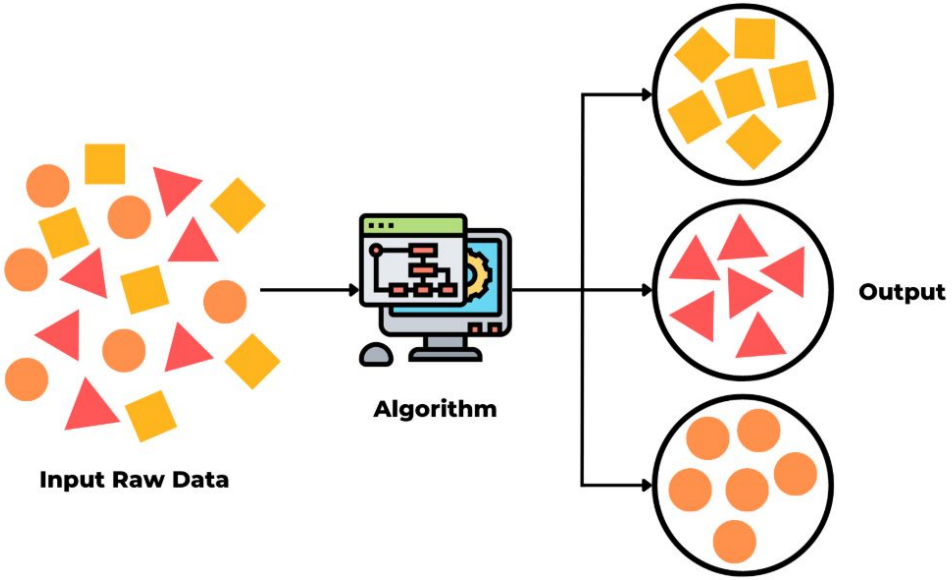
Machine learning

Lecture 6 - Unsupervised learning

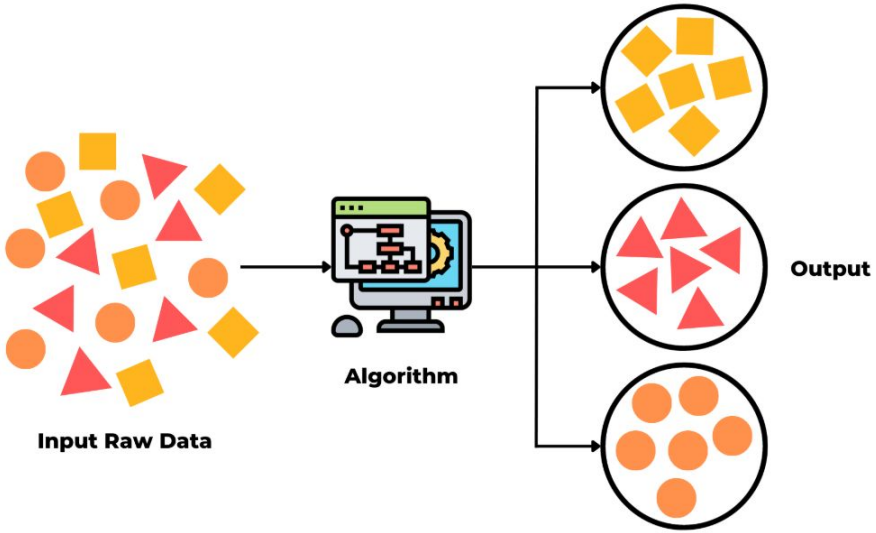
24/04/2025

Toker Gilat



Unsupervised learning

Unsupervised learning is a type of machine learning algorithm that **explores patterns in datasets without a specified target outcome.** Essentially, these algorithms are **tasked with finding 'hidden structures'** in **unlabeled data.**



https://eastgate-software.com/what-is-unsupervised-learning/

Unsupervised learning

Learning patterns from unlabeled examples:

X: Observed data (features)

