

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:
Пугачев Кирилл Дмитриевич
2 курс, группа ИТС-б-о-23-1,
11.03.02 Инфокоммуникационные
технологии и системы связи,
очная форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: ОСНОВЫ РАБОТЫ С БИБЛИОТЕКОЙ MATPLOTLIB

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

Ссылка на репозиторий: <https://github.com/chillkirill/LABA3AI>

Ход работы:

1. Выполнение практических заданий.

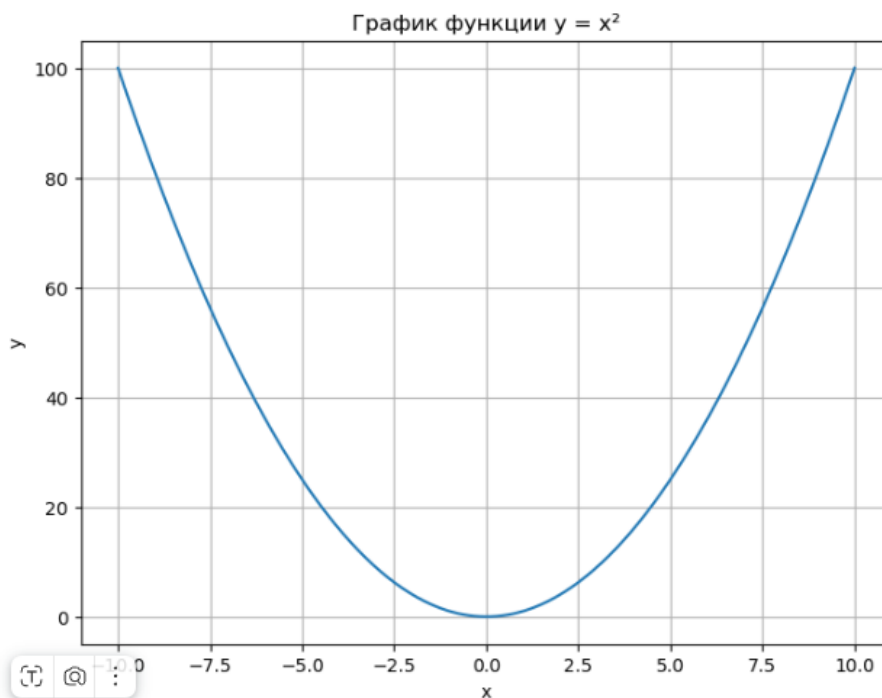
```
[1]: import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10, 10, 400) # 400 точек на интервале [-10, 10]
y = x**2                      # y = x^2 для каждой точки x

plt.figure(figsize=(8, 6)) # размер графика в дюймах
plt.plot(x, y)             # Строим график функции x и y

plt.title('График функции y = x^2')
plt.xlabel('x')             # подпись оси x
plt.ylabel('y')
plt.grid(True)              # сетка

plt.show()
```



