Módulo 0: Python

(Esta guía es opcional)

Objetivo:
*El objetivo de éste módulo es poder repasar la sintaxis del lenguaje Python y la librería Numpy, las cuales serán herramientas muy importantes en los módulos siguientes. Apóyese en el **hands-on-python** del presente curso y recursos en la web.*

Requerimientos:
- Conocimientos de Python básico (y tener Python instalado o utilizar alguna plataforma online de Python en notebooks: Google Colaboratory, Kaggle, etc)
- Uso básico de Git y GitHub.

Python

Ejercicio 1
Calcular el porcentaje que representa el valor 13 sobre un total de 97 y guardarlo en una variable llamada porcentaje.

Ejercicio 2
Con las siguientes variables construir la frase:
“El gran gran gran edificio mide 300 metros de altura”

```
txt1 = 'gran '
txt2 = 'el '
tet3 = 'mide '
txt4 = ' metros'
numero = 300
```

Ejercicio 3
Construir una función de Python que, dados dos números informe el mayor valor.

Ejercicio 4
Un automóvil puede cubrir una distancia de N kilómetros por día. ¿Cuántos días se necesitan para cubrir una ruta de M kilómetros de longitud? Construya una función de Python para resolver el problema.

Ejercicio 5
Escriba una función de python que, dado un número, imprima la frase 'El número es par' si el número es par o la frase 'El número es impar' si no lo es.

Ejercicio 6
Genere una lista llamada cuadrados que contenga el cuadrado de cada elemento de la lista `lst = [0, 1, -1, 3, 4, 8, 20, 5]`

Ejercicio 7
Dada la siguiente lista
`lst = [1, 1, -1, -3, 4, 8, 4, 4, 55, 12, 21, 50, 40, 32, 11, 27]`
Imprima todos los elementos de la lista que son menores a 12.

Ejercicio 8
Cree una función de Python que imprima una sucesión de Fibonacci de *n* elementos.
Recuerde que la secuencia de Fibonacci es la suma de los dos números anteriores y se puede mostrar matemáticamente como $F_n = F_{n-1} + F_{n-2}$ El primer y segundo elementos de la serie son 0 y 1, respectivamente.

Ejercicio 9
Cree una función de Python que reciba un número entero y retorne como resultado el factorial.

Ejercicio 10
Cree una clase llamada Complejo que represente a un número complejo. El constructor debe tener dos parámetros: real, imag. La clase debe tener métodos para sumar, restar, multiplicar y dividir números complejos. Además, debe contar con métodos para setear y obtener la parte real e imaginaria.

Ejercicio 11
Cree una clase llamada Poligono y otra llamada Cuadrado. La clase Poligono será el padre de Triangulo y Cuadrado, y contendrá las propiedades y métodos comunes de ambos (implementar método calcular_perimetro). Ambos tendrán también otra propiedad llamada color.

Numpy

Ejercicio 9
Dados los siguientes vectores: `arr1 = [2, 4, 8]` y `arr2 = [3, 1, 6]`.
Utilizando numpy, multiplicar el **arr1** por el escalar 2 y finalmente obtener la suma resultante de la operación anterior con el **arr2.**

Ejercicio 10
Dados los siguientes vectores: `arr1 = [2, 4, 8]` y `arr2 = [3, 1, 6]`.
Utilizando numpy, calcular la distancia euclídea entre ambos.

Ejercicio 11
Cree un vector de numpy de dimensión 50 con valores enteros aleatorios uniformemente distribuidos entre 0 y 100. Calcule el valor mínimo. máximo, la media y el desvío estándar. Infórmelos al usuario.

Ejercicio 12

Dado el siguiente vector: `arr = [[3, 1, 8], [7, 9, 2]]`
Informe al usuario si los valores 2 y 6 están presentes en el array.

Ejercicio 13
Dado el siguiente vector: `arr = [[0, 7, 2, 6, 0], [0, 5, 1, 0, 7]]`
Utilizando numpy cuente la cantidad de elementos diferentes a cero.

Ejercicio 14
Tod@s estamos familiarizados con la serie de Fibonacci. Cada número de la secuencia es la suma de los dos números que lo preceden. Entonces, la secuencia es: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …
Implementar la fórmula de Binet con numpy para generar la sucesión de Fibonacci

$$f_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

Fórmula de Binet

Ejercicio 15
Utilizando numpy, genera un vector de 50 valores aleatorios de una distribución uniforme. Utilizando la librería matplotlib, graficar la distribución.

Ejercicio 16
Utilizando numpy, genera un vector de 50 valores aleatorios de una distribución gaussiana, cuya media y desvío estándar se encuentre dado por el usuario. Grafique en matplotlib diferentes distribuciones (cambiando la media y/o desvío estándar).

Ejercicio 17
Dado el vector `arr = [23, 11, 8, 7, 9, 2, 5, 1, 1, 3, 9, 6]`
Utilizando numpy, encontrar los *k* valores mas pequeños, donde *k* es un valor dado por el usuario.

Ejercicio 18
Utilizando numpy, genera 2 matrices de dimensión 2 x 2 y calcula el producto interno entre ambas.

Ejercicio 19
Dado el siguiente vector: `arr = [14, 17, 12, 33, 44, 15, 6, 27, 8, 19]`
Utilizando numpy, calcular el percentil 25, 75 y 90 de los datos.

Ejercicio 20
Dados los vectores: `arr1 = [0, 1, 1]` y `arr2 = [2, 2, 1]`

Utilizando numpy, calcular la matriz de covarianza de los 2 vectores dados.

Ejercicio 21
¿Cuál es la diferencia entre un vector de forma -shape- (n,), (n,1) y (1,n)? Cree diferentes vectores para responder esta pregunta.

Integrador: Perceptrón simple

Implementar el algoritmo del Perceptrón simple en una clase que mínimamente conste de un método para entrenamiento y un método para realizar predicciones a partir de entradas dadas.

```
class Perceptron:

    def __init__(self):
        pass

    def train(self):
        pass

    def predict(self):
        pass
```

A. Entrenar un perceptrón simple para intentar resolver los problemas AND, OR y XOR utilizando los datasets que se proveen a continuación. ¿Qué conclusiones puede sacar? ¿De qué depende el resultado? ¿Funciona para todos los casos? ¿Por qué?

```
import numpy as np

# AND
X_train = np.array([[0,0],[0,1],[1,0],[1,1]])
y_train = np.array([[0, 0, 0, 1]]).T

# OR
X_train = np.array([[0,0],[0,1],[1,0],[1,1]])
y_train = np.array([[0, 1, 1, 1]]).T

# XOR
X_train = np.array([[0,0],[0,1],[1,0],[1,1]])
y_train = np.array([[0, 1, 1, 0]]).T
```

B. Agregue métodos a la clase Perceptrón que permita visualizar la tasa de error por época de entrenamiento y otro que permita visualizar la recta de separación entre clases que se va ajustando durante el entrenamiento.