

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 6

Выполнил:
Якушенко Антон Андреевич
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Проверил:
Ассистент департамента цифровых,
робототехнических систем и
электроники Хацукова А.И

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

ТЕМА: ОСНОВЫ РАБОТЫ С БИБЛЕОТЕКОЙ MATPLOTLIB

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

Ссылка на репозиторий: <https://github.com/Yakush766/LB3AI.git>

Порядок выполнения работы:

Задание 1: Построение простого графика

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

# 1. Определяем значения x
x = np.linspace(-10, 10, 400) # Создаем 400 точек между -10 и 10

# 2. Вычисляем значения y
y = x**2

# 3. Создаем график
plt.figure(figsize=(8, 6)) # Настраиваем размер графика для лучшей видимости

# Рисуем график по точкам x и y
plt.plot(x, y)

# 4. Добавляем заголовок и подписи осей
plt.title('График функции y = x^2')
plt.xlabel('x')
plt.ylabel('y')

# 5. Добавляем сетку
plt.grid(True)

# 6. Отображаем график
plt.show()
```

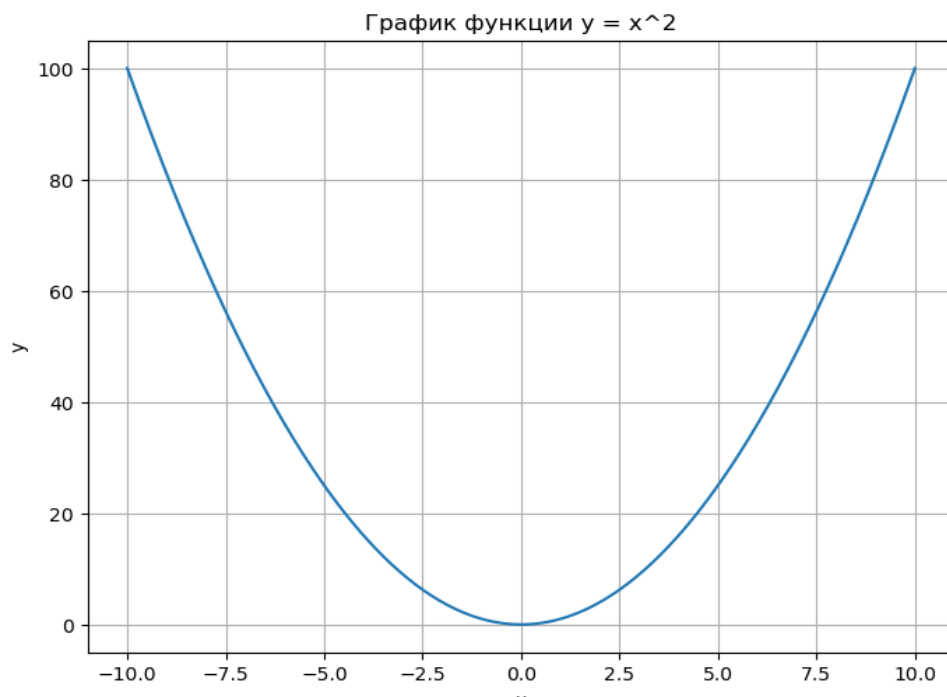


Рисунок 1. Выполненное задание 1

Задание 2: Настройка цветов и стилей

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

# 1. Определим значения x
x = np.linspace(-10, 10, 400)

# 2. Вычислим значения y для каждой функции
y1 = x # y = x
y2 = x**2 # y = x^2
y3 = x**3 # y = x^3

# 3. Создаем график
plt.figure(figsize=(10, 8)) # Настроим размер графика для лучшей видимости

# Рисуем линии с указанными цветами и стилями
plt.plot(x, y1, 'b:', label='y = x (синяя, пунктирная)') # синий, пунктир
plt.plot(x, y2, 'g-.', label='y = x^2 (зеленая, штрих-пунктирная)') # зеленый, штрих-пунктир
plt.plot(x, y3, 'r-', label='y = x^3 (красная, сплошная)') # красный, сплошной

# 4. Добавляем заголовок и подписи осей
plt.title('Графики функций y = x, y = x^2 и y = x^3')
plt.xlabel('x')
plt.ylabel('y')

# 5. Добавляем сетку
plt.grid(True)

# 6. Добавляем легенду
plt.legend()

# 7. Делаем оси одного масштаба
plt.axis('equal')

# 8. Устанавливаем пределы осей для лучшей визуализации (необязательно, но рекомендуется)
plt.xlim([-5, 5]) # Настроим пределы по оси x по необходимости
plt.ylim([-5, 5]) # Настроим пределы по оси y по необходимости, особенно для x^3

# 9. Отображаем график
plt.show()
```

