MovieLens RMSE Project for course PH125.9X

Miguel Angel Bustos Sáez

11/30/2021

Table of Contents

INTRODUCTION

Aproved the eight R programming language courses, this is the first of two projects, corresponding to finish the program and to obtain the HarvardX, Professional Certificate in Data Science. This report, it´s about the MovieLens Data Set and the submission of making a RMSE or Root-Mean Square of Error model, that can make good recommendations to movies spectators.

Because of the covid-19, many companies were obligated to offer their services or products through internet, using strong social networks campaigns as marketing strategy, but others using apps or the webpage to increment the sales transactions, good but what about make good recommendations?

The importance it´s to have the option to choose a good recommended product or service, related that were viewed or with a good evaluation criteria as a rating, star or other, for that reason it`s very important to appreciate the good recommendation products because that will push clients to buy these recommended products or services.

In this project, the main objective is to visualizing how an algorithm can reach that challange, how this project make an customized recommendation of good movies. For example, many apps, are doing this, international retailers and some national companies are determinatig the importance to implement intelligent and predicted ways to make good recommendation to customers, clients, netflix viewers, spotify listeners and many others offerings in the market.

One of the best recommendation systems is the RMSE method, this project shows a good data analysis and RMSE algorithm to make good movies recommendations.

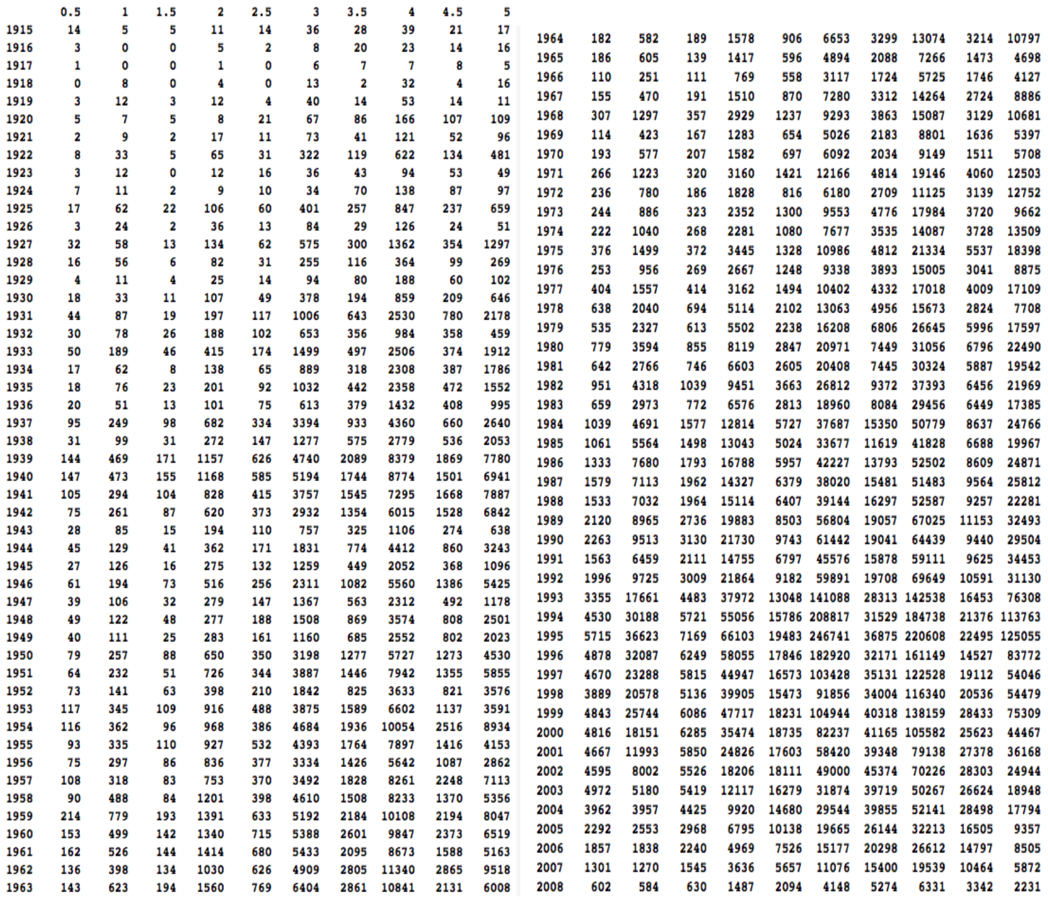
PART ONE

Edx data set at the glance

Global Visualization Dataset

This is EDX dataset table visualization, 0.5 to 5 rankings columns and year files from 1915 to 2008

```
table(edx1$year, edx1$rating)
```



