Regresión Logística cpn datos de matematicas y aprobado

Rubén Pizarro Gurrola

2022-03-26

Objetivo

Realizar y evaluar predicciones con un modelo de clasificación Regresión Logística.

Descripción

Se va a construir un modelo de regresión logística con dos variables, valor numérico de calificación de una asignatura de matemáticas entre 0 y 100 y un valor categórico de 0 y 1 que significa estado Aprobado o No aprobado.

Sustento teórico

Pendiente

El valor de AIC es una medida de calidad del modelo y tienen que ver con ajuste de los datos y las predicciones. Este valor es comparado contra si mismo es decir si se tiene establecido inicialmente o puede compararse contra otro modelo.

Se usará la función logit para probabilidades.

$$P(y = 1|x) = \Lambda(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)$$

$$\Lambda(z) = \frac{exp(z)}{1 + exp(z)}$$

Cargar librerías

```
library(ggplot2)
library(caret) # Partir datos como matriz de confusión
```

```
## Loading required package: lattice
```

Cargar o construir datos

```
# Variable dependiente LOGICA 0 O 1
estado <- as.factor(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
                      0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
                      0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
```

```
                        0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
                        1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
                        1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
                        1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
                        0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
                        0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
                        0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
                        0, 0, 0, 0, 1, 0, 0, 0, 1, 1))

# Variable independiente
matematicas <- c(41, 53, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50, 43,
                 51, 60, 62, 57, 35, 75, 45, 57, 45, 46, 66, 57, 49, 49, 57, 64,
                 63, 57, 50, 58, 75, 68, 44, 40, 41, 62, 57, 43, 48, 63, 39, 70,
                 63, 59, 61, 38, 61, 49, 73, 44, 42, 39, 55, 52, 45, 61, 39, 41,
                 50, 40, 60, 47, 59, 49, 46, 58, 71, 58, 46, 43, 54, 56, 46, 54,
                 57, 54, 71, 48, 40, 64, 51, 39, 40, 61, 66, 49, 65, 52, 46, 61,
                 72, 71, 40, 69, 64, 56, 49, 54, 53, 66, 67, 40, 46, 69, 40, 41,
                 57, 58, 57, 37, 55, 62, 64, 40, 50, 46, 53, 52, 45, 56, 45, 54,
                 56, 41, 54, 72, 56, 47, 49, 60, 54, 55, 33, 49, 43, 50, 52, 48,
                 58, 43, 41, 43, 46, 44, 43, 61, 40, 49, 56, 61, 50, 51, 42, 67,
                 53, 50, 51, 72, 48, 40, 53, 39, 63, 51, 45, 39, 42, 62, 44, 65,
                 63, 54, 45, 60, 49, 48, 57, 55, 66, 64, 55, 42, 56, 53, 41, 42,
                 53, 42, 60, 52, 38, 57, 58, 65)

datos <- data.frame(estado, matematicas)
head(datos, 10)
```

```
##    estado matematicas
## 1       0          41
## 2       0          53
## 3       0          54
## 4       0          47
## 5       0          57
## 6       0          51
## 7       0          42
## 8       0          45
## 9       0          54
## 10      0          52
```

Datos de entrenamiento

No se parten los datos de entrenamiento ni datos de validación, se utilizarán todos los datos

