In [140…

```
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import matplotlib
```

In [141…

```
df = pd.read_csv(r"D:\PG-DAI\Machine-Learning\Jan 5 DBSCAN\Mall_Customers.csv")
```

In [142…

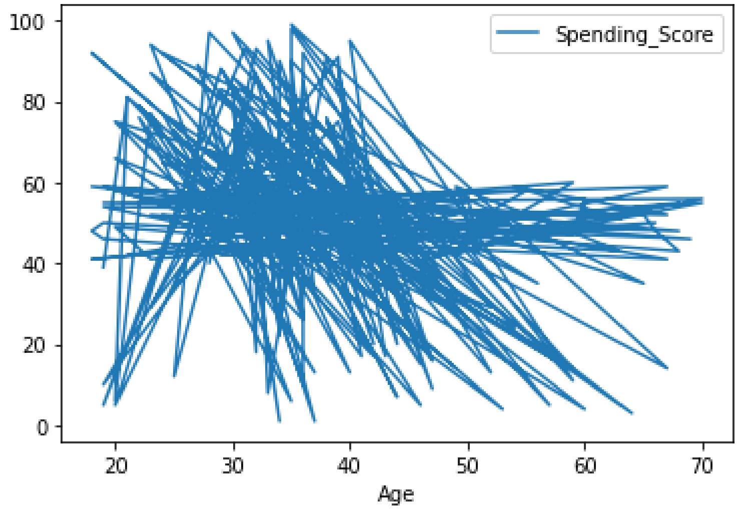
```
del df['CustomerID']
del df['Genre']
```

In [143…

```
df.plot("Age","Spending_Score")
```

Out[143…

```
<AxesSubplot:xlabel='Age'>
```



In [144…

```
del df["Age"]
```

In [145…

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
#'Fit' and transform the train set; and transform using the fit on the test set later
```

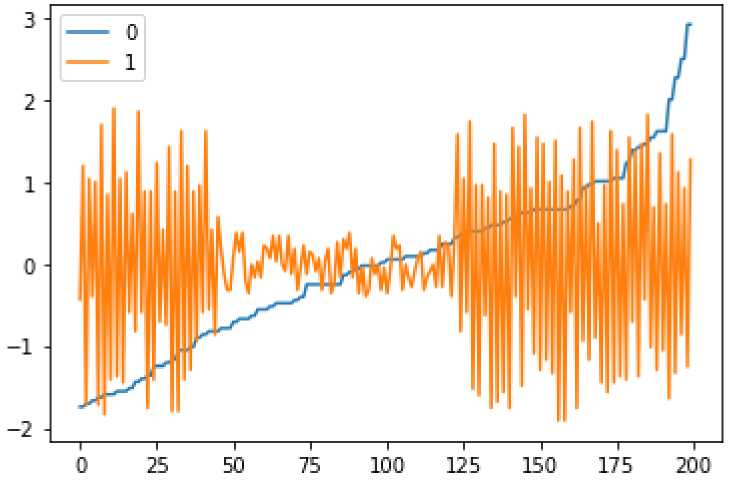
```
scaler = StandardScaler()
df = scaler.fit_transform(df)
df = pd.DataFrame(df)
```

In [146…

```
df.plot()
```

Out[146…

```
<AxesSubplot:>
```



In [147…

```
from sklearn.cluster import KMeans
k_means=KMeans(n_clusters=3,random_state=42)
k_means.fit(df)
```

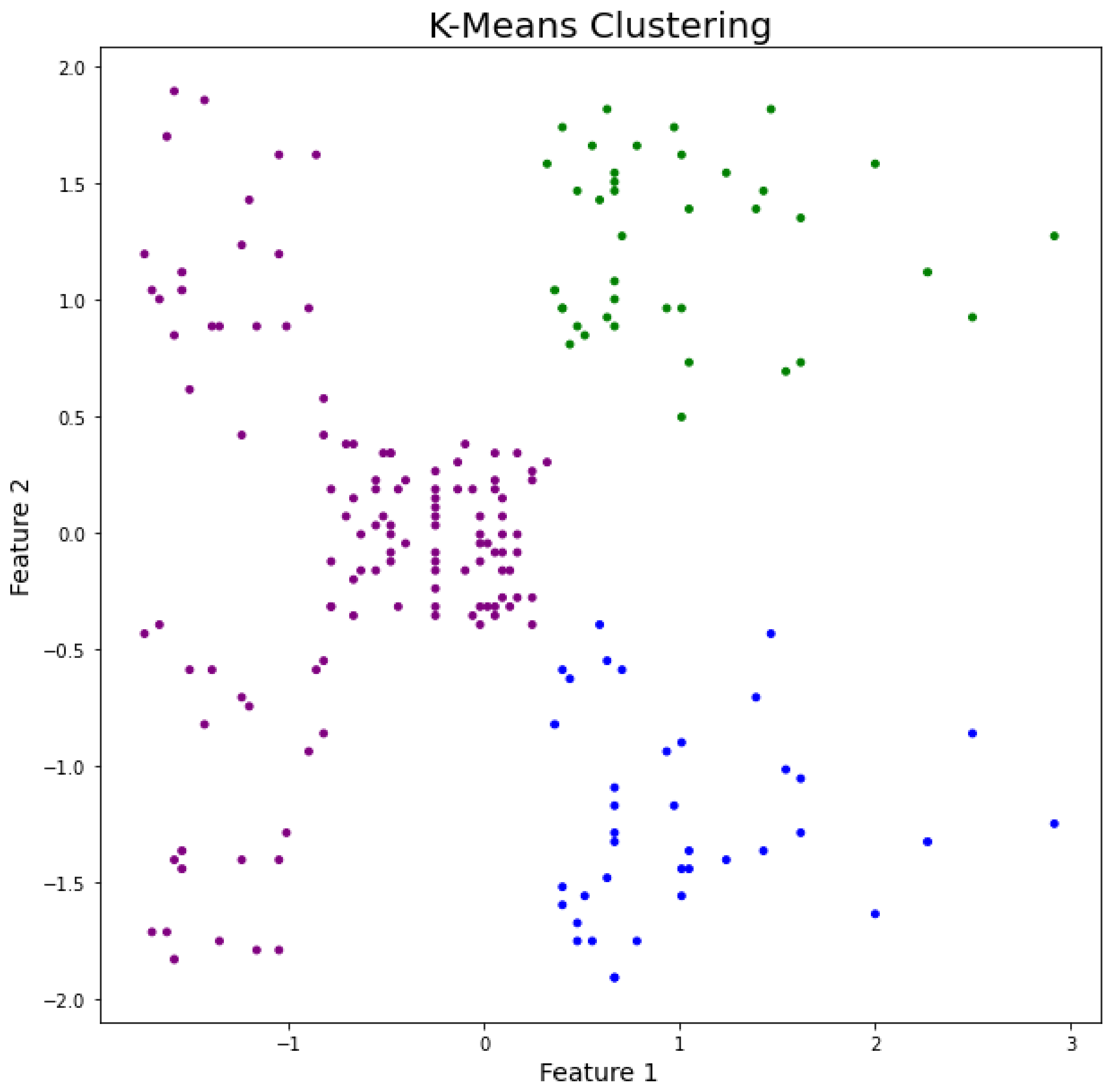
Out[147…

```
KMeans(n_clusters=3, random_state=42)
```

In [160…

```
df['KMeans_labels']=k_means.labels_

# Plotting resulting clusters
colors=['purple','red','blue','green']
plt.figure(figsize=(10,10))
plt.scatter(df.iloc[:,0:1],df.iloc[:,1:2],c=df['KMeans_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('K-Means Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



