TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH
Môn: Khai thác dữ liệu

BÀI THỰC HÀNH SỐ 2

I. Mục tiêu

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.

II. Thời gian

1. Thực hành: 4 tiết
2. Bài tập làm thêm: 8 tiết

III. Hướng dẫn chung

1. Yêu cầu cơ bản

Cho bảng dữ liệu có 6 giao dịch như sau:

Transaction ID	Items
T001	A, B, D, E
T002	B, C, E
T003	A, B, D, E
T004	A, B, C, E
T005	A, B, C, D, E
T006	B, C, D

Với min_sup = 50% và min_conf = 70%

1. Tìm tất cả các tập phổ biến từ mẫu dữ liệu trên bằng giải thuật Apriori.
2. Tìm tất cả các tập phổ biến từ mẫu dữ liệu trên bằng giải thuật FP-growth.
3. Tìm tất cả các luật kết hợp dựa trên các tập phổ biến tìm được ở câu 1.

Hướng dẫn:

1. Tìm tập phổ biến bằng giải thuật Apriori:

- Với min_sup là 50% và tổng số giao dịch là 6 => tần số xuất hiện tối thiểu của phần tử để thỏa min_sup (min_support_count) là 3.
- Tập các ứng viên 1 phần tử và tần số xuất hiện của phần tử (support count) tương ứng:

C_1 = {A: 4, B: 6, C: 4, D: 4, E: 5}

⇨ Các tập phổ biến 1 phần tử (tập các ứng viên thỏa mãn min_support_count):

L_1 = {{A}, {B}, {C}, {D}, {E}}

- Tập các ứng viên có 2 phần tử và suport count tương ứng:

$C_2 = L_1 \bowtie L_1$ = {{A, B}: 4, {A, C}: 2, {A, D}: 3, {A, E}: 4, {B, C}: 4, {B, D}: 4, {B, E}: 5, {C, D}: 2, {C, E}: 3, {D, E}: 3}

⇨ Các tập phổ biến 2 phần tử thỏa min_support_count:

L_2 = {{A, B}, {A, D}, {A, E}, {B, C}, {B, D}, {B, E}, {C, E}, {D, E}}

Tương tự ta tìm các tập phổ biến với số lượng phần tử lớn hơn.

2. Tìm tập phổ biến bằng giải thuật FP-growth:

- Các bước chính
  - ✓ Bước 1: xây dựng cây FP (frequent pattern).
  - ✓ Bước 2: xây dựng cơ sở mẫu điều kiện (conditional pattern base) cho mỗi hạng mục phổ biến (mỗi nút trên cây FP).
  - ✓ Bước 3: xây dựng cây FP điều kiện (conditional FP tree) từ mỗi cơ sở mẫu điều kiện.
  - ✓ Bước 4: khai thác đệ quy cây FP điều kiện và phát triển mẫu phổ biến cho đến khi cây FP điều kiện chỉ chứa một đường dẫn duy nhất - tạo ra tất cả các tổ hợp của mẫu phổ biến.
- Bước 1: xây dựng cây FP
  - ✓ Tìm tập phổ biến 1 phần tử.
  - ✓ Sắp xếp danh sách giảm dần theo độ hỗ trợ (support) gọi đó là F-list.
  - ✓ Dựa theo F-list, xét từng dòng dữ liệu, bỏ hết các phần tử không có trong F-list và sắp xếp thứ tự các phần tử còn lại theo F-list.
  - ✓ Tiến hành xây cây:
    - Nút gốc của cây là *null*.
    - Mỗi nút gồm tên phần tử và số lần xuất hiện.
    - Lần lượt xét từng dòng dữ liệu để thêm các phần tử vào cây. Nếu phần tử là nút mới thì số lần xuất hiện là 1, nếu phần tử đã có trong cây thì số lần xuất hiện được cộng thêm 1.
    - Dựa vào F-list tạo bảng Header, nếu phần tử xuất hiện lần đầu trong cây thì tạo một con trỏ từ bảng Header chỉ tới nó. Nếu phần tử đã xuất hiện trước đó ở

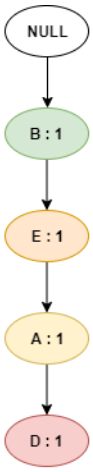
nhánh khác (đã có con trỏ từ bảng Header chỉ tới) thì tạo một con trỏ từ phần tử trước đó chỉ tới nó.

- Lặp lại với tất cả các dòng.

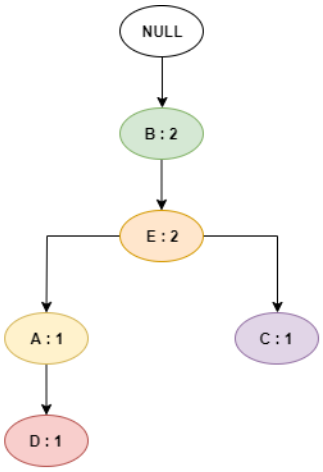
Bảng F-list

Item	Support count
B	6
E	5
A	4
C	4
D	4

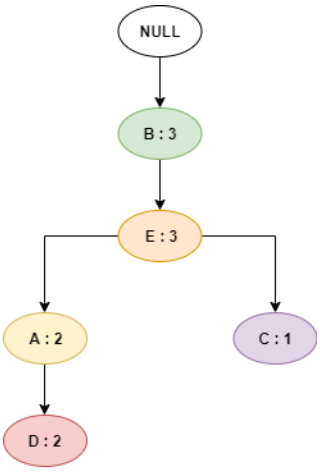
Thêm giao dịch 1 vào cây FP



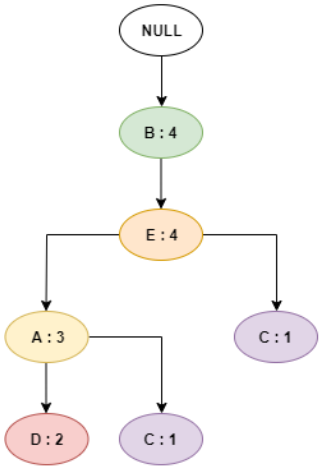
Thêm giao dịch 2 vào cây FP



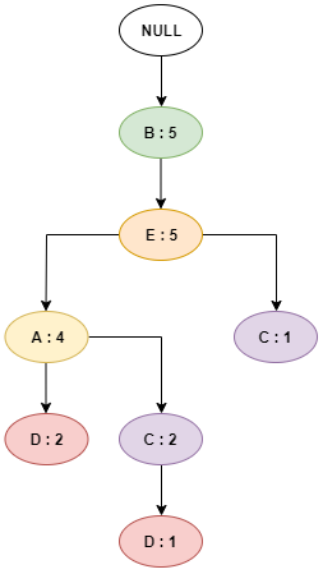
Thêm giao dịch 3 vào cây FP



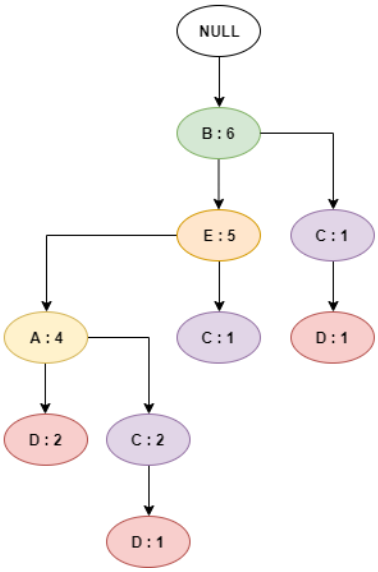
Thêm giao dịch 4 vào cây FP



Thêm giao dịch 5 vào cây FP

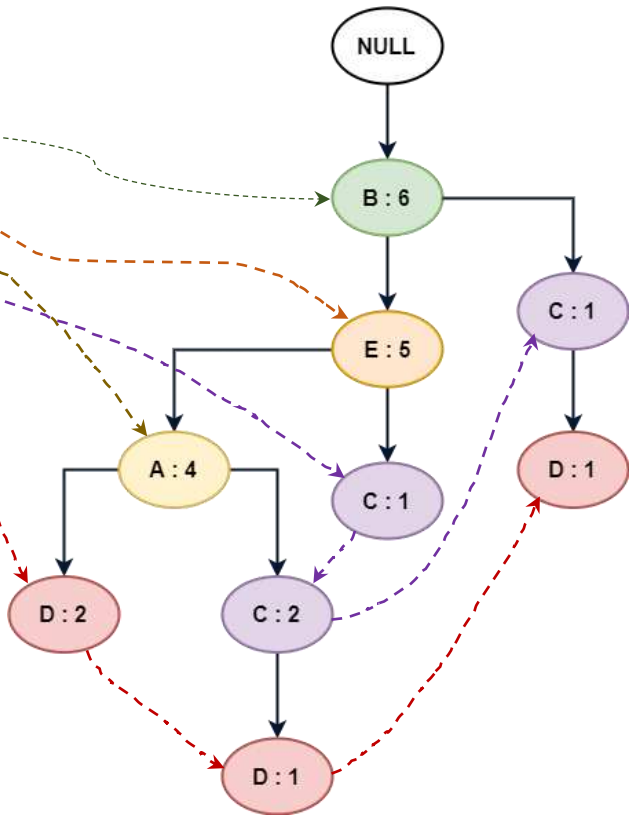


Thêm giao dịch 6 vào cây FP



Bảng Header

Item	Support count	Link
B	6	
E	5	
A	4	
C	4	
D	4	



- Bước 2: xây dựng cơ sở mẫu điều kiện
  - ✓ Xét lần lượt từng phần tử trong bảng Header theo thứ tự từ dưới lên (từ phần tử có độ hỗ trợ thấp nhất).
  - ✓ Với mỗi phần tử lấy ra:
    - Lần theo các con trỏ để tìm ra các nhánh có chứa phần tử đó.
    - Chỉ xét từ vị trí phần tử đó trở lên.
    - Lấy ra tập phần tử phía trên phần tử đó, kèm theo tần suất của chính phần tử đó. Chúng ta xem giá trị này là tần suất xuất hiện của tập phần tử phía trên phần tử đang xét.
    - Lần lượt duyệt hết bảng Header, ghi nhận từng phần tử và cơ sở mẫu điều kiện tương ứng với nó có kèm theo tần suất. Phần tử đầu tiên sẽ có cơ sở mẫu điều kiện là rỗng *null*.

Cơ sở mẫu điều kiện

Item	Conditional pattern base	Support count
D	B, E, A	2
	B, E, A, C	1
	B, C	1
C	B, E	1
	B, E, A	2
	B	1
A	B, E	4
E	B	5
B	*NULL*	

- Bước 3: xây dựng cây FP điều kiện

  (tương tự như bước 1 nhưng áp dụng trên cơ sở mẫu vừa xây dựng xong)

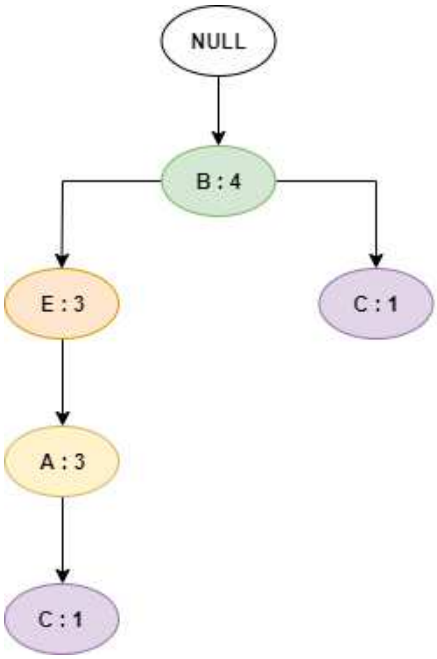
  ✓ Đối với từng phần tử, thống kê số tần suất xuất hiện của các thành phần trong cơ sở mẫu.

  ✓ Sau đó so sánh với độ hỗ trợ nhỏ nhất để loại bỏ thành phần không đạt.

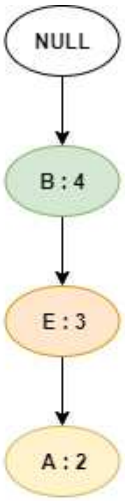
  ✓ Với các thành phần còn lại, đạt yêu cầu, tiến hành vẽ cây theo thứ tự F-list ban đầu ở bước 1.

  Các cây FP điều kiện

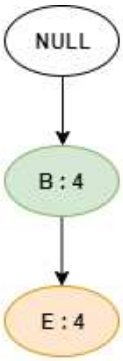
Xét D : 4



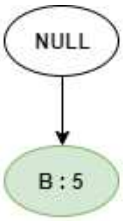
Xét C : 4



Xét A : 4



Xét E : 5



Xét B : 6

- Bước 4: xây dựng tập phổ biến

  Sau bước 3 ta sẽ có các cặp phần tử đi kèm với câu FP điều kiện của nó. Cây FP điều kiện có 2 dạng: 1 đường dẫn đơn hoặc có nhiều nhánh. Với trường hợp có nhiều nhánh ta tiến hành đệ quy trở lại từ bước 2 để biến nó thành đường dẫn đơn rồi xây dựng tập phổ biến như sau:

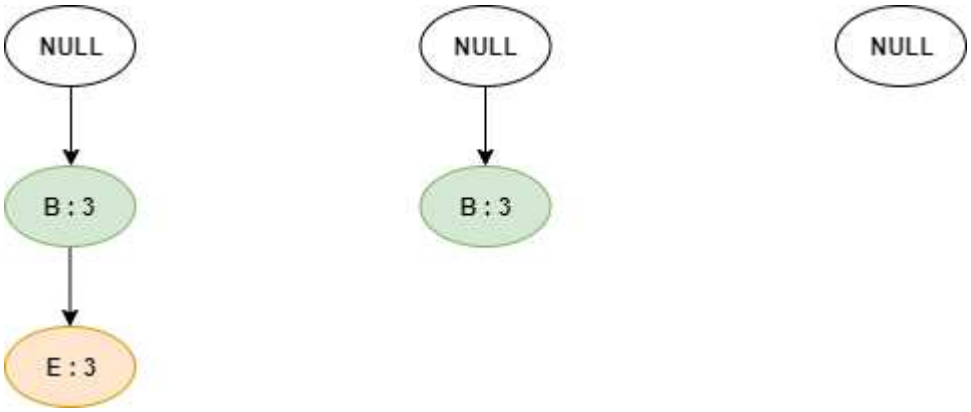
  ✓ Lấy tổ hợp các thành phần của nút trên cây.

  ✓ Tập phổ biến được xây dựng bằng cách lấy mỗi tổ hợp hợp với phần tử chủ chốt ban đầu (phần tử đi kèm với cây) và có tần suất xuất hiện bằng với tần suất nhỏ nhất của bất kỳ thành phần nào trong tập.

  Xét cây FP chứa nhiều hơn 1 nhánh D : 4, ta tiến hành đệ quy trở lại để xây dựng cây, như sau:

  Vì C chỉ có support count = 2 < min_support_count = 3, do đó ta không xét nút C.

  Xét nút A, với {D, A} : 3 ***Xét nút E, với {D, E} : 3*** ***Xét nút B, với {D, B} : 4***

  Lúc này tất cả những cây FP điều kiện đều đã có dạng đường dẫn đơn, tiến hành tổ hợp những phần tử để tìm ra các tập phổ biến 2 phần tử trở lên thỏa min_support_count:

  L_2 = {{A, D}: 3, {D, E}: 3, {B, D}: 4, {C, E}: 3, {B, C}: 4, {A, B}: 4, {A, E}: 4, {B, E}: 5}

  L_3 = {{A, B, D}: 3, {A, D, E}: 3, {B, D, E}: 3, {B, C, E}: 3, {A, B, E}: 4}

  L_4 = {{A, B, D, E}: 3}}

3. Tìm các luật kết hợp dựa trên các tập phổ biến.

   Các bước chính:

- Tạo các luật từ tập phổ biến tìm được.

- Tính confidence của các luật kết hợp
- So sánh với min_confidence để tìm các luật thỏa yêu cầu

  Ví dụ: Xét tập phổ biến *{A, B, D} : 3* có có các tập con không rỗng sau: *{A} : 4, {B} : 6, {D} : 4, {A, B} : 4, {A, D} : 3, {B, D} : 4*

A => {B, D}	3/4 = 75%
B => {A, D}	3/6 = 50%
D => {A, B}	3/4 = 75%
{A, B} => D	3/4 = 75%
{A, D} => B	3/3 = 100%
{B, D} => A	3/4 = 75%

  Với min_conf = 70%, dựa vào bảng trên ta có các luật kết hợp thỏa yêu cầu là:
  A => {B, D}, D => {A, B}, {A, B} => D, {A, D} => B, {B, D} => A.

2. Yêu cầu lập trình

Cho dữ liệu Online Retail[1] là lịch sử bán hàng của một cửa hàng bán lẻ trực tuyến tại Châu Âu. Cửa hàng này chuyên bán quà tặng độc đáo vào nhiều dịp lễ trong năm và cũng có nhiều khách hàng mua sỉ. Hãy thực hiện những yêu cầu sau:

1. Đọc dữ liệu vào chương trình
2. Tiền xử lý dữ liệu: 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.
3. Trong dữ liệu đã cho, có một số hóa đơn là hóa đơn tín dụng thay vì là hóa đơn ghi nợ vì vậy hãy xóa những hóa đơn đó. Chúng được xác định với ký tự 'C' chứa trong số hóa đơn InvoiceNo.
4. Thống kê số dòng dữ liệu theo từng quốc gia.
5. Lấy ra dữ liệu hóa đơn từ nước Anh 'United Kingdom' và gom nhóm cột Số lượng mua (Quantity) theo Số hóa đơn (InvoiceNo) và Tên mặt hàng (Description).
6. Chuyển đổi dữ liệu về dạng hot encoding, với mỗi dòng dữ liệu là một hóa đơn.
7. Chuyển đổi dữ liệu từ dạng hot encoding thành one-hot encoding.
8. Do cột 'POSTAGE' là tiền cước phí trên mỗi hóa đơn nên cần xóa nó đi.

