M2.851 - Tipología y ciclo de vida de los datos

Xuan Zheng, Albert Casanova

1. Descripción del dataset

El dataset escogido es Titanic - Machine Learning from Disaster, facilitado como ejemplo en el enunciado de la asignatura, a través del siguiente enlace:

https://www.kaggle.com/c/titanic

En este dataset encontramos información sobre los pasajeros que se encontraban a bordo en el último viaje del transatlántico RMS Titanic. Es un juego de datos "clásico" si hablamos de trabajar con algoritmos de clasificación, ya que contiene información sobre si los pasajeros sobrevivieron o no, haciéndolo interesante y siendo un buen punto de partida para practicar con este tipo de problemas.

El objetivo de trabajar con este conjunto de datos es el de comprender qué tipo de pasajeros era más probable que sobreviviera, analizando las variables de las que disponemos y tratando de dar respuesta a este problema.

2. Lectura del dataset

Hemos decidido trabajar con el fichero train.csv, ya que es el que incorpora la variable "Survived", a diferencia del fichero test.csv

Cargamos el fichero de datos.

```
titanic <- read.csv('train.csv')
```

Verificamos la estructura del juego de datos.

```
str(titanic)
```

```
## 'data.frame':    891 obs. of 12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence B
riggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ..
.
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Encontramos 891 registros, y las 12 variables que encontramos son:

- PassengerId: ID que sirve para identificar a cada uno de los pasajeros
- Survived: Indica si el pasajero sobrevivió o no. 0 = No, 1 = Yes
- Pclass: Clase a la que pertenece el pasajero, según el ticket que compró. 1 = 1st, 2 = 2nd, 3 = 3rd
- Name: Nombre del pasajero
- Sex: Genero del pasajero
- Age: Edad del pasajero
- SipSp: Número de hermanos/as o marido/mujer a bordo del Titanic
- Parch: Número de padres/madres o hijos/as a bordo del Titanic
- Ticket: Número de ticket del pasajero
- Fare: Coste del ticket
- Cabin: Camarote del pasajero
- Embarked: Puerto de embarque. C = Cherbourg, Q = Queenstown, S = Southampton

3. Limpieza de los datos

Detección de valores ceros y valores perdidos

1. Valores ceros

Con la función summary, podemos ver las columnas que contienen valores 0, por ejemplo, SibSp, Parch, Fare, etc. Pero consideramos que estos valores 0 tiene un significado real. Por lo tanto, no debemos modificarlo.

```
summary(titanic)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin              Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
```

```
##                    Max.   :512.33
##
```

2. Valores perdidos

Primero, buscamos los valores NAs que existen en el dataset e imputarlos con la mediana de edad de todos los registros.

```
colSums(is.na(titanic)) # hay valores perdidos en Age

## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0

titanic$Age[is.na(titanic$Age)] <- median(titanic$Age,na.rm=T)
```

Segundo, buscamos los strings vacíos que existen en el dataset y sustituirlos con 'Desconocido' para que queden más claros.

```
colSums(titanic=="") # hay strings vacíos en Embarked y Cabin

## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2

titanic$Embarked[titanic$Embarked==""] <- "Desconocido"
titanic$Cabin[titanic$Cabin==""] <- "Desconocido"
```

Detección de valores atípicos

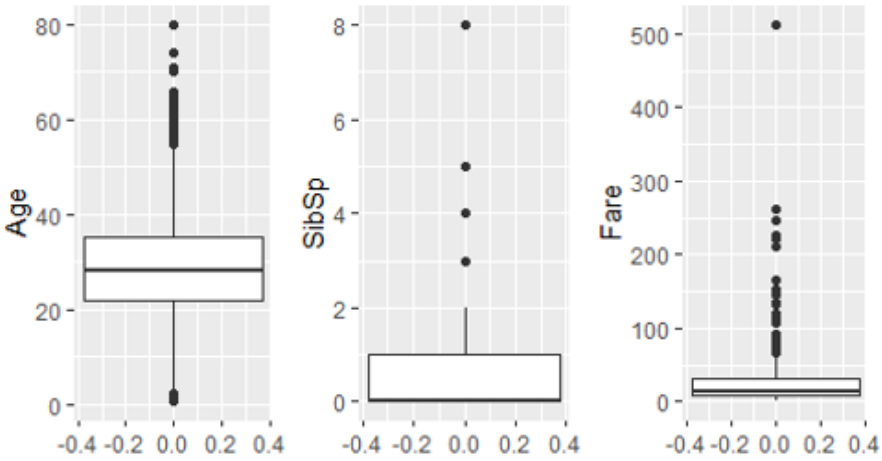
Primero, buscamos si existen valores atípicos en las variables categóricas.

