

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 ПРАКТИЧЕСКОЙ РАБОТЕ №3
дисциплины
«Искусственный интеллект и машинное обучение»
Вариант 4

Выполнила:
Баранник Софья Евгеньевна
2 курс, группа ИВТ-б-о-23-1,
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 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:

Ссылка на репозиторий GitHub: <https://github.com/S-o-n-y-a/AI-ML.git>

1. Написали код, который строит график функции $y = x^2$ на интервале $[-10, 10]$. Добавили заголовок, подписи осей и сетку.

```
: import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-10,10,50) ## (координата, координата, кол-во точек)
y = x ** 2
plt.title("График функции  $y = x^2$ ")
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
plt.plot(x, y)

: [matplotlib.lines.Line2D at 0x229a58810d0]
```

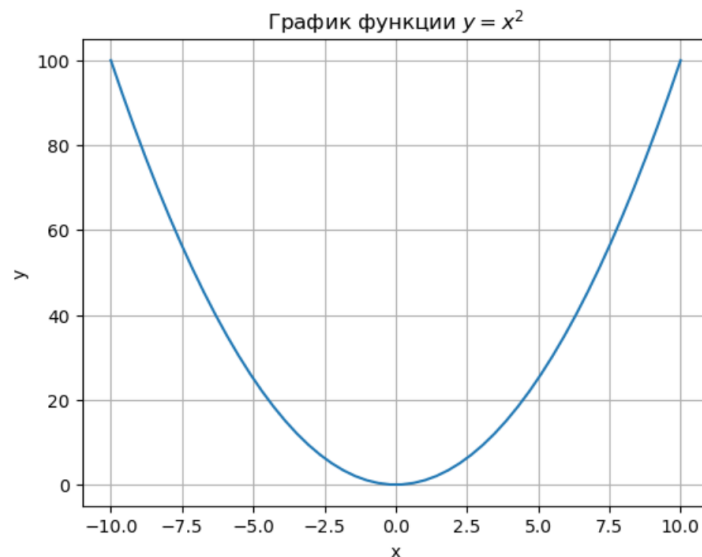


Рисунок 1. Первое задание

2. Построили три линии на одном графике:

- $y = x$ (синяя, пунктирная линия),
- $y = x^2$ (зеленая, штрихпунктирная линия),
- $y = x^3$ (красная, сплошная линия).

Добавили легенду и сделали оси одинакового масштаба.

```

x = np.linspace(-10, 10, 50)
y1 = x
y2 = x ** 2
y3 = x ** 3
plt.grid()
plt.xlabel("x", color="blue")
plt.ylabel("y", color="blue")
plt.plot(x, y1, "--b", label=r"$y = x$")
plt.plot(x, y2, "-.g", label=r"$y = x^2$")
plt.plot(x, y3, "-r", label=r"$y = x^3$")
plt.legend()
plt.ylim(-500, 500)
plt.xlim(-10, 10)

```

(-10.0, 10.0)

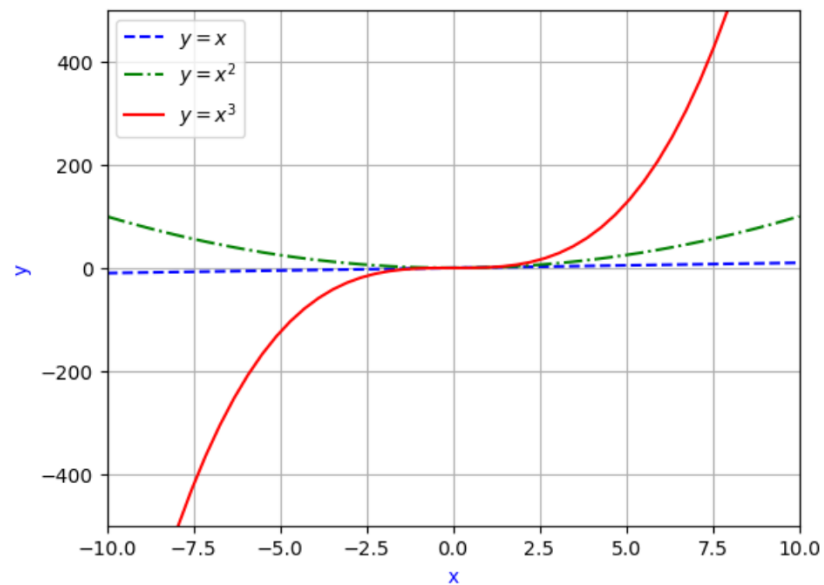


Рисунок 2. Второе задание

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

```

x = np.random.randint(-10, 10, 50)
y = np.random.randint(-10, 10, 50)
colors = x
sizes = np.abs(y) * 10 # размер маркеров y
plt.scatter(x, y, c=colors, s=sizes, alpha=0.7)
plt.xlabel("x")
plt.ylabel("y")
plt.grid()

```

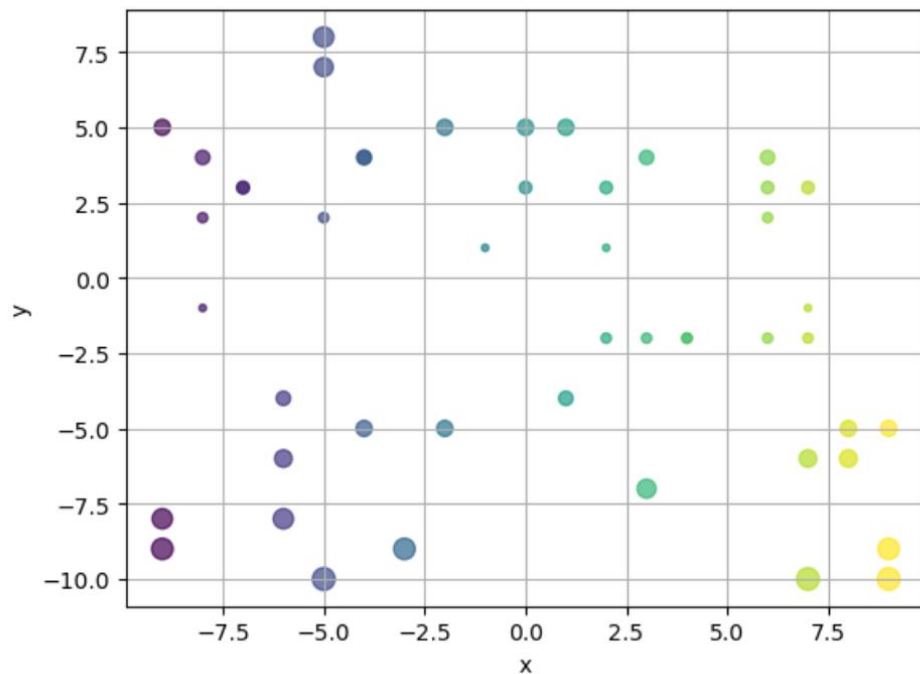


Рисунок 3. Третье задание

4. Сгенерируйте 1000 случайных чисел из нормального распределения с параметрами $\mu = 0$, $\sigma = 1$ и постройте их гистограмму с 30 бинами. Добавьте вертикальную линию в среднем значении.

```

: nums = np.random.normal(loc=0, scale=1, size=1000) # loc-распределение, scale-дисперсия
  center_num = np.mean(nums) # среднее значение
  plt.hist(nums, bins=30, edgecolor='black', alpha=0.75)
  plt.axvline(center_num, color="red", linestyle="dashed", linewidth=2, label="Среднее значение")
  plt.title("Гистограмма 1000 случайных чисел")
  plt.xlabel("Значение")
  plt.ylabel("Частота")
  plt.legend()

