# Prediction of Movie Rating

Sadia Boksh

29/01/2021

## Introduction

For this project, I will be creating a movie recommendation system using the MovieLens dataset.Here I will train a machine learning model using the inputs in one subset to predict movie ratings in the validation set.

Here I will fit a few linear models and calculate their RMSEs to check how good they fit. My first model would be a simple model that uses the movie average to predict. Eventually, I will build my algorithm to include movie effects, user effects and genre effects. Finally, I will use regularization to add penalty term to shrink the effect of smaller sample sizes towards 0.

## Analysis

First I split the Movielens data set to 10% validation and 90% train set.

Number of rows and col in the train set:

```
## [1] 9000055
```

```
## [1] 6
```

Number of zeros as ratings:

```
##    n
## 1: 0
```

Number of 3's as rating:

```
##         n
## 1: 2121240
```

Number of different movies:

```
## [1] 10677
```

Number of different user:

```
## [1] 69878
```

Number movie ratings in Drama, Comedy, Thriller, ROmance in edx dataset:

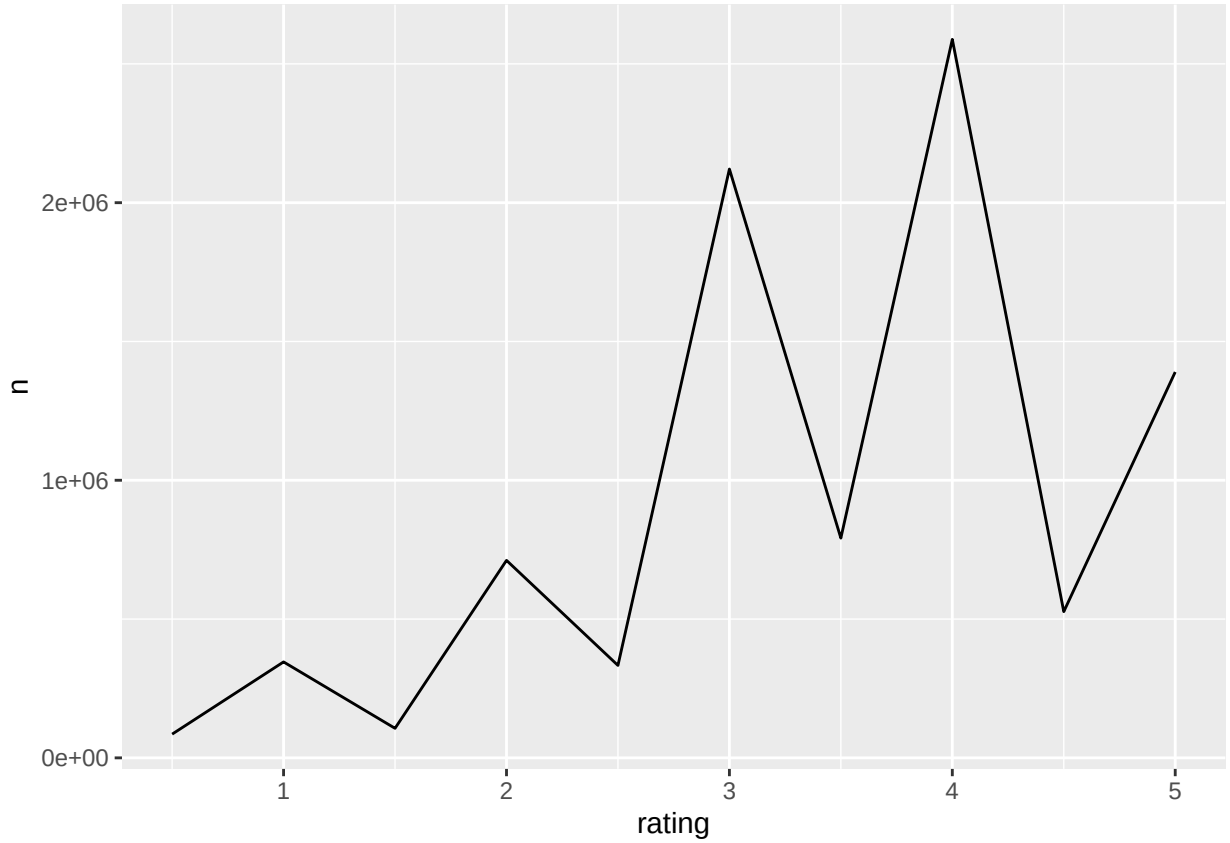
```
##      Genre       n
## 1    Drama 3910127
## 2   Comedy 3540930
## 3 Thriller 2325899
## 4  Romance 1712100
```

