Module 4

Thursday, July 03, 2025 9:48 AM

Sync Status

Clustering and classification

Labeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

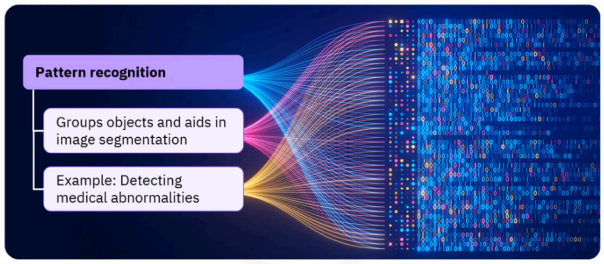
- The model operates without knowing defaults

Unlabeled dataset

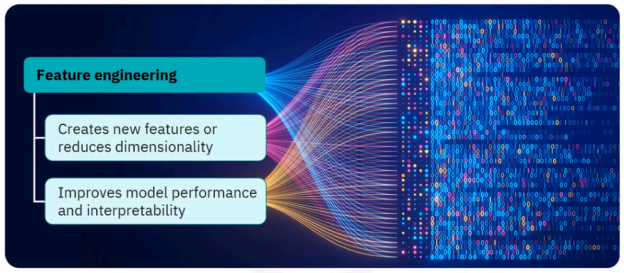
Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5

Modeling → Segmentation → K-Means

Common applications of clustering

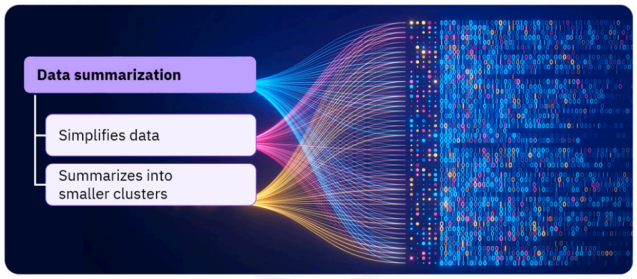


Common applications of clustering

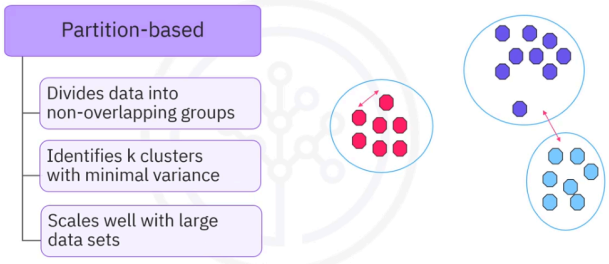


(Ctrl)

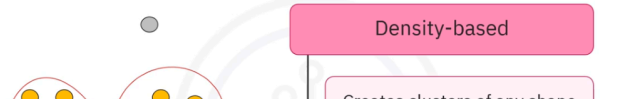
Common applications of clustering

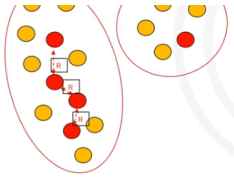


Types of clustering



Types of clustering





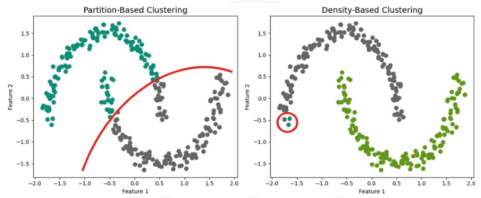
Creates clusters of any shape

Suitable for irregular clusters

Example: DBSCAN algorithm

Sync Status

Partition and density-based clustering



Partition-based clustering struggles with separation

Partitions data along a red curve

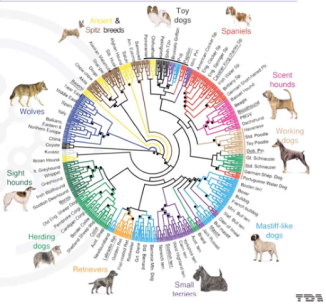
Density-based clustering separates distinct shapes

Creates third cluster of three points

Hierarchical clustering

Groups animals based on genetic similarities

Displays tree-like structure of nested clusters



Hierarchical clustering

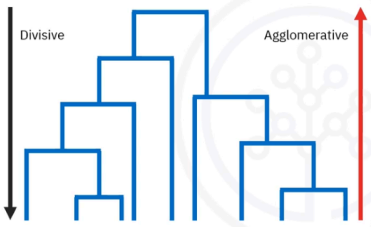
Divisive clustering

Uses a top-down approach

Starts with a single root cluster

Iteratively splits into smaller child clusters

Hierarchical clustering



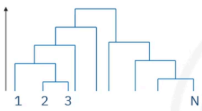
Agglomerative clustering

Employs a bottom-up approach

Starts with individual clusters

Merges similar clusters into larger parents

Agglomerative hierarchical clustering



$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Initialize N clusters with single data points