No y: There are no explicit labels or targets

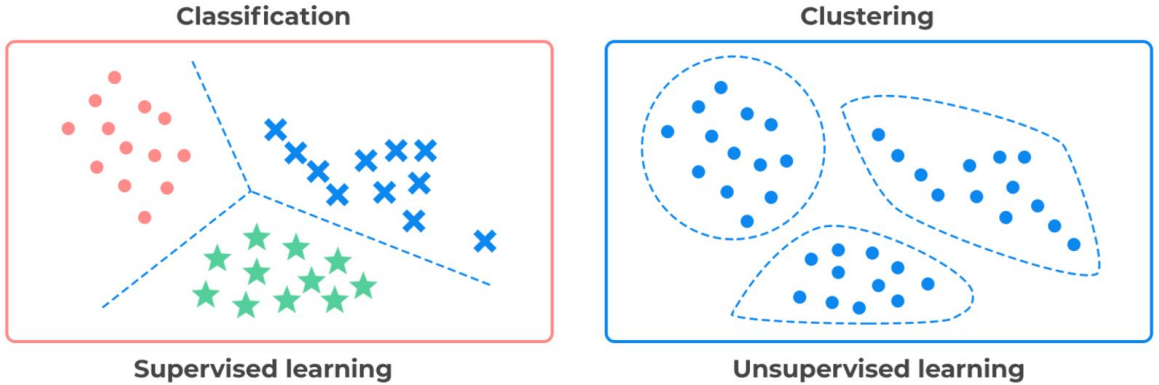
Goal: Discover hidden structure in the data

Train: train(X) → "model" (e.g., clusters, components, representations)

Predict: predict(X) → structure/embedding/cluster assignment

Evaluate: eval(X, model_output) → internal metrics (e.g., silhouette score, DBI), visual inspection

Types: Clustering, Anomaly Detection, Dimensionality Reduction, and more…

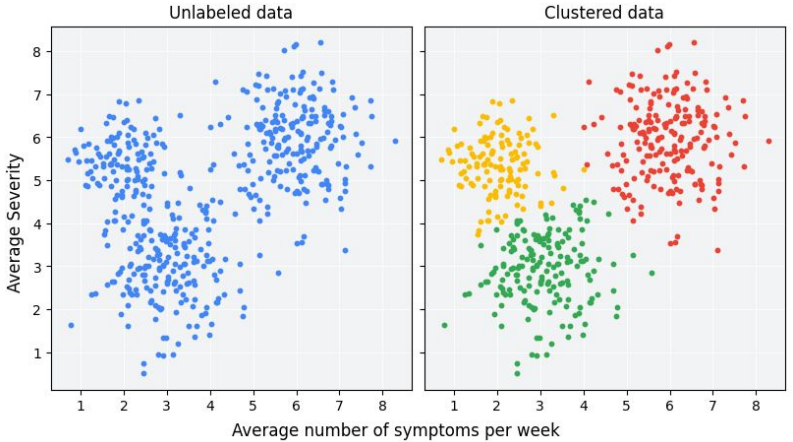


Unsupervised learning - “types”

Unsupervised learning models serve three primary tasks: **clustering, association, Anomaly Detection** and **dimensionality reduction.**

Clustering:

- **Definition:** Clustering groups similar data points into clusters based on their characteristics or similarity. It aims to identify inherent structures within the data.
- **Examples of Algorithms:** K-Means, Hierarchical Clustering
- **Applications:** Customer segmentation in marketing, Grouping similar articles in a news aggregator, Image segmentation in computer vision.

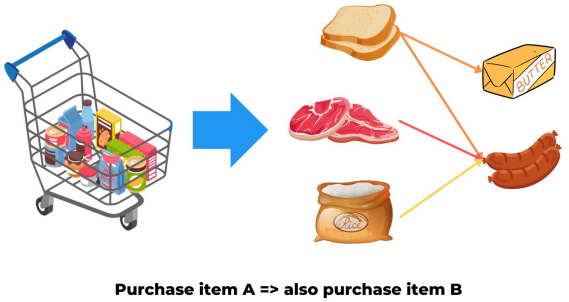


Unsupervised learning - “types”

Unsupervised learning models serve three primary tasks: **clustering, association,** and **dimensionality reduction.**

<u>Association</u>:

- **Definition:** Association finds rules and relationships between variables in a dataset. It focuses on discovering how different features or items are related.
- **Examples of Algorithms:** Apriori Algorithm, FP-Growth
- **Applications:** Market basket analysis: "People who buy bread often buy butter.", Recommendation systems: Suggesting additional products based on prior purchases



