# I Introducción

## Fundamentos Matemáticos
## Función de Pérdida y Descenso de Gradiente

Odin Eufracio

# Motivación

Algoritmos clásicos en **aprendizaje máquina** generalmente usan métodos de **optimización**, como descenso de gradiente, para la etapa de **entrenamiento o aprendizaje.**

Las redes neuronales, ***neural networks* (NN)**, son **modelos paramétricos** que también usan métodos de optimización en su etapa de aprendizaje.

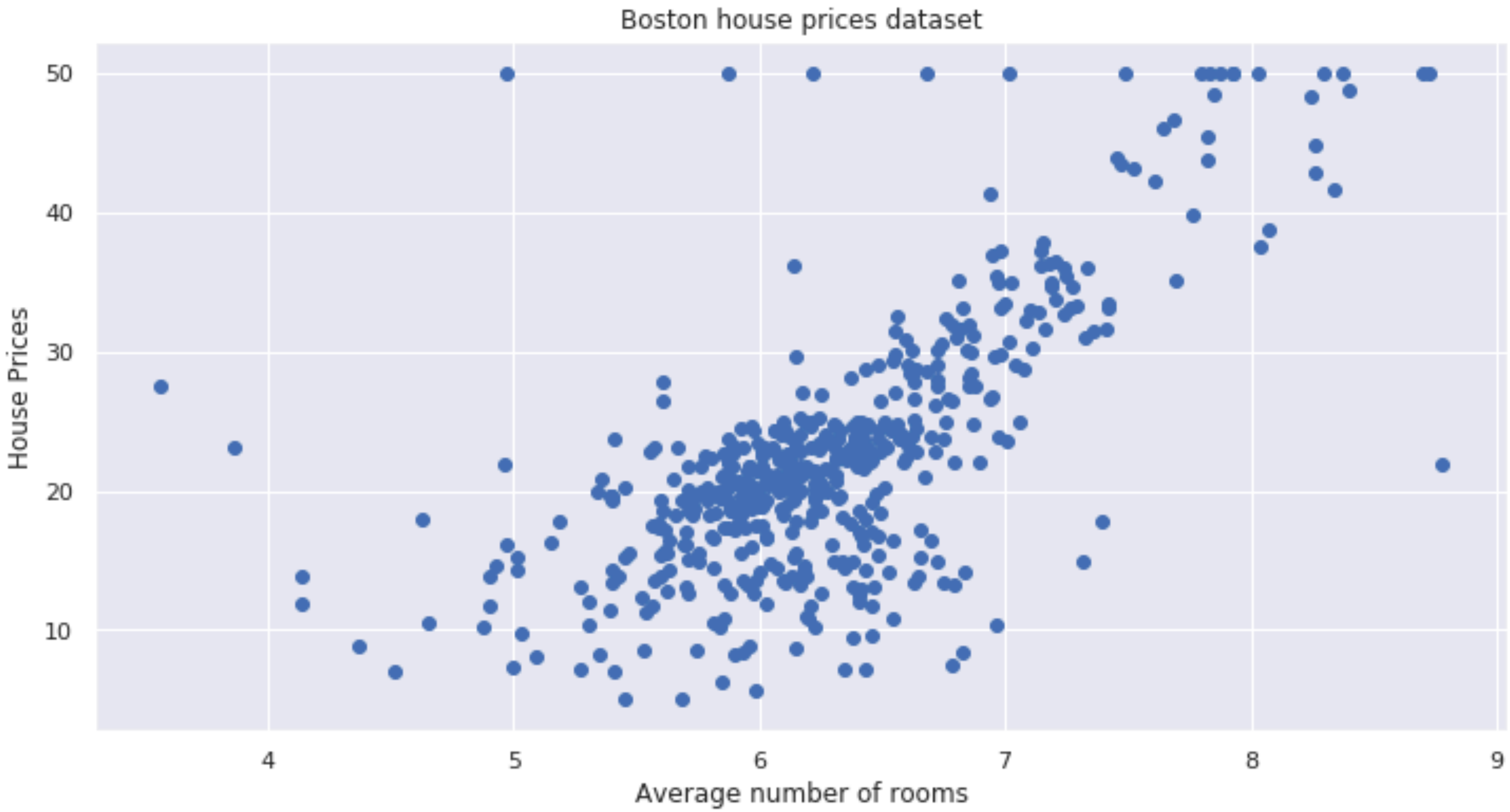
En esta primera sección, implementaremos un modelo de **Regresión Lineal y Logística** para recordar conceptos claves como:

- Modelo paramétrico.
- Funciones de activación y pérdida.
- Optimización (descenso de gradiente)
- Aprendizaje o entrenamiento.

