# Machine Learning para Inteligencia Artificial

## Modelos Lineales: Regresión Logística

Universidad ORT Uruguay

28 de Mayo, 2025

# Definiciones preliminares

## Logits

- Si un evento tiene probabilidad $p$, su odds ratio es $p/(1-p)$.
- Su logit o log-odds es $\text{logit}(p) = \ln[\text{odds}] = \ln\left[p/(1-p)\right]$.

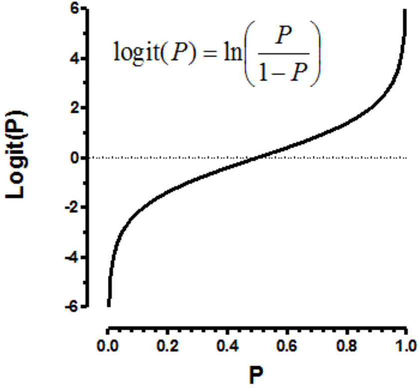
## La función sigmoidea

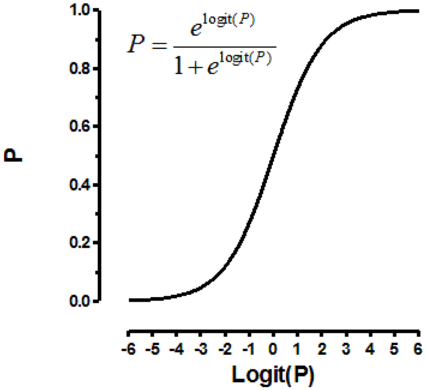
- $\text{logit} : (0,1) \to \mathbb{R}$ es invertible y su inversa es la función sigmoidea:

$$\sigma : \mathbb{R} \to (0,1)$$
$$\sigma(z) = \text{logit}^{-1}(z) = \frac{1}{1+\exp(-z)}$$

- Es una forma muy utilizada de parametrizar probabilidades.

# Logits y función sigmoidea



# Regresión logística: caso univariado (binario)

- A pesar del nombre **regresión** es un algoritmo de **clasificación**.
- Consiste en aprender la probabilidad condicional $p = \mathrm{Prob}\{y = 1 \mid x\}$.
- Es un modelo lineal en los **logits**:

$$\mathrm{logit}(p) = b + wx = [1, x] \begin{bmatrix} b \\ w \end{bmatrix} = \boldsymbol{x}^\top \boldsymbol{\theta}, \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} \in \mathbb{R}^2$$

- De forma equivalente:

$$\boxed{\mathrm{Prob}\{y = 1 \mid x\} = \sigma\left(\boldsymbol{x}^\top \boldsymbol{\theta}\right)}$$

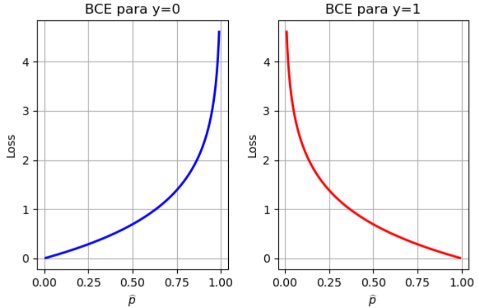
- **Clasificador**: Por defecto $h(\boldsymbol{x}) = 1$ si y solo si $\mathrm{Prob}\{y = 1 \mid x\} \geq 1/2$.

# Función de pérdida

- La función de pérdida es la **Binary Cross-Entropy (BCE)**:

$$\mathrm{BCE}\left(\widehat{p}, y\right) = \mathrm{Loss}\left(\widehat{p}, y\right) = -\Big[y \ln\left\{\widehat{p}\right\} + (1-y)\ln\left\{1-\widehat{p}\right\}\Big]$$

