NETFLIX

MOVIE RECOMMENDATION ENGINE

GROUP 8 - PRESENTATION

SUMMARY

01 WHY

- Context

02 DATA EXPLANATORY

- Modifying the database on R
- Studies of data on Tableau

03 OUR APPROACH

- Processing the data
- Our recommender system choice

04 R SHINY INTERFACE

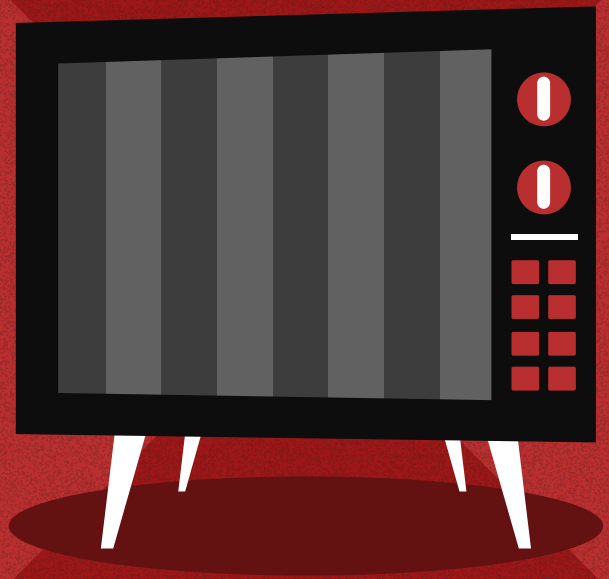
- Code explanations

05 MODELLING

- Functioning explanations
- Model test

06 CONCLUSION

- Findings
- Challenges

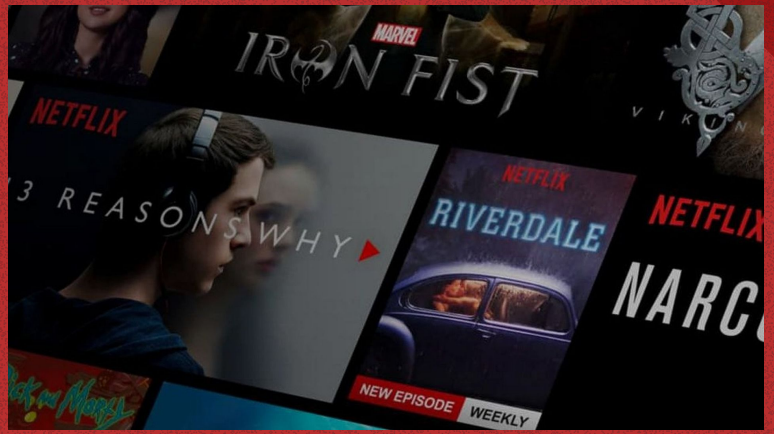


CONTEXT

As members of the Netflix team of data scientists, we wanted to develop a new movie recommendation engine for our clients

1- WHY

- Recommendation engines are used by market leaders
- Gives users recommendations based on similar user's data
- Being recommended content you like = more time on platform
- More time on the platform = lower customer churn
- For example, Netflix’s recommendation algorithms produce $ 1 billion a year in value from customer retention (India, 2019)

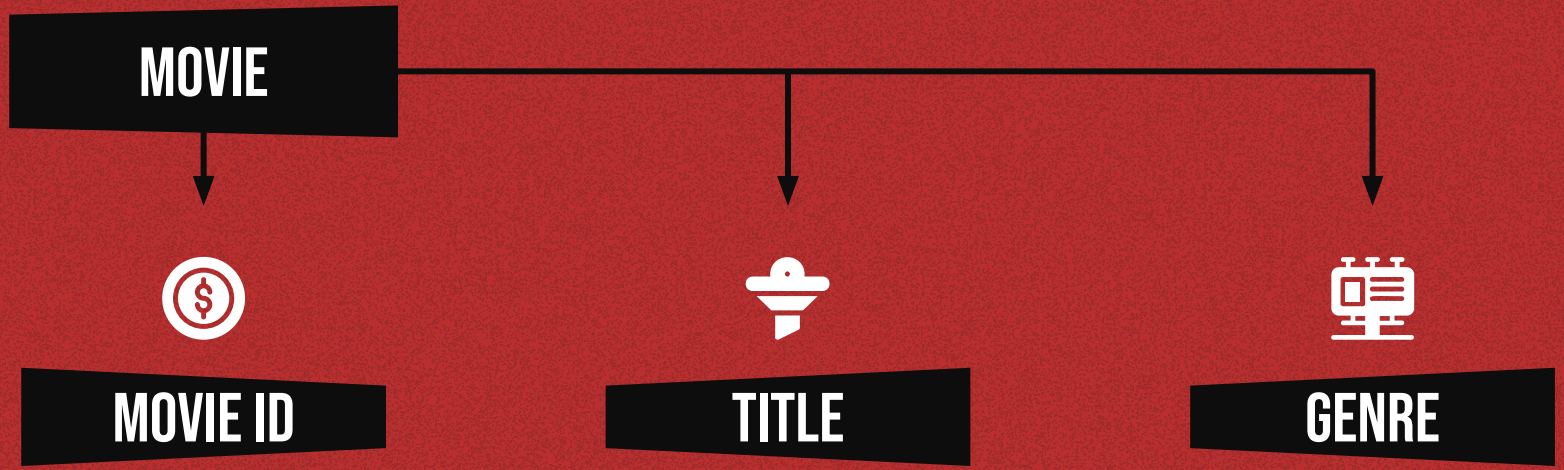


80%

Of Netflix views come from the recommendation algorithm

2- DATA EXPLANATORY

OUR DATABASE - MOVIE



18

OUR DATABASE - RATING

0-5

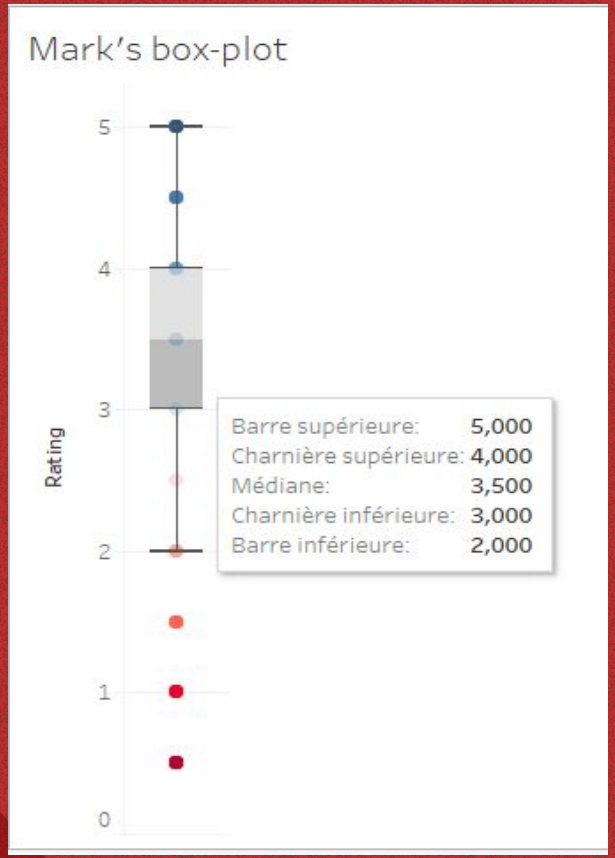
The time when the rating is given

MOVIE PER GENRES

```
genrenames = xy$genres %>% unique %>% stringr::str_split("\\|") %>% unlist %>% unique()
genres = setNames(data.table(matrix(nrow = dim(xy)[1], ncol = length(genrenames))), genrenames)

for(i in seq_along(genres)) {
  #genres[,genrenames[i] := grepl(genrenames[i],xy$genres)]
}
xy =cbind(xy,genres)
write.csv(xy,file = "full_data.csv")
```

RATING BOXPLOT



Q1

VALUE

3

MEDIAN

VALUE

3,5

Q3

VALUE

4

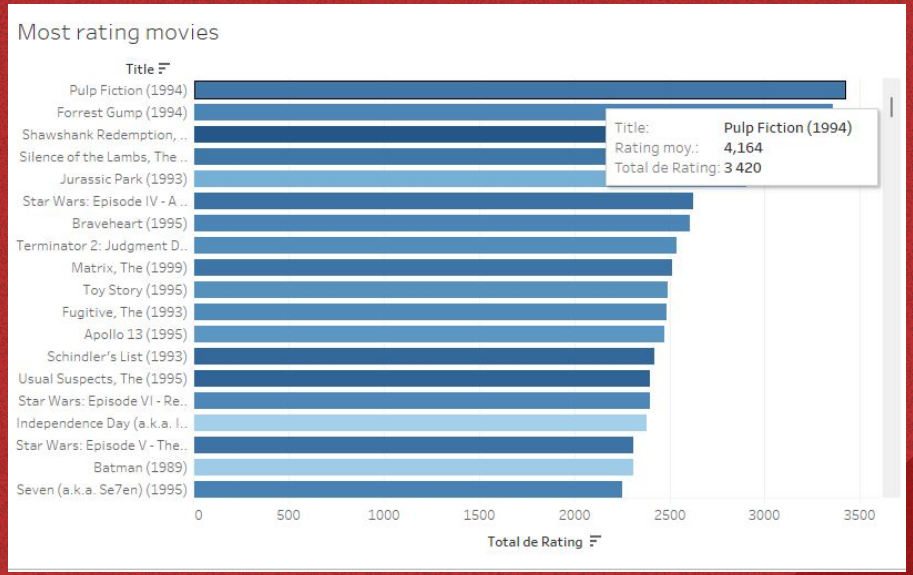
GOOD RATE

>=3,5

Consider a good rate if the rating is 3,5 or above

MOST RATED MOVIE

1. PULP FICTION (1994)
2. FORREST GUMP (1994)
3. SHAWSHANK REDEMPTION
4. SILENCE OF THE LAMBS



Pages

Colonnes: MOY(Rating)

