Bahria University,

Karachi Campus

## LAB EXPERIMENT NO.

9

## LIST OF TASKS

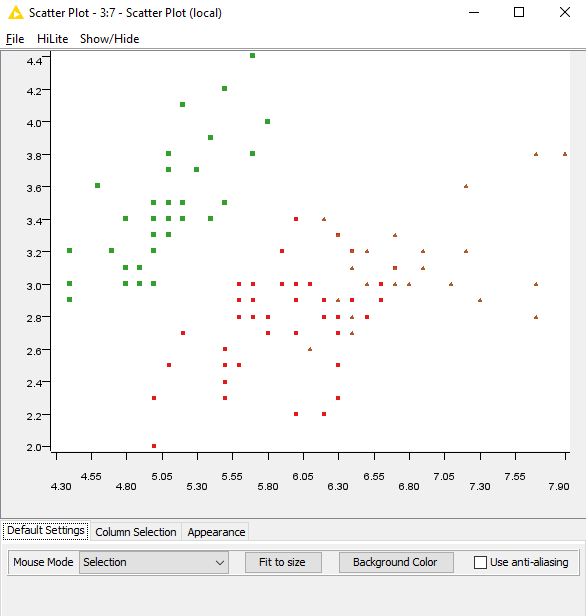
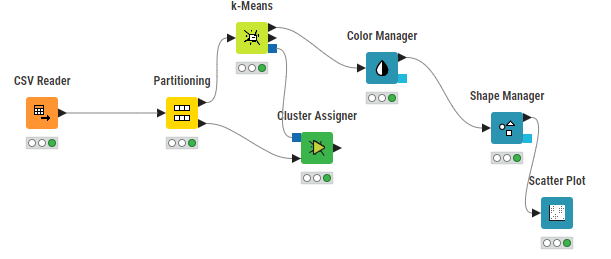
|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | You work as a data scientist at a botanic garden, and your team is interested in analyzing the Iris dataset to identify patterns in the measurements of different types of iris flowers. The dataset contains measurements for the sepal length, sepal width, petal length, and petal width of three different types of iris flowers: Setosa, versicolor, and virginica. Your goal is to use K-means clustering to group the flowers based on their measurements and determine if the measurements can be used to accurately identify the different types of iris flowers.  Perform K-Mean Clustering in KNIME on IRIS dataset |
| **2** | You work as a data scientist at a botanic garden, and your team is interested in analyzing the Iris dataset to identify patterns in the measurements of different types of iris flowers. The dataset contains measurements for the sepal length, sepal width, petal length, and petal width of three different types of iris flowers: setosa, versicolor, and virginica. Your goal is to use K-means clustering to group the flowers based on their measurements and determine if the measurements can be used to accurately identify the different types of iris flowers. |

Submitted On:

01 may 2024

(Date: DD/MM/YY)

**TASK#1:** Perform K-Mean Clustering in KNIME on IRIS dataset



**PYTHON CODE**

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import numpy as np

iris\_df = pd.read\_csv('iris.data.csv', header=None)

iris\_df.columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'species']

iris\_features = iris\_df.drop('species', axis=1)

ata

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(iris\_features)

inertia = []

for k in range(1, 11):

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

kmeans.fit(scaled\_data)

inertia.append(kmeans.inertia\_)

plt.figure(figsize=(8, 4))

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

plt.title('Elbow Method For Optimal k')

plt.xlabel('Number of clusters')

plt.ylabel('Inertia')

plt.show()

optimal\_clusters = 3

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

clusters = kmeans.fit\_predict(scaled\_data)

unique\_clusters = np.unique(clusters)

markers = ['^', 's', '\*'] # Different markers for each cluster

colors = ['blue', 'green', 'red'] # Colors for each cluster

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

for i, marker, color in zip(unique\_clusters, markers, colors):

plt.scatter(scaled\_data[clusters == i, 0], scaled\_data[clusters == i, 1], c=color, label=f'Cluster {i}', marker=marker, edgecolor='k', s=50)

plt.title('K-means Clustering on Iris Dataset (Scaled)')

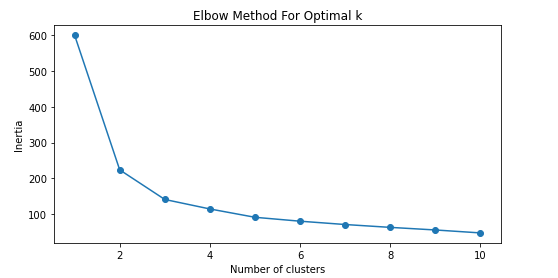
plt.xlabel('Sepal Length')

plt.ylabel('Sepal Width')

plt.legend()

plt.colorbar(label='Cluster Label')

plt.show()

A chart of clustering data

Description automatically generated

**TASK#2:** You work as a data scientist at a botanic garden, and your team is interested in analyzing the Iris dataset to identify patterns in the measurements of different types of iris flowers.

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import numpy as np

iris\_data = pd.read\_csv('iris.data.csv', header=None)

iris\_data.columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'species']

features = iris\_data.drop('species', axis=1)

standard\_scaler = StandardScaler()

standardized\_features = standard\_scaler.fit\_transform(features)

inertia\_values = []

for num\_clusters in range(1, 11):

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

    kmeans.fit(standardized\_features)

    inertia\_values.append(kmeans.inertia\_)

optimal\_num\_clusters = 3

plt.figure(figsize=(8, 4))

plt.plot(range(1, 11), inertia\_values, marker='o', color='black'

plt.plot(optimal\_num\_clusters, inertia\_values[optimal\_num\_clusters-1], marker='X', markersize=10, markeredgecolor='blue', markerfacecolor='blue', label='Optimal k')

plt.title('Elbow Method to Determine Optimal k')

plt.xlabel('Number of Clusters')

plt.ylabel('Inertia')

plt.legend()

plt.grid(True)

plt.show()

kmeans\_model = KMeans(n\_clusters=optimal\_num\_clusters, random\_state=0)

cluster\_labels = kmeans\_model.fit\_predict(standardized\_features)

unique\_labels = np.unique(cluster\_labels)

markers = ['o', 'v', 'D']

colors = ['purple', 'orange', 'cyan']

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

for cluster\_label, marker, color in zip(unique\_labels, markers, colors):

    points = standardized\_features[cluster\_labels == cluster\_label]

    plt.scatter(points[:, 0], points[:, 1], c=color, label=f'Cluster {cluster\_label}', marker=marker, edgecolor='black', s=50)

    plt.scatter(points[0, 0], points[0, 1], c='yellow', marker='s', edgecolor='black', s=100, label=f'Center of Cluster {cluster\_label}')

plt.title('K-means Clustering of Iris Data (Standardized)')

plt.xlabel('Standardized Sepal Length')

plt.ylabel('Standardized Sepal Width')

plt.legend()

plt.grid(True)

plt.show()

