

Group Assignment 1

Group 1:

Sally Johnstone, Jae Oh, Lu Chen, Ehsan Haghian, Peter Martin

Two data sets, five models

Two data sets:

1. eBay Shill Bidding
2. Online Retail

Five models:

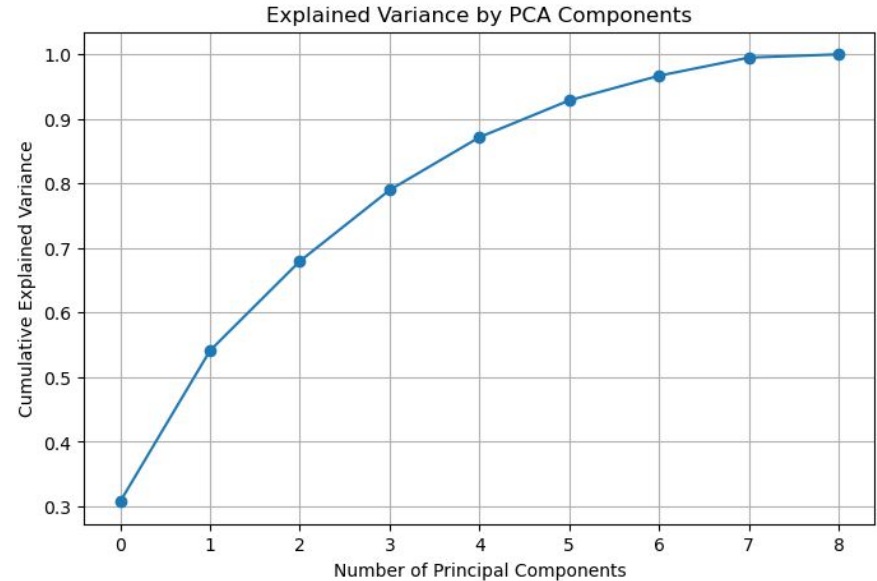
1. Partitional: Spectral Clustering
2. AHC (Agglomerative Hierarchical Clustering): BIRCH
3. Density-Based: DBSCAN
4. Grid-Based: CLIQUE
5. Model & Graph-Based: Gaussian Mixture Model

Dataset 1 - eBay Shill Bidding

- Shill Bidding: *The fraudulent practice of faking bids in online auction to artificially inflate the final price*
- 6321 Rows, 13 Features
- Known K. A row either represents an instance of shill bidding or it doesn't.
K=2
- Anomaly detection can be tackled with clustering algorithms

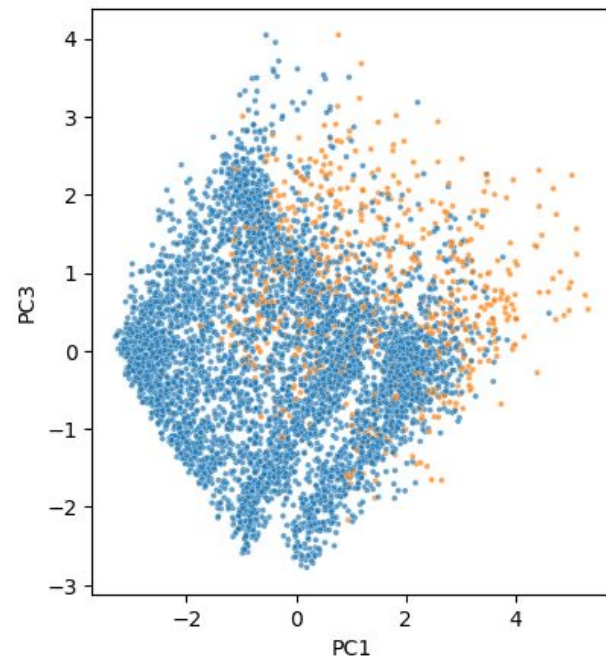
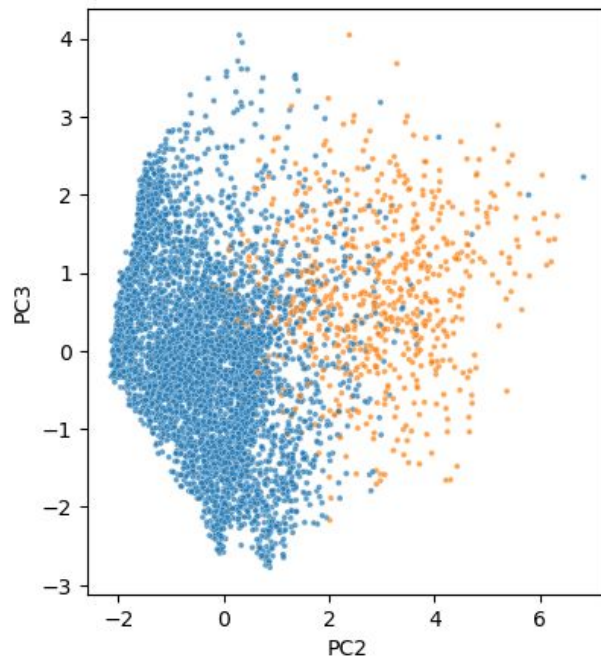
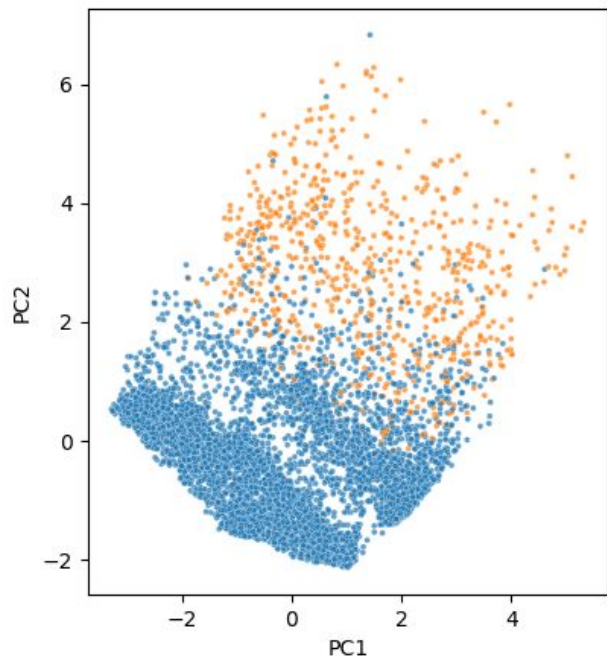
Dataset 1 - eBay Shill Bidding pre-processing

- Record_ID, Auction_ID and Bidder_ID converted to object
- Record_ID set as the index
- No missing values, no duplicate records
- Identified need to remove 'Class' variable from clustering analysis
- Used standard scalar on remaining numeric values to normalize data
- PCA to reduce variables and remove correlations



Dataset 1 - eBay Shill Bidding pre-processing

True Shill Bidding Clusters



Dataset 2: Online Retail

Dataset Overview

- UK-based registered online retail transactions.
- Contains about 542000 instances along with 6 features, including key fields like InvoiceNo, Quantity, UnitPrice, and CustomerID.
- Allow us to analyze purchasing patterns at a customer level.
- The primary goal is to perform RFM analysis.
- RFM analysis segments customers based on Recency , Frequency , and Monetary to understand customer behavior and optimize targeted marketing.

Online Retail Data Pre-Processing

- Converts the InvoiceDate column to proper datetime format for accurate date calculations.
- Ensures InvoiceNo is treated as a string to preserve leading zeros and enable string operations
- Identify and remove a significant number of outliers from CustomerID
- Calculating RFM: Compute Recency (days since last purchase), Frequency (total transactions), and Monetary (total spend) per customer
- 4,300 customer transactions were identified with their recency, frequency, and monetary values
- Log transformation to reduce skewness in the data.
- Use Min-Max Scaling on the RFM dataframe to normalize the data

What is Spectral Clustering?

