

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 3

Выполнил:
Борцов Богдан Михайлович
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:
Доцент департамента цифровых,
робототехнических систем и
электроники 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.

Ссылка на репозиторий: <https://github.com/REPONCFU/ai-jlab3>

Порядок выполнения работы:

1. Выполнение задания №1.

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

x = np.arange(-10, 10.1, 0.1)

y = x ** 2

plt.figure(figsize=(8, 6))
plt.plot(x, y, 'b-', linewidth=2)

plt.title('График функции  $y = x^2$ ', fontsize=14)
plt.xlabel('x', fontsize=12)
plt.ylabel('y =  $x^2$ ', fontsize=12)

plt.grid(True, linestyle='--', alpha=0.7)

plt.show()
```

Рисунок 1. Задание №1

2. Выполнение задания №2.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-2, 2, 200)
y1 = x      # y = x
y2 = x ** 2 # y = x^2
y3 = x ** 3 # y = x^3

plt.figure(figsize=(8, 6))

plt.plot(x, y1, 'b--', label='y = x', linewidth=2)
plt.plot(x, y2, 'g-.', label='y = x^2', linewidth=2)
plt.plot(x, y3, 'r-', label='y = x^3', linewidth=2)

plt.legend(fontsize=12)

plt.axis('equal')

plt.title('Графики функций:  $y = x$ ,  $y = x^2$ ,  $y = x^3$ ', fontsize=14)
plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)

plt.grid(True, linestyle=':', alpha=0.7)

plt.show()
```

Рисунок 2. Задание №2

3. Выполнение задания №3.

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

x = np.random.rand(50)
y = np.random.rand(50)

colors = x
sizes = 100 + 200 * y

plt.figure(figsize=(10, 6))
scatter = plt.scatter(x, y, c=colors, s=sizes, alpha=0.7, cmap='viridis')

plt.colorbar(scatter, label='Координата X (цвет)')

plt.xlabel('Координата X')
plt.ylabel('Координата Y')
plt.title('Диаграмма рассеяния: цвет от X, размер от Y')

plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Рисунок 3. Задание №3

4. Выполнение задания №4.

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

np.random.seed(42)
data = np.random.normal(loc=0, scale=1, size=1000)

plt.figure(figsize=(10, 6))
plt.hist(data, bins=30, color='skyblue', edgecolor='black', alpha=0.7, density=True)

mean = np.mean(data)
plt.axvline(mean, color='red', linestyle='--', linewidth=2, label=f'Среднее = {mean:.2f}')

plt.xlabel('Значение')
plt.ylabel('Плотность вероятности')
plt.title('Гистограмма 1000 случайных чисел из N(0, 1)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Рисунок 4. Задание №4

5. Выполнение задания №5.

```

import matplotlib.pyplot as plt

grades = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
students = [20, 35, 30, 15]

plt.figure(figsize=(10, 6))
bars = plt.bar(grades, students, color=['#4CAF50', '#8BC34A', '#FFC107', '#F44336'])

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height,
             f'{int(height)}',
             ha='center', va='bottom')

plt.title('Распределение студентов по оценкам', fontsize=14, pad=20)
plt.xlabel('Оценки', fontsize=12)
plt.ylabel('Количество студентов', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

```

Рисунок 5. Задание №5

6. Выполнение задания №6.

```

import matplotlib.pyplot as plt

grades = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
students = [20, 35, 30, 15]
colors = ['#4CAF50', '#8BC34A', '#FFC107', '#F44336']

plt.figure(figsize=(8, 8))
plt.pie(students, labels=grades, colors=colors, autopct='%1.1f%%',
        startangle=90, wedgeprops={'linewidth': 1, 'edgecolor': 'white'})

plt.title('Распределение студентов по оценкам (в процентах)', pad=20)

plt.show()

```

Рисунок 6. Задание №6

7. Выполнение задания №7.

```

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

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)

R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

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

surf = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')

ax.set_title('3D-график функции  $z = \sin(\sqrt{x^2 + y^2})$ ', pad=20)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

fig.colorbar(surf, shrink=0.5, aspect=5)

plt.tight_layout()
plt.show()

```

Рисунок 7. Задание №7

8. Выполнение задания №8.

```

import numpy as np
import matplotlib.pyplot as plt
|
x = np.linspace(-np.pi, np.pi, 100)

fig, axs = plt.subplots(2, 2, figsize=(12, 10))

axs[0, 0].plot(x, x, color='blue', linewidth=2)
axs[0, 0].set_title('Линейная функция:  $y = x$ ')
axs[0, 0].grid(True)

axs[0, 1].plot(x, x**2, color='red', linewidth=2)
axs[0, 1].set_title('Квадратичная функция:  $y = x^2$ ')
axs[0, 1].grid(True)

axs[1, 0].plot(x, np.sin(x), color='green', linewidth=2)
axs[1, 0].set_title('Тригонометрическая функция:  $y = \sin(x)$ ')
axs[1, 0].grid(True)

axs[1, 1].plot(x, np.cos(x), color='purple', linewidth=2)
axs[1, 1].set_title('Тригонометрическая функция:  $y = \cos(x)$ ')
axs[1, 1].grid(True)

plt.suptitle('Четыре базовых графика математических функций', y=1.02, fontsize=14)

plt.tight_layout()
plt.show()

```

Рисунок 8. Задание №8

9. Выполнение задания №9.

```

import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
matrix = np.random.rand(10, 10)

plt.figure(figsize=(8, 6))
plt.imshow(matrix, cmap='viridis', interpolation='nearest')

cbar = plt.colorbar()
cbar.set_label('Значения')

plt.xticks(np.arange(10))
plt.yticks(np.arange(10))
plt.xlabel('Столбцы')
plt.ylabel('Строки')

plt.title('Тепловая карта случайной матрицы 10×10', pad=20)

for i in range(10):
    for j in range(10):
        plt.text(j, i, f'{matrix[i, j]:.2f}',
                 ha='center', va='center',
                 color='white' if matrix[i, j] > 0.5 else 'black')

plt.show()

```

Рисунок 9. Задание №9

10. Выполнение индивидуального задания №1. Задачи на построение линейных графиков в Matplotlib

3. Динамика курса валюты

Постройте график изменения курса доллара к рублю за 12 месяцев. Данные: -Месяцы: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] -Курс (₽ за 1\$): [74, 76, 78, 79, 81, 79, 77, 75, 74, 73, 72, 71]

Используйте красную линию и сделайте подписи к точкам данных.

```

2]: import matplotlib.pyplot as plt

# Данные
months = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
exchange_rate = [74, 76, 78, 79, 81, 79, 77, 75, 74, 73, 72, 71]

# Создание графика
plt.figure(figsize=(10, 6))
plt.plot(months, exchange_rate, color='red', marker='o', linestyle='--', linewidth=2, markersize=8)

# Подписи точек данных
for i, (month, rate) in enumerate(zip(months, exchange_rate)):
    plt.text(month, rate + 0.5, f'{rate}', ha='center', va='bottom', fontsize=10)

# Настройка осей и заголовка
plt.title('Динамика курса доллара к рублю за 12 месяцев', fontsize=14)
plt.xlabel('Месяцы', fontsize=12)
plt.ylabel('Курс (₽ за 1$)', fontsize=12)
plt.xticks(months)
plt.grid(True, linestyle='--', alpha=0.7)

# Отображение графика
plt.tight_layout()
plt.show()

