Movie Lens

Hephzibah Akindele

2022-09-04

Introduction

The Movie Lens project involves building a recommendation system for the movielens data set provided by the dslabs package in the HarvardX Data Science Capstone Course. The actual movie lens data set has millions of ratings but this project will use the 10M version of the dataset for ease of computation.

The Dataset

The dataset consists of 6 features and 9000055 entries. The features include: userId, movieId, rating, timestamp, title, genres.

1. `userId`: an integer from 1 to 71,567 signifies the user who made the rating.
2. `movieId`: an integer from 1 to 65,133 signifies which movie was rated.
3. `rating`: a multiple of 0.5, from 0.5 to 5.0.
4. `timestamp`: a `POSIXct` object representing the time at which the rating was made.
5. `title`: the name of the movie rated, suffixed which the year of release in parentheses.

There is no missing data in the dataset. These are the first entries on the dataset:

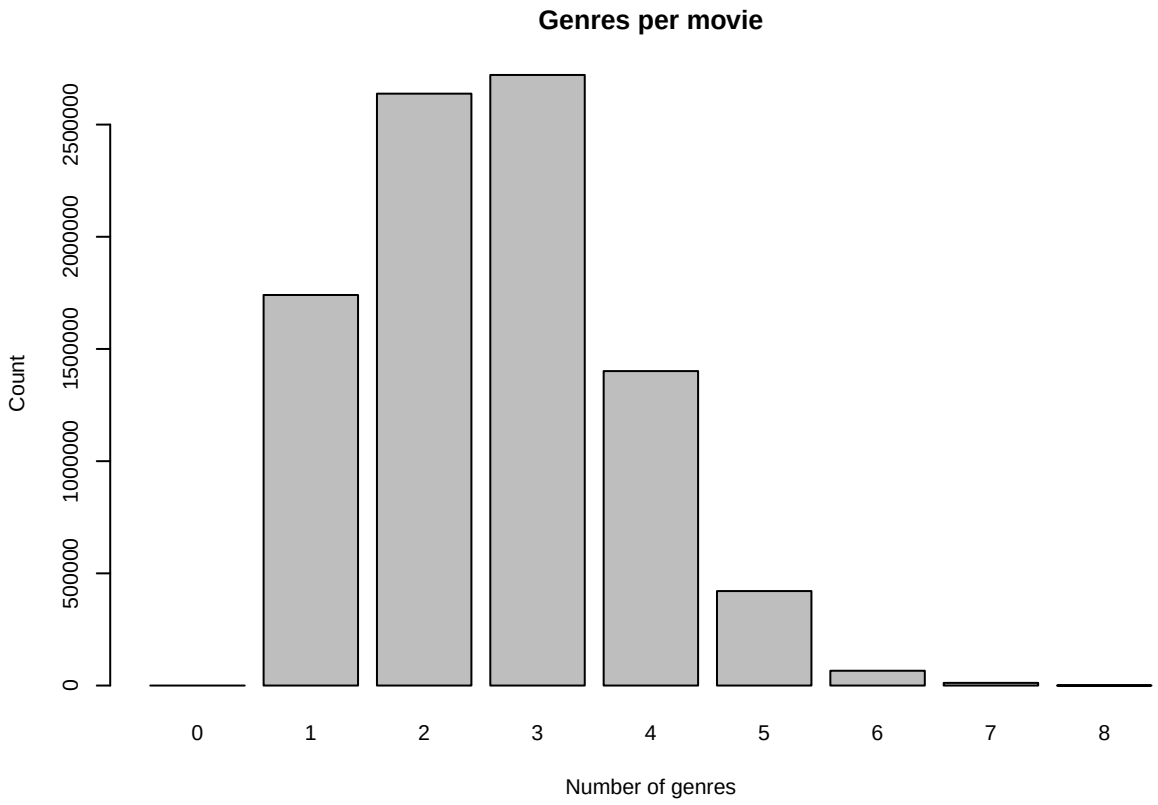
```
##    userId movieId rating timestamp                         title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
## 6:      1     355      5 838984474       Flintstones, The (1994)
##                            genres
## 1:                Comedy|Romance
## 2:          Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

There are 69878 unique User Ids i.e 69878 different users, and 10677 unique movie Ids i.e 10677different movies, 