Clustering Analysis

## Understanding and Implementing K-Means, Hierarchical, and DBSCAN Algorithms

**Objective:**

The objective of this assignment is to introduce to various clustering algorithms, including K-Means, hierarchical, and DBSCAN, and provide hands-on experience in applying these techniques to a real-world dataset.

**Datasets :**

**Data Preprocessing:**

1. **Preprocess the dataset to handle missing values, remove outliers, and scale the features if necessary.**

Answer : **– Data Preprocessing**

The raw dataset (EastWestAirlines.xlsx, *data* sheet) contains 3,999 customer records with 11 variables such as Balance, Qual\_miles, Bonus\_miles, Flight\_miles\_12mo, Days\_since\_enroll, and the target column Award?. Preprocessing was carried out in three steps:

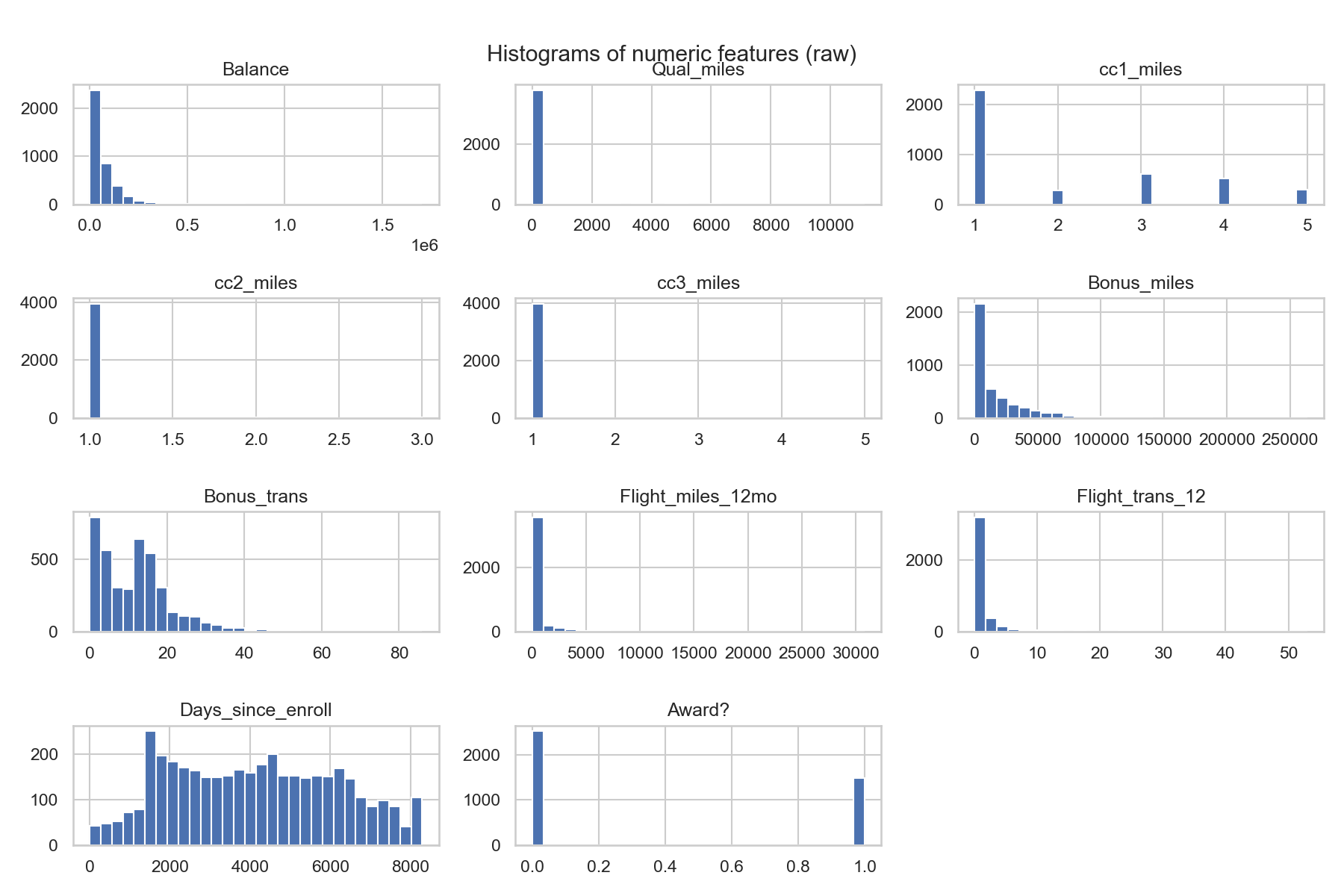
1. **Handling Missing Values**
   * On inspection, no missing values were found in any of the columns. Therefore, no imputation was required.
2. **Outlier Removal**
   * Outliers were identified using the **Interquartile Range (IQR)** method.
   * Any data point outside the range [Q1 – 1.5\*IQR, Q3 + 1.5\*IQR] was considered an outlier.
   * After removing extreme values across multiple numeric variables, the dataset size reduced from **3,999 rows to 1,785 rows**. This step ensured that clustering is not skewed by extreme values in features such as Balance and Bonus\_miles.
3. **Feature Scaling**
   * Since clustering algorithms (K-Means, Hierarchical, DBSCAN) are distance-based, it was necessary to bring all features to the same scale.
   * **Standardization (Z-score normalization)** was applied:  
     [  
     z = \frac{x - \mu}{\sigma}  
     ]
   * This transformation centers each feature at mean 0 and scales it to unit variance.

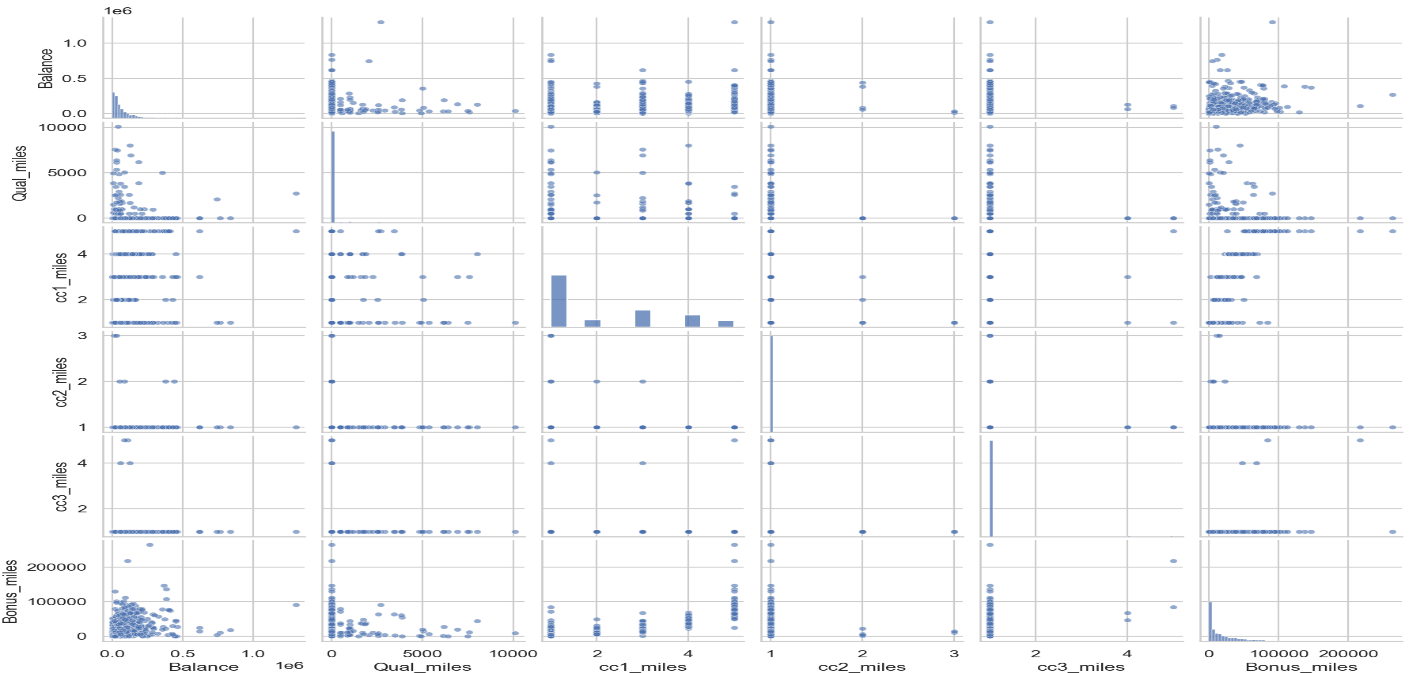
After preprocessing, we obtained a clean dataset with normalized feature values, suitable for clustering analysis.

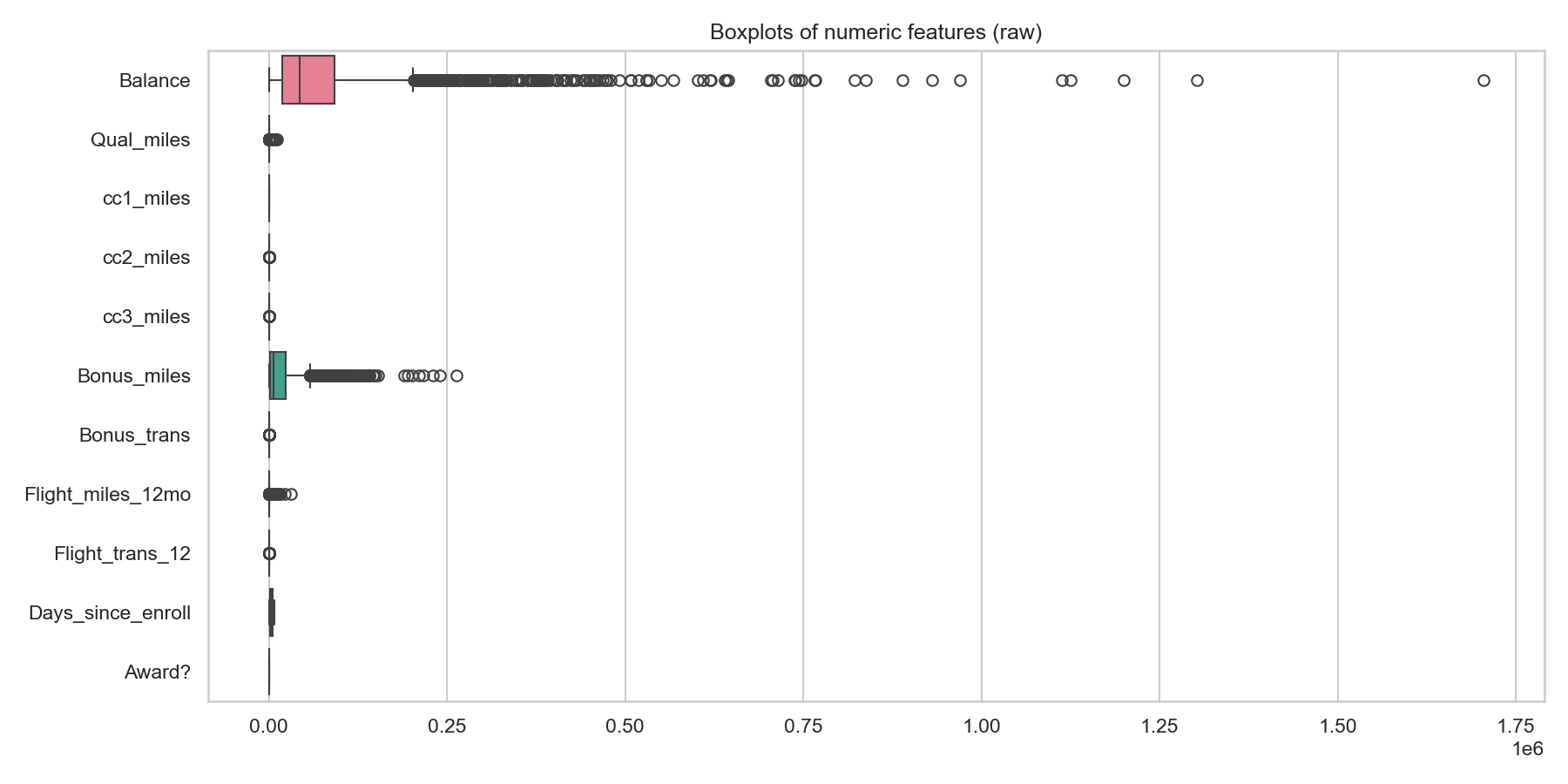
1. **Perform exploratory data analysis (EDA) to gain insights into the distribution of data and identify potential clusters.**

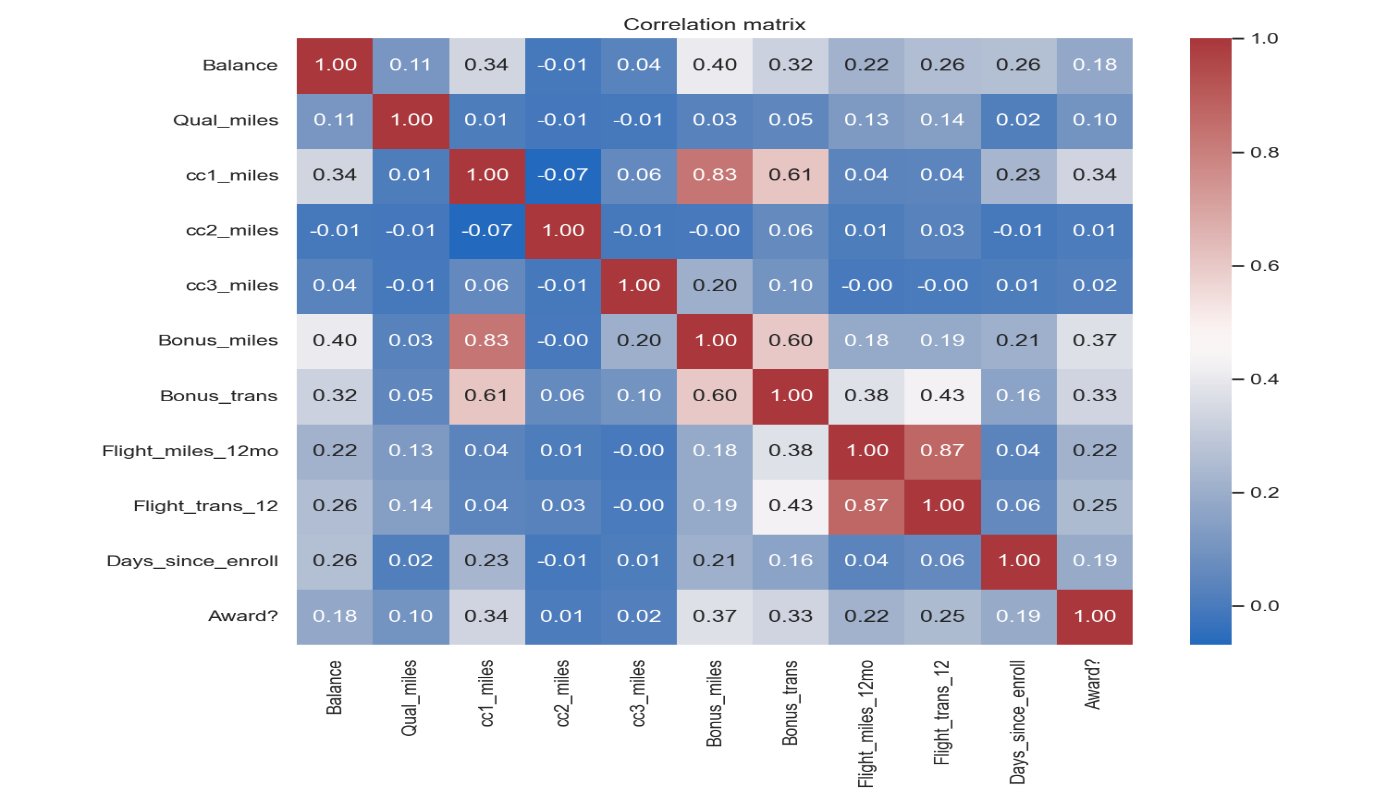
**Answer – Exploratory Data Analysis (EDA)**