9. Tìm tập phổ biến bằng thuật toán Apriori với min_sup = 3%.
10. Tạo luật kết hợp với min_conf = 50% và in ra các luật này.
11. Biểu diễn độ tin cậy, độ hỗ trợ của tập luật lên đồ thị phân tán (scatter plot).
12. Tìm tập phổ biến và luật kết hợp bằng thuật toán FP-Growth với min_sup = 3%, min_conf = 50%. So sánh kết quả với thuật toán Apriori ở trên.

Hướng dẫn: để thực hiện yêu cầu lập trình sinh viên cần cài đặt thư viện mlxtend và xlrd

1. Sau khi cài đặt, tiến hành import các thư viện cần thiết: thư viện máy học (mlxtend), thư viện hỗ trợ đọc file Excel (xlrd).

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

Sử dụng pandas để đọc tập tin dữ liệu với định dạng *.xlsx

```
df = pd.read_excel('Online Retail.xlsx')
```

Sau khi dữ liệu được đọc vào biến df, hiển thị một vài thông tin của biến này

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   InvoiceNo    541909 non-null  object        
 1   StockCode    541909 non-null  object        
 2   Description  540455 non-null  object        
 3   Quantity     541909 non-null  int64         
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64       
 6   CustomerID   406829 non-null  float64       
 7   Country      541909 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

2. Thực hiện các thao tác: 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.

```
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
```

Xem thử 10 dòng đầu của dữ liệu sau khi đã thực hiện các thao tác "làm sạch"

```
df.head(10)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

3. Có thể xem một ví dụ về loại hóa đơn tín dụng bằng câu lệnh như sau

```
df[df.InvoiceNo.str.contains('C', na=False)].head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom

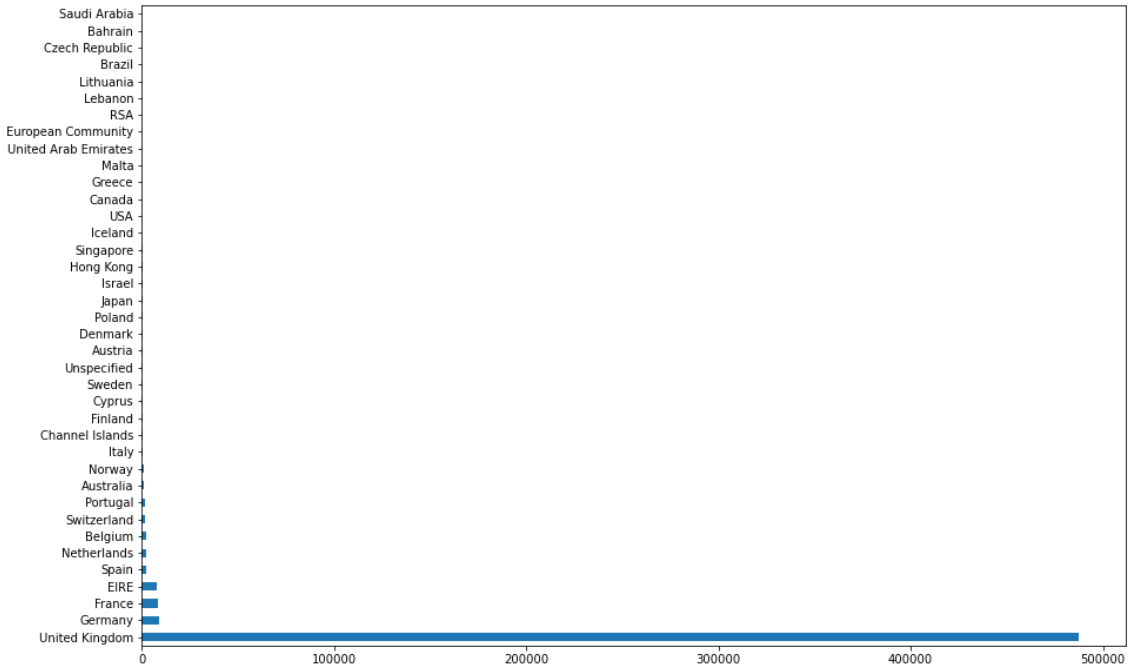
Tiến hành xóa những hóa đơn tín dụng

```
df = df[~df['InvoiceNo'].str.contains('C')]
```

4. Thống kê số dòng dữ liệu theo từng quốc gia, bằng câu lệnh sau

```
df['Country'].value_counts().plot(kind='barh', figsize=(15,10))
```

