Movie recommendation system

Amit V Singh

6/15/2019

Introduction

One of the major families of applications of machine learning in the information technology sector is the ability to make recommendations of items to potential users or customers. In year 2006, Netflix has offered a challenge to data science community. The challenge was to improve Netflix's in house software by 10% and win $1M prize.

This capstone project is based on the winner's team algorithm and is a part of the course HarvardX: PH125.9x Data Science: Capstone project. The Netflix data is not freely available so an open source dataset from movieLens '10M version of the MovieLens dataset' has been used. The aim of this project is to develop a machine learning algorithm using the inputs in one subset to predict movie ratings in the validation set. Several machine learning algorithm has been used and results have been compared to get maximum possible accuracy in prediction.

This report contains problem definition, data ingestion, exploratory analysis, modeling and data analysis, results and concluding remarks and have been written in that order.

Problem Defnition

This capstone project on 'Movie recommendation system' predicts the movie rating by a user based on users past rating of movies. The dataset used for this purpose can be found in the following links

- MovieLens 10M dataset
- MovieLens 10M dataset - zip file

The challenge is not so easy given that there are many different type of biases present in the movie reviews. It can be different social, psychological, demographic variations that changes the taste of every single users for a given particular movie. However the problem can still be designed to tackle major biases which can be expressed via mathematical equations relatively easily. The idea here is to develop a model which can effectively predict movie recommendations for a given user without our judgement being impaired due to different biases. In the algorithm, the prevalences can be suppressed using some clever mathematical tricks. This will become clear as we follow this document.

Data Ingestion

The code is provided in the edx capstone project module Create Test and Validation Sets

It basically gives a partition of the dataset for training and testing our model. At the end, the unnecessary downloaded files are removed from the working directory, which is always a good coding practice ("always clean after you cook").

Once a clean dataset is available, one must inquire the dataset features and calculate the basic summary statistics

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
```

```
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     231      5 838983392          Dumb & Dumber (1994)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
##                         genres
## 1                Comedy|Romance
## 2          Action|Crime|Thriller
## 3                        Comedy
## 4  Action|Drama|Sci-Fi|Thriller
## 5       Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
```

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18122   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35743   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35869   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53602   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000061     Length:9000061
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Data pre-processing and exploratory analysis

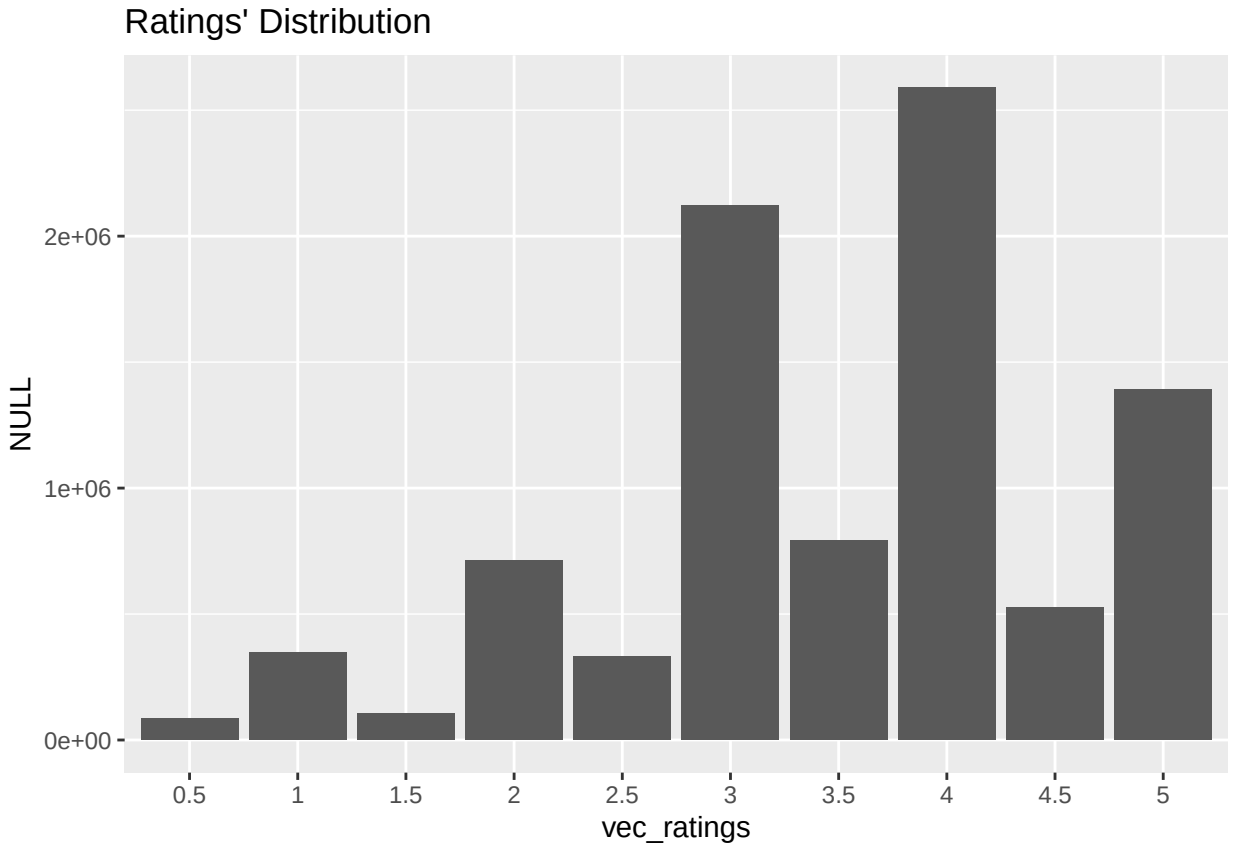
In data science, it is important that the dataset should be curated in a proper format so that it can be made ready for exploration and analysis. For example extracting timestamp feature set to date format that can be further used to perform the time dependent analysis.

Data Exploration and general statistics

Total movie ratings per genre

Ratings distribution

