Problem 2.
1. Compute the support for itemsets {e}, {b, d}, and {b, d, e} by treating each transaction ID as a market basket.

For {e} = sigma(e)/total = 8/10 = 0.8
For {b,d} = sigma(b,d)/N = 2/10 = 0.2
For (b,d,e) = 2/10 = 0.2

2. Use the results in part (a) to compute the confidence for the association rules {b, d}→{e} and {e}→{b, d}. Is confidence a symmetric measure?

For {b, d}→{e} the confidence formula is **sigma({b, d}U{e})/ sigma({b, d})**
sigma({b, d}U{e}) = 2/10 = 0.2
sigma({b, d}) = = 2/10 = 0.2
so the confidence will be : 0.2/0.2 = **100%**

For {e}→{b, d} confidence will be : 0.2/0.8 = **25%**

3. Repeat part (a) by treating each customer ID as a market basket. Each item should be treated as a binary variable.
For {e} = sigma(e)/total = 4/5 = 0.8 (e appears in 4 customers basket and total are 5)
For {b,d} = sigma(b,d)/total = 5/5 = 1 (b,d appears in 5 customers basket and total are 5)
For (b,d,e) = 4/5 = 0.8 (same as case 1st)

4. Use the results in part (c) to compute the confidence for the association rules {b, d}→{e} and {e}→{b, d}.

For {b, d}→{e} confidence will be : 0.8/1 = **80%**

For {e}→{b, d} confidence will be : 0.8/0.8 = **100%**

5. I do not think there is any relationship between s1 and s2 or c1 and c2 the reason behind is these totally different scenarios and items sets are also different.

Problem 6.
1. What is the maximum number of association rules that can be extracted from this data (including rules that have zero support)?

Equation to get max no of association rule is $R = 3^d - 2^{(d+1)} + 1$
Where d is number of items is market backet. In our case we have 6 different items so d = 6.

$$R = 3^6 - 2^7 + 1 = \mathbf{602}$$

2. What is the maximum size of frequent itemsets that can be extracted (assuming minsup>0)?

Since we have minsup>0, that means we just need to look for max number of items in basket. For ID 6 & 9 we have max items **4** this will be the maximum size of frequent itemset.

3. Write an expression for the maximum number of size-3 itemsets that can be derived from this data set.

This is something like combination question, so we have find all the 3 combination from 6.

$$6C3 = 6!/3!*3! = \mathbf{20}$$

4. Find an itemset (of size 2 or larger) that has the largest support.

If we closely look into the data sets we can observe the **bread and butter** comes together 5 times in 10 observations no any other 2 or larger size itemset are present more then 5 times.
support for **bread and butter = 5/10 = 50%**

5. Find a pair of items, a and b, such that the rules {a}→{b} and {b}→{a} have the same confidence.

In this case I would like to see the same support first then look for their confidence.

If we can see the bread and butter both have same support is **5.** Now lets calculate their confidence for { bread }$\rightarrow${ butter} = 5/5 = 1
{ butter }$\rightarrow${ bread } = 5/5 = 1

Similarly, we have one more combination beer and cookie.
So, we have **{butter, bread} , {beer, cookie} , {milk, butter} and {milk, bread}**

Problem 8.

1. List all candidate 4-itemsets obtained by a candidate generation procedure using the $F_{k-1} \times F_1$ merging strategy.

   Item and support:

1	5
2	5
3	6
4	4
5	4

   Candidate 4-itemsets using the $F_{k-1} \times F_1$ merging strategy:
   {1,2,3}: {1,2,3,4}, {1,2,3,5}
   {1,2,4}: {1,2,4,5}
   {1,2,5}: no results, since 5 is the last item.
   {1,3,4}: {1,3,4,5}
   {1,3,5} : no results, since 5 is the last item.
   {2,3,4} : {2,3,4,5}
   {2,3,5} : no results, since 5 is the last item.
   {3,4,5}: no results, since 5 is the last item.

