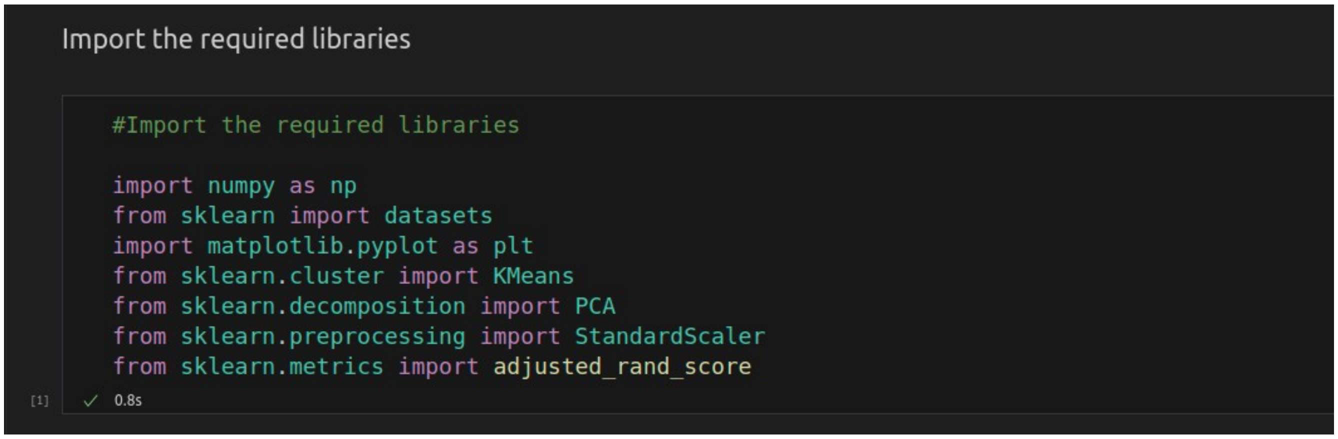
Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

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.

Solution

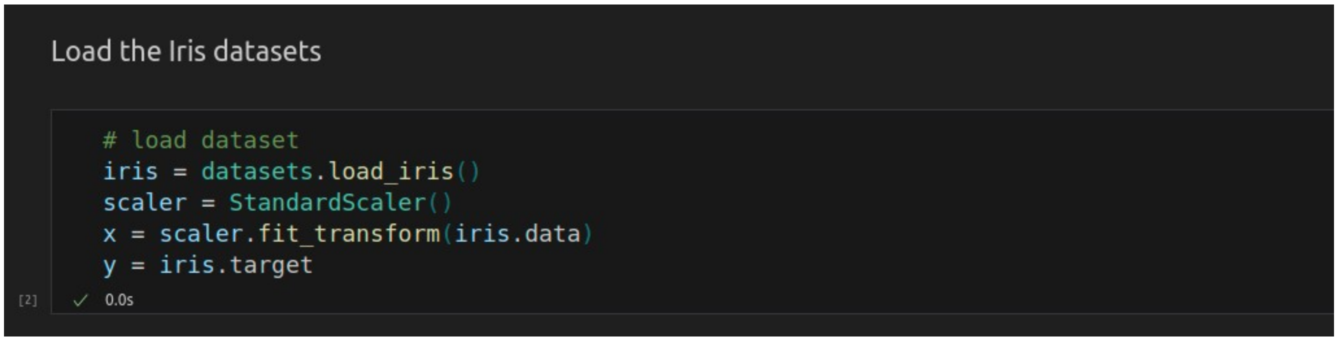


Import the required libraries

```
#Import the required libraries

import numpy as np
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
```

