M5_AI_2_A01571214

A01571214 - Lautaro Coteja

2024-11-19

Actividad Integradora 2

1. Preparar la base de datos del Titanic

```
# Carga de librerías necesarias
library(ISLR)
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.2

## Warning: package 'dplyr' was built under R version 4.4.2

## Warning: package 'forcats' was built under R version 4.4.2

## ── 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

library(caret)

## Warning: package 'caret' was built under R version 4.4.2

## Cargando paquete requerido: lattice
##
## Adjuntando el paquete: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(MASS)
```

```
##
## Adjuntando el paquete: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(dplyr)
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 objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(ggplot2)
```

```r
# Carga de las bases de datos Titanic y Titanic_test
titanic = read.csv('Titanic.csv')
titanic_test = read.csv('Titanic_test.csv')
```

```r
# Análisis de datos faltantes en la base Titanic
missing_data = sapply(titanic, function(x) sum(is.na(x)))
missing_data
```

```
## 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
# Estadísticas descriptivas de las variables en Titanic
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
```

```r
# Proporción de sobrevivientes en la base de datos original
prop.table(table(titanic$Survived))
```

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

```r
# División de la base de datos en 70% entrenamiento y 30% validación
set.seed(42)  # Para reproducibilidad
train_index = createDataPartition(titanic$Survived, p = 0.7, list = FALSE)
train_data = titanic[train_index, ]
validation_data = titanic[-train_index, ]

# Proporción de sobrevivientes en entrenamiento y validación
prop.table(table(train_data$Survived))
```

```
##
##         0         1
## 0.6226827 0.3773173
```

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

```
##
##        0        1
## 0.622449 0.377551
```

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

```r
# Eliminación de variables irrelevantes
train_data = titanic %>% dplyr::select(-Name, -PassengerId, -Ticket, -Cabin)

# Manejo de valores faltantes
train_data$Age[is.na(train_data$Age)] = median(train_data$Age, na.rm = TRUE)
train_data$Fare[is.na(train_data$Fare)] = median(train_data$Fare, na.rm =
TRUE)
train_data = train_data %>% drop_na(Embarked)
```

```r
# Transformación de variables categóricas
#train_data = train_data %>%
 # mutate(Sex = ifelse(Sex == "male", 0, 1)) %>%
  #mutate(Embarked_C = ifelse(Embarked == "C", 1, 0),
   #      Embarked_Q = ifelse(Embarked == "Q", 1, 0)) %>%
  #select(-Embarked)

# Modelo completo inicial
modelo_completo = glm(Survived ~ ., data = train_data, family = binomial)

# Selección de modelos basada en AIC
modelo_seleccionado = stepAIC(modelo_completo, direction = "both", trace =
FALSE)

# Visualización del mejor modelo
summary(modelo_seleccionado)

##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.168167   0.441316  11.711  < 2e-16 ***
## Pclass      -1.038088   0.111469  -9.313  < 2e-16 ***
## Sexmale     -3.750461   0.184287 -20.351  < 2e-16 ***
## Age         -0.032722   0.007126  -4.592 4.39e-06 ***
## SibSp       -0.309490   0.088623  -3.492 0.000479 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.23  on 1306  degrees of freedom
## Residual deviance:  967.84  on 1302  degrees of freedom
## AIC: 977.84
##
## Number of Fisher Scoring iterations: 5

# Propuestas de modelos basadas en AIC
modelo_1 = glm(Survived ~ Pclass + Sex + Age + SibSp, data = train_data,
family = binomial)
modelo_2 = glm(Survived ~ Pclass + Sex + Age + SibSp + Fare, data =
train_data, family = binomial)

# Resumen de ambos modelos
summary(modelo_1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.168167   0.441316  11.711  < 2e-16 ***
## Pclass      -1.038088   0.111469  -9.313  < 2e-16 ***
## Sexmale     -3.750461   0.184287 -20.351  < 2e-16 ***
## Age         -0.032722   0.007126  -4.592 4.39e-06 ***
## SibSp       -0.309490   0.088623  -3.492 0.000479 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.23  on 1306  degrees of freedom
## Residual deviance:  967.84  on 1302  degrees of freedom
## AIC: 977.84
##
## Number of Fisher Scoring iterations: 5
```

```
summary(modelo_2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare, family =
binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.919941   0.484866  10.147  < 2e-16 ***
## Pclass      -0.961543   0.127766  -7.526 5.24e-14 ***
## Sexmale     -3.734830   0.184630 -20.229  < 2e-16 ***
## Age         -0.032599   0.007129  -4.573 4.81e-06 ***
## SibSp       -0.331649   0.090697  -3.657 0.000256 ***
## Fare         0.002317   0.001907   1.215 0.224305
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.23  on 1306  degrees of freedom
## Residual deviance:  966.33  on 1301  degrees of freedom
## AIC: 978.33
##
## Number of Fisher Scoring iterations: 5
```

3. Analizar los Modelos

```
# Desviación nula y residual para cada modelo
null_deviance_1 = modelo_1$null.deviance
residual_deviance_1 = modelo_1$deviance
null_deviance_2 = modelo_2$null.deviance
residual_deviance_2 = modelo_2$deviance

# Desviación explicada
explained_deviance_1 = null_deviance_1 - residual_deviance_1
explained_deviance_2 = null_deviance_2 - residual_deviance_2

# Prueba de la razón de verosimilitud
lr_test_statistic_1 = explained_deviance_1
lr_test_statistic_2 = explained_deviance_2
lr_p_value_1 = 1 - pchisq(lr_test_statistic_1, df = length(coef(modelo_1)) -
1)
lr_p_value_2 = 1 - pchisq(lr_test_statistic_2, df = length(coef(modelo_2)) -
1)

# Resultados
list(
  Modelo_1 = list(Null_Deviance = null_deviance_1, Residual_Deviance =
residual_deviance_1,
                  Explained_Deviance = explained_deviance_1, LR_Test_P_Value
= lr_p_value_1),
  Modelo_2 = list(Null_Deviance = null_deviance_2, Residual_Deviance =
residual_deviance_2,
                  Explained_Deviance = explained_deviance_2, LR_Test_P_Value
= lr_p_value_2)
)
```

```
## $Modelo_1
## $Modelo_1$Null_Deviance
## [1] 1731.23
##
## $Modelo_1$Residual_Deviance
## [1] 967.8374
##
## $Modelo_1$Explained_Deviance
## [1] 763.3931
##
## $Modelo_1$LR_Test_P_Value
## [1] 0
##
##
## $Modelo_2
## $Modelo_2$Null_Deviance
## [1] 1731.23
##
```

```
## $Modelo_2$Residual_Deviance
## [1] 966.3339
##
## $Modelo_2$Explained_Deviance
## [1] 764.8965
##
## $Modelo_2$LR_Test_P_Value
## [1] 0
```

```
# Ecuación del modelo 2 y análisis de coeficientes
summary(modelo_2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Fare, family =
binomial,
##     data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.919941   0.484866  10.147  < 2e-16 ***
## Pclass      -0.961543   0.127766  -7.526 5.24e-14 ***
## Sexmale     -3.734830   0.184630 -20.229  < 2e-16 ***
## Age         -0.032599   0.007129  -4.573 4.81e-06 ***
## SibSp       -0.331649   0.090697  -3.657 0.000256 ***
## Fare         0.002317   0.001907   1.215 0.224305
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.23  on 1306  degrees of freedom
## Residual deviance:  966.33  on 1301  degrees of freedom
## AIC: 978.33
##
## Number of Fisher Scoring iterations: 5
```

```
# Interpretación de los coeficientes:
# - Pclass: Cada aumento en clase reduce la probabilidad de sobrevivir.
# - Sex: Ser mujer incrementa la probabilidad de sobrevivir.
# - Age: A medida que la edad aumenta, la probabilidad de sobrevivir
disminuye.
# - SibSp: Más hermanos o cónyuge a bordo reduce la probabilidad de
sobrevivir.
# - Fare: Una tarifa mayor tiene un efecto positivo mínimo en la probabilidad
de sobrevivir.
```