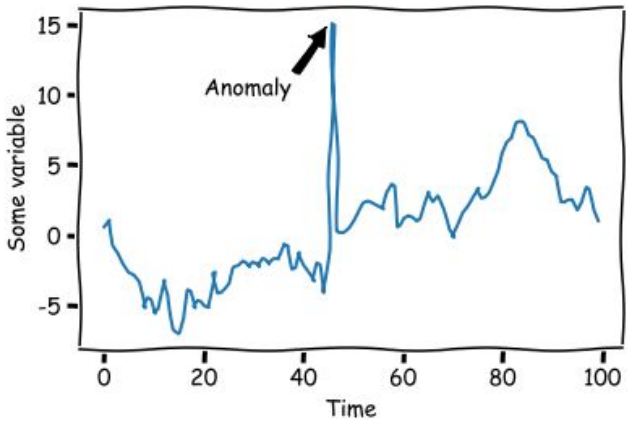
Purchase item A => also purchase item B

Unsupervised learning - “types”

Unsupervised learning models serve three primary tasks: **clustering, association, Anomaly Detection** and **dimensionality reduction.**

<u>Anomaly Detection</u>:

- **Definition:** Anomaly detection identifies data points or patterns that deviate significantly from the majority of the data.
- **Examples of Algorithms:** Isolation Forest
  One-Class SVM
- **Applications:** Fraud detection in financial transactions, Fault detection in manufacturing, Healthcare

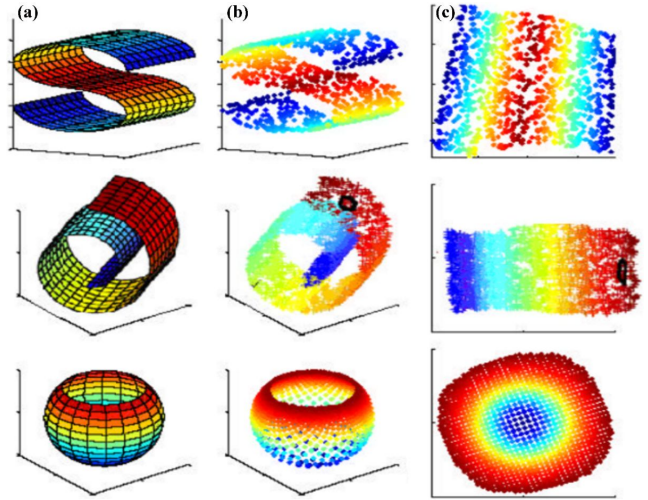


Unsupervised learning - “types”

Unsupervised learning models serve three primary tasks: **clustering, association,** and **dimensionality reduction.**

<u>Dimensionality Reduction</u>:

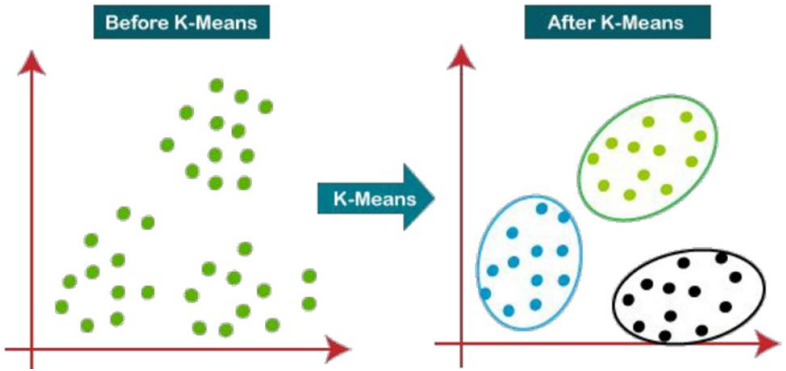
- **Definition:** Dimensionality reduction reduces the number of features in the dataset while retaining essential information. This makes the data easier to analyze and visualize.
- **Examples of Algorithms:** PCA (Principal Component Analysis)
- **Applications:** Reducing noise and lowering computational costs.



Unsupervised learning - clustering

K-means

- **Assumption:** Groups (clusters) are formed by centroids close to the data points.
- **What is it?** K-means is an iterative, centroid-based clustering algorithm that partitions a dataset into similar groups based on the distance between their centroids. The centroid, or cluster center, is either the mean or median of all the points within the cluster depending on the characteristics of the data.
- **Key Parameter:** number of clusters (k)



Unsupervised learning - clustering

K-means

How does K-means clustering work?

1. **Initialize k**
   - Randomly initialize *k* centroids, where *k* is the predefined number of clusters.
2. **Find the closest centroid & update cluster assignments**
   - Expectation Step: Assign each data point to its closest centroid based on the distance metric (e.g., Euclidean distance).
   - Maximization Step: Compute the mean of all the points within each cluster.
3. **Move the centroids**

- Update the position of each centroid to the average position of all the points assigned to its cluster.
-

Repeat steps 2 and 3 until the centroids stop changing significantly between iterations (i.e., until the algorithm converges).

