МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчёт о лабораторной работе №2 по дисциплине технологии распознавания образов

Выполнила:
Нестеренко Тамара Антоновна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ

1. Практическая часть

```
>>> import numpy as np
>>> m = np.matrix('1 2 3 4; 5 6 7 8; 9 1 5 7')
>>> print(m)
[[1 2 3 4]
 [5 6 7 8]
 [9 1 5 7]]
```

Рисунок 1 – Пример создания матрицы

```
>>> m[1, 0]
5
```

Рисунок 2 – Пример элементы матрицы с заданными координатами

```
>>> m[1, :]
matrix([[5, 6, 7, 8]])
```

Рисунок 3 – Пример вывода строки матрицы

```
>>> m[:, 2]
matrix([[3],
        [7],
        [5]])
```

Рисунок 4 – Пример вывода столбца матрицы

```
>>> m[1, 2:]
matrix([[7, 8]])
```

Рисунок 5 – Пример вывода части строки матрицы

```
>>> m[0:2, 1]
matrix([[2],
        [6]])
```

Рисунок 6 – Пример вывода части столбца матрицы

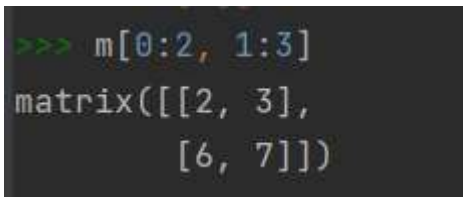


Рисунок 7 – Пример вывода непрерывной части матрциы

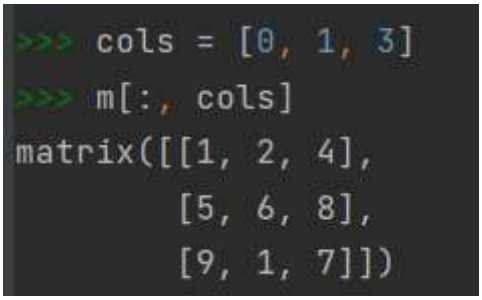


Рисунок 8 – Пример вывода произвольных столбцов / строк матрицы

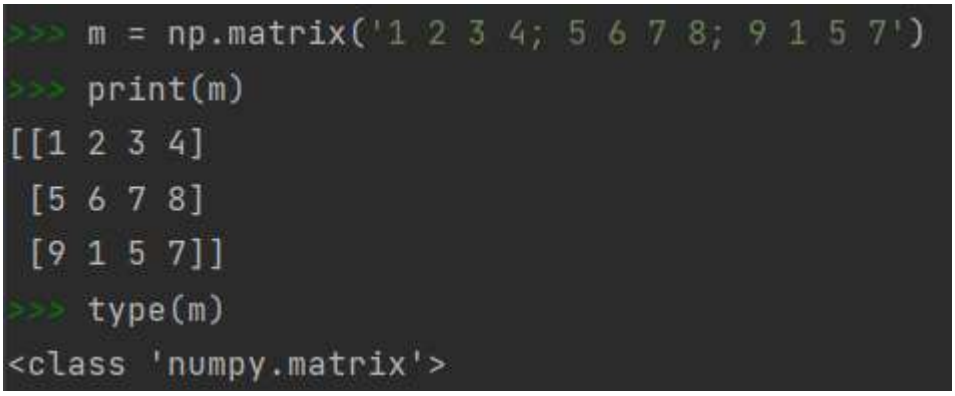


Рисунок 9 – Пример определения типа матрицы

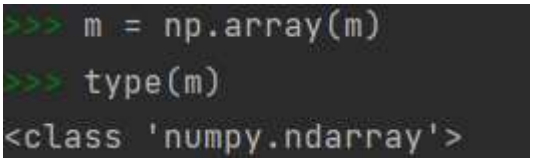


Рисунок 10 – Пример изменения типа matrix на ndarray

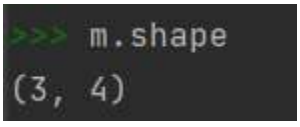


Рисунок 11 – Пример вывода размерности массива

```
>>> m.max()
9
>>> np.max(m)
9
>>> m.max()
9
```

