```
**Data Mining **
```

In [8]:

```
import pandas as pd
import numpy as np
import fim
```

```
Reading the cleaned data into a variable called data
```

In [2]:

```
data = pd.read_csv(r'F:\Docs\Big data\Assignment\cleaned_data.csv')
print(data.head())
print(data.shape)
data_list = data.values
print(data.values)
```

```
   Airline ID                          Name ICAO       Callsign  \
0           2                   135 Airways  GNL        GENERAL   
1           3                 1Time Airline  RNX        NEXTIME   
2           6  223 Flight Unit State Airline  CHD  CHKALOVSK-AVIA   
3           7              224th Flight Unit  TTF     CARGO UNIT   
4           8                    247 Jet Ltd  TWF   CLOUD RUNNER   

          Country Active
0   United States      N
1    South Africa      Y
2          Russia      N
3          Russia      N
4  United Kingdom      N
(5297, 6)
[[2 '135 Airways' 'GNL' 'GENERAL' 'United States' 'N']
 [3 '1Time Airline' 'RNX' 'NEXTIME' 'South Africa' 'Y']
 [6 '223 Flight Unit State Airline' 'CHD' 'CHKALOVSK-AVIA' 'Russia' 'N']
 ...
 [21248 'GX Airlines' 'CBG' 'SPRAY' 'China' 'Y']
 [21251 'Lynx Aviation (L3/SSX)' 'SSX' 'Shasta' 'United States' 'N']
 [21317 'Svyaz Rossiya' 'SJM' 'RussianConnecty' 'Russia' 'Y']]
```

```
This function calculates the lfi, basically what it does is that it takes the frequent
item set calculated by the aptriori and fp-growth algo and finds the itemset of maximum
length.
```

In [3]:

```
def lfi(freq_list):
    len_of_freq_list = [len(ele) for ele in freq_list]
    lfi = freq_list[len_of_freq_list.index(max(len_of_freq_list))]
    print("lfi is : ")
    print(lfi)
```

```
This function calculates the frequent items sets using appriori and fp-growth.
The rules are evaluated using lift and confidence.
lift={{P(A|B)}/{P(A)*P(B)}}}
```

confidence(A→B)=P(B|A)
where A and B beling to frequent itemset

In [4]:

```
def fi(data):
    print("Using apriori for fim : ")
    freq_list = fim.apriori(tracts = data, supp = 5)
    print("The frequent item list is : ")
    print(freq_list)
    rules = fim.apriori(tracts = data, target = 'r', eval = 'c', report = 'c')
    print("The rules are : ")
    print(rules)
    rules = fim.apriori(tracts = data, target = 'r', eval = 'l', report = 'l')
    print("The rules are (evaluated with lift): ")
    print(rules)
    print("lfi using apriori : ")
    lfi(freq_list)



    print("Using fp-growth for fim : ")
    freq_list = fim.fpgrowth(tracts = data, supp = 5)
    print("The frequent item list is : ")
    print(freq_list)
    rules = fim.fpgrowth(tracts = data, target = 'r', eval = 'c', report = 'c', conf = 60)
    print("The rules are (evaluated with confidence): ")
    print(rules)
    rules = fim.fpgrowth(tracts = data, target = 'r', eval = 'l', report = 'l',  conf = 60)
    print("The rules are (evaluated with lift): ")
    print(rules)

    print("lfi using fpgrowth is : ")
    lfi(freq_list)
```

The two algo used for calculating cfi and mfi are, IsTa and RElim.
IsTa : IsTa is a program to find closed frequent item sets by intersecting transactions (Intersecting Transactions), which is based on the insight that an item set is closed if it is the intersection of all transactions that contain it. Such an approach can be highly competitive in special cases, namely if there are few transactions and (very) many items, which is a common situation in biological data sets like gene expression data.

RElim : RElim is a program to find frequent item sets (also closed and maximal) with the relim algorithm (recursive elimination), which is inspired by the FP-growth algorithm, but does its work without prefix trees or any other complicated data structures. The main strength of this algorithm is not its speed (although it is not slow, but even outperforms Apriori and Eclat on some data sets), but the simplicity of its structure. Basically all the work is done in one recursive function of fairly few lines of code.

In [5]:

```
def cfi(data):
    print("Using relim for cfi : ")
    freq_list = fim.relim(tracts = data, target = 'c', supp = 5)
    print("The frequent item list is : ")
    print(freq_list)

    print("Using ista for cfi : ")
    freq_list = fim.ista(tracts = data, target = 'c', algo = 'p',supp = 5)
    print("The frequent item list is : ")
    print(freq_list)
```

In [6]:

```
def mfi(data):
    print("Using relim for mfi : ")
    freq_list = fim.relim(tracts = data, target = 'm', supp = 5)
    print("The frequent item list is : ")
    print(freq_list)

    print("Using ista for mfi : ")
    freq_list = fim.ista(tracts = data, target = 'm', mode = 'z', algo = 'p',supp = 5)
    print("The frequent item list is : ")
    print(freq_list)
```

```
Runner cod
```

In [7]:

```
mfi(data_list)
cfi(data_list)
fi(data_list)
```

```
Using relim for mfi :
The frequent item list is :
[(('Canada', 'N'), 265), (('United Kingdom', 'N'), 323), (('Mexico', 'N'), 4
20), (('Y',), 883), (('United States', 'N'), 858)]
Using ista for mfi :
The frequent item list is :
[(('N', 'United States'), 858), (('N', 'Mexico'), 420), (('N', 'United Kingd
om'), 323), (('N', 'Canada'), 265), (('Y',), 883)]
Using relim for cfi :
The frequent item list is :
[(('Canada', 'N'), 265), (('Canada',), 292), (('United Kingdom', 'N'), 323),
(('United Kingdom',), 362), (('Mexico', 'N'), 420), (('Mexico',), 428),
(('Y',), 883), (('United States', 'N'), 858), (('United States',), 975),
(('N',), 4414)]
Using ista for cfi :
The frequent item list is :
[(('N', 'United States'), 858), (('N', 'Mexico'), 420), (('N', 'United Kingd
om'), 323), (('N', 'Canada'), 265), (('N',), 4414), (('United States',), 97
5), (('Y',), 883), (('Mexico',), 428), (('United Kingdom',), 362), (('Canad
a',), 292)]
Using apriori for fim :
The frequent item list is :
[(('Canada',), 292), (('Canada', 'N'), 265), (('United Kingdom',), 362),
(('United Kingdom', 'N'), 323), (('Mexico',), 428), (('Mexico', 'N'), 420),
(('Y',), 883), (('United States',), 975), (('United States', 'N'), 858),
(('N',), 4414)]
The rules are :
[('N', ('United States',), 0.88), ('N', (), 0.8333018689824429)]
The rules are (evaluated with lift):
[('N', ('United States',), 1.056039873130947), ('N', (), 1.0)]
lfi using apriori :
lfi is :
(('Canada',), 292)
Using fp-growth for fim :
The frequent item list is :
[(('N',), 4414), (('United States', 'N'), 858), (('United States',), 975),
(('Y',), 883), (('Mexico', 'N'), 420), (('Mexico',), 428), (('United Kingdo
m', 'N'), 323), (('United Kingdom',), 362), (('Canada', 'N'), 265), (('Canad
a',), 292)]
The rules are (evaluated with confidence):
[('N', (), 0.8333018689824429), ('N', ('United States',), 0.88)]
The rules are (evaluated with lift):
[('N', (), 1.0), ('N', ('United States',), 1.056039873130947)]
lfi using fpgrowth is :
lfi is :
(('N',), 4414)
```