Unsupervised learning - clustering

K-means

```
Algorithm K-Means Clustering:

1. Initialize centroids
  - Randomly select k data points from the dataset as initial centroids.

2. Repeat until convergence:
  a. Assignment step:
     - For each data point in the dataset:
       i.  Calculate the distance between the data point and each centroid.
       ii. Assign the data point to the nearest centroid.

  b. Update step:
     - For each centroid:
       i.  Calculate the new centroid by taking the mean of all data points assigned to it.

3. Convergence criteria:
  - Check if the centroids have stopped moving (i.e., the changes in centroid positions are
below a certain threshold).
  - If centroids have converged, terminate the algorithm.
  - If not, repeat steps 2a and 2b.

End Algorithm
```

https://www.youtube.com/watch?v=5I3Ei69I40s

Unsupervised learning - clustering

K-means

Choosing the optimal number of clusters - The elbow method

The Elbow Method is a **visual approach** used to determine the ideal 'K' in K-means clustering. It operates by calculating the **Within-Cluster Sum of Squares (WCSS),** which is the total of the squared distances between data points and their cluster center.

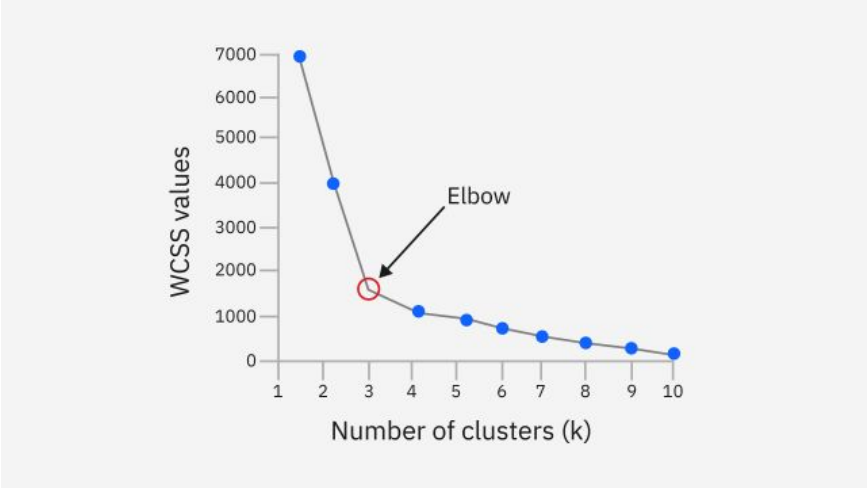
$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

However, there is a point where increasing K no longer leads to a significant decrease in WCSS, and the rate of decrease slows down. This point is often referred to as the **elbow**.

Unsupervised learning - clustering

K-means

The first step of the elbow method is to calculate the WCSS for each cluster (k). Then, the WCSS value is plotted along the y-axis and the number of clusters is plotted on the x-axis.



Unsupervised learning - clustering

K-means

Why? Balance Simplicity and Accuracy:

Few clusters (*k*): High WCSS, poor grouping.

Many clusters (*k*): Low WCSS, but overfitting.

The Elbow Point:

- Marks where adding clusters stops significantly reducing WCSS.
- Represents the best trade-off between compactness and simplicity.

Unsupervised learning - clustering

Agglomerative Clustering

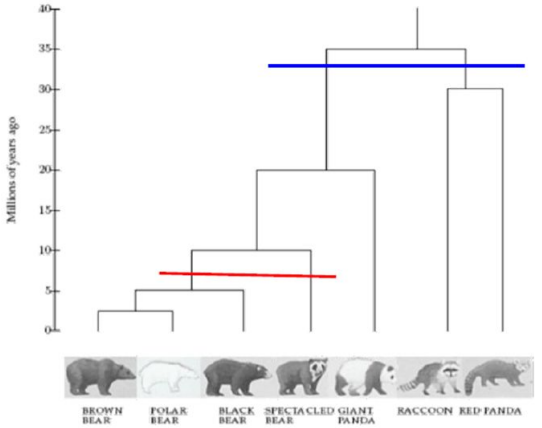
Assumption:

- Clusters are hierarchically structures
- Close clusters are (merged) clustered together.

What is it?

- A hierarchical clustering method that builds a tree of clusters (dendrogram).
- Works bottom-up, starting with each data point as its own cluster and merging clusters iteratively.

Key Parameter: Linkage Methods, n_clusters, Distance Metric



Unsupervised learning - clustering

Agglomerative Clustering

How it Works:

1. Begin with n clusters (each data point is a cluster).
2. Merge the two closest clusters based on a distance metric.
   Linkage Methods: Define how the **distance** between clusters is measured
3. Repeat until only one cluster (or desired number of clusters) remains.

Unsupervised learning - clustering

Agglomerative Clustering

Pseudocode:

```
1  while True do
2  |   a singleton cluster ← each original concept;
3  |   find a pair of clusters, (a) and (b), so that their similarity s[(a), (b)] = max(s[(m), (n)]);
4  |   if (s[(a), (b)] ≤ threshold) then
5  |   |   terminate the while loop;
6  |   else
7  |   |   merge (a) and (b) into a new cluster (a + b);
8  |   |   update M' by deleting both the row and the column corresponding to (a) and (b);
9  |   end
10 end
11 output current clusters as equivalent concept pairs;
```

Unsupervised learning - clustering

Agglomerative Clustering

Zoom in into Linkage Methods

Linkage Methods: Define how the **distance** between clusters is measured:

- Single Linkage: Minimum distance between points in clusters.
- Complete Linkage: Maximum distance between points in clusters.
- Average Linkage: Mean distance between all points in clusters.
- Ward's Method: Minimizes variance between clusters.

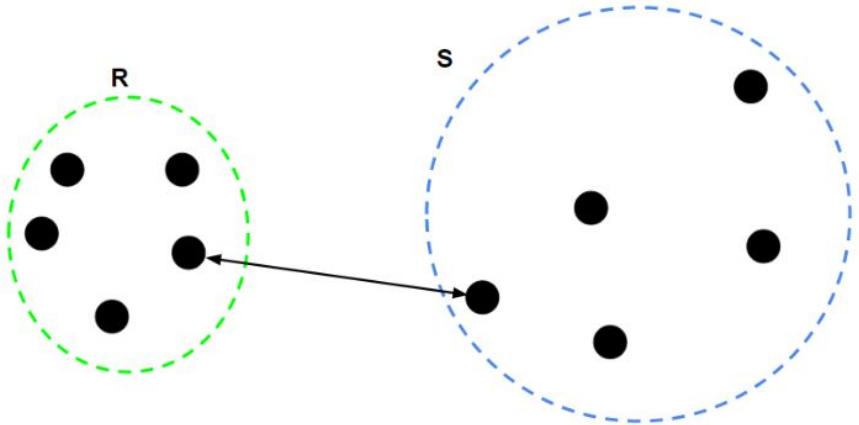
- **Distance Metric:** Defines how similarity is calculated (e.g., Euclidean, Manhattan).

Unsupervised learning - clustering

Agglomerative Clustering

Single Linkage: Minimum distance between points in clusters.

$$L(R,S) = min(D(i,j)), i \epsilon R, j \epsilon S$$



Unsupervised learning - clustering

Agglomerative Clustering

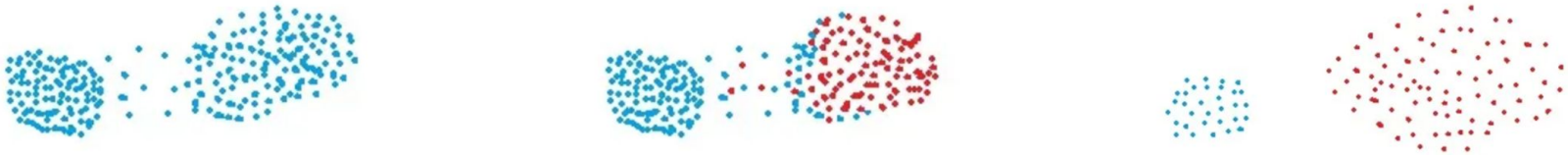
Single Linkage: Minimum distance between points in clusters.

Pros:

- Can effectively separate non-elliptical clusters as long as there is a **reasonable gap between them.**

Cons:

- Struggles to separate clusters if there is noise between them.
- Can result in elongated or **chain-like clusters** due to the "chaining effect."

