MACHINE LEARNING LAB

EXERCISE 10

Aim :

Implement Apriori or Fp Tree from scratch on the given dataset. You must do preprocessing on this data.

Algorithm :

1. Read transaction data from CSV file and preprocess it, filling missing values with zeros.
2. Prompt user to input minimum support value.
3. Define functions for Apriori algorithm: a. `frequency_count` : Count the frequency of individual items in transactions. b. `Combination_Count` : Generate new combinations of items and count their frequency, considering the minimum support. c. `Apriori` : Execute the Apriori algorithm iteratively to generate frequent itemsets.
4. Execute Apriori operations with the provided minimum support and transaction data.
5. Print the final patterns along with their support count.

Code and Output :

In [8]:

```
import pandas as pd
import numpy as np
```

In [9]:

```
#Preprocessing:
df = pd.read_csv(r"C:\Users\TEJU\Downloads\Market_Basket_Optimisation.csv", header =

df = df.fillna(0)
t = []
for i in range(0,7501):
    local_t = []
    for j in range(0,20):
        item = str(df.iloc[i,j])
        if item == '0':
            break
        local_t.append(item)
    t.append(local_t)
```

In [10]:

```
df.head()
```

Out[10]:

	0	1	2	3	4	5	6	7	8	9	10	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	gre t
1	burgers	meatballs	eggs	0	0	0	0	0	0	0	0	
2	chutney	0	0	0	0	0	0	0	0	0	0	
3	turkey	avocado	0	0	0	0	0	0	0	0	0	

	0	1	2	3	4	5	6	7	8	9	10
4	mineral water	milk	energy bar	whole wheat rice	green tea	0	0	0	0	0	0

In [11]:

```
min_sup = int(input("Enter Minimum Support Value: "))
```

```
Enter Minimum Support Value: 50
```

Functions for Apriori Algorithm

In [12]:

```
def frequency_count(min_sup,t):
    freq_count = {}
    for i in range(0,len(t)):
        unique = {}
        for j in range(0,len(t[i])):
            unique[t[i][j]] = 1
        for key in unique:
            try:
                freq_count[key] += 1
            except:
                freq_count[key] = 1
    return freq_count

def Combination_Count(freq_count,min_sup,L,t):
    new_freq_count = {}
    for key1 in freq_count:
        for key2 in freq_count:
            l1 = key1.split(",")
            l2 = key2.split(",")
            unique = set(l1).union(set(l2))
            unique = list(unique)
            unique = sorted(unique)
            itemList = ""
            for item in unique:
                if len(itemList) == 0:
                    itemList = itemList + item
                else:
                    itemList = itemList + "," + item
            if (itemList in new_freq_count) or (len(unique) != L):
                continue
            freq = 0
            for i in range(0,len(t)):
                checker = True
                for item in unique:
                    checker = checker and (item in t[i])
                    if checker == False:
                        break
                if checker == True:
                    freq += 1
            if freq >= min_sup:
                new_freq_count[itemList] = freq
    return new_freq_count

def Apriori(freq_count,min_sup,t):
    L = 1
    while(True):
        L += 1
        new_freq_count = Combination_Count(freq_count,min_sup,L,t)
        if len(new_freq_count) == 0 :
            break
        freq_count = new_freq_count
    return freq_count
```

In [13]:

```
#Calling for algorithm
freq_count = frequency_count(min_sup,t)
final_patterns = Apriori(freq_count,min_sup,t)
```

In [14]:

```
print("Itemset : Minimum Support")
for key in final_patterns:
    print(key + " : " + str(final_patterns[key]))
```

```
Itemset : Minimum Support
milk,mineral water,shrimp : 59
frozen vegetables,mineral water,shrimp : 54
mineral water,shrimp,spaghetti : 64
chocolate,mineral water,shrimp : 57
eggs,green tea,mineral water : 52
green tea,mineral water,spaghetti : 62
chocolate,green tea,mineral water : 52
eggs,french fries,green tea : 53
chocolate,green tea,spaghetti : 53
mineral water,salmon,spaghetti : 51
frozen smoothie,mineral water,spaghetti : 51
milk,mineral water,olive oil : 64
mineral water,olive oil,spaghetti : 77
chocolate,mineral water,olive oil : 62
ground beef,mineral water,olive oil : 50
burgers,eggs,mineral water : 59
burgers,milk,mineral water : 52
eggs,milk,mineral water : 98
eggs,french fries,mineral water : 52
eggs,frozen vegetables,mineral water : 68
eggs,mineral water,spaghetti : 107
chocolate,eggs,mineral water : 101
eggs,mineral water,pancakes : 59
eggs,ground beef,mineral water : 76
cake,eggs,mineral water : 54
french fries,milk,mineral water : 62
milk,mineral water,soup : 64
frozen vegetables,milk,mineral water : 83
milk,mineral water,spaghetti : 118
chocolate,milk,mineral water : 105
chicken,milk,mineral water : 50
milk,mineral water,pancakes : 59
ground beef,milk,mineral water : 83
french fries,mineral water,spaghetti : 76
chocolate,french fries,mineral water : 64
mineral water,soup,spaghetti : 56
frozen vegetables,mineral water,spaghetti : 90
chocolate,frozen vegetables,mineral water : 73
frozen vegetables,ground beef,mineral water : 69
cooking oil,mineral water,spaghetti : 57
chocolate,mineral water,spaghetti : 119
chicken,mineral water,spaghetti : 52
mineral water,spaghetti,tomatoes : 70
mineral water,pancakes,spaghetti : 86
ground beef,mineral water,spaghetti : 128
cake,mineral water,spaghetti : 53
chicken,chocolate,mineral water : 57
chocolate,mineral water,pancakes : 70
```

```
chocolate,ground beef,mineral water : 82
ground beef,mineral water,pancakes : 56
ground beef,herb & pepper,mineral water : 50
milk,olive oil,spaghetti : 54
chocolate,olive oil,spaghetti : 53
burgers,eggs,french fries : 68
burgers,eggs,spaghetti : 54
eggs,french fries,milk : 54
eggs,frozen vegetables,milk : 55
eggs,milk,spaghetti : 67
chocolate,eggs,milk : 69
eggs,french fries,spaghetti : 60
chocolate,eggs,french fries : 63
chocolate,eggs,spaghetti : 79
eggs,pancakes,spaghetti : 50
eggs,ground beef,spaghetti : 67
chocolate,french fries,milk : 55
frozen vegetables,milk,spaghetti : 62
chocolate,frozen vegetables,milk : 60
chocolate,milk,spaghetti : 82
ground beef,milk,spaghetti : 73
chocolate,french fries,spaghetti : 60
chocolate,frozen vegetables,spaghetti : 59
frozen vegetables,spaghetti,tomatoes : 50
frozen vegetables,ground beef,spaghetti : 65
chocolate,pancakes,spaghetti : 51
chocolate,ground beef,spaghetti : 69
```

Result :

Therefore, we have implemented associate rule mining with the Apriori algorithm from scratch using the given maket basket optimisation dataset.