In [149…

```
from sklearn.cluster import AgglomerativeClustering
```

```
model = AgglomerativeClustering(n_clusters=4, affinity='euclidean')
model.fit(df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:1675: FutureWarning: Feature names only support names that
are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

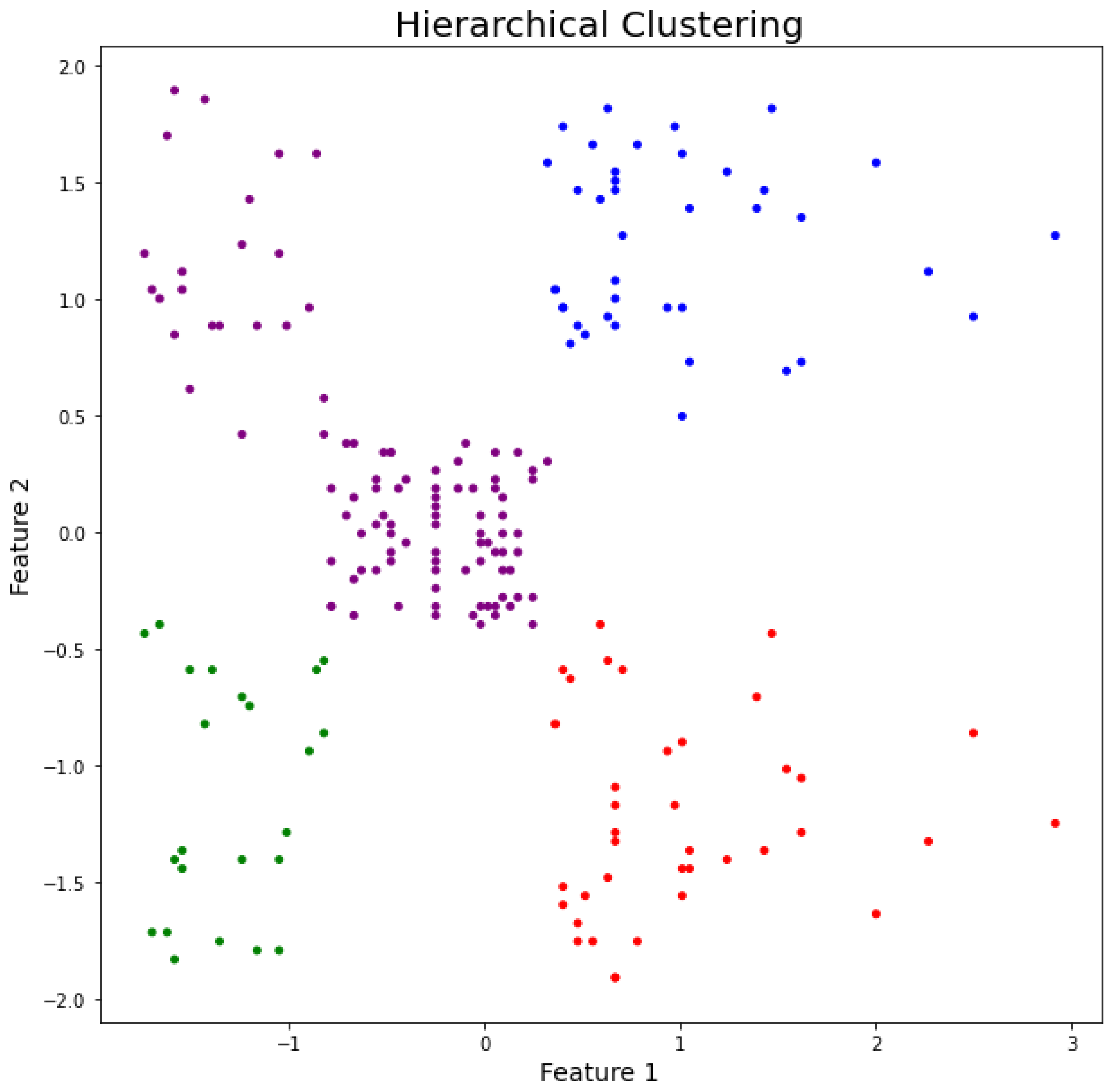
Out[149…

```
AgglomerativeClustering(n_clusters=4)
```

In [166…

```
df['HR_labels']=model.labels_

# Plotting resulting clusters
plt.figure(figsize=(10,10))
plt.scatter(df.iloc[:,0:1],df.iloc[:,1:2],c=df['HR_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('Hierarchical Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



In [151…

```
from sklearn.cluster import DBSCAN
dbscan=DBSCAN()
dbscan.fit(df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:1675: FutureWarning: Feature names only support names that
are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

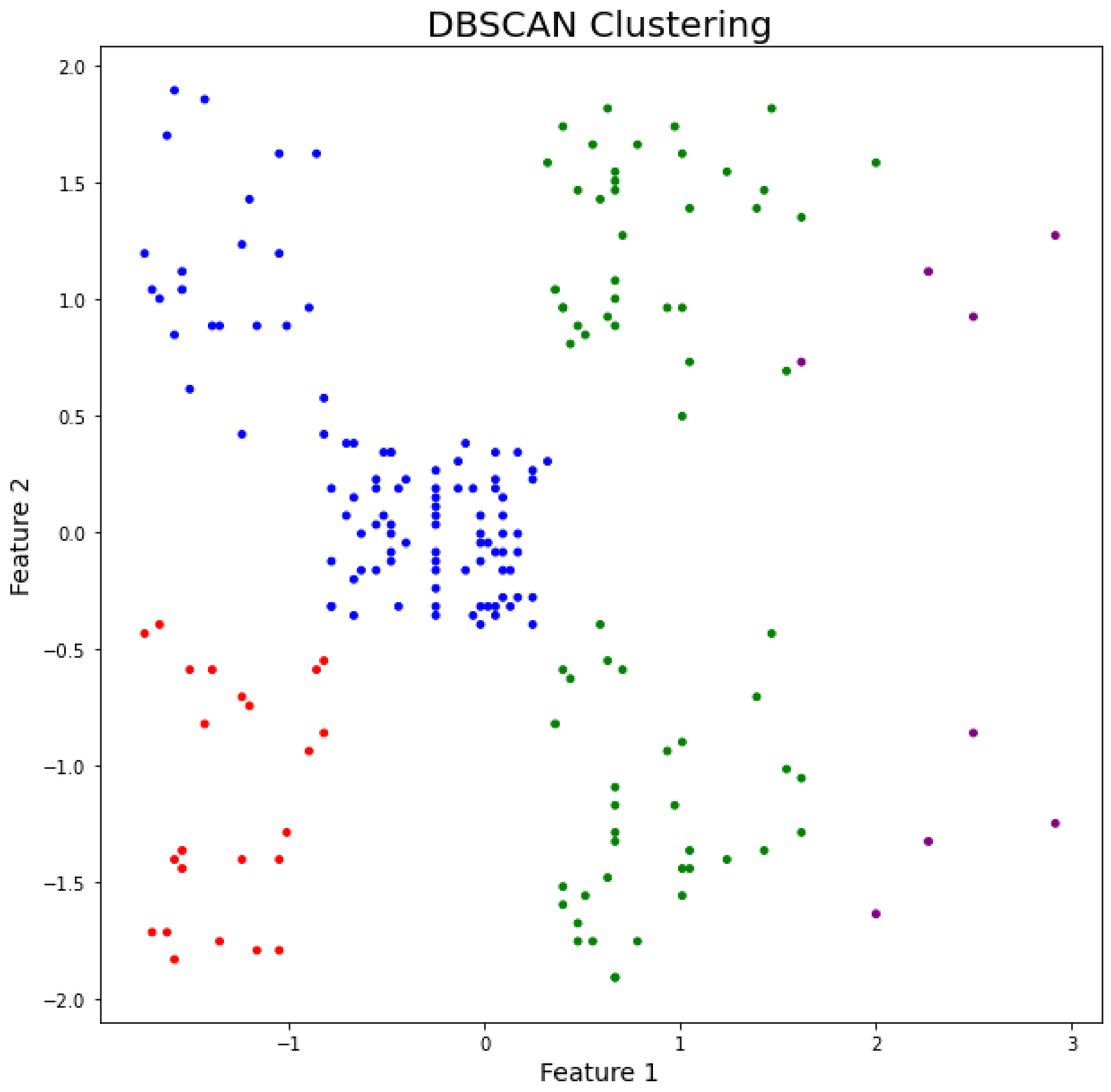
Out[151…

```
DBSCAN()
```

In [161…

```
df['DBSCAN_labels']=dbscan.labels_

# Plotting resulting clusters
plt.figure(figsize=(10,10))
plt.scatter(df.iloc[:,0:1],df.iloc[:,1:2],c=df['DBSCAN_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('DBSCAN Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



In [153…

```
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(df)
distances, indices = nbrs.kneighbors(df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:1675: FutureWarning: Feature names only support names that
are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:1675: FutureWarning: Feature names only support names that
are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

In [154…

```
#The distance variable contains an array of distances between
#a data point and its nearest data point for all data points in the dataset.
distances
```

Out[154…

