

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

Кафедра инфокоммуникаций

**Отчет
по лабораторной работе №11
«Основы работы с пакетом matplotlib»
по дисциплине:
«Введение в системы искусственного интеллекта»**

Выполнил: студент группы ИВТ-б-о-18-1 (2)
Дрищёв Данила Николаевич

_____ (подпись)

Проверил:

Воронкин Роман Александрович

_____ (подпись)

Ставрополь, 2022 г.

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

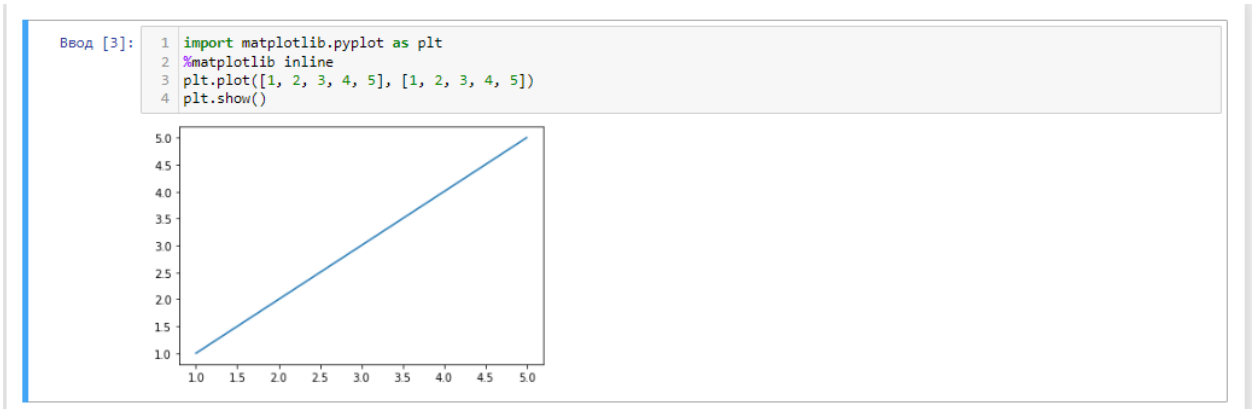
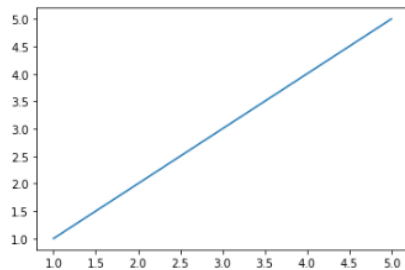


Рисунок 1 – Пример

```
Ввод [4]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
5 plt.show()
6 # Независимая (x) и зависимая (y) переменные
7 x = np.linspace(0, 10, 50)
8 y = x
9 # Построение графика
10 plt.title("Линейная зависимость y = x") # заголовок
11 plt.xlabel("x") # ось абсцисс
12 plt.ylabel("y") # ось ординат
13 plt.grid() # включение отображение сетки
14 plt.plot(x, y) # построение графика
```



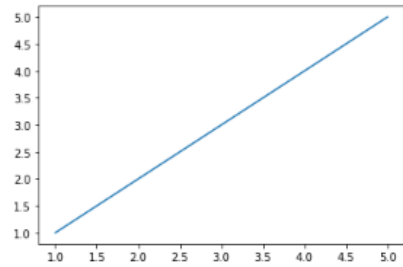
Out[4]: [<matplotlib.lines.Line2D at 0x256e1e78490>]

Рисунок 2 – Пример

```

Ввод [5]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
5 plt.show()
6 # Построение графика
7 plt.title("Линейная зависимость y = x") # заголовок
8 plt.xlabel("x") # ось абсцисс
9 plt.ylabel("y") # ось ординат
10 plt.grid() # включение отображение сетки
11 plt.plot(x, y, "r--") # построение графика

```



Out[5]: [<matplotlib.lines.Line2D at 0x256e1f59490>]

