Machine Learning para Inteligencia Artificial

Regresión con Árboles de Decisión

Universidad ORT Uruguay

2 de Abril, 2025

Ingredientes de la regresión

- **Espacio de atributos**: $\mathcal{X} \subset \mathbb{R}^D$
- **Espacio de etiquetas**: $\mathcal{Y} \subset \mathbb{R}$ (la etiqueta es una variable continua)
- **Conjunto de datos**: $T = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ muestra iid de $\mathcal{D}$
- **Función de pérdida**: por ejemplo quadratic loss $\text{Loss}(\widehat{y}, y) = (\widehat{y} - y)^2$
- **Sesgo inductivo**: $\mathcal{H} = \{h : \mathcal{X} \rightarrow \mathcal{Y}\}$

La función a aprender: h_{opt}

- Relación $\boldsymbol{x} \rightsquigarrow y$ estocástica: $p(y \mid \boldsymbol{x})$
- La mejor hipótesis $h_{\text{opt}} \in \mathcal{H}$:

$$h_{\text{opt}}(\boldsymbol{x}) = \underset{h \in \mathcal{H}}{\arg\min}\, \boldsymbol{E}\left[\text{Loss}(h(\boldsymbol{x}), y) \mid \boldsymbol{x}\right]$$

- Es la función que queremos aprender
- Desconocida en la práctica
- Datos: muestra iid de la forma

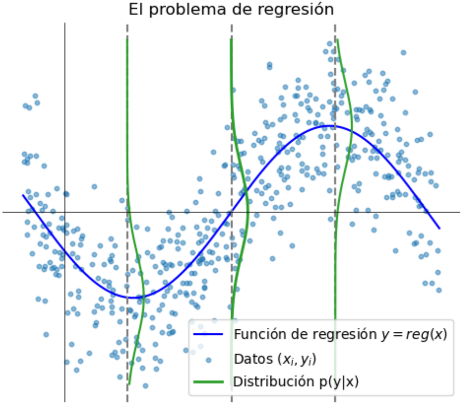
$$y_i = h_{\text{opt}}(\boldsymbol{x}_i) + \epsilon_i$$

para $i = 1, \ldots, N$

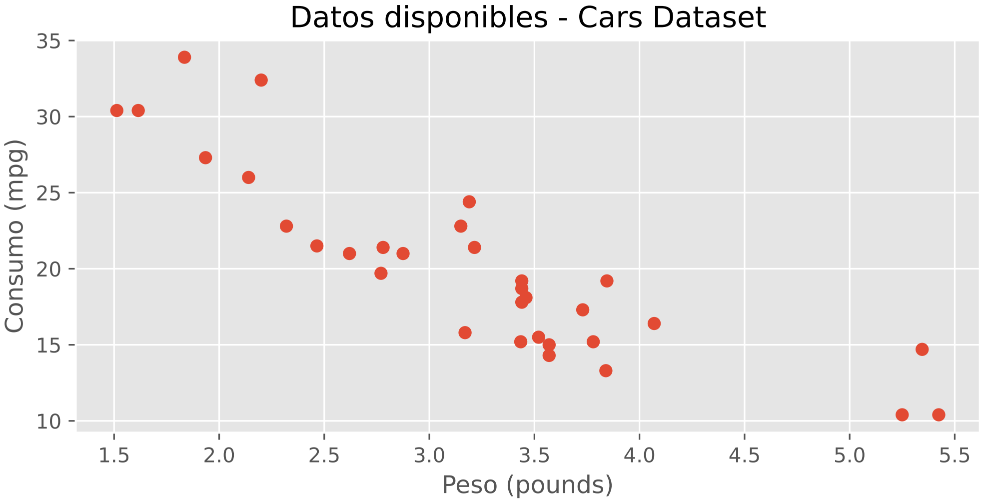
- ϵ_i error aleatorio

Si $\text{Loss}(\widehat{y}, y) = (\widehat{y} - y)^2$ quadratic loss

$$h_{\text{opt}}(\boldsymbol{x}) = \text{reg}(\boldsymbol{x}) = \boldsymbol{E}\left[y \mid \boldsymbol{x}\right]$$