- **Motivación**: máxima verosimilitud



# Binary Cross Entropy Cost = Neg Log Likelihood

$$
\begin{aligned}
\text{Cost}_T(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N} \text{BCE}\left(\widehat{p}_i, y_i\right) \\
&= -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \ln\left\{\widehat{p}_i\right\} + (1-y_i)\ln\left\{1-\widehat{p}_i\right\}\right] \\
&= -\frac{1}{N}\sum_{y_i=1}\ln\left\{\widehat{p}_i\right\} - \frac{1}{N}\sum_{y_i=0}\ln\left\{1-\widehat{p}_i\right\}
\end{aligned}
$$

$$
\boxed{\text{Cost}_T(\boldsymbol{\theta}) = \boldsymbol{E}\left\{\underbrace{-\ln\Big[\text{Prob}(y = y_i \mid \boldsymbol{x}_i)\Big]}_{\text{Neg Log Likelihood}}\right\}}
$$

# Ejemplo: predicción del sexo

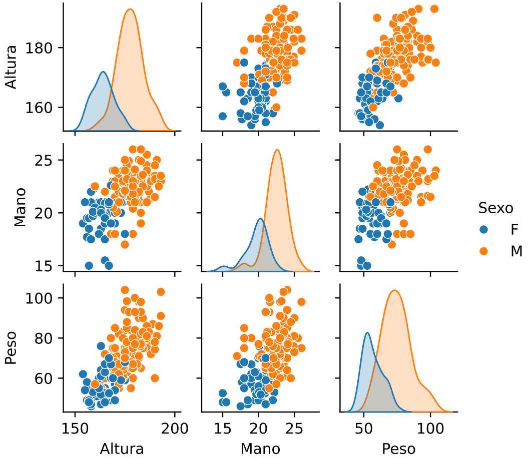
Etiqueta

- Sexo 0: Femenino, 1: Masculino

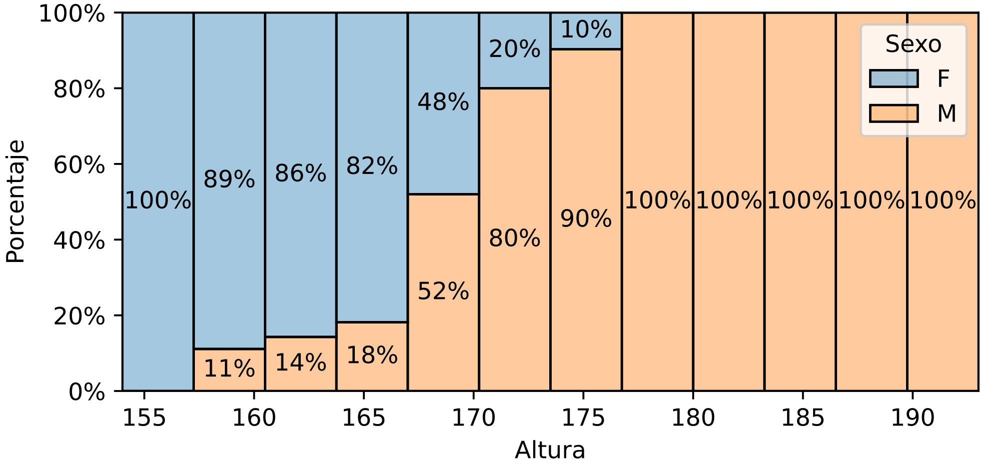
Atributos

- Altura (cm)
- Mano (cm)
- Peso (kg)

