Ex 1: Store data

Cho dữ liệu store data trong tập tin store_data.

Yêu cầu: Áp dụng thuật toán ECLAT để tính toán mức độ kết hợp giữa các item

1. Chuẩn hóa dữ liệu
2. Áp dụng ECLAT, Tìm kết quả
3. Cho biết 10 nhóm có độ kết hợp cao nhất
4. Tìm kiếm thông tin từ kết quả: trong thông tin kết quả có 'milk' không? Nếu có thì 'milk' kết hợp với item nào?"

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:
```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice_2023/Chapter12_ECLAT/'
```

In [3]:
```
import sys
from collections import defaultdict
import random
```

In [4]:
```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
```

In [5]:
```
# source code from: http://codegist.net/snippet/python/eclatpy_evertheylen_python
def tidlists(transactions):
    tl = defaultdict(set)
    for tid, t in enumerate(transactions):
        for item in t:
            tl[item].add(tid)
    return list(tl.items())

class IntersectAll:
    def __and__(self, other):
        return other
IntersectAll = IntersectAll()
```

In [6]:
```
def eclat(items, minsup=0, minlen=1):
    frequent_itemsets = {(): IntersectAll}
    def recurse(items, prefix):
        while len(items) > 0:
            item, item_tidlist = items.pop()
            l = prefix + (item,)  # l is the (ordered) tuple of items we are looking for
            new_tidlist = frequent_itemsets[prefix] & item_tidlist
            if len(new_tidlist) >= minsup:  # add frequent_itemsets to the new frequent_itemsets
                frequent_itemsets[l] = new_tidlist

            # define the new l-conditional database
            new_items = []
            for new_item, _item_tidlist in items:
                new_item_tidlist = _item_tidlist & item_tidlist
                if len(new_item_tidlist) >= minsup:
                    new_items.append((new_item, new_item_tidlist))

            # recurse, with l as prefix
            recurse(new_items, l)

    recurse(items.copy(), ())
    return {k: len(v) for k, v in frequent_itemsets.items() if len(k) >= minlen}
```

In [7]:
```
# load dữ liệu
store_data = pd.read_csv('store_data.csv', header= None)
```

In [8]:
```
# store_data.info()
```

In [9]:
```
store_data.head()
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	1
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spinac
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

In [10]:
```
records = []
for i in range(0, store_data.shape[0]):
    records.append([str(store_data.values[i,j])
                    for j in range(0, store_data.shape[1])])
```

In [11]:
```
records[0]
```

Out[11]:
```
['shrimp',
 'almonds',
 'avocado',
 'vegetables mix',
 'green grapes',
 'whole weat flour',
 'yams',
 'cottage cheese',
 'energy drink',
 'tomato juice',
 'low fat yogurt',
 'green tea',
 'honey',
 'salad',
 'mineral water',
 'salmon',
 'antioxydant juice',
 'frozen smoothie',
 'spinach',
 'olive oil']
```

In [12]:
```
tl = tidlists(records)
tl[0][0]
```

Out[12]:
```
'shrimp'
```

In [13]:
```
len(tl)
```

Out[13]:
```
121
```

In [14]:
```
for i in range(len(tl)-1):
    if tl[i][0] == 'nan':
        print(i)
        del tl[i]
```

```
23
```

In [15]:
```
result = eclat(tl, minsup=60, minlen=3)
```

In [16]:
```
# get 10 first elements
list(result.items())[:10]
```

Out[16]:
```
[(('ground beef', 'chocolate', 'spaghetti'), 69),
 (('ground beef', 'chocolate', 'mineral water'), 82),
 (('ground beef', 'spaghetti', 'frozen vegetables'), 65),
 (('ground beef', 'spaghetti', 'milk'), 73),
 (('ground beef', 'spaghetti', 'eggs'), 67),
 (('ground beef', 'spaghetti', 'mineral water'), 128),
 (('ground beef', 'frozen vegetables', 'mineral water'), 69),
 (('ground beef', 'milk', 'mineral water'), 83),
 (('ground beef', 'eggs', 'mineral water'), 76),
 (('pancakes', 'chocolate', 'mineral water'), 70)]
```

In [17]:
```
type(result)
```

Out[17]:
```
dict
```

In [18]:
```
sorted_d = sorted((value, key) for (key,value) in result.items())
sorted_d[:10]
```

Out[18]:
```
[(60, ('chocolate', 'frozen vegetables', 'milk')),
 (60, ('chocolate', 'spaghetti', 'french fries')),
 (60, ('spaghetti', 'french fries', 'eggs')),
 (62, ('chocolate', 'olive oil', 'mineral water')),
 (62, ('french fries', 'milk', 'mineral water')),
 (62, ('spaghetti', 'frozen vegetables', 'milk')),
 (62, ('spaghetti', 'mineral water', 'green tea')),
 (63, ('chocolate', 'french fries', 'eggs')),
 (64, ('chocolate', 'french fries', 'mineral water')),
 (64, ('milk', 'olive oil', 'mineral water'))]
```

In [19]:
```
sorted_d[len(sorted_d)-10:]
```

Out[19]:
```
[(83, ('ground beef', 'milk', 'mineral water')),
 (86, ('pancakes', 'spaghetti', 'mineral water')),
 (90, ('spaghetti', 'frozen vegetables', 'mineral water')),
 (98, ('milk', 'eggs', 'mineral water')),
 (101, ('chocolate', 'eggs', 'mineral water')),
 (105, ('chocolate', 'milk', 'mineral water')),
 (107, ('spaghetti', 'eggs', 'mineral water')),
 (118, ('spaghetti', 'milk', 'mineral water')),
 (119, ('chocolate', 'spaghetti', 'mineral water')),
 (128, ('ground beef', 'spaghetti', 'mineral water'))]
```

In [20]:
```
# Truc quan hoa ket qua theo result vua tim ra ???
```

In [21]:
```
# "Có Milk không? nó kết hợp với item nào?"
for k, v in result.items():
    if "milk" in k:
        print(k, ":", v)
```

```
('ground beef', 'spaghetti', 'milk') : 73
('ground beef', 'milk', 'mineral water') : 83
('chocolate', 'spaghetti', 'milk') : 82
('chocolate', 'frozen vegetables', 'milk') : 60
('chocolate', 'milk', 'eggs') : 69
('chocolate', 'milk', 'mineral water') : 105
('spaghetti', 'frozen vegetables', 'milk') : 62
('spaghetti', 'milk', 'eggs') : 67
('spaghetti', 'milk', 'mineral water') : 118
('frozen vegetables', 'milk', 'mineral water') : 83
('soup', 'milk', 'mineral water') : 64
('french fries', 'milk', 'mineral water') : 62
('milk', 'eggs', 'mineral water') : 98
('milk', 'olive oil', 'mineral water') : 64
```

In [22]:
```
# 10 san pham ma cua hang ban nhieu nhat/it nhat (theo tl) ???
```