Movielens Capstone Project HarvardX

MFR

06/03/2020

File Name: HarvardX_Capstone1_June04_2020.RMD

Introduction:

The first capstone project of the HarvardX Data Science Professional Certificate Series is a recommendation system on **MovieLens**. A 10M MovieLens dataset was made available from GroupLens Research Lab to make the analysis. To apply the required machine learning algorithms the given movielens dataset was partitioned to create **edx** and **validation** sets. Two different algorithms were used on the **edx** set to predict movie ratings in the unknown **validation** set to find the **RMSE** for accuracy determination of the predictions. **Penalized Least Squares** and **Matrix Factorization with Parallel Stochastic Gradient Descent (SGD)** methods were used to find the RMSEs keeping the genres combined. **RMSEs** were also calculated with genres separated in both methods. The **RMSE** value with genres separated showed about 10% improvement over genres combined in the **MF-SGD** method but no such change happened in the Penalized Least Squares method. Therefore, without proper consideration of genre, optimal RMSE cannot be obtained.

Executive Summary:

Machine Learning methods and tools are required to analyze the dataset for data partitioning and the subsequent analysis. Historically, Machine Learning on movie recommendation systems became popular due to some of the methods used by the winners of the **Netflix** challenges. The Netflix data is not publicly available, but a similar dataset was created by **GroupLens Research Lab** for researchers and data scientists. A 10M dataset from **GroupLens** was downloaded for the capstone project. An initial exploration of the 10M dataset was done with a set of quiz questions provided by edx. Once the dataset information and structure are known the subsequent analyses for finding the residual mean squared error (RMSE) became obvious. A proper choice of model or algorithm will result in the minimization of RMSE, and better predictability. A functional model with regularization does not ensure the reduction of the total variability due to various effects. The idea of regularization is to apply penalized regression to control the total variability of the movie and users effects on RMSE. Specifically, instead of minimizing the least square equation to find RMSE, we minimize an equation that adds a penalty to calculate the RMSE. However, from the analysis and results it was evident that modeling based on movie and user effects can't be pushed below a certain limit. The plot indicates strong evidence of user, movie and genre to push the RMSE limit to its optimal limit. By applying the Matrix Factorization with Parallel Stochastic Gradient Descent (SGD) method an improvement of about 14% over the Penalized Least Squares was achieved. Also, within MF-SGD with genres separated showed about 10% improvement over the genres combined. The results have been presented in the result section.

Preparing R Workspace environment by deleting files filled with data and values

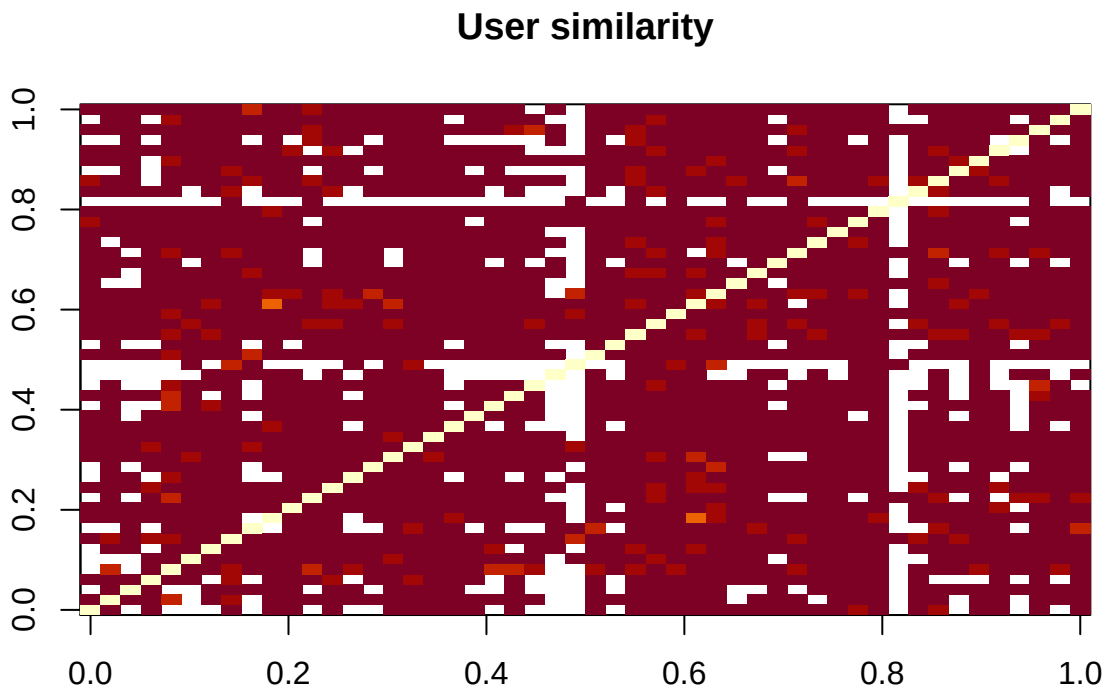
Loading the required R libraries

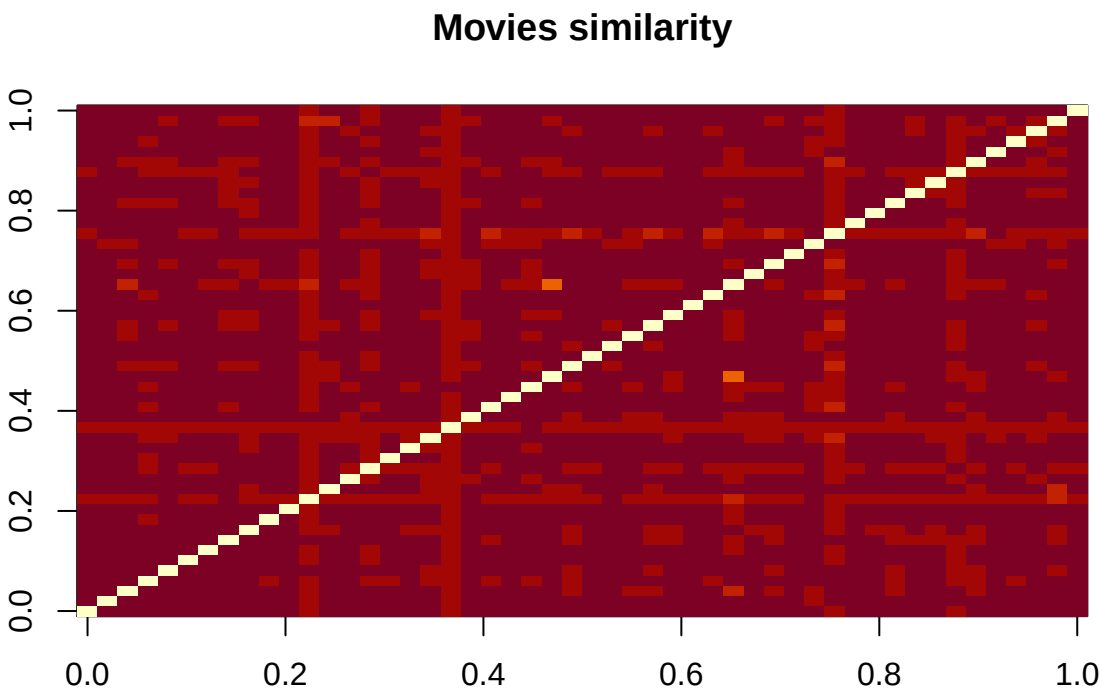
Loading 10M Movielens Dataset and Partioning to Create edx and validation sets

Data Exploration: 1. Cosine Similarity

```
## 10 x 10 sparse Matrix of class "dgCMatrix"
##
## u1  . .   . . . . . . . .
## u2  . .   . . . . . . . .
## u3  . .   . . . . . . . .
## u4  . .   . . . . . . . .
## u5  1 .   . . . . 3 . . .
## u6  . .   . . . . . . . .
## u7  . .   . . . . . . . .
## u8  . 2.5 . . 3 4 . . . .
## u9  . .   . . . . . . . .
## u10 . .   . . . . 3 . . .
```

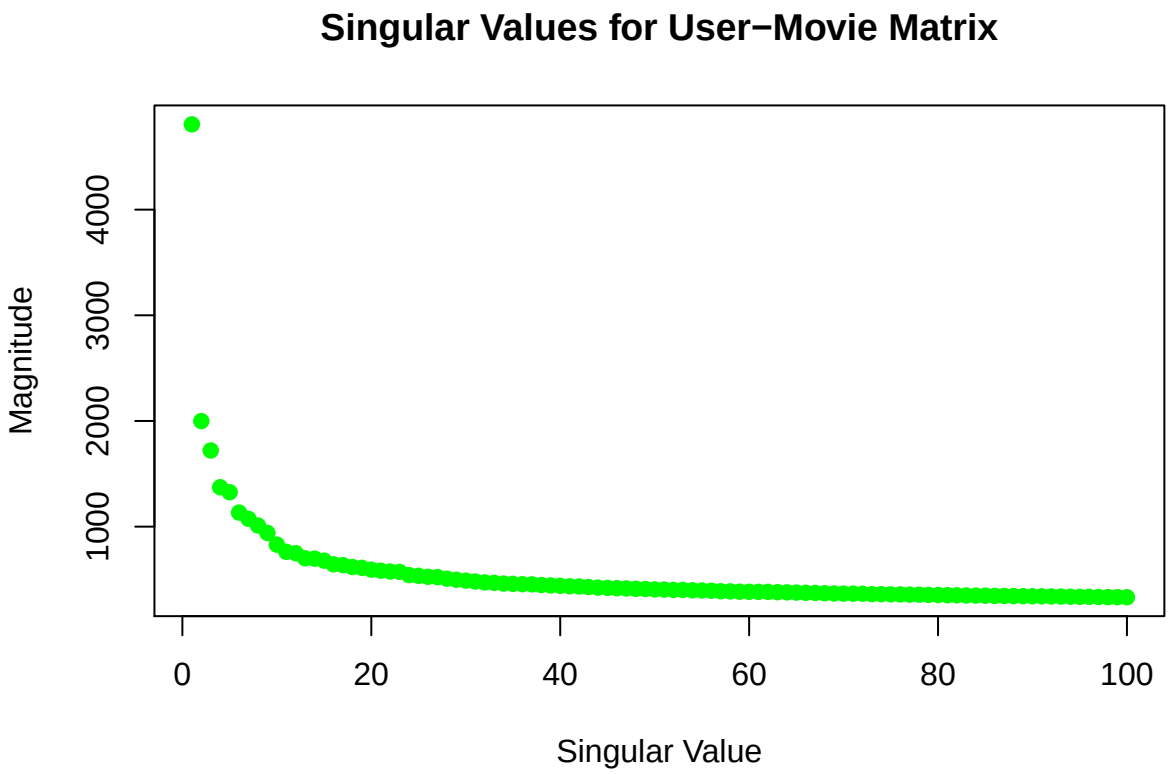
```
## 69878 x 10677 rating matrix of class 'realRatingMatrix' with 9000055 ratings.
```



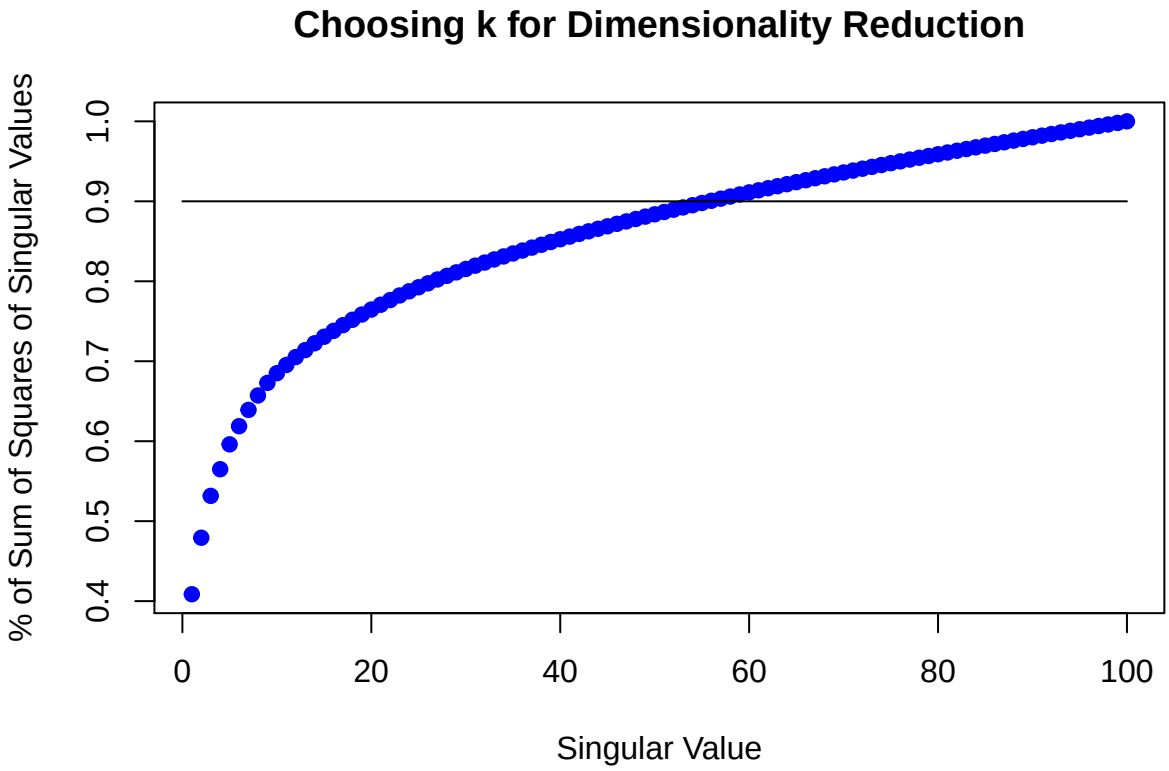


Data Exploration: 2. Dimension Reduction

Dimensionality reduction techniques such as “pca” and “svd” are used to transform the original high-dimensional space into a lower-dimension. Dimension reduction using **Irlba** package, which is a fast and memory-efficient to compute a partial SVD.


Singular Values for User−Movie Matrix
Magnitude
1000
2000
3000
4000
0
20
40
60
80
100
Singular Value

```
## [1] 0.6187623
```




```
## [1] 55
## [1] 69878    55
## [1] 55 55
## [1]    55 10677
```

Method-1: Penalized Least Squares with Regularization

Creating test and train sets with caret package to assess the accuracy of the models. The Netflix challenge used the residual mean squared error (RMSE) metric to find the winner based on the on a test set. The RMSE is then defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}\left(\hat{y}_{u,i} - y_{u,i}\right)^2} \tag{1}$$

where $y_{u,i}$ as the rating for movie i by user u and denote our prediction with $\hat{y_{u,i}}$ and N is being the number of user/movie combinations and the sum occurring over all these combinations.

In general movielens dataset was investigated with methods including but not limited to data cleaning by eliminating unwanted column and data visualization by creating tables and graphs of various formats. As mentioned earlier, two methods namely the Penalized Least Squares and Matrix Factorization with Parallel Stochastic Gradient Descent (SGD) have been applied to calculate the RMSE. Penalized least squares is a balanced data fitting technique that optimizes the variation and come up with a singular solution. A penalized least squares estimate is a surface that minimizes the penalized least squares over the class of all surfaces satisfying sufficient regularity conditions. However, the final result depends on the models that are being used not on the penalized least squares technique itself. So, a better RMSE from the penalized least

squares will depend on the appropriate choices of the models. Specifically, instead of minimizing the least square equation, we minimize an equation that has a penalty term as shown in the following equation [1]:

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i\right)^2+\lambda\sum_i b_i^2 \tag{2}$$

The first term is the least squares and the second term is the penalty that gets larger when many b_i are large. Using calculus we can actually show that the values of b_i that minimize this equation are:

$$\hat{b_i}\left(\lambda\right)=\frac{1}{\lambda+n_i}\sum_{u=1}^{n_i}\left(Y_{u,i}-\hat{\mu}\right) \tag{3}$$

where n_i is the number of ratings made for movie i.

