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

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины
«Искусственный интеллект и машинное обучение»
Вариант № 7

Выполнил:
Наталичев Данил Андреевич
2 курс, группа ИВТ-б-о-23-2,
09.03.01 «Информатика и вычислительная техника»,
направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»,
очная форма обучения

_____
(подпись)

Проверил:
Доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии
Воронкин В.И

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема работы: Основы работы с библиотекой matplotlib”

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

Порядок выполнения работы:

Ссылка на github: https://github.com/AlafurZerk/ML_Lab_3.

1. Построение простого графика.

1. Построение простого графика

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

```
[0]: import matplotlib.pyplot as plt
     import numpy as np
     from mpl_toolkits.mplot3d import Axes3D


     def plot_quadratic_function(x_start=-10, x_end=10, num_points=100):

         x = np.linspace(x_start, x_end, num_points)

         # Вычисляем значения y
         y = x**2

         # Создаем график
         plt.plot(x, y)

         # Добавляем заголовок
         plt.title("График функции y = x^2")

         # Добавляем подписи осей
         plt.xlabel("x")
         plt.ylabel("y")

         # Добавляем сетку
         plt.grid(True)

         # Отображаем график
         plt.show()


     if __name__ == "__main__":
         plot_quadratic_function()
```

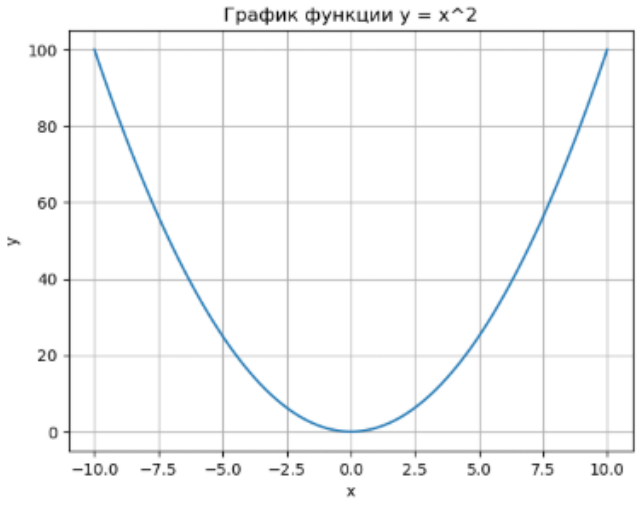

Рисунок 1 – Код и график к заданию 1

2. Стили графиков.

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


def plot_functions():

    x = np.linspace(-5, 5, 100)
    y1 = x
    y2 = x**2
    y3 = x**3

    plt.plot(x, y1, "b:", label="y = x")  # синяя, пунктирная
    plt.plot(x, y2, "g-.", label="y = x^2")  # зеленая, штрих-пунктирная
    plt.plot(x, y3, "r-", label="y = x^3")  # красная, сплошная

    plt.xlabel("x")
    plt.ylabel("y")
    plt.title("Графики функций y=x, y=x^2, y=x^3")
    plt.legend()
    plt.grid(True)
    plt.show()


if __name__ == "__main__":
    plot_functions()
```

Рисунок 2 – Код к заданию 2

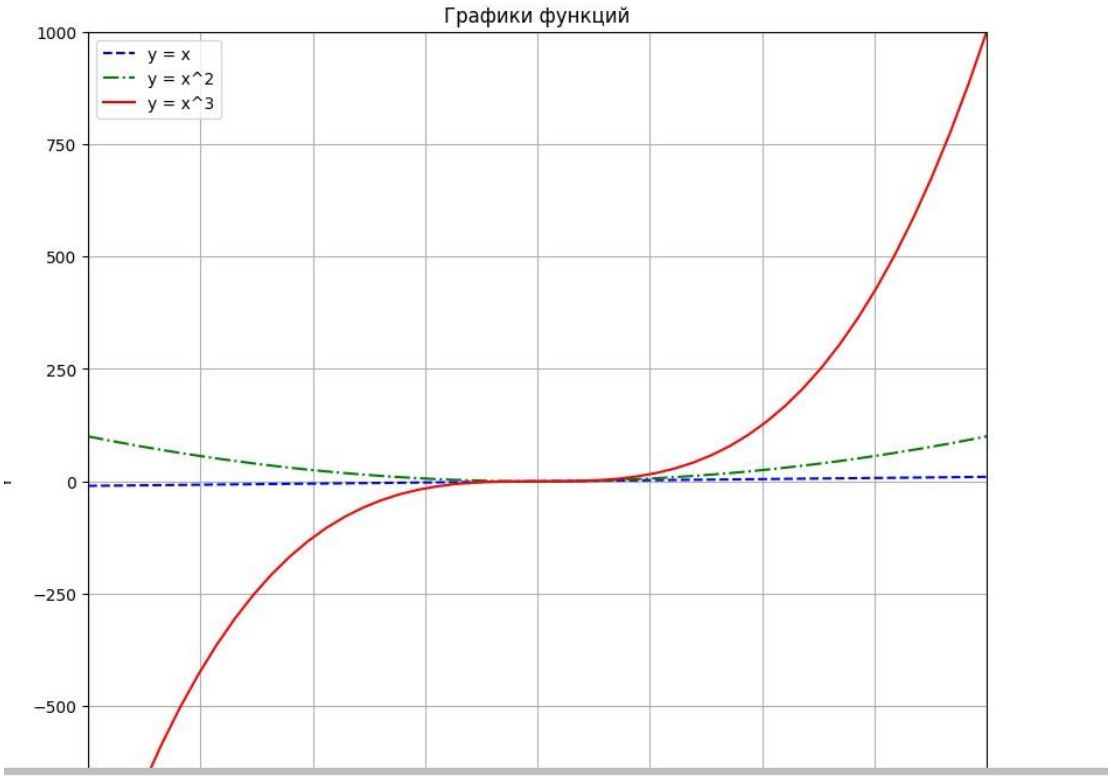

Рисунок 3 – График к заданию 2

3. Использование различных типов графики.

3. Использование различных типов графики

Описание задания:

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

```
def create_scatter_plot():

    x = np.random.rand(100) * 10
    y = np.random.rand(100) * 10
    colors = y
    sizes = y * 100

    fig, ax = plt.subplots(figsize=(8, 6))

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

    ax.set_title("Диаграмма рассеяния со случайными данными")
    ax.set_xlabel("Значения по оси X")
    ax.set_ylabel("Значения по оси Y")

    cbar = fig.colorbar(scatter, label="Значение цвета (Y)")

    ax.grid(True, linestyle='--', alpha=0.5)
    ax.set_facecolor('#f0f0f0')

    plt.show()

if __name__ == "__main__":
    create_scatter_plot()
```

Рисунок 4 – Код к заданию 3

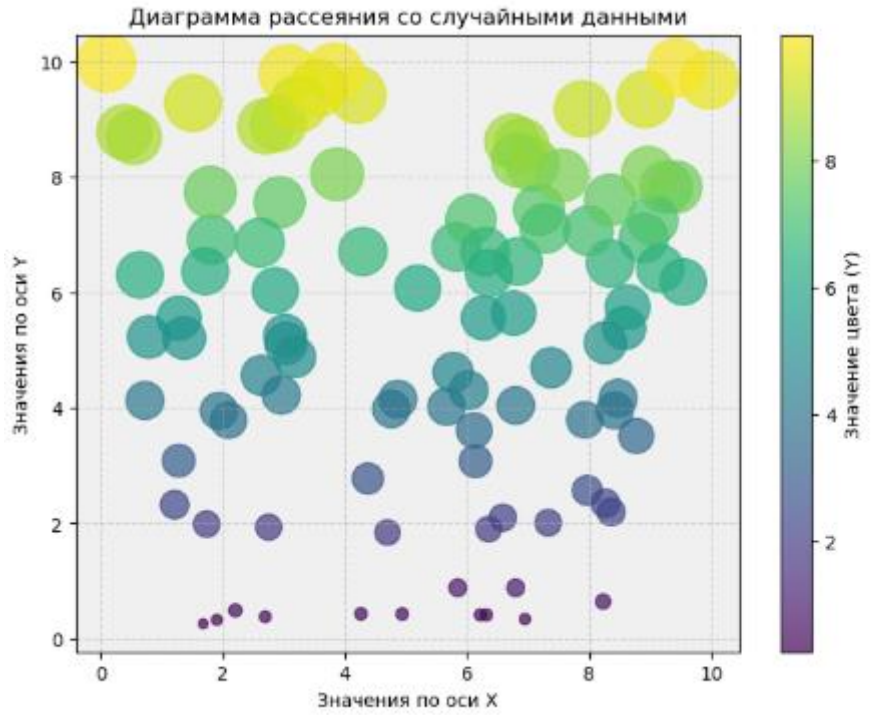

Рисунок 5 – График к заданию 3

4. Гистограмма распределения.

```python
def create_histogram():

    data = np.random.normal(0, 1, 1000)
    mean = np.mean(data)

    fig, ax = plt.subplots(figsize=(8, 6))

    ax.hist(data, bins=30, color='skyblue', edgecolor='black', alpha=0.7)
    ax.axvline(mean, color='red', linestyle='--', label=f'Среднее: {mean:.2f}')

    ax.set_title("Гистограмма нормального распределения")
    ax.set_xlabel("Значения")
    ax.set_ylabel("Частота")
    ax.legend()

    ax.grid(axis='y', linestyle='--', alpha=0.7)
    ax.set_facecolor('#f8f8f8')

    plt.show()

if __name__ == "__main__":
    create_histogram()
```

Рисунок 6 – Код к заданию 4

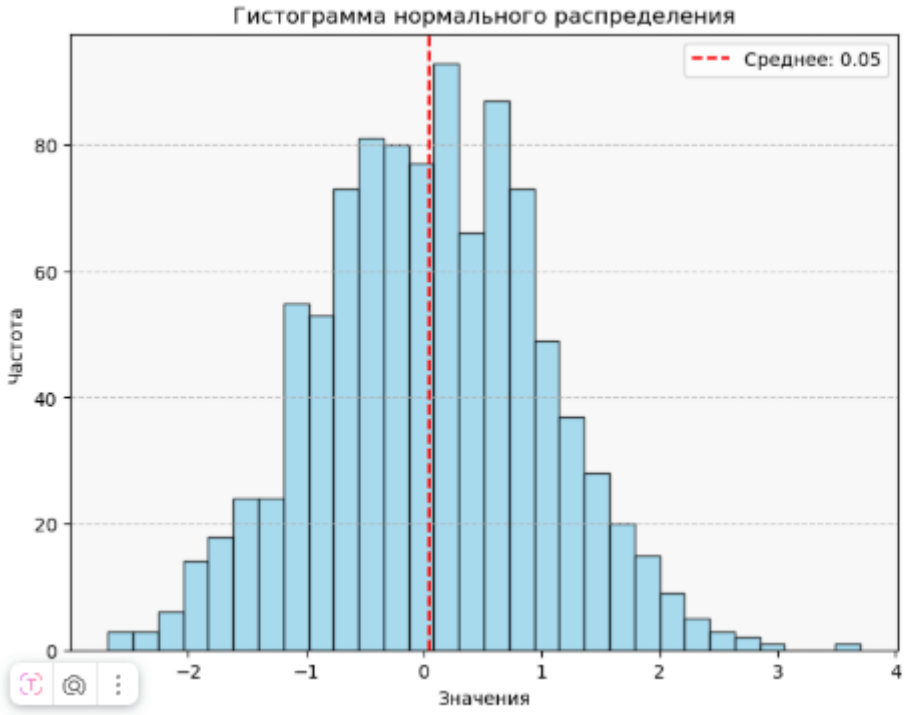

Рисунок 7 – График к заданию 4

5. Столбчатая диаграмма

```python
def create_bar_chart():

    students = {"great": 20, "good": 35, "satisfactorily": 30, "unsatisfactory": 15}
    categories = list(students.keys())
    values = list(students.values())

    fig, ax = plt.subplots(figsize=(8, 6))

    ax.bar(categories, values, color='skyblue')

    ax.set_title('Распределение студентов по оценкам')
    ax.set_xlabel("Оценки")
    ax.set_ylabel("Количество студентов")

    ax.grid(axis='y', linestyle='--', alpha=0.7)
    ax.set_facecolor('#f8f8f8')

    plt.show()


if __name__ == "__main__":
    create_bar_chart()
```

Рисунок 8 – Код к заданию 5

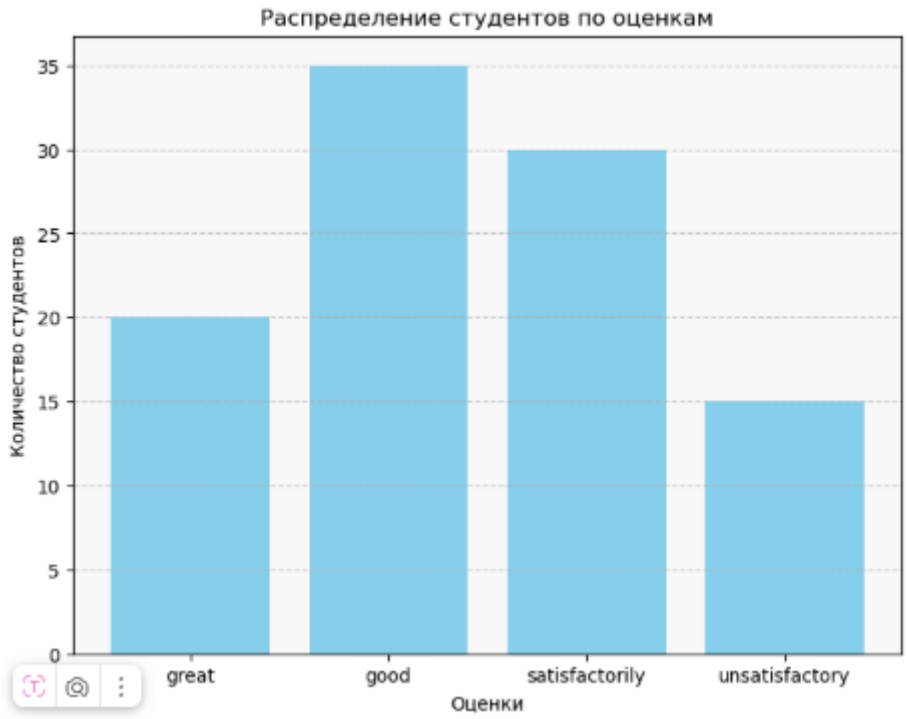

Рисунок 9 – График к заданию 5

6. Круговая диаграмма.

```python
def create_pie_chart():

    students = {"great": 20, "good": 35, "satisfactorily": 30, "unsatisfactory": 15}
    values = list(students.values())
    colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99']
    labels = [f"{key} ({value})" for key, value in students.items()]

    fig, ax = plt.subplots(figsize=(8, 6))

    ax.pie(values,
           labels=labels,
           autopct='%1.1f%%',
           colors=colors,
           startangle=90,
           wedgeprops={'edgecolor': 'black'})

    ax.set_title('Распределение студентов по оценкам')
    ax.axis('equal')

    plt.show()

if __name__ == "__main__":
    create_pie_chart()
```

Рисунок 10 – Код к заданию 6

Рисунок 11 – График к заданию 6

7. Трёхмерный график.

```python
def create_3d_surface_plot():

    x_arr = np.linspace(-5, 5, 100)
    y_arr = np.linspace(-5, 5, 100)
    X, Y = np.meshgrid(x_arr, y_arr)
    Z = np.sin(np.sqrt(X**2 + Y**2))

    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection='3d')

    ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='k', alpha=0.8)

    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    ax.set_title(r'$z = \sin(\sqrt{x^2 + y^2})$')

    plt.show()

if __name__ == "__main__":
    create_3d_surface_plot()
```

Рисунок 12 – Код к заданию 7

$$z = \sin(\sqrt{x^2 + y^2})$$

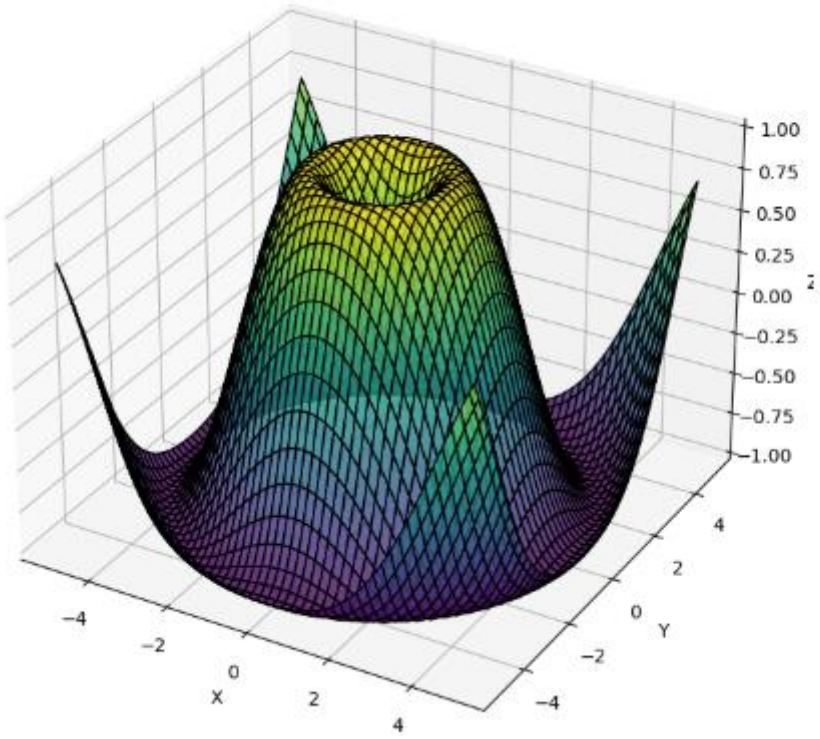

Рисунок 13 – График к заданию 7

8. Множественные подграфики(subplots).

```python
def create_subplot_grid():

    x_arr = np.linspace(-5, 5, 100)
    y1_arr = x_arr
    y2_arr = x_arr**2
    y3_arr = np.sin(x_arr)
    y4_arr = np.cos(x_arr)

    fig, axes = plt.subplots(2, 2, figsize=(10, 8))

    axes[0, 0].plot(x_arr, y1_arr, label='$y = x$', color='b')
    axes[0, 0].set_title('Линейный график')
    axes[0, 0].set_xlabel('x')
    axes[0, 0].set_ylabel('y')
    axes[0, 0].grid(True, linestyle='--', alpha=0.5)
    axes[0, 0].legend()

    axes[0, 1].plot(x_arr, y2_arr, label='$y = x^2$', color='r')
    axes[0, 1].set_title('Парабола')
    axes[0, 1].set_xlabel('x')
    axes[0, 1].set_ylabel('y')
    axes[0, 1].grid(True, linestyle='--', alpha=0.5)
    axes[0, 1].legend()

    axes[1, 0].plot(x_arr, y3_arr, label='$y = sin(x)$', color='g')
    axes[1, 0].set_title('Синус')
    axes[1, 0].set_xlabel('x')
    axes[1, 0].set_ylabel('y')
    axes[1, 0].grid(True, linestyle='--', alpha=0.5)
    axes[1, 0].legend()

    axes[1, 1].plot(x_arr, y4_arr, label='$y = cos(x)$', color='m')
    axes[1, 1].set_title('Косинус')
    axes[1, 1].set_xlabel('x')
    axes[1, 1].set_ylabel('y')
    axes[1, 1].grid(True, linestyle='--', alpha=0.5)
    axes[1, 1].legend()

    fig.suptitle('Четыре графика в одной фигуре', fontsize=14)

    plt.tight_layout(rect=[0, 0, 1, 0.95])
    plt.show()

if __name__ == "__main__":
    create_subplot_grid()
```

Рисунок 14 – Код к заданию 8

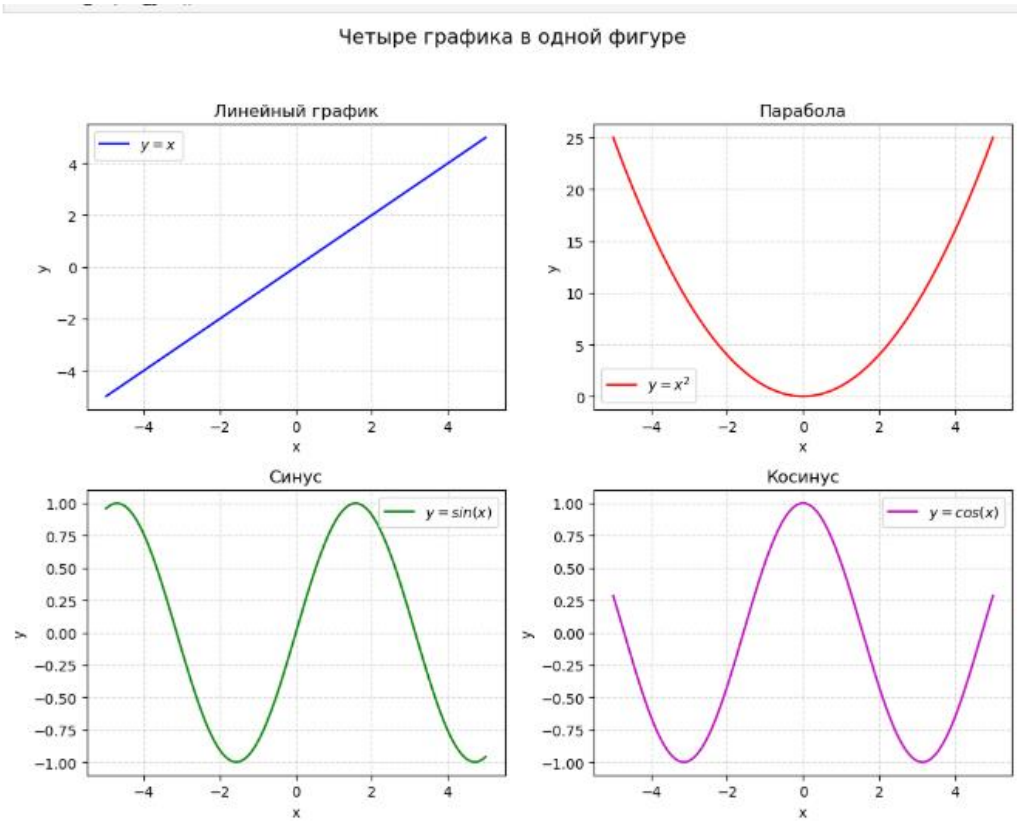

Рисунок 15 – Графики к заданию 8

9. Тепловая карта(imshow).

```
def create_pcolormesh_plot():

    data = np.random.rand(10, 10)

    fig, ax = plt.subplots(figsize=(8, 6))

    mesh = ax.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')

    fig.colorbar(mesh, ax=ax)

    ax.set_title("Pcolormesh график случайных данных")
    ax.set_xlabel("X")
    ax.set_ylabel("Y")

    plt.show()

if __name__ == "__main__":
    create_pcolormesh_plot()
```

Рисунок 16 – Код к заданию 9

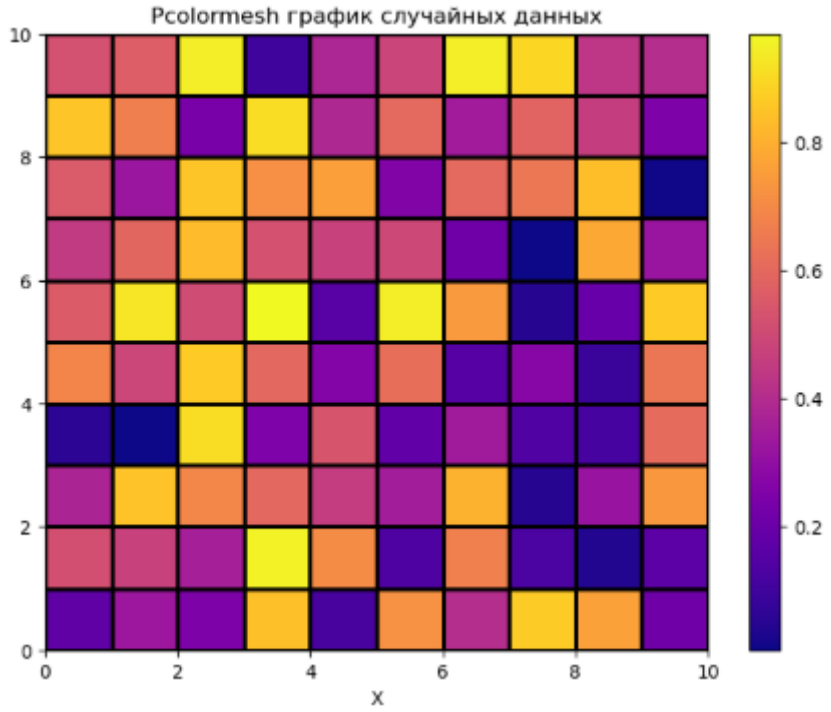

Рисунок 17 – График к заданию 9

10. Индивидуальное задание.

7. Изменение уровня загрязнения воздуха

Постройте график, показывающий изменение уровня загрязнения воздуха в течение дня.

Исходные данные:

- **Время суток (часы):** `[6, 8, 10, 12, 14, 16, 18, 20, 22, 24]`
- **Уровень загрязнения (мг/м³):** `[0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 0.8, 0.6, 0.4, 0.3]`

```
def plot_air_pollution():

    time = [6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
    pollution_level = [0.2, 0.3, 0.5, 0.7, 0.8, 0.9, 0.8, 0.6, 0.4, 0.3]

    plt.figure(figsize=(10, 6))

    plt.plot(time, pollution_level, linestyle='-', marker='o', label='Сплошная линия')
    plt.plot(time, pollution_level, linestyle='--', marker='x', label='Пунктирная линия')
    plt.plot(time, pollution_level, linestyle=':', marker='.', label='Точечная линия')

    plt.xlabel("Время суток (часы)")
    plt.ylabel("Уровень загрязнения (мг/м³)")
    plt.title("Уровень загрязнения воздуха в течение дня")
    plt.grid(True)
    plt.legend()
    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    plot_air_pollution()
```

Рисунок 18 – Код к ИЗ №1

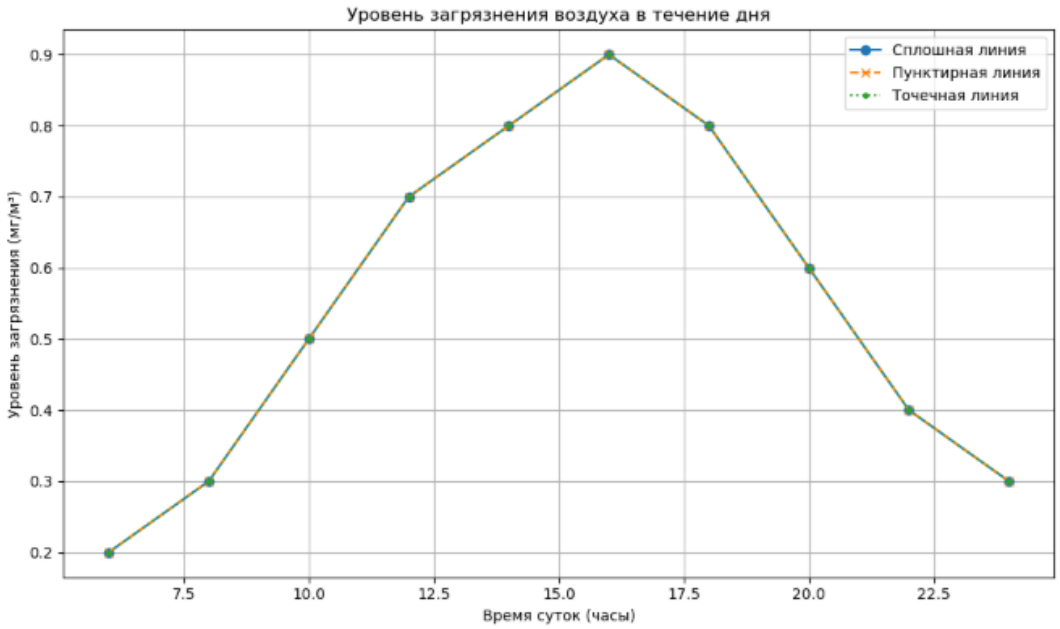

Рисунок 19 – График к ИЗ №1

7. Результаты голосования за лучший фильм года

Постройте столбчатую диаграмму, отображающую результаты голосования за лучший фильм года.

Исходные данные:

- **Фильмы:** `['Фильм A', 'Фильм B', 'Фильм C', 'Фильм D', 'Фильм E']`
- **Голоса:** `[1200, 900, 1500, 800, 1100]`

5]:
```python
def plot_movie_votes():

    movies = ['Фильм A', 'Фильм B', 'Фильм C', 'Фильм D', 'Фильм E']
    votes = [1200, 900, 1500, 800, 1100]
    colors = ['red', 'green', 'blue', 'orange', 'purple']

    # Создаем столбчатую диаграмму
    plt.figure(figsize=(10, 6))

    x = np.arange(len(movies))
    plt.bar(x, votes, color=colors)

    plt.xlabel("Фильмы")
    plt.ylabel("Количество голосов")
    plt.title("Результаты голосования за лучший фильм года")

    plt.xticks(x, movies)

    plt.grid(axis='y', alpha=0.75)

    for i, v in enumerate(votes):
        plt.text(x[i] - 0.15, v + 25, str(v), color='black', fontweight='bold')

    plt.tight_layout()

    plt.show()


if __name__ == "__main__":
    plot_movie_votes()
```

Рисунок 20 – Код к ИЗ №2

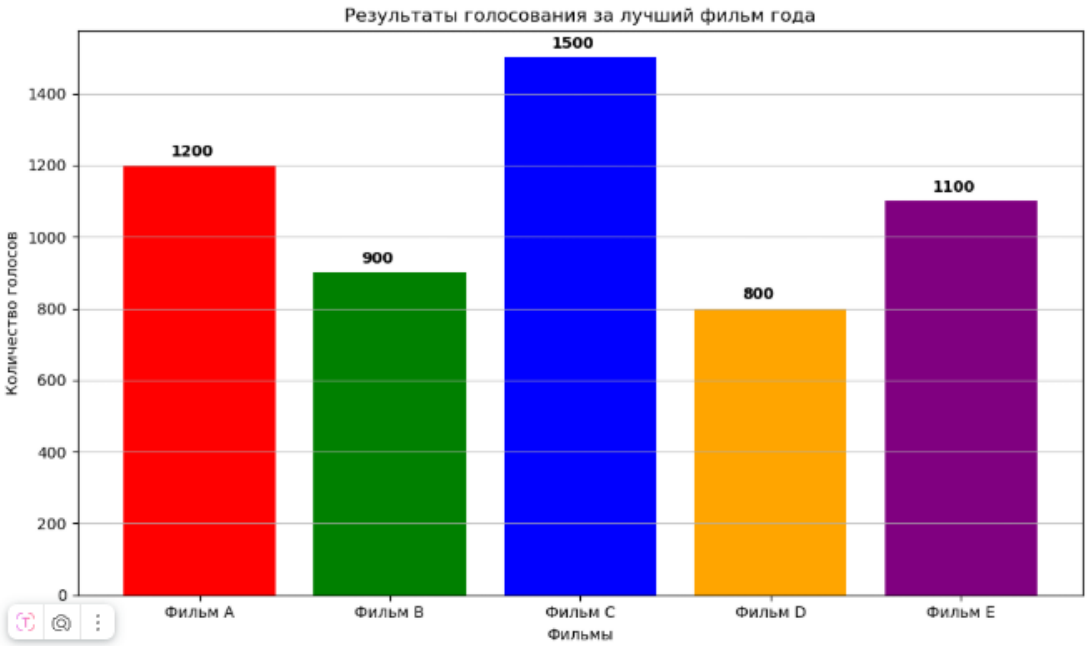

Рисунок 21 – График к ИЗ №2

7. Площадь под косинусом

Определите площадь под кривой функции $f(x) = cos(x)$ на отрезке $[0, \frac{\pi}{2}]$.

```
from scipy.integrate import quad


def visualize_integral_cosine():

    def cosine_function(x):

        return np.cos(x)

    lower_limit = 0
    upper_limit = np.pi / 2
    x = np.linspace(0, np.pi / 2, 100)
    y = cosine_function(x)

    area, error = quad(cosine_function, lower_limit, upper_limit)

    plt.figure(figsize=(8, 6))
    plt.plot(x, y, label='cos(x)')

    plt.fill_between(x, y, color='skyblue', alpha=0.4, label='Площадь под кривой')

    plt.xlabel("x")
    plt.ylabel("cos(x)")
    plt.title("Визуализация интеграла cos(x) от 0 до pi/2")

    plt.legend()
    plt.text(0.5, 0.2, f'Площадь: {area:.4f}', transform=plt.gca().transAxes, fontsize=12)

    plt.grid(True)

    plt.tight_layout()

    plt.show()


if __name__ == "__main__":
    visualize_integral_cosine()
```

Рисунок 22 – Код к ИЗ №3

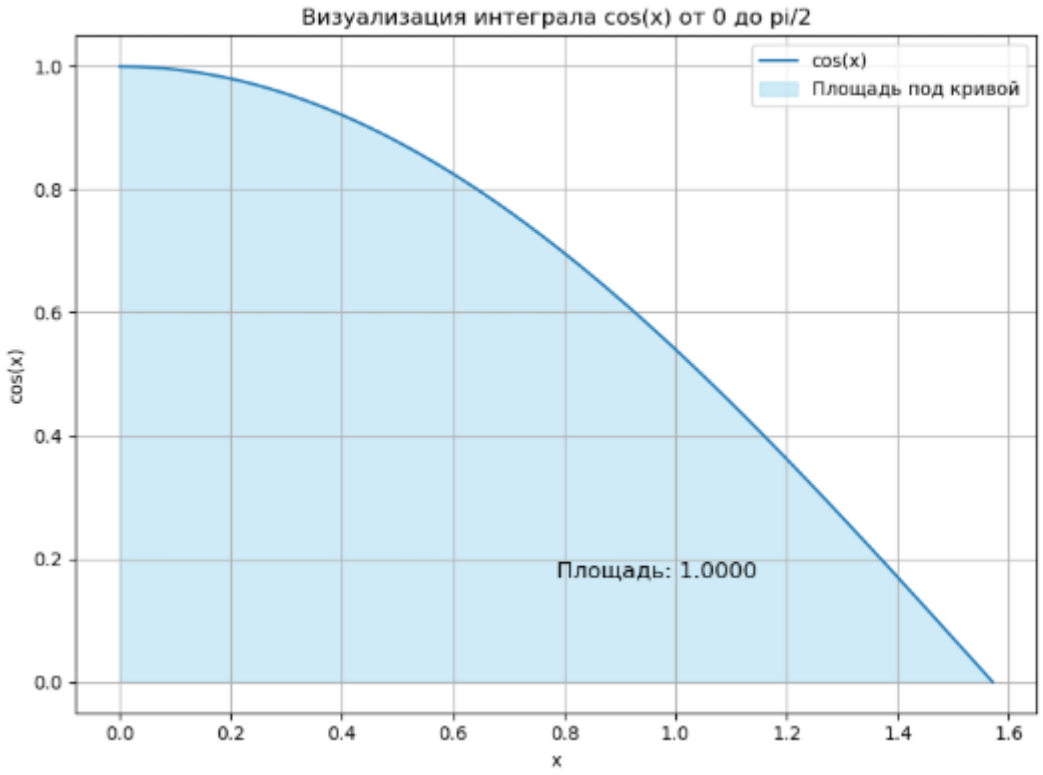

Рисунок 23 – График к ИЗ №3

7. Винтовая поверхность

Постройте 3D график спиральной поверхности, заданной параметрическими уравнениями:

- $x = r \cdot cos(\theta)$
- $y = r \cdot sin(\theta)$
- $z = \theta$

где $r \in [0, 2]$ и $\theta \in [0, 4\pi]$.

```
from mpl_toolkits.mplot3d import Axes3D


def plot_helical_surface():

    r = np.linspace(0, 2, 50)
    theta = np.linspace(0, 4 * np.pi, 50)
    r, theta = np.meshgrid(r, theta)

    x = r * np.cos(theta)
    y = r * np.sin(theta)
    z = theta

    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection='3d')

    ax.plot_surface(x, y, z, cmap='viridis')

    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.set_zlabel("Z")
    ax.set_title("График спиральной поверхности")

    ax.view_init(elev=30, azim=60)

    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    plot_helical_surface()
```

Рисунок 24 – Код к ИЗ №4

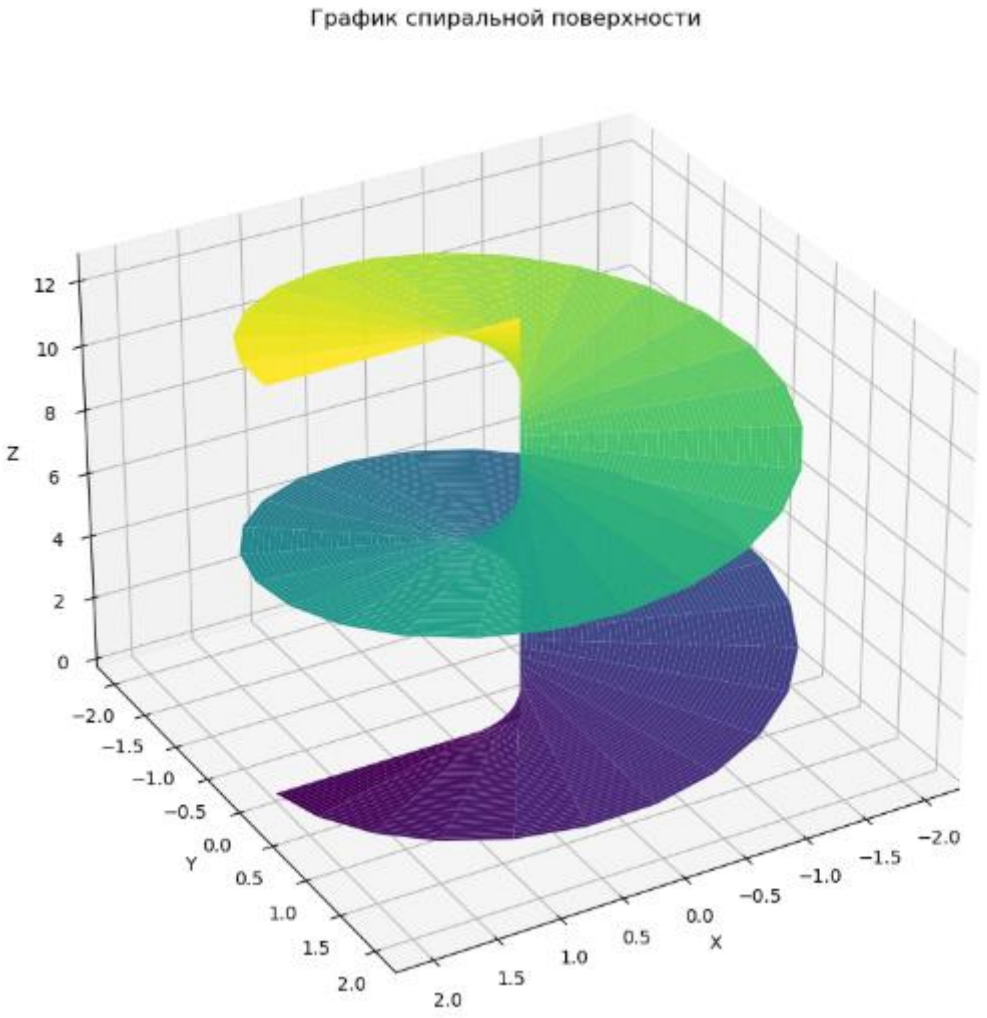

Рисунок 25 – График к ИЗ №4

Контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Установка пакета matplotlib осуществляется с помощью команды pip install matplotlib в командной строке или терминале.

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

Для корректного отображения графиков в ноутбуках Jupyter используется магическая команда %matplotlib inline.

3. Как отобразить график с помощью функции plot ?

Для отображения графика с помощью функции plot нужно использовать команду plt.plot(x, y) и затем plt.show().

4. Как отобразить несколько графиков на одном поле?

Чтобы отобразить несколько графиков на одном поле, можно использовать функцию plt.subplot(rows, cols, index) перед каждым графиком.

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

Для построения диаграмм категориальных данных используется метод plt.bar() для столбчатых диаграмм или plt.boxplot() для коробчатых диаграмм.

6. Какие основные элементы графика Вам известны?

Основные элементы графика включают оси (x, y), метки осей, легенду, заголовок, сетку и данные.

7. Как осуществляется управление текстовыми надписями на графике?

Для управления текстовыми надписями на графике используется метод plt.text(x, y, 'text') для добавления текста в указанные координаты.

8. Как осуществляется управление легендой графика?

Для управления легендой графика используется метод plt.legend().

9. Как задать цвет и стиль линий графика?

Цвет и стиль линий графика задаются с помощью параметров, таких как color, linestyle в функции plot(), например, plt.plot(x, y, color='r', linestyle='--').

10. Как выполнить размещение графика в разных полях?

Для размещения графиков в разных полях используется метод plt.subplot().

11. Как выполнить построение линейного графика с помощью matplotlib?

Линейный график строится с помощью функции plt.plot(x, y).

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

Заливка области между графиком и осью или между двумя графиками выполняется с помощью метода plt.fill_between(x, y1, y2).

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

Для выборочной заливки, которая удовлетворяет определенному условию, используется метод plt.fill_between(x, y, condition).

14. Как выполнить двухцветную заливку?

Двухцветную заливку можно сделать с помощью plt.fill_between(x, y1, y2, where=condition, color='color1', alpha=0.5).

15. Как выполнить маркировку графиков?

Для маркировки графиков используются функции plt.text() или plt.annotate() для добавления меток на график.

16. Как выполнить обрезку графиков?

Обрезка графиков осуществляется с помощью метода plt.xlim() и plt.ylim() для ограничения диапазонов осей.

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

Ступенчатый график строится с помощью функции plt.step(x, y), и его особенность в том, что линии соединяют данные ступенями.

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

Стековый график строится с помощью функции plt.stackplot(x, y), и его особенность в том, что области под графиком накладываются друг на друга.

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

Стем-график строится с помощью plt.stem(x, y), его особенность в том, что он отображает данные в виде вертикальных линий с маркерами на вершинах.

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

Точечный график строится с помощью функции plt.scatter(x, y), и его особенность в том, что отображает данные в виде точек.

21. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

Столбчатая диаграмма строится с помощью plt.bar(x, height).

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

Групповая столбчатая диаграмма строится с использованием нескольких plt.bar() с раздвигом на оси x. Столбчатая диаграмма с элементом errorbar добавляется через plt.errorbar().

23. Как выполнить построение круговой диаграммы средствами matplotlib?

Круговую диаграмму можно построить с помощью метода plt.pie(data)

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

Цветовая карта — это способ отображения данных через цвета. В matplotlib она используется с помощью функций типа plt.imshow() или plt.contourf(), где можно задавать различные схемы цветов.

25. Как отобразить изображение средствами matplotlib?

Изображение отображается с помощью plt.imshow(image).

26. Как отобразить тепловую карту средствами matplotlib?

Тепловая карта отображается с помощью plt.imshow(data, cmap='hot')

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

Для построения линейного 3D-графика используется ax.plot(x, y, z) с осью 3D.

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

Для построения точечного 3D-графика используется ax.scatter(x, y, z) с осью 3D.

29. Как выполнить построение каркасной поверхности с помощью matplotlib?

Каркасную поверхность можно построить с помощью ax.plot_wireframe(X, Y, Z).

30. Как выполнить построение трехмерной поверхности с помощью matplotlib?

Трехмерную поверхность можно построить с помощью ax.plot_surface(X, Y, Z) для 3D-графиков.

Вывод: В ходе практической работы, исследовал базовые возможности библиотеки matplotlib языка программирования Python.