**#K means cluster**

import matplotlib.pyplot as plt

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

plt.scatter(x, y)

plt.show()

from sklearn.cluster import KMeans

data = list(zip(x, y))

inertias = []

for i in range(1,11):

kmeans = KMeans(n\_clusters=i)

kmeans.fit(data)

inertias.append(kmeans.inertia\_)

plt.plot(range(1,11), inertias, marker='o')

plt.title('Elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('Inertia')

plt.show()

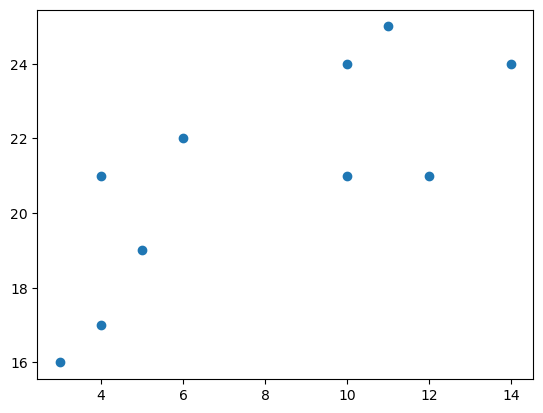
kmeans = KMeans(n\_clusters=2)

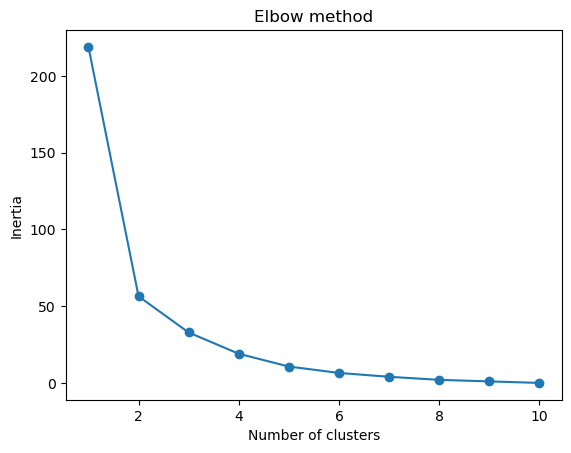
kmeans.fit(data)

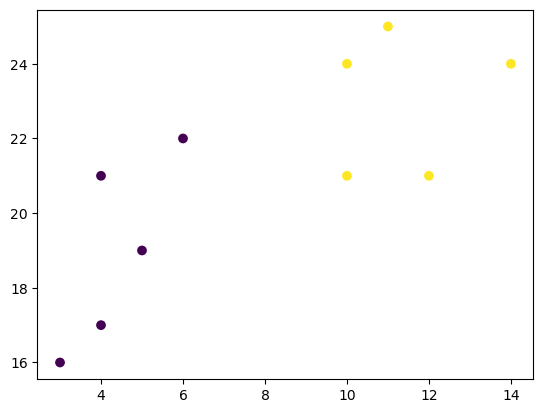
plt.scatter(x, y, c=kmeans.labels\_)

plt.show()

OUTPUT:







**#Hierarchical Clustering:**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from scipy.spatial.distance import pdist, squareform**

**from scipy.cluster.hierarchy import linkage, dendrogram**

**np.random.seed(42)**

**data\_points = np.random.rand(6, 2)**

**colors = ['r', 'g', 'b', 'c', 'm', 'y']**

**plt.figure(figsize=(8, 6))**

**for i in range(len(data\_points)):**

**plt.scatter(data\_points[i, 0], data\_points[i, 1], color=colors[i], label=f'Data Point {i+1}')**

**plt.legend()**

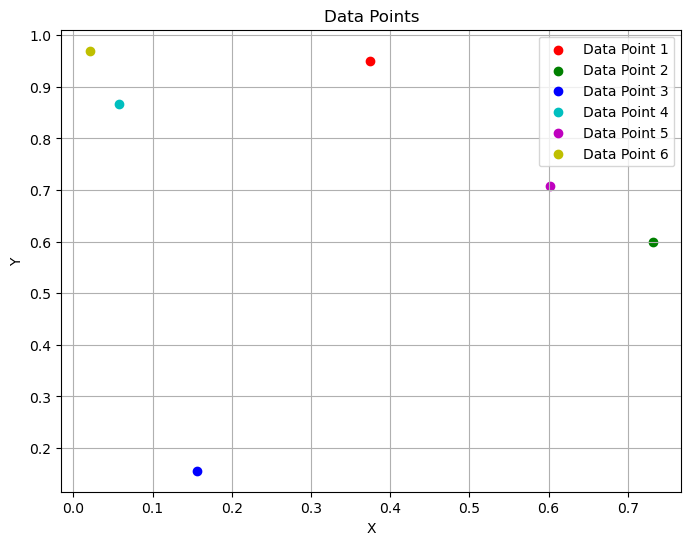
**plt.title('Data Points')**

**plt.xlabel('X')**

**plt.ylabel('Y')**

**plt.grid(True)**

**plt.show()**

****