UNIT 3.3 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)
Umaima Siddiqui (2023.KHI.DEG.033)

UNIT 3.3 GRADED ASSIGNMENT

Task:

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

Solution:

k-means clustering on the Iris dataset before applying PCA:

Unsupervised.ipynb

Code

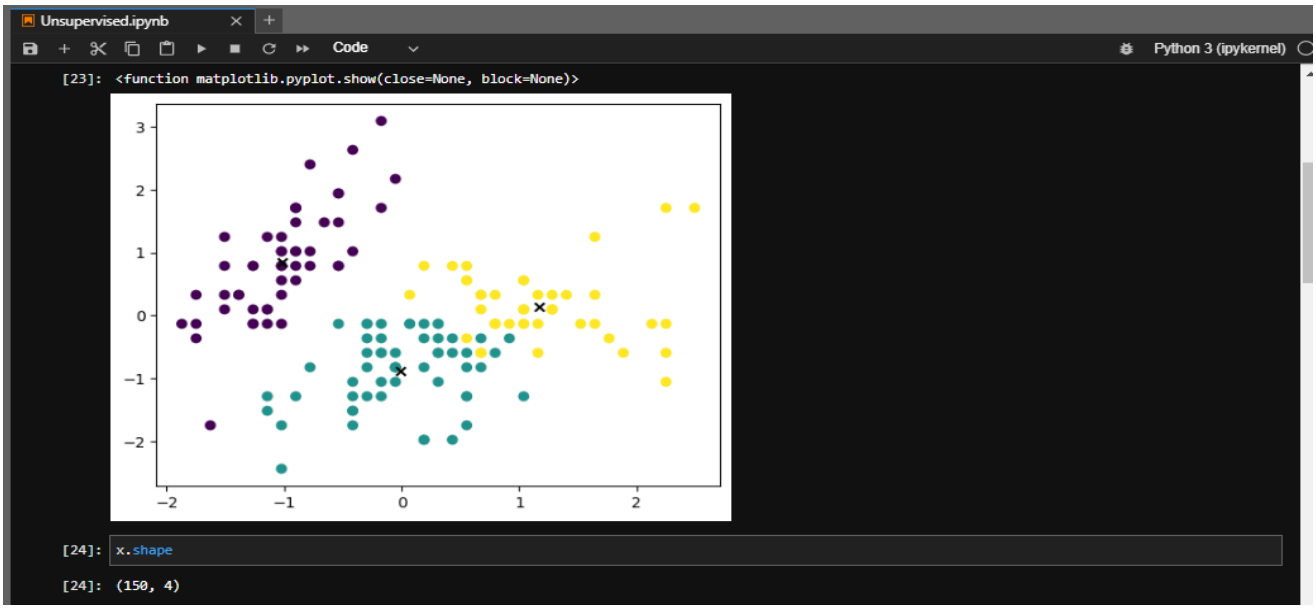
Python 3 (ipykernel)

[3]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn import datasets
```

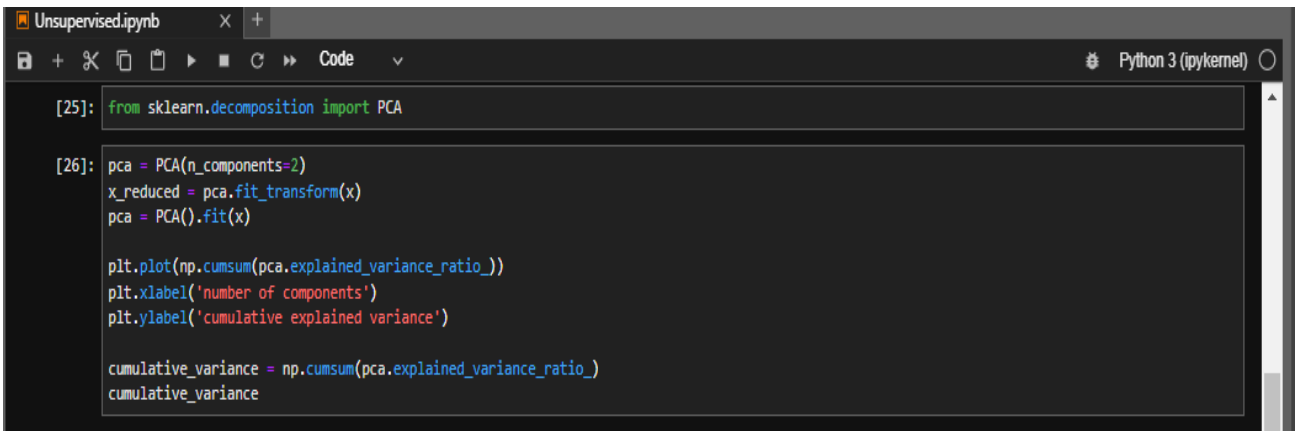
[6]:
```
iris = datasets.load_iris()
scaler = StandardScaler()
x = scaler.fit_transform(iris.data)
y = iris.target
```

[33]:
```
from sklearn.cluster import KMeans

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

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

[23]:
```
plt.scatter(x[:,0], x[:,1], c=prediction_before_pca)
plt.scatter(centroids[:,0], centroids[:,1], marker='x', color="black")
plt.show
```

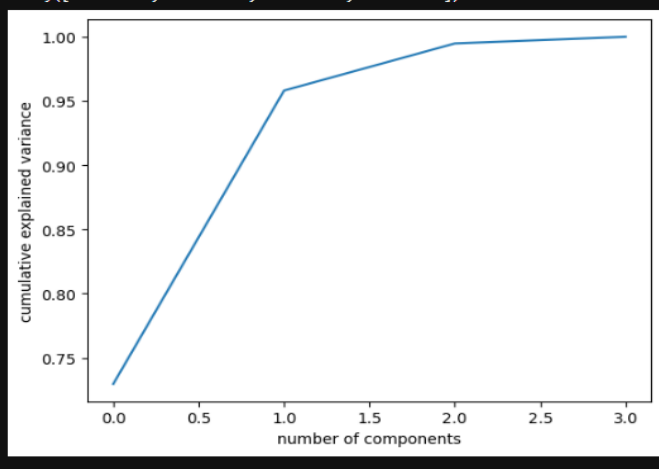


PCA (Principal component analysis):



The shape is reduced from (150, 4) to (150, 2) after applying PCA.

```
[26]: array([0.72962445, 0.95813207, 0.99482129, 1.        ])
```



k-means clustering on the Iris dataset after applying PCA:

Unsupervised.ipynb

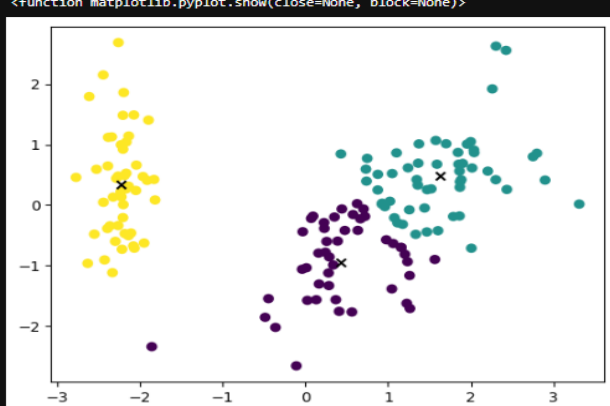
Code | Python 3 (ipykernel)

```
[27]: model = KMeans(n_clusters=3, n_init=1, max_iter=100)  # n_init --> Initialization, it means that if we set this to 10 then it would select cent
      model.fit(x_reduced)

      prediction_after_pca = model.predict(x_reduced)
      centroids = model.cluster_centers_

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

Unsupervised.ipynb

Code | Python 3 (ipykernel)

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




```
[29]: x_reduced.shape

[29]: (150, 2)
```

We can see the clear difference between the clusters by comparing both results. After applying PCA we get more well-structured clusters.

Adjusted Rand Index:

We can compute adjusted Rand Index to compare the results obtained from k-means clustering before and after applying PCA. The ARI ranges from -1 to 1, where a value of 1 indicates perfect agreement between the two clusterings, and a value of 0 indicates random clustering.

```
[31]: from sklearn.metrics import adjusted_rand_score
      adjusted_rand_index = adjusted_rand_score(prediction_before_pca, prediction_after_pca)
      print(f"Adjusted Rand Index between the original and PCA-reduced datasets: {adjusted_rand_index:.2f}")

      Adjusted Rand Index between the original and PCA-reduced datasets: 0.78
```