ML - EXP - 6

Atharva Prashant Pawar (9427) - [Batch - D]

Dataset

[ ] ↳ *1 cell hidden*

Code

```
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
import numpy as np
```

```
customer_data = pd.read_csv('hierarchical-clustering-with-python-and-scikit-learn-shopping-data.csv')
```

```
customer_data.shape
```

```
(200, 5)
```

```
customer_data.head(5)
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
customer_data.describe()
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
customer_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

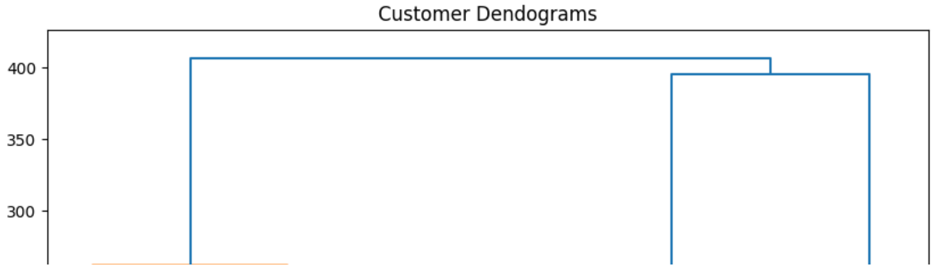
```
data = customer_data.iloc[:, 3:5].values
```

```
data
```

```
array([[ 15,  39],
       [ 15,  81],
       [ 16,   6],
       [ 16,  77],
       [ 17,  40],
       [ 17,  76],
       [ 18,   6],
       [ 18,  94],
       [ 19,   3],
       [ 19,  72],
       [ 19,  14],
       [ 19,  99],
       [ 20,  15],
       [ 20,  77],
       [ 20,  13],
       [ 20,  79],
       [ 21,  35],
       [ 21,  66],
       [ 23,  29],
       [ 23,  98],
       [ 24,  35],
       [ 24,  73],
       [ 25,   5],
       [ 25,  73],
       [ 28,  14],
       [ 28,  82],
       [ 28,  32],
       [ 28,  61],
       [ 29,  31],
       [ 29,  87],
       [ 30,   4],
       [ 30,  73],
       [ 33,   4],
       [ 33,  92],
       [ 33,  14],
       [ 33,  81],
       [ 34,  17],
       [ 34,  73],
       [ 37,  26],
       [ 37,  75],
       [ 38,  35],
       [ 38,  92],
       [ 39,  36],
       [ 39,  61],
       [ 39,  28],
       [ 39,  65],
       [ 40,  55],
       [ 40,  47],
       [ 40,  42],
       [ 40,  42],
       [ 42,  52],
       [ 42,  60],
       [ 43,  54],
       [ 43,  60],
       [ 43,  45],
       [ 43,  41],
       [ 44,  50],
       [ 44,  46],
```

```
import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))
plt.title("Customer Dendograms")
dend = shc.dendrogram(shc.linkage(data, method='ward'))
```



```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
labels_=cluster.fit_predict(data)
```

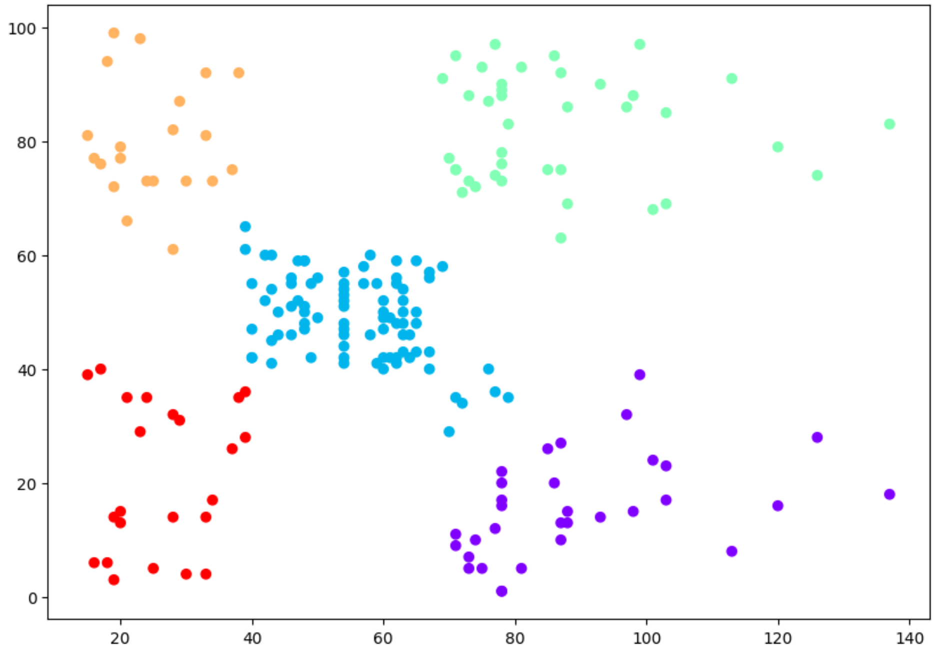
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_agglomerative.py:983: FutureWarning: Attribute `affinity` was deprecated i
  warnings.warn(
```

```
labels_
```

```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0, 2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2])
```

```
plt.figure(figsize=(10, 7))
plt.scatter(data[:,0], data[:,1], c=cluster.labels_, cmap='rainbow')
```

```
<matplotlib.collections.PathCollection at 0x79b036125ea0>
```



