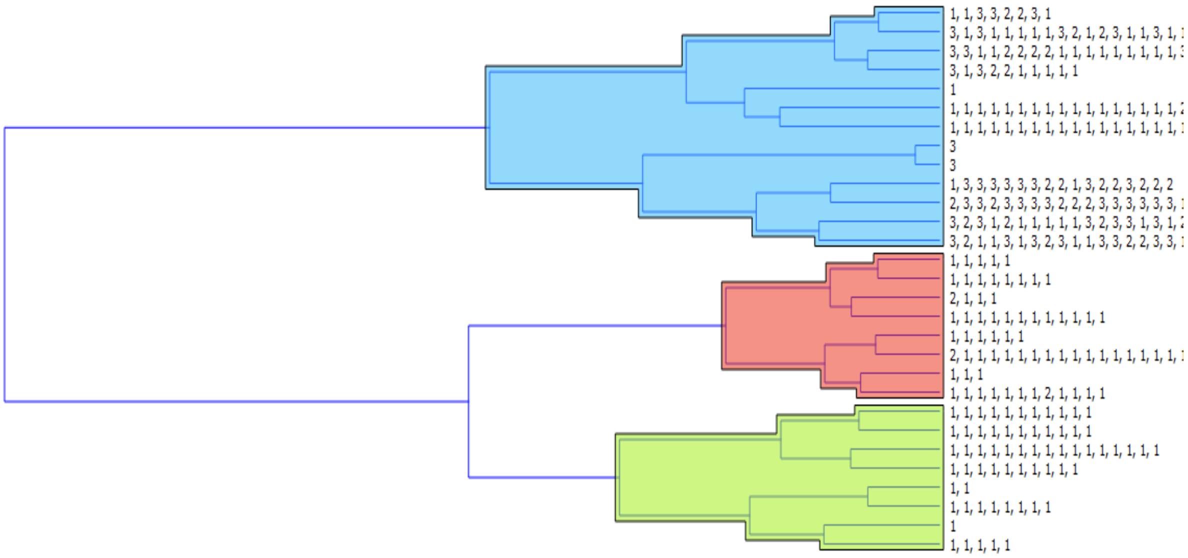
PRACTICUM PROBLEMS

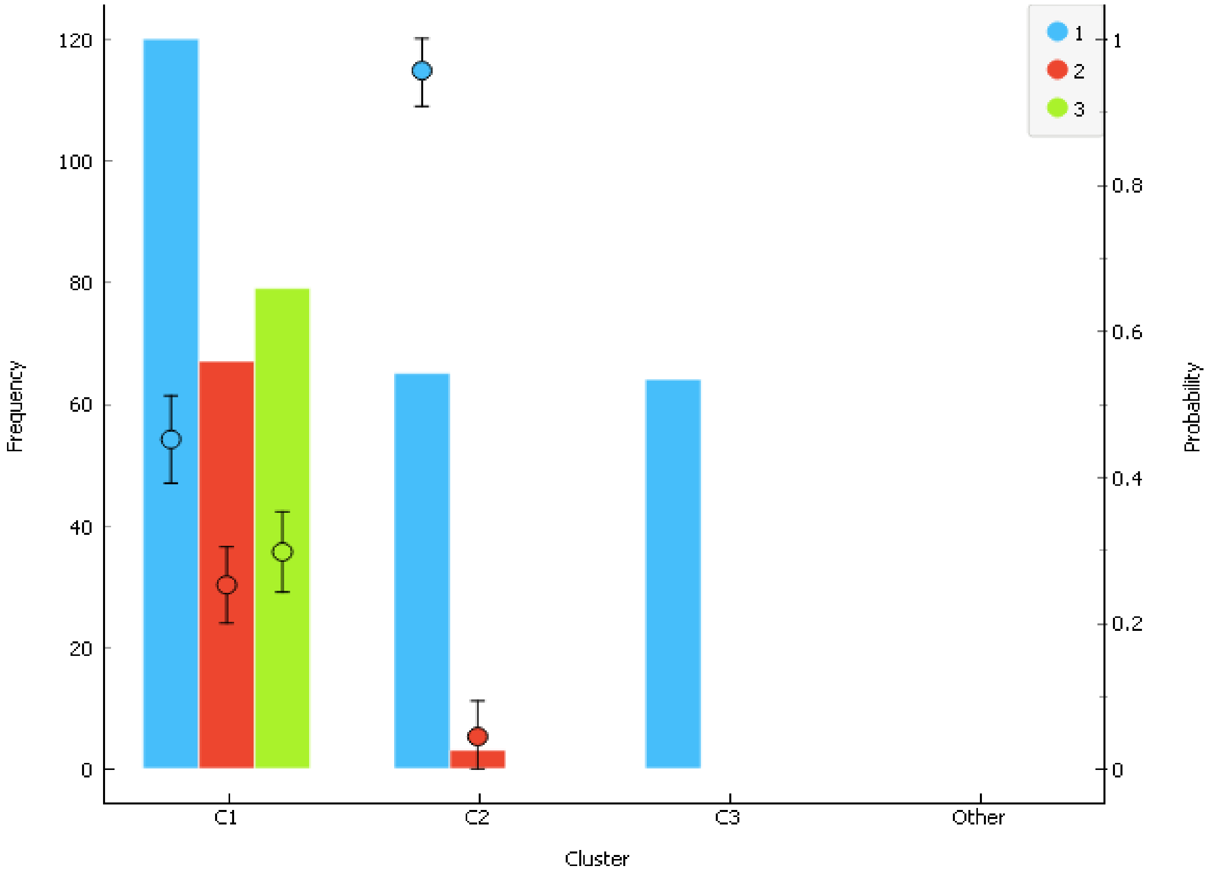
2.1 Problem 1

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:



As seen in the figure after distribution analysis, we can observe that there no clear relationship between cluster assignments and class labels. Clusters are not necessarily formed according to the class labels. Here, only Cluster C3 is pure or homogeneous with all the classes labelled as 1.Also, C3 has more entropy as compared to Cluster C2.



The probabilities calculated for each value of origin of each clusters are as below:

For Cluster C1:

1 => 0.451 + or - 0.060

2 => 0.252 + or - 0.052

3 => 0.297 + or - 0.055

For Cluster C2:

1 => 0.956 + or - 0.049

2 => 0.044 + or - 0.049

For Cluster C3:

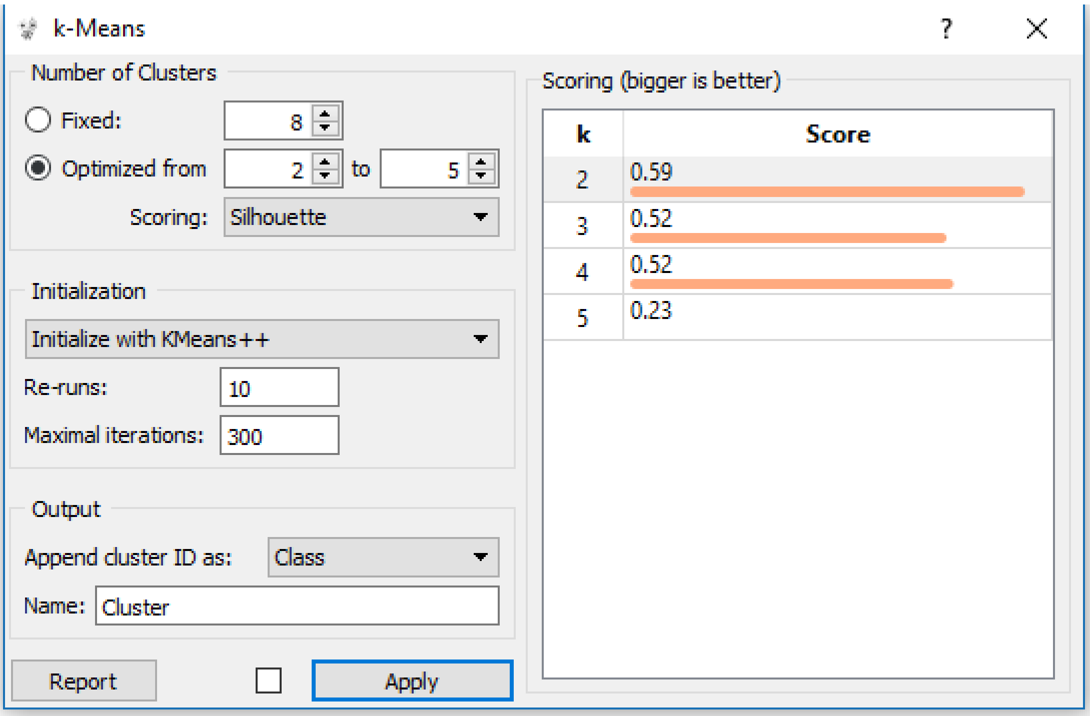
1 => 1

No, changing the max depth does not change the result in anyway.

2.2 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:



As seen the best silhouette score is obtained for 2 clusters.

The centroid of the clusters for optimized value of k=2 is as seen in the figure below.

	Clump thickness	Unif_Cell_Size	Unif_Cell_Shape	larginal_Adhesio	Single_Cell_Size	Bare_Nuclei	land_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

The distance between centroid for the best score is 13.877

Distance Matrix

	1	2
1		13.877
2	13.877	

2.3 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:

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

n_clusters = 0
for n_clusters in range_n_clusters:


    clusterer = KMeans(n_clusters=n_clusters,  init='k-means++')
    clusterer.fit(df)
    cluster_labels = clusterer.fit_predict(df)


    silhouette_avg = metrics.silhouette_score(df, cluster_labels)

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

```
For n_clusters = 2 The average silhouette_score is : 0.691398118833
For n_clusters = 3 The average silhouette_score is : 0.723403034161
For n_clusters = 4 The average silhouette_score is : 0.568219170853
For n_clusters = 5 The average silhouette_score is : 0.570738665513
For n_clusters = 6 The average silhouette_score is : 0.501258930507
```

So as seen in the above figure, silhouette score is high when we optimize clustering to 3 clusters.

The values for homogeneity and completeness are as follows: 0.187370799835 0.629506604287

So as the value of completeness is more it states that all the data points that are members of a given class are elements of the same cluster.

Larger values of homogeneity and completeness are desirable.

Mean Values:

For Cluster 1

```
C0 = df.loc[df['CLUST'] == 0]
```

```
C0.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
count	11.000000	11.0	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000
mean	1.963207	0.0	16.708182	0.090909	0.707727	5.916091	91.818182	2.323691	4.727273	386.909091	17.000000	187.546364	17.212
std	0.912947	0.0	5.457133	0.301511	0.159456	0.312366	7.972555	0.874302	0.467099	41.353245	3.191865	74.268586	6.0352
min	0.228760	0.0	8.140000	0.000000	0.520000	5.272000	79.200000	1.419100	4.000000	307.000000	14.700000	70.800000	9.8100
25%	1.500405	0.0	14.070000	0.000000	0.571500	5.733000	84.000000	1.679800	4.500000	393.500000	14.700000	128.950000	13.580
50%	2.149180	0.0	19.580000	0.000000	0.624000	5.950000	94.000000	2.283400	5.000000	403.000000	14.700000	227.610000	16.140
75%	2.413010	0.0	19.580000	0.000000	0.871000	6.115500	98.450000	2.570300	5.000000	403.000000	20.950000	244.235000	18.825
max	3.535010	0.0	21.890000	1.000000	0.871000	6.405000	100.000000	3.990000	5.000000	437.000000	21.200000	262.760000	27.800

For Cluster 2

```
C1 = df.loc[df['CLUST'] == 1]
```

```
C1.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LS
count	80.000000	80.00000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80
mean	0.081582	24.16875	6.226500	0.087500	0.464586	6.577125	49.226250	4.942381	3.550000	225.450000	17.892500	391.370625	8.
std	0.071202	32.25609	6.345701	0.284349	0.048092	0.660163	24.182559	1.813620	1.330271	21.745886	1.513716	8.875882	5.
min	0.013110	0.00000	0.460000	0.000000	0.385000	5.399000	2.900000	1.757200	1.000000	187.000000	13.600000	341.600000	1.
25%	0.034833	0.00000	2.460000	0.000000	0.439000	6.012250	32.175000	3.917500	3.000000	216.000000	17.600000	389.632500	4.
50%	0.057575	0.00000	5.070000	0.000000	0.449000	6.524500	45.750000	5.033750	3.000000	224.000000	17.900000	394.175000	6.
75%	0.096653	40.00000	6.910000	0.000000	0.488000	7.004500	62.050000	5.873750	4.000000	243.000000	18.700000	396.900000	9.
max	0.387350	100.00000	25.650000	1.000000	0.581000	8.034000	97.000000	12.126500	7.000000	265.000000	20.200000	396.900000	30

For Cluster 3

```
C2 = df.loc[df['CLUST'] == 2]
```

```
C2.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
count	102.000000	102.0	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.0000
mean	10.910511	0.0	18.572549	0.078431	0.671225	5.982265	89.913725	2.077164	23.019608	668.205882	20.195098	371.8030
std	12.120759	0.0	2.091641	0.270177	0.062720	0.722131	13.275049	0.672498	4.339504	9.763884	0.021698	35.00609
min	0.105740	0.0	18.100000	0.000000	0.532000	3.561000	40.300000	1.129600	4.000000	666.000000	20.100000	240.5200
25%	4.844605	0.0	18.100000	0.000000	0.614000	5.619500	87.675000	1.575675	24.000000	666.000000	20.200000	354.8475
50%	7.795775	0.0	18.100000	0.000000	0.693000	6.113000	95.350000	1.904700	24.000000	666.000000	20.200000	389.3650
75%	12.613775	0.0	18.100000	0.000000	0.713000	6.391250	98.775000	2.508125	24.000000	666.000000	20.200000	396.9000
max	88.976200	0.0	27.740000	1.000000	0.770000	8.780000	100.000000	4.098300	24.000000	711.000000	20.200000	396.9000

Centroid for K=3

```
centers = clust_model1.cluster_centers_
roundc= np.round(centers,1)
print(roundc)
```

```
[[ 1.09000000e+01   0.00000000e+00   1.86000000e+01   1.00000000e-01
   7.00000000e-01   6.00000000e+00   8.99000000e+01   2.10000000e+00
   2.30000000e+01   6.68200000e+02   2.02000000e+01   3.71800000e+02
   1.79000000e+01   1.74000000e+01   0.00000000e+00]
 [ 4.00000000e-01   1.57000000e+01   8.40000000e+00   1.00000000e-01
   5.00000000e-01   6.40000000e+00   6.04000000e+01   4.50000000e+00
   4.50000000e+00   3.11200000e+02   1.78000000e+01   3.83500000e+02
   1.04000000e+01   2.49000000e+01   1.50000000e+00]
 [ 1.50000000e+01  -0.00000000e+00   1.79000000e+01   0.00000000e+00
   7.00000000e-01   6.10000000e+00   8.99000000e+01   2.00000000e+00
   2.25000000e+01   6.44700000e+02   1.99000000e+01   5.78000000e+01
   2.04000000e+01   1.31000000e+01   2.00000000e+00]]
```

There is not much difference between mean values for all features and centroid for each cluster. Centroid is more precise than mean and is used as a measure of cluster location.