

Nguyễn Khắc Sơn, MSSV: 21085691

BTVN B2

Bài 1: Dựa trên những kiến thức đã học trên lớp và đọc thêm tài liệu ở nhà, em hãy lấy một ví dụ thực thể liên quan tới lĩnh vực IoT có thể giải quyết thông qua bài toán xấp xỉ hàm. Giải thích ngắn gọn lý do.

Ví dụ:

Hệ thống IoT quản lý, theo dõi chất lượng không khí của thành phố

Các cảm biến IoT được lắp đặt khắp nơi trên địa bàn để đo lường nồng độ các chất ô nhiễm như CO₂, NO₂, PM_{2.5},... và nhiệt độ, độ ẩm không khí.

Hệ thống này sẽ dự đoán chất lượng không khí trong thời gian sắp tới để đưa ra các cảnh báo sớm.

Lý do:

Xấp xỉ hàm giúp tìm ra mối quan hệ phi tuyến giữa các yếu tố môi trường và chất lượng không khí.

Từ đó, mô hình dự đoán sẽ giúp thành phố dự đoán được mức độ ô nhiễm, để kịp thời đưa ra quyết định về việc điều chỉnh hoạt động sản xuất, hoặc khuyến cáo người dân về mức độ ô nhiễm.

Bài 2: (Giải bài toán bằng bút và máy tính cầm tay)

Trong một tòa nhà thông minh, thông qua việc xử lý ảnh để đo được chiều cao của 1 một người, chương trình sẽ dự đoán cân nặng của người đó. Sử dụng hồi quy tuyến tính để xây dựng mô hình dự đoán cân nặng dựa trên chiều cao. Sau đó hãy thử dự đoán cân nặng của một người mới chuyển đến tòa nhà có chiều cao là 185 cm. Biết rằng đã có một tập dữ liệu có sẵn như sau:

Mẫu	Số phòng	Chiều cao (cm)	Cân nặng (kg)	Chỉ số cơ dân
1	301	147	49	Rất tốt
2	312	150	50	Tốt
3	320	153	51	Khá
4	401	155	52	Tốt
5	402	158	54	Trung bình
6	408	160	56	Khá
7	409	163	58	Khá
8	503	165	59	Tốt
9	507	168	60	Trung bình
10	520	170	72	Kém
11	606	173	63	Tốt
12	607	175	64	Tốt
13	608	178	66	Khá
14	609	180	67	Kém
15	610	183	68	Rất tốt

Ngày:

Bài 2:

	CAO	NANG	$x \cdot y$	x^2	y^2
1.	147	49	7203	21609	2401
2.	150	50	7500	22500	2500
3.	153	51	7803	23409	2601
4.	155	52	8060	24025	2704
5.	158	54	8532	24964	2916
6.	160	56	8960	25600	3136
7.	163	58	9454	26569	3364
8.	165	59	9735	27225	3481
9.	168	60	10080	28224	3600
10.	170	72	12240	28900	5184
11.	173	63	10899	29929	3969
12.	175	64	11200	30625	4096
13.	178	66	11748	31684	4356
14.	180	67	12060	32400	4489
15.	183	68	12444	33489	4624
SUM:	2478	889	147918	111152	53421
Avg:	$\div 15$	165,2	59,267	9861,2	27410,13
	\bar{x}	\bar{y}	$\bar{x \cdot y}$	$\bar{x^2}$	$\bar{y^2}$

$$r_{xy} = \frac{\bar{x \cdot y} - \bar{x} \cdot \bar{y}}{\sqrt{\bar{x^2} - (\bar{x})^2} \cdot \sqrt{\bar{y^2} - (\bar{y})^2}} = 0,922.17 \Rightarrow y = \theta_1 + \theta_2 x$$

$$\theta_2 = \frac{\bar{x \cdot y} - \bar{x} \cdot \bar{y}}{\bar{x^2} - (\bar{x})^2} = 0,59668$$

$$\theta_1 = \bar{y} - \theta_2 \cdot \bar{x} = -38,3145$$

$$y = -38,31 + 0,59668x$$

$$y = 70,96.13$$

Bài 3: (Giải bài toán bằng cách lập trình)

