da (2000)
Faces of Death: Fact or Fiction? (1999)
Yu-Gi-Oh! (2004)
Faces of Death 6 (1996)
Son of the Mask (2005)
Carnosaur 2 (1995)

```
##
## Following is the number of ratings of the 15 worst movies based just on Movie_bias.
```

```
##  [1]  56 199 137  32  68 339 202 313 208  14  58  80  79 165  92
```

```
##
## These movies have mostly hundreds of ratings, which provides better predictors.
```

The prediction is Y = mu_hat + Movie_Regularized_bi where Movie_Regularized_bi = Movie regularized bias with Lambda = 3.

RMSE can be calculated using the test set (final_holdout_test) and mu_hat + Movie_Regularized_bi as predictor.

```
## model_regularized_bi_rmse:  0.9438538
```

The fourth model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	
3: Predictor = mu_hat + Movie_bi + User_bu	0.8653488	
4: Predictor = mu_hat + Movie_regularized_bi, arbitrary Lambda = 3	0.9438538	3

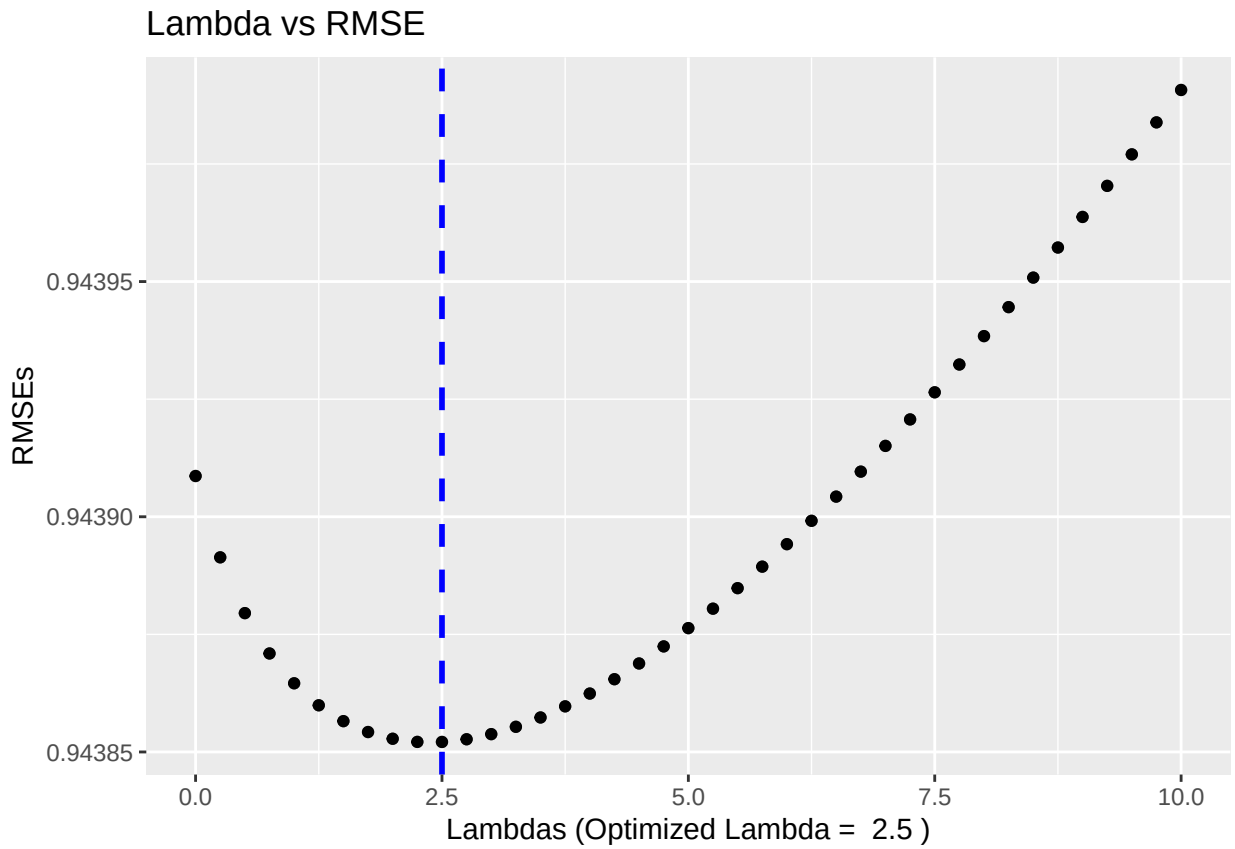
When using Movie Regularized bias instead of Movie bias, the RMSE is lower than the second model's RMSE

2.3.6 Fifth model: Y = mu_hat + Movie_Regularized_bias (Regularization) with Optimized Lamda

The fifth model will optimize the value of Lambda for the fourth model.