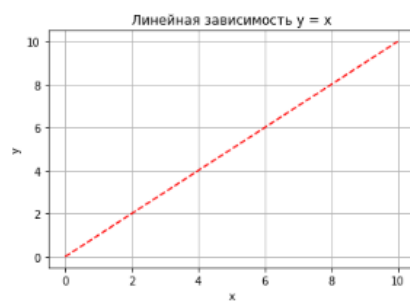
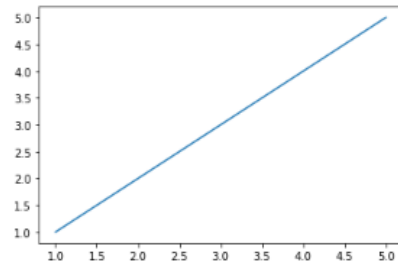


Рисунок 3 – Пример

```

Ввод [8]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
5 plt.show()
6 # Линейная зависимость
7 x = np.linspace(0, 10, 50)
8 y1 = x
9 # Квадратичная зависимость
10 y2 = [i**2 for i in x]
11
12 # Построение графика
13 plt.title("Зависимости: y1 = x, y2 = x^2") # заголовок
14 plt.xlabel("x") # ось абсцисс
15 plt.ylabel("y1, y2") # ось ординат
16 plt.grid() # включение отображения сетки
17 plt.plot(x, y1, x, y2) # построение графика

```



```

Out[8]: [<matplotlib.lines.Line2D at 0x256e203cd60>,
<matplotlib.lines.Line2D at 0x256e203cd90>]

```

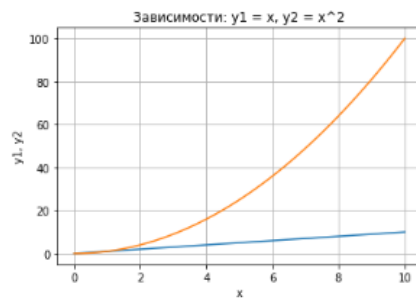


Рисунок 4 – Пример

```

Ввод [9]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
5 plt.show()
6 # Линейная зависимость
7 x = np.linspace(0, 10, 50)
8 y1 = x
9 # Квадратичная зависимость
10 y2 = [i**2 for i in x]
11 # Построение графиков
12 plt.figure(figsize=(9, 9))
13 plt.subplot(2, 1, 1)
14 plt.plot(x, y1) # построение графика
15 plt.title("Зависимости: y1 = x, y2 = x^2") # заголовок
16 plt.ylabel("y1", fontsize=14) # ось ординат
17 plt.grid(True) # включение отображение сетки
18 plt.subplot(2, 1, 2)
19 plt.plot(x, y2) # построение графика
20 plt.xlabel("x", fontsize=14) # ось абсцисс
21 plt.ylabel("y2", fontsize=14) # ось ординат
22 plt.grid(True)

