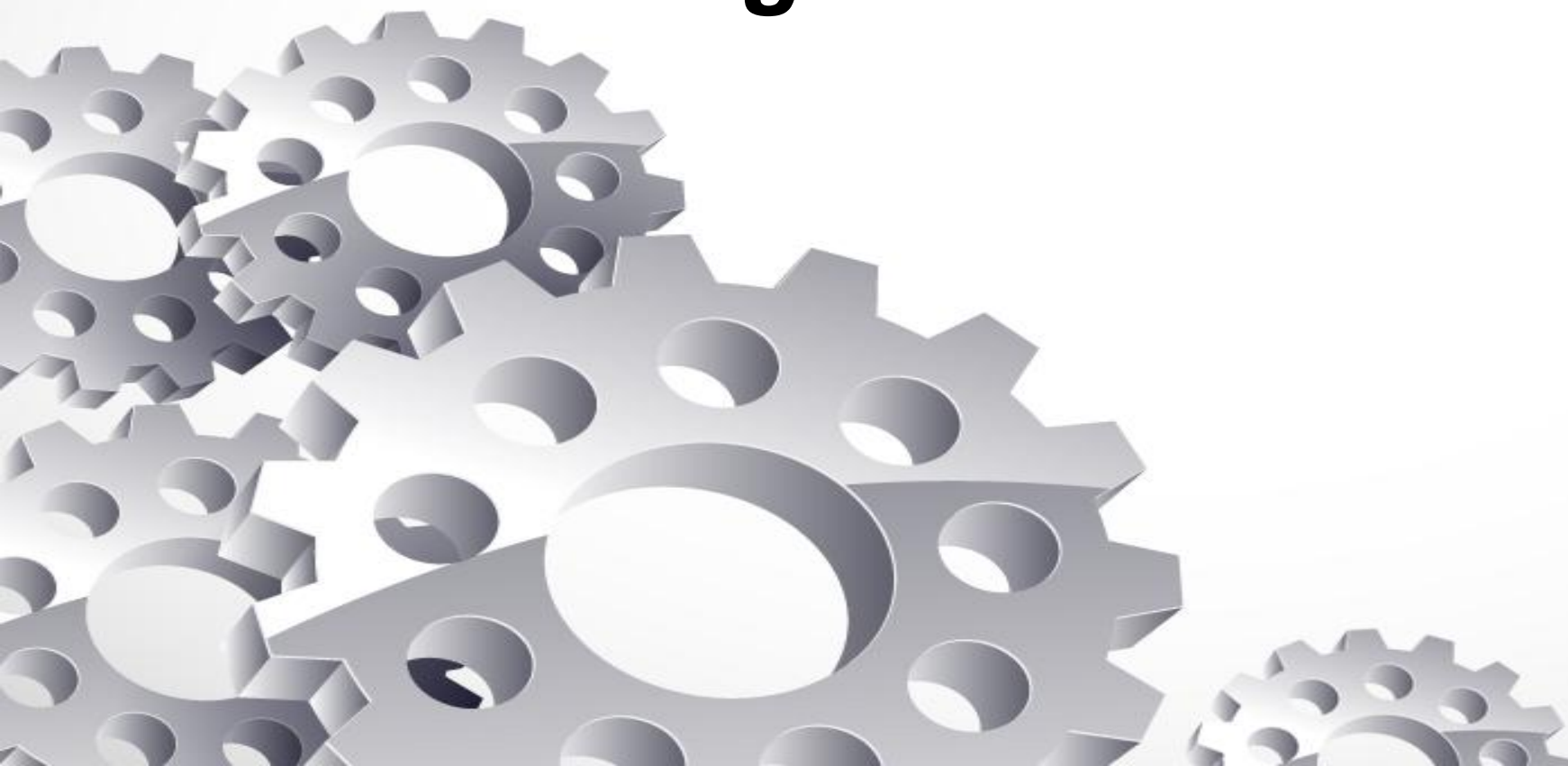


# ***Curso Introducción a la inteligencia artificial con Python***

Isabel Maniega



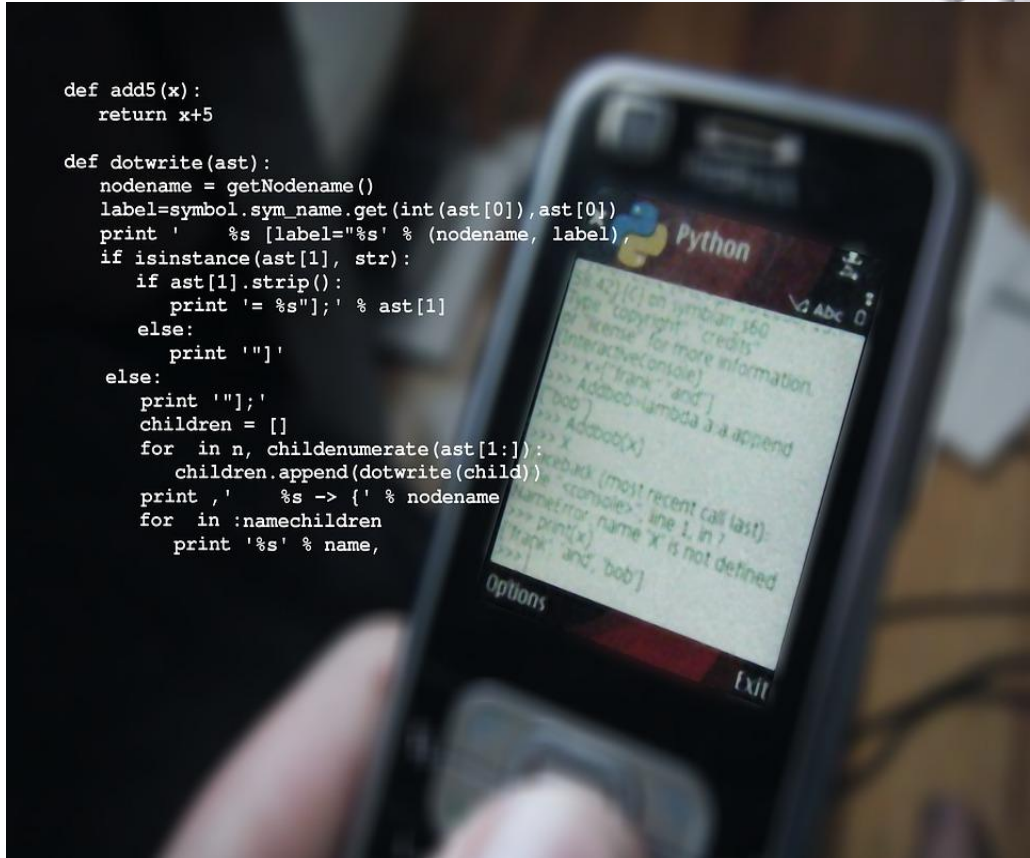
# Curso Introducción a la inteligencia artificial con Python



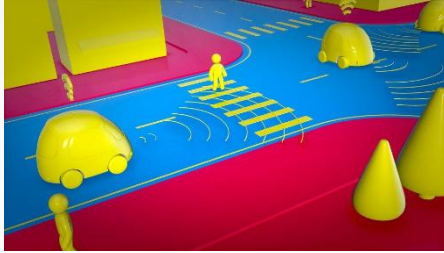
- Introducción a la IA y Python
- Fundamentos de Aprendizaje Automático con Python
- Introducción al Aprendizaje Profundo y técnicas avanzadas
- Proyectos prácticos y ética en IA
- Herramientas y plataformas de vanguardia

- 
- ```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s" % (nodename, label)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '=' %s";" % ast[1]
        else:
            print ""'
    else:
        print "];"
        children = []
        for n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print '    %s -> {' % nodename
        for n, namechildren
            print '%s' % name,
```



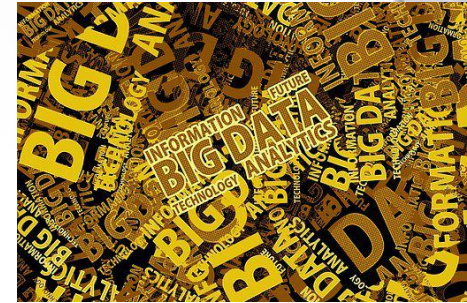
# Salidas Profesionales Python



Vehículos de conducción autónoma



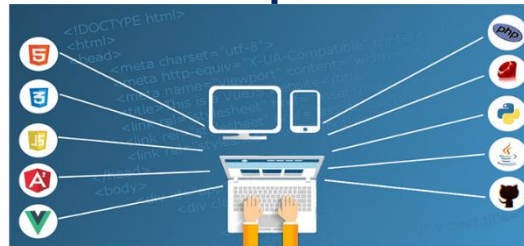
Detección de fraude



Big Data



Sistemas de recomendación



Frameworks webs



Deep Learning



# Fundamentos de Python

Python es un lenguaje de programación de alto nivel, interpretado, orientado a objetos y de uso generalizado que se utiliza para la programación de propósito general.

Un **lenguaje de alto nivel** es similar al lenguaje natural ya que usa símbolos, palabras y convenciones legibles para los humanos y nos permite expresar comandos a computadoras que son mucho más complejas que la lista de instrucciones (IL) que estas entienden.