Cuantas de cada clase

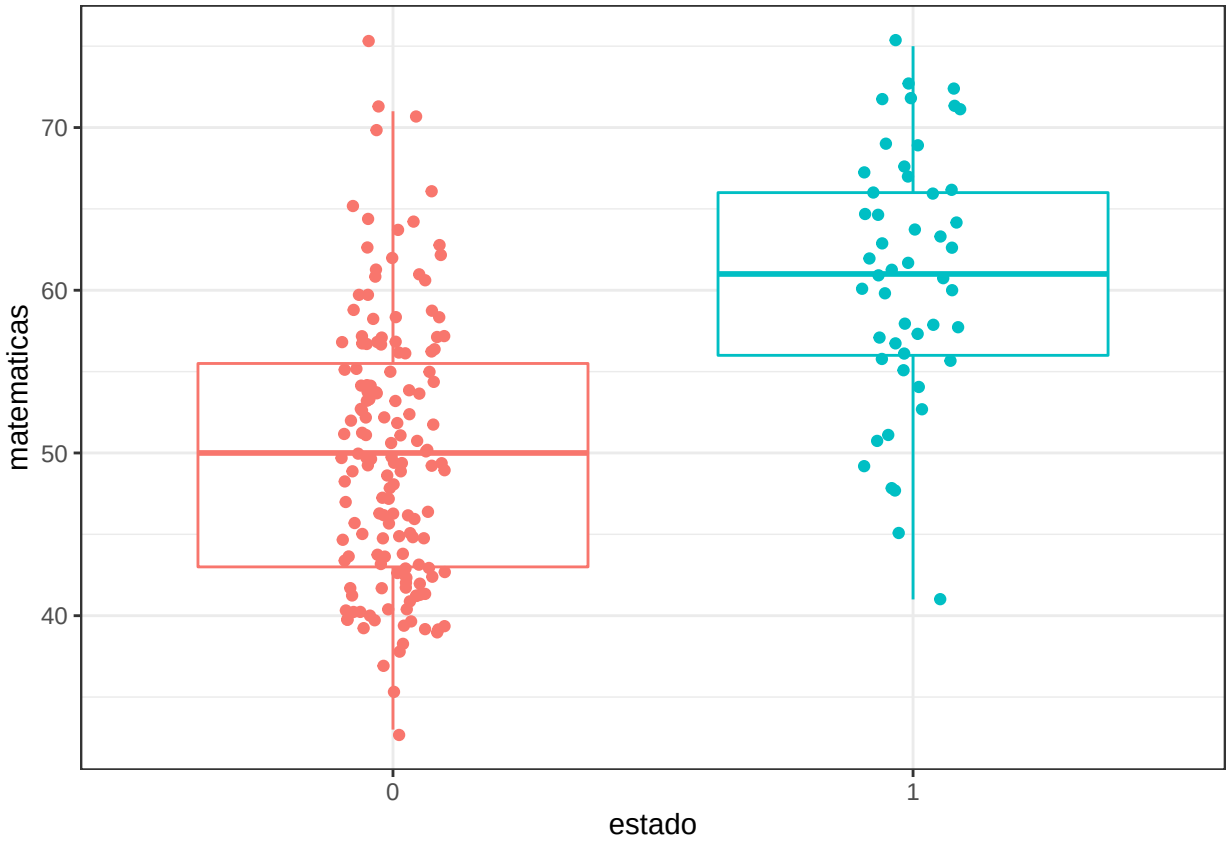
```
table(datos$estado)
```

```
##
##   0   1
## 151  49
```

Visualiza diagrama de caja

El comportamiento que tiene la varibles estado con respecto a variable matemáticas

```
ggplot(data = datos, aes(x = estado, y = matematicas, color = estado)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.1) +
  theme_bw() +
  theme(legend.position = "null")
```



Construir el modelo de Regresión logística

```
modelo.rlogis <- glm(data = datos,
              formula = estado  ~ matematicas,
              family = "binomial")

modelo.rlogis <- glm(data = datos,
              formula =  estado ~ matematicas,
              family = "binomial")
```

Summary del modelo

```
summary(modelo.rlogis)
```

```
##
## Call:
```

```
## glm(formula = estado ~ matematicas, family = "binomial", data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0332  -0.6785  -0.3506  -0.1565   2.6143
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.79394    1.48174  -6.610 3.85e-11 ***
## matematicas  0.15634    0.02561   6.105 1.03e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 222.71  on 199  degrees of freedom
## Residual deviance: 167.07  on 198  degrees of freedom
## AIC: 171.07
##
## Number of Fisher Scoring iterations: 5
```

Nuevos datos

Datos con los cuales predecir

```
nuevos <- c(59, 35, 38, 78, 80, 60)
nuevos_datos <- data.frame(matematicas = nuevos)
```

Predicciones con el modelo

```
predicciones <- predict(object = modelo.rlogis, newdata = nuevos_datos, se.fit = TRUE)
predicciones
```

```
## $fit
##          1          2          3          4          5          6
## -0.5698611 -4.3220297 -3.8530086  2.4006056  2.7132863 -0.4135208
##
## $se.fit
##         1         2         3         4         5         6
## 0.1980352 0.6045386 0.5323227 0.5628173 0.6111662 0.2049887
##
## $residual.scale
## [1] 1
```

Convertir a valor probabilístico las predicciones

Utilizar la fórmula o función logic

```
# Mediante la función logit se transforman los a probabilidades.
predicciones_prob <- exp(predicciones$fit) / (1 + exp(predicciones$fit))
predicciones_prob
```

```
##          1          2          3          4          5          6
```

```
## 0.36126887 0.01309905 0.02077505 0.91687347 0.93780610 0.39806821
```