```

```

: <matplotlib.legend.Legend at 0x1cb67ab8d70>

```

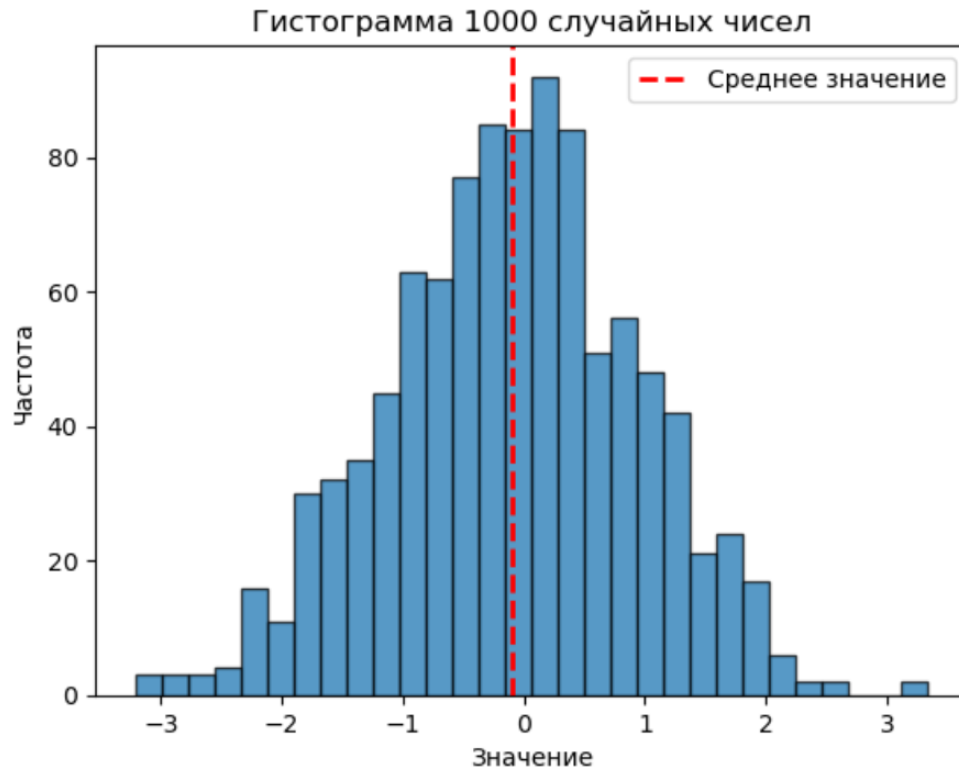


Рисунок 4. Задание 4

5. Построили столбчатую диаграмму, которая показывает количество студентов, получивших оценки:

- "Отлично" — 20 человек,
- "Хорошо" — 35 человек,
- "Удовлетворительно" — 30 человек,
- "Неудовлетворительно" — 15 человек.

Добавили подписи к осям и заголовок.

```
marks = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
amount = [20, 35, 30, 15]
plt.bar(marks, amount, color=['green', 'blue', 'orange', 'red'])
plt.xlabel("Оценки")
plt.ylabel("Количество студентов")
plt.title("Оценки студентов")
```

Text(0.5, 1.0, 'Оценки студентов')

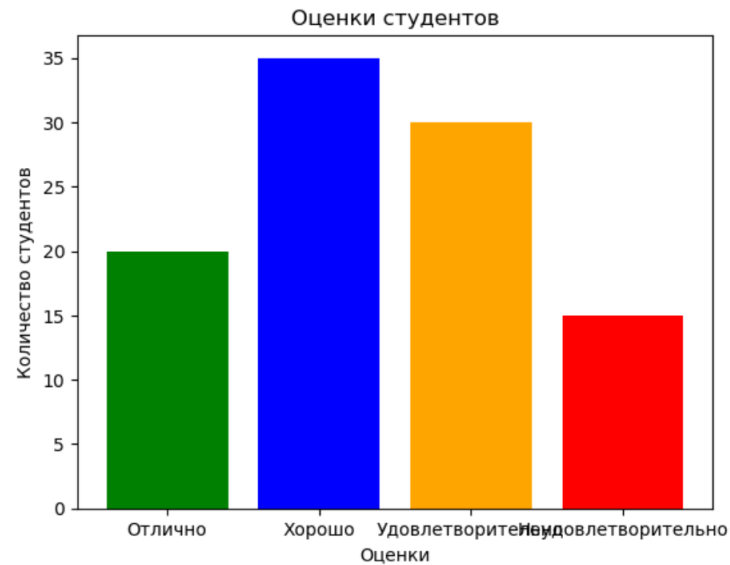


Рисунок 5. Пятое задание

6. Используя данные предыдущей задачи, построили круговую диаграмму с процентными подписями секторов.

```
: marks = ["Отлично", "Хорошо", "Удовлетворительно", "Неудовлетворительно"]
amount = [20, 35, 30, 15]
plt.pie(amount, labels=marks, autopct="%1.1f%%", colors=["green", "blue", "orange", "red"])

: ([<matplotlib.patches.Wedge at 0x1cb687fbb90>,
<matplotlib.patches.Wedge at 0x1cb687fb8c0>,
<matplotlib.patches.Wedge at 0x1cb68834800>,
<matplotlib.patches.Wedge at 0x1cb68834e90>],
[Text(0.8899186877588753, 0.6465637858537406, 'Отлично'),
Text(-0.7778174593052023, 0.7778174593052024, 'Хорошо'),
Text(-0.33991867422268845, -1.0461621742897658, 'Удовлетворительно'),
Text(0.9801072140121813, -0.4993894763020948, 'Неудовлетворительно')],
[Text(0.4854101933230228, 0.3526711559202213, '20.0%'),
Text(-0.42426406871192845, 0.4242640687119285, '35.0%'),
Text(-0.18541018593964823, -0.5706339132489631, '30.0%'),
Text(0.5346039349157352, -0.27239425980114257, '15.0%')])
```

