Actividad Integradora 2

José Carlos Sánchez Gómez

2024-11-19

```r
# Cargamos todas las librería en la lista "librerias"
librerias =
c('tidyverse','broom','ISLR','GGally','modelr','cowplot','rlang','modelr'
,'tibble','Metrics','mice','visdat',"caret")

for (lib in librerias){
  library(lib,character.only=TRUE)}
```

```
## ── Attaching core tidyverse packages ────────────────────────
tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
##
## Adjuntando el paquete: 'modelr'
##
##
## The following object is masked from 'package:broom':
##
##     bootstrap
##
##
##
## Adjuntando el paquete: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
##
##
##
## Adjuntando el paquete: 'rlang'
##
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
##
##
##
## Adjuntando el paquete: 'Metrics'
##
##
## The following object is masked from 'package:rlang':
##
##     ll
##
##
## The following objects are masked from 'package:modelr':
##
##     mae, mape, mse, rmse
##
##
##
## Adjuntando el paquete: 'mice'
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
##
##
## Cargando paquete requerido: lattice
##
##
## Adjuntando el paquete: 'caret'
##
##
## The following objects are masked from 'package:Metrics':
##
##     precision, recall
##
```

```
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
data = read.csv("C:\\Users\\jcsg6\\Downloads\\Titanic.csv")
data_test = read.csv("C:\\Users\\jcsg6\\Downloads\\Titanic_test.csv")

str(data)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Survived   : int  0 1 0 0 1 0 1 0 1 0 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen
Needs)" "Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Preparación de los datos

Eliminación de variables no significativas

```
cleaned_data = data[, c(-1, -4, -9, -11)]
```

Transformación de variables a factores

```
for(var in c('Survived', 'Pclass', 'Embarked', 'Sex'))
  cleaned_data[, var] = as.factor(cleaned_data[, var])
```

Análisis de datos faltantes

```
V = matrix(NA,ncol = 1, nrow = 8)
for(i in c(1:8)){
  V[i, ] = sum(with(cleaned_data, cleaned_data[, i]) == "")
}
V
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]   NA
## [5,]    0
## [6,]    0
## [7,]   NA
## [8,]   NA
```

No se encuentran variables con algún espacio vacio, pero si hay algunas con valores faltantes.

```
N = apply(X=is.na(cleaned_data),MARGIN = 2,FUN = sum)
P = round(100*N/length(cleaned_data[,2]),2)
NP = data.frame(as.numeric(N),as.numeric(P))
row.names(NP)= c("Survived", "Pclass", "Sex", "Age", "SibSp", "Parch",
"Fare", "Embarked")
names(NP)=c("Número","Porcentaje")
t(NP)

##            Survived Pclass Sex    Age SibSp Parch Fare Embarked
## Número            0      0   0 263.00     0     0 1.00     2.00
## Porcentaje        0      0   0  20.09     0     0 0.08     0.15
```

De las variables con datos faltantes, la edad es la que tiene mayores datos faltantes con un 20% de los datos.

```
nrow(cleaned_data)

## [1] 1309
```

Si se eliminan los 263 registros que tienen valores nulos, nos quedariamos con un total de 1043 valores. Los cuales pueden ser suficientes para obtener un modelo que de valores significativos.

```
summary(cleaned_data)

##  Survived Pclass      Sex           Age             SibSp           
Parch
##  0:815    1:323   female:466   Min.   : 0.17   Min.   :0.0000   Min.   
:0.000
##  1:494    2:277   male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st 
Qu.:0.000
##           3:709                Median :28.00   Median :0.0000   Median 
:0.000
##                                Mean   :29.88   Mean   :0.4989   Mean   
:0.385
##                                3rd Qu.:39.00   3rd Qu.:1.0000   3rd 
Qu.:0.000
##                                Max.   :80.00   Max.   :8.0000   Max.   
:9.000
##                                NA's   :263                              
##       Fare          Embarked
##  Min.   :  0.000   C   :270
##  1st Qu.:  7.896   Q   :123
##  Median : 14.454   S   :914
##  Mean   : 33.295   NA's:  2
##  3rd Qu.: 31.275
##  Max.   :512.329
##  NA's   :1
```

```r
data_no_na = na.omit(cleaned_data)
summary(data_no_na)
```

```
##  Survived Pclass      Sex           Age             SibSp
##  0:628    1:282   female:386   Min.   : 0.17   Min.   :0.0000
##  1:415    2:261   male  :657   1st Qu.:21.00   1st Qu.:0.0000
##           3:500                Median :28.00   Median :0.0000
##                                Mean   :29.81   Mean   :0.5043
##                                3rd Qu.:39.00   3rd Qu.:1.0000
##                                Max.   :80.00   Max.   :8.0000
##      Parch             Fare        Embarked
##  Min.   :0.0000   Min.   :  0.00   C:212
##  1st Qu.:0.0000   1st Qu.:  8.05   Q: 50
##  Median :0.0000   Median : 15.75   S:781
##  Mean   :0.4219   Mean   : 36.60
##  3rd Qu.:1.0000   3rd Qu.: 35.08
##  Max.   :6.0000   Max.   :512.33
```

Eliminando las filas con valores nulos, los valores no difieren mucho entre ellos. Además de que la proporcion entre ellos sigue manteniendose igual. i.e. Las medidas estadisticas de cada variable se mantiene igual.

```r
t2c = 100*prop.table(table(cleaned_data[,1]))
t2s = 100*prop.table(table(data_no_na[,1]))
t2p = c(t2s[1]/t2c[1],t2s[2]/t2c[2])
t2 = data.frame(as.numeric(t2c),as.numeric(t2s),as.numeric(t2p))
row.names(t2) = c("Murió","Sobrevivió")
names(t2) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t2,2)
```

```
##            Con NA (%) Sin NA (%) Pérdida (prop)
## Murió           62.26      60.21           0.97
## Sobrevivió      37.74      39.79           1.05
```

```r
t3c = 100*prop.table(table(cleaned_data[,2]))
t3s = 100*prop.table(table(data_no_na[,2]))
t3p = c(t3s[1]/t3c[1],t3s[2]/t3c[2],t3s[3]/t3c[3])
t3 = data.frame(as.numeric(t3c),as.numeric(t3s),as.numeric(t3p))
row.names(t3) = c("Primera","Segunda","Tercera")
names(t3) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t3,2)
```

```
##         Con NA (%) Sin NA (%) Pérdida (prop)
## Primera      24.68      27.04           1.10
## Segunda      21.16      25.02           1.18
## Tercera      54.16      47.94           0.89
```

```r
t4c = 100*prop.table(table(cleaned_data[,3]))
t4s = 100*prop.table(table(data_no_na[,3]))
t4p = c(t4s[1]/t4c[1],t4s[2]/t4c[2])
t4 = data.frame(as.numeric(t4c),as.numeric(t4s),as.numeric(t4p))
```

```r
row.names(t4) = c("Mujer","Hombre")
names(t4) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t4,2)
```

```
##        Con NA (%) Sin NA (%) Pérdida (prop)
## Mujer        35.6      37.01           1.04
## Hombre       64.4      62.99           0.98
```

```r
t9c = 100*prop.table(table(cleaned_data[,8]))
t9s = 100*prop.table(table(data_no_na[,8]))
t9p = c(t9s[1]/t9c[1],t9s[2]/t9c[2],t9s[3]/t9c[3])
t9 = data.frame(as.numeric(t9c),as.numeric(t9s),as.numeric(t9p))
row.names(t9) = c("Cherbourg","Queenstown","Southampton")
names(t9) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t9,2)
```

```
##             Con NA (%) Sin NA (%) Pérdida (prop)
## Cherbourg        20.66      20.33           0.98
## Queenstown        9.41       4.79           0.51
## Southampton      69.93      74.88           1.07
```

La variable que más se ve afectada por la eliminación de valores nulos es la de la clase del pasajero. Ya que, a pesar de que el valor de la pérdida no es grande a comparación de las demás variables, es el que más cambia su distribución de valores.

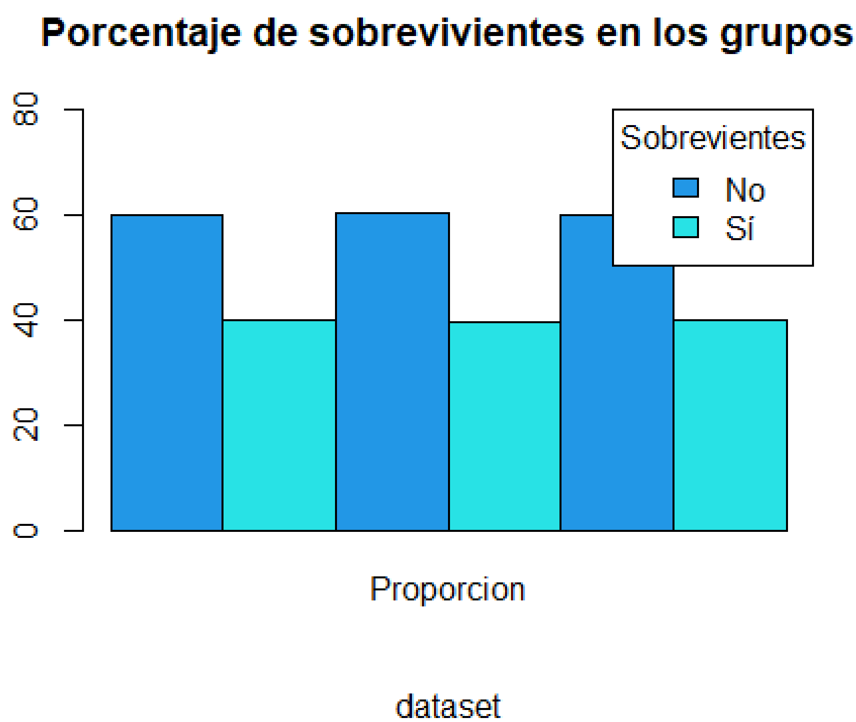
Análisis descriptivo

Partición de datos. Entrenamientos y pruebas

```r
data_partition = createDataPartition(data_no_na$Survived, p = .7, list =
FALSE, times = 1)

data_train = data_no_na[ data_partition,]  %>% as_tibble()
data_valid = data_no_na[-data_partition,] %>% as_tibble()

survived_train = 100*prop.table(table(data_train[,1]))
survived_valid = 100*prop.table(table(data_valid[,1]))
survived_all = 100*prop.table(table(data_no_na[,1]))

TablaComparativa = data.frame(
  Proporcion = c(survived_train, survived_valid, survived_all)
)
print(TablaComparativa)
```

```
##   Proporcion
## 1   60.19152
## 2   39.80848
## 3   60.25641
## 4   39.74359
## 5   60.21093
## 6   39.78907
```

```r
barplot(as.matrix(TablaComparativa), col=4:5, beside=TRUE,
main="Porcentaje de sobrevivientes en los grupos",
sub="dataset",ylim=c(0,80))
legend("topright",legend = c("No","Sí"), title = "Sobrevientes",fill =
4:5)
```



Se puede ver que la proporcion de sobrevivientes se mantiene incluso tras partir los datos en sets de datos diferentes.

