UNIVERSIDAD DE GRANADA

Contraconceptive Method Choice

Pablo Alfaro Goicoechea, Carlos Morales Aguilera, Carlos S. Sánchez Muñoz

16/12/2020

Presentación del problema

El problema escogido se puede encontrar en **Kaggle**, en este enlace. El problema es un conjunto de datos de la encuesta que se realizó en 1987 en Indonesia acerca del método anticonceptivo escogido por las mujeres. En este problema existen factores como pueden ser la educación de los integrantes del matrimonio, orientación religiosa, familia y otros factores que consideramos interesantes.

Se trata de un problema cuyo interés parte en el artículo A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. Se ha investigado previamente el estado del arte del problema, y se han revisado diversas implementaciones realizadas por otros científicos y estudiantes con el objetivo de observar los resultados obtenidos y los planteamientos realizados. Un ejemplo serían competiciones como Supervised Classification on cmc. Los resultados observados tienden a obtener por media un acierto del *50%-60%*.

El objetivo de esta práctica es ver como varían los diferentes modelos con un conjunto de datos sencillo y pocas instancias, pero a que su vez se encuentra muy desbalanceado. Los pasos a seguir serán los siguientes:

1. Leer los datos correctamente.
2. Análisis exploratorio de los datos.
3. Preprocesamiento de los datos.
4. Clasificación con diversos modelos.
5. Análisis de resultados.
6. Alternativas de planteamiento del problema.
7. Conclusiones finales sobre el problema.

Lectura de datos

El primer paso en el problema es leer correctamente los datos, para ello se utiliza la función `read.csv` teniendo en cuenta que no posee nombre de columnas, por lo que se asignará la nomenclatura *V1* para la variable 1 del conjunto.

```
f <- "C:\\Users\\power\\Desktop\\TID\\PracticaGrupal\\cmc.data.txt"
# Read dataframe
bd_cmc <- read.csv(f, header=FALSE)
# Adjust column names
names(bd_cmc) <- c('WifeAge', 'WifeEducation', 'HusbandEducation',
                   'Children', 'WifeReligion', 'WifeWorking',
                   'HusbandOcupation', 'StandardOfLiving',
                   'MediaExposure', 'ContraconceptiveMethod')
```

Una de las buenas prácticas es la de eliminar valores perdidos, por lo que previamente a tratar con los datos, nos aseguramos de eliminar los posibles valores perdidos si hubieran. En la documentación se indica que no existen valores perdidos, por lo que realmente no es necesario, aunque se realizará la operación para confirmarlo.

```
# Omit NAs
bd_cmc <- na.omit(bd_cmc)
```

Se puede observar que finalmente no existen valores perdidos, ya que el tamaño del conjunto de datos es el mismo.

Tras cargar los datos, visualizamos cuantas instancias hay de cada clase de método anticonceptivo utilizando para ello la última variable del conjunto de datos, referente al método anticonceptivo utilizado.

```r
# Group by class
classes <- table(bd_cmc$ContraconceptiveMethod)
classes
```

```
##
##   1   2   3
## 629 333 511
```

Análisis exploratorio de los datos

El primer paso es realizar una análisis exploratorio inicial de los datos, para ello visualizamos un resumen inicial de los mismos:

```r
summary(bd_cmc)
```

```
##     WifeAge      WifeEducation   HusbandEducation    Children
##  Min.   :16.00   Min.   :1.000   Min.   :1.00     Min.   : 0.000
##  1st Qu.:26.00   1st Qu.:2.000   1st Qu.:3.00     1st Qu.: 1.000
##  Median :32.00   Median :3.000   Median :4.00     Median : 3.000
##  Mean   :32.54   Mean   :2.959   Mean   :3.43     Mean   : 3.261
##  3rd Qu.:39.00   3rd Qu.:4.000   3rd Qu.:4.00     3rd Qu.: 4.000
##  Max.   :49.00   Max.   :4.000   Max.   :4.00     Max.   :16.000
##   WifeReligion     WifeWorking     HusbandOcupation StandardOfLiving
##  Min.   :0.0000   Min.   :0.0000   Min.   :1.000    Min.   :1.000
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:1.000    1st Qu.:3.000
##  Median :1.0000   Median :1.0000   Median :2.000    Median :3.000
##  Mean   :0.8506   Mean   :0.7495   Mean   :2.138    Mean   :3.134
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:3.000    3rd Qu.:4.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :4.000    Max.   :4.000
##  MediaExposure   ContraconceptiveMethod
##  Min.   :0.000   Min.   :1.00
##  1st Qu.:0.000   1st Qu.:1.00
##  Median :0.000   Median :2.00
##  Mean   :0.074   Mean   :1.92
##  3rd Qu.:0.000   3rd Qu.:3.00
##  Max.   :1.000   Max.   :3.00
```

Como hemos podido observar en las documentaciones estudiadas, se trata de un conjunto de datos desbalanceado, lo cual se puede apreciar directamente en el análisis inicial, para corroborar este hecho se comprueba si la clase está balanceada con la función is.pbalanced del paquete plm:

```r
is.pbalanced(bd_cmc)
```

```
## [1] FALSE
```

Lo primero que queremos observar en los datos es la distribución de los niveles de educación, ya que consideramos que es uno de los puntos principales de cara a esta posible decisión a priori.

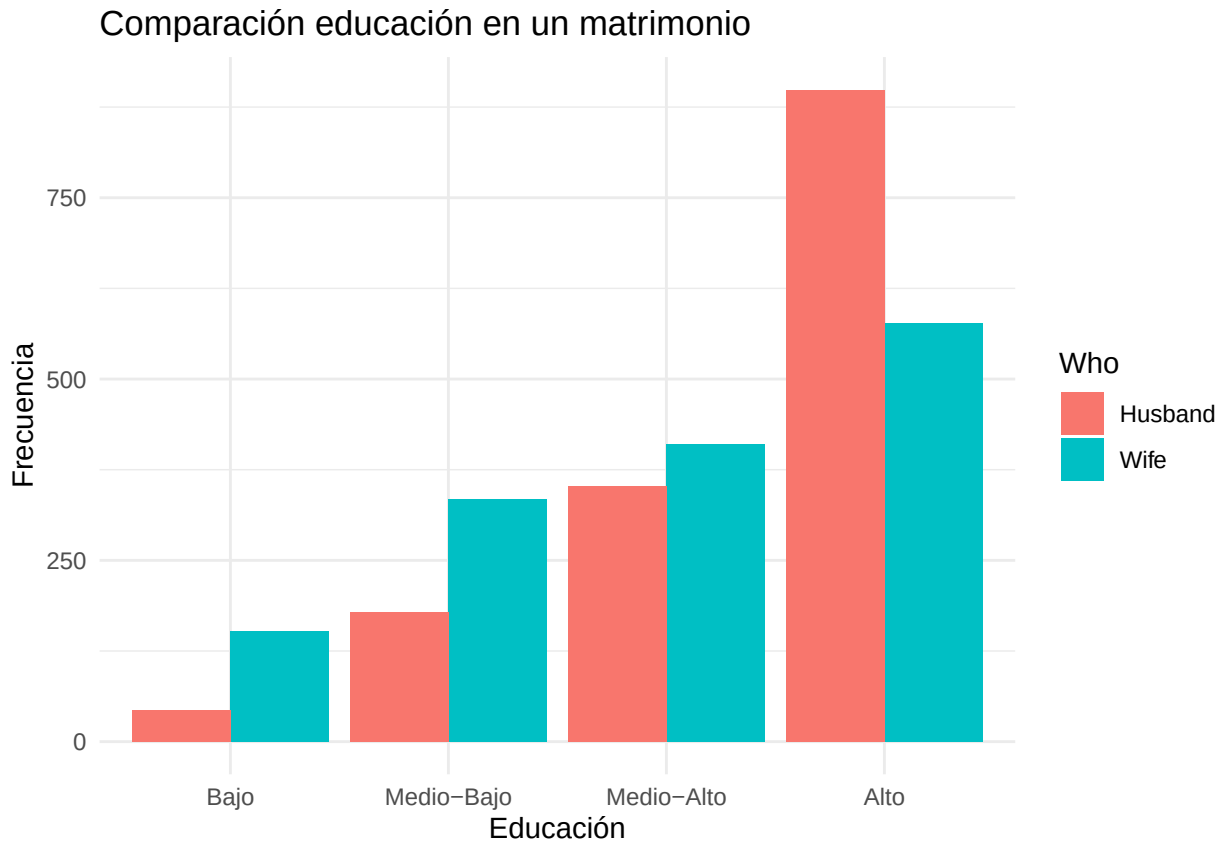
```r
# Count number of husbands, grouping by education
groupHusbands <- bd_cmc %>% group_by(HusbandEducation)
countHusbands <- groupHusbands %>% summarise(count = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# Count number of wifes, grouping by education
groupWifes <- bd_cmc %>% group_by(WifeEducation)
countWifes <- groupWifes %>% summarise(count = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

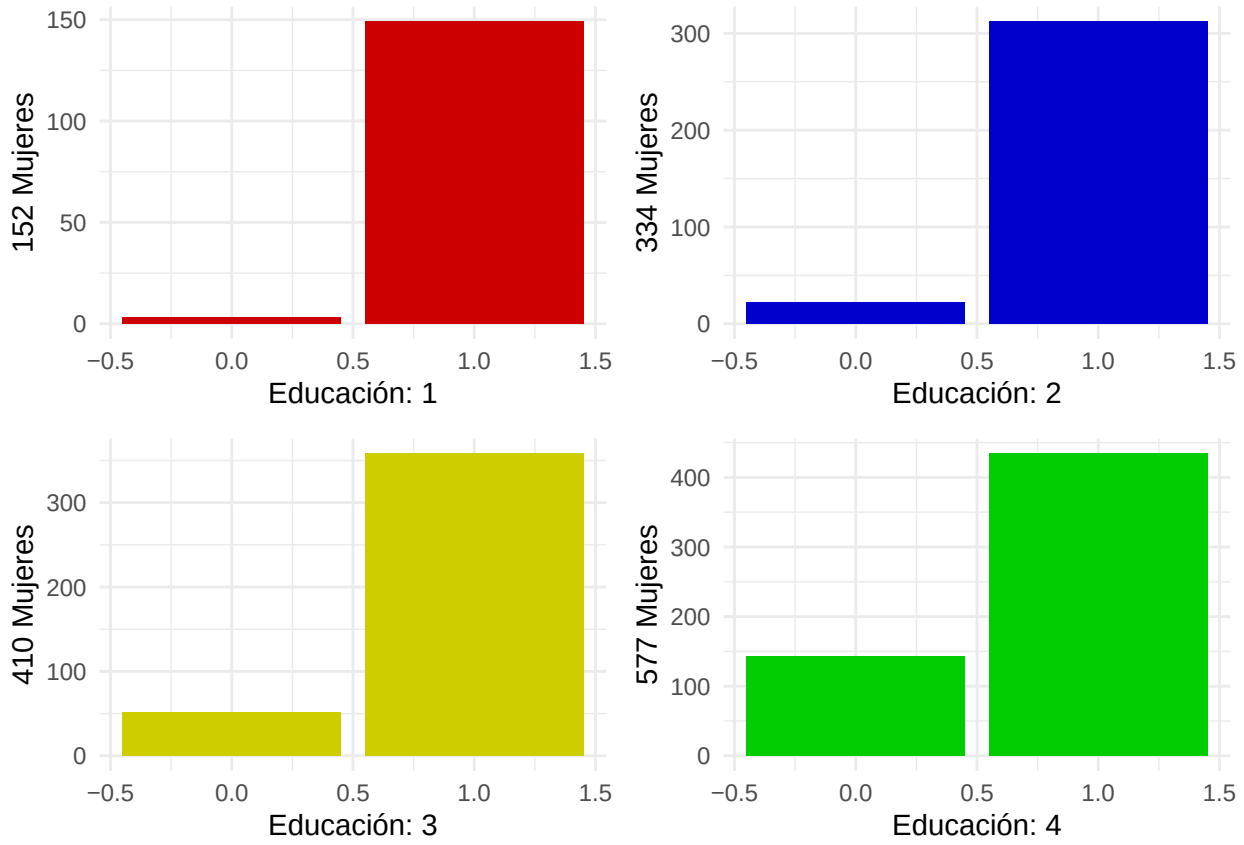
```
countHusbands$Who <- "Husband"
colnames(countHusbands) <- c("Education", "Count", "Who")

countWifes$Who <- "Wife"
colnames(countWifes) <- c("Education", "Count", "Who")

countsComb <- rbind(countWifes, countHusbands)

