**TRƯỜNG ĐẠI HỌC KINH TẾ**

**ĐẠI HỌC ĐÀ NẴNG**

**--🙞\*🕮\*🙜--**



**BÁO CÁO GIỮA KÌ**

**PHÂN TÍCH DỮ LIỆU BẰNG PYTHON**

**ĐỀ TÀI: PHÁT HIỆN LUẬT KẾT HỢP TỪ DỮ LIỆU BÁN HÀNG ONLINE**

**BẰNG THUẬT TOÁN APRIORI**

GVHD : TS.Lê Diên Tuấn

Nhóm : 8

Thành viên : Hoàng Nghĩa Đức (0986898217) 100%

Trần Thị Hằng 100%

H Ngač Niê 100%

Nguyễn Đức Thắng 100%

Nguyễn Hoàng Khánh Huyền 100%

*Đà Nẵng, 2024*

**Mục Lục**

[I. Tổng quan về đề tài 2](#_Toc163316685)

[II. Cơ sở lý thuyết 3](#_Toc163316686)

[1. Luật kết hợp là gì? 3](#_Toc163316687)

[2. Các thuật toán trong luật kết hợp. 4](#_Toc163316688)

[III. Kết quả thực nghiệm 7](#_Toc163316689)

[1. Frame work 7](#_Toc163316690)

[2. Giải thích dữ liệu 8](#_Toc163316691)

[3. Tiền xử lý dữ liệu 9](#_Toc163316692)

[4. Phân tích khám phá 13](#_Toc163316693)

[5. Luật kết hợp 16](#_Toc163316694)

[6. Chiến lược 19](#_Toc163316695)

[IV. Tài liệu tham khảo 20](#_Toc163316696)

[V. Đường dẫn video 20](#_Toc163316697)

1. **Tổng quan về đề tài**

Nhu cầu dự đoán tương lai, tối ưu hóa quyết định và khai phá tri thức mới thúc đẩy sự ra đời của luật kết hợp - một công cụ mạnh mẽ trong lĩnh vực khai phá dữ liệu. Luật kết hợp như một sợi chỉ dẫn dắt ta đi qua mê cung dữ liệu, soi sáng những mối liên hệ tưởng chừng như vô hình giữa các thuộc tính tưởng chừng như rời rạc. Nó giúp ta nhận ra những quy luật ẩn giấu, những xu hướng tiềm ẩn và đưa ra những dự đoán chính xác về tương lai.

Luật kết hợp tồn tại trong học phần phân tích dữ liệu bằng python vì những lý do sau:

* Khám phá mối liên hệ ẩn trong dữ liệu
* Tăng khả năng dự đoán
* Hỗ trợ việc ra quyết định
* Có nhiều thư viện Python hỗ trợ việc thực hiện luật kết hợp, ví dụ như apriori, fp-growth, eclat.
* Khả năng trực quan hóa dữ liệu

Sự ra đời của luật kết hợp đánh dấu một bước ngoặt trong lĩnh vực khoa học máy tính, mở ra cánh cửa cho vô số ứng dụng thực tiễn trong nhiều lĩnh vực khác nhau, từ kinh doanh, marketing, y tế cho đến khoa học.

Hãy cùng khám phá hành trình đầy thú vị của luật kết hợp, từ nguồn gốc ra đời, cách thức hoạt động cho đến những ứng dụng rộng rãi trong đời sống, để hiểu rõ hơn về sức mạnh tiềm ẩn của công cụ này trong việc khai phá tri thức và kiến tạo tương lai.

1. **Cơ sở lý thuyết**
2. **Luật kết hợp là gì?**

Association rule (luật kết hợp) là mối quan hệ kết hợp giữa các tập thuộc tính trong cơ sở dữ liệu. Là một mệnh đề logic dạng X → Y, trong đó X và Y là các tập hợp thuộc tính (itemset) trong cơ sở dữ liệu. Mệnh đề này thể hiện mối liên hệ thống kê giữa các thuộc tính, với X là điều kiện và Y là kết quả.

Các khái niệm quan trọng liên quan đến luật kết hợp:

Có hai độ đo thường được sử dụng để đánh giá luật kết hợp X → Y:

- Độ phổ biến (support) là một thước đo thống kê được sử dụng trong khai phá luật kết hợp (association rule mining) để đánh giá mức độ phổ biến của một tập hợp các thuộc tính (itemset) trong cơ sở dữ liệu

Công thức :

Sup(X → Y) = P(X ∪ Y) = Sup(XY)

- Độ tin cậy (confidence) là một thước đo thống kê được sử dụng trong khai phá luật kết hợp (association rule mining) để đánh giá mức độ tin cậy của một luật kết hợp.

Công thức :

(X → Y) = P(Y | X) = Sup(XY) / Sup(X)

- Chỉ số Lift được sử dụng để đánh giá mức độ liên quan giữa hai tập hợp thuộc tính (itemset) X và Y trong luật kết hợp X → Y.

Công thức:

Lift(X → Y) = P(X ∩ Y) / (P(X) \* P(Y))

Ý nghĩa:

* Lift > 1: X và Y có liên quan với nhau. Khi X xảy ra, khả năng Y xảy ra cao hơn so với trường hợp ngẫu nhiên.
* Lift = 1: X và Y không liên quan với nhau. Khả năng Y xảy ra sau X không khác gì so với trường hợp ngẫu nhiên.
* Lift < 1: X và Y có liên quan nghịch. Khi X xảy ra, khả năng Y xảy ra thấp hơn so với trường hợp ngẫu nhiên.

- Tập phổ biến là các mẫu (tập các mục [item], chuỗi con, tập con...) xuất hiện thường xuyên trong dữ liệu.

Công thức : Sup(X)=C(X)/|D| Trong đó :

* Sup là độ phổ biên
* C(X) là số lần xuất hiện
* D là số giao dịch

- Minsup (minimum support) và minconf (minimum confidence) là hai ngưỡng quan trọng được sử dụng trong khai phá luật kết hợp để lọc ra các tập hợp itemset và luật kết hợp có ý nghĩa.

Minsup:

* Là ngưỡng độ phổ biến tối thiểu để một tập hợp itemset được giữ lại.
* Các tập hợp itemset có độ phổ biến thấp hơn min sup sẽ bị loại bỏ.
* Việc lựa chọn min sup phù hợp giúp tập trung vào các tập hợp itemset phổ biến và có khả năng mang lại thông tin hữu ích.

Minconf:

* Là ngưỡng độ tin cậy tối thiểu để một luật kết hợp được giữ lại.
* Các luật kết hợp có độ tin cậy thấp hơn min conf sẽ bị loại bỏ.
* Việc lựa chọn min conf phù hợp giúp tập trung vào các luật kết hợp có độ tin cậy cao và có khả năng dự đoán chính xác.

