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

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**--------------------oOo--------------------**

**KHAI THÁC DỮ LIỆU**

**(Data Mining)**

**LAB 2**

Giảng viên hướng dẫn:

**ThS. Lê Hà Hoài Trung**

Sinh viên thực hiện:

**Hoàng Lê Nam – MSSV: 18521120**

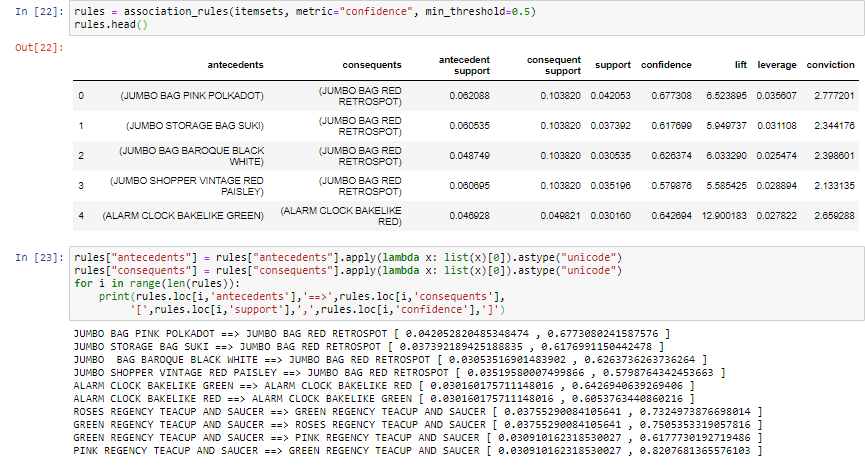
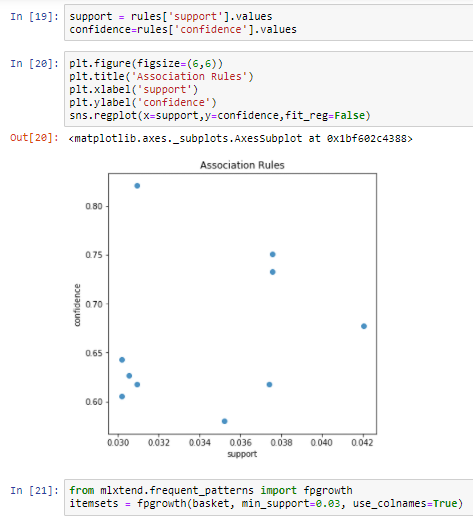
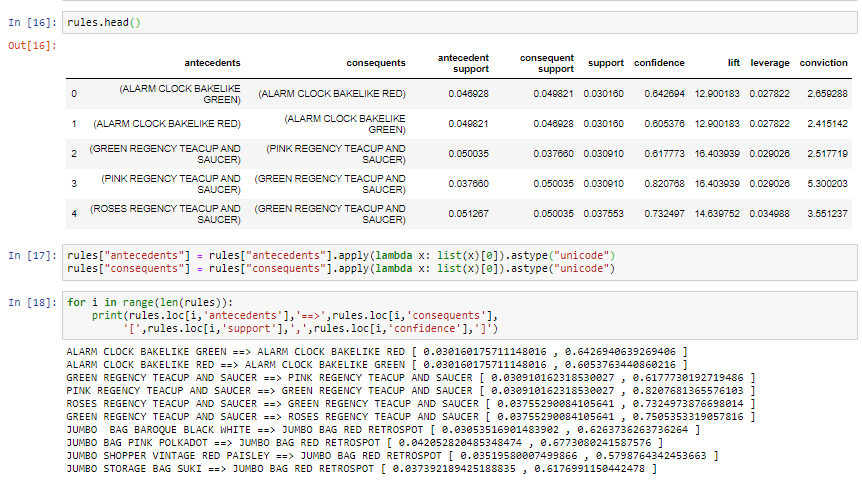
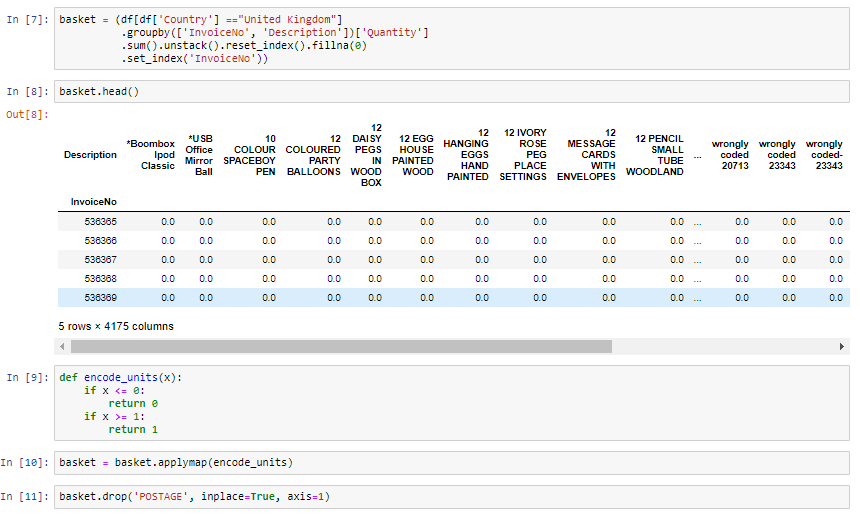
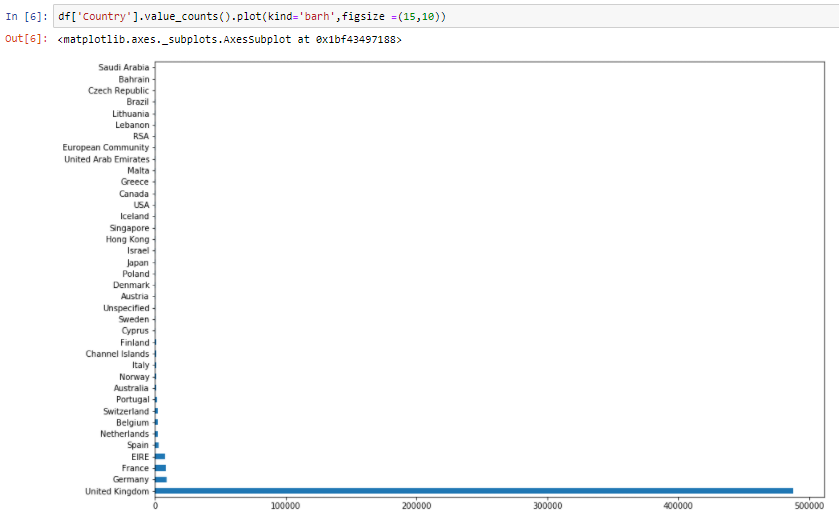
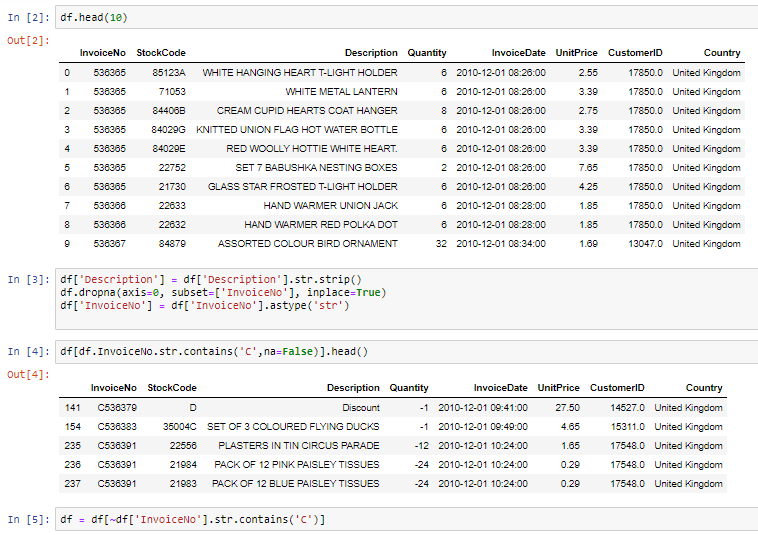
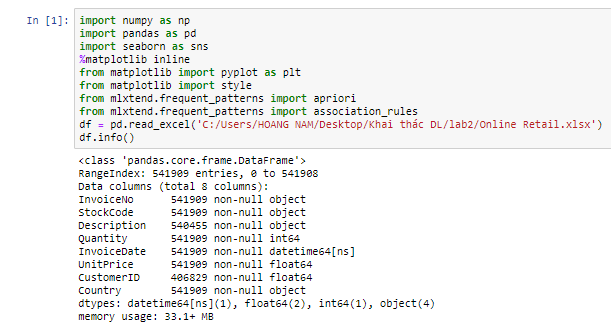
🙡🙢 Tp. Hồ Chí Minh, 3/2021 🙠🙣