Highest rated movies:

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10,677 x 3
##    movieId     n title
##      <dbl> <int> <chr>
##  1     296 31362 Pulp Fiction (1994)
##  2     356 31079 Forrest Gump (1994)
##  3     593 30382 Silence of the Lambs, The (1991)
##  4     480 29360 Jurassic Park (1993)
##  5     318 28015 Shawshank Redemption, The (1994)
##  6     110 26212 Braveheart (1995)
##  7     457 25998 Fugitive, The (1993)
##  8     589 25984 Terminator 2: Judgment Day (1991)
##  9     260 25672 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
## 10     150 24284 Apollo 13 (1995)
## # ... with 10,667 more rows
```

Greatest number of ratings:




From the plot above we can say that the half star ratings are less common than full star ratings.

## Model Fitting

### First Model

For the first model, we will use just the average of all movies for prediction. RMSE for this simple model:

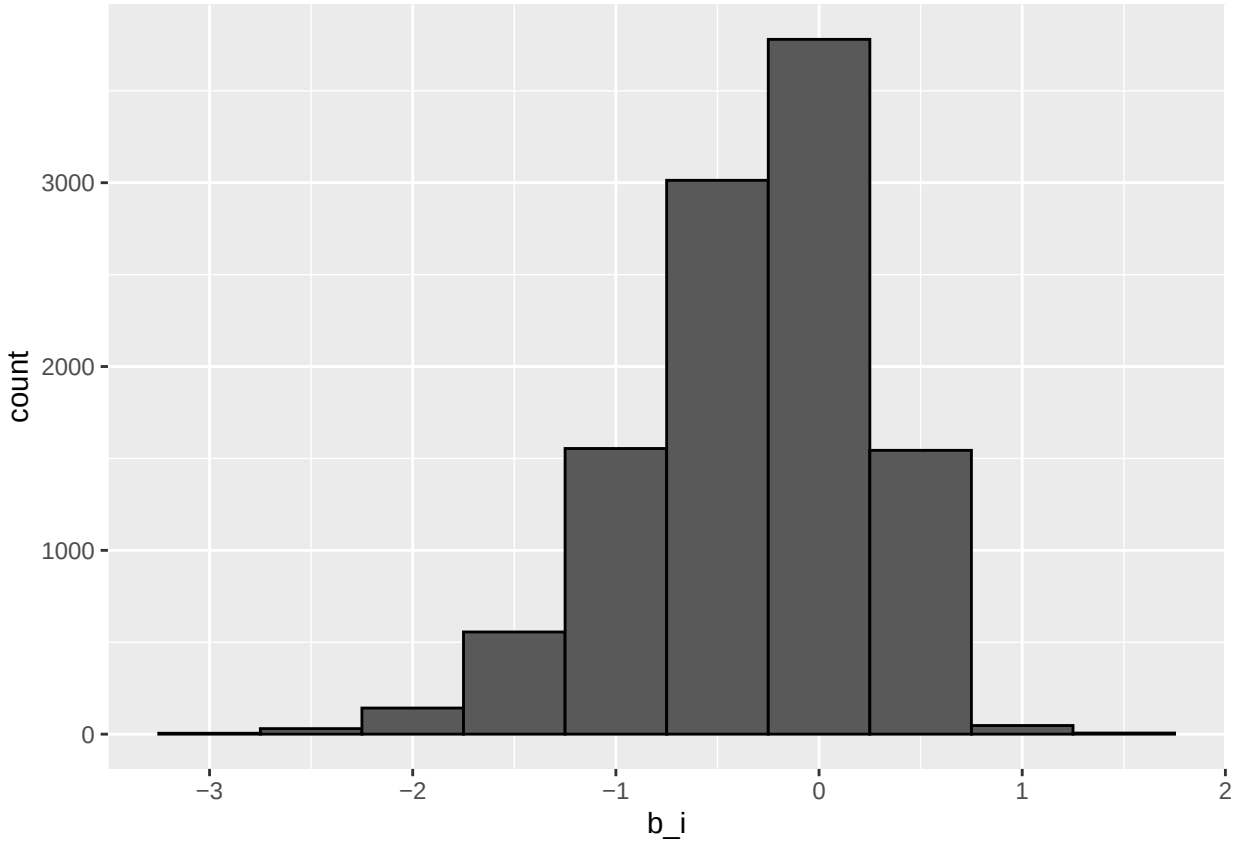
```
## [1] 1.061202
```

### Movie Effect

Next I will cater for the movie effect and RMSE for this model:

```
## [1] 0.9439087
```

We can also visualize the movie effect:

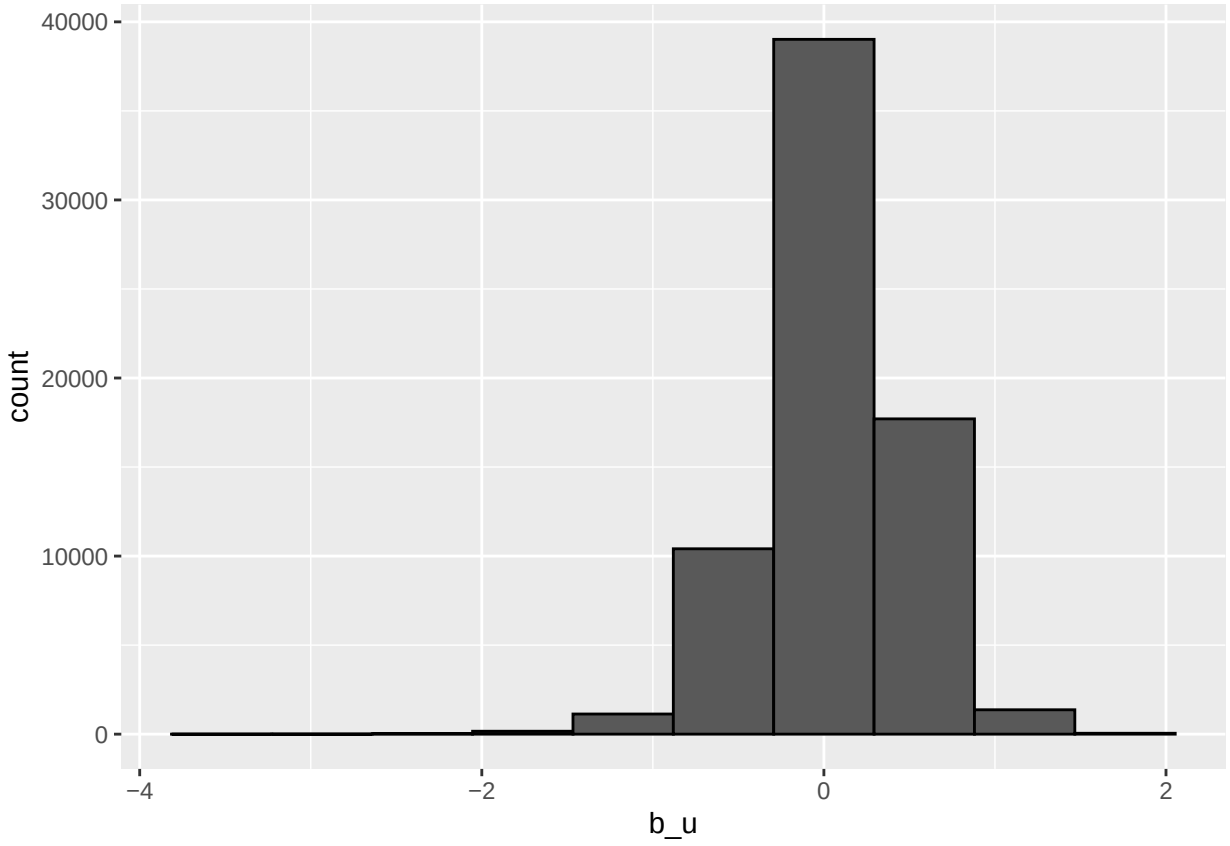



**User Effect**

I will also add the user effect to my algorithm and RMSE for this model:

```
## [1] 0.8653488
```

We can also check visually how user to user variability looks like:

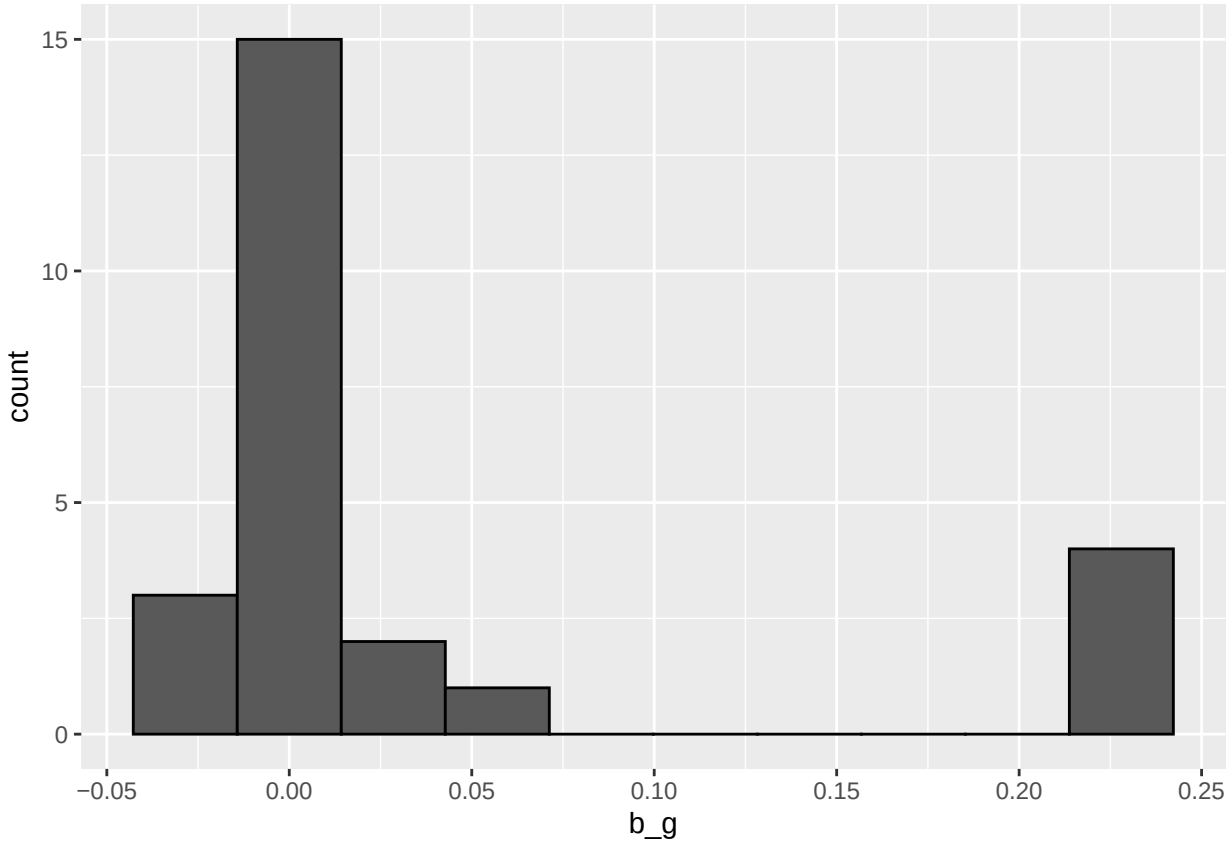



**Genre Effect**

Next I would like to see how genres take part in movie rating. For genre effect my model is:
$Y_{u,i} = \mu + b_i + b_u + \sum_{k=1}^{K} X_{u,i} b_k$ with $x_{u,i}^k = 1$ if $g_{u,i} is genre k.
RMSE for this model:

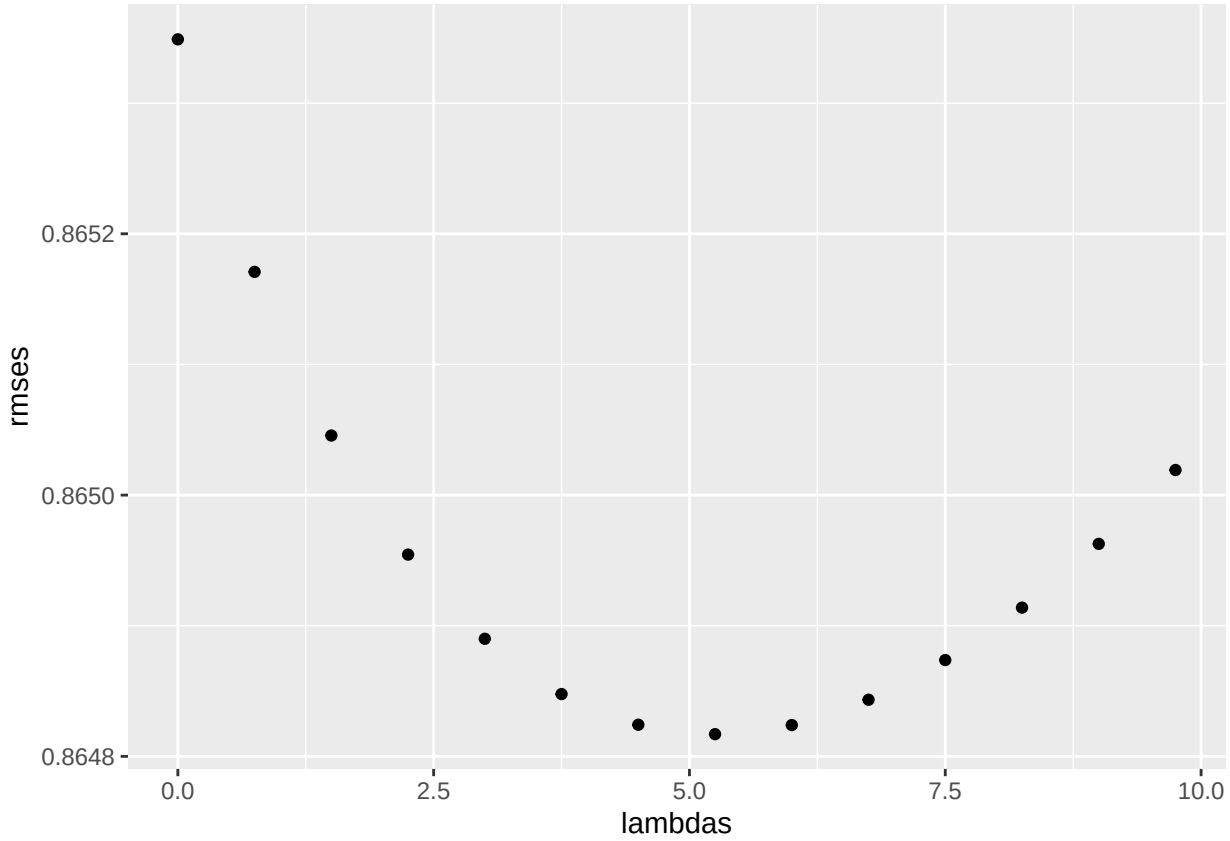
```
## [1] 0.8657769
```

and genre effect visually:




**Regularization**

Finally I will use regularization that will penalize the user and movie bias and shrink them toward 0 when sample sizes are small. I will use tuning parameter lambda from 0 to 10 and lambda = 5 gives me the best RMSE. RMSE at lambda = 5:




```
## [1] 0.864817
```

## Result

From the analysis above we can see that the RMSE keeps getting better as we include user effect, movie effect and genre effect in the algorithm and regularization makes it perform even better.

```
##                              Method      RMSE
## 1                         Just Avg 1.0612018
## 2                     Movie Effect 0.9439087
## 3                      User Effect 0.8653488
## 4                     Genre Effect 0.8657769
## 5 Regularized Movie + User Effect 0.8648170
```

## Conclusion

In summary, regularization has improved the prediction algorithm and produced the best RMSE. But this model does not cater for user movie preferences or movie rating pattern. Future work can be done in this area by doing factor analysis using Matrix factorization.