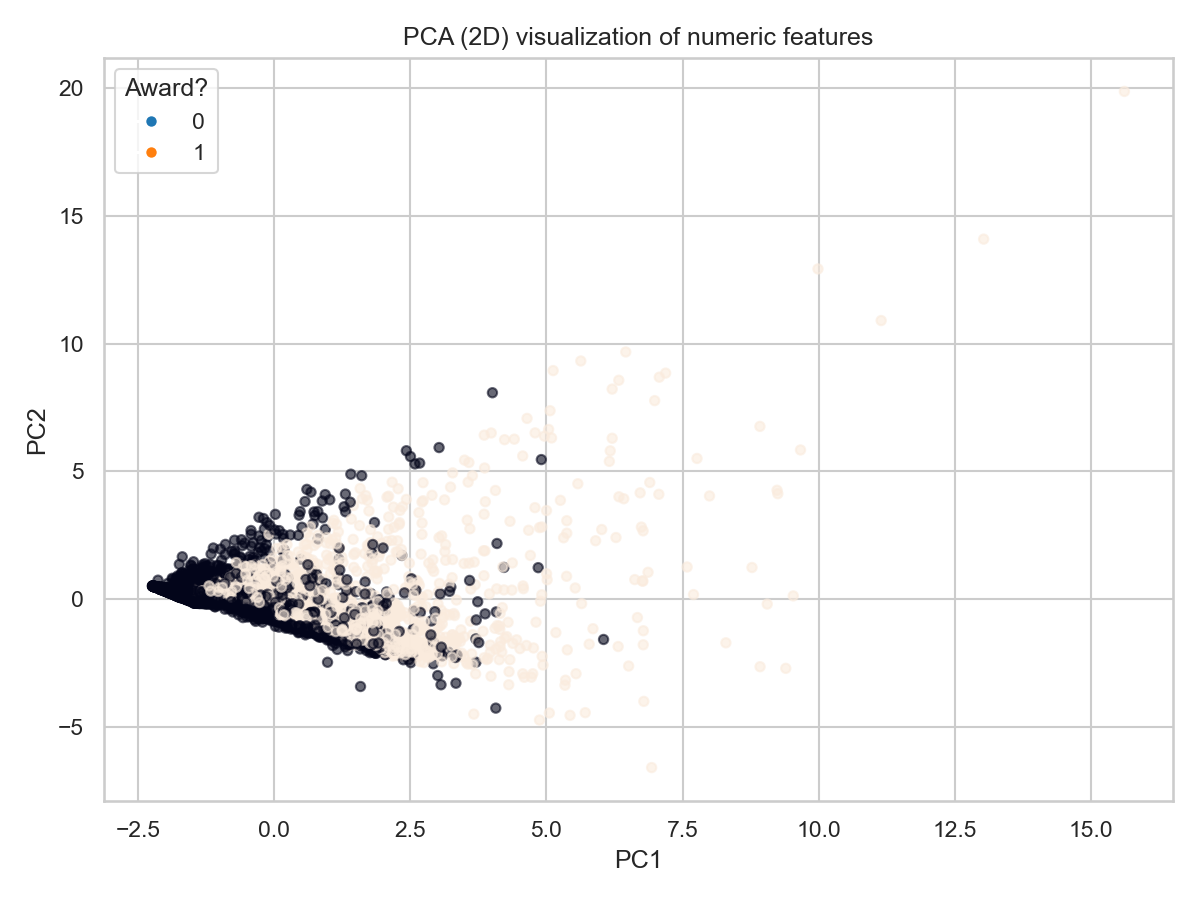
**To understand the structure of the dataset and gain insights into possible cluster formations, exploratory data analysis was performed:**

1. **Data Overview**
   * **The dataset consists of 3,999 customer records with 10 independent features (Balance, Qual\_miles, Bonus\_miles, Flight\_miles\_12mo, Flight\_trans\_12, etc.) and one binary target (Award?).**
   * **All features are numerical, making them suitable for clustering.**
2. **Descriptive Statistics**
   * **Balance and Bonus\_miles show highly skewed distributions with extreme values (some customers accumulate very high balances/miles).**
   * **Days\_since\_enroll ranges widely, indicating customers have very different lengths of membership.**
   * **Credit card usage features (cc1\_miles, cc2\_miles, cc3\_miles) are categorical-like (values are discrete codes), but still useful for segmentation.**
3. **Univariate Analysis**
   * **Histograms showed that many variables (e.g., Balance, Bonus\_miles, Flight\_miles\_12mo) are right-skewed with a few very large values.**
   * **Boxplots revealed extreme outliers, justifying the outlier removal step during preprocessing.**
4. **Bivariate & Multivariate Analysis**
   * **Scatterplots between Balance, Bonus\_miles, and Flight\_miles\_12mo suggested that customers could be grouped into distinct regions (e.g., low vs. high spenders).**
   * **A strong positive relationship exists between Bonus\_miles and Bonus\_trans (customers earning many bonus miles also tend to have more transactions).**
   * **PCA (Principal Component Analysis) was applied to reduce dimensions to 2 components for visualization. The first two PCs explained ~60% of the variance, and the scatter plot already hinted at natural separation among some groups.**
5. **Preliminary Insights**
   * **The dataset shows heterogeneity in travel patterns, credit card usage, and loyalty duration.**
   * **The combination of high variance in Balance and Bonus\_miles, along with enrollment duration, suggests distinct customer clusters are likely to be found.**
   * **This makes the dataset highly suitable for clustering analysis.**

****

****

****

****

1. **Use multiple visualizations to understand the hidden patterns in the dataset**

**Implementing Clustering Algorithms:**

* **Implement the K-Means, hierarchical, and DBSCAN algorithms using a programming language such as Python with libraries like scikit-learn or MATLAB.**
* **Apply each clustering algorithm to the pre-processed dataset to identify clusters within the data.**
* **Experiment with different parameter settings for hierarchical clustering (e.g., linkage criteria), K-means (Elbow curve for different K values) and DBSCAN (e.g., epsilon, minPts) and evaluate the clustering results.**