Es importante conocer que hay dos formas diferentes de transformar un programa de un lenguaje de programación de alto nivel a un lenguaje de máquina:

- **Compilación** - El programa fuente se traduce una vez obteniendo un archivo que contiene el código máquina (por ejemplo, un .exe si se trabaja en windows) y es este el archivo que se distribuye. Cada vez que se modifica el código fuente debe repetirse esta traducción. El programa que realiza la traducción se llama compilador o traductor.
- **Interpretación** - El programa fuente se traduce a código máquina cada vez que se ejecuta el programa. El programa que realiza este tipo de transformación se denomina intérprete ya que interpreta el código cada vez que se ejecuta. El usuario final necesitará el código fuente más el intérprete para poderlo ejecutar.

# Fundamentos de Python



|             | COMPILACIÓN                                                                                                                                                                                                                                                                                                                                                                                              | INTERPRETACIÓN                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VENTAJAS    | <ul style="list-style-type: none"><li>• La ejecución del código traducido suele ser más rápida.</li><li>• Solo el programador debe tener el compilador; el usuario final puede usar el código sin él.</li><li>• El código traducido se almacena en lenguaje máquina, ya que es muy difícil de entender, es probable que tus propios inventos y trucos de programación sigan siendo un secreto.</li></ul> | <ul style="list-style-type: none"><li>• Puedes ejecutar el código en cuanto lo completes; no hay fases adicionales de traducción.</li><li>• El código se almacena utilizando el lenguaje de programación, no el de la máquina; esto significa que puede ejecutarse en computadoras que utilizan diferentes lenguajes máquina; no se compila el código por separado para cada arquitectura diferente.</li></ul> |
| DESVENTAJAS | <ul style="list-style-type: none"><li>• La compilación en sí misma puede llevar mucho tiempo; es posible que no puedas ejecutar tu código inmediatamente después de cualquier modificación.</li><li>• Tienes que tener tantos compiladores como plataformas de hardware en las que desees que se ejecute tu código.</li></ul>                                                                            | <ul style="list-style-type: none"><li>• No esperes que la Interpretación incremente tu código a alta velocidad: tu código compartirá la potencia de la computadora con el intérprete, por lo que no puede ser realmente rápido.</li><li>• Tanto tú como el usuario final deben tener el intérprete para ejecutar el código.</li></ul>                                                                          |

# Fundamentos de Python

Python fue creado por **Guido van Rossum**, nacido en 1956 en Haarlem, Países Bajos. Por supuesto, Guido van Rossum no desarrolló y evolucionó todos los componentes de Python.

Aunque puede que conozcas a la pitón como una gran serpiente, el nombre del lenguaje de programación Python proviene de una vieja serie de comedia de la BBC llamada **Monty Python's Flying Circus**.

En el apogeo de su éxito, el equipo de Monty Python estaba realizando sus escenas en vivo para audiencias en todo el mundo, incluso en el Hollywood Bowl.

Dado que Monty Python es considerado uno de los dos nutrientes fundamentales para un programador (el otro es la pizza), el creador de Python nombró el lenguaje en honor al programa de televisión.



# Instalaciones necesarias

Python version 3.11.9: <https://www.python.org/downloads/>

| Python version | Maintenance status                   | First released       | End of support | Release schedule |
|----------------|--------------------------------------|----------------------|----------------|------------------|
| 3.14           | pre-release                          | 2025-10-01 (planned) | 2030-10        | PEP 745          |
| 3.13           | bugfix                               | 2024-10-07           | 2029-10        | PEP 719          |
| 3.12           | bugfix                               | 2023-10-02           | 2028-10        | PEP 693          |
| 3.11           | security                             | 2022-10-24           | 2027-10        | PEP 664          |
| 3.10           | security                             | 2021-10-04           | 2026-10        | PEP 619          |
| 3.9            | security                             | 2020-10-05           | 2025-10        | PEP 596          |
| 3.8            | end of life, last release was 3.8.20 | 2019-10-14           | 2024-10-07     | PEP 569          |

Windows se encuentra directamente en la tienda para su instalación (mejor opción ejecutable):

<https://www.python.org/downloads/release/python-3119/>

Ubuntu viene instalado por defecto con python 3.10.12

<https://docs.python.org/es/3.11/tutorial/>

<https://docs.python.org/es/3.11/using/windows.html>



# Python

- Comprobación de instalación: `python --version`

```
isabel@isabel-SVE1512E1EW:~$ python3 --version
Python 3.8.10
```

- Intérprete de Python:

```
isabel@isabel-SVE1512E1EW:~$ python3
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

- ¿Cómo realizar una instalación de librerías?

Existe una página con todos los repositorios de las librerías esa página es PYPI:

<https://pypi.org/>



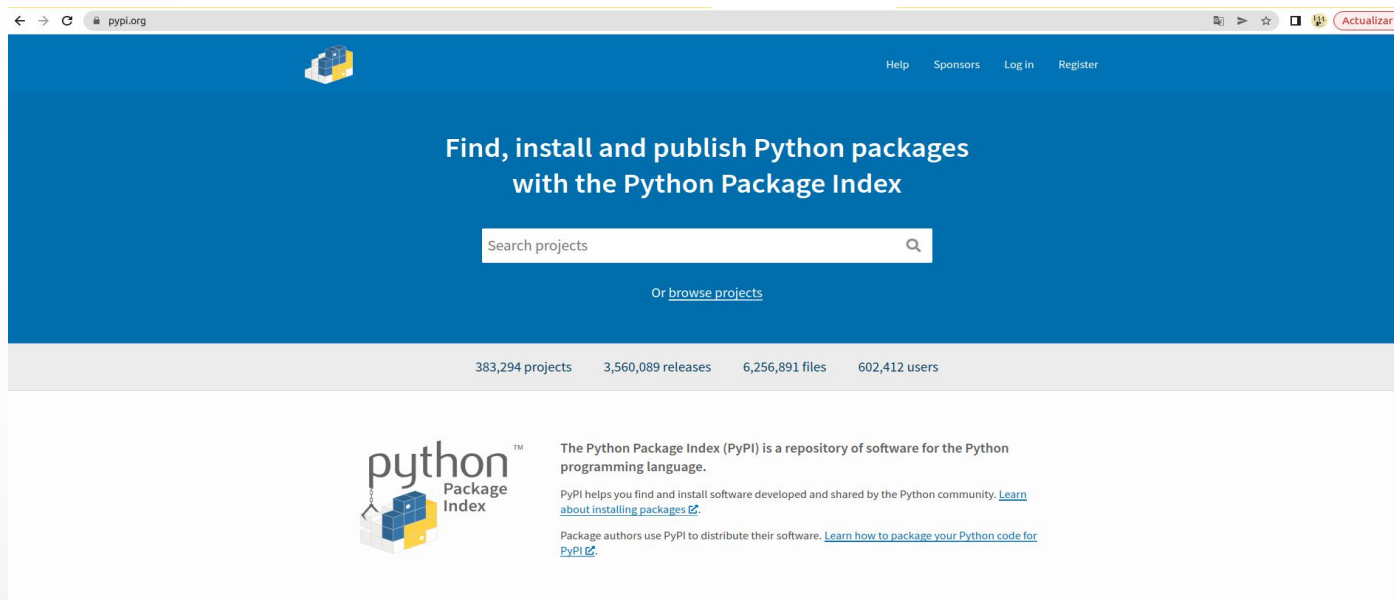
# Python

- ¿Cómo realizar una instalación de librerías?

Existe una página con todos los repositorios de las librerías esa página es PYPI:

<https://pypi.org/>

En esta página podemos encontrar toda la información relativa a documentación, ejemplos, como instalar, versiones disponibles, etc.



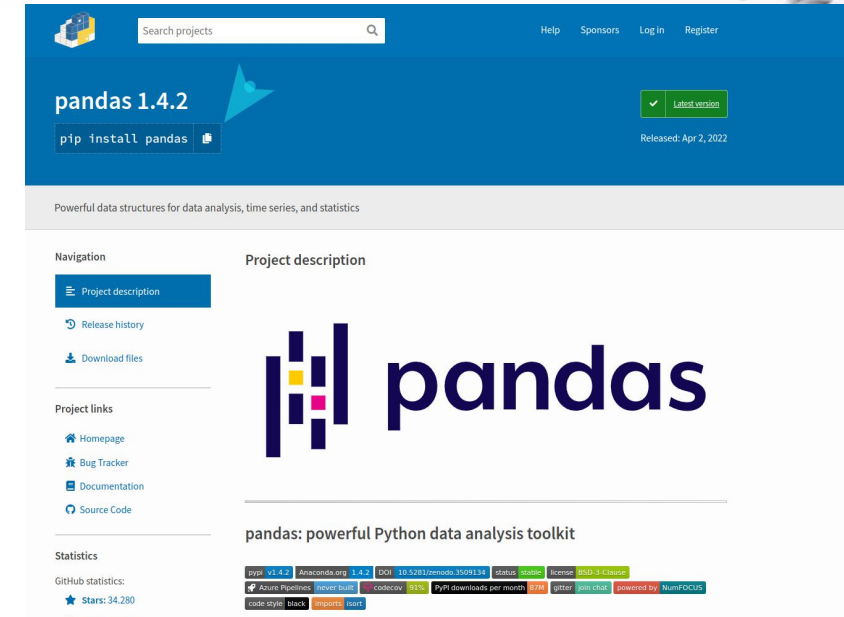
# Python

¿Cómo realizar una instalación de librerías?

1. Para la realización de la instalación iremos a esa página y buscaremos la librería a instalar, copiaremos la instrucción para su instalación tal y como se indica en la imagen:
2. Iremos a la cmd y pegaremos la instrucción y pulsaremos intro:

*En algunos sistemas operativos se pide la instalación de pip, ya que aún no está instalada*

3. Veremos que nos a realizado con éxito la instalación.



