Data Science: Capstone

PH125.9x

HarvardX Professional Certificate in Data Science

December 2020

Miguel Angel Bustos Sáez

Introduction

This inform it´s a recommendation system, based in Random Forest[1] for categorical data and RMSE[2] for numerical data, using Edx dataset joined it with Imdb ratings it gives important information to get a better forecast approximation. The steps of the project are:

1. R Libraries

    - Tidyverse
    - Magrittr
    - Lubridate
    - randomForest
    - caret
    - readr

2. Methods and analysis
3. Datasets structures
4. Data model
5. Model results

    - Random forest accuracy (For categorical variable)
    - RMSE forecast accuracy (For numerical variable)

6. Visualization
7. Conclusion

[1]https://en.wikipedia.org/wiki/Root_mean_square
[2]https://en.wikipedia.org/wiki/Random_forest

2 Methods and analysis

Data cleaning

The data cleaning process was dividing the Edx dataset in movies.csv and ratings.csv and doing an inner_joing with imdb dataset, that provided more information to the model; votes numbers, imdb average and Id user. Edx dataset had a separate **I** symbol and it had all genres in just one column, so the separate **I** symbol was removed using the separate function, and str_c function ordered each genre in each column:

genero_1	genero_2	genero_3	genero_4	genero_5	genero_6	genero_7	genero_8	genero_9	genero_10
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western
Action	Adventure	Comedy	Crime	Drama	Film-Noir	Horror	Mystery	Thriller	Western

To fill some empty cells, in year variable, it processes to take the average year, and fill some cells with the average year 1992. The genres are ordered and the missing pieces are filled with `NA`:

Titulo	Genero	genero_1	genero_2	genero_3	genero_4	genero_5	genero_6	genero_7	genero_8	genero_9	genero_10
All	All	All	All	All	All	All	All	All	All	All	All
Amelie (Fabuleux de...	Comedy|Romance	Comedy	Romance	NA	NA	NA	NA	NA	NA	NA	NA
Wild Strawberries (S...	Drama	Drama	NA	NA	NA	NA	NA	NA	NA	NA	NA
Teddy Bear (Mis) (19...	Comedy|Crime	Comedy	Crime	NA	NA	NA	NA	NA	NA	NA	NA
Talk to Her (Hable c...	Drama|Romance	Drama	Romance	NA	NA	NA	NA	NA	NA	NA	NA
Lord of the Rings: T...	Adventure|Fantasy	Adventure	Fantasy	NA	NA	NA	NA	NA	NA	NA	NA
City of God (Cidade ...	Action|Adventure...	Action	Adventure	Crime	Drama	Thriller	NA	NA	NA	NA	NA
Spanish Apartment, ...	Comedy|Drama|R...	Comedy	Drama	Romance	NA	NA	NA	NA	NA	NA	NA
Finding Nemo (2003)	Adventure|Anima...	Adventure	Animation	Children	Comedy	NA	NA	NA	NA	NA	NA
Lost in Translation (...	Comedy|Drama|R...	Comedy	Drama	Romance	NA	NA	NA	NA	NA	NA	NA
M. Hulot's Holiday (...	Comedy	Comedy	NA	NA	NA	NA	NA	NA	NA	NA	NA
Strada, La (1954)	Drama	Drama	NA	NA	NA	NA	NA	NA	NA	NA	NA

3 Datasets structures

It is edx dataset, where genres variable was separated by this **I** symbol:

```
> str(edx)
'data.frame':   9000060 obs. of  6 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  122 185 292 316 329 355 362 364 370 377 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838985046 838983525 838983421 838983392 838983392 83898
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)"
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|
```

And now this is our data, an iner_joint combination between Edx with Imdb datasets, the recommendation systems are based on this data:

```
> str(data)
'data.frame':   25475592 obs. of  9 variables:
 $ Id_usuario    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Id_pelicula   : num  296 306 307 665 899 ...
 $ Indice        : num  5 3.5 5 5 3.5 4 3.5 3.5 5 4 ...
 $ Marca_temporal: int  1147880044 1147868817 1147868828 1147878820
 $ Titulo        : chr  "Pulp Fiction (1994)" "Three Colors: Red (Tr
 "Underground (1995)" ...
 $ Genero        : chr  "Comedy|Crime|Drama|Thriller" "Drama" "Drama
 $ Ano           : num  1994 1994 1993 1995 1952 ...
 $ Promedio_Idbm : num  8.9 8.1 7.9 8.1 8.3 7 7.6 8.2 8.2 8.1 ...
 $ Numero_Votos  : num  1785576 88165 86831 54330 213897 ...
```

It is vital to have the most information as possible in genres, having well ordered in different columns, for that objective, this data is a combination between Edx and Imdb datasets, to obtain a major accuracy in the predictions.

