# Machine Learning

## (CSL7620)

**PROJECT REPORT**

**Topic:** Advanced Clustering Techniques For Customer Segmentation

**Submitted By:**

Pooja Naveen Poonia (M24CSA020)

Shivani Tiwari (M24CSA029)

Suvigya Sharma (M24CSA033)

**Submitted To:** Dr. Angshuman Paul

# Objective

- To analyze a transactional dataset from a UK-based online retail store and apply unsupervised learning (clustering) techniques.
- The aim is to group customers based on purchasing behavior, potentially aiding in identifying customer segments for targeted marketing.

## Dataset Description:

- Online Retail Dataset:Online retail is a transactional data collection comprising all transactions for a UK-based and registered online retail non-store between 01/12/2010 and 09/12/2011. The business primarily offers distinctive all-occasion gifts. Many of the firm's clients are wholesalers.
- The size of this dataset is about 541910 rows and 8 columns (invoice number, Stock code, Description, Quantity, Invoice date, Unit price, Customer ID) We have performed Unsupervised learning algorithms on this dataset like K-Means, GMM, DBSCAN and Hierarchical clustering algorithms.

## Context:

- Each row represents a customer, each column contains customer's attributes described on the column Metadata.
- The data set includes information about:Customers who left within the last month – the column is called Churn.
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device -protection, tech support, and streaming TV and movies.
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents.

## Feature Description:

- InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- Description: Product (item) name. Nominal.
- Quantity: The quantities of each product (item) per transaction. Numeric. InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.
- UnitPrice: Unit price. Numeric, Product price per unit in sterling. CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

- Country: Country name. Nominal, the name of the country where each customer resides

## Methodology

- **Importing Libraries:** All necessary libraries were imported.
- **EDA**: We checked feature correlation, multicollinearity, feature distribution and other descriptive statistics of the dataset. Outlier analysis was also performed on the dataset.
- **Preprocessing**: Basic data cleaning steps are expected to handle missing values, inconsistencies, and preparation of features.
- **Feature Engineering:** We created three new features from the dataset known as the "RFM features"
    1) Recency: How recently the customer made a purchase
    2) Frequency: How often do they purchase
    3) Monetary Value: How much they spent
- **RFM Analysis answers the following questions:**
    1) Who are our best customers?
    2) Who has the potential to be converted into more profitable customers?
    3) Which customers do we need to retain?
    4) Which group of customers is most likely to respond to our marketing campaign?
- **Clustering Algorithms**:
    - **K-Means**: A popular clustering method aiming to partition data into k clusters, where each data point belongs to the cluster with the nearest mean.
    - **MiniBatch K-Means:** A variant of K-Means that uses small, random batches of data to update cluster centroids, improving computational efficiency for large datasets.
    - **Gaussian Mixture Model (GMM)**: Probabilistic model assuming data points are generated from a mixture of Gaussian distributions.
    - **DBSCAN**: Density-based clustering algorithm useful for finding clusters of arbitrary shapes and identifying outliers.
    - **Hierarchical Clustering**: Agglomerative clustering approach where clusters are iteratively merged based on distance metrics.

- **Hyperparameter Tuning of clustering algorithms**

    **Hopkins Test**: The Hopkins Test checks if the data has natural clusters. A value close to 1 means the data is well-clustered, while a value near 0 suggests the data is randomly distributed.

- **For Determining the value of k for clustering techniques following methods were used:**

  ### 1) K-means clustering:

  - **Elbow Method:** The Elbow Method helps find the best number of clusters by plotting the sum of squared distances within clusters and looking for the point where the curve bends, indicating the point of diminishing returns.
  - **Davies-Bouldin Index:** The Davies-Bouldin Index measures cluster separation and compactness, with lower values indicating better-defined, more distinct clusters.
  - **Silhouette Analysis:** The plot with the highest average silhouette score indicates the best value for k, where the clusters are most distinct and well-separated.

  ### 2) Gaussian Mixture Model:

  - **BIC and AIC Scores for GMM:** BIC (Bayesian Information Criterion) and AIC (Akaike Information Criterion) are statistical measures used to compare the quality of different models. Both criteria assess the trade-off between model fit and complexity, with lower values indicating a better model.

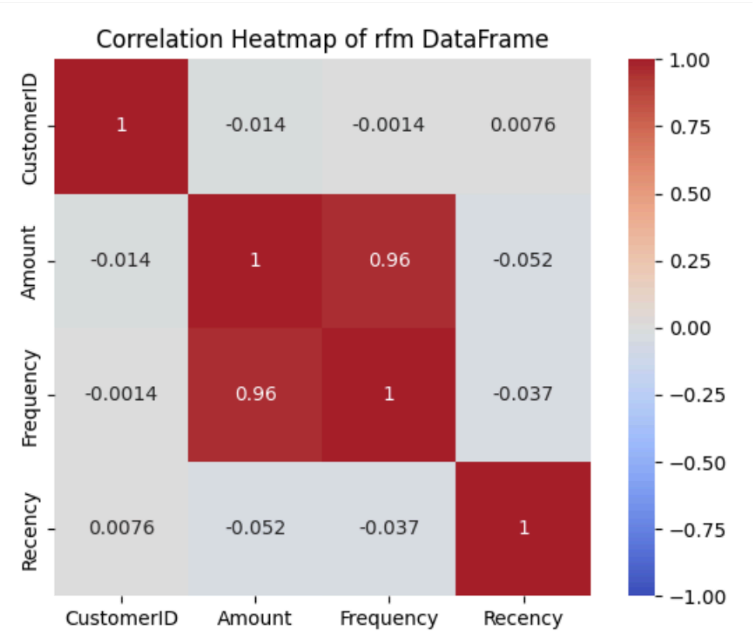
  ### 3) Hierarchical Clustering:

  **Dendrogram:** A dendrogram is a tree-like diagram used to visualize hierarchical clustering. It shows how data points or clusters are merged or split at different levels of similarity, with the height of the branches representing the distance or dissimilarity between clusters**.**

- **DBSCAN :** DBSCAN, the ε (epsilon) value was determined using the k-distance graph, where ε = 0.7 was selected based on the point of maximum distance increase
- **Integration of Streamlit for Interactive Clustering Visualization and Analysis:** Integrating Streamlit enables interactive visualization and analysis of clustering results, allowing users to explore and adjust parameters in real-time.

# Results

- **Correlation among features**



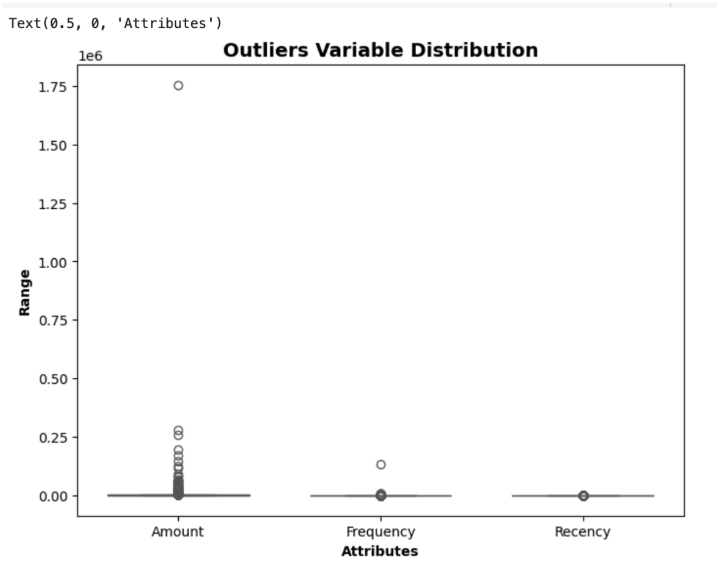
## Inference:

- Amount and Frequency are highly correlated, meaning frequent buyers tend to spend more.
- Recency is largely independent of both Amount and Frequency, suggesting that how recently a customer purchased isn't necessarily tied to how much or how often they buy.
- CustomerID serves as a unique identifier with no real correlation with other metrics, as expected.

## Multicollinearity :

```
For CustomerID and CustomerID, there is NO multicollinearity problem
For CustomerID and Amount, there is NO multicollinearity problem
For CustomerID and Frequency, there is NO multicollinearity problem
For CustomerID and Recency, there is NO multicollinearity problem
For Amount and CustomerID, there is NO multicollinearity problem
For Amount and Amount, there is NO multicollinearity problem
Multicollinearity alert between Amount - Frequency
For Amount and Recency, there is NO multicollinearity problem
For Frequency and CustomerID, there is NO multicollinearity problem
Multicollinearity alert between Frequency - Amount
For Frequency and Frequency, there is NO multicollinearity problem
For Frequency and Recency, there is NO multicollinearity problem
For Recency and CustomerID, there is NO multicollinearity problem
For Recency and Amount, there is NO multicollinearity problem
For Recency and Frequency, there is NO multicollinearity problem
For Recency and Recency, there is NO multicollinearity problem
The total number of strongly correlated features: 2
```

- ## Outlier Analysis:



```
Text(0.5, 0, 'Attributes')
```

- Amount has significant outliers, with one particularly extreme value over 1.75 million. This suggests that some customers have unusually high spending, which could be impacting the analysis.
- Frequency also has a few outliers, indicating that some customers make far more purchases than average.
- Recency has a minor outlier effect, suggesting that most customers have recent purchase dates, with only a few showing significantly older recency values.

- **Hopkins Test**

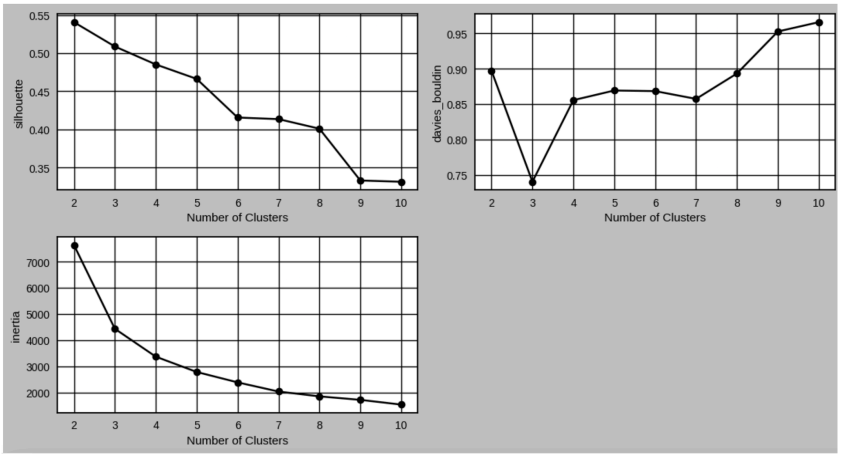
**Hopkins Score:** 0.9387048848842228

- A Hopkins Score of 0.9387 indicates that the data is highly clustered. Hopkins' statistic ranges from 0 to 1, where values close to 1 suggest that the data has a strong tendency to form clusters, and values close to 0 indicate a more uniform or random distribution.

## Clustering Techniques:

### 1. K-means Clustering

- **Determining Optimal K:** The optimal value of k according to the Silhouette Score is 2, while the optimal value according to the Davis Bouldin Index and the Elbow Method is 3



- **K-means without hyperparameter tuning (using default values):**

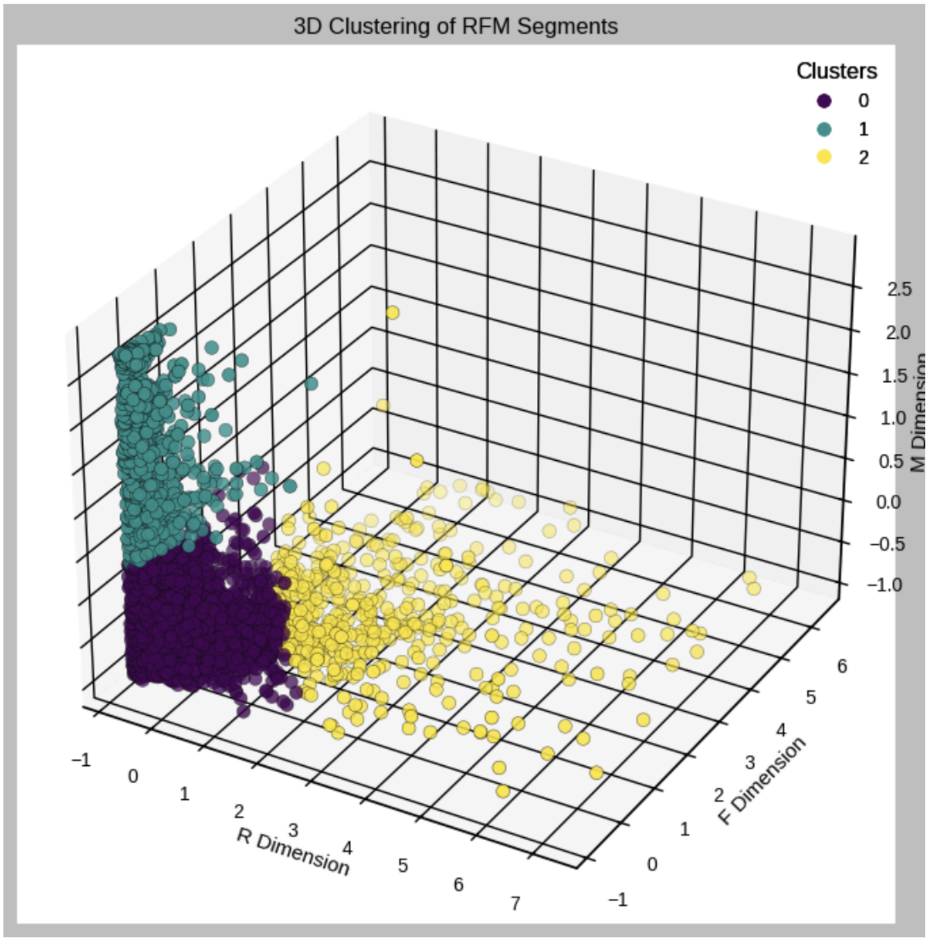
**Silhouette Score:** 0.5087447915808871
**Davies-Bouldin Index:** 0.7402170887738294

- **Silhouette Score (0.508)**: A score around 0.5 indicates that the clusters are moderately well-separated, but there is room for improvement in the clustering quality.

- **Davies-Bouldin Index (0.740)**: A lower score suggests that the clusters are relatively well-separated, but the clusters might still overlap to some extent. Lower values generally indicate better clustering performance.

Overall, the clustering is decent, but there's potential for improvement, especially with hyperparameter tuning.

## Cluster visualization



**Cluster 0 (Purple) - "New/Infrequent Shoppers"**

- These customers have recently made purchases but do so infrequently and spend relatively less. They may be new customers or occasional shoppers who need more encouragement to return and engage.

**Cluster 1 (Teal) - "Loyal Regulars"**

- These customers purchase more frequently and show a range in recency. They tend to be repeat purchasers with moderate spending levels. This group could represent loyal or consistent customers who would respond well to loyalty programs and regular engagement.

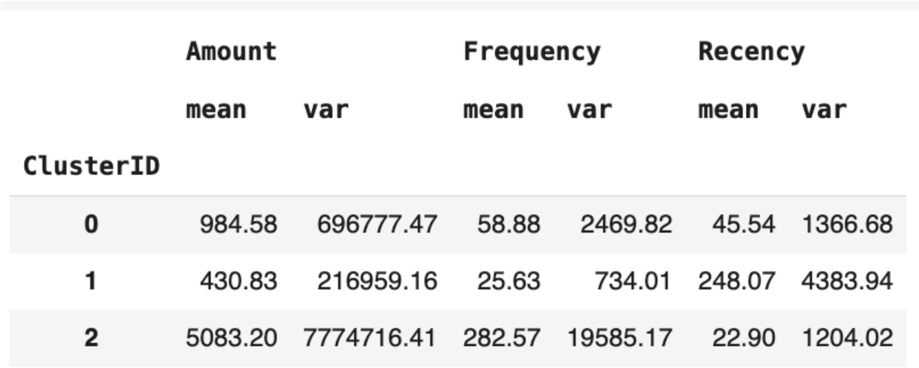
**Cluster 2 (Yellow) - "High-Value/Big Spenders"**

- This group consists of customers who make significant purchases (high monetary value), even if they don't buy as frequently. They are high-value customers who might not need frequent engagement but are prime targets for premium offers or VIP treatment.

## Cluster counts:

| **ClusterID** | Count |
|---|---|
| **0** | 2712 |
| **1** | 1052 |
| **2** | 492 |

## Cluster Mean and Variance:



| | Amount | | Frequency | | Recency | |
|---|---|---|---|---|---|---|
| | mean | var | mean | var | mean | var |
| **ClusterID** | | | | | | |
| **0** | 984.58 | 696777.47 | 58.88 | 2469.82 | 45.54 | 1366.68 |
| **1** | 430.83 | 216959.16 | 25.63 | 734.01 | 248.07 | 4383.94 |
| **2** | 5083.20 | 7774716.41 | 282.57 | 19585.17 | 22.90 | 1204.02 |

## Inference:

- **Cluster 0**: Customers in this cluster have moderate amounts and frequencies, with relatively low recency. Their spending is consistent, but they don't make purchases frequently.
- **Cluster 1**: This group has lower amounts and frequencies, but a significantly higher recency, indicating they are recent customers who might not have spent much yet.

- **Cluster 2**: Customers here show very high spending amounts and frequencies, but with a much lower recency, suggesting they were highly active in the past but may have reduced their recent activity.

## K means with Hyperparameter tuning:

- ### The initialization method

**k-means++**: The default method, which attempts to select initial centroids that are spread out, aiming for better convergence and a more stable solution.

**random**: Randomly selects initial centroids, which might lead to less stable solutions or suboptimal clustering.

| index | init | silhouette_score |
|---|---|---|
| **0** | k-means++ | 0.5087516559582256 |
| **1** | random | 0.5087516559582256 |

1. n_init'auto' or int, default='auto': Number of times the k-means algorithm is run with different centroid seeds.

| index | n_init | silhouette_score |
|---|---|---|
| **0** | 6 | 0.5086407990053111 |
| **1** | 7 | 0.5086407990053111 |
| **2** | 8 | 0.5086407990053111 |
| **3** | 9 | 0.5086407990053111 |
| **4** | 10 | 0.5086407990053111 |
| **5** | 11 | 0.5086407990053111 |
| **6** | 12 | 0.5086407990053111 |
| **7** | 13 | 0.5086407990053111 |
| **8** | 14 | 0.5086407990053111 |
| **9** | 15 | 0.5086407990053111 |

**Inference:** The Silhouette Score remained constant at 0.51 for all values of n_init, suggesting that increasing the number of initializations did not significantly improve clustering performance. This implies that the algorithm quickly converges to a good solution, regardless of how many times it is initialized.

2. # max_iterint, default=300: Maximum number of iterations of the k-means algorithm for a single run.

| Index | max_iter | silhouette_score |
|---:|---:|---:|
| **0** | 295 | 0.5086407990053111 |
| **1** | 296 | 0.5086407990053111 |
| **2** | 297 | 0.5086407990053111 |
| **3** | 298 | 0.5086407990053111 |
| **4** | 299 | 0.5086407990053111 |
| **5** | 300 | 0.5086407990053111 |
| **6** | 301 | 0.5086407990053111 |
| **7** | 302 | 0.5086407990053111 |
| **8** | 303 | 0.5086407990053111 |
| **9** | 304 | 0.5086407990053111 |
| **10** | 305 | 0.5086407990053111 |

**Inference:** The Silhouette Score remained steady at 0.51 across all values of max_iter, indicating that increasing the number of iterations did not have any notable impact on the clustering outcome.

3. algorithm{“lloyd”, “elkan”}, default=”lloyd”: K-means algorithm to use. The classical EM-style algorithm is "lloyd". The "elkan" variation can be more efficient on some datasets with well-defined clusters, by using the triangle inequality.

| ndex | algorithm | silhouette_score |
|---|---|---|
| **0** | lloyd | 0.5086407990053111 |

| | | |
|---|---|---|
| **1** | elkan | 0.5086407990053111 |
| **2** | auto | 0.5086407990053111 |
| **3** | full | 0.5086407990053111 |

**Inference:** All four algorithm options (lloyd, elkan, auto, and full) produced a Silhouette Score of 0.51, indicating that the algorithm choice did not influence clustering performance for this dataset. This suggests that either the traditional or optimized approaches work similarly well in this context.

## 2. Mini Batch K-means clustering

1. **Initial Observations:**
    - Silhouette Score : The default initialization method of k-means++ produced a moderate Silhouette Score of 0.305,
    - Silhouette Score : Using the random initialization method improved the clustering performance significantly, with the Silhouette Score increasing to 0.5076. This indicates that the clusters are more well-separated, and the algorithm is performing better at distinguishing the data points.

2. **Varying Batch Sizes:**
    - The analysis was expanded to assess how batch_size affects the clustering performance. By testing different batch sizes (ranging from 1010 to 1029), it was observed that smaller or larger batch sizes led to a decline in clustering quality, as indicated by lower Silhouette Scores.

| **index** | **batch_size** | **silhouette_score** |
|---|---|---|
| **0** | 1010 | 0.5059631265306294 |
| **1** | 1011 | 0.5078312539469305 |
| **2** | 1012 | 0.5094175020706089 |
| **3** | 1013 | 0.5027956823243245 |
| **4** | 1014 | 0.4941584447309687 |
| **5** | 1015 | 0.49943191562487865 |
| **6** | 1016 | 0.2872961159405938 |
| **7** | 1017 | 0.5061312070878995 |

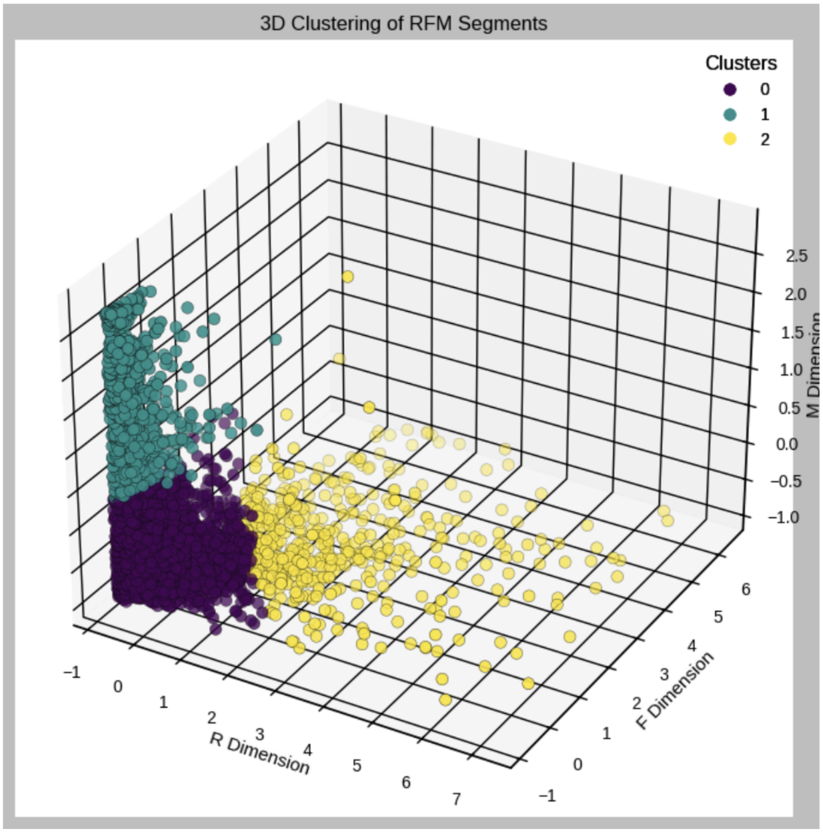
|  |  |  |
|---|---|---|
| **8** | 1018 | 0.2867381863256137 |
| **9** | 1019 | 0.3047804917206261 |
| **10** | 1020 | 0.5075945750784661 |
| **11** | 1021 | 0.5074599345231009 |
| **12** | 1022 | 0.5102159588178651 |
| **13** | 1023 | 0.5087188179923268 |
| **14** | 1024 | 0.5076203542507055 |
| **15** | 1025 | 0.5025047807142415 |
| **16** | 1026 | 0.5086768373790214 |
| **17** | 1027 | 0.50094979169377 |
| **18** | 1028 | 0.5056498359721788 |
| **19** | 1029 | 0.48601000645272235 |

**Inference :** The best-performing batch size was found to be 1022, yielding the highest Silhouette Score of 0.51. This indicates that for this dataset, a batch size around this value produces the most optimal clustering result, ensuring better-defined clusters.

3. **Final Evaluation with Optimal Batch Size:**

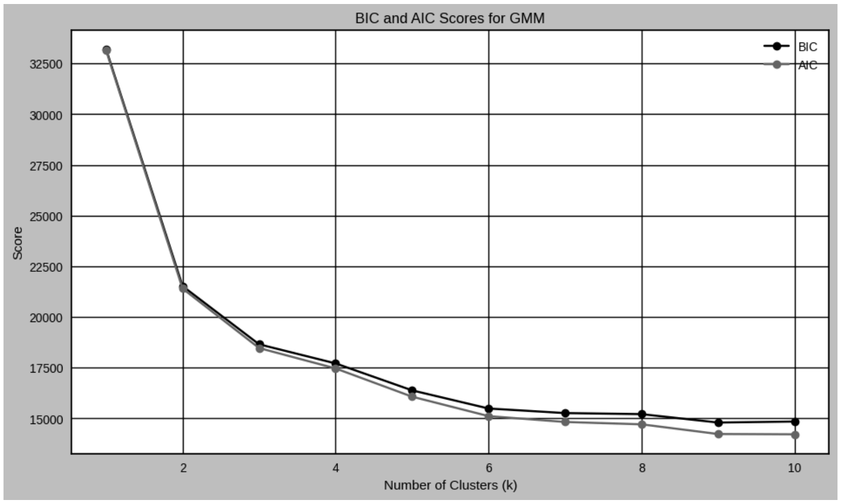
- After identifying batch_size = 1022 as the optimal setting, the clustering algorithm was re-executed with this configuration. The results showed:
- Silhouette Score: The Silhouette Score of 0.5102 confirms the good clustering performance with this batch size, indicating clear separation between clusters.
- Davies-Bouldin Score: The Davies-Bouldin score of 0.7387 is relatively low, suggesting that the clusters are compact and well-separated.Inertia: The inertia of 4445.66 represents the sum of squared distances between data points and their respective cluster centroids.

## Visualization of Mini-batch k-means clustering:



## 3. Gaussian Mixture Model

- Plot BIC and AIC scores to find the optimal number of components:



- The Gaussian Mixture Model (GMM) analysis, based on the BIC and AIC scores, revealed that the optimal number of clusters is 4. This was determined by observing the point where both the BIC and AIC scores reached their lowest values, indicating the best balance between model complexity and fit.

- **Silhouette Score for GMM with 4 Clusters:**
- For n_components = 4, the GMM yielded a Silhouette Score of 0.305, which suggests that the clusters are not very well-separated.

## 3. Impact of Hyperparameters:

### a. Maximum Iterations (max_iter):

- To explore the effect of max_iter, the algorithm was run with different iteration values from 90 to 109. The Silhouette Score remained constant at 0.30 across all iterations, suggesting that increasing the number of iterations did not improve the clustering performance. This indicates that the model converged to a stable solution early in the process, and further iterations did not enhance the clustering quality.

| index | max_iter | silhouette_score |
|---|---|---|
| **0** | 90 | 0.304808286369521 |
| **1** | 91 | 0.304808286369521 |
| **2** | 92 | 0.304808286369521 |

| | | |
|---|---|---|
| **3** | 93 | 0.304808286369521 |
| **4** | 94 | 0.304808286369521 |
| **5** | 95 | 0.304808286369521 |
| **6** | 96 | 0.304808286369521 |
| **7** | 97 | 0.304808286369521 |
| **8** | 98 | 0.304808286369521 |
| **9** | 99 | 0.304808286369521 |
| **10** | 100 | 0.304808286369521 |
| **11** | 101 | 0.304808286369521 |
| **12** | 102 | 0.304808286369521 |
| **13** | 103 | 0.304808286369521 |
| **14** | 104 | 0.304808286369521 |
| **15** | 105 | 0.304808286369521 |
| **16** | 106 | 0.304808286369521 |
| **17** | 107 | 0.304808286369521 |
| **18** | 108 | 0.304808286369521 |
| **19** | 109 | 0.304808286369521 |

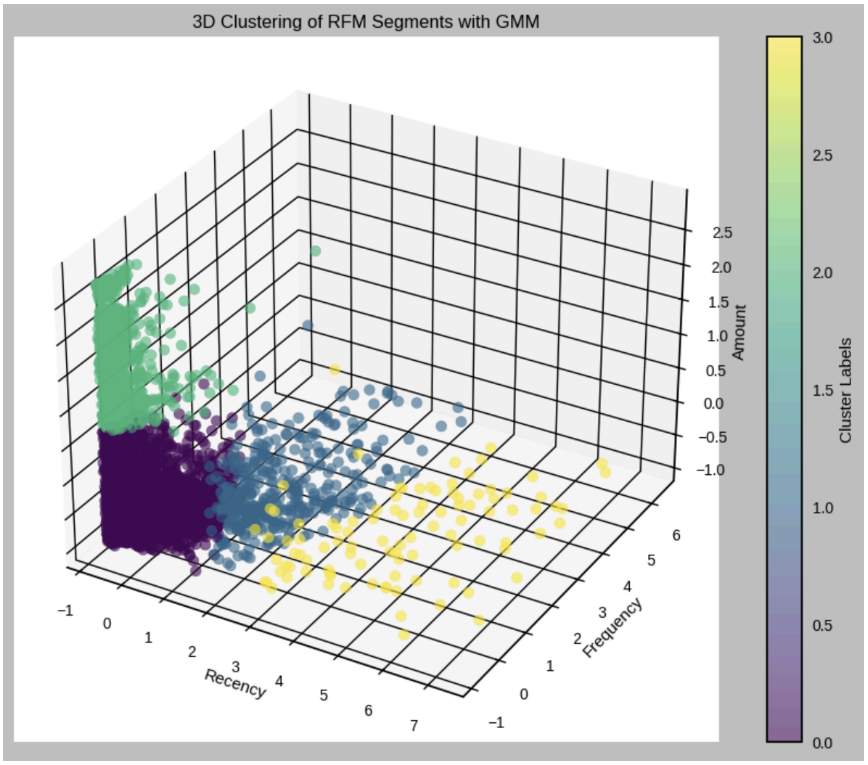
## b. Covariance Type:

The covariance_type parameter defines the structure of the covariance matrix used by the GMM. The following covariance types were tested:

- tied: All components share the same general covariance matrix.
- diag: Each component has its own diagonal covariance matrix.
- spherical: Each component has its own single variance.
- full: Each component has its own general covariance matrix.