Más adelante veremos que las NN, **deep learning (DL)**, puede verse como un **conjunto apilado de modelos simples** como la Regresión Lineal o Logística.

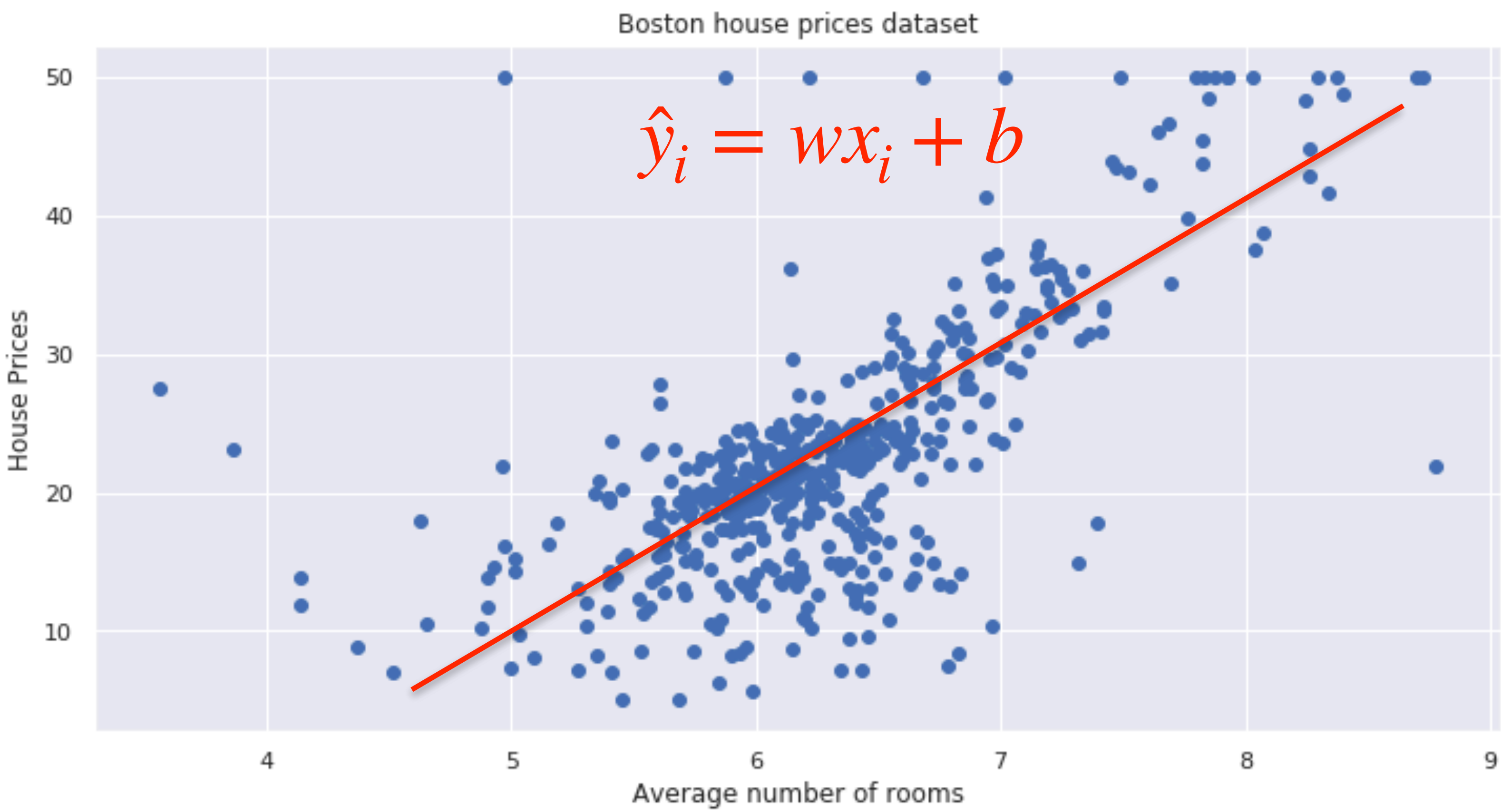
# Regresión Lineal

# Regresión Lineal: Boston house prices data set




Cómo podemos modelar estos datos?

# Regresión Lineal: Boston house prices data set




Cuál es el modelo?

Cuáles son los parámetros del modelo?

Cómo estimamos los parámetros (óptimos) del modelo propuesto?

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

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

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

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

Formulamos el problema de optimización

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

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

Formulamos el problema de optimización

$$\min_{w,b} \; J(w, b)$$

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

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

Formulamos el problema de optimización

$$\min_{w,b} \; J(w, b)$$

Ahora, cómo resolvemos este problema de optimización?

# Regresión Lineal: Boston house prices data set

Definimos una función de costo o función de pérdida (***Loss Function***)

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

Formulamos el problema de optimización

$$\min_{w,b}\ J(w,b)$$

Ahora, cómo resolvemos este problema de optimización?

Búsqueda en linea, ***gradient descent (GD).***

$$x^{t+1} = x^t + \alpha p^t$$

Usando GD, nuestro modelo **aprenderá** los parámetros óptimos.

Vamos a **entrenar** nuestro modelo.

# Regresión Lineal: Boston house prices data set

Búsqueda en linea, ***gradient descent (GD).***

$$x^{t+1} = x^t + \alpha p^t$$

GD es un **método iterativo**, donde el número de iteraciones se le conoce como ***epochs.***

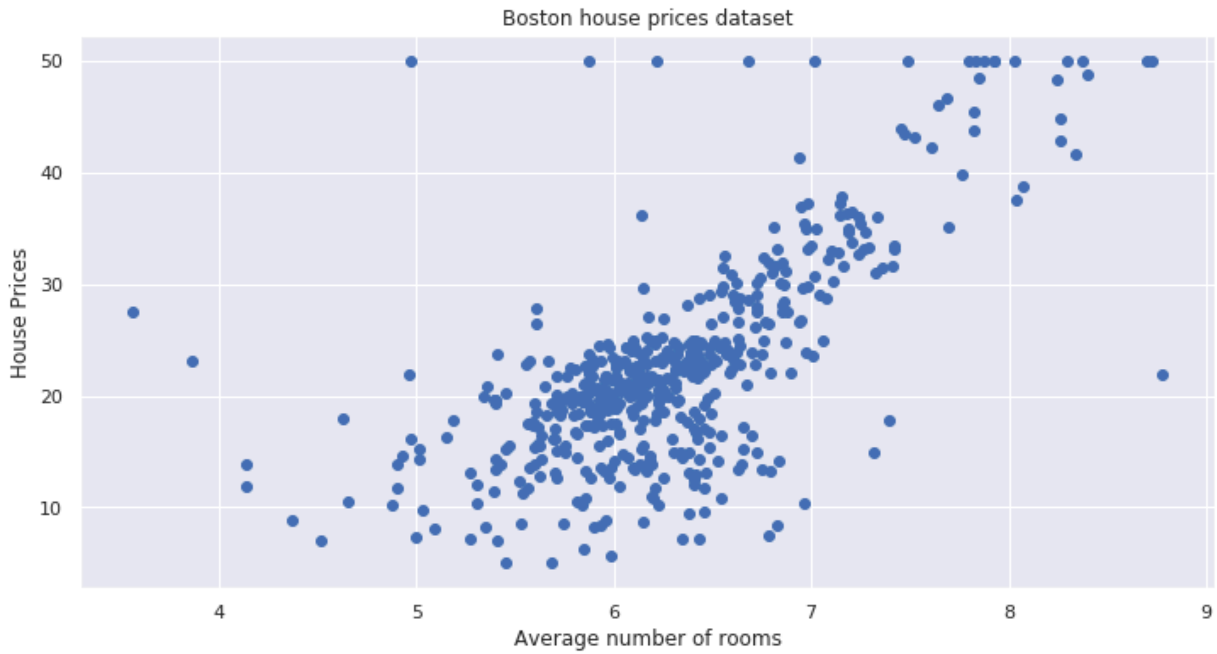
El vector $p$ se le conoce como la **dirección de descenso**, donde el negativo del gradiente de la función $-\nabla J$ es la **dirección de máximo descenso**.

$$p^t = -\nabla J(x^t)$$

En cada iteración, $x$ se mueve *un poco* en la dirección de $p$**,** donde $\alpha$ es el tamaño de paso, ***learning rate***, en que se moverá x.

# Actividad1

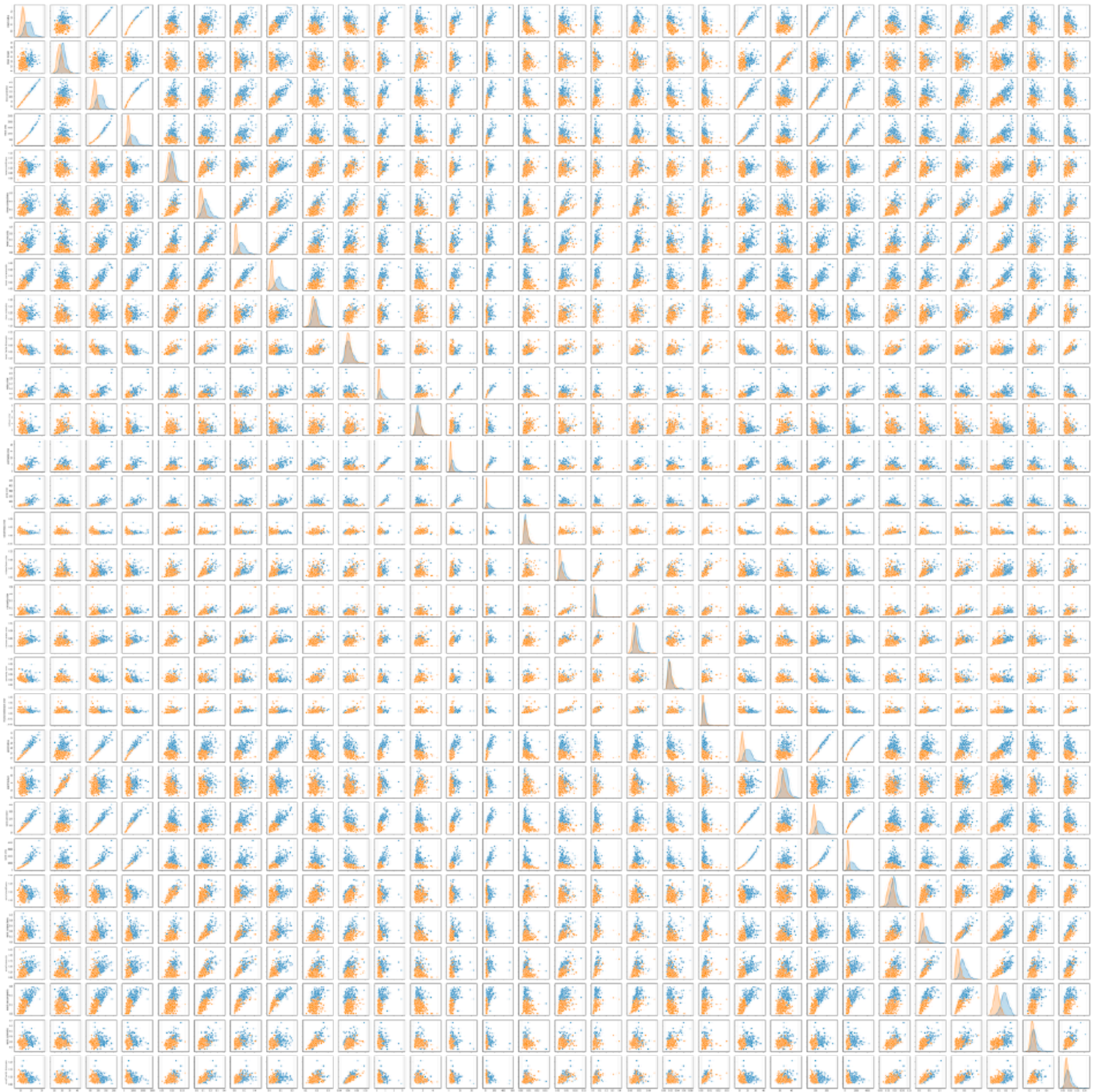
Terminar el notebook **1.1_RegresionLineal.ipynb**



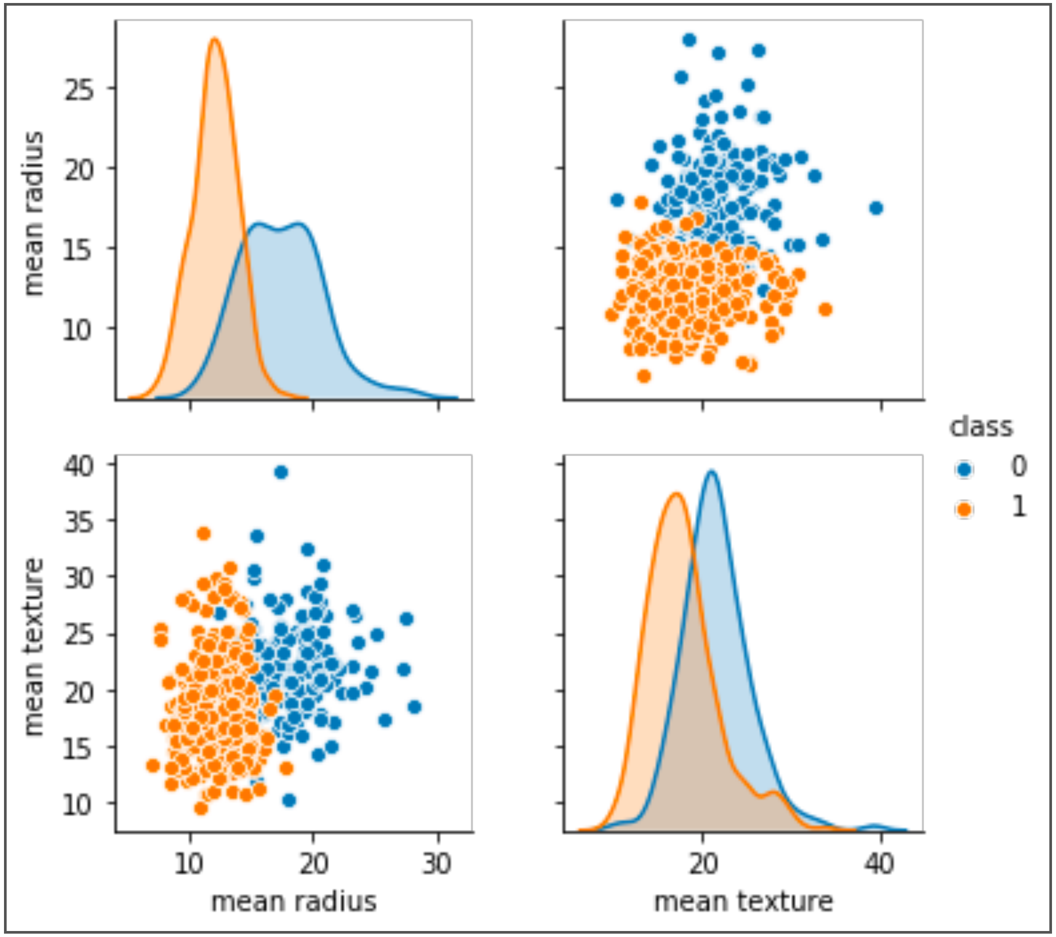

Retroalimentación.