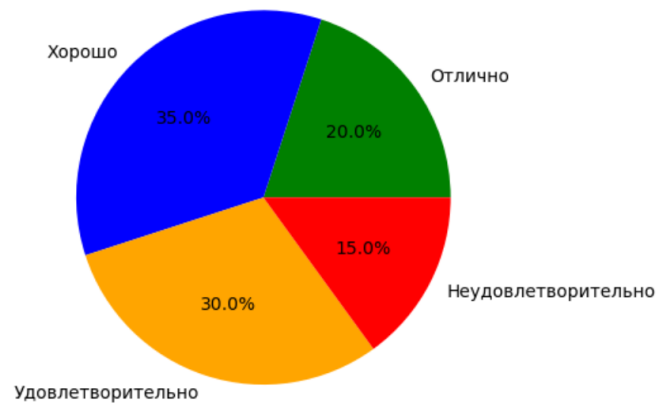


Рисунок 6. Шестое задание

7. Используя `mpl_toolkits.mplot3d`, постройте 3D-график функции $z = \sin(x^2 + y^2)$ на сетке значений x, y в диапазоне $[-5, 5]$.

```
: from mpl_toolkits.mplot3d import Axes3D
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
z = np.sin(np.sqrt(X**2 + Y**2))
fig = plt.figure()
funct = fig.add_subplot(111, projection='3d')
color_map = funct.plot_surface(X, Y, z, cmap='viridis')
funct.set_xlabel('x')
funct.set_ylabel('y')
funct.set_zlabel('z')

: Text(0.5, 0, 'z')
```

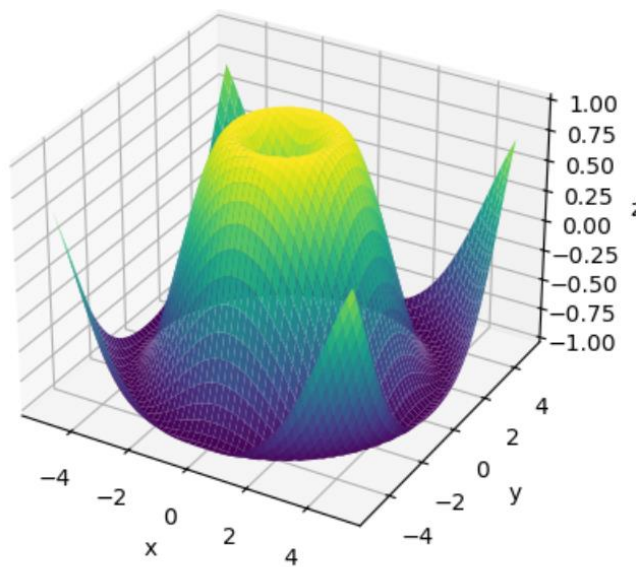


Рисунок 7. Седьмое задание

8. Построили четыре графика в одной фигуре (2×2 сетка):

- Линейный график $y = x$,
- Парабола $y = x^2$,
- Синус $y = \sin(x)$,
- Косинус $y = \cos(x)$.

Добавили подписи к осям и заголовок.

```

x = np.linspace(-10, 10, 400)
fig, functions = plt.subplots(2, 2, figsize=(10, 8))
functions[0, 0].plot(x, x)
functions[0, 0].set_title('Линейный график:  $y = x$ ')
functions[0, 1].plot(x, x**2)
functions[0, 1].set_title('Парабола:  $y = x^2$ ')
functions[1, 0].plot(x, np.sin(x))
functions[1, 0].set_title('Синус:  $y = \sin(x)$ ')
functions[1, 1].plot(x, np.cos(x))
functions[1, 1].set_title('Косинус:  $y = \cos(x)$ ')

```

Рисунок 8. Код восьмого задания

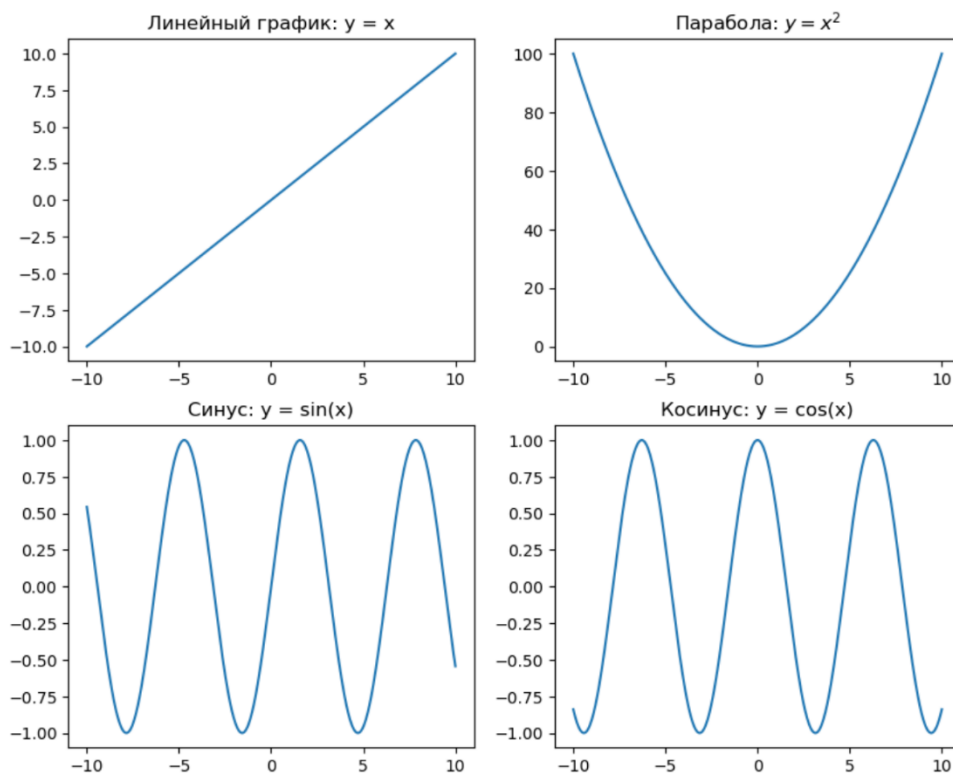


Рисунок 9. Графики восьмого задания

9. Создали случайную матрицу 10×10 с элементами от 0 до 1 и визуализировали её как тепловую карту с цветовой шкалой.


```
matrix = np.random.rand(10, 10)
plt.imshow(matrix, cmap='viridis')
plt.colorbar()
```

<matplotlib.colorbar.Colorbar at 0x229a7190b60>

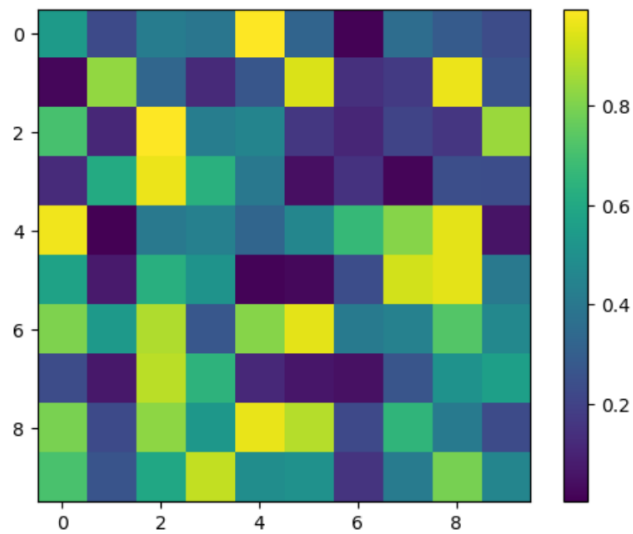


Рисунок 10. Девятое задание

10. В компании зарплата сотрудников зависит от стажа работы:

- Стаж (лет): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
- Зарплата (тыс. ₽): [40, 42, 45, 50, 55, 60, 65, 70, 75, 80].

Добавили сетку и изменили толщину линии графика.

```
import matplotlib.pyplot as plt
import numpy as np
years = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
salary = [40, 42, 45, 50, 55, 60, 65, 70, 75, 80]
plt.title("Зависимость зарплаты сотрудников от стажа работы")
plt.xlabel("Стаж (лет)")
plt.ylabel("Зарплата (тыс. Р)")
plt.grid()
plt.plot(years, salary, linewidth=4)
```

[<matplotlib.lines.Line2D at 0x1994d4a8830>]

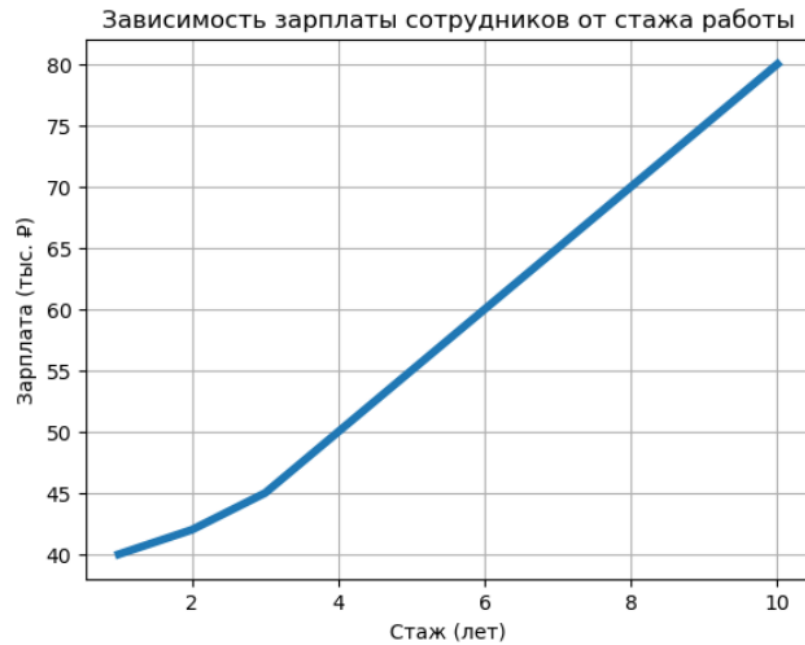


Рисунок 11. Первое индивидуальное задание

11. В компании есть автомобили разных марок:

- Марки: ['Toyota', 'BMW', 'Mercedes', 'Ford', 'Honda'],
- Количество автомобилей: [25, 15, 10, 20, 18].

