WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I
Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

Theory:

Has it ever happened that you're out to buy something, and you end up buying a lot more than you planned? It's a Phenomenon known as **Impulsive Buying** and Big Retailers take advantage of Machine Learning and **Apriori Algorithm** and make sure that we tend to buy more.
Apriori algorithm uses frequent itemsets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset. Frequent Itemset is an itemset whose support value is greater than a threshold value(support).

Apriori Algorithm

Apriori algorithm uses frequent itemsets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset. Frequent Itemset is an itemset whose support value is greater than a threshold value(support).

Let's say we have the following data of a store.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Iteration 1: Let's assume the support value is 2 and create the item sets of the size of 1 and calculate their support values.

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

As you can see here, item 4 has a support value of 1 which is less than the min support value. So we are going to **discard {4}** in the upcoming iterations. We have the final Table F1.

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Iteration 2: Next we will create itemsets of size 2 and calculate their support values. All the combinations of items set in F1 are used in this iteration.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Only Items present in F1

C2

Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

Itemsets having Support less than 2 are eliminated again. In this case **{1,2}.** Now, Let's understand what is pruning and how it makes Apriori one of the best algorithm for finding frequent itemsets.

Pruning: We are going to divide the itemsets in C3 into subsets and eliminate the subsets that are having a support value less than 2.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

C3

Itemset	In F2?
{1,2,3}, {1,2}, {1,3}, {2,3}	NO
{1,2,5}, {1,2}, {1,5}, {2,5}	NO
{1,3,5},{1,5}, {1,3}, {3,5}	YES
{2,3,5}, {2,3}, {2,5}, {3,5}	YES

Iteration 3: We will discard **{1,2,3}** and **{1,2,5}** as they both contain **{1,2}.** This is the main highlight of the Apriori Algorithm.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

Iteration 4: Using sets of F3 we will create C4.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

C3

Itemset	Support
{1,2,3,5}	1

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

Since the Support of this itemset is less than 2, we will stop here and the final itemset we will have is F3.
Note: Till now we haven't calculated the confidence values yet.

With F3 we get the following itemsets:

For I = {1,3,5}, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}
For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}

Applying Rules: We will create rules and apply them on itemset F3. Now let's assume a minimum confidence value is **60%.**

For every subsets S of I, you output the rule

- S –> (I-S) (means S recommends I-S)
- if **support(I) / support(S) >= min_conf value**

{1,3,5}

Rule 1: {1,3} –> ({1,3,5} – {1,3}) means 1 & 3 –> 5

Confidence = support(1,3,5)/support(1,3) = 2/3 = **66.66% > 60%**

Hence Rule 1 is **Selected**

Rule 2: {1,5} –> ({1,3,5} – {1,5}) means 1 & 5 –> 3

Confidence = support(1,3,5)/support(1,5) = 2/2 = **100% > 60%**

Rule 2 is **Selected**

Rule 3: {3,5} –> ({1,3,5} – {3,5}) means 3 & 5 –> 1

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

Confidence = support(1,3,5)/support(3,5) = 2/3 = **66.66% > 60%**

Rule 3 is **Selected**

Rule 4: {1} –> ({1,3,5} – {1}) means 1 –> 3 & 5

Confidence = support(1,3,5)/support(1) = 2/3 = **66.66% > 60%**

Rule 4 is **Selected**

Rule 5: {3} –> ({1,3,5} – {3}) means 3 –> 1 & 5

Confidence = support(1,3,5)/support(3) = 2/4 = **50% <60%**

Rule 5 is **Rejected**

Rule 6: {5} –> ({1,3,5} – {5}) means 5 –> 1 & 3

Confidence = support(1,3,5)/support(5) = 2/4 = 50% < 60%

Rule 6 is **Rejected**

This is how you create rules in Apriori Algorithm and the same steps can be implemented for the itemset **{2,3,5}.** Try it for yourself and see which rules are accepted and which are rejected.

