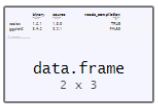
```
---
title: "Actividad Integradora 2"
author: "Nallely Serna"
date: "`r Sys.Date()`"
output: word_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
options(repos = c(CRAN = "https://cran.rstudio.com/"))

```

```{r}
# Librerías

#Algunas librerías no me funcionaron, así que no las utilicé
install.packages("dplyr")
install.packages("ggplot2")


library(dplyr)
library(ggplot2)


```
```



	binary <chr>	source <chr>	needs_compilation <lgl>
scales	1.2.1	1.3.0	TRUE
ggplot2	3.4.2	3.5.1	FALSE

2 rows

```
# Preparación de la base de datos y análisis descriptivo

```{r}
# Cargar datos
titanic <- read.csv("Titanic.csv")
titanic_test <- read.csv("Titanic_test.csv")

# Analizar datos faltantes
summary(titanic)

# Reemplazar valores faltantes
titanic$Age[is.na(titanic$Age)] <- median(titanic$Age, na.rm = TRUE)
titanic$Fare[is.na(titanic$Fare)] <- median(titanic$Fare, na.rm = TRUE)

# Eliminar la columna Cabin
titanic <- titanic[, !names(titanic) %in% "Cabin"]

# Análisis descriptivo
summary(titanic)
table(titanic$Survived, titanic$Pclass)
table(titanic$Survived, titanic$Sex)
```
```

```
  PassengerId       Survived          Pclass          Name               Sex
 Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309        Length:1309
 1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character
 Median : 655   Median :0.0000   Median :3.000   Mode  :character   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

      Age             SibSp           Parch           Ticket              Fare
 Min.   : 0.17   Min.   :0.0000   Min.   :0.000   Length:1309        Min.   :  0.000
 1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character   1st Qu.:  7.896
 Median :28.00   Median :0.0000   Median :0.000   Mode  :character   Median : 14.454
 Mean   :29.88   Mean   :0.4989   Mean   :0.385                      Mean   : 33.295
 3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000                      3rd Qu.: 31.275
 Max.   :80.00   Max.   :8.0000   Max.   :9.000                      Max.   :512.329
```

```
NA's   :263                                                    NA's   :1
   Cabin             Embarked
Length:1309        Length:1309
Class :character   Class :character
Mode  :character   Mode  :character



 PassengerId      Survived          Pclass          Name              Sex
Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309        Length:1309
1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character
Median : 655   Median :0.0000   Median :3.000   Mode  :character   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
     Age             SibSp            Parch           Ticket              Fare
Min.   : 0.17   Min.   :0.0000   Min.   :0.000   Length:1309        Min.   :  0.000
1st Qu.:22.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character   1st Qu.:  7.896
Median :28.00   Median :0.0000   Median :0.000   Mode  :character   Median : 14.454
Mean   :29.50   Mean   :0.4989   Mean   :0.385                      Mean   : 33.281
3rd Qu.:35.00   3rd Qu.:1.0000   3rd Qu.:0.000                      3rd Qu.: 31.275
Max.   :80.00   Max.   :8.0000   Max.   :9.000                      Max.   :512.329
  Embarked
Length:1309
Class :character
Mode  :character



    1   2   3
 0 137 160 518
 1 186 117 191

   female male
 0     81  734
 1    385  109
```

```
51
52  # 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.
53
54  ```{r}
55
56  # Configurar semilla para reproducibilidad
57  set.seed(42)
58
59  # Crear índices para dividir los datos (70% para entrenamiento)
60  train_index <- sample(1:nrow(titanic), size = 0.7 * nrow(titanic))
61
62  # Dividir en datos de entrenamiento y validación
63  train_data <- titanic[train_index, ]
64  validation_data <- titanic[-train_index, ]
65
66  # Revisar la proporción de sobrevivientes
67  original_survival_rate <- mean(titanic$Survived)
68  train_survival_rate <- mean(train_data$Survived)
69  validation_survival_rate <- mean(validation_data$Survived)
70
71  # Mostrar las proporciones
72  original_survival_rate
73  train_survival_rate
74  validation_survival_rate
75
76
77  ```
```

