**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 3**

**GVHD: Trần Vũ Hoàng**

**SVTH: Nguyễn Công Trung**

**MSSV: 2341109**

**Tp. Hồ Chí Minh tháng 07 năm 2024**

# Mục lục

[Mục lục i](#_Toc171712406)

[Danh sách hình ảnh ii](#_Toc171712407)

[Chương 1. Đề bài 3](#_Toc171712408)

[Chương 2. BÀI LÀM 4](#_Toc171712409)

[2.1 Code 4](#_Toc171712410)

[2.2 Kết quả 5](#_Toc171712411)

[Chương 3. KẾT LUẬN 9](#_Toc171712412)

# Danh sách hình ảnh

[Hình 1. Learning rate = 1, số vòng lặp = 100 5](#_Toc171712400)

[Hình 2. Learning rate = 0.1, số vòng lặp = 100 6](#_Toc171712401)

[Hình 3. Learning rate = 0.01, số vòng lặp = 100 7](#_Toc171712402)

[Hình 4. Learning rate = 0.1, số vòng lặp = 1000 8](#_Toc171712403)

# Đề bài

Trong tập dữ liệu tôi gửi bao gồm 84 hàng và 2 cột, trong đó:

• 84 hàng là số lượng dữ liệu (m)

• 2 cột: cột đầu là đặc trưng (x), cột cuối là nhãn ( y )

Các bạn làm các công việc sau:

1. Biểu diễn dữ liệu

2. Chọn mô hình "h" phù hợp (bậc bao nhiêu?)

3. Chuyển bài toán đa bậc thành bài toán đa biến và chuẩn hóa dữ liệu: scale dữ liệu, formatkích thước dữ liệu

4. Viết chương trình cho phép học các tham số của mô hình hồi quy tuyến tính đa biến

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

6. Biển diễn đường cong học được và dữ liệu trên cùng 1 hình ảnh.

# BÀI LÀM

## Code

import polars as pl

import numpy as np

import matplotlib.pyplot as plt

data = pl.read\_csv(r'ex3.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"), (pl.col("x") \*\* 2).alias("x\_squared"))

standardized\_x = standardized\_x.select(["constant"] + [col for col in standardized\_x.columns if col != "constant"])

n = len(standardized\_x.columns)

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

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

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

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

alpha = 0.01

loop = 1000

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))

# Create a figure and a set of subplots

fig, axs = plt.subplots(2)

# Plot the cost function on the first subplot

axs[0].plot(J\_values)

axs[0].set\_xlabel('Iteration')

axs[0].set\_ylabel('Cost (J)')

axs[0].set\_title('Cost Function')

axs[1].plot(standardized\_x['x'], standardized\_y, "b.")

x\_line = np.linspace(standardized\_x['x'].min(), standardized\_x['x'].max(), m).reshape(m, 1)

x\_line\_poly = np.hstack((np.ones((x\_line.shape[0], 1)), x\_line, x\_line\*\*2))

y\_new = np.matmul(x\_line\_poly, theta)

axs[1].plot(x\_line, y\_new, "r-", linewidth = 2, label ="Predictions")

axs[1].set\_xlabel('x')

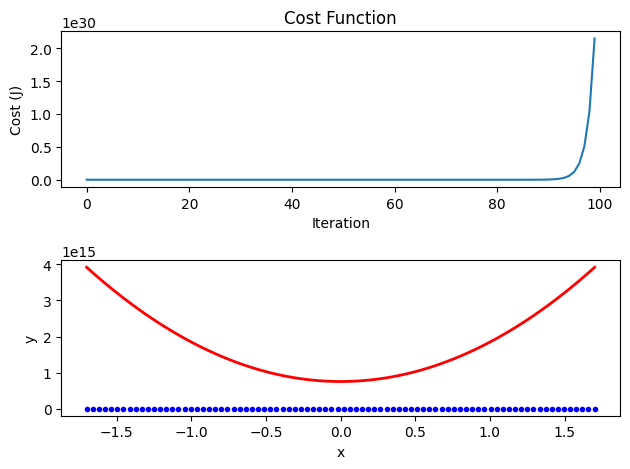
axs[1].set\_ylabel('y')

# Layout so plots do not overlap

fig.tight\_layout()

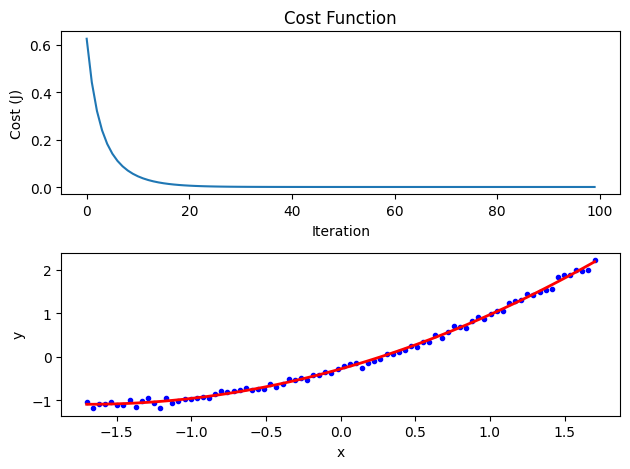
## Kết quả

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



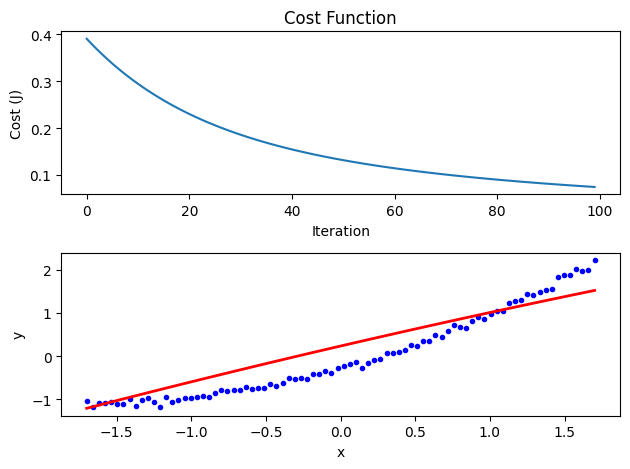
Hình . Learning rate = 1, số vòng lặp = 100

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



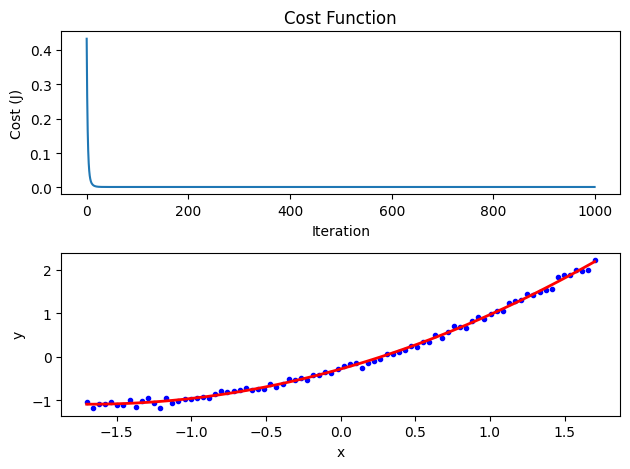
Hình . Learning rate = 0.1, số vòng lặp = 100

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



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

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



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

# KẾT LUẬN

Khi đã chuẩn hóa dữ liệu, lúc này có thể tăng learning rate lên cao, từ đó hội tụ nhanh hơn.