Ejercicio Tema 4. Métodos de regularización, Regresión Logística y validación clasificada.

Máster en Ciencia de Datos. Módulo: Análisis exploratorio de datos

Ana Navarro Quiles

Curso 2022/2023

Objetivos:

Se pretende practicar los conceptos relacionados con los métodos de regularización (Regresión Ridge, Lasso y Elastic Net), Regresión Logística y validación clasificada

Ejercicios propuestos

Ejercicio 1

El banco de datos **diabetes** contiene información de seguimiento de 149 pacientes con diabetes. El objetivo es estudiar el efecto de las variables sobre la mortalidad. Las variables contenidas en el fichero son las siguientes:

- Mortalidad por diabetes, variable *MORT*
- Edad del paciente, variable *EDAT*
- Índice de masa corporal, variable *BMI*
- Resultado del electrocardiograma, variable *ECG*
- Antecedentes coronarios, variable *CHD*
- Presión arterial sistólica y diastólica, variables *SBP* y *DBP*, respectivamente

a) Ajusta el modelo de regresión más adecuado para analizar la mortalidad teniendo en cuenta todas las variables indicadas en el enunciado.

– Calcula el porcentaje de predicciones acertadas usando todos los datos. Para ello, haz la tabla de clasificación correspondiente.

– ¿Quienes son más propensos a morir por diabetes, los que tienen antecedentes coronarios o los que no?

– Calcula dicha probabilidad para aquellos pacientes con una edad de 45 años, un índice de masa corporal de 30, un electrocardiograma Normal y una presión SBP=135 y DBP= 70.

```
load('datosTema2.Rdata')
ajuste.mortalidad.completo <- glm(MORT ~ EDAT+BMI+ECG+CHD+SBP+DBP, data = diabetes,
                                  family = binomial())
summary(ajuste.mortalidad.completo)
```

```
##
## Call:
## glm(formula = MORT ~ EDAT + BMI + ECG + CHD + SBP + DBP, family = binomial(),
##     data = diabetes)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.6146  -0.5007  -0.3806  -0.2491   2.4809
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.046883   2.289525  -2.204 0.027501 *
## EDAT         0.087593   0.025389   3.450 0.000561 ***
## BMI         -0.014249   0.041707  -0.342 0.732613
## ECGFrontera  1.387768   1.265330   1.097 0.272745
## ECGAnormal   3.048167   1.282509   2.377 0.017467 *
## CHDSi       -1.555899   1.205173  -1.291 0.196698
## SBP         -0.008573   0.014911  -0.575 0.565337
## DBP          0.001597   0.002309   0.691 0.489271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134.80  on 148  degrees of freedom
## Residual deviance: 104.19  on 141  degrees of freedom
## AIC: 120.19
##
## Number of Fisher Scoring iterations: 5
```

```
prob<-predict(ajuste.mortalidad.completo, type="response")
pred <- factor(prob>0.5,labels= levels(diabetes$MORT))

table(diabetes$MORT,pred)
```

```
##         pred
##          Vivo Muerto
##   Vivo    120      4
##   Muerto   15     10
```

```
130/149*100
```

```
## [1] 87.24832
```

```
exp(coef(ajuste.mortalidad.completo)["CHDSi"]) #Es menos probable morir de diabetes
```

```
##     CHDSi
## 0.2109997
```

```
# si se tiene antecedentes coronarios, la probabilidad de si es un 21% de la que no.


predict(ajuste.mortalidad.completo,data.frame(EDAT=45,BMI=30,ECG=c('Normal'),
                                              SBP=135,DBP=70,CHD=c('Si','No')),
                                              type = 'response')
```

```
##          1          2
## 0.01576463 0.07055483
```

```
# La probabilidad de Si es 0.0158 y de No 0.0705.
```