```

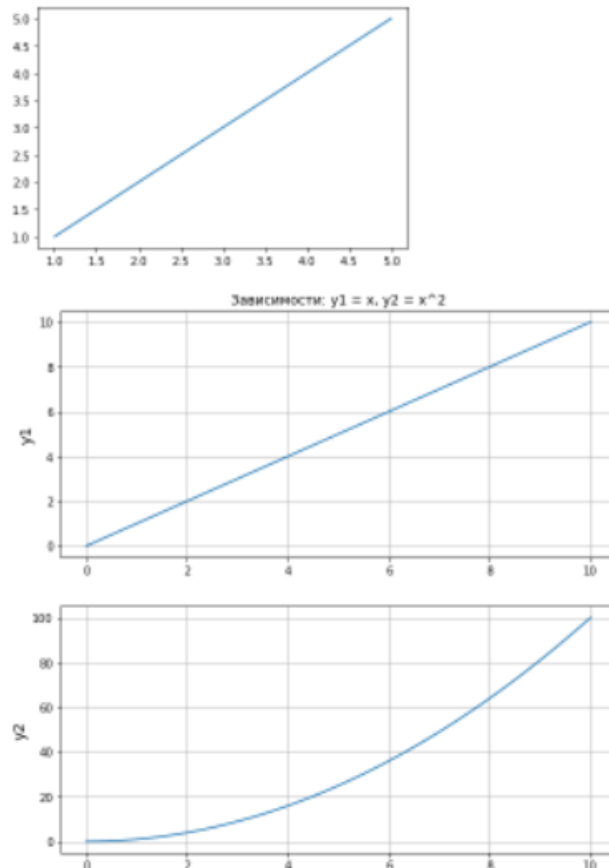


Рисунок 5 – Пример

```

Ввод [11]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3 %matplotlib inline
4 fruits = ["apple", "peach", "orange", "bannana", "melon"]
5 counts = [34, 25, 43, 31, 17]
6 plt.bar(fruits, counts)
7 plt.title("Fruits!")
8 plt.xlabel("Fruit")
9 plt.ylabel("Count")

```

Out[11]: Text(0, 0.5, 'Count')

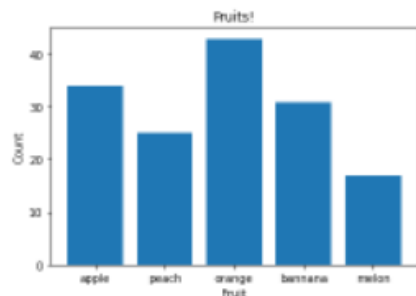


Рисунок 6 – Пример

```

Ввод [12]: 1 import matplotlib.pyplot as plt
2 from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
3 AutoMinorLocator)
4
5
6 import numpy as np
7 x = np.linspace(0, 10, 10)
8 y1 = 4*x
9 y2 = [i**2 for i in x]
10 fig, ax = plt.subplots(figsize=(8, 6))
11 ax.set_title("Графики зависимостей: y1=4*x, y2=x^2", fontsize=16)
12 ax.set_xlabel("x", fontsize=14)
13 ax.set_ylabel("y1, y2", fontsize=14)
14 ax.grid(which="major", linewidth=1.2)
15 ax.grid(which="minor", linestyle="--", color="gray", linewidth=0.5)
16 ax.scatter(x, y1, c="red", label="y1 = 4*x")
17 ax.plot(x, y2, label="y2 = x^2")
18 ax.legend()
19 ax.xaxis.set_minor_locator(AutoMinorLocator())
20 ax.yaxis.set_minor_locator(AutoMinorLocator())
21 ax.tick_params(which='major', length=10, width=2)
22 ax.tick_params(which='minor', length=5, width=1)
23 plt.show()

```

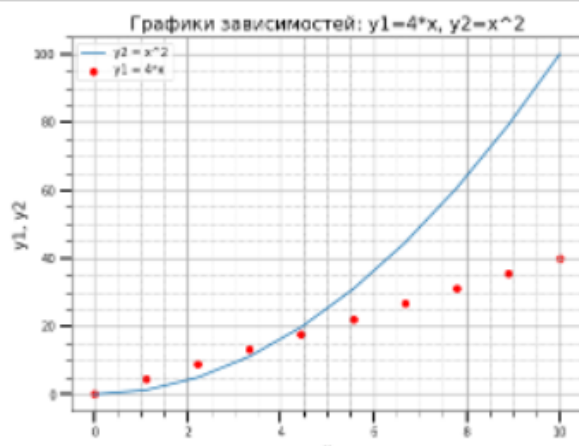


Рисунок 7 – Пример

Ввод [14]:

```

1 #Построение графиков
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 plt.plot()
5 plt.plot([1, 7, 3, 5, 11, 1])

