## **TRỰC QUAN HÓA DỮ LIỆU - THƯ VIỆN MATPLOTLIB**

### **Giới thiệu**

Data Visualization (trực quan hóa dữ liệu) là biểu diễn đồ họa của dữ liệu và thông tin. Bằng cách sử dụng [các yếu tố trực quan như biểu đồ, đồ thị và bản đồ](https://www.tableau.com/learn/articles/data-visualization/glossary), các công cụ trực quan hóa dữ liệu giúp chúng ta dễ dàng tiếp cận để xem và hiểu về dữ liệu như các xu hướng, ngoại lệ và kiểu dáng. Trong thế giới của dữ liệu lớn, các công cụ và công nghệ trực quan hóa dữ liệu là rất cần thiết để phân tích một lượng lớn thông tin và đưa ra các quyết định dựa trên dữ liệu. Python có các thư viện tuyệt vời để trực quan hóa dữ liệu. Sự kết hợp của Pandas , Numpy và Matplotlib có thể giúp tạo gần như tất cả các loại biểu đồ hình ảnh hóa. Trong bài đọc này, chúng ta sẽ bắt đầu với việc xem xét một số biểu đồ đơn giản và các thuộc tính khác nhau của biểu đồ.

### **Data Visualization là gì?**

Data Visualization được dịch là trực quan hóa dữ liệu, có nguồn gốc từ lĩnh vực [Thống kê](https://en.wikipedia.org/wiki/Statistics) và do đó thường được coi là một nhánh của [Thống kê mô tả](https://en.wikipedia.org/wiki/Descriptive_Statistics). Nó là cách biểu diễn dữ liệu dày đặc và phức tạp dưới dạng đồ họa như hình ảnh, biểu đồ, bảng đồ trực quan. Từ đó, truyền tải thông tin đến người xem một cách sinh động hơn, dễ hiểu hơn, giúp người xem dễ dàng nắm bắt những hiểu biết đầy đủ (insights), các yếu tố thống kê cần thiết để ra quyết định.

### **Tại sao cần Data Visualization?**

Các nghiên cứu khoa học chỉ ra rằng, con người phản ứng với hình ảnh tốt hơn rất nhiều so với các dạng trình bày khác như ký hiệu, chữ viết, con số. Chúng ta biểu diễn dữ liệu bằng hình ảnh giúp người dùng dễ dàng nhận biết được xu hướng, sự phân bố, sự tương quan giữa các đặc tính của dữ liệu. Một biểu đồ đơn giản cũng có thể biểu diễn thay thế cho rất nhiều dữ liệu. Một biểu đồ tốt có thể giúp những người không cần có kiến thức về chuyên môn, kỹ thuật cũng có thể nắm bắt được những yếu tố cơ bản của dữ liệu. Trực quan hóa dữ liệu có thể thể hiện dữ liệu thuộc nhiều loại và kích thước khác nhau: từ một vài điểm dữ liệu đến tập dữ liệu đa biến lớn. Cụ thể hơn, Data Visualization giúp chúng ta:

* Xác định bức tranh toàn cảnh: dễ dàng xem và phân tích một hoặc nhiều yếu tố, thành phần đặc trưng (features) trong dữ liệu lớn.
* Xác định xu hướng hiện tại và dự báo xu hướng tương lai của dữ liệu.
* Xác định mối liên hệ giữa các thành phần của dữ liệu hay giữa các tập hợp dữ liệu.
* Tăng khả năng truyền tải thông tin, đặc trưng của dữ liệu tới những đối tượng khác.
* Hỗ trợ đưa ra các quyết định dựa trên dữ liệu
* Đánh giá, so sánh thông tin dễ dàng thông qua các chỉ số được biểu diễn.

### **Giai đoạn nào cần Data Visualization?**

* Giai đoạn tiền xử lý dữ liệu: Trong giai đoạn này, việc mô hình hóa dữ liệu cung cấp các hiểu biết cơ bản về dữ liệu đầu vào mà chúng ta đang có như sự phân bố, sự khuyết thiếu, độ nhiễu, các giá trị ngoại lai, ...
* Giai đoạn trích chọn đặc trưng: Mô hình hóa dữ liệu ở giai đoạn này thường kết hợp với các thuật toán ranking/selection feature để đưa ra các biểu đồ đánh giá, so sánh, các bảng score cho từng feature từ đó có thể tính toán được sự phù hợp của feature đối với thuật toán, số lượng feature tối ưu.
* Giai đoạn đánh giá mô hình: Đây là giai đoạn mà việc mô hình hóa dữ liệu được sử dụng nhiều nhất giúp thể hiện kết quả của mô hình học máy, so sánh kết quả giữa các mô hình học máy với nhau, kết quả của quá trình parameter tunning... Đặc biệt việc mô hình hóa dữ liệu trong giai đoạn này còn phục vụ cho quá trình xây dựng các tài liệu business, maketing cho sản phẩm.

**Một số phương pháp biểu diễn phổ biến:**

* Biểu đồ
* Bảng
* Đồ thị
* Bản đồ
* Infographics
* Dashboard

**Một số bước cơ bản trong quá trình trực quan hóa dữ liệu**

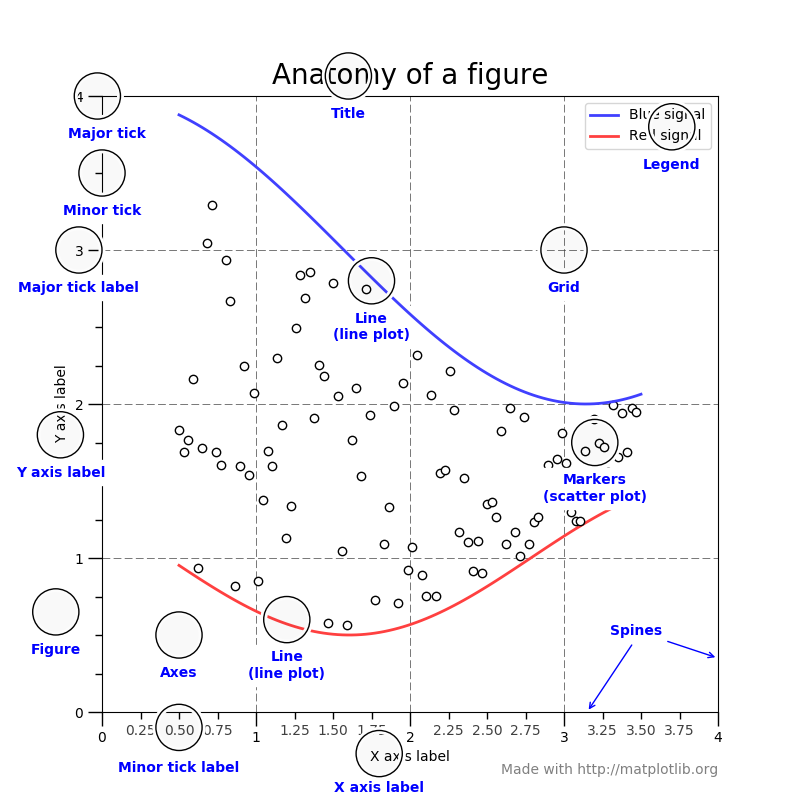
* **Xác định mục tiêu**: Điều quan trọng là phải hiểu rõ ràng về mục tiêu và thông tin cần truyền đạt của bạn. Điều này sẽ xác định loại dữ liệu nào là cần thiết, loại phân tích cần thiết và loại biểu đồ trực quan hiệu quả nhất để truyền đạt các khám phá hoặc phát hiện của bạn.

*Ví dụ: Phân tích giá gạo của một khu vực qua các năm bao gồm: so sánh giá, xu hướng thay đổi về giá, mức độ tương quan/ảnh hưởng của giá gas lên giá gạo.*

* **Thu thập dữ liệu và xác định các trường, thuộc tính cần thiết**: Dữ liệu có thể được thu thập từ nhiều tài nguyên. Tuy nhiên, không phải tất cả dữ liệu đều hữu ích. Chúng ta cần lọc ra những dữ liệu cần thiết để phục vụ mục tiêu đã đặt ra.
* **Chọn phương pháp biểu diễn phù hợp**: Mỗi phương pháp biểu diễn đều có mục đích và nguyên tắc sử dụng khác nhau. Tùy vào thông tin cần truyền đạt, chúng ta cần chọn biểu đồ, bảng, đồ thị, … một cách phù hợp.
* **Biểu diễn**: sử dụng một số công cụ/thư viện có sẵn để tạo và tinh chỉnh biểu đồ, đồ thị,…

**Thư viện Matplotlib**

Matplotlib được cho là module hỗ trợ trực quan hóa dữ liệu và đồ thị phổ biến nhất cho Python. Nó là một thư viện toàn diện để tạo các hình ảnh trực quan cả tĩnh và động. Thư viện này rất linh hoạt và có rất nhiều tiện ích mặc định được tích hợp sẵn giúp chúng ta dễ dàng tạo và tinh chỉnh nhiều loại biểu đồ. Một Matplotlib Figure có thể được phân loại thành nhiều phần như dưới đây:



* Figure: Như một cái cửa sổ chứa tất cả những gì các bạn sẽ vẽ trên đó.
* Axes: Thành phần chính của một figure là các axes (những vùng nhỏ hơn với không gian dữ liệu). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.
* Axis: Axes chứa 2 (hoặc 3 trong trường hợp 3D) Axis. Mỗi Axis là một chiều (trục) như một đối tượng đảm nhiệm việc tạo các giới hạn biểu đồ và tạo ra các tick (các dấu trên trục) và ticklabels (các chuỗi ghi nhãn cho các tick).
* Artist: Mọi thứ mà các bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.

**Hướng dẫn cài đặt Thư viện Matplotlib**

* Để cài đặt được thư viện này thì trước tiên các bạn cần phải cài Python trên máy tính của bạn trước. Nếu bạn sử dụng Python phiên bản từ 2.7.9 trở lên (đối với Python 2) hoặc 3.4 trở lên (với Python 3), mặc định pip đã được cài đặt sẵn cùng Python.
* Đối với hệ điều hành Window: Nhấn tổ hợp Windows + S để mở cửa sổ tìm kiếm => nhập vào từ khóa “cmd” => chọn Command Prompt => nhập “pip install matplotlib” để cài đặt version mới nhất.
* Đối với hệ điều hành khác

Debian / Ubuntu: sudo apt-get install python3-matplotlib

Fedora: sudo dnf install python3-matplotlib

Red Hat: sudo yum install python3-matplotlib

Arch: sudo pacman -S python-matplotlib

* Nếu các bạn sử dụng Anaconda thì Matplotlib đã được tích hợp sẵn.

#### Tạo biểu đồ đơn giản

* Import thư viện Matplotlib để sử dụng bằng câu lệnh:

import matplotlib.pyplot as plt

* Tạo dữ liệu mẫu

x = [2, 4, 6, 8, 10]  
y = [12.2, 16.3, 20.5, 25.4, 31.2]

Chú ý: kích thước của x và y phải bằng nhau.

* Vẽ biểu đồ:

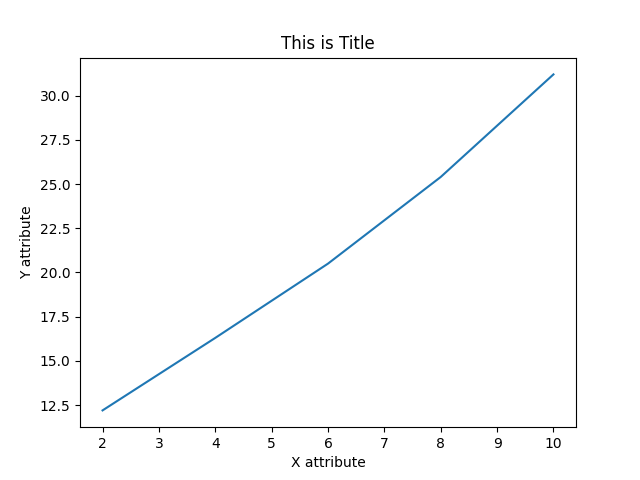
plt.plot(x, y)

plt.title('This is Title')  
plt.xlabel('X attribute')  
plt.ylabel('Y attribute')

plt.show()

Chú ý: mặc định vẽ một đồ thị bằng cách nối các điểm bằng đường thẳng (matplotlib.lines.Line2D).

Kết quả:



#### Một số tinh chỉnh biểu đồ đơn giản

* Thêm tiêu đề

plt.title('This is Title')

* Thay đổi thuộc tính tiêu đề:

plt.title('This is Title', fontdict = {'fontsize': 12,  
 'fontweight' : 8,  
 'verticalalignment': 'baseline',  
 'horizontalalignment': 'left'})

* Gán nhãn các trục

plt.xlabel('X attribute')  
plt.ylabel('Y attribute')

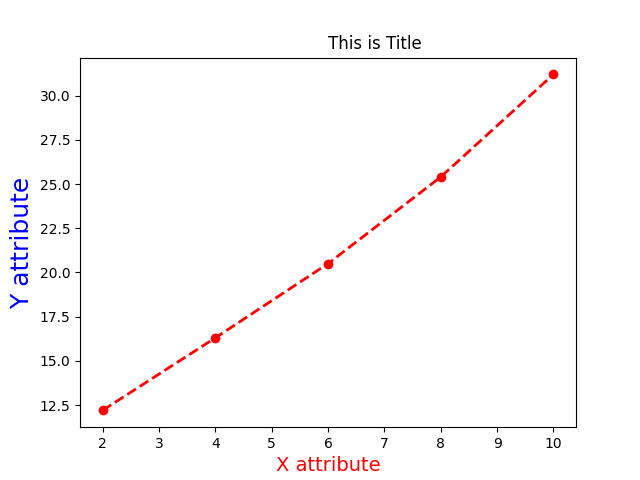
* Thay đổi thuộc tính nhãn các trục

plt.xlabel('X attribute', fontsize=14, color='red')  
plt.ylabel('Y attribute', fontsize=18, color='blue')

* Thay đổi thuộc tính đường

plt.plot(x, y, color='red', linewidth=2, linestyle='dashed', marker = 'o')

Kết quả:



### **Tổng kết**

Qua bài đọc này, chúng ta đã biết các khái niệm về Data Visualization và một số kỹ thuật cơ bản để bắt đầu việc trực quan hóa dữ liệu. Chúng ta cũng đã được tiếp cận thư viện Matplotlib thông qua một số ví dụ vẽ biểu đồ và tinh chỉnh biểu đồ đơn giản.