HarvardX Data Science Certificate - PH125.9x

MovieLens Project

Youssef Bougarrani

07/24/2021

Contents

1 Introduction :

This project is related to the capstone of the HarvardX data science professional certificate PH125.9x. It's on the famous MovieLens recommendation system. MovieLens itself is a research site run by GroupLens Research group at the University of Minnesota. The first automated recommender system was developed there in 1993. Recommendation systems use ratings that users have given to items to make specific recommendations for other users. The goal of this project will be to predict ratings based on a training data set. The data set will be uploaded from grouplens.org website.

After an exploratory analysis, the best recommender system built on the training data set will be testesd on a validation data set. The performance of the models is based on the root mean squared error RMSE.

1.1 Session information

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.2 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.1.0    magrittr_2.0.1    tools_4.1.0       htmltools_0.5.1.1
##  [5] yaml_2.2.1        stringi_1.6.2     rmarkdown_2.8     knitr_1.33
##  [9] stringr_1.4.0     xfun_0.23         digest_0.6.27     rlang_0.4.11
## [13] evaluate_0.14
```

1.2 Dataset Description

The dataset is available in several snapshots. The ones that were used in this analysis were the 10 millions ratings (10 millions rows).

1.3 Loading data

First the data is imported from the url address and partitionned to edx and validation data sets.

2 Analysis

2.1 Cleaning

2.1.1 Cleaning edx data set

We will perform necessary cleaning and some transformations so that the data better suits our needs.

We convert timestamp to a human readable object on edx data set.

And we extract the year of release.

userId	movieId	rating	timestamp	title	release	genres
1	122	5	1996	Boomerang (1992)	1992	Comedy|Romance
1	185	5	1996	Net, The (1995)	1995	Action|Crime|Thriller
1	292	5	1996	Outbreak (1995)	1995	Action|Drama|Sci-Fi|Thriller
1	316	5	1996	Stargate (1994)	1994	Action|Adventure|Sci-Fi
1	329	5	1996	Star Trek: Generations (1994)	1994	Action|Adventure|Drama|Sci-Fi
1	355	5	1996	Flintstones, The (1994)	1994	Children|Comedy|Fantasy

We separate genre to have one unique genre per each column.

userId	movieId	rating	timestamp	title	release	genres
1	122	5	1996	Boomerang (1992)	1992	Comedy
1	122	5	1996	Boomerang (1992)	1992	Romance
1	185	5	1996	Net, The (1995)	1995	Action
1	185	5	1996	Net, The (1995)	1995	Crime
1	185	5	1996	Net, The (1995)	1995	Thriller
1	292	5	1996	Outbreak (1995)	1995	Action

2.1.2 Cleaning validation data set

The validation data will NOT be used for training, developing, or selecting the algorithm and it will ONLY be used for evaluating the RMSE of the final algorithm.

We do simply cleaning as we dit for edx data set. We convert timestamp to a human readable object.

And we extract the year of release. Then, we separate genre.

2.2 Initial exploration and visualization

In this section we will take the first look at the loaded data frame.

Dimensions

The data is 9000055 observations and 6 columns.

```
## [1] 23371423        7
```

Head

A preview of the data is shown below

userId	movieId	rating	timestamp	title	release	genres
1	122	5	1996	Boomerang (1992)	1992	Comedy
1	122	5	1996	Boomerang (1992)	1992	Romance
1	185	5	1996	Net, The (1995)	1995	Action
1	185	5	1996	Net, The (1995)	1995	Crime
1	185	5	1996	Net, The (1995)	1995	Thriller
1	292	5	1996	Outbreak (1995)	1995	Action
1	292	5	1996	Outbreak (1995)	1995	Drama
1	292	5	1996	Outbreak (1995)	1995	Sci-Fi
1	292	5	1996	Outbreak (1995)	1995	Thriller
1	316	5	1996	Stargate (1994)	1994	Action
1	316	5	1996	Stargate (1994)	1994	Adventure
1	316	5	1996	Stargate (1994)	1994	Sci-Fi

Missing values

Ok, looks like there is no missing data.

