

RIORI | MARKET BASKET

N PHẨM THƯỜNG MUA ĐI KÈM



MARKET BASKET

- Giả sử giỏ hàng hóa gồm nhiều hóa đơn, mỗi hóa đơn là danh sách các sản hàng đã mua, thuật toán Apriori dùng để xác định các mối quan hệ kết hợp khách hàng mua trong cùng một giao dịch.
- Tìm xem sản phẩm nào hay được mua cùng nhau, từ đó đưa ra chiến lược kinh doanh hợp cho khách hàng.
- InvoiceNo, Stockcode, Description, Quantity
- Tất cả các sản phẩm trong tập dữ liệu
- Lấy dữ liệu → Phân tích bài toán → Trực quan hóa

Các khái niệm liên quan đến thuật toán Apriori

Án Apriori: là một thuật toán được sử dụng để tìm ra (luật kết hợp) giữa các sản phẩm trong một tập dữ liệu như các đơn hàng trong siêu thị hay giỏ hàng online.

Nhà: Tìm ra các mối quan hệ giữa các sản phẩm để từ đó kết hợp có ý nghĩa, giúp doanh nghiệp hiểu rõ hành vi mua sắm và gợi ý thông minh cho khách hàng.

Các khái niệm liên quan đến thuật toán Apriori

hợp các sản phẩm

itemset: Các tập hợp sản phẩm xuất hiện với tần suất cao trong dữ liệu (high support)

tần suất xuất hiện của itemset, tính bằng:

$$Support(A) = \frac{X}{Y}$$

(nửa A, Y: Tổng số giao dịch)

tần suất mua sản phẩm B nếu đã mua sản phẩm A

$$Confidence(A \Rightarrow B) = \frac{S}{T}$$

Độ hữu ích của mối quan hệ, bằng:

$$Lift(A \Rightarrow B) = \frac{Confidence(A \Rightarrow B)}{Support(A)}$$

Sản phẩm có liên quan chặt chẽ, thường được mua cùng.
Thường ảnh hưởng đến nhau.

Không làm tăng khả năng mua B

Các ví dụ liên quan đến thuật toán Apriori

u thị mini nhỏ muốn biết
đồm nào thường được mua
:

ang hóa hợp lý hơn
phẩm cho khách
nh thu nhờ hiểu hành vi mua
ệu đơn hàng như sau:

Đơn hàng

1

2

3

4

5

Các ví dụ liên quan đến thuật toán Apriori

Độ dung: Một sản phẩm phải
đóng ít nhất là 60% số giao
lượng số những khách hàng đã
mua A, có ít nhất 70% trong số
những người mua A cũng mua B
thì mới chấp nhận "nếu mua A thì sẽ mua B".

$$confidence = 0.7$$

$$support = 0.6$$

Đơn hàng

1

2

3

4

5

Các ví dụ liên quan đến thuật toán Apriori

biến ($\text{support} \geq 0.6$):

{Sữa}, {Trứng}

, {Sữa, Trứng}, {Bánh mì},

hẩm trở lên, ta tạo luật

→ Khả năng mua B

Trứng}:

ng

cả Sữa và Trứng: 3

Sữa: 4

ce = $3/4 = 0.75$

Sữa

Trứng: 4

ce = $3/4 = 0.75$

Tập sản phẩm	Số đơn hàng c
Sữa	4
Bánh mì	4
Trứng	3
Bơ	1
Sữa + Bánh mì	3
Sữa + Trứng	3
Bánh mì + Trứng	3
Sữa + Bánh mì + Trứng	2
Sữa + Bơ	0
Bơ + Trứng	1

Các ví dụ liên quan đến thuật toán Apriori

đảm trở lên, ta tạo luật

Khả năng mua B

rứng}:

g

Sữa và Trứng: 3

a: 4

e = $3/4 = 0.75$

a

ứng: 4

e = $3/4 = 0.75$

= $1.25 \rightarrow$ Khách hàng

này khả năng họ mua

so với trung bình.

