Data Mining

Lab - 7 (Part 2)

Raj Vekariya | 23010101298

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
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(columns='Transaction')
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 [8]:

```
df.sum()
```

Out[8]:

```
bread     5
butter    3
coffee    2
eggs      2
jam       2
milk      3
dtype: int64
```

Step 4: Define Apriori Function

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

In [13]:

```
from itertools import combinations
def AprioriFun(df, minSupport):
    n=len(df)
    result=[]

    for i in [1,2,3]:
        for items in combinations(df.columns,i):
            mask=df[list(items)].all(axis=1)
            support=mask.sum()/n
            print(f"{frozenset(items)}->{round(support,2)}")
            if support>=minSupport:
                result.append((frozenset(items),round(support,2)))

    return result
```

Step 5: Run Apriori

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

In [15]:

```
frequent_itemsets=AprioriFun(df,minSupport=0.5)

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

```
frozenset({'bread'})->0.83
frozenset({'butter'})->0.5
frozenset({'coffee'})->0.33
frozenset({'eggs'})->0.33
frozenset({'jam'})->0.33
frozenset({'milk'})->0.5
frozenset({'butter', 'bread'})->0.5
frozenset({'coffee', 'bread'})->0.17
frozenset({'eggs', 'bread'})->0.17
frozenset({'jam', 'bread'})->0.17
frozenset({'milk', 'bread'})->0.5
frozenset({'butter', 'coffee'})->0.0
frozenset({'butter', 'eggs'})->0.0
frozenset({'butter', 'jam'})->0.17
frozenset({'milk', 'butter'})->0.33
frozenset({'eggs', 'coffee'})->0.17
frozenset({'coffee', 'jam'})->0.17
frozenset({'milk', 'coffee'})->0.0
frozenset({'eggs', 'jam'})->0.17
frozenset({'milk', 'eggs'})->0.17
frozenset({'milk', 'jam'})->0.0
frozenset({'butter', 'coffee', 'bread'})->0.0
frozenset({'butter', 'bread', 'eggs'})->0.0
frozenset({'butter', 'jam', 'bread'})->0.17
frozenset({'milk', 'butter', 'bread'})->0.33
frozenset({'eggs', 'coffee', 'bread'})->0.0
frozenset({'coffee', 'jam', 'bread'})->0.0
frozenset({'milk', 'coffee', 'bread'})->0.0
frozenset({'eggs', 'jam', 'bread'})->0.0
frozenset({'milk', 'eggs', 'bread'})->0.17
frozenset({'milk', 'jam', 'bread'})->0.0
frozenset({'butter', 'coffee', 'eggs'})->0.0
frozenset({'butter', 'coffee', 'jam'})->0.0
frozenset({'milk', 'butter', 'coffee'})->0.0
frozenset({'butter', 'jam', 'eggs'})->0.0
frozenset({'milk', 'butter', 'eggs'})->0.0
frozenset({'milk', 'butter', 'jam'})->0.0
frozenset({'eggs', 'coffee', 'jam'})->0.17
frozenset({'milk', 'eggs', 'coffee'})->0.0
frozenset({'milk', 'coffee', 'jam'})->0.0
frozenset({'milk', 'eggs', 'jam'})->0.0
{'bread'}->support:0.83
{'butter'}->support:0.5
{'milk'}->support:0.5
{'butter', 'bread'}->support:0.5
{'milk', 'bread'}->support:0.5
```

Step 6 Display as a DataFrame

In [16]:

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

Out[16]:

	Itemset	Support
0	(bread)	0.83
1	(butter)	0.50
2	(milk)	0.50
3	(butter, bread)	0.50
4	(milk, bread)	0.50

In [ ]:

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

In [ ]:

Extra : - > Define Apriori Function without itertools

In [ ]:

In [ ]: