



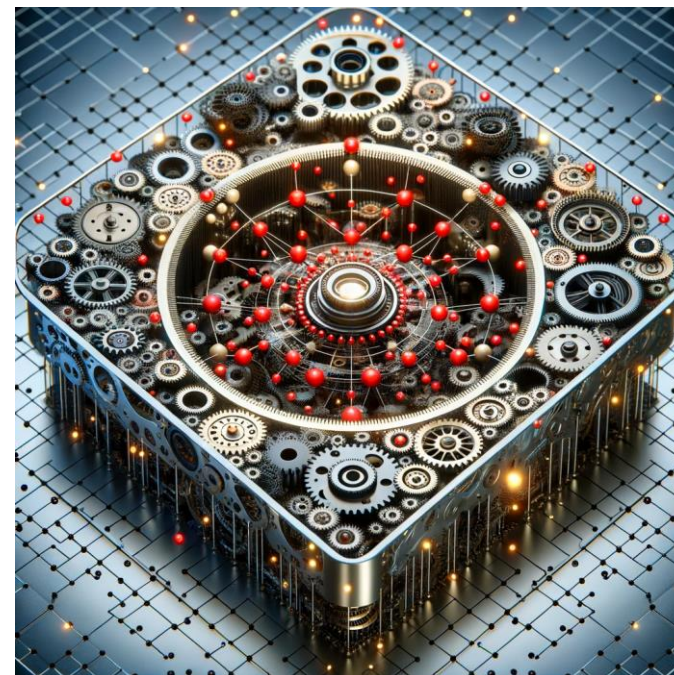
Regresión Lineal

Entrenamiento/Construcción



Obtención de parámetros: Visión General

- FUNDAMENTAL: Entrenar un modelo es obtener el valor de los parámetros del modelo a partir de los datos del set de Train
- En el caso de la Regresión Lineal, es calcular los pesos o coeficientes de la fórmula de la regresión lineal
- Estos parámetros se obtienen generalmente siguiendo un mismo proceso (que veremos a continuación muy someramente)
- Pero también es posible que exista un procedimiento matemático exacto para obtenerlos (como es el caso de la regresión lineal, pero no es lo normal)



Obtención de parámetros: Proceso (I)

- Se escoge una “función de pérdida” (*loss function*): una medida del error entre las predicciones del modelo y los valores reales
- Ejemplos comunes: Error Cuadrático Medio (para regresión) y Entropía Cruzada (para clasificación).

El objetivo del proceso es obtener el valor de los parámetros que hacen que esa función de pérdida tenga el valor mínimo considerando los datos de entrenamiento



Obtención de parámetros: Proceso (II)

- Forma “Bruta”:
 - Obtener para cada combinación de valores de los parámetros el valor de la función de pérdida para el dataset de entrenamiento
 - Quedarnos con la combinación de valores que tiene el valor más pequeño para la función de pérdida
 - Es impracticable para parámetros de valores continuos: quizás la computación cuántica nos haga ir a ese método...mientras

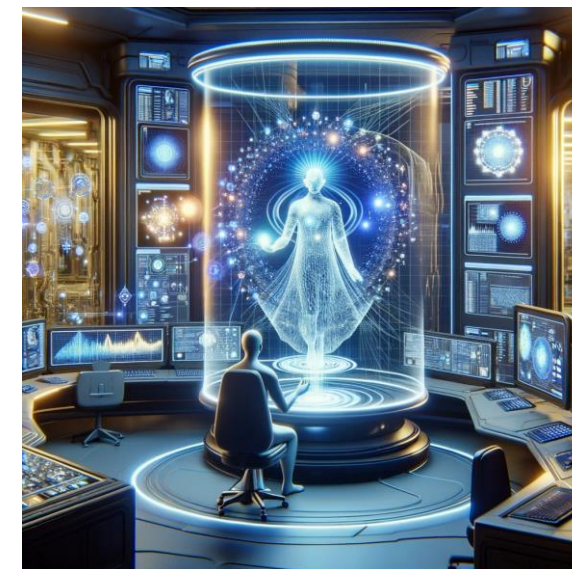
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$



Obtención de parámetros: Proceso (III)

- Forma “habitual”:
 - Existen algoritmos que van probando combinaciones de los parámetros de una forma más eficiente
 - No prueban todas las combinaciones de valores de los parámetros
 - Las combinaciones se escogen de una forma “inteligente”
 - En algunos casos no se prueban tampoco todos los datos con cada combinación elegida. (Algoritmos estocásticos)



El método dominante hoy en día: Gradiente descendente (y en general su variante estocástica)



Obtención de parámetros: Función de pérdida

En el caso de la regresión lineal la función de pérdida es también una función de evaluación y que ya conocemos:

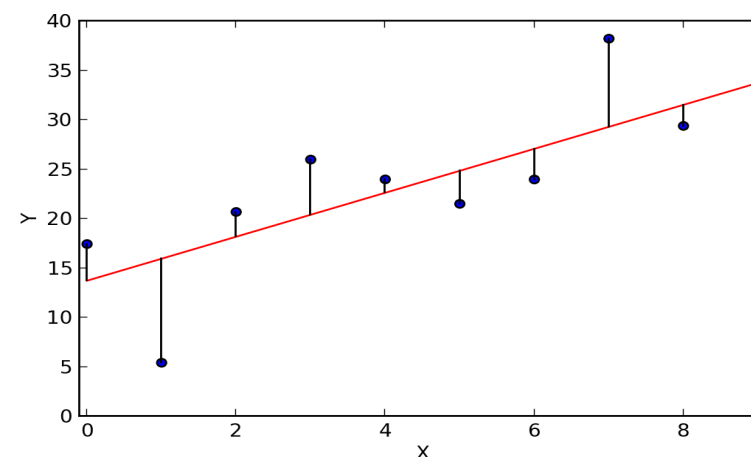
Error cuadrático medio (Mean Squared Error)

N = n.º observaciones

Y = valores reales

\hat{Y} = valores predichos

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

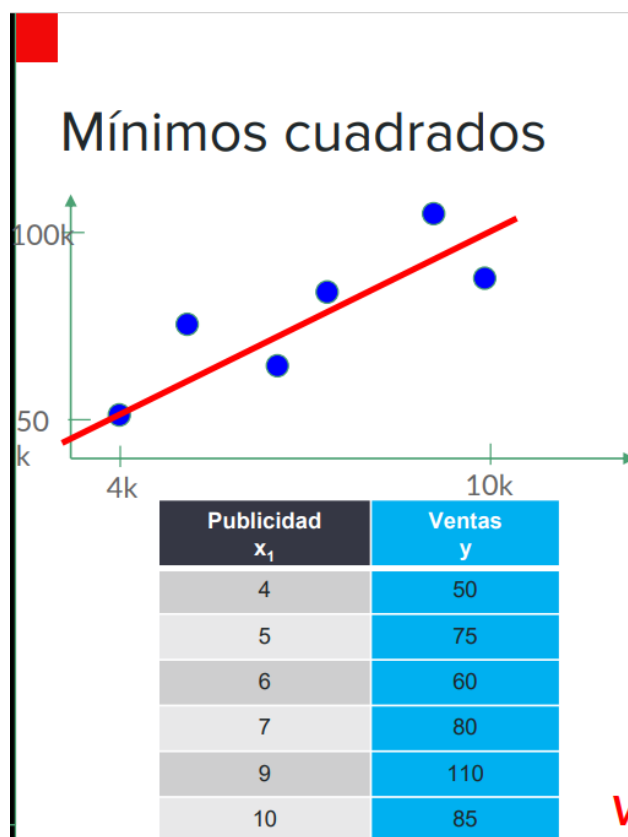


Aunque muchas funciones de pérdida son también funciones de evaluación de modelos, no siempre es así, sobre todo en los modelos que se emplean en problemas de clasificación



Obtención de parámetros: Mínimos Cuadrados

En el caso específico de la regresión lineal y la relación que establece entre target y features, existe una solución matemática exacta (que no quiere decir que sea perfecta 😊): *El método de los mínimos cuadrados*



$$w = (X^T X)^{-1} X^T y$$

$$X = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 6 \\ 1 & 7 \\ 1 & 9 \\ 1 & 10 \end{bmatrix}, y = \begin{bmatrix} 50 \\ 75 \\ 60 \\ 80 \\ 110 \\ 85 \end{bmatrix}$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 5 & 6 & 7 & 9 & 10 \end{bmatrix}$$

$$\Rightarrow w = (X^T X)^{-1} X^T y \\ = \begin{bmatrix} 27.85 \\ 7.14 \end{bmatrix}$$