b) Intenta mejorar el modelo anterior utilizando la metodología stepwise para la elección de variables y vuelve a contestar a las preguntas del apartado anterior (considerando en la predicciones las variables que contenga tu nuevo modelo). ¿Hay cambios importantes en las conclusiones?

```r
ajuste.mortalidad.completo <- glm(MORT ~ EDAT+BMI+ECG+CHD++SBP+DBP, data = diabetes,
                                  family = binomial())
ajuste.mortalidad<- step(ajuste.mortalidad.completo , direction = 'both',trace=0)
summary(ajuste.mortalidad)
```

```
##
## Call:
## glm(formula = MORT ~ EDAT + ECG + CHD, family = binomial(), data = diabetes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5969  -0.5118  -0.3717  -0.2554   2.4662
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.41135    1.30506  -4.913 8.98e-07 ***
## EDAT         0.08548    0.02267   3.770 0.000163 ***
## ECGFrontera  1.31414    1.25997   1.043 0.296948
## ECGAnormal   3.01864    1.28112   2.356 0.018460 *
## CHDSi       -1.55785    1.18742  -1.312 0.189532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134.80  on 148  degrees of freedom
## Residual deviance: 105.03  on 144  degrees of freedom
## AIC: 115.03
##
## Number of Fisher Scoring iterations: 5
```

```r
prob<-predict(ajuste.mortalidad, type="response")
pred <- factor(prob>0.5,labels= levels(diabetes$MORT))
table(diabetes$MORT,pred)
```

```
##         pred
##          Vivo Muerto
##   Vivo    121      3
##   Muerto   15     10
```

```r
131/149*100
```

```
## [1] 87.91946
```

```r
exp(coef(ajuste.mortalidad)["CHDSi"]) #Es menos probable morir de diabetes si
```

```
##     CHDSi
## 0.2105887
```

```r
# se tiene antecedentes coronarios, la probabilidad de si es un 21% de la que no.

predict(ajuste.mortalidad,data.frame(EDAT=45,ECG=c('Normal'),CHD=c('Si','No')),
        type = 'response')
```

```
##          1          2
## 0.01594288 0.07143693
```

c) Intenta mejorar el modelo anterior utilizando la metodología regsubsets para la elección de variables y vuelve a contestar a las preguntas del apartado anterior (considerando en la predicciones las variables que contenga tu nuevo modelo). ¿Hay cambios importantes en las conclusiones?

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```
ajuste.mortalidad.reg <- regsubsets(MORT ~ EDAT+BMI+ECG+CHD+SBP+DBP, data = diabetes)
resumen<-summary(ajuste.mortalidad.reg)
which.min(resumen$cp)
```

```
## [1] 2
```

```
colnames(resumen$which)[resumen$which[2,]==T]
```

```
## [1] "(Intercept)" "EDAT"        "ECGAnormal"
```

```
ajuste.mortalidad_2 <- glm(MORT ~ EDAT+ECG, data = diabetes,family = binomial())

prob<-predict(ajuste.mortalidad_2, type="response")
pred <- factor(prob>0.5,labels= levels(diabetes$MORT))
table(diabetes$MORT,pred)
```

```
##         pred
##          Vivo Muerto
##   Vivo    119      5
##   Muerto   16      9
```

```
128/149*100
```

```
## [1] 85.90604
```

```
# No podemos contestar con criterio la pregunta, dado que ya no está la variable ECG

AIC(ajuste.mortalidad,ajuste.mortalidad_2) # perdemos respecto al de stepwise un poco
```

```
##                     df      AIC
## ajuste.mortalidad    5 115.0290
## ajuste.mortalidad_2  4 115.2725
```

```
BIC(ajuste.mortalidad,ajuste.mortalidad_2) # Ganamos respecto al de stepwise un poco
```

```
##                     df      BIC
## ajuste.mortalidad    5 130.0487
## ajuste.mortalidad_2  4 127.2883
```

```
anova(ajuste.mortalidad,ajuste.mortalidad_2,test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: MORT ~ EDAT + ECG + CHD
## Model 2: MORT ~ EDAT + ECG
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       144     105.03
## 2       145     107.27 -1  -2.2435   0.1342
```

```
#Vemos que CHD no es significativo
```

c) Ahora queremos obtener información sobre el electrocardiograma a partir del resto de variables (excepto muerte). Utilizar análisis discriminante

