

# Introducción a Machine Learning

**Regresión Linear**  
MSc. Marco Sobrevilla

# Objetivo



- Aprender o recordar conceptos de regresión y métodos de ajuste

# Agenda

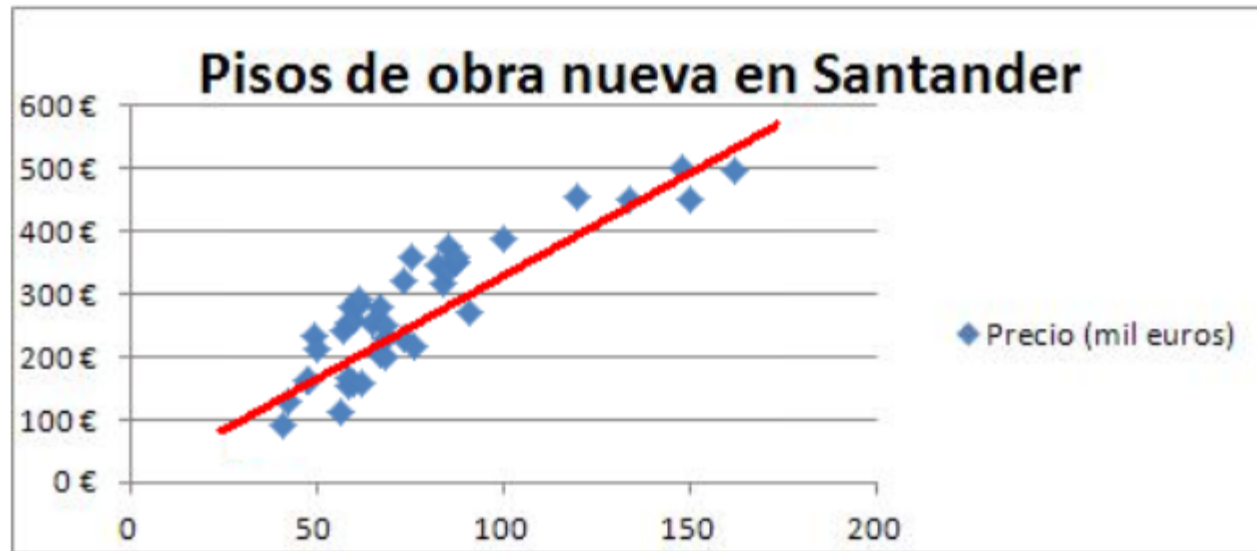


- **Regresión**
- **Gradiente Descendiente**

# Regresión



**boost**  
accelerate your learning



# Regresión - Ejemplo

Superficie en $m^2$ (x)	Precio en mil euros (y)
59	280
68	252
75	360
41	93
...	...

Notación:

$m$  = número de ejemplos

$x$  = variable de entrada (superficie),  $x^{(i)}$

$y$  = variable de salida (precio),  $y^{(i)}$

- Objetivo: Hallar una recta mejor **se aproxime** a los datos

# Regresión

- Sea:

$$h_w(x) = w_0 \cdot 1 + w_1 \cdot x_1 + \dots + w_M \cdot x_M$$

- **Objetivo:** Minimizar la función de costo (*Dada por el error cuadrático*)

$$L(w) = \frac{1}{2 * M} \sum_{i=1}^M (h_w(x^i) - y^i)^2$$

# ¿Cómo minimizar?

- ¿Cómo minimizar?
  - Ecuación Normal
    - Puede ser demasiado caro
    - Costo computacional:  $O(M^3)$
    - No es escalable: no aplicable para  $M > 100$
    - ☹️

# Optimización basada en Gradientes



- Funciona mejor para  $M > 100$
- Algoritmos mas usados
  - *Gradient Descent/Ascent*
  - *Stochastic Gradient Descent/Ascent*
  - AdaGrad, ADAM (populares en *Deep Learning*)
  - Alg. basados en momentos, decaimiento, entre otros.

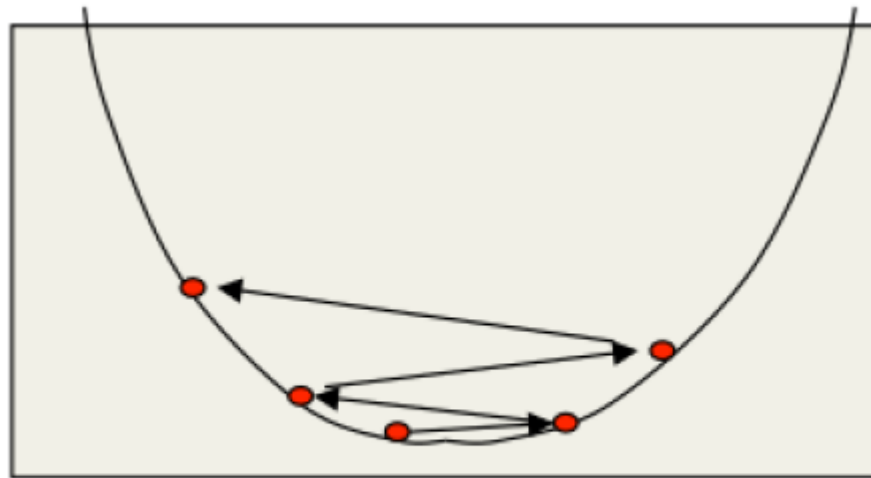


# Gradiente Descendiente

- Para cada muestra

$$w = w - \alpha * \frac{dL}{dw}$$

- Donde  $\alpha$  es la velocidad o ratio de aprendizaje
  - Si  $\alpha$  es muy pequeño: Aprendizaje lento
  - Si  $\alpha$  es muy grande: Puede no converger



$w \longrightarrow$

Para  $h_w(x) = 1 + w_1 * x_1$

**Fin** 😊