```
array([[0.        , 0.08564307],
       [0.        , 0.15990848],
       [0.        , 0.07633886],
       [0.        , 0.0544428 ],
       [0.        , 0.08564307],
       [0.        , 0.0544428 ],
       [0.        , 0.07633886],
       [0.        , 0.19782504],
       [0.        , 0.12255989],
       [0.        , 0.17303595],
       [0.        , 0.0544428 ],
       [0.        , 0.15753602],
       [0.        , 0.0544428 ],
       [0.        , 0.07764312],
       [0.        , 0.0544428 ],
       [0.        , 0.07764312],
       [0.        , 0.11450829],
       [0.        , 0.24511979],
       [0.        , 0.22357696],
       [0.        , 0.15753602],
       [0.        , 0.11450829],
       [0.        , 0.03816943],
       [0.        , 0.19475561],
       [0.        , 0.03816943],
       [0.        , 0.19084715],
       [0.        , 0.19475561],
       [0.        , 0.0544428 ],
       [0.        , 0.33025172],
       [0.        , 0.0544428 ],
       [0.        , 0.19782504],
       [0.        , 0.11450829],
       [0.        , 0.15267772],
       [0.        , 0.11450829],
       [0.        , 0.19084715],
       [0.        , 0.12255989],
       [0.        , 0.19475561],
       [0.        , 0.12255989],
       [0.        , 0.13834957],
       [0.        , 0.10888561],
       [0.        , 0.13834957],
       [0.        , 0.0544428 ],
       [0.        , 0.19084715],
       [0.        , 0.0544428 ],
       [0.        , 0.12091014],
       [0.        , 0.10888561],
       [0.        , 0.15528624],
       [0.        , 0.12091014],
       [0.        , 0.13834957],
       [0.        , 0.        ],
       [0.        , 0.        ],
       [0.        , 0.08651797],
       [0.        , 0.03816943],
       [0.        , 0.08651797],
       [0.        , 0.03816943],
       [0.        , 0.0544428 ],
       [0.        , 0.12091014],
       [0.        , 0.08564307],
       [0.        , 0.0544428 ],
       [0.        , 0.0544428 ],
       [0.        , 0.07633886],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.0544428 ],
       [0.        , 0.03816943],
       [0.        , 0.03882156],
       [0.        , 0.        ],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.        ],
       [0.        , 0.03882156],
       [0.        , 0.0544428 ],
       [0.        , 0.19084715],
       [0.        , 0.08564307],
       [0.        , 0.0544428 ],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.07764312],
       [0.        , 0.07764312],
       [0.        , 0.03882156],
       [0.        , 0.08651797],
       [0.        , 0.07633886],
       [0.        , 0.08651797],
       [0.        , 0.08564307],
       [0.        , 0.07633886],
       [0.        , 0.0544428 ],
       [0.        , 0.03816943],
       [0.        , 0.0544428 ],
       [0.        , 0.03816943],
       [0.        , 0.07764312],
       [0.        , 0.07764312],
       [0.        , 0.03882156],
       [0.        , 0.03816943],
       [0.        , 0.03816943],
       [0.        , 0.03882156],
       [0.        , 0.03816943],
       [0.        , 0.11450829],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03816943],
       [0.        , 0.07633886],
       [0.        , 0.03816943],
       [0.        , 0.0544428 ],
       [0.        , 0.03816943],
       [0.        , 0.07764312],
       [0.        , 0.0544428 ],
       [0.        , 0.0544428 ],
       [0.        , 0.03816943],
       [0.        , 0.07633886],
       [0.        , 0.07633886],
       [0.        , 0.0544428 ],
       [0.        , 0.10888561],
       [0.        , 0.07633886],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.11646468],
       [0.        , 0.08564307],
       [0.        , 0.17303595],
       [0.        , 0.20857963],
       [0.        , 0.08651797],
       [0.        , 0.0544428 ],
       [0.        , 0.17128613],
       [0.        , 0.07764312],
       [0.        , 0.        ],
       [0.        , 0.07764312],
       [0.        , 0.        ],
       [0.        , 0.0544428 ],
       [0.        , 0.08564307],
       [0.        , 0.07633886],
       [0.        , 0.12091014],
       [0.        , 0.07764312],
       [0.        , 0.0544428 ],
       [0.        , 0.12091014],
       [0.        , 0.0544428 ],
       [0.        , 0.07633886],
       [0.        , 0.16332841],
       [0.        , 0.15990848],
       [0.        , 0.08564307],
       [0.        , 0.13834957],
       [0.        , 0.17303595],
       [0.        , 0.08564307],
       [0.        , 0.0544428 ],
       [0.        , 0.07764312],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.07764312],
       [0.        , 0.07764312],
       [0.        , 0.03882156],
       [0.        , 0.03882156],
       [0.        , 0.        ],
       [0.        , 0.07764312],
       [0.        , 0.        ],
       [0.        , 0.0544428 ],
       [0.        , 0.08564307],
       [0.        , 0.19294031],
       [0.        , 0.19294031],
       [0.        , 0.16332841],
       [0.        , 0.08564307],
       [0.        , 0.07633886],
       [0.        , 0.20857963],
       [0.        , 0.12255989],
       [0.        , 0.08564307],
       [0.        , 0.236036  ],
       [0.        , 0.03816943],
       [0.        , 0.07633886],
       [0.        , 0.11646468],
       [0.        , 0.12255989],
       [0.        , 0.03816943],
       [0.        , 0.236036  ],
       [0.        , 0.07764312],
       [0.        , 0.236036  ],
       [0.        , 0.19475561],
       [0.        , 0.20603662],
       [0.        , 0.28226971],
       [0.        , 0.08651797],
       [0.        , 0.19475561],
       [0.        , 0.08651797],
       [0.        , 0.28226971],
       [0.        , 0.35147277],
       [0.        , 0.08564307],
       [0.        , 0.49771891],
       [0.        , 0.20603662],
       [0.        , 0.22357696],
       [0.        , 0.08564307],
       [0.        , 0.75614615],
       [0.        , 0.40968723],
       [0.        , 0.44715391],
       [0.        , 0.40968723],
       [0.        , 0.30021064],
       [0.        , 0.51910784],
       [0.        , 0.30021064],
       [0.        , 0.57183643],
       [0.        , 0.54622499]])
```

In [155…

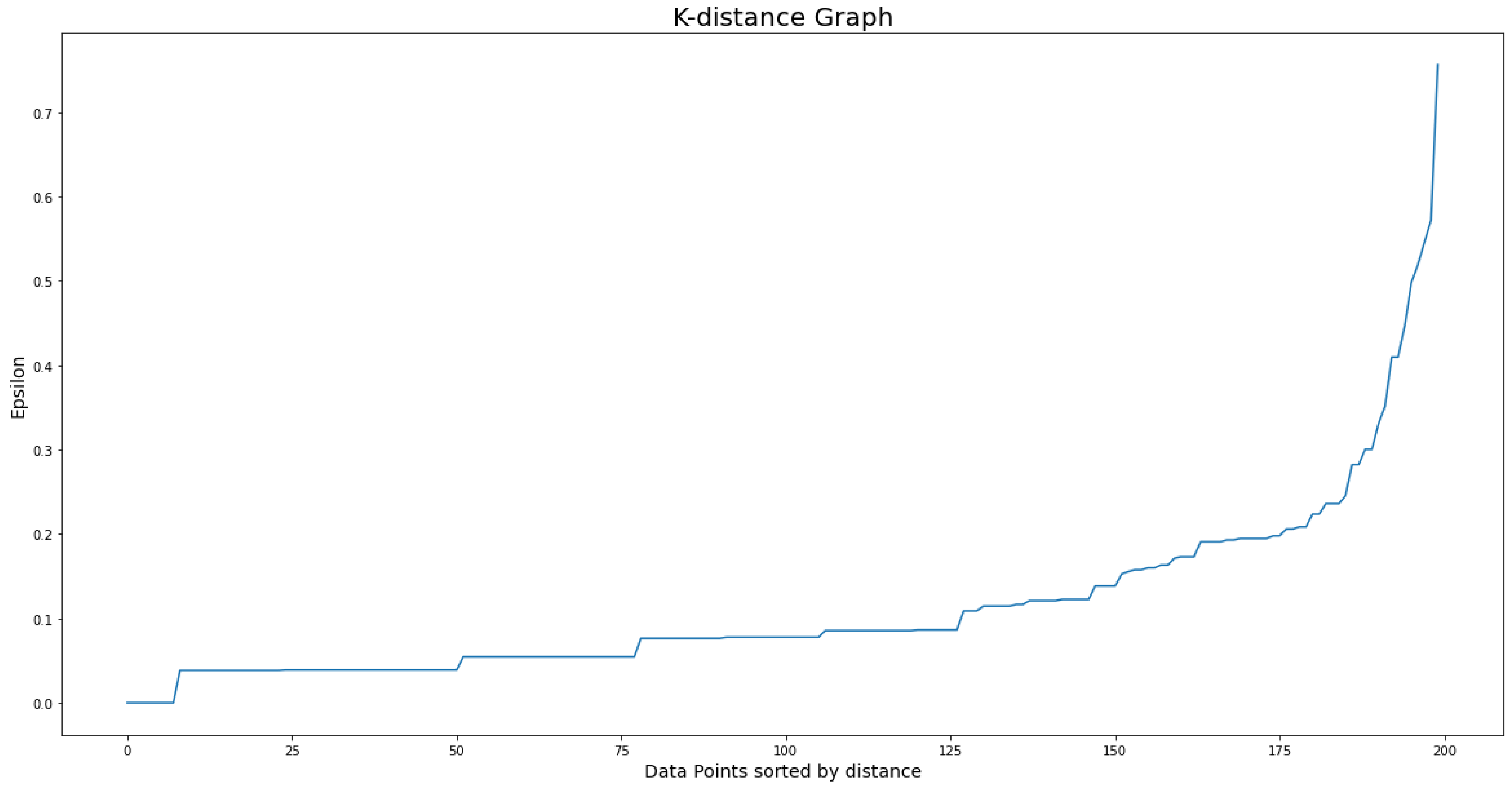
```
indices
```

Out[155…