```
(env) isabel@isabel-SVE1512E1EW:~/atom$ pip install pandas
Collecting pandas
  Using cached pandas-1.4.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
Collecting pytz>=2020.1
  Using cached pytz-2022.1-py2.py3-none-any.whl (503 kB)
Collecting numpy>=1.18.5; platform_machine != "aarch64" and platform_machine != "arm64" and python_version < "3.10"
  Downloading numpy-1.22.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.9 MB)
    |████████████████████| 16.9 MB 10.5 MB/s
Collecting python-dateutil>=2.8.1
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.22.4 pandas-1.4.2 python-dateutil-2.8.2 pytz-2022.1 six-1.16.0
(env) isabel@isabel-SVE1512E1EW:~/atom$
```

# Instalaciones necesarias

- Jupyter notebook: <https://jupyter.org/install>



## Jupyter Notebook

Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
jupyter notebook
```





# ***Regresión***

# Regresión Lineal Simple

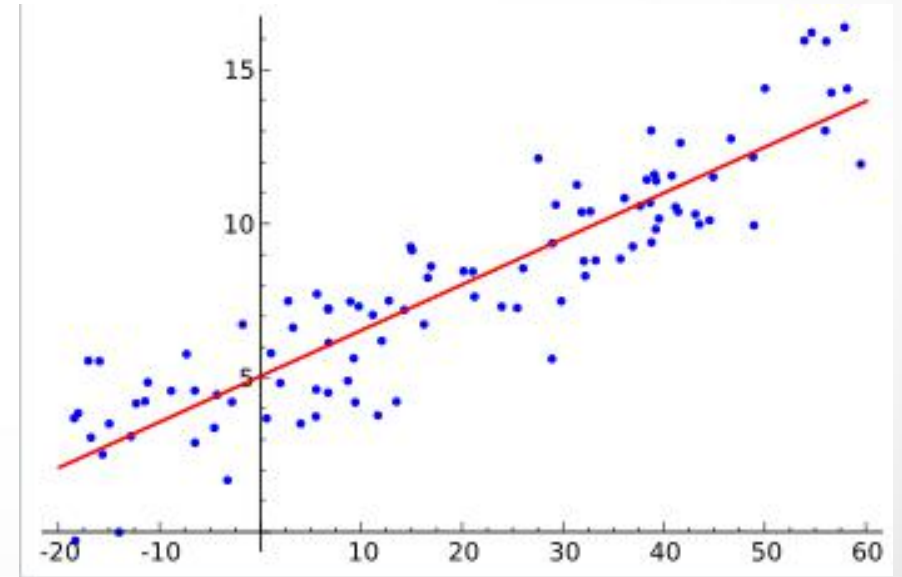
- Basado en la ecuación de la recta:

$$y = b + mx$$

Diagram illustrating the components of the linear regression equation  $y = b + mx$ :

- $y$ : Variable dependiente (Dependent Variable)
- $b$ : Constante (Constant)
- $m$ : Coeficiente (Coefficient)
- $x$ : Variable Independiente (Independent Variable)

Ir al ejemplo práctico Regresión Lineal Simple

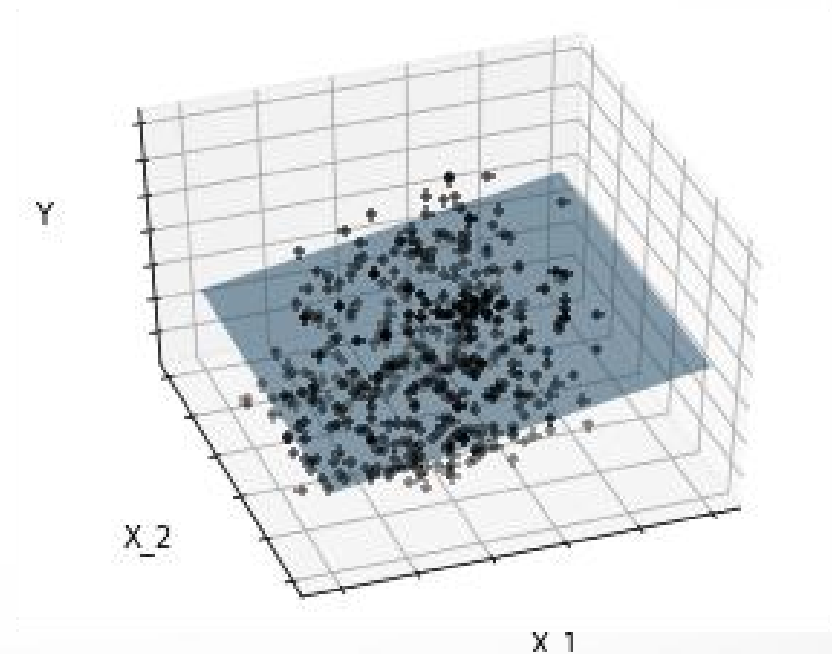


# Regresión Lineal Multiple

- Basado en la ecuación:

$$y = b_0 + m_1 x_1 + m_2 x_2 + \dots + m_n x_n$$

Ir al ejemplo práctico Regresión Lineal Multiple

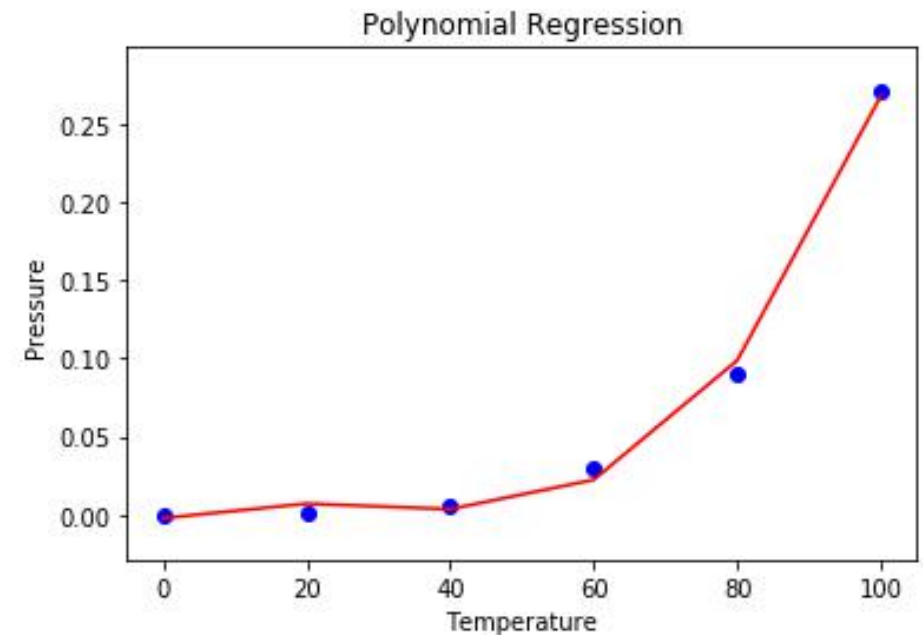


# Regresión Lineal Polinomial

- Basado en la ecuación:

$$y = b_0 + m_1 x_1 + m_2 x_1^2 + \dots + m_n x_1^n$$

Ir al ejemplo práctico Regresión Lineal Polinomial







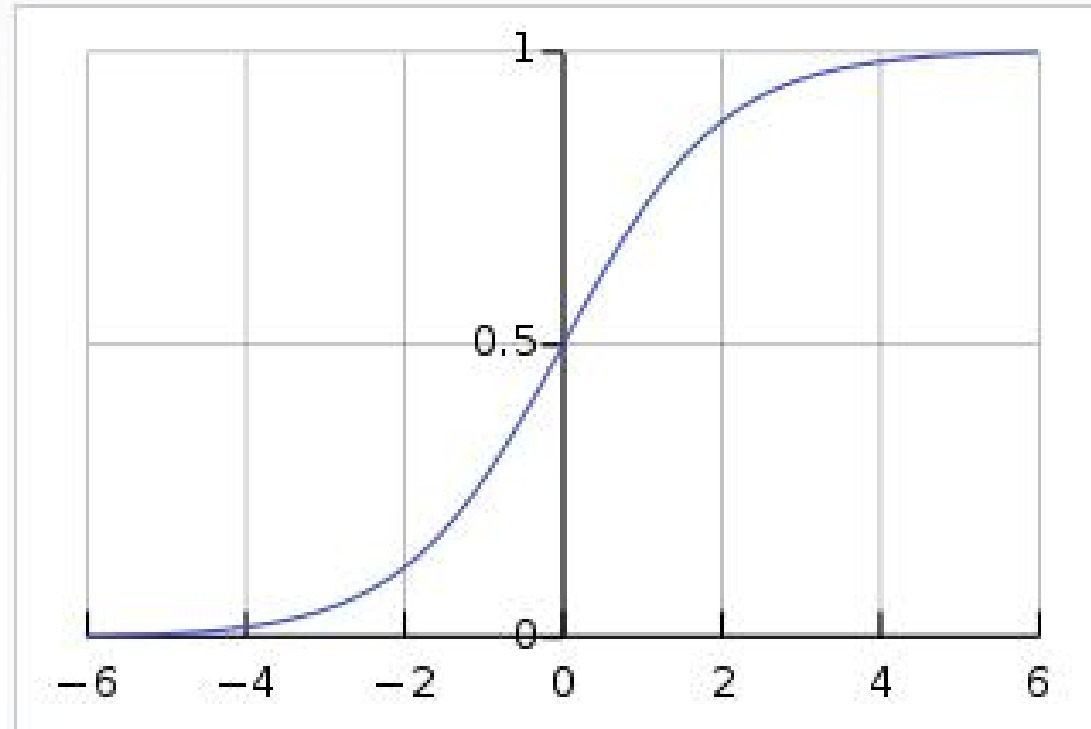
# ***Clasificación***

# Logistic Regression



- Es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras.
- Trata de correlacionar la probabilidad de una variable cualitativa binaria (asumiremos que puede tomar los valores reales "0" y "1") con una variable escalar  $x$ . La idea es que la regresión logística aproxime la probabilidad de obtener "0" (no ocurre cierto suceso) o "1" (ocurre el suceso) con el valor de la variable explicativa  $x$ . En esas condiciones, la probabilidad aproximada del suceso se aproximará mediante una función logística

# Logistic Regression



Función logística con  $\beta_0 + \beta_1 x + e$  en el eje horizontal y  $\pi(x)$  en el eje vertical.

Ir al ejemplo práctico Logistic Regression

[https://es.wikipedia.org/wiki/Regresi%C3%B3n\\_log%C3%ADstica](https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica)

# K-Nearest Neighbors



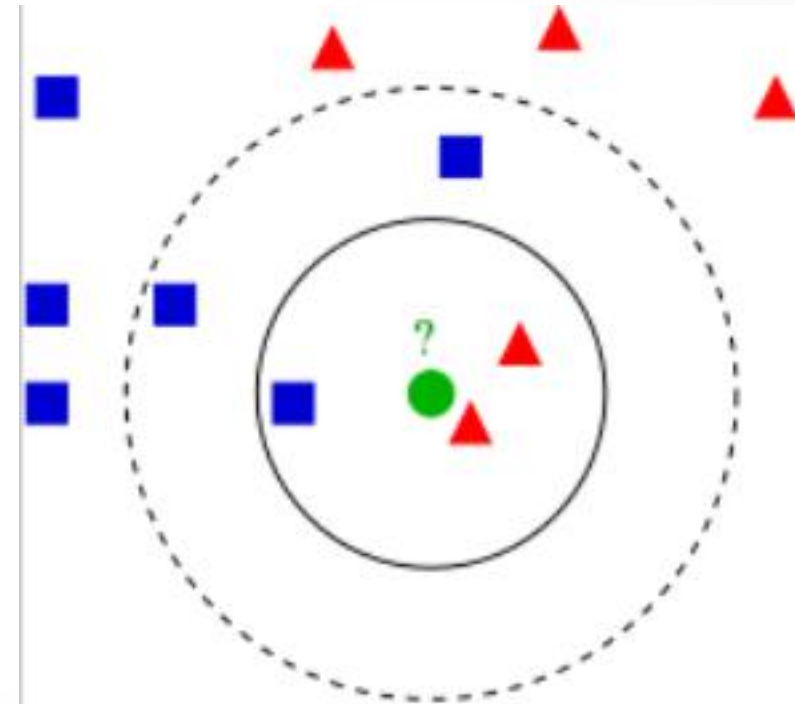
- Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento  $x$  pertenezca a la clase  $C_j$  a partir de la información proporcionada por el conjunto de prototipos.



# K-Nearest Neighbors

- Ejemplo del algoritmo Knn. El ejemplo que se desea clasificar es el círculo verde. Para  $k = 3$  este es clasificado con la clase triángulo, ya que hay solo un cuadrado y 2 triángulos, dentro del círculo que los contiene. Si  $k = 5$  este es clasificado con la clase cuadrado, ya que hay 2 triángulos y 3 cuadrados, dentro del círculo externo.