# Regresión Logística

# Regresión Logística: Breast cancer wisconsin dataset

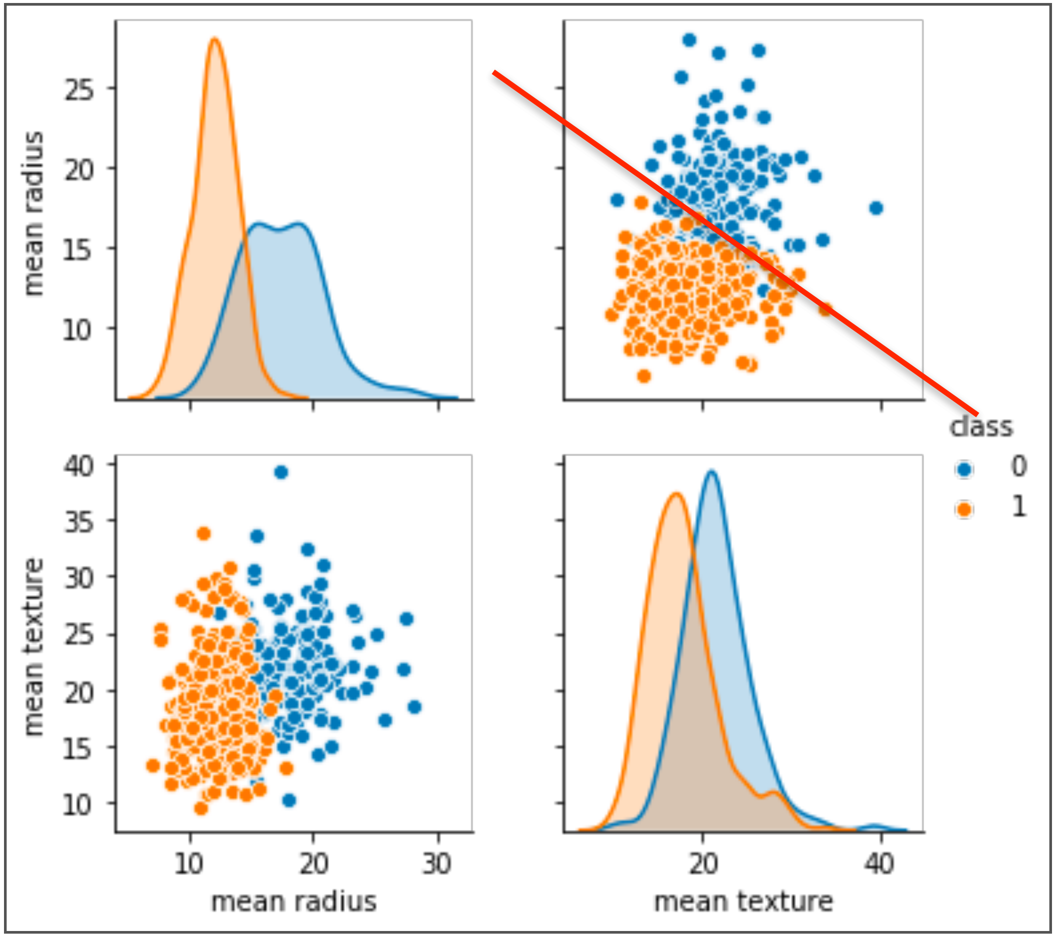


# Regresión Logística: Breast cancer wisconsin dataset



# Regresión Logística: Breast cancer wisconsin dataset

$$\hat{y}_i = \sigma(w^T x_i + b)$$

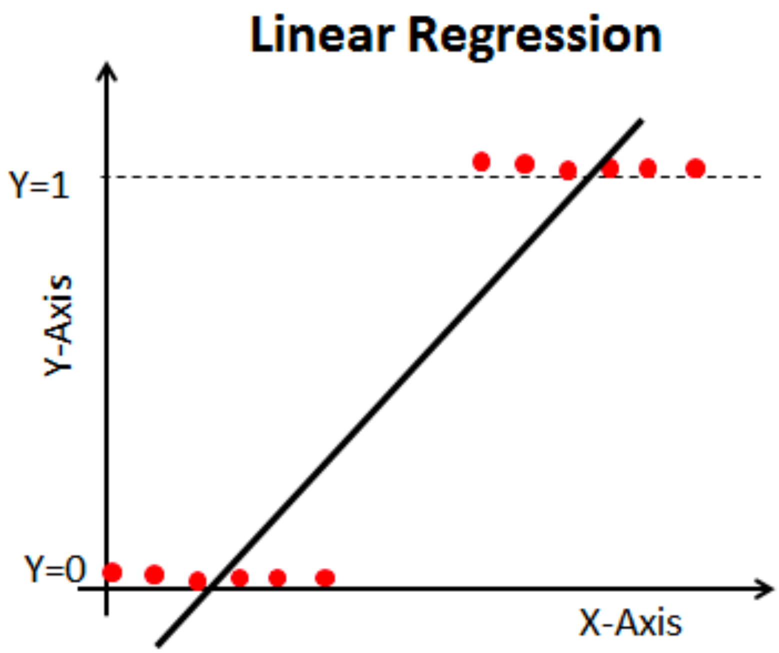


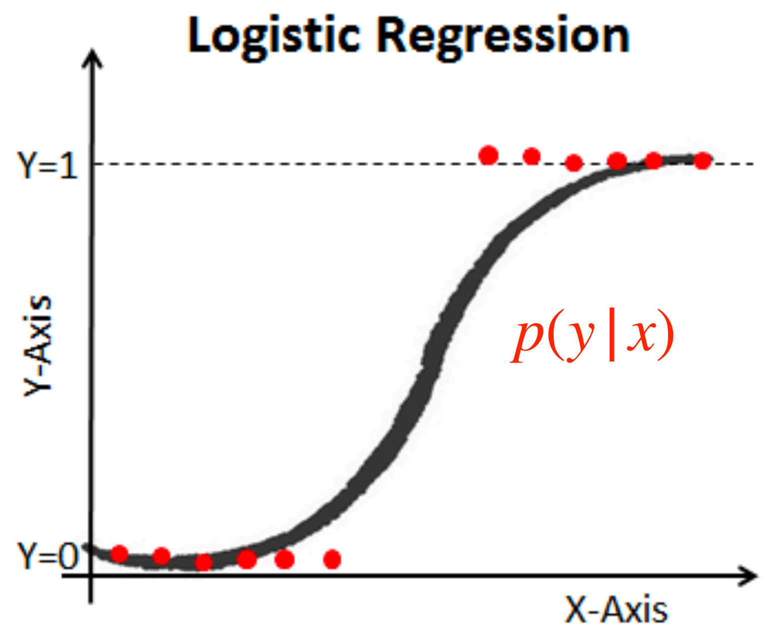
# Regresión Logística: Breast cancer wisconsin dataset

$$\hat{y}_i = wx_i + b$$

$$\hat{y}_i = \sigma(w^T x_i + b)$$

$$\sigma(w^t x + b) = \frac{1}{1 + e^{-(w^t x + b)}}$$



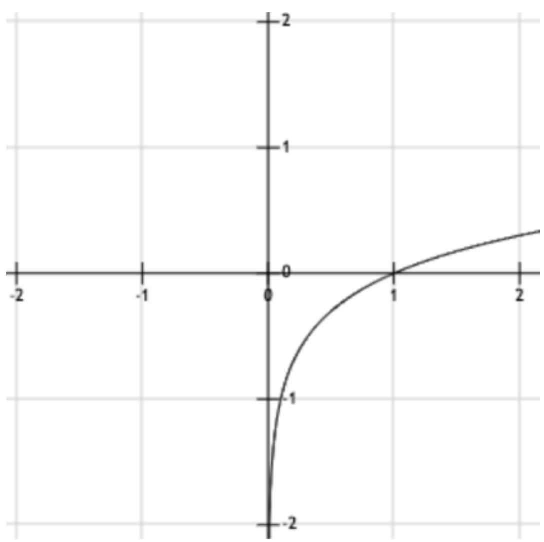




[Imagen tomada de ] https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

