In [38]:
```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
df = pd.read_csv("BreadBasket.csv")
df
```

Out[38]:

	Tx	products
0	0	MILK,BREAD,BISCUIT
1	1	BREAD,MILK,BISCUIT,CORNFLAKES
2	2	BREAD,TEA,BOURNVITA
3	3	JAM,MAGGI,BREAD,MILK
4	4	MAGGI,TEA,BISCUIT
5	5	BREAD,TEA,BOURNVITA
6	6	MAGGI,TEA,CORNFLAKES
7	7	MAGGI,BREAD,TEA,BISCUIT
8	8	JAM,MAGGI,BREAD,TEA
9	9	BREAD,MILK
10	10	COFFEE,COCK,BISCUIT,CORNFLAKES
11	11	COFFEE,COCK,BISCUIT,CORNFLAKES
12	12	COFFEE,SUGER,BOURNVITA
13	13	BREAD,COFFEE,COCK
14	14	BREAD,SUGER,BISCUIT
15	15	COFFEE,SUGER,CORNFLAKES
16	16	BREAD,SUGER,BOURNVITA
17	17	BREAD,COFFEE,SUGER
18	18	BREAD,COFFEE,SUGER
19	19	TEA,MILK,COFFEE,CORNFLAKES

In [39]:
```
data = list(df["products"].apply(lambda x:x.split(",") ))
data
```

Out[39]:
```
[['MILK', 'BREAD', 'BISCUIT'],
 ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['JAM', 'MAGGI', 'BREAD', 'MILK'],
 ['MAGGI', 'TEA', 'BISCUIT'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['MAGGI', 'TEA', 'CORNFLAKES'],
 ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
 ['JAM', 'MAGGI', 'BREAD', 'TEA'],
 ['BREAD', 'MILK'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'COCK'],
 ['BREAD', 'SUGER', 'BISCUIT'],
 ['COFFEE', 'SUGER', 'CORNFLAKES'],
 ['BREAD', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

Apriori Algorithm

In [40]:
```
from mlxtend.preprocessing import TransactionEncoder
a = TransactionEncoder()
a_data = a.fit(data).transform(data)
df = pd.DataFrame(a_data,columns=a.columns_)
# df = df.replace(False,0)
df
```

Out[40]:

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	True	False	True	False	False	False	False	False	True	False	False
1	True	False	True	False	False	True	False	False	True	False	False
2	False	True	True	False	False	False	False	False	False	False	True
3	False	False	True	False	False	False	True	True	True	False	False
4	True	False	False	False	False	False	False	True	False	False	True
5	False	True	True	False	False	False	False	False	False	False	True
6	False	False	False	False	False	True	False	True	False	False	True
7	True	False	True	False	False	False	False	True	False	False	True
8	False	False	True	False	False	False	True	True	False	False	True
9	False	False	True	False	False	False	False	False	True	False	False
10	True	False	False	True	True	True	False	False	False	False	False
11	True	False	False	True	True	True	False	False	False	False	False
12	False	True	False	False	True	False	False	False	False	True	False
13	False	False	True	False	True	False	False	False	False	False	False
14	True	False	True	False	False	False	False	False	False	True	False
15	False	False	False	False	True	True	False	False	False	True	False
16	False	True	True	False	False	False	False	False	False	True	False
17	False	False	True	False	True	False	False	False	False	True	False
18	False	False	True	False	True	False	False	False	False	True	False
19	False	False	False	False	False	True	False	False	True	False	True

In [41]:
```
from mlxtend.frequent_patterns import apriori

apriori(df, min_support=0.2)
```

Out[41]:

	support	itemsets
0	0.35	(0)
1	0.20	(1)
2	0.65	(2)
3	0.40	(4)
4	0.30	(5)
5	0.25	(7)
6	0.25	(8)
7	0.30	(9)
8	0.35	(10)
9	0.20	(0, 2)
10	0.20	(8, 2)
11	0.20	(9, 2)
12	0.20	(2, 10)
13	0.20	(4, 5)
14	0.20	(9, 4)
15	0.20	(10, 7)

In [42]:
```
df=apriori(df, min_support=0.2, use_colnames=True)
df
```

Out[42]:

	support	itemsets
0	0.35	(BISCUIT)
1	0.20	(BOURNVITA)
2	0.65	(BREAD)
3	0.40	(COFFEE)
4	0.30	(CORNFLAKES)
5	0.25	(MAGGI)
6	0.25	(MILK)
7	0.30	(SUGER)
8	0.35	(TEA)
9	0.20	(BISCUIT, BREAD)
10	0.20	(MILK, BREAD)
11	0.20	(SUGER, BREAD)
12	0.20	(TEA, BREAD)
13	0.20	(COFFEE, CORNFLAKES)
14	0.20	(COFFEE, SUGER)
15	0.20	(MAGGI, TEA)

In [43]:
```
df_ar = association_rules(df, metric = "confidence", min_threshold = 0.6)
df_ar
```

Out[43]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.75	0.250000
1	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.05	0.035714
2	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80	0.571429
3	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80	0.571429
4	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.25	0.750000

FP Grouth Algorithm

In [45]:
```
df = pd.read_csv("BreadBasket.csv")
df
```

Out[45]:

	Tx	products
0	0	MILK,BREAD,BISCUIT
1	1	BREAD,MILK,BISCUIT,CORNFLAKES
2	2	BREAD,TEA,BOURNVITA
3	3	JAM,MAGGI,BREAD,MILK
4	4	MAGGI,TEA,BISCUIT
5	5	BREAD,TEA,BOURNVITA
6	6	MAGGI,TEA,CORNFLAKES
7	7	MAGGI,BREAD,TEA,BISCUIT
8	8	JAM,MAGGI,BREAD,TEA
9	9	BREAD,MILK
10	10	COFFEE,COCK,BISCUIT,CORNFLAKES
11	11	COFFEE,COCK,BISCUIT,CORNFLAKES
12	12	COFFEE,SUGER,BOURNVITA
13	13	BREAD,COFFEE,COCK
14	14	BREAD,SUGER,BISCUIT
15	15	COFFEE,SUGER,CORNFLAKES
16	16	BREAD,SUGER,BOURNVITA
17	17	BREAD,COFFEE,SUGER
18	18	BREAD,COFFEE,SUGER
19	19	TEA,MILK,COFFEE,CORNFLAKES

In [46]:
```
data = list(df["products"].apply(lambda x:x.split(",") ))
data
```

Out[46]:
```
[['MILK', 'BREAD', 'BISCUIT'],
 ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['JAM', 'MAGGI', 'BREAD', 'MILK'],
 ['MAGGI', 'TEA', 'BISCUIT'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['MAGGI', 'TEA', 'CORNFLAKES'],
 ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
 ['JAM', 'MAGGI', 'BREAD', 'TEA'],
 ['BREAD', 'MILK'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'COCK'],
 ['BREAD', 'SUGER', 'BISCUIT'],
 ['COFFEE', 'SUGER', 'CORNFLAKES'],
 ['BREAD', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

In [47]:
```
from mlxtend.preprocessing import TransactionEncoder
a = TransactionEncoder()
a_data = a.fit(data).transform(data)
df = pd.DataFrame(a_data,columns=a.columns_)
# df = df.replace(False,0)
df
```

Out[47]:

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	True	False	True	False	False	False	False	False	True	False	False
1	True	False	True	False	False	True	False	False	True	False	False
2	False	True	True	False	False	False	False	False	False	False	True
3	False	False	True	False	False	False	True	True	True	False	False
4	True	False	False	False	False	False	False	True	False	False	True
5	False	True	True	False	False	False	False	False	False	False	True
6	False	False	False	False	False	True	False	True	False	False	True
7	True	False	True	False	False	False	False	True	False	False	True
8	False	False	True	False	False	False	True	True	False	False	True
9	False	False	True	False	False	False	False	False	True	False	False
10	True	False	False	True	True	True	False	False	False	False	False
11	True	False	False	True	True	True	False	False	False	False	False
12	False	True	False	False	True	False	False	False	False	True	False
13	False	False	True	False	True	False	False	False	False	False	False
14	True	False	True	False	False	False	False	False	False	True	False
15	False	False	False	False	True	True	False	False	False	True	False
16	False	True	True	False	False	False	False	False	False	True	False
17	False	False	True	False	True	False	False	False	False	True	False
18	False	False	True	False	True	False	False	False	False	True	False
19	False	False	False	False	False	True	False	False	True	False	True

In [48]:
```
from mlxtend.frequent_patterns import fpgrowth

fpgrowth(df, min_support=0.2)
```

Out[48]:

	support	itemsets
0	0.65	(2)
1	0.35	(0)
2	0.25	(8)
3	0.30	(5)
4	0.35	(10)
5	0.20	(1)
6	0.25	(7)
7	0.40	(4)
8	0.30	(9)
9	0.20	(0, 2)
10	0.20	(8, 2)
11	0.20	(4, 5)
12	0.20	(10, 2)
13	0.20	(10, 7)
14	0.20	(9, 4)
15	0.20	(9, 2)

In [52]:
```
df=fpgrowth(df, min_support=0.2, use_colnames=True)
df
```

Out[52]:

	support	itemsets
0	0.65	(BREAD)
1	0.35	(BISCUIT)
2	0.25	(MILK)
3	0.30	(CORNFLAKES)
4	0.35	(TEA)
5	0.20	(BOURNVITA)
6	0.25	(MAGGI)
7	0.40	(COFFEE)
8	0.30	(SUGER)
9	0.20	(BISCUIT, BREAD)
10	0.20	(MILK, BREAD)
11	0.20	(COFFEE, CORNFLAKES)
12	0.20	(TEA, BREAD)
13	0.20	(MAGGI, TEA)
14	0.20	(COFFEE, SUGER)
15	0.20	(SUGER, BREAD)

In [55]:
```
df_ar = association_rules(df, metric = "confidence", min_threshold = 0.2)
df_ar
```

Out[55]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(BISCUIT)	(BREAD)	0.35	0.65	0.2	0.571429	0.879121	-0.0275	0.816667	-0.174603
1	(BREAD)	(BISCUIT)	0.65	0.35	0.2	0.307692	0.879121	-0.0275	0.938889	-0.282051
2	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.750000	0.250000
3	(BREAD)	(MILK)	0.65	0.25	0.2	0.307692	1.230769	0.0375	1.092333	0.535714
4	(COFFEE)	(CORNFLAKES)	0.40	0.30	0.2	0.500000	1.666667	0.0800	1.400000	0.666667
5	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.800000	0.571429
6	(TEA)	(BREAD)	0.35	0.65	0.2	0.571429	0.879121	-0.0275	0.816667	-0.174603
7	(BREAD)	(TEA)	0.65	0.35	0.2	0.307692	0.879121	-0.0275	0.938889	-0.282051
8	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.250000	0.750000
9	(TEA)	(MAGGI)	0.35	0.25	0.2	0.571429	2.285714	0.1125	1.750000	0.865385
10	(COFFEE)	(SUGER)	0.40	0.30	0.2	0.500000	1.666667	0.0800	1.400000	0.666667
11	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.800000	0.571429
12	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.050000	0.035714
13	(BREAD)	(SUGER)	0.65	0.30	0.2	0.307692	1.025641	0.0050	1.011111	0.071429