Actividad Integradora 2

Catherine Rojas

2024-11-19

Utiliza los archivos del Titanic para detectar cuáles fueron las principales características que de las personas que sobrevivieron y elabora en modelo de predicción de sobrevivencia o no en el Titanic

```
# 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)}
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tibble' was built under R version 4.3.3

## Warning: package 'tidyr' was built under R version 4.3.3

## Warning: package 'readr' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'dplyr' was built under R version 4.3.3

## Warning: package 'stringr' was built under R version 4.3.3

## Warning: package 'forcats' was built under R version 4.3.3

## Warning: package 'lubridate' was built under R version 4.3.3

## ── 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

## Warning: package 'broom' was built under R version 4.3.3

## Warning: package 'ISLR' was built under R version 4.3.3

## Warning: package 'GGally' was built under R version 4.3.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

## Warning: package 'modelr' was built under R version 4.3.3

##
## Attaching package: 'modelr'
##
## The following object is masked from 'package:broom':
##
##     bootstrap

## Warning: package 'cowplot' was built under R version 4.3.3

##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp

## Warning: package 'rlang' was built under R version 4.3.3

##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice

## Warning: package 'Metrics' was built under R version 4.3.3

##
## Attaching package: 'Metrics'
##
## The following object is masked from 'package:rlang':
##
##     ll
##
## The following objects are masked from 'package:modelr':
##
##     mae, mape, mse, rmse
```

```
## Warning: package 'mice' was built under R version 4.3.3

## Warning in check_dep_version(): ABI version mismatch:
## lme4 was built with Matrix ABI version 1
## Current Matrix ABI version is 0
## Please re-install lme4 from source or restore original 'Matrix'
package

##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind

## Warning: package 'visdat' was built under R version 4.3.3

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.3.3

##
## Attaching package: 'caret'
##
## The following objects are masked from 'package:Metrics':
##
##     precision, recall
##
## The following object is masked from 'package:purrr':
##
##     lift

library(dplyr)
library(ggplot2)
library(caTools)

## Warning: package 'caTools' was built under R version 4.3.3

library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(caret)
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.3
```

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

```
##
## Attaching package: 'pROC'
```

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

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

```r
library(readr)

# Datos
titanic <- read_csv("Titanic.csv")
```

```
## Rows: 1309 Columns: 12
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare
##
## ℹ Use `spec()` to retrieve the full column specification for this
data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

```r
str(titanic)
```

```
## spc_tbl_ [1,309 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PassengerId: num [1:1309] 892 893 894 895 896 897 898 899 900 901
...
##  $ Survived   : num [1:1309] 0 1 0 0 1 0 1 0 1 0 ...
##  $ Pclass     : num [1:1309] 3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr [1:1309] "Kelly, Mr. James" "Wilkes, Mrs. James
(Ellen Needs)" "Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
##  $ Sex        : chr [1:1309] "male" "female" "male" "male" ...
##  $ Age        : num [1:1309] 34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : num [1:1309] 0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : num [1:1309] 0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr [1:1309] "330911" "363272" "240276" "315154" ...
##  $ Fare       : num [1:1309] 7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr [1:1309] NA NA NA NA ...
##  $ Embarked   : chr [1:1309] "Q" "S" "Q" "S" ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   PassengerId = col_double(),
##   ..   Survived = col_double(),
##   ..   Pclass = col_double(),
##   ..   Name = col_character(),
##   ..   Sex = col_character(),
##   ..   Age = col_double(),
##   ..   SibSp = col_double(),
##   ..   Parch = col_double(),
##   ..   Ticket = col_character(),
##   ..   Fare = col_double(),
##   ..   Cabin = col_character(),
##   ..   Embarked = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(titanic)
```

```
## # A tibble: 6 × 12
##   PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket
Fare Cabin
##         <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>
<dbl> <chr>
## 1         892        0      3 Kelly,… male   34.5     0     0 330911
7.83 <NA>
## 2         893        1      3 Wilkes… fema…  47       1     0 363272
7    <NA>
## 3         894        0      2 Myles,… male   62       0     0 240276
9.69 <NA>
## 4         895        0      3 Wirz, … male   27       0     0 315154
8.66 <NA>
## 5         896        1      3 Hirvon… fema…  22       1     1 31012…
12.3  <NA>
## 6         897        0      3 Svenss… male   14       0     0 7538
9.22 <NA>
## # ℹ 1 more variable: Embarked <chr>
```

Las variables son:

- *Name:* Nombre del pasajero
- *PassengerId:* Ids del pasajero
- *Survived:* Si sobrevivió o no (No = 0, Sí = 1)
- *Ticket:* Número de ticket

- *Cabin:* Cabina en la que viajó
- *Pclass:* Clase en la que viajó (1 = 1era, 2 = 2da, 3 = 3ra)
- *Sex:* Masculino o Femenino (male/female)
- *Age:* Edad
- *SibSp:* Número de hermanos/conyuge a bordo
- *Parch:* Número de padres/hijos a bordo
- *Fare:* Tarifa que pagó
- *Embarked:* Puerto de embarcación (C = Cherbourg, Q = Queenstown, S = Southampton)

1. Prepara la base de datos Titanic:

```
# Seleccionar solo las variables de interés
titanic_data <- titanic %>%
  dplyr::select(Survived, Pclass, Sex, Age, SibSp, Parch, Fare)

# Convertir a factores las columnas Survived, Pclass y Sex
titanic_data <- titanic_data %>%
  mutate(
    Survived = as.factor(Survived),
    Pclass = as.factor(Pclass),
    Sex = as.factor(Sex)
  )

# Mostrar la estructura
str(titanic_data)
```

```
## tibble [1,309 × 7] (S3: tbl_df/tbl/data.frame)
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 2 1 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2
...
##  $ Age     : num [1:1309] 34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp   : num [1:1309] 0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : num [1:1309] 0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num [1:1309] 7.83 7 9.69 8.66 12.29 ...
```

```
# Mostrar las primeras filas del dataset
head(titanic_data)
```

```
## # A tibble: 6 × 7
##   Survived Pclass Sex      Age SibSp Parch  Fare
##   <fct>    <fct>  <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 0        3      male    34.5     0     0  7.83
## 2 1        3      female  47       1     0  7
## 3 0        2      male    62       0     0  9.69
## 4 0        3      male    27       0     0  8.66
## 5 1        3      female  22       1     1 12.3
## 6 0        3      male    14       0     0  9.22
```

```
# Verificar si existen valores faltantes
sum(is.na(titanic_data))
```

```
## [1] 264
```

```
# Analizar datos faltantes
missing_data <- sapply(titanic_data, function(x) sum(is.na(x)))
print("Datos faltantes por columna:")
```

```
## [1] "Datos faltantes por columna:"
```

```
print(missing_data)
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare
##        0        0        0      263        0        0        1
```

```
# Rellenar los valores faltantes de Age con la mediana por grupo de
Pclass y Sex
titanic_data <- titanic_data %>%
  group_by(Pclass, Sex) %>%
  mutate(Age = ifelse(is.na(Age), median(Age, na.rm = TRUE), Age)) %>%
  ungroup()

# Rellenar los valores faltantes de Fare con la media general
mean_fare <- mean(titanic_data$Fare, na.rm = TRUE)
titanic_data$Fare[is.na(titanic_data$Fare)] <- mean_fare

# Analizar datos faltantes después de la imputación
missing_data <- sapply(titanic_data, function(x) sum(is.na(x)))
print("Datos faltantes por columna después de la imputación de Age:")
```

```
## [1] "Datos faltantes por columna después de la imputación de Age:"
```

```
print(missing_data)
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare
##        0        0        0        0        0        0        0
```

