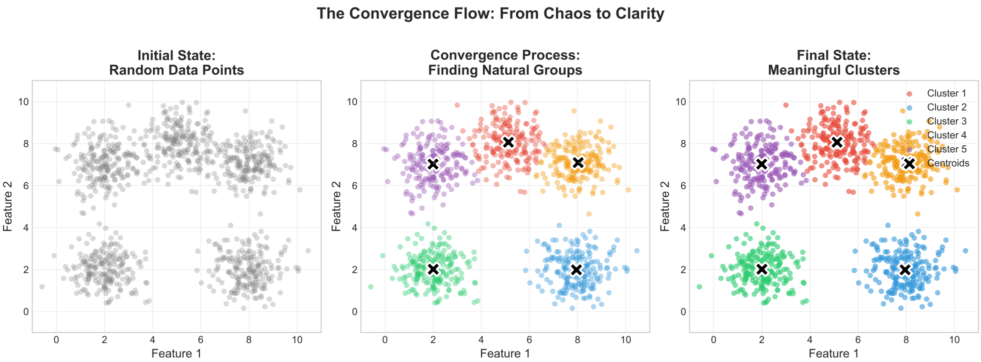
Machine Learning for Smarter Innovation

Week 1: Foundations & Clustering

Augmenting the Empathize Phase with ML

BSc Course in AI-Enhanced Innovation



The Convergence Flow: Order from Chaos

Watch 5000 data points self-organize into meaningful clusters

That visualization shows the end result.

But every innovation journey starts with a problem.

What problem does clustering solve?

Let's discover why traditional design thinking needs ML augmentation.

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

AI-Enhanced Innovation

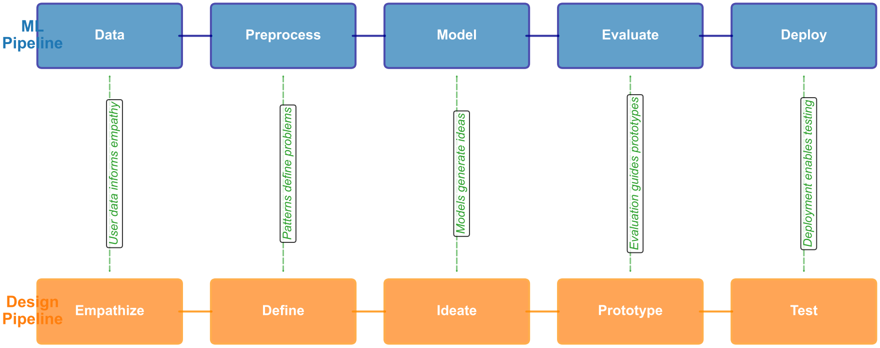
- **Scale**: Analyze millions of data points
- **Speed**: Real-time insights
- **Objectivity**: Data-driven discovery
- **Patterns**: Find non-obvious relationships
- **Iteration**: Continuous learning

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

The Dual Pipeline

Where ML Meets Design Thinking

The Convergence: ML Meets Design Thinking



Each connection point creates innovation opportunities

The Dual Pipeline (Continued)

Understanding Both Worlds

ML Pipeline

Data → Preprocess → Model → Evaluate → Deploy

- Collect user behavior
- Clean and transform
- Train algorithms
- Validate accuracy
- Scale to production

Design Pipeline

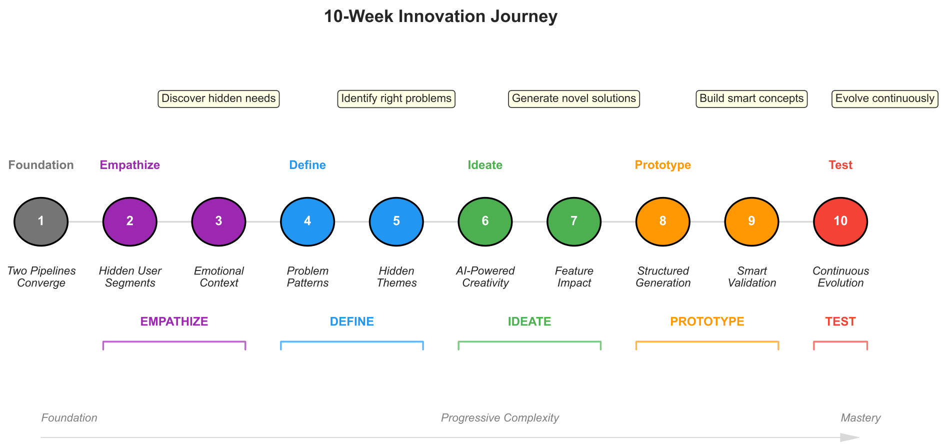
Empathize → Define → Ideate → Prototype → Test

- Understand users
- Frame problems
- Generate solutions
- Build concepts
- Validate with users

Integration = Innovation at Scale

Your Innovation Journey

10 Weeks to AI-Powered Design Mastery



Your Innovation Journey (Continued)

What You'll Master in Each Stage

Stage	Weeks	Innovation Unlocked
Empathize	1-2	Discover hidden user needs at scale
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 Deep User Understanding

What We'll Learn:

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

Design Applications:

- Create data-driven personas
- Map user journeys by segment
- Identify pain points systematically
- Prioritize design efforts
- Scale empathy to thousands

Goal: Transform data points into human insights

We've seen the challenge:
Thousands of users with hidden patterns

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 User Segmentation Problem

5000 Users - Are They All the Same?

The Pain

Current Reality:

- One-size-fits-all solutions
- Generic user personas
- Missed opportunities
- Unhappy edge cases

The Cost:

- 73% of users feel misunderstood
- Features nobody uses
- Wasted development time

The Question

What if we could...