2. List all candidate 4-itemsets obtained by the candidate generation procedure in Apriori.

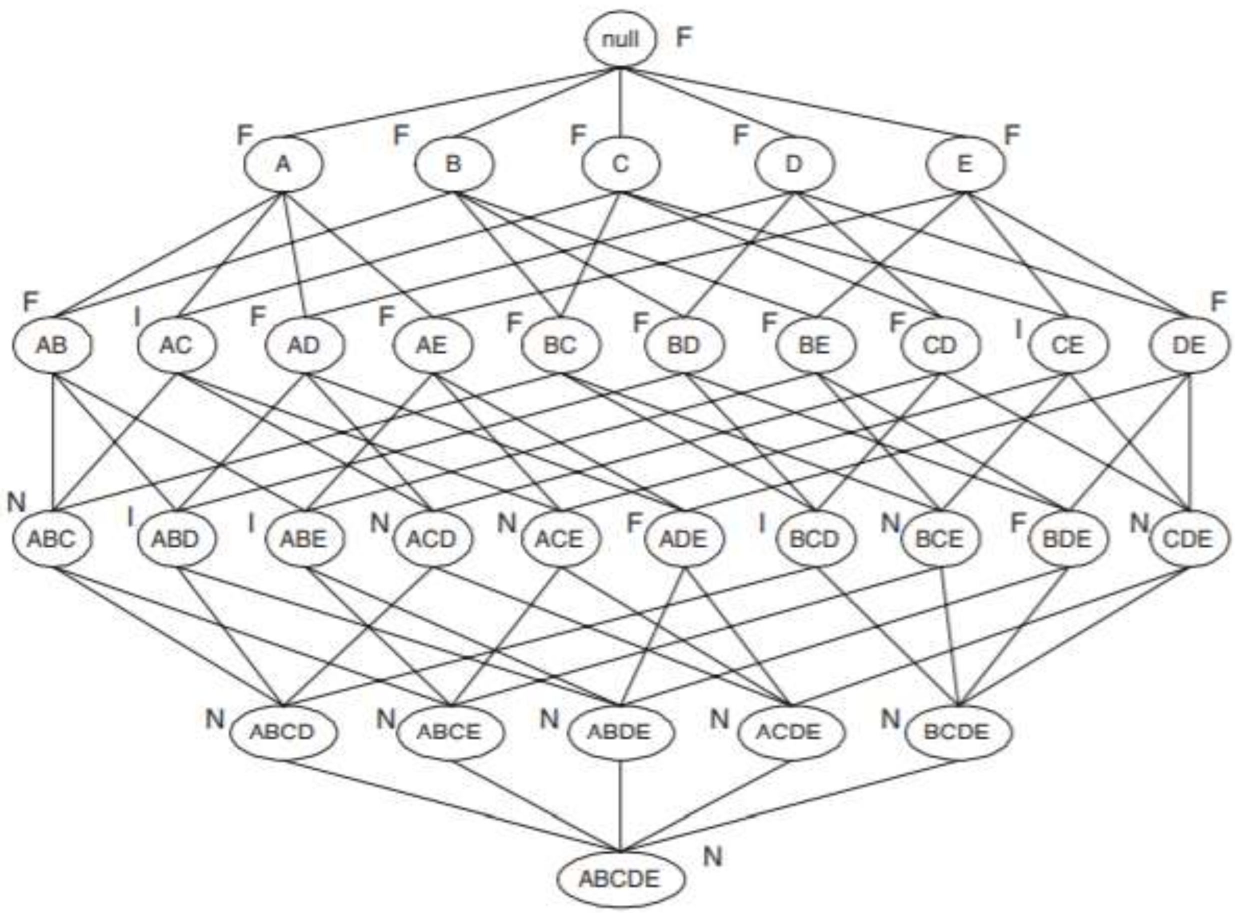
   From the previous question we did 4-itemsets from 3-itemsets. So our result will be same as above:
   {1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 4, 5}, {1, 3, 4, 5}, {2, 3, 4, 5}

3. List all candidate 4-itemsets that survive the candidate pruning step of the Apriori algorithm.

   Sunsets of {1, 2, 3, 4} are {1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4} set of frequent 3-itemsets.

   Sunsets of {1, 2, 3, 5} are {1, 2, 3}, {1, 2, 5}, {1, 3, 5}, {2, 3, 5} set of frequent 3-itemsets.

Problem 9.



