

Name :

Rahimasiddiqui(2303.KHI.DEG.030)

Peer Name:

Osama Abdul Razzak (2303.KHI.DEG.029)

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.

In the beginning we import the required libraries

```
[3]: import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

And store the Iris dataset to variable 'Iris_data'

```
[4]: iris_data = datasets.load_iris()
```

Assign the dataset Features to variable 'x' and target value to variable 'y'

```
[5]: x = iris_data.data
y = iris_data.target
```

Name :

Rahimasiddiqui(2303.KHI.DEG.030)

Peer Name:

Osama Abdul Razzak (2303.KHI.DEG.029)

Then we create K-Mean clustering model and setting cluster value to 3 because in Iris datasets, there are 3 different species of Iris flower, in which we want to cluster data point into it

n_init = 1 mean that the algorithm will be only run at once with single random initialization of centroids

max_iter = 100 indicates that the algorithm will perform 100 iteration then fit our model to the 'x' datasets using fit() method

Predict the cluster assignment for each data point

Storing result in 'all predictions' variable

And find centroid, by using cluster_centers_ attribute

Using K-Mean

```
[6]: model = KMeans(n_clusters=3, n_init=1, max_iter=100)
      model.fit(x)

      all_predictions = model.predict(x)
      centroids = model.cluster_centers_
      centroids      | #centroid of each species w.r.t to features(Sepal_Lenght, Sepal_Width, Petal_Lenght, Petal_Width)

[6]: array([[5.88360656, 2.74098361, 4.38852459, 1.43442623],
            [5.006      , 3.428      , 1.462      , 0.246      ],
            [6.85384615, 3.07692308, 5.71538462, 2.05384615]])
```

After that we visualize the cluster assignment and centroid using plt.scatter() and plt.show()

Name :

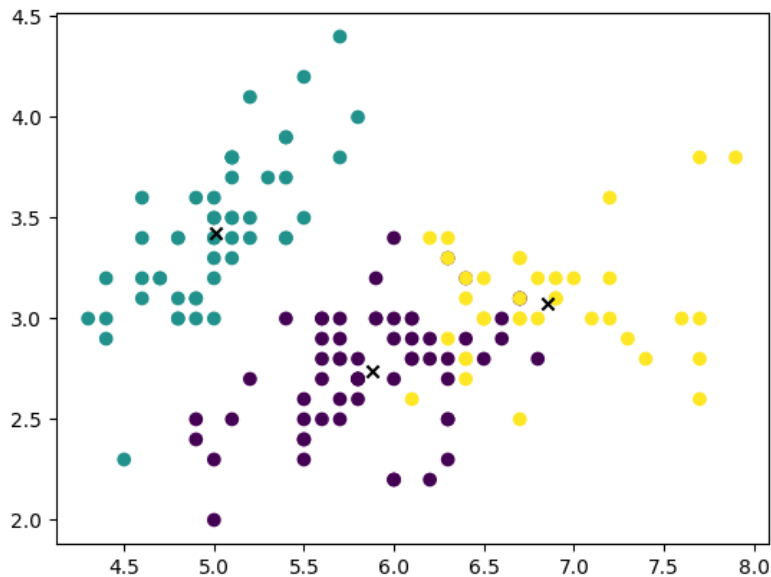
Rahimasiddiqui(2303.KHI.DEG.030)

Peer Name:

Osama Abdul Razzak (2303.KHI.DEG.029)

```
[7]: plt.scatter(x[:,0], x[:,1], c=all_predictions)
plt.scatter(centroids[:,0], centroids[:,1], marker='x', color="black")
plt.show
```

```
[7]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Then we Use elbow method for confirming that either we selected the optimal number of cluster or not

Using Elbow method to determine optimal number of cluster

```
[7]: k_values = []
inertia_scores = []

for k in range(2,10):
    model = KMeans(n_clusters=k)
    model.fit(x)
    inertia_scores.append(model.inertia_)
    k_values.append(k)

module_of_second_derivative = np.abs(np.diff(np.diff(inertia_scores)))

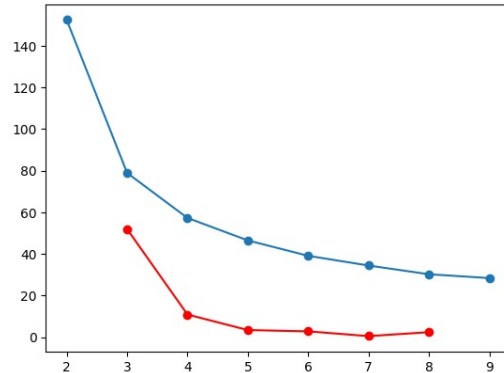
plt.plot(k_values, inertia_scores)
plt.scatter(k_values, inertia_scores)
plt.plot(k_values[1:-1], module_of_second_derivative, color='red')
plt.scatter(k_values[1:-1], module_of_second_derivative, color='red')
plt.show()
```

Name :

Rahimasiddiqui(2303.KHI.DEG.030)

Peer Name:

Osama Abdul Razzak (2303.KHI.DEG.029)



And from plot, we observe the sharp change appears at $k=3$, so it means, we select optimal number of clusters

After that, we apply PCA (Principal component Analysis) to reduce the datasets to two dimension using 'PCA()' function

```
▼ Apply PCA to reduced the dataset to two dimension
```

```
[8]: # Apply PCA to reduce the dataset to 2 dimensions
pca = PCA(n_components=2)
x_reduced = pca.fit_transform(x)
x_reduced.shape
```

```
[8]: (150, 2)
```

Similarly, apply K-mean on reduced datasets

```
[29]: model_pca = KMeans(n_clusters=3, n_init=1, max_iter=100)
model_pca.fit(x_reduced)
```

```
[29]: ► KMeans
```

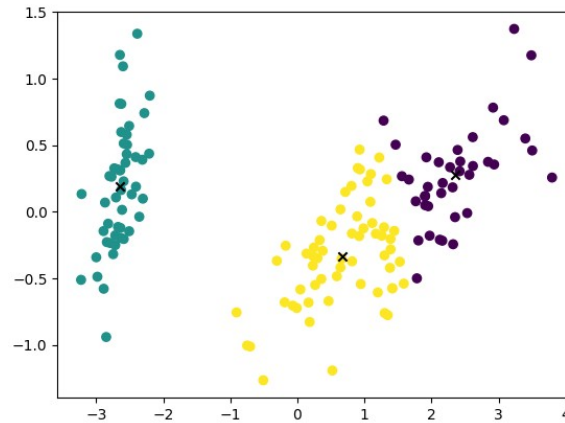
```
[27]: all_predictions_pca = model_pca.predict(x_reduced)
centroids_pca = model_pca.cluster_centers_
plt.scatter(x_reduced[:,0], x_reduced[:,1], c=all_predictions_pca)
plt.scatter(centroids_pca[:,0], centroids_pca[:,1], marker='x', color="black")
plt.show()
```

Name :

Rahimasiddiqui(2303.KHI.DEG.030)

Peer Name:

Osama Abdul Razzak (2303.KHI.DEG.029)



So, from both plots, we observe that first plot shows the clustering of Iris dataset based on all four features while the other plot shows the clustering on reduced dataset to two dimensions

Additionally, there is wide space between clusters in second plot, it means that we lost some information and on the other hand, clusters are very close it means almost all information is covered and some information is overlapped

Also, second plot show less overlapping of information as compared to first plot