Program:

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
def encode_units(x):
    if x <= 0:
```

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

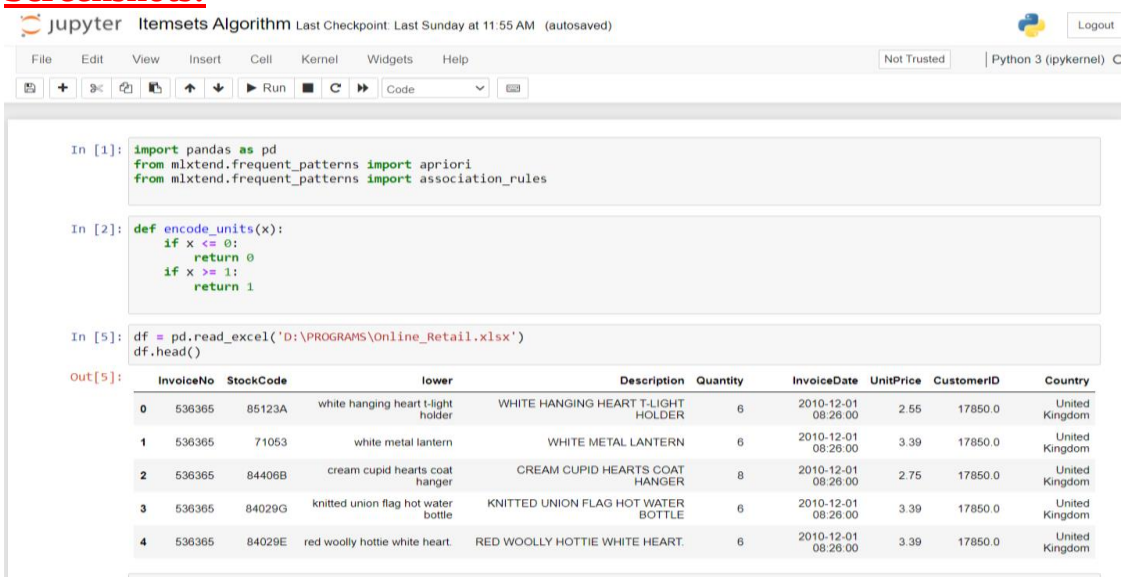
Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.

