

Diplomatura Universitaria en Ciencia de Datos

Año 2024

Clase 4: Introducción a NumPy

Problema 1: Creación de arrays con Numpy.

- Crear dos arrays de NumPy y asignarlos en variables. El primero debe ser de 1 dimensión y el segundo de 2 dimensiones. Aplicar en cada uno los métodos **shape**, **ndim** y **size**.
- Crear un array de ceros de dimensión (3,3) y un array de unos de dimensión (2,4).
- Crear un array definido con valores en el intervalo abierto [10,30) con un step de 2.
- Crear dos arrays de 2D con valores aleatorios de la distribución normal. Utilizar **np.random.randn()** y **np.random.normal()**. ¿Cuáles son las diferencias?
- Crear un array con el comando **linspace**.

Problema 2: Operaciones con arrays en NumPy.

- Crear dos arrays y aplicar las operaciones **arr1 + arr2** y **arr1*arr2**. ¿Qué condiciones deben cumplir arr1 y arr2?
- Verificar la multiplicación de un escalar por un array.
- De un array de 2D obtener la matriz traspuesta.
- Realizar la multiplicación de dos matrices de dimensiones apropiadas.

Problema 3: Indexación y slicing de arrays de NumPy.

- Crear un array de dimensión (4,4) y extraer: el elemento de la fila 3 y columna 2; la segunda fila; y la tercera columna.
- Del array del ejercicio anterior, ¿Qué pasa si hacemos **arr[1] = np.array([10,10,10,10])**?
- ¿Y qué pasa si hacemos **arr[1] = 99**?

Problema 4: Manipulación de arrays en NumPy.

- Crear un array de 12 elementos en una fila. Utilizar el comando **reshape()**. ¿Qué argumentos de entrada puede tomar? Mostrar dos ejemplos.
- En el array del ejercicio anterior, aplicar el comando **split()**. Mostrar 3 ejemplos.
- Crear dos arrays de 1D y usar el operador **concatenate**. Luego usar el mismo operador en dos arrays de dimensión (2,2). Probar con el parámetro **axis=0**, y **axis=1**.

Problema 5: Operaciones estadísticas en NumPy.

- Crear un array de dimensión (3,3) y aplicar los métodos **np.mean()**, **np.std()**, **np.var()** y **np.sum()**.
- Repetir el ejercicio anterior utilizando el parámetro **axis=0** y **axis=1**.

Problema 6: Utilización de booleanos en NumPy.

- Crear un arreglo de 4 elementos y aplicar la máscara [False, True, True, False]. ¿Qué resulta?
- Crear un arreglo unidimensional de números enteros aleatorios y filtrar por los valores mayores al promedio, números impares y números pares menores al promedio.
- En un arreglo bidimensional A de valores aleatorios enteros, ¿qué diferencia hay entre hacer **A > A.mean()** y **A [A > A.mean()]**?

***Problema 6:** Se requiere armar un array a partir de un loop de 100 ciclos en los cuales se genera un array de 100x100 aleatoriamente (Randn), si la suma de los elementos de ese array es mayor a 0 que lo ponga como parte del array de salida. El array de salida va a ser de dimensiones (n,100,100) donde n es la cantidad de arrays cuya suma supere 0.

***Problema 7:** Realice una serie de funciones para matrices que:

- a) Realizar una función que dada una matriz de n por n, responda si la matriz es simétrica o no (Salida True/False).
- b) Realizar una función que verifique si una matriz de dimensión n es ortogonal, es decir si el producto de ella por su traspuesta es la identidad.
- c) Realizar una función que determine la norma de Frobenius (raíz cuadrada de la suma de los cuadrados de las componentes de una matriz dada).
- d) Realizar una función que determine la distancia de Frobenius entre dos matrices (suma de las diferencias de las componentes al cuadrado).