

Machine Learning for Smarter Innovation

Week 1: Foundations & Clustering

Discovering Innovation Patterns with ML

BSc Course in AI-Enhanced Innovation

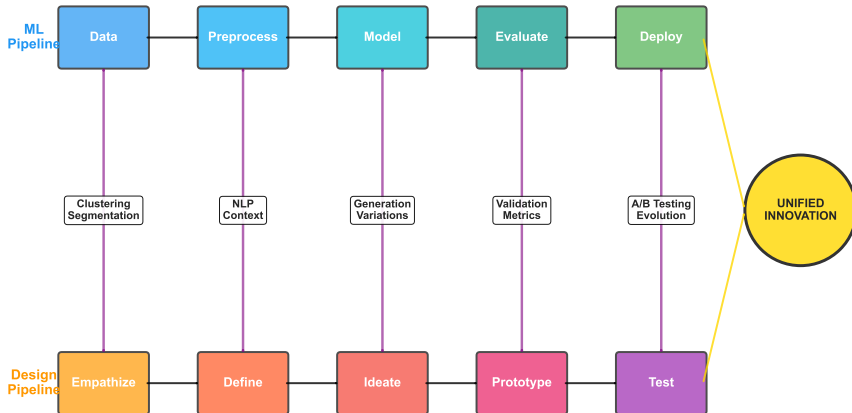
Machine Learning + Innovation + Design Thinking

The Power of Convergent Methodologies

The Unified Innovation Pipeline

Where Technology Amplifies Human Creativity

Technical Mastery



PART 1

Foundation & Context

What we'll explore:

- Why traditional design hits limits
- How ML amplifies human insight
- The dual pipeline approach
- Your learning journey ahead

Setting the stage for transformation

Part 1: Learning Objectives

What You'll Master in This Section

By the end of Part 1, you will:

- **Understand** the limitations of traditional innovation approaches
- **Recognize** how ML enhances human creativity
- **Explain** the dual pipeline methodology
- **Navigate** the 10-week learning journey
- **Identify** Week 1's role in the overall course

Success Criteria

- Can articulate 3+ traditional design limitations
- Can describe ML's value proposition
- Can map ML pipeline to design pipeline
- Understand clustering's role in innovation

The Innovation Challenge

Why Traditional Design Needs AI Enhancement

Traditional Design Limits

- **Scale:** Can analyze 50 ideas, not 50,000
- **Speed:** Months for insights
- **Bias:** Designer's perspective dominates
- **Patterns:** Miss hidden connections
- **Iteration:** Slow feedback loops

AI-Enhanced Innovation

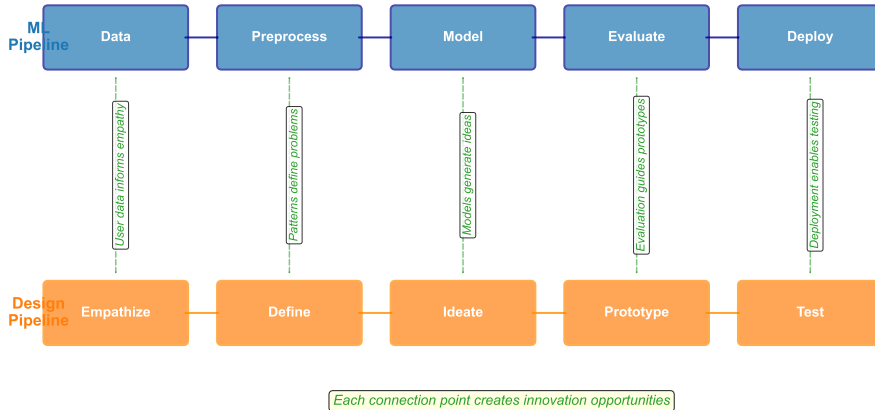
- **Scale:** Analyze millions of data points
- **Speed:** Real-time insights
- **Objectivity:** Data-driven discovery
- **Patterns:** Find non-obvious relationships
- **Iteration:** Continuous learning

The Promise: 100x more insights, 10x faster innovation

The Dual Pipeline

Where ML Meets Design Thinking

The Convergence: ML Meets Design Thinking



The Dual Pipeline (Continued)

Understanding Both Worlds

ML Pipeline

Data → Preprocess → Model → Evaluate → Deploy

- Collect innovation data
- Clean and transform
- Train algorithms
- Validate accuracy
- Scale to production

Design Pipeline

Empathize → Define → Ideate → Prototype → Test

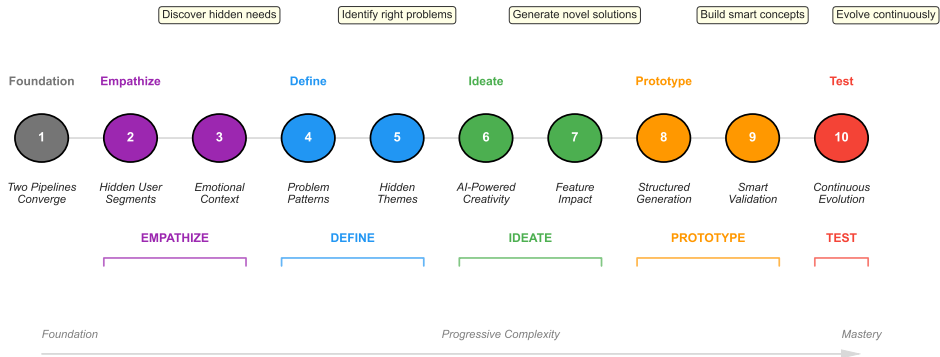
- Understand innovation needs
- Frame problems
- Generate solutions
- Build concepts
- Validate innovation impact

Integration = Innovation at Scale

Your Innovation Journey

10 Weeks to AI-Powered Design Mastery

10-Week Innovation Journey



Your Innovation Journey (Continued)

What You'll Master in Each Stage

Stage	Weeks	Innovation Unlocked
Discover	1-2	Find hidden innovation opportunities
Define	3-4	Identify the right problems to solve
Ideate	5-6	Generate novel solutions with AI
Prototype	7-8	Build smart, adaptive concepts
Test	9-10	Evolve through continuous learning

This Week: Clustering for Innovation Pattern Discovery

Week 1: Clustering for Innovation

From Scattered Ideas to Innovation Patterns

What We'll Learn:

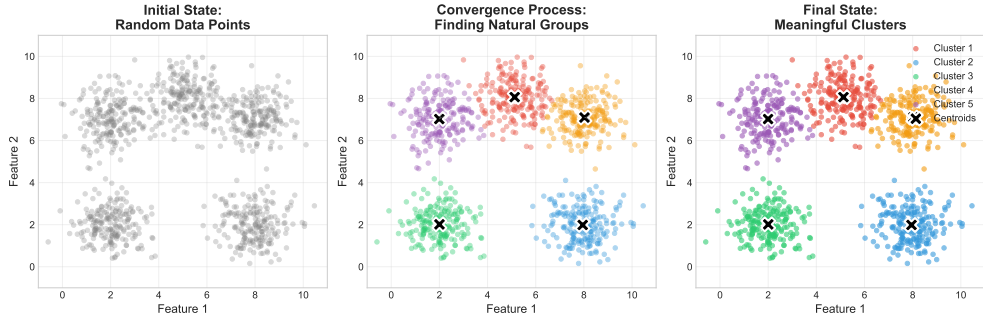
- How clustering reveals innovation categories
- K-means algorithm fundamentals
- Finding the optimal number of clusters
- Quality metrics for validation
- Advanced clustering techniques

Design Applications:

- Create innovation archetypes
- Map innovation evolution paths
- Identify opportunities systematically
- Prioritize design efforts
- Scale analysis to thousands of ideas

Goal: Transform scattered ideas into innovation patterns

The Convergence Flow: From Chaos to Clarity



The Convergence Flow: Order from Chaos
Watch 5000 innovation ideas self-organize into meaningful patterns

Now Let's Get Technical

From Understanding the Problem to Finding Solutions

We've seen the challenge:

Thousands of innovation ideas with hidden connections

Traditional approach:

Manual segmentation based on demographics

The ML solution:

Let the data reveal its own natural groups

Enter: Clustering Algorithms

PART 2

Technical Core

What we'll master:

- K-means clustering algorithm
- Finding optimal K with elbow method
- Distance metrics and quality measures
- Advanced techniques (DBSCAN, Hierarchical)
- Feature importance analysis

Building your ML toolkit

Part 2: Learning Objectives

Technical Skills You'll Develop

By the end of Part 2, you will:

- **Implement** K-means clustering
- **Apply** the elbow method
- **Calculate** distance metrics
- **Evaluate** cluster quality
- **Compare** different algorithms
- **Select** appropriate methods

Hands-on Skills

- Code K-means from scratch
- Interpret silhouette scores
- Choose between algorithms
- Tune parameters effectively
- Handle different data shapes
- Preprocess data correctly

The Innovation Classification Problem

5000 Ideas - How Do They Connect?

The Pain

Current Reality:

- One-size-fits-all solutions
- Generic innovation categories
- Missed opportunities
- Unhappy edge cases

The Cost:

- Most innovations get misclassified
- Features with low adoption rates
- Inefficient resource allocation

The Question

What if we could...

- Find natural innovation clusters?
- Discover innovation patterns?
- Innovate at scale?
- Identify opportunity gaps?

We can!

Solution: Clustering

What is Clustering?

Finding Natural Groups in Innovation Data

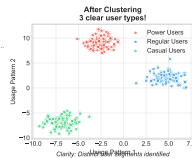
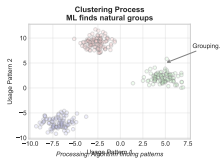
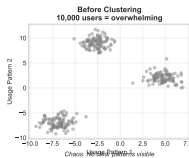
Clustering Finds:

- Natural groupings
- Similar approaches
- Hidden patterns
- Innovation relationships

Key Insight:

Innovations with similar features address similar opportunities

From Chaos to Clarity Through Clustering



K-Means: The Workhorse Algorithm

How It Organizes Your Innovations

The Process:

- 1 Choose K (number of clusters)
- 2 Place K random centroids
- 3 Assign points to nearest centroid
- 4 Move centroids to cluster mean
- 5 Repeat until stable

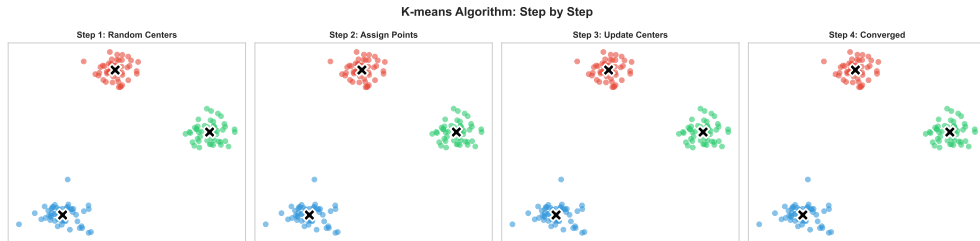
Strengths:

- Fast and scalable
- Easy to understand
- Works well for spherical clusters



K-Means in Action

Step-by-Step Convergence



Iteration 1 → Iteration 3 → Iteration 5 → **Converged**

The Goldilocks Problem

Too Few vs. Too Many Groups

Too Few (K)

Oversimplification

- Mixed segments
- Lost nuance
- Generic solutions

Just Right (K)

Optimal Balance

- Clear segments
- Actionable insights
- Manageable complexity

Too Many (K)

Analysis Paralysis

- Overfitting
- Tiny segments
- Impossible to act on

How do we find the sweet spot?

The Elbow Method

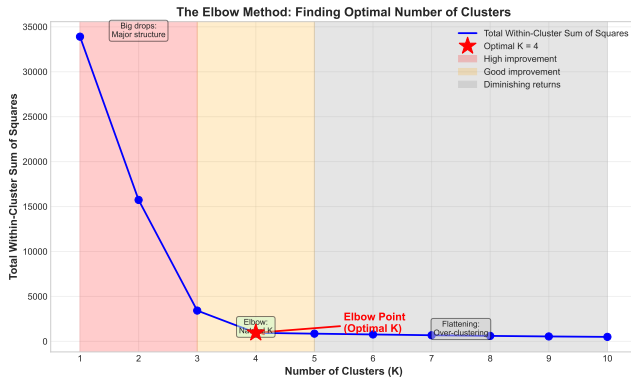
Finding the Optimal Number of Clusters

Finding the Elbow:

- Plot inertia vs K
- Look for the “elbow”
- Balance between:
 - Too few: Mixed groups
 - Too many: Overfitting

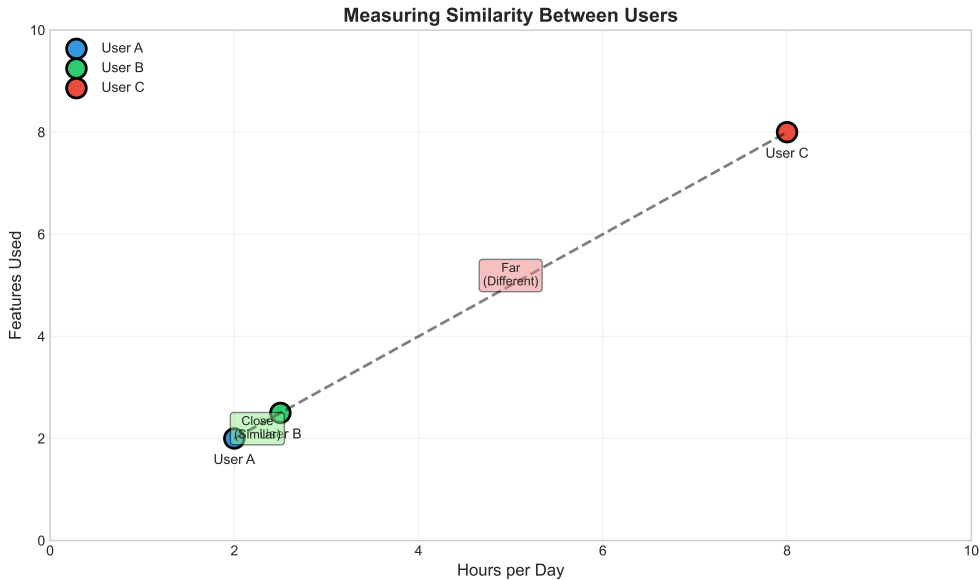
Optimal K = 5

Best trade-off between simplicity and accuracy



Distance Metrics

How We Measure Similarity



Cluster Quality Metrics

How Good Are Your Groups?

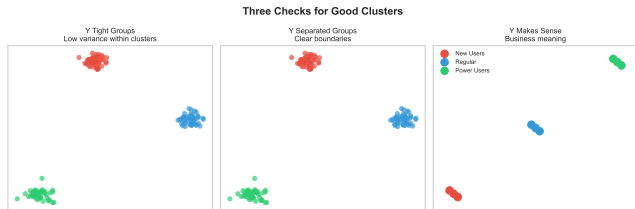
Silhouette Score:

- Ranges from -1 to +1
- Higher = better separation
- Our score: **0.73**

What it measures:

- Within-cluster cohesion
- Between-cluster separation
- Overall cluster validity

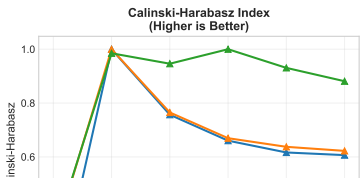
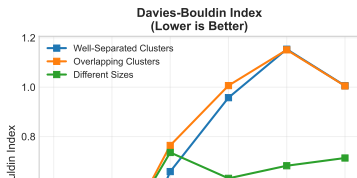
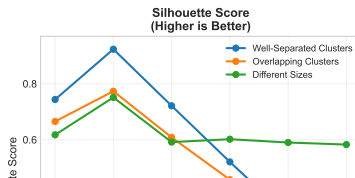
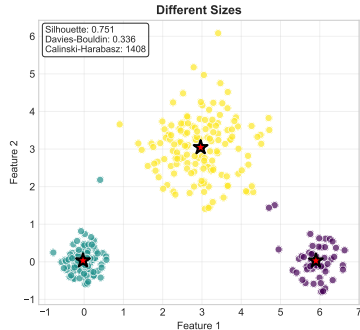
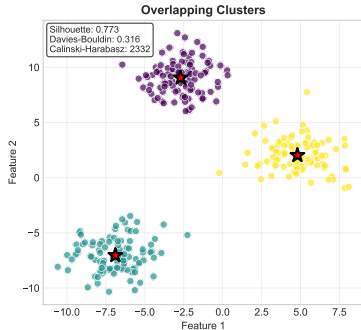
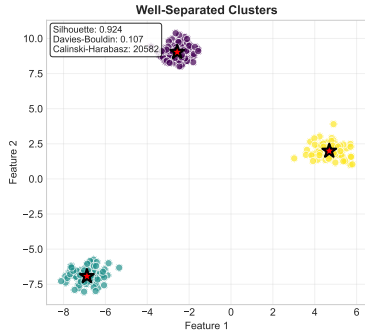
0.73 = Strong clusters!



Comparing Evaluation Metrics

Different Metrics for Different Data Patterns

Clustering Evaluation Metrics Comparison How Different Metrics Behave on Various Data Patterns



K-Means Assumes Spherical Clusters

But what about:

- Innovations connected through technology stacks
- Domain-specific innovation clusters
- Evolution patterns (incremental, disruptive)
- Outliers and noise points

K-Means Forces Round Pegs into Round Holes

Solution: Density-Based Clustering

DBSCAN: Density-Based Clustering

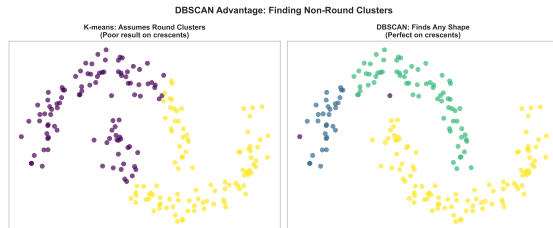
Finding Natural Boundaries, Not Forcing Shapes

DBSCAN Advantages:

- No need to specify K
- Finds arbitrary shapes
- Identifies outliers
- Handles noise well

Perfect for:

- Non-spherical patterns
- Varying densities
- Outlier detection
- Exploratory analysis

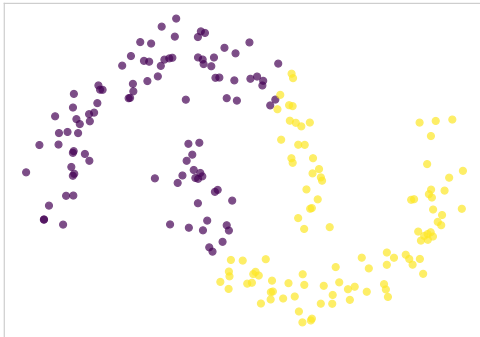


DBSCAN: Complex Patterns

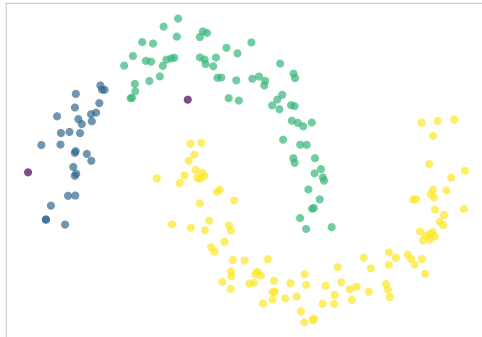
When K-Means Isn't Enough

DBSCAN Advantage: Finding Non-Round Clusters

K-means: Assumes Round Clusters
(Poor result on crescents)



DBSCAN: Finds Any Shape
(Perfect on crescents)



K-Means: Forces spherical shapes — DBSCAN: Finds natural boundaries

Choosing the Right Algorithm

Comparison of Clustering Methods

Algorithm	Speed	Shape	Outliers	Params	Best For
K-Means	Fast $O(nkt)$	Spherical clusters	Sensitive	K only	Quick segments
DBSCAN	Medium $O(n \log n)$	Any shape	Robust (detects)	eps, MinPts	Complex shapes
Hierarchical	Slow $O(n^2)$	Any shape	Moderate	Distance threshold	Multi-level analysis
GMM	Medium $O(nkt)$	Elliptical clusters	Moderate	K, covariance	Overlapping groups

Choose K-Means when:

- Speed is critical
- Clusters are roughly equal size
- You know K in advance

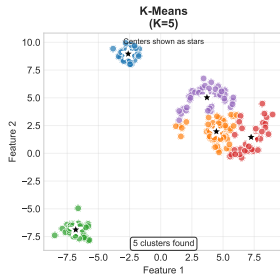
Choose DBSCAN when:

- Clusters have irregular shapes
- Outliers need identification
- Density varies across data

Algorithm Visual Comparison

Same Data, Different Approaches

Clustering Algorithms Visual Comparison Same Data, Different Approaches

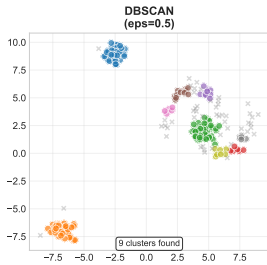


K-Means (K=5)

- ☐ Fast and scalable
- ☐ Spherical clusters
- ☐ Fixed K required
- ☐ Sensitive to outliers

Best for: Quick segmentation
with known cluster count

Complexity: $O(nk)$

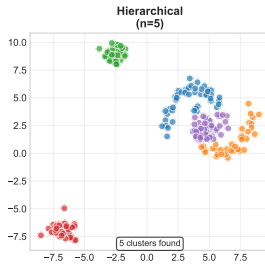


DBSCAN (eps=0.5)

- ☐ Finds arbitrary shapes
- ☐ Identifies outliers
- ☐ No K needed
- ☐ Sensitive to parameters

Best for: Anomaly detection
and irregular patterns

Complexity: $O(n \log n)$

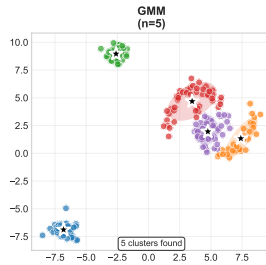


Hierarchical (n=5)

- ☐ Dendrogram output
- ☐ No K needed initially
- ☐ Interpretable
- ☐ Computationally expensive

Best for: Taxonomies and
exploring relationships

Complexity: $O(n^2)$



GMM (n=5)

- ☐ Soft assignments
- ☐ Elliptical clusters
- ☐ Probabilistic
- ☐ Assumes Gaussian distribution

Best for: Overlapping groups
and uncertainty modeling

Complexity: $O(nk)$

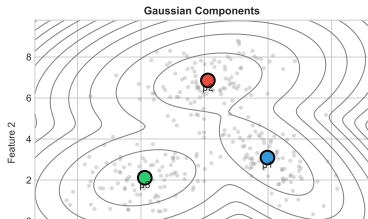
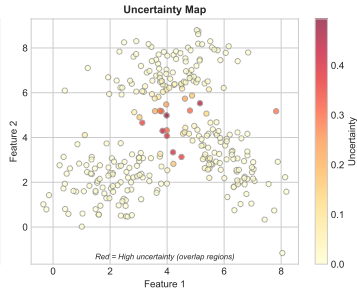
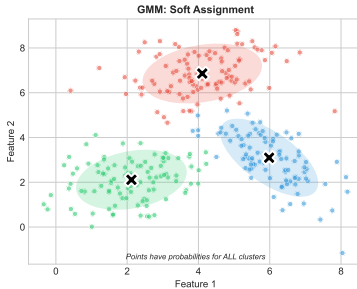
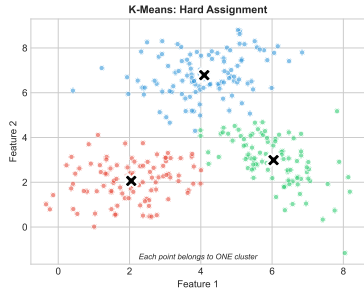
Dataset: Mix of 3 Gaussian blobs and 2 mean-changed clusters (350 points total)

Gaussian Mixture Models (GMM)

Soft Clustering for Overlapping Innovation Categories

Gaussian Mixture Models (GMM): Soft Clustering for Innovation

Beyond Hard Boundaries: Probabilistic Innovation Classification



GMM vs K-means

GMM Advantages:

- Soft assignments (probabilities)
- Captures cluster shape (elliptical)
- Handles overlapping clusters
- Provides uncertainty estimates
- Models data generation process

K-means Advantages:

- Faster computation
- Simpler interpretation
- Less parameters
- More stable results
- Works well for spherical clusters

When to use GMM:

- Overlapping innovation categories
- Noisy data

Innovation Category Probabilities

Innovation	Tech	Service	Social
AI Assistant	0.85	0.10	0.05
Sharing Platform	0.30	0.45	0.25
Green Energy	0.60	0.15	0.25
Digital Health	0.40	0.50	0.10

Fixed K Gives One View

But real relationships are hierarchical:

- Organization: Company → Department → Team → Individual
- Geography: Country → Region → City → Neighborhood
- Products: Category → Subcategory → Brand → SKU
- Innovations: All → Categories → Sub-types → Specific solutions

K-means: Pick 5 groups and that's it

What if we need flexibility?