Values of Lambda will be used in an iteration to find the Lambda that minimizes RMSE. Lambda will be tested between 0 and 10, with an increment of 0.25. This means the following values will be tested: 0, 0.25, 0.5, 0.75, 1, 1.25, and so on up to 10.

```
## The following plot shows Lambda versus RMSE.
```



The prediction is Y = mu_hat + Movie_Regularized_bi with Optimized Lambda where Movie_Regularized_bi = Movie regularized bias with Lambda optimized between 0 and 10 at 0.25 as interval

RMSE can be calculated using the test set (final_holdout_test) and mu_hat + Movie_Regularized_bi as predictor.

```
## Lambda that minimizes RMSE:  2.5
```

```
## Minimum RMSE:  0.9438521
```

The fifth model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	
3: Predictor = mu_hat + Movie_bi + User_bu	0.8653488	
4: Predictor = mu_hat + Movie_regularized_bi, arbitrary Lambda = 3	0.9438538	3
5: Predictor = mu_hat + Movie_regularized_bi, optimized Lambda = 2.5	0.9438521	2.5

When Lambda is optimized (Lambda = 2.5), RMSE is lower than the RMSE when Lambda is assigned a value of 3.

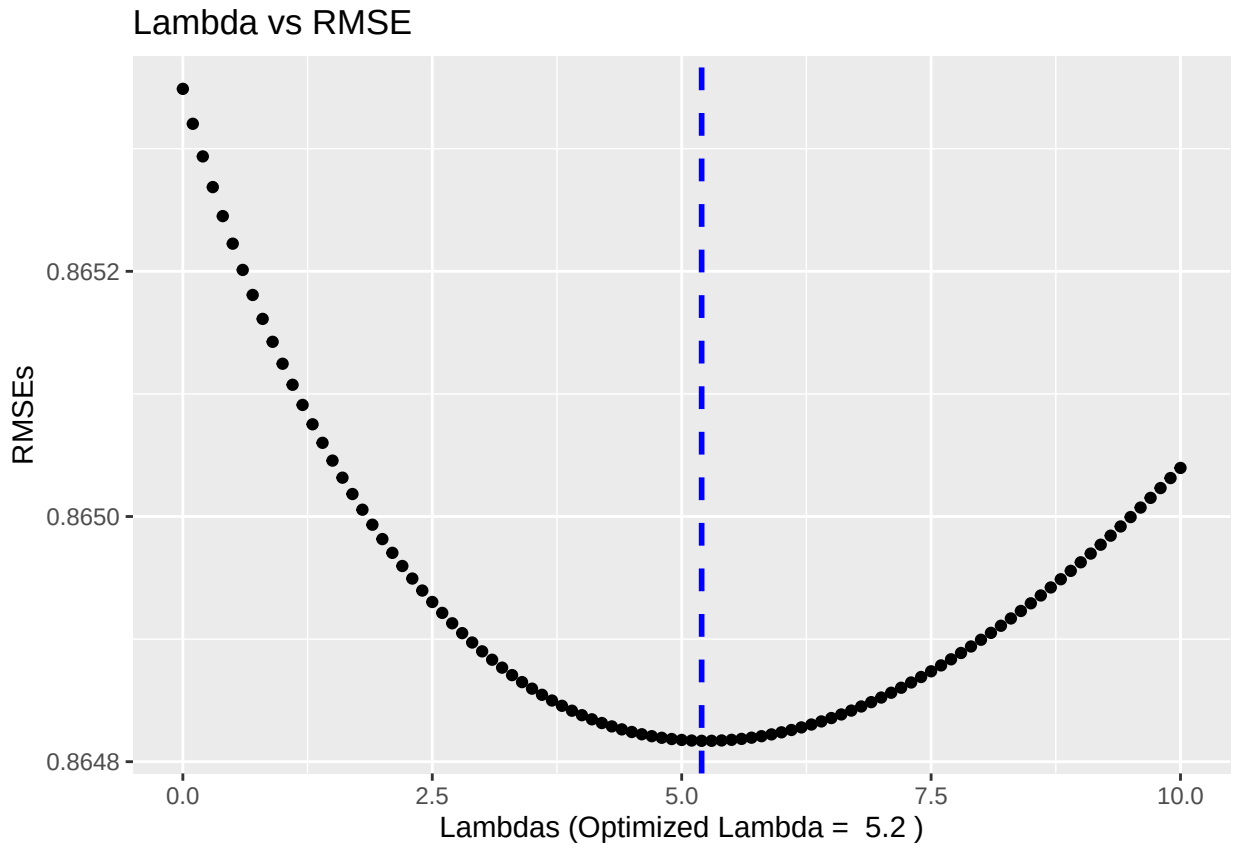
2.3.7 Sixth model: Y = mu_hat + Movie_Regularized_bias + User_Regularized_bias (Regularization) with Optimized Lamda