	Missing Values
userId	0
movieId	0
rating	0
timestamp	0
title	0
release	0
genres	0

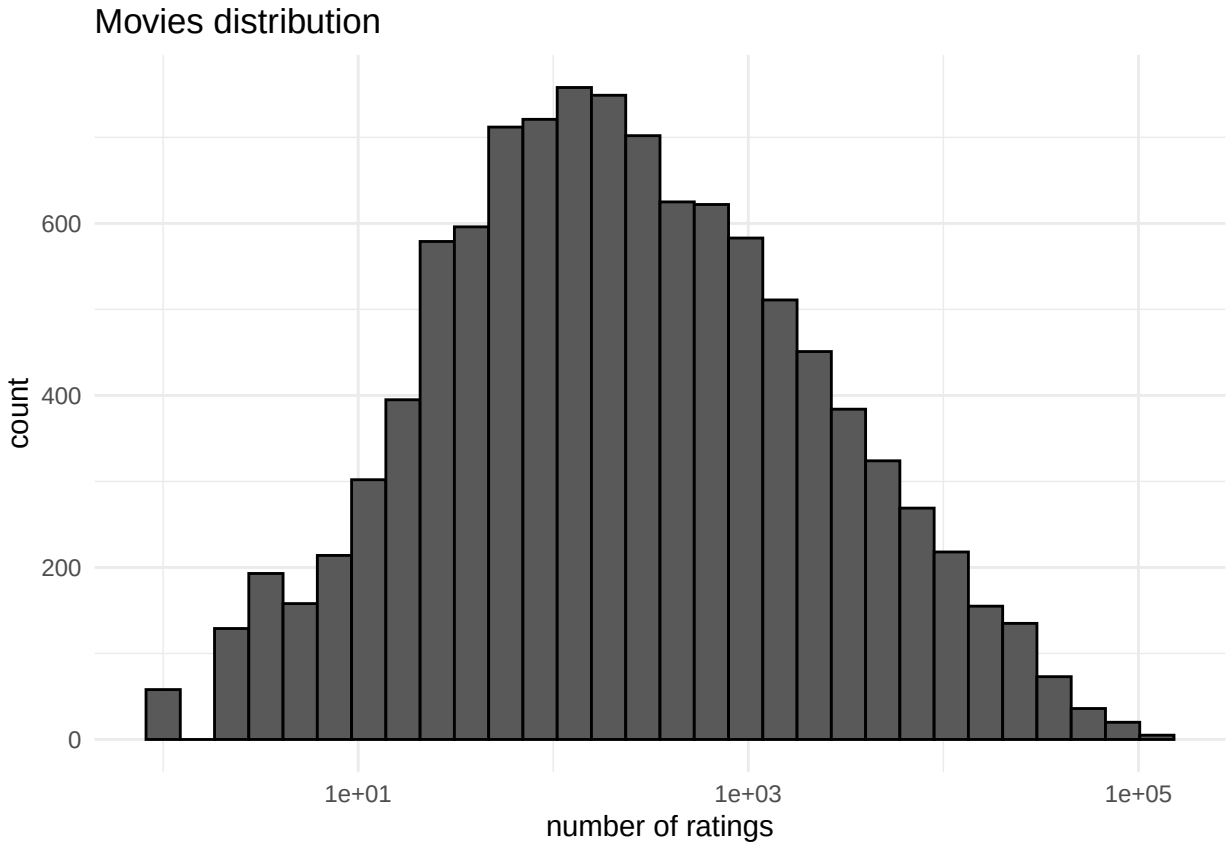
Number of Users and Movies

Unique Users	Unique Movies
69878	10677

2.3 Exploratory Data Analysis

Movie

First we can plot the number of ratings per movie.