Kết quả được thể hiện bằng biểu đồ



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

```
basket = df[df['Country'] == "United Kingdom"].groupby(['InvoiceNo', 'Description'])['Quantity']
```

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

```
basket = basket.sum().unstack().reset_index().fillna(0).set_index('InvoiceNo')
```

Xem dữ liệu sau khi chuyển về dạng hot encoding

```
basket.head(10)
```

Description	*Boombox Ipod Classic	*USB Office Mirror Ball	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 DAISY PEGS IN WOOD BOX	12 EGG HOUSE PAINTED WOOD	12 HANGING EGGS HAND PAINTED	12 IVORY ROSE PEG PLACE SETTINGS	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	...
InvoiceNo											
536365	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536366	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536367	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536368	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536369	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536371	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536372	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536373	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536374	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536375	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

10 rows × 4175 columns

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

```
def encode_data(datapoint):
    if datapoint <= 0:
        return 0
    if datapoint >= 1:
        return 1
```

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

```
basket = basket.applymap(encode_data)
```

8. Xóa cột 'POSTAGE'

```
basket.drop('POSTAGE', inplace=True, axis=1)
```

9. Áp dụng thuật toán Apriori với min_sup = 3% để tìm tập phổ biến

```
itemsets = apriori(basket, min_support=0.03, use_colnames=True)
```

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

```
itemsets.head(10)
```

	support	itemsets
0	0.045803	(6 RIBBONS RUSTIC CHARM)
1	0.031124	(60 CAKE CASES VINTAGE CHRISTMAS)
2	0.040339	(60 TEATIME FAIRY CAKE CASES)
3	0.046928	(ALARM CLOCK BAKELIKE GREEN)
4	0.035142	(ALARM CLOCK BAKELIKE PINK)
5	0.049821	(ALARM CLOCK BAKELIKE RED)
6	0.036214	(ANTIQUE SILVER T-LIGHT GLASS)
7	0.073445	(ASSORTED COLOUR BIRD ORNAMENT)
8	0.042267	(BAKING SET 9 PIECE RETROSPOT)
9	0.035089	(BATHROOM METAL SIGN)

10. Tạo luật kết hợp với min_conf = 50%

```
rules = association_rules(itemsets, metric="confidence", min_threshold=0.5)
```

Xem thông tin về tập luật

```
rules.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   antecedents         10 non-null     object 
 1   consequents         10 non-null     object 
 2   antecedent support  10 non-null     float64
 3   consequent support  10 non-null     float64
 4   support             10 non-null     float64
 5   confidence          10 non-null     float64
 6   lift                10 non-null     float64
 7   leverage            10 non-null     float64
 8   conviction          10 non-null     float64
dtypes: float64(7), object(2)
memory usage: 848.0+ bytes
```

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

```
rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")
rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("unicode")
```

Viết lệnh in ra các luật đã tìm được

```
for i in range(len(rules)):
    print(rules.loc[i, 'antecedents'],' ==> ', rules.loc[i, 'consequents'],
          ' [', rules.loc[i, 'support'], ', ', rules.loc[i, 'confidence'], ']')
```

```
ALARM CLOCK BAKELIKE GREEN  ==>  ALARM CLOCK BAKELIKE RED  [ 0.030160175711148016 ,  0.6426940639269406 ]
ALARM CLOCK BAKELIKE RED  ==>  ALARM CLOCK BAKELIKE GREEN  [ 0.030160175711148016 ,  0.6053763440860216 ]
GREEN REGENCY TEACUP AND SAUCER  ==>  PINK REGENCY TEACUP AND SAUCER  [ 0.030910162318530027 ,  0.6177730192719486 ]
PINK REGENCY TEACUP AND SAUCER  ==>  GREEN REGENCY TEACUP AND SAUCER  [ 0.030910162318530027 ,  0.8207681365576103 ]
ROSES REGENCY TEACUP AND SAUCER  ==>  GREEN REGENCY TEACUP AND SAUCER  [ 0.03755290084105641 ,  0.7324973876698014 ]
GREEN REGENCY TEACUP AND SAUCER  ==>  ROSES REGENCY TEACUP AND SAUCER  [ 0.03755290084105641 ,  0.7505353319057816 ]
JUMBO  BAG BAROQUE BLACK WHITE  ==>  JUMBO BAG RED RETROSPOT  [ 0.03053516901483902 ,  0.6263736263736264 ]
JUMBO BAG PINK POLKADOT  ==>  JUMBO BAG RED RETROSPOT  [ 0.042052820485348474 ,  0.6773080241587576 ]
JUMBO SHOPPER VINTAGE RED PAISLEY  ==>  JUMBO BAG RED RETROSPOT  [ 0.03519580007499866 ,  0.5798764342453663 ]
JUMBO STORAGE BAG SUKI  ==>  JUMBO BAG RED RETROSPOT  [ 0.037392189425188835 ,  0.6176991150442478 ]
```

