M2.951 Tipologia i cicle de vida de les dades: pràctica 2

Autor: Eric Farran Moreno i Jordi Alvarez Pitarque

2024-06-06

Índex

1 Descripció del dataset

El conjunt de dades top250movies presenta una estructura de 250 observacions i 10 característiques, i classifica les 250 pel · lícules més ben valorades pels usuaris d'una de les bases de dades cinematogràfiques més potents actualment com és la d'*Internet Movie Database*, coneguda popularment com IMDb. Aquesta extracció, actualitzada l'11 d'abril del 2024, es va obtenir aplicant tècniques de web scraping de forma responsable i ètica, respectant les condicions de servei d'IMDb i sense sobrecarregar els seus servidors.

En aquest projecte analític, essencialment, es vol estudiar quina relació guarda la rendibilitat de les pel · lícules amb les altres característiques que es recullen en el dataset generat en la pràctica 1. La condició de rendibilitat pels metratges s'esbrinarà més endavant, en l'apartat de generació de noves característiques, considerant només el pressupost, els ingressos totals i un llindar, encara per definir. Aquesta nou atribut serà fixat més endavant com a variable de classe per a procedir amb l'anàlisi de les dades.

A continuació es mostra un resum de les variables del dataset i del seu contingut, indicant el tipus de dada segons R:

```
# Importació dataset
  top250movies <- read.csv('dataset/top250movies.csv')
  glimpse(top250movies)
```

```
## Rows: 250
## Columns: 10
## $ Title          <chr> "Original title: The Shawshank Redemption", "Original t~
## $ Genre          <chr> "Drama", "Crime, Drama", "Action, Crime, Drama", "Crime~
## $ Year           <int> 1994, 1972, 2008, 1974, 1957, 1993, 2003, 1994, 2001, 1~
## $ Classification <chr> "13", "18", "12", "18", "A", "12", "12", "18", "12", "1~
## $ Duration       <chr> "2h 22m", "2h 55m", "2h 32m", "3h 22m", "1h 36m", "3h 1~
## $ Rating         <dbl> 9.3, 9.2, 9.0, 9.0, 9.0, 9.0, 9.0, 8.9, 8.9, 8.8, 8.8, ~
## $ Review         <chr> "2.9M", "2M", "2.9M", "1.4M", "862K", "1.4M", "2M", "2.~
## $ Director       <chr> "Frank Darabont", "Francis Ford Coppola", "Christopher ~
## $ Budget         <chr> "$25,000,000 (estimated)", "$6,000,000 (estimated)", "$~
## $ Collection     <chr> "$28,905,764", "$250,342,030", "$1,008,486,720", "$47,9~
```

Breu descripció dels atributs:

- **Title**: Títol original
- **Genre**: Conjunt de gèneres
- **Year**: Any de llançament
- **Classification**: Classificació en diferents format
- **Duration**: Durada en hores i minuts
- **Rating**: Qualificació promig
- **Review**: Número de ressenyes
- **Director**: Director o directora
- **Budget**: Pressupost en moneda local
- **Collection**: Ingressos de taquillatge totals

2 Integració i selecció de les dades d'interès

Atès que només hi ha disponible una font de dades per aquest projecte, no es requereix la integració de diferents datasets però si s'exlouen el títol, la classificació i el director de la pel · lícula per considerar-se dades irrellevants en aquest estudi. A més a més, cada títol, i gairebé cada director, és únic al conjunt de dades, fet que comporta que no aportin valor afegit a l'anàlisi.

```
# Es selecciona el dataset
  names <- colnames(top250movies)
  exclude <- c('Title',
               'Classification',
               'Director')
  selection <- names[!(names %in% exclude)]

  top250movies <- subset(x = top250movies,
                         select = selection)
```