```
##  [1] 5.0 3.0 2.0 4.5 3.5 4.0 1.0 1.5 2.5 0.5
```

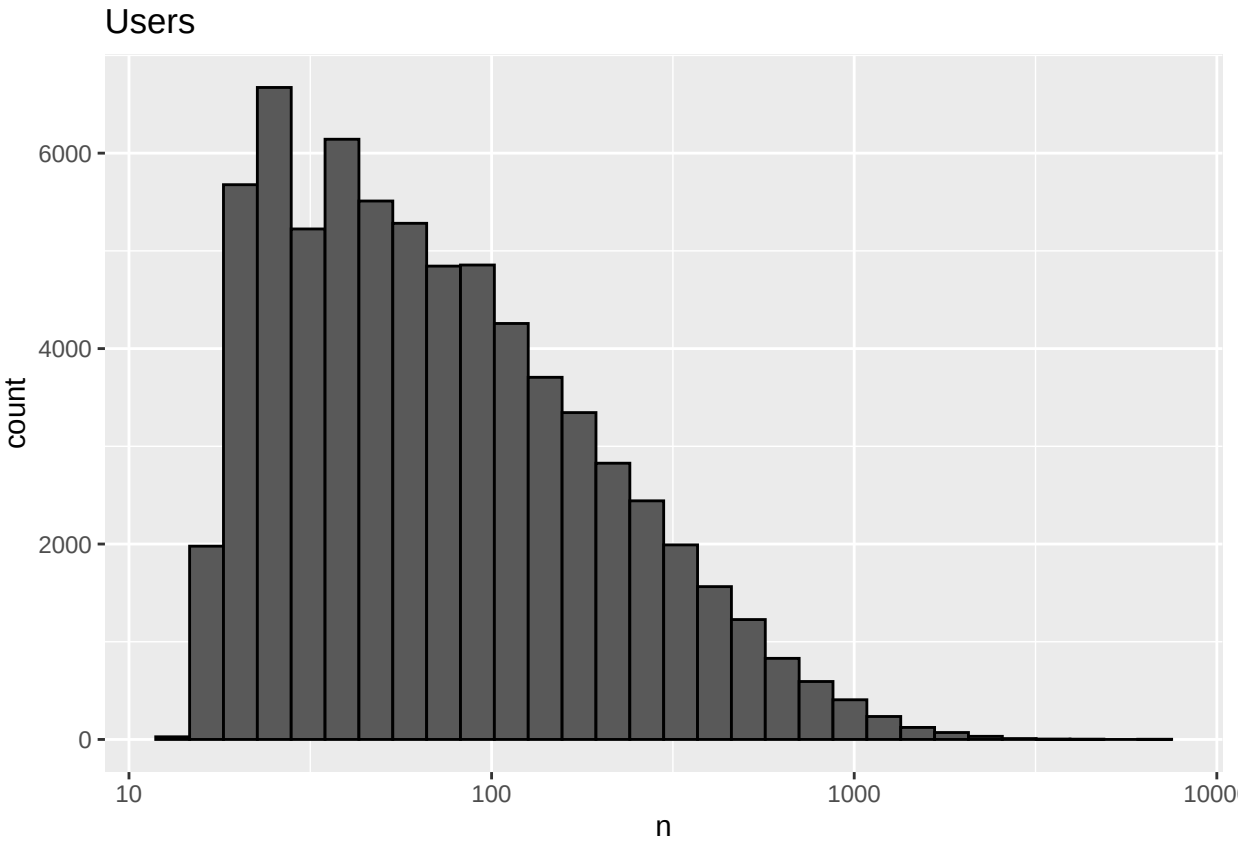


The above rating distribution shows that the users have a general tendency to rate movies between 3 and 4. This is a very general conclusion. We should further explore the effect of different features to make a good predictive model.

Data Analysis Strategies

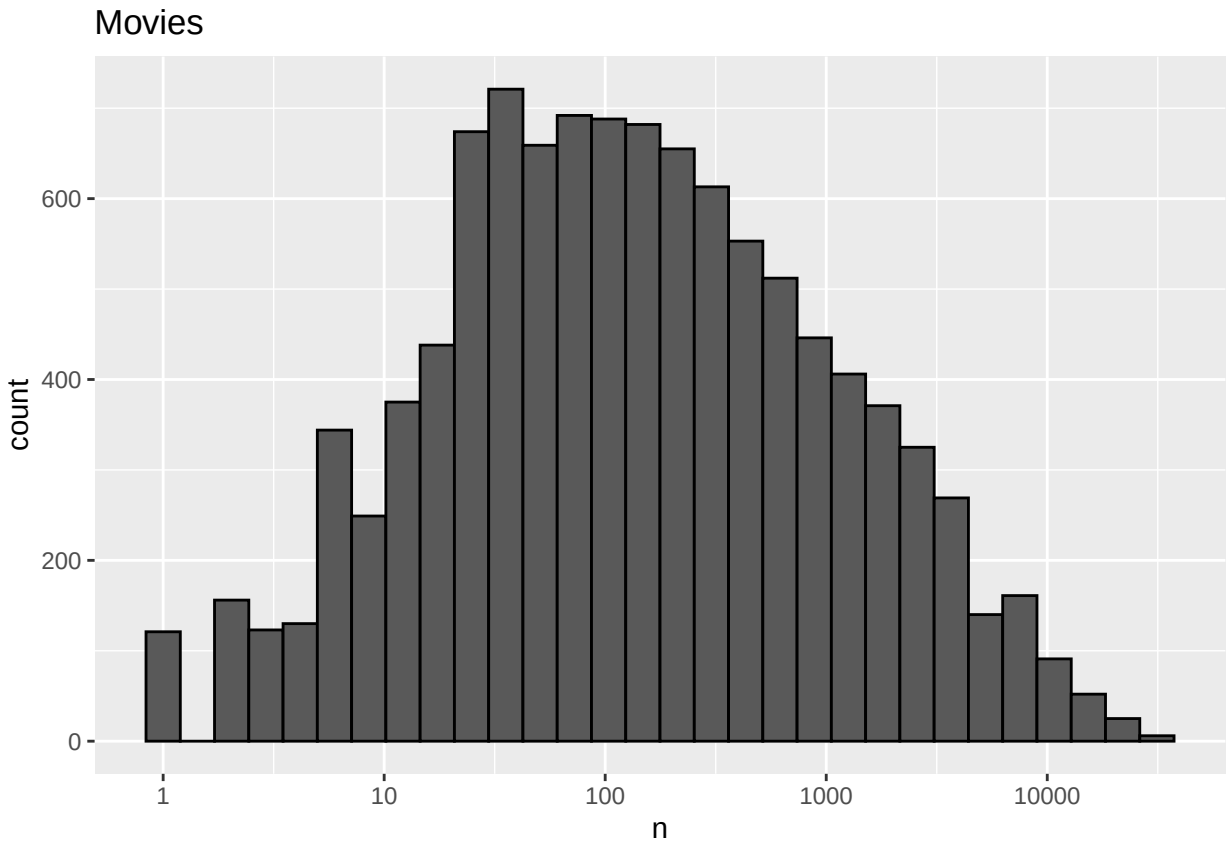
- Some movies are rated more often than others (e.g. blockbusters are rated higher). How to incorporate this in our model: find movie bias.
- Some users are positive and some have negative reviews because of their own personal liking/disliking regardless of movie. How to address this characteristics: find users bias.
- The popularity of the movie genre depends strongly on the contemporary issues. So we should also explore the time dependent analysis. How to approach this idea: find the genre popularity over the years
- Do the users mindset also evolve over time? This can also effect the average rating of movies over the years. How do visualize such effect: plot rating vs release year

The distribution of each user's ratings for movies. This shows the users bias



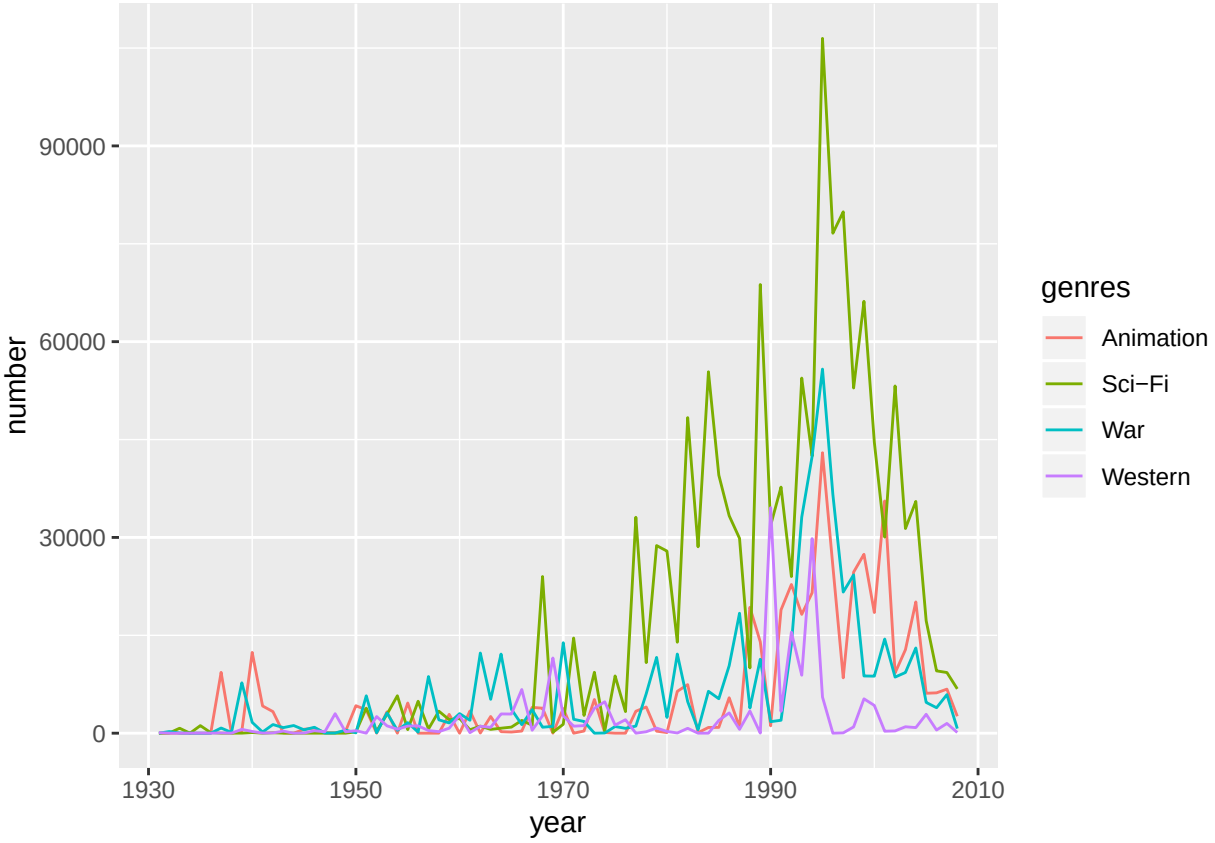
Above plot shows that not every user is equally active. Some users have rated very few movie and their opinion may bias the prediction results.

Some movies are rated more often than others.Below is their distribution. This explores movie biases.



The histogram shows some movies have been rated very few number of times. So they should be given lower importance in movie prediction.

Genres popularity per year. Here we tackle the issue of temporal evolution of users taste over different popular genre.

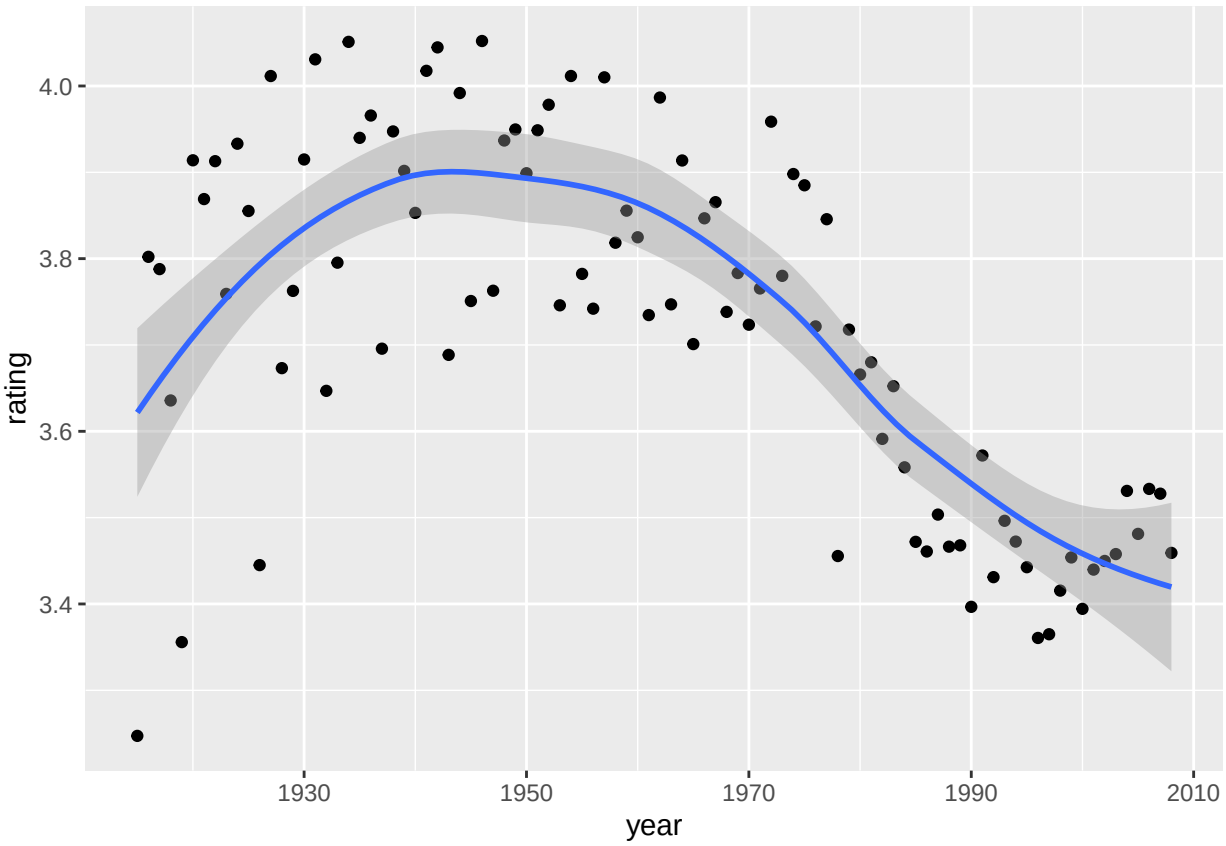


This plots depicts some genre become more popular over others for different period of time.

Rating vs release year