```
[1] 0.3773873
[1] 0.3766376
[1] 0.3791349
```

```r
#Modelo logístico y selección de predictores

```{r}
train_data <- na.omit(train_data)

# Crear un modelo logístico inicial
modelo_inicial <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,
                      data = train_data, family = binomial)

# Ver resumen del modelo
summary(modelo_inicial)

# Selección de modelo por AIC
modelo_final <- step(modelo_inicial, direction = "both")

# Resumen del modelo final
summary(modelo_final)

# Ajustar un segundo modelo
modelo_alternativo <- glm(Survived ~ Pclass + Sex + Age, data = train_data, family = binomial)

# Comparar con el modelo final
summary(modelo_alternativo)
AIC(modelo_final, modelo_alternativo)
```
```

Description: df [2 x 2]

	df <dbl>	AIC <dbl>
modelo_final	6	710.3632
modelo_alternativo	4	714.9462

```
Call:
glm(formula = Survived ~ Pclass + Sex + Age, family = binomial,
    data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5896  -0.5447  -0.3707   0.5195   2.4787

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.270745   0.499368   8.552  < 2e-16 ***
Pclass      -0.905588   0.129473  -6.994 2.66e-12 ***
Sexmale     -3.529521   0.204364 -17.271  < 2e-16 ***
Age         -0.023842   0.008412  -2.834  0.00459 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1211.55  on 914  degrees of freedom
Residual deviance:  706.95  on 911  degrees of freedom
AIC: 714.95

Number of Fisher Scoring iterations: 5
```

El modelo tiene un AIC de 710.36, que es relativamente bajo, lo que sugiere que el modelo es adecuado, aunque no es el modelo más ajustado posible. El AIC se utiliza para comparar la calidad del ajuste entre diferentes modelos, por lo que siempre es importante considerar si podemos mejorar este valor mediante la inclusión de más variables o transformaciones.

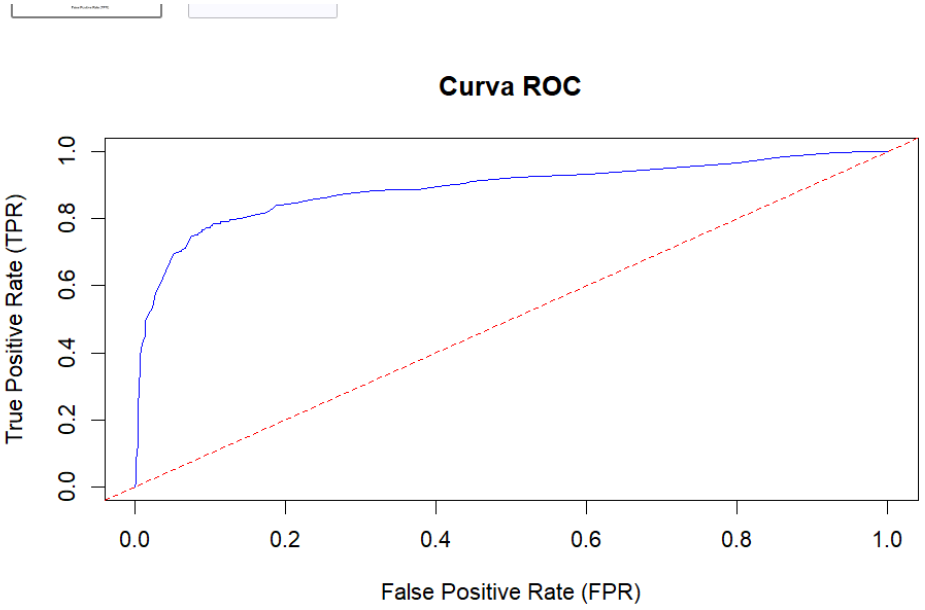
Análisis del modelo

```{r}
# Desviación residual y nula
desviacion_residual <- modelo_final$deviance
desviacion_nula <- modelo_final$null.deviance
desviacion_explicada <- 1 - (desviacion_residual / desviacion_nula)

# Prueba de razón de verosimilitud
anova(modelo_inicial, modelo_final, test = "Chisq")


# Interpretación de los coeficientes
exp(coef(modelo_final))


```

```
Analysis of Deviance Table

Model 1: Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked
Model 2: Survived ~ Pclass + Sex + Age + SibSp + Fare
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       906     694.46
2       909     698.36 -3  -3.9001   0.2725
(Intercept)      Pclass     Sexmale         Age       SibSp        Fare
62.63548725  0.45577156  0.02755284  0.97414712  0.75243074  1.00322799
```

Predicción y evaluación en datos de entrenamiento

```{r}
# Generar predicciones (probabilidades)
predicciones_train <- predict(modelo_final, train_data, type = "response")