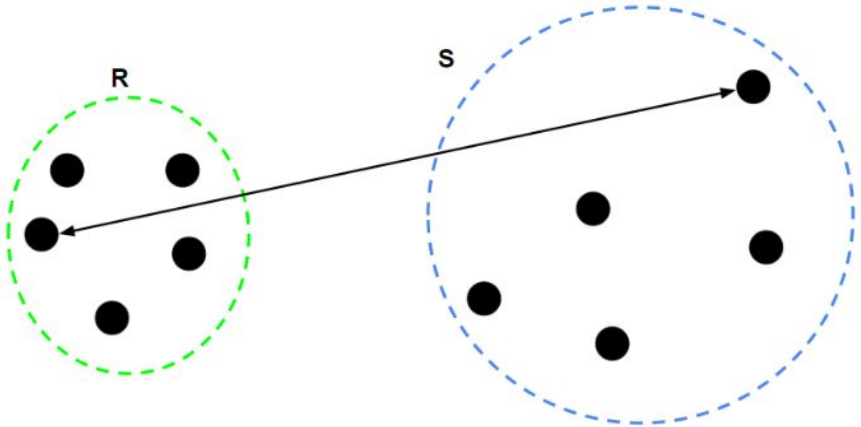


Unsupervised learning - clustering

Agglomerative Clustering

Complete Linkage: Maximum distance between points in clusters.

$$L(R,S) = max(D(i,j)), i \epsilon R, j \epsilon S$$



Unsupervised learning - clustering

Agglomerative Clustering

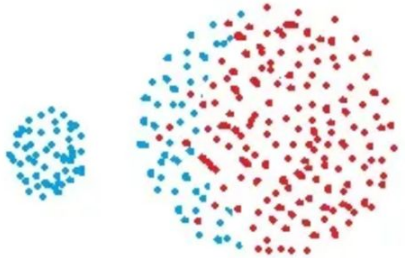
Complete Linkage: Maximum distance between points in clusters.

Pros:

- Performs well in separating clusters, especially in the presence of noise between clusters.

Cons:

- Biased towards globular clusters
- Tends to break large clusters

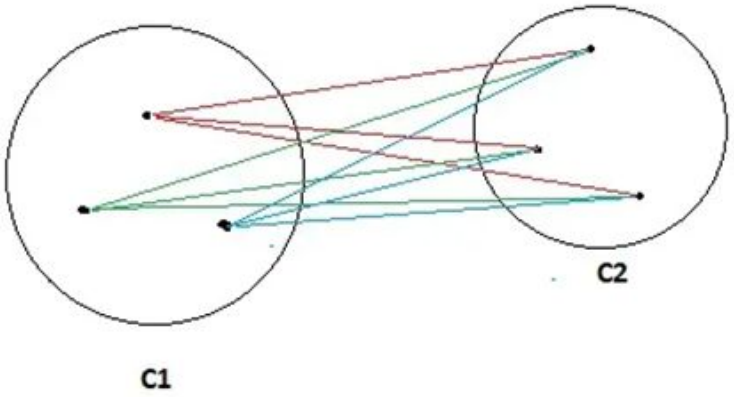


Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



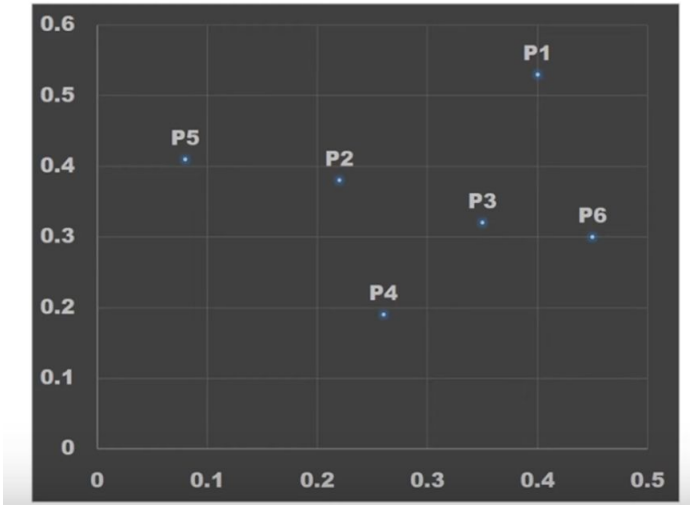
Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30



Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1						
P2	$\sqrt{(0.40-0.22)^2+(0.53-0.38)^2}$					
P3						
P4						
P5						
P6						

Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.28	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.28	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Unsupervised learning - clustering

Agglomerative Clustering

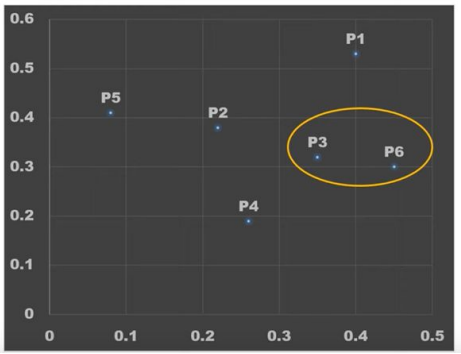
Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.28	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0



Dentrogram

Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.28	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Distance between {P3,P6} and P1 using Average Linkage

= AVG(dist (P3,P1) , dist(P6,P1)

= (0.22+0.24)/2

= 0.23

New Distance Matrix

	P1	P2	P4	P5	P3,P6
P1	0				
P2	0.23	0			
P4	0.37	0.19	0		
P5	0.34	0.14	0.28	0	
P3,P6	0.23				

Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.28	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Distance between {P3,P6} and {P3.P6} using Average Linkage

New Distance Matrix

	P1	P2	P4	P5	P3,P6
P1	0				
P2	0.23	0			
P4	0.37	0.19	0		
P5	0.34	0.14	0.28	0	
P3,P6	0.23	0.19	0.19	0.34	0

Unsupervised learning - clustering

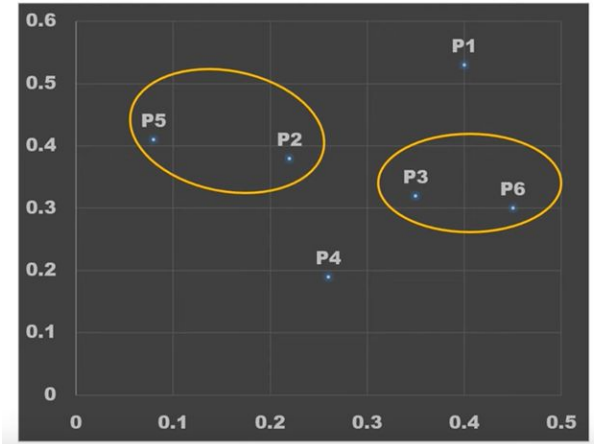
Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

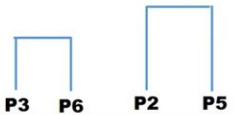
We'll choose Euclidean Distance: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance Matrix

	P1	P2	P4	P5	P3,P6
P1	0				
P2	0.23	0			
P4	0.37	0.19	0		
P5	0.34	0.14	0.28	0	
P3,P6	0.23	0.19	0.19	0.34	0



Dentrogram



Unsupervised learning - clustering

Agglomerative Clustering

Average Linkage - Mean distance between all points in clusters.

Pros:

- Handles noise between clusters well, making it robust in challenging datasets.
- Better suited for irregular or elongated clusters compared to complete linkage.

Cons:

- Tends to favor globular clusters, which may not represent more complex structures.
- Computational Cost

Unsupervised learning - clustering

Ward's Method

Ward's Method: Ward's method merges clusters in a way that minimizes the total within-cluster variance. At each step, it chooses the pair of clusters whose merger results in the smallest increase in the total sum of squared distances within clusters.

Pros:

- Produces clusters that are well-separated and compact, making it ideal for datasets with spherical or dense clusters.

Cons:

- Computational Intensity
- Assumes Euclidean Space

Unsupervised learning - clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Assumption:

- Clusters are regions of high density separated by regions of low density.

What is it?

- A density-based clustering algorithm that groups data points into clusters based on their density.
- Identifies regions of high density as clusters and labels sparse points as noise.

Key Parameter:

- No need to define the number of clusters (k) beforehand.
- Instead, tune density parameters: eps (neighborhood radius) and min_samples (minimum points to form a dense region).

Unsupervised learning - clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

How it Works:

Algorithm Steps:

1. Choose an unvisited data point.
2. Check its neighbors:
   If it has at least min_samples neighbors within eps distance:
      Mark it as a core point.
      Expand the cluster by including neighbors.
   If it is within eps distance of a core point but doesn't meet min_samples
      mark it as a border point.
   Otherwise,
      mark it as noise (non-clustered point).