Here, a general trend of movie viewers and their rating habits can be explored.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The general trend shows modern users relatively rate movies lower.

Data Analysis: Model Preparation

Loss function

One of them main goal of machine learning is to minimize the root mean square error (RMSE) or so called loss function. The loss function is be used to measure accuracy of training and predictive models given by following formula

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

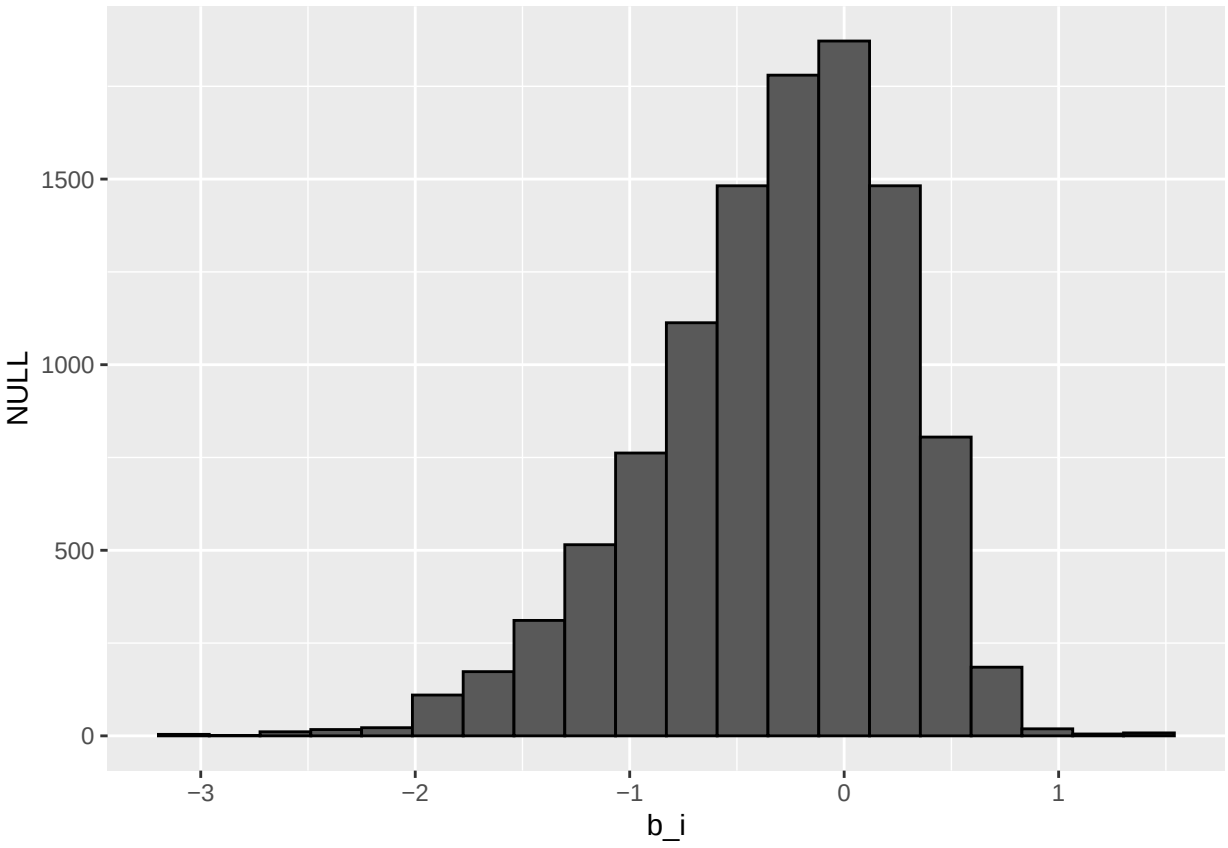
The quality of the model will be assessed by the RMSE (the lower the better).

```
## [1] 3.512464
```

Biases that overshadow our judgement

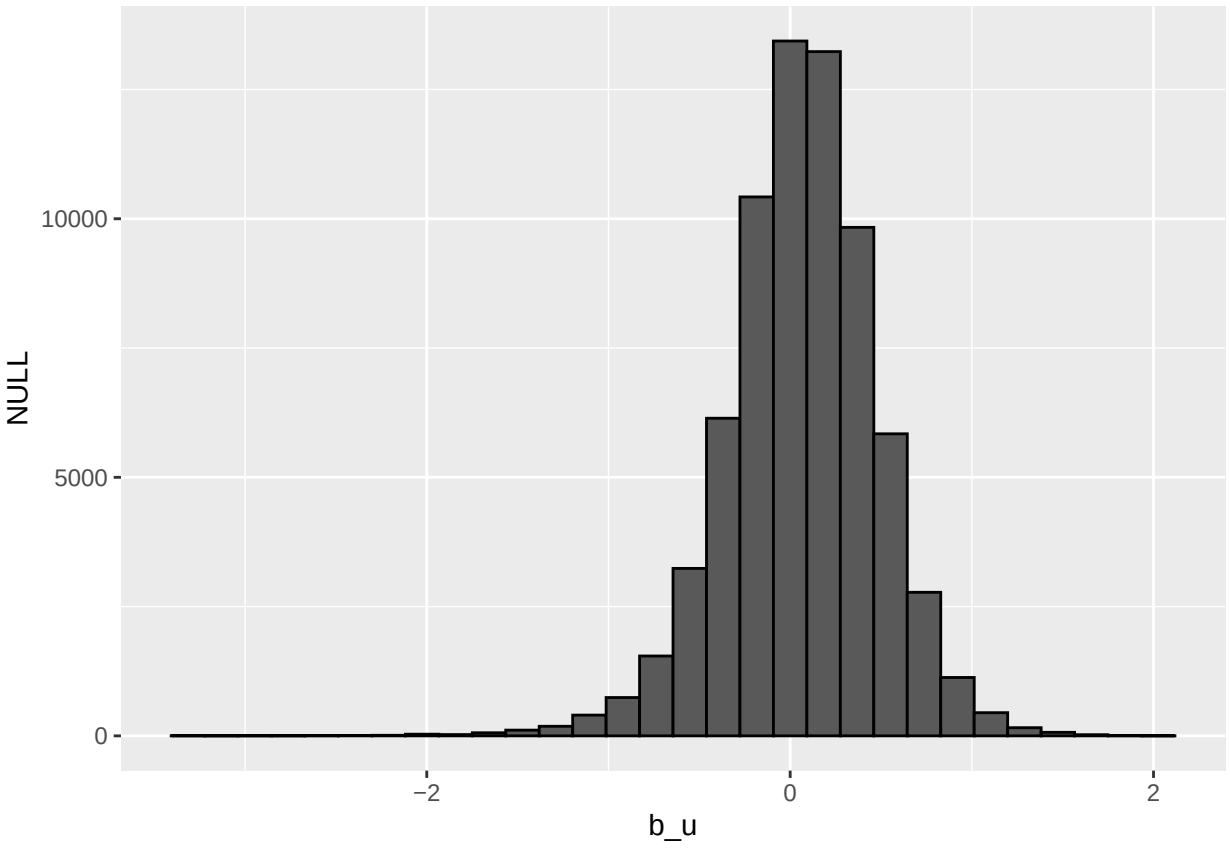
Penalty Term (b_i) - Movie Effect

Different movies are rated differently. As shown in the exploration, the histogram is not symmetric and is skewed towards negative rating effect. The movie effect can be taken into account by taking he difference from mean rating as shown in the following chunk of code.



Penalty Term (b_u) - User Effect

Different users are different in terms of how they rate movies. Some cranky users may rate a good movie lower or some very generous users just don't care for assessment. We have already seen this pattern in our data exploration plot (user bias). We can calculate it using this code.



Model Creation

Baseline Model

It???s simply a model which ignores all the feathers and simply calculates mean rating. This model acts as a baseline model and we will try to improve RMSE relative to this baseline standard model. It can be represented mathematically in the following way

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

where μ the "true" rating for all movies with $\varepsilon_{i,u}$ independent errors sampled from the same distribution centered at 0.