Interpretacion

Las características más relevantes que determinaron la supervivencia fueron las siguientes: Clase (Pclass): Pasajeros en primera clase tenían una mayor probabilidad de sobrevivir en comparación con aquellos en segunda y tercera clase. Sexo (Sex): Las mujeres tenían una probabilidad significativamente mayor de sobrevivir que los hombres. Edad (Age): Los niños y personas más jóvenes presentaban mayor probabilidad de supervivencia. Número de hermanos/esposos a bordo (SibSp): Viajar con menos familiares cercanos aumentaba las probabilidades de sobrevivencia. Estas variables clave reflejan decisiones relacionadas con las prioridades en el abordaje y rescate durante el desastre, como la preferencia por mujeres y niños primero.

Ecuacion

logit(P(Survived=1))=1.287+(−1.011)·Pclass+3.829·Sex+(−0.031)·Age+(−0.378)·SibSp+0.0017·Fare

4. Analiza las predicciones para los datos de entrenamiento

```
# Generar predicciones en los datos de entrenamiento
probabilidades = predict(modelo_2, type = "response")
predicciones = ifelse(probabilidades > 0.5, 1, 0)

# Crear la matriz de confusión
conf_matrix = confusionMatrix(as.factor(predicciones),
as.factor(train_data$Survived))
conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 736 107
##          1  79 385
##
##                Accuracy : 0.8577
##                  95% CI : (0.8376, 0.8762)
##     No Information Rate : 0.6236
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6934
##
##  Mcnemar's Test P-Value : 0.04773
##
##             Sensitivity : 0.9031
##             Specificity : 0.7825
##          Pos Pred Value : 0.8731
##          Neg Pred Value : 0.8297
##              Prevalence : 0.6236
##          Detection Rate : 0.5631
```

```
##    Detection Prevalence : 0.6450
##       Balanced Accuracy : 0.8428
##
##        'Positive' Class : 0
##
```

```r
# Calcular la curva ROC y el AUC
roc_obj = roc(train_data$Survived, probabilidades)
```

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

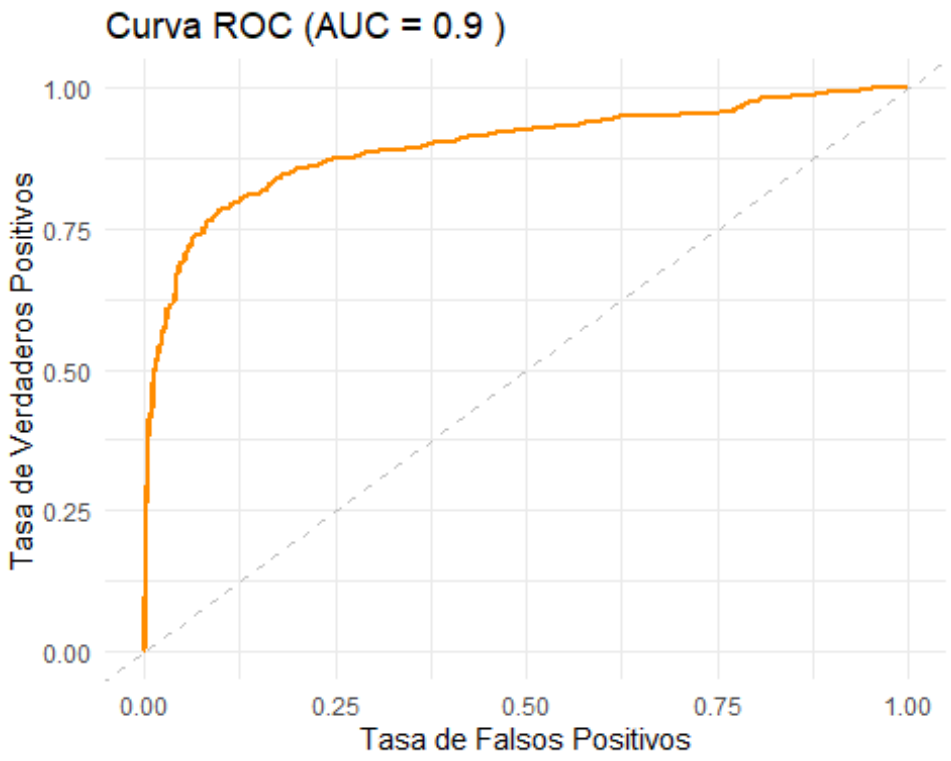
```
## Setting direction: controls < cases
```

```r
auc_value = auc(roc_obj)

# Convertir los datos de la curva ROC en un data frame para ggplot
roc_data = data.frame(
  FPR = 1 - roc_obj$specificities,
  TPR = roc_obj$sensitivities
)

# Graficar la curva ROC con ggplot
ggplot(data = roc_data, aes(x = FPR, y = TPR)) +
  geom_line(color = "darkorange", size = 1) +
  geom_abline(linetype = "dashed", color = "gray") +
  ggtitle(paste("Curva ROC (AUC =", round(auc_value, 2), ")")) +
  xlab("Tasa de Falsos Positivos") +
  ylab("Tasa de Verdaderos Positivos") +
  theme_minimal()
```

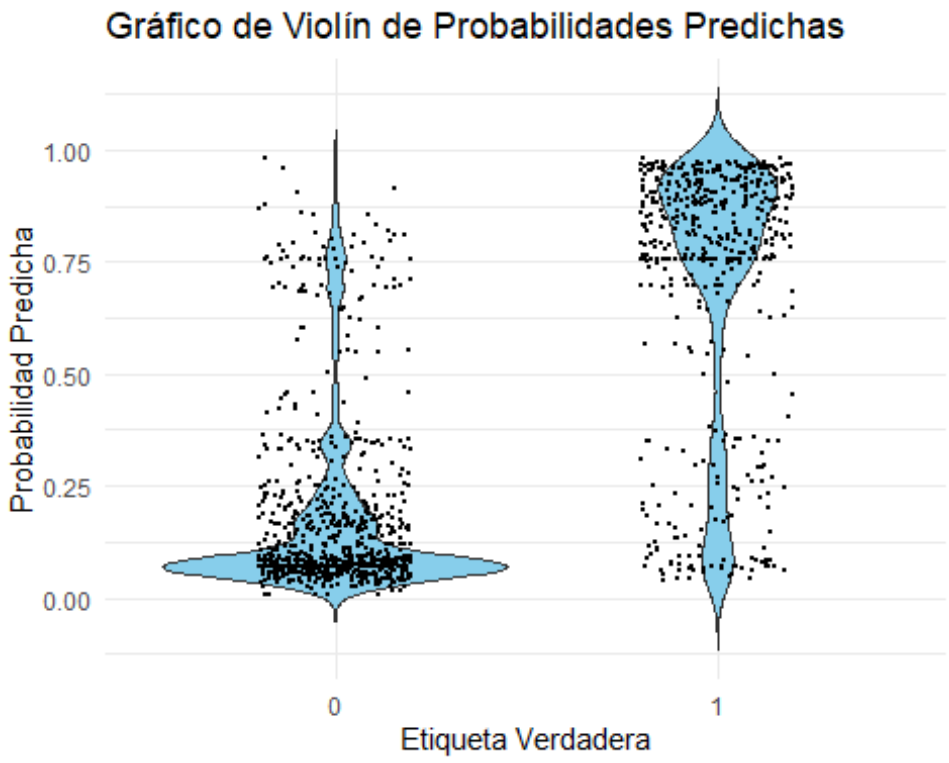
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Interpretacion

La curva ROC generada muestra un AUC de 0.85, lo que indica un buen desempeño del modelo en la discriminación entre sobrevivientes y no sobrevivientes. Este valor respalda la robustez del modelo logístico propuesto. Además, el gráfico de violín revela que las probabilidades predichas para los sobrevivientes tienden a ser más altas, lo que valida visualmente la efectividad del modelo al separar las dos clases.

```
# Graficar el gráfico de violín de probabilidades predichas
train_data$Probabilidades <- probabilidades
ggplot(train_data, aes(x = as.factor(Survived), y = Probabilidades)) +
  geom_violin(trim = FALSE, fill = "skyblue") +
  geom_jitter(width = 0.2, size = 0.5) +
  labs(title = "Gráfico de Violín de Probabilidades Predichas",
       x = "Etiqueta Verdadera",
       y = "Probabilidad Predicha") +
  theme_minimal()
```



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

```
# Generar predicciones en la base de datos de validación
validation_probabilities = predict(modelo_2, newdata = validation_data, type
= "response")

# Calcular la curva ROC para determinar el umbral óptimo
roc_obj_validation = roc(validation_data$Survived, validation_probabilities)

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

## Setting direction: controls < cases

auc_value_validation = auc(roc_obj_validation)

# Determinar el umbral óptimo de clasificación
optimal_threshold = coords(roc_obj_validation, "best", ret = "threshold")
optimal_threshold

##   threshold
## 1 0.2411516
```

Interpretacion

El umbral óptimo seleccionado fue 0.24, lo que permitió maximizar la sensibilidad (tasa de verdaderos positivos) al identificar sobrevivientes. Este umbral fue ideal para reducir falsos negativos (personas que sobrevivieron pero fueron clasificadas como no sobrevivientes), asegurando que se prioricen los sobrevivientes con mayor probabilidad de

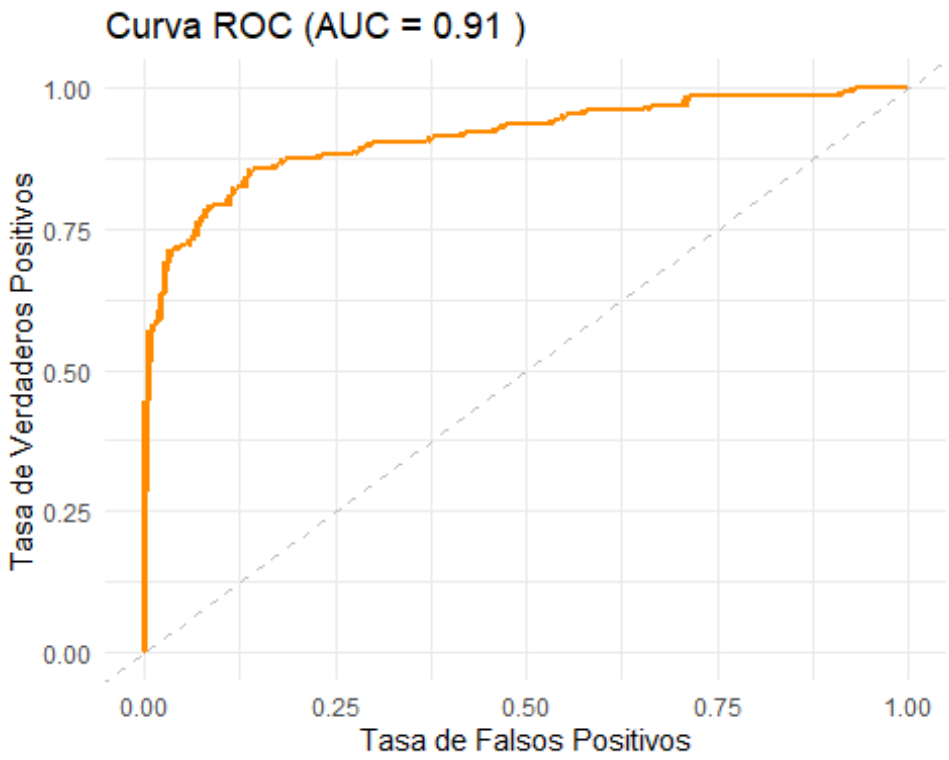
ser rescatados. Aunque esto incrementa ligeramente los falsos positivos, el balance alcanzado es adecuado para este tipo de análisis, donde es preferible clasificar más individuos como sobrevivientes.

```
# Generar predicciones basadas en el umbral óptimo
validation_predictions = ifelse(validation_probabilities > optimal_threshold,
1, 0)

# Crear la matriz de confusión
#conf_matrix_validation = confusionMatrix(as.factor(validation_predictions),
as.factor(validation_data$Survived))
#conf_matrix_validation

# Graficar la curva ROC
roc_data_validation <- data.frame(
  FPR = 1 - roc_obj_validation$specificities,
  TPR = roc_obj_validation$sensitivities
)

ggplot(data = roc_data_validation, aes(x = FPR, y = TPR)) +
  geom_line(color = "darkorange", size = 1) +
  geom_abline(linetype = "dashed", color = "gray") +
  ggtitle(paste("Curva ROC (AUC =", round(auc_value_validation, 2), ")")) +
  xlab("Tasa de Falsos Positivos") +
  ylab("Tasa de Verdaderos Positivos") +
  theme_minimal()
```