3 Neteja de les dades

3.1 Definició de funcions

Es defineixen les funcions que s'empraran en la neteja i visualització de les dades.

```
# Eliminació de patrons
  top250movies.remove <- function(column,
                                  pattern){
    input <- top250movies[[column]]
    result <- str_remove_all(string = input,
                             pattern = pattern)
    return(result)
  }

# Conversió del temps
  top250movies.time_conversion <- function(column){
    input <- top250movies[[column]]
    time <- c()
    for (i in 1:length(input)) {
      value <- str_split(string = input[i],
                         pattern = ' ')[[1]]
      hours <- 0
      minutes <- 0

      if (is.na(value[2])) {
        if (grepl(pattern = 'h',
                  x = value[1])) {
          hours <- as.numeric(str_remove_all(string = value[1],
                                             pattern = '[a-z]'))
        } else {
          minutes <- as.numeric(str_remove_all(string = value[1],
                                               pattern = '[a-z]'))
        }
      } else {
```

```
          hours <- as.numeric(str_remove_all(string = value[1],
                                             pattern = '[a-z]'))
          minutes <- as.numeric(str_remove_all(string = value[2],
                                             pattern = '[a-z]'))
          }
      time[i] <- hours*60 + minutes
      }
    return(time)
    }

# Conversió d'unitats
  top250movies.units_conversion <- function(column){
    input <- top250movies[[column]]
    units <- c()
    for(i in 1:length(input)){
      ifelse(test = grepl(pattern = 'K',
                          x = input[i]),
             yes = units[i] <- paste(str_remove_all(string = input[i],
                                                    pattern = '[^0-9]'),
                                     '000',
                                     sep = ''),
             no = ifelse(test = grepl(pattern = '\\.',
                                      x = input[i]),
                         yes = units[i] <- paste(str_remove_all(string = input[i],
                                                                pattern = '[^0-9]'),
                                                 '00000',
                                                 sep = ''),
                         no = units[i] <- paste(str_remove_all(string = input[i],
                                                               pattern = '[^0-9]'),
                                                '000000',
                                                sep = '')))
    }
    return(units)
  }

# Conversió d'unitats monetàries
  top250movies.dollar_conversion <- function(column,
                                             policy){
    input <- top250movies[[column]]
    conversion <- c()
    for (i in 1:length(input)) {
      ifelse(test = is.na(input[i]),
             yes = conversion[i] <- NA,
             no = conversion[i] <-
               subset(x = get(policy),
                      subset = Local == str_replace_all(string = input[i],
                                                        pattern = '[0-9]',
                                                        replacement = ''),
                      select = Dolar,
                      drop = TRUE) *
               as.numeric(str_replace_all(string = input[i],
                                          pattern = '[^0-9]',
                                          replacement = '')))
```

```
      }
    return(conversion)
    }

# Càlcul de la proporció de valors perduts
  top250movies.na_prop <- function(column){
    input <- top250movies[[column]]
    result <- cat(column,
                  ': ',
                  round(x = sum(is.na(input))/length(input),
                        digits = 3)*100,
                  '%',
                  '\n',
                  sep = '')
    return(result)
  }

# Detecció de patrons
  top250movies.detect <- function(column,
                                  pattern){
    input <- top250movies[[column]]
    result <- as.numeric(grepl(pattern = pattern,
                               x = input))
    return(result)
  }

# Valoració en termes de rendibilitat
  top250movies.is_profitable <- function(minimum_criteria){
    budget <- top250movies[['Budget']]
    collection <- top250movies[['Collection']]
    value <- collection - minimum_criteria * budget > 0
    result <- as.numeric(value)
    return(result)
  }

# Normalització
  top250movies.scale <- function(column){
    input <- top250movies[[column]]
    result <- (input- mean(input))/sd(input)
    return(result)
  }

# Creació de gràfics de dispersió:
  top250movies.kmeans_plot <- function(data,
                                       k_model,
                                       var1,
                                       var2,
                                       class){
    ggplot(data = data,
           aes(x = .data[[var1]],
               y = .data[[var2]],
               color = factor(k_model$cluster),
               shape = class)) +
```

```
      geom_point() +
      labs(title = paste(var1,
                         "vs",
                         var2),
           x = var1,
           y = var2,
           color = "Cluster",
           shape = class) +
      theme_minimal() +
      theme(text = element_text(size = 7),
            axis.title = element_text(size = 8),
            axis.text = element_text(size = 7),
            legend.title = element_text(size = 7),
            legend.text = element_text(size = 7),
            legend.position = "bottom")
  }

# Combinació de gràfics
  top250movies.combine_plots <- function(plots_list){
    plots <- get(plots_list)

    # Extracció de llegenda
    legend_plot <- cowplot::get_legend(plots[[1]])

    # Combinació de gràfics
    combined_plots <- cowplot::plot_grid(plotlist = lapply(plots,
                                                           function(p){
                                                             p + theme(legend.position = "none")
                                                             }),
                                         labels = NULL,
                                         ncol = 3)

    # Llegenda
    grid_plot <- cowplot::plot_grid(combined_plots,
                                    legend_plot,
                                    ncol = 1,
                                    rel_heights = c(10, 1))

    # Titol
    title <- ggdraw() +
      draw_label(plots_list,
      fontface = 'bold',
      x = 0,
      hjust = 0) +
      theme(plot.margin = margin(0, 0, 0, 7))

    # Result
    final_plot <- plot_grid(title,
                            grid_plot,
                            ncol = 1,
                            rel_heights = c(0.1, 1))
    return(final_plot)
  }
```

3.2 Eliminació de caràcters superflus

S'eliminen els caràcters innecessaris en les columnes del conjunt de dades.

```
# Budget
  top250movies$Budget <- top250movies.remove(column = 'Budget',
                                             pattern = '\\(estimated\\)|,|\\s')

# Collection
  top250movies$Collection <- top250movies.remove(column = 'Collection',
                                                 pattern = '[^0-9]')
```