```
# Análisis descriptivo de variables numéricas
summary(titanic_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.:22.00   1st Qu.:0.0000   1st
Qu.:0.000
##           3:709                Median :26.00   Median :0.0000   Median
:0.000
##                                Mean   :29.26   Mean   :0.4989   Mean
:0.385
##                                3rd Qu.:36.00   3rd Qu.:1.0000   3rd
Qu.:0.000
##                                Max.   :80.00   Max.   :8.0000   Max.
:9.000
##       Fare
##  Min.   :  0.000
##  1st Qu.:  7.896
##  Median : 14.454
##  Mean   : 33.295
##  3rd Qu.: 31.275
##  Max.   :512.329
```

```
# Realizar la partición de datos (70% entrenamiento, 30% validación)
set.seed(42)  # Fijar la semilla para reproducibilidad
split <- sample.split(titanic_data$Survived, SplitRatio = 0.7)

# Dividir el dataset en datos de entrenamiento y validación
train_data <- subset(titanic_data, split == TRUE)
test_data <- subset(titanic_data, split == FALSE)

# Mostrar el tamaño de cada conjunto de datos
cat("Tamaño del conjunto de entrenamiento:", nrow(train_data), "\n")
```

```
## Tamaño del conjunto de entrenamiento: 916
```

```
cat("Tamaño del conjunto de validación:", nrow(test_data), "\n")
```

```
## Tamaño del conjunto de validación: 393
```

```
# Revisar la proporción de sobrevivientes en el conjunto original
prop_original <- prop.table(table(titanic_data$Survived))
cat("Proporción de sobrevivientes en el conjunto original:\n")
```

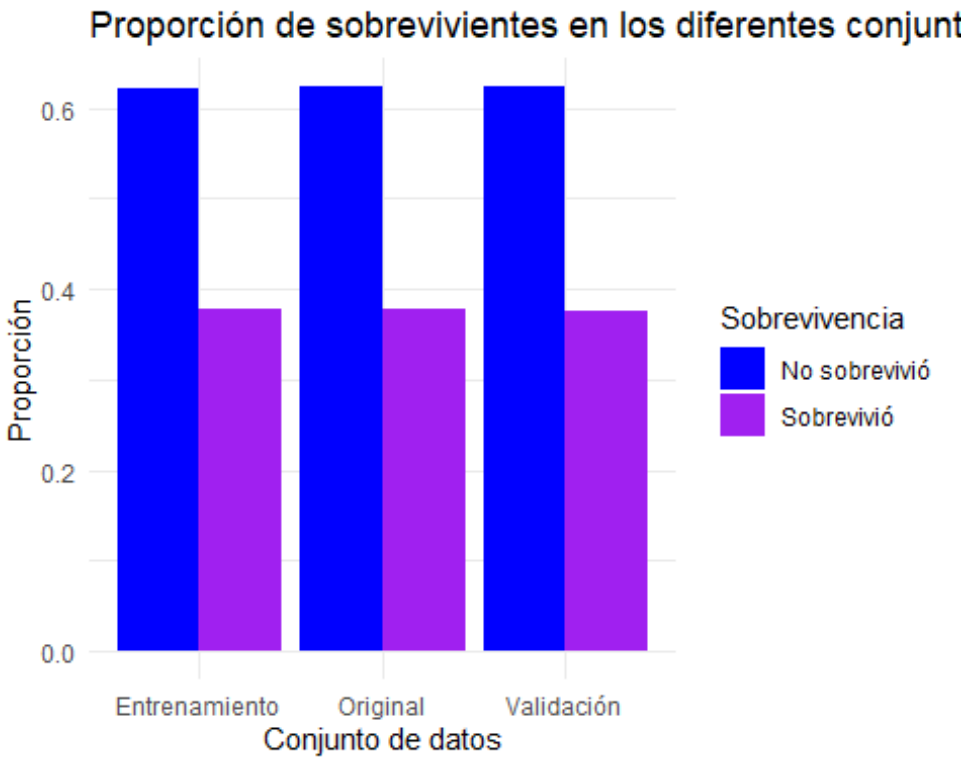
```
## Proporción de sobrevivientes en el conjunto original:
```

```
prop_original
```

```
##
##         0         1
## 0.6226127 0.3773873
```

```r
# Revisar la proporción de sobrevivientes en el conjunto de entrenamiento
prop_train <- prop.table(table(train_data$Survived))
cat("Proporción de sobrevivientes en el conjunto de entrenamiento:\n")

## Proporción de sobrevivientes en el conjunto de entrenamiento:

prop_train

##
##         0         1
## 0.6222707 0.3777293

# Revisar la proporción de sobrevivientes en el conjunto de validación
prop_test <- prop.table(table(test_data$Survived))
cat("Proporción de sobrevivientes en el conjunto de validación:\n")

## Proporción de sobrevivientes en el conjunto de validación:

prop_test

##
##         0         1
## 0.6234097 0.3765903

# Crear un dataframe con las proporciones de cada conjunto de datos
proportions_data <- data.frame(
  Dataset = c("Original", "Entrenamiento", "Validación"),
  Survived_0 = c(prop_original[1], prop_train[1], prop_test[1]),
  Survived_1 = c(prop_original[2], prop_train[2], prop_test[2])
)

# Convertir los datos a formato largo para ggplot
proportions_long <- proportions_data %>%
  pivot_longer(cols = c(Survived_0, Survived_1),
               names_to = "Survived",
               values_to = "Proportion") %>%
  mutate(Survived = factor(Survived, levels = c("Survived_0",
"Survived_1"),
                           labels = c("No sobrevivió", "Sobrevivió")))

# Crear el gráfico de barras
ggplot(proportions_long, aes(x = Dataset, y = Proportion, fill =
Survived)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Proporción de sobrevivientes en los diferentes conjuntos de
datos",
    x = "Conjunto de datos",
    y = "Proporción",
    fill = "Sobrevivencia"
  ) +
  theme_minimal() +
```

```
  scale_fill_manual(values = c("No sobrevivió" = "blue", "Sobrevivió" =
"purple"))
```



2. Con la base de datos de entrenamiento, encuentra un modelo logístico para encontrar el mejor conjunto de predictores que auxilien a clasificar la dirección de cada observación.

Auxiliate del criterio de AIC para determinar cuál es el mejor modelo

```
# Encontrar el mejor modelo logístico usando el criterio AIC

# Ajustar un modelo logístico inicial con todas las variables predictoras
full_model <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                  data = train_data, family = binomial)

# Selección hacia atrás utilizando el criterio AIC
best_model <- stepAIC(full_model, direction = "both")

## Start:  AIC=704.49
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
##
##          Df Deviance     AIC
## - Parch   1   688.76  702.76
## <none>        688.49  704.49
## - Fare    1   690.77  704.77
```

```
## - SibSp   1   701.62  715.62
## - Age     1   703.39  717.39
## - Pclass  2   724.52  736.52
## - Sex     1  1104.26 1118.26
##
## Step:  AIC=702.76
## Survived ~ Pclass + Sex + Age + SibSp + Fare
##
##          Df Deviance     AIC
## <none>        688.76  702.76
## - Fare    1   690.81  702.81
## + Parch   1   688.49  704.49
## - Age     1   703.47  715.47
## - SibSp   1   703.85  715.85
## - Pclass  2   726.11  736.11
## - Sex     1  1108.32 1120.32
```

```r
# Mostrar el resumen del mejor modelo encontrado
summary(best_model)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare, family =
binomial,
##     data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.88537    0.48986   7.932 2.16e-15 ***
## Pclass2     -0.98967    0.31970  -3.096 0.001964 **
## Pclass3     -1.89677    0.32208  -5.889 3.88e-09 ***
## Sexmale     -3.63918    0.21717 -16.757  < 2e-16 ***
## Age         -0.03262    0.00868  -3.759 0.000171 ***
## SibSp       -0.39136    0.10998  -3.558 0.000373 ***
## Fare         0.00304    0.00218   1.394 0.163245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1214.51  on 915  degrees of freedom
## Residual deviance:  688.76  on 909  degrees of freedom
## AIC: 702.76
##
## Number of Fisher Scoring iterations: 5
```