```

Out[14]: [

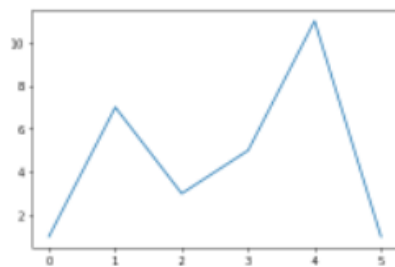


Рисунок 8 – Пример

Ввод [15]:

```

1 #Построение графиков
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 plt.plot()
5 x = [1, 5, 10, 15, 20]
6 y = [1, 7, 3, 5, 11]
7 plt.plot(x, y, label='steel price')
8 plt.title('Chart price', fontsize=15)
9 plt.xlabel('Day', fontsize=12, color='blue')
10 plt.ylabel('Price', fontsize=12, color='blue')
11 plt.legend()
12 plt.grid(True)
13 plt.text(15, 4, 'grow up!')

```

Out[15]: Text(15, 4, 'grow up!')

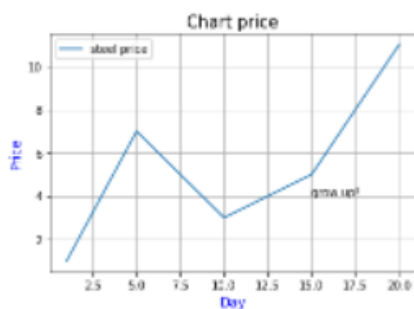


Рисунок 9 – Пример

Ввод [16]:

```

1 #Построение графиков
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 plt.plot()
5 x = [1, 5, 10, 15, 20]
6 y1 = [1, 7, 3, 5, 11]
7 y2 = [i*1.2 + 1 for i in y1]
8 y3 = [i*1.2 + 1 for i in y2]
9 y4 = [i*1.2 + 1 for i in y3]
10 plt.plot(x, y1, '-', x, y2, '--', x, y3, '-.', x, y4, ':')

```

Out[16]: [
 <matplotlib.lines.Line2D at 0x256e2269f40>,
 <matplotlib.lines.Line2D at 0x256e2269f10>,
 <matplotlib.lines.Line2D at 0x256e2275190>]

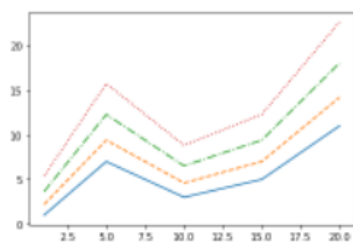


Рисунок 10 – Пример

```

Ввод [18]: 1 #Работа с функцией subplot()
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 # Исходный набор данных
5 x = [1, 5, 10, 15, 20]
6 y1 = [1, 7, 3, 5, 11]
7 y2 = [i*1.2 + 1 for i in y1]
8 y3 = [i*1.2 + 1 for i in y2]
9 y4 = [i*1.2 + 1 for i in y3]
10 # Настройка размеров подложки
11 plt.figure(figsize=(12, 7))
12 # Вывод графиков
13 plt.subplot(2, 2, 1)
14 plt.plot(x, y1, '-')
15 plt.subplot(2, 2, 2)
16 plt.plot(x, y2, '-')
17 plt.subplot(2, 2, 3)
18 plt.plot(x, y3, '-')
19 plt.subplot(2, 2, 4)
20 plt.plot(x, y4, '-')

```

Out[18]: [matplotlib.lines.Line2D at 0x256e34fa910]

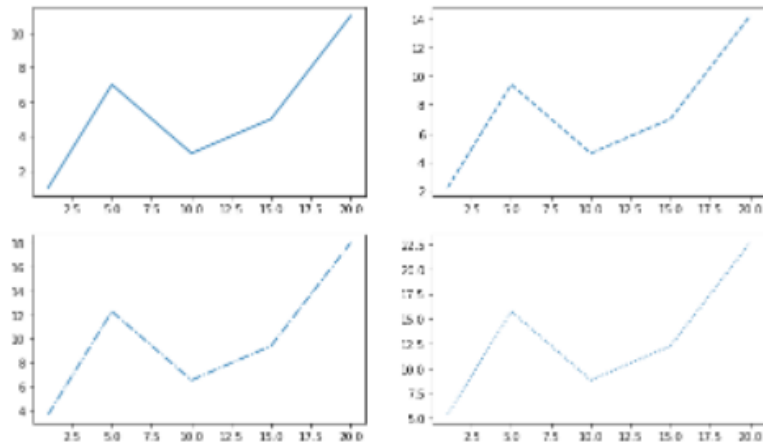


Рисунок 11 – Пример

Вывод: исследовал базовые возможности библиотеки matplotlib языка программирования Python.

