Bondad de ajuste

Contents

1 Introduccion

Se estima el siguiente modelo de regresión logística:

```
d = read.csv("datos/MichelinNY.csv")
m1 = glm(InMichelin ~ Food + Decor + Service + Price, data = d, family = binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial,
##     data = d)
##
## 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(m1, 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 predicho mal 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,]
```

```
m2 = glm(InMichelin ~ Food + Decor + Service + Price, data = train, family = binomial)
test_prob = predict(m2, 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 define 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 (*residual deviance*).
- D0: la desviación del modelo en el que solo se estima β_0 (*null deviance*).

```
source("funciones/logit_funciones.R")
(D1 = abs(2*logit_logL(coef(m1),d$InMichelin,model.matrix(m1))) )
```

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

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

```
##
## Call:
## glm(formula = InMichelin ~ 1, family = binomial, data = d)
##
## 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*logit_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, y $R^2 \approx 0$ implica un mal ajuste. Es decir, $R^2 \approx 0$ significa que la verosimilitud de ambos modelos es muy parecida, luego $\beta_1 \approx \beta_2 \approx \beta_k \approx 0$.

4 Contraste para un grupo de coeficientes

Supongamos que tenemos dos modelos:

$$\pi_i = \frac{exp(x_i^T \beta)}{1 + exp(x_i^T \beta)}$$

$$\pi_{Ai} = \frac{exp(x_{Ai}^T \beta_A)}{1 + exp(x_{Ai}^T \beta_A)}$$

donde β_A es un subconjunto de β. Supongamos que $dim(\beta_A) = m$ y $dim(\beta) = k$, con $m < k$. Si β_B representa los parámetros que están en β pero no están en β_A, se puede resolver el siguiente contraste:

$$H_0 : \beta_B = 0, \quad H_1 : \beta_B \neq 0$$

Para resolver el contraste se utiliza el estadístico

$$G = D_A - D_1$$

donde D_1 es la desviación del modelo con parámetros β y D_A es la desviación del modelo con parámetros β_A. Se puede demostrar que:

- $D_1 \sim \chi^2_{n-k}$.
- $D_A \sim \chi^2_{n-m}$.

Por otro lado se cumple que la suma (o resta) de dos χ^2 también tiene distribución χ^2:

$$\chi^2_a + \chi^2_b \sim \chi^2_{a+b}$$

por lo que la distribución del estadístico G es:

$$G = D_A - D_1 \sim \chi^2_{(n-m)-(n-k)} = \chi^2_{k-m}$$

Si la hipótesis nula es cierta, es decir, $\beta_B = 0$, se cumpliría que $D_1 \approx D_A$, y el estadístico G debería tomar valores pequeños. Por contra, si $\beta_B \neq 0$, el estadístico G debería tomar valores altos. Por tanto:

- Si $G < \chi^2_\alpha$ no se rechaza la hipótesis nula.
- Si $G \geq \chi^2_\alpha$ se rechaza la hipótesis nula.

Por ejemplo, queremos resolver el contraste con hipótesis nula $\beta_1 = \beta_2 = 0$:

```
mA = glm(InMichelin ~ Service + Price, data = d, family = binomial)
(DA = abs(2*logit_logL(coef(mA),d$InMichelin,model.matrix(mA))) )
```

```
## [1] 161.0588
```

```
(G = DA - D1)
```

```
## [1] 12.66182
```

```
# valor crítico del contraste
k = length(coef(m1))
m = length(coef(mA))
qchisq(0.95, df = k-m)
```

```
## [1] 5.991465
```

Luego se rechaza la hipótesis nula.

5 Contraste de bondad de ajuste

Utilizando el contraste anterior entre el modelo con todos los regresores y el modelo con solo β_0 se puede analizar la bondad del modelo. Es decir, se puede contrastar:

- H0: el modelo estimado NO es adecuado ($\beta_1 = \beta_2 = \cdots = \beta_k = 0$)
- H1: el modelo estimado es adecuado.

El estadístico del contraste es

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

En este caso:

```
D0
```

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

```
D1
```

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

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

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

```
k = length(coef(m1))
(pvalor = 1-pchisq(G, k-1))
```

```
## [1] 6.661338e-16
```

Luego el modelo es muy adecuado.

Como vemos, las desviaciones necesarias para resolver este contraste se indican en el *summary()*.

```
summary(m1)
```

```
##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial,
##     data = d)
##
## 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
```