Adding Outliers

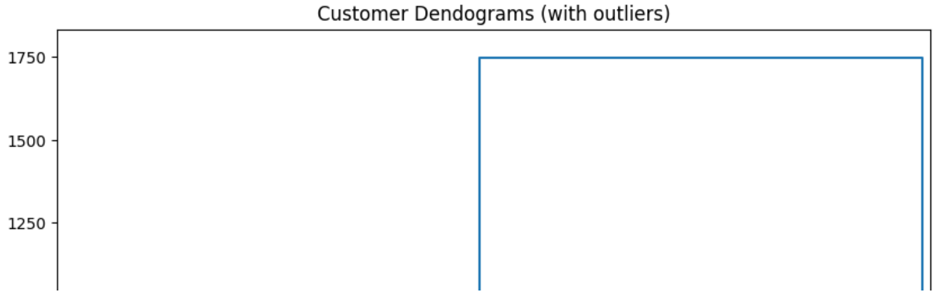
```
# Create some outliers in the data
outliers = np.array([[700, 100], [800, 500], [750, 150]])  # Example outlier points
data = customer_data.iloc[:, 3:5].values
data_with_outliers = np.vstack((data, outliers))

data_with_outliers
```

```
    array([[ 15,  39],
           [ 15,  81],
           [ 16,   6],
           [ 16,  77],
           [ 17,  40],
           [ 17,  76],
           [ 18,   6],
           [ 18,  94],
           [ 19,   3],
           [ 19,  72],
           [ 19,  14],
           [ 19,  99],
           [ 20,  15],
           [ 20,  77],
           [ 20,  13],
           [ 20,  79],
           [ 21,  35],
           [ 21,  66],
           [ 23,  29],
           [ 23,  98],
           [ 24,  35],
           [ 24,  73],
           [ 25,   5],
           [ 25,  73],
           [ 28,  14],
           [ 28,  82],
           [ 28,  32],
           [ 28,  61],
           [ 29,  31],
           [ 29,  87],
           [ 30,   4],
           [ 30,  73],
           [ 33,   4],
           [ 33,  92],
           [ 33,  14],
           [ 33,  81],
           [ 34,  17],
           [ 34,  73],
           [ 37,  26],
           [ 37,  75],
           [ 38,  35],
           [ 38,  92],
           [ 39,  36],
           [ 39,  61],
           [ 39,  28],
           [ 39,  65],
           [ 40,  55],
           [ 40,  47],
           [ 40,  42],
           [ 40,  42],
           [ 42,  52],
           [ 42,  60],
           [ 43,  54],
           [ 43,  60],
           [ 43,  45],
           [ 43,  41],
           [ 44,  50],
           [ 44,  46],
```

```
# Plot the dendrogram with outliers
plt.figure(figsize=(10, 7))
plt.title("Customer Dendograms (with outliers)")
dend = shc.dendrogram(shc.linkage(data_with_outliers, method='ward'))

plt.show()
```

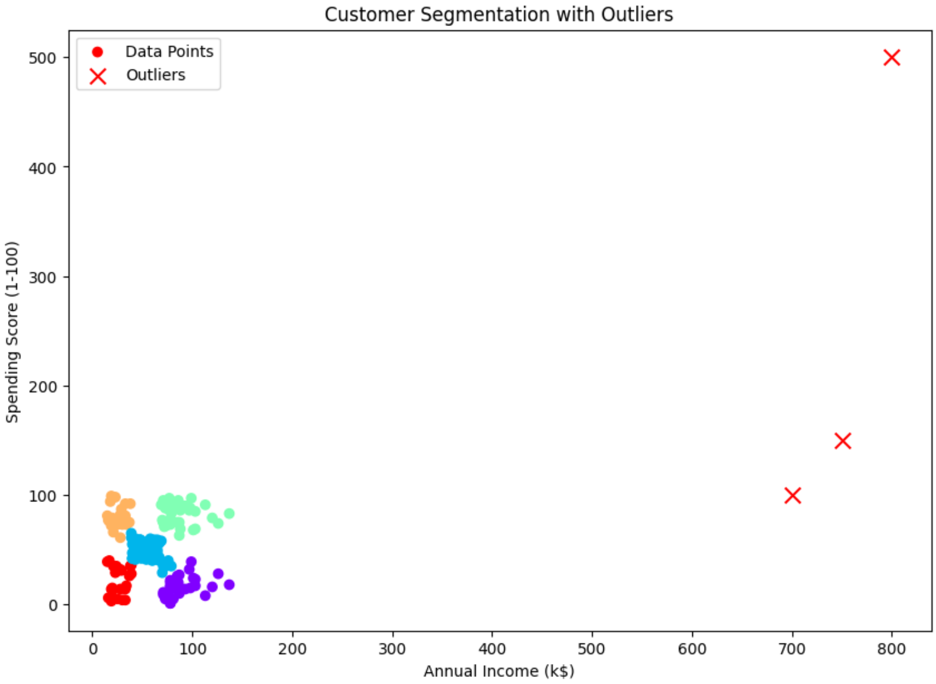


```
# Scatter plot with cluster labels (including outliers)
plt.figure(figsize=(10, 7))

# Plot the data points (excluding outliers)
plt.scatter(data[:, 0], data[:, 1], c=cluster.labels_[:data.shape[0]], cmap='rainbow', label='Data Points')

# Plot the outliers with a different marker and color
plt.scatter(data_with_outliers[-3:, 0], data_with_outliers[-3:, 1], c='red', marker='x', s=100, label='Outliers')

plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Customer Segmentation with Outliers')
plt.legend()
plt.show()
```



● ×