2.1 Problem 1

Load the auto-mpg sample dataset into the Orange application - ensure that origin is set as a target attribute type, as it will be used as a class label. Perform a Hierarchical Clustering using Linkage set to Average, after calculating Distances, with Pruning set to a Max Depth of 5. Also, set Selection to Top N with a value of 3. This will result in a shallow tree of depth 5, and a final cut resulting in 3 clusters. Examine the resulting clusters (C1, C2, C3) via Distributions analysis - is there a clear relationship between the cluster assignment and class label (1,2,3)? What are the probabilities calculated for each value of origin for each cluster? Does changing the Max Depth affect the results in any way?

Answer.

The auto-mpg sample dataset is loaded the orange application with the origin is set as a target attribute type which is shown in Figure 1.

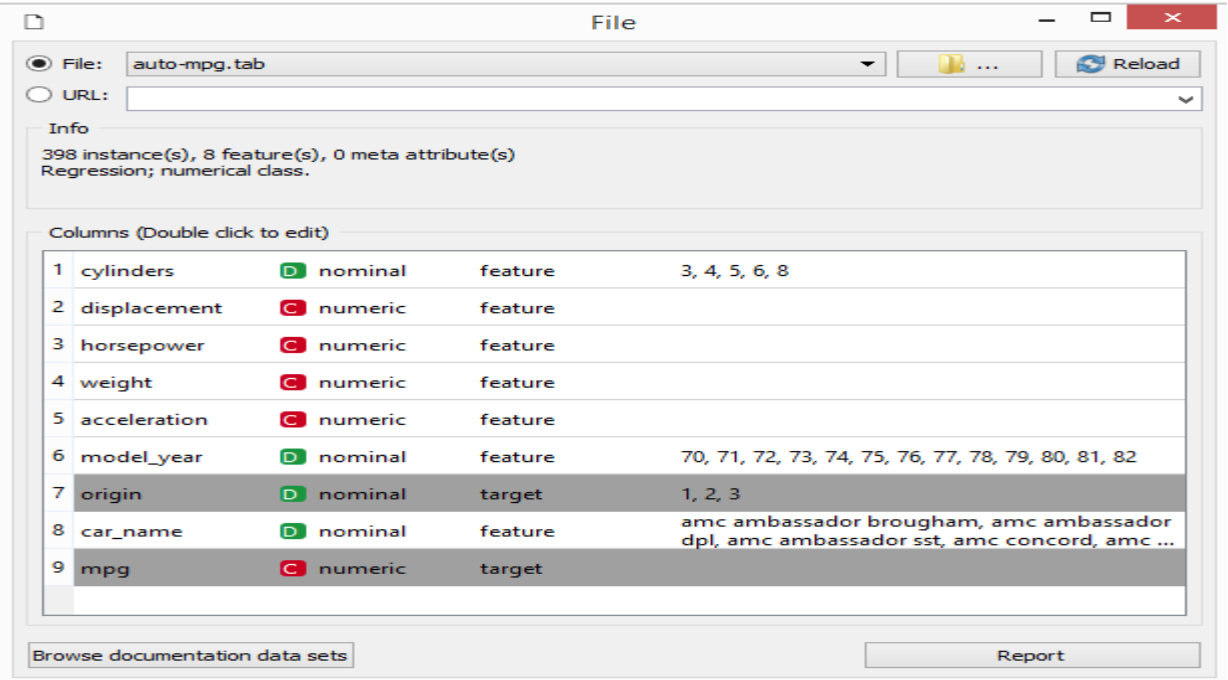


Figure 1. Origin is set as target.

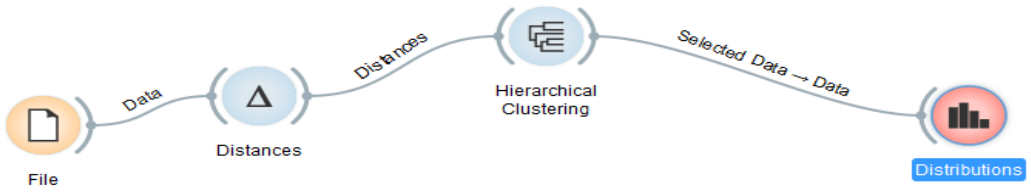


Figure 2. Overall diagram

Hierarchal clustering is done using the linkage set to the average, after calculating the distances with pruning set to the max depth of 5 which is shown in figure 2 and figure 3.

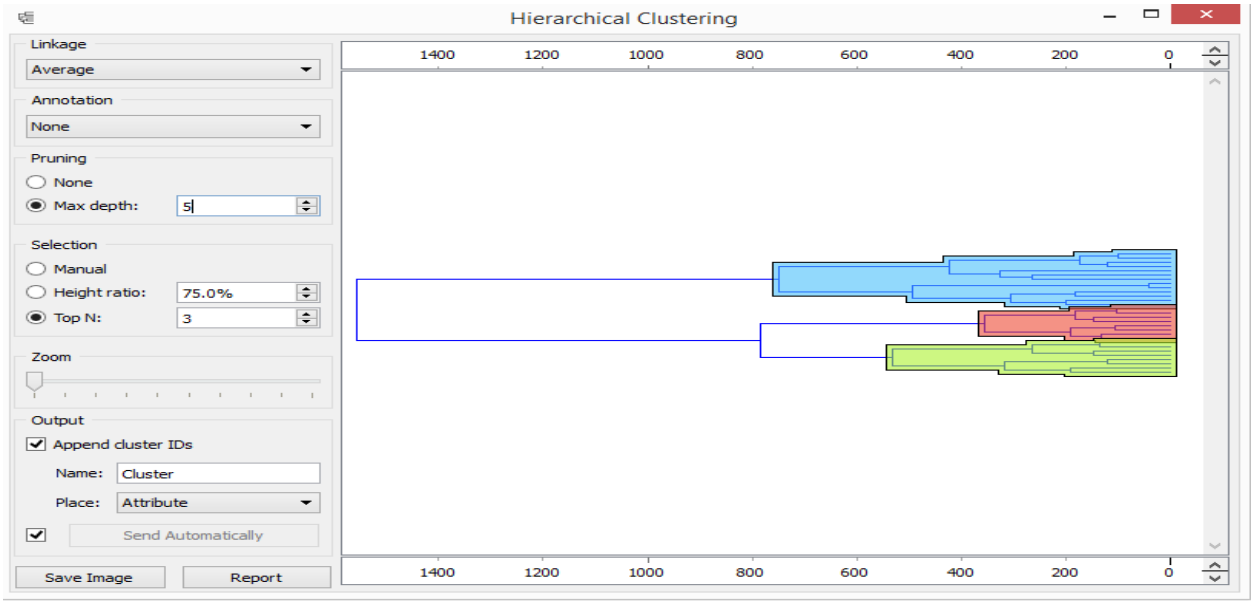


Figure 3. Hierarchical clustering

Resulting clusters are examined using the distribution analysis which is shown in figure 4.

