Trabajo entregable final, Módulo 4: Modelos predictivos 2, Programa: experto en Ciencias de Datos

Ing. Leonel Morales

2022-09-17

Presentación del Problema

Se tiene una data llamada "germancredit", donde se encuentra información de los clientes de un banco que son malos o buenos pagadores, ademas, se tiene información de su edad, numero de tarjetas de crédito, cantidad de letras pagadas. Se desea construir dos modelos logit y Probit, de donde se sirva para pronosticar. En Efecto;

- duration:= es la variable que representa el numéro de años del crédito.
- installment:= representa a la variable: número de cuotas pagadas.
- age:= edad del cliente.
- cards:= número de tarjetas de crédito que posee el cliente.

Solución

Cargamos la base de datos en R-studio y la delimitamos,

```
germancredit <- read.csv("C:/Users/Admin/Desktop/germancredit.csv")
attach(germancredit)
credit<-germancredit[c("Default","duration","installment","age","cards")]
```

A continuación presentamos el modelo no lineal dado por:

$$Default_i = duration_i + installment_i + age_i + cards_i, \qquad \forall\, i \in [1, 1000],$$

de donde la variable Default es categorica es decir, es 1 si es un buen pagador y es 0 si es un mal pagador.

Ajuste de los modelos

Empezamos con el ajuste del modelo logit, esto es:

```
logitModel<-glm(Default~.,family = binomial(link = "logit"),data = credit)
summary(logitModel)
```

```
Call:
glm(formula = Default ~ ., family = binomial(link = "logit"),
    data = credit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5187  -0.8535  -0.7055   1.2195   2.1793

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.251089   0.356280  -3.512 0.000446 ***
duration     0.037013   0.005761   6.424 1.33e-10 ***
installment  0.141097   0.065578   2.152 0.031429 *
age         -0.018499   0.006755  -2.739 0.006172 **
cards       -0.131029   0.129223  -1.014 0.310595
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1163.5  on 995  degrees of freedom
AIC: 1173.5

Number of Fisher Scoring iterations: 4
```

Ahora bien, puesto que es un modelo no lineal no se puede interpretar directamente los resultados, pero notemos que la variable "cards" no es significativa, en consecuencia, se puede construir otro modelos sin esta variable y analizar.

De manera similar, ajustamos un modelo Probit:

```
probitModel<-glm(Default~.,family = binomial(link = "probit"),data = credit)
summary(probitModel)
```

```
Call:
glm(formula = Default ~ ., family = binomial(link = "probit"),
    data = credit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5086  -0.8538  -0.7039   1.2258   2.2107

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.773221   0.210589  -3.672 0.000241 ***
duration     0.022587   0.003481   6.490 8.61e-11 ***
installment  0.081749   0.038861   2.104 0.035414 *
age         -0.010642   0.003952  -2.693 0.007077 **
cards       -0.080756   0.076367  -1.057 0.290299
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1163.4  on 995  degrees of freedom
AIC: 1173.4

Number of Fisher Scoring iterations: 4
```

Nótese que, de manera similar al Logit, éste modelo tiene una variable no significativa que es "cards".

Contraste Hosmer y Lemeshow para los dos modelos

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  credit$Default, fitted(logitModel)
X-squared = 8.0255, df = 8, p-value = 0.431
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  credit$Default, fitted(probitModel)
X-squared = 9.1897, df = 8, p-value = 0.3265
```

es evidente que el $p - valor < 0.05$ para los dos contrastes, esto significa que los dos modelos tienen un buen ajuste, más aún, se debe validar este supuesto con más contrastes. Para el efecto usamos la matriz de Confusión.

Matriz de Confusión

Hallamos la media de los valores ajustados de cada modelo y a esos valores los denotamos como el umbral para los dos modelos respectivamente.

