Thư viện Python và R cho khoa học dữ liệu

**Python** và **R** là những ngôn ngữ được ưa thích dành cho **khoa học dữ liệu** do hệ sinh thái thư viện phong phú, hỗ trợ cộng đồng rộng rãi và các công cụ mạnh mẽ để phân tích thống kê, thao tác dữ liệu, học máy và trực quan hóa. Bạn có thể chọn Python hoặc R cho Khoa học dữ liệu. Vì vậy, nếu bạn muốn biết về tất cả các thư viện được sử dụng trong Python và R cho Khoa học dữ liệu thì bài viết này là dành cho bạn. Trong bài viết này, tôi sẽ hướng dẫn bạn tất cả các thư viện Python và R dành cho Khoa học dữ liệu.

# Tại sao Python và R là ngôn ngữ ưa thích cho khoa học dữ liệu?

Python và R là những ngôn ngữ được ưa thích dành cho khoa học dữ liệu do hệ sinh thái thư viện phong phú, hỗ trợ cộng đồng rộng rãi và các công cụ mạnh mẽ để phân tích thống kê, thao tác dữ liệu, học máy và trực quan hóa. Cú pháp và khả năng đọc của Python làm cho nó thân thiện với người mới bắt đầu và thuận lợi cho việc cộng tác mã hóa. Khả năng tạo ra các biểu đồ chất lượng xuất bản của R rất thuận lợi cho việc khám phá và trình bày dữ liệu.

Cả hai ngôn ngữ đều có cộng đồng mạnh mẽ, đóng góp tích cực vào việc phát triển và bảo trì thư viện, giúp việc tìm giải pháp cho các vấn đề khoa học dữ liệu trở nên dễ dàng hơn. Họ cũng có nguồn tài nguyên khổng lồ, bao gồm các hướng dẫn, tài liệu và diễn đàn trực tuyến, cho phép các cá nhân nhanh chóng tìm hiểu và áp dụng các khái niệm Khoa học dữ liệu.

Vì vậy, mức độ phổ biến của Python và R trong Khoa học dữ liệu bắt nguồn từ các thư viện toàn diện, tính dễ sử dụng và sự hỗ trợ cộng đồng mạnh mẽ, khiến chúng trở thành công cụ linh hoạt để các nhà khoa học dữ liệu phân tích, thao tác và trực quan hóa dữ liệu một cách hiệu quả.

# Thư viện Python và R cho khoa học dữ liệu

Trong phần này, tôi sẽ hướng dẫn bạn qua tất cả các thư viện Python và R được sử dụng cho Khoa học dữ liệu trong ngành.

## Thư viện Python và R để thao tác và chuyển đổi dữ liệu

Thao tác và chuyển đổi dữ liệu đề cập đến việc sửa đổi và định hình lại bộ dữ liệu theo nhu cầu phân tích cụ thể. Nó liên quan đến việc thực hiện các hoạt động như lọc, sắp xếp, hợp nhất, tổng hợp và tái cấu trúc dữ liệu để rút ra những hiểu biết có ý nghĩa và chuẩn bị cho phân tích sâu hơn.

Trong Python, một trong những thư viện chính để thao tác dữ liệu là **Pandas** . Pandas cung cấp một tập hợp cấu trúc dữ liệu phong phú, chẳng hạn như DataFrames, cho phép các nhà phân tích tổ chức và thao tác dữ liệu một cách hiệu quả. Một thư viện khác thường được sử dụng trong Python để thao tác dữ liệu là **NumPy** . NumPy cung cấp khả năng tính toán số mạnh mẽ và hỗ trợ các mảng và ma trận đa chiều.

Trong R, một trong những thư viện chính để thao tác dữ liệu là **dplyr**. Dplyr cung cấp một tập hợp các chức năng được thiết kế đặc biệt cho các tác vụ thao tác dữ liệu hiệu quả. Và gói **tidyr** trong R bổ sung cho dplyr bằng cách cung cấp các chức năng sắp xếp và định hình lại dữ liệu.

Vì vậy, dưới đây là các thư viện Python và R để thao tác và chuyển đổi dữ liệu với tài nguyên học tập của chúng:

* [**NumPy**](https://numpy.org/doc/stable/user/absolute_beginners.html):
* [**Pandas**](https://pandas.pydata.org/docs/user_guide/10min.html#min)
* [**dplyr**](https://dplyr.tidyverse.org/articles/dplyr.html)
* [**tidyr**](https://tidyr.tidyverse.org/articles/tidy-data.html)

## Thư viện Python và R để phân tích thống kê

Phân tích thống kê là một nhánh của phân tích dữ liệu tập trung vào việc rút ra những hiểu biết và kết luận có ý nghĩa từ dữ liệu bằng các kỹ thuật thống kê. Nó liên quan đến việc khám phá dữ liệu, xác định các mẫu, kiểm tra các giả thuyết, ước tính các tham số và đưa ra suy luận về quần thể.

Trong Python, một trong những thư viện nổi bật để phân tích thống kê là **SciPy** . SciPy được xây dựng dựa trên chức năng của NumPy và cung cấp các mô-đun bổ sung cho tính toán khoa học và phân tích thống kê. Nó cung cấp các chức năng cho các bài kiểm tra thống kê phổ biến, bao gồm T-Test, ANOVA, tương quan và hồi quy. Một thư viện khác thường được sử dụng trong Python để phân tích thống kê là **Pandas** . Nó cung cấp khả năng tổng hợp, nhóm và thống kê tóm tắt dữ liệu, giúp thuận tiện cho EDA và thống kê mô tả.

Trong R, phân tích thống kê là chức năng cốt lõi của ngôn ngữ. R cung cấp một bộ đầy đủ các hàm và gói tích hợp để phân tích thống kê. Gói cơ sở R cung cấp các chức năng tính toán thống kê cơ bản, phân phối xác suất, kiểm tra giả thuyết, v.v. Nó cung cấp các chức năng cho mô hình tuyến tính, mô hình phi tuyến tính, phân tích chuỗi thời gian và phân tích đa biến.

Vì vậy, dưới đây là một số tài nguyên để tìm hiểu thư viện Python và cách phân tích thống kê của R:

* [**SciPy**](https://docs.scipy.org/doc/scipy/tutorial/index.html)
* [**Pandas**](https://pandas.pydata.org/docs/user_guide/10min.html#)
* [**R for Statistical Analysis**](https://learningstatisticswithr.com/book/)

## Thư viện Python và R để trực quan hóa dữ liệu

Trực quan hóa dữ liệu là một khía cạnh quan trọng của phân tích dữ liệu liên quan đến việc biểu diễn dữ liệu ở các định dạng trực quan như biểu đồ, đồ thị và sơ đồ. Nó cho phép các nhà phân tích truyền đạt hiệu quả các thông tin, mẫu và mối quan hệ phức tạp có trong dữ liệu cho cả các bên liên quan về kỹ thuật và phi kỹ thuật.

Trong Python, một trong những thư viện để trực quan hóa dữ liệu là **Matplotlib** . Matplotlib cung cấp một bộ sưu tập toàn diện các hàm và lớp để tạo ra nhiều hình ảnh trực quan khác nhau, bao gồm biểu đồ đường, biểu đồ thanh, biểu đồ phân tán, biểu đồ và bản đồ nhiệt.

Để trực quan hóa dữ liệu trong R, thư viện nổi bật là **ggplot2** . Dựa trên ngữ pháp của khái niệm đồ họa, ggplot2 cung cấp một khuôn khổ mạnh mẽ và linh hoạt để tạo các hình ảnh trực quan phức tạp.

Cả Python và R đều cung cấp các thư viện trực quan hóa dữ liệu tương tác. **Plotly** , với tư cách là một thư viện Python, cho phép các nhà phân tích tạo ra các hình ảnh trực quan tương tác và dựa trên web có thể được nhúng trong các ứng dụng web hoặc notebooks. Plotly cũng có sẵn trong R, cùng với gói “shiny”, cho phép các nhà phân tích xây dựng bảng điều khiển tương tác và ứng dụng web bằng hình ảnh trực quan.

Vì vậy, dưới đây là các thư viện Python và R để trực quan hóa dữ liệu với tài nguyên học tập của chúng:

* [**Matplotlib**](https://matplotlib.org/stable/tutorials/introductory/quick_start.html)
* [**ggplot2**](https://ggplot2.tidyverse.org/reference/index.html)
* [**Plotly for Python**](https://plotly.com/python/)
* [**Plotly for R**](https://plotly.com/r/)

## Thư viện Python và R để lập mô hình và đánh giá

Lập mô hình và đánh giá là các thành phần không thể thiếu trong quy trình làm việc của Khoa học dữ liệu, nơi bạn xây dựng các mô hình thống kê hoặc học máy để hiểu rõ hơn, đưa ra dự đoán hoặc phân loại dữ liệu. Quá trình này bao gồm việc lựa chọn một mô hình phù hợp, đào tạo nó dựa trên dữ liệu có sẵn, đánh giá hiệu suất của nó và tinh chỉnh nó để có kết quả tối ưu.

Trong Python, một trong những thư viện nổi bật để lập mô hình và đánh giá là **Scikit-learn** . Scikit-learn cung cấp một bộ sưu tập toàn diện các thuật toán và công cụ học máy để phân loại, hồi quy, phân cụm và giảm kích thước. Một thư viện khác thường được sử dụng trong Python để lập mô hình và đánh giá là **StatsModels** . StatsModels tập trung vào mô hình thống kê, cung cấp nhiều kỹ thuật thống kê để kiểm tra giả thuyết, hồi quy tuyến tính, phân tích chuỗi thời gian, v.v.

