import numpy as np

from sklearn.preprocessing import StandardScaler

import numpy as np

#data

X = np.array([150, 180, 164, 162, 181, 182, 173, 190, 171, 170, 181, 182, 189, 184, 209, 210])

y = np.array([51, 52, 54, 53, 55, 59, 61, 59, 63, 76, 64, 66, 69, 72, 70, 80])

# Initialize theta parameters and learning rate

theta0 = 0

theta1 = 0

learning\_rate = 1e-8

iterations = 30000

# Chuẩn hóa dữ liệu

scaler = StandardScaler()

data = X.reshape(-1, 1)

X = scaler.fit\_transform(data)

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

for i in range(iterations):

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

print("theta0: ", theta0, "theta1: ", theta1)