[]: | ✚ 🖨 ⬆ ⬇

Рисунок 1. Практическое задание 1

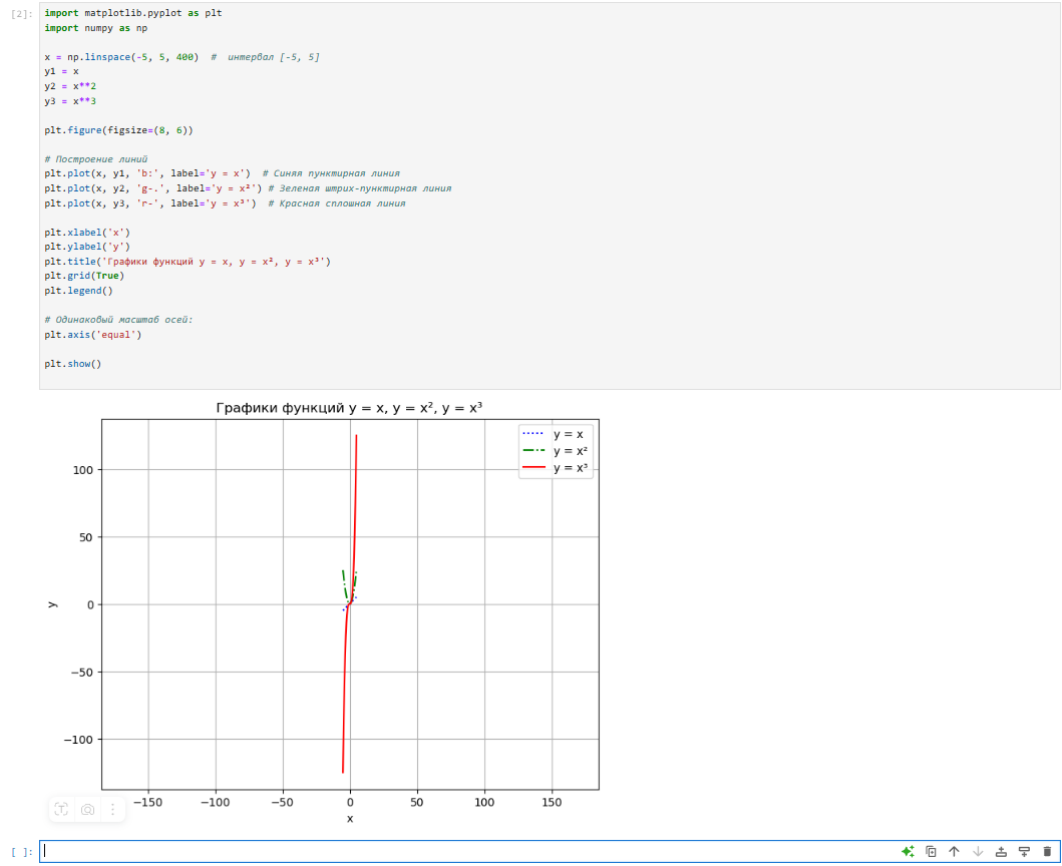


Рисунок 2. Практическое задание 2

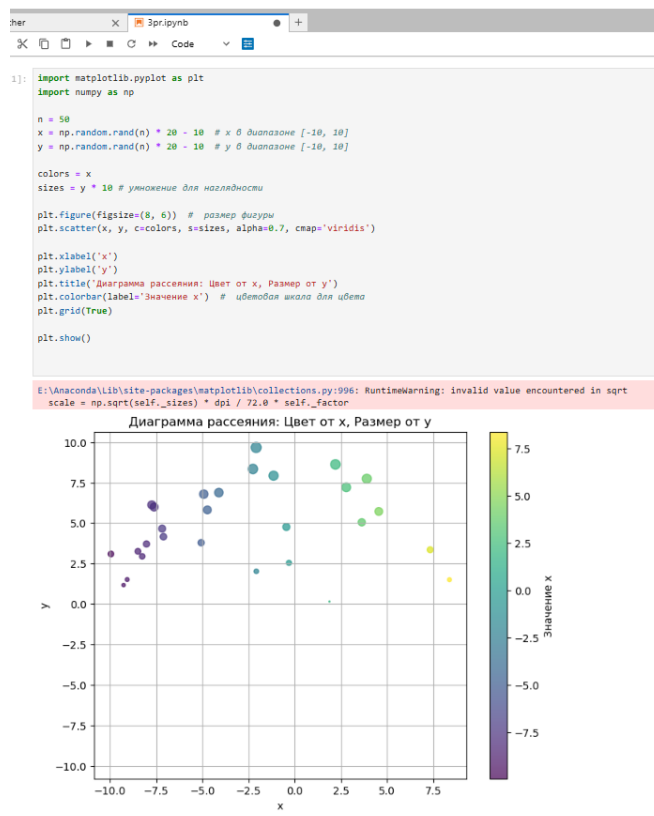


Рисунок 3. Практическое задание 3

```
[1]: import matplotlib.pyplot as plt
import numpy as np

mu = 0 # математическое ожидание
sigma = 1 # стандартное отклонение
num_points = 1000
data = np.random.normal(mu, sigma, num_points)

num_bins = 30
plt.figure(figsize=(8, 6))
n, bins, patches = plt.hist(data, num_bins, facecolor='blue', alpha=0.7)

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

plt.xlabel('Значение')
plt.ylabel('Частота')
plt.title('Гистограмма нормального распределения')
plt.grid(True)
plt.legend()

plt.show()
```

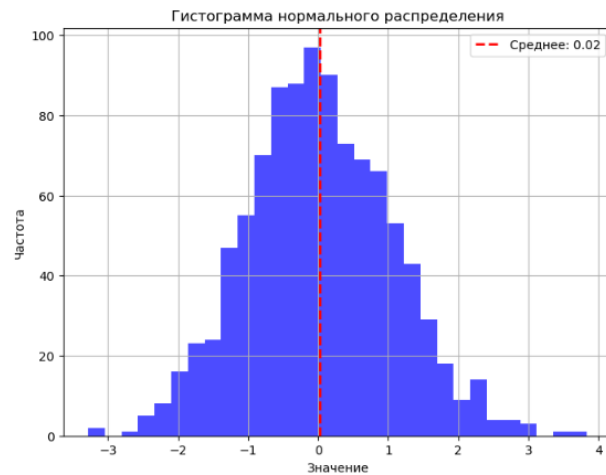


Рисунок 4. Практическое задание 4

```
[1]: import matplotlib.pyplot as plt
import numpy as np

grades = ['Отлично', 'Хорошо', 'Удовлетворительно', 'Неудовлетворительно']
num_students = [20, 35, 30, 15]

# столбчатая диаграмма
plt.figure(figsize=(8, 6))
plt.bar(grades, num_students, color='skyblue')

plt.xlabel('Оценка') # ос X
plt.ylabel('Количество студентов') # ос Y
plt.title('Распределение оценок студентов')
plt.grid(axis='y', alpha=0.5)

for i, value in enumerate(num_students):
    plt.text(i, value + 0.5, str(value), ha='center', va='bottom')

plt.show()
```

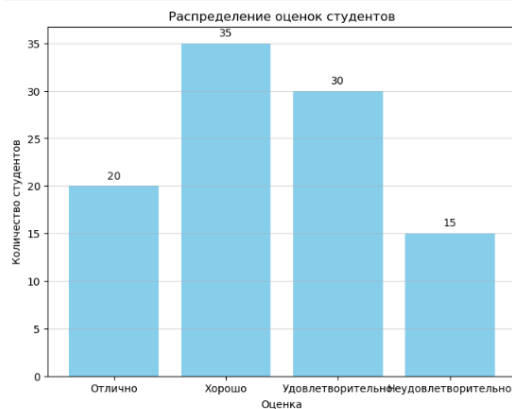


Рисунок 5. Практическое задание 5

```
[1]: import matplotlib.pyplot as plt
import numpy as np

grades = ['Отлично', 'Хорошо', 'Удовлетворительно', 'Неудовлетворительно']
num_students = [20, 35, 30, 15]

# круговая диаграмма
plt.figure(figsize=(8, 6))

plt.pie(num_students, labels=grades, autopct='%1.1f%%', startangle=90, colors=['gold', 'lightgreen', 'lightcoral', 'lightskyblue'])

plt.title('Распределение оценок студентов (круговая диаграмма)')
plt.axis('equal') # круг а не эллипс

plt.show()
```

Распределение оценок студентов (круговая диаграмма)

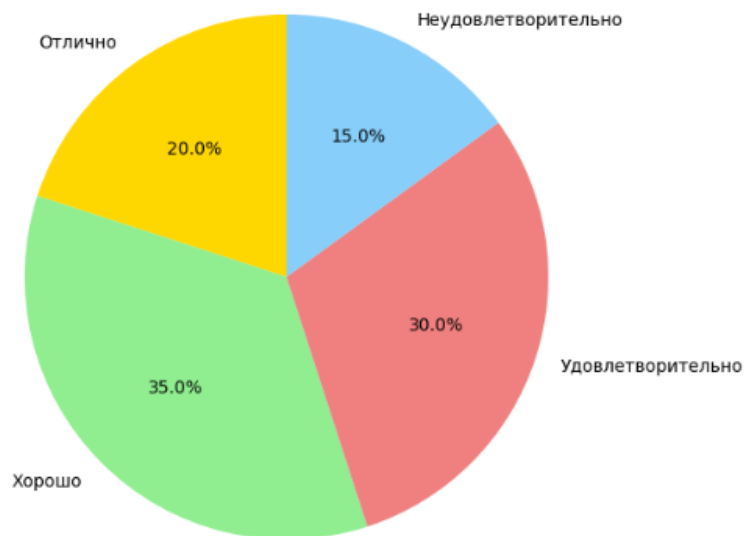


Рисунок 6. Практическое задание 6

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

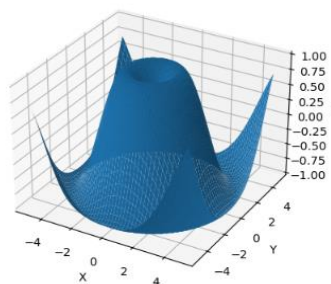
# сетка значений x и y
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)

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

# 3D-график
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

plt.show()
```



[1]:

Рисунок 7. Практическое задание 7

```
[2]: import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-5, 5, 100) # 100 точек в диапазоне от -5 до 5

fig, axes = plt.subplots(2, 2, figsize=(10, 8)) # фигура 2x2

# Линейный график y = x
axes[0, 0].plot(x, x)
axes[0, 0].set_title('Линейный график y = x')
axes[0, 0].set_xlabel('x')
axes[0, 0].set_ylabel('y')
axes[0, 0].grid(True)

# Парабола y = x^2
axes[0, 1].plot(x, x**2)
axes[0, 1].set_title('Парабола y = x^2')
axes[0, 1].set_xlabel('x')
axes[0, 1].set_ylabel('y')
axes[0, 1].grid(True)

# Синус y = sin(x)
axes[1, 0].plot(x, np.sin(x))
axes[1, 0].set_title('Синус y = sin(x)')
axes[1, 0].set_xlabel('x')
axes[1, 0].set_ylabel('y')
axes[1, 0].grid(True)

# Косинус y = cos(x)
axes[1, 1].plot(x, np.cos(x))
axes[1, 1].set_title('Косинус y = cos(x)')
axes[1, 1].set_xlabel('x')
axes[1, 1].set_ylabel('y')
axes[1, 1].grid(True)

# подгонка графиков
plt.tight_layout()

plt.show()
```

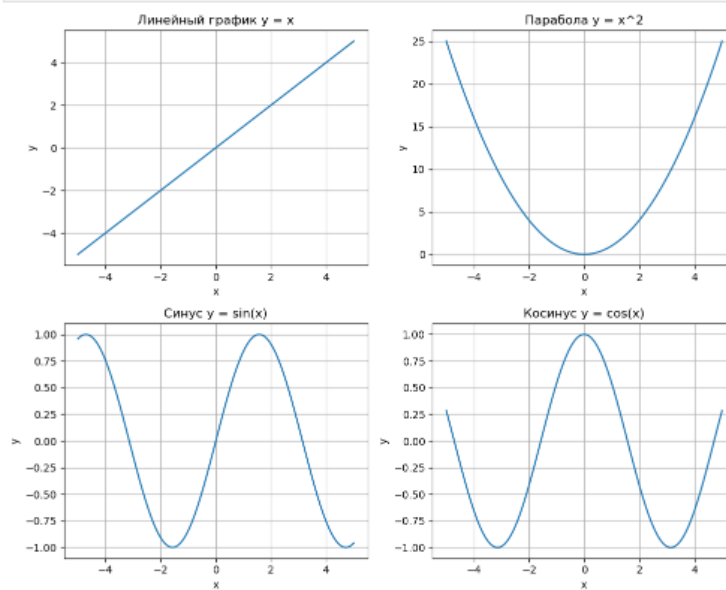


Рисунок 8. Практическое задание 8

```
[1]: import numpy as np
import matplotlib.pyplot as plt

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

# матрица как тепловая карта
plt.imshow(matrix, cmap='viridis', interpolation='nearest') # цветовая палитра
plt.colorbar(label='Значение') # цветовая шкала

plt.title('Тепловая карта случайной матрицы 10x10')
plt.xlabel('Столбец')
plt.ylabel('Строка')
plt.xticks(np.arange(10)) # все ответы на оси x
plt.yticks(np.arange(10)) # все ответы на оси y

plt.show()
```

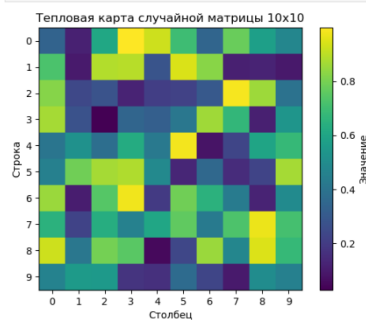


Рисунок 9. Практическое задание 9

```
[1]: import matplotlib.pyplot as plt

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]

plt.figure(figsize=(10, 6))
plt.plot(months, exchange_rate, color='red', marker='o', linestyle='-')

# подписи к точкам данных
for i, rate in enumerate(exchange_rate):
    plt.text(months[i], rate, str(rate), ha='center', va='bottom') # текст над каждой точкой

plt.xlabel('Месяц')
plt.ylabel('Курс (₽ за 1$)')
plt.title('Изменение курса доллара к рублю за 12 месяцев')
plt.xticks(months) # все месяцы на оси x
plt.grid(True) # сетка для лучшей читаемости

plt.show()
```

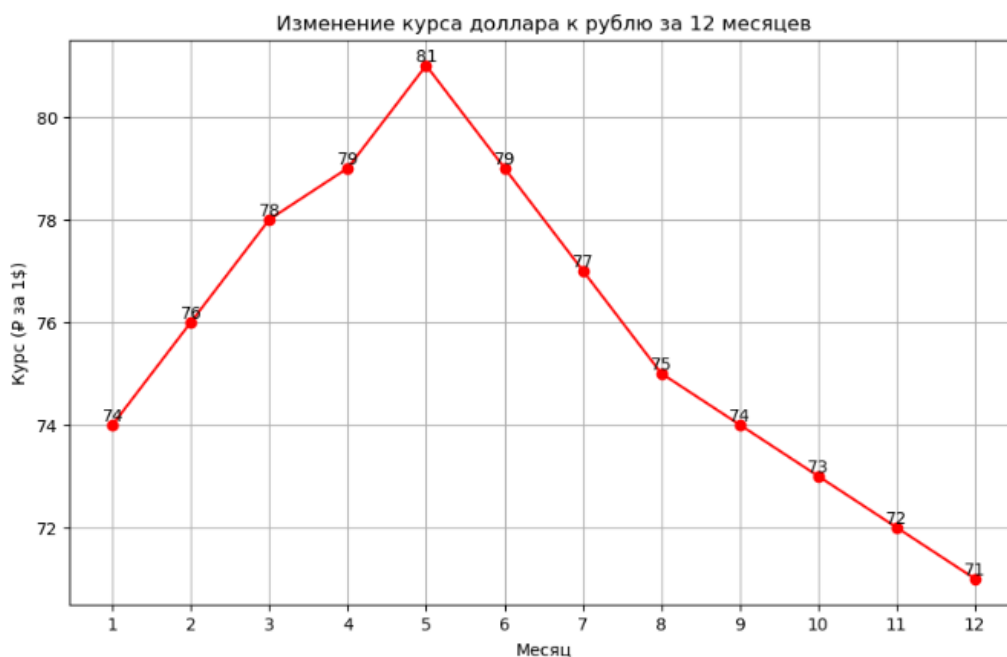


Рисунок 10. Индивидуальное задание 1

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.

Рисунок 11. Первое индивидуальное задание к 3 варианту

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.

Рисунок 12. Второе индивидуальное задание к 3 варианту

```
[1]: import matplotlib.pyplot as plt

days = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс']
visitors = [150, 180, 200, 220, 300, 450, 500]

# цвета для будних и выходных дней
colors = ['blue'] * 5 + ['green'] * 2 # Первые 5 дней - будние (синий) последние 2 - выходные (зеленый)