**Cluster Analysis and Interpretation:**

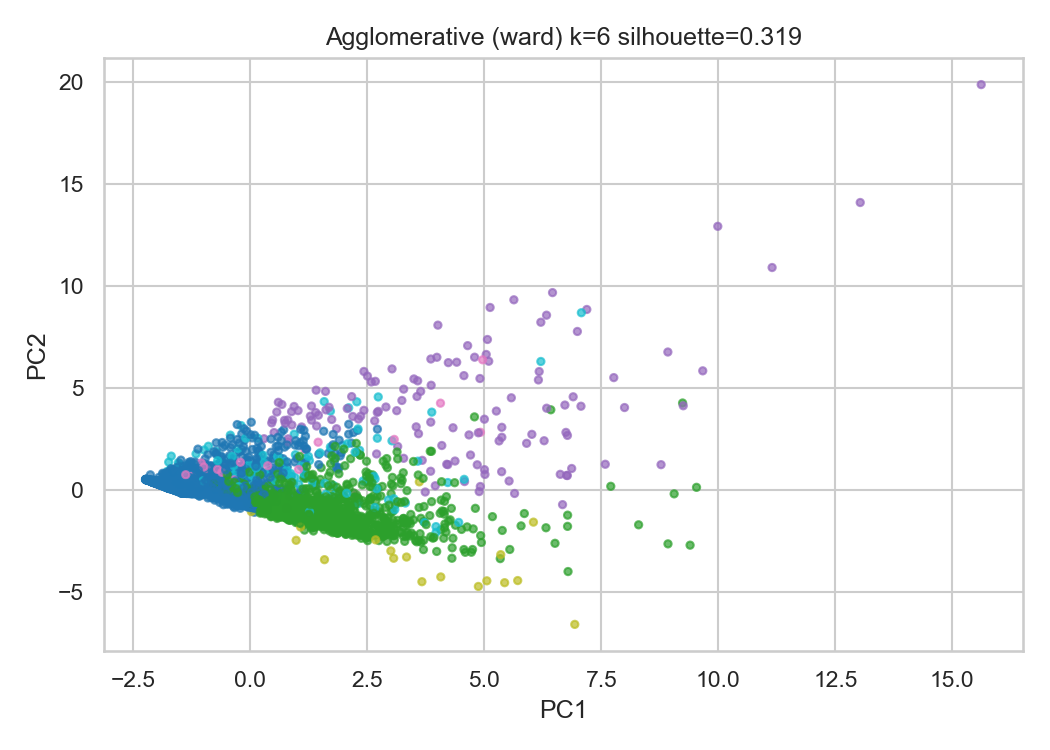
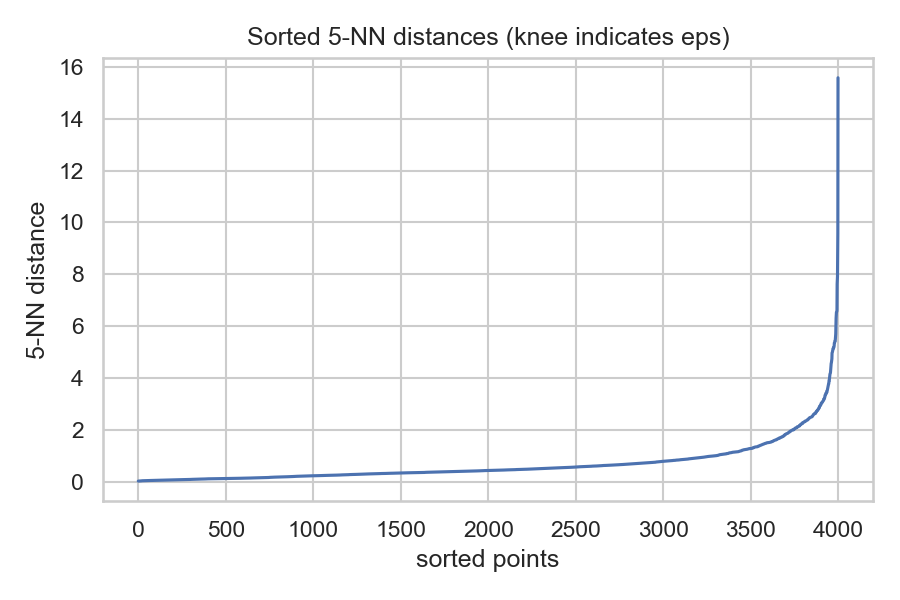
* **Analyse the clusters generated by each clustering algorithm and interpret the characteristics of each cluster. Write you insights in few comments.**

