Data Mining Project: Clustering on Mall Dataset

Group 8[1]
Daoyuan GUO Jinghong TAN Guofu TANG

November 2025

[1] Project No.

Table of Contents

Project Background & Dataset Description

The *Mall Customer Dataset* [6] (`https://www.kaggle.com/datasets/shwetabh123/mall-customers/data`) is a tabular dataset containing approximately 200 rows and five fields: `CustomerID`, `Gender`, `Age`, `Annual Income ($)`, and `Spending Score (1--100)`. Its clean schema makes it suitable for clustering, internal evaluation, and 2D visualization.

Project Tasks & Requirements (A–C)

What we'll do (aligned with project spec):

- **Task A:** Data acquisition, understanding and processing.
  - Clean/standardize mall customer data; document schema and basic statistics.
  - Engineer and select effective features for clustering-ready data.
- **Task B:** Clustering with data visualization.
  - Implement multiple clustering methods; compare internal metrics.
  - Provide 2D visualizations and study hyperparameter effects.
  - Choose one method and assign human-readable cluster labels.
- **Task C (Optional):** Modeling using AI-related tools.
  - Use LLM-assisted implementations, evaluate stability across seeds.
  - Provide concise interpretations and actionable suggestions.

Contributions

- Daoyuan GUO: leader, Task A, slides/report
- Jinghong TAN: member, Task B
- Guofu TANG: member, Task C

A(a) Dataset Schema and Basic Info

We've identified a brief data information via different ways.

`pd.DataFrame.info()`:

Table: Dataset Overview: Mall Customer Records

Column #	Column Name	Non-Null Count	Data Type
0	CustomerID	200 non-null	int64
1	Genre	200 non-null	object
2	Age	200 non-null	int64
3	Annual Income (k$)	200 non-null	int64
4	Spending Score (1-100)	200 non-null	int64

Additional Dataset Info:

- RangeIndex: 200 entries (0 to 199)
- Data types distribution: int64 (4 columns), object (1 column)
- Memory usage: 7.9+ KB

A(a) Metadata, Schema Details & Distributions

Also, we derive detailed column schema for further insights.

	name	dtype	non_null	nulls	unique	examples	min	max	mean	memory_bytes
0	CustomerID	int64	200	0	200	[1, 2, 3]	1.0	200.0	100.50	1732
1	Genre	object	200	0	2	[Female, Male]	NaN	NaN	NaN	10956
2	Age	int64	200	0	51	[32, 35, 31]	18.0	70.0	38.85	1732
3	Annual Income (k$)	int64	200	0	64	[54, 78, 60]	15.0	137.0	60.56	1732
4	Spending Score (1-100)	int64	200	0	84	[42, 55, 73]	1.0	99.0	50.20	1732

Figure: Manual metadata retrieval.

From `Kaggle`:

- 56% Female and 44% Male
- Right-skewed distribution on age and annual income
- 3-peak distribution assembling GMM distribution for spending score

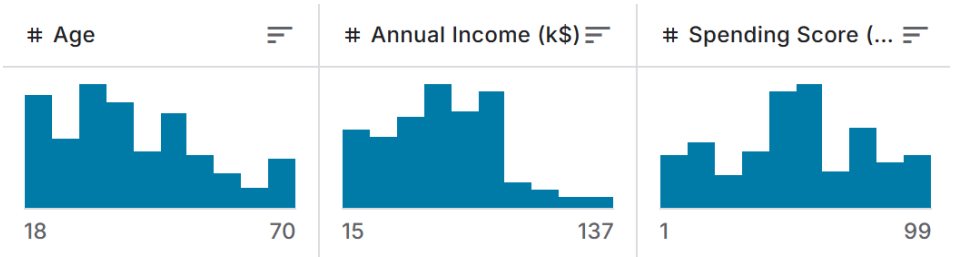


Figure: Data Distribution.

A(b) Data Cleaning & First-Stage Preprocessing

- Missing Values: None
- Duplication (after throwing out Customer ID): 0
- Outlier Detection (contamination rate=0.5):
  - Isolation Forest: 10
  - Local Outlier Factor: 10
  - Intersection: 7
- Labeling Male as 1, Female as 0

	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	21	15	81
1	0	23	16	77
2	0	31	17	40
3	0	22	17	76
4	0	35	18	6

Figure: Example of Cleaned Data.

A(c) Feature Engineering (Construction & Binning)

Feature Engineering:

1. Polynomial Features of Degree 2.
2. Discretization:
   - Binning on Age based on Kmeans (with best silhouette score)
   - Binning on Income based on Quartiles
3. Feature additions:
   - Income/Age
   - Spending Score/Income

Results 1:

- Age Binning: 6 groups, with silhouette 0.6.
- Interpretation: Applied the same Logistic Regression with accuracy and results improved.

A(c) Feature Selection: Filter Methods

Feature Selection:

1. Filter Methods

A brief filtering on the dataset, including variance filtering and correlation filtering (deleting 0-variance and high-correlation features).

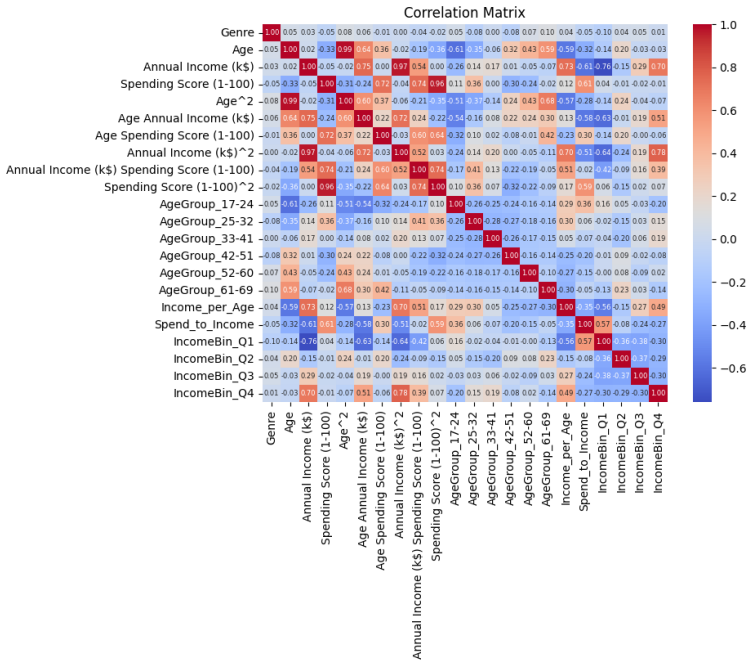


Figure: Correlation Matrix of the processed dataset. Further results could be referred at Table 13.

A(c) Feature Selection: Wrapper & Embedded Methods

2. Wrapper Methods
Selection on the dataset based on best silhouette score based on KMeans (Appendix 3).

- Leave at least 10 items for further feature selection
- Keep 3 non-binary features for initialization (sparse data lead to duplications)

3. Embedded Methods
Selection on the dataset based on Random Forest predicting KMeans Labels (based on wrapper methods).

- Feature Importances based on Random Forest (select 13)
- Final version: combine with majority vote
- Initial: 5$\Rightarrow$ Engineering: 22$\Rightarrow$ Selection: 7

A(c) Final Preprocessed Dataset (Preview)

Final Result (after standard scaling):

	Age Annual Income (k$)	AgeGroup_61-69	Annual Income (k$)	IncomeBin_Q1	IncomeBin_Q2	IncomeBin_Q3	Income_per_Age
0	-1.699593	-0.290292	-1.866777	1.646675	-0.591312	-0.623289	-1.137799
1	-1.655382	-0.290292	-1.825592	1.646675	-0.591312	-0.623289	-1.158515
2	-1.522749	-0.290292	-1.784407	1.646675	-0.591312	-0.623289	-1.322240
3	-1.650377	-0.290292	-1.784407	1.646675	-0.591312	-0.623289	-1.072825
4	-1.436830	-0.290292	-1.743222	1.646675	-0.591312	-0.623289	-1.360152

Figure: Preprocessed data.

Task B Overview: Clustering & Visualization Roadmap

Goal. Discover latent customer structure from engineered features and communicate segment differences through 2D visualizations and interpretable labels.

This section addresses Requirement B:

(a) **Algorithms & comparison:** implement multiple clustering methods (K-Medoids, GMM, HDBSCAN) and compare their performance.

(b) **2D visualization:** use t-SNE to inspect potential group structures.

(c) **Hyperparameter effects:** study how K / $n_{\text{components}}$ / density parameters affect clustering quality.

(d) **Final choice & labels:** select one method (HDBSCAN) and assign human-readable labels with brief segment profiles.

B(a) Implemented Algorithms & Internal Metrics

Implemented clustering algorithms.

- **K-Medoids (PAM):**
  - Prototype-based clustering with medoids (actual data points) as centers.
  - More robust to outliers; supports non-Euclidean metrics.
- **GMM:**
  - Model-based clustering, soft assignments via Gaussian components.
- **HDBSCAN:**
  - Density-based; automatically infers number of clusters.
  - Explicit **noise/outlier** label -1.

Internal evaluation metrics.

- Silhouette score (higher is better).
- Calinski–Harabasz index (CH, higher is better).
- Davies–Bouldin index (DB, lower is better).

B(a) Best Configurations & Model Comparison

Best configuration per algorithm.

- **K-Medoids:** $k = 7$, Euclidean distance, max_iter $= 500$.
- **GMM:** $n_{\text{components}} = 7$, covariance_type $=$ tied, max_iter $= 200$.
- **HDBSCAN:** min_cluster_size $= 15$, min_samples $= 15$, epsilon $= 2.0$, metric $=$ Manhattan.

Table: Comparison of optimal models (Requirement B(a))

Method	Silhouette	Calinski–Harabasz	Davies–Bouldin
K-Medoids	0.6002	229.75	0.6240
GMM	0.6249	230.42	0.5754
HDBSCAN	**0.6506**	**333.34**	**0.5714**

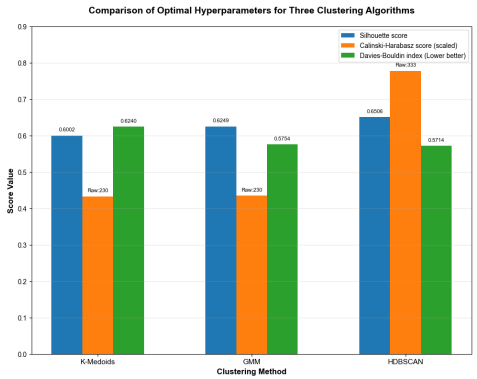


Figure: Metric comparisons between three models.

Observation.

- HDBSCAN dominates on all three metrics and also identifies outliers, making it a strong candidate for the final segmentation model.

B(a) Hyperparameter Estimation: Inertia & BIC

- Hyperparameters are first estimated by key indicators (like inertia for K-Medoids, Bayesian Information Criterion for GMM)
- They are then tuned based on internal metrics like silhouette score, Calinski–Harabasz index.

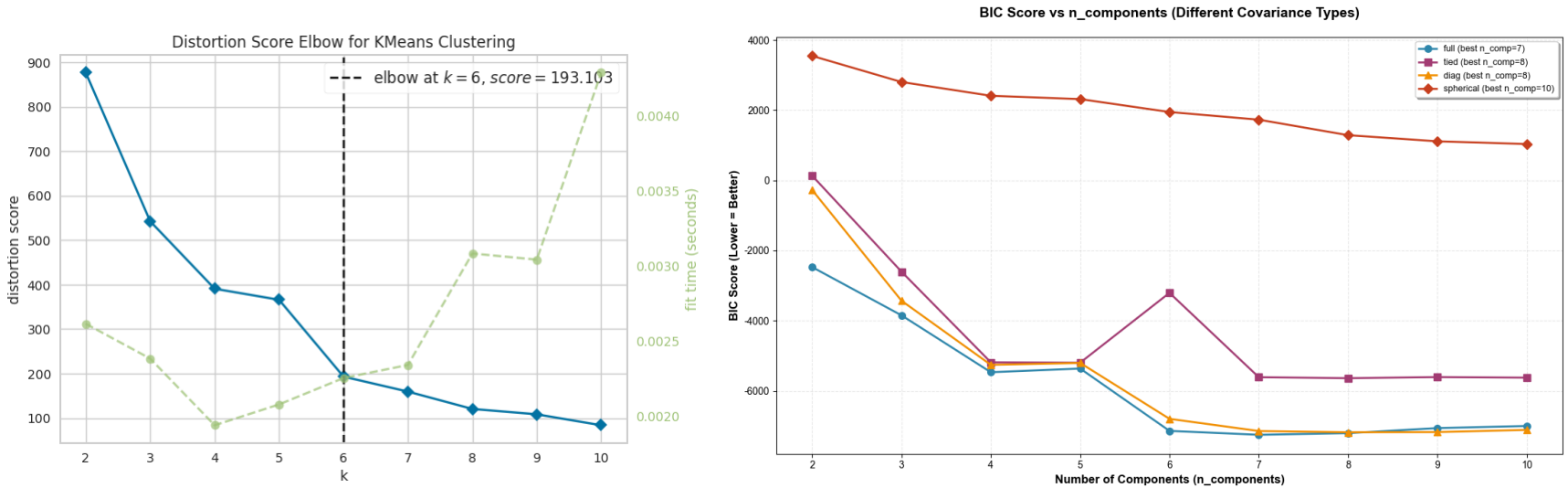


Figure: Inertia of K-Means for K-Medoids (left), and BIC scores by n components (right).

B(b) 2D Visualization: t-SNE Embedding & Feature Interpretation

t-SNE visualizations.

- Apply 2D t-SNE on the full standardized feature matrix.
- Plot t-SNE coordinates colored by cluster labels from each method:
  - K-Medoids (with medoids highlighted).
  - GMM (best configuration).
  - HDBSCAN (best configuration, with noise points shown in black).

Feature–embedding correlations.

- Compute Pearson correlations between original features and t-SNE axes:

Feature	Corr(t-SNE_1)	Corr(t-SNE_2)
Annual Income (k$)	0.875	-0.558
IncomeBin_Q1	-0.937	0.210
IncomeBin_Q2	0.242	0.781
Income_per_Age	0.650	-0.494

- t-SNE_1 mainly reflects income level and related bins; t-SNE_2 emphasizes middle vs high income and Income_per_Age differences.

B(b) 2D t-SNE Visualizations by Model

t-SNE visualizations based on clustering labels (by models with best hyperparameters).

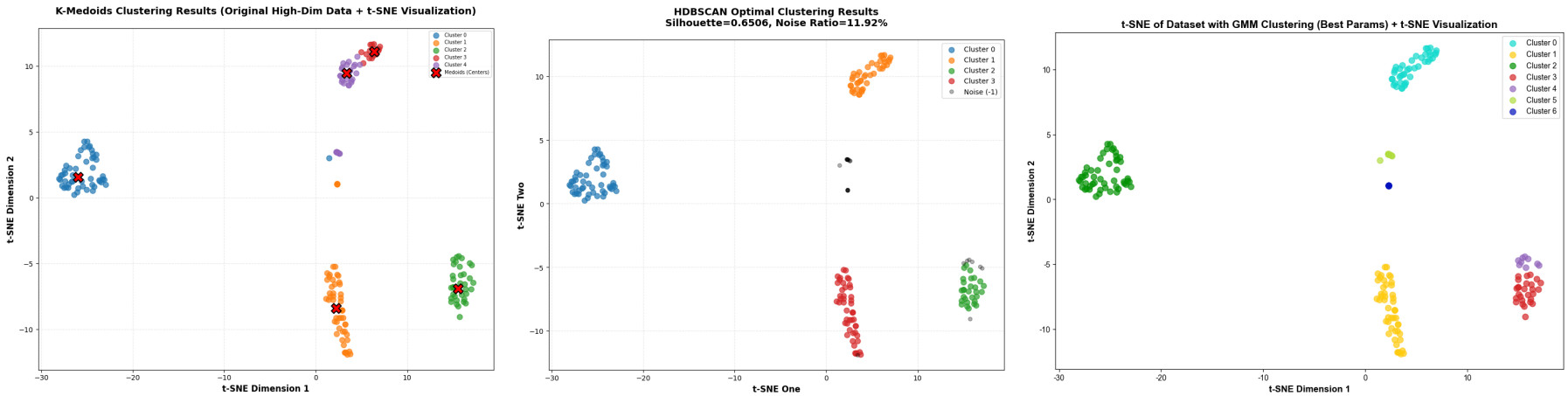


Figure: K-Medoids clustering result (left), HDBSCAN visualization (middle), and GMM visualization (right).

Visual takeaway (Requirement B(b)).

- The 2D visualization confirms that the final HDBSCAN segments correspond to distinct regions in the income–age feature space.
- This provides an intuitive, presentation-ready view of customer groups.

B(c) Hyperparameter Effects: K-Medoids (Number of Clusters)

K-Medoids grid over K.

- $K \in \{3, 4, 5, 6, 7, 8, 9\}$, max_iter $\in \{300, 500, 800\}$, metric $\in \{\text{Euclidean}, \text{Manhattan}\}$.