Solution: See the full hierarchy, cut where needed

Hierarchical Clustering

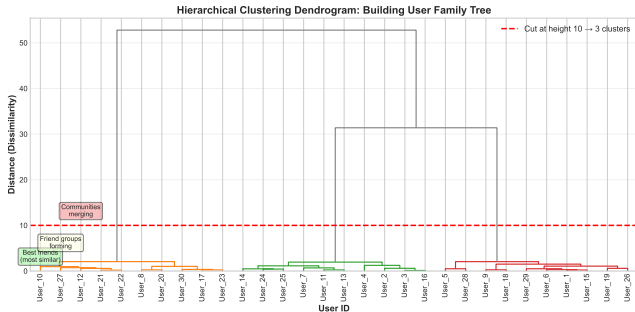
Building a Tree of Relationships

Dendrogram Benefits:

- Shows cluster hierarchy
- Multiple granularities
- Natural relationships
- No preset K needed

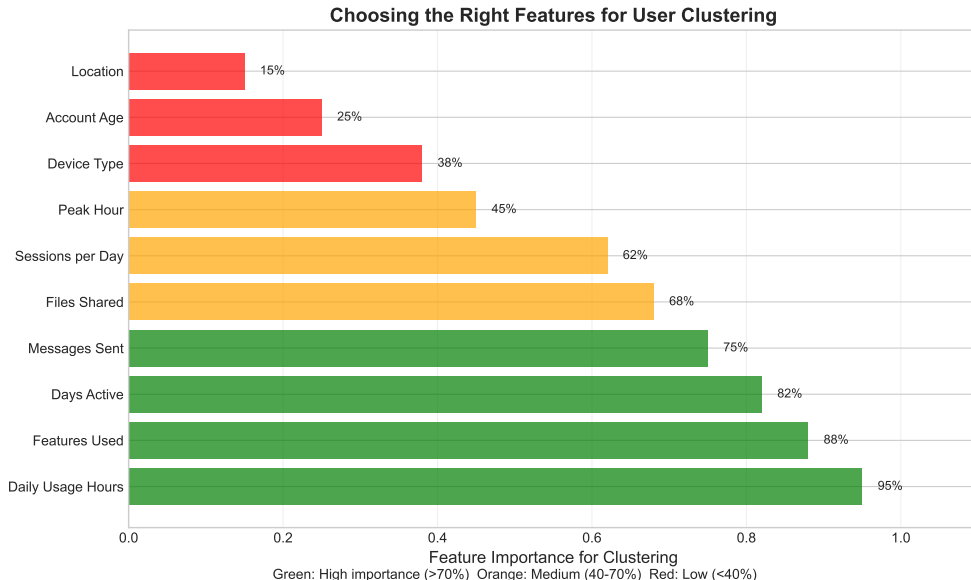
Cut the tree at any level:

- High cut = Few clusters
- Low cut = Many clusters
- Choose based on needs



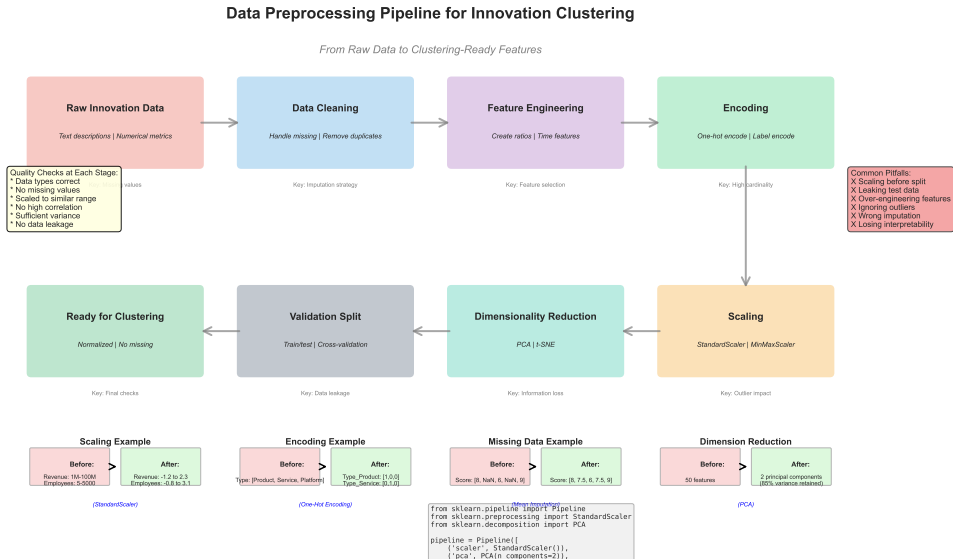
What Drives the Clusters?

Feature Importance Analysis



Data Preprocessing Pipeline

From Raw Data to Clustering-Ready Features



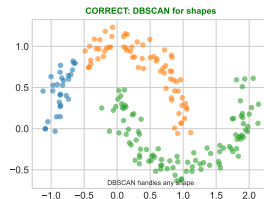
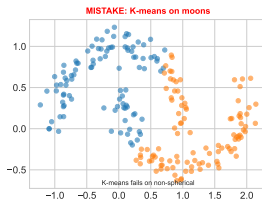
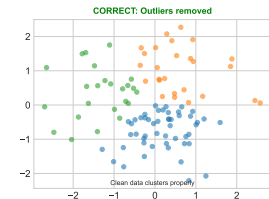
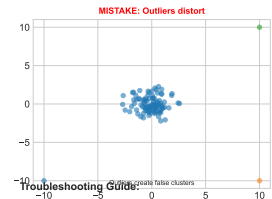
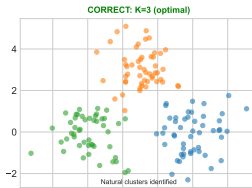
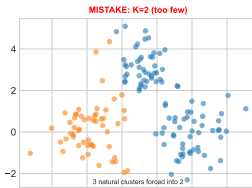
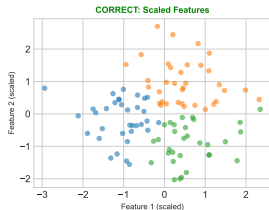
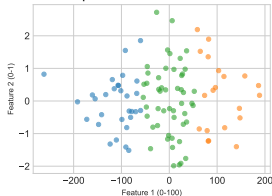
Common Mistakes & Troubleshooting

Learn from These Pitfalls

Common Clustering Mistakes & Troubleshooting Guide

Learn from These Mistakes to Master Clustering

Visual Examples of Common Mistakes:



Troubleshooting Guide:

Problem

Poor separation

GOLDEN RULES:
1. Always scale your features

Symptoms

Low silhouette score

Difficult to merge

WARNING SIGNS:

Solution

Try different K or algorithm

Get random state, increase n_init

Prevention

Use elbow method

SUCCESS INDICATORS:

silhouette

Parameter Tuning Guidelines

Recommended Ranges and Best Practices

Clustering Parameter Tuning Guidelines

Recommended Ranges, Methods, and Best Practices

K-Means

Parameter	Range	Default	Tuning Method
n_clusters (K)	2-10	3-5	Elbow/Silhouette
init	['k-means++', 'random']	k-means++	Always k-means++
n_init	10-100	10	More for stability
max_iter	100-1000	300	Increase if no convergence
tol	1e-6 to 1e-2	1e-4	Smaller for precision

DBSCAN

Parameter	Range	Default	Tuning Method
eps	0.01-2.0	0.5	k-distance plot
min_samples	3-20	2*dim	Domain knowledge
metric	['euclidean', 'manhattan']	euclidean	Data dependent
algorithm	['auto', 'ball_tree']	auto	Auto is fine
leaf_size	10-50	30	Memory vs speed

GMM

Parameter	Range	Default	Tuning Method
n_components	2-10	3-5	BIC/AIC
covariance_type	['full', 'diag', 'full_spherical']	full	Start full, simplify
max_iter	50-500	100	Monitor convergence
n_init	1-10	1	More for stability
init_params	['kmeans', 'random']	kmeans	kmeans faster

Tuning Strategies

Grid Search

Pros: Exhaustive, Reproducible, Simple

Cons: Slow, Curse of dimensionality

Use when: Small parameter space

Random Search

Pros: Faster, Better for many params, Parallelizable

Cons: May miss optimum, Not reproducible

Use when: Large parameter space

Bayesian Opt

Pros: Efficient, Learns from history, Fewer iterations

Cons: Complex, Overhead for simple problems

Use when: Expensive evaluations

Validation Metrics

Metric	Range	Interpretation	Use For
Silhouette	[-1, 1]	Higher is better	General quality
Davies-Bouldin	[0, ∞)	Lower is better	Cluster separation
Calinski-Harabasz	[0, ∞)	Higher is better	Dense clusters
Inertia	[0, ∞)	Lower is better	K-means only
BIC/AIC	(-∞, ∞)	Lower is better	GMM selection

Tuning Best Practices

1. Start with defaults, then tune
2. Use cross-validation when possible
3. Consider computational budget
4. Log all experiments
5. Visualize parameter effects
6. Use domain knowledge
7. Check stability across runs
8. Don't overfit to metrics

IMPORTANT:
No metric is perfect!
Always validate with:
• Visual inspection
• Domain expertise
• Business goals

We've mastered the technical tools:

Clustering, metrics, quality measures

But clusters are just numbers...

