CLUSTERING ANALYSIS

Finding Natural Groups in Your Data

Class Notes & Handout

Predictive Analytics & Data Mining

Dahlkemper School of Business | Gannon University

Spring 2026

Prepared by: Dr. Benyawarath “Yaa” Nithithanatchinnapat

Contents

1. What Is Clustering?

Imagine you run a retail chain with 50,000 customers. You know their ages, incomes, purchase histories, and shopping frequency. But you have no pre-existing labels like “high-value” or “at-risk.” You just have raw data. Clustering helps you discover those groups naturally, without anyone telling the algorithm what to look for.

The Big Idea

Clustering is unsupervised learning: the algorithm finds natural groupings in your data based on similarity—no labels, no target variable, no right answer to train on. You’re discovering structure, not predicting outcomes.

Clustering vs. Classification

Students often confuse these two. Here’s the key distinction:

Feature	Classification	Clustering
Learning Type	Supervised	Unsupervised
Labels Required?	Yes – you need known outcomes	No – the algorithm finds groups
Goal	Predict a category for new data	Discover hidden structure
Example	Will this customer default? (Yes/No)	What groups exist among customers?

Why Should Business Professionals Care?

Clustering is one of the most widely used techniques in business analytics because it answers a fundamental question: Who are my customers, really? Beyond customer segmentation, organizations use clustering for fraud detection, supply chain optimization, market research, risk profiling, and much more.

Business Value

Clustering lets you move from “one-size-fits-all” to tailored strategies. Instead of sending the same marketing email to 50,000 people, you send five different versions to five distinct customer segments. That’s how Netflix recommends shows, how Amazon suggests products, and how banks flag unusual transactions.

2. K-Means Clustering

K-Means is the most widely used clustering algorithm in practice. It's fast, intuitive, and works well when your clusters are roughly spherical and similarly sized. Think of it as asking: "If I had to place k magnets in this data, where would each magnet land to attract the nearest points?"

How K-Means Works

The algorithm follows a simple iterative process:

Step 1. **Choose k** – Decide how many clusters you want. This is the hardest part, and we'll talk about it shortly.

Step 2. **Place initial centroids** – Randomly place k center points (centroids) in the data space.

Step 3. **Assign each point** – Calculate the distance from every data point to each centroid. Assign each point to its closest centroid.

Step 4. **Recalculate centroids** – Move each centroid to the average (mean) position of all points assigned to it.

Step 5. **Repeat Steps 3–4** – Keep going until centroids stop moving (convergence).

> ⚠ **Watch Out: Local Optima**
> Because centroids start randomly, K-Means can converge to different solutions each run. Best practice: run the algorithm multiple times with different starting points and pick the best result. Most software does this automatically.

Choosing k: The Elbow Method

The biggest question in K-Means: how many clusters? Too few and you lose meaningful distinctions. Too many and you're just memorizing noise.

The Elbow Method works like this: run K-Means for k = 1, 2, 3, ... and plot the total within-cluster sum of squared errors (SSE) for each k. As k increases, SSE always goes down. But at some point, the rate of improvement drops sharply—that's the "elbow." That's your sweet spot.

Think of it like squeezing a sponge: the first squeeze gets most of the water out. Each additional squeeze gets less and less. The elbow is where additional squeezes stop being worth the effort.

Distance Matters: Measuring Similarity

K-Means uses distance to decide which points belong together. The most common measure is Euclidean distance—the straight-line distance between two points. But the choice matters:

- **Euclidean distance:** Works well for continuous numerical features like age, income, and spending scores.
- **Cosine similarity:** Better for text data or when direction matters more than magnitude.
- **Manhattan distance:** Useful when features have very different scales or when outliers are a concern.

Key Rule: Always Normalize!

If income is in the thousands and age is in the tens, income will dominate the distance calculation. Always standardize (z-score) or normalize (min-max) your features before clustering. This gives every feature an equal voice.

K-Means: Strengths & Limitations

Strengths	Limitations
Fast and efficient, even on large datasets	Must specify k in advance
Easy to understand and explain to stakeholders	Produces only sphere-shaped clusters
Scales well to millions of records	Sensitive to outliers (they pull centroids)
Widely implemented in every analytics tool	Different starting points = different results

3. Hierarchical Clustering

What if you don't want to pick the number of clusters upfront? Hierarchical clustering builds a tree-like structure (called a dendrogram) that shows how data points merge into clusters step by step. You choose where to "cut" the tree to get the number of clusters you want.

