

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Департамент цифровых, робототехнических
систем и электроники
Институт перспективной инженерии

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3
дисциплины «Искусственный интеллект и машинное обучение»

Выполнил:
Нерсесян Арман Нельсикович
2 курс, группа ИВТ-б-о-23-2,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Работу проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Института перспективной
инженерии Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Основы работы с библиотекой matplotlib.

Цель: исследовать базовые возможности библиотеки matplotlib языка программирования Python.

Ход работы:

Задание 1. Построение простого графика

Напишите код, который строит график функции $y = x^2$ на интервале $[-10, 10]$. Добавьте заголовок, подписи осей и сетку.

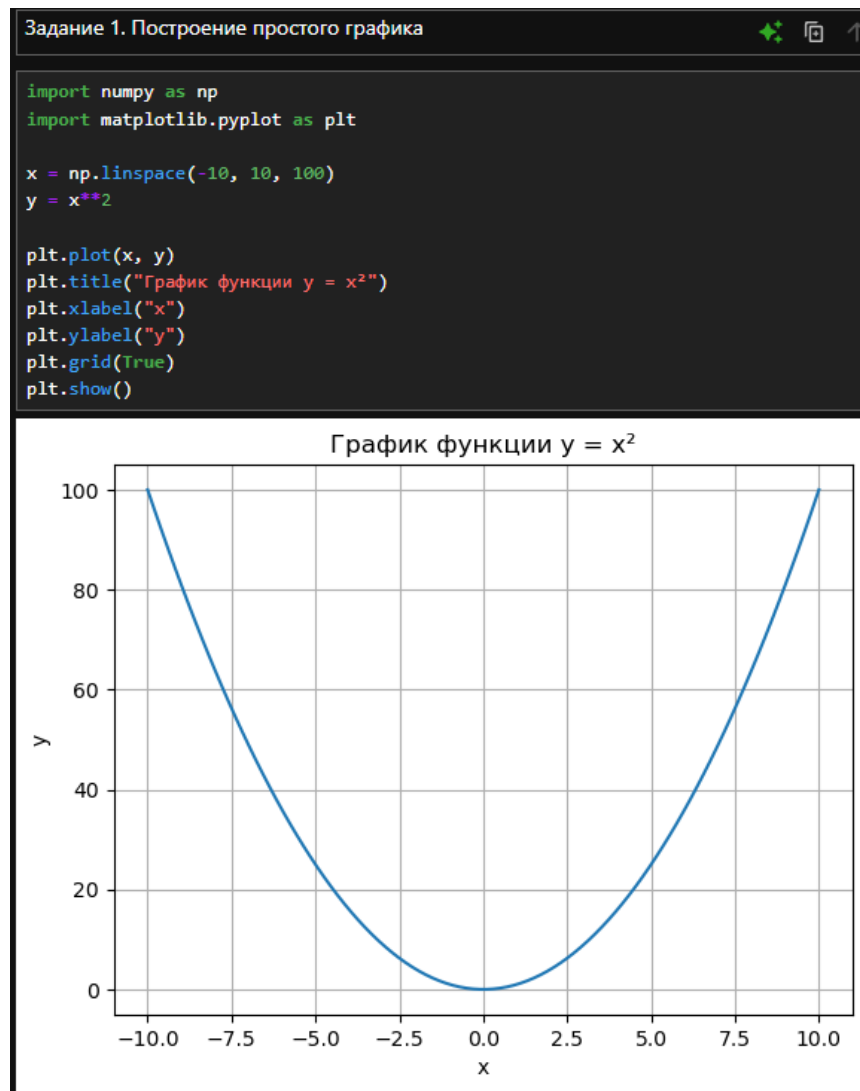


Рисунок 1 – Задание 1

Задание 2. Настройка стилей и цветов

Постройте три линии на одном графике:

$y = x$ (синяя, пунктирная линия),

$y = x^2$ (зеленая, штрих-пунктирная линия),

$y = x^3$ (красная, сплошная линия).

Добавьте легенду и сделайте оси одинакового масштаба.