Ir al ejemplo práctico K-Nearest Neighbors



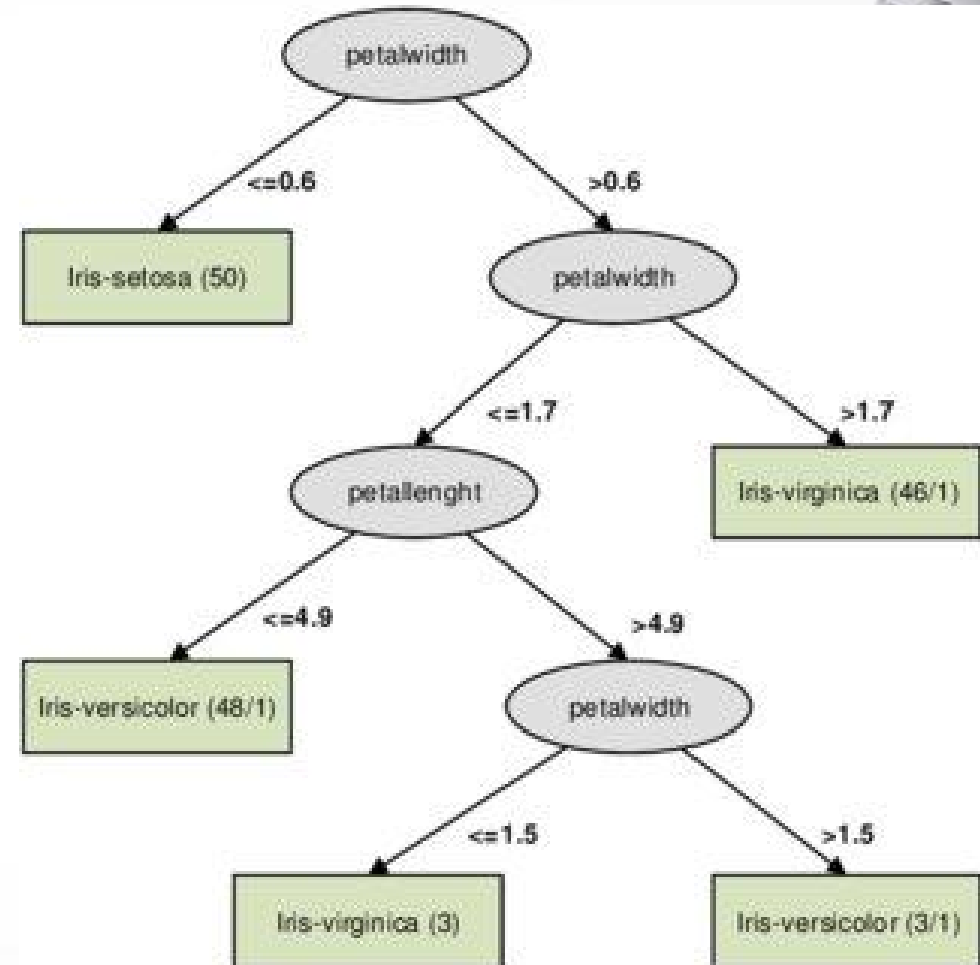
# Decision Tree Classification



- Los modelos de árbol, donde la variable de destino puede tomar un conjunto finito de valores se denominan árboles de clasificación. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase. Los árboles de decisión, donde la variable de destino puede tomar valores continuos (por lo general números reales) se llaman árboles de regresión

# Decission Tree Classification

Ir al ejemplo práctico Decission Tree Clasification





# ***Association Rule Learning***



# Associate Rule Learning

- Las reglas de asociación se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos
- Ejemplo en los datos de ventas de un supermercado, indicaría que un consumidor que compra cebollas y verdura a la vez, es probable que compre también carne.



# Apriori Algoritmo

Procede identificando los ítems individuales frecuentes en la base y extendiéndolos a conjuntos de mayor tamaño siempre y cuando esos conjuntos de datos aparezcan suficientemente seguidos en dicha base de datos.

**PASO 1:**

Elegir la frecuencia para cada uno de los ítems



**PASO 2:**

Coger todos los subconjuntos que forman cada uno de las distintas combinaciones de ítems y su frecuencia



**PASO 3:**

Coger todos los subconjuntos aumentando un ítem cada vez y ver su frecuencia



**PASO 4:**

Repetir el paso 3 hasta que sólo quede un Cluster con los ítems con mayor frecuencia.

# Apriori Algoritmo



## PASO 1:

*Conjuntos:*

{1,2,3,4}  
{1,2,4}  
{1,2}  
{2,3,4}  
{2,3}  
{3,4}  
{2,4}

## PASO 2:

| Ítem | Confianza |
|------|-----------|
| {1}  | 3         |
| {2}  | 6         |
| {3}  | 4         |
| {4}  | 5         |

## PASO 3:

| Ítem  | Confianza |
|-------|-----------|
| {1,2} | 3         |
| {1,3} | 1         |
| {1,4} | 2         |
| {2,3} | 3         |
| {2,4} | 4         |
| {3,4} | 3         |

## PASO 4:

| Ítem    | Confianza |
|---------|-----------|
| {2,3,4} | 2         |

Ir al ejemplo práctico Apriori Algoritmo

# Market Basket Analysis

Análisis de la cesta de mercado, que se usa para observar el comportamiento de compra de los consumidores en los supermercados.

Se basa en el el algoritmo Apriori:

**PASO 1:**

Elegir la frecuencia para cada uno de los ítems

**PASO 2:**

Coger todos los subconjuntos que forman cada uno de las distintas combinaciones de ítems y su frecuencia

**PASO 3:**

Coger todos los subconjuntos aumentando un ítem cada vez y ver su frecuencia

**PASO 4:**

Repetir el paso 3 hasta que sólo quede un Cluster con los ítems con mayor frecuencia.

[Ir al ejemplo práctico Market Basket Analysis](#)

