COMP 4432 Machine Learning

Lesson 8: Unsupervised Learning

Agenda

- Unsupervised Learning
  - Intro and comparison
- Outlier Detection
- Clustering
  - Importance of EDA
  - Methods
  - Evaluation

Unsupervised vs. Supervised

- Supervised models include targets
  - Continuous values in regression
    - Price of house, Diabetes progression, Count of bike rentals
  - Group labels in classification
    - Survival aboard Titanic

Unsupervised vs. Supervised

- Evaluation of model quality
  - Supervised
    - RMSE, R2
    - Log-Loss, Confusion Matrix, AUC
  - Unsupervised
    - ?

Isolation Forest

- Unsupervised method used for outlier detection
  - Effective with high dimensional data

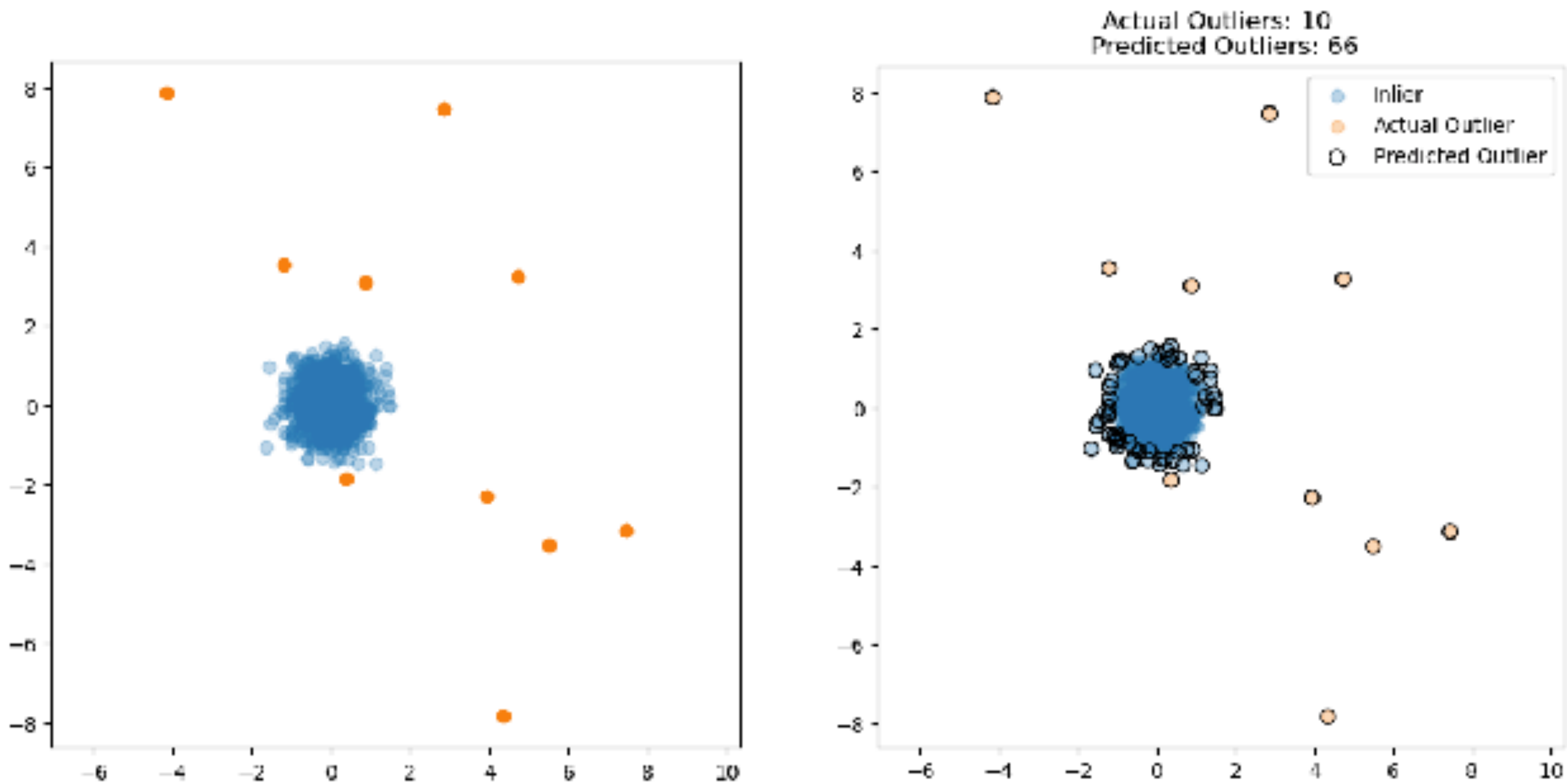
Isolation Forest

- Unsupervised method used for outlier detection
  - Effective with high dimensional data
- Constructs if-then-else logic, but completely random
  - Randomly select a feature/dimension/axis
  - Randomly select a cut point between the minimum and maximum values of selected feature
  - Iterates until all instances are isolated

Isolation Forest

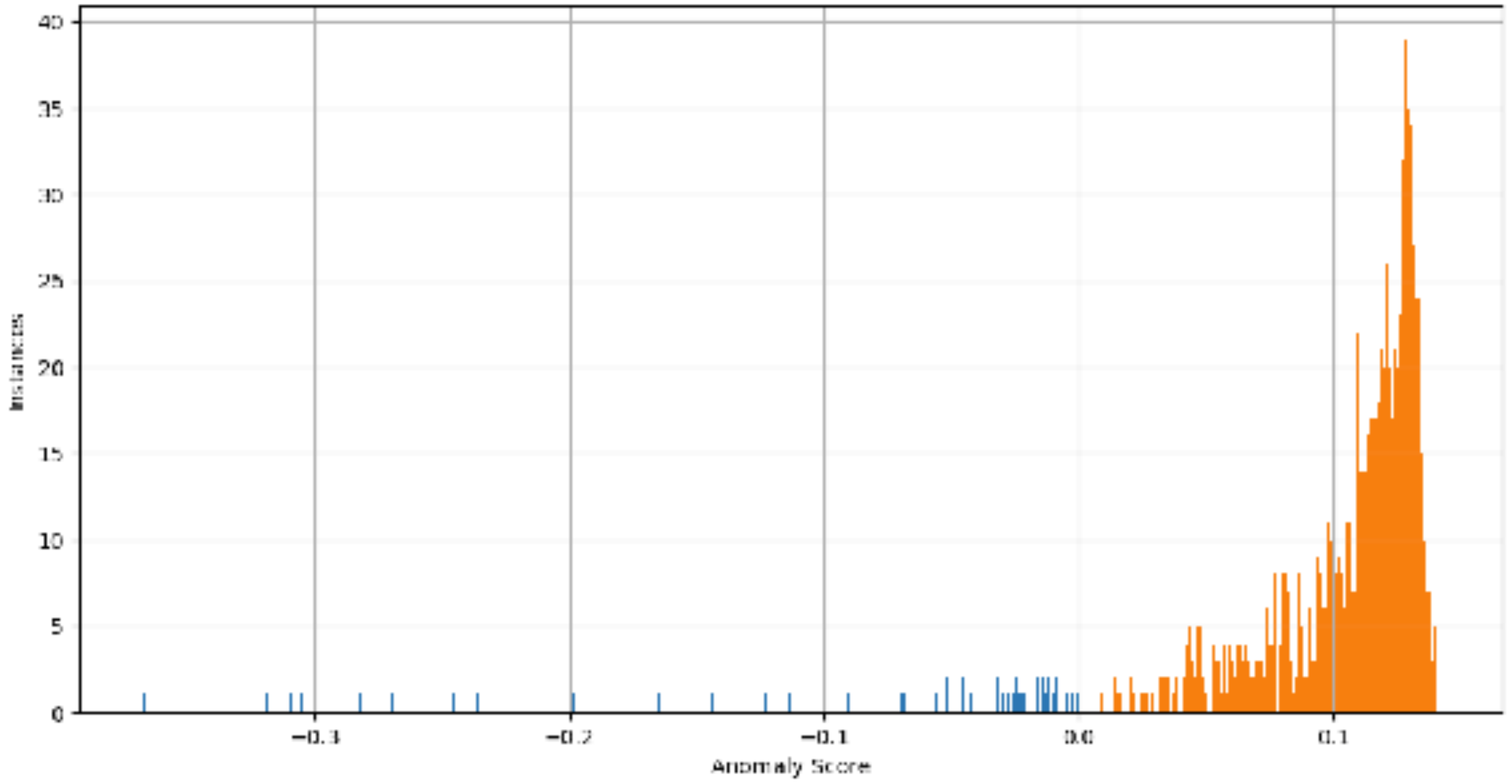
- Unsupervised method used for outlier detection
  - Effective with high dimensional data
- Constructs if-then-else logic, but completely random
  - Randomly select a feature/dimension/axis
  - Randomly select a cut point between the minimum and maximum values of selected feature
  - Iterates until all instances are isolated
- Outliers are far from other data, so on average, they get isolated in fewer iterations (shorter path length)

Isolation Forest



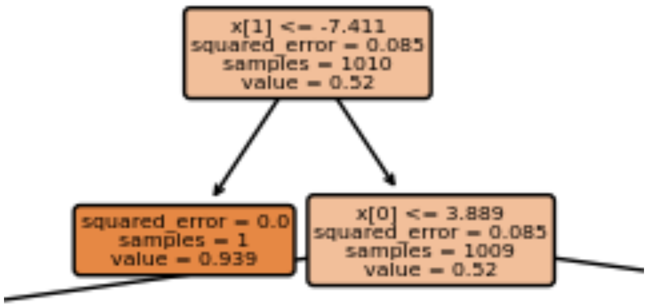
Isolation Forest

- [SciKit Documentation]()
- Contamination
  - *“the proportion of outliers in the data set”*
  - Not necessary known beforehand