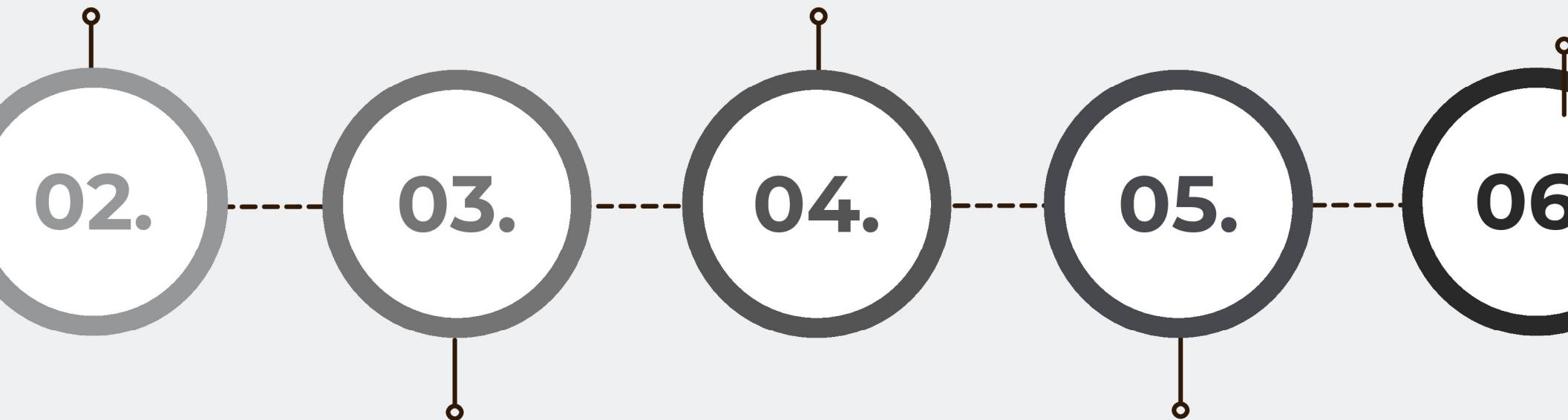
Luật	Support	Confidence
Sữa → Bánh mì	0.6	$0.6 / 0.8 = 0.75$
Bánh mì → Sữa	0.6	0.75
Sữa → Trứng	0.6	0.75
Trứng → Sữa	0.6	$0.6 / 0.6 = 1.00$
Bánh mì → Trứng	0.6	0.75
Trứng → Bánh mì	0.6	1.00

| MARKET BASKET

ơn → Nhị phân hóa
0: không mua)

Tạo ra các luật kết hợp với
mức confidence nhất định.

Đổi mã sản
tên s



Tối ưu min_support

→ Cân bằng giữa số itemset và
tổng số sản phẩm

Sắp xếp luật

theo Lift để tìm
luật mạnh

→

Đã được làm sạch & Đếm số lượng đơn hàng duy nhất và tổng

đữ data đã được làm sạch và
chiết là “InvoiceNo”,
“Description” và “Quantity”.

đơn hàng duy nhất và tổng
để hỗ trợ cho việc tính min

```
# Đếm số lượng đơn hàng (Invoice)
so_don_hang = df['InvoiceNo'].nunique()
print(f'Tổng số đơn hàng: {so_don_hang}')
```

Tổng số đơn hàng: 6708

```
so_san_pham = df['Description'].nunique()
so_san_pham
```

✓ 0.0s

3200

encoding & Chuyển đổi các ô trong bảng về dạng nhị phân

c encoding với mỗi dòng đại diện cho một hóa đơn, mỗi cột là số lượng sản phẩm đó trong hóa đơn.

trong bảng về dạng nhị phân với sản phẩm đó được mua (số là 1 và không được mua thì đặt là 0.

```
# Tạo bảng one-hot encoding cho các sản phẩm theo đơn hàng
basket = (df.groupby(['InvoiceNo', 'stockCode'])['Quantity']
           .sum().unstack().reset_index().fillna(0)
           .set_index('InvoiceNo'))
```

```
# Chuyển đổi số lượng thành binary (0 hoặc 1)
basket_sets = basket.applymap(lambda x: 1 if x > 0 else 0)
```

Tìm với các min support khác nhau rồi chọn ra min support phù hợp

một vài giá trị min_support khác nhau để chọn ra min_support thích hợp
min_sup in [0.03, 0.02, 0.015]:

```
frequent_itemsets = apriori(basket_sets, min_support=min_sup, use_colnames=True)
print(f"Min support: {min_sup}, số lượng itemset: {len(frequent_itemsets)})")
```

support: 0.03, số lượng itemset: 127

support: 0.02, số lượng itemset: 334

support: 0.015, số lượng itemset: 672

Chọn min support 0.02

giá trị min_support = 0.02 và áp dụng Apriori với min_support đã chọn

```
frequent_itemsets = apriori(basket_sets, min_support=0.02, use_colnames=True)
print(f"Tim thấy {len(frequent_itemsets)} tập phổ biến")
frequent_itemsets.head(10))
```