ggplot(countsComb, aes(x = Education, y = Count, fill = Who)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(limits =  c("Bajo", "Medio-Bajo", "Medio-Alto", "Alto")) +
  labs(x = "Educación", y = "Frecuencia",
       title = "Comparación educación en un matrimonio") +
  theme_minimal()
```



Se puede observar claramente que no sigue una distribución, sino que la educación no es equitativa en estos casos.

A continuación se ha decidido realizar una comparativa que relacione la cantidad de mujeres religiosas existen frente a las que no, para cada nivel de educación considerado:

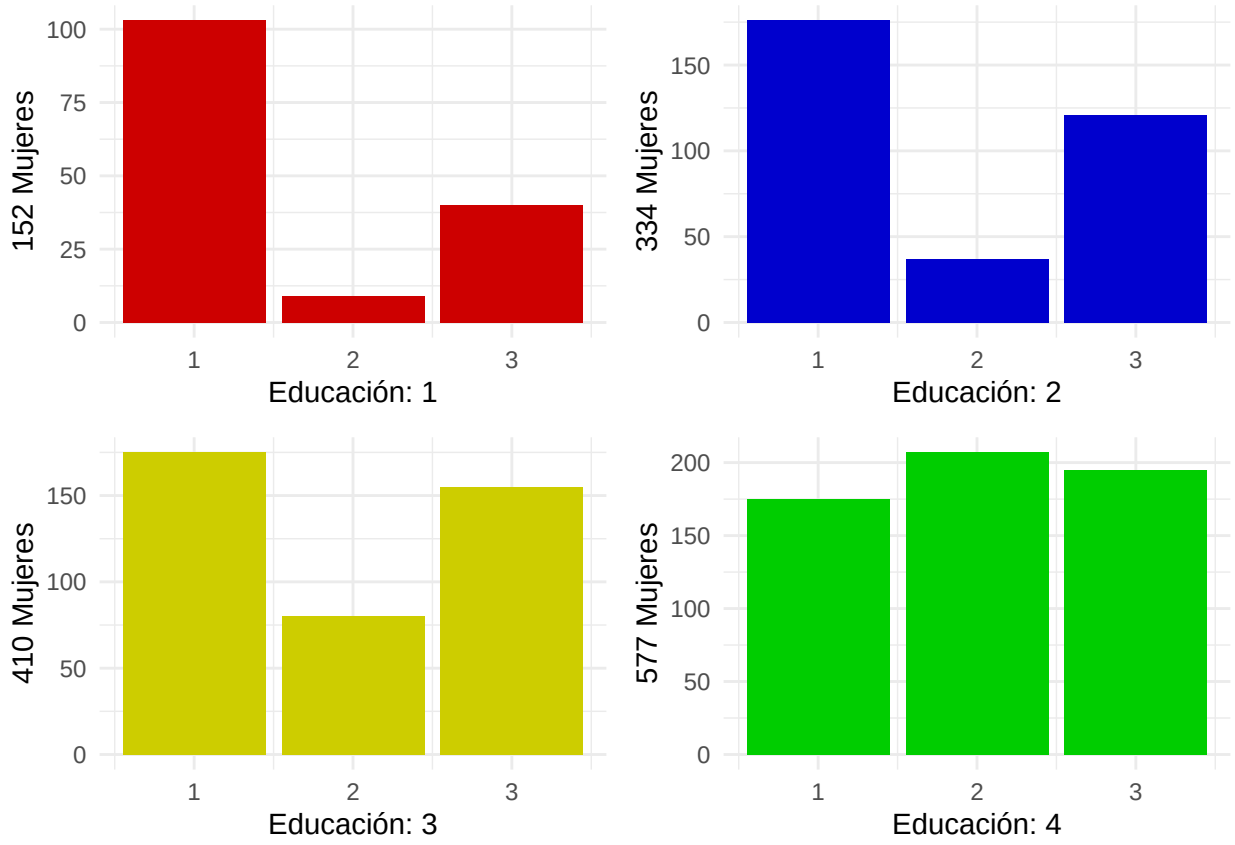
```
# Group wifes by education
groupWifes_1 <- bd_cmc[which(bd_cmc$WifeEducation == 1),]
groupWifes_2 <- bd_cmc[which(bd_cmc$WifeEducation == 2),]
groupWifes_3 <- bd_cmc[which(bd_cmc$WifeEducation == 3),]
groupWifes_4 <- bd_cmc[which(bd_cmc$WifeEducation == 4),]

# Basic bar plot of WifeReligion
# Education 1
plot1 <- ggplot(groupWifes_1, aes(x=WifeReligion)) +
  geom_bar( fill="red3") +
  labs(x = "Educación: 1",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Education 2
plot2 <- ggplot(groupWifes_2, aes(x=WifeReligion)) +
  geom_bar( fill="blue3") +
  labs(x = "Educación: 2",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
  theme_minimal()

# Education 3
plot3 <- ggplot(groupWifes_3, aes(x=WifeReligion)) +
  geom_bar( fill="yellow3") +
  labs(x = "Educación: 3",
       y = paste(nrow(groupWifes_3), "Mujeres"))  +
  theme_minimal()

# Education 4
plot4 <- ggplot(groupWifes_4, aes(x=WifeReligion)) +
  geom_bar( fill="green3") +
  labs(x = "Educación: 4",
       y = paste(nrow(groupWifes_4), "Mujeres"))  +
  theme_minimal()

# Plot grid with four plots
grid.arrange(plot1, plot2, plot3, plot4, ncol=2, nrow=2)
```



Se puede observar, que la proporción de mujeres creyentes es considerablemente superior al de mujeres no creyentes, y se puede observar además que cuanto menor es el nivel de educación, mayor la proporción de mujeres creyentes, por lo que existe una relación lógica entre ambas.

```
# Group wifes by education
groupWifes_1 <- bd_cmc[which(bd_cmc$WifeEducation == 1),]
groupWifes_2 <- bd_cmc[which(bd_cmc$WifeEducation == 2),]
groupWifes_3 <- bd_cmc[which(bd_cmc$WifeEducation == 3),]
groupWifes_4 <- bd_cmc[which(bd_cmc$WifeEducation == 4),]

# Basic bar plot of Contraconceptive Method
# Education 1
plot1 <- ggplot(groupWifes_1, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="red3") +
  labs(x = "Educación: 1",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Education 2
plot2 <- ggplot(groupWifes_2, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="blue3") +
  labs(x = "Educación: 2",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
  theme_minimal()

# Education 3
```

```r
plot3 <- ggplot(groupWifes_3, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="yellow3") +
  labs(x = "Educación: 3",
       y = paste(nrow(groupWifes_3), "Mujeres"))  +
  theme_minimal()

# Education 4
plot4 <- ggplot(groupWifes_4, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="green3") +
  labs(x = "Educación: 4",
       y = paste(nrow(groupWifes_4), "Mujeres"))  +
  theme_minimal()

# Plot grid with four plots
grid.arrange(plot1, plot2, plot3, plot4, ncol=2, nrow=2)
```

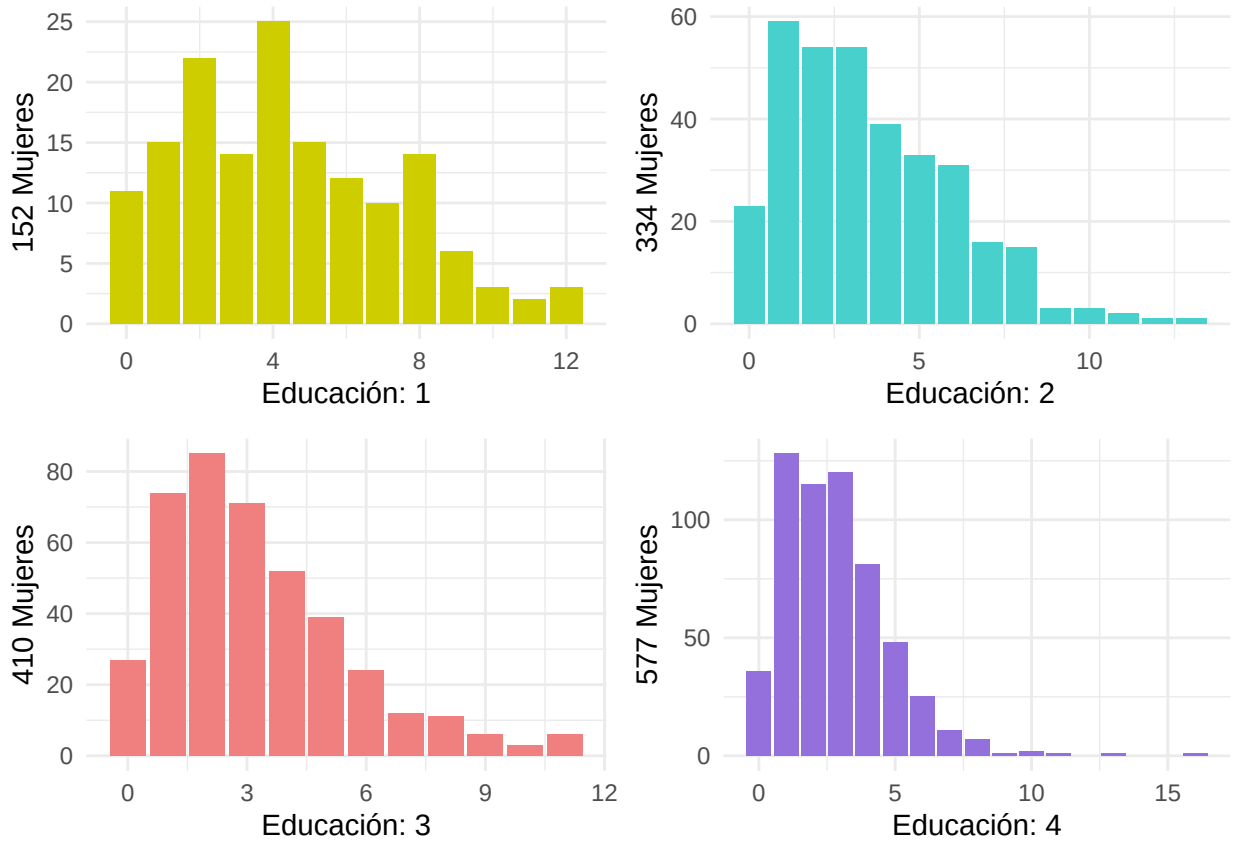


Tal y como se puede contemplar, también existe una relación entre la educación y el método anticonceptivo escogido, ya que los métodos a largo plazo son más utilizados cuanto mayor es la educación, y en general la proporción de utilización de métodos anticonceptivos crece proporcionalmente a la educación de la mujer.

Tras observar esta información, cabría esperar que el número de hijos por familia fuera menor cuanto mayor es la educación de la mujer, ya que está comprobado que esta por media utiliza más anticonceptivos. A continuación comprobamos dicha teoría:

```r
# Group wifes by education
groupWifes_1 <- bd_cmc[which(bd_cmc$WifeEducation == 1),]
```

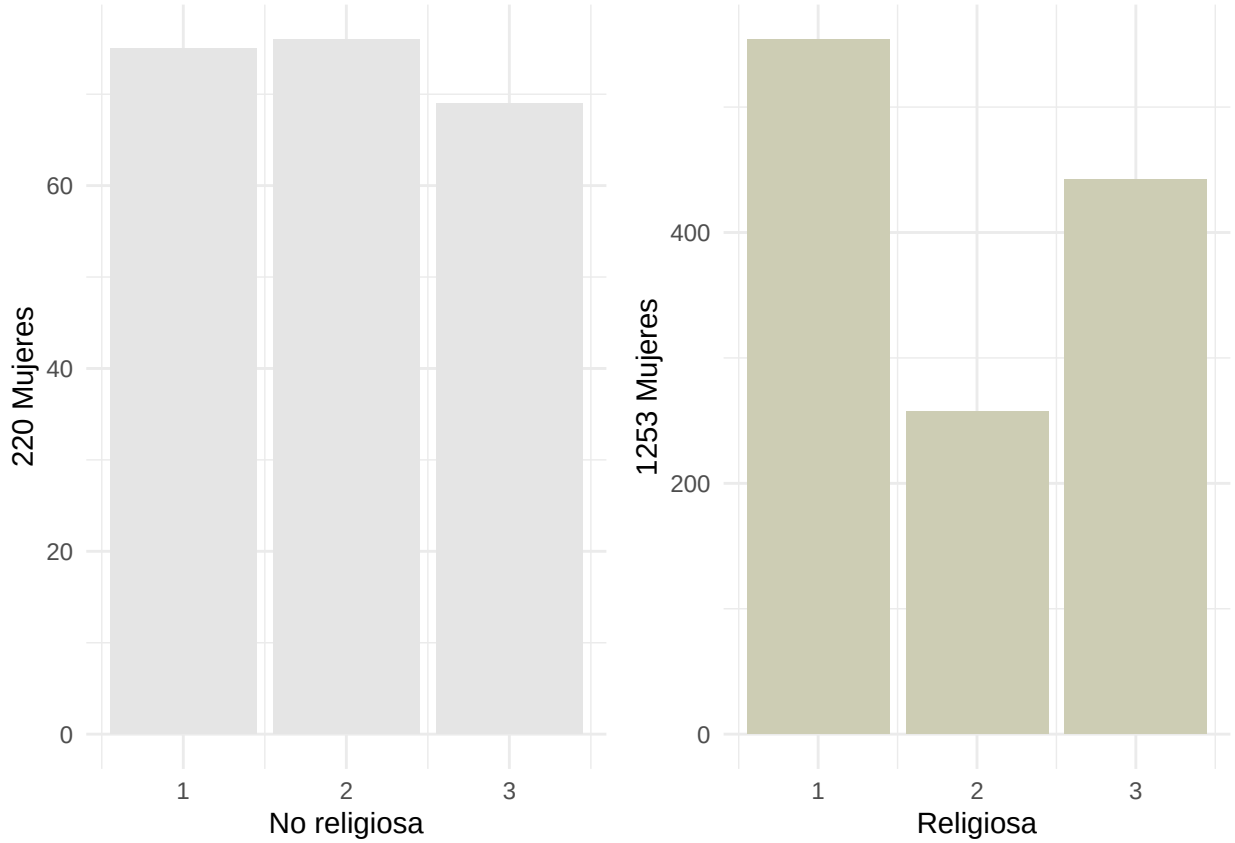
```r
groupWifes_2 <- bd_cmc[which(bd_cmc$WifeEducation == 2),]
groupWifes_3 <- bd_cmc[which(bd_cmc$WifeEducation == 3),]
groupWifes_4 <- bd_cmc[which(bd_cmc$WifeEducation == 4),]

