Análisis de Regresión (2021-3)

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Especialización en Estadística Aplicada

Prof. Sébastien Lozano Forero (slozanof@libertadores.edu.co)

Introducción a la Regresión Logística

Tabla de contenidos

Modelo de Regresión Logística

Suponga alguna de las siguientes situaciones

- Un sujeto operado se infecta o no durante cierto lapso post-operatorio.
- Un bebé nace con o sin malformación congénita.
- Un paciente hospitalizado muere o no antes del alta.
- A los tres meses de vida, un niño ha dejado de lactar o aún se mantiene alimentándose con leche materna.
- Un año después de una intervención quirúrgica, se ha resuelto o no el problema que la origino´.
- Después de un tratamiento de quimioterapia en un paciente con cáncer de pulmón se observa alguno de los siguientes resultados sobre la enfermedad: aumento, no cambio, remisión parcial, remisión completa.

El modelo de regresión lineal tradicional utiliza la linealidad para describir la relación entre el valor esperado de la variable respuesta y un conjunto de variables explicativas asumiendo que la distribución de la variable respuesta es normal. Los modelos lineales generalizados (MLG) extiende el modelo de regresión lineal tradicional con el fin de poder caracterizar variables respuestas con distribuciones no normales y funciones no lineales de la media. Los MLG tienen tres componentes:

1. Componente aleatorio:

Este componente especifica la variable de respuesta y y su distribución de probabilidad.

Nota 1: La distribución de la variable respuesta debe ser un miembro de la familia exponencial de distribuciones. Se dice que una distribución con densidad de probabilidad o función de masa para una variable aleatoria y pertenece a la familia exponencial de distribuciones sí, y solo sí, puede expresarse de la forma general dada por:

$$f(y;\theta,\phi) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right]$$

donde

- θ es llamado el parámetro natural o canónico y representa la localización.
- ϕ es llamado el parámetro de dispersión y representa la escala.
- $b(\theta)$ se define como la función cumulante y es importante porque relaciona el parámetro canónico con la media y varianza de y.

Nota 2: Las observaciones sobre esa distribución $y = (y_1, y_2, \cdots, y_n)^\top$ se consideran independientes.

A continuación, los principales elementos para tres de las distribuciones que pueden modelarse con GLMs

	Regresión Lineal	**Regresión Poisson**	**Regresión Logística**
$Y \mid \mathbf{X} = \mathbf{x}$	$N(\mu(\mathbf{x}), \sigma^2)$	$\text{Pois}(\lambda(\mathbf{x}))$	$\text{Bern}(p(\mathbf{x}))$
Nombre de la Distribución	Normal	Poisson	Bernoulli (Binomial)
$\text{E}[Y \mid \mathbf{X} = \mathbf{x}]$	$\mu(\mathbf{x})$	$\lambda(\mathbf{x})$	$p(\mathbf{x})$

	Regresión Lineal	Regresión Poisson	Regresión Logística
Soporte	Real: $(-\infty, \infty)$	Entero: $0, 1, 2, \ldots$	Entero: $0, 1$
Uso	Data Numérica	Data de Conteo (Entero)	Data binaria
Nombre la ligación	Identity	Log	Logit
Link Function	$\eta(\mathbf{x}) = \mu(\mathbf{x})$	$\eta(\mathbf{x}) = \log(\lambda(\mathbf{x}))$	$\eta(\mathbf{x}) = \log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right)$
Mean Function	$\mu(\mathbf{x}) = \eta(\mathbf{x})$	$\lambda(\mathbf{x}) = e^{\eta(\mathbf{x})}$	$p(\mathbf{x}) = \frac{1}{1+e^{-\eta(\mathbf{x})}}$

2. Componente sistemático o predictor lineal:

El componente sistemático o predictor lineal es una combinación lineal del conjunto de p covariables identificadas por el analista que asocia el efecto estas variables auxiliares sobre la media de la variable respuesta.

Para un vector de parámetros $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^\top$ desconocidos (a ser estimados posteriormente) y una matriz de información X que contiene todos los n (Nota: $p < n$) valores observados de un conjunto de p variables explicativas, el predictor lineal tiene la forma $X\beta$.

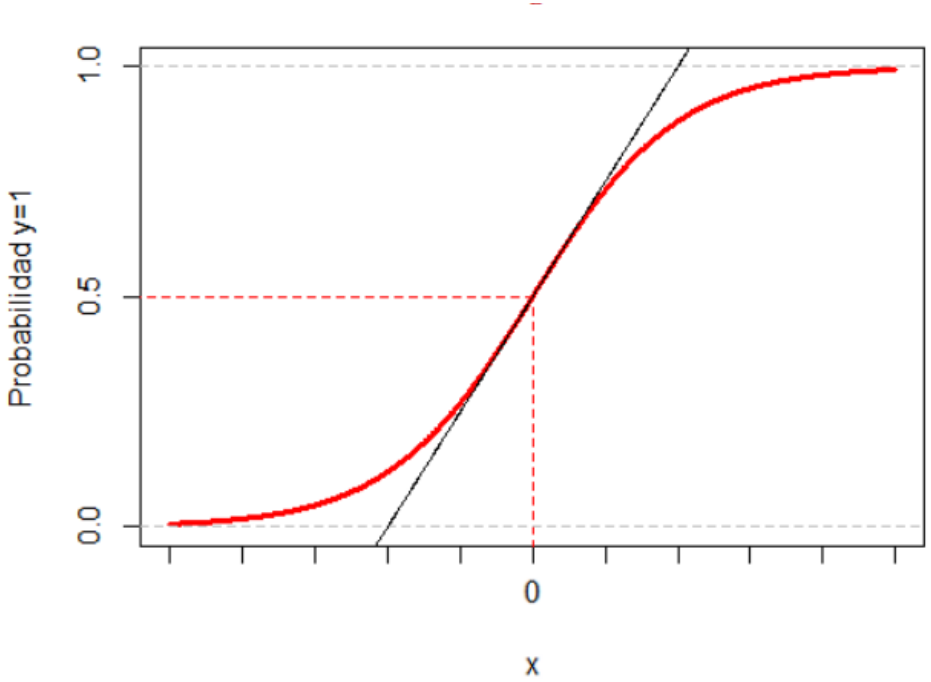
La recta de regresión de un modelo de regresión lineal se extiende de forma ilimitada entre $(-\infty, \infty)$. Si bien los valores de la recta de regresión se interpretan en el rango de valores de x observados en la muestra y tienen un sentido interpretativo, descartando valores de predicción imposibles a partir de los datos estudiados. No obstante, también puede suceder que a pesar de considerar el rango de valores de la muestra los valores pronosticados sean valores imposibles. Es el caso que se puede dar cuando consideramos en la regresión lineal variables dicotómicas de la variable dependiente, codificadas con 0 y 1, donde los valores predichos pueden ser inferiores a 0 y superiores a 1, fuera del rango definido por la variable dependiente. La regresión logística resuelve este tipo de problema usando una función no lineal como es la función logística. Con esta función se pueden efectuar predicciones comprendidas entre un mínimo y un máximo. El modelo de regresión logística es un modelo no lineal que utiliza el método de máxima verosimilitud, un procedimiento iterativo que en fases sucesivas ajusta el modelo.

En últimas, este problema del "idioma" que hablan los componentes sistemático y aleatorio se resuelve con

3. Una función de enlace

En una función g que aplicada a cada componente de $E(y)$ lo relaciona con el predictor lineal, es decir, $g(E(y)) = X\beta$. Usualmente se denota con el símbolo η.

En otras palabras, la función de enlace es una transformación de la media de la variable respuesta de modo que los efectos de las covariables sean aditivos y las restricciones sobre los datos se mantengan



De esta manera, se la función sigmoide (sigmoid function) es útil para poder "traducir" la información de un componente a otro. Es decir, es una excelente opción (ojo, no es la única) a ser una función de ligación

$$f(x) = \frac{1}{1+e^{-x}}$$

Así, el **modelo de regresión logística** se gesta del interés de poder predecir el comportamiento de las variables $y_1, \cdots, y_n \sim Br(p)$, con

$$\begin{cases} p = P(y_t = 1), t = 1, 2 \cdots, n. \\ 1-p = P(y_t = 0), t = 1, 2 \cdots, n., \end{cases}$$

a partir del conocimiento consignado en las variables exógenas. De esta manera, el modelo está dado por

$$p = P(y_t | x_1, \cdots, x_p) = \frac{1}{1+\exp(-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \cdots - \beta_p x_p)}$$

o de forma equivalente, como

$$\log\left(\frac{p}{1-p}\right) = \log\left(\frac{P(y_t|x_1,\cdots,x_p)}{1-P(y_t|x_1,\cdots,x_p)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

Donde:

- $\beta_0, \beta_1, \cdots, \beta_p$ son parámetros poblacionales a ser estimados.
- y representa la *variable respuesta*
- $x_1, \cdots x_p$ representan las *variables predictoras*

In [ ]:
```
library(tidyverse)
library(ISLR)
datos <- Default

# Se recodifican los niveles No, Yes a 1 y 0
datos <- datos %>%
         select(default, balance) %>%
         mutate(default = recode(default,
                                 "No"  = 0,
                                 "Yes" = 1))
head(datos)
```

In [ ]:
```
# Ajuste de un modelo lineal por mínimos cuadrados.
modelo_lineal <- lm(default ~ balance, data = datos)

# Representación gráfica del modelo.
ggplot(data = datos, aes(x = balance, y = default)) +
  geom_point(aes(color = as.factor(default)), shape = 1) +
  geom_smooth(method = "lm", color = "gray20", se = FALSE) +
  theme_bw()  +
  labs(title = "Regresión lineal por mínimos cuadrados",
       y = "Probabilidad default") +
  theme(legend.position = "none")
```

y peor aún, para predecir:

In [ ]:
```
predict(object = modelo_lineal, newdata = data.frame(balance = 10000))
```

Resultados Generales

Proceso de estimación de parámetros

Podemos escribir el modelo de regresión muestral correspondiente al modelo como

$$\log\left(\frac{p}{1-p}\right) = \log\left(\frac{P(y_t|x_1,\cdots,x_p)}{1-P(y_t|x_1,\cdots,x_p)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

En notación matricial, este modelo de regresión muestral se representa como

$$\log\left(\frac{p}{1-p}\right) = \log\left(\frac{P(Y|X)}{1-P(Y|X)}\right) = X\beta$$

donde

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

Deseamos encontrar el vector de estimadores de mínimos cuadrados, β, que minimiza la función de log-verosimilitud

$$L(\beta) = \sum_{i=1}^{p} y_i \ln(p_i) + (1-y_i)\ln(1-p_i)$$

El resultado de tal proceso de estimación no tiene forma cerrada (es decir, se puede despegar "a manos"), por tanto, son necesarios algunos métodos númericos para hallar tales estimativas (típicamente son esquemas iterativos al estilo Newton-Rhapson).

pruebas de hipótesis

En general, va a ser de nuestro interés obtener evidencia empírica de la validez estadística de los parámetros que indexan el modelo. Para esto, se hace necesario introducir pruebas de hipótesis para los parámetros.

Considere, la prueba de las hipótesis para β_i:

$$H_0 : \beta_i = \beta_{i0} (\beta_{i0} \text{conocido})$$
$$H_1 : \beta_i \neq \beta_{i0}$$

que tiene como estadístico de prueba:

$$Z_{Est} = \frac{\hat{\beta}_1 - \beta_{10}}{s\{\hat{\beta}_1\}} \sim N(0,1)$$

y la regla de decisión para el nivel de significancia α es p-valor $< \alpha$ donde p-valor $= 2P(Z > |Z_{Est}|)$ con $Z \sim N(0,1)$.

Test de razón de verosimilitud

Considere el siguiente modelo saturado,

$$\log\left(\frac{p(\mathbf{x_i})}{1-p(\mathbf{x_i})}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{(p-1)} x_{i(p-1)} + \epsilon_i$$

El modelo tiene $p-1$ predictores, para un total of p parámetros. Notamos como $\hat{\beta}_{\text{Full}}$ al vector de estimaciones de máxima verosimilitud

Ahora, considere el siguiente modelo nulo

$$\log\left(\frac{p(\mathbf{x_i})}{1-p(\mathbf{x_i})}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{(q-1)} x_{i(q-1)} + \epsilon_i$$

donde $q < p$. Este modelo $q-1$ predictors, en total q parámetros. Nuevamente, sea $\hat{\beta}_{\text{Full}}$ el vector de estimaciones de máxima verosimilitud

La diferencia entre estos dos modelos se puede estudiar a través de una prueba de hipótesis

$$H_0 : \beta_q = \beta_{q+1} = \cdots = \beta_{p-1} = 0.$$

Lo que implica que el modelo reducido está anidado en el modelo saturado. Así se define la estadística, D,

$$D = -2\log\left(\frac{L(\hat{\boldsymbol{\beta}}_{\text{Null}})}{L(\hat{\boldsymbol{\beta}}_{\text{Full}})}\right) = 2\log\left(\frac{L(\hat{\boldsymbol{\beta}}_{\text{Full}})}{L(\hat{\boldsymbol{\beta}}_{\text{Null}})}\right) = 2\left(\ell(\hat{\beta}_{\text{Full}}) - \ell(\hat{\beta}_{\text{Null}})\right)$$

donde L denota la función de verosimilitud y ℓ denota la función de log-verosimilitud. Para tamaños de muestra grande (más de 50), este estadístico de test tiene distribución chi cuadrado.

$$D \overset{\text{approx}}{\sim} \chi^2_k$$

con $k = p - q$, la diferencia en el número de parámetros en los dos modelos.

Este test, frecuentemente referido como **test de razón de verosimilitud**, es análogo a ANOVA (F) para regresión logística. Curiosamente, para implementar este test se usa la función `anova()` de `R`.

Razón de chances (odds ratio)

En la regresión lineal simple, se modela el valor de la variable dependiente y en función del valor de las variables independientes X. Sin embargo, en la regresión logística, se modela la probabilidad de que la variable respuesta y pertenezca al nivel de referencia 1 en función del valor que adquieran los predictores, mediante el uso de LOG of ODDs.

Supónga que la probabilidad de que un evento sea verdadero es de 0.8, por lo que la probabilidad de evento falso es de 1 - 0.8 = 0.2. Los ODDs o razón de probabilidad se definen como el ratio entre la probabilidad de evento verdadero y la probabilidad de evento falso pq. En este caso los ODDs de verdadero son 0.8 / 0.2 = 4, lo que equivale a decir que se esperan 4 eventos verdaderos por cada evento falso.

La trasformación de probabilidades a ODDs es monótona, si la probabilidad aumenta también lo hacen los ODDs, y viceversa. El rango de valores que pueden tomar los ODDs es de $[0, \infty]$. Dado que el valor de una probabilidad está acotado entre $[0, 1]$ se recurre a una trasformación log (existen otras) que consiste en el logaritmo natural de los ODDs. Esto permite convertir el rango de probabilidad previamente limitado a $[0, 1]$ a $[-\infty, +\infty]$.

p	odds	Log(odds)

0.001|0.001001| -6.906755| 0.01| 0.010101| -4.59512| 0.2| 0.25| -1.386294| 0.3| 0.4285714| -0.8472978| 0.4| 0.6666667| -0.4054651| 0.5| 1| 0| 0.6| 1.5| 0.4054651| 0.7| 2.333333| 0.8472978| 0.8| 4| 1.386294| 0.9| 9| 2.197225| 0.999| 999| 6.906755| 0.9999| 9999| 9.21024|

Los ODDs y el logaritmo de ODDs cumplen que:

- Si p(verdadero) = p(falso), entonces odds(verdadero) = 1
- Si p(verdadero) < p(falso), entonces odds(verdadero) < 1
- Si p(verdadero) > p(falso), entonces odds(verdadero) > 1
- A diferencia de la probabilidad que no puede exceder el 1, los ODDs no tienen límite superior.
- Si odds(verdadero) = 1, entonces logit(p) = 0
- Si odds(verdadero) < 1, entonces logit(p) < 0
- Si odds(verdadero) > 1, entonces logit(p) > 0
- La transformación logit no existe para p = 0

Partición de la muestra (in/out)

Los datos de entrenamiento o “training data” son los datos que se usan para *entrenar* (en una perspectiva más amplia, *aprender*) un modelo. La calidad del modelo de regresión logística va a ser directamente proporcional a la calidad de los datos. Por ello las labores de limpieza, depuración o “data wrangling” deberán ser importantes

Los datos de prueba, validación o “testing data“ son los datos que para comprobar si el modelo que hemos generado a partir de los datos de entrenamiento “funciona”. Es decir, si las respuestas predichas por el modelo para un caso totalmente nuevo son acertadas o no.

Es importante que el conjunto de datos de prueba tenga un volumen suficiente como para generar resultados estadísticamente significativos, y a la vez, que sea representativo del conjunto de datos global.

Normalmente el conjunto de datos se suele repartir en un 80% de datos de entrenamiento y un 20% de datos de test, pero se puede variar la proporción según el caso. Lo importante es ser siempre conscientes de que hay que evitar el sobreajuste u “overfitting”.

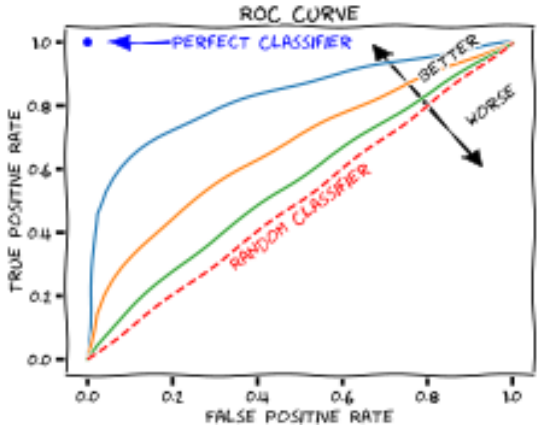
Sobreajuste (overfitting)

El sobreajuste ocurre cuando un modelo está “sobre-entrenado”. Son modelos complejos que se ajusta tan milimétricamente al conjunto de datos a partir del cual se han creado, que pierden gran parte de su poder predictivo, y ya no son útiles para otros conjuntos de datos. Esto se debe a que los datos siempre tienen cierto grado de error o imprecisión, e intentar ajustarse demasiado a ellos, complica el modelo inútilmente al mismo tiempo que le resta utilidad.

Subajuste (underfitting)

El underfitting o subajuste es justamente el caso contrario. Ocurre cuando el conjunto de datos de entrenamiento es insuficiente, con ruido en alguna de sus dimensiones o, en definitiva, poco representativo. Como consecuencia, nos lleva a un modelo excesivamente simple, con poco valor predictor. Por ello, para generar un buen modelo, es importante encontrar el punto medio entre ambas tendencias.

Curva ROC



Una curva ROC (receiver operating characteristic) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva representa dos parámetros:

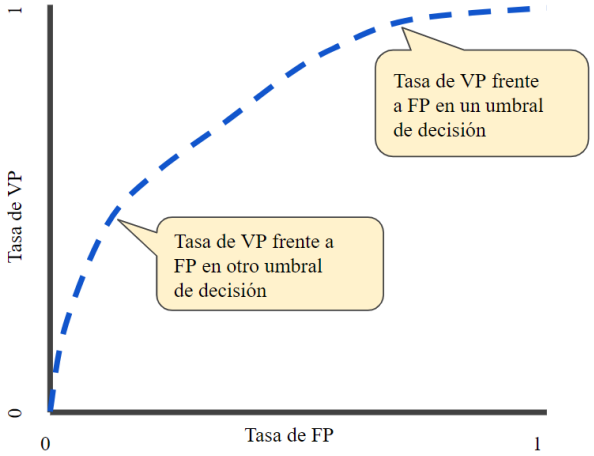
- Tasa de verdaderos positivos
- Tasa de falsos positivos

Tasa de verdaderos positivos (TPR) es sinónimo de exhaustividad y, por lo tanto, se define de la siguiente manera:

$$TRP = \frac{VP}{VP + FN}$$

Tasa de falsos positivos (FPR) se define de la siguiente manera:

$$TPR = \frac{FP}{FP + VN}$$

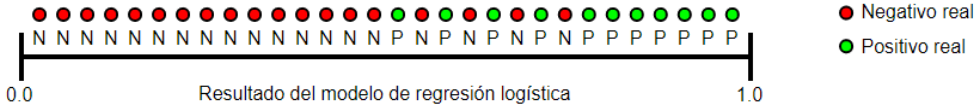


Una curva ROC representa TPR frente a FPR en diferentes umbrales de clasificación. Reducir el umbral de clasificación clasifica más elementos como positivos, por lo que aumentarán tanto los falsos positivos como los verdaderos positivos.

Para calcular los puntos en una curva ROC, en teoría se debería evaluar el modelo de regresión logística muchas veces con diferentes umbrales de clasificación, pero esto es ineficiente. Afortunadamente, existe un algoritmo eficiente basado en ordenamiento que puede brindarnos esta información, denominado AUC.

AUC: Área bajo la curva ROC

El AUC proporciona una medición agregada del rendimiento en todos los umbrales de clasificación posibles. Una forma de interpretar el AUC es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. Observa, a modo de ilustración, los siguientes ejemplos, que están ordenados de izquierda a derecha en orden ascendente con respecto a las predicciones de regresión logística:

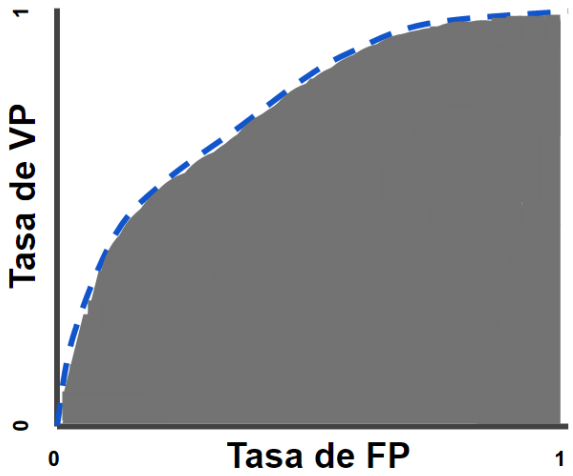


El AUC representa la probabilidad de que un ejemplo aleatorio positivo (verde) se posicione a la derecha de un ejemplo aleatorio negativo (rojo).

El AUC oscila en valor del 0 al 1. Un modelo cuyas predicciones son un 100% incorrectas tiene un AUC de 0.0; otro cuyas predicciones son un 100% correctas tiene un AUC de 1.0.

El AUC es conveniente por las dos razones siguientes:

- El AUC es invariable con respecto a la escala. Mide qué tan bien se clasifican las predicciones, en lugar de sus valores absolutos.
- El AUC es invariable con respecto al umbral de clasificación. Mide la calidad de las predicciones del modelo, sin tener en cuenta qué umbral de clasificación se elige.



Sin embargo, estas dos razones tienen algunas advertencias, que pueden limitar la utilidad del AUC en determinados casos:

- La invariabilidad de escala no siempre es conveniente. Por ejemplo, en algunas ocasiones, realmente necesitamos resultados de probabilidad bien calibrados, y el AUC no nos indicará eso.

- La invariabilidad del umbral de clasificación no siempre es conveniente. En los casos en que hay amplias discrepancias en las consecuencias de los falsos negativos frente a los falsos positivos, es posible que sea fundamental minimizar un tipo de error de clasificación. Por ejemplo, al realizar la detección de spam de correo electrónico, es probable que quieras priorizar la minimización de los falsos positivos (aunque eso resulte en un aumento significativo de los falsos negativos). El AUC no es una métrica útil para este tipo de optimización.

Supuestos

El modelo de regresión logístico como tantísimos otros (casi absolutamente todos) modelos estadísticos, deberá cumplir una serie de supuestos que permitan concluir que el mismo es una buena versión simplificada de la información y, por tanto, tiene sentido usarlo para establecer dichas relaciones ded causación.m

- Independencia: las observaciones tienen que ser independientes unas de otras.
- Relación lineal entre el logaritmo natural de odds y la variable continua:
- La regresión logística no precisa de una distribución normal de la variable continua independiente.
- Número de observaciones: no existe una norma establecida al respecto, pero se recomienda entre 50 a 100 observaciones.

Validación de supuestos

In [ ]:
```
set.seed(123)
x = rnorm(1000)
eta = 4 + 2*x
p = 1/(1 + exp(-eta))
y = rbinom(n = 1000, size = 1, prob = p)
base <- data.frame(y, x)
head(base)
```

Al haber simulado el modelo $\log\left(\frac{p}{1-p}\right) = 4 + 2x_t$, ($\beta_0 = 4, \beta_1 = 2$ y $p = P(y = 1)$), debe ser natural que el modelo ajustado sea similar, veamos.