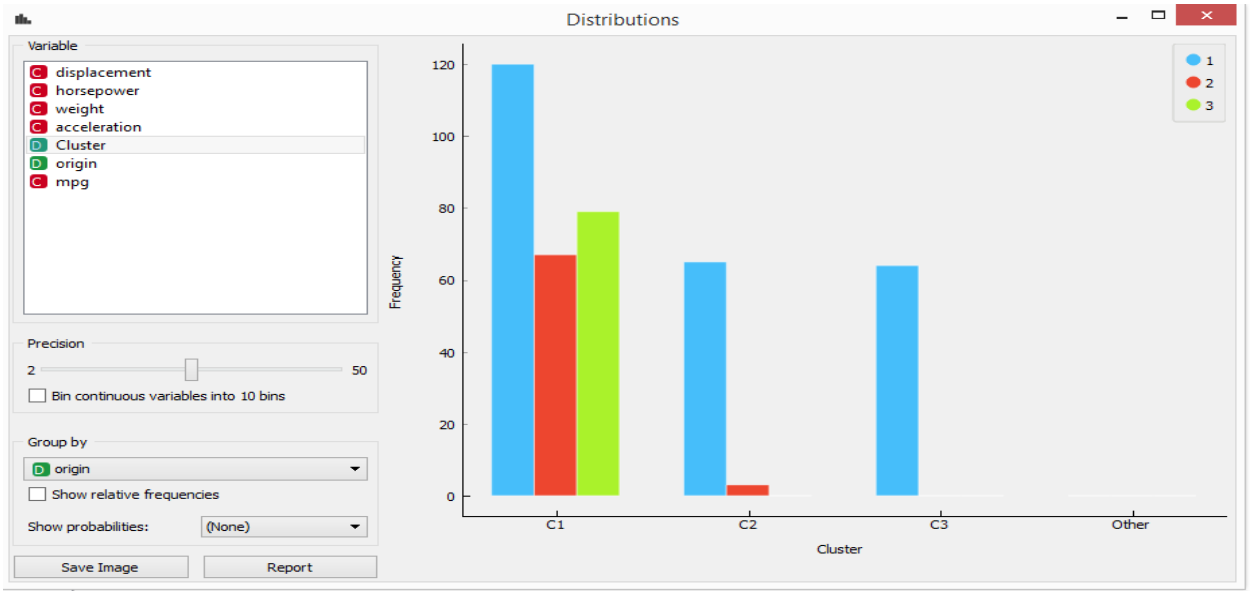


Figure 4. Distribution analysis of resulting clusters.

From the figure 4, the relation between cluster assignment and class label is not clear. As we can see from the diagram for C1 there are all the three class label. But for C2 there are only 2 class label, and for c3 only one class label. So the relation is not clear. Even if we change the depth size the relation remains same.

The probability calculated for each value of origin for each cluster is

For C1 : P(1)=0.451, P(2)=.252, P(3)= .297

For C2: P(1)=.956,P(2)=.044

For C3: As here only one label is present so we cannot determine probability.

Max Depth doesn't effect the result in any way. If make its value to default, the result is same.

Problem 2. Load the breast-cancer-wisconsin-cont dataset into the Orange application, and run a k-means analysis with the number of clusters Optimized From values for k from 2 to 5. Use Silhouette scoring - what is the score for each value of k? For the best score, what are the coordinates of the centroids? What are the distances between the centroids for the best score?

Answer.

First the breast-cancer-wisconsin-cont dataset is loaded into the orange application. A k-means analysis with the number of clusters optimized from values for k from 2 to 5 which is shown in figure 1. Using the silhouette scoring we can see the score for each value of k which is shown in figure 1. The coordinates of the centroids can be seen by creating a data table with centroid as data which is shown in figure 3.

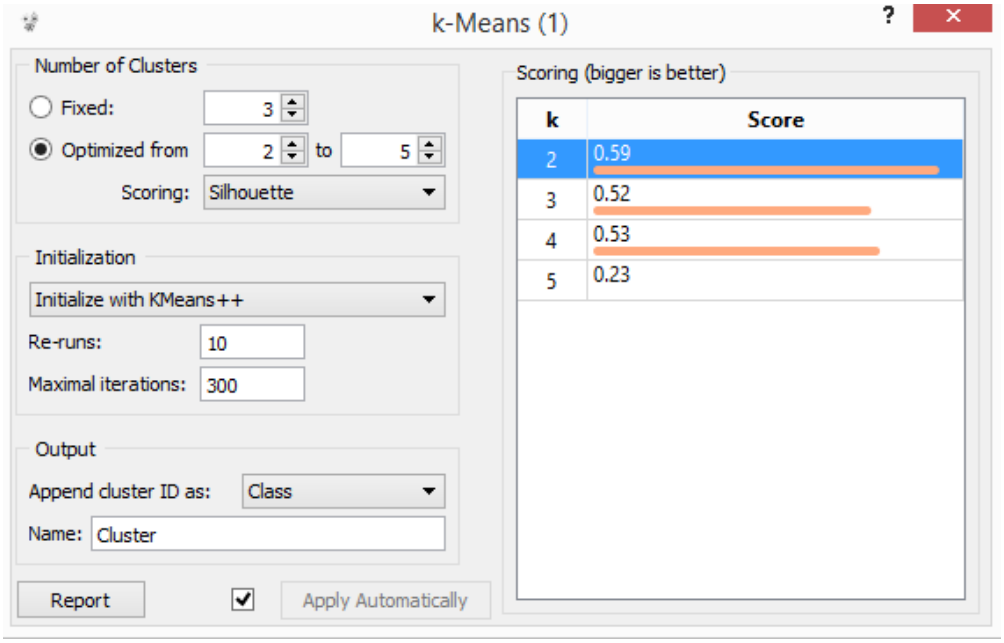


Figure 1. k-Means for the data with optimized values.

Data Table

	Clump thickness	Unif_Cell_Size	Unif_Cell_Shape	Marginal_Adhesio	Single_Cell_Size	Bare_Nuclei	Bland_Chromatin	Normal_Nucleoli	Mitoses
1	2.597	0.805	0.946	0.844	1.619	0.849	1.606	0.793	0.620
2	6.700	6.360	6.289	5.286	4.988	7.509	5.624	5.541	2.108

Figure 2. Coordinates for the centroid.

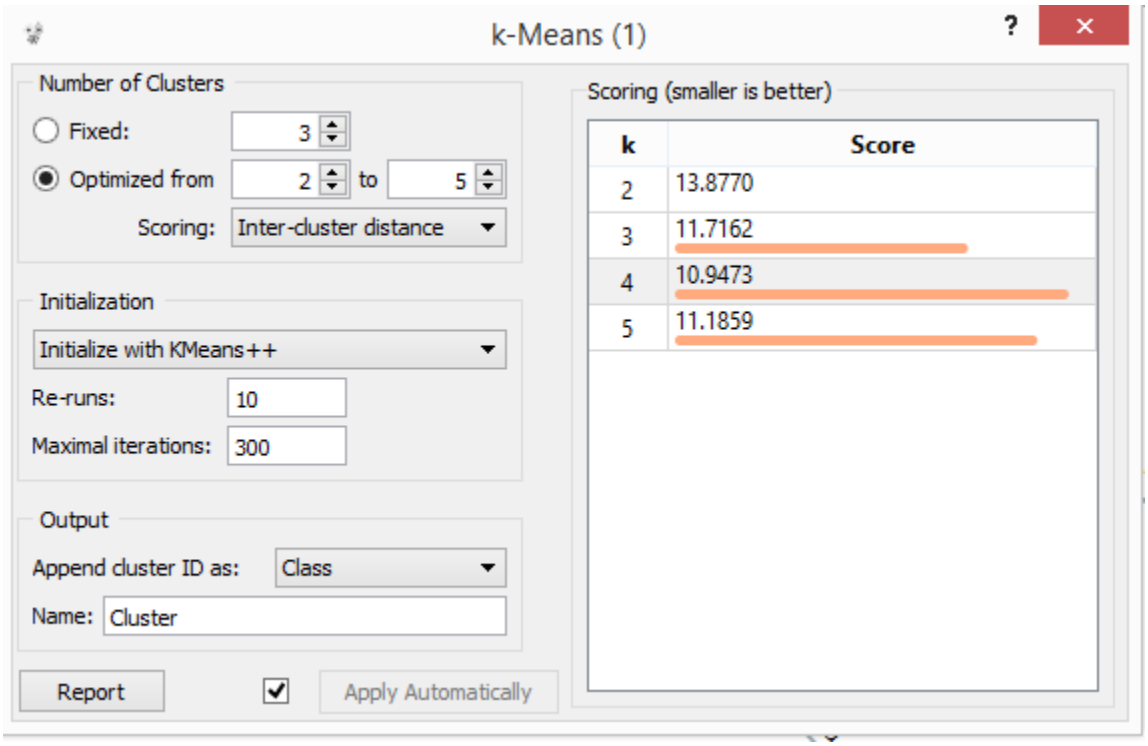


Figure 3. Inter-cluster distance with best score.