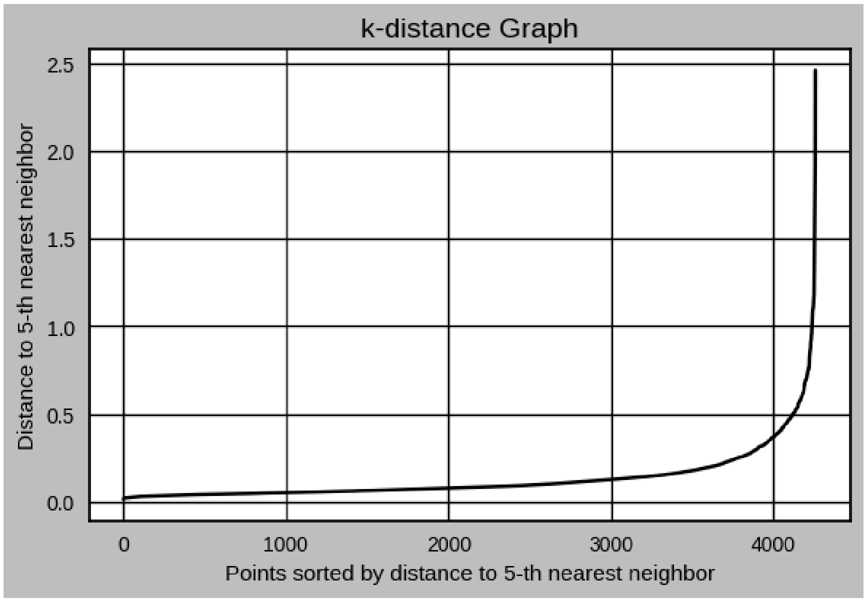
| index | covariance_types | silhouette_score |
|---|---|---|
| **0** | tied | 0.498297920747326 |
| **1** | diag | 0.12601442960080042 |
| **2** | spherical | 0.4339142357414723 |
| **3** | full | 0.304808286369521 |

## Visualization of GMM:



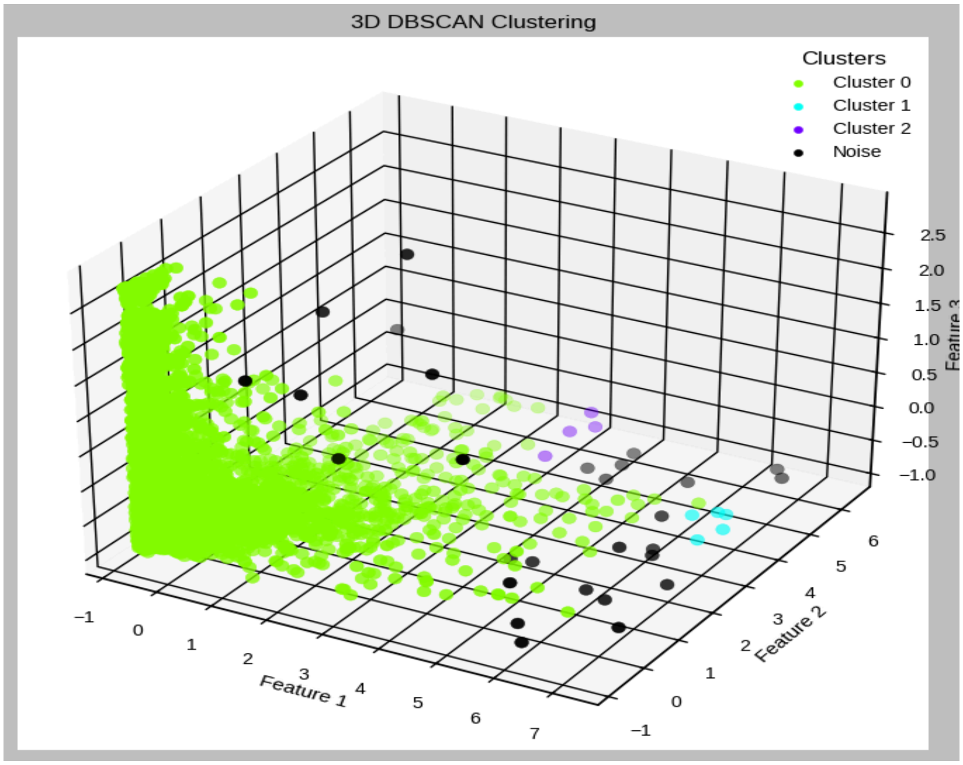
## 4. DBSCAN Clustering:

- **Determining optimal epsilon:**



**Inference :** The k-distance graph was plotted to determine the optimal epsilon value for DBSCAN. From the plot, we identified a noticeable "elbow" at a distance of 0.7, which was selected as the epsilon value for clustering.

## Clustering:



- The DBSCAN algorithm was applied with an epsilon value of 0.7 and a minimum of 5 samples per cluster. The results yielded 3 distinct clusters and identified 28 noise points (outliers) that did not belong to any cluster. A 3D scatter plot was generated to visualize the clustering, where each cluster is represented by a different color, and the noise points are shown in black.

### Silhouette Score Analysis:

- The Silhouette Score was calculated to evaluate the quality of the clusters formed by the DBSCAN algorithm. Noise points (outliers) were excluded from this calculation to ensure an accurate assessment of the clustering. The resulting Silhouette Score was 0.686, indicating that the clusters are well-defined and separated, with most points fitting well within their assigned clusters. A score closer to 1 suggests better-defined clusters, while values closer to 0 indicate overlapping clusters.