```
        return 0
    if x >= 1:
        return 1
df = pd.read_excel('D:\PROGRAMS\Online_Retail.xlsx')
df.head()
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
df
basket = (df[df['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
basket
basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
basket_sets
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()
rules[ (rules['lift'] >= 6) & (rules['confidence'] >= 0.8) ]
```

Screenshots:



jupyter Itemsets Algorithm Last Checkpoint: Last Sunday at 11:55 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

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

In [2]:
```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1
```

In [5]:
```
df = pd.read_excel('D:\PROGRAMS\Online_Retail.xlsx')
df.head()
```

Out[5]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

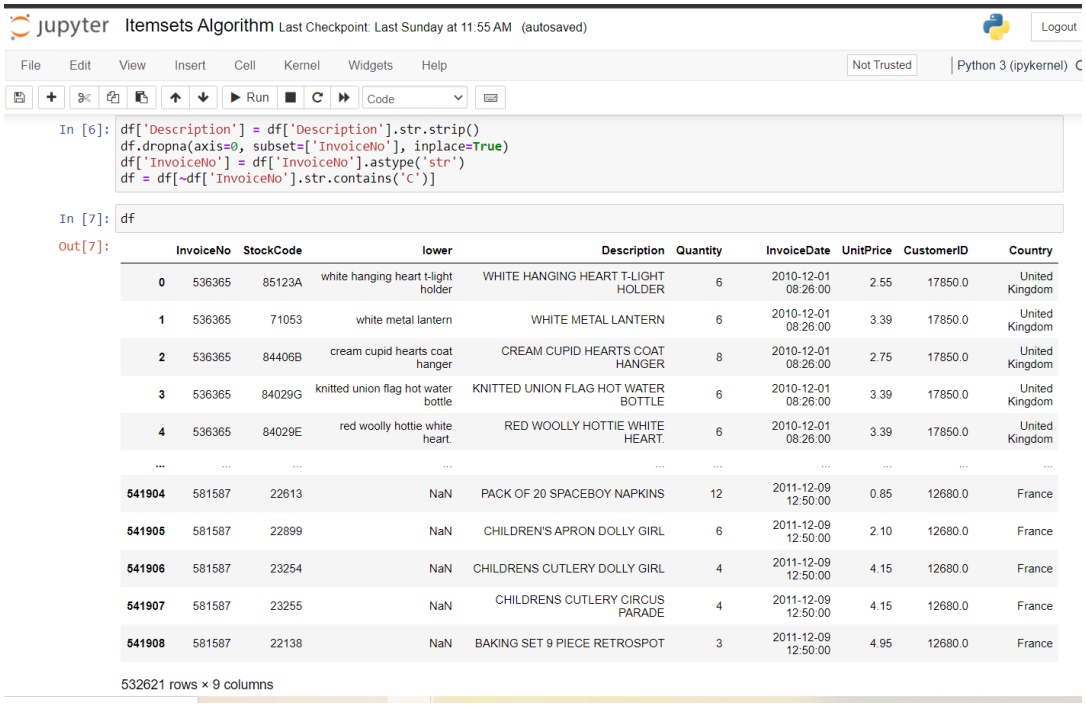
WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.



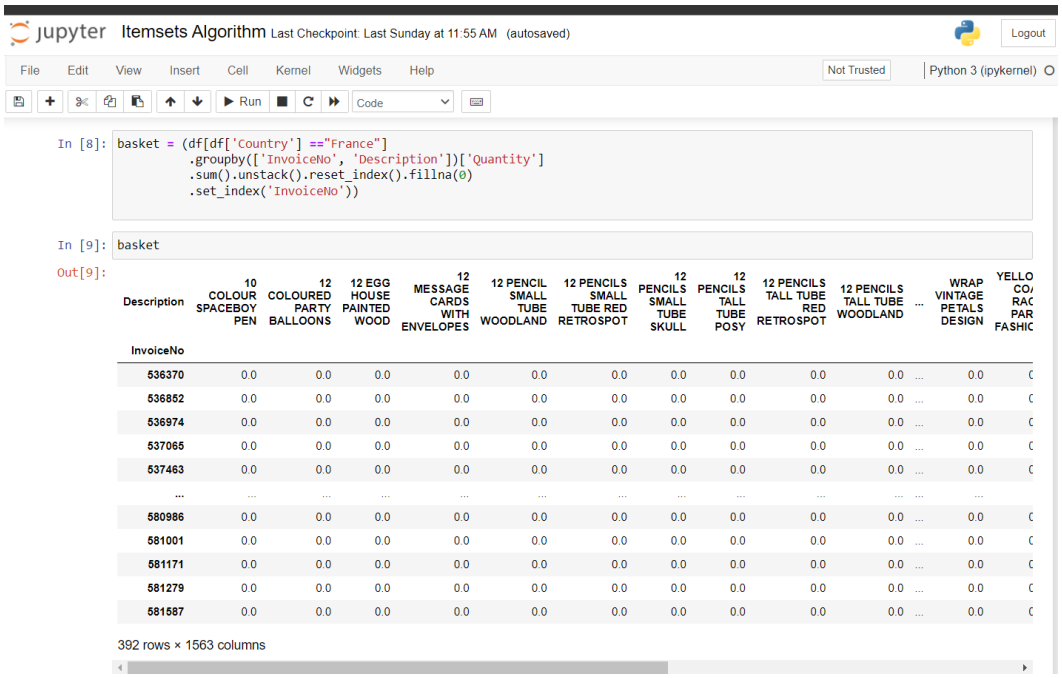
In [6]:
```
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
```

In [7]:
```
df
```

Out[7]:

	InvoiceNo	StockCode	lower	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	white hanging heart t-light holder	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	white metal lantern	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	cream cupid hearts coat hanger	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	knitted union flag hot water bottle	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	red woolly hottie white heart.	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...	...
541904	581587	22613	NaN	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	NaN	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
541906	581587	23254	NaN	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
541907	581587	23255	NaN	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
541908	581587	22138	NaN	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

532621 rows × 9 columns



In [8]:
```
basket = (df[df['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

In [9]:
```
basket
```

Out[9]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	...	WRAP VINTAGE PETALS DESIGN	YELLO CO RAC PAR FASHIC
InvoiceNo													
536370	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
536852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
536974	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
537065	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
537463	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	
580986	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
581001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
581171	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
581279	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0
581587	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0

