

Лабораторна №2

Лінійна регресія. Метод найменших квадратів. Інтерполяція

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

Хід роботи

Варіант 26(11)

2. Побудова лінійної регресії

Згідно з варіантом для побудови моделі використано наступні експериментальні дані:

11	X	28	14	54	16	22	15
	Y	-15	10	4	5	11	28

Для розрахунків використано бібліотеку scikit-learn клас LinearRegression.

Програма автоматично розрахувала коефіцієнти прямої, що мінімізує квадратичне

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

print("--- ЗАВДАННЯ 2: Лінійна Регресія (Варіант 11) ---")

# Дані варіанта 11
X_train = np.array([28, 14, 54, 16, 22, 15]).reshape(-1, 1)
y_train = np.array([-15, 10, 4, 5, 11, 28])

# Модель лінійної регресії
model = LinearRegression()
model.fit(X_train, y_train)

b0 = model.intercept_
b1 = model.coef_[0]

print(f"Рівняння регресії: y = {b0:.2f} + {b1:.2f}x")
print(f"Коефіцієнти: b0 = {b0:.4f}, b1 = {b1:.4f}")

y_pred = model.predict(X_train)

plt.figure(figsize=(12, 5))

# Графік регресії
plt.subplot(1, 2, 1)
plt.scatter(X_train, y_train, color='red', label='Експериментальні точки')
```

Зм.

Рс

Перевір.

Реценз.

Н. Контр.

Зав.каф.

Звіт з лабораторної
роботи №1

ФІКТ, гр. ІПЗ-22-3

```

plt.plot(X_train, y_pred, color='blue', linewidth=2, label='Регресія')
plt.title("Завдання 2: Лінійна регресія (Вар. 11)")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.grid(True)

print("\n--- ЗАВДАННЯ 3: Інтерполяція ---")

# Дані для інтерполяції (не змінюються)
x_interp = np.array([0.1, 0.3, 0.4, 0.6, 0.7])
y_interp = np.array([3.2, 3.0, 1.0, 1.8, 1.9])

# Поліном 4-го степеня
coefficients = np.polyfit(x_interp, y_interp, 4)
poly_func = np.poly1d(coefficients)

print("Коефіцієнти полінома (від найвищого степеня):")
print(coefficients)
print("\nРівняння полінома:")
print(poly_func)

# Перевірка точок
check_points = [0.2, 0.5]
print("\nРезультати перевірки точок:")
for pt in check_points:
    val = poly_func(pt)
    print(f" -> Значення функції в точці x={pt}: y={val:.4f}")

# Графік інтерполяції
plt.subplot(1, 2, 2)
x_smooth = np.linspace(min(x_interp), max(x_interp), 100)
y_smooth = poly_func(x_smooth)

plt.plot(x_smooth, y_smooth, color='green', label='Інтерполяційний поліном')
plt.scatter(x_interp, y_interp, color='red', s=50, zorder=5, label='Вузли інтерполяції')
plt.scatter(check_points, poly_func(check_points), color='blue', marker='x', s=100, zorder=5, label='Точки 0.2 і 0.5')