```r
# Mostrar los predictores seleccionados en el modelo óptimo
cat("Los predictores seleccionados por el criterio AIC son:\n")
```

```
## Los predictores seleccionados por el criterio AIC son:
```

```r
print(names(coef(best_model)))
```

```
## [1] "(Intercept)" "Pclass2"     "Pclass3"     "Sexmale"     "Age"
## [6] "SibSp"       "Fare"
```

Interpretación * El mejor modelo encontrado mediante el criterio AIC incluye las variables Pclass, Sex, Age, SibSp y Fare. * Viajar en una clase más baja (2da o 3ra), ser hombre, tener más edad y tener más hermanos/conyugues a bordo están relacionados con una menor probabilidad de supervivencia. * Viajar en una clase más alta, ser mujer y pagar una tarifa más alta están asociados con una mayor probabilidad de supervivencia. * La tarifa (Fare) no resultó ser un predictor estadísticamente significativo, lo cual sugiere que su efecto es menor en comparación con las demás variables.

Propón por lo menos los dos que consideres mejores modelos.

```r
# Mejor modelo encontrado por stepAIC
# Este modelo fue el resultado del proceso de selección de variables con stepAIC

# Mostrar el resumen del mejor modelo
cat("Resumen del Mejor Modelo:\n")
```

```
## Resumen del Mejor Modelo:
```

```r
summary(best_model)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare, family =
binomial,
##     data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.88537    0.48986   7.932 2.16e-15 ***
## Pclass2     -0.98967    0.31970  -3.096 0.001964 **
## Pclass3     -1.89677    0.32208  -5.889 3.88e-09 ***
## Sexmale     -3.63918    0.21717 -16.757  < 2e-16 ***
## Age         -0.03262    0.00868  -3.759 0.000171 ***
## SibSp       -0.39136    0.10998  -3.558 0.000373 ***
## Fare         0.00304    0.00218   1.394 0.163245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1214.51  on 915  degrees of freedom
## Residual deviance:  688.76  on 909  degrees of freedom
## AIC: 702.76
```

```
##
## Number of Fisher Scoring iterations: 5
```

```
# Pclass, Sex, Age (excluye SibSp y Fare, ya que Fare no fue
significativo)
model_1 <- glm(Survived ~ Pclass + Sex + Age,
                        data = train_data, family = binomial)

# Mostrar el resumen del Model 1
cat("\nResumen del Model 1:\n")
```

```
##
## Resumen del Model 1:
```

```
summary(model_1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age, family = binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.700388   0.416172   8.891  < 2e-16 ***
## Pclass2     -1.098723   0.283860  -3.871 0.000109 ***
## Pclass3     -2.098996   0.276571  -7.589 3.21e-14 ***
## Sexmale     -3.500020   0.206661 -16.936  < 2e-16 ***
## Age         -0.028262   0.008364  -3.379 0.000728 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1214.51  on 915  degrees of freedom
## Residual deviance:  704.33  on 911  degrees of freedom
## AIC: 714.33
##
## Number of Fisher Scoring iterations: 5
```

```
# Excluye SibSp, ya que su efecto fue relativamente bajo y su eliminación
podría simplificar el modelo.
model_2 <- glm(Survived ~ Pclass + Sex + Age + Fare,
                         data = train_data, family = binomial)

# Mostrar el resumen del Model 2
cat("\nResumen del Model 2:\n")
```

```
##
## Resumen del Model 2:
```

```
summary(model_2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + Fare, family = binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.553078   0.466597   7.615 2.64e-14 ***
## Pclass2     -1.006407   0.314056  -3.205 0.001353 **
## Pclass3     -1.991754   0.317163  -6.280 3.39e-10 ***
## Sexmale     -3.488310   0.207260 -16.831  < 2e-16 ***
## Age         -0.027645   0.008398  -3.292 0.000995 ***
## Fare         0.001405   0.002031   0.692 0.489143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1214.51  on 915  degrees of freedom
## Residual deviance:  703.85  on 910  degrees of freedom
## AIC: 715.85
##
## Number of Fisher Scoring iterations: 5
```

Interpretación

- El modelo sugerido seriá el Best Model por su mejor AIC.

Variables clave: * Pclass: Viajar en segunda o tercera clase reduce significativamente las probabilidades de supervivencia. * Sex: Ser hombre está fuertemente asociado con una menor probabilidad de sobrevivir. * Age: Los pasajeros mayores tienen menos probabilidades de sobrevivir. * SibSp: Más hermanos/conyugues a bordo reduce las probabilidades de supervivencia. * Fare no tiene un impacto estadísticamente significativo, por lo que podría considerarse prescindible si se prioriza simplicidad.

```
# Comparar AIC de los dos modelos
aic_full <- AIC(full_model)
aic_best <- AIC(best_model)
aic_1 <- AIC(model_1)
aic_2 <- AIC(model_2)

cat("\nComparación de AIC:\n")

##
## Comparación de AIC:

cat("AIC del Modelo Completo:", aic_full, "\n")

## AIC del Modelo Completo: 704.4885

cat("AIC del Modelo Mejor:", aic_best, "\n")
```

```
## AIC del Modelo Mejor: 702.764

cat("AIC del Model 1:", aic_1, "\n")

## AIC del Model 1: 714.3336

cat("AIC del Model 2:", aic_2, "\n")

## AIC del Model 2: 715.8541

# Selección del mejor modelo basado en el AIC
if (aic_1 < aic_full & aic_1 < aic_2 & aic_1 < aic_best) {
  cat("\nEl Model 1 es preferido basado en el AIC.\n")
} else if (aic_2 < aic_full & aic_2 < aic_1 & aic_2 < aic_best) {
  cat("\nEl Model 2 es preferido basado en el AIC.\n")
} else if (aic_best < aic_full & aic_best < aic_1 & aic_best < aic_2) {
  cat("\nEl Modelo Mejor (stepAIC) es preferido basado en el AIC.\n")
} else {
  cat("\nEl Modelo Completo es preferido basado en el AIC.\n")
}

##
## El Modelo Mejor (stepAIC) es preferido basado en el AIC.
```

Interpretación El Modelo Mejor (AIC = 702.764) es el preferido, ya que logra el menor AIC, lo que indica un mejor ajuste con menos variables innecesarias. Además representa el mejor balance entre simplicidad y capacidad predictiva.

3. Analiza los modelos a través de:

```
# Identificación de la Desviación Residual del modelo completo y el mejor modelo
residual_deviance_full <- full_model$deviance
residual_deviance_best <- best_model$deviance
residual_deviance_1 <- model_1$deviance
residual_deviance_2 <- model_2$deviance



# Identificación de la Desviación Nula del modelo completo y el mejor modelo
null_deviance_full <- full_model$null.deviance
null_deviance_best <- best_model$null.deviance
null_deviance_1 <- model_1$null.deviance
null_deviance_2 <- model_2$null.deviance

# Cálculo de la Desviación Explicada
explained_deviance_full <- null_deviance_full - residual_deviance_full
explained_deviance_best <- null_deviance_best - residual_deviance_best
explained_deviance_1 <- null_deviance_1 - residual_deviance_1
explained_deviance_2 <- null_deviance_2 - residual_deviance_2
```