```r
A = glm(Survived ~ ., data = data_train, family = "binomial")

step(A, direction = "both", trace = 1)

## Start:  AIC=579.24
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked
##
##            Df Deviance    AIC
## - Embarked  2   559.91 575.91
## - Fare      1   559.33 577.33
## - Parch     1   560.65 578.65
## <none>          559.24 579.24
## - SibSp     1   563.54 581.54
## - Age       1   571.04 589.04
## - Pclass    2   588.93 604.93
## - Sex       1   890.30 908.30
##
## Step:  AIC=575.91
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
```

```
##
##             Df Deviance    AIC
## - Fare      1   560.11 574.11
## - Parch     1   561.29 575.29
## <none>          559.91 575.91
## - SibSp     1   564.56 578.56
## + Embarked  2   559.24 579.24
## - Age       1   572.03 586.03
## - Pclass    2   591.94 603.94
## - Sex       1   894.82 908.82
##
## Step:  AIC=574.11
## Survived ~ Pclass + Sex + Age + SibSp + Parch
##
##             Df Deviance    AIC
## - Parch     1   561.31 573.31
## <none>          560.11 574.11
## + Fare      1   559.91 575.91
## - SibSp     1   564.61 576.61
## + Embarked  2   559.33 577.33
## - Age       1   572.44 584.44
## - Pclass    2   613.33 623.33
## - Sex       1   897.65 909.65
##
## Step:  AIC=573.31
## Survived ~ Pclass + Sex + Age + SibSp
##
##             Df Deviance    AIC
## <none>          561.31 573.31
## + Parch     1   560.11 574.11
## + Fare      1   561.29 575.29
## + Embarked  2   560.65 576.65
## - SibSp     1   568.03 578.03
## - Age       1   572.83 582.83
## - Pclass    2   613.91 621.91
## - Sex       1   902.72 912.72

##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family =
"binomial",
##     data = data_train)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age
SibSp
##     4.00206     -1.24783     -2.05145     -3.59991     -0.02853     -
0.32517
##
## Degrees of Freedom: 730 Total (i.e. Null);  725 Residual
```

```
## Null Deviance:       982.8
## Residual Deviance: 561.3     AIC: 573.3
```

```r
B = glm(Survived ~ Pclass * Sex * Age * Fare, data = data_train, family =
"binomial")
step(B, direction = "both", trace = 1)
```

```
## Start:  AIC=542.64
## Survived ~ Pclass * Sex * Age * Fare
##
##                         Df Deviance    AIC
## <none>                       494.64 542.64
## - Pclass:Sex:Age:Fare  2   503.07 547.07

##
## Call:  glm(formula = Survived ~ Pclass * Sex * Age * Fare, family =
"binomial",
##     data = data_train)
##
## Coefficients:
##             (Intercept)                  Pclass2
Pclass3
##               0.7095599                3.8010433
1.4152880
##                 Sexmale                      Age
Fare
##              -0.1702467                0.1673174                -
0.0122024
##          Pclass2:Sexmale          Pclass3:Sexmale
Pclass2:Age
##              -9.2851052               -3.1230695                -
0.2503726
##             Pclass3:Age              Sexmale:Age
Pclass2:Fare
##              -0.1599855               -0.2062273                -
0.0095276
##            Pclass3:Fare             Sexmale:Fare
Age:Fare
##              -0.0479947                0.0101199
0.0002871
##      Pclass2:Sexmale:Age      Pclass3:Sexmale:Age
Pclass2:Sexmale:Fare
##               0.4326797                0.1549112
0.2604976
##     Pclass3:Sexmale:Fare         Pclass2:Age:Fare
Pclass3:Age:Fare
##               0.0121440                0.0014845                -
0.0034266
##         Sexmale:Age:Fare  Pclass2:Sexmale:Age:Fare
Pclass3:Sexmale:Age:Fare
##              -0.0002549               -0.0155925
```

```
0.0059135
##
## Degrees of Freedom: 730 Total (i.e. Null);  707 Residual
## Null Deviance:       982.8
## Residual Deviance: 494.6     AIC: 542.6
```

```r
summary(B)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass * Sex * Age * Fare, family =
"binomial",
##     data = data_train)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.7095599  6.5685629   0.108   0.9140
## Pclass2                  3.8010433  7.2915599   0.521   0.6022
## Pclass3                  1.4152880  6.6607990   0.212   0.8317
## Sexmale                 -0.1702467  6.6666493  -0.026   0.9796
## Age                      0.1673174  0.3093651   0.541   0.5886
## Fare                    -0.0122024  0.0414878  -0.294   0.7687
## Pclass2:Sexmale         -9.2851052  7.6721128  -1.210   0.2262
## Pclass3:Sexmale         -3.1230695  6.8146652  -0.458   0.6467
## Pclass2:Age             -0.2503726  0.3213492  -0.779   0.4359
## Pclass3:Age             -0.1599855  0.3130730  -0.511   0.6093
## Sexmale:Age             -0.2062273  0.3106308  -0.664   0.5068
## Pclass2:Fare            -0.0095276  0.1369550  -0.070   0.9445
## Pclass3:Fare            -0.0479947  0.0749616  -0.640   0.5220
## Sexmale:Fare             0.0101199  0.0426687   0.237   0.8125
## Age:Fare                 0.0002871  0.0019180   0.150   0.8810
## Pclass2:Sexmale:Age      0.4326797  0.3341634   1.295   0.1954
## Pclass3:Sexmale:Age      0.1549112  0.3160169   0.490   0.6240
## Pclass2:Sexmale:Fare     0.2604976  0.1712124   1.521   0.1281
## Pclass3:Sexmale:Fare     0.0121440  0.0873788   0.139   0.8895
## Pclass2:Age:Fare         0.0014845  0.0042802   0.347   0.7287
## Pclass3:Age:Fare        -0.0034266  0.0038939  -0.880   0.3789
## Sexmale:Age:Fare        -0.0002549  0.0019354  -0.132   0.8952
## Pclass2:Sexmale:Age:Fare -0.0155925  0.0071030  -2.195   0.0281 *
## Pclass3:Sexmale:Age:Fare  0.0059135  0.0042673   1.386   0.1658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 982.80  on 730  degrees of freedom
## Residual deviance: 494.64  on 707  degrees of freedom
## AIC: 542.64
##
## Number of Fisher Scoring iterations: 9
```

