Actividad Integradora 2

Saúl Francisco Vázquez del Río

2024-11-19

Bibliotecas

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

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

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

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

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

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. Utiliza en las siguientes bases de datos:

Base de datos del Titanic: TitanicDownload Titanic Base de datos de prueba: Titanic_test Download Titanic_test Las variables para la base de datos son las siguientes (excluye aquellas que no sean de interés para el análisis):

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) Se te recomienda seguir los siguientes pasos:

Prepara la base de datos Titanic:

Analiza los datos faltantes Realiza un análisis descriptivo 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.

```
M <- read.csv("C:\\Users\\saulv\\OneDrive\\Escritorio\\Septimo
semestre\\Titanic.csv")
str(M)
```

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

```
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Preparación de la base de datos

Ajustando las variables

```
# Eliminar variables:
M1 <- M[,c(-4,-9,-11)]

#Transformar a factores:
for(var in c('Survived','Pclass','Embarked','Sex'))
  M1[,var] <-as.factor(M1[,var])
```

Análisis de datos faltantes

```
V = matrix(NA,ncol=1,nrow=9)
for(i in c(1:9)){
  V[i,] <- sum(with(M1,M1[,i])=="")}
V
```

0
0
0
0
NA
0
0
NA
NA

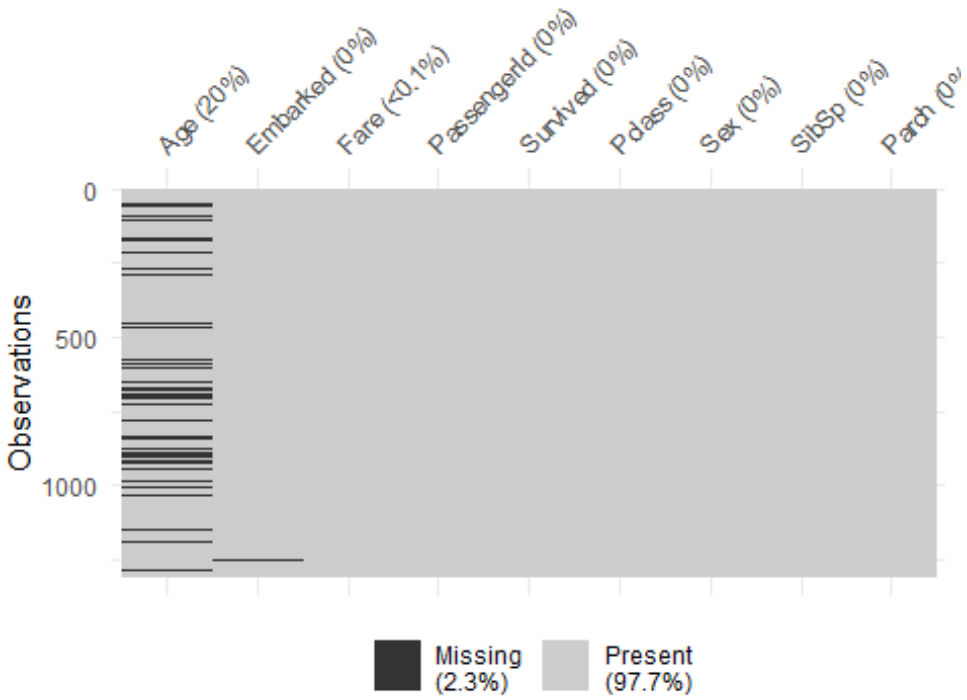
No hay variables que contengan espacios en blanco, pero hay vaeiables que contienen NA

```
N = apply(X=is.na(M1),MARGIN = 2,FUN = sum)
P = round(100*N/length(M1[,2]),2)
NP = data.frame(as.numeric(N),as.numeric(P))
row.names(NP)= c("PassengerId", "Survived", "Pclass", "Sex", "Age",
"SibSp", "Parch", "Fare", "Embarked")
names(NP)=c("Número","Porcentaje")
t(NP)
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked

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

```
vis_miss(M1,sort_miss = TRUE)
```



Se observa que las variables que Age, Embarked y Fare tienen datos con NA.

Ante esto se borraran los datos de las variables que contengan NA

```
## Sin borrar NA
summary(M1[,-1])
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0:815	1:323	female:466	Min. : 0.17	Min. :0.0000	Min. :0.000	Min. : 0.000	C :270
1:494	2:277	male :843	1st Qu.:21.00	1st Qu.:0.0000	1st Qu.:0.000	1st Qu.: 7.896	Q :123
NA	3:709	NA	Median :28.00	Median :0.0000	Median :0.000	Median : 14.454	S :914
NA	NA	NA	Mean	Mean	Mean	Mean :	NA's:

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
			:29.88	:0.4989	:0.385	33.295	2
NA	NA	NA	3rd Qu.:39.00	3rd Qu.:1.0000	3rd Qu.:0.000	3rd Qu.: 31.275	NA
NA	NA	NA	Max. :80.00	Max. :8.0000	Max. :9.000	Max. :512.329	NA
NA	NA	NA	NA's :263	NA	NA	NA's :1	NA

```
## Borrando NA
M2 = na.omit(M1)
summary(M2[,-1])
```

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

Ante esto se eliminaron 266 observacion y nos quedamos con un total de 1043 observaciones para el modelo, lo cual esta correcta ya que si podemos entrenar un modelo con estas obervacion que pueda dar una prediccion correcta.

Durante la eliminacion de los datos NA todas las variables se vieron afectadas.

##Sobrevivientes

```
t2c = 100*prop.table(table(M1[,2]))
t2s = 100*prop.table(table(M2[,2]))
t2p = c(t2s[1]/t2c[1],t2s[2]/t2c[2])
t2 = data.frame(as.numeric(t2c),as.numeric(t2s),as.numeric(t2p))
row.names(t2) = c("Murió","Sobrevivió")
names(t2) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t2,2)
```

	Con NA (%)	Sin NA (%)	Pérdida (prop)
Murió	62.26	60.21	0.97

	Con NA (%)	Sin NA (%)	Pérdida (prop)
Sobrevivió	37.74	39.79	1.05

Para la variable de la supervivencia del titanic con la eliminacion de los datos la probabilidad de sobrevivir aumento un 1.05 haciendo que sea más probable la supervivencia y bajando la probabilidad de muerte.

##Clase en que viajó

```
t3c = 100*prop.table(table(M1[,3]))
t3s = 100*prop.table(table(M2[,3]))
t3p = c(t3s[1]/t3c[1],t3s[2]/t3c[2],t3s[3]/t3c[3])
t3 = data.frame(as.numeric(t3c),as.numeric(t3s),as.numeric(t3p))
row.names(t3) = c("Primera","Segunda","Tercera")
names(t3) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t3,2)
```

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

Para la variable de clase en la que viajo se oberva que ahora estan más distribuidos, haciendo que los datos sean más equitativos entre las categorias de las clases permitiendo muestras más representativas para el modelo.

##Sexo

```
t4c = 100*prop.table(table(M1[,4]))
t4s = 100*prop.table(table(M2[,4]))
t4p = c(t4s[1]/t4c[1],t4s[2]/t4c[2])
t4 = data.frame(as.numeric(t4c),as.numeric(t4s),as.numeric(t4p))
row.names(t4) = c("Mujer","Hombre")
names(t4) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t4,2)
```

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

Al igual que la vriable de clases de viaje los datos ahora estan más distribuidos en el sexo permitiendo muestras más representativas para el modelo.

##Puerto de embarcación

```
t9c = 100*prop.table(table(M1[,9]))
t9s = 100*prop.table(table(M2[,9]))
t9p = c(t9s[1]/t9c[1],t9s[2]/t9c[2],t9s[3]/t9c[3])
t9 = data.frame(as.numeric(t9c),as.numeric(t9s),as.numeric(t9p))
```

```
row.names(t9) = c("Cherbourg","Queenstown","Southampton")
names(t9) = c("Con NA (%)","Sin NA (%)","Pérdida (prop)")
round(t9,2)
```

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

Para la variable del puerto de embarcacion tuvo un incremento una categoria haciendo que los datos no esten equitativos, generando un desequilibro en los datos que posiblemente en el entrenamiento del modelo de más peso a la categoria de Southampton.

Partición. Entrenamiento y prueba

Se toma el 70% de la muestra como entrenamiento y el 30% para prueba.

```
M_indice <- createDataPartition(M2$Survived, p = .7, list = FALSE, times
= 1)

M_train <- M2[ M_indice,]  %>% as_tibble()
M_valid <- M2[-M_indice,] %>% as_tibble()
```

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.

Entrenamiento

Primero se escojera el modelo con un mejor AIC

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

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

## Start:  AIC=563.8
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked
##
##               Df Deviance    AIC
## - Embarked     2   543.73 561.73
## - Parch        1   542.09 562.09
## - Fare         1   542.33 562.33
## - PassengerId  1   542.40 562.40
## <none>             541.80 563.80
```

```
## - SibSp        1   545.93 565.93
## - Age          1   557.69 577.69
## - Pclass       2   584.37 602.37
## - Sex          1   887.79 907.79
##
## Step:  AIC=561.73
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp + Parch +
##     Fare
##
##               Df Deviance    AIC
## - Fare         1   543.99 559.99
## - Parch        1   544.08 560.08
## - PassengerId  1   544.28 560.28
## <none>             543.73 561.73
## + Embarked     2   541.80 563.80
## - SibSp        1   548.37 564.37
## - Age          1   560.50 576.50
## - Pclass       2   590.43 604.43
## - Sex          1   892.96 908.96
##
## Step:  AIC=559.99
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp + Parch
##
##               Df Deviance    AIC
## - Parch        1   544.54 558.54
## - PassengerId  1   544.57 558.57
## <none>             543.99 559.99
## + Fare         1   543.73 561.73
## + Embarked     2   542.33 562.33
## - SibSp        1   548.98 562.98
## - Age          1   560.54 574.54
## - Pclass       2   607.28 619.28
## - Sex          1   893.71 907.71
##
## Step:  AIC=558.54
## Survived ~ PassengerId + Pclass + Sex + Age + SibSp
##
##               Df Deviance    AIC
## - PassengerId  1   545.13 557.13
## <none>             544.54 558.54
## + Parch        1   543.99 559.99
## + Fare         1   544.08 560.08
## + Embarked     2   542.89 560.89
## - SibSp        1   550.93 562.93
## - Age          1   560.98 572.98
## - Pclass       2   607.70 617.70
## - Sex          1   902.44 914.44
##
## Step:  AIC=557.13
## Survived ~ Pclass + Sex + Age + SibSp
```

```
##
##               Df Deviance    AIC
## <none>              545.13 557.13
## + PassengerId  1   544.54 558.54
## + Parch        1   544.57 558.57
## + Fare         1   544.63 558.63
## + Embarked     2   543.55 559.55
## - SibSp        1   551.37 561.37
## - Age          1   561.57 571.57
## - Pclass       2   607.81 615.81
## - Sex          1   903.53 913.53

##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family =
"binomial",
##     data = M_train)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age
SibSp
##     4.33136     -1.20770     -2.27814     -3.73318     -0.03392     -
0.34384
##
## Degrees of Freedom: 730 Total (i.e. Null);  725 Residual
## Null Deviance:       982.8
## Residual Deviance: 545.1     AIC: 557.1
```

- Identifica el mejor modelo de acuerdo con el AIC
- Selecciona la última variable que elminó el comando *step*. Prueba dos modelos, uno con esa variable y otro sin ella.

Modelo B

- Prueba el modelo incluyendo la última variable que eliminó el comando *step*.
- Indica cuáles son las variables que incluye.
- Interpreta la significancia global (de todo el modelo) y la individual (de cada una de las variables)

```
B = glm(formula = Survived ~ PassengerId + Pclass + Sex + Age + SibSp +
Parch +
    Fare, family = "binomial", data = M_train)
summary(B)

##
## Call:
## glm(formula = Survived ~ PassengerId + Pclass + Sex + Age + SibSp +
##     Parch + Fare, family = "binomial", data = M_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.6580437  0.5784875   8.052 8.14e-16 ***
```

```
## PassengerId -0.0002197  0.0002949  -0.745 0.456382
## Pclass2     -1.2935710  0.3510242  -3.685 0.000229 ***
## Pclass3     -2.3947531  0.3710903  -6.453 1.09e-10 ***
## Sexmale     -3.7758221  0.2527208 -14.941  < 2e-16 ***
## Age         -0.0344608  0.0086456  -3.986 6.72e-05 ***
## SibSp       -0.3115121  0.1477216  -2.109 0.034964 *
## Parch       -0.0764199  0.1290745  -0.592 0.553810
## Fare        -0.0013224  0.0026173  -0.505 0.613394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 982.80  on 730  degrees of freedom
## Residual deviance: 543.73  on 722  degrees of freedom
## AIC: 561.73
##
## Number of Fisher Scoring iterations: 5
```

Modelo C

- Prueba el modelo tal como te lo recomendó el comando *step*.
- Indica cuáles son las variables que incluye.
- Interpreta la significancia global (de todo el modelo) y la individual (de cada una de las variables)

```
C = glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare
, family = "binomial", data = M_train)
summary(C)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare, family = "binomial", data = M_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.518137   0.544271   8.301  < 2e-16 ***
## Pclass2     -1.302605   0.350705  -3.714 0.000204 ***
## Pclass3     -2.391491   0.371067  -6.445 1.16e-10 ***
## Sexmale     -3.776885   0.252636 -14.950  < 2e-16 ***
## Age         -0.034512   0.008652  -3.989 6.64e-05 ***
## SibSp       -0.306355   0.147323  -2.079 0.037573 *
## Parch       -0.076255   0.129171  -0.590 0.554961
## Fare        -0.001391   0.002612  -0.532 0.594445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 982.80  on 730  degrees of freedom
```

```
## Residual deviance: 544.28  on 723  degrees of freedom
## AIC: 560.28
##
## Number of Fisher Scoring iterations: 5
```

Analiza los modelos a través de:

Identificación de la Desviación residual de cada modelo Identificación de la Desviación nula Cálculo de la Desviación Explicada Prueba de la razón de verosimilitud Define cuál es el mejor modelo Escribe su ecuación, analiza sus coeficientes y detecta el efecto de cada predictor en la clasificación.

Análisis de los modelos B y C

Resumen de los indicadores importantes de los modelos B y C

Compara el AIC, la *Null Deviance* y la *Residual Deviance* de los modelos B y C. Extrae los valores con los modelos con los comandos:

- B$aic
- B$deviance
- B$null.deviance

Elabora una tabla comparativa

```
# Extraer indicadores
B_metrics <- c(AIC = B$aic, Null_Deviance = B$null.deviance,
Residual_Deviance = B$deviance)
C_metrics <- c(AIC = C$aic, Null_Deviance = C$null.deviance,
Residual_Deviance = C$deviance)

# Comparar en tabla
model_comparison <- rbind(B = B_metrics, C = C_metrics)
round(model_comparison, 2)
```

	AIC	Null_Deviance	Residual_Deviance
B	561.73	982.8	543.73
C	560.28	982.8	544.28

###Cálculo de la Desviación Explicada

```
# Calcular la Desviación Explicada para el Modelo B
null_deviance_B <- B$null.deviance
residual_deviance_B <- B$deviance
desviacion_explicada_B <- (1 - (residual_deviance_B / null_deviance_B)) *
100

# Calcular la Desviación Explicada para el Modelo C
null_deviance_C <- C$null.deviance
residual_deviance_C <- C$deviance
```

```r
desviacion_explicada_C <- (1 - (residual_deviance_C / null_deviance_C)) *
100

# Imprimir los resultados
cat("Desviación Explicada del Modelo B: ", round(desviacion_explicada_B,
2), "%\n")
```

```
## Desviación Explicada del Modelo B:  44.68 %
```

```r
cat("Desviación Explicada del Modelo C: ", round(desviacion_explicada_C,
2), "%\n")
```

```
## Desviación Explicada del Modelo C:  44.62 %
```

Prueba de razón de verosimilitud

H_0: El modelo con predictores explica mejor la variable respuesta: $log\left(\frac{p}{1-p}\right)$ que el modelo nulo

H_1: El modelo nulo explica mejor la variable respuesta: $log\left(\frac{p}{1-p}\right)$ (la probabilidad es constante)

Se calcula el estadístico de χ^2 para la razón de verosimilutud a partir de las *Deviance* de los modelos.

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

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

```
## numeric(0)
```

Interpreta en el contexto del problema

Comparación entre los modelos B y C

Se pueden comparar los modelo B y C para ver si hay una diferencia significativa entre ambos con la misma razón de verosimilitud utilizando el comando ANOVA y la prueba LR.

```r
library(car)
```

```
## Cargando paquete requerido: carData
```

```
##
## Adjuntando el paquete: 'car'
```

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

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

anova(B,C,test="LR")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
722	543.7283	NA	NA	NA
723	544.2836	-1	-0.5552716	0.4561717

Modelo Seleccionado

Define los coeficientes del modelo seleccionado.

```
coefficientes = C$coefficients
coefficientes

##  (Intercept)      Pclass2      Pclass3      Sexmale          Age
SibSp
##  4.518137172 -1.302605101 -2.391491110 -3.776885171 -0.034511969 -
0.306355207
##        Parch         Fare
## -0.076255023 -0.001390676
```

El modelo que yo escoji es el modelo C osea el modelo 2

##Analiza las predicciones para los datos de entrenamiento Elabora la matriz de confusión Elabora la Curva ROC Elabora el gráfico de violín Concluye sobre el modelo basándote en las predicciones de los datos de entrenamiento.

Gráfica el modelo

Para percibir el efecto de cada variable, grafica cada variable contra los valores predichos por el modelo. Aunque en el modelo, la variable respuesta es:

$$\hat{y} = log\left(\frac{p}{1-p}\right)$$

con el subcomando: *fitted.values* del comando *glm* se obtienen las probabilidades estimadas para los valores datos. R despeja las probabilidades:

$$\hat{p} = \left(\frac{e^{\hat{y}}}{1+e^{\hat{y}}}\right)$$

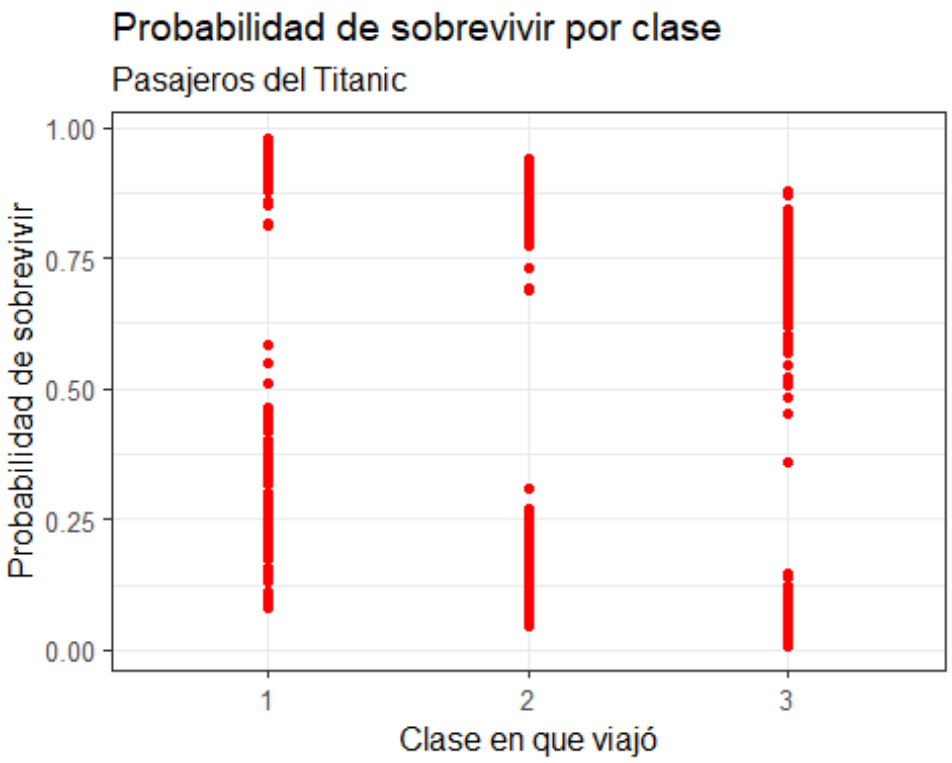
Así que interpretar el efecto de cada variable, se grafica cada una de ellas contra los valores predichos para la probabilidad de sobrevivencia.

Para hacer los gráficos se ejemplifica con:

Clase en que viajó el pasajero

```
p_pred = C$fitted.values
M_pred = data.frame(M_train[,c(2,3,4,5,6)],p_pred)

ggplot(M_pred, aes( x = Pclass)) +
geom_point(aes(y=M_pred$p_pred), size=1.5,color="red") +
  labs(x="Clase en que viajó", y="Probabilidad de sobrevivir",
       title="Probabilidad de sobrevivir por clase",
       subtitle="Pasajeros del Titanic",
       col="")+
  theme_bw(base_size = 12)
```

```
## Warning: Use of `M_pred$p_pred` is discouraged.
## i Use `p_pred` instead.
```



Grafica y concluye cómo cambia la probabilidad predicha con cada variable que resultó significativa

Predicciones

Se hace el análisis con el modelo seleccionado

Matriz de confusión

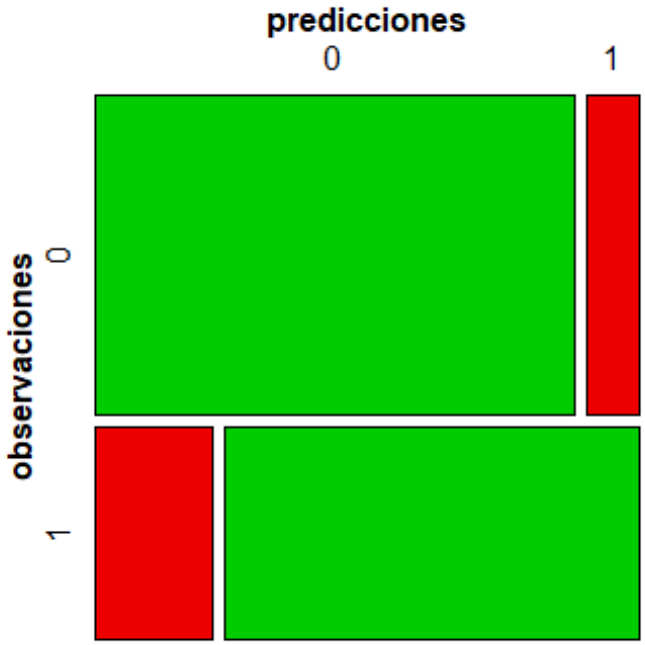
```
library(vcd)
```

```
## Cargando paquete requerido: grid
```

```
##
## Adjuntando el paquete: 'vcd'

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

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

observaciones/predicciones	0	1
0	396	44
1	64	227

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




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

## La Exactitud (accuracy) del modelo es 0.8522572

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

```
## La Sensibilidad del modelo es 0.9

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

## La Especificidad del modelo es 0.7800687

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

## La Precisión del modelo es 0.8608696
```

Define si el modelo es bueno o no.

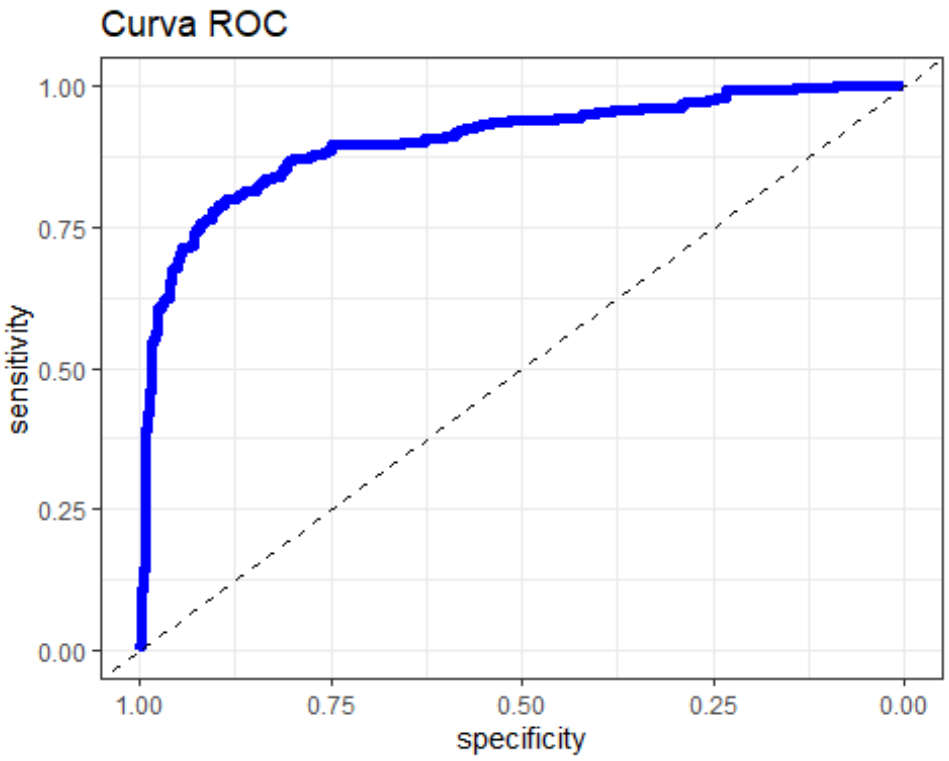
El modelo 2 es bueno ya que este tiene desde inicio una precision de 0.86 haciendo que este modelo sea confiable

Curva ROC

Para hacer la curva, es necesario crear las predicciones para el data set de entrenamiento. El comando *roc* cálculará la sensibilidad y la especificidad para los datos obtenidos.

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

library(pROC)

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

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

##
## Adjuntando el paquete: 'pROC'

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

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

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

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

## Setting direction: controls < cases

ROC

##
## Call:
## roc.default(response = M_train$Survived, predictor = pred)
```

```
##
## Data: pred in 440 controls (M_train$Survived 0) < 291 cases
(M_train$Survived 1).
## Area under the curve: 0.9033

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



Nota: Se grafica Especificidad, pero en realidad se está graficando 1 - Especificidad.

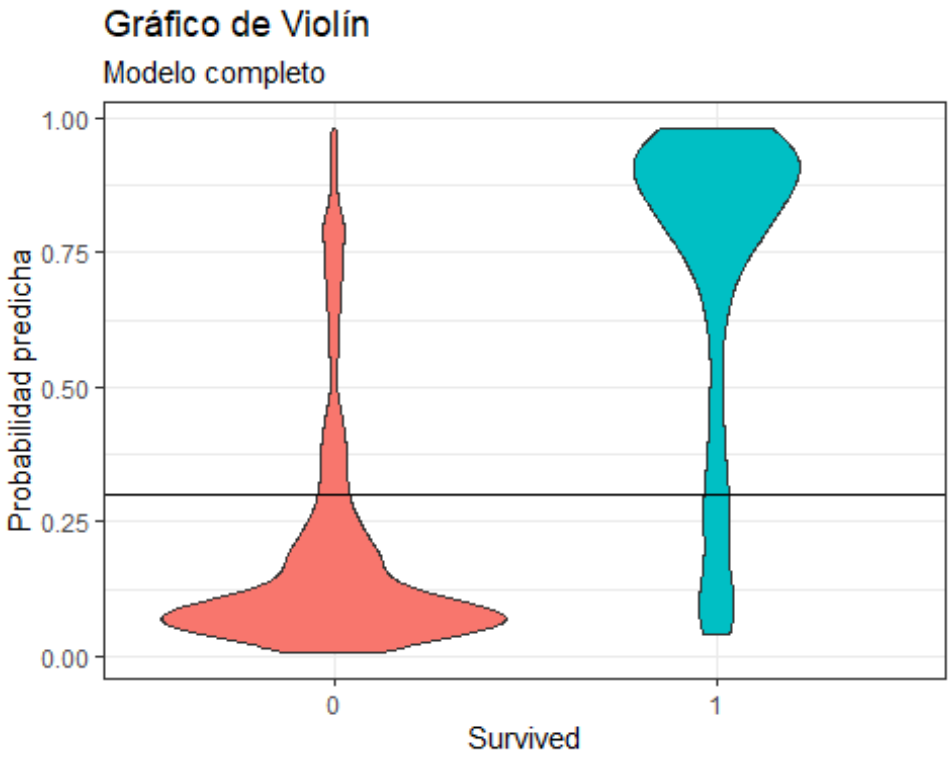
Interpreta el gráfico y la salida que da el comando *roc*

Gráfico de violín

Se crea la base de datos para el gráfico, se usan las predicciones ya elaboradas para el gráfico ROC y las clasificaciones originales (*train$M_Survived*).

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

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

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



Validación del modelo con la base de datos de validación

Elije un umbral de clasificación óptimo Elabora la matriz de confusión con el umbral de clasificación óptimo

Validación