Repeat: Continue until all points are visited and assigned as a core point, border point, or noise.

https://www.youtube.com/watch?v=RDZUdRSDOok

Unsupervised learning - clustering

How Do We Evaluate Unsupervised Learning?

- Unlike supervised learning, there are no ground truth labels to compare predictions against.
- Evaluation focuses on the quality of discovered patterns and cluster characteristics.

Common Evaluation Metrics

- Silhouette Score: Measures how well-separated and compact clusters are.
- Davies-Bouldin Index (DBI): Measures average similarity ratio between clusters.

Key Considerations:

- The choice of metric depends on the algorithm and dataset.
- Always visualize results to complement quantitative evaluation.
- Evaluate different parameter settings (e.g., *k* for K-Means, eps for DBSCAN) to ensure optimal results.

Unsupervised learning - clustering

Silhouette Score

The Silhouette Score measures how well each point fits within its assigned cluster compared to other clusters.

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

a: Mean distance from a point to all other points in its cluster (intra-cluster distance).
b: Mean distance from a point to all points in the nearest other cluster (inter-cluster distance).

The Silhouette Score **ranges from -1 to 1**:

- 1: Perfect clustering. The point is very close to points in its cluster and far from points in other clusters.
- 0: Overlapping clusters. The point is equally close to points in its cluster and the nearest other cluster.
- -1: Misclassified point. The point is closer to another cluster than its own.

Unsupervised learning - clustering

Davies-Bouldin Index (DBI)

The Davies-Bouldin Index (DBI) evaluates the ratio of intra-cluster distances (cluster compactness) to inter-cluster distances (cluster separation). **A lower DBI indicates better clustering.**

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$$

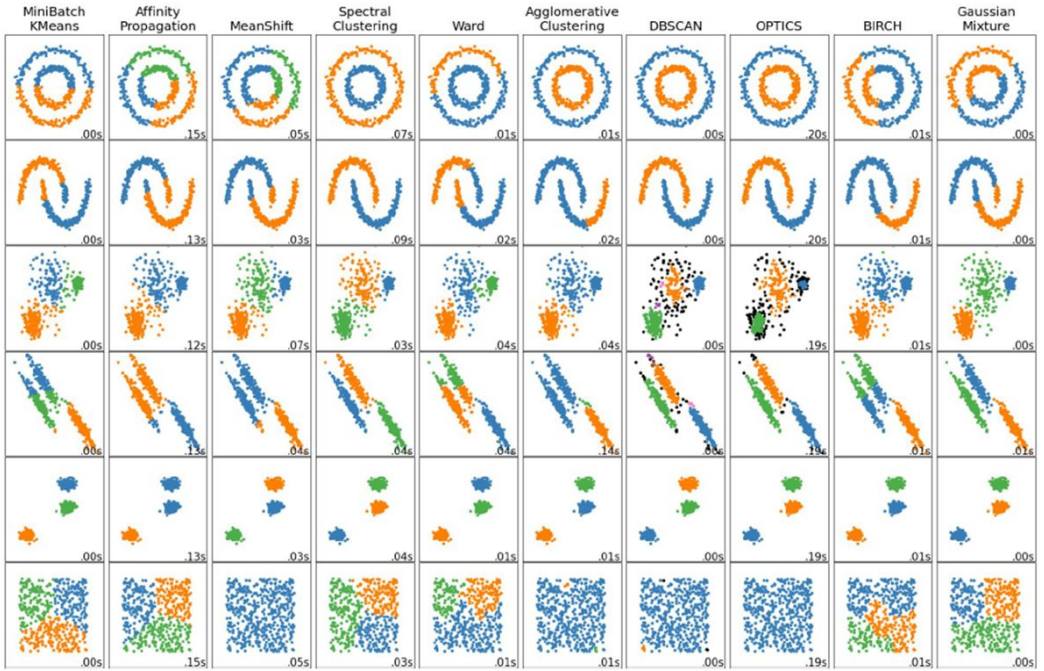
σ_i: Average distance of all points in cluster *i* to the centroid of *i* (intra-cluster distance).
$d(c_i,c_j)$: Distance between centroids of clusters *i* and j (inter-cluster distance).

The DBI Score **ranges from 0 to** ∞:

- 0: Perfect clustering. Intra-cluster distances are minimal, and inter-cluster distances are maximal.
- Higher Values (>1): Poor clustering. Clusters are either too spread out, too close to each other, or both.

Unsupervised learning - clustering

So many others...



Hands on !

Open Google colab notebook - L6_Clustering_Student