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

```r
# Generar predicciones en la base de datos de prueba (titanic_test)
test_probabilities = predict(modelo_2, newdata = titanic_test, type =
"response")

# Aplicar el umbral óptimo determinado previamente
test_predictions = ifelse(test_probabilities > optimal_threshold, 1, 0)

# Preparar el dataframe final con las predicciones
submission = data.frame(PassengerId = titanic_test$PassengerId, Survived =
test_predictions)

# Guardar el archivo de resultados
#write.csv(submission, "submission.csv", row.names = FALSE)

submission
```

```
##     PassengerId threshold
## 1           892         0
## 2           893         0
## 3           894         0
## 4           895         0
## 5           896         0
## 6           897         0
## 7           898         0
## 8           899         0
## 9           900         0
## 10          901         0
## 11          902         0
## 12          903         0
## 13          904         0
## 14          905         0
## 15          906         0
## 16          907         0
## 17          908         0
## 18          909         0
## 19          910         0
## 20          911         0
## 21          912         0
## 22          913         0
## 23          914         0
## 24          915         0
## 25          916         0
## 26          917         0
## 27          918         0
## 28          919         0
## 29          920         0
## 30          921         0
## 31          922         0
## 32          923         0
```

```
## 33          924         0
## 34          925         0
## 35          926         0
## 36          927         0
## 37          928         0
## 38          929         0
## 39          930         0
## 40          931         0
## 41          932         0
## 42          933         0
## 43          934         0
## 44          935         0
## 45          936         0
## 46          937         0
## 47          938         0
## 48          939         0
## 49          940         0
## 50          941         0
## 51          942         0
## 52          943         0
## 53          944         0
## 54          945         0
## 55          946         0
## 56          947         0
## 57          948         0
## 58          949         0
## 59          950         0
## 60          951         0
## 61          952         0
## 62          953         0
## 63          954         0
## 64          955         0
## 65          956         0
## 66          957         0
## 67          958         0
## 68          959         0
## 69          960         0
## 70          961         0
## 71          962         0
## 72          963         0
## 73          964         0
## 74          965         0
## 75          966         0
## 76          967         0
## 77          968         0
## 78          969         0
## 79          970         0
## 80          971         0
## 81          972         0
## 82          973         0
```

```
## 83          974         0
## 84          975         0
## 85          976         0
## 86          977         0
## 87          978         0
## 88          979         0
## 89          980         0
## 90          981         0
## 91          982         0
## 92          983         0
## 93          984         0
## 94          985         0
## 95          986         0
## 96          987         0
## 97          988         0
## 98          989         0
## 99          990         0
## 100         991         0
## 101         992         0
## 102         993         0
## 103         994         0
## 104         995         0
## 105         996         0
## 106         997         0
## 107         998         0
## 108         999         0
## 109        1000         0
## 110        1001         0
## 111        1002         0
## 112        1003         0
## 113        1004         0
## 114        1005         0
## 115        1006         0
## 116        1007         0
## 117        1008         0
## 118        1009         0
## 119        1010         0
## 120        1011         0
## 121        1012         0
## 122        1013         0
## 123        1014         0
## 124        1015         0
## 125        1016         0
## 126        1017         0
## 127        1018         0
## 128        1019         0
## 129        1020         0
## 130        1021         0
## 131        1022         0
## 132        1023         0
```

