

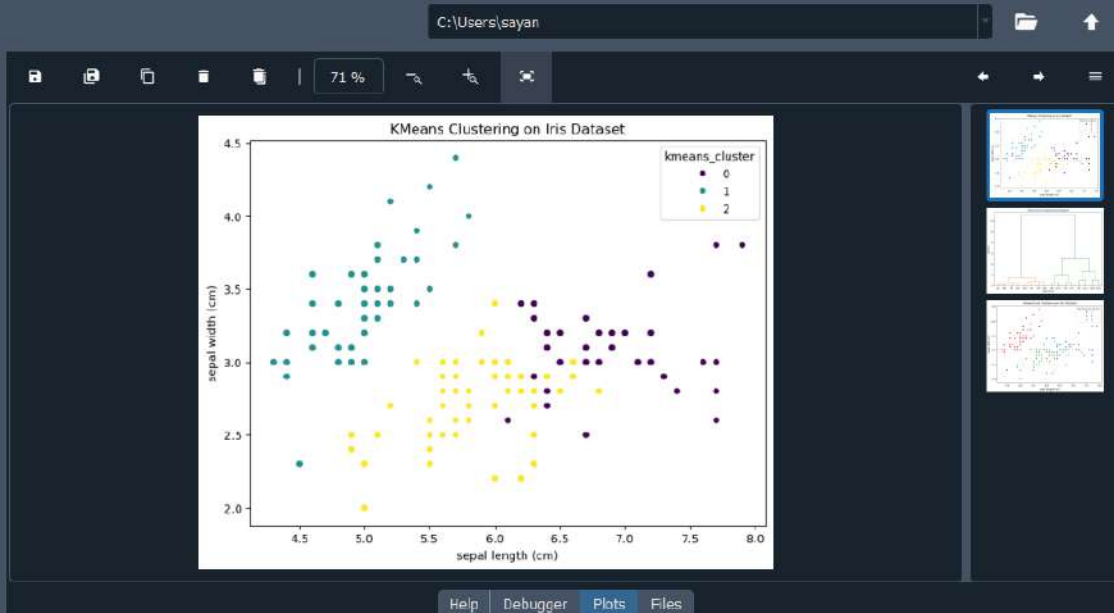
```
Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\sayan\spyder-py3\main_program.py

main_program.py* X

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.datasets import load_iris
6 from sklearn.cluster import KMeans
7 from scipy.cluster.hierarchy import linkage, dendrogram, fcluster
8
9 # 1. Loading and Preprocessing
10 # Load the Iris dataset
11 iris = load_iris()
12
13 # Create DataFrame
14 iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
15
16 # Display basic information
17 print(iris_df.head())
18
19 # 2A. KMeans Clustering
20
21 # Explanation:
22 # KMeans clustering partitions data into K clusters by minimizing the variance within each cl
23 # It is suitable for the Iris dataset due to its well-separated clusters in the feature space
24
25 # Apply KMeans
26 kmeans = KMeans(n_clusters=3, random_state=42)
27 iris_df['kmeans_cluster'] = kmeans.fit_predict(iris_df)
28
29 # Visualize KMeans Clusters
30
31 plt.figure(figsize=(8, 6))
32 sns.scatterplot(x=iris_df.iloc[:, 0], y=iris_df.iloc[:, 1], hue=iris_df['kmeans_cluster'], pa
33 plt.title('KMeans Clustering on Iris Dataset')
34 plt.xlabel(iris.feature_names[0])
35 plt.ylabel(iris.feature_names[1])
36 plt.show()
37
38 # 2B. Hierarchical Clustering
39
```



Console 1/A X

```
In [1]: %runfile C:/Users/sayan/.spyder-py3/main_program.py --wdir
sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)
0 5.1 3.5 1.4 0.2
1 4.9 3.0 1.4 0.2
2 4.7 3.2 1.3 0.2
3 4.6 3.1 1.5 0.2
4 5.0 3.6 1.4 0.2
C:\Users\sayan\AppData\Local\spyder-6\envs\spyder-runtime\Lib\site-
packages\sklearn\cluster\_kmeans.py:1419: UserWarning: KMeans is known to have a memory leak on
Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
warnings.warn(

Python Console History
```

Inline Conda: spyder-runtime (Python: 3.11.11) LSP: Python Line 31, Col 4 UTF-8 CRLF RW Mem 84%

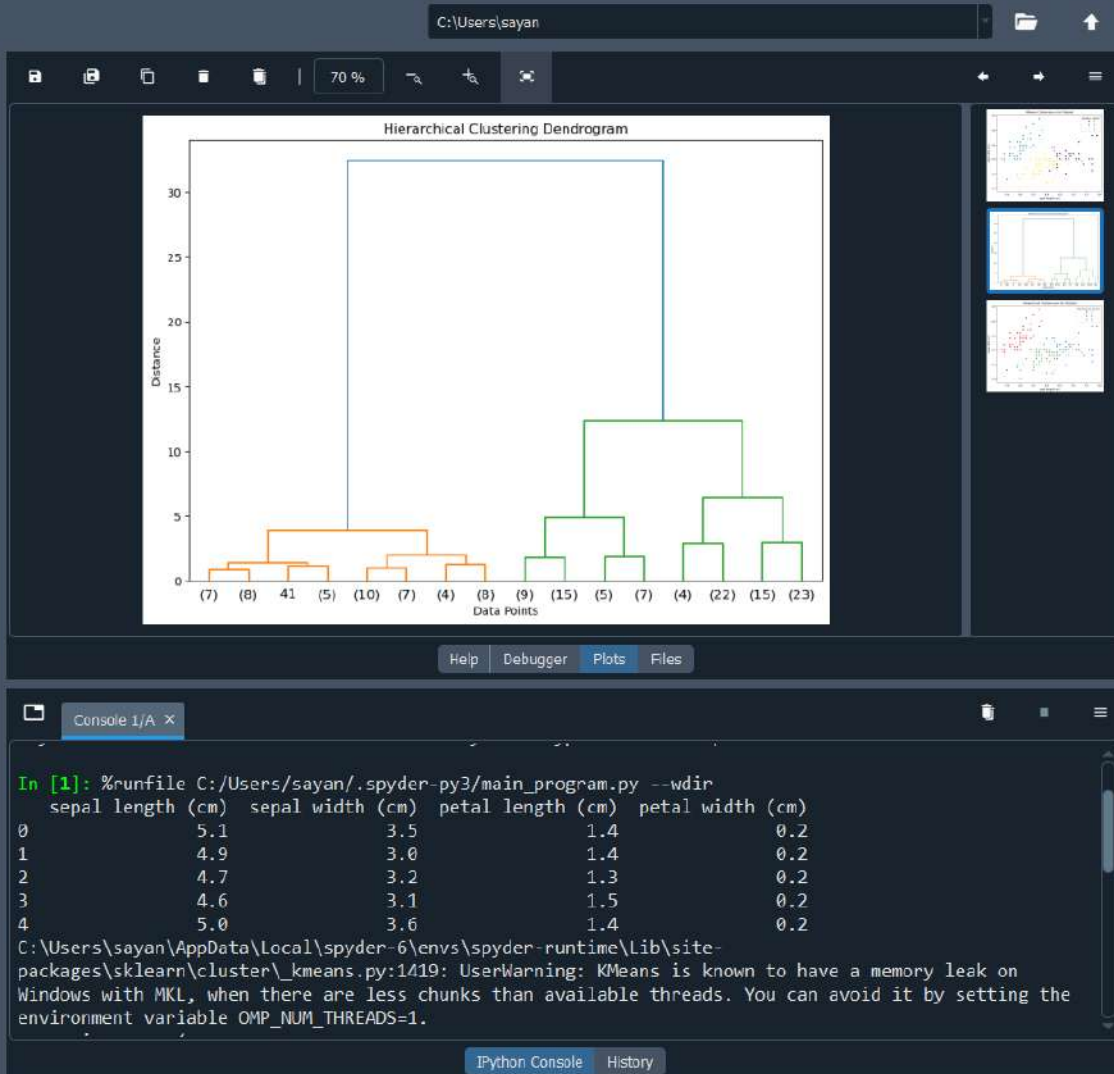
```
Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\sayan\.spyder-py3\main_program.py

main_program.py* X

1 import numpy as np
2 Container object exposing keys as attributes.
3
4 Bunch objects are sometimes used as an output for
5 functions and methods.
6 They extend dictionaries by enabling values to be
7 accessed by key, 'bunch["value_key"]', or by an
8 attribute, 'bunch.value_key'.
9 ...
10 # Load the Iris dataset
11 iris = load_iris()
12
13 # Create DataFrame
14 iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
15
16 # Display basic information
17 print(iris_df.head())
18
19 # 2A. KMeans Clustering
20
21 # Explanation:
22 # KMeans clustering partitions data into K clusters by minimizing the variance within each cl
23 # It is suitable for the Iris dataset due to its well-separated clusters in the feature space
24
25 # Apply KMeans
26 kmeans = KMeans(n_clusters=3, random_state=42)
27 iris_df['kmeans_cluster'] = kmeans.fit_predict(iris_df)
28
29 # Visualize KMeans Clusters
30
31 plt.figure(figsize=(8, 6))
32 sns.scatterplot(x=iris_df.iloc[:, 0], y=iris_df.iloc[:, 1], hue=iris_df['kmeans_cluster'], pa
33 plt.title('KMeans Clustering on Iris Dataset')
34 plt.xlabel(iris.feature_names[0])
35 plt.ylabel(iris.feature_names[1])
36 plt.show()
37
38 # 2B. Hierarchical Clustering
39
```



```
Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\sayan\.spyder-py3\main_program.py

main_program.py* X

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.datasets import load_iris
6 from sklearn.cluster import KMeans
7 from scipy.cluster.hierarchy import linkage, dendrogram, fcluster
8
9 # 1. Loading and Preprocessing
10 # Load the Iris dataset
11 iris = load_iris()
12
13 # Create DataFrame
14 iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
15
16 # Display basic information
17 print(iris_df.head())
18
19 # 2A. KMeans Clustering
20
21 # Explanation:
22 # KMeans clustering partitions data into K clusters by minimizing the variance within each cl
23 # It is suitable for the Iris dataset due to its well-separated clusters in the feature space
24
25 # Apply KMeans
26 kmeans = KMeans(n_clusters=3, random_state=42)
27 iris_df['kmeans_cluster'] = kmeans.fit_predict(iris_df)
28
29 # Visualize KMeans Clusters
30
31 plt.figure(figsize=(8, 6))
32 sns.scatterplot(x=iris_df.iloc[:, 0], y=iris_df.iloc[:, 1], hue=iris_df['kmeans_cluster'], pa
33 plt.title('KMeans Clustering on Iris Dataset')
34 plt.xlabel(iris.feature_names[0])
35 plt.ylabel(iris.feature_names[1])
36 plt.show()
37
38 # 2B. Hierarchical Clustering
39
```

