

Business Data Analysis

MGM509

Lecture 5

Similarity and Segmentation: Neighbors,
Clusters, and Business Use Cases

Lecture
5

What we will cover

1. How to compute and apply similarity in data mining
2. The logic and mechanics of k-Nearest Neighbors (k-NN)
3. How clustering discovers natural groupings in data
4. Strengths and limitations of similarity-based approaches
5. Business use cases: targeting, recommendation, risk detection

Lecture
5

Recall from Lecture 1: 8 Canonical Data Mining Tasks (1/2)

Classification

- Assign labels to predefined categories (e.g., spam vs. not spam)

Regression

- Predict numeric outcomes (e.g., revenue, prices)

Clustering

- Group similar records (e.g., customer segmentation)

Similarity Matching

- Finding items/entities similar to a target (e.g., recommender systems)

Lecture
5

Recall from Lecture 1: 8 Canonical Data Mining Tasks (2/2)

Co-occurrence Grouping

- Discover patterns in frequently co-occurring items (e.g., market basket analysis)

Profiling

- Identify typical characteristics or behaviors of groups (e.g., loyal customers)

Link Prediction

- Anticipate connections between entities (e.g., friend recommendations)

Data Reduction

- Simplify datasets to essential variables (e.g., dimensionality reduction)

Lecture
05

Similarity, Distance, and Neighbors



Lecture
5

What Is Similarity-Based Reasoning?

- Core idea: similar items are likely to behave similarly
- Classification based on similarity is natural, intuitive, and effective.
- Instead of learning rules, models look for known similar examples
- Underlies techniques like:
 - k-Nearest Neighbors (k-NN)
 - Clustering
 - Case-based recommendation systems

Lecture
5

Why Similarity Matters in Data Science?

- Applications:
 - Recommendation engines: People who bought X also bought Y
 - Customer segmentation: Group similar behavior profiles
 - Anomaly detection: Detect points that aren't like others
 - Search relevance: Rank results based on closeness
- Assumption: “Things that are similar in the data representation are likely to be similar in behavior.”

Lecture 5

Types of Similarity-Based Tasks

Task Type	Goal	Example
Classification	Types of Similarity-Based Tasks	Predict if a customer will churn
Clustering	Group similar items together	Segment customers by usage pattern
Recommendation	Segment customers by usage pattern	Segment customers by usage pattern

**Lecture
5****Structured vs. Unstructured Similarity****Structured Data**

- Uses numeric or categorical features
- Similarity measured by distance or overlap
- Example: Similarity of customers based on tenure, MonthlyCharges

Unstructured Data:

- Text, images, audio → must be converted to structured form
- Use embeddings, tokenization, etc.
- Example: Similarity between product descriptions using text vectors

Lecture
5

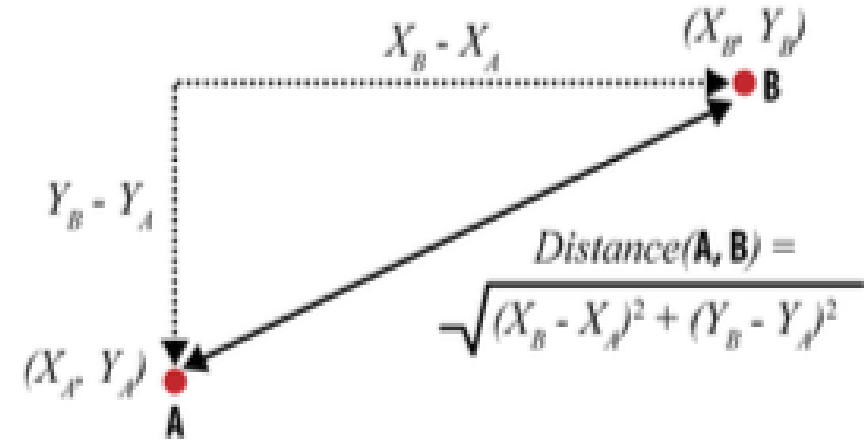
Measuring Similarity – Why It Matters

- Similarity (or its inverse, distance) is the foundation of:
 - Classification via neighbors (k-NN)
 - Grouping via clustering
 - Information retrieval and ranking
- Choosing the right similarity measure affects:
 - Which items are considered “nearby”
 - The accuracy and usefulness of predictions

Lecture
5

Euclidean Distance – The Default Metric

- Measures straight-line distance between two points in feature space
- Useful when distance in raw feature space correlates with similarity in outcome
- Formula: $d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$



Lecture
5

Manhattan Distance – A Simpler Alternative

- Also known as L1 norm or city-block distance
- Formula: $d(\mathbf{x}, \mathbf{y}) = \sum_i^n |x_i - y_i|$
- Use Cases:
 - When data has many categorical or ordinal features
 - More robust to outliers than Euclidean
- Example: Predicting churn based on count-like features such as number of calls, complaints, etc.

Lecture
5Cosine Similarity – Comparing Direction,
Not Magnitude

- Measures angle between two vectors, not their distance
- Cosine similarity focuses on pattern, not size
- Formula: $\cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$
- Best for high-dimensional sparse data and text analysis (e.g., comparing customer reviews or documents)
- Interpretation: 1 = identical direction, 0 = orthogonal (unrelated), -1 = opposite direction

Lecture
5

Handling Categorical Variables in Similarity

- You cannot directly use distance formulas on text or categories
- Options:
 - One-hot encoding (for nominal values)
 - Ordinal encoding (if order matters)
 - Matching coefficient (for binary variables)
- Jaccard similarity: $\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$
- Ex.: Comparing customers based on plan type, device type, region

Lecture
5

The Importance of Scaling and Normalization

- In similarity-based learning, irrelevant or unscaled features can mislead the model
- Features with larger ranges dominate distance calculations
- **Solutions:**
 - Min-max scaling: transforms values to $[0, 1]$
 - Z-score normalization: centers values (mean = 0, std = 1)
- Always normalize when using distance-based methods

Lecture
5Introducing the k-Nearest Neighbors
(k-NN) Algorithm

- A non-parametric, instance-based learning method
- To classify a new example:

Measure its
distance to every
training point



Find the k
closest (nearest)
examples

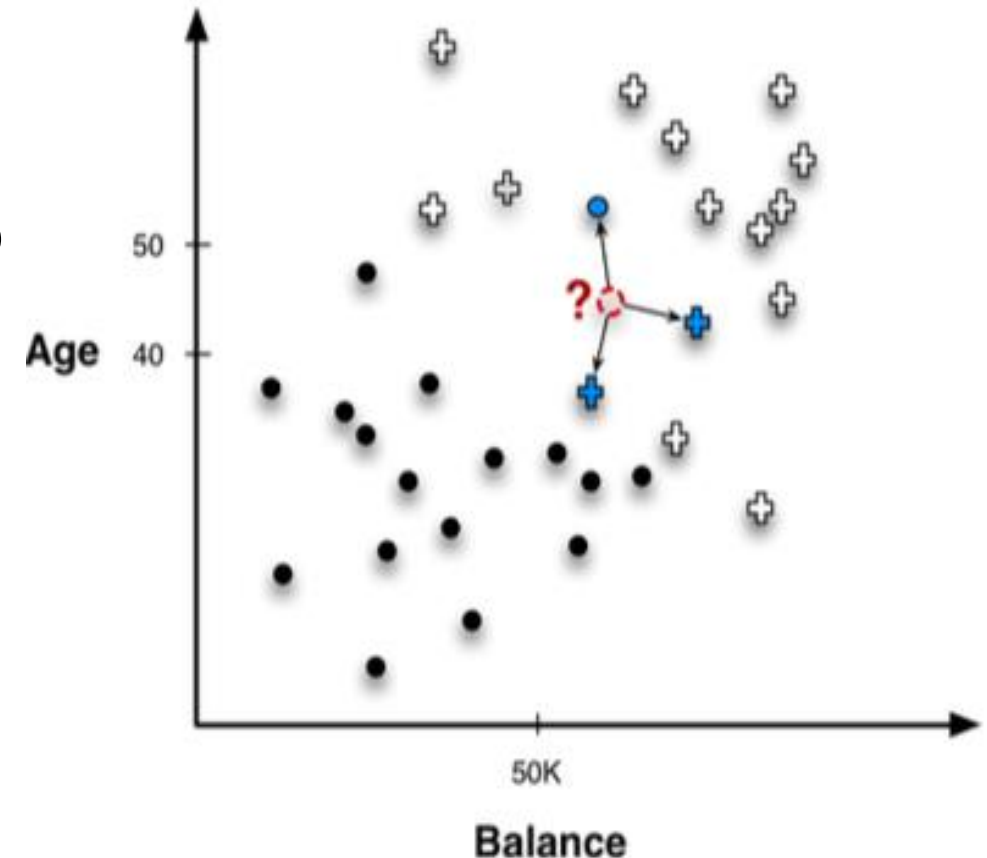


Predict the most
common label
among them

Lecture
5

Intuition Behind k-NN

- No training phase — the model is just stored examples
- The model “learns” by comparing to what it has seen before
- Assumption: Similar inputs have similar outputs
- Example: If most of the 5 closest customers churned, this one probably will too.



Lecture
5

Choosing the Right Value of K

Value of k	Behavior
Low (e.g., 1–3)	Sensitive to noise (overfitting)
High (e.g., 15–20)	Smooths decision boundary (underfitting)

- Optimal k balances bias and variance
- Typically selected using cross-validation

Too small and the model memorizes. Too large and it becomes vague.

Lecture
5

Weighted vs. Unweighted Voting

- Basic k-NN: All neighbors contribute equally
- Weighted k-NN: Closer neighbors contribute more
- **Weighting schemes:**
 - Inverse distance weighting
 - Gaussian kernel-based weighting
- Benefit: Helps reduce impact of borderline or distant examples

Lecture
5

Limitations of k-NN

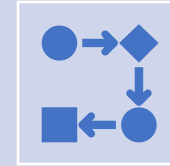
Issue	Explanation
Slow at prediction	Must compute distance to every point
Sensitive to irrelevant features	No built-in feature weighting
High memory use	Stores entire training set
Curse of dimensionality	Similarity breaks down in high dimensions

Lecture
5

When k-NN Works Well



You have a moderate-sized dataset



Features are clean, scaled, and meaningful



Classes are clearly clustered in feature space



You want a simple, interpretable baseline

Lecture
5

Strengths of k-Nearest Neighbors (k-NN)

Simple &
Intuitive: Easy to
understand and
explain

No training phase:
Just store the data

Flexible decision
boundaries:
Works well with
non-linear data

Naturally adapts
as more data is
added

- Use Cases:
 - Customer churn prediction
 - Product recommendation (based on user similarity)
 - Risk assessment based on historical patterns
- k-NN is a perfect example of reasoning by analogy

**Lecture
5****Weaknesses of k-NN**

Challenge	Description
Scalability	Slow with large datasets ($O(n)$ prediction time)
Memory-intensive	Must store and search full dataset
Noise sensitivity	A few bad neighbors can mislead classification
Needs scaling	Distorted by features on different scales

Best used on smaller, well-prepared datasets or with performance optimizations (e.g., KD-trees)

Lecture
5

Curse of Dimensionality

- In high-dimensional space, all points tend to become equally distant and distances lose their discriminatory power.
- Consequences: Similarity-based reasoning breaks down and k-NN struggles to find meaningful neighbors
- **Mitigation:**

Dimensionality
reduction
(PCA)

Feature
selection

Distance
metric
learning

Lecture
5

Best practices for using k-NN effectively:



Scale features
(min-max or z-
score
normalization)



Remove
irrelevant
features (feature
selection)



Use weighted
voting if
neighbors vary
in proximity



Tune k using cross-
validation



Apply dimensionality
reduction (PCA,
autoencoders) for
high-dimensional data



Use approximate
neighbors or
indexing (e.g., Ball
Trees) for speed



Cleaning your data is as
important as choosing
the right k

Lecture
5Real-World Business Use Case – Customer
Churn

- **Goal:** Predict which customers are likely to churn based on their similarity to past churners
- **Steps:**

Represent customers by features (e.g., monthly charges, contract type, usage)



For a new customer, find k most similar past customers



If majority of neighbors churned → flag as high risk

- **Benefits:**
 - Transparent logic: “We’re concerned because similar customers left”
 - Useful in marketing, customer retention, telecom, insurance

Lecture
05

Clustering



Lecture
5

What Is Clustering?

- Clustering is an unsupervised learning technique that:
 - Finds natural groupings in unlabeled data
 - Assigns each data point to the most similar group (cluster)
 - Uncovers structure where no labels exist.”
- **Real-world use:**
 - Segmenting customers by behavior
 - Detecting fraud or anomalies
 - Organizing large volumes of unstructured data

Lecture
5

Introducing k-Means Clustering

- k-Means groups data into k clusters based on similarity
- Each point belongs to the cluster with the closest centroid
- **Objective:** minimize within-cluster distance
- Algorithm steps:

Initialize k
centroids
randomly



Assign each
point to nearest
centroid



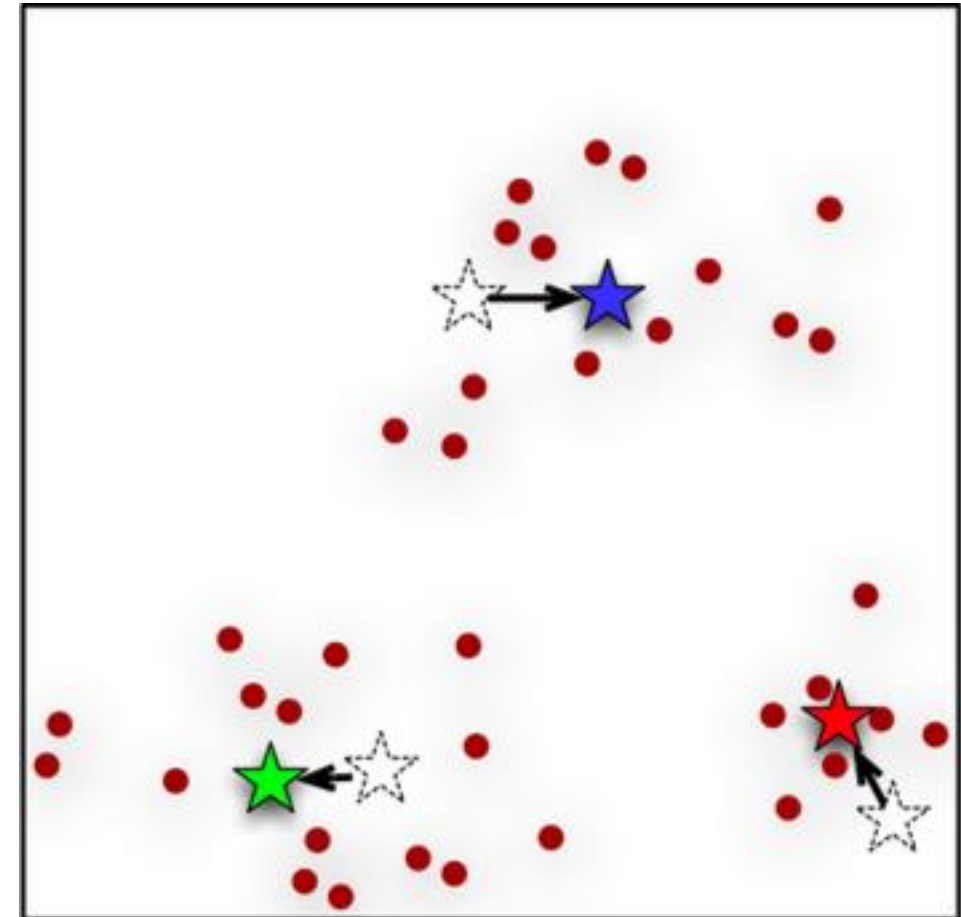
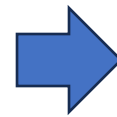
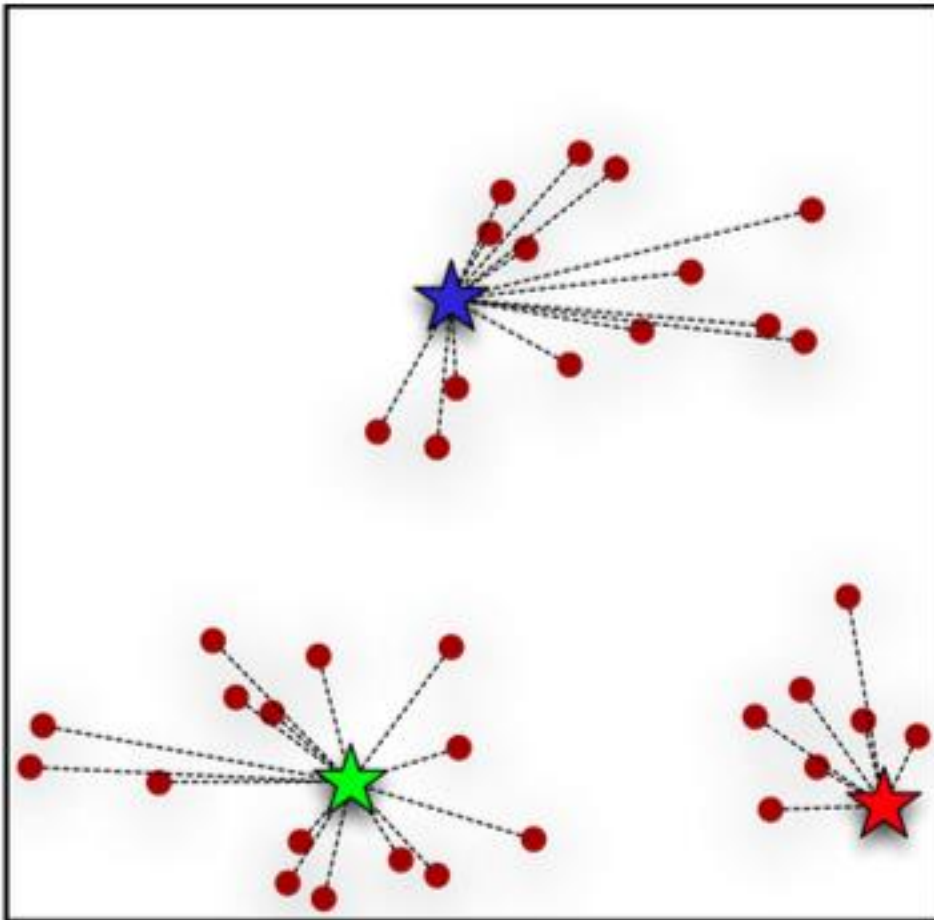
Recompute
centroids from
assigned points



Repeat until
convergence

Lecture
5

Introducing k-Means Clustering



Lecture
5

Choosing the Right Number of Clusters (k)

- How do we decide k? Try multiple values and evaluate performance
- Method: Elbow Technique
 - Plot number of clusters (k) vs. total within-cluster variance
 - The “elbow” point = where adding more clusters stops yielding big improvements
- Business context helps interpret what clusters mean

Lecture 5

Cluster Evaluation Metrics

Metric	Purpose
Within-cluster distance	Tightness of clusters
Between-cluster distance	Separation of different groups
Silhouette score	Measures how well points fit their own cluster vs. others

- Good clusters:
 - High intra-cluster similarity
 - Low inter-cluster similarity
- Clusters must be meaningful — not just mathematically neat

**Lecture
5****Other Clustering Algorithms**

- **Hierarchical Clustering**
 - Builds a tree (dendrogram) of nested clusters
 - Useful when the number of clusters is unknown
- **DBSCAN (Density-Based)**
 - Groups points in dense regions
 - Detects outliers naturally
- **When to avoid k-Means:**
 - Non-spherical clusters
 - Varying density
 - Presence of noise/outliers

Lecture
5

Customer Segmentation via Clustering

- Group customers based on:
 - Demographics
 - Usage patterns
 - Product preferences
- Use case:
 - Tailored marketing
 - Targeted product design
 - Personalized offers
- **Example:** k-Means used to create 5 customer segments in a telecom company — each with distinct churn rates.