```
names <- c('Survived', 'Pclass', 'Sex', 'Embarked')
for (n in names) {
  print(unique(titanic[n]))
}

##   Survived
## 1        0
## 2        1
##    Pclass
## 1       3
## 2       1
## 10      2
##      Sex
## 1   male
## 2 female
##       Embarked
## 1            S
## 2            C
## 6            Q
## 62 Desconocido
```

Vemos que las columnas categóricas no contienen valores erróneos.

Aplicamos el diagrama de caja en las columnas numéricas para encontrar los valores extremos.

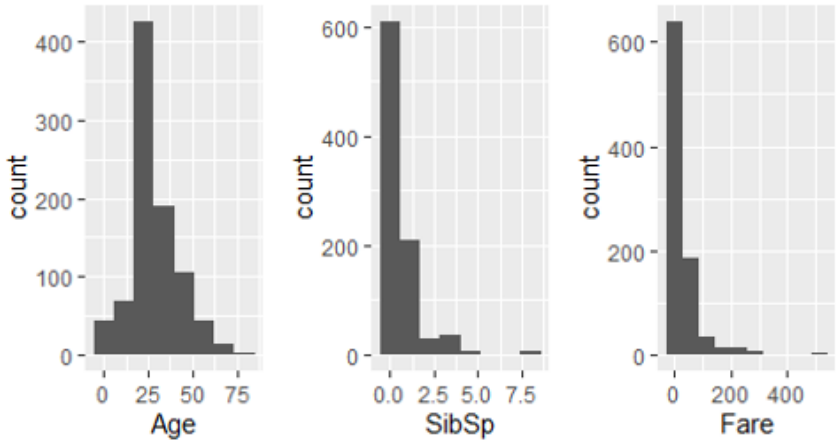
```
age_boxplot <- ggplot(data=titanic, aes(x=Age))+geom_boxplot()+ coord_flip()
sibsp_boxplot <- ggplot(data=titanic, aes(x=SibSp))+geom_boxplot()+ coord_flip()
fare_boxplot <- ggplot(data=titanic, aes(x=Fare))+geom_boxplot()+ coord_flip()

grid.arrange(age_boxplot, sibsp_boxplot, fare_boxplot, ncol = 3, heights = c(2,1))
```



Podemos ver que existen valores extremos en las columnas *Age*, *SibSp* y *Fare*.

Mostramos los histogramas para verificar la distribución de dichas columnas.

```
age_hist <- ggplot(data=titanic, aes(x=Age))+geom_histogram(bins=8)
sibsp_hist <- ggplot(data=titanic, aes(x=SibSp))+geom_histogram(bins=8)
fare_hist <- ggplot(data=titanic, aes(x=Fare))+geom_histogram(bins=10)

grid.arrange(age_hist, sibsp_hist, fare_hist, ncol = 3, heights = c(2,1))
```



Como los registros de valores extremos de *SibSp* y *Fare* son muy pocos, los eliminamos. Vemos que la distribución de *Age* más o menos sigue la normalidad.

```
titanic <- titanic[titanic$SibSp!=8, ]
titanic <- titanic[titanic$Fare<500, ]
```

4. Análisis de los datos

Vamos a eliminar variables que no aportan demasiado al estudio que estamos realizando, estas son PassengerId, Name, Ticket y Cabin.

```
titanic <- titanic %>% select(-c(PassengerId, Name, Ticket, Cabin))
```

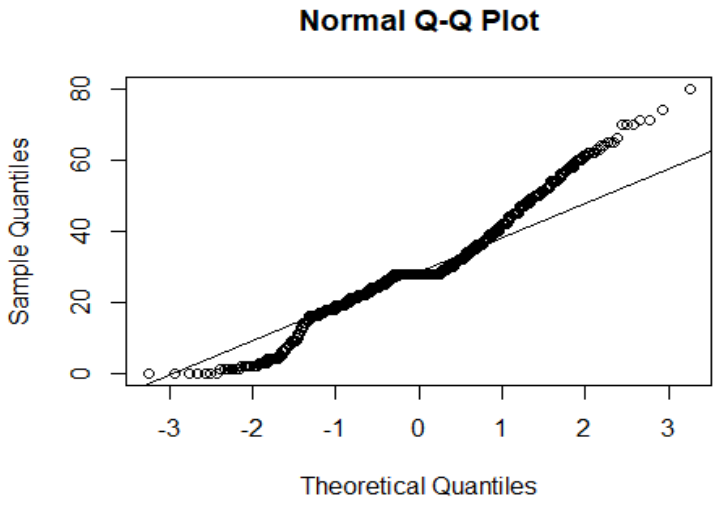
Transformamos las variables Survived, Pclass, Sex y Embarked a factor. La variable Age a numérico entero.

```
titanic$Survived<-as.factor(titanic$Survived)
titanic$Pclass<-as.factor(titanic$Pclass)
titanic$Sex<-as.factor(titanic$Sex)
titanic$Embarked<-as.factor(titanic$Embarked)
titanic$Age<-as.integer(titanic$Age)
```

Comprobación de la normalidad

Con las funciones qqnorm y qqline podemos realizar una inspección visual de las variables Age y Fare.

```
qqnorm(titanic$Age)
qqline(titanic$Age)
```



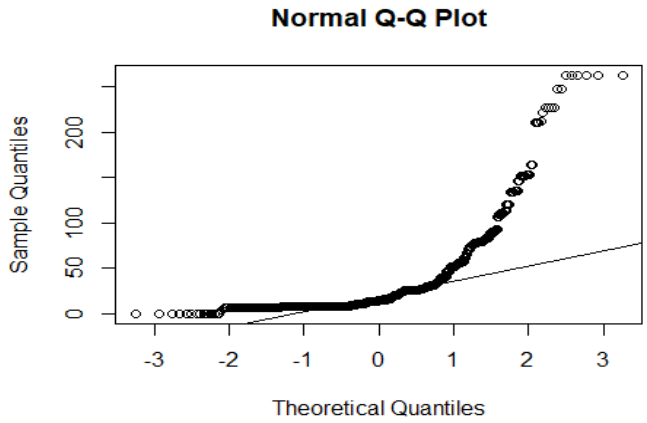
Como podemos observar, los puntos no están sobre la línea diagonal, por lo que podemos descartar normalidad en los datos.

Aplicamos el test de Lilliefors para comprobar la normalidad.

```
lillie.test(titanic$Age)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  titanic$Age
## D = 0.14675, p-value < 2.2e-16
```

Con el valor del p-value, que es menor que el nivel de significancia (por ejemplo 0.05), podemos rechazar la hipótesis nula de normalidad de los datos.

```
qqnorm(titanic$Fare)
qqline(titanic$Fare)
```



Como podemos observar, los puntos no están sobre la línea diagonal, por lo que podemos descartar normalidad en los datos.

```
lillie.test(titanic$Fare)
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  titanic$Fare
## D = 0.26105, p-value < 2.2e-16
```

Con el valor del p-value, que es menor que el nivel de significancia (por ejemplo 0.05), podemos rechazar la hipótesis nula de normalidad de los datos.

Homogeneidad de la varianza

```
bartlett.test(Fare~Survived, data=titanic)

##
##  Bartlett test of homogeneity of variances
##
## data:  Fare by Survived
## Bartlett's K-squared = 100.77, df = 1, p-value < 2.2e-16

bartlett.test(Age~Survived, data=titanic)

##
##  Bartlett test of homogeneity of variances
##
## data:  Age by Survived
## Bartlett's K-squared = 3.8861, df = 1, p-value = 0.04869
```

Bajo un nivel de significación de 0.05, los grupos de *Fare* y *Age* clasificados por *Survived* no cumplen la homogeneidad de la varianza.

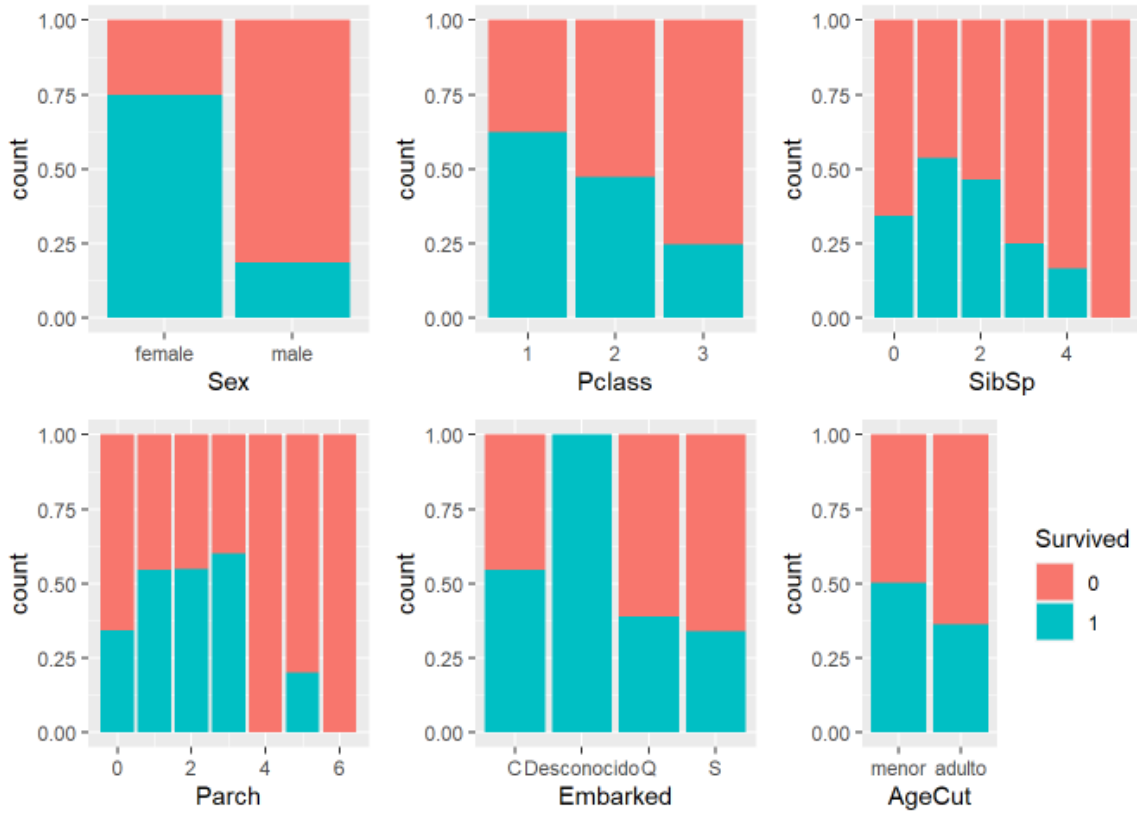
Aplicación de pruebas estadísticas para comparar los grupos de datos

Primera visualización

Para tener una idea general sobre la distribución de las variables, hacemos la primera visualización.

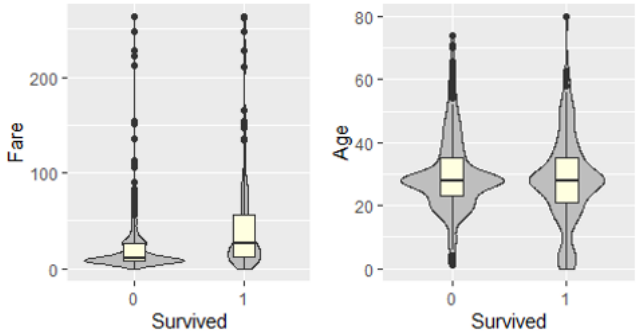
1. Visualizamos las variables categóricas contra la variable *Survived*

```
titanic$AgeCut <- cut(titanic$Age, breaks = c(-1, 18, Inf), labels = c('menor', 'adulto')
)

plot_gender <- ggplot(data=titanic, aes(x=Sex, fill=Survived))+geom_bar(position = 'fill'
, show.legend = FALSE)
plot_pclass <- ggplot(data=titanic, aes(x=Pclass, fill=Survived))+geom_bar(position = 'fi
ll', show.legend = FALSE)
plot_siblings <- ggplot(data=titanic, aes(x=SibSp, fill=Survived))+geom_bar(position = 'f
ill', show.legend = FALSE)
plot_parch <- ggplot(data=titanic, aes(x=Parch, fill=Survived))+geom_bar(position = 'fill
', show.legend = FALSE)
plot_embarked <- ggplot(data=titanic, aes(x=Embarked, fill=Survived))+geom_bar(position =
'fill', show.legend = FALSE)
plot_age_cut <- ggplot(data=titanic, aes(x=AgeCut, fill=Survived))+geom_bar(position = 'f
ill')

grid.arrange(plot_gender, plot_pclass, plot_siblings, plot_parch, plot_embarked, plot_age
_cut, ncol=3)
```



2. Visualizamos las variables numéricas contra la variable *Survived*

