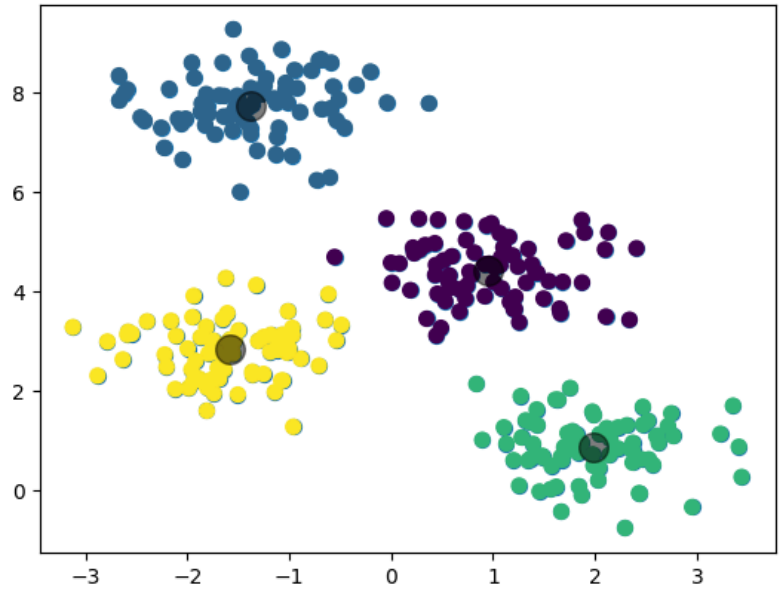
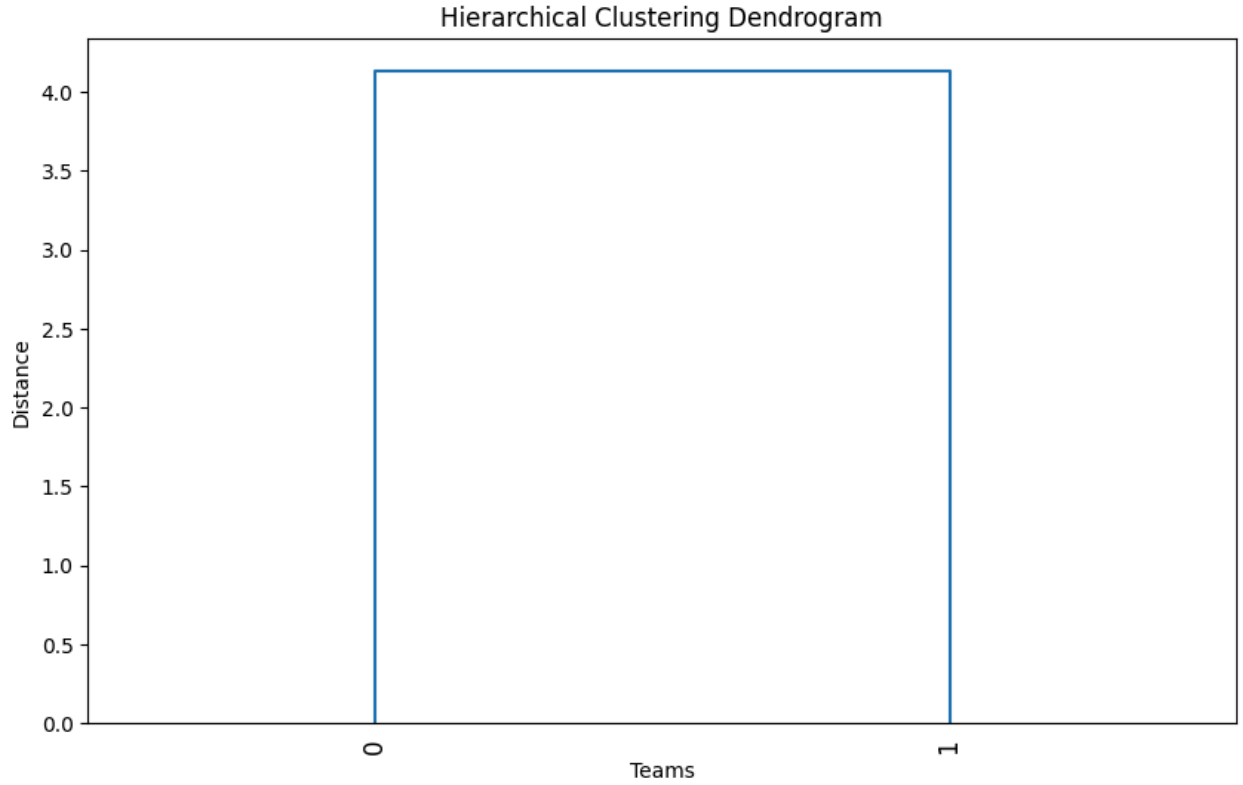
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs

# Generating synthetic data
X, y = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)

plt.scatter(X[:, 0], X[:, 1], s=50)
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
from sklearn.cluster import KMeans

plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_

plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)

plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_
  warnings.warn(
```



```
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage

# Sample data (replace with your dataset)
# Each row represents a data point, and columns represent features
# Here, we have hypothetical features like batting average, bowling average, etc.
data = np.array([
    [35.2, 28.5, 42.7],  # Example feature values for a team
    [38.7, 30.1, 41.2],
    # Add more data points as needed
])

# Compute the linkage matrix using hierarchical clustering
# You can choose different linkage methods and distance metrics as per your requirement
# Here, we are using 'ward' linkage and Euclidean distance
Z = linkage(data, method='ward')

# Plot the dendrogram
plt.figure(figsize=(10, 6))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Teams')
plt.ylabel('Distance')
dendrogram(
    Z,
    leaf_rotation=90.,
    leaf_font_size=12.,
)
plt.show()

# Predict the winning IPL team based on the dendrogram
# You can choose the appropriate threshold to cut the dendrogram and form clusters
# For simplicity, let's assume the top two clusters represent the finalists
# You can adjust this based on your data and requirements
num_clusters = 2
cluster_labels = np.arange(len(data))
finalists = []
for i in range(num_clusters):
    cluster_data_indices = np.where(cluster_labels == i)[0]
    finalists.append(cluster_data_indices)

# Sample output: You would replace this with the actual names of IPL teams
ipl_teams = ["Team A", "Team B", "Team C", "Team D", "Team E", "Team F", "Team G", "Team H"]
print("Predicted finalists for 2024 IPL:")
for finalist in finalists:
    print(", ".join([ipl_teams[i] for i in finalist]))
```




```
Predicted finalists for 2024 IPL:
Team A
Team B
```