```
array([[  0,   4],
       [  1,   3],
       [  2,   6],
       [  3,   5],
       [  4,   0],
       [  5,   3],
       [  6,   2],
       [  7,  11],
       [  8,   6],
       [  9,   5],
       [ 10,  14],
       [ 11,  19],
       [ 12,  10],
       [ 13,  15],
       [ 14,  10],
       [ 15,  13],
       [ 16,  20],
       [ 17,   9],
       [ 18,  26],
       [ 19,  11],
       [ 20,  16],
       [ 21,  23],
       [ 22,  30],
       [ 23,  21],
       [ 24,  34],
       [ 25,  35],
       [ 26,  28],
       [ 27,  17],
       [ 28,  26],
       [ 29,  25],
       [ 30,  32],
       [ 31,  37],
       [ 32,  30],
       [ 33,  41],
       [ 34,  36],
       [ 35,  25],
       [ 36,  34],
       [ 37,  39],
       [ 38,  44],
       [ 39,  37],
       [ 40,  42],
       [ 41,  33],
       [ 42,  40],
       [ 43,  51],
       [ 44,  38],
       [ 45,  43],
       [ 46,  52],
       [ 47,  54],
       [ 49,  48],
       [ 49,  48],
       [ 50,  52],
       [ 51,  53],
       [ 52,  50],
       [ 53,  51],
       [ 54,  57],
       [ 55,  48],
       [ 56,  58],
       [ 57,  54],
       [ 58,  62],
       [ 59,  57],
       [ 60,  61],
       [ 61,  60],
       [ 62,  64],
       [ 63,  65],
       [ 64,  66],
       [ 68,  65],
       [ 66,  64],
       [ 67,  69],
       [ 68,  65],
       [ 69,  67],
       [ 70,  73],
       [ 71,  79],
       [ 72,  67],
       [ 73,  70],
       [ 74,  85],
       [ 75,  76],
       [ 76,  78],
       [ 77,  74],
       [ 78,  80],
       [ 79,  82],
       [ 80,  78],
       [ 81,  75],
       [ 82,  79],
       [ 83,  79],
       [ 84,  81],
       [ 85,  74],
       [ 86,  88],
       [ 87,  90],
       [ 88,  86],
       [ 89,  96],
       [ 90,  87],
       [ 91,  93],
       [ 92,  99],
       [ 93,  91],
       [ 94,  98],
       [ 95,  97],
       [ 96,  92],
       [ 97,  92],
       [ 98,  94],
       [ 99,  92],
       [100, 105],
       [101, 109],
       [102, 117],
       [103, 104],
       [104, 103],
       [105,  98],
       [106, 115],
       [107, 113],
       [108, 105],
       [109, 101],
       [110, 106],
       [111, 103],
       [112, 116],
       [113, 107],
       [114, 109],
       [115, 106],
       [116, 112],
       [117, 119],
       [118, 116],
       [119, 120],
       [120, 119],
       [121, 118],
       [122, 119],
       [123, 127],
       [124, 132],
       [125, 129],
       [126, 132],
       [127, 141],
       [128, 130],
       [131, 129],
       [130, 128],
       [131, 129],
       [132, 126],
       [133, 139],
       [134, 140],
       [135, 143],
       [136, 134],
       [137, 139],
       [138, 130],
       [139, 137],
       [140, 134],
       [141, 149],
       [142, 146],
       [143, 151],
       [144, 138],
       [145, 141],
       [146, 160],
       [147, 159],
       [148, 152],
       [149, 155],
       [150, 154],
       [151, 155],
       [152, 148],
       [153, 157],
       [154, 150],
       [155, 151],
       [156, 158],
       [157, 153],
       [156, 158],
       [159, 147],
       [160, 146],
       [161, 143],
       [162, 158],
       [163, 149],
       [164, 168],
       [165, 171],
       [166, 176],
       [167, 173],
       [168, 164],
       [169, 177],
       [170, 174],
       [171, 165],
       [172, 170],
       [173, 167],
       [174, 170],
       [175, 173],
       [176, 174],
       [177, 171],
       [178, 174],
       [179, 183],
       [180, 184],
       [181, 183],
       [182, 178],
       [183, 181],
       [184, 180],
       [185, 183],
       [186, 190],
       [187, 177],
       [188, 182],
       [189, 183],
       [190, 186],
       [191, 195],
       [192, 194],
       [193, 189],
       [194, 192],
       [195, 197],
       [196, 194],
       [197, 195],
       [198, 196],
       [199, 197]], dtype=int64)
```

In [156…

