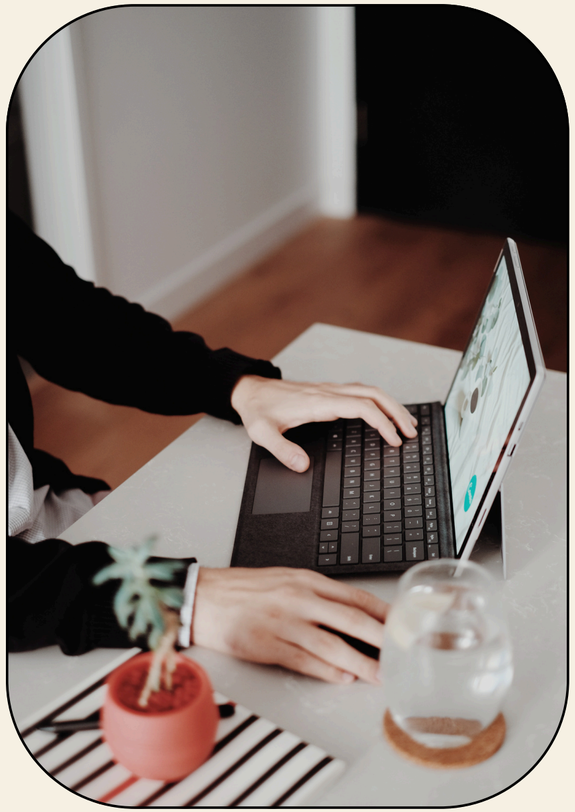
BSC_DPDM24



DATA REDUCED_MARKETBASKET

HW 3

1. *วิเคราะห์พฤติกรรมผู้บริโภค (GROUPBY CUSTOMER ID) แต่ละทวีป*

2. *การวิเคราะห์ผลิตภัณฑ์ตามพฤติกรรมของลูกค้า HIGH-VALUE และพฤติกรรมการซื้อของลูกค้า HIGH-VALUE ในยุโรปในช่วงคริสต์มาสและปีใหม่ (ที่กำหนดจากยอดการซื้อรวมสูงสุด) มักจะซื้อผลิตภัณฑ์ประเภทไหนร่วมกันในช่วงเทศกาลและซื้อสินค้าอะไรแตกต่างจากลูกค้าทั่วไป*

IMPORT DATA

```
[8]  1 import pandas as pd
     2 import os

[9]  1 from google.colab import drive
     2 drive.mount('/content/drive')
     3 path = '/content/drive/MyDrive/bsc_dpdm24_data'

     Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[10] 1 data = pd.read_csv(os.path.join(path,'reduced_marketbasket.csv'),encoding='latin-1')
```

```
[11] 1 data
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/01/2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/01/2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/01/2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/01/2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/01/2010 08:26	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/09/2011 12:50	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/09/2011 12:50	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/09/2011 12:50	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/09/2011 12:50	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/09/2011 12:50	4.95	12680.0	France

541909 rows × 8 columns

DATA PREPARETION

- แยก เดือน/วัน/ปี ออกจาก เวลา
- แปลงเดือนเช่นจาก 01 -> 1 ให้เป็นรูปแบบเดียวกัน

```
[ ]  1 data_test
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	date	day	month	year
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/01/2010 08:26	2.55	17850.0	United Kingdom	12/01/2010	1	12	2010
1	536365	71053	WHITE METAL LANTERN	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/01/2010 08:26	2.75	17850.0	United Kingdom	12/01/2010	1	12	2010
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010
...	...	...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/09/2011 12:50	0.85	12680.0	France	12/09/2011	9	12	2011
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/09/2011 12:50	2.10	12680.0	France	12/09/2011	9	12	2011
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/09/2011 12:50	4.95	12680.0	France	12/09/2011	9	12	2011

DATA PREPARETION

แบ่งทวีป

```python
# สร้าง dictionary สำหรับ mapping ประเทศไปยังทวีป
continent_mapping = {
    "Australia": "Oceania",
    "Austria": "Europe",
    "Bahrain": "Asia",
    "Belgium": "Europe",
    "Brazil": "South America",
    "Canada": "North America",
    "Channel Islands": "Europe",
    "Cyprus": "Asia",
    "Czech Republic": "Europe",
    "Denmark": "Europe",
    "EIRE": "Europe",
    "European Community": "Europe",
    "Finland": "Europe",
    "France": "Europe",
    "Germany": "Europe",
    "Greece": "Europe",
    "Hong Kong": "Asia",
    "Iceland": "Europe",
    "Israel": "Asia",
    "Italy": "Europe",
    "Japan": "Asia",
    "Lebanon": "Asia",
    "Lithuania": "Europe",
    "Malta": "Europe",
    "Netherlands": "Europe",
    "Norway": "Europe",
    "Poland": "Europe",
    "Portugal": "Europe",
    "RSA": "Africa",
    "Saudi Arabia": "Asia",
    "Singapore": "Asia",
    "Spain": "Europe",
    "Sweden": "Europe",
    "Switzerland": "Europe",
    "USA": "North America",
    "United Arab Emirates": "Asia",
    "United Kingdom": "Europe",
    "Unspecified": "Unspecified"
}
```

```python
data_test['Continent'] = data_test['Country'].map(continent_mapping)
```

```python
data_test
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	date	day	month	year	Continent
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/01/2010 08:26	2.55	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe
1	536365	71053	WHITE METAL LANTERN	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/01/2010 08:26	2.75	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe
...	...	...	...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/09/2011 12:50	0.85	12680.0	France	12/09/2011	9	12	2011	Europe
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/09/2011 12:50	2.10	12680.0	France	12/09/2011	9	12	2011	Europe
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011	Europe
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011	Europe
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/09/2011 12:50	4.95	12680.0	France	12/09/2011	9	12	2011	Europe

541909 rows × 13 columns

Code สำหรับหาชื่อสินค้า จาก Column "Description"

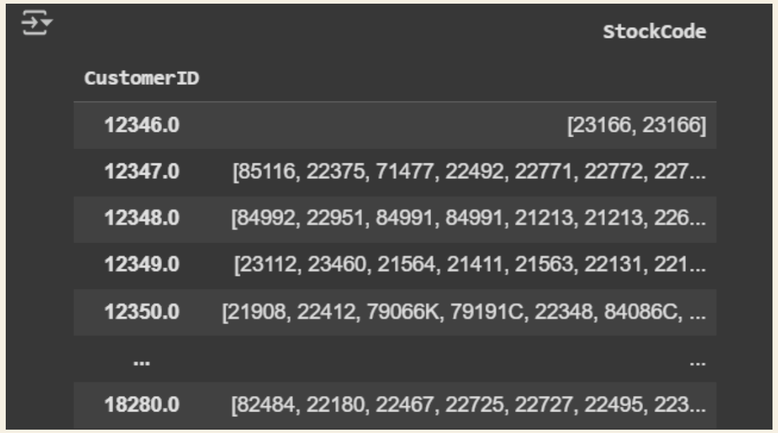
```
[ ]  1 def finditem(x,y) :
     2   L=set(data[(data['StockCode']==x)|(data['StockCode']==y)]['Description'])
     3   return L
```

```
[ ]  1 def finditems(x, y='', z='', w='') :
     2   I =set(data[(data['StockCode']==x)|(data['StockCode']==y)|(data['StockCode']==z)|(data['StockCode']==w)]['Description'])
     3   return I
```

EUROPE

(groupby customer id)

```
[ ]
# prompt: เลือกข้อมูลที่เป็นทวีป Europe
# Filter data for Europe
europe_data = data_test[data_test['Continent'] == 'Europe']

# Group by CustomerID for Europe
europe_customer_data = europe_data.groupby('CustomerID')['StockCode'].apply(list)

europe_customer_data
```



CustomerID	StockCode
12346.0	[23166, 23166]
12347.0	[85116, 22375, 71477, 22492, 22771, 22772, 227...
12348.0	[84992, 22951, 84991, 84991, 21213, 21213, 226...
12349.0	[23112, 23460, 21564, 21411, 21563, 22131, 221...
12350.0	[21908, 22412, 79066K, 79191C, 22348, 84086C, ...
...	...
18280.0	[82484, 22180, 22467, 22725, 22727, 22495, 223...

```
min_sup = 0.07

frequentItemsetsEU= list(apriori(europe_customer_data,min_support = min_sup))
for i in frequentItemsetsEU:
    if i[0].__len__()>1:
        print('yes')
        print(i)
```

EUROPE

(groupby customer id)

```
[ ]   1 finditem('22423','22699')
```

yes
RelationRecord(items=frozenset({'22697', '22699'}), support=0.07279870580078576, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'22697', '22699'}), confidence=0.07279870580078576, lift=1.0), OrderedStatistic(items_base=frozenset({'22697'}), items_add=frozenset({'22699'}), confidence=0.8267716535433071, lift=8.55847116000452), OrderedStatistic(items_base=frozenset({'22699'}), items_add=frozenset({'22697'}), confidence=0.7535885167464115, lift=8.55847116000452)])

22423 : REGENCY CAKESTAND 3 TIER

confidence=0.8268
lift=8.5584

confidence=0.7536
lift=8.5584

22699 : ROSES REGENCY TEACUP AND SAUCER

NORTH AMERICA

(groupby customer id)

CustomerID	StockCode
12558.0	[22712, 23344, 85099B, 23209, 79191C, 23392, 8...
12607.0	[22551, 21915, 22619, 22138, 21524, 21668, 216...
12646.0	[23173, 22423, 22962, 22963, 23328, 22907, 211...
12733.0	[22722, 22979, 84987, 22720, 22993, 47580, 229...
15388.0	[20886, 79030D, 21132, 84879, 84755, 85116, 71...
17443.0	[37370]
17444.0	[22417, 22951, 22939, 23184, 85232D, 22626, 22...
17844.0	[21993, 23295, 23293, 23296, 23294]

```
[ ]  1 min_sup = 0.25
     2
     3 frequentItemsetsNA= list(apriori(NorthAmerica_customer_data,min_support = min_sup))
     4 for i in frequentItemsetsNA:
     5     if i[0].__len__()>1:
     6         print('yes')
     7         print(i)
```

RelationRecord(items=frozenset({'21122', '21121'}), support=0.25, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'21122', '21121'}), confidence=0.25, lift=1.0), OrderedStatistic(items_base=frozenset({'21121'}), items_add=frozenset({'21122'}), confidence=1.0, lift=4.0), OrderedStatistic(items_base=frozenset({'21122'}), items_add=frozenset({'21121'}), confidence=1.0, lift=4.0)])

22734 : SET OF 6 RIBBONS VINTAGE CHRISTMAS

21123 : SET/10 IVORY POLKADOT PARTY CANDLES

23513 : EMBROIDERED RIBBON REEL SUSIE

21121 : SET/10 RED POLKADOT PARTY CANDLES

23293 : SET OF 12 FAIRY CAKE BAKING CASES

21122 : SET/10 PINK POLKADOT PARTY CANDLES

21124 : SET/10 BLUE POLKADOT PARTY CANDLES

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=0.6667 lift=2.6667

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

confidence=1.0 lift=4.0

SOUTH AMERICA

(groupby customer id)

มีลูกค้าคนเดียว และทั้ง 32 StockCode อยู่ใน InvoiceNo = 550201

```
[31]  1 # Filter data for South America
      2 SouthAmerica_data = data_test[data_test['Continent'] == 'South America']
      3
      4 # Group by CustomerID for North America
      5 SouthAmerica_customer_data = SouthAmerica_data.groupby('CustomerID')['StockCode'].apply(list)
      6
      7 SouthAmerica_customer_data
```

CustomerID	StockCode
12769.0	[22423, 22699, 22697, 22698, 22366, 21430, 226...

dtype: object

```
 1 # prompt: นับจำนวน StockCodeแต่ละอัน ที่ลูกค้า CustomerID = 12769.0 ซื้อ
 2
 3 # Filter data for customer ID 12769.0
 4 customer_12769_data = data_test[data_test['CustomerID'] == 12769.0]
 5
 6 # Count the occurrences of each StockCode for customer 12769.0
 7 stockcode_counts = customer_12769_data['StockCode'].value_counts()
 8
 9 # Print the counts
10 stockcode_counts
```

StockCode	count
22423	1
22699	1
23052	1
23049	1
23051	1
23050	1
23053	1

ทั้ง 32 StockCode เป็น 1 ทั้งหมด

SOUTH AMERICA

(groupby customer id)

```python
# prompt: นับจำนวน StockCodeแต่ละอัน ที่ลูกค้า CustomerID = 12769.0 ซื้อ พร้อมใส่Column Description และ Quantity
# Filter data for CustomerID 12769.0
customer_data = data_test[data_test['CustomerID'] == 12769.0]

# Group by StockCode and count occurrences, then reset index
stock_counts = customer_data.groupby('StockCode').agg(
    Description=('Description', 'first'),  # Get the first description for each StockCode
    Quantity=('Quantity', 'sum')  # Sum the quantities for each StockCode
).reset_index()

# Sort by Quantity in descending order
stock_counts = stock_counts.sort_values(by='Quantity', ascending=False)

stock_counts
```

	StockCode	Description	Quantity
16	22630	DOLLY GIRL LUNCH BOX	24
7	21430	SET/3 RED GINGHAM ROSE STORAGE BOX	24
22	22993	SET OF 4 PANTRY JELLY MOULDS	24
21	22722	SET OF 6 SPICE TINS PANTRY DESIGN	24
20	22699	ROSES REGENCY TEACUP AND SAUCER	24
19	22698	PINK REGENCY TEACUP AND SAUCER	24
18	22697	GREEN REGENCY TEACUP AND SAUCER	24
31	84971S	SMALL HEART FLOWERS HOOK	24
12	22423	REGENCY CAKESTAND 3 TIER	16
4	21166	COOK WITH WINE METAL SIGN	12
15	22629	SPACEBOY LUNCH BOX	12
5	21181	PLEASE ONE PERSON METAL SIGN	12
13	22488	NATURAL SLATE RECTANGLE CHALKBOARD	12
14	22494	EMERGENCY FIRST AID TIN	12
17	22662	LUNCH BAG DOLLY GIRL DESIGN	10
8	21770	OPEN CLOSED METAL SIGN	10
11	22382	LUNCH BAG SPACEBOY DESIGN	10
10	22366	DOORMAT AIRMAIL	10
6	21260	FIRST AID TIN	6
9	21906	PHARMACIE FIRST AID TIN	6
29	23178	JAM CLOCK MAGNET	6
30	23179	CLOCK MAGNET MUM'S KITCHEN	6
1	15056N	EDWARDIAN PARASOL NATURAL	3
3	20679	EDWARDIAN PARASOL RED	3
2	15056P	EDWARDIAN PARASOL PINK	3
0	15056BL	EDWARDIAN PARASOL BLACK	3
23	23049	RECYCLED ACAPULCO MAT RED	2
24	23050	RECYCLED ACAPULCO MAT GREEN	2
25	23051	RECYCLED ACAPULCO MAT BLUE	2
26	23052	RECYCLED ACAPULCO MAT TURQUOISE	2
27	23053	RECYCLED ACAPULCO MAT PINK	2
28	23054	RECYCLED ACAPULCO MAT LAVENDER	2

SOUTH AMERICA

(groupby customer id)

รายการสินค้าที่ซื้อในปริมาณมากที่สุด (24 ชิ้น)

รายการสินค้าที่ซื้อในปริมาณกลาง (10-16 ชิ้น)

รายการสินค้าที่ซื้อในปริมาณน้อย (น้อยกว่า 10 ชิ้น)

OCEANIA

(groupby customer id)

```
# เลือกข้อมูลที่เป็นทวีป Oceania

# Filter data for Oceania
oceania_data = data_test[data_test['Continent'] == 'Oceania']

# Group by CustomerID for Oceania
oceania_customer_data = oceania_data.groupby('CustomerID')['StockCode'].apply(list)

oceania_customer_data
```

```
[109] min_sup = 0.4
      frequentItemsetsOC= list(apriori(oceania_customer_data,min_support = min_sup))
      for i in frequentItemsetsOC:
          if i[0].__len__()>1:
              print('yes')
              print(i)
```

CustomerID	StockCode
12386.0	[22567, 22915, 22926, 22953, 21906, 22495, 225...
12388.0	[84970L, 71459, 22429, 22262, 47590B, 47590A, ...
12393.0	[21581, 22619, 84997B, 20727, 20726, 22383, 21...
12415.0	[22078, 22079, 22080, 22077, 22505, 22516, 225...
12422.0	[20728, 20713, 21937, 21936, 21932, 20717, 225...
12424.0	[22583, 22584, 22812, 22813, 22222, 22333, 224...
12431.0	[22941, 21622, 21791, 35004C, 35004G, 85014B, ...
12434.0	[22333, 22094, 21217, 20665, 22232, 23052, 230...
16321.0	[22087, 85123A, 21056, 22960, 22087, 22325, 22...

OCEANIA

(groupby customer id)

yes
RelationRecord(items=frozenset({'22617', '22138'}), support=0.4444444444444444,
ordered_statistics=[OrderedStatistic(items_base=frozenset(),
items_add=frozenset({'22617', '22138'}), confidence=0.4444444444444444, lift=1.0),
OrderedStatistic(items_base=frozenset({'22138'}), items_add=frozenset({'22617'}), confidence=1.0, lift=2.25),
OrderedStatistic(items_base=frozenset({'22617'}), items_add=frozenset({'22138'}), confidence=1.0, lift=2.25)])

22138 BAKING SET 9 PIECE RETROSPOT

confidence=1.0
lift=2.5

confidence=1.0
lift=2.5

22617 BAKING SET SPACEBOY DESIGN,
mouldy, thrown away

AFRICA

(groupby customer id)

```
[78] # เลือกข้อมูลที่เป็นทวีป Africa

     # Filter data for Africa
     africa_data = data_test[data_test['Continent'] == 'Africa']

     # Group by CustomerID for Africa
     africa_customer_data = africa_data.groupby('CustomerID')['StockCode'].apply(list)

     africa_customer_data
```

CustomerID	StockCode
12446.0	[21238, 21243, 23240, 23209, 23201, 23205, 219...

```
# filter data for customer ID 12446.0
customer_12446_data = data_test[data_test['CustomerID']==12446.0]

# Count the occurrences of each StockCode for customer 12446.0
stockcode_counts = customer_12446_data['StockCode'].value_counts()
stockcode_counts
```

StockCode	count
21238	1
21242	1
21340	1
85066	1
23407	1
23469	1
23284	1
22727	1
22624	1

ทั้ง 64 StockCode
เป็น 1 ทั้งหมด

AFRICA

(groupby customer id)

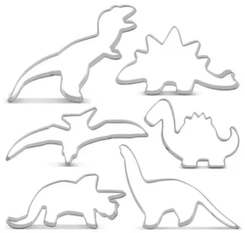
```
# Filter data for CustomerID = 12446.0
customer_data = data_test[data_test['CustomerID'] == 12446.0]

# Group by StockCode for CustomerID = 12446.0
stock_counts = customer_data.groupby('StockCode').agg(
    Description=('Description', 'first'),
    Quantity=('Quantity', 'sum')
).reset_index()

stock_counts = stock_counts.sort_values(by='Quantity', ascending=False)

stock_counts
```

	StockCode	Description	Quantity
31	22915	ASSORTED BOTTLE TOP MAGNETS	12
26	22620	4 TRADITIONAL SPINNING TOPS	12
23	22585	PACK OF 6 BIRDY GIFT TAGS	12
54	84375	SET OF 20 KIDS COOKIE CUTTERS	12
9	21889	WOODEN BOX OF DOMINOES	12
16	22384	LUNCH BAG PINK POLKADOT	10
2	20718	RED RETROSPOT SHOPPER BAG	10
3	20725	LUNCH BAG RED RETROSPOT	10
35	23201	JUMBO BAG ALPHABET	10
56	85099B	JUMBO BAG RED RETROSPOT	10
36	23203	JUMBO BAG VINTAGE DOILY	10
20	22478	BIRDHOUSE GARDEN MARKER	10
37	23205	CHARLOTTE BAG VINTAGE ALPHABET	10
38	23209	LUNCH BAG VINTAGE DOILY	10
17	22411	JUMBO SHOPPER VINTAGE RED PAISLEY	10
0	20676	RED RETROSPOT BOWL	8
1	20677	PINK POLKADOT BOWL	8
52	47591D	PINK FAIRY CAKE CHILDRENS APRON	8
6	21243	PINK POLKADOT PLATE	8
5	21242	RED RETROSPOT PLATE	8
4	21238	RED RETROSPOT CUP	8
39	23240	SET OF 4 KNICK KNACK TINS DOILY	6
44	23348	CHILDRENS TOY COOKING UTENSIL SET	6
48	23469	CARD HOLDER LOVE BIRD SMALL	6
49	23528	SPACEBOY WALL ART	6
34	23169	CLASSIC GLASS COOKIE JAR	6
32	22960	JAM MAKING SET WITH JARS	6
42	23298	SPOTTY BUNTING	6
29	22666	RECIPE BOX PANTRY YELLOW DESIGN	6
8	21580	RABBIT DESIGN COTTON TOTE BAG	6
19	22467	GUMBALL COAT RACK	6
13	22178	VICTORIAN GLASS HANGING T-LIGHT	6
15	22381	TOY TIDY PINK POLKADOT	5
14	22380	TOY TIDY SPACEBOY	5
10	21936	RED RETROSPOT PICNIC BAG	5
50	47566	PARTY BUNTING	4

AFRICA

(groupby customer id)






รายการสินค้าที่ซื้อในปริมาณมากที่สุด (12 ชิ้น)






รายการสินค้าที่ซื้อในปริมาณปานกลาง (8-10 ชิ้น)



รายการสินค้าที่ซื้อในปริมานี่น้อยกว่า 8 ชิ้น

UNSPECIFIED

(groupby customer id)

RelationRecord(items=frozenset({'22585', '22150'}), support=0.5, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'22585', '22150'}), confidence=0.5, lift=1.0), OrderedStatistic(items_base=frozenset({'22150'}), items_add=frozenset({'22585'}), confidence=1.0, lift=2.0), OrderedStatistic(items_base=frozenset({'22585'}), items_add=frozenset({'22150'}), confidence=1.0, lift=2.0)])

22585 3 STRIPEY MICE FELTCRAFT

confidence=1.0

lift= 2.0

lift= 2.0

confidence=1.0

22150 PACK OF 6 BIRDY GIFT TAGS

ANALYSIS OF PRODUCTS BASED ON HIGH-VALUE CUSTOMER BEHAVIOR

DEFINE HIGH-VALUE CUSTOMERS

- High-Value Customers are typically defined as those who contribute the most to the total revenue.
- Calculate the total purchase amount for each customer.

```
# Calculate total purchase amount for each customer
data_test['TotalPurchase'] = data_test['Quantity'] * data_test['UnitPrice']
customer_total_spending = data_test.groupby('CustomerID')['TotalPurchase'].sum().reset_index()
customer_total_spending
```

	CustomerID	TotalPurchase
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40
...	...	...
4367	18280.0	180.60
4368	18281.0	80.82
4369	18282.0	176.60
4370	18283.0	2094.88
4371	18287.0	1837.28

4372 rows × 2 columns

DEFINE HIGH-VALUE CUSTOMERS

- Identify the top 10% of customers based on total purchase amount as High-Value Customers.

```
# Identify High-Value Customers (top 10%)
high_value_threshold = customer_total_spending['TotalPurchase'].quantile(0.9)
high_value_customers = customer_total_spending[customer_total_spending['TotalPurchase'] >= high_value_threshold]['CustomerID']
high_value_customers
```

	CustomerID
1	12347.0
10	12357.0
12	12359.0
15	12362.0
20	12370.0
...	...
4324	18223.0
4326	18225.0
4327	18226.0
4330	18229.0
4347	18251.0

438 rows × 1 columns

dtype: float64

FILTER DATA FOR FESTIVE SEASONS

- Focus on transactions during the festive season (e.g., December for Christmas and New Year).

```
# Filter data for festive season (December)
festive_data = data_test[(data_test['month'] == 12) & (data_test['day'].between(1, 31))]
festive_data
```

```
/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the resul
  and should_run_async(code)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	date	day	month	year	Continent	TotalPurchase
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/01/2010 08:26	2.55	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe	15.30
1	536365	71053	WHITE METAL LANTERN	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/01/2010 08:26	2.75	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/01/2010 08:26	3.39	17850.0	United Kingdom	12/01/2010	1	12	2010	Europe	20.34
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/09/2011 12:50	0.85	12680.0	France	12/09/2011	9	12	2011	Europe	10.20
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/09/2011 12:50	2.10	12680.0	France	12/09/2011	9	12	2011	Europe	12.60
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011	Europe	16.60
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/09/2011 12:50	4.15	12680.0	France	12/09/2011	9	12	2011	Europe	16.60
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/09/2011 12:50	4.95	12680.0	France	12/09/2011	9	12	2011	Europe	14.85

68006 rows × 14 columns

ANALYZE PURCHASING BEHAVIOR

- Prepare Transaction Data

```
# Group transactions by InvoiceNo and collect StockCode as lists
high_value_transactions = high_value_festive_data.groupby('InvoiceNo')['StockCode'].apply(list)
high_value_transactions
```

```
/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run
  and should_run_async(code)
```

InvoiceNo	StockCode
536365	[85123A, 71053, 84406B, 84029G, 84029E, 22752,...
536366	[22633, 22632]
536370	[22728, 22727, 22726, 21724, 21883, 10002, 217...
536372	[22632, 22633]
536373	[85123A, 71053, 84406B, 20679, 37370, 21871, 2...
...	...
C581460	[22197, 22107]
C581468	[21314, 22098]
C581499	[M]
C581568	[21258]
C581569	[84978, 20979]

1099 rows × 1 columns

ANALYZE PURCHASING BEHAVIOR

- Clean Transaction Data
- replace non-string values (e.g., NaN) in the transaction data with Unknown.

```python
def clean_transactions(transactions):
    cleaned_transactions = []
    for transaction in transactions:
        cleaned_transaction = [item if isinstance(item, str) else "Unknown" for item in transaction]
        cleaned_transactions.append(cleaned_transaction)
    return cleaned_transactions

high_value_transactions_cleaned = clean_transactions(high_value_transactions)
general_transactions_cleaned = clean_transactions(general_transactions)
```

ANALYZE PURCHASING BEHAVIOR

- Generate Association Rules

```python
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules

def get_association_rules(transactions, min_support=0.02):
    # Encode transactions
    te = TransactionEncoder()
    te_ary = te.fit(transactions).transform(transactions)
    df = pd.DataFrame(te_ary, columns=te.columns_)

    # Generate frequent itemsets
    frequent_itemsets = apriori(df, min_support=min_support, use_colnames=True)

    if frequent_itemsets.empty:
        print("No frequent itemsets found. Consider lowering min_support.")
        return pd.DataFrame()

    # Generate association rules
    rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
    return rules.sort_values('confidence', ascending=False)

# Generate rules for High-Value and General Customers
high_value_rules = get_association_rules(high_value_transactions_cleaned, min_support=0.015)
general_rules = get_association_rules(general_transactions_cleaned, min_support=0.015)
```

ANALYZE PURCHASING BEHAVIOR

- Generate Association Rules

```
# Display the number of rules
print(f"Number of High-Value Customer Rules: {len(high_value_rules)}")
print(f"Number of General Customer Rules: {len(general_rules)}")
```

```
Number of High-Value Customer Rules: 888
Number of General Customer Rules: 37590
```

FOCUS ON HIGH-VALUE CUSTOMER RULES

Identify Top Rules

- Sort the rules by lift to prioritize the most meaningful associations.

```
high_value_rules_sorted = high_value_rules.sort_values(by='lift', ascending=False)
high_value_rules_sorted = high_value_rules_sorted.head(10)  # Display top 10 rules
high_value_rules_sorted
```

```
/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the fu
  and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
850	(84029G, 71053)	(85123A, 84029E, 21730)	0.015469	0.016379	0.015469	1.000000	61.055556	0.015215	inf	0.999076
608	(85123A, 71053)	(21730, 84029G)	0.016379	0.015469	0.015469	0.944444	61.055556	0.015215	17.721565	1.000000
807	(85123A, 84029E, 21730)	(22752, 71053)	0.016379	0.015469	0.015469	0.944444	61.055556	0.015215	17.721565	1.000000
609	(21730, 84029G)	(85123A, 71053)	0.015469	0.016379	0.015469	1.000000	61.055556	0.015215	inf	0.999076
815	(85123A, 71053)	(21730, 22752, 84029E)	0.016379	0.015469	0.015469	0.944444	61.055556	0.015215	17.721565	1.000000
840	(21730, 84029E, 84029G)	(85123A, 71053)	0.015469	0.016379	0.015469	1.000000	61.055556	0.015215	inf	0.999076
851	(21730, 84029G)	(85123A, 84029E, 71053)	0.015469	0.016379	0.015469	1.000000	61.055556	0.015215	inf	0.999076
822	(21730, 71053)	(85123A, 22752, 84029E)	0.016379	0.015469	0.015469	0.944444	61.055556	0.015215	17.721565	1.000000
810	(21730, 22752, 84029E)	(85123A, 71053)	0.015469	0.016379	0.015469	1.000000	61.055556	0.015215	inf	0.999076
845	(85123A, 71053)	(21730, 84029E, 84029G)	0.016379	0.015469	0.015469	0.944444	61.055556	0.015215	17.721565	1.000000

REFINE GENERAL CUSTOMER RULES

Filter Rules by Confidence and Lift

- To reduce the number of rules and focus on the most significant ones, filter the rules based on confidence and lift

```
general_rules_filtered = general_rules[
    (general_rules['confidence'] > 0.7) & (general_rules['lift'] > 1.5)
]
print(f"Number of Filtered General Customer Rules: {len(general_rules_filtered)}")

```

```
Number of Filtered General Customer Rules: 20163
```

REFINE GENERAL CUSTOMER RULES

Identify Common Patterns

- Look for rules that appear frequently across General Customers to understand overall market trends.

```
[292]  1 general_rules_sorted = general_rules_filtered.sort_values(by='support', ascending=False)
       2 general_rules_sorted = general_rules_sorted.head(10) # Display top 10 rules
       3 general_rules_sorted
```

```
/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatical
  and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
2408	(22727)	(22726)	0.043792	0.043792	0.032457	0.741176	16.924983	0.030540	3.694441	0.984007
2409	(22726)	(22727)	0.043792	0.043792	0.032457	0.741176	16.924983	0.030540	3.694441	0.984007
2375	(22697)	(22699)	0.038640	0.038125	0.029366	0.760000	19.934595	0.027893	4.007814	0.988013
2374	(22699)	(22697)	0.038125	0.038640	0.029366	0.770270	19.934595	0.027893	4.184744	0.987483
1941	(DOT)	(22197)	0.036064	0.061309	0.026275	0.728571	11.883673	0.024064	3.458337	0.950116
9204	(22111, 22835)	(22112)	0.030397	0.069037	0.026275	0.864407	12.520997	0.024177	6.865855	0.948980
9205	(22111, 22112)	(22835)	0.037094	0.058733	0.026275	0.708333	12.060307	0.024096	3.227202	0.952412
11984	(22867, 22865)	(22866)	0.036064	0.051005	0.025760	0.714286	14.004329	0.023920	3.321484	0.963335
1254	(22083)	(22086)	0.032973	0.097888	0.025760	0.781250	7.981086	0.022532	4.123942	0.904529
11983	(22866, 22865)	(22867)	0.035033	0.065430	0.025760	0.735294	11.237842	0.023468	3.530597	0.944090

COMPARE UNIQUE PRODUCT ASSOCIATIONS BETWEEN HIGH-VALUE AND GENERAL CUSTOMERS

```python
# Compare unique product associations between High-Value and General Customers
unique_high_value_associations = high_value_rules_sorted[
    ~high_value_rules_sorted['antecedents'].isin(general_rules_sorted['antecedents'])
]

# Print unique associations
print("Unique High-Value Customer Holiday Product Associations:")
for idx, row in unique_high_value_associations.iterrows():
    antecedents = list(row['antecedents'])
    consequents = list(row['consequents'])
    print(f"If bought {antecedents}, likely to buy {consequents}")
```

```
Unique High-Value Customer Holiday Product Associations:
If bought ['84029G', '71053'], likely to buy ['85123A', '84029E', '21730']
If bought ['85123A', '71053'], likely to buy ['21730', '84029G']
If bought ['85123A', '84029E', '21730'], likely to buy ['22752', '71053']
If bought ['21730', '84029G'], likely to buy ['85123A', '71053']
If bought ['85123A', '71053'], likely to buy ['21730', '22752', '84029E']
If bought ['21730', '84029E', '84029G'], likely to buy ['85123A', '71053']
If bought ['21730', '84029G'], likely to buy ['85123A', '84029E', '71053']
If bought ['21730', '71053'], likely to buy ['85123A', '22752', '84029E']
If bought ['21730', '22752', '84029E'], likely to buy ['85123A', '71053']
If bought ['85123A', '71053'], likely to buy ['21730', '84029E', '84029G']
```

UNIQUE ASSOCIATIONS

```python
# Function to filter out nan values from descriptions
def clean_descriptions(descriptions):
    return {desc for desc in descriptions if pd.notna(desc)}

# Loop through each row in the unique_high_value_associations DataFrame
for idx, row in unique_high_value_associations.iterrows():
    antecedents = list(row['antecedents'])
    consequents = list(row['consequents'])

    # Print the association
    print(f"If bought {antecedents}, likely to buy {consequents}")

    # Find and print descriptions for antecedents
    print("Descriptions for antecedents:")
    for item in antecedents:
        descriptions = finditem(item, '')
        descriptions = clean_descriptions(descriptions)  # Clean descriptions
        print(f"  StockCode {item}: {', '.join(descriptions) if descriptions else 'No description available'}")

    # Find and print descriptions for consequents
    print("Descriptions for consequents:")
    for item in consequents:
        descriptions = finditem(item, '')
        descriptions = clean_descriptions(descriptions)  # Clean descriptions
        print(f"  StockCode {item}: {', '.join(descriptions) if descriptions else 'No description available'}")

    # Alternatively, use finditems for multiple items at once
    print("Descriptions for antecedents (using finditems):")
    descriptions = finditems(*antecedents)
    descriptions = clean_descriptions(descriptions)  # Clean descriptions
    print(f"  {', '.join(descriptions) if descriptions else 'No descriptions available'}")

    print("Descriptions for consequents (using finditems):")
    descriptions = finditems(*consequents)
    descriptions = clean_descriptions(descriptions)  # Clean descriptions
    print(f"  {', '.join(descriptions) if descriptions else 'No descriptions available'}")

    print("\n" + "="*50 + "\n")  # Add a separator for better readability
```

UNIQUE ASSOCIATIONS

```
/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please p
  and should_run_async(code)
If bought ['84029G', '71053'], likely to buy ['85123A', '84029E', '21730']
Descriptions for antecedents:
  StockCode 84029G: KNITTED UNION FLAG HOT WATER BOTTLE
  StockCode 71053: WHITE METAL LANTERN, WHITE MOROCCAN METAL LANTERN
Descriptions for consequents:
  StockCode 85123A: CREAM HANGING HEART T-LIGHT HOLDER, ?, wrongly marked carton 22804, WHITE HANGING HEART T-LIGHT HOLDER
  StockCode 84029E: RED WOOLLY HOTTIE WHITE HEART.
  StockCode 21730: GLASS STAR FROSTED T-LIGHT HOLDER
Descriptions for antecedents (using finditems):
  WHITE METAL LANTERN, WHITE MOROCCAN METAL LANTERN, KNITTED UNION FLAG HOT WATER BOTTLE
Descriptions for consequents (using finditems):
  WHITE HANGING HEART T-LIGHT HOLDER, GLASS STAR FROSTED T-LIGHT HOLDER, wrongly marked carton 22804, RED WOOLLY HOTTIE WHITE HEART., CREAM HANGING HEART T-LIGHT HOLDER, ?

==================================================

If bought ['85123A', '71053'], likely to buy ['21730', '84029G']
Descriptions for antecedents:
  StockCode 85123A: CREAM HANGING HEART T-LIGHT HOLDER, ?, wrongly marked carton 22804, WHITE HANGING HEART T-LIGHT HOLDER
  StockCode 71053: WHITE METAL LANTERN, WHITE MOROCCAN METAL LANTERN
Descriptions for consequents:
  StockCode 21730: GLASS STAR FROSTED T-LIGHT HOLDER
  StockCode 84029G: KNITTED UNION FLAG HOT WATER BOTTLE
Descriptions for antecedents (using finditems):
  WHITE HANGING HEART T-LIGHT HOLDER, wrongly marked carton 22804, WHITE MOROCCAN METAL LANTERN, WHITE METAL LANTERN, CREAM HANGING HEART T-LIGHT HOLDER, ?
Descriptions for consequents (using finditems):
  GLASS STAR FROSTED T-LIGHT HOLDER, KNITTED UNION FLAG HOT WATER BOTTLE

==================================================

If bought ['85123A', '84029E', '21730'], likely to buy ['22752', '71053']
Descriptions for antecedents:
  StockCode 85123A: CREAM HANGING HEART T-LIGHT HOLDER, ?, wrongly marked carton 22804, WHITE HANGING HEART T-LIGHT HOLDER
  StockCode 84029E: RED WOOLLY HOTTIE WHITE HEART.
  StockCode 21730: GLASS STAR FROSTED T-LIGHT HOLDER
Descriptions for consequents:
  StockCode 22752: SET 7 BABUSHKA NESTING BOXES
  StockCode 71053: WHITE METAL LANTERN, WHITE MOROCCAN METAL LANTERN
Descriptions for antecedents (using finditems):
  WHITE HANGING HEART T-LIGHT HOLDER, GLASS STAR FROSTED T-LIGHT HOLDER, wrongly marked carton 22804, RED WOOLLY HOTTIE WHITE HEART., CREAM HANGING HEART T-LIGHT HOLDER, ?
Descriptions for consequents (using finditems):
  SET 7 BABUSHKA NESTING BOXES, WHITE METAL LANTERN, WHITE MOROCCAN METAL LANTERN

==================================================
```

OBSERVATIONS

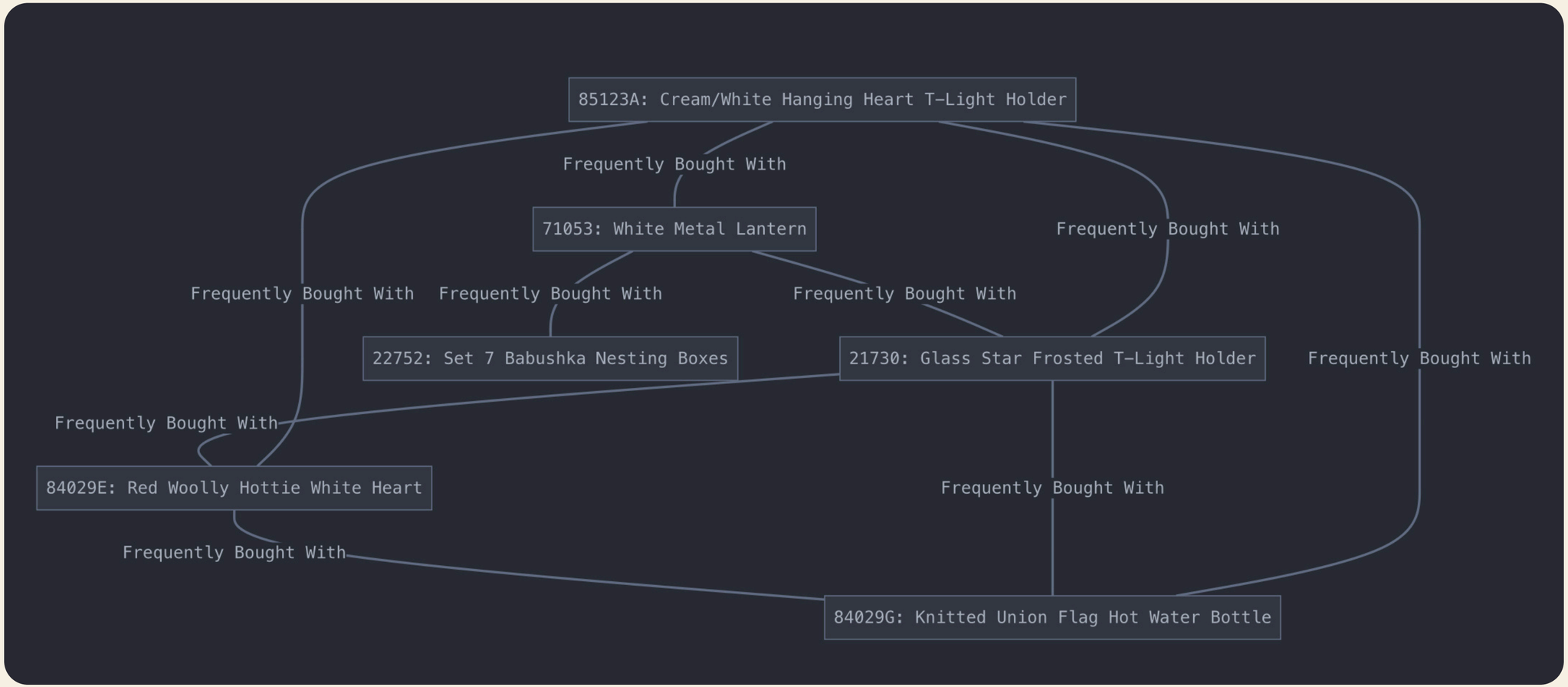
Several products seem to be frequently associated

1. 85123A (Cream/White Hanging Heart T-Light Holder) 
2. 71053 (White Metal/Moroccan Lantern) 
3. 21730 (Glass Star Frosted T-Light Holder) 
4. 84029G (Knitted Union Flag Hot Water Bottle) 
5. 84029E (Red Woolly Hottie White Heart) 
6. 22752 (Set 7 Babushka Nesting Boxes) 

OBSERVATIONS

The analysis suggests these products have strong co-purchase relationships.
For example, if someone buys a Glass Star Frosted T-Light Holder (21730) and a Knitted Union Flag Hot Water Bottle (84029G), they are likely to also purchase a Cream Hanging Heart T-Light Holder (85123A) and a White Metal Lantern (71053).

UNIQUE ASSOCIATIONS



SEGMENT BY REGION

- Analyze High-Value Customer Behavior by Continent

```python
def analyze_continent_behavior(continent):
    continent_high_value = high_value_festive_data[
        high_value_festive_data['Continent'] == continent
    ]
    return continent_high_value['Description'].value_counts().head()

# Compare holiday product preferences across continents
for continent in ['Europe', 'North America', 'Asia']:
    print(f"\n{continent} High-Value Customer Top Products:")
    print(analyze_continent_behavior(continent))
```

```
Europe High-Value Customer Top Products:
Description
WHITE HANGING HEART T-LIGHT HOLDER    104
REGENCY CAKESTAND 3 TIER               98
PAPER CHAIN KIT 50'S CHRISTMAS         94
CHOCOLATE HOT WATER BOTTLE             83
PAPER CHAIN KIT VINTAGE CHRISTMAS      77
Name: count, dtype: int64

North America High-Value Customer Top Products:
Series([], Name: count, dtype: int64)

Asia High-Value Customer Top Products:
Description
PARTY BUNTING                          2
MINI WOODEN HAPPY BIRTHDAY GARLAND     2
WOODEN HAPPY BIRTHDAY GARLAND          2
PAPER BUNTING COLOURED LACE            2
FELTCRAFT PRINCESS CHARLOTTE DOLL      2
```

KEY INSIGHTS FROM THE HIGH-VALUE CUSTOMER PRODUCT DATA

Europe (Top 5 Products)

1

WHITE HANGING HEART T-LIGHT HOLDER (104 purchases)

2

REGENCY CAKESTAND 3 TIER (98 purchases)

3

PAPER CHAIN KIT 50'S CHRISTMAS (94 purchases)

4

CHOCOLATE HOT WATER BOTTLE (83 purchases)

5

PAPER CHAIN KIT VINTAGE CHRISTMAS (77 purchases)

KEY INSIGHTS FROM THE HIGH-VALUE CUSTOMER PRODUCT DATA

Asia (Top 5 Products)

1

PARTY BUNTING (2 purchases)

2

MINI WOODEN HAPPY BIRTHDAY GARLAND (2 purchases)

3

WOODEN HAPPY BIRTHDAY GARLAND (2 purchases)

4

PAPER BUNTING COLOURED LACE (2 purchases)

5

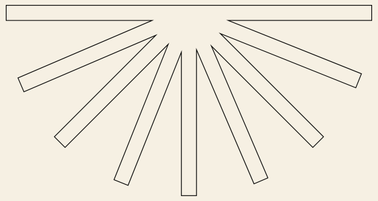
FELTCRAFT PRINCESS CHARLOTTE DOLL (2 purchases)

OBSERVATIONS

1. Europe shows significantly higher purchase volumes
2. European customers seem to prefer home decor and seasonal items.
3. Asian data shows very low purchase numbers.
4. North America has no data in this analysis.

MEMBERS

653020213-2	นางสาว พรรณรมณ ราชคมน์
653020217-4	นาย รัชชานนท์ พันกาฬสินธุ์
653020573-2	นางสาว พรวลัย ฟ็อกซ์ออล



THANK YOU

for your attention!