Contenido

[1. Numpy. 1](#_Toc142988439)

[1.1 Arreglos 1](#_Toc142988440)

[1.1.1 Accesos en arreglos bidimensionales. 1](#_Toc142988441)

[1.1.2 Operaciones con arreglos 1](#_Toc142988442)

[1.2 Forma o shape de un arreglo. 1](#_Toc142988443)

[2. Información de interes. 2](#_Toc142988444)

# Numpy.

Numpy es una librería

Suele llamarse np

## Arreglos

### Accesos en arreglos bidimensionales.

X[i] te dará la i-esima

Para obtener la fila i-esima hacemos:

Para obtener la columna j-esima hacemos:

### Operaciones con arreglos

Si tenemos un arreglo np.array([1,2,3,4,5]) podemos hacer la siguientes operaciones:

* x = np.array([1,2,3,4,5]);
* x += a 🡪 Implica sumarle “a” a cada elemento del vector.
* x -= b 🡪 Implica restarle “b” a cada elemento del vector.

## Forma o shape de un arreglo.

Cuando se habla de la "forma" o "shape" de un arreglo en términos de programación, especialmente en el contexto de bibliotecas como NumPy, nos referimos a las dimensiones de este arreglo. Por ejemplo, si tienes una matriz (o un arreglo bidimensional) que tiene 3 filas y 5 columnas, su forma es (3, 5).

En el contexto de un conjunto de datos, una forma (N, D) podría indicar N filas que representan una entidad y D podría representar las características de cada entidad, por ejemplo, podríamos tener una matriz (20, 5) tal que tenemos 20 personas con 5 características cada una (edad, peso, trabajo, estudios e ingresos mensuales).

Además, el shape de un np.array no se limita a dos dimensiones, si no que puede tener 1, 2 o más de dos dimensiones en general.

|  |  |  |
| --- | --- | --- |
|  | Resultado | Aclaraciones |
| np.shape(mi\_array\_o\_matriz) | (N\_filas, N\_columnas,…) |  |
| My\_array\_o\_matriz.shape | (n\_rows, n\_filas, …) | La propiedad shape del objeto np.array es una tupla de longitud “n” siendo “n” el numero de dimensiones del tensor. |

# Información de interes.

* Tanto a las tuplas, listas como los np array se accede a sus valores con corchetes rectangulares.