Capstone Project Report

Name: **Pooja Ramdas Kadam**
Course: AI & ML (Batch - 4)

Problem Statement
Perform Hierarchical Clustering from scratch and also using sklearn to perform wholesale customer segmentation based on their annual spending on products. You can use this dataset. Use the threshold to
1. Divide the dataset into two clusters.
2. To divide the dataset into k clusters, such that the distance between the two clusters is greater than a given threshold (this threshold can be anything passed to the function).

Prerequisites
Along with Python below packages needed to be installed

Matplotlib
Sklearn
Scipy
Pandas

Dataset Used
https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale%20customers%20data.csv

Implementation
Import required libraries and load data

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import scipy.cluster.hierarchy as shc
from sklearn.cluster import AgglomerativeClustering
```

Load data

In [2]:
```
df = pd.read_csv('Wholesale customers data.csv')
df.head(10)
```

Out[2]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185
5	2	3	9413	8259	5126	666	1795	1451
6	2	3	12126	3199	6975	480	3140	545
7	2	3	7579	4956	9426	1669	3321	2566
8	1	3	5963	3648	6192	425	1716	750
9	2	3	6006	11093	18881	1159	7425	2098

Normalize data

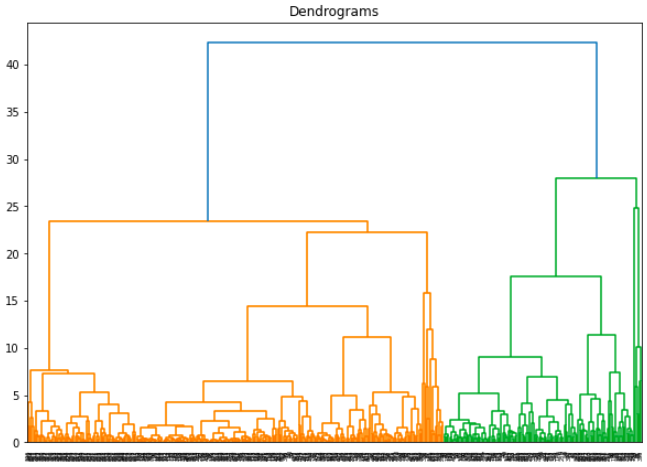
In [3]:
```
scaler = StandardScaler()
scaler = scaler.fit_transform(df)
df = pd.DataFrame(scaler, columns=df.columns)
df.head(10)
```

Out[3]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	1.448652	0.590668	0.052933	0.523568	-0.041115	-0.589367	-0.043569	-0.066339
1	1.448652	0.590668	-0.391302	0.544458	0.170318	-0.270136	0.086407	0.089151
2	1.448652	0.590668	-0.447029	0.408538	-0.028157	-0.137536	0.133232	2.243293
3	-0.690297	0.590668	0.100111	-0.624020	-0.392977	0.687144	-0.498588	0.093411
4	1.448652	0.590668	0.840239	-0.052396	-0.079356	0.173859	-0.231918	1.299347
5	1.448652	0.590668	-0.204806	0.334067	-0.297637	-0.496155	-0.228138	-0.026224
6	1.448652	0.590668	0.009950	-0.352316	-0.102849	-0.534512	0.054280	-0.347854
7	1.448652	0.590668	-0.349981	-0.113981	0.155359	-0.289315	0.092286	0.369601
8	-0.690297	0.590668	-0.477901	-0.291409	-0.185336	-0.545854	-0.244726	-0.275079
9	1.448652	0.590668	-0.474497	0.718495	1.151423	-0.394488	0.954031	0.203461

Visualize dendograms

In [4]:
```
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
dend = shc.dendrogram(shc.linkage(df, method='ward'))
```



Apply AgglomerativeClustering

In [5]:
```
clusters = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='ward')
clusters.fit_predict(df)
```

Out[5]:
```
array([0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1])
```

In [6]:
```
plt.scatter(df['Milk'], df['Grocery'], c=clusters.labels_)
```