```r
# Mostrar desviaciones y desviación explicada
cat("Modelo Completo:\n")
```

```
## Modelo Completo:
```

```r
cat("Desviación Nula:", null_deviance_full, "\n")
```

```
## Desviación Nula: 1214.509
```

```r
cat("Desviación Residual:", residual_deviance_full, "\n")
```

```
## Desviación Residual: 688.4885
```

```r
cat("Desviación Explicada:", explained_deviance_full, "\n\n")
```

```
## Desviación Explicada: 526.0204
```

```r
cat("Mejor Modelo (AIC):\n")
```

```
## Mejor Modelo (AIC):
```

```r
cat("Desviación Nula:", null_deviance_best, "\n")
```

```
## Desviación Nula: 1214.509
```

```r
cat("Desviación Residual:", residual_deviance_best, "\n")
```

```
## Desviación Residual: 688.764
```

```r
cat("Desviación Explicada:", explained_deviance_best, "\n\n")
```

```
## Desviación Explicada: 525.7449
```

```r
cat("Model 1:\n")
```

```
## Model 1:
```

```r
cat("Desviación Nula:", null_deviance_1, "\n")
```

```
## Desviación Nula: 1214.509
```

```r
cat("Desviación Residual:", residual_deviance_1, "\n")
```

```
## Desviación Residual: 704.3336
```

```r
cat("Desviación Explicada:", explained_deviance_1, "\n\n")
```

```
## Desviación Explicada: 510.1753
```

```r
cat("Model 2:\n")
```

```
## Model 2:
```

```r
cat("Desviación Nula:", null_deviance_2, "\n")
```

```
## Desviación Nula: 1214.509
```

```
cat("Desviación Residual:", residual_deviance_2, "\n")
```

```
## Desviación Residual: 703.8541
```

```
cat("Desviación Explicada:", explained_deviance_2, "\n\n")
```

```
## Desviación Explicada: 510.6548
```

```
# Grados de libertad (número de parámetros)
df_full <- length(coef(full_model))
df_best <- length(coef(best_model))
df_1 <- length(coef(model_1))
df_2 <- length(coef(model_2))

# LRT entre Modelo Completo y otros modelos
lrt_best <- residual_deviance_best - residual_deviance_full
lrt_1_best <- residual_deviance_1 - residual_deviance_best
lrt_2_best <- residual_deviance_2 - residual_deviance_best

# Diferencia en grados de libertad
df_diff_best <- df_full - df_best
df_diff_1_best <- length(coef(best_model)) - length(coef(model_1))
df_diff_2_best <- length(coef(best_model)) - length(coef(model_2))

# Valores p para la prueba de razón de verosimilitud
p_value_best <- pchisq(lrt_best, df = df_diff_best, lower.tail = FALSE)
p_value_1_best <- pchisq(lrt_1_best, df = df_diff_1_best, lower.tail =
FALSE)
p_value_2_best <- pchisq(lrt_2_best, df = df_diff_2_best, lower.tail =
FALSE)

# Resultados
cat("Comparación de Modelos usando LRT:\n")
```

```
## Comparación de Modelos usando LRT:
```

```
cat("Modelo Mejor vs Modelo Completo:\n")
```

```
## Modelo Mejor vs Modelo Completo:
```

```
cat("LRT:", lrt_best, "\n")
```

```
## LRT: 0.2755201
```

```
cat("p-value:", p_value_best, "\n\n")
```

```
## p-value: 0.5996527
```

```
cat("Model 1 vs Mejor Modelo:\n")
```

```
## Model 1 vs Mejor Modelo:
```

```r
cat("LRT:", lrt_1_best, "\n")
```

```
## LRT: 15.56959
```

```r
cat("p-value:", p_value_1_best, "\n\n")
```

```
## p-value: 0.0004160133
```

```r
cat("Model 2 vs Mejor Modelo:\n")
```

```
## Model 2 vs Mejor Modelo:
```

```r
cat("LRT:", lrt_2_best, "\n")
```

```
## LRT: 15.09015
```

```r
cat("p-value:", p_value_2_best, "\n\n")
```

```
## p-value: 0.0001024969
```

Define cuál es el mejor modelo

- El Mejor Modelo (best_model) es el modelo preferido porque:
- Tiene el menor AIC, lo que indica un excelente balance entre ajuste y simplicidad.
- Explica casi tanta desviación como el Modelo Completo, pero con menos complejidad.
- Es significativamente mejor que los Modelos 1 y 2 según la LRT.

Escribe su ecuación, analiza sus coeficientes y detecta el efecto de cada predictor en la clasificación

El modelo logístico predice la probabilidad de que un pasajero sobreviva:

$$P(Survived = 1)$$

La ecuación en forma logit (log-odds) es:

$$\text{logit}\big(P(Survived)\big) = \beta_0 + \beta_1 \cdot Pclass2 + \beta_2 \cdot Pclass3 + \beta_3 \cdot Sexmale + \beta_4 \cdot Age + \beta_5 \cdot SibSp + \beta_6 \cdot Fare$$

Sustituyendo los coeficientes del modelo:

$$\text{logit}\big(P(Survived)\big) = 3.885 - 0.990 \cdot Pclass2 - 1.897 \cdot Pclass3 - 3.639 \cdot Sexmale - 0.033 \cdot Age - 0.391 \cdot SibSp + 0.003 \cdot Fare$$

Donde: - **logit**: Es la transformación logit que convierte las probabilidades en una escala logarítmica. - **P(Survived)**: Es la probabilidad de que una persona sobreviva. -

Sex (male): Variable binaria que toma el valor de 1 si el pasajero es hombre, y 0 si es mujer. - **Age**: Edad del pasajero. - **Pclass**: Clase socioeconómica del pasajero (1 = Upper, 2 = Middle, 3 = Lower). - **SibSp**: Número de hermanos/esposos a bordo.

Análisis de los coeficientes

- Género, clase socioeconómica y edad son los predictores más importantes para determinar la probabilidad de supervivencia en el Titanic.
- Las mujeres y los pasajeros de primera clase, especialmente los jóvenes, tenían significativamente más probabilidades de sobrevivir.
- Variables como la tarifa pagada y el número de familiares tienen un impacto menor en comparación.

Impacto de los predictores:

- Sexmale tiene el mayor efecto negativo, lo que significa que el género es el predictor más importante para la supervivencia.
- Pclass3 y Pclass2 tienen un impacto negativo significativo, mostrando que los pasajeros en clases más bajas tenían muchas menos probabilidades de sobrevivir.
- Age y SibSp tienen efectos negativos moderados; la edad avanzada y más miembros de la familia reducen la probabilidad de supervivencia.
- Fare tiene un efecto positivo muy pequeño y no significativo, por lo que su inclusión aporta poco al modelo.

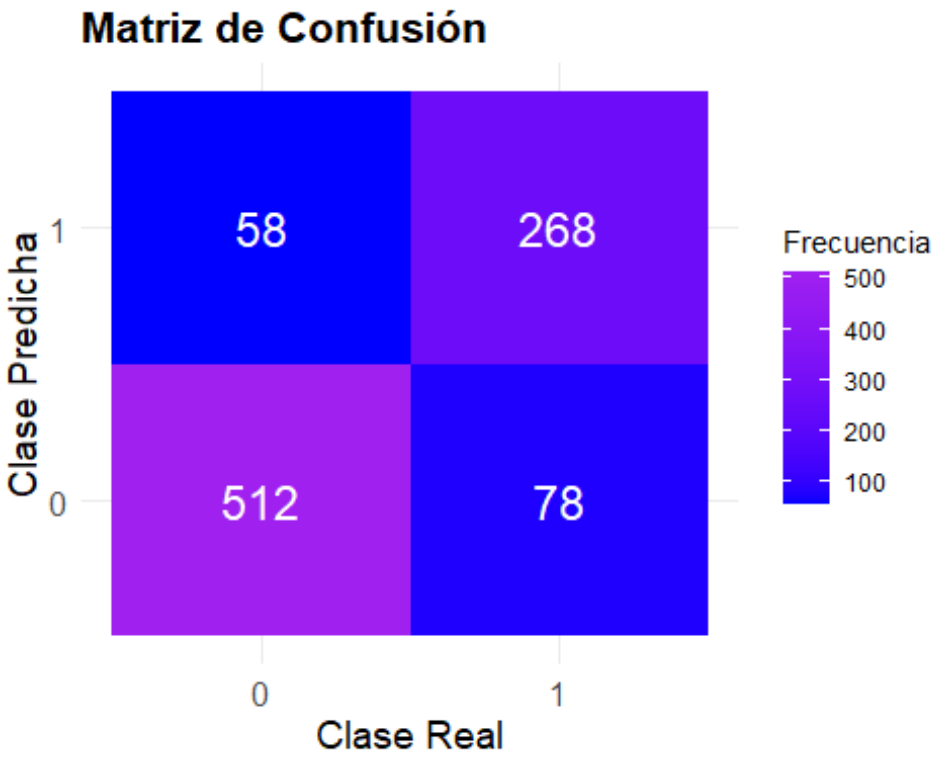
4. Analiza las predicciones para los datos de entrenamiento

