MovieLens

Kicheol Kim

Nov. 04. 2021

Introduction

MovieLens dataset is movie rating data, provided by GroupLens (research lab at University of Minnesota), to build a recommender system. This project is developing movie recommendation algorithm using MovieLens 10M dataset. MovieLens 10M dataset includes 10 million ratings for 10,000 movies by 72,000 users.
RMSE (root-mean-square error) will be used to evaluate predictions in the validation set.

- RMSE defined by :

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

Download and create dataset

- Create edx set (training and test set), validation set (final hold-out test set) :
  MovieLens 10M dataset was downloaded from the GroupLens website (https://grouplens.org/datasets/movielens/10m/). The dataset split into edx (90% of total ratings) and validation (10% of total ratings) set. The edx set will be used for training and test, and validation set will be used for final model validation.

1) Exploratory analysis

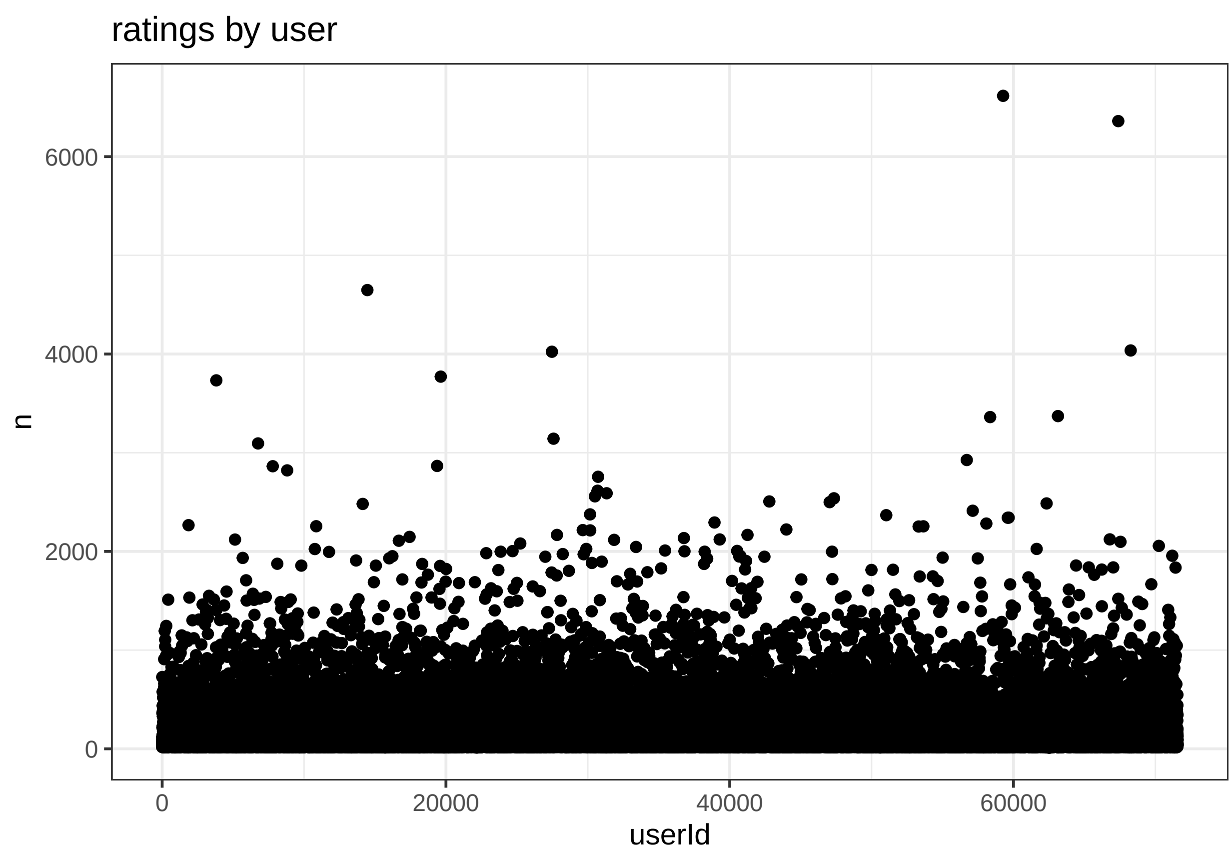
The training set (edx) includes 9000055 ratings consist of 69878 users and 10677 movies. The validation set (validation) includes 999999 ratings consist of 68534 users and 9809 movies.

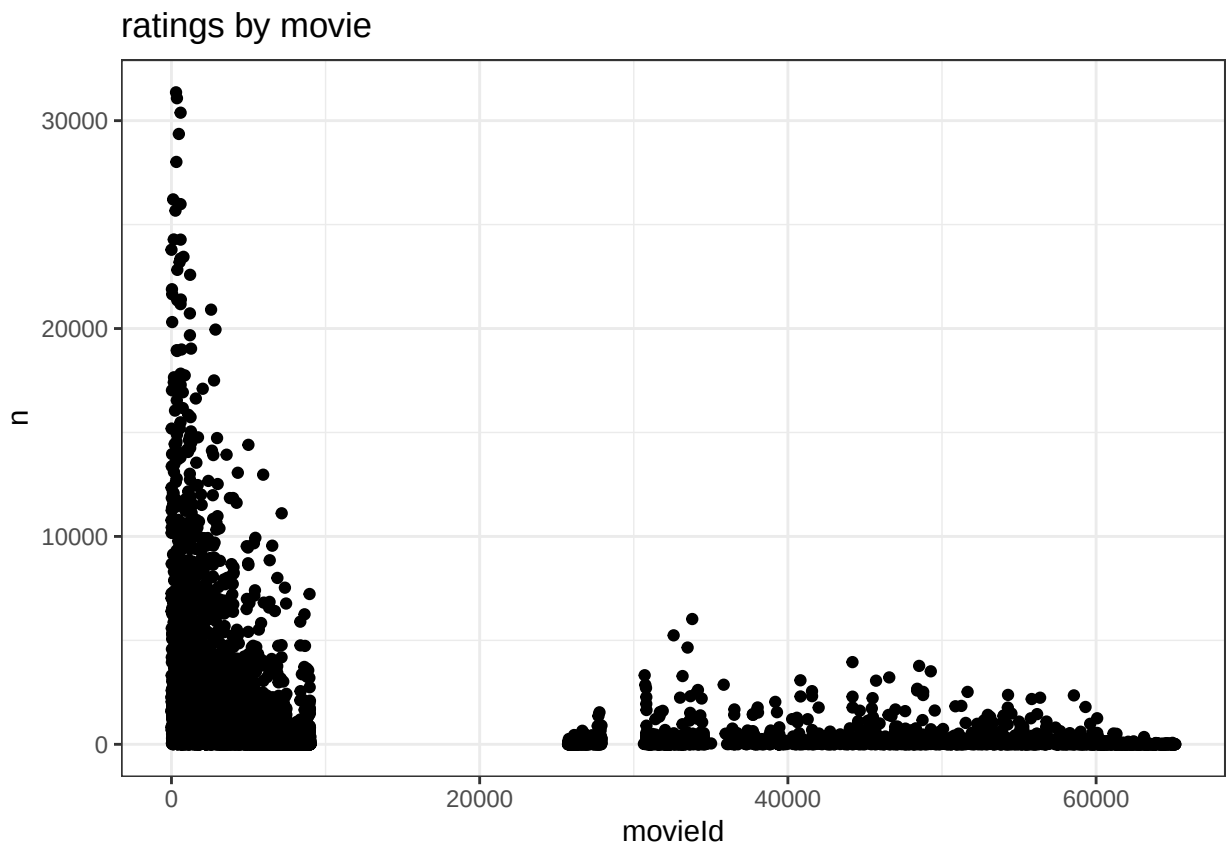
In the edx dataset, a user who rate most movies rated 6616 movies. On the other hand, a user who least movie rated was only rated 10 movies. The most rated movie was Pulp Fiction (1994) (number of ratings = 31362), 126 movies were rated by 1 user only.

```
##    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
```

```
##    userId movieId rating timestamp
## 1:      1     231      5 838983392
## 2:      1     480      5 838983653
## 3:      1     586      5 838984068
## 4:      2     151      3 868246450
## 5:      2     858      2 868245645
## 6:      2    1544      3 868245920
##                                                   title
## 1:                                 Dumb & Dumber (1994)
## 2:                                 Jurassic Park (1993)
## 3:                                    Home Alone (1990)
## 4:                                       Rob Roy (1995)
## 5:                                Godfather, The (1972)
## 6: Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                   genres
## 1:                                Comedy
## 2:       Action|Adventure|Sci-Fi|Thriller
## 3:                        Children|Comedy
## 4:               Action|Drama|Romance|War
## 5:                            Crime|Drama
## 6: Action|Adventure|Horror|Sci-Fi|Thriller
```





2) pre-process dataset

I converted timestamp into the date value. In addition, since genre have multiple rows for the same movie after separated rows, I make new dataset with separated rows for genre and used it for genre training.

Training

- Create function for RMSE computation

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

Method 1. average ratings

The average ratings can be used for simplest recommendation system.

method	RMSE
average ratings	1.061202

- RMSE for average ratings = 1.0612018

Method 2. movie effect

Some movie tends to get a good rating, but some movie are not. Different movies are rated differently. As we are adding movie effect in the model, we can consider it in the recommendation system.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087

- RMSE for movie effect = 0.9439087

Method 3. movie + user effect

Likewise, some user tends to give a good rating, but some user give a bad rating. We can also consider user effect in the training. - Number of ratings for each user (top 10 user with most number of ratings)

```
## Selecting by n
```

```
## # A tibble: 10 x 2
##    userId     n
##     <int> <int>
##  1  59269  6616
##  2  67385  6360
##  3  14463  4648
##  4  68259  4036
##  5  27468  4023
##  6  19635  3771
##  7   3817  3733
##  8  63134  3371
##  9  58357  3361
## 10  27584  3142
```

Here, I tested user effect only and movie+user effect together.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488

Although it's not as much as movie effect, user effect also improved RMSE. In addition, when I combined both movie and user effect, it improved RMSE much better.

- RMSE for user effect only = 0.9947953
- RMSE for movie+user effect = 0.8653488

Method 4. movie + user + date effect

We also have the date that user rated a movie. The year or date (season) also could affect ratings.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393

The date effect slightly improved RMSE (1.0603797) compared to RMSE using average rating (1.0612018). The combination of movie+user+date effect provides the best RMSE so far.

- RMSE for date effect only = 1.0603797
- RMSE for movie+user+date effect = 0.8648393

Method 5. movie + user + date + genres effect

We have one more feature that is genre of the movie. Since each movie categorized into multiple genres, I split genres into individual genre (multiple genres for a movie) to apply genre effect in the training.

```
##    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                time       date
## 1:                Comedy|Romance 1996-08-02 11:24:06 1996-08-02
## 2:          Action|Crime|Thriller 1996-08-02 10:58:45 1996-08-02
## 3:  Action|Drama|Sci-Fi|Thriller 1996-08-02 10:57:01 1996-08-02
## 4:       Action|Adventure|Sci-Fi 1996-08-02 10:56:32 1996-08-02
## 5: Action|Adventure|Drama|Sci-Fi 1996-08-02 10:56:32 1996-08-02
## 6:       Children|Comedy|Fantasy 1996-08-02 11:14:34 1996-08-02
```

```
## # A tibble: 6 x 8
##   userId movieId rating timestamp title      genres  time                date
##    <int>   <dbl>  <dbl>     <int> <chr>      <chr>   <dttm>              <chr>
## 1      1     122      5 838985046 Boomerang~ Comedy  1996-08-02 11:24:06 1996-0~
## 2      1     122      5 838985046 Boomerang~ Romance 1996-08-02 11:24:06 1996-0~
## 3      1     185      5 838983525 Net, The ~ Action  1996-08-02 10:58:45 1996-0~
## 4      1     185      5 838983525 Net, The ~ Crime   1996-08-02 10:58:45 1996-0~
## 5      1     185      5 838983525 Net, The ~ Thrill~ 1996-08-02 10:58:45 1996-0~
## 6      1     292      5 838983421 Outbreak ~ Action  1996-08-02 10:57:01 1996-0~
```

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393
genre effect	1.0519422
movie+user+date+genre effect	0.8626515

The genre doesn't have stong impact to RMSE although it's more than date effect. The combination of movie+user+date+genre effect provides the best RMSE.

- RMSE for genre effect only = 1.0519422
- RMSE for movie+user+date+genre effect = 0.8626515

Regularization

Can we more improve the current RMSE? Some non-famous movies have high average ratings because of very small number of ratings.

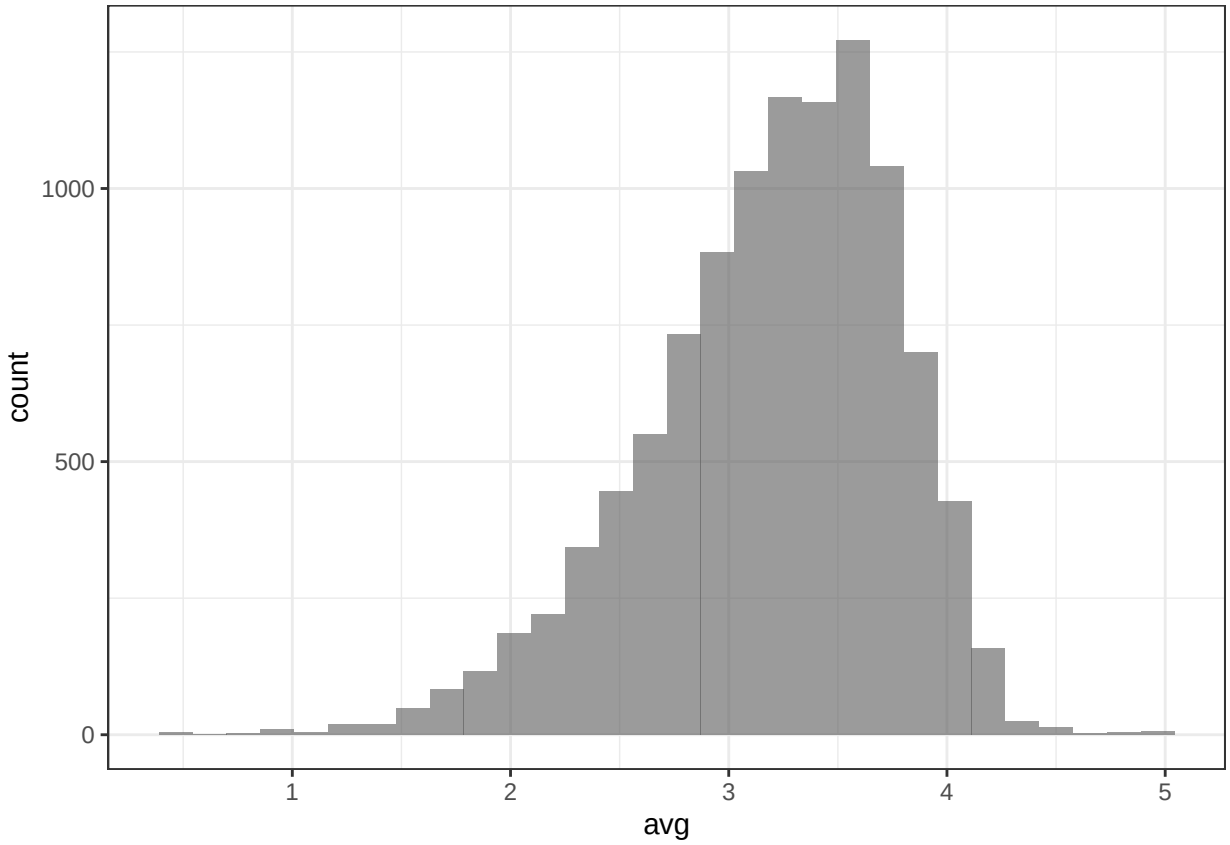
For example, average rating of "Hellhounds on My Trail (1999)" is 5 but it's rated only 1 time. This average rating is higher than "Pulp Fiction (1994)" which is rated 31362 times and average rating is 4.15.

