**Міністерство освіти та науки України**

**Київський національний економічний університет**

**імені Вадима Гетьмана**

**Кафедра інформаційних систем**

**в економіці**

Дисципліна **“Системи і методи штучного інтелекту”**

**ЗВІТ**

**з лабораторної роботи №4.2**

**“Лінійна регресія. Метод найменших квадратів. Інтерполяція”**

**Підготував:**

студент групи ІН-401

спеціальності “**Комп’ютерні науки**”

Михайлик В.А.

Київ – 2025

**Хід роботи**

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

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

У цьому завданні було опрацьовано базові теоретичні засади машинного навчання, зокрема — методи регресії, як один із видів задач контрольованого навчання. Основна увага приділялася простій лінійній регресії, яка дозволяє моделювати залежність цільової змінної від однієї незалежної змінної. Було показано, як побудувати таку модель, використовуючи метод найменших квадратів — через мінімізацію суми квадратів відхилень фактичних значень від передбачених.

Розглянутий приклад з чотирма точками даних ілюструє практичне застосування цього підходу: розраховано коефіцієнти β₀ і β₁, що формують регресійну пряму. Побудована модель дозволяє робити кількісні прогнози і є основою для більш складних моделей, таких як поліноміальна регресія або нейронні мережі.

Таким чином, вивчений матеріал формує фундамент для розуміння принципів побудови та навчання регресійних моделей у задачах прогнозування в машинному навчанні.

**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.

**17 номер у журналі – 2 варіант**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

# Вхідні експериментальні дані для аналізу

X = np.array([-1, -1, 0, 1, 2, 3]).reshape(-1, 1)  # Незалежна змінна

Y = np.array([-1, 0, 1, 1, 3, 5])  # Залежна змінна

# Ініціалізація та тренування моделі методом найменших квадратів

model = LinearRegression()

model.fit(X, Y)

# Визначення коефіцієнтів лінійної апроксимації

a = model.coef\_[0]

b = model.intercept\_

print(f"Апроксимуюча функція: Y = {a:.2f}X + {b:.2f}")

# Прогнозування Y для побудови прямої

Y\_pred = model.predict(X)

# Побудова графіка

plt.scatter(X, Y, color='red', label='Експериментальні точки')

plt.plot(X, Y\_pred, color='blue', label=f'Апроксимуюча пряма: Y = {a:.2f}X + {b:.2f}')

plt.xlabel('X')

plt.ylabel('Y')

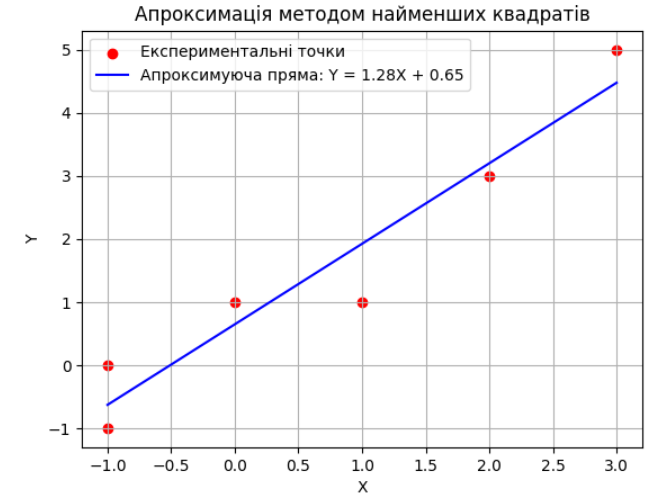
plt.title('Апроксимація методом найменших квадратів')

plt.legend()

plt.grid(True)

plt.show()

****

****

Код виконує побудову простої лінійної регресійної моделі на основі невеликого набору експериментальних даних, де незалежна змінна X=[−1,−1,0,1,2,3], а залежна — Y=[−1,0,1,1,3,5]. За допомогою методу найменших квадратів було знайдено коефіцієнти прямої регресії Y=aX+b, яка найкраще описує залежність між цими змінними. Отримане рівняння відображає загальну тенденцію зміни Y зі збільшенням X.

На побудованому графіку червоними точками позначені вихідні експериментальні значення, що демонструють певну варіацію навколо прямої — особливо видно відхилення у точці (1, 1), яка дещо вибивається із загального тренду. Синя пряма, зображена на графіку, є результатом апроксимації: вона проходить якомога ближче до всіх точок, мінімізуючи відстань (у квадраті) між фактичними значеннями та прогнозованими. Це наочно демонструє, як лінійна модель узагальнює залежність між вхідною та вихідною змінними навіть при наявності певного шуму в даних. Графік також містить легенду, сітку та підписи осей, що полегшує інтерпретацію результатів.

**Завдання 3.** Виконати інтерполяцію функції, задану в табличній формі в п'яти точках. Розрахунки виконати в середовищі Python.

import numpy as np

import matplotlib.pyplot as plt

from numpy.polynomial import Polynomial

# Створюємо експериментальні точки для аналізу

experimental\_points\_x = np.array([0.1, 0.3, 0.4, 0.6, 0.7])

experimental\_points\_y = np.array([3.2, 3.0, 1.0, 1.8, 1.9])

# Розрахунок коефіцієнтів за допомогою методу найменших квадратів

polynomial\_coefficients = np.polyfit(experimental\_points\_x, experimental\_points\_y, deg=4)

interpolation\_polynomial = Polynomial(polynomial\_coefficients[::-1])

print("Знайдені коефіцієнти полінома:")

for index, coef in enumerate(polynomial\_coefficients):

    print(f"Коефіцієнт при x^{4-index} = {coef:.4f}")

# Функція для обчислення значень полінома

def calculate\_polynomial\_value(x):

    return interpolation\_polynomial(x)

# Візуалізація результатів

x\_continuous = np.linspace(0, 0.8, 100)

y\_continuous = calculate\_polynomial\_value(x\_continuous)

plt.figure(figsize=(12, 8))

plt.scatter(experimental\_points\_x, experimental\_points\_y,

           color='blue', s=100, label='Експериментальні точки', zorder=5)

plt.plot(x\_continuous, y\_continuous, 'r-',

         label='Інтерполяційна крива', linewidth=2)

plt.xlabel('Значення x')

plt.ylabel('Значення y')

plt.title('Інтерполяція експериментальних даних')

plt.grid(True, linestyle='--', alpha=0.7)

plt.legend(fontsize=10)

plt.show()

# Перевірка точності інтерполяції

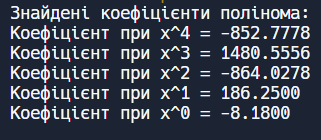
test\_points = np.array([0.2, 0.5])

