Aprendizaje automático en Python

Dr. Luis Gerardo de la Fraga

Departamento de Computación
Cinvestav Zacatenco

Correo-e: fraga@cs.cinvestav.mx

10 de noviembre de 2022

¿Qué es aprendizaje automático?

- Trata de extraer conocimiento de los datos.
- Es un área de investigación entre las áreas de:
  - Estadística
  - Inteligencia artificial
  - Ciencias de la computación
- También se conoce como "análisis predictivo" o "aprendizaje estadístico"
- AA es parte de la Inteligencia Artificial

Áreas de AA

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje por reforzamiento

Aprendizaje supervisado

Necesitamos contar con datos y las clases a las que pertenecen.
Existen dos problemas

1. **Clasificación**, donde se tiene una función que regresa un valor discreto, este valor indica a que clase pertenece el patrón desconocido.
2. **Regresión**, necesitamos una función que nos regrese un valor real.

Aprendizaje automático en Python

1. Necesitamos conocer el lenguaje de programación Python
2. Necesitamos conocer los módulos de Python:
3. **Numpy**, para localizar y usar vectores y matrices
4. **Matplotlib**, para realizar gráficas
5. y **Scikit-learn**
   `https://scikit-learn.org/`
6. **OpenCV** es otro sistema con algoritmos de AA
   `https://www.opencv.org/`

Algoritmos para clasificación

1. Modelos lineales
2. Máquinas de vectores de soporte
3. Vecinos más cercanos
4. Bayes ingenuo
5. Árboles de decisión
6. Redes neuronales
7. Métodos de ensamble

Estos mismos métodos pueden usarse para resolver problemas de **regresión**.

- En aprendizaje supervisado necesitamos contar con datos
- Si alguna vez que quieran un problema de AA y les dicen,
  - "no tenemos los datos, pero son muy fáciles de generar"
  - "podríamos generar miles de instancias"
  - "no nos cuesta nada generar más datos"

  El proyecto va a fracasar.
- Si no se tienen los datos, el proyecto se debe dividir precisamente en dos partes:
  1. Recabar la información, los datos.
  2. Procesar los datos. Aplicar las distintas técnicas de AA.

Un primer programa en python

```
print( "¡Hola mundo!" )
```

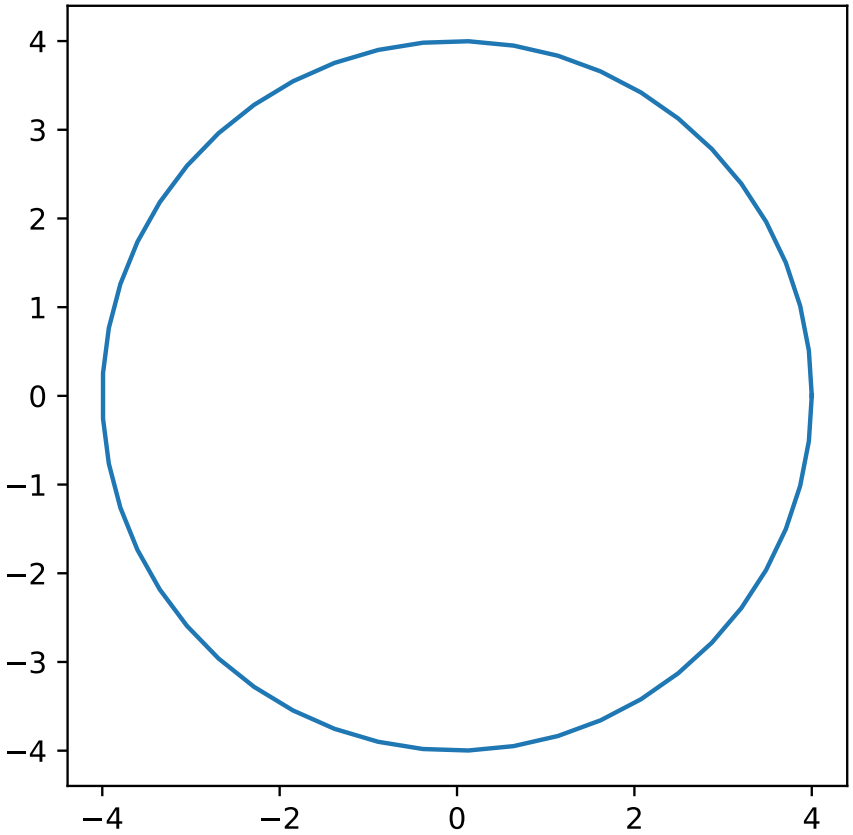
- Si se almacena este programa en el archivo `hola.py`
- Se ejecuta como
  `python3 hola.py`

Numpy

```
import numpy as np

# En python de forma natural existen tres tipos de datos:
# variables, listas y diccionarios. Esta es una lista
L1 = [1, 2, 3, 4, 5]

print( L1 )
print( 2*L1 )

va = np.array( L1 )

print( va )
print( 2*va )
```

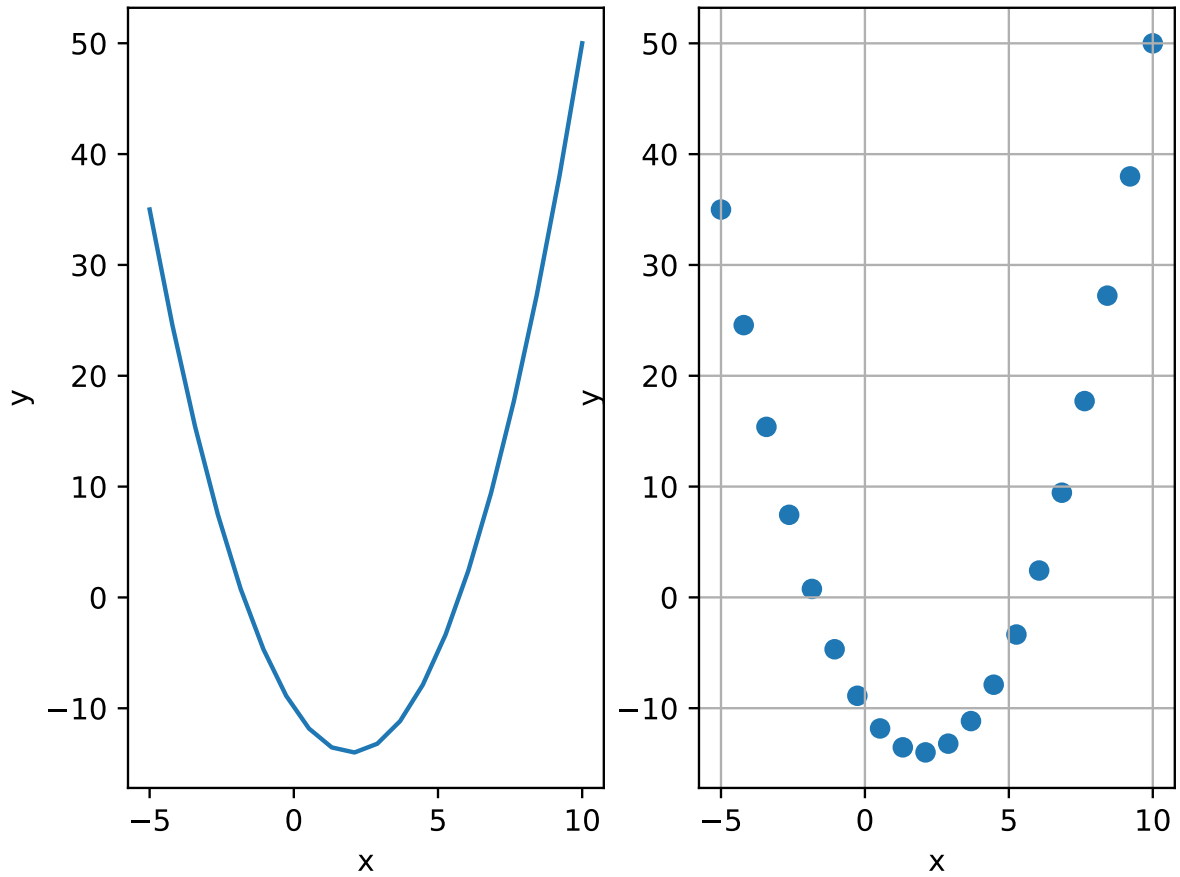
Matplotlib. Gráfica de un círculo

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

# Matplotlib: https://matplotlib.org/

# Gráfica de un círculo en forma paramétrica
#
vt = np.linspace( 0, 2.0 * np.pi, 50 )
vx = 4.0 * np.cos( vt )
vy = 4.0 * np.sin( vt )

fig, ax = plt.subplots( )

ax.plot( vx, vy )
ax.set_aspect( 1 )

plt.show()
# plt.savefig( "circulo.png" )
```


4
3
2
1
0
−1
−2
−3
−4
−4
−2
0
2
4

Matplotlib. Dos gráficas

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

vx = np.linspace( -5, 10, 20 )
vy = vx*vx - 4.0*vx - 10.0

fig, (ax1, ax2) = plt.subplots( 1, 2 )

ax1.set_xlabel( 'x' )
ax1.set_ylabel( 'y' )
ax1.plot( vx, vy )

ax2.set_xlabel( 'x' )
ax2.set_ylabel( 'y' )
ax2.scatter( vx, vy )
ax2.grid( )

plt.show()
```


50
40
30
20
10
0
−10
y
−5
0
5
10
x
50
40
30
20
10
0
−10
y
−5
0
5
10
x

Un problema de regresión

- Tenemos un conjunto de datos $P = \{p_1, p_2, \ldots, p_n\}$
- Vamos a ajustar el modelo $y = a_1 x + a_0$ a los datos
- Cada dato p_i es un vector

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = [x_i, y_i]^{\mathsf{T}}$$

Escribiendo el problema en forma matricial

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

o

$$Xa = y$$

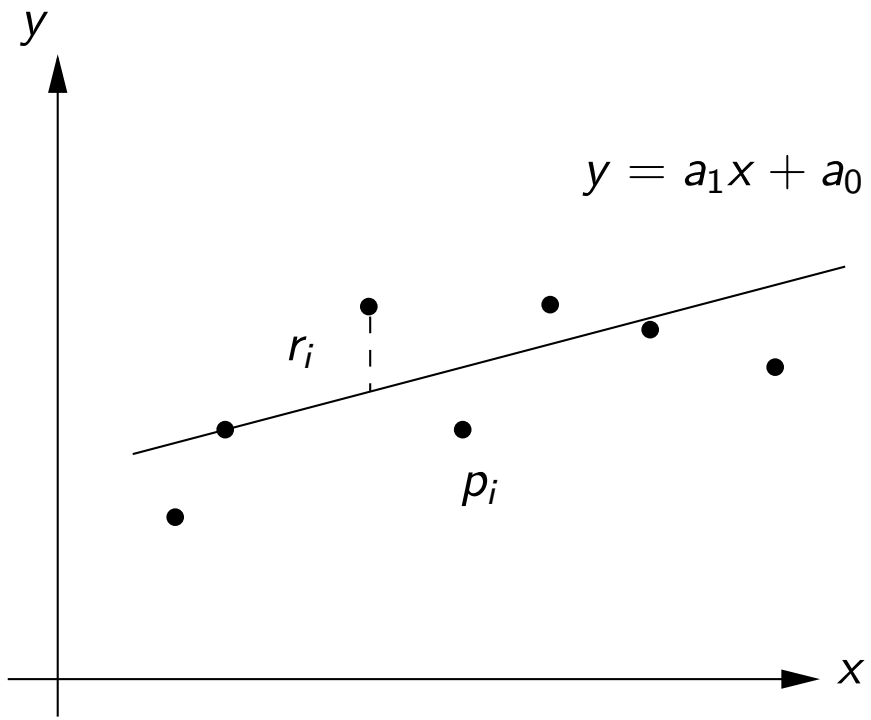
Que se puede resolver si contamos con exactamente dos puntos, y si tenemos más de dos puntos lo resolvermos usando las ecuaciones normales

$$X^{\mathsf{T}} X a = X^{\mathsf{T}} y \tag{1}$$

$$a = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y$$

Si X es una matriz de tamaño $n \times m$, el producto $X^{\mathsf{T}} X$ resulta en una matrix $(m \times n)(n \times m) = m \times m$, que se puede invertir.

El problema de regresión resuelto como un problema de optimización



Definimos el residuo como:

$$r_i = y_i - y = y_i - (a_1x_i + a_0)$$

Podemos calcular el vector a de la incógnitas minimizando

$$s = \sum_{i=1}^{n} r_i.$$

Pero esta suma va cancelando los valores de los residuos.
Podríamos intentar resolver

$$\text{Minimizar } s = \sum_{i=1}^{n} |r_i|.$$

Pero la función valor absoluto no se puede resolver analíticamente.
Podríamos intentar

$$\text{Minimizar } s = \sum_{i=1}^{n} r_i^2.$$

Y aquí si podríamos resolver el problema.

La solución es calcular las derivadas parciales

$$\frac{\partial s}{\partial a_1} \text{ y } \frac{\partial s}{\partial a_0},$$

las igualamos a cero y resolvemos.

$$\begin{aligned}
\frac{\partial s}{\partial a_1} &= \frac{1}{\partial a_1}\sum_{i=1}^{n} r_i^2, \\
&= \frac{1}{\partial a_1}\sum [y_i - (a_1 x_i + a_0)]^2 \\
&= \frac{1}{\partial a_1}\sum (y_i - a_1 x_i - a_0))^2 \\
&= \sum 2(y_i - a_1 x_i - a_0)(-x_i) \\
&= 2\left(-\sum x_i y_i + a_1 \sum x_i^2 + a_0 \sum x_i\right)
\end{aligned}$$

Igualando a cero

$$2\left(-\sum x_i y_i + a_1 \sum x_i^2 + a_0 \sum x_i\right) = 0$$

$$a_1 \sum x_i^2 + a_0 \sum x_i = \sum x_i y_i$$

$$\frac{\partial s}{\partial a_0} = \frac{1}{\partial a_0} \sum_{i=1}^{n} r_i^2,$$
$$= \frac{1}{\partial a_0} \sum [y_i - (a_1 x_i + a_0)]^2$$
$$= \frac{1}{\partial a_0} \sum (y_i - a_1 x_i - a_0))^2$$
$$= \sum 2(y_i - a_1 x_i - a_0)(-1)$$
$$= 2\left(-\sum y_i + a_1 \sum x_i + a_0 \sum 1\right)$$

Igualando a cero

$$2\left(-\sum y_i + a_1 \sum x_i + a_0 \sum 1\right) = 0$$
$$a_1 \sum x_i + n a_0 = \sum y_i$$

Y llegamos al sistema de ecuaciones

$$a_1 \sum x_i^2 + a_0 \sum x_i = \sum x_i y_i$$
$$a_1 \sum x_i + n a_0 = \sum y_i$$

En forma matricial

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

que es exactamente las ecuaciones normales en (1)

Vamos a realizar tres prácticas con tres algoritmos lineales en Scikit Learn

1. Una regresión lineal
2. Clasificación usando un método lineal
3. Un clasificador usando una máquina de vectores de soporte