In [53]:
```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

In [115…
```
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='Combine')
df1=df.drop(['Total'], axis=1)
df1=df1.iloc[:,2:16].values
#scaler = MinMaxScaler()
scaler= StandardScaler()
# transform data
df1 = scaler.fit_transform(df1)
```

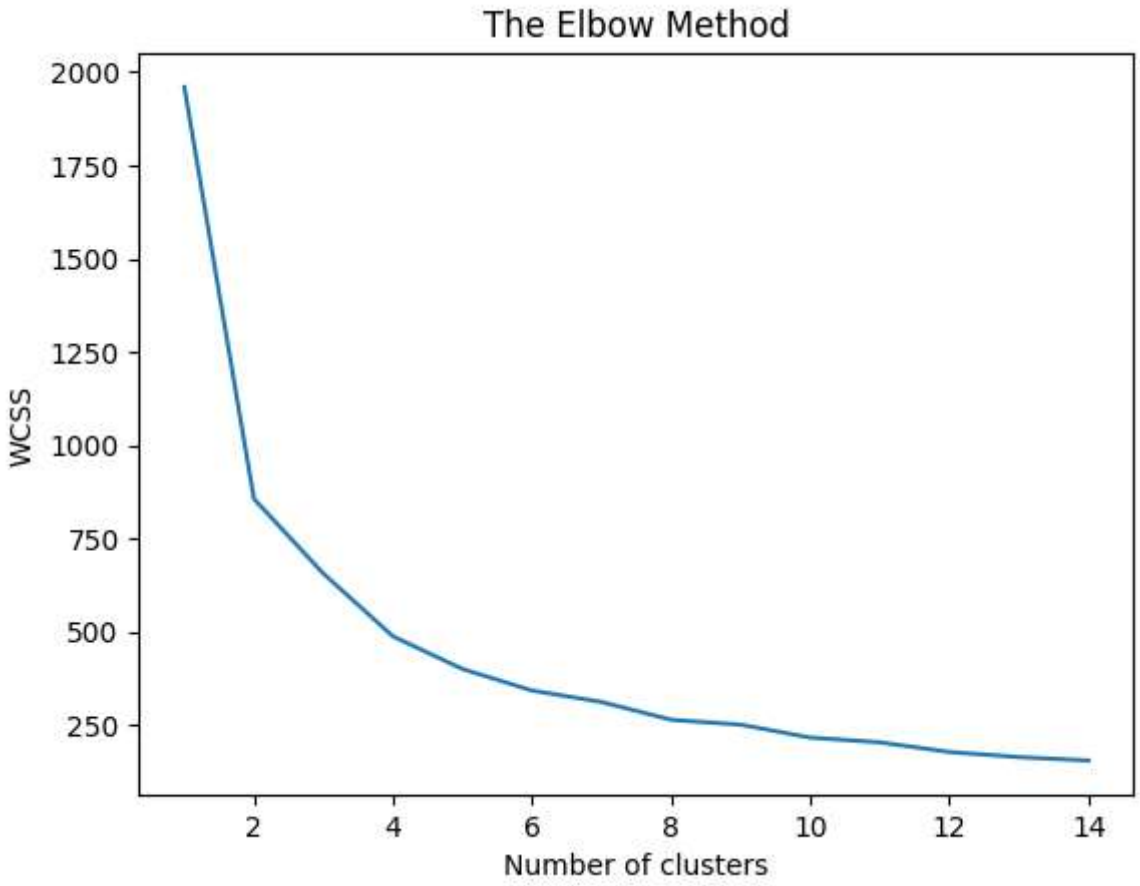
In [116…
```
from sklearn.cluster import KMeans

#create a list for the wcss parameter
wcss = []
#test with 14 clusters
for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state =0)
    kmeans.fit(df1)
    wcss.append(kmeans.inertia_)
```

In [117…
```
wcss
```

Out[117]:
```
[1959.9999999999998,
 855.6534496397571,
 655.1212184737583,
 486.91986933112577,
 399.38508550318085,
 341.62218823099954,
 310.9301481602454,
 263.20938587417163,
 250.17989379692776,
 215.44306674325688,
 202.77601585872424,
 176.8332191391253,
 163.2040640324577,
 154.02144254225445]
```

In [118…
```
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [119…
```
km=KMeans(n_clusters = 4, init = 'k-means++', random_state = 0)
y_kmeans=km.fit_predict(df1)
```

In [120…
```
y_kmeans
```

Out[120]:
```
array([3, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 3, 0, 0, 3, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 2, 3, 0, 0, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3,
       1, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3,
       0, 0, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 3, 0, 0, 3, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 0, 0, 3, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 3, 1, 0, 0, 0])
```

In [121…
```
df['cluster']=y_kmeans
df.head()
```

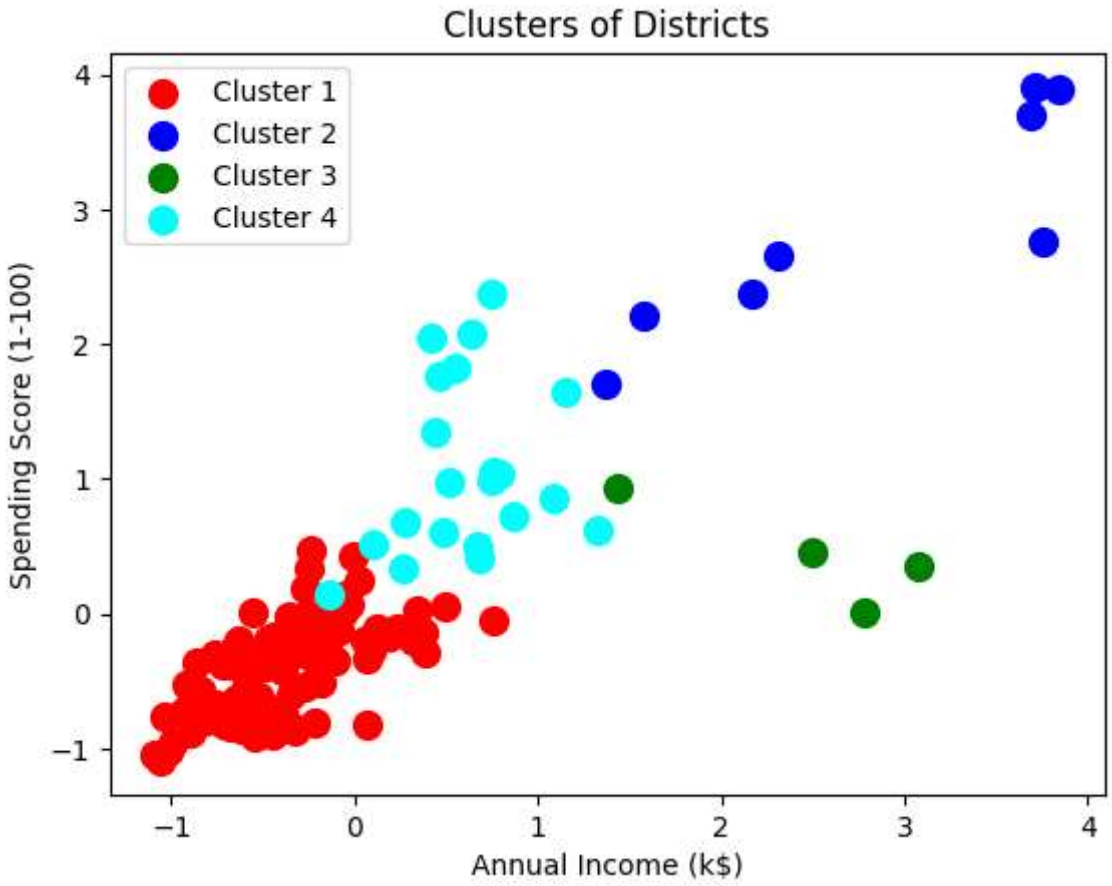
Out[121]:

	SrNo	State/UT/District	Homicide/Murder	Causing death by negligence	Hurt	Assault on woman	Kidnapping and abduction	Human trafficking	Ra
0	1	Ahmednagar	254	620	1826	543	369	0	
1	2	Akola	114	116	1345	245	76	0	
2	3	Amravati	201	354	2133	461	190	0	
3	4	Aurangbad	139	386	1869	335	242	0	
4	5	Beed	206	308	1233	248	91	0	

5 rows × 24 columns

In [122…

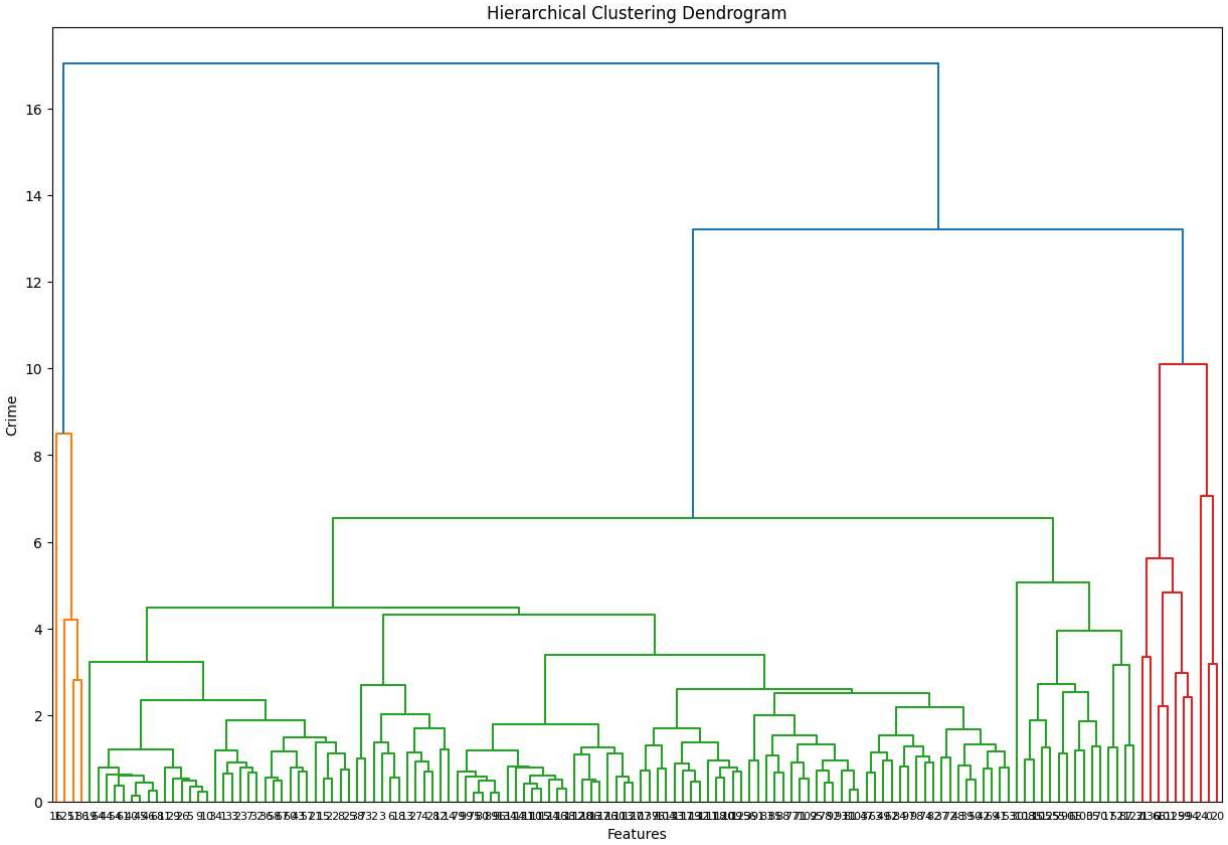
```
#plt.scatter(df['SrNo'],df['cluster'])
#for col in df.columns:
 #   print(col)
plt.scatter(df1[y_kmeans == 0, 0], df1[y_kmeans == 0, 1], s = 100, c = 'red', label =
plt.scatter(df1[y_kmeans == 1, 0], df1[y_kmeans == 1, 1], s = 100, c = 'blue', label =
plt.scatter(df1[y_kmeans == 2, 0], df1[y_kmeans == 2, 1], s = 100, c = 'green', label
plt.scatter(df1[y_kmeans == 3, 0], df1[y_kmeans == 3, 1], s = 100, c = 'cyan', label =
#plt.scatter(df1[y_kmeans == 4, 0], df1[y_kmeans == 4, 1], s = 100, c = 'magenta', lab
#plt.scatter(df1[y_kmeans == 5, 0], df1[y_kmeans == 5, 1], s = 100, c = 'black', label
#plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c
plt.title('Clusters of Districts')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



In [123…
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
```

In [124…
```
z = linkage(df1, method="complete",metric="euclidean")
```

In [125…
```
plt.figure(figsize=(15, 10))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Features')
plt.ylabel('Crime')
sch.dendrogram(z,
    leaf_rotation=0.,  # rotates the x axis labels
    leaf_font_size=8.,  # font size for the x axis labels
)
plt.show()
```


Hierarchical Clustering Dendrogram
Crime
Features
0
2
4
6
8
10
12
14
16