Two Approaches

Agglomerative (Bottom-Up): Start with every point as its own cluster. Repeatedly merge the two closest clusters until you have one big cluster. This is the most popular approach.

Divisive (Top-Down): Start with one giant cluster containing everything. Repeatedly split it into smaller groups. Less common in practice, but useful for certain applications.

> **The Dendrogram: Your Visual Roadmap**
> A dendrogram is a tree diagram where the y-axis shows the distance (or dissimilarity) at which clusters merge. Tall vertical lines mean clusters were very different from each other when they merged. Short lines mean they were quite similar. Cut the tree horizontally at a chosen height to get your desired number of clusters.

How Agglomerative Clustering Works

1. Assign each data point to its own individual cluster (n data points = n clusters).
2. Build a distance matrix showing the distance between every pair of clusters.
3. Find the two closest clusters and merge them into one.
4. Update the distance matrix to reflect the new merged cluster.
5. Repeat steps 3–4 until all points belong to a single cluster (or you reach your desired number of clusters).

Linkage Methods: How Do We Measure Cluster Distance?

When two clusters each contain multiple points, how do you measure the distance between them? There are four main approaches:

Linkage Type	How It Works	Best For
Single	Shortest distance between any two points in the clusters	Finding elongated or chain-like clusters
Complete	Longest distance between any two points in the clusters	Finding compact, roughly equal-sized clusters
Average	Mean distance between all pairs of points across clusters	A balanced middle ground between single and complete
Centroid	Distance between the average position (centroid) of each cluster	When clusters are roughly spherical

Hierarchical Clustering: Strengths & Limitations

Strengths	Limitations
No need to specify k upfront	Can’t undo a bad merge once it’s done
Dendrogram gives a rich visual of data structure	Slow on large datasets ($O(n^3)$ time complexity)
Produces the same result every time (deterministic)	Hard to interpret the dendrogram with many data points
Flexible: can cut tree at any level for different numbers of clusters	Memory-intensive for very large datasets

4. DBSCAN (Density-Based Clustering)

What if your clusters aren't round blobs? What if they're oddly shaped, or some points are just noise that doesn't belong anywhere? That's where DBSCAN shines. DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.

> **The Core Idea**
> If a point has enough neighbors within a certain radius, it belongs to a dense region. Dense regions form clusters. Points that aren't near enough other points are labeled as noise (outliers). No forcing every point into a group.

Two Key Parameters

Epsilon (ε or Radius): The search radius around each point. If enough points fall within this radius, the area is considered dense.

MinPts (Minimum Points): The minimum number of points required within the epsilon radius to qualify an area as dense. Includes the point itself.

Three Types of Points

Point Type	Definition	Analogy
Core Point	Has at least MinPts neighbors within the epsilon radius	The popular kid – many friends nearby
Border Point	Fewer than MinPts neighbors, but within epsilon of a core point	A friend of the popular kid, but not popular themselves
Outlier/Noise	Not a core point and not within epsilon of any core point	The loner – not close enough to any group

How DBSCAN Works

6. Pick a random unvisited point.
7. Check if it's a core point (enough neighbors within epsilon).
8. If yes: start a new cluster. Add all reachable core points and their border points to this cluster.
9. If no: mark it as noise for now (it might become a border point later if a nearby cluster expands to it).
10. Move to the next unvisited point and repeat until every point has been visited.

DBSCAN: Strengths & Limitations

Strengths	Limitations
Finds clusters of any shape (not just spheres)	Sensitive to epsilon and MinPts parameters

Automatically identifies outliers/noise	Struggles when clusters have very different densities
No need to specify number of clusters upfront	Not ideal for very high-dimensional data
Robust to outliers (they’re just labeled as noise)	Determining the right epsilon can be tricky

5. Comparing the Three Approaches

Each clustering method has its own personality. Choosing the right one depends on your data, your business question, and what you need from the results.

Feature	K-Means	Hierarchical	DBSCAN
Cluster Shape	Spherical only	Flexible (depends on linkage)	Arbitrary shapes
Specify # Clusters?	Yes (must choose k)	No (cut the tree later)	No (determined by density)
Handles Outliers?	No – forces all points into a cluster	Poorly – outliers distort merges	Yes – labels them as noise
Scalability	Excellent (large datasets)	Poor (small datasets only)	Good (medium to large)
Reproducibility	Varies (random start)	Always the same result	Always the same result
Ease of Explanation	Very easy	Moderate (dendrogram)	Moderate (density concept)

🎯 Quick Decision Guide

Use K-Means when you have large data and expect roughly even, round groups. Use Hierarchical when you have smaller data and want to explore the structure without committing to a specific k. Use DBSCAN when your clusters have irregular shapes, you expect outliers, or you're working with spatial/geographic data.