```
agg_mean_fare <- aggregate(titanic$Fare,by=list(titanic$Survived),FUN=mean)
plot_mean_fare <- ggplot(titanic, aes(x = Survived, y = Fare)) + geom_violin(fill = "grey
") + geom_boxplot(width = .2, fill = "lightyellow")

agg_mean_age <- aggregate(titanic$Age,by=list(titanic$Survived),FUN=mean)
plot_mean_age <- ggplot(titanic, aes(x = Survived, y = Age)) + geom_violin(fill = "grey")
+ geom_boxplot(width = .2, fill = "lightyellow")

grid.arrange(plot_mean_fare, plot_mean_age, ncol=2, heights = c(2,1))
```



Con las visualizaciones anteriores, podemos observar que existe mucha diferencia en la probabilidad de supervivencia por género, clase, edad. Para comprobar nuestras suposiciones, realizamos los contrastes de hipótesis.

Contraste de hipótesis

1. Las variables independientes categóricas

```
# Usamos la medida de Cramer V para encontrar la asociación entre las 2 var. nominales
gender_survived <- table(titanic$Survived, titanic$Sex)
cramerV(gender_survived) # rcompanion

## Cramer V
##   0.5506

pclass_survived <- table(titanic$Survived, titanic$Pclass)
cramerV(pclass_survived)

## Cramer V
##   0.3321

embarked_survived <- table(titanic$Survived, titanic$Embarked)
cramerV(embarked_survived)

## Cramer V
##   0.1732

sibling_survived <- table(titanic$Survived, titanic$SibSp)
cramerV(sibling_survived)

## Cramer V
##   0.1951
```

```
parch_survived <- table(titanic$Survived, titanic$Parch)
cramerV(parch_survived)

## Cramer V
##   0.1886
```

Sabemos que los valores de Cramer V entre 0.1 y 0.3 nos indican que la asociación estadística es baja, y entre 0.3 y 0.5 se puede considerar una asociación media. Finalmente, si los valores fueran superiores a 0.5 (no es el caso), la asociación estadística entre las variables sería alta.

Por lo tanto, las variables de *Sex* y *Pclass* tienen una asociación más alta entre todas.

2. ANOVA

```
# Usamos ANOVA para determinar si las variables son significantes
gmodel <- glm(Survived ~ . - AgeCut, data = titanic, family = binomial(link = logit))
anova(gmodel, test = "Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       880    1174.13
## Pclass    2   97.738       878    1076.39 < 2.2e-16 ***
## Sex       1  259.603       877     816.79 < 2.2e-16 ***
## Age       1   20.706       876     796.08 5.355e-06 ***
## SibSp     1    8.881       875     787.20  0.002881 **
## Parch     1    0.566       874     786.63  0.451759
## Fare      1    0.044       873     786.59  0.834267
## Embarked  3    4.777       870     781.81  0.188908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como hemos observado anteriormente, *PClass*, *Sex*, *Age* han demostrado un nivel más significativo entre todas las variables independientes.

Regresión logística y predicción

```
# Separamos en dataset de train y de test
set.seed(123)
split = sort(sample(nrow(titanic), nrow(titanic)*0.8))

train <- titanic[split,]
test <- titanic[-split,]
```

El primer modelo de regresión logística contará con las variables *PClass*, *Sex*, y *Age*, que son las más relevantes.

```r
model_logist <- glm(Survived ~ Pclass + Sex + Age , data = train, family ="binomial")

summary(model_logist)
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6502  -0.6354  -0.4141   0.6330   2.4655
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.549454   0.407718   8.706  < 2e-16 ***
## Pclass2     -1.151023   0.293513  -3.922  8.8e-05 ***
## Pclass3     -2.320968   0.271221  -8.557  < 2e-16 ***
## Sexmale     -2.691944   0.213160 -12.629  < 2e-16 ***
## Age         -0.033927   0.008294  -4.091  4.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 928.36  on 703  degrees of freedom
## Residual deviance: 622.27  on 699  degrees of freedom
## AIC: 632.27
##
## Number of Fisher Scoring iterations: 5
```

