ASSIGNMENT 2

Frequent itemset and Association rule mining

SET A
Q1.

In [1]:
```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

In [3]:
```
transactions = [['bread', 'milk'],
                ['bread', 'diaper','beer','eggs'],
                ['milk','diaper','beer','coke'],
                ['bread', 'milk','diaper','beer'],
                ['bread','milk','diaper','coke']]
```

In [4]:
```
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
df
```

Out[4]:

	beer	bread	coke	diaper	eggs	milk
0	False	True	False	False	False	True
1	True	True	False	True	True	False
2	True	False	True	True	False	True
3	True	True	False	True	False	True
4	False	True	True	True	False	True

In [6]:
```
freq_items = apriori(df, min_support = 0.6, use_colnames = True)
print(freq_items)
```

```
   support        itemsets
0      0.6          (beer)
1      0.8         (bread)
2      0.8        (diaper)
3      0.8          (milk)
4      0.6  (diaper, beer)
5      0.6 (bread, diaper)
6      0.6   (bread, milk)
7      0.6  (milk, diaper)
```

Most frequent Items include :

Beer
Bread
Diaper
Milk

Items purchased altogether most frequently :

[Frequent Itemsets]

Diaper and Beer
Bread and Diaper
Bread and Diaper
Milk and Diaper

In [7]:
```
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
rules
```

Out[7]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	(beer)	(diaper)	0.6	0.8	0.6	1.00	1.2500	0.12	inf
0	(diaper)	(beer)	0.8	0.6	0.6	0.75	1.2500	0.12	1.6
2	(bread)	(diaper)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8
3	(diaper)	(bread)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8
4	(bread)	(milk)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8
5	(milk)	(bread)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8
6	(milk)	(diaper)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8
7	(diaper)	(milk)	0.8	0.8	0.6	0.75	0.9375	-0.04	0.8

Association Rule :

{Beer} → {Diaper}

Support = 60 %

60 % transactions show that Diaper is bought with the purchase of Beer

Confidence = 100 %
(First Beer then Diaper) i.e. Diaper can be bought alone. But if beer is bought, then there is a 100% chance that diaper is bought.

The confidence level for the rule is 1.00

100 % of customers who purchased Diaper, had already purchased Beer.

Lift = 1.25

Since Lift > 1, Diaper and Beer are dependent.

Finally, the lift of 1.25 tells us that diaper is 1.25 times more likely to be bought by the customers who buy beer compared to the default likelihood of the sale of diaper.

Set B
Q1.

In [8]:
```
import pandas as pd
data=pd.read_csv('C:\\Market_Basket_Optimisation.csv')
data.describe
data.info()
data.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7500 entries, 0 to 7499
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   shrimp             7500 non-null   object
 1   almonds            5746 non-null   object
 2   avocado            4388 non-null   object
 3   vegetables mix     3344 non-null   object
 4   green grapes       2528 non-null   object
 5   whole weat flour   1863 non-null   object
 6   yams               1368 non-null   object
 7   cottage cheese     980 non-null    object
 8   energy drink       653 non-null    object
 9   tomato juice       394 non-null    object
 10  low fat yogurt     255 non-null    object
 11  green tea          153 non-null    object
 12  honey              86 non-null     object
 13  salad              46 non-null     object
 14  mineral water      24 non-null     object
 15  salmon             7 non-null      object
 16  antioxydant juice  3 non-null      object
 17  frozen smoothie    3 non-null      object
 18  spinach            2 non-null      object
 19  olive oil          0 non-null      float64
dtypes: float64(1), object(19)
memory usage: 1.1+ MB
```

Out[8]:
```
(7500, 20)
```

In [9]:
```
# Getting the list of transactions from the dataset
transactions = []
for i in range(0, len(data)):
 transactions.append([str(data.values[i,j]) for j in range(0, len(data.columns))])
```

In [10]:
```
#Encoding Transactions
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
df
```

In [ ]:
```
# Training Apriori algorithm on the dataset
!pip install apyori
from apyori import apriori
```

In [12]:
```
model=apriori(transactions, min_support = 0.003, min_confidence = 0.2, min_lift = 3, min_length = 2,max_length
model_table=list(model)
model_table
```

In [13]:
```
#Print Results
antecedents = [tuple(result[2][0][0])[0] for result in model_table]
consequents = [tuple(result[2][0][1])[0] for result in model_table]
support = [result[1] for result in model_table]
confidence = [result[2][0][2] for result in model_table]
lift= [result[2][0][3] for result in model_table]
new_data = list(zip(antecedents,consequents,support,confidence,lift))
Final_table=pd.DataFrame(new_data,columns=["Antecedents", "Consequents", "Support", "Confidence","Lift"])
Final_table
```

Out[13]:

	Antecedents	Consequents	Support	Confidence	Lift
0	light cream	chicken	0.004533	0.290598	4.843305
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790327
2	pasta	escalope	0.005867	0.372881	4.700185
3	fromage blanc	honey	0.003333	0.245098	5.178128
4	herb & pepper	ground beef	0.016000	0.323450	3.291555
5	tomato sauce	ground beef	0.005333	0.377358	3.840147
6	light cream	olive oil	0.003200	0.205128	3.120612
7	whole wheat pasta	olive oil	0.008000	0.271493	4.130221
8	pasta	shrimp	0.005067	0.322034	4.514494

Association Rule :

{Light cream} → {chicken}

Confidence = 29 %

(First light cream then chicken) But if antecedent is bought, then there is a 29% chance that the consequent is bought.
The confidence level for the rule is 0.29

29 % of customers who purchased chicken, had already purchased light cream.

Lift = 4.84

Since Lift > 1, light cream and chicken are dependent.