Exercise 5a Date: 22/01/2025

K Means Clustering - Two-dimensional dataset

Aim: To apply the K-Means clustering algorithm to a two-dimensional dataset, to identify clusters within the data.

Algorithm:

K-Means is an unsupervised learning algorithm used for clustering.

The goal is to partition the data into distinct clusters based on similarity, with each cluster representing data points that are more similar to each other than to those in other clusters.

The K-Means algorithm works by partitioning the dataset into `k` clusters, where each data point belongs to the cluster with the nearest mean.

The algorithm follows these steps:

1. **Initialization**: Choose `k` initial cluster centroids randomly from the dataset.
2. **Assignment Step**: Assign each data point to the nearest centroid. This creates `k` clusters.
3. **Update Step**: Recalculate the centroids of each cluster based on the data points assigned to it.
4. **Convergence**: Repeat the assignment and update steps until the centroids no longer change significantly or a maximum number of iterations is reached.

Step 1: Import necessary libraries

- Import necessary Python libraries such as `pandas`, `sklearn`, and `matplotlib` to handle data manipulation, model training, and visualization.

Step 2: Load the Dataset

- Load the Iris dataset using `load_iris` from `sklearn.datasets`. Select the first two features (Sepal Length and Sepal Width) as `X` and set the target variable `y`.

Step 3: Preprocess the Data

- Scale the feature data using `StandardScaler` to standardize it, ensuring that all features are normalized and have equal importance.

Step 4: Apply KMeans clustering

- Initialize the KMeans model with `n_clusters=3` and fit it to the scaled data. The algorithm assigns each data point to one of the three clusters.

Step 5: Add cluster labels to the dataset

- After performing the clustering, add the cluster labels to the dataset to identify which data points belong to which cluster.

Step 6: Visualize the results

- Plot the clustering results using the first two features (Sepal Length and Sepal Width) with each point colored by its assigned cluster. The plot shows how the K-Means algorithm has grouped the data into three clusters.

Importing necessary libraries

In [5]:
```
import pandas as pd
import warnings
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
warnings.filterwarnings("ignore")
```

Load the Dataset

In [6]:
```
iris = load_iris()
X = iris.data[:, :2]
y = iris.target
feature_names = iris.feature_names[:2]
target_names = iris.target_names
```

Preprocess the Data

In [7]:
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Applying KMeans clustering

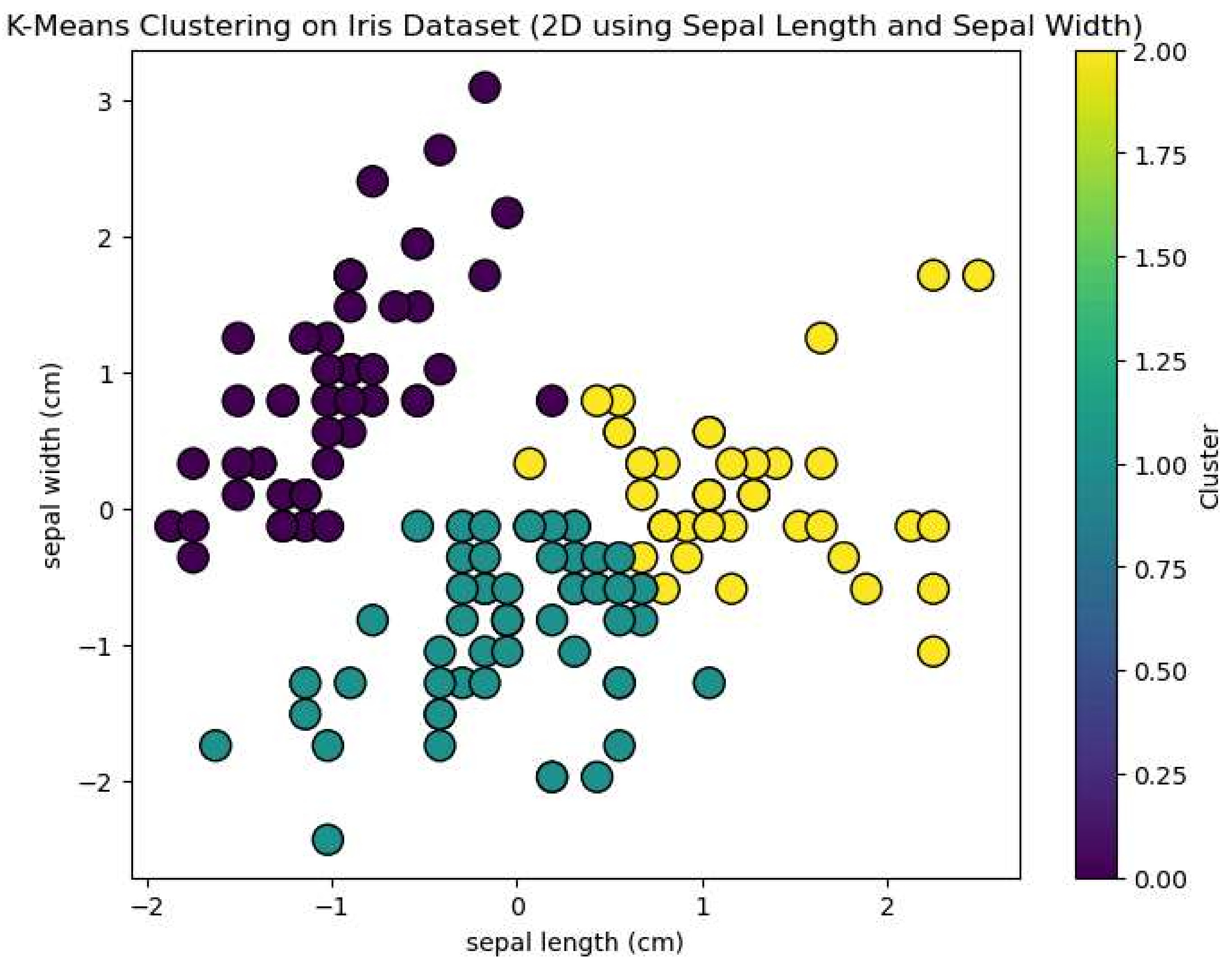
In [8]:
```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_scaled)
y_kmeans = kmeans.predict(X_scaled)
```

Adding cluster labels to the dataset

In [10]:
```
df = pd.DataFrame(X_scaled, columns=feature_names)
df['Cluster'] = y_kmeans
```

Plotting the clustering result (2D visualization)

In [11]:
```
plt.figure(figsize=(8, 6))
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c=df['Cluster'], cmap='viridis', edgecolor='k', s=150)
plt.xlabel(feature_names[0])
plt.ylabel(feature_names[1])
plt.title('K-Means Clustering on Iris Dataset (2D using Sepal Length and Sepal Width)')
plt.colorbar(label='Cluster')
plt.show()
```



Result

The K-Means algorithm successfully identified three clusters within the dataset that correspond to the three iris species.