Vemos que, con la selección de las variables más relevantes, el criterio AIC es de 632.27.

```r
# Realizamos predicción sobre el test
pred_test = predict(model_logist, test, type = "response")

# Definimos a partir de qué probabilidad queremos que asigne a 1 o a 0 y mostramos la mat
riz de confusión
pred_test = as.factor(ifelse(pred_test >= 0.5, yes = 1, no = 0))
confusionMatrix(test$Survived, pred_test)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 82 17
##          1 25 53
##
##                Accuracy : 0.7627
##                  95% CI : (0.6931, 0.8233)
##     No Information Rate : 0.6045
##     P-Value [Acc > NIR] : 6.374e-06
##
##                   Kappa : 0.5134
##
##  Mcnemar's Test P-Value : 0.2801
##
##             Sensitivity : 0.7664
##             Specificity : 0.7571
##          Pos Pred Value : 0.8283
##          Neg Pred Value : 0.6795
##              Prevalence : 0.6045
##          Detection Rate : 0.4633
```

```
##    Detection Prevalence : 0.5593
##       Balanced Accuracy : 0.7617
##
##        'Positive' Class : 0
##
```

Obtenemos una precisión del 76,27%

Creamos un segundo modelo de regresión logística, esta vez con todas las variables excepto *AgeCut*.

```
# Creamos el modelo de regresión logística
model_logist2 <- glm(Survived ~ . - AgeCut , data = train, family ="binomial")
summary(model_logist2)
```

```
##
## Call:
## glm(formula = Survived ~ . - AgeCut, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5787  -0.6051  -0.4014   0.6049   2.5261
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.320224   0.559344   7.724 1.13e-14 ***
## Pclass2              -1.174515   0.349975  -3.356 0.000791 ***
## Pclass3              -2.463998   0.359270  -6.858 6.97e-12 ***
## Sexmale              -2.784245   0.230563 -12.076  < 2e-16 ***
## Age                  -0.041881   0.008995  -4.656 3.22e-06 ***
## SibSp                -0.279164   0.131164  -2.128 0.033307 *
## Parch                -0.077207   0.130971  -0.589 0.555527
## Fare                 -0.001245   0.003532  -0.353 0.724433
## EmbarkedDesconocido  11.942039 535.411305   0.022 0.982205
## EmbarkedQ             0.313922   0.421101   0.745 0.455982
## EmbarkedS            -0.325907   0.268923  -1.212 0.225552
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 928.36  on 703  degrees of freedom
## Residual deviance: 608.79  on 693  degrees of freedom
## AIC: 630.79
##
## Number of Fisher Scoring iterations: 12
```

El resultado del criterio de AIC es más bajo, de 630.79. Por lo tanto, escogeríamos el segundo modelo.

```
# Realizamos predicción sobre el test
pred_test2 = predict(model_logist2, test, type = "response")

# Definimos a partir de qué probabilidad queremos que asigne a 1 o a 0 y mostramos la mat
riz de confusión
pred_test2 = as.factor(ifelse(pred_test2 >= 0.5, yes = 1, no = 0))
confusionMatrix(test$Survived, pred_test2)
```

```
## Confusion Matrix and Statistics
##
```

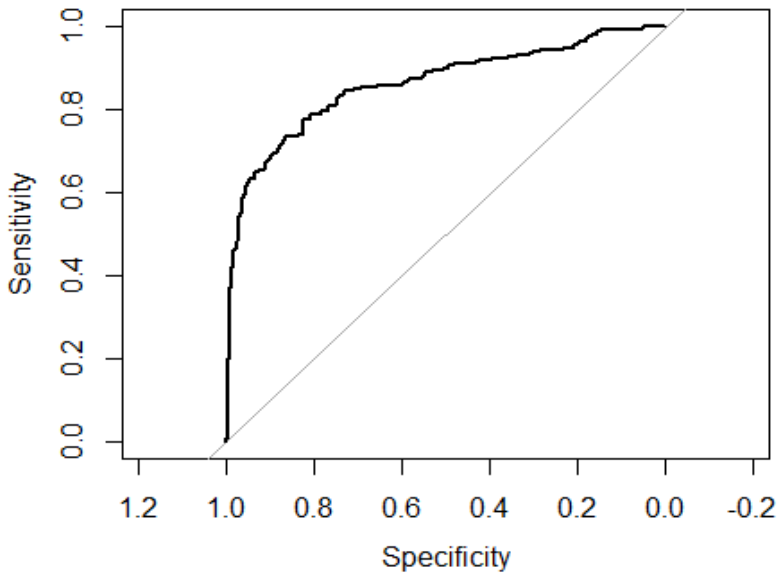
```
##           Reference
## Prediction  0  1
##          0 86 13
##          1 24 54
##
##                Accuracy : 0.791
##                  95% CI : (0.7236, 0.8483)
##     No Information Rate : 0.6215
##     P-Value [Acc > NIR] : 9.598e-07
##
##                   Kappa : 0.5695
##
##  Mcnemar's Test P-Value : 0.1002
##
##             Sensitivity : 0.7818
##             Specificity : 0.8060
##          Pos Pred Value : 0.8687
##          Neg Pred Value : 0.6923
##              Prevalence : 0.6215
##          Detection Rate : 0.4859
##    Detection Prevalence : 0.5593
##       Balanced Accuracy : 0.7939
##
##        'Positive' Class : 0
```

