

## Лабораторна робота № 4

Лінійна регресія. Метод найменших квадратів. Інтерполяція

Мета роботи: Опрацювати поняття «лінійна регресія» і дослідити метод найменших квадратів та набути навички роботи в середовищі Python.

Хід роботи

**Завдання 1.** Ретельно опрацювати теоретичні відомості з лекційного курсу

### Завдання 2.

Експериментально отримані N-значень величини Y при значеннях величини X. Відшукати параметри функції за методом найменших квадратів.

Побудувати графіки, де в декартовій системі координат нанести експериментальні точки і графік апроксимуючої функції.

Варіантів всього 15 а в мене 27, тому  $27 - 15 =$  Варіант 12

12	X	13,33	21	63,75	20,87	40,42	30,27
	Y	10,48	21,03	23,02	41,25	27,16	51,5

Лістинг:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Експериментальні дані
X = np.array([13.33, 21, 63.75, 20.87, 40.42, 30.27])
Y = np.array([10.48, 21.03, 23.02, 41.25, 27.16, 51.5])

# Визначаємо лінійну функцію для апроксимації
def linear_func(x, a, b):
    return a * x + b

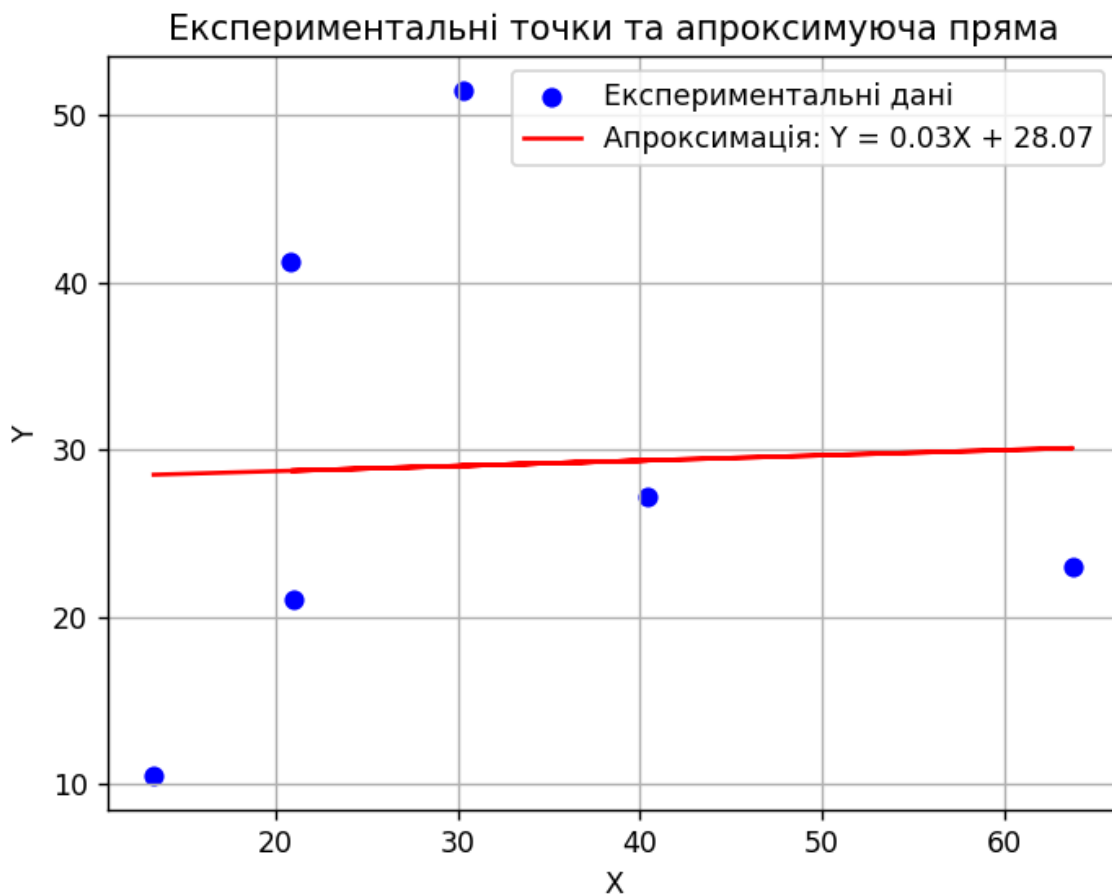
# Застосовуємо метод найменших квадратів для знаходження параметрів a і b
params, _ = curve_fit(linear_func, X, Y)
a, b = params

# Створюємо графік
plt.scatter(X, Y, color='blue', label='Експериментальні дані') # Експериментальні
# точки
```

```
plt.plot(X, linear_func(X, a, b), color='red', label=f'Апроксимація: Y = {a:.2f}X + {b:.2f}') # Лінійна апроксимація