Lignes: Title

Filtres

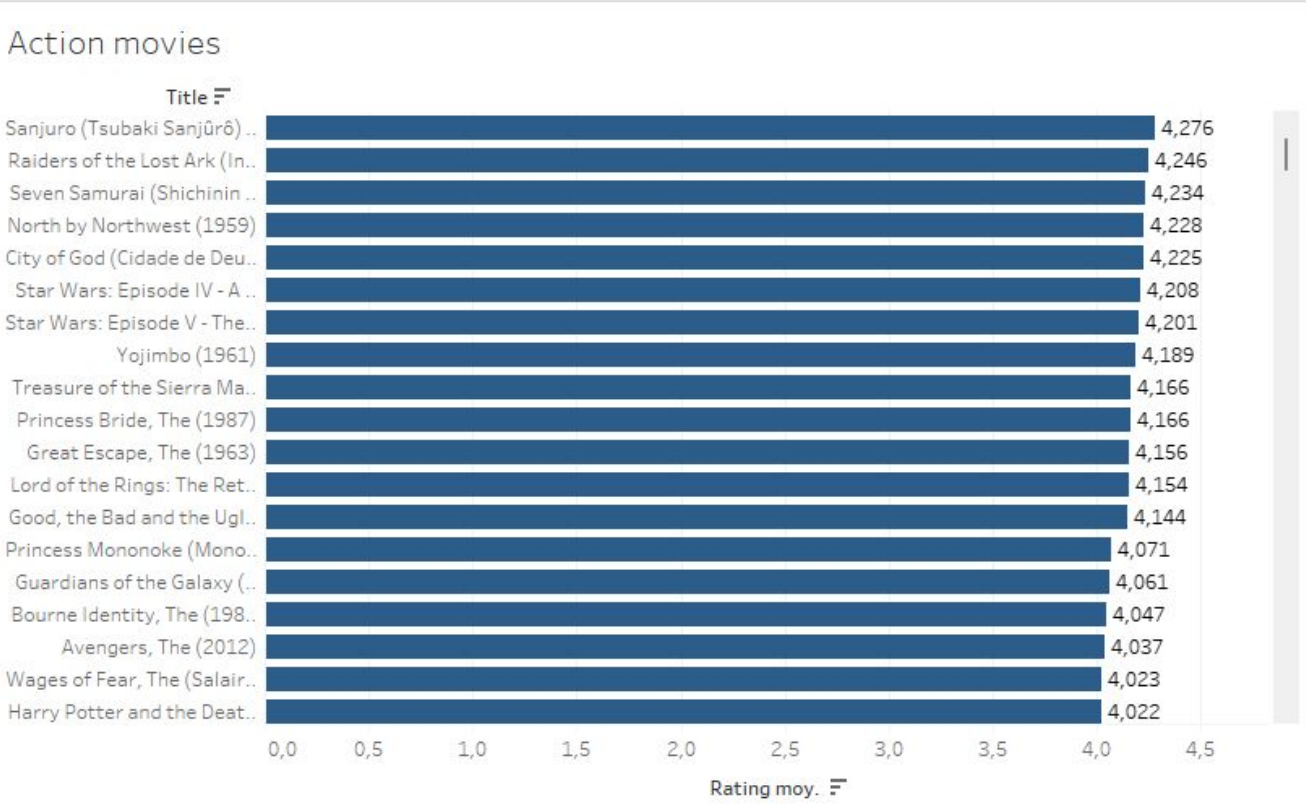
- Action : Vrai
- Adventure : Vrai
- CPT(Rating)

Repères

Automatique

Couleur | Taille | Étiquette | Détails | Infobulle

MOY(Rating)

