

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

«Основы работы с библиотекой NumPy»

Отчет по лабораторной работе № 3.2

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Халимендик Я. Д. « » 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

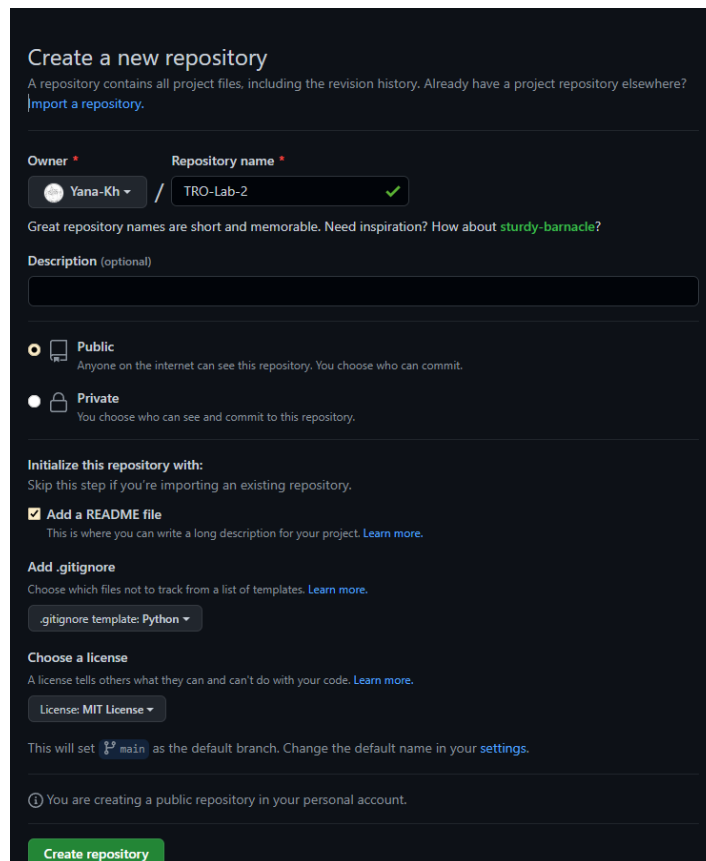
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: исследовать базовые возможности библиотеки NumPy языка программирования Python.

Ход работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

Yana-Kh / TRO-Lab-2 ✓

Great repository names are short and memorable. Need inspiration? How about [sturdy-barnacle](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git>git clone https://github.com/Yana-Kh/TRO-Lab-2.git
Cloning into 'TRO-Lab-2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\ynakh\OneDrive\Рабочий стол\Git>
```

Рисунок 2 – Клонирование репозитория

4. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\TRO-Lab-2>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/ynakh/OneDrive/Рабочий стол\Git\TRO-Lab-2/.git/hooks]

C:\Users\ynakh\OneDrive\Рабочий стол\Git\TRO-Lab-2>_
```

Рисунок 3 – Организация репозитория в соответствии с моделью git-flow

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\TRO-Lab-2>git add .
C:\Users\ynakh\OneDrive\Рабочий стол\Git\TRO-Lab-2>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   .gitignore

C:\Users\ynakh\OneDrive\Рабочий стол\Git\TRO-Lab-2>git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/ynakh/OneDrive/Рабочий стол\Git\TRO-Lab-2/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/ynakh/OneDrive/Рабочий стол\Git\TRO-Lab-2/.git/COMMIT_EDITMSG to Unix format...
[main 479ac06] ignore
1 file changed, 34 insertions(+)
```

Рисунок 4 – Дополнение файла .gitignore

6. Проработать примеры лабораторной работы.

```
In [7]: 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 5 – Пример создания матрицы

Элемент матрицы с заданными координатами

```
In [8]: m[1, 0]
```

```
Out[8]: 5
```

Рисунок 6 – Пример элементы матрицы с заданными координатами

Строка матрицы

```
In [9]: m[1, :]
```

```
Out[9]: matrix([[5, 6, 7, 8]])
```

Рисунок 7 – Пример вывода строки матрицы

```
In [10]: m[:, 2]
```

```
Out[10]: matrix([[3],  
                 [7],  
                 [5]])
```

Рисунок 8 – Пример вывода столбца матрицы

Часть строки матрицы

```
In [11]: m[1, 2:]
```

```
Out[11]: matrix([[7, 8]])
```

