

## 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.

**Час виконання:** 4 години

### Зміст 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.

Варіант 17 = Варіант 2

2	X	-1	-1	0	1	2	3
	Y	-1	0	1	1	3	5

Лістинг:

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

X = np.array([-1, -1, 0, 1, 2, 3])
Y = np.array([-1, 0, 1, 1, 3, 5])

def linear_func(x, a, b):
    return a * x + b

params, _ = curve_fit(linear_func, X, Y)
a, b = params

X_smooth = np.linspace(X.min() - 5, X.max() + 5, 500)
Y_smooth = linear_func(X_smooth, a, b)

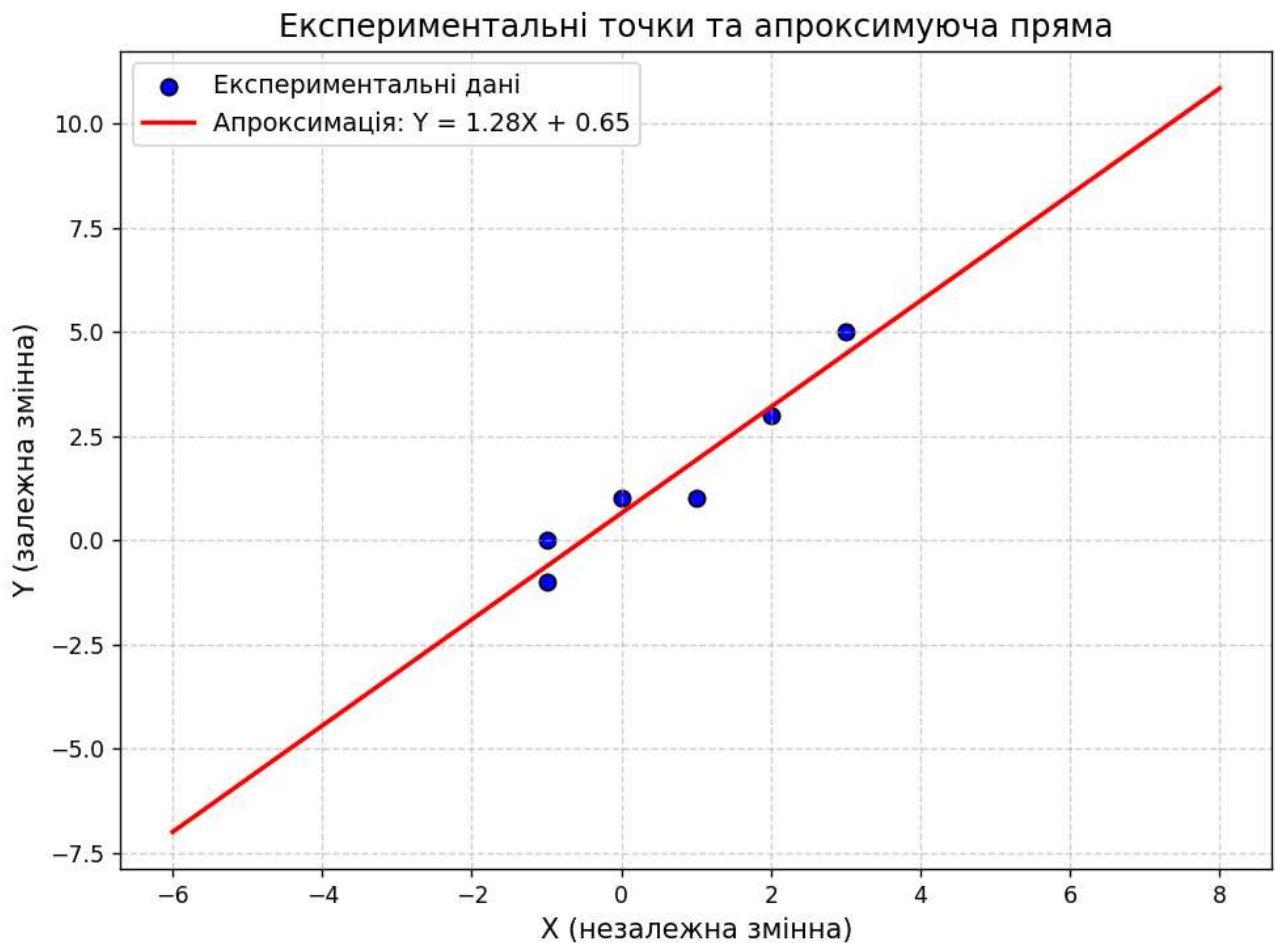
plt.figure(figsize=(8, 6))
plt.scatter(X, Y, color='blue', s=50, edgecolor='black', label='Експериментальні дані')
```

```
plt.plot(X_smooth, Y_smooth, color='red', linewidth=2, label=f'Апроксимація:  $Y = {a:.2f}X + {b:.2f}$ ')

plt.xlabel('X (незалежна змінна)', fontsize=12)
plt.ylabel('Y (залежна змінна)', fontsize=12)
plt.title('Експериментальні точки та апроксимуюча пряма', fontsize=14)
plt.legend(fontsize=11)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()

plt.show()
```

Результат виконання роботи:



### 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.

Лістинг:

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

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

coefficients = np.polyfit(x, y, 4)
polynomial = np.poly1d(coefficients)

y_02 = polynomial(0.2)
y_05 = polynomial(0.5)

print("Коефіцієнти полінома:", coefficients)
print(f"Значення функції в точці x = 0.2: {y_02:.3f}")
print(f"Значення функції в точці x = 0.5: {y_05:.3f}")

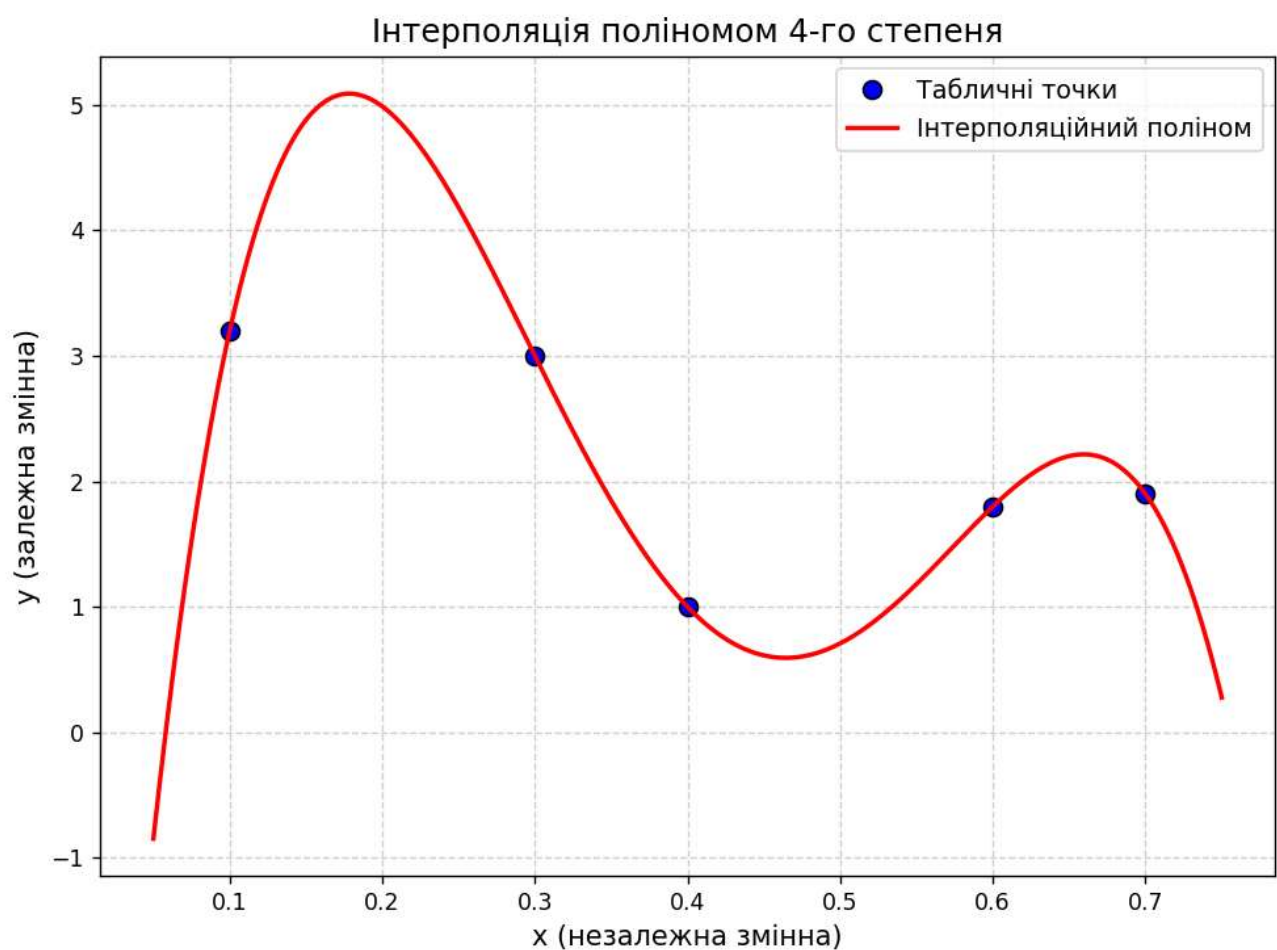
x_range = np.linspace(min(x) - 0.05, max(x) + 0.05, 500)
y_range = polynomial(x_range)

plt.figure(figsize=(8, 6))
plt.plot(x, y, 'o', color='blue', markersize=8, label='Табличні точки',
markededgecolor='black')
plt.plot(x_range, y_range, '-', color='red', linewidth=2, label='Інтерполяційний поліном')
```

```
plt.xlabel('x (незалежна змінна)', fontsize=12)
plt.ylabel('y (залежна змінна)', fontsize=12)
plt.title('Інтерполяція поліномом 4-го степеня', fontsize=14)
plt.legend(fontsize=11)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()

plt.show()
```

Результат виконання роботи:



Git: <https://github.com/PavlenkoOks/AI>