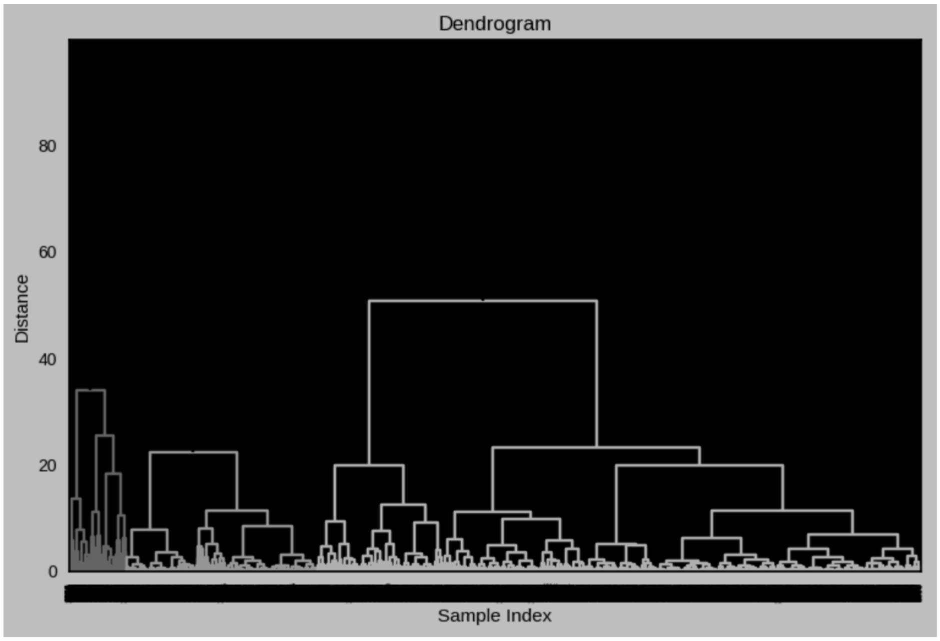
### Analysis

- The analysis should highlight customer segments derived from the clustering, providing insights into customer behaviors (e.g., high vs. low spenders).
- Comparative insights on clustering performance and interpretability among the algorithms (e.g., GMM's flexibility vs. DBSCAN's handling of noise).

## 5. Hierarchical clustering

### Finding optimal k:

- The first step involved constructing a dendrogram using hierarchical clustering. The Ward's method was used to compute the linkage matrix. A horizontal line was drawn across the largest vertical gap in the dendrogram, which appeared to be between the values around 40 and 60 on the vertical axis. This gap suggested that the optimal number of clusters could be 2 or 3.



**Optimal k = 2**

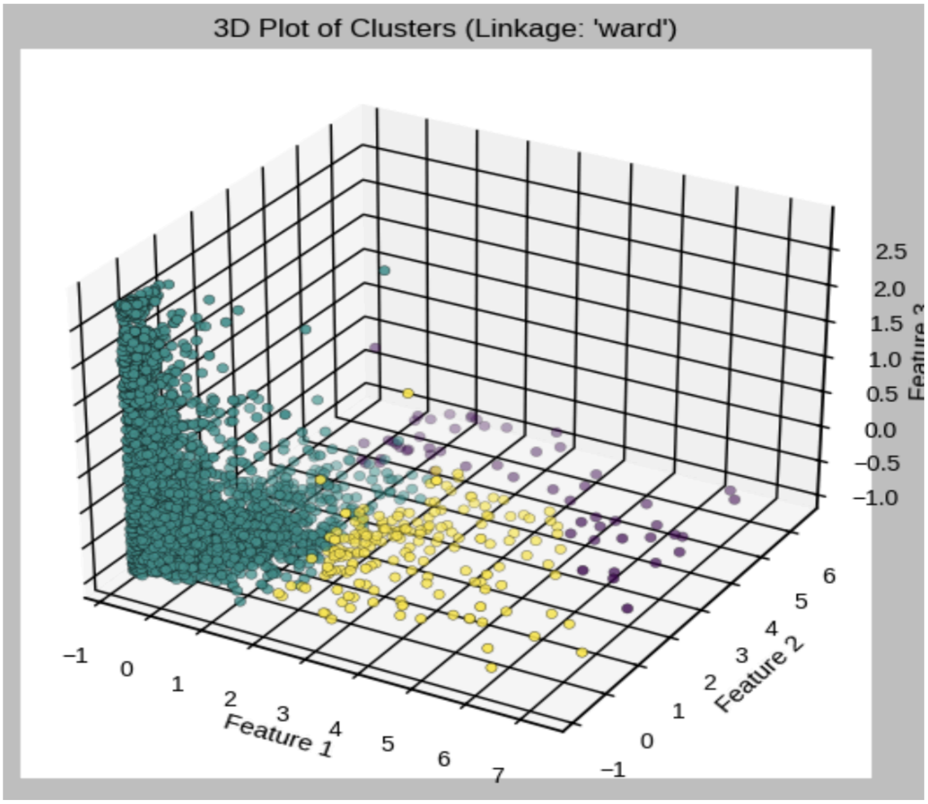
- Agglomerative Hierarchical Clustering using different linkage methods ('ward', 'complete', 'average', 'single') and calculates the silhouette score for each to evaluate the clustering performance.

## Hyperparameters:

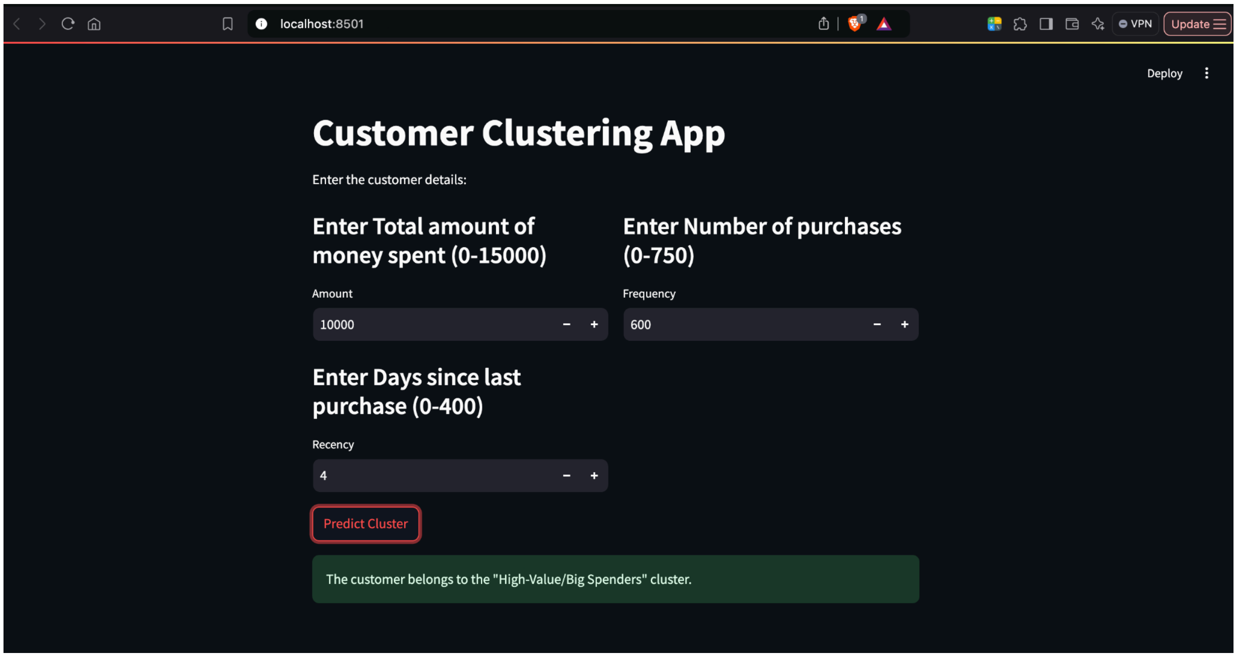
- **Ward:** Minimizes the total within-cluster variance.
- **Complete**: Uses the maximum distance between clusters.
- **Average:** Uses the average distance between clusters.
- **Single:** Uses the minimum distance between clusters.