# Basic bar plot of Contraconceptive Method
# Education 1
plot1 <- ggplot(groupWifes_1, aes(x=Children)) +
  geom_bar( fill="yellow3") +
  labs(x = "Educación: 1",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Education 2
plot2 <- ggplot(groupWifes_2, aes(x=Children)) +
  geom_bar( fill="mediumturquoise") +
  labs(x = "Educación: 2",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
  theme_minimal()

# Education 3
plot3 <- ggplot(groupWifes_3, aes(x=Children)) +
  geom_bar( fill="lightcoral") +
  labs(x = "Educación: 3",
       y = paste(nrow(groupWifes_3), "Mujeres"))  +
  theme_minimal()

# Education 4
plot4 <- ggplot(groupWifes_4, aes(x=Children)) +
  geom_bar( fill="mediumpurple") +
  labs(x = "Educación: 4",
       y = paste(nrow(groupWifes_4), "Mujeres"))  +
  theme_minimal()

# Plot grid with four plots
grid.arrange(plot1, plot2, plot3, plot4, ncol=2, nrow=2)
```



Se puede ver que se confirma nuestra teoría, y que las familias con una mujer con mayor educación, tienden a ser menos numerosas.

A continuación, la idea es ver como influye la religión en el método anticonceptivo utilizado, para ello se visualizará la elección en base a si es religiosa la mujer o no:

```
# Group wifes by religion
groupWifes_1 <- bd_cmc[which(bd_cmc$WifeReligion == 0),]
groupWifes_2 <- bd_cmc[which(bd_cmc$WifeReligion == 1),]

# Basic bar plot of Contraconceptive Method
# Non-Religious
plot1 <- ggplot(groupWifes_1, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="gray90") +
  labs(x = "No religiosa",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Religious
plot2 <- ggplot(groupWifes_2, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="lightyellow3") +
  labs(x = "Religiosa",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
  theme_minimal()

# Plot grid with two plots
grid.arrange(plot1, plot2, ncol=2)
```

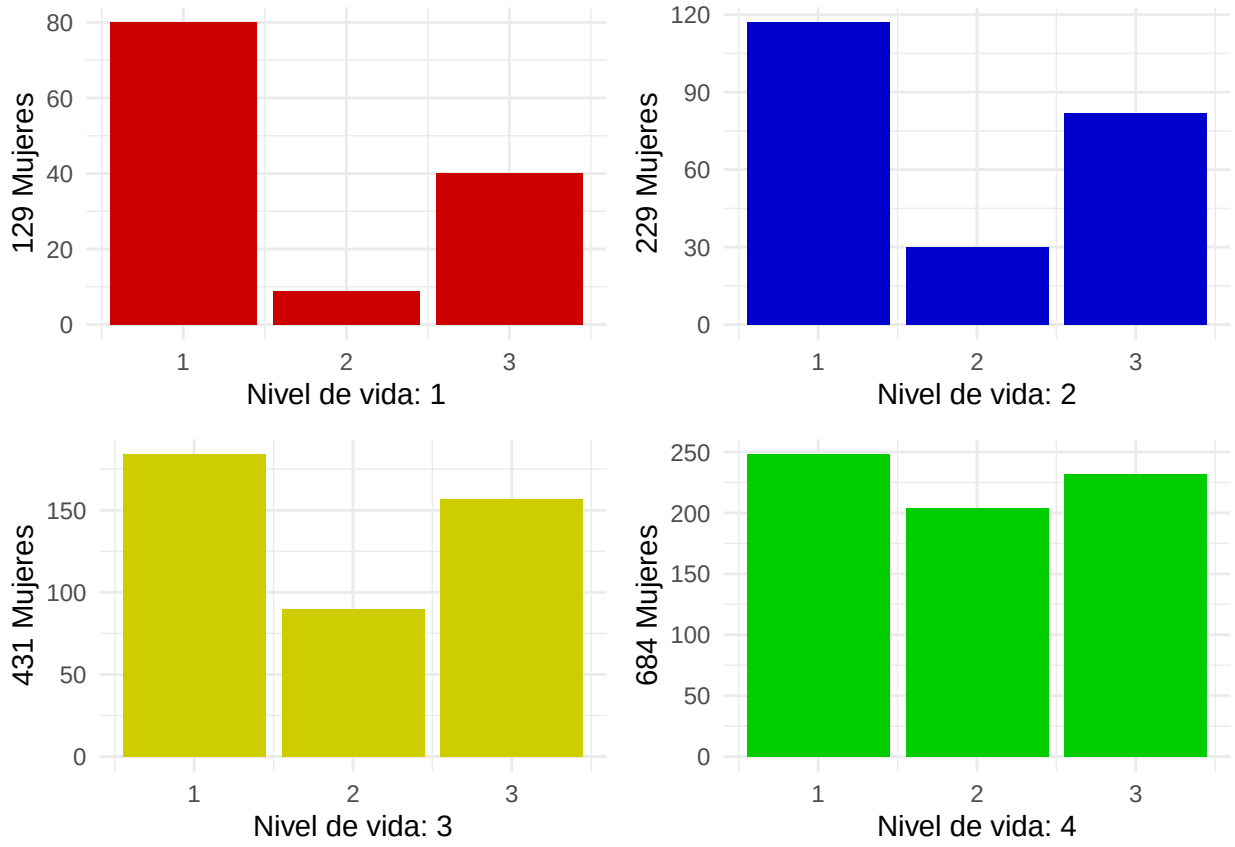


Se puede observar que cuando una mujer es religiosa, tiende a no utilizar anticonceptivos, evitando sobre todo los de largo plazo, mientras que en las mujeres religiosas, no existe realmente nada que indique una preferencia, por lo que obtenemos como información que la mujer religiosa tiende a no utilizar como primera elección.

Ahora, procedemos a ver como afecta realmente el nivel de vida en cuanto a la elección de método anticonceptivo:

```
# Group wifes by standard of living
groupWifes_1 <- bd_cmc[which(bd_cmc$StandardOfLiving == 1),]
groupWifes_2 <- bd_cmc[which(bd_cmc$StandardOfLiving == 2),]
groupWifes_3 <- bd_cmc[which(bd_cmc$StandardOfLiving == 3),]
groupWifes_4 <- bd_cmc[which(bd_cmc$StandardOfLiving == 4),]

# Basic bar plot of Contraconceptive Method
# Standard of Living 1
plot1 <- ggplot(groupWifes_1, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="red3") +
  labs(x = "Nivel de vida: 1",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Standard of Living 2
plot2 <- ggplot(groupWifes_2, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="blue3") +
  labs(x = "Nivel de vida: 2",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
```

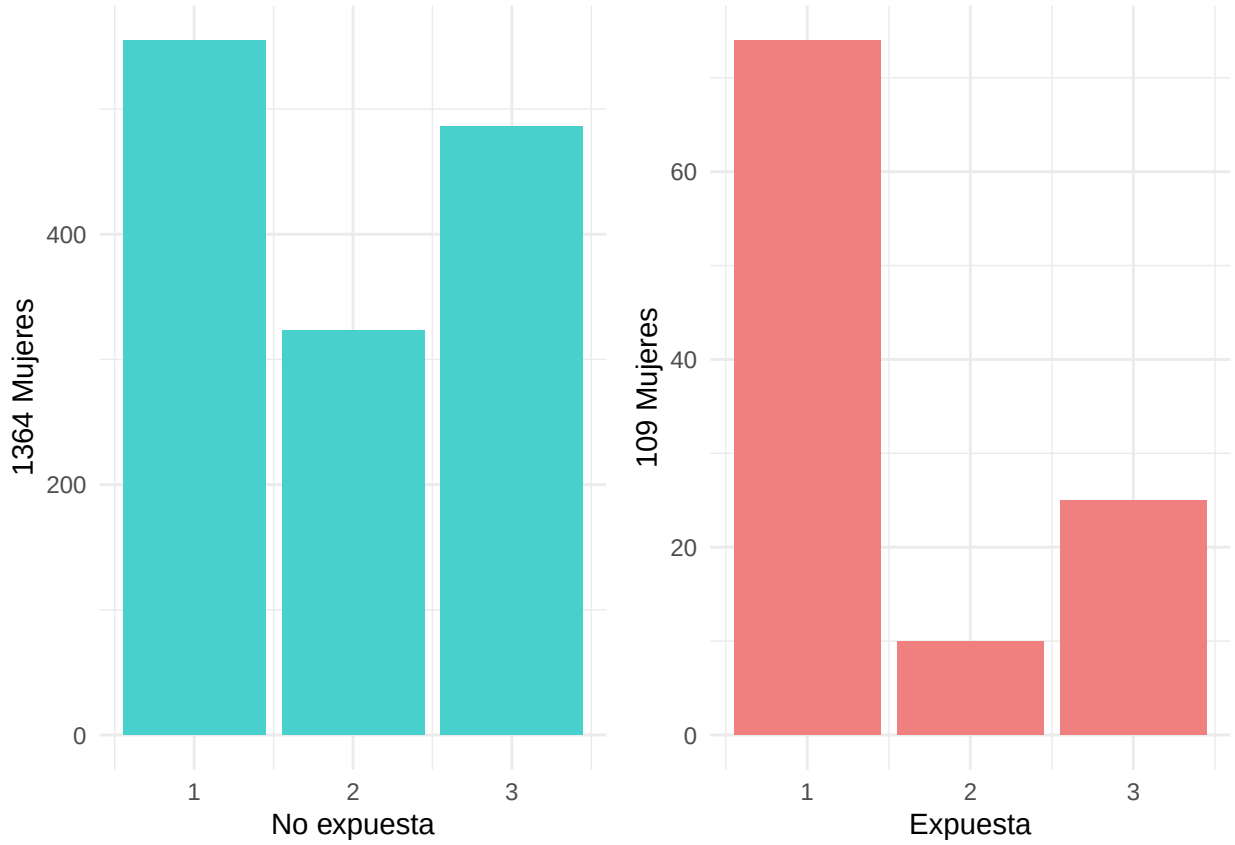
```
  theme_minimal()

# Standard of Living 3
plot3 <- ggplot(groupWifes_3, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="yellow3") +
  labs(x = "Nivel de vida: 3",
       y = paste(nrow(groupWifes_3), "Mujeres"))  +
  theme_minimal()

# Standard of Living 4
plot4 <- ggplot(groupWifes_4, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="green3") +
  labs(x = "Nivel de vida: 4",
       y = paste(nrow(groupWifes_4), "Mujeres"))  +
  theme_minimal()

# Plot grid with four plots
grid.arrange(plot1, plot2, plot3, plot4, ncol=2, nrow=2)
```



Como se puede observar, realmente el nivel de vida afecta a la elección pero en pequeña medida, ya que aunque varían mínimamente las proporciones, no existen cambios considerables.

Por último, para poder terminar de comprender el estado del problema, se ha decidido evaluar si la exposición a los medios afecta en alguna medida a la elección tomada sobre anticonceptivos.

