|  |  |  |
| --- | --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |  |

**ĐỒ ÁN MÔN HỌC**

**TÌM HIỂU VỀ CÔNG CỤ MÃ NGUỒN SCRAPY VÀ THU THẬP DỮ LIỆU SẢN PHẨM TRÊN SÀN THƯƠNG MẠI ĐIỆN TỬ**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : Ths.Lê Nhật Tùng

Sinh viên thực hiện :2286400001 Phạm Quốc An

2286400481 Nguyễn Thị Thu Ngân

TP. Hồ Chí Minh, 2024

|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC**

**TÌM HIỂU VỀ CÔNG CỤ MÃ NGUỒN SCRAPY VÀ THU THẬP DỮ LIỆU SẢN PHẨM TRÊN SÀN THƯƠNG MẠI ĐIỆN TỬ**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : Ths.Lê Nhật Tùng

Sinh viên thực hiện :2286400001 Phạm Quốc An

2286400481 Nguyễn Thị Thu Ngân

TP. Hồ Chí Minh, 2024

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

TPHCM, Ngày… tháng 10 năm 2024

**Giáo viên hướng dẫn**

(Ký tên, đóng dấu)

**LỜI CAM ĐOAN**

Chúng tôi, Phạm Quốc An và Nguyễn Thị Thu Ngân, xin cam đoan rằng:

Toàn bộ nội dung của bài báo cáo này là kết quả từ quá trình nghiên cứu và làm việc chung của cả hai chúng tôi. Các thông tin được trình bày trong báo cáo đều được thu thập từ các nguồn đáng tin cậy và đã được xử lý cẩn thận.

Chúng tôi đảm bảo rằng không có bất kỳ hành vi sao chép hay sử dụng thông tin không đúng đắn nào từ các nguồn khác. Tất cả các tài liệu tham khảo đã được ghi nguồn rõ ràng và tuân thủ đúng các quy định về trích dẫn học thuật.

Bài báo cáo này là công trình nghiên cứu chung của chúng tôi và chưa từng được nộp hoặc công bố ở bất kỳ đâu trước đây. Chúng tôi cam kết hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của nội dung báo cáo này.

Chúng tôi hy vọng rằng bài báo cáo này sẽ mang đến một góc nhìn tổng quát và chi tiết về chủ đề “Tìm hiểu về công cụ mã nguồn Scrapy và thu thập dữ liệu....” và đồng thời đóng góp phần nhỏ vào việc nghiên cứu trong lĩnh vực nghiên cứu này.

TPHCM, ngày… tháng 10 năm 2024

**Sinh viên**

Phạm Quốc An,

Nguyễn Thị Thu Ngân

**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT VÀ TỪ KHÓA**

**MỤC LỤC**

**DANH SÁCH BẢNG**

**DANH SÁCH HÌNH VẼ**

**CHƯƠNG 1: TỔNG QUAN**

* 1. **Giới thiệu đề tài**

Trong thời đại thị trường trực tuyến phát triển mạnh mẽ, nghiên cứu và thu thập dữ liệu từ cửa hàng Grandcru trực tuyến đang trở thành công cụ quan trọng giúp doanh nghiệp hiểu rõ hơn về hành vi của người tiêu dùng và từ đó tối ưu hóa chiến lược kinh doanh. Đề tài "Phân tích và thu thập dữ liệu trên sàn thương mại điện tử hướng đến mục tiêu cung cấp một cách tiếp cận khoa học và có hệ thống để hỗ trợ doanh nghiệp cải thiện hiệu quả kinh doanh và tăng cường sức cạnh tranh trên thị trường. Ví dụ như thu thập dữ liệu về lượng truy cập trang web, thời gian mỗi khách hàng dành tại cửa hàng trực tuyến hay thông tin về các sản phẩm được mua nhiều nhất. Từ những thông tin này, doanh nghiệp có thể điều chỉnh chiến lược tiếp thị, cải thiện trải nghiệm mua sắm trực tuyến hay thậm chí là tối ưu hóa dịch vụ khách hàng. Những phân tích và dữ liệu này chính là chìa khóa giúp doanh nghiệp đạt được thành công và phát triển bền vững trong một thị trường cạnh tranh ngày càng khốc liệt.

* 1. **Nhiệm vụ của đề tài**

Trong dự án "Thu thập và phân tích các cửa hàng trực tuyến trên thương mại Grandcru, chúng tôi sẽ sử dụng công nghệ thu thập dữ liệu để tìm kiếm thông tin từ trang web thương mại điện tử. Điều này giúp chúng tôi thu thập dữ liệu về sản phẩm, đánh giá của khách hàng và các yếu tố liên quan khác. Sau khi thu thập dữ liệu, chúng tôi sẽ áp dụng các phương pháp phân tích để khám phá xu hướng của người tiêu dùng. Ví dụ, chúng tôi có thể xem xét cách giá cả ảnh hưởng đến quyết định mua hàng của khách hàng hoặc đánh giá của họ về sản phẩm. Thông tin này sẽ giúp chúng tôi cải thiện chiến lược kinh doanh, tối ưu hóa trải nghiệm mua sắm trực tuyến và tăng hiệu quả hoạt động thương mại điện tử.

***1.2.1 Tính cấp thiết của đề tài***

Ngày nay, sự phát triển nhanh chóng của công nghệ và sự phổ biến của thương mại điện tử đã thay đổi hoàn toàn cách chúng ta mua sắm. Các cửa hàng trực tuyến không chỉ đơn thuần là một nền tảng mua sắm mà còn là không gian cạnh tranh khốc liệt giữa các thương hiệu. Thương mại điện tử, một trong những thương hiệu bán lẻ điện máy lớn nhất Việt Nam, đang phải đối mặt với không ít áp lực từ các đối thủ cạnh tranh và sự thay đổi liên tục của hành vi người tiêu dùng. Với quy mô lớn và mạng lưới bán hàng trên toàn quốc, thương mại điện tử cần tối ưu hóa không chỉ chiến lược kinh doanh mà còn cả quản lý hoạt động, phân phối sản phẩm và cải thiện trải nghiệm của khách hàng. Không chỉ vậy, các yếu tố như giá cả và đánh giá của người dùng trên các nền tảng thương mại điện tử cũng ngày càng ảnh hưởng đến quyết định mua hàng. Điều này đặt ra yêu cầu cấp thiết đối với thương mại là tìm ra các phương pháp khoa học và hiệu quả để khai thác dữ liệu, từ đó xây dựng các chiến lược cụ thể để duy trì và phát triển lợi thế cạnh tranh.

Ngoài ra, khi thị trường tiêu dùng ngày càng đa dạng và kỳ vọng của khách hàng ngày càng tăng, việc phân tích dữ liệu lớn từ các hoạt động trực tuyến như đánh giá sản phẩm và hành vi mua sắm trở nên vô cùng quan trọng. Những thông tin này không chỉ giúp doanh nghiệp tối ưu hóa các chiến lược tiếp thị mà còn góp phần nâng cao chất lượng dịch vụ và sự hài lòng của khách hàng.

Từ những lý do trên, có thể thấy việc nghiên cứu, triển khai các phương pháp phân tích, thu thập dữ liệu từ hoạt động thương mại điện tử không chỉ là yêu cầu cấp thiết mà còn là chìa khóa giúp thương mại điện tử thích nghi và phát triển trong thời đại công nghệ số đang lên mạnh mẽ.

***1.2.2 Ý nghĩa khoa học và thực tiễn của đề tài***

Ý nghĩa khoa học: Nghiên cứu này đóng góp vào lĩnh vực khoa học dữ liệu và thương mại điện tử thông qua việc áp dụng các công cụ như web scraping để phân tích hành vi tiêu dùng. Việc này không chỉ giúp khám phá các xu hướng mới ra ứng dụng tiềm năng trong nghiên cứu marketing và quản trị.

Ý nghĩa thực tiễn: trong thực tiễn, việc phân tích dữ liệu giúp doanh nghiệp tối ưu hóa chiến lược kinh doanh, từ giá cả đến dịch vụ khách hàng. Dữ liệu thu thập được giúp cải thiện trải nghiệm người dùng và nâng cao khả năng cạnh tranh trên thị trường, đồng thời phát hiện và xử lý các vấn đề sớm hơn, giúp tăng doanh thu và hiệu quả vận hành.

**1.3 Mục tiêu**

***1.3.1 Mục tiêu tổng quát***

Đề tài nhằm cung cấp một phương pháp phân tích và thu thập dữ liệu trên sàn thương mại điện tử Grandcru, giúp hiểu rõ hơn về hành vi tiêu dùng và các yếu tố ảnh hưởng đến quyết định mua hàng. Từ đó đưa ra các đề xuất tối ưu hóa chiến lược kinh doanh, cải thiện trải nghiệm người dùng và nâng cao khả năng cạnh tranh của doanh nghiệp trong thị trường thương mại điện tử đang phát triển mạnh mẽ.

***1.3.2 Mục tiêu cụ thể***

Với bài viết nghiên cứu này, chúng tôi sẽ thu thập dữ liệu về sản phẩm và đánh giá khách hàng trên thương mại điện tử, sau đó phân tích hành vi mua sắm để hiểu rõ hơn về xu hướng tiêu dùng và sự ảnh hưởng của các chương trình khuyến mãi. Tiếp theo, nghiên cứu sẽ đánh giá mức độ hài lòng của khách hàng, từ đó đề xuất các chiến lược kinh doanh tối ưu, bao gồm cải thiện dịch vụ, điều chỉnh giá và chiến lược tiếp thi để giúp thương mại điện tử nâng cao khả năng cạnh tranh trong thị trường.

**1.4 Đối tượng và phạm vi**

***1.4.1 Đối tượng***

Đối tượng nghiên cứu của đề tài là các sản phẩm được bán trên sàn thương mại điện tử thương mại điện tử. Các sản phẩm này thuộc nhiều danh mục khác nhau như điện tử, gia dụng, thiết bị số và phụ kiện,... Các sản phẩm này có sự đa dạng về loại, giá cả và lượng tiêu thụ. Bằng cách sử dụng mã nguồn mở Scrapy để thu nhập và phân tích dữ liệu từ các sản phẩm này, đề tài nhằm hiểu rõ hơn về xu hướng giá và các yếu tố khác ảnh hưởng đến hiệu suất kinh doanh trên sàn thương mại điện tử. Từ đó, nghiên cứu sẽ cung cấp một cái nhìn tổng quan giúp các doanh nghiệp có thể đưa ra các đề xuất nhằm cải thiện chiến lược kinh doanh.

***1.4.2 Phạm vi***

Đề tài tập trung vào việc thu thập và phân tích dữ liệu sản phẩm từ sàn thương mại điện tử thương mại điện tử bằng cách sử dụng Scrapy để tự động thu thập thông tin từ các trang sản phẩm như tên, hình ảnh, giá cả. Dựa trên kết quả thu thập và phân tích dữ liệu, đề tài sẽ đưa ra đề xuất các chiến lược kinh doanh phù hợp. Phạm vi nghiên cứu sẽ tập trung vào các sản phẩm thuộc một số danh mục tiêu biểu và trong một khoảng thời gian giới hạn nhằm đảm bảo dữ liệu được thu thập đầy đủ và chính xác.

**1.5 Phương pháp nghiên cứu**

***1.5.1 Phương pháp nghiên cứu sơ bộ***

Trước khi tiến hành thu thập dữ liệu, chúng tôi đã thực hiện nghiên cứu sơ bộ để hiểu rõ hơn về Scrapy, bao gồm cách thức hoạt động của nó và những lợi ích mà framework này mang lại trong việc thu thập dữ liệu từ các trang web. Thông qua việc nghiên cứu sơ bộ, chúng tôi sẽ xác định các vấn đề cụ thể cần giải quyết và đề xuất các phương pháp nghiên cứu phù hợp.

***1.5.2 Phương pháp nghiên cứu tài liệu***

Chúng tôi sẽ thực hiện nghiên cứu tài liệu để tìm hiểu về các nghiên cứu liên quan đến việc thu thập dữ liệu từ sàn thương mại điện tử. Qua việc đánh giá các nghiên cứu trước đây, chúng tôi sẽ rút ra kinh nghiệm về cách thu thập, xử lý và quản lý dữ liệu. Đồng thời, nghiên cứu các phương pháp phân tích dữ liệu liên quan để xác định cách tiếp cận phù hợp và áp dụng chúng vào đề tài.

***1.5.3 Phương pháp nghiên cứu thống kê***

Sau khi hoàn thành việc thu thập dữ liệu từ thương mại điện tử, chúng tôi sẽ tiến hành phân tích và đánh giá dữ liệu bằng cách sử dụng các phương pháp thống kê. Các phương pháp thống kê bao gồm việc phân tích đơn biến để mô tả các đặc tính của dữ liệu như giá sản phẩm, danh mục hoặc phân tích tương quan giữa các yếu tố khác nhau. Các phép tính này sẽ giúp tôi đưa ra nhận định về xu hướng thị trường và sự biến động của giá cả trong các danh mục sản phẩm khác nhau.

***1.5.4 Phương pháp thực nghiệm***

Chúng tôi sẽ sử dụng Scrapy để tiến hành thực nghiệm thu thập dữ liệu từ sàn thương mại điện tử. Các bước thực nghiệm bao gồm việc thu thập thông tin sản phẩm từ các danh mục cụ thể và đồng thời đảm bảo quá trình thu thập diễn ra một cách liên tục và ổn định. Chúng tôi cũng sẽ tiến hành đánh giá kết quả thu thập, so sánh dữ liệu thu được với dữ liệu thực tế và phân tích hiệu quả của quá trình này.

***1.5.5 Phương pháp đánh giá***

Chúng tôi sẽ thực hiện đánh giá hiệu quả của việc sử dụng Scrapy bằng cách so sánh độ chính xác của dữ liệu thu thập được với dữ liệu thực tế trên trang thương mại điện tử. Các chỉ số về thời gian thu thập, tần suất truy cập, và chất lượng dữ liệu sẽ được phân tích để đề xuất các hướng cải tiến.

**1.6 Những đóng góp nghiên cứu của đề tài**

***1.6.1 Trong lĩnh vực học thuật***

Đề tài đóng góp một phần đáng kể trong việc mở rộng hiểu biết về việc thu thập dữ liệu tự động trên các sàn thương mại điện tử bằng công cụ Scrapy. Bằng cách áp dụng Scrapy vào việc thu thập và xử lý dữ liệu từ thương mại điện tử, nghiên cứu này cung cấp một ví dụ thực tiễn về khả năng của các công cụ mã nguồn mở trong lĩnh vực thu thập dữ liệu web. Kết quả của nghiên cứu cũng đóng góp vào việc phát triển một cơ sở dữ liệu phong phú về các sản phẩm điện tử, từ đó tạo nền tảng cho các nghiên cứu khác về thương mại điện tử, phân tích hành vi tiêu dùng và ứng dụng học máy. Đồng thời, nghiên cứu này cũng bổ sung thêm tài liệu về phương pháp và quy trình thu thập dữ liệu hiệu quả, phục vụ cho các nghiên cứu trong tương lai.

***1.6.2 Trong thực tiễn kinh doanh***

Đề tài không chỉ dừng lại ở nghiên cứu lý thuyết mà còn mang tính thực tiễn cao. Doanh nghiệp có thể sử dụng dữ liệu này để phân tích thị trường, xác định xu hướng tiêu dùng và từ đó phát triển các chiến lược kinh doanh hiệu quả hơn. Đặc biệt, thông tin về giá giúp các công ty tối ưu hóa quy trình định giá, cải thiện chiến lược tiếp thị và tăng khả năng cạnh tranh. Những kết quả này có thể dẫn đến tăng doanh thu, cải thiện hiệu quả kinh doanh và vị thế vững chắc hơn trên thị trường thương mại điện tử.

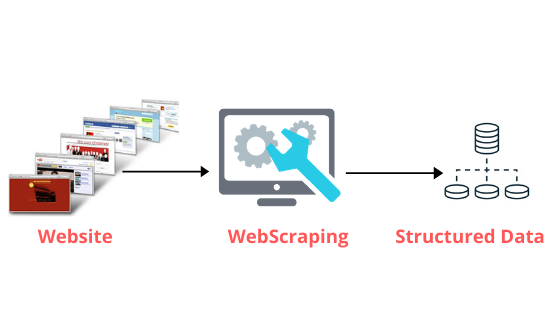
**Chương 2: CƠ SỞ LÝ THUYẾT**

**2.1 Web scraping**

***2.1.1 Giới thiệu về Web Scraping***

Web scraping là một kỹ thuật dùng để tự động thu thập dữ liệu từ các website. Chúng có nhiệm vụ trích xuất dữ liệu trực tiếp từ World Wide Web thông qua giao thức HTTP hoặc trình duyệt web. Một số trường hợp sử dụng chính của việc thu thập dữ liệu web bao gồm theo dõi giá, thông tin giá cả , theo dõi tin tức, tạo khách hàng tiềm năng và nghiên cứu thị trường cùng nhiều mục đích khác. Cách thức hoạt động của nó như sau:

* Xác định trang web mục tiêu.
* Thu thập các URL của trang web bạn muốn lấy dữ liệu.
* Gửi yêu cầu đến các URL để nhận HTML của trang web.
* Sử dụng bộ phân tích để tìm kiếm dữ liệu trong HTML.
* Lưu dữ liệu trích xuất vào các định dạng như JSON, CSV hoặc các định dạng khác có cấu trúc.



Hình 2.1.1.1: Hình ảnh cách thức hoạt động của Web Scraping.[1]

***2.1.2 Ưu điểm và hạn chế***

Ưu điểm:

* Web scraping cho phép tự động hóa việc thu thập thông tin từ nhiều trang web khác nhau, giúp tiết kiệm thời gian và công sức so với việc thu thập thủ công.
* Người dùng có thể thu thập và xử lý dữ liệu từ hàng ngàn trang web trong một khoảng thời gian ngắn.
* Người dùng có thể thiết lập các công cụ Web Scraping để tự động cập nhật dữ liệu theo thời gian thực, giúp thông tin luôn chính xác và mới nhất.
* Dữ liệu thu thập được có thể được sử dụng để phân tích xu hướng, đánh giá cạnh tranh và đưa ra quyết định kinh doanh chính xác hơn .

Hạn chế:

* Việc thu thập dữ liệu từ các trang web có thể vi phạm điều khoản dịch vụ của trang web đó, dẫn đến các vấn đề pháp lý.
* Nếu gửi quá nhiều yêu cầu trong một khoảng thời gian ngắn, quá trình scraping có thể gây quá tải cho máy chủ của trang web, dẫn đến việc bị chặn hoặc làm giảm hiệu suất của trang web.
* Các trang web sử dụng JavaScript hoặc các công nghệ tải động có thể gây khó khăn cho việc trích xuất dữ liệu.
* Để thực hiện Web Scraping hiệu quả, người dùng cần có kiến thức về lập trình, đặc biệt là trong việc sử dụng các thư viện và công cụ như BeautifulSoup, Scrapy hay Selenium.

***2.1.3 Ứng dụng của Web Scraping***

Web Scraping có ứng dụng đa dạng trong nhiều lĩnh vực khác nhau, dưới đây là một số ứng dụng phổ biến:

* Thu thập và phân tích thông tin về giá cả và sản phẩm giúp doanh nghiệp đưa ra quyết định thông minh và tối ưu hóa chiến lược kinh doanh dựa trên dữ liệu chính xác.
* Tạo ra nội dung chất lượng bằng cách thu thập thông tin từ nhiều nguồn khác nhau trên Internet, từ đó thu hút sự chú ý của khách hàng và nâng cao giá trị thương hiệu.
* Theo dõi hoạt động của đối thủ cạnh tranh và các công ty trong ngành giúp doanh nghiệp hiểu rõ chiến lược kinh doanh, từ đó cải thiện sức mạnh cạnh tranh.
* Các nhà nghiên cứu có thể sử dụng web scraping để thu thập dữ liệu từ các bài báo, diễn đàn hoặc trang web học thuật.

**2.2 Ngôn ngữ Python**

***2.2.1 Giới thiệu về Python***

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học. Quá trình này có các phương pháp phổ biến như sau:

* In ra màn hình: Sử dụng hàm print() để xuất dữ liệu trực tiếp ra giao diện dòng lệnh.
* Ghi vào tệp: Sử dụng các thao tác tệp (như open(), write(), read()) để ghi kết quả vào các tệp văn bản, CSV, JSON, v.v.
* Chuyển đổi thành các định dạng khác: Sử dụng thư viện như pandas để trích xuất dữ liệu dưới dạng bảng và xuất ra các định dạng như Excel, CSV.
* Trích xuất từ các Python hoặc web: Sử dụng các thư viện như requests để lấy dữ liệu từ Python hoặc web.
* Lưu trữ dữ liệu: Dùng các cơ sở dữ liệu như SQLite, MySQL để lưu và truy xuất dữ liệu trực từ Python.

***2.2.2 Ưu điểm và hạn chế***

*Ưu điểm:*

* Dễ học và sử dụng: Python có cú pháp rõ ràng và dễ hiểu, giúp người học tập trung vào giải quyết vấn đề thay vì phải lo lắng về các chi tiết phức tạp của ngôn ngữ.
* Cộng đồng lớn và hỗ trợ mạnh mẽ: Python có một cộng đồng đông đảo và năng động, cung cấp hàng ngàn thư viện và công cụ hữu ích.
* Đa năng và linh hoạt: Python được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồ phát triển trang web, khoa học dữ liệu, trí tuệ nhân tạo, …
* Hiệu suất và năng suất: Dù đơn giản nhưng vẫn cung cấp hiệu suất đáng kể, công cụ mạnh mẽ giúp giảm thiểu thời gian phát triển và tăng hiệu suất công việc.
* Tích hợp tốt và mở rộng: cho phép tích hợp dễ dàng với các ngôn ngữ khác và hỗ trợ viết các module mở rộng để mở rộng khả năng của nó.