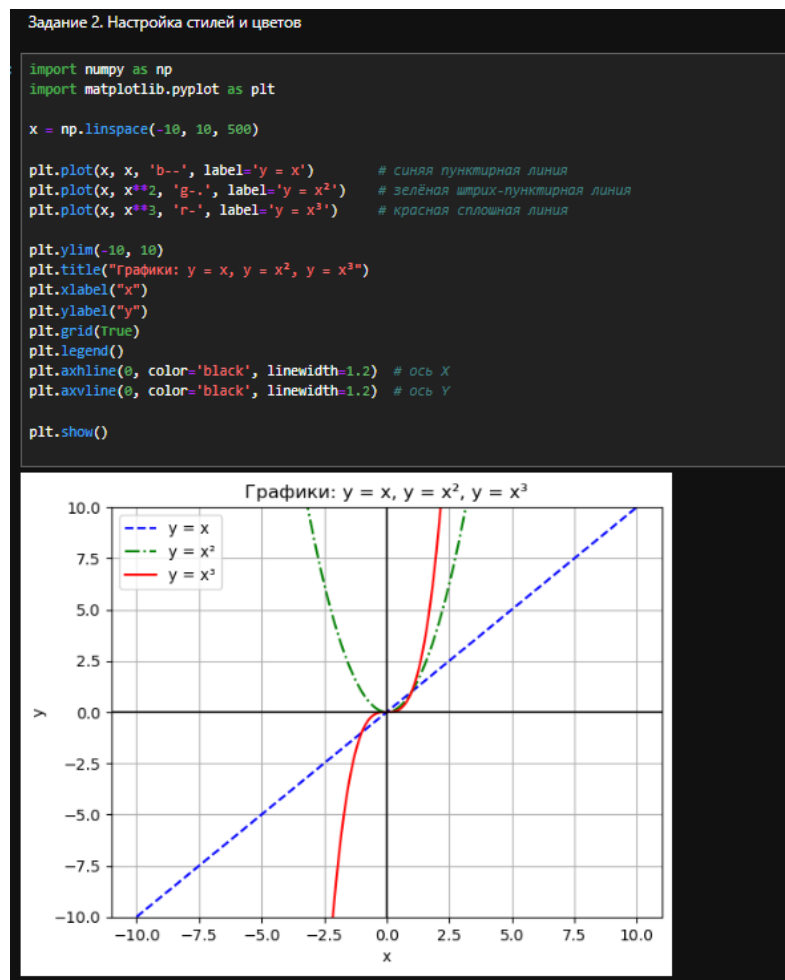


Рисунок 2 – Задание 2

Задание 3. Использование различных типов графиков

Сгенерировал 50 случайных точек и построил диаграмму рассеяния (scatter plot), где цвет точек зависит от их координаты по оси x, а размер точек зависит от координаты по оси y.

Задание 3. Использование различных типов графиков

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)
x = np.random.rand(50) * 100
y = np.random.rand(50) * 100

colors = x
sizes = y

plt.scatter(x, y, c=colors, s=sizes, cmap='viridis', alpha=0.7, edgecolors='black')

plt.title("Диаграмма рассеяния")
plt.xlabel("x")
plt.ylabel("y")
plt.colorbar(label='Значение x')
plt.grid(True)
plt.show()
```

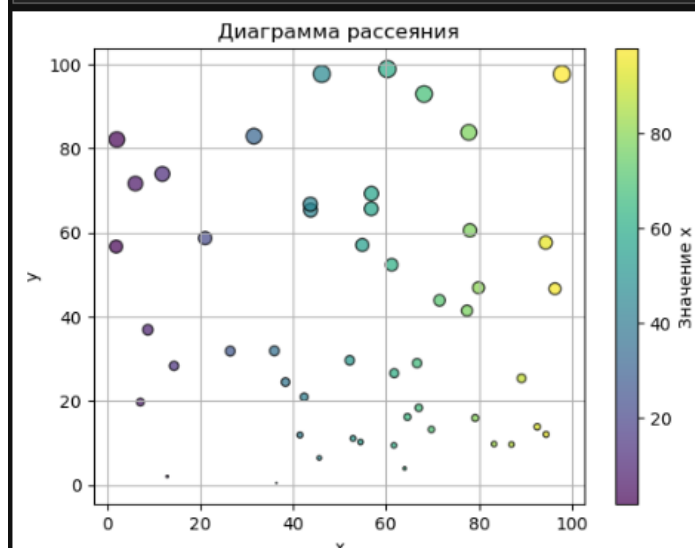


Рисунок 3 – Задание 3

Задание 4. Гистограмма распределения

Сгенерировал 1000 случайных чисел из нормального распределения с параметрами $\mu = 0$, $\sigma = 1$ и построил их гистограмму с 30 бинами. Добавил вертикальную линию в среднем значении.

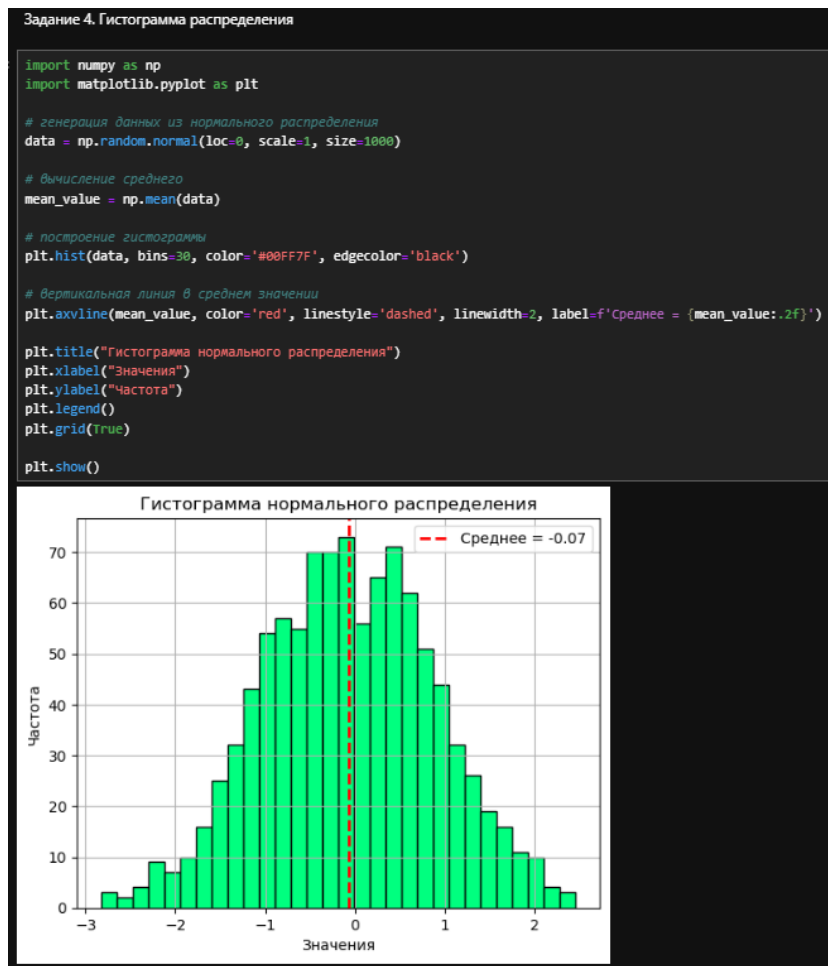


Рисунок 4 – Задание 4

Задание 5. Столбчатая диаграмма

Создайте столбчатую диаграмму, которая показывает количество студентов, получивших оценки:

- "Отлично" — 20 человек,
- "Хорошо" — 35 человек,
- "Удовлетворительно" — 30 человек,
- "Неудовлетворительно" — 15 человек.

Добавьте подписи к осям и заголовок.

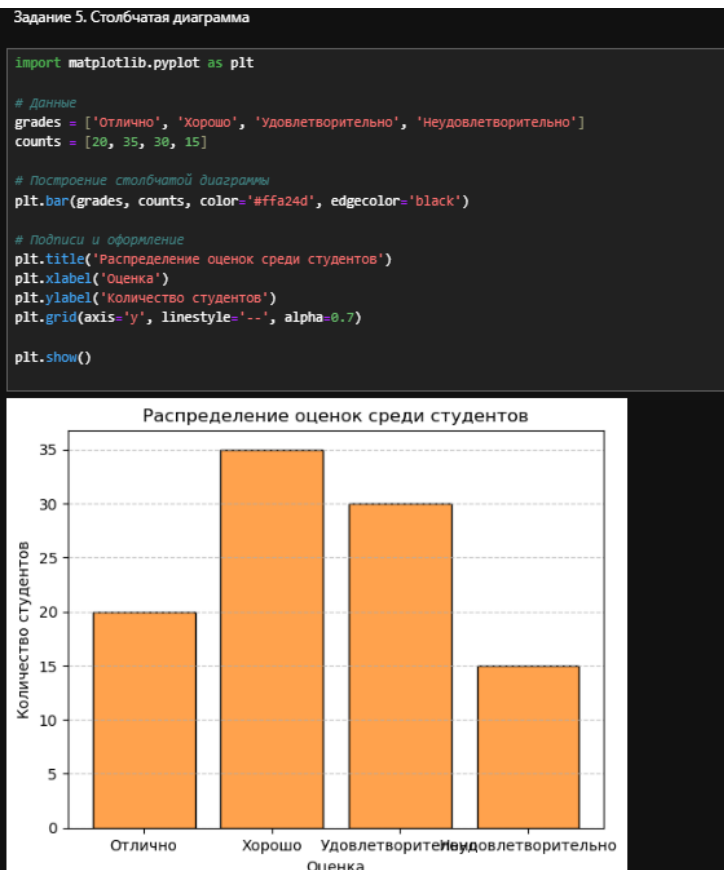


Рисунок 5 – Задание 5

Задание 6. Круговая диаграмма

Используя данные предыдущей задачи, построил круговую диаграмму с процентными подписями секторов.

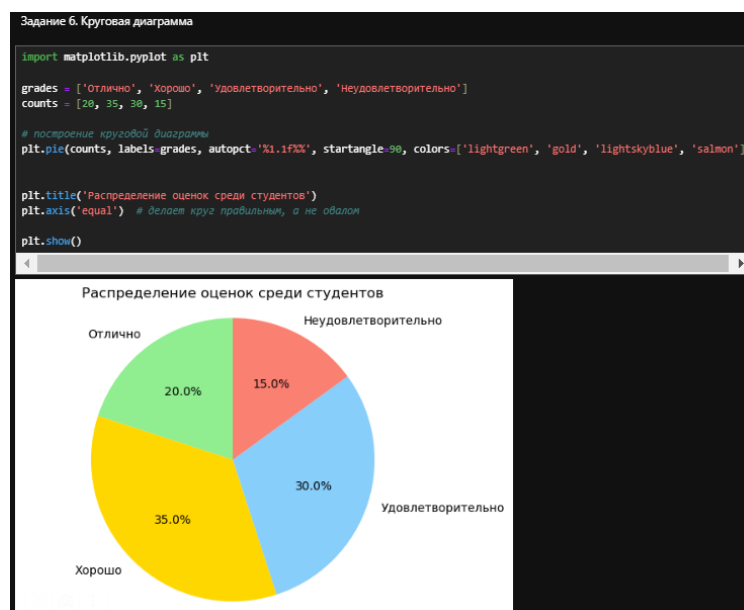


Рисунок 6 – Задание 6

Задание 7. Трехмерный график поверхности