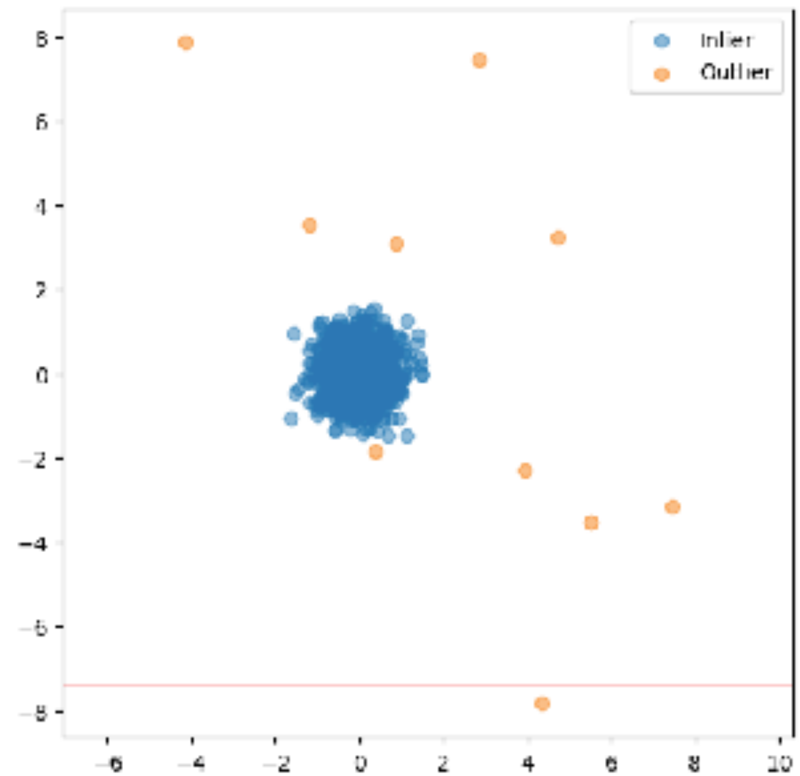


Isolation Forest

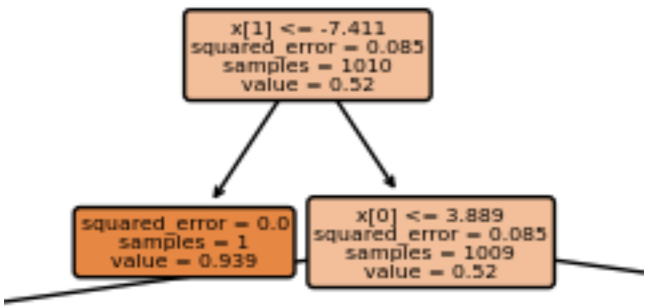
Isolation Forest

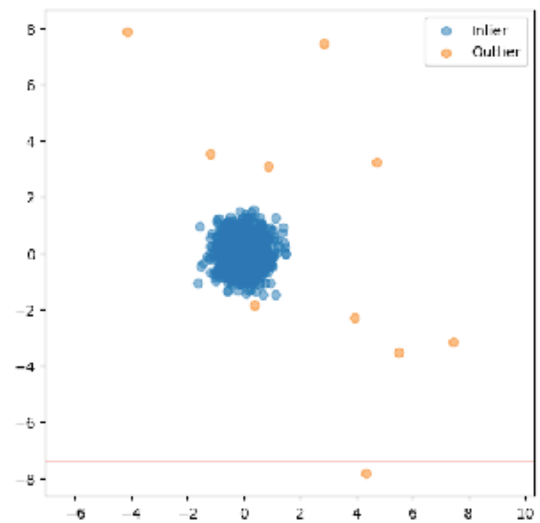


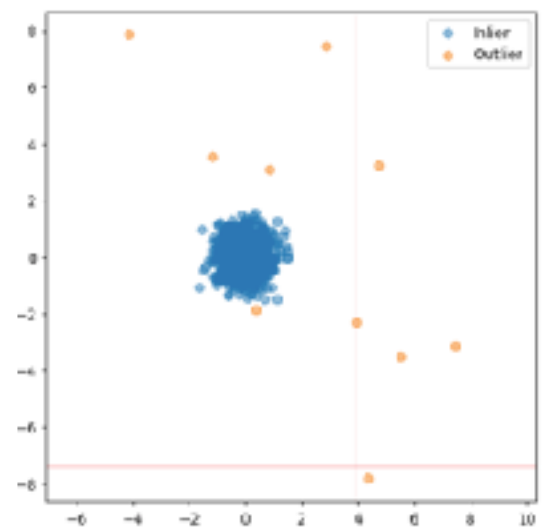
Depth = 2



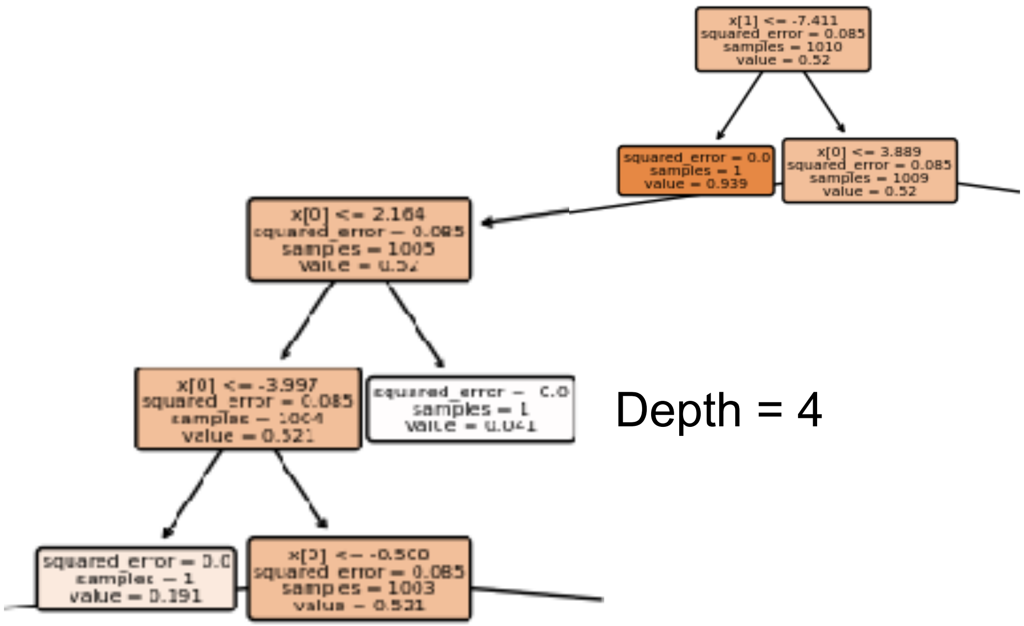
Isolation Forest





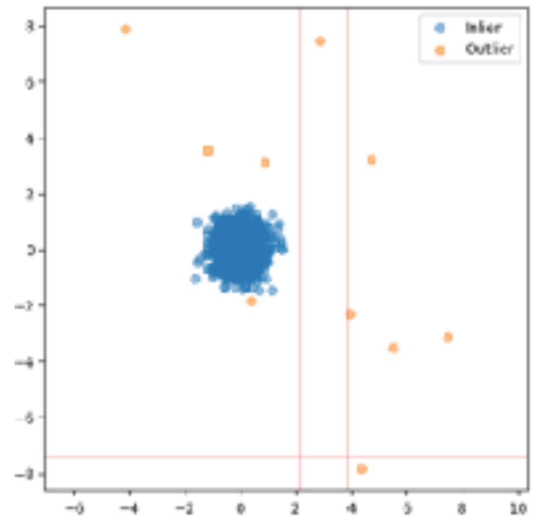


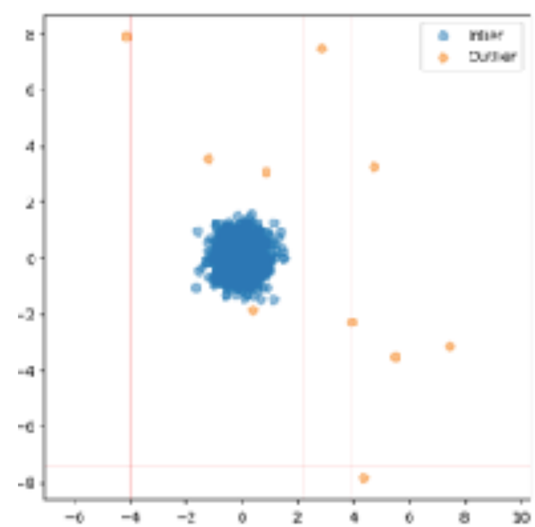
Isolation Forest



Depth = 4

Depth = 5





Isolation Forest

Clustering

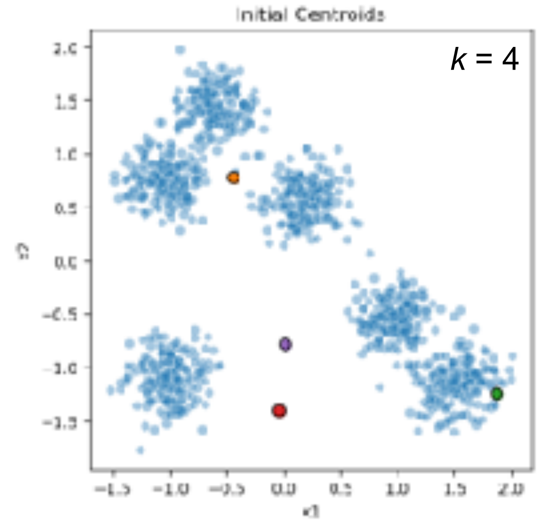
- Partition *n* instances of data into *k* groups
  - *k* is less than or equal to *n*
- The correct *answer* isn’t know beforehand
- Multiple algorithms
  - K-Means
  - DBScan
  - Hierarchal

