```python
# -*- coding: utf-8 -*-
"""
Created on Sat Apr  7 15:32:15 2018

@author: cml
"""

from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster import hierarchy
from sklearn.metrics.pairwise import euclidean_distances
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram

#Courtesy of https://raw.githubusercontent.com/scikit-learn/scikit-learn/
#70cf4a676caa2d2dad2e3f6e4478d64bcb0506f7/examples/cluster/
#plot_hierarchical_clustering_dendrogram.py
def plot_dendrogram(model, **kwargs):

    # Children of hierarchical clustering
    children = model.children_

    # Distances between each pair of children
    # Since we don't have this information, we can use a uniform one for plotting
    distance = np.arange(children.shape[0])

    # The number of observations contained in each cluster level
    no_of_observations = np.arange(2, children.shape[0]+2)

    # Create linkage matrix and then plot the dendrogram
    linkage_matrix = np.column_stack([children, distance,
    no_of_observations]).astype(float)

    # Plot the corresponding dendrogram
    dendrogram(linkage_matrix, **kwargs)

np.random.seed(2)
X = np.random.rand(20,20)

print(euclidean_distances(X, X))

clusterComplete = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
                                  compute_full_tree='auto',
                                  linkage='complete').fit(X)

plt.title('Hierarchical Clustering Dendrogram (complete)')
plot_dendrogram(clusterComplete, labels=clusterComplete.labels_)
plt.show()

print("Labels: ", clusterComplete.labels_)
print("No. leaves: ", clusterComplete.n_leaves_)
print("No. components: ", clusterComplete.n_components_)
print("No. clusters: ", clusterComplete.n_clusters)

clusterWard = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
                                  compute_full_tree='auto',
                                  linkage='ward').fit(X)

plt.title('Hierarchical Clustering Dendrogram (ward)')
plot_dendrogram(clusterWard, labels=clusterWard.labels_)
plt.show()

print("Labels: ", clusterWard.labels_)

clusterAverage = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
                                  compute_full_tree='auto',
```

```
                                    linkage='average').fit(X)

plt.title('Hierarchical Clustering Dendrogram (average)')
plot_dendrogram(clusterAverage, labels=clusterAverage.labels_)
plt.show()

print("Labels: ", clusterAverage.labels_)
```