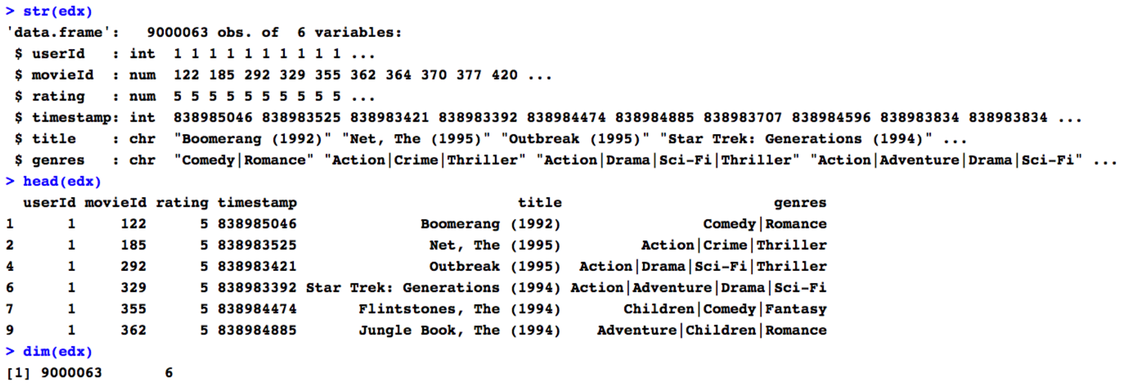
```
         0.5       1     1.5       2     2.5       3     3.5       4     4.5       5
1915      14       5       5      11      14      36      28      39      21      17
1916       3       0       0       5       2       8      20      23      14      16
1917       1       0       0       1       0       6       7       7       8       5
1918       0       8       0       4       0      13       2      32       4      16
1919       3      12       3      12       4      40      14      53      14      11
1920       5       7       5       8      21      67      86     166     107     109
1921       2       9       2      17      11      73      41     121      52      96
1922       8      33       5      65      31     322     119     622     134     481
1923       3      12       0      12      16      36      43      94      53      49
1924       7      11       2       9      10      34      70     138      87      97
1925      17      62      22     106      60     401     257     847     237     659
1926       3      24       2      36      13      84      29     126      24      51
1927      32      58      13     134      62     575     300    1362     354    1297
1928      16      56       6      82      31     255     116     364      99     269
1929       4      11       4      25      14      94      80     188      60     102
1930      18      33      11     107      49     378     194     859     209     646
1931      44      87      19     197     117    1006     643    2530     780    2178
1932      30      78      26     188     102     653     356     984     358     459
1933      50     189      46     415     174    1499     497    2506     374    1912
1934      17      62       8     138      65     889     318    2308     387    1786
1935      18      76      23     201      92    1032     442    2358     472    1552
1936      20      51      13     101      75     613     379    1432     408     995
1937      95     249      98     682     334    3394     933    4360     660    2640
1938      31      99      31     272     147    1277     575    2779     536    2053
1939     144     469     171    1157     626    4740    2089    8379    1869    7780
1940     147     473     155    1168     585    5194    1744    8774    1501    6941
1941     105     294     104     828     415    3757    1545    7295    1668    7887
1942      75     261      87     620     373    2932    1354    6015    1528    6842
1943      28      85      15     194     110     757     325    1106     274     638
1944      45     129      41     362     171    1831     774    4412     860    3243
1945      27     126      16     275     132    1259     449    2052     368    1096
1946      61     194      73     516     256    2311    1082    5560    1386    5425
1947      39     106      32     279     147    1367     563    2312     492    1178
1948      49     122      48     277     188    1508     869    3574     808    2501
1949      40     111      25     283     161    1160     685    2552     802    2023
1950      79     257      88     650     350    3198    1277    5727    1273    4530
1951      64     232      51     726     344    3887    1446    7942    1355    5855
1952      73     141      63     398     210    1842     825    3633     821    3576
1953     117     345     109     916     488    3875    1589    6602    1137    3591
1954     116     362      96     968     386    4684    1936   10054    2516    8934
1955      93     335     110     927     532    4393    1764    7897    1416    4153
1956      75     297      86     836     377    3334    1426    5642    1087    2862
1957     108     318      83     753     370    3492    1828    8261    2248    7113
1958      90     488      84    1201     398    4610    1508    8233    1370    5356
1959     214     779     193    1391     633    5192    2184   10108    2194    8047
1960     153     499     142    1340     715    5388    2601    9847    2373    6519