Используя `mpl_toolkits.mplot3d`, постройте 3D-график функции $z = \sin(\sqrt{x^2 + y^2})$ на сетке значений x, y в диапазоне $[-5, 5]$.

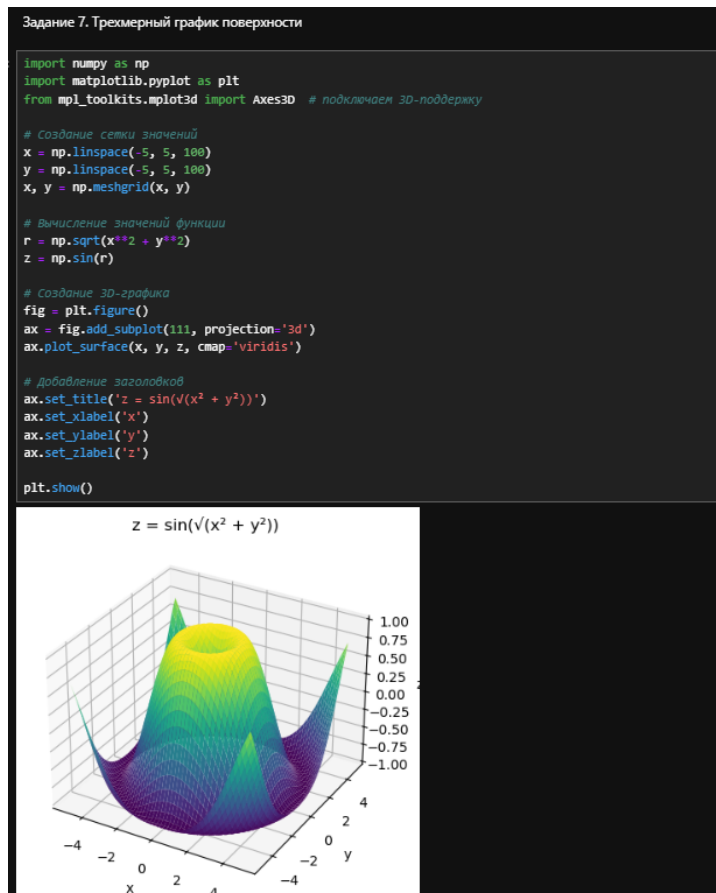


Рисунок 7 – Задание 7

Задание 8. Множественные подграфики (subplots)

Постройте четыре графика в одной фигуре (2×2 сетка):

1. Линейный график $y = x$
2. Парабола $y = x^2$
3. Синус $y = \sin(x)$
4. Косинус $y = \cos(x)$

Добавьте заголовки к каждому подграфику.

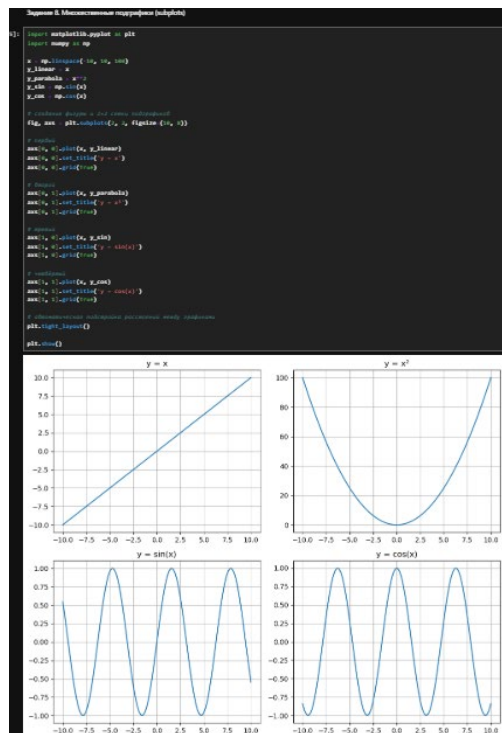


Рисунок 8 – Задание 8

Задание 9. Тепловая карта (imshow)

Создайте случайную матрицу 10×10 с элементами от 0 до 1 и визуализируйте её как тепловую карту с цветовой шкалой.

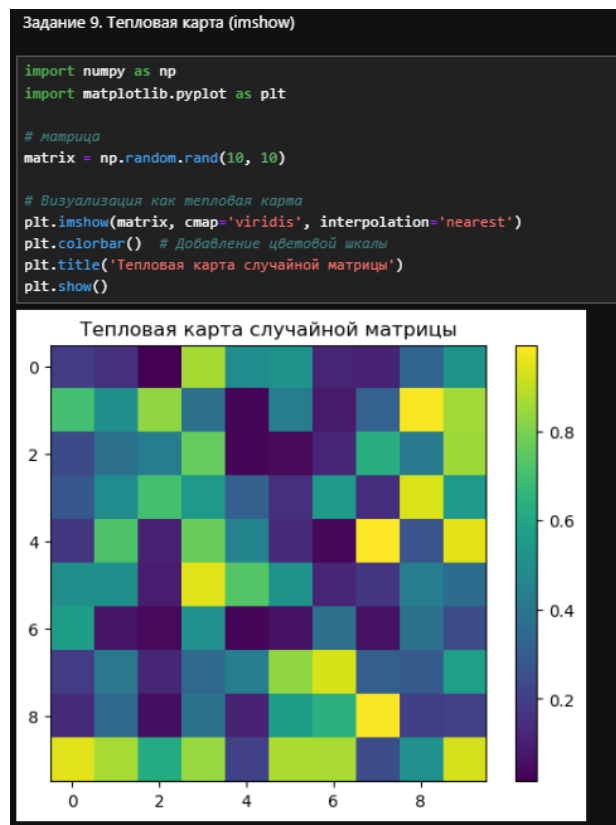


Рисунок 9 – Задание 9

Индивидуальное задание 1. Распад радиоактивного элемента

Исследователи измеряли распад радиоактивного вещества:

- Время (часы): [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Количество атомов (млн): [100, 85, 72, 60, 50, 42, 35, 30, 26, 22, 19]

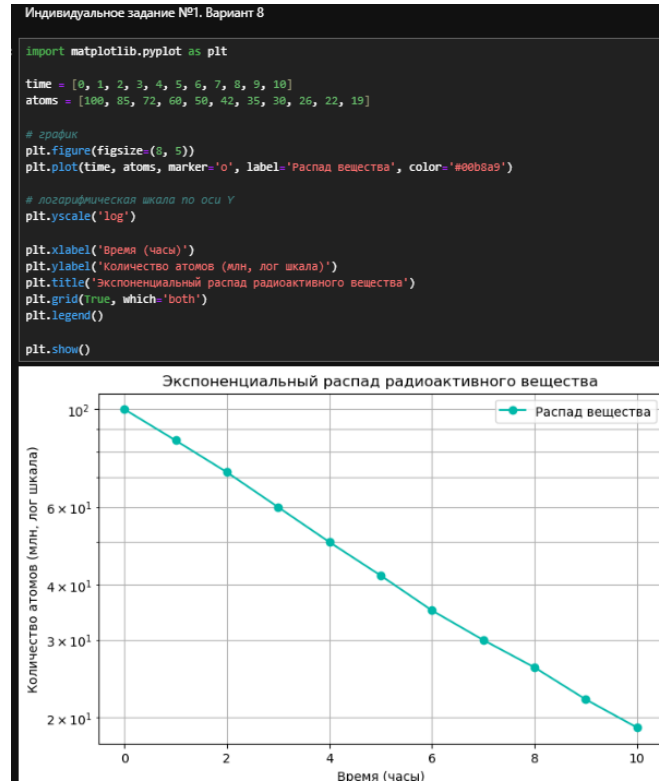


Рисунок 10 – Индивидуальное задание 1

Индивидуальное задание 2. Длительность обучения в университетах разных стран.

Сравнение среднего времени обучения в университетах разных стран:

- Страны: ['США', 'Германия', 'Франция', 'Япония', 'Россия']
- Лет обучения: [4, 3.5, 3, 4.5, 4]

Отобразите диаграмму горизонтально и добавьте сетку.



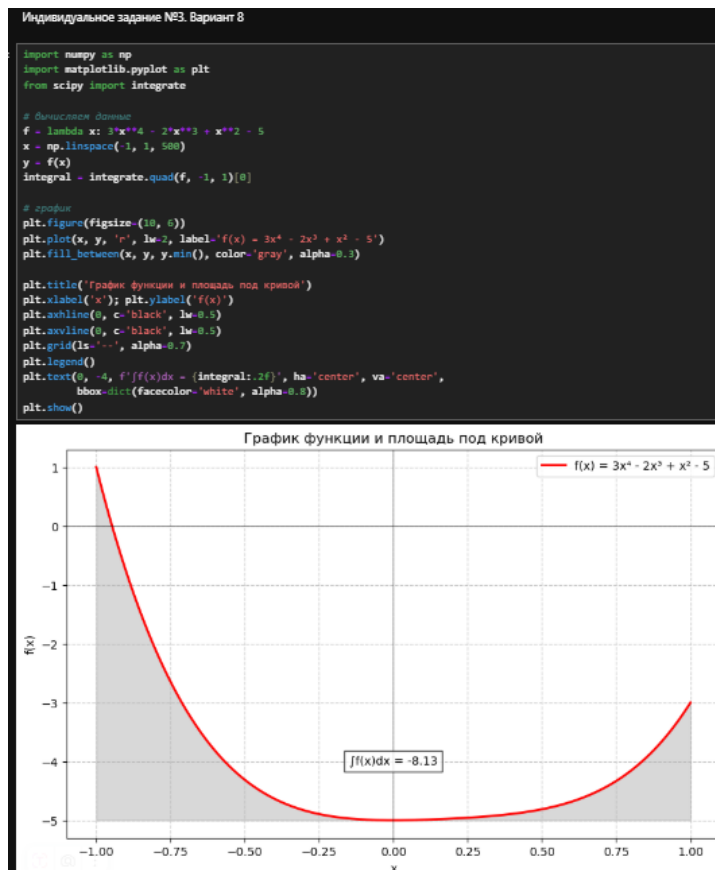
Рисунок 11 – Индивидуальное задание 2

Индивидуальное задание 3. Площадь под многочленом

Вычислить интеграл:

$$f(x) = 3x^4 - 2x^3 + x^2 - 5$$

на интервале $[-1, 1]$.



Индивидуальное задание 4. Волнообразная поверхность

Визуализируйте функцию:

$$f(x, y) = \sin(x^2 + y^2)$$

на $x, y \in [-2\pi, 2\pi]$.

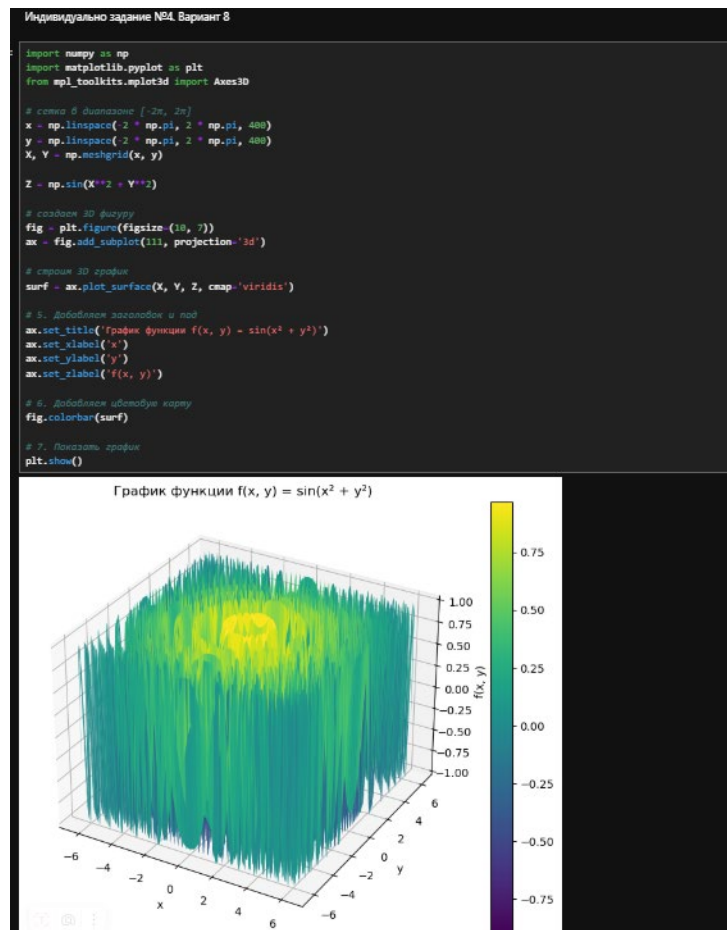


Рисунок 13 – Индивидуальное задание 4

1. Как осуществляется установка пакета matplotlib?

Устанавливается через терминал или командную строку с помощью `pip install matplotlib`. В Jupyter можно установить через `!pip install matplotlib`.

2. Какая "магическая" команда должна присутствовать в ноутбуках Jupyter для корректного отображения графиков matplotlib?

`%matplotlib inline` — специальная команда, которая позволяет отображать графики прямо внутри ячеек Jupyter Notebook.

3. Как отобразить график с помощью функции plot ?

Необходимо передать данные по оси X и Y в функцию `plot()`, затем вызвать `show()` для отображения графика.

4. Как отобразить несколько графиков на одном поле?

Можно вызвать `plot()` несколько раз с разными данными, добавив к ним легенду и отличив по цвету, стилю или маркерам.

5. Какой метод Вам известен для построения диаграмм категориальных данных?

Основной метод — `bar()` для вертикальных столбцов и `barh()` для горизонтальных. Также используется `pie()` для круговых диаграмм.

6. Какие основные элементы графика Вам известны?

Основные элементы графика включают оси (x и y), заголовок, метки осей, линии сетки, легенду, текстовые надписи и сами графические данные (линии, маркеры, столбцы и т. д.).

7. Как осуществляется управление текстовыми надписями на графике?

Управление текстовыми надписями на графике осуществляется с помощью функций `text()`, `xlabel()`, `ylabel()` и `title()`. Эти функции позволяют задавать текст, его положение, размер, шрифт и другие параметры.

8. Как осуществляется управление легендой графика?

Управление легендой графика выполняется с помощью функции `legend()`. Можно указать её положение, заголовок, шрифт, а также скрыть или отобразить. Легенда автоматически формируется из меток, заданных в функциях построения графиков.

9. Как задать цвет и стиль линий графика?

Цвет и стиль линий графика задаются через параметры `color` и `linestyle` в функциях построения, например, `plot(x, y, color='red', linestyle='--')`. Можно использовать HEX-коды, RGB-значения или стандартные названия цветов.

10. Как выполнить размещение графика в разных полях?

Размещение графиков в разных полях выполняется с помощью `subplots()`. Например, `fig, (ax1, ax2) = plt.subplots(1, 2)` создаст два поля для графиков в одной строке.

11. Как выполнить построение линейного графика с помощью matplotlib?

Линейный график строится с помощью `plot(x, y)`, где `x` и `y` — массивы данных. Можно добавлять метки, изменять стиль и цвет линии.

12. Как выполнить заливку области между графиком и осью? Между двумя графиками?

Заливка области между графиком и осью выполняется с помощью `fill_between(x, y)`. Между двумя графиками — `fill_between(x, y1, y2)`.

13. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию?

Выборочная заливка выполняется с использованием условия в `fill_between()`, например:

```
plt.fill_between(x, y, where=(y > 0), color='green')
```

14. Как выполнить двухцветную заливку?

Двухцветная заливка реализуется двумя вызовами `fill_between()` с разными условиями и цветами.

15. Как выполнить маркировку графиков?

Маркировка графиков выполняется с помощью параметра `marker` в `plot()`, например, `marker='o'` для кружков. Также можно задавать размер и цвет маркеров.

16. Как выполнить обрезку графиков?

Обрезка графиков выполняется с помощью `set_xlim()` и `set_ylim()`, которые ограничивают отображаемую область.

17. Как построить ступенчатый график? В чем особенность ступенчатого графика?

Ступенчатый график строится с помощью `step(x, y)`. Его особенность в том, что значения меняются резко, как ступеньки, а не плавно.

18. Как построить стековый график? В чем особенность стекового графика?

Стековый график строится с помощью `stackplot(x, y1, y2)`. Его особенность в том, что значения суммируются, показывая вклад каждого компонента.

19. Как построить stem-график? В чем особенность stem-графика?

Stem-график строится с помощью `stem(x, y)`. Он отображает данные в виде вертикальных линий с маркерами на концах, что удобно для дискретных сигналов.

20. Как построить точечный график? В чем особенность точечного графика?

Точечный график строится с помощью `scatter(x, y)`. Его особенность в том, что он отображает отдельные точки без соединения линиями, что полезно для визуализации распределения.

21. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

Столбчатые диаграммы строятся с помощью `bar(x, height)` для вертикальных и `barh(y, width)` для горизонтальных. Можно задавать ширину, цвет и подписи.

22. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с errorbar элементом?

Групповая столбчатая диаграмма — это несколько столбцов для каждой категории, расположенных рядом. Диаграмма с `errorbar` включает отображение погрешностей в виде вертикальных линий.

23. Как выполнить построение круговой диаграммы средствами matplotlib?

Круговая диаграмма строится с помощью `pie(sizes, labels=labels)`. Можно задавать цвета, выносы и тени.

24. Что такое цветовая карта? Как осуществляется работа с цветовыми картами в matplotlib?

Цветовая карта (colormap) определяет переход цветов для визуализации данных. В matplotlib используются функции imshow() или scatter() с параметром cmap.

25. Как отобразить изображение средствами matplotlib?

Изображение отображается с помощью imshow(array), где array — массив пикселей. Можно настраивать цветовую схему и интерполяцию.

26. Как отобразить тепловую карту средствами matplotlib?

Тепловая карта строится с помощью imshow() или pcolormesh() с цветовой картой, например cmap='hot'.

27. Как выполнить построение линейного 3D-графика с помощью matplotlib?

Линейный 3D-график строится с помощью plot3D(x, y, z) в объекте Axes3D.

28. Как выполнить построение точечного 3D-графика с помощью matplotlib?

Точечный 3D-график строится с помощью scatter3D(x, y, z) с возможностью настройки цвета и размера точек.

29. Как выполнить построение каркасной поверхности с помощью matplotlib?

Каркасная поверхность строится с помощью plot_wireframe(X, Y, Z), где X, Y, Z — координаты точек.

30. Как выполнить построение трехмерной поверхности с помощью matplotlib?

Трехмерная поверхность строится с помощью plot_surface(X, Y, Z). Можно задавать цветовую карту и прозрачность.

Вывод: исследовали базовые возможности библиотеки matplotlib языка программирования Python.