*Hạn Chế:*

* Tốc độ thực ti: So với một số ngôn ngữ khác như C++ hay Java, Python có tốc độ thực thi chậm hơn do quá trình biên dịch tương đối nhanh chóng.
* Gói thư viện không thống nhất: Do sự phát triển nhanh chóng của cộng đồng, có thể một số thư viện không thống nhất hoặc không được duy trì tốt.
* Bảo mật; Trích xuất dữ liệu qua Python, đặc biệt là từ các nguồn nhạy cảm (API hoặc cơ sở dữ liệu), có thể gặp rủi ro nếu không được bảo mật đugs cách.
* Khả năng xử lý: Python có GIL (Global Interpreter Lock), điều này hạn chế khả năng xử lý và ảnh hưởng đến hiệu suất và xử lý đồng thời lượng dữ liệu lớn.

**2.2.3 Ứng dụng của Python**

Ngôn ngữ Python được ứng dụng đa dạng trong nhiều lĩnh vực khác nhau, bao gồm:

* Python được sử dụng rộng rãi trong phát triển web hay game một cách dễ dàng.
* Python là ngôn ngữ hàng đầu trong khoa học dữ liệu và phân tích dữ liệu nhờ vào các thư viện mạnh mẽ như Pandas, NumPy, Matplotlib,.... Python giúp các nhà khoa học và nhà phân tích xử lý, phân tích và trực quan hóa dữ liệu một cách hiệu quả.
* Python là công cụ tuyệt vời cho việc tự động hóa các tác vụ thông thường hoặc viết các script đơn giản để xử lý các công việc lặp đi lặp lại như quản lý file, gửi email tự động, hay thu thập dữ liệu từ web.

**2.3 Thư viện Scrapy**

***2.3.1 Giới thiệu về Scrapy***

Scrapy là một thư viện Python được tạo ra để quét và xây dựng các chương trình thu thập dữ liệu web. Nó nhanh chóng, đơn giản và có thể điều hướng qua nhiều trang web mà không mất nhiều công sức.



Hình 2.3.1.1: Hình ảnh logo của Scrapy

***2.3.2 Lý do sử dụng***

Scrapy là một framework mạnh mẽ dùng để crawling và thu thập dữ liệu từ các website một cách nhanh chóng và hiệu quả. Với khả năng xử lý song song, nó có thể thu thập dữ liệu từ nhiều trang cùng lúc, giúp tiết kiệm thời gian. Scrapy cũng rất dễ mở rộng, cho phép bạn tùy chỉnh các chức năng như quản lý proxy, cookies, và cài đặt delay để tránh bị chặn khi truy cập trang web.

Ngoài ra, Scrapy hỗ trợ nhiều định dạng lưu trữ như JSON, CSV, XML, và có thể lưu trữ dữ liệu trực tiếp vào các cơ sở dữ liệu như MongoDB hoặc MySQL. Nó cũng tích hợp tốt với các thư viện như Selenium để xử lý trang web có sử dụng JavaScript. Hơn nữa, Scrapy có một cộng đồng lớn và tài liệu phong phú, giúp việc học tập và sử dụng dễ dàng hơn.

Scrapy lý tưởng cho các dự án thu thập dữ liệu lớn, phân tích thị trường, và phát triển ứng dụng dựa trên dữ liệu từ web.

***2.3.3 Ưu điểm và hạn chế***

*Ưu điểm*

* Hiệu quả và Tốc độ: Về hiệu suất và tốc độ trong việc thu thập dữ liệu web, Scrapy được cho là nền tảng tốt nhất hiện nay. Nó xử lý các yêu cầu không đồng bộ, nghĩa là nó có thể thực hiện nhiều yêu cầu cùng một lúc. Điều này làm cho nó rất phù hợp để thu thập dữ liệu web với thông lượng cao.
* Trích xuất dữ liệu có cấu trúc: Với Scrapy, việc thu thập dữ liệu trở thành một nhiệm vụ đơn giản nhờ các tính năng phổ quát được tích hợp trong nền tảng. Các chức năng Selector và item pipeline cho phép kết hợp và định dạng dữ liệu, do đó cho phép tiếp cận việc thu thập dữ liệu theo cách có hệ thống.
* Khả năng mở rộng: Scrapy hỗ trợ các tính năng mở rộng và tùy chỉnh. Chức năng của nó có thể được sửa đổi và mở rộng để có thể xuất hiện với các nhu cầu thu thập dữ liệu khác nhau. Theo cách này, nó trở nên phù hợp với mọi loại tác vụ thu thập dữ liệu web.

*Hạn chế*

* Kết xuất JavaScript: Vì Scrapy là trình thu thập dữ liệu web, tôi có thể nói rằng nó có thể gặp khó khăn với các trang web có nhiều dữ liệu động và sử dụng Javascript để kết xuất nội dung. Tuy nhiên, nó xử lý một số tương tác dựa trên Java nhưng có thể bị hạn chế trong các tình huống mà nội dung tương tác cần hoạt động bên trong trình duyệt.
* Đường cong học tập: Những người tham gia thu thập dữ liệu web lần đầu có thể gặp một số khó khăn khi sử dụng Scrapy. Việc nắm bắt cách thức hoạt động của khuôn khổ cơ bản của nó, đặc biệt là các con nhện và đường ống, có thể là vấn đề đối với những người mới sử dụng khuôn khổ này.
* Tương tác trình duyệt hạn chế: Nhiều người dùng khuôn khổ thu thập dữ liệu web sẽ lưu ý rằng Scrapy không cung cấp nhiều tương tác trình duyệt như Selenium. Do đó, nếu cốt lõi của tác vụ thu thập dữ liệu là nhấp chuột xung quanh các trang web động, Scrapy có thể không phù hợp nhất.

