

Problemas propuestos de Regresión Logística

Francisco Javier Mercader Martínez

Problema 1

El fichero **processed.cleveland.data**, contiene los datos correspondientes a un estudio sobre enfermedad cardíaca por *Cleveland Clinic Foundation*.

El fichero contiene un total de 14 columnas, correspondientes a las siguientes variables: *age*, *sex*, *cp*, *trestbps*, *chol*, *fbs*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, *ca*, *tal* y *num*. La variable “num” toma valores 0, 1, 2, 3 y 4, indicando el tipo de anomalía cardíaca. El valor 0 indica ausencia de enfermedad, mientras que el resto de valores indican algún tipo de anomalía. Para la descripción detallada de cada variable, puede consultarse el fichero **heart-disease.names**.

Se desea realizar un análisis de Regresión Logística con el fin de predecir la presencia (o no) de enfermedad cardíaca en función del resto de variables (predictores). Se pide:

- 1) Importar los datos del fichero **processed.cleveland.data** y poner el nombre de cada variable como se indica en el enunciado. Sustituir la variable “num” por una nueva variable llamada “disease” que valga 0 si no hay enfermedad y que valga 1 cuando haya anomalía cardíaca.

```
mydata <- read.table("../data/processed.cleveland.data", sep = ",",
                     col.names = c("age", "sex", "cp", "trestbps", "chol", "fbs",
                                   "restecg", "thalach", "exang", "olpeak", "slope",
                                   "ca", "tal", "num"))

# Crear la nueva variable 'disease'
mydata$disease <- ifelse(mydata$num > 0, 1, 0)
mydata <- subset(mydata, select = -num)
```

De esta forma elimino la columna num para que disease la sustituya

- 2) Eliminar todas las filas que tengan algún valor perdido. **Importante:** confirmar primero si todas las variables son de tipo numérico para identificar adecuadamente los valores perdidos.

```
summary(mydata)
```

