

Лабораторна робота №2

Вступ до numpy і matplotlib

Для продовження роботи необхідне знайомство зі стандартними типами python, такими як списки та словники. У цьому зошиті надано вправи для того, щоб навчитися працювати з масивами (матрицями) numpy, та зручні способи виконання операцій над ними.

Розглянемо рівняння для прямої $y = mx + c$, де m позначає нахил, а c - зсув. Нехай необхідно вирішити задачу лінійної регресії, у якій m і c - це невідомі параметри, які визначаються з даних. Тоді m замінюється на β_1 і c на β_0 , де β_i позначає параметр, який потрібно визначити. Таким чином, формула прямої лінії буде записана $y = \beta_1x + \beta_0$ або $y = \beta_0 + \beta_1x$.

Основне завдання, яке потрібно виконати в цьому зошиті, полягає у тому, що необхідно задати послідовність чисел (кілька значень x) і створити послідовність виводу (кілька значень y). Нехай нахил лінії 5, а зсув дорівнює -1.

```
In [1]: m = 5
        c = -1
        x = [0, 1, 2, 3, 4, 5, 6]
```

Будемо зберігати наші декілька значень x у Python-списку.

```
In [2]: # Завдання: Сформувати вихідний список значень y
        # запишіть код формування тут:
        y = [m * x_i + c for x_i in x]
```

```
In [3]: print(y)

[-1, 4, 9, 14, 19, 24, 29]
```

Хоча списки характерні саме для мови Python, але вони громіздкі при виконанні масових чисельних чи матричних операцій.

numpy

Виконати те ж саме завдання, використовуючи numpy.

```
In [4]: # Завдання: імпортуємо NumPy
        # запишіть код формування масиву тут:
        m = 5
        c = -1
        x = [i for i in range(50)]
```

```
In [5]: import numpy as np
        X = np.array(x)
        Y = m*X + c
```

```
In [6]: print(Y)

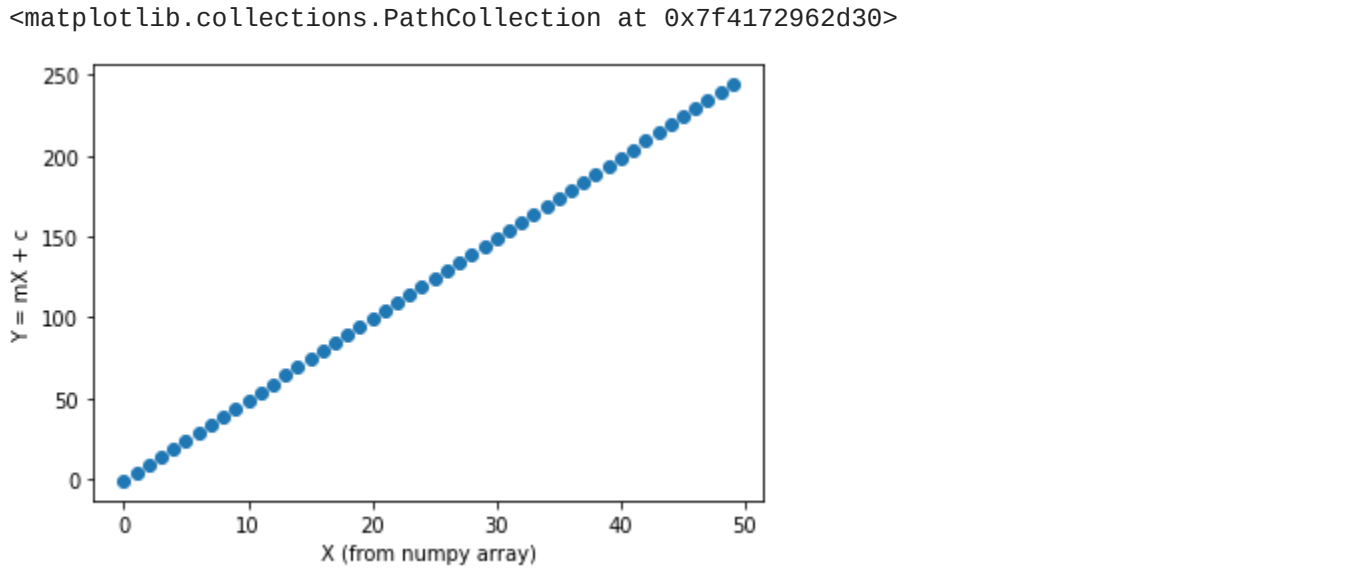
[ -1   4   9  14  19  24  29  34  39  44  49  54  59  64  69  74  79  84
  89  94  99 104 109 114 119 124 129 134 139 144 149 154 159 164 169 174
 179 184 189 194 199 204 209 214 219 224 229 234 239 244]
```

matplotlib

Тепер у нас є відповідні вхідні та вихідні значення, X та Y . Для наглядності було б добре їх бачити на графіку. Хоча функція plot дуже корисна для послідовностей, у нашому випадку тут маємо послідовність точок, які лежать на прямій. Це означає, що на простому графіку лінії неможливо визначити, чи є одна пряма лінія між двома кінцевими точками, чи це є ряд точок. Показати окремі точки більш інформативно, тому тут ми будемо використовувати scatter plot, а не line plot.

```
In [7]: import matplotlib.pyplot as plt
```

```
In [8]: # Завдання: Створіть scatter plot для X і Y
        #Підказка: або зверніться до документації matplotlib, або після імпорту pyplot використайте
        # Бажано знайти відповідну scatter plot функцію
        # Пишіть код програми тут:
        chart = plt.scatter(X, Y)
        plt.xlabel("X (from numpy array)")
        plt.ylabel("Y = mX + c")
        chart
```



Висновок

Після успішного завершення цього зошита ми набули необхідних навичок користування Jupyter notebook для розробки послідовності маніпуляцій з даними, які включають описовий відформатований текст. Застосували NumPy і скористалися типами даних і операціями, які дозволяють швидко виконувати операції з даними. Саме тому Python та його екосистема чудово підходить для обробки та прикладного аналізу наукових даних. Бібліотека Pandas, яку ми будемо застосовувати, є обгорткою модуля NumPy. В ході виконання завдання застосовано візуалізацію для даних, яку ми також будемо використовувати в наступних зошитах.

```
In [ ]:
```