

Web Mining Lab Assignment-12

Aryan Vigyat

20BCE1452

Problem Statement

To determine the association between various products in the basket by analysing the customer purchase pattern of multiple items.

TASK 1- Frequency of Top 50 most popular Items

```
In [ ]: import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

```
In [ ]: df=pd.read_csv('dataset.csv', header=None)
```

```
In [ ]: df.head(5)
```

```
Out[ ]:
```

	0	1	2	3	4	5	6	7	8
0	yogurt	pork	sandwich bags	lunch meat	all-purpose	flour	soda	butter	vegetables
1	toilet paper	shampoo	hand soap	waffles	vegetables	cheeses	mixes	milk	sandwich bags
2	soda	pork	soap	ice cream	toilet paper	dinner rolls	hand soap	spaghetti sauce	milk
3	cereals	juice	lunch meat	soda	toilet paper	all-purpose	NaN	NaN	NaN
4	sandwich loaves	pasta	tortillas	mixes	hand soap	toilet paper	vegetables	vegetables	paper towels

5 rows × 34 columns

```
In [ ]: df=df.replace(np.nan, " ")
df
```

Out[]:

	0	1	2	3	4	5	6	7
0	yogurt	pork	sandwich bags	lunch meat	all-purpose	flour	soda	butter
1	toilet paper	shampoo	hand soap	waffles	vegetables	cheeses	mixes	milk
2	soda	pork	soap	ice cream	toilet paper	dinner rolls	hand soap	spaghetti sauce
3	cereals	juice	lunch meat	soda	toilet paper	all-purpose		
4	sandwich loaves	pasta	tortillas	mixes	hand soap	toilet paper	vegetables	vegetables
...
1134	sugar	beef	sandwich bags	hand soap	paper towels	paper towels	all-purpose	beef
1135	coffee/tea	dinner rolls	lunch meat	spaghetti sauce	pasta	vegetables	cereals	dinner rolls
1136	beef	lunch meat	eggs	poultry	vegetables	tortillas	beef	beef
1137	sandwich bags	ketchup	milk	poultry	cheeses	soap	toilet paper	yogurt
1138	soda	laundry detergent	vegetables	shampoo	vegetables			

1139 rows × 34 columns

In []: `#unique values in table`

```

items = set()
for col in df:
    items.update(df[col].unique())
print(items)
#number of unique values without nan
print(len(items))

```

```

'', 'all- purpose', ' sandwich loaves', 'mixes', 'sandwich bags', 'paper towels', ' cheeses', 'cheeses', 'hand soap', 'milk', ' hand soap', ' toilet paper', ' spaghetti sauce', ' vegetables', ' sandwich bags', 'waffles', 'tortillas', 'eggs', ' shampoo', ' tortillas', 'sugar', ' eggs', 'coffee/tea', ' ice cream', ' paper towels', 'pasta', 'laundry detergent', 'sandwich loaves', ' bagels', 'soda', 'soap', 'cereals', 'ketchup', 'spaghetti sauce', 'aluminum foil', ' cereals', ' individual meals', 'poultry', 'individual meals', 'shampoo', ' milk', 'pork', ' poultry', 'bagels', 'dinner rolls', ' fruits', ' dishwashing liquid/detergent', ' lunch meat', ' aluminum foil', ' dinner rolls', ' pork', ' ketchup', 'flour', 'butter', ' soda', 'fruits', ' mixes', 'ice cream', ' sugar', 'vegetables', ' butter', ' yogurt', ' pasta', 'lunch meat', ' flour', ' coffee/tea', 'beef', 'toilet paper', ' soap', ' laundry detergent', ' all- purpose', ' beef', 'yogurt', 'juice', ' waffles', 'dishwashing liquid/detergent', ' juice'}

```

77

In []: `listdf = df.values.tolist()`

```

listdf=np.array(listdf)
listdf=listdf.flatten()

```

```
listdf=listdf.tolist()  
listdf
```


,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

'sandwich loaves',

'pasta',

'tortillas',

'mixes',

'hand soap',

'toilet paper',

'vegetables',

'vegetables',

'paper towels',

'vegetables',

'flour',

'vegetables',

'pork',

'poultry',

'eggs',

'vegetables',

'pork',

'spaghetti sauce',

'vegetables',

'milk',

'waffles',

'individual meals',

'vegetables',

'dinner rolls',

'all-purpose',

'soda',

'yogurt',

,

,

,

,

,

,

,

,

,

'laundry detergent',

'toilet paper',

'eggs',

'toilet paper',

'vegetables',

'bagels',

'dishwashing liquid/detergent',

'cereals',

'paper towels',

'laundry detergent',

'butter',

'cereals',

'bagels',

',
'ketchup',
' coffee/tea',
' toilet paper',
' pork',
' flour',
' milk',
' soda',
' dishwashing liquid/detergent',
' eggs',
' tortillas',
' tortillas',
' vegetables',
' fruits',
' sugar',
' soap',
' dishwashing liquid/detergent',
' juice',
' sandwich loaves',
' poultry',
' eggs',
' flour',
' bagels',
' soda',
' bagels',
' soda',
' beef',
' coffee/tea',
' dishwashing liquid/detergent',
' cheeses',
' beef',
' soap',
' lunch meat',
' pork',
' waffles',
' sandwich loaves',
' ice cream',
' soda',
' bagels',
' dishwashing liquid/detergent',
' eggs',
' sugar',
' waffles',
' individual meals',
' toilet paper',
' pasta',
' sandwich loaves',
' cheeses',
' vegetables',
' hand soap',
' vegetables',
' vegetables',
' ketchup',
' poultry',
' dinner rolls',
' dishwashing liquid/detergent',
' coffee/tea',
' milk',
' shampoo',
' ketchup',
' sandwich loaves',

' aluminum foil',
' vegetables',
' shampoo',
' cereals',
' dinner rolls',
' coffee/tea',
' poultry',
' butter',
' juice',
' paper towels',
' vegetables',
' beef',
' mixes',
' lunch meat',
' bagels',
' tortillas',
' aluminum foil',
' individual meals',
' beef',
' juice',
' soda',
' hand soap',
' dinner rolls',
' flour',
' poultry',
' sandwich loaves',
' sugar',
' yogurt',
,

' laundry detergent',
' pork',
' pasta',
' cheeses',
' fruits',
' sugar',
' lunch meat',
' laundry detergent',
' paper towels',
' butter',
' sandwich loaves',
' ice cream',
' waffles',
' all- purpose',
' sandwich bags',
' lunch meat',
' sandwich loaves',
' poultry',
' vegetables',
' shampoo',
' dinner rolls',
' milk',
' soda',
' mixes',
,


```
',  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
'flour',  
' bagels',  
' cheeses',  
' sandwich loaves',  
' toilet paper',  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
,  
...]
```

Number of transactions for each item

```
In [ ]: from collections import Counter  
c = Counter( listdf )
```

```
In [ ]: from collections import OrderedDict  
dict1=OrderedDict(sorted(c.items()))  
del dict1[' ']  
print(dict1)
```

```
OrderedDict([(' all- purpose', 522), (' aluminum foil', 534), (' bagels', 544), (' beef', 538), (' butter', 531), (' cereals', 558), (' cheeses', 545), (' coffee/tea', 547), (' dinner rolls', 538), (' dishwashing liquid/detergent', 520), (' egg', 539), (' flour', 522), (' fruits', 503), (' hand soap', 476), (' ice cream', 548), (' individual meals', 524), (' juice', 532), (' ketchup', 524), (' laundry detergent', 515), (' lunch meat', 541), (' milk', 519), (' mixes', 536), (' paper towels', 530), (' pasta', 509), (' pork', 507), (' poultry', 607), (' sandwich bags', 502), (' sandwich loaves', 490), (' shampoo', 535), (' soap', 556), (' soda', 573), (' spaghetti sauce', 508), (' sugar', 513), (' toilet paper', 539), (' tortillas', 498), (' vegetables', 1617), (' waffles', 545), (' yogurt', 519), ('all- purpose', 29), ('aluminum foil', 32), ('bagels', 29), ('beef', 23), ('butter', 24), ('cereals', 33), ('cheeses', 33), ('coffee/tea', 18), ('dinner rolls', 29), ('dish washing liquid/detergent', 31), ('eggs', 31), ('flour', 33), ('fruits', 26), ('hand soap', 26), ('ice cream', 31), ('individual meals', 20), ('juice', 38), ('ketchup', 24), ('laundry detergent', 27), ('lunch meat', 32), ('milk', 36), ('mixes', 18), ('paper towels', 26), ('pasta', 33), ('pork', 24), ('poultry', 33), ('sandwich bags', 34), ('sandwich loaves', 33), ('shampoo', 27), ('soap', 18), ('soda', 24), ('spaghetti sauce', 28), ('sugar', 20), ('toilet paper', 30), ('tortillas', 45), ('vegetables', 85), ('waffles', 30), ('yogurt', 26)])
```

Task 1: Identify the frequency of most popular 50 items

Popularity of the item based on the number of transactions for the given item

```
In [ ]: import itertools  
out=dict(itertools.islice(dict1.items(), 50))  
for key, value in out.items():  
    print(str(key)+" : "+ " "+str(value))
```

```
all- purpose: 522  
aluminum foil: 534  
bagels: 544  
beef: 538  
butter: 531  
cereals: 558  
cheeses: 545  
coffee/tea: 547  
dinner rolls: 538  
dishwashing liquid/detergent: 520  
eggs: 539  
flour: 522  
fruits: 503  
hand soap: 476  
ice cream: 548  
individual meals: 524  
juice: 532  
ketchup: 524  
laundry detergent: 515  
lunch meat: 541  
milk: 519  
mixes: 536  
paper towels: 530  
pasta: 509  
pork: 507  
poultry: 607  
sandwich bags: 502  
sandwich loaves: 490  
shampoo: 535  
soap: 556  
soda: 573  
spaghetti sauce: 508  
sugar: 513  
toilet paper: 539  
tortillas: 498  
vegetables: 1617  
waffles: 545  
yogurt: 519  
all- purpose: 29  
aluminum foil: 32  
bagels: 29  
beef: 23  
butter: 24  
cereals: 33  
cheeses: 33  
coffee/tea: 18  
dinner rolls: 29  
dishwashing liquid/detergent: 31  
eggs: 31  
flour: 33
```

Task 2- APRIORI ALGORITHM

Task 2: Run aprior algorithm using Python for the following support and confidence

Case 1 (minimum support=0.15 and minimum confidence=0.6 (60%))

Identify the Itemset and derive association rules.

```
In [ ]: itemset = set(items)
encoded_vals = []
for index, row in df.iterrows():
    rowset = set(row)
    labels = {}
    uncommons = list(itemset - rowset)
    commons = list(itemset.intersection(rowset))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals.append(labels)
encoded_vals[0]
ohe_df = pd.DataFrame(encoded_vals)
```

```
In [ ]: ohe_df
```

Out[]:

	cereals	all-purpose	sandwich loaves	mixes	individual meals	sandwich bags	paper towels	poultry	cheeses	individu me
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	1
2	1	0	1	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
...
1134	1	0	0	0	0	0	0	0	0	1
1135	1	0	0	0	1	0	0	0	0	1
1136	0	0	0	0	1	0	0	0	0	0
1137	0	0	0	0	0	1	0	0	0	1
1138	0	0	0	0	0	0	0	0	0	0

1139 rows × 77 columns

```
In [ ]: freq_items = apriori(ohe_df, min_support=0.15, use_colnames=True, verbose=1)
freq_items
```

Processing 4476 combinations | Sampling itemset size 4