Until we connect them to innovation opportunities

Let's transform data into innovation insights

Each cluster represents innovation opportunities and patterns

PART 3

Innovation Pattern Analysis

What we'll create:

- Data-driven innovation archetypes
- Innovation pattern maps per category
- Cluster-specific journeys
- Opportunity heat maps
- Design priority matrices

Where ML reveals innovation patterns

Part 3: Learning Objectives

Innovation Applications You'll Master

By the end of Part 3, you will:

- **Create** innovation archetypes
- **Map** innovation patterns
- **Design** opportunity matrices
- **Analyze** innovation lifecycles
- **Build** ecosystem maps
- **Prioritize** innovation efforts

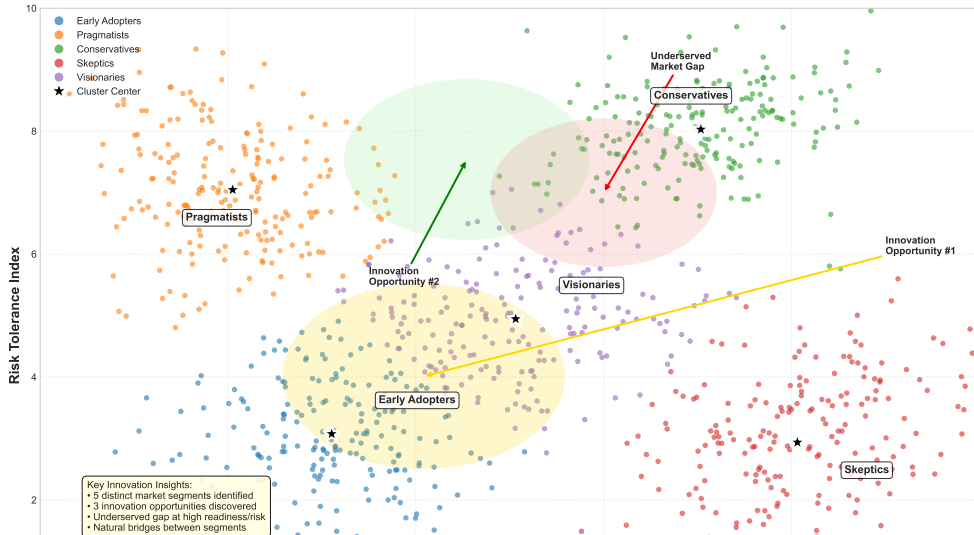
Design Outcomes

- Innovation taxonomy framework
- Cluster-based strategies
- Data-driven prioritization
- Opportunity identification
- Pattern recognition skills
- Ecosystem understanding

From Data Points to Innovation Insights

Bridging the Technical-Human Gap

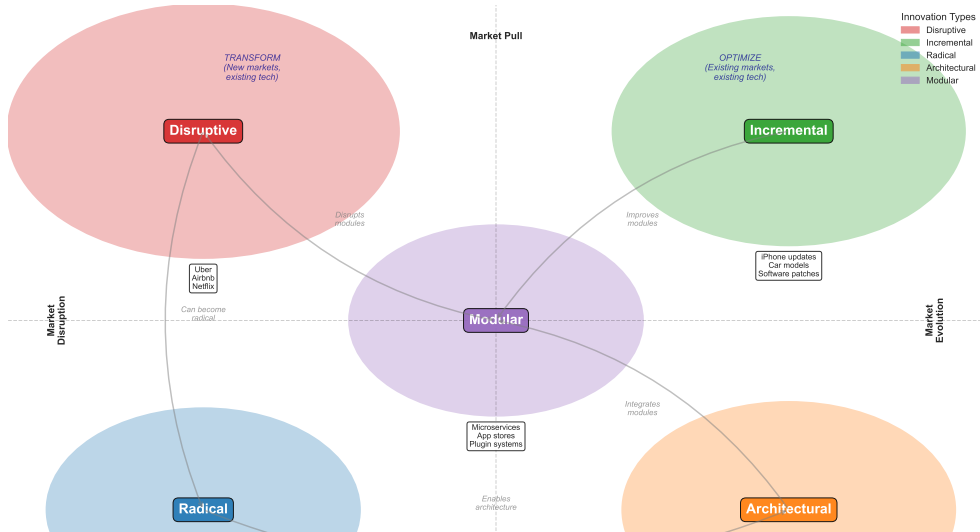
Innovation Pattern Discovery Through Clustering Revealing Hidden Market Opportunities



AI-Generated Innovation Archetypes

Data-Driven Character Development

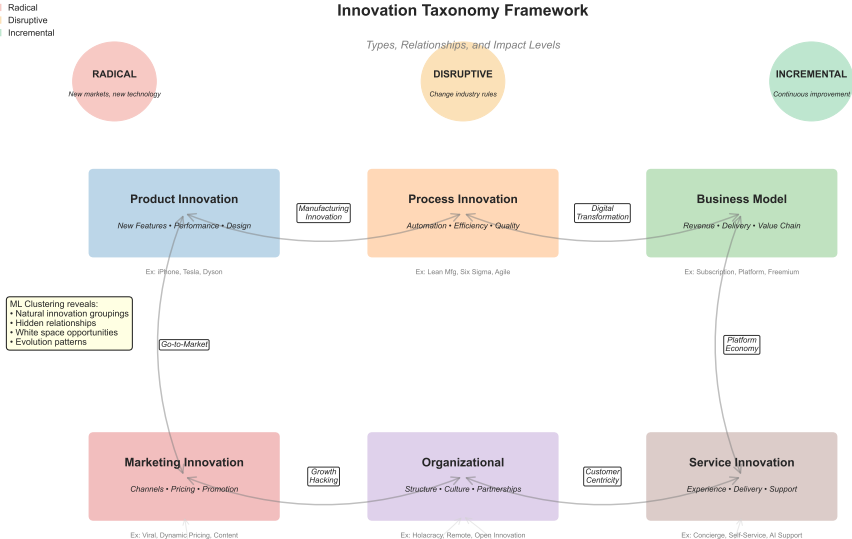
Innovation Archetypes Discovery Five Distinct Patterns from Clustering Analysis



Innovation Taxonomy Framework

Types, Relationships, and Impact Levels

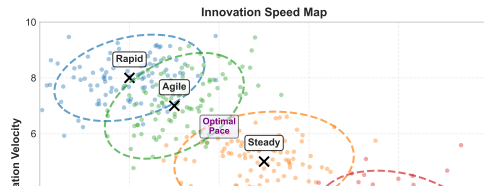
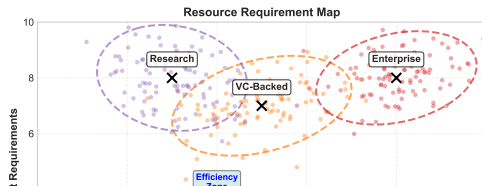
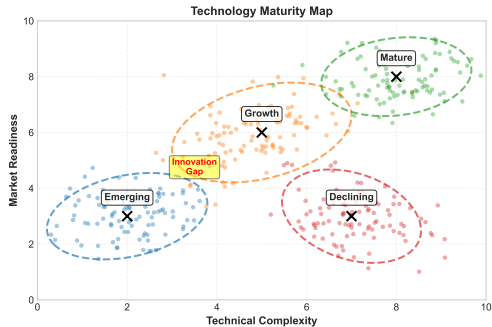
Impact Levels
Radical
Disruptive
Incremental



Innovation Pattern Mapping by Cluster

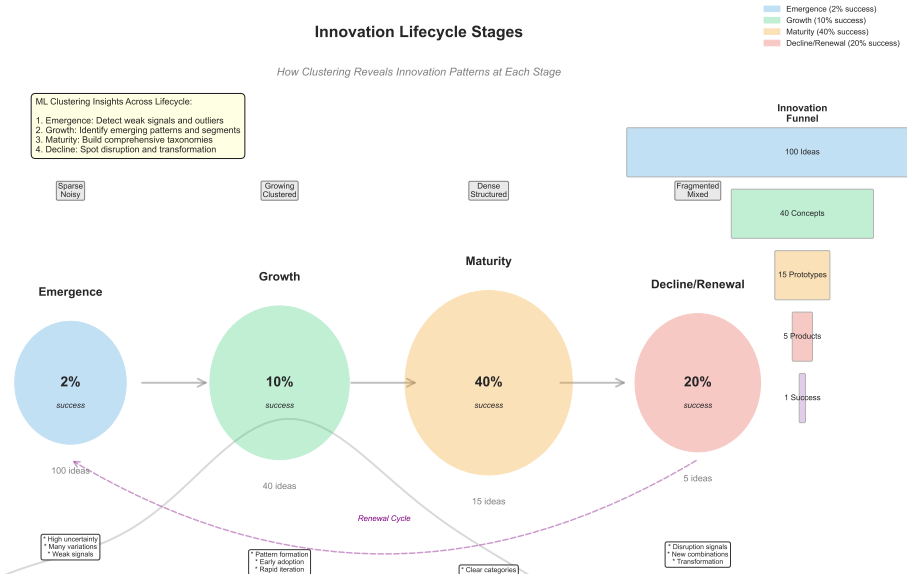
Understanding Each Category's Impact

Innovation Pattern Maps Four Perspectives on Innovation Categories



Innovation Lifecycle Stages

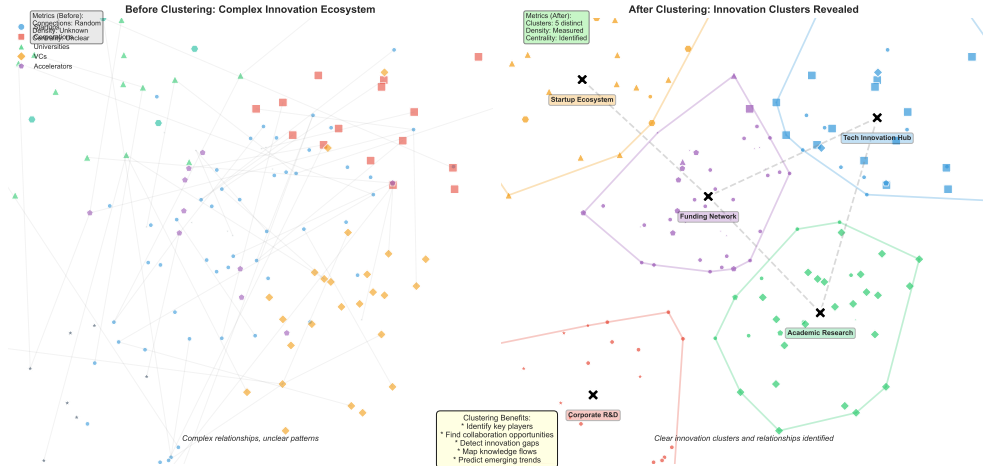
How Clustering Reveals Innovation Patterns at Each Stage



Innovation Ecosystem Mapping

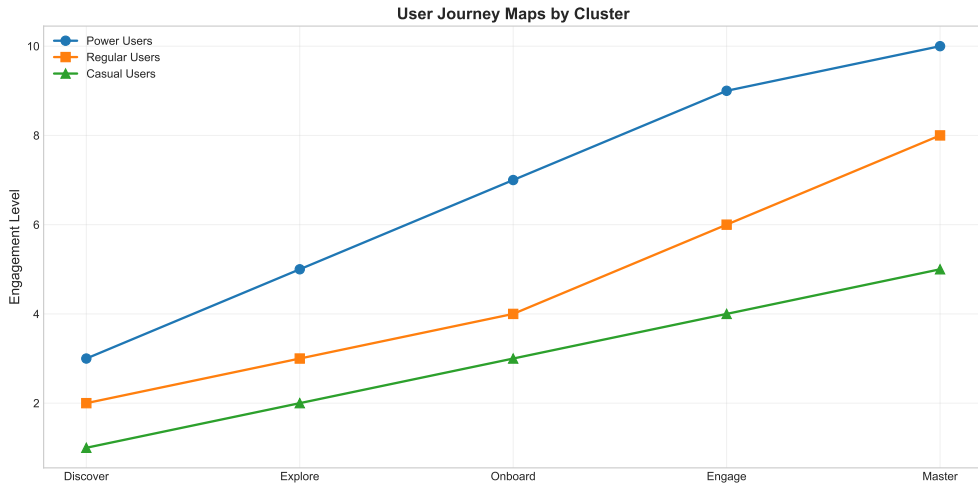
From Complex Networks to Clear Clusters

Innovation Ecosystem Mapping with Clustering



Different Evolution Paths for Innovation Types

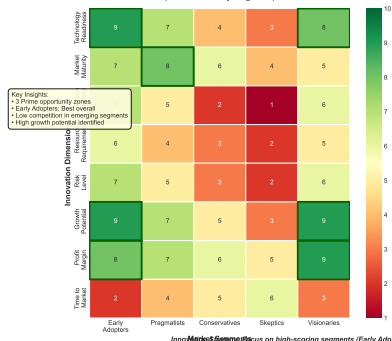
Innovation Lifecycle Patterns



Innovation Opportunities by Cluster

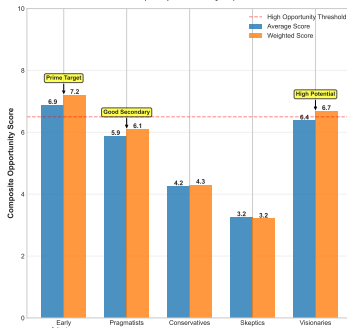
Where Each Category Has Potential

Innovation Opportunity Heatmap
(Raw Scores by Segment)



Market Segments focus on high-scoring segments (Early Adopters, Visionaries) with low competition and high growth potential

Aggregated Innovation Opportunities
(Composite Analysis)



Key Findings:

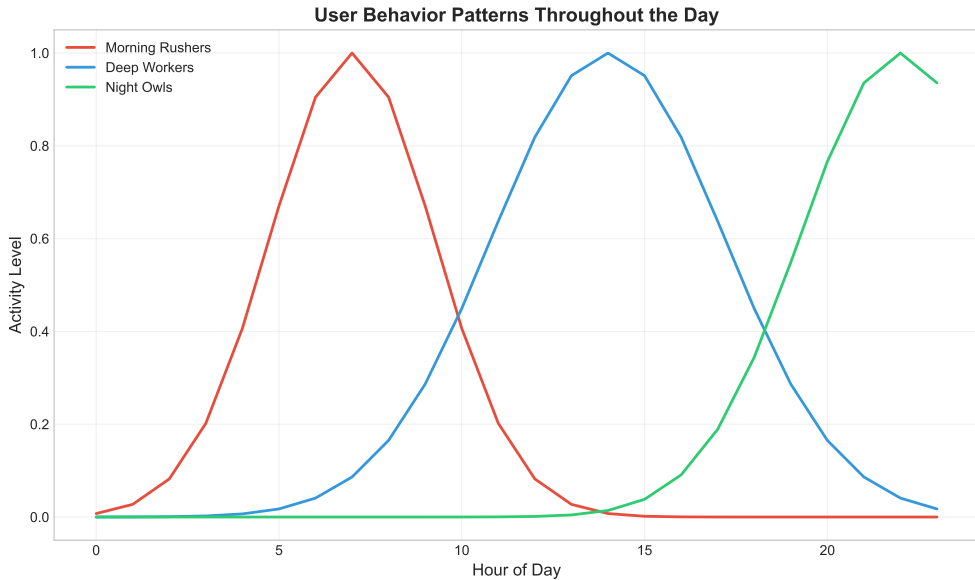
- Emerging tech: Early stage
- Disruptive: Scalability
- Incremental: Integration
- Platform-based: Network effects

Design implication:

One solution won't fit all!

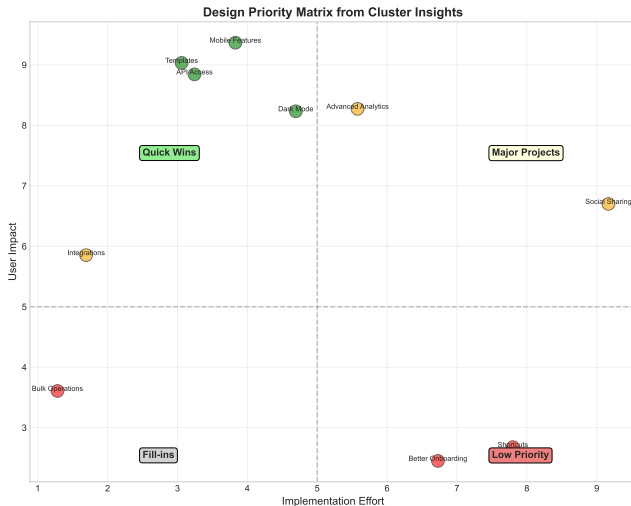
Innovation Patterns Revealed

What Clusters Tell Us About Evolution



Design Priority Matrix

Where to Focus Your Efforts



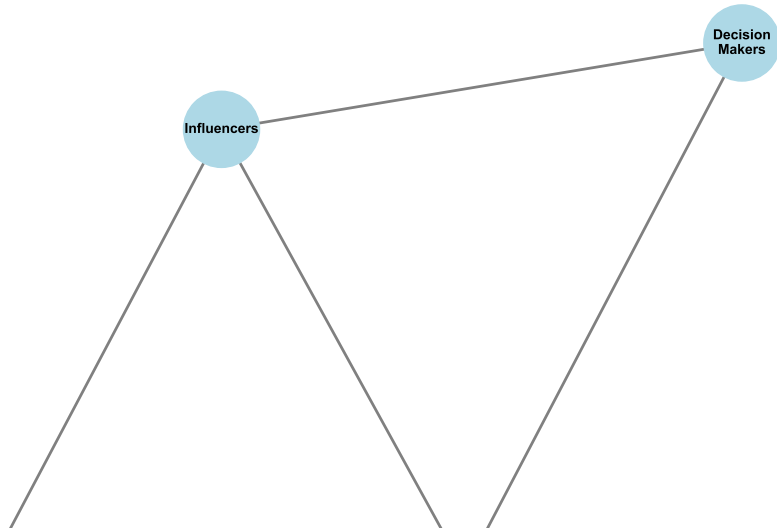
Priority Quadrants:

- **High Impact + High Effort**
Strategic initiatives
- **High Impact + Low Effort**
Quick wins
- **Low Impact + Low Effort**
Fill-ins
- **Low Impact + High Effort**
Avoid

Understanding Innovation Ecosystems

Network Analysis of Innovation Connections

Stakeholder Network from Cluster Analysis



You've learned:

- The clustering algorithms
- How to validate quality
- Design applications

Now let's see it in action

Real companies using these exact techniques
to drive innovation breakthroughs

PART 4

Summary & Practice

What we'll do:

- See real-world success patterns
- Consolidate key learnings
- Practice with exercises
- Preview next week
- Explore resources

From learning to doing

Real-World Clustering Patterns

Common Applications and Success Metrics

Clustering in Real-World Applications



Common Applications:

- Innovation portfolio management
- Technology trend clustering
- Opportunity space mapping
- Anomaly detection

Typical Results:

- Engagement: +35-45%
- Retention: +20-30%
- Conversion: +15-25%
- Processing time: -60%

Key Takeaways

What We've Learned

Technical Skills

- K-means clustering algorithm
- Choosing optimal K with elbow method
- Silhouette scores for validation
- DBSCAN for complex shapes
- Hierarchical clustering

Design Applications

- Data-driven innovation archetypes
- Segment-specific journeys
- Opportunity identification
- Priority matrices
- Scaled innovation analysis

Clustering transforms data into actionable innovation insights

Implementation Checklist

Ensuring Successful Clustering Projects

Data Preparation

- ☐ Collect relevant features
- ☐ Handle missing values
- ☐ Standardize/normalize data
- ☐ Remove outliers if needed
- ☐ Feature engineering complete
- ☐ Data quality verified

Quality Assurance

- ☐ Silhouette score ≥ 0.5
- ☐ Cluster sizes balanced
- ☐ Visual inspection done
- ☐ Stability tested
- ☐ Business sense verified
- ☐ Edge cases handled

Algorithm Selection

- ☐ Choose distance metric
- ☐ Select clustering method
- ☐ Determine optimal K
- ☐ Validate with metrics

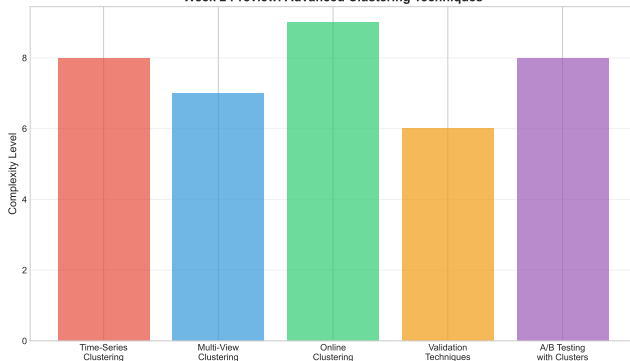
Common Pitfalls

- ✗ Forgetting to scale features
- ✗ Wrong distance metric
- ✗ Forcing unnatural K
- ✗ Ignoring outliers

Next Week: Advanced Clustering

Going Deeper into Innovation Patterns

Week 2 Preview: Advanced Clustering Techniques



Week 2 Topics:

- Density-based clustering
- Gaussian mixture models
- Clustering validation
- Feature engineering
- Real-time clustering

Design Focus:

- Dynamic innovation tracking
- Evolving innovation landscapes
- Predictive opportunity analysis
- Micro-innovation detection

Technical Resources

Papers:

- MacQueen, J. (1967). K-means
- Ester et al. (1996). DBSCAN
- Rousseeuw (1987). Silhouettes

Tools:

- scikit-learn clustering
- Orange data mining
- KNIME analytics

Design Resources

Books:

- “Design Thinking” - Tim Brown
- “Sprint” - Jake Knapp
- “Lean UX” - Jeff Gothelf

Applications:

- Miro (journey mapping)
- Figma (archetype creation)
- Optimal Workshop

Questions? Let's discuss!

Glossary of Technical Terms

Key Concepts Quick Reference

Clustering Algorithms:

- **K-Means:** Partitions data into K predefined clusters
- **DBSCAN:** Density-based spatial clustering
- **Hierarchical:** Builds cluster tree (dendrogram)
- **GMM:** Gaussian Mixture Models, soft clustering

Key Parameters:

- **K:** Number of clusters
- **eps:** Neighborhood radius (DBSCAN)
- **min_samples:** Minimum points for density
- **n_init:** Number of random initializations

Evaluation Metrics:

- **Silhouette:** Cluster cohesion vs separation $[-1,1]$
- **Inertia:** Sum of squared distances to centroids
- **Davies-Bouldin:** Ratio of within to between distances
- **Calinski-Harabasz:** Ratio of dispersions

Innovation Terms:

- **Empathy Mapping:** Understanding user perspectives
- **Pain Points:** User problems/frustrations
- **User Archetypes:** Representative user groups
- **Innovation Ecosystem:** Connected stakeholders

Implementation Checklist

Your Step-by-Step Guide to Success

Data Preparation:

- ☐ Collect innovation feedback data
- ☐ Clean and remove duplicates
- ☐ Handle missing values
- ☐ Normalize/standardize features
- ☐ Create feature vectors

Algorithm Selection:

- ☐ Analyze data distribution
- ☐ Choose appropriate algorithm
- ☐ Set initial parameters
- ☐ Prepare validation strategy

Implementation:

- ☐ Run clustering algorithm
- ☐ Calculate evaluation metrics
- ☐ Visualize results (PCA/t-SNE)
- ☐ Validate with domain experts
- ☐ Iterate and refine

Innovation Application:

- ☐ Map clusters to user personas
- ☐ Identify innovation opportunities
- ☐ Create targeted solutions
- ☐ Design prototype features
- ☐ Test with user groups

Ready? Start with data preparation and work your way down!