Clustering reveals differences that can drive better decisions.

Lecture
5

Limitations of Clustering

Challenge	Description
Subjectivity	Different algorithms → different clusters
Sensitivity	Initial conditions can skew results
Lack of labels	Hard to evaluate objectively
Unstable outcomes	May vary with feature scaling or noise

Clusters don't always mean something real — always interpret them in context.

Lecture
5

Business Use Case – Customer Segmentation

- A telecom company segments customers into:
 - Heavy data users
 - Price-sensitive subscribers
 - Loyal, long-tenure users
- Impact:
 - Personalized offers
 - Improved customer retention
 - Focused marketing campaigns

Lecture
5

Recommendation Systems Using Neighbors

- Two main types:
 - **User-based:** Recommend what similar users liked → “People like you also watched...” (Netflix),
 - **Item-based:** Recommend items similar to what the user liked → “Customers who bought this also bought...” (Amazon)
- **Method:**
 - Use cosine similarity or collaborative filtering
 - Find nearest neighbors to generate recommendations

Lecture
5Fraud Detection with Clustering and
Neighbors

- Fraud often appears as outliers or rare patterns in transaction data
- k-NN and clustering help detect anomalies by comparing with normal behavior
- **Techniques:**
 - Cluster legitimate behavior → detect deviations
 - Use k-NN to find transactions that don't resemble past records
- **Benefit:** Unsupervised methods help when labeled fraud data is scarce
- “Fraud is different — and difference is what clustering finds.”

Lecture
5

Medical Diagnosis via Similarity Matching

- In healthcare, similarity-based techniques are used to:
 - Match symptoms to known cases
 - Group patients for treatment comparison
 - Identify rare disease profiles
- Example: A patient's record is compared to historical cases using k-NN. Neighbors' diagnoses or outcomes guide recommendations
- Why it works: Medical history and outcomes often correlate closely with pattern similarity)

Lecture
5

Social Network & Web Analysis

- Similarity and clustering can reveal:
 - Communities within networks
 - Influencers and behavior clusters
 - Topic groups in forums or content streams
- Use Cases:
 - Detect fake accounts or botnets
 - Recommend connections (LinkedIn, Facebook)
 - Cluster posts/comments with similar themes
- These techniques scale well and adapt to unstructured or semi-structured data.

Lecture
5Checklist – Before Using Similarity-
Based Models

- ☐ Are features scaled and cleaned appropriately?
- ☐ Are irrelevant or noisy features removed?
- ☐ Is the similarity measure appropriate for your data type?
- ☐ Have you validated the model using holdout or cross-validation?
- ☐ Do the clusters or neighbors make sense in your business context?

Similarity-based models are powerful — but demand careful
preprocessing and interpretation

**Lecture
4****Key Points****Three fundamental ideas:**

- **1. Similarity** – A measurable notion of “closeness” in data
- **2. Neighbors (k-NN)** – Making predictions based on nearby past cases
- **3. Clusters** – Finding natural groupings without predefined labels

Key Principle:

- **“Similar cases are likely to have similar outcomes”** – a powerful idea across prediction, detection, and personalization

Techniques:

- **k-NN**: You have labeled data and want interpretable predictions
- **k-Means**: You want to group customers or items based on behavior
- **DBSCAN**: You need to find outliers or dense clusters
- **Cosine Similarity**: You’re working with text or sparse data

Choice of **similarity metric** and **feature preparation** strongly influences results

For Discussion Panel and Next Lecture:

- Chapter 6 “Similarity, Neighbors, and Clusters”
- Use the open-source Telco Customer Churn dataset available on [Kaggle](#).
- **Use K-Means to cluster the data into 5 different Groups.**
- **What are the characteristics of each group?**

**Lecture
5**



**For Discussion Panel
and Next Lecture**



Thank You