

# Machine Learning

### **Temáticas:**

- Machine Learning.
- Regresión Lineal.
- Regresión Logística.

## Machine Learning

**El aprendizaje**, es quizás el más común de los tipos de algoritmos de Machine Learning. El aprendizaje supervisado consiste en aproximar una respuesta a partir de datos de entrada y respuestas conocidas.

**Python** es un lenguaje de programación de alto nivel orientado a objetos creado por Guido Van Rossum, un científico de la computación y matemático holandés que creó Python y se lo regaló al mundo en 1989. El lenguaje se publicó en 1991 y desde entonces se ha vuelto uno de los lenguajes de programación más utilizados y el más apreciado por muchas personas alrededor del mundo. Algunas de las características más atractivas de Python son:



# Machine Learning

**Aprendizaje supervisado** tiene muchas aplicaciones. Sin embargo, vamos a considerar dos formas de aprendizaje supervisado inicialmente.

**Clasificación:** El objetivo en clasificación es tomar un vector de entrada y asignarlo a una de las clases discretas , donde . En el escenario más común las clases se asumen como disjuntas. El espacio de entrada entonces, es dividido en fronteras de decisión o superficies de decisión.

**Regresión:** El objetivo principal de la regresión es predecir el valor de uno o más targets continuos, dado el valor de un vector de entrada.



**Youtube:** Youtube usa Python para diferentes propósitos: control de los videos, templates para los sitios web, acceder a datos, entre otros.

## ¿Quién usa Machine learning con Python?



**Youtube:** Youtube usa Python para diferentes propósitos: control de los videos, templates para los sitios web, acceder a datos, entre otros.



**Google App Engine:** Google App Engine es una muestra eminente de aplicaciones escritas en Python, permite crear aplicaciones web con lenguaje de programación Python, utilizando su rica colección de librerías y frameworks.



**NASA** Utilizan NASA desde hace más de dos décadas debido a que es fácil realizar mantenimiento del código, es flexible, versátil y tiene una curva de aprendizaje rápida.

## REGRESIÓN LINEAL

La regresión es un tipo de aprendizaje supervisado que busca predecir el valor de un blanco de salida continuo a partir de un vector de entrada. Y para realizar aprendizaje supervisado, es necesario decidir cómo representar funciones de hipótesis.

Inicialmente, en el caso de funciones lineales, es posible representar las funciones de hipótesis de la siguiente manera.

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

## REGRESIÓN LINEAL

Los parámetros, también llamados pesos (weights). También es posible escribir lo anterior de la siguiente manera.

$$h_{\theta}(x) = \sum_{i=0}^d \theta_i x_i = \theta^T x$$

Ahora, es importante en un conjunto de entrenamiento, escoger la función de costo que permita hacer que la hipótesis sea tan cercana a 'y' como sea posible para obtener unos parámetros apropiados. Definimos una función de costo:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^n (h_{\theta}(x^{(i)}) - y^{(i)})^2$$





## ¿Qué hacemos con los datos?



## Programando una regresión lineal

```
import numpy as np
import matplotlib.pyplot as plt

def linReg(X, y, theta=0, alpha=0.01, num_iters=100):
    m = y.size
    costs=[]
    for i in range(num_iters):
        y_hat = np.dot(X, theta)
        costi = 1.0/2.0 * sum((y_hat-y)**2)
        costs.append(costi)
        theta = theta - alpha * (1.0/m) * np.dot(X.T, y_hat-y)
    return theta, np.array(costs)
```

Funciones  
Variables  
Parámetros

## Programando una regresión lineal

```
#Generamos un conjunto de datos sintético de prueba.
```

```
x=np.linspace(-1,5,10)
```

```
y=7*x
```

```
y=y+(np.random.rand(10)*2-1)*7
```

```
#Realizamos el ajuste
```

```
t,c=linReg(x,y)
```

```
#Graficamos el ajuste
```

```
plt.plot(x,t*x,color='k',label='fit')
```

```
plt.scatter(x,y,color='blue',label='datos')
```

```
plt.legend()
```

```
plt.title('Regresión lineal, descenso de gradiente')
```

```
plt.show()
```

```
#Graficamos la función de costo en función de las iteraciones
```

```
plt.plot(c)
```

```
plt.xlabel('Iteración')
```

```
plt.ylabel(r'$J(\theta)$')
```

```
plt.title('Función de costo')
```