1961     162     526     144    1414     680    5433    2095    8673    1588    5163
1962     136     398     134    1030     626    4909    2805   11340    2865    9518
1963     143     623     194    1560     769    6404    2861   10841    2131    6008

1964     182     582     189    1578     906    6653    3299   13074    3214   10797
1965     186     605     139    1417     596    4894    2088    7266    1473    4698
1966     110     251     111     769     558    3117    1724    5725    1746    4127
1967     155     470     191    1510     870    7280    3312   14264    2724    8886
1968     307    1297     357    2929    1237    9293    3863   15087    3129   10681
1969     114     423     167    1283     654    5026    2183    8801    1636    5397
1970     193     577     207    1582     697    6092    2034    9149    1511    5708
1971     266    1223     320    3160    1421   12166    4814   19146    4060   12503
1972     236     780     186    1828     816    6180    2709   11125    3139   12752
1973     244     886     323    2352    1300    9553    4776   17984    3720    9662
1974     222    1040     268    2281    1080    7677    3535   14087    3728   13509
1975     376    1499     372    3445    1328   10986    4812   21334    5537   18398
1976     253     956     269    2667    1248    9338    3893   15005    3041    8875
1977     404    1557     414    3162    1494   10402    4332   17018    4009   17109
1978     638    2040     694    5114    2102   13063    4956   15673    2824    7708
1979     535    2327     613    5502    2238   16208    6806   26645    5996   17597
1980     779    3594     855    8119    2847   20971    7449   31056    6796   22490
1981     642    2766     746    6603    2605   20408    7445   30324    5887   19542
1982     951    4318    1039    9451    3663   26812    9372   37393    6456   21969
1983     659    2973     772    6576    2813   18960    8084   29456    6449   17385
1984    1039    4691    1577   12814    5727   37687   15350   50779    8637   24766
1985    1061    5564    1498   13043    5024   33677   11619   41828    6688   19967
1986    1333    7680    1793   16788    5957   42227   13793   52502    8609   24871
1987    1579    7113    1962   14327    6379   38020   15481   51483    9564   25812
1988    1533    7032    1964   15114    6407   39144   16297   52587    9257   22281
1989    2120    8965    2736   19883    8503   56804   19057   67025   11153   32493
1990    2263    9513    3130   21730    9743   61442   19041   64439    9440   29504
1991    1563    6459    2111   14755    6797   45576   15878   59111    9625   34453
1992    1996    9725    3009   21864    9182   59891   19708   69649   10591   31130
1993    3355   17661    4483   37972   13048  141088   28313  142538   16453   76308
1994    4530   30188    5721   55056   15786  208817   31529  184738   21376  113763
1995    5715   36623    7169   66103   19483  246741   36875  220608   22495  125055
1996    4878   32087    6249   58055   17846  182920   32171  161149   14527   83772
1997    4670   23288    5815   44947   16573  103428   35131  122528   19112   54046
1998    3889   20578    5136   39905   15473   91856   34004  116340   20536   54479
1999    4843   25744    6086   47717   18231  104944   40318  138159   28433   75309
2000    4816   18151    6285   35474   18735   82237   41165  105582   25623   44467
2001    4667   11993    5850   24826   17603   58420   39348   79138   27378   36168
2002    4595    8002    5526   18206   18111   49000   45374   70226   28303   24944
2003    4972    5180    5419   12117   16279   31874   39719   50267   26624   18948
2004    3962    3957    4425    9920   14680   29544   39855   52141   28498   17794
2005    2292    2553    2968    6795   10138   19665   26144   32213   16505    9357
2006    1857    1838    2240    4969    7526   15177   20298   26612   14797    8505
2007    1301    1270    1545    3636    5657   11076   15400   19539   10464    5872
2008     602     584     630    1487    2094    4148    5274    6331    3342    2231
```