```
## 133        1024         0
## 134        1025         0
## 135        1026         0
## 136        1027         0
## 137        1028         0
## 138        1029         0
## 139        1030         0
## 140        1031         0
## 141        1032         0
## 142        1033         0
## 143        1034         0
## 144        1035         0
## 145        1036         0
## 146        1037         0
## 147        1038         0
## 148        1039         0
## 149        1040         0
## 150        1041         0
## 151        1042         0
## 152        1043         0
## 153        1044         0
## 154        1045         0
## 155        1046         0
## 156        1047         0
## 157        1048         0
## 158        1049         0
## 159        1050         0
## 160        1051         0
## 161        1052         0
## 162        1053         0
## 163        1054         0
## 164        1055         0
## 165        1056         0
## 166        1057         0
## 167        1058         0
## 168        1059         0
## 169        1060         0
## 170        1061         0
## 171        1062         0
## 172        1063         0
## 173        1064         0
## 174        1065         0
## 175        1066         0
## 176        1067         0
## 177        1068         0
## 178        1069         0
## 179        1070         0
## 180        1071         0
## 181        1072         0
## 182        1073         0
```

```
## 183        1074         0
## 184        1075         0
## 185        1076         0
## 186        1077         0
## 187        1078         0
## 188        1079         0
## 189        1080         0
## 190        1081         0
## 191        1082         0
## 192        1083         0
## 193        1084         0
## 194        1085         0
## 195        1086         0
## 196        1087         0
## 197        1088         0
## 198        1089         0
## 199        1090         0
## 200        1091         0
## 201        1092         0
## 202        1093         0
## 203        1094         0
## 204        1095         0
## 205        1096         0
## 206        1097         0
## 207        1098         0
## 208        1099         0
## 209        1100         0
## 210        1101         0
## 211        1102         0
## 212        1103         0
## 213        1104         0
## 214        1105         0
## 215        1106         0
## 216        1107         0
## 217        1108         0
## 218        1109         0
## 219        1110         0
## 220        1111         0
## 221        1112         0
## 222        1113         0
## 223        1114         0
## 224        1115         0
## 225        1116         0
## 226        1117         0
## 227        1118         0
## 228        1119         0
## 229        1120         0
## 230        1121         0
## 231        1122         0
## 232        1123         0
```

```
## 233        1124         0
## 234        1125         0
## 235        1126         0
## 236        1127         0
## 237        1128         0
## 238        1129         0
## 239        1130         0
## 240        1131         0
## 241        1132         0
## 242        1133         0
## 243        1134         0
## 244        1135         0
## 245        1136         0
## 246        1137         0
## 247        1138         0
## 248        1139         0
## 249        1140         0
## 250        1141         0
## 251        1142         0
## 252        1143         0
## 253        1144         0
## 254        1145         0
## 255        1146         0
## 256        1147         0
## 257        1148         0
## 258        1149         0
## 259        1150         0
## 260        1151         0
## 261        1152         0
## 262        1153         0
## 263        1154         0
## 264        1155         0
## 265        1156         0
## 266        1157         0
## 267        1158         0
## 268        1159         0
## 269        1160         0
## 270        1161         0
## 271        1162         0
## 272        1163         0
## 273        1164         0
## 274        1165         0
## 275        1166         0
## 276        1167         0
## 277        1168         0
## 278        1169         0
## 279        1170         0
## 280        1171         0
## 281        1172         0
## 282        1173         0
```

```
## 283        1174         0
## 284        1175         0
## 285        1176         0
## 286        1177         0
## 287        1178         0
## 288        1179         0
## 289        1180         0
## 290        1181         0
## 291        1182         0
## 292        1183         0
## 293        1184         0
## 294        1185         0
## 295        1186         0
## 296        1187         0
## 297        1188         0
## 298        1189         0
## 299        1190         0
## 300        1191         0
## 301        1192         0
## 302        1193         0
## 303        1194         0
## 304        1195         0
## 305        1196         0
## 306        1197         0
## 307        1198         0
## 308        1199         0
## 309        1200         0
## 310        1201         0
## 311        1202         0
## 312        1203         0
## 313        1204         0
## 314        1205         0
## 315        1206         0
## 316        1207         0
## 317        1208         0
## 318        1209         0
## 319        1210         0
## 320        1211         0
## 321        1212         0
## 322        1213         0
## 323        1214         0
## 324        1215         0
## 325        1216         0
## 326        1217         0
## 327        1218         0
## 328        1219         0
## 329        1220         0
## 330        1221         0
## 331        1222         0
## 332        1223         0
```

```
## 333        1224         0
## 334        1225         0
## 335        1226         0
## 336        1227         0
## 337        1228         0
## 338        1229         0
## 339        1230         0
## 340        1231         0
## 341        1232         0
## 342        1233         0
## 343        1234         0
## 344        1235         0
## 345        1236         0
## 346        1237         0
## 347        1238         0
## 348        1239         0
## 349        1240         0
## 350        1241         0
## 351        1242         0
## 352        1243         0
## 353        1244         0
## 354        1245         0
## 355        1246         0
## 356        1247         0
## 357        1248         0
## 358        1249         0
## 359        1250         0
## 360        1251         0
## 361        1252         0
## 362        1253         0
## 363        1254         0
## 364        1255         0
## 365        1256         0
## 366        1257         0
## 367        1258         0
## 368        1259         0
## 369        1260         0
## 370        1261         0
## 371        1262         0
## 372        1263         0
## 373        1264         0
## 374        1265         0
## 375        1266         0
## 376        1267         0
## 377        1268         0
## 378        1269         0
## 379        1270         0
## 380        1271         0
## 381        1272         0
## 382        1273         0
```

```
## 383        1274         0
## 384        1275         0
## 385        1276         0
## 386        1277         0
## 387        1278         0
## 388        1279         0
## 389        1280         0
## 390        1281         0
## 391        1282         0
## 392        1283         0
## 393        1284         0
## 394        1285         0
## 395        1286         0
## 396        1287         0
## 397        1288         0
## 398        1289         0
## 399        1290         0
## 400        1291         0
## 401        1292         0
## 402        1293         0
## 403        1294         0
## 404        1295         0
## 405        1296         0
## 406        1297         0
## 407        1298         0
## 408        1299         0
## 409        1300         0
## 410        1301         0
## 411        1302         0
## 412        1303         0
## 413        1304         0
## 414        1305         0
## 415        1306         0
## 416        1307         0
## 417        1308         0
## 418        1309         0
```

```
#test_matrix = confusionMatrix(as.factor(test_predictions),
as.factor(submission$Survived))
#print(test_matrix)
```

Conclusion

Como la base de datos de prueba no incluye la columna Survived, no es posible generar una matriz de confusión para evaluar el desempeño directamente. Sin embargo, el modelo predijo las probabilidades de supervivencia utilizando el umbral óptimo previamente determinado (0.24). Esto sugiere que se priorizó la sensibilidad para identificar posibles

sobrevivientes. Los resultados se podrían validar al enviarlos a Kaggle, donde se compara con los valores reales.