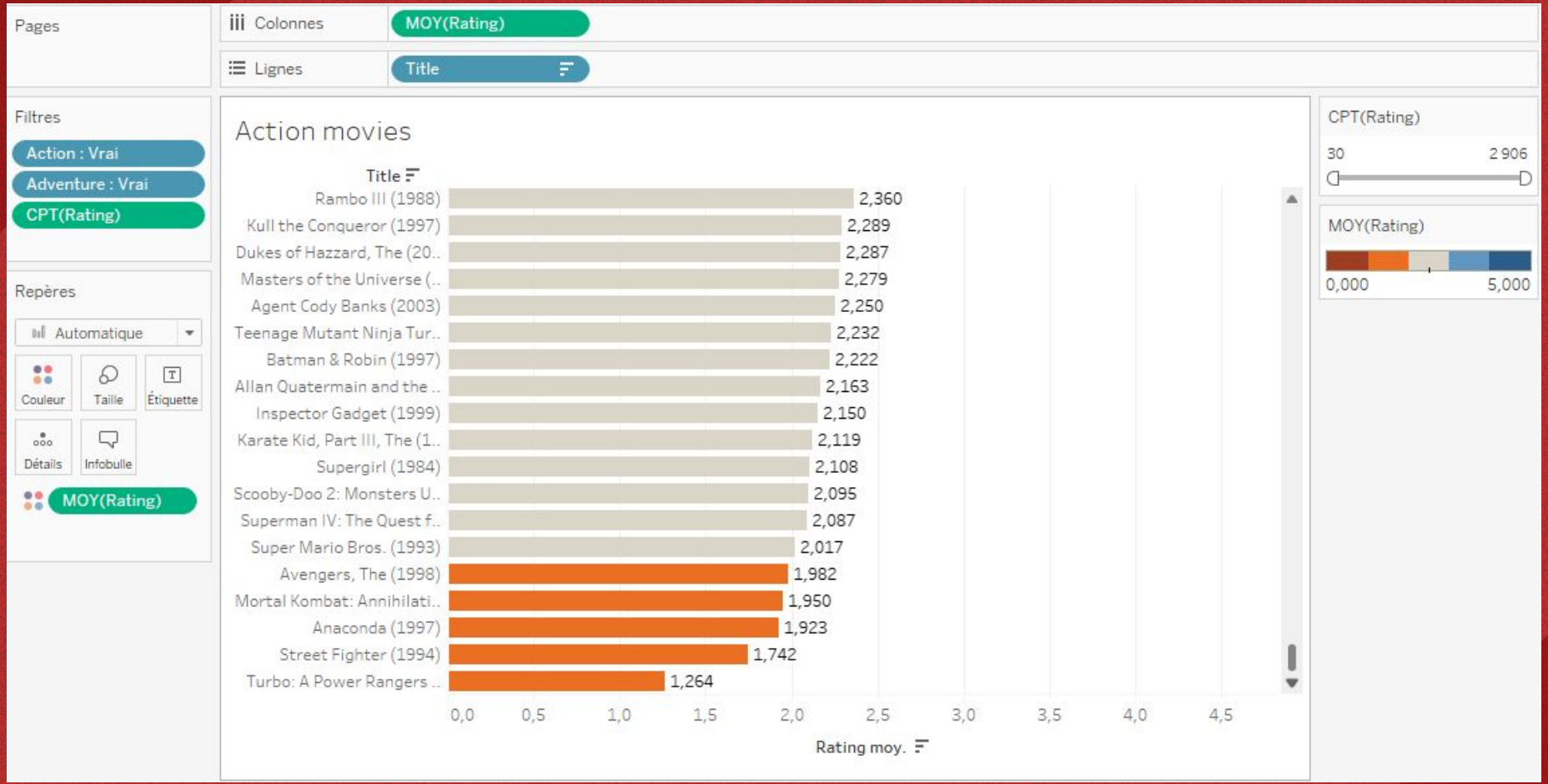


CPT(Rating)

30 2 906

MOY(Rating)

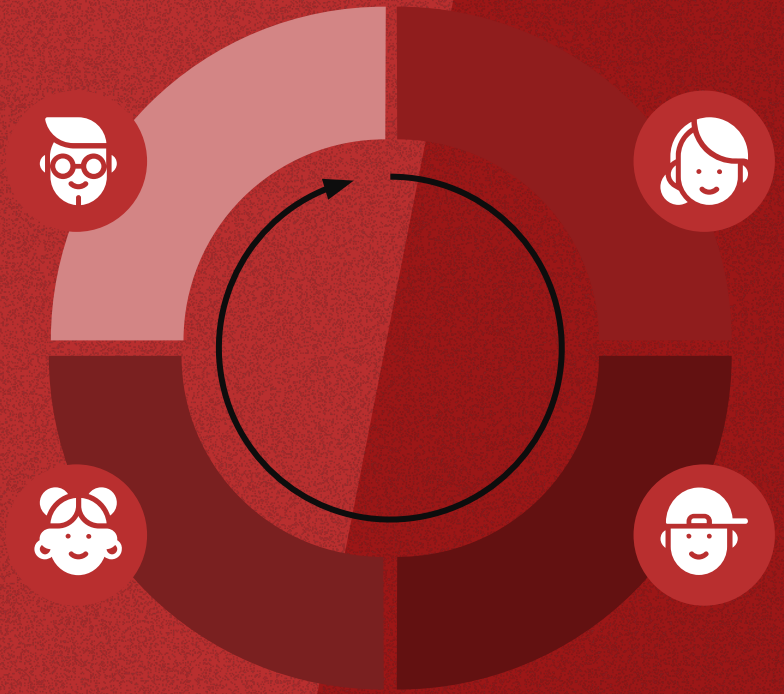
0,000 5,000


Pages
Colonnes
MOY(Rating)
Lignes
Title
Filtres
Action : Vrai
Adventure : Vrai
CPT(Rating)
Repères
Automatique
Couleur
Taille
Étiquette
Détails
Infobulle
MOY(Rating)
Action movies
Title
Rambo III (1988) 2,360
Kull the Conqueror (1997) 2,289
Dukes of Hazzard, The (20.. 2,287
Masters of the Universe (.. 2,279
Agent Cody Banks (2003) 2,250
Teenage Mutant Ninja Tur.. 2,232
Batman & Robin (1997) 2,222
Allan Quatermain and the .. 2,163
Inspector Gadget (1999) 2,150
Karate Kid, Part III, The (1.. 2,119
Supergirl (1984) 2,108
Scooby-Doo 2: Monsters U.. 2,095
Superman IV: The Quest f.. 2,087
Super Mario Bros. (1993) 2,017
Avengers, The (1998) 1,982
Mortal Kombat: Annihilati.. 1,950
Anaconda (1997) 1,923
Street Fighter (1994) 1,742
Turbo: A Power Rangers .. 1,264
0,0 0,5 1,0 1,5 2,0 2,5 3,0 3,5 4,0 4,5
Rating moy.
CPT(Rating)
30
2 906
MOY(Rating)
0,000
5,000

TOP 3 MOVIES REGARDING EACH GENRE

ACTION	HORROR	ROMANCE
Tsubaki Sanjûrô	Silence Lambs	Sunset Boulevard
Raiders of the Lost Ark	Diabolique	Brief Encounter
Seven Samurai	Peeping Tom	Casablanca

3- OUR APPROACH

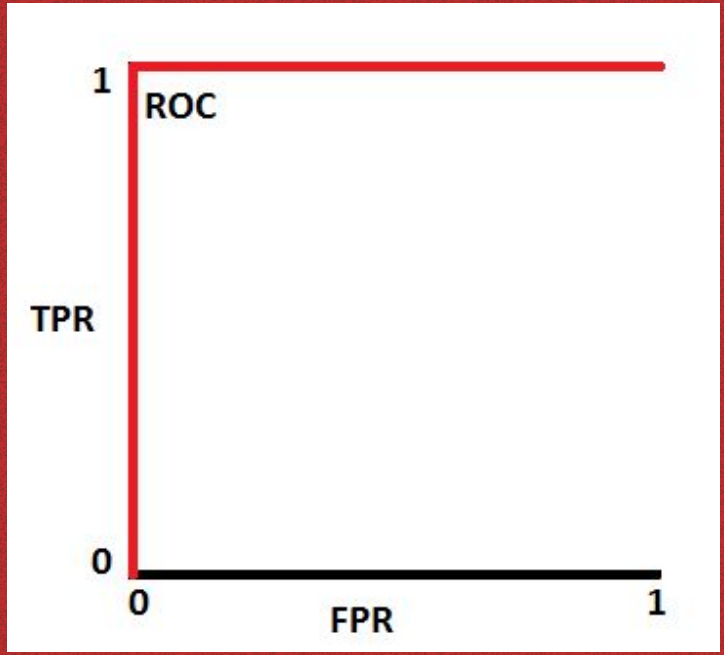


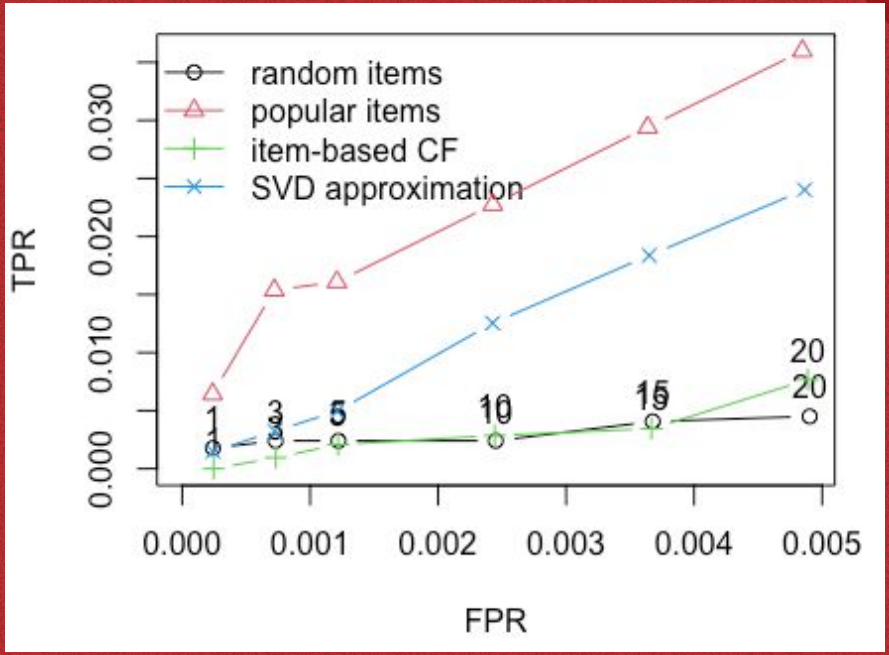
PROCESSING THE DATA

- Merge the datasets into one
- Chose only movies that were rated more than 10 times
- Change the order of the columns
- Change the data into a matrix

CHOICE OF THE RECOMMENDER SYSTEM

- Usage of the Recommenderlab library
- Usage of the ROC curve
- Popular method is the best choice
- Usage of the UBCF method


1
ROC
TPR
0
0
FPR
1


random items
popular items
item-based CF
SVD approximation
TPR
0.000
0.010
0.020
0.030
0.000
0.001
0.002
0.003
0.004
0.005
FPR

4- R SHINY INTERFACE

INTERFACE BASES

MOVIE CHOICES

The user has to choose three movies that he likes

MARK CHOICES

For each movie the user has to pick a rate between three four and five

SOME VERY USEFUL FUNCTIONS

FLUIDPAGE

It allows us to create the page

FLUIDROW

Main function to build our interface

SELECTINPUT

It creates a drop-down list

RADIOBUTTONS

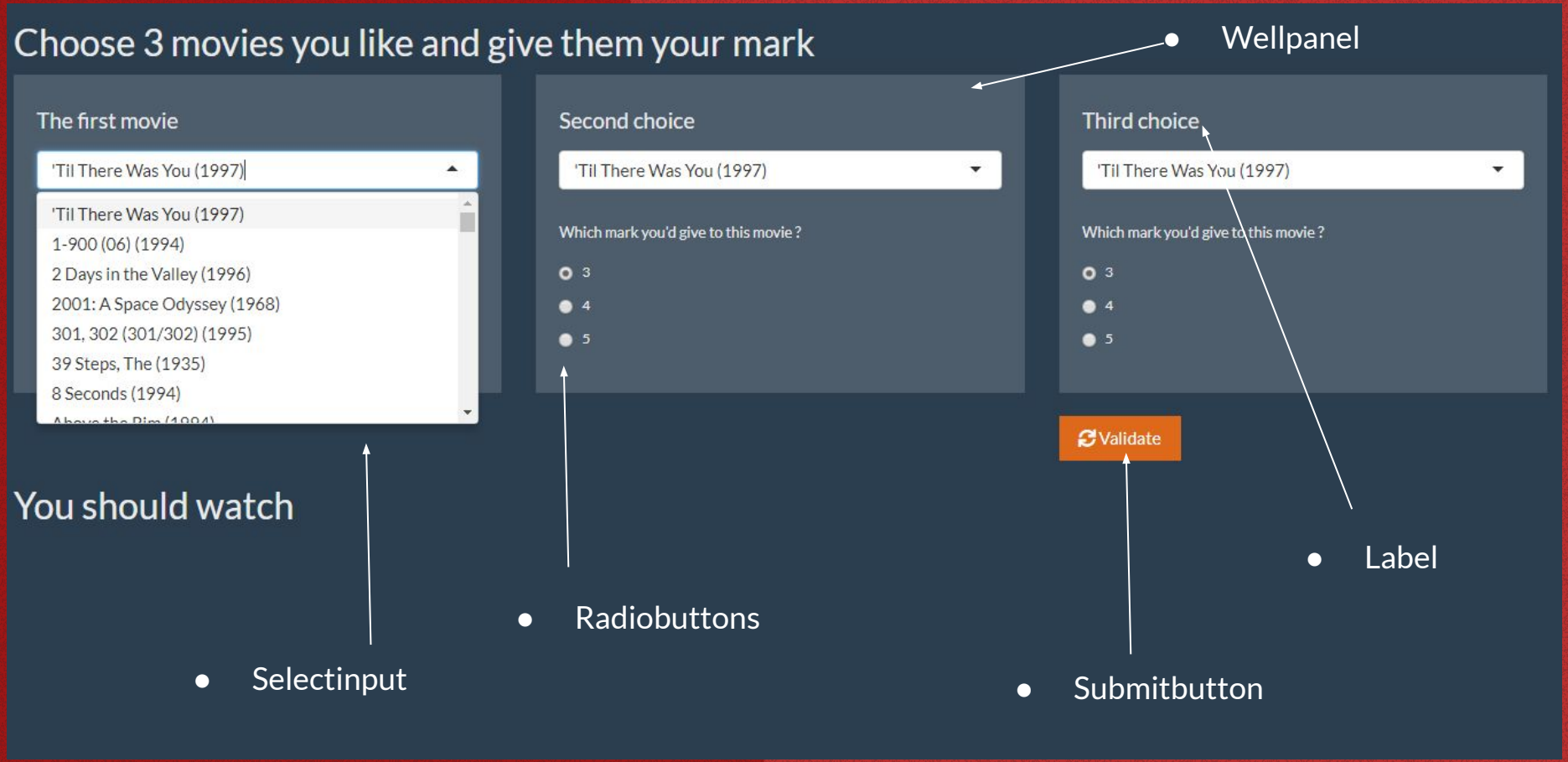
It allows us to create a choice between 3, 4 and 5

WELLPANEL

It delimits each choice

LABEL

It creates a text above the choice


Choose 3 movies you like and give them your mark
The first movie
'Til There Was You (1997)
'Til There Was You (1997)
1-900 (06) (1994)
2 Days in the Valley (1996)
2001: A Space Odyssey (1968)
301, 302 (301/302) (1995)
39 Steps, The (1935)
8 Seconds (1994)
Second choice
'Til There Was You (1997)
Which mark you'd give to this movie ?
3
4
5
Third choice
'Til There Was You (1997)
Which mark you'd give to this movie ?
3
4
5
Validate
You should watch
Wellpanel
Label
Radiobuttons
Selectinput
Submitbutton

OUR SCRIPT

```
ui <- shinyUI(fluidPage(theme = shinytheme("superhero"),
  titlePanel("Choose 3 movies you like and give them your mark"),
  fluidRow(
    column(4,wellPanel(
           selectInput("select", label = h4("The first movie"),
                       choices = sort(as.character(movie_data$title[1:1000]))),
           radioButtons("select4", label = h6("Which mark you'd give to this movie ?"),
                        choices = 3:5,))),
    column(4,wellPanel(
           selectInput("select2", label = h4("Second choice"),
                       choices = sort(as.character(movie_data$title[1:1000]))),
           radioButtons("select5", label = h6("Which mark you'd give to this movie ?"),
                        choices = 3:5,))),
    column(4,wellPanel(
           selectInput("select3", label = h4("Third choice "),
                       choices = sort(as.character(movie_data$title[1:1000]))),
          radioButtons("select6", label = h6("Which mark you'd give to this movie ?"),
                       choices = 3:5,)),
          submitButton("Validate", icon = icon("refresh"))



    ),

    column(10,
           h2("You should watch  "),
           tableOutput("table"))
```

OTHER FUNCTIONS

COLUMN

This allows us to space out the different things

H

It allows to adjust the size of the text

SORT

It classifies the film in alphabetical order which is very useful for choosing

5- MODELING

MODELING

CLASSIFICATION, SEQUENCE AND DETERMINATION

```
count = table(total2$userId)
count = as.data.frame(count)
colnames(count) = c("userId", "Freq")
#install.packages("dplyr")
library(dplyr)
count = filter(count, Freq >= 10)
total3 = merge(total2, count, by = "userId")
total3 = total3[ , -which(names(total3) %in% c("Freq"))]
total3 = total3[1:10000,]

nb_movies_rated = length(unique(total3$movieId))
nb_movies = length(unique(movie_data$movieId))
movie_data = movie_data[-which((nb_movies %in% nb_movies_rated) == FALSE),]  #we remove all the movies that have not been rated

rownames(movie_data) = NULL
row_num <- which(movie_data[,2] == input)
row_num2 <- which(movie_data[,2] == input2)
row_num3 <- which(movie_data[,2] == input3)

userSelect <- matrix(NA,nb_movies_rated)      #creation of an empty matrix
userSelect[row_num] <- input4
userSelect[row_num2] <- input5
userSelect[row_num3] <- input6
```

Establish table to know the frequencies of notation of each users

filter the frequencies and keep the users with an average rating above 10.