Ejemplo


Datos disponibles - Cars Dataset
Consumo (mpg)
Peso (pounds)
35
30
25
20
15
10
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5

Predecir con una constante (sin hacer preguntas)

Consideremos un sesgo inductivo muy simple:

$$\mathcal{H}_0 = \{h : \mathcal{X} \to \mathcal{Y} \mid h \text{ es constante}\}$$

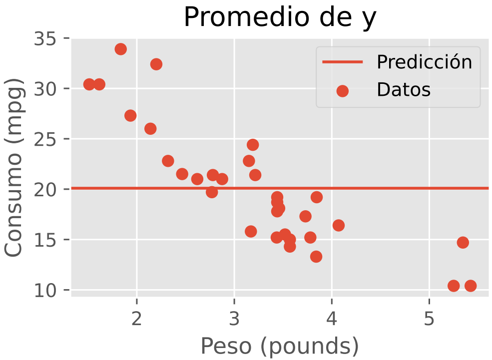
y la función de pérdida cuadrática.

Costo empírico $=$ Mean Squared Error (MSE)

$$J_T(h) = \mathrm{MSE}(h) = \frac{1}{|T|}\sum_{i=1}^{N}(h(x_i) - y_i)^2$$

Es sencillo ver (derivar e igualar a cero) que

$$\text{promedio} = \bar{y} = \arg\min_{h \in \mathcal{H}_0} \mathrm{MSE}(h)$$



Me permito hacer una pregunta

Consideremos ahora el sesgo inductivo $\mathcal{H}_1$ donde $h(x) = \begin{cases} c_1 & \text{si } x \leq v \\ c_2 & \text{si } x > v \end{cases}$

Queremos (c_1, c_2, v) que minimizan la MSE:

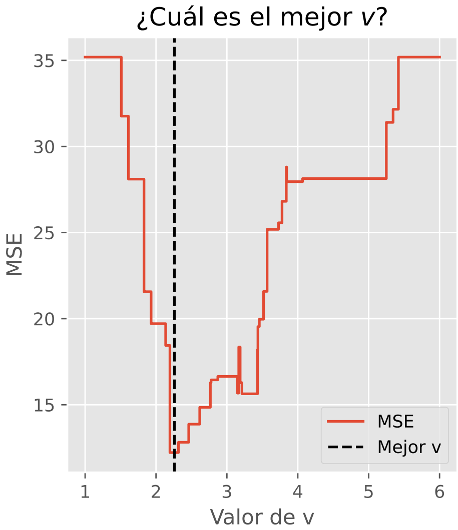
$$J_T(h) = \text{MSE}(h) = \frac{1}{|T|}\left[\sum_{x_i < v}(y_i - c_1)^2 + \sum_{x_i \geq v}(y_i - c_2)^2\right]$$

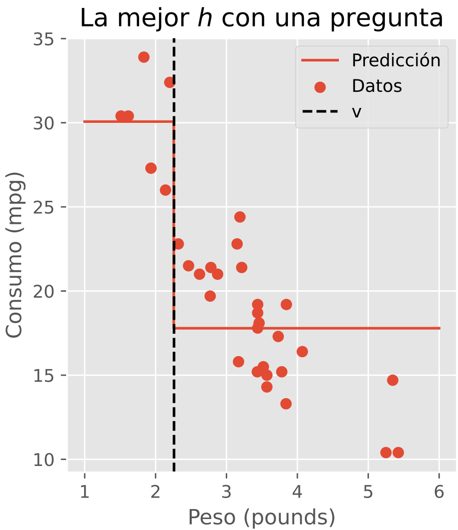
Fijando v, la MSE se minimiza para

$$\begin{cases} c_1 = \bar{y}_1 = \text{promedio}\{y_i : x_i \leq v\} \\ c_2 = \bar{y}_2 = \text{promedio}\{y_i : x_i > v\} \end{cases}$$

¿Cuál es el mejor v?

Me permito hacer una pregunta





$\mathcal{H}_1 = \{$árboles de profundidad 1$\}$

Una pregunta P: ¿$x \leq v$? divide los datos en dos:

P: ¿$x \leq v$?