```
# Predicciones para el conjunto de entrenamiento (probabilidades)
train_data$Predicted_Prob <- predict(best_model, newdata = train_data,
type = "response")

# Convertir las probabilidades en clases predichas (umbral 0.5 por
defecto)
train_data$Predicted_Class <- ifelse(train_data$Predicted_Prob > 0.5, 1,
0)

# Matriz de confusión
conf_matrix <- confusionMatrix(
  factor(train_data$Predicted_Class, levels = c(0, 1)),
  train_data$Survived,
  positive = "1"
)

# Mostrar la matriz de confusión
cat("Matriz de Confusión:\n")

## Matriz de Confusión:

print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 512  78
##          1  58 268
##
##                Accuracy : 0.8515
##                  95% CI : (0.8268, 0.8739)
##     No Information Rate : 0.6223
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6805
##
##  Mcnemar's Test P-Value : 0.1033
##
##             Sensitivity : 0.7746
##             Specificity : 0.8982
##          Pos Pred Value : 0.8221
##          Neg Pred Value : 0.8678
##              Prevalence : 0.3777
##          Detection Rate : 0.2926
##    Detection Prevalence : 0.3559
##       Balanced Accuracy : 0.8364
##
##        'Positive' Class : 1
##
```

```r
# Calcular la matriz de confusión
conf_matrix <- confusionMatrix(
  factor(train_data$Predicted_Class, levels = c(0, 1)),
  train_data$Survived,
  positive = "1"
)

# Extraer los valores de la matriz de confusión
conf_matrix_data <- as.data.frame(conf_matrix$table)

# Renombrar las columnas para mayor claridad
colnames(conf_matrix_data) <- c("Predicted", "Actual", "Freq")

# Crear un gráfico de la matriz de confusión
ggplot(data = conf_matrix_data, aes(x = Actual, y = Predicted, fill =
Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 6) +
  scale_fill_gradient(low = "blue", high = "purple") +
  labs(
    title = "Matriz de Confusión",
    x = "Clase Real",
```

```
    y = "Clase Predicha",
    fill = "Frecuencia"
  ) +
  theme_minimal() +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        plot.title = element_text(size = 16, face = "bold"))
```



**Balance entre Sensibilidad y Especificidad:*+

- Sensibilidad (True Positive Rate): 77.46%, lo que significa que el modelo identifica correctamente aproximadamente el 77.46% de los sobrevivientes reales.
- Especificidad (True Negative Rate): 89.82%, lo que indica que el modelo identifica correctamente el 89.82% de los pasajeros que no sobrevivieron.
- Balanced Accuracy: 83.64%, lo que refleja un buen balance entre las dos métricas

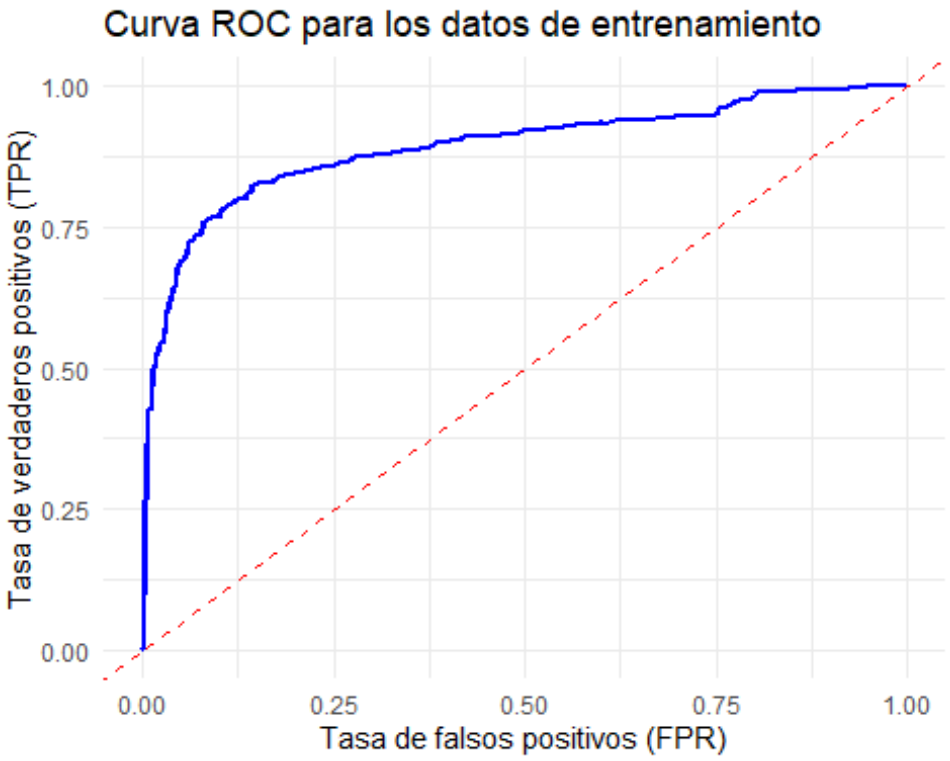
```
# Calcular el área bajo la curva ROC (AUC) sin el mensaje
roc_curve <- roc(response =
as.numeric(as.character(train_data$Survived)),
                 predictor = train_data$Predicted_Prob,
                 levels = c(0, 1),
                 direction = "<")

