

MPSI_Lab03

December 1, 2023

```
[3]: import numpy as np
```

1 Zadania do samodzielnego wykonania

1.1 Ćwiczenie 1

Poniżej podano dwie tablice A i B:

```
[4]: A = np.array([[3, 4, 5],  
                  [8, 3, 1]])  
B = np.array([[0, 5, 2],  
              [4, 2, 1]])
```

Połącz te tablice w następujący sposób:

```
[ ]: array([[3, 4, 5],  
           [8, 3, 1],  
           [0, 5, 2],  
           [4, 2, 1]])
```

```
[[3 4 5]  
 [8 3 1]  
 [0 5 2]  
 [4 2 1]]
```

Wydrukuj wynik do konsoli. Wskazówka: Użyj funkcji `np.concatenate()`.

1.2 Ćwiczenie 2

Podana jest poniższa tablica zawierająca dane data oraz tablica zawierająca klasę docelową target dla danego wiersza:

```
[7]: data = np.array([[4.3, 4.2],  
                      [3.1, 3.6]])  
target = np.array([[0],  
                   [1]])
```

Połącz te tablice w jedną tak jak pokazano poniżej:

```
array([[4.3, 4.2, 0. ], [3.1, 3.6, 1. ]])
```

1.3 Ćwiczenie 3

Podane są następujące trzy jednowymiarowe tablice:

```
[9]: feature1 = np.array([1.6, 0.9, 2.2])
     feature2 = np.array([0.4, 1.3, 3.2])
     feature3 = np.array([1.4, 0.3, 1.2])
```

Wydrukuj wynik do konsoli.

Oczekiwany wynik: `[[1.6 0.4 1.4] [0.9 1.3 0.3] [2.2 3.2 1.2]]`

Przekształć każdą z tych tablic w kolumnę i połącz je w jedną dużą tablicę. Wydrukuj wynik do konsoli. Wskazówka: Użyj funkcji `np.column_stack()`.

1.4 Ćwiczenie 4

Podana jest następująca tablica numpy A:

```
[11]: np.random.seed(42)

A = np.random.randint(low=0, high=7, size=(5, 8))
A[:, :2] = 0
A[:, -2:] = 1
```

przekształć w tablicę poniżej:

```
[ ]: array([[0., 0., 0., 0., 0., 0.],
          [0., 1., 1., 1., 1., 0.],
          [0., 1., 1., 1., 1., 0.],
          [0., 1., 1., 1., 1., 0.],
          [0., 1., 1., 1., 1., 0.],
          [0., 0., 0., 0., 0., 0.]])
```

Podziel tablicę na trzy części tak, aby dwie pierwsze kolumny utworzyły tablicę A1, cztery kolejne tablicę A2 i ostatnie dwie tablicę A3.

Wydrukuj wynik do konsoli. Wskazówka: Użyj funkcji `np.split()`.

1.5 Ćwiczenie 5

Podana jest następująca tablica numpy A:

```
[12]: np.random.seed(42)

A = np.random.randint(low=0, high=2, size=(10, 6))
```

Oblicz łączną liczbę niezerowych elementów dla tej tablicy. Wydrukuj wynik do konsoli.

1.6 Ćwiczenie 6

Podana jest tablica numpy A:

```
[13]: np.random.seed(42)
      A = np.random.randn(10, 4)
```

Ustaw opcję biblioteki numpy tak, aby wydrukować tablicę z określoną precyzją. Ustaw jego wartość na 4 i wydrukuj tablicę A do konsoli.

1.7 Ćwiczenie 7

Podana jest następująca tablica numpy A:

```
[14]: A = np.array([1.2e-6, 1.7e-7])
```

Ustaw wartość precyzji na 8 oraz odpowiednią opcję biblioteki numpy tak, aby pominąć notację matematyczną. Wydrukuj tablicę A do konsoli. Wskazówka: Użyj funkcji `np.intersect1d()`.

1.8 Ćwiczenie 8

Podana jest następująca tablica numpy A:

```
[15]: np.random.seed(42)

      A = np.random.randn(8, 4)
```

Usuń trzecią kolumnę z tablicy A. Resztę tablicy wydrukuj do konsoli.

1.9 Ćwiczenie 9

Dana jest jednowymiarowa tablica v (wektor):

```
[16]: v = np.array([3, 4, -2])
```

Oblicz normę wektora v. Wydrukuj wynik do konsoli.

1.10 Ćwiczenie 10

Podana jest poniższa tablica numpy A:

```
[17]: A = np.random.randint(10, size=(100, 30))
```

Ustaw opcję biblioteki numpy tak, aby wydrukować 10 elementów brzegowych i następnie wydrukuj tablicę A