Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

Assignment (3.3)

Perform k-means clusterization on the Iris dataset. Repeat the procedure on the dataset reduced with PCA, and then compare the results.

Solution

1. First we **import libraries**.

```
[1]  ✓ 0.8s
```

```
from sklearn import datasets as dsets, cluster as clst, decomposition as dComp
import matplotlib.pyplot as plt
import numpy as np
```

2. Then **load Iris data** in **iris_data variable**.

```
[2]  ✓ 0.0s
```

```
iris_data = dsets.load_iris()
```

3. Then find the number **k** using **Elbow** method.

```
[24]  ✓ 0.9s
```

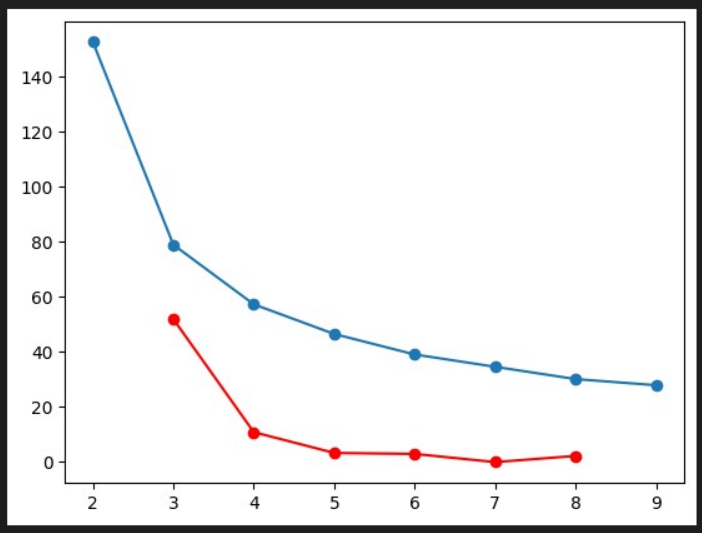
```
k_values = []
intertia_scores = []
for k in range(2,10):
    model = clst.KMeans(n_clusters=k)
    model.fit(iris_data.data)
    intertia_scores.append(model.inertia_)
    k_values.append(k)
module_of_second_derivative = np.abs(np.diff(np.diff(intertia_scores)))
```

```
... /home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
```

4. Then plot its **graph**.

```
plt.plot(k_values, intertia_scores)
plt.scatter(k_values, intertia_scores)
plt.plot(k_values[1:-1], module_of_second_derivative, color='red')
plt.scatter(k_values[1:-1], module_of_second_derivative, color='red')
plt.show()
```

[4] ✓ 0.2s



Here we find the number **k**, which is **3**.

5. Now **create** the model and **train** it and predict the **result**.

```
model = clst.KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(iris_data.data)
```

[5] ✓ 0.0s

```
KMeans(max_iter=100, n_clusters=3, n_init=1)
```

```
all_predictions = model.predict(iris_data.data)
centroids = model.cluster_centers_
centroids.shape
```

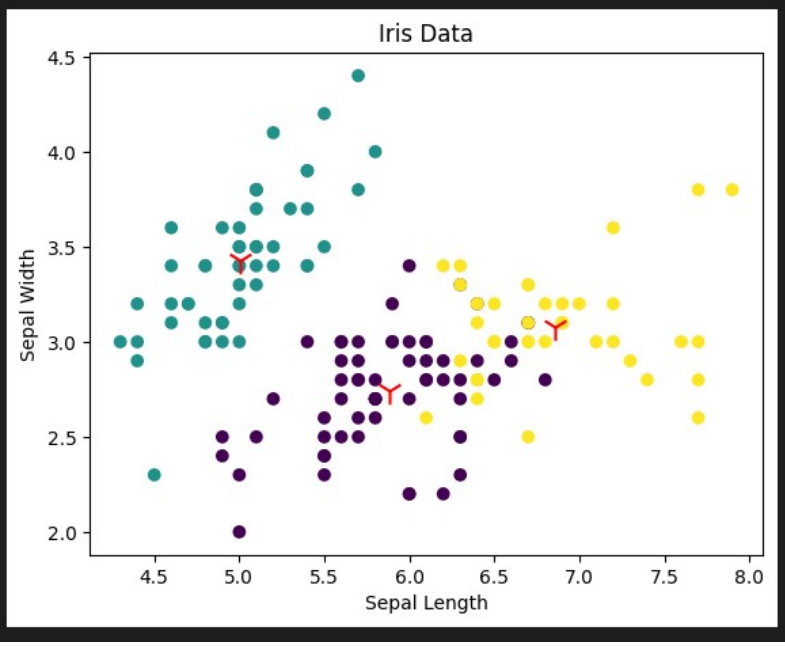
[6] ✓ 0.0s

```
(3, 4)
```

6. Now **plot** the **graph**.

```
plt.scatter(iris_data.data[:, 0], iris_data.data[:, 1], c=all_predictions)
plt.scatter(centroids[:, 0], centroids[:, 1], marker='1', s=200, c='red')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.title('Iris Data')
plt.show()
```

[7] ✓ 0.1s



7. Now we do **PCA**.

```
pca = dComp.PCA(n_components=2)
x_reduced = pca.fit_transform(iris_data.data)
x_reduced.shape
```

[8] ✓ 0.0s

```
(150, 2)
```

8. Now again we **make a model** and **train** it.

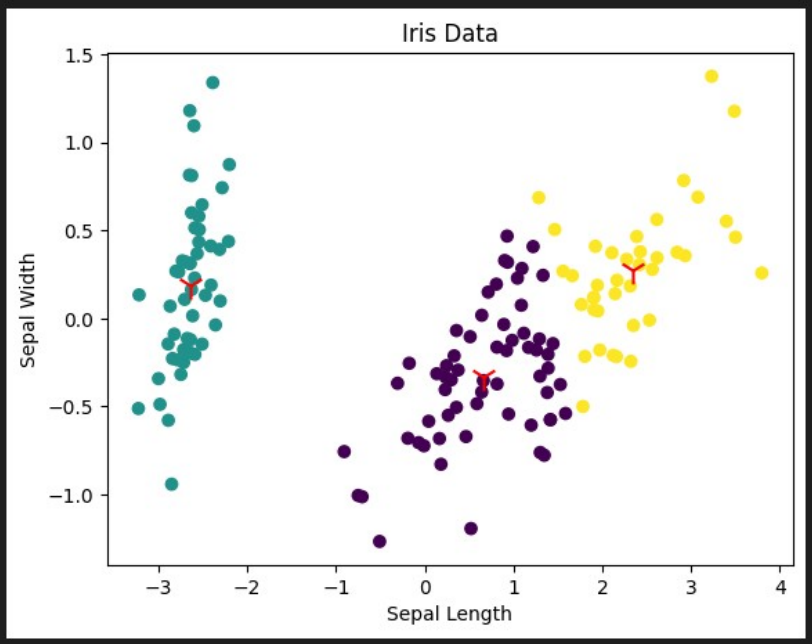
```
model_pca = clst.KMeans(n_clusters=3, n_init=1, max_iter=100)
model_pca.fit(x_reduced)
all_predictions_pca = model_pca.predict(x_reduced)
centroids_pca = model_pca.cluster_centers_
```

[9] ✓ 0.0s

9. Now **plot** the graph.

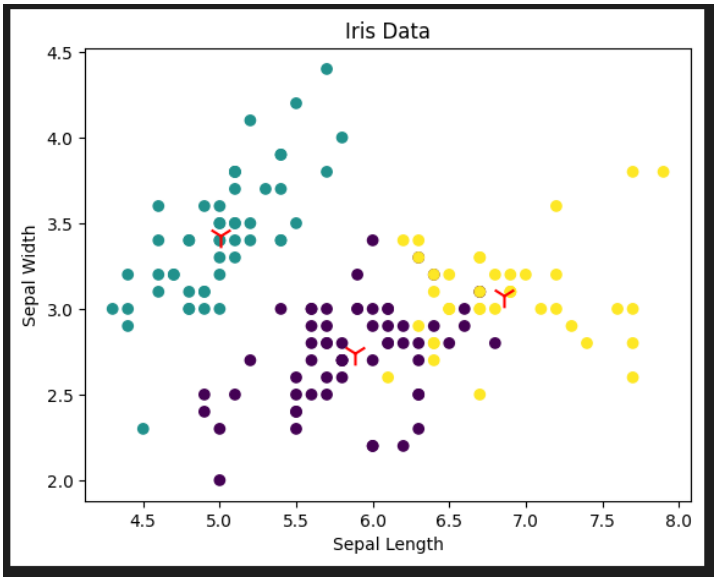
```
plt.scatter(x_reduced[:, 0], x_reduced[:, 1], c=all_predictions)
plt.scatter(centroids_pca[:, 0], centroids_pca[:, 1], marker='1', s=200, c='red')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.title('Iris Data')
plt.show()
```

[10] ✓ 0.2s

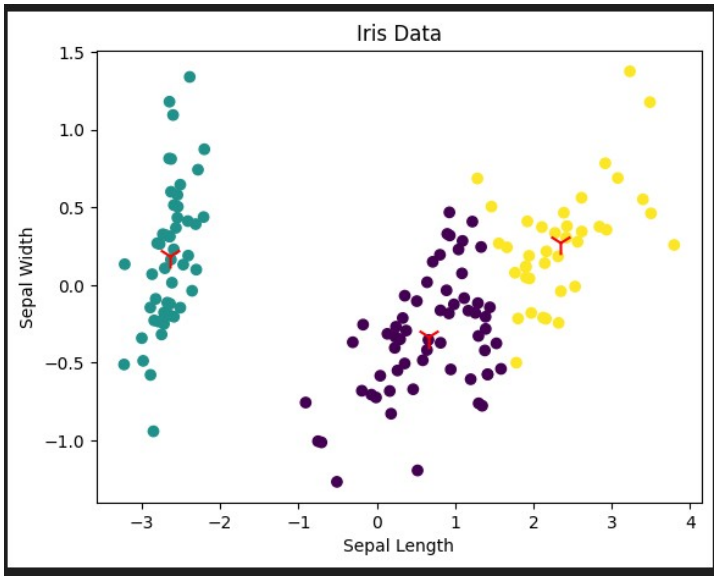


Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

Conclusion



Before PCA



After PCA

Here we can clearly see that **before PCA** our data is scattered and mix with other clusters and when we perform **PCA** our clusters become more **organized** and we can easily read it.