In [ ]:
```
library(caret)
trainIndex <- createDataPartition(base$y,p=0.8,list=FALSE)
```

In [ ]:
```
train <- base[trainIndex, ]
test <- base[-trainIndex, ]
```

In [ ]:
```
dim(train)
dim(test)
```

In [ ]:
```
# Regresión lineal
fit_lm  = lm(y ~ x, data = train)
# Regresión logística
fit_glm = glm(y ~ x, data = train, family = binomial(link = "logit"))
```

In [ ]:
```
plot(y ~ x, data = base,
     pch = 20, ylab = "Probabilida estimada",
     main = "Regresión lineal vs logística")
grid()
abline(fit_lm, col = "darkorange")
curve(predict(fit_glm, data.frame(x), type = "response"),
```

```
      add = TRUE, col = "dodgerblue", lty = 2)
legend("topleft", c("Lineal", "Logística", "Data"), lty = c(1, 2, 0),
       pch = c(NA, NA, 20), lwd = 2, col = c("darkorange", "dodgerblue", "black"))
```

In [ ]:
```
summary(fit_glm)
```

In [ ]:
```
par(mfrow=c(2,2))
plot(fit_glm)
```

In [ ]:
```
pchisq(390.13 - 265.18, df=799-798, lower.tail=F)
```

In [ ]:
```
anova(fit_glm,test = "LRT")
```

In [ ]:
```
glm_link_scores <- predict(fit_glm, test, type="link")
glm_link_scores
```

In [ ]:
```
glm_response_scores <- predict(fit_glm, test, type="response")
glm_response_scores
```

In [ ]:
```
predicted.classes <- ifelse(glm_response_scores > 0.5, 1, 0)
predicted.classes
```

In [ ]:
```
test$y
```

In [ ]:
```

```

In [ ]:
```
library(pROC)
roc_info <- roc(test$y, predicted.classes, legacy.axes = TRUE)
```

In [ ]:
```
par(pty = "s") # square
roc(test$y, predicted.classes, plot = TRUE, legacy.axes = TRUE,
    percent = TRUE, xlab = "Porcentaje Falsos positivos",
    ylab = "Porcentaje verdaderos postivios", col = "#377eb8", lwd = 2,
    print.auc = TRUE, print.auc.x =45, partial.auc = c(100, 90), # en terminos de especificidad
    auc.polygon.col = "#377eb850")
```

Ejemplo de Aplicación

Se dispone de un registro que contiene cientos de emails con información de cada uno de ellos. El objetivo de estudio es intentar crear un modelo que permita filtrar qué emails son “spam” y cuáles no, en función de determinadas características. Ejemplo extraído del libro OpenIntro Statistics.

In [ ]:
```
library(tidyverse)
library(MASS)
library(car)
library(e1071)
library(caret)
library(cowplot)
library(caTools)
library(pROC)
library(ggcorrplot)
```

In [ ]:
```
telco  = read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
telco
```

In [ ]:
```
glimpse(telco)
```

In [ ]:
```
options(repr.plot.width = 6, repr.plot.height = 4)
missing_data <- telco %>% summarise_all(funs(sum(is.na(.))/n()))
missing_data <- gather(missing_data, key = "variables", value = "percent_missing")
ggplot(missing_data, aes(x = reorder(variables, percent_missing), y = percent_missing)) +
geom_bar(stat = "identity", fill = "red", aes(color = I('white')), size = 0.3)+
xlab('variables')+
```

```
coord_flip()+
theme_bw()
```

- Solo faltan 11 datos en el campo TotalCharges, así que elimine esas filas del conjunto de datos.
- Hay tres variables continuas y son Tenure, MonthlyCharges y TotalCharges. SeniorCitizen está en forma 'int', que se puede cambiar a categórica

In [ ]:
```
telco <- telco[complete.cases(telco),]

telco$SeniorCitizen <- as.factor(ifelse(telco$SeniorCitizen==1, 'YES', 'NO'))
```

In [ ]:
```
theme1 <- theme_bw()+
theme(axis.text.x = element_text(angle = 0, hjust = 1, vjust = 0.5),legend.position="none")
theme2 <- theme_bw()+
theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),legend.position="none")
```

In [ ]:
```
options(repr.plot.width = 6, repr.plot.height = 4)
telco %>%
group_by(Churn) %>%
summarise(Count = n())%>%
mutate(percent = prop.table(Count)*100)%>%
ggplot(aes(reorder(Churn, -percent), percent), fill = Churn)+
geom_col(fill = c("#FC4E07", "#E7B800"))+
geom_text(aes(label = sprintf("%.2f%%", percent)), hjust = 0.01,vjust = -0.5, size =3)+
theme_bw()+
xlab("Churn") +
ylab("Percent")+
ggtitle("Churn Percent")
```

La columna CHURN da cuenta de la cantidad de Clientes que se fueron durante el último mes. Alrededor del 26% de los clientes abandonaron la plataforma en el último mes.

In [ ]:
```
options(repr.plot.width = 12, repr.plot.height = 8)
plot_grid(ggplot(telco, aes(x=gender,fill=Churn))+ geom_bar()+ theme1,
          ggplot(telco, aes(x=SeniorCitizen,fill=Churn))+ geom_bar(position = 'fill')+theme1,
```

```
          ggplot(telco, aes(x=Partner,fill=Churn))+ geom_bar(position = 'fill')+theme1,
          ggplot(telco, aes(x=Dependents,fill=Churn))+ geom_bar(position = 'fill')+theme1,
          ggplot(telco, aes(x=PhoneService,fill=Churn))+ geom_bar(position = 'fill')+theme1,
          ggplot(telco, aes(x=MultipleLines,fill=Churn))+ geom_bar(position = 'fill')+theme_bw()+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          align = "h")
```

- Género: el porcentaje de abandono es casi igual en el caso de hombres y mujeres
- El porcentaje de abandono es mayor en el caso de las personas mayores
- Los clientes con socios y dependientes tienen una tasa de abandono más baja en comparación con aquellos que no tienen socios y dependientes.

In [ ]:
```
options(repr.plot.width = 12, repr.plot.height = 8)
plot_grid(ggplot(telco, aes(x=InternetService,fill=Churn))+ geom_bar(position = 'fill')+ theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=OnlineSecurity,fill=Churn))+ geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=OnlineBackup,fill=Churn))+ geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=DeviceProtection,fill=Churn))+ geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=TechSupport,fill=Churn))+ geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=StreamingTV,fill=Churn))+ geom_bar(position = 'fill')+theme_bw()+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          align = "h")
```

- La tasa de abandono es mucho mayor en el caso de los servicios de Internet de fibra óptica.
- Los clientes que no tienen servicios como No OnlineSecurity, OnlineBackup y TechSupport abandonaron la plataforma el mes pasado.

In [ ]:
```
plot_grid(ggplot(telco, aes(x=StreamingMovies,fill=Churn))+
          geom_bar(position = 'fill')+ theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=Contract,fill=Churn))+
          geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
```

```
          ggplot(telco, aes(x=PaperlessBilling,fill=Churn))+
          geom_bar(position = 'fill')+theme1+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          ggplot(telco, aes(x=PaymentMethod,fill=Churn))+
          geom_bar(position = 'fill')+theme_bw()+
          scale_x_discrete(labels = function(x) str_wrap(x, width = 10)),
          align = "h")
```

- Un porcentaje mayor de Clientes con suscripción mensual se ha ido en comparación con Clientes con contrato de uno o dos años.
- El porcentaje de abandono es mayor en el caso de los clientes que tienen la opción de facturación electrónica.
- Los clientes que tienen el método de pago ElectronicCheck tienden a abandonar la plataforma más en comparación con otras opciones.

In [ ]:
```
options(repr.plot.width =10, repr.plot.height = 5)
ggplot(telco, aes(y= tenure, x = "", fill = Churn)) +
geom_boxplot()+
theme_bw()+
xlab(" ")
```

Tenure: La permanencia media de los clientes que se han ido es de unos 10 meses.

In [ ]:
```
ggplot(telco, aes(y= MonthlyCharges, x = "", fill = Churn)) +
geom_boxplot()+
theme_bw()+
xlab(" ")
```

MonthlyCharges: Los clientes que se han ido tienen cargos mensuales elevados. La mediana está por encima de 75.

In [ ]:
```
ggplot(telco, aes(y= TotalCharges, x = "", fill = Churn)) +
geom_boxplot()+
theme_bw()+
xlab(" ")
```

TotalCharges:* La mediana de los cargos totales de los clientes que han abandonado es baja.

In [ ]:
```
options(repr.plot.width =6, repr.plot.height = 4)
telco_cor <- round(cor(telco[,c("tenure", "MonthlyCharges", "TotalCharges")]), 1)
```

```
ggcorrplot(telco_cor,  title = "Correlation")+theme(plot.title = element_text(hjust = 0.5))
```

Preparación de datos

Estandarizando valores diferentes en variables categóricas

In [ ]:
```
telco <- data.frame(lapply(telco, function(x) {
                  gsub("No internet service", "No", x)}))

telco <- data.frame(lapply(telco, function(x) {
                  gsub("No phone service", "No", x)}))
```

Estandarizando variables numéricas

In [ ]:
```
num_columns <- c("tenure", "MonthlyCharges", "TotalCharges")
telco[num_columns] <- sapply(telco[num_columns], as.numeric)

telco_int <- telco[,c("tenure", "MonthlyCharges", "TotalCharges")]
telco_int <- data.frame(scale(telco_int))
```

Clasificar la antigüedad por cliente

In [ ]:
```
#max(telco$tenure)
#min(telco$tenure)
telco <- mutate(telco, tenure_bin = tenure)

telco$tenure_bin[telco$tenure_bin >=0 & telco$tenure_bin <= 12] <- '0-1 y'
telco$tenure_bin[telco$tenure_bin > 12 & telco$tenure_bin <= 24] <- '1-2 y'
telco$tenure_bin[telco$tenure_bin > 24 & telco$tenure_bin <= 36] <- '2-3 y'
telco$tenure_bin[telco$tenure_bin > 36 & telco$tenure_bin <= 48] <- '3-4 y'
telco$tenure_bin[telco$tenure_bin > 48 & telco$tenure_bin <= 60] <- '4-5 y'
telco$tenure_bin[telco$tenure_bin > 60 & telco$tenure_bin <= 72] <- '5-6 y'

telco$tenure_bin <- as.factor(telco$tenure_bin)
```

In [ ]:
```
options(repr.plot.width =6, repr.plot.height = 3)
ggplot(telco, aes(tenure_bin, fill = tenure_bin)) + geom_bar()+ theme1
```

In [ ]:
```
telco_cat <- telco[,-c(1,6,19,20)]
dummy<- data.frame(sapply(telco_cat,function(x) data.frame(model.matrix(~x-1,data =telco_cat))[,-1]))
head(dummy)
```

In [ ]:
```
telco_final <- cbind(telco_int,dummy)
head(telco_final)
```

Partición de la base de datos

In [ ]:
```
#Splitting the data
set.seed(123)
indices = sample.split(telco_final$Churn, SplitRatio = 0.7)
train = telco_final[indices,]
test = telco_final[!(indices),]
```

Modelo saturado

In [ ]:
```
modelo_1 = glm(Churn ~ ., data = train, family = "binomial")
summary(modelo_1)
```

Pequeño atajo en lma búsqueda de un buen modelo

In [ ]:
```
modelo_2 <- stepAIC(modelo_1, direction="both")
```

In [ ]:
```
summary(modelo_2)
```

In [ ]:
```
pchisq(5699.5-4143.1, df=4921-4904, lower.tail=FALSE)
```

Podemos utilizar el factor de inflación de la varianza (vif) para eliminar los predictores redundantes o las variables que tienen una alta

multicolinealidad entre ellos. La multicolinealidad existe cuando dos o más variables predictoras están muy relacionadas entre sí y luego se vuelve difícil entender el impacto de una variable independiente sobre la variable dependiente.

El factor de inflación de la varianza (VIF) se utiliza para medir la multicolinealidad entre las variables predictoras en un modelo. Un predictor que tiene un VIF de 2 o menos generalmente se considera seguro y se puede suponer que no está correlacionado con otras variables predictoras. Cuanto mayor sea el VIF, mayor es la correlación de la variable predictora con otras variables predictoras. Sin embargo, los predictores con un VIF alto pueden tener un valor p alto (o muy significativo), por lo tanto, necesitamos ver la importancia de la variable predictora antes de eliminarla de nuestro modelo.

In [ ]:
```
vif(modelo_2)
```

In [ ]:
```
model_3 <-glm(formula = Churn ~ tenure + MonthlyCharges + SeniorCitizen +
    Partner + InternetService.xFiber.optic + InternetService.xNo +
    OnlineSecurity + OnlineBackup + TechSupport +
    StreamingTV + Contract.xOne.year + Contract.xTwo.year + PaperlessBilling +
    PaymentMethod.xElectronic.check, family = "binomial", data = train)
summary(model_3)
vif(model_3)
```

In [ ]:
```
final_model <- glm(formula = Churn ~ tenure + MonthlyCharges + SeniorCitizen +
    Partner + InternetService.xFiber.optic + InternetService.xNo +
    OnlineSecurity + OnlineBackup + TechSupport +
    Contract.xOne.year + Contract.xTwo.year + PaperlessBilling +
    PaymentMethod.xElectronic.check, family = "binomial", data = train)

summary(final_model)
vif(final_model)
```

Evaluación del modelo

In [ ]:
```
pred <- predict(final_model, type = "response", newdata = test[,-24])
summary(pred)
test$prob <- pred
```

```
# Usando 50% como punto de corte

pred_churn <- factor(ifelse(pred >= 0.50, "Yes", "No"))
actual_churn <- factor(ifelse(test$Churn==1,"Yes","No"))
table(actual_churn,pred_churn)
```

In [ ]:

```
cutoff_churn <- factor(ifelse(pred >=0.50, "Yes", "No"))
conf_final <- confusionMatrix(cutoff_churn, actual_churn, positive = "Yes")
accuracy <- conf_final$overall[1]
sensitivity <- conf_final$byClass[1]
specificity <- conf_final$byClass[2]
accuracy
sensitivity
specificity
```

In [ ]:

```
perform_fn <- function(cutoff)
{
  predicted_churn <- factor(ifelse(pred >= cutoff, "Yes", "No"))
  conf <- confusionMatrix(predicted_churn, actual_churn, positive = "Yes")
  accuray <- conf$overall[1]
  sensitivity <- conf$byClass[1]
  specificity <- conf$byClass[2]
  out <- t(as.matrix(c(sensitivity, specificity, accuray)))
  colnames(out) <- c("sensitivity", "specificity", "accuracy")
  return(out)
}
```

In [ ]:

```
options(repr.plot.width =8, repr.plot.height =6)
summary(pred)
s = seq(0.01,0.80,length=100)
OUT = matrix(0,100,3)

for(i in 1:100)
{
  OUT[i,] = perform_fn(s[i])
}

plot(s, OUT[,1],xlab="Cutoff",ylab="Valor",cex.lab=1.5,cex.axis=1.5,ylim=c(0,1),
```

```
     type="l",lwd=2,axes=FALSE,col=2)
axis(1,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)
axis(2,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)
lines(s,OUT[,2],col="darkgreen",lwd=2)
lines(s,OUT[,3],col=4,lwd=2)
box()
legend("bottom",col=c(2,"darkgreen",4,"darkred"),text.font =3,inset = 0.02,
       box.lty=0,cex = 0.8,
       lwd=c(2,2,2,2),c("Sensitivity","Specificity","Accuracy"))
abline(v = 0.32, col="red", lwd=1, lty=2)
axis(1, at = seq(0.1, 1, by = 0.1))
```

Finalmente, curva ROC

In [ ]:

```
glm.roc <- roc(response = test$Churn, predictor = as.numeric(pred))
plot(glm.roc,      legacy.axes = TRUE, print.auc.y = 1.0, print.auc = TRUE)
```

In [ ]:

```
```