19 different genres: 'Action', 'Adventure', 'Animation', 'Children', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'IMAX', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western' and entries without genres are indicated by '(no genres listed)'.

Exploratory Data Analysis

Genres

A look at the first six entries on the data set shows that there are multiple genre combinations, the following plot shows the highest number of genres combined.

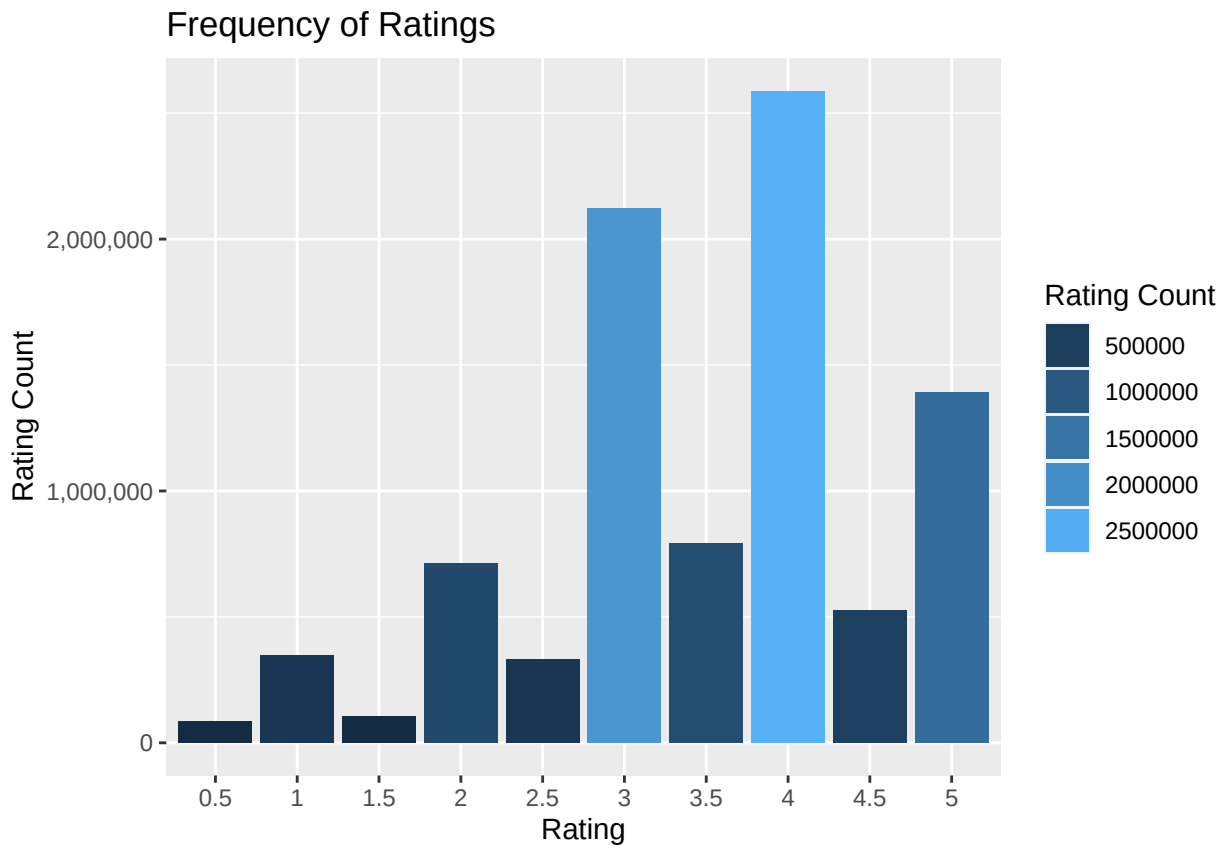


The genres feature has 796 different genre combinations and while analyzing the importance of each of the 19 genres might improve the accuracy of the machine learning model, computational limitations would not allow this project explore that aspect. Instead the genre combinations will be analyzed as they are.

Ratings

The average rating for all movies in the data set is NA . Analyzing the ratings feature reveals that the most frequently given ratings, the movie with the highest average rating, the users that rated the most movies, the most rated genre combination, the highest rated genre combination. etc.

The Most Frequently Given Ratings



The table shows the top 3 most given ratings are 4, 3 and 5 respectively.

The Highest Rated Movies

The table below shows the movies with highest average ratings and the lowest average ratings. It looks like the movies closer to the extremes of the rating scale i.e closer to 5 and 0.5, have only a few ratings.