##	age	sex	cp	trestbps
##	Min. :29.00	Min. :0.0000	Min. :1.000	Min. : 94.0
##	1st Qu.:48.00	1st Qu.:0.0000	1st Qu.:3.000	1st Qu.:120.0
##	Median :56.00	Median :1.0000	Median :3.000	Median :130.0
##	Mean :54.44	Mean :0.6799	Mean :3.158	Mean :131.7
##	3rd Qu.:61.00	3rd Qu.:1.0000	3rd Qu.:4.000	3rd Qu.:140.0
##	Max. :77.00	Max. :1.0000	Max. :4.000	Max. :200.0
##	chol	fbs	restecg	thalach
##	Min. :126.0	Min. :0.0000	Min. :0.0000	Min. : 71.0
##	1st Qu.:211.0	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:133.5
##	Median :241.0	Median :0.0000	Median :1.0000	Median :153.0
##	Mean :246.7	Mean :0.1485	Mean :0.9901	Mean :149.6
##	3rd Qu.:275.0	3rd Qu.:0.0000	3rd Qu.:2.0000	3rd Qu.:166.0

```
## Max. :564.0 Max. :1.0000 Max. :2.0000 Max. :202.0
## exang olpeak slope ca
## Min. :0.0000 Min. :0.00 Min. :1.000 Length:303
## 1st Qu.:0.0000 1st Qu.:0.00 1st Qu.:1.000 Class :character
## Median :0.0000 Median :0.80 Median :2.000 Mode :character
## Mean :0.3267 Mean :1.04 Mean :1.601
## 3rd Qu.:1.0000 3rd Qu.:1.60 3rd Qu.:2.000
## Max. :1.0000 Max. :6.20 Max. :3.000
## tal disease
## Length:303 Min. :0.0000
## Class :character 1st Qu.:0.0000
## Mode :character Median :0.0000
## Mean :0.4587
## 3rd Qu.:1.0000
## Max. :1.0000
```

```
# Las columnas ca y tal son de tipo chr
mydata$ca <- as.numeric(mydata$ca)
```

```
## Warning: NAs introducidos por coerción
```

```
mydata$tal <- as.numeric(mydata$tal)
```

```
## Warning: NAs introducidos por coerción
```

```
# Como nos da el aviso de que hay valores NA's, vamos a eliminar las filas que
# los contienen
```

```
mydata <- na.omit(mydata)
```

```
# El DataFrame original tenía 303 filas y ahora hay 297.
summary(mydata)
```

```
## age sex cp trestbps
## Min. :29.00 Min. :0.0000 Min. :1.000 Min. : 94.0
## 1st Qu.:48.00 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:120.0
## Median :56.00 Median :1.0000 Median :3.000 Median :130.0
## Mean :54.54 Mean :0.6768 Mean :3.158 Mean :131.7
## 3rd Qu.:61.00 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:140.0
## Max. :77.00 Max. :1.0000 Max. :4.000 Max. :200.0
## chol fbs restecg thalach
## Min. :126.0 Min. :0.0000 Min. :0.0000 Min. : 71.0
## 1st Qu.:211.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:133.0
## Median :243.0 Median :0.0000 Median :1.0000 Median :153.0
## Mean :247.4 Mean :0.1448 Mean :0.9966 Mean :149.6
## 3rd Qu.:276.0 3rd Qu.:0.0000 3rd Qu.:2.0000 3rd Qu.:166.0
## Max. :564.0 Max. :1.0000 Max. :2.0000 Max. :202.0
## exang olpeak slope ca
## Min. :0.0000 Min. :0.000 Min. :1.000 Min. :0.0000
## 1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:1.000 1st Qu.:0.0000
## Median :0.0000 Median :0.800 Median :2.000 Median :0.0000
## Mean :0.3266 Mean :1.056 Mean :1.603 Mean :0.6768
## 3rd Qu.:1.0000 3rd Qu.:1.600 3rd Qu.:2.000 3rd Qu.:1.0000
## Max. :1.0000 Max. :6.200 Max. :3.000 Max. :3.0000
## tal disease
```

```
## Min.      :3.000   Min.      :0.0000
## 1st Qu.:3.000   1st Qu.:0.0000
## Median :3.000   Median :0.0000
## Mean    :4.731   Mean    :0.4613
## 3rd Qu.:7.000   3rd Qu.:1.0000
## Max.    :7.000   Max.    :1.0000
```

3) Pasar a tipo factor las variables que por naturaleza sean de tipo categórico.

```
mydata$sex <- as.factor(mydata$sex)
mydata$fbs <- as.factor(mydata$fbs)
mydata$exang <- as.factor(mydata$exang)
mydata$disease <- as.factor(mydata$disease)
```

4) Dividir el conjunto de datos en entrenamiento y prueba (70% entrenamiento, 30% prueba). Tomar semilla 123.

```
library(caTools)

set.seed(123)

split <- sample.split(mydata$disease, SplitRatio = 0.7)

# Conjunto de entrenamiento
train_data <- subset(mydata, split == TRUE)

# Conjunto de test
test_data <- subset(mydata, split == FALSE)
```

5) Con los datos de entrenamiento, obtener el modelo ajustado de Regresión Logística usando todos los predictores. ¿Son todos los predictores significativos?

```
modelo_ajustado <- glm(disease ~ ., data = train_data, family = "binomial")
summary(modelo_ajustado)
```

```
##
## Call:
## glm(formula = disease ~ ., family = "binomial", data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.971802   3.481717  -2.290 0.022043 *
## age         -0.021535   0.030711  -0.701 0.483175
## sex1         1.032931   0.619496   1.667 0.095440 .
## cp          0.879473   0.246117   3.573 0.000352 ***
## trestbps     0.022726   0.012669   1.794 0.072846 .
## chol         0.004629   0.004528   1.022 0.306573
## fbs1        -1.015074   0.685242  -1.481 0.138517
## restecg      0.131747   0.228000   0.578 0.563375
## thalach     -0.016290   0.012481  -1.305 0.191819
## exang1       1.394518   0.509005   2.740 0.006150 **
## olpeak       0.199241   0.267522   0.745 0.456413
## slope       0.581427   0.458175   1.269 0.204439
## ca          1.131167   0.311410   3.632 0.000281 ***
## tal         0.322145   0.123885   2.600 0.009313 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 287.12  on 207  degrees of freedom
## Residual deviance: 138.63  on 194  degrees of freedom
## AIC: 166.63
##
## Number of Fisher Scoring iterations: 6
```