```
load('datosTema2.Rdata')


library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

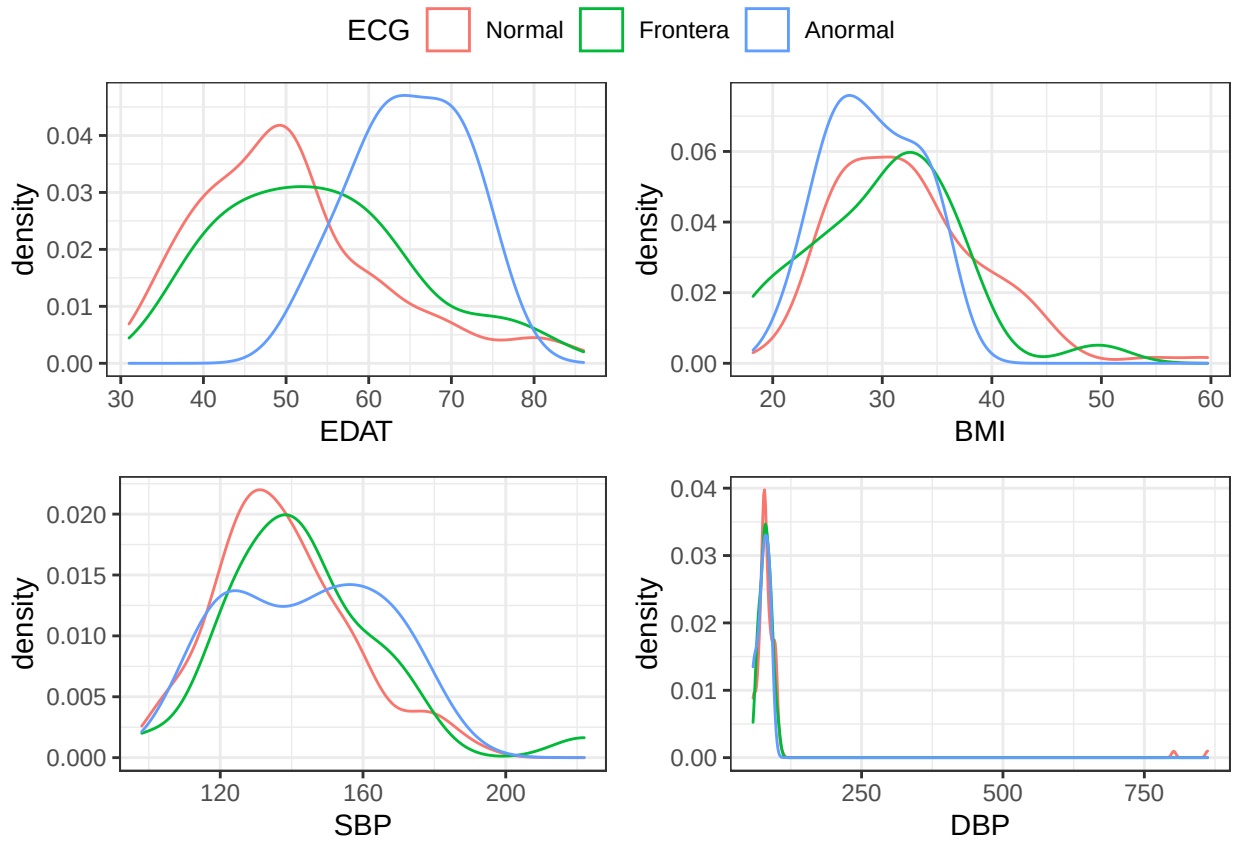
```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.1.3
```

```
plot1 <- ggplot(data = diabetes, aes(x = EDAT)) +
  geom_density(aes(colour = ECG)) + theme_bw()
plot2 <- ggplot(data = diabetes, aes(x = BMI)) +
  geom_density(aes(colour = ECG)) + theme_bw()
plot3 <- ggplot(data = diabetes, aes(x = SBP)) +
  geom_density(aes(colour = ECG)) + theme_bw()
plot4 <- ggplot(data = diabetes, aes(x = DBP)) +
  geom_density(aes(colour = ECG)) + theme_bw()
# la función grid.arrange del paquete grid.extra permite ordenar
# graficos de ggplot2
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.1.3
```

```
ggarrange(plot1, plot2, plot3, plot4, common.legend = TRUE)
```



```r
# La gráfica sugiera que únicamente debemos tener en cuenta EDAT y CHD.


library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```r
m_lda <- lda(ECG ~ EDAT+BMI+CHD+SBP+DBP, data=diabetes)
diabetes2<-diabetes[4:11][-3][-3]
predLDA<-predict(m_lda, newdata=diabetes2)
t<-table(diabetes2$ECG, predLDA$class, dnn = c("Clase real", "Clase predicha"))
t
```

```
##           Clase predicha
## Clase real Normal Frontera Anormal
##   Normal       99        8       4
##   Frontera      0       24       3
##   Anormal       0        6       5
```

```r
n<-sum(t)
100*sum(diag(t))/n # Aquí simplemente vemos cuantos hemos "acertado". Es decir,
```

```
## [1] 85.90604
```

```r
# que precisión tenemos 85.90604%

# Podriamos hacer una la predicción

p<-predict(m_lda, newdata=data.frame(EDAT=45,BMI=30,SBP=135,DBP=70,CHD=c('Si','No')))
p #Fijaros que nos da:
```

```
## $class
## [1] Frontera Normal
## Levels: Normal Frontera Anormal
##
## $posterior
##        Normal     Frontera      Anormal
## 1 0.008104658 0.9024163670 8.947897e-02
## 2 0.999603530 0.0002998999 9.657005e-05
##
## $x
##         LD1        LD2
## 1  2.576311 -1.0322953
## 2 -1.184240 -0.3844613
```

```r
#P(ECG=normal | CHD=Si)=0.0081... P(ECG=Frontera | CHD=Si)=0.90... y P(ECG=Anormal | CHD=Si)=0.0089...
sum(p$posterior[1,])
```