# столбчатая диаграмма
plt.figure(figsize=(8, 6))
plt.bar(days, visitors, color=colors)

plt.xlabel('День недели')
plt.ylabel('Количество посетителей')
plt.title('Количество посетителей за неделю')

plt.show()
```



Рисунок 13. Индивидуальное задание 2

3. Площадь под экспонентой

Определите площадь под функцией:

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

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

Рисунок 14. Третье индивидуальное задание к 3 варианту

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

def f(x):
    return np.exp(-x)

# интервал интегрирования
a = 0
b = 2

# площадь под кривой (определенный интеграл)
area, error = quad(f, a, b)

# данные для графика
x = np.linspace(a, b, 100) # 100 точек между a и b для плавного графика
y = f(x)

# график функции
plt.figure(figsize=(8, 6))
plt.plot(x, y, 'r-', linewidth=2, label=r'$f(x) = e^{-x}$') # Красная линия для функции

# Закрашивание области под кривой
x_fill = np.linspace(a, b, 100)
y_fill = f(x_fill)
plt.fill_between(x_fill, y_fill, color='skyblue', alpha=0.5, label='Площадь под кривой')

# подписи и заголовок
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title(f'Площадь под кривой f(x) = e^{-x} на [{a}, {b}]')
plt.grid(True) # сетка для наглядности
plt.legend()

# значение площади на графике
plt.text(0.5 * (a + b), 0.2 * np.max(y), f'Площадь = {area:.4f}', fontsize=12, ha='center')

plt.show()
```

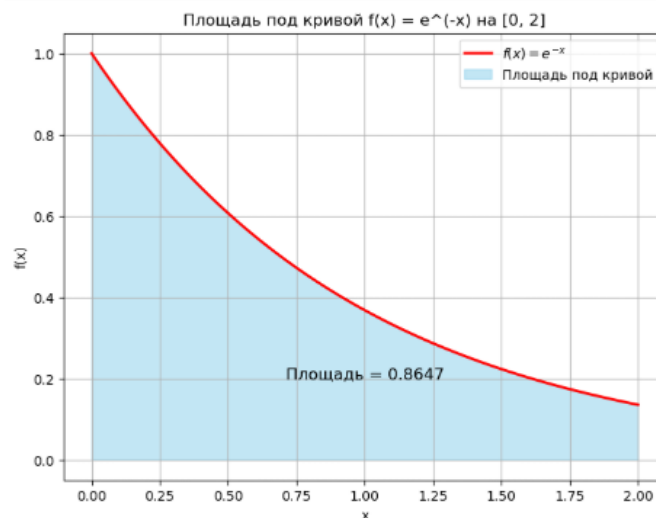


Рисунок 15. Индивидуальное задание 3

3. Гауссовский холм

Постройте поверхность:

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

на отрезке $x, y \in [-3, 3]$.

Рисунок 16. Четвертое индивидуальное задание к 3 варианту

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

# функция
def f(x, y):
    return np.exp(-(x**2 + y**2))

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

# значения функции на сетке
Z = f(X, Y)

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

# построение поверхности
surf = ax.plot_surface(X, Y, Z, cmap='viridis')

ax.set_title('График функции f(x, y) = e^(-(x^2 + y^2))')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('f(x, y)')

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

plt.show()
```

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

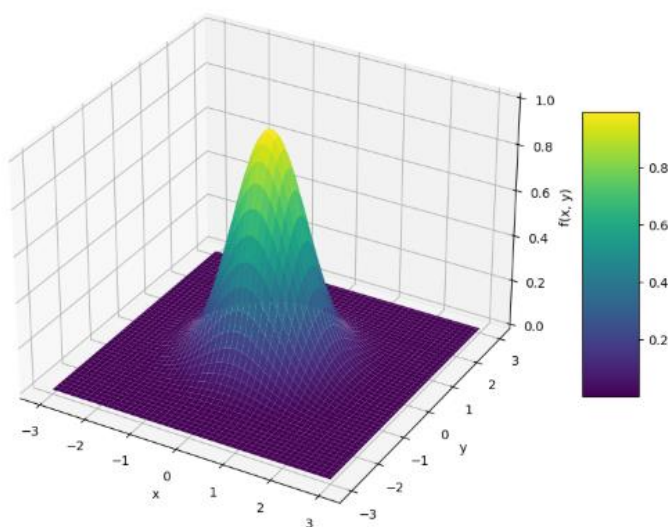


Рисунок 17. Индивидуальное задание 4

2. Создание репозитория и работа с ним.

Создайте новый репозиторий

Репозиторий содержит все файлы проекта, включая историю изменений. У вас уже есть репозиторий проекта в другом месте? [Импортируйте репозиторий.](#)

Обязательные поля отмечены звездочкой (*).

Владелец *

охладите курилл

/

Название репозитория *

LABAZAI

Доступен LABAZAI.

Хорошие названия репозитория короткие и запоминающиеся. Нужно вдохновение? Как насчёт [специального приключения?](#)

Описание (необязательно)

☒

Публичный

любой пользователь интернета может увидеть этот репозиторий. Вы выбираете, кто может совершать коммиты.

☐

Частное

Вы сами выбираете, кто может просматривать этот репозиторий и фиксировать его в нем.

Инициализируйте этот репозиторий с помощью:

☒ **Добавьте файл README**
Здесь вы можете написать подробное описание своего проекта. [Узнайте больше о файлах README.](#)

Добавить .gitignore

.шаблон gitignore: **Отсутствует**

Выберите, какие файлы не отслеживать, из списка шаблонов. [Узнайте больше об игнорировании файлов.](#)

Выберите лицензию

Лицензия: **Отсутствует**

Лицензия сообщает другим пользователям, что они могут и чего не могут делать с вашим кодом. [Узнайте больше о лицензиях.](#)

Это сделает `main` веткой по умолчанию. Измените имя по умолчанию в ваших [настройках](#).

Вы создаете публичный репозиторий в своем личном кабинете.

Создать репозиторий

Рисунок 18. Создание репозитория

LABAZAI

Общедоступный

Булавка

Разворачивать 1

Главная

1 Филиал

0 Тегов

Поиск

Перейти к файлу

Добавить файл

Код

охладите куриллу

Первоначальная фиксация

d32d04a-сейчас

1 Совершить

.гитиньоре

Первоначальная фиксация

сейчас

ЛИЦЕНЗИЯ

Первоначальная фиксация

сейчас

README.md

Первоначальная фиксация

сейчас

ПРОЧИТАЙ МЕНЯ

MIT license

ЛАБАЗАИ

Рисунок 19. Созданный репозиторий

```
C:\Users\user>git clone https://github.com/chillkirill/LABA3AI.git
Cloning into 'LABA3AI'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рисунок 20. Клонирование репозитория

```
C:\Users\user\LABA3AI>git add .
warning: in the working copy of '3laba/1ind.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/1pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/2ind.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/2pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/3ind.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/3pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/4ind.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/4pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/5pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/6pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/7pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/8pr.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '3laba/9pr.ipynb', LF will be replaced by CRLF the next time Git touches it

C:\Users\user\LABA3AI>git commit -m "Добавление файлов с которыми работал"
[main cbac8c3] Добавление файлов с которыми работал
13 files changed, 998 insertions(+)
create mode 100644 3laba/1ind.ipynb
create mode 100644 3laba/1pr.ipynb
create mode 100644 3laba/2ind.ipynb
create mode 100644 3laba/2pr.ipynb
create mode 100644 3laba/3ind.ipynb
create mode 100644 3laba/3pr.ipynb
create mode 100644 3laba/4ind.ipynb
create mode 100644 3laba/4pr.ipynb
create mode 100644 3laba/5pr.ipynb
create mode 100644 3laba/6pr.ipynb
create mode 100644 3laba/7pr.ipynb
create mode 100644 3laba/8pr.ipynb
create mode 100644 3laba/9pr.ipynb

C:\Users\user\LABA3AI>git push
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 647.28 KiB | 19.04 MiB/s, done.
Total 16 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/chillkirill/LABA3AI.git
 d32d04a..cbac8c3  main -> main

C:\Users\user\LABA3AI>
```

Рисунок 21. Создание коммита с несколькими рабочими файлами

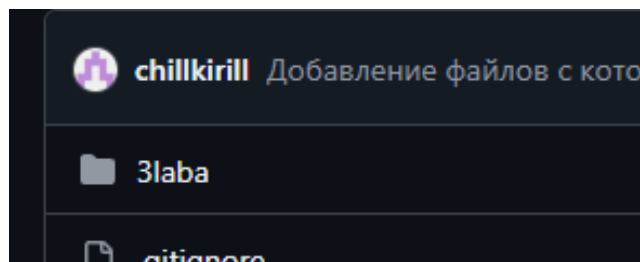


Рисунок 22. Результат коммита

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)`