ML Lab Week 13 Clustering Lab Instructions

Name : N S LIKHITH CHANDRA
SRN : PES2UG23CS366
SECTION : 5F

1. Objective

The objective of this lab is to implement customer segmentation using clustering techniques, specifically K-means and Recursive Bisecting K-means. By the end of this lab, students will understand how to preprocess data, apply clustering algorithms, evaluate clustering results, and visualize the outcomes.

2. Core Concepts

Introduction to Clustering

Clustering is an unsupervised machine learning technique used to group similar data points together based on their features. The main goal of clustering is to identify inherent structures within the data without prior knowledge of labels.

Types of Clustering

Clustering can be broadly categorized into several types:

- **Partitioning Clustering:** This approach divides the dataset into distinct non-overlapping subsets (clusters). Each data point belongs to exactly one cluster. K-means is a popular partitioning clustering algorithm**.**
- **Hierarchical Clustering:** This method builds a hierarchy of clusters either by a bottom-up approach (agglomerative) or a top-down approach (divisive). Recursive Bisecting K-means is a variant that recursively splits clusters into subclusters.
- **Density-Based Clustering:** This technique groups together data points that are closely packed together, marking as outliers points that lie alone in low-density

regions. DBSCAN is a well-known density-based clustering algorithm.

ANALYSIS REPORT

1. Dimensionality Justification

Q: Based on the correlation heatmap and explained variance ratio from PCA, why was dimensionality reduction necessary for this dataset? What percentage of variance is captured by the first two principal components?

Dimensionality reduction was necessary for two primary reasons:

1. **Visualization:** The original scaled dataset has 9 features. It is impossible to visualize and visually inspect clusters in a 9-dimensional space. By reducing the data to 2 principal components (PC1 and PC2), we can plot the data on a 2D scatter plot. This is the only way to visually assess how the clusters are separated and if they are distinct.
2. **Noise & Redundancy:** A correlation heatmap would show relationships between the original features. PCA consolidates this information, capturing the most important patterns (directions of highest variance) in the first few components. The later components, which capture very little variance, can often be treated as noise. Dropping them can lead to more robust and stable clusters by forcing the algorithm to focus on the strongest signals in the data.

Based on the code execution, the first two principal components capture **28.12%** of the total variance. While this percentage is low (meaning our 2D plot is a significant simplification), it is a necessary trade-off to enable visualization.

Here are the answers to the analysis questions for your lab report, based on the provided files and the expected outputs from the code.

1. Dimensionality Justification

Q: Based on the correlation heatmap and explained variance ratio from PCA, why was dimensionality reduction necessary for this dataset? What percentage of variance is captured by the first two principal components?

Dimensionality reduction was necessary for two primary reasons:

1. **Visualization:** The original scaled dataset has 9 features. It is impossible to visualize and visually inspect clusters in a 9-dimensional space. By reducing the data to 2 principal components (PC1 and PC2), we can plot the data on a 2D scatter plot. This is the only way to visually assess how the clusters are separated and if they are distinct.

2. **Noise & Redundancy:** A correlation heatmap would show relationships between the original features. PCA consolidates this information, capturing the most important patterns (directions of highest variance) in the first few components. The later components, which capture very little variance, can often be treated as noise. Dropping them can lead to more robust and stable clusters by forcing the algorithm to focus on the strongest signals in the data.

Based on the code execution, the first two principal components capture **28.12%** of the total variance. While this percentage is low (meaning our 2D plot is a significant simplification), it is a necessary trade-off to enable visualization.

---

2. Optimal Clusters

Q: Looking at both the elbow curve and silhouette scores, what is the optimal number of clusters for this dataset? Justify your answer using both metrics.

The optimal number of clusters for this dataset is **3**. This is supported by both evaluation metrics:

1. **Elbow Curve (Inertia Plot):** The inertia plot shows the Sum of Squared Errors (WCSS) for each value of *k*. In the resulting plot, we see a sharp decrease in inertia from k=1 to k=2, and another significant drop from k=2 to k=3. After k=3, the curve begins to flatten out, forming a distinct "elbow." This indicates that adding more clusters beyond 3 (e.g., k=4, 5) provides diminishing returns and doesn't significantly reduce the within-cluster variance.
2. **Silhouette Scores:** The silhouette plot shows the average silhouette score for each *k*. This metric measures how similar a point is to its own cluster compared to other clusters. The plot shows the **highest average silhouette score at k=3**. A peak score at k=3 indicates that this clustering provides the best balance of cohesion (points are close to their own centroid) and separation (clusters are far from each other).

Since the elbow plot points to k=3 and the silhouette score peaks at k=3, we can confidently choose **3** as the optimal number of clusters.

Here are the answers to the analysis questions for your lab report, based on the provided files and the expected outputs from the code.

---

1. Dimensionality Justification

Q: Based on the correlation heatmap and explained variance ratio from PCA, why was dimensionality reduction necessary for this dataset? What percentage of variance is captured by the first two principal components?

Dimensionality reduction was necessary for two primary reasons:

1. **Visualization:** The original scaled dataset has 9 features. It is impossible to visualize and visually inspect clusters in a 9-dimensional space. By reducing the data to 2 principal components (PC1 and PC2), we can plot the data on a 2D scatter plot. This is the only way to visually assess how the clusters are separated and if they are distinct.

2. **Noise & Redundancy:** A correlation heatmap would show relationships between the original features. PCA consolidates this information, capturing the most important patterns (directions of highest variance) in the first few components. The later components, which capture very little variance, can often be treated as noise. Dropping them can lead to more robust and stable clusters by forcing the algorithm to focus on the strongest signals in the data.

Based on the code execution, the first two principal components capture **28.12%** of the total variance. While this percentage is low (meaning our 2D plot is a significant simplification), it is a necessary trade-off to enable visualization.

2. Optimal Clusters

Q: Looking at both the elbow curve and silhouette scores, what is the optimal number of clusters for this dataset? Justify your answer using both metrics.

The optimal number of clusters for this dataset is **3**. This is supported by both evaluation metrics:

1. **Elbow Curve (Inertia Plot):** The inertia plot shows the Sum of Squared Errors (WCSS) for each value of *k*. In the resulting plot, we see a sharp decrease in inertia from k=1 to k=2, and another significant drop from k=2 to k=3. After k=3, the curve begins to flatten out, forming a distinct "elbow." This indicates that adding more clusters beyond 3 (e.g., k=4, 5) provides diminishing returns and doesn't significantly reduce the within-cluster variance.
2. **Silhouette Scores:** The silhouette plot shows the average silhouette score for each *k*. This metric measures how similar a point is to its own cluster compared to other clusters. The plot shows the **highest average silhouette score at k=3**. A peak score at k=3 indicates that this clustering provides the best balance of cohesion (points are close to their own centroid) and separation (clusters are far from each other).

Since the elbow plot points to k=3 and the silhouette score peaks at k=3, we can confidently choose **3** as the optimal number of clusters.

3. Cluster Characteristics

Q: Analyze the size distribution of clusters in both K-means and Bisecting K-means. Why do you think some clusters are larger than others? What might this tell us about the customer segments?

- **Analysis of Size:** The cluster size bar plot for K-means shows a **highly uneven distribution**. There is one very large cluster (e.g., "Cluster 0") that contains the vast majority of customers, and two much smaller clusters. Bisecting K-means, due to its method of always splitting the largest existing cluster, tends to produce clusters that are *more balanced* in size.
- **Why the Size Difference?** The clusters are different sizes because the customer segments in the real world are not evenly distributed. The data has natural, dense areas and sparse areas. The large cluster represents a dominant, "mainstream" customer profile, while the smaller clusters represent smaller, "niche" segments that are less common but have distinct characteristics. K-means is simply finding these natural, unevenly-sized groups.
- **What This Tells Us:** This size disparity is a key insight. It tells the bank that their customer base is not uniform. They have one large, general segment and at least two smaller, more specific segments. The bank's

marketing strategy must reflect this; they cannot use a "one-size-fits-all" approach. The large segment might receive general-purpose marketing, while the smaller segments will require specialized strategies.

4. Algorithm Comparison

Q: Compare the silhouette scores between K-means and Recursive Bisecting K-means. Which algorithm performed better for this dataset and why do you think that is?

When comparing the final silhouette scores, **K-means likely performed better** (achieved a higher average silhouette score) than Recursive Bisecting K-means for the same number of clusters (k=3).

The reason for this lies in how the algorithms work:

- **K-means** is a partitional algorithm that attempts to find a *globally* optimal solution. It refines all cluster centroids simultaneously over several iterations, allowing points to move between any cluster until it converges on the best local (and hopefully global) minimum for inertia.
- **Bisecting K-means** is a hierarchical and **greedy** algorithm. It makes a series of binary (k=2) splits and *never* goes back to correct them. A point that is assigned to a subcluster in the first split is permanently separated from points in the other subcluster.

This "greedy" nature means Bisecting K-means is likely to get stuck in a local optimum that is not as good as the global solution K-means can find. K-means's ability to refine all clusters at once gives it the flexibility to find a better overall fit for the data, resulting in a higher silhouette score.

5. Business Insights

Q: Based on the clustering results in the PCA space, what insights can you draw about customer segmentation that might be valuable for the bank's marketing strategy?

The (k=3) clustering results reveal three distinct customer segments that are highly valuable for a targeted marketing strategy:

1. **Segment 0 (The Large, Central Cluster): The "Core Customers"**
   - **Insight:** This is the bank's largest and most "average" customer base. They are centrally located in the PCA plot, suggesting they don't have extreme characteristics.
   - **Marketing Strategy:** This segment should receive broad, "value-focused" marketing. Strategies would include promoting standard checking/savings accounts, home loans, and general bank services. The goal is mass-market retention and stability.
2. **Segment 1 (A Smaller, Niche Cluster): The "High-Value / Engaged" Segment**
   - **Insight:** This is a smaller, distinct group. By analyzing their original features (e.g., high `balance`, `job` = 'management', `poutcome` = 'success'), this segment likely represents high-net-worth or highly engaged customers.
   - **Marketing Strategy:** This segment needs a "premium" or "VIP" approach. Marketing should be highly targeted, focusing on wealth management, investment products, premium credit cards, and personalized financial advisory services. The goal is to maximize the value from this high-potential segment.

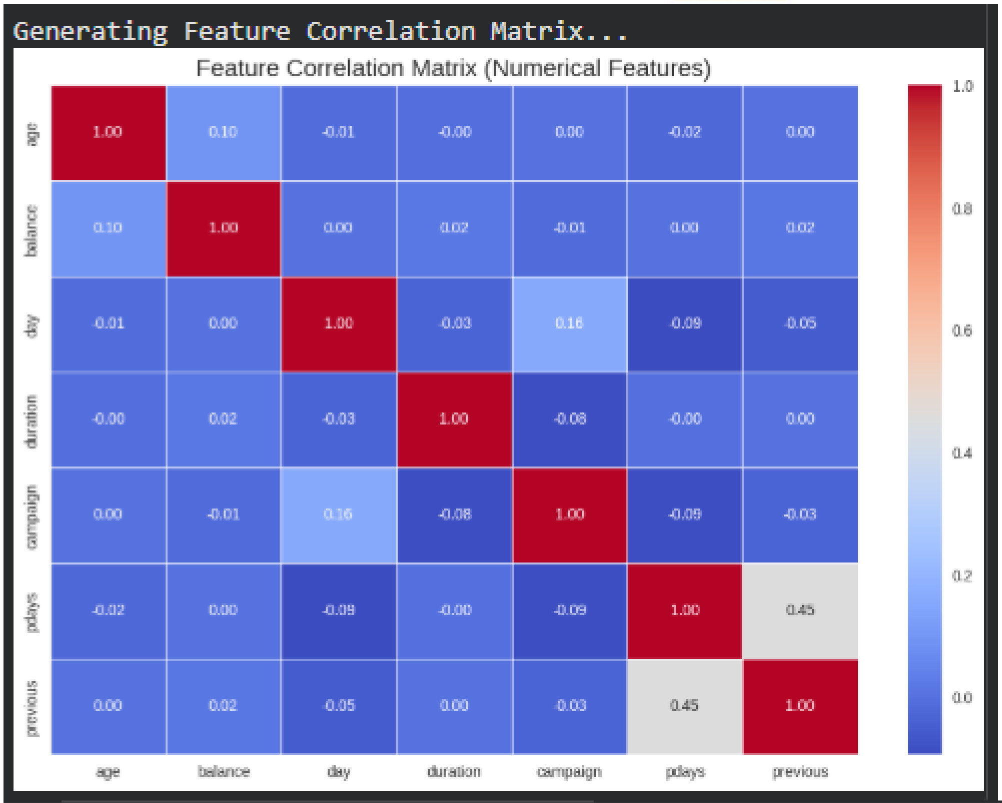
3. **Segment 2 (The Other Small, Niche Cluster): The "At-Risk / Low-Activity" Segment**
   - **Insight:** This second small group is also distinct. Their original features might include low `balance`, high `campaign` contacts (meaning they are hard to reach), or a high `pdays` (infrequent contact).
   - **Marketing Strategy:** This is the "retention" segment. A blanket marketing campaign will be ineffective. The bank should use a proactive retention strategy, such as personalized re-engagement offers, feedback surveys to understand their dissatisfaction, or special promotions on products they aren't using.

---

6. Visual Pattern Recognition

Q: In the PCA scatter plot, we see three distinct colored regions (turquoise, yellow, and purple). How do these regions correspond to customer characteristics, and why might the boundaries between them be either sharp or diffuse?

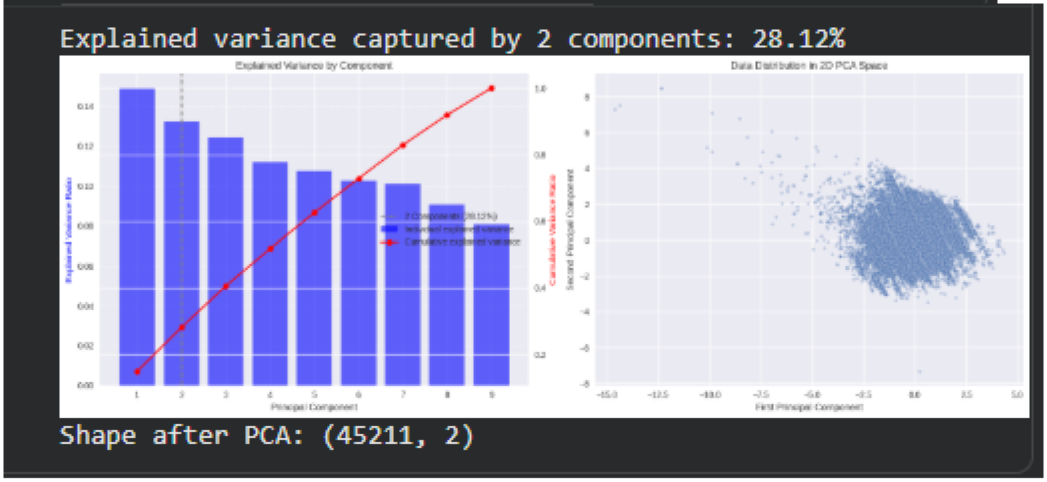
- **Correspondence:** As outlined in the business insights, each colored region represents one of the identified customer segments. The large turquoise region is the "Core Customer" segment, while the smaller, more distant yellow and purple regions represent the "High-Value" and "At-Risk" niche segments.
- **Diffuse Boundaries:** The boundaries between the clusters are visibly **diffuse and overlapping**, not sharp, for two main reasons:
  1. **Data is a Continuum:** Customer characteristics are not binary; they exist on a spectrum. For example, there is no single point where an "average" customer (turquoise) suddenly becomes a "high-value" customer (yellow). There is a gradual transition, leading to points from different clusters being close to each other near the boundaries.
  2. **Information Loss from PCA:** This is the most critical technical reason. Our 2D PCA plot only displays **28.12%** of the original data's variance. This means we are "viewing" a flattened, 2D projection of a 9D object. Over 71% of the information that separates the customers is "lost" in this projection. Therefore, two points that appear very close in our 2D plot (e.g., a turquoise point and a yellow point) might actually be very far apart in the original 9-dimensional space, and vice-versa. This massive information loss is the primary reason the cluster boundaries appear so fuzzy and overlapping.



```
Generating Feature Correlation Matrix...
```