6. Clustering in Action: Industry Use Cases

The real power of clustering shows up when you apply it to specific business problems. Below are practical examples across industries your careers will touch. For each, we identify the clustering type best suited and explain why.

6.1 Sports Analytics

K-Means: Player Performance Segmentation

NBA and soccer teams use K-Means to segment players into performance tiers based on stats like points, assists, rebounds, and defensive metrics. For example, clustering NBA players by shooting efficiency and usage rate reveals groups such as “high-volume scorers,” “role players,” and “defensive specialists.” This directly informs trade decisions, contract negotiations, and game strategy.

Hierarchical: Training Load Profiling

Sports science teams use hierarchical clustering to group athletes by physiological responses to training, such as heart rate variability, sprint speed, and recovery time. The dendrogram helps coaches see which athletes respond similarly to workload, allowing them to design group-specific training programs rather than one-size-fits-all regimens.

DBSCAN: Spatial Play Pattern Analysis

In soccer and basketball, DBSCAN is used to analyze player movement data captured by GPS trackers. Because player positioning creates irregular, non-spherical clusters on the field, DBSCAN naturally finds “hot zones” where players tend to cluster during certain plays. Isolated movements that don’t belong to any pattern are flagged as noise, helping coaches identify both tendencies and anomalies.

6.2 Economics

K-Means: Country Economic Grouping

Economists use K-Means to classify countries by indicators like GDP per capita, inflation rate, unemployment, and trade balance. The resulting clusters (e.g., “developed economies,” “emerging markets,” “fragile states”) help international organizations like the World Bank tailor development programs, trade agreements, and aid allocation strategies.

Hierarchical: Industry Sector Similarity

Researchers use hierarchical clustering to analyze how economic sectors co-move during recessions. By clustering industries based on revenue volatility, employment changes, and GDP contribution, the dendrogram reveals which sectors are structurally linked—useful for understanding economic contagion and designing diversified industrial policy.

DBSCAN: Underground Economy Detection

Tax authorities use DBSCAN on transaction-level data to identify clusters of unusual economic activity that may indicate informal or underground economies. Because legitimate economic activity follows dense, regular patterns while illicit activity appears in scattered, low-density regions, DBSCAN naturally separates the two and flags noise points for investigation.

6.3 Accounting

K-Means: Expense Category Optimization

Large organizations use K-Means to cluster general ledger entries by amount, timing, frequency, and department to identify natural expense groupings. This helps controllers discover whether existing chart-of-account categories actually reflect real spending patterns, leading to more meaningful financial reporting and budget allocations.

Hierarchical: Audit Risk Tiering

Auditing firms cluster client engagements by risk characteristics: industry volatility, internal control quality, prior audit findings, and materiality thresholds. The hierarchical structure lets audit managers see which clients group together at different risk levels, enabling better allocation of senior auditors to high-risk engagements.

DBSCAN: Journal Entry Anomaly Detection

Forensic accountants apply DBSCAN to journal entry data to flag unusual postings. Normal entries form dense clusters around recurring transactions (payroll, rent, inventory purchases). Entries that fall outside these dense regions—such as large round-number entries posted at odd times—get classified as noise, making them candidates for fraud investigation.

6.4 Finance

K-Means: Portfolio Construction and Client Segmentation

Wealth management firms use K-Means to segment clients by risk tolerance, investment horizon, account size, and asset preferences. This allows advisors to build model portfolios for each segment rather than customizing every single account, dramatically improving operational efficiency while still delivering personalized investment strategies.

Hierarchical: Credit Risk Taxonomy

Banks use hierarchical clustering to build a taxonomy of borrower risk profiles based on credit scores, debt-to-income ratios, loan-to-value ratios, and payment history. The dendrogram shows natural groupings from “prime” to “subprime,” and the ability to cut at different heights lets risk managers set different thresholds for loan approval depending on market conditions.

DBSCAN: Fraudulent Transaction Detection

Financial institutions apply DBSCAN to real-time credit card transaction data. Normal spending patterns form dense clusters (grocery stores, gas stations, regular online purchases). Fraudulent transactions—foreign ATM withdrawals at 3 AM, rapid purchases across multiple cities—appear as outliers. DBSCAN labels these as noise, triggering automatic alerts for investigation.

6.5 Operations & Supply Chain

K-Means: Warehouse Location Optimization

Logistics companies use K-Means to cluster customer delivery addresses by geographic coordinates. The resulting centroids suggest optimal warehouse or distribution center locations that minimize total delivery distance. Amazon, FedEx, and UPS use variations of this approach to decide where to build fulfillment centers.

Hierarchical: Supplier Risk Assessment

Procurement teams cluster suppliers by performance metrics: on-time delivery rate, defect rate, price stability, and lead time consistency. The hierarchical view helps supply chain managers identify which suppliers are interchangeable (close in the dendrogram) and which are unique (isolated branches), directly informing sourcing diversification strategies.

DBSCAN: Manufacturing Defect Detection

Quality control teams in manufacturing apply DBSCAN to sensor readings from production lines (temperature, pressure, vibration, speed). Normal operation creates dense clusters of readings. Defective runs produce isolated data points that fall outside the normal density regions. DBSCAN naturally separates these, enabling real-time alerts when equipment drifts out of specification.

7. Quick-Reference: Use Cases at a Glance

Industry	Algorithm	Use Case	What You Cluster	Business Decision
Sports	K-Means	Player segmentation	Performance stats	Trades, contracts, strategy
	Hierarchical	Training load profiling	Physiological responses	Group-specific training
	DBSCAN	Play pattern analysis	GPS movement data	Tactical game planning
Economics	K-Means	Country classification	GDP, inflation, trade data	Policy and aid allocation
	Hierarchical	Sector co-movement	Revenue, employment data	Diversification policy
	DBSCAN	Underground economy	Transaction-level data	Tax enforcement targeting
Accounting	K-Means	Expense optimization	Ledger entries	Better cost allocation
	Hierarchical	Audit risk tiering	Client risk characteristics	Auditor assignment
	DBSCAN	Journal entry anomalies	Posting data	Fraud investigation
Finance	K-Means	Client segmentation	Risk, horizon, assets	Model portfolio design
	Hierarchical	Credit risk taxonomy	Borrower profiles	Loan approval thresholds
	DBSCAN	Fraud detection	Transaction patterns	Real-time fraud alerts
Ops / Supply Chain	K-Means	Warehouse placement	Delivery coordinates	Facility location decisions
	Hierarchical	Supplier risk assessment	Performance metrics	Sourcing diversification
	DBSCAN	Defect detection	Sensor readings	Real-time quality alerts

8. Key Takeaways

Before you apply clustering to any business problem, keep these principles in mind:

- **Start with the business question.** Don't just cluster because you can. Ask: What decision will these groups inform? Who needs to act on these segments?
- **Normalize your data.** Features measured on different scales will distort your clusters. Always standardize or normalize before running the algorithm.
- **There's no single "right" answer.** Clustering is exploratory. Different algorithms, different parameters, and different features will produce different groupings. The best solution is the one that's useful for your business.
- **Validate your clusters.** After clustering, profile each group. Do the clusters make business sense? Can you describe each segment to a non-technical stakeholder? If not, revisit your approach.
- **Watch out for the curse of dimensionality.** Distance becomes less meaningful in very high-dimensional data. Use feature selection or dimensionality reduction (like PCA) before clustering if you have many variables.
- **Outliers matter.** Decide upfront whether outliers are noise to be removed (DBSCAN handles this naturally) or important signals to be preserved (K-Means will force them into clusters).

The Bottom Line

Clustering isn't about the math—it's about finding actionable groups that lead to smarter business decisions. The best clustering solution is the one that gets deployed and creates value, not the one with the fanciest algorithm.

9. Check Your Understanding

Use these questions to test your grasp of the key concepts:

11. A marketing team has 200,000 customer records and needs to create 5 customer personas quickly. Which clustering algorithm would you recommend, and why?
12. Explain why normalizing features is critical before applying K-Means. What could go wrong if you skip this step?
13. A data scientist runs K-Means with k=3 three times and gets three different results. Why does this happen, and what should they do about it?
14. Your company's fraud detection team is analyzing credit card transactions. Some legitimate transactions are being forced into the "fraudulent" cluster by K-Means. Why might DBSCAN be a better choice for this problem?
15. A researcher is studying 50 genes and wants to see which ones have similar expression patterns at various levels of detail. Which algorithm would let them explore groupings at multiple resolutions? What visualization tool would help?

16. Compare single-linkage and complete-linkage in hierarchical clustering. How does each method define the distance between two clusters, and how does that affect the shape of the resulting clusters?
17. An operations manager wants to find optimal warehouse locations for a national retailer. Describe how K-Means could be applied and explain the role of centroids in this context.
18. Give an example from finance or accounting where DBSCAN's ability to identify outliers as noise provides a direct business advantage over K-Means.