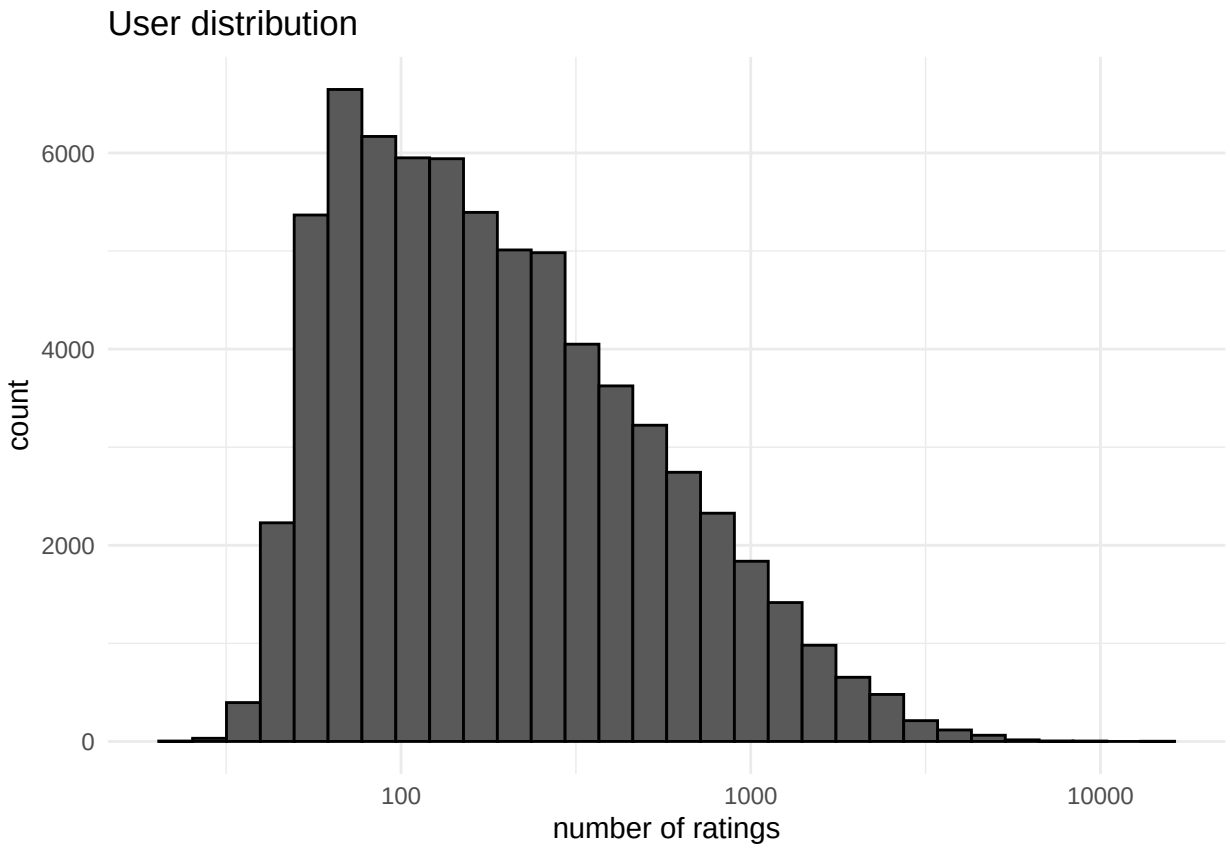
It can be seen that majority of movies have been rated approximately between 50 to 150 times.

First twelve most rated movies

movieId	title	n
356	Forrest Gump (1994)	124316
1	Toy Story (1995)	118950
480	Jurassic Park (1993)	117440
380	True Lies (1994)	114115
588	Aladdin (1992)	105865
592	Batman (1989)	97108
364	Lion King, The (1994)	94605
296	Pulp Fiction (1994)	94086
780	Independence Day (a.k.a. ID4) (1996)	93796
593	Silence of the Lambs, The (1991)	91146

User

The number of users' ratings frequency can also be plotted.



The plot represents that majority of users have rated approximately between 30 to 120 movies.

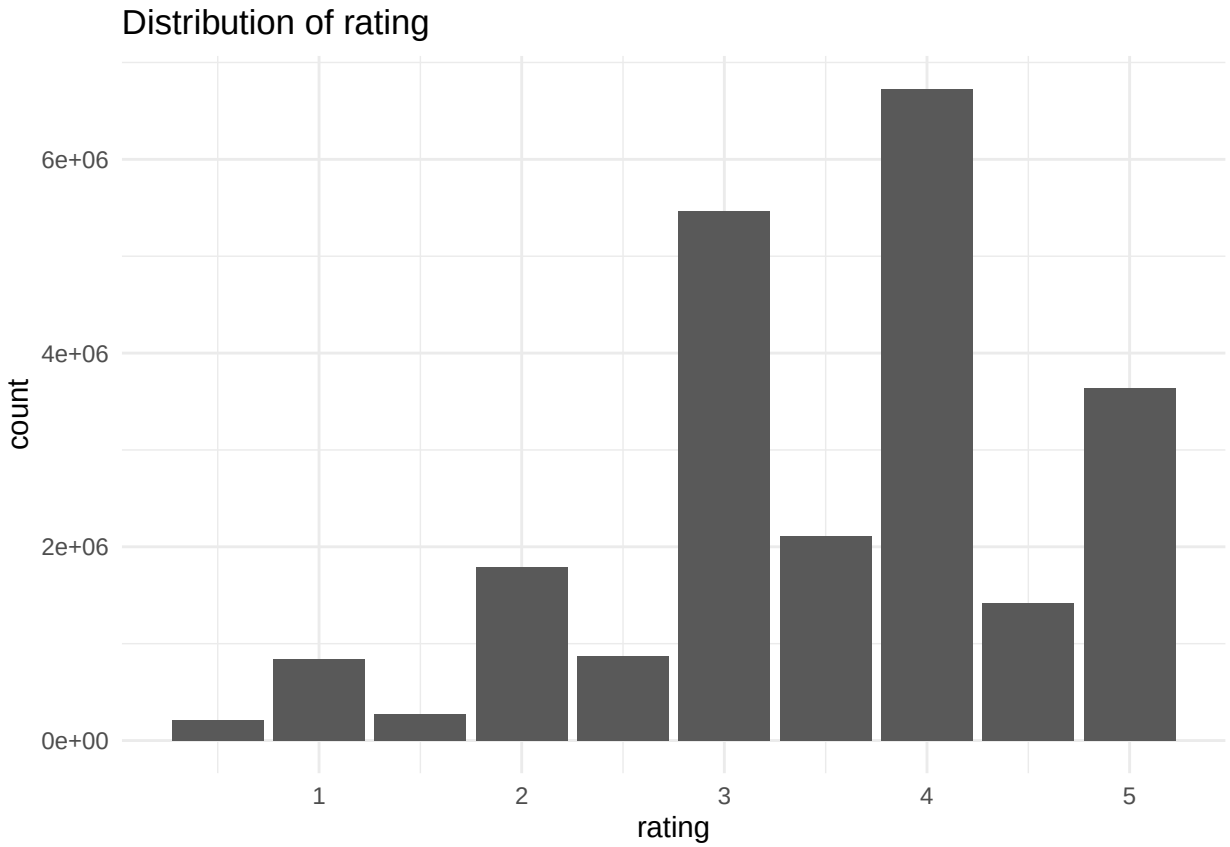
Ten most active users

userId	n
59269	13496
67385	13360
14463	9121
27468	8953
3817	8637
68259	8519
19635	8221
58357	7649
63134	7521
6757	7087

Rating

There are 10 possible choices rating between 0.5 and 5.

rate	count
0.5	215932
1	844336
1.5	276711
2	1794243
2.5	874290
3	5467061
3.5	2110690
4	6730401
4.5	1418248
5	3639511



As can be seen, users who have rated movies tend to assign higher scores in general. Actually, most ratings are a 4-star, followed by a 3-star rating.

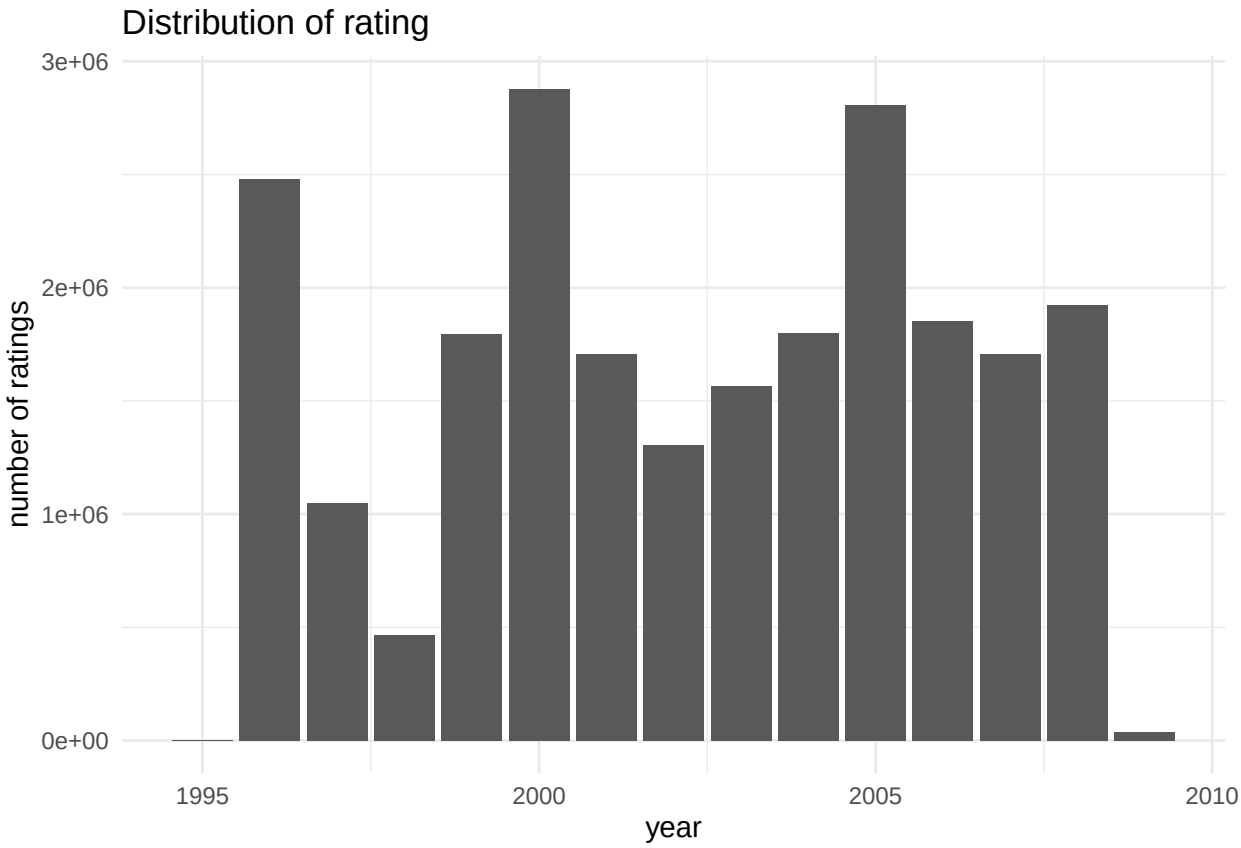
Year

The rating starts in 1995 and ends in 2009.

```
## [1] 1995 2009
```

Number of rating per year :

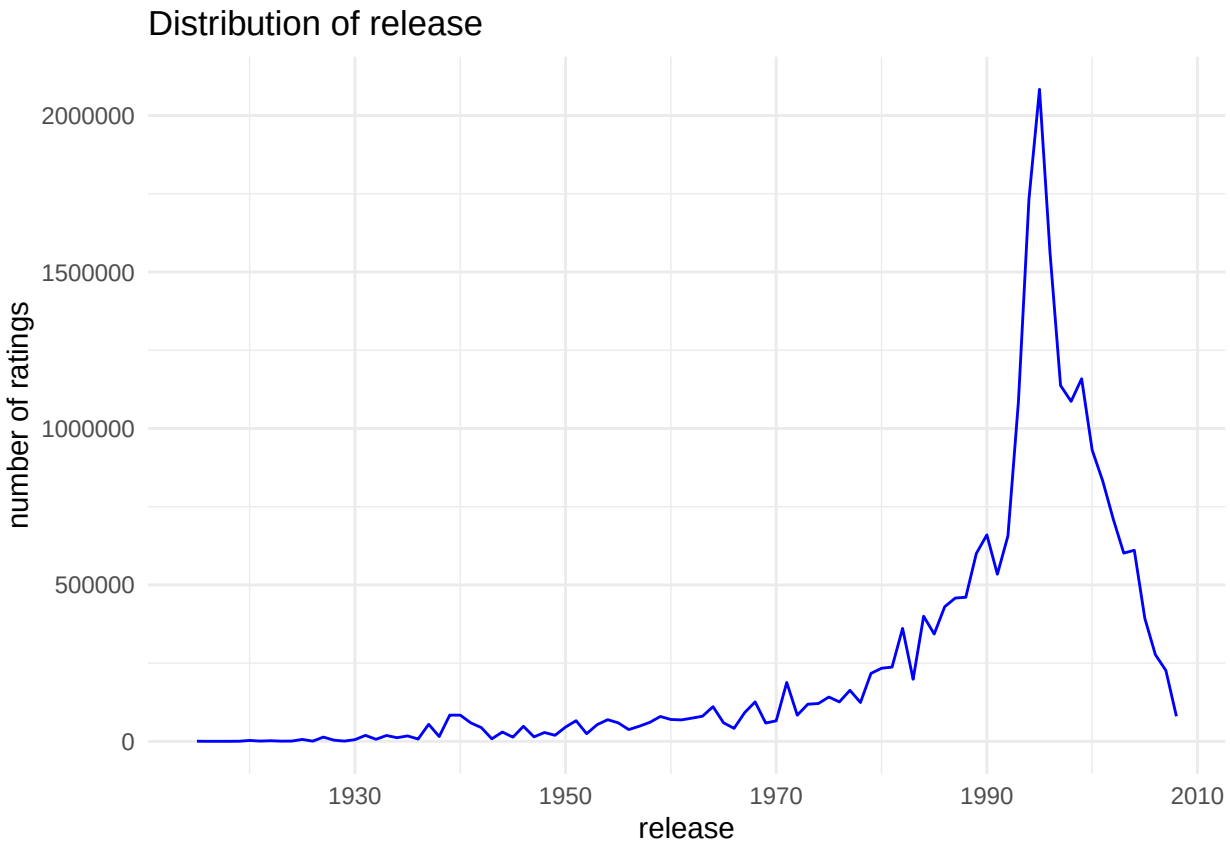
year	count
2000	2879265
2005	2807323
1996	2480192
2008	1921946
2006	1852912
2004	1799565
1999	1795921
2007	1708356
2001	1706355
2003	1567347
2002	1303338
1997	1046676
1998	466296
2009	35925
1995	6



Year of release

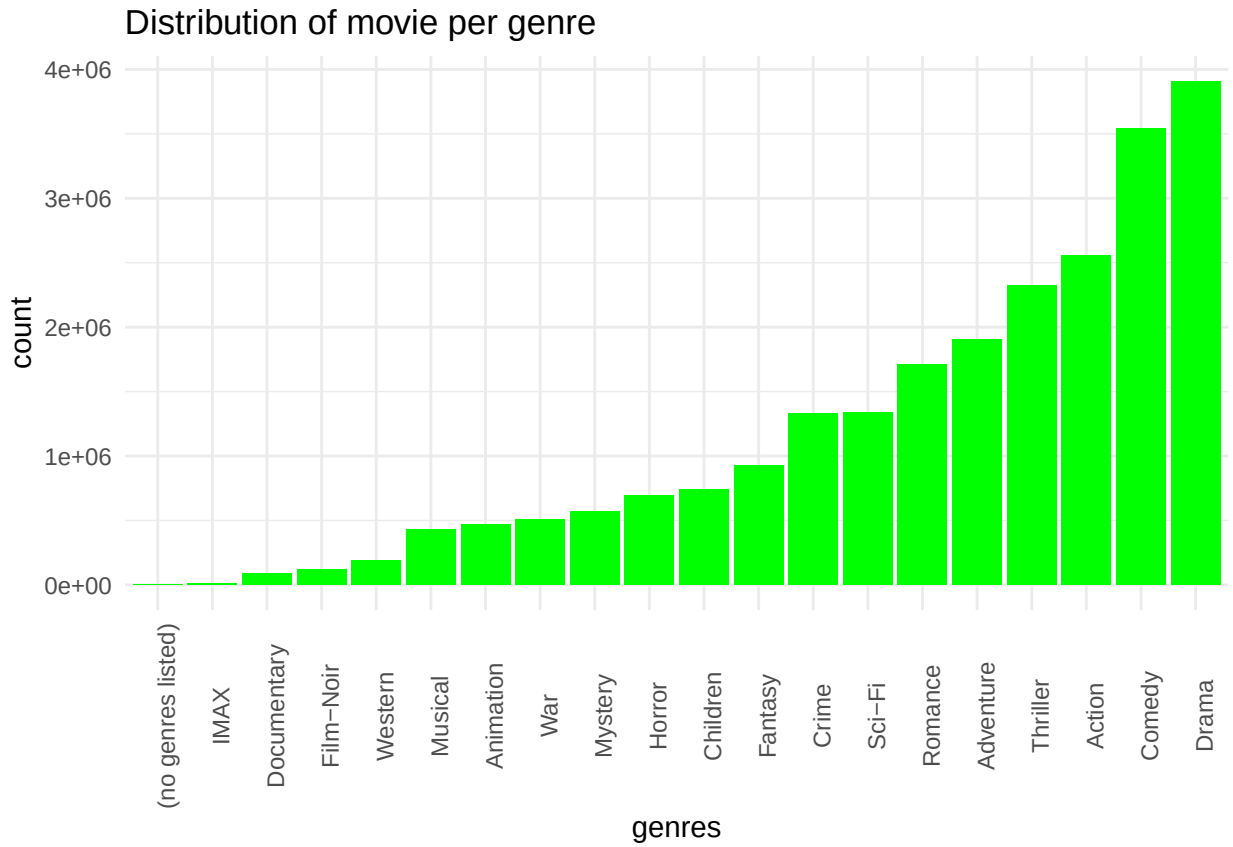
Year of release starts in 1915.

```
## [1] 1915 2008
```



Genre

genre	count
Drama	3910127
Comedy	3540930
Action	2560545
Thriller	2325899
Adventure	1908892
Romance	1712100
Sci-Fi	1341183
Crime	1327715
Fantasy	925637
