

# Relatório - 4 - Prática: Principais Bibliotecas e Ferramentas Python para Aprendizado de Máquina

## (I)

Paulo Victor Sousa de Almeida

### Parte I - NUMPY

#### 01- Numpy

O NumPy é uma biblioteca em python utilizada para computação numérica e científica, oferecendo suporte para matriz multidimensional(arrays) e funções matemáticas de alto nível.

Importando biblioteca NumPy.

por convenção utilizaremos a referência “np” na importação da biblioteca NumPy

```
In [2]: 1 import numpy as np
```

As principais funções do numpy utilizadas no curso foram:

np.array() : cria um array a partir de uma sequência(linh, tuplas)

```
In [6]: 1 np.array(minha_lista)|  
Out[6]: array([1, 2, 3])
```

np.arange() : cria um array com sequência de valores

```
In [9]: 1 np.arange(0, 10)  
Out[9]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

np.zeros() : cria um array preenchidos por zeros

```
In [11]: 1 np.zeros(3)
Out[11]: array([0., 0., 0.])
```

`np.ones()` : cria um array preenchidos por uns

```
In [16]: 1 np.ones((3,3))
Out[16]: array([[1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.]])
```

`np.eye()` : cria um array de matriz identidade

```
In [17]: 1 np.eye(4)
Out[17]: array([[1., 0., 0., 0.],
               [0., 1., 0., 0.],
               [0., 0., 1., 0.],
               [0., 0., 0., 1.]])
```

`np.linspace()` : cria um array com valores igualmente espaçados em um intervalo

```
In [26]: 1 np.linspace(0,10, 6)
Out[26]: array([ 0.,  2.,  4.,  6.,  8., 10.] )
```

`np.random.rand()` : cria um array de números aleatórios entre 0 e 1

```
In [27]: 1 np.random.rand(5)
Out[27]: array([0.52086978, 0.99153649, 0.56543457, 0.64565458, 0.05890272])
```

`np.random.randn()` : cria um array de números aleatórios, distribuição gaussiana

```
In [31]: 1 np.random.randn(4)
Out[31]: array([ 0.94450078, -0.31807085, -0.52841966,  0.75211784])
```



## Operações Aritméticas Numpy

### Soma de Arrays

```
In [6]: arr + arr
Out[6]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30])
```

### Subtração de Arrays

```
In [7]: arr - arr
Out[7]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

### Multiplicação de Arrays

```
In [8]: arr * arr
Out[8]: array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225])
```

### Divisão de Arrays

exibe uma mensagem de aviso quando a divisão for incorreta, retorna o valor “nan” quando a divisão for impossível (0/0).

```
In [9]: arr / arr
C:\Users\victo\AppData\Local\Temp\ipykernel_1272\3001117470.py:1: RuntimeWarning: invalid value encountered in divide
arr / arr
Out[9]: array([nan,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])
```

exibe uma mensagem de aviso quando a divisão for incorreta, retorna o valor “inf” quando a divisão for infinita (1/0).

```
In [9]: 1 / arr
C:\Users\victo\AppData\Local\Temp\ipykernel_6644\1528351372.py:1: RuntimeWarning: divide by zero encountered in divide
1 / arr
Out[9]: array([      inf,  1.,  0.5,  0.33333333,  0.25,  0.2,  0.16666667,  0.14285714,  0.125,  0.11111111,  0.1,  0.09090909,  0.08333333,  0.07692308,  0.07142857,  0.06666667])
```

## Funções aritméticas Nump

### sqrt() raiz quadrada

```
In [13]: np.sqrt(arr)

Out[13]: array([0.          , 1.          , 1.41421356, 1.73205081, 2.          ,
                2.23606798, 2.44948974, 2.64575131, 2.82842712, 3.          ,
                3.16227766, 3.31662479, 3.46410162, 3.60555128, 3.74165739,
                3.87298335])
```

### exp() exponencial

```
In [14]: np.exp(arr)

Out[14]: array([1.00000000e+00, 2.71828183e+00, 7.38905610e+00, 2.00855369e+01,
                5.45981500e+01, 1.48413159e+02, 4.03428793e+02, 1.09663316e+03,
                2.98095799e+03, 8.10308393e+03, 2.20264658e+04, 5.98741417e+04,
                1.62754791e+05, 4.42413392e+05, 1.20260428e+06, 3.26901737e+06])
```

### mean() média

```
In [15]: np.mean(arr)

Out[15]: 7.5
```

### std() desvio padrão

```
In [16]: np.std(arr)

Out[16]: 4.6097722286464435
```

### sin() seno

```
In [17]: np.sin(arr)

Out[17]: array([ 0.          ,  0.84147098,  0.90929743,  0.14112001, -0.7568025 ,
                -0.95892427, -0.2794155 ,  0.6569866 ,  0.98935825,  0.41211849,
                -0.54402111, -0.99999021, -0.53657292,  0.42016704,  0.99060736,
                0.65028784])
```

max() valor máximo

```
In [19]: np.max(arr)
```

```
Out[19]: 15
```

min() valor mínimo

```
In [18]: np.min(arr)
```

```
Out[18]: 0
```