3.3 Conversió

S'expressen d'acord a un mateix criteri els valors dels atributs requerits.

```
# Duration
  top250movies$Duration <- top250movies.time_conversion(column = 'Duration')

# Review
  top250movies$Review <- top250movies.units_conversion(column = 'Review')

# Budget
  # Es defineix la política de conversió a dòlar
  Monetary_policy <- data.frame('Local' = c('R$',
                                            'FRF',
                                            'DKK',
                                            'DEM',
                                            'A$',
                                            ' ',
                                            '€',
                                            '₩',
                                            '¥',
                                            '£',
                                            '$'),
                                'Dolar' = c(0.055,
                                            0.164261,
                                            0.14,
                                            0.0018,
                                            0.66,
                                            0.012,
                                            1.08,
                                            0.00073,
                                            0.0064,
                                            1.26,
                                            1))

  top250movies$Budget <- top250movies.dollar_conversion(column = 'Budget',
                                                        policy = 'Monetary_policy')
```

3.4 Canvi de tipus de dades

Es modifica la naturalesa dels atributs del conjunt de dades.

```
for (column in colnames(top250movies)) {

  # Variables de tipus integer
    if (column %in% c('Year',
                      'Duration',
                      'Review',
                      'Budget',
                      'Collection')) {
      top250movies[[column]] <- as.integer(top250movies[[column]])
    }

  # Variables de tipus numeric
    if (column == 'Rating') {
      top250movies[[column]] <- as.numeric(top250movies[[column]])
    }

}
```

3.5 Valors extrems

Es cerquen i suprimeixen els valors extrems per a les variables Budget i Collection. Totes les altres variables de caràcter numèric, com per exemple 'Rating', es considera que contenen valors legítims.

```
# Valors extrems
  for (column in colnames(top250movies)) {
    if (column %in% c('Budget',
                      'Collection')) {
      outlier <- boxplot(top250movies[[column]])$out
      top250movies[[column]][top250movies[[column]] %in% outlier] <- NA
      }
    }
```

3.6 Imputació

Primerament es cerca la proporció de valors perduts per a cada característica.

```
# Proporció de dades perdudes
  for (column in colnames(top250movies)) {
    top250movies.na_prop(column = column)
  }
```

```
## Genre: 0.4%
## Year: 0.4%
## Duration: 0%
## Rating: 0.4%
## Review: 0%
## Budget: 16%
## Collection: 17.6%
```

S'adverteix que la proporció de valors perduts no supera en cap cas el 18%. Per tant es procedeix amb la imputació per k-NN considerant a la resta de dades disponibles.

```
# Imputació per k-NN
  for (column in colnames(top250movies)) {
    if (anyNA(top250movies[[column]])) {
      top250movies <- kNN(data = top250movies,
                          variable = column,
                          metric = 'grower',
                          imp_var = FALSE)
    }
  }
```

3.7 Transformacions dicotòmiques

Es dicotomitza l'atribut 'Genre' i s'elimina la característica original.

```
# Genre
  genre_list <- unique(str_trim(unlist(strsplit(top250movies$Genre,
                                                split = ','))))
  genre_list <- genre_list[!is.na(genre_list)]

  for (genre in genre_list) {
    top250movies[[genre]] <- as.factor(top250movies.detect(column = 'Genre',
                                                           pattern = genre))
  }

  top250movies <- subset(x = top250movies,
                         select = -Genre)
```

3.8 Generació de noves característiques

En base a les dades ja existents, s'obté els beneficis de la pel · lícula i si és rendible només si 'Collection' és, almenys, 2 vegades superior a 'Budget'.

```
# Beneficis per pel·lícula
  top250movies$Net_income <- top250movies$Collection - top250movies$Budget

# Es comprova si la pel·lícula és rendible només si Collection és el doble que Budget
  top250movies$Profitable <- as.factor(top250movies.is_profitable(minimum_criteria = 2))
```

3.9 Normalització

```
# Es normalitzen els atributs numèrics
  for (column in colnames(top250movies)) {
    if ((class(top250movies[[column]]) %in% c('numeric',
                                              'integer')) &
        (column != 'Year')) {
      top250movies[[paste(column,
                          '_z-score',
                          sep = '')]] <- top250movies.scale(column)
    }
  }
```

3.10 Visualització i exportació del conjunt de dades tractat

Es presenta una instantànea del resultat final del procés de neteja i s'exporta.

```
# S'observa el resultat final
  head(x = top250movies,
       n = 3)
```

```
##   Year Duration Rating  Review   Budget Collection Drama Crime Action Biography
## 1 1994      142    9.3 2900000 25000000   28905764     1     0      0         0
## 2 1972      175    9.2 2000000  6000000  250342030     1     1      0         0
## 3 2008      152    9.0 2900000 93000000  538375067     1     1      1         0
##   History Adventure Western Romance Sci-Fi Fantasy Mystery Family Thriller War
## 1       0         0       0       0      0       0       0      0        0   0
## 2       0         0       0       0      0       0       0      0        0   0
## 3       0         0       0       0      0       0       0      0        0   0
##   Comedy Animation Music Horror Film-Noir Musical Sport Net_income Profitable
## 1      0         0     0      0         0       0     0    3905764          0
## 2      0         0     0      0         0       0     0  244342030          1
## 3      0         0     0      0         0       0     0  445375067          1
##   Duration_z-score Rating_z-score Review_z-score Budget_z-score
## 1        0.3978111       4.171914       3.954787     0.04087938
## 2        1.5009846       3.749827       2.341194    -0.62591814
## 3        0.7321061       2.905651       3.954787     2.42731263
##   Collection_z-score Net_income_z-score
## 1         -0.6850432         -0.7726885
## 2          0.5644160          0.7430101
## 3          2.1896491          2.0103126
```

```
# S'exporta top250movies
  write.csv(x = top250movies,
            file = 'dataset/top250movies_clean.csv',
            row.names = FALSE)
```

4 Modelització i anàlisi

4.1 Model supervisat

S'escull un mètode de classificació per al model supervisat. En concret s'implementa un arbre de decisió per a crear regles que determinin si una pel · lícula és profitosa depenent de les variables independents. Per a tal objectiu, es disposa de les instàncies correctament etiquetades a la variable de classe dicotòmica 'Profitable', generada a partir dels atributs 'Budget' i 'Collection'. S'estima que una pel · lícla **és profitosa, si la recaudació dobla al pressupost destinat per a la seva producció**.

4.1.1 Generació dels conjunts d'entrenament i test

Per a realitzar el model s'exclouen els camps 'Collection' i 'Budget' per estar funcionalment lligats amb la variable objectiu 'Profitable'.

Es registra l'extració en un nou dataset i, a continuació, es presenta un resum del subconjunt de dades:

```
# Es selecionen les característiques d'interès
  names <- colnames(top250movies)
  exclude <- c('Duration',
               'Budget',
               'Collection',
               'Net_income',
               'Duration_z-score',
               'Rating_z-score',
               'Review_z-score',
               'Budget_z-score',
               'Collection_z-score',
               'Net_income_z-score')
  selection <- names[!(names %in% exclude)]

  top250movies_id3b <- subset(x = top250movies,
                              select = selection)
```

Per validar l'arbre de decisió, és necessari dividir el conjunt de dades en un conjunt d'entrenament i un conjunt de test. El conjunt d'entrenament serà utilitzat per a construir i afinar el model, mentre que el conjunt de test servirà per avaluar la seva precisió.

```
# Es separa la variable de classe de la resta de la data
  set.seed(666)
  y <- subset(x = top250movies_id3b,
              select = Profitable,
              drop = TRUE)
  X <- subset(x = top250movies_id3b,
              select = -Profitable)
```

De manera dinàmica, separem les dades en funció del paràmetre *split_prop* amb un valor de 3:

```
# Es genera el conjunt d'entrenament i test
  split_prop <- 3
  indexes = sample(x = 1:nrow(top250movies_id3b),
                   size = floor(nrow(top250movies_id3b)*(split_prop-1)/split_prop))
```

```
  trainX <- X[indexes,]
  trainy <- y[indexes]
  testX <- X[-indexes,]
  testy <- y[-indexes]
```

4.1.2 Modelització i visualització

Per a crear l'arbre de decisió, es fa ús de la funció C5.0 de la llibreria C50. Els paràmetres que requereix la funció són: la matriu de camps predictius (el conjunt d'entrenament X) i el camp classificador (el conjunt d'entrenament y):

```
# Es genera el model
  model <- C50::C5.0(x = trainX,
                     y = trainy)
# Es grafica el model
  plot(model,
       gp = gpar(fontsize = 7.0))
```

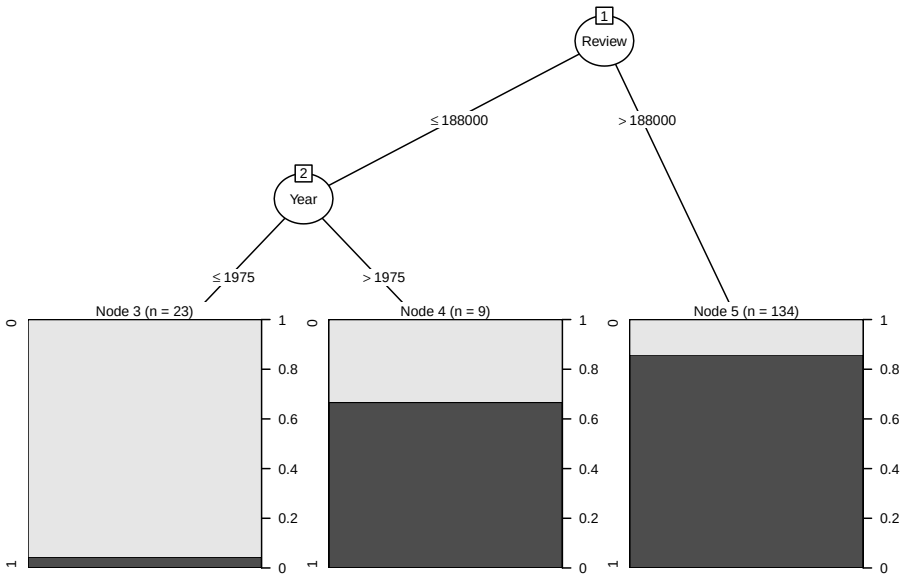


Figura 1: Arbre de decisió

De l'arbre resultant es conclou el següent:

- En un **92%** dels casos, si la pel · licula és igual o anterior al 1975, i les ressenyes son iguals o inferiors a les 188000, aleshores, **NO ÉS PROFITOSA**.
- En canvi, si la filmació és superior al 1975, en un **87%** dels casos la pel · lícula **ÉS PROFITOSA** amb independència del nombre de ressenyes.
- En el cas de superar les 188000 ressenyes, la pel · lícula **ÉS PROFITOSA** en un **85%** dels casos.

Com és comprovat, el conjunt de dades permet extreure conclusions molt valuoses que no es poden deduir fàcilment, doncs l'algoritme només classifica erròniament el 13.9% de la data.

4.2 Model No supervisat

Per a la construcció del model no supervisat s'ignora la variable de classe, que en el nostre conjunt correspont a la variable dicotòmica *Profitable*.

Com es desconeix d'inici el nombre d'agrupacions o classes naturals a les que pertanyen les instàncies, es fa ús del mètode k-means per agrupar les observacions segons les variables independents disponibles. Per a dur-ho a terme, s'avaluen, amb diferents nombre de clústers (k), les mètriques SSW (*Sum of Squared Within*), SSB (*Sum of Squared Between*) i el coeficient de *Silhouette*.

- **SSW** - Homogeneïtat entre grups. Minimització de distàncies intragrup.
- **SSB** - Heterogeneïtat entre grups. Maximització de distàncies intragrup.
- **Coeficient de Silhouette** - Intervals que indiquen si la mostra està al grup correcte.

4.2.1 Modelització

Es sap que el nombre correcte de k és 2 perquè es treballa amb una variable dicotòmica. Per a procedir amb l'aplicació el mètode *k-means*, primerament es cerca el nombre de clústers òptim.

```
# Es seleccionen les variables d'interès
  top250movies.all <- subset(x = top250movies,
                             select = c('Profitable',
                                        'Rating_z-score',
                                        'Review_z-score',
                                        'Budget_z-score',
                                        'Collection_z-score'))
  top250moviesNormalitzada <- subset(x = top250movies.all,
                                     select = -Profitable)

# S'avaluen k de 1 a 10
  distance <- cluster::daisy(top250moviesNormalitzada)
  valores <- seq(from=1,
                 to=10,
                 by=1)

# Resultats de les mètriques
  resultats_ssw <- rep(NA, length(valores))
  resultats_ssb <- rep(NA, length(valores))
  resultats_silhouette <- rep(NA, length(valores))

  for (k in valores[-1]) {
    set.seed(123)
    model <- stats::kmeans(x = top250moviesNormalitzada,
                           centers = k)
    clusters <- model$cluster
    resultats_silhouette[k] <- summary(cluster::silhouette(x = clusters,
                                                           dist = distance))$avg.width
    resultats_ssw[k] <- model$tot.withinss
    resultats_ssb[k] <- model$betweenss
  }
  resultats_kmeans <- data.frame(valores,
                                 resultats_silhouette,
                                 resultats_ssw,
                                 resultats_ssb) %>% tidyr::drop_na()
```

4.2.2 Visualització

Es generen els gràfics per a visualitzar les mètriques registrades *Coeficient de Silhouette*, *SSW* i *SSB* per a interpretar el nombre idoni de clústers.

