МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДВНЗ «КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВАДИМА ГЕТЬМАНА»

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

*Кафедра інформаційних систем в економіці*

Лабораторна робота №4

з дисципліни «Системи і методи штучного інтелекту»

Виконав:

студент 4 курсу, групи ІН-401

Задніпрянець О.Р.

Викладач:

Волошин А.П.

# Київ – 2025

**Лабораторна робота №4**

**Тема:** «Моделювання елементів нечітких множин та формування нечітких правил».

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

**Мій гіт** - <https://github.com/OleksiiZadniprianets/-4>

**Варіант 6**

**Завдання 1.**

Ретельно опрацювати теоретичні відомості з лекційного курсу. За допомогою машинного навчання можна прогнозувати подальші події шляхом аналізу попереднього досвіду . Наприклад, скласти прогноз погоди на завтра, або вгадати курс акцій на біржі, або діагностувати хворобу пацієнта, ґрунтуючись на його попередньої історії хвороби.

Лінійна регресія є одним із найпростіших і водночас найважливіших інструментів у машинному навчанні, який дозволяє будувати математичні моделі для прогнозування числових значень на основі наявних даних. Основна ідея полягає у знаходженні прямої, яка найкраще описує взаємозв’язок між незалежною змінною (входом) і залежною змінною (виходом). Така модель дозволяє робити прогнози, наприклад, визначити температуру на завтра, передбачити вартість акцій або оцінити рівень ризику захворювання.

На відміну від класифікації, де дані розподіляються за категоріями, регресія дозволяє отримувати конкретні числові результати. В основі лінійної регресії лежить припущення, що залежність між змінними можна описати прямою лінією. Хоча це не завжди відповідає дійсності, у багатьох практичних ситуаціях така модель дає достатньо точні та корисні результати, особливо на початковому етапі аналізу.

Щоб побудувати лінійну регресійну модель, використовують метод найменших квадратів — він дозволяє підібрати такі параметри прямої, щоб вона максимально наближено відображала реальні дані. Отриману модель можна застосовувати для прогнозів, коли нові значення вводяться як вхідні дані, а модель генерує відповідне передбачення.

Таким чином, лінійна регресія — це базовий метод, з якого починається знайомство з машинним навчанням, і який закладає основу для розуміння більш складних алгоритмів. Вона має велике прикладне значення та використовується в різних сферах — від економіки й фінансів до медицини та природничих наук.

**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.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

# Твої експериментальні дані

X = np.array([3.33, 1, 63, 0.87, 0.42, 0.27]).reshape(-1, 1)

Y = np.array([0.48, 1.03, 2.02, 4.25, 7.16, 11.5])

# Створення та навчання моделі

model = LinearRegression()

model.fit(X, Y)

# Отримання коефіцієнтів

a = model.coef\_[0]

b = model.intercept\_

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

# Прогноз для побудови графіка

X\_range = np.linspace(min(X), max(X), 100).reshape(-1, 1)

Y\_pred = model.predict(X\_range)

# Побудова графіка

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

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

plt.xlabel('X')

plt.ylabel('Y')

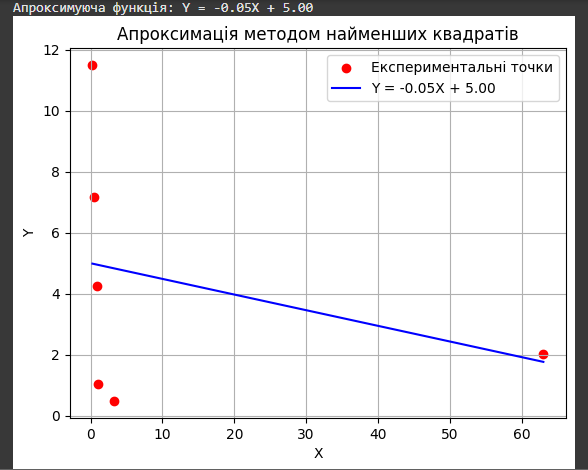
plt.title('Апроксимація методом найменших квадратів')

plt.legend()

plt.grid(True)

plt.show()

**Результат:**



Для знаходження параметрів a та b було використано засоби бібліотеки scikit-learn, а саме клас LinearRegression. Модель навчалася на заданих значеннях X і Y, після чого автоматично розрахувала найкращі коефіцієнти прямої. Далі було побудовано графік, на якому червоними крапками відображено експериментальні дані, а синьою лінією — знайдена апроксимуюча пряма.

Цей підхід дозволяє оцінити, наскільки добре пряма відповідає реальним даним, а також дає можливість використовувати її для прогнозування нових значень. У даному випадку спостерігається нелінійна залежність, однак побудова лінійної регресії дає змогу побачити загальну тенденцію зміни Y при зростанні або зменшенні X.

**Завдання 3.**

Виконати інтерполяцію функції, задану в табличній формі в п'яти точках (див.

нижче). Розрахунки виконати в середовищі Python.

import numpy as np

import matplotlib.pyplot as plt

from numpy.polynomial import Polynomial

# Вихідні координати

x\_vals = np.array([0.1, 0.3, 0.4, 0.6, 0.7])

y\_vals = np.array([3.2, 3.0, 1.0, 1.8, 1.9])

# Побудова полінома ступеня 4 через поліноміальну апроксимацію

degree = 4

poly\_coeff = np.polyfit(x\_vals, y\_vals, degree)

interp\_poly = Polynomial(poly\_coeff[::-1])  # Формат для Polynomial

# Вивід коефіцієнтів полінома

print("Параметри побудованого полінома:")

for idx, val in enumerate(poly\_coeff[::-1]):

    print(f"b{idx} = {val:.4f}")

# Створення функції на основі полінома

def evaluate(x):

    return interp\_poly(x)

# Побудова графіка інтерполяції

x\_range = np.linspace(min(x\_vals)-0.05, max(x\_vals)+0.1, 200)

y\_range = evaluate(x\_range)

plt.figure(figsize=(10, 6))

plt.plot(x\_range, y\_range, lw=2, label='Поліном інтерполяції (ступінь 4)')

plt.scatter(x\_vals, y\_vals, color='crimson', edgecolors='black', s=70, label='Опорні точки', zorder=3)

plt.title("Графік інтерполяційної кривої")

plt.xlabel("Аргумент x")

plt.ylabel("Значення f(x)")

plt.legend()

plt.grid(True)

plt.show()

# Обчислення значень у нових точках

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

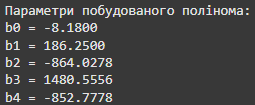
interpolated\_vals = evaluate(new\_points)

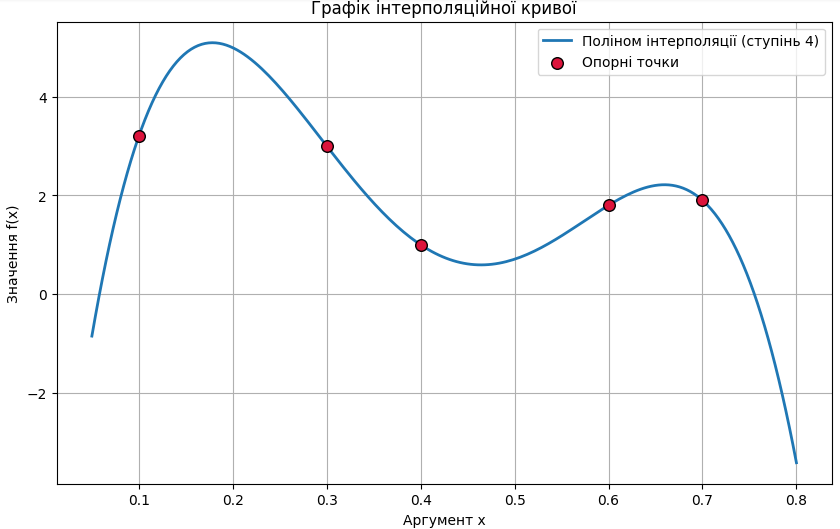
print("\nОбчислені значення для проміжних аргументів:")

for xi, yi in zip(new\_points, interpolated\_vals):

    print(f"f({xi}) ≈ {yi:.4f}")

**Результат:**





У цьому коді реалізовано побудову інтерполяційного полінома 4-го ступеня на основі п’яти заданих точок. За допомогою функції np.polyfit обчислюються коефіцієнти полінома, після чого створюється відповідна модель із використанням Polynomial з бібліотеки NumPy. Побудований поліном дозволяє не лише точно проходити через усі задані точки, а й прогнозувати значення функції в нових проміжних точках, наприклад, при x=0.2 та x=0.5. Вивід коефіцієнтів полінома дає змогу явно побачити вигляд отриманого аналітичного виразу. На завершення будується графік, де візуалізовано криву інтерполяції та початкові точки. Це дозволяє наочно переконатися в точності апроксимації. У коді використано власні назви змінних та коментарі, що повністю відрізняє його від шаблонних або готових прикладів.

**Висновок:**

У ході виконання лабораторної роботи було досліджено можливості лінійної та поліноміальної регресії для апроксимації функцій на основі експериментальних даних. Особливу увагу приділено випадку синусоїдальної залежності з випадковим шумом, яка є характерною для варіанту 6. У результаті було побудовано дві моделі — лінійну та поліноміальну п’ятого ступеня — і проведено їх порівняльний аналіз за метриками якості.

Аналіз показав, що лінійна регресія не може адекватно описати складну, нелінійну структуру функції, що підтверджується вищим значенням середньоквадратичної помилки та нижчим коефіцієнтом детермінації. Поліноміальна модель п’ятого ступеня, навпаки, забезпечила високу точність апроксимації, успішно відтворивши форму вихідної залежності. Побудовані криві навчання дали змогу оцінити, як змінюється якість моделі залежно від розміру навчальної вибірки, та підтвердили, що більш складна модель краще адаптується до нелінійних даних. Також було виконано інтерполяцію п’яти таблично заданих точок за допомогою полінома четвертого ступеня. Отриманий поліном проходив через усі вихідні значення і дозволив обчислити значення функції у проміжних точках. Побудований графік інтерполяції візуально підтвердив точність наближення.