Los predictores con un valor p pequeño (generalmente menor a 0.05) son considerados significativos. En este caso, los predictores significativos son:

- (Intercept)
- cp
- exangl
- ca
- tal

Estos predictores tienen un valor p menor a 0.05, lo que indica que hay una fuerte evidencia de que estos predictores tienen un efecto significativo en la variable de respuesta **disease**.

- 6) Obtener las predicciones para los datos del conjunto de prueba, es decir, la probabilidad predicha de padecer cardíaca para cada individuo del conjunto de testeo.

```
predict(modelo_ajustado, newdata = test_data, type = "response")
```

```
##      4      6      8     14     17     18     26
## 0.34893535 0.02278467 0.28723231 0.10297714 0.23293333 0.24688767 0.05759789
##      31     33     34     35     37     40     43
## 0.07036064 0.12215276 0.58771304 0.24808972 0.94367812 0.35342295 0.18896880
##      46     51     52     53     54     56     59
## 0.66448604 0.02082296 0.34553883 0.49235927 0.03197073 0.98707158 0.59148777
##      64     66     67     72     74     90     94
## 0.01396034 0.99580876 0.25181826 0.76896489 0.80424216 0.05687044 0.02240736
##      97    102    103    108    118    120    122
## 0.95468208 0.01351296 0.26553558 0.70885182 0.08609493 0.93846688 0.99219822
##     126     127     128     129     130     132     134
## 0.02980775 0.99683843 0.96985953 0.02651056 0.04265816 0.65516576 0.51163510
##     138     139     142     152     155     156     161
## 0.71635547 0.92290473 0.26850677 0.21608609 0.96896446 0.98210957 0.02452786
##     163     164     175     178     179     190     194
## 0.04182684 0.15740948 0.96601353 0.97849105 0.40245394 0.97506230 0.86618648
##     195     196     198     199     200     204     211
## 0.09316698 0.95192115 0.53911457 0.01332241 0.06568729 0.20241060 0.02585124
##     213     214     218     228     229     230     232
## 0.23436954 0.94248761 0.53190674 0.10317672 0.88378520 0.70912146 0.91630418
##     236     239     241     243     249     251     253
## 0.98674772 0.02994631 0.03168760 0.08189462 0.86326256 0.82030866 0.97340735
##     254     261     265     266     268     274     275
## 0.04900710 0.16995412 0.94100651 0.95418009 0.44248885 0.08818604 0.11818867
##     276     299     300     301     302
## 0.22314803 0.14187489 0.91103186 0.96164053 0.06273943
```

- 7) Veamos ahora el problema de Regresión Logística como un problema de clasificación. Usando las predicciones del apartado anterior y tomando como punto de corte la probabilidad de 0.5, obtener la clase predicha para los individuos del conjunto de prueba. Medir la eficiencia del modelo calculando la

matriz de confusión, accuracy, sensibilidad y especificidad.

- 8) Para los datos del conjunto de prueba, obtener la curva ROC del método de clasificación, calcular el AUC (área bajo la curva) e interpretar el resultado.
- 9) Repetir el análisis (apartado 5 y siguientes) pero aplicando primero los métodos de selección de regresores, con el fin de proponer un modelo más parsimonioso.