392 rows × 1563 columns

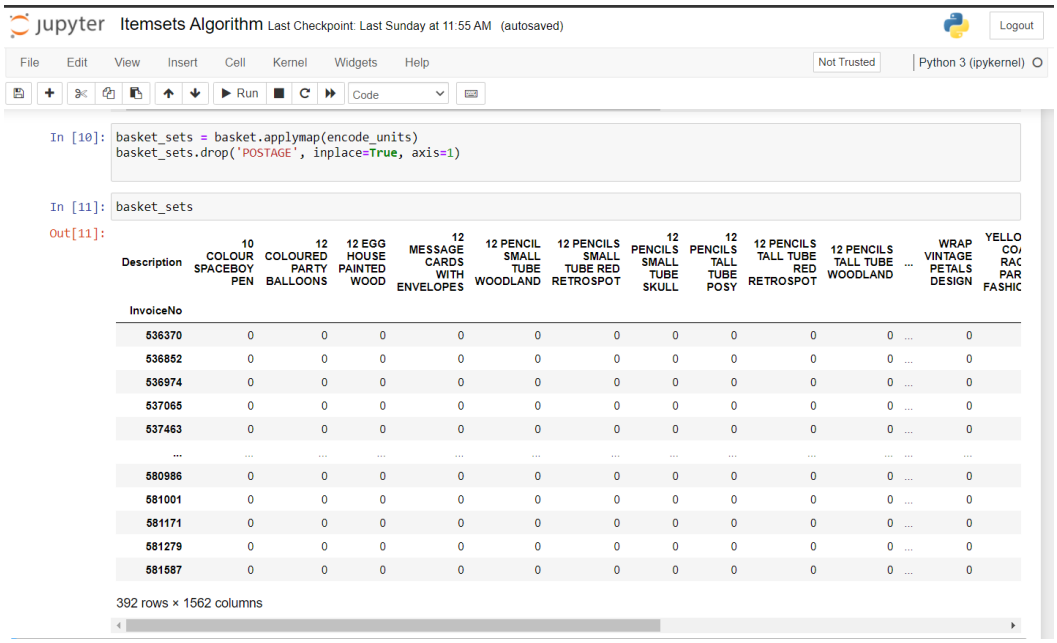
WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.



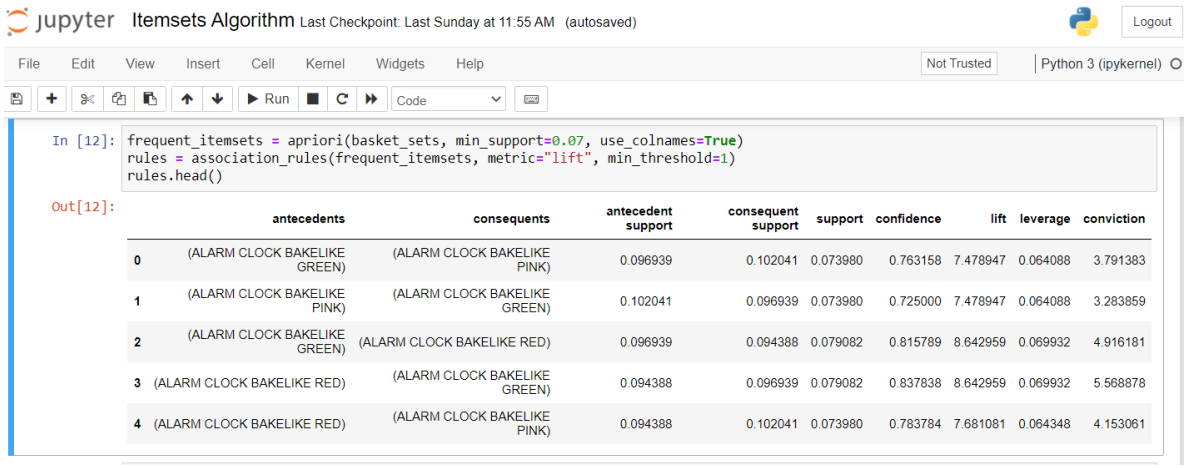

```
In [10]: basket_sets = basket.applymap(encode_units)
         basket_sets.drop('POSTAGE', inplace=True, axis=1)

In [11]: basket_sets
```