2. What is the percentage of frequent itemsets (with respect to all itemsets in the lattice)?

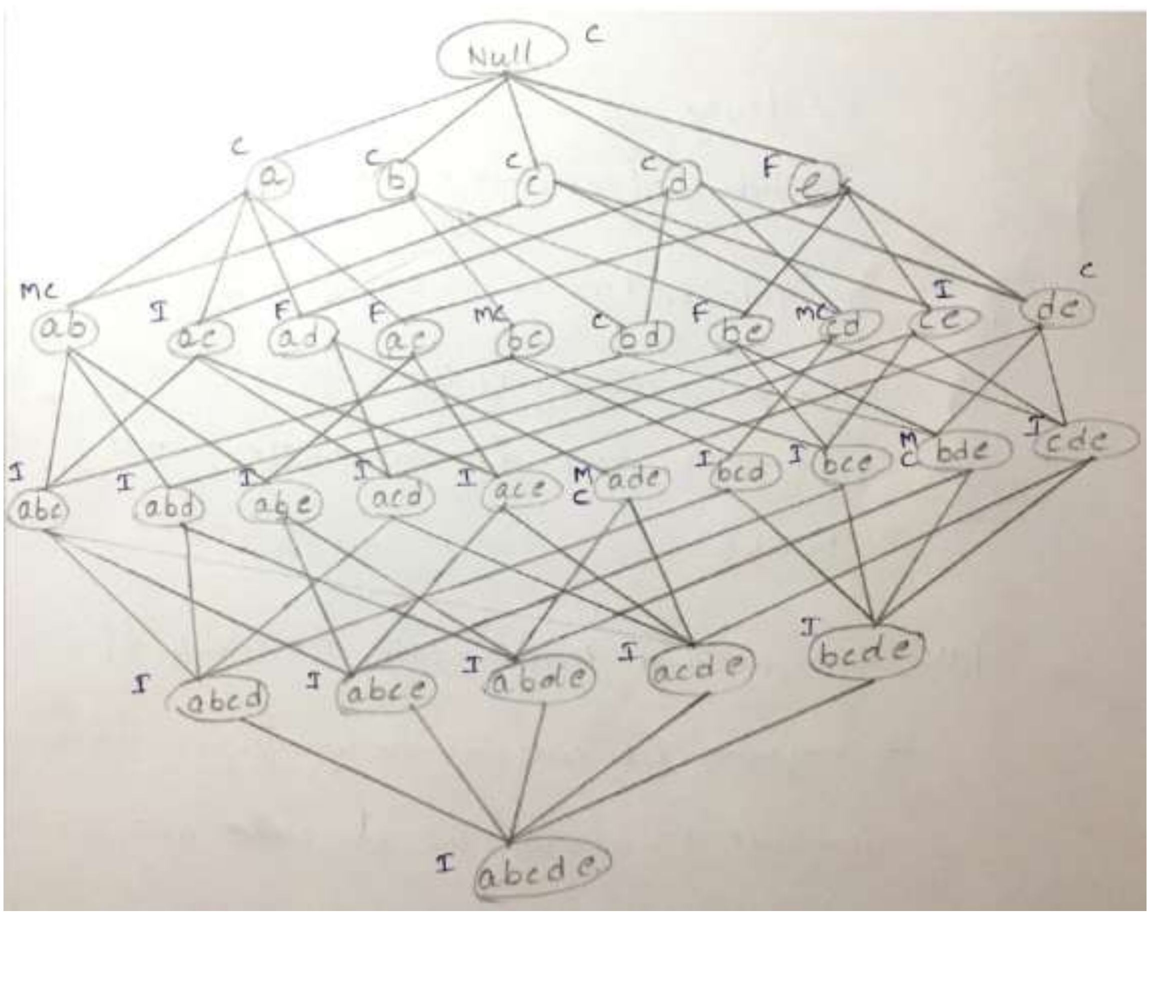
We have total 16 F and total nodes include root and bottom 32.
Percentage= 16/32 = 50%

3. What is the pruning ratio of the Apriori algorithm on this data set?
   Pruning ratio = N/total
   Here we can see we have 11 nodes so we can calculate the percentage = 11/32 = 34.4%

4. What is the false alarm rate (i.e., percentage of candidate item sets that are found to be infrequent after performing support counting)?

   For false alarm rate we are interested in I: infrequent item. So we have total 5 I and percentage will be 5/32 = 15.6%

Problem 12.