K-Means Clustering

- Algorithm workflow

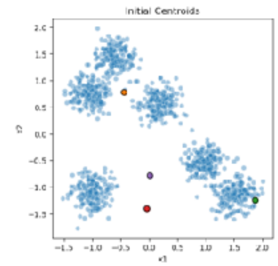
K-Means Clustering

- Algorithm workflow
  - Selection of initial centroids
    - Random versus KMeans++

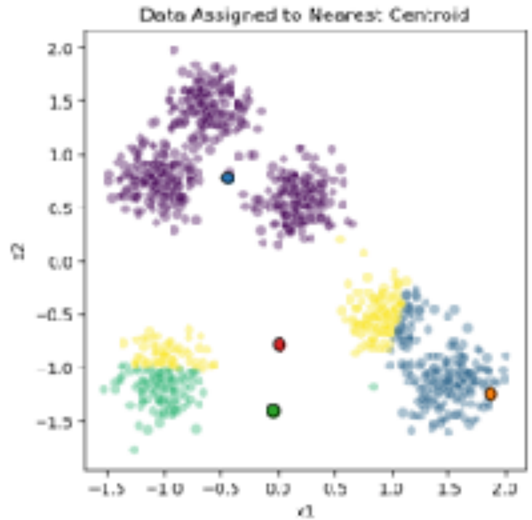


K-Means Clustering

- Algorithm workflow
  - Selection of initial centroids
    - Random versus KMeans++
  - Update
    - Data points are assigned to nearest centroid

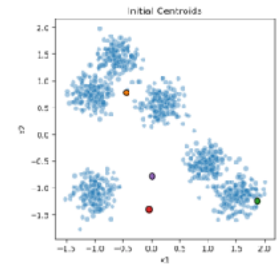


k = 4

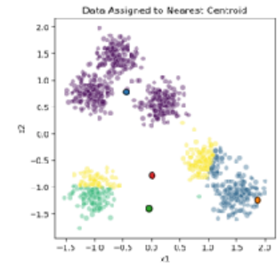


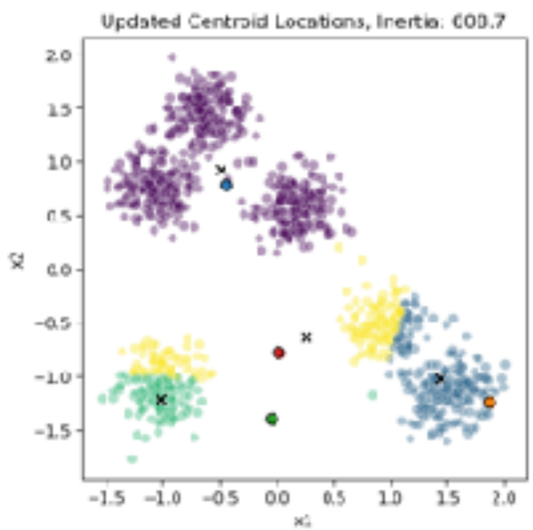
K-Means Clustering

- Algorithm workflow
  - Selection of initial centroids
    - Random versus KMeans++
  - Update
    - Data points are assigned to nearest centroid
    - Average position amongst each group is calculated



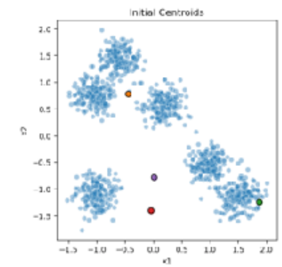
k = 4



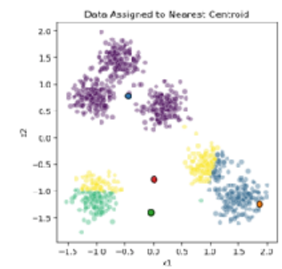


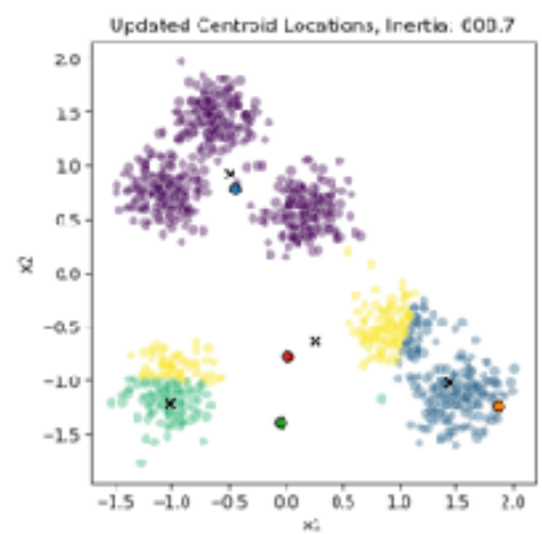
K-Means Clustering

- Algorithm workflow
  - Selection of initial centroids
    - Random versus KMeans++
  - Update
    - Data points are assigned to nearest centroid
    - Average position amongst each group is calculated
    - Each centroid is updated to the group's average value
  - Update again… and again…
- [SciKit KMeans Documentation]