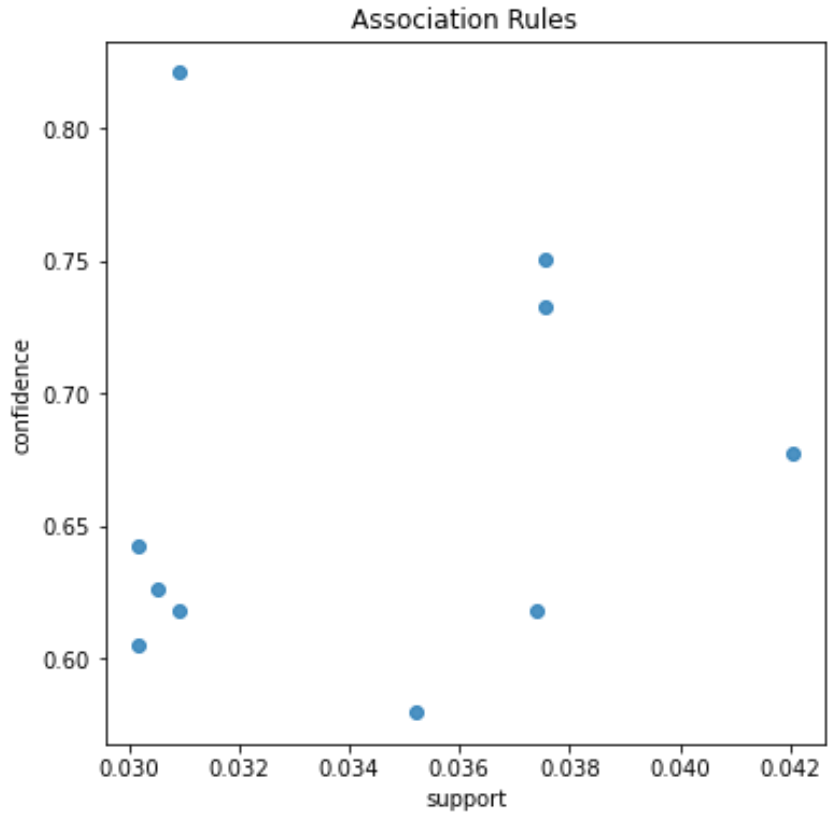
11. Lấy giá trị độ hỗ trợ và độ tin cậy của tập luật

```
support=rules['support'].values
confidence=rules['confidence'].values
```

Biểu diễn các thông tin này lên biểu đồ

```
plt.figure(figsize=(6, 6))
plt.title('Association Rules')
plt.xlabel('support')
plt.ylabel('confidence')
sns.regplot(x=support, y=confidence, fit_reg=False)
```

Kết quả thu được



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

```
from mlxtend.frequent_patterns import fpgrowth

itemsets = fpgrowth(basket, min_support=0.03, use_colnames=True)
```

Phần còn lại sinh viên thực hiện như đã hướng dẫn ở trên.

IV. Thực hành

1. (Cơ bản) Cho bảng dữ liệu ở một cửa hàng tạp hóa có 6 giao dịch như sau:

Transaction ID	Items
T1	HotDogs, Buns, Ketchup
T2	HotDogs, Buns
T3	HotDogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	HotDogs, Coke, Chips

Với min_sup = 33.33% và min_conf = 60%, sinh viên thực hiện lại các yêu cầu trên.

2. (Cơ bản) Cho bảng dữ liệu ở một cửa hàng văn phòng phẩm như sau:

TID	KÉO	COMPA	THƯỚC	TẬP TRẮNG	BÚT BI	BÚT MÀU	TẨY
T1		x		x	x		
T2	x		x	x	x		
T3		x		x	x		
T4	x	x		x	x		
T5			x				
T6					x		
T7				x			
T8							x
T9						x	x
T10						x	

Với min_sup = 30% và min_conf = 80%, sinh viên thực hiện lại các yêu cầu trên.

