**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

----------🙞🙜🕮🙞🙜----------

**KHAI THÁC DỮ LIỆU**

**Bài thực hành số 2**

***Giảng viên hướng dẫn:***

**ThS. Hà Lê Hoài Trung.**

***Sinh viên thực hiện:***

**Trần Văn Điêp – 18520601.**

***Lớp:*** **IS252.L21**

🙡🙢 Tp. Hồ Chí Minh, 4/2021 🙠🙣

**Mục lục**

[Phần I: Lý Thuyết Tập Phổ Biến Và Tập Kết Hợp 2](#_Toc69202873)

[Phần II: Thực Hành 2](#_Toc69202874)

[1. Hướng dẫn thực hành: 2](#_Toc69202875)

[2. Bài tập thực hành: 9](#_Toc69202876)

[2.1. Bài tập 1: 9](#_Toc69202877)

[2.2. Bài tập 2: 16](#_Toc69202881)

[2.3. Bài tập 3: 20](#_Toc69202882)

[2.4. Bài tập 4: 25](#_Toc69202883)

[2.5. Bài tập 5: 30](#_Toc69202884)

[2.6. Bài tập 6: 42](#_Toc69202885)

# Phần I: Lý Thuyết Tập Phổ Biến Và Tập Kết Hợp

1. Hiểu và vận dụng giải thuật Apriori, FP-growth để tìm các tập phổ biến.

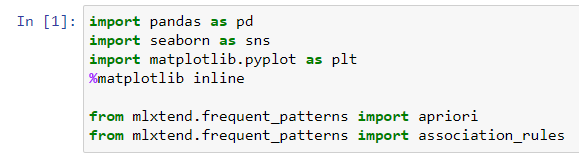
2. Hiểu và vận dụng kỹ thuật tìm luật kết hợp dựa trên tập phổ biến tối đại.

3. Hiểu các phương pháp đánh giá các luật kết hợp tìm được.

# Phần II: Thực Hành

## Hướng dẫn thực hành:

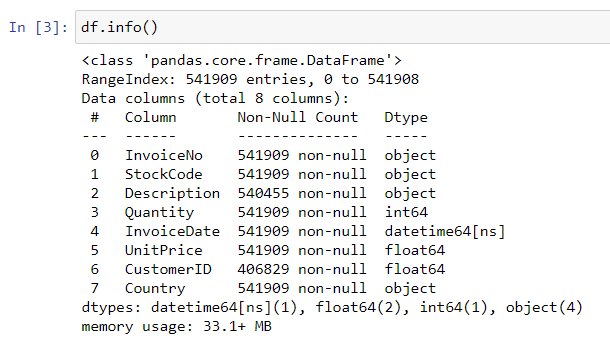
* Hướng dẫn:
* import thư viện cần sử dụng cho bài toán:



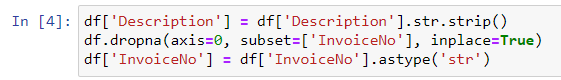
* Đọc file dữ liệu bằng pandas:



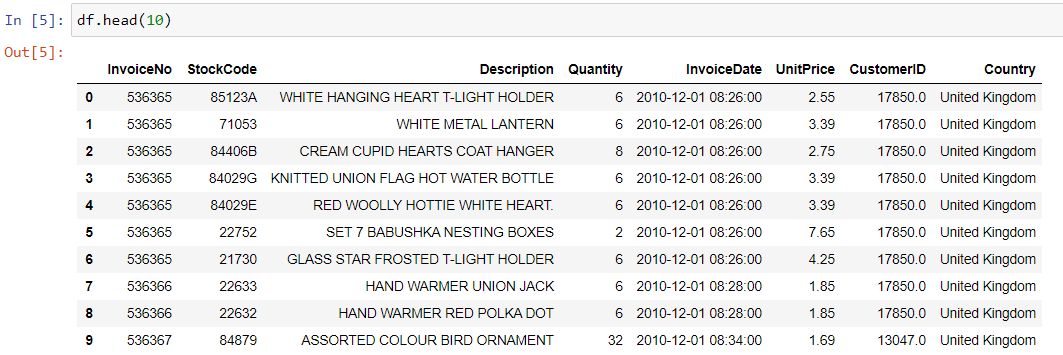
* Hiển thị thông tin của dữ liệu vừa được đọc:



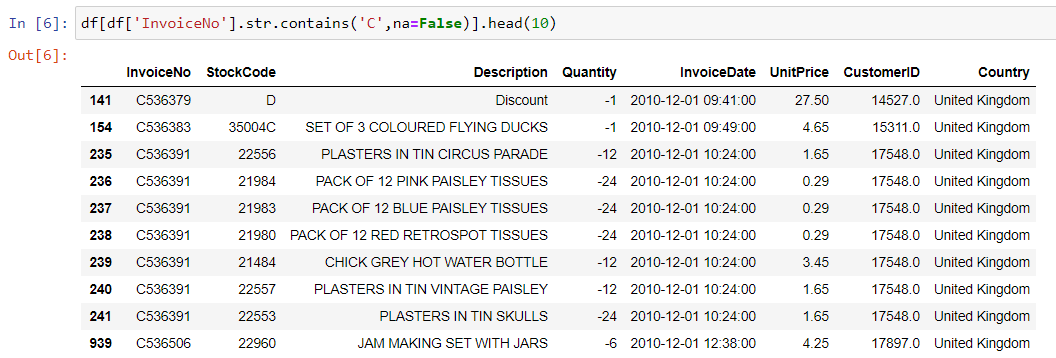
* Cắt bỏ các ký tự thừa ở tên mặt hàng mua (cột Description), xóa các dòng dữ liệu không có số hóa đơn (cột InvoiceNo) và chuyển nó về kiểu dữ liệu chuỗi:



* Hiển thị 10 phần tử đầu của dataframe:



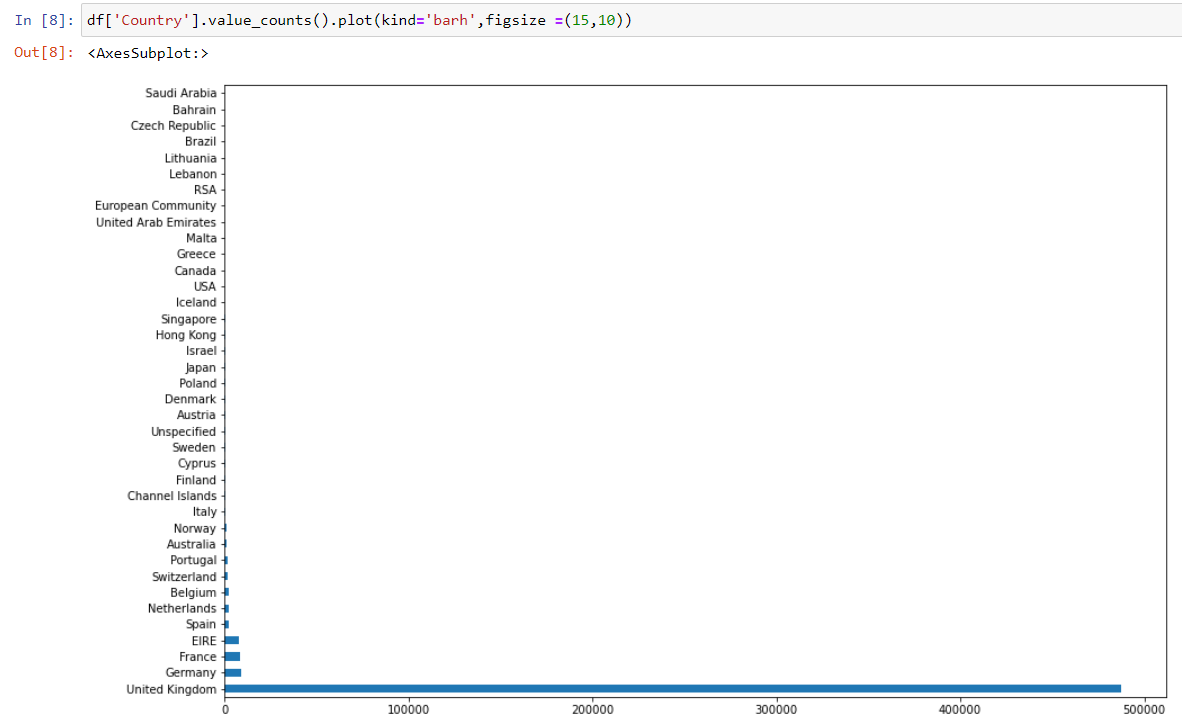
* Hiển thị 10 phần tử đầu hóa đơn tính dụng:



* Tiến hành xóa những hóa đơn tín dụng:



* Thống kê số dòng dữ liệu theo từng quốc gia:



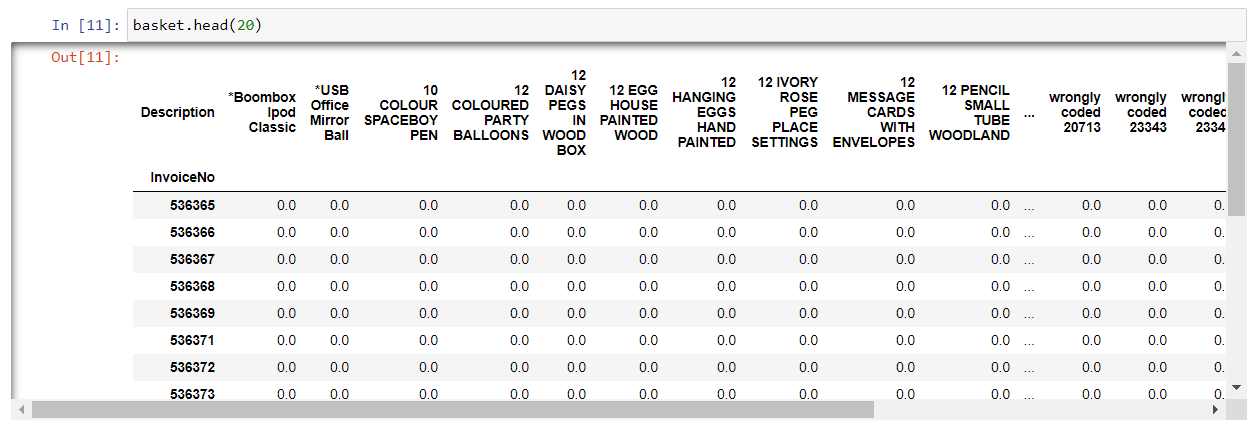
* Chỉ xét các hóa đơn từ nước Anh và nhóm dữ liệu theo Số hóa đơn và Tên mặt hàng:



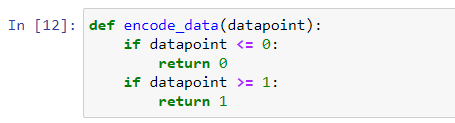
* Chuyển đổi dữ liệu về dạng hot encoding, với mỗi dòng dữ liệu là một hóa đơn:



* Xem dữ liệu sau khi chuyển về dạng hot encoding:



* Tạo hàm biến đổi mỗi điểm dữ liệu có số lượng (Quantity) lớn hơn 0 thành 1:



* Chuyển đổi dữ liệu từ dạng hot encoding thành one-hot encoding:



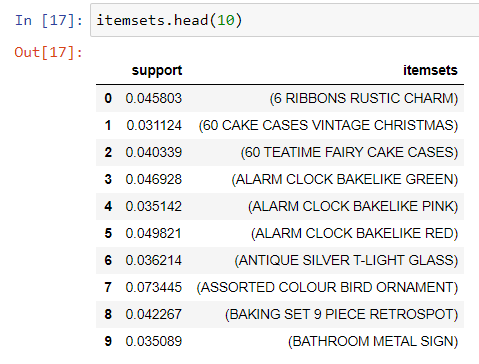
* Xóa cột ‘POSTAGE’:



* Áp dụng thuật toán Apriori với min\_sup = 3% để tìm tập phổ biến:



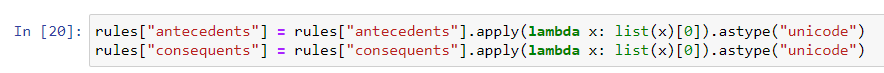
* Xem 10 phần tử đầu tiên trong tập phổ biến tìm được:



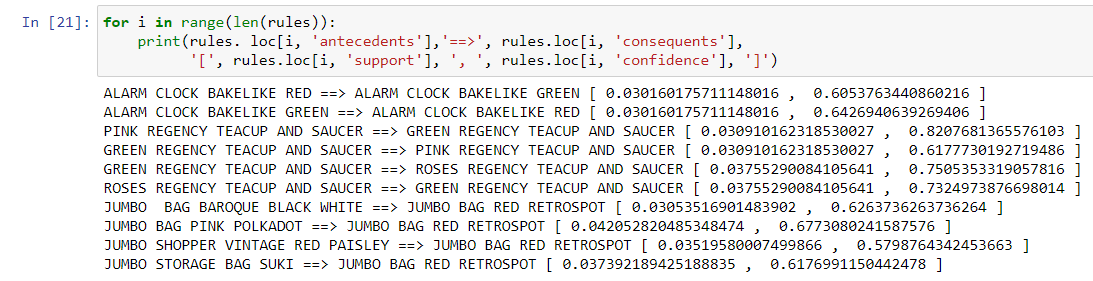
* Tạo luật kết hợp với min\_conf = 50%:



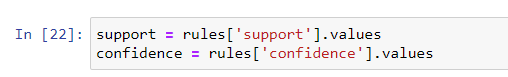
* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):



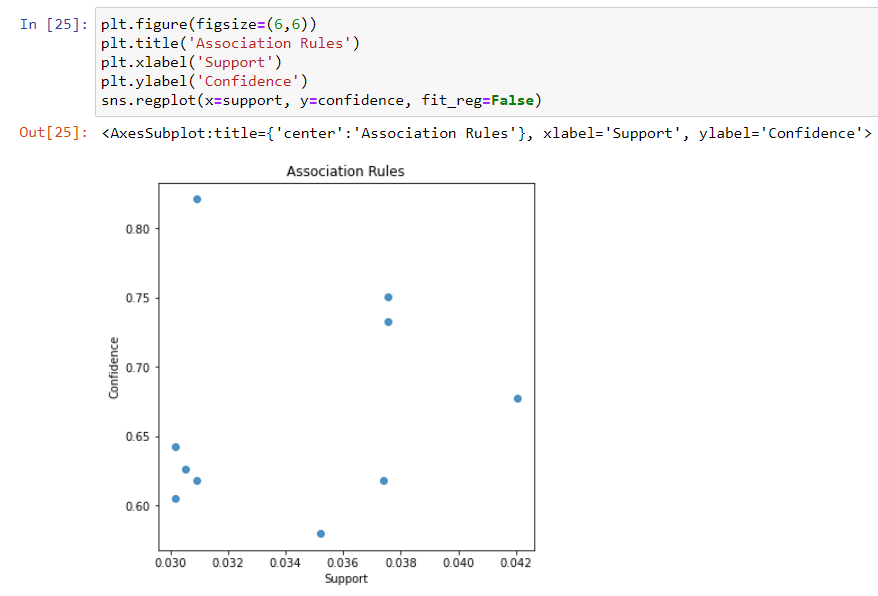
* Viết lệnh in ra các luật đã tìm được:



* Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật:



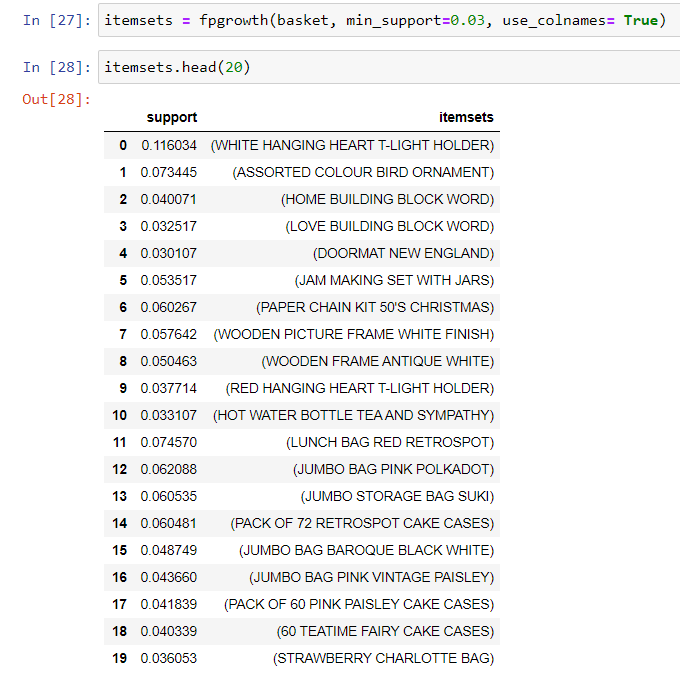
* Biểu diễn các thông tin này lên biểu đồ:



* Import module fpgrowth từ thư viện mlxtend và thực hiện tìm tập phổ biến bằng thuật toán FP-Growth:



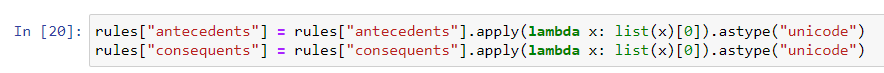
* Áp dụng thuật toán FP-Growth với min\_sup = 3% để tìm tập phổ biến:



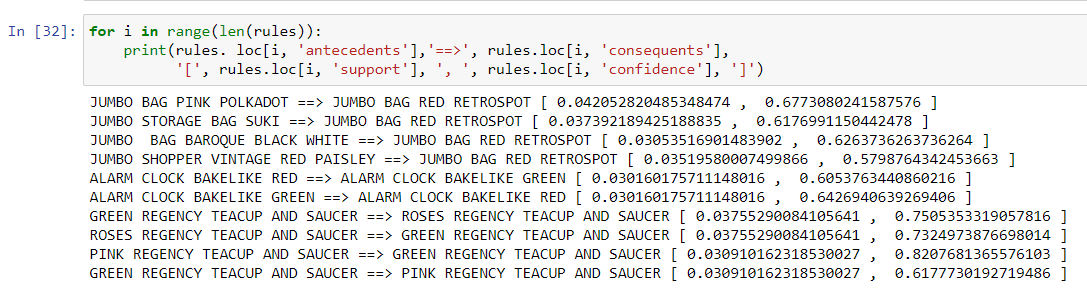
* Tạo luật kết hợp với min\_conf = 50%:



* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):



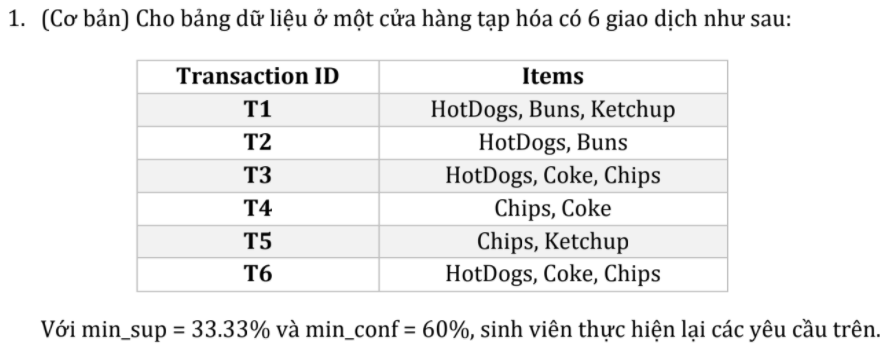
* Viết lệnh in ra các luật đã tìm được:



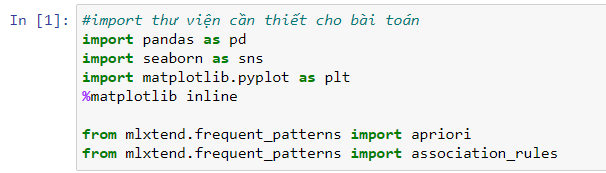
## Bài tập thực hành:

### Bài tập 1:

* Yêu cầu:



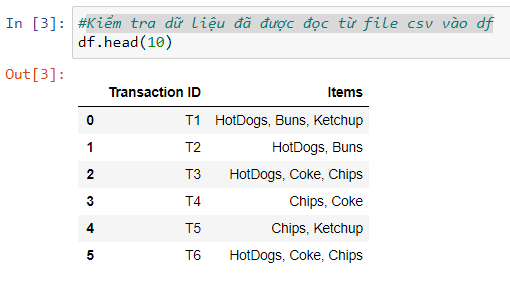
* Giải quyết:
* import thư viện cần thiết cho bài toán:



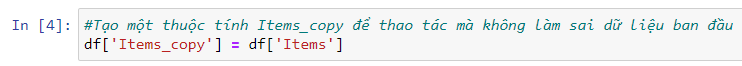
* Đọc dữ liệu từ file csv:



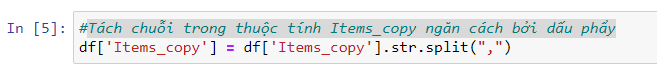
* Kiểm tra dữ liệu đã được đọc từ file csv vào df:



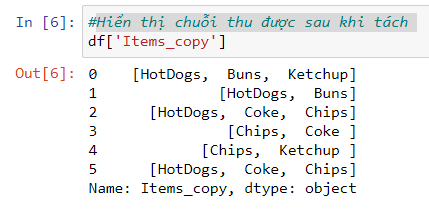
* Tạo một thuộc tính Items\_copy để thao tác mà không làm sai dữ liệu ban đầu:



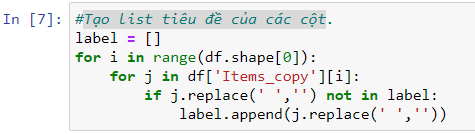
* Tách chuỗi trong thuộc tính Items\_copy ngăn cách bởi dấu phẩy:



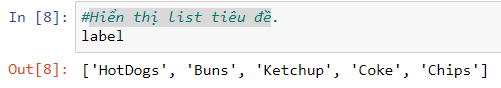
* Hiển thị chuỗi thu được sau khi tách:



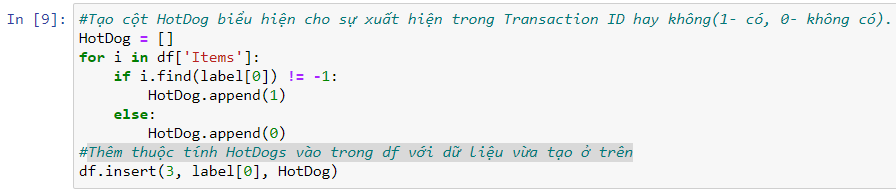
* Tạo list tiêu đề của các cột:



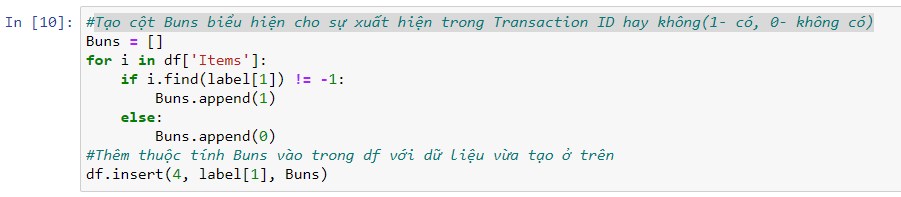
* Hiển thị list tiêu đề:



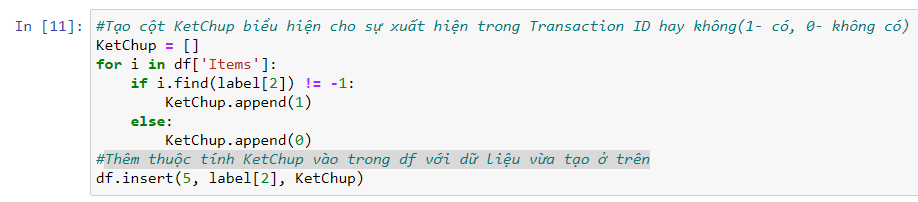
* Tạo cột HotDog biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính HotDogs vào trong df với dữ liệu vừa tạo ở trên:



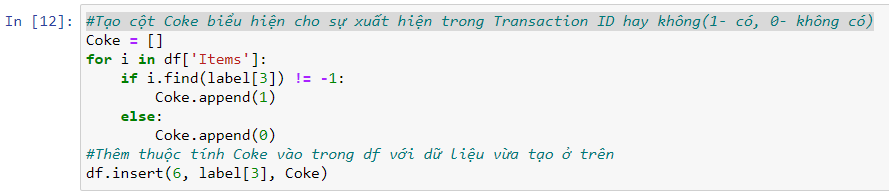
* Tạo cột Buns biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính Buns vào trong df với dữ liệu vừa tạo ở trên:



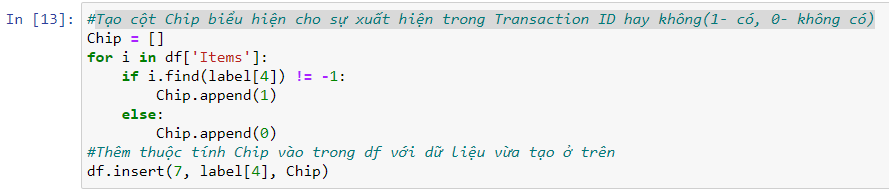
* Tạo cột KetChup biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính KetChup vào trong df với dữ liệu vừa tạo ở trên:



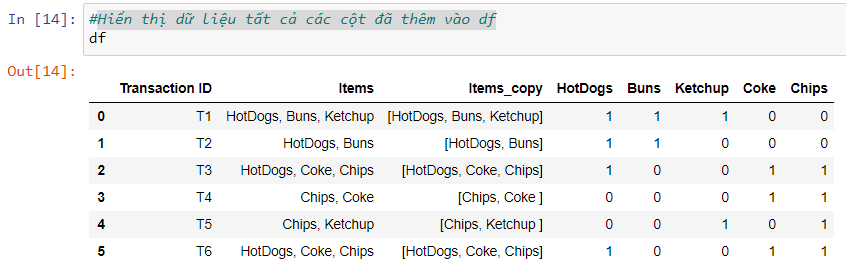
* Tạo cột Coke biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính Coke vào trong df với dữ liệu vừa tạo ở trên:



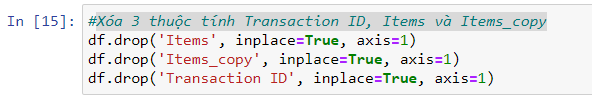
* Tạo cột Chip biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính Chip vào trong df với dữ liệu vừa tạo ở trên:



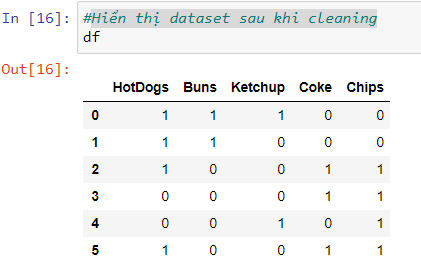
* Hiển thị dữ liệu tất cả các cột đã thêm vào df:



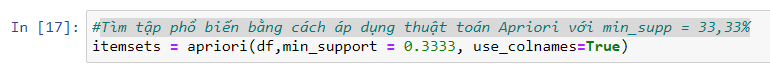
* Xóa 3 thuộc tính Transaction ID, Items và Items\_copy:



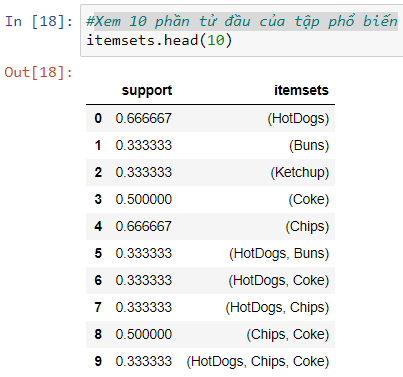
* Hiển thị dataset sau khi cleaning:



* Tìm tập phổ biến bằng cách áp dụng thuật toán Apriori với min\_supp = 33,33%:



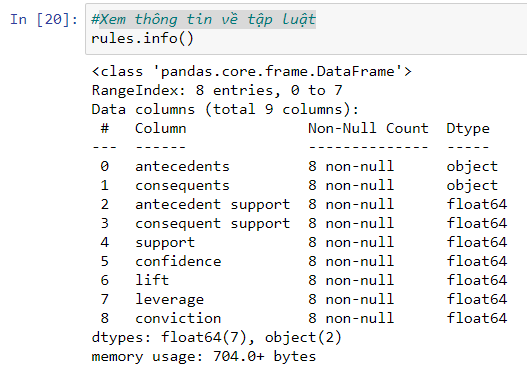
* Xem 10 phần tử đầu của tập phổ biến:



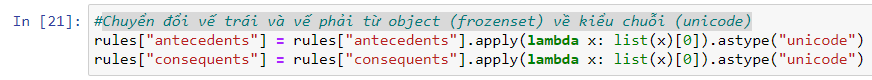
* Tạo luật kết hợp với min\_conf = 60%:



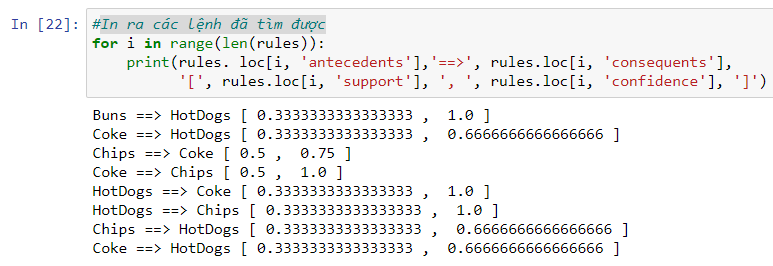
* Xem thông tin về tập luật:



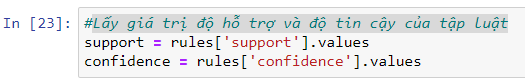
* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



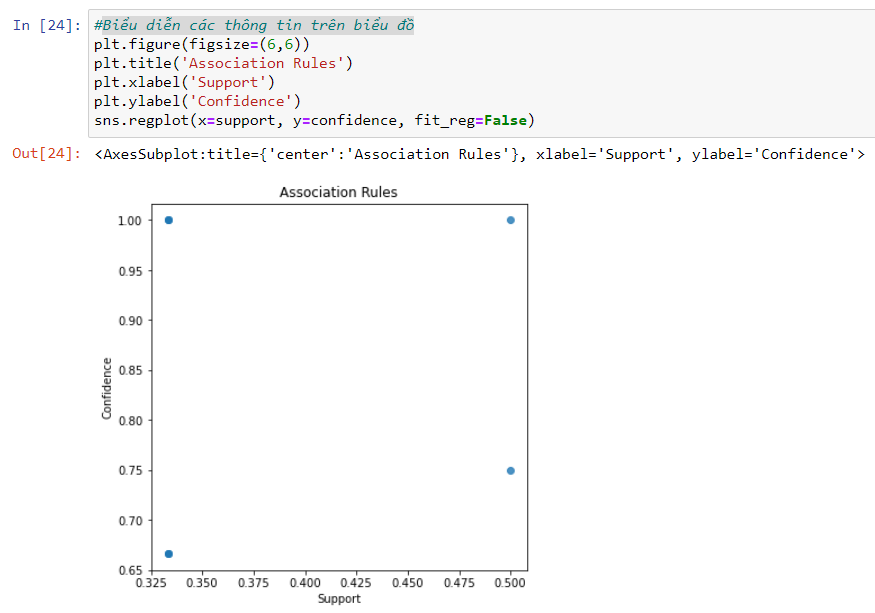
* In ra các lệnh đã tìm được:



* Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật:



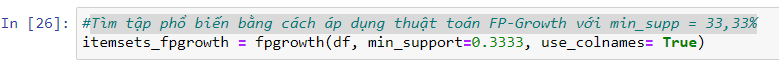
* Biểu diễn các thông tin trên biểu đồ:



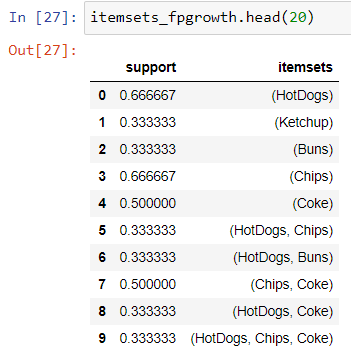
* import thư viện fpgrowth để tìm tập phổ biến thông qua giải thuật FP-Growth:



* Tìm tập phổ biến bằng cách áp dụng thuật toán FP-Growth với min\_supp = 33,33%:



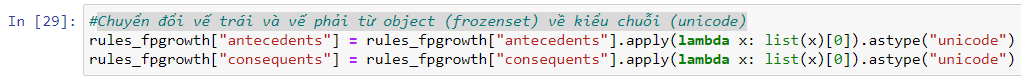
* Xem 10 phần tử đầu của tập phổ biến:



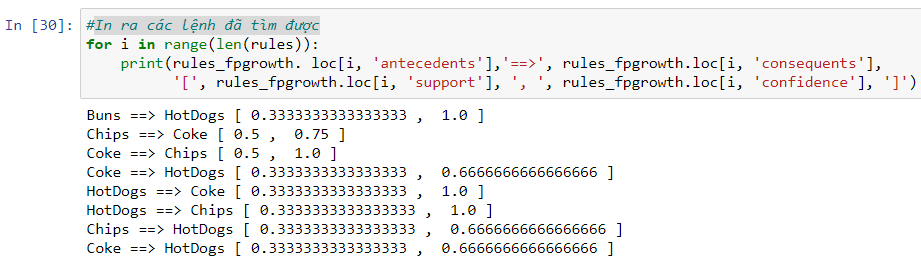
* Tạo luật kết hợp với min\_conf = 60%:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



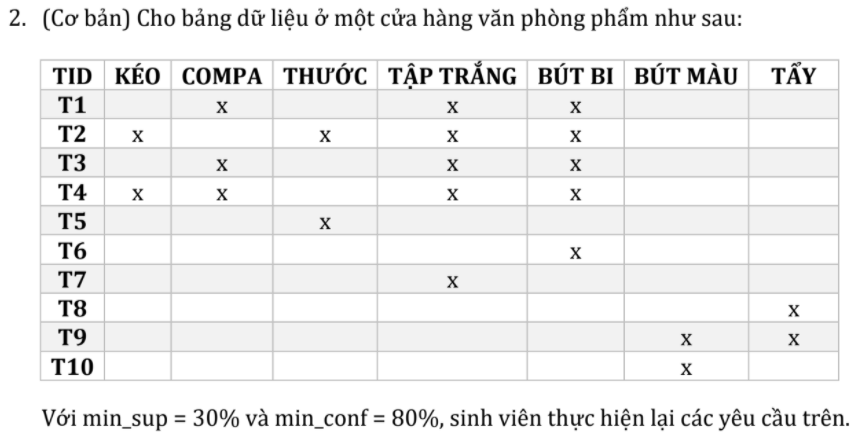
* In ra các lệnh đã tìm được:



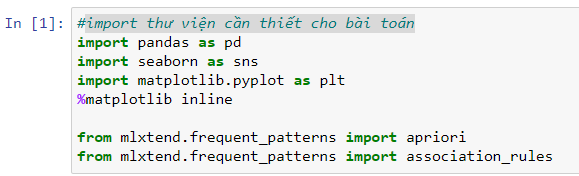


### Bài tập 2:

* Yêu cầu:



* Giải quyết:
* import thư viện cần thiết cho bài toán:



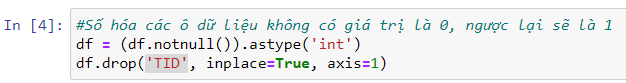
* Đọc dữ liệu từ file csv:



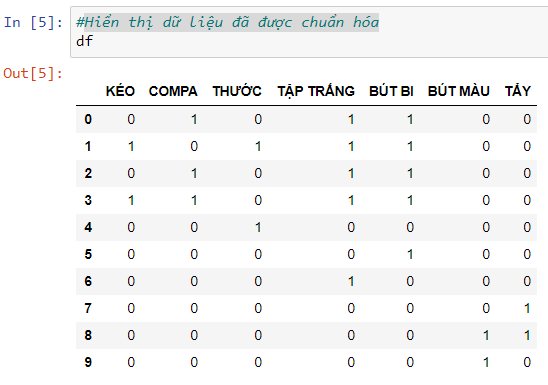
* Hiển thị 10 phần tử từ dữ liệu vừa đọc:



* Số hóa các ô dữ liệu không có giá trị là 0, ngược lại sẽ là 1, xóa cột dữ liệu 'TID':



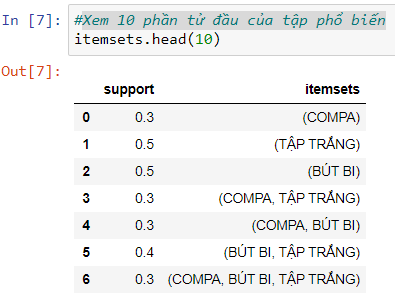
* Hiển thị dữ liệu đã được chuẩn hóa:



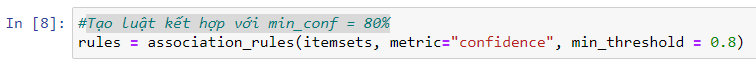
* Tìm tập phổ biến bằng cách áp dụng thuật toán Apriori với min\_supp = 30%:



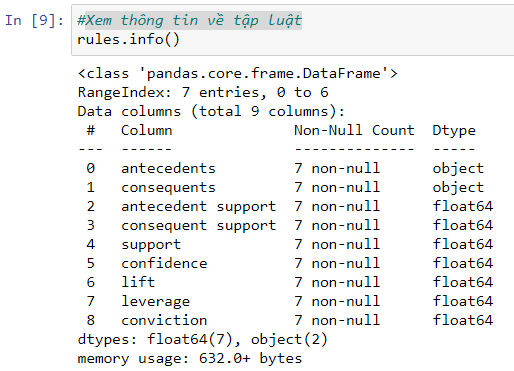
* Xem 10 phần tử đầu của tập phổ biến:



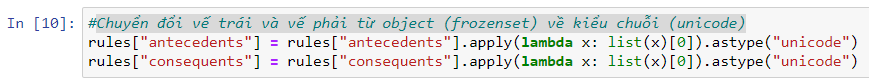
* Tạo luật kết hợp với min\_conf = 80%:



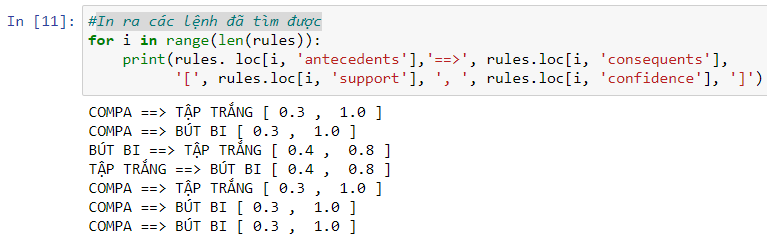
* Xem thông tin về tập luật:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



* In ra các lệnh đã tìm được:



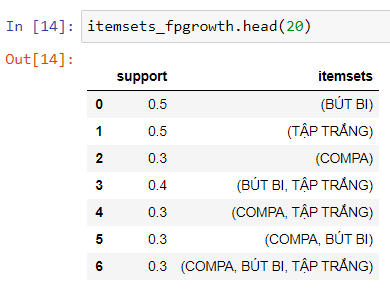
* import thư viện fpgrowth để tìm tập phổ biến thông qua giải thuật FP-Growth:



* Tìm tập phổ biến bằng cách áp dụng thuật toán FP-Growth với min\_supp = 30%:



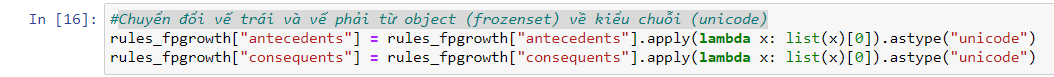
* Xem 20 phần tử đầu của tập phổ biến:



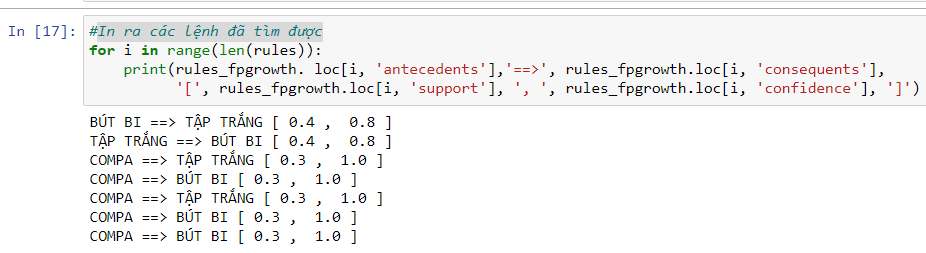
* Tạo luật kết hợp với min\_conf = 80%:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



* In ra các lệnh đã tìm được:

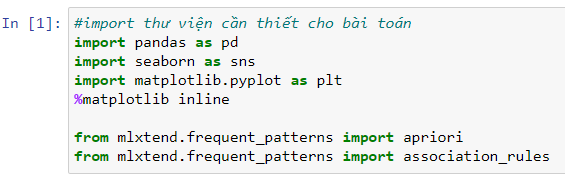


### Bài tập 3:

* Yêu cầu:



* Giải quyết:
* import thư viện cần thiết cho bài toán:



* Đọc dữ liệu từ file csv:



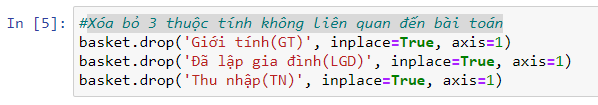
* Hiển thị 10 phần tử đầu vừa đọc từ file:



* Tạo một dataframe chứa các thuộc tính cần xử lý:



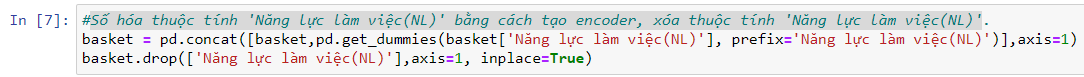
* Xóa bỏ 3 thuộc tính không liên quan đến bài toán:



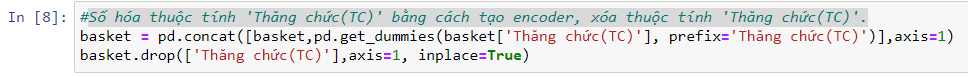
* Hiển thị 10 phần tử đầu sau khi xóa bỏ những thuộc tính dư thừa:



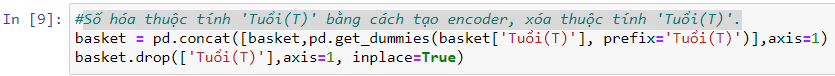
* Số hóa thuộc tính 'Năng lực làm việc(NL)' bằng cách tạo encoder, xóa thuộc tính 'Năng lực làm việc(NL)':



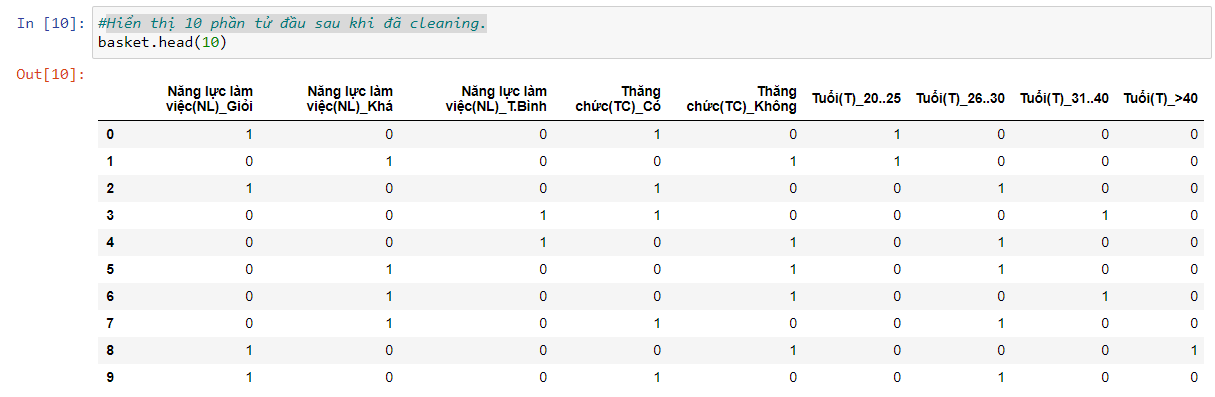
* Số hóa thuộc tính 'Thăng chức(TC)' bằng cách tạo encoder, xóa thuộc tính 'Thăng chức(TC)':



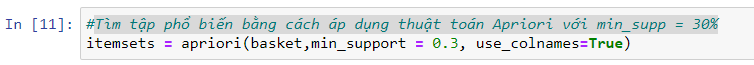
* Số hóa thuộc tính 'Tuổi(T)' bằng cách tạo encoder, xóa thuộc tính 'Tuổi(T)':



* Hiển thị 10 phần tử đầu sau khi đã cleaning:



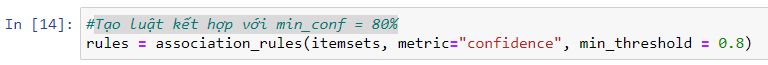
* Tìm tập phổ biến bằng cách áp dụng thuật toán Apriori với min\_supp = 30%:



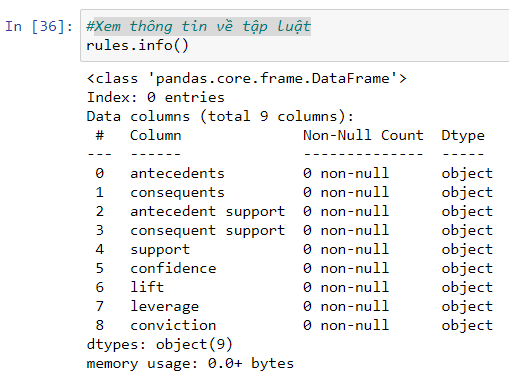
* Xem 10 phần tử đầu của tập phổ biến:



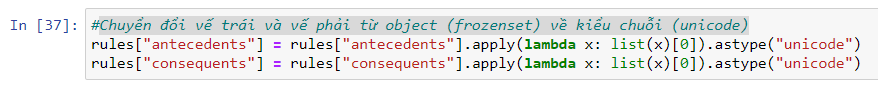
* Tạo luật kết hợp với min\_conf = 80%:



* Xem thông tin về tập luật:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



* In ra các lệnh đã tìm được:



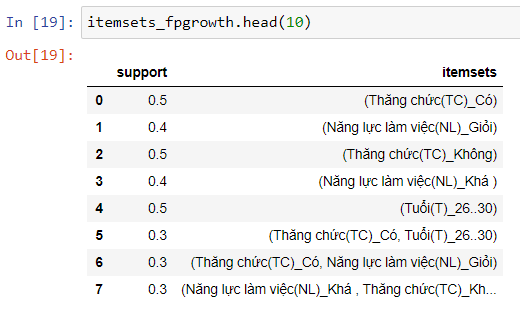
* import thư viện fpgrowth để tìm tập phổ biến thông qua giải thuật FP-Growth:



* Tìm tập phổ biến bằng cách áp dụng thuật toán FP-Growth với min\_supp = 30%:



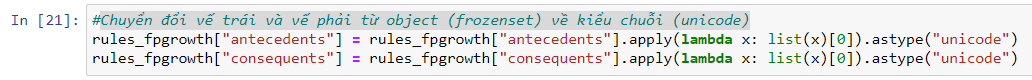
* Hiển thị 10 phần tử đầu tiên trong tập phổ biến:



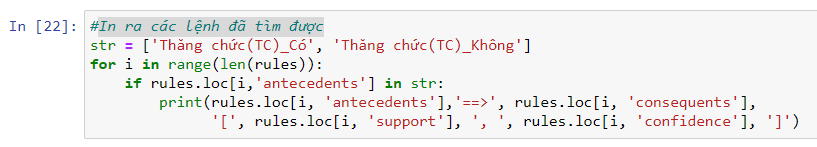
* Tạo luật kết hợp với min\_conf = 80%:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode):



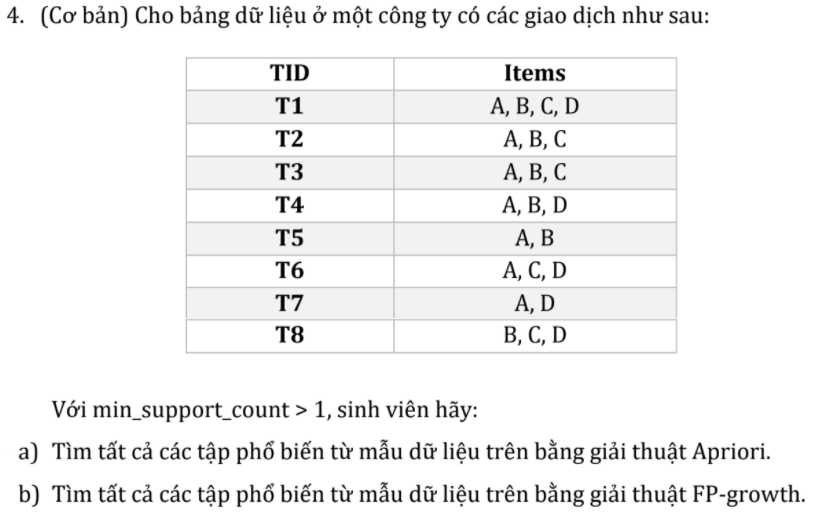
* In ra các lệnh đã tìm được:



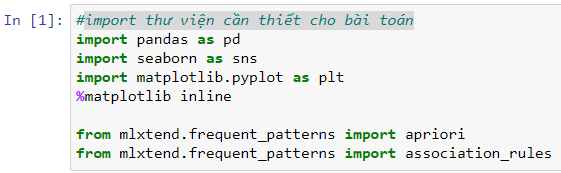
* Kết luận:
* Không có luật kết hợp nào phù hợp với mức min\_sup và mức min\_conf đã cho.