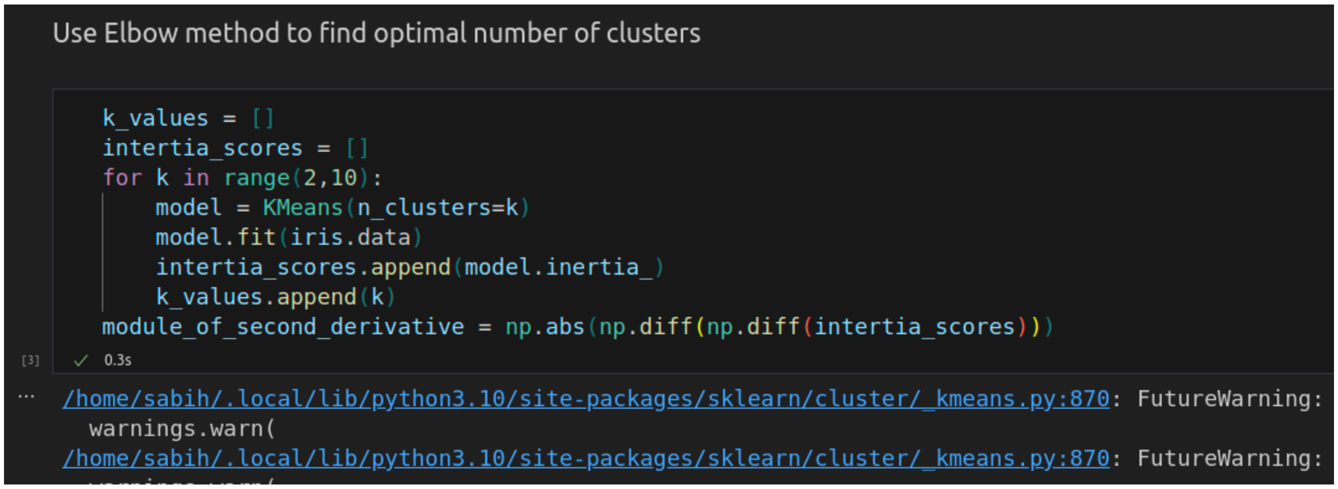
[1] ✓ 0.8s



Load the Iris datasets

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

[2] ✓ 0.0s



Use Elbow method to find optimal number of clusters

```
k_values = []
intertia_scores = []
for k in range(2,10):
    model = KMeans(n_clusters=k)
    model.fit(iris.data)
    intertia_scores.append(model.inertia_)
    k_values.append(k)
module_of_second_derivative = np.abs(np.diff(np.diff(intertia_scores)))
```

[3] ✓ 0.3s

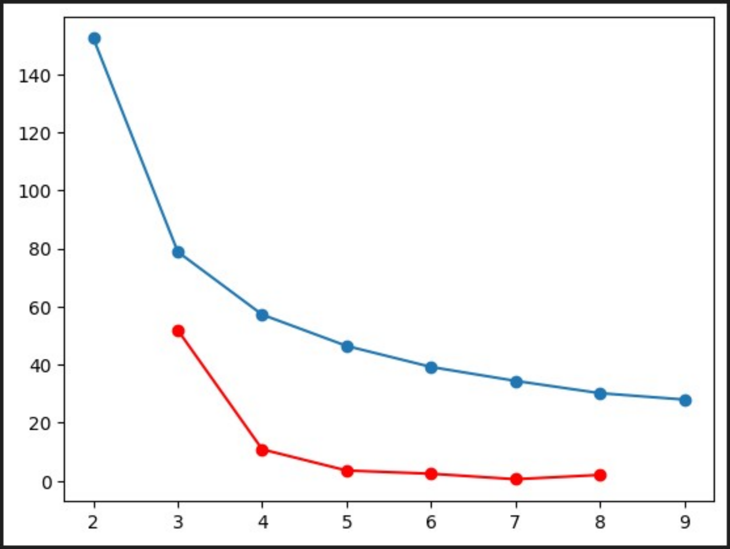
```
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/home/sabih/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
```

Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

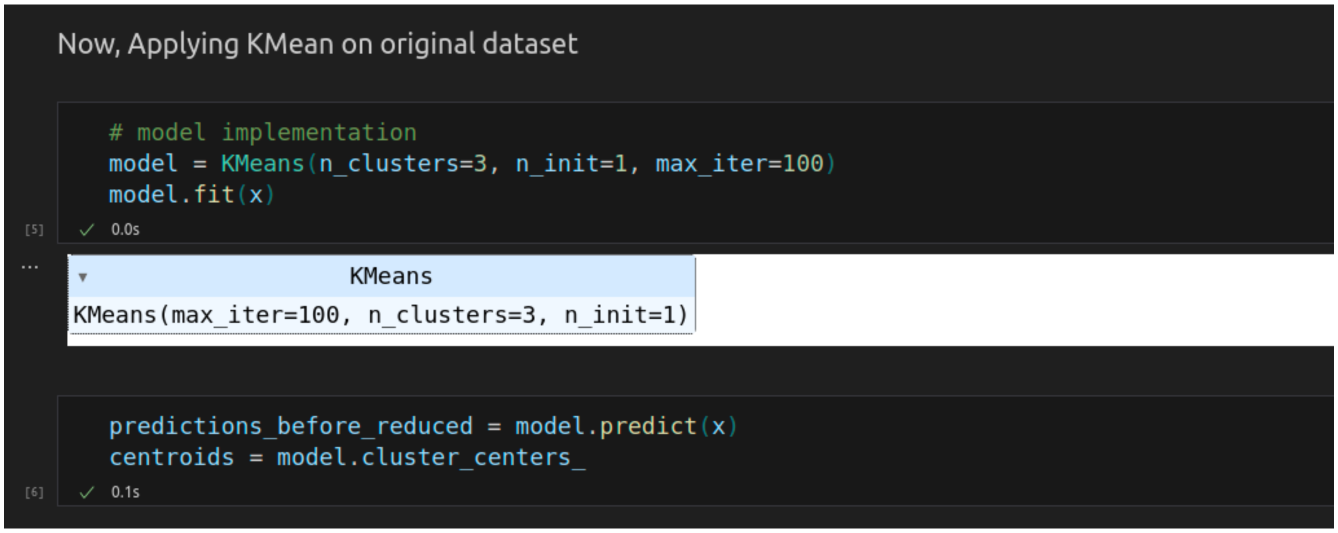
Show the Elbow graph

```
plt.plot(k_values, intertia_scores)
plt.scatter(k_values, intertia_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()
```

[4] ✓ 0.2s



And from plot it is observed the sharp curves appear at k=3



Now, Applying KMean on original dataset

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

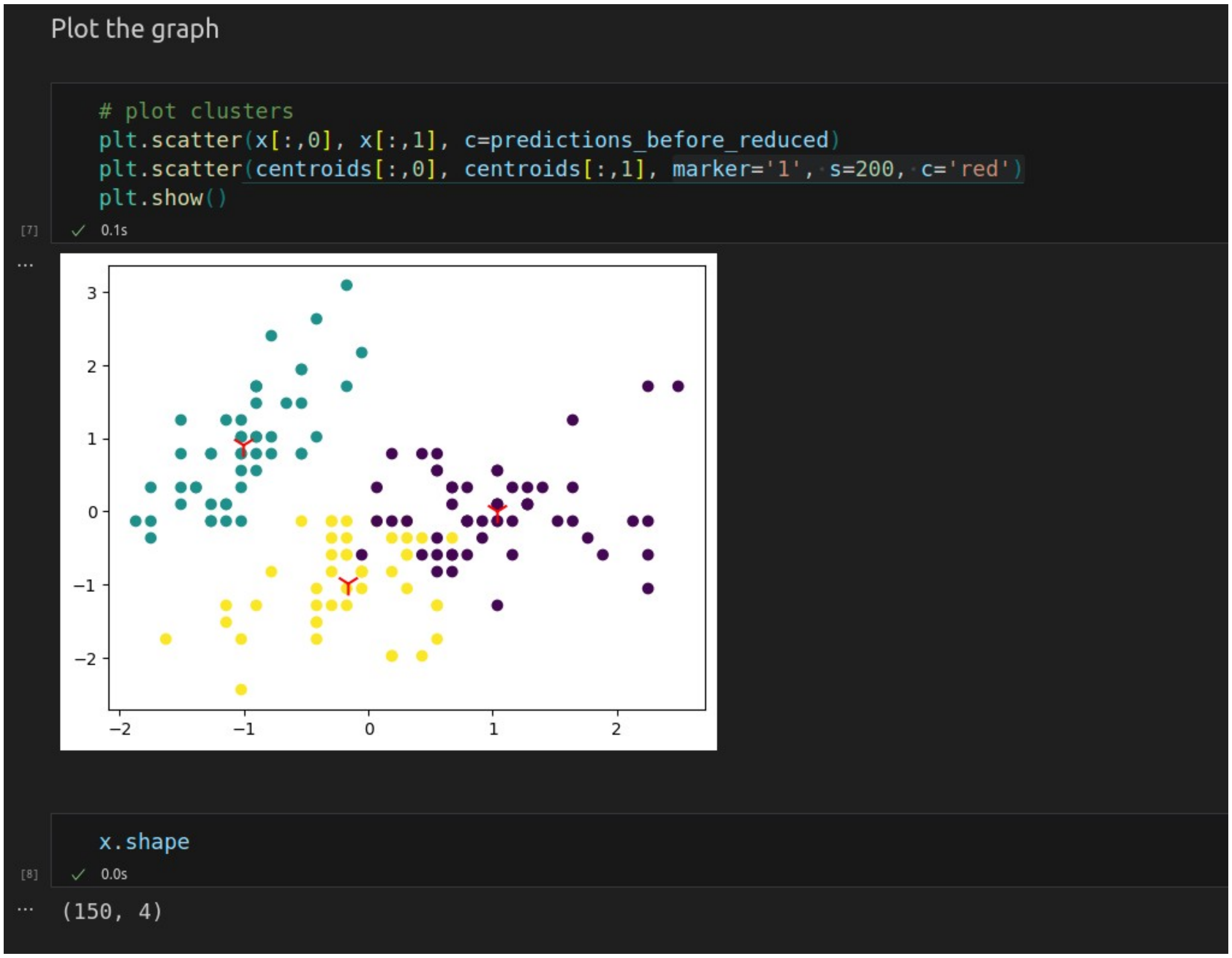
[5] ✓ 0.0s

```
KMeans
KMeans(max_iter=100, n_clusters=3, n_init=1)
```

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

[6] ✓ 0.1s

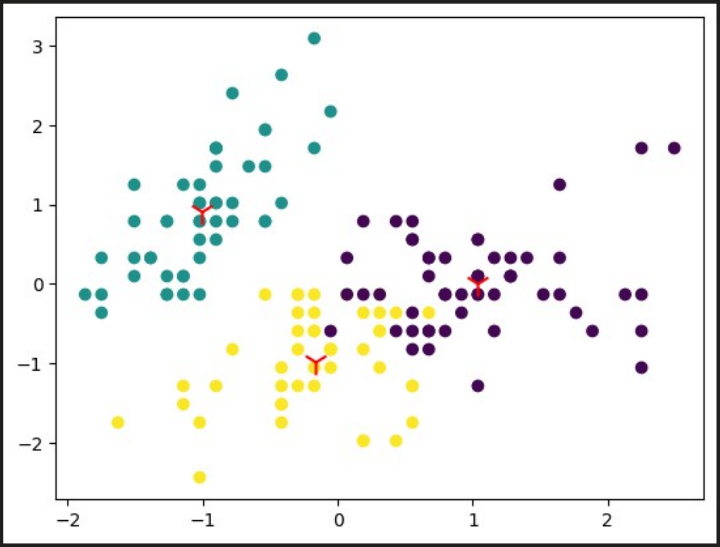
Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)



Plot the graph

```
# plot clusters
plt.scatter(x[:,0], x[:,1], c=predictions_before_reduced)
plt.scatter(centroids[:,0], centroids[:,1], marker='1', s=200, c='red')
plt.show()
```

