Talha & Moiz Assignment 3.3

May 25, 2023

1 Talha Khan (2303.009.KHI.DEG)

2 Mohammad Moiz Khan(2303.KHI.DEG.022)

[13]:
```
import numpy as np
import pandas as pd
from sklearn import datasets
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import adjusted_rand_score
import warnings
warnings.filterwarnings('ignore')
```

3 Loading the Iris Datasets

[14]:
```
# load dataset
iris = datasets.load_iris()
scaler = StandardScaler()
x = scaler.fit_transform(iris.data)
y = iris.target
```

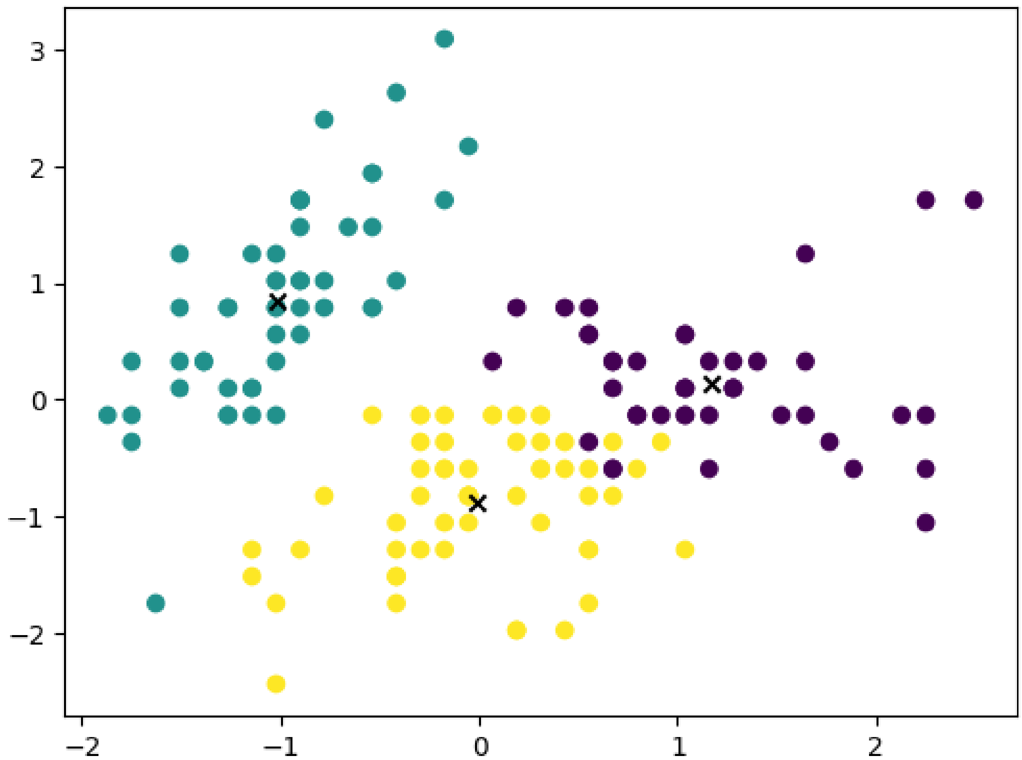
4 Applying K-Means CLustering on Datasets Before Reduction

[15]:
```
# model implementation
model = KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(x)
```

[15]:
```
KMeans(max_iter=100, n_clusters=3, n_init=1)
```

[16]:
```
predictions_before_reduced = model.predict(x)
centroids = model.cluster_centers_
```

[17]:
```
# plot clusters
plt.scatter(x[:,0], x[:,1], c=predictions_before_reduced)
plt.scatter(centroids[:,0], centroids[:,1], marker='x', color="black")
plt.show()
```



[18]:
```
x.shape
```

[18]: (150, 4)

5 Applying PCA (Principal Componenet Analysis) on Datasets

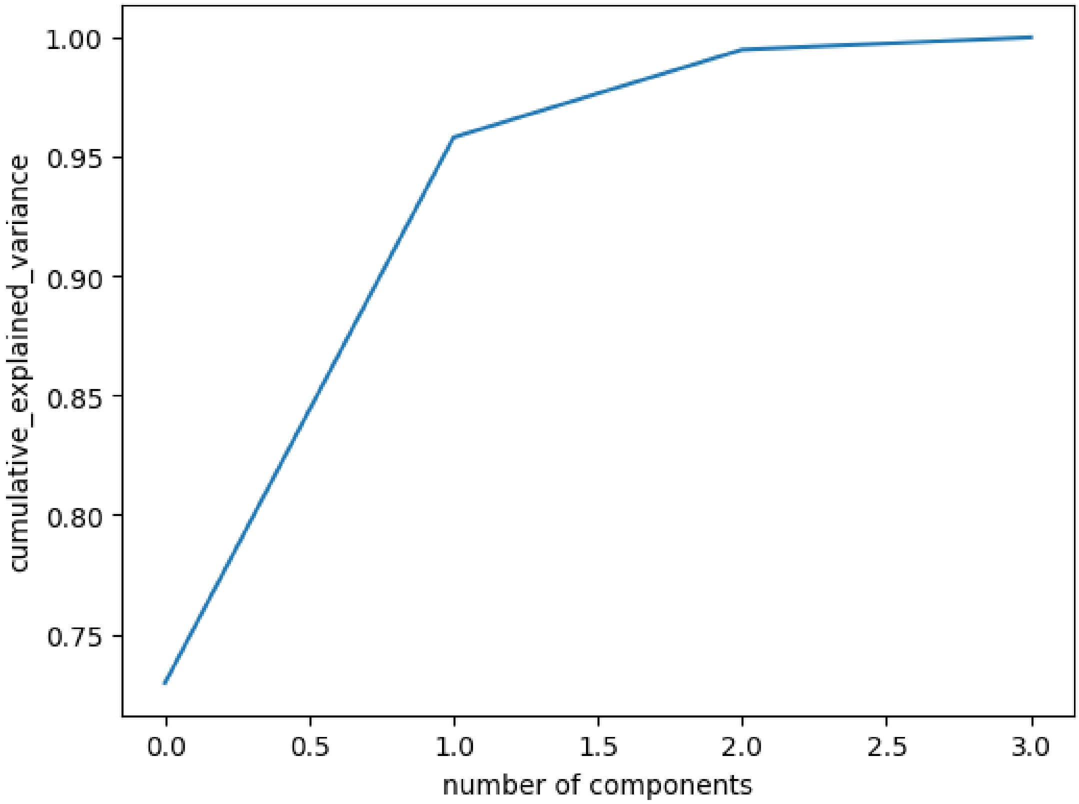
[19]:
```
# dimensionality reduction using PCA
pca = PCA(n_components=2)
x_reduced = pca.fit_transform(x)
pca = PCA().fit(x)
```

[20]:
```
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel("number of components")
plt.ylabel("cumulative_explained_variance")

cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
```

```
cumulative_variance
```

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



6 We can see clearly that shape reduction from (150,4) to (150,2) occurs after applying PCA.

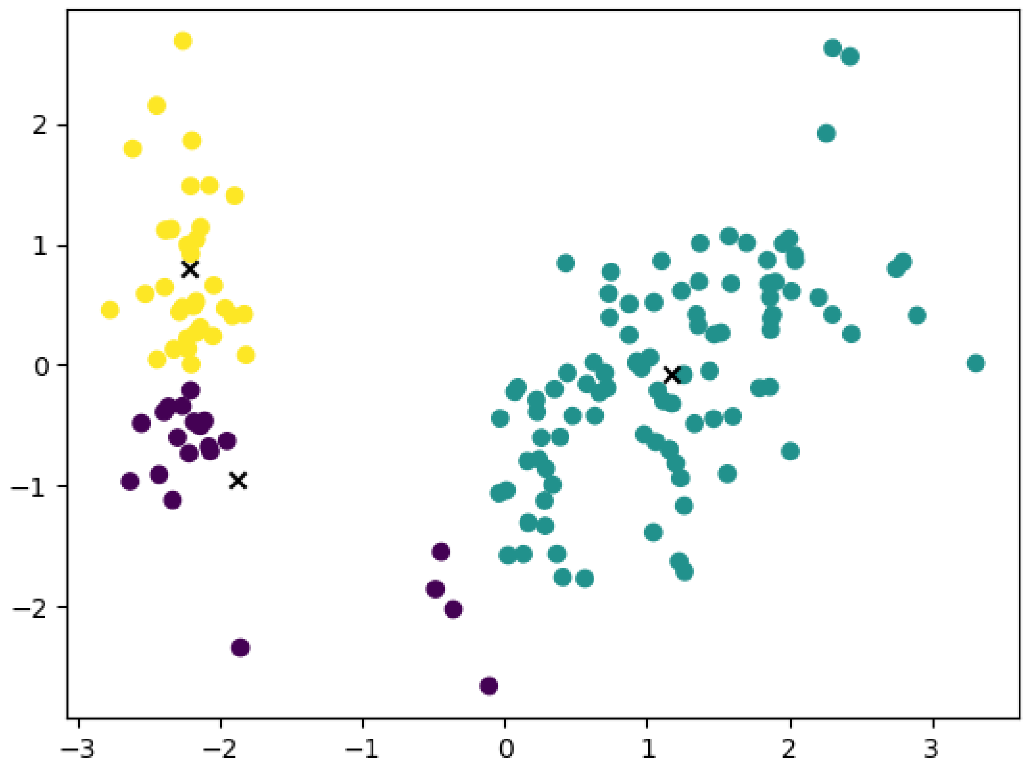
7 Applying K-Means Clustering on Datasets after Reduction

[21]:
```
# model implementation on reduced dimensioned data
model = KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(x_reduced)
```

[21]: KMeans(max_iter=100, n_clusters=3, n_init=1)

[22]:
```
predictions_after_reduced = model.predict(x_reduced)
centroids_PCA = model.cluster_centers_
```

[23]:
```
# plot clusters
plt.scatter(x_reduced[:,0], x_reduced[:,1], c=predictions_after_reduced)
plt.scatter(centroids_PCA[:,0], centroids_PCA[:,1], marker='x', color="black")
plt.show()
```



[24]:
```
x_reduced.shape
```

[24]:
```
(150, 2)
```

8 It is observed that the PCA reduces the dimensionality of the data and removes the less informative dimensions. As a result, the remaining dimensions contain more relevant information and are more suitable for clustering.

[25]:
```
adjusted_rand_index =␣
  ↪adjusted_rand_score(predictions_before_reduced,predictions_after_reduced)
print(f"Adjusted rand index between original and PCA reduced datasets:␣
  ↪{adjusted_rand_index:.2f}")
```

```
Adjusted rand index between original and PCA reduced datasets: 0.43
```

9 The function “adjusted_rand_score” calculate the adjusted Rand index. This function takes the two sets of predictions as inputs and returns a value that represents the similarity between them. The adjusted Rand index ranges from -1 to 1, where 1 indicates perfect agreement, 0 indicates random labeling, and negative values indicate disagreement.