

# NHẬP MÔN KHOA HỌC DỮ LIỆU

## CHUẨN BỊ DỮ LIỆU

ThS. Vũ Hoài Thư



# Nội dung

- 1 Thu thập dữ liệu
- 2 Làm sạch dữ liệu
  - Xử lý giá trị thiếu
  - Xử lý giá trị sai hoặc không nhất quán
- 3 Co giãn và chuẩn hoá dữ liệu
- 4 Giảm chiều và biến đổi dữ liệu

# Chuẩn bị dữ liệu

Chuẩn bị dữ liệu là quá trình chuẩn bị dữ liệu thô sao cho phù hợp để xử lý và phân tích dữ liệu. Những bước chính bao gồm:

- Thu Thập dữ liệu
- Làm sạch dữ liệu
- Co giãn và chuẩn hoá dữ liệu
- Giảm chiều và biến đổi dữ liệu

## Thu thập dữ liệu

# Thu thập dữ liệu

- Thu thập dữ liệu là một quá trình tổng hợp tất cả các thông tin từ nhiều nguồn khác nhau và lưu trữ dữ liệu trong một hệ thống đã được thiết lập sẵn.
- Các nguồn dữ liệu có thể bao gồm các hình ảnh, văn bản, video, âm thanh, dữ liệu từ mạng xã hội, website trực tuyến hay các nguồn dữ liệu khác.
- Quá trình thu thập dữ liệu có thể được thực hiện thông qua nhiều phương pháp khác nhau ví dụ như khảo sát, phân tích dữ liệu, thăm dò ý kiến, từ dữ liệu lớn và các công cụ thu thập dữ liệu từ nhiều nguồn trực tuyến.

# Thu thập dữ liệu

Để dễ dàng cho việc áp dụng các phương pháp và kỹ thuật, học phần "Nhập môn khoa học dữ liệu" chủ yếu tập trung giới thiệu việc thu thập dữ liệu từ các nguồn website trực tuyến (Web crawler) như:

- Thu thập dữ liệu web có liên quan tới các thuật ngữ như nhện (Spiders ),
- Trình cào dữ liệu (Crawler),
- Bot (Robots).

=> Lý do lớn nhất khi tập trung vào thu thập dữ liệu trên các website trực tuyến là do các tài nguyên trên các website có sự phổ biến rộng rãi và luôn sẵn sàng trên mạng Internet.

# Thu thập dữ liệu

Có hai loại dữ liệu chính có sẵn trên web được sử dụng bởi các thuật toán khai phá dữ liệu:

1. *Web content information - Thông tin nội dung web*: Chứa hai thành phần có thể được khai thác cùng nhau hoặc độc lập.
2. *Web usage data - Dữ liệu sử dụng web*: Dữ liệu này tương ứng với các mô thức hoạt động của người dùng được kích hoạt bởi các ứng dụng Web.

# Thuật toán thu thập cơ bản

BasicCrawler( URLs:  $S$ , Thuật toán lựa chọn:  $A$ )

**begin:**

1:  $FrontierList = S$ ;

2: **Repeat**

3: Sử dụng thuật toán  $A$  để lựa chọn URL từ  $X \in FrontierList$ ;

4:  $FrontierList = FrontierList \setminus X$  ;

5: Tìm nạp dữ liệu từ URL  $X$  và thêm cơ sở dữ liệu lưu trữ;

6: Thêm tất cả các URL có liên quan trong  $X$  và nạp vào cuối  $FrontierList$ ;

7: **Until** Điều kiện kết thúc

**end**



# Thu thập ưu tiên

- Trong một trình thu thập ưu tiên, chỉ những trang web đáp ứng tiêu chí do người dùng xác định mới được thu thập dữ liệu.
- Các tiêu chí này có thể là từ khóa có trong trang, chủ đề văn bản (xác định bằng thuật toán học máy), tiêu chí địa lý về vị trí trang hoặc kết hợp các tiêu chí khác nhau.
- Các thay đổi trong trình thu thập ưu tiên được thực hiện với thuật toán thu thập cơ bản.

# Đa luồng

- Khi trình thu thập đưa ra yêu cầu về một URL và đợi yêu cầu đó, lúc đó hệ thống sẽ không hoạt động, không có công việc nào được thực hiện đến khi trình thu thập kết thúc. Điều này dẫn đến một sự lãng phí tài nguyên, do đó, đa luồng là một phương pháp để tăng tốc độ thu thập thông tin.
- Ý tưởng là sử dụng nhiều luồng của trình thu thập để cập nhật cấu trúc dữ liệu dùng chung cho các URL đã truy cập và kho lưu trữ dữ liệu của trang. Trong triển khai thực tế của các công cụ tìm kiếm lớn, trình thu thập được phân phối theo vùng địa lý với mỗi "*sub-crawl*" để thu thập các trang trong khoảng cách địa lý gần.

# Tránh các bẫy Spider

- Lý do chính khiến thuật toán thu thập luôn truy cập các trang Web khác nhau vì nó duy trì một danh sách các URL đã truy cập trước đó cho mục đích so sánh. Tuy nhiên, một số trang web mua sắm tạo các URL động trong đó trang cuối cùng đã truy cập được thêm vào cuối để cho phép máy chủ ghi lại chuỗi hành động của người dùng trong URL để phân tích trong tương lai.
- Ví dụ: khi người dùng nhấp vào liên kết cho trang 2 từ *http://www.examplesite.com/page1*, URL mới được tạo động sẽ là *http://www.examplesite.com/page1/page2*. Các trang được truy cập thêm sẽ tiếp tục được thêm vào cuối URL, ngay cả khi các trang này đã được truy cập trước đó.

# Phát hiện sự trùng lặp

- Một trong những vấn đề chính với các trang Web được thu thập nhiều lần. Do đó, trình thu thập phải có khả năng phát hiện các bản sao trùng lặp. Một cách tiếp cận được gọi là *shingling* thường được sử dụng để giải quyết vấn đề này.
- Một *K-shingle* từ một tài liệu văn bản là một chuỗi gồm  $k$  từ xuất hiện liên tiếp trong văn bản. Một *shingling* cũng có thể được xem giống như một *k-gram*.
- Thông thường, giá trị  $k$  nằm trong khoảng từ 5 đến 10 từ, tùy thuộc vào kích thước của tập văn bản và lĩnh vực áp dụng.

# Ví dụ

- Xét câu: "Lan có một chú cún con, lông của nó trắng như tuyết."
- Bộ 2-shingle được trích ra từ câu này là "Lan có", "có một", "một chú", "chú cún", "cún con", "con lông", "lông của", "của nó", "nó trắng", "trắng như" và "như tuyết".
- Gọi  $S_1$  và  $S_2$  là  $k$ -shingles được trích xuất từ hai tài liệu  $D_1$  và  $D_2$ . Sự tương đồng dựa trên shingling giữa  $D_1$  và  $D_2$  được tính bằng hệ số Jaccard giữa  $S_1$  và  $S_2$ :

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

# Áp dụng

- Triển khai trình thu thập dữ liệu bằng cách sử dụng thuật toán theo chiều rộng.
- Thực hành tìm hiểu một số website dữ liệu có cung cấp API như website về chỉ số thời tiết và dân số. Sau đó, thu thập và hiển thị dữ liệu thu thập được.
- Tìm hiểu và thu thập dữ liệu ba dự án thực tế sau:
  1. Champions League Table [ ESPN ]
  2. Product Tracker [ Amazon ]
  3. Scraper Application [ GUI ].

## Làm sạch dữ liệu

# Làm sạch dữ liệu

Việc làm sạch dữ liệu rất quan trọng bởi vì quá trình thu thập dữ liệu thường có sai sót. Có một số nguyên nhân gây ra thiếu giá trị hoặc sai số trong quá trình thu thập dữ liệu. Có một số nội dung quan trọng của việc làm sạch dữ liệu:

- Xử lý giá trị thiếu,
- Xử lý giá trị sai hoặc không nhất quán.



# Xử lý giá trị thiếu

Trước tiên cần phải hiểu rõ bản chất của dữ liệu, rồi sau đó sẽ đưa ra giải pháp phù hợp để xử lý giá trị thiếu.

- Loại bỏ hoàn toàn các bản ghi có dữ liệu bị thiếu (thường loại bỏ nếu số giá trị thiếu chỉ chiếm khoảng dưới 3% tổng số quan sát trong 1 biến nhất định).
- Ước tính các giá trị còn thiếu, từ là thay thế giá trị thiếu bằng một giá trị khác. Việc thay thế bằng giá trị nào sẽ phụ thuộc vào việc bản chất của giá trị thiếu trong những trường hợp đó là gì.
- Thiết kết các nhóm phân tích sao cho nó có thể hoạt động với các giá trị còn thiếu.

# Xử lý giá trị thiếu

Nếu cần phải thay thế missing values bằng một giá trị khác, thì nên thay thế bằng giá trị nào?

- Trường hợp biến có giá trị thiếu là biến số - numeric: Có thể thay thế bằng những giá trị như: 0, median, mean, sử dụng nội suy, tùy vào từng trường hợp nhất định.
- Trường hợp biến có giá trị thiếu là biến categorical: Có thể nhóm những trường hợp giá trị thiếu vào 1 nhóm, đặt tên cụ thể (ví dụ: Missing....)

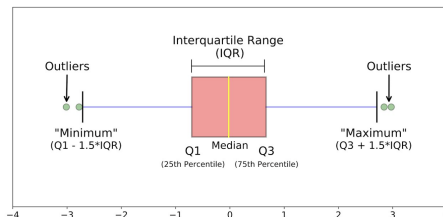
# Xử lý giá trị sai hoặc không nhất quán

Một số phương pháp chính được sử dụng để xử lý các giá trị sai hoặc không nhất quán như sau:

- Phát hiện sự không nhất quán: Điều này thường được thực hiện khi dữ liệu có sẵn từ các nguồn khác nhau ở các định dạng khác nhau.
- Kiến thức miền: Một lượng kiến thức miền rất lớn luôn có sẵn dưới dạng phạm vi của các thuộc tính hoặc quy tắc xác định mối quan hệ giữa các thuộc tính khác nhau.
- Phương pháp tập trung vào dữ liệu: Ta sử dụng hành vi thống kê của dữ liệu để phát hiện các giá trị ngoại lệ. Có rất nhiều phương pháp để phát hiện quan sát bất thường chẳng hạn là dựa vào giá trị trung bình và độ lệch tiêu chuẩn.

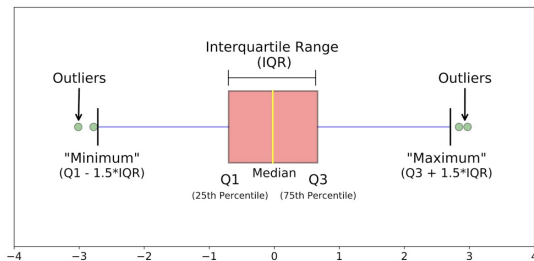
# Sử dụng biểu đồ hộp phát hiện giá trị ngoại lai

- Biểu đồ hộp mô tả một số đặc trưng quan trọng của tập dữ liệu như tâm, độ phân tán, mức độ đối xứng và cũng là một cách để phát hiện các quan sát bất thường. Biểu đồ hộp cho biết 3 điểm tứ phân vị  $Q_1$ ,  $Q_2$ ,  $Q_3$ , min, max trên một hộp chữ nhật.
- Một cạnh của hình chữ nhật nằm tại vị trí tứ phân vị thứ nhất  $Q_1$ , cạnh đối diện ở vị trí của điểm tứ phân vị thứ 3,  $Q_3$ , khoảng tứ phân vị  $IQR = Q_3 - Q_1$ .



# Sử dụng biểu đồ hộp phát hiện giá trị ngoại lai

Từ điểm  $Q_1$  ta vẽ đoạn thẳng theo hướng đi ra hướng giá trị nhỏ nhất của dữ liệu với độ dài là  $1,5 \times IQR$  và từ điểm  $Q_3$  vẽ đoạn thẳng đi ra hướng giá trị lớn nhất của dữ liệu với độ dài là  $1,5 \times IQR$  (các đoạn thẳng này được gọi là "đuôi dưới" và "đuôi trên"). Các quan sát nằm ngoài hình chữ nhật và hai đuôi này được biểu diễn bằng các ký hiệu "o" đơn lẻ. Các điểm nằm ngoài hai đuôi được xem là các quan sát bất thường.



# Sử dụng biểu đồ hộp phát hiện giá trị ngoại lai

**Ví dụ:** Cho dãy số liệu sau:

199, 201, 236, 269, 271, 278, 283, 291, 301, 303, 371

Tính  $Q_1, Q_2, Q_3, IQR$  và cho biết có giá trị ngoại lai không? Minh hoạ bằng biểu đồ hộp trên python.

## Sử dụng $\bar{x}$ và $s$ phát hiện giá trị ngoại lai

Nếu dữ liệu tuân theo phân bố chuẩn thì khoảng 95% số quan sát nằm trong khoảng  $\bar{x} - 2s, \bar{x} + 2s$  và 99,74% số quan sát sẽ nằm trong khoảng  $(\bar{x} - 3s, \bar{x} + 3s)$ , trong đó  $\bar{x}$  là trung bình mẫu và  $s$  là độ lệch chuẩn mẫu. Người ta xem các quan sát có giá trị không nằm trong khoảng  $(\bar{x} - 2s, \bar{x} + 2s)$  là quan sát bất thường. Như vậy để tìm quan sát bất thường theo phương pháp này ta thực hiện các bước sau:

**Bước 1:** Tính trung bình mẫu  $\bar{x}$  và độ lệch chuẩn mẫu  $s$ .

**Bước 2:** Tìm khoảng  $(\bar{x} - 2s, \bar{x} + 2s)$  và xác định bất thường.

Một số tài liệu coi quan sát bất thường là quan sát không thuộc khoảng  $(\bar{x} - 3s, \bar{x} + 3s)$ . Việc sử dụng khoảng  $(\bar{x} - 2s, \bar{x} + 2s)$  (khoảng có độ rộng  $4s$ ) hay khoảng  $(\bar{x} - 3s, \bar{x} + 3s)$  (khoảng có độ rộng  $6s$ ) tùy thuộc vào người dùng và tập dữ liệu.

# Sử dụng $\bar{x}$ và $s$ phát hiện giá trị ngoại lai

**Ví dụ:** Cho dãy số liệu sau

199, 201, 236, 269, 271, 278, 283, 291, 301, 303, 371.

Xác định bất thường của dãy số liệu trên sử dụng khoảng  $(\bar{x} - 2s, \bar{x} + 2s)$  và sử dụng khoảng  $(\bar{x} - 3s, \bar{x} + 3s)$



# Bài tập

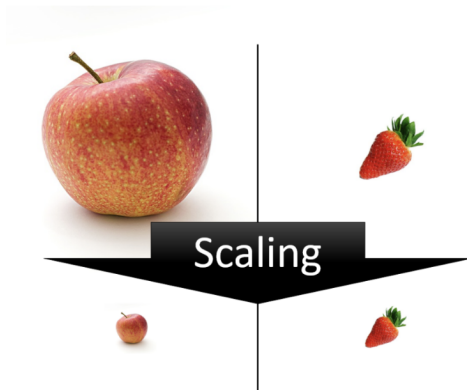
Viết chương trình python xác định giá trị ngoại lai cho các biến có trong dữ liệu Boston.csv theo hai cách:

- Sử dụng biểu đồ hộp
- Sử dụng  $\bar{x}$  và  $s$

## Co giãn và chuẩn hoá dữ liệu

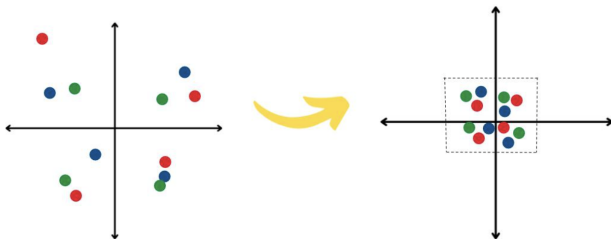
# Co giãn và chuẩn hoá dữ liệu

Trong nhiều tình huống, các thuộc tính khác nhau của dữ liệu sử dụng các thang co giãn khác nhau và do đó ta không thể so sánh các thuộc tính với nhau được.



# Co giãn và chuẩn hoá dữ liệu

Ví dụ: một thuộc tính như tuổi tác được vẽ theo tỷ lệ rất khác so với một thuộc tính như tiền lương. Thuộc tính tiền lương thường có độ lớn lớn hơn thuộc tính tuổi tác. Kết quả là, bất kỳ hàm tổng hợp nào được tính toán trên các thuộc tính khác nhau sẽ bị chi phối bởi thuộc tính có độ lớn lớn hơn.



# Co giãn và chuẩn hoá dữ liệu

Để giải quyết vấn đề này, người ta thường sử dụng phép *chuẩn hóa dữ liệu*. Giả sử thuộc tính thứ  $j$  có kỳ vọng  $\mu_j$  và độ lệch chuẩn  $\sigma_j$ . Khi đó, giá trị  $x_i^j$  của thuộc tính thứ  $j$  trong bản ghi  $\bar{X}_i$  có thể được chuẩn hóa như sau:

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j}$$

Phần lớn các giá trị được chuẩn hóa sẽ nằm trong đoạn  $[-3, 3]$  khi nó tuân theo phân phối chuẩn (Xem Nội dung chuẩn hoá dữ liệu ở Chương 1).

# Co giãn và chuẩn hoá dữ liệu

Cách tiếp cận thứ hai sử dụng *phép co giãn cực đại–cực tiểu* để ánh xạ tất cả các thuộc tính vào đoạn  $[0, 1]$ . Đặt  $\min_j$  và  $\max_j$  tương ứng là giá trị nhỏ nhất, giá trị lớn nhất của thuộc tính  $j$ . Khi đó, giá trị  $x_i^j$  của thuộc tính thứ  $j$  trong bản ghi  $\bar{X}_i$  được co giãn như sau:

$$y_i^j = \frac{x_i^j - \min_j}{\max_j - \min_j}$$

Cách tiếp cận này không hiệu quả khi các giá trị cực đại và cực tiểu là các giá trị ngoại lệ do một số lỗi trong quá trình thu thập dữ liệu.

# Co giãn và chuẩn hoá dữ liệu

Ví dụ: giả sử thuộc tính tuổi bị lỗi trong quá trình thu thập dữ liệu làm thêm một số 0 vào một tuổi, dẫn đến giá trị tuổi là 800 tuổi thay vì 80. Trong trường hợp này, hầu hết dữ liệu được co giãn theo thuộc tính tuổi sẽ nằm trong đoạn  $[0, 0.1]$ , do đó thuộc tính này có thể không được nhấn mạnh. Với các tình huống như này, ta nên dùng phép chuẩn hoá dữ liệu hoặc loại bỏ giá trị ngoại lai trước khi chuẩn hoá dữ liệu.

# Bài tập

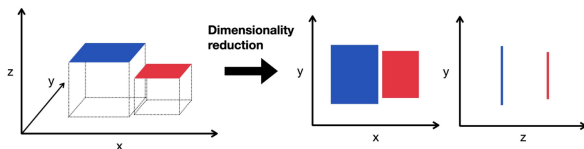
Viết chương trình python chuẩn hoá dữ liệu cho các biến định lượng trong dữ liệu Boston.csv theo hai cách tiếp cận đã học.



## Giảm chiều và biến đổi dữ liệu

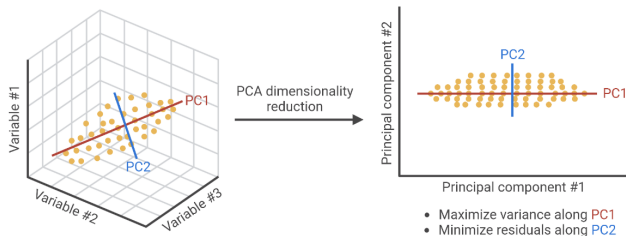
# Giảm chiều dữ liệu

- Mục tiêu của việc giảm chiều dữ liệu là để biểu diễn dữ liệu gọn hơn. Khi kích thước dữ liệu nhỏ hơn, việc áp dụng các thuật toán phức tạp và tốn kém về mặt tính toán sẽ dễ dàng hơn nhiều.
- Việc giảm chiều dữ liệu có thể là giảm về số lượng hàng (số bản ghi) hoặc về số lượng cột (số chiều). Giảm chiều dữ liệu có thể dẫn đến mất mát thông tin.
- Vì vậy, cần sử dụng những thuật toán để bù đắp cho việc mất thông tin do giảm chiều dữ liệu. Nhiều kiểu giảm chiều dữ liệu khác nhau được sử dụng trong các ứng dụng khác nhau.



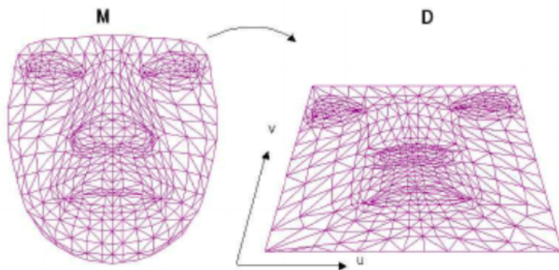
# Giảm chiều dữ liệu

Có nhiều phương pháp giảm chiều dữ liệu, một trong các phương pháp hiệu quả là phương pháp phân tích thành phần chính (Principal Component Analysis - PCA).



# Phương pháp phân tích thành phần chính (PCA)

Ý tưởng của thuật toán PCA là tìm một hệ không gian mới và tối đa hóa phương sai dữ liệu của không gian mới đó. Sau đó lựa chọn ra  $k$  chiều có phương sai lớn nhất (giả thuyết rằng dữ liệu càng phân tán, phương sai càng lớn thì càng có giá trị).



# Thuật toán PCA

- Bước 1: Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n$$

- Bước 2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\hat{x} = x_n - \bar{x}$$

- Bước 3: Tính ma trận hiệp phương sai:

$$S = \frac{1}{N} \hat{X} \hat{X}^T$$

- Bước 4: Tính các trị riêng và vector riêng có chuẩn bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.

# Thuật toán PCA (tiếp)

- Bước 5: Chọn  $K$  vector riêng ứng với  $K$  trị riêng lớn nhất để xây dựng ma trận  $U_K$  có các cột tạo thành một hệ trực giao.  $K$  vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hoá.
- Bước 6: Chiếu dữ liệu ban đầu đã chuẩn hoá  $\hat{X}$  xuống không gian con tìm được.
- Bước 7: Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới:

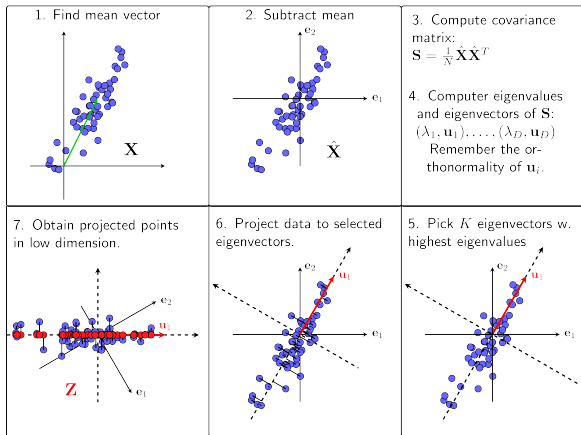
$$Z = U_K^T \hat{X}$$

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$x \approx U_K Z + \bar{x}$$

# Thuật toán PCA (Tiếp)

## PCA procedure



# Ví dụ thuật toán PCA

Giả sử có dữ liệu quan sát về chiều cao, cân nặng và chỉ số BMI của 5 người sau:

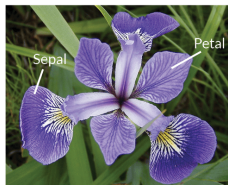
Tên	Chiều cao (cm)	Cân nặng (kg)	BMI
A	160	55	21.5
B	170	65	22.5
C	180	80	24.7
D	175	70	22.9
E	165	60	22.0

Sử dụng PCA, giảm chiều dữ liệu gốc về 2 chiều.



# Bài tập

Triển khai PCA với Python cho dữ liệu hoa IRIS.



**Iris Versicolor**



**Iris Setosa**



**Iris Virginica**