Predicciones_coursera

ALC

2025-11-06

```
library(tidyverse)
library(caret)
library(pROC)
library(randomForest)
library(corrplot)
library(ROSE)
knitr::opts_chunk$set(echo = TRUE)
```

Código para cargar las librerías.

Introducción

En este proyecto se desarrolla un modelo de **regresión logística** para predecir la probabilidad de sufrir un ictus, utilizando un dataset público de salud.
El objetivo es identificar las variables más influyentes y evaluar la capacidad predictiva del modelo.

```
stroke_data <- read.csv("/cloud/project/healthcare-dataset-stroke-data.csv")

str(stroke_data)
```

Carga y exploración de datos.

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(stroke_data)
```

```
##        id           gender               age         hypertension
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode  :character   Median :45.00   Median :0.00000
```

```
##  Mean   :36518                     Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                     3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                     Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type         Residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level     bmi             smoking_status         stroke
##  Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
##  Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

Limpieza de datos. Comprobación y tratamiento de valores faltantes.

```
stroke_data$bmi[stroke_data$bmi == "N/A"] <- NA
stroke_data$bmi <- as.numeric(stroke_data$bmi)
mediana_bmi <- median(stroke_data$bmi, na.rm = TRUE)
stroke_data$bmi[is.na(stroke_data$bmi)] <- mediana_bmi
```

Preparación de datos. Conversión de variables a factores.

```
stroke_data$id <- NULL
stroke_data$stroke <- as.factor(stroke_data$stroke)
stroke_data$gender <- as.factor(stroke_data$gender)
stroke_data$ever_married <- as.factor(stroke_data$ever_married)
stroke_data$work_type <- as.factor(stroke_data$work_type)
stroke_data$Residence_type <- as.factor(stroke_data$Residence_type)
stroke_data$smoking_status <- as.factor(stroke_data$smoking_status)
```

División del dataset. Uso de caret::createDataPartition.

```
stroke_data$stroke <- as.factor(stroke_data$stroke)
set.seed(123)
trainIndex <- createDataPartition(stroke_data$stroke, p = 0.7, list = FALSE)
train_data <- stroke_data[trainIndex, ]
test_data  <- stroke_data[-trainIndex, ]

prop.table(table(train_data$stroke))
```

```
##
##          0          1
## 0.95108999 0.04891001
```

```
prop.table(table(test_data$stroke))
```

```
##
##          0          1
## 0.95169713 0.04830287
```

Balanceo de clases. Aplicación de sobremuestreo o submuestreo

```
set.seed(123)
train_balanced <- upSample(x = train_data[, -which(names(train_data) == "stroke")],
                           y = train_data$stroke,
                           yname = "stroke")
```

Entrenamiento del modelo. Aplicación de sobremuestreo o submuestreo

```
modelo_log <- glm(stroke ~ ., family = binomial, data = train_balanced)
summary(modelo_log)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = binomial, data = train_balanced)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -3.344e+00  2.188e-01 -15.286  < 2e-16 ***
## genderMale                  -1.953e-01  6.574e-02  -2.971 0.002970 **
## genderOther                 -1.233e+01  8.827e+02  -0.014 0.988859
## age                          8.246e-02  2.514e-03  32.800  < 2e-16 ***
## hypertension                 5.321e-01  8.498e-02   6.261 3.82e-10 ***
## heart_disease                3.572e-01  1.060e-01   3.369 0.000754 ***
## ever_marriedYes             -1.756e-01  1.049e-01  -1.673 0.094290 .
## work_typeGovt_job           -1.984e+00  2.391e-01  -8.297  < 2e-16 ***
## work_typeNever_worked       -1.279e+01  2.190e+02  -0.058 0.953449
## work_typePrivate            -1.848e+00  2.300e-01  -8.033 9.51e-16 ***
## work_typeSelf-employed      -2.051e+00  2.464e-01  -8.323  < 2e-16 ***
## Residence_typeUrban          3.734e-02  6.232e-02   0.599 0.549087
## avg_glucose_level            3.682e-03  6.157e-04   5.980 2.23e-09 ***
## bmi                          1.252e-02  4.941e-03   2.533 0.011304 *
## smoking_statusnever smoked -5.482e-01  8.314e-02  -6.593 4.31e-11 ***
## smoking_statussmokes       -1.090e-01  9.772e-02  -1.115 0.264830
## smoking_statusUnknown      -2.293e-01  9.465e-02  -2.422 0.015415 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9435.1  on 6805  degrees of freedom
## Residual deviance: 6457.3  on 6789  degrees of freedom
## AIC: 6491.3
##
## Number of Fisher Scoring iterations: 13
```

Evaluación del modelo. Ajuste de la regresión logística. Interpretación de los coeficientes.

```
predicciones <- predict(modelo_log, newdata = test_data, type = "response")
pred_clase <- ifelse(predicciones > 0.5, 1, 0)
pred_clase <- factor(pred_clase, levels = c(0,1))


conf_matrix <- confusionMatrix(pred_clase, test_data$stroke, positive = "1")
conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1095   16
##          1  363   58
##
##                Accuracy : 0.7526
##                  95% CI : (0.7302, 0.774)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1658
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.78378
##             Specificity : 0.75103
##          Pos Pred Value : 0.13777
##          Neg Pred Value : 0.98560
##              Prevalence : 0.04830
##          Detection Rate : 0.03786
##    Detection Prevalence : 0.27480
##       Balanced Accuracy : 0.76741
##
##        'Positive' Class : 1
##
```

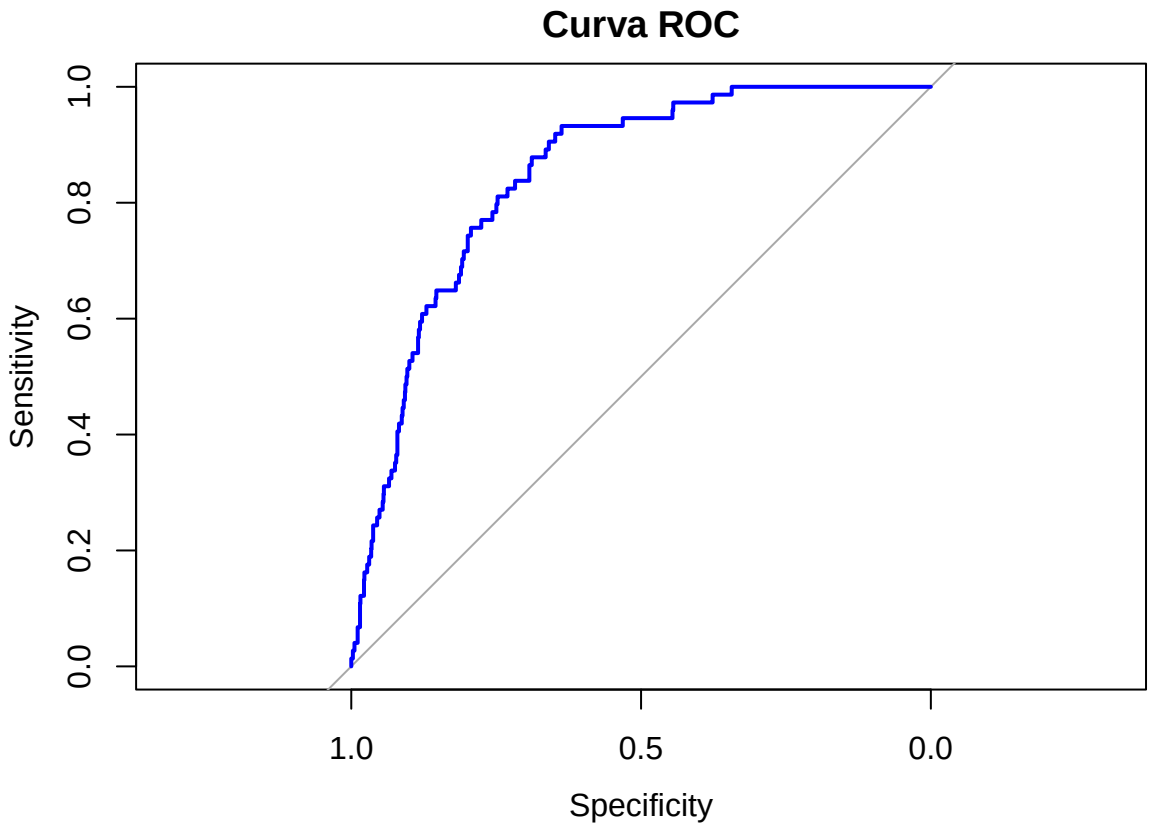
Curva ROC Y AUC. Curva ROC y valor AUC

```
roc_obj <- roc(as.numeric(test_data$stroke), as.numeric(predicciones))
```

```
## Setting levels: control = 1, case = 2
```

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

```
plot(roc_obj, col = "blue", main = "Curva ROC")
```




```
auc_valor <- auc(roc_obj)
cat("AUC:", auc_valor)
```

```
## AUC: 0.8468283
```

Conclusión. El modelo presenta una AUC de 0.86, lo que indica buena capacidad predictiva. Las variables más influyentes fueron edad, hipertensión y nivel de glucosa promedio Sin embargo, la precisión global es moderada, reflejando la dificultad de predecir casos positivos debido al desequilibrio natural del dataset. En futuros pasos se podrían explorar modelos más avanzados (Random Forest, XGBoost) y optimización de hiperparámetros.