Objective Function (Inertia):

$$J = \sum_{i=1}^n \sum_{j=1}^k w_{ij} ||x_i - \mu_j||^2$$

Where:

- n = number of data points
- k = number of clusters
- $w_{ij} = 1$ if x_i belongs to cluster j , 0 otherwise
- μ_j = centroid of cluster j

Update Rules:

- 1 Assignment: $c^{(i)} = \arg \min_j ||x^{(i)} - \mu_j||^2$
- 2 Update: $\mu_j = \frac{1}{|S_j|} \sum_{i \in S_j} x^{(i)}$

Appendix: Distance Metrics

Mathematical Definitions

Euclidean Distance:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Manhattan Distance:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Minkowski Distance:

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Cosine Similarity:

$$\cos(\theta) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

Jaccard Distance:

$$J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Mahalanobis Distance:

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

Silhouette Score for point i :

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

Where:

- $a(i)$ = average distance to points in same cluster
- $b(i)$ = average distance to points in nearest neighbor cluster

Interpretation:

- $s(i) \approx 1$: Well clustered
- $s(i) \approx 0$: On border between clusters
- $s(i) \approx -1$: Misclassified

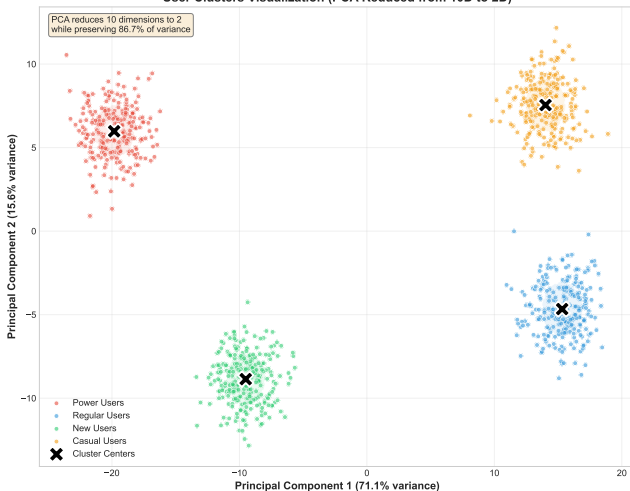
Overall Score:

$$S = \frac{1}{n} \sum_{i=1}^n s(i)$$

Appendix: PCA for Cluster Visualization

Dimensionality Reduction

User Clusters Visualization (PCA Reduced from 10D to 2D)



PCA Process:

- 1 Standardize data
- 2 Compute covariance matrix
- 3 Find eigenvectors/values
- 4 Select top 2 components
- 5 Transform data

Variance Explained:

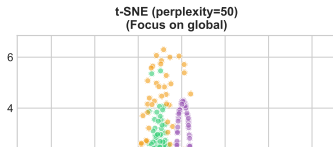
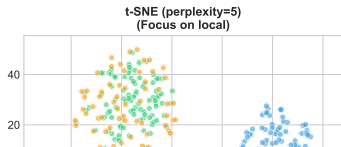
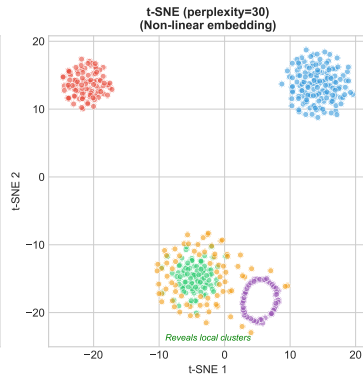
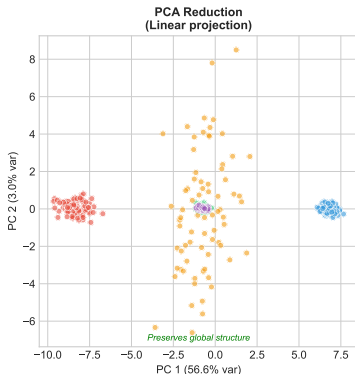
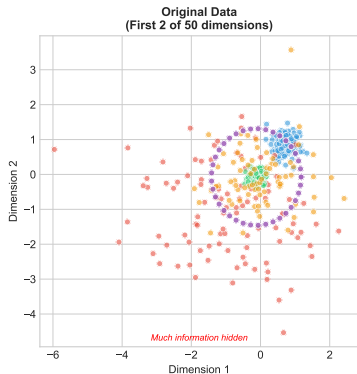
- PC1: 45.2%
- PC2: 28.7%
- Total: 73.9%

Dimensionality Reduction: PCA vs t-SNE

Revealing Hidden Patterns in High-Dimensional Innovation Space

Dimensionality Reduction: PCA vs t-SNE for Innovation Data

Revealing Hidden Patterns in High-Dimensional Innovation Space



Method Comparison

	PCA	t-SNE
Speed	Fast	Slow
Scalability	Excellent	Limited

Key Parameters:

- ϵ (eps): Maximum distance between points
- MinPts: Minimum points to form dense region

Point Classification:

- **Core point:** Has \geq MinPts within ϵ
- **Border point:** Within ϵ of core point
- **Noise point:** Neither core nor border

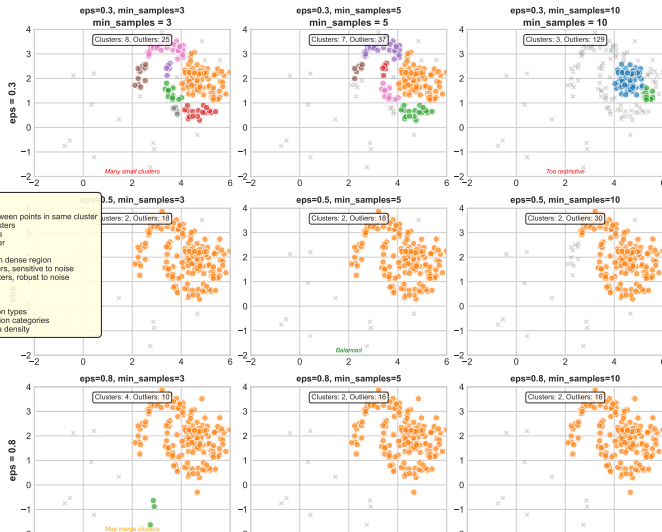
Algorithm Steps:

- 1 Find all core points
- 2 Form clusters from core points within ϵ
- 3 Assign border points to clusters
- 4 Mark remaining as noise

DBSCAN Parameter Tuning

Impact of eps and min_samples on Clustering Results

DBSCAN Parameter Tuning: Impact on Innovation Clustering



Parameter Guidelines:

eps (epsilon): Maximum distance between points in same cluster

- * Small eps → Many small, tight clusters
- * Large eps → Fewer, larger clusters
- * Too large → All points in one cluster

min_samples: Minimum points to form dense region

- * Small min_samples → More clusters, sensitive to noise
- * Large min_samples → Fewer clusters, robust to noise
- * Too large → Many outliers

For Innovation Data:

- * Use small eps for distinct innovation types
- * Use larger eps for broader innovation categories
- * Adjust min_samples based on data density

Tuning Strategy:

1. Start with k-distance plot
2. Look for 'elbow' in plot
3. Set eps at elbow point
4. min_samples = 2*dimensions
5. Validate with domain knowledge

Appendix: Python Implementation

Ready-to-Use Code Snippets

K-Means Example:

```
from sklearn.cluster import KMeans
import numpy as np

# Generate data
X = np.random.randn(1000, 2)

# Fit K-means
kmeans = KMeans(n_clusters=3,
                random_state=42)
labels = kmeans.fit_predict(X)

# Get centroids
centroids = kmeans.cluster_centers_
```

DBSCAN Example:

```
from sklearn.cluster import DBSCAN

# Fit DBSCAN
dbscan = DBSCAN(eps=0.3,
                min_samples=5)
labels = dbscan.fit_predict(X)

# Identify outliers
outliers = labels == -1
n_clusters = len(set(labels)) - 1

print(f"Clusters: {n_clusters}")
print(f"Outliers: {sum(outliers)}")
```

Appendix: Implementation Guidelines

Practical Considerations

Data Preparation

- Standardize features
- Handle missing values
- Remove outliers (if needed)
- Feature selection/engineering
- Consider scaling methods

Validation Methods

- Silhouette score
- Davies-Bouldin index
- Calinski-Harabasz score
- Visual inspection
- Domain expert review

Algorithm Selection

- K-means: Spherical, similar size
- DBSCAN: Arbitrary shapes
- Hierarchical: Nested structure
- GMM: Overlapping clusters

Common Pitfalls

- Not scaling features
- Wrong distance metric
- Ignoring outliers
- Over-clustering
- Forcing clusters

Glossary of Technical Terms

Key Concepts Reference

Algorithms:

- **K-means:** Partitions data into K spherical clusters
- **DBSCAN:** Density-based clustering for arbitrary shapes
- **GMM:** Gaussian Mixture Models for soft clustering
- **Hierarchical:** Tree-based clustering approach

Metrics:

- **Silhouette:** Measures cluster separation (-1 to 1)
- **Inertia:** Sum of squared distances to centroids
- **Davies-Bouldin:** Ratio of within to between cluster distance

Concepts:

- **Centroid:** Center point of a cluster
- **Elbow Method:** Technique to find optimal K
- **Outlier:** Data point not belonging to any cluster
- **Convergence:** When algorithm stops improving

Preprocessing:

- **Standardization:** Zero mean, unit variance
- **Normalization:** Scale to [0,1] range
- **PCA:** Principal Component Analysis
- **t-SNE:** t-distributed Stochastic Neighbor Embedding