Việc lựa chọn minsup và minconf phụ thuộc vào các yếu tố như:

* Loại dữ liệu
* Mục đích phân tích

1. **Các thuật toán trong luật kết hợp.**

* **Apriori**

Thuật toán Apriori, hay còn được gọi là "ông tổ" của các thuật toán khai thác mô hình tần suất, Apriori dùng để tìm ra các itemset thường xuyên xuất hiện cùng nhau trong một tập dữ liệu lớn các giao dịch, ví dụ như dữ liệu mua sắm của khách hàng….

Apriori sử dụng một cách tiếp cận lặp đi lặp lại được gọi là tìm kiếm theo cấp độ, trong đó các tập mục k được sử dụng để khám phá các tập mục (k + 1). Đầu tiên, tập hợp các tập 1 mục phổ biến được tìm thấy bằng cách quét cơ sở dữ liệu để tích lũy số lượng cho từng mục và thu thập các mục đó thỏa mãn độ hỗ trợ tối thiểu. Tập kết quả được ký hiệu là L1. Tiếp theo, L1 được sử dụng để tìm L2, tập hợp gồm 2 mục phổ biến, được sử dụng để tìm L3, v.v., cho đến khi không thể tìm thấy tập k mục phổ biến nào nữa. Việc tìm ra mỗi Lk yêu cầu phải quét toàn bộ cơ sở dữ liệu một lần.

Ý tưởng chính:

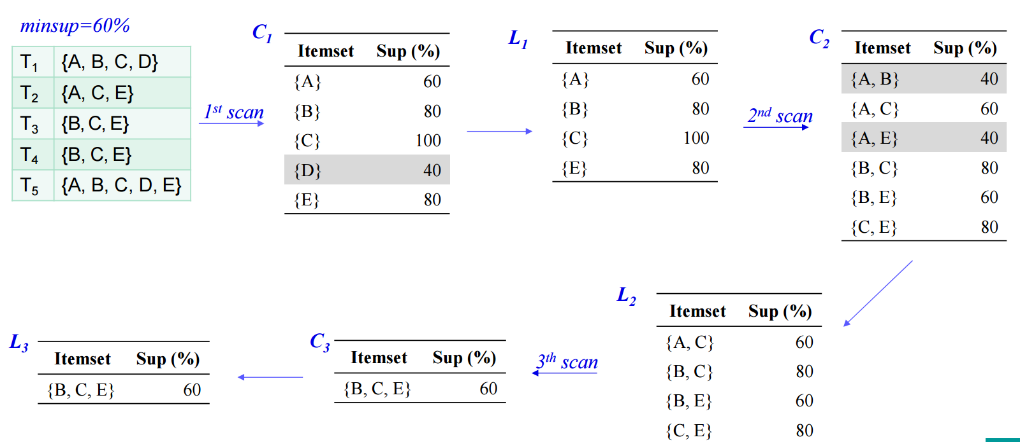
* Tên của thuật toán là Apriori vì nó sử dụng kiến thức đã có từ trước (prior) về các thuộc tính, vật phẩm thường xuyên xuất hiện trong cơ sở dữ liệu.
* Thuật toán tìm tất cả các tập phổ biến (frequent itemset) có độ phổ biến (support) thỏa mãn độ phổ biến tối thiểu (minsup).
* Để cải thiện hiệu quả, thuật toán sử dụng thuộc tính quan trọng gọi là thuộc tính Apriori để giảm phạm vi tìm kiếm, giúp tăng hiệu suất của thuật toán. (Thuộc tính này nói rằng trong một tập dữ liệu lớn, nếu một itemset không phải là phổ biến - tức là không đạt ngưỡng phổ biến tối thiểu, thì tất cả các itemsets lớn hơn chứa itemset đó cũng không phổ biến. VD nếu {A, B} không phổ biến, thì {A, B, C} cũng không phổ biến, và chúng ta không cần kiểm tra nó.)

Phương pháp:

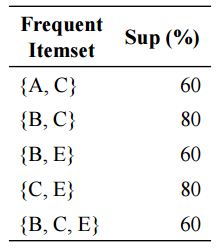
* Tìm tất cả các tập phổ biến 1 hạng mục tính toán tần suất xuất hiện của chúng, gọi là hỗ trợ (support). Mọi item có giá trị hỗ trợ lớn hơn hoặc bằng ngưỡng hỗ trợ tối thiểu (min support) được coi là "phổ biến".
* Tạo các tập ứng viên k+1 hạng mục từ tập phổ biến có k hạng mục: Từ các tập phổ biến với k hạng mục (k-itemsets) đã được tìm thấy ở các bước trước, ta tạo ra các tập ứng viên mới bằng cách kết hợp chúng với nhau để tạo thành các tập ứng viên lớn hơn có k+1 hạng mục. Nguyên tắc Apriori được áp dụng ở đây để loại bỏ bất kỳ ứng viên nào không thể trở thành phổ biến do chứa các subset không phổ biến.
* Kiểm tra độ phổ biến của các ứng viên: Sau khi tạo ra các tập ứng viên k+1-itemsets, cơ sở dữ liệu được quét để tính tần suất xuất hiện của mỗi tập ứng viên. Ứng viên nào đạt đủ ngưỡng hỗ trợ tối thiểu thì được giữ lại và coi là tập phổ biến
* Dừng khi không tạo được tập phổ biến hoặc tập ứng viên: Quá trình này tiếp tục lặp lại cho đến khi không còn tạo ra được thêm tập PB mới nào - nghĩa là không còn tập ứng viên nào đạt ngưỡng hỗ trợ tối thiểu - hoặc khi không thể kết hợp các tập phổ biến k-itemsets hiện tại để tạo ra các tập ứng viên k+1-itemsets hợp lệ. Khi đến điểm dừng, quy trình sẽ kết thúc và ta sẽ có được danh sách cuối cùng của tất cả các tập hạng mục phổ biến.
* Từ frequent item + minconf ta tìm ra các luật kết hợp

Ví dụ: Chọn minsup=60% và minconf=80%, tìm luật kết hợp của bộ dữ liệu sau:

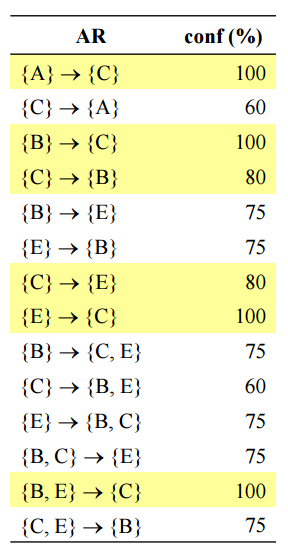
* Lần quét dữ liệu đầu tiên, xác định tất cả các item đơn lẻ trong cơ sở dữ liệu và tính toán tần suất xuất hiện của chúng và kết quả ở bảng C1. Bắt đầu với bảng C1, chọn item có giá trị sup lớn hơn hoặc bằng ngưỡng sup tối thiểu (min support) được coi là "phổ biến" và kết quả ở bảng L1
* Lần quét dữ liệu thứ 2 từ các tập phổ biến với 1 hạng mục (bảng L1) đã được tìm thấy ở các bước trước, ta tạo ra các tập ứng viên mới bằng cách kết hợp chúng với nhau để tạo thành các tập ứng viên lớn hơn có k+1=2 hạng mục và tính toán sup cho từng tập ứng viên
* Sau khi tạo ra các tập ứng viên 2 hạng mục, cơ sở dữ liệu được quét để tính tần suất xuất hiện của mỗi tập ứng viên. Ứng viên nào đạt đủ ngưỡng hỗ trợ tối thiểu thì được giữ lại và coi là tập phổ biến kết quả ở bảng L2
* Tiếp tục các bước tương tự ta tìm thêm được tập phổ biến ở L3

****

* Quá trình này tiếp tục lặp lại cho đến khi không còn tạo ra được thêm tập PB mới nào. Khi đến điểm dừng, quy trình sẽ kết thúc và ta sẽ có được danh sách cuối cùng của tất cả các tập hạng mục phổ biến.



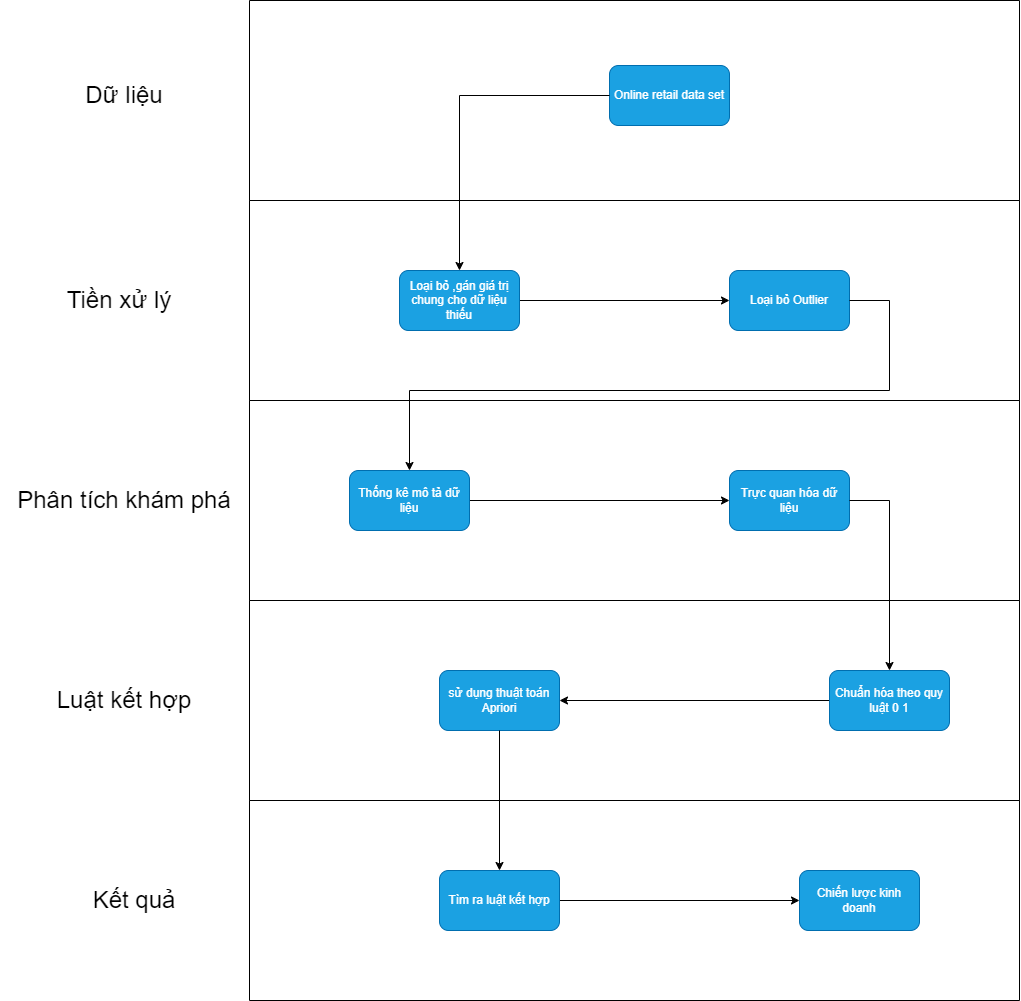
* Từ frequent item + minconf=80%, ta có 6 luật kết hợp:



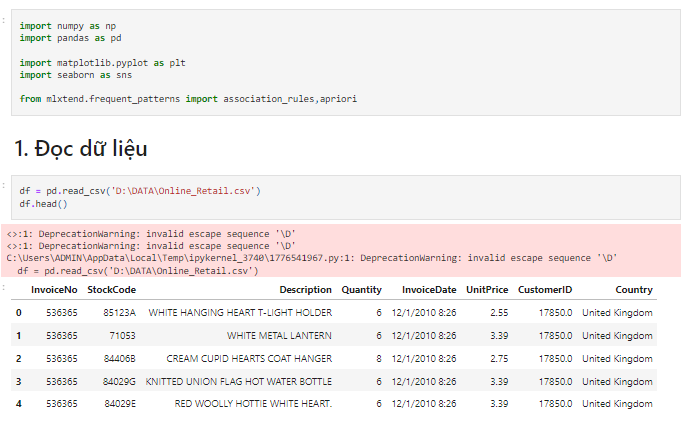
Ưu điểm của FP-Growth:

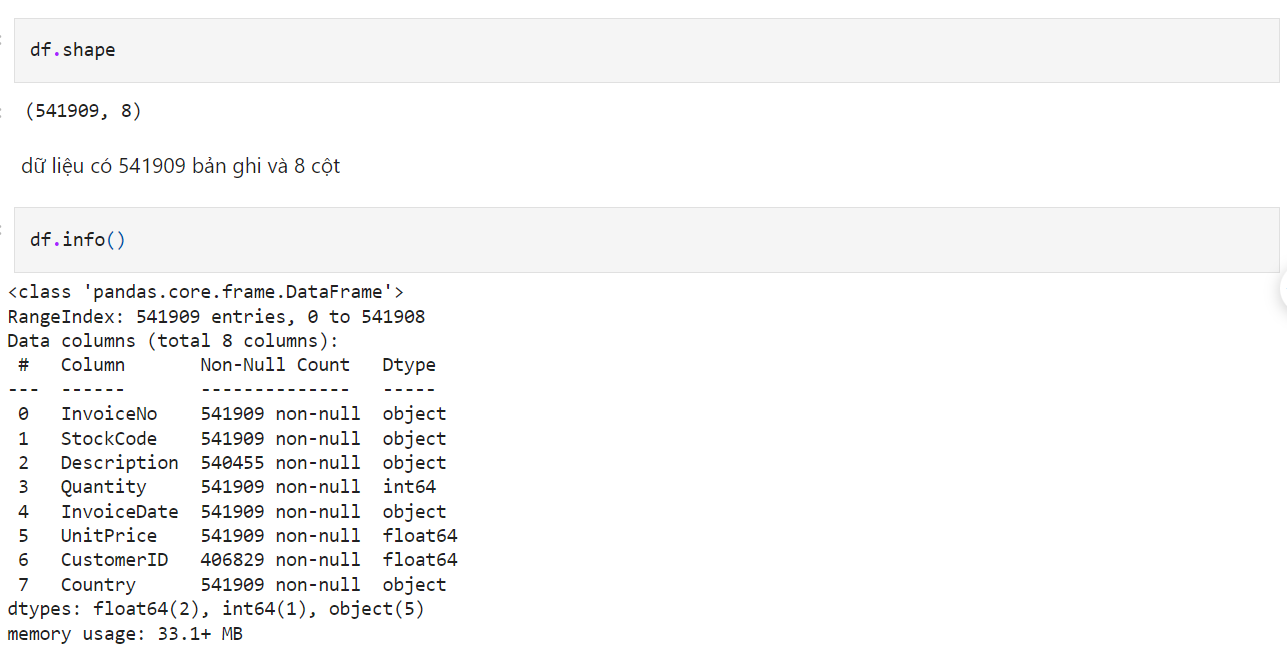
* Apriori có một cách tiếp cận đơn giản khi duyệt qua các tập dữ liệu - nó chỉ tạo và kiểm tra các tập ứng viên lớn hơn dựa trên tập thường xuyên đã biết, trong khi đó các thuật toán khác như FP-Growth yêu cầu cấu trúc dữ liệu phức tạp hơn
* Nguyên tắc cơ bản của Apriori rất rõ ràng - mọi tập con của một tập phổ biến cũng phải là phổ biến, giúp loại bỏ các tập ứng viên không cần thiết mà không cần kiểm tra chúng đối với cơ sở dữ liệu.
* Thuật toán Apriori dễ dàng mở rộng và tùy chỉnh để giải quyết các loại vấn đề khác nhau.
* Apriori cho phép người sử dụng điều chỉnh độ phức tạp của thuật toán thông qua việc đặt ngưỡng hỗ trợ.
* Apriori không phụ thuộc mạnh mẽ vào cấu trúc cụ thể của dữ liệu như FP-Growth, ít rủi ro khi gặp phải những thay đổi nhỏ trong cấu trúc dữ liệu.
* **Eclat algorithm**
* **FP-growth algorith**