# Hướng dẫn thực hành

* Hướng dẫn thực hiện theo thứ tự sau
* import thư viện cần sử dụng cho bài toán In[1]
* Đọc file dữ liệu bằng pandas In [1]
* Hiển thị thông tin của dữ liệu vừa được đọc In[2]
* 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 In[3]
* Hiển thị 10 phần tử đầu hóa đơn tính dụng In[4]
* Tiến hành xóa những hóa đơn tín dụng In[5]
* Thống kê số dòng dữ liệu theo từng quốc gia In[6]
* Chỉ xét các hóa đơn từ nước Anh và nhóm dữ liệu theo Số hóa đơn và Tên mặt hàng In[7]
* Chuyển đổi dữ liệu về dạng hot encoding, với mỗi dòng dữ liệu là một hóa đơn In [7]
* Xem dữ liệu sau khi chuyển về dạng hot encoding In[8]
* Tạo hàm biến đổi mỗi điểm dữ liệu có số lượng (Quantity) lớn hơn 0 thành 1 In[9]
* Chuyển đổi dữ liệu từ dạng hot encoding thành one-hot encoding In[10]
* Xóa cột ‘POSTAGE’ In[11]
* Áp dụng thuật toán Apriori với min\_sup = 3% để tìm tập phổ biến In[13]
* Xem 10 phần tử đầu tiên trong tập phổ biến tìm được In[14]
* Tạo luật kết hợp với min\_conf = 50% In[15]
* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode) In[17]
* Viết lệnh in ra các luật đã tìm được In[18]
* Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật In[19]
* Biểu diễn các thông tin này lên biểu đồ In[20]
* Import module fpgrowth từ thư viện mlxtend và thực hiện tìm tập phổ biến bằng thuật toán FP-Growth In[21]
* Áp dụng thuật toán FP-Growth với min\_sup = 3% để tìm tập phổ biến In[22]
* Tạo luật kết hợp với min\_conf = 50% In[23]
* Chuyển đổi vế trái và vế phải từ kiểu object (frozenset) về kiểu chuỗi (unicode)

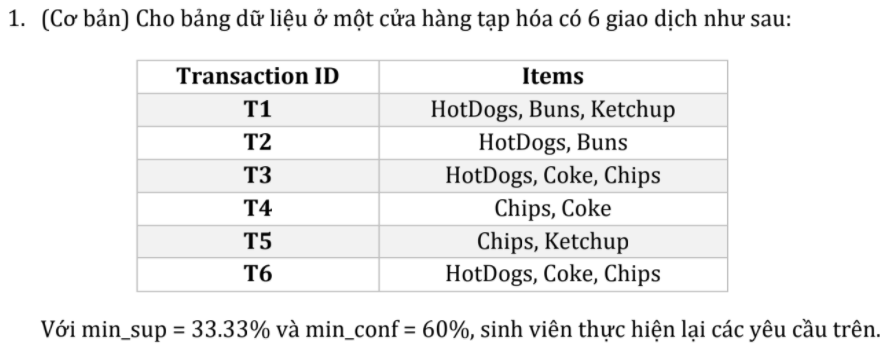
In[23]

* Viết lệnh in ra các luật đã tìm được In[23]

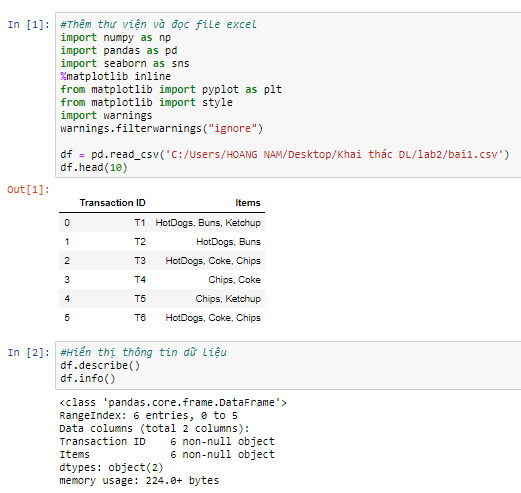
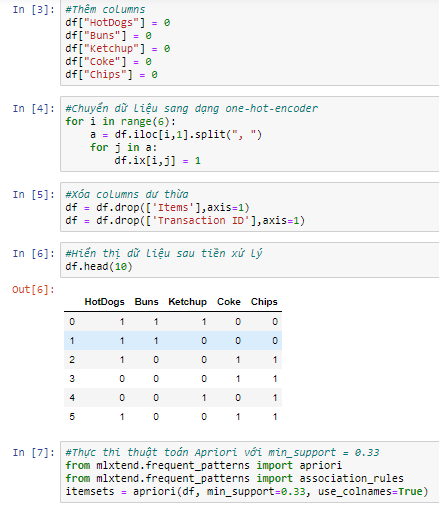
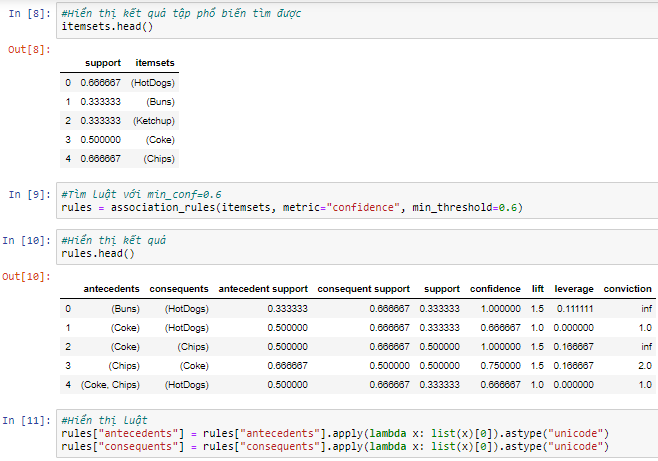
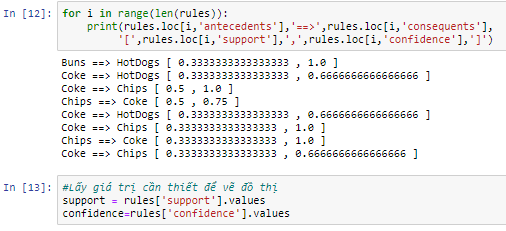
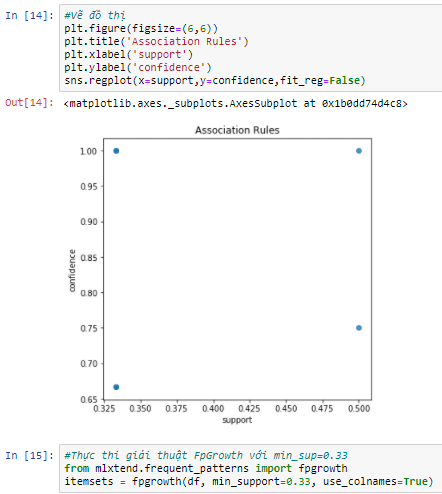
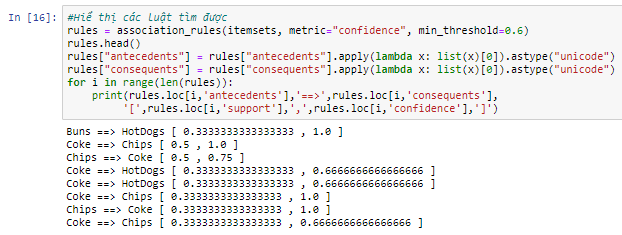


# Bài 1

* Yêu cầu:

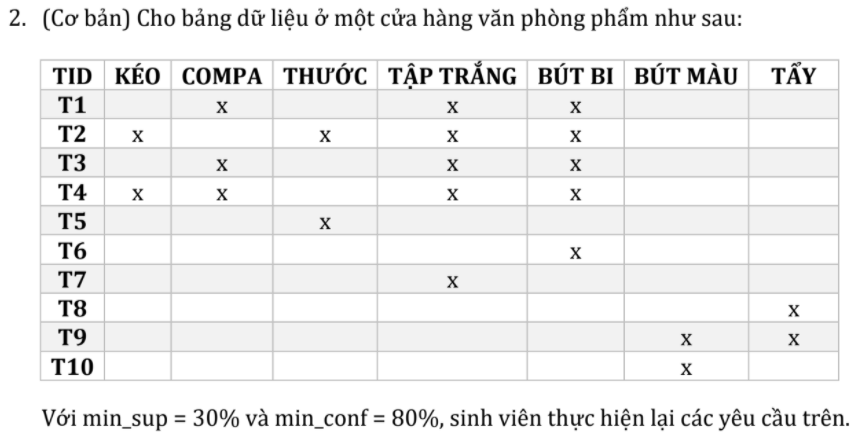


* Giải quyết

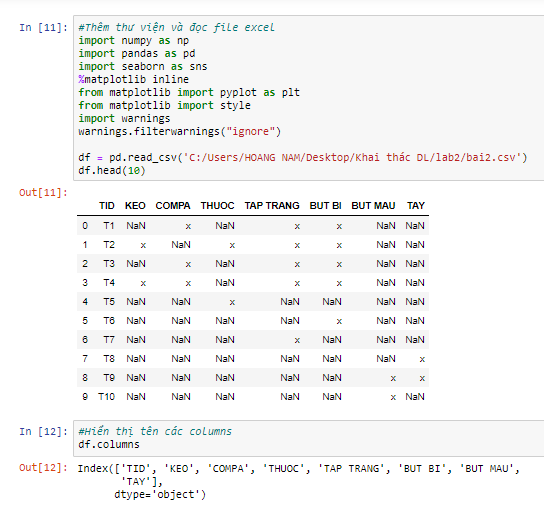
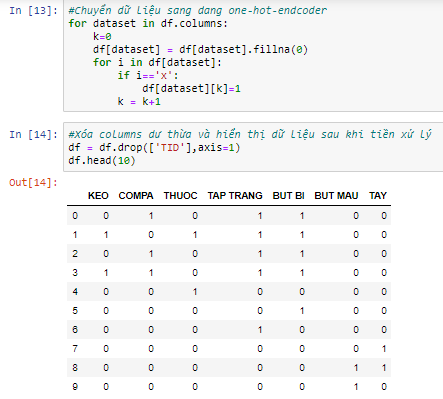
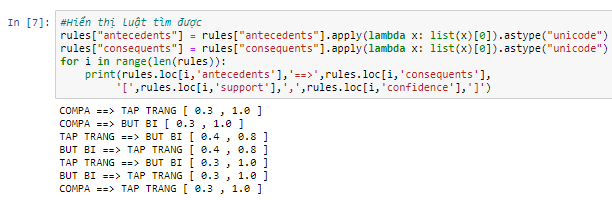
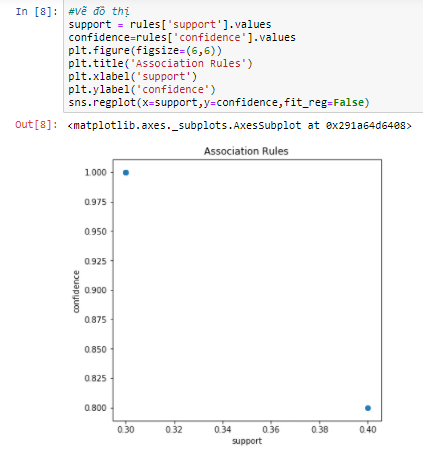
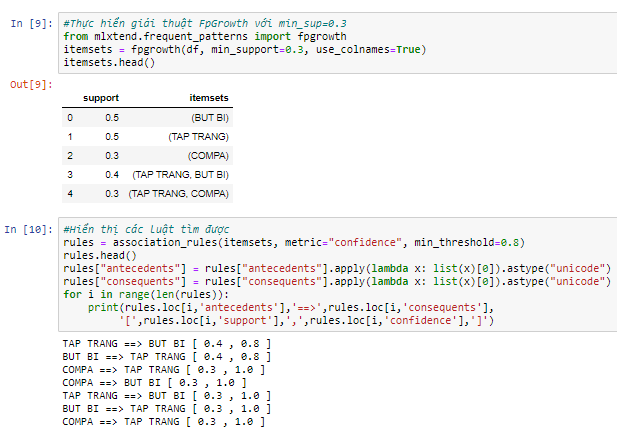
     

# Bài 2

* Yêu cầu:



* Giải quyết:

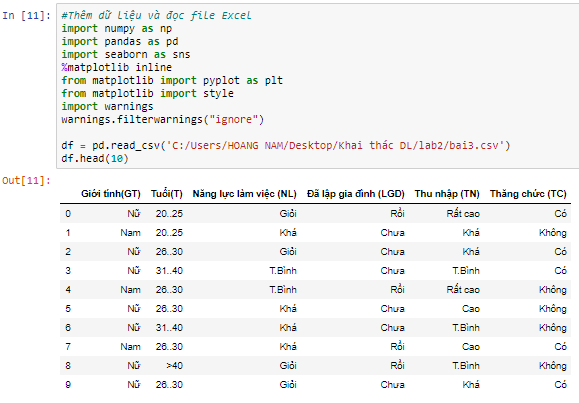
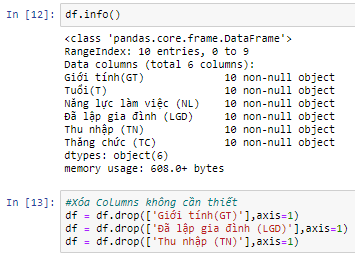
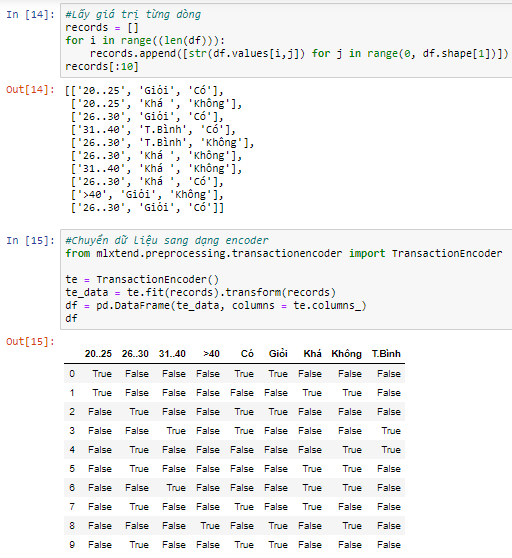
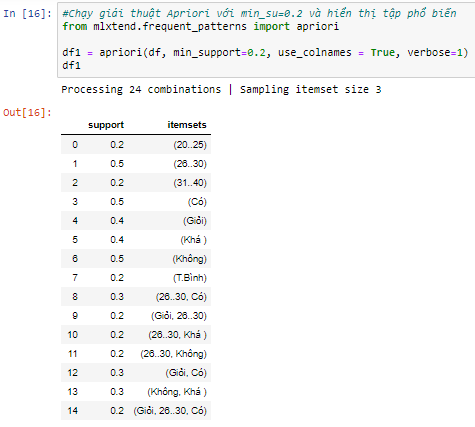
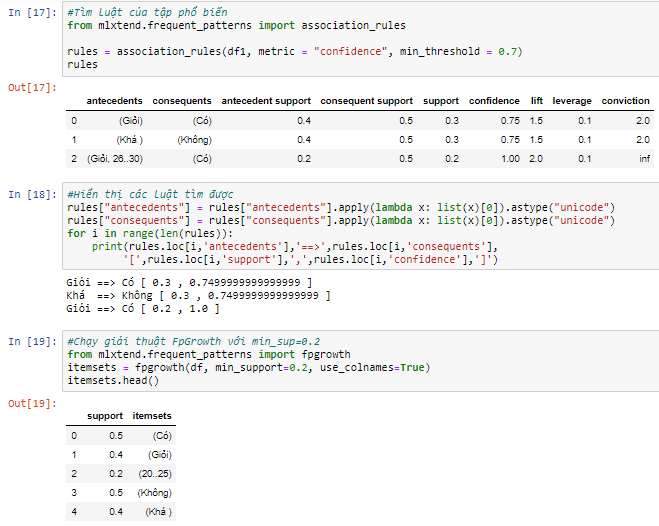
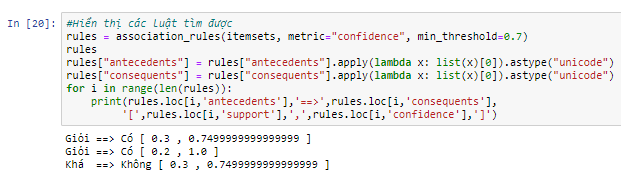
     

# Bài 3

* Yêu cầu:

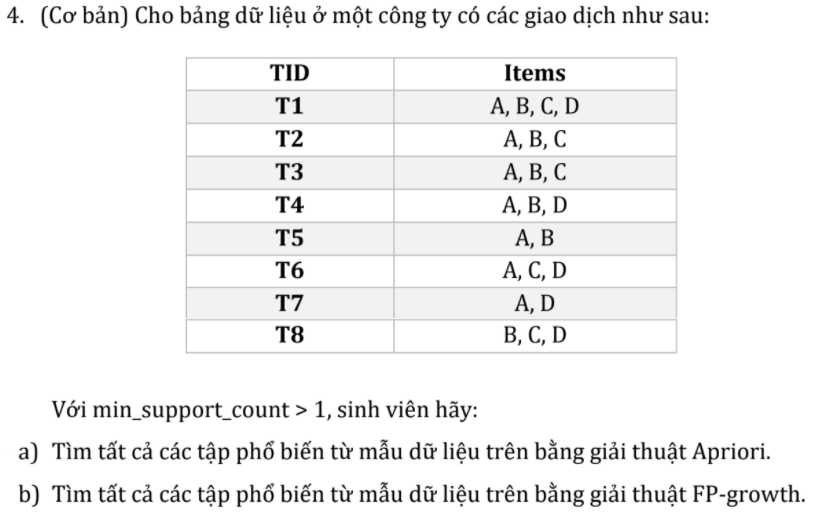


* Giải quyết:

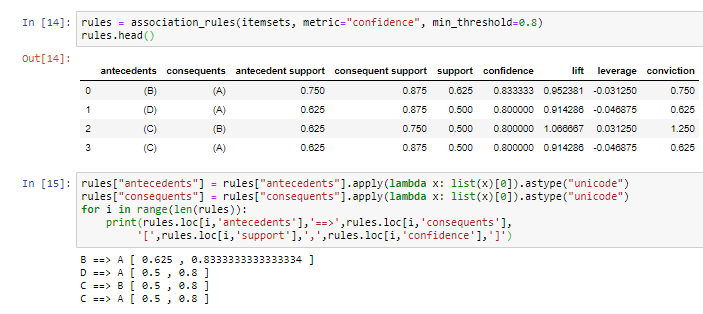
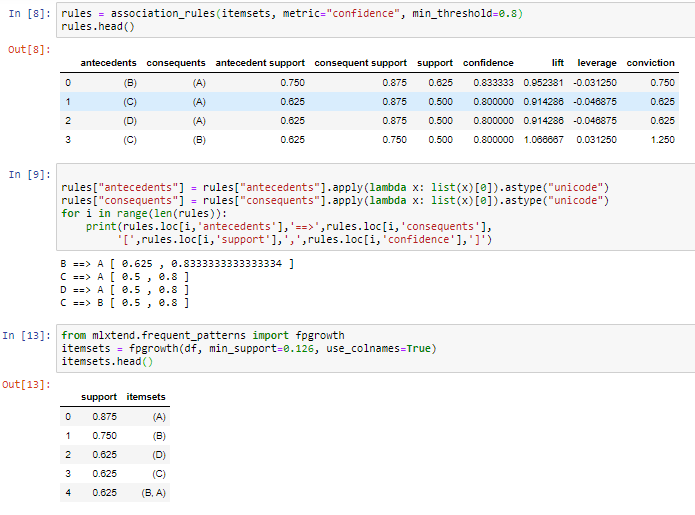
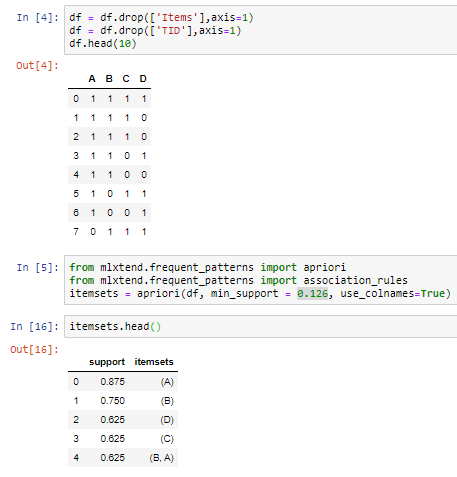
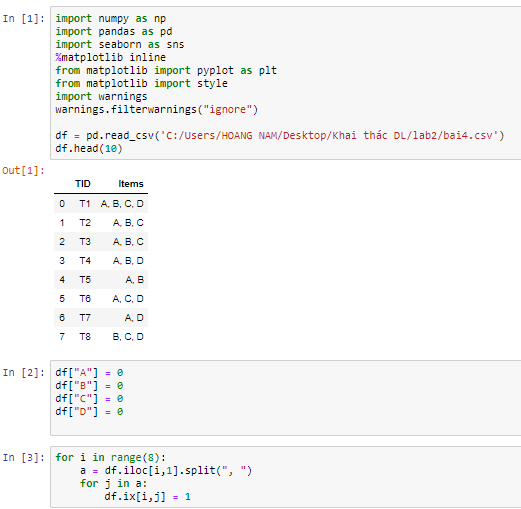
     