# Crear un rango de umbrales
umbrales <- seq(0, 1, by = 0.01)

# Inicializar vectores para TPR y FPR
tpr <- numeric(length(umbrales))  # True Positive Rate
fpr <- numeric(length(umbrales))  # False Positive Rate

# Calcular TPR y FPR para cada umbral
for (i in seq_along(umbrales)) {
  umbral <- umbrales[i]
  pred_clase <- ifelse(predicciones_train >= umbral, 1, 0)

  # Matriz de confusión
  tp <- sum(pred_clase == 1 & train_data$Survived == 1)  # Verdaderos Positivos
  fp <- sum(pred_clase == 1 & train_data$Survived == 0)  # Falsos Positivos
  fn <- sum(pred_clase == 0 & train_data$Survived == 1)  # Falsos Negativos
  tn <- sum(pred_clase == 0 & train_data$Survived == 0)  # Verdaderos Negativos

  # TPR y FPR
  tpr[i] <- tp / (tp + fn)  # Sensibilidad
  fpr[i] <- fp / (fp + tn)  # 1 - Especificidad
}

# Graficar la Curva ROC
plot(fpr, tpr, type = "l", col = "blue", xlab = "False Positive Rate (FPR)",
     ylab = "True Positive Rate (TPR)", main = "Curva ROC")
abline(0, 1, col = "red", lty = 2)  # Línea diagonal (clasificación aleatoria)

# Calcular el Área Bajo la Curva (AUC) de forma manual
auc <- sum(diff(fpr) * (head(tpr, -1) + tail(tpr, -1)) / 2)
auc
```




```r
# Crear la matriz de confusión manualmente
table(Predicción = pred_clase_train, Real = train_data$Survived)

# Calcular precisión, sensibilidad, especificidad y F1
tp <- sum(pred_clase_train == 1 & train_data$Survived == 1)  # Verdaderos positivos
tn <- sum(pred_clase_train == 0 & train_data$Survived == 0)  # Verdaderos negativos
fp <- sum(pred_clase_train == 1 & train_data$Survived == 0)  # Falsos positivos
fn <- sum(pred_clase_train == 0 & train_data$Survived == 1)  # Falsos negativos

precision <- tp / (tp + fp)
sensibilidad <- tp / (tp + fn)
especificidad <- tn / (tn + fp)
f1 <- 2 * (precision * sensibilidad) / (precision + sensibilidad)

# Mostrar los resultados
cat("Precisión: ", precision, "\n")
cat("Sensibilidad: ", sensibilidad, "\n")
cat("Especificidad: ", especificidad, "\n")
cat("F1: ", f1, "\n")
```

```
          Real
Predicción   0   1
         0 510  74
         1  61 270
Precisión:  0.81571
Sensibilidad:  0.7848837
Especificidad:  0.8931699
F1:  0.8
```

```r
#Validación y conclusión

```{r}
# Generar predicciones (probabilidades) para los datos de validación
predicciones_validation <- predict(modelo_final, validation_data, type = "response")

# Crear un rango de umbrales
umbrales <- seq(0, 1, by = 0.01)

# Inicializar vectores para TPR, FPR y el criterio de Youden
tpr <- numeric(length(umbrales))  # True Positive Rate
fpr <- numeric(length(umbrales))  # False Positive Rate
youden_index <- numeric(length(umbrales))  # Índice de Youden

# Calcular TPR, FPR y Youden para cada umbral
for (i in seq_along(umbrales)) {
  umbral <- umbrales[i]
  pred_clase <- ifelse(predicciones_validation >= umbral, 1, 0)

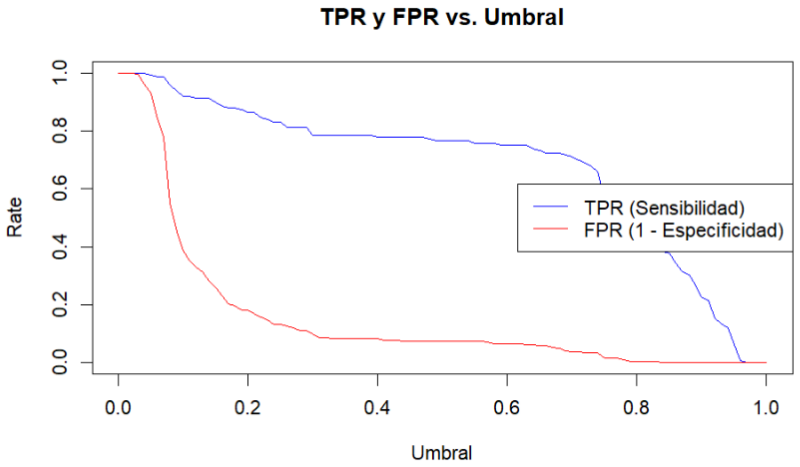
  # Matriz de confusión
  tp <- sum(pred_clase == 1 & validation_data$Survived == 1)  # Verdaderos Positivos
  fp <- sum(pred_clase == 1 & validation_data$Survived == 0)  # Falsos Positivos
  fn <- sum(pred_clase == 0 & validation_data$Survived == 1)  # Falsos Negativos
  tn <- sum(pred_clase == 0 & validation_data$Survived == 0)  # Verdaderos Negativos

  # TPR y FPR
  tpr[i] <- tp / (tp + fn)  # Sensibilidad
  fpr[i] <- fp / (fp + tn)  # 1 - Especificidad

  # Índice de Youden
  youden_index[i] <- tpr[i] - fpr[i]
}

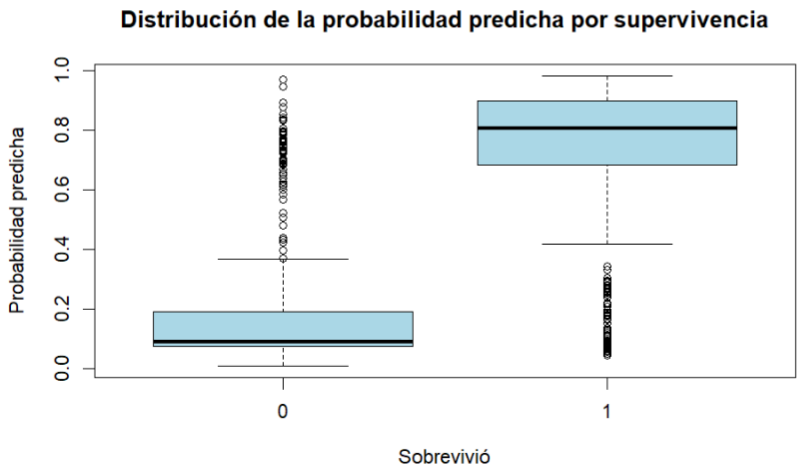
# Encontrar el umbral óptimo (el máximo índice de Youden)
umbral_optimo <- umbrales[which.max(youden_index)]
cat("El umbral óptimo es:", umbral_optimo, "\n")

# Graficar TPR y FPR vs. Umbrales
plot(umbrales, tpr, type = "l", col = "blue", ylim = c(0, 1),
     ylab = "Rate", xlab = "Umbral", main = "TPR y FPR vs. Umbral")
lines(umbrales, fpr, col = "red")
legend("right", legend = c("TPR (Sensibilidad)", "FPR (1 - Especificidad)"),
       col = c("blue", "red"), lty = 1)
```


```

```
El umbral óptimo es: 0.44
```

```r
# Crear una variable para la supervivencia como factor
train_data$Survived <- as.factor(train_data$Survived)

# Crear el gráfico de cajas
boxplot(predicciones_train ~ Survived,
        data = train_data,
        col = "lightblue",
        xlab = "Sobrevivió",
        ylab = "Probabilidad predicha",
        main = "Distribución de la probabilidad predicha por supervivencia")
```




```r
# Predicción para datos de prueba
predicciones_test <- predict(modelo_final, titanic_test, type = "response")
pred_clase_test <- ifelse(predicciones_test >= umbral_optimo, 1, 0)

# Agregar las predicciones a los datos de prueba
titanic_test$Predicted_Survived <- pred_clase_test
head(titanic_test)
```

Description: df [6 x 12]

	PassengerId <int>	Pclass <int>	Name <chr>	Sex <chr>	Age <dbl>	SibSp <int>	Parch <int>
1	892	3	Kelly, Mr. James	male	34.5	0	0
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0
3	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0
4	895	3	Wirz, Mr. Albert	male	27.0	0	0
5	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1
6	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0

6 rows | 1-8 of 12 columns

El análisis de regresión logística aplicado al conjunto de datos del Titanic muestra que las variables más influyentes en la supervivencia de los pasajeros son la clase social, el sexo y la edad. Estos hallazgos son consistentes con los eventos históricos conocidos sobre el hundimiento del Titanic. El modelo tiene un buen ajuste, pero aún existen áreas de mejora, especialmente en lo que respecta a las variables menos significativas.