Transactions comprising the item/total transactions=Support(item)

Confidence(bread->milk)=Transactions comprising bread and milk/Transactions comprising bread

Lift=Confidence(bread->milk)/support(bread)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("/content/Groceries_dataset.csv")
df
```

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	05-01-2015	whole milk
2	2300	19-09-2015	pip fruit
3	1187	12-12-2015	other vegetables
4	3037	01-02-2015	whole milk
...	...	...	...
38760	4471	08-10-2014	sliced cheese
38761	2022	23-02-2014	candy
38762	1097	16-04-2014	cake bar
38763	1510	03-12-2014	fruit/vegetable juice
38764	1521	26-12-2014	cat food

38765 rows × 3 columns

```
df['single_transaction']=df['Member_number'].astype(str)+'_'+df['Date'].astype(str)
df.head()
```

	Member_number	Date	itemDescription	single_transaction
0	1808	21-07-2015	tropical fruit	1808_21-07-2015
1	2552	05-01-2015	whole milk	2552_05-01-2015
2	2300	19-09-2015	pip fruit	2300_19-09-2015
3	1187	12-12-2015	other vegetables	1187_12-12-2015
4	3037	01-02-2015	whole milk	3037_01-02-2015

```
df2=pd.crosstab(df['single_transaction'],df['itemDescription'])
df2.head()
```

itemDescription	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	bags	baking powder	bathroom cleaner	beef	berries	...	turkey	vinegar	waff
single_transaction														
1000_15-03-2015	0	0	0	0	0	0	0	0	0	0	...	0	0	
1000_24-06-2014	0	0	0	0	0	0	0	0	0	0	...	0	0	
1000_24-07-2015	0	0	0	0	0	0	0	0	0	0	...	0	0	
1000_25-11-2015	0	0	0	0	0	0	0	0	0	0	...	0	0	
1000_27-05-2015	0	0	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 167 columns

```
def encode(item_freq):
    res=0
    if item_freq>0:
      res=1
    return res
basket_input=df2.applymap(encode)
```

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
frequent_itemsets=apriori(basket_input,min_support=0.001,use_colnames=True)
rules=association_rules(frequent_itemsets,metric="lift")
rules.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(bottled water)	(UHT-milk)	0.060683	0.021386	0.001069	0.017621	0.823954	-0.000228	0.996168	-0.185312
1	(UHT-milk)	(bottled water)	0.021386	0.060683	0.001069	0.050000	0.823954	-0.000228	0.988755	-0.179204
2	(other vegetables)	(UHT-milk)	0.122101	0.021386	0.002139	0.017515	0.818993	-0.000473	0.996060	-0.201119
3	(UHT-milk)	(other vegetables)	0.021386	0.122101	0.002139	0.100000	0.818993	-0.000473	0.975443	-0.184234

```
rules.sort_values(["support","confidence","lift"],axis=0,ascending=True)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
599	(whole milk)	(pot plants)	0.157923	0.007819	0.001002	0.006348	0.811821	-0.000232	0.998519	-0.215852
545	(other vegetables)	(semi-finished bread)	0.122101	0.009490	0.001002	0.008210	0.865133	-0.000156	0.998710	-0.150796
536	(other vegetables)	(pot plants)	0.122101	0.007819	0.001002	0.008210	1.049991	0.000048	1.000394	0.054233
616	(rolls/buns)	(soft cheese)	0.110005	0.010025	0.001002	0.009113	0.909052	-0.000100	0.999080	-0.101053
304	(rolls/buns)	(detergent)	0.110005	0.008621	0.001002	0.009113	1.057037	0.000054	1.000496	0.060629
...	...	...	...	...	...	...	...	...	...	...
551	(soda)	(other vegetables)	0.097106	0.122101	0.009691	0.099794	0.817302	-0.002166	0.975219	-0.198448
694	(whole milk)	(yogurt)	0.157923	0.085879	0.011161	0.070673	0.822940	-0.002401	0.983638	-0.203508
695	(yogurt)	(whole milk)	0.085879	0.157923	0.011161	0.129961	0.822940	-0.002401	0.967861	-0.190525
623	(whole milk)	(rolls/buns)	0.157923	0.110005	0.013968	0.088447	0.804028	-0.003404	0.976350	-0.224474