Homework 4.0

Problem 1: Hierarchical Clustering on Auto-MPG Dataset

1. Introduction

The Auto-MPG dataset, sourced from the UCI Machine Learning Repository, contains information about automobile fuel efficiency, along with attributes such as displacement, horsepower, weight, and origin. This report aims to explore the underlying structure within the dataset by applying hierarchical clustering, specifically focusing on continuous features. The goal is to uncover natural groupings of automobiles and assess the relationship between these clusters and the 'origin' class label (representing the geographic origin of the car).

2. Data Loading and Preprocessing

The Auto-MPG dataset was loaded into a Pandas DataFrame from the UCI Machine Learning Repository URL. The following continuous features were selected for clustering: 'mpg', 'displacement', 'horsepower', 'weight', and 'acceleration'. Missing values in the 'horsepower' feature were imputed using the mean of the available 'horsepower' values to ensure data completeness and compatibility with the clustering algorithm. Specifically, non-numeric entries in 'horsepower' were converted to NaN (Not a Number) and replaced with the calculated mean.

3. Hierarchical Clustering Methodology

Hierarchical clustering was performed using the `sklearn.cluster.AgglomerativeClustering` class from the scikit-learn library. The 'average' linkage criterion was employed, which defines the distance between two clusters as the average distance between all pairs of points, one from each cluster. The default Euclidean distance was used as the affinity metric. The number of clusters, `n_clusters`, was set to 3.

4. Results and Analysis

4.1 Mean and Variance

Results for mean of original data are to be presented in a table1.

Results for variance of original data are to be presented in a table2.

Table 1: Mean of Features by Cluster(Oringinal Data)

Feature	Cluster 0	Cluster 1	Cluster 2
mpg	26.177441	14.528866	43.700000
displacement	144.304714	348.020619	91.750000
horsepower	86.490964	161.804124	49.000000
weight	2598.414141	4143.969072	2133.750000
acceleration	16.425589	12.641237	22.875000

Table 2: Variance of Features by Cluster (Oringinal Data)

Feature	Cluster 0	Cluster 1	Cluster 2
mpg	41.303375	4.771033	0.300000
displacement	3511.485383	2089.499570	12.250000
horsepower	295.270673	674.075816	4.000000
weight	299118.709664	193847.051117	21672.916667
acceleration	4.875221	3.189948	2.309167

4.2 Mean and Variance(used origin)

We used origin as a class label, and the obtained results are as follows

Table 3: Mean of Features by Origin (Original Data)

Feature	Origin 1	Origin 2	Origin 3
mpg	20.083534	27.891429	30.450633
displacement	245.901606	109.142857	102.708861
horsepower	118.814769	81.241983	79.835443
weight	3361.931727	2423.300000	2221.227848
acceleration	15.033735	16.787143	16.172152

Table 4: Variance of Features by Origin (Original Data)

Feature	Origin 1	Origin 2	Origin 3
mpg	40.997026	45.211230	37.088685
displacement	9702.612255	509.950311	535.465433
horsepower	1569.532304	410.659789	317.523856
weight	631695.128385	240142.328986	102718.485881
acceleration	7.568615	9.276209	3.821779

4.3 Comparison of Clustering Results and Origin Class

Table 5: Cluster Distribution by Origin

	Origin 1	**Origin 2**	**Origin 3**
Cluster 0	152	66	79
Cluster 1	97	0	0
Cluster 2	0	4	0

Visual Cluster Class

Visualizations showing the relationship between clusters and the 'origin' class label will be included.

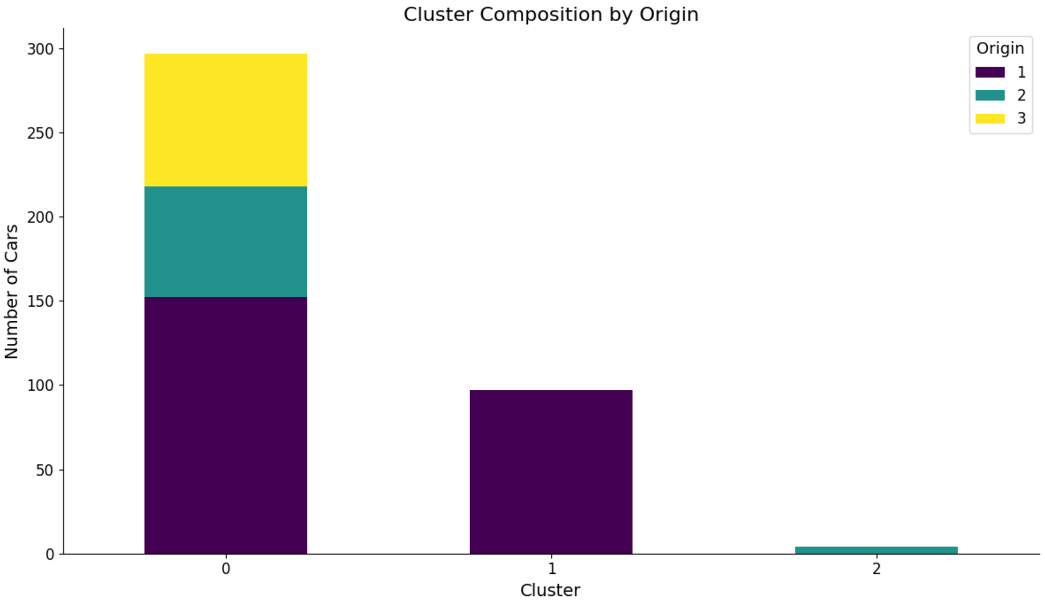

Figure 1: Relationships between specified features

Figure 1 directly visualizes the relationship between cluster assignments and car origins. By displaying the distribution of cars from each origin across different clusters, the chart allows for an assessment of how well the clusters reflect the underlying class structure.

Analysis

1. Cluster 0 Analysis: This cluster exhibits a mixed composition, containing a substantial number of items from all three origins (152 from Origin 1, 66 from Origin 2, and 79 from Origin 3). This indicates that Cluster 0 does not strongly correlate with any single origin.

2. Cluster 1 Analysis: This cluster shows a strong association with Origin 1. Almost all the items within Cluster 1 (97 out of 97) originate from Origin 1. This suggests that the clustering algorithm successfully identified a distinct group composed almost entirely of items from Origin 1. Cluster 1 is, therefore, a good representation of Origin 1 items within this dataset.

3. Cluster 2 Analysis: This cluster is exclusively composed of items from Origin 2. While only containing a small number of items (4), the fact that all members come from Origin 2 indicates a clear relationship between Cluster 2 and Origin 2. The clustering process, in this case, has perfectly isolated a group of Origin 2 items, albeit a small one.

So these are relationship between cluster assignment and class label.

4.4 Relationships between specified features

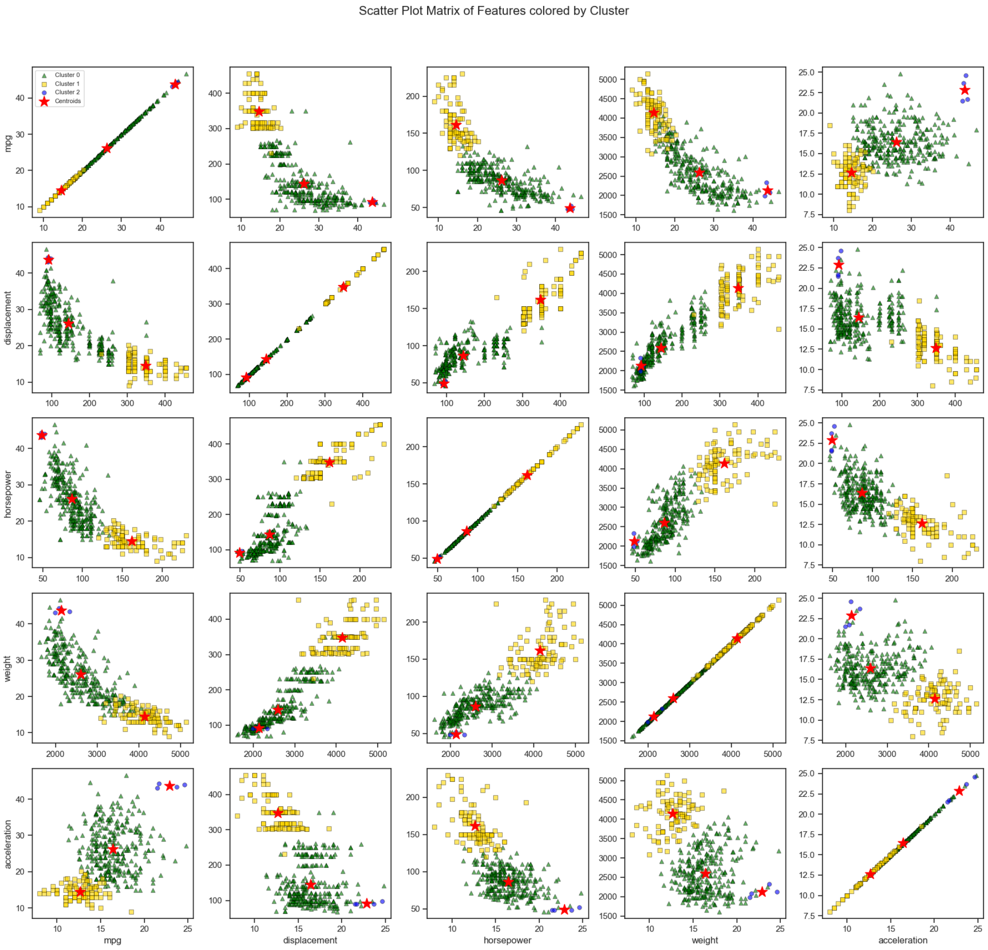

Figure 2: Relationships between specified features

Problem 2: K-Means Clustering on Boston Housing Dataset

1. Introduction

The Boston Housing dataset, sourced from the UCI Machine Learning Repository, contains information on various attributes of houses in the Boston area, such as per capita crime rate, average number of rooms, tax rates, and more. This report aims to apply the K-Means clustering algorithm to explore the underlying structure within the dataset, focusing on continuous features. The objective is to identify natural groupings of houses and evaluate their relationships with key variables, such as the median house price (MEDV).

2. Data Loading and Preprocessing

The dataset was loaded using `sklearn.datasets.load_boston()`. To ensure that features with different scales do not disproportionately influence the clustering results, all selected features were standardized using the `StandardScaler` from scikit-learn. This process transformed the data to have a mean of 0 and a standard deviation of 1, improving the convergence speed and quality of the K-Means algorithm.

3. K-Means Clustering Methodology

The K-Means clustering algorithm was implemented using the `sklearn.cluster.KMeans` class from scikit-learn. To determine the optimal number of clusters (`n_clusters`), the silhouette score was calculated for a range of cluster numbers from 2 to 6. The silhouette score measures how similar an object is to its own cluster compared to other clusters, with higher values indicating better-defined clusters. The `n_clusters` value yielding the highest silhouette score was selected as the optimal number of clusters.

Figure 2 displays a scatter plot matrix visualizing the relationships between specified features ('mpg', 'displacement', 'horsepower', 'weight', 'acceleration'), with each data point colored and marked to represent its cluster assignment. Cluster centroids are also shown as red stars, providing a visual representation of cluster distribution across the feature space and highlighting potential clustering-related patterns.

4. Results and Analysis

4.1 Determination of Optimal Number of Clusters

The silhouette scores for different values of `n_clusters` were computed and analyzed. The results are summarized in table 6.

Table 6: Silhouette Scores for Different Cluster Numbers

n_clusters	Silhouette Score
2	0.36012
3	0.25749
4	0.26582
5	0.28782
6	0.26248

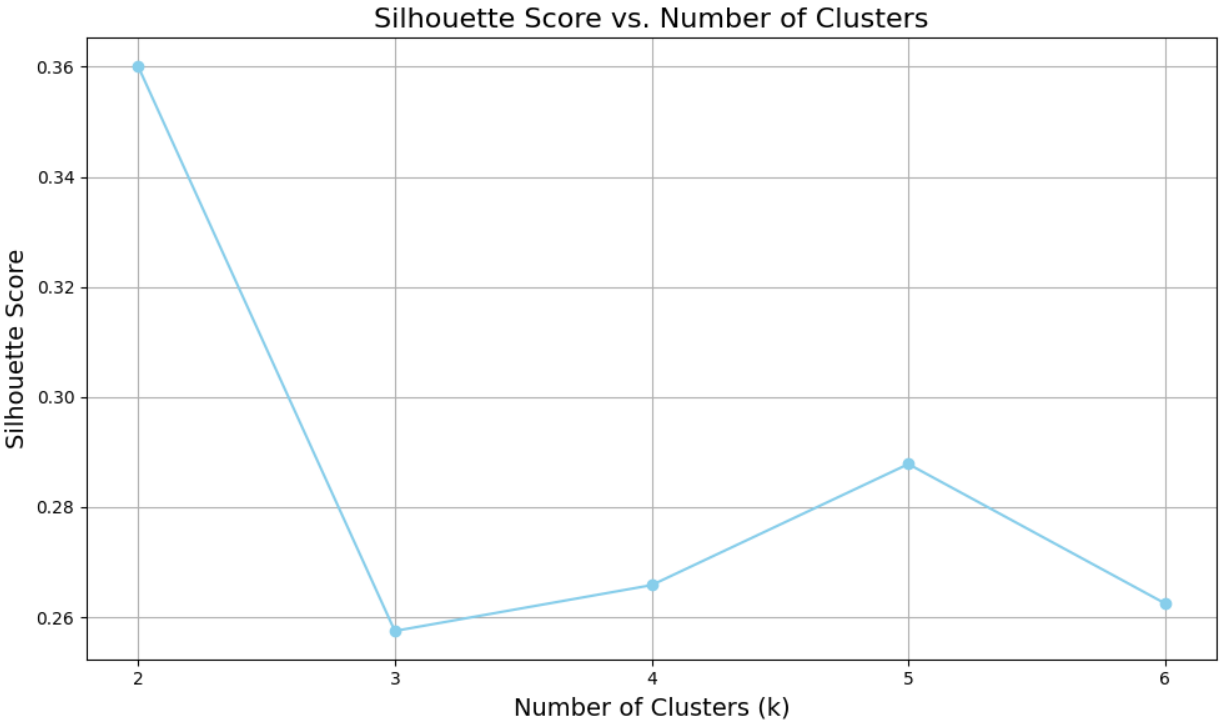

Figure 3: Relationships between specified features

4.2 Cluster Feature Means

The mean values of each feature within each cluster were calculated and are presented in the table below:

Table 7: Cluster Feature Means

Feature	Cluster 0	Cluster 1
CRIM	0.26117	9.84473
ZN	17.47720	0.00000
INDUS	6.88505	19.03972
CHAS	0.06991	0.06780
NOX	0.48701	0.68050
RM	6.45542	5.96718
AGE	56.33921	91.31808
DIS	4.75687	2.00724
RAD	4.47113	18.98870
TAX	301.91793	605.85876
PTRATIO	17.83739	19.60452
B	386.44787	301.33170
LSTAT	9.46830	18.57277

4.3 Centroid Coordinates

The centroid coordinates of each feature are shown in Table 12.

Table 8: Centroid Coordinates

Feature	Cluster 0	Cluster 1
CRIM	-0.390124	0.725146
ZN	0.262392	-0.487722
INDUS	-0.620368	1.153113
CHAS	0.002912	-0.005412
NOX	-0.584675	1.086769
RM	0.243315	-0.452263
AGE	-0.435108	0.808760
DIS	0.457222	-0.849865
RAD	-0.583801	1.085145
TAX	-0.631460	1.173731
PTRATIO	-0.285808	0.531248
B	0.326451	-0.606793
LSTAT	-0.446421	0.829787

4.4 Comparison of Cluster Feature Means and Centroid Coordinates

The following table presents the comparison between cluster feature means and centroid coordinates (to be filled with actual values).

Table 9: Difference Between Cluster Means and Centroid Coordinates

Feature	Cluster 0 Difference	Cluster 1 Difference
CRIM	5.5511e-17	-1.1102e-16
ZN	-1.1102e-16	-4.9960e-16
INDUS	-3.3307e-16	-1.1102e-15
CHAS	3.3393e-16	2.4373e-16
NOX	-1.1102e-16	-6.6613e-16
RM	5.5511e-17	2.2204e-16
AGE	-5.5511e-17	-3.3307e-16
DIS	-3.8858e-16	1.1102e-16
RAD	3.3307e-16	-1.1102e-15
TAX	-1.1102e-16	-3.9968e-15
PTRATIO	0.0000e+00	-1.4433e-15
B	2.7756e-16	1.1102e-16
LSTAT	0.0000e+00	-3.3307e-16

Problem 3: K-Means Clustering on Wine Dataset

1. Introduction

The Wine dataset, available from the UCI Machine Learning Repository and accessible via `sklearn.datasets.load_wine()`, contains chemical analysis results of wines from three different cultivars in Italy. The dataset includes 13 continuous features, such as alcohol content, malic acid, and total phenols, along with class labels indicating the cultivar type. This report applies the K-Means clustering algorithm to the scaled dataset with a fixed number of clusters (`n_clusters = 3`) to explore natural groupings of wines. Additionally, the analysis evaluates the clustering quality using Homogeneity and Completeness metrics, leveraging the actual class labels, and interprets the insights provided by these metrics.

2. Data Loading and Preprocessing

The Wine dataset was loaded into a Pandas DataFrame using the `sklearn.datasets.load_wine()` function from the scikit-learn library. The dataset comprises 178 samples, 13 continuous features, and a target variable representing the three wine cultivars (class labels: 0, 1, 2). All 13 features were selected for clustering. To ensure feature comparability, all features were standardized using `StandardScaler`, resulting in a mean of 0 and a standard deviation of 1.

3. K-Means Clustering Methodology

The K-Means clustering algorithm was applied to the standardized Wine dataset with `n_clusters = 3`, aligning with the known number of wine cultivars. The algorithm was configured with 'k-means++' initialization, a maximum of 300 iterations, and 10 initializations to ensure robust results. To evaluate the quality of the clustering with respect to the actual class labels, Homogeneity and Completeness metrics were calculated using scikit-learn's `sklearn.metrics` module.

4. Results and Analysis

4.1 Cluster Assignment and Evaluation Metrics

The K-Means algorithm assigned each of the 178 wine samples to one of three clusters. To assess the quality of the clustering with respect to the true class labels (cultivars), two evaluation metrics were calculated using scikit-learn's `sklearn.metrics` module: Homogeneity and Completeness. The results are presented below:

Table 10: Homogeneity and Completeness Scores

Metric	Score
Homogeneity	0.8788432003662366
Completeness	0.8729636016078731

4.2 Interpretation of Homogeneity

Homogeneity quantifies the degree to which each cluster is composed of data points belonging to a single, uniform class. It reflects the extent to which a clustering algorithm has successfully grouped together data points that share a common characteristic or origin. A high homogeneity score signifies that the clustering process has effectively separated distinct classes into discrete clusters, minimizing the presence of foreign or dissimilar elements within each group.

4.3 Interpretation of Completeness

Completeness evaluates the extent to which all data points belonging to a given class are assigned to the same cluster. It reflects the ability of a clustering algorithm to group together all instances that share a common label or characteristic, without scattering them across multiple clusters. A high completeness score indicates that the clustering solution has successfully consolidated all members of a specific class into a single, cohesive group.

4.4 Insights Provided by These Metrics

1. High Homogeneity (0.879): The clustering exhibits high purity, meaning clusters primarily consist of samples from a single class.

2. High Completeness (0.873): The clustering demonstrates strong cohesiveness, indicating that samples from the same class are mostly grouped within a single cluster.

3. Overall Good Clustering Performance: The combination of high homogeneity and completeness suggests that the clustering solution is effective and well-balanced.

4. Slight Bias Towards Homogeneity: A slightly higher homogeneity score compared to completeness hints at a potential preference for creating pure clusters, possibly at the expense of minor class fragmentation. However, the scores are very close, suggesting this effect is minimal.