La precisión también mejora, hemos obtenido una precisión del 79.1%.

ROC y AUC

Con los datos del segundo modelo, que es el con el que mejor precisión obtenemos, dibujamos la curva ROC.

```
prob=predict(model_logist2, train, type="response")
r=roc(train$Survived, prob, data=train)

plot(r)
```



Calculamos el área debajo de la curva.

```
auc(r)

## Area under the curve: 0.8592
```

Con los resultados del AUC, podemos decir que el modelo discrimina de forma excelente.

Árbol de decisión

```
# Separamos las variables independientes y la variable dependiente
trainX <- train[2:8]
trainy <- as.factor(train[,1])
testX <- test[2:8]
testy <- as.factor(test[,1])
```

Primero, usamos todas las variables originales para construir un árbol de decisión, y observamos las reglas de clasificación.

```
tree <- C50::C5.0(trainX, trainy, rules = TRUE)
summary(tree)

##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jun 06 18:24:51 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 704 cases (8 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (14, lift 1.5)
##  Pclass = 3
##  Age > 30
##  Embarked in {C, Q}
##  ->  class 0  [0.938]
##
## Rule 2: (187/23, lift 1.4)
##  Pclass = 3
##  Fare <= 10.5167
##  Embarked = S
##  ->  class 0  [0.873]
##
## Rule 3: (27/3, lift 1.4)
##  Pclass = 3
##  Sex = female
##  Fare > 17.4
##  Embarked = S
##  ->  class 0  [0.862]
##
## Rule 4: (462/82, lift 1.3)
##  Sex = male
##  ->  class 0  [0.821]
##
```

```
## Rule 5: (15, lift 2.5)
##  Sex = male
##  Age <= 9
##  SibSp <= 2
##  ->  class 1  [0.941]
##
## Rule 6: (128/7, lift 2.5)
##  Pclass in {1, 2}
##  Sex = female
##  ->  class 1  [0.938]
##
## Rule 7: (27/2, lift 2.4)
##  Sex = female
##  Fare > 10.5167
##  Fare <= 17.4
##  Embarked = S
##  ->  class 1  [0.897]
##
## Rule 8: (17/1, lift 2.4)
##  Pclass = 1
##  SibSp > 0
##  Parch <= 0
##  Embarked = C
##  ->  class 1  [0.895]
##
## Rule 9: (68/11, lift 2.2)
##  Sex = female
##  Age <= 30
##  Embarked in {C, Q}
##  ->  class 1  [0.829]
##
## Rule 10: (15/2, lift 2.2)
##  Pclass = 1
##  Age <= 54
##  Fare > 26
##  Fare <= 30.6958
##  Embarked = S
##  ->  class 1  [0.824]
##
## Default class: 0
##
##
## Evaluation on training data (704 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##      10   86(12.2%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     420    23    (a): class 0
##      63   198    (b): class 1
##
##
##  Attribute usage:
##
##   96.16% Sex
##   53.13% Pclass
```

```
##   49.57% Embarked
##   36.36% Fare
##   15.91% Age
##    4.55% SibSp
##    2.41% Parch
##
##
## Time: 0.0 secs
```