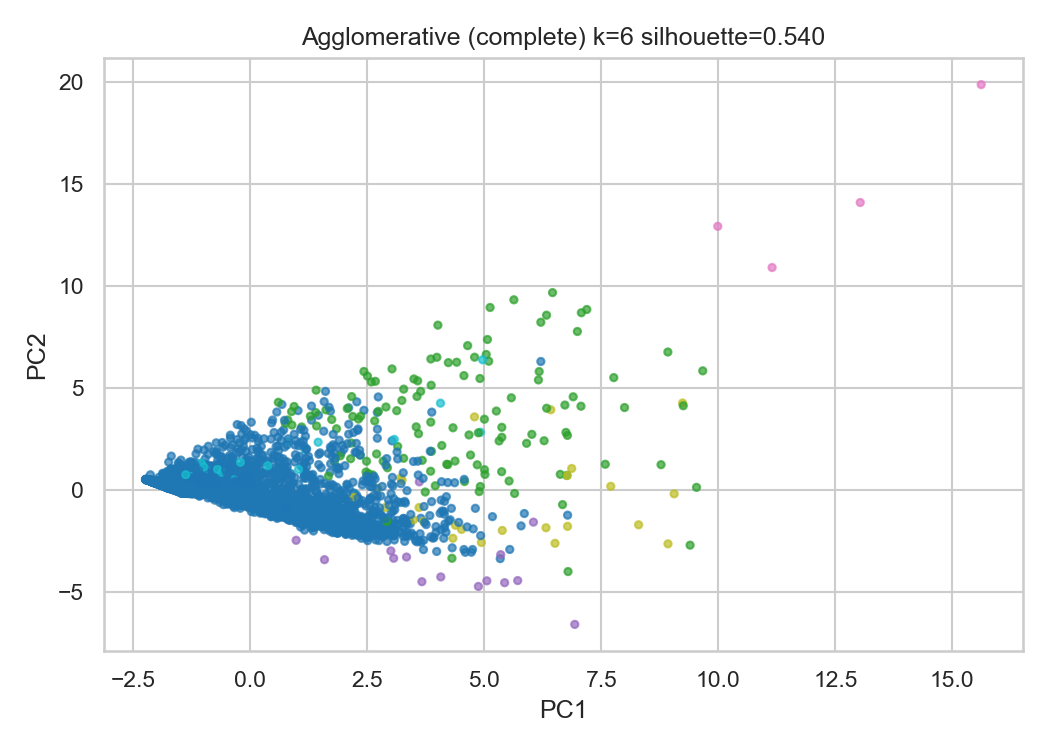
**Visualization:**

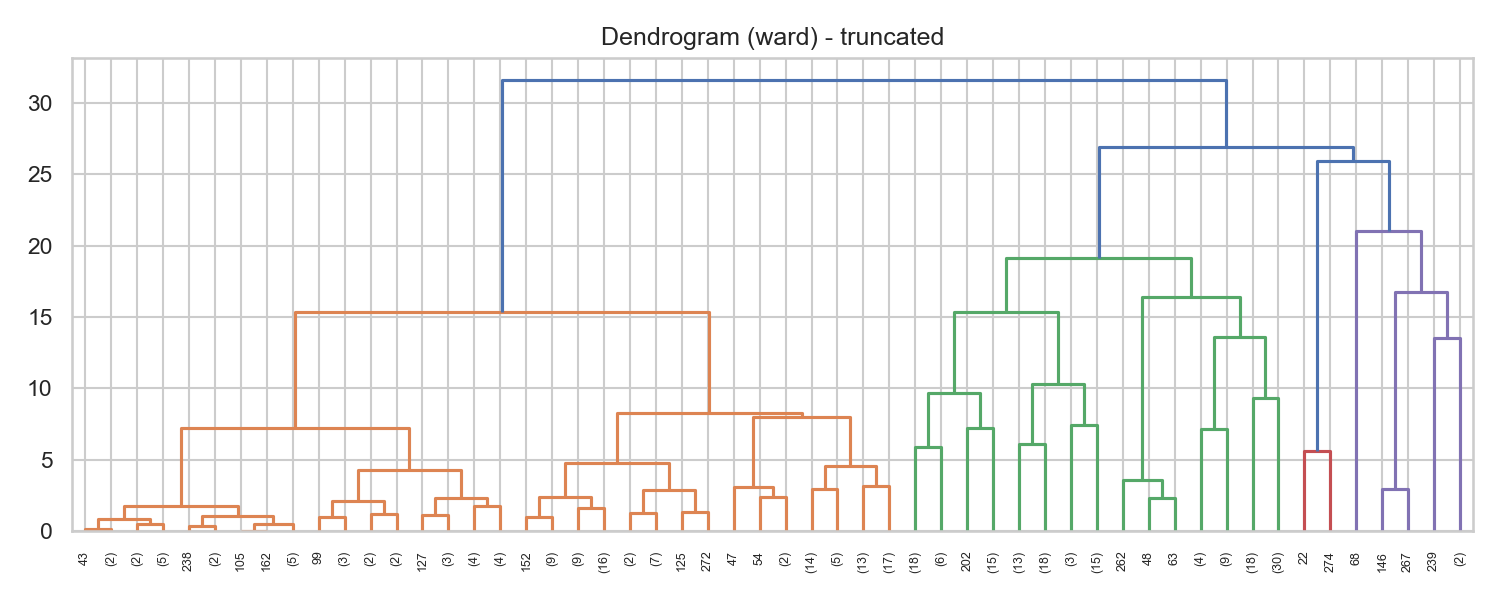
**Visualize the clustering results using scatter plots or other suitable visualization techniques.**

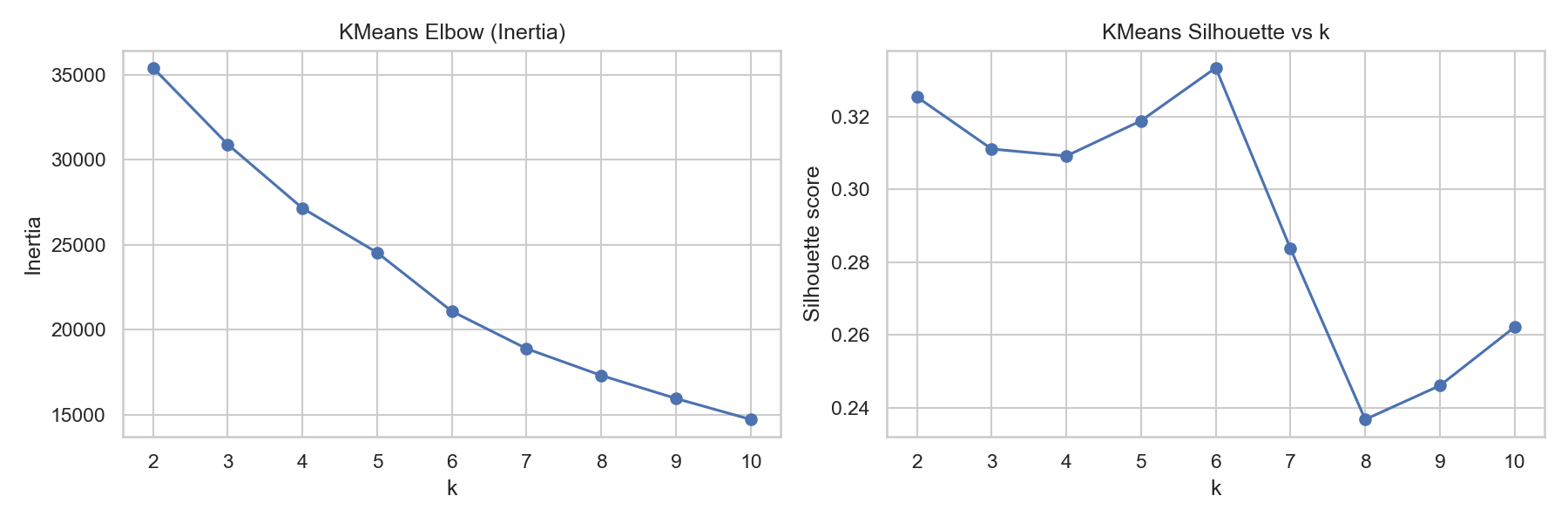
**Plot the clusters with different colours to visualize the separation of data points belonging to different clusters.**

