Modelos de Clasificación

2025-05-13

Javier Ramirez Cervantes

Se cargan los datos para realizar el modelo logit y las librerias dplyr y ggplot2 para trabajar el modelo:

```
library(dplyr)
```

```
##
## Adjuntando el paquete: '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)

# Se crea el DataFrame con todas las observaciones
edad_x1= c(23, 45, 36, 50, 29, 31, 38, 27, 42, 34, 48, 25, 46, 26, 28,
          41, 33, 44, 30, 47, 32, 39, 26, 43, 35, 49, 24, 37, 29, 51)

ingreso_anual_x2= c(35000, 65000, 50000, 80000, 40000, 45000, 55000, 38000, 75000, 52000, 68000, 36000,


## Clase, donde 1 es que si realiza una compra, con respecto a las otras variables, y 0 que no lo hace:

clase_y= c(0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0,
          1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1)


df = data.frame( edad_x1 = edad_x1 ,
                  ingreso_anual_x2 = ingreso_anual_x2,
                  clase_y = clase_y)


head(df, 3)
```

```
##   edad_x1 ingreso_anual_x2 clase_y
## 1      23            35000       0
## 2      45            65000       1
## 3      36            50000       0
```

Paso 1: Analisis descriptivo de las variables

- Variable de Clase Y

Desbalanceo agrupando por Clase y haciendo el conteo de casos que caen en una categoria o en otra.

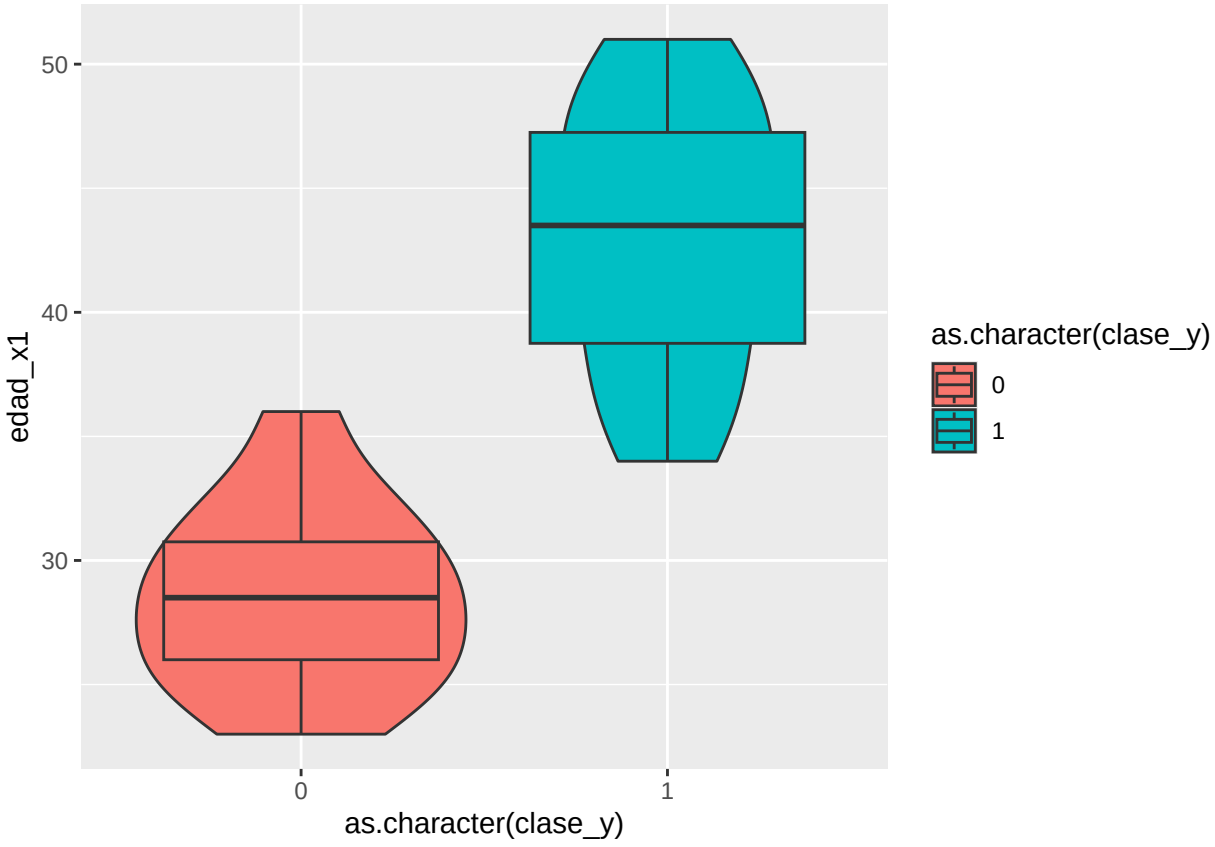
```
df %>%
  group_by(clase_y) %>%
  summarise(Numero = n())
```

```
## # A tibble: 2 x 2
##   clase_y Numero
##     <dbl>  <int>
## 1       0     14
## 2       1     16
```

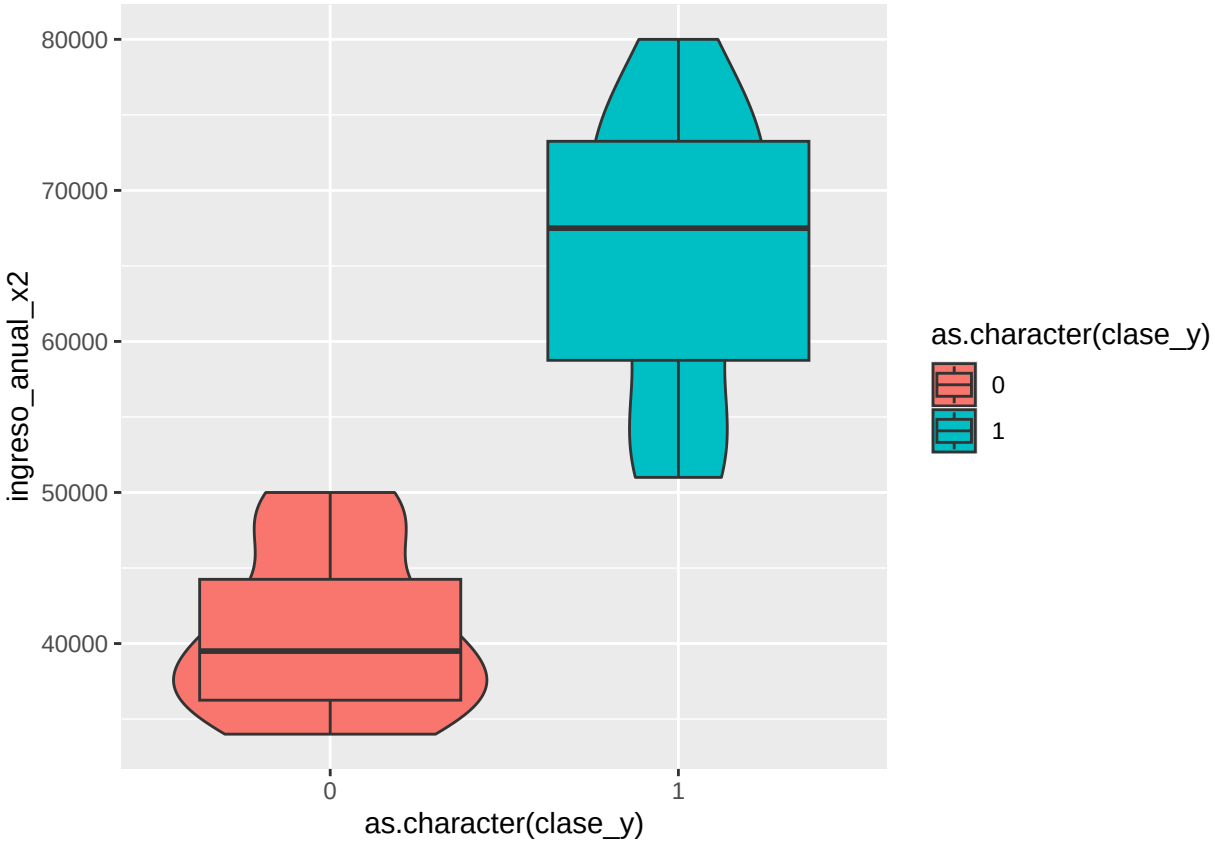
Los resultados muestran que de un total de 30 datos, 16 son de clase uno, y 14 de clase 0, por lo cual se observa que hay mas datos de personas que si realizaron la compra.