# Bài 4

* Yêu cầu:

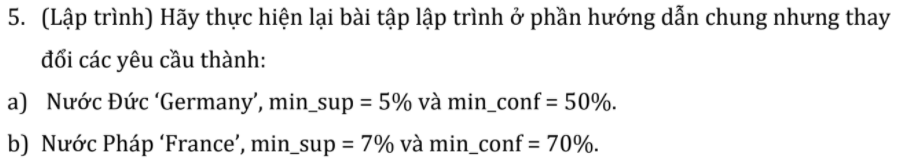


* Giải quyết:



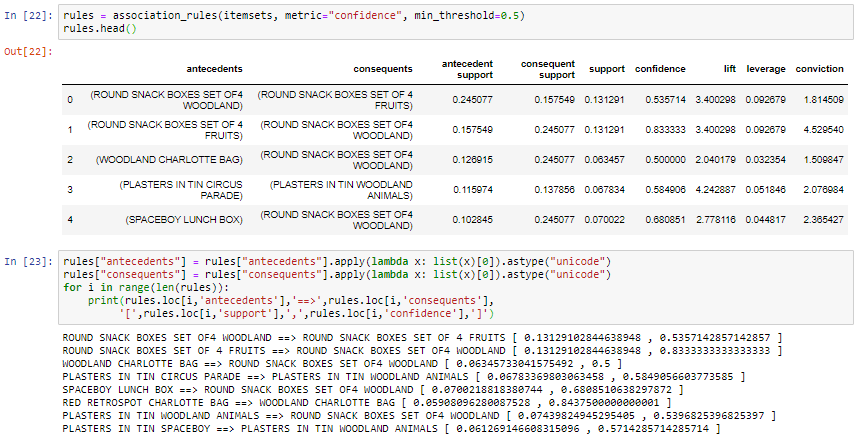
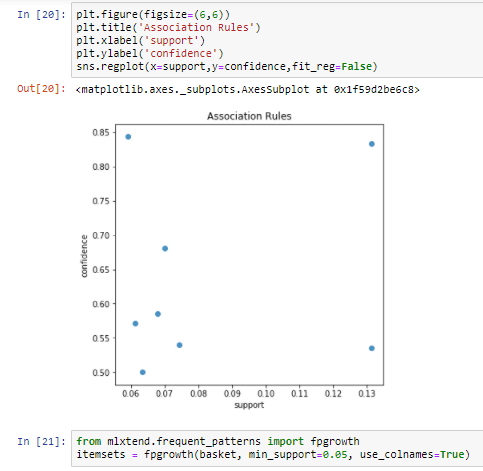
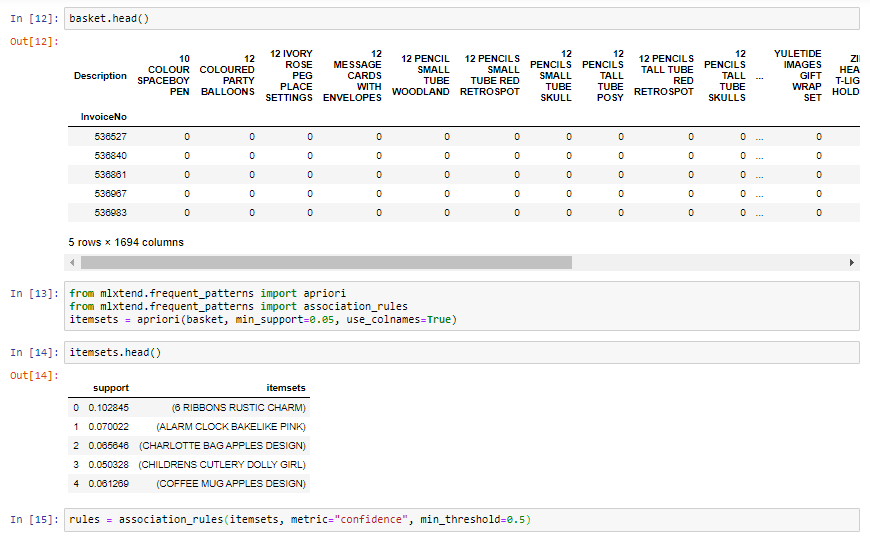
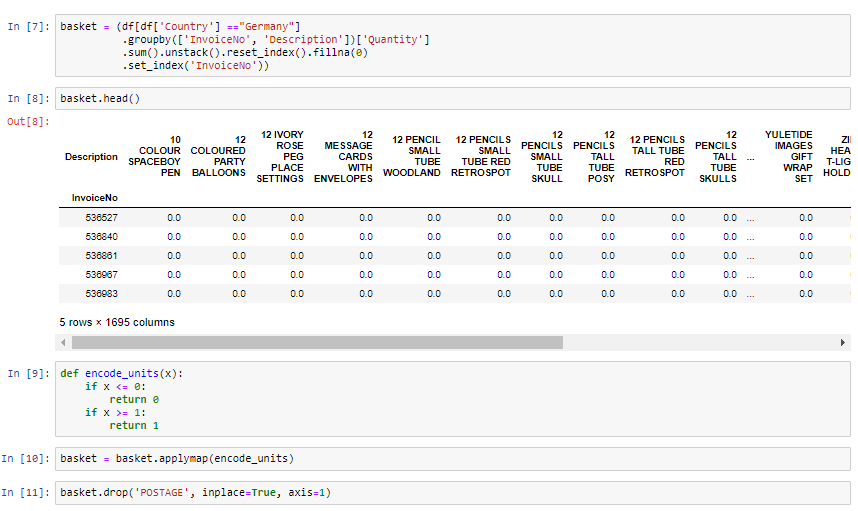
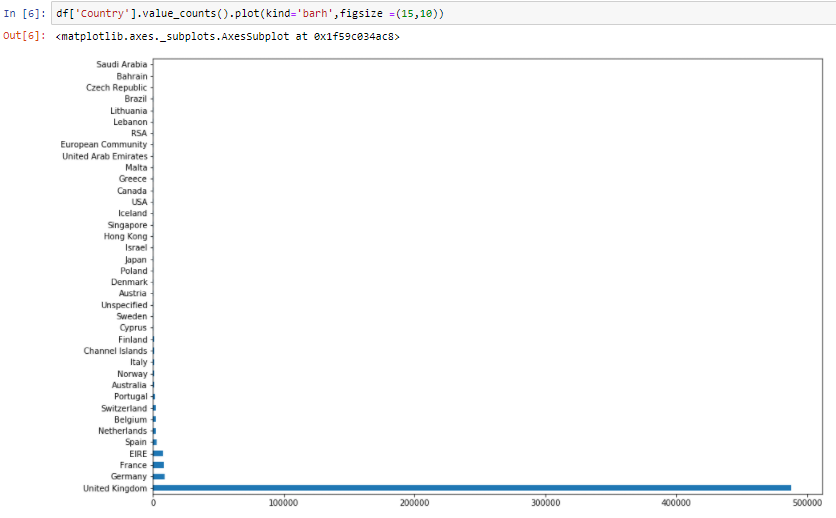
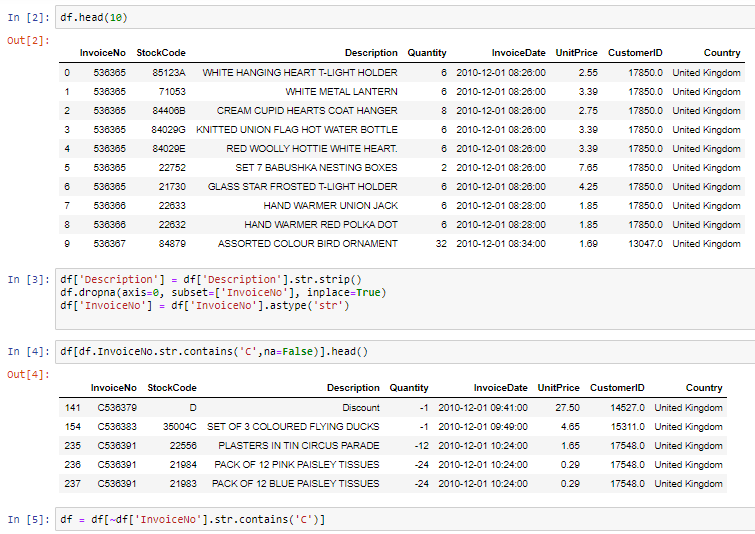
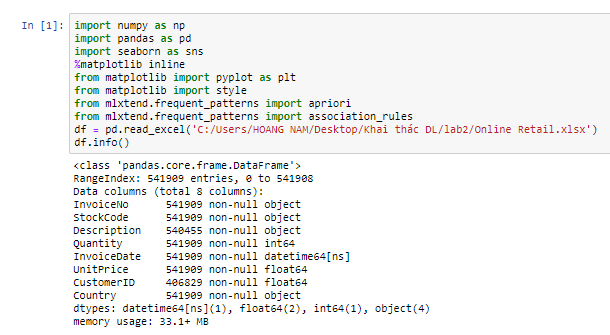
# Bài 5

* Yêu cầu:

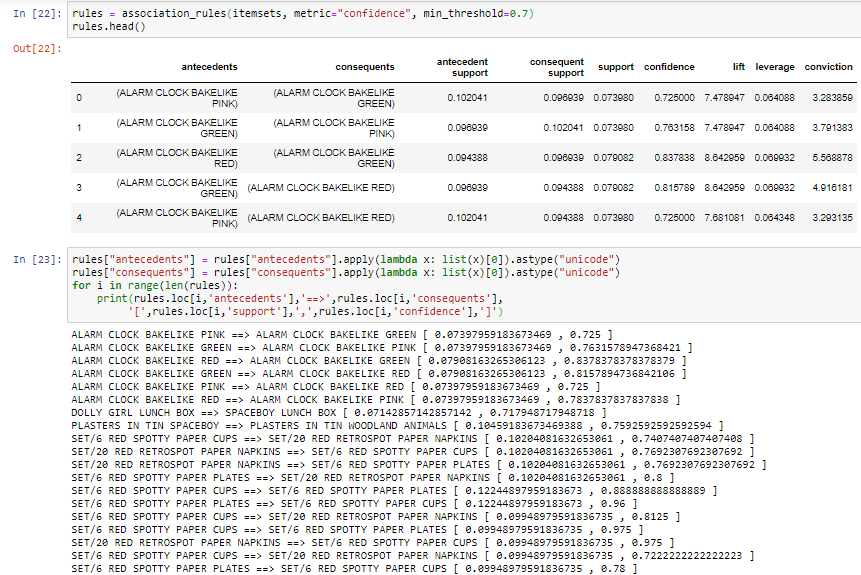
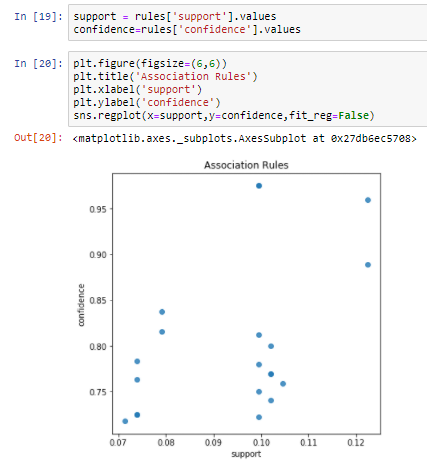
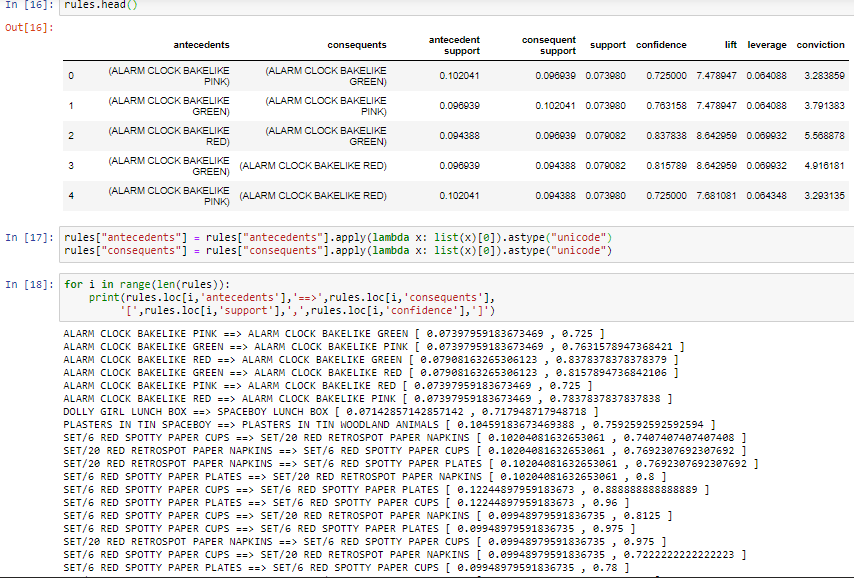
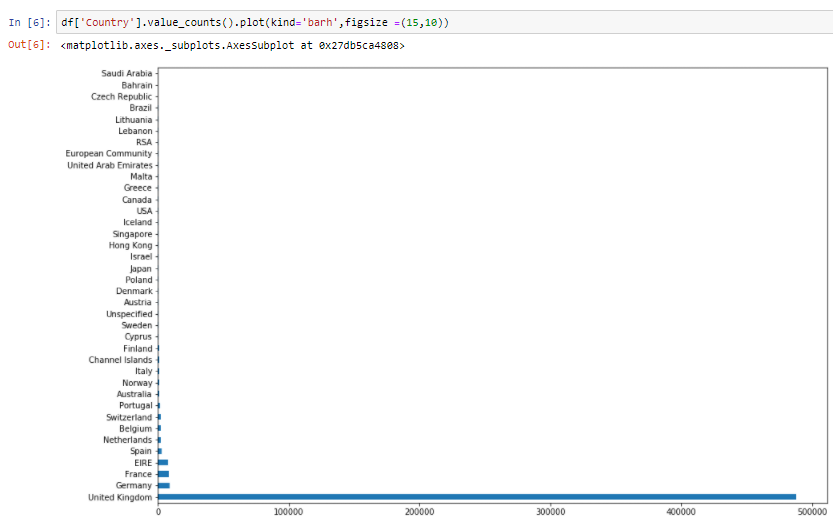
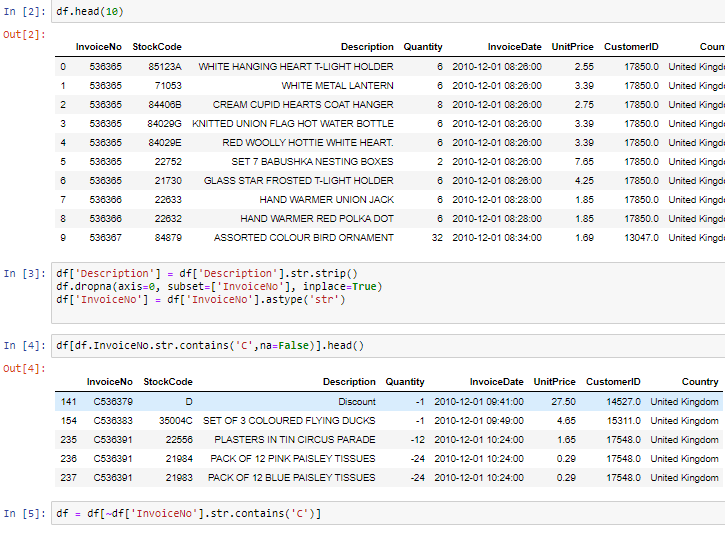
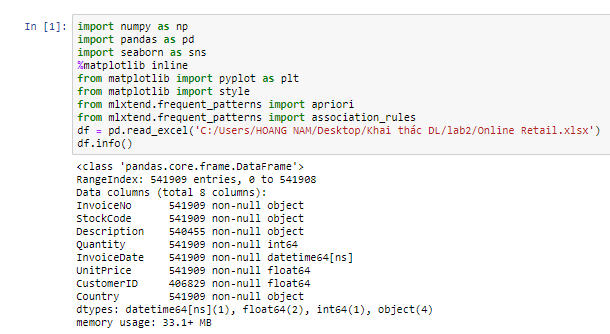