Рисунок 12 – Пример вызова статистики

```
>>> m.max(axis=1)
array([4, 8, 9])
```

Рисунок 13 – Пример поиска максимального элемента в каждой строке

```
>>> m.max(axis=0)
array([9, 6, 7, 8])
```

Рисунок 14 – Пример поиска максимального элемента в каждом столбце

```
>>> m.mean()
4.833333333333333
>>> m.mean(axis=1)
array([2.5, 6.5, 5.5])
>>> m.sum()
58
>>> m.sum(axis=0)
array([15,  9, 15, 19])
```

Рисунок 15 – Пример некоторых методов для расчёта статистики

```
>>> nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
>>> letters = np.array(['a', 'b', 'c', 'd', 'a', 'e', 'b'])
```

Рисунок 16 – Пример использования boolean массива для доступа к ndarray

```
>>> a = True
>>> b = 5 > 7
>>> print(b)
False
```

Рисунок 17 – Пример работы с типом данных boolean

```
>>> less_then_5 = nums < 5
>>> less_then_5
array([ True,  True,  True,  True, False, False, False, False, False,
       False])
```

Рисунок 18 – Пример построения на базе ndarray

```
>>> pos_a = letters == 'a'
>>> pos_a
array([ True, False, False, False,  True, False, False])
```

Рисунок 19 – Пример массивов массивы с элементами типа boolean. Пример создания создали boolean массива, в котором на месте элементов из nums, которые меньше пяти стоит True, в остальных случаях – False.

```
>>> print(m)
[[1 2 3 4]
 [5 6 7 8]
 [9 1 5 7]]
>>> mod_m = np.logical_and(m>=3, m<=7)
>>> mod_m
matrix([[False, False,  True,  True],
        [ True,  True,  True, False],
        [False, False,  True,  True]])
>>> m[mod_m]
matrix([[3, 4, 5, 6, 7, 5, 7]])
```

Рисунок 20 – Пример использования функции logical_and

```
>>> nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
>>> nums[nums < 5]
array([1, 2, 3, 4])
```

Рисунок 21 – Пример иллюстрации массива

```
>>> nums[nums < 5] = 10
>>> print(nums)
[10 10 10 10  5  6  7  8  9 10]
```

Рисунок 22 – Пример заполнения массива число 10

```
>>> m[m > 7] = 25
>>> print(m)
[[ 1  2  3  4]
 [ 5  6  7 25]
 [25  1  5  7]]
```

Рисунок 23 – Пример заполнения массива числом 25

```
>>> np.arange(10)
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> np.arange(5, 12)
array([ 5,  6,  7,  8,  9, 10, 11])
>>> np.arange(1, 5, 0.5)
array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5])
```

Рисунок 24 – Пример использования функции np.arange()

```
>>> a = [[1, 2], [3, 4]]
>>> np.matrix(a)
matrix([[1, 2],
        [3, 4]])
```

Рисунок 25 – Пример использования np.matrix со списков в Python

```
>>> b = np.array([[5, 6], [7, 8]])
>>> np.matrix(b)
matrix([[5, 6],
        [7, 8]])
```

Рисунок 26 – Пример использования np.array с массивом Numpy

```
>>> np.matrix('[1, 2; 3, 4]')
matrix([[1, 2],
        [3, 4]])
```

Рисунок 27 – Пример использования np.matrix в Matlab

```
>>> np.zeros((3, 4))
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

Рисунок 28 – Пример использования np.zeros()

```
>>> np.eye(3)
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

Рисунок 29 – Пример использования np,eye()

```
>>> A = np.array([[1, 2,3], [4, 5, 6], [7, 8, 9]])
>>> A
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

Рисунок 30 – Пример создания двумерной матрицы размера 3x3

```
>>> np.ravel(A)
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Рисунок 31 – Пример работы функции np.ravel()

```
>>> np.ravel(A, order='C')
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Рисунок 32 – Пример работы функции np.ravel()

```
>>> np.ravel(A, order='F')
array([1, 4, 7, 2, 5, 8, 3, 6, 9])
```

Рисунок 33 – Пример работы функции np.ravel()

```
>>> a = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> np.where(a % 2 == 0, a * 10, a / 10)
array([ 0. ,  0.1, 20. ,  0.3, 40. ,  0.5, 60. ,  0.7, 80. ,  0.9])
```

Рисунок 34 – Пример работы функции np.whare()

```
>>> a = np.random.rand(10)
>>> a
array([0.02618136, 0.1343769 , 0.56092075, 0.19345277, 0.718868  ,
       0.85485578, 0.04227395, 0.27862628, 0.05742005, 0.56470213])
>>> np.where(a > 0.5, True, False)
array([False, False,  True, False,  True,  True, False, False, False,
        True])
>>> np.where(a > 0.5, 1, -1)
array([-1, -1,  1, -1,  1,  1, -1, -1, -1,  1])
```

Рисунок 34 – Пример работы функции np.where()

```
>>> x = np.linspace(0, 1, 5)
>>> x
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
>>> y = np.linspace(0, 2, 5)
>>> y
array([0. , 0.5, 1. , 1.5, 2. ])
```

Рисунок 35 – Пример работы функции np.linspace()

```
>>> xg, yg = np.meshgrid(x, y)
>>> xg
array([[0.  , 0.25, 0.5 , 0.75, 1.  ],
       [0.  , 0.25, 0.5 , 0.75, 1.  ],
       [0.  , 0.25, 0.5 , 0.75, 1.  ],
       [0.  , 0.25, 0.5 , 0.75, 1.  ],
       [0.  , 0.25, 0.5 , 0.75, 1.  ]])
>>> yg
array([[0. , 0. , 0. , 0. , 0. ],
       [0.5, 0.5, 0.5, 0.5, 0.5],
       [1. , 1. , 1. , 1. , 1. ],
       [1.5, 1.5, 1.5, 1.5, 1.5],
       [2. , 2. , 2. , 2. , 2. ]])
```

Рисунок 36 – Пример работы функции np.meshgrid()

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
x = np.linspace(0, 1, 5)
y = np.linspace(0, 2, 5)
xg, yg = np.meshgrid(x, y)
plt.plot(xg, yg, color="r", marker="*", linestyle="none")
```

```
[<matplotlib.lines.Line2D at 0x196d1a15760>,
 <matplotlib.lines.Line2D at 0x196d1a15820>,
 <matplotlib.lines.Line2D at 0x196d1a15940>,
 <matplotlib.lines.Line2D at 0x196d1a15a60>,
 <matplotlib.lines.Line2D at 0x196d1a15b80>]
```

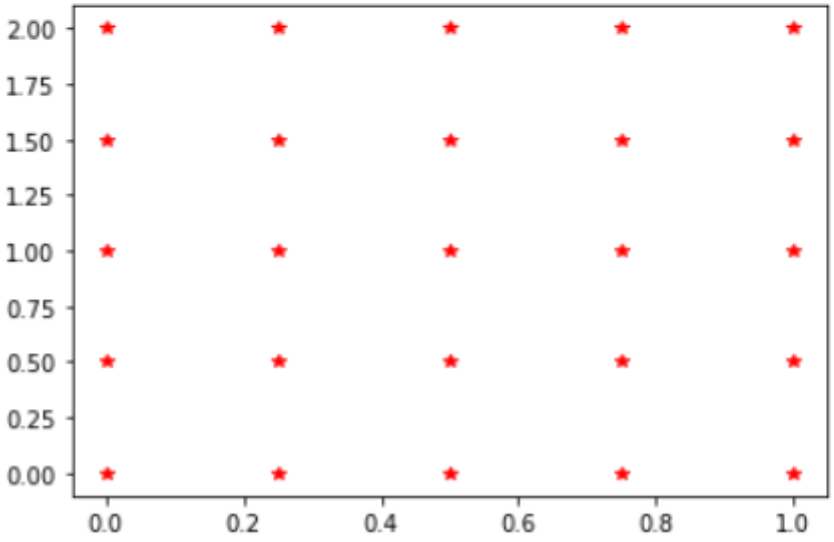


Рисунок 37 – Пример использования matplotlib

```
>>> np.random.permutation(7)
array([4, 5, 0, 2, 3, 1, 6])
>>> a = ['a', 'b', 'c', 'd', 'e']
>>> np.random.permutation(a)
array(['a', 'd', 'e', 'b', 'c'], dtype='<U1')
```

