**Experiment 05**

**Aim:** Implementation of Clustering Algorithm Using

1. k-means

2. Hierarchical (single/complete/average)

**Code:**

**K means-clustering (iris dataset)**

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

iris = load\_iris()

n\_clusters = 3

kmeans = KMeans(n\_clusters=n\_clusters, max\_iter=100, random\_state=0)

kmeans.fit(iris.data)

cluster\_labels = kmeans.predict(iris.data)

import matplotlib.pyplot as plt

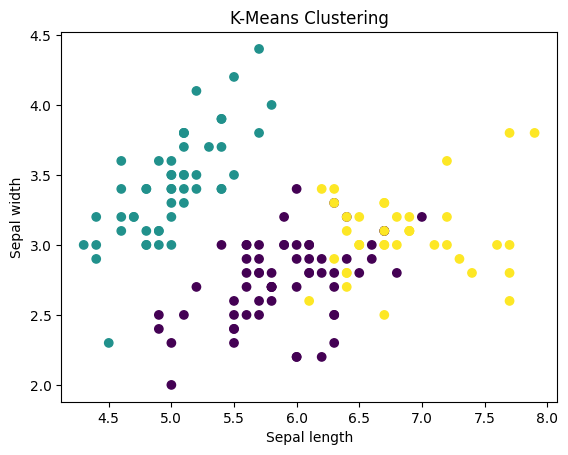
plt.scatter(iris.data[:, 0], iris.data[:, 1], c=cluster\_labels)

plt.xlabel("Sepal length")

plt.ylabel("Sepal width")

plt.title("K-Means Clustering")

plt.show()



**Hierarchical Clustering**

from sklearn.cluster import AgglomerativeClustering

clustering = AgglomerativeClustering(linkage='ward', affinity='euclidean')

clustering.fit(iris.data)

cluster\_labels = clustering.labels\_

import scipy.cluster.hierarchy as sch

#single

dendrogram = sch.dendrogram(sch.linkage(iris.data, method='single'))

plt.show()

#average

dendrogram = sch.dendrogram(sch.linkage(iris.data, method='average'))

plt.show()

#complete

dendrogram = sch.dendrogram(sch.linkage(iris.data, method='complete'))

plt.show()

A graph of a graph

Description automatically generated with medium confidence

A diagram of a city

Description automatically generated

A diagram of a city

Description automatically generated with medium confidence

**Elbow optimization**

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

iris = load\_iris()

k\_range = range(1, 10)

wss\_scores = []

for k in k\_range:

    kmeans = KMeans(n\_clusters=k, random\_state=0)

    kmeans.fit(iris.data)

    wss\_scores.append(kmeans.inertia\_)

import matplotlib.pyplot as plt

plt.plot(k\_range, wss\_scores, marker="o")

plt.xlabel("Number of clusters")

plt.ylabel("Within-sum-of-squares")

plt.title("Elbow Method")

plt.show()

A graph with a line

Description automatically generated

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.datasets import load\_iris

iris = load\_iris()

n\_clusters = 3

kmeans = KMeans(n\_clusters=n\_clusters, max\_iter=100, random\_state=0)

kmeans.fit(iris.data)

cluster\_labels = kmeans.predict(iris.data)

import matplotlib.pyplot as plt

plt.scatter(iris.data[:, 0], iris.data[:, 1], c=cluster\_labels)

plt.xlabel("Sepal length")

plt.ylabel("Sepal width")

plt.title("K-Means Clustering")

plt.show()

A diagram of a number of dots

Description automatically generated