A Linear model with average rating and different BIASES are used as Predictors:

$$\hat{Y}=\mu+b_i+b_u+b_g+b_t$$

$$\mu=Average\ rating\ of\ all\ movies$$

$$b_i=Bias\ based\ on\ Movies$$

$$b_u=Bias\ based\ on\ Users$$

$$b_g=Bias\ based\ on\ Genres$$

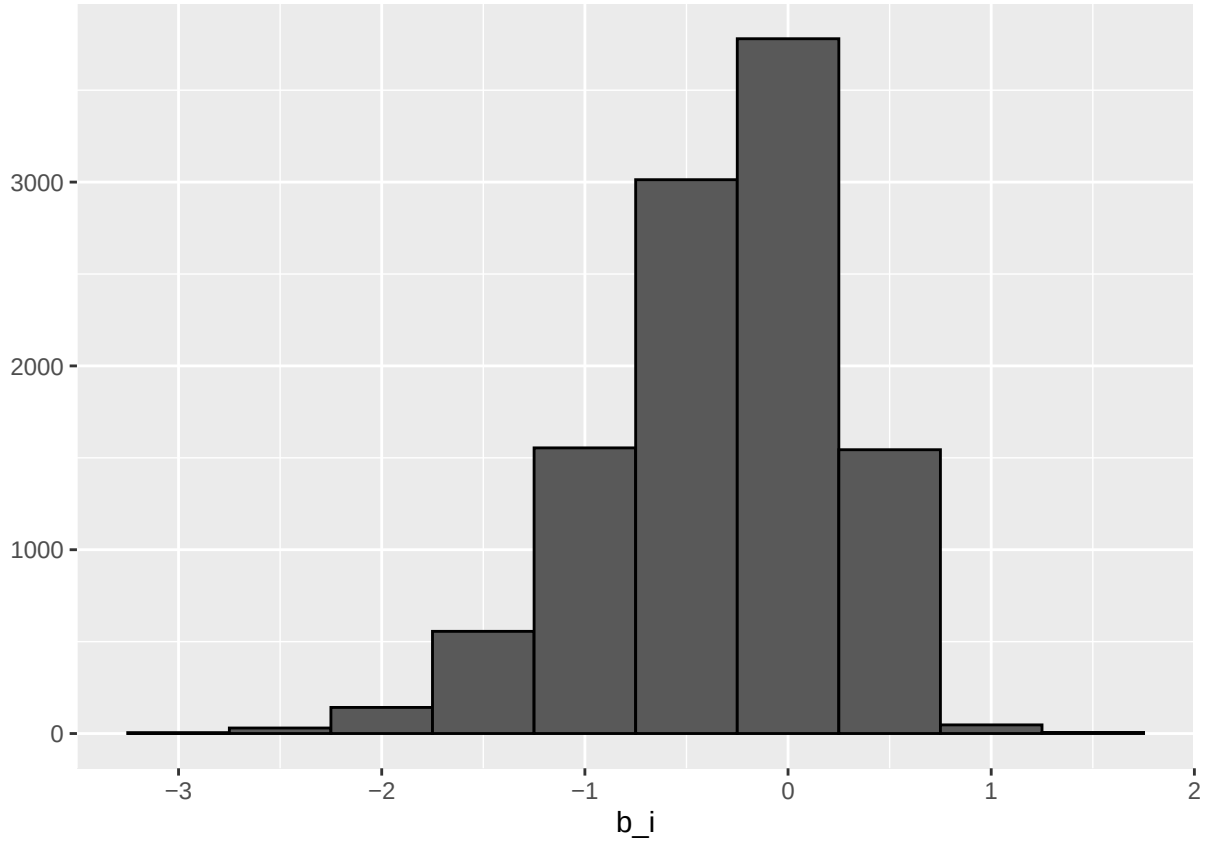
$$b_t=Bias\ based\ on\ Date\ the\ movie\ is\ rated$$

A function that computes the RMSE for vectors of ratings and their corresponding predictors

Average RMSE of Overall Ratings

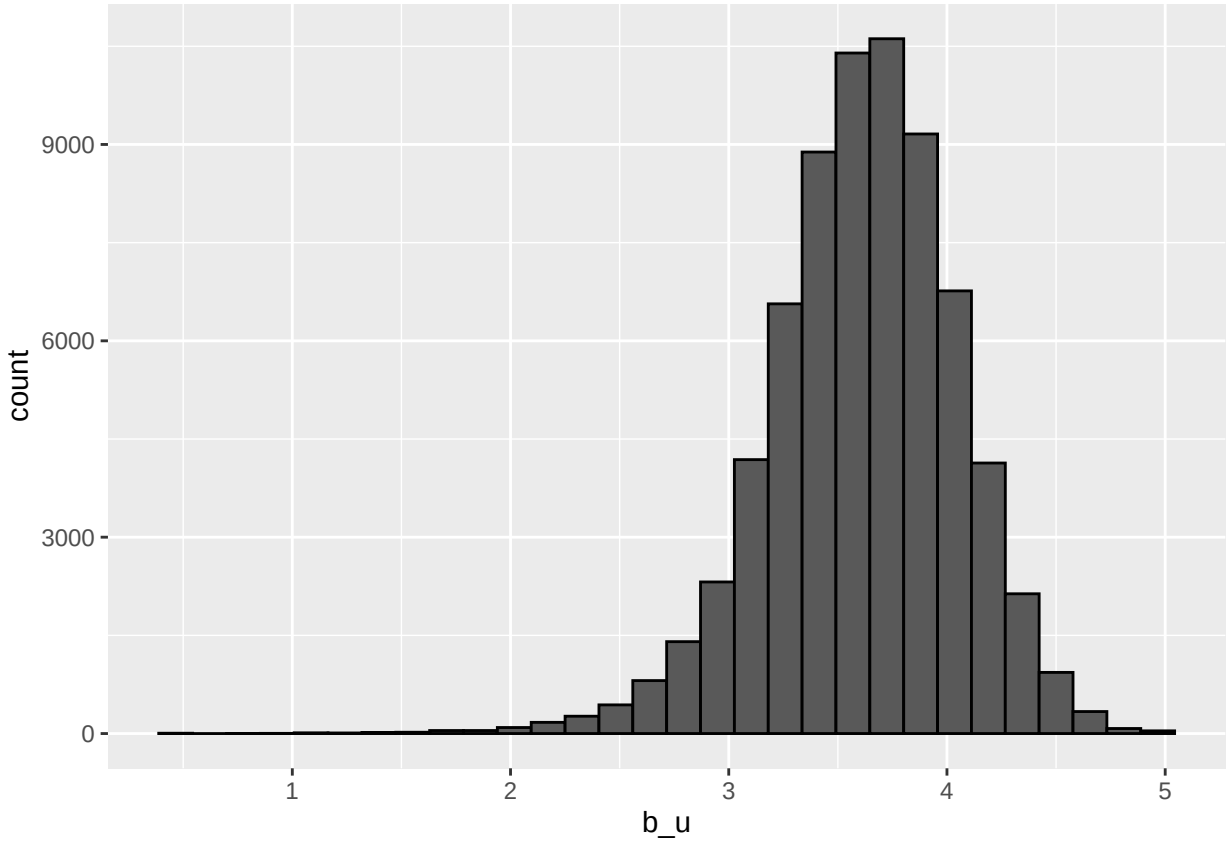
```
## # A tibble: 1 x 2
##   method                            RMSE
##   <chr>                            <dbl>
## 1 Average RMSE of Overall Ratings  1.06
```

RMSE with Movie Effect

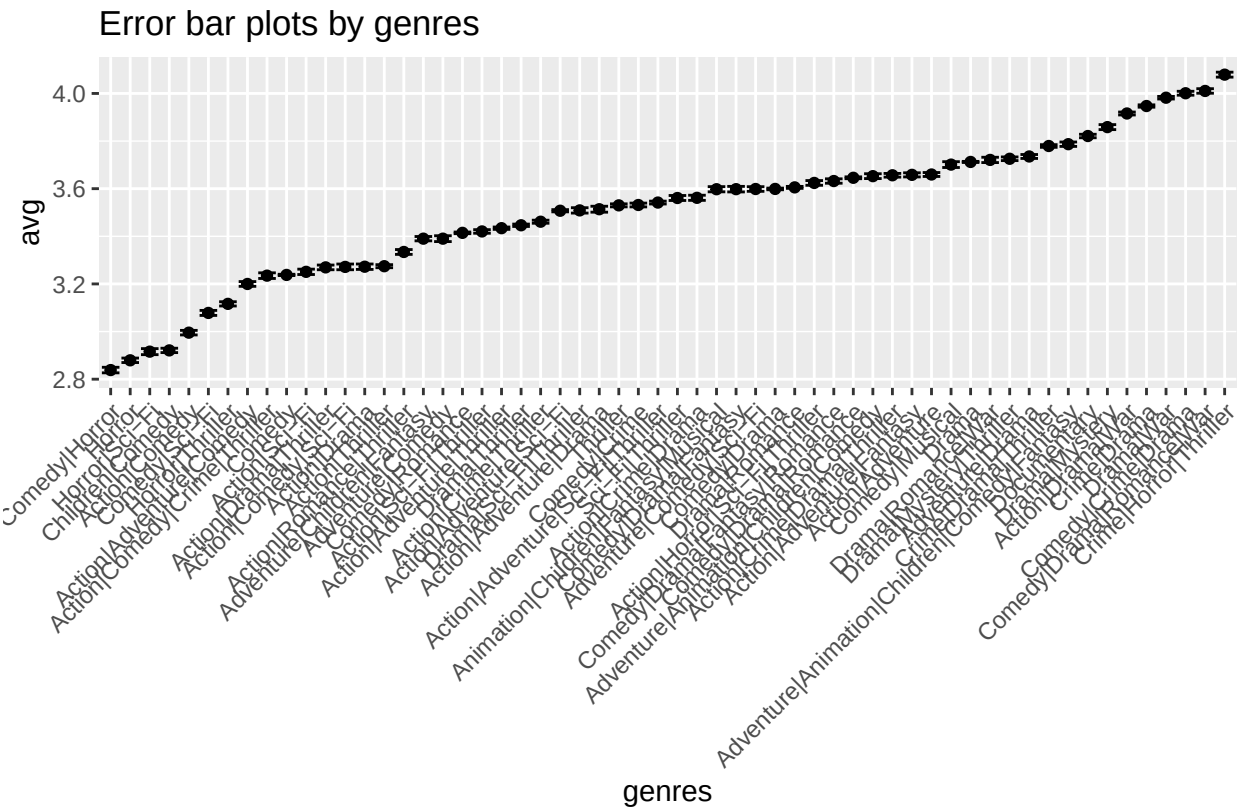



```
## # A tibble: 2 x 2
##   method                              RMSE
##   <chr>                              <dbl>
## 1 Average RMSE of Overall Ratings 1.06
## 2 RMSE with Movie Effect          0.944
```