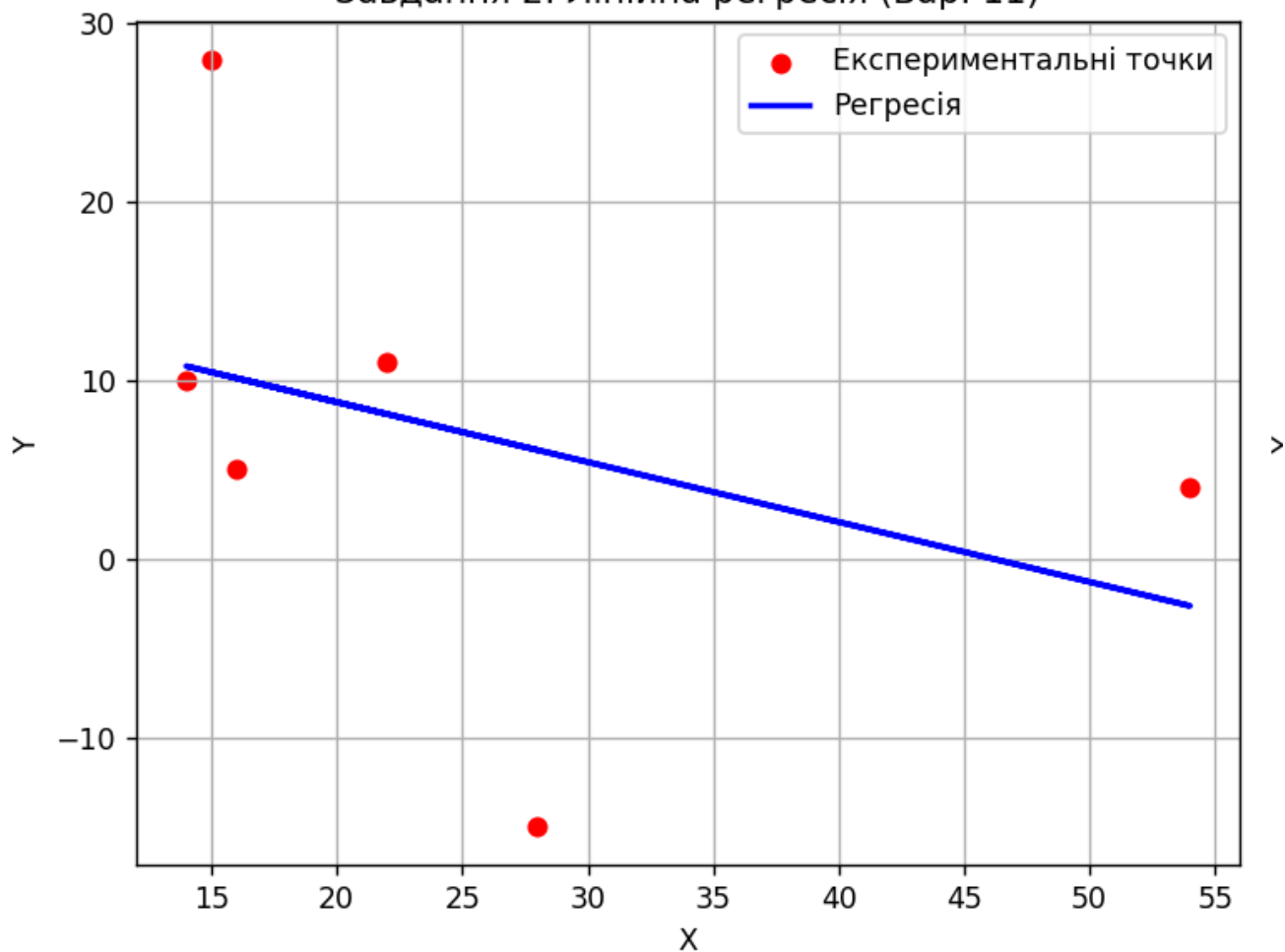
plt.title("Завдання 3: Інтерполяція")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()