Calculations:

MinSup = 3

Transactions 624:

1. {a,b,d,e}
2. {b,c,d}
3. {a,b,d,e}
4. {a,c,d,e}
5. {b,c,d,e}
6. {b,d,e}
7. {c,d}
8. {a,b,c}
9. {a,d,e}
10. {b,d}

Step 1	Sup.
a	5
b	7
c	5
d	9
e	6

Step 2	Sup.
ab	3
~~ac~~	~~2~~ ×
ad	4
ae	4
bc	3
bd	6
be	4
cd	4
~~ce~~	~~2~~ ×
de	6.

Step 3	Sup.
abc	1
abd	2
abe	2
bcd	2
bce	1
acd	1
ace	1
ade	4 max
bde	4
cde	2.

maximal frequent itemsets are:

ade – 4

bde – 4

Problem 13.

a) {b}→{c}

	C	Not C
B	3	4
Not B	2	1

{a}→{d}

	D	Not D
A	4	1
Not A	5	0

{b}→{d}

	D	Not D
B	6	1
Not B	3	0

{e}→{c}

	C	Not C
E	2	4
Not E	3	1

{C}→{a}

	A	Not A
C	2	3
Not C	3	2

b) Total transactions = 10

Support:	Value	Rank
S(b to c)	3/10 = 0.3	3
S(a to d)	0.4	2
S(b to d)	0.6	1
S(e to c)	0.2	4
S(c to a)	0.2	4

Confidence	Value	Rank
C(b to c)	3/7 = 0.42	3
C(a to d)	0.8	2
C(b to d)	0.85	1
C(e to c)	0.33	5
C(c to a)	0.4	4

Interest:

Interest	Value	Rank
I(b to c)	0.3*0.5/0.77 = 0.214	3
I(a to d)	0.72	2
I(b to d)	0.771	1
I(e to c)	0.16	5
I(c to a)	0.2	4

IS(X→Y)=P(X, Y)P(X)P(Y).

IS = P(X,Y)/(squrt(P(X)P(Y))

IS	Value	Rank
IS(b to c)	0.3/squrt (0.5*0.77) = 0.507	3
IS(a to d)	0.596	2
IS(b to d)	0.756	1
IS(e to c)	0.365	5
IS(c to a)	0.4	4

Klosgen:

$$\text{Klosgen}(x \rightarrow y) = \sqrt{P(x,y)} \times P((y|x) - P(y))$$

$$\text{where} \quad P(y|x) = \frac{P(x,y)}{P(x)}$$

Klosgen	Value	Rank
Klosgen (b to c)	= - 0.039	2
Klosgen (a to d)	= - 0.063	4
Klosgen (b to d)	= - 0.033	1
Klosgen (e to c)	= - 0.075	5
Klosgen (c to a)	= - 0.045	3

Odd Ratio:

$$\text{Odds ratio}(x \rightarrow y) = \frac{P(x,y)\, P(\bar{x},\bar{y})}{P(\bar{x},y) . P(x,\bar{y})}$$

Odd Ratio	Value	Rank
Odd Ratio (b to c)	= 0.375	2
Odd Ratio (a to d)	= 0	4
Odd Ratio (b to d)	= 0.16	3
Odd Ratio (e to c)	= 0	4
Odd Ratio (c to a)	= 0.44	1

Problem 20.

Table 1:

	B	Not B
A	9	1
Not A	1	89

Support:
S(A) = 10/100 = 0.1
S(B) = 10/100 = 0.1
S(A, B) = 9/100 = 0.09
Interest:
I(A, B) = P(A, B)/P(A)P(B) = 900/100 = 9

Correlation Coefficient:
= f11*f00 – f01*f10/squrt(f1+*f+1*f0+*f+0)
= 0.89

Confident:
C(A to B) = 0.9
C(B to A) = 0.9

Table 2:

	B	Not B
A	89	1
Not A	1	9

Support:
S(A) = 90/100 = 0.9
S(B) = 90/100 = 0.9
S(A, B) = 89/100 = 0.89
Interest:
I(A, B) = P(A, B)/P(A)P(B) = 89*100/90*90 = 1.09

Correlation Coefficient:
= f11*f00 – f01*f10/squrt(f1+*f+1*f0+*f+0)
= 0.89

Confident:
C(A to B) = 0.98
C(B to A) = 0.98

What conclusions can you draw from the results of (a) and (b)?