[7] ✓ 0.1s

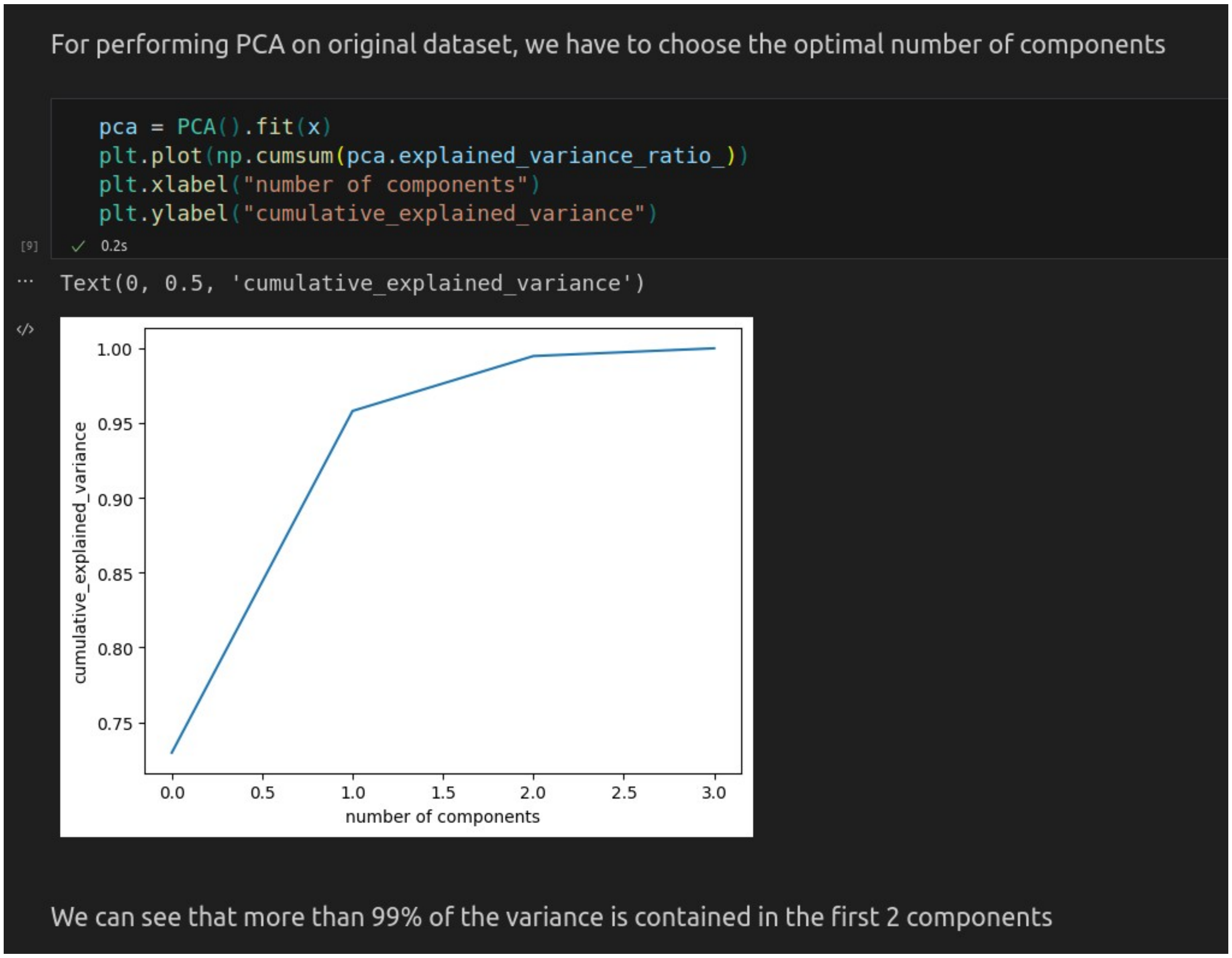


```
x.shape
```

[8] ✓ 0.0s

```
(150, 4)
```

Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

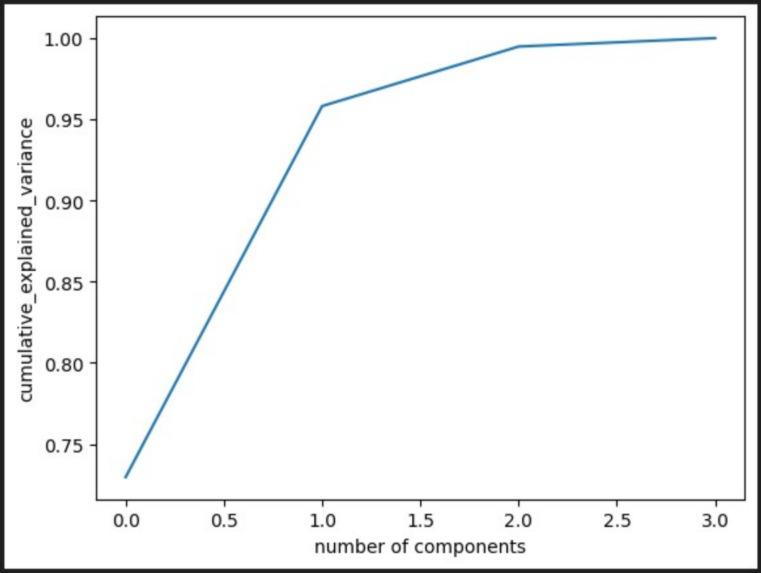


For performing PCA on original dataset, we have to choose the optimal number of components

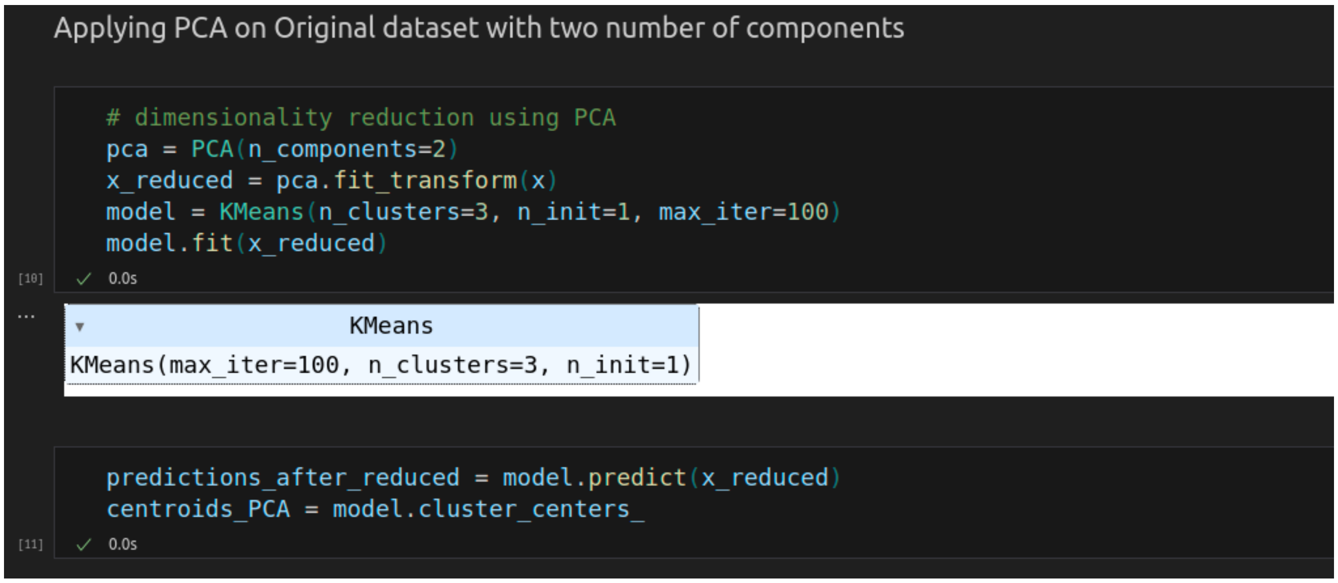
```
pca = PCA().fit(x)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel("number of components")
plt.ylabel("cumulative_explained_variance")
```

```
[9]  ✓ 0.2s
```

```
Text(0, 0.5, 'cumulative_explained_variance')
```



We can see that more than 99% of the variance is contained in the first 2 components



Applying PCA on Original dataset with two number of components

```
# dimensionality reduction using PCA
pca = PCA(n_components=2)
x_reduced = pca.fit_transform(x)
model = KMeans(n_clusters=3, n_init=1, max_iter=100)
model.fit(x_reduced)
```

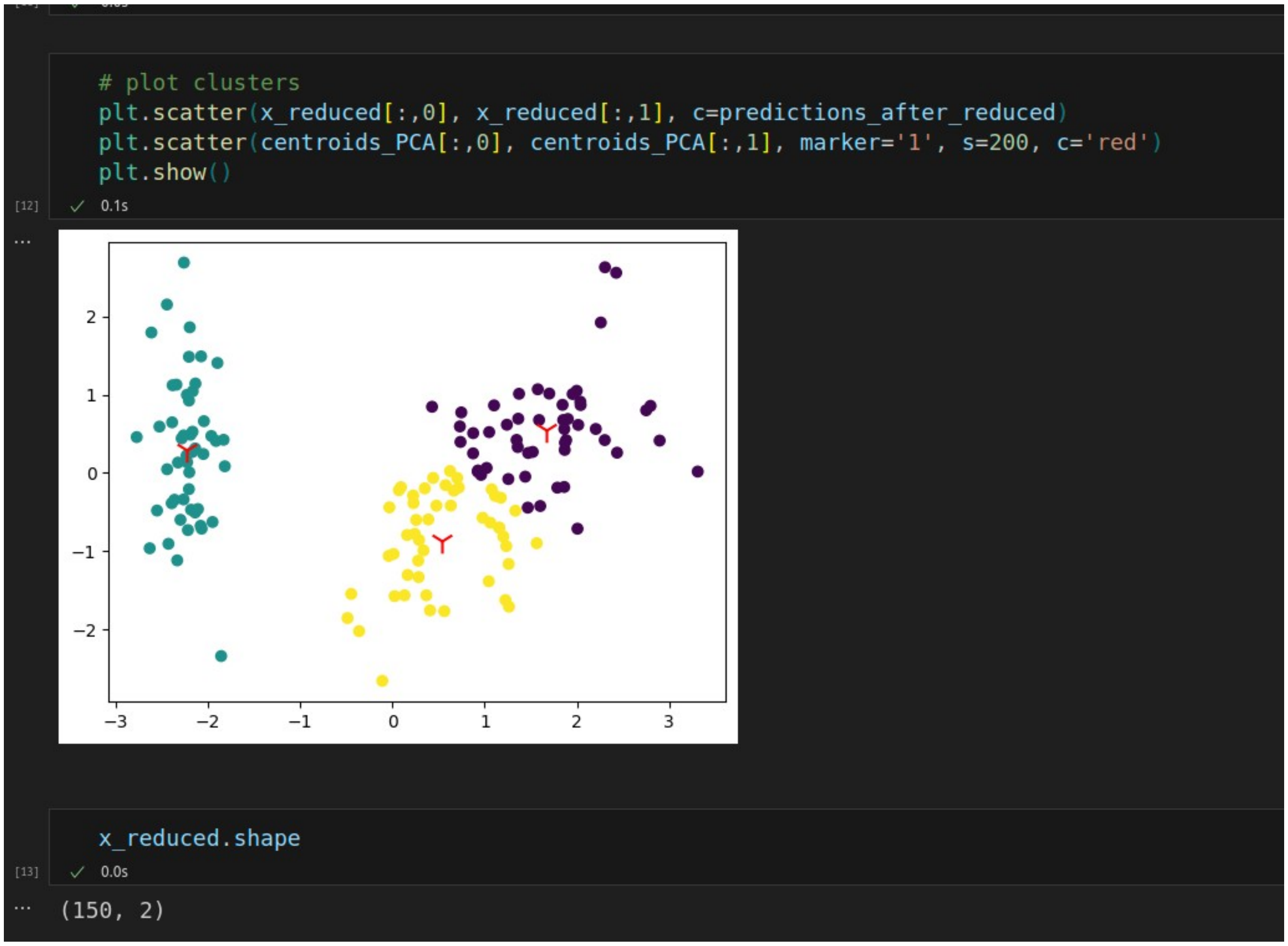
```
[10]  ✓ 0.0s
```

```
▾                KMeans
KMeans(max_iter=100, n_clusters=3, n_init=1)
```

