

Machine Learning for Smarter Innovation

Week 1: Foundations & Clustering

Discovering Innovation Patterns with ML

BSc Course in AI-Enhanced Innovation

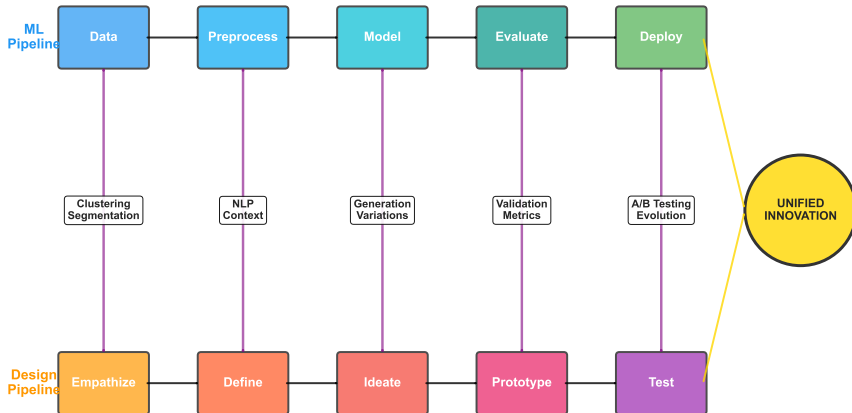
Machine Learning + Innovation + Design Thinking

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

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

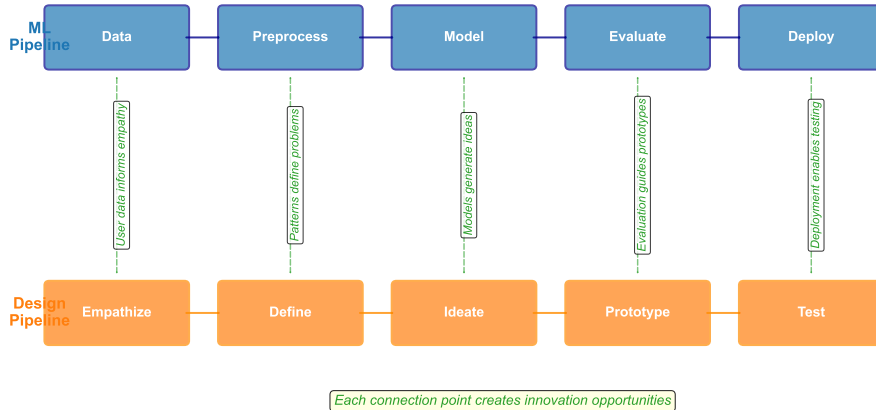
- **Scale:** Analyze millions of data points
- **Speed:** Real-time insights
- **Objectivity:** Data-driven discovery
- **Patterns:** Find non-obvious relationships
- **Iteration:** Continuous learning

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

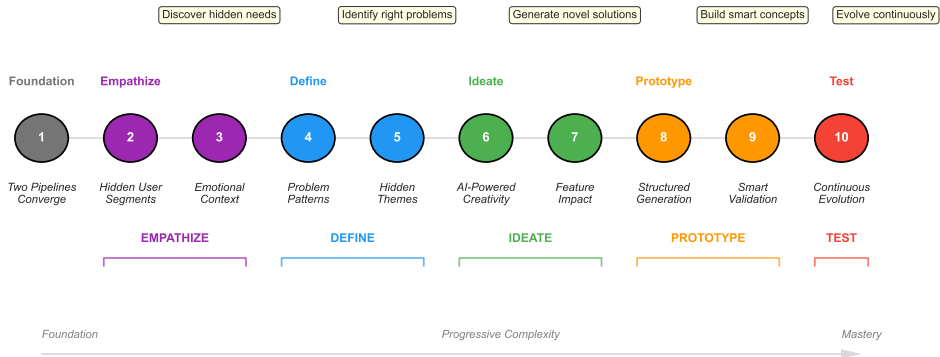
- Understand innovation needs
- Frame problems
- Generate solutions
- Build concepts
- Validate innovation impact

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
Empathize	1-2	Discover 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 Discovery

Week 1: Clustering for Innovation

From Scattered Ideas to Innovation Patterns

What We'll Learn:

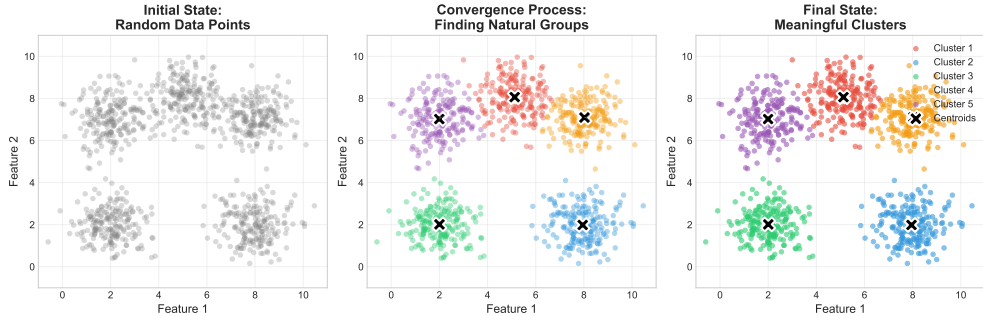
- How clustering reveals innovation categories
- K-means algorithm fundamentals
- Finding the optimal number of clusters
- Quality metrics for validation
- Advanced clustering techniques

Design Applications:

- Create innovation archetypes
- Map innovation evolution paths
- Identify pain points 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

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?
- Personalize at scale?
- Identify opportunity gaps?

We can!

Solution: Clustering

What is Clustering?

Finding Natural Groups in Innovation Data

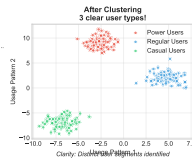
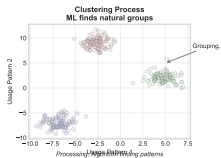
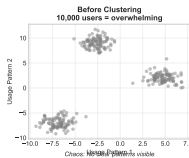
Clustering Finds:

- 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

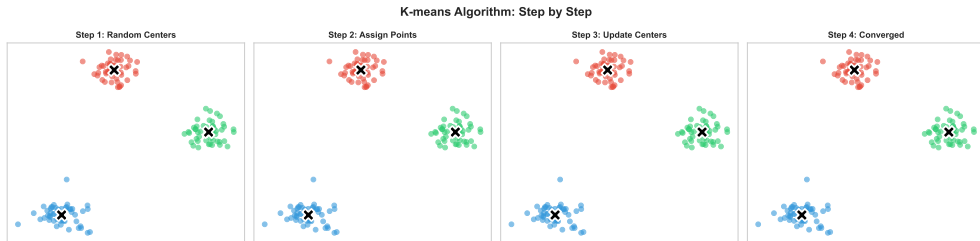
Strengths:

- 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

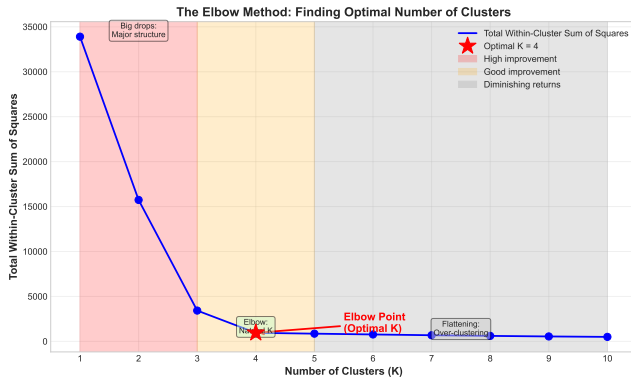
Finding the Optimal Number of Clusters

Finding the Elbow:

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

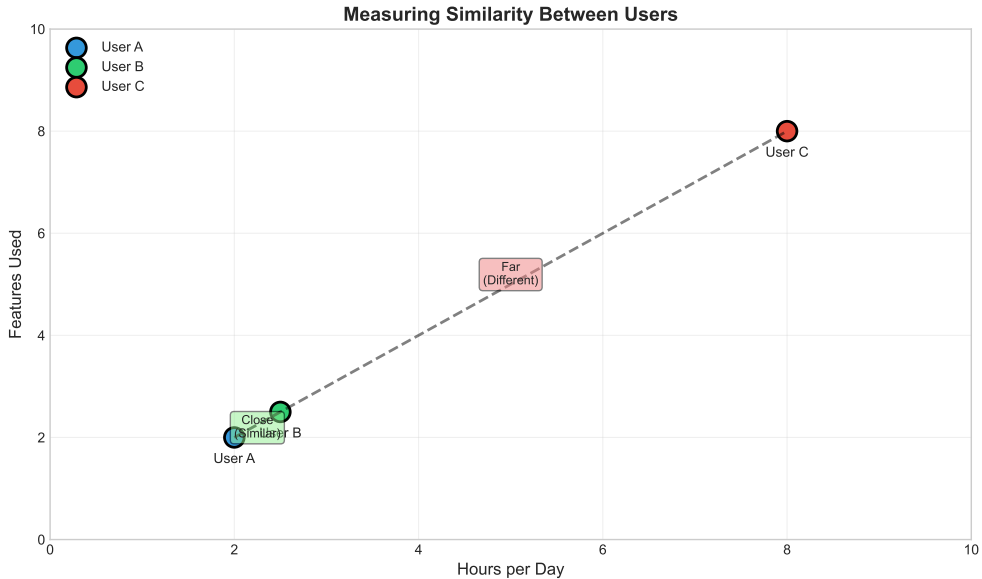
Optimal K = 5

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!



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

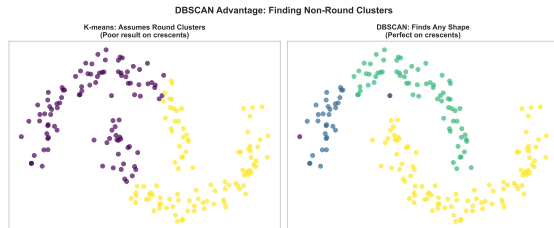
Finding Natural Boundaries, Not Forcing Shapes

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

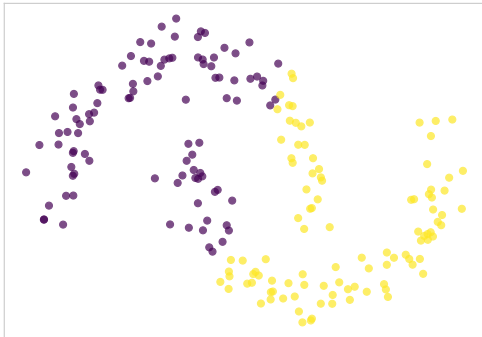


DBSCAN: Complex Patterns

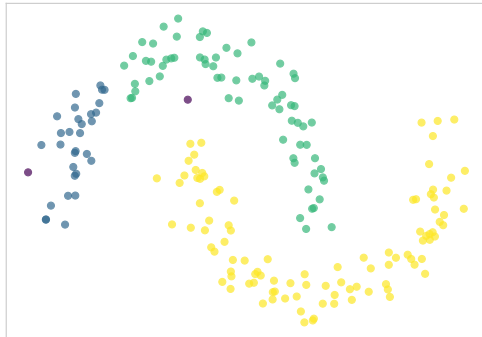
When K-Means Isn't Enough

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

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

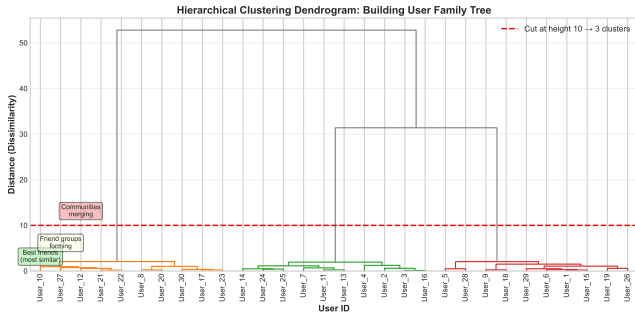
Building a Tree of Relationships

Dendrogram Benefits:

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

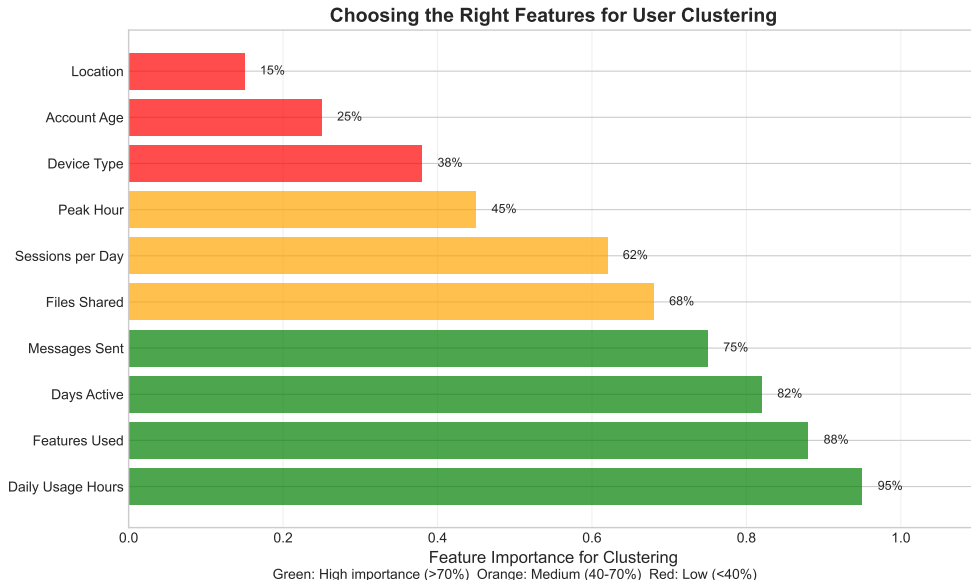
Cut the tree at any level:

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



What Drives the Clusters?

Feature Importance Analysis



We've mastered the technical tools:

Clustering, metrics, quality measures

But clusters are just numbers...

Until we connect them to human needs

Let's transform data into empathy

Each cluster represents innovation opportunities and patterns

PART 3

Design Integration

What we'll create:

- Data-driven personas
- Empathy maps per segment
- Cluster-specific journeys
- Pain point heat maps
- Design priority matrices

