Bondad de ajuste

Contents

1 Introduccion **1**

2 Criterio de la matriz de confusión **2**

3 R-cuadrado en regresión logística **2**

4 Contraste de bondad de ajuste **3**

1 Introduccion

Se estima el siguiente modelo de regresión logística:

```
d = read.csv("datos/MichelinNY.csv")
m = glm(InMichelin ~ Food + Decor + Service + Price, data = d, family = binomial)
summary(m)
```

```
##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial,
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3923  -0.6723  -0.3810   0.7169   1.9694
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.19745    2.30896  -4.850 1.24e-06 ***
## Food          0.40485    0.13146   3.080  0.00207 **
## Decor         0.09997    0.08919   1.121  0.26235
## Service      -0.19242    0.12357  -1.557  0.11942
## Price         0.09172    0.03175   2.889  0.00387 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 148.40  on 159  degrees of freedom
## AIC: 158.4
##
## Number of Fisher Scoring iterations: 6
```

El objetivo es analizar como de bueno es el modelo de regresión logística que se ha estimado.

2 Criterio de la matriz de confusión

El método más sencillo es calcular el error de predicción del modelo en la base de datos. Esto se hace con la matriz de confusión.

```
pred_prob = predict(m, newdata = d, type = "response")
n = nrow(d)
pred_y = rep(0, n)
pred_y[pred_prob > 0.5] = 1
# matriz de confusion
(t = table(d$InMichelin, pred_y))
```

```
##    pred_y
##      0  1
##   0 81  9
##   1 20 54
```

Por tanto, se han predicho bien $81 + 54 = 135$ datos de un todal de 164. Se han mal bien $9 + 20 = 29$ datos de un todal de 164. El error del modelo es $29 / 164 = 17.68\%$.

Cuando el objetivo de principal de la regresión logística sea la predicción, la bondad del modelo se puede calcular construyendo la matriz de confusión en un test set:

```
set.seed(123)
pos_train = sample(1:n, round(0.8*n), replace = F)
train = d[pos_train,]
test = d[-pos_train,]
```

```
m1 = glm(InMichelin ~ Food + Decor + Service + Price, data = train, family = binomial)
test_prob = predict(m1, newdata = test, type = "response")
n_test = nrow(test)
pred_y = rep(0, n_test)
pred_y[test_prob > 0.5] = 1
# matriz de confusion
(t = table(test$InMichelin, pred_y))
```

```
##    pred_y
##      0  1
##   0 19  0
##   1  5  9
```

3 R-cuadrado en regresión logística

Otra manera de calcular la bondad del modelo es definir un R^2 de manera similar a como se hizo en regresión lineal. Se han propuesto muchas formas de definir este R^2, pero quizá la más usada es:

$$R^2 = 1 - \frac{D_1}{D_0}$$

donde D es la desviación del modelo (deviance en inglés). Se calcula como el doble de la verosimilitud del modelo calculada en los parámetros estimados (en valor absoluto):

$$D = |2logL(\hat{\beta})|$$

$$logL(\hat{\beta}) = \sum_{i=1}^{n}(y_i log\hat{\pi}_i + (1 - y_i)log(1 - \hat{\pi}_i))$$

$$\hat{\pi}_i = \frac{exp(x_i^T \hat{\beta})}{1 + exp(x_i^T \hat{\beta})}$$

Se definen dos desviaciones:

- D1: la desviación del modelo analizado.
- D0: la desviación del modelo en el que solo se estima β_0.

```
source("logit_funciones.R")
(D1 = abs(2*logL(coef(m),d$InMichelin,model.matrix(m))) )
```

```
## [1] 148.3969
```

```
m0 = glm(InMichelin ~ 1, data = d, family = binomial)
summary(m0)
```

```
##
## Call:
## glm(formula = InMichelin ~ 1, family = binomial, data = d)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.095  -1.095  -1.095   1.262   1.262
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1957     0.1569  -1.247    0.212
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 225.79  on 163  degrees of freedom
## AIC: 227.79
##
## Number of Fisher Scoring iterations: 3
```

```
(D0 = abs(2*logL(coef(m0),d$InMichelin,model.matrix(m0))) )
```

```
## [1] 225.7888
```

```
(R2 = 1 - D1/D0)
```

```
## [1] 0.3427622
```

Si $R^2 \approx 1$ el modelo se ajusta muy bien a los datos.

4 Contraste de bondad de ajuste

Se puede resolver el siguiente contraste:

- H0: el modelo estimado es adecuado.
- H1: el modelo estimado no es adecuado.

El estadístico del contraste es

$$G = D_0 - D_1 \sim \chi^2_{n-k-1}$$

Para valores grandes del estadístico, quiere decir que la verosimilitud de ambos modelos es muy diferente, luego el modelo es adecuado. Para valores pequelos de G, ambos modelos son muy parecidos, luego los regresores no describen bien la variable respuesta.

En este caso:

```
(G = D0 - D1)
```

```
## [1] 77.39187
```

```
n = nrow(d)
k = length(coef(m)) - 1
(pvalor = 1-pchisq(G, n - k - 1))
```

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

Luego el modelo es muy adecuado.