

Министерство науки и высшего образования РФ
Пензенский государственный университет
Кафедра «Вычислительная техника»

Отчёт
по лабораторной работе №1
по курсу «Нейронные сети в решении практических задач»
на тему «Основы Python. Обучение нейрона»

Выполнил:
студент группы 23ВИ2
Юров Д.М.

Принял:
Митрохин М.А.

Пенза 2026

Задание на лабораторную работу:

- 1) Активируйте созданную в предыдущей работе среду (conda activate dnnenv);
- 2) Запустите IDE Spyder;
- 3) Откройте файл «Lab1_python_basics.py»;
- 4) Прочитайте комментарии и выполните код;
- 5) После ознакомления с кодом выполните Задание 1;
- 6) Откройте файл «lab1_neuron.py»;
- 7) Сделайте каталог лабораторной работы текущим;
- 8) Прочитайте комментарии и выполните код;
- 9) После ознакомления с кодом выполните Задание 2.

Задание 1:

- 1) Создайте список, заполненный случайными числами;
- 2) Создайте цикл, который проходит все элементы списка и суммирует только четные значения;
- 3) Выведите полученную сумму на экран в консоли.

Задание 2:

- 1) Измените программу так, чтобы нейрон работал с тремя признаками

Ход выполнения работы:

1. Активировал среду «dnnenv», созданную в предыдущей работе (рис. 1).

```
C:\Users\izada>conda activate dnnenv  
(dnnenv) C:\Users\izada>_
```

Рисунок 1 - активированная среда

2. Запустил IDE Spyder (рис. 2).



Рисунок 2 - запуск Spyder

3. Открыл файл Lab2_pythonBasics.py (рис. 3).

```
Lab2_pythonBasics.py x  
# -*- coding: utf-8 -*-  
'''  
Created on Thu Feb 20 14:27:38 2025  
@author: ПользовательHP  
'''  
  
## !!! Программа должна запускаться в режиме интерпретатора !!!  
## !!! Читайте комментарии и только после этого запускайте код !!!  
  
# Рассмотрим основы языка python  
# Программа на языке python обычно состоит из нескольких блоков  
  
# 1 блок импорта пакетов (библиотек, модулей) обычно находится в начале файла,  
# но импорты работают в любом месте кода  
  
# импорт может быть всей библиотеки  
import time  
  
# или отдельных функций (модулей, объектов и т.п.) из библиотеки  
from random import randint
```

Рисунок 3 - файл "Lab2_pythonBasics"

4. Прочитал комментарии и выполнил код (рис. 4).

```
In [1]: %runfile D:/PSU/4_term/TGNS/metodichki/lab2/Lab2_python_basics.py --wdir
5
<class 'int'>
0.5
<class 'float'>
.5
<class 'str'>
0.5
<class 'str'>
.5
.5 - это дробное число
р
робно
.5 - это дробно
е число
.-тдб
7
[2, 3]
['Пенза', 'Самара', 'Саратов', 12, 33]
['Пенза', 'Самара', 'Саратов', 12, 33, 0.589]
['Пенза', 'Самара', 'Саратов', 12, 33, 0.589, 1, 2, 3]
[0, 2, 3, 5, 7]
1
2
3
4
5
1
0
1
1
0
10
9
8
7
6
Есть
```

Рисунок 4 - результат работы программы

Задание 1

1. Создал список, заполненный случайными числами (рис. 5).

```
1 import random
2
3 n = 20;
4 a, b = 0, 100;
5 lst = [random.randint(a, b) for _ in range(n)]
```

Рисунок 5 - список

2. Создал цикл, который проходит все элементы списка и суммирует только четные значения (рис. 6).

```
9     sum = 0;
10    for i in lst:
11        if (i % 2 == 0):
12            sum += i;
13
14    print(sum)
```

