

# Hoja informativa: Distancia entre vectores

## Práctica

```
# Producto escalar de vectores

import numpy as np

dot_value1 = np.dot(vector1, vector2)
dot_value2 = vector1 @ vector2
```

```
# Distancia euclidiana entre vectores

import numpy as np
from scipy.spatial import distance

d = distance.euclidean(a, b)
```

```
# Distancia Manhattan entre vectores

import numpy as np
from scipy.spatial import distance

d = distance.cityblock(a, b)
```

```
# Índices de elementos mínimos y máximos en la matriz

index = np.array(distances).argmin() # índice de elemento mínimo
index = np.array(distances).argmax() # índice de elemento máximo
```

```
# Creación de clase

class ClassName:
```

```
def fit(self, arg1, arg2, ...): # método de clase
    # contenido del método
```

## Teoría

**Producto escalar** es una operación que da como resultado un número (**escalar**) que es igual a la suma de los productos elemento por elemento de los elementos de dos vectores.

**Distancia euclidiana** entre los vectores  $a=(x_1, x_2, \dots, x_n)$  y  $b=(y_1, y_2, \dots, y_n)$  es la suma de los cuadrados de las diferencias de coordenadas:

$$d_2(a, b) = \sqrt{(y_1 - x_1)^2 + \dots + (y_n - x_n)^2} = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

**Distancia Manhattan o distancia entre manzanas** es la suma de módulos de diferencias de coordenadas de vectores  $a=(x_1, x_2, \dots, x_n)$  y  $b=(y_1, y_2, \dots, y_n)$ :

$$d_1(a, b) = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| = \sum_{i=1}^n |x_i - y_i|$$

**Clase** (class) es un nuevo tipo de datos con sus propios métodos y atributos.