

Machine Learning para Inteligencia Artificial

Regularización en Árboles de Decisión

Universidad ORT Uruguay

9 de Abril, 2025

Regularización

- Modificación a un algoritmo de aprendizaje que pretende reducir el error verdadero (en \mathcal{D}) pero no su error empírico (en T).
- Consiste en agregar términos de **penalización** a la función de costo para desalentar modelos complejos:

$$\left(\begin{array}{c} \text{Costo} \\ \text{nuevo} \end{array} \right) = \left(\begin{array}{c} \text{Costo} \\ \text{original} \end{array} \right) + \alpha \times \left(\begin{array}{c} \text{Penalización} \\ \text{a la complejidad} \end{array} \right)$$

- α es un parámetro que controla la importancia del término de regularización.

Regularización en árboles: Cost Complexity Pruning

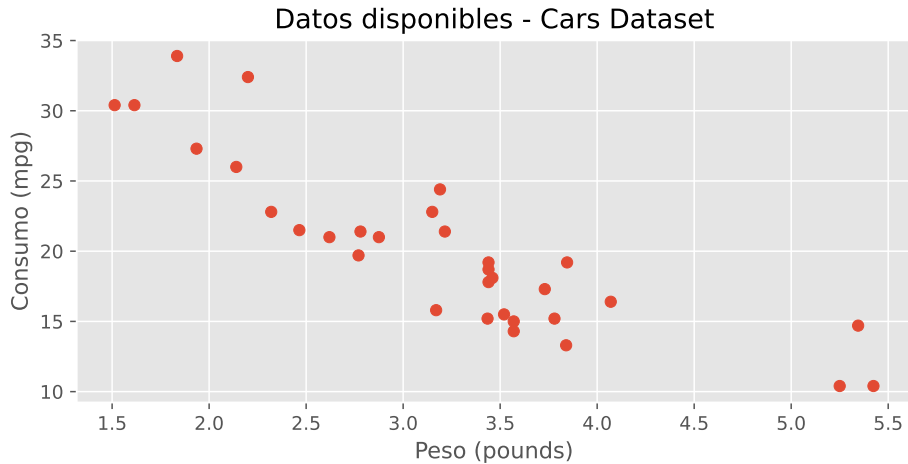
- En un Árbol de Regresión, la MSE se puede escribir:

$$\text{MSE}_T(h) = \sum_{H \in \text{hojas}(h)} \frac{|H|}{|T|} \cdot \text{Var}(y; H)$$

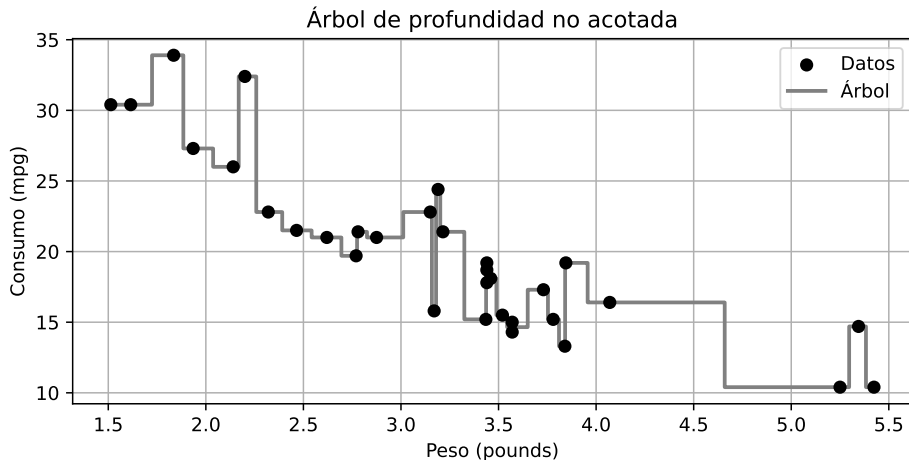
El nuevo costo regularizado es:

$$\text{Cost}_T^\alpha(h) = \text{MSE}_T(h) + \alpha \cdot |\text{hojas}(h)|$$

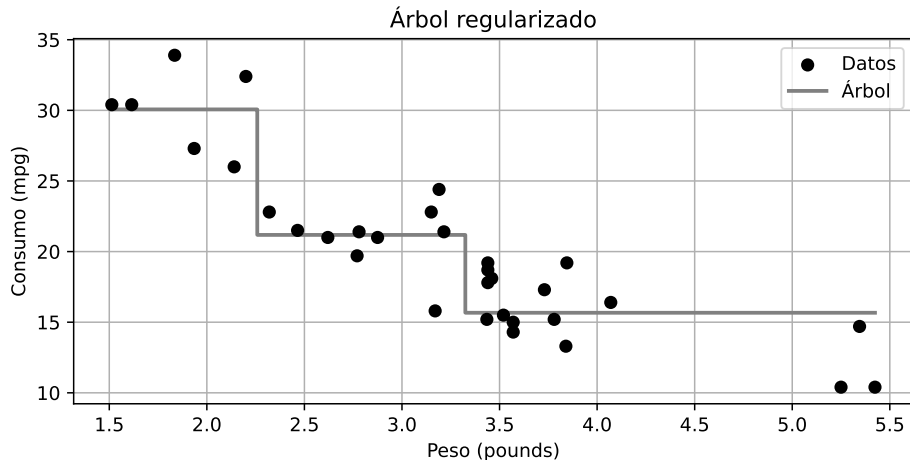
Ejemplo - Cars Dataset



Ejemplo - Árbol sin regularizar



Ejemplo - Árbol regularizado con el mejor α



Procedimiento para buscar el mejor α

- **Input:** A árbol completo
- **Output:** Lista de sub-árboles $[A_0, A_1, \dots, A_k]$ y alphas $[\alpha_0, \dots, \alpha_k]$

1. Inicializar $\text{Lista}_A = [A]$ y $\text{Lista}_\alpha = [0]$.
2. Mientras A tiene nodos internos repetir:
 - 2.1 Inicializar $\alpha_{\min} = \infty$, $n_{\min} = \text{None}$
 - 2.2 Para cada nodo interno $n \in A$ repetir:
 - 2.2.1 $A_n =$ sub-árbol con raíz en n ; $M_n = \text{MSE}(n)$ como hoja; $M(A_n) = \text{MSE}(A_n)$
 - 2.2.2 Si $|A_n| > 1$: $\alpha_n = (M_n - M(A_n)) / (|A_n| - 1)$.
Si $\alpha_n < \alpha_{\min}$: $\alpha_{\min} = \alpha_n$; $n_{\min} = n$
 - 2.3 Actualizar A podando A_n a una hoja para $n = n_{\min}$.
 - 2.4 Agregar A a Lista_A y α_{\min} a Lista_α .

De esta lista se selecciona el mejor α con CV

¿Por qué funciona? (OPCIONAL!)

■ Si $\alpha > \alpha_n$:

$$\begin{aligned}M(A_n) + \alpha|A_n| &= M(A_n) + \alpha_n|A_n| + (\alpha - \alpha_n)|A_n| \\&> M_n + \alpha_n \cdot 1 + (\alpha - \alpha_n)|A_n| \\&> M_n + \alpha_n \cdot 1 + (\alpha - \alpha_n) \\&= M_n + \alpha \cdot 1.\end{aligned}$$

■ Es decir

(Costo penalizado de árbol podado) $<$ (Costo penalizado de árbol s/podar)

para todo $\alpha > \alpha_n$.

Bibliografía

- An introduction to statistical learning with applications in Python. Capítulo 8.1.