Interest, Support and Confidence are non-invariant whereas Correlation Coefficient is invariant. The Correlation Coefficient does not change even when the inversed. This is due to the Correlation Coefficient properties which takes both presence and absence into account.

problem 2.1

In [36]:
```
import numpy as np
import pandas as pd
import mlxtend as ml # for apriori model
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [24]:
```
dataset = pd.read_excel('https://archive.ics.uci.edu/ml/machine-learning-datab
ases/00352/Online%20Retail.xlsx')
dataset.head(3)
```

Out[24]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Co
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Uni King
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	Uni King
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	Uni King

In [25]:
```
dataset['Description'] = dataset['Description'].str.strip()
dataset.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
dataset['InvoiceNo'] = dataset['InvoiceNo'].astype('str')
dataset = dataset[~dataset['InvoiceNo'].str.contains('C')]
basket = (dataset[dataset['Country']=="France"].groupby(['InvoiceNo', 'Descrip
tion'])['Quantity'].sum().unstack().reset_index().fillna(0).set_index('Invoice
No'))
```

In [26]:
```
basket.head(3)
```

Out[26]:

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 PENCIL SMAL TUBE RE RETROSPO
InvoiceNo						
536370	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
536974	0.0	0.0	0.0	0.0	0.0	0.0

3 rows × 1563 columns

In [27]:
```
## chnage numbers into binary 0 and 1, all positive will be 1 AND all negative
will be 0
def encode_data(datapoint):
    if datapoint <= 0:
        return 0
    if datapoint >= 1:
        return 1
```

In [28]:
```
basket_set= basket.applymap(encode_data)
```

In [29]:
```
basket_set.drop('POSTAGE', inplace=True, axis=1)
basket_set.head(3)
```

Out[29]:

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 PENCIL SMAL TUBE RE RETROSPO
InvoiceNo						
536370	0	0	0	0	0	0
536852	0	0	0	0	0	0
536974	0	0	0	0	0	0
537065	0	0	0	0	0	0
537463	0	0	0	0	0	0
...	...	...	...	...	...	...
580986	0	0	0	0	0	0
581001	0	0	0	0	0	0
581171	0	0	0	0	0	0
581279	0	0	0	0	0	0
581587	0	0	0	0	0	0

392 rows × 1562 columns

In [30]:
```
frequent_itemsets = apriori(basket_set, min_support=0.05, use_colnames=True)
```

In [33]:
```
frequent_itemsets = frequent_itemsets.sort_values(by='support', ascending=False)
frequent_itemsets.head(10)
```

Out[33]:

	support	itemsets
46	0.188776	(RABBIT NIGHT LIGHT)
52	0.181122	(RED TOADSTOOL LED NIGHT LIGHT)
44	0.170918	(PLASTERS IN TIN WOODLAND ANIMALS)
40	0.168367	(PLASTERS IN TIN CIRCUS PARADE)
59	0.158163	(ROUND SNACK BOXES SET OF4 WOODLAND)
26	0.153061	(LUNCH BAG RED RETROSPOT)
31	0.142857	(LUNCH BOX WITH CUTLERY RETROSPOT)
65	0.137755	(SET/6 RED SPOTTY PAPER CUPS)
50	0.137755	(RED RETROSPOT MINI CASES)
42	0.137755	(PLASTERS IN TIN SPACEBOY)

In [42]:
```
# get association rule form confident
rules_confidence = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)
rules_confidence.sort_values(by='confidence', ascending = False).head(2)
```

Out[42]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
10	(SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.09949	0.975	7.644000
12	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.09949	0.975	7.077778

In [43]:
```
# get association rule form confident
rules_confidence = association_rules(frequent_itemsets, metric="lift", min_thr
eshold=0.5)
rules_confidence.sort_values(by='lift', ascending = False).head(2)
```

Out[43]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
82	(PACK OF 6 SKULL PAPER CUPS)	(PACK OF 6 SKULL PAPER PLATES)	0.063776	0.056122	0.05102	0.800000	14.254545
83	(PACK OF 6 SKULL PAPER PLATES)	(PACK OF 6 SKULL PAPER CUPS)	0.056122	0.063776	0.05102	0.909091	14.254545

Itemset {RABBIT NIGHT LIGHT} has highest support = 0.188776

Association rules with highest lift = 14.254545

1- {Pack of 6 Skull Paper Cups} → {Pack of 6 Skull Paper Plates}

a. Consequent -> {Pack of 6 Skull Paper Plates}

b. Antecedent -> {Pack of 6 Skull Paper Cups}

2- {Pack of 6 Skull Paper Plates} → {Pack of 6 skull Paper Cups}

a. Consequent -> {Pack of 6 Skull Paper Cups}

b. Antecedent -> {Pack of 6 Skull Paper Plates}

Association rules with highest confidence = 0.975000

1- {SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...} → {SET/6 RED SPOTTY PAPER CUPS}

a. Antecedent -> {SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...}

b. Consequent -> {SET/6 RED SPOTTY PAPER CUPS}

2- {SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...} → {SET/6 RED SPOTTY PAPER PLATES}

a. Antecedent -> {SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...}

b. Consequent -> {SET/6 RED SPOTTY PAPER PLATES}

The rule with the highest lift is not the same as the rule with highest confidence. Because higher the confidence, greater the chances of the consequent being purchased. The larger the lift, the greater the link (correlation coff)between the two items.

Problem 2.2

In [44]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving 75000-out2-binary.csv to 75000-out2-binary.csv
```