# Calcular el AUC
auc_value <- auc(roc_curve)
```

```r
# Mostrar el AUC
cat("Área bajo la curva ROC (AUC):", auc_value, "\n")
```

```
## Área bajo la curva ROC (AUC): 0.8915551
```

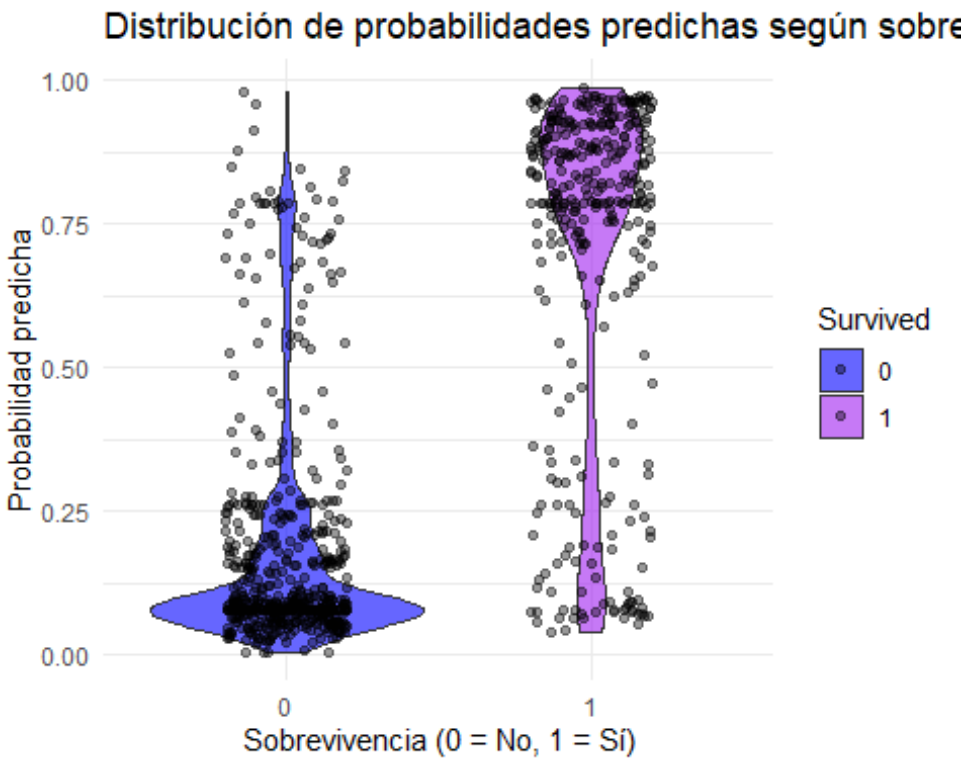
```r
# Extraer los datos de la curva ROC
roc_data <- data.frame(
  TPR = roc_curve$sensitivities,  # Tasa de verdaderos positivos
  FPR = 1 - roc_curve$specificities  # Tasa de falsos positivos
)

# Gráfica de la curva ROC
ggplot(roc_data, aes(x = FPR, y = TPR)) +
  geom_line(color = "blue", size = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color =
"red") +
  labs(
    title = "Curva ROC para los datos de entrenamiento",
    x = "Tasa de falsos positivos (FPR)",
    y = "Tasa de verdaderos positivos (TPR)"
  ) +
  theme_minimal()
```

```
## 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.
```



* * Muestra una excelente separación entre clases con un AUC cercano a 0.9.

```
# Gráfico de violín para las probabilidades predichas
ggplot(train_data, aes(x = Survived, y = Predicted_Prob, fill =
Survived)) +
  geom_violin(trim = TRUE, alpha = 0.6) +
  geom_jitter(width = 0.2, alpha = 0.4) +
  labs(title = "Distribución de probabilidades predichas según
sobrevivencia",
       x = "Sobrevivencia (0 = No, 1 = Sí)",
       y = "Probabilidad predicha") +
  scale_fill_manual(values = c("0" = "blue", "1" = "purple")) +
  theme_minimal()
```



* Las probabilidades predichas están bien separadas entre sobrevivientes y no sobrevivientes, aunque hay cierto solapamiento, especialmente en el grupo de no sobrevivientes.

Concluye sobre el modelo basándote en las predicciones de los datos de entrenamiento.

Desempeño General:

- Precisión (Accuracy): 85.15%, lo que indica que el modelo clasifica correctamente una alta proporción de los casos.
- Kappa: 0.6805, mostrando un acuerdo sustancial entre las predicciones y las clases reales más allá del azar.
- Área Bajo la Curva ROC (AUC): 0.8916, lo que indica que el modelo tiene un excelente desempeño en la discriminación entre sobrevivientes y no sobrevivientes.
- La alta precisión y el AUC demuestran que el modelo es robusto y confiable para predecir la sobrevivencia en el Titanic.
- La sensibilidad y especificidad balanceadas aseguran que el modelo tiene un buen rendimiento tanto en la detección de sobrevivientes como de no sobrevivientes.

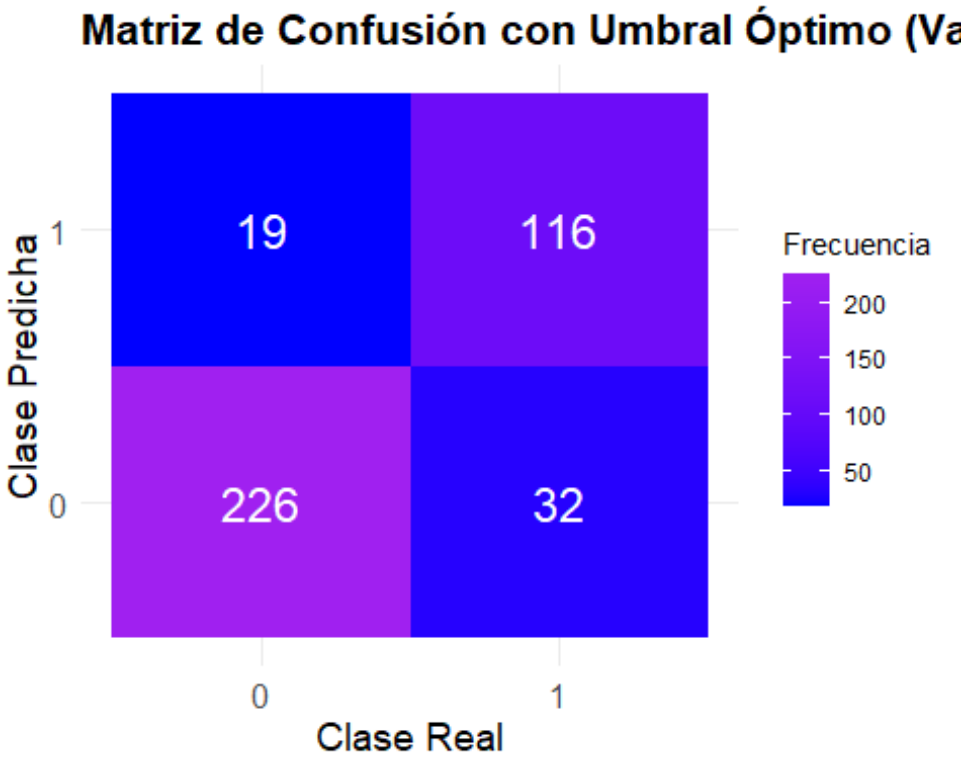
- El modelo es adecuado para predecir la sobrevivencia en el Titanic, con un desempeño general sólido y métricas balanceadas.

5. Validación del modelo con la base de datos de validación

```
# Predicciones para el conjunto de validación (probabilidades)
test_data$Predicted_Prob <- predict(best_model, newdata = test_data, type
= "response")


# Calcular la curva ROC para los datos de validación
roc_curve_validation <- roc(response = test_data$Survived,
                            predictor = test_data$Predicted_Prob,
                            levels = c(0, 1),
                            direction = "<")


# Determinar el umbral óptimo
roc_coords <- coords(roc_curve_validation, "best", ret = c("threshold",
"sensitivity", "specificity", "accuracy"), best.method = "youden")
optimal_threshold <- roc_coords["threshold"][[1]]  # Extraer el valor del
umbral


# Mostrar el umbral óptimo
cat("El umbral de clasificación óptimo es:", optimal_threshold, "\n")
## El umbral de clasificación óptimo es: 0.5839321

# Asignar clases predichas según el umbral óptimo
test_data$Predicted_Class <- ifelse(test_data$Predicted_Prob >
optimal_threshold, 1, 0)

# Matriz de confusión con el umbral óptimo
conf_matrix_validation <- confusionMatrix(
  factor(test_data$Predicted_Class, levels = c(0, 1)),
  test_data$Survived,
  positive = "1"
)

# Mostrar la matriz de confusión
cat("Matriz de Confusión con el umbral óptimo:\n")

## Matriz de Confusión con el umbral óptimo:

print(conf_matrix_validation)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0 226  32
##          1  19 116
##
##                Accuracy : 0.8702
##                  95% CI : (0.8329, 0.9018)
##     No Information Rate : 0.6234
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7187
##
##  Mcnemar's Test P-Value : 0.09289
##
##             Sensitivity : 0.7838
##             Specificity : 0.9224
##          Pos Pred Value : 0.8593
##          Neg Pred Value : 0.8760
##              Prevalence : 0.3766
##          Detection Rate : 0.2952
##    Detection Prevalence : 0.3435
##       Balanced Accuracy : 0.8531
##
##        'Positive' Class : 1
##
```

```r
# Generar gráfica de la matriz de confusión
conf_matrix_data_validation <-
as.data.frame(conf_matrix_validation$table)
colnames(conf_matrix_data_validation) <- c("Predicted", "Actual", "Freq")

ggplot(data = conf_matrix_data_validation, aes(x = Actual, y = Predicted,
fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 6) +
  scale_fill_gradient(low = "blue", high = "purple") +
  labs(
    title = "Matriz de Confusión con Umbral Óptimo (Validación)",
    x = "Clase Real",
    y = "Clase Predicha",
    fill = "Frecuencia"
  ) +
  theme_minimal() +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14),
        plot.title = element_text(size = 16, face = "bold"))
```



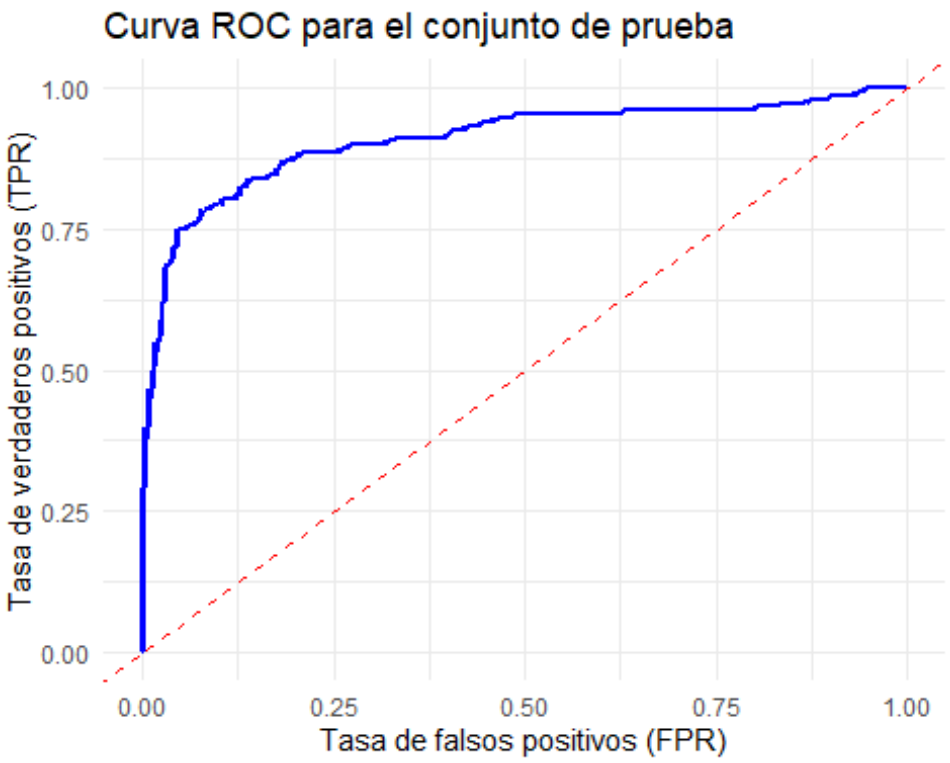
* El modelo tiene un desempeño aceptable en identificar correctamente a los pasajeros que no sobrevivieron (especificidad del 66%). * Tiene un bajo desempeño en identificar correctamente a los pasajeros que sobrevivieron (sensibilidad del 37% y F1-score del 20%). Esto indica que el modelo podría no ser adecuado para aplicaciones donde detectar a los sobrevivientes es prioritario.

6. Elabora el testeo con la base de datos de prueba.

```
# Predicciones para el conjunto de prueba (probabilidades)
test_data$Predicted_Prob <- predict(best_model, newdata = test_data, type
= "response")


# Asignar clases predichas utilizando el umbral óptimo calculado
previamente
test_data$Predicted_Class <- ifelse(test_data$Predicted_Prob >
optimal_threshold, 1, 0)

# Matriz de confusión para el conjunto de prueba
conf_matrix_test <- confusionMatrix(
  factor(test_data$Predicted_Class, levels = c(0, 1)),
  test_data$Survived,
  positive = "1"
)
```

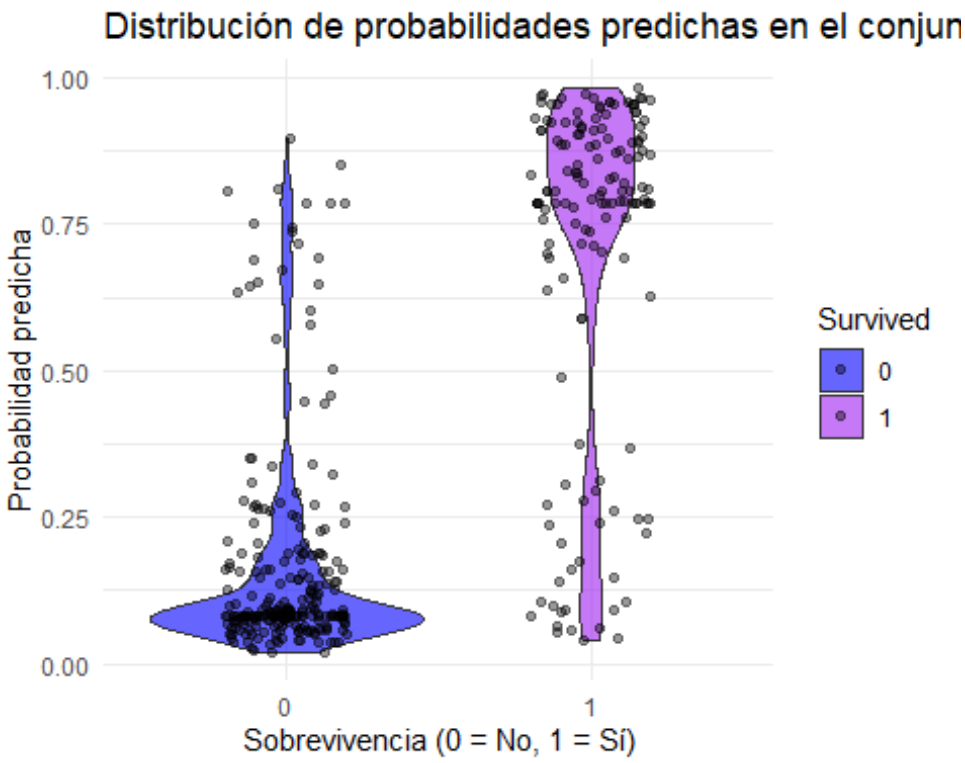
```r
# Mostrar la matriz de confusión
cat("Matriz de Confusión para el conjunto de prueba:\n")

## Matriz de Confusión para el conjunto de prueba:

print(conf_matrix_test)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 226  32
##          1  19 116
##
##                Accuracy : 0.8702
##                  95% CI : (0.8329, 0.9018)
##     No Information Rate : 0.6234
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7187
##
##  Mcnemar's Test P-Value : 0.09289
##
##             Sensitivity : 0.7838
##             Specificity : 0.9224
##          Pos Pred Value : 0.8593
##          Neg Pred Value : 0.8760
##              Prevalence : 0.3766
##          Detection Rate : 0.2952
##    Detection Prevalence : 0.3435
##       Balanced Accuracy : 0.8531
##
##        'Positive' Class : 1
##