Table: Metrics for K-Medoids per K

K	Best Silhouette	Best CH index
3	0.5136	141.61
4	0.5535	154.82
5	0.5279	143.00
6	0.5058	125.54
7	**0.6002**	229.75
8	0.5771	**247.85**
9	0.4941	239.77

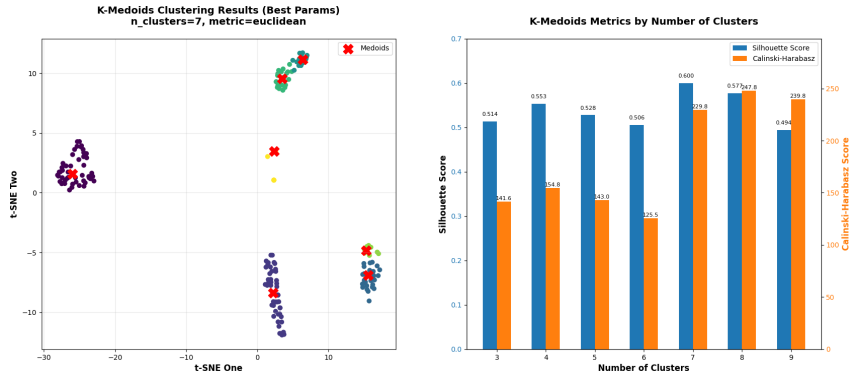


Figure: K-Medoids metrics by number of clusters.

Interpretation (Requirement B(c)).

- Very small K under-fit heterogeneity; large K fragment segments.
- $K = 7$ balances silhouette and CH, and is used as K-Medoids' best setting.

B(c) Hyperparameter Effects: GMM (BIC & Metrics)

Step 1: BIC vs $n_{\text{components}}$.

- Fix covariance_type = full, max_iter = 500, tol = 10^{-3}.
- Scan $n_{\text{components}} \in \{2, \ldots, 10\}$.
- BIC achieves minimum at $n_{\text{components}} = 7$; recommended search interval: $[5, 9]$.

Step 2: BIC vs covariance type.

- For each covariance_type $\in \{\text{full}, \text{tied}, \text{diag}, \text{spherical}\}$, determine the best $n_{\text{components}}$:
  - diag: best $n = 8$; full: $n = 7$;
  - tied: $n = 8$; spherical: $n = 10$.
- Best n under different covariance types are broadly consistent with the BIC-minimum range.

Step 3: Internal metrics (silhouette & CH).

- Silhouette-optimal:
  - $n_{\text{components}} = 7$, covariance_type = tied, max_iter = 200, tol = 10^{-4}.
  - Silhouette = 0.6249, CH = 230.42.
- CH-optimal:
  - $n_{\text{components}} = 10$, covariance_type = spherical.
  - Silhouette = 0.5048, CH = 305.05.

B(c) Hyperparameter Effects: GMM (Visual Metrics)

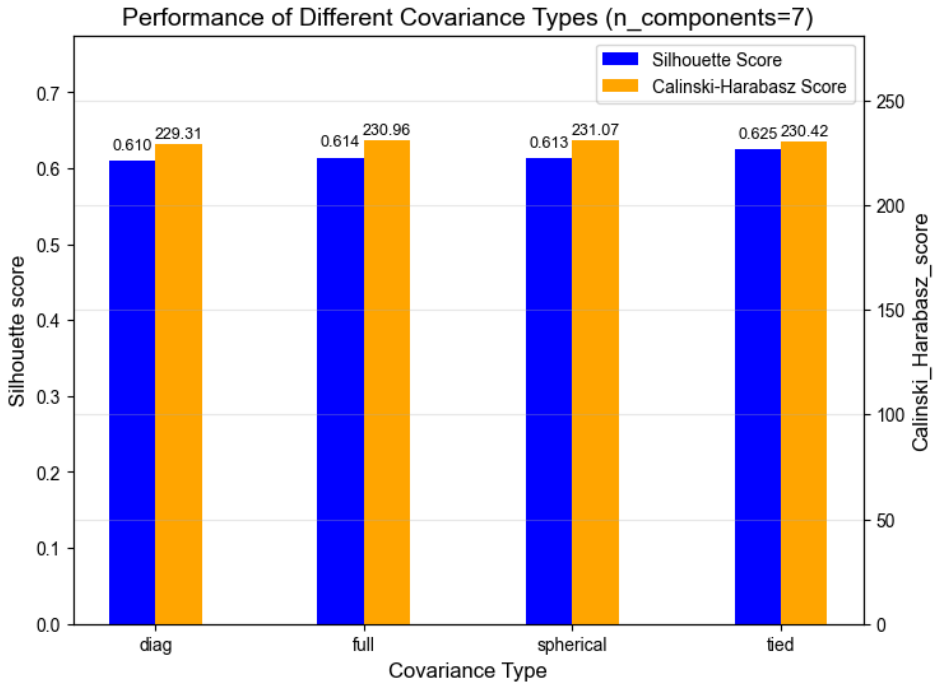


Figure: Metrics by covariance type in GMM.

B(c) Hyperparameter Effects: HDBSCAN Grid Search

Hyperparameter grid.

- min_cluster_size $\in \{10, 15, 20, 25\}$.
- min_samples $\in \{5, 10, 15\}$ with min_samples $\leq$ min_cluster_size.
- cluster_selection_epsilon $\in \{1.0, 1.5, 2.0\}$.
- cluster_selection_method $=$ eom; metric $\in \{\text{Euclidean}, \text{Manhattan}\}$.

Best configuration.

- min_cluster_size $= 15$, min_samples $= 15$, cluster_selection_epsilon $= 2.0$, method $=$ eom, metric $=$ Manhattan.
- Metrics:
  - Silhouette $=$ **0.6506**.
  - Calinski–Harabasz $=$ **333.34**.
  - Noise ratio $= 11.92\%$.
  - Effective clusters $= 4$ (excluding noise).

Effect (Requirement B(c)).

- Increasing min_cluster_size makes clusters more stable but may merge smaller groups into noise.
- A moderate epsilon (2.0) with Manhattan distance yields the best trade-off between compact clusters and a manageable noise fraction.

B(d) Final Model Choice: HDBSCAN

Motivation for choosing HDBSCAN.

- Outperforms K-Medoids and GMM on:
  - Silhouette (0.6506 vs 0.6002 vs 0.6249).
  - Calinski–Harabasz (333.34 vs 229.75 vs 230.42).
  - Davies–Bouldin (0.5714 vs 0.6240 vs 0.5754).
- Automatically identifies:
  - Four dense clusters.
  - A set of noise points (outliers) labeled -1.

Business interpretation (Requirement B(d)).

- HDBSCAN's ability to distinguish dense groups from noise is well-suited for separating **high-consumption** vs **low-consumption** regimes and isolating anomalous customers.
- We therefore use HDBSCAN as the final segmentation model.

B(d) Label Assignment Overview (t-SNE visualization)

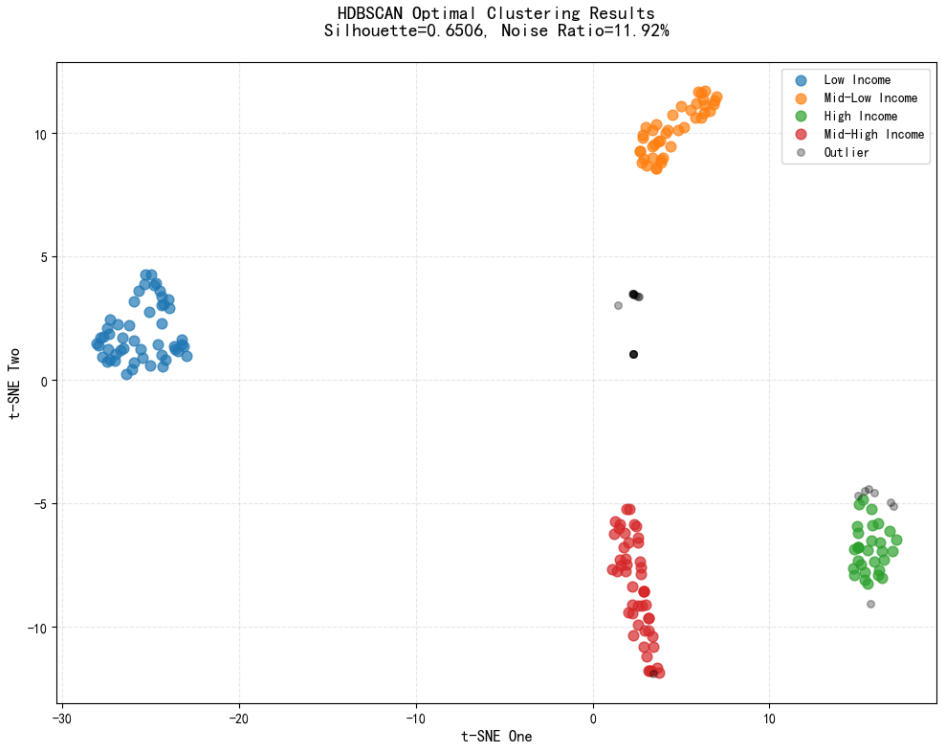


Figure: Final visualization with t-SNE provided with human-readable segment labels.

B(d) Human-Readable Segment Labels

Mapping numeric clusters to semantic segments.

- After fitting the best HDBSCAN model, numeric cluster IDs are mapped as:

$$-1 \rightarrow \text{Outlier}, 0 \rightarrow \text{Low Income}, 1 \rightarrow \text{Mid-Low Income},$$

$$2 \rightarrow \text{High Income}, 3 \rightarrow \text{Mid-High Income}.$$

- The labeled dataset is exported as `HDBSCAN_Clustered_result.csv` for downstream analysis.

Brief segment profiles.

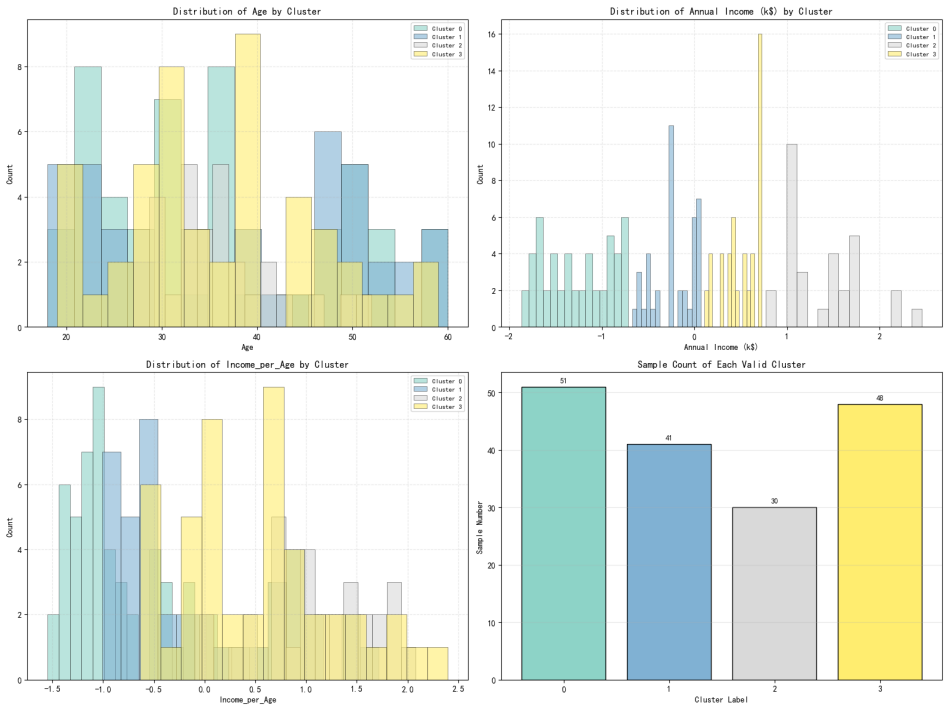
- **Low Income (Cluster 0):** low annual income and low Income_per_Age across a wide age range; conservative purchasing capacity.
- **Mid-Low Income (Cluster 1):** slightly higher income and Income_per_Age; potential to respond to affordable promotions and up-selling.
- **Mid-High Income (Cluster 3):** solid income with higher Income_per_Age; likely to buy value-added and mid-premium products.
- **High Income (Cluster 2):** highest income and high Income_per_Age; key high-value segment for premium offerings and loyalty programs.
- **Outlier (-1):** atypical age–income combinations; monitored separately for data quality or niche targeting.

B(d) Segment Distributions and Joint Patterns

Univariate distributions by segment.

- Histograms of:
  - Age.
  - Annual Income (k$).
  - Income_per_Age.

  reveal monotonic shifts from Low Income to High Income clusters.
- A bar chart of segment sizes quantifies the relative market share of each group.



B(d) Segment Distributions Visualization

Age–income joint analysis.

- Scatter plots:
  - Age vs Annual Income (k$).
  - Age vs Income_per_Age.

  colored by HDBSCAN segment, show:
  - **Low Income:** low income at almost all ages.
  - **High Income:** concentrated at higher income levels, generally mid-age.
  - **Mid-Low / Mid-High:** intermediate bands with different slopes in Income_per_Age.
- These visual patterns support a clear narrative about how consumption capacity co-evolves with age and income.

B(d) Segment Visualizations: Joint Analysis

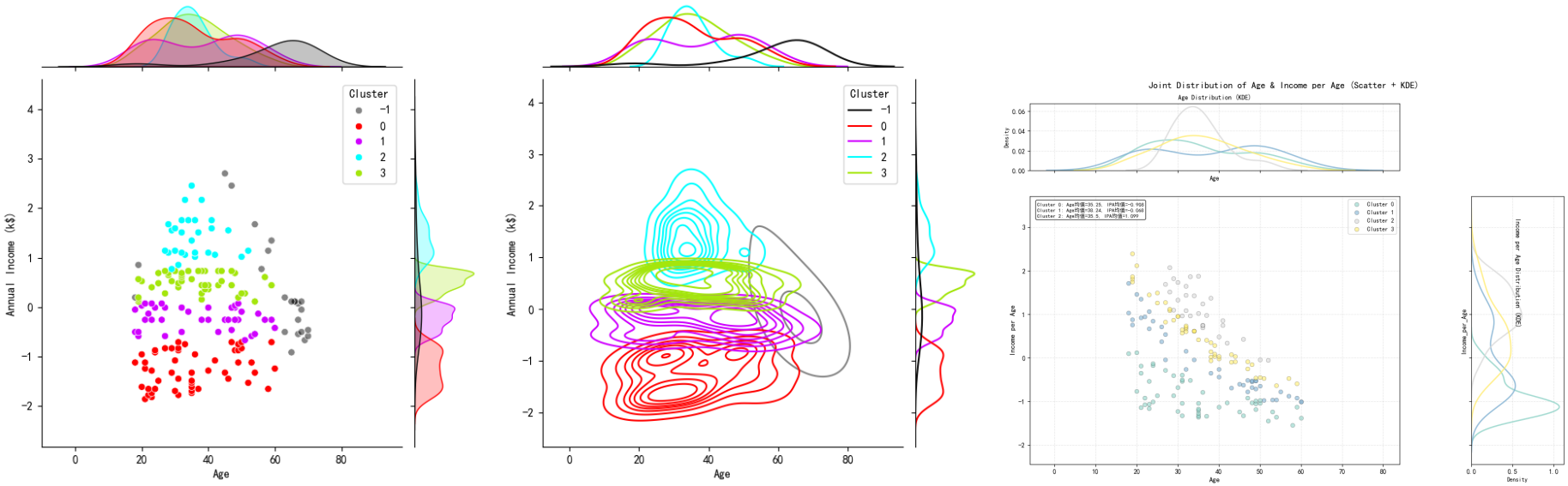


Figure: Scatter point distribution (left), and kde distribution (middle), and scatter point distribution on Age vs. Income per Age (right). Three figures on the distribution by age and income reveals differences in income while similar distribution among age.

C(a–b) AI-Generated Clustering Models & Metrics

Models include:

- KMeans (with KMeans++ by default, k=3 by silhouette)
- Gaussian Mixture (n_components=6 by Bayesian Information Criterion and Akaike Information Criterion)

Table: Clustering Evaluation Metrics

Method	**Silhouette**	**CH**	**DB**
KMeans	**0.321911**	**96.810592**	**1.122176**
GMM	0.220108	65.478527	1.324864

C(b) Visualization & Comparison (PCA)

By metrics and visualizations, KMeans enables better clustering by:

- Samples within clusters are more compact and the boundaries between clusters are clearer (higher Silhouette coefficient)
- Differences between clusters are more significant and the distribution of samples within clusters is more concentrated (higher CH index)
- Balance between intra-cluster compactness and inter-cluster separation is better (lower DB index)

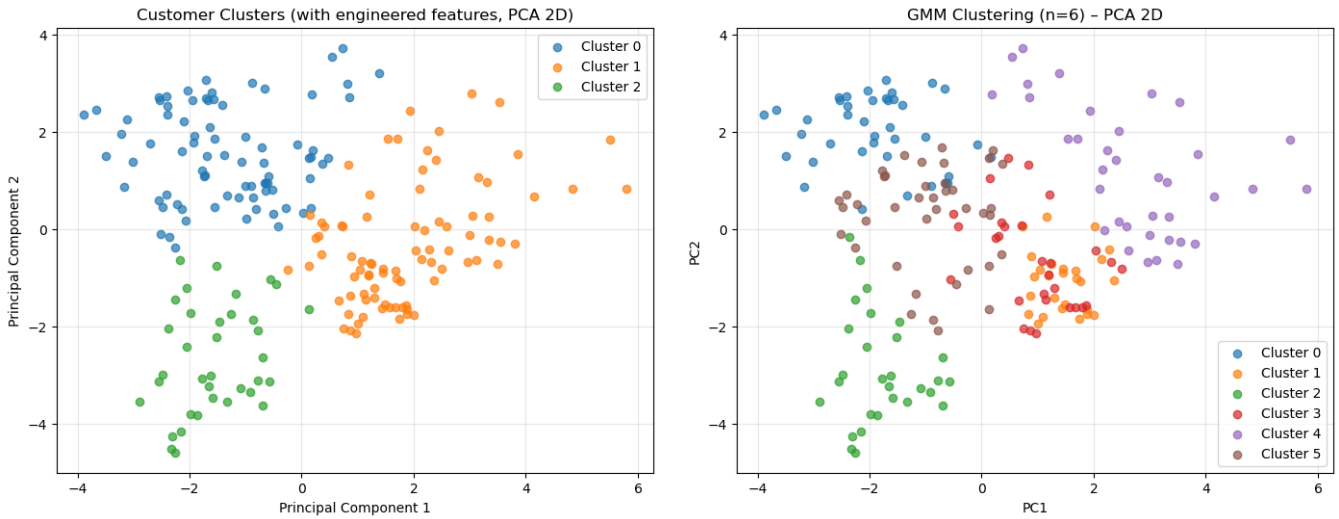


Figure: KMeans Visualization by PCA (left), and GMM Visualization by PCA (right).