Elección de un umbral de clasificación optimo.

Elección del umbral de clasificación (punto de corte)

Se trabaja con la base de datos de validación (*M_valid*) y se realiza el gráfico de la Exactitud, Sensibilidad, Especificidad y Precisión para distintos valores del umbral de clasificación. Se siguen los siguientes pasos:

1. Predicción en los datos de validación con el modelo elegido (en el ejemplo, el B)

2. Se definen los umbrales de clasificación: irán desde 0.05 hasta 0.95.
3. Se definen las métricas de la matriz de confusión para cada umbral de clasificación
4. Se prepara el conjunto de datos: se quitan los NA y se agrega la columna de umbrales de clasificaición
5. Se le da un formato a la base de datos para que pueda ser graficada más fácilmente.

Generación de base de datos para graficar

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

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

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

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

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

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

Formato de datos

- Se crea la variable *métrica* que será una variable categórica para las métricas (Exactitud, Sensibilidad, Especificidad y Precisión)
- Los valores de las métricas se ponen en una sola columna.

- Se identifican las métricas para los distintos umbrales con la variable 'umbral'.

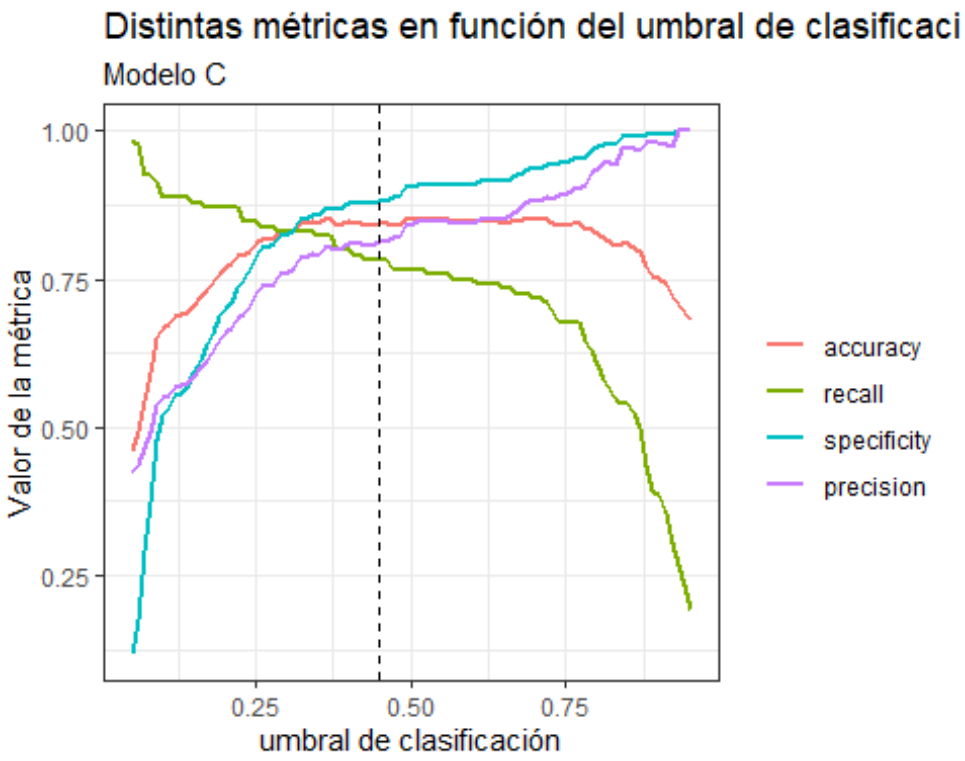
```
library(reshape2)

##
## Adjuntando el paquete: 'reshape2'
```

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

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

Gráfica

En la gráfica se define cuál es el mejor umbral de clasificación dependiendo de cuál métrica es más importante en el contexto del problema (Exactitud, Sensibilidad, Especificidad o Precisión). Si no hay una métrica de preferencia, se opta por escoger el máximo valor de que pueden tener estás métricas en conjunto. En cualquier caso da valores a u para mover el umbral de clasificación y observar como se comporta con respecto a las métricas.