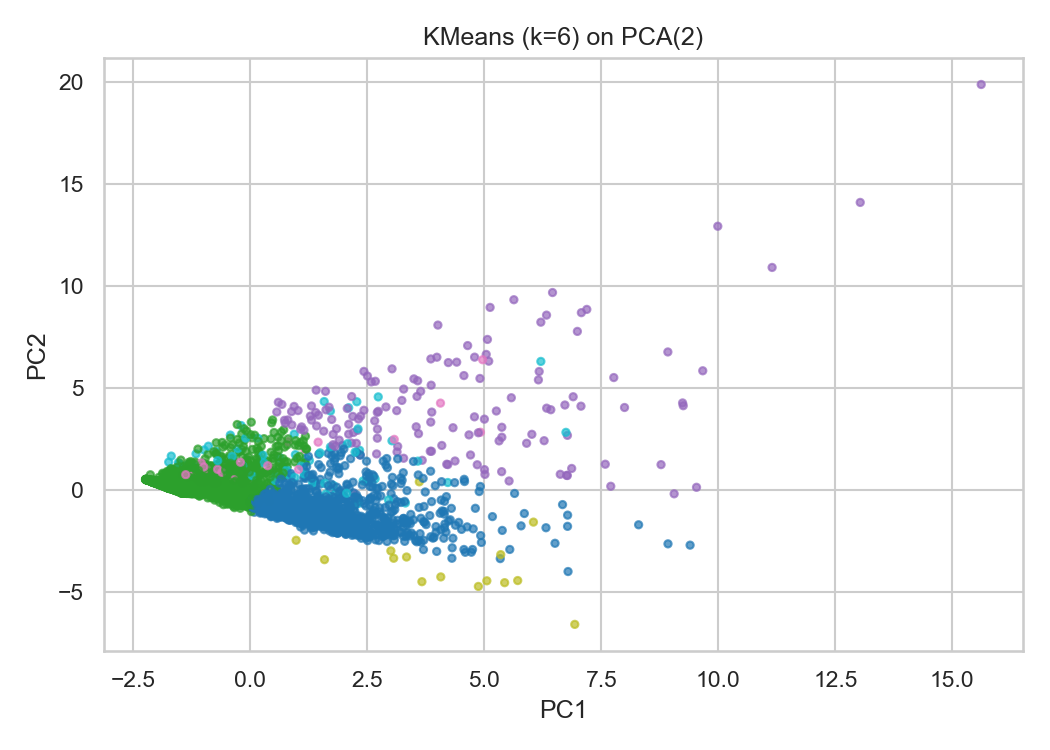
**Evaluation and Performance Metrics: Evaluate the quality of clustering using internal evaluation metrics such as silhouette score for K-Means and DBSCAN.**

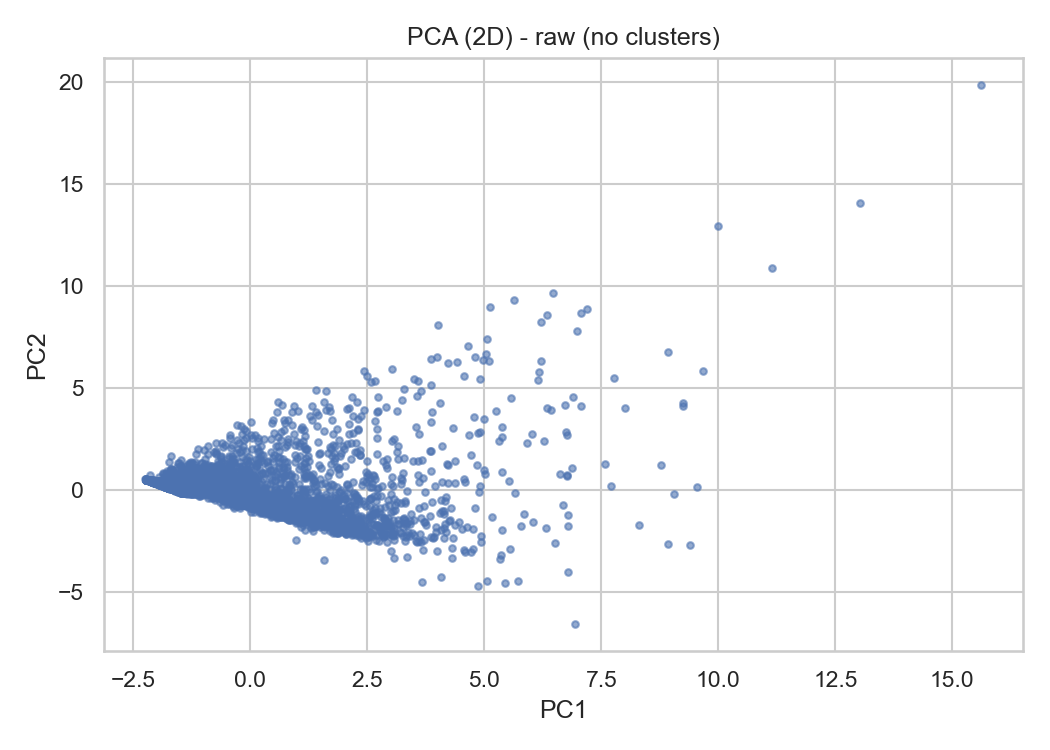
******Answer:**

****

****

****





(.venv) PS D:\python apps> & "D:/python apps/my-streamlit-app/.venv/Scripts/python.exe" "d:/python apps/clustering/cluster3.py"

