

```
In [1]: ► import pandas as pd
from sklearn import datasets

# Load the Iris dataset
iris = datasets.load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

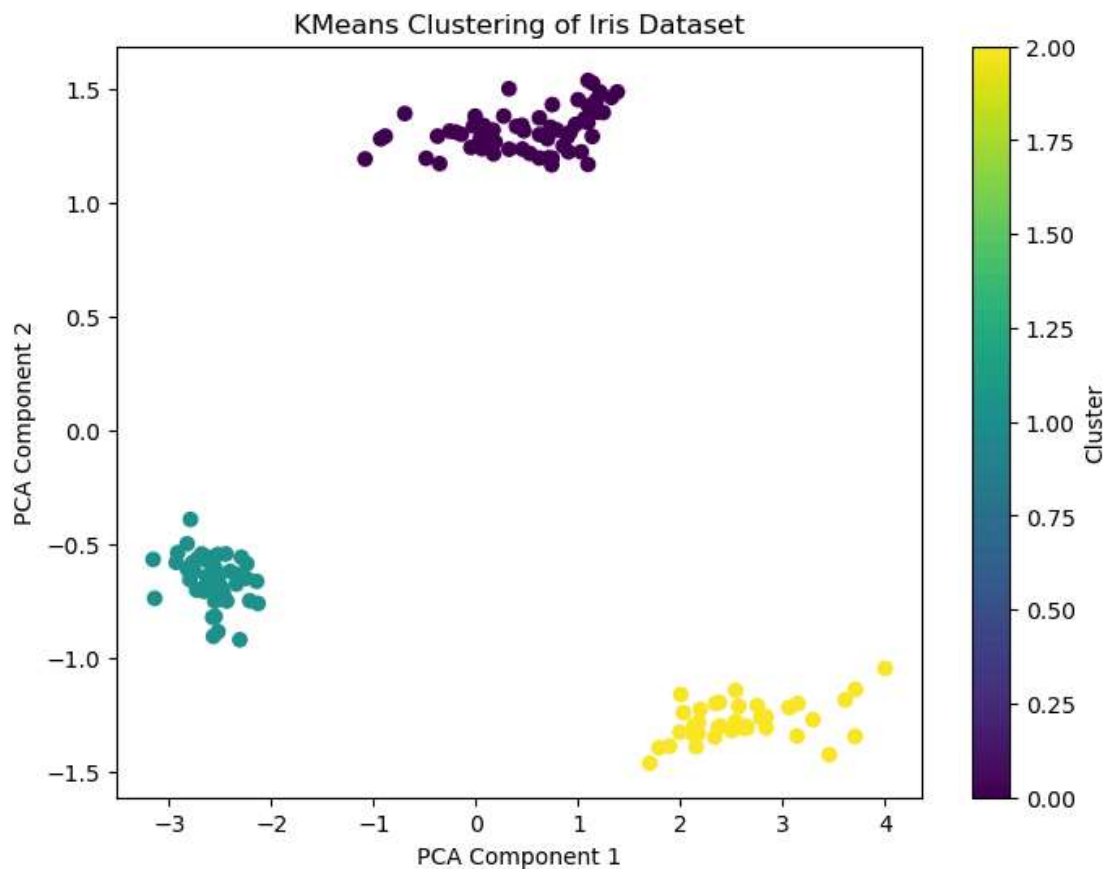
# Display the first few rows
print(df.head())
```

	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

```
In [13]: ▶ import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import os
os.environ["OMP_NUM_THREADS"] = "1"
# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
df['KMeans_Cluster'] = kmeans.fit_predict(df)

# Visualize the clusters using PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(df)

plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=df['KMeans_Cluster'], cn
plt.title('KMeans Clustering of Iris Dataset')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(label='Cluster')
plt.show()
```



```
In [14]: from scipy.cluster.hierarchy import dendrogram, linkage
import seaborn as sns

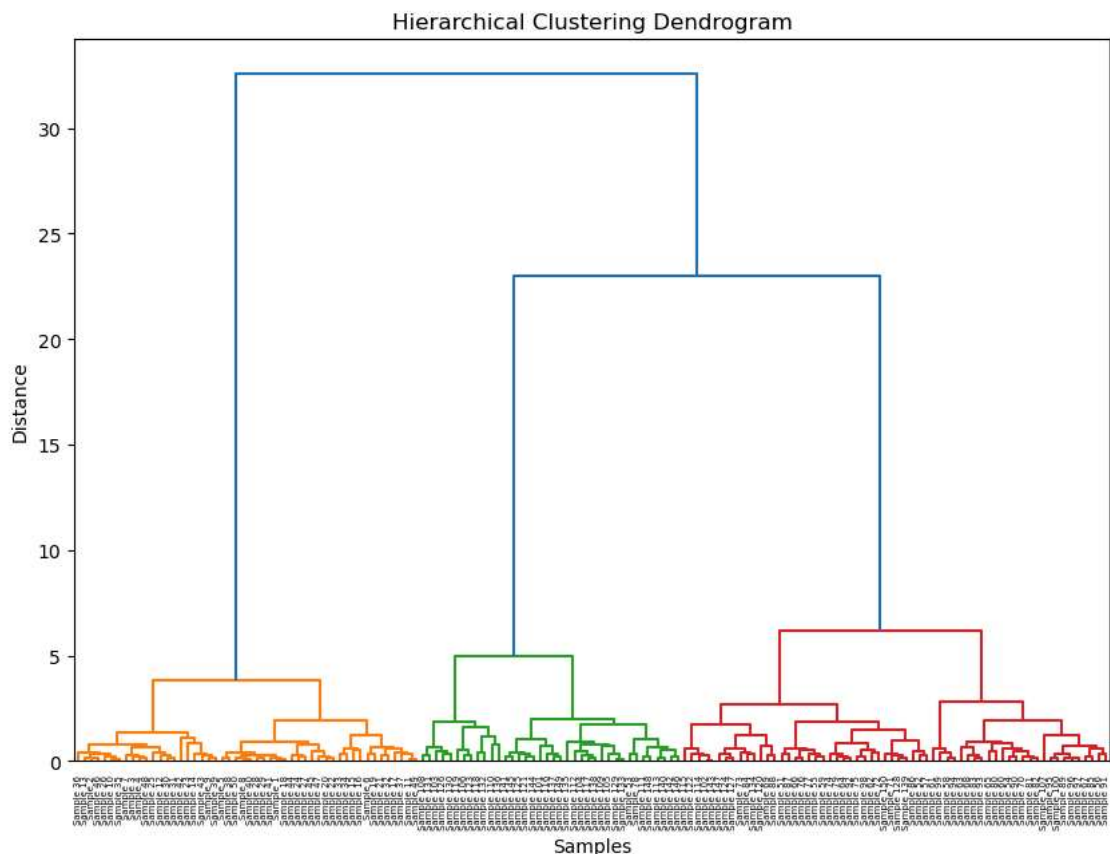
# Perform Hierarchical clustering
Z = linkage(df, method='ward')

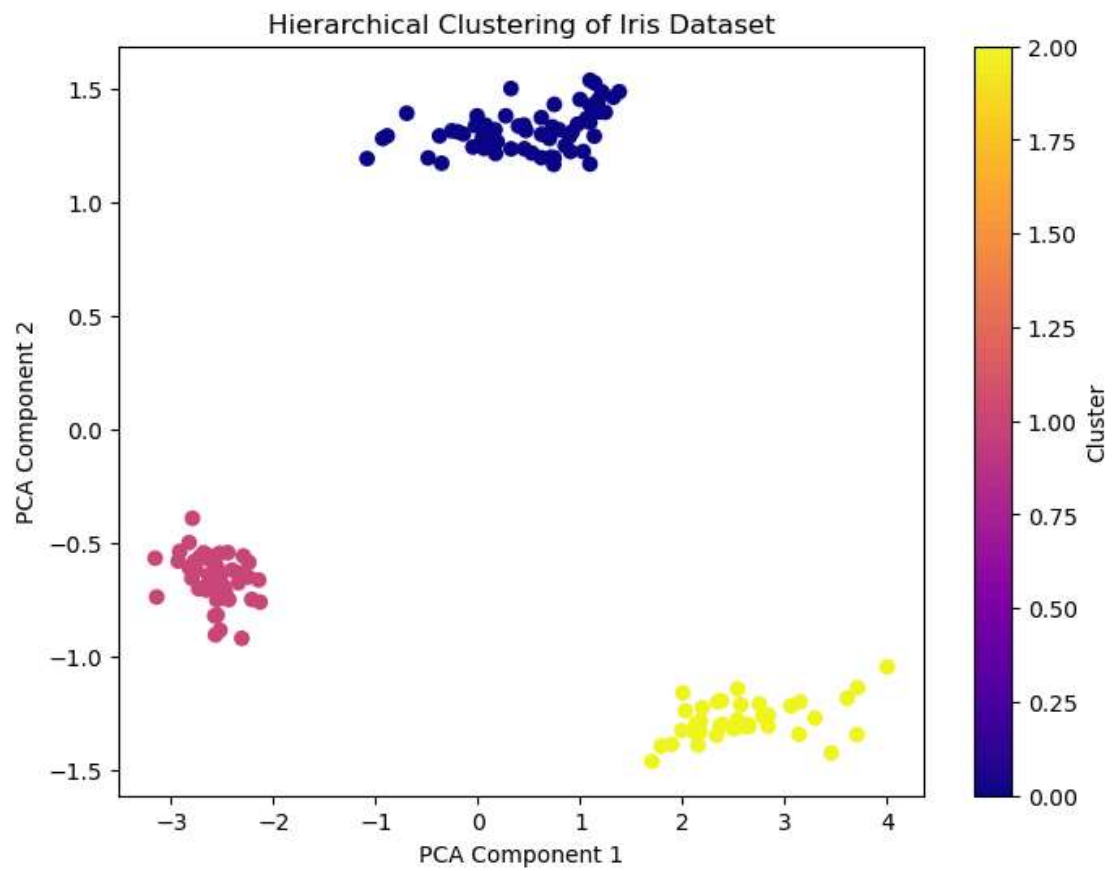
# Create a dendrogram
plt.figure(figsize=(10, 7))
dendrogram(Z, labels=[f'Sample {i+1}' for i in range(len(df))]) # Generic
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Samples')
plt.ylabel('Distance')
plt.show()

# Apply cluster labels based on the dendrogram
from sklearn.cluster import AgglomerativeClustering

hierarchical = AgglomerativeClustering(n_clusters=3)
df['Hierarchical_Cluster'] = hierarchical.fit_predict(df)

# Visualize the clusters using PCA
plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=df['Hierarchical_Cluster'])
plt.title('Hierarchical Clustering of Iris Dataset')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(label='Cluster')
plt.show()
```





In []: ▶