Diego M. Dell'Era

2º Parcial

Regresión Logística

1 Ajustar un modelo de regresión logística para el conjunto de datos census.txt. En las planillas var.xlsx y categorizaciones.xlsx, se encontrará la información con el contenido original del conjunto de datos y algunas re categorizaciones. El conjunto de datos contiene información de una población con la cual se intenta predecir si puede o no cobrar más de 50.000 dólares anuales. La clase indica 1 en el caso que dicha persona cobra más de 50.000 dólares anuales y 0 sino es así.

1.1.1 Separe las poblaciones en entrenamiento y validación en forma aleatoria en 70/30 (genere una semilla aleatoria e indique el valor de la semilla en el software que utilizó). Indique que cantidad de casos quedaron para cada ambiente.

1.1.2 Ajuste el mejor modelo posible de regresión logística. Indique el modelo con todas sus variables y en el caso de que contenga variables categóricas que beta corresponde a cada categoría original.

1.1.3 Calcular el AUC y el gráfico ROC en entrenamiento y validación indicando también el total de casos de cada una de las clases.

1.1.4 Selecciones el 25% de los individuos en el ambiente de validación de acuerdo a la siguiente lógica. Entregue los resultados indicados:

- Al azar e indique la cantidad de individuos que cobran más de 50.000 dólares.
- Utilizando el modelo desarrollado en el punto 1.1.2 e indique la cantidad de individuos que cobran más de 50.000 dólares.

1.1.5 Calcular y/o obtener los siguientes resultados:

- Indicar en cuanto sería el impacto en modificar una unidad de por lo menos una variable continua del modelo.
- Indicar si hay puntos incluyentes con COOK.
- Indicar que método de selección de variables se utilizó y explicar su funcionamiento.
- Mostrar el estadístico de Hosmer-Lemeshow en el último paso del modelo.

1.1.6 Entregue un conjunto de datos en formato texto con el siguiente formato:

Caso
Indicador de Entrenamiento o Validación
Clase
Probabilidad calculada con el modelo utilizado para resolver el punto 1.1.4

////

0. leer datos:

```
data <-
read.csv('census_examen.dat',sep='\t',colClasses=c("numeric","numeric","factor","factor"
,"factor","factor","numeric","factor","factor","factor","factor","factor","factor","fact
or","factor","factor","factor","numeric","numeric","numeric","factor","factor","factor",
"factor","factor","factor","factor","factor","factor","factor","numeric","factor","facto
r","factor","factor","factor","factor","factor","factor","numeric","factor"))
```

1.1.1. separar training/test

```
# semilla

set.seed(12345)

# 70% / 30%

sample_size <- floor(0.70 * nrow(data))
train_indices <- sample(seq_len(nrow(data)), size = sample_size)
training          <- data[train_indices,]
testing           <- data[-train_indices,]
testing.sin.clase <- testing[, -which(names(testing) %in% c("Clase"))]

dim(testing): 4503   registros

dim(training): 10507    registros
```

1.1.2 ajustar modelo:

```
# sólo con intercept

modelo.vacio <- glm(Clase ~ 1, family = binomial, data = training)

# con todas las variables

modelo.full <- glm(Clase ~ AAGE + ACLSWKR_C + ADTIND_C + ADTOCC_C + AHGA_C + AHRSPAY +
AHSCOL_C + AMARITL_C + AMJIND_C + AMJOCC_C + ARACE_C + AREORGN_C + ASEX_C + AUNMEM_C +
AUNTYPE_C + AWKSTAT_C + CAPGAIN + CAPLOSS + DIVVAL + FILESTAT_C + GRINREG_C + GRINST_C +
HHDFMX_C + HHDREL_C + MIGMTR1_C + MIGMTR3_C + MIGMTR4_C + MIGSAME_C + MIGSUN_C + NOEMP +
PARENT_C + PEFNTVTY_C + PEMNTVTY_C + PENATVTY_C + PRCITSHP_C + SEOTR_C + VETQVA_C +
VETYN_C + WKSWORK, family = binomial, data = training)

# modelo 'introducir', desde el vacío hasta el completo

modelo.fw <- step(modelo.vacio, scope=list(lower=modelo.vacio, upper=modelo.full),
direction="forward")

# el modelo resultante:
# Clase ~ AHGA_C + ADTOCC_C + CAPGAIN + HHDFMX_C + ASEX_C + DIVVAL + WKSWORK + AAGE +
CAPLOSS + NOEMP + ACLSWKR_C + PEMNTVTY_C + AMJOCC_C + SEOTR_C + ADTIND_C + AUNMEM_C +
AHSCOL_C

# las variables significativas de este modelo forward

## ADTOCC_C2   -1.232e+00  2.784e-01  -4.425 9.66e-06 ***
## ADTOCC_C3   -9.795e-01  3.172e-01  -3.088 0.002015 **
## CAPGAIN      1.431e-04  1.742e-05   8.215  < 2e-16 ***
## HHDFMX_C9   -1.745e+00  3.362e-01  -5.189 2.11e-07 ***
## ASEX_CM      1.033e+00  1.402e-01   7.368 1.73e-13 ***
## DIVVAL       1.723e-04  2.240e-05   7.692 1.45e-14 ***
## WKSWORK      4.496e-02  6.638e-03   6.773 1.26e-11 ***
## AAGE         3.014e-02  4.765e-03   6.324 2.54e-10 ***
```