Out[11]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	...	WRAP VINTAGE PETALS DESIGN	YELLO CO/ RAC PAR FASHIC
InvoiceNo													
536370	0	0	0	0	0	0	0	0	0	0	...	0	
536852	0	0	0	0	0	0	0	0	0	0	...	0	
536974	0	0	0	0	0	0	0	0	0	0	...	0	
537065	0	0	0	0	0	0	0	0	0	0	...	0	
537463	0	0	0	0	0	0	0	0	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
580986	0	0	0	0	0	0	0	0	0	0	...	0	
581001	0	0	0	0	0	0	0	0	0	0	...	0	
581171	0	0	0	0	0	0	0	0	0	0	...	0	
581279	0	0	0	0	0	0	0	0	0	0	...	0	
581587	0	0	0	0	0	0	0	0	0	0	...	0	

392 rows × 1562 columns




```
In [12]: frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
         rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
         rules.head()
```

Out[12]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE PINK)	0.096939	0.102041	0.073980	0.763158	7.478947	0.064088	3.791383
1	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.102041	0.096939	0.073980	0.725000	7.478947	0.064088	3.283859
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878
4	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE PINK)	0.094388	0.102041	0.073980	0.783784	7.681081	0.064348	4.153061

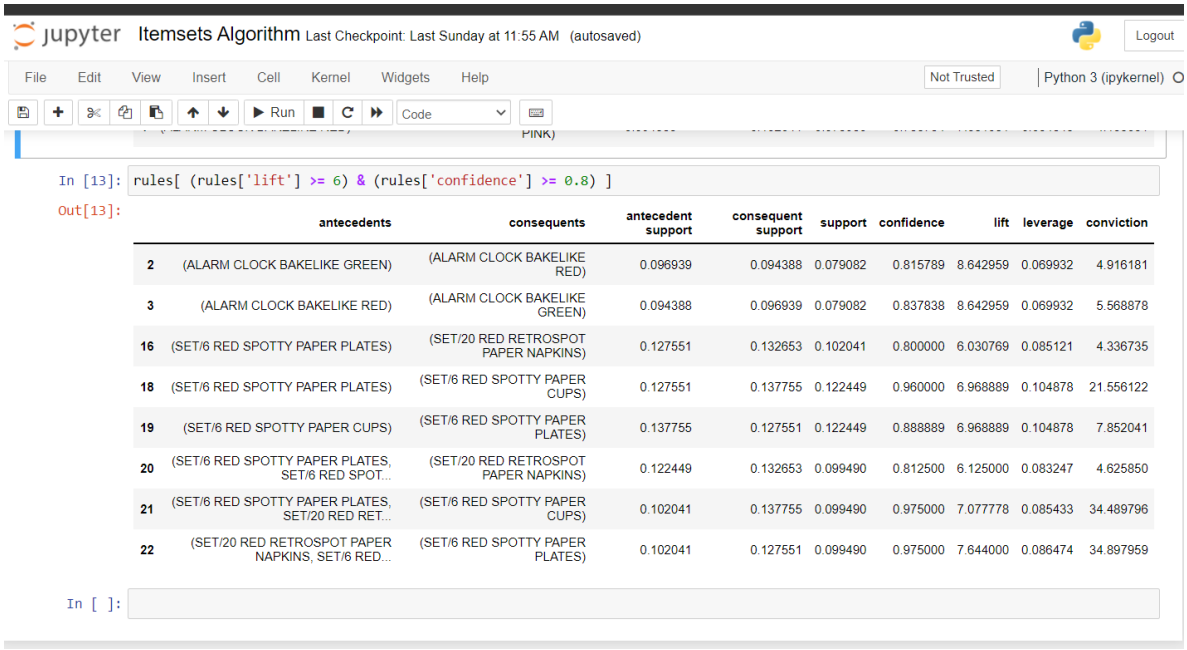
WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 10

Title: Implement algorithm for finding Frequent Itemsets for a given minimum support.



In [13]:
```
rules[ (rules['lift'] >= 6) & (rules['confidence'] >= 0.8) ]
```

Out[13]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
2	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878
16	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.127551	0.132653	0.102041	0.800000	6.030769	0.085121	4.336735
18	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.127551	0.137755	0.122449	0.960000	6.968889	0.104878	21.556122
19	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.137755	0.127551	0.122449	0.888889	6.968889	0.104878	7.852041
20	(SET/6 RED SPOTTY PAPER PLATES, SET/6 RED SPOT...	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.122449	0.132653	0.099490	0.812500	6.125000	0.083247	4.625850
21	(SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.099490	0.975000	7.077778	0.085433	34.489796
22	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.099490	0.975000	7.644000	0.086474	34.897959

In [ ]: