Regresión Logística para Clasificar Spam

Javier Adanaqué

24 de Abril, 2017

Resumen

Se elabora un modelo de Regresión Logística que nos permita clasificar emails como spam.

1. Introducción

El objetivo del presente documento es elaborar un primer modelo de Regresión Logística Simple que nos permita clasificar los emails como spam o no-spam usando lo aprendido en clase (introductoria). Para ello se está usando una base de datos de emails con diferentes características sobre las mismas.

Primero se realizará una exploración rápida de algunas variables y luego se contruirá el modelo, mostrando la mayoría de las veces el código usado para llegar a ellos.

Al ser un modelo de Regresión Logística **Simple** nos concentraremos en conocer si una sola variable nos ayuda a predecir si un mail es spam o no. Para ello se usará, en este primero modelo, nuestra intuición y conocimiento sobre emails para determinar qué variable nos podría predecir mejor si es spam o no.

2. Data

Para el análisis se está usando la data `SPAM.txt`, ubicada entre los datasets distribuidos para la clase. Esta es una base de 4601 e-mails (correos electrónicos) y sus características:

```
# Se lee la data con Python. El resto del análisis se trabaja con R.
import pandas as pd

spam_dat = pd.read_table("data/SPAM.txt")
print(spam_dat.iloc[:6, [0, 1, 2, 52, 55, 56, 57]])
```

make	address	all	charDollar	capitalLong	capitalTotal	tipo
0.00	0.64	0.64	0.000	61	278	spam
0.21	0.28	0.50	0.180	101	1028	spam
0.06	0.00	0.71	0.184	485	2259	spam
0.00	0.00	0.00	0.000	40	191	spam
0.00	0.00	0.00	0.000	40	191	spam
0.00	0.00	0.00	0.000	15	54	spam

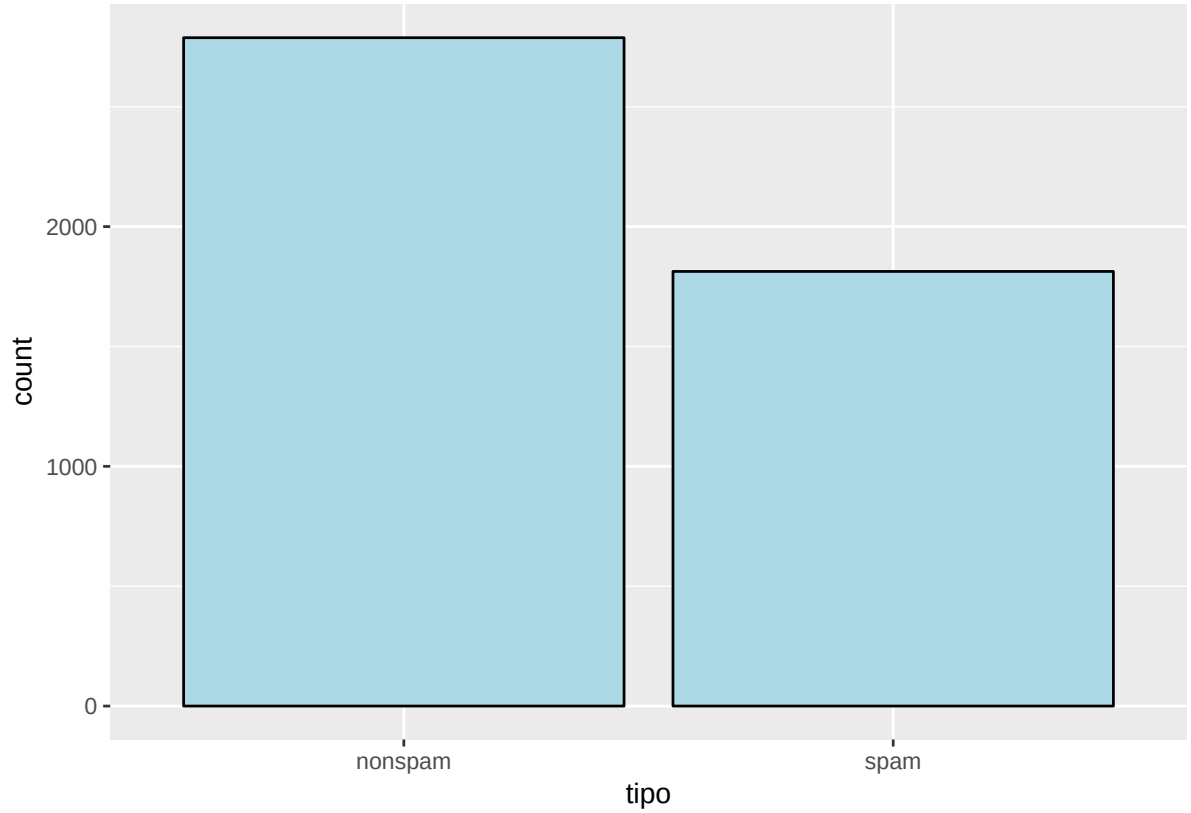
Como se puede observar en las primeras seis observaciones (y algunas columnas), nos muestra diferentes características sobre los e-mails, incluido el tipo de email: "spam" o "nonspam".

Descripción de variables:

- Primeras 48 variables: Frecuencia (relativa) del nombre de la variable en el correo electrónico. Si el nombre de la variable empieza con 'num (e.g., `num857`), entonces indica la frecuencia del número correspondiente (e.g., 857).
- Las variables 49-54 indican la frecuencia de los caracteres ';', '(', '[', '!', '$', and '#'.
- Las variables 55-57 contienen características relacionadas a las letras mayúsculas en el mail.
- `tipo`: Indica si el e-mail es "spam" o "nonspam".

3. Análisis exploratorio

A continuación exploraremos algunas variables que parecen relevantes para determinar si un e-mail es spam o no. Empezaremos explorando nuestra variable que queremos explicar, `tipo`:



Vemos que hay un buen número de de e-mails clasificados como "spam" y "nonspam", lo cual ayuda para la elaboración del modelo (más difícil sería si tuviéramos mucho de uno y muy poco del otro).

Ahora exploramos las variables que nos pueden ayudar a predecir si un e-mail es spam. Nuestra intuición nos dice que una gran mayoría de spams buscan ofrecerte algo tentador,

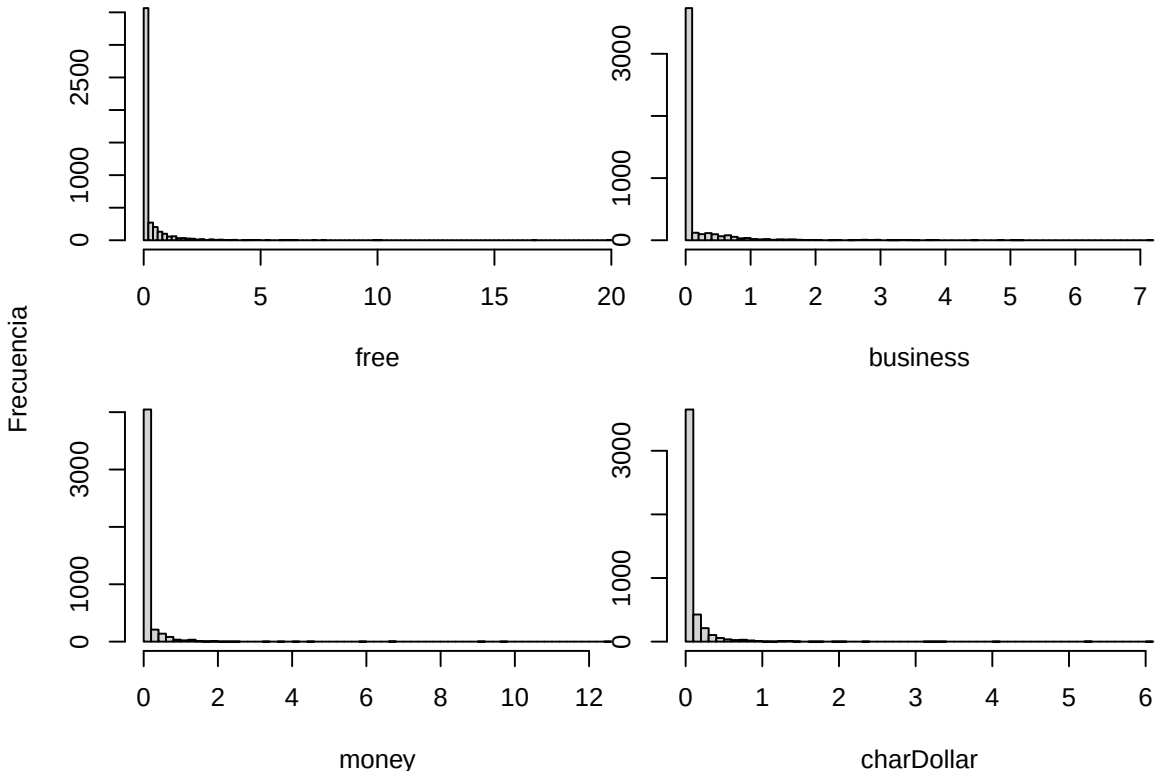
de manera que lo leas y abras el link que se encuentre adentro (de haber alguno). Entonces, nos concentraremos en alguna variable que nos indique la frecuencia de términos/caracteres como estos: “dinero”, “gratis”, “$”, “negocio” y similares;

```
names(spam_dat)
```

```
##  [1] "make"              "address"           "all"
##  [4] "num3d"             "our"               "over"
##  [7] "remove"            "internet"          "order"
## [10] "mail"              "receive"           "will"
## [13] "people"            "report"            "addresses"
## [16] "free"              "business"          "email"
## [19] "you"               "credit"            "your"
## [22] "font"              "num000"            "money"
## [25] "hp"                "hpl"               "george"
## [28] "num650"            "lab"               "labs"
## [31] "telnet"            "num857"            "data"
## [34] "num415"            "num85"             "technology"
## [37] "num1999"           "parts"             "pm"
## [40] "direct"            "cs"                "meeting"
## [43] "original"          "project"           "re"
## [46] "edu"               "table"             "conference"
## [49] "charSemicolon"     "charRoundbracket"  "charSquarebracket"
## [52] "charExclamation"   "charDollar"        "charHash"
## [55] "capitalAve"        "capitalLong"       "capitalTotal"
## [58] "tipo"
```

Exploremos las variables `free`, `business`, `money` y `charDollar` (caracter: $), que parecen relevantes:

```
par(mfrow = c(2, 2),
    mar = c(3.9, 0, 1, 1.2),
    oma = c(1, 3.9, 0, 0))
for (i in c("free", "business", "money", "charDollar")) {
    hist(spam_dat[, i], breaks = 80, col = "lightgray",
         xlab = i, main = NULL)
}
title(ylab = "Frecuencia", outer = TRUE, line = 3)
```




```
par(mfrow = c(1, 1),
    mar = c(5.1, 4.1, 4.1, 2.1),
    oma = rep(0, 4))
```

Observamos que las 4 variables se encuentran altamente sesgadas. Esto es entendible, dado que en un correo no se suele repetir una palabra/caracter demasiadas veces; por eso es que, en las 4 variables seleccionadas, la mayoría de observaciones se encuentran entre 0 y 2 (por ciento), salvo unos cuantos outliers.

```
summary(spam_dat[, c("free", "business", "money", "charDollar")])
```

```
##       free            business          money            charDollar      
##  Min.   : 0.0000   Min.   :0.0000   Min.   : 0.00000   Min.   :0.00000  
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:0.00000  
##  Median : 0.0000   Median :0.0000   Median : 0.00000   Median :0.00000  
##  Mean   : 0.2488   Mean   :0.1426   Mean   : 0.09427   Mean   :0.07581  
##  3rd Qu.: 0.1000   3rd Qu.:0.0000   3rd Qu.: 0.00000   3rd Qu.:0.05200  
##  Max.   :20.0000   Max.   :7.1400   Max.   :12.50000   Max.   :6.00300
```

De ahora en adelante, continuaremos el análisis usando `charDollar` como variable explicativa. Esto, porque es una variable que definitivamente se repite bastante en los correos spam (ver siguiente gráfico) y porque es una variable cuyo rango no muy extenso (entre 0 y 6, como se puede observar en la tabla anterior).

```
boxplot(charDollar ~ tipo, data = spam_dat, ylim = c(0, .8),
        ylab = "Frecuencia del caracter $ en e-mail (%)", col="lightgray")
```

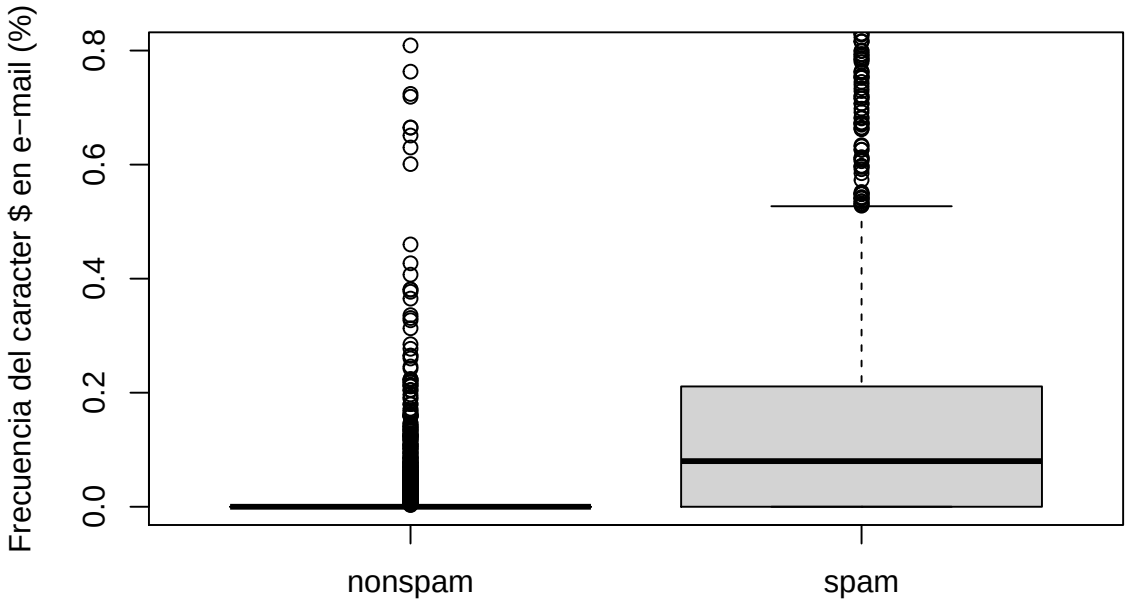


Figura 1: Diagrama de Cajas de 'charDollar' según 'tipo' de e-mail. Se limitó el eje vertical (zoom) para mayor claridad; de lo contrario, los outliers llegarían hasta 6.003.

4. Modelamiento

Como se indica al inicio, para el modelo de clasificación se usará Regresión Logística Simple, el cual vamos a especificar de la siguiente manera:

$$\ln\left(\frac{p}{1-p}\right) = \alpha + \beta * charDollar$$

Ahora se estiman los parámetros y determina la significancia de los mismos:

```
spam_dat$tipo_num <- ifelse(spam_dat$tipo == "spam", 1, 0)
modelo_spam <- glm(tipo_num ~ charDollar, family = binomial(), data = spam_dat)
summary(modelo_spam)
```

```
##
## Call:
## glm(formula = tipo_num ~ charDollar, family = binomial(), data = spam_dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.5494  -0.7650  -0.7650   0.7259   1.6563
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.07909    0.03865  -27.92   <2e-16 ***
## charDollar  14.51198    0.61254   23.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6170.2  on 4600  degrees of freedom
## Residual deviance: 4843.1  on 4599  degrees of freedom
## AIC: 4847.1
##
## Number of Fisher Scoring iterations: 7
```

Por el p-value, y el tamaño del coeficiente, se puede observar que `charDollar` es altamente significativo. El intercepto también es significativo con un p-value de $< 2e^{-}16$. Esto nos da claridad sobre la significancia del modelo, pero de todas maneras, en este primer modelo, se hace una prueba más para saber qué tan efectivo es el modelo clasificando, con un cutoff de 0.5 (50 % de probabilidad).

```
tab_clasif <- table(spam_dat$tipo,
                    ifelse(predict(modelo_spam, type = "response")>0.5, "spam", "nons
tab_clasif <- as.data.frame.matrix(tab_clasif)
tab_clasif[, "%correctos"] <- c(tab_clasif[1, 1]/sum(tab_clasif[1, ]),
                                tab_clasif[2, 2]/sum(tab_clasif[2, ]))
tab_clasif[, "%correctos"] <- paste(format(tab_clasif[, "%correctos"]*100, digits = 2
tab_clasif
```

	nonspam	spam	%correctos
nonspam	2680	108	96.13 %
spam	892	921	50.80 %

- Porcentaje de correctos (Accuracy): 78.27 %
- Sensibilidad: 50.8 %
- Especificidad: 96.13 %

Como vemos, en general nuestro modelo es bueno; sin embargo, para nuestro objetivo, clasificar spam, no es tan bueno dado que sólo clasifica correctamente alrededor de un 50 % de los spams.

Finalmente, veamos algunas observaciones y sus estimados:

```
estimados <- data.frame(tipo_observado = spam_dat$tipo,
        prob_estimada = predict(modelo_spam, type = "response"),
        tipo_estimado = ifelse(predict(modelo_spam, type = "response")>0.5, "spam", "
estimados[c(1, 5, 50, 100, 1000, 2000, 3000, 4000), ]
```

	tipo_observado	prob_estimada	tipo_estimado
1	spam	0.2536777	nonspam
5	spam	0.2536777	nonspam
50	spam	0.9053951	spam
100	spam	0.9985984	spam
1000	spam	0.7810588	spam
2000	nonspam	0.2536777	nonspam
3000	nonspam	0.2536777	nonspam
4000	nonspam	0.3916061	nonspam

5. Conclusiones

El modelo, a nivel general, clasifica correctamente un 78 % de las observaciones. Sin embargo, no es bueno clasificando los spams, con sólo un 50 % de efectividad en este caso; resultado esperado, dado que la variable `charDollar` se encuentra demasiado sesgada positivamente, con una gran porción de datos alrededor de 0. Además, para clasificar e-mails no es suficiente una sola variable.

Queda pendiente probar con más variables, hacer más tests y dividir la data en "train" y "test" para mejor validación del modelo.