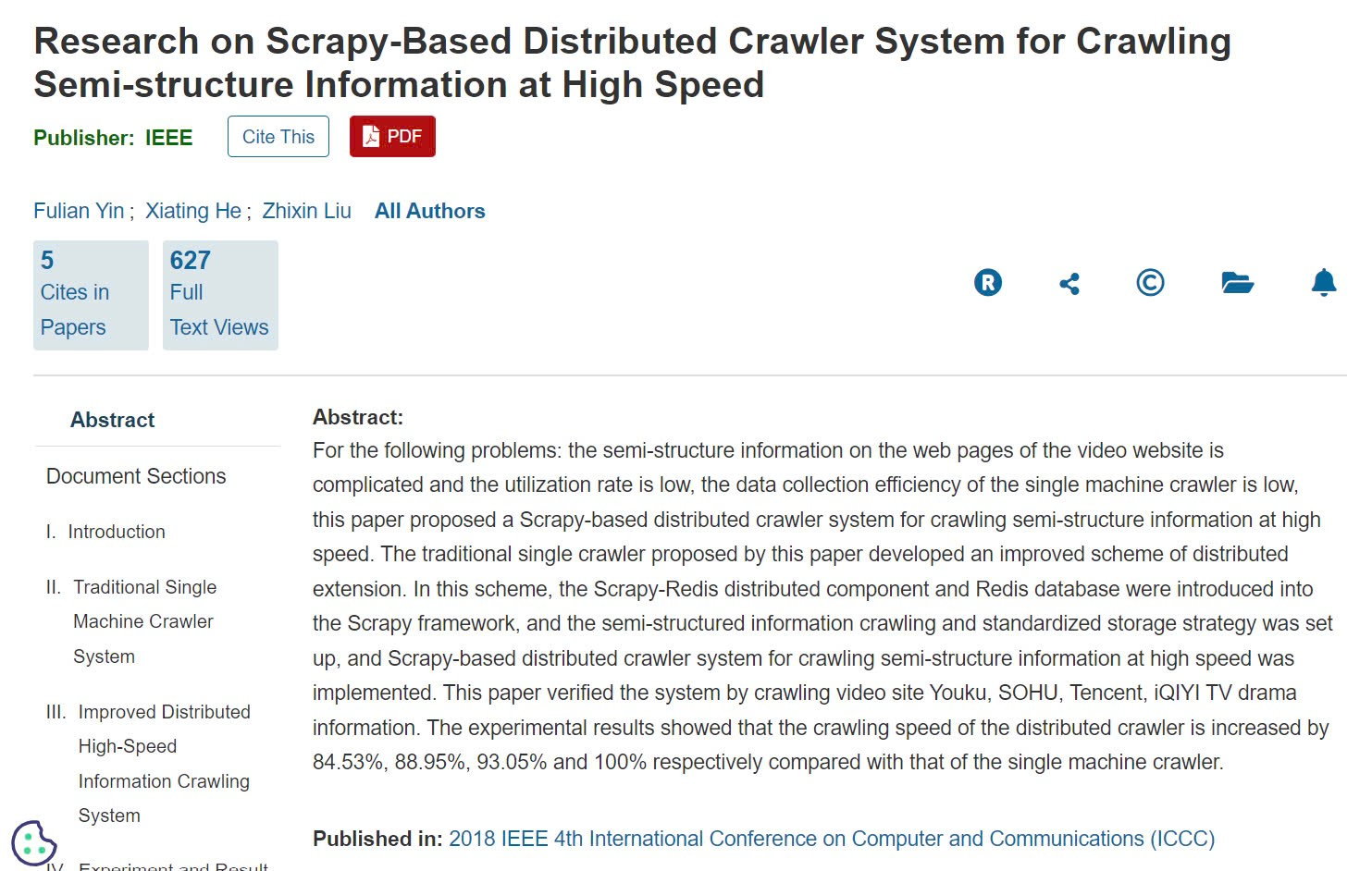
***2.3.4 Ứng dụng của Scrapy***

Một số ứng dụng phổ biến của Scrapy:

* Cho phép người dùng dễ dàng thu thập dữ liệu từ các trang web khác nhau, bao gồm giá sản phẩm, mô tả sản phẩm, thông tin bài viết và nhiều loại dữ liệu khác.
* Với khả năng xử lý HTML và XML mạnh mẽ, Scrapy có thể phân tích cấu trúc của trang web và trích xuất thông tin theo yêu cầu, giúp người dùng hiểu rõ hơn về dữ liệu mà họ đang làm việc.
* Scrapy có thể được sử dụng để theo dõi hoạt động của đối thủ cạnh tranh, thu thập thông tin về giá cả và chiến lược kinh doanh của họ. Điều này giúp doanh nghiệp có cái nhìn tổng quan hơn về thị trường.
* Scrapy có thể tự động thu thập và cập nhật nội dung từ các trang web theo lịch trình định sẵn. Điều này hữu ích cho các trang web cần thường xuyên cập nhật thông tin như tin tức, giá cả sản phẩm, hoặc các bài viết mới.

***2.3.5 Ứng dụng đã có của Scrapy***

Trước đây, Scrapy chỉ là một công cụ thu thập dữ liệu đơn giản và ít được biết đến. Tuy nhiên, nhờ sự cải tiến về mặt tính năng và hiệu suất, nó đã trở thành một framework mạnh mẽ và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ nghiên cứu thị trường đến phân tích những dữ liệu lớn. Với Scrapy, người dùng có thể dễ dàng thu thập dữ liệu từ các trang web một cách tự động, điều này đặc biệt hữu ích đối với các sàn thương mại điện tử, tiếp thị kỹ thuật số,... . Không chỉ vậy, Scrapy còn cho phép tích hợp với các công cụ phân tích và xử lý dữ liệu, giúp tối ưu hóa quy trình từ việc thu thập, làm sạch cho đến phân tích thông tin. Dưới đây là bài báo về việc sử dụng Scrapy để thu thập dữ liệu và vẫn còn nhiều bài nghiên cứu và bài báo đã được công bố, cho thấy được sự hiệu quả của Scrapy trong việc thu thập và xử lý dữ liệu.



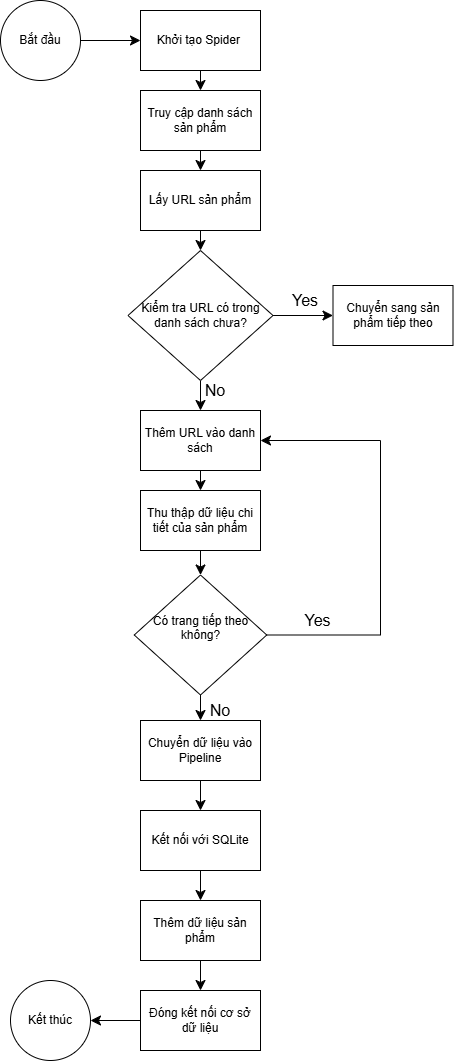
Hình 2.3.4.1: Bài nghiên cứu về hệ thống thu thập thông tin phân tán dựa trên Scrapy để thu thập thông tin bán cấu trúc với tốc độ cao.[2]



Hình 2.3.4.1: Bài báo phân tích đặc điểm hành vi người dùng và thu thập dữ liệu dựa trên Scrapy trên Taobao. [3]

**CHƯƠNG 3: PHƯƠNG PHÁP THỰC NGHIỆM**

* 1. **Sơ đồ quá trình thu thập và xử lý dữ liệu**

****

* 1. **Phương pháp thu thập dữ liệu**

Trong quá trình thu thập dữ liệu sản phẩm trên sàn thương mại điện tử Grandcru.vn, chúng tôi đã áp dụng công cụ mã nguồn mở Scrapy để tự động hóa và tối ưu hóa quy trình này. Sử dụng Scrapy, chúng tôi có thể gửi yêu cầu tới API của trang web, thu thập và trích xuất dữ liệu sản phẩm một cách hiệu quả và có cấu trúc. Các kết quả thu thập được lưu trữ dưới dạng tệp JSON, giúp phân tích và xử lý dữ liệu thêm dễ dàng hơn. Quá trình này được thực hiện theo các bước sau:

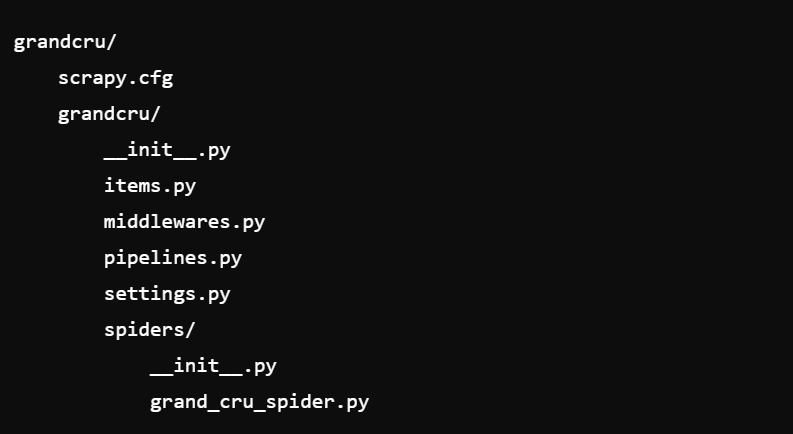
***3.2.1 Cài đặt Scrapy và tạo cấu trúc Scrapy***

Cài đặt Scrapy:

* Để cài đặt Scrapy, người dùng có thể sử dụng câu lệnh *pip install scrapy* trong Terminal, đây là lệnh cài đặt tiêu chuẩn cho các thư viện của Python. Lệnh này sẽ tự động tải và cài đặt tất cả các thành phần cần thiết mà Scrapy cần để hoạt động. Để kiểm tra cài đặt, người dùng có thể chạy lệnh *scrapy*, nếu danh sách các lệnh phụ của Scrapy xuất hiện, điều đó có nghĩa là cài đặt đã thành công và Scarpy đã sẵn sàng để bắt đầu làm việc.



Hình 3.1 Hình ảnh câu lệnh để cài đặt Scrapy

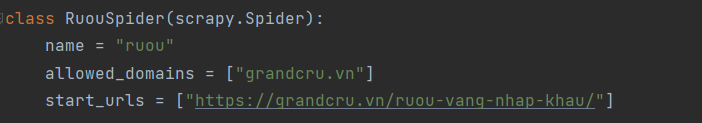


Hình 3.2: Hình ảnh thông tin cấu trúc dự án

Tạo cấu trúc Scrapy:

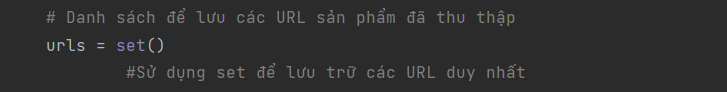
* Khi Scrapy được cài đặt thành công, để khởi tạo một dự án mới ta sử dụng lệnh *scrapy startproject <tên\_dự\_án>.* Lệnh này sẽ tự động thiết lập một cấu trúc thư mục và tệp tin cần thiết, bao gồm thư mục **spiders** là nơi lưu trữ các tệp spider chịu trách nhiệm thu thập dữ liệu từ trang web; file **items.py** để định nghĩa các trường dữ liệu cần lấy; **setting.py** để quản lý các thiết lập quan trọng như giới hạn tốc độ thu thập và các cấu hình liên quan khác và **pipelines.py** là nơi xử lý dữ liệu sau khi thu thập. Với cấu trúc được chuẩn hóa, Scrapy cho phép phát triển các dự án thu thập dữ liệu một cách linh hoạt, hỗ trợ dễ dàng cho việc bảo trì và mở rộng trong tương lai.
  + 1. ***Truy xuất thông tin cửa hàng***
* Sử dụng Scrapy để truy xuất thông tin về các cửa hàng trên Grandcru.vn.
* Bằng cách lấy url (là địa chỉ duy nhất để xác định một tài nguyên trên Internet) của một sản phẩm ngẫu nhiên trên nền tảng Grandcru.vn, chúng tôi xác định đường dẫn chung của sản phẩm và trích xuất file JSON để thu thập danh sách các ID sản phẩm từ đường dẫn đã xác định.
* Tiếp theo, thu thập thông tin từ Scrapy của cửa hàng, bao gồm: 'name', 'price', 'img', ' url ' , ' wine\_type ', ' grape\_variety ', 'alcohol\_ concentration ', ' volume ', ' vintage ' của mỗi sản phẩm trong cửa hàng.
  + 1. ***Truy xuất thông tin và khởi tạo danh sách của trang web***

Trước hết chúng tôi xây dựng một lớp tên là RuouSpider, kế thừa từ lớp Spider của Scrapy. Đặt tên cho spider này là “ruou”, dùng để nhận diện khi chạy lệnh để khởi động spider. Giới hạn phạm vi thu thập chỉ ở trong miền (“grandcru.vn”), tiếp đến chúng tôi khai báo một danh sách URL khởi đầu của trang web (“https://grandcru.vn/ruou-vang-nhap-khau/”) mà spider sẽ bắt đầu thu thập dữ liệu. Với mỗi sản phẩm, spider sẽ gửi yêu cầu đến các URL chi tiết này để lấy các thông tin cụ thể như tên rượu, giá, mô tả, và các thuộc tính khác của từng loại rượu vang.

******

Hình 3.3: Đoạn code thiết lập một trình thu thập dữ liệu từ trang “grandcru.vn”.

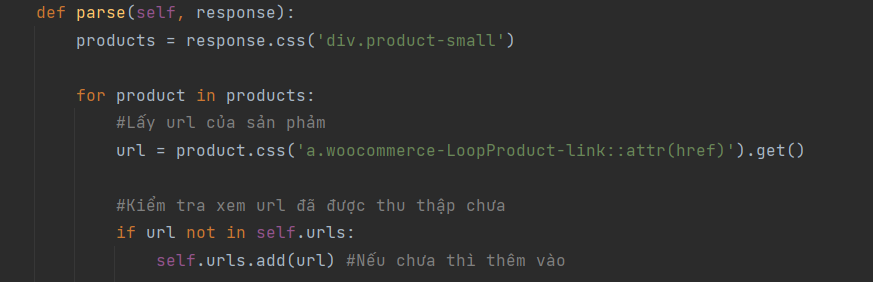
Sau khi thiết lập xong các cấu hình cho spider, chúng tôi khởi tạo một **set** có tên ‘urls’ để lưu trữ các URL sản phẩm đã thu thập. Việc dùng **set** không chỉ giúp quản lý dữ liệu một cách hiệu quả mà còn đảm bảo rằng mỗi URL chỉ được thêm vào một lần, từ đó ngăn chặn được sự trùng lặp giúp giảm thiểu việc gửi yêu cầu không cần thiết đến cùng một trang và cải thiện hiệu suất thu thập.



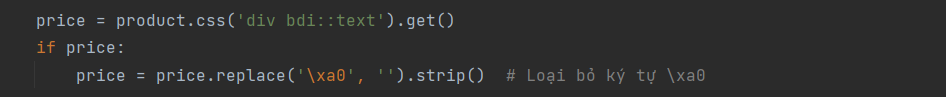
Hình 3.4: Code dùng để lưu trữ các URL sản phẩm đã thu thập

* + 1. ***Thu thập dữ liệu từ trang web***

Chúng tôi khởi tạo phương thức parse(), nơi spider tiếp nhận phẩn hồi từ trang web và xử lý để thu thập danh sách các sản phẩm một cách có hệ thống. Phương thức này không chỉ tìm và lấy các liên kết đến chi tiết sản phẩm, mà còn truy cập vào các trang tiếp theo (nếu có) nhằm thu thập đầy đủ tất cả các liên kết sản phẩm.