Children	737994
Horror	691485
Mystery	568332
War	511147
Animation	467168
Musical	433080
Western	189394
Film-Noir	118541
Documentary	93066
IMAX	8181
(no genres listed)	7



2.4 Data analysis

2.4.1 Residual Mean Square Error

We create a function of RMSE to measure the accuracy of prediction.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

2.4.2 Create train_set and test_set

train_set is 75% of whole edx data, and test_set is 25% remaining. We select movies and users who aren't in the train_set and remove them from the test_set, then add them to the train_set.

3 Results

3.1 Step-by-step model

3.1.1 Just the mean : Predict rating as the average

If all movies are rated by the average, the RMSE will be :

method	rmse
just the avg	1.052046

3.1.2 Movie effect

RMSE of movie effect added to just the average.

method	rmse
just the avg	1.0520464
movie_eff	0.9410267

3.1.3 User effect

RMSE of user effect added to just the average and movie effect.

method	rmse
just the avg	1.0520464
movie_eff	0.9410267
user_eff	0.8578161

3.1.4 Genre effect

RMSE of genre effect added to just the average, movie effect and user effect.

method	rmse
just the avg	1.0520464
movie_eff	0.9410267
user_eff	0.8578161
genre_eff	0.8577266

3.1.5 Timestamp effect

RMSE of timestamp effect added to just the average, movie effect, user effect, and genre effect.