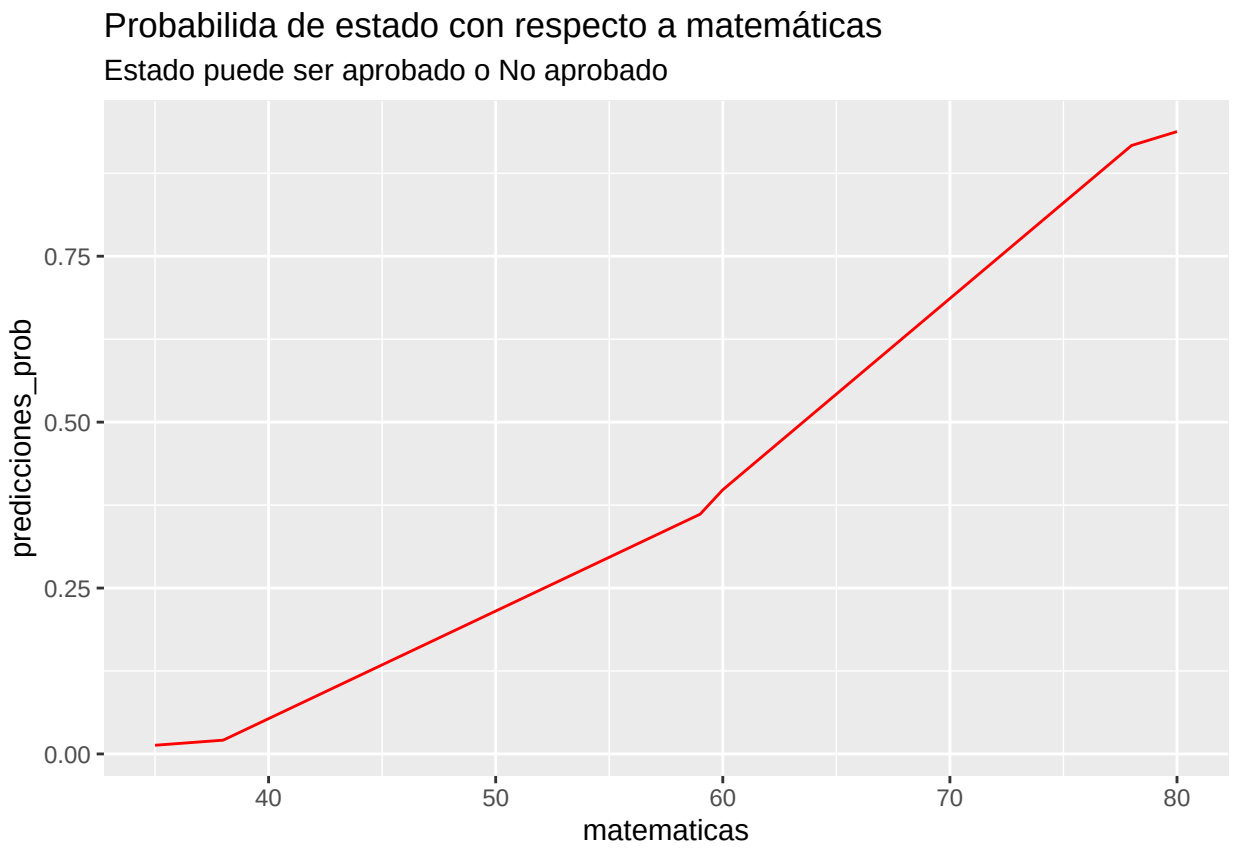
Construir un data.frame con los valore nuevos, las predicciones y los valores probabilísticos

```
comparaciones <- data.frame(nuevos_datos , predicciones, predicciones_prob)
comparaciones
```

```
##   matematicas        fit    se.fit residual.scale predicciones_prob
## 1          59 -0.5698611 0.1980352              1        0.36126887
## 2          35 -4.3220297 0.6045386              1        0.01309905
## 3          38 -3.8530086 0.5323227              1        0.02077505
## 4          78  2.4006056 0.5628173              1        0.91687347
## 5          80  2.7132863 0.6111662              1        0.93780610
## 6          60 -0.4135208 0.2049887              1        0.39806821
```

Gráfica de S Sigmoide

```
ggplot(data = comparaciones) +
  geom_line(aes(x = matematicas, y = predicciones_prob), col='red') +
  labs(title = "Probabilida de estado con respecto a matemáticas", subtitle = "Estado puede ser aprobad
```



Coeficientes

```
b0 <- modelo.rlogis$coefficients[1]
b1 <- modelo.rlogis$coefficients[2]
```

Predecir de manera manual

```
nuevos <- c(50, 67, 80, 60)

predicciones2 <- b0 + b1 * nuevos
predicciones2
```

```
## [1] -1.9769243  0.6808617  2.7132863 -0.4135208
```

Convertir a valores probabilísticos

Determinar e

```
e <- exp(1)
```

```
probs <- e^(b0 + b1 * nuevos) / (1 + e^(b0 + b1 * nuevos))
probs
```

```
## [1] 0.1216471 0.6639310 0.9378061 0.3980682
```

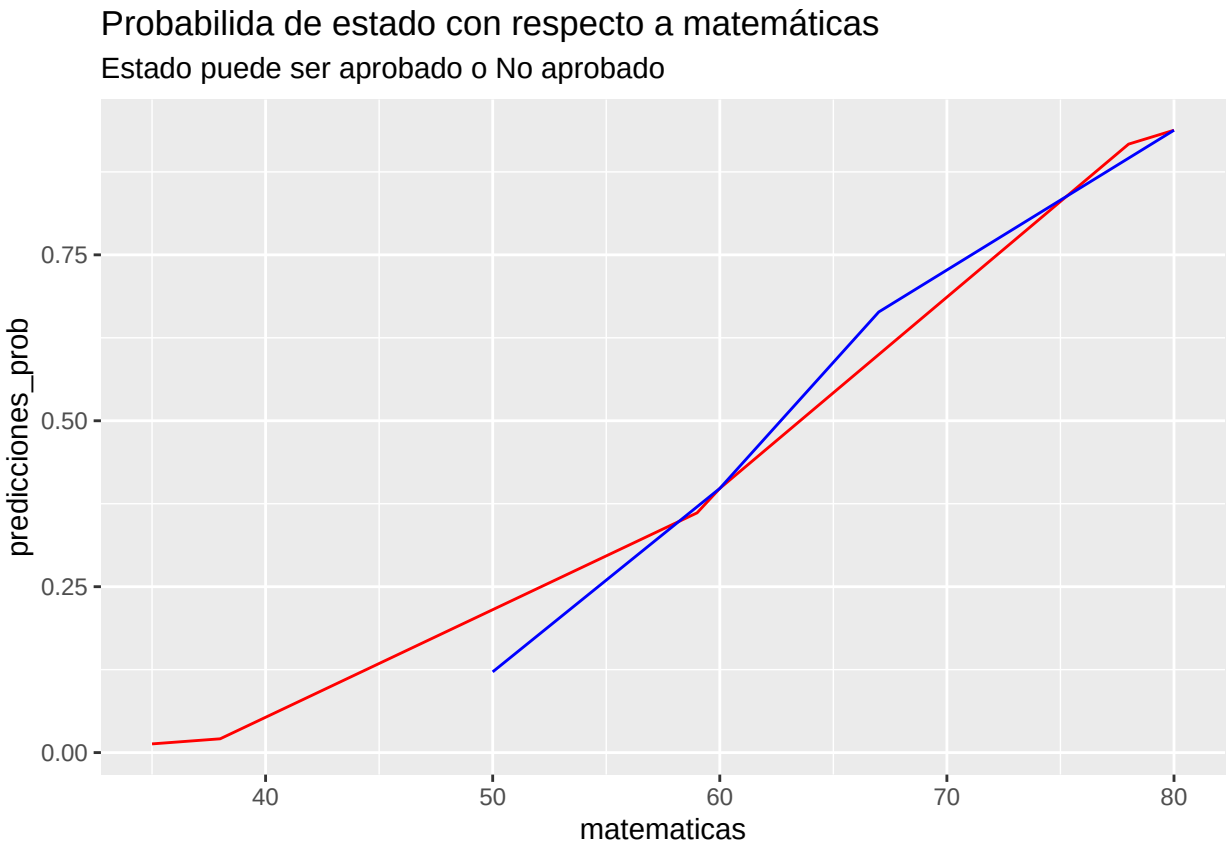
Integrar en un datase nuevo solo para graficar

```
nuevos.datos.frame <- data.frame(nuevos, probs)
nuevos.datos.frame
```

```
##   nuevos     probs
## 1     50 0.1216471
## 2     67 0.6639310
## 3     80 0.9378061
## 4     60 0.3980682
```

Nuevas predicciones

```
ggplot() +
  geom_line(data = comparaciones, aes(x = matematicas, y = predicciones_prob), col='red') +
  geom_line(data = nuevos.datos.frame, aes(x = nuevos, y = probs), col='blue') +
  labs(title = "Probabilida de estado con respecto a matemáticas", subtitle = "Estado puede ser aprobad
```



Interpretación

Bibliografía