### Bài tập 4:

* Yêu cầu:



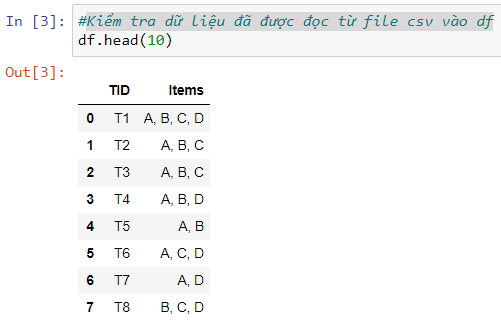
* Giải quyết:
* import thư viện cần thiết cho bài toán:



* Đọc dữ liệu từ file csv:



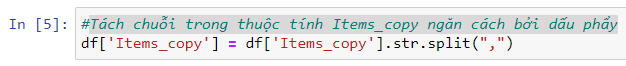
* Kiểm tra dữ liệu đã được đọc từ file csv vào df:



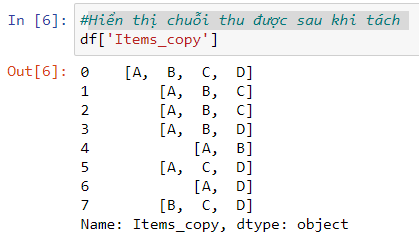
* Tạo một thuộc tính Items\_copy để thao tác mà không làm sai dữ liệu ban dầu:



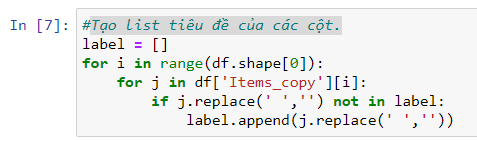
* Tách chuỗi trong thuộc tính Items\_copy ngăn cách bởi dấu phẩy:



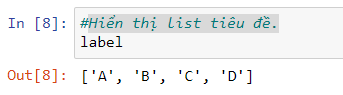
* Hiển thị chuỗi thu được sau khi tách:



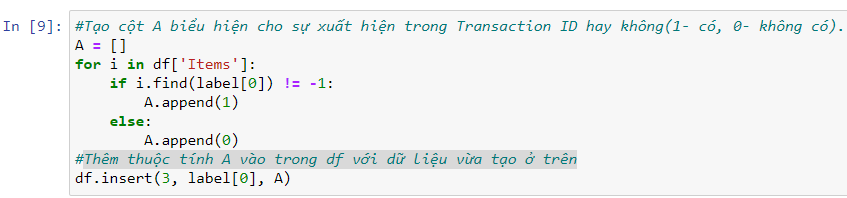
* Tạo list tiêu đề của các cột:



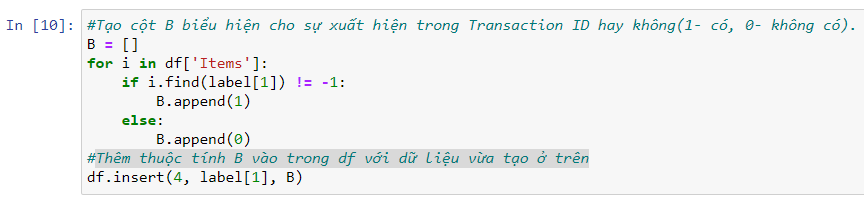
* Hiển thị list tiêu đề:



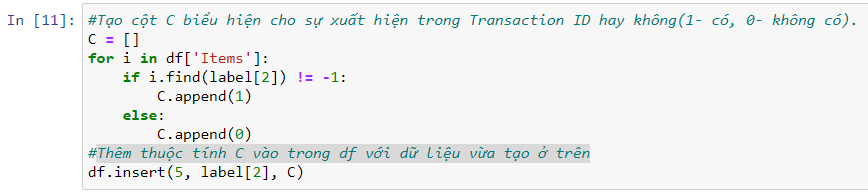
* Tạo cột A biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính A vào trong df với dữ liệu vừa tạo ở trên:



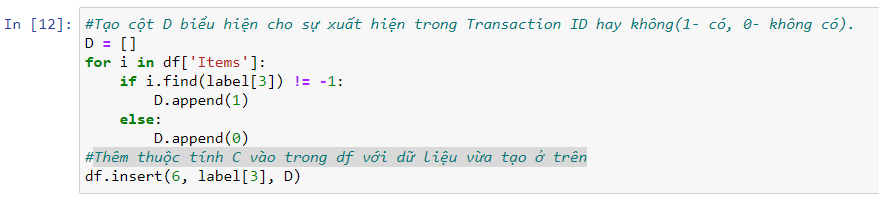
* Tạo cột B biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính B vào trong df với dữ liệu vừa tạo ở trên:



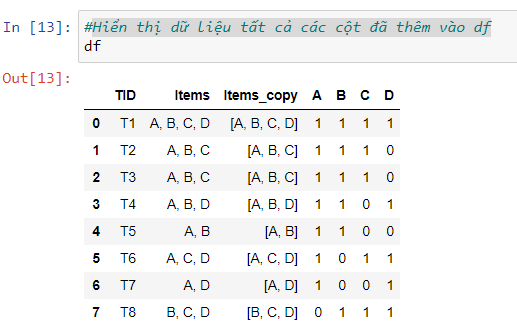
* Tạo cột C biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính C vào trong df với dữ liệu vừa tạo ở trên:



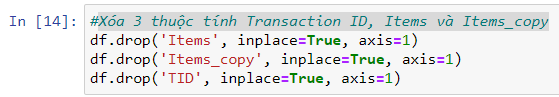
* Tạo cột D biểu hiện cho sự xuất hiện trong Transaction ID hay không(1- có, 0- không có), Thêm thuộc tính C vào trong df với dữ liệu vừa tạo ở trên:



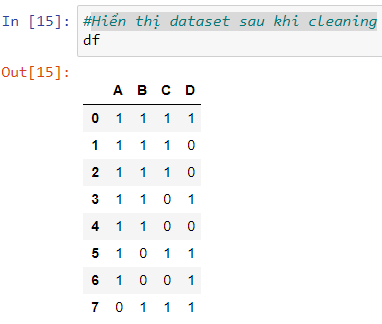
* Hiển thị dữ liệu tất cả các cột đã thêm vào df:



* Xóa 3 thuộc tính Transaction ID, Items và Items\_copy:



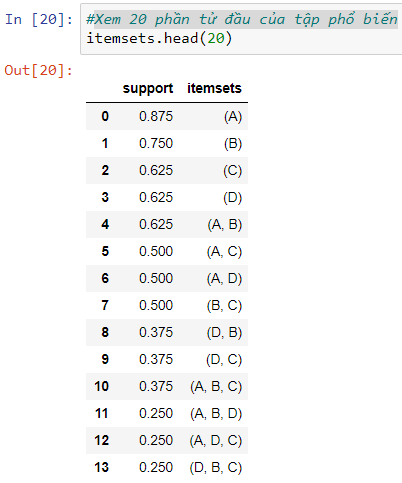
* Hiển thị dataset sau khi cleaning:



* Tìm tập phổ biến bằng cách áp dụng thuật toán Apriori với min\_supp\_count > 1 <=> min\_supp > 12,5%:



* Xem 20 phần tử đầu của tập phổ biến:



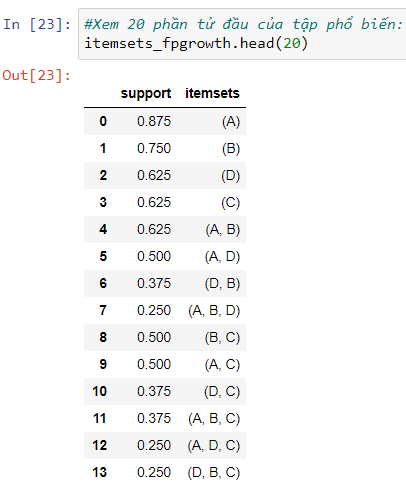
* import thư viện fpgrowth để tìm tập phổ biến thông qua giải thuật FP-Growth:



* Tìm tập phổ biến bằng cách áp dụng thuật toán FP-Growth với min\_supp\_count > 1 <=> min\_supp > 12,5%:

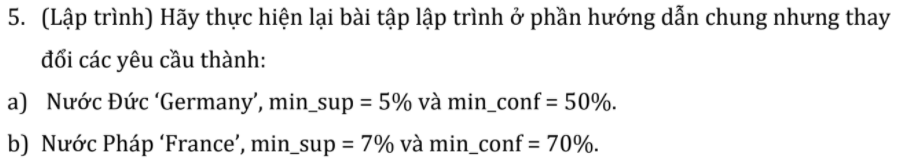


* Xem 20 phần tử đầu của tập phổ biến:



### Bài tập 5:

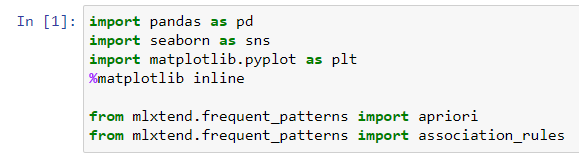
* Yêu cầu:



* Giải quyết:

1. Giải quyết yêu cầu 5.a:

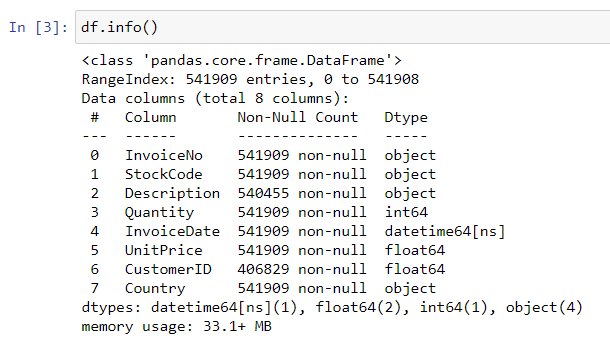
* import thư viện cần sử dụng cho bài toán:



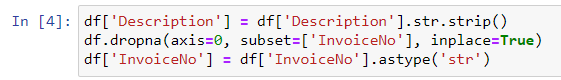
* Đọc file dữ liệu bằng pandas:



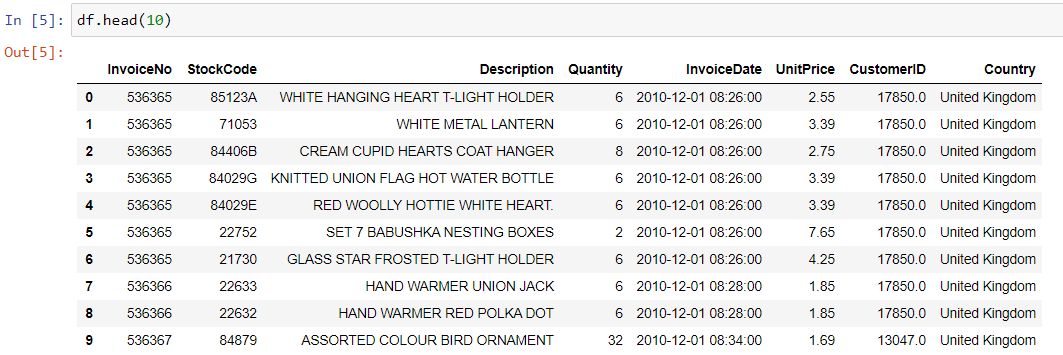
* Hiển thị thông tin của dữ liệu vừa được đọc:



* Cắt bỏ các ký tự thừa ở tên mặt hàng mua (cột Description), xóa các dòng dữ liệu không có số hóa đơn (cột InvoiceNo) và chuyển nó về kiểu dữ liệu chuỗi:



* Hiển thị 10 phần tử đầu của dataframe:



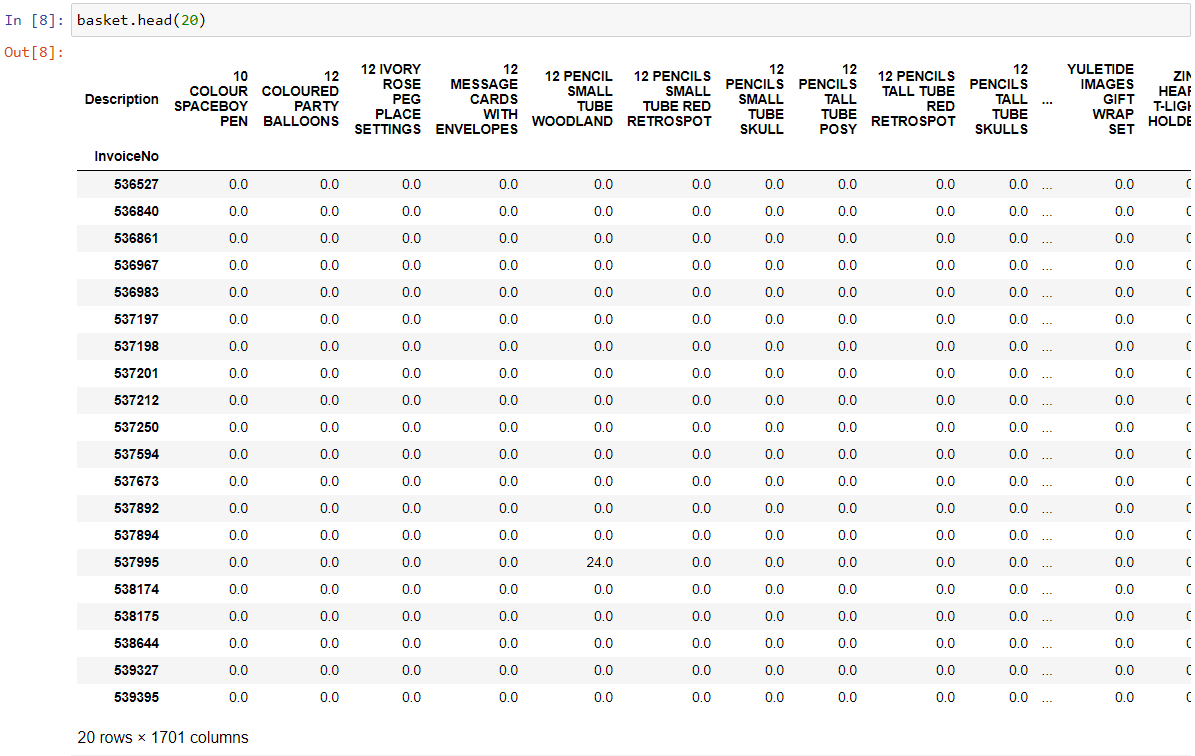
* Chỉ xét các hóa đơn từ nước Germany và nhóm dữ liệu theo Số hóa đơn và Tên mặt hàng:



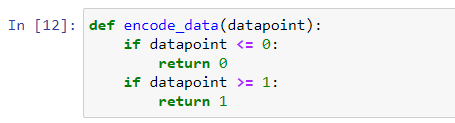
* Chuyển đổi dữ liệu về dạng hot encoding, với mỗi dòng dữ liệu là một hóa đơn:



* Xem dữ liệu sau khi chuyển về dạng hot encoding:



* Tạo hàm biến đổi mỗi điểm dữ liệu có số lượng (Quantity) lớn hơn 0 thành 1:



* Chuyển đổi dữ liệu từ dạng hot encoding thành one-hot encoding:



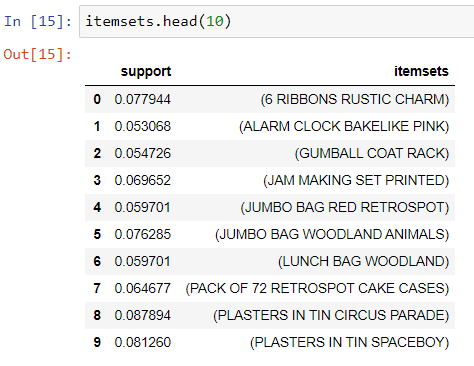
* Xóa cột ‘POSTAGE’:



* Áp dụng thuật toán Apriori với min\_sup = 5% để tìm tập phổ biến:



* Xem 10 phần tử đầu tiên trong tập phổ biến tìm được:



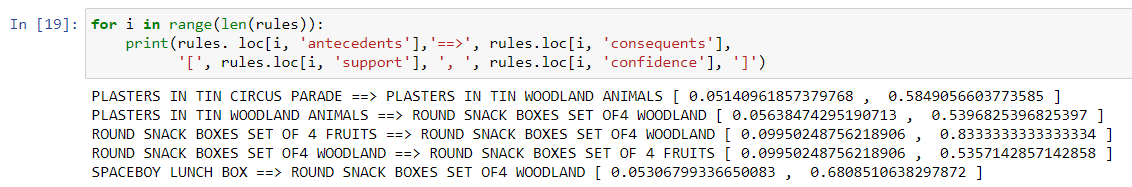
* Tạo luật kết hợp với min\_conf = 50%:



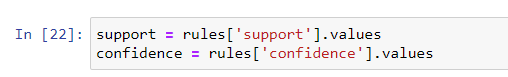
* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):



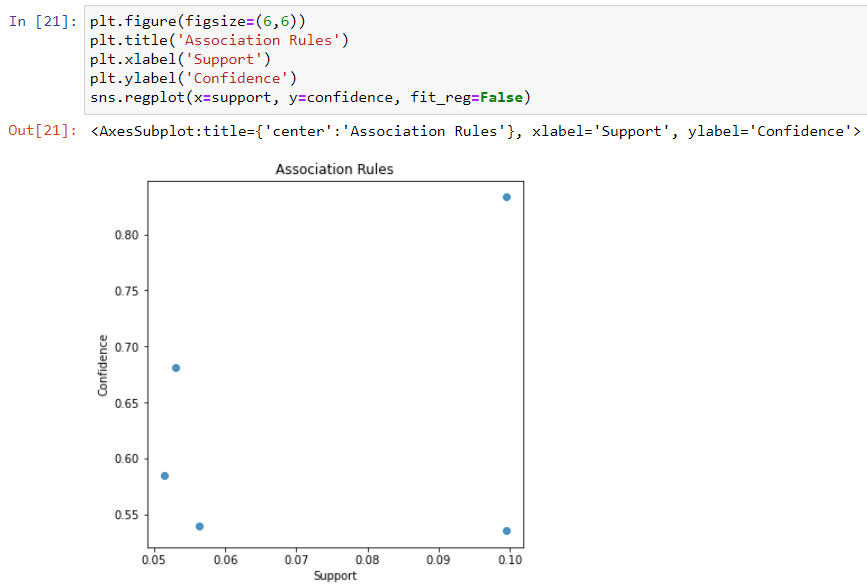
* Viết lệnh in ra các luật đã tìm được:



* Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật:



* Biểu diễn các thông tin này lên biểu đồ:



* Import module fpgrowth từ thư viện mlxtend và thực hiện tìm tập phổ biến bằng thuật toán FP-Growth:



* Áp dụng thuật toán FP-Growth với min\_sup = 5% để tìm tập phổ biến:



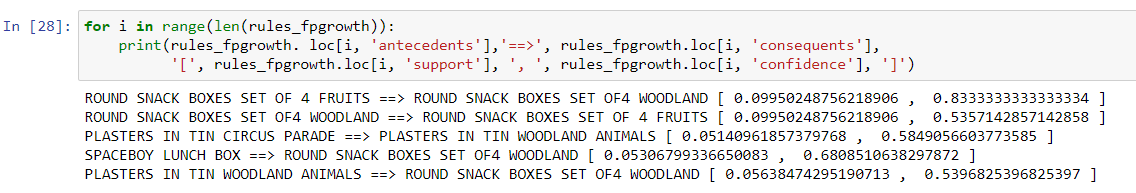
* Tạo luật kết hợp với min\_conf = 50%:



* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):

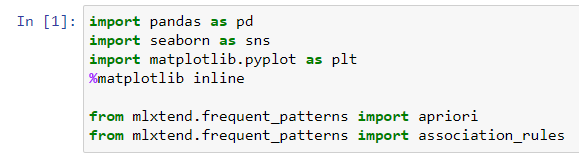


* Viết lệnh in ra các luật đã tìm được:



1. Giải quyết yêu cầu 5.b:

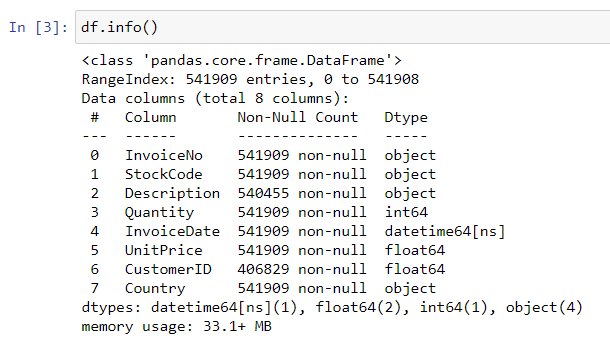
* import thư viện cần sử dụng cho bài toán:



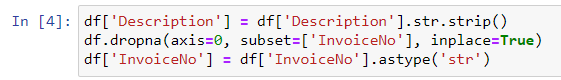
* Đọc file dữ liệu bằng pandas:



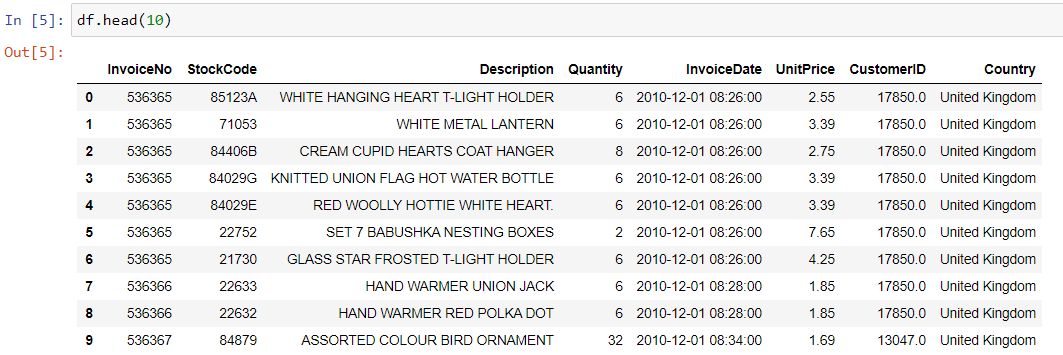
* Hiển thị thông tin của dữ liệu vừa được đọc:



* Cắt bỏ các ký tự thừa ở tên mặt hàng mua (cột Description), xóa các dòng dữ liệu không có số hóa đơn (cột InvoiceNo) và chuyển nó về kiểu dữ liệu chuỗi:



* Hiển thị 10 phần tử đầu của dataframe:



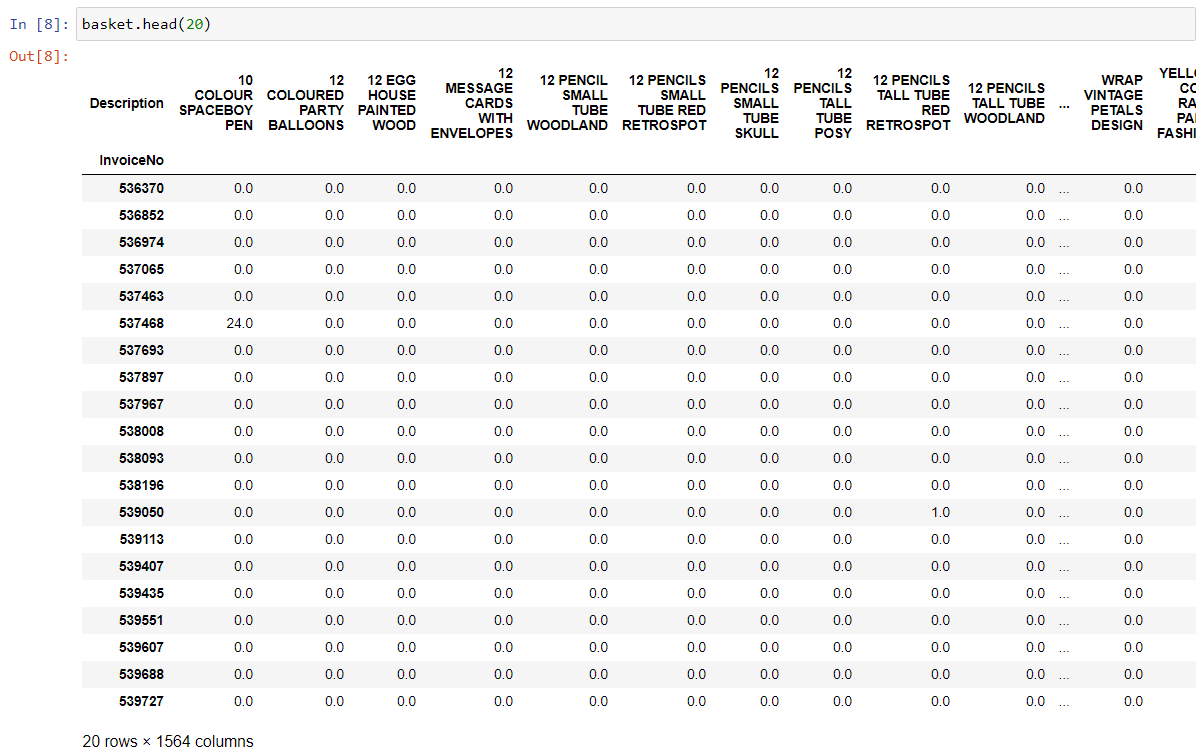
* Chỉ xét các hóa đơn từ nước France và nhóm dữ liệu theo Số hóa đơn và Tên mặt hàng:



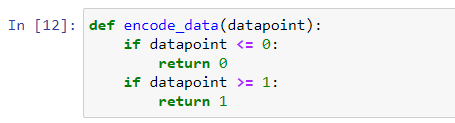
* Chuyển đổi dữ liệu về dạng hot encoding, với mỗi dòng dữ liệu là một hóa đơn:



* Xem dữ liệu sau khi chuyển về dạng hot encoding:



* Tạo hàm biến đổi mỗi điểm dữ liệu có số lượng (Quantity) lớn hơn 0 thành 1:



* Chuyển đổi dữ liệu từ dạng hot encoding thành one-hot encoding:



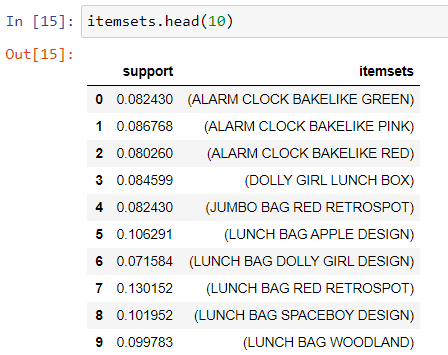
* Xóa cột ‘POSTAGE’:



* Áp dụng thuật toán Apriori với min\_sup = 7% để tìm tập phổ biến:



* Xem 10 phần tử đầu tiên trong tập phổ biến tìm được:



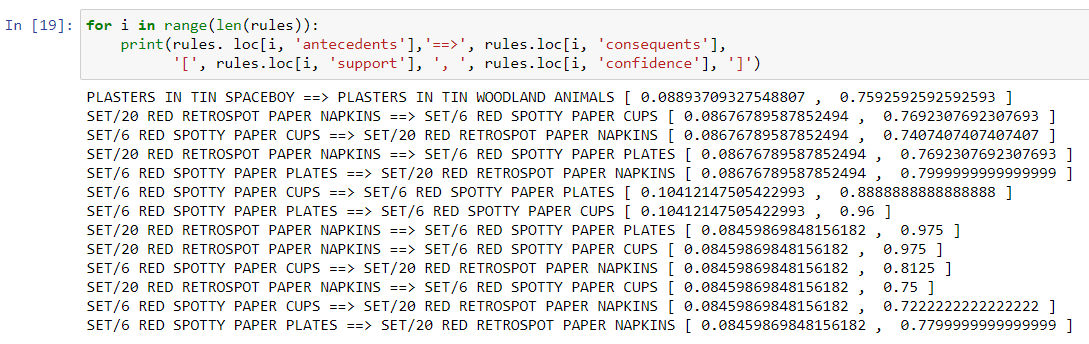
* Tạo luật kết hợp với min\_conf = 70%:



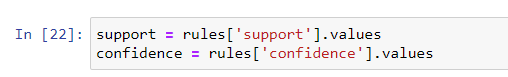
* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):



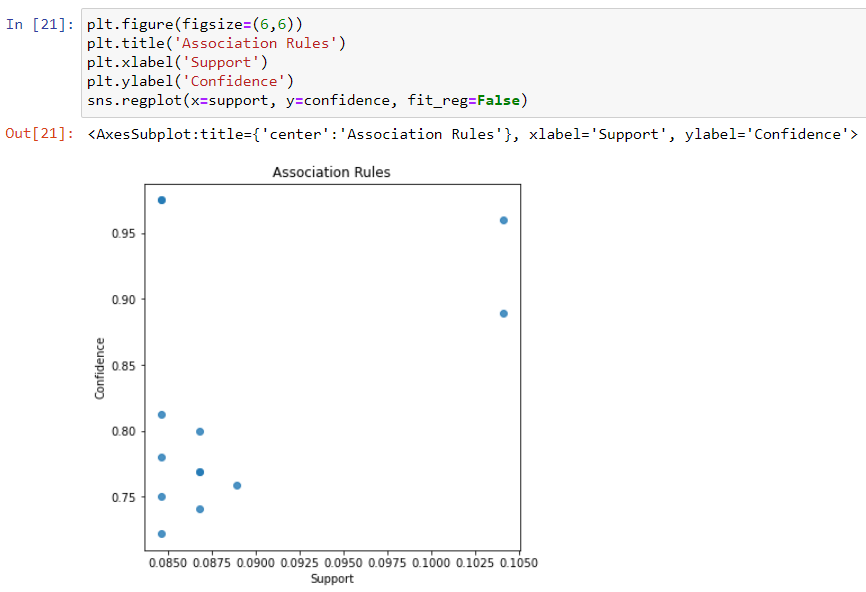
* Viết lệnh in ra các luật đã tìm được:



* Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật:



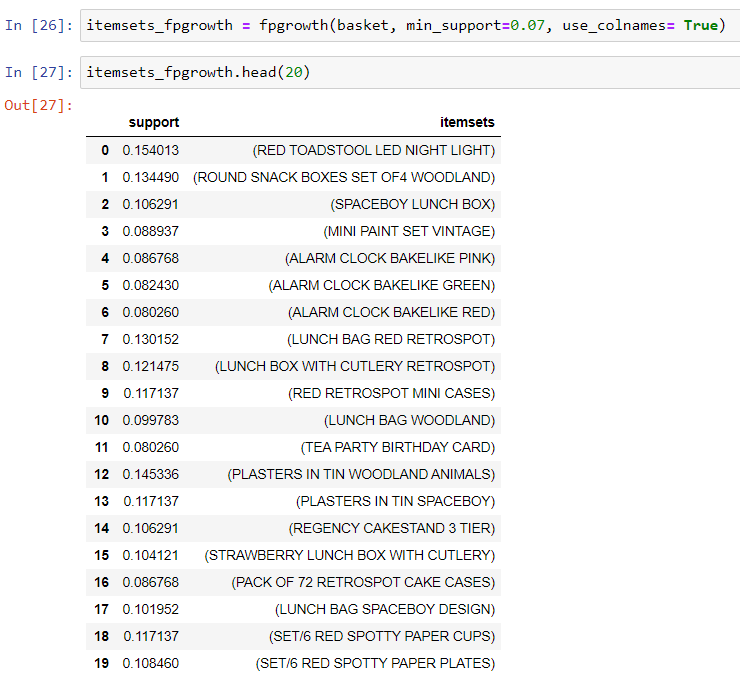
* Biểu diễn các thông tin này lên biểu đồ:



* Import module fpgrowth từ thư viện mlxtend và thực hiện tìm tập phổ biến bằng thuật toán FP-Growth:



* Áp dụng thuật toán FP-Growth với min\_sup = 7% để tìm tập phổ biến:



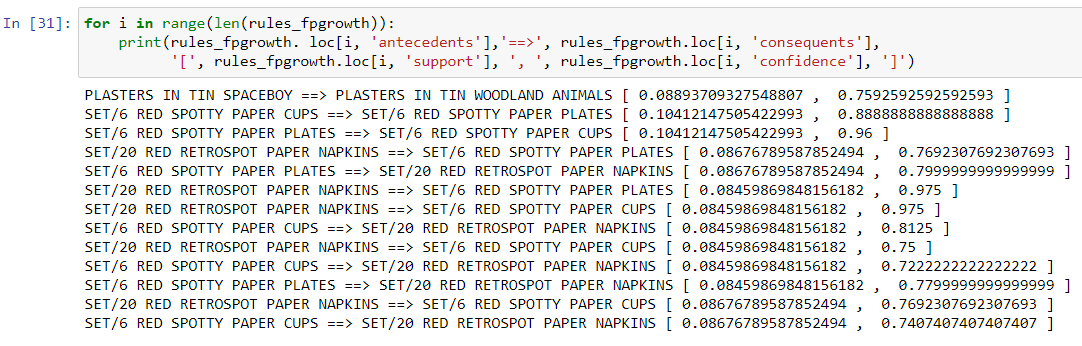
* Tạo luật kết hợp với min\_conf = 70%:



* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode):

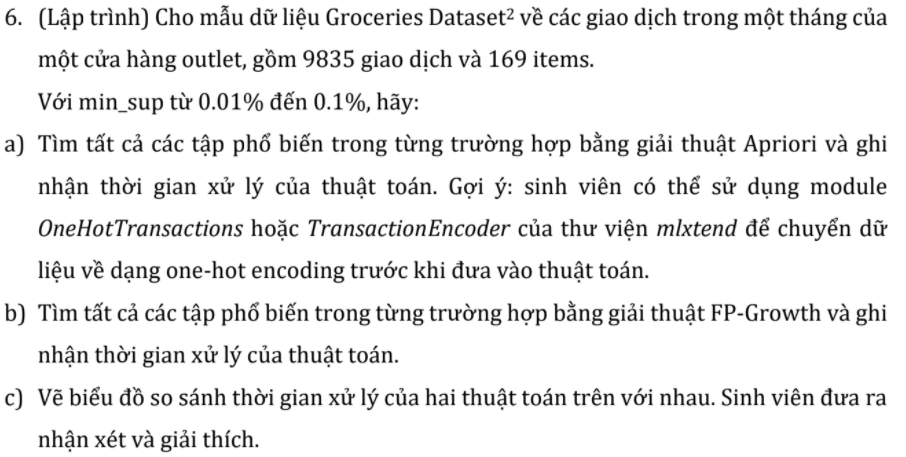


* Viết lệnh in ra các luật đã tìm được:

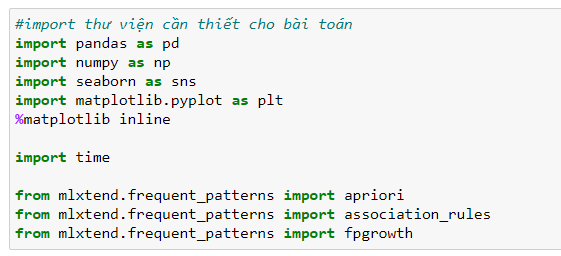


### Bài tập 6:

* Yêu cầu:



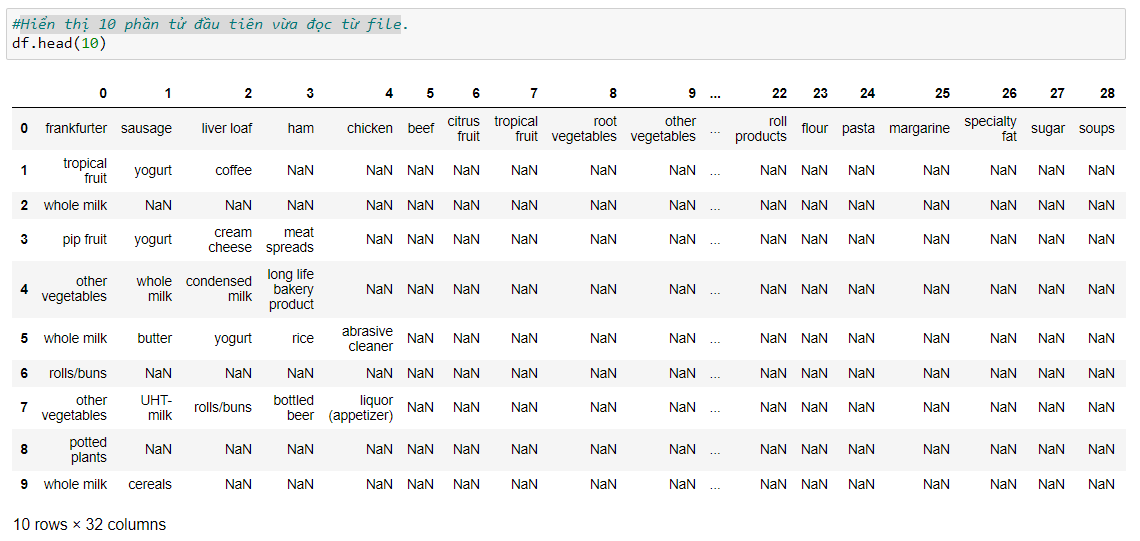
* Giải quyết:
* import thư viện cần thiết cho bài toán:



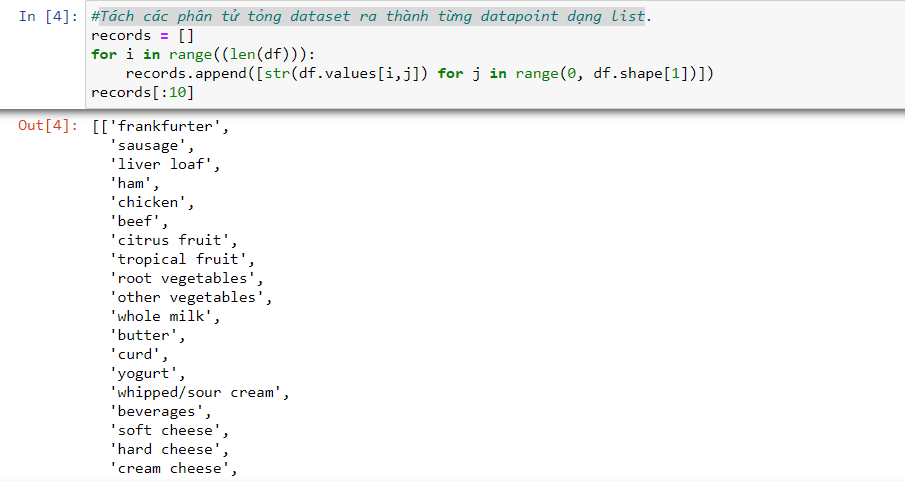
* Đọc dữ liệu từ file csv:



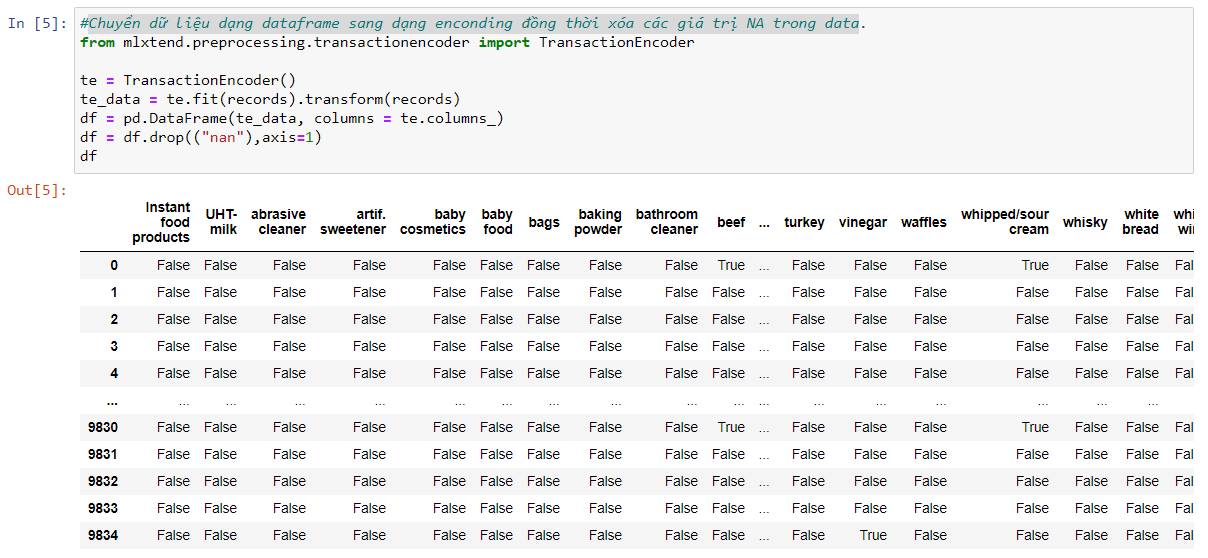
* Hiển thị 10 phần tử đầu tiên vừa đọc từ file:



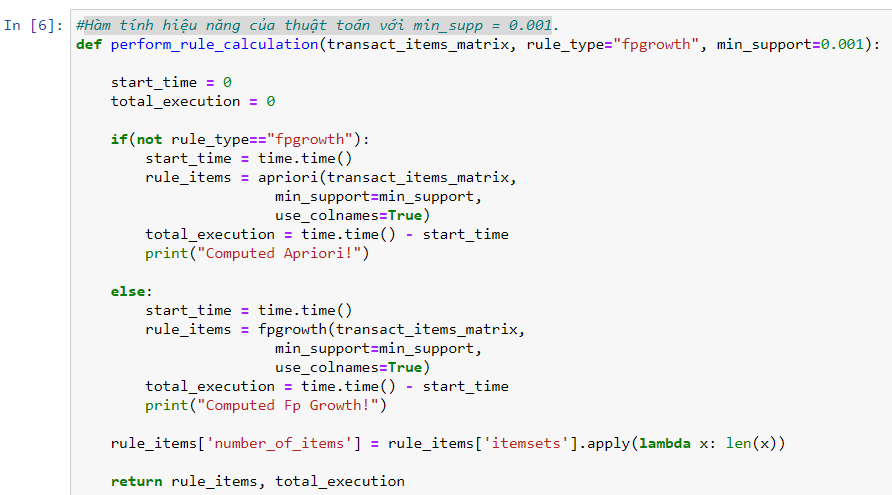
* Tách các phân tử tỏng dataset ra thành từng datapoint dạng list:



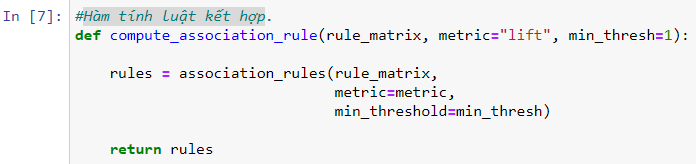
* Chuyển dữ liệu dạng dataframe sang dạng enconding đồng thời xóa các giá trị NA trong data:



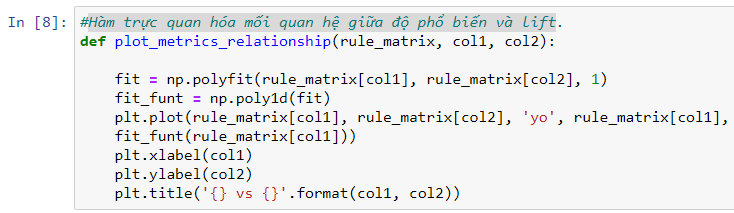
* Hàm tính hiệu năng của thuật toán với min\_supp = 0.001:



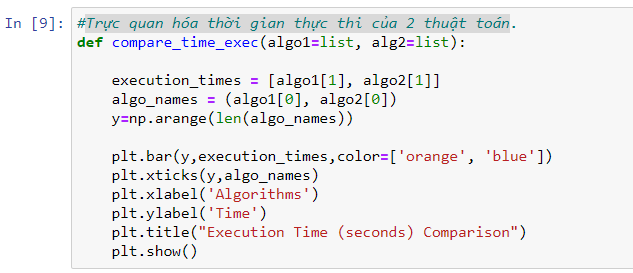
* Hàm tính luật kết hợp:



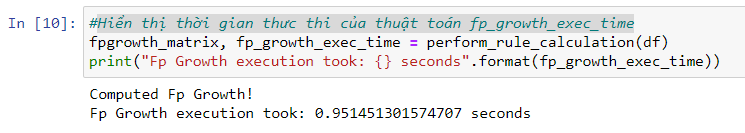
* Hàm trực quan hóa mối quan hệ giữa độ phổ biến và lift:



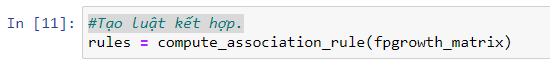
* Trực quan hóa thời gian thực thi của 2 thuật toán:



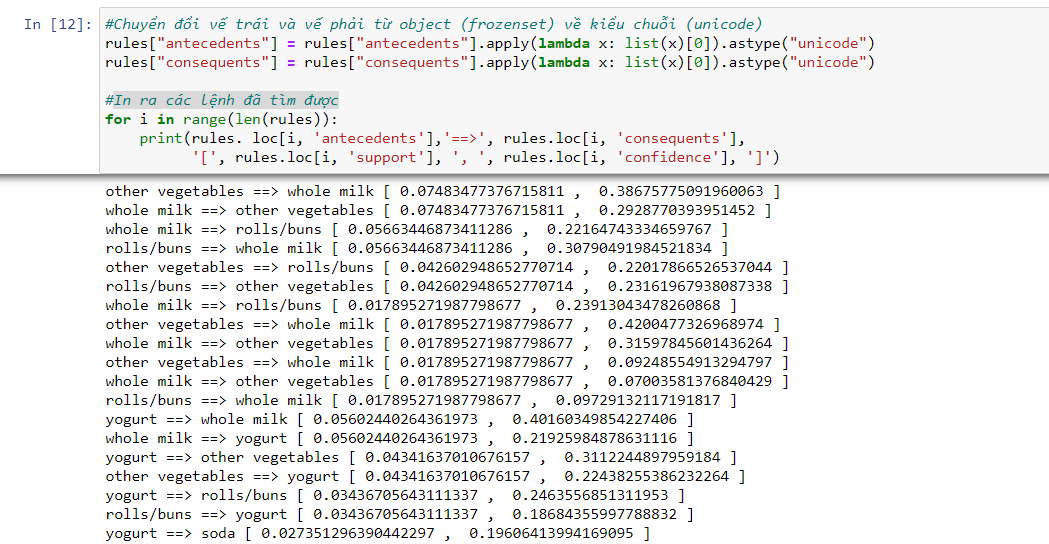
* Hiển thị thời gian thưc thi của thuật toán fp\_growth\_exec\_time:



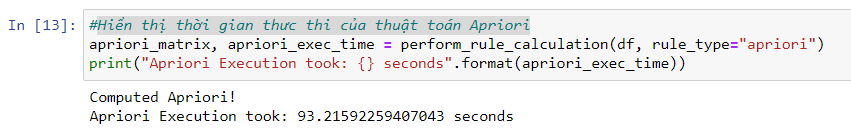
* Tạo luật kết hợp:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode), In ra các lệnh đã tìm được:



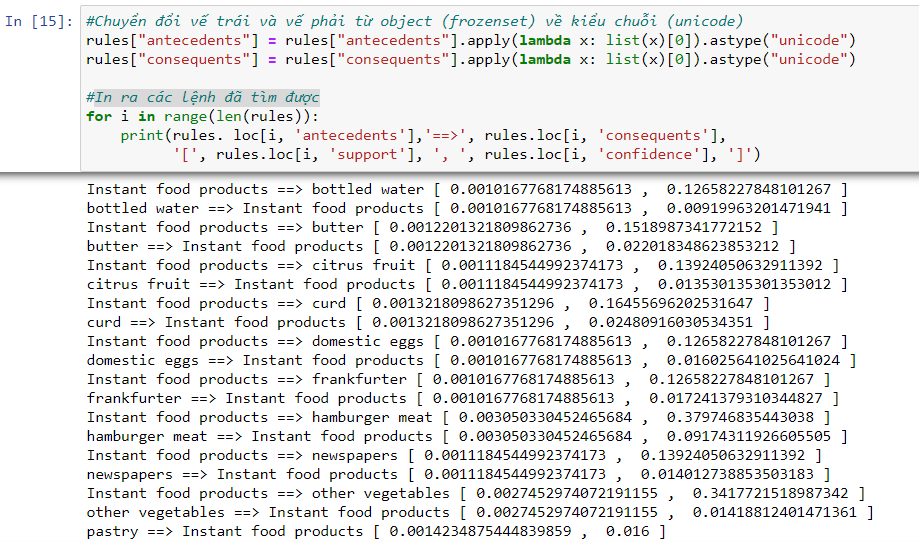
* Hiển thị thời gian thưc thi của thuật toán Apriori:



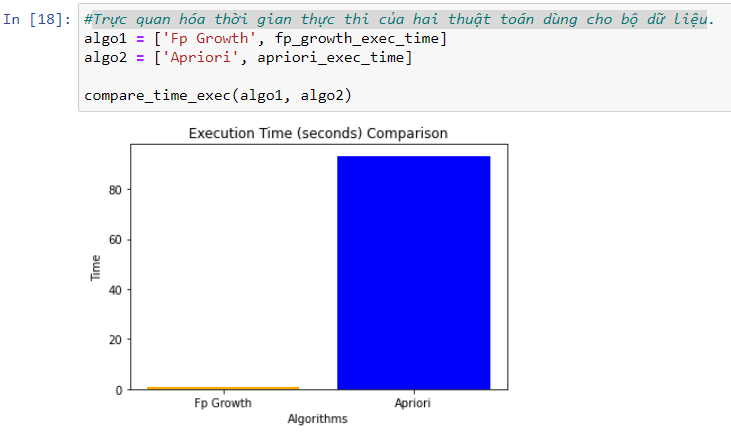
* Tạo luật kết hợp:



* Chuyển đổi vế trái và vế phải từ object (frozenset) về kiểu chuỗi (unicode), In ra các lệnh đã tìm được:



* Trực quan hóa thời gian thực thi của hai thuật toán dùng cho bộ dữ liệu:



* Nhận xét:
* Ta thấy thời gian thực thi của 2 thuật toán chênh lệch nhau khá lớn.
* Thuật toán Fp-Growth có thời gian thực thi chưa đầy 1 giây nhanh gấp 93 lần so với thuật toán Apriori. Vì vậy cho thấy sự phù hợp của thuật toán Fp-Growth đối với bộ dữ liệu của bài toán này.
* Hạn chế:
* Apriori: kéo nhiều Mục ứng viên; quá nhiều lần quét cơ sở dữ liệu; yêu cầu không gian bộ nhớ lớn.
* FP - Growth: FP-Tree tốn kém để xây dựng sẽ tiêu tốn nhiều bộ nhớ hơn.
* Giải thích:

|  |  |  |
| --- | --- | --- |
|  | Apriori | Fp\_Growth |
| Kỹ thuật | Tìm kiếm phần tử đâu tiên theo chiều rộng và theo thuộc tính của Apriori | Tìm kiếm theo phân nhóm(dưới hình thức cây). |
| Quét cơ sở dữ liệu | Cơ sở dữ liệu được quét cho mỗi lần tập hợp mục ứng viên được tạo. | Cơ sở dữ liệu chỉ được quét hai lần. |