với mức confidence nhất định và sắp xếp luật theo Lift giảm dần

```
in_confidence = 0.3
s(frequent_itemsets, metric="confidence", min_threshold=0.3)

uật theo lift giảm dần
sort_values('lift', ascending=False)
```

	antecedents	support	confidence
119	(2311, 2312)	0.021467	0.3
120	(2311, 2313)	0.021467	0.3
69	(2191, 2192)	0.022063	0.3
70	(2191, 2193)	0.022063	0.3
214	(22697, 22698)	0.027579	0.3
211	(22697, 22699)	0.027579	0.3
212	(22698, 22699)	0.027579	0.3
213	(22699, 22700)	0.027579	0.3
113	(22699, 22701)	0.027579	0.3
114	(22699, 22702)	0.027579	0.3

Tên sản phẩm để dễ dàng trong việc quan sát và phân bi

ển ánh xạ mã sản phẩm với tên sản phẩm

```
es = df[['StockCode', 'Description']].drop_duplicates()  
c = dict(zip(product_names['StockCode'], product_names['Description']))
```

đổi từ mã sản phẩm sang tên sản phẩm

```
_codes_to_names(itemset):  
    tuple(product_dict.get(item, str(item)) for item in itemset)
```

ho kết quả

```
ecedents_names'] = rules['antecedents'].apply(convert_codes_to_names)  
equent_names'] = rules['consequents'].apply(convert_codes_to_names)
```

kết quả với tên sản phẩm

```
[['antecedents_names', 'consequents_names', 'support', 'confidence', 'lift']]
```

Start Chat to Generate Code (Ctrl+Shift+C)

```
6 lift cao (> 10) và confidence cao (> 0.5)
rules[(rules['lift'] > 10) & (rules['confidence'] > 0.5)]
mạnh (lift > 10, confidence > 0.5): {len(strong_rules)})")
```

Luật mạnh nhất

```
rules[['antecedents_names', 'consequents_names', 'support', 'confidence',
```

• sản phẩm thường đi kèm

đồ thị có hướng
ph()

các luật kết hợp và thêm các cạnh vào đồ thị

n rules.iterrows():

nts = list(row['antecedents'])

nts = list(row['consequents'])

qua các sản phẩm trong antecedents và consequents, tạo cạnh
antecedents:

c in consequents:

G.add_edge(a, c, lift=row['lift'], confidence=row['confidence'])

• sản phẩm thường đi kèm

mạng

```
figsize=(14, 10))
```

các node

```
spring_layout(G, k=0.5, seed=42) # Layout đẹp
```

số của cạnh từ thuộc tính 'lift'

```
['lift'] for (u, v, d) in G.edges(data=True)]
```

e, nhãn và cạnh của đồ thị

```
nx.draw(G, pos, node_size=1000, node_color='lightblue')
```

```
nx.draw(G, pos, font_size=9)
```

```
nx.draw(G, pos, edge_color=weights, edge_cmap=plt.cm.plasma, arrows=True)
```

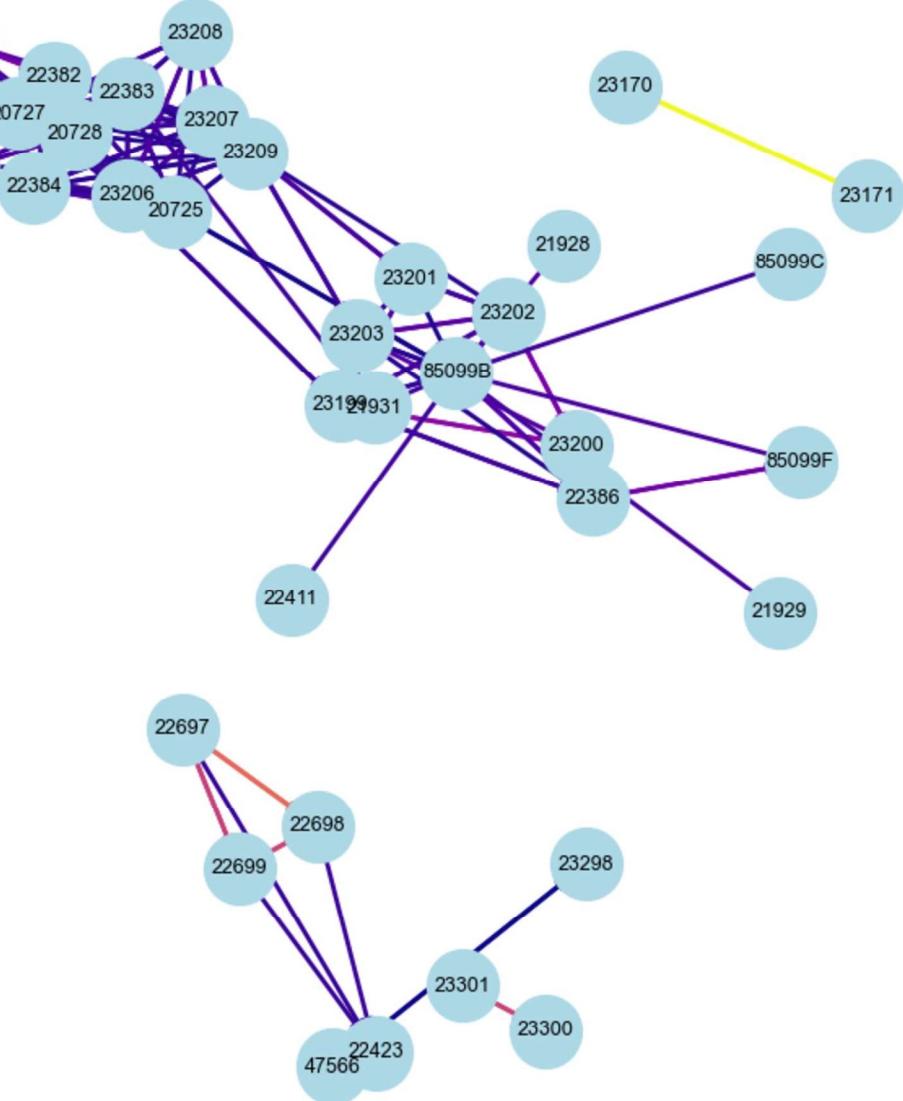
iêu đề và ẩn trục

iêu đồ mạng: Các luật kết hợp sản phẩm', fontsize=16)

```
'")
```

→ sản phẩm thường đi kèm

g: Các luật kết hợp sản phẩm



1. Có nhiều cụm sản phẩm g

→ Như cụm trung tâm (chứa 23209, 85099B, v.v.)

→ Ý nghĩa: Các sản phẩm này cùng nhau → nên trưng bày g xuất combo khi khách chọn