“The good of this code, is that we can easily visualize the year and rating”

Global Visualization Dataset

Structure view

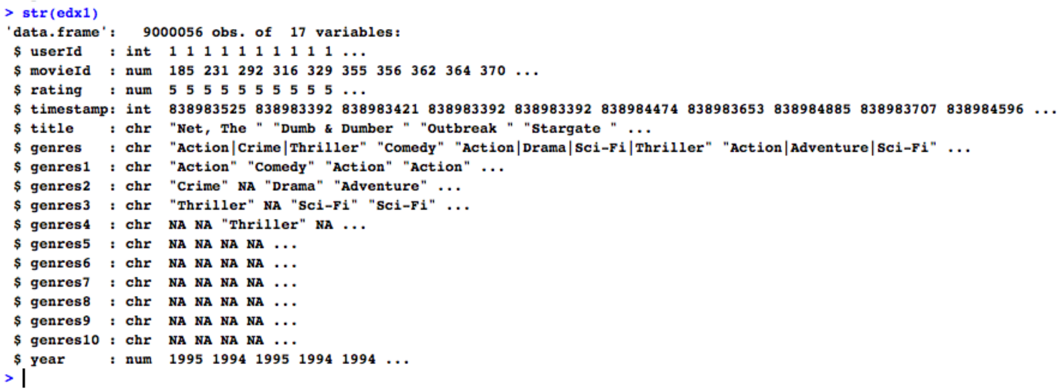
```
str(edx)
head(edx)
dim(edx)
```



```
> str(edx)
'data.frame':	9000063 obs. of  6 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  122 185 292 329 355 362 364 370 377 420 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838985046 838983525 838983421 838983392 838984474 838984885 838983707 838984596 838983834 838983834 ...
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Star Trek: Generations (1994)" ...
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Adventure|Drama|Sci-Fi" ...
> head(edx)
  userId movieId rating timestamp                         title                        genres
1      1     122      5 838985046              Boomerang (1992)                Comedy|Romance
2      1     185      5 838983525               Net, The (1995)         Action|Crime|Thriller
4      1     292      5 838983421               Outbreak (1995)  Action|Drama|Sci-Fi|Thriller
6      1     329      5 838983392 Star Trek: Generations (1994) Action|Adventure|Drama|Sci-Fi
7      1     355      5 838984474       Flintstones, The (1994)       Children|Comedy|Fantasy
9      1     362      5 838984885       Jungle Book, The (1994)    Adventure|Children|Romance
> dim(edx)
[1] 9000063       6
```

Edx1 it`s the ordered data, Each variable in each column

Each genre in each column, including year variable

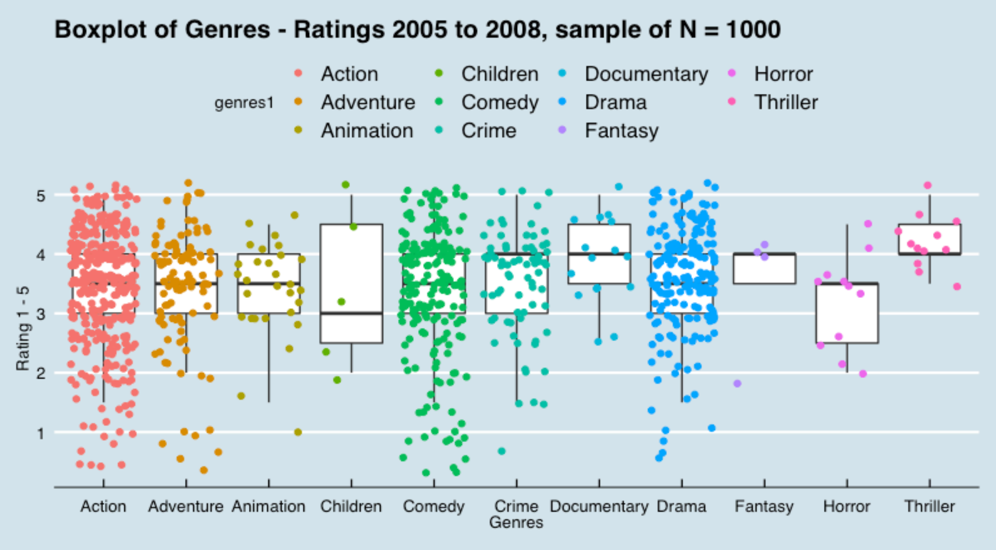


```
> str(edx1)
'data.frame':	9000056 obs. of  17 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  185 231 292 316 329 355 356 362 364 370 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838983525 838983392 838983421 838983392 838983392 838984474 838983653 838984885 838983707 838984596 ...
 $ title    : chr  "Net, The " "Dumb & Dumber " "Outbreak " "Stargate " ...
 $ genres   : chr  "Action|Crime|Thriller" "Comedy" "Action|Drama|Sci-Fi|Thriller" "Action|Adventure|Sci-Fi" ...
 $ genres1  : chr  "Action" "Comedy" "Action" "Action" ...
 $ genres2  : chr  "Crime" NA "Drama" "Adventure" ...
 $ genres3  : chr  "Thriller" NA "Sci-Fi" "Sci-Fi" ...
 $ genres4  : chr  NA NA "Thriller" NA ...
 $ genres5  : chr  NA NA NA NA ...
 $ genres6  : chr  NA NA NA NA ...
 $ genres7  : chr  NA NA NA NA ...
 $ genres8  : chr  NA NA NA NA ...
 $ genres9  : chr  NA NA NA NA ...
 $ genres10 : chr  NA NA NA NA ...
 $ year     : num  1995 1994 1995 1994 1994 ...
> |
```

Edx1 ranking performance visualization between 2005 to 2008, taking a sample of 1000 observations

The latest years of the data set visualization has a lot of action ranked values

```
muestra1 = edx1 %>%
  filter(year %in% 2005:2008) %>%
  sample_n(1000)

ggplot(data = muestra1, aes(genres1, rating))+
  geom_boxplot(outlier.shape = NA)+
  geom_point(position = "jitter", aes(color = genres1))+
  theme_economist()+
  labs(x = "Genres", y = "Rating 1 - 5", title = "Boxplot of Genres - Ratings 
2005 to 2008, sample of N = 1000")
```

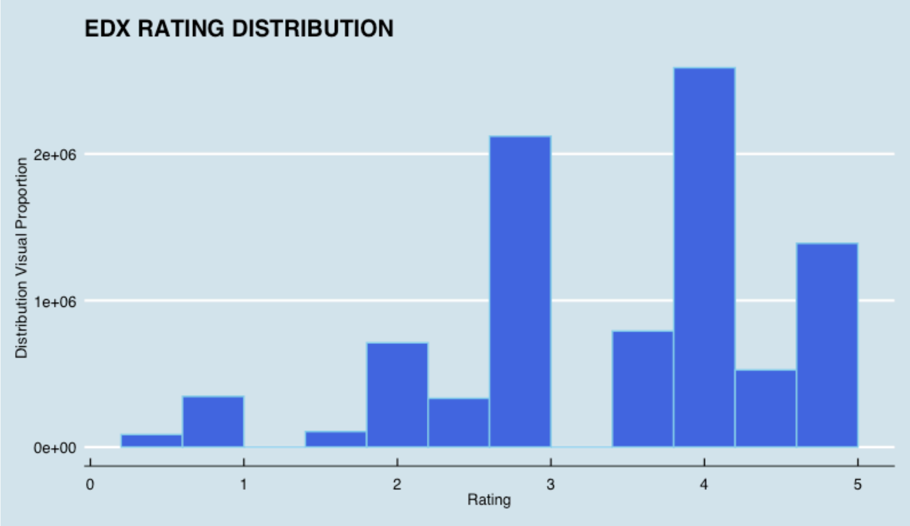


"Ratings boxplot from 2005 to 2008"

Edx1 ranking distribution

Now the ranked distribution from 0.5 to 5 is visualizing through this barplot

```
edx1 %>%
  ggplot(aes(rating))+
  geom_histogram(binwidth = 0.4, fill = "royalblue", color = "skyblue")+
  theme_economist()+
  labs(x = "Rating", y = "Distribution Visual Proportion", title = "EDX RATIN
G DISTRIBUTION")
```

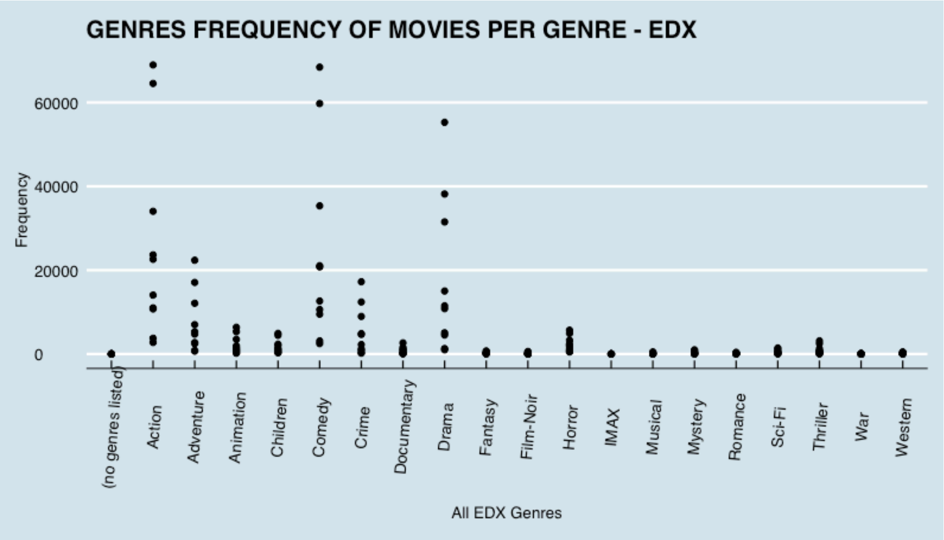


"Four is the most frequently rating in all data set"

Frequency of movies per Genre, the most viewed

This shows the most movies viewed in all data set, the leaders are; Action, Comedy and Drama

```
genres_frequency <- as.data.frame(table(edx1$genres1, edx1$rating))
class(genres_frequency)

gf <- ggplot(data = genres_frequency)
gf + geom_point(aes(x = Var1, y = Freq/10))+
  labs(x = "All EDX Genres", y = "Frequency", title = "GENRES FREQUENCY OF MO
VIES PER GENRE - EDX")+
  theme_economist()+
  theme(axis.text.x = element_text(angle = 85, vjust = 0.6))
```

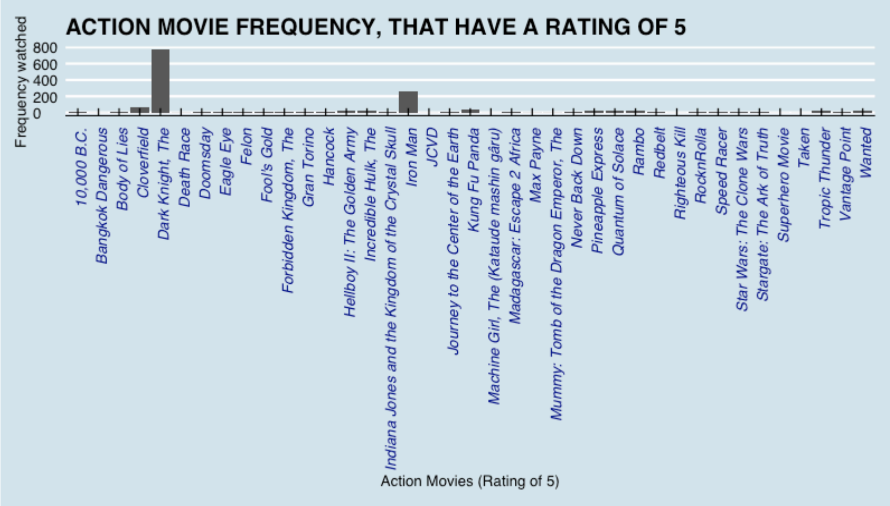


"The most frequented movies per genre are action and comedy"

Action Movie Frequency, that have a rating of 5

The dark knight and the Iron Man are highly the most ranked movies of 5 ranked movies

```
top_watched <- edx1 %>%
  filter(rating == "5", genres1 == "Action", year == "2008")

tg <- ggplot(data = top_watched)
tg + geom_bar(aes(x = title))+
  labs(x = "Action Movies (Rating of 5)", y = "Frequency watched", title = "A
CTION MOVIE FREQUENCY, THAT HAVE A RATING OF 5")+
  theme_economist()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.8, color = "darkblue
", face = "italic", hjust = 1))
```



"The most action movies rated 5 viewed"

PART TWO

Modeling data

The part 1 was critical to ordered the variables of genre and year and visualize it Now, that we have these variables in Edx, we can work more efficient in a RMSE The Root-mean-square error (RMSE) is the accuracy measure that calculate the prediction error rate. What are the residuals between "actual rating" and "predicted rating"? RMSE gives that answer In other words, the RMSE its the measure of how spread out these residuals are and how concentrated the data is around the line of best fit So its the difference between "Actual Ratings" and the Forecast or "predicted ratings"

test_set & forecast_rating

test_set

```
mu <- mean(edx1$rating)

set.seed(1)
test_set %>%
  left_join(edx1 %>%
              group_by(movieId) %>%
              summarise(fe = mean(rating - mu)), by = "movieId")
join <- test_set %>%
  left_join(edx1 %>%
              group_by(movieId) %>%
              summarise(fe = mean(rating - mu)), by = "movieId")



forecast_rating <- mu + join$fe
forecast_rating
```

Create bootstrap sample with createDataPartition

```
set.seed(1)
test_index <- createDataPartition(y = edx1$rating, times = 1, p = 0.5, list =
FALSE)
train_set <- edx1[-test_index,]
test_set <- edx1[test_index,]
```

Visualizing the data sets

```
str(edx1)
str(test_index)
str(train_set)
str(test_set)
```

Selecting the test_set_r and the forecast_rating

```
valid <- test_set %>%
  semi_join(edx1, by = "movieId") %>%
  semi_join(edx1, by = "userId") #Return all rows from x with a match in y

length(test_set)

test_set_r <- test_set$rating
forecast_rating <- mu + join$fe
```

First RMSE [1] 0.9436553

Using the test_set_r: 0.5 to 5.0, and forecast_rating: different values (2.939235 3.738641 3.419572...)

```
mu <- mean(edx1$rating)
mu

set.seed(1)
difference <- test_set_r-forecast_rating
rmse <- sqrt(mean(difference^2))

rmse
[1] 0.9436553

or:

sqrt(mean((test_set_r-forecast_rating)^2))

[1] 0.9436553
```

Second RMSE [1] 1.102293

Using the test_set_r: 0.5 to 5.0, and now forecast_rating: using ceiling function, that rounds up to the nearest integer, with more realistic number. This way shows that the residual reaches: [1] 1.102293.- And this is still too far:

```
set.seed(1)
forecast_rating2 <- ceiling(forecast_rating)
sqrt(mean((test_set_r-forecast_rating2)^2))
[1] 1.102293
```

Third RMSE [1] 0.7399887

Customizing what`s good for the Movie Espectators

Creating two vectors called test_sample and forecast_sample, we obtain #[1] 0.7399887 residuals value choosing in the range from 3.5 to 5.0 range.

Making good recommendations movies, over 3.5 ranked movies, is the way to get an optimal RMSE, for that reason, it`s neccessary to fix some parameters, that make a better recommendations:

Just replace the parameter, each of these, here we have some vectors parameters:

```
parameter1 <- 3.0
parameter2 <- 3.5
parameter3 <- 4.0
parameter4 <- 4.5
parameter5 <- 5.0
```

What will be the RMSE results, in the recommendation is more than these ratings?, here we have some vectors parameters results:

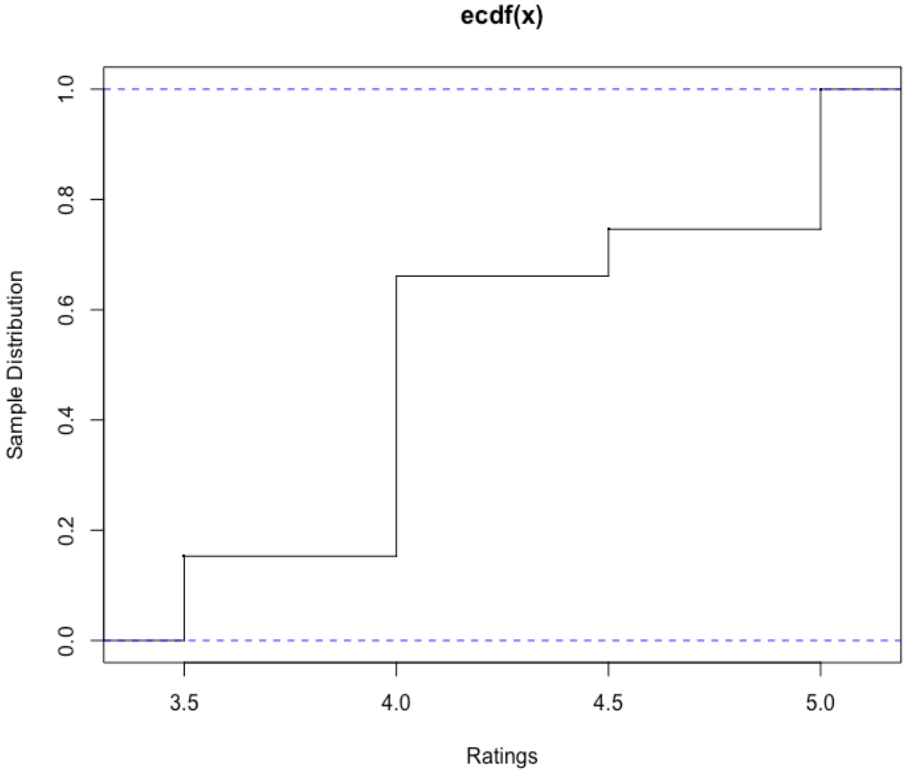
```
parameter1 : approximately [1] 1.000417 RMSE result
parameter2 : approximately [1] 0.7399887 RMSE result
parameter3 : approximately [1] 0.6379002 RMSE result
parameter4 : approximately [1] 0.3106445 RMSE result
parameter5 : approximately [1] 0 RMSE result, this is because it`s the limit
ranking
```

The best RMSE it`s obtained using parameter3 [1] 0.6379002 rmse result, anyway , just using parameter2, we obtained a great recommendation [1] 0.7399887 RMSE result.

These vector parameters, can be changed in this code, and we will have different RMSE results, this code including >= symbol, so it is the best way to make recommendations to clients and obtain a better RMSE, making good recommensations movies, over 3.5 ranked value, that value is parameter 2:

```
set.seed(1)
index1 <- edx1$rating >= parameter2 #<- You must here write the parameter vec
```

```
tor 1, 2, 3, 4 or 5
test_sample <- edx1$rating[index1] %>% sample(size = 3000)
test_sample

set.seed(1)
index2 <- join$rating >= parameter2 #<- You must here write the parameter vec
tor 1, 2, 3, 4 or 5
forecast_sample <- join$rating[index2] %>% sample(size = 3000)
forecast_simple_mu <- join$fe[forecast_sample]+ mu
forecast_sample



plot.ecdf(test_sample, forecast_sample, ylab = "Sample Distribution",  xlab =
"Ratings", col.01line = "blue", verticals = TRUE)
```



"The sample N = 3000 ranking distribution, as ECDF Empirical Distribution Function"

To obtain a RMSE value

```
sqrt(mean((test_sample-forecast_sample)^2))

[1] 0.7399887
```

CONCLUSION AND RECOMMENDATION

With the intention to make good recommendations and not bad recommendations, the conclusion is recommend good movies, this is equal or greater than 3.5 rated.

For that reason, if the intention is to make a good recommendation, the critical decision is recommend a good movie to the viewers, that recommendation is over 3.5, and the gap will be more close, as a residual of 0.7399887 RMSE.

Nowadays many companies are offering online products and services, they won`t recommend a bad products, books, movies, licences, etcetera, these companies will recommend products well evaluated, to guarantee a great experience, happens the same in this project, it is necessary make good recomendations.

The recommendation is to recommend all movies that are equal or more than 3.5 ranked, so the spread will be roughly 0.7399887 RMSE and that guarantee the clients will have a great experience.

Biography

Data Science HarvardX - Professional Certificate in Data Science

Raphael Irizarry onnline dsbook - Introduction to Data Science

Raphael Irizarry book - Introduction to Data Science, CRC

Raphael Irizarry - Webpage

ImperialX, Imperial College Business School - EDX Course, Data Analysis Essentials

Root Mean Squared Deviation - Wikipedia

About optimizing RMSE or choose samples

The ratings members provided for recent movies provided more predictive power than older ratings - 2007 NETFLIX PRIZE

They can sample a few videos before settling on one - TECHDIRT

Root-mean square of the differences between the predicted values and observed values - Even though it is feasible to conduct manual classification, it is generally very time consuming and error prone when dealing with a large number of samples - NATURE

Having an accurate understanding of starch supports critical ration development and drives economic decisions. Invest in proper sampling, sample frequently to understand true variation - HOARDS

What is a Recommender System? A recommender system is an intelligent system that predicts the rating and preferences of users on products - ANALITICALSINDIAMAG