Regresión Logística

Es un tipo de modelo estadístico, se usa para problemas de clasificación, en su versión básica es un **clasificador binario**.

Definición del modelo

Este modelo consta de tres componentes

1. Componente aleatorio

Formado por una variable aleatoria y que *sigue una Distribución de Bernoulli* con parámetro π,

$$y_i \backsim Be(\pi_i)$$

es decir, que la variable y solo puede tomar dos valores posibles (0 o 1). La función de distribución de Bernoulli es:

$$f(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} \quad y_i = 0, 1$$

como y_i es conocido, entonces podemos reescribirlo como una función que dependa de π_i:

$$f(\pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} \quad i = 1, \ldots, n$$

- y_i variable respuesta (0 o 1) de la *i-esima* observación.
- n el número de observaciones.

2. Componente Sistémico

Consiste de *transformaciones lineales*

$$\begin{aligned} z_i &= \theta_0 1 + \theta_1 x_{i1} + \ldots + \theta_{p-1} x_{i(p-1)} \\ &= \boldsymbol{\theta}^T x_i \quad i = 1, \ldots, n \quad (\text{Expresado matricialmente}) \end{aligned}$$

- θ es el vector de parámetros.
- x_i corresponde a la *i-ésima* observación, consiste de p características (o regresores).
- n el número de observaciones.

3. Función enlace

Conecta el componente aleatorio y el componente sistémico a través de la función *Logit*

$$Logit(\pi_i) = log(\frac{\pi_i}{1-\pi_i}) = z_i \quad i = 1, \dots, n$$

despejando π_i:

$$\pi_i = \frac{e^{z_i}}{1+e^{z_i}} = \frac{1}{1+e^{-z_i}}$$

escribiendo π_i como una función de z_i, tenemos:

$$\sigma(z_i) = \frac{1}{1+e^{-z_i}} \quad \text{(función sigmoide)}$$

> ***¿Por qué se usa la función logit?***
>
> *por que convierte cualquier combinación lineal en una probabilidad en [0,1].*

Entrenamiento

El objetivo es encontrar los mejores parámetros θ.

Estimador de máxima verosimilitud

Notemos que tomando el componente aleatorio y podemos obtener una función de verosimilitud.

$$\begin{aligned} f(\pi_1, \dots, \pi_n) &= f(\pi_1) \cdot \ldots \cdot f(\pi_n) \\ &= \prod_{i=1}^{n} f(\pi_i) \\ &= \prod_{i=1}^{n} \pi_i^{y_i} (1-\pi_i)^{1-y_i} \end{aligned}$$

aplicando logaritmo a $f(\pi_1, \dots, \pi_n)$:

$$\log(f) = \sum_{i=1}^{n} y_i \log(\pi_i) + (1-y_i) \log(1-\pi_i)$$

escribiendo $\log(f) = l$ tenemos:

$$l(\pi_1, \dots, \pi_n) = \sum_{i=1}^{n} y_i \log(\pi_i) + (1-y_i) \log(1-\pi_i)$$

recordar que π_i es una *función sigmoide* que depende de z_i, además $z_i = \boldsymbol{\theta}^T x_i$, entonces la **función de verosimilitud** $l(\pi_1, \dots, \pi_n)$ en realidad es una función que depende de los parámetros θ, por lo tanto podemos reescribir:

$$l(\pi_1, \dots, \pi_n) = l(\theta)$$

De esta forma podemos obtener un estimador de máxima verosimilitud para el vector θ:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ l(\theta)$$

para definirlo en **términos de minimización**, reescribimos $J(\theta) = -l(\theta)$, de obteniendo así una **función de costo**:

$$J(\theta) = -\sum_{i=1}^{n}[y_i \log{(\sigma(\theta^T x_i))} + (1-y_i)\log{(1-\sigma(\theta^T x_i))}]$$

- recordar que $\pi_i = \sigma(\theta^T x_i)$.

ahora el objetivo es encontrar los parámetros θ que **minimicen la función de costo**:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ J(\theta)$$

Estimación por el método Gradiente Descendente

Mediante este método podemos minimizar la función de costo y encontrar los mejores parámetros θ.

1. Definir una *tasa de aprendizaje* α y un número de *epochs*.
2. Inicializar θ como un **vector columna** con valores aleatorios.
3. Iterar el número de *epocs* y actualizar los valores de θ en cada iteración:

    for epoc to *epochs* do

        $\theta := \theta - \alpha \nabla J(\theta)$

    end

Luego, el reto es calcular el gradiente de la función de costo $\nabla J(\theta)$.

Gradiente de la función de costo

$$\nabla J(\theta) = \frac{\partial J(\theta)}{\partial \theta}$$

realizado el cálculo y expresado en forma matricial, tenemos:

$$\nabla J(\theta) = -X^T(y - \sigma(X\theta))$$

> *se omitió el desarrollo del cálculo por cuestiones de simplicidad*

> ***Recordar**:*
>
> - *y debe ser un vector columna.*
> - *X debe tener la primera columna de 1's.*

> - $\sigma(z) = \frac{1}{1+e^{-z}}$
> - *realizar una normalización z-score de los datos, excepto la columna de 1's*

Predicción

Dado un nuevo conjunto de datos X estimar las respuesta $\hat{y}$, en otras palabras clasificar los datos.

Para realizar la clasificación, para cada x_i debemos calcular:

$$\sigma(\theta^T x_i) = \frac{1}{1 + e^{-\theta^T x_i}}$$

también podríamos tomar todo el conjunto de datos X:

$$\sigma(X\theta) = \frac{1}{1 + e^{-X\theta}}$$

luego podríamos considerar arbitrariamente un umbral, por ejemplo de 0,5 para determinar si $\hat{y}$ será 1 o 0:

- $\sigma(\theta^T x_i) \geq 0.5$ entonces $\hat{y} = 1$
- $\sigma(\theta^T x_i) < 0.5$ entonces $\hat{y} = 0$

Es importante indicar que el umbral cambia según el problema que se está resolviendo.