```
C:\Users\Acer\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:11
1: DeprecationWarning: DataFrames with non-bool types result in worse computational performance and their support might be discontinued in the future. Please use a DataFrame with bool type
warnings.warn(
```

Out[]:

	support	itemsets
0	0.380158	(cereals)
1	0.337138	(sandwich loaves)
2	0.365233	(individual meals)
3	0.371378	(cheeses)
4	0.332748	(hand soap)
...
413	0.242318	(pork, vegetables,)
414	0.269535	(vegetables, ice cream,)
415	0.260755	(, vegetables, shampoo)
416	0.256365	(, vegetables, flour)
417	0.269535	(, vegetables, soap)

418 rows × 2 columns

In []: `rules = association_rules(freq_items, metric="confidence", min_threshold=0.6)`
rules

Out[]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(cereals)	()	0.380158	0.974539	0.367867	0.967667	0.992949	-0.002611
1	(cereals)	(vegetables)	0.380158	0.715540	0.289728	0.762125	1.065104	0.017710
2	(sandwich loaves)	()	0.337138	0.974539	0.322212	0.955729	0.980699	-0.006340
3	(sandwich loaves)	(vegetables)	0.337138	0.715540	0.266901	0.791667	1.106391	0.025660
4	(individual meals)	()	0.365233	0.974539	0.351185	0.961538	0.986660	-0.004740
...
252	(vegetables, flour)	()	0.266901	0.974539	0.256365	0.960526	0.985621	-0.003740
253	(flour)	(, vegetables)	0.337138	0.693591	0.256365	0.760417	1.096348	0.022530
254	(, soap)	(vegetables)	0.352063	0.715540	0.269535	0.765586	1.069942	0.017610
255	(vegetables, soap)	()	0.285338	0.974539	0.269535	0.944615	0.969295	-0.008530
256	(soap)	(, vegetables)	0.370500	0.693591	0.269535	0.727488	1.048872	0.012550

257 rows × 9 columns

In []: `rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")`
rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("unicode")

In []: rules

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	cereals		0.380158	0.974539	0.367867	0.967667	0.992949	-0.002610
1	cereals	vegetables	0.380158	0.715540	0.289728	0.762125	1.065104	0.017710
2	sandwich loaves		0.337138	0.974539	0.322212	0.955729	0.980699	-0.006340
3	sandwich loaves	vegetables	0.337138	0.715540	0.266901	0.791667	1.106391	0.025660
4	individual meals		0.365233	0.974539	0.351185	0.961538	0.986660	-0.004740
...
252	vegetables		0.266901	0.974539	0.256365	0.960526	0.985621	-0.003740
253	flour		0.337138	0.693591	0.256365	0.760417	1.096348	0.022530
254		vegetables	0.352063	0.715540	0.269535	0.765586	1.069942	0.017610
255	vegetables		0.285338	0.974539	0.269535	0.944615	0.969295	-0.008530
256	soap		0.370500	0.693591	0.269535	0.727488	1.048872	0.012550

257 rows × 9 columns

Case 1- Itemset with Support

In []:

```
for index, row in freq_items.iterrows():
    print(row['itemsets'])
    print("Support: ")
    print(row['support'])
    print("=====")
```

```
frozenset({' cereals'})
Support:
0.3801580333625988
=====
frozenset({' sandwich loaves'})
Support:
0.33713784021071114
=====
frozenset({' individual meals'})
Support:
0.3652326602282704
=====
frozenset({' cheeses'})
Support:
0.3713784021071115
=====
frozenset({' hand soap'})
Support:
0.33274802458296754
=====
frozenset({' milk'})
Support:
0.35733099209833186
=====
frozenset({' toilet paper'})
Support:
0.3608428446005268
=====
frozenset({' poultry'})
Support:
0.4003511852502195
=====
frozenset({' spaghetti sauce'})
Support:
0.35645302897278314
=====
frozenset({' fruits'})
Support:
0.353819139596137
=====
frozenset({' dishwashing liquid/detergent'})
Support:
0.3687445127304653
=====
frozenset({' ketchup'})
Support:
0.35908691834942935
=====
frozenset({' tortillas'})
Support:
0.34416154521510095
=====
frozenset({' eggs'})
Support:
0.3722563652326602
=====
frozenset({' sugar'})
Support:
0.3494293239683933
=====
frozenset({' paper towels'})
```

```
Support:  
0.3485513608428446  
=====  
frozenset({' yogurt'})  
Support:  
0.3669885864793679  
=====  
frozenset({' pasta'})  
Support:  
0.3485513608428446  
=====  
frozenset({' bagels'})  
Support:  
0.3669885864793679  
=====  
frozenset({' coffee/tea'})  
Support:  
0.37050043898156276  
=====  
frozenset({' waffles'})  
Support:  
0.3775241439859526  
=====  
frozenset({' juice'})  
Support:  
0.35996488147497807  
=====  
frozenset({' '})  
Support:  
0.974539069359087  
=====  
frozenset({' soda'})  
Support:  
0.37576821773485514  
=====  
frozenset({' mixes'})  
Support:  
0.3687445127304653  
=====  
frozenset({' laundry detergent'})  
Support:  
0.3617208077260755  
=====  
frozenset({' all- purpose'})  
Support:  
0.3608428446005268  
=====  
frozenset({' butter'})  
Support:  
0.35294117647058826  
=====  
frozenset({' beef'})  
Support:  
0.36259877085162423  
=====  
frozenset({' vegetables'})  
Support:  
0.7155399473222125  
=====  
frozenset({' lunch meat'})  
Support:
```

0.3748902546093064
=====

frozenset({' sandwich bags'})
Support:
0.3476733977172959
=====

frozenset({' aluminum foil'})
Support:
0.36611062335381916
=====

frozenset({' dinner rolls'})
Support:
0.3722563652326602
=====

frozenset({' pork'})
Support:
0.34064969271290607
=====

frozenset({' ice cream'})
Support:
0.37664618086040386
=====

frozenset({' shampoo'})
Support:
0.353819139596137
=====

frozenset({' flour'})
Support:
0.33713784021071114
=====

frozenset({' soap'})
Support:
0.37050043898156276
=====

frozenset({' cereals', ' individual meals'})
Support:
0.1527655838454785
=====

frozenset({' cereals', ' cheeses'})
Support:
0.15715539947322213
=====

frozenset({' cereals', ' milk'})
Support:
0.15803336259877085
=====

frozenset({' cereals', ' toilet paper'})
Support:
0.15100965759438104
=====

frozenset({' cereals', ' poultry'})
Support:
0.16417910447761194
=====

frozenset({' cereals', ' spaghetti sauce'})
Support:
0.15452151009657594
=====

frozenset({' cereals', ' fruits'})
Support:
0.1527655838454785

```
=====
frozenset({' cereals', ' dishwashing liquid/detergent'})
Support:
0.16066725197541704
=====
frozenset({' cereals', ' eggs'})
Support:
0.1562774363476734
=====
frozenset({' cereals', ' paper towels'})
Support:
0.1527655838454785
=====
frozenset({' yogurt', ' cereals'})
Support:
0.16154521510096576
=====
frozenset({' cereals', ' bagels'})
Support:
0.15100965759438104
=====
frozenset({' coffee/tea', ' cereals'})
Support:
0.15891132572431957
=====
frozenset({' ', ' cereals'})
Support:
0.3678665496049166
=====
frozenset({' cereals', ' mixes'})
Support:
0.16330114135206322
=====
frozenset({' cereals', ' laundry detergent'})
Support:
0.15715539947322213
=====
frozenset({' beef', ' cereals'})
Support:
0.1562774363476734
=====
frozenset({' vegetables', ' cereals'})
Support:
0.2897278314310799
=====
frozenset({' cereals', ' lunch meat'})
Support:
0.15891132572431957
=====
frozenset({' cereals', ' aluminum foil'})
Support:
0.1527655838454785
=====
frozenset({' cereals', ' dinner rolls'})
Support:
0.15978928884986832
=====
frozenset({' cereals', ' ice cream'})
Support:
0.15013169446883232
=====
```

```
frozenset({' cereals', ' soap'})  
Support:  
0.15100965759438104  
=====  
frozenset({' individual meals', ' sandwich loaves'})  
Support:  
0.15013169446883232  
=====  
frozenset({' sandwich loaves', ' poultry'})  
Support:  
0.15364354697102722  
=====  
frozenset({' ', ' sandwich loaves'})  
Support:  
0.3222124670763828  
=====  
frozenset({' vegetables', ' sandwich loaves'})  
Support:  
0.266900790166813  
=====  
frozenset({' cheeses', ' individual meals'})  
Support:  
0.1527655838454785  
=====  
frozenset({' individual meals', ' poultry'})  
Support:  
0.16066725197541704  
=====  
frozenset({' individual meals', ' bagels'})  
Support:  
0.15188762071992976  
=====  
frozenset({' ', ' individual meals'})  
Support:  
0.35118525021949076  
=====  
frozenset({' individual meals', ' soda'})  
Support:  
0.15803336259877085  
=====  
frozenset({' vegetables', ' individual meals'})  
Support:  
0.2721685689201054  
=====  
frozenset({' individual meals', ' lunch meat'})  
Support:  
0.16417910447761194  
=====  
frozenset({' individual meals', ' aluminum foil'})  
Support:  
0.1527655838454785  
=====  
frozenset({' individual meals', ' soap'})  
Support:  
0.15364354697102722  
=====  
frozenset({' cheeses', ' toilet paper'})  
Support:  
0.1527655838454785  
=====  
frozenset({' cheeses', ' poultry'})
```

```
Support:  
0.16593503072870938  
=====  
frozenset({' cheeses', ' spaghetti sauce'})  
Support:  
0.15188762071992976  
=====  
frozenset({' ketchup', ' cheeses'})  
Support:  
0.15013169446883232  
=====  
frozenset({' cheeses', ' eggs'})  
Support:  
0.15803336259877085  
=====  
frozenset({' yogurt', ' cheeses'})  
Support:  
0.15978928884986832  
=====  
frozenset({' cheeses', ' bagels'})  
Support:  
0.15715539947322213  
=====  
frozenset({' coffee/tea', ' cheeses'})  
Support:  
0.16242317822651448  
=====  
frozenset({' cheeses', ' waffles'})  
Support:  
0.16066725197541704  
=====  
frozenset({' ', ' cheeses'})  
Support:  
0.35996488147497807  
=====  
frozenset({' cheeses', ' butter'})  
Support:  
0.15013169446883232  
=====  
frozenset({' vegetables', ' cheeses'})  
Support:  
0.2914837576821773  
=====  
frozenset({' cheeses', ' lunch meat'})  
Support:  
0.1527655838454785  
=====  
frozenset({' cheeses', ' sandwich bags'})  
Support:  
0.15715539947322213  
=====  
frozenset({' cheeses', ' aluminum foil'})  
Support:  
0.15188762071992976  
=====  
frozenset({' cheeses', ' dinner rolls'})  
Support:  
0.15539947322212466  
=====  
frozenset({' cheeses', ' ice cream'})  
Support:
```

0.16593503072870938
=====
frozenset({' cheeses', ' soap'})
Support:
0.1562774363476734
=====
frozenset({' ', ' hand soap'})
Support:
0.31870061457418786
=====
frozenset({' vegetables', ' hand soap'})
Support:
0.257243195785777
=====
frozenset({' milk', ' toilet paper'})
Support:
0.15100965759438104
=====
frozenset({' milk', ' poultry'})
Support:
0.16154521510096576
=====
frozenset({' milk', ' dishwashing liquid/detergent'})
Support:
0.15891132572431957
=====
frozenset({' milk', ' eggs'})
Support:
0.1527655838454785
=====
frozenset({' yogurt', ' milk'})
Support:
0.15100965759438104
=====
frozenset({' ', ' milk'})
Support:
0.3388937664618086
=====
frozenset({' soda', ' milk'})
Support:
0.15715539947322213
=====
frozenset({' milk', ' mixes'})
Support:
0.15013169446883232
=====
frozenset({' milk', ' laundry detergent'})
Support:
0.15013169446883232
=====
frozenset({' vegetables', ' milk'})
Support:
0.27568042142230026
=====
frozenset({' poultry', ' toilet paper'})
Support:
0.15452151009657594
=====
frozenset({' eggs', ' toilet paper'})
Support:
0.15013169446883232

```
=====
frozenset({' paper towels', ' toilet paper'})
Support:
0.15013169446883232
=====
frozenset({' bagels', ' toilet paper'})
Support:
0.15188762071992976
=====
frozenset({' coffee/tea', ' toilet paper'})
Support:
0.15100965759438104
=====
frozenset({' waffles', ' toilet paper'})
Support:
0.16066725197541704
=====
frozenset({' toilet paper', ' juice'})
Support:
0.15978928884986832
=====
frozenset({' ', ' toilet paper'})
Support:
0.3424056189640035
=====
frozenset({' soda', ' toilet paper'})
Support:
0.15013169446883232
=====
frozenset({' beef', ' toilet paper'})
Support:
0.15013169446883232
=====
frozenset({' vegetables', ' toilet paper'})
Support:
0.26865671641791045
=====
frozenset({' lunch meat', ' toilet paper'})
Support:
0.15188762071992976
=====
frozenset({' toilet paper', ' soap'})
Support:
0.1562774363476734
=====
frozenset({' spaghetti sauce', ' poultry'})
Support:
0.15803336259877085
=====
frozenset({' fruits', ' poultry'})
Support:
0.15539947322212466
=====
frozenset({' dishwashing liquid/detergent', ' poultry'})
Support:
0.17208077260755048
=====
frozenset({' ketchup', ' poultry'})
Support:
0.15452151009657594
=====
```

```
frozenset({' tortillas', ' poultry'})  
Support:  
0.15452151009657594  
=====  
frozenset({' poultry', ' eggs'})  
Support:  
0.16505706760316066  
=====  
frozenset({' sugar', ' poultry'})  
Support:  
0.16681299385425813  
=====  
frozenset({' yogurt', ' poultry'})  
Support:  
0.16330114135206322  
=====  
frozenset({' bagels', ' poultry'})  
Support:  
0.15100965759438104  
=====  
frozenset({' coffee/tea', ' poultry'})  
Support:  
0.16417910447761194  
=====  
frozenset({' waffles', ' poultry'})  
Support:  
0.15188762071992976  
=====  
frozenset({' poultry', ' juice'})  
Support:  
0.16242317822651448  
=====  
frozenset({' ', ' poultry'})  
Support:  
0.3863037752414399  
=====  
frozenset({' soda', ' poultry'})  
Support:  
0.16330114135206322  
=====  
frozenset({' mixes', ' poultry'})  
Support:  
0.17032484635645304  
=====  
frozenset({' poultry', ' laundry detergent'})  
Support:  
0.16505706760316066  
=====  
frozenset({' all- purpose', ' poultry'})  
Support:  
0.16154521510096576  
=====  
frozenset({' poultry', ' butter'})  
Support:  
0.15452151009657594  
=====  
frozenset({' beef', ' poultry'})  
Support:  
0.15715539947322213  
=====  
frozenset({' vegetables', ' poultry'})
```

```
Support:  
0.31431079894644426  
=====  
frozenset({' lunch meat', ' poultry'})  
Support:  
0.1694468832309043  
=====  
frozenset({' sandwich bags', ' poultry'})  
Support:  
0.15715539947322213  
=====  
frozenset({' aluminum foil', ' poultry'})  
Support:  
0.16066725197541704  
=====  
frozenset({' dinner rolls', ' poultry'})  
Support:  
0.18086040386303776  
=====  
frozenset({' pork', ' poultry'})  
Support:  
0.15364354697102722  
=====  
frozenset({' ice cream', ' poultry'})  
Support:  
0.16505706760316066  
=====  
frozenset({' shampoo', ' poultry'})  
Support:  
0.15803336259877085  
=====  
frozenset({' poultry', ' soap'})  
Support:  
0.17120280948200176  
=====  
frozenset({' spaghetti sauce', ' eggs'})  
Support:  
0.1527655838454785  
=====  
frozenset({' spaghetti sauce', ' bagels'})  
Support:  
0.1527655838454785  
=====  
frozenset({' spaghetti sauce', ' waffles'})  
Support:  
0.15539947322212466  
=====  
frozenset({' spaghetti sauce', ' juice'})  
Support:  
0.16066725197541704  
=====  
frozenset({' ', ' spaghetti sauce'})  
Support:  
0.34679543459174716  
=====  
frozenset({' soda', ' spaghetti sauce'})  
Support:  
0.15013169446883232  
=====  
frozenset({' spaghetti sauce', ' laundry detergent'})  
Support:
```

```
0.15013169446883232
=====
frozenset({' vegetables', ' spaghetti sauce'})
Support:
0.276558384547849
=====
frozenset({' lunch meat', ' spaghetti sauce'})
Support:
0.15364354697102722
=====
frozenset({' spaghetti sauce', ' dinner rolls'})
Support:
0.16242317822651448
=====
frozenset({' dishwashing liquid/detergent', ' fruits'})
Support:
0.1562774363476734
=====
frozenset({' fruits', ' bagels'})
Support:
0.15013169446883232
=====
frozenset({' ', ' fruits'})
Support:
0.33977172958735735
=====
frozenset({' beef', ' fruits'})
Support:
0.15715539947322213
=====
frozenset({' vegetables', ' fruits'})
Support:
0.26865671641791045
=====
frozenset({' lunch meat', ' fruits'})
Support:
0.15452151009657594
=====
frozenset({' ice cream', ' fruits'})
Support:
0.15188762071992976
=====
frozenset({' dishwashing liquid/detergent', ' eggs'})
Support:
0.15539947322212466
=====
frozenset({' dishwashing liquid/detergent', ' bagels'})
Support:
0.15978928884986832
=====
frozenset({' dishwashing liquid/detergent', ' waffles'})
Support:
0.16066725197541704
=====
frozenset({' ', ' dishwashing liquid/detergent'})
Support:
0.3582089552238806
=====
frozenset({' soda', ' dishwashing liquid/detergent'})
Support:
0.15364354697102722
```

```
=====
frozenset({' dishwashing liquid/detergent', ' mixes'})
Support:
0.17208077260755048
=====
frozenset({' dishwashing liquid/detergent', ' laundry detergent'})
Support:
0.15364354697102722
=====
frozenset({' all- purpose', ' dishwashing liquid/detergent'})
Support:
0.1527655838454785
=====
frozenset({' vegetables', ' dishwashing liquid/detergent'})
Support:
0.2853380158033363
=====
frozenset({' lunch meat', ' dishwashing liquid/detergent'})
Support:
0.16154521510096576
=====
frozenset({' dishwashing liquid/detergent', ' dinner rolls'})
Support:
0.15539947322212466
=====
frozenset({' ice cream', ' dishwashing liquid/detergent'})
Support:
0.15539947322212466
=====
frozenset({' dishwashing liquid/detergent', ' soap'})
Support:
0.15013169446883232
=====
frozenset({' ketchup', ' eggs'})
Support:
0.15452151009657594
=====
frozenset({' coffee/tea', ' ketchup'})
Support:
0.15364354697102722
=====
frozenset({' ketchup', ' waffles'})
Support:
0.1562774363476734
=====
frozenset({' ketchup', ' juice'})
Support:
0.15013169446883232
=====
frozenset({' ', ' ketchup'})
Support:
0.34591747146619845
=====
frozenset({' ketchup', ' mixes'})
Support:
0.1562774363476734
=====
frozenset({' ketchup', ' laundry detergent'})
Support:
0.15013169446883232
=====
```

```
frozenset({' vegetables', ' ketchup'})  
Support:  
0.27568042142230026  
=====  
frozenset({' ketchup', ' dinner rolls'})  
Support:  
0.1527655838454785  
=====  
frozenset({' ketchup', ' soap'})  
Support:  
0.15803336259877085  
=====  
frozenset({' ', ' tortillas'})  
Support:  
0.33099209833187004  
=====  
frozenset({' tortillas', ' mixes'})  
Support:  
0.1527655838454785  
=====  
frozenset({' vegetables', ' tortillas'})  
Support:  
0.26163301141352063  
=====  
frozenset({' paper towels', ' eggs'})  
Support:  
0.15452151009657594  
=====  
frozenset({' yogurt', ' eggs'})  
Support:  
0.15891132572431957  
=====  
frozenset({' pasta', ' eggs'})  
Support:  
0.15891132572431957  
=====  
frozenset({' bagels', ' eggs'})  
Support:  
0.15715539947322213  
=====  
frozenset({' waffles', ' eggs'})  
Support:  
0.16066725197541704  
=====  
frozenset({' eggs', ' juice'})  
Support:  
0.15364354697102722  
=====  
frozenset({' ', ' eggs'})  
Support:  
0.3555750658472344  
=====  
frozenset({' soda', ' eggs'})  
Support:  
0.17295873573309922  
=====  
frozenset({' mixes', ' eggs'})  
Support:  
0.15100965759438104  
=====  
frozenset({' all- purpose', ' eggs'})
```

```
Support:  
0.15100965759438104  
=====  
frozenset({' vegetables', ' eggs'})  
Support:  
0.30728709394205445  
=====  
frozenset({' dinner rolls', ' eggs'})  
Support:  
0.16242317822651448  
=====  
frozenset({' ice cream', ' eggs'})  
Support:  
0.15188762071992976  
=====  
frozenset({' eggs', ' soap'})  
Support:  
0.15539947322212466  
=====  
frozenset({' yogurt', ' sugar'})  
Support:  
0.15013169446883232  
=====  
frozenset({' sugar', ' waffles'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' sugar'})  
Support:  
0.33801580333625986  
=====  
frozenset({' sugar', ' soda'})  
Support:  
0.1527655838454785  
=====  
frozenset({' vegetables', ' sugar'})  
Support:  
0.2844600526777875  
=====  
frozenset({' sugar', ' lunch meat'})  
Support:  
0.15100965759438104  
=====  
frozenset({' sugar', ' dinner rolls'})  
Support:  
0.15188762071992976  
=====  
frozenset({' ', ' paper towels'})  
Support:  
0.3362598770851624  
=====  
frozenset({' paper towels', ' mixes'})  
Support:  
0.15100965759438104  
=====  
frozenset({' vegetables', ' paper towels'})  
Support:  
0.27480245829675154  
=====  
frozenset({' lunch meat', ' paper towels'})  
Support:
```

```
0.1562774363476734
=====
frozenset({' paper towels', ' aluminum foil'})
Support:
0.15100965759438104
=====
frozenset({' ice cream', ' paper towels'})
Support:
0.1562774363476734
=====
frozenset({' yogurt', ' waffles'})
Support:
0.16066725197541704
=====
frozenset({' yogurt', ' juice'})
Support:
0.15978928884986832
=====
frozenset({' yogurt', ' '})
Support:
0.35294117647058826
=====
frozenset({' yogurt', ' soda'})
Support:
0.15715539947322213
=====
frozenset({' yogurt', ' mixes'})
Support:
0.15188762071992976
=====
frozenset({' yogurt', ' all- purpose'})
Support:
0.15715539947322213
=====
frozenset({' yogurt', ' vegetables'})
Support:
0.30026338893766463
=====
frozenset({' yogurt', ' aluminum foil'})
Support:
0.16154521510096576
=====
frozenset({' yogurt', ' dinner rolls'})
Support:
0.15715539947322213
=====
frozenset({' pasta', ' waffles'})
Support:
0.15100965759438104
=====
frozenset({' ', ' pasta'})
Support:
0.33362598770851626
=====
frozenset({' pasta', ' soda'})
Support:
0.15891132572431957
=====
frozenset({' vegetables', ' pasta'})
Support:
0.2660228270412643
```

```
=====
frozenset({' pasta', ' dinner rolls'})
Support:
0.15188762071992976
=====
frozenset({' ice cream', ' pasta'})
Support:
0.15452151009657594
=====
frozenset({' bagels', ' juice'})
Support:
0.15013169446883232
=====
frozenset({' ', ' bagels'})
Support:
0.35294117647058826
=====
frozenset({' bagels', ' mixes'})
Support:
0.15188762071992976
=====
frozenset({' vegetables', ' bagels'})
Support:
0.28094820017559263
=====
frozenset({' aluminum foil', ' bagels'})
Support:
0.1562774363476734
=====
frozenset({' coffee/tea', ' waffles'})
Support:
0.15978928884986832
=====
frozenset({' ', ' coffee/tea'})
Support:
0.3555750658472344
=====
frozenset({' coffee/tea', ' soda'})
Support:
0.1527655838454785
=====
frozenset({' coffee/tea', ' vegetables'})
Support:
0.276558384547849
=====
frozenset({' coffee/tea', ' lunch meat'})
Support:
0.15452151009657594
=====
frozenset({' coffee/tea', ' aluminum foil'})
Support:
0.1527655838454785
=====
frozenset({' coffee/tea', ' dinner rolls'})
Support:
0.15188762071992976
=====
frozenset({' coffee/tea', ' ice cream'})
Support:
0.15100965759438104
=====
```

