SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

20PAIE51J- MACHINE LEARNING (UNSUPERVISED MODEL)

Hiearchial Clustering

a. Import required Library (2 marks)

In [1]:
```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.cluster import AgglomerativeClustering
from sklearn.preprocessing import LabelEncoder
from scipy.cluster.hierarchy import cophenet, cut_tree, dendrogram, linkage
from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
from scipy.spatial.distance import pdist
```

b. Read the dataset (tab, csv, xls, txt, inbuilt dataset). (1 mark)

In [2]:
```
data = pd.read_csv('MPA-1_forestfires.csv')
data.head()
```

Out[2]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

c. Perform explanotory data analysis on the dataset. (3 marks)

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   X       517 non-null    int64  
 1   Y       517 non-null    int64  
 2   month   517 non-null    object 
 3   day     517 non-null    object 
 4   FFMC    517 non-null    float64
 5   DMC     517 non-null    float64
 6   DC      517 non-null    float64
 7   ISI     517 non-null    float64
 8   temp    517 non-null    float64
 9   RH      517 non-null    int64  
 10  wind    517 non-null    float64
 11  rain    517 non-null    float64
 12  area    517 non-null    float64
dtypes: float64(8), int64(3), object(2)
memory usage: 52.6+ KB
```

In [4]:
```
data.nunique()
```

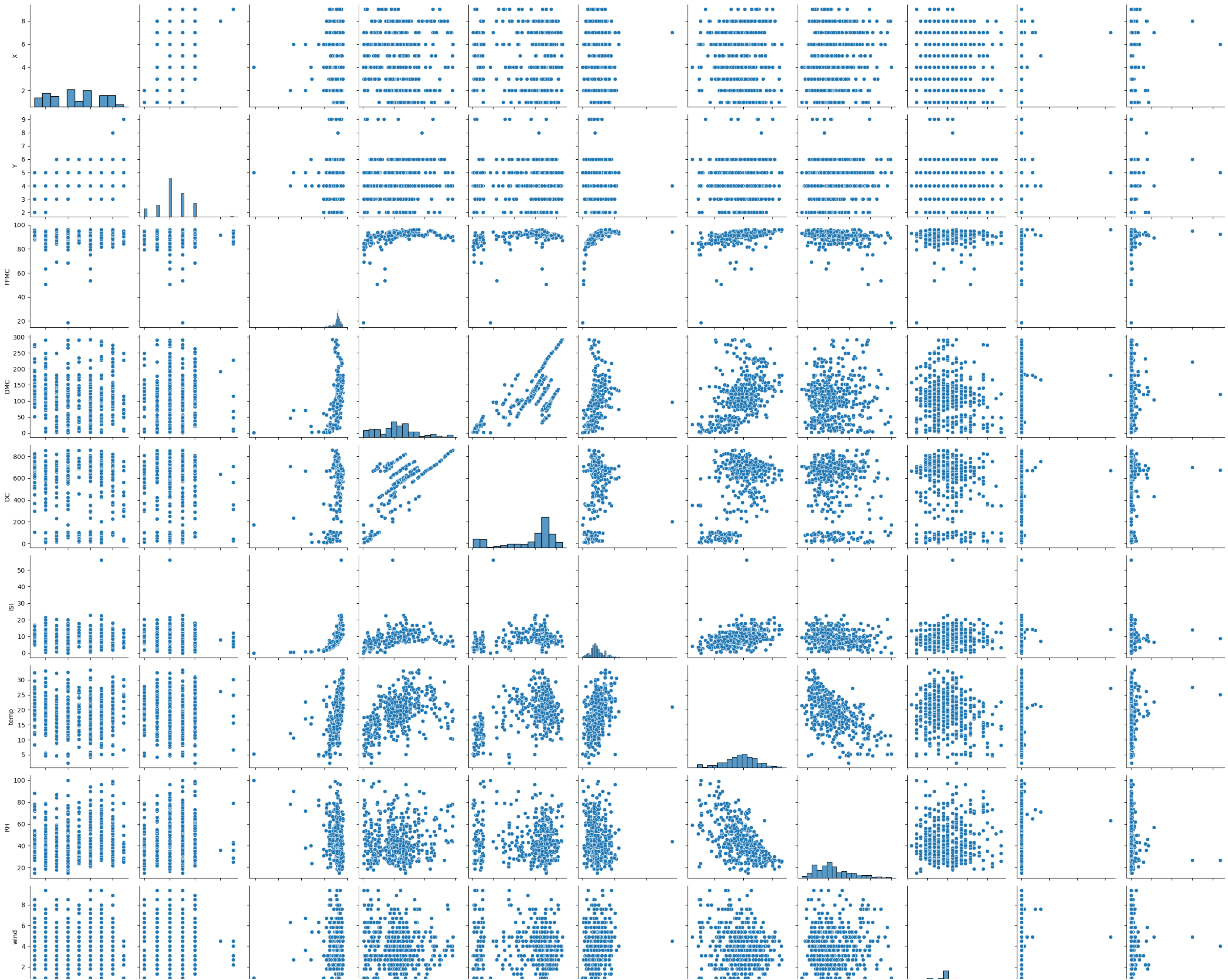
Out[4]:
```
X          9
Y          7
month     12
day        7
FFMC     106
DMC      215
DC       219
ISI      119
temp     192
RH        75
wind      21
rain       7
area     251
dtype: int64
```

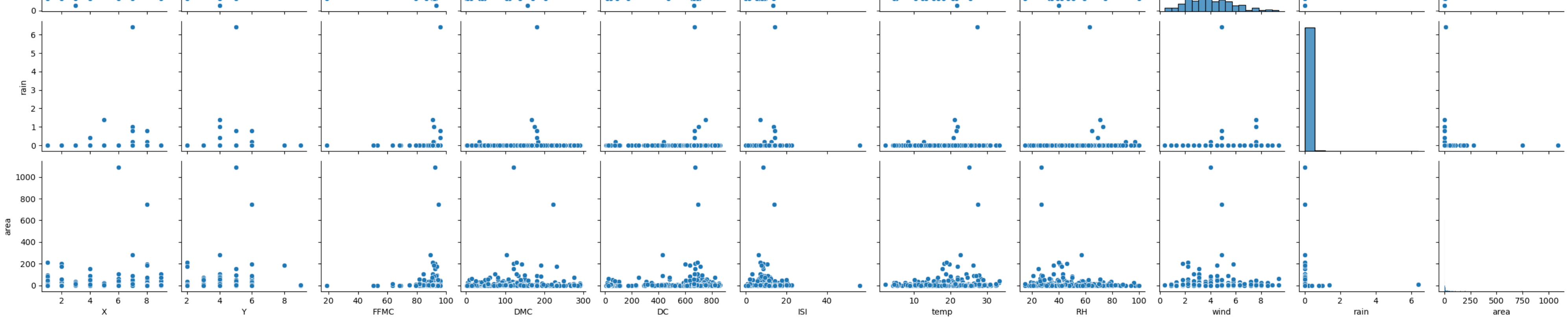
In [5]:
```
data.isnull().sum()
```

Out[5]:
```
X        0
Y        0
month    0
day      0
FFMC     0
DMC      0
DC       0
ISI      0
temp     0
RH       0
wind     0
rain     0
area     0
dtype: int64
```

d. Plot the datapoints using Scatter Plot. (3 marks)

In [6]:
```
sns.pairplot(data)
plt.show()
```


X
Y
FFMC
DMC
DC
ISI
temp
RH
wind



e. Apply five methods of agglomerative hierachial clustering. [Single, complete, average, centroid and ward's linkage method] (2 marks)

In [7]:
```
# applying Label encoder to modify two object type dimensions
le = LabelEncoder()
data['month'] = le.fit_transform(data['month'])
data['day'] = le.fit_transform(data['day'])
# applying Scaler
scaler =MinMaxScaler()
data_scaled = scaler.fit_transform(data)
pd.DataFrame(data_scaled)
```

Out[7]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.750	0.428571	0.636364	0.000000	0.870968	0.086492	0.101325	0.090909	0.192926	0.423529	0.700000	0.00000	0.000000
1	0.750	0.285714	0.909091	0.833333	0.927742	0.118194	0.775419	0.119430	0.508039	0.211765	0.055556	0.00000	0.000000
2	0.750	0.285714	0.909091	0.333333	0.927742	0.146795	0.796294	0.119430	0.398714	0.211765	0.100000	0.00000	0.000000
3	0.875	0.571429	0.636364	0.000000	0.941935	0.110958	0.081623	0.160428	0.196141	0.964706	0.400000	0.03125	0.000000
4	0.875	0.571429	0.636364	0.500000	0.910968	0.172984	0.110590	0.171123	0.295820	0.988235	0.155556	0.00000	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	0.375	0.142857	0.090909	0.500000	0.811613	0.191592	0.771315	0.033868	0.823151	0.200000	0.255556	0.00000	0.005904
513	0.125	0.285714	0.090909	0.500000	0.811613	0.191592	0.771315	0.033868	0.633441	0.658824	0.600000	0.00000	0.049769
514	0.750	0.285714	0.090909	0.500000	0.811613	0.191592	0.771315	0.033868	0.610932	0.647059	0.700000	0.00000	0.010231
515	0.000	0.285714	0.090909	0.333333	0.976774	0.499311	0.711622	0.201426	0.752412	0.317647	0.400000	0.00000	0.000000