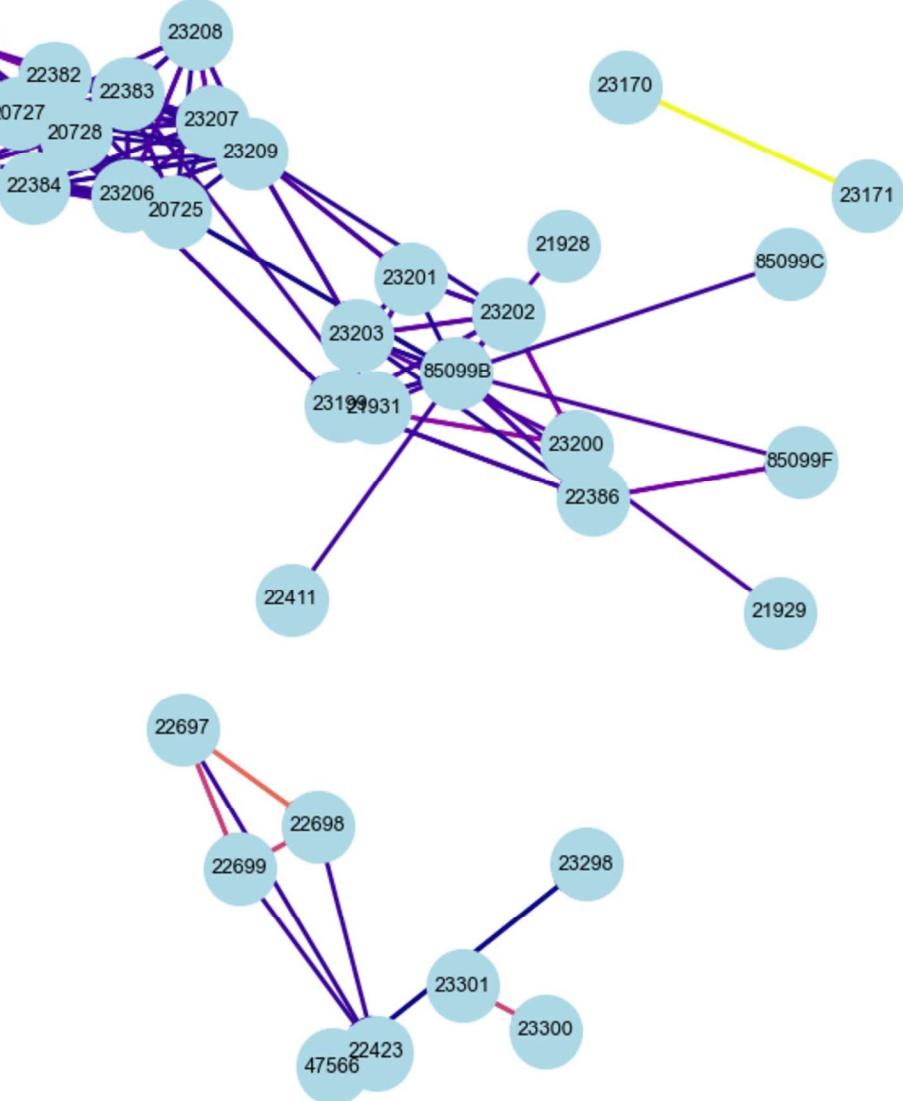
2. Có một số cụm rời rạc và

→ Như nhóm gồm 22355, 207 kết với các cụm khác.

→ Ý nghĩa: Đây có thể là sản phẩm sản phẩm niche, chỉ đi với nh hưởng nhiều đến hệ thống c

→ sản phẩm thường đi kèm

g: Các luật kết hợp sản phẩm



3. Một số cặp sản phẩm có liên quan (tím/xanh đậm)

- Ví dụ cặp 22698–22697 hoặc 23208–23209
- Ý nghĩa: Những luật có độ cao hơn hoặc lift cao hơn → mua trước để xuất / khuyến mãi cặp

4. Một vài sản phẩm là trung tâm

- Như mã 85099B kết nối với nhiều sản phẩm
- Ý nghĩa: Đây là sản phẩm “hàng phổ biến, dễ kết hợp với rất phù hợp để làm sản phẩm chính để upsell)

O 2 SẢN PHẨM

O 3 SẢN PHẨM

Antecedents	Consequents	Support
REGENCY TEA PLATE ROSES	REGENCY TEA PLATE GREEN	0.0
REGENCY TEA PLATE GREEN	REGENCY TEA PLATE ROSES	0.0
SCANDINAVIAN PAISLEY PICNIC BAG	PINK VINTAGE PAISLEY PICNIC BAG	0.0
PINK VINTAGE PAISLEY PICNIC BAG	SCANDINAVIAN PAISLEY PICNIC BAG	0.0
PINK REGENCY TEACUP AND SAUCER	GREEN REGENCY TEACUP AND SAUCER, ROSES REGENCY	0.0
GREEN REGENCY TEACUP AND SAUCER, ROSES REGENCY	PINK REGENCY TEACUP AND SAUCER	0.0
PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY	GREEN REGENCY TEACUP AND SAUCER	0.0
GREEN REGENCY TEACUP AND SAUCER	PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY	0.0
GREEN REGENCY TEACUP AND SAUCER	PINK REGENCY TEACUP AND SAUCER	0.0
PINK REGENCY TEACUP	GREEN REGENCY TEACUP	0.0

O 2 SẢN PHẨM

O 3 SẢN PHẨM

Antecedents	Consequents	Sup.
PINK REGENCY TEACUP AND SAUCER	GREEN REGENCY TEACUP AND SAUCER, ROSES REGENCY TEACUP AND SAUCER	0.0
GREEN REGENCY TEACUP AND SAUCER, ROSES REGENCY TEACUP AND SAUCER	PINK REGENCY TEACUP AND SAUCER	0.0
PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY TEACUP AND SAUCER	GREEN REGENCY TEACUP AND SAUCER	0.0
GREEN REGENCY TEACUP AND SAUCER	PINK AND SAUCER PINK REGENCY TEACUP AND SAUCER, ROSES REGENCY TEACUP AND SAUCER	0.0
ROSES REGENCY TEACUP AND SAUCER	GREEN REGENCY TEACUP AND SAUCER, PINK REGENCY TEACUP AND SAUCER	0.0
GREEN REGENCY TEACUP AND SAUCER, PINK REGENCY TEACUP	ROSES REGENCY TEACUP AND SAUCER	0.0

4

ANK YOU