Graficos de violin

```
df %>%
  ggplot(aes(x=as.character(clase_y),y=edad_x1, fill=as.character(clase_y)))+
  geom_violin()+
  geom_boxplot()
```



```
df %>%
  ggplot(aes(x=as.character(clase_y),y=ingreso_anual_x2, fill=as.character(clase_y)))+
  geom_violin()+
  geom_boxplot()
```



En estos graficos se puede observar, que cuando la edad es mas proxima a los 44 años, una persona es mas propensa a realizar una compra, mientras que si es menor de los 30, con una edad aproximada a los 29, la persona no realiza la compra

En cuanto al segundo grafico, la compra se realiza con un ingreso anual aproximado a los 69,000, mientras que con un ingreso anual de 40,000, no se realiza la compra.

- Variable de Edad X1

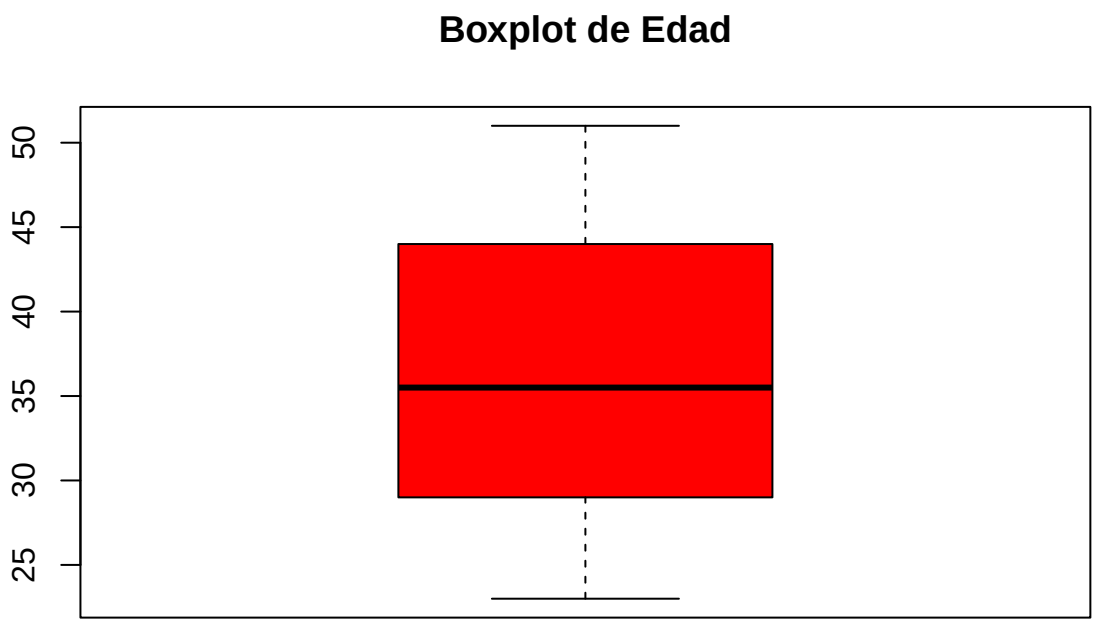
```
ed <- df %>%
  summarise(Media = mean(edad_x1),
            Mediana = median(edad_x1),
            Desviacion = sd(edad_x1) )

Rango = range(edad_x1)

ed
```

```
##      Media Mediana Desviacion
## 1 36.26667    35.5   8.689801
```

```
Rango
```

```
## [1] 23 51
```

```
boxplot(edad_x1, main="Boxplot de Edad", col = "red")
```



En los resultados se muestra que la media esta cerca de la mediana aunque con un ligero sesgo a la derecha, esto se comprueba con el grafico de caja. En cuanto al valor de la desviacion estandar es alto con respecto al rango de los datos, que va de 23 a 51.

