

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»

Выполнил:
Митряшкина Дарина Сергеевна
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.

Порядок выполнения работы:

1. Создан новый ноутбук, добавлены необходимые библиотеки

✓ Визуализация данных с использованием matplotlib

В этом ноутбуке реализованы задания по построению графиков, гистограмм, 3D-визуализации и индивидуальные задачи. Библиотеки: matplotlib, numpy.

```
[ ]
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.integrate import quad
%matplotlib inline
```

Рисунок 1. Новый ноутбук для выполнения заданий

2. Выполнение задания №1.

✓ Задание 1: График $y = x^2$

```
[ ]
x = np.linspace(-10, 10, 400)
y = x**2
plt.plot(x, y)
plt.title("График функции  $y = x^2$ ")
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True)
plt.show()
```

Рисунок 2. Задание №1

3. Выполнение задания №2.

✓ Задание 2: Несколько графиков на одном

```
[ ]
x = np.linspace(-10, 10, 400)
plt.plot(x, x, 'b--', label='y = x')
plt.plot(x, x**2, 'g-.', label='y = x^2')
plt.plot(x, x**3, 'r-', label='y = x^3')
plt.legend()
plt.title("Графики нескольких функций")
plt.axis('equal')
plt.grid(True)
plt.show()
```

Рисунок 3. Задание №2

4. Выполнение задания №3.

✓ Задание 3: Диаграмма рассеяния

```
[ ]
x = np.random.rand(50)
y = np.random.rand(50)
colors = x
sizes = y * 1000
plt.scatter(x, y, c=colors, s=sizes, alpha=0.7)
plt.colorbar(label='Значения X')
plt.title("Диаграмма рассеяния")
plt.show()
```

Рисунок 4. Задание №3

5. Выполнение задания №4.

✓ Задание 4: Гистограмма нормального распределения

```
[ ]
data = np.random.normal(0, 1, 1000)
plt.hist(data, bins=30, alpha=0.7, color='steelblue')
plt.axvline(np.mean(data), color='red', linestyle='dashed', linewidth=2)
plt.title("Гистограмма нормального распределения")
plt.show()
```

Рисунок 5. Задание №4

6. Выполнение задания №5.

✓ Задание 5: Столбчатая диаграмма по оценкам студентов

```
[ ]
labels = ['Отлично', 'Хорошо', 'Удовл.', 'Неуд.']
values = [20, 35, 30, 15]
plt.bar(labels, values, color='skyblue')
plt.title("Распределение оценок студентов")
plt.xlabel("Оценка")
plt.ylabel("Количество")
plt.show()
```

Рисунок 6. Задание №5

7. Выполнение задания №6.

✓ Задание 6: Круговая диаграмма по оценкам

```
[ ]
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title("Проценты оценок студентов")
plt.axis('equal')
plt.show()
```

Рисунок 7. Задание №6

8. Выполнение задания №7.

✓ Задание 7: 3D-график функции $z = \sin(\sqrt{x^2 + y^2})$

```
[ ]
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')
ax.set_title("z = sin(sqrt(x^2 + y^2))")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("z")
plt.show()
```

Рисунок 8. Задание №7

9. Выполнение задания №8.

✓ Задание 8: Сетка из 4 графиков

```
[ ]
x = np.linspace(-10, 10, 400)
fig, axs = plt.subplots(2, 2, figsize=(10, 8))

axs[0, 0].plot(x, x)
axs[0, 0].set_title("y = x")

axs[0, 1].plot(x, x**2)
axs[0, 1].set_title("y = x^2")

axs[1, 0].plot(x, np.sin(x))
axs[1, 0].set_title("y = sin(x)")

axs[1, 1].plot(x, np.cos(x))
axs[1, 1].set_title("y = cos(x)")

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

Рисунок 9. Задание №8

10. Выполнение задания №9.

✓ Задание 9: Тепловая карта

```
[ ]
matrix = np.random.rand(10, 10)
plt.imshow(matrix, cmap='hot', interpolation='nearest')
plt.colorbar()
plt.title("Тепловая карта 10x10")
plt.show()
```

Рисунок 10. Задание №8

11. Выполнение индивидуального задания №1.

✓ Индивидуальное задание 1: Температурные изменения

```
[ ]
hours = [0, 3, 6, 9, 12, 15, 18, 21, 24]
temps = [-5, -6, -4, 2, 8, 12, 10, 3, -2]
colors = ['blue' if t < 0 else 'orange' for t in temps]
plt.bar(hours, temps, color=colors)
plt.title("Температура в течение суток")
plt.xlabel("Время (часы)")
plt.ylabel("Температура (°C)")
plt.grid(True)
plt.show()
```

Рисунок 11. Индивидуальное задание №1

12. Выполнение индивидуального задания №2.

✓ Индивидуальное задание 2: Публикации по областям

```
[ ]
fields = ['Математика', 'Физика', 'Биология', 'Химия', 'Комп. науки']
counts = [120, 90, 150, 110, 180]
colors = ['gray'] * 5
max_index = counts.index(max(counts))
colors[max_index] = 'red'

plt.barh(fields, counts, color=colors)
plt.title("Научные публикации по областям знаний")
plt.xlabel("Количество публикаций")
plt.show()
```

Рисунок 12. Индивидуальное задание №2

13. Выполнение индивидуального задания №3.

Индивидуальное задание 3: Площадь под кривой $f(x) = 2x + 3$

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

# Функция
f = lambda x: 2 * x + 3

# Интервал
x = np.linspace(-2, 2, 500)
y = f(x)

# Площадь под кривой методом трапеций
area = np.trapz(y, x)

# Построение графика
plt.plot(x, y, label='f(x) = 2x + 3')
plt.fill_between(x, y, color='skyblue', alpha=0.4)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.title(f'Площадь под кривой: {area:.4f}')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()
plt.grid(True)
plt.show()
```

Рисунок 13. Индивидуальное задание №3

14. Выполнение индивидуального задания №4.

▼ Индивидуальное задание 4: Экспоненциальный гребень

```

x = np.linspace(-3, 3, 200)
y = np.linspace(-3, 3, 200)
X, Y = np.meshgrid(x, y)
R = np.sqrt(X**2 + Y**2)
Z = np.exp(-R) * np.cos(2 * np.pi * R)

fig = plt.figure(figsize=(12, 6))

ax1 = fig.add_subplot(121, projection='3d')
surf = ax1.plot_surface(X, Y, Z, cmap='plasma')
ax1.set_title("Экспоненциальный гребень (поверхность)")
ax1.set_xlabel("x")
ax1.set_ylabel("y")
ax1.set_zlabel("z")
fig.colorbar(surf, ax=ax1)

ax2 = fig.add_subplot(122, projection='3d')
ax2.plot_wireframe(X, Y, Z, color='green')
ax2.set_title("Экспоненциальный гребень (каркас)")
ax2.set_xlabel("x")
ax2.set_ylabel("y")
ax2.set_zlabel("z")
```

Рисунок 14. 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, заголовок, 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 изучены и практически реализованы ключевые возможности библиотеки Matplotlib,

предназначенной для визуализации данных в Python. Были рассмотрены основные функции построения графиков: линейных, точечных, столбчатых, круговых, тепловых и 3D-графиков. Также освоены средства настройки графиков — цвета, стили, легенды, подписи, сетка, масштаб и размещение нескольких графиков на одном или нескольких полях. Особое внимание было уделено функциям `plot()`, `scatter()`, `bar()`, `pie()`, `fill_between()`, а также работе с трехмерной визуализацией через `plot_surface()`, `wireframe()`, `scatter3D()` и `step`-графиками. Кроме того, были рассмотрены методы отображения изображений и построения цветowych карт, управления областями отображения и условными заливками.

Ссылка на Google Colab:

https://colab.research.google.com/drive/1GhFdMBbf20sgBdeeHgONLJg_1FKQcl6F?usp=sharing

Ссылка на Git Hub: https://github.com/darina-rtm/third_lab_ai.git