```

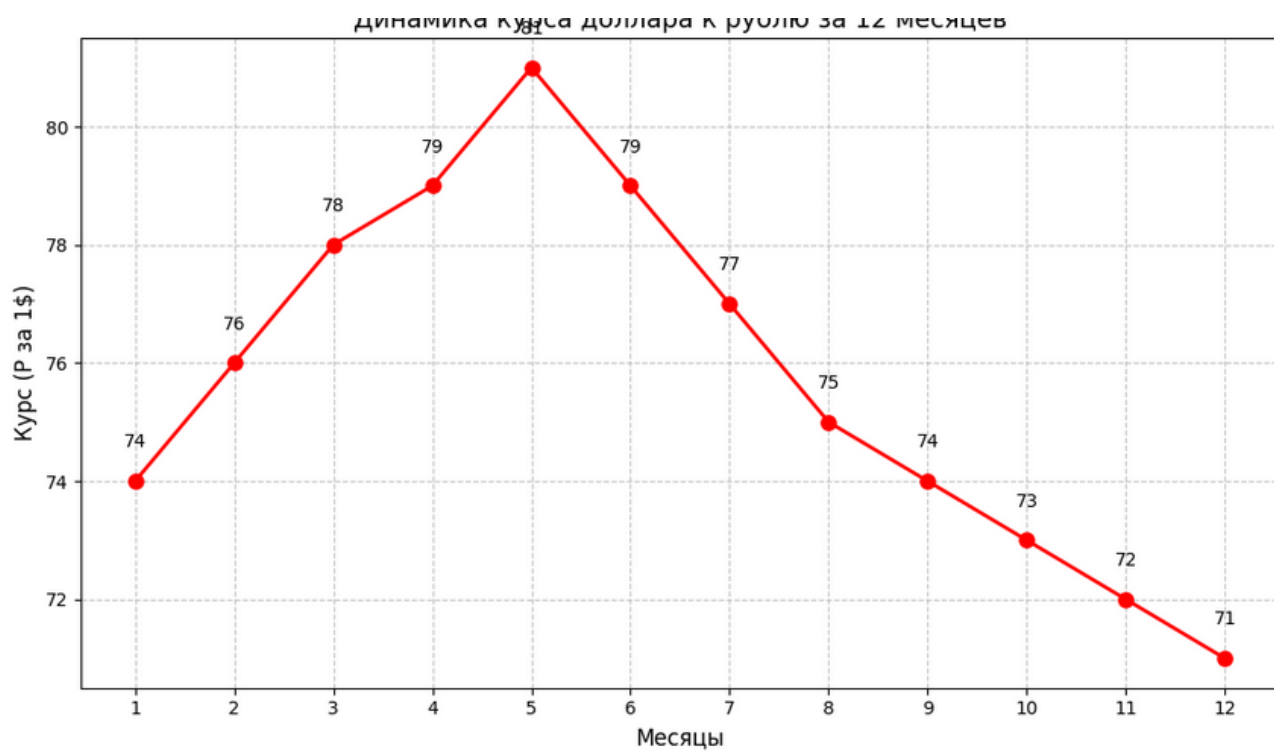


Рисунок 10. Индивидуальное задание №1

11. Выполнение индивидуального задания №2.

Задачи на построение столбчатых диаграмм в Matplotlib

3. Количество посетителей в магазине по дням недели

Фиксировалось количество посетителей за неделю:

- Дни недели: ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс']
- Посетители: [150, 180, 200, 220, 300, 450, 500]

Используйте разный цвет для будних и выходных дней.

```
import matplotlib.pyplot as plt

# Данные
days = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс']
visitors = [150, 180, 200, 220, 300, 450, 500]
colors = ['blue', 'blue', 'blue', 'blue', 'blue', 'green', 'green'] # Будни - синий, выходные - зеленый

# Создание столбчатой диаграммы
plt.figure(figsize=(10, 6))
bars = plt.bar(days, visitors, color=colors)

# Подписи значений над столбцами
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height}',
             ha='center', va='bottom', fontsize=10)

# Настройка осей и заголовка
plt.title('Количество посетителей в магазине по дням недели', fontsize=14)
plt.xlabel('Дни недели', fontsize=12)
plt.ylabel('Количество посетителей', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7, axis='y')

# Отображение графика
plt.tight_layout()
plt.show()
```

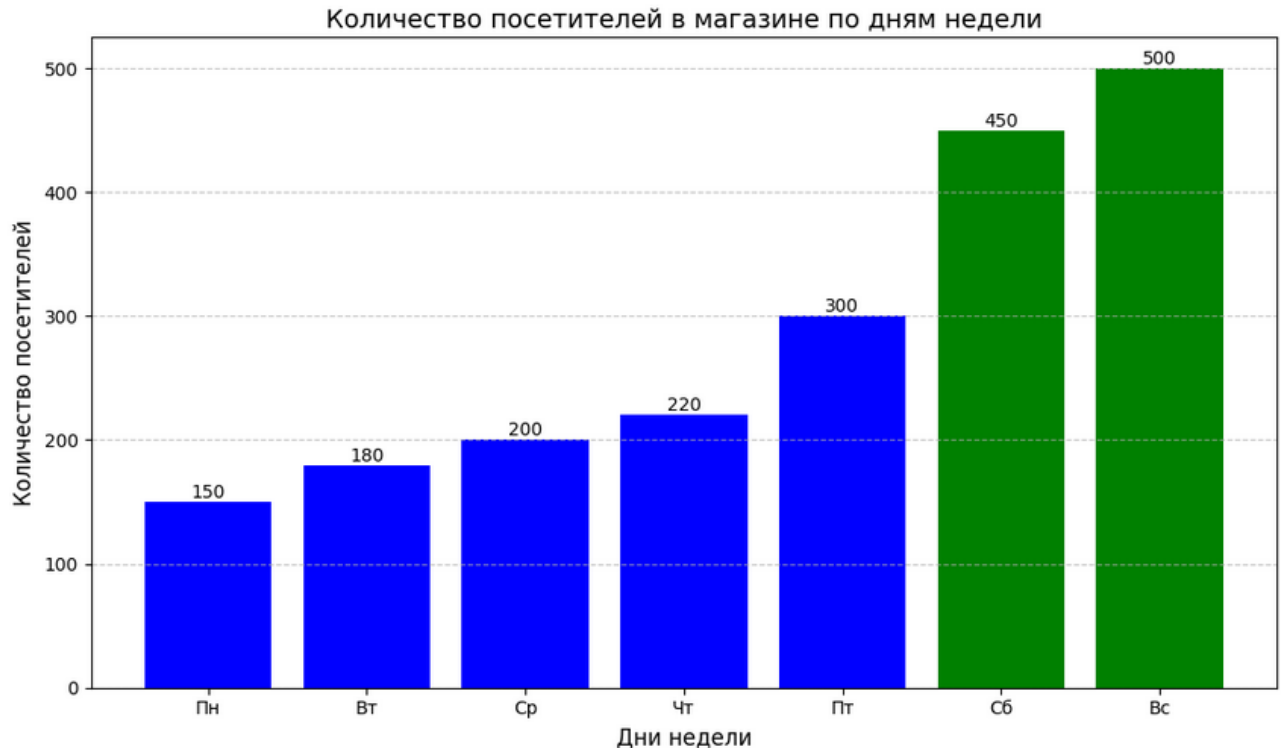


Рисунок 11. Индивидуальное задание №2

12. Выполнение индивидуального задания №3.

3. Площадь под экспонентой

Определите площадь под функцией:

$$f(x) = e^{-x}$$

на интервале $[0, 2]$. Постройте график и закрасьте область.

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

# Определение функции
def f(x):
    return np.exp(-x)

# Интервал
a, b = 0, 2

# Создание данных для графика
x = np.linspace(a, b, 100)
y = f(x)

# Вычисление интеграла методом прямоугольников (аппроксимация)
dx = 0.001 # Шаг для численного интегрирования
x_integral = np.arange(a, b + dx, dx)
y_integral = f(x_integral)
integral_value = np.sum(y_integral * dx) # Сумма площадей прямоугольников

# Построение графика
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'r-', linewidth=2, label=f'f(x) = e^{-x}')
plt.fill_between(x, y, color='lightblue', alpha=0.5,
                label=f'Площадь ≈ {integral_value:.4f} (численный метод)')

# Настройка графика
plt.title('Площадь под кривой f(x) = e^{-x} на интервале [0, 2]', fontsize=14)
plt.xlabel('x', fontsize=12)
plt.ylabel('f(x)', fontsize=12)
plt.legend(fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7)

# Отображение графика
plt.tight_layout()
plt.show()
```

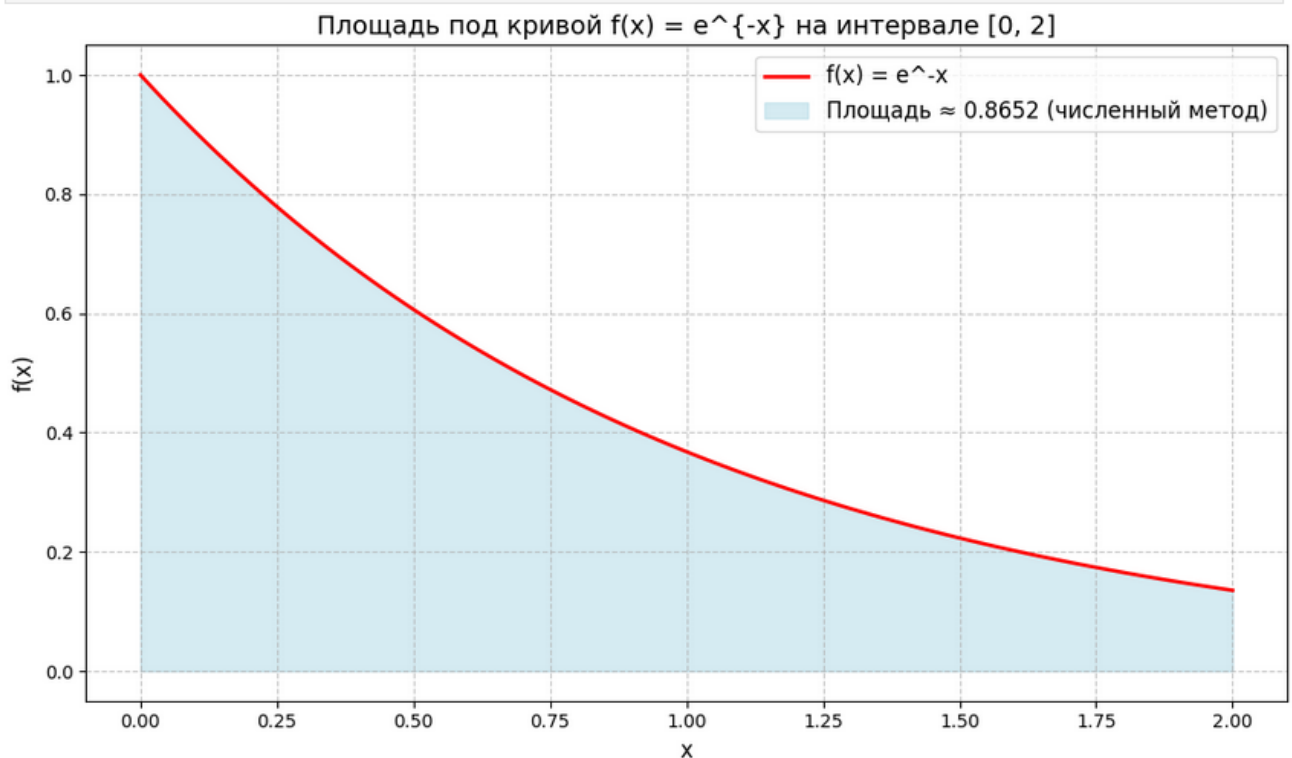


Рисунок 12. Индивидуальное задание №3

13. Выполнение индивидуального задания №4.

3. Гауссовский холм

Постройте поверхность:

$$f(x, y) = e^{-(x^2 + y^2)}$$

на интервале $x, y \in [-3, 3]$.

```
: import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

# Создаем сетку значений x и y
x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
X, Y = np.meshgrid(x, y)

# Вычисляем значения функции f(x, y) = e^{-(x^2 + y^2)}
Z = np.exp(-(X**2 + Y**2))

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

# Рисуем поверхность с цветовой картой
surf = ax.plot_surface(X, Y, Z, cmap=cm.viridis, linewidth=0, antialiased=True)

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

# Настраиваем заголовок и подписи осей
ax.set_title('Гауссовский холм: $f(x, y) = e^{-(x^2 + y^2)}$', fontsize=14)
ax.set_xlabel('x', fontsize=12)
ax.set_ylabel('y', fontsize=12)
ax.set_zlabel('f(x, y)', fontsize=12)

# Устанавливаем угол обзора
ax.view_init(elev=30, azim=45)

plt.tight_layout()
plt.show()
```

Гауссовский холм: $f(x, y) = e^{-(x^2 + y^2)}$

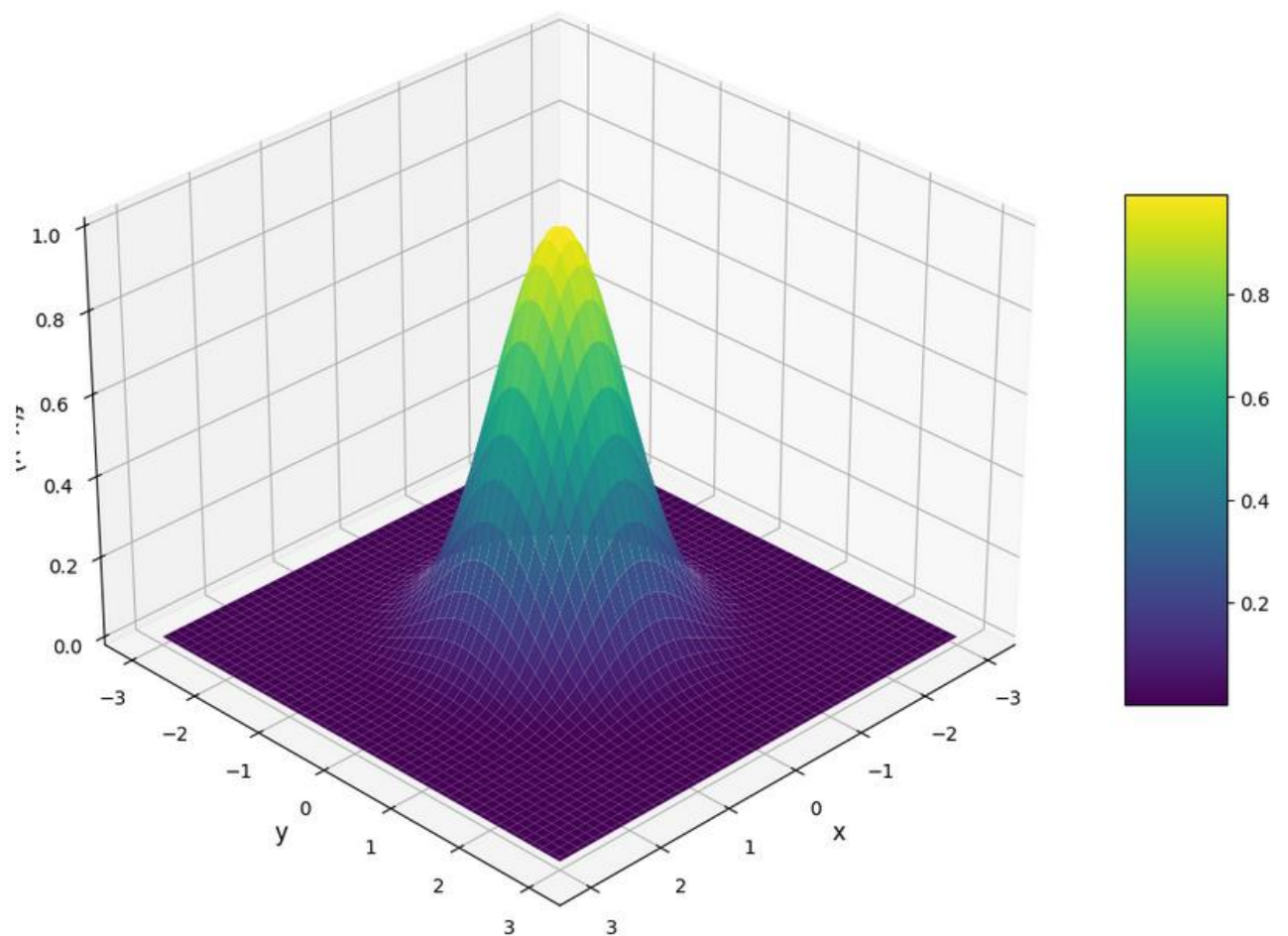


Рисунок 13. Индивидуальное задание №4

Ответы на контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Командой: `pip install matplotlib`

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

`%matplotlib inline`

3. Как отобразить график с помощью функции `plot()`?

Сначала создаётся массив x , затем $y = f(x)$, и строится график:

```
plt.plot(x, y)
```

```
plt.show()
```

4. Как отобразить несколько графиков на одном поле?

Вызвать `plot()` несколько раз до `show()`:

```
plt.plot(x, y1, label='график 1')
```

```
plt.plot(x, y2, label='график 2')
```

5. Какой метод известен для построения диаграмм категориальных данных?

Метод `bar()` или `barh()` для столбчатых диаграмм.

6. Какие основные элементы графика вам известны?

Оси, подписи осей, заголовки, легенда, сетка, линии, маркеры, цвет.

7. Как осуществляется управление текстовыми надписями на графике?

Через `plt.title()`, `plt.xlabel()`, `plt.ylabel()`, `plt.text()`, `plt.annotate()`.

8. Как осуществляется управление легендой графика?

Через `plt.legend()`. Опции: `loc`, `fontsize`, `title` и др.

9. Как задать цвет и стиль линий графика?

Аргументы plot():

```
plt.plot(x, y, color='red', linestyle='--', linewidth=2)
```

10. Как выполнить размещение графика в разных полях?

Через plt.subplots() или plt.subplot(). Например:

```
fig, axs = plt.subplots(2, 2)
```

```
axs[0, 0].plot(x, y)
```

11. Как выполнить построение линейного графика с помощью matplotlib?

Через plt.plot(x, y). Это по умолчанию линейный график.

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

Используется plt.fill_between():

```
plt.fill_between(x, y, color='lightblue')
```

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

Через условие в fill_between():

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

14. Как выполнить двухцветную заливку?

Несколько вызовов fill_between() с разными условиями.

15. Как выполнить маркировку графиков?

Через параметр label в plot(), и plt.legend().

16. Как выполнить обрезку графиков?

Через plt.xlim() и plt.ylim(), например:

```
plt.xlim(-5, 5)
```

17. Как построить ступенчатый график? В чем особенность? Используется `plt.step()`. Он отображает данные в виде "лестницы", полезен для дискретных изменений.

18. Как построить стекированный график? Используется `plt.stackplot()`:

```
plt.stackplot(x, y1, y2, labels=['y1','y2'])
```

19. Как построить stem-график? Через `plt.stem()`. Он показывает точки и вертикальные линии от оси.

20. Как построить точечный график? Через `plt.scatter(x, y)`. Особенность: визуализирует распределение точек по координатам.

21. Как осуществляется построение столбчатых диаграмм? С помощью `plt.bar(x, height)` или `plt.barh()`.

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

Групповая: несколько столбцов рядом для одной категории.
`errorbar`: диаграмма с отображением погрешности (`plt.errorbar`).

23. Как выполнить построение круговой диаграммы? `plt.pie(values, labels=..., autopct='% 1.1f%%')`

24. Что такое цветовая карта? Как работать с цветами в `matplotlib`? Цветовая карта (`colormap`) — отображает числовые значения в цвет. Пример:

```
plt.imshow(data, cmap='hot')
```

25. Как отобразить изображение средствами matplotlib?
Через `plt.imshow(img)`, где `img` — массив (например, из OpenCV или PIL).

26. Как отобразить тепловую карту средствами matplotlib?
`plt.imshow(matrix, cmap='hot')` или через `seaborn`.
Добавляют `plt.colorbar()` для шкалы.

27. Как выполнить построение линейного 3D-графика?
Через `ax.plot3D(x, y, z)`, где `ax` — `subplot(111, projection='3d')`.

28. Как построить точечную 3D-диаграмму?
`ax.scatter3D(x, y, z)`

29. Как выполнить построение каркасной поверхности?
`ax.plot_wireframe(X, Y, Z)`

30. Как построить трёхмерную поверхность?
`ax.plot_surface(X, Y, Z, cmap='viridis')`

Вывод: В ходе выполнения лабораторной работы были успешно освоены и применены ключевые возможности библиотеки Matplotlib, предназначенной для визуализации данных в Python. В рамках работы исследовались различные типы графиков, такие как линейные, точечные, столбчатые, круговые, тепловые карты и трёхмерные визуализации. Особое внимание уделялось настройке внешнего вида графиков: подбору цветовых схем, стилей линий, добавлению легенд, подписей осей, нанесению сетки, корректировке масштаба, а также компоновке нескольких графиков на одном изображении для удобства сравнения данных. Полученные навыки позволяют эффективно представлять данные в наглядной и информативной форме.