Рисунок 9 – Пример вывода части строки матрицы

Часть столбца матрицы

```
In [12]: m[0:2, 1]
```

```
Out[12]: matrix([[2],  
                 [6]])
```

Рисунок 10 – Пример вывода части столбца матрицы

```
In [13]: m[0:2, 1:3]
Out[13]: matrix([[2, 3],
                [6, 7]])
```

Рисунок 11 – Пример вывода непрерывной части матрицы

Произвольные столбцы / строки матрицы

```
In [14]: cols = [0, 1, 3]
        m[:, cols]
Out[14]: matrix([[1, 2, 4],
                [5, 6, 8],
                [9, 1, 7]])
```

Рисунок 12 – Пример вывода произвольных столбцов/строк матрицы

Расчет статистик по данным в массиве

```
In [17]: 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]]

In [18]: type(m)
Out[18]: numpy.matrix

In [19]: m = np.array(m)
        type(m)
Out[19]: numpy.ndarray
```

Рисунок 13 – Пример расчета статистик по данным в массиве

Размерность массива

```
In [20]: m.shape
Out[20]: (3, 4)
```

Рисунок 14 – Пример вывода размерности массива

Вызов функции расчета статистики

```
In [21]: m.max()
Out[21]: 9

In [22]: np.max(m)
Out[22]: 9
```

Рисунок 15 – Пример вызова статистики

Расчет статистик по строкам или столбцам массива

```
In [24]: m.max(axis=1)
Out[24]: array([4, 8, 9])

In [25]: m.max(axis=0)
Out[25]: array([9, 6, 7, 8])
```

Рисунок 16 – Пример поиска максимального элемента в каждой строке/столбце

Использование boolean массива для доступа к ndarray

```
In [26]: nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
         letters = np.array(['a', 'b', 'c', 'd', 'a', 'e', 'b'])

In [27]: less_than_5 = nums < 5
         less_than_5
Out[27]: array([ True,  True,  True,  True, False, False, False, False, False,
               False])

In [28]: pos_a = letters == 'a'
         pos_a
Out[28]: array([ True, False, False, False,  True, False, False])

In [30]: mod_m = np.logical_and(m>=3, m<=7)
         mod_m
Out[30]: array([[False, False,  True,  True],
               [ True,  True,  True, False],
               [False, False,  True,  True]])

In [31]: m[mod_m]
Out[31]: array([3, 4, 5, 6, 7, 5, 7])

In [32]: nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
         nums[nums < 5]
Out[32]: array([1, 2, 3, 4])

In [33]: nums[nums < 5] = 10
         print(nums)
[10 10 10 10  5  6  7  8  9 10]

In [34]: m[m > 7] = 25
         print(m)
[[ 1  2  3  4]
 [ 5  6  7 25]
 [25  1  5  7]]
```

Использование boolean массива для доступа к ndarray

```
In [26]: nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
         letters = np.array(['a', 'b', 'c', 'd', 'a', 'e', 'b'])

In [27]: less_than_5 = nums < 5
         less_than_5
Out[27]: array([ True,  True,  True,  True, False, False, False, False, False,
               False])
```

Рисунок 17 – Пример использования boolean массива для доступа к ndarray

Дополнительные функции

```
In [35]: np.arange(10)
Out[35]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [36]: np.arange(5, 12)
Out[36]: array([ 5, 6, 7, 8, 9, 10, 11])

In [37]: np.arange(1, 5, 0.5)
Out[37]: array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5])
```

Рисунок 18 – Пример использования функции np.arange()

```
In [39]: b = np.array([[5, 6], [7, 8]])
         np.matrix(b)
Out[39]: matrix([[5, 6],
                [7, 8]])
```

Рисунок 19 – Пример использования np.matrix

```
In [40]: np.zeros((3, 4))
Out[40]: array([[0., 0., 0., 0.],
               [0., 0., 0., 0.],
               [0., 0., 0., 0.]])
```

Рисунок 20 – Пример использования np.zeros()

```
In [41]: np.eye(3)
Out[41]: array([[1., 0., 0.],
               [0., 1., 0.],
               [0., 0., 1.]])
```

Рисунок 21 – Пример использования np.eye()

```
In [42]: A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
         A
Out[42]: array([[1, 2, 3],
               [4, 5, 6],
               [7, 8, 9]])

In [43]: np.ravel(A)
Out[43]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Рисунок 22 – Пример работы функции np.ravel()