1. **Kết quả thực nghiệm**
2. **Frame work**



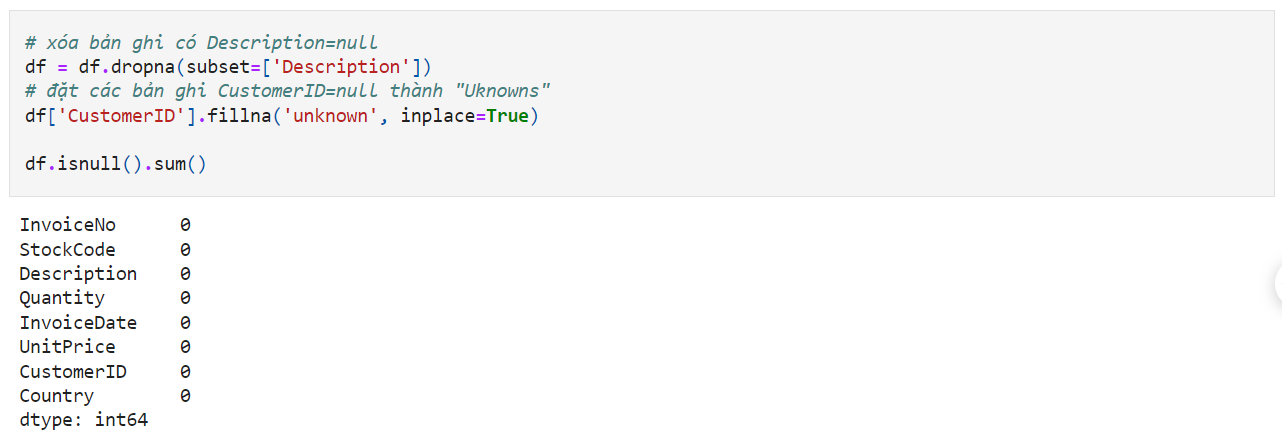
1. **Giải thích dữ liệu**

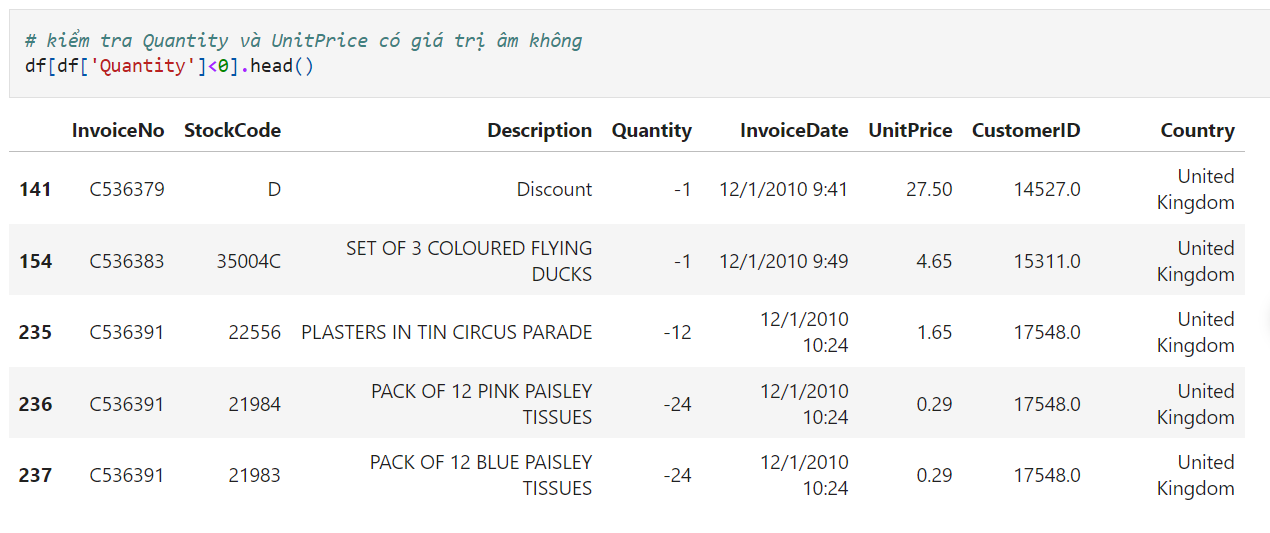
****

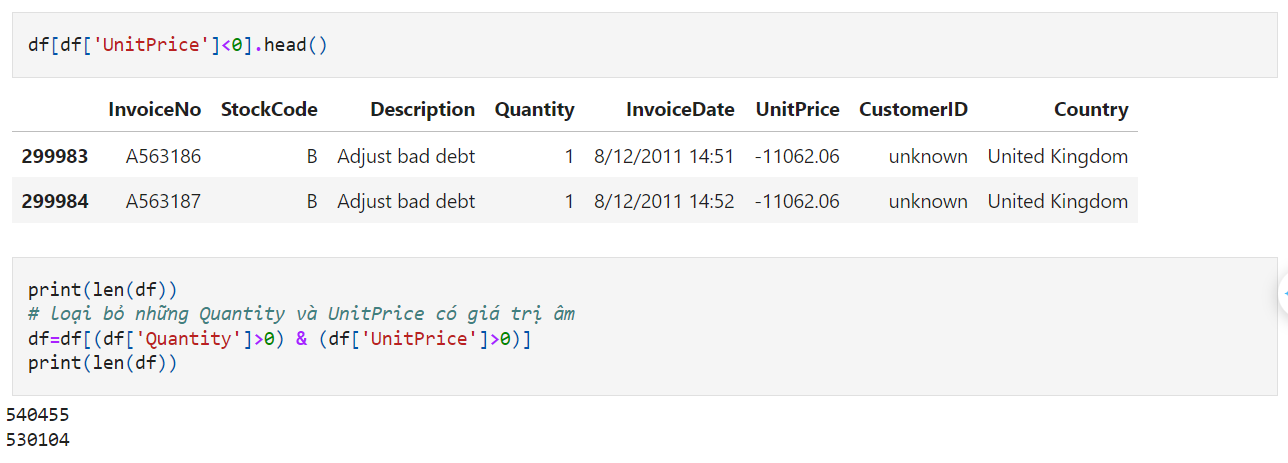
****

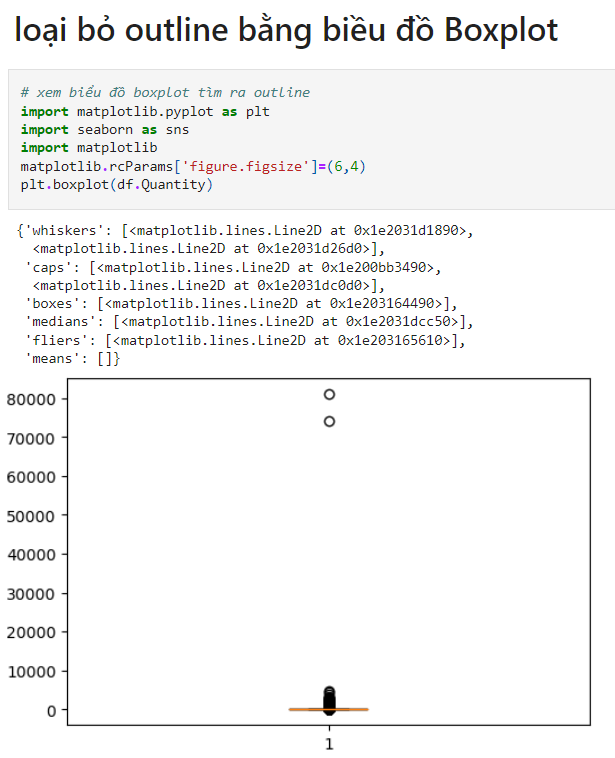
1. **Tiền xử lý dữ liệu**

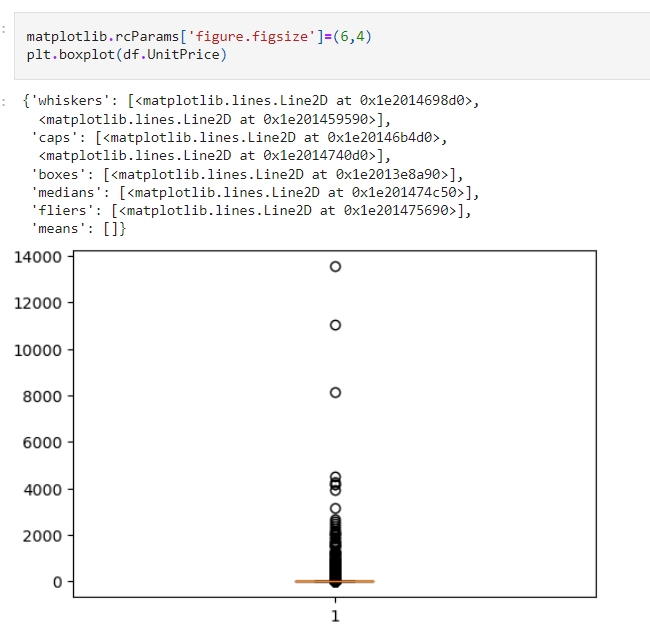


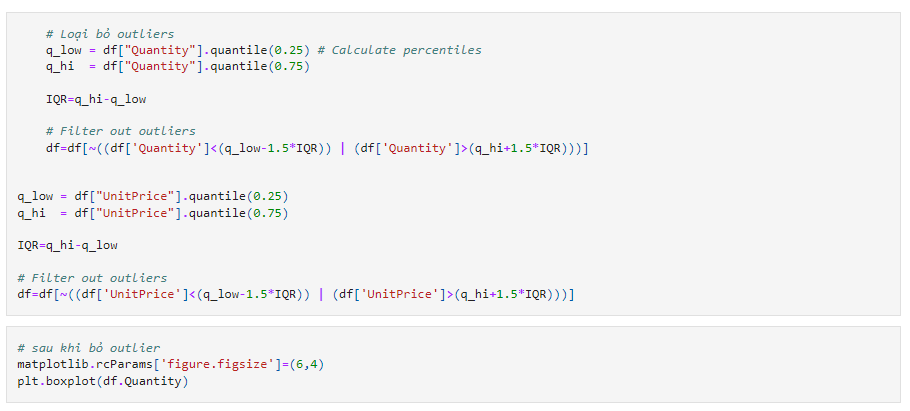


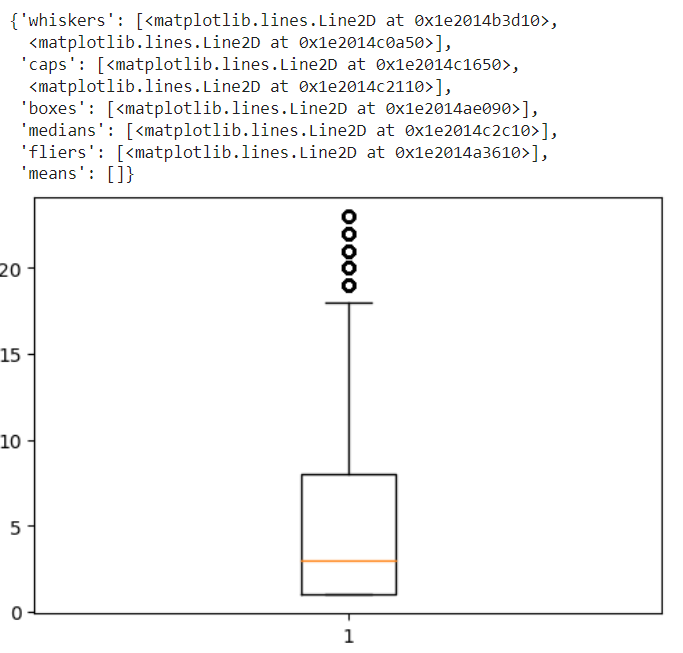


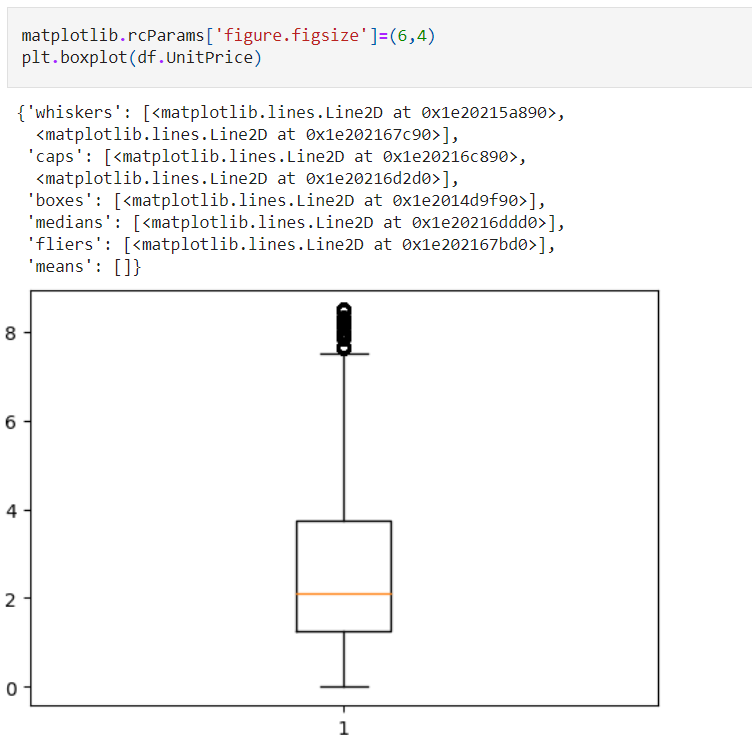






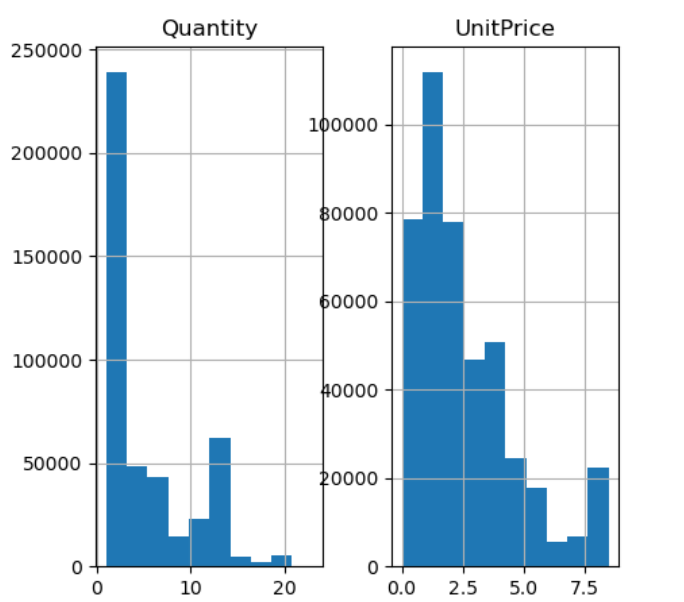






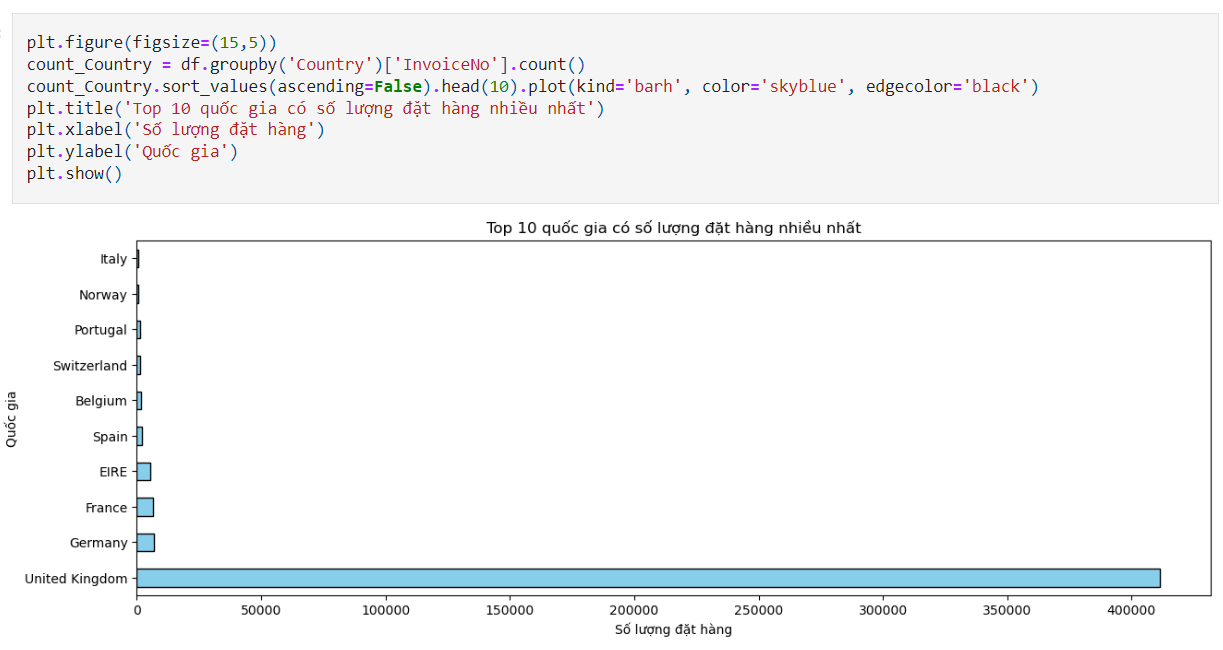
1. **Phân tích khám phá**

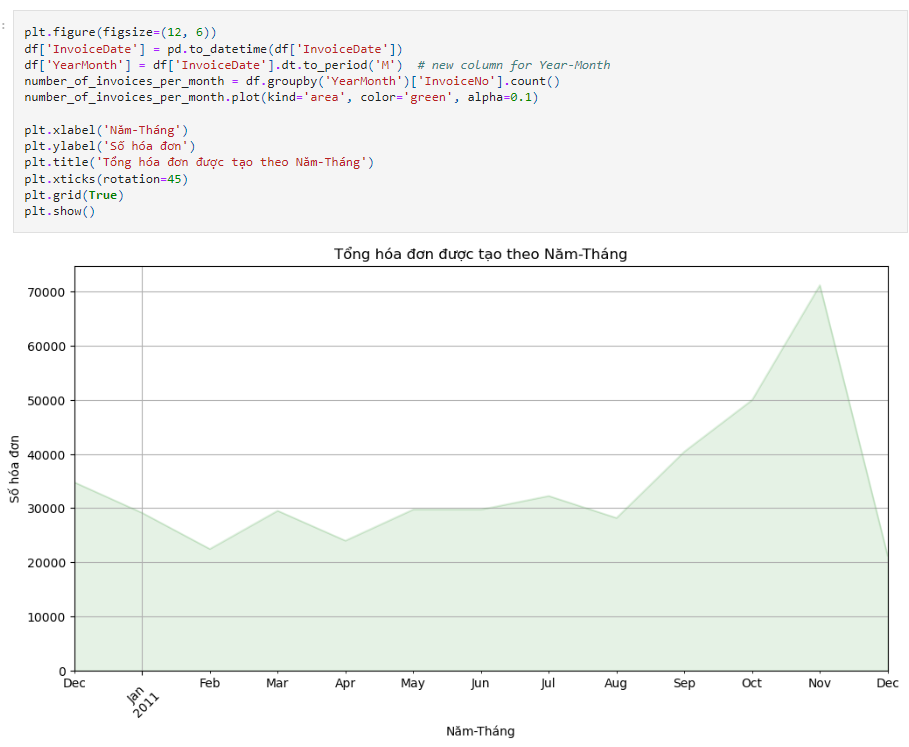


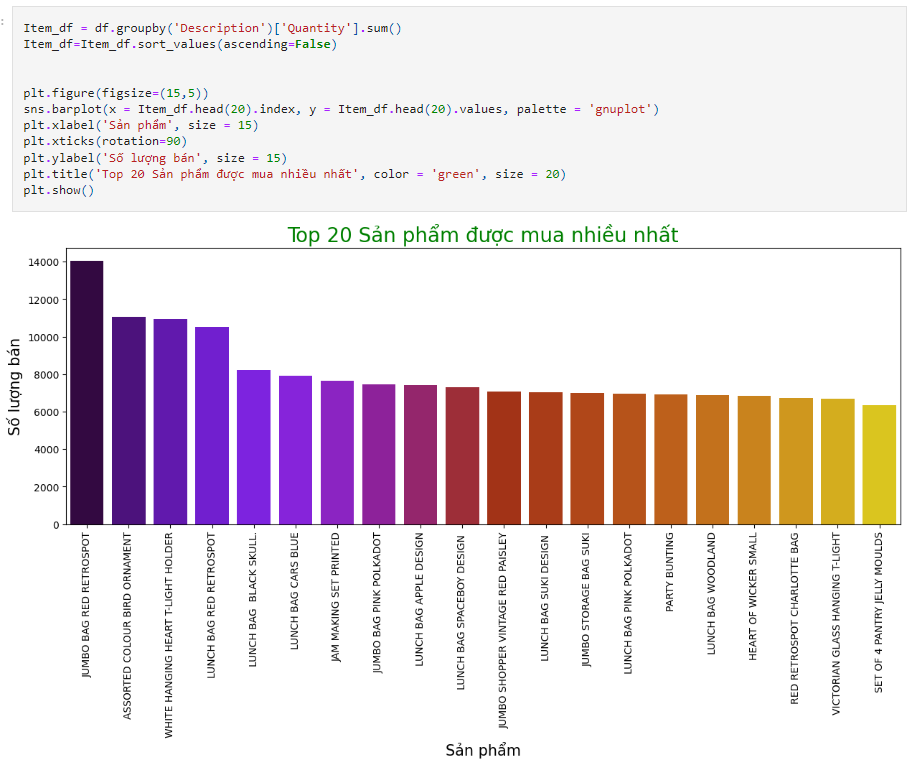




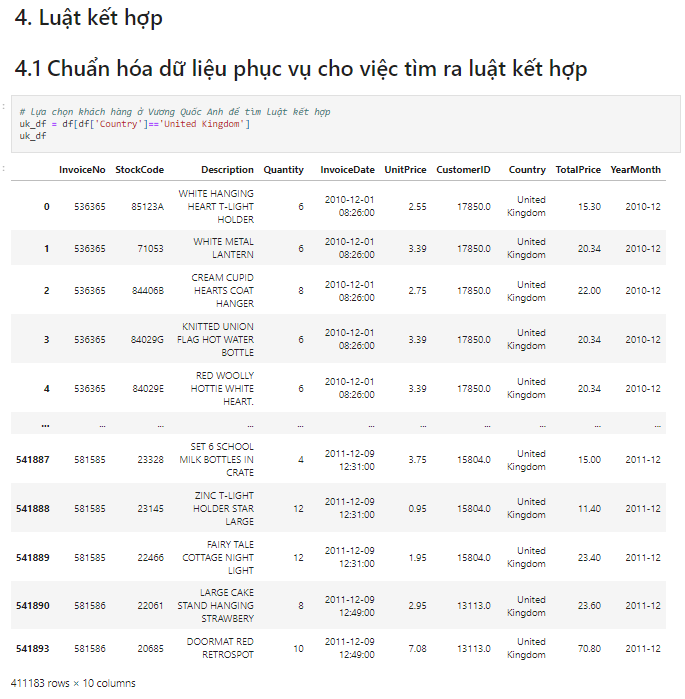


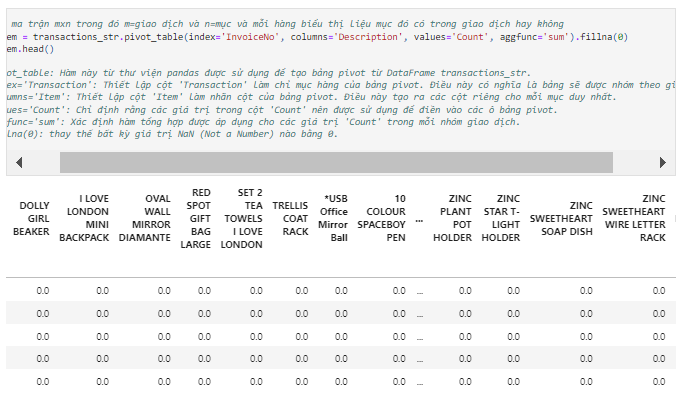


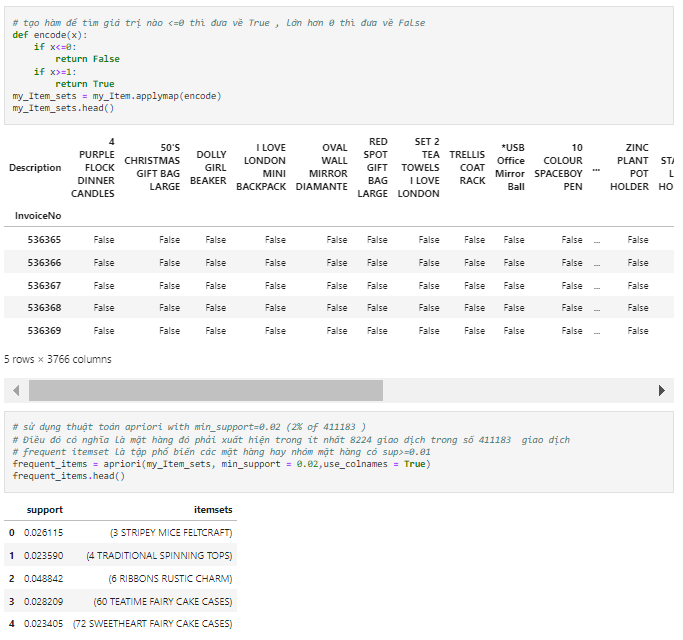


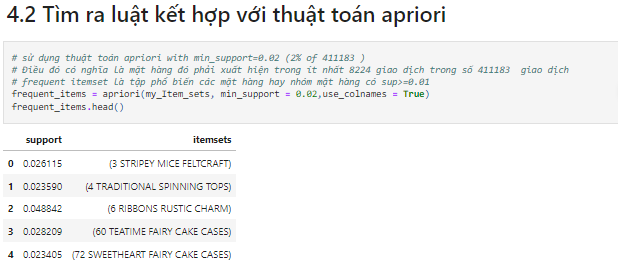


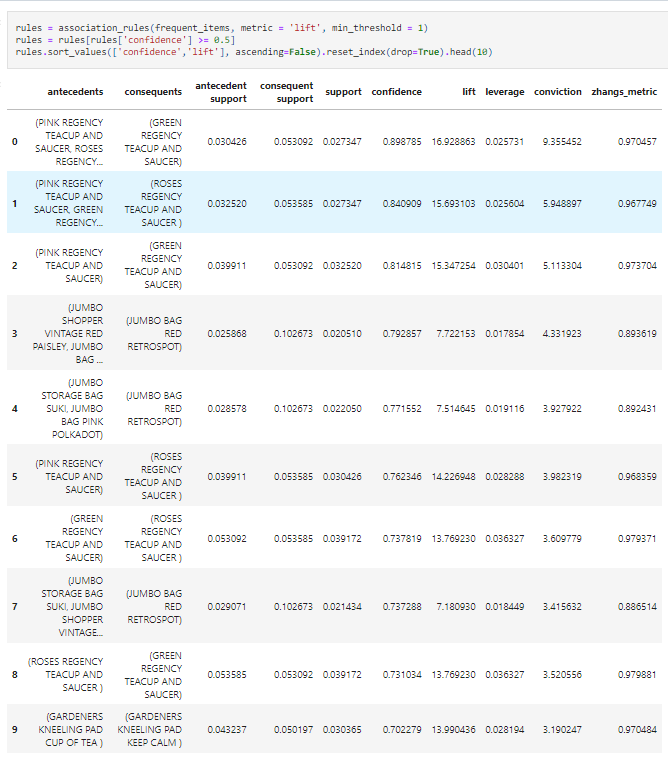
1. **Luật kết hợp**

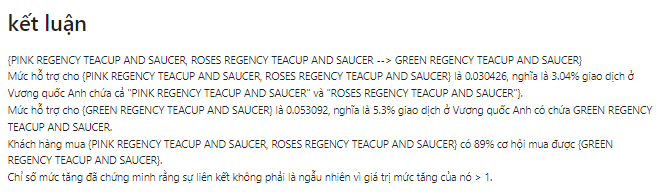












1. **Chiến lược**

* Tập trung vào nhóm khách hàng tiềm năng:
* Khách hàng đã mua "PINK REGENCY TEACUP AND SAUCER" và "ROSES REGENCY TEACUP AND SAUCER".
* Khách hàng ở Vương quốc Anh.
* Đề xuất sản phẩm:
* "GREEN REGENCY TEACUP AND SAUCER".
* Giảm giá cho khách hàng mua "GREEN REGENCY TEACUP AND SAUCER" cùng với "PINK REGENCY TEACUP AND SAUCER" hoặc "ROSES REGENCY TEACUP AND SAUCER".
* Miễn phí vận chuyển cho khách hàng ở Vương quốc Anh.
* Kênh bán hàng:
* Bán hàng trực tuyến thông qua trang web và mạng xã hội.
* Bán hàng tại các cửa hàng bán lẻ.
* Hoạt động quảng bá:
* Quảng cáo trực tuyến nhắm mục tiêu đến nhóm khách hàng tiềm năng.
* Gửi email quảng cáo đến khách hàng đã mua "PINK REGENCY TEACUP AND SAUCER" và "ROSES REGENCY TEACUP AND SAUCER".
* Tham gia các hội chợ và triển lãm liên quan đến đồ gia dụng.

1. **Tài liệu tham khảo**

**Tham khảo qua 2 sách :**

DATA\_MINING\_CONCEPTS\_AND\_TECHNIQUES\_3RD

LEARNING DATA MINING WITH PYTHON

1. **Đường dẫn video**

https://drive.google.com/drive/folders/1skERK94y1Q7XEe\_QLQPjyPgVdz2rRzco?usp=sharing&fbclid=IwAR0tY7FKPT2Mps9-DFEMeNLHiMSpmY0zZBD0ZiSVRj3OhCV8oE71Om0NqvM\_aem\_AUBTOGM\_W\_HYTKUdxk7W1f6Fn3QlIrS4nPx5\_yK2-J\_mG-kIcXjb6sYyNJE2pzsCQ4F\_hfCX0W1gKXaQzEm52LHb