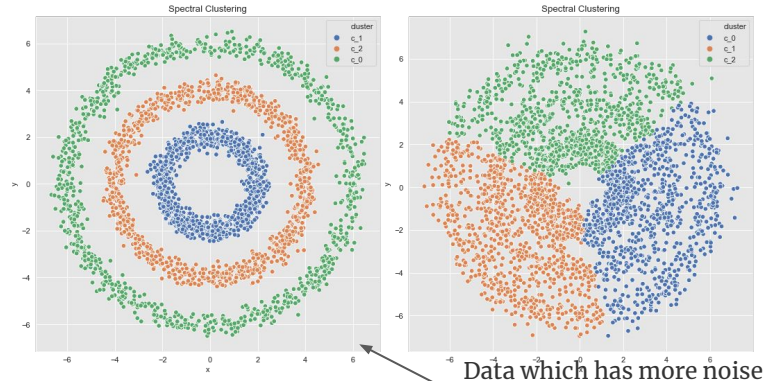
- Graph-based Clustering

- Procedures: data $S = \{s_1, s_2, \dots, s_i, s_n\}$

- Create Adjacency matrix (how close two points in space) $A_{i,j} \approx \exp(-\alpha \|s_i - s_j\|^2)$
- Compute normalized Laplacian Matrix (L)

- Compute first k eigenvectors of L and stack them w.r.t k smallest eigenvalues

$$M = [\mathbf{v}_1 \mathbf{v}_2 \dots \mathbf{v}_k] \in \mathbb{R}^{n \times k}$$



$$L = \begin{bmatrix} L_1 & & 0 \\ & L_2 & \\ 0 & & L_3 \end{bmatrix}$$

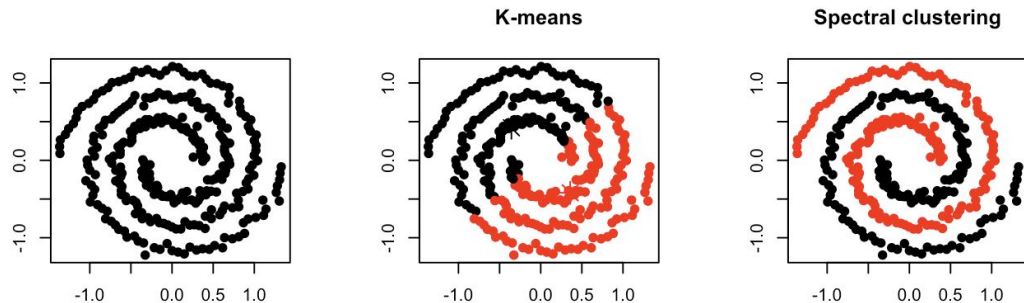


Figure: K-means algorithm uses the compactness and Spectral Clustering uses Connectivity approach.

What is Spectral Clustering?

- Graph-based Clustering
- Procedures: data $S = \{s_1, s_2, \dots, s_i, s_n\}$

- Create Adjacency matrix (how close two points in space) $A_{i,j} \approx \exp(-\alpha \|s_i - s_j\|^2)$
- Compute normalized Laplacian Matrix (L)
- Compute first k eigenvectors of L and stack them w.r.t k smallest eigenvalues
- Let \mathbf{y}_i be the vector corresponding to the i-th row of M $\mathbf{y}_i = (v_{i1}, v_{i2}, \dots, v_{ik}) \in \mathbb{R}^k$
- Cluster the points $y_i \in \mathbb{R}^k$ with the k-means algorithm into clusters C_1, \dots, C_k
- Assign points s_i according to the cluster

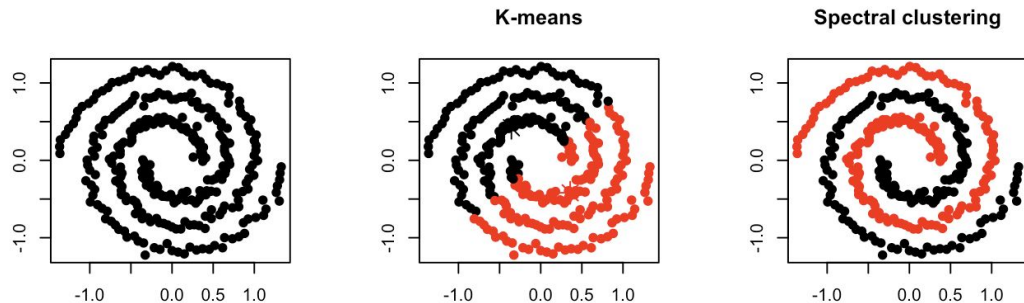
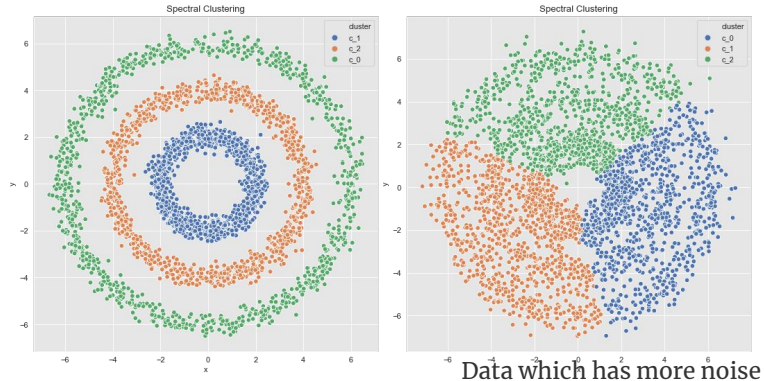
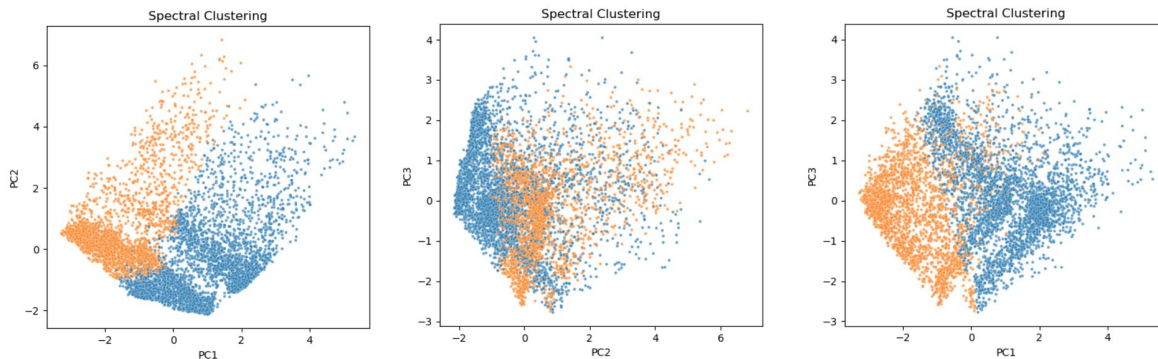


Figure: K-means algorithm uses the compactness and Spectral Clustering uses Connectivity approach.

Spectral Clustering: eBay Shill Bidding Dataset (n=2)



Spectral clustering can be sensitive to noise and outliers.



Spectral Clustering: Online Retail Dataset (n=4)

🍊 VIP Orange Members

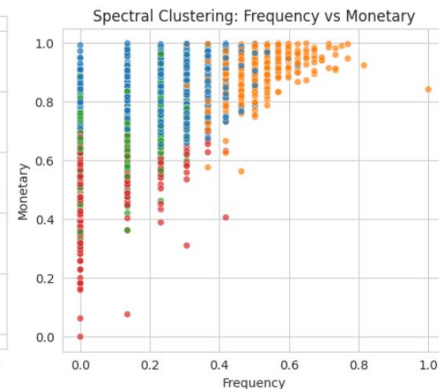
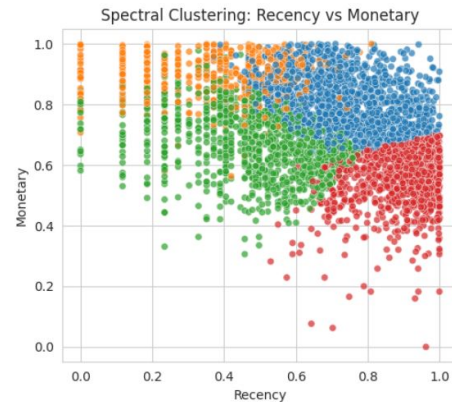
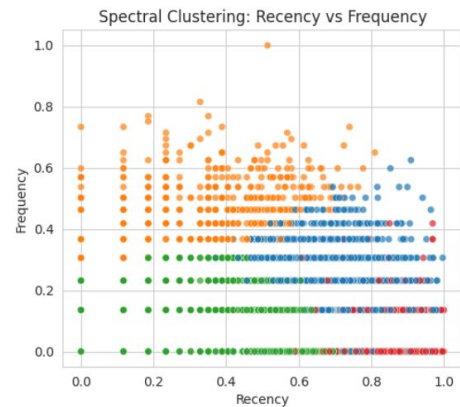
- **Always Shopping:** Active any time $[0, 0.9]$
- **Luxury Lovers:** Frequently buy expensive items $[0.8, 1]$

🔴 Basic Red Members

- **Recent Shoppers:** Active only lately $[0.6, 1]$
- **Infrequent & Economical:** Prefer inexpensive products $[0, 0.6]$

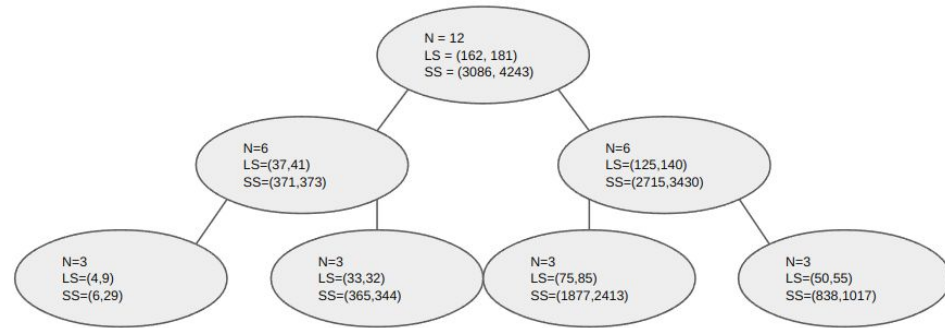
🟡 Blue Members & 🟢 Green Members

- **More Expensive:** Blue buys pricier items than Green
 - a. 🟡 $[0.6, 1]$
 - b. 🟢 $[0.3, 0.8]$
- **More Recent:** Blue purchase more recent than Green
 - a. 🟡 $[0.5, 1]$
 - b. 🟢 $[0, 0.8]$



What is BIRCH?