| index | linkage | silhouette_score |
|---|---|---|
| 0 | ward | 0.6253439970876682 |
| 1 | complete | 0.6874031163200027 |
| 2 | average | 0.6855945110684218 |
| 3 | single | 0.5423217474697355 |

## Visualization



## Streamlit Deployment



# Customer Clustering App

Enter the customer details:

### Enter Total amount of money spent (0-15000)

Amount: 10000

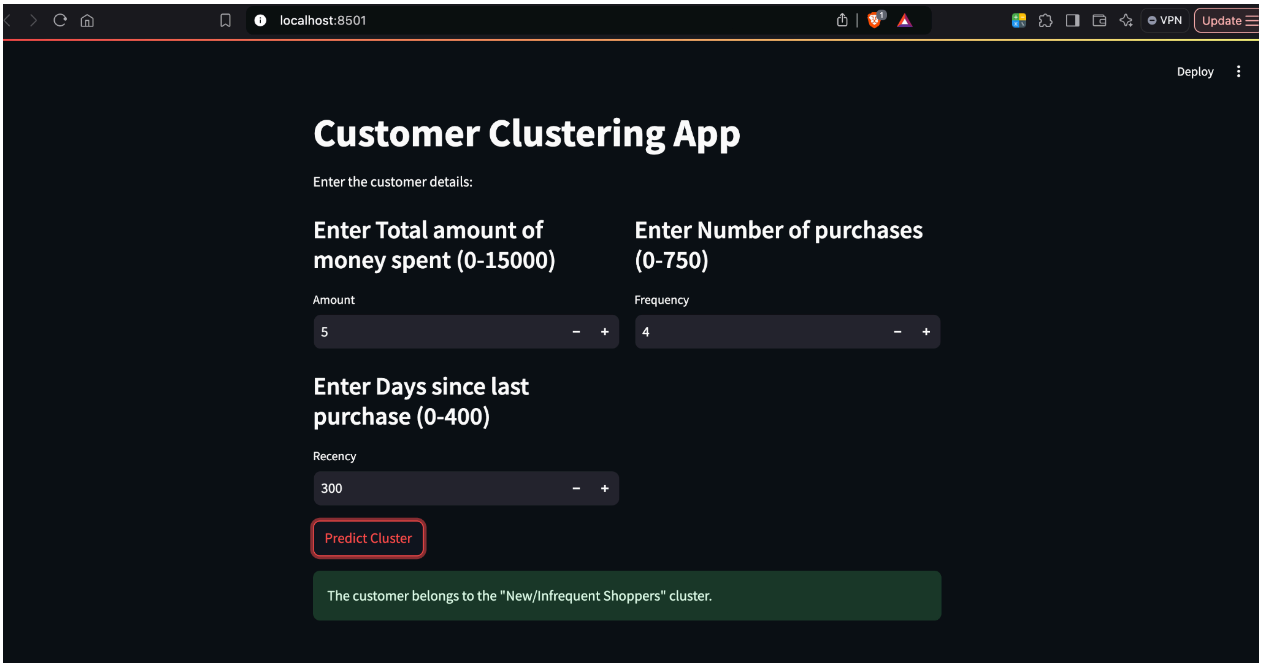
### Enter Number of purchases (0-750)

Frequency: 600

### Enter Days since last purchase (0-400)

Recency: 4

The customer belongs to the "High-Value/Big Spenders" cluster.



# Customer Clustering App

Enter the customer details:

### Enter Total amount of money spent (0-15000)

Amount: 5

### Enter Number of purchases (0-750)

Frequency: 4

### Enter Days since last purchase (0-400)

Recency: 300

The customer belongs to the "New/Infrequent Shoppers" cluster.

## Analysis

- **Best K-means Model after hyperparameter tuning:**After tuning the hyperparameters, K-Means with 3 clusters, k-means++ initialization, and default parameters provided the best clustering result, achieving a Silhouette Score of 0.51.

- **Best Mini-batch K-means Model after hyperparameter tuning:** MiniBatchKMeans with 3 clusters and random initialization produced the best result among the configurations tested, with a Silhouette Score of 0.5102 and a Davies-Bouldin Score of 0.7387.

- **Best GMM model after hyperparameter tuning:** The best clustering configuration for the GMM model was found to be n_components = 4 and covariance_type = 'tied', providing the highest Silhouette Score of 0.498 and a low Davies-Bouldin Score of 0.922.
- **DBSCAN:**After hyperparameter tuning, DBSCAN with ε = 0.7 and MinPts = 5 provided reasonably good results, with a Silhouette Score of 0.685
- **Best Hierarchical clustering Model after hyperparameter tuning:** Agglomerative Hierarchical Clustering with 2 clusters and Complete or Average Linkage produced the best clustering results. The Silhouette Score of 0.69 and Davies-Bouldin Score of 0.56 indicate that the clustering is strong, with well-defined clusters.

## Conclusion

- Agglomerative Hierarchical Clustering and Mini-Batch K-Means provided the most optimal results in terms of cluster separation and compactness, each clustering technique demonstrated strengths under different conditions. Agglomerative Hierarchical Clustering stood out in terms of both Silhouette and Davies-Bouldin Scores