* Giải quyết:

## **Germany**

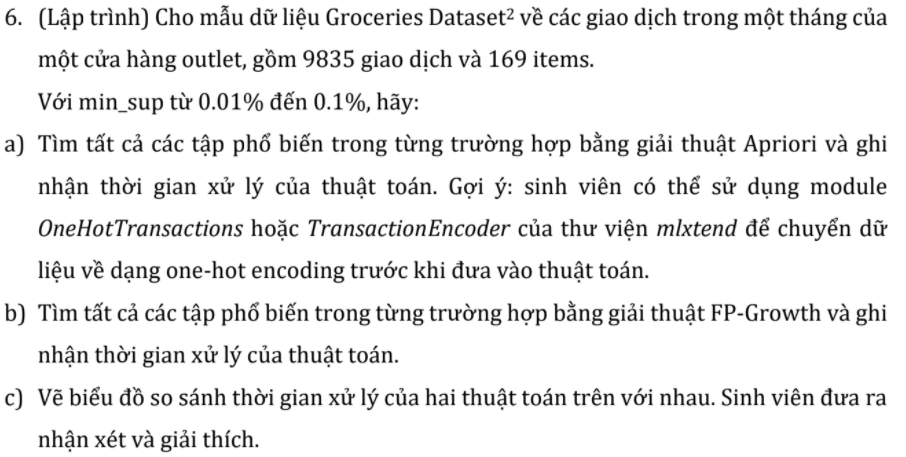


## **France**

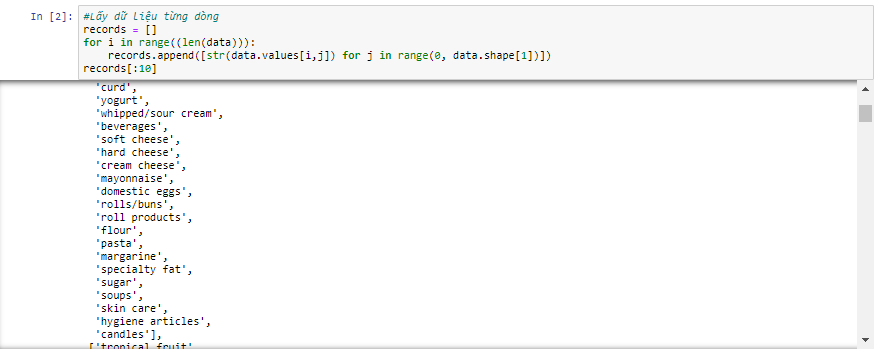
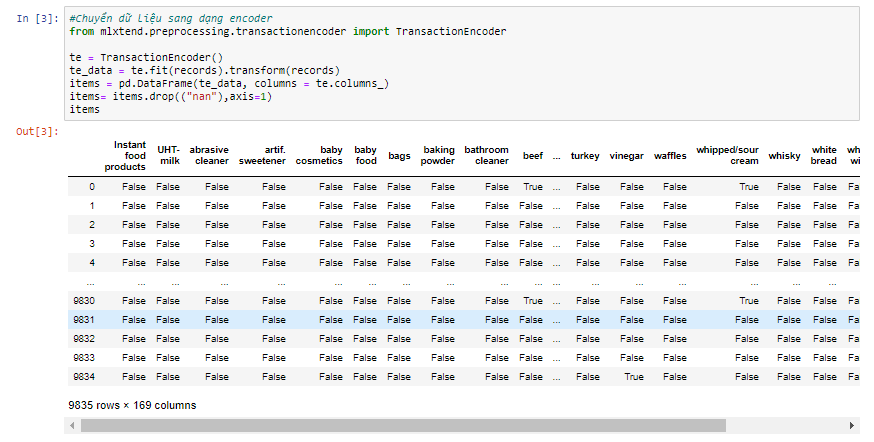
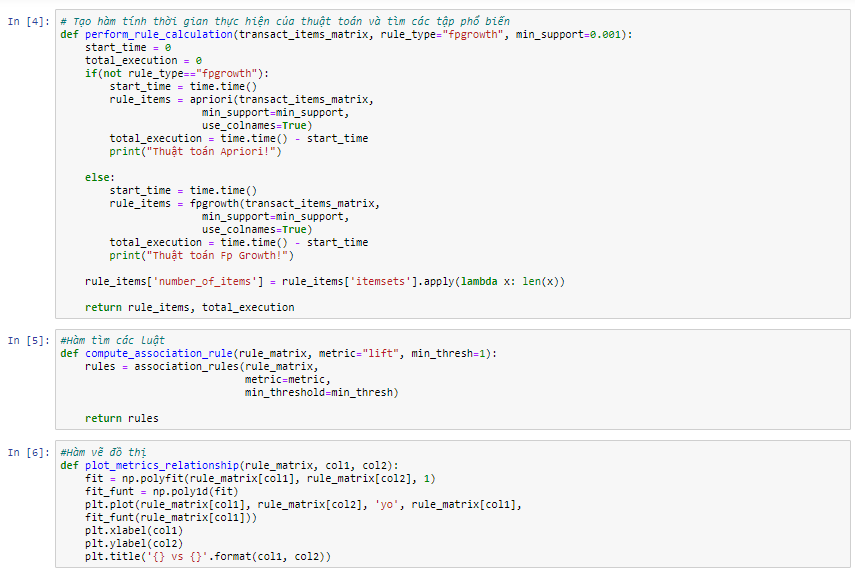
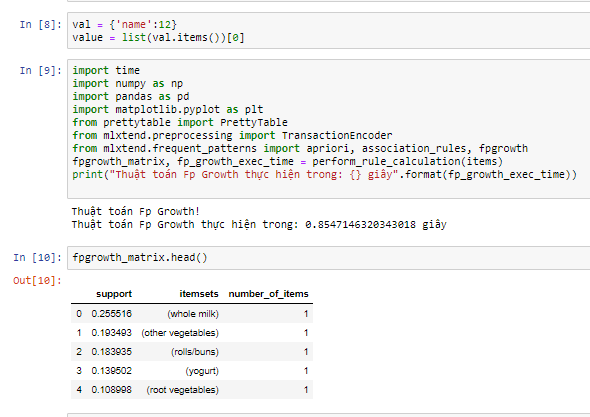


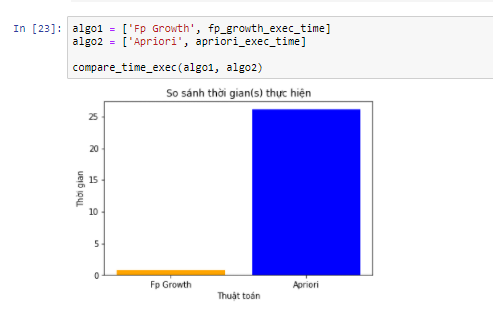
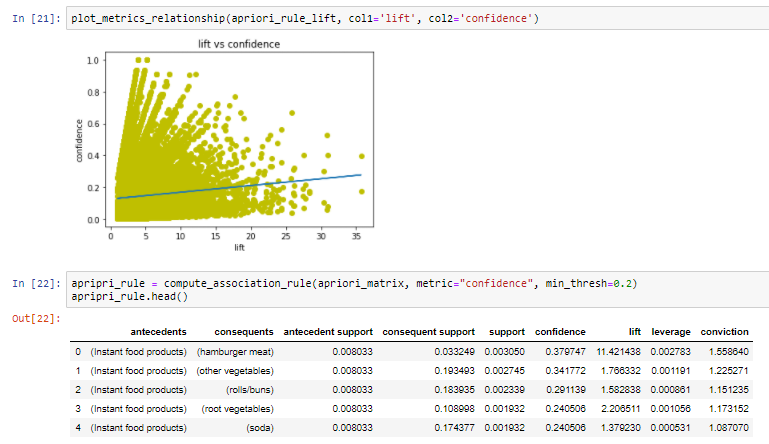
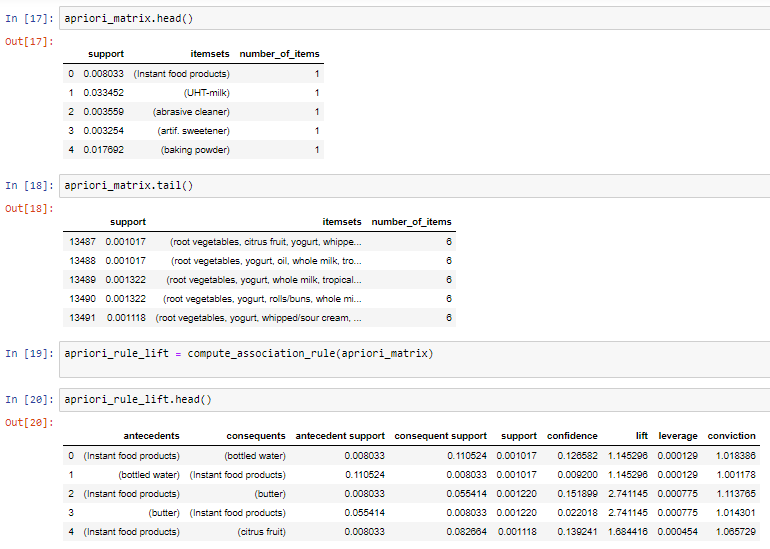
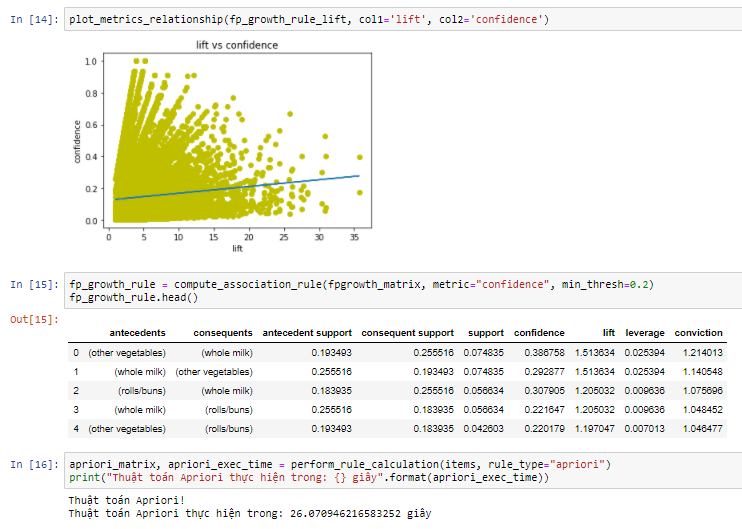
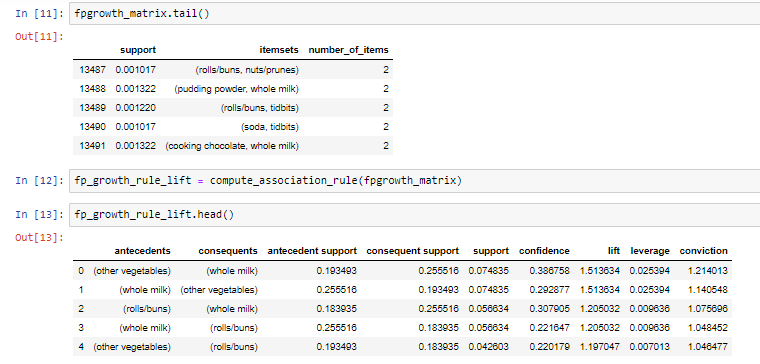
# Bài 6

* Yêu cầu:



* Giải quyết:



* 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. |