Compute a distance matrix for clusters

Display distances between each pair of points

K-means algorithm

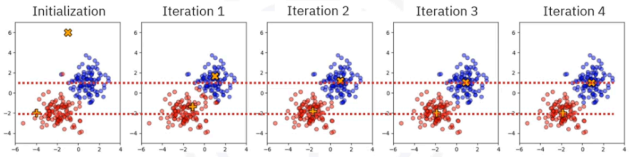
1. **Initialize the algorithm:**
- Select the number of clusters, k
- Randomly select k centroids

2. **Iteratively assign points to clusters and update centroids:**
- Compute distance matrix
- Assign each point to cluster with nearest centroid
- Update cluster centroids as the mean position

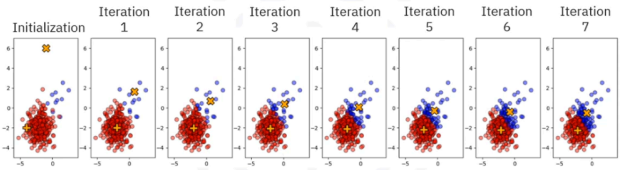
3. **Repeat until centroids stabilize or max iterations reached**

Sync Status

K-means clustering in action



K-means failure with imbalanced clusters



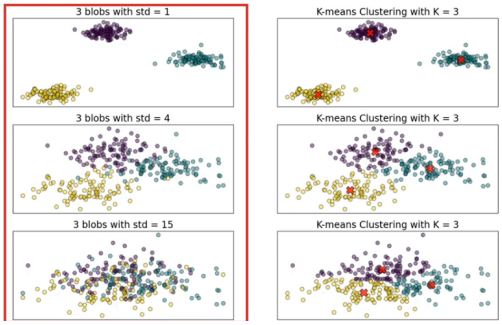
K-means optimization

Goal: Minimize within-cluster sum of squares:

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

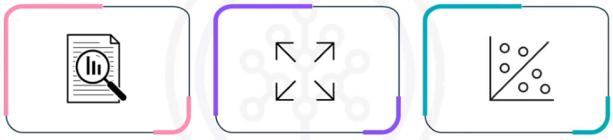
- K = Number of clusters
- $C_i = i^{\text{th}}$ cluster
- x = Data point
- μ_i = Centroid of cluster C_i
- $\|x - \mu_i\|^2$ = Squared distance between x and its cluster centroid

K-means experiments: K=3



Determining k

Choosing k is feasible when:



Data is separable

Difficult to visualize for high-dimensional spaces

Consider scatterplots between variable pairs to check for separability

Sync Status

Recap

K-means:

- Iterative, centroid-based clustering algorithm
- Partitions data set into similar groups
- Clustering algorithm categorizes data points into clusters
- Doesn't perform well on imbalanced clusters and assumes that clusters are convex
- Objective: Minimize within-cluster variance
- Heuristic techniques for gauging k-means performance:
  - Silhouette analysis
  - Elbow method
  - Davies-Bouldin index

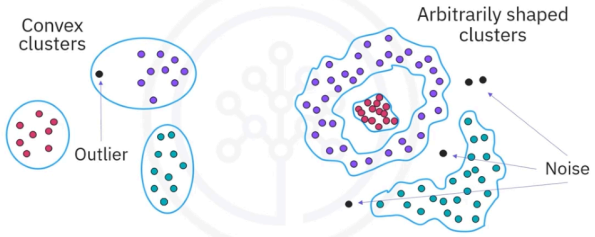
DBSCAN clustering

Density-based spatial clustering algorithm

Creates clusters centered around spatial centroids

User provides density value

Centroid and density-based clustering

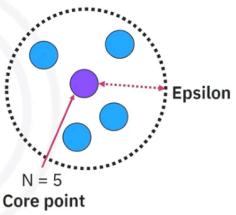


DBSCAN algorithm

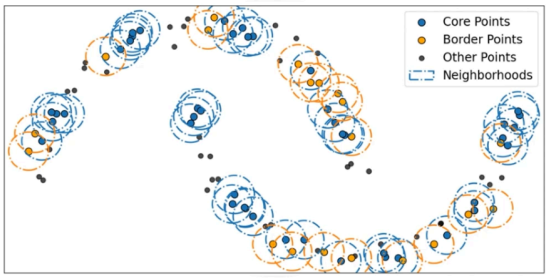
Parameters: N, epsilon

Every data point is labelled as one of the following:

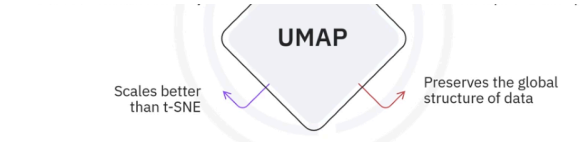
- **Core point:** Has at least N points within the epsilon neighborhood
- **Border point:** Non-core points within a core-point neighborhood
- **Noise point:** Isolated from all core-point neighborhoods
- **Clusters:** Core points and border points
- **Noise:** Non-clustered points



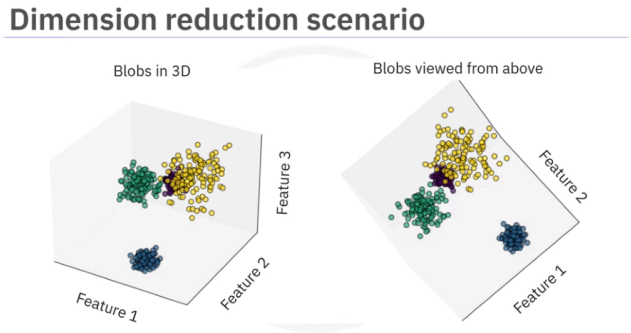
Core and border points



Uniform Manifold Approximation and Projection (UMAP)



Sync Status



Module 4 Summary and Highlights

Congratulations! You have completed this lesson. At this point in the course, you know:

- Clustering is a machine learning technique used to group data based on similarity, with applications in customer segmentation and anomaly detection.
- K-means clustering partitions data into clusters based on the distance between data points and centroids but struggles with imbalanced or non-convex clusters.
- Heuristic methods such as silhouette analysis, the elbow method, and the Davies-Bouldin Index help assess k-means performance.
- DBSCAN is a density-based algorithm that creates clusters based on density and works well with natural, irregular patterns.
- HDBSCAN is a variant of DBSCAN that does not require parameters and uses cluster stability to find clusters.
- Hierarchical clustering can be divisive (top-down) or agglomerative (bottom-up) and produces a dendrogram to visualize the cluster hierarchy.
- Dimension reduction simplifies data structure, improves clustering outcomes, and is useful in tasks such as face recognition (using eigenfaces).
- Clustering and dimension reduction work together to improve model performance by reducing noise and simplifying feature selection.
- PCA, a linear dimensionality reduction method, minimizes information loss while reducing dimensionality and noise in data.
- t-SNE and UMAP are other dimensionality reduction techniques that map high-dimensional data into lower-dimensional spaces for visualization and analysis.

Your grade: 100%

Your latest: **100%•**
Your highest: **100%•**
To pass you need at least 60%. We keep your highest score.

1.

Question 1

How might dimension reduction enhance model performance during the clustering process?

Increases feature count

Simplifies data and improves efficiency

Prevents feature loss

Removes preprocessing steps

Correct

Dimension reduction simplifies data structures, aiding visualization, and enhancing computational efficiency in clustering. However, this might come at the cost of reduced accuracy if too much information is lost in the process.

1 / 1 point

2.

Question 2

How can clustering facilitate feature selection in a dataset?

Identifies redundant features

Treats all features equally

Increases dataset dimensionality

Eliminates feature engineering

Correct

By grouping similar features together, you can select a representative feature from each cluster, which helps in simplifying the model and reducing the risk of overfitting.

1 / 1 point

3.

Question 3

How does Principal Component Analysis (PCA) contribute to model accuracy in face recognition?

Ensures equal representation

Increases training features

Extracts key facial features

Randomly selects faces

Correct

PCA extracts eigenfaces highlighting key facial features, which can be selected to reduce dimensionality, helping to accurately identify faces while reducing computational load.

1 / 1 point

4.

Question 4

Which dimensionality reduction algorithm works with complex, high-dimensional data that requires local and global structure preservation for clustering?

Uniform Manifold Approximation and Projection (UMAP)

Principal Component Analysis (PCA)

t-distributed Stochastic Neighbor Embedding (t-SNE)

Dimensionality reduction is irrelevant when working with complex, high-dimensional data

Correct

UMAP is designed to preserve local and global data structures, making it suitable for clustering applications.

1 / 1 point

5.

Question 5

What is the primary purpose of dimensionality reduction algorithms?

Increase data set features

Remove all data noise

Simplify data and maintain information content

Enhance data complexity

Correct

The main purpose of dimensionality reduction is to simplify the data set while preserving critical information.

Sync Status

Cheat Sheet: Building Unsupervised Learning Models

Unsupervised learning models

Model Name	Brief Description	Code Syntax
UMAP	UMAP (Uniform Manifold Approximation and Projection) is used for dimensionality reduction. **Pros:** High performance, preserves global structure. **Cons:** Sensitive to parameters. **Applications:** Data visualization, feature extraction. **Key hyperparameters:** • **n_neighbors:** Controls the local neighborhood size (default = 15). • **min_dist:** Controls the minimum distance between points in the embedded space (default = 0.1). • **n_components:** The dimensionality of the embedding (default = 2).	`from umap.umap_ import UMAP` `umap = UMAP(n_neighbors=15, min_dist=0.1, n_components=2)`
t-SNE	t-SNE (t-Distributed Stochastic Neighbor Embedding) is a nonlinear dimensionality reduction technique. **Pros:** Good for visualizing high-dimensional data. **Cons:** Computationally expensive, prone to overfitting. **Applications:** Data visualization, anomaly detection. **Key hyperparameters:** • **n_components:** The number of dimensions for the output (default = 2). • **perplexity:** Balances attention between local and global aspects of the data (default = 30). • **learning_rate:** Controls the step size during optimization (default = 200).	`from sklearn.manifold import TSNE` `tsne = TSNE(n_components=2, perplexity=30, learning_rate=200)`
PCA	PCA (principal component analysis) is used for linear dimensionality reduction. **Pros:** Easy to interpret, reduces noise. **Cons:** Linear, may lose information in nonlinear data. **Applications:** Feature extraction, compression. **Key hyperparameters:** • **n_components:** Number of principal components to retain (default = 2). • **whiten:** Whether to scale the components (default = False). • **svd_solver:** The algorithm to compute the components (default = 'auto').	`from sklearn.decomposition import PCA` `pca = PCA(n_components=2)`
DBSCAN	DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm. **Pros:** Identifies outliers, does not require the number of clusters. **Cons:** Difficult with varying density clusters. **Applications:** Anomaly detection, spatial data clustering. **Key hyperparameters:** • **eps:** The maximum distance between two points to be considered neighbors (default = 0.5). • **min_samples:** Minimum number of samples in a neighborhood to form a cluster (default = 5).	`from sklearn.cluster import DBSCAN` `dbscan = DBSCAN(eps=0.5, min_samples=5)`
HDBSCAN	HDBSCAN (Hierarchical DBSCAN) improves on DBSCAN by handling varying density clusters. **Pros:** Better handling of varying densities. **Cons:** Can be slower than DBSCAN. **Applications:** Large datasets, complex clustering problems. **Key hyperparameters:** • **min_cluster_size:** The minimum size of clusters (default = 5).	`import hdbscan` `clusterer = hdbscan.HDBSCAN(min_cluster_size=5)`

Sync Status

	• **min_samples:** Minimum number of samples to form a cluster (default = 10).	
K-Means clustering	K-Means is a centroid-based clustering algorithm that groups data into k clusters. **Pros:** Efficient, simple to implement. **Cons:** Sensitive to initial cluster centroids. **Applications:** Customer segmentation, pattern recognition. **Key hyperparameters:** • **n_clusters:** Number of clusters (default = 8). • **init:** Method for initializing the centroids ('k-means++' or 'random', default = 'k-means++'). • **n_init:** Number of times the algorithm will run with different centroid seeds (default = 10).	`from sklearn.cluster import KMeans` `kmeans = KMeans(n_clusters=3)`

Sync Status

Associated fuctions used

Method	Brief Description	Code Syntax
make_blobs	Generates isotropic Gaussian blobs for clustering.	`from sklearn.datasets import make_blobs` `X, y = make_blobs(n_samples=100, centers=2, random_state=42)`
multivariate_normal	Generates samples from a multivariate normal distribution.	`from numpy.random import multivariate_normal` `samples = multivariate_normal(mean=[0, 0], cov=[[1, 0], [0, 1]], size=100)`

plotly.express.scatter_3d	Creates a 3D scatter plot using Plotly Express.	`import plotly.express as px`
		`fig = px.scatter_3d(df, x='x', y='y', z='z')`
		`fig.show()`

geopandas.GeoDataFrame	Creates a GeoDataFrame from a Pandas DataFrame.	`import geopandas as gpd` `gdf = gpd.GeoDataFrame(df, geometry='geometry')`
geopandas.to_crs	Transforms the coordinate reference system of a GeoDataFrame.	`gdf = gdf.to_crs(epsg=3857)`

contextily.add_basemap	Adds a basemap to a GeoDataFrame plot for context.	`import contextily as ctx`
		`ax = gdf.plot(figsize=(10, 10))`
		`ctx.add_basemap(ax)`

pca.explained_variance_ratio_	Returns the proportion of variance explained by each principal component.	`from sklearn.decomposition import PCA`
		`pca = PCA(n_components=2)`
		`pca.fit(X)`
		`variance_ratio = pca.explained_variance_ratio_`