```
umb<-mean(logitModel$fitted.values)
umb
```

```
[1] 0.3
```

```
umbPro<-mean(probitModel$fitted.values)
umbPro
```

```
[1] 0.2999361
```

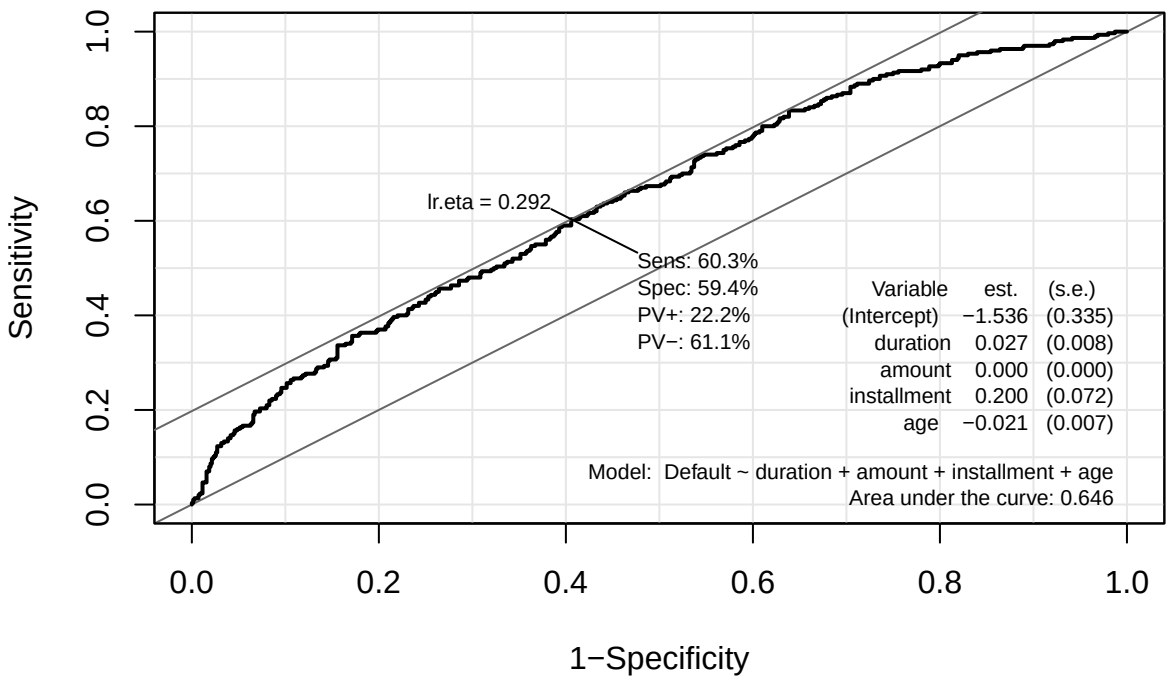
de modo que, la matriz de confusion para Logit es dada por:

```
$rawtab
       resp
          0   1
  FALSE 441 133
  TRUE  259 167

$classtab
       resp
```

```
               0         1
  FALSE 0.6300000 0.4433333
  TRUE  0.3700000 0.5566667

$overall
[1] 0.608

$mcFadden
[1] 0.04769035


$rawtab
       resp
          0   1
  FALSE 435 130
  TRUE  265 170

$classtab
       resp
                0         1
  FALSE 0.6214286 0.4333333
  TRUE  0.3785714 0.5666667

$overall
[1] 0.605

$mcFadden
[1] 0.04777674
```

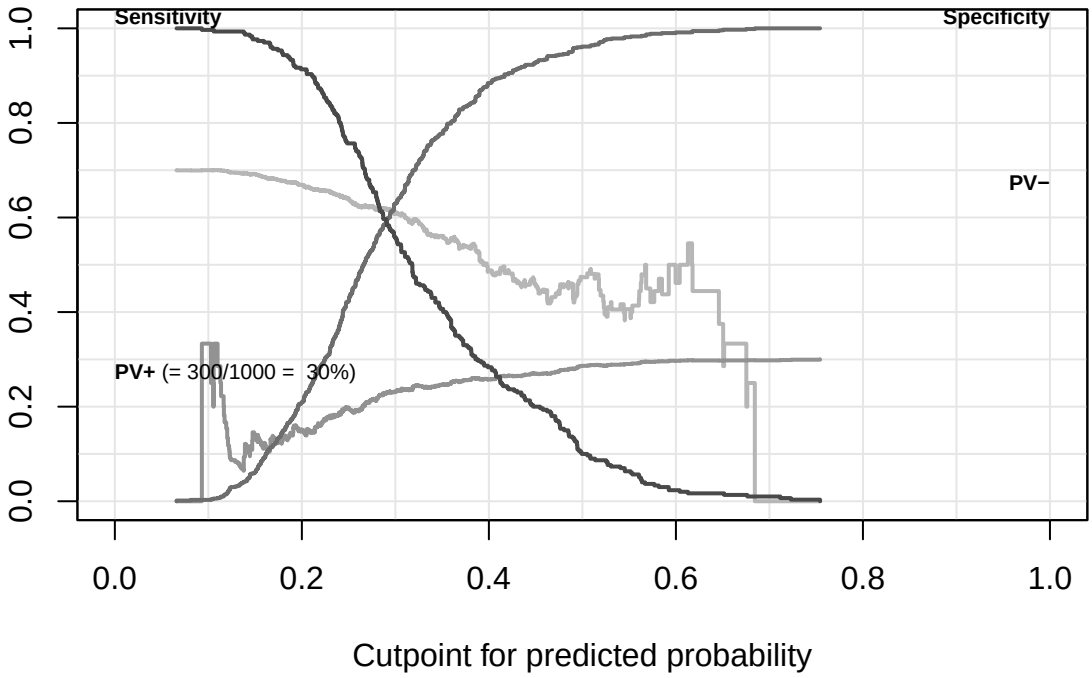
Podemos observar que el porcentaje de clasificación para logit es de 60.8, mientras que para el logit es de 60.5, entonces podemos tener una breve idea que estos modelo no funcionarían pues no daría una buena predicción.

Evaluando la capacidad predictiva a traves de otros criterios (ROC) para Logit



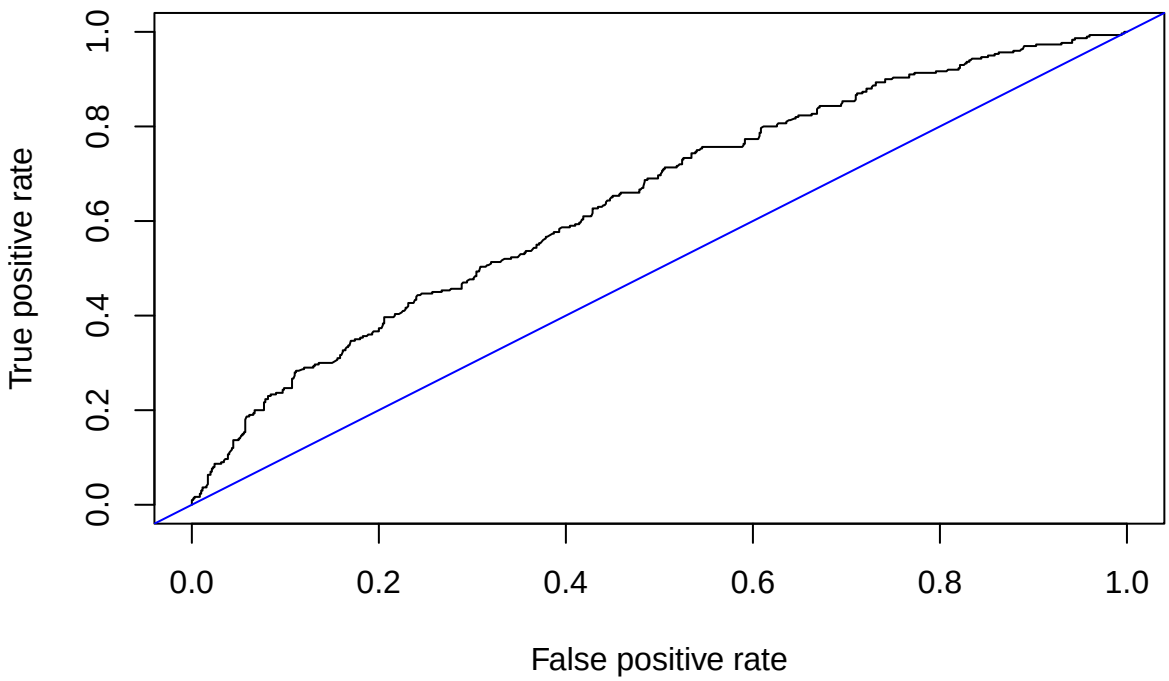