Рисунок 38 – Пример использования np.random.permutation()

```
>>> arr = np.linspace(0, 10, 5)
>>> arr
array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

Рисунок 39 – Пример использования np.linspace()

```
>>> arr_mix = np.random.permutation(arr)
>>> arr_mix
array([ 5. ,  7.5, 10. ,  2.5,  0. ])
```

Рисунок 40 – Пример использования np.random.permutation()

```
>>> index_mix = np.random.permutation(len(arr_mix))
>>> index_mix
array([0, 1, 3, 4, 2])
>>> arr[index_mix]
array([ 0. ,  2.5,  7.5, 10. ,  5. ])
```

Рисунок 41 – Пример использования np.random.permutation()

2. Вопросы для защиты

1. Каково назначение библиотеки NumPy?

Numpy – это библиотека для языка программирования Python, которая предоставляет в распоряжение разработчика инструменты для эффективной работы с многомерными массивами и высокопроизводительные вычислительные алгоритмы.

2. Что такое массивы ndarray?

Ndarray - это (обычно фиксированный размер) многомерный контейнер элементов одного типа и размера. Количество измерений и элементов в массиве определяется его формой, которая является кортежем из N натуральных чисел, которые определяют размеры каждого измерения.

3. Как осуществляется доступ к частям многомерного массива?

Элементы матрицы с заданными координатами: m[1,0]

Строка матрицы: m[1, :]

Столбец матрицы: m[:, 1]

Часть строки матрицы: m[1, 2:]

Часть столбца матрицы: m[0:2, 1]

Непрерывная часть матрицы: m[0:2, 1:3]

Произвольные столбцы / строки матрицы: cols = [0, 1, 2]; m[:, cols]

4. Как осуществляется расчет статистик по данным?

Размерность массива: m.shape

Вызов функции расчёта статистики: m.max()

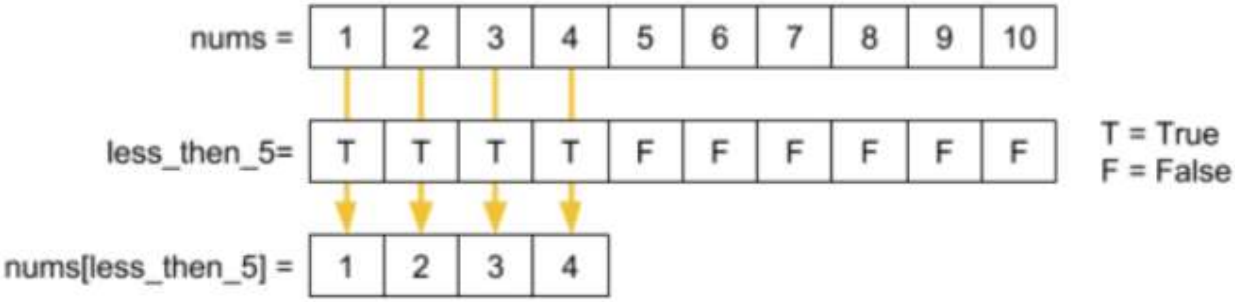
Расчёт статистики по строкам или столбцам массива: m.max(axis=1); m.max(axis=0)

Индексы элементов с максимальным значением (по осям): argmax
Индексы элементов с минимальными значением (по осям): argmin
Максимальные значения элементов (по осям): max
Минимальные значения элементов (по осям): min
Средние значения элементов (по осям): mean
Произведение всех элементов (по осям): prod
Стандартное отклонение (по осям): std
Сумма всех элементов (по осям): sum
Дисперсия (по осям): var

5. Как выполняется выборка данных из массивов ndarray?

```
>>> less_then_5 = nums < 5
>>> less_then_5
array([ True,  True,  True,  True, False, False, False, False, False,
       False])
```

Если мы переменную *less_then_5* передадим в качестве списка индексов для *nums*, то получим массив, в котором будут содержаться элементы из *nums* с индексами равными индексам *True* позиций массива *less_then_5*, графически это будет выглядеть так.




```
>>> nums[less_then_5]
array([1, 2, 3, 4])
```

Ссылка на репозиторий: https://github.com/tamaranesterenko/TRO_LR_2