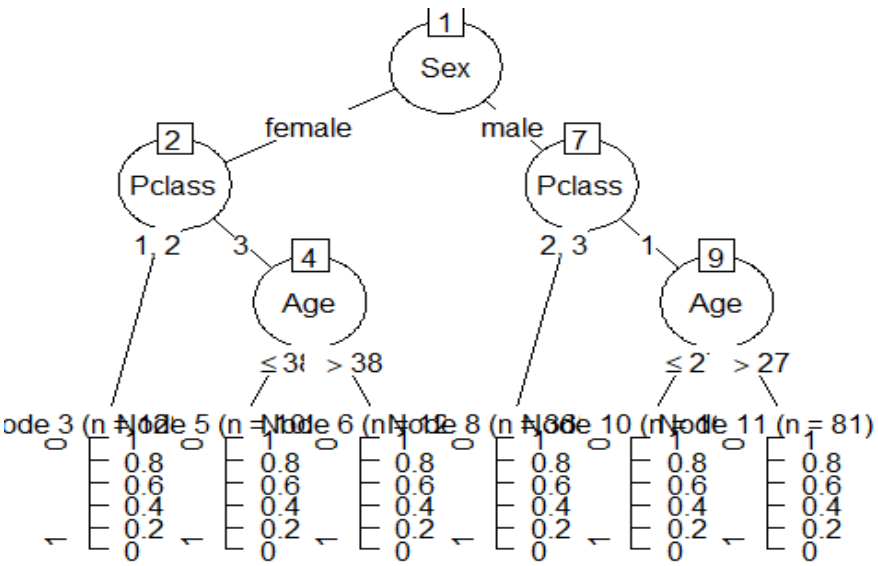
Generamos predicciones con el conjunto de test.

```
pred <- predict(tree, testX, type='class')

table(testy, Predicted=pred)

##      Predicted
## testy  0  1
##     0 89 10
##     1 27 51

sum(pred==testy)/length(pred)

## [1] 0.7909605
```

Obtenemos una precisión del 79.1%.

Para visualizar mejor las reglas importantes del árbol, volvemos a construir el modelo con las variables *PClass*, *Sex*, y *Age* y mostramos el árbol.

```
trainX_selected <- train[2:4]
tree_selected <- C50::C5.0(trainX_selected, trainy)

plot(tree_selected)
```



Generación .csv de los datos finales analizados

Con la siguiente función guardamos el archivo con los datos finales en un nuevo fichero .csv con nombre “train_clean.csv”

```
write.csv(titanic, "train_clean.csv", row.names = FALSE)
```

5. Conclusiones

Tras un primer análisis exploratorio de las variables que conforman el dataset, analizamos los valores cero, los valores perdidos y vacíos, así como los valores extremos.

En cuanto a los valores cero, no decidimos modificarlos ya que consideramos que tienen sentido, por ejemplo, aparecen en las variables SibSp, Parch, Fare, … Para los valores perdidos, vemos que únicamente aparecen en la variable Age, por lo que imputamos la mediana de la edad en los registros con “NA’s”. En los strings vacíos, que aparecen en las variables Cabin y Embarked, sustituímos por “Desconocido”. Analizando los valores extremos, decidimos eliminar los de las variables SibSp y Fare.

En cuanto al análisis de normalidad, encontramos que tanto la variable Age y como Fare, que son las variables numéricas que hemos mantenido en el dataset, no siguen una distribución normal de los datos. Tampoco cumplen la homogeneidad de la varianza.

Hemos realizado visualizaciones de las variables categóricas y numéricas contra la variable objetivo Survived, donde observamos que existe diferencia según el género, la edad, o la clase a la que pertenecía el pasajero. Tras realizar contrastes de hipótesis concluimos que las variables Sex, Age y Pclass tienen son las más significativas de las variables independientes.

Con estos datos, hemos generado predicciones mediante regresiones logísticas y árboles de decisión. Hemos realizado un primer modelo de regresión logística únicamente con las variables mencionadas anteriormente, pero también un segundo con la totalidad de las variables del dataset. Se obtiene una mejor precisión con la totalidad de las variables, pasando de 76,27% a 79,1%. También ha mejorado el valor del criterio de Akaike, de 632,27 a 630,79. Para el segundo modelo, el de mayor precisión, se muestra la curva ROC y el valor auc, de 0.8592, por lo que el modelo discrimina de forma excelente. Para el árbol de decisión hemos utilizado todas las variables originales, y observando las reglas de clasificación, las variables más utilizadas son Sex, Pclass, Embarked, Fare y Age. En las predicciones hemos obtenido un 79,1% de precisión. Finalmente, visualizamos el árbol de decisión con las variables más importantes a la hora de generar reglas.

Por lo tanto, respondiendo a la pregunta que se planteaba como objetivo de este análisis, era más probable sobrevivir en caso de ser mujer, menor de edad, o pertenecer a primera clase.

Contribuciones	Firma
Investigación previa	XZ, AC
Redacción de las respuestas	XZ, AC
Desarrollo del código	XZ, AC