```
## [1] 1.060651
```

```
## # A tibble: 1 x 2
##   method           RMSE
##   <chr>           <dbl>
## 1 Using mean only  1.06
```

Movie Effect Model

An improvement in the RMSE is achieved by adding the movie effect and can be formulated as follows

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

where the additional term b_i is movie effect.

method	RMSE
Using mean only	1.0606506
Movie Effect Model	0.9437046

```
## # A tibble: 2 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Using mean only    1.06
## 2 Movie Effect Model 0.944
```

The error has drop by 5% and motivates us to move on this path further.

Movie and User Effect Model

Given that movie and users biases both obscure the prediction of movie rating, a further improvement in the RMSE is achieved by adding the user effect. These words can be manifested in mathematical language

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

where b_u stands for user effect.

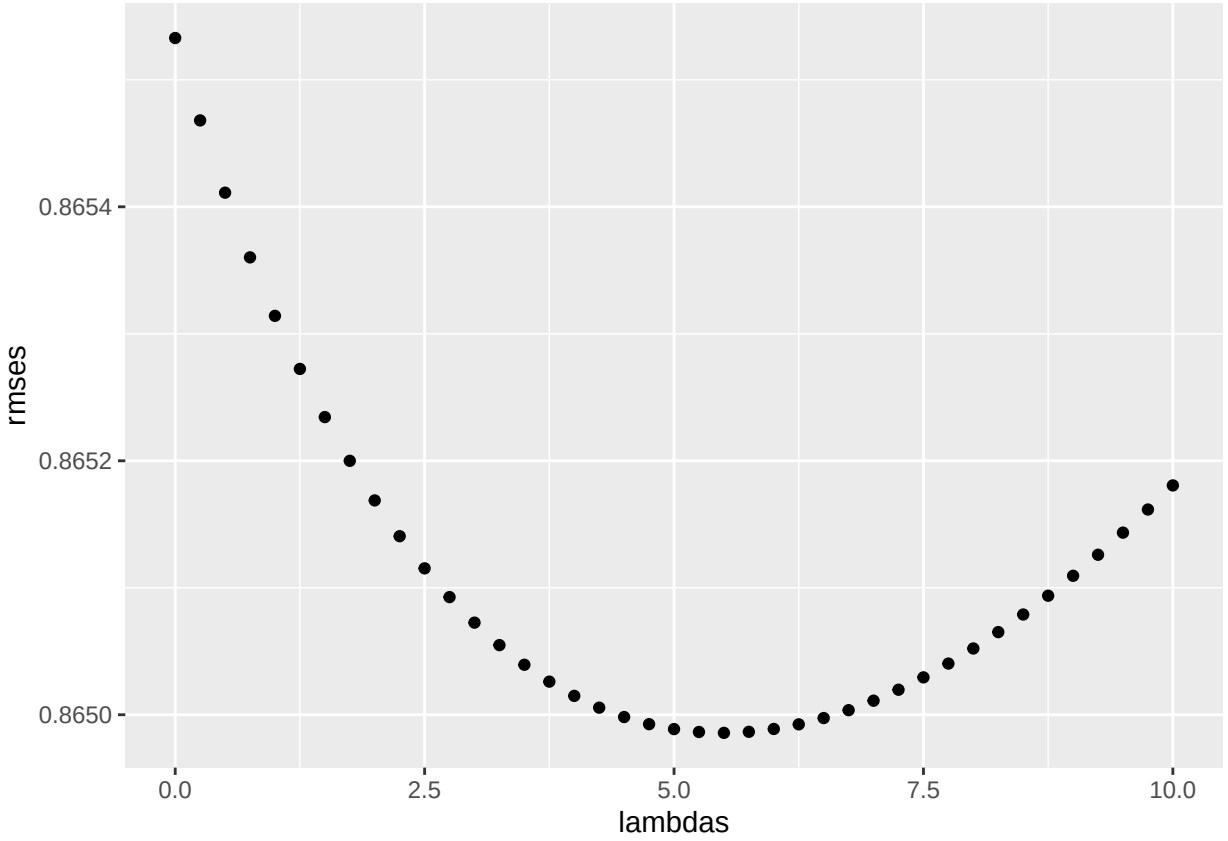
method	RMSE
Using mean only	1.0606506
Movie Effect Model	0.9437046
Movie and User Effect Model	0.8655329

```
## # A tibble: 3 x 2
##   method                       RMSE
##   <chr>                       <dbl>
## 1 Using mean only             1.06
## 2 Movie Effect Model          0.944
## 3 Movie and User Effect Model 0.866
```

This is a good improvement from our last model.

Regularization based approach (motivated by Netflix challenge)

We have noticed in our data exploration, some users are more actively participated in movie reviewing. There are also users who have rated very few movies (less than 30 movies). On the other hand, some movies are rated very few times (say 1 or 2). These are basically noisy estimates that we should not trust. Additionally, RMSE are sensitive to large errors. Large errors can increase our residual mean squared error. So we must put a penalty term to give less importance to such effect.



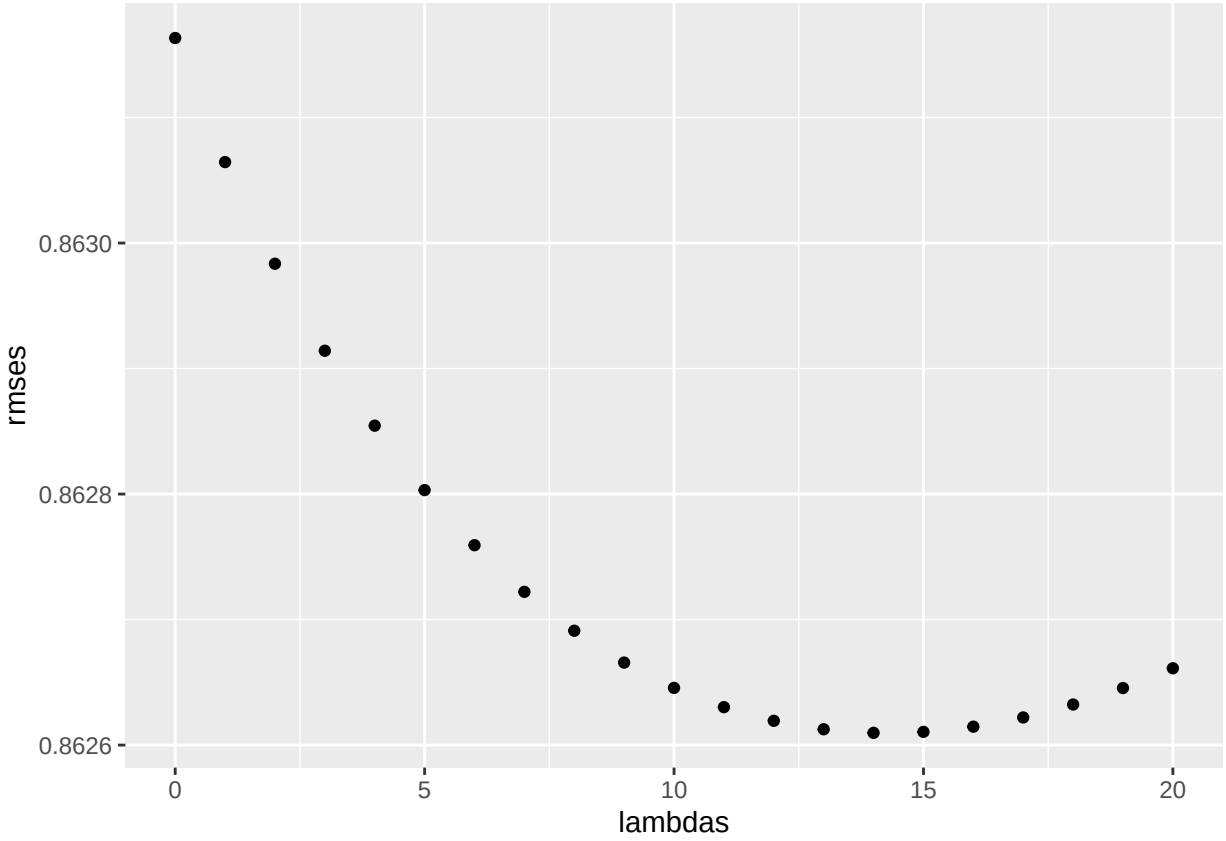

```
## [1] 5.5
```

method	RMSE
Using mean only	1.0606506
Movie Effect Model	0.9437046
Movie and User Effect Model	0.8655329
Regularized Movie and User Effect Model	0.8649857

```
## # A tibble: 4 x 2
##   method                                RMSE
##   <chr>                                <dbl>
## 1 Using mean only                      1.06
## 2 Movie Effect Model                   0.944
## 3 Movie and User Effect Model          0.866
## 4 Regularized Movie and User Effect Model 0.865
```

Regularization using movies, users, years and genres.

The approach utilized in the above model is implemented below with the added genres and release year effects.




```
## [1] 14
```

method	RMSE
Using mean only	1.0606506
Movie Effect Model	0.9437046
Movie and User Effect Model	0.8655329
Regularized Movie and User Effect Model	0.8649857
Reg Movie, User, Year, and Genre Effect Model	0.8626097

Results

RMSE overview

The RMSE values for the used models are shown below:

method	RMSE
Using mean only	1.0606506
Movie Effect Model	0.9437046
Movie and User Effect Model	0.8655329
Regularized Movie and User Effect Model	0.8649857
Reg Movie, User, Year, and Genre Effect Model	0.8626097

Concluding Remarks

The RMSE table shows an improvement of the model over different assumptions. The simplest model 'Using mean only' calculates the RMSE more than 1, which means we may miss the rating by one star (not good!!). Then incorporating 'Movie effect' and 'Movie and user effect' on model gives an improvement by 5% and 13.5%. This is substantial improvement given the simplicity of the model. A deeper insight in the data revealed some data point in the feathers have large effect on errors. So a regularization model was used to penalize such data points. The final RMSE is 0.8623 with an improvement over 13.3% with respect to the baseline model. This implies we can trust our prediction for movie rating given by a users.

References

1. https://github.com/johnfelipe/MovieLens-2
2. https://github.com/cmrad/Updated-MovieLens-Rating-Prediction