Roll No - 130

Enrollment No - 24010101694

Lab - 7 (Part 2)

Karan Sonagara

Step 1: Load the Dataset

Load the `Tdata.csv` file and display the first few rows.

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
df=pd.read_csv("Tdata.csv")
df
```

Out[4]:

	Transaction	bread	butter	coffee	eggs	jam	milk
0	T1	1	1	0	0	0	1
1	T2	1	1	0	0	1	0
2	T3	1	0	0	1	0	1
3	T4	1	1	0	0	0	1
4	T5	1	0	1	0	0	0
5	T6	0	0	1	1	1	0

Step 2: Drop the 'Transaction' Column

We're only interested in the items (not the transaction IDs).

In [7]:

```
df = df.drop("Transaction",axis=1)
df
```

Out[7]:

	bread	butter	coffee	eggs	jam	milk
0	1	1	0	0	0	1
1	1	1	0	0	1	0
2	1	0	0	1	0	1
3	1	1	0	0	0	1
4	1	0	1	0	0	0
5	0	0	1	1	1	0

Step 3: Count Single Items

See how many transactions include each item.

In [40]:

```
df.sum()
```

Out[40]:

```
Transaction    T1T2T3T4T5T6
bread                     5
butter                    3
coffee                    2
eggs                      2
jam                       2
milk                      3
dtype: object
```

Step 4: Define Apriori Function

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

In [43]:

```
from itertools import combinations

def findf(df,min_support):
    n = len(df)
    result = []


    for k in [1,2,3]:
        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


finals = findf(df, min_support=0.5)

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

```
{'Transaction'} -> support: 1.0
{'bread'} -> support: 0.83
{'butter'} -> support: 0.5
{'milk'} -> support: 0.5
{'Transaction', 'bread'} -> support: 0.83
{'butter', 'Transaction'} -> support: 0.5
{'Transaction', 'milk'} -> support: 0.5
{'butter', 'bread'} -> support: 0.5
{'milk', 'bread'} -> support: 0.5
{'butter', 'Transaction', 'bread'} -> support: 0.5
{'Transaction', 'milk', 'bread'} -> support: 0.5
```

Step 5: Run Apriori

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

In [52]:

```
finals = findf(df, min_support=0.6)

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

