Clustering

Clustering

Clustering is an **unsupervised learning technique** used to group similar data points together based on their features. The goal is to discover hidden patterns or structures in data **without using labeled outputs**.

Clustering algorithms analyze the **similarity** between data points and assign them to the same cluster if they share common characteristics.

📌 **Example:**

- Grouping customers based on purchasing behavior.
- Segmenting images based on color patterns.

Clustering Algorithms in Machine Learning

Clustering is an **unsupervised learning** technique used to group similar data points based on their features. Different clustering algorithms follow different approaches to form clusters. Below are the most commonly used clustering algorithms:

1️⃣ K-Means Clustering (Partition-Based)

⚠️ K-means clustering

How It Works:

1️⃣ Choose the number of clusters **k**.

2️⃣ Randomly initialize k **cluster centroids**.

3️⃣ Assign each data point to the **nearest** centroid.

4️⃣ Recalculate centroids based on the mean of assigned points.

5️⃣ Repeat until centroids no longer change (convergence).

📌 **Best For:** Well-separated, spherical clusters.

❌ **Limitations:** Sensitive to the choice of kk and outliers.

2️⃣ Hierarchical Clustering (Tree-Based)

Types:

🔷 **Agglomerative (Bottom-Up Approach)** → Start with each point as its own cluster, then merge the closest ones until one cluster remains.

🔷 **Divisive (Top-Down Approach)** → Start with all data in one cluster, then split recursively.

📌 **Best For:** When the number of clusters is unknown and a hierarchical structure is needed.

❌ **Limitations:** Computationally expensive for large datasets.

3️⃣ DBSCAN (Density-Based Clustering)

How It Works:

1️⃣ Define **core points** (having at least **minPts** neighbors within a radius **ε**).

2️⃣ Expand clusters by connecting core points.

3️⃣ Mark **outliers** as noise if they don't belong to any cluster.

📌 **Best For:** Non-spherical clusters, detecting noise & anomalies.

❌ **Limitations:** Struggles with varying densities.

4️⃣ Gaussian Mixture Model (GMM) (Model-Based Clustering)

How It Works:

1️⃣ Assume data is generated from multiple **Gaussian distributions**.

2️⃣ Estimate the probability of each point belonging to a cluster using **Expectation-Maximization (EM)**.

3️⃣ Assign points based on the highest probability.

📌 **Best For:** Overlapping clusters with different shapes & sizes.

❌ **Limitations:** Computationally complex, assumes Gaussian distribution.

Comparison of Clustering Algorithms

Algorithm	Best Use Case	Handles Noise?	Needs k?	Cluster Shape
K-Means	Large datasets, well-separated clusters	❌ No	✅ Yes	Spherical
Hierarchical	Small datasets, tree-like relationships	❌ No	❌ No	Any shape
DBSCAN	Anomaly detection, irregular clusters	✅ Yes	❌ No	Arbitrary
GMM	Soft clustering, mixed distributions	❌ No	✅ Yes	Elliptical

Final Thoughts

✅ **K-Means** → Simple, fast, and works well for structured data.

✅ **Hierarchical** → Useful when relationships between clusters matter.

✅ **DBSCAN** → Great for anomaly detection and non-uniform data.

✅ **GMM** → Good for probabilistic clustering of overlapping clusters.

Evaluation Metrics

Since clustering is an **unsupervised learning** technique, evaluating its performance is different from classification or regression. Below are some key evaluation metrics:

1️⃣ Silhouette Score

Silhouette Score measures how well data points are clustered. It calculates how similar a point is to its **own cluster** compared to other clusters.

Formula:

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

Where:

- $a(i)$ = Average distance from point i to **all other points in the same cluster**.
- $b(i)$ = Average distance from point i to **points in the nearest different cluster**.

Interpretation:

- **S(i) close to 1** → Well-clustered data.
- **S(i) close to 0** → Overlapping clusters.
- **S(i) close to -1** → Misclassified data point.

✅ Best for **checking cluster compactness and separation**.

2️⃣ Normalized Mutual Information (NMI) Score

NMI measures the quality of clustering by comparing it to a **ground-truth labeling**. It evaluates how much information is shared between the predicted and actual clusters.

Formula:

$$NMI = \frac{2 \times I(X, Y)}{H(X) + H(Y)}$$

Where:

- I(X, Y) → Mutual Information between true and predicted labels.
- H(X), H(Y), → Entropy of true and predicted labels.

Interpretation:

- **NMI = 1** → Perfect clustering (identical to true labels).
- **NMI = 0** → Random or poor clustering.

✅ Best for comparing clustering results with **ground truth**.

3️⃣ Entropy (Cluster Purity Measure)

Entropy measures the **degree of randomness** or **impurity** within clusters. It tells us how mixed or pure a cluster is.

Formula:

$$H(C) = -\sum_{i=1}^{k} P(i) \log P(i)$$

Where:

- P(i) is the proportion of points in cluster i belonging to the **correct** class.

Interpretation:

- **Lower entropy** → **Better clustering** (each cluster contains mostly one class).
- **Higher entropy** → Poor clustering (mix of multiple classes).

✅ Used to evaluate **how well a cluster contains a single type of data**.

Which Metric Should You Use?

Metric	When to Use?
Silhouette Score	When ground truth is **not available** (unsupervised evaluation).
NMI Score	When comparing with **true labels** (supervised clustering evaluation).
Entropy	When checking **cluster purity** (how mixed clusters are).