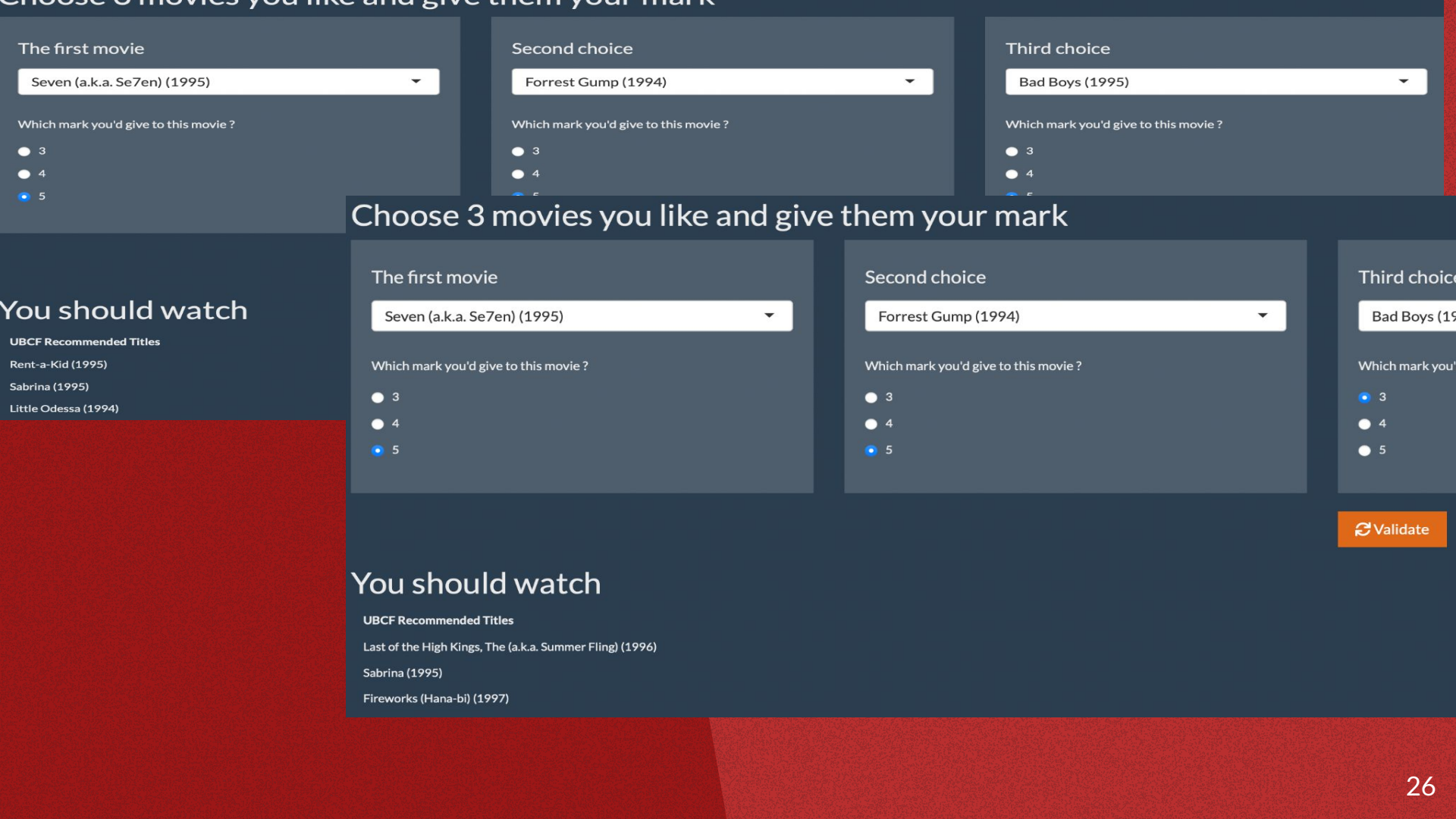
sorted the movies and notations of each users in ascending order and we have chosen to keep the first 10k information.

MODELING

TRANSPOSE, COMBINE AND RECOMMENDER SYSTEM (UBCF MODEL)

```
174   total3 <- dcast(total3, userId~movieId, value.var = "rating", na.rm=FALSE)  #transposing of the total3 matrix
175   total3 <- total3[,-1]
176
177   userSelect <- t(userSelect)
178   colnames(userSelect) <- colnames(total3)
179   total3 <- rbind(userSelect, total3)  #association of our empty matrix and our realRatingMatrix
180   total3 <- as.matrix(total3)
181
182   r3 <- as(total3, "realRatingMatrix")
183
184   #Create Recommender Model. "UBCF" stands for user-based collaborative filtering
185
186   r_POPULAR <- Recommender(r3, method = "UBCF")
187   recom <- predict(r_POPULAR, r3[1], n = 3)
188
189   recom_list <- as(recom, "list")
190   recom_result <- data.frame(matrix(0,3))
191   for (i in c(1:3)){
192     recom_result[i,1] <- movie_data[as.integer(recom_list[[1]][i]),2]
193   }
194   colnames(recom_result) <- "UBCF Recommended Titles"
195   return(recom_result)
196 }
197 shinyApp(ui, server)
```

transpose all our information and combine the matrices to obtain a large table.


The first movie
Seven (a.k.a. Se7en) (1995)
Which mark you'd give to this movie ?
3
4
5
Second choice
Forrest Gump (1994)
Which mark you'd give to this movie ?
3
4
Third choice
Bad Boys (1995)
Which mark you'd give to this movie ?
3
4
You should watch
UBCF Recommended Titles
Rent-a-Kid (1995)
Sabrina (1995)
Little Odessa (1994)
Choose 3 movies you like and give them your mark
The first movie
Seven (a.k.a. Se7en) (1995)
Which mark you'd give to this movie ?
3
4
5
Second choice
Forrest Gump (1994)
Which mark you'd give to this movie ?
3
4
5
Third choic
Bad Boys (19
Which mark you'
3
4
5
Validate
You should watch
UBCF Recommended Titles
Last of the High Kings, The (a.k.a. Summer Fling) (1996)
Sabrina (1995)
Fireworks (Hana-bi) (1997)

CONCLUSION

DATA

- R programming

MODELLING

- Make a table, filter and then matrix

CHALLENGES

- Working with data
- Which method

RESULT

- A new movie recommendation engine

THANKS!