act_int_2_A01742161

Rogelio Lizárraga

2024-11-19

Prepara la base de datos Titanic:

Analiza los datos faltantes

```
# Cargar librerías necesarias
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret) # Para partición de datos
```

```
## Loading required package: lattice
```

```
# Leer las bases de datos
titanic <- read.csv("Titanic.csv")
titanic_test <- read.csv("Titanic_test.csv")

# Mostrar un resumen de la base
str(titanic)
```

```
## '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
##  $ 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" ...
```

```r
summary(titanic)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3774   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   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
##     Ticket               Fare            Cabin             Embarked
##  Length:1309        Min.   :  0.000   Length:1309        Length:1309
##  Class :character   1st Qu.:  7.896   Class :character   Class :character
##  Mode  :character   Median : 14.454   Mode  :character   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
```

Como podemos observar, PassengerId, Name, Ticket, Cabin y Embarked tienen una influencia baja, y aunque Name o Cabin pudiesen proporcionar información útil. Se necesitaría un preprocesamiento más adecuado.

Para esto, trabajaremos con las variables de Sex, Age, Pclass, SibSp, Parch y Fare, pues son relevantes para el análisis del problema.

```r
# Revisar valores faltantes
sapply(titanic, function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           2
```

```r
# Visualizar datos faltantes
library(VIM) # Visualización de valores faltantes
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

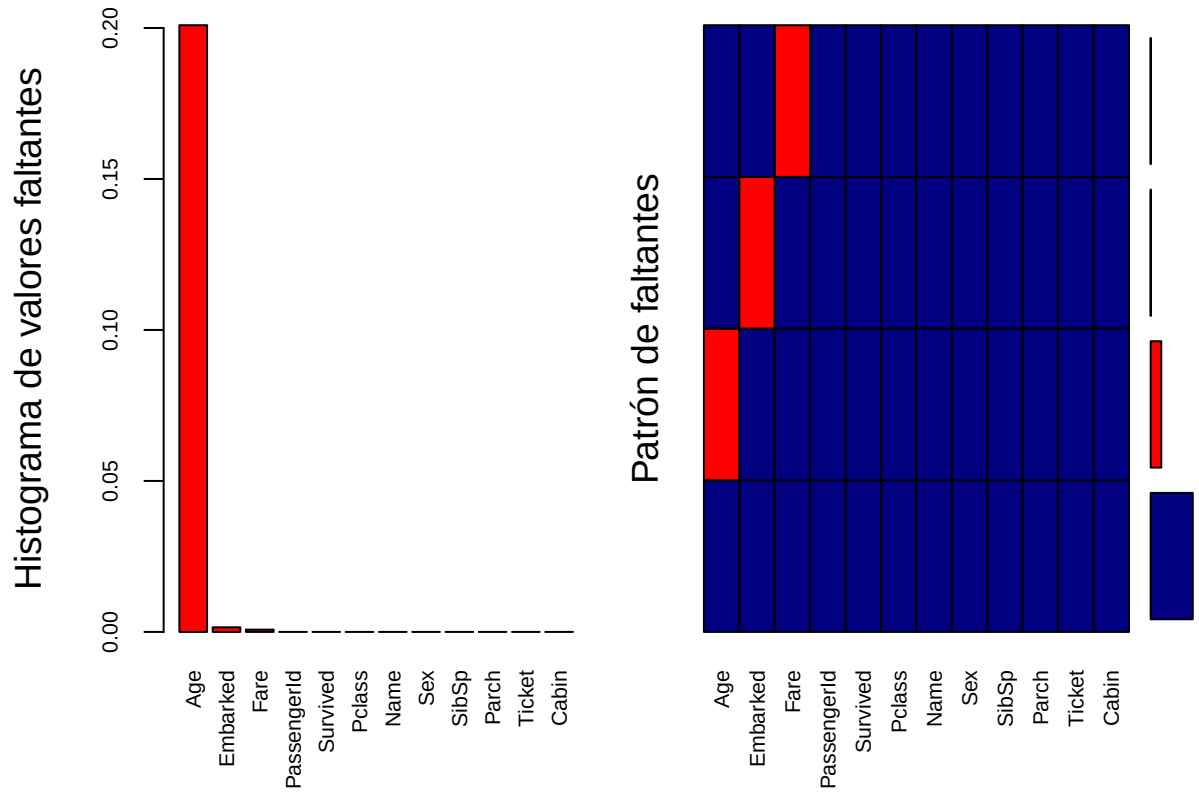
```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
aggr(titanic, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
     labels=names(titanic), cex.axis=0.7, gap=3, ylab=c("Histograma de valores faltantes","Patrón de fal
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Variables sorted by number of missings:
##     Variable        Count
##          Age 0.2009167303
```

```
##     Embarked 0.0015278839
##         Fare 0.0007639419
##  PassengerId 0.0000000000
##     Survived 0.0000000000
##       Pclass 0.0000000000
##         Name 0.0000000000
##          Sex 0.0000000000
##        SibSp 0.0000000000
##        Parch 0.0000000000
##       Ticket 0.0000000000
##        Cabin 0.0000000000
```

Como podemos observar, la proporción de valores faltantes es alta, por lo que afectará de manera significativa los resultados de nuestro modelo. Se eliminarán los valores nulos para esta situación.

