In [1]:
```
dataset = [['Coffee','Milk','sugar','bread','eggs','chocos'],
           ['coffee','milk','bread','butter','biscuits','chips'],
           ['milk','coffee','cakes','chips','bread'],
           ['milk','juices','bread','cakes','choclates'],
           ['bread','juices','cakes','bhel','ice cream','cheese']]
```

In [2]:
```
import pandas as pd
!pip install mlxtend
from mlxtend.preprocessing import TransactionEncoder
```

```
Requirement already satisfied: mlxtend in c:\users\manoj\anaconda3\lib\site-pac
kages (0.18.0)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\manoj\anaconda3\li
b\site-packages (from mlxtend) (3.2.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\manoj\anaconda3\lib\si
te-packages (from mlxtend) (1.19.5)
Requirement already satisfied: joblib>=0.13.2 in c:\users\manoj\anaconda3\lib\s
ite-packages (from mlxtend) (0.16.0)
Requirement already satisfied: scipy>=1.2.1 in c:\users\manoj\anaconda3\lib\sit
e-packages (from mlxtend) (1.5.0)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\manoj\anaconda3
\lib\site-packages (from mlxtend) (0.23.1)
Requirement already satisfied: pandas>=0.24.2 in c:\users\manoj\anaconda3\lib\s
ite-packages (from mlxtend) (1.0.5)
Requirement already satisfied: setuptools in c:\users\manoj\anaconda3\lib\site-
packages (from mlxtend) (49.2.0.post20200714)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\manoj\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\u
sers\manoj\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.
7)
Requirement already satisfied: cycler>=0.10 in c:\users\manoj\anaconda3\lib\sit
e-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\manoj\anaconda3
\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\manoj\anaconda3
\lib\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\manoj\anaconda3\lib\sit
e-packages (from pandas>=0.24.2->mlxtend) (2020.1)
Requirement already satisfied: six in c:\users\manoj\anaconda3\lib\site-package
s (from cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.15.0)
```

In [3]:
```
te = TransactionEncoder()
te_try = te.fit(dataset).transform(dataset)
```

In [4]:
```
df=pd.DataFrame(te_try,columns=te.columns_)
```

In [5]:
```
df
```

Out[5]:

	Coffee	Milk	bhel	biscuits	bread	butter	cakes	cheese	chips	choclates	chocos	coffee
0	True	True	False	False	True	False	False	False	False	False	True	False
1	False	False	False	True	True	True	False	False	True	False	False	True
2	False	False	False	False	True	False	True	False	True	False	False	True
3	False	False	False	False	True	False	True	False	False	True	False	False
4	False	False	True	False	True	False	True	True	False	False	False	False

In [6]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
apriori(df,min_support = 0.6)
```

Out[6]:

	support	itemsets
0	1.0	(4)
1	0.6	(6)
2	0.6	(15)
3	0.6	(4, 6)
4	0.6	(4, 15)

In [7]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
apriori(df,min_support = 0.8)
```

Out[7]:

	support	itemsets
0	1.0	(4)

In [8]:
```
frequent_itemsets = apriori(df,min_support = 0.5,use_colnames=True)
```

In [9]:
```
equent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x:len(x))
```

In [10]:
```
emsets[(frequent_itemsets['length'] == 2)& (frequent_itemsets['support']>=0.8) ]
```

Out[10]:

	support	itemsets	length

In [11]:
```
frequent_itemsets[(frequent_itemsets['length'] == 3)]
```

Out[11]:

	support	itemsets	length

In [12]:
```
frequent_itemsets[(frequent_itemsets['support'] >=1) ]
```

Out[12]:

	support	itemsets	length
0	1.0	(bread)	1

In [13]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

frequent_factors = apriori(df, use_colnames= True,min_support= 0.1,max_len =4)

rules = association_rules(frequent_factors,metric='lift', min_threshold=1)

print(rules.head(10).sort_values(by = 'confidence', ascending= False))
```

```
  antecedents consequents  antecedent support  consequent support  support  \
0      (Milk)    (Coffee)                 0.2                 0.2      0.2
1    (Coffee)      (Milk)                 0.2                 0.2      0.2
3    (Coffee)     (bread)                 0.2                 1.0      0.2
4    (chocos)    (Coffee)                 0.2                 0.2      0.2
5    (Coffee)    (chocos)                 0.2                 0.2      0.2
6      (eggs)    (Coffee)                 0.2                 0.2      0.2
7    (Coffee)      (eggs)                 0.2                 0.2      0.2
8     (sugar)    (Coffee)                 0.2                 0.2      0.2
9    (Coffee)     (sugar)                 0.2                 0.2      0.2
2     (bread)    (Coffee)                 1.0                 0.2      0.2

   confidence  lift  leverage  conviction
0         1.0   5.0      0.16         inf
1         1.0   5.0      0.16         inf
3         1.0   1.0      0.00         inf
4         1.0   5.0      0.16         inf
5         1.0   5.0      0.16         inf
6         1.0   5.0      0.16         inf
7         1.0   5.0      0.16         inf
8         1.0   5.0      0.16         inf
9         1.0   5.0      0.16         inf
2         0.2   1.0      0.00         1.0
```

In [14]:
```
'{\'bread\'}'

pple = rules[(rules['consequents'].astype(str).str.contains(target, na= False))]
pple
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
3	(Coffee)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
10	(Milk)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
18	(bhel)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
28	(biscuits)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
38	(butter)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
...	...	...	...	...	...	...	...	...
1143	(juices, cakes, milk)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
1155	(juices, ice cream, cheese)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
1168	(coffee, milk, chips)	(bread)	0.4	1.0	0.4	1.0	1.0	0.0
1182	(choclates, juices, milk)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0
1197	(chocos, eggs, sugar)	(bread)	0.2	1.0	0.2	1.0	1.0	0.0

90 rows × 9 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Type *Markdown* and LaTeX: α^2