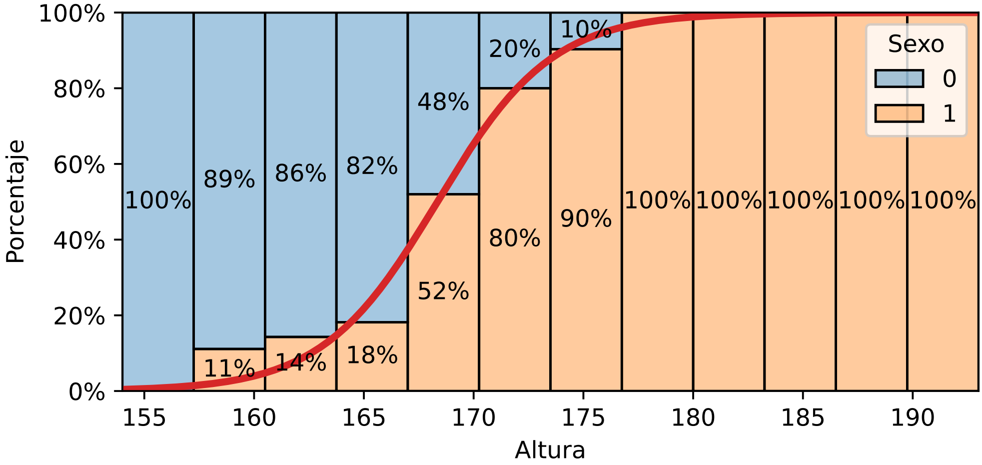
| Sexo | Altura | Mano | Peso |
|---|---|---|---|
| **F** | 164.0 | 19.7 | 56.9 |
| **M** | 177.8 | 22.5 | 74.7 |



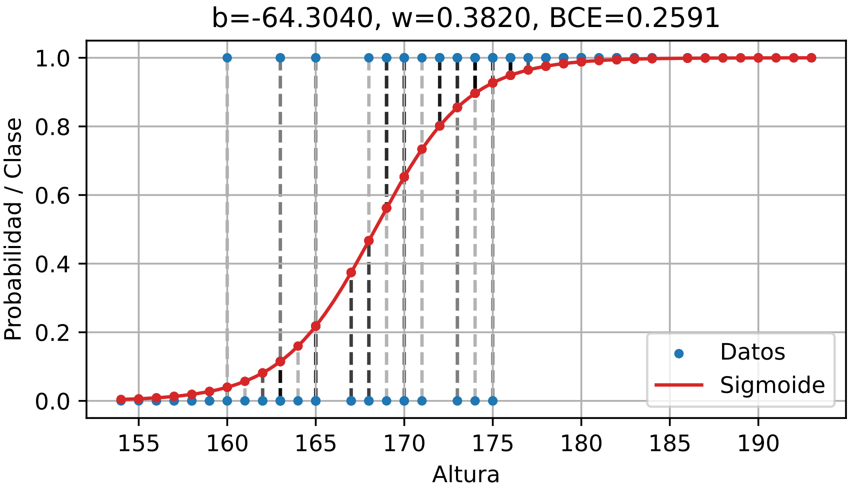
# Ejemplo: proporción en función de Altura



# Ejemplo: aproximamos la proporción con una sigmoide



# Ejemplo: función de pérdida y sigmoide



# Regresión logística: entrenamiento

- Queremos encontrar el $\boldsymbol{\theta} = [b, w]^\top$ que **minimiza** el riesgo empírico:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \mathrm{Cost}_T(\boldsymbol{\theta}) \right\}$$

- A diferencia con Regresión Lineal, **no hay fórmula** para $\widehat{\boldsymbol{\theta}}$.
- Debe calcularse **numéricamente** con algún algoritmo de optimización.
- En sklearn disponemos de los siguientes métodos de optimización:

**lbfgs, liblinear, newton-cg, newton-cholesky, sag, saga**

# Regresión logística: caso multivariado (binario)

- **Espacio de atributos**: $\mathcal{X} \subset \mathbb{R}^D$

- Ahora el modelo es

$$\widehat{p} = \sigma\left(\boldsymbol{x}^\top \boldsymbol{\theta}\right) = \frac{1}{1 + e^{-\boldsymbol{x}^\top \boldsymbol{\theta}}}$$

en donde los vectores $\boldsymbol{x}$ y $\boldsymbol{\theta}$ están dados por

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x^{(1)} \\ \vdots \\ x^{(D)} \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$$

# Regresión logística: caso multivariado (binario)

- El producto matricial representa

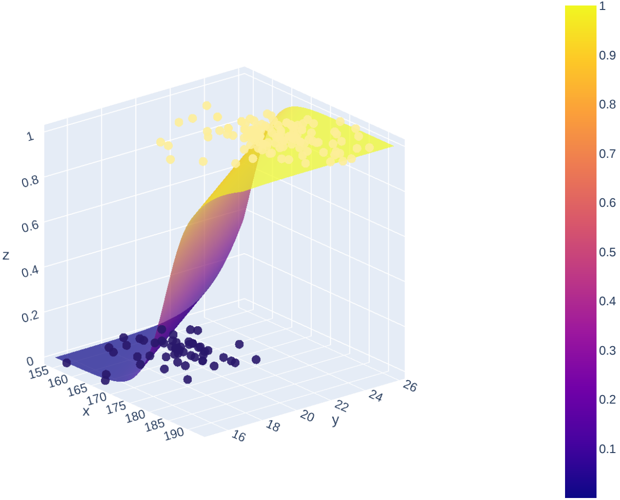
$$\text{logit}\left(\widehat{p}\right) = \begin{bmatrix} 1 & x^{(1)} & \dots & x^{(D)} \end{bmatrix} \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix} = b + \sum_{j=1}^{D} w_j x^{(j)}$$

- Considerando la matriz de diseño y el vector de etiquetas (o targets):

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_1^{(D)} \\ 1 & x_2^{(1)} & \dots & x_2^{(D)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N^{(1)} & \dots & x_N^{(D)} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

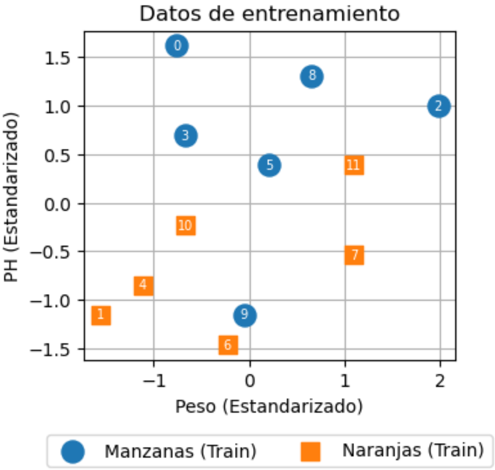
- La función de pérdida compara $\widehat{\boldsymbol{p}} = \sigma\left(\boldsymbol{X}\boldsymbol{\theta}\right)$ con $\boldsymbol{y}$ al igual que antes.

# Ejemplo: predicción del Sexo (2D)



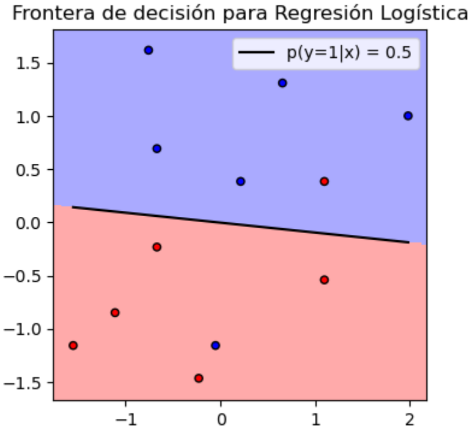
# Ejemplo: clasificando manzanas y naranjas

| id | Peso (g) | PH | Fruta |
|---|---|---|---|
| 0 | 139 | 3.9 | 0 (manzana) |
| 1 | 130 | 3.0 | 1 (naranja) |
| 2 | 170 | 3.7 | 0 (manzana) |
| 3 | 140 | 3.6 | 0 (manzana) |
| 4 | 135 | 3.1 | 1 (naranja) |
| 5 | 150 | 3.5 | 0 (manzana) |
| 6 | 145 | 2.9 | 1 (naranja) |
| 7 | 160 | 3.2 | 1 (naranja) |
| 8 | 155 | 3.8 | 0 (manzana) |
| 9 | 147 | 3.0 | 0 (manzana) |
| 10 | 140 | 3.3 | 1 (naranja) |
| 11 | 160 | 3.5 | 1 (naranja) |



# Ejemplo: clasificando manzanas y naranjas

La **frontera de decisión** es la curva $\boldsymbol{x}^\top \boldsymbol{\theta} = 0$ (o de forma equivalente $\widehat{p} = 1/2$)