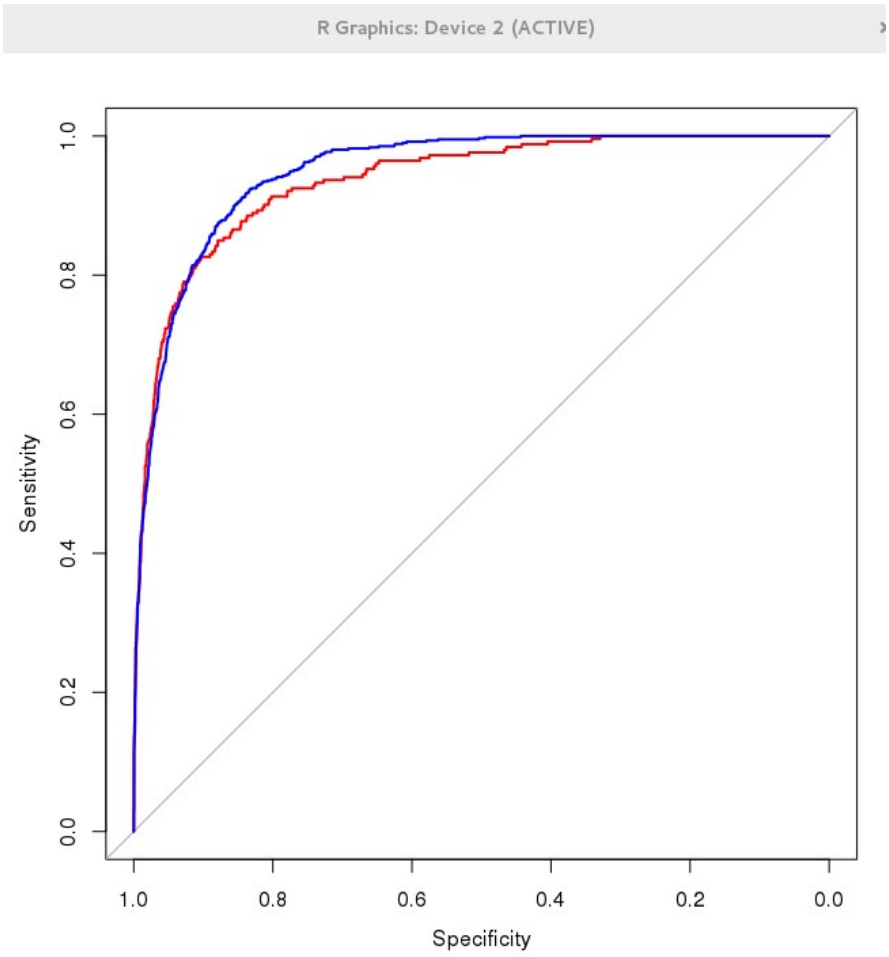
```
## CAPLOSS       6.578e-04  1.131e-04   5.815 6.06e-09 ***
## NOEMP         1.738e-01  3.354e-02   5.181 2.20e-07 ***
## ACLSWKR_C7   -1.348e+00  4.118e-01  -3.274 0.001059 **
## PEMNTVTY_C4 -1.918e+00  5.281e-01  -3.633 0.000280 ***
## AMJOCC_C11    2.581e+00  5.326e-01   4.845 1.26e-06 ***
## AMJOCC_C12    9.471e-01  3.806e-01   2.488 0.012829 *
## AMJOCC_C14    9.803e-01  4.399e-01   2.229 0.025844 *
## AMJOCC_C6     8.896e-01  4.181e-01   2.128 0.033339 *
## AMJOCC_C13    1.025e+00  4.251e-01   2.411 0.015915 *
## AMJOCC_C3     1.128e+00  3.853e-01   2.927 0.003423 **
## AMJOCC_C10    1.125e+00  3.690e-01   3.047 0.002308 **
## AMJOCC_C8     1.582e+00  4.313e-01   3.669 0.000244 ***
## SEOTR_C1      5.515e-01  2.542e-01   2.170 0.030044 *
## SEOTR_C2     -4.128e-01  1.840e-01  -2.243 0.024869 *
## AUNMEM_C9     3.754e-01  1.610e-01   2.333 0.019672 *
```

```
# tratamos de mejorar el modelo anterior recodificando los niveles de las variables
categóricas que tienen betas parecidos (tanto en AMJOCC_C como en ADTOCC_C)

library(car)

# fusionamos niveles 3, 8 y 10
data$AMJOCC_C_030810 <- recode(data$AMJOCC_C,
'1=0;2=0;3=1;4=0;5=0;6=1;7=0;8=1;9=0;10=1;11=0;12=0;13=0;14=0;15=0')