```
## `summarise()` has grouped output by 'movieId'. You can override using the
## `.groups` argument.
```

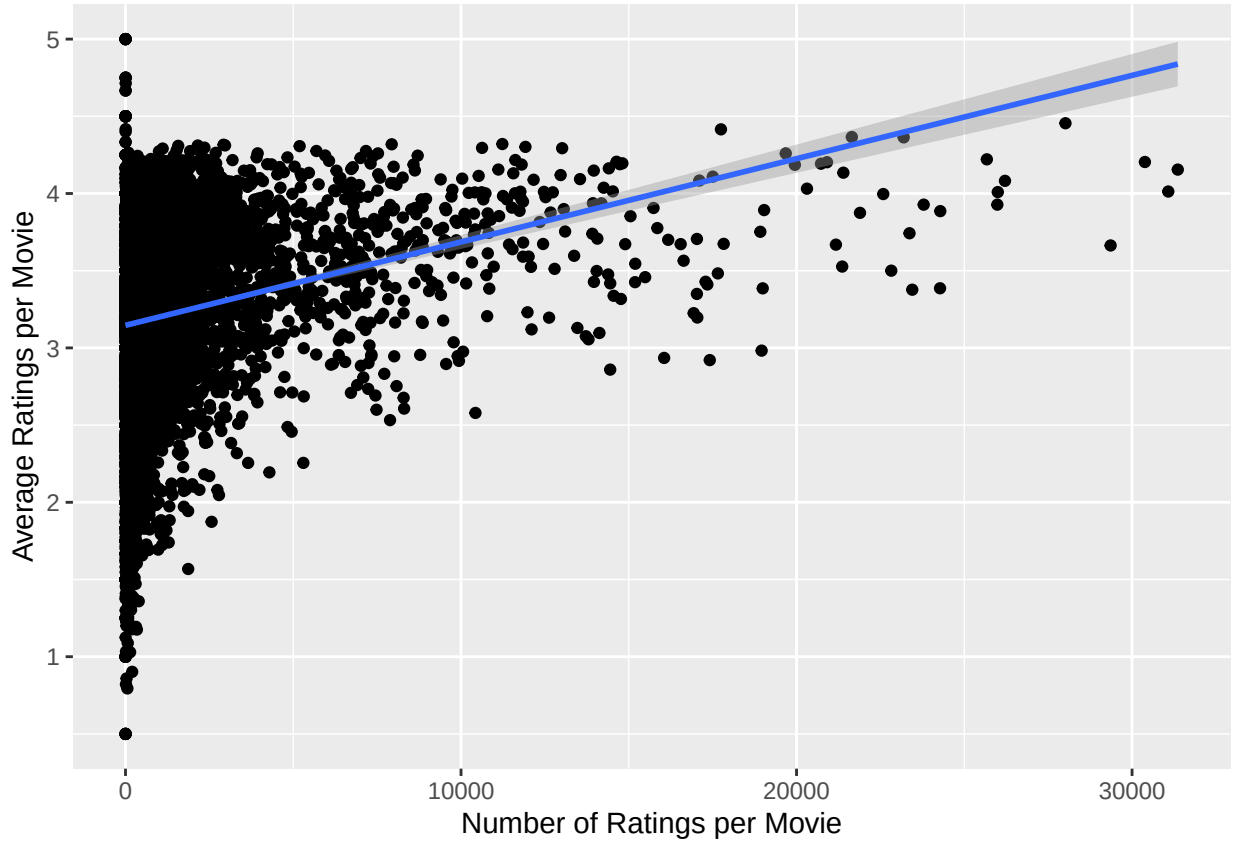
```
## # A tibble: 6 x 4
## # Groups:   movieId [6]
##   movieId title                                     average_rating count
##     <dbl> <chr>                                              <dbl> <int>
## 1    3226 Hellhounds on My Trail (1999)                         5     1
## 2   33264 Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)                  5     2
## 3   42783 Shadows of Forgotten Ancestors (1964)                5     1
## 4   51209 Fighting Elegy (Kenka erejii) (1966)                 5     1
## 5   53355 Sun Alley (Sonnenallee) (1999)                       5     1
## 6   64275 Blue Light, The (Das Blaue Licht) (1932)             5     1
```

```
## # A tibble: 6 x 4
## # Groups:   movieId [6]
##   movieId title                                     average_rating count
##     <dbl> <chr>                                              <dbl> <int>
## 1    8859 SuperBabies: Baby Geniuses 2 (2004)                0.795    56
```

```
## 2    5805 Besotted (2001)                                0.5       2
## 3    8394 Hi-Line, The (1999)                            0.5       1
## 4   61768 Accused (Anklaget) (2005)                      0.5       1
## 5   63828 Confessions of a Superhero (2007)              0.5       1
## 6   64999 War of the Worlds 2: The Next Wave (2008)      0.5       2
```

The scatter plot for the data helps us visualize a trend in the data. Movies that have been rated more times seem to have a higher average rating, until it gets closer to 5. Intuitively this makes some sense as popular good movies will be seen by more people and get more good ratings increasing the average ratings. This trend indicates that there will be some merit in including the frequency of rating per movie as a feature for the modelling process.

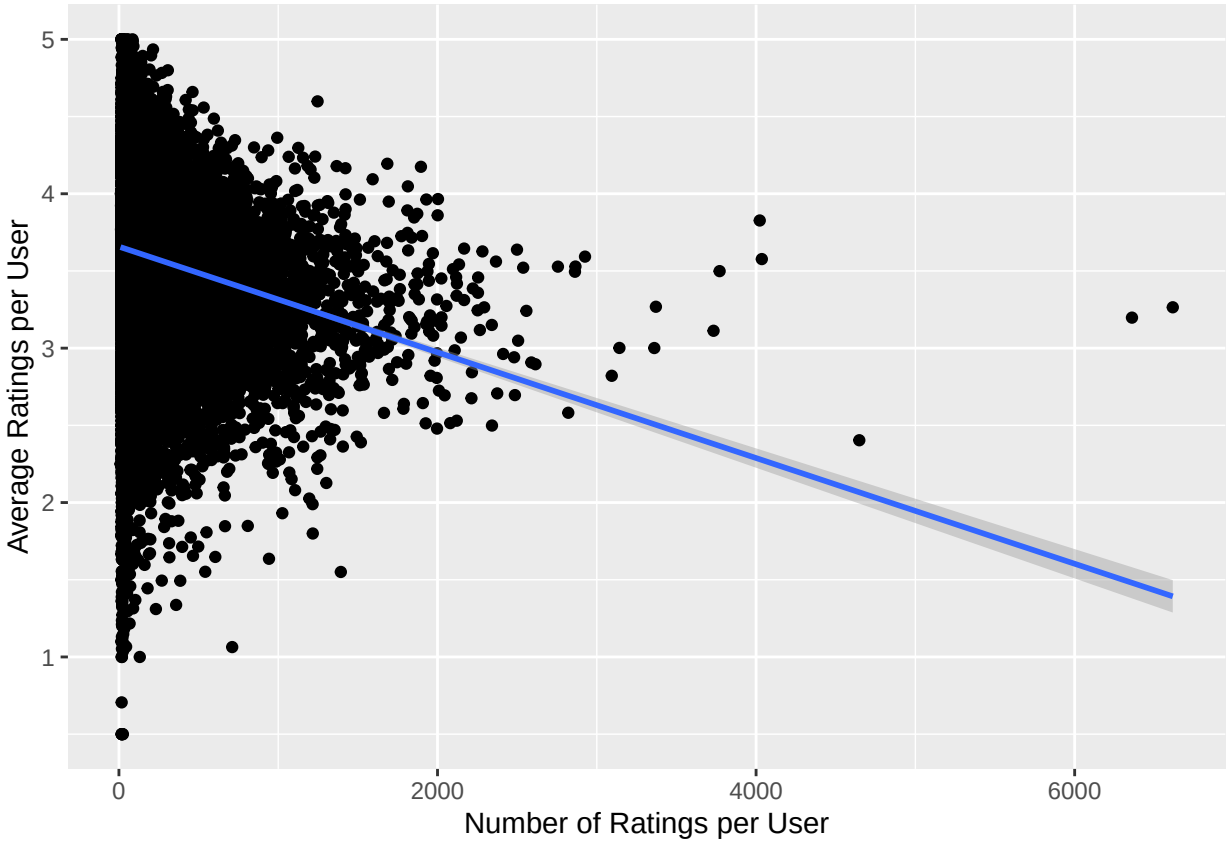
```
## `geom_smooth()` using formula 'y ~ x'
```



Ratings Per User

It's impossible for every user to rate every movie, so looking at the number of movies each user rated might give some insights to the users rating behavior.