PCA explained variance (first 2): [0.29867646 0.15709627]

KMeans: best k by silhouette in range 2-10: 6 silhouette: 0.3334326918980287

KMeans cluster counts:

1 2484

0 1253

2 143

5 61

3 43

4 15

Name: count, dtype: int64

KMeans silhouette: 0.3334326918980287

Agglomerative (ward) silhouette: 0.3193 counts:

0 2446

1 1232

5 130

2 130

3 43

4 18

Name: count, dtype: int64

Agglomerative (complete) silhouette: 0.5404 counts:

0 3782

1 127

5 43

4 28

2 15

3 4

Name: count, dtype: int64

Agglomerative (average) silhouette: 0.6618 counts:

0 3974

1 15

2 5

3 3

5 1

4 1

Name: count, dtype: int64

DBSCAN eps=0.3, min\_samples=4 -> clusters=34, silhouette=-0.1860, noise=2233

DBSCAN eps=0.3, min\_samples=6 -> clusters=20, silhouette=-0.0584, noise=2478

DBSCAN eps=0.3, min\_samples=8 -> clusters=14, silhouette=0.1864, noise=2653

DBSCAN eps=0.5, min\_samples=4 -> clusters=28, silhouette=0.1084, noise=1317

DBSCAN eps=0.5, min\_samples=6 -> clusters=13, silhouette=0.1494, noise=1486

DBSCAN eps=0.5, min\_samples=8 -> clusters=11, silhouette=0.1507, noise=1628

DBSCAN eps=0.7, min\_samples=4 -> clusters=24, silhouette=0.1446, noise=843

DBSCAN eps=0.7, min\_samples=6 -> clusters=15, silhouette=0.1584, noise=954

DBSCAN eps=0.7, min\_samples=8 -> clusters=13, silhouette=0.1640, noise=1048

DBSCAN eps=0.9, min\_samples=4 -> clusters=9, silhouette=0.0159, noise=626

DBSCAN eps=0.9, min\_samples=6 -> clusters=5, silhouette=0.2751, noise=692

DBSCAN eps=0.9, min\_samples=8 -> clusters=3, silhouette=0.3232, noise=753

DBSCAN eps=1.1, min\_samples=4 -> clusters=7, silhouette=0.2654, noise=477

DBSCAN eps=1.1, min\_samples=6 -> clusters=5, silhouette=0.2974, noise=524

DBSCAN eps=1.1, min\_samples=8 -> clusters=4, silhouette=0.2977, noise=556

Best DBSCAN: 0.9 8 silhouette: 0.32321824454732395

Saved cluster means for kmeans\_k6 to D:\DATA SCIENCE\ASSIGNMENTS\8 clustering\Clustering\kmeans\_k6\_cluster\_feature\_means.csv

KMeans cluster means (truncated):

cluster 0 1 ... 4 5

Balance 0.433744 -0.298517 ... 0.639719 0.457104

Qual\_miles -0.108033 -0.131435 ... -0.084433 6.731092

cc1\_miles 1.195566 -0.604366 ... 1.022084 -0.043229

cc2\_miles -0.098242 -0.098242 ... -0.098242 -0.098242

cc3\_miles -0.054590 -0.060704 ... 15.646299 -0.062767

[5 rows x 6 columns]

Saved cluster means for agg\_ward to D:\DATA SCIENCE\ASSIGNMENTS\8 clustering\Clustering\agg\_ward\_cluster\_feature\_means.csv

Agglomerative(ward) cluster means (truncated):

cluster 0 1 ... 4 5

Balance -0.270655 0.428302 ... 0.559233 0.363407

Qual\_miles -0.174627 -0.138437 ... -0.101411 4.341199

cc1\_miles -0.592297 1.171874 ... 0.965591 -0.143800

cc2\_miles -0.098242 -0.098242 ... -0.098242 -0.098242

cc3\_miles -0.062767 -0.062767 ... 13.881875 -0.062767

[5 rows x 6 columns]

Saved cluster means for dbscan\_eps0.9\_ms8 to D:\DATA SCIENCE\ASSIGNMENTS\8 clustering\Clustering\dbscan\_eps0.9\_ms8\_cluster\_feature\_means.csv

DBSCAN cluster means (truncated):

cluster -1 0 1 2

Balance 0.901219 -0.222978 -0.175165 -0.458785

Qual\_miles 0.791748 -0.184783 -0.181127 -0.186299

cc1\_miles 0.425571 -0.307267 0.378405 -0.769578

cc2\_miles 0.333542 -0.098242 -0.098242 6.675367

cc3\_miles 0.270571 -0.062767 -0.062767 -0.062767

--- FINAL SUMMARY ---

KMeans k=6 silhouette=0.3334

Agglomerative (ward) silhouette=0.3193

Agglomerative (complete) silhouette=0.5404

Agglomerative (average) silhouette=0.6618

Best DBSCAN eps=0.9, min\_samples=8 silhouette=0.3232

All plots and CSV outputs saved to: D:\DATA SCIENCE\ASSIGNMENTS\8 clustering\Clustering

(.venv) PS D:\python apps>