Data Science Capstone - Movielens Recommendation Model

Carlos Yáñez Santibáñez

February 20, 2020

Abstract

Abstract

Contents

1 Introduction

This document presents a machine learning model that aim to predict (recommend) movie ratings for particular users of a streaming or review platform. This report is a capstone assignment for HarvardX's Professional Certificate in Data Science, which can be taken at the edX platform. The program is available at https://www.edx.org/professional-certificate/harvardx-data-science.

The data used in this excercise comes from the Movielens 10M Dataset. After using the donwload code provided in the course, the resulting dataframe contains observations with an individual movie rating from a particular user, each with the below attributes:

1. **userId** : Unique user identifier
2. **movieId** : Unique movie identified
3. **rating** : Rating given to this movie by the particular user.
4. **timestamp** : Timestamp indicating when the the user submitted the rating.
5. **title** : Title of the film and its release year in brackets. Please note that different movies can have the same name (e.g. remakes), thus movieId is a better unique identifier.
6. **genres** : List of all genres in which this movie can be clasiffied.

The goal of this project is to generate a model that can predict a particular movie rating for each user, as close as possible to the actual rating given to each film. In order to assess the model performance, the **Root Square Mean Error (RMSE)** will be calculated for a validation dataset. Training and validation dataset are generated using the code provided in the assignment instructions. In order to aim for a full mark, this report will target for a RMSE lower than **0.86490**.

The starting point of this project is the model presented in section 33.7 of the course's textbook. From there, the following steps are presented:

1. Analysis of the textbook's model and possible ways to improve it.
2. Improvement to the model via user clustering.
3. Tuning improved model.
4. Evaluation against validation dataset.
5. Conclusion.

The following sections present the above in detail.

2 Methods and Analysis

As mentioned in the introduction, this reports starts with the model presented in the textbook and then explore options to improve segmenting the users. However before conducting any modelling, the data needs to be cleaned up a little bit and then split into training and testing set.

In terms of data cleaning, three operations have been considered, namely:

- Remove the year from the title and storing in a different column. This may be useful for modelling.
- For the same reasons, convert the timestamp into a year number.

- Finally, a sequential number will be added (*row_id*). This will be create a single unique ID for each record (instead of a userId and movieId combination) and maybe useful to speed up filtering, given the large size of the dataset.

For this purposes, the below function **Tidy_Up** has been created. The code for this function is available on the included R file. Using the below code the train and test sets are created.

```
# tidy up data
edx_tidy_temp <- Tidy_Up(edx)

# divide Train and Test datasets
set.seed(200, sample.kind = "Rounding")
edx_train_test <- Train_Test(edx_tidy_temp)

# remove temporary and original dataset to avoid filling up memory.
rm("edx_tidy_temp", "edx")
```

Below, this is a sample of the generated training set (*edx_train_test$train*).

Table 1: Training Dataset - Sample

userId	movieId	rating	timestamp	title	genres	release_year	rating_date	rating_year	row_id
1	122	5	838985046	Boomerang	Comedy|Romance	1992	1996-08-02	1996	1
1	185	5	838983525	Net, The	Action|Crime|Thriller	1995	1996-08-02	1996	2
1	292	5	838983421	Outbreak	Action|Drama|Sci-Fi|Thriller	1995	1996-08-02	1996	3
1	316	5	838983392	Stargate	Action|Adventure|Sci-Fi	1994	1996-08-02	1996	4
1	329	5	838983392	Star Trek: Generations	Action|Adventure|Drama|Sci-Fi	1994	1996-08-02	1996	5
1	355	5	838984474	Flintstones, The	Children|Comedy|Fantasy	1994	1996-08-02	1996	6
1	356	5	838983653	Forrest Gump	Comedy|Drama|Romance|War	1994	1996-08-02	1996	7
1	362	5	838984885	Jungle Book, The	Adventure|Children|Romance	1994	1996-08-02	1996	8
1	364	5	838983707	Lion King, The	Adventure|Animation|Children|Drama|Musical	1994	1996-08-02	1996	9
1	370	5	838984596	Naked Gun 33 1/3: The Final Insult	Action|Comedy	1994	1996-08-02	1996	10

Once completed data cleaning and split, the first step is to implement the model presented in the textbook and asses it's results. This model estimates rating by assuming that a particular rating from a particular user can be calculated as a deviation (bias) from the average rating for all movies (the *true* rating). In order to account for cases where movies and users don't have many reviews against them, weighting parametres have been added. This can be expressed by the below equation:

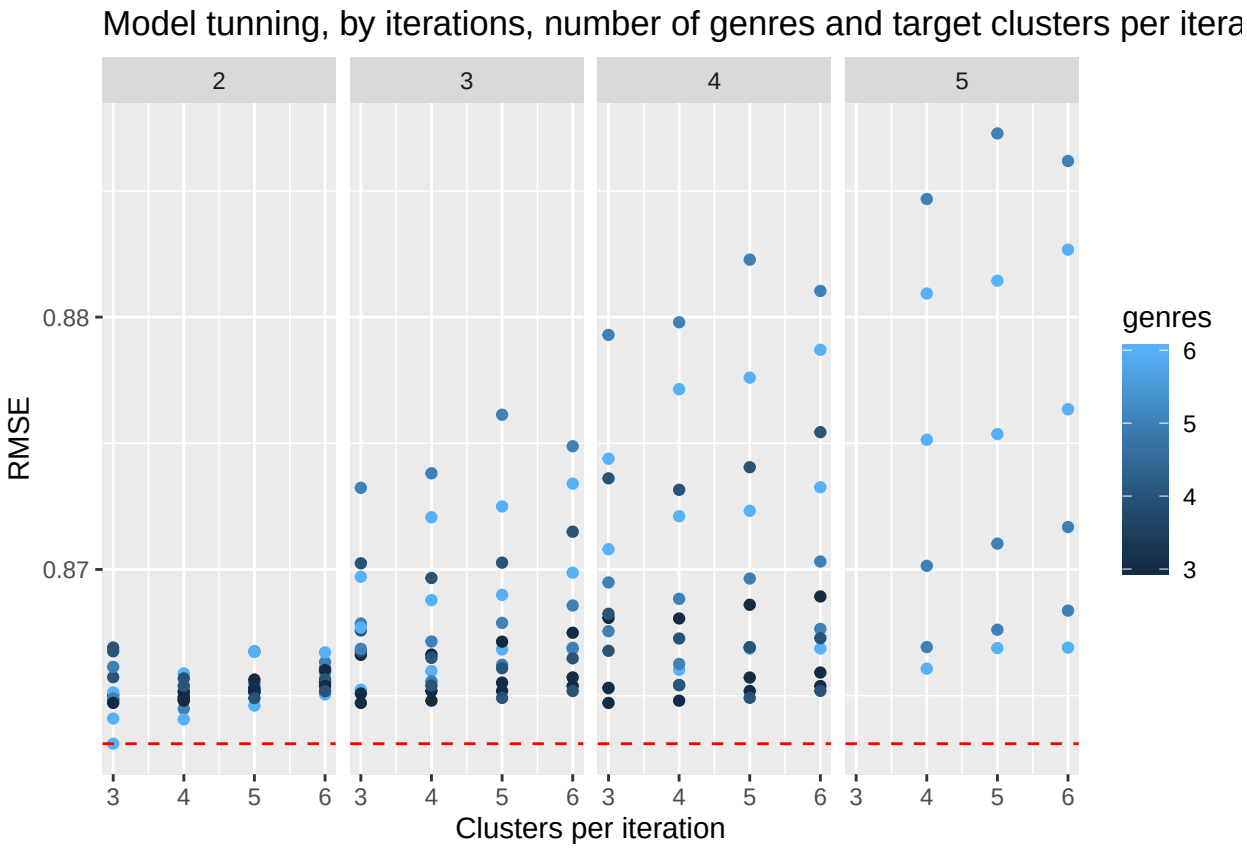
$$predictedrating = \hat{\mu} + b_i + b_u$$

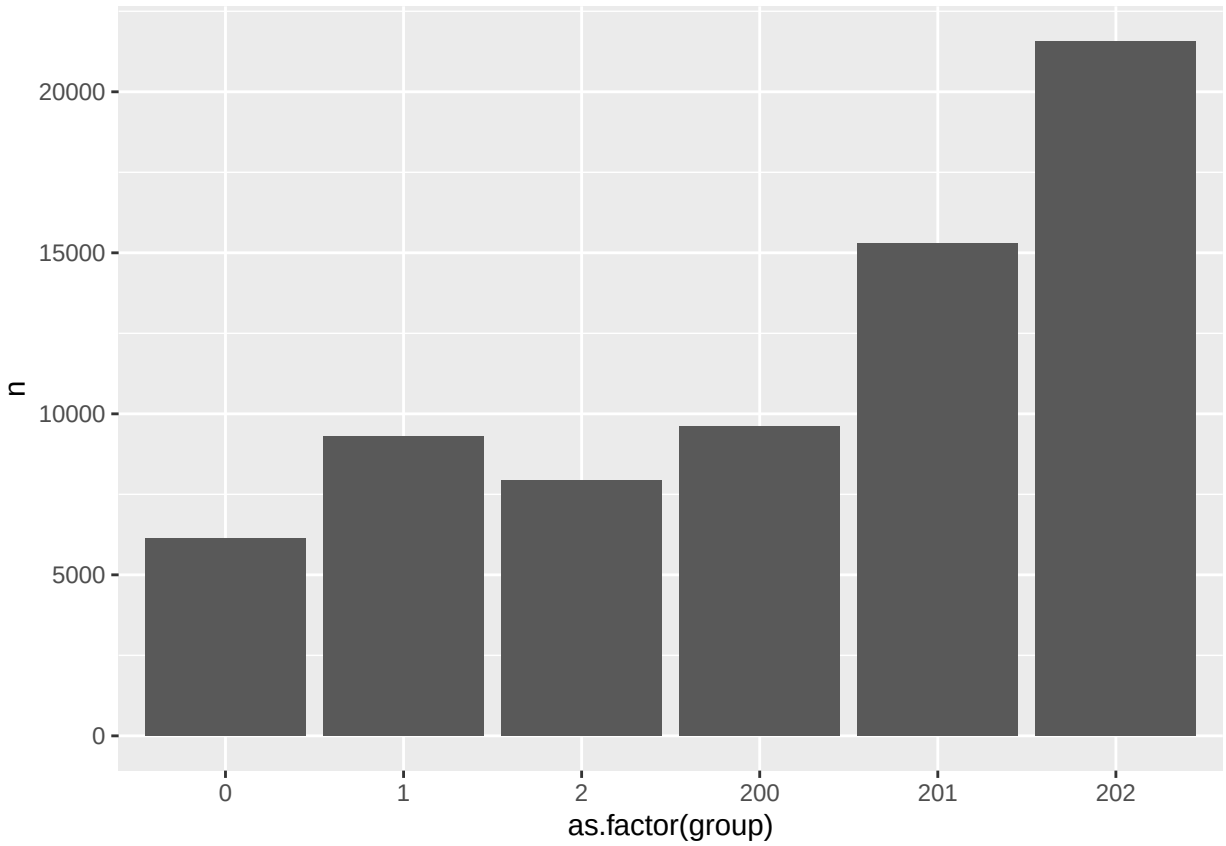
where $\hat{(\mu)}$ is the average rating for all movies in the dataset, and b_i and b_u being the movie bias and user bias (respectively), defined as follows:

$$b_i = (1)\sum_{j=1}^{n}(\hat{\mu} - rating_j)$$

Table 2: Optimal Clustering Parametres

clustering	iterations	cutoff	genres	cluster_n	RMSE	RMSE_1	RMSE_2	RMSE_3	method_1	method_2	method_3
GMM	2	0.5	6	3	0.8630969	0.8629808	1.050733	NaN	0.9994422	0.0005578	0






```
## # A tibble: 6 x 2
##   group     n
##   <dbl> <int>
## 1     0  6140
## 2     1  9287
## 3     2  7944
## 4   200  9622
## 5   201 15308
## 6   202 21577
```

```
## # A tibble: 20 x 11
##        k lambda_1 lambda_2  RMSE RMSE_1 RMSE_2 RMSE_3 method_1 method_2 method_3
##    <dbl>    <dbl>    <dbl> <dbl>  <dbl>  <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
##  1     1      2        4   0.862  0.862   1.10    NaN    0.999 0.000619        0
##  2     1      2        4.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
##  3     1      2        3.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
##  4     1      2.5      4   0.862  0.862   1.10    NaN    0.999 0.000619        0
##  5     1      2.5      4.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
##  6     1      2        5   0.862  0.862   1.10    NaN    0.999 0.000619        0
##  7     1      2.5      3.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
##  8     1      2.5      5   0.862  0.862   1.10    NaN    0.999 0.000619        0
##  9     1      2        3   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 10     1      2        5.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
```

```
## 11     1      2.5      3   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 12     1      2.5      5.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
## 13     1      2        6   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 14     1      2        2.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
## 15     1      2.5      6   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 16     1      2.5      2.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
## 17     1      1.5      4   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 18     1      3        4   0.862  0.862   1.10    NaN    0.999 0.000619        0
## 19     1      1.5      4.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
## 20     1      3        4.5 0.862  0.862   1.10    NaN    0.999 0.000619        0
## # ... with 1 more variable: cases <int>
```

```
## # A tibble: 5 x 5
## # Groups:   k, lambda_1 [2]
##       k lambda_1 lambda_2 Avg_RMSE Avg_Method_1
##   <dbl>    <dbl>    <dbl>    <dbl>        <dbl>
## 1     3      2        4      0.862        0.999
## 2     3      2.5      4      0.862        0.999
## 3     3      2        3.5    0.862        0.999
## 4     3      2.5      3.5    0.862        0.999
## 5     3      2        4.5    0.862        0.999
```

