Graded Assignment 3.3 K-means Cluster.

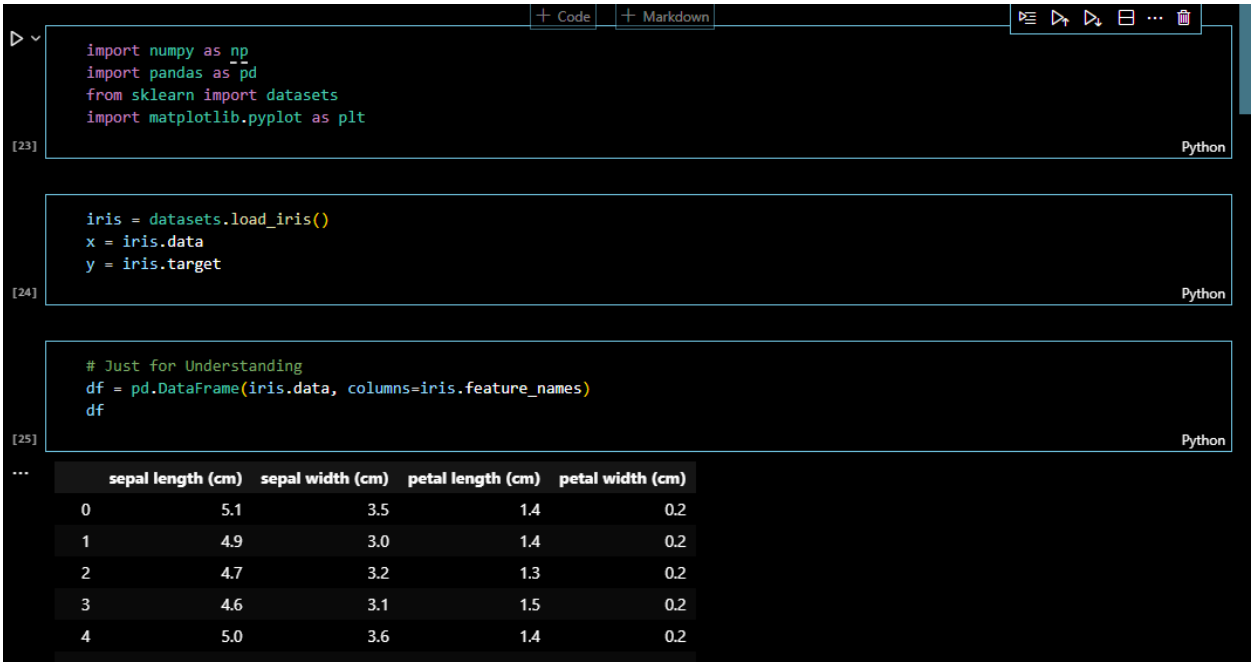
Submitted by:

Saif Ur Rehman (2303.KHI.DEG.007)

Ali Nasir (2302.KHI.DEG.012)

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

Import all the required libraries, then we imported all the dataset required for the analysis.



```
import numpy as np
import pandas as pd
from sklearn import datasets
import matplotlib.pyplot as plt
```

[23] Python

```
iris = datasets.load_iris()
x = iris.data
y = iris.target
```

[24] Python

```
# Just for Understanding
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df
```

[25] Python

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Now we have implemented K means cluster on the dataset it can be shown that number of clusters is 3, n_init = 1 and max iteration is 100.

K-means

- n_clusters = The number of clusters to form as well as the number of centroids to generate.
- n_init = Number of times the k-means algorithm is run with different centroids
- max_iter = Maximum number of iterations of the k-means algorithm for a single run.

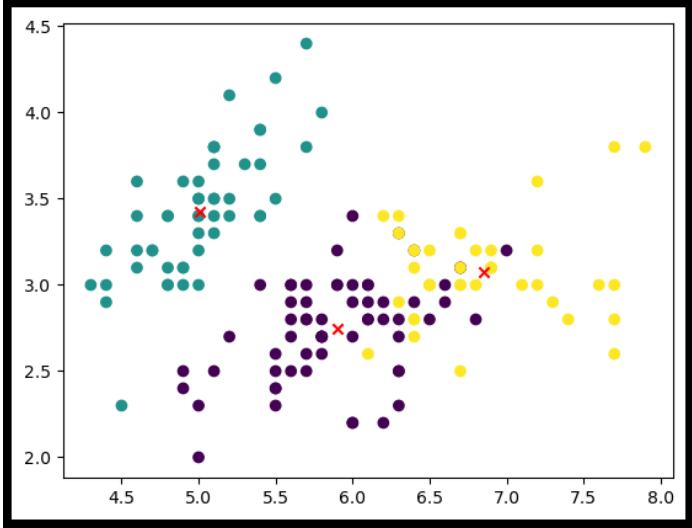
```
from sklearn.cluster import KMeans

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

all_predictions = model.predict(x)
centroids = model.cluster_centers_
```

Python

The scatter plot shows the impact of predictions on the model it is clearly seen that the clusters are overlapping with each other, hence we need to implement PCA.

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

Python



Here we have again uploaded the data and applied K means with PCA

K-means with PCA

```
from sklearn.decomposition import PCA

iris = datasets.load_iris()
x2 = iris.data
y2 = iris.target

pca = PCA(n_components=2)
x_pca = pca.fit_transform(x2)
```

[28] Python

```
x_pca.shape
```

[29] Python

```
(150, 2)
```

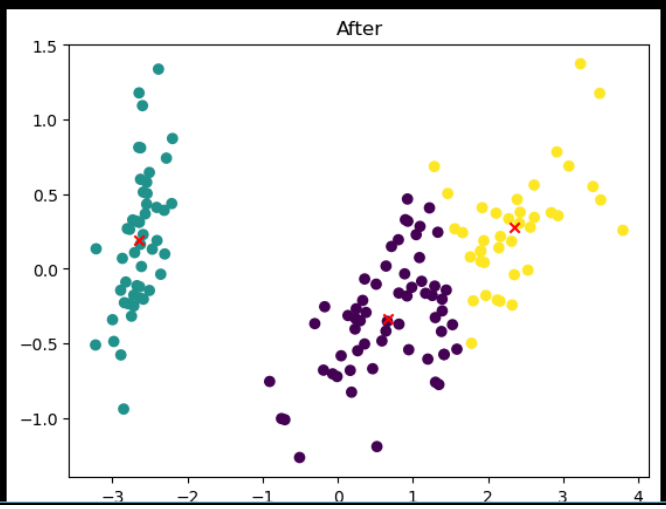
```
model = KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(x_pca)

all_predictions = model.predict(x_pca)
centroids = model.cluster_centers_
```

[30] Python

The results are showing the now the clusters are not overlapping with each other.

```
plt.scatter(x_pca[:,0], x_pca[:,1], c=all_predictions)
plt.scatter(centroids[:,0], centroids[:,1], marker='x', color="red")
plt.title('After')
plt.show()
```

[20] Python



If we see both of the images it is clear that the PCA has proven to resolve the overlapping of data as it reduce the dimension of our dataset

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

Python