# Regresión logística: funciones base

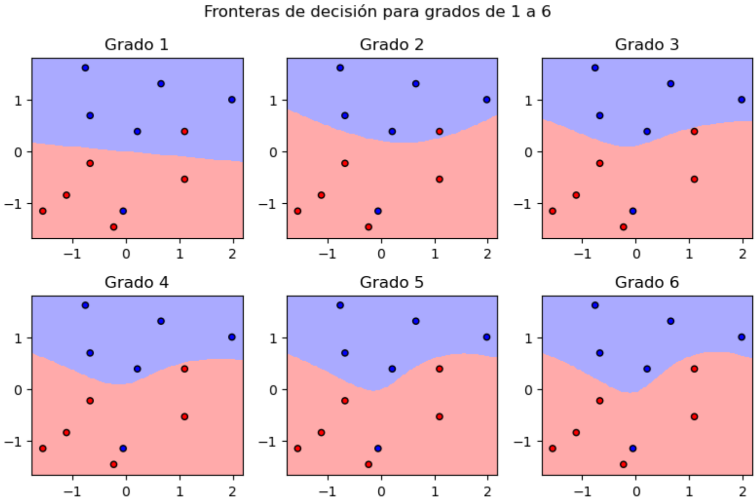
- El sesgo inductivo es

$$\mathcal{H} = \left\{ h_{\boldsymbol{\theta}} : \boldsymbol{x} \mapsto \sigma\left(b + w_1 h_1(\boldsymbol{x}) + w_2 h_2(\boldsymbol{x}) + \cdots + w_K h_K(\boldsymbol{x})\right) \right\}$$

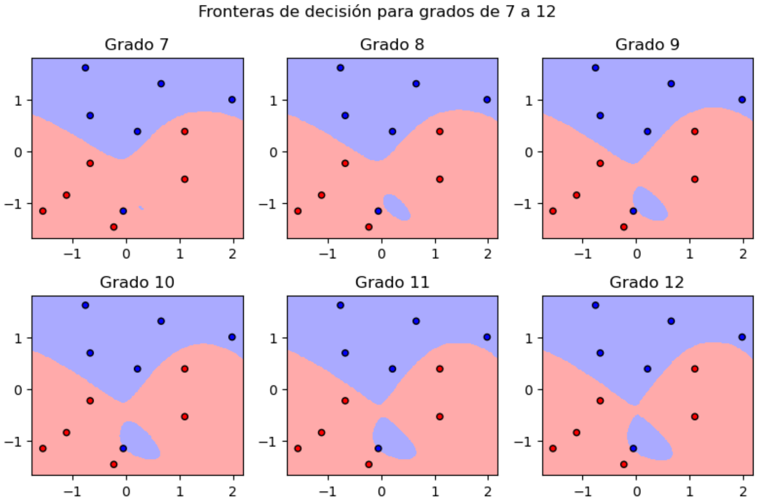
- Casos particulares son:
  - Regresión polinomial: cuando $h_j(\boldsymbol{x})$ son productos de potencias $x^{(j)}$
  - Regresión trigonométrica: cuando $h_j(\boldsymbol{x})$ son funciones trigonométricas
- La relación entre $\text{logit}\left(\widehat{p}\right)$ y los coeficientes $\boldsymbol{\theta} = (b, \boldsymbol{w})$ sigue siendo lineal
- Mismo procedimiento al caso multivariado considerando la matriz de diseño:

$$\boldsymbol{X} = \begin{bmatrix} 1 & h_1\left(\boldsymbol{x}_1\right) & \cdots & h_K\left(\boldsymbol{x}_1\right) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & h_1\left(\boldsymbol{x}_N\right) & \cdots & h_K\left(\boldsymbol{x}_N\right) \end{bmatrix}$$

# Ejemplo: clasificando manzanas y naranjas



# Ejemplo: clasificando manzanas y naranjas



# Bibliografía


- An introduction to statistical learning with applications in Python. Cap 4.2.
- Machine Learning - A First Course for Engineers and Scientists. Capítulo 3.2.