```
library(ggplot2)

u = 0.45 #Se dio un valor arbitrario, tú modificalo de acuerdo al
criterio que selecciones.

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

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



Define cuál es el mejor umbral en donde se obtienen las mejores métricas Recall, Accuracy, Sensitivity y Specificity.

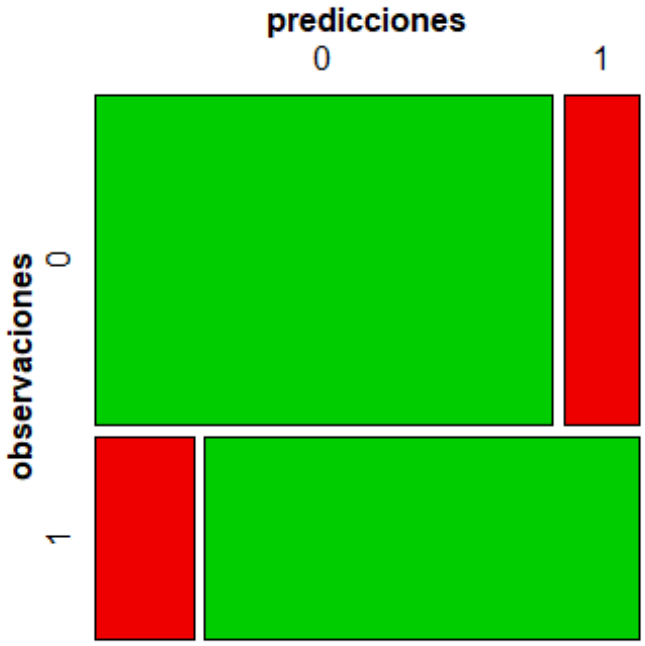
Matriz de confusión con el umbral de clasificación optimo

De acuerdo al umbral seleccionado, calcula la matriz de confusión y las métricas obtenidas. Indica si mejora la predicción con respecto al umbral de u = 0.5, que es el que se maneja por default.

```
prediccionesV = ifelse(pred_val > 0.45, yes = 1, no = 0)
M_Cv <- table(prediccionesV, M_valid$Survived, dnn = c("observaciones",
"predicciones"))
M_Cv
```

observaciones/predicciones	0	1
0	166	27
1	22	97

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




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

## La Exactitud (accuracy) del modelo es 0.8429487

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

## La Sensibilidad del modelo es 0.8601036

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

## La Especificidad del modelo es 0.8151261

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

## La Precisión del modelo es 0.8829787
```

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? Interpreta los coeficientes del modelo Define cuál es el mejor umbral de clasificación y por qué

Las caracteristicas principales que influyen en el modelo son:

Sex: Las mujeres tienen más probabilidades de sobrevivir en el Titanic, lo que podría reflejarse en un coeficiente positivo para la categoría femenina.

Pclass: Las personas en las primera clase tienen más probabilidades de sobrevivir debido a las mejores condiciones de evacuación. Pclass 2 y 3, por otro lado, debido a que estas clases no tenian condiciones de evacuacion están asociadas a una menor probabilidad de supervivencia.

Age: Los niños y los jóvenes podrían haber tenido más probabilidades de sobrevivir debido a siempre poner a los niños primero sobre todo para su rescate.

SibSp y Parch: Tener familiares a bordo podría tener un efecto positivo o negativo en la supervivencia. Es posible que aquellos con pocos o ningún familiar a bordo tuvieran una mayor probabilidad de sobrevivir, mientras que aquellos con familiares a bordo podrían haber estado en un grupo más grande que dificultaba la evacuación.

Ticket y Embarked: Los pasajeros con boletos de primera clase o aquellos que embarcaron en puertos como Cherburgo tenían más probabilidades de sobrevivir.

Para los coeficientes como hay valores altos en estos hay una significacia entre estos lo que hace que el modelo tenga una mejor prediccion.

Para mi modelo el mejor umbarl de clasificacion fue de 0.45 ya que este aumenta la acuraccy y la presicion del modelo haciendo que las prredicciones sean más acertadas