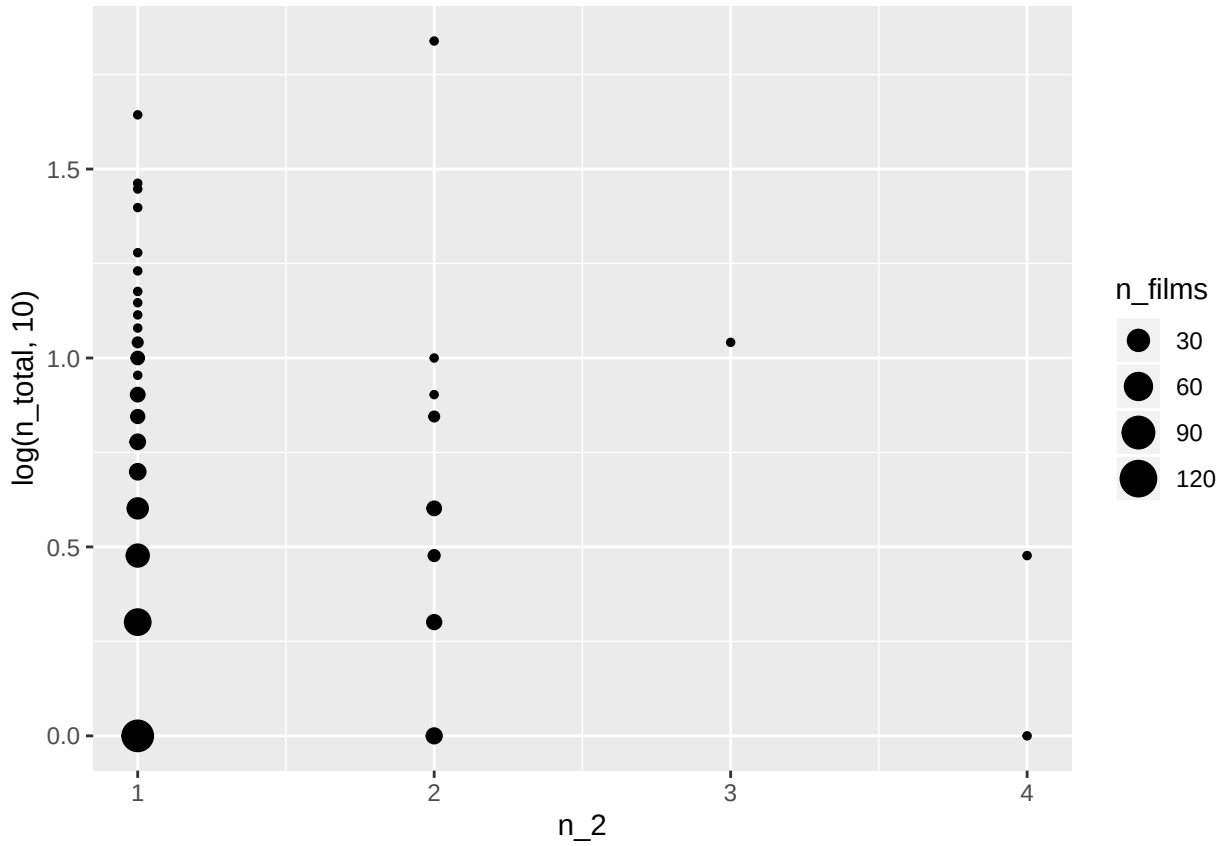

```
## # A tibble: 1 x 7
```

```
##    RMSE RMSE_1 RMSE_2 RMSE_3 method_1 method_2 method_3
##   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
## 1 0.862  0.862   1.05    NaN    0.999 0.000558        0
```

```
## # A tibble: 441 x 2
##    title                                  n
##    <chr>                              <int>
##  1 Dying of Laughter (Muertos de risa)     4
##  2 Bitter Sugar (Azúcar amarga)            3
##  3 Evil Aliens                             3
##  4 Reflections In A Golden Eye             3
##  5 All Night Long                          2
##  6 Anna                                    2
##  7 As in Heaven ( Så som i Himmelen )      2
##  8 Baby Doll                               2
##  9 Battle for Haditha                      2
## 10 Biker Boyz                              2
## # ... with 431 more rows
```

3 Results

```
## # A tibble: 1 x 7
##    RMSE RMSE_1 RMSE_2 RMSE_3 method_1 method_2 method_3
##   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
## 1 0.861  0.861   1.08    NaN     1.00 0.000473        0
```



title	n_2	n_total
Road House	2	69
Hills Have Eyes, The	1	44
River, The	1	29
Cradle 2 the Grave	1	28
Heaven	1	25
Walking Dead, The	1	19
Sunrise: A Song of Two Humans	1	17
Revenge of Frankenstein, The	1	15
Innocence	1	14
Dark Star	1	13

4 Conclusion