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

```r
sum(p$posterior[2,])
```

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

```r
# Hemos dicho que antes parecía solo importar la EDAT y CHD. Vamos a hacerlo solo
# con esas dos:
```

```r
library(MASS)
m_lda_1 <- lda(ECG ~ EDAT+CHD, data=diabetes)
diabetes2<-diabetes[4:11][-2:-6]

predLDA<-predict(m_lda_1, newdata=diabetes2)
t<-table(diabetes2$ECG, predLDA$class, dnn = c("Clase real", "Clase predicha"))
t
```

```
##           Clase predicha
## Clase real Normal Frontera Anormal
##   Normal       99        9       3
##   Frontera      0       24       3
##   Anormal       0        7       4
```

```r
n<-sum(t)
100*sum(diag(t))/n # Aquí simplemente vemos cuantos hemos "acertado". Es decir,
```

```
## [1] 85.2349
```

```r
# que precisión tenemos, 85.23%%

# Podriamos hacer uno predicción

p<-predict(m_lda_1, newdata=data.frame(EDAT=45,CHD=c('Si','No')))
p #Fijaros que nos da:
```

```
## $class
## [1] Frontera Normal
## Levels: Normal Frontera Anormal
##
## $posterior
##       Normal     Frontera      Anormal
## 1 0.00847633 0.8977573395 9.376633e-02
## 2 0.99960491 0.0003058926 8.919884e-05
##
## $x
##         LD1        LD2
## 1  2.563725 -1.1011550
## 2 -1.204928 -0.4368889
```

```r
#P(ECG=normal | CHD=Si)=0.0084... P(ECG=Frontera | CHD=Si)=0.897... y
# P(ECG=Anormal | CHD=Si)=0.00937... Y análogo para no. Si hacemos la suma da 1.
sum(p$posterior[1,])
```

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

```r
sum(p$posterior[2,])
```

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

```r
# Vemos que no hay mucho cambio y nos hemos quitado algunas variables.
```