PROJECT REPORT

Assignment Project Report: Construction of hashing tree
NAME: RISHI RAJ SINGH
AI AND ML B-4

Problem Statement:
Perform Hierarchical Clustering from scratch and also using sklearn to perform wholesale customer segmentation based on their annual spending on products. 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:
- Software: Python 3
Tools:
- Pandas
- Numpy
- Matplotlib

Method Used :
Hierarchical clustering is the hierarchical decomposition of the data based on group similarities. It allows us to build tree structures from data similarities and see how different sub-clusters relate to each other, and how far apart data points are. It gives us a tree-type structure based on the hierarchical series of nested clusters. A diagram called Dendrogram graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged, or clusters are broken apart. Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of

clusters, where each cluster is distinct from the other cluster, and the objects within each cluster are broadly similar to each other.

Implementation:

1. Load all required libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Loading Dataset

In [2]:
```
# Reading the libary
data= pd.read_csv('Wholesale customers data.csv')
data.head()
```

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

3. Pre-processing Data

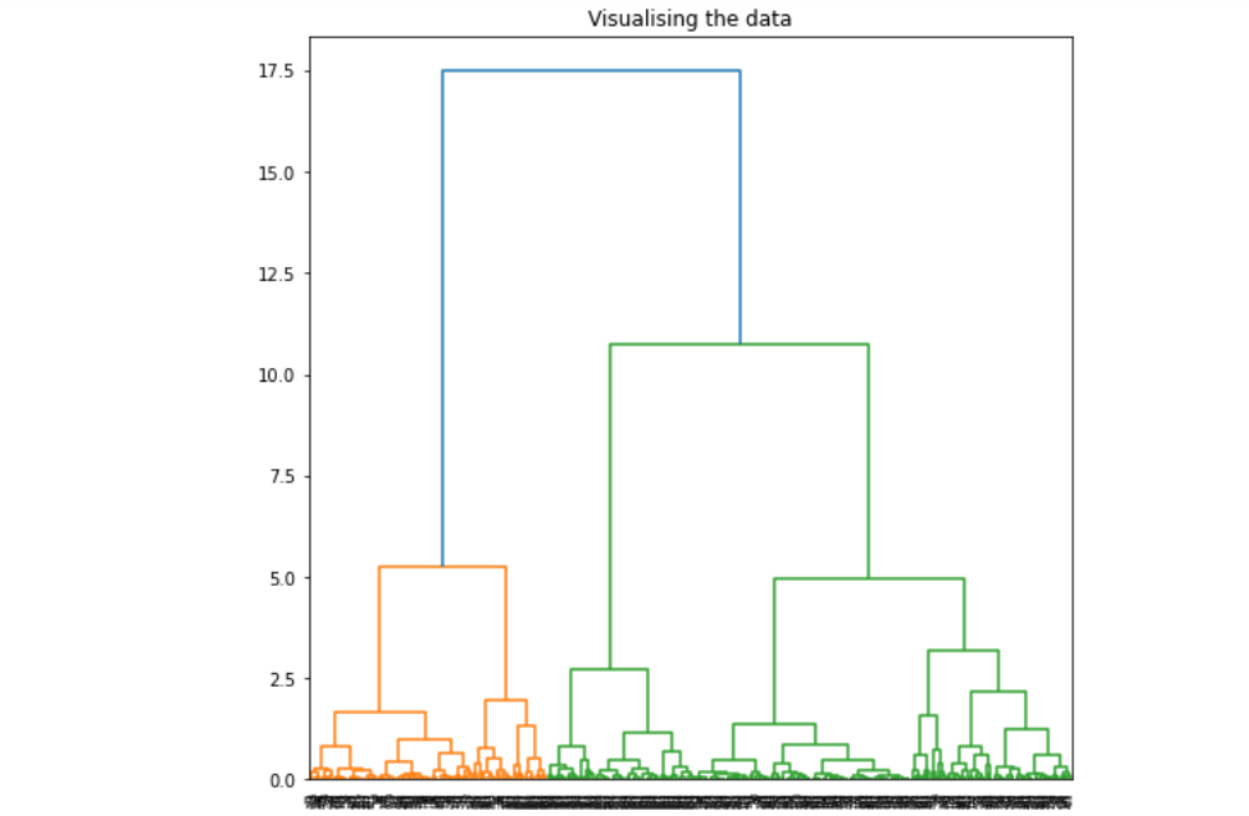
In [5]:
```
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.decomposition import PCA
```

4. Scaling Data

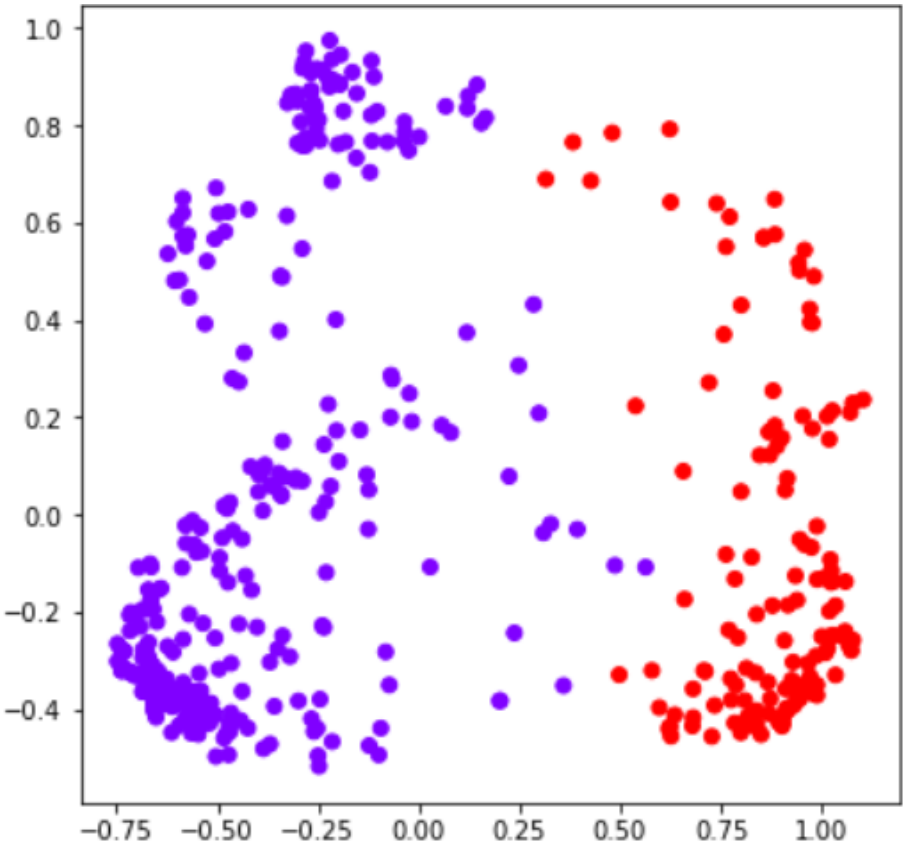
In [6]:

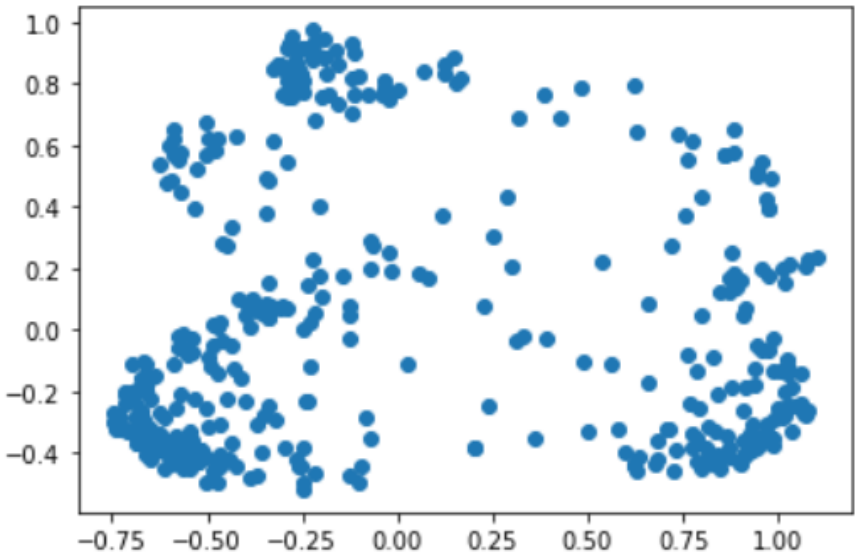
```
# Scaling the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# Normalizing the data
# follows a Gaussian distribution
data_normalized = normalize(data_scaled)

# Converting the numpy array into a pandas DataFrame
data_normalized = pd.DataFrame(data_normalized)
```

5. Plotting dendrograms



6. Cluster graph:


1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

Final:

```
clusters = hierarchical(data_principal,4)
colors = ['blue', 'red', 'purple', 'teal']
for cluster_index, cluster in enumerate(clusters):
    for point_index, point in enumerate(cluster):
        plt.plot([point[0]], [point[1]], marker='o', markersize=3, color=colors[cluster_index])
```