I Worked with Muhammad Asim (2211-016-DEG-KHI)

In [1]:

```python
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

In [2]:

```python
iris = datasets.load_iris()
```

K-Means With-out PCA

In [3]:

```python
x_Kmeans = iris.data
y_Kmeans = iris.target
```

In [4]:

```python
x_Kmeans.shape
```

Out[4]:

```
(150, 4)
```

In [5]:

```python
model1 = KMeans(n_clusters=3, n_init=1, max_iter=100)
model1.fit(x_Kmeans)

all_predictions_Kmeans = model1.predict(x_Kmeans)
centroids_Kmeans = model1.cluster_centers_
```

In [6]:

```python
centroids_Kmeans
```

Out[6]:

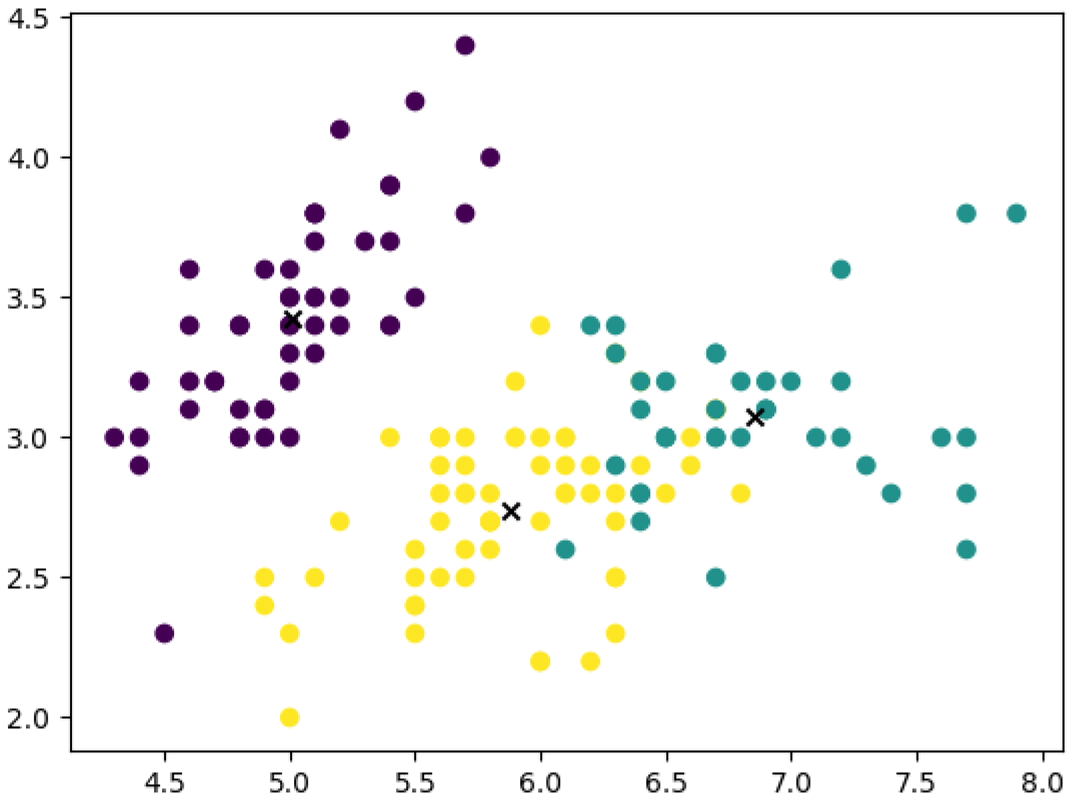
```
array([[5.006     , 3.428     , 1.462     , 0.246     ],
       [6.85384615, 3.07692308, 5.71538462, 2.05384615],
       [5.88360656, 2.74098361, 4.38852459, 1.43442623]])
```

In [7]:

```python
plt.scatter(x_Kmeans[:,0], x_Kmeans[:,1], c=all_predictions_Kmeans)
plt.scatter(centroids_Kmeans[:,0], centroids_Kmeans[:,1], marker='x', color='
plt.show
```

Out[7]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Applying PCA

In [8]:

```
pca = PCA(n_components=2)
x_reduced = pca.fit_transform(x_Kmeans)

x_reduced.shape
```

Out[8]:

```
(150, 2)
```

K-Means With PCA

In [9]:

```
model = KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(x_reduced)

all_predictions_PCA = model.predict(x_reduced)
centroids_PCA = model.cluster_centers_
```

In [10]:

```
centroids_PCA
```

Out[10]:

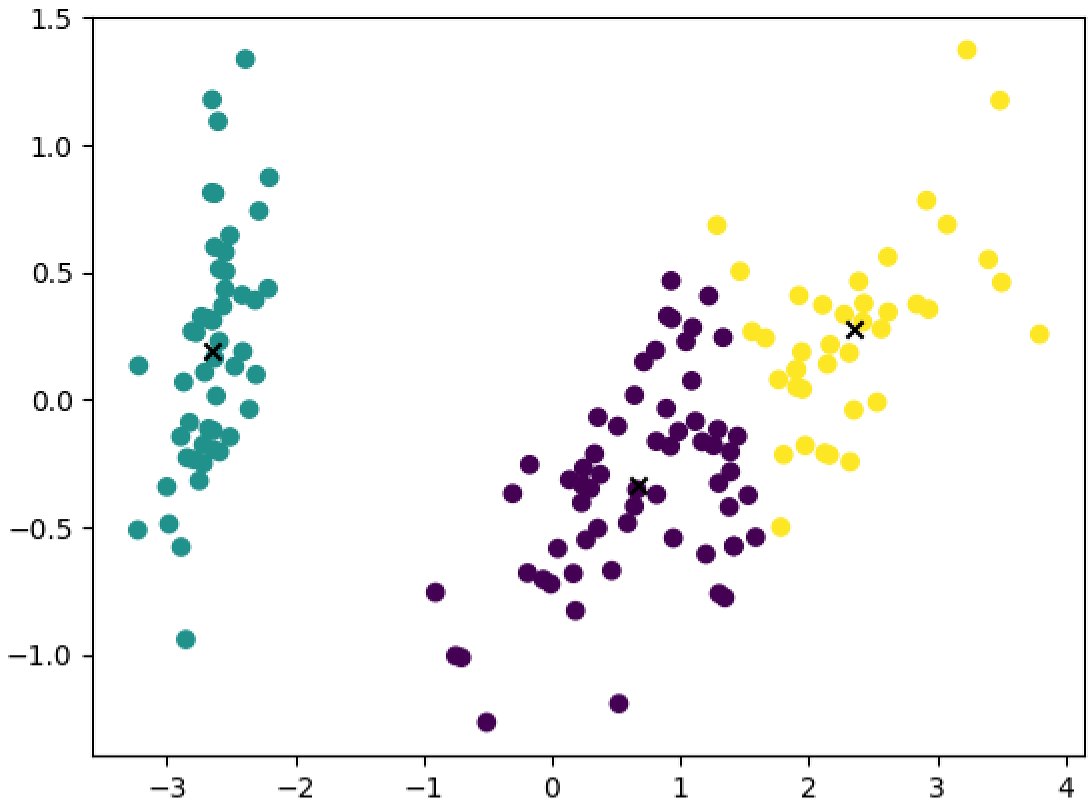
```
array([[ 0.66567601, -0.3316042 ],
       [-2.64241546,  0.19088505],
       [ 2.34652659,  0.27393856]])
```

In [11]:

```
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
```

Out[11]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Result Comparison

Calculating the mean of all distances from the centroid.

In [12]:

```
#K_Means
dist = np.linalg.norm(x_Kmeans - centroids_Kmeans[0,:],axis=1).reshape(-1,1)
dist.mean()
```

Out[12]:

```
2.8464251868829016
```

In [13]:

```
#PCA
dist = np.linalg.norm(x_reduced - centroids_PCA[0,:],axis=1).reshape(-1,1)
dist.mean()
```

Out[13]:

```
1.8602476484380313
```