In the sixth model, User regularized bias will be added to Movie regularized bias (the previous model.) Lambda will also be optimized for a minimum RMSE.

Lambda will be tested between 0 and 10, with an increment of 0.1. This means the following values will be tested: 0, 0.1, 0.2, 0.3, and so on up to 10.

The following plot shows Lambda versus RMSE.

```
## The following plot shows Lambda versus RMSE.
```



The prediction is Y = mu_hat + Movie_Regularized_bi + User_Regularized_bu with Optimized Lambda where Movie_Regularized_bi = Movie regularized bias; User_Regularized_bu = User Regularized bias with Lambda optimized between 0 and 10 at 0.1 as interval

RMSE can be calculated using the test set (final_holdout_test) and mu_hat + Movie_Regularized_bi+ User_Regularized_bu as predictor.

```
## Optimum Lambda:  5.2
```

```
## Minimum RMSE:  0.864817
```

The sixth model and its RMSE are as follows:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	
3: Predictor = mu_hat + Movie_bi + User_bu	0.8653488	

method	RMSE	Lambda
4: Predictor = mu_hat + Movie_regularized_bi, arbitrary Lambda = 3	0.9438538	3
5: Predictor = mu_hat + Movie_regularized_bi, optimized Lambda = 2.5	0.9438521	2.5
6: Predictor = mu_hat + Movie_regularized_bi + User_regularized_bu, optimized Lambda = 5.2	0.8648170	5.2

3. RESULTS

The following table shows all six models and their RMSEs:

method	RMSE	Lambda
1: Predictor = mu_hat	1.0612018	
2: Predictor = mu_hat + Movie_bi	0.9439087	
3: Predictor = mu_hat + Movie_bi + User_bu	0.8653488	
4: Predictor = mu_hat + Movie_regularized_bi, arbitrary Lambda = 3	0.9438538	3
5: Predictor = mu_hat + Movie_regularized_bi, optimized Lambda = 2.5	0.9438521	2.5
6: Predictor = mu_hat + Movie_regularized_bi + User_regularized_bu, optimized Lambda = 5.2	0.8648170	5.2

RMSE has been decreasing as every model incorporated additional features.

The last model incorporated regularization with both Movie bias and User bias and Lambda was optimized resulting in the lowest RMSE.

Predictor = mu_hat + Movie_regularized_bi + User_regularized_bu

```
## Minimum RSME ( 0.864817 ) was found with Regularization model including
```

```
## Movie and User biases for optimized lambda =  5.2
```

The sixth model has an RMSE that is lower than: 0.86490 and is the model with the lowest RMSE. This model will provide the best movie suggestions to Netflix users.

4. CONCLUSION

In this analysis, six models were considered. Each model was based on a subset of 10 million rows of the MovieLens dataset.

The models were reducing RMSE to get to the minimum RMSE found when Regularization was applied to both Movie bias and User bias together with optimizing Lambda.

Predictor = mu_hat + Movie_regularized_bi + User_regularized_bu

Using regularization, RMSE was minimum when Lambda was optimized through iteration. That was the Sixth model.

With this model we built a recommendation system that will predict which movies users would like to watch.

One limitation of the analysis is that it only considered 10 million records, for future studies, it is recommended to:

- apply a larger dataset
- apply updated versions of the dataset as they are coming

One element that was not included in this analysis is “genre.”

“genre” can be incorporated to create a seventh model and calculate the RMSE:

Seventh model: Y = movie_bi + User_bu + Genre_bg