```r
C = glm(Survived ~ ., family = "binomial", data = data_train)
step(C, direction = "both", trace = 1)
```

```
## Start:  AIC=579.24
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked
##
##            Df Deviance    AIC
## - Embarked  2   559.91 575.91
## - Fare      1   559.33 577.33
## - Parch     1   560.65 578.65
## <none>          559.24 579.24
## - SibSp     1   563.54 581.54
## - Age       1   571.04 589.04
## - Pclass    2   588.93 604.93
## - Sex       1   890.30 908.30
##
## Step:  AIC=575.91
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
##
##            Df Deviance    AIC
## - Fare      1   560.11 574.11
## - Parch     1   561.29 575.29
## <none>          559.91 575.91
## - SibSp     1   564.56 578.56
## + Embarked  2   559.24 579.24
## - Age       1   572.03 586.03
## - Pclass    2   591.94 603.94
## - Sex       1   894.82 908.82
##
## Step:  AIC=574.11
## Survived ~ Pclass + Sex + Age + SibSp + Parch
##
##            Df Deviance    AIC
## - Parch     1   561.31 573.31
## <none>          560.11 574.11
## + Fare      1   559.91 575.91
## - SibSp     1   564.61 576.61
## + Embarked  2   559.33 577.33
## - Age       1   572.44 584.44
## - Pclass    2   613.33 623.33
## - Sex       1   897.65 909.65
##
## Step:  AIC=573.31
## Survived ~ Pclass + Sex + Age + SibSp
##
##            Df Deviance    AIC
## <none>          561.31 573.31
## + Parch     1   560.11 574.11
## + Fare      1   561.29 575.29
## + Embarked  2   560.65 576.65
```

```
## - SibSp     1   568.03 578.03
## - Age       1   572.83 582.83
## - Pclass    2   613.91 621.91
## - Sex       1   902.72 912.72

##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family =
"binomial",
##     data = data_train)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age
SibSp
##     4.00206     -1.24783     -2.05145     -3.59991     -0.02853     -
0.32517
##
## Degrees of Freedom: 730 Total (i.e. Null);  725 Residual
## Null Deviance:       982.8
## Residual Deviance: 561.3     AIC: 573.3
```

Analisis de los modelos

```
comparacion = data.frame(
  Modelo = c("A", "B"),
  AIC = c(A$aic, B$aic),
  Deviance = c(A$deviance, B$deviance),
  NullDeviance = c(A$null.deviance, B$null.deviance)
)
print(comparacion)

##   Modelo      AIC Deviance NullDeviance
## 1      A 579.2445 559.2445     982.7966
## 2      B 542.6417 494.6417     982.7966
```

La null deviance se mantiene igual a través de los dos modelos. La AIC y Deviance decrementó en el modelo con interacción entre sus variables.

Desviación explicada

```
cat("A pseudo r^2:", 1 - (A$deviance / A$null.deviance), "\n")

## A pseudo r^2: 0.4309662

cat("B pseudo r^2:", 1 - (B$deviance / B$null.deviance))

## B pseudo r^2: 0.4966998
```

El modelo B obtuvo una mayor desviación explicada

Prueba de razón de verisimilitud

```
Diferencia = A$null.deviance-A$deviance
gl = A$df.null - A$df.deviance
```

```
pchisq(Diferencia,gl,lower.tail = FALSE)

## numeric(0)

Diferencia = B$null.deviance-B$deviance
gl = B$df.null - B$df.deviance

pchisq(Diferencia,gl,lower.tail = FALSE)

## numeric(0)

library(car)

## Cargando paquete requerido: carData

##
## Adjuntando el paquete: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

anova(A, B,test="LR")

## Analysis of Deviance Table
##
## Model 1: Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare +
Embarked
## Model 2: Survived ~ Pclass * Sex * Age * Fare
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       721     559.24
## 2       707     494.64 14   64.603 1.801e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modelo Seleccionado

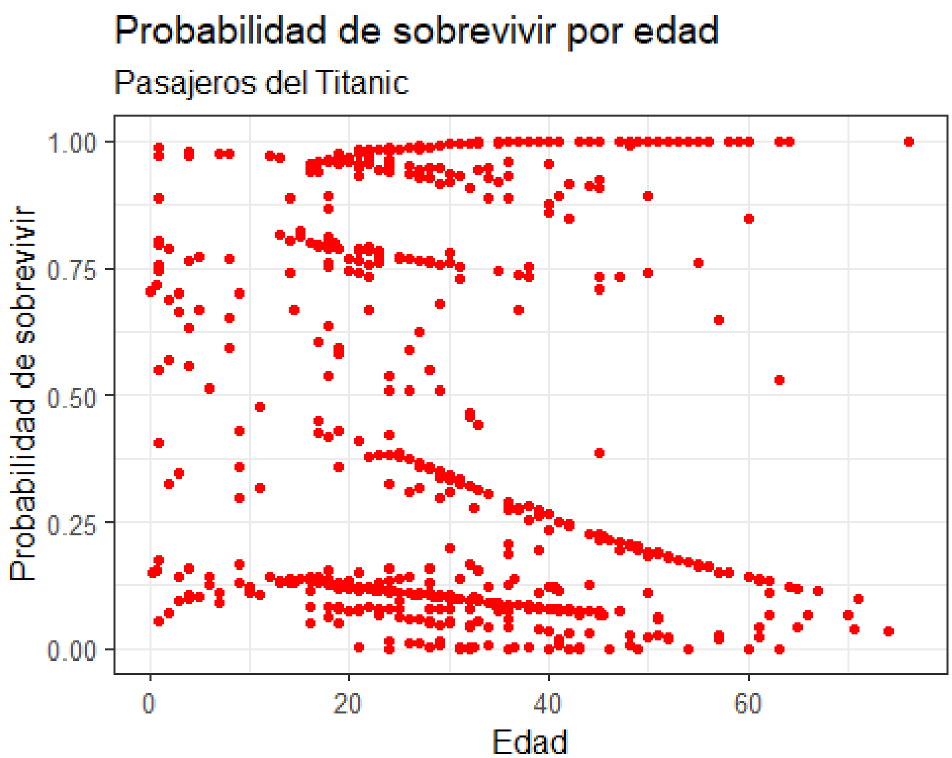
```
coeficientes = B$coefficients
coeficientes

##                (Intercept)                     Pclass2
Pclass3
##               0.7095598816               3.8010432924
1.4152880040
##                    Sexmale                         Age
Fare
##              -0.1702466668               0.1673173781            -
0.0122023566
```

```
##          Pclass2:Sexmale          Pclass3:Sexmale
Pclass2:Age
##            -9.2851051644            -3.1230695370             -
0.2503725887
##              Pclass3:Age               Sexmale:Age
Pclass2:Fare
##            -0.1599854767            -0.2062272750             -
0.0095275848
##             Pclass3:Fare              Sexmale:Fare
Age:Fare
##            -0.0479946564             0.0101198872
0.0002870917
##      Pclass2:Sexmale:Age      Pclass3:Sexmale:Age
Pclass2:Sexmale:Fare
##             0.4326797072             0.1549111740
0.2604975624
##     Pclass3:Sexmale:Fare         Pclass2:Age:Fare
Pclass3:Age:Fare
##             0.0121439532             0.0014844766             -
0.0034266135
##         Sexmale:Age:Fare Pclass2:Sexmale:Age:Fare
Pclass3:Sexmale:Age:Fare
##            -0.0002549481            -0.0155924737
0.0059134558
```

Gráfica del modelo

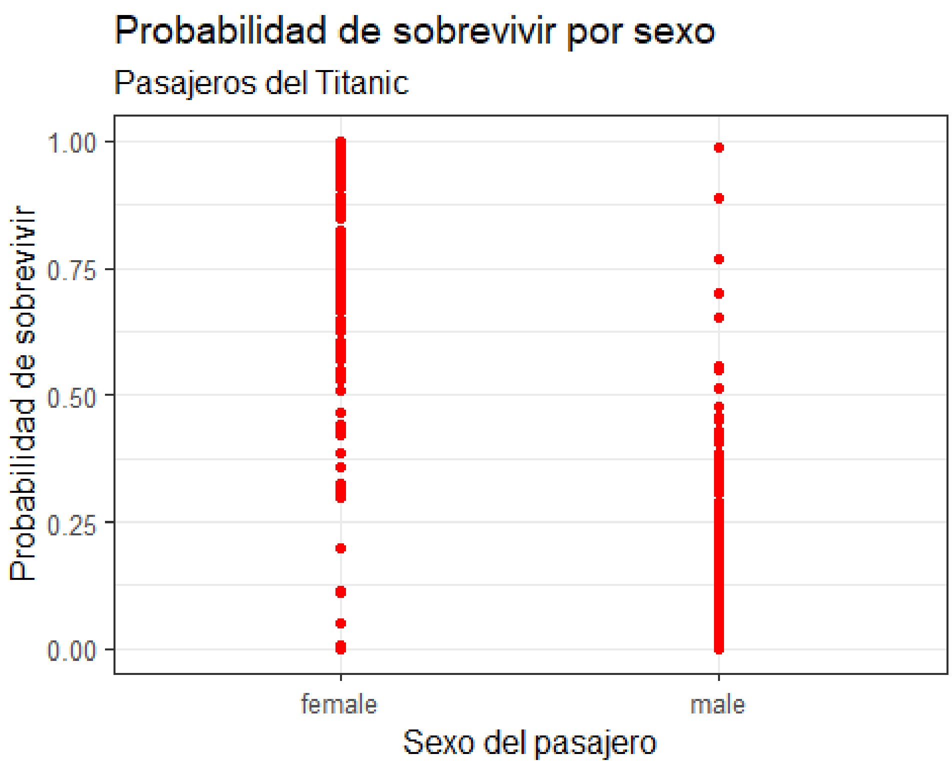
Edad del pasajero

```
p_pred = B$fitted.values
M_pred = data.frame(data_train[,c(1,2,3,4,5)],p_pred)

ggplot(M_pred, aes( x = Age)) +
geom_point(aes(y=p_pred), size=1.5,color="red") +
  labs(x="Edad", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por edad",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```



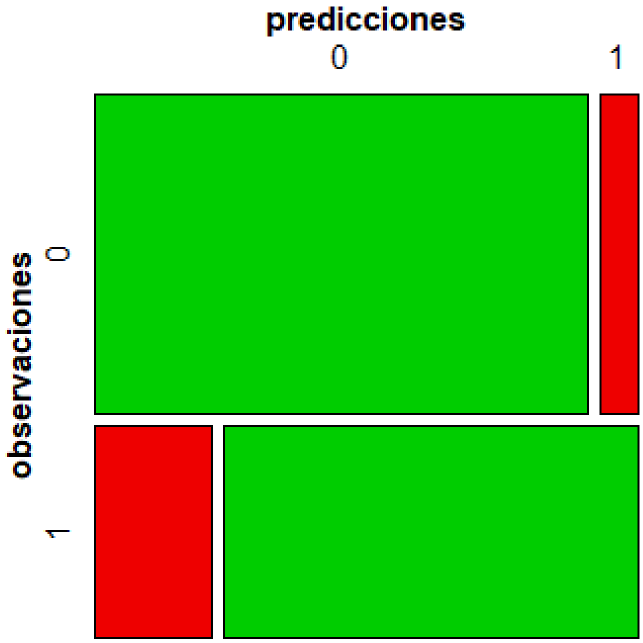
Género del pasajero

```
p_pred = B$fitted.values
M_pred = data.frame(data_train[,c(1,2,3,4,5)],p_pred)

ggplot(M_pred, aes( x = Sex)) +
geom_point(aes(y=p_pred), size=1.5,color="red") +
  labs(x="Sexo del pasajero", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por sexo",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```



Predicciones

```
library(vcd)

## Cargando paquete requerido: grid

##
## Adjuntando el paquete: 'vcd'

## The following object is masked from 'package:ISLR':
##
##     Hitters

predicciones <- ifelse(test = B$fitted.values > 0.5, yes = 1, no = 0)
M_C <- table(B$model$Survived, predicciones, dnn = c("observaciones",
"predicciones"))
M_C

##              predicciones
## observaciones   0   1
##             0 408  32
##             1  64 227

mosaic(M_C, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2,
2)))
```



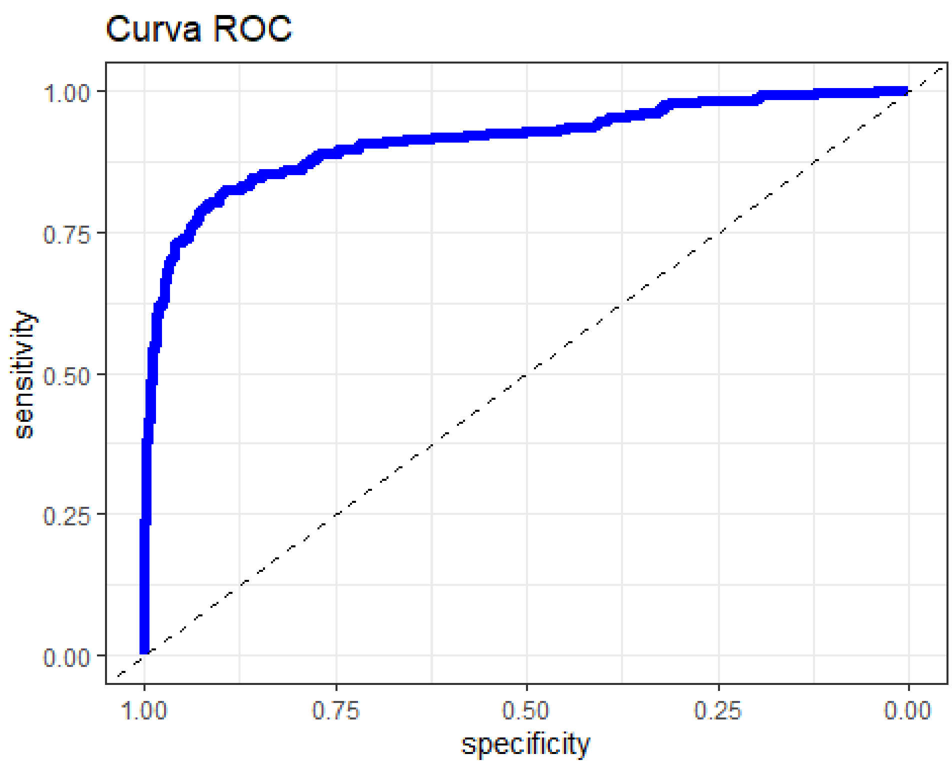

```
Ac = (M_C[1,1]+M_C[2,2])/sum(M_C)
cat("La Exactitud (accuracy) del modelo es", Ac,"\n")

## La Exactitud (accuracy) del modelo es 0.8686731

Se = M_C[1,1]/sum(M_C[1,])
cat("La Sensibilidad del modelo es", Se,"\n")

## La Sensibilidad del modelo es 0.9272727

Sp = M_C[2,2]/sum(M_C[2,])
cat("La Especificidad del modelo es", Sp,"\n")

## La Especificidad del modelo es 0.7800687

P = M_C[1,1]/sum(M_C[,1])
cat("La Precisión del modelo es", P,"\n")

## La Precisión del modelo es 0.8644068
```

La precisión y la sensibilidad del modelo nos dice que el modelo es bueno, y es apto de proveer buenas predicciones.