****** Hình 3.5 Đoạn code dùng để thu thập danh sách sản phẩm và trích xuất các liên kết

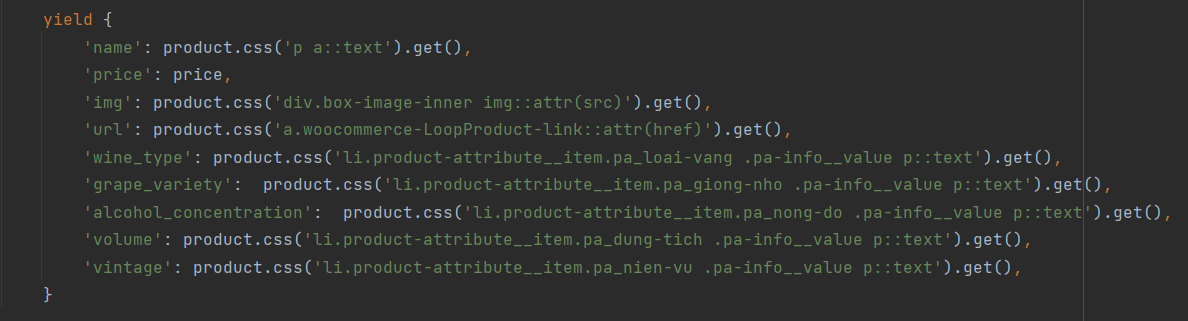
* Việc đầu tiên là lấy danh sách sản phẩm bằng cách sử dụng câu lệnh “*products = response.css('div.product-small')”* trong đó CSS selector “*div.product-small”* được sử dụng để tìm tất cả các phẩn tử sản phẩm trên trang, sau đó được lưu vào biến **products**.
* Tiếp đến duyệt qua từng sản phẩm, ta sử dụng vòng lặp *“for product in products”,* mã lệch sẽ lặp qua từng sản phẩm trong danh sách **products**. Để lấy URL từ thẻ liên kết của từng sản phẩm, chúng tôi sử dụng lệnh *“url = product.css('a.woocommerce-LoopProduct-link::attr(href)').get()”.* Cụ thể, nó sử dụng selector để tìm thuộc tính href của thẻ <a> có lớp “*woocommerce-LoopProduct-link”*, từ đó trích xuất liên kết đến trang chi tiết của sản phẩm. Phương thức *“.get()”* được sử dụng để lấy giá trị đầu tiên từ danh sách các kết quả trả về từ phương thức .css(), nếu không tìm thấy thẻ nào, nó sẽ trả về giá trị None.
* Cuối cùng, kiểm tra xem URL đã được thu thập hay chưa bằng cách sử dụng lệnh *“if url not in self.url”*. Nếu URL chưa có trong ‘*self.urls’*, nó sẽ được thêm vào thông qua *“self.urls.add(url)”*. Việc này giúp đảm bảo rằng mỗi URL sản phẩm chỉ được lưu lại một lần, ngăn chặn sự trùng lặp và tối ưu hóa quá trình thu thập dữ liệu.



Hình 3.6: Đoạn code lấy giá sản phẩm và loại bỏ ký tự dư thừa

Để lấy giá sản phẩm, chúng tôi sử dụng câu lệnh *“product.css('div bdi::text').get()”*, trong đó **product** đại diện cho từng sản phẩm trong danh sách. Câu lệnh này truy cập vào thẻ <bdi> nằm bên trong thẻ <div> trong phần tử sản phẩm, nơi chứa thông tin về giá. Nếu giá tồn tại, đoạn mã sẽ thực hiện xử lý để loại bỏ ký tự “\xa0” bằng phương thức *“replace()”* và thay thế bằng chuỗi rỗng (‘ ’) . Phương thức *“.strip()”* được sử dụng để loại bỏ các ký tự khoảng trắng ở đầu và cuối chuỗi, giúp đảm bảo rằng giá trị cuối cùng chỉ chứa nội dung chính xác mà không có khoảng trắng thừa.

Thông qua phương thức “yield”*,* các dữ liệu của sản phẩm sẽ được lưu trữ dưới dạng từ điển và gửi đi để xử lý hoặc lưu trữ tiếp trong pipeline.

****

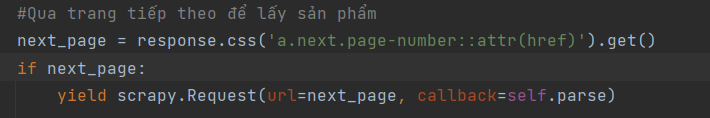
Hình 3.7: Đoạn code thu thập thông tin chi tiết về các sản phẩm

Đoạn mã này thực hiện theo các bước sau:

* Câu lệnh *“'name': product.css('p a::text').get()”*:trong đó giá trị cho khóa **name** được lấy từ thẻ <a> nằm trong thẻ <p>. Đoạn mã sử dụng CSS selector “p a::text” để tìm nội dung văn bản của thẻ <a> và lấy tên sản phẩm.
* Câu lệnh *“'price': price”*: giá trị **price** đã được thu thập và xử lý trước đó, nên đoạn mã chỉ cần đưa vào trường price để lưu lại giá của sản phẩm.
* Câu lệnh *“'img': product.css('div.box-image-inner img::attr(src)').get()”* : để lấy URL của ảnh sản phẩm. “'div.box-image-inner” dùng để xác định vị trí của ảnh sản phẩm nằm trong thẻ <div> có lớp “box-image-inner”. “img::attr(src)” để lấy giá trị thuộc tính “src” từ thẻ <img>, trả về đường dẫn của ảnh sản phẩm.
* Câu lệnh *“'url': product.css('a.woocommerce-LoopProduct-link::attr(href)').get()”*: tìm thẻ <a> có lớp “woocommerce-LoopProduct-link” và trích xuất thuộc tính “href”, cung cấp liên kết trực tiếp đến trang chi tiết của sản phẩm.
* Câu lệnh *“'wine\_type': product.css('li.product-attribute\_\_item.pa\_loai-vang .pa-info\_\_value p::text').get()”*: truy xuất thông tin về loại rượu của sản phẩm. Sử dụng CSS selector “li.product-attribute\_\_item.pa\_loai-vang .pa-info\_\_value p::text” để tìm thẻ <p> chứa văn bản trong danh sách thuộc tính “product-attribute\_\_item”, từ đó lấy loại rượu.
* Câu lệnh *“'grape\_variety': product.css('li.product-attribute\_\_item.pa\_giong-nho .pa-info\_\_value p::text').get()”*: lấy thông tin về giống nho của sản phẩm. Thông qua selector “li.product-attribute\_\_item.pa\_giong-nho .pa-info\_\_value p::text”, đoạn mã sẽ trỏ đến thẻ <p> chứa văn bản mô tả giống nho và trích xuất nội dung.
* Câu lệnh *“'alcohol\_concentration': product.css('li.product-attribute\_\_item.pa\_nong-do .pa-info\_\_value p::text').get()”*: lấy nồng độ cồn của rượu. Selector “li.product-attribute\_\_item.pa\_nong-do .pa-info\_\_value p::text” được sử dụng để tìm thẻ <p> chứa văn bản, cho phép trích xuất nồng độ cồn.
* Câu lệnh *“'volume': product.css('li.product-attribute\_\_item.pa\_dung-tich .pa-info\_\_value p::text').get()”*: truy xuất thông tin về dung tích của sản phẩm. Selector “li.product-attribute\_\_item.pa\_dung-tich .pa-info\_\_value p::text” giúp xác định và lấy giá trị dung tích từ thẻ <p>, cung cấp dung tích chuẩn.
* Câu lệnh *“'vintage': product.css('li.product-attribute\_\_item.pa\_nien-vu .pa-info\_\_value p::text').get()”*: lấy thông tin về niên vụ của sản phẩm. Sử dụng CSS selector “li.product-attribute\_\_item.pa\_nien-vu .pa-info\_\_value p::text” để trích xuất văn bản từ thẻ <p>, nhằm cung cấp năm sản xuất hoặc niên vụ của chai rượu.

Tất cả các câu lệnh trên đều sử dụng “.get()” để chỉ lấy kết quả đầu tiên khớp với selector, đảm bảo thu thập thông tin chi tiết một cách chính xác cho từng sản phẩm.

Đoạn mã sau thực hiện chức năng kiểm tra và điều hướng tự động đến trang tiếp theo trong quá trình thu thập dữ liệu, giúp đảm bảo rằng tất cả các trang sản phẩm có sẵn đều được truy xuất để thu thập thông tin một cách đầy đủ.



Hình 3.8: Đoạn code để điều hướng tới trang tiếp theo

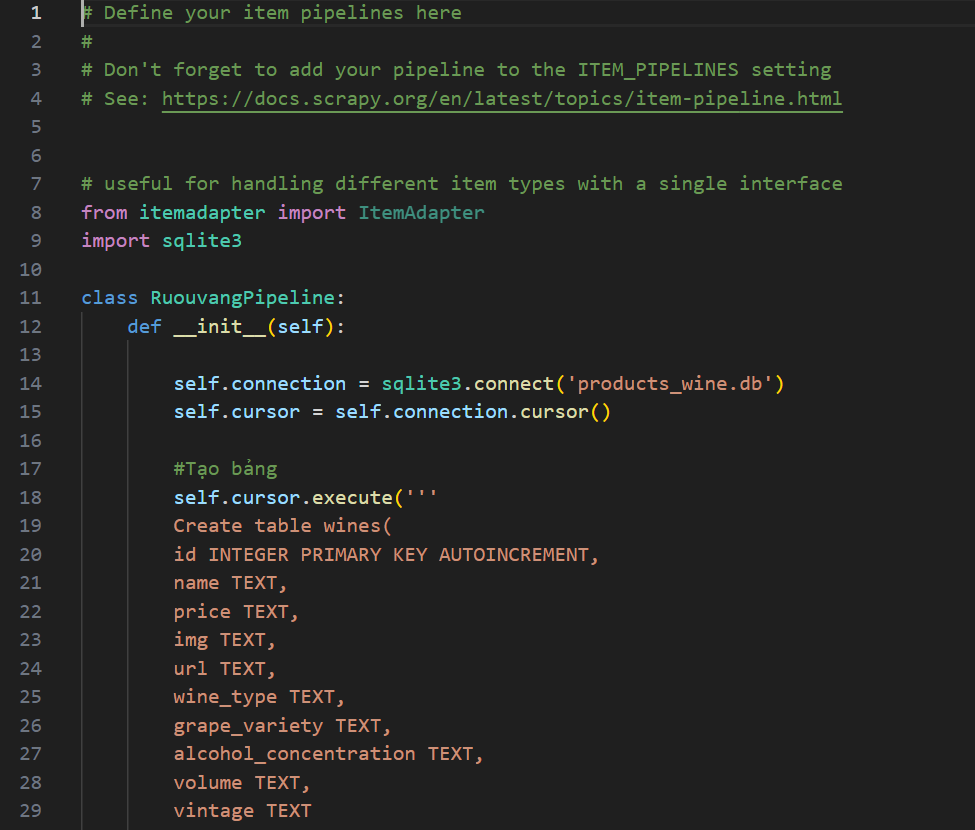
* Dòng lệnh *“next\_page = response.css('a.next.page-number::attr(href)').get()”*: để tìm kiếm liên kết dẫn đến trang tiếp theo trên trang web, nó sẽ tìm thẻ <a> có cả hai lớp “next” và “page-number”, sau đó lấy giá trị của thuộc tính href, nơi chứa liên kết URL của trang kế tiếp.
* Tiếp theo là khối lệnh “if next\_page:”, đây là câu điều kiện nhằm kiểm tra xem biến “next\_page” có chứa URL hợp lệ không. Nếu có giá trị, mã sẽ thực thi các câu lệnh tiếp theo để điều hướng tới trang kế tiếp, ngược lại, nếu không có URL nào hợp lệ, quá trình phân trang sẽ kết thúc.
* Cuối cùng, dòng lệnh “yield scrapy.Request(url=next\_page, callback=self.parse)” sẽ thực hiện yêu cầu (scrapy.Request) đến URL của trang tiếp theo nếu có. Lệnh “callback=self.parse” đảm bảo rằng khi nhận được phản hồi từ trang kế tiếp, hàm “parse” sẽ được gọi lại để tiếp tục xử lý nội dung của trang đó. Điều này giúp tạo ra một quy trình lặp lại, cho phép Scrapy tiếp tục điều hướng qua các trang liên tiếp để thu thập dữ liệu cho đến khi không còn trang nào nữa.
  1. **Mô tả dữ liệu:**

Dữ liệu được thu về các thông tin của cửa hàng, chi tiết của sản phẩm về chất lượng sản phẩm. Dữ liệu được thu thập gồm 9 đặc chưng, bao gồm các thông tin về:

|  |  |  |
| --- | --- | --- |
| **Tên Biến** | **Mô Tả** | **Kiểu dữ liệu** |
| name | Tên sản phẩm | String |
| price | Giá sản phẩm | Float hoặc String |
| img | Hình ảnh sản phẩm | String |
| url | Link của sản phẩm | String |
| wine\_type | Loại vang | String |
| grape\_variety | Giống nho | String |
| alcohol\_concentration | Nồng độ | Float |
| volume | Dung tích | Float |
| vintage | Niên vụ sản xuất | Integer |

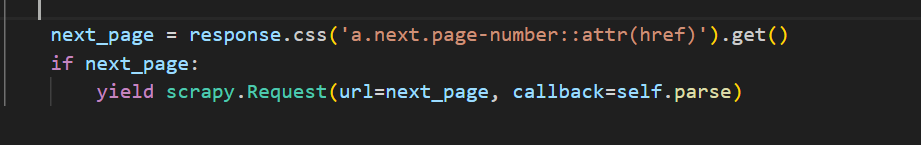
Bảng 3.2.1: Bảng mô tả các biến và kiểu dữ liệu

* 1. **Lưu và trích xuất dữ liệu**



Hình 3.7 Dữ liệu thu thập được từ spider của bạn sẽ được lưu vào tệp JSON và được trích xuất ra SQLite.

* + 1. ***Xử lý phân trang***

******

Hình 3.8 Phân ra sang tiếp theo để lấy sản phẩm

* **response.css(...)**: Sử dụng phương thức “**css()**” để tìm các phần tử HTML chứa liên kết đến trang tiếp theo.
* **'a.next.page-number'**: Đây là selector CSS để chọn thẻ <a> có class **next page-number**. Đây thường là thẻ **<a>** mà các trang phân trang (pagination) sử dụng để chuyển sang trang kế tiếp.