Darshan UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Task-4 Apply Apriori on Online Retail Dataset

137 | Vishal Baraiya | 23010101014

Load the Dataset

In [1]:
```
import pandas as pd
from itertools import combinations
```

In [2]:
```
OnlineRetail = pd.read_csv("D:\\VS_CODES\\DataMining\\ProjectDataMining\\Dataset\\O
OnlineRetail
```

Out[2]:

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

541909 rows × 8 columns

copy it into a new DataFrame

In [3]:
```
df = OnlineRetail.copy()
df
```

Out[3]:

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

541909 rows × 8 columns

Now you can clean, filter, and transform 'df' without changing 'OnlineRetail'

1. Keep only necessary columns

In [4]:
```
df = df[['InvoiceNo', 'Description', 'Quantity']]
df
```

Out[4]:

	InvoiceNo	Description	Quantity
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	WHITE METAL LANTERN	6
2	536365	CREAM CUPID HEARTS COAT HANGER	8
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6
...	...	...	...
541904	581587	PACK OF 20 SPACEBOY NAPKINS	12
541905	581587	CHILDREN'S APRON DOLLY GIRL	6
541906	581587	CHILDRENS CUTLERY DOLLY GIRL	4
541907	581587	CHILDRENS CUTLERY CIRCUS PARADE	4
541908	581587	BAKING SET 9 PIECE RETROSPOT	3

541909 rows × 3 columns

2. Remove rows with negative quantity

In [5]:
```
df = df[df['Quantity'] > 0]
df
```

Out[5]:

	InvoiceNo	Description	Quantity
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	WHITE METAL LANTERN	6
2	536365	CREAM CUPID HEARTS COAT HANGER	8
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6
...	...	...	...
541904	581587	PACK OF 20 SPACEBOY NAPKINS	12
541905	581587	CHILDREN'S APRON DOLLY GIRL	6
541906	581587	CHILDRENS CUTLERY DOLLY GIRL	4
541907	581587	CHILDRENS CUTLERY CIRCUS PARADE	4
541908	581587	BAKING SET 9 PIECE RETROSPOT	3

531285 rows × 3 columns

3. Remove nulls in InvoiceNo and Description

In [6]:
```
df.dropna(subset=['InvoiceNo', 'Description'], inplace=True)
df
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_23432\3517667345.py:1: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  df.dropna(subset=['InvoiceNo', 'Description'], inplace=True)
```

Out[6]:

	InvoiceNo	Description	Quantity
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	WHITE METAL LANTERN	6
2	536365	CREAM CUPID HEARTS COAT HANGER	8
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6
...	...	...	...
541904	581587	PACK OF 20 SPACEBOY NAPKINS	12
541905	581587	CHILDREN'S APRON DOLLY GIRL	6
541906	581587	CHILDRENS CUTLERY DOLLY GIRL	4
541907	581587	CHILDRENS CUTLERY CIRCUS PARADE	4
541908	581587	BAKING SET 9 PIECE RETROSPOT	3

530693 rows × 3 columns

4. Group by InvoiceNo and Description, then sum Quantity

In [7]:
```
basket = df.groupby(['InvoiceNo', 'Description'])['Quantity'].sum().unstack().filln
```

5. Convert to 1 if item was bought, else 0

In [8]:
```
df_item = basket.applymap(lambda x: 1 if x > 0 else 0)
print(df_item.head())
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_23432\1542778015.py:1: FutureWarning: Dat
aFrame.applymap has been deprecated. Use DataFrame.map instead.
  df_item = basket.applymap(lambda x: 1 if x > 0 else 0)
```

Description	4 PURPLE FLOCK DINNER CANDLES	50'S CHRISTMAS GIFT BAG LARGE	\
InvoiceNo			
536365	0	0	
536366	0	0	
536367	0	0	
536368	0	0	
536369	0	0	

Description	DOLLY GIRL BEAKER	I LOVE LONDON MINI BACKPACK	\
InvoiceNo			
536365	0	0	
536366	0	0	
536367	0	0	
536368	0	0	
536369	0	0	

Description	I LOVE LONDON MINI RUCKSACK	NINE DRAWER OFFICE TIDY	\
InvoiceNo			
536365	0	0	
536366	0	0	
536367	0	0	
536368	0	0	
536369	0	0	

Description	OVAL WALL MIRROR DIAMANTE	RED SPOT GIFT BAG LARGE	\
InvoiceNo			
536365	0	0	
536366	0	0	
536367	0	0	
536368	0	0	
536369	0	0	

Description	SET 2 TEA TOWELS I LOVE LONDON	SPACEBOY BABY GIFT SET	...	\
InvoiceNo			...	
536365	0	0	...	
536366	0	0	...	
536367	0	0	...	
536368	0	0	...	
536369	0	0	...	

Description	returned	taig adjust	test	to push order througha s stock was	\
InvoiceNo					
536365	0	0	0	0	
536366	0	0	0	0	
536367	0	0	0	0	
536368	0	0	0	0	
536369	0	0	0	0	

Description	website fixed	wrongly coded 20713	wrongly coded 23343	\
InvoiceNo				
536365	0	0	0	
536366	0	0	0	
536367	0	0	0	
536368	0	0	0	
536369	0	0	0	

```
Description  wrongly marked  wrongly marked 23343  \
InvoiceNo
536365                   0                     0
536366                   0                     0
536367                   0                     0
536368                   0                     0
536369                   0                     0

Description  wrongly sold (22719) barcode
InvoiceNo
536365                                  0
536366                                  0
536367                                  0
536368                                  0
536369                                  0

[5 rows x 4077 columns]
```

Define Apriori Function

This function finds frequent itemsets of size 1, 2, and 3 with minimum support.

In [9]:
```python
def find_frequent_itemsets(df,min_support):
    n = len(df)
    result = []

    for k in [1,2,3]: # for 1-item, 2-item, 3-item
        for items in combinations(df.columns,k):
            mask = df[list(items)].all(axis = 1)
            support = mask.sum()/n
            if support >= min_support:
                result.append((frozenset(items),round(support,2)))

    return result
```

Run Apriori

Set `min_support = 0.6` and display the frequent itemsets.

In [10]:
```python
frequent_itemsets = find_frequent_itemsets(df,min_support=0.6)

for itemset, support in frequent_itemsets:
    print(f"{set(itemset)} -> support : {support}")
```

```
{'InvoiceNo'} -> support : 1.0
{'Description'} -> support : 1.0
{'Quantity'} -> support : 1.0
{'InvoiceNo', 'Description'} -> support : 1.0
{'Quantity', 'InvoiceNo'} -> support : 1.0
{'Quantity', 'Description'} -> support : 1.0
{'Quantity', 'InvoiceNo', 'Description'} -> support : 1.0
```

Display as a DataFrame

In [11]:
```
result_df = pd.DataFrame(frequent_itemsets, columns=['Itemset','Support'])
result_df
```

Out[11]:

	Itemset	Support
0	(InvoiceNo)	1.0
1	(Description)	1.0
2	(Quantity)	1.0
3	(InvoiceNo, Description)	1.0
4	(Quantity, InvoiceNo)	1.0
5	(Quantity, Description)	1.0
6	(Quantity, InvoiceNo, Description)	1.0

In [ ]: