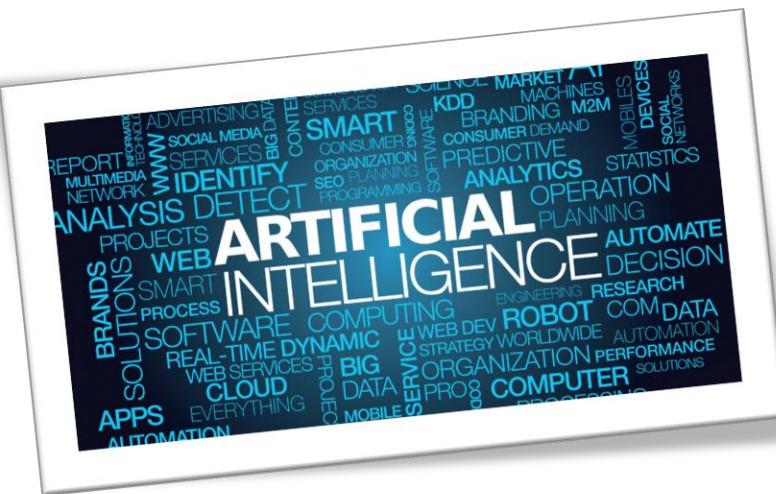


LẬP TRÌNH TRÍ TUỆ NHÂN TẠO



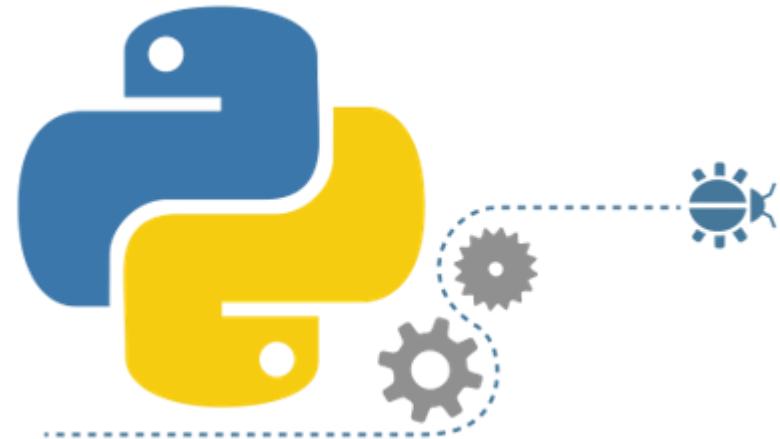
THS. VƯƠNG XUÂN CHÍ

VXCHI@NTT.EDU.VN

0903 270 567



CHƯƠNG 10

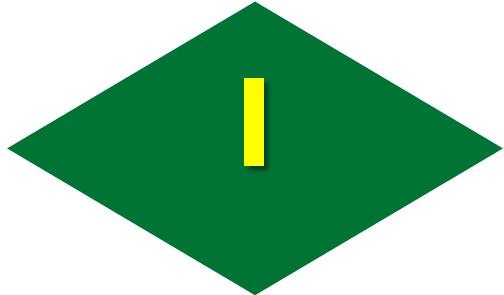


Thư viện scikit-learn



Nội dung

- 1. Học máy*
- 2. Một số loại bài toán học máy*
- 3. Thư viện học máy scikit-learn*



Học máy

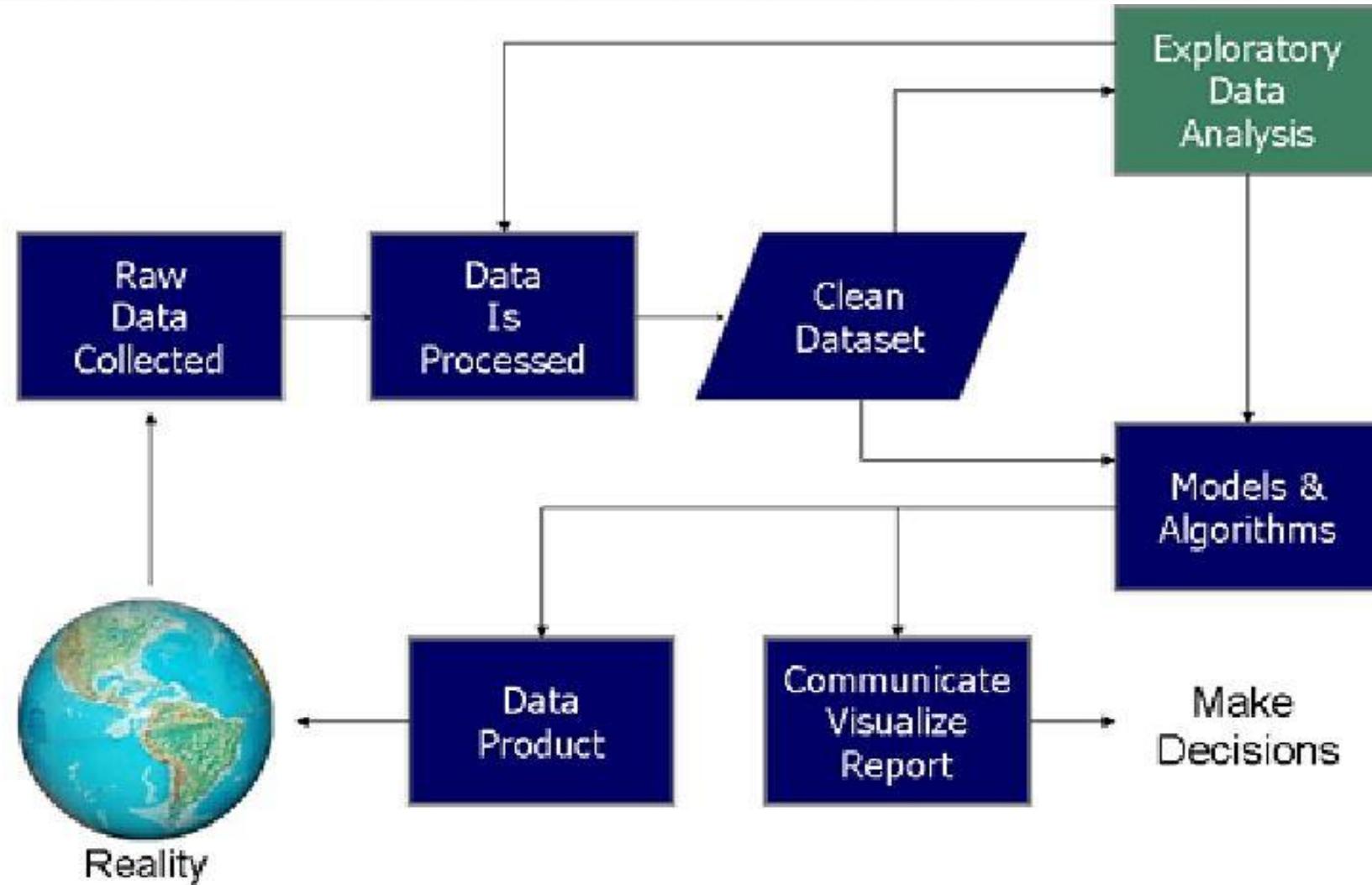


Học máy

- Máy học là một thuật ngữ đề cập đến các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ, đồng thời cải thiện hiệu suất theo thời gian.
- Học máy là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển.
- Học máy là phương pháp quan trọng để xử lý dữ liệu trong ngành data science, bên cạnh những phương pháp truyền thống khác.
- Máy học như là lĩnh vực dự đoán:
 - ví dụ X với các tính năng đặc biệt, dự đoán Y về nó. Những dự đoán này có thể là về tương lai



Quá trình xử lý dữ liệu





2

Một số loại bài toán học máy



Một số bài toán thực tế

- Hệ thống phân loại email
- Nhận dạng chữ viết từ ảnh
- Ước lượng giá cả của sản phẩm
- Dự báo thời tiết
- Đánh giá trạng thái của người qua ảnh/video
- Trả lời tự động (chat bot)
- Gợi ý sản phẩm phù hợp với nhu cầu khách hàng
- Tự động chơi trò chơi
- Mô phỏng giọng nói của một người nào đó



Các lớp bài toán cơ bản

- Học có giám sát (**supervised learning**): học cách tiên đoán đầu ra theo mẫu cho trước
 - ✓ Tập mẫu cho trước, cho cả đầu bài và kết quả
 - ❖ Cho email, chỉ rõ trước đây là spam, đây không phải spam
 - ✓ Mô hình được huấn luyện trên tập mẫu
 - ✓ Thử nghiệm bằng cách cho đầu bài, mô hình tiên đoán kết quả, mô hình đoán càng chính xác càng tốt
 - ❖ Cho một email mới, máy tính đoán xem có phải spam không?
 - ✓ Có 2 loại cơ bản:
 - ❖ Hồi quy (regression): đầu ra là số hoặc vector
 - ❖ Phân lớp (classification): đầu ra thường là xác suất dự báo



Các lớp bài toán cơ bản

- Học không giám sát ([unsupervised learning](#)): tự khai phá các đặc trưng nội tại hợp lý của đầu vào
 - ✓ Chỉ cho mẫu vào, không cho biết đầu ra
 - Cho tập băng ghi âm lời nói của một người
 - ✓ Hệ thống tự học trên các mẫu mà không có định hướng
 - Tạo ra một đoạn phát âm theo ngữ điệu của người đã cho
 - ✓ Một vài chiến lược cơ bản:
 - Biến đổi dữ liệu đầu vào có số chiều cao thành dữ liệu có số chiều thấp hơn
 - Dữ liệu có số chiều cao nhưng các đặc trưng thành phần có tính “kinh tế” ([economical](#)) hơn
- Gom cụm dữ liệu đầu vào



Các lớp bài toán cơ bản

Học tăng cường (reinforcement learning): hiệu chỉnh các siêu tham số (hyperparameter) để cực đại hóa lợi ích trong tương lai

- Cho bối cảnh và các quy tắc
 - **Bàn cờ Vây và các quy tắc của trò chơi cờ Vây**
- Ứng với mỗi hành động (hoặc chuỗi hành động), có một phần thưởng tương ứng
 - **Đặt một quân sẽ bị mất điểm, không được hoặc được điểm**
- Hệ thống tự điều chỉnh chuỗi hành động sao cho được phần thưởng lớn nhất
 - **Hệ thống học cách chơi để thắng người chơi giỏi nhất**



3

Thư viện học máy scikit-learn

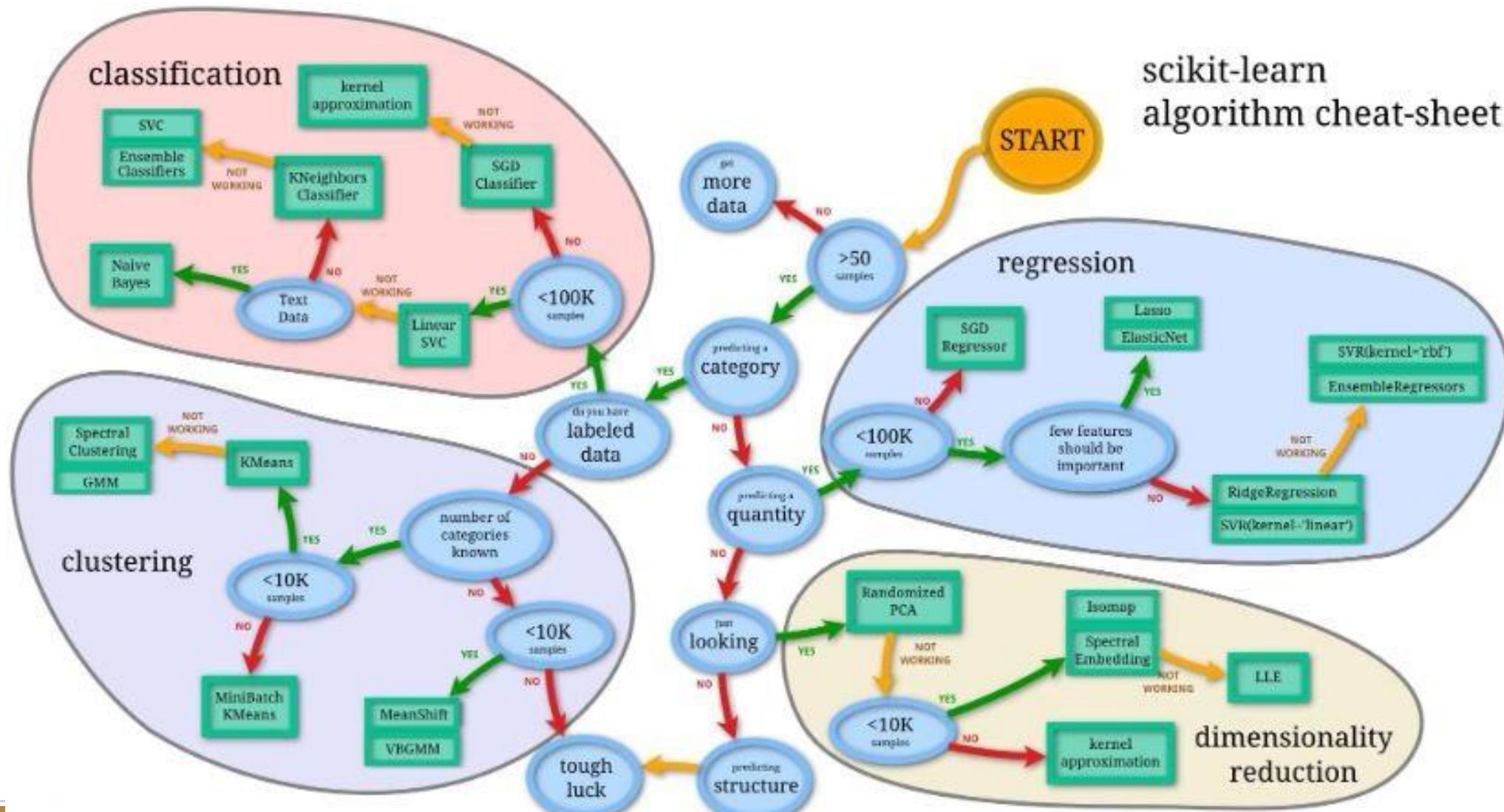


Thư viện học máy scikit-learn

- **Scikit-learn** xuất phát là một dự án trong một cuộc thi lập trình của Google vào năm 2007, người khởi xướng dự án là David Cournapeau
- Sau đó nhiều viện nghiên cứu và các nhóm ra nhập, đến năm 2010 mới có bản đầu tiên (v0.1 beta)
- Scikit-learn cung cấp gần như tất cả các loại thuật toán học máy cơ bản (khoảng vài chục) và vài trăm biến thể của chúng, cùng với đó là các kĩ thuật xử lý dữ liệu đã được chuẩn hóa
- Cài đặt: **pip install scikit-learn scipy**



Chọn thuật toán học máy phù hợp





Ví dụ: dự báo cân nặng của người

- Tập mẫu quan sát có n người
 - Gồm tên, chiều cao, cân nặng
 - Và nhiều loại chỉ số khác nữa
- Xây dựng một mô hình dự báo về cân nặng người, dựa trên các chỉ số còn lại
 - Trong trường hợp bài toán, chúng ta cố gắng dự báo cân nặng từ chiều cao
 - Thực tế thì cân nặng phụ thuộc vào nhiều thông số khác nữa, như giới tính, vòng eo,...

	A	B	C
1	Ten	Cao	Nang
2	A	147	49
3	B	150	50
4	C	153	51
5	D	155	51
6	E	168	60
7	F	170	62
8	G	173	68
9	H	175	65
10	I	178	66
11	J	180	71
12	K	183	68
13	L	165	59
14	M	163	58
15	N	160	56
16	O	158	54
17	P	169	62
18	Q	172	63
19	S	170	62
20	T	176	62
21	U	180	69



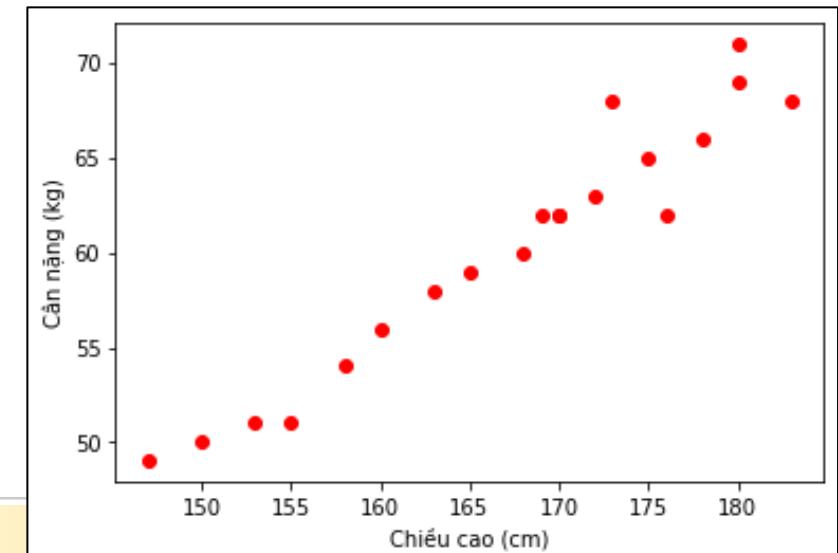
Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ I:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn import linear_model, metrics

# đọc dữ liệu từ file csv
df = pd.read_csv("nguoil.csv", index_col = 0)
print(df)

# vẽ biểu đồ minh họa dataset
plt.plot(df.Cao, df.Nang, 'ro')
plt.xlabel('Chiều cao (cm)')
plt.ylabel('Cân nặng (kg)')
plt.show()
```





Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ I:

```
# sử dụng hồi quy tuyến tính
x = df.loc[:, ['Cao']].values # x là dữ liệu đầu vào
y = df.Nang.values # y là dữ liệu đầu ra
model = linear_model.LinearRegression() # loại mô hình
model.fit(x, y) # tập huấn trên dữ liệu
# in một số thông tin về mô hình
mse = metrics.mean_squared_error(model.predict(x), y)
print("Tổng bình phương sai số trên tập mẫu:", mse)
print("Hệ số hồi quy:", model.coef_)
print("Sai số:", model.intercept_)
print(f"Công thức: [Nặng] = {model.coef_} x [Cao] + {model.intercept_}")
```

```
Tổng bình phương sai số trên tập mẫu: 2.5466855190291047
Hệ số hồi quy: [0.61529398]
Sai số: -42.60791799222339
Công thức: [Nặng] = [0.61529398] x [Cao] +-42.60791799222339
```

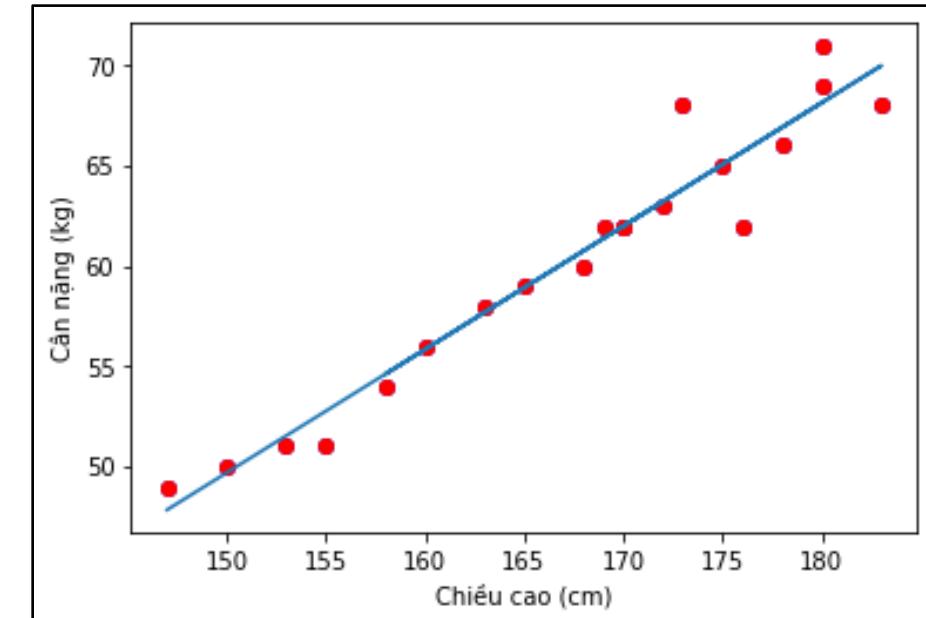


Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ I:

```
# vẽ lại sơ đồ
plt.scatter(X, y, c='b')
plt.plot(X, model.predict(X))
plt.show()

# dự báo một số tình huống
while True:
    x = float(input("Nhập chiều cao
(nhập 0 để dừng): "))
    if x <= 0: break
    print("Người cao", x, "cm, dự báo
cân nặng", model.predict([[x]]))
```





Mở rộng: thêm cột giới tính

VÍ DỤ 2:

- Vẫn dữ liệu cũ, bổ sung thêm cột giới tính (Nam/Nu)
- Sử dụng phương pháp cũ, để xem giới tính ảnh hưởng như thế nào đến cân nặng

	A	B	C	D
1	Ten	Gioitinh	Cao	Nang
2	A	Nu	147	49
3	B	Nu	150	50
4	C	Nu	153	51
5	D	Nam	155	51
6	E	Nu	168	60
7	F	Nam	170	62
8	G	Nu	173	68
9	H	Nam	175	65
10	I	Nam	178	66
11	J	Nam	180	71
12	K	Nam	183	68
13	L	Nam	165	59
14	M	Nu	163	58
15	N	Nu	160	56
16	O	Nu	158	54
17	P	Nam	169	62
18	Q	Nam	172	63
19	S	Nu	170	62
20	T	Nam	176	62
21	U	Nam	180	69



Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ 2:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn import linear_model, metrics

# đọc dữ liệu từ file csv
df = pd.read_csv("nguois.csv", index_col = 0)
print(df)
# thêm cột mới, giới tính Nam = 1, giới tính Nữ = 0
df['GT'] = df.Gioitinh.apply(lambda x: 1 if x=='Nam' else 0)
print(df)
```



Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ 2:

```
# sử dụng hồi quy tuyến tính
X = df.loc[:, ['Cao', 'GT']].values # X là dữ liệu đầu vào
y = df.Nang.values # y là dữ liệu đầu ra
model = linear_model.LinearRegression() # loại mô hình
model.fit(X, y) # tập huấn trên dữ liệu

# in một số thông tin về mô hình
mse = metrics.mean_squared_error(model.predict(X), y)
print("Tổng bình phương sai số trên tập mẫu:", mse)
print("Hệ số hồi quy:", model.coef_)
print("Sai số:", model.intercept_)
print(f"Công thức: [Nặng] = {model.coef_} x [Cao, Giới tính] + {model.intercept_}")
```



Dự báo sử dụng hồi quy tuyến tính

VÍ DỤ 2:

```
# dự báo một số tình huống
while True:
    x = float(input("Nhập chiều cao (nhập 0 để dừng): "))
    if x <= 0: break
    print("Nam giới cao", x, "cm, dự báo cân nặng",
model.predict([[x, 1]]))
    print("Nữ giới cao", x, "cm, dự báo cân nặng",
model.predict([[x, 0]]))
```



Tóm lại

- Học máy
- Một số loại bài toán học máy
- Thư viện học máy scikit-learn



**THANK
YOU**

THS. VƯƠNG XUÂN CHÍ
VXCHI@NTT.EDU.VN

0903 270 567