3. (Cơ bản) CSDL về Nhân viên được cho trong bảng sau:

	Giới tính (GT)	Tuổi (T)	Năng lực làm việc (NL)	Đã lập gia đình (LGD)	Thu nhập (TN)	Thăng chức (TC)
1	Nữ	20..25	Giỏi	Rồi	Rất cao	Có
2	Nam	20..25	Khá	Chưa	Khá	Không
3	Nữ	26..30	Giỏi	Chưa	Khá	Có
4	Nữ	31..40	T.Bình	Chưa	T.Bình	Có
5	Nam	26..30	T.Bình	Rồi	Rất cao	Không
6	Nữ	26..30	Khá	Chưa	Cao	Không
7	Nữ	31..40	Khá	Chưa	T.Bình	Không
8	Nam	26..30	Khá	Rồi	Cao	Có
9	Nữ	>40	Giỏi	Rồi	T.Bình	Không
10	Nữ	26..30	Giỏi	Chưa	Khá	Có

Cho B = {Tuổi, Năng lực làm việc, Thăng chức}. Hãy tìm tất cả các luật kết hợp có vế phải chỉ gồm thuộc tính Thăng chức (TC) thỏa ngưỡng minsup=30% và minconf = 80%.

4. (Cơ bản) Cho bảng dữ liệu ở một công ty có các giao dịch như sau:

TID	Items
T1	A, B, C, D
T2	A, B, C
T3	A, B, C
T4	A, B, D
T5	A, B
T6	A, C, D
T7	A, D
T8	B, C, D

Với min_support_count > 1, sinh viên hãy:

a) Tìm tất cả các tập phổ biến từ mẫu dữ liệu trên bằng giải thuật Apriori.
b) Tìm tất cả các tập phổ biến từ mẫu dữ liệu trên bằng giải thuật FP-growth.

5. (Lập trình) Hãy thực hiện lại bài tập lập trình ở phần hướng dẫn chung nhưng thay đổi các yêu cầu thành:

a) Nước Đức 'Germany', min_sup = 5% và min_conf = 50%.
b) Nước Pháp 'France', min_sup = 7% và min_conf = 70%.

6. (Lập trình) Cho mẫu dữ liệu Groceries Dataset[2] về các giao dịch trong một tháng của một cửa hàng outlet, gồm 9835 giao dịch và 169 items.
   Với min_sup từ 0.01% đến 0.1%, hãy:

a) Tìm tất cả các tập phổ biến trong từng trường hợp bằng giải thuật Apriori và ghi nhận thời gian xử lý của thuật toán. Gợi ý: sinh viên có thể sử dụng module *OneHotTransactions* hoặc *TransactionEncoder* của thư viện *mlxtend* để chuyển dữ liệu về dạng one-hot encoding trước khi đưa vào thuật toán.
b) Tìm tất cả các tập phổ biến trong từng trường hợp bằng giải thuật FP-Growth và ghi nhận thời gian xử lý của thuật toán.
c) Vẽ biểu đồ so sánh thời gian xử lý của hai thuật toán trên với nhau. Sinh viên đưa ra nhận xét và giải thích.

V. Bài tập thêm

1. Trong các phương pháp tìm kiếm luật kết hợp trên ta sử dụng hai giá trị min_sup và min_conf để đánh giá các luật tìm được. Tuy nhiên trong thực tế, nếu chỉ sử dụng hai giá trị này thì mô hình vẫn có thể sinh ra một số luật phi lí. Vì thế để giới hạn vấn đề này ta có thể bổ sung thêm một giá trị để đánh giá luật kết hợp đó là tính tương quan giữa hai vế của luật.
   Sinh viên tìm hiểu hai phương pháp phân tích tính tương quan giữa hai vế của luật sử dụng giá trị **Lift** và $\boldsymbol{\chi^2}$ và sử dụng để đánh giá các luật tìm được ở phần thực hành.
2. Chọn một ngôn ngữ lập trình, cài đặt thuật toán Apriori.
3. Tìm hiểu thuật toán: Apriori+, FPMax
4. Chọn 1 trong các kĩ thuật sau: Dùng bảng băm, giảm số lượng giao dịch trong tập giao dịch, chia nhỏ tập giao dịch và lấy mẫu trên tập giao dịch để cải tiến giải thuật Apriori đã viết trong bài tập thêm 2.

VI. Tài liệu tham khảo

1. *Online Retail Dataset*, UCI Machine Learning Repository
2. *Groceries Dataset*, [Michael Hahsler et al., 2006] Michael Hahsler, Kurt Hornik, and Thomas Reutterer (2006) Implications of probabilistic data modeling for mining association rules. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nuernberger, and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 598–605. Springer-Verlag.;
3. *Agrawal, Rakesh, and Ramakrishnan Srikant*. "Fast algorithms for mining association rules." Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215. 1994.
4. *Han, Jiawei, Jian Pei, Yiwen Yin, and Runying Mao*. "Mining frequent patterns without candidate generation. "A frequent-pattern tree approach." Data mining and knowledge discovery 8, no. 1 (2004): 53-87.