# Додаємо підписи та легенду
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Експериментальні точки та апроксимуюча пряма')
plt.grid(True)
plt.show()
```

Figure 1



### Завдання № 3:

Виконати інтерполяцію функції, задану в табличній формі в п'яти точках (див. нижче). Розрахунки виконати в середовищі Python.

Вектори даних:

$$x := \begin{pmatrix} 0.1 \\ 0.3 \\ 0.4 \\ 0.6 \\ 0.7 \end{pmatrix} \quad y := \begin{pmatrix} 3.2 \\ 3 \\ 1 \\ 1.8 \\ 1.9 \end{pmatrix}$$

Алгоритм розв'язку завдання № 3:

1. Заповнення матриці **X**;
2. Отримання коефіцієнтів інтерполяційного полінома;
3. Визначення функції полінома (прийняти поліном степеню 4);
4. Побудова графіка функції для інтерполюючого полінома;
5. Визначити значення функції в проміжних точках зі значеннями 0,2 і 0,5.

Для реалізації обчислювальних алгоритмів рекомендується використання онлайн середовищ тестування (наприклад repl.it, trinket, і.т.д.)

Захист лабораторної роботи передбачає виконання практичних завдань поставлених в роботі, та виконання завдань теоретичного характеру.

Лістинг:

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

# Вектори даних
x = np.array([0.1, 0.3, 0.4, 0.6, 0.7])
y = np.array([3.2, 3, 1, 1.8, 1.9])

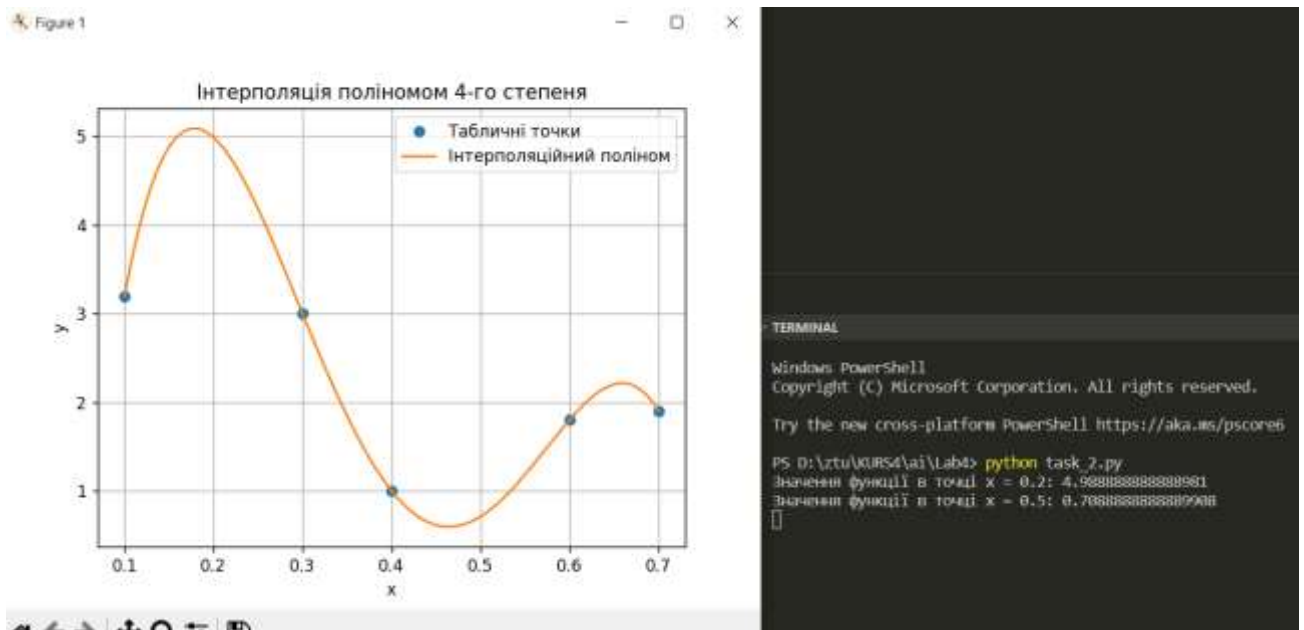
# Знаходимо коефіцієнти інтерполяційного полінома 4-го степеня
coefficients = np.polyfit(x, y, 4)
polynomial = np.poly1d(coefficients)

# Визначення значень функції в точках 0.2 та 0.5
y_02 = polynomial(0.2)
y_05 = polynomial(0.5)

# Виведемо значення функції у проміжних точках
print(f"Значення функції в точці x = 0.2: {y_02}")
print(f"Значення функції в точці x = 0.5: {y_05}")
```

```
# Побудова графіка інтерполюючого полінома
x_range = np.linspace(0.1, 0.7, 100) # діапазон для графіку
y_range = polynomial(x_range)

plt.plot(x, y, 'o', Label='Табличні точки') # експериментальні точки
plt.plot(x_range, y_range, '-', Label='Інтерполяційний поліном') # графік полінома
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('Інтерполяція поліномом 4-го степеня')
plt.grid(True)
plt.show()
```



GIT: [https://github.com/Alhim616/AI\\_Labs\\_Yanushevych](https://github.com/Alhim616/AI_Labs_Yanushevych)