```
# Eliminar filas con valores faltantes
titanic_clean <- na.omit(titanic)

# Verificar si quedan valores faltantes
sapply(titanic_clean, function(x) sum(is.na(x)))
```

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

```
# Comparar tamaño antes y después de limpiar
cat("Tamaño original:", nrow(titanic), "filas\n")
```

```
## Tamaño original: 1309 filas
```

```
cat("Tamaño después de limpiar:", nrow(titanic_clean), "filas\n")
```

```
## Tamaño después de limpiar: 1043 filas
```

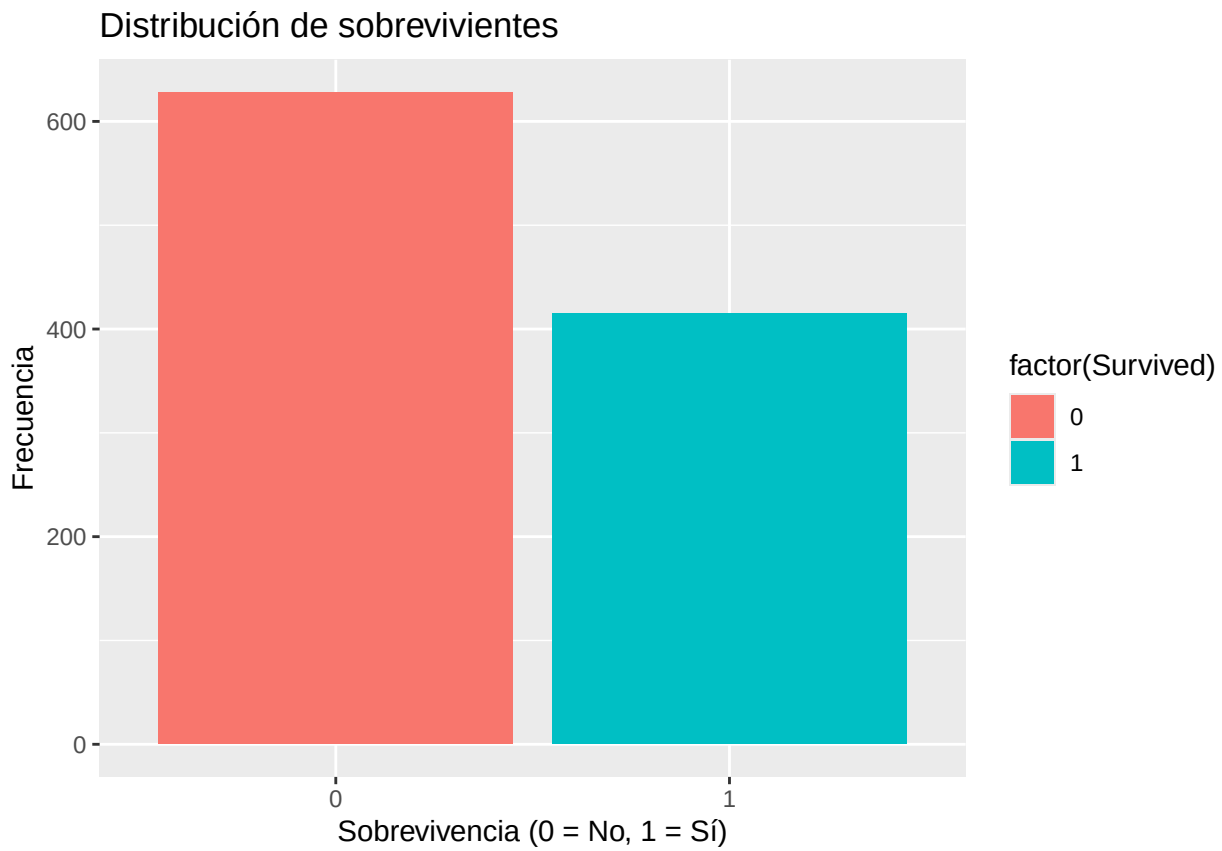
```
# Comprobar proporción de sobrevivientes después de limpiar
prop.table(table(titanic_clean$Survived))
```

```
##
##         0         1
## 0.6021093 0.3978907
```

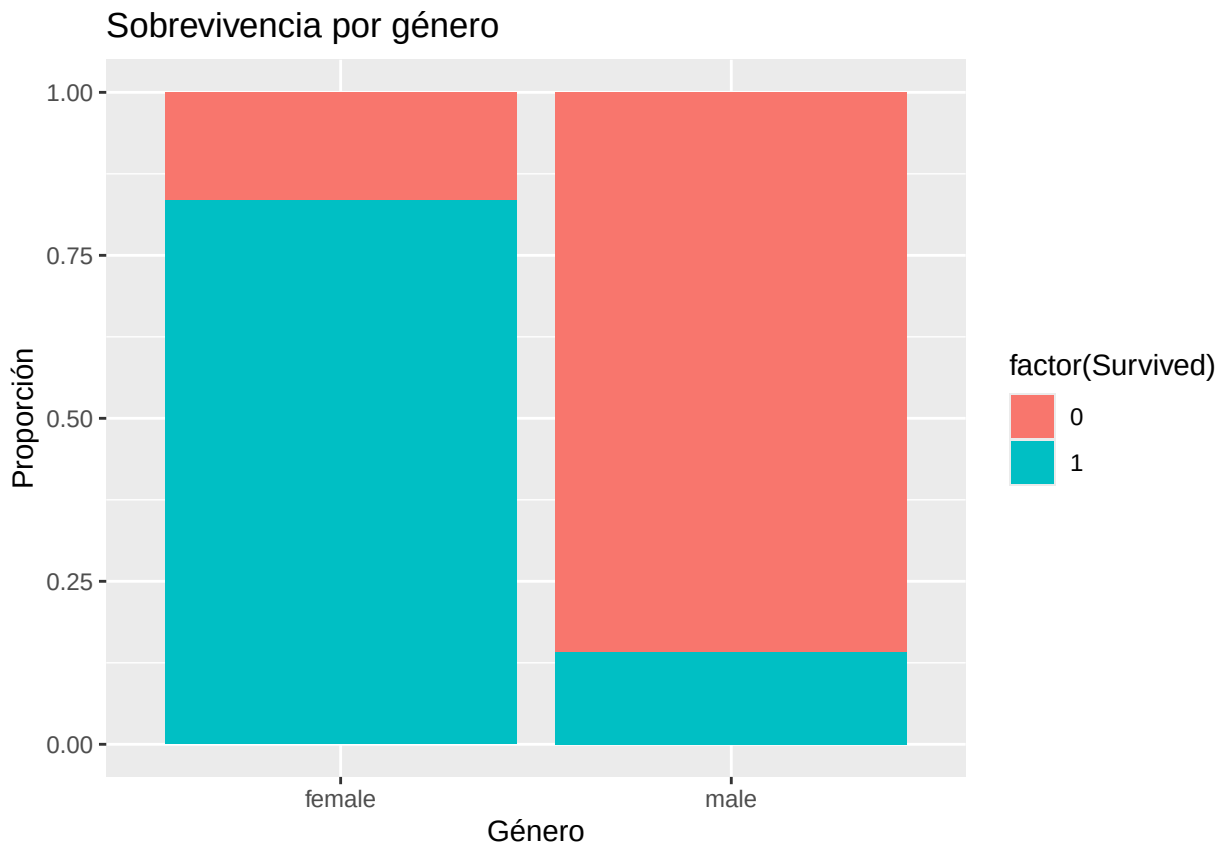
```
titanic = titanic_clean
```

Realiza un análisis descriptivo

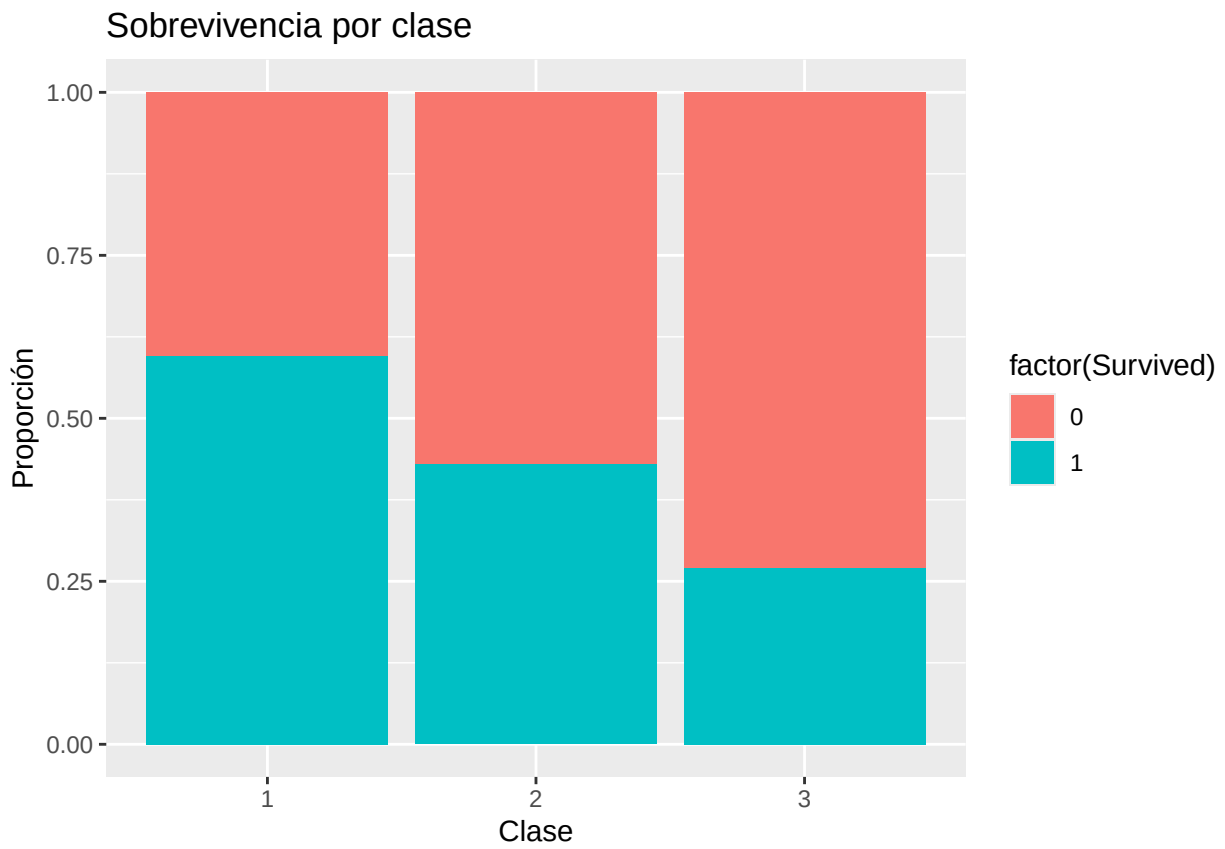
```
# Distribución de sobrevivientes
ggplot(titanic, aes(x = factor(Survived), fill = factor(Survived))) +
  geom_bar() +
  labs(title = "Distribución de sobrevivientes", x = "Sobrevivencia (0 = No, 1 = Sí)", y = "Frecuencia")
```




```r
# Sexo y sobrevivencia
ggplot(titanic, aes(x = Sex, fill = factor(Survived))) +
  geom_bar(position = "fill") +
  labs(title = "Sobrevivencia por género", x = "Género", y = "Proporción")
```




```
# Clase y sobrevivencia
ggplot(titanic, aes(x = factor(Pclass), fill = factor(Survived))) +
  geom_bar(position = "fill") +
  labs(title = "Sobrevivencia por clase", x = "Clase", y = "Proporción")
```



Observamos cómo la mayoría de las personas no se sobrevivieron en el incidente del titanic, con más del 60% de personas. Por otro lado, observamos cómo el género influye en la sobrevivencia, pues la gran mayoría de mujeres sobrevivió, con respecto a los hombres, los cuales muy pocos sobrevivieron. Finalmente, observamos cómo la clase influye, pues la clase 1 tuvo mayor proporción de sobrevivencia que la clase 2 y 3.

Haz una partición de los datos (70-30) para el entrenamiento y la validación. Revisa la proporción de sobrevivientes para la partición y la base original.

```
# Partición de datos
set.seed(123) # Para reproducibilidad
trainIndex <- createDataPartition(titanic$Survived, p = 0.7, list = FALSE)
trainData <- titanic[trainIndex, ]
validData <- titanic[-trainIndex, ]

# Verificar la proporción de sobrevivientes en cada conjunto
cat("Proporción en conjunto de entrenamiento:\n")
```

```
## Proporción en conjunto de entrenamiento:
```

```
prop.table(table(trainData$Survived))
```

```
##
##         0         1
## 0.6073871 0.3926129
```

```r
cat("\nProporción en conjunto de validación:\n")
```

```
##
## Proporción en conjunto de validación:
```

```r
prop.table(table(validData$Survived))
```

```
##
##         0         1
## 0.5897436 0.4102564
```

```r
cat("\nProporción en toda la base de datos:\n")
```

```
##
## Proporción en toda la base de datos:
```

```r
prop.table(table(titanic$Survived))
```

```
##
##         0         1
## 0.6021093 0.3978907
```

Observamos cómo se mantiene una proporcióm similar para la base de datos de entrenamiento y validación, con respecto de la original.

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.

Propón por lo menos los dos que consideres mejores modelos.

```r
null_model <- glm(Survived ~ 1, data = trainData, family = binomial)
full_model <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                  data = trainData, family = binomial)
mixed_model <- step(null_model,
                    scope = list(lower = null_model, upper = full_model),
                    direction = "both")
```

```
## Start:  AIC=981.4
## Survived ~ 1
##
##          Df Deviance    AIC
## + Sex     1   642.21 646.21
## + Pclass  1   924.26 928.26
```

```
## + Fare    1   940.49 944.49
## + Parch   1   969.78 973.78
## + Age     1   976.02 980.02
## <none>        979.40 981.40
## + SibSp   1   978.49 982.49
##
## Step:  AIC=646.21
## Survived ~ Sex
##
##          Df Deviance    AIC
## + Pclass  1   599.81 605.81
## + Fare    1   625.32 631.32
## + SibSp   1   640.15 646.15
## <none>        642.21 646.21
## + Age     1   641.52 647.52
## + Parch   1   641.57 647.57
## - Sex     1   979.40 981.40
##
## Step:  AIC=605.81
## Survived ~ Sex + Pclass
##
##          Df Deviance    AIC
## + Age     1   581.68 589.68
## <none>        599.81 605.81
## + SibSp   1   598.42 606.42
## + Fare    1   599.26 607.26
## + Parch   1   599.55 607.55
## - Pclass  1   642.21 646.21
## - Sex     1   924.26 928.26
##
## Step:  AIC=589.68
## Survived ~ Sex + Pclass + Age
##
##          Df Deviance    AIC
## + SibSp   1   577.11 587.11
## <none>        581.68 589.68
## + Parch   1   580.59 590.59
## + Fare    1   581.40 591.40
## - Age     1   599.81 605.81
## - Pclass  1   641.52 647.52
## - Sex     1   890.73 896.73
##
## Step:  AIC=587.11
## Survived ~ Sex + Pclass + Age + SibSp
##
##          Df Deviance    AIC
## <none>        577.11 587.11
## + Fare    1   576.36 588.36
## + Parch   1   577.00 589.00
## - SibSp   1   581.68 589.68
## - Age     1   598.42 606.42
## - Pclass  1   638.76 646.76
## - Sex     1   890.69 898.69
```

```r
summary(mixed_model)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age + SibSp, family = binomial,
##     data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4787  -0.5708  -0.3683   0.4981   2.5203
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.341614   0.580188   9.207  < 2e-16 ***
## Sexmale     -3.438301   0.233736 -14.710  < 2e-16 ***
## Pclass      -1.110879   0.150124  -7.400 1.36e-13 ***
## Age         -0.037868   0.008423  -4.496 6.94e-06 ***
## SibSp       -0.259574   0.124230  -2.089   0.0367 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 979.40  on 730  degrees of freedom
## Residual deviance: 577.11  on 726  degrees of freedom
## AIC: 587.11
##
## Number of Fisher Scoring iterations: 5
```

```r
cat("AIC del modelo seleccionado (Mixed Stepwise):", AIC(mixed_model), "\n")
```

```
## AIC del modelo seleccionado (Mixed Stepwise): 587.1135
```

Como podemos observar, el mejor modelo de AIC involucra las variables de Pclass, Sex, Age y SbSip con un valor de 587.1135. Para ello, realizaremos dos modelos: el mejor con AIC y este modelo, pero con interacción
Mejor modelo, con respecto a AIC

```r
model_AIC <- glm(Survived ~ Pclass + Sex + Age + SibSp,
                 data = trainData, family = binomial)
model_AIC_2 <- glm(Survived ~ Pclass * Sex * Pclass * Age, data = trainData, family = binomial)
```

Analiza los modelos a través de:

Identificación de la Desviación residual de cada modelo

Identificación de la Desviación nula

```r
# Desviación nula y residual para model_AIC
cat("Desviación nula (model_AIC):", model_AIC$null.deviance, "\n")
```

```
## Desviación nula (model_AIC): 979.3975
```

```r
cat("Desviación residual (model_AIC):", model_AIC$deviance, "\n")
```

```
## Desviación residual (model_AIC): 577.1135
```

```r
# Desviación nula y residual para model_AIC_2
cat("Desviación nula (model_AIC_2):", model_AIC_2$null.deviance, "\n")
```

```
## Desviación nula (model_AIC_2): 979.3975
```

```r
cat("Desviación residual (model_AIC_2):", model_AIC_2$deviance, "\n")
```

```
## Desviación residual (model_AIC_2): 561.9124
```

La desviación nulo representa el error del modelo base sin predictores, que solo considera la media de la variable objetivo. En esta, se obtuvo un valor de 577.11 para el modelo 1, mientras se obtuvo un modelo de 581.6785 para el modelo 2. Por otro lado, la desviación residual representa el error del modelo después de incluir los predictores. Como podemos observar, se obtiene una desviación de 577.11 para el modelo 1, mientras se obtiene una desviación residual de 561.9124 para el segundo modelo.

Cálculo de la Desviación Explicada

```r
# Desviación explicada para model_AIC
dev_exp_model_AIC <- (model_AIC$null.deviance - model_AIC$deviance) / model_AIC$null.deviance
cat("Desviación explicada (model_AIC):", round(dev_exp_model_AIC * 100, 2), "%\n")
```

```
## Desviación explicada (model_AIC): 41.07 %
```

```r
# Desviación explicada para model_AIC_2
dev_exp_model_AIC_2 <- (model_AIC_2$null.deviance - model_AIC_2$deviance) / model_AIC_2$null.deviance
cat("Desviación explicada (model_AIC_2):", round(dev_exp_model_AIC_2 * 100, 2), "%\n")
```

```
## Desviación explicada (model_AIC_2): 42.63 %
```

Como podemos observar, obtenemos un 41.07% de desviación explicada para el modelo sin interacción y un 42.63% para el modelo con interacción

Prueba de la razón de verosimilitud

H_0 = no hay diferencia significativa entre el modelo nulo y el alternativo. H_1 = hay una diferencia significativa entre el modelo nulo y el alternativo.

```r
# Prueba de razón de verosimilitud entre los modelos
lr_test <- anova(model_AIC, model_AIC_2, test = "Chisq")
print(lr_test)
```

```
## Analysis of Deviance Table
##
## Model 1: Survived ~ Pclass + Sex + Age + SibSp
## Model 2: Survived ~ Pclass * Sex * Pclass * Age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       726     577.11
## 2       723     561.91  3   15.201 0.001653 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Debido a que nuestro valor p <0.05, rechazamos la hipotésis inicial que indica que la diferencia de modelos no es significativa, por lo que la diferencia sí es significativa y el modelo 2 es estadísticamente significativo (mejor). ## Define cuál es el mejor modelo ## Escribe su ecuación, analiza sus coeficientes y detecta el efecto de cada predictor en la clasificación. Debido a que el modelo con interacción es mejor, de acuerdo con nuestro análisis, nos quedaremos con eso.

```r
best_model = model_AIC_2
summary(best_model)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass * Sex * Pclass * Age, family = binomial,
##     data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0601  -0.5758  -0.3863   0.3560   2.6161
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         8.35609    2.39639   3.487 0.000489 ***
## Pclass             -2.51719    0.83387  -3.019 0.002539 **
## Sexmale            -7.54353    2.58624  -2.917 0.003537 **
## Age                -0.06429    0.05553  -1.158 0.246935
## Pclass:Sexmale      1.97641    0.92398   2.139 0.032434 *
## Pclass:Age          0.01764    0.02025   0.871 0.383580
## Sexmale:Age         0.03978    0.06123   0.650 0.515916
## Pclass:Sexmale:Age -0.02858    0.02374  -1.204 0.228725
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 979.40  on 730  degrees of freedom
## Residual deviance: 561.91  on 723  degrees of freedom
## AIC: 577.91
##
## Number of Fisher Scoring iterations: 6
```

Observando los resultados de nuestro estadístico z y nuestros valores p, vemos que el intercepto, Pclass, Sexmale y la interacción de Pclass con sexmale rechazan la hipótesis inicial con un nivel de significancia de 0.05, por lo que estas variables son significativas para el modelo. Para el resto de coeficientes no se rechaza la hipótesis inicial, por lo que no son estadísticamente significativas.

Posteriormente, analizamos los efectos de cada predictor: Intercepto:8.356098.35, el cual representa el valor de log-odds cuando todas las variables predictoras son cero. Pclass: -2.51719, lo cual ndica que pertenecer a una clase más baja reduce las probabilidades de supervivencia. Sexmale: -7.54353), lo cual indica que ser hombre reduce significativamente las probabilidades de supervivencia en comparación con ser mujer. Age: -0.06429, por lo que la edad no tiene un efecto importante en este modelo. Pclass: Sexmale 1.97641, la interacción entre la clase y ser hombre tiene un impacto positivo en la probabilidad de sobrevivir, pero con una magnitud menor que los efectos principales. Pclass:Age 0.01764, la interacción entre la edad y clase tiene un impacto positivo en la probabilidad de sobrevivir con un efecto menor. Sexmale:Age 0.03978, la interacción entre ser hombre y la edad tiene un impacto positivo en la probabilidad de sobrevivir con un efecto menor. Pclass:Sexmale: Age 0.01764, la interacción entre la edad, ser hombre y la clase tiene un impacto positivo en la probabilidad de sobrevivir con un efecto menor.

Ecuación del modelo

$$\text{logit}(\text{Survived}) = 8.35609 - 2.51719 \cdot \text{Pclass} - 7.54353 \cdot \text{Sexmale} - 0.06429 \cdot \text{Age} + 1.97641 \cdot (\text{Pclass} : \text{Sexmale}) + 0.01764 \cdot (\text{Pclass} : \text{Age}$$

Es decir, por cada incremento en una unidad en Pclass, por ejemplo, disminuye en -2.5179 el valor del logit de Survived.

Analiza las predicciones para los datos de entrenamiento

Elabora la matriz de confusión

```
# Predicciones en el conjunto de entrenamiento
train_predictions <- predict(best_model, newdata = trainData, type = "response")

# Convertir probabilidades a clases (0 o 1)
train_predicted_classes <- ifelse(train_predictions > 0.5, 1, 0)

train_predicted_classes <- factor(train_predicted_classes, levels = c(0, 1))
trainData$Survived <- factor(trainData$Survived, levels = c(0, 1))
# Calcular precisión, sensibilidad y especificidad
library(caret)
confusionMatrix(train_predicted_classes, trainData$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 398  70
##          1  46 217
##
##                Accuracy : 0.8413
##                  95% CI : (0.8128, 0.8671)
##     No Information Rate : 0.6074
##     P-Value [Acc > NIR] : < 2e-16
```

```
##
##                   Kappa : 0.6623
##
##  Mcnemar's Test P-Value : 0.03272
##
##             Sensitivity : 0.8964
##             Specificity : 0.7561
##          Pos Pred Value : 0.8504
##          Neg Pred Value : 0.8251
##              Prevalence : 0.6074
##          Detection Rate : 0.5445
##    Detection Prevalence : 0.6402
##       Balanced Accuracy : 0.8262
##
##        'Positive' Class : 0
##
```

Como podemos observar, el modelo tiene una exactitud del 85%, lo cual indica que el modelo tiene un buen ajuste a la predicción de si la persona sobrevivió o no.

Elabora la Curva ROC

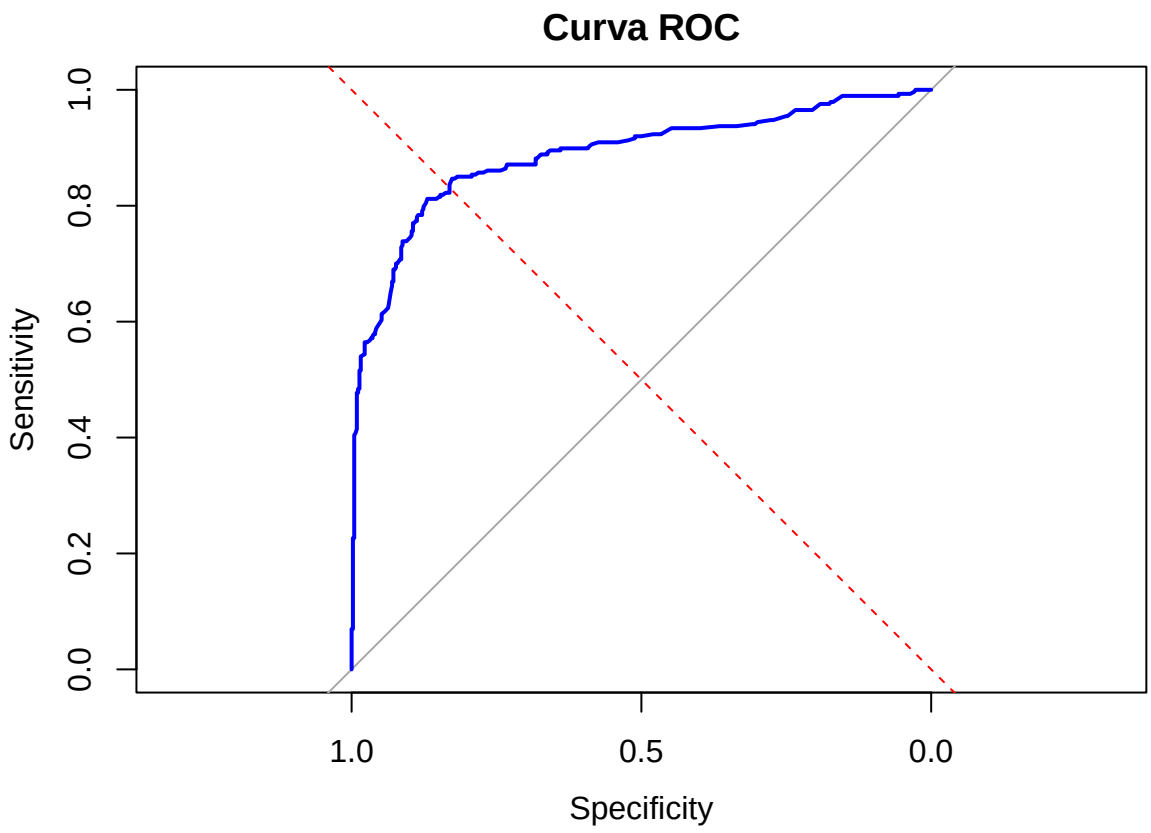
```
# Calcular valores para la curva ROC
library(pROC)
```

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

##
## Attaching package: 'pROC'

## The following object is masked from 'package:colorspace':
##
##     coords

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

```
roc_curve <- roc(trainData$Survived, train_predictions)
```

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

## Setting direction: controls < cases
```

```
# Graficar la curva ROC
plot(roc_curve, col = "blue", lwd = 2, main = "Curva ROC")
abline(a = 0, b = 1, lty = 2, col = "red") # Línea de referencia
```

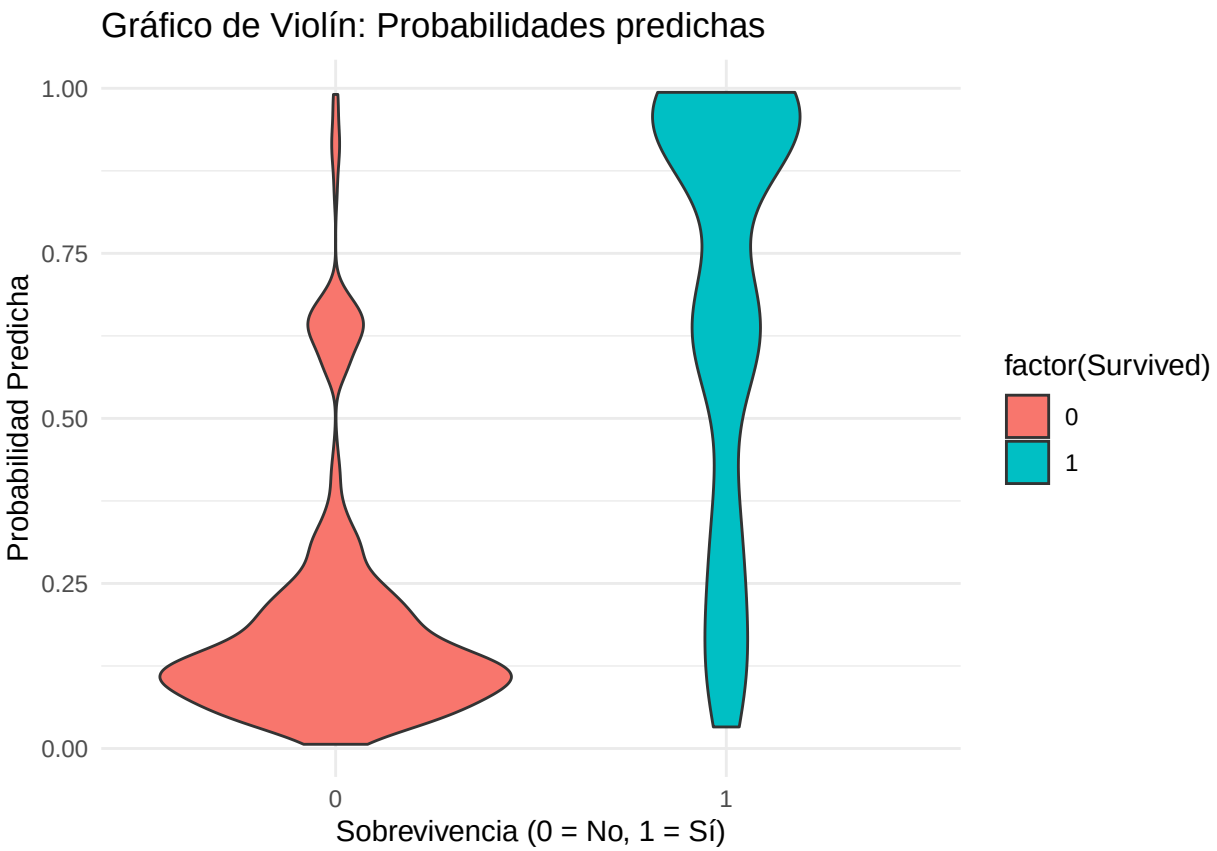



```
cat("AUC:", auc(roc_curve), "\n")
```

```
## AUC: 0.8871951
```

Como podemos observar, obtenemos un AUC de 0.887 confirma que el modelo es bastante bueno para clasificar correctamente las observaciones de entrenamiento. Sin embargo, se tiene que evaluar cómo se comporta en la prueba y ver si obtiene mejores resultados. ## Elabora el gráfico de violín

```
# Añadir las predicciones al conjunto de entrenamiento
trainData$Predicted_Prob <- train_predictions

# Graficar el gráfico de violín
library(ggplot2)
ggplot(trainData, aes(x = factor(Survived), y = Predicted_Prob, fill = factor(Survived))) +
  geom_violin() +
  labs(title = "Gráfico de Violín: Probabilidades predichas",
       x = "Sobrevivencia (0 = No, 1 = Sí)", y = "Probabilidad Predicha") +
  theme_minimal()
```



Como se observa en el gráfico de violín,hay una clara separación entre las distribuciones de ambas clases, por lo que el mode tiene un buen desempeño, ya que distingue bien entre sobrevivientes y no sobrevivientes. El modelo predice probabilidades consistentes con las clases observadas.

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

Fortalezas: El modelo tiene un buen desempeño general en términos de precisión y capacidad discriminativa. La sensibilidad (89.64%) asegura que el modelo predice correctamente la mayoría de los pasajeros que no sobrevivieron. La especificidad es razonable, aunque podría mejorarse para clasificar mejor a los sobrevivientes.

Limitaciones: La precisión para identificar sobrevivientes (especificidad) es más baja, lo cual pudiese perjudicar los resultados en prueba.

Decisión: El modelo es robusto para los datos de entrenamiento. Sin embargo, se debe validar en un conjunto de datos de validación y prueba para confirmar que el desempeño se generaliza y no está sobreajustado.

Validación del modelo con la base de datos de validación

Elije un umbral de clasificación óptimo

```r
valid_predictions <- predict(best_model, newdata = validData, type = "response")
library(pROC)
roc_curve_valid <- roc(validData$Survived, valid_predictions)
```

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

```
## Setting direction: controls < cases
```

```r
optimal_threshold <- coords(roc_curve_valid, "best", ret = "threshold")[[1]]

# Mostrar el umbral óptimo
cat("Umbral óptimo:", optimal_threshold, "\n")
```

```
## Umbral óptimo: 0.4094755
```

Como podemos observar, el umbral óptimo de clasificación es 0.4094755, por lo que las probs mayores a este umbral serán clasificadas como 1 (sobrevivió), mientras el resto se clasificará como 0 (no sobrevivió). ## Elabora la matriz de confusión con el umbral de clasificación óptimo

```r
# Convertir probabilidades a clases según el umbral óptimo
valid_predicted_classes <- ifelse(valid_predictions > optimal_threshold, 1, 0)

# Convertir a factores y alinear niveles
valid_predicted_classes <- factor(valid_predicted_classes, levels = c(0, 1))
validData$Survived <- factor(validData$Survived, levels = c(0, 1))

# Construir la matriz de confusión
conf_matrix_valid <- confusionMatrix(valid_predicted_classes, validData$Survived)
print(conf_matrix_valid)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 165  18
##          1  19 110
##
##                Accuracy : 0.8814
##                  95% CI : (0.8403, 0.9151)
##     No Information Rate : 0.5897
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7552
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8967
##             Specificity : 0.8594
##          Pos Pred Value : 0.9016
##          Neg Pred Value : 0.8527
##              Prevalence : 0.5897
```

```
##          Detection Rate : 0.5288
##    Detection Prevalence : 0.5865
##       Balanced Accuracy : 0.8781
##
##        'Positive' Class : 0
##
```

El modelo logístico ajustado basado en las variables Sex, Pclass, Age, y SibSp demuestra: Efectividad: Clasifica correctamente el 88.14% de las observaciones con un AUC de 0.887, lo que respalda su alta capacidad discriminativa. Balance: Tiene un buen balance entre sensibilidad y especificidad, lo que lo hace adecuado para tareas de clasificación en este contexto (87.81% de exactitud balanceada). Robustez: Generaliza bien en los datos de validación no vistos, lo que indica que no está sobreajustado.

Elabora el testeo con la base de datos de prueba.