$$\text{Ventas} = 27.85 + 7.14 * \text{Publicidad}$$

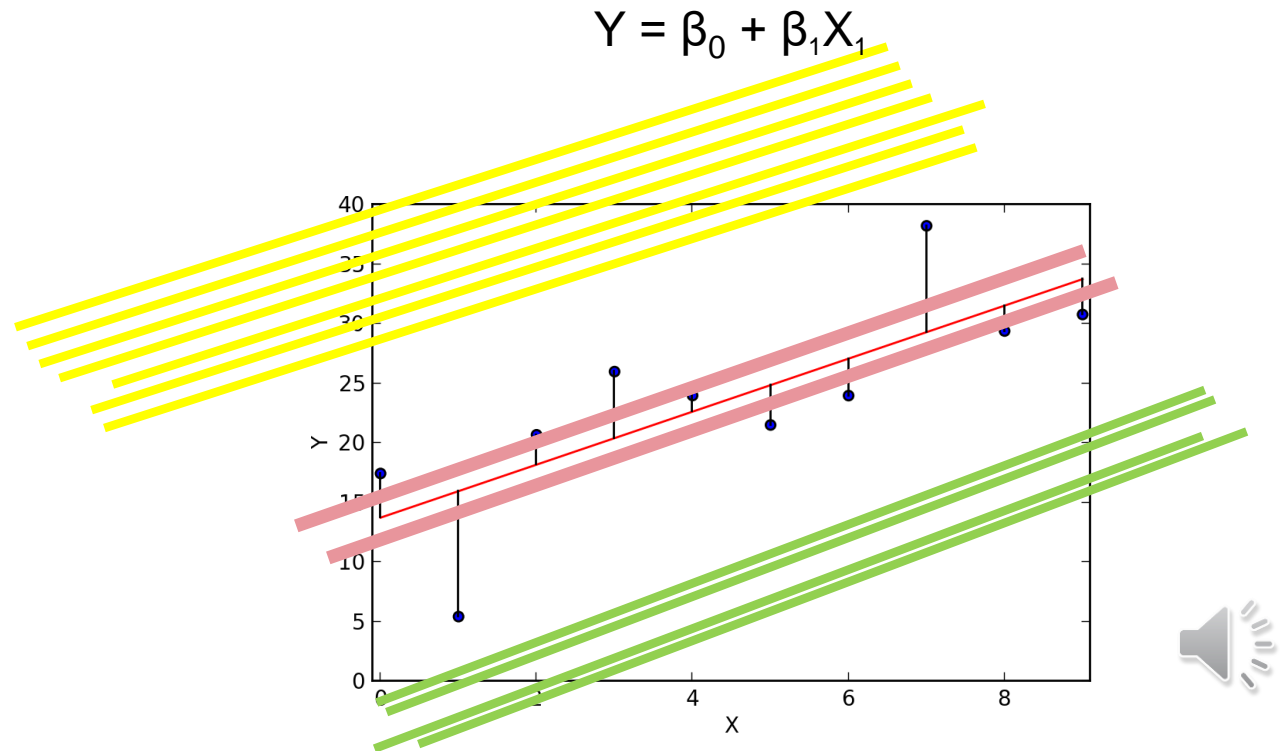
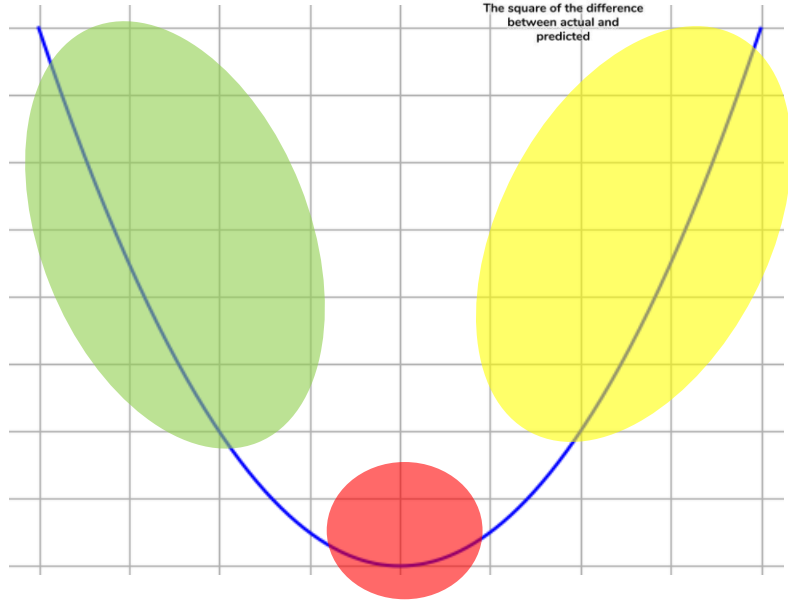


Obtención de parámetros: Gradiente Descendente

Pero también se puede resolver utilizando el método del gradiente descendente.

$$MSE = \frac{1}{n} \sum \left(y - \hat{y} \right)^2$$

The square of the difference
between actual and
predicted



Gradient Descent en Regresión Lineal

1. Función de costes

$$J(w, b) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (y_i - (wx_i + b))^2$$

2. Gradiente – Derivadas parciales parciales

$$\nabla J(w, b) = \begin{bmatrix} \frac{\delta J}{\delta w} \\ \frac{\delta J}{\delta b} \end{bmatrix} = \begin{bmatrix} \frac{2}{n} \sum_{i=1}^n -x_i (y_i - (wx_i + b)) \\ \frac{2}{n} \sum_{i=1}^n -(y_i - (wx_i + b)) \end{bmatrix}$$

3. Vemos cuánto de lejos estamos del mínimo

4. Actualizamos w y b para la siguiente iteración

$$w = w - \alpha \frac{\delta J(w, b)}{\delta w}$$
$$b = b - \alpha \frac{\delta J(w, b)}{\delta b}$$

Learning rate (α): parámetro que determina el salto, definido por nosotros.

5. Acaba el algoritmo cuando alcanzamos la convergencia