```
{'Transaction'} -> support: 1.0
{'bread'} -> support: 0.83
{'Transaction', 'bread'} -> support: 0.83
```

Step 6 Display as a DataFrame

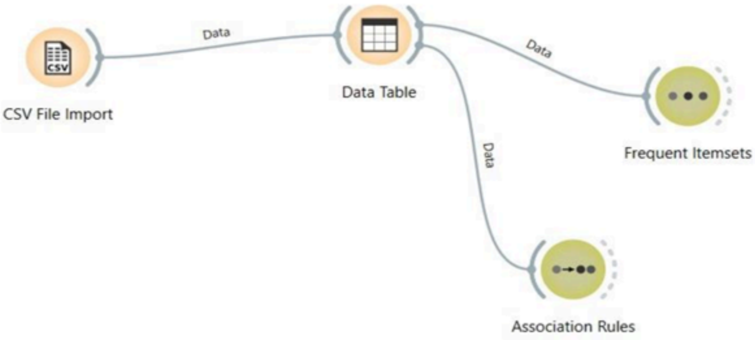
In [50]:

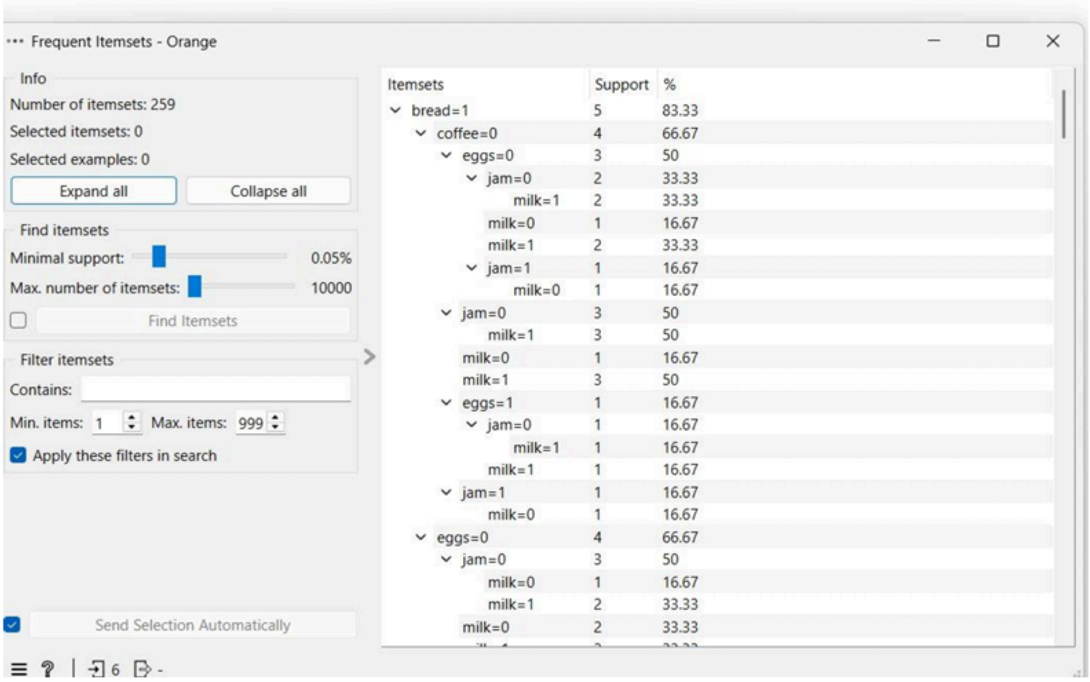
```
resulr = pd.DataFrame(finals,columns=["Itemset","Support"])
resulr
```

Out[50]:

	Itemset	Support
0	(Transaction)	1.00
1	(bread)	0.83
2	(butter)	0.50
3	(milk)	0.50
4	(Transaction, bread)	0.83
5	(butter, Transaction)	0.50
6	(Transaction, milk)	0.50
7	(butter, bread)	0.50
8	(milk, bread)	0.50
9	(butter, Transaction, bread)	0.50
10	(Transaction, milk, bread)	0.50

Orange Tool : - >Generate Same Frequent Patterns in Orange tools





Info

Number of itemsets: 259

Selected itemsets: 0

Selected examples: 0

Find itemsets

Minimal support: 0.05%

Max. number of itemsets: 10000

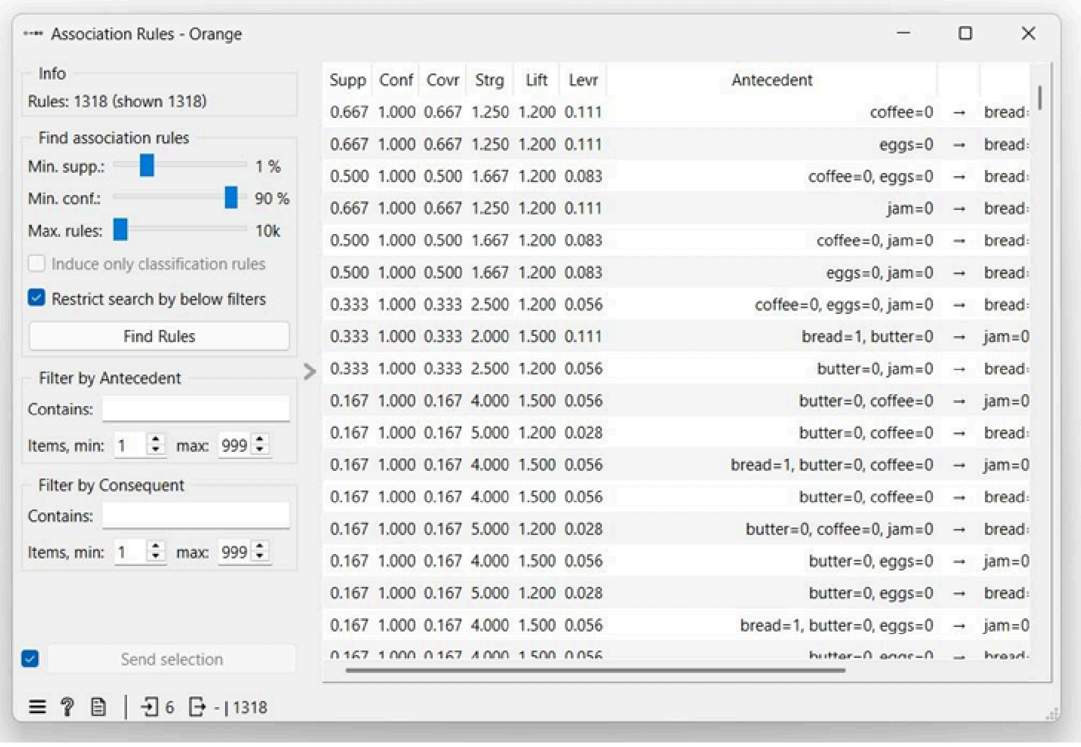
Filter itemsets

Contains:

Min. items: 1 Max. items: 999

☑ Apply these filters in search

Itemsets	Support	%
⌄ bread=1	5	83.33
⌄ coffee=0	4	66.67
⌄ eggs=0	3	50
⌄ jam=0	2	33.33
milk=1	2	33.33
milk=0	1	16.67
milk=1	2	33.33
⌄ jam=1	1	16.67
milk=0	1	16.67