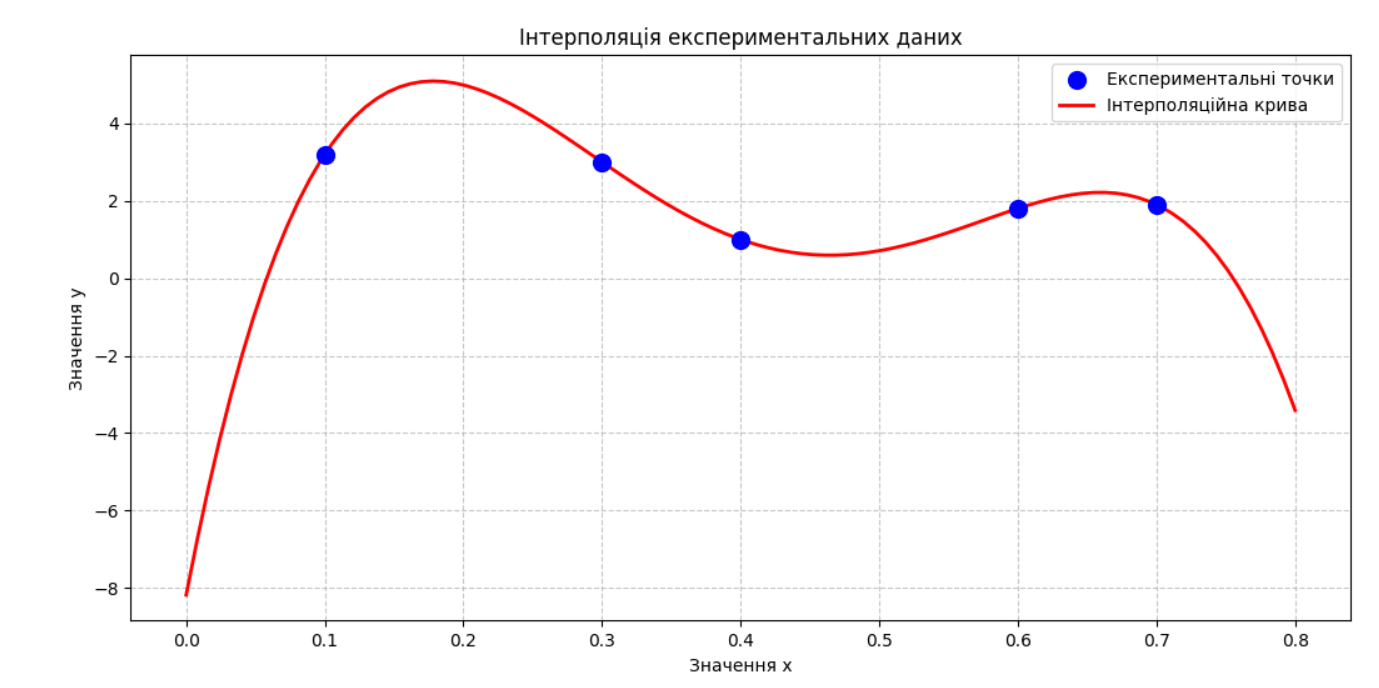
interpolated\_values = calculate\_polynomial\_value(test\_points)

print("\nРезультати обчислень у контрольних точках:")

for x, y in zip(test\_points, interpolated\_values):

    print(f"При x = {x}: f(x) = {y:.4f}")





Даний код реалізує інтерполяцію експериментальних даних за допомогою полінома 4-го степеня. За результатами обчислень було отримано коефіцієнти полінома (від -852.7778 до -8.1800), які дозволяють апроксимувати задані точки. Візуалізація демонструє, що отриманий поліном добре проходить через експериментальні точки, а розрахунки в контрольних точках (x = 0.2 та x = 0.5) дозволяють оцінити значення функції в проміжних положеннях, що є головною метою інтерполяції.

**Висновок:** у ході лабораторної роботи було детально вивчено поняття лінійної регресії та метод найменших квадратів, які є фундаментальними для розв’язання задач прогнозування в машинному навчанні. На основі теоретичних відомостей було засвоєно принципи побудови моделі простої лінійної регресії, включаючи обчислення модельних коефіцієнтів шляхом мінімізації суми квадратів відхилень — тобто реалізації критерію найменших квадратів.

Практична частина передбачала побудову лінійної апроксимації на основі експериментальних даних із подальшою візуалізацією результатів. Це дозволило не лише закріпити навички використання бібліотеки scikit-learn у Python, а й наочно проаналізувати, наскільки ефективно пряма відображає тенденцію зміни залежної змінної. Побудовані графіки продемонстрували, що навіть проста модель може дати адекватні прогнози у випадках, коли залежність між змінними є близькою до лінійної.

Також було розглянуто основи інтерполяції функцій, зокрема поліноміальної, та реалізовано побудову інтерполяційного полінома за заданими вузлами. Це дало змогу порівняти підходи регресії (наближення з урахуванням помилки) й інтерполяції (точне відновлення значень у відомих точках), що є важливим для розуміння точності і гнучкості математичних моделей у прикладних задачах.