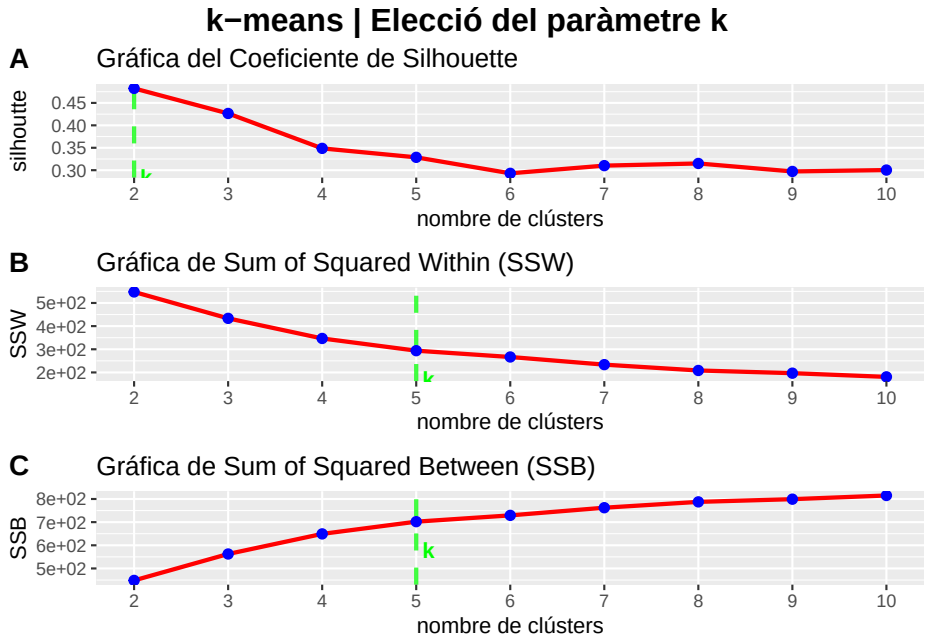


Figura 2: Mètriques k-mean

A. *Coeficient de Silhouette*: la gràfica mostra com s'ajusta cada punt al clúster assignat. Un valor alt indica que els punts estàn ajustats al seu propi grup i lluny dels veins. S'observa un pic més pronunciat per a k=2 que apunta a que l'agrupació de 2 classes té un ajustament més adequat en comparació amb els altres valors de k.

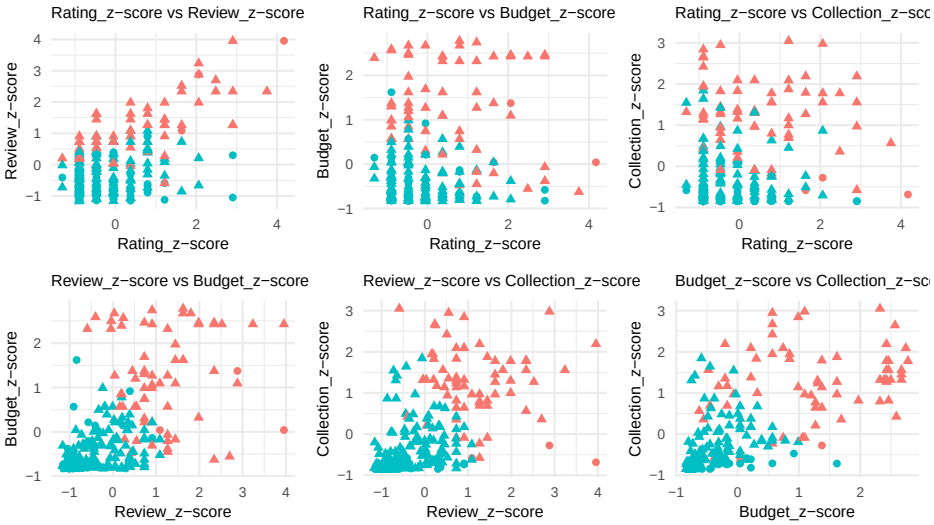
B. *SSW*: la gràfica mesura les distàncies al quadrat de cada punt en un clúster al centroide del grup, indicador de la cohesió del clúster. Un valor menor indica que els punts estan més a prop del centroide, fet desitjable. D'acord amb el gràfic, el *SSW* disminueix significativament fins a k=5, i a partir d'aquí, els canvis són menys pronunciats, i no redueix substancialment la variació dins els clústers.

C. *SSB*: la gràfica mesura la suma de les distàncies al quadrat entre els centroides dels clústers i el centroide global de tots els punts, reflectint la separació entre clústers. Un valor alt indica que els clústers estan més dispersos, i per tant millor definits. Al gràfic s'observa un agument a cada increment de k, però igual que passa amb el *SSW*, el canvi és menys pronunciat després de k = 5. Això suggereix la millora no és tan evident després d'arribar als 5 clústers.

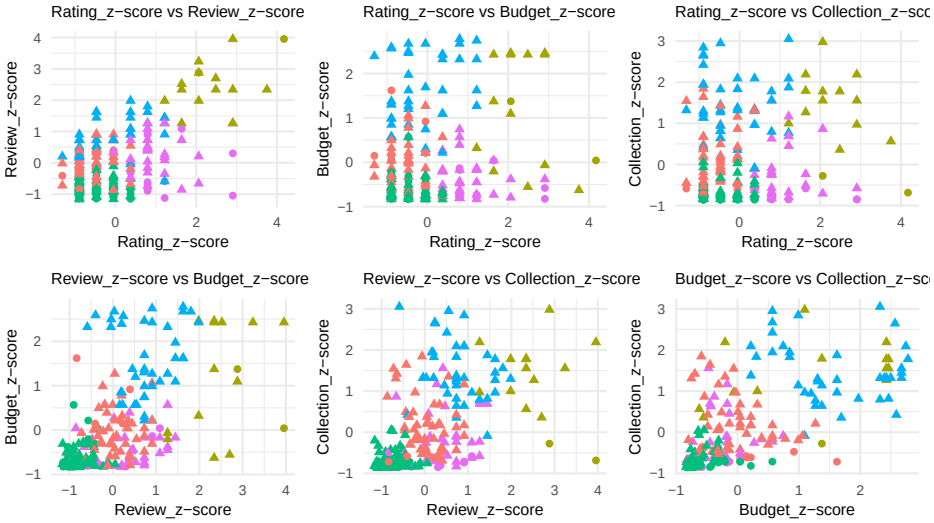
Sorprent els resultats de *SSW* i *SSB*, degut a que el nombre esperat de clústers és 2. Per entendre millor aquests resultats, visualitzem a continuació les agrupacions per a k = 2 i K = 5 en les diverses combinacions de variables disponibles.

Primer, es crea el model *k-means* per cada valor de k desitjat, i a continuació, s'empra la funció per a graficar les relacions entre parells de variables per a k.

plots_k2



plots_k5



Com s'observa, el nombre de clústers més adient és k = 2. Per a k = 5 hi ha classes que queden superposades. L'elecció de 2 clústers s'alinea amb la recomanació del coeficient de *Silhouette* que es tracta d'un indicador robust de la qualitat de l'agrupament, i encara que hi ha disminucions marginals per a SSW i augments marginals per a SSB després de k = 2, no es justifica l'elecció d'un major nombre de clústers.

4.3 Contrast d'hipòtesi

Es vol estudiar si la proporció de pel · lícules rendibles és major quan el pressupost d'aquestes és superior a la mitjana, *high*, al 99% de confiança.

Taula 1: Contrast

	Low	High	Sum
0	57	11	68
1	107	75	182
Sum	164	86	250

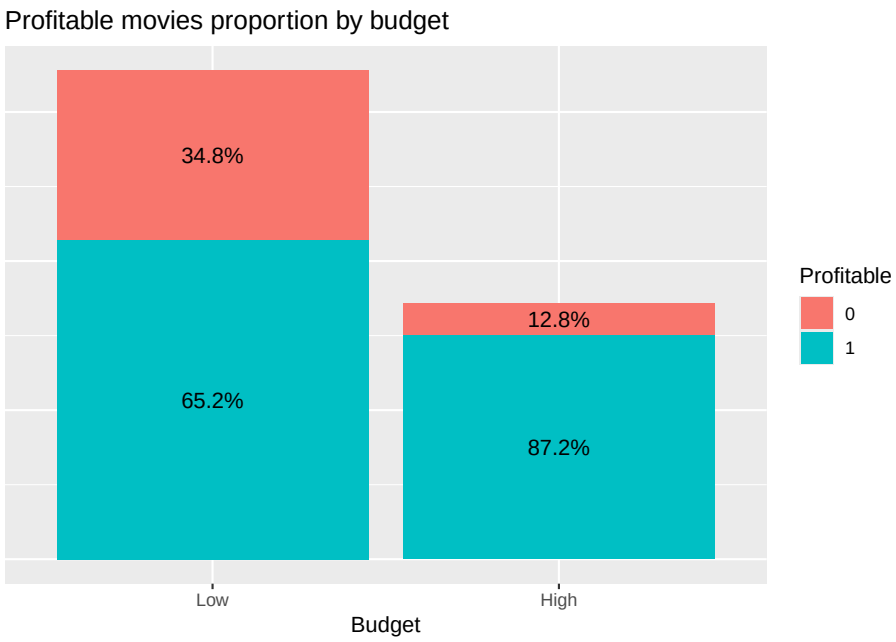


Figura 3: Proporció de pel · lícules rendibles per pressupost

4.3.1 Hipòtesi nul · la i alternativa

$H_0 : \frac{p_1}{p_2} = 1$

$H_1 : \frac{p_1}{p_2} > 1$

$p_1 :=$ proporció problacional de pel · lícules rendibles amb pressupost superior o igual a la mitjana mostral

$p_2 :=$ proporció problacional de pel · lícules rendibles amb pressupost inferior a la mitjana mostral

4.3.2 Test estadístic

El test que s'empra en aquest apartat és el contrast de la proporció, o el contrast de dades categòriques corregit, de dues mostres per a la cua superior on es compleix que, d'acord a la taula i sota H_0:

$$x^{2*} = \sum_i \sum_j \frac{(|O_{ij} - E_{ij}| - 0.5)^2}{E_{ij}} \sim X_1^2$$

4.3.3 Contrast

```r
# Es calculen els paràmetres
  group_1 <- subset(x = top250movies,
                    subset = Budget >= mean(Budget),
                    select = Profitable,
                    drop = TRUE)
  group_2 <- subset(x = top250movies,
                    subset = Budget < mean(Budget),
                    select = Profitable,
                    drop = TRUE)

  x_1 <- sum(as.numeric(group_1) - 1)
  x_2 <- sum(as.numeric(group_2) - 1)

  n_1 <- length(group_1)
  n_2 <- length(group_2)

# Es calcula el test
  test <- prop.test(x = c(x_1, x_2),
                    n = c(n_1, n_2),
                    alternative = 'greater',
                    conf.level = 0.99)

# S'obté l'estadístic
  x2 <- test$statistic[[1]]
```

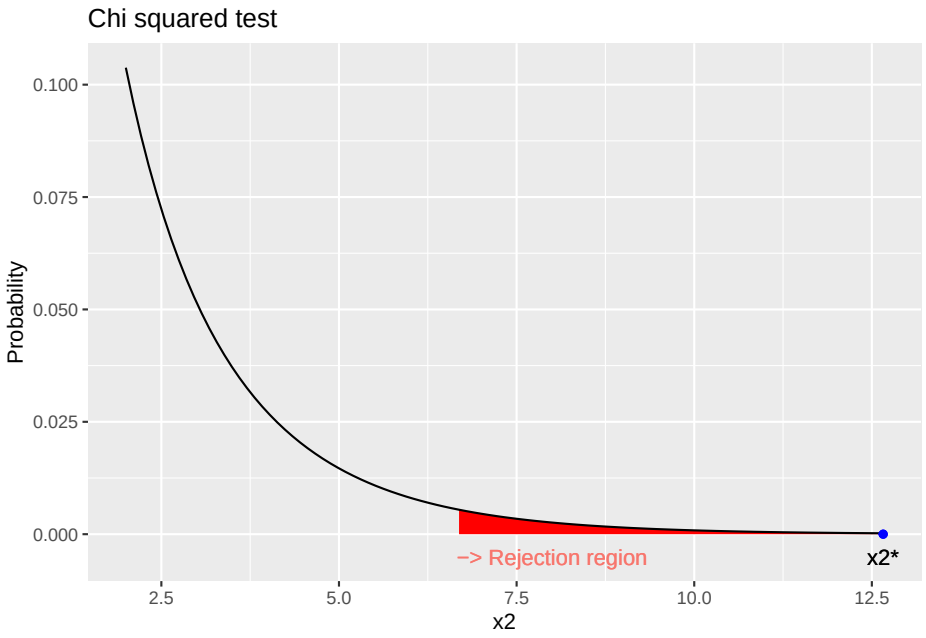


Figura 4: Contrast de cua superior de la distribució Chi quadrat

4.3.4 Interpretació

El resultat obtingut permet rebutjar H_0 amb $\alpha = 0.01$.

S'observa en el gràfic que x^{2*} és troba situat dins la zona de rebuig, és a dir, que és superior al valor crític en aquest cas particular i, en conseqüència, amb les dades disponibles, es demostra estadísticament que la proporció de pel · lícules rendibles és major en el conjunt de metratges que sobrepassen el pressupost promig respecte als altres.

5 Resolució del problema.

Després de treballar i analitzar el conjunt de dades referents a les 250 més ben valorades pels usuaris d'IMDb, es conclou que la condició de rendibilitat d'una pel · lícula és pot expressar prou bé en funció de l'any i les ressenyes, d'acord al model supervisat, i també en funció de si el pressupost és elevat o no, segons el constrast estadístic.

Es considera que una pel · lícula és rendible si els ingressos (Collection) doblen el pressupost (Budget) de la mateixa. La creació d'aquesta característica resulta convenient per aquest estudi doncs, principalment, permet l'elaboració de models supervisats, els quals ajuden a predir si un metratge resultarà exitos en termes econòmics.

Per respondre a la pregunta inicial, *"quina relació guarda la rendibilitat de les pel · lícules amb les altres característiques del dataset?"* es presenten les següents conclusions:

- Els projectes que no superen les 188000 ressenyes i són anteriors a l'any 1976 no assoleixen rendibilitat. En canvi, si ho fan les pel · licules posteriors al 1975.

Possiblement d'aquí bé l'expressió del cinema com a 7è art, inicialment, a l'hora de produir una pel · lícula, la recaudació no esdevenia una prioritat sinó que, més aviat, es cercava transmitir un missatge.

- Metratges amb ressenyes superiors a les 188000 unitats tendeixen a esdevenir profitoses en un 85% de les observacions del conjunt de dades.

D'acord a l'anterior s'observa una relació evident entre la condició de rendibilitat amb el nombre de ressenyes dels usuaris.

- La proporció de pel · lícules rendibles és estadísticament superior en el conjunt de filmacions amb un pressupost superior al promig.

Aquest detall sembla suggerir que les pel · lícules d'acció o ciència ficció tendeixen a ser més rendibles que d'altres gèneres. Però aquesta hipòtesi caldrà contrastar-la en un altre moment.

6 Codi font

El codi R i els datasets estan disponibles al respositori de GitHub.[1]

7 Vídeo

El vídeo amb la presentació d'aquesta pràctica es troba disponible al repositori del drive amb el nom M2.951_Farran_Eric_i_Alvarez_Jordi_PR2_Video.mp4.[2]

[1] https://github.com/efarran0/TiCdV_DADES_PR2

[2] https://drive.google.com/drive/folders/1xdBgqwmP2MLW0iR6CtHhKRgyHWnQYR7J?usp=drive_link

Taula 2: Contribucions

Contribucions	**Signatura**
Investigació prèvia	EFM, JAP
Redacció de respostes	EFM, JAP
Desenvolupament del codi	EFM, JAP
Participació al vídeo	EFM, JAP