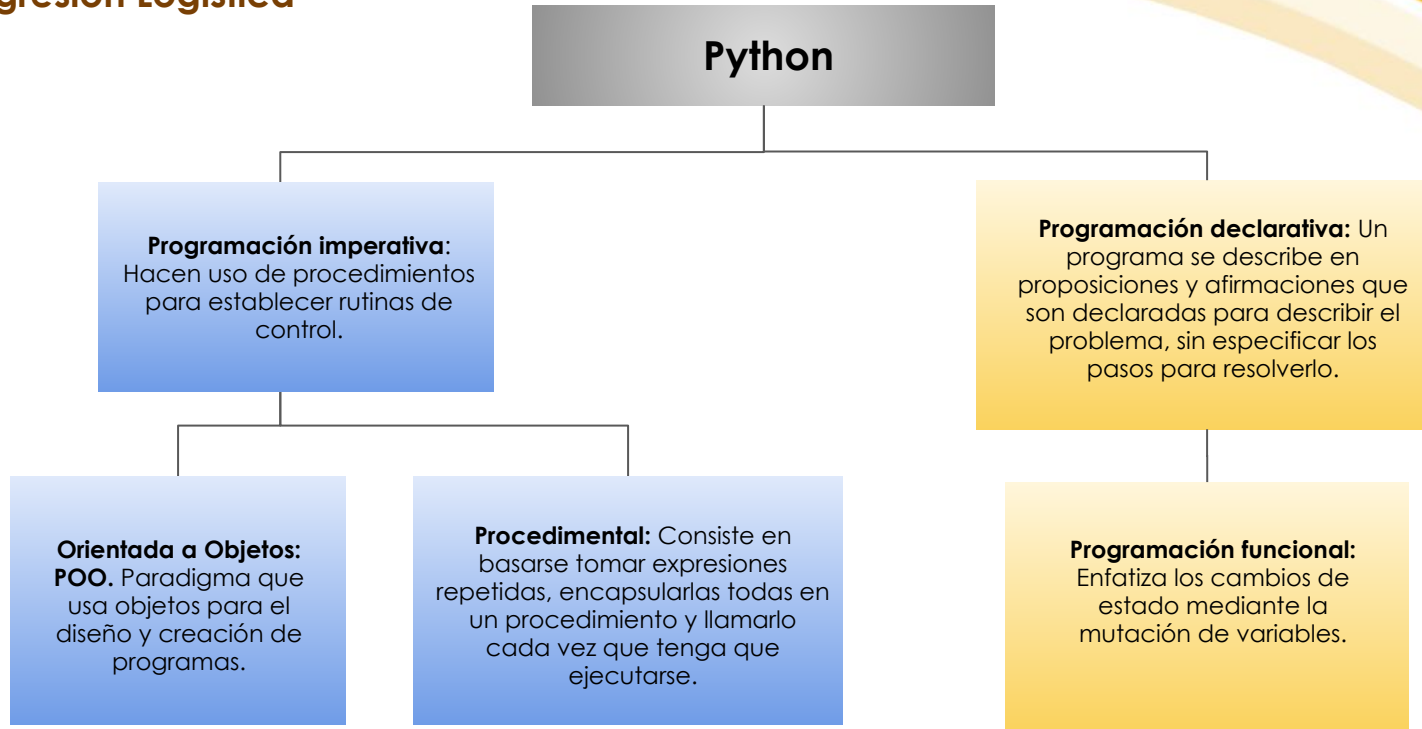
```
plt.show()
```

Funciones

Variables

Parámetros

# Regresión Logística



## Regresión Logística

La regresión logística, es un modelo lineal para clasificación no para regresión. La regresión logística también es conocida como regresión logit o log-lineal. En este modelo, las probabilidades que describen las posibles salidas de un ensayo único se modelan utilizando la función logística.



## Regresión Logística

El problema de clasificación es similar al problema de regresión excepto que los valores de  $y$  únicamente pueden tomar un número finito de valores discretos.

Para solucionar el problema de clasificación, es posible partir de una hipótesis diferente, para ello, es posible partir del hecho que se busca una salida que puede estar únicamente entre 0 y 1. Una buena escogencia es la siguiente:

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

## Regresión Logística

Esta función se conoce como función sigmoideal, y tiende a 1 cuando  $z$  va hacia infinito y a 0 cuando  $z$  va a menos infinito.

$$g(z) = \frac{1}{1 + e^{-z}}$$

