NAAN MUDHALVAN – IBM SKILL
ARTIFICIAL INTELLIGENCE
GROUP PROJECT

Project Title: Market Basket Insight
Phase 3 Submission

S.NO	GROUP MEMBERS NAME	NM ID	E-MAIL ID
1	DURAI BHUVANESH.CM	au820321106013	cmduraicmdurai12@gmail.com
2	SUJITHKUMAR.R	au820321106037	sujithkumarrao333@gmail.com
3	NARENDHIRAN.R	au820321106027	naveenrc430@gmail.com
4	VIMAL.M	au820321106039	mm4795231@gmail.com
5	KARUNAMOORTHI.K	au820321106022	karunamoorthy8012@gmail.com

MARKET BASKET ANALYSIS USING PYTHON

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together [1].

E.g. the rule **{cucumbers, tomatoes} -> {sunflower oil}** found in the sales data of a supermarket would indicate that if a customer buys cucumbers and tomatoes together, they are likely to also buy sunflower oil.

1. Import Libraries

For market basket analysis I'm going to use mlxtend. For other purposes (reading data, working with data, visualizing data) I'll use all well-known libraries like numpy, pandas etc.

In [1]:

```
import numpy as np
import pandas as pd
import squarify
import matplotlib.pyplot as plt

# for market basket analysis
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
```

2. Read data

In [2]:

```
df = pd.read_xlsx(' C:\marketbasket\Assignment1_Data.xlsx ', header = None)
```

In [3]:

```
df.head(5) # looking for the first 5 rows in dataset
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	shrimp	almonds	avocado	vegetables mix	green grap	whole wea	yams	cottage chee	energy drink	tomato jui	low fat yogu	green te	honey	salad	mineral water	salmon	antioxyd ant juice	frozen smoothie	spinach	olive oil

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
					es	tflour		se		ce	rt	a								
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

3. Visualize data

Here I decided to count all unique values through all columns and build some visualitions. E.g. if we have 5 'almonds' in first column, 3 'almonds' in second column etc, so, will have 8 'almonds' in total.

In [4]:

```
df_res = pd.DataFrame()
for i in range(len(df.columns)):
    df_res = df_res.append(df[i].value_counts())
```

In [5]:

```
df_res.head(3)
```

Out[5]:

	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	body spray	...	whole wheat flour	whole wheat pasta	whole wheat rice	yams	yogurt cake	zucchini	water spray	tea	napkins	asparagus
0	11.0	18.0	3.0	57.0	5.0	6.0	3.0	9.0	4.0	1.0	...	8.0	950.0	470.0	240.0	31.0	NaN	NaN	NaN	NaN	NaN
1	29.0	10.0	2.0	64.0	5.0	8.0	9.0	310.0	8.0	130.0	...	5.0	680.0	920.0	250.0	38.0	50.0	1.0	10.0	NaN	NaN
2	35.0	12.0	5.0	46.0	4.0	120.0	18.0	150.0	13.0	140.0	...	8.0	330.0	690.0	240.0	32.0	40.0	1.0	2.0	NaN	NaN

3 rows × 120 columns

In [6]:

```
df_sum = df_res.sum()
df_sum = df_sum.sort_values(ascending=False)
```

In [7]:

```
df_sum
```

Out[7]:

```
mineral water     1788.0
eggs              1348.0
spaghetti         1306.0
french fries      1282.0
chocolate         1230.0
                   ...
```

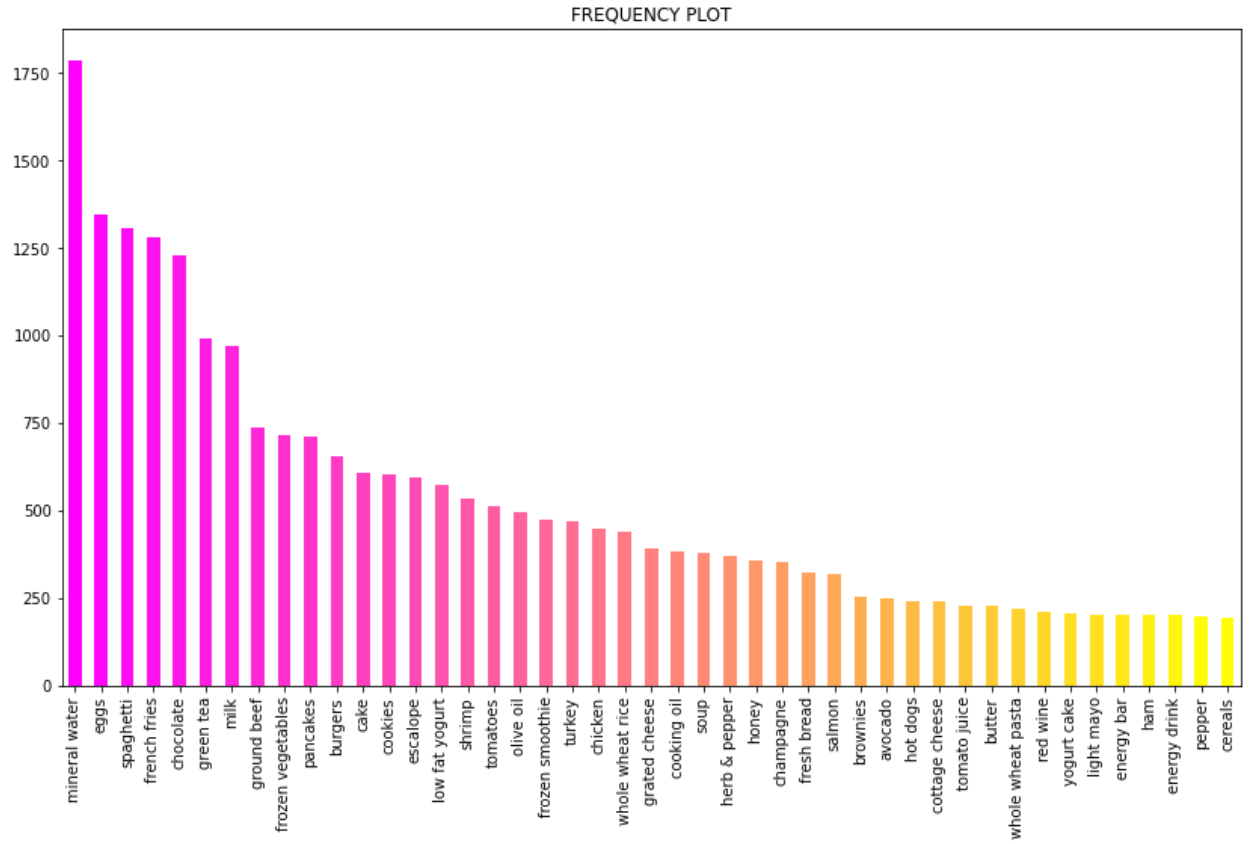
```
bramble            14.0
cream               7.0
napkins             5.0
water spray         3.0
 asparagus          1.0
Length: 120, dtype: float64
```

After counting all values through all columns, we can build a **frequency plot**.

```
In [8]:

linkcode
```

```
plt.figure(figsize=(14,8))
plt.title("FREQUENCY PLOT")
cnt = 45 # plot only first 'cnt' values
color = plt.cm.spring(np.linspace(0, 1, cnt))
df_sum.head(cnt).plot.bar(color = color)
plt.xticks(rotation = 'vertical')
plt.grid(False)
plt.axis('on')
plt.show()
```



Also we can frequency plot, but in the form of **heat map**:

```
In [9]:
plt.figure(figsize=(15,15))
cnt = 40 # plot only first 'cnt' values
color = plt.cm.hot(np.linspace(0, 1, cnt))
df_part = df_sum.head(cnt)
squarify.plot(sizes = df_part.values, label = df_part.index, alpha=.8, color = color,
text_kwargs={'fontsize':8})
plt.axis('off')
plt.show()
```