516	0.625	0.142857	0.818182	0.833333	0.784516	0.006547	0.115867	0.019608	0.308682	0.188235	0.455556	0.00000	0.000000

517 rows × 13 columns

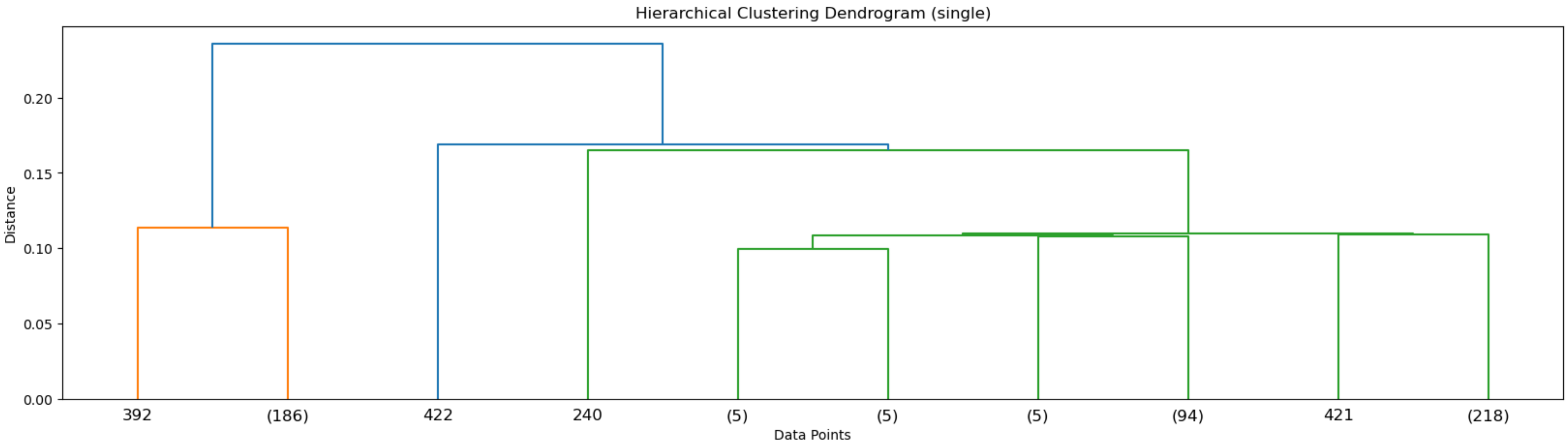
In [8]:
```
# applying PCA for dim reduction
pca = PCA(n_components=2)
X = pca.fit_transform(data_scaled)
X
```

Out[8]:
```
array([[-0.01566767,  0.90742016],
       [-0.34156282, -0.06303277],
       [-0.45518297,  0.14470228],
       ...,
       [ 0.42678376,  0.04385613],
       [ 0.32126669, -0.28562606],
       [-0.06096833,  0.51151893]])
```

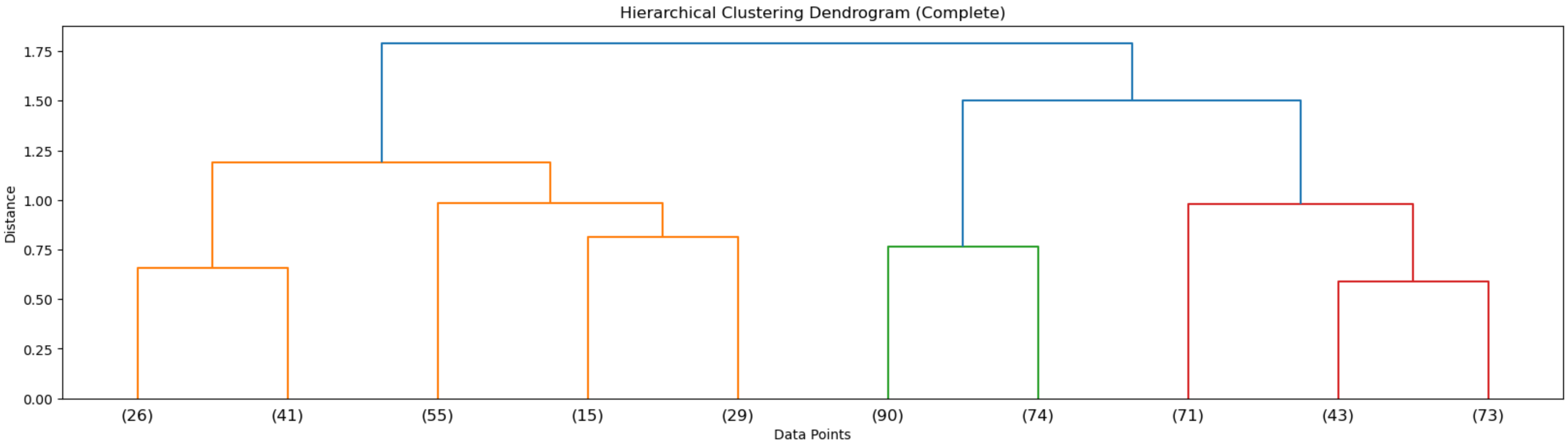
In [9]:
```python
# Perform clustering
single_labels = linkage(X, method='single', metric='euclidean')
complete_labels =linkage(X, method='complete', metric='euclidean')
average_labels = linkage(X, method='average', metric='euclidean')
ward_labels = linkage(X, method='ward', metric='euclidean')
centroid_labels = linkage(X, method='centroid', metric='euclidean')
```

f. Draw dendrogram for the above five clustering methods. (2 marks)

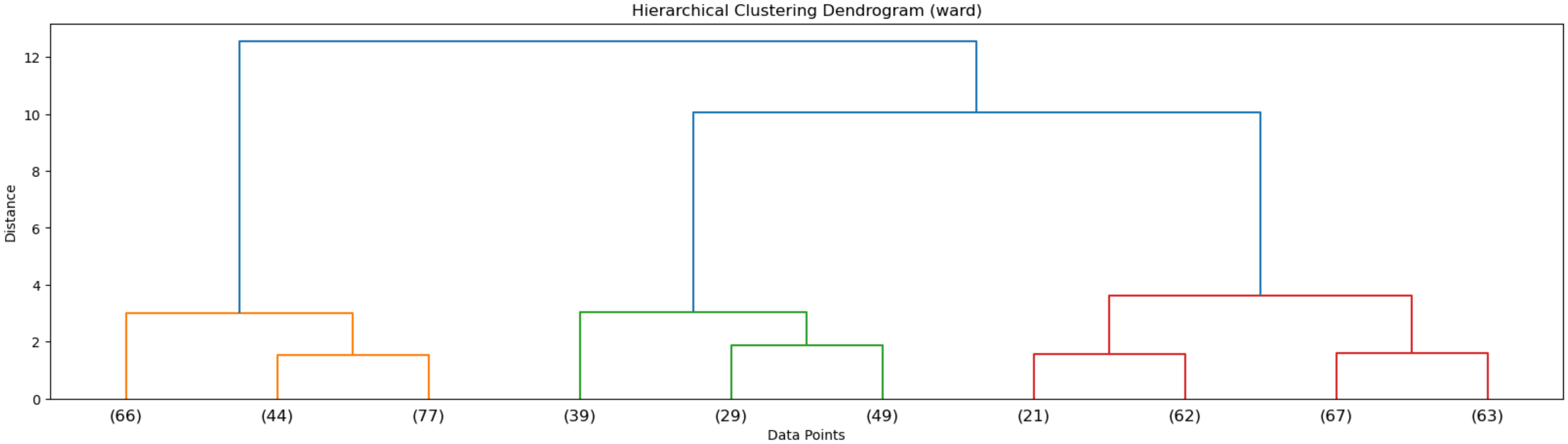
In [10]:
```python
plt.figure(figsize=(20, 5))
dendrogram(single_labels,p=10, truncate_mode='lastp')
plt.title('Hierarchical Clustering Dendrogram (single)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



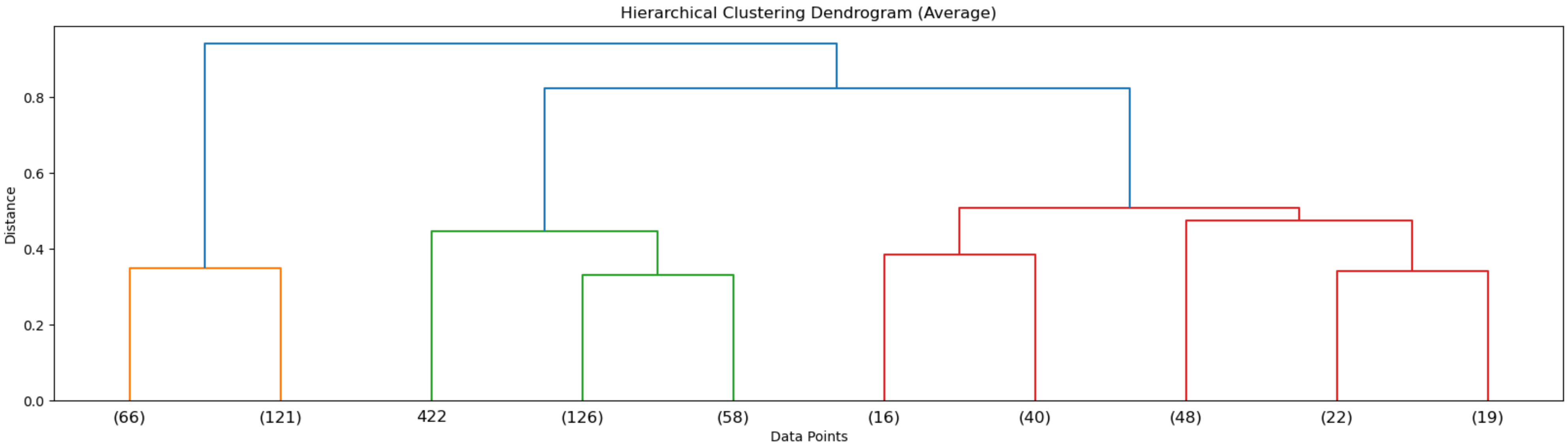
In [11]:
```
plt.figure(figsize=(20, 5))
dendrogram(complete_labels,p=10, truncate_mode='lastp')
plt.title('Hierarchical Clustering Dendrogram (Complete)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



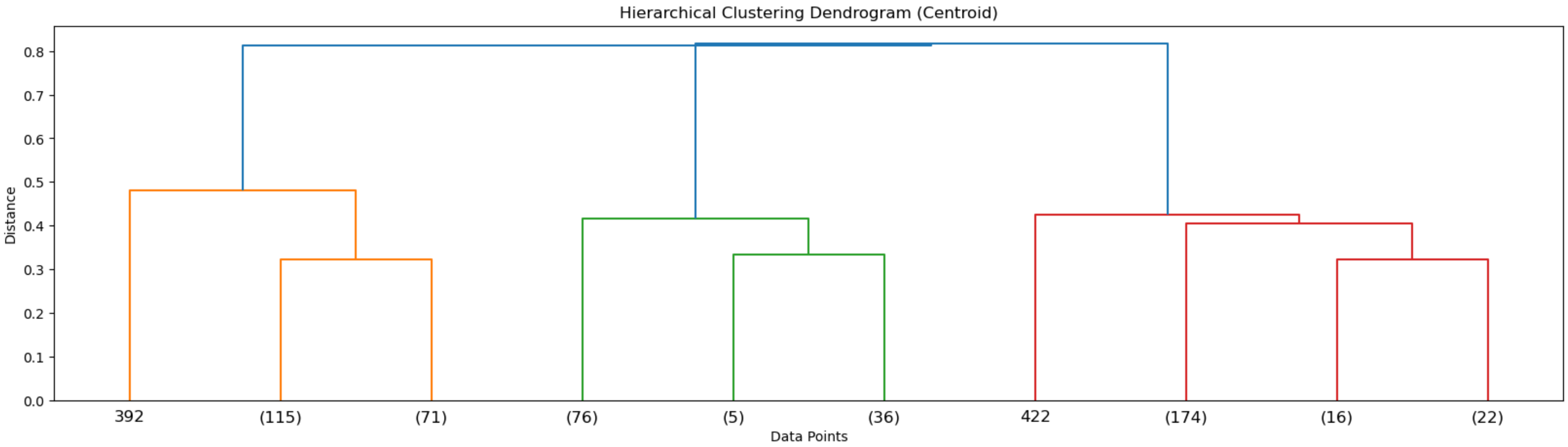
In [12]:
```
plt.figure(figsize=(20, 5))
dendrogram(ward_labels,p=10, truncate_mode='lastp')
plt.title('Hierarchical Clustering Dendrogram (ward)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



