Homework 4.0

1.1 Problem 1

Load the auto-mpg sample dataset from the UCI Machine Learning Repository (auto-mpg.data) into Python using a Pandas dataframe. Using only the continuous fields as features, impute any missing values with the mean, and perform Hierarchical Clustering (Use sklearn.cluster.AgglomerativeClustering) with linkage set to average and the default affinity set to a euclidean. Set the remaining parameters to obtain a shallow tree with 3 clusters as the target. Obtain the mean and variance values for each cluster and compare these values to the values obtained for each class if we used origin as a class label. Is there a Clear relationship between cluster assignment and class label?

The mean and variance values for each cluster are compared below with those obtained for each class when using the origin field as the class label.

```
Hierarchical Cluster Stats:
                            mpg            displacement               \
                           mean        var         mean          var
hierarchical_cluster
0                     26.177441  41.303375   144.304714  3511.485383
1                     14.528866   4.771033   348.020619  2089.499570
2                     43.700000   0.300000    91.750000    12.250000

                     horsepower                  weight                 \
                           mean         var         mean            var
hierarchical_cluster
0                     86.120275  294.554450  2598.414141  299118.709664
1                    161.804124  674.075816  4143.969072  193847.051117
2                     49.000000    4.000000  2133.750000   21672.916667

                     acceleration
                             mean       var
hierarchical_cluster
0                       16.425589  4.875221
1                       12.641237  3.189948
2                       22.875000  2.309167

Origin Class Stats:
              mpg             displacement               horsepower  \
             mean        var          mean          var        mean
origin
1       20.083534  40.997026    245.901606  9702.612255  119.048980
2       27.891429  45.211230    109.142857   509.950311   80.558824
3       30.450633  37.088685    102.708861   535.465433   79.835443

                       weight                   acceleration
              var        mean             var          mean       var
origin
1     1591.833657  3361.931727  631695.128385     15.033735  7.568615
2      406.339772  2423.300000  240142.328986     16.787143  9.276209
3      317.523856  2221.227848  102718.485881     16.172152  3.821779
```

The contingency table compares the cluster assignments obtained from hierarchical clustering with the actual class labels represented by the origin field.

```
Hierarchical vs Origin:
 hierarchical_cluster    0   1  2
origin
1                      152  97  0
2                       66   0  4
3                       79   0  0
```

The results show that the relationship between the clusters and the class labels is not clearly defined. Vehicles labeled as origin 1 (USA) are spread across cluster 0 and cluster 1, while origin 2 (Europe) appears mostly in cluster 0 with a few in cluster 2. Additionally, origin 3 (Japan) is entirely assigned to cluster 0. This distribution indicates that the clustering algorithm did not effectively separate the data based on the origin classes, and there is no strong alignment between cluster assignments and class labels.

1.2 Problem 2

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

k	Silhouette score
2	0.3501
3	0.2753
4	0.2027
5	0.2640
6	0.2400

The optimal number of clusters is k=2, as it yields the highest silhouette score of 0.2700 among all tested values.The silhouette score measures how well each sample is clustered, considering both cohesion and separation, with higher values indicating better-defined clusters. Since the score is highest when k=5, this implies that the clusters are more compact and well-separated at this value.

I first calculated the mean values for all features in each cluster for the optimal clustering, and the results are shown below.

```
Mean values of all features in each cluster for the optimal clustering:
                    crim         zn      indus      chas       nox        rm  \
kmeans_cluster
0               0.263946  17.477204   6.919818  0.069909  0.487215  6.456544
1               9.839575   0.000000  18.975085  0.067797  0.680124  5.965096

                      age       dis        rad         tax    ptratio  \
kmeans_cluster
0               56.382067  4.751124   4.474164  302.209726  17.818237
1               91.238418  2.017920  18.983051  605.316384  19.640113

                         b      lstat       medv
kmeans_cluster
0               386.643891   9.417812  25.782067
1               300.967345  18.666610  16.493220
```

Next, I calculated the centroid coordinates under the standardized data space, and the results are shown below.

```
The centroid coordinateMean values in each cluster:
       crim        zn     indus      chas       nox        rm       age  \
0 -0.389801  0.262392 -0.615294  0.002912 -0.582916  0.244913 -0.433584
1  0.724546 -0.487722  1.143682 -0.005412  1.083499 -0.455233  0.805928

        dis       rad       tax   ptratio         b     lstat      medv
0  0.454491 -0.583452 -0.629727 -0.294662  0.328600 -0.453497  0.353641
1 -0.844789  1.084495  1.170509  0.547705 -0.610788  0.842942 -0.657333
```

I also calculated the centroid coordinates in the original data space by applying inverse transformation to the standardized centroids.

```
The centroid coordinateMean values (in original space) in each cluster:
       crim            zn      indus      chas       nox        rm        age  \
0  0.263946  1.747720e+01   6.919818  0.069909  0.487215  6.456544  56.382067
1  9.839575  1.243450e-14  18.975085  0.067797  0.680124  5.965096  91.238418

        dis        rad         tax    ptratio           b      lstat  \
0  4.751124   4.474164  302.209726  17.818237  386.643891   9.417812
1  2.017920  18.983051  605.316384  19.640113  300.967345  18.666610

       medv
0  25.782067
1  16.493220
```

The two types of centroid coordinates were then compared with the mean values from the optimal clustering results, and the comparison results are shown below.

Mean values of all features in each cluster vs. Centroid coordinates (in scaled space):

	crim	zn	indus	chas	nox	\
cluster0_mean	0.263946	17.477204	6.919818	0.069909	0.487215	
cluster1_mean	9.839575	0.000000	18.975085	0.067797	0.680124	
cluster0_centroid	-0.389801	0.262392	-0.615294	0.002912	-0.582916	
cluster1_centroid	0.724546	-0.487722	1.143682	-0.005412	1.083499	

	rm	age	dis	rad	tax	\
cluster0_mean	6.456544	56.382067	4.751124	4.474164	302.209726	
cluster1_mean	5.965096	91.238418	2.017920	18.983051	605.316384	
cluster0_centroid	0.244913	-0.433584	0.454491	-0.583452	-0.629727	
cluster1_centroid	-0.455233	0.805928	-0.844789	1.084495	1.170509	

	ptratio	b	lstat	medv
cluster0_mean	17.818237	386.643891	9.417812	25.782067
cluster1_mean	19.640113	300.967345	18.666610	16.493220
cluster0_centroid	-0.294662	0.328600	-0.453497	0.353641
cluster1_centroid	0.547705	-0.610788	0.842942	-0.657333

Mean values of all features in each cluster vs. Centroid coordinates (in original space):

	crim	zn	indus	chas	nox	\
cluster0_mean	0.263946	1.747720e+01	6.919818	0.069909	0.487215	
cluster1_mean	9.839575	0.000000e+00	18.975085	0.067797	0.680124	
cluster0_centroid	0.263946	1.747720e+01	6.919818	0.069909	0.487215	
cluster1_centroid	9.839575	1.243450e-14	18.975085	0.067797	0.680124	

	rm	age	dis	rad	tax	\
cluster0_mean	6.456544	56.382067	4.751124	4.474164	302.209726	
cluster1_mean	5.965096	91.238418	2.017920	18.983051	605.316384	
cluster0_centroid	6.456544	56.382067	4.751124	4.474164	302.209726	
cluster1_centroid	5.965096	91.238418	2.017920	18.983051	605.316384	

	ptratio	b	lstat	medv
cluster0_mean	17.818237	386.643891	9.417812	25.782067
cluster1_mean	19.640113	300.967345	18.666610	16.493220
cluster0_centroid	17.818237	386.643891	9.417812	25.782067
cluster1_centroid	19.640113	300.967345	18.666610	16.493220

Absolute difference between cluster means and centroid coordinates:

	crim	zn	indus	chas	nox	\
diff_cluster0	6.106227e-16	3.552714e-15	3.552714e-15	6.938894e-17	0.0	
diff_cluster1	1.776357e-15	1.243450e-14	1.065814e-14	5.551115e-17	0.0	

	rm	age	dis	rad	tax	\
diff_cluster0	0.0	0.0	8.881784e-16	8.881784e-16	0.000000e+00	
diff_cluster1	0.0	0.0	0.000000e+00	7.105427e-15	6.821210e-13	

	ptratio	b	lstat	medv
diff_cluster0	0.000000e+00	0.0	0.000000e+00	3.552714e-15
diff_cluster1	3.552714e-15	0.0	7.105427e-15	0.000000e+00

By comparing and calculating the absolute differences, it can be concluded that the centroid coordinates produced by K-Means are essentially the mean values for all features in each cluster after standardization.

1.3 Probelem 3

Load the wine dataset (sklearn.datasets.load wine()) into Python using a Pandas dataframe. Perform a K-Means analysis on scaled data, with the number of clusters set to 3. Given the actual class labels, calculate the Homogeneity/Completeness for the optimal k - what information does each of these metrics provide?

Homogeneity	Completeness
0.8788432003662366	0.8729636016078731

Homogeneity measures whether each cluster contains only members of a single class. Homogeneity=0.8788432003662366 indicates that most clusters contain only members of a single class, which means that the clustering algorithm has achieved high purity and the internal consistency within clusters is strong.

Completeness measures whether all members of a given class are assigned to the same cluster. Completeness=0.8729636016078731 indicates that most samples of each true class are grouped into the same cluster, suggesting that the clustering algorithm effectively captures the underlying class structure and avoids splitting classes across multiple clusters.