```

In [46]: a = np.random.rand(10)
a
Out[46]: array([0.87636044, 0.32859234, 0.90092636, 0.31485   , 0.3353488 ,
               0.09833499, 0.5377768 , 0.21516775, 0.20184881, 0.84665619])

In [47]: np.where(a > 0.5, True, False)
Out[47]: array([ True, False,  True, False, False, False,  True, False, False,
               True])

In [48]: np.where(a > 0.5, 1, -1)
Out[48]: array([ 1, -1,  1, -1, -1, -1,  1, -1, -1,  1])

```

Рисунок 23 – Пример работы функции np.where()

```

In [50]: x = np.linspace(0, 1, 5)
x
Out[50]: array([0. , 0.25, 0.5 , 0.75, 1.  ])

In [52]: y = np.linspace(0, 2, 5)
y
Out[52]: array([0. , 0.5, 1. , 1.5, 2.  ])

```

Рисунок 24 – Пример работы функции np.linspace()

```

In [53]: xg, yg = np.meshgrid(x, y)
xg
Out[53]: 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.  ]])

In [54]: yg
Out[54]: 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. ]])

In [55]: import matplotlib.pyplot as plt
%matplotlib inline

In [56]: plt.plot(xg, yg, color="r", marker="*", linestyle="none")
Out[56]: [<matplotlib.lines.Line2D at 0x27409fc6790>,
          <matplotlib.lines.Line2D at 0x27409fc67f0>,
          <matplotlib.lines.Line2D at 0x27409fc68e0>,
          <matplotlib.lines.Line2D at 0x27409fc6a00>,
          <matplotlib.lines.Line2D at 0x27409fc6b20>]

```

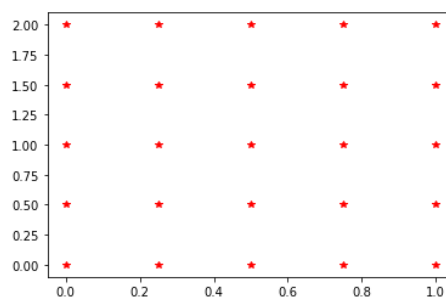


Рисунок 25 – Пример работы функции np.meshgrid()


```
In [57]: np.random.permutation(7)
Out[57]: array([4, 5, 3, 0, 2, 6, 1])

In [58]: a = ['a', 'b', 'c', 'd', 'e']

In [59]: np.random.permutation(a)
Out[59]: array(['b', 'c', 'a', 'd', 'e'], dtype='<U1')
```

Рисунок 26 – Пример использования np.random.permutation()

```
In [60]: arr = np.linspace(0, 10, 5)
arr
Out[60]: array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

Рисунок 27 – Пример использования np.linspace()

```
In [61]: arr_mix = np.random.permutation(arr)
arr_mix
Out[61]: array([ 0. ,  7.5,  5. , 10. ,  2.5])

In [62]: index_mix = np.random.permutation(len(arr_mix))
index_mix
Out[62]: array([0, 4, 2, 3, 1])

In [63]: arr[index_mix]
Out[63]: array([ 0. , 10. ,  5. ,  7.5,  2.5])
```

Рисунок 28 – Пример использования np.random.permutation()

7. Решить задания в ноутбуках, выполненных преподавателем.

Создайте массив с 5 любыми числами:

```
In [3]: my = np.array([5, 5, 2, 0, 3])
my
Out[3]: array([5, 5, 2, 0, 3])
```

Рисунок 29 – Решение задачи

Создайте массив из 5 чисел. Возведите каждый элемент массива в степень 3

```
In [7]: #Возьму ранее созданный массив my
print("my:", my)
print('\nmy ** 3:', my ** 3)

my: [5 5 2 0 3]

my ** 3: [125 125 8 0 27]
```

Рисунок 30 – Решение задачи

Создайте два массива одинаковой длины. Выведите массив, полученный делением одного массива на другой.

```
In [11]: my2 = np.array([1, 2, 2, 4, 5])

print("my:", my)
print("my2:", my2)
print("a / b =", my / my2)

my: [5 5 2 0 3]
my2: [1 2 2 4 5]
a / b = [5.  2.5 1.  0.  0.6]
```

