Exercise 5b Date: 22/01/2025

K Means Clustering - Multidimensional dataset

Aim: To apply the K-Means clustering algorithm to a multidimensional 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`. Split the data into features (`X`) and the target variable (`y`).

Step 3: Preprocess the data

- Scale the feature data to standardize it using `StandardScaler`, ensuring that all features are given equal importance by normalizing them.

Step 4: Apply K-Means 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

- After the clustering is completed, add the cluster labels (assignments) to the dataset to identify which data points belong to each cluster.

Step 6: Visualize the results

- Plot the results of the clustering using the first two features of the Iris dataset, coloring each point according to its assigned cluster. The plot visualizes how the K-Means algorithm has grouped the data into three clusters.

Import necessary libraries

In [6]:
```
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 [7]:
```
iris = load_iris()
X = iris.data
y = iris.target
feature_names = iris.feature_names
target_names = iris.target_names
```

Preprocess the data

In [8]:
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Applying K Means Clustering

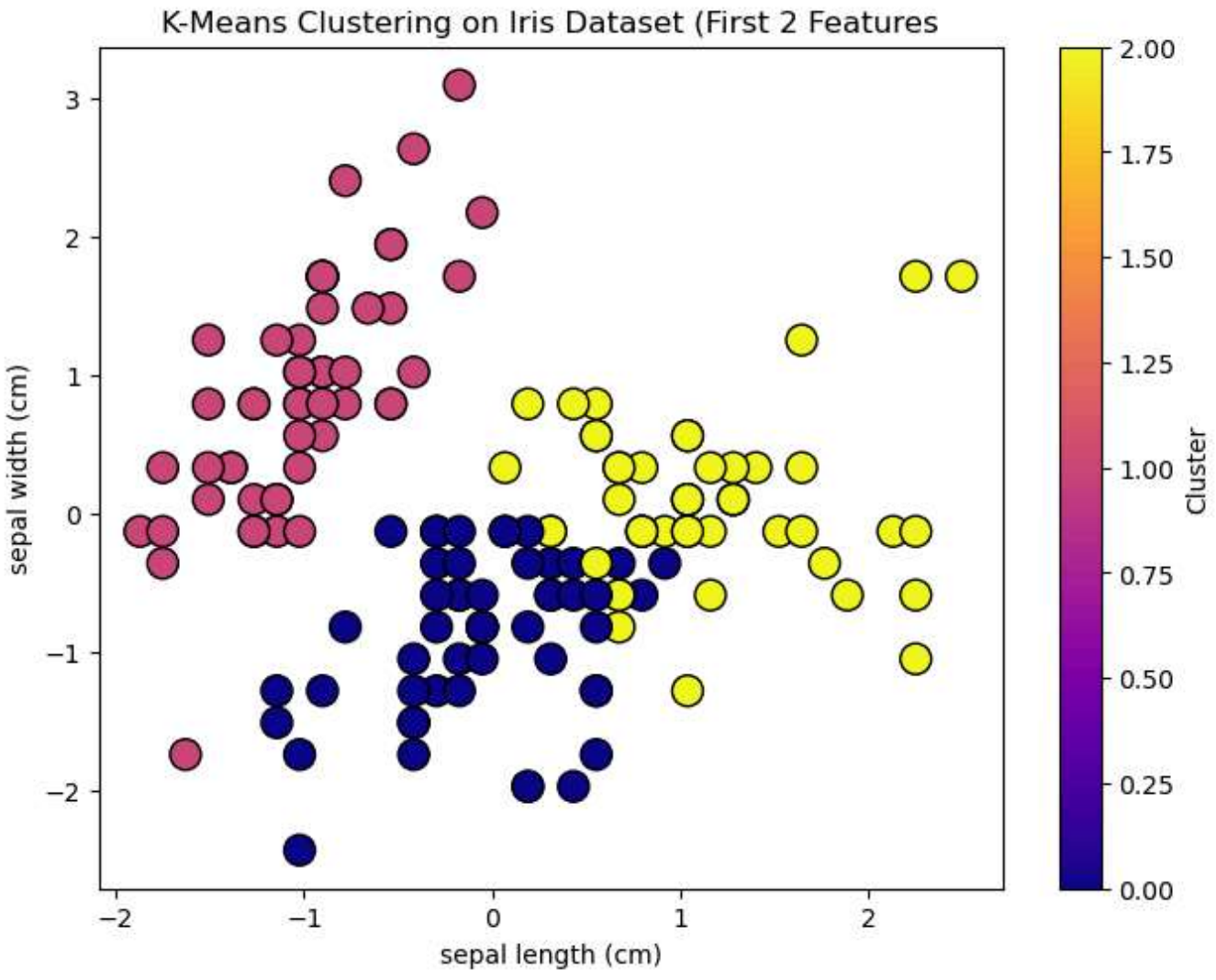
In [9]:
```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_scaled)
y_kmeans = kmeans.predict(X_scaled)
```

Adding cluster labels

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

Plot the clustering results

In [11]:
```
plt.figure(figsize=(8, 6))
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c=df['Cluster'], cmap='plasma', edgecolor='k', s=150)
plt.xlabel(feature_names[0])
plt.ylabel(feature_names[1])
plt.title('K-Means Clustering on Iris Dataset (First 2 Features')
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.