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

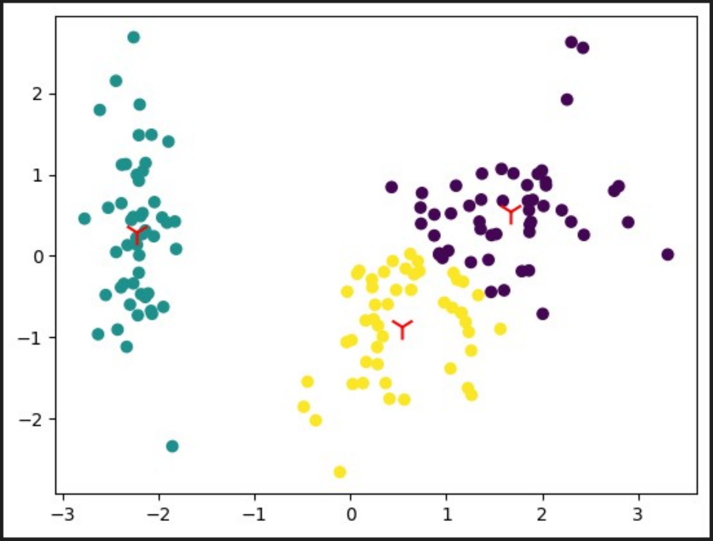
```
[11]  ✓ 0.0s
```

Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)



```
# plot clusters
plt.scatter(x_reduced[:,0], x_reduced[:,1], c=predictions_after_reduced)
plt.scatter(centroids_PCA[:,0], centroids_PCA[:,1], marker='1', s=200, c='red')
plt.show()
```

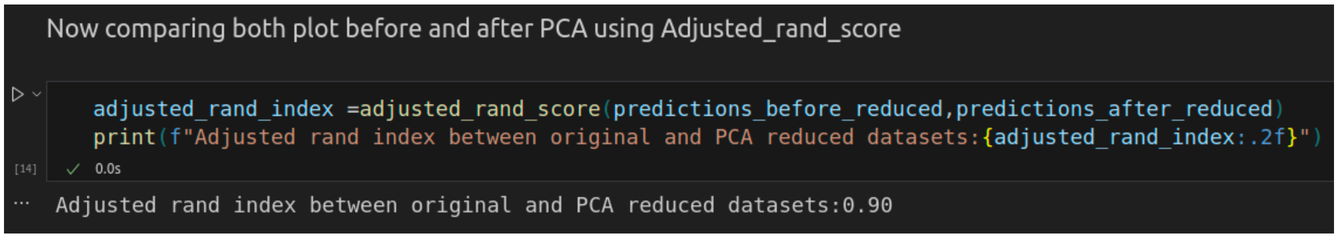
[12] ✓ 0.1s



```
x_reduced.shape
```

[13] ✓ 0.0s

```
(150, 2)
```



Now comparing both plot before and after PCA using Adjusted_rand_score

```
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}")
```

[14] ✓ 0.0s

```
Adjusted rand index between original and PCA reduced datasets:0.90
```