Problem 1

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [2]:
```
data=pd.read_excel("https://archive.ics.uci.edu/ml/machine-learning-database
data.head()
```

Out[2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	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	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	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	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

In [3]:
```
1 data['Country'].value_counts().sort_values()
```

Out[3]:
```
Saudi Arabia                10
Bahrain                     19
Czech Republic              30
Brazil                      32
Lithuania                   35
Lebanon                     45
RSA                         58
European Community          61
United Arab Emirates        68
Malta                      127
Greece                     146
Canada                     151
Iceland                    182
Singapore                  229
Hong Kong                  288
USA                        291
Israel                     297
Poland                     341
Japan                      358
Denmark                    389
Austria                    401
Unspecified                446
Sweden                     462
Cyprus                     622
Finland                    695
Channel Islands            758
Italy                      803
Norway                    1086
Australia                 1259
Portugal                  1519
Switzerland               2002
Belgium                   2069
Netherlands               2371
Spain                     2533
EIRE                      8196
France                    8557
Germany                   9495
United Kingdom          495478
Name: Country, dtype: int64
```

In [4]:
```
1 data.shape
```

Out[4]:
```
(541909, 8)
```

In [5]:
```
dataFrance=data[data['Country']=='France']
dataFrance=dataFrance[['InvoiceNo','Description']]
dataFrance.head()
```

Out[5]:

	InvoiceNo	Description
26	536370	ALARM CLOCK BAKELIKE PINK
27	536370	ALARM CLOCK BAKELIKE RED
28	536370	ALARM CLOCK BAKELIKE GREEN
29	536370	PANDA AND BUNNIES STICKER SHEET
30	536370	STARS GIFT TAPE

In [6]:
```
newData=pd.DataFrame(columns=dataFrance["Description"].unique())
newData.insert(loc=0,column="InvoiceNo",value=dataFrance["InvoiceNo"].unique
newData.head()
```

Out[6]:

	InvoiceNo	ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET	STARS GIFT TAPE	INFLATABLE POLITICAL GLOBE	VINTAGE HEADS AND TAILS CARD GAME	SET RETR T
0	536370	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	536852	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	536974	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	537065	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	537463	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 1566 columns

In [7]:
```
1 newData = newData.replace(np.nan, False)
2 newData.head()
```

Out[7]:

	InvoiceNo	ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET	STARS GIFT TAPE	INFLATABLE POLITICAL GLOBE	VINTAGE HEADS AND TAILS CARD GAME	SET RETR T
0	536370	False	False	False	False	False	False	False	
1	536852	False	False	False	False	False	False	False	
2	536974	False	False	False	False	False	False	False	
3	537065	False	False	False	False	False	False	False	
4	537463	False	False	False	False	False	False	False	

5 rows × 1566 columns

In [8]:
```
1 newData.shape
```

Out[8]: (461, 1566)

In [9]:
```
1 for idx, row in dataFrance.iterrows():
2     invoice, item= row["InvoiceNo"], row["Description"]
3     newData.loc[newData['InvoiceNo'] == invoice, item]=True
```

In [10]:
```
1 newData
```

Out[10]:

	InvoiceNo	ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET	STARS GIFT TAPE	INFLATABLE POLITICAL GLOBE	VINTAGE HEADS AND TAILS CARD GAME	S RE
0	536370	True	True	True	True	True	True	True	
1	536852	False	False	False	False	False	False	True	
2	536974	False	False	False	False	False	False	False	
3	537065	True	True	True	False	False	False	False	
4	537463	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
456	581001	True	False	True	False	False	False	False	
457	581171	False	False	False	False	False	False	False	
458	581279	False	False	False	False	False	False	False	
459	C581316	False	False	False	False	False	False	False	
460	581587	True	True	True	False	False	False	False	

461 rows × 1566 columns

In [11]:
```
1 newData = newData.drop("InvoiceNo", axis=1)
```

In [12]:
```
1 frequentItemsets = apriori(newData, min_support=0.05, use_colnames=True)
2 frequentItemsets.head()
```

Out[12]:

	support	itemsets
0	0.086768	(ALARM CLOCK BAKELIKE PINK)
1	0.080260	(ALARM CLOCK BAKELIKE RED)
2	0.084599	(ALARM CLOCK BAKELIKE GREEN)
3	0.138829	(ROUND SNACK BOXES SET OF4 WOODLAND)
4	0.106291	(SPACEBOY LUNCH BOX)

In [13]:
```
confidence = association_rules(frequentItemsets, metric="confidence", min_th
confidence.head()
```

Out[13]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	c
0	(JUMBO BAG WOODLAND ANIMALS)	(POSTAGE)	0.065076	0.67462	0.065076	1.0	1.482315	0.021174	

In [14]:
```
lift = association_rules(frequentItemsets, metric="lift", min_threshold=11)
lift.head()
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(CHILDRENS CUTLERY SPACEBOY)	(CHILDRENS CUTLERY DOLLY GIRL)	0.058568	0.062907	0.05423	0.925926	14.719029	0.050546
1	(CHILDRENS CUTLERY DOLLY GIRL)	(CHILDRENS CUTLERY SPACEBOY)	0.062907	0.058568	0.05423	0.862069	14.719029	0.050546

From the above tables we can see tThe rule with the highest confidence and lift are not same. ***confidence*** determines how frequently items in Y appear in transactions that contain X. ***Interest factor*** is able to detect meaningful patterns in the data which makes it one of the best measures to analyze indeoendence of varaibles.

Problem 2

In [34]:
```
binaryData=pd.read_csv('75000-out2-binary.csv')
binaryData.head()
```

Out[34]:

	Transaction Number	Chocolate Cake	Lemon Cake	Casino Cake	Opera Cake	Strawberry Cake	Truffle Cake	Chocolate Eclair	Coffee Eclair	Vanilla Eclair
0	1	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	1	0
2	3	0	0	0	1	0	0	0	0	0
3	4	0	0	0	0	0	1	0	0	0
4	5	0	0	0	0	0	0	1	0	0

5 rows × 51 columns

In [35]:
```
newData1=binaryData[['Chocolate Cake','Chocolate Coffee']]
newData1.head()
```

Out[35]:

	Chocolate Cake	Chocolate Coffee
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

In [36]:
```
newData1['Chocolate Cake'].value_counts()
```

Out[36]:
```
0    68735
1     6265
Name: Chocolate Cake, dtype: int64
```

In [37]:
```
newData1['Chocolate Coffee'].value_counts()
```

Out[37]:
```
0    68764
1     6236
Name: Chocolate Coffee, dtype: int64
```

In [43]:
```
groupCount=newData1.groupby([newData1['Chocolate Cake']==1,newData1['Chocola
groupCount
```

Out[43]:

		Chocolate Cake	Chocolate Coffee
Chocolate Cake	Chocolate Coffee		
False	False	65802	65802
	True	2933	2933
True	False	2962	2962
	True	3303	3303

In [45]:
```
print(newData1["Chocolate Coffee"].corr(newData1["Chocolate Cake"]))

```

```
0.48556649252787826
```

In [46]:
```
print(newData1["Chocolate Cake"].corr(newData1["Chocolate Coffee"]))
```

```
0.48556649252787837
```

From the above results we can see that the two items are symmetric binary variables
From the above table we can see that the items coexists ort does not exist at all which make the corealation same irrespective of the order.