**THƯ VIỆN PANDAS - PYTHON**

## I. Pandas là gì?

- Pandas là một thư viện Python cung cấp các cấu trúc dữ liệu nhanh, mạnh mẽ, linh hoạt và mang hàm ý. Tên thư viện được bắt nguồn từ panel data (bảng dữ liệu). Pandas được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc (dạng bảng, đa chiều, có tiềm năng không đồng nhất) và dữ liệu chuỗi thời gian.

- Mục tiêu của Pandas là trở thành khối căn bản (building block) cấp cao cơ bản cho công việc thực tế, phân tích dữ liệu thế giới thực trong Python, và rộng hơn là trở thành công cụ thao tác / phân tích mã nguồn mở mạnh mẽ và linh hoạt nhất có sẵn trong bất kì loại ngôn ngữ lập trình nào.

## II. Vì sao bạn nên chọn Pandas?

**- Pandas rất phù hợp với nhiều loại dữ liệu khác nhau:**

+ Dữ liệu dạng bảng với các cột được nhập không đồng nhất, như trong bảng SQL hoặc bảng tính Excel.

+ Dữ liệu chuỗi thời gian theo thứ tự và không có thứ tự (không nhất thiết phải có tần số cố định).

+ Dữ liệu ma trận tuỳ ý (được nhập đồng nhất hoặc không đồng nhất) với nhãn hàng và cột.

+ Bất kì hình thức khác của các bộ dữ liệu quan sát / thống kê. Dữ liệu thực sự không cần phải được dán nhãn vào cấu trúc dữ liệu pandas.

+ Pandas được xây dựng dựa trên **Numpy**. Hai cấu trúc dữ liệu chính của Pandas là **Series** (1 chiều) và **DataFrame** (2 chiều) xử lý được phần lớn các trường hợp điển hình trong tài chính, thống kê, khoa học xã hội và nhiều lĩnh vực kỹ thuật.

**- Ưu điểm của Pandas:**

+ Dễ dàng xử lý dữ liệu mất mát, được biểu thị dưới dạng NaN, trong dữ liệu dấu phẩy động cũng như dấu phẩy tĩnh theo ý người dùng mong muốn: bỏ qua hoặc chuyển sang 0.

+ Khả năng thay đổi kích thước: các cột có thể được chèn và xoá khỏi DataFrame và các đối tượng chiều cao hơn.

+ Căn chỉnh dữ liệu tự động và rõ ràng: các đối tượng có thể được căn chỉnh rõ ràng với một bộ nhãn hoặc người dùng chỉ cần bỏ qua các nhãn và để Series, DataFrame,… tự động căn chỉnh dữ liệu cho bạn trong các tính toán.

+ Chức năng group by mạnh mẽ, linh hoạt để thực hiện các hoạt động kết hợp phân tách áp dụng trên các tập dữ liệu, cho cả dữ liệu tổng hợp và chuyển đổi.

+ Dễ dàng chuyển đổi dữ liệu rời rạc (ragged), chỉ mục khác nhau (differently-indexed) trong các cấu trúc dữ liệu khác của Python và Numpy thành các đối tượng DataFrame.

+ Cắt lát (slicing) thông minh dựa trên nhãn, lập chỉ mục ưa thích (fancy indexing) và tập hợp lại (subsetting) các tập dữ liệu lớn.

+ Gộp (merging) và nối (joining) các tập dữ liệu trực quan.

+ Linh hoạt trong định hình lại (reshaping) và xoay (pivoting) các tập dữ liệu.

+ Dán nhãn phân cấp (hierarchical) của các trục (có thể có nhiều nhãn trên mỗi đánh dấu).

+ Các công cụ IO mạnh mẽ để tải dữ liệu từ các tệp phẳng (flat file) như CSV và delimited, tệp Excel, cơ sở dữ liệu và lưu/tải dữ liệu từ định dạng HDF5 cực nhanh.

+ Chức năng theo chuỗi thời gian (time series) cụ thể: tạo phạm vi ngày và chuyển đổi tần số, thống kê cửa sổ di chuyển, dịch chuyển ngày và độ trễ.

+ Tích hợp tốt với các thư viện khác của Python như Sicpy, Matplotlib, Plotly,…

+ Hiệu suất tốt.

## III. Các lĩnh vực ứng dụng phổ biến của Pandas

- **Làm sạch và tiền xử lý dữ liệu**: Pandas là công cụ hàng đầu cho việc này. Nó cung cấp các hàm tiện lợi để xử lý giá trị thiếu, chuẩn hoá, biến đổi kiểu dữ liệu và tái cấu trúc dữ liệu thô.

- **Phân tích khám phá dữ liệu (EDA)**: Pandas giúp khám phá và hiểu dữ liệu dễ dàng hơn. Bạn có thể nhanh chóng tính toán thống kê mô tả, lọc dữ liệu theo điều kiện phức tạp và trực quan hoá nhờ tích hợp với các thưu viện như Matplotlib.

- **Kỹ thuật đặc trưng (Feature Engineering)**: Thư viện này hỗ trợ hiệu quả việc tạo đặc trưng mới. Người dùng có thể tính toán thống kê tổng hợp theo nhóm, tạo biến giả (dummy variables) và áp dụng các hàm biến đổi tuỳ chỉnh lên dữ liệu.

- **Phân tích chuỗi thời gian**: Pandas có các tính năng tích hợp mạnh mẽ cho dữ liệu chuỗi thời gian. Việc xử lý dữ liệu theo dấu thời gian, tái lấy mẫu (resampling) hay tính toán thống kê trượt (rolling statistics) trở nên đơn giản hơn nhiều.

- **Chuẩn bị dữ liệu cho khoa học dữ liệu**: Pandas là bước nền tảng để chuẩn bị dữ liệu cho mô hình học máy. Sau khi làm sạch, tiền xử lý và biến đổi bằng Pandas, dữ liệu có cấu trúc sẵn sàng cho các thư viện như Scikit-Learn hay TensorFlow.

## IV. Cài đặt và khai báo thư viện Pandas

**1. Cài đặt:**

- Sử dụng pip và gõ lệnh: pip install pandas

- Hoặc bằng Anaconda, dùng lệnh: conda install pandas

- Lưu ý: Bạn cần cài đặt thư viện Numpy trước (nếu bạn cài bằng Anaconda thì Numpy đã có sẵn).

**2. Khai báo:**

import pandas as pd

## V. Thao tác với cấu trúc dữ liệu cơ bản của Pandas

Pandas có 2 cấu trúc dữ liệu cơ bản là:

- Series (1 chiều)

- DataFrame (2 chiều)

- Panel (3 chiều) từng là một cấu trúc dữ liệu trong Pandas trước khi bị gỡ bỏ từ phiên bản 0.25.

**1. Series:**

Series([data, index, dtype, name, copy,...])

- Series là mảng 1 chiều giống như mảng Numpy, hay như 1 cột của 1 bảng, nhưng nó bao gồm thêm 1 bảng đánh label. Series có thể được khởi tạo thông qua Numpy, kiểu dictionary hoặc các dữ liệu vô hướng bình thường. Series có nhiều thuộc tính như index, array, values, dtype,… Bạn có thể thực hiện chuyển đổi Series sang dạng dtype xác định, tạo bảng copy, trả về dạng bool của 1 thành phần, chuyển Series từ DatetimeIndex sang PeriodIndex,…

- Một số ví dụ về thao tác với Series:

**+ Tạo Series:**

**VD1:** Không truyền index:

A black rectangle with white text

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

Pandas sẽ mặc định truyền index từ 0 đến len(data) - 1.

**VD2:** Có truyền index:

A black background with white text

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD3:** Tạo Series từ dictionary:

A black background with colorful letters

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

Chúng ta tạo dict có index a, b, d, f, g. Sau đó tạo Series từ dict data này nhưng các index c và e không có trong dict nên dữ liệu tại các index này bị thiếu (missing data). Pandas hiển thị NaN để báo các dữ liệu này bị trống.

**VD4:** Tạo Series từ Scalar:

Nếu dữ liệu là một giá trị Scalar, index phải được cung cấp. Giá trị sẽ được lặp lại để phù hợp với độ dài của index.

A black background with orange and white text

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**+ Truy cập dữ liệu từ Series với index và vị trí:**

Truy cập dữ liệu của Series tương tự với ndarray trong Numpy.

**VD5:** Lấy dữ liệu tại index cụ thể:

A black background with colorful text

AI-generated content may be incorrect.

Output:



**VD6:** Lấy dữ liệu từ đầu đến vị trí index cụ thể:

A computer screen with numbers and symbols

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD7:** Lấy dữ liệu theo vị trí 2 dữ liệu đầu:

A computer screen with numbers and symbols

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD8:** Lấy 3 dữ liệu cuối:

A computer screen with numbers and symbols

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**+ Chuyển đổi sang dạng khác:**

**VD9:** Lấy dạng array của Series bằng numpy.asarray:

A computer screen with numbers and symbols

AI-generated content may be incorrect.

Output:



**2. DataFrame:**

DataFrame([data, index, columns, dtype, copy])

- DataFrame là cấu trúc dữ liệu gắn nhãn 2 chiều với các cột và hàng như bảng tính (spreadsheet) hoặc bảng (table). Giống như Series, DataFrame có thể chứa bất kì loại dữ liệu nào. Một điều quan trọng cần làm nổi bật là tất cả các cột trong khung dữ liệu là Series Pandas. Vì vậy, một DataFrame là sự kết hợp của nhiều Series đóng vai trò như các cột. DataFrame được sử dụng rộng rãi và là một trong những cấu trúc dữ liệu quan trọng nhất.

- Một số ví dụ về thao tác với DataFrame:

**+ Tạo DataFrame:**

**VD1:** Tạo DataFrame từ dict các Series 1:

A computer screen shot of a program code

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD2:** Tạo DataFrame từ dict các Series 2:

A computer screen with text

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

Ngoài ra còn nhiều cách tạo DataFrame khác như từ dict các ndarray/list, từ list của dict, từ một Series,…

**+ Các thao tác chọn, thêm, xoá cột:**

**VD3:** Chọn cột (column selection):

A computer screen shot of a code

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD4:** Một số cách thêm cột (column addtion):

A screenshot of a computer program

AI-generated content may be incorrect.

Output:

A black screen with white and orange text

AI-generated content may be incorrect.

**VD5:** Xoá cột (column deletion):

Có thể xoá cột bằng lệnh def hoặc hàm pop.

A computer screen shot of a program

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**+ Lập chỉ mục / lựa chọn:**

Những điều cơ bản của việc lập chỉ mục như sau:

A white background with black lines

AI-generated content may be incorrect.

**VD1:** Chọn dòng theo label:

A computer screen shot of a code

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD2:** Chọn dòng theo vị trí nguyên:

A computer screen shot of a code

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.

**VD3:** Cắt (slice) các dòng:

A computer screen shot of a code

AI-generated content may be incorrect.

Output:

A black rectangle with white text

AI-generated content may be incorrect.