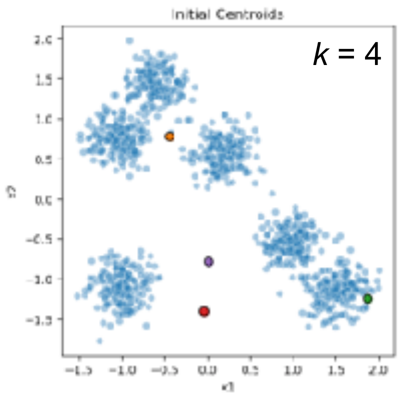


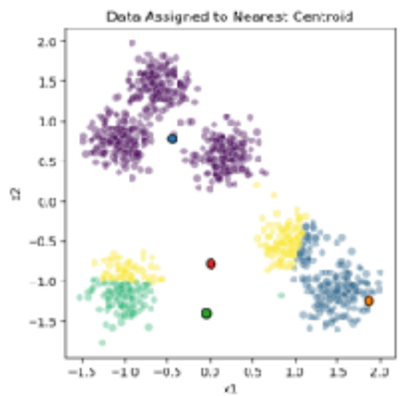
k = 4

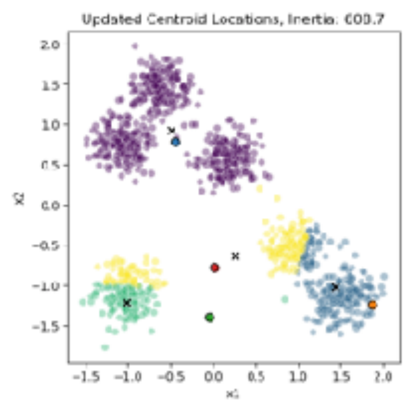


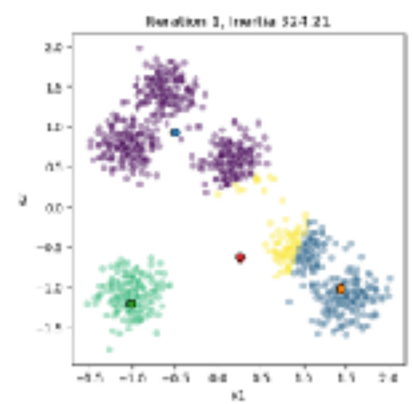


K-Means Clustering

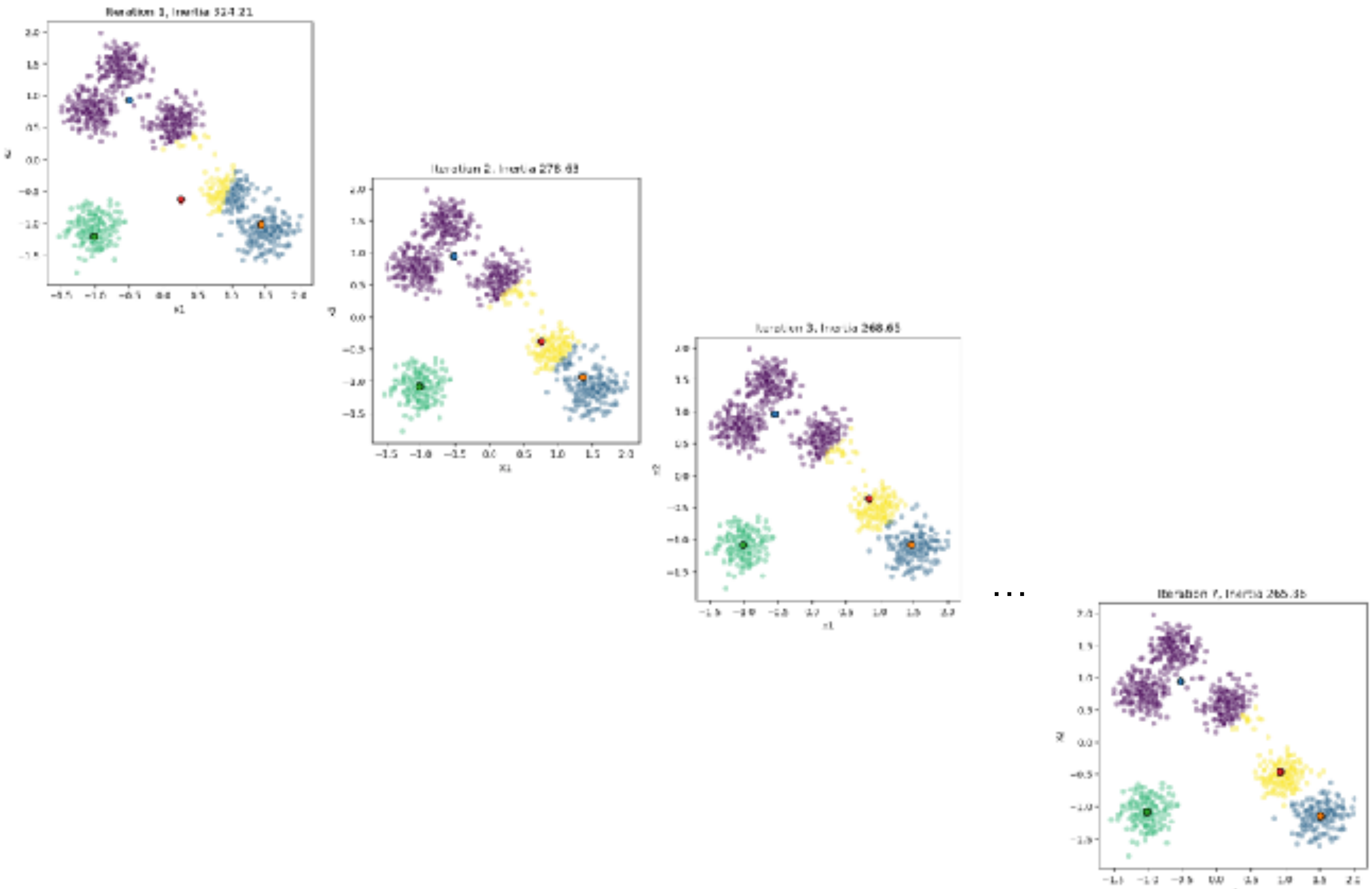








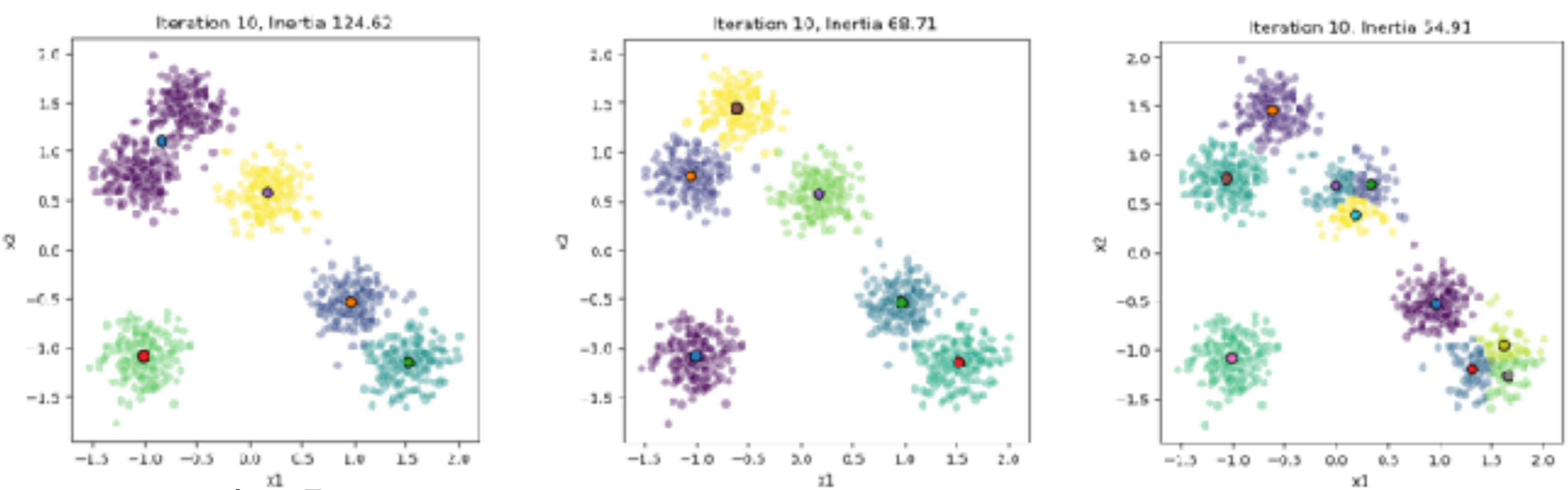
K-Means Clustering



Inertia

From SciKit documentation:
Sum of squared distances of samples to their closest cluster center

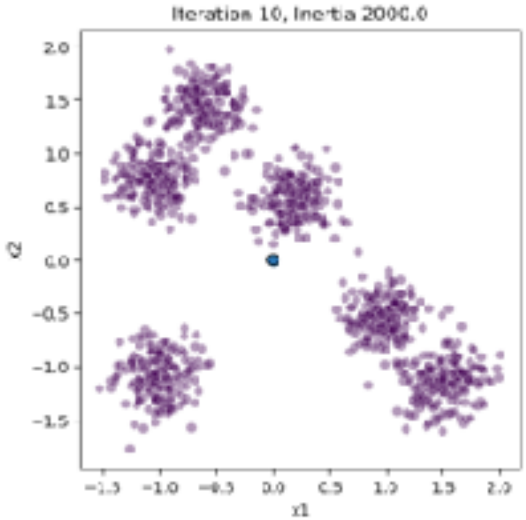
$$J = SSE = \sum_{k} \sum_{i \in k} ||x_i^{(k)} - c_k||^2$$



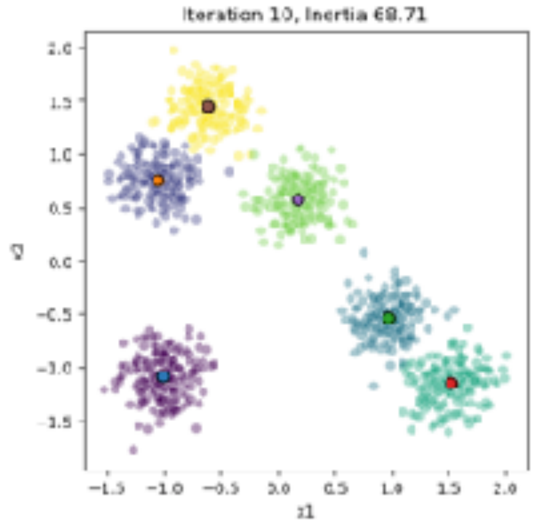
k = 5 *k* = 6 *k* = 10

Inertia

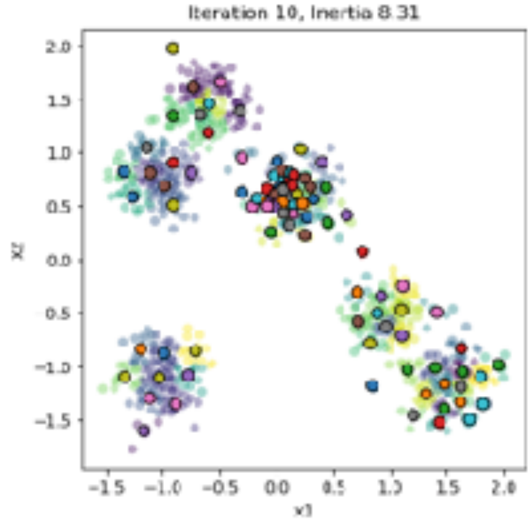
$$J = SSE = \sum_k \sum_{i \in k} ||x_i^{(k)} - c_k||^2$$



k = 1



k = 6

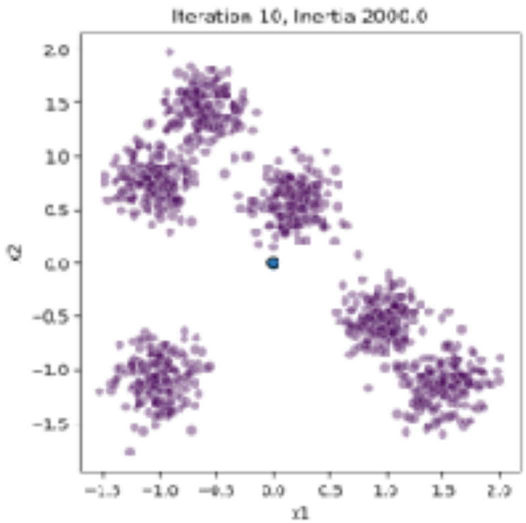


k = 100

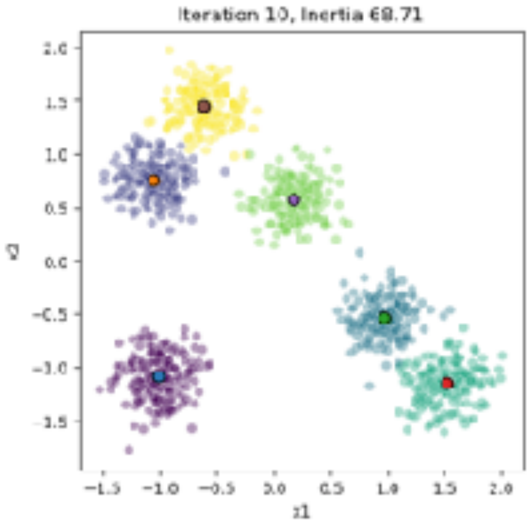
Number of Clusters?