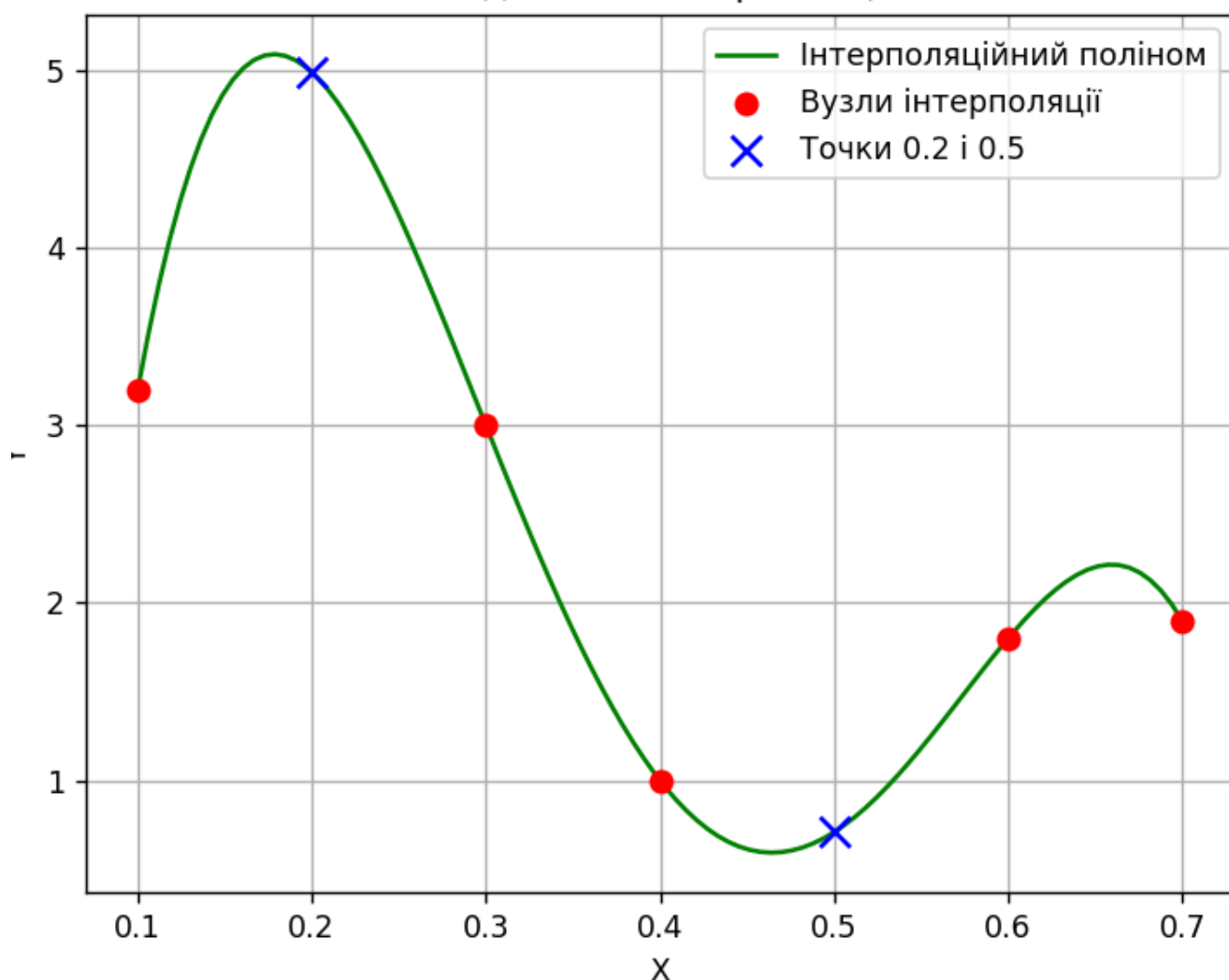
```

		Чайковський А.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1	Арк.
		Маєвський О				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2: Лінійна регресія (Вар. 11)



Завдання 3: Інтерполяція



--- ЗАВДАННЯ 2: Лінійна Регресія (Варіант 11) ---

Рівняння регресії: $y = 15.51 + -0.34x$

Коефіцієнти: $b_0 = 15.5062$, $b_1 = -0.3358$

--- ЗАВДАННЯ 3: Інтерполяція ---

Коефіцієнти полінома (від найвищого степеня):

$[-852.77777778 \quad 1480.55555556 \quad -864.02777778 \quad 186.25 \quad -8.18]$

Рівняння полінома:

$-852.8 x^4 + 1481 x^3 - 864 x^2 + 186.3 x - 8.18$

Результати перевірки точок:

-> Значення функції в точці $x=0.2$: $y=4.9889$

-> Значення функції в точці $x=0.5$: $y=0.7089$

□

		Чайковський А.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1	Арк.
		Маєвський О				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: Під час виконання лабораторної роботи я ознайомився з методами машинного навчання для задач регресії.

Github- [AntonChaikovskiy/AI-systems](https://github.com/AntonChaikovskiy/AI-systems)

		Чайковський А.В			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1	Арк.
		Маєвський О				5
Змн.	Арк.	№ докум.	Підпис	Дата		