C(b) Hyperparameter Selection for AI-Generated Models

Figures show that:

- k=3 is a plausible hyperparameter for elbow method and silhouette score method in KMeans.
- Akaike Information Criterion and Bayesian Information Criterion both indicate n_component=6 as the best setting.

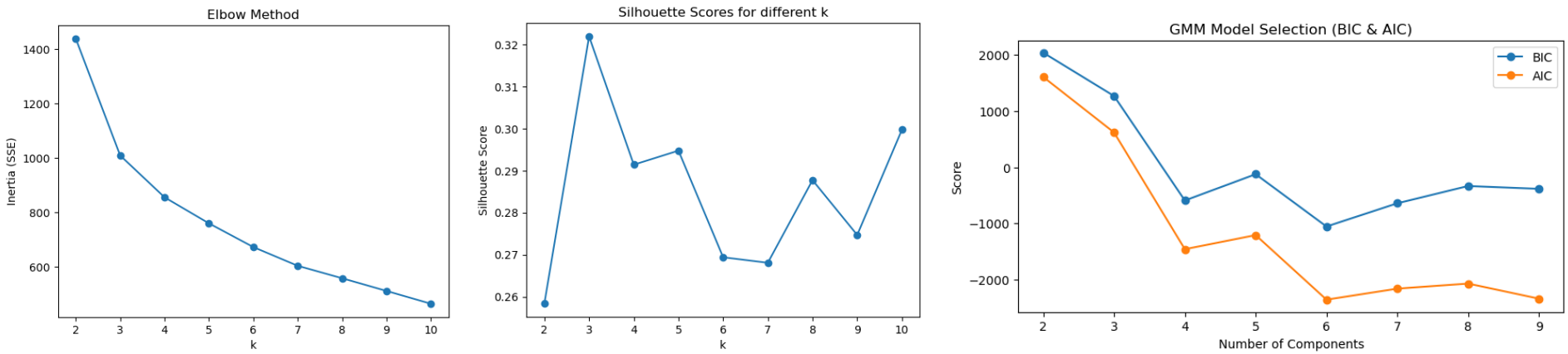


Figure: Elbow method (left), Silhouette Score (middle), and AIC+BIC (right).

C(c) Stability Check: ARI Across Seeds

KMeans ARI across seeds (10 runs):

$$[0.503,\ 0.479,\ 0.500,\ 0.407,\ 0.378,\\ 0.391,\ 0.356,\ 0.395,\ 0.426,\ 0.401]$$

GMM ARI across seeds (20 runs):

$$[0.467,\ 0.442,\ 0.414,\ 0.497,\ 0.424,\\ 0.492,\ 0.515,\ 0.494,\ 0.487,\ 0.455,\\ 0.534,\ 0.529,\ 0.456,\ 0.471,\ 0.672,\\ 0.309,\ 0.590,\ 0.591,\ 0.452,\ 0.448]$$

(Values rounded to 3 decimal places for display.)

C(c) Stability Check: Summary Statistics

Summary Statistics:

Table: Stability statistics (ARI across seeds)

Method	#Runs	Mean ARI	Std ARI	Min / Max
KMeans	10	0.424	**0.049**	**0.356** / 0.503
GMM	20	**0.487**	0.074	0.309 / **0.672**

- Both methods show **moderate stability** (ARI mostly in the range $\approx 0.4 - 0.5$).
- **KMeans**:
  - Lower mean ARI (≈ 0.424).
  - More **consistent** across seeds (lower Std).
- **GMM**:
  - Higher mean ARI (≈ 0.487): better stability *on average*.
  - Higher Std and wider range: some seeds yield very good solutions (ARI ≈ 0.67), others are noticeably worse (ARI ≈ 0.31).

C(d) KMeans Cluster Sizes and Profiles

Table: Cluster sizes

Cluster	0	1	2
Count	79	85	36

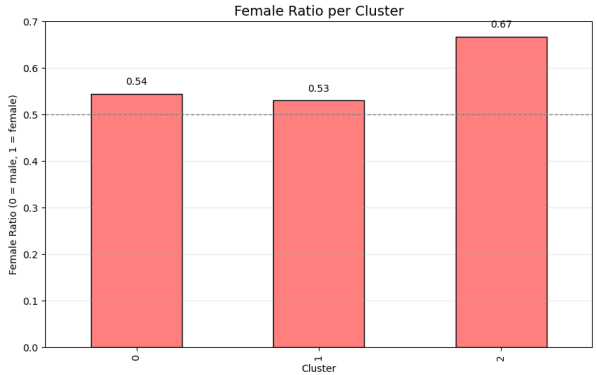


Figure: Female distribution in three clusters.

Table: Cluster profile (mean values)

Cluster	Age	Inc (k$)	Score	Spend/Inc	Inc/Age	Age_group	Inc_tier
0	53.14	54.27	37.22	0.73	1.05	2.57	0.86
1	30.54	80.25	56.59	0.75	2.69	0.80	1.40
2	27.11	27.89	63.61	2.56	1.06	0.42	0.08

C(d) KMeans Cluster 0: Older, Mid-Income, Low-Spending

Key statistics (KMeans Cluster 0)

- Average age: ≈ 53.26
- Average annual income: ≈ 52.57k
- Average spending score: ≈ 37.14
- Spending per income: ≈ 0.75
- Age group mean: ≈ 2.51 (primarily ages 36–50 and 50+)
- Income tier mean: ≈ 0.91 (mostly mid-income)

Interpretation

- Predominantly older customers with mid-level income and consistently low spending.
- Lowest spending efficiency among all clusters: very conservative purchasing despite moderate income.
- Represents a large and stable but **low-value** segment.
- More suitable for **long-term, low-cost engagement** than for aggressive campaigns.

C(d) KMeans Cluster 1: Young to Mid-Age, High-Income, Moderate-Spending

Key statistics (KMeans Cluster 1)

- Average age: ≈ 31.01
- Average annual income: ≈ 79.57k
- Average spending score: ≈ 55.26
- Spending per income: ≈ 0.73
- Age group mean: ≈ 1.06 (mainly ages 26–35)
- Income tier mean: ≈ 1.69 (mainly high-income)

Interpretation

- Younger, financially strong customers with **highest income** across clusters.
- Spending score is only **moderate**: clear *latent consumption potential*.
- Ideal for:
  - targeted marketing,
  - premium product recommendations,
  - loyalty programs to activate higher spending.

C(d) KMeans Cluster 2: Very Young, Low-Income but High-Spending

Key statistics (KMeans Cluster 2)

- Average age: ≈ 26.61
- Average annual income: ≈ 27.36k
- Average spending score: ≈ 66.88
- Spending per income: ≈ 2.73 (highest among clusters)
- Age group mean: ≈ 0.48 (mainly ≤ 25)
- Income tier mean: ≈ 0.06 (mostly low-income)

Interpretation

- Most behaviorally unique segment: lowest income but **highest spending score**.
- Extremely high spending-per-income ratio: highly active, consumption-driven younger customers.
- Likely to respond strongly to:
  - promotions,
  - social-media campaigns,
  - discount-driven marketing.
- Small in size but **high engagement** and potential for long-term high LTV if retained.

C(d) KMeans Segmentation: High-Value Ratios

Table: High-value and demographic ratios by cluster

Cluster	HighSpender	HighIncome	Female	Young	Old	Young_female
0	0.000	0.076	0.544	0.000	0.658	0.000
1	0.412	0.400	0.529	0.365	0.000	0.153
2	0.528	0.000	0.667	0.528	0.000	0.333

- Cluster 0: virtually no high spenders; older-heavy segment.
- Cluster 1: balanced high-income and high-spender ratios; sizeable young group.
- Cluster 2: highest share of high spenders and young females, despite low income levels.

C(d) KMeans Segmentation: Age and Income Tiers

Table: Age group distribution by cluster

Cluster	AgeGroup_0	AgeGroup_1	AgeGroup_2	AgeGroup_3
0	0.000000	0.000000	0.43038	0.56962
1	0.400000	0.400000	0.20000	0.00000
2	0.583333	0.416667	0.00000	0.00000

Table: Income tier distribution by cluster

Cluster	IncomeTier_0	IncomeTier_1	IncomeTier_2
0	0.215190	0.708861	0.075949
1	0.000000	0.600000	0.400000
2	0.916667	0.083333	0.000000

C(d) KMeans Segmentation: Marketing Implications

- **Cluster 0** (Older, mid-income, low-spending)
  - Large, stable but low-value segment.
  - Focus on **maintenance**: low-cost communication, basic loyalty programs.
- **Cluster 1** (Young / mid-age, high-income, moderate-spending)
  - High income with latent spending potential.
  - Target with **premium offerings**, personalized recommendations, and **loyalty / upsell** campaigns.
- **Cluster 2** (Very young, low-income, high-spending)
  - Highly active, promotion-sensitive customers.
  - Ideal for **social-media campaigns**, discounts, and engagement-driven strategies.
  - Aim to cultivate **long-term high-LTV** relationships.

C(d) GMM Customer Segmentation (6 Components)

Gaussian Mixture Model (GMM) with Engineered Features

- Model: Gaussian Mixture Model with **6 components**.
- Features:
  - Demographic: Age, Annual Income (k$), Spending Score (1–100).
  - Engineered: Spending_per_Income, Income_per_Age, Age_group, Income_tier, Age_sq, Income_sq.
  - **Genre_encoded**: gender proxy, $0 =$ male, $1 =$ female.
- Output: six well-differentiated segments with distinct demographic, behavioral and gender patterns.

Table: GMM cluster profile means (key dimensions)

Cluster	Genre	Age	Inc (k$)	Score	Spend/Inc	Inc_tier
0	0.28	59.26	50.18	38.54	0.78	0.77
1	1.00	27.67	71.22	63.15	0.88	1.15
2	0.59	26.48	24.03	66.21	2.88	0.00
3	0.00	29.97	65.90	53.34	0.84	1.00
4	0.47	38.00	99.58	47.03	0.48	2.00
5	1.00	42.68	50.98	41.80	0.86	0.70

C(d) GMM Cluster Profiles (1/3)

Cluster 0 — Older, Mid-Income, Male-Majority, Low-to-Moderate Spenders

- Center: Age ≈ 59.3, Income ≈ 50.2k, Score ≈ 38.5, Genre_encoded ≈ 0.28 (male-majority).
- Income tier ≈ 0.77 (mid-income), Spending_per_Income ≈ 0.78.
- Older, predominantly male, conservative spending despite moderate income.
- Large, low-risk but **low-growth** segment.

Cluster 1 — Very Young, High-Income, All-Female, High Spenders

- Center: Age ≈ 27.7, Income ≈ 71.2k, Score ≈ 63.1, Genre_encoded $= 1.0$ (100% female).
- Income tier ≈ 1.15, Income_per_Age ≈ 2.63 (very high).
- Very young, financially strong female consumers with high discretionary spending.
- **High-income, high-engagement** segment: ideal for premium brands and loyalty programs.

C(d) GMM Cluster Profiles (2/3)

Cluster 2 — Very Young, Low-Income, Female-Majority, High Spending Intensity

- Center: Age ≈ 26.5, Income ≈ 24.0k, Score ≈ 66.2, Genre_encoded ≈ 0.59 (female-majority).
- Income tier ≈ 0 (lowest), Spending_per_Income ≈ 2.88 (highest of all clusters).
- Young, low-income but **extremely high** spending intensity.
- Ideal for discount-based, social-media and influencer-driven campaigns.

Cluster 3 — Young Male Professionals, High Income, Controlled Spending

- Center: Age ≈ 30.0, Income ≈ 65.9k, Score ≈ 53.3, Genre_encoded $= 0.0$ (100% male).
- Income_per_Age ≈ 2.35, Income tier ≈ 1.0.
- Young males with strong purchasing power and **moderate**, selective spending.
- High-value but **not impulsive**, likely focused on tech, sports, and lifestyle goods.

C(d) GMM Cluster Profiles (3/3)

Cluster 4 — Mid-Age, Very High Income, Balanced Gender, Moderate Spending

- Center: Age ≈ 38.0, Income ≈ 99.6k, Score ≈ 47.0, Genre_encoded ≈ 0.47 (balanced).
- Income tier ≈ 2.0 (highest), Spending_per_Income ≈ 0.48.
- Very high-income, mid-age, spending is stable but not aggressive.
- Good candidates for long-term retention, VIP and premium product strategies.

Cluster 5 — Older Mid-Age, All-Female, Mid-Income, Low-to-Moderate Spenders

- Center: Age ≈ 42.7, Income ≈ 51.0k, Score ≈ 41.8, Genre_encoded $= 1.0$ (100% female).
- Age_group ≈ 1.88 (36–50), Spending_per_Income ≈ 0.86.
- Female-only, mid-age, mid-income, with predictable moderate spending.
- Stable, household-oriented segment; reliable but **not high-growth**.

C(d) GMM: Gender and Behavioral Patterns

1. **Gender strongly differentiates clusters**
   - All-female clusters: **1** and **5**.
   - All-male cluster: **3**.
   - Female-majority: **2**; Male-majority: **0**; Balanced: **4**.
   - $\Rightarrow$ Gender is a key axis of behavioral segmentation, beyond age/income.
2. **Income does not fully explain spending**
   - Cluster 4: highest income (≈ 100k) but only moderate score (≈ 47).
   - Cluster 2: lowest income (≈ 24k) but very high score (≈ 66) and highest Spending_per_Income.
   - $\Rightarrow$ Spending behavior is shaped by **preferences and lifestyle**, not just financial capacity.

C(d) GMM: High-Engagement vs Conservative Segments

High-engagement young female segments

- **Cluster 1**: high-income, high spending, all-female.
- **Cluster 2**: low-income but extremely high spending intensity, female-majority.
- $\Rightarrow$ Most commercially valuable for growth, campaigns and cross-selling.

Conservative, older segments

- **Cluster 0** and **Cluster 5**: older, mid-income, low-to-moderate spending.
- Provide stable revenue, but low responsiveness to aggressive marketing.
- Better suited to **maintenance and retention** rather than growth-focused actions.

C(d) GMM: Strategic Recommendations

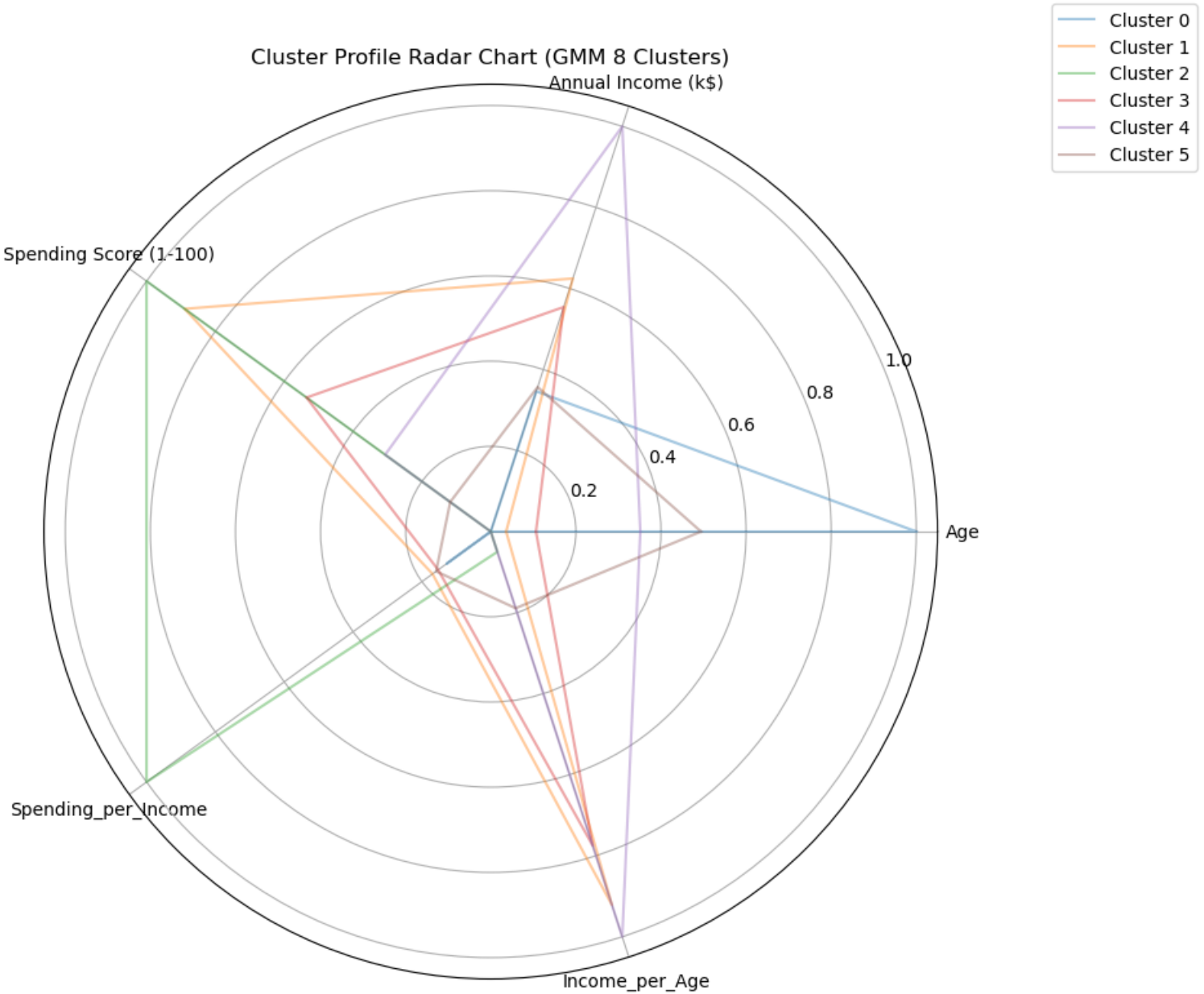
High-priority target segments

- **Cluster 1**: young, high-income, all-female
  - Premium positioning, personalized offers, loyalty and subscription programs.
- **Cluster 2**: young, low-income but high-intensity spenders
  - Social-media, influencer and discount-driven campaigns.
- **Cluster 3**: high-income young males
  - Tech, sports, and lifestyle-focused promotions; selective premium offers.

Mid- and low-priority segments

- **Cluster 4**: very high-income, controlled spending
  - VIP / loyalty programs, long-term retention, brand-building.
- **Clusters 0 & 5**: older conservative groups
  - Stability-based marketing, essential products, low-cost engagement.

C(d) GMM Cluster Shape (Radar Chart)



C(d) GMM: Female Ratio per Cluster

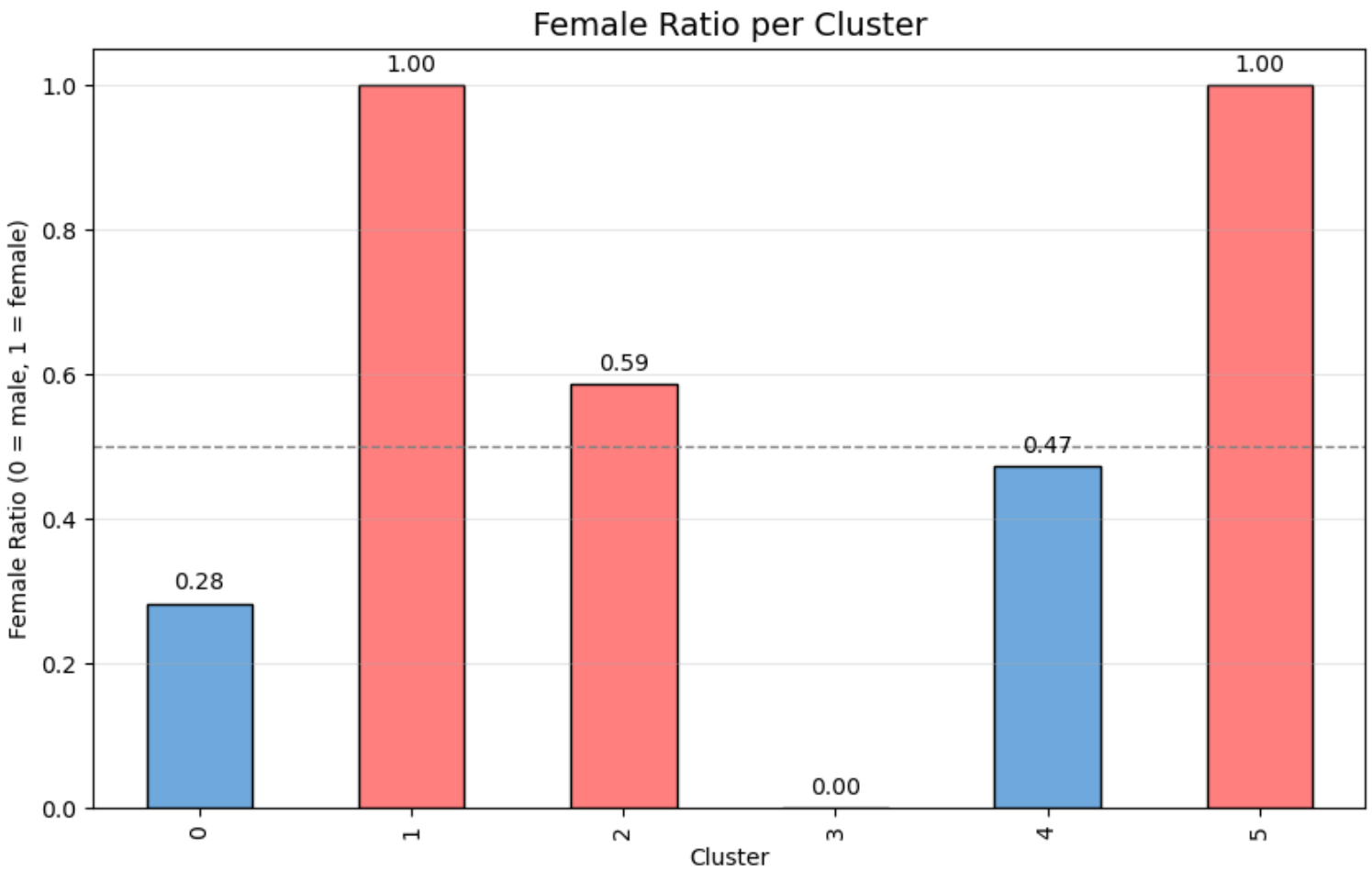


Figure: Female ratio by GMM cluster (Genre_encoded, 0 = male, 1 = female).

Conclusion

End-to-end pipeline.

- Delivered a full pipeline from schema audit, outlier removal and feature engineering (Task A) to clustering, internal validation and segment profiling (Task B).
- Combined filter, wrapper and embedded selection to obtain a low-redundancy, clustering-ready feature space.

Model selection.

- Benchmarked K-Medoids, GMM and HDBSCAN using Silhouette, Calinski–Harabasz and Davies–Bouldin indices.
- HDBSCAN (Silhouette ≈ 0.65, best CH, lowest DB) with explicit noise handling was chosen as the deployment model.

Segmentation insights & AI tools.

- t-SNE / PCA confirmed income- and Income_per_Age–driven separation; semantic labels (Low–High Income + Outliers) map directly to marketing actions.
- LLM-assisted KMeans / GMM provide independent, moderately stable baselines (ARI) that corroborate the main segmentation narrative.

Limitations & Future Work

Data & evaluation.

- Single small cross-sectional dataset (200 customers, one mall) without transaction history or temporal dynamics.
- Rely solely on internal criteria (Silhouette, CH, DB, ARI); no external labels or business KPIs to quantify uplift.

Methodology.

- HDBSCAN is sensitive to scaling, metric choice and hyperparameters; cluster count and noise rate may shift under alternative settings.
- t-SNE is used as a non-linear visualization only and may distort distances; we did not explore deep or semi-supervised clustering.

AI tools and outlook.

- LLM-generated code is validated only on this clean, small dataset; robustness on larger, noisier corpora is unknown.
- Future work: richer behavioral/time-series features, external business KPIs for validation, cross-dataset stress tests, and deep representation learning for scalable segmentation.

References I

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[3] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2): 224–227, 1979.

[4] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1): 193–218, 1985.

[5] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 (2):461–464, 1978.

[6] SHWETABH123. Mall customers dataset. `https://www.kaggle.com/datasets/shwetabh123/mall-customers`, 2025. Accessed: 2025-11-27.