⌄ jam=0	3	50
milk=1	3	50
milk=0	1	16.67
milk=1	3	50
⌄ eggs=1	1	16.67
⌄ jam=0	1	16.67
milk=1	1	16.67
milk=1	1	16.67
⌄ jam=1	1	16.67
milk=0	1	16.67
⌄ eggs=0	4	66.67
⌄ jam=0	3	50
milk=0	1	16.67
milk=1	2	33.33
milk=0	2	33.33



Association Rules - Orange

Info

Rules: 1318 (shown 1318)

Find association rules

Min. supp.: 1 %

Min. conf.: 90 %

Max. rules: 10k

☐ Induce only classification rules

☑ Restrict search by below filters

Find Rules

Filter by Antecedent

Contains:

Items, min: 1 max: 999

Filter by Consequent

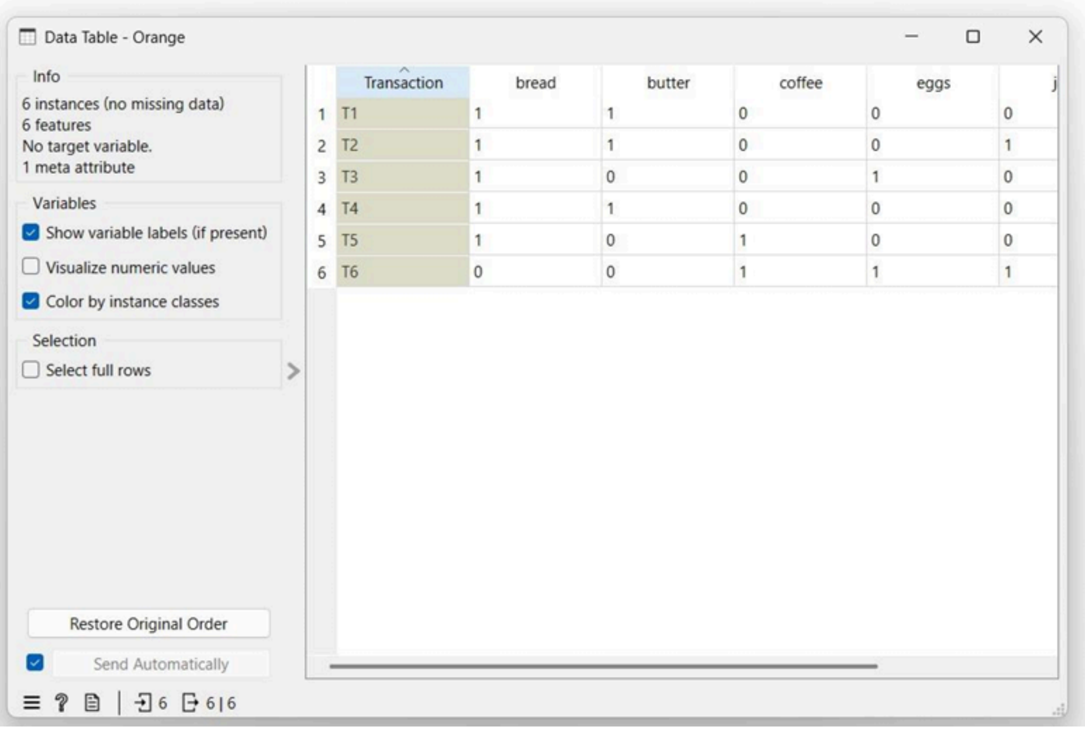
Contains:

Items, min: 1 max: 999

☑ Send selection

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		
0.667	1.000	0.667	1.250	1.200	0.111	coffee=0	→	bread=
0.667	1.000	0.667	1.250	1.200	0.111	eggs=0	→	bread=
0.500	1.000	0.500	1.667	1.200	0.083	coffee=0, eggs=0	→	bread=
0.667	1.000	0.667	1.250	1.200	0.111	jam=0	→	bread=
0.500	1.000	0.500	1.667	1.200	0.083	coffee=0, jam=0	→	bread=
0.500	1.000	0.500	1.667	1.200	0.083	eggs=0, jam=0	→	bread=
0.333	1.000	0.333	2.500	1.200	0.056	coffee=0, eggs=0, jam=0	→	bread=
0.333	1.000	0.333	2.000	1.500	0.111	bread=1, butter=0	→	jam=0
