**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**

**-----------------⸙∆⸙-----------------**



**ĐỒ ÁN MÔN HỌC**

**Bài tập 2**

**GVHD: Trần Vũ Hoàng**

**SVTH: Nguyễn Công Trung**

**MSSV: 2341109**

**Tp. Hồ Chí Minh tháng 07 năm 2024**

# Mục lục

[Mục lục i](#_Toc171712059)

[Danh sách hình ảnh ii](#_Toc171712060)

[Chương 1. Đề bài 3](#_Toc171712061)

[Chương 2. BÀI LÀM 4](#_Toc171712062)

[2.1 Code 4](#_Toc171712063)

[2.2 Kết quả 5](#_Toc171712064)

[Chương 3. KẾT LUẬN 10](#_Toc171712065)

# Danh sách hình ảnh

[Hình 1. Learning rate = 1, số vòng lặp = 100 5](#_Toc171712051)

[Hình 2. Theta tìm được 6](#_Toc171712052)

[Hình 3. Learning rate = 0.1, số vòng lặp = 100 6](#_Toc171712053)

[Hình 4. Theta tìm được 7](#_Toc171712054)

[Hình 5. Learning rate = 0.01, số vòng lặp = 100 7](#_Toc171712055)

[Hình 6. Theta tìm được 8](#_Toc171712056)

[Hình 7. Learning rate = 0.1, số vòng lặp = 1000 8](#_Toc171712057)

[Hình 8. Theta tìm được 9](#_Toc171712058)

# Đề bài

Trong tập dữ liệu tôi gửi bao gồm 17000 hàng và 9 cột, trong đó:

• 17000 hàng là số lượng dữ liệu (m)

• 9 cột: 8 cột đầu ( n ) là các đặc trưng (x), cột cuối cùng là giá nhà ( y )

Các bạn làm các công việc sau:

1. Chuẩn hóa dữ liệu: scale dữ liệu, format kích thước dữ liệu

2. Viết chương trình cho phép học các tham số của mô hình hồi quy tuyến tínhđa biến

3. Tính J ở mỗi vòng lặp, và vẽ biểu đồ J ở các giá trị learning rate khác nhau sau khi chạy hết các vòng lặp.

4. Kiểm chứng các theta mà các bạn tìm được.

# BÀI LÀM

## Code

import polars as pl

import numpy as np

import matplotlib.pyplot as plt

data = pl.read\_csv(r'ex2.csv')

columns = data.columns

n = len(columns)-1

m = len(data)

x = data.select(columns[:n])

y = data.select(columns[-1])

mean\_x = x.select([pl.col(col).mean().alias(f"mean\_{col}") for col in x.columns])

std\_x = x.select([pl.col(col).std().alias(f"std\_{col}") for col in x.columns])

mean\_y = y.mean()

std\_y = y.std()

standardized\_x = x.with\_columns(

    [(pl.col(col) - mean\_x[f"mean\_{col}"]) / std\_x[f"std\_{col}"] for col in x.columns]

)

standardized\_y = (y - mean\_y[y.columns[0]]) / std\_y[y.columns[0]]

standardized\_x = standardized\_x.with\_columns(pl.lit(1).alias("constant"))

standardized\_x = standardized\_x.select(["constant"] + columns[:n])

x\_mat = standardized\_x.to\_numpy().reshape(-1, n+1)

y\_mat = standardized\_y.to\_numpy().reshape(-1, 1)

theta\_normal\_equ = np.matmul(np.matmul(np.linalg.inv(np.matmul(x\_mat.T, x\_mat)), x\_mat.T), y\_mat)

theta = np.random.rand(n+1, 1)

H = np.matmul(x\_mat, theta)

alpha = 1

loop = 100

J\_values = np.zeros((loop, 1))

for i in range(loop):

    theta = theta - (alpha/m)\*np.matmul(x\_mat.T, (H-y\_mat))

    H = np.matmul(x\_mat, theta)

    J\_values[i] = (1/(2\*m))\*np.matmul((H-y\_mat).T, (H-y\_mat))

print(theta)

print(theta\_normal\_equ)

# Create a figure and a set of subplots

fig, axs = plt.subplots()

# Plot the cost function on the first subplot

axs.plot(J\_values)

axs.set\_xlabel('Iteration')

axs.set\_ylabel('Cost (J)')

axs.set\_title('Cost Function')

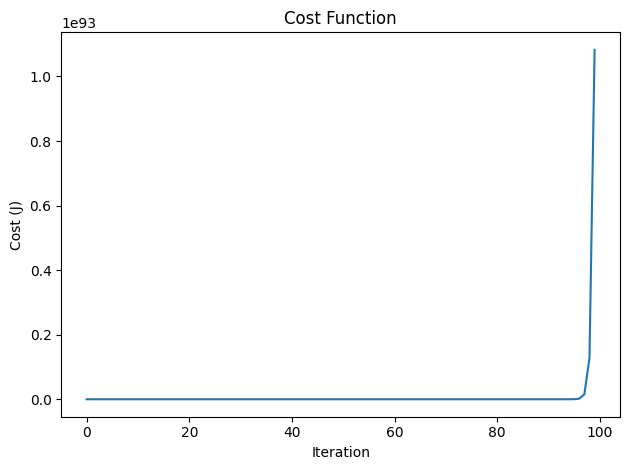
# Layout so plots do not overlap

fig.tight\_layout()

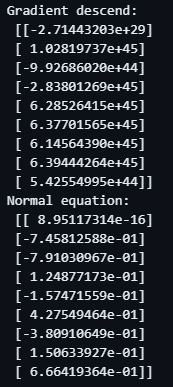
plt.show()

## Kết quả

Thông số learning rate = 1, số vòng lặp = 100.

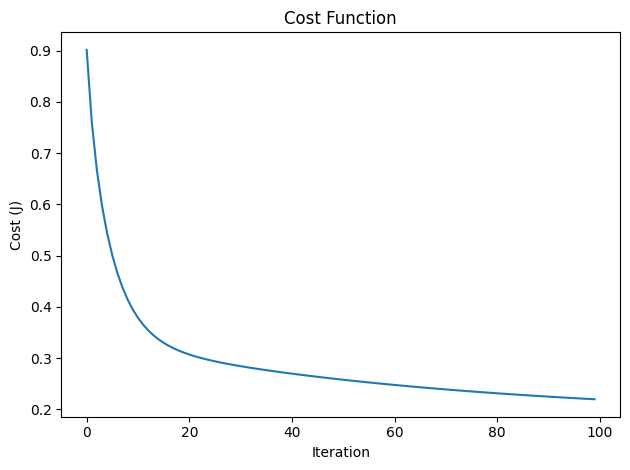


Hình . Learning rate = 1, số vòng lặp = 100

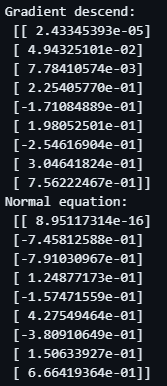


Hình . Theta tìm được

Thông số learning rate = 0.1, số vòng lặp = 100.

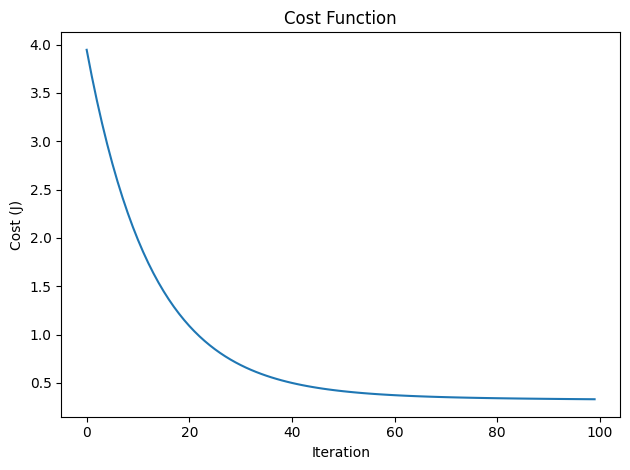


Hình . Learning rate = 0.1, số vòng lặp = 100

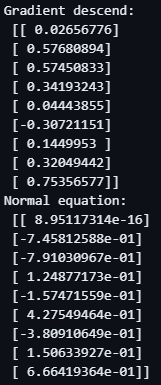


Hình . Theta tìm được

Thông số learning rate = 0.01, số vòng lặp = 100.

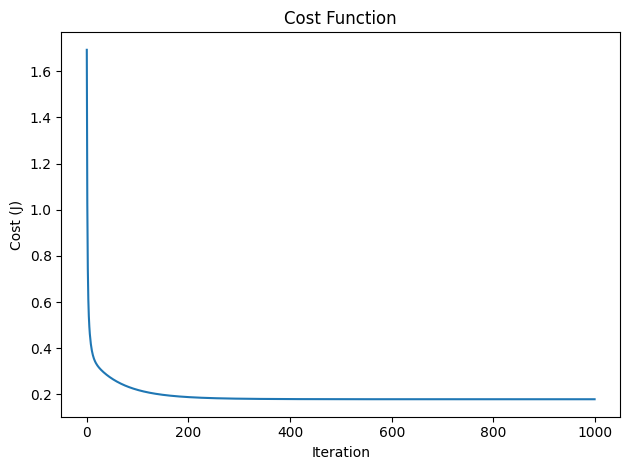


Hình . Learning rate = 0.01, số vòng lặp = 100

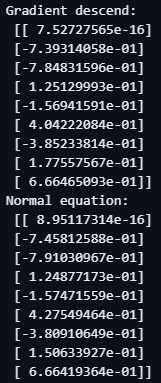


Hình . Theta tìm được

Thông số learning rate = 0.1, số vòng lặp = 1000.



Hình . Learning rate = 0.1, số vòng lặp = 1000



Hình . Theta tìm được

# KẾT LUẬN

Khi đã chuẩn hóa dữ liệu, lúc này có thể tăng learning rate lên cao, từ đó hội tụ nhanh hơn. Với phương pháp gradient descend, khi đã đạt hội tụ, bộ thông số sẽ gần giống với phương pháp normal equation.