RMSE with Movie + User Effects



RMSE with Movie + User + Genres Effects



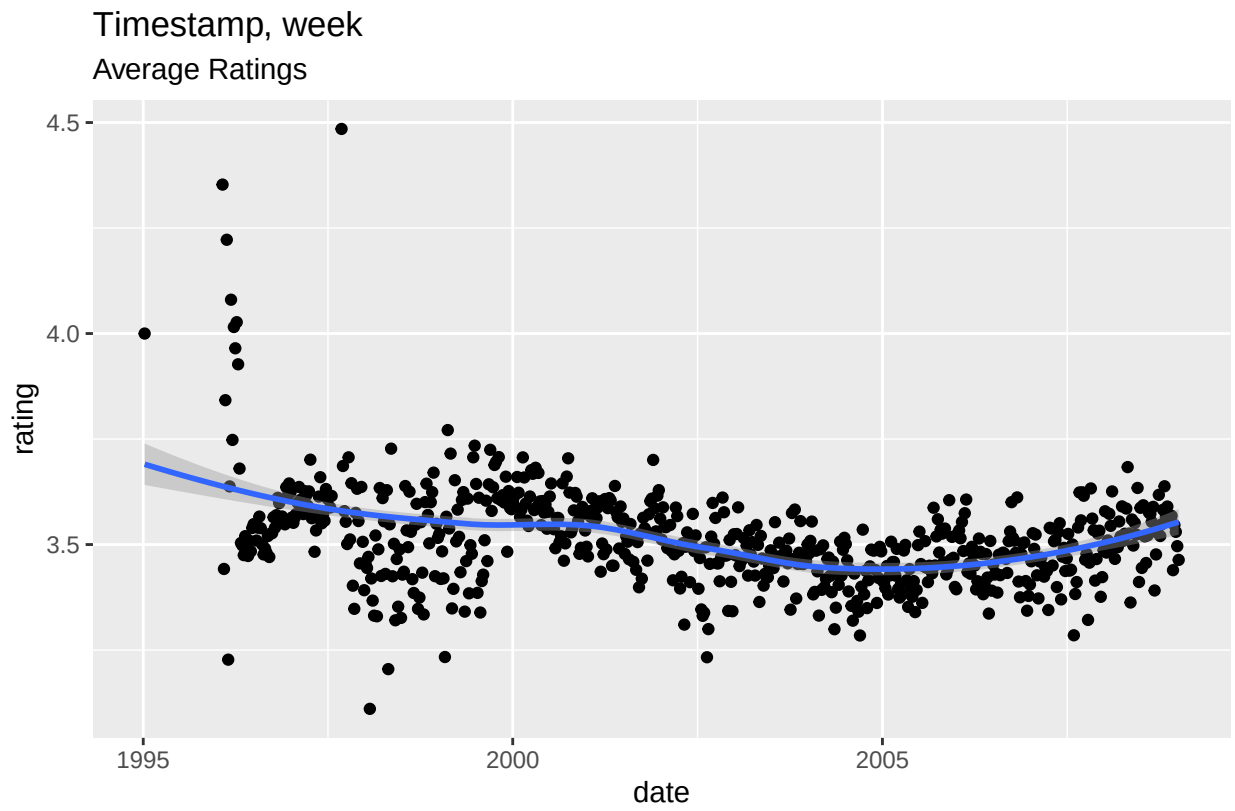
Genre analysis on edx dataset

```
## # A tibble: 4 x 2
##   method                                 RMSE
##   <chr>                                 <dbl>
## 1 Average RMSE of Overall Ratings       1.06
## 2 RMSE with Movie Effect                0.944
## 3 RMSE with Movie + User Effects        0.865
## 4 RMSE with Movie + User + Genres Effects 0.865
```

RMSE with Movie + User + Genres + Year Effect

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
##   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                date year_rated title_year
## 1                Comedy|Romance 1996-08-02 11:24:06       1996       1992
## 2         Action|Crime|Thriller 1996-08-02 10:58:45       1996       1995
## 3  Action|Drama|Sci-Fi|Thriller 1996-08-02 10:57:01       1996       1995
## 4       Action|Adventure|Sci-Fi 1996-08-02 10:56:32       1996       1994
## 5 Action|Adventure|Drama|Sci-Fi 1996-08-02 10:56:32       1996       1994
## 6       Children|Comedy|Fantasy 1996-08-02 11:14:34       1996       1994
```

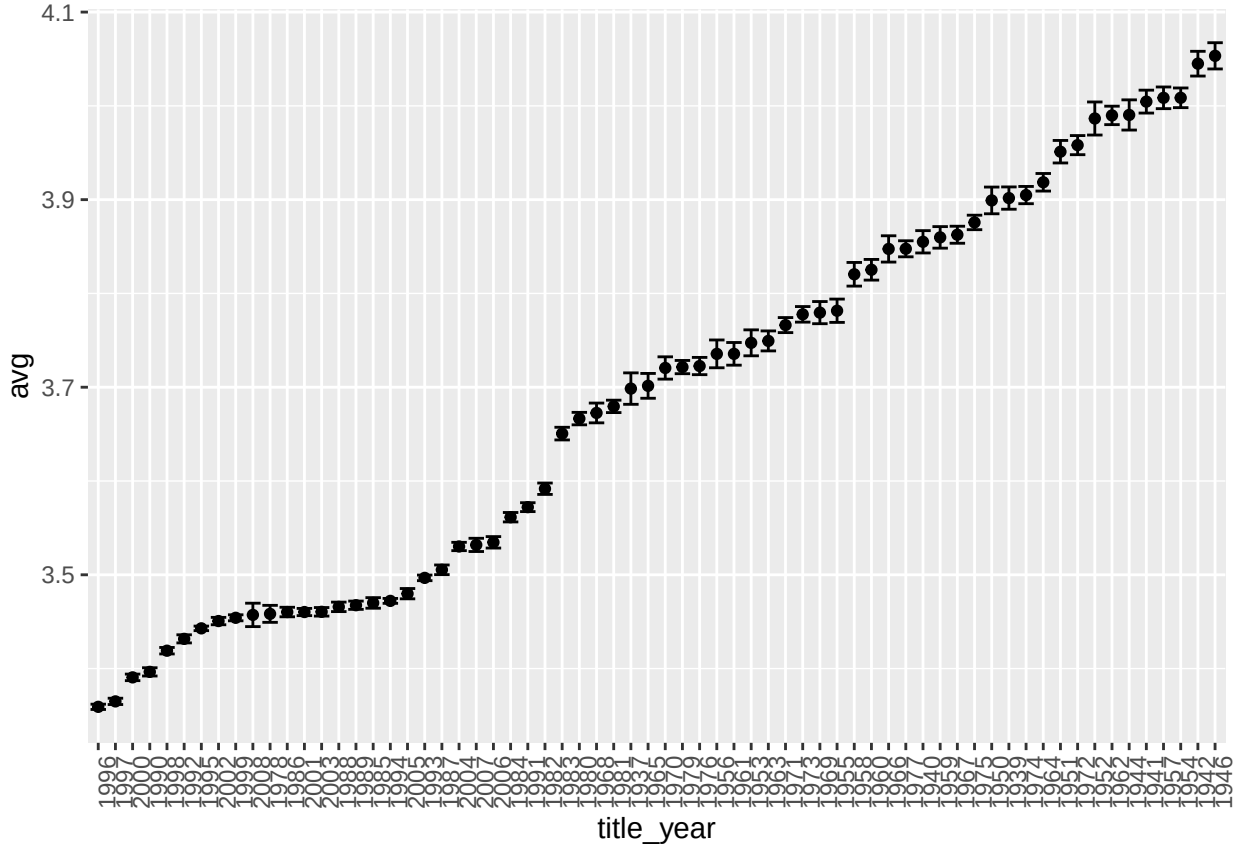
```
##   userId movieId rating                         title
## 1      1     122      5              Boomerang (1992)
## 2      1     185      5               Net, The (1995)
## 3      1     292      5               Outbreak (1995)
## 4      1     316      5               Stargate (1994)
## 5      1     329      5 Star Trek: Generations (1994)
## 6      1     355      5       Flintstones, The (1994)
##                         genres                date year_rated title_year
## 1                Comedy|Romance 1996-08-02 11:24:06       1996       1992
## 2         Action|Crime|Thriller 1996-08-02 10:58:45       1996       1995
```

```
## 3  Action|Drama|Sci-Fi|Thriller 1996-08-02 10:57:01       1996       1995
## 4       Action|Adventure|Sci-Fi 1996-08-02 10:56:32       1996       1994
## 5 Action|Adventure|Drama|Sci-Fi 1996-08-02 10:56:32       1996       1994
## 6       Children|Comedy|Fantasy 1996-08-02 11:14:34       1996       1994
```

```
##   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                date year_rated
## 1                                 Comedy 1996-08-02 10:56:32       1996
## 2        Action|Adventure|Sci-Fi|Thriller 1996-08-02 11:00:53       1996
## 3                      Children|Comedy 1996-08-02 11:07:48       1996
## 4               Action|Drama|Romance|War 1997-07-07 03:34:10       1997
## 5                           Crime|Drama 1997-07-07 03:20:45       1997
## 6 Action|Adventure|Horror|Sci-Fi|Thriller 1997-07-07 03:25:20       1997
```

```
##   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                date year_rated
## 1                                 Comedy 1996-08-02 10:56:32       1996
## 2        Action|Adventure|Sci-Fi|Thriller 1996-08-02 11:00:53       1996
## 3                      Children|Comedy 1996-08-02 11:07:48       1996
## 4               Action|Drama|Romance|War 1997-07-07 03:34:10       1997
## 5                           Crime|Drama 1997-07-07 03:20:45       1997
## 6 Action|Adventure|Horror|Sci-Fi|Thriller 1997-07-07 03:25:20       1997
##   title_year
## 1       1994
## 2       1993
## 3       1990
## 4       1995
## 5       1972
## 6       1997
```

```
##   userId movieId rating                                                 title
## 1      1     231      5                                      Dumb & Dumber (1994)
## 2      1     480      5                                      Jurassic Park (1993)
## 3      1     586      5                                         Home Alone (1990)
## 4      2     151      3                                           Rob Roy (1995)
## 5      2     858      2                                      Godfather, The (1972)
## 6      2    1544      3 Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                    genres                date year_rated
## 1                                  Comedy 1996-08-02 10:56:32       1996
## 2         Action|Adventure|Sci-Fi|Thriller 1996-08-02 11:00:53       1996
## 3                         Children|Comedy 1996-08-02 11:07:48       1996
## 4                  Action|Drama|Romance|War 1997-07-07 03:34:10       1997
## 5                              Crime|Drama 1997-07-07 03:20:45       1997
## 6 Action|Adventure|Horror|Sci-Fi|Thriller 1997-07-07 03:25:20       1997
##   title_year
## 1       1994
## 2       1993
## 3       1990
## 4       1995
## 5       1972
## 6       1997
```




```
## # A tibble: 5 x 2
##   method                                RMSE
##   <chr>                                <dbl>
## 1 Average RMSE of Overall Ratings       1.06
## 2 RMSE with Movie Effect                0.944
```

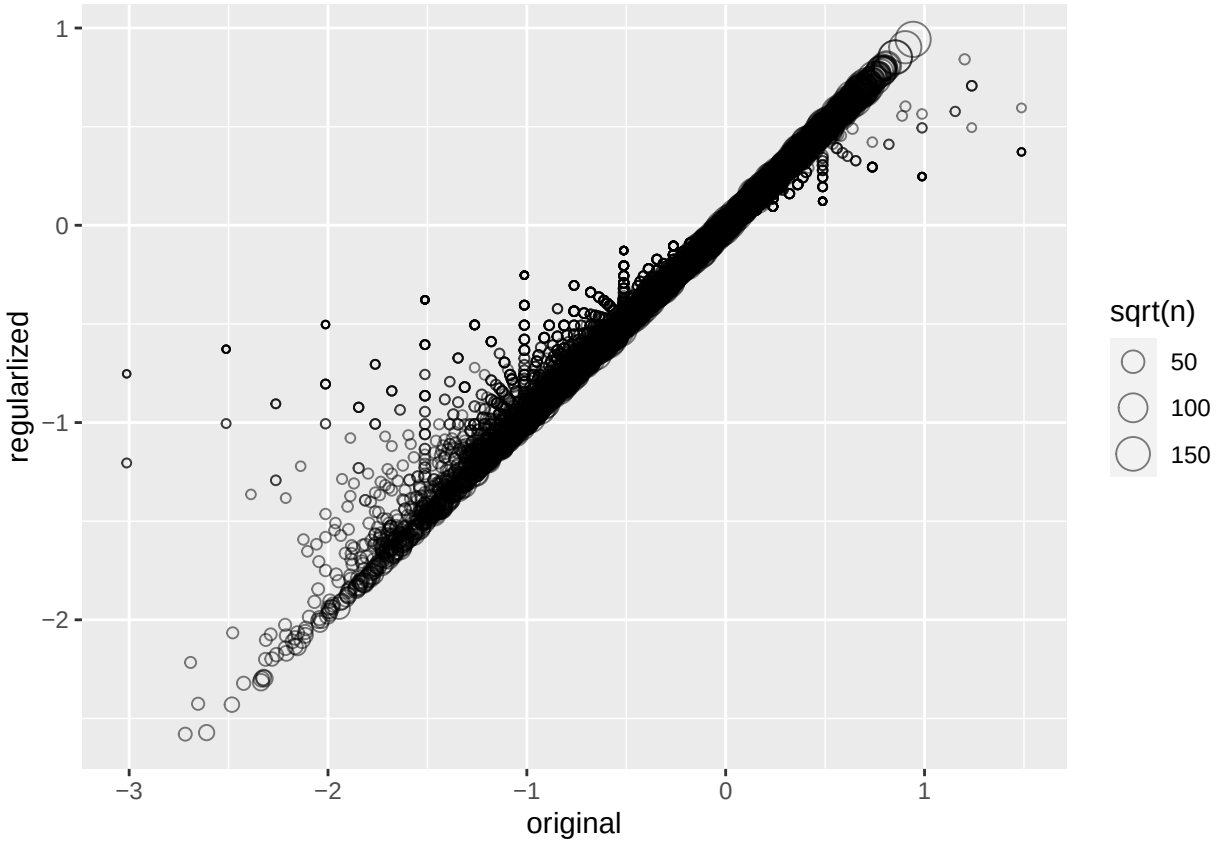
```
## 3 RMSE with Movie + User Effects                0.865
## 4 RMSE with Movie + User + Genres Effects       0.865
## 5 RMSE with Movie + User + Genres + Year Effect 0.865
```

RMSE with Regularized Movie Effect

```
##                               title  residual
## 1            PokÃ©mon Heroes (2003)  3.970803
## 2  Shawshank Redemption, The (1994) -3.955131
## 3  Shawshank Redemption, The (1994) -3.955131
## 4  Shawshank Redemption, The (1994) -3.955131
## 5             Godfather, The (1972) -3.915366
## 6             Godfather, The (1972) -3.915366
## 7             Godfather, The (1972) -3.915366
## 8        Usual Suspects, The (1995) -3.865854
## 9        Usual Suspects, The (1995) -3.865854
## 10       Usual Suspects, The (1995) -3.865854
```

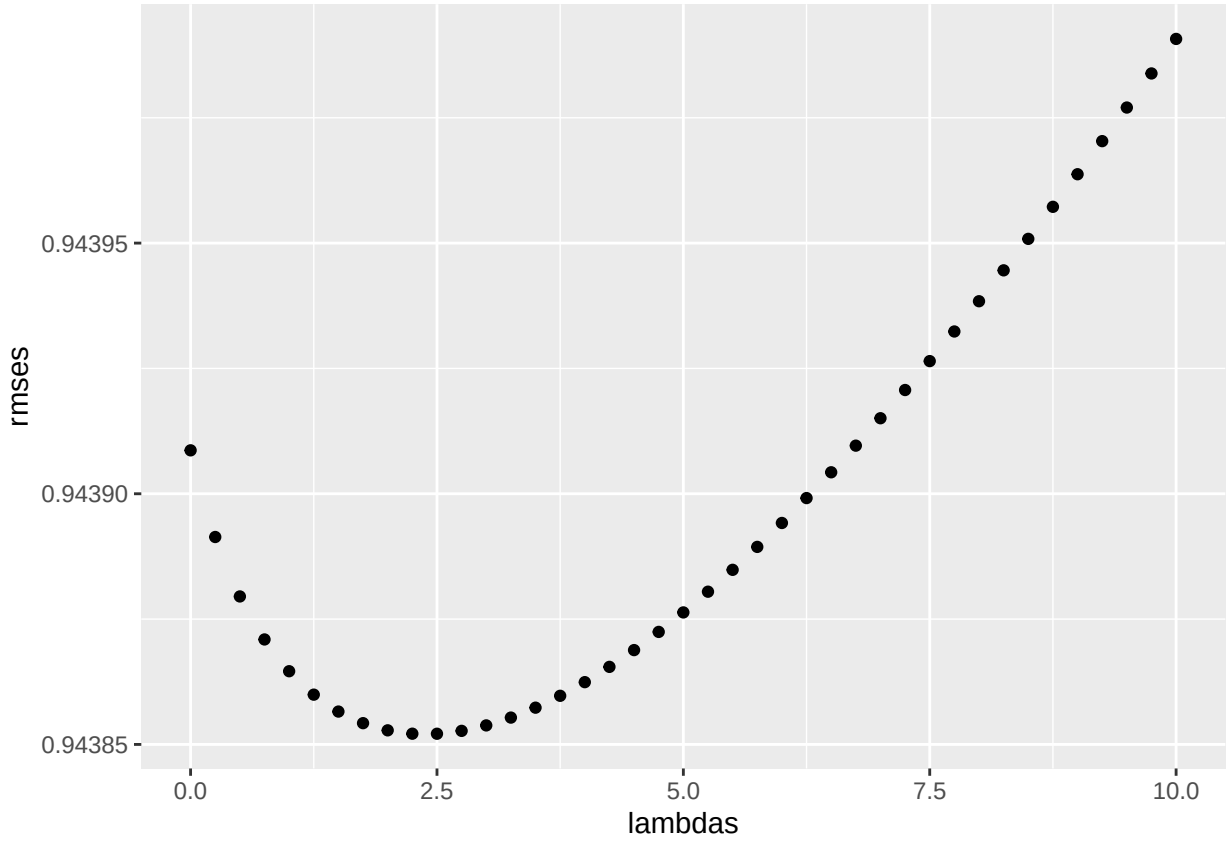
```
## # A tibble: 10 x 2
##    title                                                                  b_i
##    <chr>                                                                <dbl>
##  1 Hellhounds on My Trail (1999)                                         1.49
##  2 Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)                                   1.49
##  3 Shadows of Forgotten Ancestors (1964)                                 1.49
##  4 Fighting Elegy (Kenka erejii) (1966)                                  1.49
##  5 Sun Alley (Sonnenallee) (1999)                                        1.49
##  6 Blue Light, The (Das Blaue Licht) (1932)                              1.49
##  7 Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo pe~  1.24
##  8 Human Condition II, The (Ningen no joken II) (1959)                   1.24
##  9 Human Condition III, The (Ningen no joken III) (1961)                 1.24
## 10 Constantine's Sword (2007)                                            1.24
```

```
## # A tibble: 10 x 2
##    title                                    b_i
##    <chr>                                  <dbl>
##  1 Besotted (2001)                        -3.01
##  2 Hi-Line, The (1999)                    -3.01
##  3 Accused (Anklaget) (2005)              -3.01
##  4 Confessions of a Superhero (2007)      -3.01
##  5 War of the Worlds 2: The Next Wave (2008) -3.01
##  6 SuperBabies: Baby Geniuses 2 (2004)    -2.72
##  7 Hip Hop Witch, Da (2000)               -2.69
##  8 Disaster Movie (2008)                  -2.65
##  9 From Justin to Kelly (2003)            -2.61
## 10 Criminals (1996)                       -2.51
```

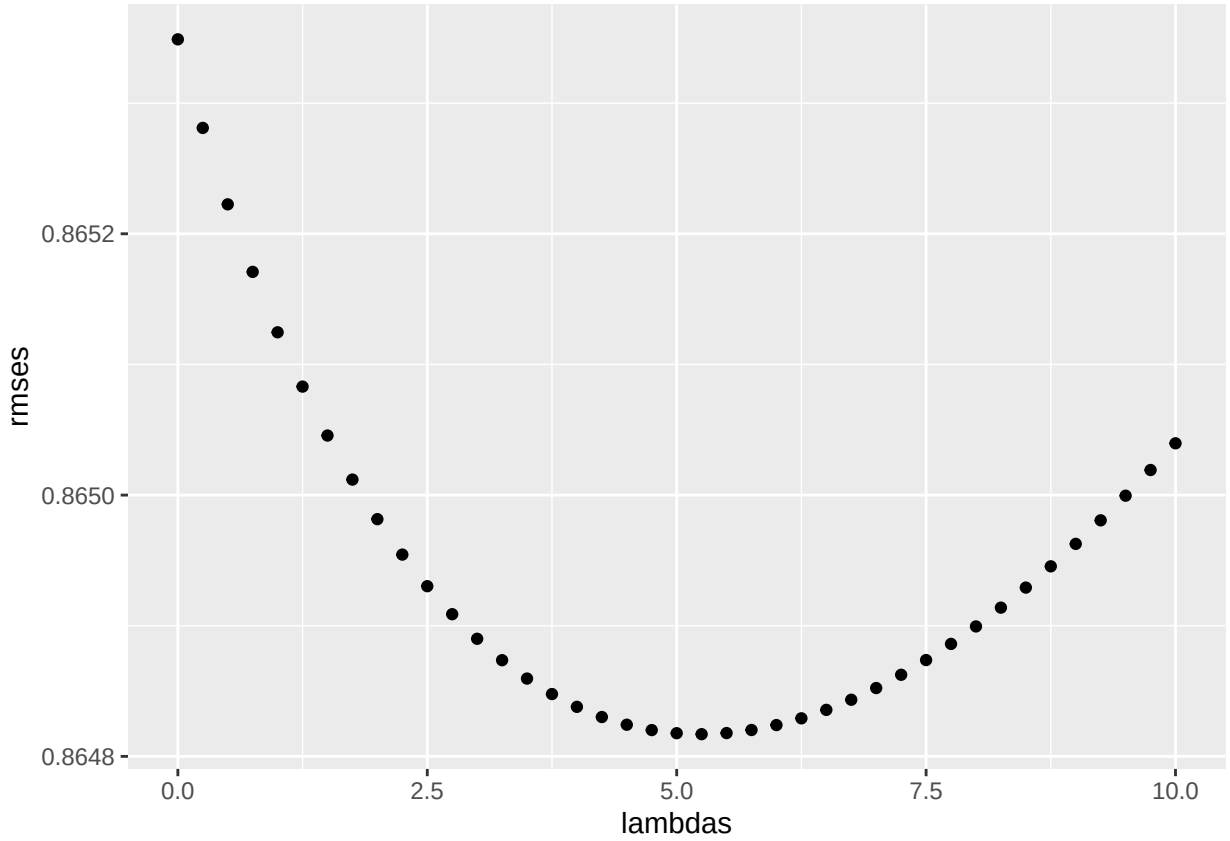



```
## # A tibble: 6 x 2
##   method                                          RMSE
##   <chr>                                          <dbl>
## 1 Average RMSE of Overall Ratings                1.06
## 2 RMSE with Movie Effect                         0.944
## 3 RMSE with Movie + User Effects                 0.865
## 4 RMSE with Movie + User + Genres Effects        0.865
## 5 RMSE with Movie + User + Genres + Year Effect 0.865
## 6 RMSE with Regularized Movie Effect             0.944
```

RMSE with Regularized Movie + User Effect