- Variable de Ingreso anual X2

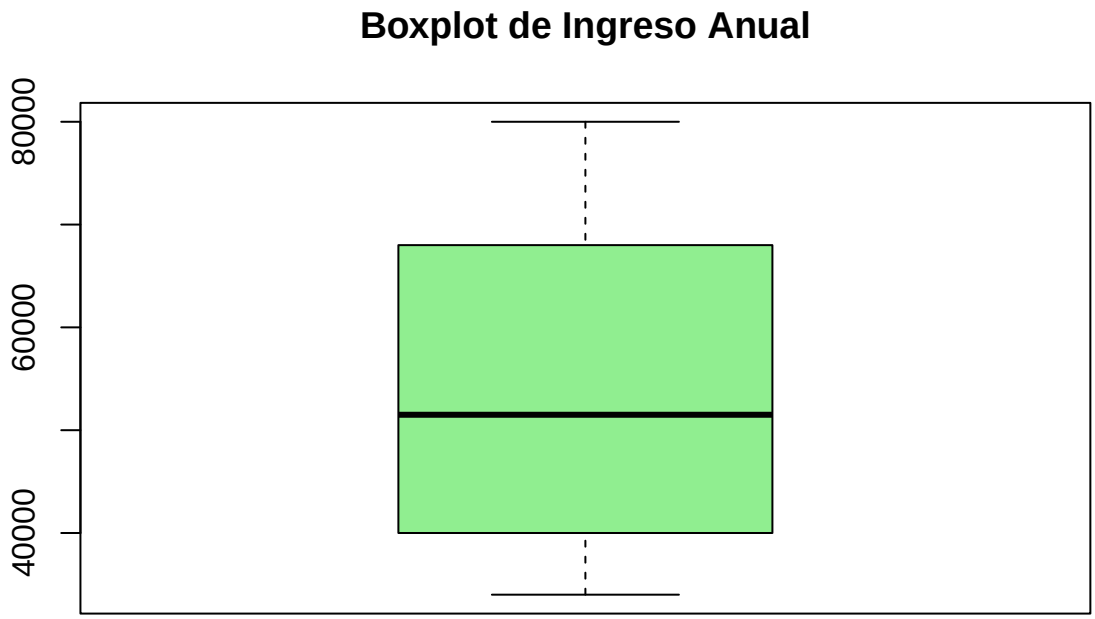
```
ing <- df %>%
  summarise(Media = mean(ingreso_anual_x2),
            Mediana = median(ingreso_anual_x2),
            Desviacion = sd(ingreso_anual_x2) )

Rango = range(ingreso_anual_x2)

ing
```

```
##   Media Mediana Desviacion
## 1 54200   51500   14898.28
```

```
Rango
```

```
## [1] 34000 80000
```

```
boxplot(ingreso_anual_x2, main="Boxplot de Ingreso Anual", col = "lightgreen")
```



En los resultados se muestra que la media esta cerca de la mediana aunque con sesgo a la derecha, esto se comprueba con el grafico de caja. En cuanto al valor de la desviacion estandar es alto con respecto al rango de los datos, que va de 34,000 a 80,000.

Paso 2: Construccion del modelo de clasificacion

Estimacion del modelo de regresion logistica:

```
x_train = df[,c('edad_x1', 'ingreso_anual_x2')]
y_train = df[,'clase_y']
modelo = glm(y_train ~ x_train$edad_x1 + x_train$ingreso_anual_x2, family ="binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(modelo)
```

```
##
## Call:
## glm(formula = y_train ~ x_train$edad_x1 + x_train$ingreso_anual_x2,
##     family = "binomial")
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.152e+03  4.423e+05  -0.003    0.998
## x_train$edad_x1          -1.028e+01  1.072e+04  -0.001    0.999
## x_train$ingreso_anual_x2  3.003e-02  1.305e+01   0.002    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.1455e+01  on 29  degrees of freedom
## Residual deviance: 1.2741e-08  on 27  degrees of freedom
## AIC: 6
##
## Number of Fisher Scoring iterations: 25
```

Hipotesis nula: H0 -> El coeficiente no es significativo
Hipotesis alternativa: Ha -> El coeficiente es significativo

Reglas de decision: p-value > 0.05 No se rechaza H0 // p-value < 0.05 Se rechaza H0

Segun los resultados de los p-values de cada uno de los coeficientes, los resultados sugieren que los coeficientes no son significativos

Se calcula el valor de la pseudo r2 ajustada pseudo R^2 ajustada = 1- ((residual desviance - k)/ (null desviance))

```r
pseudo_r2_adj = 1 - ((modelo$deviance - 3)/(modelo$null.deviance))
pseudo_r2_adj
```

```
## [1] 1.072367
```

De acuerdo con los resultados, el valor de la pseudo r^2 es mayor a uno, lo cual indica un problema de sobreajuste del modelo

Paso 3: Calculo de Probabilidades

Prediccion de las probabilidades del modelo, para comparar resultados:

```r
predicciones = predict(modelo, data.frame( newdata = x_train),
                       type ='response')
head(predicciones, 3)
```

```
##            1            2            3
## 2.220446e-16 1.000000e+00 9.475280e-10
```

Se convierten las probabilidades estimadas del modelo en categorias para especificar los cambios de categoria acorde a las probabilidades que se calcularon, usando un threshold igual a 0.5:

```
clasificacion = ifelse(predicciones < 0.5,0,1)
clasificacion
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  0  1  0  1  0  0  1  0  1  1  1  0  1  0  0  1  0  1  0  1  0  1  0  1  1  1
## 27 28 29 30
##  0  1  0  1
```

Se observa el balanceo de la clasificacion

```
table(clasificacion)
```

```
## clasificacion
##  0  1
## 14 16
```

Se mantienen los mismos resultados que el balanceo original

Paso 4: Evaluación de la Matriz de Confusión y Métricas de Desempeño

Se construye la Matriz de confusion:

```
confusion = table(clasificacion, df$clase_y)
confusion
```

```
##
## clasificacion  0  1
##             0 14  0
##             1  0 16
```

Esta es la matriz de confusion, en la cual se observa que los resultados entre la clasificacion estimada por el modelo y los originales son identicos puesto que no hay ningun valor en false negative o en false positive.