Ответы на вопросы:

1. Как осуществляется установка пакета matplotlib?

Сделать это очень просто, достаточно в терминале выполнить команду: `pip install matplotlib` и пакет со всеми зависимостями будет установлен. Также можно зайти в репозиторий: <https://pypi.org> набрать в поиске matplotlib и появится список доступных пакетов. По умолчанию, устанавливается последняя версия.

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

Магическая команда `%matplotlib` настраивает Jupyter Notebook для отображения графиков с помощью Matplotlib. По умолчанию используется стандартный графический бэкенд от Matplotlib, и ваши графики отображаются в отдельном окне. На заметку: Вы можете изменить бэкенд Matplotlib, передав аргумент в магическую команду `%matplotlib`.

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

Команда `plot(y)` строит график элементов одномерного массива `y` в зависимости от номера элемента; если элементы массива `y` комплексные, то строится график `plot(real(y), imag(y))`. Если `Y` - двумерный действительный массив, то строятся графики для столбцов; в случае комплексных элементов их мнимые части игнорируются.

Команда `plot(x, y)` соответствует построению обычной функции, когда одномерный массив `x` соответствует значениям аргумента, а одномерный массив `y` - значениям функции. Когда один из массивов `X` или `Y` либо оба двумерные, реализуются следующие построения:

если массив `Y` двумерный, а массив `x` одномерный, то строятся графики для столбцов массива `Y` в зависимости от элементов вектора `x`;

если двумерным является массив `X`, а массив `y` одномерный, то строятся графики столбцов массива `X` в зависимости от элементов вектора `y`;

если оба массива `X` и `Y` двумерные, то строятся зависимости столбцов массива `Y` от столбцов массива `X`.

Команда `plot(x, y, s)` позволяет выделить график функции, указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной `s`.

4. Как отобразить несколько графиков на одном поле?

Для того чтобы отобразить несколько независимых графиков в одном окне - предназначена функция `subplot()` из пакета `pylab`.

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

Seaborn — библиотека для создания статистических графиков на Python. Она построена на основе matplotlib и тесно интегрируется со структурами данных pandas. Seaborn помогает вам изучить и понять данные. Его функции построения графиков работают с датасетами и выполняют все необходимые преобразования для создания информативных графиков.

6. Какие основные элементы графика Вам известны?

Основные элементы графика следующие: поле графика, геометрические знаки, пространственные ориентиры, масштаб, экспликация графика.

Поле графика - пространство, в котором размещаются геометрические знаки, образующие график. Он характеризуется форматом и соотношением сторон.

7. Как осуществляется управление текстовыми надписями на графике?

В части текстового наполнения при построении графика выделяют следующие составляющие:

заголовок поля (title);

заголовок фигуры (suptitle);

подписи осей (xlabel, ylabel);

тестовый блок на поле графика (text), либо на фигуре (figtext);

аннотация (annotate) – текст и указатель.

8. Как осуществляется управление легендой графика?

Для отображения легенды на графике используется функция legend()

9. Как задать цвет и стиль линий графика?

Чтобы задать цвет закрашивания графического объекта в python используется команда `obj.setFill ("цвет")`. Пример программы на Python, которая рисует закрашенную синюю окружность `from graphics import* win = GraphWin ("Окно для графики", 400, 400) obj = Circle (Point (200, 200), 50) obj.setFill ("blue") obj.draw (win) win.getMouse () win.close ()`.

10. Как выполнить размещение графика в разных полях?

Самый простой способ представить графики в отдельных полях – это использовать функцию `subplot()` для задания их мест размещения.