Please complete the assigned problems to the best of your abilities. Ensure that your work is entirely your own, external resources are only used as permitted by the instructor, and all allowed sources are given proper credit for non-original content.

1. Practicum Problems

These problems will primarily reference the lecture materials and the examples given in class using Python. It is suggested that a Jupyter/IPython notebook be used for programmatic components.

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?

Although there is a certain correlation between the clustering structure and the "origin" categories, this relationship is partial and ambiguous, and the clustering results cannot be considered a reliable substitute for the class labels.

After calculating the mean and variance of each feature within each cluster, I compared these statistics with those based on using "origin" as the class label. The comparison reveals significant differences in feature distributions across clusters. For example, Cluster 1 shows the highest average horsepower and weight, indicating it contains vehicles with larger engines and greater power; in contrast, Cluster 0 has the highest average mpg, typically representing more fuel-efficient cars.

When analyzing the clustering results alongside the "origin" attribute using a cross-tabulation, it is evident that certain origins are more concentrated in specific clusters. Vehicles from origin 3 (Europe) are mostly found in Cluster 0, while those from origin 1 (USA) are more widely spread but tend to appear more in Cluster 1. Notably, origin 2 (Japan) vehicles almost completely avoid Cluster 1, suggesting that the clustering reflects some regional characteristics. Despite these partial correspondences, the relationship is neither strong nor definitive. Vehicles from multiple origins appear in the same cluster, and origin 1 vehicles are distributed fairly evenly across all three clusters. This indicates that while clustering captures some differences among origins—especially with Cluster 1 representing large American cars and Cluster 0 representing economy cars—it does not clearly distinguish between the origin categories overall.

```
Cluster Mean and Variance:
               mpg              displacement                horsepower  \
              mean        var         mean          var          mean
cluster
0        27.365414  41.976309   131.934211  2828.083391    83.834615
1        13.889062   3.359085   358.093750  2138.213294   167.046875
2        17.510294   8.829892   278.985294  2882.492318   124.470588

                        weight                  acceleration
                var          mean            var         mean         var
cluster
0        368.053623  2459.511278  182632.099872    16.298120    5.718298
1        756.521577  4398.593750   74312.340278    13.025000    3.591429
2        713.088674  3624.838235   37775.809263    15.105882   10.556980

Origin Mean and Variance:
              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

Cluster vs Origin Cross-tab:
origin     1   2   3
cluster
0        120  67  79
1         64   0   0
2         65   3   0
```

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?

From the calculation results,

```
k=2, Silhouette Score=0.3601
k=3, Silhouette Score=0.2448
k=4, Silhouette Score=0.2275
k=5, Silhouette Score=0.2389
k=6, Silhouette Score=0.2291
```

it can be seen that since the Silhouette coefficient is highest when k=2 (0.3601), we choose k=2 as the optimal number of clusters. Next, based on the clustering result with k=2, I calculated the mean values of each feature within each cluster and compared them with the cluster centers . The observation shows that **the feature means within each cluster are almost exactly the same as the centroid coordinates in the original feature space.**

This may because, in each iteration of the K-Means algorithm, the centroids are updated to the mean of all samples in each feature dimension within the cluster. Therefore, when the algorithm converges, the centroid essentially represents the central position of the samples in that cluster—that is, their average. For example, in cluster 0, both the mean of crim and the centroid coordinate are 0.261172, with only minimal numerical differences due to floating-point precision.

```
Cluster Feature Means:
              crim         zn      indus      chas       nox        rm  \
cluster
0         0.261172  17.477204   6.885046  0.069909  0.487011  6.455422
1         9.844730   0.000000  19.039718  0.067797  0.680503  5.967181

                age       dis        rad         tax    ptratio           b  \
cluster
0         56.339210  4.756868   4.471125  301.917933  17.837386  386.447872
1         91.318079  2.007242  18.988701  605.858757  19.604520  301.331695

              lstat       medv
cluster
0          9.468298  25.749848
1         18.572768  16.553107

Centroid Coordinates (scaled features):
       crim        zn     indus      chas       nox        rm       age  \
0 -0.390124  0.262392 -0.620368  0.002912 -0.584675  0.243315 -0.435108
1  0.725146 -0.487722  1.153113 -0.005412  1.086769 -0.452263  0.808760

        dis       rad       tax   ptratio         b     lstat
0  0.457222 -0.583801 -0.631460 -0.285808  0.326451 -0.446421
1 -0.849865  1.085145  1.173731  0.531248 -0.606793  0.829787

Centroid Coordinates (original feature space):
       crim            zn      indus      chas       nox        rm        age  \
0  0.261172  1.747720e+01   6.885046  0.069909  0.487011  6.455422  56.339210
1  9.844730  1.243450e-14  19.039718  0.067797  0.680503  5.967181  91.318079

        dis        rad         tax    ptratio           b      lstat
0  4.756868   4.471125  301.917933  17.837386  386.447872   9.468298
1  2.007242  18.988701  605.858757  19.604520  301.331695  18.572768
```

1.3 Problem 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?

The Homogeneity metric measures whether the samples within each cluster belong to the same actual category. A higher score indicates that the clustering results better preserve the structure of the true labels. In this case, a relatively high score (0.88) means that most samples within each cluster belong to the same category, suggesting good clustering performance.

The Completeness metric assesses whether all samples of a given actual category are assigned to the same cluster. A higher score indicates that the clustering results can effectively group all samples of the same class together. A high score (0.87) suggests that the algorithm successfully clusters samples of the same class into the same group.

Together, these two metrics provide a comprehensive evaluation of clustering quality: Homogeneity focuses on the internal consistency of clusters, while Completeness emphasizes the coverage of each class within clusters. Based on these scores, we can conclude that K-Means clustering performs well on this dataset, effectively separating the samples into three clusters while maintaining high inter-class consistency and intra-class completeness.

```
(0.8788432003662366, 0.8729636016078731, array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2]))
```

END