```
## [1] 2.5
```




```
## [1] 5.25
```

```
## # A tibble: 7 x 2
##   method                                    RMSE
##   <chr>                                    <dbl>
## 1 Average RMSE of Overall Ratings          1.06
## 2 RMSE with Movie Effect                   0.944
## 3 RMSE with Movie + User Effects           0.865
## 4 RMSE with Movie + User + Genres Effects  0.865
## 5 RMSE with Movie + User + Genres + Year Effect 0.865
## 6 RMSE with Regularized Movie Effect       0.944
## 7 RMSE with Regularized Movie + User Effect 0.865
```

Method-2: Matrix Factorization with SGD

Matrix Factorization with Parallel Stochastic Gradient Descent (SGD) specifically that applies to Recommender System was used to verify and compare the RMSE with the penalized least squares technique [2]. Matrix Factorization with SGD has a recosystem package [3], and is a R wrapper of the LIBMF library developed by Yu-Chin Juan, Wei-Sheng Chin, Yong Zhuang, Bo-Wen Yuan, Meng-Yuan Yang, and Chih-Jen Lin [4]. LIBMF library is an open source library for recommender system using parallel matrix factorization. It is also a high-performance C++ library for large scale matrix factorization. LIBMF itself is a parallelized library, meaning that users can take advantage of multicore CPUs to speed up the computation. It also utilizes some advanced CPU features to further improve the performance.

Since recosystem is a wrapper of LIBMF, it inherits most of the features of LIBMF, and additionally provides a number of user-friendly R functions to simplify data processing and model building. Also, unlike most other R packages for statistical modeling that store the whole dataset and model object in memory, LIBMF

(and hence recosystem) can significantly reduce memory use, for instance the constructed model that contains information for prediction can be stored in the hard disk, and output result can also be directly written into a file rather than be kept in memory. The main task of the recommender system is to predict unknown entries in the rating matrix based on observed values. Matrix factorization is known to be effective in recommender systems. As the input data set is often large, an MF solution is time-consuming. There are mainly three algorithms to solve MF namely coordinate gradient descent(CGD), alternate least square(ALS), and stochastic gradient descent(SGD).

In a recommendation system similar to Netflix or MovieLens dataset, there is a matrix containing users and a set of movies. Given that each users have rated some movies in the system, we would like to predict how the users would rate the movies that they have not yet rated, such that we can make recommendations to the users. As an example we can create a matrix with information about the existing ratings as shown in the table below. Assuming we have 5 users and 5 movies, and ratings are integers ranging from 0.5 to 5, with the "0" indicating that the user has not yet rated the movie.

```
   M1  M2  M3  M4  M5
U1 1   X   3   X   5
U2 4.5 2   5   1.5 3
U3 2   5   2.5 X   1
U4 3.5 2   X   5   2
U5 4   1.5 4   2.5 4
```

Therefore, the idea of predicting the missing ratings as indicated with "0" in the matrix would be consistent with the existing ratings in the matrix.

The intuition with matrix factorization is to find the missing elements based on some hidden features such as title, actors, genres and other similar features that both the users prefer during rating a movie.

The task of finding these hidden features can help us to predict a rating with respect to a certain user and a certain movie, because the features associated with the user should match with the features associated with the movie.

Of course the underlying assumption is that the number of features would be smaller than the number of users and the number of movies to be rated. This assumtion is very much acceptable from a practical poiint of view since it is not reasonable to assume that each user is associated with a unique feature. In the given dataset we discovered that no_users = 69878, no_movies = 10677, no_titles = 10677, no_genres = 797, and no_ratings = 10. Actually, number of individual genres is only 20 but with various combinations it comes out to be 797 which is still a much smaller number than the no_users = 69878 and no_movies = 10677. Based on what we discussed we can then design a matrix factorization method by assuming a set U of users, and a set of M movies. Let R of size $|U|\times|M|$ be the matrix that contains all the ratings that the users have assigned to the movies. Also, we assume that we would like to discover K latent features. Matrix Factorization Method consists of two matrices P (of size $|U|\times|K|$) and Q (of size $|M|\times|K|$) such that their product apprioximates |R| and is given in the equation below [5]:

$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q^T} = \hat{\mathbf{R}} \tag{4}$$

Calculation Showing How R and $\hat{R}$ are Related to Matrices P and Q

```
## R
```

```
## [[1.  0.  3.  0.  5. ]
##  [4.5 2.  5.  1.5 3. ]
##  [2.  5.  2.5 0.  1. ]
##  [3.5 2.  0.  5.  2. ]
##  [4.  1.5 4.  2.5 4. ]]
```

```
## Iteration: 10 ; error = 1.9443
## Iteration: 20 ; error = 1.3994
```

```
## Iteration: 30 ; error = 1.1906
## Iteration: 40 ; error = 1.0368
## Iteration: 50 ; error = 0.9605
```

```
## P
```

```
## [[ 1.59327756  1.50348412]
##  [-0.37958127 -0.86088548]
##  [-0.95962186  2.21051695]
##  [-0.31556435  0.34326282]
##  [ 0.19400413 -0.66754127]]
```

```
## Q
```

```
## [[-0.78305304 -1.09372942]
##  [-1.35093263  0.59953787]
##  [-0.12017559 -0.83268086]
##  [ 0.78614736  2.86205348]
##  [ 1.39423204 -0.5161905 ]]
```

```
## R_hat = P x Q
```

```
## [[ 1.02380658  1.18759298  2.90459548 10.25574658  5.03371315]
##  [ 4.71299659  1.99361954  4.66880657  1.49619287  3.06191159]
##  [ 2.14587305  4.95659447  2.51897239 10.16867064  1.00572879]
##  [ 3.10086228  2.38407626  3.41344882  4.74786319  2.28460288]
##  [ 4.03834907  1.32062801  4.4248481   2.48644352  3.74778681]]
```