Complement Implementation Details

Age Binning Details:

Table: Age Binning result

Age Group	17-24	25-32	33-41	42-51	52-60	61-69
Count	36	43	42	38	19	15

Logistic Regression:

- Pick Genre as labels to conduct classification task
- Remain parameters as default
- Previous: Accuracy 0.5751, Roc AUC 0.5016
- Afterwards: Accuracy 0.6788, Roc AUC 0.6458

Complement Implementation Details

Table: Feature Selection Process for Customer Data

Selection Stage	Feature List
Numeric Columns (Initial)	Genre, Age, Annual Income (k$), Spending Score (1-100), Age^2, Age Annual Income (k$), Age Spending Score (1-100), Annual Income (k$)^2, Annual Income (k$) Spending Score (1-100), Spending Score (1-100)^2, AgeGroup_17-24, AgeGroup_25-32, AgeGroup_33-41, AgeGroup_42-51, AgeGroup_52-60, AgeGroup_61-69, Income_per_Age, Spend_to_Income, IncomeBin_Q1, IncomeBin_Q2, IncomeBin_Q3, IncomeBin_Q4
Kept after Variance Filter	Genre, Age, Annual Income (k$), Spending Score (1-100), Age^2, Age Annual Income (k$), Age Spending Score (1-100), Annual Income (k$)^2, Annual Income (k$) Spending Score (1-100), Spending Score (1-100)^2, AgeGroup_17-24, AgeGroup_25-32, AgeGroup_33-41, AgeGroup_42-51, AgeGroup_52-60, AgeGroup_61-69, Income_per_Age, Spend_to_Income, IncomeBin_Q1, IncomeBin_Q2, IncomeBin_Q3, IncomeBin_Q4
Dropped (High Correlation > 0.9)	Age^2, Annual Income (k$)^2, Spending Score (1-100)^2
Kept after Correlation Filter (Final)	Genre, Age, Annual Income (k$), Spending Score (1-100), Age Annual Income (k$), Age Spending Score (1-100), Annual Income (k$) Spending Score (1-100), AgeGroup_17-24, AgeGroup_25-32, AgeGroup_33-41, AgeGroup_42-51, AgeGroup_52-60, AgeGroup_61-69, Income_per_Age, Spend_to_Income, IncomeBin_Q1, IncomeBin_Q2, IncomeBin_Q3, IncomeBin_Q4

Complement Implementation Details

Wrapper selects (silhouette=0.5193 with k=7):
`['IncomeBin_Q4', 'IncomeBin_Q1', 'Genre', 'AgeGroup_61-69', 'AgeGroup_52-60', 'Annual Income (k$)', 'IncomeBin_Q3', 'IncomeBin_Q2', 'Age Annual Income (k$)', 'Income_per_Age']`
Embedded Feature Importance:

Table: Feature Importances (Descending Order)

Feature	Importance	Feature	Importance	Feature	Importance
Spending Score (1-100)	0.1264	Annual Income (k$)	0.1244	Annual Income (k$) Spending Score (1-100)	0.1132
Spend_to_Income	0.0938	Age	0.0906	Age Annual Income (k$)	0.0880
IncomeBin_Q1	0.0778	Income_per_Age	0.0689	Age Spending Score (1-100)	0.0644
AgeGroup_17-24	0.0506	AgeGroup_61-69	0.0408	IncomeBin_Q2	0.0308
IncomeBin_Q3	0.0067	AgeGroup_25-32	0.0066	IncomeBin_Q4	0.0059
AgeGroup_42-51	0.0058	AgeGroup_33-41	0.0023	AgeGroup_52-60	0.0018
Genre	0.0012				

Appendix: Model Selection Criteria (AIC & BIC)

Context in this project

- For Gaussian Mixture Models (GMM), we compare candidate models with different numbers of components.
- **AIC** and **BIC** are used to trade off fit vs. complexity.
- Lower AIC/BIC values indicate a better balance between data likelihood and model parsimony.

Akaike Information Criterion (AIC)

- Introduced by Akaike [1].
- $\hat{L}$ is the maximized likelihood, k the number of free parameters:$\mathrm{AIC} = 2k - 2\ln \hat{L}$.
- Penalizes models with more parameters linearly in k.

Bayesian Information Criterion (BIC)

- Proposed by Schwarz [5].
- For sample size n: $\mathrm{BIC} = k\ln n - 2\ln \hat{L}$.
- Stronger penalty on model complexity than AIC (via $\ln n$).
- Often favors more parsimonious GMMs, especially for large n.

Appendix: Adjusted Rand Index (ARI)

Purpose

- Measures agreement between two partitions of the same dataset and adjusts for chance [4].
- Used here to quantify stability across random seeds and to compare clustering solutions.

Definition

- Let n_{ij} be the contingency table between partitions U and V, with row sums $a_i = \sum_j n_{ij}$, column sums $b_j = \sum_i n_{ij}$, and $n = \sum_{ij} n_{ij}$.
- Using $\binom{x}{2} = x(x-1)/2$,

$$\mathrm{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{n}{2}}.$$

Interpretation

- $\mathrm{ARI} = 1$: identical partitions.
- $\mathrm{ARI} \approx 0$: random agreement.
- $\mathrm{ARI} < 0$: worse than random.

Appendix: Calinski–Harabasz (CH) Index

Purpose

- Evaluates clustering quality using **between-cluster separation** vs. **within-cluster compactness**.
- Originally proposed by Caliński and Harabasz [2].

Definition

- Let n be the total number of points, k the number of clusters.
- Within-cluster dispersion: $W_k = \sum_{r=1}^{k} \sum_{x \in C_r} \|x - \mu_r\|^2$, where μ_r is the centroid of cluster C_r.
- Between-cluster dispersion: $B_k = \sum_{r=1}^{k} n_r \|\mu_r - \mu\|^2$, where $n_r = |C_r|$ and μ is the global mean.
- The Calinski–Harabasz index: $\mathrm{CH}(k) = \frac{B_k/(k-1)}{W_k/(n-k)}$.

Interpretation

- **Higher** CH index $\Rightarrow$ better-defined, more separated clusters.

Appendix: Davies–Bouldin (DB) Index

Purpose

- Measures average **cluster similarity** based on within-cluster scatter and between-cluster separation [3].
- Used for internal validation of clustering structures.

Definition

- For each cluster i, define $s_i = \frac{1}{n_i} \sum_{x \in C_i} \|x - \mu_i\|$ as the average distance of points to their centroid.
- Let $d_{ij} = \|\mu_i - \mu_j\|$ be the distance between centroids.
- Define $R_{ij} = \frac{s_i + s_j}{d_{ij}}, \quad i \neq j$, and for each i, $R_i = \max_{j \neq i} R_{ij}$.
- The Davies–Bouldin index: $\mathrm{DB} = \frac{1}{k} \sum_{i=1}^{k} R_i$.

Interpretation

- **Lower** DB index $\Rightarrow$ more compact and better-separated clusters.

Appendix: Clustering Models: K-Medoids & HDBSCAN

K-Medoids vs. K-Means. Compared to K-Means, K-Medoids find medoids instead of centroids at each iteration, making the result more robust to noises.

HDBSCAN Intuition:

- Dense clusters: MST branches persisting over density levels.
- Sparse regions: become noise or transient clusters.

Appendix: HDBSCAN Cluster Tree & Stability

1. **Build Cluster Hierarchy:**
   - MST edges (sorted by d_{mrd}) as merging/splitting events.
   - Cluster tree records evolution of connected components.
2. **Condense by `min_cluster_size`:**
   - Prune branches with size $<$ `min_cluster_size`.
   - Condense tree into meaningful clusters.
3. **Cluster Stability:**
   - Stability: persistence over density threshold.
   - Select clusters to maximize total stability, avoid overlap.
   - Unassigned points labeled as noise (-1).

Project Results:

- Hyperparameters: `min_cluster_size` $= 15$, `min_samples` $= 15$, $\varepsilon = 2.0$, Manhattan metric.
- 4 dense clusters $+ \approx 12\%$ noise.
- Best internal metrics (silhouette, CH, DB).

Appendix: Local Outlier Factor (LOF)

1. **Core Principle: Local Density Deviation**
   - Calculate k-distance: Distance to the k-th nearest neighbor for each point.
   - Define reachability distance: Maximum of k-distance of target point and actual distance between two points.
   - Compute local reachability density (LRD): Inverse of average reachability distance from k neighbors.
2. **Outlier Score Calculation:**
   - LOF = Average LRD of k neighbors / LRD of current point.
   - Score > 1: Point is less dense than neighbors (potential outlier); Score $= 1$: Point is in normal cluster.
3. **Key Characteristics:**
   - Relies on local density (adapts to non-uniform data distributions).
   - Sensitive to parameter k (too small: noisy results; too large: misses local outliers).

Appendix: Isolation Forest

1. **Core Principle: Anomaly Isolation via Random Partitioning**
   - Builds ensemble of "isolation trees" (binary trees) via random splits.
   - Anomalies: Fewer splits needed to isolate (shorter path length in trees).
   - Normals: More splits needed (longer path length).
2. **Tree Construction & Path Length:**
   - Randomly select a feature and a split value (between min/max of the feature).
   - Recursively split until each node has 1 point (isolated) or reaches max tree depth.
   - Path length: Number of splits from root to isolated point.
3. **Anomaly Score Calculation:**
   - Normalize path length using average path length of "normal" points (from reference trees).
   - Score $\in$ [0,1]: Score $>$ 0.5 (anomaly); Score $=$ 0.5 (normal); Score .5 (certain normal).