0.333	1.000	0.333	2.500	1.200	0.056	butter=0, jam=0	→	bread=
0.167	1.000	0.167	4.000	1.500	0.056	butter=0, coffee=0	→	jam=0
0.167	1.000	0.167	5.000	1.200	0.028	butter=0, coffee=0	→	bread=
0.167	1.000	0.167	4.000	1.500	0.056	bread=1, butter=0, coffee=0	→	jam=0
0.167	1.000	0.167	4.000	1.500	0.056	butter=0, coffee=0	→	bread=
0.167	1.000	0.167	5.000	1.200	0.028	butter=0, coffee=0, jam=0	→	bread=
0.167	1.000	0.167	4.000	1.500	0.056	butter=0, eggs=0	→	jam=0
0.167	1.000	0.167	5.000	1.200	0.028	butter=0, eggs=0	→	bread=
0.167	1.000	0.167	4.000	1.500	0.056	bread=1, butter=0, eggs=0	→	jam=0
0.167	1.000	0.167	4.000	1.500	0.056	butter=0, eggs=0	→	bread=

6 | - | 1318



Data Table - Orange

Info

6 instances (no missing data)
6 features
No target variable.
1 meta attribute

Variables

☑ Show variable labels (if present)

☐ Visualize numeric values

☑ Color by instance classes

Selection

☐ Select full rows

Restore Original Order

☑ Send Automatically

	Transaction	bread	butter	coffee	eggs	j
1	T1	1	1	0	0	0
2	T2	1	1	0	0	1
3	T3	1	0	0	1	0
4	T4	1	1	0	0	0
5	T5	1	0	1	0	0
6	T6	0	0	1	1	1

6 | 6 | 6