- **B**alanced **I**terative **R**educing and **C**lustering using **H**ierarchies
- Problem: Your dataset is too big, and other clustering algorithms do not scale well. How can you proceed?
- Solution: Create a “summary” of the data called a clustering feature tree. This tree stores information about the clusters without storing the data itself. **Works as standalone algorithm or results can be input to another clustering method!**
- Limitations:
 - Possibility of catastrophic cancellation
- BETULA - a 2022 update to the algorithm.



How does BIRCH work?

Hyperparameters:

- Threshold - affects the maximum radius of a cluster. Set lower to create additional, more compact clusters.
- Branching Factor - The maximum number of subclusters to which an internal node can point.
- Number Of Clusters (sklearn implementation, optional) - If an int - the BIRCH model's output is fed into AgglomerativeClustering using n as this number. Can also pass in any clustering model object.

Data Structure: Build a clustering feature (height balanced) tree by processing each row in the data. All nodes represent a cluster. Leaf nodes hold the following information: Number of samples in cluster, Linear Sum, Squared Sum, Centroid (in sklearn), Squared Norm of centroid (in sklearn). Non leaf nodes are an array of pointers to sub clusters. They hold data equal to the sum of its subclusters.

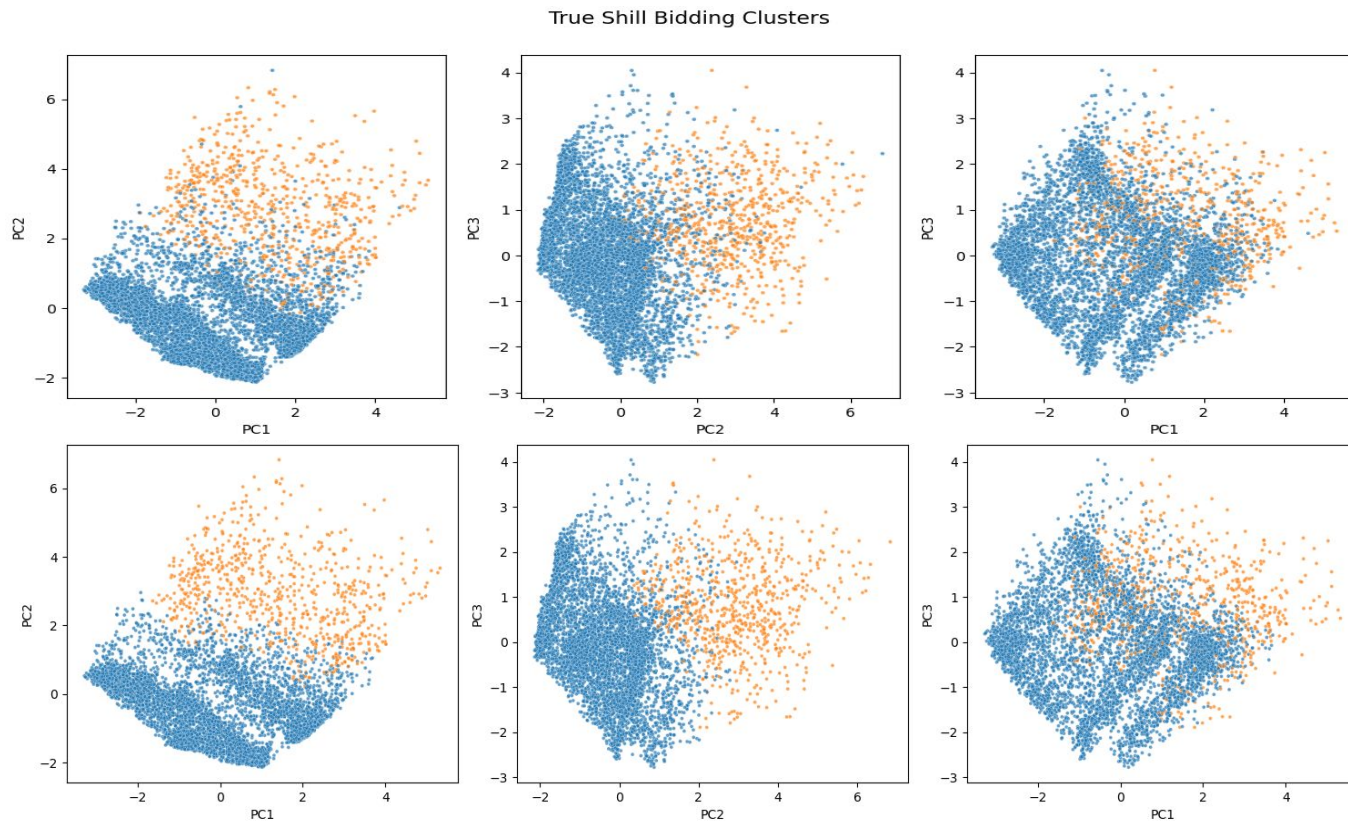
Algorithm:

1. For each instance in the dataset:
 - a. Merge new sample into tree root subcluster such that the resulting clusters has the smallest radius. Move down the trees child nodes continuing the same process until a leaf node is hit. Update subcluster metrics to account for new datapoint at leaf node and all its parents.
 - b. While updating a leaf, if the cluster radius now exceeds the threshold, create a new leaf node.
 - c. If a new leaf node is created such that the parent node's number of subclusters exceeds the branching factor, then split the parent node and redistribute the clusters. Note that this can result in additional splits in the parent nodes.
2. (Optionally) Prune the leaves of the tree to reduce outliers and group clusters.
3. (Optionally) Pass results into another clustering algorithm, such as AgglomerativeClustering.

BIRCH Results - Shill Bidding

Hyperparameters used:

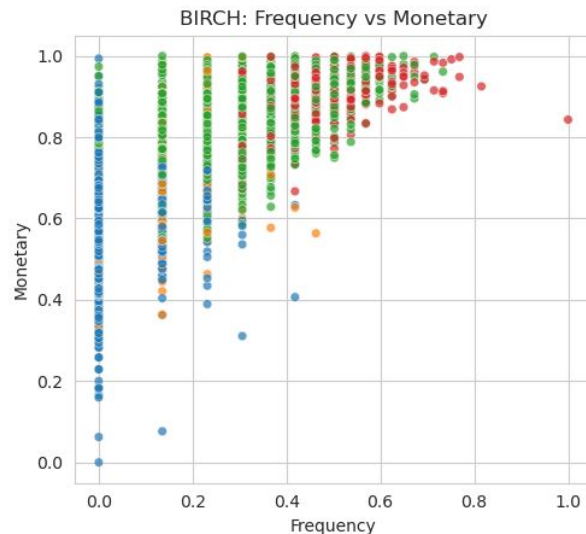
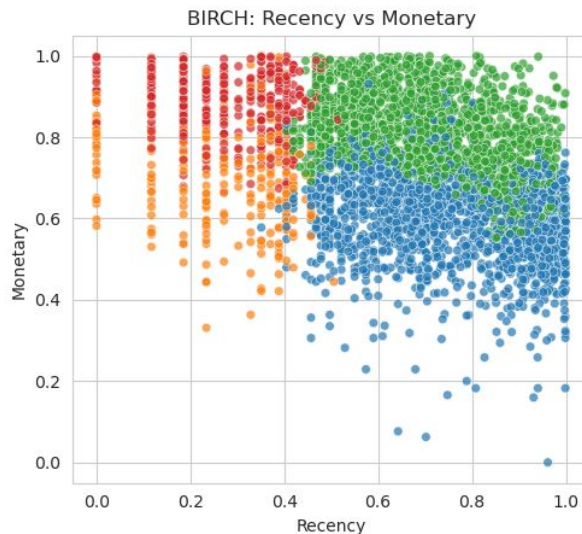
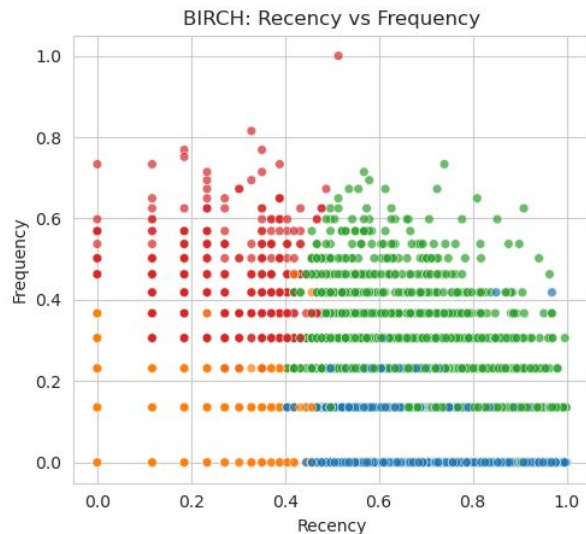
- Threshold = 0.4
- Branching Factor = 45
- Number of Clusters = 2



BIRCH Results - Online Retail

Hyperparameters used:

- Threshold = 0.1
- Branching Factor = 43
- Number of Clusters = 4



What is DBSCAN? (Density Based Spatial Clustering of Applications with Noise)

Density-based clustering algorithm that groups data points into clusters based on their proximity

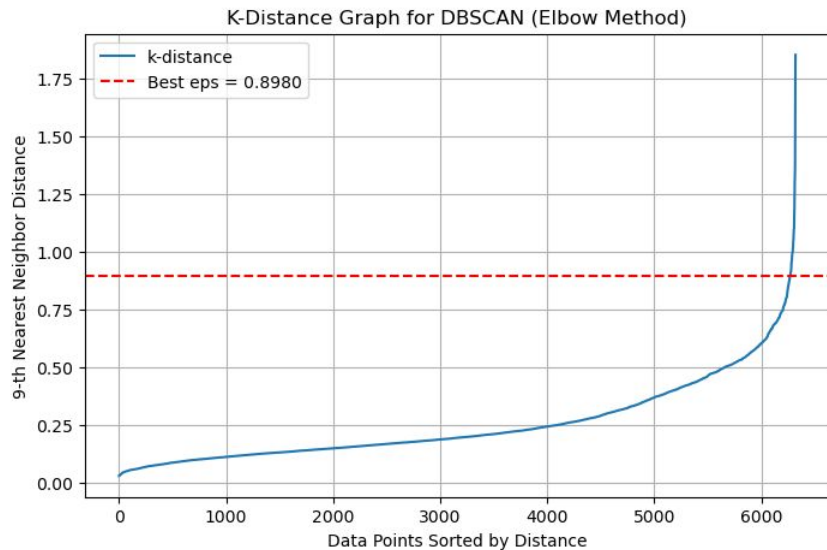
Identifies arbitrarily shaped clusters and **detects noise** (outliers)

Two key hyperparameters:

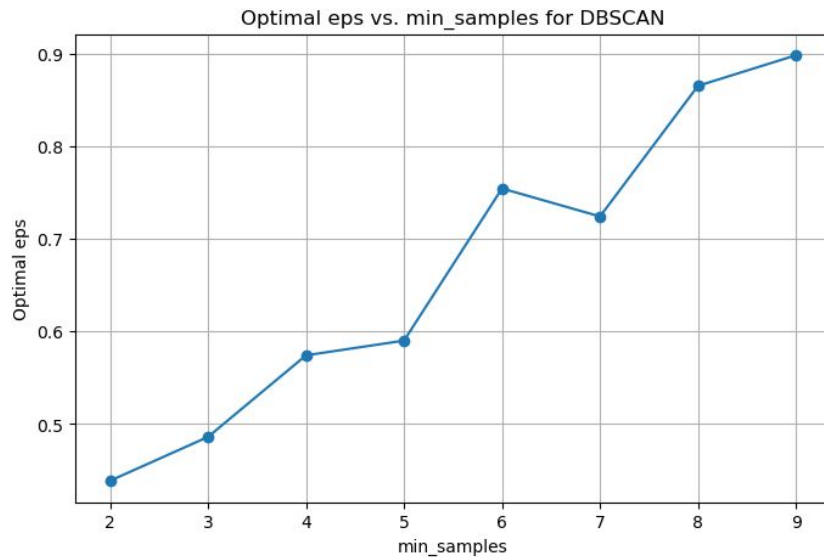
1. **eps (epsilon, neighborhood radius)**: Defines the maximum distance between two points for them to be considered part of the same cluster.
2. **min_samples**: The minimum number of points required to form a **dense region** (cluster core).

DBSCAN setting parameters for Shill bidding dataset

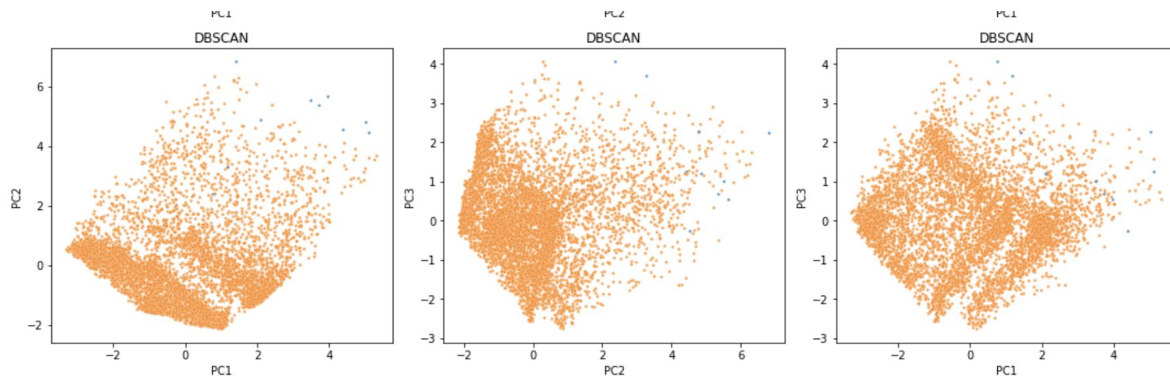
Use elbow method to determine optimal epsilon
(eps) f: 0.8979802091362066



Optimal min_samples = 9



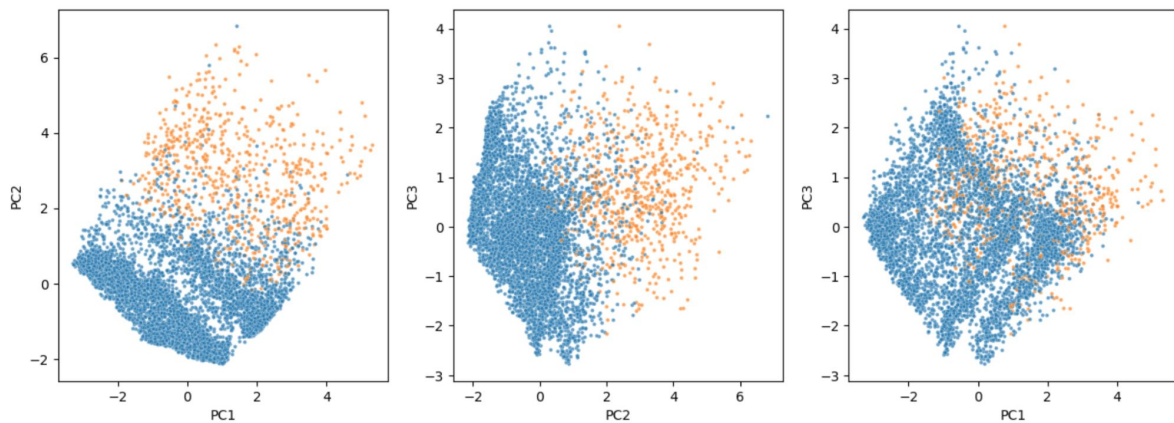
DBSCAN: eBay Shill Bidding Dataset (n=2)



Cluster Counts:

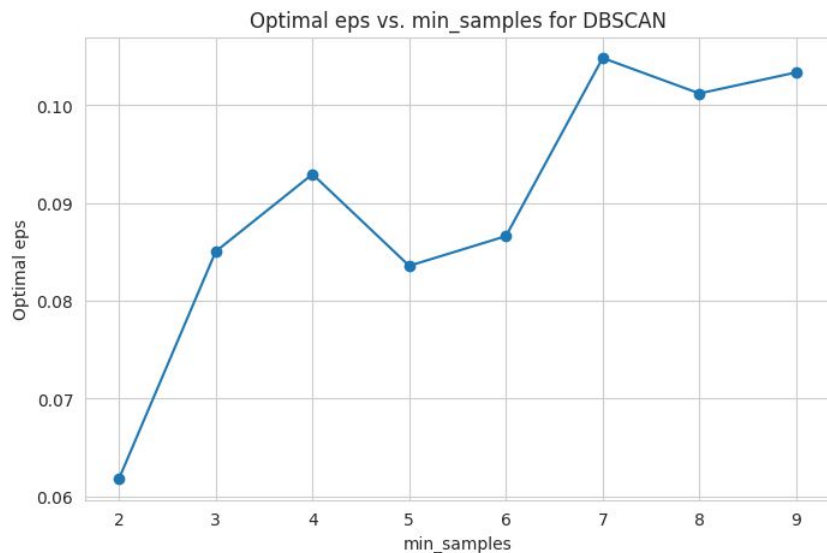
0	5157
-1	1048
5	31
4	24
3	23
1	21
2	17

True Shill Bidding Clusters

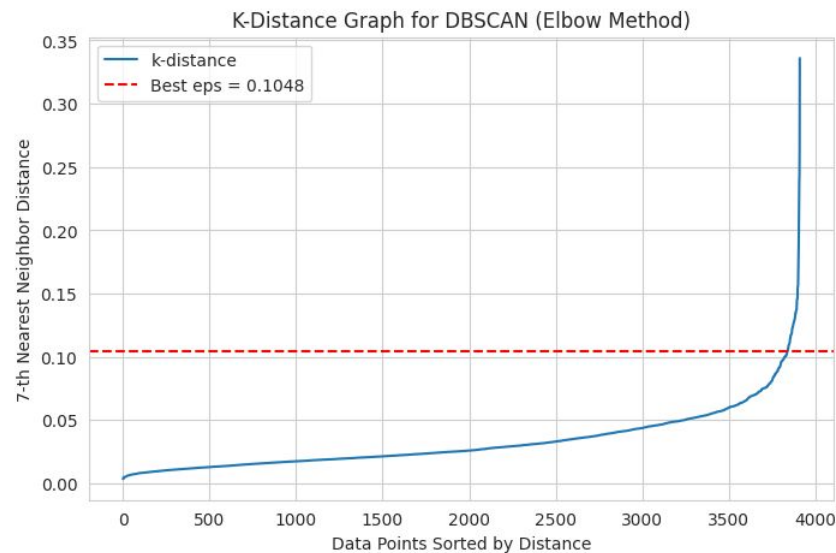


DBSCAN Parameters Adjustments for Online Retail dataset

Best min_samples = 7



Best eps = 0.10



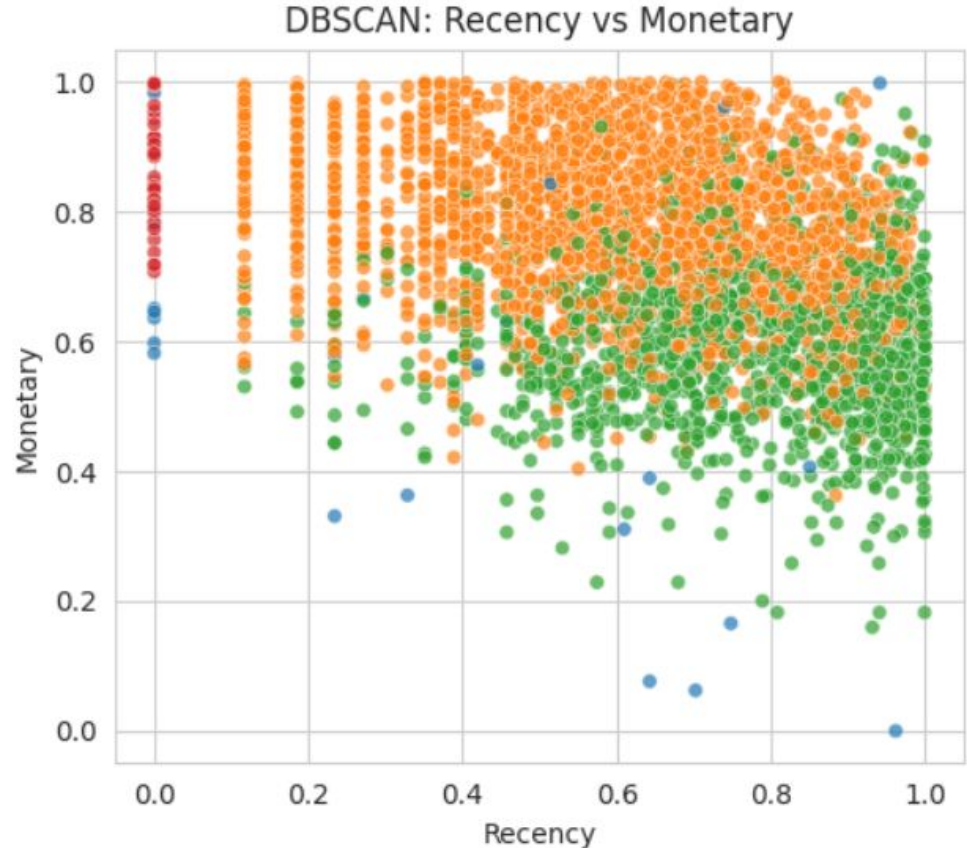
DBSCAN Online retail dataset

Silhouette Score: 0.23 (weak cluster separation)

Calinski-Harabasz Index (CHI): 814 (suggests dense but overlapping clusters)

Observations:

- DBSCAN detected 4 clusters and labeled outliers
- Significant overlap between clusters (orange & green) reduces segmentation quality.
- Not be the best fit for this dataset



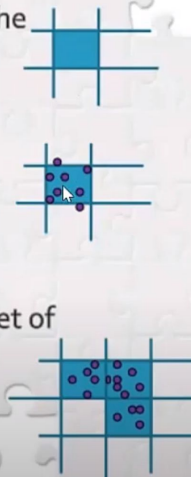
What is CLIQUE?

- Density-based and grid-based subspace clustering algorithm
 - Clustering starts at single dimension subspace and move upwards towards higher dimension
- Tunable Parameters:
 - Intervals: number of equal parts each dimension of the data space is divided into
 - Threshold: minimum number of points a cell must contain to be considered "dense".

✱ Unit : After forming a grid structure on the space, each rectangular cell is called a Unit.

✱ Dense : A unit is dense, if the fraction of total data points contained in the unit exceeds the input model parameter.

✱ Cluster : A cluster is defined as a maximal set of connected dense units.



CLIQUE

Bottom-up approach

Dense C1 = (Age= 30-40 , Salary= 30k-40k)

Dense C2 = (Vacation=2-3 Weeks , Salary= 30k-40k)

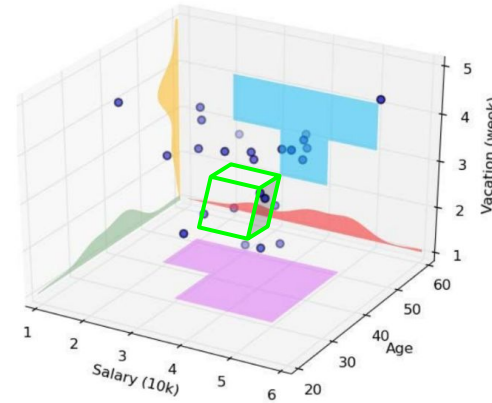
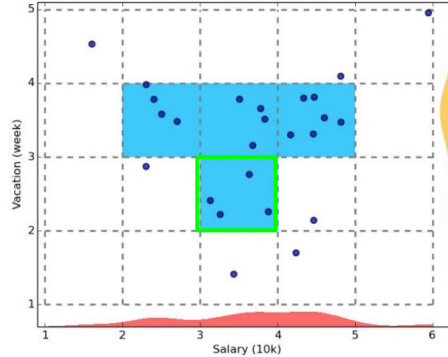
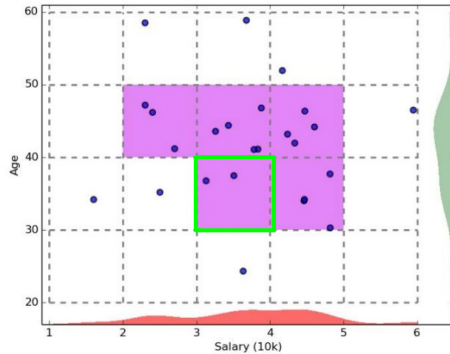
Intersection:

C1 and C2 is different in one dimension

C:C1 intersection C2 = (Age= 30-40 , Salary= 30k-40k, Vacation=2-3 Weeks)

Retention:

Retain C if the selectivity is greater than the threshold else discard it

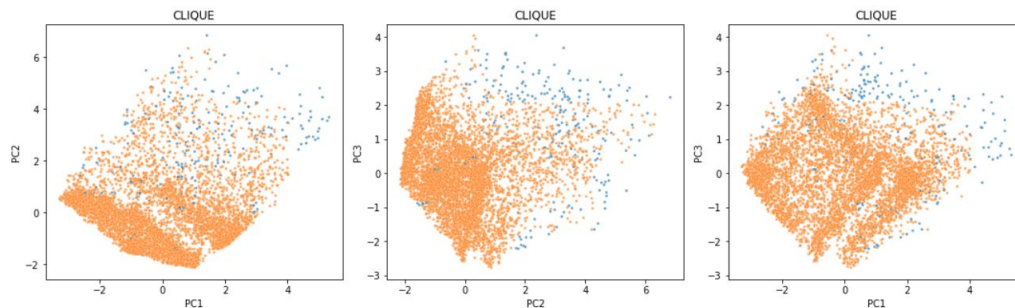


Apriori principle: *If a collection of points S is a cluster in a k -dimensional space, then S is also part of a cluster in any $(k-1)$ dimensional projections of this space*

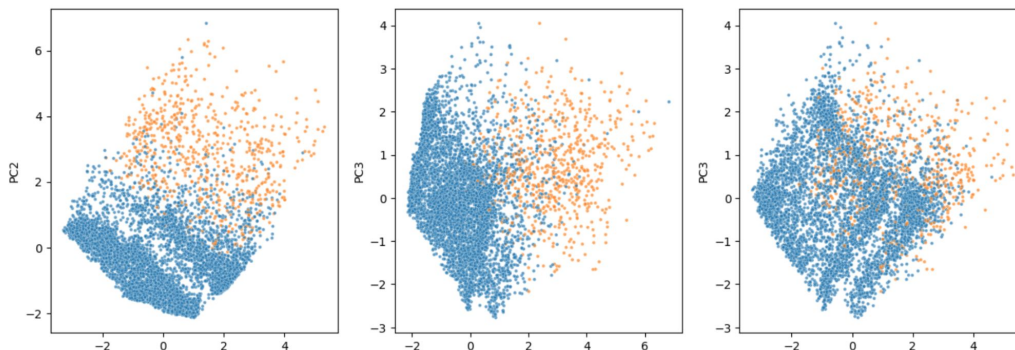
CLIQUE: eBay Shill Bidding Dataset

Intervals = 7
Threshold = 8

2 groups found.



True Shill Bidding Clusters

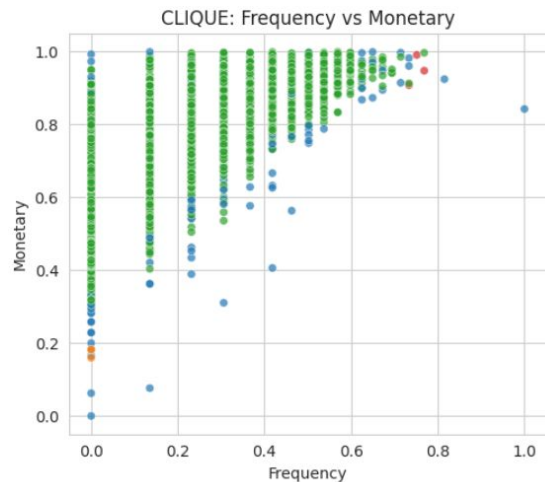
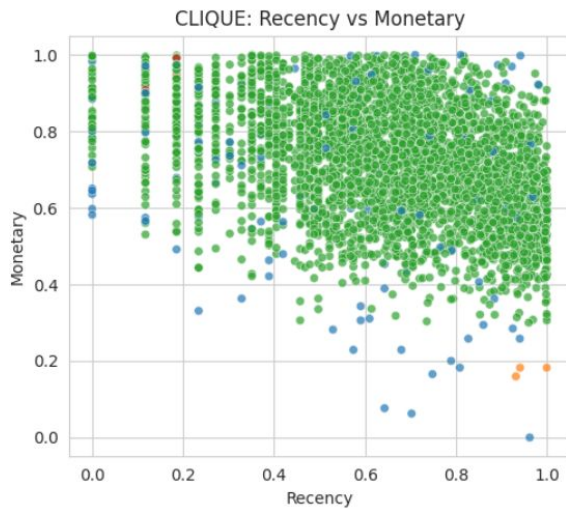
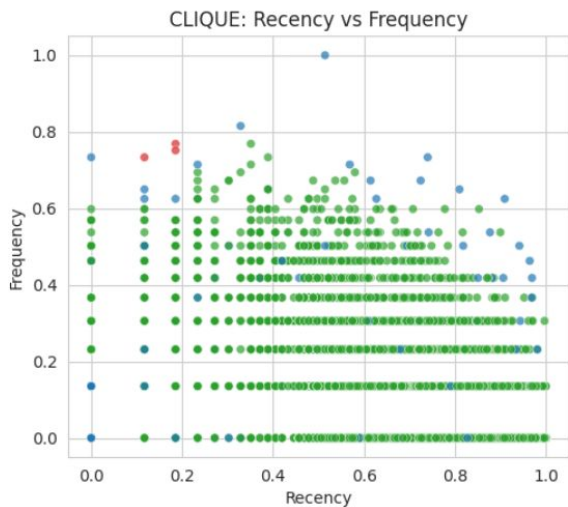


CLIQUE: Online Retail Dataset

Intervals = 10
Threshold = 2

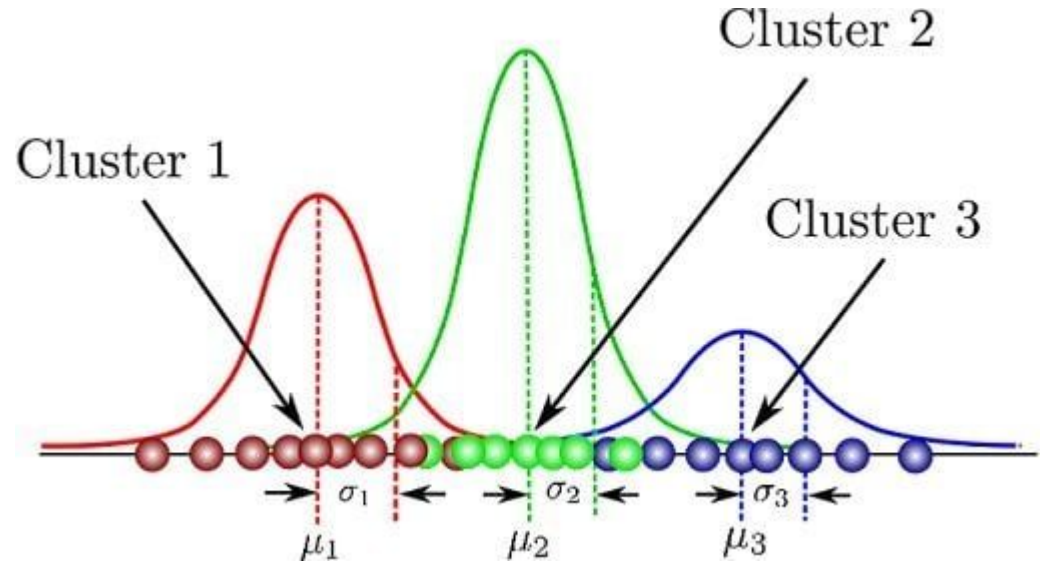
4 groups found. But too much noise.

Sparse Population in Cells in two dimensions but dense population in cells in recency and monetary dimension



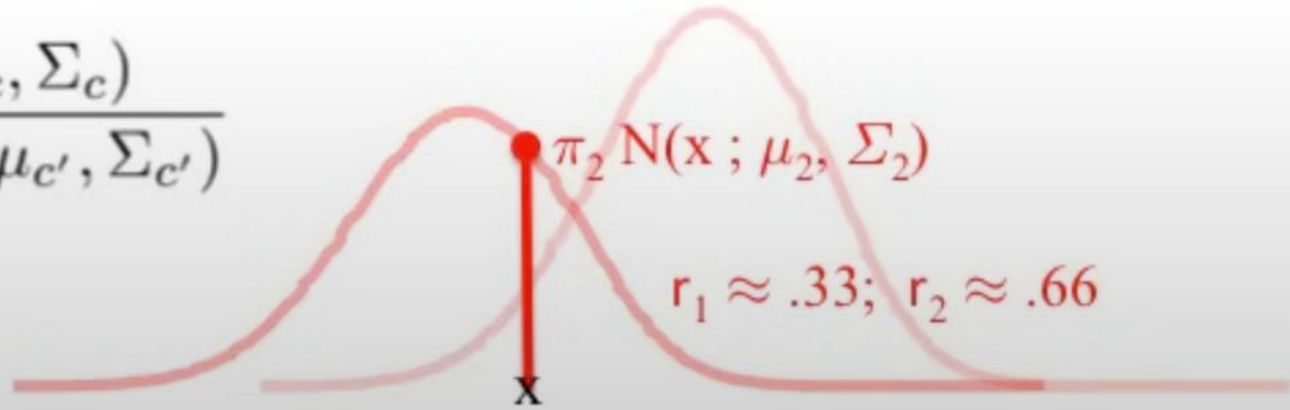
What is Gaussian Mixture Model?

- Assume clusters are Gaussian Distribution.
- Initialization -> Expectation -> Maximization -> Iterate until convergence.
- Initialization: # of clusters, initial clusters' means and stds, threshold for convergence.
- Included in SKLearn.
- Good for large and overlapping data.



What is Gaussian Mixture Model?

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i ; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i ; \mu_{c'}, \Sigma_{c'})}$$

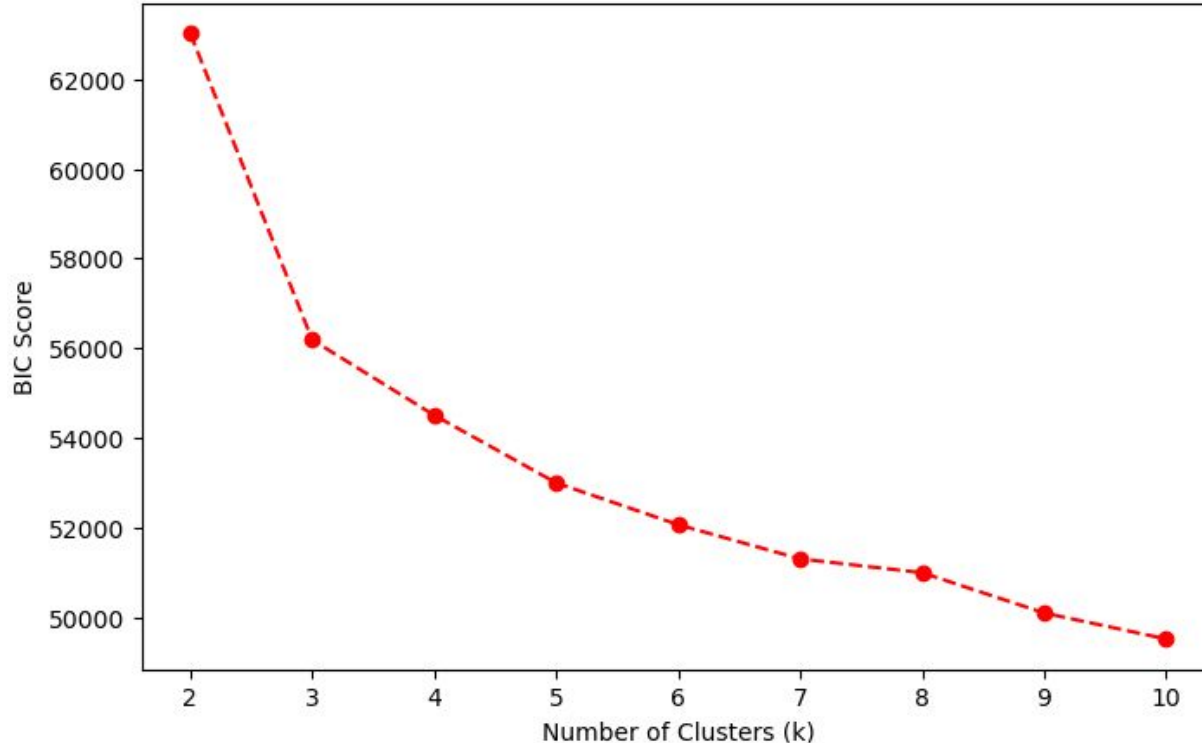


- Expectation step: Determine clusters for each x (r_c is prob. of belonging to c cluster).
- Maximization step: Update mean, std and pdf of clusters.
- Convergence when log-likelihood is stable.

$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \mathcal{N}(x_i ; \mu_c, \Sigma_c) \right]$$

Gaussian Mixture Model Results: Shill Bidding

Optimal Number of Clusters for GMM (BIC)



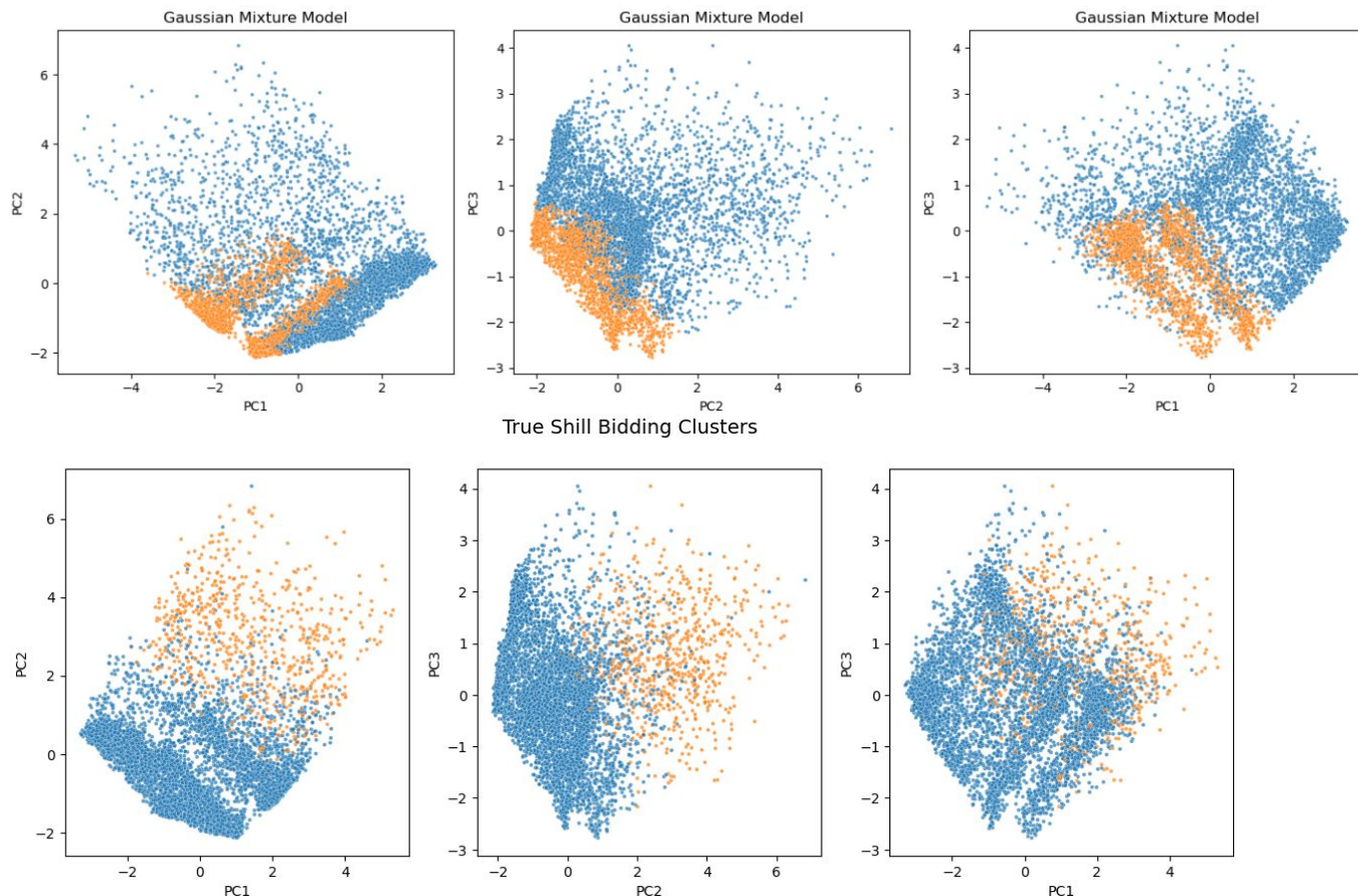
Bayesian Information
Criterion

$$\text{BIC} = k \ln(n) - 2 \ln(\hat{L})$$

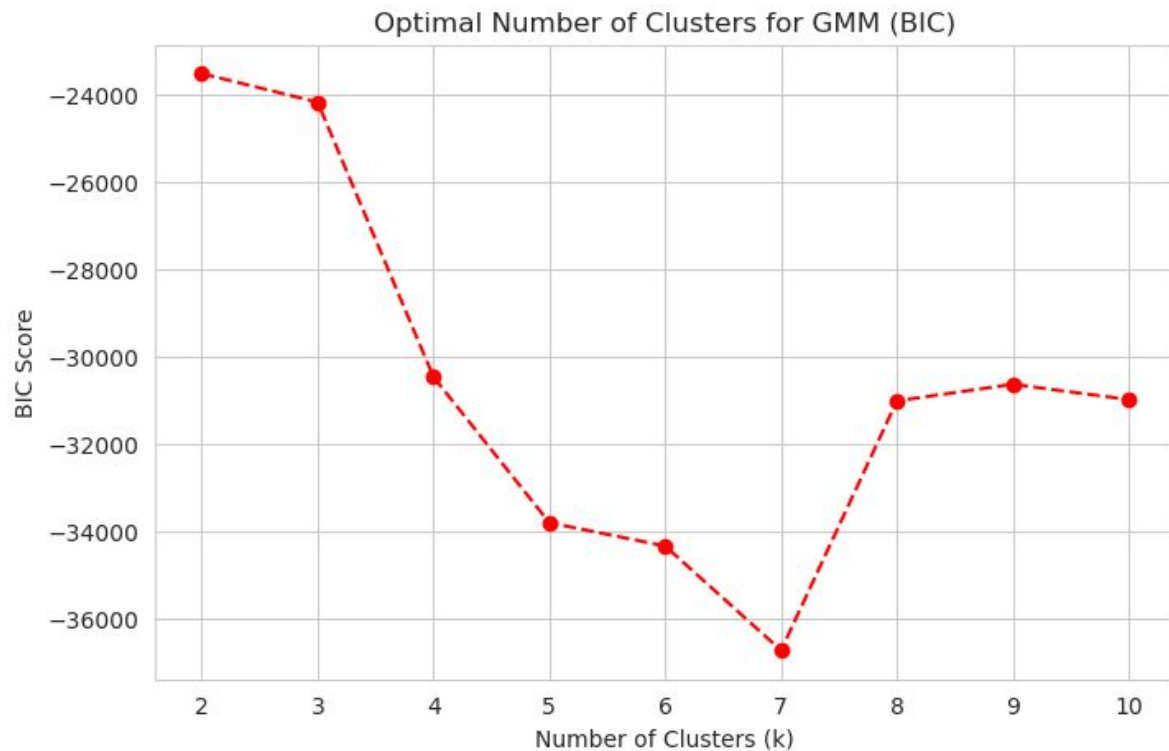
Smaller better, but no
minimum

Use # of clusters = 2

Gaussian Mixture Model Results: Shill Bidding (n=2)



Gaussian Mixture Model Results: Online Retail



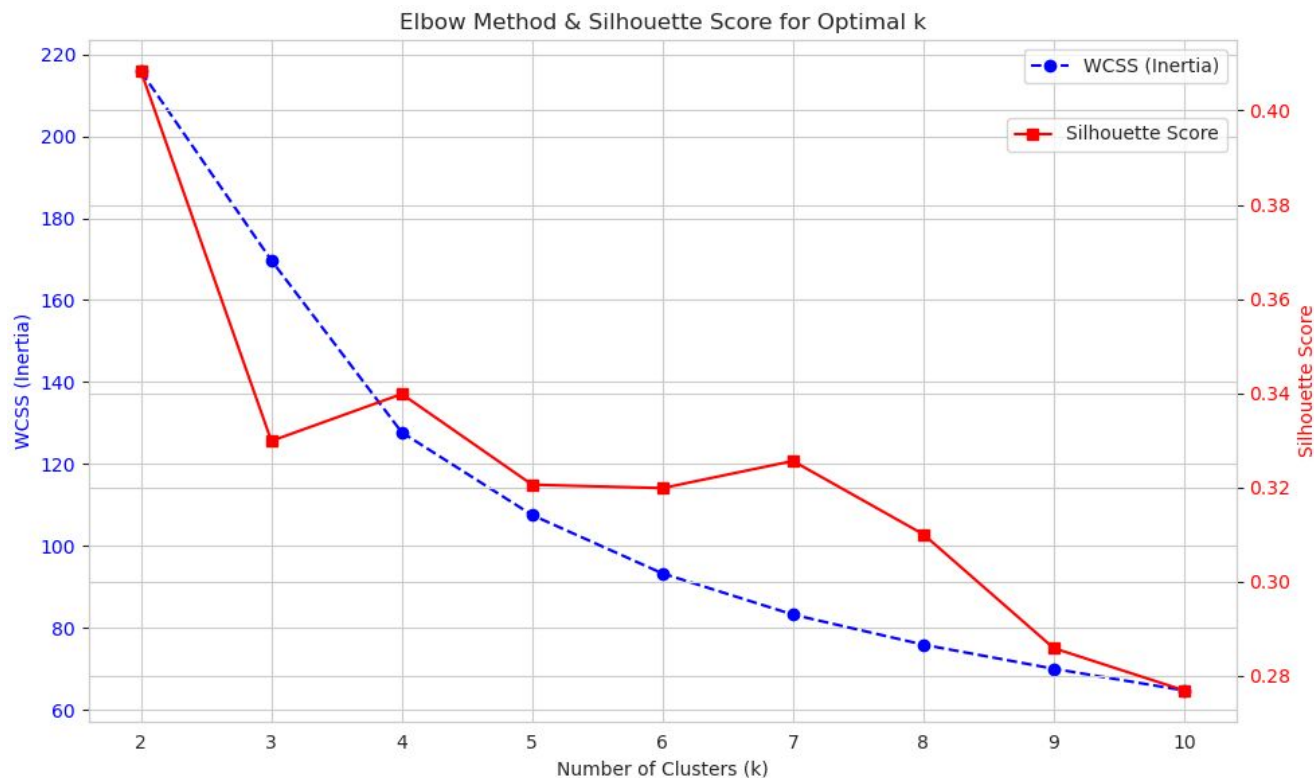
Bayesian Information
Criterion

$$\text{BIC} = k \ln(n) - 2 \ln(\hat{L})$$

Smaller the better

Possible # of clusters: 7

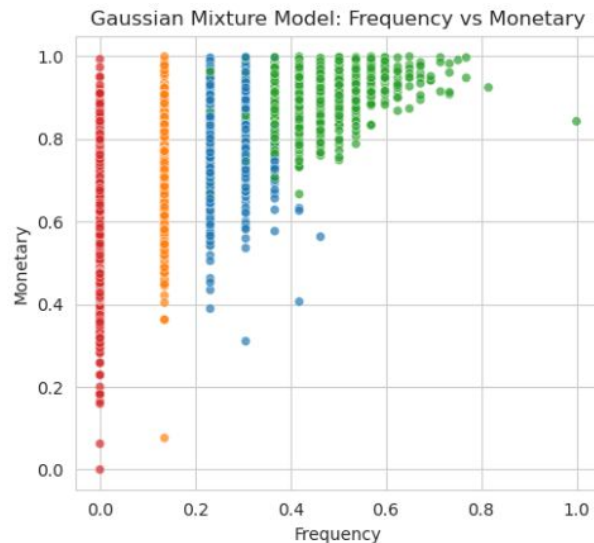
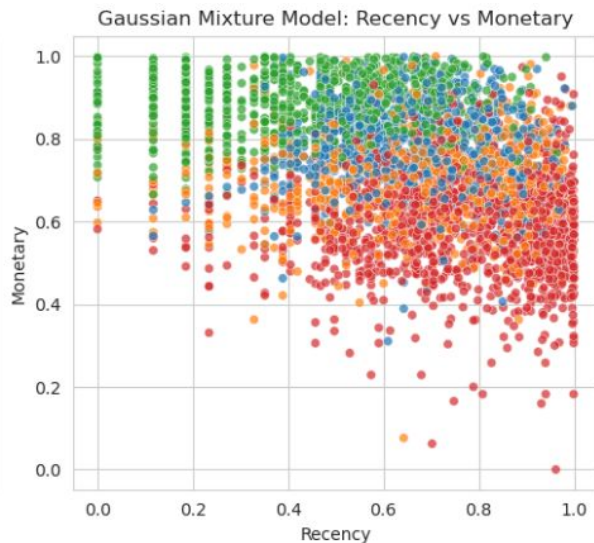
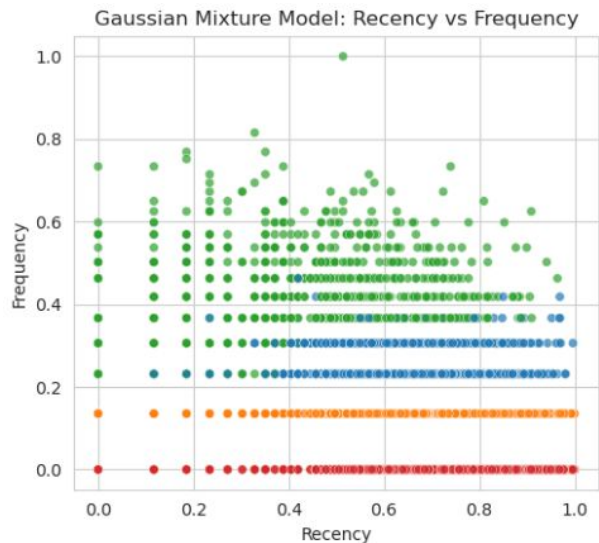
Gaussian Mixture Model Results: Online Retail



Possible # of clusters: 4

BIC for GMM and WCSS&Silhouette score for k-means do not match

Gaussian Mixture Model Results: Online Retail



Works poorly for recency vs monetary









Useful links for Gaussian Mixture Model

- <https://builtin.com/articles/gaussian-mixture-model>
- https://akireeva.com/GMM_visually_explained
- https://www.youtube.com/watch?v=qMTuMa86NzU&ab_channel=AlexanderIhler
- <https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture>

Shill Bidding model scores (n=2)

 Best

 Worst







	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index	Adjusted-Rand Score
Spectral Clustering	0.330729	1.288741	 2857.991950	 0.005217
BIRCH	0.391242	1.111091	1985.032342	 0.773850
DBSCAN	 0.493524	 0.753521	 58.219351	0.020483
Gaussian Mixture Model	 0.181679	1.307371	1818.371861	0.016704
CLIQUE	0.336017	 1.551502	400.060088	0.237312

- Silhouette Score: How well data is clustered. Larger the better (-1~1).
- Davies-Bouldin Index: How well data is clustered. Smaller the better.
- Calinski-Harabasz Index: Clusters are well separated. Larger the better.
- Adjusted-Rnd Score: predicted cluster vs true cluster

Online retail model validation scores(n=4)

 Best

 Worst

	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
Spectral Clustering	 0.275979	 1.093322	 2160.668231
BIRCH	0.254163	1.126216	1740.743758
DBSCAN	0.239561	1.744273	814.233085
Gaussian Mixture Model	0.165016	2.033186	1464.965495
CLIQUE	 0.135205	 2.232572	 22.406536

- Silhouette Score: How well data is clustered. Larger the better (-1~1).
- Davies-Bouldin Index: How well data is clustered. Smaller the better.
- Calinski-Harabasz Index: Clusters are well separated. Larger the better.
- Adjusted-Rnd Score: predicted cluster vs true cluster

Which one works better?

- Shill Bidding: BIRCH works the best.
- Online Retail: Spectral clustering works the best, BIRCH the second.
- Possible explanations for each results:
 - Spectral clustering: Better with more connected dataset (Online Retail)
 - BIRCH: Results more likely to have come from Hierarchical Clustering step using Ward linkage, which is useful for noisy data. (Shill Bidding)
 - DBSCAN: Failed to due to varying densities and overlapping clusters.
 - CLIQUE: Better with high-dimensional and data with varying densities (Shill Bidding has more density variation)
 - Gaussian Mixture Model: Better with large, overlapping, complex datasets, and sensitive to the initial condition.