Identifying the optimal value of *k* is the hyperparameter tuning of K-Means

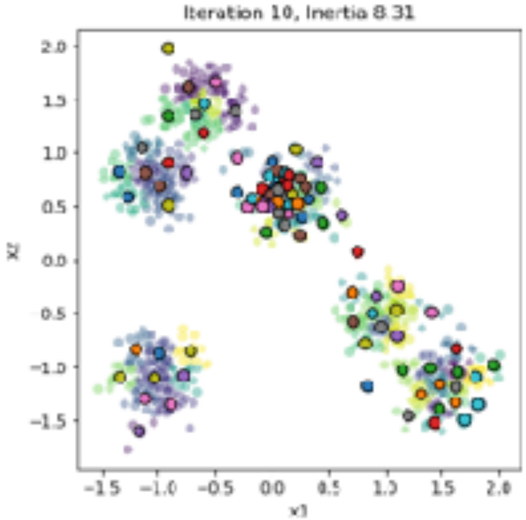
[SciKit KMeans Documentation]



k = 1



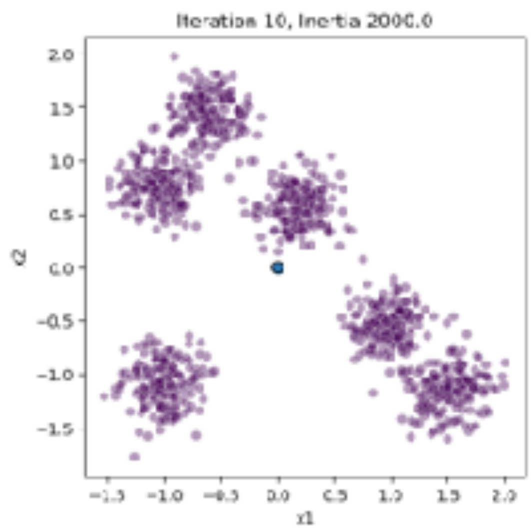
k = 6



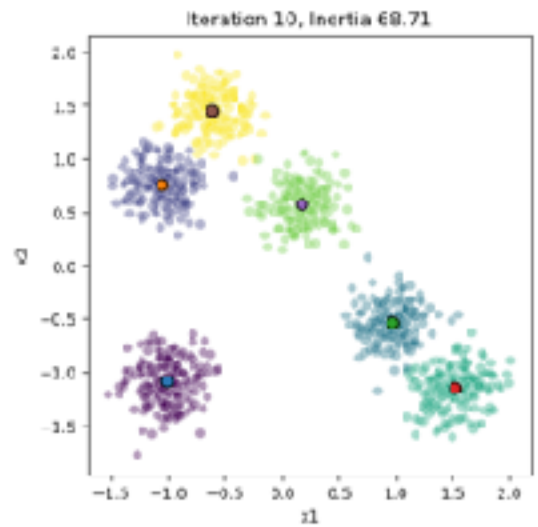
k = 100

Estimate Number of Clusters

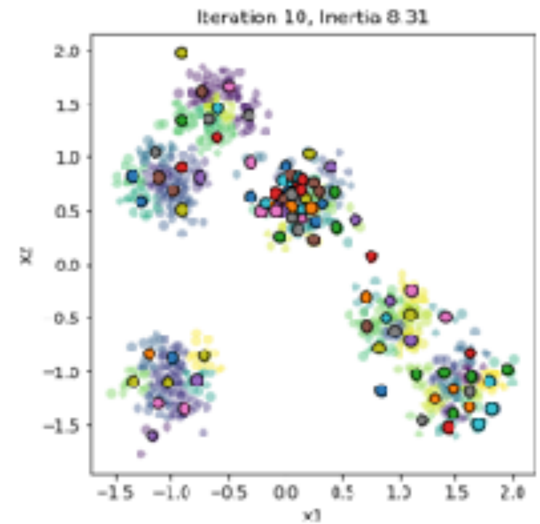
$$J = SSE = \sum_{k} \sum_{i \in k} ||x_i^{(k)} - c_k||^2$$



k = 1



k = 6

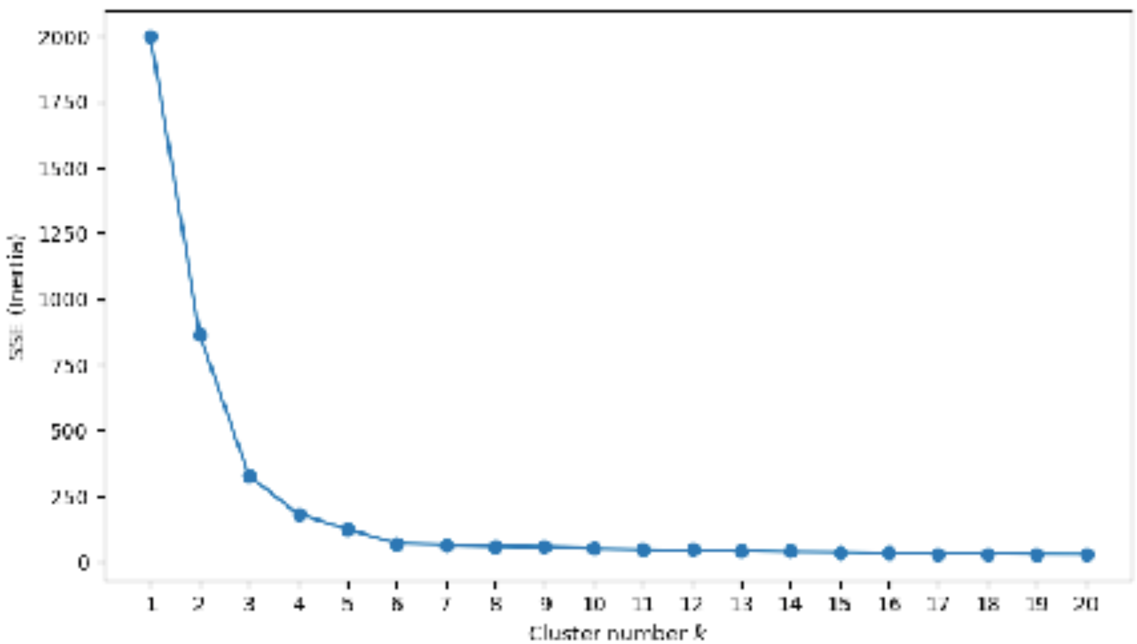


k = 100

Cannot simply minimize the SSE cost function for *k*.
k = *n* is most optimal solution.

Estimate Number of Clusters

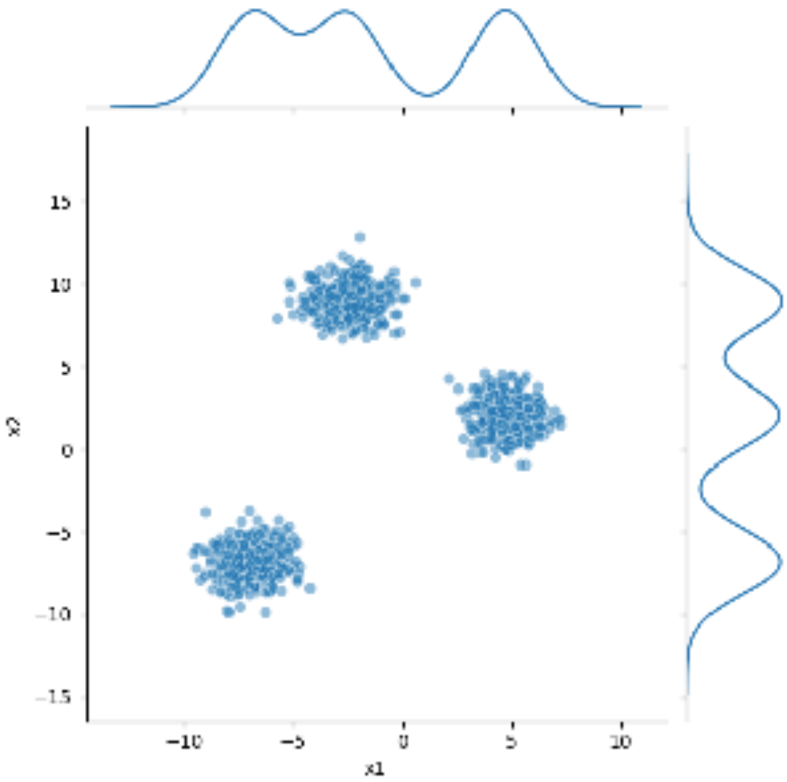
$$J = SSE = \sum_{k} \sum_{i \in k} ||x_i^{(k)} - c_k||^2$$



At what value of *k* does the observed incremental decrease in the SSE curve decrease?
Subjective methodology

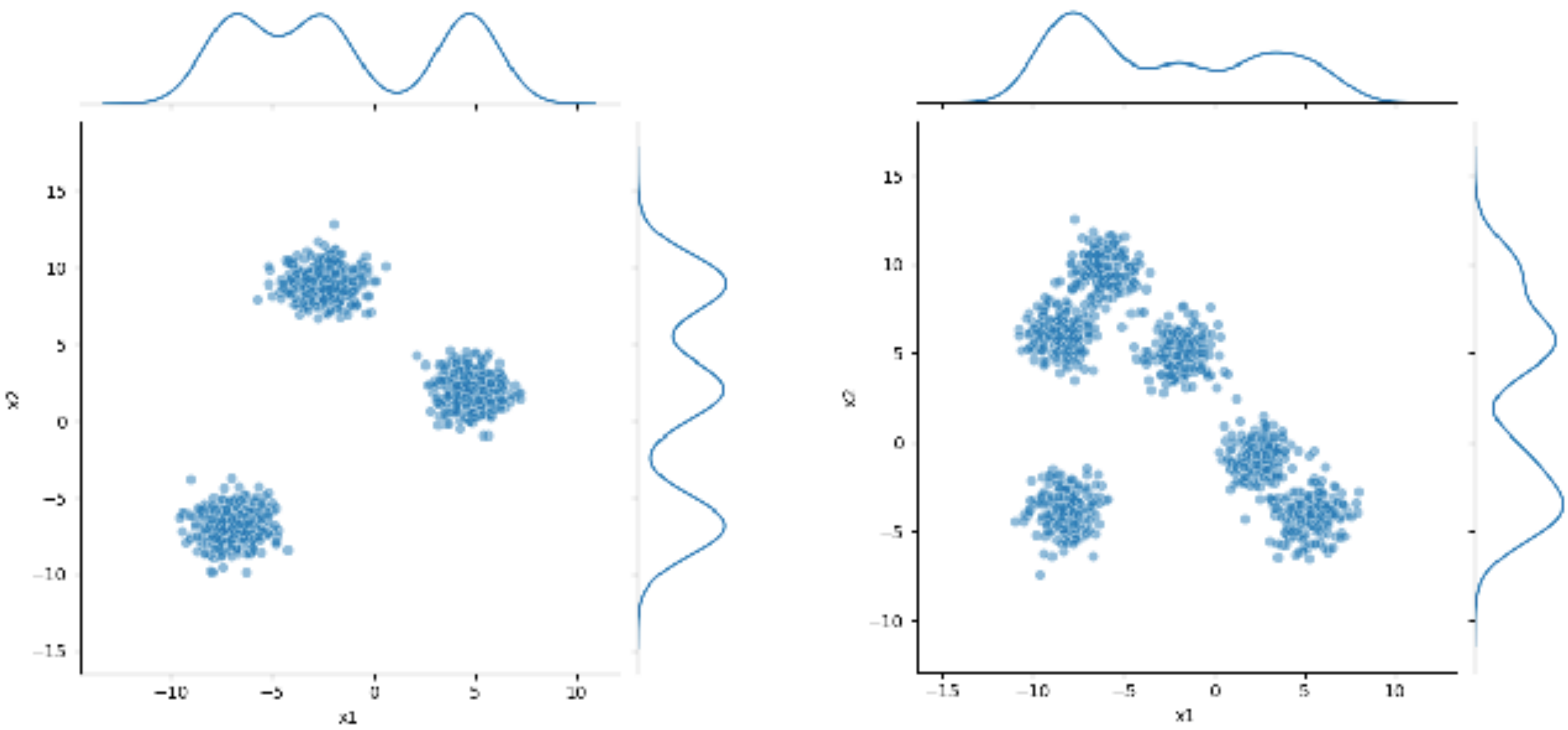
Estimate Number of Clusters

A little EDA can go a long way…



Estimate Number of Clusters

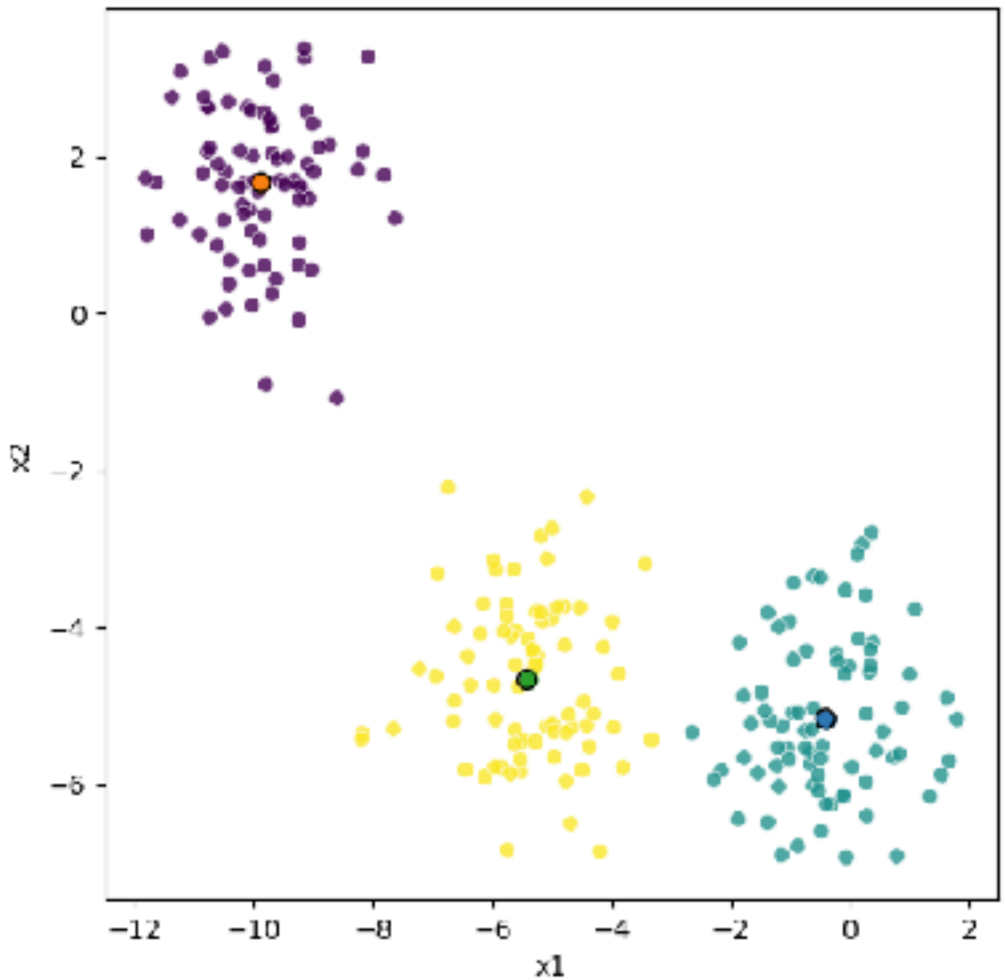
A little EDA can go a long way…



Silhouette Analysis

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample i.

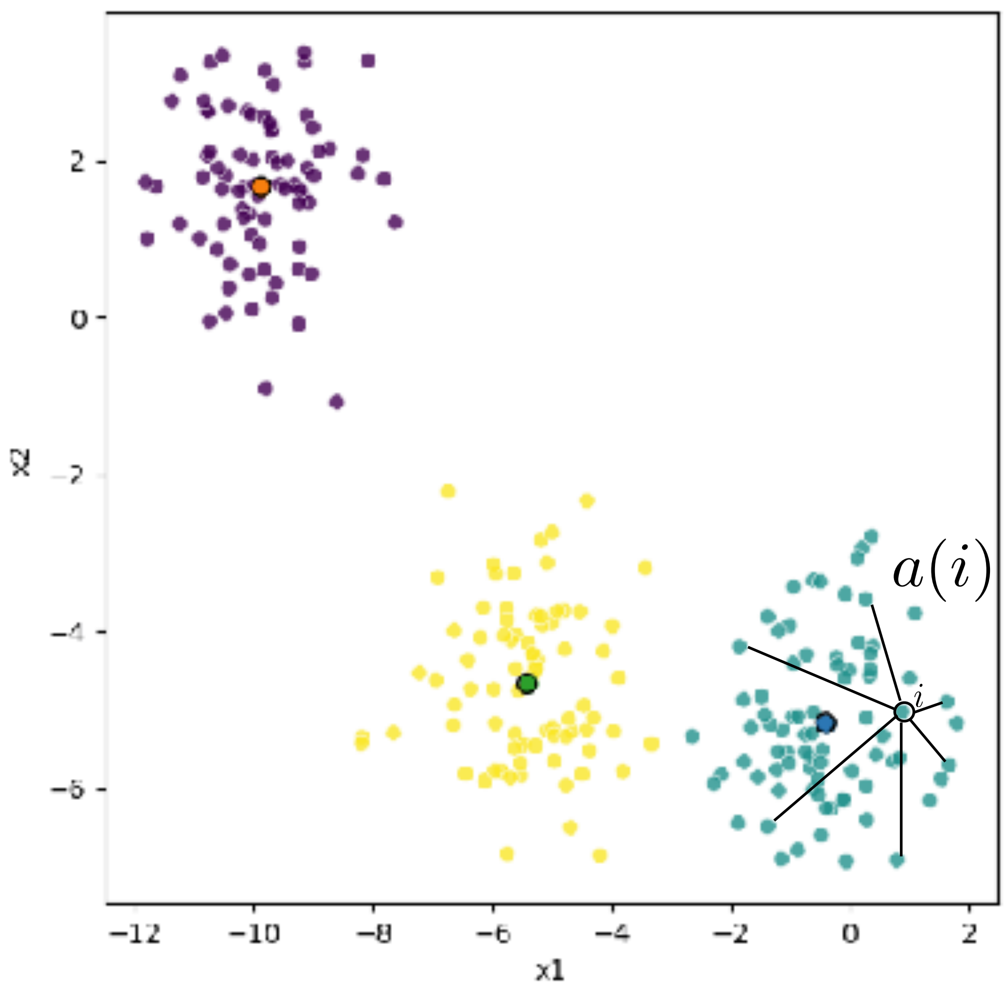
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Silhouette Analysis

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample i

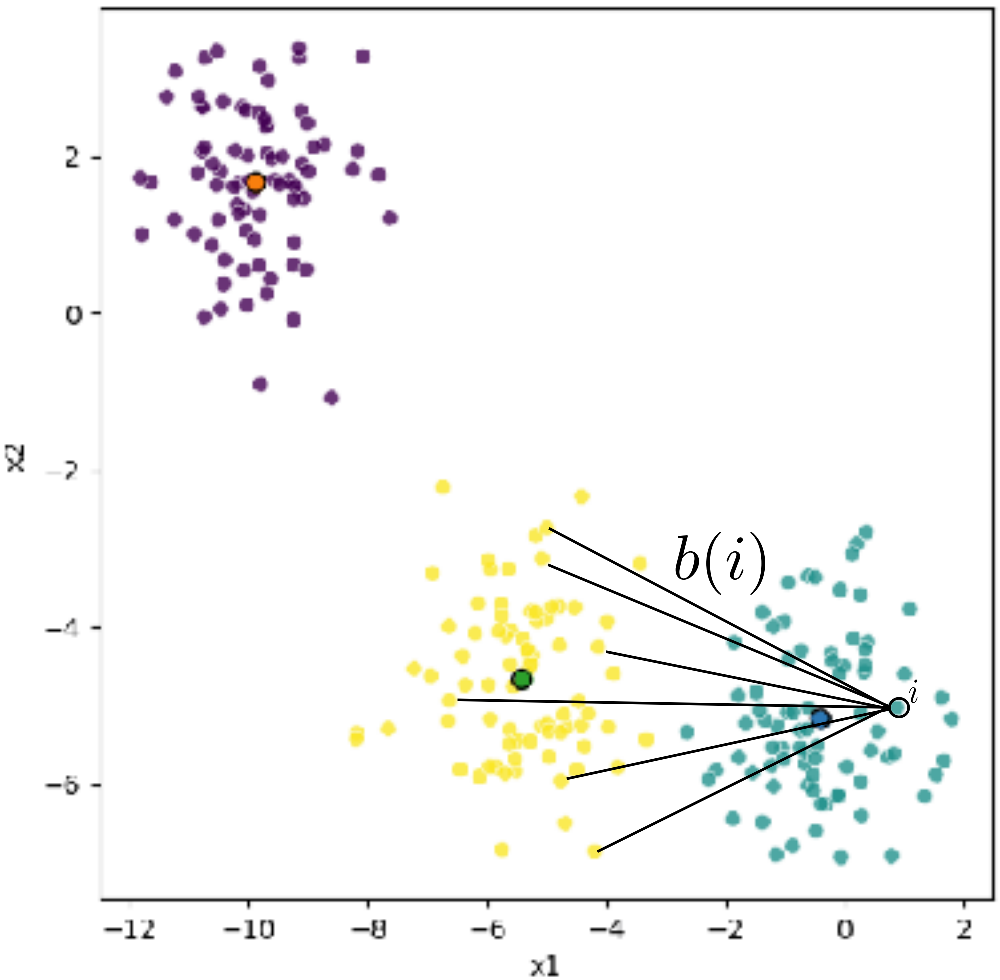
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Silhouette Analysis

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Silhouette Analysis

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$b(i) >> a(i)$

$b(i) - a(i) \approx b(i)$

$$\frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \approx \frac{b(i)}{b(i)} = 1$$

$a(i) >> b(i)$

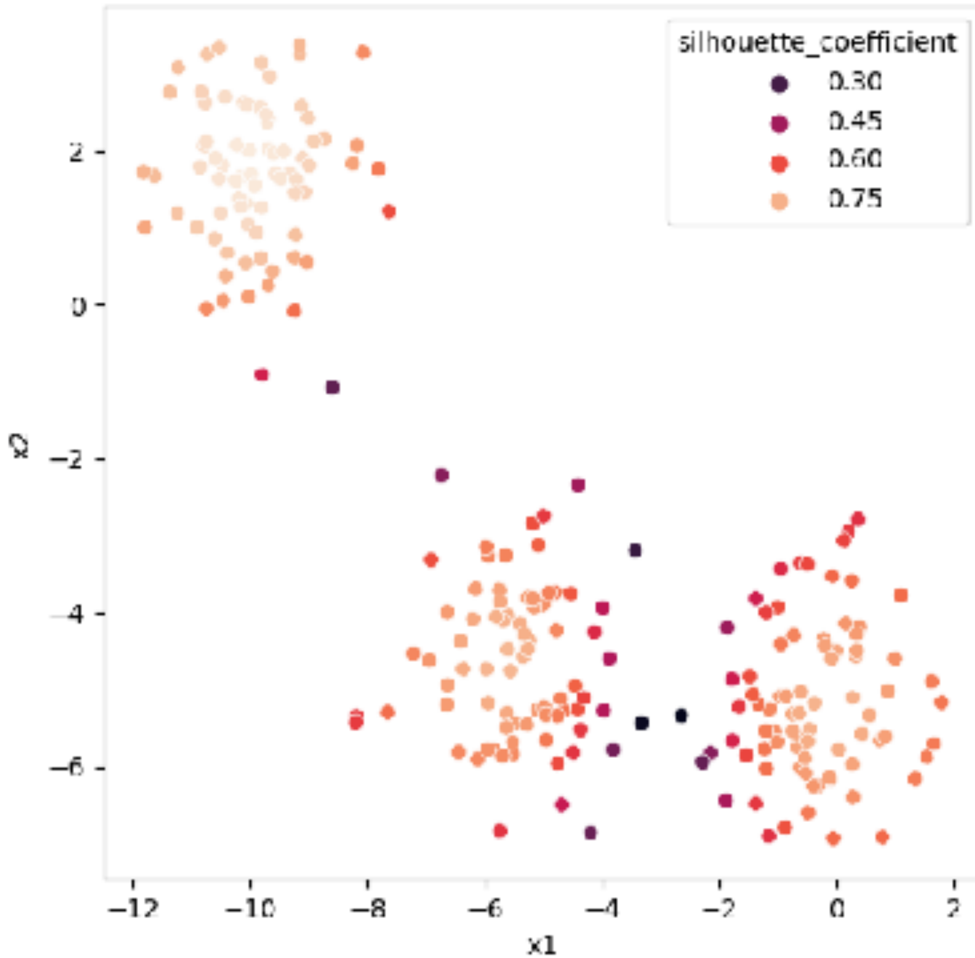
$b(i) - a(i) \approx -a(i)$

$$\frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \approx \frac{-a(i)}{a(i)} = -1$$

$a(i) \approx b(i)$

$b(i) - a(i) \approx 0$

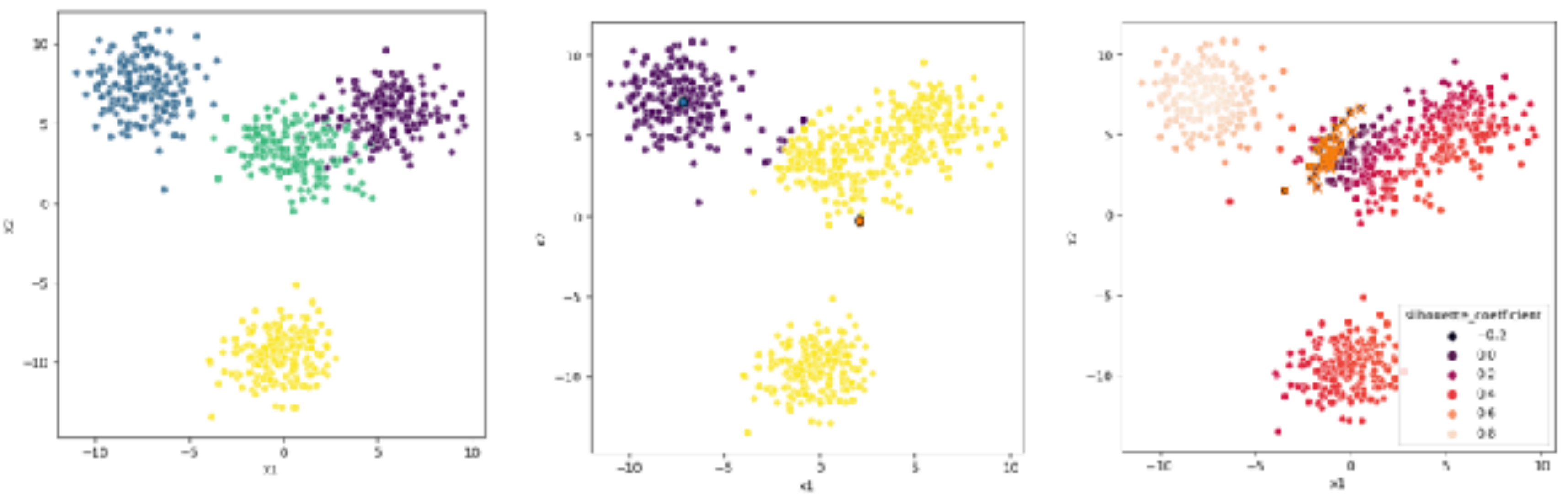
$$\frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \approx 0$$



Silhouette Analysis

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

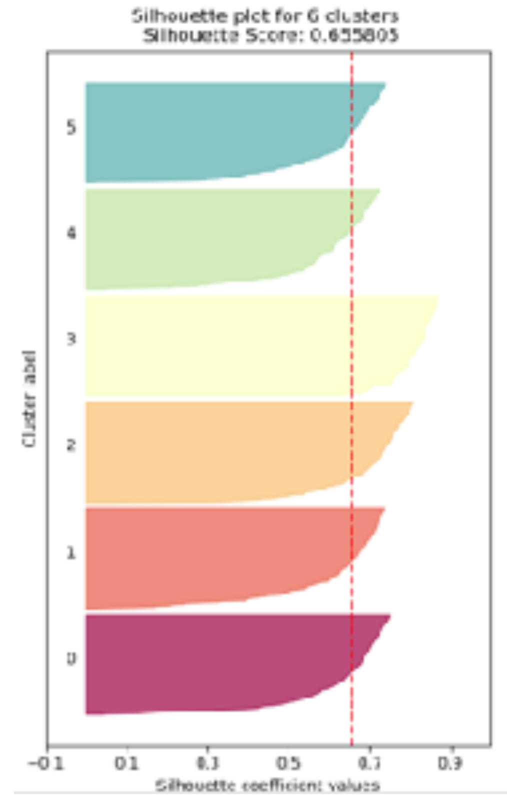


Silhouette Analysis

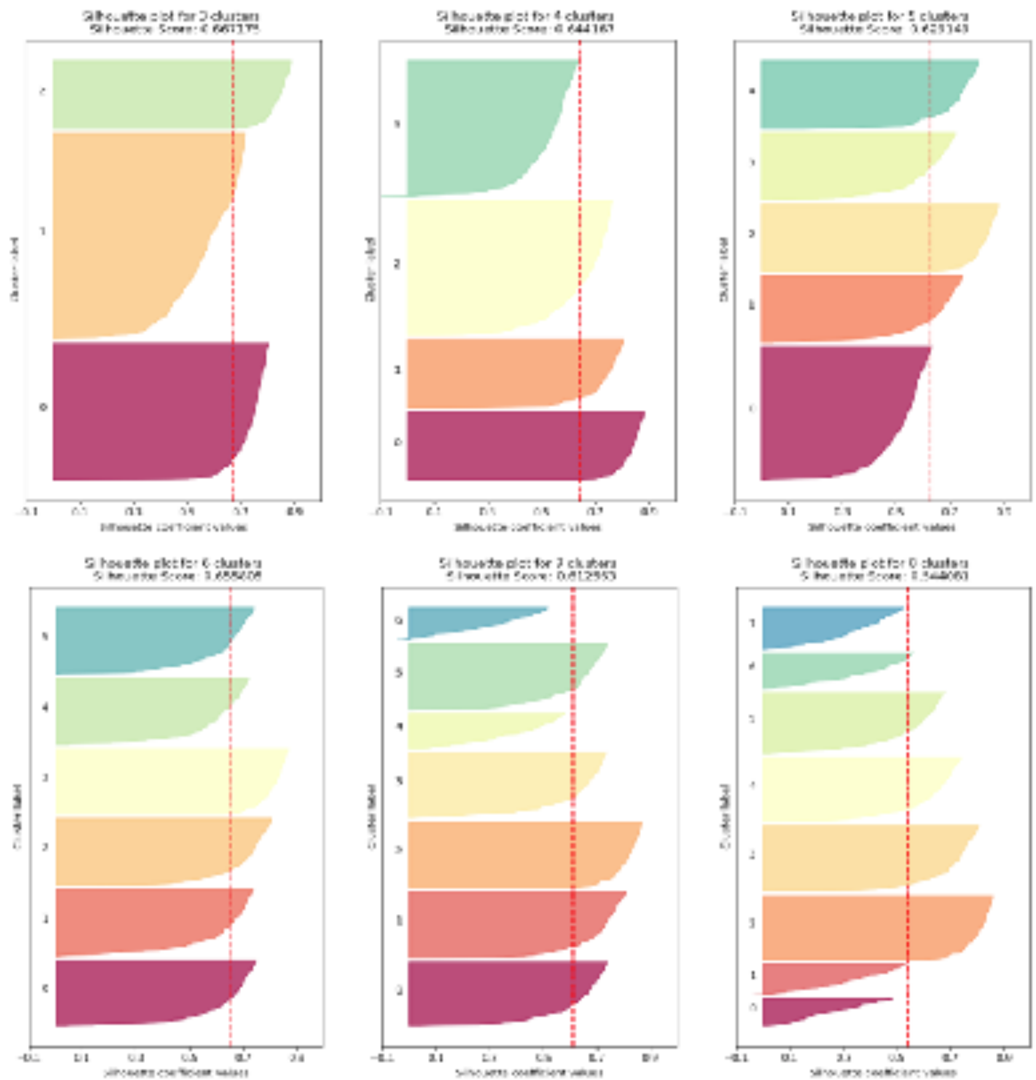
Can be helpful in estimating the number of clusters.

A clustering method has been trained with *k* number of clusters.
The silhouette coefficient is calculated for every sample *i* in the data.
The average value of these coefficients is the Silhouette Score.
The coefficients are sorted and plotted per cluster.

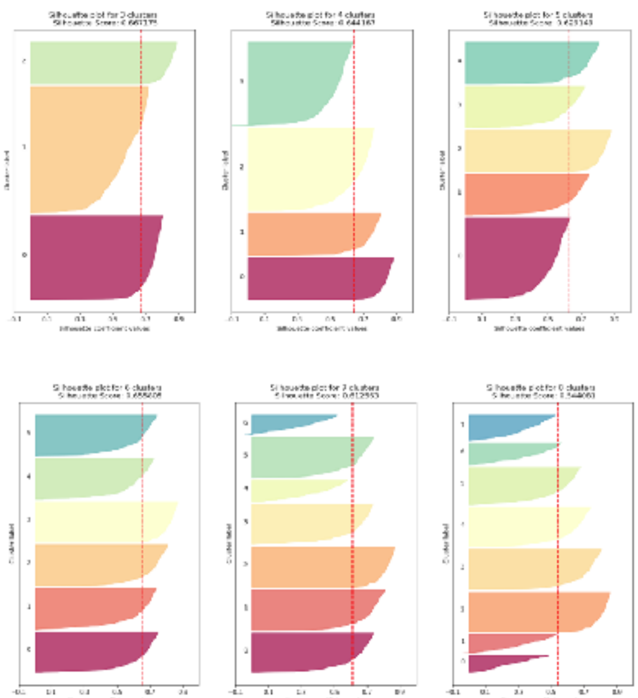
Returning to the earlier discussion
The Silhouette pilot for *k* = 6 shows:
- The clusters are approximately the same height, implying the clusters have similar sizes.
- All clusters are above the average value (Silhouette Score).

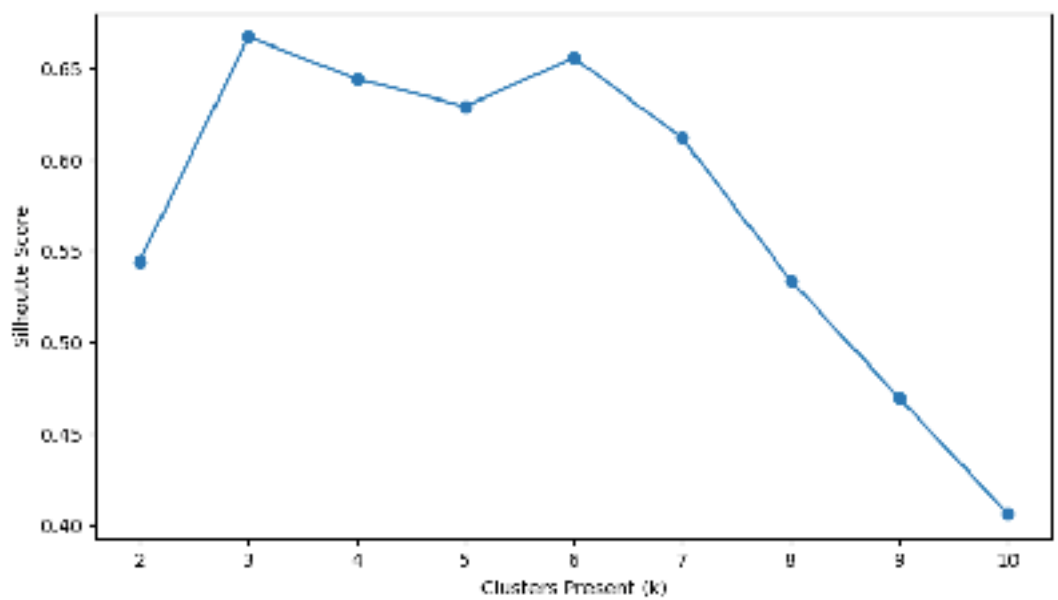


Silhouette Analysis



Silhouette Analysis





Evaluation Metrics

- Adjusted Rand Score
  - SciKit Documentation
  - Requires ground truth labels (???)
- Davies-Bouldin Index
  - SciKit Documentation
  - Assesses the separation and compactness of clusters. Good clusters are those that have low intra-cluster variation and high inter-cluster separation.
  - The smaller the DBI (always greater than zero), the better the clustering quality.

$$S_k = \frac{1}{||C_k||} \sum_{i \in C_k}^{||C_k||} (||x_i - C_k||^q)^{1/q}$$

$$M_{k_i,k_j} = ||C_{k_i}, C_{k_j}||$$

$$R_{k_i,k_j} = \frac{S_{k_i} + S_{k_j}}{M_{k_i,k_j}}$$

$$DBI = \frac{1}{K} \sum_{K} \max_i(R_{k_i,k_j})$$

Evaluation Metrics

- Adjusted Rand Score
  - [SciKit Documentation]()
  - Requires ground truth labels (???)
  - X = [1, 1, 0, 0, 0, 0]
  - Y = [0, 0, 0, 1, 0, 1]

	Y0	Y1	Row Sum
X1	2	0	a1 =2
X0	2	2	a2 =4
Col Sum	b1= 4	b2=2	

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