Curva ROC

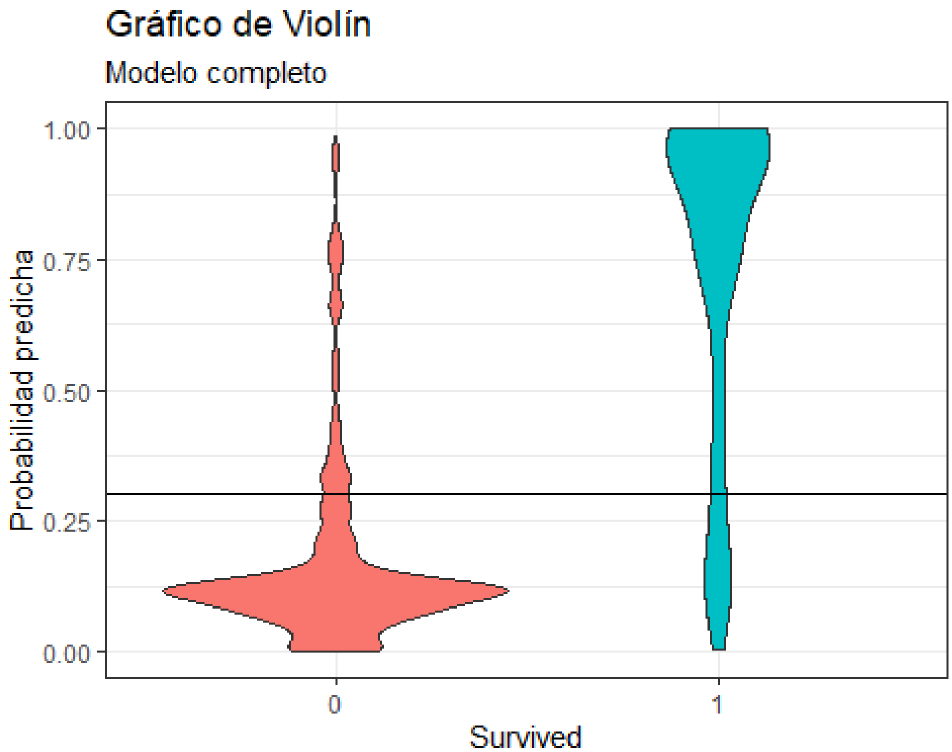
```
pred = predict(B, data = data_train, type = 'response')

library(pROC)

## Warning: package 'pROC' was built under R version 4.4.2
```

```
## Type 'citation("pROC")' for a citation.

##
## Adjuntando el paquete: 'pROC'

## The following object is masked from 'package:Metrics':
##
##     auc

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

ROC <- roc(response=data_train$Survived, predictor=pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

ROC

##
## Call:
## roc.default(response = data_train$Survived, predictor = pred)
##
## Data: pred in 440 controls (data_train$Survived 0) < 291 cases
(data_train$Survived 1).
## Area under the curve: 0.9126

ggroc(ROC, color = "blue", size = 2) + geom_abline(slope = 1, intercept =
1, linetype ='dashed') + labs(title = "Curva ROC") + theme_bw()
```




```r
v_d = data.frame(Survived=data_train$Survived,pred=pred)

ggplot(data=v_d, aes(x=Survived, y=pred, group=Survived,
fill=factor(Survived))) +
  geom_violin() + geom_abline(aes(intercept=0.3,slope=0))+
  theme_bw() +
  guides(fill=FALSE) +
  labs(title='Gráfico de Violín', subtitle='Modelo completo',
y='Probabilidad predicha')
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use
"none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning
was
## generated.
```

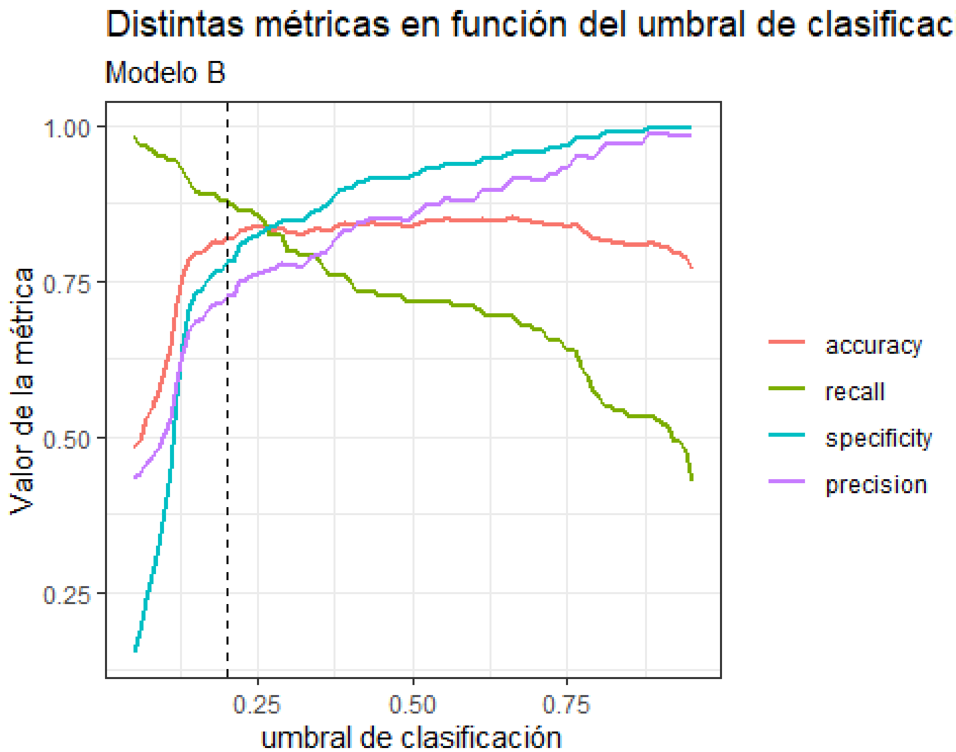


Podemos entender que el modelo hace predicciones de manera correcta. La distribución de 0 y 1 esta acorde a los valores que se deberian de predecir. Dando resultados verdaderos

Validacion del modelo

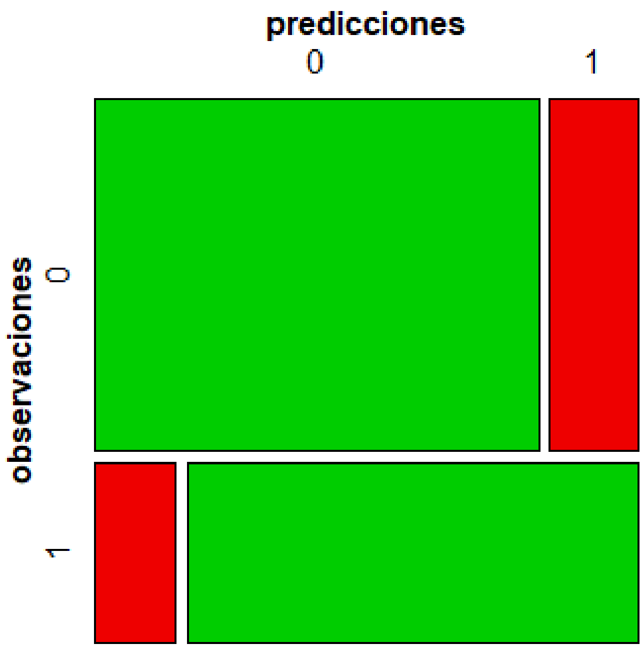
```
pred_val = predict(B, newdata=data_valid, type='response')
clase_real = data_valid$Survived

datosV = data.frame(accuracy=NA, recall=NA, specificity = NA,
precision=NA)

for (i in 5:95){
  clase_predicha = ifelse(pred_val>i/100,1,0)

##Creamos la matriz de confusión
cm= table(clase_predicha,clase_real)

## AccurAcy: Proporción de correctamente predichos
datosV[i,1] = (cm[1,1]+cm[2,2])/(cm[1,1]+cm[1,2]+cm[2,1]+cm[2,2])
## Recall: Tasa de positivos correctamente predichos
datosV[i,2] = (cm[2,2])/(cm[1,2]+cm[2,2])
## Specificity: Tasa de negativos correctamente predichos
datosV[i,3] = cm[1,1]/(cm[1,1]+cm[2,1])
## Precision: Tasa de bien clasificados entre los clasificados como
positivos
datosV[i,4] = cm[2,2]/(cm[2,1]+cm[2,2])
```

```
}

## Se limpia el conjunto de datos
datosV = na.omit(datosV)
datosV$umbral = seq(0.05,0.95,0.01)

library(reshape2)

##
## Adjuntando el paquete: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

datosV_m <- reshape2::melt(datosV,id.vars=c('umbral'))
colnames(datosV_m)[2] <- c('Metrica')

library(ggplot2)

u = 0.20 #Se dio un valor arbitrario, tú modificalo de acuerdo al
criterio que selecciones.

ggplot(data=datosV_m, aes(x=umbral,y=value,color=Metrica)) +
geom_line(size=1) + theme_bw() +
  labs(title= 'Distintas métricas en función del umbral de
clasificación',
       subtitle= 'Modelo B',
       color="", x = 'umbral de clasificación', y = 'Valor de la
métrica') +
  geom_vline(xintercept=u, linetype="dashed", color = "black")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2
3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning
was
## generated.
```



Observando a la gráfica, podemos seleccionar a 0.48 como nuestro umbral, ya que en este punto es dónde todas las métricas se encuentran al mismo nivel. A partir de aqui, el recall disminuye drasticamente, y el accuracy igual disminuye, pero no de la misma manera.

```
prediccionesV = ifelse(pred_val > 0.48, yes = 1, no = 0)
M_Cv <- table(prediccionesV, data_valid$Survived, dnn =
c("observaciones", "predicciones"))
M_Cv

##               predicciones
## observaciones   0   1
##             0 172  34
##             1  16  90

mosaic(M_Cv, shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2,
2)))
```




```r
AcV = (M_Cv[1,1]+M_Cv[2,2])/sum(M_Cv)
cat("La Exactitud (accuracy) del modelo es", AcV,"\n")

## La Exactitud (accuracy) del modelo es 0.8397436

SeV = M_Cv[1,1]/sum(M_Cv[1,])
cat("La Sensibilidad del modelo es", SeV,"\n")

## La Sensibilidad del modelo es 0.8349515

SpV = M_Cv[2,2]/sum(M_Cv[2,])
cat("La Especificidad del modelo es", SpV,"\n")

## La Especificidad del modelo es 0.8490566

PV = M_Cv[1,1]/sum(M_Cv[,1])
cat("La Precisión del modelo es", PV,"\n")

## La Precisión del modelo es 0.9148936
```

Conclusiones

Las carectiristicas que afectaron principalmente al modelo para decidir si una persona sobrevivia o no son el sexo y la edad. Las mujeres dentro del rango de 20-40 años de edad, fueron las personas que tenían más probabilidad de sobrevivir. Esto se explica a la cultura que que exisitia en esa época, dónde se priorizaba a las mujeres y los niños sobre los hombres (quienes son los que tienen menos indice de sobrevivencia). Los valores p de los coeficientes del modelo son grandes, lo que representan una significancia para la prediccion del modelo, es decir, las relaciones de las variables del modelo son significativas para la prediccion. El umbral de decision se decidio que quedara en 0.48 ya que es el punto en el que las metricas convergen de mejor manera, este cambio hizo que el valor de la precisión del modelo aumentara.