# fusionamos niveles 6, 12 y 14
data$AMJOCC_C_061214 <- recode(data$AMJOCC_C,
'1=0;2=0;3=0;4=0;5=0;6=1;7=0;8=0;9=0;10=0;11=0;12=1;13=0;14=1;15=0')

# fusionamos niveles 2 y 3
data$ADTOCC_C_0203 <- recode(data$ADTOCC_C, '1=0;2=1;3=1;4=0;5=0;6=0')

# y quitamos las variables que no eran significativas en el modelo anterior

modelo.fw.recodificado <- glm(Clase ~ ADTOCC_C_0203 + CAPGAIN + HHDFMX_C + ASEX_C +
DIVVAL + WKSWORK + AAGE + CAPLOSS + NOEMP + ACLSWKR_C + AMJOCC_C_061214 +
AMJOCC_C_030810 + SEOTR_C + + AUNMEM_C, family = binomial, data = training)

# sacamos la curva ROC otra vez... área bajo la curva: 0.9309
# nos quedamos con éste, porque es más simple y la diferencia en desempeño es chica (~
0.005)

# coeficientes (betas) del modelo
round(coef(modelo.fw.recodificado), 2)

Clase = -7.92 – 1.28 * ADTOCC_C_02031 -0.02 * HHDFMX_C(2) + 0.06 * HHDFMX_C(3) -2.01
HHDFMX_C(9) + 1.22 ASEX_CM + 0.05 * WKSWORK + 0.03 * AAGE + 0.22 * NOEMP + 0.01 *
ACLSWKR_C(2) -8.09 * ACLSWKR_C(3) + 0.06 * ACLSWKR_C(4) + 0.89 ACLSWKR_C(5) + 0.42
ACLSWKR_C(6) – 0.92 ACLSWKR_C(7) -11.59 * ACLSWKR_C(8) + 0.13 * ACLSWKR_C(9) + 0.27
AMJOCC_C_0612141 + 1.12 * AMJOCC_C_0308101 + 0.65 * SEOTR_C(1) -0.34 * SEOTR_C(2) – 0.23
* AUNMEM_C(1) + 0.25 * AUNMEM_C(9)
```

1.1.3 curva ROC

```
library(pROC)
prob.pred.testing <- predict(modelo.fw.recodificado, testing.sin.clase, type =
c("response"))
g.testing <- roc(Clase ~ prob.pred.testing, data = testing)
plot(g.testing, col = "red")
prob.pred.training <- predict(modelo.fw.recodificado, type = c("response"))
g.training <- roc(Clase ~ prob.pred.training, data = training)
lines(g.training, col = "blue")
```

```
# Área bajo la curva en testing: 0.9309
```



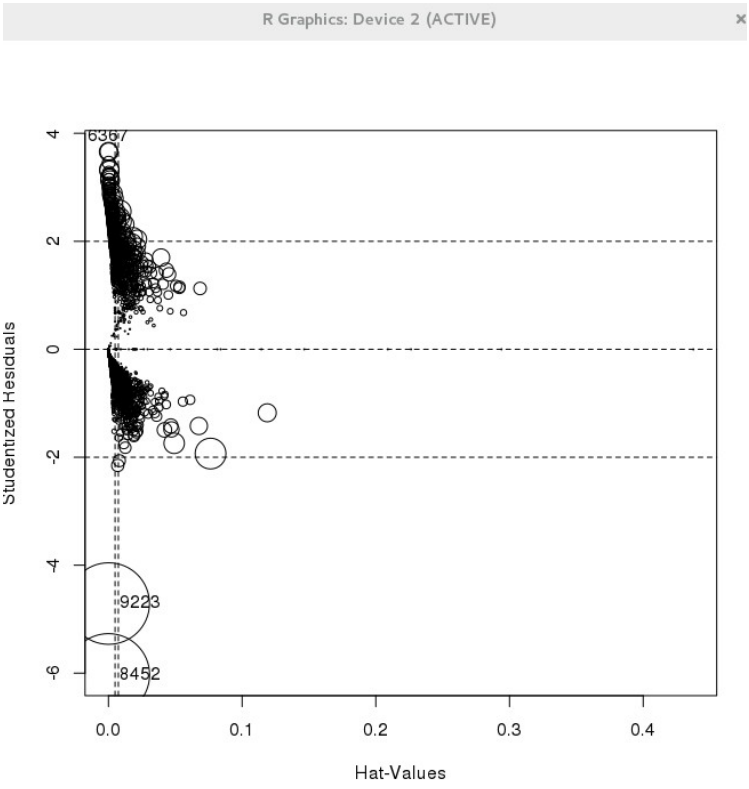
1.1.4 muestras

```
# muestra de 25% al azar de testing

muestra_indices <- sample(seq_len(nrow(testing)), size = 1125)
muestra_testing <- testing[muestra_indices,]
summary(muestra_testing$Clase)

   0    1
1074   51

# hay 51 individuos que ganan más de $50.000, si tomamos una muestra al azar.

# ahora aplicamos el modelo sobre testing:

testing_sin_clase <- testing[,-41]
predicciones <- predict(modelo.fw.recodificado, testing_sin_clase, type = c("response"))
predicciones_02 <- round(predicciones,2)

# ordenamos de menor a mayor, y extraemos el último 25%

predicciones_02_ordenadas_25_porciento <- sort(predicciones_02)[3379:4503]

# calculamos cuántos de los individuos tienen una probabilidad de tener la clase 1 según
el modelo
```

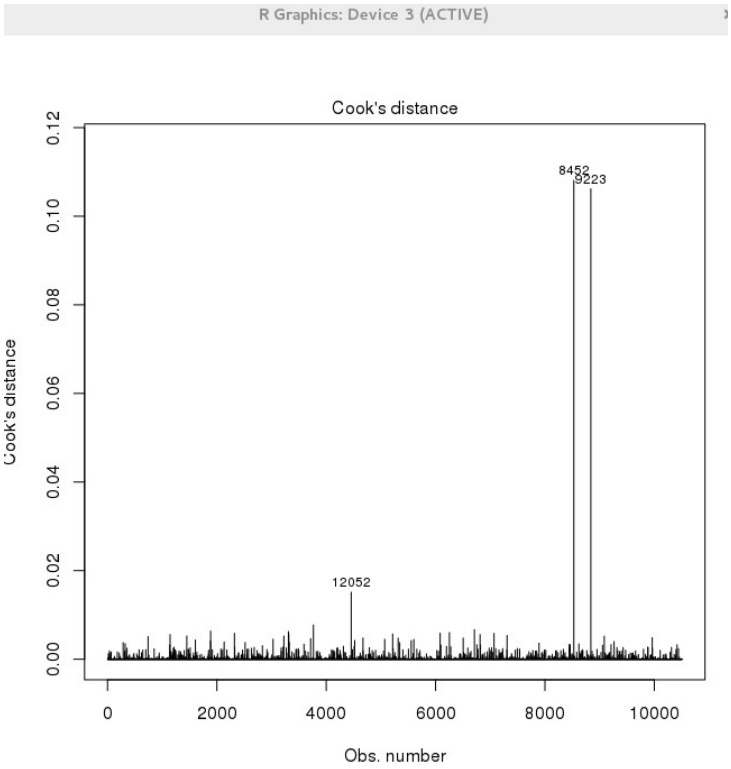
```
length(which(predicciones_02_ordenadas_25_porciento >= 0.5))
# 107

# 107 > 51 => concluimos que conviene aplicar el modelo para detectar a estos individuos
(o por lo menos es mejor que el azar).
```

1.1.5

```
* para una variable continua: NOEMP → exp(beta de NOEMP) → 1.24. Entonces se interpreta
que los odds de ganar +$50.000 aumentan en 24% por cada unidad adicional de NOEMP.

* influyentes por distancia de Cook:
```




```
influencePlot(modelo.fw.recodificado, id.method="identify")


# algunos outliers según el gráfico:

# data[8452,"caso"] → 167842
# data[9223,"caso"] →183334
# data[6367,"caso"] → 126056

cutoff <- 4 / (nrow(data) - length(modelo.fw.recodificado$coefficients) - 2)
plot(modelo.fw.recodificado, which=4, cook.levels=cutoff)
```



(los mismos 2 que antes)

* método 'Forward' utilizado en 1.1.2 empieza con modelo sin predictoras (sólo el intercepto) y va agregando la variable cuyo p-valor al calcular un estadístico (AIC en R) sea menor; así itera hasta que se detiene cuando ya no puede agregar ninguna variable.