Your grade: 100%

Your latest: **100%•**
Your highest: **100%•**
To pass you need at least 71%. We keep your highest score.

1.
Question 1

A data scientist receives a large transaction log without any labels. What is the primary objective of applying an unsupervised algorithm to this data?

Combine every record into one typical average

Predict distinctive fraud labels for each transaction

Discover hidden structures and similarities without labels

Remove all outlier records before any pattern search

Correct

Unsupervised learning reveals natural groupings or patterns that exist in raw, unlabeled data.

1 / 1 point

2.

Question 2

A retail company wants to categorize its customers into distinct groups to optimize its marketing strategy with a dendrogram. How does hierarchical clustering help decide an optimal number of clusters?

Hierarchical clustering is specifically designed to handle high-dimensional datasets, so it performs better than other clustering methods.

Hierarchical clustering can automatically remove outliers from the dataset, which helps produce cleaner clusters.

Hierarchical clustering enables the visualization of customer groups at various levels of similarity, making it easier to decide on an optimal number of clusters.

Hierarchical clustering produces the same clusters every time, which is essential for consistent results.

Correct

Hierarchical clustering is particularly useful when a company needs to understand the nested structure of its data and visualize relationships at multiple levels. This process can help in selecting an appropriate number of clusters based on business needs.

1 / 1 point

3.

Question 3

A video streaming service uses K-means for recommending personalized content to its users based on their viewing history. Which of the following attributes would be the most relevant for clustering users?

The user's sign-up anniversary data alone

The user's total subscription payments since joining

The set of genres watched most frequently by each user

The brand of device used to stream content

Correct

Using genres of most-watched shows as the primary feature for clustering helps identify users with similar viewing habits. K-means clustering works best when the features used for grouping are directly relevant to the intended outcome.

1 / 1 point

4.

Question 4

A bank wants to detect fraudulent transactions by analyzing customer spending patterns using density-based spatial clustering of applications with noise (DBSCAN). Why is DBSCAN a good fit for this analysis?

It only works with transactions of similar value to generate a cluster.

It groups data based on density and can isolate suspicious events.

It requires defining a fixed number of clusters to work accurately.

It forces every transaction into a cluster of suspicious events.

Correct

DBSCAN is useful to cluster dense areas of typical spending behavior while detecting outliers, such as transactions that deviate significantly. These outliers can be flagged for further fraud investigation.

1 / 1 point

5.

Question 5

Why is customer-behavior data reduced with t-SNE primarily in two-dimensional scatter plots?

Enforces linear projections of high-dimensional data

Allocates equal distance between every pair of reduced points

Maintains neighborhood similarities, aiding visual discovery of customer segments

Automatically label each segment by demographic type

Correct

t-SNE emphasizes near-neighbor relationships, which surface natural segments in a 2-D scatter.

1 / 1 point

6.

Question 6

A research team is studying environmental factors affecting plant growth, using data on soil pH, temperature, and humidity. They use principal component analysis (PCA) to simplify the analysis. Which aspect of PCA makes it a suitable technique for this research?

Reduces data to key components for simple analysis

Identifies nonlinear relationships in the data

Combines all variables into one component

Forces all features to be uncorrelated

Correct

PCA simplifies data by reducing dimensions and highlighting the most important variables, making it easier to analyze key environmental factors.

1 / 1 point

7.

Question 7

A company wants to visualize high-dimensional data on customer demographics and behavior to understand customer segments. They decide to use t-distributed stochastic neighbor embedding (t-SNE) for dimensionality reduction to two or three dimensions. Which aspect of t-SNE makes it suitable for this task?

Combines all features into one dimension

Works best on linear data in one dimension

Forces all features to be uncorrelated in two dimensions

Preserves local relationships between similar data points

Correct

t-SNE is great at preserving local relationships between similar data points when reducing dimensions, making it useful for visualizing clusters and patterns in complex data.