```
# Plotting K-distance Graph
distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.figure(figsize=(20,10))
plt.plot(distances)
plt.title('K-distance Graph',fontsize=20)
plt.xlabel('Data Points sorted by distance',fontsize=14)
plt.ylabel('Epsilon',fontsize=14)
plt.show()
```



The optimum value of epsilon is at the point of maximum curvature in the K-Distance Graph, which is 30 in this case. Now, it's time to find the value of minPoints. The value of minPoints also depends on domain knowledge. This time I am taking minPoints as 6:

In [163…

```
from sklearn.cluster import DBSCAN
dbscan_opt=DBSCAN(eps=0.2,min_samples=5)
dbscan_opt.fit(df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:1675: FutureWarning: Feature names only support names that
are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
DBSCAN(eps=0.2)
```

Out[163…

In [164…

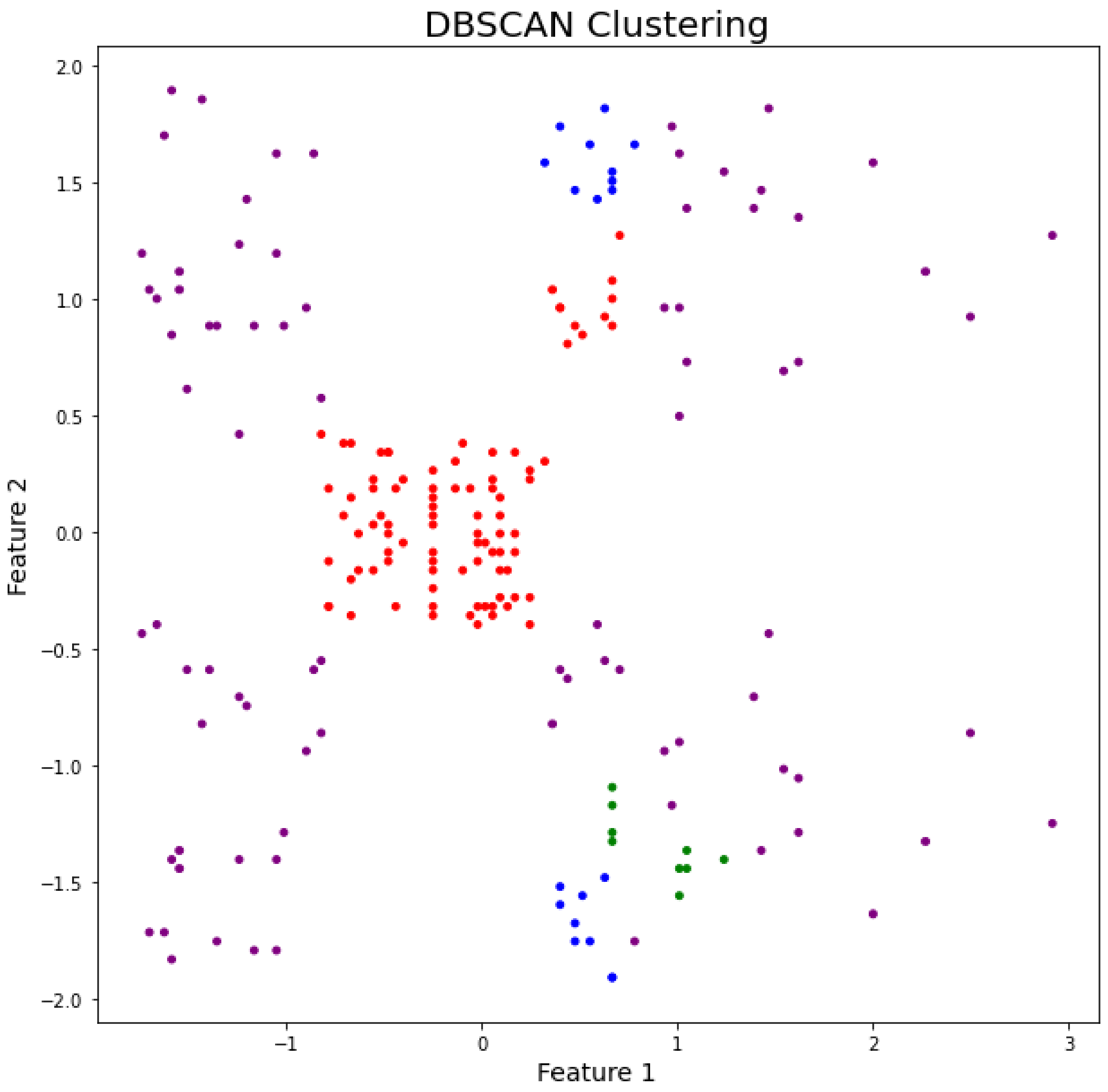
```
df['DBSCAN_opt_labels']=dbscan_opt.labels_
df['DBSCAN_opt_labels'].value_counts()
```

Out[164…

```
 1    78
-1    77
 2    11
 4    10
 3     9
 0     6
 6     5
 5     4
Name: DBSCAN_opt_labels, dtype: int64
```

In [165…

```
# Plotting the resulting clusters
plt.figure(figsize=(10,10))
plt.scatter(df.iloc[:,0:1],df.iloc[:,1:2],c=df['DBSCAN_opt_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('DBSCAN Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```


DBSCAN Clustering
Feature 2
Feature 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1
0
1
2
3