$H_{\text{true}} = \{(x_i, y_i) : P(x_i) = \text{true}\}$
Predicción: promedio$\{y_i : P(x_i) = \text{true}\}$

$H_{\text{false}} = \{(x_i, y_i) : P(x_i) = \text{false}\}$
Predicción: promedio$\{y_i : P(x_i) = \text{false}\}$

- La MSE de la predicción promedio $\bar{y}$ en una hoja H es igual a la varianza

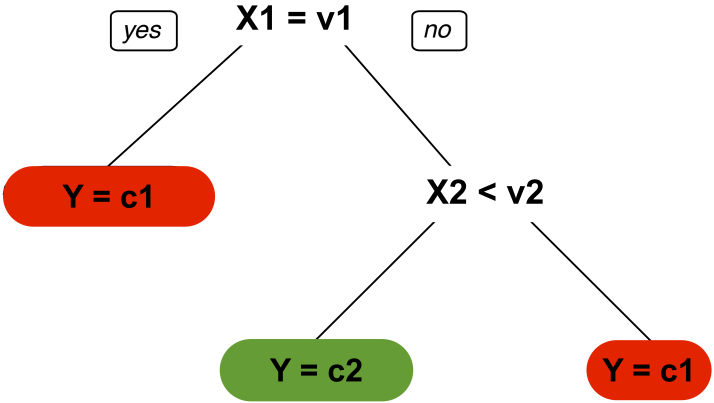
$$\text{MSE}_H(\bar{y}) = \frac{1}{|H|} \sum_H (y - \bar{y})^2 = \text{Var}(y; H)$$

¿Qué es un árbol de decisión?

Nodos = preguntas

Arcos = respuestas

Hojas = predicción



¿Cómo se construye?

Algoritmo *greedy*

Se comienza por la raíz igual al conjunto T de datos de entrenamiento

Se repite el siguiente proceso para cada hoja H

- Se elige una pregunta P para dividir H
- Se construye un arco por cada respuesta $a \in \{\text{True}, \text{False}\}$ a P
- Se desciende por cada arco a con el conjunto:

$$H_a = \{s \in H \mid P(s) = a\} \quad a \in \{\text{True}, \text{False}\}$$

La predicción en una hoja del árbol es el promedio.

La mejor pregunta

Sea P: ¿A < v? una pregunta (en el caso categórico P: ¿A = v?)

$$H_a = \{s \in H \mid P(s) = a\} \quad a \in \{\text{True}, \text{False}\}$$

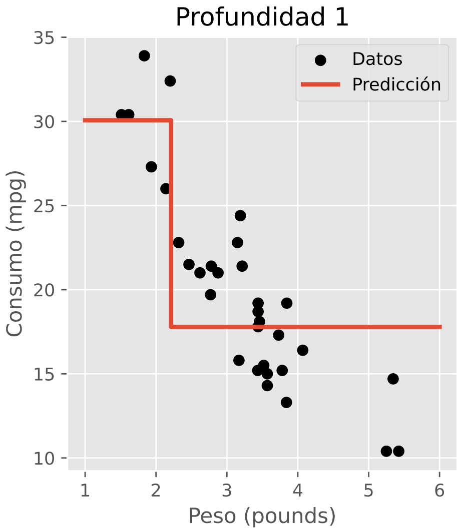
La varianza esperada después de dividir una hoja H usando P es:

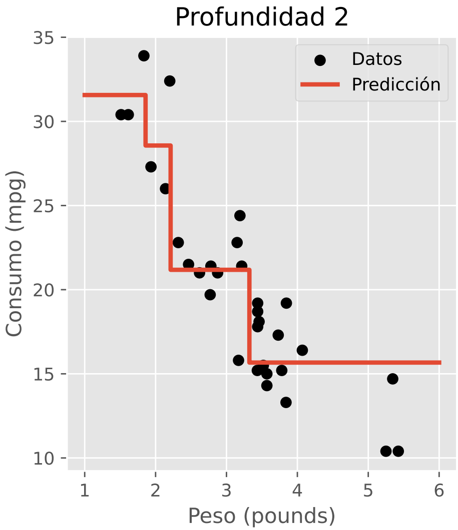
$$\boxed{\text{Var}_P(H) = \frac{|H_{\text{True}}|}{|H|} \cdot \text{Var}(y; H_{\text{True}}) + \frac{|H_{\text{False}}|}{|H|} \cdot \text{Var}(y; H_{\text{False}})}$$