The histogram shows distribution of average ratings. But the top or bottom average rating movies seem doesn't makes sense. They are not famous movies but have very high average ratings. However, they were rated by 1 or very few users.

```
## # A tibble: 10,677 x 4
## # Groups:   movieId [10,677]
##    movieId     n title                                                avg_rating
##      <dbl> <int> <chr>                                                     <dbl>
##  1     296 31362 Pulp Fiction (1994)                                        4.15
##  2     356 31079 Forrest Gump (1994)                                        4.01
##  3     593 30382 Silence of the Lambs, The (1991)                           4.20
##  4     480 29360 Jurassic Park (1993)                                       3.66
##  5     318 28015 Shawshank Redemption, The (1994)                           4.46
##  6     110 26212 Braveheart (1995)                                          4.08
##  7     457 25998 Fugitive, The (1993)                                       4.01
##  8     589 25984 Terminator 2: Judgment Day (1991)                          3.93
##  9     260 25672 Star Wars: Episode IV - A New Hope (a.k.a. Star War~       4.22
## 10     150 24284 Apollo 13 (1995)                                           3.89
## # ... with 10,667 more rows
```

```
## # A tibble: 10,677 x 4
## # Groups:   movieId [10,677]
##    movieId     n title                                               avg_rating
##      <dbl> <int> <chr>                                                    <dbl>
##  1    3191     1 Quarry, The (1998)                                         3.5
##  2    3226     1 Hellhounds on My Trail (1999)                              5
##  3    3234     1 Train Ride to Hollywood (1978)                             3
```

```
##  4    3356     1 Condo Painting (2000)                                    3
##  5    3383     1 Big Fella (1937)                                         3
##  6    3561     1 Stacy's Knights (1982)                                   1
##  7    3583     1 Black Tights (1-2-3-4 ou Les Collants noirs) (1960)      3
##  8    4071     1 Dog Run (1996)                                           1
##  9    4075     1 Monkey's Tale, A (Les Château des singes) (1999)         1
## 10    4820     1 Won't Anybody Listen? (2000)                             2
## # ... with 10,667 more rows
```




```
## # A tibble: 10 x 4
##    movieId     n avg_rating title
##      <dbl> <int>      <dbl> <chr>
##  1    3226     1       5    Hellhounds on My Trail (1999)
##  2    5194     4       4.75 Who's Singin' Over There? (a.k.a. Who Sings Over Th~
##  3   26048     4       4.75 Human Condition II, The (Ningen no joken II) (1959)
##  4   26073     4       4.75 Human Condition III, The (Ningen no joken III) (196~
##  5   33264     2       5    Satan's Tango (Sátántangó) (1994)
##  6   42783     1       5    Shadows of Forgotten Ancestors (1964)
##  7   51209     1       5    Fighting Elegy (Kenka erejii) (1966)
##  8   53355     1       5    Sun Alley (Sonnenallee) (1999)
##  9   64275     1       5    Blue Light, The (Das Blaue Licht) (1932)
## 10   65001     2       4.75 Constantine's Sword (2007)

## # A tibble: 17 x 4
##    movieId     n avg_rating title
```

```
##       <dbl> <int>      <dbl> <chr>
##  1     604     2      1     Criminals (1996)
##  2    2228     2      1     Mountain Eagle, The (1926)
##  3    3561     1      1     Stacy's Knights (1982)
##  4    4071     1      1     Dog Run (1996)
##  5    4075     1      1     Monkey's Tale, A (Les Château des singes) (1999)
##  6    5702     1      1     When Time Ran Out... (a.k.a. The Day the World Ende~
##  7    5805     2      0.5   Besotted (2001)
##  8    6189     1      1     Dischord (2001)
##  9    6483   199      0.902 From Justin to Kelly (2003)
## 10    7282    14      0.821 Hip Hop Witch, Da (2000)
## 11    8394     1      0.5   Hi-Line, The (1999)
## 12    8859    56      0.795 SuperBabies: Baby Geniuses 2 (2004)
## 13   55324     1      1     Relative Strangers (2006)
## 14   61348    32      0.859 Disaster Movie (2008)
## 15   61768     1      0.5   Accused (Anklaget) (2005)
## 16   63828     1      0.5   Confessions of a Superhero (2007)
## 17   64999     2      0.5   War of the Worlds 2: The Next Wave (2008)
```

Therefore, we have to consider the number of ratings in each movie to improve recommendation system. The regularization penalize this effect in the model. Each step includes finding the best lambda, and apply the lambda for regularization.

