

IA y aprendizaje automático

Dr. Luis Gerardo de la Fraga

Correo-e: fraga@cs.cinvestav.mx
Departamento de Computación
Cinvestav Zacatenco

19 de agosto, 2023

Contenido:

1. Introducción a python
2. Programación en python
3. Aprendizaje automático con SciKit Learn

¿Por qué usar python?

- ▶ Es un lenguaje de alto nivel
- ▶ Es fácil construir programas rápidamente
- ▶ Es uno de los lenguajes más usados
- ▶ Es el lenguaje sugerido para programar en RaspBerryPi

Lenguajes de alto nivel

- ▶ C es un lenguaje de nivel medio
- ▶ Los lenguajes de alto nivel son interpretados
- ▶ No se definen tipos de datos
- ▶ Tienen una máquina de tiempo de ejecución (para uso de memoria con un recolector de basura)
- ▶ Se puede empotrar funciones en C dentro de python si se necesita eficiencia.

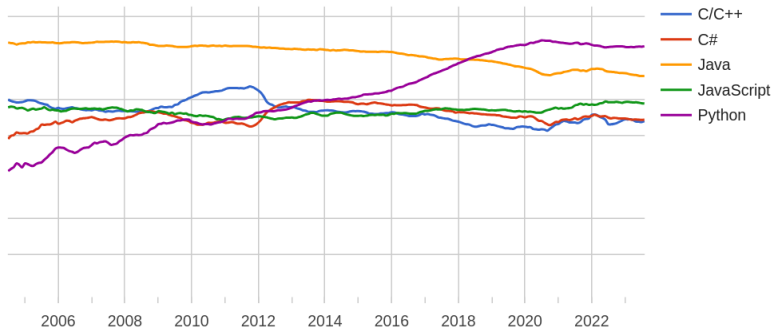
Prototipado rápido

- ▶ Son ideales para realizar programas que procesan texto y generan texto
- ▶ Para generación dinámica de páginas WEB
- ▶ Si se tiene una idea, puede obtenerse un programa funcional en minutos u horas

Posición	Lenguaje	Uso %	Tendencia
1	Python	28.04 %	+0.3 %
2	Java	15.78 %	-1.3 %
3	JavaScript	9.27 %	-0.2 %
4	C#	6.77 %	-0.2 %
5	C/C++	6.59 %	+0.4 %
6	PHP	5.01 %	-0.4 %
7	R	4.35 %	+0.0 %
8	TypeScript	3.09 %	+0.3 %
9	Swift	2.54 %	+0.5 %
10	Objective-C	2.15 %	+0.1 %

<http://pypl.github.io/PYPL.html> al 18 de agosto, 2023.

PYPL Popularity of Programming Language



<http://pypl.github.io/PYPL.html> al 18 de agosto, 2023.

<http://www.raspberrypi.org/>



Se puede comprar en www.newark.com.mx

Formas de usar python

- ▶ Existen dos formas principales para trabajar en python:
 1. A través de una interfaz como jupyter
 2. Usando la línea de comandos y un editor

Jupyter

- ▶ Jupyter es un ambiente interactivo para ejecutar código dentro del navegador WEB
- ▶ Jupyter puede usarse con otros lenguajes de programación
- ▶ Dentro de las notas de jupyter se puede incorporar código, texto e imágenes

https:

`//jupyter-notebook.readthedocs.io/en/stable/notebook.html`

En este taller usaré la línea de comandos

- ▶ Todos los programas de python son archivos
- ▶ Solo se necesita un editor (usaré “vi”) y la terminal
- ▶ Este el ambiente común de desarrollo de Unix o GNU/Linux
- ▶ Esta pensado para incorporar programas dentro de otros programas
- ▶ Documentación en <https://python.org/>

Python en 15 minutos

```
# Mi primer programa
# en python
#
i = 0
while i < 10 :
    i += 1
    print( i )
```

- ▶ Con # se inicia un comentario
- ▶ El inicio y fin de bloque se indican con la indentación
- ▶ No hay tipos
- ▶ La variable i es una variable entera porque está inicializada con un valor entero

```
#  
# Funciones en python  
#  
def f( x ) :  
    v = (x-1.0)*(x-4.0)  
    return v  
  
print( f(4.0) )
```

En un archivo que se llama
funcion.py

```
#  
# Funciones en python  
#  
def f( x ) :  
    v = (x-1.0)*(x-4.0)  
    return v
```

En otro archivo:

```
import funcion  
  
x = -1.0  
while x < 4.1 :  
    print( x, funcion.f( x ) )  
    x += 0.5
```

Paradigmas de programación en python

1. Imperativa
2. En objetos
3. Funcional

Archivo punto.py

```
class Punto :
    def __init__( self, x, y ) :
        self.x = x
        self.y = y

    def __add__( self, A ) :
        p = Punto(self.x+A.x,
                  self.y+A.y)
        return p

    def print( self ) :
        print( self.x, self.y )
```

En el archivo objetos.py

```
import punto

A = punto.Punto( 2, 3 )
A.print()

B = punto.Punto( 1, 2 )
B.print()

C = A + B
C.print()
```


Aprendizaje de máquina

Esta rama de la ciencia en la intersección de

- ▶ Estadística,
- ▶ Inteligencia Artificial, y
- ▶ Ciencias de la Computación

Science Kit Learn

- ▶ Es un software escrito en Python para realizar tareas de aprendizaje automático
- ▶ Disponible en <https://scikit-learn.org/>
- ▶ Resuelve problemas de **aprendizaje supervisado**.
- ▶ También tiene implementado algunos algoritmos de agrupamiento, que es **aprendizaje no supervisado**.

Aprendizaje supervisado

- ▶ Necesitamos datos
- ▶ Los datos los dividimos en datos de entrenamiento y prueba
- ▶ Aplicamos un algoritmo de aprendizaje supervisado como:
 1. Los vecinos más cercanos
 2. Mínimos cuadrados
 3. La regresión logística
 4. Máquinas de vectores de soporte
 5. Árboles de decisión
 6. Redes neuronales
 7. Etcétera

Software de trabajo

1. Python
2. Numpy
3. Matplotlib (<https://matplotlib.org>)
4. Sklearn (<https://scikit-learn.org>)

```
import numpy as np
```

```
A = [1, 2, 3, 4]
```

```
B = 2*A
```

```
print( B )
```

```
A = np.array( [1, 2, 3, 4] )
```

```
B = 2*A
```

```
print( B )
```

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

vx = np.linspace( -2, 8, 30 )
vy = (vx + 1)*(vx - 4)

fig, ax = plt.subplots( )

ax.plot( vx, vy )
ax.grid( )

plt.show( )
```

- ▶ Mi correo-e: `fraga@cs.cinvestav.mx`
- ▶ Mi página personal:
`https://cs.cinvestav.mx/~fraga/`
- ▶ El contenido visto hoy estará disponible en:
`https://cs.cinvestav.mx/~fraga/Charlas`

¡Gracias!