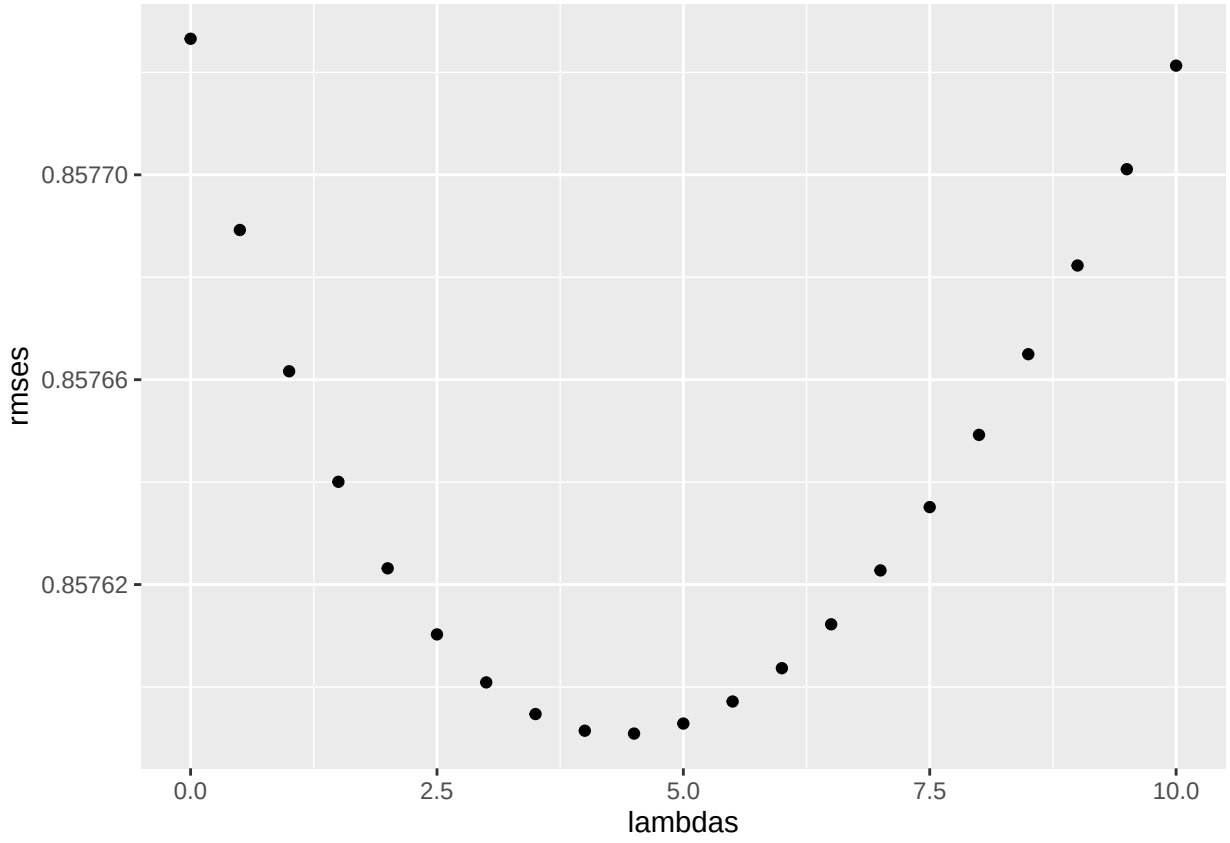
method	rmse
just the avg	1.0520464
movie_eff	0.9410267
user_eff	0.8578161
genre_eff	0.8577266
timestamp_eff	0.8577203

3.2 Regularization

The supposed "best" and "worst" movies were rated by very few users. These movies were mostly obscure ones. This is because with just a few users, we have more uncertainty. These are noisy estimates that we should not trust, especially when it comes to prediction. Large errors can increase our RMSE, so we would rather be conservative when unsure.

Regularization permits us to penalize large estimates that are formed using small sample sizes. It has commonalities with the Bayesian approach.

We build regularization with user_score, movie_score, and genre_score.
Lambda is a tuning parameter. We Calculate the best of it.



Lambda which minimises RMSE is 4.5

3.2.0.1 Predicting

method	rmse
just the avg	1.0520464
movie_eff	0.9410267
user_eff	0.8578161
genre_eff	0.8577266
timestamp_eff	0.8577203
regularization	0.8575909

4 Conclusion

For the final model, we used the best performing model in the previous section, which is the regularised model. The validation data is set in a way so the users and movies in the data are all present in the edx data.

we apply the best model on the validation set.

RMSE of validation

method	rmse
just the avg	1.0520464
movie_eff	0.9410267
user_eff	0.8578161
genre_eff	0.8577266
timestamp_eff	0.8577203
regularization	0.8575909
validation	0.8633693

The final RMSE is 0.863. There is much ways for improvement. One of them is to stratify the user_scores by genres, as the same user might love some genre but hate others.