- control total variability of the movie effect

```
# lambda variables to select best parameter
lambdas <- seq(0, 15, 0.2)
```

Method 1-1. regularized movie: choosing parameter

- Find best lambda in a movie effect

Best lambda for movie = 2.4

Method 1-2. regularized movie

Using the best lambda for movie effect, I've regularized movie effect in the model.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393
genre effect	1.0519422
movie+user+date+genre effect	0.8626515
regularized movie effect	0.9438521

The regularized movie effect slightly improved RMSE (0.9438521) compared to movie effect without regularization (RMSE = 0.9439087).

Method 2-1. regularized movie + user: choosing parameter

Previous result showed that regularized for movie improved RMSE. How about regularization for user?

- Find best lambda in a movie+user effect

Best lambda for movie+user = 5.2

Method 2-2. regularized movie + user

Using the best lambda for movie + user effect, I've regularized for both movie and user effect in the model.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393
genre effect	1.0519422
movie+user+date+genre effect	0.8626515
regularized movie effect	0.9438521
regularized movie+user effect	0.8648170

The regularized movie+user effect slightly improved RMSE (0.864817) compared to movie+user effect without regularization (RMSE = 0.8653488).

Method 3-1. regularized movie + user + date: choosing parameter

Adding the date as another feature in the regularization model.

- Find best lambda in movie + user + date effect

Best lambda for movie+user+date = 5.6

Method 3-2. regularized movie + user + date

Using the best lambda for movie + user + date effect, I've regularized for movie, user, and date effect in the model.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393
genre effect	1.0519422

method	RMSE
movie+user+date+genre effect	0.8626515
regularized movie effect	0.9438521
regularized movie+user effect	0.8648170
regularized movie+user+date effect	0.8642720

The regularized movie+user+date effect slightly improved RMSE (0.864272) compared to movie+user+date effect without regularization (RMSE = 0.8648393).

Method 4-1. regularized movie + user + date + genre: choosing parameter

Here, I regularized for all features including movie, user, date, and genre.

- Find best lambda in movie + user + date + genre effect

Best lambda for movie+user+date+genre = 5.8

Method 4-2. regularized movie + user + date + genres

Now, I added genre effect for the regulariztaion. Using the best lambda for movie + user + date + genre effect, I've regularized for movie, user, date, and genre effect in the model.

method	RMSE
average ratings	1.0612018
movie effect	0.9439087
user effect	0.9947953
movie+user effect	0.8653488
date effect	1.0603797
movie+user+date effect	0.8648393
genre effect	1.0519422
movie+user+date+genre effect	0.8626515
regularized movie effect	0.9438521
regularized movie+user effect	0.8648170
regularized movie+user+date effect	0.8642720
regularized movie+user+date+genre effect	0.8621096

The regularized movie + user + date + genre effect slightly improved RMSE (0.8621096) compared to movie + user + date effect without regularization (RMSE = 0.8626515).

Conclusion

Using 10M movie rating dataset (9M training and test, 1M validation), I've developed movie rating prediction system. Movie feature shows most prediction and genre shows least prediction strength.

The best prediction model is 'regularized movie + user + date + genre effect' with final RMSE = 0.8621096.

method	RMSE
regularized movie+user+date+genre effect	0.8621096
movie+user+date+genre effect	0.8626515
regularized movie+user+date effect	0.8642720
regularized movie+user effect	0.8648170
movie+user+date effect	0.8648393
movie+user effect	0.8653488
regularized movie effect	0.9438521
movie effect	0.9439087
user effect	0.9947953
genre effect	1.0519422
date effect	1.0603797
average ratings	1.0612018