```
## Global bias:
```

```
## 3.0476190476190474
```

```
## User bias:
```

```
## [ 0.54360834  0.10196464  0.43992176 -0.1430286   0.08793117]
```

```
## Item bias:
```

```
## [ 0.32460485 -1.15261943  0.75676361  1.10891628 -0.00282864]
```

In the above equation each row of P represents the strength of the associations between a user and the features and each row of Q represents the strength of the associations between a movie and the features. To get the prediction of a rating of a movie m_j by u_i, we can calculate the dot product of their vectors as given below:

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{K} p_{ik} q_{kj} \tag{5}$$

We need to initialize two matrices P and Q with some values and then to calculate how different their product is from actual ratings matrix, R, by minimizing the difference iteratively. To further tune the result we will apply regularization to avoid overfitting.

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 \tag{6}$$

To minimize the error, we differentiate the above equation with respect ot p_{ik} and q_{kj}.

$$\frac{\partial e_{ij}^2}{\partial p_{ik}} = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij} q_{kj} \tag{7}$$

$$\frac{\partial e_{ij}^2}{\partial q_{ik}} = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik} \tag{8}$$

The updated values of both p_{ik} and q_{kj} are then calculated by the following equations:

$$p'_{ik} = p_{ik} + \alpha\frac{\partial e_{ij}^2}{\partial p_{ik}} = p_{ik} + 2\alpha e_{ij}q_{kj} \tag{9}$$

$$q'_{ik} = q_{kj} + \alpha\frac{\partial e_{ij}^2}{\partial q_{kj}} = q_{kj} + 2\alpha e_{ij}p_{ik} \tag{10}$$

In the above equations α is a constant whose value determines the rate of approaching the minimum. Normally, α is considered small so that we don't skip the minimum and end up oscillating around the minimum.

We can check the overall error as calculated using the following equation and determine when we should stop the process. We are not really trying to come up with P and Q such that we can reproduce R exactly. Instead, we will only try to minimize the errors of the observed user-movie pairs. Considering T be a set of tuples, each of which is in the form of (u_i, d_j, r_{ij}), such that T contains all the observed user-movie pairs together with the associated ratings, we are only trying to minimise every e_{ij} for $(u_i, d_j, r_{ij}) \in T$.

T is our training dataset. As for the rest of the unknowns, we will be able to determine their values once the associations between the users, movies and features have been learnt.

Using the above update rules, we can then iteratively perform the operation until the error converges to its minimum. We can check the overall error as calculated using the following equation and determine when we should stop the process.

$$e = \sum_{(u_i,d_j,r_{ij})\in T} e_{ij} = \sum_{(u_i,d_j,r_{ij})\in T} (r_{ij} - \sum_{k=1}^{K} p_{ik}q_{kj})^2 \tag{11}$$

A common extension to this basic algorithm is to introduce regularization to avoid overfitting. This is done by adding a parameter and modify the squared error as follows:

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik}q_{kj})^2 + \frac{\beta}{2}\sum_{k=1}^{K}(||P||^2 - ||Q||^2) \tag{12}$$

In other words, the new parameter β is used to control the magnitudes of the user-feature and movie-feature vectors such that P and Q would give a good approximation of R without having to contain large numbers. In practice, β is set to some values in the order of 0.02. The new update rules for this squared error can be obtained by a procedure similar to the one described above. The new update equaions are as follows:

$$p'_{ik} = p_{ik} + \alpha\frac{\partial e_{ij}^2}{\partial p_{ik}} = p_{ik} + \alpha(2e_{ij}q_{jk} - \beta p_{ik}) \tag{13}$$

$$q'_{ik} = q_{kj} + \alpha\frac{\partial e_{ij}^2}{\partial q_{kj}} = q_{kj} + \alpha(2e_{ij}p_{ik} - \beta q_{kj}) \tag{14}$$

A rating is being generated, with some biases to the ratings. Hence, we can also add biases in order to better model how a rating is being generated using the following model:

$$\hat{r_{ij}} = b + bu_i + bd_j + \sum_{k=1}^{K} p_{ik} q_{kj} \tag{15}$$

where b is the global bias (which can be easily estimated by using the mean of all ratings), bu_i is the bias of $user_i$, and bd_j is te bias of movie j.we can derive the update rules for the user biases and movie biases easily:

$$bu_i' = bu_i + \alpha(e_{ij} - \beta bu_i) \tag{16}$$

$$bd_j' = bd_j + \alpha(e_{ij} - \beta bd_j) \tag{17}$$

In practice, the process of factorization will converge faster if biases are included in the model.

RMSE using Matrix Factorization with Stochastic Gradient Descent (SGD)

The usage of 'recosystem' package is really straightforward. Here the training_dataset variable represents a dataframe with userid, movieid, rating columns. Parameters are set arbitrarily: the number of factors (dim) is 30, regularization for P and Q factors (costp_l2, costq_l2) is set to 0.001, and convergence can be controlled by a number of iterations (niter = 10) and learning rate (lrate = 0.1). The user can also control the parallelization using the nthread = 6 parameter:

Mean squared error (abbreviated MSE) and root mean square error (RMSE) refer to the amount by which the values predicted by an estimator differ from the quantities being estimated (typically outside the sample from which the model was estimated).

We calculate the standard deviation of the residuals (prediction errors) RMSE. Between the predicted ratings and the real ratings . If one or more predictors are significant, the second step is to assess how well the model fits the data by inspecting the Residuals Standard Error (RSE).

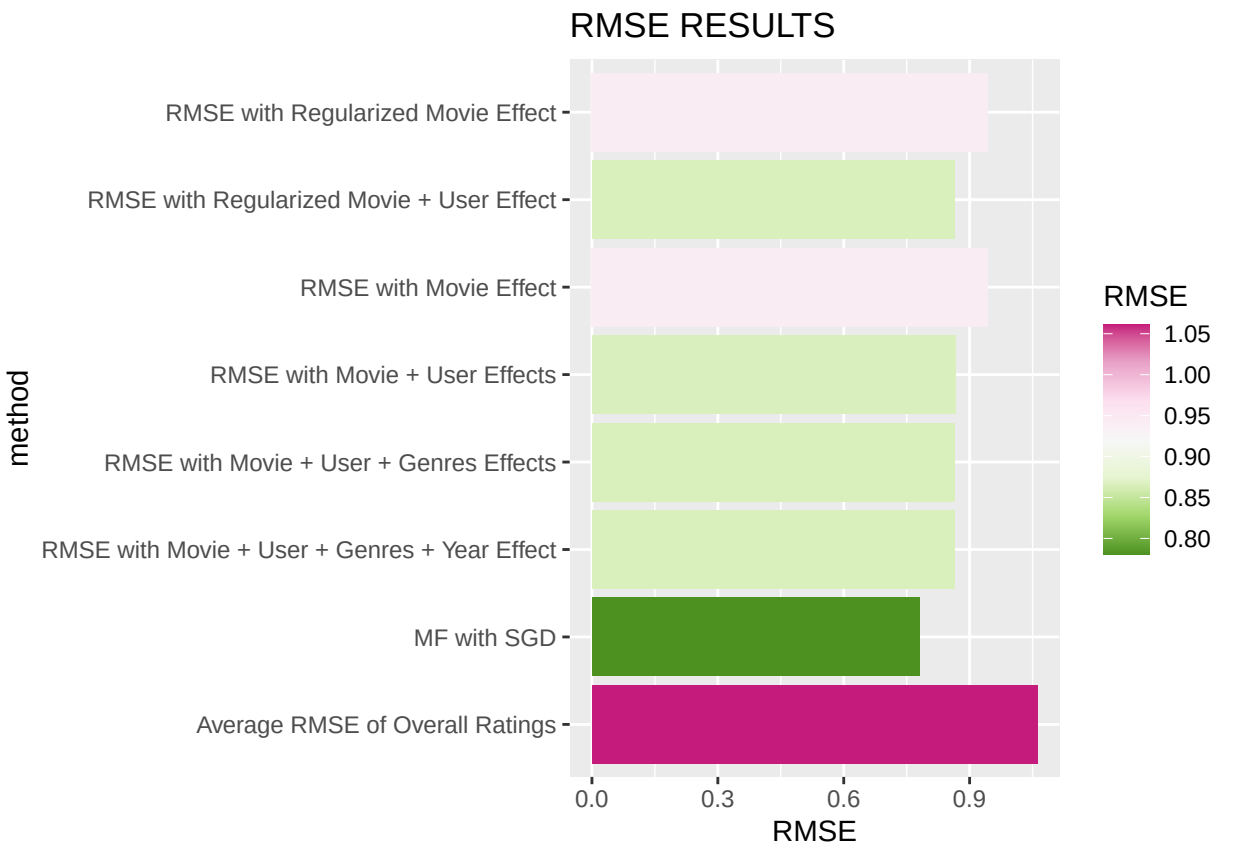
```
## [1] 9000055       3
```

```
## [1] 999999      3
```

```
## prediction output generated at C:\Users\MfR\AppData\Local\Temp\Rtmpu8WWpK\file1f4cf025090
```

```
## [1] 999999      1
```

========== ========== method RMSE ========== ========== Average RMSE of Overall Ratings 1.0612018 RMSE with Movie Effect 0.9439087 RMSE with Movie + User Effects 0.8653488 RMSE with Movie + User + Genres Effects 0.8649469 RMSE with Movie + User + Genres + Year Effect 0.8649469 RMSE with Regularized Movie Effect 0.9438538 RMSE with Regularized Movie + User Effect 0.8648170 MF with SGD 0.7807480 ========== ==========



Results & Discussions

In analyzing the movielens dataset I observed that RMSE limit can be pushed down to an optimal limit with a model involving movie, user and genres. A good model must need to consider the effect of genre for reaching the optimal RMSE. However, one immediate problem I encountered with my Dell Laptop (about 9 years old, Intel i7 processor with 2.00 GHz clock speed, recently being updated to with 16 GB RAM) is that it cannot handle training and test datasets involving movieId, userId, rating and genres simultaneously. Even with movieId, userId and rating, computation time is more than an hour with Matrix Factorization with Parallel Stochastic Gradient Descent (SGD) method. In order to observe the effect of genre on movie rating I separated the genres. In this regard first I extract the title year and mutate it as a separate column from the title itself, then in the second step I dropped the timestamp and in the third step I separated the genres and replaced the original column with genres separated. The modified edx dataset with genres separated will create matrix which is very large, about 2.6 times bigger than the edx dataset. This is because user and movie are related to individual genres in many more combinations than the genres combined. These matrices having dimensions with (25967194,3) and (999999,3) for training set and validation set respectively. Considering these newly created sets I ran both the methods. In the Penalized method with genres separated and genres combined are 0.865 and 0.865 respectively which are same. In the Penalized method these results are as expected. In order to see any tangible effect of genres in Penalized method data modeling has to be carried out with a modified model. In the Matrix Factorization with SGD the results With genres combined is 0.781 and genres separated the RMSE is about 0.686. The effect of separating the genres made about 10% improvement over the genres combined in the MF with SGD. Matrix Factorization with Parallel Stochastic Gradient Descent (SGD) showed an improvement of about 15% over the Penalized Least Squares. However, computational time in the MF SGD method with genres separated is not presented in this report due to its extremely high computational time. One thing is very much evident that without proper consideration of genres in our analysis we cannot find the optimal RMSE in either penalized regression method or with Parallel Stochastic Gradient Descent SGD method.

Acknowledgements:

My RMSE results are purely based on known algorithms namely Penalized Least Squares and Matrix Factorization with Stochastic Gradient Descent (SGD) methods. To me learning the existing algorithms are important and it is vital to come up with a better algorithm in the near future. Of course my sources of knowledge on Data Science are the online dsbook by Prof. Irizarry, online sources including but not limited to RPubs, DataCamp, Kaggle, GitHub, etc. So, I must duly acknowledge Prof. Rafael A. Irizarry for putting together an online Data Science professional package for us to learn remotely from the different parts of the world. Of course I am grateful to the edx support group for their relentless effort to keep many of us connected with the uninterrupted and fully functional IT support and logistics for playing the videos and running the codes. As I mentioned earlier that I started this program without even knowing the name of programming language called R. My self-confidence was boosted once I finished all eight courses within a short span of time. I am thankful to my peers through discussion forum of edx for their positive and valuable discussions and dialogues that really helped me in solving various R problems and related assignments. Finally I look forward to team up with peers and partners to continue with Machine Learning in R and Python for useful applications in the near future.

References:

1. **https://rafalab.github.io/dsbook/**
2. **https://rafalab.github.io/dsbook/matrix-factorization.html**
3. **https://github.com/yixuan/recosystem**
4. **http://www.csie.ntu.edu.tw/~cjlin/libmf/**
5. **http://www.albertauyeung.com/post/python-matrix-factorization/**