```
## `geom_smooth()` using formula 'y ~ x'
```

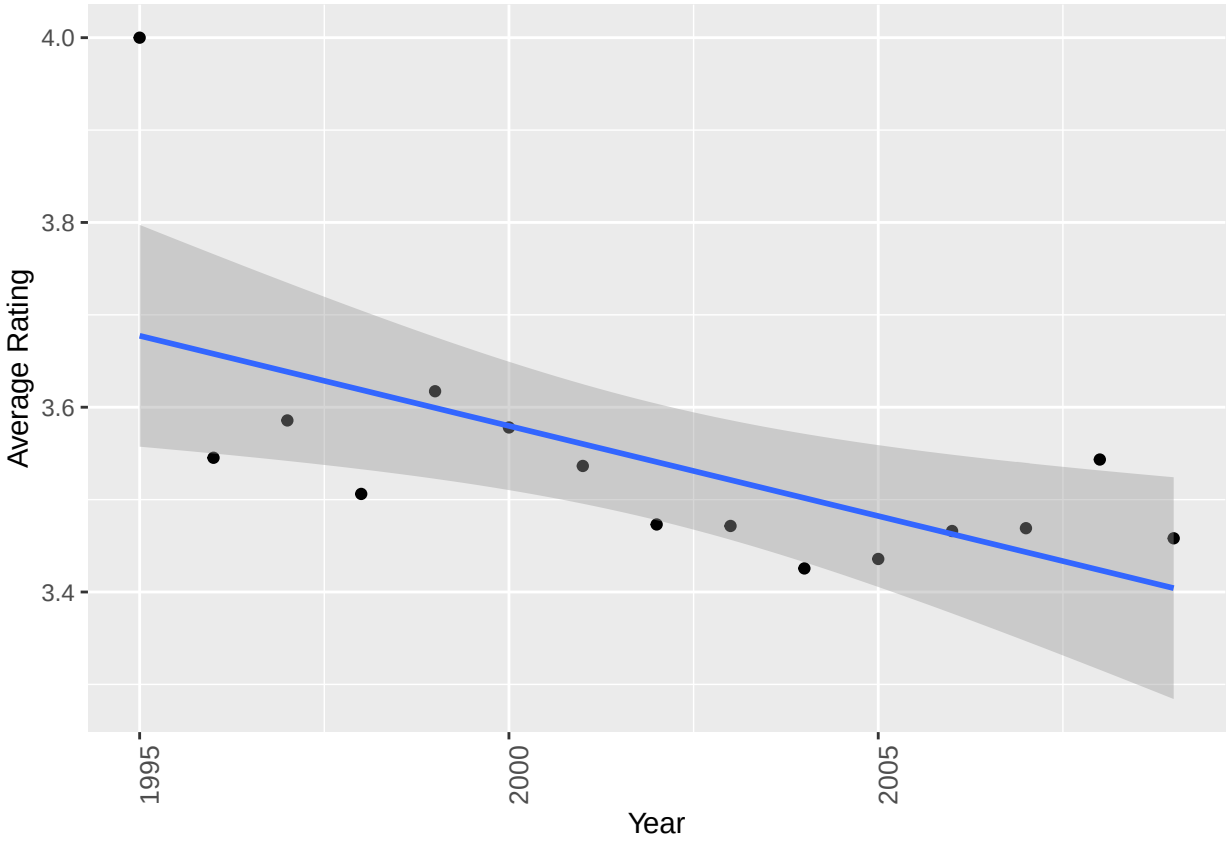


The scatter plot shows that there's a slight trend, the users average rating decreases as the number of movies a user rates increases. Thus the number of movies a user has rated and the users average rating have some predictive value and should be included in the modelling process.

Timestamp

No significant trends were found when comparing the year each movie was rated to the average ratings for that year, and as such average rating per year would be excluded from the modelling processes.