Se elige P correspondiente al par (A, v) que minimiza la varianza:

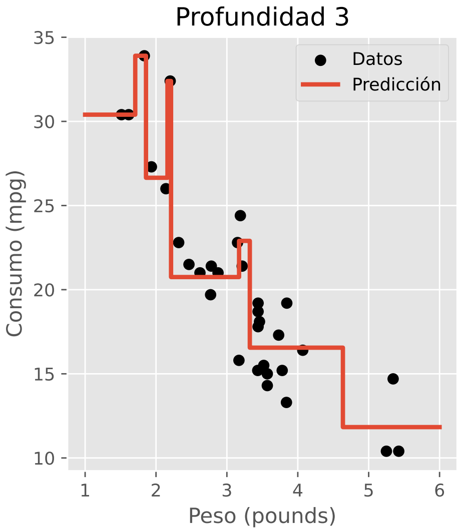
$$P^* = \arg\min_P \text{Var}_P(H)$$

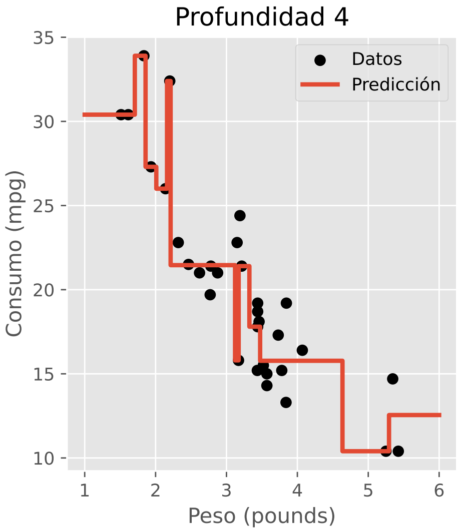
Recursive Binary Splitting



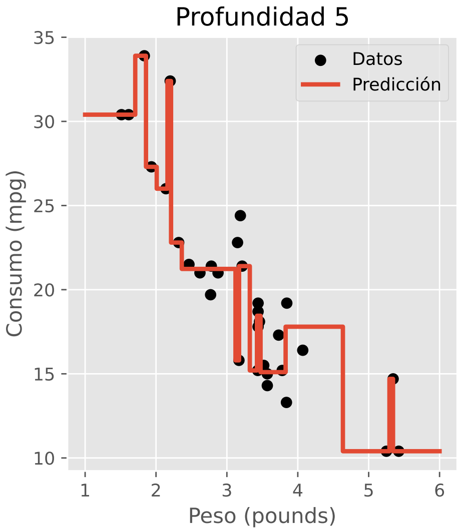


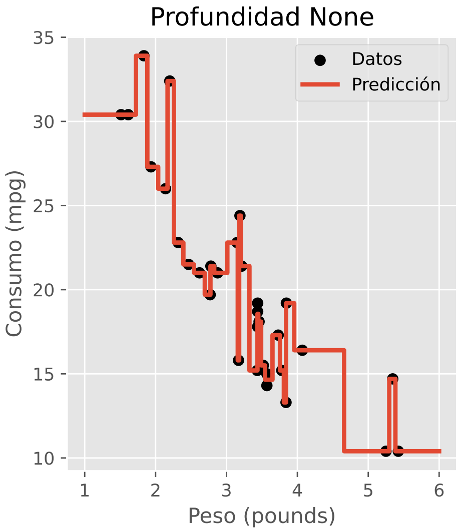
Recursive Binary Splitting





Recursive Binary Splitting





Bibliografía

- An introduction to statistical learning with applications in Python. Cap 8.
- Machine Learning - A First Course for Engineers and Scientists. Cap 2.3.
- Machine Learning Refined: Foundations, Algorithms, and Applications. Cap 14.
- Raschka, S. Introduction to Machine Learning. Lecture 6. (2021)