Se calculan las metricas

```
# TP = true positive/ FP = false positive/ FN = false negative
#precision = TP / (TP + FP)
precision <- 16 / (16 + 0)
print(paste("precision es igual a", precision))
```

```
## [1] "precision es igual a 1"
```

```
#recall / sensibilidad = TP / (TP + FN)
recall <- 16 / (16 + 0)
print(paste("recall es igual a", recall))
```

```
## [1] "recall es igual a 1"
```

```r
# f1 score = 2* ( (precision*recall)/(precision + recall) )
f1_score <- 2 * ( (precision*recall)/(precision + recall) )
print(paste("f1 score es igual a", f1_score))
```

```
## [1] "f1 score es igual a 1"
```

Estos resultados son perfectos debidos al problema del sobreajuste del modelo, el cual se observa desde el valor de la Pseudo R2 ajustada y se confirma con la matriz de confusion

Paso 5: Impacto del Umbral de Clasificación

Se genera la simulacion de diferentes valores del threshold para evaluar cual es el valor optimo que permite obtener el mejor f1 score:

```r
model <- glm(clase_y ~ edad_x1 + ingreso_anual_x2, family ="binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
y_prob <- predict(model, df, type='response')
thresholds <- seq(0, 1, length.out=100)
precisions <- vector()
recalls <- vector()
f1_scores <- vector()

for (threshold in thresholds) {
  y_pred <- ifelse(y_prob >= threshold, 1, 0)
  precision <- sum(y_pred[clase_y==1]==1)/ sum(y_pred==1)
  recall <- sum(y_pred[clase_y==1]==1)/ sum(clase_y==1)
  f1 <- 2 * ((precision*recall)/(precision+recall))

  precisions <- c(precisions, precision)
  recalls <- c(recalls, recall)
  f1_scores <- c(f1_scores, f1)
}

max_f1_index <- which.max(f1_scores)
optimal_threshold <- thresholds[max_f1_index]
print(paste("el valor optimo del threshold es", optimal_threshold))
```

```
## [1] "el valor optimo del threshold es 0.0101010101010101"
```

```r
optimal_precision <- precisions[max_f1_index]
print(paste("El valor optimo de la precision es", optimal_precision))
```

```
## [1] "El valor optimo de la precision es 1"
```

```r
optimal_recalls <- recalls[max_f1_index]
print(paste( "El valor optimo del Recall es", optimal_recalls))
```

```
## [1] "El valor optimo del Recall es 1"
```

```r
optimal_f1_scores <- f1_scores[max_f1_index]
print(paste("el valor optimo del f1 score es", optimal_f1_scores))
```

```
## [1] "el valor optimo del f1 score es 1"
```

De acuerdo con los resultados anteriores, el valor del threshold que optimiza la capacidad predictiva del modelo para clasificar es 0.0101010101010101, threshold con el cual los valores de la precision, el recall y el f1 score son iguales a uno. Sin embargo, tomando en cuenta los resultados previos del modelo y sus problemas de sobrejauste, los resultados no son fiables debido a dichos problemas.

En un modelo normal, la relevancia de definir un threshold adecuado puede ayudar a ajustar el modelo para obtener mejores resultados puesto que un threshold demasiado alto o bajo, puede cambiar completamente la forma en la que clasifica el modelo, lo cual se puede observar en los valores de la precision, recall y, principalmente en el del f1 score. En caso de que se haga una mala clasificacion, el modelo no es muy eficiente debido a que no cumple con su objetivo, puesto que podria clasificar en una categoria cuando en realidad corresponde a otra. Por ejemplo, en un modelo de clasificacion de riesgo creditico, encontrar un threshold optimo supone fijar el umbral correcto para otorgar credito a los clientes que tengan baja probabilidad de incumplimiento de pago y evitar otorgar creditos a clientes con alta probabilidad de incumplimiento de pago.