Where ML meets human-centered design

From Data Points to Innovation Insights

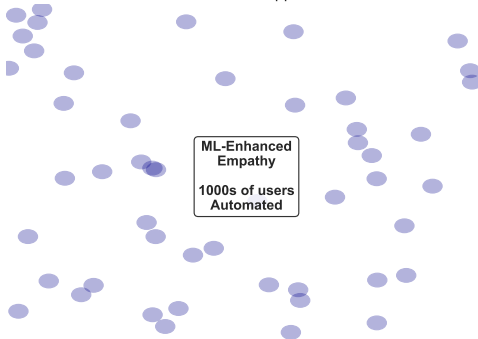
Bridging the Technical-Human Gap

Scaling Empathy with Machine Learning

Traditional Approach



ML-Enhanced Approach



Each cluster represents innovation opportunities

AI-Generated Innovation Archetypes

Data-Driven Character Development

Data-Driven Persona Cards

Power Paula

Age: 32

Role: Manager

Usage: 7h/day

Regular Rob

Age: 28

Role: Developer

Usage: 4h/day

Casual Carl

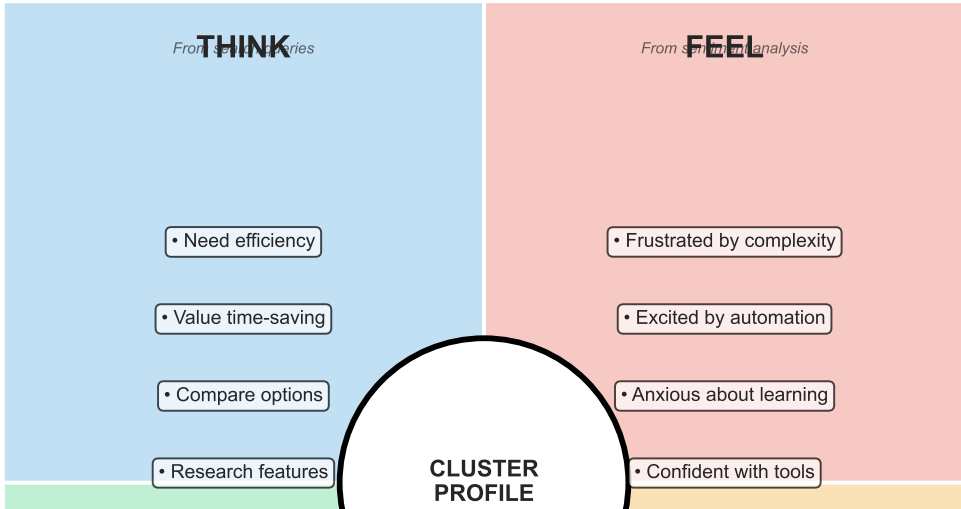
Age: 24

Role: Student

Usage: 1h/day

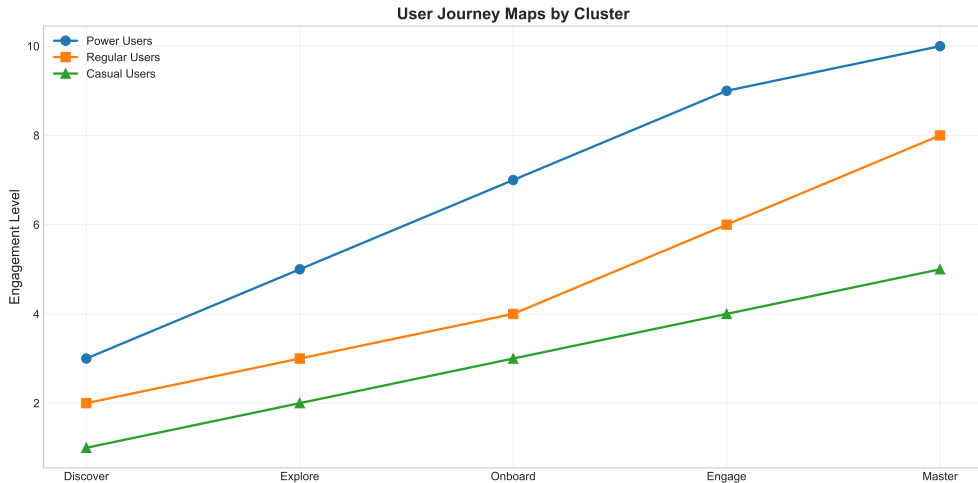
Disruptive — Incremental — Platform-Based — Service-Oriented — Hybrid Models

Empathy Map: Data-Driven User Understanding



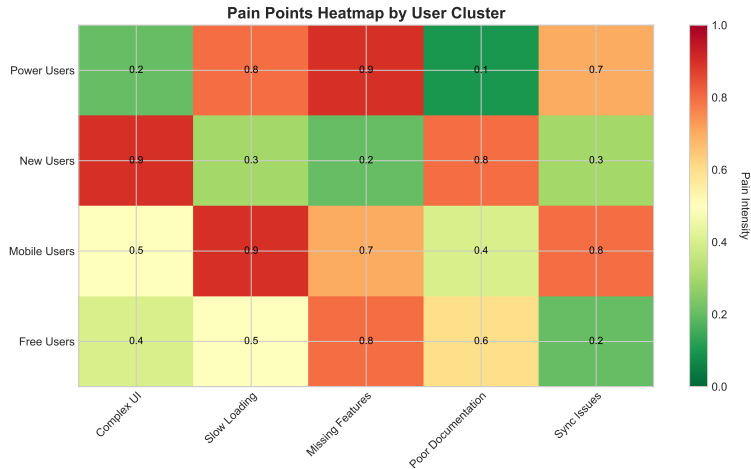
Different Evolution Paths for Innovation Types

Innovation Lifecycle Patterns



Innovation Opportunities by Cluster

Where Each Category Has Potential



Key Findings:

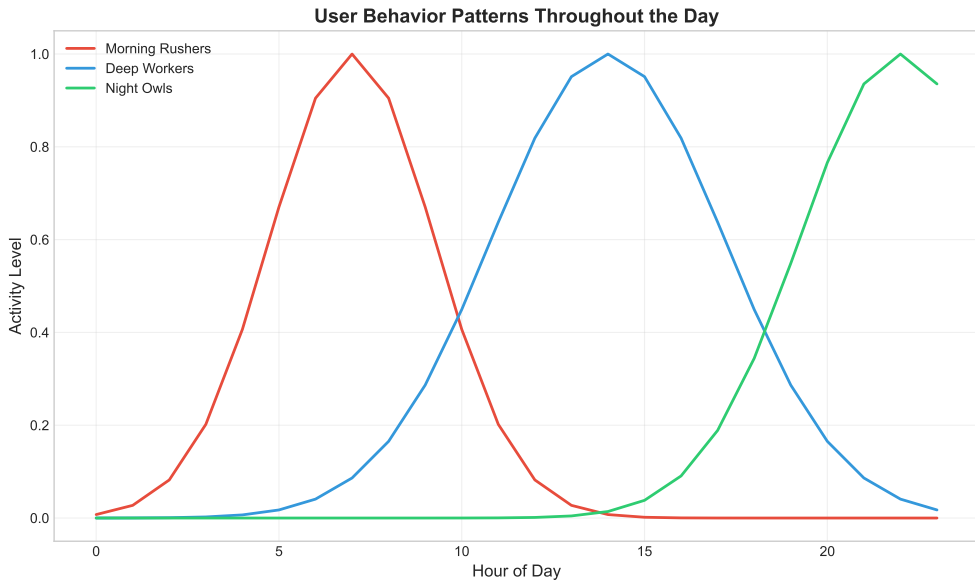
- Emerging tech: Early stage
- Disruptive: Scalability
- Incremental: Integration
- Platform-based: Network effects

Design implication:

One solution won't fit all!

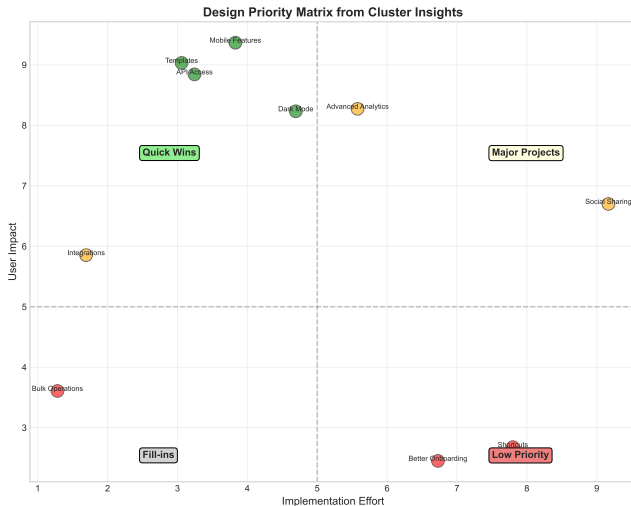
Innovation Patterns Revealed

What Clusters Tell Us About Evolution



Design Priority Matrix

Where to Focus Your Efforts



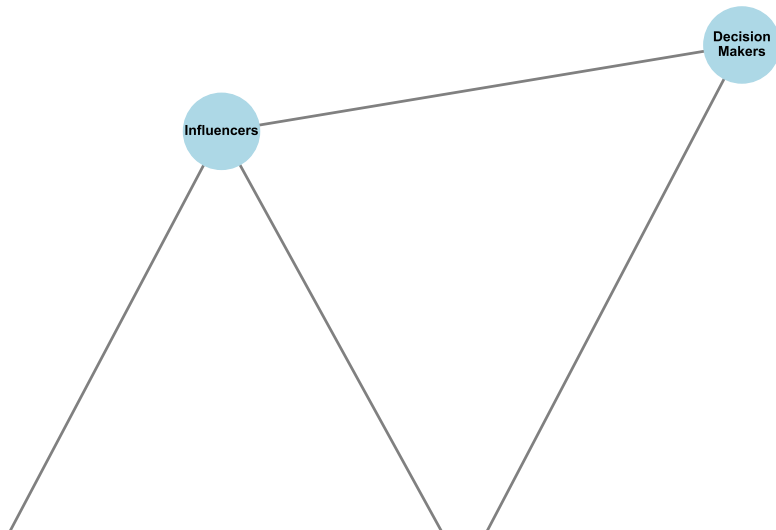
Priority Quadrants:

- **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 transform their user experience

PART 4

Summary & Practice

What we'll do:

- See real-world success (Spotify)
- 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 personas
- Segment-specific journeys
- Pain point identification
- Priority matrices
- Scaled empathy

Clustering transforms data into actionable user 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

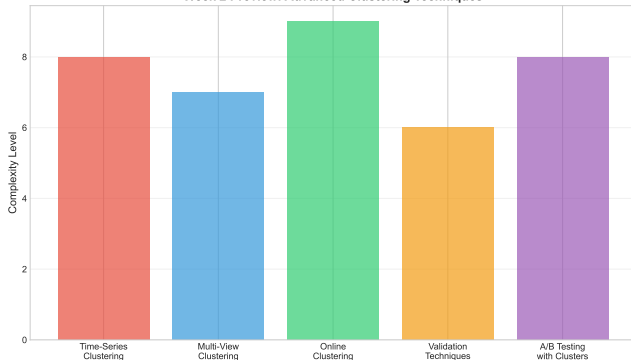
Common Pitfalls

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

Next Week: Advanced Clustering

Going Deeper into User Understanding

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 (persona creation)
- Optimal Workshop

Questions? Let's discuss!

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%

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

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