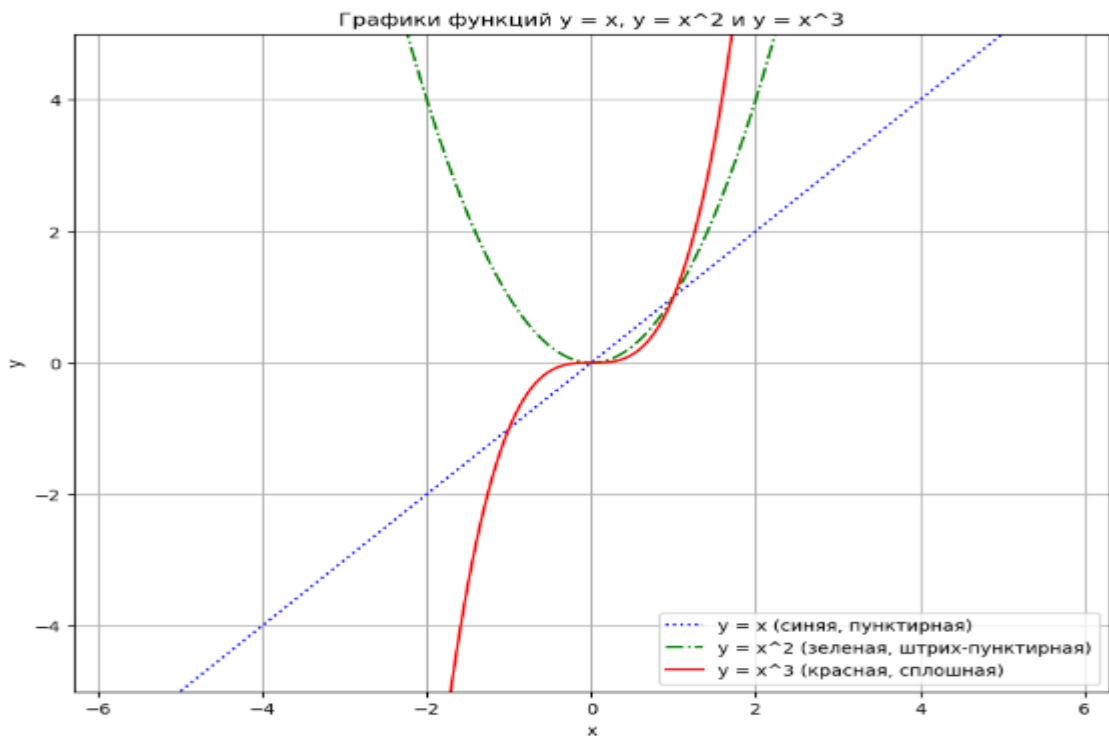


Рисунок 2. Выполненное задание 2

Задание 3: Использование различных типов графиков

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

# 1. Генерируем случайные данные
количество_точек = 50
x = np.random.rand(количество_точек) # Случайные x координаты между 0 и 1
y = np.random.rand(количество_точек) # Случайные y координаты между 0 и 1

# 2. Определяем цвета и размеры на основе x и y
цвета = x # Цвет на основе x-координаты
размеры = y * 100 # Размер на основе y-координаты, масштабируем для видимости

# 3. Создаем диаграмму рассеяния
plt.figure(figsize=(8, 6)) # Настраиваем размер графика

# s - размер точек, c - цвет точек, cmap - цветовая карта
plt.scatter(x, y, s=размеры, c=цвета, cmap='viridis', alpha=0.7)

# 4. Добавляем подписи осей и заголовок
plt.xlabel('Координата X')
plt.ylabel('Координата Y')
plt.title('Диаграмма Рассеяния: Цвет по X, Размер по Y')

# 5. Добавляем цветовую шкалу
plt.colorbar(label='Значение Координаты X')

# 6. Отображаем график
plt.show()
```

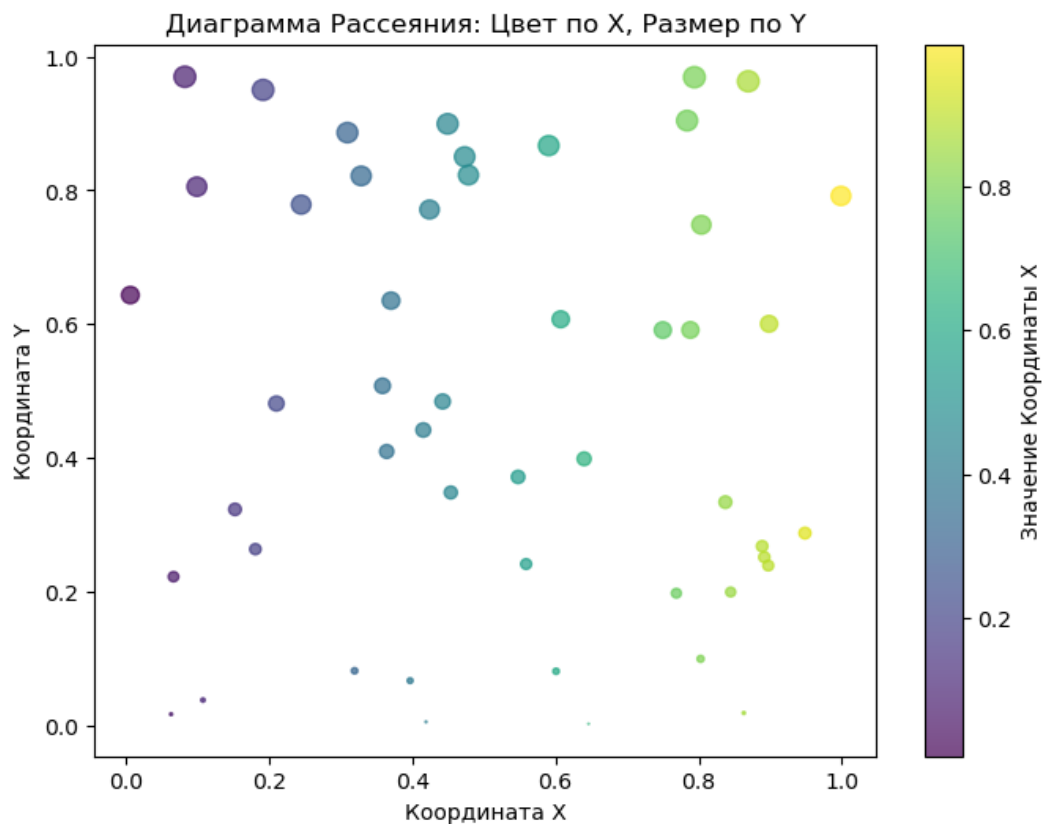


Рисунок 3. Выполненное задание 3

Задание 4: Гистограмма распределения

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

# 1. Генерируем 1000 случайных чисел из нормального распределения
количество_чисел = 1000
среднее_значение = 0 #  $\mu$ 
стандартное_отклонение = 1 #  $\sigma$ 

случайные_числа = np.random.normal(среднее_значение, стандартное_отклонение, количество_чисел)

# 2. Строим гистограмму
количество_корзин = 30
plt.figure(figsize=(10, 6)) # Настраиваем размер графика

# hist возвращает значения и границы корзин
n, bins, patches = plt.hist(случайные_числа, количество_корзин, facecolor='blue', alpha=0.75)

# 3. Добавляем вертикальную линию в среднем значении
plt.axvline(среднее_значение, color='red', linestyle='dashed', linewidth=2, label='Среднее Значение')

# 4. Добавляем подписи осей и заголовок
plt.xlabel('Значение')
plt.ylabel('Частота')
plt.title('Гистограмма Нормального Распределения')

# 5. Добавляем легенду
plt.legend()

# 6. Добавляем сетку (необязательно, но улучшает читаемость)
plt.grid(True)

# 7. Отображаем график
plt.show()
```

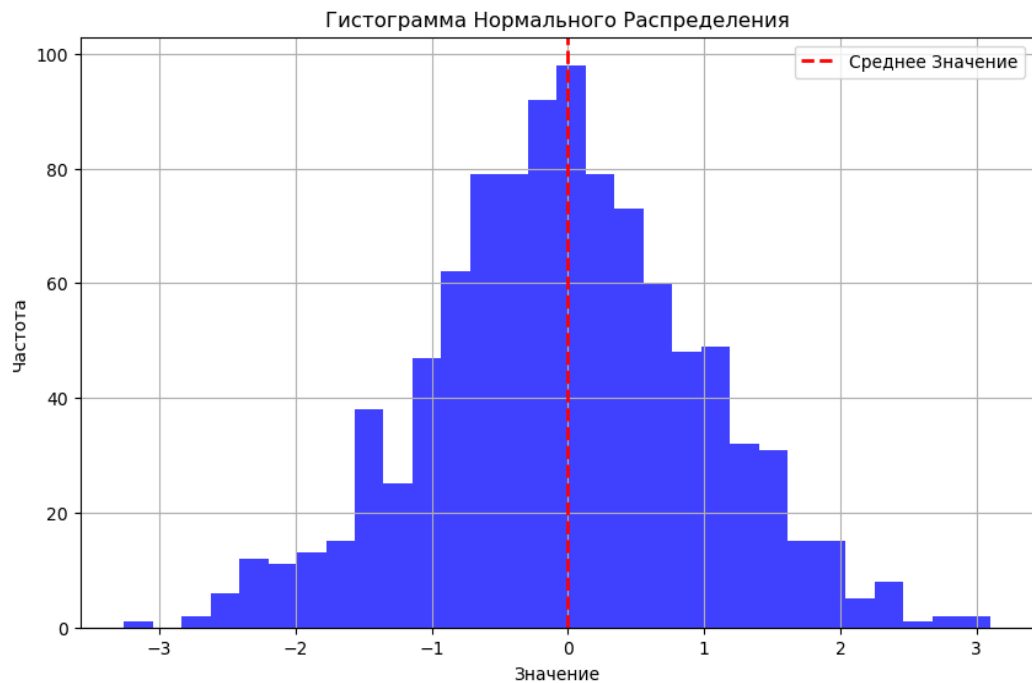


Рисунок 4. Выполненное задание 4

Задание 5: Столбчатая диаграмма

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

# 1. Данные для столбчатой диаграммы
оценки = ['Отлично', 'Хорошо', 'Удовлетворительно', 'Неудовлетворительно']
количество_студентов = [20, 35, 30, 15]

# 2. Создаем столбчатую диаграмму
plt.figure(figsize=(8, 6)) # Настраиваем размер графика

# Создаем столбцы, задаем положение по оси x
# np.arange(len(оценки)) - создаёт последовательность чисел [0, 1, 2, 3]
plt.bar(np.arange(len(оценки)), количество_студентов, color='skyblue')

# 3. Добавляем подписи осей и заголовок
plt.xlabel('Оценка')
plt.ylabel('Количество Студентов')
plt.title('Распределение Оценок Студентов')

# 4. Заменяем числовые подписи на оси x на названия оценок
# xticks - positions to put ticks
# labels - the labels for the ticks
plt.xticks(np.arange(len(оценки)), оценки) # Задаем подписи для оси x

# 5. Добавляем значения над столбцами (необязательно, но наглядно)
for i, значение in enumerate(количество_студентов):
    plt.text(i, значение + 0.5, str(значение), ha='center') # выводим значения над столбцами

# 6. Добавляем сетку (необязательно)
plt.grid(axis='y', alpha=0.5)

# 7. Отображаем график
plt.show()
```

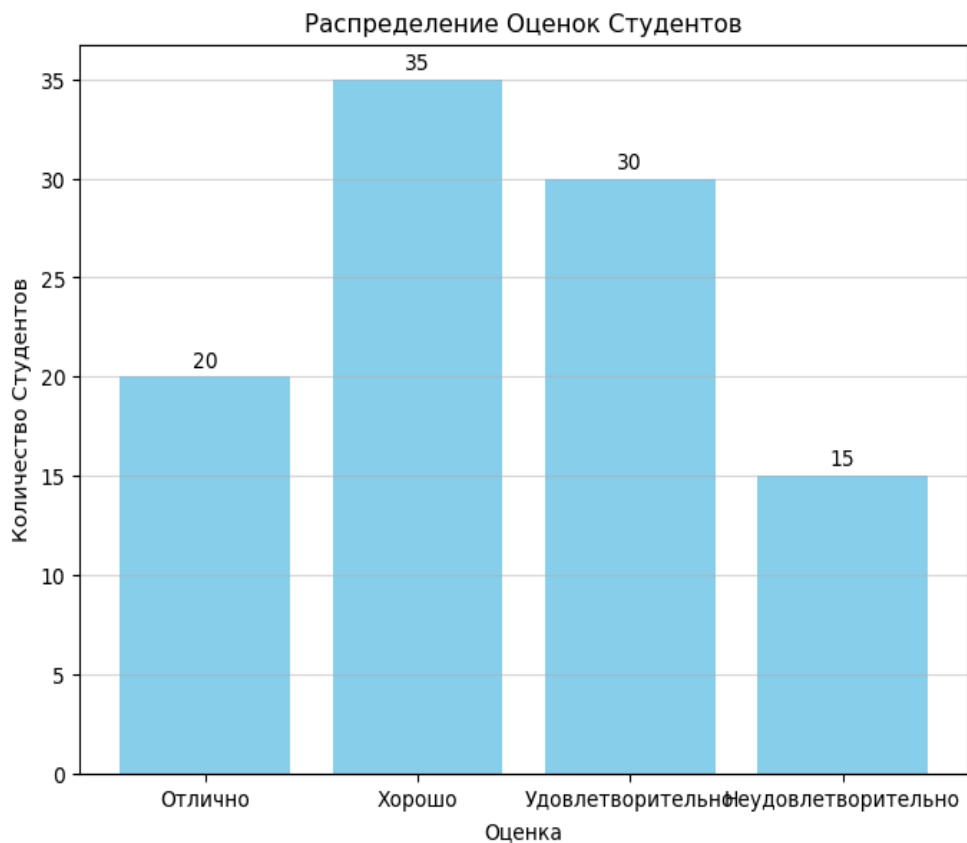


Рисунок 5. Выполненное задание 5

Задание 6: Круговая диаграмма

```
import matplotlib.pyplot as plt

# 1. Данные для круговой диаграммы (из предыдущей задачи)
оценки = ['Отлично', 'Хорошо', 'Удовлетворительно', 'Неудовлетворительно']
количество_студентов = [20, 35, 30, 15]

# 2. Создаем круговую диаграмму
plt.figure(figsize=(8, 6)) # Настраиваем размер графика

# autopct - формат отображения процентов, startangle - угол начала отсчета
plt.pie(количество_студентов, labels=оценки, autopct='%1.1f%%', startangle=90,
        wedgeprops={'linewidth': 1, 'edgecolor': 'white'}) # Добавляем разделительные линии

# 3. Добавляем заголовок
plt.title('Распределение Оценок Студентов (Круговая Диаграмма)')

# 4. Делаем круговую диаграмму кругом (а не эллипсом)
plt.axis('equal')

# 5. Отображаем график
plt.show()
```

Распределение Оценок Студентов (Круговая Диаграмма)

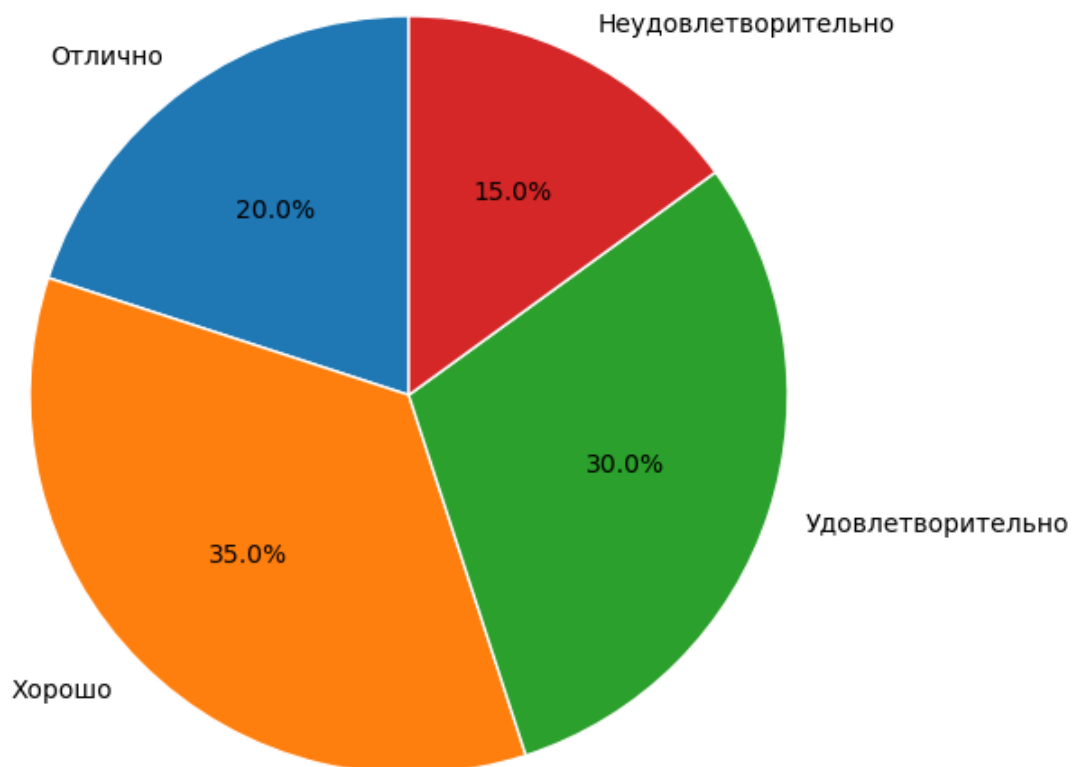


Рисунок 6. Выполненное задание 6

Задание 7: Трёхмерный график поверхности

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

# Создаем сетку значений x и y
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)

# Вычисляем значения z
z = np.sin(np.sqrt(x**2 + y**2))

# Создаем 3D график
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Рисуем поверхность
ax.plot_surface(x, y, z)

# Добавляем подписи к осям
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Добавляем заголовок
ax.set_title('3D График функции  $z = \sin(\sqrt{x^2 + y^2})$ ')

# Отображаем график
plt.show()
```

3D График функции $z = \sin(\sqrt{x^2 + y^2})$

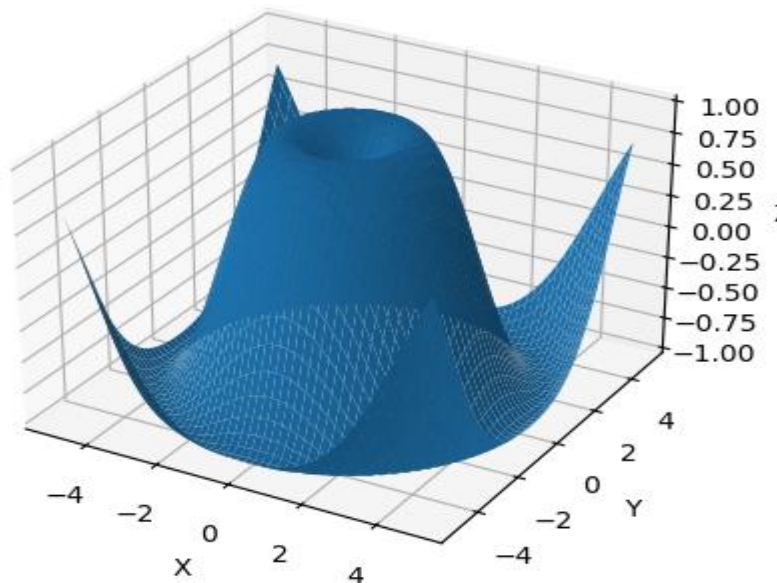


Рисунок 7. Выполненное задание 7

Задание 8: Множественные подграфики

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

# Данные для графиков
x = np.linspace(-5, 5, 100) # Создаем 100 точек между -5 и 5
y1 = x
y2 = x**2
y3 = np.sin(x)
y4 = np.cos(x)

# Создаем фигуру и подграфики
fig, axes = plt.subplots(2, 2, figsize=(10, 8)) # 2x2 сетка, размер фигуры

# График 1: y = x (линейный график)
axes[0, 0].plot(x, y1)
axes[0, 0].set_title('Линейный график: y = x')
axes[0, 0].set_xlabel('x')
axes[0, 0].set_ylabel('y')

# График 2: y = x^2 (парабола)
axes[0, 1].plot(x, y2)
axes[0, 1].set_title('Парабола: y = x^2')
axes[0, 1].set_xlabel('x')
axes[0, 1].set_ylabel('y')

# График 3: y = sin(x) (синус)
axes[1, 0].plot(x, y3)
axes[1, 0].set_title('Синус: y = sin(x)')
axes[1, 0].set_xlabel('x')
axes[1, 0].set_ylabel('y')

# График 4: y = cos(x) (косинус)
axes[1, 1].plot(x, y4)
axes[1, 1].set_title('Косинус: y = cos(x)')
axes[1, 1].set_xlabel('x')
axes[1, 1].set_ylabel('y')

# Добавляем общее название для всей фигуры (необязательно)
fig.suptitle('Четыре графика в одной фигуре', fontsize=16)

# Автоматическая настройка расположения подграфиков, чтобы избежать перекрытий
plt.tight_layout()

# Отображаем график
plt.show()
```

Четыре графика в одной фигуре

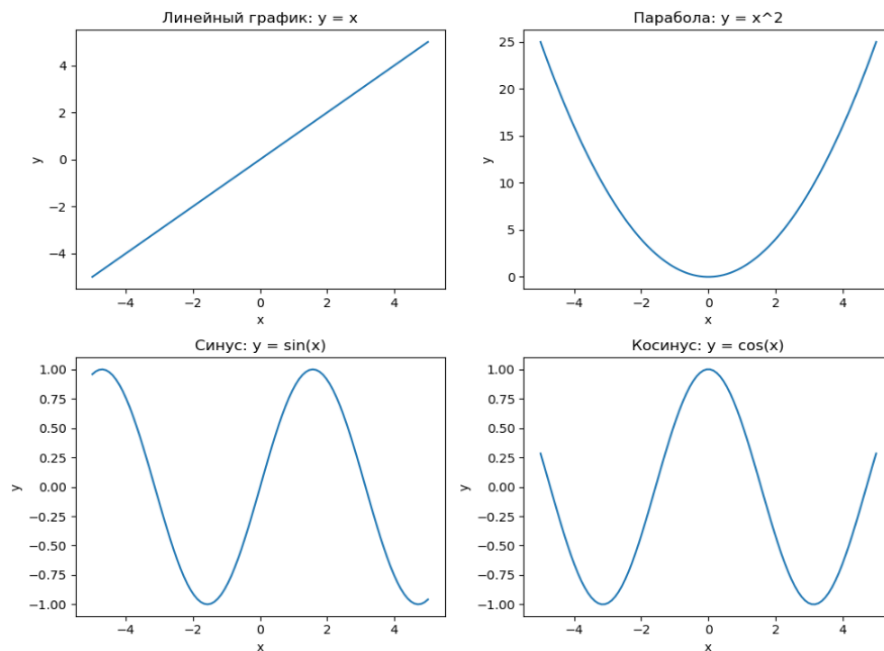


Рисунок 8. Выполненное задание 8

Задание 9: Тепловая карта

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

# Создаем случайную матрицу 10x10 с элементами от 0 до 1
matrix = np.random.rand(10, 10)

# Визуализируем матрицу как тепловую карту
plt.imshow(matrix, cmap='viridis', interpolation='nearest')

# Добавляем цветовую шкалу (colorbar)
plt.colorbar()

# Убираем отметки на осях (необязательно)
plt.xticks([])
plt.yticks([])

# Добавляем заголовок (необязательно)
plt.title('Тепловая карта случайной матрицы 10x10')

# Отображаем график
plt.show()
```

Тепловая карта случайной матрицы 10x10

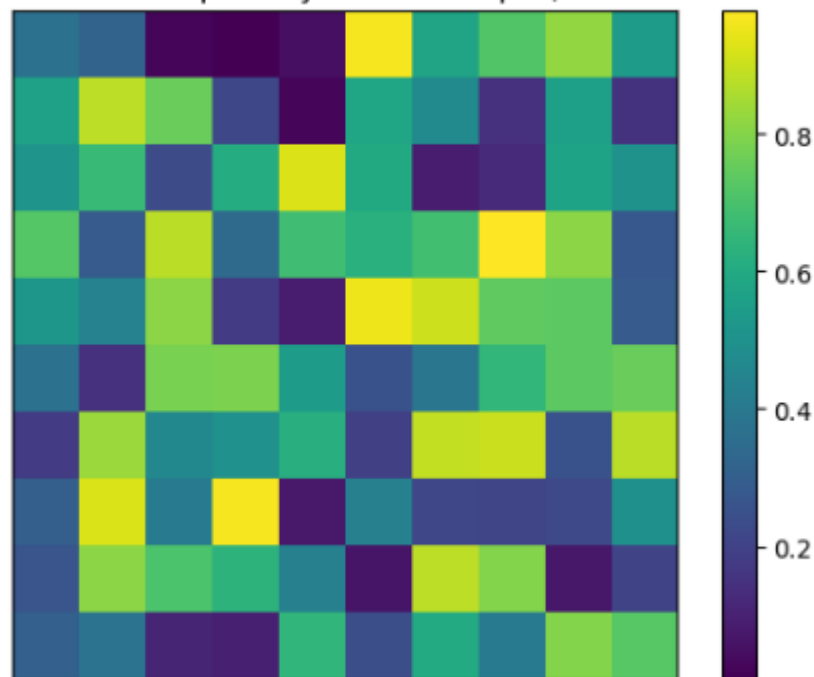


Рисунок 9. Выполненное задание 9

Выполнение индивидуального задания:

Задание 1: Среднесуточное потребление электроэнергии

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

# Данные
время = np.array([0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24])
потребление = np.array([0.8, 0.7, 0.6, 0.5, 0.9, 1.2, 1.5, 1.6, 1.7, 1.8, 1.6, 1.2, 1.0])

# Строим график
plt.figure(figsize=(10, 6)) # Размер графика (ширина, высота)
plt.plot(время, потребление, marker='o', linestyle='-', color='blue') # Линия графика

# Затеняем область под кривой
plt.fill_between(время, потребление, alpha=0.3, color='skyblue') # alpha - прозрачность

# Добавляем подписи к осям и заголовок
plt.xlabel('Время суток (часы)')
plt.ylabel('Потребление (кВт·ч)')
plt.title('Среднесуточное потребление электроэнергии')

# Добавляем сетку для лучшей читаемости (необязательно)
plt.grid(True)

# Отображаем график
plt.show()
```



Рисунок 10. Выполненное индивидуальное задание

Задание 2: Вклад разных отраслей в экономику страны

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

# Данные
отрасли = ['IT', 'Производство', 'Туризм', 'Финансы', 'Транспорт']
доля_ввп = np.array([15, 25, 10, 30, 20])

# Рассчитываем среднее значение доли
среднее_значение = np.mean(доля_ввп)

# Создаем столбчатую диаграмму
plt.figure(figsize=(10, 6)) # Размер графика (ширина, высота)
plt.bar(отрасли, доля_ввп, color='skyblue') # Столбцы

# Добавляем горизонтальную линию
plt.axhline(среднее_значение, color='red', linestyle='--', label=f'Среднее: {среднее_значение:.2f}%') # axhline - горизонтальная линия

# Добавляем подписи к осям и заголовок
plt.xlabel('Отрасль')
plt.ylabel('Доля в ВВП (%)')
plt.title('Вклад разных отраслей в экономику страны')

# Добавляем легенду для горизонтальной линии
plt.legend()

# Добавляем сетку для лучшей читаемости (необязательно)
plt.grid(axis='y', alpha=0.5)

# Отображаем график
plt.tight_layout() # Чтобы метки не обрезались
plt.show()
```

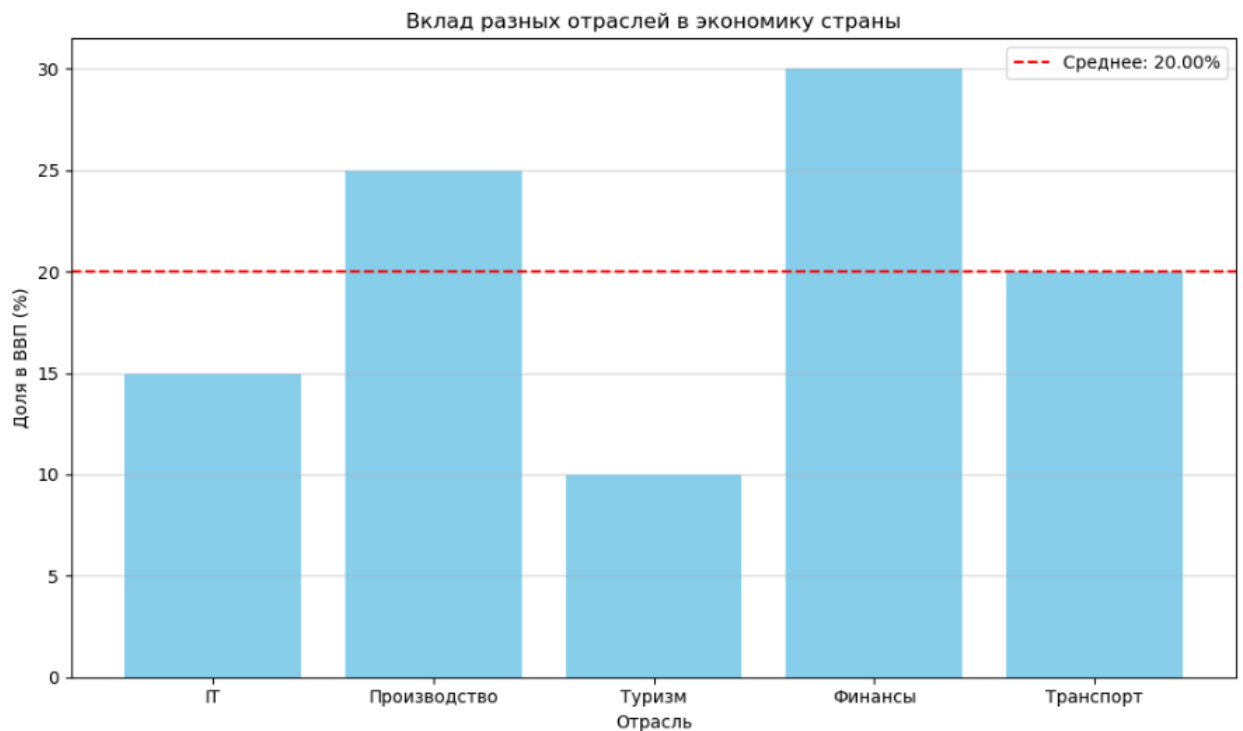


Рисунок 11. Выполненное индивидуальное задание

Задание 3: Площадь под кубической параболой

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import integrate

# Функция
def f(x):
    return x**3 - 2*x**2 + x

# Интервал интегрирования
a = -1
b = 2

# Вычисляем определенный интеграл
result = integrate.quad(f, a, b)
площадь = result[0]

# Создаем значения x для построения графика
x = np.linspace(a-1, b+1, 400) # Расширяем диапазон, чтобы график не обрезался по краям
y = f(x)

# Строим график
fig, ax = plt.subplots(figsize=(10, 6)) # Создаем фигуру и оси
ax.plot(x, y, 'r', linewidth=2) # График функции

# Закрашиваем область под кривой
x_fill = np.linspace(a, b, 400)
y_fill = f(x_fill)
ax.fill_between(x_fill, y_fill, color='skyblue', alpha=0.5) # Заливка области

# Добавляем горизонтальную линию y=0 и вертикальные линии для границ интегрирования
ax.axhline(0, color='black', linewidth=0.5)
ax.axvline(a, color='gray', linestyle='--', label=f'x = {a}')
ax.axvline(b, color='gray', linestyle='--', label=f'x = {b}')

# Добавляем подписи к осям и заголовок
ax.set_xlabel('x')
ax.set_ylabel('f(x)')
ax.set_title(f'График функции f(x) = x^3 - 2x^2 + x\nПлощадь под кривой от {a} до {b}: {площадь:.2f}')

# Добавляем легенду
ax.legend()

# Отображаем график
plt.grid(True) # Добавляем сетку
plt.show()
```

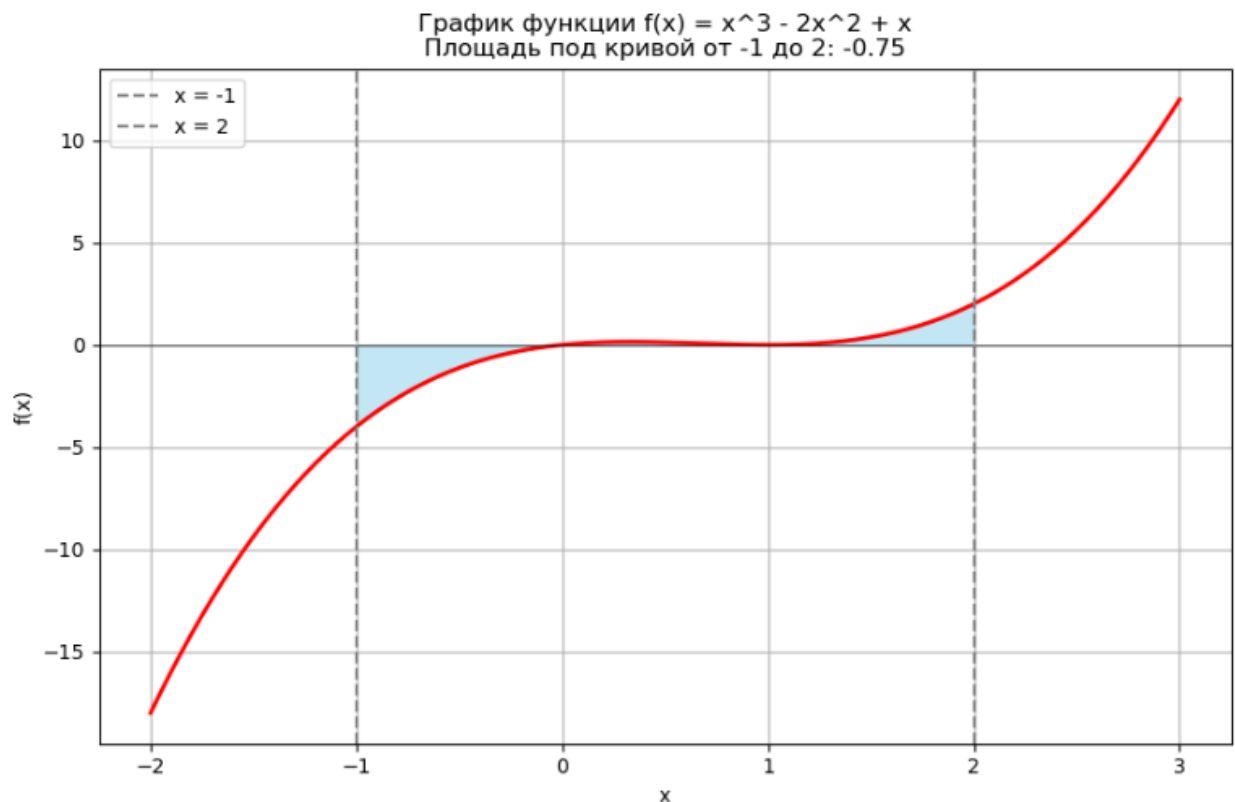


Рисунок 12. Выполненное индивидуальное задание

Задание 4: Гиперболическая поверхность

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Определяем функцию
def f(x, y):
    return 1 / (1 + x**2 + y**2)

# Задаем пределы для x и y
x_min, x_max = -3, 3
y_min, y_max = -3, 3

# Создаем сетку значений x и y
x = np.linspace(x_min, x_max, 100)
y = np.linspace(y_min, y_max, 100)
x, y = np.meshgrid(x, y)

# Вычисляем значения z
z = f(x, y)

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

# Рисуем поверхность
surf = ax.plot_surface(x, y, z, cmap='viridis', edgecolor='none') # cmap - цветовая карта

# Добавляем цветовую карту (colorbar)
fig.colorbar(surf, shrink=0.5, aspect=5)

# Добавляем подписи к осям
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('f(x, y)')

# Задаем пределы осей
ax.set_xlim(x_min, x_max)
ax.set_ylim(y_min, y_max)

# Добавляем заголовок
ax.set_title('Гиперболическая поверхность  $f(x, y) = 1 / (1 + x^2 + y^2)$ ')

# Отображаем график
plt.show()
```

Гиперболическая поверхность $f(x, y) = 1 / (1 + x^2 + y^2)$

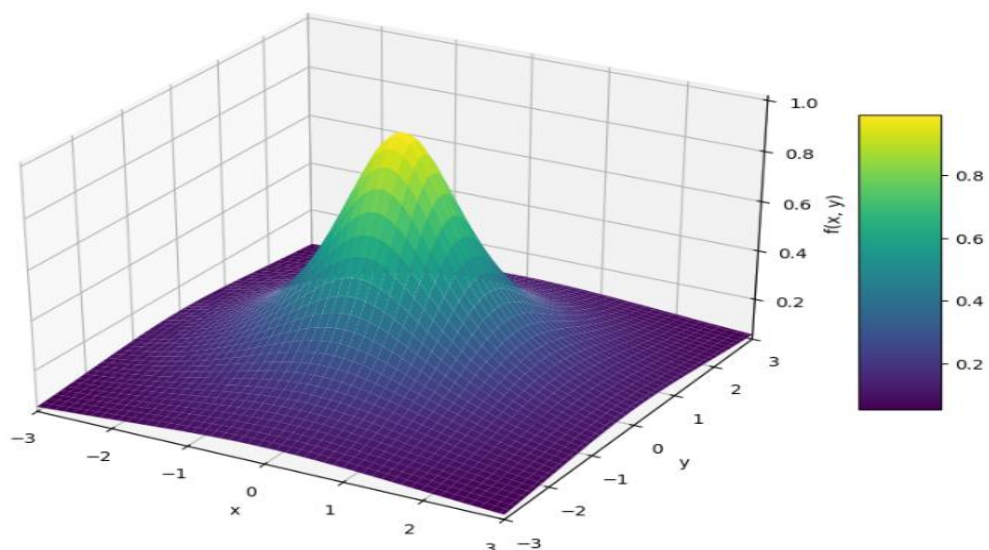


Рисунок 13. Выполненное индивидуальное задание

Вывод: в ходе этой лабораторной работы были исследованы базовые возможности библиотеки `matplotlib` языка программирования Python. Были построены самые различные графики, а также разное оформление.

Ответы на контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

`pip install matplotlib`

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

`%matplotlib inline` или `%matplotlib notebook`

3. Как отобразить график с помощью функции plot ?

Отображение графика с plot: `plt.plot(x, y); plt.show()`

4. Как отобразить несколько графиков на одном поле?

Несколько графиков на одном поле: `plt.plot(x1, y1); plt.plot(x2, y2); plt.show()`

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

Диаграммы категориальных данных: Столбчатые диаграммы (`plt.bar`) или круговые диаграммы (`plt.pie`)

6. Какие основные элементы графика Вам известны?

Основные элементы графика: Заголовок, оси (с подписями), данные (линии, точки, столбцы и т.д.), легенда, сетка.

7. Как осуществляется управление текстовыми надписями на графике?

Управление текстовыми надписями: `plt.title()`, `plt.xlabel()`, `plt.ylabel()`, `plt.text()`

8. Как осуществляется управление легендой графика?

Управление легендой: `plt.legend()`

9. Как задать цвет и стиль линий графика?

Цвет и стиль линий: `plt.plot(x, y, color='red', linestyle='--')`

10. Как выполнить размещение графика в разных полях?

Размещение графика в разных полях: `plt.subplot()`

11. Как выполнить построение линейного графика с помощью matplotlib?

Линейный график: `plt.plot(x, y)`

12. Как выполнить заливку области между графиком и осью?

Между двумя графиками?

Заливка области: `plt.fill_between()`, `plt.fill_between(x, y1, y2)`

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

Выборочная заливка: `plt.fill_between(x, y, where=(y > threshold))`

14. Как выполнить двухцветную заливку?

Двухцветная заливка: Использовать несколько вызовов `plt.fill_between` с разными условиями и цветами.

15. Как выполнить маркировку графиков?

Маркировка графиков: Параметр `marker` в `plt.plot()` (например, `plt.plot(x, y, marker='o')`)

16. Как выполнить обрезку графиков?

Обрезка графиков: `plt.xlim()`, `plt.ylim()`

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

Ступенчатый график: `plt.step()`. Особенность: Горизонтальные и вертикальные линии, соединяющие точки.

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

Стековый график: `plt.stackplot()`. Особенность: Показывает вклад каждой серии данных в общую сумму.

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

Stem-график: `plt.stem()`. Особенность: Вертикальные линии от оси до значений, часто с маркером наверху.

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

Точечный график: `plt.scatter()`. Особенность: Отображает отдельные точки без линий. Размер и цвет точек можно менять в зависимости от данных.

21. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

Столбчатые диаграммы: `plt.bar()`

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

Групповая столбчатая диаграмма: Несколько столбцов рядом друг с другом для разных категорий. Столбчатая диаграмма с `errorbar`: Столбец с добавленными "усами погрешности" для отображения разброса данных.

23. Как выполнить построение круговой диаграммы средствами matplotlib?

Круговая диаграмма: `plt.pie()`

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

Цветовая карта: Сопоставление значений с цветами. Работа: Используется с `cmap` параметром в функциях, таких как `imshow` и `scatter`. Можно выбрать из предустановленных ('viridis', 'jet', 'gray' и т.д.) или создать свою.

25. Как отобразить изображение средствами matplotlib?

Отображение изображения: `plt.imshow()`

26. Как отобразить тепловую карту средствами matplotlib?

Тепловая карта: Отображение матрицы данных с использованием цветовой карты. Используется `plt.imshow()` или `plt.pcolormesh()`

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

Линейный 3D-график: `ax.plot(x, y, z)` (где `ax` - объект `Axes3D`)

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

Точечный 3D-график: `ax.scatter(x, y, z)`

29. Как выполнить построение каркасной поверхности с помощью matplotlib?

Каркасная поверхность: `ax.plot_wireframe(X, Y, Z)`

30. Как выполнить построение трехмерной поверхности с помощью matplotlib?

Трехмерная поверхность: `ax.plot_surface(X, Y, Z)`