

**Name:- Raj Khatri**  
**Roll Number:- AC-1235**  
**Subject:- Data Mining**  
**Semester:- 6**  
**Practical - 4**

The dataset is bread basket

In [1]:

```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori
```

In [2]:

```
grocery_df = pd.read_csv("bread basket.csv")
```

In [3]:

```
grocery_df
```

Out[3]:

Transaction		Item	date_time	period_day	weekday_weekend
0	1	Bread	30-10-2016 09:58	morning	weekend
1	2	Scandinavian	30-10-2016 10:05	morning	weekend
2	2	Scandinavian	30-10-2016 10:05	morning	weekend
3	3	Hot chocolate	30-10-2016 10:07	morning	weekend
4	3	Jam	30-10-2016 10:07	morning	weekend
...	...	...	...	...	...
20502	9682	Coffee	09-04-2017 14:32	afternoon	weekend
20503	9682	Tea	09-04-2017 14:32	afternoon	weekend
20504	9683	Coffee	09-04-2017 14:57	afternoon	weekend
20505	9683	Pastry	09-04-2017 14:57	afternoon	weekend
20506	9684	Smoothies	09-04-2017 15:04	afternoon	weekend

20507 rows x 5 columns

The items are converted into list according to transaction

In [4]:

```
df = grocery_df.groupby('Transaction')['Item'].apply(
    list).reset_index(name='Item List')
df
```

Out [4]:

Transaction		Item List
0	1	[Bread]
1	2	[Scandinavian, Scandinavian]
2	3	[Hot chocolate, Jam, Cookies]
3	4	[Muffin]
4	5	[Coffee, Pastry, Bread]
...	...	...
9460	9680	[Bread]
9461	9681	[Truffles, Tea, Spanish Brunch, Christmas common]
9462	9682	[Muffin, Tacos/Fajita, Coffee, Tea]
9463	9683	[Coffee, Pastry]
9464	9684	[Smoothies]

9465 rows × 2 columns

The data is encoded

In [5]:

```
from mlxtend.preprocessing import TransactionEncoder
encoded = TransactionEncoder()
encoded_array = encoded .fit_transform(df['Item List'])
encoded_array
```

Out [5]:

```
array([[False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       ...,
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False]])
```

In [10]:

```
arr_to_df = pd.DataFrame(encoded_array, columns=encoded.columns_)
```

Minimum support is taken as 3% and threshold as 50%

In [ ]:

```
frequent_itemsets = apriori(arr_to_df, min_support=0.03, use_colnames=True)
print(frequent_itemsets)
```

	support	itemsets
0	0.036344	(Alfajores)
1	0.327205	(Bread)
2	0.040042	(Brownie)
3	0.103856	(Cake)
4	0.478394	(Coffee)
5	0.054411	(Cookies)
6	0.039197	(Farm House)
7	0.058320	(Hot chocolate)
8	0.038563	(Juice)
9	0.061807	(Medialuna)
10	0.038457	(Muffin)
11	0.086107	(Pastry)
12	0.071844	(Sandwich)
13	0.034548	(Scone)
14	0.034443	(Soup)

```
15 0.142631 (Tea)
16 0.033597 (Toast)
17 0.090016 (Coffee, Bread)
18 0.054728 (Coffee, Cake)
19 0.035182 (Medialuna, Coffee)
20 0.047544 (Coffee, Pastry)
21 0.038246 (Sandwich, Coffee)
22 0.049868 (Tea, Coffee)
```

In [ ]:

```
from mlxtend.frequent_patterns import association_rules
rules = association_rules(
    frequent_itemsets, metric="confidence", min_threshold=0.5)
rules
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Cake)	(Coffee)	0.103856	0.478394	0.054728	0.526958	1.101515	0.005044	1.102664
1	(Medialuna)	(Coffee)	0.061807	0.478394	0.035182	0.569231	1.189878	0.005614	1.210871
2	(Pastry)	(Coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351	1.164682
3	(Sandwich)	(Coffee)	0.071844	0.478394	0.038246	0.532353	1.112792	0.003877	1.115384