Рисунок 31 – Решение задачи

Создайте 2 массива из 5 элементов. Проверьте условие "Элементы первого массива меньше 6, элементы второго массива делятся на 3"

```
In [14]: print("my:", my)
print("my < 6: ", my < 6)
print("\nmy2:", my2)
print("my2 % 3 == 0: ", my2 % 3 == 0)

my: [5 5 2 0 3]
my < 6: [ True  True  True  True  True]

my2: [1 2 2 4 5]
my2 % 3 == 0: [False False False False False]
```

Теперь проверьте условие "Элементы первого массива делятся на 2 или элементы второго массива больше 2"

```
In [15]: print("my:", my)
print("my2:", my2)
print("my % 2 == 0 or my2 > 2: ", (my % 2 == 0) | (my2 > 2))

my: [5 5 2 0 3]
my2: [1 2 2 4 5]
my % 2 == 0 or my2 > 2: [False False  True  True  True]
```

Рисунок 32 – Решение задачи

Создайте массив с элементами от 1 до 20. Выведите все элементы, которые больше 5 и не делятся на 2

Подсказка: создать массив можно с помощью функции `np.arange()`, действие которой аналогично функции `range`, которую вы уже знаете.

```
In [17]: c = np.arange(1, 20)
c

Out[17]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
 18, 19])

In [20]: print("c[(c > 5) & (c % 2 != 0)]", c[(c > 5) & (c % 2 != 0)])

c[(c > 5) & (c % 2 != 0)] [ 7  9 11 13 15 17 19]
```

Рисунок 33 – Решение задачи

Выполните операции, перечисленные ниже:

```
In [25]: print("a =", a)
print("b =", b)
print("Разность между a и b:", a - b)
print("Квадраты элементов b:", b ** 2)
print("Половины произведений элементов массивов a и b:", a*b / 2)
print()
print("Максимальный элемент b:", np.max(b))
print("Сумма элементов массива b:", np.sum(b))
print("Индекс максимального элемента b:", np.argmax(b))

a = [1 2 3 4 5]
b = [0.1 0.2 0.3 0.4 0.5]
Разность между a и b: [0.9 1.8 2.7 3.6 4.5]
Квадраты элементов b: [0.01 0.04 0.09 0.16 0.25]
Половины произведений элементов массивов a и b: [0.05 0.2 0.45 0.8 1.25]

Максимальный элемент b: 0.5
Сумма элементов массива b: 1.5
Индекс максимального элемента b: 4
```

Рисунок 34 – Решение задачи

Задайте два массива: [5, 2, 3, 12, 4, 5] и ['f', 'o', 'o', 'b', 'a', 'r']

Выведите буквы из второго массива, индексы которых соответствуют индексам чисел из первого массива, которые больше 1, меньше 5 и делятся на 2

```
In [27]: ex1 = np.array([5, 2, 3, 12, 4, 5])
ex2 = np.array(['f', 'o', 'o', 'b', 'a', 'r'])

print("Буквы: ", ex2[(ex1 > 1) & (ex1 < 5) & (ex1 % 2 == 0)])

Буквы:  ['o' 'a']
```

Рисунок 35 – Решение задачи

Задание №1: Создайте два массива: в первом должны быть четные числа от 2 до 12 включительно, а в другом числа 7, 11, 15, 18, 23, 29.

1. Сложите массивы и возведите элементы получившегося массива в квадрат.

2. Выведите все элементы из первого массива, индексы которых соответствуют индексам тех элементов второго массива, которые больше 12 и дают остаток 3 при делении на 5.

3. Проверьте условие "Элементы первого массива делятся на 4, элементы второго массива меньше 14". (Подсказка: в результате должен получиться массив с True и False)

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

```
In [12]: a = np.arange(2, 13, 2)
b = np.array([7, 11, 15, 18, 23, 29])

print("a = ", a)
print("b = ", b)
print((a + b) ** 2)

a = [ 2  4  6  8 10 12]
b = [ 7 11 15 18 23 29]
[ 81 225 441 676 1089 1681]
```

Рисунок 36 – Решение первой задачи

```
In [13]: print("a = ", a)
print("b = ", b)

print(a[(b > 12) & (b % 5 == 3)])

a = [ 2  4  6  8 10 12]
b = [ 7 11 15 18 23 29]
[ 8 10]
```

Рисунок 37 – Решение второй задачи

```
In [16]: print("Первое условие: ", a % 4 == 0)
print("Первое условие: ", b < 14)

Первое условие: [False  True False  True False  True]
Первое условие: [ True  True False False False False]
```

Рисунок 38 – Решение третьей задачи

Задание №2:

- Найдите интересный для вас датасет.
- Рассчитайте подходящие описательные статистики для признаков объектов в выбранном датасете
- Проанализируйте и прокомментируйте содержательно получившиеся результаты
- Все комментарии оформляйте строго в ячейках формата markdown

```

In [18]: import csv
import numpy as np

In [34]: with open('data.csv', 'r', newline='') as csvf:
data = csv.reader(csvf, delimiter=';')
values = []
years = []
for row in data:
    if row[0] == "Country or Area":
        continue
    values.append(float(row[5]))
    years.append(int(row[2]))

values_arr = np.array(values)
years_arr = np.array(years)

```

Рисунок 39 – Решение задачи

```

In [41]: print(f"Среднее значение количества женщин к 100 мужчинам: {round(np.mean(values_arr),2)}")
print(f"Средний год рождения: {round(np.mean(years_arr),2)}")

print(f"\nМаксимальное значение отношения: {np.max(values_arr)}")
print(f"Максимальное значение года: {np.max(years_arr)}")

print(f"\nМинимальное значение отношения: {np.min(values_arr)}")
print(f"Минимальное значение года: {np.min(years_arr)}")

print(f"\nСреднее отклонение отношения: {round(np.std(values_arr),2)}")
print(f"Среднее отклонение года: {round(np.std(years_arr),2)}")

```

Среднее значение количества женщин к 100 мужчинам: 99.34
Средний год рождения: 1995.46

Максимальное значение отношения: 200.6
Максимальное значение года: 2004

Минимальное значение отношения: 69.0
Минимальное значение года: 1985

Среднее отклонение отношения: 9.23
Среднее отклонение года: 5.65

Рисунок 40 – Решение задачи

- 1) Получается, что выборка была из людей 1985-2004 года рождения (на 2023 год 19-38 лет).
- 2) Наибольшее отношение девушек к мужчинам 2 к 1
- 3) В среднем отклонение отношения составляет 9.23

Рисунок 41 – Выводы

8. Создать ноутбук, в котором выполнить решение индивидуального задания. Ноутбук должен содержать условие индивидуального задания. При

решении индивидуального задания не должны быть использованы условный оператор `if`, а также операторы циклов `while` и `for`, а только средства библиотеки NumPy. Привести в ноутбуке обоснование принятых решений. Номер варианта индивидуального задания необходимо уточнить у преподавателя.

Вариант 29 (10)

10. Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

```
In [18]: import numpy as np

n = int(input("Введите количество строк: "))
m = int(input("Введите количество столбцов: "))
slau = np.random.randint(0, 15, size = (n, m))
slau
```

Введите количество строк: 3
Введите количество столбцов: 5

```
Out[18]: array([[ 2, 12, 10,  4,  8],
                [ 6, 13,  8, 12,  8],
                [10,  0,  7, 11,  6]])
```

Приведение к треугольному виду:

```
In [20]: tr_slau = np.triu(slau)
tr_slau
```

```
Out[20]: array([[ 2, 12, 10,  4,  8],
                [ 0, 13,  8, 12,  8],
                [ 0,  0,  7, 11,  6]])
```

Найдем количество строк

```
In [37]: rows, columns = tr_slau.shape
print(f"Количество строк: {rows}")
```

Количество строк: 3

Найти количество строк, среднее арифметическое элементов которых меньше заданной величины

```
In [39]: num = int(input("Введите ограничение: "))

ed_mat = np.ones((rows, ))
mean_row_slau = tr_slau.mean(axis=1)

print(f"\nСредние значения: {mean_row_slau}")
print(f"Количество строк: {int(ed_mat[mean_row_slau < num].sum())}")
```

Введите ограничение: 7

Средние значения: [7.2 8.2 4.8]
Количество строк: 1

Рисунок 42 – Решение задачи

9. Зафиксируйте сделанные изменения в репозитории.

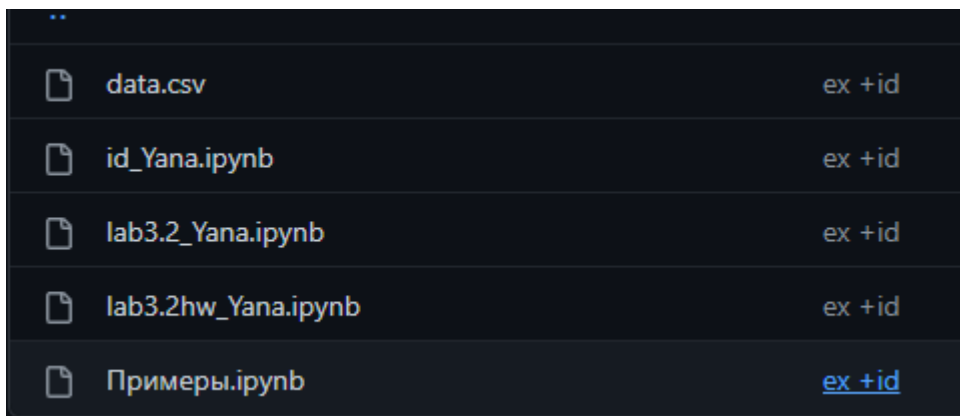


Рисунок 43 – Фиксирование изменений в репозитории

10. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем

Задача: Решение СЛАУ методом Гаусса.

Решение СЛАУ методом Гаусса

Создание расширенной матрицы (A|B)

```
In [198]: import numpy as np

n = int(input("Введите количество строк: "))
m = int(input("Введите количество столбцов: "))
slau = np.random.randint(0, 15, size = (n, m + 1))

#slau_b = slau_a[:, m]
print(slau)

Введите количество строк: 4
Введите количество столбцов: 4
[[ 1  9  9  7  5]
 [ 0  5 10  9 12]
 [ 4  1 13  3 14]
 [ 9 13 12 11  6]]
```

Приведение к треугольному виду

```
In [199]: for j in range(n - 1):
          for i in range(n - 1):
              if i < j:
                  i = j
                  mn = slau[i + 1, j] / slau[j, j]
                  slau[i + 1] = slau[i + 1] - slau[j]* mn
```

Рисунок 44 – Решение задачи

Разделение на А и В

```
In [200]: slau_a = slau[:, 0:m]
slau_b = slau[:, m]
print(slau_a, slau_b)

[[ 1  9  9  7]
 [ 0  5 10  9]
 [ 0  0 47 38]
 [ 0  0  0 15]] [ 5 12 78 12]
```

```
In [201]: x = np.ones([1, n])
x
```

```
Out[201]: array([[1., 1., 1., 1.]])
```

Рисунок 45 – Решение задачи

Нахождение x

```
In [202]: rank_a = np.linalg.matrix_rank(slau[:, 0:m])
rank_ab = np.linalg.matrix_rank(slau)

if rank_a == rank_ab:
    print("Система совместна")
    if rank_a >= m:
        print("Система определена")
        for j in range(n - 1, -1, -1):
            x[0, j] = slau_b[j] / np.sum(slau_a[j, :] * x)
        print(x)
    else:
        print("Система не определена")
else:
    print("Система не совместна, решений нет")
```

```
Система совместна
Система определена
[[0.24369198 0.53865961 1.00775194 0.8      ]]
```

Рисунок 46 – Решение задачи

Вопросы для защиты работы

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?

Через срезы:

- Произвольный элемент ($m[i,j]$)
- Строка ($m[i, :]$)
- Столбец матрицы ($m[:, j]$)
- Часть строки/столбца матрицы ($m[i, j:], m[0:i, j]$)
- Непрерывная часть матрицы ($m[i_1:i_2, j_1:j_2]$)
- Произвольные столбцы/строки матрицы ($col = [0, 1, 2]; m[:, col]$)

4. Как осуществляется расчет статистик по данным?

shape – Размерность массива

argmax – Индексы элементов с максимальным значением (по осям)

argmin – Индексы элементов с минимальным значением (по осям)

max – Максимальные значения элементов (по осям)

min – Минимальные значения элементов (по осям)

mean – Средние значения элементов (по осям)

prod – Произведение всех элементов (по осям)

std – Стандартное отклонение (по осям)

sum – Сумма всех элементов (по осям)

var – Дисперсия (по осям)

5. Как выполняется выборка данных из массивов ndarray?

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