```
## `geom_smooth()` using formula 'y ~ x'
```



The time feature

Modelling

Model Evaluation (RMSE)

The metric for evaluating accuracy of the models in the project will be the Root Mean Square Error:

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

where y denotes the true values of movie ratings in the test set $\mathcal{T}$;

and $\hat{y}$ denotes the estimated values.

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

The 'edx' data set will be partitioned into a training set and a test set. These partitions will be used to evaluate the different models, and different stages of each model.

The validation data set is reserved for evaluating the RMSE of the final model.

Predicting with the mean

Firstly, we will make predictions with just the mean; to serve as a bench mark for accuracy of the models. A model that performs worse than blindly picking the mean rating, is a terrible model. This model assumes the variations in the ratings are due to random error.

This is the equation for making predictions based on just the mean:

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

where $Y_{u,i}$ is the prediction,

$\epsilon_{u,i}$ is the independent error, and

μ the expected "true" rating for all movies.

Predicting the mean gives the following naive RMSE 1.0599094. The smaller the RMSE the more accurate the predictions. This RMSE will serve as a benchmark for the predictions we will make with the other models for this project.

Linear Regression

The exploratory data analysis highlighted some features that have some predictive value like the number of times a movie was rated and the number of times a user rated a movie.

A linear model with just the data in the original data set would have a formula:

$$Y = \alpha + \beta_1(\text{userId}) + \beta_2(\text{movieId}) + \beta_3(\text{genres}) + \beta_4(\text{timestamp}) + \epsilon$$

and an RMSE of 1.0575638 which is a little less than the Naive RMSE 1.0599094 gotten from predicting with the mean.

Adding terms from the exploratory data analysis, would give us this formula:

$$Y = \alpha + \beta_1(\text{userId}) + \beta_2(\text{movieId}) + \beta_3(\text{genres}) +$$

$$\beta_4(\text{mode\_user\_rating}) + \beta_5(\text{mode\_movie\_rating}) + \beta_6(\text{rating\_count}) +$$

$$\beta_7(\text{average\_rating\_user}) + \beta_8(\text{average\_rating\_movie}) + \epsilon$$

This new model improves the RMSE to 0.8708413 , which is a great improvement from the RMSE of the previous linear model (1.0575638).

Conclusion

The aim of this project is to produce a model with an RMSE less than 0.87750 and the final linear model RMSE on the validation set is '0.8445729'. The Exploratory Data Analysis proved very helpful as the insights highlighted terms that drastically reduced the RMSE.

Recommendations that could not be done due computational limitations, but would further reduce the RMSE and improve modelling accuracy include:

1. Creating dummy variables for the genres feature of the edx data set and analyzing them individual instead of in combinations.
2. Using Random Forest Regression to predict the ratings instead of a multiple linear regression.