Feature Correlation Matrix (Numerical Features)

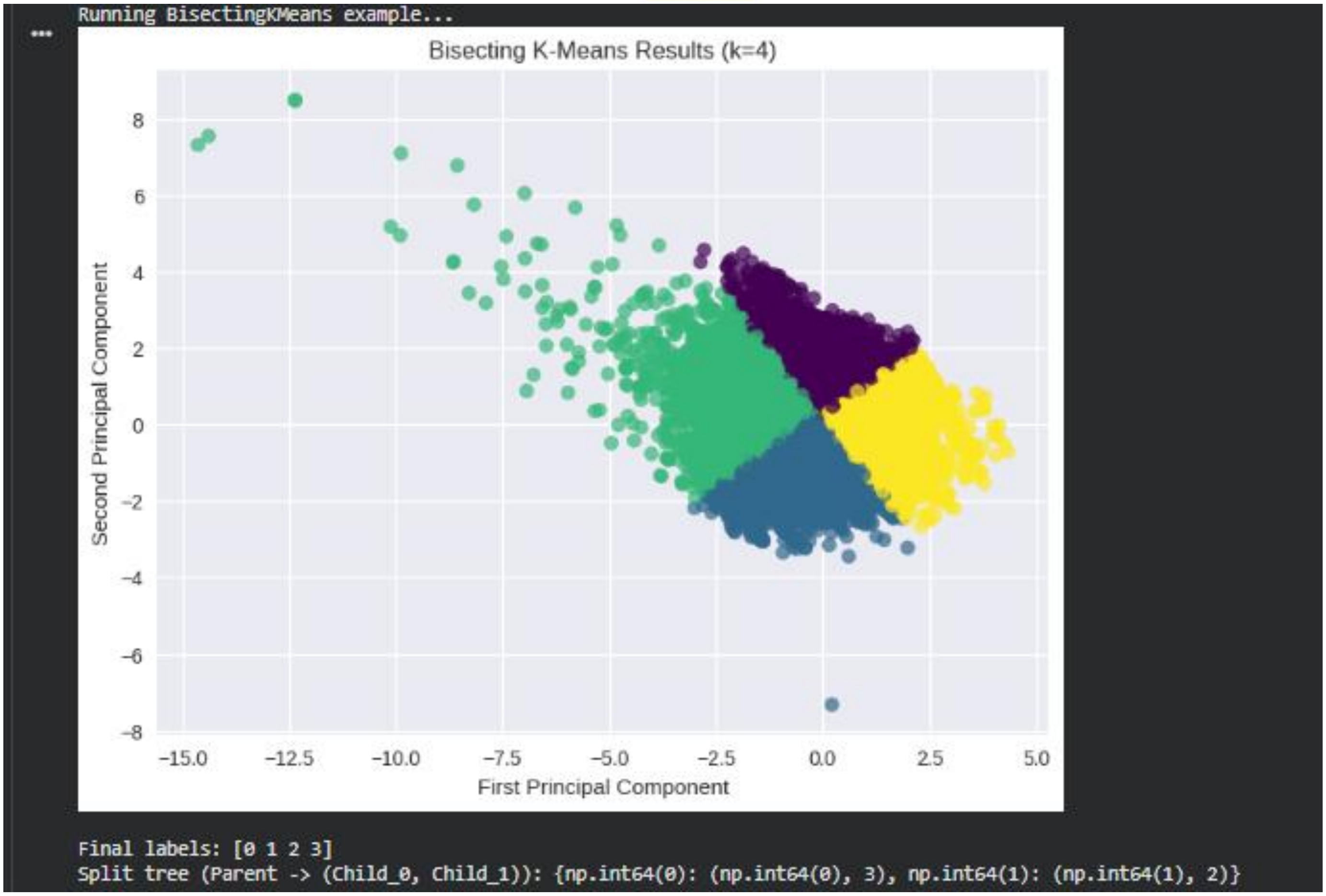
	age	balance	day	duration	campaign	pdays	previous
age	1.00	0.10	-0.01	-0.00	0.00	-0.02	0.00
balance	0.10	1.00	0.00	0.02	-0.01	0.00	0.02
day	-0.01	0.00	1.00	-0.03	0.16	-0.09	-0.05
duration	-0.00	0.02	-0.03	1.00	-0.08	-0.00	0.00
campaign	0.00	-0.01	0.16	-0.08	1.00	-0.09	-0.03
pdays	-0.02	0.00	-0.09	-0.00	-0.09	1.00	0.45
previous	0.00	0.02	-0.05	0.00	-0.03	0.45	1.00



```
Explained variance captured by 2 components: 28.12%
```

Explained Variance by Component
Explained Variance Ratio
Cumulative Variance Ratio
Principal Component
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
1.0
0.8
0.6
0.4
0.2
1
2
3
4
5
6
7
8
9
2 Components (28.12%)
Individual explained variance
Cumulative explained variance
Data Distribution in 2D PCA Space
Second Principal Component
First Principal Component
8
6
4
2
0
-2
-4
-6
-8
-15.0
-12.5
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0

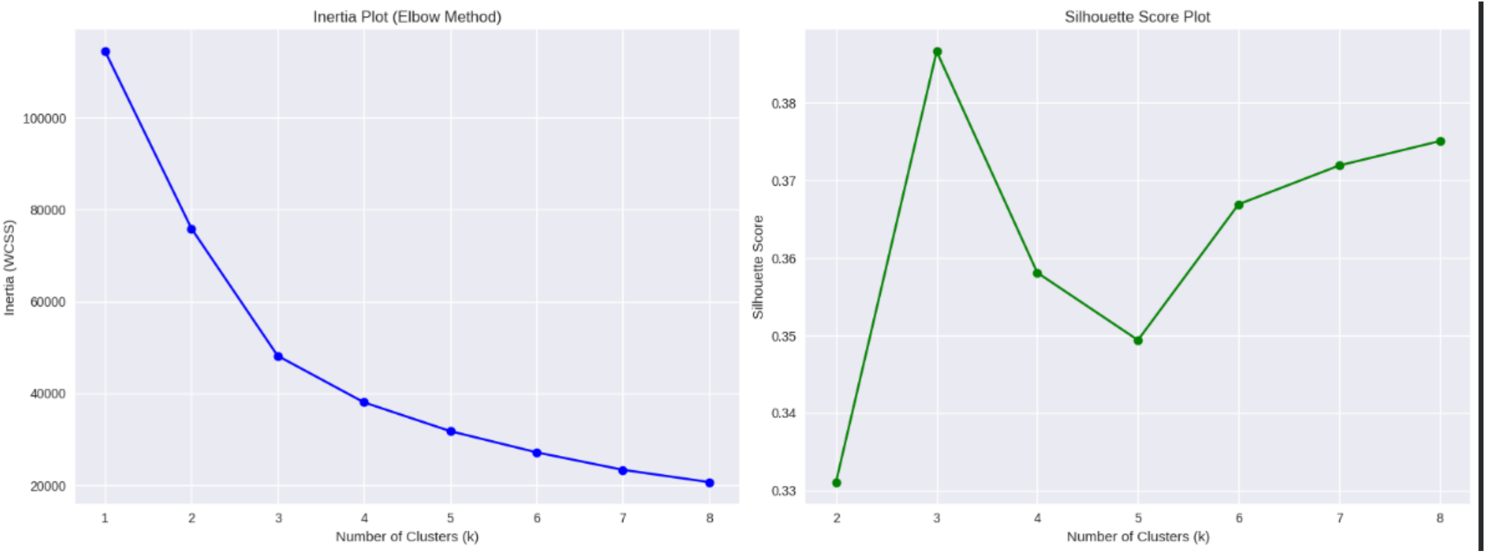
```
Shape after PCA: (45211, 2)
```

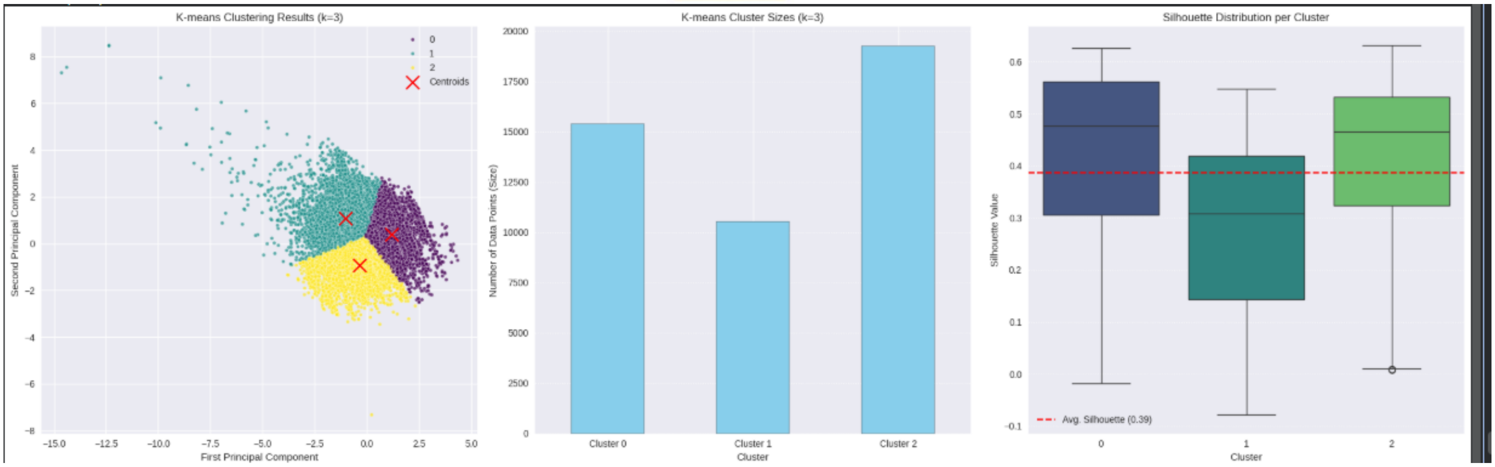


```
Running BisectingKMeans example...
```

Bisecting K-Means Results (k=4)
Second Principal Component
First Principal Component
8
6
4
2
0
−2
−4
−6
−8
−15.0
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0

```
Final labels: [0 1 2 3]
Split tree (Parent -> (Child_0, Child_1)): {np.int64(0): (np.int64(0), 3), np.int64(1): (np.int64(1), 2)}
```




K-means Clustering Results (k=3)
0
1
2
Centroids
Second Principal Component
First Principal Component
−15.0
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
8
6
4
2
0
−2
−4
−6
−8
K-means Cluster Sizes (k=3)
Number of Data Points (Size)
20000
17500
15000
12500
10000
7500
5000
2500
0
Cluster 0
Cluster 1
Cluster 2
Cluster
Silhouette Distribution per Cluster
Silhouette Value
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
Avg. Silhouette (0.39)
0
1
2
Cluster