```r
# Group wifes by education
groupWifes_1 <- bd_cmc[which(bd_cmc$MediaExposure == 0),]
groupWifes_2 <- bd_cmc[which(bd_cmc$MediaExposure == 1),]

# Basic bar plot of Contraconceptive Method
# Non-Religious
plot1 <- ggplot(groupWifes_1, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="mediumturquoise") +
  labs(x = "No expuesta",
       y = paste(nrow(groupWifes_1), "Mujeres"))  +
  theme_minimal()

# Religious
plot2 <- ggplot(groupWifes_2, aes(x=ContraconceptiveMethod)) +
  geom_bar( fill="lightcoral") +
  labs(x = "Expuesta",
       y = paste(nrow(groupWifes_2), "Mujeres"))  +
  theme_minimal()

# Plot grid with two plots
grid.arrange(plot1, plot2, ncol=2)
```



Se ve que finalmente es una variable que afecta en menor medida a la elección del anticonceptivo empleado.

Una vez comprendidas las distintas partes del problema, evaluadas las diferentes variables y razonada su importancia, se puede proceder a realizar el problema con el objetivo de conseguir un predictor que nos

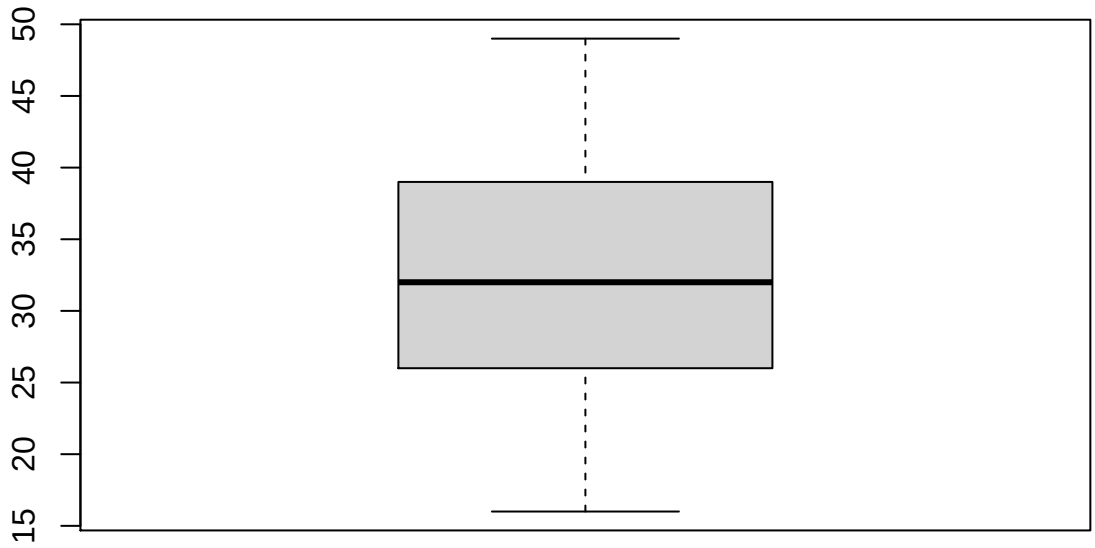
permita clasificar según el método anticonceptivo escogido.

Preprocesamiento de los datos

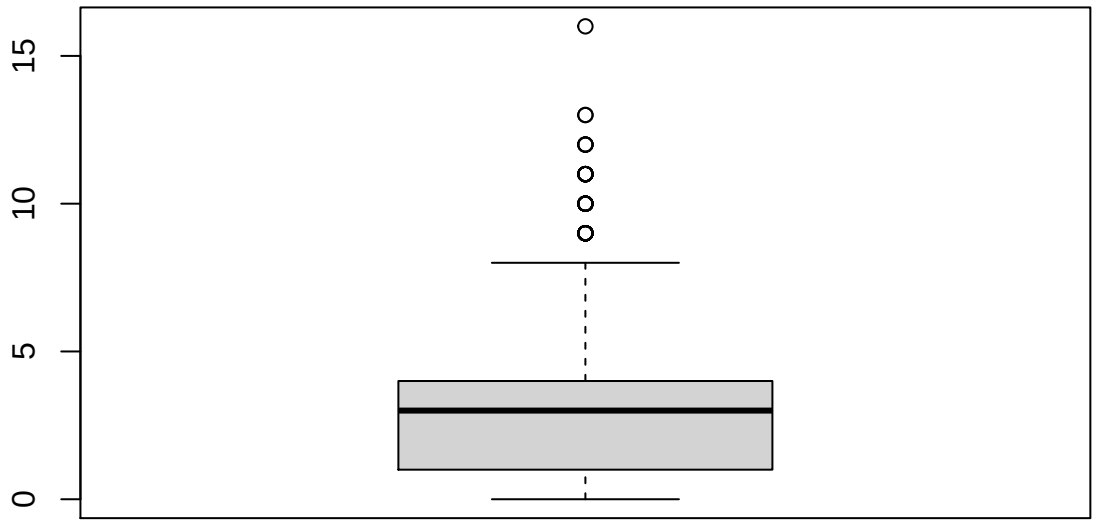
Detección de outliers

A continuación, se procede a evaluar los posibles *outliers* de las variables numéricas, para ello se realizan **boxplots** de las variables numéricas (*WifeAge* y *Children*), y a continuación se evaluan los posibles *outliers* para determinar si son casos especiales o atípicos.

```
# Boxplot WifeAge
boxplot(bd_cmc$WifeAge)
```




```
# Boxplot Children
boxplot(bd_cmc$Children)
```



Como se puede ver, existen posibles *outliers* dentro del número de hijos por familia, por lo que para determinar si se tratan realmente de *outliers* se visualizaran el número máximo, mínimo y cuartiles de dicha característica, y a continuación se mostraran los casos que se determinan como *outliers*.

```
# Print summary of Children
summary(bd_cmc$Children)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   3.261   4.000  16.000
```

```
# Print summary of possible outliers
summary(boxplot.stats(bd_cmc$Children)$out)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00    9.00   10.00   10.33   11.00   16.00
```

Tal y como se puede observar, son datos especiales, en los que se posee un número de hijos elevado, más considerando que la media se encuentra en *3.261* hijos por familia, pero no son *outliers* ni datos erróneos, sino casos especiales. Tras razonar y comprender su funcionalidad, y teniendo en cuenta que posteriormente se realizará una discretización de valores, se ha considerado descartar estos datos, ya que representan familias muy numerosas, y pueden confundirse con familias de menor cantidad de hijos pero a su vez muy numerosas por igual.

Además, estos datos suponen un *3,05%* del conjunto de datos, por lo que realmente no es una gran pérdida de información y sin embargo obtenemos un conjunto de datos más balanceado.

A continuación eliminamos los datos indicados y podemos observar que el conjunto no varía apenas, pero los límites se acotan, por lo que la eliminación de dichos datos no ha supuesto un desbalanceamiento de los datos ni una pérdida de información que afecte en el estudio.

```
# Remove Children outliers
outliers <- boxplot.stats(bd_cmc$Children)$out
for(i in 1:length(outliers))
  bd_cmc <- bd_cmc[!bd_cmc$Children == outliers[i], ]

# Print summary of Children
summary(bd_cmc$Children)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   3.039   4.000   8.000
```

Discretización de variables

Una de las técnicas principales de preprocesamiento de datos consisten en la discretización de valores continuos, o discretización de valores categóricos como agrupación de estas categorías en otras que abarquen más categorías en una misma. Para ello se ha utilizado la función cut, perteneciente al paquete *base* del propio *R*.

Las variables que se consideran para una discretización son las dos variables númericas examinadas anteriormente: *WifeAge* y *Children*. Para ello se realizan las siguientes discretizaciones tras estudiar dichas variables:

- **WifeAge**: Tras analizar la mayoría de edad en países árabes, y los ciclos de vida de una mujer, se han establecido los siguientes rangos.
  - *Menor de edad* (1): 16-20 años.
  - *Joven* (2): 21-30 años.
  - *Adulta* (3): 31-40 años.
  - *Mayor* (4): 41-50 años.
- **Children**: Teniendo en cuenta las características estadísticas de esta variable, se han establecido rangos basados en la media y cuartiles.
  - *Sin hijos* (1): 0 hijos.
  - *Pocos* (2): 1-2 hijos
  - *Media* (3): 3-4 hijos
  - *Numerosa* (4): 5-8 hijos.

```
# Discretize WifeAge by range (Minor, Young, Adult, Old)
bd_cmc[,"WifeAge"] <- cut(bd_cmc[,"WifeAge"],
                          breaks = c(16,21,31,41,50),
                          labels = c(1,2,3,4), right = FALSE)
# Discretize Children by range (No children, Few, Medium, Numerous)
bd_cmc[,"Children"] <- cut(bd_cmc[,"Children"],
                                breaks = c(0,1,3,5,9),
                                labels = c(1,2,3,4), right = FALSE)
```

Por último, para comprobar la integridad de los datos obtenidos, se procede a volver a eliminar los valores perdidos:

```
# Omit NAs
bd_cmc <- na.omit(bd_cmc)
```

Normalización de variables

Una vez discretizadas las variables, se procede a evaluar las variables categóricas, de cara a poder ser procesadas fácilmente por los siguientes modelos. Las variables siguen un orden lógico, por lo que se planteará utilizarlas como variables numéricas, frente al uso de variables *dummy*. No se normalizará la variable *ContraconceptiveMethod*.

Para ello, se normalizan las variables numéricas haciendo uso de la función que definimos para realización normalización min_max.

```
# Definition of min_max normalization function
min_max_norm <- function(x) {
    (x - min(x)) / (max(x) - min(x))
}
```

A continuación se normalizan las variables:

```
# Transform all variables to numeric
for(i in 1:length(bd_cmc)) bd_cmc[,i] <- as.numeric(bd_cmc[,i])

# Normalize variables
for(i in 1:length(bd_cmc[-1])) bd_cmc[,i] <- min_max_norm(bd_cmc[,i])
```

Por último, transformamos las variables binarias en booleanas:

```
# Convert binary variables into logicals
bd_cmc[,"WifeReligion"] <- ifelse(bd_cmc[,"WifeReligion"] == 1, TRUE, FALSE)
bd_cmc[,"WifeWorking"] <- ifelse(bd_cmc[,"WifeWorking"] == 1, TRUE, FALSE)
bd_cmc[,"MediaExposure"] <- ifelse(bd_cmc[,"MediaExposure"] == 1, TRUE, FALSE)
```

Selección de características

Una de las técnicas aprendidas previamente es la de selección de características, por lo que para ello se utilizan diversos métodos con el fin de detectar la importancia de las diferentes variables a tratar:

- Boruta, utilizando para ello una selección de variables significativas, teniendo en cuenta tentativas y confirmar si las variables deben permanecer en el modelo o pueden ser eliminadas, y recibiendo información sobre su importancia.
- Entrenar un modelo lineal y observar que variables son necesarias para construir dicho modelo.

Una vez analizados ambos resultados, se realizará un análisis y se decidirá que variables se han de eliminar.

Para el modelo de **Boruta** se ha utilizado las funciones Boruta, getSelectedAttributes, TentativeRoughFix y attStats del paquete Boruta.

Para ello entrenamos un modelo con **Boruta**, obteniendo los atributos seleccionados y realizando un arreglo para obtener las tentativas (si las hay). A continuación obtendremos de nuevos los atributos y mostramos finalmente la información obtenida.

```r
# Perform Boruta search
boruta_output <- Boruta(ContraconceptiveMethod~., data = bd_cmc, doTrace=0)

# Get significant variables including tentatives
boruta_signif <- getSelectedAttributes(boruta_output, withTentative = TRUE)

# Do a tentative rough fix
roughFixMod <- TentativeRoughFix(boruta_output)
```

```
## Warning in TentativeRoughFix(boruta_output): There are no Tentative attributes!
## Returning original object.
```

```r
boruta_signif <- getSelectedAttributes(roughFixMod)

# Variable Importance Scores
imps <- attStats(roughFixMod)
imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]

# Print variable importance
print(imps2[order(-imps2$meanImp), ])
```

```
##                   meanImp  decision
## Children         53.056733 Confirmed
## WifeAge          36.343449 Confirmed
## WifeEducation    16.105547 Confirmed
## HusbandEducation  7.776937 Confirmed
## MediaExposure     6.726552 Confirmed
## StandardOfLiving  5.409528 Confirmed
## HusbandOcupation  4.991830 Confirmed
```

Tras observar con *Boruta* las variables con mayor importancia, se procede a realizar el **Modelo lineal**. Para ello se ha utilizado la función stepAIC, la cual nos permite hacer distintos tipos de regresiones de características, pero en nuestro caso indicamos que realice de tipo *both* (*backward* y *forward*). Para ello primero entrenamos el modelo con un modelo lineal utilizando la función lm.

```r
# Train the linear model
model <- lm(ContraconceptiveMethod~., data = bd_cmc)
# Get the stepwise regression model
step.model <- stepAIC(model, direction = "both", trace = FALSE)
# Get the model
anova <- step.model$anova
anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## ContraconceptiveMethod ~ WifeAge + WifeEducation + HusbandEducation +
##     Children + WifeReligion + WifeWorking + HusbandOcupation +
##     StandardOfLiving + MediaExposure
##
## Final Model:
```

```
## ContraconceptiveMethod ~ WifeAge + WifeEducation + Children +
##     HusbandOcupation + StandardOfLiving + MediaExposure
##
##
##                  Step Df  Deviance Resid. Df Resid. Dev       AIC
## 1                                       1418   940.5184 -576.3312
## 2 - HusbandEducation  1 0.2446601      1419   940.7631 -577.9598
## 3      - WifeWorking  1 0.7265631      1420   941.4897 -578.8574
## 4     - WifeReligion  1 0.9361858      1421   942.4258 -579.4381
```

A continuación construimos un modelo de regresión para establecer una variables predictoras añadiendo y elimininando predictores basándonos en el valor p, para ello utilizamos la función ols_step_both_p del paquete olsrr:

```
# Regression model
ols_step_both_p(model)
```

```
##
##                                 Stepwise Selection Summary
## ----------------------------------------------------------------------------------------------------
##                             Added/                   Adj.
## Step        Variable        Removed     R-Square    R-Square      C(p)         AIC         RMSE
## ----------------------------------------------------------------------------------------------------
##    1        Children        addition       0.032       0.031    174.7690    3633.4981    0.8623
##    2        WifeAge         addition       0.094       0.093     73.0650    3539.7303    0.8342
##    3     WifeEducation      addition       0.126       0.124     22.5620    3490.6810    0.8197
##    4    StandardOfLiving    addition       0.132       0.130     15.0180    3483.2015    0.8173
##    5    HusbandOcupation    addition       0.136       0.133      9.7680    3477.9590    0.8155
##    6     MediaExposure      addition       0.139       0.136      6.8760    3475.0503    0.8144
## ----------------------------------------------------------------------------------------------------
```

Una vez construido el modelo, se examinan las variables que se han utilizado para construir el modelo y cuales se pueden descartar para analizarlas junto al modelo de **Boruta**:

Variable	Estado
Children	Aceptada
WifeAge	Aceptada
WifeEducation	Aceptada
MediaExposure	Aceptada
StandardOfLiving	Aceptada
HusbandOcupation	Aceptada
HusbandEducation	Aceptada
WifeWorking	Rechazada
WifeReligion	Rechazada
ContraconceptiveMethod	**Clasificatoria**

Viendo los resultados obtenidos, y tal y como hemos visto en el análisis inicial de datos, la información que se descarta es relevante para el problema, pero no permite obtenir conclusiones adicionales sobre el mismo, por lo que se procede a eliminarlas del problema.

```
bd_cmc[,"WifeWorking"] <- NULL
bd_cmc[,"WifeReligion"] <- NULL
```

Creación de grupos de entrenamiento y test

Para comprobar que el modelo de árbol de decisión se realiza una validación cruzada, por lo que se necesitan dos conjuntos: *train* y *test*. El conjunto de entrenamiento comprenderá el *80%* de los datos tomados para entrenar el árbol de decisión que clasifique el problema, y el conjunto de testeo se empleará para validad el modelo de ajuste obtenido.

Realmente lo ideal sería una validación cruzada del tipo K-fold, pero necesitaríamos obtener de entrada un conjunto de datos de test y otro de entrenamiento, por lo que realizarlos nosotros no quitaría la aleatoriedad de los mismos.

```
# Function definition
create_train_test <- function(data, size = 0.8, train = TRUE) {
  # Get number of rows
  n_row = nrow(data)
  # Get percentage of rows
  total_row = size * n_row
  # Get training sample indexes
  train_sample <- 1: total_row
  if (train == TRUE) {
    # Train dataset
    return (data[train_sample, ])
  } else {
    # Test dataset
    return (data[-train_sample, ])
  }
}
```

Modelos de clasificación

Para la realización de la práctica se ha decidido tomar diferentes modelos de clasificación y ver como estos son capaces de adaptarse a las condiciones del problema. Los modelos escogidos son:

- Árboles de decisión, con la función rpart.
- Random Forest, con la función randomForest.
- Red Neuronal Artificial, con nnet.
- Naive Bayes, con naiveBayes.
- K-Nearest Neighbour, con knn.

Tras observar los resultados obtenidos, se analizaran los resultados obtenidos, se obtendrán una serie de conclusiones sobre los modelos, y se obtendrá conocimiento suficiente para poder analizar modelos de predicción de métodos anticonceptivos usados.

Todos estos modelos a su vez serán comprobados mediante matrices de confusión con la función confusionMatrix de la librería caret y visualizandolas gráficamente.

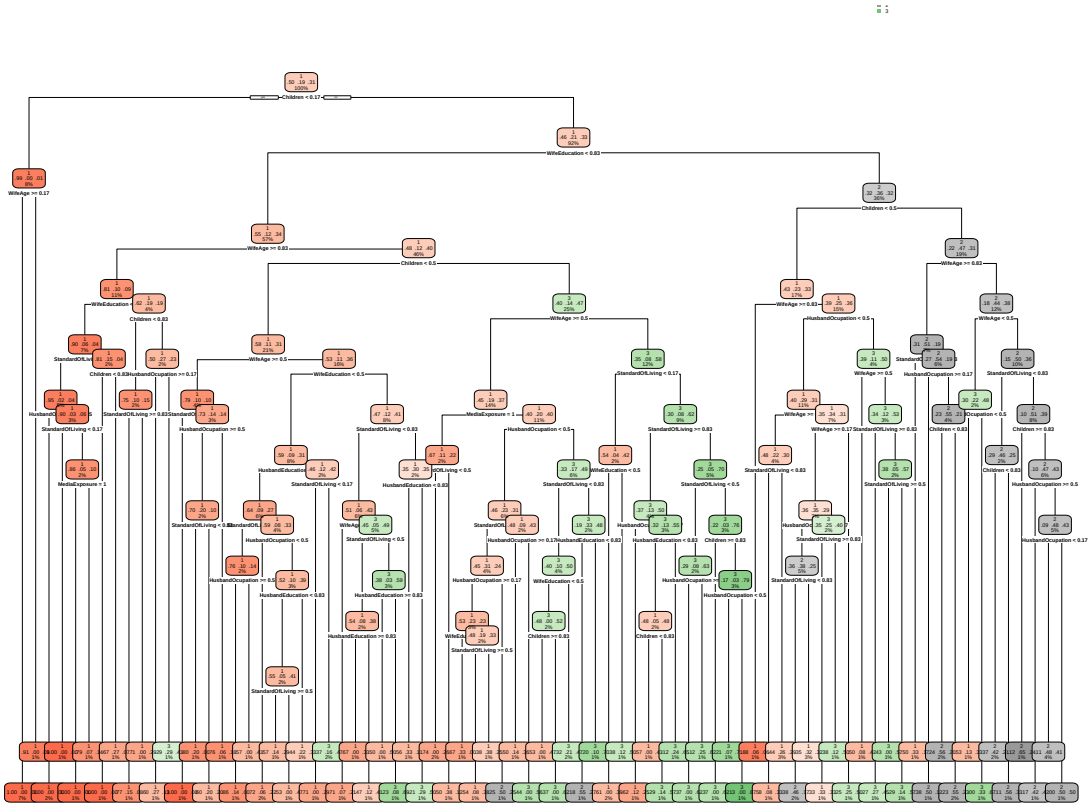
Árboles de decisión

El primer modelo de clasificación escogido es el árbol de decisión binario, el cual dado un conjunto de datos, divide este por características de forma que se tome una decisión o su opuesta, para clasificar. Se ha definido una función que solicita un conjunto de entrenamiento y realiza la clasificación en un árbol de decisión binario. Para ello se ha utilizado la librería rpart, concretamente haciendo uso tanto de la función rpart como de su representación mediante rpart.plot, perteneciente a la librería rpart.plot.

Además, se indica el parámetro `cp=-1` para que se explore el árbol entero, que aunque no se represente entero en el árbol dibujado, el modelo tendrá un mayor ajuste.

```
# Set train and test data
data_train <- create_train_test(bd_cmc, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc, 0.8, train = FALSE)

# Adjust the model
fit <- rpart(ContraconceptiveMethod~., data = data_train, method = 'class', cp=-1)

# Plot the model
rpart.plot(fit)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



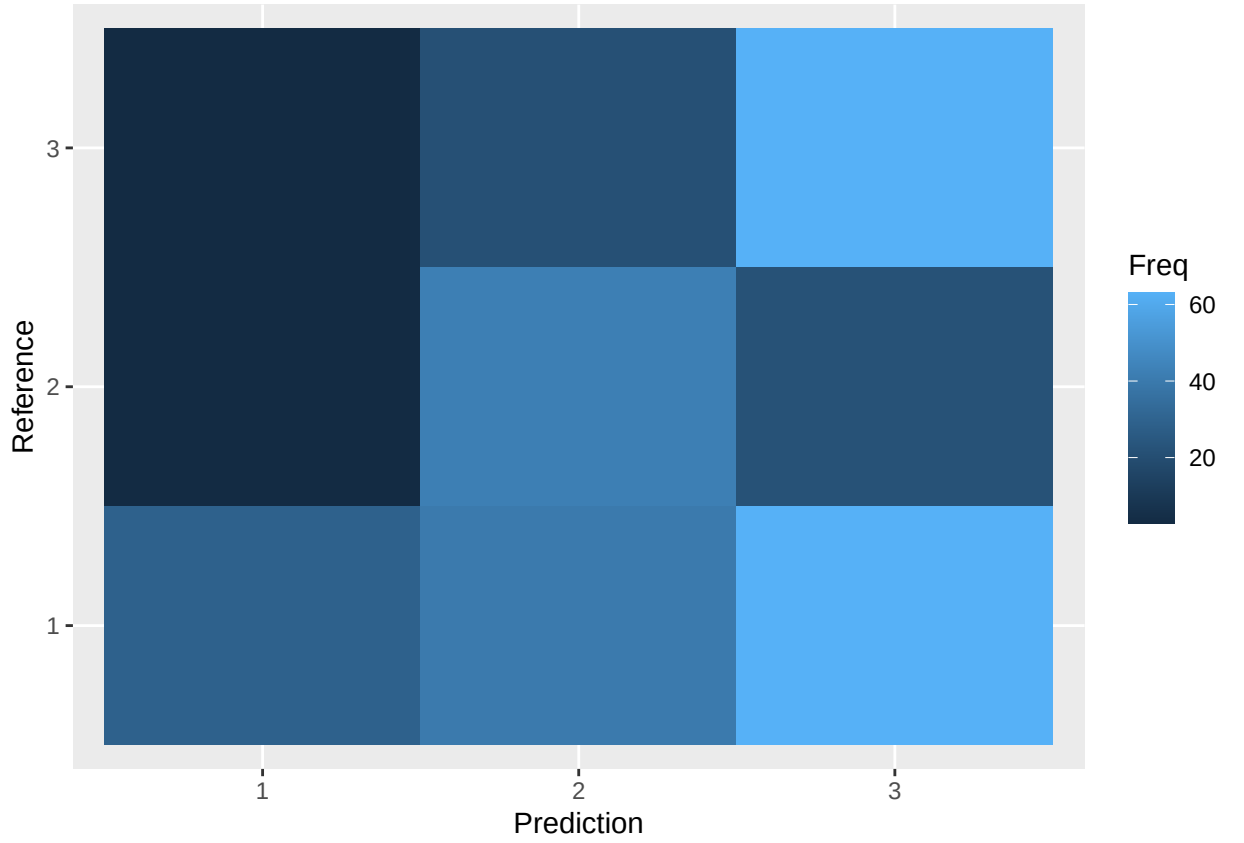
Una vez realizado el modelo de árbol de decisión, se procede a estimar el grupo de test, crear la matrix de confusión y ver el *accuracy* obtenido:

```r
# Predict the test data class
predicted_class <-predict(fit, data_test, type='class')

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
dt_accuracy <- matrix$overall["Accuracy"]

# Print confusion matrix
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 29  3  3
##          2 40 42 21
##          3 63 22 63
##
## Overall Statistics
##
##                Accuracy : 0.4685
##                  95% CI : (0.4095, 0.5282)
##     No Information Rate : 0.4615
##     P-Value [Acc > NIR] : 0.4288
##
##                   Kappa : 0.2426
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.2197   0.6269   0.7241
## Specificity            0.9610   0.7215   0.5729
## Pos Pred Value         0.8286   0.4078   0.4257
## Neg Pred Value         0.5896   0.8634   0.8261
## Prevalence             0.4615   0.2343   0.3042
## Detection Rate         0.1014   0.1469   0.2203
## Detection Prevalence   0.1224   0.3601   0.5175
## Balanced Accuracy      0.5904   0.6742   0.6485
```

A continuación se visualiza la matriz de confusión asociada al modelo de árbol de decisión:

```r
# Plot confusion matrix
matrix_table <- data.frame(matrix$table)
ggplot(matrix_table, aes(x=Prediction, y=Reference, fill=Freq)) + geom_tile()
```



Random Forest

El siguiente modelo a observar se trata de Random Forest, el cual dado un conjunto de datos, divide este por características de forma que se tome una decisión o su opuesta, para clasificar, formando varios árboles de decisión con decisiones diferentes. Tras crear una seria de árboles de decisión, el algoritmo se encarga de realizar una votación entre los distintos árboles para obtener un modelo final. Para ello se ha utilizado la librería randomForest, utilizando para ello la función randomForest.

Para la realización del modelo se escoge un total de 500 árboles de decisión (*ntree*), y se escoge 3 como número de variables seleccionadas aleatoriamente para la división (*mtry*), tras estudiar y probar diferentes configuraciones del modelo.

Una vez realizado el modelo de Random Forest, se procede a estimar el grupo de test, crear la matrix de confusión y ver el *accuracy* obtenido:

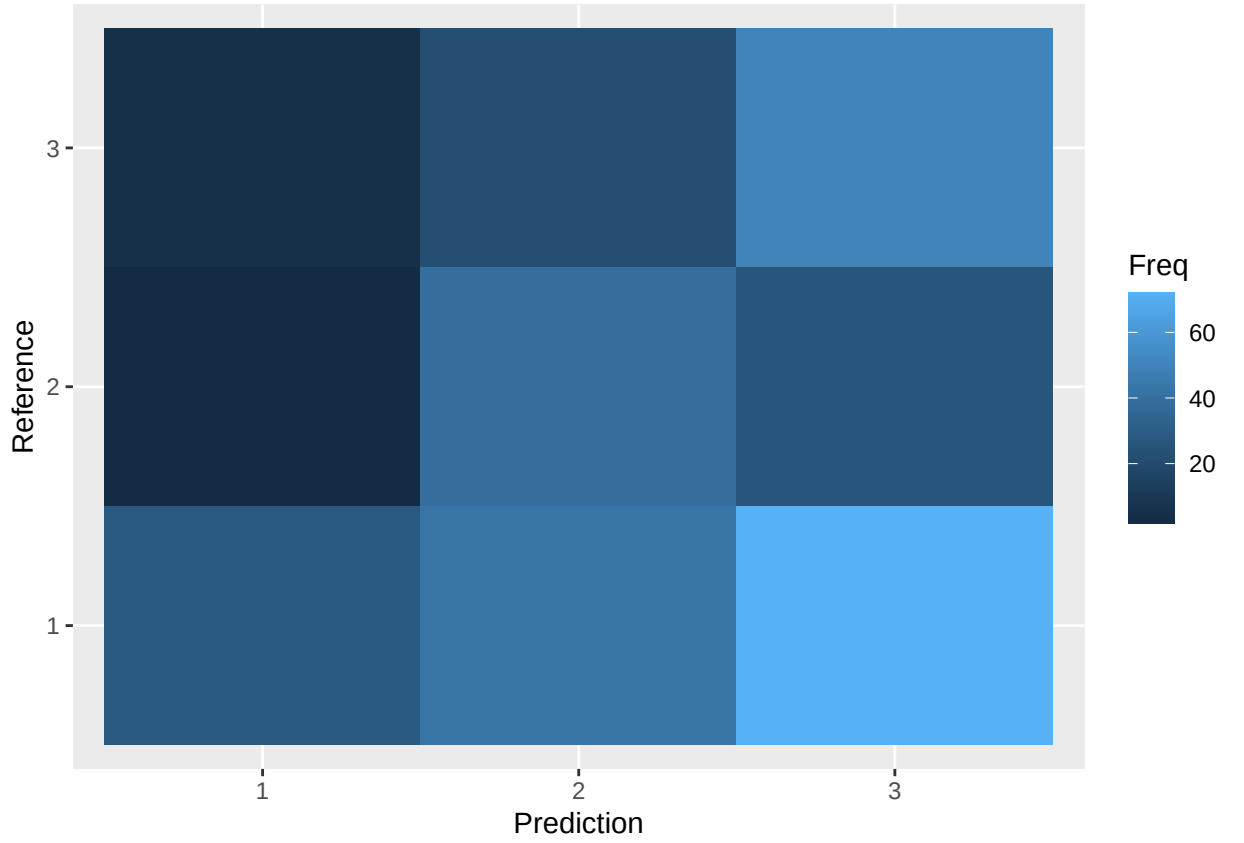
```
# Set train and test data
data_train <- create_train_test(bd_cmc, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc, 0.8, train = FALSE)


# Adjust the model
fit <- randomForest(as.factor(ContraconceptiveMethod)~., data = data_train,
                    ntree=500, mtry=3)

# Predict the test data class
predicted_class <-predict(fit, data_test)
```

```r
# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
rf_accuracy <- matrix$overall["Accuracy"]

# Print confusion matrix
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 28  2  5
##          2 42 39 22
##          3 72 26 50
##
## Overall Statistics
##
##                Accuracy : 0.4091
##                  95% CI : (0.3516, 0.4685)
##     No Information Rate : 0.4965
##     P-Value [Acc > NIR] : 0.9988
##
##                   Kappa : 0.1742
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.1972   0.5821   0.6494
## Specificity            0.9514   0.7078   0.5311
## Pos Pred Value         0.8000   0.3786   0.3378
## Neg Pred Value         0.5458   0.8470   0.8043
## Prevalence             0.4965   0.2343   0.2692
## Detection Rate         0.0979   0.1364   0.1748
## Detection Prevalence   0.1224   0.3601   0.5175
## Balanced Accuracy      0.5743   0.6449   0.5902
```

A continuación se visualiza la matriz de confusión asociada al modelo de Random Forest:

```r
# Plot confusion matrix
matrix_table <- data.frame(matrix$table)
ggplot(matrix_table, aes(x=Prediction, y=Reference, fill=Freq)) + geom_tile()
```



Redes Neuronales

El tercer modelo seleccionado es el de Red Neuronal Articifial, el cual recibe en las neuronas de entrada un conjunto de datos, que son procesados por las diferentes capas de neuronas ocultas (previamente configuradas) de la red, para obtener en la salida un modelo que a través del aprendizaje puede ajustarse mediante pesos al conjunto de datos.

Para ello se ha utilizado la librería nnet, empleando la función nnet.

Para la realización del modelo se han utilizado una serie de configuraciones distintas, y tras estudiar y analizar los resultados, y analizar el conjunto de datos, se han establecido los siguientes parámetros:

- Número de capas ocultas: 3 (*size*).
- Número máximo de iteraciones: 300 (*maxit*).
- Permitir conexiones entre la entrada y la salida mediante el parámetro *skip*.

Se ha comprobado que el modelo se adapta a un sistema de 2 capas de neuronas ocultas, salvo que se añada un número demasiado elevado de capas, por lo que 2 es una buena estimación.

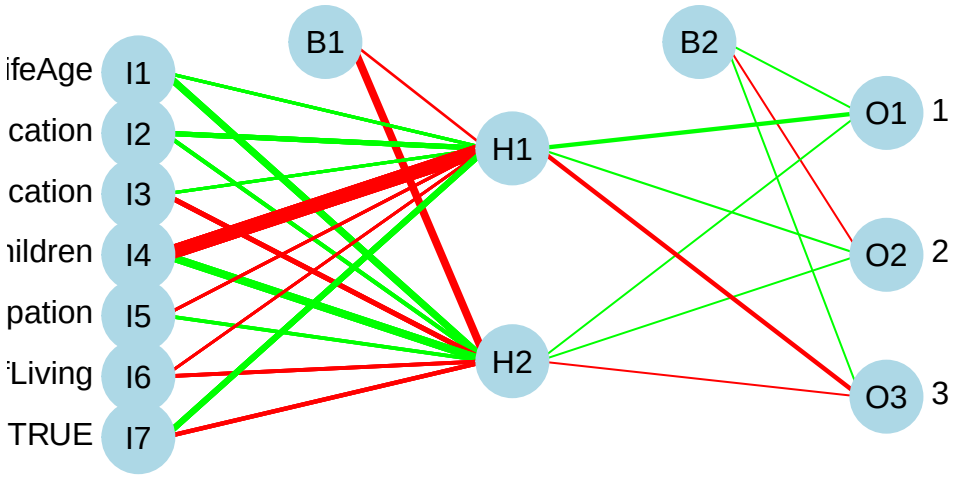
Se han utilizado estos parámetros ya que un ajuste excesivo del modelo es incapaz de predecir correctamente el conjunto de datos y añade una complejidad excesiva, por lo que con una configuración de 2 capas de neuronas ocultas se pueden obtener unos buenos resultados, además de que el modelo llega a converger antes de las 300 iteraciones.

Para visualizar la red neuronal se utiliza la función plotnet del paquete NeuralNetTools.

```r
# Set train and test data
data_train <- create_train_test(bd_cmc, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc, 0.8, train = FALSE)

# Adjust the model
nn=nnet(as.factor(ContraconceptiveMethod)~ ., data=data_train, size=2,
        maxit=300, skip=TRUE)
```

```
## # weights:  46
## initial  value 1536.045140
## iter  10 value 1027.723980
## iter  20 value 1012.059110
## iter  30 value 986.825237
## iter  40 value 968.068437
## iter  50 value 962.964165
## iter  60 value 959.145874
## iter  70 value 958.528160
## iter  80 value 957.953244
## iter  90 value 956.550154
## iter 100 value 956.206442
## iter 110 value 956.186624
## iter 120 value 956.123694
## iter 130 value 956.017582
## iter 140 value 955.944944
## iter 150 value 955.919073
## iter 160 value 955.902422
## iter 170 value 955.883047
## iter 180 value 955.605134
## final  value 955.293317
## converged
```

```r
# Plot the model
plotnet(nn, pos_col="green", neg_col="red", max_sp=TRUE)
```



Una vez realizado el modelo de Red Neuronal Artificial, se procede a estimar el grupo de test, crear la matrix de confusión y ver el *accuracy* obtenido:

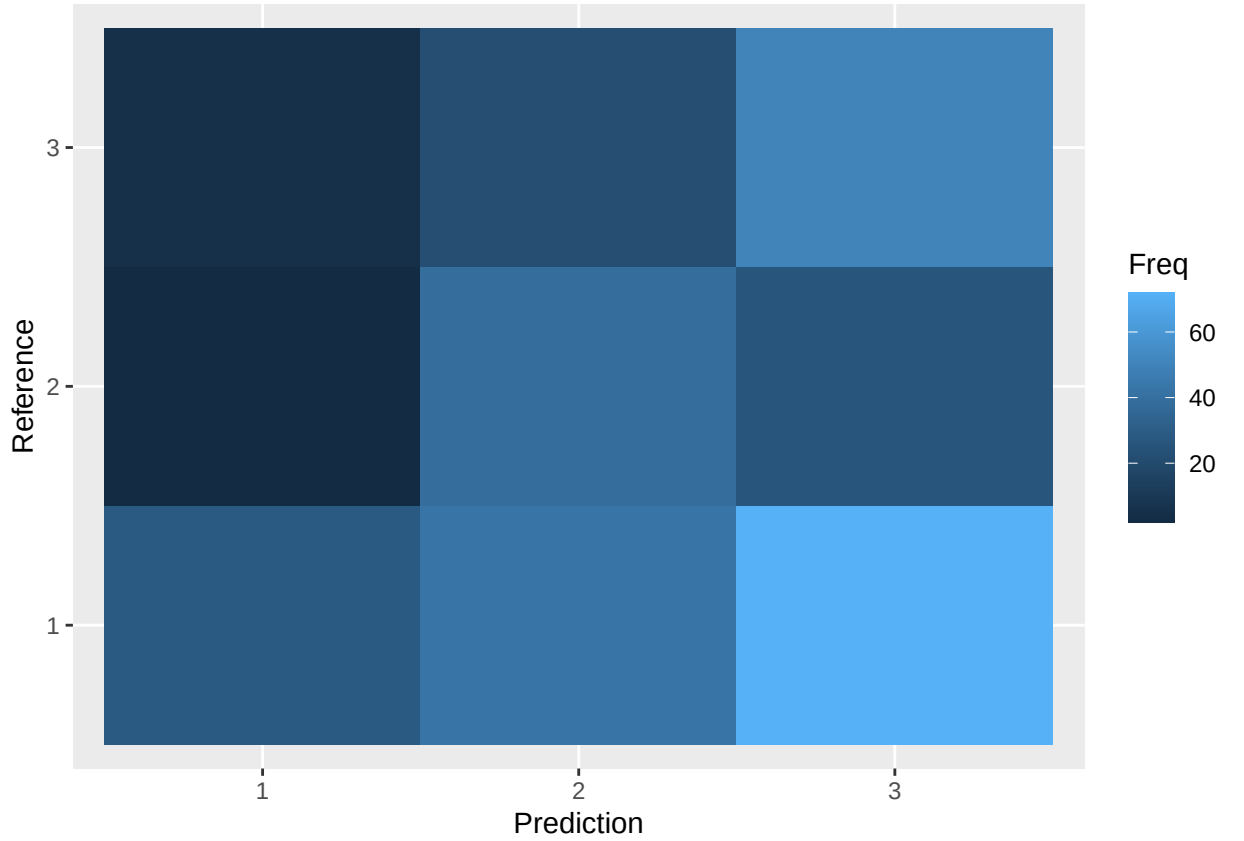
```
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nn_accuracy <- matrix$overall["Accuracy"]

# Print confusion matrix
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 28  2  5
##          2 42 39 22
##          3 72 26 50
##
## Overall Statistics
##
##                Accuracy : 0.4091
##                  95% CI : (0.3516, 0.4685)
##     No Information Rate : 0.4965
```

```
##     P-Value [Acc > NIR] : 0.9988
##
##                   Kappa : 0.1742
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.1972   0.5821   0.6494
## Specificity            0.9514   0.7078   0.5311
## Pos Pred Value         0.8000   0.3786   0.3378
## Neg Pred Value         0.5458   0.8470   0.8043
## Prevalence             0.4965   0.2343   0.2692
## Detection Rate         0.0979   0.1364   0.1748
## Detection Prevalence   0.1224   0.3601   0.5175
## Balanced Accuracy      0.5743   0.6449   0.5902
```

A continuación se visualiza la matriz de confusión asociada al modelo de Red Neuronal Artificial:

```
# Plot confusion matrix
matrix_table <- data.frame(matrix$table)
ggplot(matrix_table, aes(x=Prediction, y=Reference, fill=Freq)) + geom_tile()
```



Naive Bayes

El cuarto modelo seleccionado es el de Naive Bayes, el cual asume que la presencia de una característica no es dependiente de la existencia de otra, permitiendo un entrenamiento del modelo mediante las frecuencias relativas de las características del conjunto de entrenamiento. Para ello crea un modelo que mezcla probabilidad y frecuencia, definiendo así las probabilidades de que se de una determinada clase a partir de sus características de forma independiente.

Para ello se ha utilizado la librería e1071, utilizando la función naiveBayes.

A continuación se pueden observar las diferentes probabilidades individuales de cada característica:

```
# Set train and test data
data_train <- create_train_test(bd_cmc, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc, 0.8, train = FALSE)

# Adjust the model
fit=naiveBayes(as.factor(ContraconceptiveMethod)~ ., data=data_train)

# Print the model
fit
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         1         2         3
## 0.4991243 0.1943958 0.3064799
##
## Conditional probabilities:
##    WifeAge
## Y        [,1]      [,2]
##   1 0.5795322 0.3036107
##   2 0.6351351 0.2586012
##   3 0.4819048 0.2356810
##
##    WifeEducation
## Y        [,1]      [,2]
##   1 0.5643275 0.3483080
##   2 0.8363363 0.2531670
##   3 0.6628571 0.3191611
##
##    HusbandEducation
## Y        [,1]      [,2]
##   1 0.7695906 0.2950004
##   2 0.8918919 0.2271743
##   3 0.8266667 0.2389620
##
##    Children
## Y        [,1]      [,2]
##   1 0.4918129 0.3283625
```

```
##   2 0.6666667 0.2653057
##   3 0.5971429 0.2483786
##
##    HusbandOcupation
## Y        [,1]      [,2]
##   1 0.4011696 0.2810117
##   2 0.2747748 0.2975927
##   3 0.4142857 0.2867981
##
##    StandardOfLiving
## Y        [,1]      [,2]
##   1 0.6502924 0.3486870
##   2 0.8378378 0.2511507
##   3 0.7019048 0.3246670
##
##    MediaExposure
## Y       FALSE       TRUE
##   1 0.88596491 0.11403509
##   2 0.96396396 0.03603604
##   3 0.95714286 0.04285714
```

Una vez realizado el modelo de Naive Bayes, se procede a estimar el grupo de test, crear la matrix de confusión y ver el *accuracy* obtenido:

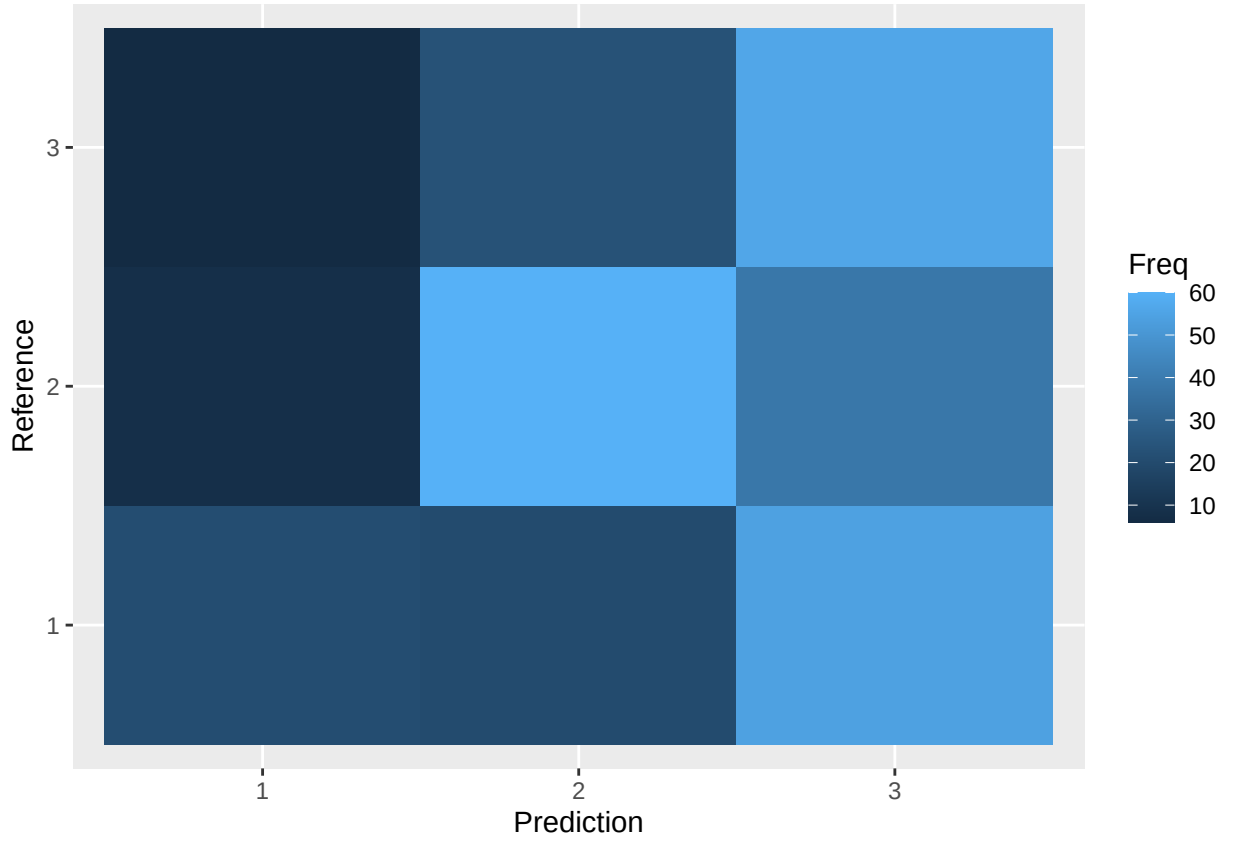
```
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nb_accuracy <- matrix$overall["Accuracy"]

# Print confusion matrix
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 21  8  6
##          2 20 60 23
##          3 54 38 56
##
## Overall Statistics
##
##                Accuracy : 0.479
##                  95% CI : (0.4199, 0.5386)
##     No Information Rate : 0.3706
##     P-Value [Acc > NIR] : 0.0001158
##
##                   Kappa : 0.2248
##
##  Mcnemar's Test P-Value : 3.103e-10
##
## Statistics by Class:
```

```
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity           0.22105   0.5660   0.6588
## Specificity           0.92670   0.7611   0.5423
## Pos Pred Value        0.60000   0.5825   0.3784
## Neg Pred Value        0.70518   0.7486   0.7899
## Prevalence            0.33217   0.3706   0.2972
## Detection Rate        0.07343   0.2098   0.1958
## Detection Prevalence  0.12238   0.3601   0.5175
## Balanced Accuracy     0.57388   0.6636   0.6006
```

A continuación se visualiza la matriz de confusión asociada al modelo de Naive Bayes:

```
# Plot confusion matrix
matrix_table <- data.frame(matrix$table)
ggplot(matrix_table, aes(x=Prediction, y=Reference, fill=Freq)) + geom_tile()
```



KNN

El último modelo seleccionado es el de K-Nearest Neighbors, el cual dado un conjunto de entrada con diferentes clases, analiza si un elemento posee dentro de su rango más vecinos de una clase u otra, y clasifica en aquella clase en la que posea más vecinos dentro de un rango con k vecinos. Este modelo sigue una distribución espacial de los elementos.

Para la realización de este modelo se ha hecho uso de la librería class, concretamente de la función knn.

Tras probar diferentes parámetros, y observar el tamaño del conjunto de datos, se ha tomado como valor k=10, es decir, los 10 vecinos más cercanos.

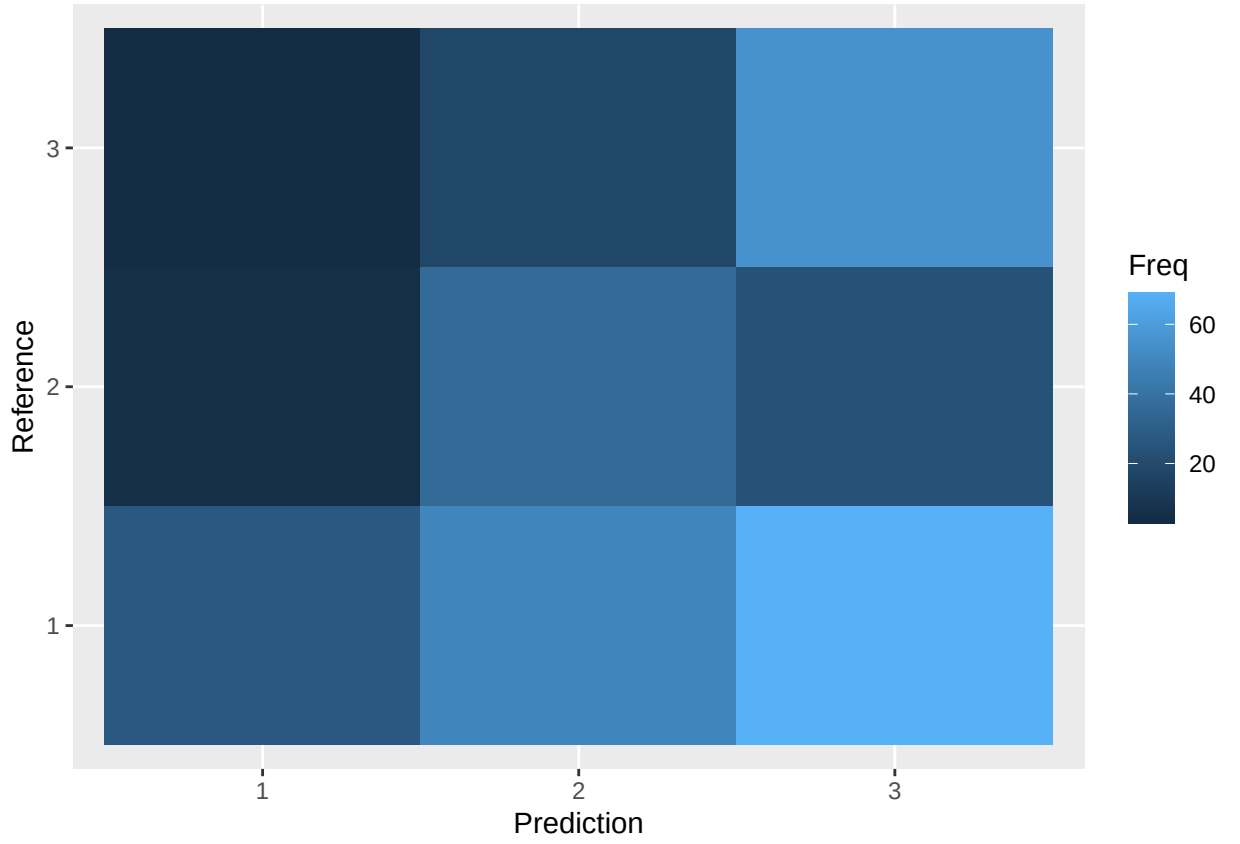
Una vez realizado el modelo de KNN, se procede a estimar el grupo de test, crear la matrix de confusión y ver el *accuracy* obtenido:

```
# Extract train and test labels
train.label <- data_train[, "ContraconceptiveMethod"]
test.label <- data_test[, "ContraconceptiveMethod"]
data_train[,"ContraconceptiveMethod"] <- NULL
data_test[,"ContraconceptiveMethod"] <- NULL

# Adjust the model
predicted_class <- knn(data_train,data_test,cl=train.label,k=10)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(test.label), predicted_class)
knn_accuracy <- matrix$overall["Accuracy"]

# Print confusion matrix
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3
##          1 27  5  3
##          2 49 36 18
##          3 69 24 55
##
## Overall Statistics
##
##                Accuracy : 0.4126
##                  95% CI : (0.355, 0.4721)
##     No Information Rate : 0.507
##     P-Value [Acc > NIR] : 0.9994
##
##                   Kappa : 0.1826
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity           0.18621   0.5538   0.7237
## Specificity           0.94326   0.6968   0.5571
## Pos Pred Value        0.77143   0.3495   0.3716
## Neg Pred Value        0.52988   0.8415   0.8478
## Prevalence            0.50699   0.2273   0.2657
## Detection Rate        0.09441   0.1259   0.1923
## Detection Prevalence  0.12238   0.3601   0.5175
## Balanced Accuracy     0.56473   0.6253   0.6404
```

A continuación se visualiza la matriz de confusión asociada al modelo de KNN:

```r
# Plot confusion matrix
matrix_table <- data.frame(matrix$table)
ggplot(matrix_table, aes(x=Prediction, y=Reference, fill=Freq)) + geom_tile()
```



Comparativa Accuracy

Tras realizar los diferentes modelos, se procede a visualizar una tabla comparativa del *accuracy* obtenido por los diferentes modelos:

```r
models_name <- c("Árbol de decisión", "Random Forest",
                 "Red Neuronal Artificial", "Naive Bayes", "KNN")
models_acc <- c(dt_accuracy, rf_accuracy, nn_accuracy,
                nb_accuracy, knn_accuracy)
acc_table <- data.frame(models_name, models_acc)
names(acc_table) <- c("Modelos", "Accuracy")

kable(acc_table)
```

Modelos	Accuracy
Árbol de decisión	0.4685315
Random Forest	0.4090909
Red Neuronal Artificial	0.4090909
Naive Bayes	0.4790210

Modelos	Accuracy
KNN	0.4125874

Análisis Resultados

Tras realizar el preprocesamiento, análisis del conjunto de datos, clasificación en distintos modelos, comparativa de modelos y análisis de resultados, se han llegado a las siguientes conclusiones:

- Se trata de un conjunto de datos desbalanceado, por lo que el más mínimo error en un procedimiento se traduce o en resultados erróneos o en resultados pobres, por lo que cada paso dado debe ser planteado, analizado y justificado.
- Uno de los puntos complicados que se trató en la realización de la práctica fue el de selección de características, ya que al tratarse de variables categóricas, muchos modelos no podían realizarse o perdían bastante efectividad debido a estas variables, aunque se planteó utilizar una conversión en **Dummy variables**, tras realizar estudio sobre diferentes opciones se decidió utilizar una categorización ordinal, para que no perdieran el orden lógico las variables y no añadir información extra y que se traduzca en un nuevo problema computacional.
- Tras analizar los distintos modelos, se puede ver que realmente es un problema complejo de decidir realmente que características son las más importantes, aunque se puede observar que en los diferentes modelos se hace notable que las variables **WifeEducation**, **Children** y **WifeAge** son las más representativas del conjunto, y realizando un análisis del conjunto de datos, es un planteamiento coherente.
- No es de extrañar obtener en un conjunto tan desbalanceado unos resultados aparentemente no muy buenos, ya que las variables no siguen entre sí todas una relación lógica y existe ruido en el mismo, por lo que los resultados no serán muy elevados, sin embargo, algo positivo es haber obtenido unos resultados tan similares con diferentes modelos, en los que se aproximan a las mismas conclusiones.

Alternativas

Selección aleatoria de instancias

La primera alternativa propuesta consiste en realizar el problema sobre un conjunto de datos aleatoriamente escogido, y considerablemente inferior al conjunto inicial, con el objetivo de evitar que la falta de balance afecte demasiado en el problema. Para realizar la selección de subconjuntos de datos se va a emplear la función sample_n del paquete tidyverse.

Realizamos el modelo de árboles de decisión:

```
bd_cmc_red <- sample_n(bd_cmc, size=500)

# Set train and test data
data_train <- create_train_test(bd_cmc_red, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc_red, 0.8, train = FALSE)

# Adjust the model
fit <- rpart(ContraconceptiveMethod~., data = data_train, method = 'class', cp=-1)

# Predict the test data class
predicted_class <-predict(fit, data_test, type='class')
```

```r
# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
dt_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Random Forest:

```r
# Adjust the model
fit <- randomForest(as.factor(ContraconceptiveMethod)~., data = data_train,
                    ntree=500, mtry=3)

# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
rf_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Red Neuronal Artificial:

```r
# Adjust the model
nn=nnet(as.factor(ContraconceptiveMethod)~ ., data=data_train, size=2,
        maxit=300, skip=TRUE)
```

```
## # weights:  46
## initial  value 627.254711
## iter  10 value 367.215335
## iter  20 value 348.500052
## iter  30 value 338.799420
## iter  40 value 336.202654
## iter  50 value 335.355940
## iter  60 value 333.937616
## iter  70 value 333.135817
## iter  80 value 332.716269
## iter  90 value 332.373971
## iter 100 value 332.315734
## iter 110 value 332.230892
## iter 120 value 331.944271
## iter 130 value 331.831048
## iter 140 value 331.781384
## iter 150 value 331.577415
## iter 160 value 331.365801
## iter 170 value 331.257843
## iter 180 value 331.184750
## iter 190 value 331.147579
## iter 200 value 331.142924
## iter 210 value 331.124311
## iter 220 value 331.063806
## iter 230 value 330.884637
## iter 240 value 330.844291
## iter 250 value 330.837625
## iter 260 value 330.834659
## iter 270 value 330.824637
```

```
## iter 280 value 330.792002
## iter 290 value 330.789645
## iter 300 value 330.784978
## final  value 330.784978
## stopped after 300 iterations
```

```
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nn_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Naive Bayes:

```
# Adjust the model
fit=naiveBayes(as.factor(ContraconceptiveMethod)~ ., data=data_train)
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nb_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de KNN:

```
# Extract train and test labels
train.label <- data_train[, "ContraconceptiveMethod"]
test.label <- data_test[, "ContraconceptiveMethod"]
data_train[,"ContraconceptiveMethod"] <- NULL
data_test[,"ContraconceptiveMethod"] <- NULL

# Adjust the model
predicted_class <- knn(data_train,data_test,cl=train.label,k=10)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(test.label), predicted_class)
knn_accuracy <- matrix$overall["Accuracy"]
```

Creamos la tabla de comparativas:

```
models_name <- c("Árbol de decisión", "Random Forest",
                 "Red Neuronal Artificial", "Naive Bayes", "KNN")
models_acc <- c(dt_accuracy, rf_accuracy, nn_accuracy,
                nb_accuracy, knn_accuracy)
acc_table <- data.frame(models_name, models_acc)
names(acc_table) <- c("Modelos", "Accuracy")

kable(acc_table)
```

Modelos	Accuracy
Árbol de decisión	0.51
Random Forest	0.49
Red Neuronal Artificial	0.49
Naive Bayes	0.48
KNN	0.51

Como es de esperar en un modelo tan desbalanceado y con ruido, al reducir el número de instancias, los modelos serán capaces de adaptarse con un mejor resultado a los distintos subconjuntos, por lo que los resultados obtenidos serán ligeramente mejores, aunque en este caso se reduzca a la mitad el número de instancias, se puede observar que tampoco existe una gran diferencia.

Transformación del problema a un problema de clasificación más sencillo

Otra de las alternativas es trasladar el problema a un problema de clasificación en el que se evaluan dos posibilidades: Que la mujer utilice métodos anticonceptivos o no. Posteriormente se realizaría un problema de clasificación entre los diferentes métodos anticonceptivos.

Para probar esta teoría, se ha realizado un ejemplo de como se realizaría la primera parte del problema, es decir, un problema de clasificación que determine cuando se usa o no un método anticonceptivo. Para ello lo primero es reunir ambos tipos de métodos anticonceptivos en una sola variable:

```
bd_cmc_2 <- bd_cmc
bd_cmc_2[,"ContraconceptiveMethod"] <- ifelse(
  bd_cmc[,"ContraconceptiveMethod"] ==3 | bd_cmc[,"ContraconceptiveMethod"] ==2,
  2, 1)
```

Realizamos el modelo de árboles de decisión:

```
# Set train and test data
data_train <- create_train_test(bd_cmc_2, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc_2, 0.8, train = FALSE)

# Adjust the model
fit <- rpart(ContraconceptiveMethod~., data = data_train, method = 'class', cp=-1)

# Predict the test data class
predicted_class <-predict(fit, data_test, type='class')

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
dt_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Random Forest:

```
# Adjust the model
fit <- randomForest(as.factor(ContraconceptiveMethod)~., data = data_train,
                    ntree=500, mtry=3)

# Predict the test data class
predicted_class <-predict(fit, data_test)
```

```r
# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
rf_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Red Neuronal Artificial:

```r
# Adjust the model
nn=nnet(as.factor(ContraconceptiveMethod)~ ., data=data_train, size=2,
        maxit=300, skip=TRUE)
```

```
## # weights:  26
## initial  value 851.854009
## iter  10 value 681.184431
## iter  20 value 672.884102
## iter  30 value 671.834752
## iter  40 value 671.767690
## iter  50 value 668.661432
## iter  60 value 666.737816
## iter  70 value 666.153676
## iter  80 value 666.050172
## iter  90 value 665.980506
## iter 100 value 665.837391
## iter 110 value 665.777773
## iter 120 value 665.755355
## iter 130 value 665.730170
## iter 140 value 665.719000
## iter 150 value 665.667390
## iter 160 value 664.634641
## iter 160 value 664.634641
## final  value 664.634641
## converged
```

```r
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nn_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Naive Bayes:

```r
# Adjust the model
fit=naiveBayes(as.factor(ContraconceptiveMethod)~ ., data=data_train)
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nb_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de KNN:

```
# Extract train and test labels
train.label <- data_train[, "ContraconceptiveMethod"]
test.label <- data_test[, "ContraconceptiveMethod"]
data_train[,"ContraconceptiveMethod"] <- NULL
data_test[,"ContraconceptiveMethod"] <- NULL

# Adjust the model
predicted_class <- knn(data_train,data_test,cl=train.label,k=10)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(test.label), predicted_class)
knn_accuracy <- matrix$overall["Accuracy"]
```

Creamos la tabla de comparativas:

```
models_name <- c("Árbol de decisión", "Random Forest",
                 "Red Neuronal Artificial", "Naive Bayes", "KNN")
models_acc <- c(dt_accuracy, rf_accuracy, nn_accuracy,
                nb_accuracy, knn_accuracy)
acc_table <- data.frame(models_name, models_acc)
names(acc_table) <- c("Modelos", "Accuracy")

kable(acc_table)
```

Modelos	Accuracy
Árbol de decisión	0.7307692
Random Forest	0.7482517
Red Neuronal Artificial	0.7482517
Naive Bayes	0.7272727
KNN	0.7447552

Se puede observar que los resultados son considerablemente mejores, por lo que a continuación procedemos a examinar cuando se escoge un método u otro, para ello solo seleccionamos la instancias donde se utiliza un método anticonceptivo:

```
bd_cmc_3 <- bd_cmc
bd_cmc_3 <- bd_cmc_3[-which(bd_cmc_3$ContraconceptiveMethod == 1),]
```

Realizamos el modelo de árboles de decisión:

```
# Set train and test data
data_train <- create_train_test(bd_cmc_3, 0.8, train = TRUE)
data_test <- create_train_test(bd_cmc_3, 0.8, train = FALSE)

# Adjust the model
fit <- rpart(ContraconceptiveMethod~., data = data_train, method = 'class', cp=-1)

# Predict the test data class
predicted_class <-predict(fit, data_test, type='class')
```

```
# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
dt_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Random Forest:

```
# Adjust the model
fit <- randomForest(as.factor(ContraconceptiveMethod)~., data = data_train,
                    ntree=500, mtry=3)

# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
rf_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Red Neuronal Artificial:

```
# Adjust the model
nn=nnet(as.factor(ContraconceptiveMethod)~ ., data=data_train, size=2,
        maxit=300, skip=TRUE)
```

```
## # weights:  26
## initial  value 597.326327
## iter  10 value 406.338479
## iter  20 value 406.137457
## iter  30 value 398.827747
## iter  40 value 392.475718
## iter  50 value 387.384250
## iter  60 value 386.753325
## iter  70 value 386.356558
## iter  80 value 385.763010
## iter  90 value 382.947741
## iter 100 value 381.327823
## final  value 381.088235
## converged
```

```
# Predict the test data class
predicted_class <-predict(fit, data_test)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nn_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de Naive Bayes:

```
# Adjust the model
fit=naiveBayes(as.factor(ContraconceptiveMethod)~ ., data=data_train)
# Predict the test data class
predicted_class <-predict(fit, data_test)
```

```
# Create the confusion matrix
matrix<-confusionMatrix(as.factor(data_test$ContraconceptiveMethod), predicted_class)
nb_accuracy <- matrix$overall["Accuracy"]
```

Realizamos el modelo de KNN:

```
# Extract train and test labels
train.label <- data_train[, "ContraconceptiveMethod"]
test.label <- data_test[, "ContraconceptiveMethod"]
data_train[,"ContraconceptiveMethod"] <- NULL
data_test[,"ContraconceptiveMethod"] <- NULL

# Adjust the model
predicted_class <- knn(data_train,data_test,cl=train.label,k=10)

# Create the confusion matrix
matrix<-confusionMatrix(as.factor(test.label), predicted_class)
knn_accuracy <- matrix$overall["Accuracy"]
```

Creamos la tabla de comparativas:

```
models_name <- c("Árbol de decisión", "Random Forest",
                 "Red Neuronal Artificial", "Naive Bayes", "KNN")
models_acc <- c(dt_accuracy, rf_accuracy, nn_accuracy,
                nb_accuracy, knn_accuracy)
acc_table <- data.frame(models_name, models_acc)
names(acc_table) <- c("Modelos", "Accuracy")

kable(acc_table)
```

Modelos	Accuracy
Árbol de decisión	0.6121212
Random Forest	0.6000000
Red Neuronal Artificial	0.6000000
Naive Bayes	0.6787879
KNN	0.6727273

Se puede observar que finalmente los resultados son mejores y esta podría ser una vía de estudio interesante.

Conclusiones

Tras realizar el análisis exploratorio, realizar la eliminación de características, examinar las clasificaciones realizadas y comprender el estado del conjunto de datos, se han llegado a las siguientes conclusiones:

- Las variables más interesantes de cara a la clasificación son las de la edad de la mujer, el número de hijos y la educación de la mujer, ya que son las que se encuentran más relacionadas con ella y con la elección del método.
- Existen una serie de relaciones entre las diferentes variables:

- Las mujeres religiosas tienden a no utilizar anticonceptivos.
- Mayor edad de la mujer implica no usar anticonceptivos.
- Mayor educación de la mujer implica usar anticonceptivos, especialmente de largo plazo.
- Mayor número de hijos implica usar anticonceptivos.
- Las mujeres islámicas tienden a no utilizar anticonceptivos.
- Un nivel de vida mayor implica usar anticonceptivos.
- La educación de la mujer tiende a ser más importante que la del hombre en esta decisión.
- Por lo general la opción principal es no usar anticonceptivos, mientras que la menos deseada son los anticonceptivos a largo plazo.

Realmente se puede observar que es un problema complejo que seguramente requiere de más factores para poder determinar con mayor exactitud y poder realizar una clasificación más precisa, alguna posible sugerencia sería recabar información sobre los ingresos de la familia. Se destaca que la primera variable a considerar suele ser el número de hijos, lo cual es bastante significativo, por lo que ha sido finalmente una buena decisión excluir casos extremos.

Se han utilizado estos modelos para poder comprender y predecir, aunque en este problema la tarea de comprender el comportamiento de las mujeres indonesias es un problema realmente complejo y es más importante que realmente predecir que decisión toman, sino comprender el trasfondo de detrás del mismo.

Consideramos que es un problema interesante en la actualidad, ya que la educación sobre el tema a dichas mujeres puede suponer una ventaja que permita poseer un mayor conocimiento y poder tomas decisiones sobre la no concepción. El estudio de dicho problema debería ser materia de estudio en Indonesia de cara al empoderamiento de la mujer en países donde no existe dicha cultura, y con herramientas como este estudio se puede llegar a proporcionar determinadas conclusiones que permitan alcanzar dicho objetivo.