Data_science_project_on_Hierarchical_clustering

June 23, 2021

[1]:
```python
#importing the required modules
import numpy as np
import pandas as pd
#import pandas_profiling as pp
from matplotlib import pyplot as plt
import streamlit as st
%matplotlib inline
```

[2]:
```
!wget -O chemical composion of ceramics.csv http://archive.ics.uci.edu/ml/
↪machine-learning-databases/00583/Chemical%20Composion%20of%20Ceramic.csv
```

```
--2021-06-22 12:22:34--  http://composion/
Resolving composion (composion)… failed: nodename nor servname provided, or
not known.
wget: unable to resolve host address ‘composion’
--2021-06-22 12:22:34--  http://of/
Resolving of (of)… failed: nodename nor servname provided, or not known.
wget: unable to resolve host address ‘of’
--2021-06-22 12:22:34--  http://ceramics.csv/
Resolving ceramics.csv (ceramics.csv)… failed: nodename nor servname provided,
or not known.
wget: unable to resolve host address ‘ceramics.csv’
--2021-06-22 12:22:34--  http://archive.ics.uci.edu/ml/machine-learning-
databases/00583/Chemical%20Composion%20of%20Ceramic.csv
Resolving archive.ics.uci.edu (archive.ics.uci.edu)… 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:80…
connected.
HTTP request sent, awaiting response… 200 OK
Length: 8672 (8.5K) [application/x-httpd-php]
Saving to: ‘chemical’

chemical            100%[===================>]   8.47K  --.-KB/s    in 0s

2021-06-22 12:22:34 (22.4 MB/s) - ‘chemical’ saved [8672/8672]

FINISHED --2021-06-22 12:22:34--
Total wall clock time: 0.7s
Downloaded: 1 files, 8.5K in 0s (22.4 MB/s)
```

[3]:
```
st.title('Welcome to Data Science 1 Web Application')

uploaded_file = st.file_uploader("Choose a file")
if uploaded_file is not None:
  df = pd.read_csv(uploaded_file)
  st.write(df)



st.header('\nClustering of chemical composition of Ceramics')
st.subheader('Introduction:\n\n')



st.write("""
In general, clustering is a method/technique of dividing data points into␣
 ↪distinct groups such that data points belonging to same group have more␣
 ↪similarity to other data points in the same group that those in other groups.
 ↪ This method of identifying and grouping similar data points in a large␣
 ↪dataset into one cluster is one of the most widely used technique in machine␣
 ↪learning and data science. This unsupervised learning technique is␣
 ↪frequently used in statistical data analysis and machine learning to design␣
 ↪a model to predict the future assumptions. Moreover clustering is used in␣
 ↪several fields like pattern recognition, image analysis and bioinformatics.

Based on cluster model and the type of algorithms used, clustering can be␣
 ↪categorised into five distinct types:\n
1) Hierarchical clustering\n
2) Partitioning based clustering (k-means clustering)\n
3) Density based clustering \n
4) Distribution based clustering \n
5) Grid based clustering\n

So let us start with an example dataset.\n

""")
```

```
2021-06-22 12:22:35.409 WARNING root:
  Warning: to view this Streamlit app on a browser, run it with the
following
  command:

    streamlit run /opt/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py [ARGUMENTS]
```

[4]:
```
st.subheader('Description of dataset:\n\n')
st.write("""
There are multiple features in the given dataset. Each rows represents the␣
 ↪ceramic sample from two different kilns
used for the categorization and each columns represents the amount of chemical␣
 ↪elements presented in the respective
ceramic samples.
The given dataset consists of 88 instances and 19 different attributes. The␣
 ↪chosen dataset describes the chemical composition of celadon body and glaze␣
 ↪from Longquan kiln and Jingdezhen kiln in China respectively. The chemical␣
 ↪composition was examined by using Energy Dispersive X-ray Fluorescence␣
 ↪(EDXRF) technology. The classification of ceramic samples based on their␣
 ↪composition was performed using a statistical technique called Random Forest.
 ↪ The samples are categorized according to different cultural eras and kilns␣
 ↪that were used to manufacture them. 44 samples were used to examine the␣
 ↪chemical composition of celadon body and the same samples were examined␣
 ↪again to determine the composition of glaze in those samples.

""")
```

[ ]:

```
```

[5]:

```
#reading the dataset
data = "Chemical"
df1 = pd.read_csv(data)
df1.head()


st.write(df1.head(20))
```

[6]:

```
st.subheader('Goal of the project:\n\n')
st.write("""
The aim of the project is to classify celadon body and glaze based on their␣
↪chemical composition. Furthermore, the analysis regarding similarities and␣
↪differences between ceramic composition during celadon manufacture in␣
↪Longquan kiln and Jingdezhen kiln is also performed. Hierarchical clustering␣
↪method is applied to obtain the results of clustering analysis.

""")
```

[7]:

```
#since the dataset is labeled, i assigned two dummy variables for "Part"␣
↪attribute as "Part_Body" = 1 and "Part_Glaze" = 0
#this is very useful to determine the accuracy of a model later

df1_part= pd.get_dummies(df1[["Part"]])

#adding two columns at the end of dataset
df = pd.concat([df1,df1_part],axis =1)

#defining the target variables i.e. the Ceramic Names are assigned to either␣
↪"Body" as value 1 or "Glaze" as valaue 0
```

```
#x = df.iloc[:,(20)].values
y = df.iloc[:,(19)].values
df.head()
```

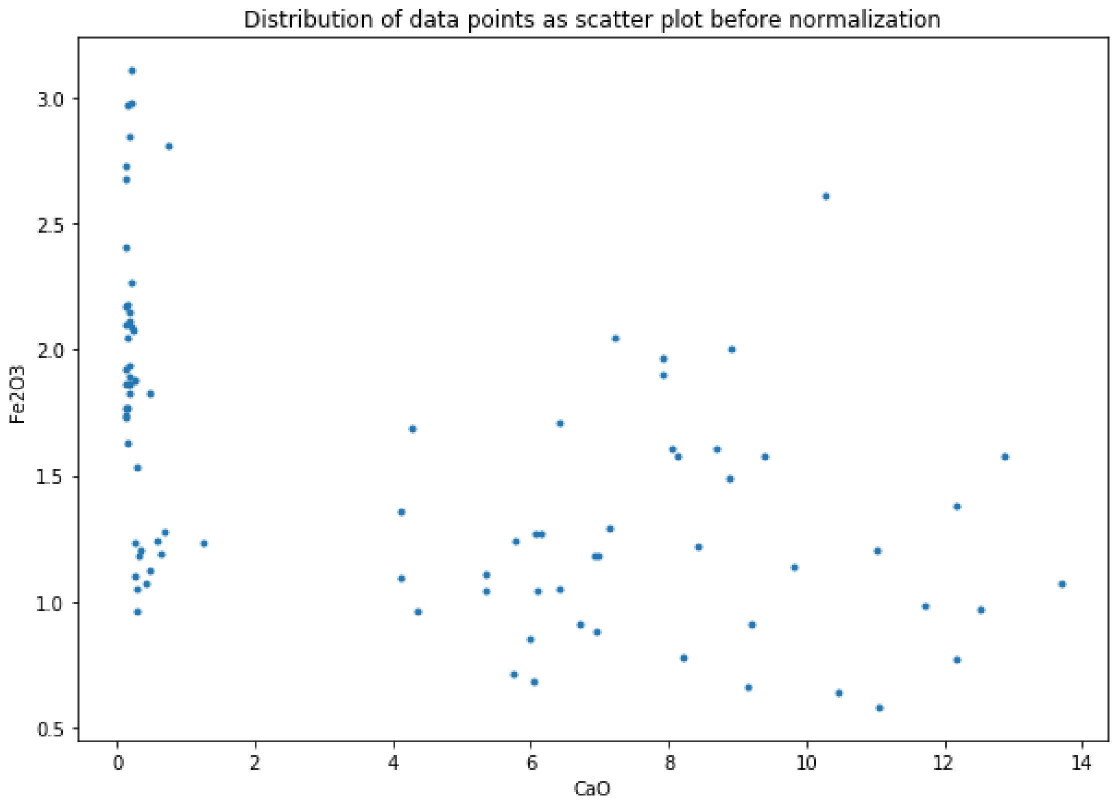
```
[7]:   Ceramic Name  Part  Na2O   MgO  Al2O3   SiO2   K2O   CaO  TiO2  Fe2O3  …  \
     0      FLQ-1-b  Body  0.62  0.38  19.61  71.99  4.84  0.31  0.07   1.18  …
     1      FLQ-2-b  Body  0.57  0.47  21.19  70.09  4.98  0.49  0.09   1.12  …
     2      FLQ-3-b  Body  0.49  0.19  18.60  74.70  3.47  0.43  0.06   1.07  …
     3      FLQ-4-b  Body  0.89  0.30  18.01  74.19  4.01  0.27  0.09   1.23  …
     4      FLQ-5-b  Body  0.03  0.36  18.41  73.99  4.33  0.65  0.05   1.19  …

        CuO  ZnO  PbO2  Rb2O  SrO  Y2O3  ZrO2  P2O5  Part_Body  Part_Glaze
     0   10   70    10   430    0    40    80    90          1           0
     1   20   80    40   430  -10    40   100   110          1           0
     2   20   50    50   380   40    40    80   200          1           0
     3   20   70    60   380   10    40    70   210          1           0
     4   40   90    40   360   10    30    80   150          1           0

     [5 rows x 21 columns]
```

```
[8]: #cleaning the dataset for the analysis
     # since the first two columns are non categorical, they need to be removed for␣
      ↪the analysis
     df.pop("Ceramic Name")
     df.pop("Part")
     df.head()
```

```
[8]:    Na2O   MgO  Al2O3   SiO2   K2O   CaO  TiO2  Fe2O3  MnO  CuO  ZnO  PbO2  \
     0  0.62  0.38  19.61  71.99  4.84  0.31  0.07   1.18  630   10   70    10
     1  0.57  0.47  21.19  70.09  4.98  0.49  0.09   1.12  380   20   80    40
     2  0.49  0.19  18.60  74.70  3.47  0.43  0.06   1.07  420   20   50    50
     3  0.89  0.30  18.01  74.19  4.01  0.27  0.09   1.23  460   20   70    60
     4  0.03  0.36  18.41  73.99  4.33  0.65  0.05   1.19  380   40   90    40

        Rb2O  SrO  Y2O3  ZrO2  P2O5  Part_Body  Part_Glaze
     0   430    0    40    80    90          1           0
     1   430  -10    40   100   110          1           0
     2   380   40    40    80   200          1           0
     3   380   10    40    70   210          1           0
     4   360   10    30    80   150          1           0
```

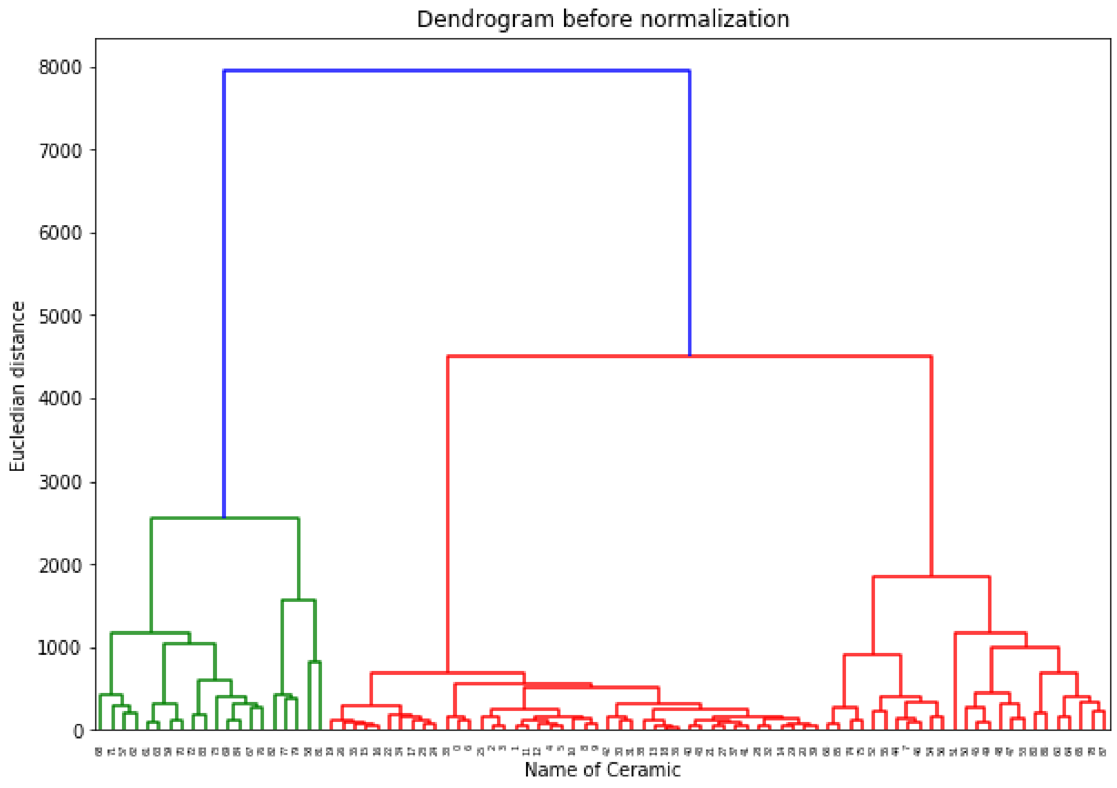
```
[9]: st.subheader('Implementation and results:\n\n')
     st.write("""
     Before starting with the clustering analysis, let's analyse the feature dataset␣
      ↪by visualizing the distribution of datapoints of some features.

     """)
```

[10]:
```
# Scatter plot to visualize the distribution of data points before normalizing␣
 ↪the dataset
plot_before_normalization = plt.figure(figsize = (10,7))
#ax= plt.scatter(df.iloc[:, 5], df.iloc[:, 7], marker='.')
ax = plt.scatter(df["CaO"],df["Fe2O3"],marker ='.')
plt.title("Distribution of data points as scatter plot before normalization")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")


#st.bar_chart(df["Fe2O3"])
#st.bar_chart(df["CaO"])
st.write(plot_before_normalization)
```



[11]:
```
st.write("""
Let's look at the dendrogram to have a visualization of how it looks like␣
 ↪before normalizing the dataset. We can compare this dendrogram with the␣
 ↪dendrogram after normalizing the datasete in order to understand why is it␣
 ↪so important to normalize the data before starting with any type of data␣
 ↪analysis. Here we use "Euclidean distance" as a distance measure and ward␣
 ↪linkage as the linkage criterion.
""")
```

[12]:
```
#ploting the dendrogram of datapoints before normalizing the dataset
from scipy.cluster import hierarchy
dend_before_normalization = plt.figure(figsize=(10, 7))
plt.title("Dendrogram before normalization")
plt.xlabel("Name of Ceramic")
plt.ylabel("Eucledian distance")
#cluster_predict = cluster.fit_predict(df)
dendrogram = hierarchy.dendrogram(hierarchy.linkage(df, method='ward'))
#print("\nAccuracy of ward linkage before normalization: ",metrics.
 ↪adjusted_rand_score(cluster_predict, y))



st.write(dend_before_normalization)
#st.write("\nAccuracy of ward linkage before normalization: ",metrics.
 ↪adjusted_rand_score(cluster_predict, y))
```



[13]:
```
st.write("""
Let's do one more step before starting with acutal clustering analysis. The␣
 ↪feature dataset above needs to be cleaned. In this step the dataset with␣
 ↪null values are either dropped or treated with the respective process.␣
 ↪Removing the dataset with null values can result in loss of information. So,␣
 ↪we need to be careful before doing this. In our case, the given dataset does␣
 ↪not contain any null values. Let's creat our feature dataset by droping the␣
 ↪first two columns representing "Ceramic Name" and "Part".
\n
Now we have our feature dataset. In the next step let's normalize the feature␣
 ↪dataset so that all the variables are scaled same. Now the model is not␣
 ↪biased towards the variable with higher values.

""")
```

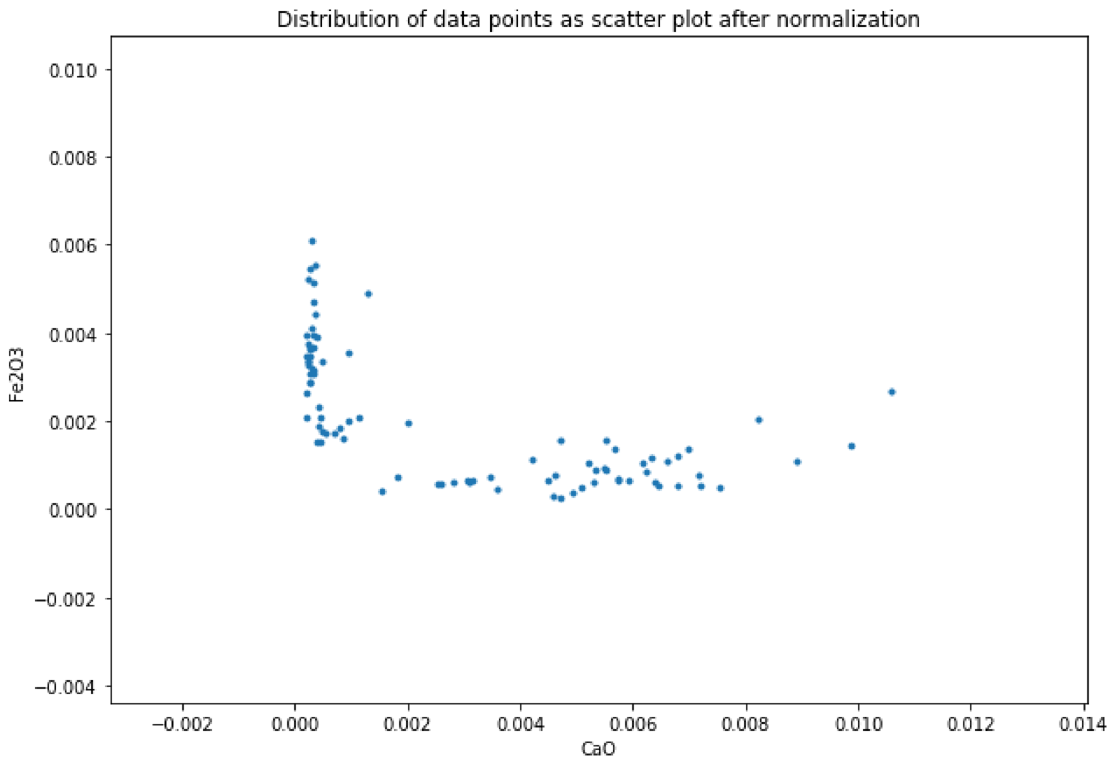
[ ]:
```

```

[14]:
```
#normalising the data so that all the variables are scaled same. Now the model␣
 ↪is not biased towards the variable with higher values
from sklearn import preprocessing
data_normalized = preprocessing.normalize(df)
data_normalized = pd.DataFrame(data_normalized, columns=df.columns)
data_normalized.head()


st.write (data_normalized)
```

[15]:
```
st.write("""
The above dataset is cleaned. Now the dataset is ready for our clustering␣
 ↪analysis.
Before starting with the actual clustering analysis, let's analyse the feature␣
 ↪dataset by visualizing the distribution of datapoints of some features after␣
 ↪normalization.
""")
```

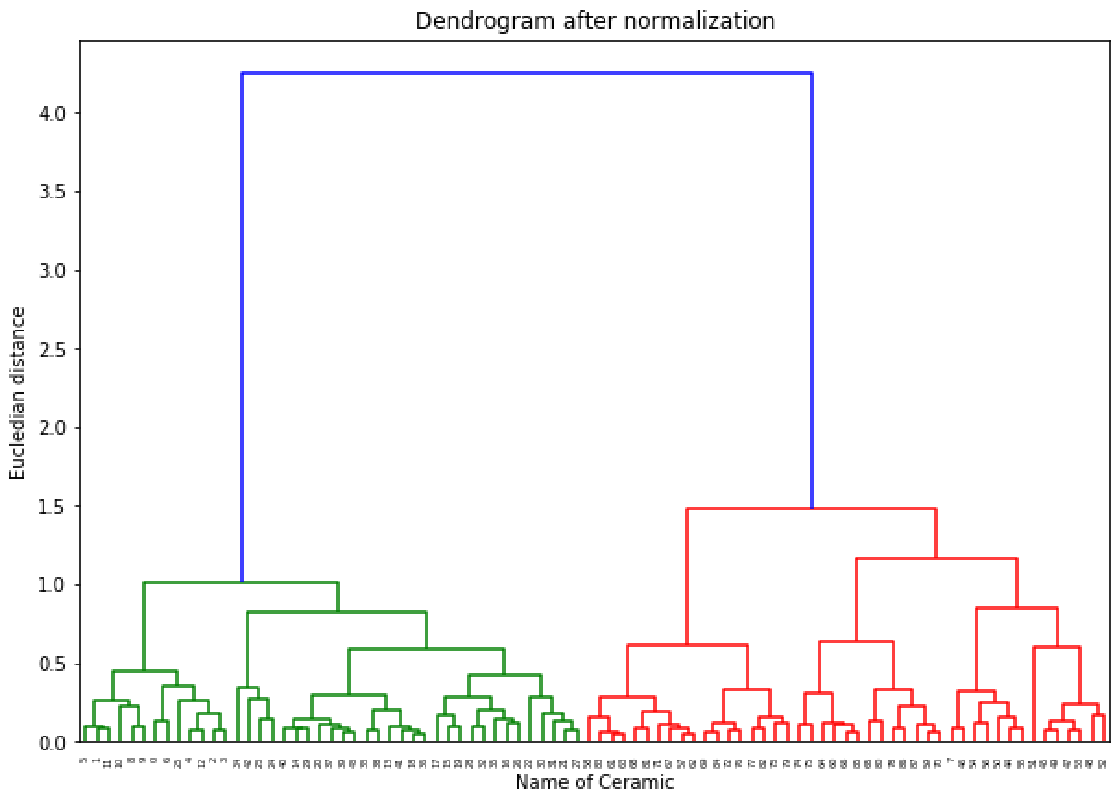
[ ]:
```

```

[16]:
```
# Scatter plot to visualize the distribution of data points after normalizing␣
 ↪the dataset
#now the dataset looks bit better
plot_after_normalization= plt.figure(figsize = (10,7))
#ax = plt.scatter(data_normalized.iloc[0:,5], data_normalized.iloc[0:,7],␣
 ↪marker='.')
ax = plt.scatter(data_normalized["CaO"],data_normalized["Fe2O3"],marker ='.')
plt.title("Distribution of data points as scatter plot after normalization")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")


st.write(plot_after_normalization)
```



[17]:
```
st.write("""
Now let's visualize the dendrogram after normalizing our feature dataset. This␣
↪looks much better than the previous dendrogram produced before normalizing␣
↪the dataset.
""")
```

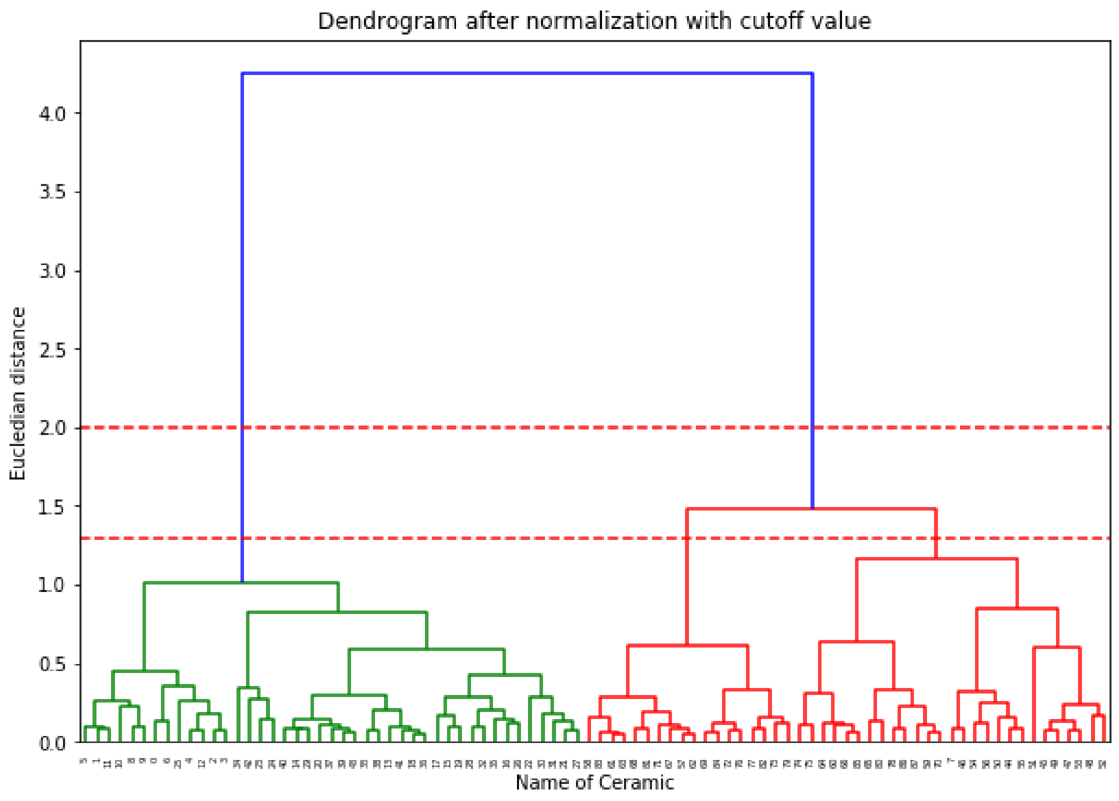
[18]:
```
#ploting the dendrogram of datapoints after normalizing the dataset
dend_after_normalization = plt.figure(figsize=(10, 7))
plt.title("Dendrogram after normalization")
plt.xlabel("Name of Ceramic")
plt.ylabel("Eucledian distance")
dendrogram = hierarchy.dendrogram(hierarchy.linkage(data_normalized,␣
↪method='ward'))


st.write(dend_after_normalization)
```



[19]:
```
st.write("""
Here the x-axis represents the ceramic samples and y-axis represents the␣
 ↪Euclidean distance between those samples. Furthermore, we need to define a␣
 ↪cutoff threshold in order to determine the optimal number of clusters.
For that we look at the longest vertical line without any horizontal line␣
 ↪passing through it. Then the number of vertical lines intersected by this␣
 ↪newly drawn horizontal line are counted. These intersections is equal to the␣
 ↪number of optimal clusters formed by the respective linkage criterion.
""")
```

[20]:
```
dend_with_cutoff = plt.figure(figsize=(10, 7))
plt.title("Dendrogram after normalization with cutoff value")
plt.xlabel("Name of Ceramic")
plt.ylabel("Eucledian distance")
dendrogram = hierarchy.dendrogram(hierarchy.linkage(data_normalized,␣
 ↪method='ward'))

# defining the threshold to cut the dendrogram
plt.axhline(y=2, color='r', linestyle='--')
plt.axhline(y=1.3, color='r', linestyle='--')
```

```
st.write(dend_with_cutoff)
```



[21]:
```
st.write("""
In the above dendrogram the blue line on the left is the longest vertical line␣
 ↪with no horizontal line passing through it. A dashed line (threshold) is␣
 ↪drawn at 2.0 in y-axis and the number of intersections between the new␣
 ↪dashed line and the vertical blue line is counted. In our case the optimal␣
 ↪number clusters is 2. This threshold basically determines the shortest␣
 ↪distance needed to form a separate cluster. If we draw a line further below␣
 ↪the number of intersections between threshold and vertical line increases␣
 ↪resulting more number of clusters.
""")
```

[22]:
```
st.write("""
Now, let's try clustering our dataset with different linkage criterion and␣
 ↪different values of k.
""")
```

[23]:
```
#creating selectbox in streamlit webapp in order to chose the number of clusters
sel_col,disp_col = st.beta_columns(2)
```

```
k = sel_col.selectbox(
    'Choose the value of k',
    options=[1, 2, 3,4,5],index=1
    )
Algorithm = sel_col.selectbox('Choose the algorithm you want to run',
                              options = ['k-means Clustering','Hierarchical␣
 ↪Clustering',
                                        'BFR Clustering'],index = 1)
```

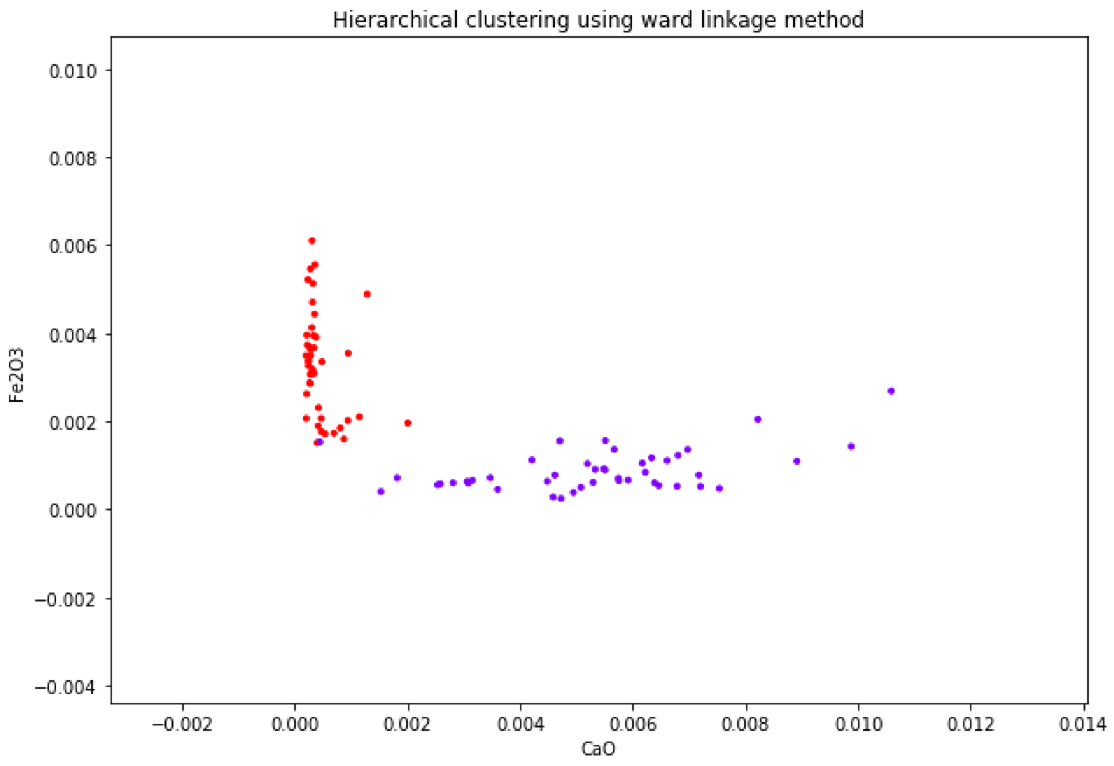
[24]:
```
st.subheader('Ward linkage clustering:\n\n')
st.write("""
In the figure below two features namely Fe2O3 and CaO which are responsible for␣
 ↪the composition of ceramic are chosen and clustered using ward linkage␣
 ↪method. The distance metric used for clustering is "Euclidean Distance". As␣
 ↪we know that the chemical composition of celadon body and glaze should be␣
 ↪different, it is expected that two separate clusters, each representing␣
 ↪celadon body and celadon glaze respectively, would be obtained for the␣
 ↪selected features. As expected Ward linkage is able to detect two clusters␣
 ↪distinguishing the difference in chemical composition of celadon body and␣
 ↪glaze. In addition, Rand score is calculated to estimate the accuracy of␣
 ↪this linkage method. Rand index is a function that measures the similarity␣
 ↪of two assignments, ground truth: labels_true and prediction: labels_pred,␣
 ↪ignoring permutations. The Rand score for ward linkage is 0.95. This shows␣
 ↪that ward linkage is successful for providing promising clustering results␣
 ↪for the selected features.
""")
```

[25]:
```
# for performing hiererchical clustering using scikitlearn package
from sklearn.cluster import AgglomerativeClustering
from sklearn import metrics
#defining the number of clusters that we wanted. 2 is the optimal number of␣
 ↪cluster in our case
#clustering is performed using ward linkage method
#k = 2
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
 ↪linkage='ward')
cluster_predict = cluster.fit_predict(data_normalized)
print ("Cluser prediction: ",cluster_predict)
print ("\nTraget variable: ",y )
#print ("\nCluster labels: ",cluster.labels_)
ward_linkage = plt.figure(figsize = (10,7))
#ax = plt.scatter(data_normalized.iloc[:,5], data_normalized.iloc[:,7], c =␣
 ↪cluster_predict,cmap='rainbow',marker='.')
ax = plt.scatter(data_normalized["CaO"],data_normalized["Fe2O3"], c =␣
 ↪cluster_predict, cmap = 'rainbow', marker ='.')
```

```
#plt.scatter(data_normalized.iloc[:,1], data_normalized.iloc[:,2], c =␣
↪cluster_predict,cmap='rainbow')
#plt.scatter(data_normalized.iloc[:,3], data_normalized.iloc[:,4], c =␣
↪cluster_predict,cmap='rainbow')


# accuracy of clustering is calculated.
# cluster evaluation is done using external validation index called Rand index
print("Accuracy of ward linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict, y))

#trying other evaluation metrices for comparison
print("Accuracy of ward linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
print("Accuracy of ward linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))

plt.title("Hierarchical clustering using ward linkage method")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")
#plt.figtext(0.7,0.8,"Cluster1(Red)\nCluster2(Blue)",fontsize = 12)


st.write(ward_linkage)
st.write("Accuracy of ward linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict, y))
st.write("Accuracy of ward linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
st.write("Accuracy of ward linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))
```

```
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
```



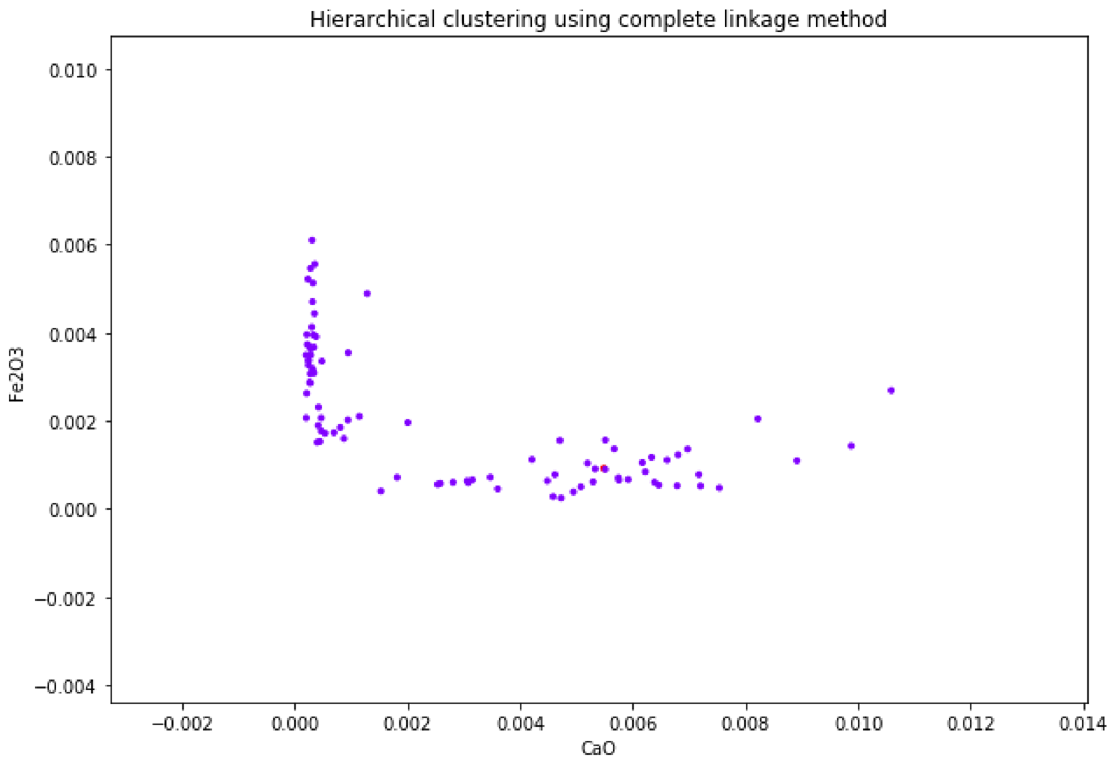
[26]:
```
st.subheader('Complete linkage clustering:\n\n')
st.write("""
Next, Complete linkage clustering is performed for the same features. The␣
 ↪default "Euclidean Distance" is used as distance metric. As we can see, this␣
 ↪linkage criterion is not able to distinguish these features in separate␣
 ↪clusters. Rand score is calculated to evaluate the accuracy of this␣
 ↪clustering method. The value of rand score calculated is 0.0. This shows␣
 ↪that complete linkage criterion is not optimal for the given dataset and␣
 ↪therefore performs bad clustering. However, rand score of 0.0 does not␣
 ↪necessarily mean that none of the features are assigned correctly into␣
 ↪separate clusters. Normally ,random (uniform) label assignments have the␣
 ↪rand score close to 0.0.
""")
```

[27]:
```
#clustering is performed using complete linkage method
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
 ↪linkage='complete')
cluster_predict = cluster.fit_predict(data_normalized)
print ("Cluster prediction: ",cluster_predict)
print ("\nTarget variable: ",y)
#print ("Cluster labels: ",cluster.labels_)
complete_linkage = plt.figure(figsize =(10,7))
#ax = plt.scatter(data_normalized.iloc[:,5], data_normalized.iloc[:,7], c =␣
 ↪cluster_predict, cmap='rainbow',marker = '.')
ax = plt.scatter(data_normalized["CaO"],data_normalized["Fe2O3"], c =␣
 ↪cluster_predict, cmap = 'rainbow', marker ='.')
print("Accuracy of complete linkage using rand score: ",metrics.
 ↪adjusted_rand_score(cluster_predict, y))
print("Accuracy of complete linkage using mutual info score: ",metrics.
 ↪adjusted_mutual_info_score(cluster_predict,y))
print("Accuracy of complete linkage using v-measure score: ",metrics.
 ↪v_measure_score(cluster_predict,y))
plt.title("Hierarchical clustering using complete linkage method")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")
#plt.figtext(0.7,0.8,"Cluster1(Blue)",fontsize = 12)


st.write(complete_linkage)
st.write("Accuracy of complete linkage using rand score: ",metrics.
 ↪adjusted_rand_score(cluster_predict, y))
st.write("Accuracy of complete linkage using mutual info score: ",metrics.
 ↪adjusted_mutual_info_score(cluster_predict,y))
st.write("Accuracy of complete linkage using v-measure score: ",metrics.
 ↪v_measure_score(cluster_predict,y))
```

```
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
```

```
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
```



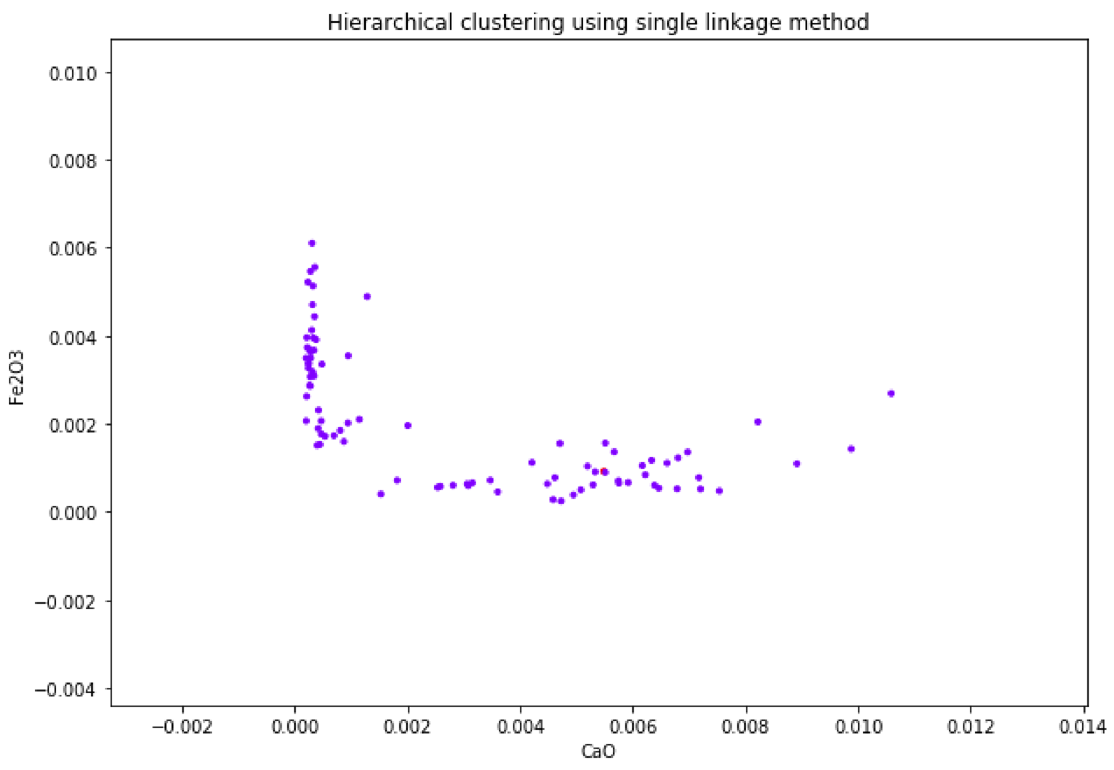
[28]:
```
st.subheader('Single linkage clustering:\n\n')
st.write("""
Next, single linkage is chosen as the linkage criterion. The distance measure␣
 ↪is again the default "Euclidean Distance". As above in the complete linkage␣
 ↪clustering, single linkage method also fails to cluster the features into␣
 ↪different clusters. The rand score calculated here is 0.0. This shows that␣
 ↪single and complete linkage criterion are not optimal for the given dataset.
""")
```

[29]:
```
#clustering is performed using single linkage method
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
 ↪linkage='single')
cluster_predict = cluster.fit_predict(data_normalized)
print ("Cluster prediction: ",cluster_predict)
print ("\nTarget variable: ",y)
single_linkage = plt.figure(figsize=(10,7))
#ax = plt.scatter(data_normalized.iloc[:,5], data_normalized.iloc[:,7], c =␣
↪cluster_predict, cmap='rainbow',marker = '.')
ax = plt.scatter(data_normalized["CaO"],data_normalized["Fe2O3"], c =␣
↪cluster_predict, cmap = 'rainbow', marker ='.')
print("Accuracy of single linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict, y))
print("Accuracy of single linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
print("Accuracy of single linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))
plt.title("Hierarchical clustering using single linkage method")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")
#plt.figtext(0.7,0.8,"Cluster1(Blue)",fontsize = 12)



st.write(single_linkage)
st.write("Accuracy of single linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict, y))
st.write("Accuracy of single linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
st.write("Accuracy of single linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))
```

```
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
```

```
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
```

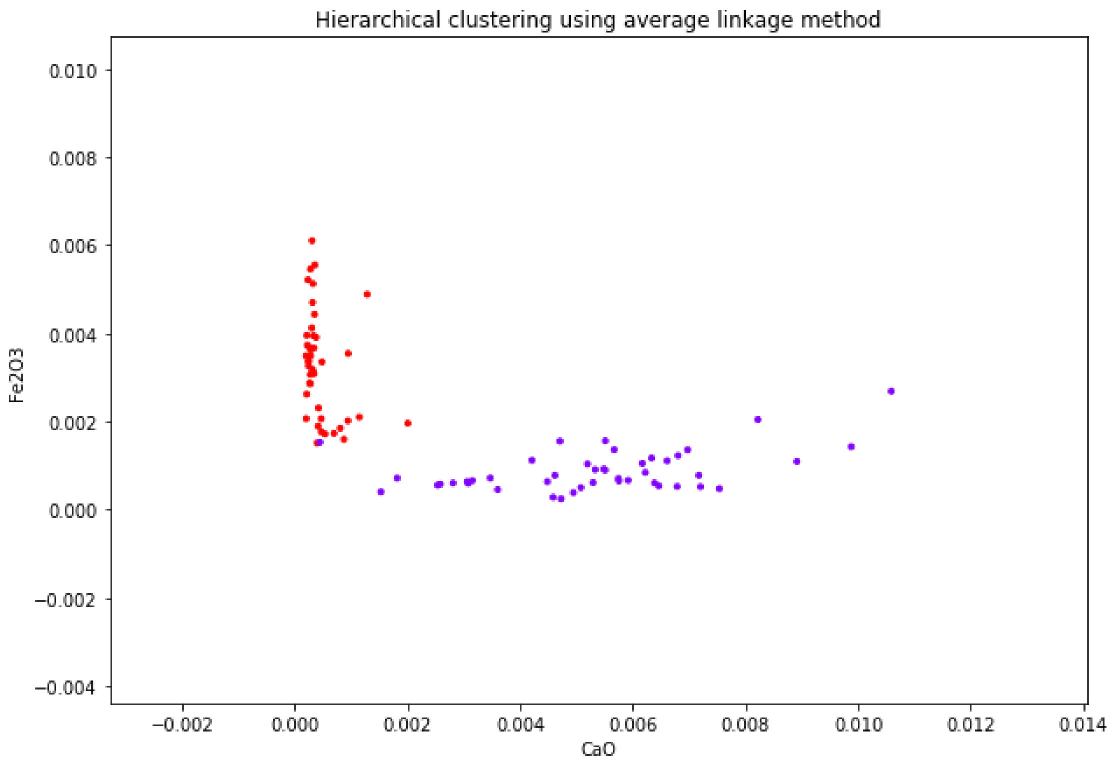


[30]:
```
st.subheader('Average linkage clustering:\n\n')
st.write("""
Finally, average linkage clustering is performed for the same selected features.
 ↪ As before default "Euclidean distance" is chosen as distance metric. In␣
 ↪contrast to complete and single linkage clustering, average and ward linkage␣
 ↪clustering produce much favourable resuts. They both have the Rand score of␣
 ↪0.95 and are also able to cluster two different features into separate␣
 ↪clusters.
""")
```

[31]:
```
#clustering is performed using average linkage method
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
 ↪linkage='average')
cluster_predict = cluster.fit_predict(data_normalized)
print ("Cluster prediction: ",cluster_predict)
print ("\nTarget variable: ",y)
#print (cluster.labels_)
average_linkage = plt.figure(figsize=(10,7))
```

```
#ax = plt.scatter(data_normalized.iloc[:,5], data_normalized.iloc[:,7], c =␣
↪cluster_predict, cmap='rainbow',marker = '.')
ax = plt.scatter(data_normalized["CaO"],data_normalized["Fe2O3"], c =␣
↪cluster_predict, cmap = 'rainbow', marker ='.')
print ("\nAccuracy of average linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict,y))
print("Accuracy of average linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
print("Accuracy of average linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))
plt.title("Hierarchical clustering using average linkage method")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")
#plt.figtext(0.7,0.8,"Cluster1(Red)\nCluster2(Blue)",fontsize = 12)



st.write(average_linkage)
st.write("Accuracy of average linkage using rand score: ",metrics.
↪adjusted_rand_score(cluster_predict, y))
st.write("Accuracy of average linkage using mutual info score: ",metrics.
↪adjusted_mutual_info_score(cluster_predict,y))
st.write("Accuracy of average linkage using v-measure score: ",metrics.
↪v_measure_score(cluster_predict,y))
```

```
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857

/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
```



[32]:
```
st.subheader('Analysis of the clusters:\n\n')
st.write("""
For the chosen value of k = 2, hierarchical clustering successfully clusters␣
 ↪celadon body (Red cluster) and glaze (blue cluster) into two distinct␣
 ↪cluters. From the above figure we can conclude that celadon body is composed␣
 ↪of more Fe2O3 and less Cao. Likewise, looking at the chemical composition of␣
 ↪celadon glaze, we can clearly see that celadon glaze contains more CaO and␣
 ↪less Fe2O3. This shows that ward and average linkage method have precicted␣
 ↪the clusters with high accuracy. This means, ward and average linkage␣
 ↪criterions have produced optimal clustering results for the given dataset.  ␣
 ↪
""")
```

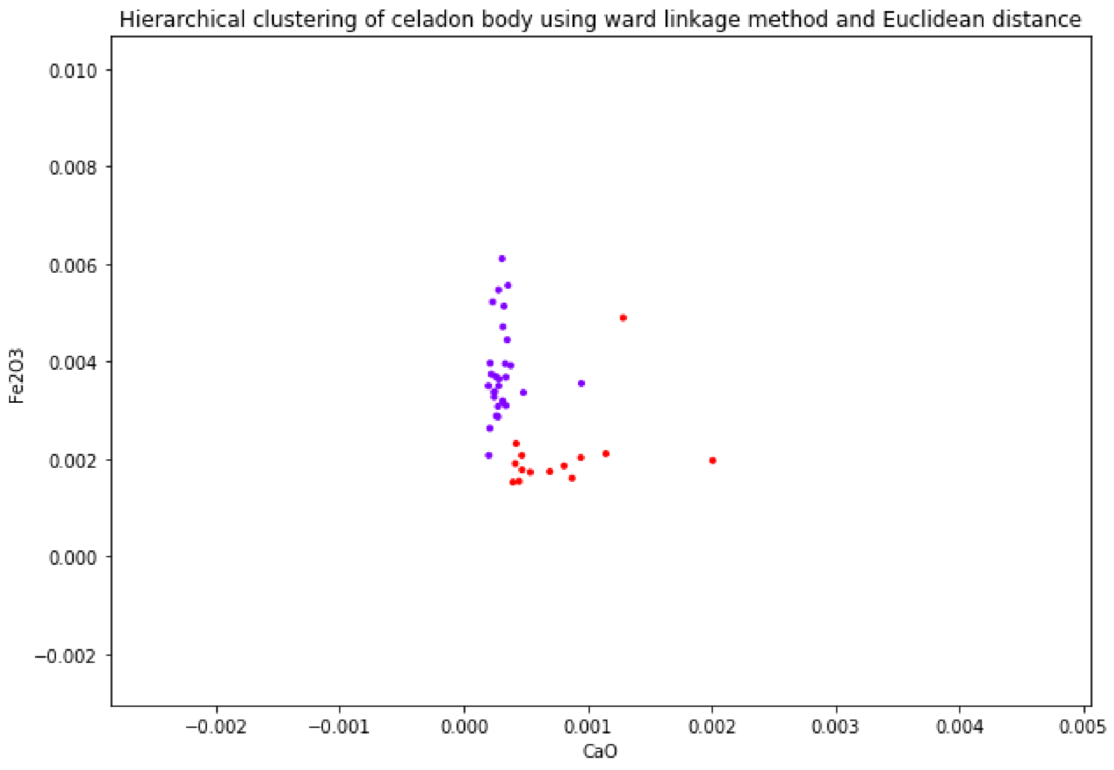
[33]:
```
st.subheader('Analysis of celadon body:\n\n')
st.write("""
For this analysis the dataset was filtered again droping all the rows with␣
↪category "Glaze". This analysis is performed in order to evaluate whether␣
↪there are any differences in the chemical composition of celadon body␣
↪manufactured in two different kilns: Longquan kiln and Jingdezhen kiln.␣
↪There are two ditinct clusters formed where it can be seen that CaO contents␣
↪of celadon body in Jingdezhen are higher than those in Longquan, whereas the␣
↪Fe2O3 contents of celadon body in Longquan are higher than those of␣
↪Jingdezehn. This indicates that the chemical composition of bodies is also␣
↪different in those two separate kilns.
""")
```

[34]:
```
# filtering only the columns with category "Body"
df2 = data_normalized.iloc[0:43, 0:17]
df2.head()
```

[34]:
```
       Na2O      MgO     Al2O3      SiO2       K2O       CaO      TiO2  \
0  0.000795  0.000487  0.025137  0.092280  0.006204  0.000397  0.000090
1  0.000941  0.000776  0.034984  0.115715  0.008222  0.000809  0.000149
2  0.000793  0.000308  0.030104  0.120903  0.005616  0.000696  0.000097
3  0.001371  0.000462  0.027742  0.114281  0.006177  0.000416  0.000139
4  0.000053  0.000635  0.032492  0.130586  0.007642  0.001147  0.000088

      Fe2O3       MnO       CuO       ZnO      PbO2      Rb2O       SrO  \
0  0.001513  0.807564  0.012818  0.089729  0.012818  0.551194  0.000000
1  0.001849  0.627359  0.033019  0.132076  0.066038  0.709906 -0.016509
2  0.001732  0.679775  0.032370  0.080926  0.080926  0.615034  0.064740
3  0.001895  0.708575  0.030808  0.107827  0.092423  0.585345  0.015404
4  0.002100  0.670668  0.070597  0.158842  0.070597  0.635369  0.017649

       Y2O3      ZrO2      P2O5
0  0.051274  0.102548  0.115366
1  0.066038  0.165094  0.181604
2  0.064740  0.129481  0.323702
3  0.061615  0.107827  0.323480
4  0.052947  0.141193  0.264737
```

[35]:
```
#Clustering of the dataset with only Celadon body
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
↪linkage='ward')
cluster_predict = cluster.fit_predict(df2)
ward_linkage_body = plt.figure(figsize=(10,7))
ax = plt.scatter(df2['CaO'],␣
↪df2['Fe2O3'],c=cluster_predict,cmap='rainbow',marker='.')
plt.title("Hierarchical clustering of celadon body using ward linkage method␣
↪and Euclidean distance")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")
```

```
st.write (ward_linkage_body)
```



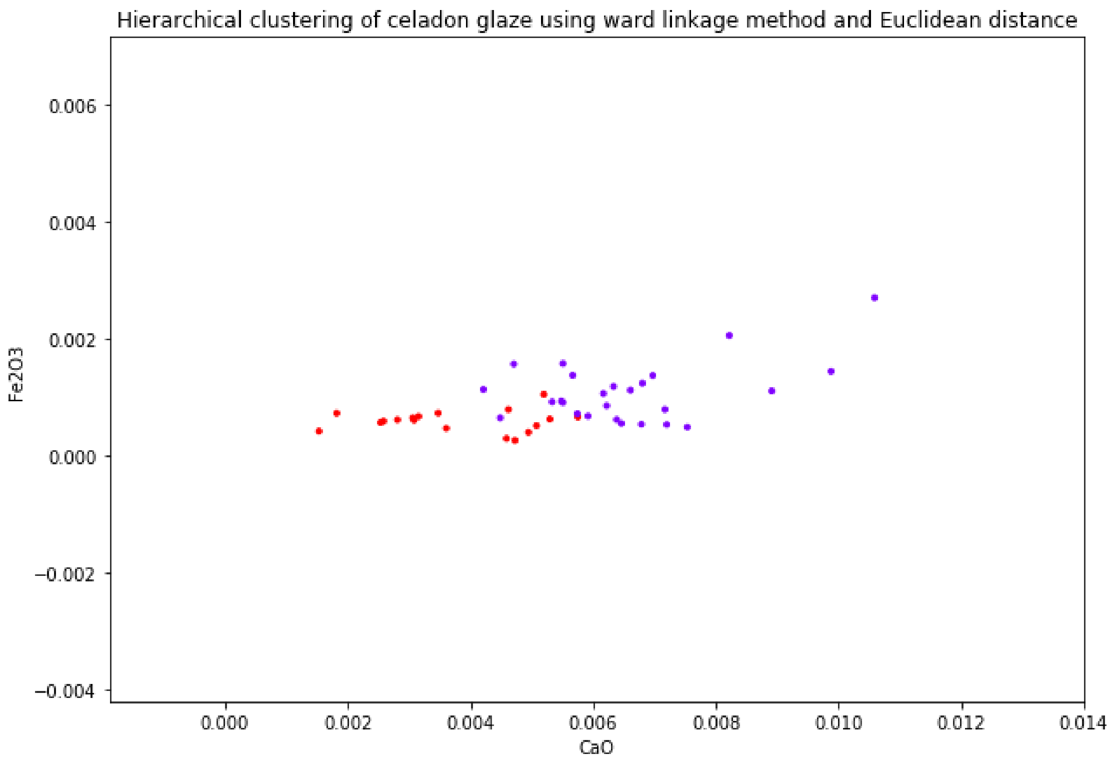
```
[36]: st.subheader('Analysis of celadon glaze:\n\n')
      st.write("""
      For this analysis the dataset was filtered droping all the rows with category␣
       ↪"Body". Two separarate clusters are formed and it is observed that CaO␣
       ↪contents of celadon glaze in Longquan are higer than those in Jingdezhen. In␣
       ↪case of Fe2O3 contents, no any significant differences are observed.
      """)
```

```
[37]: #filtering only columns with category "Glaze"
      df3 =data_normalized.iloc[44:, 0:17]
      df3.head()
```

```
[37]:        Na2O       MgO     Al2O3      SiO2       K2O       CaO      TiO2  \
      44  0.001267  0.000091  0.014915  0.097184  0.007445  0.006974  0.000065
      45  0.001087  0.000350  0.009646  0.051200  0.003610  0.006609  0.000082
      46  0.001231  0.000270  0.015987  0.081926  0.005227  0.009880  0.000082
      47  0.000097  0.000284  0.008611  0.046618  0.002974  0.008916  0.000042
      48  0.000265  0.000738  0.009426  0.049445  0.004000  0.005670  0.000057
```

```
       Fe2O3       MnO       CuO       ZnO      PbO2      Rb2O       SrO  \
44  0.001358  0.718337  0.026121  0.078364  0.026121  0.404881  0.195910
45  0.001109  0.707086  0.022329  0.029772  0.000000  0.260505  0.186075
46  0.001430  0.691506  0.023441  0.105484  0.046882  0.433657  0.105484
47  0.001095  0.665582  0.055465  0.048532  0.027733  0.228794  0.110930
48  0.001362  0.573446  0.043008  0.086017  0.000000  0.229379  0.057345

        Y2O3      ZrO2      P2O5
44  0.026121  0.065303  0.509366
45  0.022329  0.044658  0.625213
46  0.046882  0.093764  0.539141
47  0.027733  0.048532  0.693315
48  0.028672  0.057345  0.774153
```

[38]:
```
#Clustering of the dataset with only Celadon glaze
cluster = AgglomerativeClustering(n_clusters=k, affinity='euclidean',␣
 ↪linkage='ward')
cluster_predict = cluster.fit_predict(df3)
ward_linkage_glaze = plt.figure(figsize=(10,7))
ax = plt.scatter(df3['CaO'],␣
 ↪df3['Fe2O3'],c=cluster_predict,cmap='rainbow',marker='.')
plt.title("Hierarchical clustering of celadon glaze using ward linkage method␣
 ↪and Euclidean distance")
plt.xlabel("CaO")
plt.ylabel("Fe2O3")


st.write(ward_linkage_glaze)
```



[39]:
```
st.subheader('Conclusion:\n\n')
st.write("""
Hierarchical clustering analysis was performed on the dataset containing 44␣
↪different samples with 17 different chemical compositions. Two important␣
↪features were chosen for the analysis. The outcome of whole analysis can be␣
↪evaluated and verified by using other features as well. As stated on the␣
↪begining of the project, the main objective of the project was to evaluate␣
↪given samples and cluster them with various strategies accordingly.␣
↪Hierarchical clustering with "ward-" and "average-" linkage method are␣
↪successful in producing the promising results with higer accuracy rate.␣
↪Clustering with other linkage criterion ("single-" and "complete-" linkage)␣
↪is not optimal for the given dataset. In general, Agglomerative clustering␣
↪has a "rich get richer" behaviour which leads to uneven cluster sizes and␣
↪therefore generate less accurate results. This is also the case for␣
↪"single-"and "complete-" linkage method during our analysis. However,␣
↪"single-" and "complete-" linkage criteria can be very efficient to compute␣
↪hierarchical clustering for larger datasets. During any data science␣
↪analysis, it is very useful and recommended to use the evaluation measures␣
↪to get an idea on how efficiently the implemented algorithm is performing.␣
↪Varying the distance metric can sometimes lead to more accurate results. But␣
↪it is necessary to understand the dataset in prior to applying them.
""")
```

```
[40]: #print(df3_normalized[['CaO']].mean())
      #print (df3_normalized[['Fe2O3']].mean())
      #print (df2_normalized[['CaO']].mean())
      #print (df2_normalized[['Fe2O3']].mean())
```

```
[41]: #https://github.com/ChristianFJung/NotebookToWebApp
      !jupyter nbconvert   --to script␣
       ↪Data_science_project_on_Hierarchical_clustering.ipynb
      !awk '!/ipython/' Data_science_project_on_Hierarchical_clustering.py >  temp.py␣
       ↪&& mv temp.py app.py && rm Data_science_project_on_Hierarchical_clustering.py
      !streamlit run app.py
```

```
[NbConvertApp] Converting notebook
Data_science_project_on_Hierarchical_clustering.ipynb to script
[NbConvertApp] Writing 28705 bytes to
Data_science_project_on_Hierarchical_clustering.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.178.24:8501

Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
```

```
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavior
of AMI will change in version 0.22. To match the behavior of 'v_measure_score',
AMI will use average_method='arithmetic' by default.
  FutureWarning)
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

```
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
```

```
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
```

```
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857
```

```
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.9545397214364223
Accuracy of ward linkage using mutual info score:  0.9207206603043875
Accuracy of ward linkage using v-measure score:  0.921553538364857
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.0
Accuracy of complete linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of complete linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.0
Accuracy of single linkage using mutual info score:  -1.5190107840345266e-16
Accuracy of single linkage using v-measure score:  0.021029315759475423
Cluster prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
```

```
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9545397214364223
Accuracy of average linkage using mutual info score:  0.9207206603043875
Accuracy of average linkage using v-measure score:  0.921553538364857
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
Cluser prediction:  [1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 2 0 0 0 2 2 2 0 2 2 2
 0 0 2 2 0 2 0 2 2 2 2 0 0 0]

Traget variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of ward linkage using rand score:  0.7185788893507057
Accuracy of ward linkage using mutual info score:  0.6172113491937017
Accuracy of ward linkage using v-measure score:  0.745055773020594
Cluster prediction:  [2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of complete linkage using rand score:  0.9325755618703178
Accuracy of complete linkage using mutual info score:  0.8521501935103011
Accuracy of complete linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
```

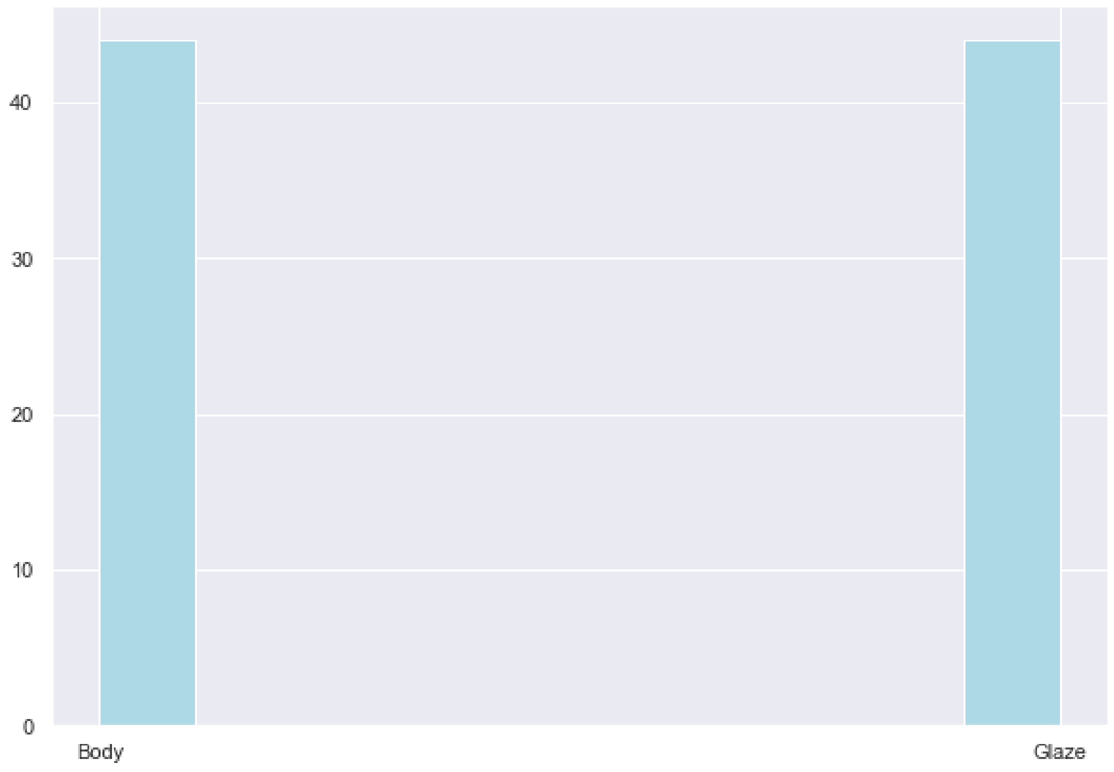
```
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Accuracy of single linkage using rand score:  0.9325755618703178
Accuracy of single linkage using mutual info score:  0.8521501935103011
Accuracy of single linkage using v-measure score:  0.8870505958964081
Cluster prediction:  [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Target variable:  [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Accuracy of average linkage using rand score:  0.9325755618703178
Accuracy of average linkage using mutual info score:  0.8521501935103011
Accuracy of average linkage using v-measure score:  0.8870505958964081
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
^C
```

1 comparison with k-means clustering

[49]:
```
#This part of implementation is adopted from Xheni Hoxha, modified and␣
 ↪implemented to my dataset
import warnings
warnings.filterwarnings('ignore')
import ipywidgets as widgets
from IPython.display import display, clear_output
import pandas as pd
import seaborn as sns
sns.set()
```

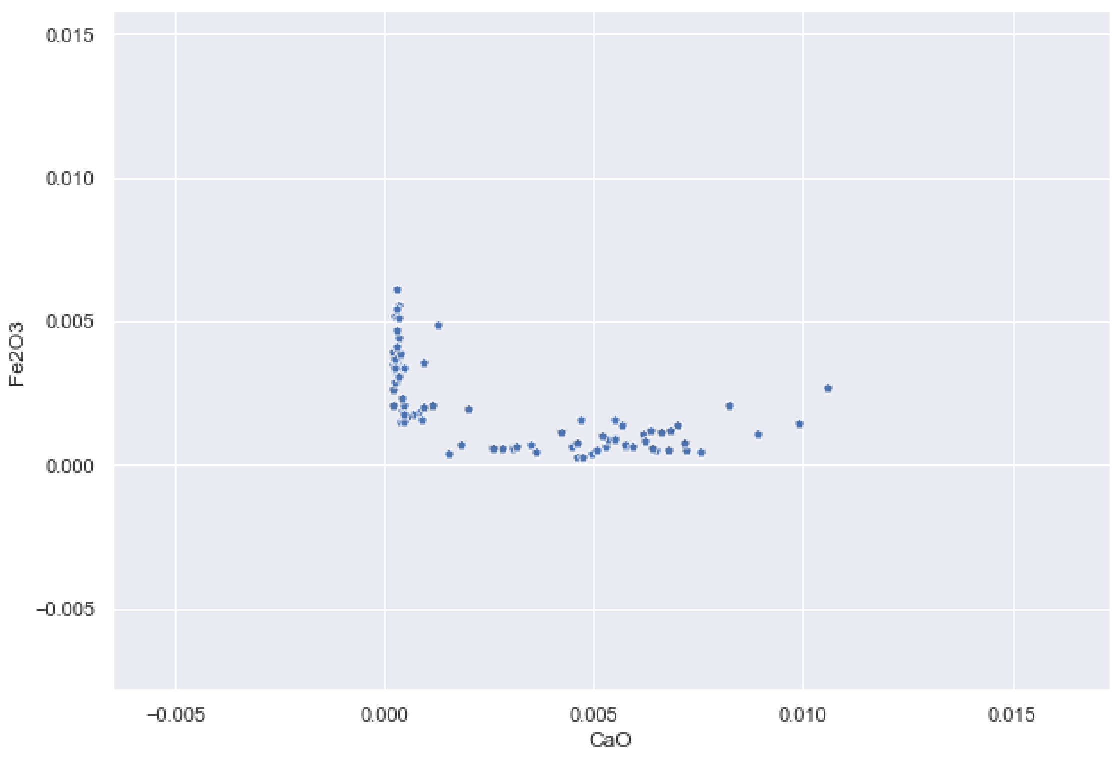
[50]:
```
plt.figure (figsize =(10,7))
plt.hist(df1["Part"],color="lightblue")
```

[50]:
```
(array([44.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 44.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



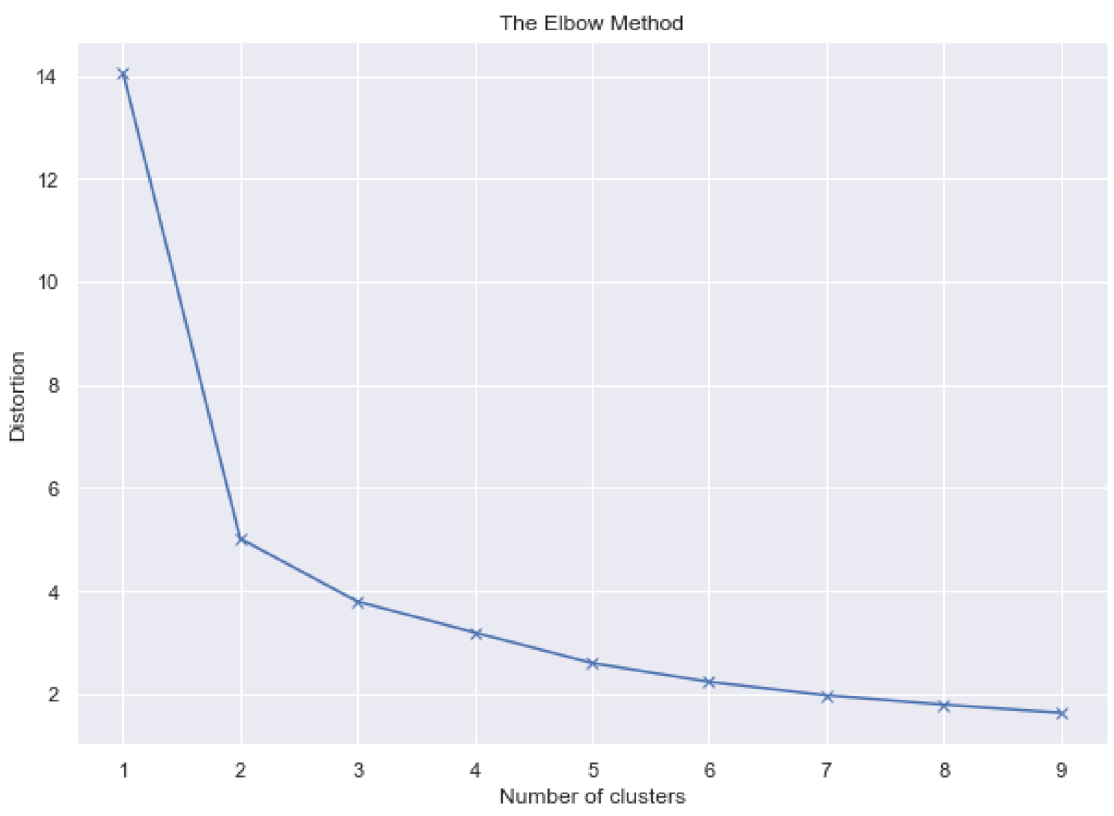
```
[44]: #Scatter plot of two features to see the distrubution of points
      plt.figure (figsize =(10,7))
      sns.scatterplot(data = data_normalized, x = "CaO", y = "Fe2O3",␣
       ↪palette="tab10", legend="full",marker ='p')
```

```
[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1a299b7e50>
```



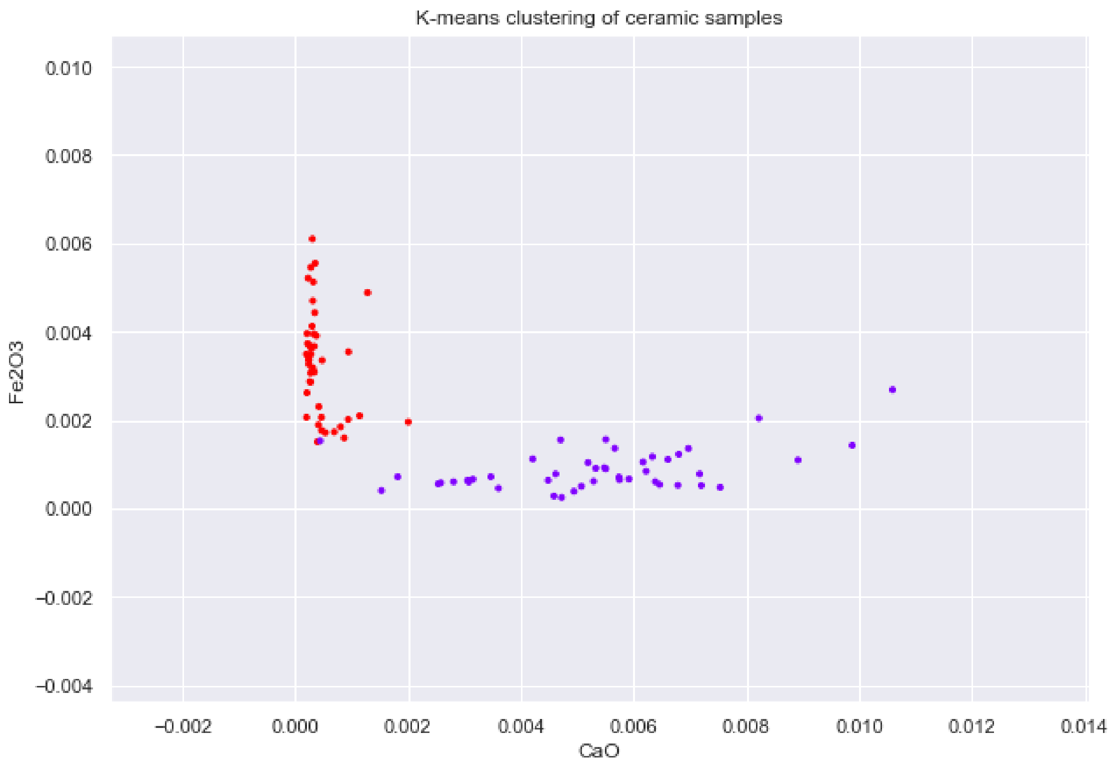
[45]:
```
df = pd.DataFrame(data_normalized,columns=["Fe2O3","CaO"])
```

[46]:
```
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
sns.set()
distortions = []
K = range(1,10)
for k in K:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(data_normalized)
    distortions.append(kmeans.inertia_)

plt.figure(figsize=(10,7))
plt.plot(K, distortions, 'bx-')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.title('The Elbow Method')
plt.show()
```



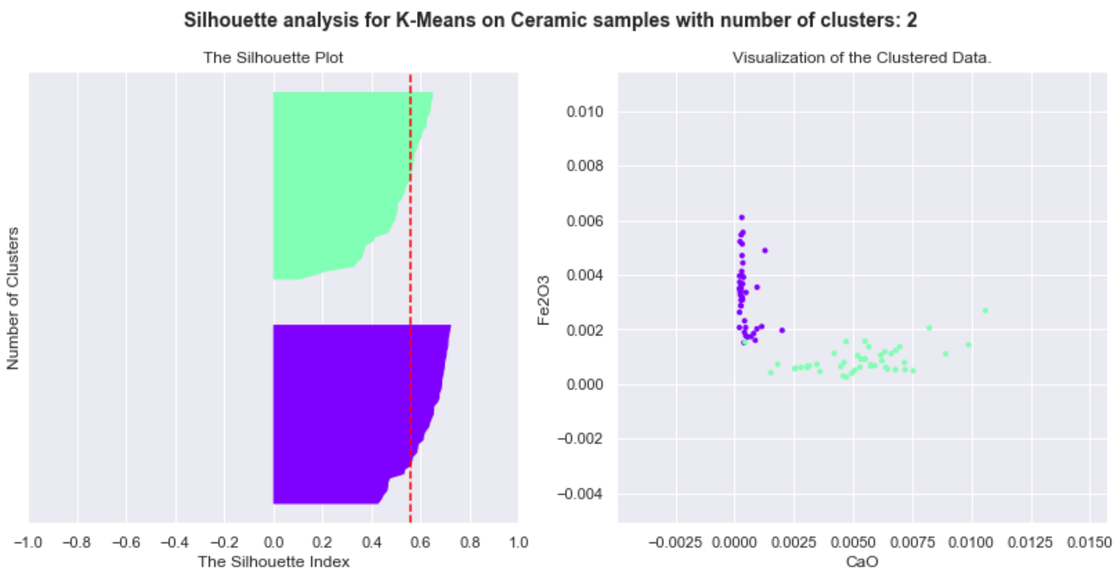
```
[47]: import statsmodels.api as sm
      import seaborn as sns
      sns.set()

      kmeans = KMeans(2)
      kmeans.fit(data_normalized)
      clusters = kmeans.fit_predict(data_normalized)

      plot = data_normalized.copy()
      plt.figure(figsize =(10,7))
      plot['Clusters'] = clusters
      plt.scatter(plot['CaO'],plot['Fe2O3'],c=plot['Clusters'],cmap= 'rainbow',marker␣
       ↪='.')
      plt.title("K-means clustering of ceramic samples")
      plt.xlabel("CaO")
      plt.ylabel("Fe2O3")
```

```
[47]: Text(0, 0.5, 'Fe2O3')
```

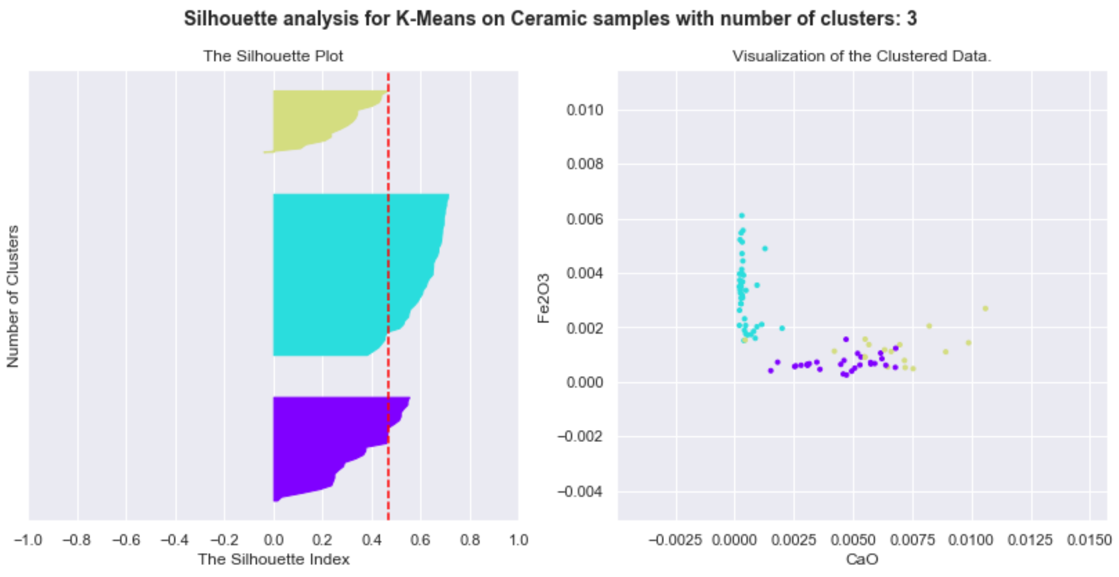


[48]:
```
#sihhouette coefficient was used as validation index
from sklearn.metrics import silhouette_samples, silhouette_score
import matplotlib.cm as cm

range_n_clusters = [2, 3, 4, 5, 6]

for n_clusters in range_n_clusters:
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(14, 6)
    ax1.set_xlim([-1, 1])

    clusterer = KMeans(n_clusters=n_clusters, random_state=10)
    cluster_labels = clusterer.fit_predict(data_normalized)

    # The silhouette_score gives the average value for all the samples.
    # This gives a perspective into the density and separation of the formed
    # clusters
    silhouette_avg = silhouette_score(data_normalized, cluster_labels)
    print("For number of clusters: ", n_clusters,
          "the average silhouette index is :", silhouette_avg)

    # Compute the silhouette scores for each sample
    sample_silhouette_values = silhouette_samples(data_normalized,␣
↪cluster_labels)

    y_lower = 12
    for i in range(n_clusters):
        ith_cluster_silhouette_values = \
            sample_silhouette_values[cluster_labels == i]

        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.rainbow(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0, ith_cluster_silhouette_values,
                          facecolor=color, edgecolor=color)

        y_lower = y_upper + 10

    ax1.set_title("The Silhouette Plot")
    ax1.set_xlabel("The Silhouette Index")
    ax1.set_ylabel("Number of Clusters")
    ax1.axvline(x=silhouette_avg, color="red", linestyle="--")
    ax1.set_yticks([])  # Clear the yaxis labels / ticks
    ax1.set_xticks([-1,-0.8,-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1])

    colors = cm.rainbow(cluster_labels.astype(float) / n_clusters)
    ax = plt.scatter(data_normalized["CaO"],df["Fe2O3"],marker ='.', c=colors)

    centers = clusterer.cluster_centers_
    # Draw white circles at cluster centers


    ax2.set_title("Visualization of the Clustered Data.")
    ax2.set_xlabel("CaO")
    ax2.set_ylabel("Fe2O3")


    plt.suptitle(("Silhouette analysis for K-Means on Ceramic samples "
                  "with number of clusters: %d" % n_clusters),
                 fontsize=14, fontweight='bold')


    plt.show()
```

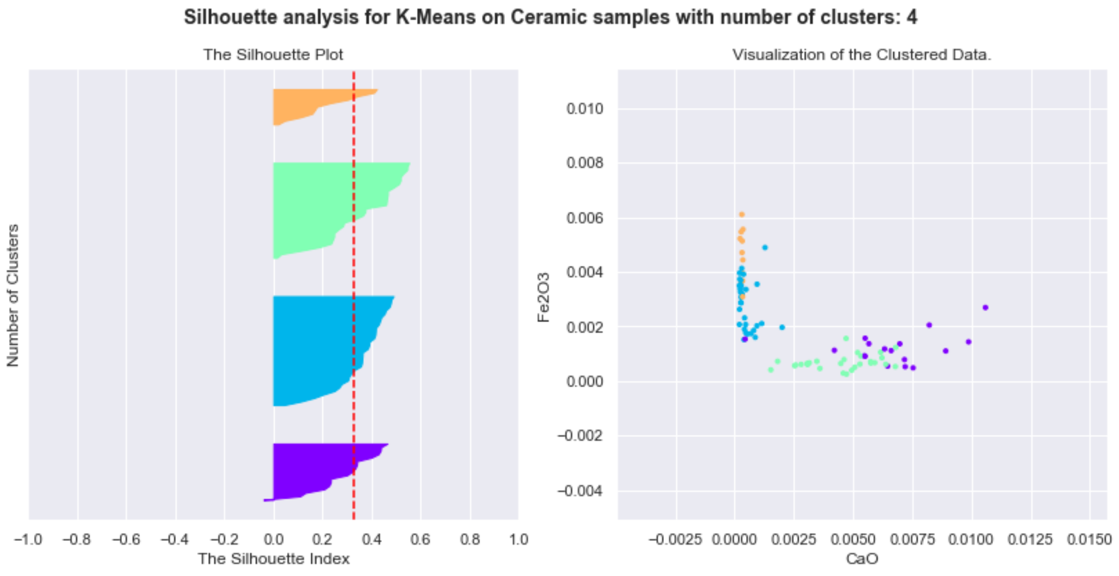
```
For number of clusters:  2 the average silhouette index is : 0.5580909282609473
```



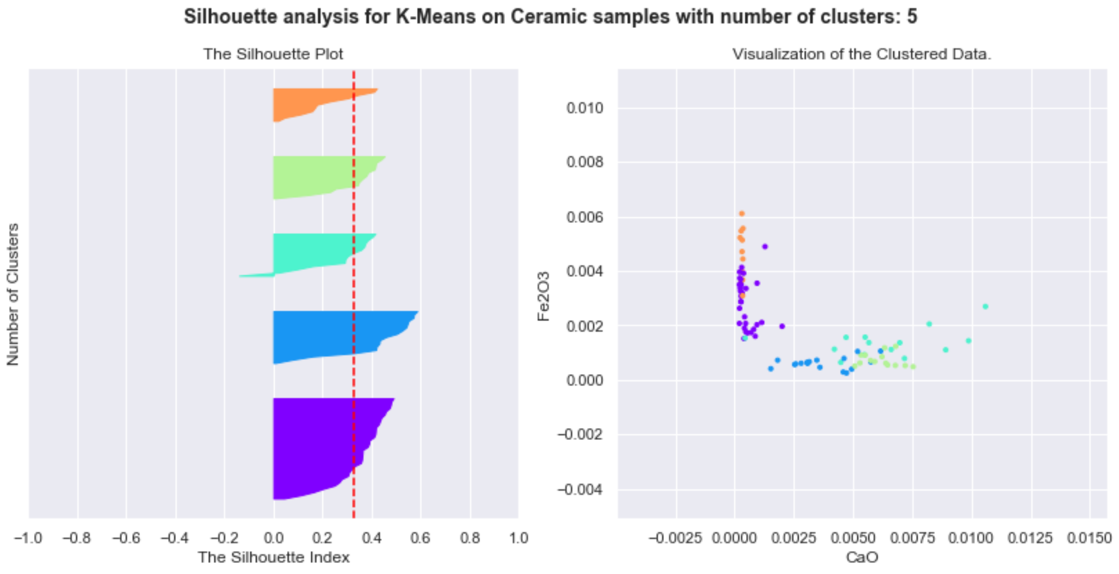
```
For number of clusters:  3 the average silhouette index is : 0.4675067677018909
```



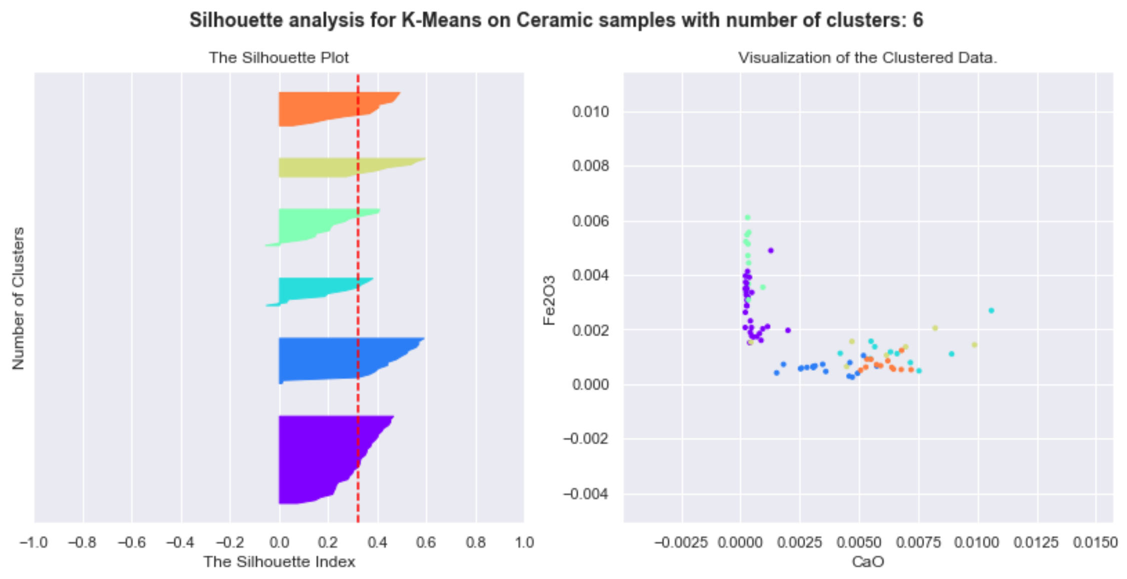
```
For number of clusters:  4 the average silhouette index is : 0.3267330699215132
```



```
For number of clusters:  5 the average silhouette index is : 0.3305482951242935
```



```
For number of clusters:  6 the average silhouette index is : 0.3209020676791835
```


Silhouette analysis for K-Means on Ceramic samples with number of clusters: 6
The Silhouette Plot
Number of Clusters
The Silhouette Index
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Visualization of the Clustered Data.
Fe2O3
CaO
0.010
0.008
0.006
0.004
0.002
0.000
−0.002
−0.004
−0.0025
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150

```
[ ]:
```