Добавили отображение числовых значений над столбцами.

```

brands = ['Toyota', 'BMW', 'Mercedes', 'Ford', 'Honda']
quantity = [25, 15, 10, 20, 18]
for i in range(len(brands)):
    plt.text(i, quantity[i] + 0.5, str(quantity[i]), ha='center')
plt.xlabel("Марки")
plt.ylabel("Количество автомобилей")
plt.title("Марки автомобилей")
plt.bar(brands, quantity)

```

<BarContainer object of 5 artists>

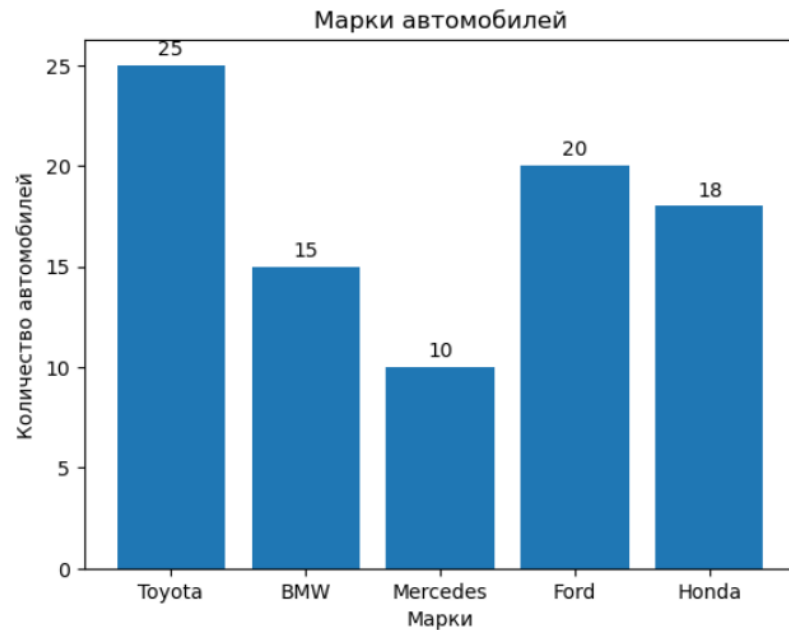


Рисунок 12. Второе индивидуальное задание

12. Вычислили площадь под графиком функции $f(x) = \ln(x)$ на интервале $[1, 4]$. Закрасили соответствующую область.

```

a = 1
b = 4
x = np.linspace(a, b, 200)
y = np.log(x)
area = np.trapz(y, x)
print("Площадь под графиком: ", area)
plt.plot(x, y)
plt.fill_between(x, y, where=(x>=a) & (x<=b))

```

Площадь под графиком: 2.545163240434021

<matplotlib.collections.PolyCollection at 0x1994d750ad0>

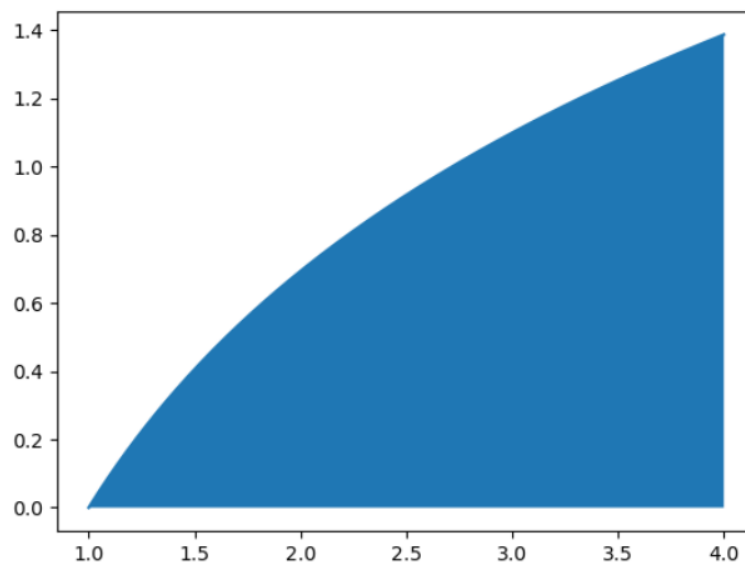


Рисунок 12. Третье индивидуальное задание

13. Построили график функции $f(x, y) = \sin(x) \cdot \cos(y)$ в диапазонах $x, y \in [-\pi, \pi]$.

```

from mpl_toolkits.mplot3d import Axes3D
x = np.linspace(-np.pi, np.pi, 100)
y = np.linspace(-np.pi, np.pi, 100)
X, Y = np.meshgrid(x, y)
Z = np.sin(X) * np.cos(Y)
fig = plt.figure()
funct = fig.add_subplot(111, projection='3d')
color_map = funct.plot_surface(X, Y, Z, cmap='viridis')
fig.colorbar(color_map, shrink=0.5, aspect=5)
funct.set_title("$f(x, y) = \sin(x) * \cos(y)$")
funct.set_xlabel('x')
funct.set_ylabel('y')
funct.set_zlabel('z')

```

Text(0.5, 0.5, 'z')

$$f(x, y) = \sin(x) * \cos(y)$$

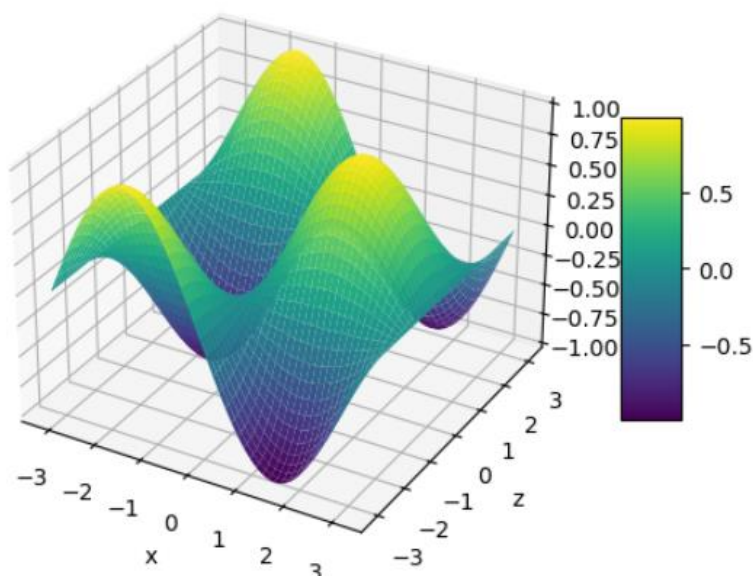


Рисунок 13. Четвертое индивидуальное задание

Ответы на контрольные вопросы:

1. Как осуществляется установка пакета matplotlib?

Через команду `pip install matplotlib`.

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

Через команду `%matplotlib inline`.

3. Как отобразить график с помощью функции plot?

Через команду `plt.plot` или `plt.show()`.

4. Как отобразить несколько графиков на одном поле?

Используется передача нескольких наборов данных в `plt.plot()` или метод `plt.subplot()`.

Разница между методами:

- В первом случае оба графика отображаются в одном поле.
- Во втором случае создаются две области (subplot).

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

Для построения диаграмм категориальных данных используется метод `bar()`.

6. Какие основные элементы графика Вам известны?

Основные элементы графика `matplotlib`:

- Оси (Axes) – координатные оси графика.
- Функция (Plot) – линия, соединяющая точки.
- Заголовок (Title) – название графика.
- Подписи осей (`xlabel`, `ylabel`) – названия осей X и Y.
- Легенда (Legend) – описание кривых на графике.
- Сетка (Grid) – вспомогательные линии для удобства чтения данных.
- Маркировка (Ticks) – деления на осях X и Y.

7. Как осуществляется управление текстовыми надписями на графике?

Управление текстом выполняется с помощью следующих функций:

- `plt.xlabel("Название оси X", fontsize=«Размер»)` – подпись оси X.
- `plt.ylabel("Название оси Y", fontsize=«Размер»)` – подпись оси Y.
- `plt.title("Заголовок", fontsize=«Размер»)` – заголовок графика.
- `plt.text(x, y, "Текст", fontsize=«Размер»)` – произвольный текст в точке

(x, y).

8. Как осуществляется управление легендой графика?

Легенда добавляется с помощью `plt.legend()` и имеет несколько параметров:

- `loc="upper left"` – положение (верхний левый угол).
- `fontsize=10` – размер шрифта.
- `title="Обозначения"` – заголовок легенды.
- `title_fontsize=12` – размер шрифта заголовка

9. Как задать цвет и стиль линий графика?

Цвет и стиль задаются аргументами `color`, `linestyle`, `marker` в `plt.plot()`.

10. Как выполнить размещение графика в разных полях?

Размещение нескольких графиков выполняется с помощью `plt.subplots()`.

11. Как выполнить построение линейного графика с помощью `matplotlib`?

Линейный график строится с помощью функции `plot()`:

Построение графика,

`plt.plot(x, y, color='blue', linestyle='-', marker='o', label="Линейный график")`.

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

Используется `fill_between()`:

Например: `fill_between(x, y, y2, where=(y > y2))` – заливает область между $\sin(x)$ и $\cos(x)$, где $\sin(x) > \cos(x)$.

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

Можно использовать `where` в `fill_between()`:

Например:

– `where=(y > 0)` – заливает область только там, где $y > 0$ (зеленым).

– `where=(y < 0)` – заливает область только там, где $y < 0$ (красным).

14. Как выполнить двухцветную заливку?

`plt.fill_between(x, y, where=(y > 0), color='blue', alpha=0.3)` # Верхняя часть.

`plt.fill_between(x, y, where=(y < 0), color='red', alpha=0.3)` # Нижняя часть.

15. Как выполнить маркировку графиков?

Маркировка (метки) выполняется с помощью `text()` и `annotate()` (добавляет метку с указателем).

16. Как выполнить обрезку графиков?

Графики можно обрезать с помощью `xlim()` и `ylim()`.

Например:

- `plt.plot(x, y)`
- `plt.xlim(2, 8)` # Ограничение оси X от 2 до 8.

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

Ступенчатый график строится с помощью `plt.step()`. Он отображает изменения дискретных данных, где значения остаются постоянными до определенного момента.

Особенности:

- Значение остается постоянным до изменения.
- Используется для дискретных процессов (например, квантованные сигналы, изменение состояний).
- Параметр `where` задает тип отрисовки ('pre', 'mid', 'post').

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

Стековый (Stacked) график – это несколько графиков, сложенных друг на друга (`plt.stackplot()`). Используется для отображения накопленных данных.

Особенности:

- Значения накапливаются, создавая суммарный эффект.
- Позволяет видеть вклад каждой категории в общий результат.
- Часто используется в экономике, финансах, статистике.

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

Stem-график (диаграмма стебель-лист) (`plt.stem()`) – это вертикальные линии, обозначающие дискретные точки.

Особенности:

- Подходит для дискретных данных.
- Используется для анализов сигналов, цифровых измерений.
- Показывает изолированные значения вместо непрерывной линии.

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

Точечный график (scatter plot) (`plt.scatter()`) показывает распределение точек без соединения линиями.

Особенности:

- Используется для анализа взаимосвязей между переменными.
- Позволяет разброс точек по координатам.
- Полезен в статистике, машинном обучении.

21. Как осуществляется построение столбчатых диаграмм с помощью `matplotlib`?

Столбчатая диаграмма (`plt.bar()`) отображает категориальные данные.

Например:

Данные

```
categories = ["A", "B", "C", "D"]
```

```
values = [15, 30, 45, 10]
```

Построение столбчатой диаграммы

```
plt.bar(categories, values, color=['blue', 'green', 'red', 'purple'])
```

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

Групповая столбчатая диаграмма используется для сравнения нескольких категорий внутри одной группы. В `matplotlib` это достигается с помощью `plt.bar()` с разными смещениями по оси X.

`errorbar` показывает разброс данных с указанием доверительных интервалов или погрешностей.

23. Как выполнить построение круговой диаграммы средствами `matplotlib`?

Круговая диаграмма (`plt.pie()`) используется для отображения долей категорий.

`autopct='%'` – добавляет проценты.

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

Цветовая карта (colormap) – это градиент цветов, используемый для отображения данных на графиках (heatmap, 3D-графики, scatter и др.).

25. Как отобразить изображение средствами matplotlib?

Для отображения изображений используется plt.imshow(). Чаще всего изображения загружаются через PIL или OpenCV.

- mimg.imread('image.jpg') – загружает изображение.
- imshow(img) – отображает изображение.
- axis("off") – скрывает оси.

Дополнительно:

Если изображение в формате NumPy ((height, width, 3)), его можно напрямую передать в imshow().

26. Как отобразить тепловую карту средствами matplotlib?

Тепловая карта (heatmap) создается с помощью imshow() и цветовой карты cmap.

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

Линейный 3D-график создается с помощью plot3D().

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

Точечный 3D-график создается с помощью scatter3D().

29. Как выполнить построение каркасной поверхности с помощью matplotlib?

Каркасная поверхность (wireframe) используется для отображения структуры 3D-объекта.

plot_wireframe() строит скелетную модель поверхности.

30. Как выполнить построение трехмерной поверхности с помощью matplotlib?

Полная 3D-поверхность (plot_surface()) строится с заливкой цветом.

Вывод: в ходе лабораторной работы были исследованы базовые возможности библиотеки NumPy языка программирования Python.