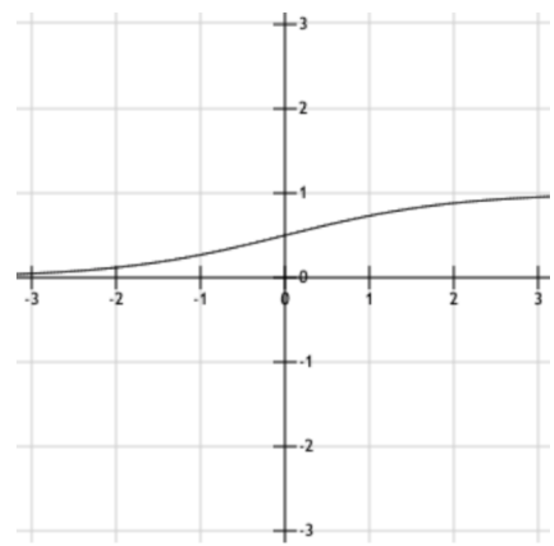
# Regresión Logística: Breast cancer wisconsin dataset

De igual forma, definimos una función de pérdida (***Loss Function***), con $\sigma(x)$ como una función de activación (***activation function***)

$$J(w, b) = \frac{1}{2}\sum_{i=1}^{n} - y_i \log[\sigma(w^t x_i + b)] - (1 - y_i)\log[1 - \sigma(w^t x_i + b)]$$



$\log(x)$



$$\frac{1}{1 + e^{-x}}$$

**Odin Eufracio**

Centro de Investigación en Matemáticas - CIMAT
Jalisco SN, Mineral de Valenciana Gto. Gto.

Office: D307
Phone: (+52) 473 732 7155  ext. 4730
E-Mail: odin.eufracio@cimat.mx