4 Data model

The data, was saved as data.rda, taken the major and equal 4 índice or number rating. The *training_u* is a length of 50% and *testing_u* vectors with the 50% of users. *Training* it´s a filter of the id_users of *training_u*, selecting the variables, and adding it a generos column, here it is the code.

The data has been divided in two objects; *trainung_u* and *testing_u*:

```
data = data %>% filter(Genero != "(no genres listed)" & Indice >= 4)

Usuarios = unique(data$Id_usuario)
training_u = sample(Usuarios, length(Usuarios)*0.50,replace = F)
testing_u = Usuarios[-training_u]

training = data %>%
  filter(Id_usuario %in% training_u)%>%
  select(Indice,Ano,Generos,Marca_temporal,Promedio_Idbm,Numero_Votos)%>%
  mutate(Generos = factor(Generos))
```

And here it is a random forest and the set.seed:

```
set.seed(0)
modelo = randomForest(Generos ~  ., data = training,
                      ntree=30,
                      method="class",
                      norm.votes=FALSE,
                      do.trace=10,
                      proximity = FALSE,
                      importance=TRUE)

muestra_2 = list(testing_u, modelo)
save(muestra_2, file="muestra_2.rda")
```

5 Model results

I choose two algorithms approximation; Random Forest for categorical data, and RMSE for numerical data.

5.1 Random forest for categorical data

This code with the OOB estimate error rate of 68.29%:

```
[[2]]

Call:
 randomForest(formula = Generos ~ ., data = training, ntree = 30,      method = "class", norm.votes = FALSE,
y = FALSE,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 30
No. of variables tried at each split: 2

        OOB estimate of  error rate: 68.29%
Confusion matrix:
            Action Adventure Animation Children Comedy  Crime Documentary   Drama Fantasy Film-Noir Horror
Action      512370    228677     13410     7724  94154  98728         286  161530   47469       461  22819
Adventure   252618    375029     53725    65513 113541  10361         883  119752  114518       614   6330
Animation    17751     71411     82494    79021  59403   1858         409   26315   37051         6    745
Children     12503     81347     75452   107744  76552   1533         183   35131   49319         1    191
Comedy      100586    128853     51290    70650 827068  86070        4689  298922   70374       441  21265
Crime       145489     15476      3073     2666 115719 319131         597  274911    3194     16138  22241
Documentary    348       834       432      182   6116    482       74958    5411      45         1    112
Drama       172623     95054     18948    39400 263126 225139        3207 1447829   49027      7781  24281
Fantasy      50863    121424     42073    58388  88629   3128          79   63176  161122       191   9811
Film-Noir     2138      2064        11        0    480  20051           0   13314     160     19279    198
Horror       30518      6147       977      210  26557  18290         102   37531   11261       404 125570
IMAX         49995     39551      7867     9021   4185  13046         770   28279   17496         0   1253
Musical       1893     15913     21880    32417  43240    468        2974   32790   23057         7   2043
Mystery      35349     11316      2033     1744  13538  62818         218  106683    8215     13390  20616
Romance      35371     30897     13498    11492 266865  12005          92  341426   42826      2115   2536
Sci-Fi      205633    144221     13290     9905  46111  12783         197   92452   14009       596  36395
Thriller    264094     71076      1693      210  46188 185235          49  274035   12680     14218  68646
War          48808     14358      1097      280  30158   2972        2219  151361    1110         4    172
Western      23541     14382       262      427  10816   1458          11   25257     170         0    516
            IMAX Musical Mystery Romance Sci-Fi Thriller    War Western class.error
Action     36579    2046   20263   34381 185093   221049  40743   12809   0.7056345
Adventure  36061   11481    7593   22291 133220    77956  10876   13922   0.7370587
Animation   4761   19777    1661   13845  10700     1309   1547     174   0.8082596
Children    4841   21371    1501   12381   9572      294    256     313   0.7803317
Comedy      2558   29193    9130  201026  44708    45872  28139    8915   0.5925270
Crime       8136     703   38647    9632   9930   184982   3075     897   0.7283152
Documentary  645    2439     190      49    119       39   2156       7   0.2073389
Drama      12061   16962   67824  216055  72354   201685 107479   17568   0.5266062
Fantasy    18255   12641    6410   32083   9886    12427   1853     168   0.7673688
Film-Noir      0      11    9915    2955    346    12815     10       4   0.7698057
Horror      1798    1586   24763    3311  33381    63098    102     405   0.6746984
IMAX       51795    2986    1812    2749  29390    12915   1743      43   0.8115833
Musical     2153   43066      64   21197   1351      310    518     191   0.8246013
Mystery     3927      93  131863   11906  24635   107965    517     553   0.7634231
Romance     2843   19404   13248  271696  15874    24868  20140    3285   0.7596634
Sci-Fi     30308    1249   23197   10969 266407   114334   1184    2633   0.7403119
Thriller   13786     607  118491   27622 109244   436276   8840    1444   0.7362989
War         1887     605   16149    1396  10975   102123   1778    7367455
Western       68     258     788    5808   1025     1881   1257   38225   0.6969877
```