In [13]:
```
plt.figure(figsize=(20, 5))
dendrogram(average_labels,p=10, truncate_mode='lastp')
plt.title('Hierarchical Clustering Dendrogram (Average)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



In [14]:
```python
plt.figure(figsize=(20, 5))
dendrogram(centroid_labels,p=10, truncate_mode='lastp')
plt.title('Hierarchical Clustering Dendrogram (Centroid)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.show()
```



g. Calculate Cophenet Coorelation coefficient for the above five methods. (4 marks)

In [15]:
```python
# Calculate the pairwise distances between the data points
for i in [[single_labels, 'single'], [complete_labels, 'complete'], [centroid_labels, 'centroid'], [average_labels, 'ave
    cophenet_coeff, _ = cophenet(i[0], pdist(X))
    print("Cophenetic Correlation Coefficient for ",i[1]," Hierarchical Clustering :", cophenet_coeff)
```

```
Cophenetic Correlation Coefficient for  single  Hierarchical Clustering : 0.7504796651106578
Cophenetic Correlation Coefficient for  complete  Hierarchical Clustering : 0.7552694974754341
Cophenetic Correlation Coefficient for  centroid  Hierarchical Clustering : 0.8484844484458671
Cophenetic Correlation Coefficient for  average  Hierarchical Clustering : 0.8365892002059653
Cophenetic Correlation Coefficient for  ward  Hierarchical Clustering : 0.841992125501152
```

h. Plot the best method labels using the scatter plot. (3 marks)

In [18]:

```python
# Perform centriod linkage clustering
Z = linkage(X, method="average")
data["Cluster"] = pd.Series(cut_tree(Z, n_clusters=4).flatten())
data
```

Out[18]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	Cluster
0	7	5	7	0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	0
1	7	4	10	5	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	1
2	7	4	10	2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	1
3	8	6	7	0	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	0
4	8	6	7	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	4	3	1	3	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	3
513	2	4	1	3	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	3
514	7	4	1	3	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	2
515	1	4	1	2	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	3
516	6	3	9	5	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	0

517 rows × 14 columns

In [19]:
```
sns.scatterplot(data=data, x=X[:,0], y=X[:,1], hue=data['Cluster'])
plt.show()
```