```
frozenset({' waffles', ' juice'})  
Support:  
0.15539947322212466  
=====  
frozenset({' ', ' waffles'})  
Support:  
0.36259877085162423  
=====  
frozenset({' soda', ' waffles'})  
Support:  
0.16505706760316066  
=====  
frozenset({' all- purpose', ' waffles'})  
Support:  
0.15452151009657594  
=====  
frozenset({' vegetables', ' waffles'})  
Support:  
0.2993854258121159  
=====  
frozenset({' lunch meat', ' waffles'})  
Support:  
0.16856892010535557  
=====  
frozenset({' aluminum foil', ' waffles'})  
Support:  
0.15452151009657594  
=====  
frozenset({' waffles', ' dinner rolls'})  
Support:  
0.1562774363476734  
=====  
frozenset({' ice cream', ' waffles'})  
Support:  
0.15364354697102722  
=====  
frozenset({' waffles', ' soap'})  
Support:  
0.16417910447761194  
=====  
frozenset({' ', ' juice'})  
Support:  
0.34416154521510095  
=====  
frozenset({' soda', ' juice'})  
Support:  
0.15013169446883232  
=====  
frozenset({' vegetables', ' juice'})  
Support:  
0.2783143107989464  
=====  
frozenset({' aluminum foil', ' juice'})  
Support:  
0.16066725197541704  
=====  
frozenset({' ice cream', ' juice'})  
Support:  
0.1527655838454785  
=====  
frozenset({' shampoo', ' juice'})
```

```
Support:  
0.15978928884986832  
=====  
frozenset({' ', ' soda'})  
Support:  
0.35908691834942935  
=====  
frozenset({' ', ' mixes'})  
Support:  
0.35908691834942935  
=====  
frozenset({' ', ' laundry detergent'})  
Support:  
0.35118525021949076  
=====  
frozenset({' ', ' all- purpose'})  
Support:  
0.34328358208955223  
=====  
frozenset({' ', ' butter'})  
Support:  
0.34328358208955223  
=====  
frozenset({' beef', ' '})  
Support:  
0.3424056189640035  
=====  
frozenset({' ', ' vegetables'})  
Support:  
0.6935908691834943  
=====  
frozenset({' ', ' lunch meat'})  
Support:  
0.35996488147497807  
=====  
frozenset({' ', ' sandwich bags'})  
Support:  
0.3362598770851624  
=====  
frozenset({' ', ' aluminum foil'})  
Support:  
0.35030728709394204  
=====  
frozenset({' ', ' dinner rolls'})  
Support:  
0.35645302897278314  
=====  
frozenset({' pork', ' '})  
Support:  
0.32484635645302895  
=====  
frozenset({' ', ' ice cream'})  
Support:  
0.3608428446005268  
=====  
frozenset({' shampoo', ' '})  
Support:  
0.33801580333625986  
=====  
frozenset({' ', ' flour'})  
Support:
```

0.32396839332748023
=====
frozenset({' ', ' soap'})
Support:
0.3520632133450395
=====
frozenset({' vegetables', ' soda'})
Support:
0.2897278314310799
=====
frozenset({' soda', ' aluminum foil'})
Support:
0.1527655838454785
=====
frozenset({' soda', ' dinner rolls'})
Support:
0.16154521510096576
=====
frozenset({' ice cream', ' soda'})
Support:
0.15539947322212466
=====
frozenset({' soda', ' soap'})
Support:
0.16769095697980685
=====
frozenset({' mixes', ' laundry detergent'})
Support:
0.15100965759438104
=====
frozenset({' vegetables', ' mixes'})
Support:
0.28094820017559263
=====
frozenset({' aluminum foil', ' mixes'})
Support:
0.16330114135206322
=====
frozenset({' dinner rolls', ' mixes'})
Support:
0.1562774363476734
=====
frozenset({' all- purpose', ' laundry detergent'})
Support:
0.1527655838454785
=====
frozenset({' vegetables', ' laundry detergent'})
Support:
0.2897278314310799
=====
frozenset({' dinner rolls', ' laundry detergent'})
Support:
0.1527655838454785
=====
frozenset({' vegetables', ' all- purpose'})
Support:
0.2730465320456541
=====
frozenset({' all- purpose', ' soap'})
Support:
0.15539947322212466

```
=====
frozenset({' vegetables', ' butter'})
Support:
0.26865671641791045
=====
frozenset({' lunch meat', ' butter'})
Support:
0.15013169446883232
=====
frozenset({' beef', ' vegetables'})
Support:
0.27568042142230026
=====
frozenset({' beef', ' lunch meat'})
Support:
0.15100965759438104
=====
frozenset({' beef', ' ice cream'})
Support:
0.15539947322212466
=====
frozenset({' beef', ' soap'})
Support:
0.15452151009657594
=====
frozenset({' vegetables', ' lunch meat'})
Support:
0.29499561018437226
=====
frozenset({' vegetables', ' sandwich bags'})
Support:
0.2721685689201054
=====
frozenset({' vegetables', ' aluminum foil'})
Support:
0.2906057945566286
=====
frozenset({' vegetables', ' dinner rolls'})
Support:
0.2914837576821773
=====
frozenset({' pork', ' vegetables'})
Support:
0.25636523266022826
=====
frozenset({' vegetables', ' ice cream'})
Support:
0.2835820895522388
=====
frozenset({' shampoo', ' vegetables'})
Support:
0.2739244951712028
=====
frozenset({' vegetables', ' flour'})
Support:
0.266900790166813
=====
frozenset({' vegetables', ' soap'})
Support:
0.2853380158033363
=====
```

```
frozenset({' ice cream', ' lunch meat'})  
Support:  
0.15715539947322213  
=====  
frozenset({' lunch meat', ' soap'})  
Support:  
0.1562774363476734  
=====  
frozenset({' aluminum foil', ' dinner rolls'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ice cream', ' aluminum foil'})  
Support:  
0.1562774363476734  
=====  
frozenset({' shampoo', ' dinner rolls'})  
Support:  
0.15452151009657594  
=====  
frozenset({' ice cream', ' soap'})  
Support:  
0.15364354697102722  
=====  
frozenset({' ', ' cereals', ' cheeses'})  
Support:  
0.15188762071992976  
=====  
frozenset({' ', ' cereals', ' milk'})  
Support:  
0.15013169446883232  
=====  
frozenset({' ', ' cereals', ' poultry'})  
Support:  
0.15715539947322213  
=====  
frozenset({' ', ' cereals', ' spaghetti sauce'})  
Support:  
0.15188762071992976  
=====  
frozenset({' ', ' cereals', ' dishwashing liquid/detergent'})  
Support:  
0.15715539947322213  
=====  
frozenset({' yogurt', ' cereals', ' '})  
Support:  
0.15452151009657594  
=====  
frozenset({' ', ' cereals', ' mixes'})  
Support:  
0.15978928884986832  
=====  
frozenset({' ', ' cereals', ' laundry detergent'})  
Support:  
0.15013169446883232  
=====  
frozenset({' vegetables', ' cereals', ' '})  
Support:  
0.2791922739244952  
=====  
frozenset({' ', ' cereals', ' lunch meat'})
```

```
Support:  
0.15013169446883232  
=====  
frozenset({' ', ' cereals', ' dinner rolls'})  
Support:  
0.1527655838454785  
=====  
frozenset({' ', ' vegetables', ' sandwich loaves'})  
Support:  
0.2528533801580334  
=====  
frozenset({' ', ' individual meals', ' poultry'})  
Support:  
0.15452151009657594  
=====  
frozenset({' ', ' individual meals', ' soda'})  
Support:  
0.15013169446883232  
=====  
frozenset({' ', ' vegetables', ' individual meals'})  
Support:  
0.2598770851624232  
=====  
frozenset({' ', ' individual meals', ' lunch meat'})  
Support:  
0.1562774363476734  
=====  
frozenset({' ', ' cheeses', ' poultry'})  
Support:  
0.15978928884986832  
=====  
frozenset({' ', ' cheeses', ' eggs'})  
Support:  
0.15100965759438104  
=====  
frozenset({' yogurt', ' cheeses', ' '})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' cheeses', ' bagels'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' cheeses', ' coffee/tea'})  
Support:  
0.15452151009657594  
=====  
frozenset({' ', ' cheeses', ' waffles'})  
Support:  
0.15364354697102722  
=====  
frozenset({' vegetables', ' cheeses', ' '})  
Support:  
0.2835820895522388  
=====  
frozenset({' ', ' cheeses', ' sandwich bags'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' cheeses', ' ice cream'})  
Support:
```

0.16154521510096576
=====

frozenset({' ', ' vegetables', ' hand soap'})
Support:
0.24495171202809482
=====

frozenset({' ', ' milk', ' poultry'})
Support:
0.15100965759438104
=====

frozenset({' ', ' milk', ' dishwashing liquid/detergent'})
Support:
0.15013169446883232
=====

frozenset({' ', ' vegetables', ' milk'})
Support:
0.2598770851624232
=====

frozenset({' ', ' waffles', ' toilet paper'})
Support:
0.15013169446883232
=====

frozenset({' ', ' vegetables', ' toilet paper'})
Support:
0.2528533801580334
=====

frozenset({' ', ' spaghetti sauce', ' poultry'})
Support:
0.1527655838454785
=====

frozenset({' ', ' dishwashing liquid/detergent', ' poultry'})
Support:
0.1694468832309043
=====

frozenset({' poultry', ' ', ' eggs'})
Support:
0.15539947322212466
=====

frozenset({' ', ' sugar', ' poultry'})
Support:
0.16066725197541704
=====

frozenset({' yogurt', ' ', ' poultry'})
Support:
0.15539947322212466
=====

frozenset({' ', ' poultry', ' coffee/tea'})
Support:
0.1562774363476734
=====

frozenset({' poultry', ' ', ' juice'})
Support:
0.1527655838454785
=====

frozenset({' ', ' soda', ' poultry'})
Support:
0.1527655838454785
=====

frozenset({' ', ' mixes', ' poultry'})
Support:
0.16593503072870938

```
=====
frozenset({' ', ' poultry', ' laundry detergent'})
Support:
0.15978928884986832
=====
frozenset({' ', ' all- purpose', ' poultry'})
Support:
0.1527655838454785
=====
frozenset({' ', ' vegetables', ' poultry'})
Support:
0.30114135206321335
=====
frozenset({' ', ' lunch meat', ' poultry'})
Support:
0.16242317822651448
=====
frozenset({' ', ' sandwich bags', ' poultry'})
Support:
0.15100965759438104
=====
frozenset({' ', ' aluminum foil', ' poultry'})
Support:
0.15188762071992976
=====
frozenset({' ', ' dinner rolls', ' poultry'})
Support:
0.17295873573309922
=====
frozenset({' ', ' ice cream', ' poultry'})
Support:
0.15539947322212466
=====
frozenset({' ', ' shampoo', ' poultry'})
Support:
0.15100965759438104
=====
frozenset({' ', ' poultry', ' soap'})
Support:
0.15978928884986832
=====
frozenset({' vegetables', ' dinner rolls', ' poultry'})
Support:
0.15188762071992976
=====
frozenset({' ', ' spaghetti sauce', ' waffles'})
Support:
0.15013169446883232
=====
frozenset({' ', ' spaghetti sauce', ' juice'})
Support:
0.15452151009657594
=====
frozenset({' ', ' vegetables', ' spaghetti sauce'})
Support:
0.26865671641791045
=====
frozenset({' ', ' spaghetti sauce', ' dinner rolls'})
Support:
0.1562774363476734
=====
```

```
frozenset({' ', ' dishwashing liquid/detergent', ' fruits'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' vegetables', ' fruits'})  
Support:  
0.257243195785777  
=====  
frozenset({' ', ' dishwashing liquid/detergent', ' bagels'})  
Support:  
0.15364354697102722  
=====  
frozenset({' ', ' dishwashing liquid/detergent', ' waffles'})  
Support:  
0.15452151009657594  
=====  
frozenset({' ', ' dishwashing liquid/detergent', ' mixes'})  
Support:  
0.16681299385425813  
=====  
frozenset({' ', ' dishwashing liquid/detergent', ' laundry detergent'})  
Support:  
0.15013169446883232  
=====  
frozenset({' ', ' vegetables', ' dishwashing liquid/detergent'})  
Support:  
0.2774363476733977  
=====  
frozenset({' ', ' lunch meat', ' dishwashing liquid/detergent'})  
Support:  
0.15539947322212466  
=====  
frozenset({' ', ' ketchup', ' mixes'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' ketchup', ' vegetables'})  
Support:  
0.2642669007901668  
=====  
frozenset({' ', ' vegetables', ' tortillas'})  
Support:  
0.24846356453028973  
=====  
frozenset({' yogurt', ' ', ' eggs'})  
Support:  
0.15100965759438104  
=====  
frozenset({' ', ' waffles', ' eggs'})  
Support:  
0.1527655838454785  
=====  
frozenset({' ', ' soda', ' eggs'})  
Support:  
0.16154521510096576  
=====  
frozenset({' ', ' vegetables', ' eggs'})  
Support:  
0.2923617208077261  
=====  
frozenset({' ', ' dinner rolls', ' eggs'})
```

```
Support:  
0.15364354697102722  
=====  
frozenset({' ', ' vegetables', ' sugar'})  
Support:  
0.27480245829675154  
=====  
frozenset({' ', ' vegetables', ' paper towels'})  
Support:  
0.2642669007901668  
=====  
frozenset({' yogurt', ' ', ' waffles'})  
Support:  
0.1527655838454785  
=====  
frozenset({' yogurt', ' ', ' juice'})  
Support:  
0.15100965759438104  
=====  
frozenset({' yogurt', ' ', ' soda'})  
Support:  
0.15013169446883232  
=====  
frozenset({' yogurt', ' vegetables', ' '})  
Support:  
0.28884986830553117  
=====  
frozenset({' yogurt', ' ', ' aluminum foil'})  
Support:  
0.1527655838454785  
=====  
frozenset({' ', ' vegetables', ' pasta'})  
Support:  
0.2528533801580334  
=====  
frozenset({' ', ' vegetables', ' bagels'})  
Support:  
0.26865671641791045  
=====  
frozenset({' ', ' waffles', ' coffee/tea'})  
Support:  
0.15013169446883232  
=====  
frozenset({' ', ' vegetables', ' coffee/tea'})  
Support:  
0.2642669007901668  
=====  
frozenset({' ', ' soda', ' waffles'})  
Support:  
0.15539947322212466  
=====  
frozenset({' ', ' vegetables', ' waffles'})  
Support:  
0.286215978928885  
=====  
frozenset({' ', ' lunch meat', ' waffles'})  
Support:  
0.15803336259877085  
=====  
frozenset({' ', ' waffles', ' soap'})  
Support:
```

```
0.15188762071992976
=====
frozenset({' ', ' vegetables', ' juice'})
Support:
0.2660228270412643
=====
frozenset({' ', ' aluminum foil', ' juice'})
Support:
0.1527655838454785
=====
frozenset({' ', ' vegetables', ' soda'})
Support:
0.2739244951712028
=====
frozenset({' ', ' soda', ' dinner rolls'})
Support:
0.15100965759438104
=====
frozenset({' ', ' soda', ' soap'})
Support:
0.15539947322212466
=====
frozenset({' ', ' vegetables', ' mixes'})
Support:
0.2730465320456541
=====
frozenset({' ', ' aluminum foil', ' mixes'})
Support:
0.15539947322212466
=====
frozenset({' ', ' dinner rolls', ' mixes'})
Support:
0.15100965759438104
=====
frozenset({' ', ' vegetables', ' laundry detergent'})
Support:
0.28094820017559263
=====
frozenset({' ', ' vegetables', ' all- purpose'})
Support:
0.2589991220368745
=====
frozenset({' ', ' vegetables', ' butter'})
Support:
0.2607550482879719
=====
frozenset({' beef', ' vegetables', ' '})
Support:
0.2589991220368745
=====
frozenset({' ', ' vegetables', ' lunch meat'})
Support:
0.28182616330114135
=====
frozenset({' ', ' vegetables', ' sandwich bags'})
Support:
0.26251097453906935
=====
frozenset({' ', ' vegetables', ' aluminum foil'})
Support:
0.276558384547849
```

```
=====
frozenset({' ', ' vegetables', ' dinner rolls'})
Support:
0.2783143107989464
=====
frozenset({' pork', ' vegetables', ' '})
Support:
0.24231782265144863
=====
frozenset({' vegetables', ' ice cream', ' '})
Support:
0.26953467954345917
=====
frozenset({' ', ' vegetables', ' shampoo'})
Support:
0.2607550482879719
=====
frozenset({' ', ' vegetables', ' flour'})
Support:
0.25636523266022826
=====
frozenset({' ', ' vegetables', ' soap'})
Support:
0.26953467954345917
=====
```

Case 1- Association Rules

```
In [ ]: for index, row in rules.iterrows():
    print("Rule:"+ row['antecedents']+ "->"+ row['consequents'])
    print("Confidance: ")
    print( row['confidence'])
    print("Support:")
    print( row['support'])
    print("=====")
```

Rule: cereals->
Confidance:
0.9676674364896074
Support:
0.3678665496049166
=====

Rule: cereals-> vegetables
Confidance:
0.7621247113163971
Support:
0.2897278314310799
=====

Rule: sandwich loaves->
Confidance:
0.9557291666666667
Support:
0.3222124670763828
=====

Rule: sandwich loaves-> vegetables
Confidance:
0.7916666666666667
Support:
0.266900790166813
=====

Rule: individual meals->
Confidance:
0.9615384615384616
Support:
0.35118525021949076
=====

Rule: individual meals-> vegetables
Confidance:
0.7451923076923078
Support:
0.2721685689201054
=====

Rule: cheeses->
Confidance:
0.9692671394799055
Support:
0.35996488147497807
=====

Rule: cheeses-> vegetables
Confidance:
0.7848699763593381
Support:
0.2914837576821773
=====

Rule: hand soap->
Confidance:
0.9577836411609497
Support:
0.31870061457418786
=====

Rule: hand soap-> vegetables
Confidance:
0.7730870712401055
Support:
0.257243195785777
=====

Rule: milk->

Confidance:
0.9484029484029484
Support:
0.3388937664618086
=====

Rule: milk-> vegetables
Confidance:
0.7714987714987716
Support:
0.27568042142230026
=====

Rule: toilet paper->
Confidance:
0.948905109489051
Support:
0.3424056189640035
=====

Rule: toilet paper-> vegetables
Confidance:
0.7445255474452555
Support:
0.26865671641791045
=====

Rule: poultry->
Confidance:
0.9649122807017544
Support:
0.3863037752414399
=====

Rule: poultry-> vegetables
Confidance:
0.7850877192982456
Support:
0.31431079894644426
=====

Rule: spaghetti sauce->
Confidance:
0.9729064039408868
Support:
0.34679543459174716
=====

Rule: spaghetti sauce-> vegetables
Confidance:
0.7758620689655172
Support:
0.276558384547849
=====

Rule: fruits->
Confidance:
0.9602977667493797
Support:
0.33977172958735735
=====

Rule: fruits-> vegetables
Confidance:
0.7593052109181141
Support:
0.26865671641791045
=====

Rule: dishwashing liquid/detergent->
Confidance:

0.9714285714285714
Support:
0.3582089552238806
=====

Rule: dishwashing liquid/detergent-> vegetables
Confidance:
0.7738095238095238
Support:
0.2853380158033363
=====

Rule: ketchup->
Confidance:
0.9633251833740831
Support:
0.34591747146619845
=====

Rule: ketchup-> vegetables
Confidance:
0.7677261613691931
Support:
0.27568042142230026
=====

Rule: tortillas->
Confidance:
0.9617346938775511
Support:
0.33099209833187004
=====

Rule: tortillas-> vegetables
Confidance:
0.7602040816326531
Support:
0.26163301141352063
=====

Rule: eggs->
Confidance:
0.9551886792452831
Support:
0.3555750658472344
=====

Rule: eggs-> vegetables
Confidance:
0.8254716981132076
Support:
0.30728709394205445
=====

Rule: sugar->
Confidance:
0.9673366834170853
Support:
0.33801580333625986
=====

Rule: sugar-> vegetables
Confidance:
0.8140703517587939
Support:
0.2844600526777875
=====

Rule: paper towels->
Confidance:
0.964735516372796

Support:
0.3362598770851624
=====

Rule: paper towels-> vegetables
Confidance:
0.7884130982367759
Support:
0.27480245829675154
=====

Rule: yogurt->
Confidance:
0.9617224880382775
Support:
0.35294117647058826
=====

Rule: yogurt-> vegetables
Confidance:
0.8181818181818182
Support:
0.30026338893766463
=====

Rule: pasta->
Confidance:
0.9571788413098237
Support:
0.33362598770851626
=====

Rule: pasta-> vegetables
Confidance:
0.7632241813602015
Support:
0.2660228270412643
=====

Rule: bagels->
Confidance:
0.9617224880382775
Support:
0.35294117647058826
=====

Rule: bagels-> vegetables
Confidance:
0.7655502392344498
Support:
0.28094820017559263
=====

Rule: coffee/tea->
Confidance:
0.9597156398104266
Support:
0.3555750658472344
=====

Rule: coffee/tea-> vegetables
Confidance:
0.7464454976303317
Support:
0.276558384547849
=====

Rule: waffles->
Confidance:
0.9604651162790698
Support:

0.36259877085162423
=====

Rule: waffles-> vegetables
Confidance:
0.7930232558139536
Support:
0.2993854258121159
=====

Rule: juice->
Confidance:
0.9560975609756096
Support:
0.34416154521510095
=====

Rule: juice-> vegetables
Confidance:
0.773170731707317
Support:
0.2783143107989464
=====

Rule: soda->
Confidance:
0.955607476635514
Support:
0.35908691834942935
=====

Rule: mixes->
Confidance:
0.9738095238095239
Support:
0.35908691834942935
=====

Rule: laundry detergent->
Confidance:
0.9708737864077669
Support:
0.35118525021949076
=====

Rule: all- purpose->
Confidance:
0.9513381995133819
Support:
0.34328358208955223
=====

Rule: butter->
Confidance:
0.972636815920398
Support:
0.34328358208955223
=====

Rule: beef->
Confidance:
0.9443099273607748
Support:
0.3424056189640035
=====

Rule: -> vegetables
Confidance:
0.7117117117117118
Support:
0.6935908691834943

=====

Rule: vegetables->
Confidance:
0.9693251533742332
Support:
0.6935908691834943

=====

Rule: lunch meat->
Confidance:
0.9601873536299766
Support:
0.35996488147497807

=====

Rule: sandwich bags->
Confidance:
0.9671717171717171
Support:
0.3362598770851624

=====

Rule: aluminum foil->
Confidance:
0.9568345323741007
Support:
0.35030728709394204

=====

Rule: dinner rolls->
Confidance:
0.9575471698113208
Support:
0.35645302897278314

=====

Rule: pork->
Confidance:
0.9536082474226804
Support:
0.32484635645302895

=====

Rule: ice cream->
Confidance:
0.9580419580419581
Support:
0.3608428446005268

=====

Rule: shampoo->
Confidance:
0.955334987593052
Support:
0.33801580333625986

=====

Rule: flour->
Confidance:
0.9609375
Support:
0.32396839332748023

=====

Rule: soap->
Confidance:
0.9502369668246445
Support:
0.3520632133450395

=====

Rule: soda-> vegetables
Confidance:
0.7710280373831775
Support:
0.2897278314310799
=====

Rule: mixes-> vegetables
Confidance:
0.761904761904762
Support:
0.28094820017559263
=====

Rule: laundry detergent-> vegetables
Confidance:
0.8009708737864077
Support:
0.2897278314310799
=====

Rule: all- purpose-> vegetables
Confidance:
0.75669099756691
Support:
0.2730465320456541
=====

Rule: butter-> vegetables
Confidance:
0.7611940298507462
Support:
0.26865671641791045
=====

Rule: beef-> vegetables
Confidance:
0.7602905569007263
Support:
0.27568042142230026
=====

Rule: lunch meat-> vegetables
Confidance:
0.7868852459016393
Support:
0.29499561018437226
=====

Rule: sandwich bags-> vegetables
Confidance:
0.78282828282829
Support:
0.2721685689201054
=====

Rule: aluminum foil-> vegetables
Confidance:
0.7937649880095923
Support:
0.2906057945566286
=====

Rule: dinner rolls-> vegetables
Confidance:
0.7830188679245282
Support:
0.2914837576821773
=====

Rule: pork-> vegetables

Confidance:
0.7525773195876289
Support:
0.25636523266022826
=====

Rule: ice cream-> vegetables
Confidance:
0.7529137529137528
Support:
0.2835820895522388
=====

Rule: shampoo-> vegetables
Confidance:
0.7741935483870968
Support:
0.2739244951712028
=====

Rule: flour-> vegetables
Confidance:
0.7916666666666667
Support:
0.266900790166813
=====

Rule: soap-> vegetables
Confidance:
0.7701421800947869
Support:
0.2853380158033363
=====

Rule: cereals->
Confidance:
0.9664804469273742
Support:
0.15188762071992976
=====

Rule: cereals->
Confidance:
0.9500000000000001
Support:
0.15013169446883232
=====

Rule: cereals->
Confidance:
0.9572192513368984
Support:
0.15715539947322213
=====

Rule: cereals->
Confidance:
0.9829545454545454
Support:
0.15188762071992976
=====

Rule: cereals->
Confidance:
0.9781420765027322
Support:
0.15715539947322213
=====

Rule: yogurt->
Confidance:

0.9565217391304348
Support:
0.15452151009657594
=====

Rule: cereals->
Confidance:
0.978494623655914
Support:
0.15978928884986832
=====

Rule: cereals->
Confidance:
0.9553072625698324
Support:
0.15013169446883232
=====

Rule: vegetables->
Confidance:
0.9636363636363637
Support:
0.2791922739244952
=====

Rule: -> vegetables
Confidance:
0.7589498806682577
Support:
0.2791922739244952
=====

Rule: cereals->
Confidance:
0.7344110854503464
Support:
0.2791922739244952
=====

Rule: cereals->
Confidance:
0.9447513812154698
Support:
0.15013169446883232
=====

Rule: cereals->
Confidance:
0.9560439560439561
Support:
0.1527655838454785
=====

Rule: -> vegetables
Confidance:
0.784741144414169
Support:
0.2528533801580334
=====

Rule: vegetables->
Confidance:
0.9473684210526316
Support:
0.2528533801580334
=====

Rule: sandwich loaves->
Confidance:
0.7500000000000001

Support:
0.2528533801580334
=====

Rule: individual meals->
Confidance:
0.9617486338797814
Support:
0.15452151009657594
=====

Rule: individual meals->
Confidance:
0.9500000000000001
Support:
0.15013169446883232
=====

Rule: -> vegetables
Confidance:
0.7400000000000001
Support:
0.2598770851624232
=====

Rule: vegetables->
Confidance:
0.9548387096774194
Support:
0.2598770851624232
=====

Rule: individual meals->
Confidance:
0.7115384615384617
Support:
0.2598770851624232
=====

Rule: individual meals->
Confidance:
0.9518716577540107
Support:
0.1562774363476734
=====

Rule: cheeses->
Confidance:
0.9629629629629631
Support:
0.15978928884986832
=====

Rule: cheeses->
Confidance:
0.9555555555555556
Support:
0.15100965759438104
=====

Rule: yogurt->
Confidance:
0.945054945054945
Support:
0.15100965759438104
=====

Rule: cheeses->
Confidance:
0.9608938547486033
Support:

0.15100965759438104
=====

Rule: coffee/tea->
Confidance:
0.9513513513513514
Support:
0.15452151009657594
=====

Rule: cheeses->
Confidance:
0.9562841530054645
Support:
0.15364354697102722
=====

Rule: vegetables->
Confidance:
0.9728915662650602
Support:
0.2835820895522388
=====

Rule: -> vegetables
Confidance:
0.7878048780487804
Support:
0.2835820895522388
=====

Rule: cheeses->
Confidance:
0.7635933806146572
Support:
0.2835820895522388
=====

Rule: cheeses->
Confidance:
0.9608938547486033
Support:
0.15100965759438104
=====

Rule: cheeses->
Confidance:
0.9735449735449736
Support:
0.16154521510096576
=====

Rule: -> vegetables
Confidance:
0.7685950413223142
Support:
0.24495171202809482
=====

Rule: vegetables->
Confidance:
0.9522184300341298
Support:
0.24495171202809482
=====

Rule: hand soap->
Confidance:
0.7361477572559366
Support:
0.24495171202809482

=====

Rule: milk->
Confidance:
0.9347826086956522
Support:
0.15100965759438104

=====

Rule: milk->
Confidance:
0.9447513812154698
Support:
0.15013169446883232

=====

Rule: -> vegetables
Confidance:
0.766839378238342
Support:
0.2598770851624232

=====

Rule: vegetables->
Confidance:
0.9426751592356689
Support:
0.2598770851624232

=====

Rule: milk->
Confidance:
0.7272727272727273
Support:
0.2598770851624232

=====

Rule: toilet paper->
Confidance:
0.9344262295081968
Support:
0.15013169446883232

=====

Rule: -> vegetables
Confidance:
0.7384615384615385
Support:
0.2528533801580334

=====

Rule: vegetables->
Confidance:
0.9411764705882354
Support:
0.2528533801580334

=====

Rule: toilet paper->
Confidance:
0.7007299270072993
Support:
0.2528533801580334

=====

Rule: spaghetti sauce->
Confidance:
0.9666666666666668
Support:
0.1527655838454785

=====

Rule: poultry->
Confidance:
0.9846938775510203
Support:
0.1694468832309043
=====

Rule: eggs->
Confidance:
0.9414893617021276
Support:
0.15539947322212466
=====

Rule: sugar->
Confidance:
0.9631578947368421
Support:
0.16066725197541704
=====

Rule: yogurt->
Confidance:
0.9516129032258064
Support:
0.15539947322212466
=====

Rule: coffee/tea->
Confidance:
0.9518716577540107
Support:
0.1562774363476734
=====

Rule: poultry->
Confidance:
0.9405405405405407
Support:
0.1527655838454785
=====

Rule: soda->
Confidance:
0.935483870967742
Support:
0.1527655838454785
=====

Rule: poultry->
Confidance:
0.9742268041237112
Support:
0.16593503072870938
=====

Rule: poultry->
Confidance:
0.9680851063829788
Support:
0.15978928884986832
=====

Rule: all- purpose->
Confidance:
0.9456521739130436
Support:
0.1527655838454785
=====

Rule: -> vegetables

Confidance:
0.7795454545454545
Support:
0.30114135206321335
=====

Rule: vegetables->
Confidance:
0.9581005586592178
Support:
0.30114135206321335
=====

Rule: poultry->
Confidance:
0.7521929824561403
Support:
0.30114135206321335
=====

Rule: lunch meat->
Confidance:
0.9585492227979274
Support:
0.16242317822651448
=====

Rule: sandwich bags->
Confidance:
0.9608938547486033
Support:
0.15100965759438104
=====

Rule: aluminum foil->
Confidance:
0.9453551912568305
Support:
0.15188762071992976
=====

Rule: poultry->
Confidance:
0.9563106796116505
Support:
0.17295873573309922
=====

Rule: ice cream->
Confidance:
0.9414893617021276
Support:
0.15539947322212466
=====

Rule: shampoo->
Confidance:
0.9555555555555556
Support:
0.15100965759438104
=====

Rule: poultry->
Confidance:
0.9333333333333333
Support:
0.15978928884986832
=====

Rule: poultry-> vegetables
Confidance:

0.8398058252427184
Support:
0.15188762071992976
=====

Rule: spaghetti sauce->
Confidance:
0.9661016949152543
Support:
0.15013169446883232
=====

Rule: spaghetti sauce->
Confidance:
0.9617486338797814
Support:
0.15452151009657594
=====

Rule: -> vegetables
Confidance:
0.7746835443037974
Support:
0.26865671641791045
=====

Rule: vegetables->
Confidance:
0.9714285714285714
Support:
0.26865671641791045
=====

Rule: spaghetti sauce->
Confidance:
0.7536945812807881
Support:
0.26865671641791045
=====

Rule: spaghetti sauce->
Confidance:
0.9621621621621623
Support:
0.1562774363476734
=====

Rule: fruits->
Confidance:
0.9662921348314606
Support:
0.15100965759438104
=====

Rule: -> vegetables
Confidance:
0.7571059431524547
Support:
0.257243195785777
=====

Rule: vegetables->
Confidance:
0.957516339869281
Support:
0.257243195785777
=====

Rule: fruits->
Confidance:
0.727047146401985

Support:
0.257243195785777
=====

Rule: bagels->
Confidance:
0.9615384615384616
Support:
0.15364354697102722
=====

Rule: waffles->
Confidance:
0.9617486338797814
Support:
0.15452151009657594
=====

Rule: mixes->
Confidance:
0.9693877551020409
Support:
0.16681299385425813
=====

Rule: dishwashing liquid/detergent->
Confidance:
0.9771428571428572
Support:
0.15013169446883232
=====

Rule: -> vegetables
Confidance:
0.7745098039215687
Support:
0.2774363476733977
=====

Rule: vegetables->
Confidance:
0.9723076923076921
Support:
0.2774363476733977
=====

Rule: dishwashing liquid/detergent->
Confidance:
0.7523809523809524
Support:
0.2774363476733977
=====

Rule: lunch meat->
Confidance:
0.9619565217391304
Support:
0.15539947322212466
=====

Rule: ketchup->
Confidance:
0.9662921348314606
Support:
0.15100965759438104
=====

Rule: -> vegetables
Confidance:
0.7639593908629441
Support:

0.2642669007901668
=====

Rule: vegetables->
Confidance:
0.9585987261146496
Support:
0.2642669007901668
=====

Rule: ketchup->
Confidance:
0.7359413202933984
Support:
0.2642669007901668
=====

Rule: -> vegetables
Confidance:
0.7506631299734748
Support:
0.24846356453028973
=====

Rule: vegetables->
Confidance:
0.9496644295302014
Support:
0.24846356453028973
=====

Rule: tortillas->
Confidance:
0.7219387755102041
Support:
0.24846356453028973
=====

Rule: yogurt->
Confidance:
0.9502762430939227
Support:
0.15100965759438104
=====

Rule: eggs->
Confidance:
0.9508196721311476
Support:
0.1527655838454785
=====

Rule: soda->
Confidance:
0.9340101522842639
Support:
0.16154521510096576
=====

Rule: -> vegetables
Confidance:
0.822222222222223
Support:
0.2923617208077261
=====

Rule: vegetables->
Confidance:
0.9514285714285715
Support:
0.2923617208077261

```
=====
Rule: eggs->
Confidance:
0.7853773584905661
Support:
0.2923617208077261
=====

Rule: eggs->
Confidance:
0.945945945945946
Support:
0.15364354697102722
=====

Rule: -> vegetables
Confidance:
0.812987012987013
Support:
0.27480245829675154
=====

Rule: vegetables->
Confidance:
0.9660493827160495
Support:
0.27480245829675154
=====

Rule: sugar->
Confidance:
0.7864321608040201
Support:
0.27480245829675154
=====

Rule: -> vegetables
Confidance:
0.7859007832898172
Support:
0.2642669007901668
=====

Rule: vegetables->
Confidance:
0.9616613418530351
Support:
0.2642669007901668
=====

Rule: paper towels->
Confidance:
0.7581863979848866
Support:
0.2642669007901668
=====

Rule: yogurt->
Confidance:
0.9508196721311476
Support:
0.1527655838454785
=====

Rule: yogurt->
Confidance:
0.945054945054945
Support:
0.15100965759438104
=====
```

Rule: yogurt->
Confidance:
0.9553072625698324
Support:
0.15013169446883232
=====

Rule: yogurt->
Confidance:
0.9619883040935672
Support:
0.28884986830553117
=====

Rule: yogurt-> vegetables
Confidance:
0.818407960199005
Support:
0.28884986830553117
=====

Rule: yogurt->
Confidance:
0.7870813397129186
Support:
0.28884986830553117
=====

Rule: yogurt->
Confidance:
0.9456521739130436
Support:
0.1527655838454785
=====

Rule: -> vegetables
Confidance:
0.7578947368421053
Support:
0.2528533801580334
=====

Rule: vegetables->
Confidance:
0.9504950495049505
Support:
0.2528533801580334
=====

Rule: pasta->
Confidance:
0.7254408060453401
Support:
0.2528533801580334
=====

Rule: -> vegetables
Confidance:
0.7611940298507462
Support:
0.26865671641791045
=====

Rule: vegetables->
Confidance:
0.9562499999999999
Support:
0.26865671641791045
=====

Rule: bagels->

Confidance:
0.7320574162679425
Support:
0.26865671641791045
=====

Rule: coffee/tea->
Confidance:
0.9395604395604396
Support:
0.15013169446883232
=====

Rule: -> vegetables
Confidance:
0.7432098765432098
Support:
0.2642669007901668
=====

Rule: coffee/tea->
Confidance:
0.9555555555555555
Support:
0.2642669007901668
=====

Rule: coffee/tea->
Confidance:
0.7132701421800948
Support:
0.2642669007901668
=====

Rule: soda->
Confidance:
0.9414893617021276
Support:
0.15539947322212466
=====

Rule: -> vegetables
Confidance:
0.7893462469733656
Support:
0.286215978928885
=====

Rule: vegetables->
Confidance:
0.9560117302052786
Support:
0.286215978928885
=====

Rule: waffles->
Confidance:
0.758139534883721
Support:
0.286215978928885
=====

Rule: lunch meat->
Confidance:
0.9375
Support:
0.15803336259877085
=====

Rule: waffles->
Confidance:

0.9251336898395721
Support:
0.15188762071992976
=====

Rule: -> vegetables
Confidance:
0.7729591836734695
Support:
0.2660228270412643
=====

Rule: vegetables->
Confidance:
0.9558359621451106
Support:
0.2660228270412643
=====

Rule: juice->
Confidance:
0.7390243902439024
Support:
0.2660228270412643
=====

Rule: aluminum foil->
Confidance:
0.9508196721311476
Support:
0.1527655838454785
=====

Rule: -> vegetables
Confidance:
0.7628361858190709
Support:
0.2739244951712028
=====

Rule: vegetables->
Confidance:
0.9454545454545455
Support:
0.2739244951712028
=====

Rule: soda->
Confidance:
0.7289719626168225
Support:
0.2739244951712028
=====

Rule: soda->
Confidance:
0.9347826086956522
Support:
0.15100965759438104
=====

Rule: soda->
Confidance:
0.9267015706806282
Support:
0.15539947322212466
=====

Rule: -> vegetables
Confidance:
0.7603911980440098

Support:
0.2730465320456541
=====

Rule: vegetables->
Confidance:
0.971875
Support:
0.2730465320456541
=====

Rule: mixes->
Confidance:
0.7404761904761905
Support:
0.2730465320456541
=====

Rule: aluminum foil->
Confidance:
0.9516129032258064
Support:
0.15539947322212466
=====

Rule: mixes->
Confidance:
0.9662921348314606
Support:
0.15100965759438104
=====

Rule: -> vegetables
Confidance:
0.8
Support:
0.28094820017559263
=====

Rule: vegetables->
Confidance:
0.9696969696969697
Support:
0.28094820017559263
=====

Rule: laundry detergent->
Confidance:
0.7766990291262136
Support:
0.28094820017559263
=====

Rule: -> vegetables
Confidance:
0.7544757033248083
Support:
0.2589991220368745
=====

Rule: vegetables->
Confidance:
0.9485530546623795
Support:
0.2589991220368745
=====

Rule: all- purpose->
Confidance:
0.7177615571776156
Support:

0.2589991220368745
=====

Rule: -> vegetables
Confidance:
0.7595907928388748
Support:
0.2607550482879719
=====

Rule: vegetables->
Confidance:
0.9705882352941176
Support:
0.2607550482879719
=====

Rule: butter->
Confidance:
0.7388059701492538
Support:
0.2607550482879719
=====

Rule: beef->
Confidance:
0.9394904458598727
Support:
0.2589991220368745
=====

Rule: beef-> vegetables
Confidance:
0.7564102564102565
Support:
0.2589991220368745
=====

Rule: beef->
Confidance:
0.7142857142857143
Support:
0.2589991220368745
=====

Rule: -> vegetables
Confidance:
0.7829268292682926
Support:
0.28182616330114135
=====

Rule: vegetables->
Confidance:
0.9553571428571429
Support:
0.28182616330114135
=====

Rule: lunch meat->
Confidance:
0.7517564402810304
Support:
0.28182616330114135
=====

Rule: -> vegetables
Confidance:
0.7806788511749347
Support:
0.26251097453906935

=====

Rule: vegetables->
Confidance:
0.9645161290322579
Support:
0.26251097453906935

=====

Rule: sandwich bags->
Confidance:
0.755050505050505
Support:
0.26251097453906935

=====

Rule: -> vegetables
Confidance:
0.7894736842105263
Support:
0.276558384547849

=====

Rule: vegetables->
Confidance:
0.9516616314199396
Support:
0.276558384547849

=====

Rule: aluminum foil->
Confidance:
0.7553956834532374
Support:
0.276558384547849

=====

Rule: -> vegetables
Confidance:
0.7807881773399015
Support:
0.2783143107989464

=====

Rule: vegetables->
Confidance:
0.9548192771084337
Support:
0.2783143107989464

=====

Rule: dinner rolls->
Confidance:
0.7476415094339622
Support:
0.2783143107989464

=====

Rule: pork->
Confidance:
0.9452054794520548
Support:
0.24231782265144863

=====

Rule: pork-> vegetables
Confidance:
0.745945945945946
Support:
0.24231782265144863

=====

Rule: pork->
Confidance:
0.7113402061855669
Support:
0.24231782265144863
=====

Rule: vegetables->
Confidance:
0.9504643962848297
Support:
0.26953467954345917
=====

Rule: -> vegetables
Confidance:
0.7469586374695864
Support:
0.26953467954345917
=====

Rule: ice cream->
Confidance:
0.7156177156177156
Support:
0.26953467954345917
=====

Rule: -> vegetables
Confidance:
0.7714285714285715
Support:
0.2607550482879719
=====

Rule: shampoo->
Confidance:
0.9519230769230769
Support:
0.2607550482879719
=====

Rule: shampoo->
Confidance:
0.7369727047146402
Support:
0.2607550482879719
=====

Rule: -> vegetables
Confidance:
0.7913279132791328
Support:
0.25636523266022826
=====

Rule: vegetables->
Confidance:
0.9605263157894736
Support:
0.25636523266022826
=====

Rule: flour->
Confidance:
0.7604166666666666
Support:
0.25636523266022826
=====

Rule: -> vegetables

```

Confidance:
0.7655860349127183
Support:
0.26953467954345917
=====
Rule: vegetables->
Confidance:
0.9446153846153845
Support:
0.26953467954345917
=====
Rule: soap->
Confidance:
0.7274881516587678
Support:
0.26953467954345917
=====
```

Case 2 (minimum support=0.3 and minimum confidence=0.7)

Identify the Itemset and derive association rules.

```
In [ ]: freq_items = apriori(ohe_df, min_support=0.30, use_colnames=True, verbose=1)
freq_items
```

Processing 1497 combinations | Sampling itemset size 3

C:\Users\Acer\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:11
1: DeprecationWarning: DataFrames with non-bool types result in worse computational performance and their support might be discontinued in the future. Please use a DataFrame with bool type
warnings.warn(

	support	itemsets
0	0.380158	(cereals)
1	0.337138	(sandwich loaves)
2	0.365233	(individual meals)
3	0.371378	(cheeses)
4	0.332748	(hand soap)
...
76	0.360843	(, ice cream)
77	0.338016	(shampoo,)
78	0.323968	(, flour)
79	0.352063	(, soap)
80	0.301141	(, vegetables, poultry)

81 rows × 2 columns

```
In [ ]: rules = association_rules(freq_items, metric="confidence", min_threshold=0.7)
rules
```

Out[]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lever
0	(cereals)	()	0.380158	0.974539	0.367867	0.967667	0.992949	-0.002
1	(sandwich loaves)	()	0.337138	0.974539	0.322212	0.955729	0.980699	-0.006
2	(individual meals)	()	0.365233	0.974539	0.351185	0.961538	0.986660	-0.004
3	(cheeses)	()	0.371378	0.974539	0.359965	0.969267	0.994590	-0.001
4	(hand soap)	()	0.332748	0.974539	0.318701	0.957784	0.982807	-0.005
5	(milk)	()	0.357331	0.974539	0.338894	0.948403	0.973181	-0.009
6	(toilet paper)	()	0.360843	0.974539	0.342406	0.948905	0.973696	-0.009
7	(poultry)	()	0.400351	0.974539	0.386304	0.964912	0.990122	-0.003
8	(poultry)	(vegetables)	0.400351	0.715540	0.314311	0.785088	1.097196	0.027
9	(spaghetti sauce)	()	0.356453	0.974539	0.346795	0.972906	0.998325	-0.000
10	(fruits)	()	0.353819	0.974539	0.339772	0.960298	0.985387	-0.005
11	(dishwashing liquid/detergent)	()	0.368745	0.974539	0.358209	0.971429	0.996808	-0.001
12	(ketchup)	()	0.359087	0.974539	0.345917	0.963325	0.988493	-0.004
13	(tortillas)	()	0.344162	0.974539	0.330992	0.961735	0.986861	-0.004
14	(eggs)	()	0.372256	0.974539	0.355575	0.955189	0.980144	-0.007
15	(eggs)	(vegetables)	0.372256	0.715540	0.307287	0.825472	1.153635	0.040
16	(sugar)	()	0.349429	0.974539	0.338016	0.967337	0.992609	-0.002
17	(paper towels)	()	0.348551	0.974539	0.336260	0.964736	0.989940	-0.003
18	(yogurt)	()	0.366989	0.974539	0.352941	0.961722	0.986849	-0.004
19	(yogurt)	(vegetables)	0.366989	0.715540	0.300263	0.818182	1.143447	0.037
20	(pasta)	()	0.348551	0.974539	0.333626	0.957179	0.982186	-0.006
21	(bagels)	()	0.366989	0.974539	0.352941	0.961722	0.986849	-0.004
22	(coffee/tea)	()	0.370500	0.974539	0.355575	0.959716	0.984789	-0.005
23	(waffles)	()	0.377524	0.974539	0.362599	0.960465	0.985558	-0.005
24	(juice)	()	0.359965	0.974539	0.344162	0.956098	0.981077	-0.006
25	(soda)	()	0.375768	0.974539	0.359087	0.955607	0.980574	-0.007
26	(mixes)	()	0.368745	0.974539	0.359087	0.973810	0.999251	-0.000
27	(laundry detergent)	()	0.361721	0.974539	0.351185	0.970874	0.996239	-0.001
28	(all- purpose)	()	0.360843	0.974539	0.343284	0.951338	0.976193	-0.008
29	(butter)	()	0.352941	0.974539	0.343284	0.972637	0.998048	-0.000
30	(beef)	()	0.362599	0.974539	0.342406	0.944310	0.968981	-0.010
31	()	(vegetables)	0.974539	0.715540	0.693591	0.711712	0.994650	-0.003
32	(vegetables)	()	0.715540	0.974539	0.693591	0.969325	0.994650	-0.003

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lever
33	(lunch meat)	()	0.374890	0.974539	0.359965	0.960187	0.985273	-0.005
34	(sandwich bags)	()	0.347673	0.974539	0.336260	0.967172	0.992440	-0.002
35	(aluminum foil)	()	0.366111	0.974539	0.350307	0.956835	0.981833	-0.006
36	(dinner rolls)	()	0.372256	0.974539	0.356453	0.957547	0.982564	-0.006
37	(pork)	()	0.340650	0.974539	0.324846	0.953608	0.978522	-0.007
38	(ice cream)	()	0.376646	0.974539	0.360843	0.958042	0.983072	-0.006
39	(shampoo)	()	0.353819	0.974539	0.338016	0.955335	0.980294	-0.006
40	(flour)	()	0.337138	0.974539	0.323968	0.960938	0.986043	-0.004
41	(soap)	()	0.370500	0.974539	0.352063	0.950237	0.975063	-0.009
42	(, poultry)	(vegetables)	0.386304	0.715540	0.301141	0.779545	1.089451	0.024
43	(vegetables, poultry)	()	0.314311	0.974539	0.301141	0.958101	0.983132	-0.005
44	(poultry)	(, vegetables)	0.400351	0.693591	0.301141	0.752193	1.084491	0.023

```
In [ ]: rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")
rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("unicode")
```

Case 2- Itemset and Support

```
In [ ]: for index, row in freq_items.iterrows():
    print(row['itemsets'])
    print("Support: ")
    print(row['support'])
    print("=====")
```

```
frozenset({' cereals'})
Support:
0.3801580333625988
=====
frozenset({' sandwich loaves'})
Support:
0.33713784021071114
=====
frozenset({' individual meals'})
Support:
0.3652326602282704
=====
frozenset({' cheeses'})
Support:
0.3713784021071115
=====
frozenset({' hand soap'})
Support:
0.33274802458296754
=====
frozenset({' milk'})
Support:
0.35733099209833186
=====
frozenset({' toilet paper'})
Support:
0.3608428446005268
=====
frozenset({' poultry'})
Support:
0.4003511852502195
=====
frozenset({' spaghetti sauce'})
Support:
0.35645302897278314
=====
frozenset({' fruits'})
Support:
0.353819139596137
=====
frozenset({' dishwashing liquid/detergent'})
Support:
0.3687445127304653
=====
frozenset({' ketchup'})
Support:
0.35908691834942935
=====
frozenset({' tortillas'})
Support:
0.34416154521510095
=====
frozenset({' eggs'})
Support:
0.3722563652326602
=====
frozenset({' sugar'})
Support:
0.3494293239683933
=====
frozenset({' paper towels'})
```

```
Support:  
0.3485513608428446  
=====  
frozenset({' yogurt'})  
Support:  
0.3669885864793679  
=====  
frozenset({' pasta'})  
Support:  
0.3485513608428446  
=====  
frozenset({' bagels'})  
Support:  
0.3669885864793679  
=====  
frozenset({' coffee/tea'})  
Support:  
0.37050043898156276  
=====  
frozenset({' waffles'})  
Support:  
0.3775241439859526  
=====  
frozenset({' juice'})  
Support:  
0.35996488147497807  
=====  
frozenset({' '})  
Support:  
0.974539069359087  
=====  
frozenset({' soda'})  
Support:  
0.37576821773485514  
=====  
frozenset({' mixes'})  
Support:  
0.3687445127304653  
=====  
frozenset({' laundry detergent'})  
Support:  
0.3617208077260755  
=====  
frozenset({' all- purpose'})  
Support:  
0.3608428446005268  
=====  
frozenset({' butter'})  
Support:  
0.35294117647058826  
=====  
frozenset({' beef'})  
Support:  
0.36259877085162423  
=====  
frozenset({' vegetables'})  
Support:  
0.7155399473222125  
=====  
frozenset({' lunch meat'})  
Support:
```

0.3748902546093064
=====

frozenset({' sandwich bags'})
Support:
0.3476733977172959
=====

frozenset({' aluminum foil'})
Support:
0.36611062335381916
=====

frozenset({' dinner rolls'})
Support:
0.3722563652326602
=====

frozenset({' pork'})
Support:
0.34064969271290607
=====

frozenset({' ice cream'})
Support:
0.37664618086040386
=====

frozenset({' shampoo'})
Support:
0.353819139596137
=====

frozenset({' flour'})
Support:
0.33713784021071114
=====

frozenset({' soap'})
Support:
0.37050043898156276
=====

frozenset({' ', ' cereals'})
Support:
0.3678665496049166
=====

frozenset({' ', ' sandwich loaves'})
Support:
0.3222124670763828
=====

frozenset({' ', ' individual meals'})
Support:
0.35118525021949076
=====

frozenset({' ', ' cheeses'})
Support:
0.35996488147497807
=====

frozenset({' ', ' hand soap'})
Support:
0.31870061457418786
=====

frozenset({' ', ' milk'})
Support:
0.3388937664618086
=====

frozenset({' ', ' toilet paper'})
Support:
0.3424056189640035

```
=====
frozenset({' ', ' poultry'})
Support:
0.3863037752414399
=====
frozenset({' vegetables', ' poultry'})
Support:
0.31431079894644426
=====
frozenset({' ', ' spaghetti sauce'})
Support:
0.34679543459174716
=====
frozenset({' ', ' fruits'})
Support:
0.33977172958735735
=====
frozenset({' ', ' dishwashing liquid/detergent'})
Support:
0.3582089552238806
=====
frozenset({' ', ' ketchup'})
Support:
0.34591747146619845
=====
frozenset({' ', ' tortillas'})
Support:
0.33099209833187004
=====
frozenset({' ', ' eggs'})
Support:
0.3555750658472344
=====
frozenset({' vegetables', ' eggs'})
Support:
0.30728709394205445
=====
frozenset({' ', ' sugar'})
Support:
0.33801580333625986
=====
frozenset({' ', ' paper towels'})
Support:
0.3362598770851624
=====
frozenset({' yogurt', ' '})
Support:
0.35294117647058826
=====
frozenset({' yogurt', ' vegetables'})
Support:
0.30026338893766463
=====
frozenset({' ', ' pasta'})
Support:
0.33362598770851626
=====
frozenset({' ', ' bagels'})
Support:
0.35294117647058826
=====
```

```
frozenset({' ', ' coffee/tea'})  
Support:  
0.3555750658472344  
=====  
frozenset({' ', ' waffles'})  
Support:  
0.36259877085162423  
=====  
frozenset({' ', ' juice'})  
Support:  
0.34416154521510095  
=====  
frozenset({' ', ' soda'})  
Support:  
0.35908691834942935  
=====  
frozenset({' ', ' mixes'})  
Support:  
0.35908691834942935  
=====  
frozenset({' ', ' laundry detergent'})  
Support:  
0.35118525021949076  
=====  
frozenset({' ', ' all- purpose'})  
Support:  
0.34328358208955223  
=====  
frozenset({' ', ' butter'})  
Support:  
0.34328358208955223  
=====  
frozenset({' beef', ' '})  
Support:  
0.3424056189640035  
=====  
frozenset({' ', ' vegetables'})  
Support:  
0.6935908691834943  
=====  
frozenset({' ', ' lunch meat'})  
Support:  
0.35996488147497807  
=====  
frozenset({' ', ' sandwich bags'})  
Support:  
0.3362598770851624  
=====  
frozenset({' ', ' aluminum foil'})  
Support:  
0.35030728709394204  
=====  
frozenset({' ', ' dinner rolls'})  
Support:  
0.35645302897278314  
=====  
frozenset({' pork', ' '})  
Support:  
0.32484635645302895  
=====  
frozenset({' ', ' ice cream'})
```

```
Support:  
0.3608428446005268  
=====  
frozenset({' shampoo', ' '})  
Support:  
0.33801580333625986  
=====  
frozenset({' ', ' flour'})  
Support:  
0.32396839332748023  
=====  
frozenset({' ', ' soap'})  
Support:  
0.3520632133450395  
=====  
frozenset({' ', ' vegetables', ' poultry'})  
Support:  
0.30114135206321335  
=====
```

Case 2- Association Rules

```
In [ ]: for index, row in rules.iterrows():  
    print("Rule:"+ row['antecedents']+ "->"+ row['consequents'])  
    print("Confidance: ")  
    print( row['confidence'])  
    print("Support:")  
    print( row['support'])  
    print("====")
```

Rule: cereals->
Confidance:
0.9676674364896074
Support:
0.3678665496049166
=====

Rule: sandwich loaves->
Confidance:
0.9557291666666667
Support:
0.3222124670763828
=====

Rule: individual meals->
Confidance:
0.9615384615384616
Support:
0.35118525021949076
=====

Rule: cheeses->
Confidance:
0.9692671394799055
Support:
0.35996488147497807
=====

Rule: hand soap->
Confidance:
0.9577836411609497
Support:
0.31870061457418786
=====

Rule: milk->
Confidance:
0.9484029484029484
Support:
0.3388937664618086
=====

Rule: toilet paper->
Confidance:
0.948905109489051
Support:
0.3424056189640035
=====

Rule: poultry->
Confidance:
0.9649122807017544
Support:
0.3863037752414399
=====

Rule: poultry-> vegetables
Confidance:
0.7850877192982456
Support:
0.31431079894644426
=====

Rule: spaghetti sauce->
Confidance:
0.9729064039408868
Support:
0.34679543459174716
=====

Rule: fruits->

Confidance:
0.9602977667493797
Support:
0.33977172958735735
=====

Rule: dishwashing liquid/detergent->
Confidance:
0.9714285714285714
Support:
0.3582089552238806
=====

Rule: ketchup->
Confidance:
0.9633251833740831
Support:
0.34591747146619845
=====

Rule: tortillas->
Confidance:
0.9617346938775511
Support:
0.33099209833187004
=====

Rule: eggs->
Confidance:
0.9551886792452831
Support:
0.3555750658472344
=====

Rule: eggs-> vegetables
Confidance:
0.8254716981132076
Support:
0.30728709394205445
=====

Rule: sugar->
Confidance:
0.9673366834170853
Support:
0.33801580333625986
=====

Rule: paper towels->
Confidance:
0.964735516372796
Support:
0.3362598770851624
=====

Rule: yogurt->
Confidance:
0.9617224880382775
Support:
0.35294117647058826
=====

Rule: yogurt-> vegetables
Confidance:
0.8181818181818182
Support:
0.30026338893766463
=====

Rule: pasta->
Confidance:

0.9571788413098237
Support:
0.33362598770851626
=====

Rule: bagels->
Confidance:
0.9617224880382775
Support:
0.35294117647058826
=====

Rule: coffee/tea->
Confidance:
0.9597156398104266
Support:
0.3555750658472344
=====

Rule: waffles->
Confidance:
0.9604651162790698
Support:
0.36259877085162423
=====

Rule: juice->
Confidance:
0.9560975609756096
Support:
0.34416154521510095
=====

Rule: soda->
Confidance:
0.955607476635514
Support:
0.35908691834942935
=====

Rule: mixes->
Confidance:
0.9738095238095239
Support:
0.35908691834942935
=====

Rule: laundry detergent->
Confidance:
0.9708737864077669
Support:
0.35118525021949076
=====

Rule: all- purpose->
Confidance:
0.9513381995133819
Support:
0.34328358208955223
=====

Rule: butter->
Confidance:
0.972636815920398
Support:
0.34328358208955223
=====

Rule: beef->
Confidance:
0.9443099273607748

Support:
0.3424056189640035
=====

Rule: -> vegetables
Confidance:
0.7117117117117118
Support:
0.6935908691834943
=====

Rule: vegetables->
Confidance:
0.9693251533742332
Support:
0.6935908691834943
=====

Rule: lunch meat->
Confidance:
0.9601873536299766
Support:
0.35996488147497807
=====

Rule: sandwich bags->
Confidance:
0.9671717171717171
Support:
0.3362598770851624
=====

Rule: aluminum foil->
Confidance:
0.9568345323741007
Support:
0.35030728709394204
=====

Rule: dinner rolls->
Confidance:
0.9575471698113208
Support:
0.35645302897278314
=====

Rule: pork->
Confidance:
0.9536082474226804
Support:
0.32484635645302895
=====

Rule: ice cream->
Confidance:
0.9580419580419581
Support:
0.3608428446005268
=====

Rule: shampoo->
Confidance:
0.955334987593052
Support:
0.33801580333625986
=====

Rule: flour->
Confidance:
0.9609375
Support:

```

0.32396839332748023
=====
Rule: soap->
Confidance:
0.9502369668246445
Support:
0.3520632133450395
=====
Rule: -> vegetables
Confidance:
0.7795454545454545
Support:
0.30114135206321335
=====
Rule: vegetables->
Confidance:
0.9581005586592178
Support:
0.30114135206321335
=====
Rule: poultry->
Confidance:
0.7521929824561403
Support:
0.30114135206321335
=====
```

Case 3 (minimum support=0.4 and minimum confidence=0.85)

Identify the Itemset and derive association rules.

```
In [ ]: freq_items = apriori(ohe_df, min_support=0.4, use_colnames=True, verbose=1)
freq_items
```

Processing 6 combinations | Sampling itemset size 2

C:\Users\Acer\anaconda3\lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:11
1: DeprecationWarning: DataFrames with non-bool types result in worse computational performance and their support might be discontinued in the future. Please use a DataFrame with bool type
warnings.warn(

```
Out[ ]:   support    itemsets
          0    0.400351  ( poultry)
          1    0.974539    ( )
          2    0.715540  ( vegetables)
          3    0.693591  ( , vegetables)
```

```
In [ ]: rules = association_rules(freq_items, metric="confidence", min_threshold=0.85)
rules
```

```
Out[ ]:   antecedents  consequents  antecedent support  consequent support  support  confidence      lift      leverage
          0  ( vegetables)        ( )       0.71554     0.974539    0.693591    0.969325    0.99465  -0.003731
```



```
In [ ]: rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("uniform")
rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("uniform")
```

Case 3- Itemset and Support

```
In [ ]: for index, row in freq_items.iterrows():
    print(row['itemsets'])
    print("Support: ")
    print(row['support'])
    print("=====")

frozenset({' poultry'})
Support:
0.4003511852502195
=====
frozenset({' '})
Support:
0.974539069359087
=====
frozenset({' vegetables'})
Support:
0.7155399473222125
=====
frozenset({' ', ' vegetables'})
Support:
0.6935908691834943
=====
```

CASE 3- Association Rules

```
In [ ]: for index, row in rules.iterrows():
    print("Rule:"+ row['antecedents']+ "->"+ row['consequents'])
    print("Confidance: ")
    print( row['confidence'])
    print("Support:")
    print( row['support'])
    print("=====")
```

```
Rule: vegetables->
Confidance:
0.9693251533742332
Support:
0.6935908691834943
=====
```

There is no set of items with a confidence score of or more than 0.85