Errors have different class.errors fluctuations: 0.70 for action, 0.73 for adventure, 0.80 for animation, etc.

Unique() function shows the id_users, the 1990 user has been chosen here:

```
unique(testing$Id_usuario)

testing_1900 = testing %>% filter(Id_usuario == 1900) %>%
  select(Indice,Ano,Genero,Marca_temporal,Promedio_Idbm,Numero_Votos) %>%
  mutate(Genero = factor(Genero))
testing_1900$Genero = fct_expand(testing_1900$Genero,levels(factor(testing$Genero))[2:20])
```

Genres and movies suggestions for the 1990 user

	Indice	Ano	Genero	Marca_temporal	Promedio_Idbm	Numero_Votos
1	3.0	1995	Comedy|Romance	1301834435	6.3	36058
2	2.0	1995	Action|Romance|Western	1301834430	7.2	169053
3	2.5	1995	Horror|Sci-Fi	1301834464	5.8	72994
4	3.5	1977	Action|Adventure|Sci-Fi	1301834854	8.6	1207095
5	4.5	1994	Comedy|Crime|Drama|Thriller	1301835024	8.9	1785576
6	4.5	1994	Crime|Drama	1301835077	9.3	2285846
7	5.0	1994	Comedy|Drama|Romance|War	1301834954	8.8	1761399
8	3.5	1994	Action|Comedy	1301834425	6.5	95555
9	3.0	1993	Thriller	1301835195	7.8	262807
10	4.0	1991	Action|Sci-Fi	1301834882	8.5	977571
11	4.0	1972	Crime|Drama	1301835761	9.2	1577887
12	3.0	1941	Film-Noir|Mystery	1301834493	8.0	147039
13	3.5	1964	Comedy|Drama|Musical|Romance	1301834532	7.8	85000
14	3.0	1988	Action|Crime|Thriller	1301835145	8.2	769718
15	2.5	1996	Comedy|Drama	1301834537	7.2	74180
16	4.0	1980	Action|Adventure|Sci-Fi	1301834826	8.7	1135176
17	4.0	1962	Adventure|Drama|War	1301834486	8.3	262358

Showing 1 to 18 of 86 entries, 6 total columns

Genres suggestions for the 1990 user:

```
> generos_sugerir = names(sort(table(prediccion_1900),decreasing = T)[1:3])
> generos_sugerir
[1] "Drama"     "Adventure" "Comedy"
```

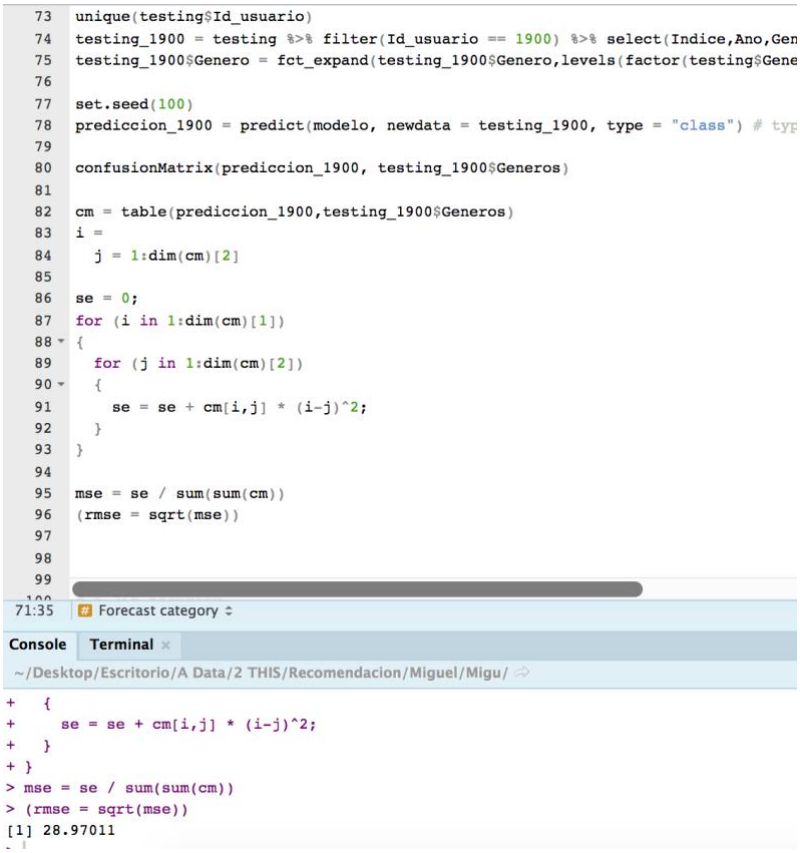
Movies suggestions for the 1990 user:

```
i1 = data %>% filter(Genero %in% "Drama" & Promedio_Idbm >= 8 & Ano >= 2015) %>% distinct(Titulo)
```

	Titulo
1	Room (2015)
2	Lion (2016)
3	Dangal (2016)
4	Bohemian Rhapsody (2018)
5	Wonder (2017)
6	Shoplifters (2018)
7	Fly Away Solo (2015)
8	The Wild Pear Tree (2018)

5.2 RMSE for numerical data

Here it is a RMSE approximation, it´s 28.97 the numerical approximation to obtain a recommendation movie, this approximation is obtained based on a confusion matrix[3] and RMSE result:



```
unique(testing$Id_usuario)
testing_1900 = testing %>% filter(Id_usuario == 1900) %>% select(Indice,Ano,Gen
testing_1900$Genero = fct_expand(testing_1900$Genero,levels(factor(testing$Gene

set.seed(100)
prediccion_1900 = predict(modelo, newdata = testing_1900, type = "class") # typ

confusionMatrix(prediccion_1900, testing_1900$Generos)

cm = table(prediccion_1900,testing_1900$Generos)
i =
  j = 1:dim(cm)[2]

se = 0;
for (i in 1:dim(cm)[1])
{
  for (j in 1:dim(cm)[2])
  {
    se = se + cm[i,j] * (i-j)^2;
  }
}

mse = se / sum(sum(cm))
(rmse = sqrt(mse))
```

71:35 Forecast category

Console Terminal

~/Desktop/Escritorio/A Data/2 THIS/Recomendacion/Miguel/Migu/

```
+   {
+     se = se + cm[i,j] * (i-j)^2;
+   }
+ }
> mse = se / sum(sum(cm))
> (rmse = sqrt(mse))
[1] 28.97011
```

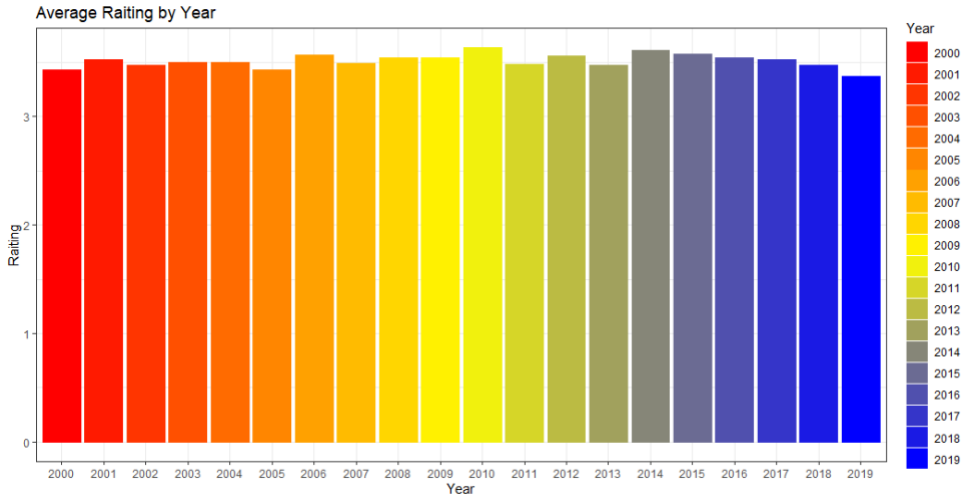
Continuous with our testing, here we can see again our 1990 user, we obtain in the descendant arrange; 7.87 for drama and 6.4 for comedy genres:

```
> testing_1900 %>%
+   filter(Generos %in% generos_sugerir) %>%
+   group_by(Generos) %>%
+   summarise(idbm = mean(Promedio_Idbm)) %>%
+   arrange(desc(idbm))
# A tibble: 2 x 2
  Generos  idbm
  <fct>    <dbl>
1 Drama    7.87
2 Comedy   6.4
>
```

[3]https://en.wikipedia.org/wiki/Confusion_matrix

6 Visualization

In this histogram we have the average ratings from 2000 to 2019 of data dataset:



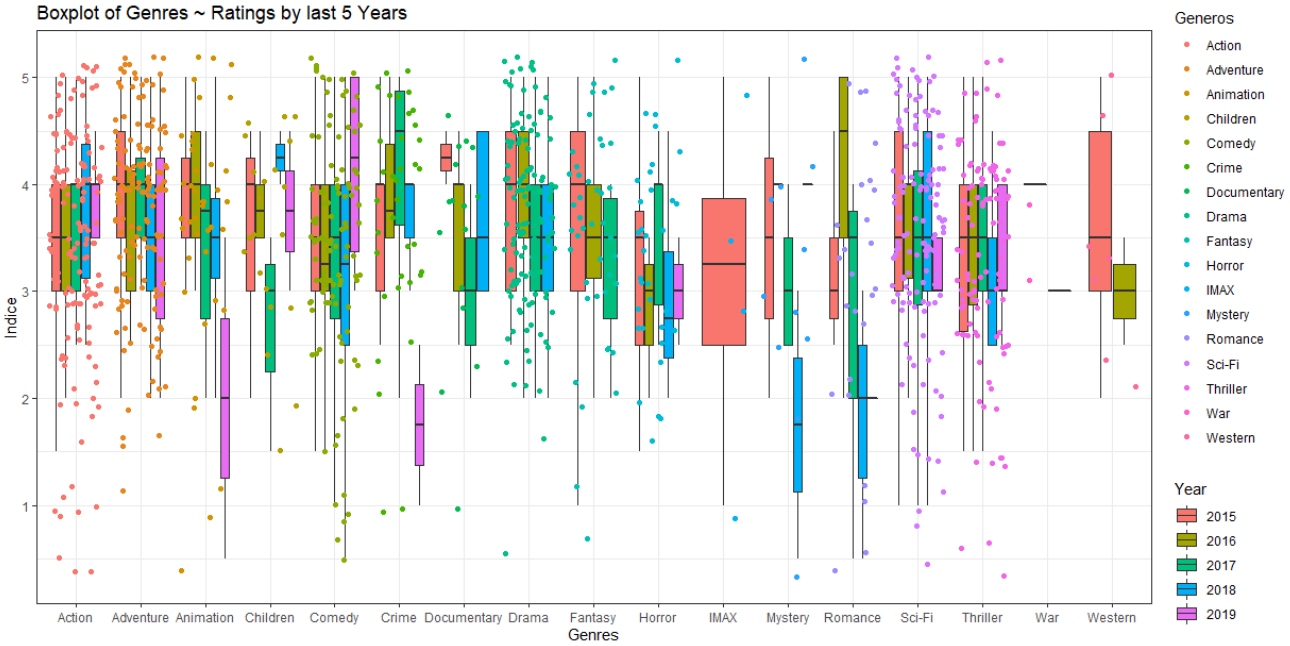
*Ratings at moving, are also knowing as **content rating**, it rates the suitability on TV broadcasts, movies, internet, music, comics books or video games to its audience* [4].

Is very important have the information of content rating because provide guidance to consumers, particularly parents, to help them decide whether or not to watch any movie. For example, Netflix decide maturity ratings by country and for kids 7+, for teens 13+ and adult 16+, 18+ . The good of Netflix is that the user can choose maturity ratings or block shows[5]. In Netflix the average of watching tv is two hours per user, concluding all, data and visualization is very important to take decisions.

[4]https://en.wikipedia.org/wiki/Content_rating#:~:text=A%20content%20rating%20(also%20known,suitable%20to%20view%20said%20media

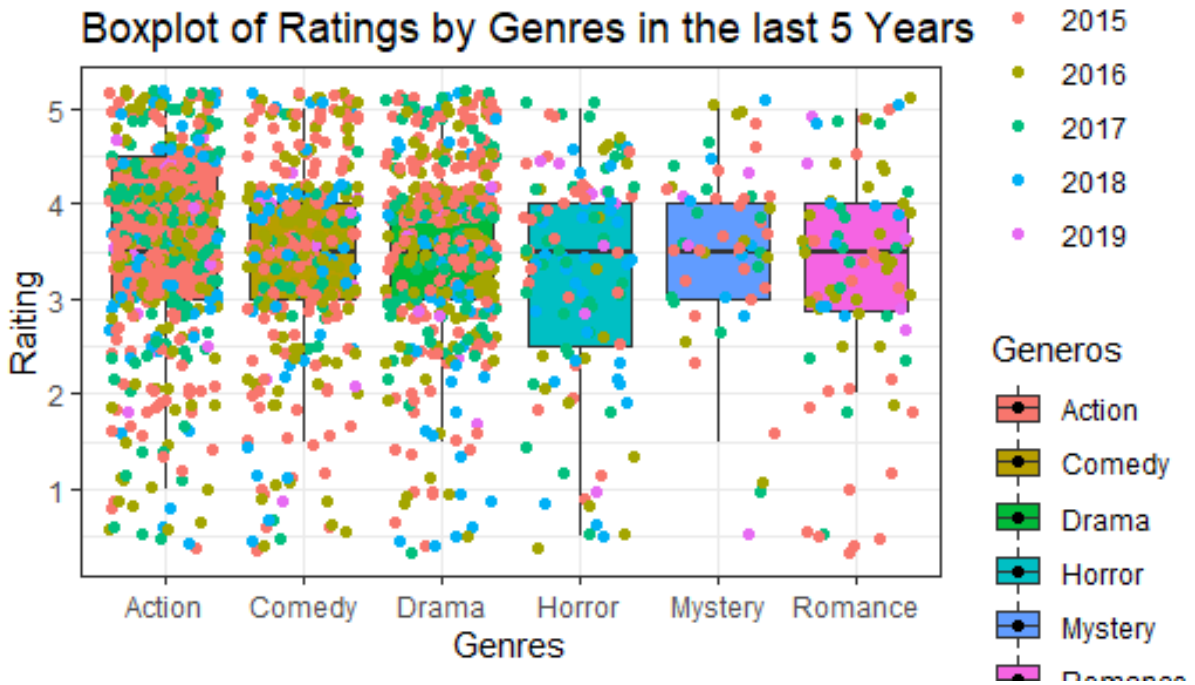
[5]https://help.netflix.com/en/node/2064

17 genres rated 1 to 5 (From 2015 to 2019):



Here are some genres, in the years 2015 to 2019. Action, comedy and drama rocks!

Horror is very fluctuated, mystery is highly rated, romance is very scattered:



Conclusion

The main process in this project is the movies.csv, many movies were repeated, and the process was fixing this database without movies repeated, obtained 62.423 observations and three variables. In the column genres it continuous with more than one genre, but in the code, this | character was removed. Also, this project has an IMDB ratings data set 1.076.066 observations and three variables; ID, average rating and number of votes. Both datasets were mutate in the code.

The courses helped me to have tools to sort the data, unite and based on this establish the RMSE and the Random Forrest to obtain effective predictions, and for this, I decided to establish these two algorithms, both categorical and numerical, thus having a more global clarity about of the data, with this, the histograms and boxplots presented in a more clear and efficient visualization.

The edx team, always where very efficient in all responses, Rafael Irizarry classes where very nutritive in any knowledge and great tips to use the R function in different examples. I conclude this course was very hard, it takes me a lot of hours to understand some examples, I spent almost two years to finish this whole program, and I will continuous studying statistics and R languaje.