Trong R, gói **caret** được sử dụng để lập mô hình và đánh giá. Caret (Đào tạo phân loại và hồi quy) cung cấp giao diện thống nhất cho một bộ thuật toán học máy đa dạng để so sánh và huấn luyện các mô hình. Nó hỗ trợ các nhiệm vụ như phân loại, hồi quy và lựa chọn tính năng. Dấu mũ cung cấp các chức năng tiền xử lý dữ liệu, đào tạo mô hình, đánh giá hiệu suất và điều chỉnh siêu tham số.

Vì vậy, dưới đây là các thư viện Python và R để lập mô hình và đánh giá bằng tài nguyên học tập của chúng:

* [**Scikit-learn**](https://scikit-learn.org/stable/getting_started.html)
* [**StatsModels**](https://www.statsmodels.org/stable/gettingstarted.html)
* [**caret**](https://topepo.github.io/caret/)

## Thư viện Python và R để web scraping

Web scraping có nghĩa là trích xuất dữ liệu từ các trang web bằng cách tự động điều hướng các trang web, truy xuất nội dung của chúng và phân tích thông tin mong muốn. Nó cho phép các nhà phân tích thu thập lượng lớn dữ liệu từ các nguồn trực tuyến khác nhau một cách hiệu quả và tự động hóa quá trình trích xuất.

Trong Python, một trong những thư viện chính để web scraping là **BeautifulSoup**. BeautifulSoup cho phép bạn phân tích cú pháp các tài liệu HTML và XML, giúp trích xuất các phần tử và dữ liệu cụ thể từ các trang web dễ dàng hơn.

Trong R, thư viện chính để web scraping là **rvest** . Rvest cung cấp các chức năng tương tự như BeautifulSoup trong Python, cho phép bạn phân tích cú pháp và trích xuất dữ liệu từ tài liệu HTML.

Vì vậy, dưới đây là các thư viện Python và R để tìm kiếm trên web với các tài nguyên học tập của chúng:

* [**BeautifulSoup**](https://www.crummy.com/software/BeautifulSoup/bs4/doc/#quick-start)
* [**rvest**](https://rvest.tidyverse.org/articles/rvest.html)

## Thư viện Python và R cho Mạng thần kinh và Học sâu

Mạng lưới thần kinh và học sâu là những kỹ thuật tiên tiến trong Học máy bắt chước cấu trúc và hoạt động của bộ não con người. Những kỹ thuật này liên quan đến việc xây dựng các mạng lưới phức tạp gồm các nơ-ron nhân tạo được kết nối với nhau, cho phép máy móc học hỏi và đưa ra dự đoán từ các mẫu và dữ liệu phức tạp.

Trong Python, một trong những thư viện nổi bật dành cho mạng lưới thần kinh và học sâu là **TensorFlow** . TensorFlow cung cấp một hệ sinh thái toàn diện để xây dựng và đào tạo mạng lưới thần kinh, bao gồm cả các mô hình học sâu. Nó cung cấp API cấp cao có tên **Keras**, giúp đơn giản hóa quá trình xây dựng mạng thần kinh bằng cách cung cấp giao diện thân thiện với người dùng. Một thư viện phổ biến khác để học sâu bằng Python là **PyTorch** . PyTorch được biết đến rộng rãi nhờ biểu đồ tính toán động, cho phép thiết kế và tùy chỉnh mô hình linh hoạt và trực quan.

Trong R, thư viện chính cho mạng nơron và học sâu là **TensorFlow** .

Vì vậy, dưới đây là các thư viện Python và R dành cho Mạng thần kinh và Học sâu với các tài nguyên học tập của chúng:

* [**TensorFlow**](https://www.tensorflow.org/tutorials/quickstart/beginner)
* [**PyTorch**](https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html)
* [**TensorFlow**](https://tensorflow.rstudio.com/tutorials/) cho R

# Kết luận

Dưới đây là tất cả các thư viện Python cần thiết cho Khoa học dữ liệu:

1. NumPy
2. Pandas
3. SciPy
4. Matplotlib
5. Plotly
6. Scikit-learn
7. StatsModels
8. BeautifulSoup
9. TensorFlow
10. PyTorch

Và dưới đây là tất cả các thư viện R cần thiết cho Khoa học dữ liệu:

1. dplyr
2. tidyr
3. ggplot2
4. Plotly
5. Caret
6. Rvest
7. TensorFlow

Tôi hy vọng bạn thích bài viết này về tất cả các thư viện Python và R dành cho Khoa học dữ liệu