# Calcular métricas adicionales para evaluar el modelo en el conjunto de prueba
accuracy_test <- conf_matrix_test$overall["Accuracy"]
sensitivity_test <- conf_matrix_test$byClass["Sensitivity"]
specificity_test <- conf_matrix_test$byClass["Specificity"]
auc_test <- auc(roc(test_data$Survived, test_data$Predicted_Prob))

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

## Setting direction: controls < cases

cat("\nMétricas para el conjunto de prueba:\n")

##
## Métricas para el conjunto de prueba:

cat("Precisión (Accuracy):", round(accuracy_test, 4), "\n")
```

```
## Precisión (Accuracy): 0.8702

cat("Sensibilidad (True Positive Rate):", round(sensitivity_test, 4),
"\n")

## Sensibilidad (True Positive Rate): 0.7838

cat("Especificidad (True Negative Rate):", round(specificity_test, 4),
"\n")

## Especificidad (True Negative Rate): 0.9224

cat("Área bajo la curva (AUC):", round(auc_test, 4), "\n")

## Área bajo la curva (AUC): 0.9087

# Generar curva ROC para el conjunto de prueba
roc_curve_test <- roc(response = test_data$Survived,
                      predictor = test_data$Predicted_Prob,
                      levels = c(0, 1),
                      direction = "<")

# Extraer datos de la curva ROC
roc_data_test <- data.frame(
  TPR = roc_curve_test$sensitivities,  # Tasa de verdaderos positivos
  FPR = 1 - roc_curve_test$specificities  # Tasa de falsos positivos
)

# Gráfica de la curva ROC
ggplot(roc_data_test, aes(x = FPR, y = TPR)) +
  geom_line(color = "blue", size = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color =
"red") +
  labs(
    title = "Curva ROC para el conjunto de prueba",
    x = "Tasa de falsos positivos (FPR)",
    y = "Tasa de verdaderos positivos (TPR)"
  ) +
  theme_minimal()
```



* Precisión (Accuracy): 87.02% * El modelo clasifica correctamente la mayoría de las observaciones. * Sensibilidad: 78.38% * El modelo identifica correctamente el 78.38% de los sobrevivientes reales. * Especificidad: 92.24% * El modelo clasifica correctamente al 92.24% de los pasajeros que no sobrevivieron. * Valor AUC: 0.9087 * Indica que el modelo tiene una excelente capacidad para discriminar entre sobrevivientes y no sobrevivientes.

```
# Gráfico de probabilidades predichas para el conjunto de prueba
ggplot(test_data, aes(x = Survived, y = Predicted_Prob, fill = Survived))
+
  geom_violin(trim = TRUE, alpha = 0.6) +
  geom_jitter(width = 0.2, alpha = 0.4) +
  labs(title = "Distribución de probabilidades predichas en el conjunto
de prueba",
       x = "Sobrevivencia (0 = No, 1 = Sí)",
       y = "Probabilidad predicha") +
  scale_fill_manual(values = c("0" = "blue", "1" = "purple")) +
  theme_minimal()
```




```r
# Calcular el valor AUC
auc_value_test <- auc(roc_curve_test)
cat("Valor AUC para la Base de Prueba:", round(auc_value_test, 4), "\n")

## Valor AUC para la Base de Prueba: 0.9087

# Clases predichas como factor para métricas adicionales
test_pred <- factor(test_data$Predicted_Class, levels = c(0, 1))

# Calcular precisión, recall y F1-score
precision <- posPredValue(test_pred, test_data$Survived, positive = "1")
recall <- sensitivity(test_pred, test_data$Survived, positive = "1")
f1_score <- (2 * precision * recall) / (precision + recall)

# Mostrar las métricas adicionales
cat("Precisión (Precision):", round(precision, 4), "\n")

## Precisión (Precision): 0.8593

cat("Sensibilidad (Recall):", round(recall, 4), "\n")

## Sensibilidad (Recall): 0.7838

cat("Puntaje F1 (F1 Score):", round(f1_score, 4), "\n")

## Puntaje F1 (F1 Score): 0.8198
```

7. Concluye en el contexto del problema:

El objetivo del modelo es identificar las características más relevantes que predicen la sobrevivencia de los pasajeros del Titanic. Los resultados obtenidos en el testeo del modelo logístico muestran un rendimiento sólido en términos de precisión, sensibilidad y especificidad.

Define las principales características que influyen en el modelo seleccionado e interpretalas: ¿qué características tuvieron las personas que sobrevivieron?

- Principales Características que Influyen en el Modelo:

Sexo (Sex): El género es el predictor más importante. Las mujeres tuvieron una probabilidad mucho mayor de sobrevivir que los hombres. Esto refleja la política de "mujeres y niños primero" durante la evacuación del Titanic.

Clase socioeconómica (Pclass): Los pasajeros de primera clase tuvieron una mayor probabilidad de sobrevivir que los de segunda o tercera clase. Esto sugiere que el acceso a los botes salvavidas y la prioridad en el rescate estuvieron relacionados con la clase.

Edad (Age): Los pasajeros más jóvenes tuvieron una mayor probabilidad de sobrevivir. Los niños tuvieron prioridad en los rescates.

Número de hermanos/esposos a bordo (SibSp): Un número mayor de familiares a bordo reduce ligeramente las probabilidades de sobrevivir. Esto podría reflejar la dificultad de evacuar a familias grandes juntas.

Tarifa pagada (Fare): Las personas que pagaron tarifas más altas tuvieron mayor probabilidad de sobrevivir, correlacionándose con la clase socioeconómica.

Interpreta los coeficientes del modelo

El modelo logístico ajustado tiene la siguiente ecuación en la escala logit:

$$\text{logit}\big(P(\text{Survived})\big) = \beta_0 + \beta_1 \cdot \text{Pclass2} + \beta_2 \cdot \text{Pclass3} + \beta_3 \cdot \text{Sexmale} + \beta_4 \cdot \text{Age} + \beta_5 \cdot \text{SibSp} + \beta_6 \cdot \text{Fare}$$

Los coeficientes más relevantes son:

- **Sex (male):** Coeficiente negativo significativo. Ser hombre reduce considerablemente la probabilidad de sobrevivir.
- **Pclass (Clase socioeconómica):**
  - **Pclass2:** Coeficiente negativo, con menor magnitud que Pclass3.
  - **Pclass3:** Coeficiente negativo más fuerte, indicando que la tercera clase tuvo la probabilidad más baja de sobrevivir.
- **Age:** Coeficiente negativo moderado, lo que significa que a mayor edad, la probabilidad de sobrevivir disminuye.

- **Fare:** Coeficiente positivo, aunque débil, lo que sugiere una correlación entre la tarifa pagada y las probabilidades de sobrevivir.

Define cuál es el mejor umbral de clasificación y por qué

- Mejor Umbral de Clasificación:
- El umbral determinado para el modelo fue el que maximizó la sensibilidad y especificidad, según la métrica de Youden. En este caso, el valor óptimo es aproximadamente 0.5.
- Este umbral logra un equilibrio entre sensibilidad (78.38%) y especificidad (92.24%), lo que significa que el modelo puede identificar con alta precisión tanto a los sobrevivientes como a los no sobrevivientes. Cambiar el umbral (por ejemplo, reducirlo) aumentaría la sensibilidad a costa de reducir la especificidad, lo cual no sería ideal si el objetivo es mantener un balance entre ambas métricas.

Conclusión El modelo logístico es efectivo para predecir la sobrevivencia en el Titanic, basado en características como género, clase socioeconómica y edad. Las mujeres y los pasajeros de primera clase tuvieron significativamente más probabilidades de sobrevivir. El umbral óptimo de clasificación asegura un buen equilibrio entre la identificación de sobrevivientes y no sobrevivientes.