```
testData <- read.csv("Titanic_test.csv")
testData <- testData %>% select(PassengerId, Pclass, Sex, Age, SibSp, Fare)
str(testData)
```

```
## 'data.frame':    418 obs. of  6 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ 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 ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
```

```
# Imputación de mediana para age y fare
testData$Age[is.na(testData$Age)] <- median(testData$Age, na.rm = TRUE)
testData$Fare[is.na(testData$Fare)] <- median(testData$Fare, na.rm = TRUE)

sapply(testData, function(x) sum(is.na(x)))
```

```
## PassengerId      Pclass         Sex         Age       SibSp        Fare
##           0           0           0           0           0           0
```

```
testData$Sex <- factor(testData$Sex, levels = c("male", "female"))
test_predictions <- predict(best_model, newdata = testData, type = "response")

optimal_threshold <- 0.4095
test_predicted_classes <- ifelse(test_predictions > optimal_threshold, 1, 0)
```

```
head(test_predicted_classes)
```

```
## 1 2 3 4 5 6
## 0 1 0 0 1 0
```

Estos son los resultados de los primeros seis datos para la prueba, pero no podemos comparar con las etiquetas reales, pues no se tienen estas.

#Concluye en el contexto del problema: ## Define las principales características que influyen en el modelo seleccionado e interpretalas: ¿qué características tuvieron las personas que sobrevivieron? Sex: El coeficiente negativo para Sexmale indica que ser hombre reduce significativamente la probabilidad de sobrevivir, por lo que las mujeres tuvieron prioridad para abordar los botes salvavidas, lo que explica su mayor tasa de sobrevivencia. Pclass: El coeficiente negativo para Pclass muestra que los pasajeros de clases más bajas (clase 3) tenían menor probabilidad de sobrevivir que los de primera clase. Esto indica que los pasajeros de primera clase tuvieron más acceso a los botes salvavidas, mientras que las clases bajas enfrentaron barreras para llegar a las cubiertas superiores. Age: El coeficiente negativo para Age indica que las personas más jóvenes tenían una mayor probabilidad de sobrevivir. SibSp: Un coeficiente negativo sugiere que los pasajeros con más familiares enfrentaron dificultades adicionales para sobrevivir. ## Interpreta los coeficientes del modelo

$$\text{logit}(\text{Survived}) = 8.35609 - 2.51719 \cdot \text{Pclass} - 7.54353 \cdot \text{Sexmale} - 0.06429 \cdot \text{Age} + 1.97641 \cdot (\text{Pclass} : \text{Sexmale}) + 0.01764 \cdot (\text{Pclass} : \text{Age}$$

Intercepto (8.356098.35609): Si todas las variables son 0, el logit de supervivencia es 8.35609. Es el valor base. -2.51719·Pclass: Por cada incremento en una unidad en Pclass, disminuye en 2.51719 el valor del logit de Survived, manteniendo constantes las demás variables. -7.54353·Sexmale: Si el pasajero es hombre, el logit de Survived disminuye en 7.54353 en comparación con si es mujer, manteniendo constantes las demás variables. -0.06429·Age-0.06429·Age: Por cada incremento en una unidad en Age (es decir, un año más de edad), disminuye en 0.06429 el valor del logit de SurvivedSurvived, manteniendo constantes las demás variables. 1.97641·(Pclass:Sexmale): Por cada incremento conjunto en una unidad en Pclass y Sexmale (es decir, por ser hombre en una peor clase socioeconómica), el logit de Survived aumenta en 1.97641, manteniendo constantes las demás variables. 0.01764·(Pclass:Age): Por cada incremento conjunto en una unidad en Pclass y Age (es decir, ser de una peor clase y un año más viejo), aumenta en 0.01764 el logit de Survived, manteniendo constantes las demás variables. 0.03978·(Sexmale:Age): Por cada incremento conjunto en una unidad en Sexmale y Age (es decir, un año más de edad en un hombre), el logit de Survived aumenta en 0.03978, manteniendo constantes las demás variables. -0.02858·(Pclass:Sexmale:Age): Por cada incremento conjunto en una unidad en Pclass, Sexmale, y Age (es decir, un hombre de una peor clase y un año más viejo), disminuye en 0.02858 el logit de Survived, manteniendo constantes las demás variables.

Define cuál es el mejor umbral de clasificación y por qué

Umbral seleccionado: 0.4095 Este umbral fue identificado utilizando la curva ROC para maximizar el balance entre sensibilidad y especificidad. Sensibilidad (89.67%): Alta capacidad para identificar correctamente a los pasajeros que no sobrevivieron. Especificidad (85.94%): Alta capacidad para identificar correctamente a los pasajeros que sobrevivieron. Balanced Accuracy (87.81%): Demuestra un balance óptimo entre ambas métricas.

Este umbral es óptimo para el problema porque logra un equilibrio adecuado entre minimizar falsos negativos (no identificar sobrevivientes) y falsos positivos (clasificar incorrectamente no sobrevivientes).

Conclusión Final

Las características clave para sobrevivir al Titanic fueron: ser mujer, estar en primera clase, ser joven y viajar sin familiares. El modelo logístico entrenado generaliza bien para la validación. El umbral de 0.4095 maximiza el desempeño del modelo, logrando un balance adecuado entre sensibilidad y especificidad.