el gráfico muestra que el área bajo la curva entre especificidad y sensitividad es de 0.646, lo ideal sería que permanezca en un intervalo mayor o igual a 0.80, por tanto se confirma que el modelo no es bueno para una predicción.

Por otro lado hallamos el punto de corte óptimo real o umbral para el modelo Logit.



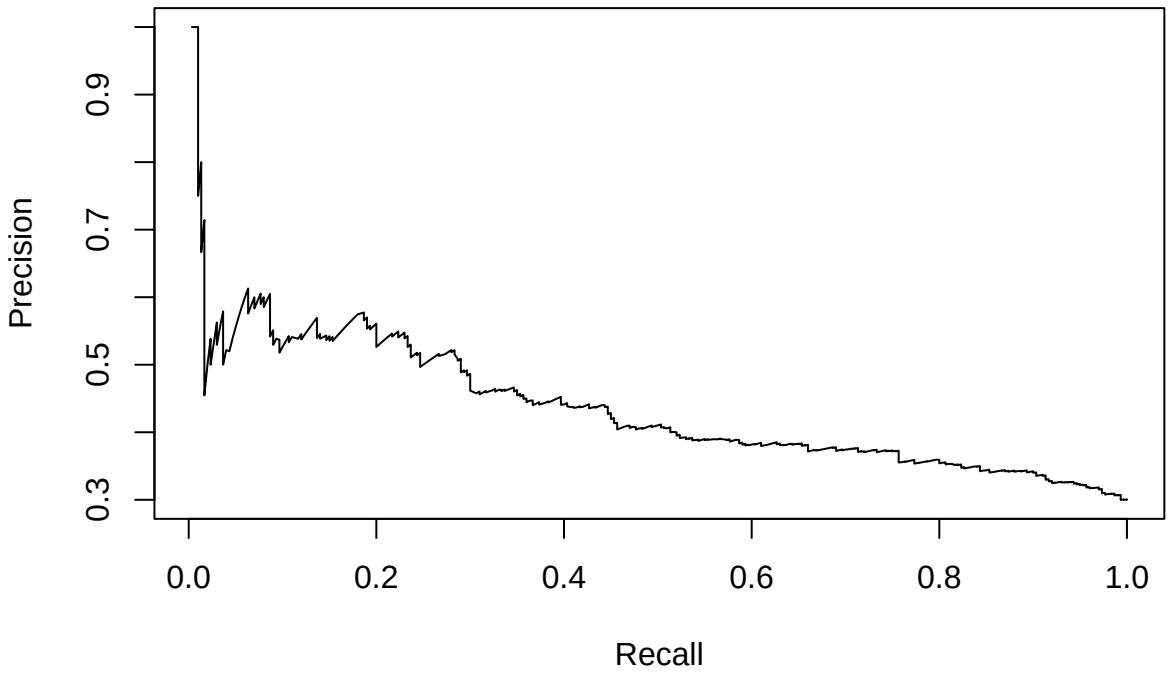
Observación. Para el modelo logit, coincide el valor del umbral hallado por la media de los valores ajustados siendo 0.30

Evaluando la capacidad predictiva curva ROC para Probit



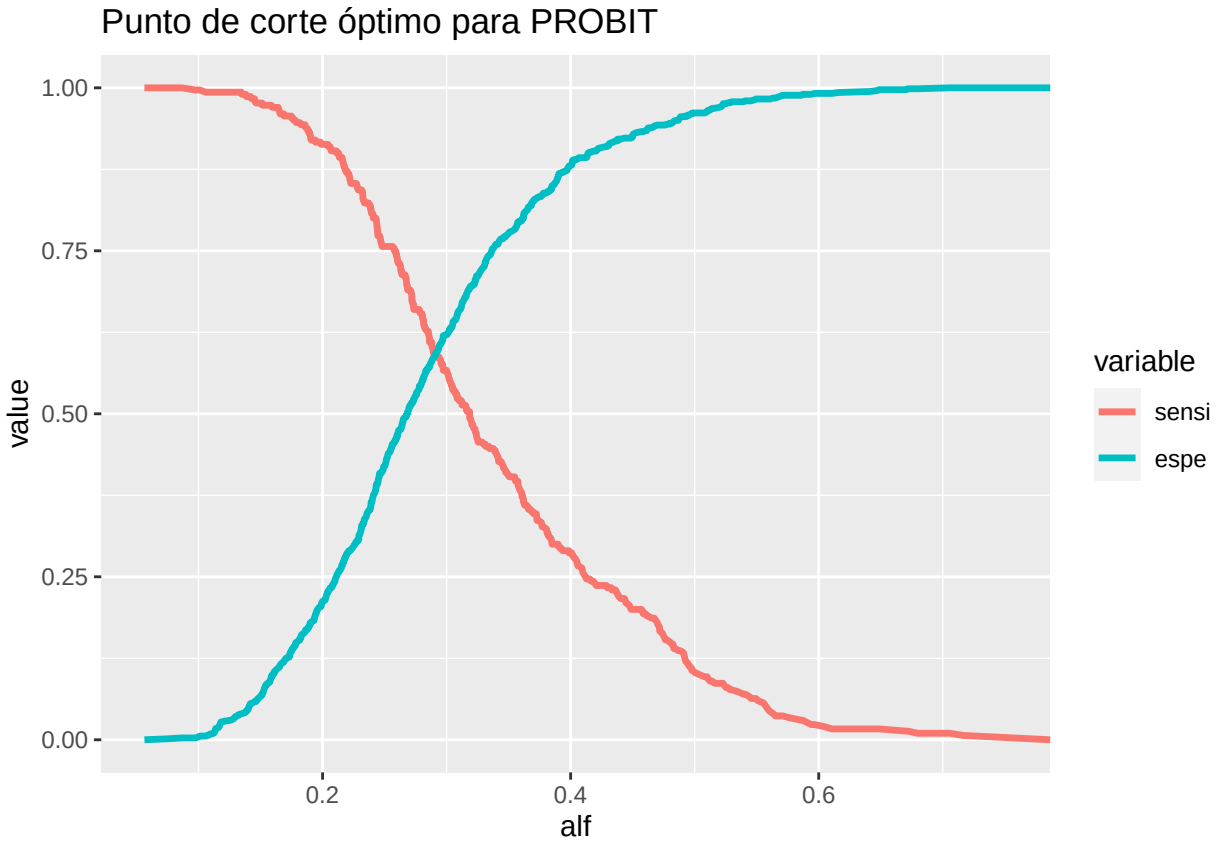
Observación. El área bajo la curva ROC no se acerca a 1, en consecuencia Probit no es un buen modelo.

Curvas de precisión



Vemos que la curva no cae lentamente, así tenemos un indicio que este modelo no funcionará.

Punto de corte óptimo para Probit



Usando la función **ggplotly** de la librería plotly, recuperamos el valor del umbral, siendo, 0.29115213, ya además cambiamos el umbral en la matriz de confusión solo para el probit, pues para el logit es el mismo valor.

```
$rawtab
       resp
          0   1
  FALSE 414 123
  TRUE  286 177

$classtab
       resp
                 0         1
  FALSE 0.5914286 0.4100000
  TRUE  0.4085714 0.5900000

$overall
[1] 0.591

$mcFadden
[1] 0.04777674
```

Comparación de los dos modelos

```
Calls:
logitModel: glm(formula = Default ~ ., family = binomial(link = "logit"),
    data = credit)
probitModel: glm(formula = Default ~ ., family = binomial(link = "probit"),
    data = credit)

=================================================
                     logitModel     probitModel
-------------------------------------------------
  (Intercept)         -1.251089***    -0.773221***
                      (0.356280)      (0.210589)
  duration             0.037013***     0.022587***
                      (0.005761)      (0.003481)
  installment          0.141097*       0.081749*
                      (0.065578)      (0.038861)
  age                 -0.018499**     -0.010642**
                      (0.006755)      (0.003952)
  cards               -0.131029       -0.080756
                      (0.129223)      (0.076367)
-------------------------------------------------
  Nagelkerke R-sq.     0.080           0.080
  Deviance          1163.464        1163.358
  AIC               1173.464        1173.358
  BIC               1198.003        1197.897
  N                 1000            1000
=================================================
  Significance: *** = p < 0.001; ** = p < 0.01;
                * = p < 0.05
```

Así, según AIC, BIC, N, el modelo probit es mejor, pero debido a la medida de los **overall** de los dos modelos se concluye que ninguno sirve.

Cambiando el umbral para los dos modelos

```
$rawtab
       resp
          0   1
  FALSE 649 240
  TRUE   51  60

$classtab
       resp
                  0          1
  FALSE 0.92714286 0.80000000
  TRUE  0.07285714 0.20000000

$overall
[1] 0.709

$mcFadden
[1] 0.04769035
```

```
$rawtab
       resp
          0   1
  FALSE 649 240
  TRUE   51  60

$classtab
       resp
                  0          1
  FALSE 0.92714286 0.80000000
  TRUE  0.07285714 0.20000000

$overall
[1] 0.709

$mcFadden
[1] 0.04777674
```

Cambiando el umbral de manera arbitraria a 0.45, se observa que clasifica mejor y los overall suben, no obstante, siguen siendo malos modelos de predicción pues *overall* $\in [0.80, 0.95]$.

Pronósticando

Se usan los dos modelos para pronósticos pero notemos que ninguno de los dos está bien definido. Suponga que se requiere predecir la probabilidad de que sea buen pagador si, la duración del préstamo es de 10 años, número de cuotas pagadas es de 2, la edad es de: 24 años, y con un número de tarjetas de crédito igual a 3.

```
predic<-data.frame(duration=10,installment=2,age=24,cards=3)
predict(logitModel,newdata = predic,type = "response")
```

```
        1
0.1921907
```

```
predict(probitModel,newdata = predic,type = "response")
```

```
        1
0.1890124
```

Suponga que se requiere predecir la probabilidad de que sea buen pagador si, la duración del préstamo es de 6 años, número de cuotas pagadas es de 24, la edad es de: 43 años, y con un número de tarjetas de crédito igual a 2.

```
predic2<-data.frame(duration=6,installment=24,age=43,cards=2)
predict(logitModel,newdata = predic2,type = "response")
```

```
        1
0.7857994
```

Conlclusiones

- La probabilidad de que sea un buen pagador, para la priemra predicción, según estos parametros es del 19.21 por ciento, mientras que para probit es de 18.90.
- Para la segunda predicción se tiene que, la probabilidad que sea un buen pagador es del 78.57 por ciento para Logit.
- Ningun modelo es bueno para predecir pues el porcentaje de clasificación está por debajo del 80 por ciento.
- Quitando la variable que no es significativa **cards** se llega a que ninguno de los dos modelos tampoco es bueno.