Value 13.8770 is the distance between the centroids for the best score.

Problem 3

Load the Boston dataset (sklearn.datasets.load boston()) into Python using a Pandas dataframe. Perform a K-Means analysis on unscaled data, with the number of clusters ranging from 2 to 6. Provide the Silhouette score to justify which value of k is optimal. What information do the values of Homogeneity/Completeness provide as well? Calculate the mean values for all features in each cluster for the optimal clustering - how do these values differ from the centroid coordinates?

Answer.

The Boston dataset is loaded to the python using the Pandas dataframe. K-means analysis is performed on unscaled data, with the number of cluster ranging from 2 to 6 which is shown in the figure 1. The value of k=3 is optimal as it has the maximum silhouette score which can be seen in the output in figure 2. A clustering result satisfies completeness if all the data points that are the member of a given class are elements of the same cluster on the other hand a clustering results satisfies homogeneity if all the clusters contain only data points which are members of a single class. For both the metric is independent of the absolute values of the label which can be seen in the figure 2. The mean value differs from the centroid where the cluster is not uniform that is scattering is not proper. If scattering is uniform then the centroid and the mean value are the same.

```
from sklearn import datasets

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics.cluster import homogeneity_score
from sklearn.metrics.cluster import completeness_score

boston = datasets.load_boston()
range_n_clusters = [2, 3, 4, 5, 6]

for n_clusters in range_n_clusters:

    clusterer = KMeans(n_clusters=n_clusters)
    labels_pred = clusterer.fit_predict(boston.data)
    # 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(boston.data,labels_pred)
    print("For n_clusters =", n_clusters)
    print("The average silhouette_score is :", silhouette_avg)
    target_labels = boston.target
    print("The homogeneity score is :",homogeneity_score(target_labels, labels_pred))
    print("The completeness_score is :",completeness_score(target_labels,
                                labels_pred))
    print("")
```

Figure 1. The screenshot of the code.

```
For n_clusters = 2
The average silhouette_score is : 0.691398118833
The homogeneity score is : 0.070186194715
The completeness_score is : 0.627029136728

For n_clusters = 3
The average silhouette_score is : 0.723403034161
The homogeneity score is : 0.0921583560761
The completeness_score is : 0.639770837023

For n_clusters = 4
The average silhouette_score is : 0.568219170853
The homogeneity score is : 0.135137885887
The completeness_score is : 0.601684007444

For n_clusters = 5
The average silhouette_score is : 0.570738665513
The homogeneity score is : 0.148658835525
The completeness_score is : 0.620034051928

For n_clusters = 6
The average silhouette_score is : 0.487892222354
The homogeneity score is : 0.190855562956
The completeness_score is : 0.633794034428
```

Figure 2. Output screen.

4)

a) First Calculate probability for one point in a sample size K varying b/w 2 & 100.

$$P = \frac{\text{Number of ways to select one centroid from each cluster}}{\text{Number of ways to select K}}$$

$$= \frac{k!}{k^k}$$

$K = 2$; $P = \frac{2!}{2^2} = 0.5$

$K = 3$; $P_3 = \frac{3!}{3^3} = 0.22$

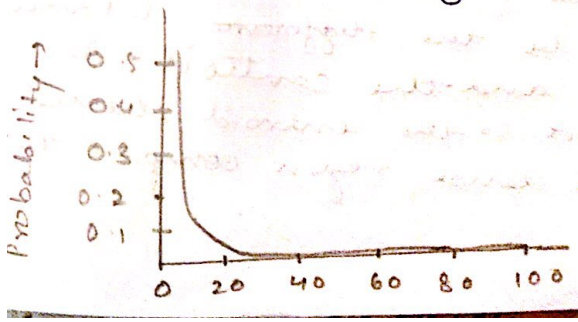
$K = 4$; $P_4 = \frac{4!}{4^4} = 0.09$

$K = 5$; $P_5 = \frac{5!}{5^5} = 0.0384$

$K = 6 \Rightarrow P_6 = \frac{6!}{6^6} = 0.0154$

⋮

As we go further more, we will observe that the probability is coming close to zero. So a graph can be plotted as:



4

b) $P = \frac{K!}{K^K}$

As sample size is $2K$.

$P = \frac{2K}{K^K}$ as the sample size is $2K$

Now if the value of $K = 10$ then,

$P = \frac{2K!}{K^K} = \frac{2 \times (10!)}{10^{10}} = 0.000728$

For $K = 100$;

$P = \frac{2K!}{K^K} = \frac{(2) \times (100!)}{100^{100}} = 1.867 \times 10^{-44} \simeq 0$

For $K = 1000$;

$P = \frac{2K!}{K^K} = \frac{2 \times (1000!)}{1000^{1000}} \simeq 0$

7. In order to minimize the squared error when finding K cluster, more centroids should be allocated to the denser region. As the cluster have more points in it, The squared error will be the ~~difference~~ distance between the points and the centroid. As the points are more so the centroid should be more in a more dense region compare to other. A

71.

Total SSE is the sum of the SSE for each separate attribute. If SSE for one variable is low for all clusters then it means that the variable is constant and is of no use in dividing the data into groups. If the SSE is low for one cluster, then this ~~cluster~~ attribute defines the cluster.

If SSE is high for all clusters then it could be a noise

If the SSE of an attribute is high for one cluster that it will be noise for that cluster & will not provide any useful data from that. We could use the idea of eliminating the attributes that have poor distinguishing power b/w the clusters. As both ~~the~~ high and low are useless for clustering so the idea of removing it is good

17.

a) One dimensional points $\{6, 12, 18, 24, 30, 42, 45\}$

Now we have to assign create two centroids cluster by assigning each point to the nearest centroid.

∴ for $\{18, 45\}$ two clusters would be: $\{6, 12, 18, 24, 30\}$ with 18 as centroid & other would be $\{42, 45\}$ as 42 is more close to 45.

Centroid = 18

∴ Error = $(18-6)^2 + (18-12)^2 + (18-18)^2 + (18-24)^2 + (18-30)^2$

= 360

In the same way for the second cluster error comes out to be 18.

Total error for both the clusters is

360 + 18 = 378

b) There would not be any change in the solution as there is no weak condition present as per now and there will not be any further iteration. The above solution is a stable for centroid.

c)

c) Single link refers to the proximity of two cluster i.e. the minimum of the distance between a point of the centroid. So if centroid is {18, 45} then two clusters would be:

{6, 12, 18, 24, 30} & {42, 48}

d) MIN or Single link seems to produce the most natural clustering as it does not take error. the clustering with the lowest squared

e) MIN or single link can be center-based. As if we see one set of center, it will give the desired result of cluster. Also it Min or single link technique can be contigo -us cluster as different cluster shares the same border.

f) K-means objective is to minimize the squared error. For that it breaks a large clusters into small ones. Also the cluster should be well seperated from one another. That is why it is unnatural one.

21.

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Total
#1	1	1	0	11	4	676	693
#2	27	89	333	827	253	33	1562
#3	326	465	8	105	16	29	949
Total	354	555	341	943	273	738	3204

Entropy :-

Entropy of a cluster $= \sum_{j=1}^{L} P_{ij} \log_2 P_{ij}$

$P_{ij} = m_{ij}/m_i$

i.e Probability that a is a member of class i belongs to class j & L = no. of class

m_{ij} & m_i are the number of objects of class j in cluster & no. of objects in cluster i.

$\therefore$ Entropy #1 $= -\frac{1}{693} \log \frac{1}{693} - \frac{1}{693} \log \frac{1}{693} - \frac{11}{693} \log \frac{11}{693}$

$- \frac{4}{693} \log \frac{4}{693} - \frac{676}{693} \log \frac{676}{693}$

$= 0.2$

Similarly we can calculate

Entropy #2 = 1.84

Entropy #3 = 1.70

Total entropy = $\frac{693}{3204} \times 0.2 + \frac{1562}{3204} \times 1.84 + \frac{949}{3204} \times 1.70$

$= 1.44$

Now Purity of a cluster = $\max_i P_{ij}$

& Overall purity = $\sum_{i=1}^{k} \frac{m_i}{m} P_i$

Purity [cluster #1] = $\frac{676}{693} = 0.98$

Purity #2 = $\frac{827}{1562} = 0.53$

Purity #3 = $\frac{465}{949} = 0.489$

0.489

Overal Purity = $\frac{693}{3204} \times 0.98 + \frac{1562}{3204} \times 0.53 + \frac{949}{3204} \times$

$= 0.61$

22.

a) Yes, there will be difference between the two sets of points. The one which is distributed uniformly will have the uniform density throughout the unit square while the other will have non uniform density which means it will have either high density at some points & low density at some points.

b) With K=10 clusters, random set of points will have a lower SSE then the uniform one.

c) In the uniform dataset, DBSCAN will merge all the points into one cluster. After that it will check the threshold and it will classify them accordingly. On the other hand in density based clustering, regions of higher density are seperated by regions of lower based.

23.

Table of cluster label

Point	Cluster label
P_1	1
P_2	1
P_3	2
P_4	2

Similarly matrix

Point	P_1	P_2	P_3	P_4
P_1	1	0.8	0.65	0.55
P_2	0.8	1	0.7	0.6
P_3	0.65	0.7	1	0.9
P_4	0.55	0.6	0.9	1

We have two cluster i.e with the cluster label 1 & 2 which are $\{P_1, P_2\}$ & $\{P_3, P_4\}$.

For calculating Silhoutte coefficient for points, we need to calculate distance matrix.

Point	P_1	P_2	P_3	P_4
P_1	0	0.2	0.35	0.45
P_2	0.2	0	0.3	0.4
P_3	0.35	0.35	0	0.1
P_4	0.45	0.4	0.1	0

Now Silhoutte coefficient for points $= 1 - a/b$

a = diverage distance from one point to other in same cluster

b = average distance of a point to point in another cluster.

for Point P_1: $SC_1 = 1 - a/b = 1 - \frac{0.2}{(0.35 + 0.45)/2}$

$= \underline{0.5}$

For Point P_2: $SC_2 = 1 - 0.2/(0.7)/2 = \underline{0.428}$

Similarly $SC_3 = \underline{0.69}$

$SC_4 = \underline{0.764}$

Average for SC for cluster 1 = $\frac{0.5 + 0.428}{2}$

$= 0.464$

Average for cluster 2 = $\frac{0.69 + 0.7}{2}$

$= 0.727$

Overage average SC = $\left(\frac{0.464 + 0.727}{2}\right)$

$= 0.5555$

24.

Table of cluster label

Point	Cluster label
P_1	1
P_2	1
P_3	2
P_4	2

Similarity matrix

Point	P_1	P_2	P_3	P_4
P_1	1	0.8	0.65	0.55
P_2	0.8	1	0.7	0.6
P_3	0.65	0.7	1	0.9
P_4	0.55	0.6	0.9	1

Ideal Similarity matrix.

Point	P_1	P_2	P_3	P_4
P_1	1	1	0	0
P_2	1	1	0	0
P_3	0	0	1	1
P_4	0	0	1	1

Now to find SD where $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$

$$\sigma = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \mu_i)}{n}}$$

For x $\mu_x = \frac{4.2}{6} = 0.7$

$\sigma_x = 0.13$

For y $\mu_y = 2/6 = 0.33$

$\sigma_y = 0.52$

$$\text{cov}(x,y) = \frac{1}{n-1} \sum_{n=1}^{n} (x_n - \mu_i)(y_n - \mu_y)$$

$$= -0.200$$

$$\text{Corr}(x,y) = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y} = -0.227$$