In [45]:
```
import io
dataset_75 = pd.read_csv(io.BytesIO(uploaded['75000-out2-binary.csv']))
# Dataset is now stored in a Pandas Dataframe
```

In [48]:
```
dataset_75.head(3)
```

Out[48]:

	Transaction Number	Chocolate Cake	Lemon Cake	Casino Cake	Opera Cake	Strawberry Cake	Truffle Cake	Chocolate Eclair	Coffee Eclair
0	1	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	1
2	3	0	0	0	1	0	0	0	0

In [49]:
```
# Chocolate Coffee and Chocolate Cake items
# confusion matrix for Chocolate Coffee and Chocolate Cake
confusion_matrix = pd.crosstab(dataset_75['Chocolate Coffee'],dataset_75['Choc
olate Cake'], margins= True )
```

In [50]:
```
print(confusion_matrix)
```

```
Chocolate Cake       0     1    All
Chocolate Coffee
0                65802  2962  68764
1                 2933  3303   6236
All              68735  6265  75000
```

In [56]:
```
## f parameters
f11 = 3303; f00 = 65802; f10 = 2933; f01=2962
f0p = 68764; f1p = 6236; fp0 = 68735; fp1 = 6265
N = 75000
```

In [58]:
```
import math
```

In [61]:
```
cor_coff = ((f11*f00 - f01*f10)/(math.sqrt(f0p*f1p*fp0*fp1)))

print("Correlation Coefficient Φ for Coffee and cake:", cor_coff)
```

```
Correlation Coefficient Φ for Coffee and cake: 0.4855664925278768
```

In [59]:
```
math.sqrt(f0p*f1p*fp0*fp1)
```

Out[59]:
```
429717583.9167297
```

In [62]:
```
# Chocolate Cake and Chocolate Coffee items
# confusion matrix for Chocolate Cake and Chocolate Coffee items
confusion_matrix_2 = pd.crosstab(dataset_75['Chocolate Cake'],dataset_75['Choc
olate Coffee'], margins= True )
print(confusion_matrix_2)
```

```
Chocolate Coffee      0     1    All
Chocolate Cake
0                 65802  2933  68735
1                  2962  3303   6265
All               68764  6236  75000
```

In [63]:
```
## f parameters new
f11 = 3303; f00 = 65802; f10 = 2962; f01 = 2933
fp0 = 68764; fp1 = 6236; f0p = 68735; f1p = 6265
N = 75000
```

In [64]:
```
cor_coff_2 = ((f11*f00 - f01*f10)/(math.sqrt(f0p*f1p*fp0*fp1)))

print("Correlation Coefficient Φ for Cake and Coffee:", cor_coff_2)
```

```
Correlation Coefficient Φ for Cake and Coffee: 0.4855664925278768
```

Since both the correlation coefficient are equal, So we can say that the two itemsets (Chocolate Coffee and Chocolate Cake) anf (Chocolate Cake and Chocolate Coffee) are symmetric binary variables.