Predict Movie rating with MovieLens data

Partha Sarathi Mukherjee

5/12/2020

Executive Summary

Predicting a movie rating has huge financial impact. A good predictive rating will mean that the movie should be released in a massive scale with a big upfront advertisement budget. So it is essential that we get rating with high accuracy.

The MovieLens data set is a comprehensive dataset thou for this illustration we will limit the dataset to 10 million.

The dataset comprises of the following attributes

- Movie details like (MovieId, Title)
- Movie attributes like (Generes, timestamp)
- Movie viewers(userId)
- Movie ratings(rating)

A movie can belong in 1 or more of 19 different Genre.

Using the dataset provided, the objective is to build a machine learning model that can predict future movie ratings.

Key steps performed to build movie rating predictor machine learning model are:-

1. Data Visualization – How the rating data is distributed among the attributes?
2. Data Cleanup – Bad and missing data are trimmed.
3. Build Models – Use Matrix Factorization to build recomendation models.
4. Model Predictions – Models are evaluated against Test sets
5. Evaluate Models – Target is to build a model with RMSE < 0.86490

Analysis

There 2 methods succeeded in the original Netflix challenge

1. Matrix Factorization
2. Ensemble Model

In this task we use Matrix factorization. It is better suited for recommendation system and gives better RMSE with fewer variables.

Overall data distribution

```
library(tidyverse)
library(dplyr)
```

```
## [1] "Dimensions"
```

```
## [1] 9000055       6
```

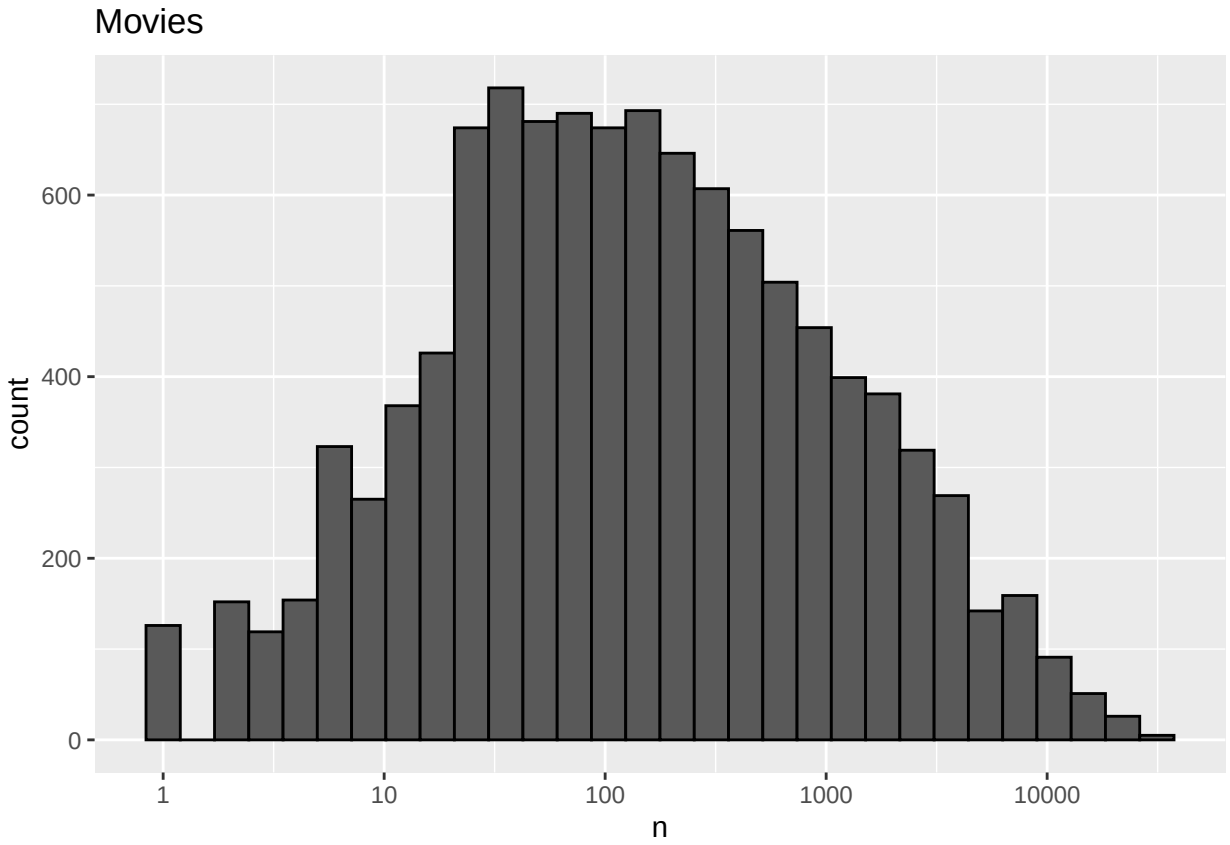
```
## [1] "Summary"
```

```
## '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 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|A
```

```
## [1] "Summary of distinct users and movies"
```

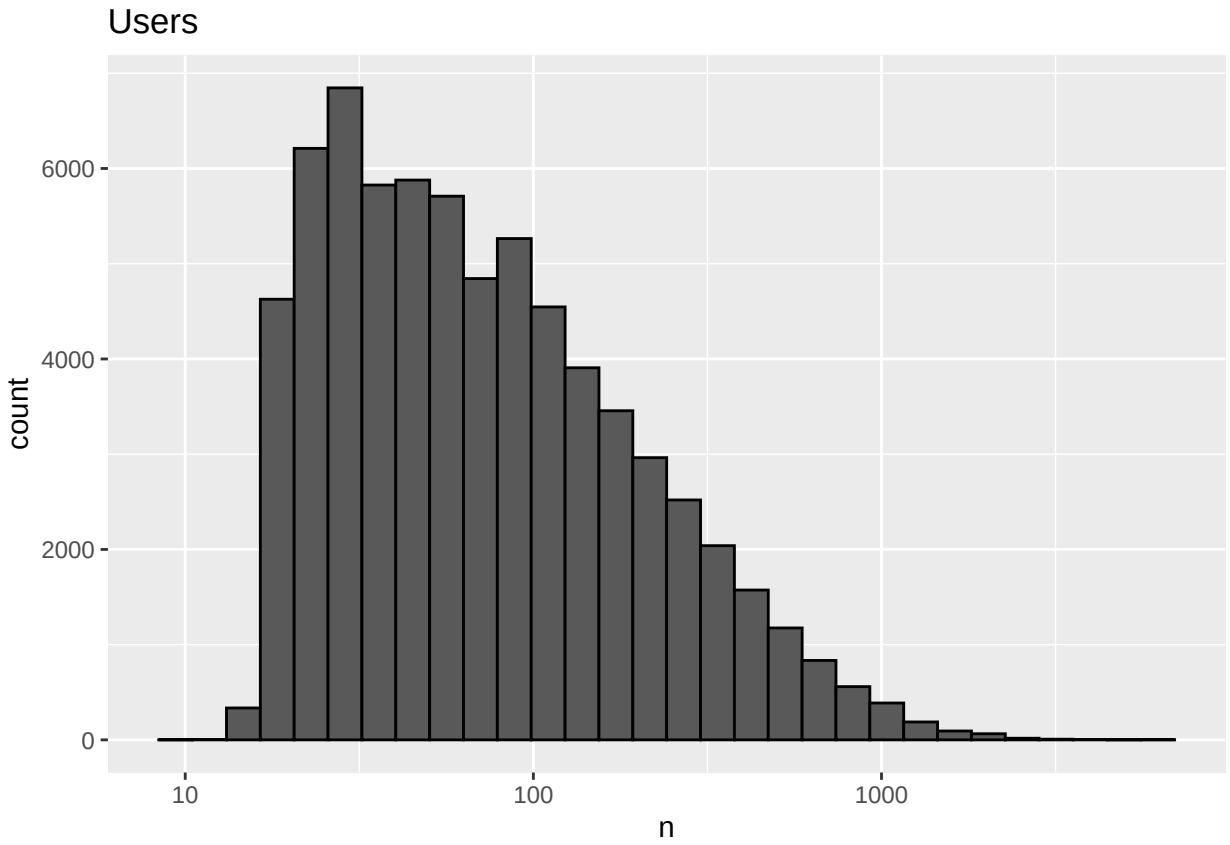
```
##   users movies
## 1 69878  10677
```

Distribution of rating per movie



Conclusion: Remove movies not rated by atleast 25 users.

Distribution of rating per user



Conclusion: Remove users not rating more than 25 movies.

Data Abnormality

On denormalization of the data and looking into all the genre, we found that only one movie had no rating (Pull My Daisy (1958)). The movie id is 8606.

Conclusion: Remove movie "Pull My Daisy" (1958) from the dataset.

Data Cleaning

Based on above conclusions, data is cleaned as

1. Remove users rating less than 25 movies – Removed 7551 users. 155124 movies removed. 2.29% decrease
2. Remove movies not rated by atleast 25 users + movie id 8606 – Removed 30576 movies. 0.4 % decrease.

Cleaned Data now looks like:

```
## [1] "Dimensions"

## [1] 8769418       6

## [1] "Summary"

## 'data.frame':    8769418 obs. of  6 variables:
##  $ userId   : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ movieId  : num  110 151 213 1148 1246 ...
##  $ rating   : num  4.5 4.5 5 4 4 4 3.5 3 3.5 2 ...
##  $ timestamp: int  1136075500 1133571026 1136075789 1133571121 1133570967 1133571071 1133571205 1133
##  $ title    : chr  "Braveheart (1995)" "Rob Roy (1995)" "Burnt by the Sun (Utomlyonnye solntsem) (19
##  $ genres   : chr  "Action|Drama|War" "Action|Drama|Romance|War" "Drama" "Animation|Children|Comedy|
```

Modeling – Martix Factorization

Steps to build Matrix Factorization model as:-

Step 1: Average Movie Rating

Build an model which predicts the same rating (average movie rating) regardless of the user.

Step 2: Add Movie Effect on Rating

Compute the movie effect based on the average difference between the actual rating and the average rating. Build a model with average movie rating from step 1 and movie effect value.

Step 3: Add User Effect on Rating

Compute the user effect based on the average difference between actual rating and the sum of average and movie rating obtained in step 2. Build a model with average movie rating from step 1, movie effect value from step 2 and user effect from this step.

Result

Using the above 3 steps, the final model RMSE dropped to 0.862.

Best performance requirement for this problem is achieved (RMSE $<$ 0.86490).

Method	RMSE
Step 1: Average Movie Rating	1.0579260
Step 2: Add Movie Effect on Rating	0.9404095
Step 2: Add User Effect on Rating	0.8628846

Conclusion

The Matrix Factorization method is the best approach to recommendation systems.

More steps can added to improve the solution like

1. User's Age: Amount of time user is in system from user's first rating,
2. Movie's Age: Amount of time passed from the first rating.
3. Movie's Expectation: How many people already rated the movie

and so on. . .