import numpy as np

import csv

# Đọc dữ liệu từ file CSV và tách thành 4 mảng

def read\_csv\_to\_arrays(file\_path):

tv, radio, newspaper, sales = [], [], [], []

with open(file\_path, newline='', encoding='utf-8') as csvfile:

reader = csv.reader(csvfile)

next(reader)

for row in reader:

tv.append(float(row[0]))

radio.append(float(row[1]))

newspaper.append(float(row[2]))

sales.append(float(row[3]))

return tv, radio, newspaper, sales

file\_path = "Practice2\_Chapter2.csv"

tv, radio, newspaper, sales = read\_csv\_to\_arrays(file\_path)

tv = np.array(tv)

sales = np.array(sales)

radio = np.array(radio)

newspaper = np.array(newspaper)

# Initialize theta parameters and learning rate

theta0 = 0

theta1 = 0

theta2 = 0

theta3 = 0

learning\_rate = 1e-8

iterations = 30000

# Sales = theta0 + theta1 \* X1 + theta2 \* X2 + theta3 \* X3

# theta0 = theta01 + theta02 + theta03

def predict(X, theta0, theta1):

return theta0 + theta1 \* X

def cost\_function(X, Y, theta0, theta1):

m = len(X)

return (1 / (2 \* m)) \* np.sum((predict(X, theta0, theta1) - Y) \*\* 2)

def gradient\_descent(X, Y, theta0, theta1, learning\_rate):

m = len(X)

gradient0 = (1 / m) \* np.sum(predict(X, theta0, theta1)-Y)

gradient1 = (1 / m) \* np.sum((predict(X, theta0, theta1) - Y) \* X)

new\_theta0 = theta0 - learning\_rate \* gradient0

new\_theta1 = theta1 - learning\_rate \* gradient1

return new\_theta0, new\_theta1

def linear\_gression(X, Y, theta0, theta1, learning\_rate):

for i in range(iterations):

theta0, theta1 = gradient\_descent(X, Y, theta0, theta1, learning\_rate)

return theta0, theta1

theta01, theta1 = linear\_gression(tv, sales, theta0, theta1, learning\_rate)

theta02, theta2 = linear\_gression(radio, sales, theta0, theta2, learning\_rate)

theta03, theta3 = linear\_gression(newspaper, sales, theta0, theta3, learning\_rate)

print(theta01 + theta02 + theta03)

print(theta1)

print(theta2)

print(theta3)

