Silhouette Method for optimal value of k in KMeans

Silhouette analysis can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1].

Silhouette coefficients near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

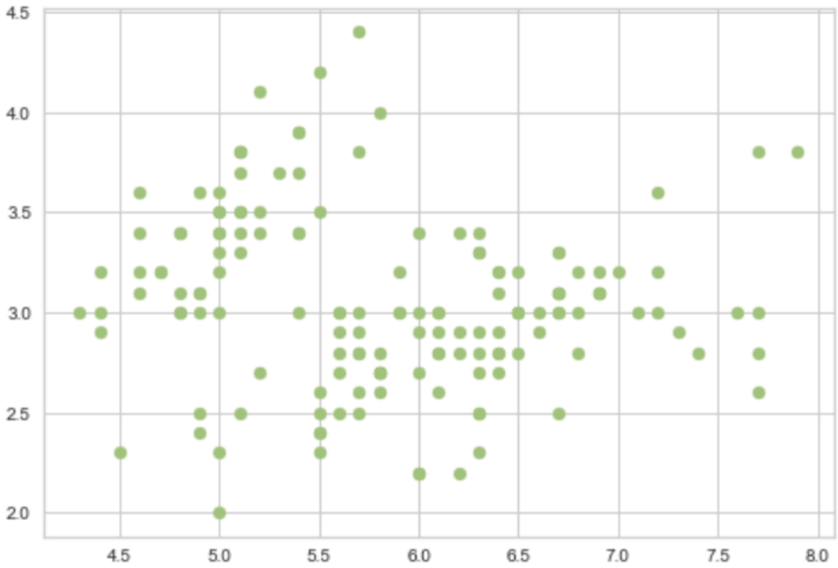
In this example the silhouette analysis is used to choose an optimal value for n_clusters.

Code:

Load Iris dataset for clustering purpose and import matplotlib, KMeans as follows,

```
from sklearn.datasets import load_iris

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

data= load_iris()

x=data.data

y=data.target

#plotting the dataset

plt.scatter(x[:, 0], x[:, 1],c='g');
```

Output:

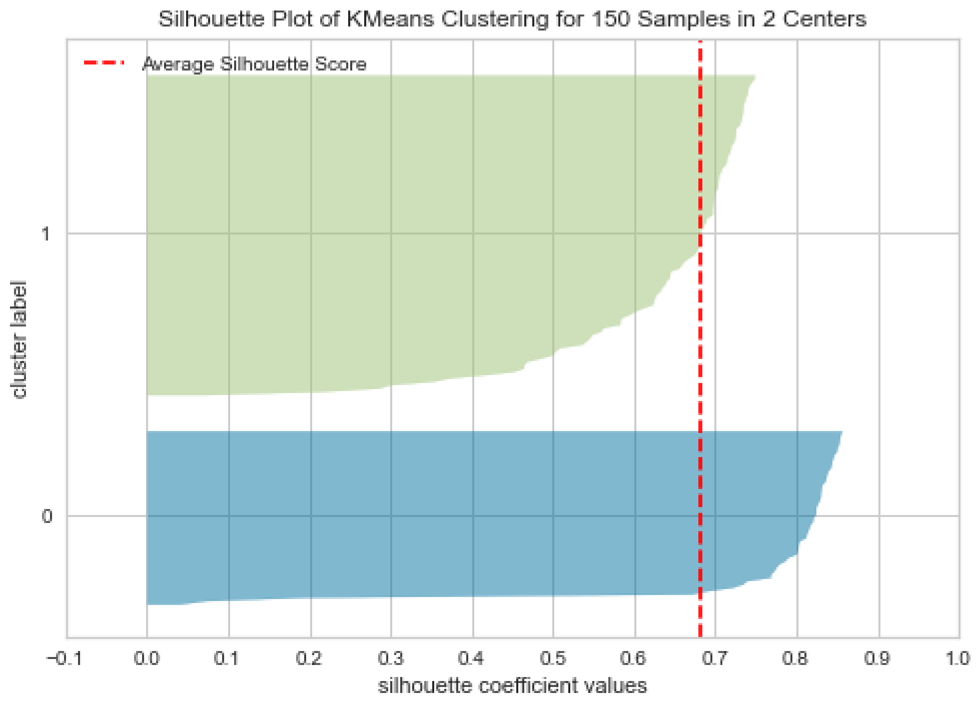


In this section we train our model(kmeans) on different clusters and calculate silhouette score, plotted using silhouette visualization function as follows,

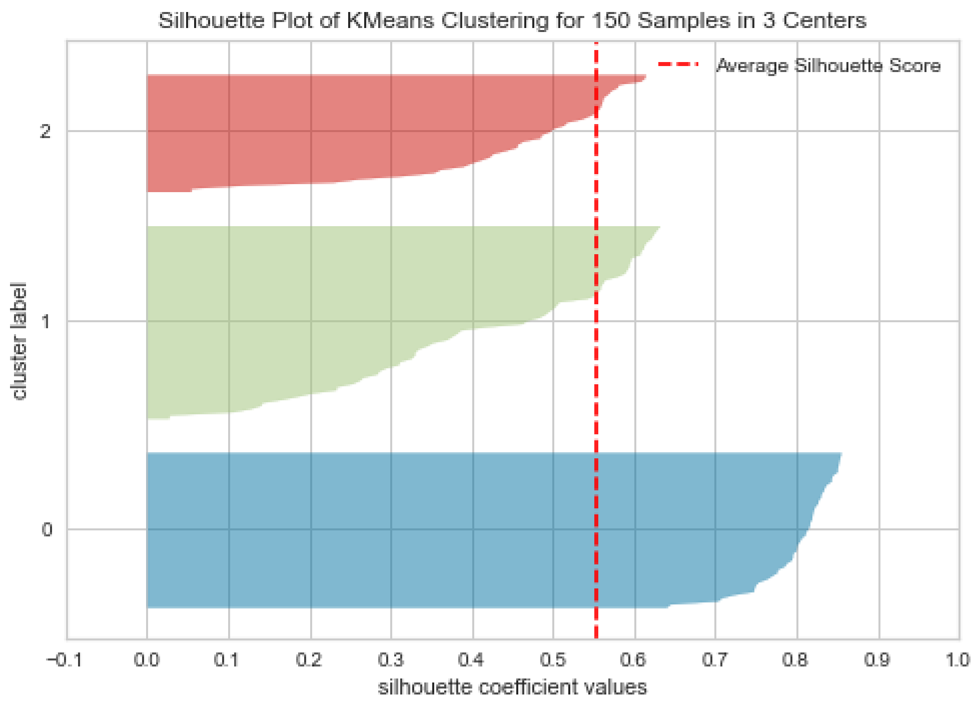
```
from sklearn.metrics import silhouette_score

k = [2, 3, 4, 5]


for n_clusters in k:

    model = KMeans(n_clusters=n_clusters, random_state=10)

    cluster_labels = model.fit_predict(x)



     # The silhouette_score gives the average value for all the
samples.

     # This gives a perspective into the density and separation
of the formed

    # clusters

                 visualizer = SilhouetteVisualizer(model,
colors='yellowbrick')
```

```
    visualizer.fit(x)        # Fit the data to the visualizer

    visualizer.show()

    silhouette_avg = silhouette_score(x, cluster_labels)

    print("For n_clusters =", n_clusters,"The average
silhouette_score is :", silhouette_avg)
```

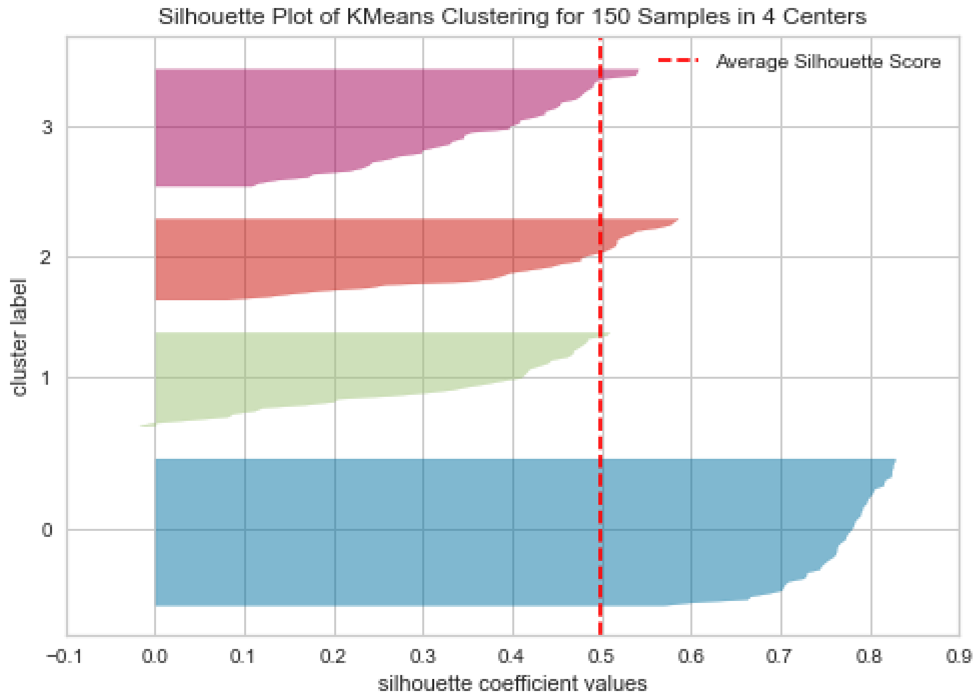
Output:



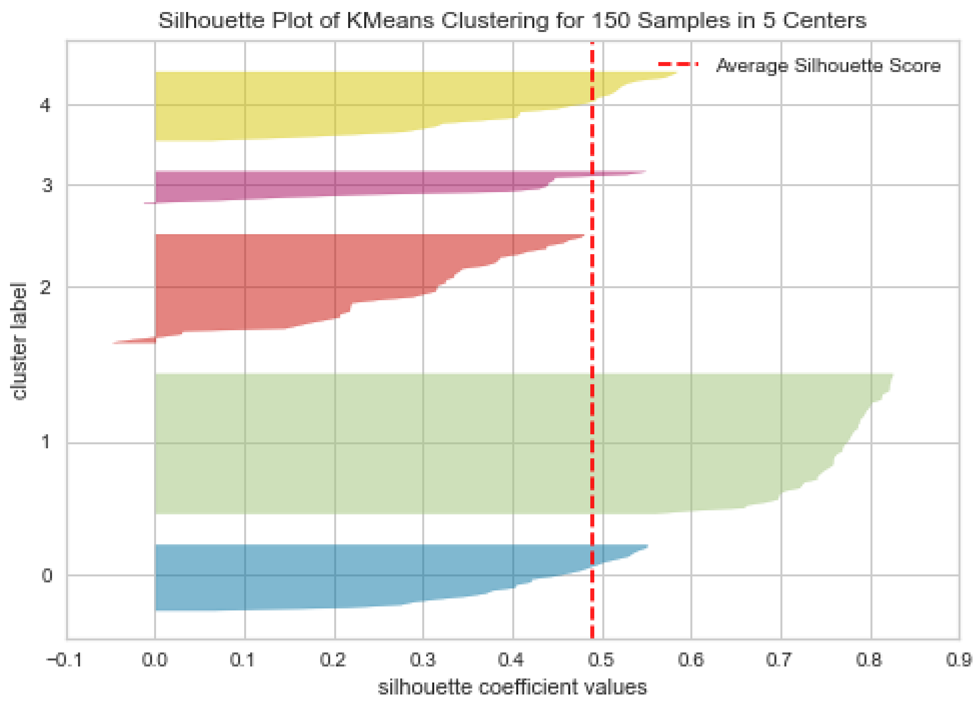
For n_clusters = 2 The average silhouette_score is : 0.681046169211746



For n_clusters = 3 The average silhouette_score is : 0.5528190123564091



For n_clusters = 4 The average silhouette_score is : 0.4980505049972866



Here is the Silhouette analysis done on the above plots to select an optimal value for n_clusters.

The value of n_clusters as 3 looks to be suboptimal for the given data due to the following reasons:

- Presence of clusters with below-average silhouette scores
- Wide fluctuations in the size of the silhouette plots.

The value of 3 for n_clusters looks to be the optimal one. The silhouette score for each cluster is above average silhouette scores. Also, the fluctuation in size is similar. The thickness of the silhouette plot representing each cluster also is a deciding point. For the plot with n_cluster 3 (top right), the thickness is more uniform than the plot with n_cluster as 2 ,4, 5 with one cluster thickness much more than the other. Thus, one can select the optimal number of clusters as 3.