Sử dụng Python, lập một chương trình dự đoán tự động. Đầu vào nhập một mẫu dữ liệu mới là chiều cao của một người (ví dụ một người mới chuyển đến tòa nhà có chiều cao là 185 cm). Đầu ra in ra kết quả dự đoán cân nặng (kg). So sánh kết quả thu được với kết quả tính toán ở bài 2.

```
dudoan_cannang_2.py > nhap_du_lieu_moi
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from sklearn.linear_model import LinearRegression
4
5  def du_doan_can_nang(model):
6      # Nhập chiều cao mới từ bàn phím
7      new_height = float(input("Nhập chiều cao mới (cm): "))
8      # Dự đoán cân nặng dựa trên chiều cao mới
9      predicted_weight = model.predict(np.array([[new_height]]))
10     # In ra kết quả dự đoán
11     print(f"Cân nặng dự đoán cho người có chiều cao {new_height} cm là {predicted_weight[0]:.2f} kg.")
12     return new_height, predicted_weight[0]
13
14 def nhap_du_lieu_moi():
15     # Nhập số lượng mẫu
16     so_mau = int(input("Nhập số lượng mẫu dữ liệu: (tối thiểu 15 mẫu) "))
17     while so_mau < 15:
18         print("Bạn phải nhập tối thiểu 15 mẫu dữ liệu!")
19         so_mau = int(input("Nhập số lượng mẫu dữ liệu: (tối thiểu 15 mẫu) "))
20
21     # Nhập chiều cao tất cả mẫu cùng lúc
22     chieu_cao_str = input(f"Nhập chiều cao của {so_mau} mẫu (cách nhau bằng dấu cách): ")
23     chieu_cao_moi = list(map(float, chieu_cao_str.split()))
24
25     # Kiểm tra số lượng mẫu có khớp không
26     while len(chieu_cao_moi) != so_mau:
27         print(f"Bạn phải nhập đủ {so_mau} mẫu chiều cao!")
28         chieu_cao_str = input(f"Nhập chiều cao của {so_mau} mẫu (cách nhau bằng dấu cách): ")
29         chieu_cao_moi = list(map(float, chieu_cao_str.split()))
30
31     # Nhập cân nặng tất cả mẫu cùng lúc
32     can_nang_str = input(f"Nhập cân nặng của {so_mau} mẫu (cách nhau bằng dấu cách): ")
33     can_nang_moi = list(map(float, can_nang_str.split()))
34
35     # Kiểm tra số lượng mẫu cân nặng có khớp không
36     while len(can_nang_moi) != so_mau:
37         print(f"Bạn phải nhập đủ {so_mau} mẫu cân nặng!")
38         can_nang_str = input(f"Nhập cân nặng của {so_mau} mẫu (cách nhau bằng dấu cách): ")
```

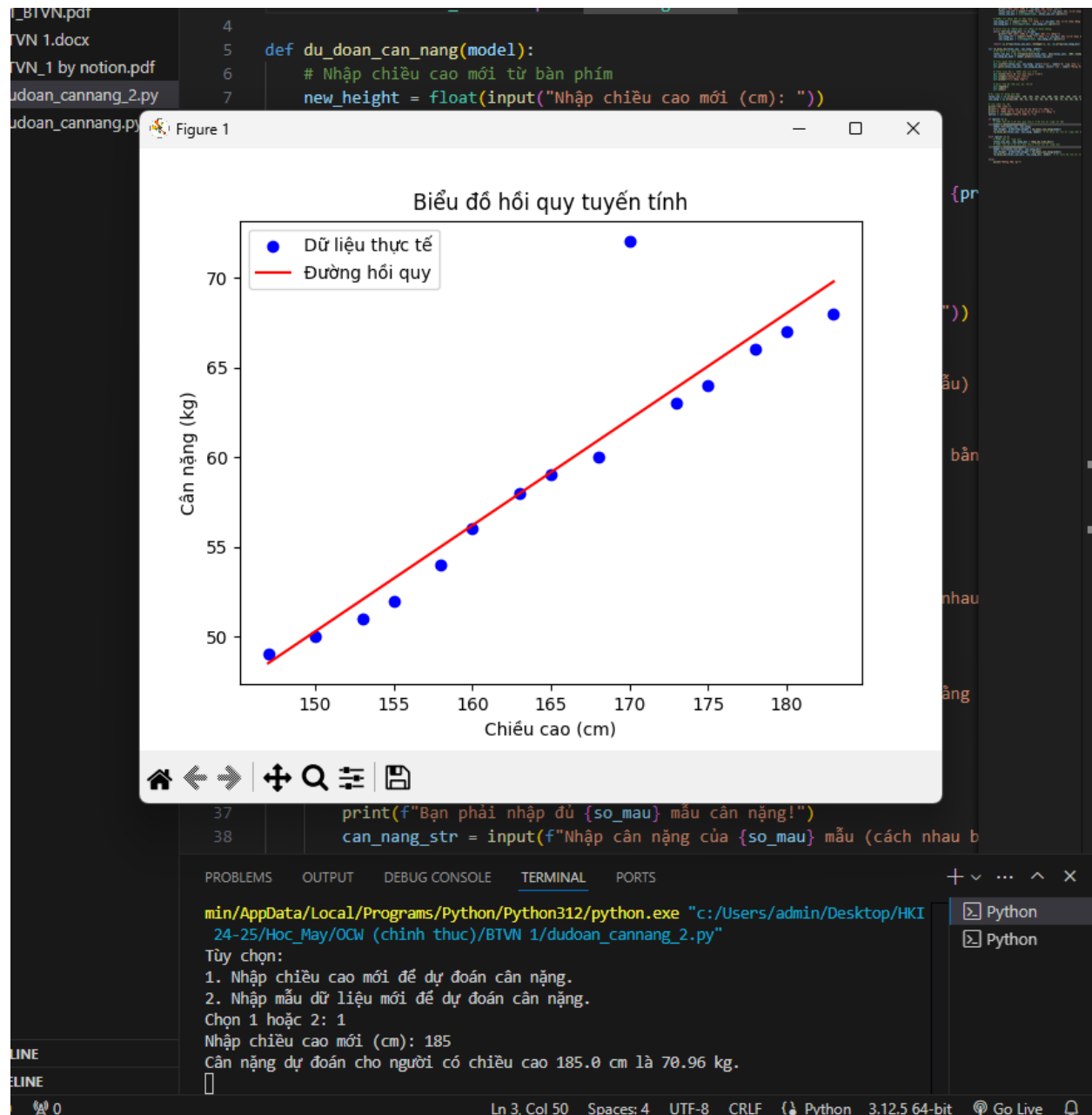
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

dudoan_cannang_2.py > ...

```
14 def nhap_du_lieu_moi():
35     # Kiểm tra số lượng mẫu cân nặng có khớp không
36     while len(can_nang_moi) != so_mau:
37         print(f"Bạn phải nhập đủ {so_mau} mẫu cân nặng!")
38         can_nang_str = input(f"Nhập cân nặng của {so_mau} mẫu (cách nhau bằng dấu cách): ")
39         can_nang_moi = list(map(float, can_nang_str.split()))
40
41     return np.array(chieu_cao_moi).reshape(-1, 1), np.array(can_nang_moi)
42
43 def ve_bieu_do(chieu_cao, can_nang, model):
44     # Dự đoán cân nặng dựa trên chiều cao
45     chieu_cao_moi = np.linspace(min(chieu_cao), max(chieu_cao), 100).reshape(-1, 1)
46     can_nang_du_doan = model.predict(chieu_cao_moi)
47
48     # Trực quan hóa dữ liệu
49     plt.scatter(chieu_cao, can_nang, color='blue', label='Dữ liệu thực tế') # Dữ liệu gốc
50     plt.plot(chieu_cao_moi, can_nang_du_doan, color='red', label='Đường hồi quy') # Đường hồi quy
51
52     # Thêm tiêu đề và nhãn cho các trục
53     plt.title("Biểu đồ hồi quy tuyến tính")
54     plt.xlabel("Chiều cao (cm)")
55     plt.ylabel("Cân nặng (kg)")
56
57     # Hiển thị đồ thị với chú thích
58     plt.legend()
59     plt.show()
60
61 # Dữ liệu từ bảng ban đầu
62 chieu_cao = np.array([147, 150, 153, 155, 158, 160, 163, 165, 168, 170, 173, 175, 178, 180, 183]).reshape(-1, 1)
63 can_nang = np.array([49, 50, 51, 52, 54, 56, 58, 59, 60, 62, 63, 64, 66, 67, 68])
64
```

```
64
65 # Lựa chọn chế độ
66 print("Tùy chọn:")
67 print("1. Nhập chiều cao mới để dự đoán cân nặng.")
68 print("2. Nhập mẫu dữ liệu mới để dự đoán cân nặng.")
69 option = int(input("Chọn 1 hoặc 2: "))
70
71 if option == 1:
72     # Khởi tạo mô hình hồi quy tuyến tính với dữ liệu có sẵn
73     model = LinearRegression()
74     model.fit(chieu_cao, can_nang)
75     new_height, predicted_weight = du_doan_can_nang(model)
76     ve_bieu_do(chieu_cao, can_nang, model) # Vẽ biểu đồ với dữ liệu ban đầu
77
78 elif option == 2:
79     # Nhập mẫu dữ liệu mới
80     chieu_cao_moi, can_nang_moi = nhap_du_lieu_moi()
81     # Khởi tạo mô hình hồi quy tuyến tính với dữ liệu mới
82     model = LinearRegression()
83     model.fit(chieu_cao_moi, can_nang_moi)
84     new_height, predicted_weight = du_doan_can_nang(model)
85     ve_bieu_do(chieu_cao_moi, can_nang_moi, model) # Vẽ biểu đồ với dữ liệu mới
86
87 else:
88     print("Không hợp lệ!")
89
```

Kết quả thực hiện:



Bài tập VD3: dự đoán chiều dài cuộn dây thép

VD3> Mẫu	x Năng	y đầu
1	30	70
2	40	90
3	40	100
4	50	120
5	50	130
6	50	150
7	60	160
8	70	190
9	70	200
10	80	200
11	80	220
12	80	230

$$\hat{y} = f(x) = \theta_0 + \theta_1 x$$

$$\Rightarrow \hat{y} = -20 + 3x$$

$$\left\{ \begin{array}{l} r = 0,98076 \\ \theta_0 = -20 \\ \theta_1 = 3 \end{array} \right.$$

Vậy: với $x = 35$

\Rightarrow Dự đoán:

$$\hat{y} = -20 + 3 \cdot (35)$$

$$\hat{y} = 85$$

Nhập CASIO 570

\rightarrow MODE 3 2 để nhập mẫu

\rightarrow SHIFT 5 \rightarrow 1 : θ_0

\rightarrow 2 : θ_1

\rightarrow 3 : r : hệ số tương quan mẫu