Рисунок 6 - цикл подсчета четных значений

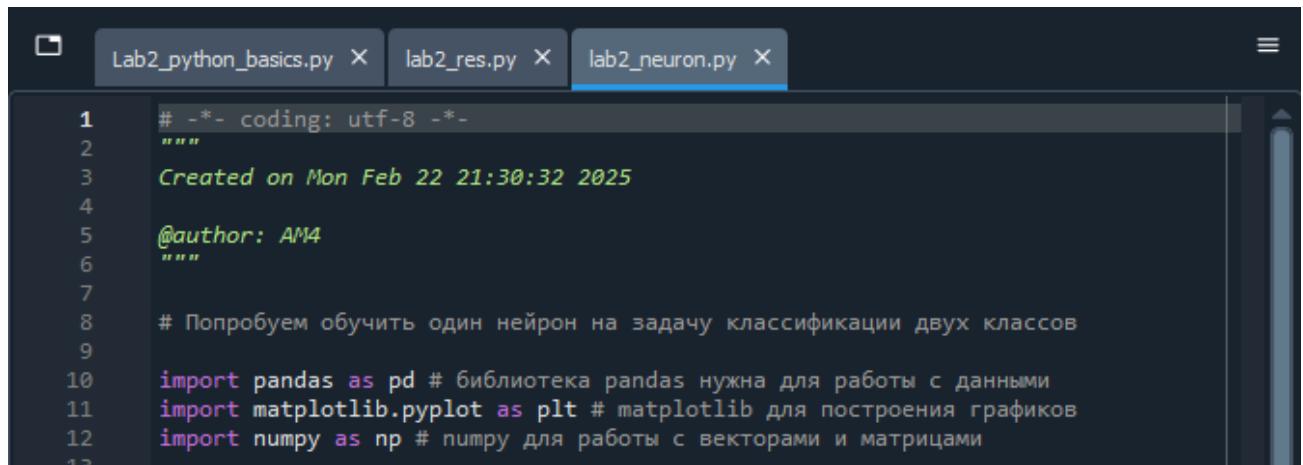
3. Вывел полученную сумму на экран в консоли (рис. 7).

```
In [4]: %runfile D:/PSU/4_term/TGNS/res/lab2/lab2_res.py --wdir
spisok:
[11, 22, 66, 60, 88, 83, 63, 38, 92, 91, 61, 77, 16, 21, 73, 67, 64, 100, 40,
18]

summa chetnih znacheniy: 604
```

Рисунок 7 - результат работы программы

4. Открыл файл «lab2_neuron.py» (рис. 8).



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Feb 22 21:30:32 2025
4
5 @author: AM4
6
7
8 # Попробуем обучить один нейрон на задачу классификации двух классов
9
10 import pandas as pd # библиотека pandas нужна для работы с данными
11 import matplotlib.pyplot as plt # matplotlib для построения графиков
12 import numpy as np # numpy для работы с векторами и матрицами
13
```

Рисунок 8 - файл "lab2_neuron.py"

5. Сделал каталог лабораторной работы текущим (рис. 9).

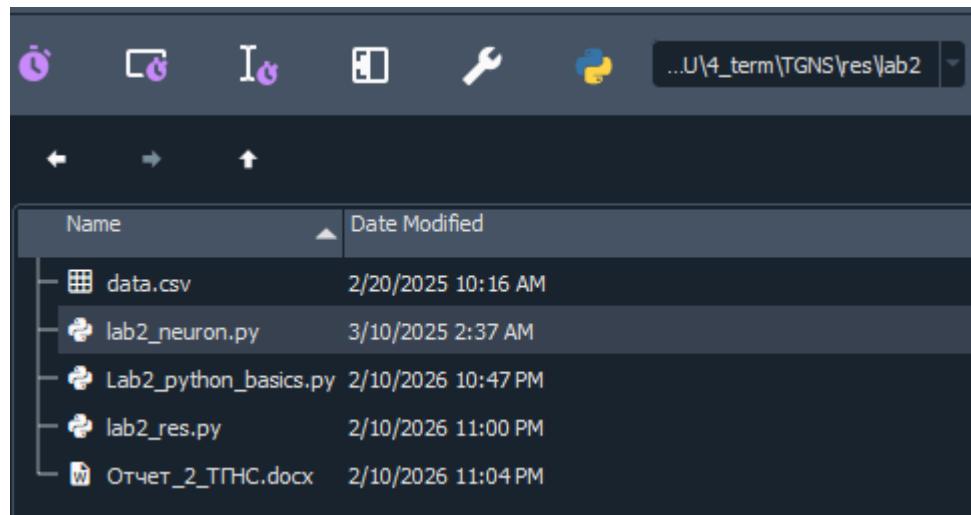
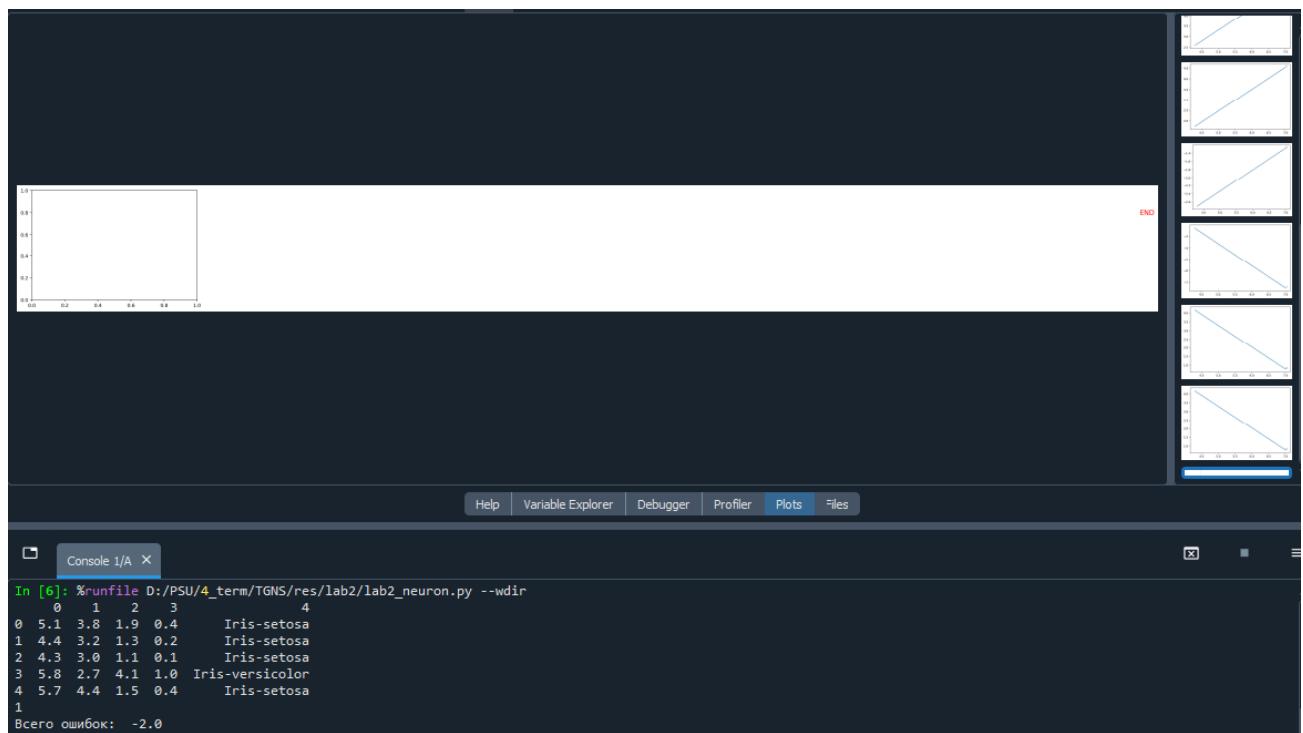


Рисунок 9 - текущий каталог

6. Прочитал комментарии и выполнил код (рис. 10).



Задание 2

- Изменил программу так, чтобы нейрон работал с тремя признаками (рис. 11) и протестировал программу (рис. 12).

```
X = df.iloc[:, [0, 1, 2]].values

# Признаки в X, ответы в y - посмотрим на плоскости как выглядят
#plt.figure()
#plt.scatter(X[y==1, 0], X[y==1, 1], color='red', marker='o')
#plt.scatter(X[y==-1, 0], X[y==-1, 1], color='blue', marker='x')

#3D график
#fig = plt.figure()
#ax = fig.add_subplot(111, projection='3d')
#ax.scatter(X[y==1, 0], X[y==1, 1], X[y==1, 2], color='red', marker='o')
#ax.scatter(X[y==-1, 0], X[y==-1, 1], X[y==-1, 2], color='blue', marker='x')

# переходим к созданию нейрона
# функция нейрона:
# значение = w1*признак1+w2*признак2+w0
# ответ = 1, если значение > 0
# ответ = -1, если значение < 0

def neuron(w,x):
    if((w[1]*x[0] + w[2]*x[1] + w[3]*x[2] + w[0])>=0):
        predict = 1
    else:
        predict = -1
    return predict
```

Рисунок 11 - измененный код

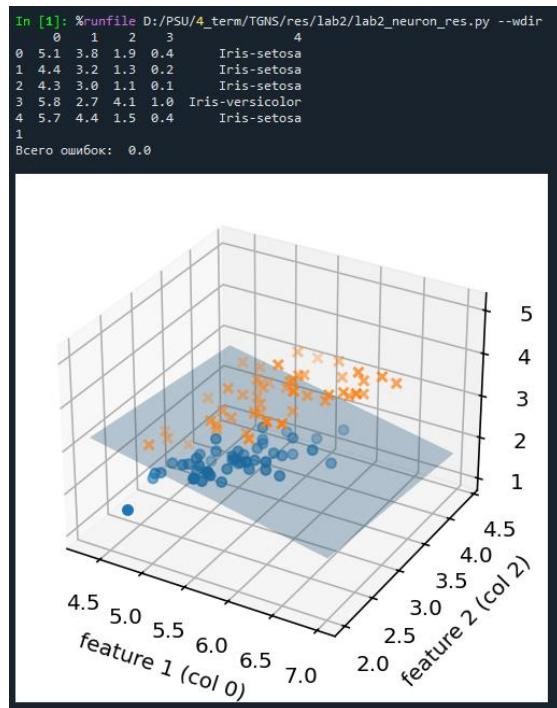


Рисунок 12 - результат работы измененной программы

Вывод: в ходе выполнения лабораторной работы изучил основы python, получил знания, необходимые для модернизации тестового кода и изменил тестовый код по полученному заданию.