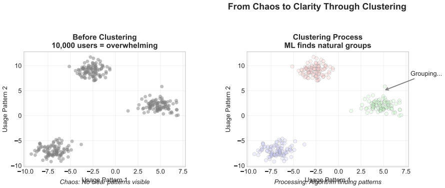
- Find natural user groups?
- Discover hidden segments?
- Personalize at scale?
- Understand real needs?

We can!

Solution: Clustering

What is Clustering?

Finding Natural Groups in Data



Clustering Finds:

- Natural groupings
- Similar behaviors
- Hidden segments
- Pattern relationships

Key Insight:
Users who behave similarly likely have similar needs

K-Means: The Workhorse Algorithm

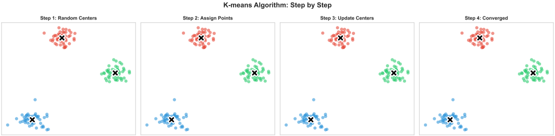
How It Organizes Your Users

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

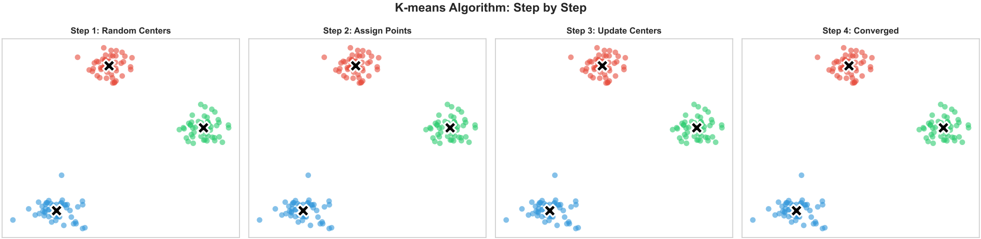
Strengths:

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



K-Means in Action

Step-by-Step Convergence



Iteration 1 $\rightarrow$ Iteration 3 $\rightarrow$ Iteration 5 $\rightarrow$ **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

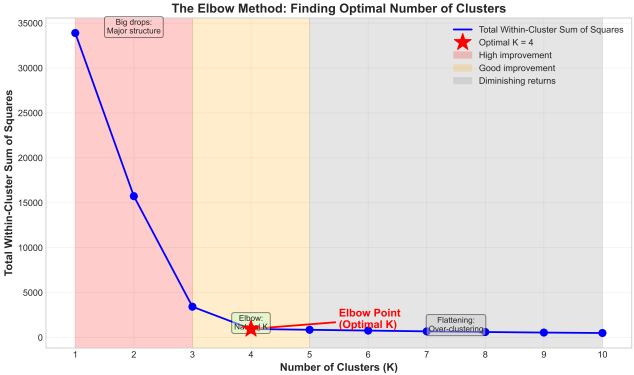
- Clear segments
- Actionable insights
- Manageable complexity

Too Many (K

Analysis Paralysis

- Overfitting
- Tiny segments
- Impossible to act on

How do we find the sweet spot?



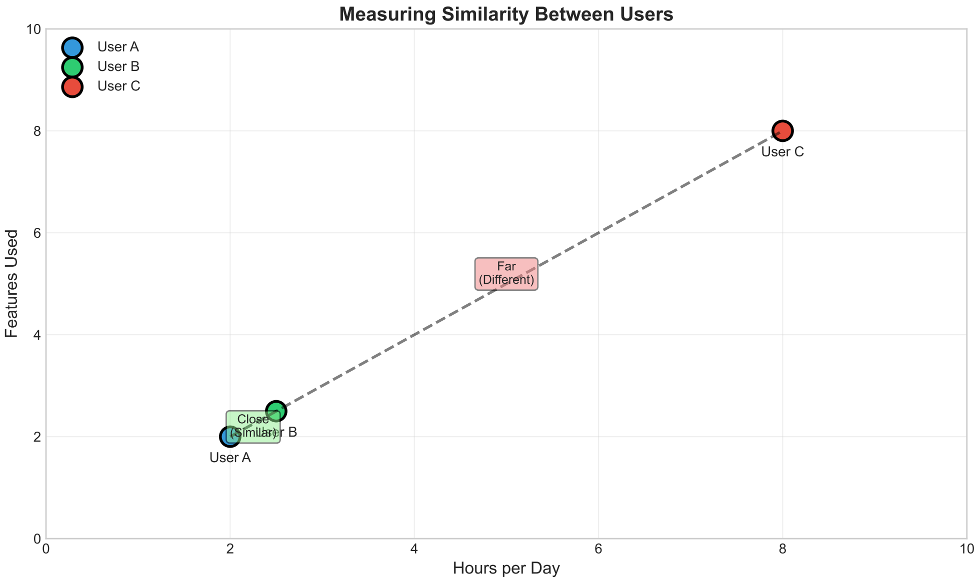
Finding the Elbow:

- 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

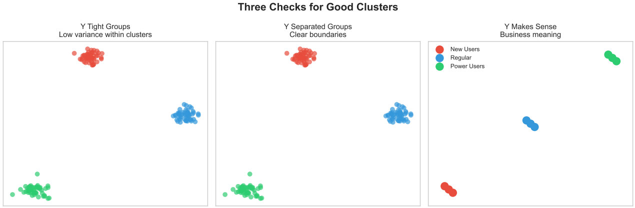
Distance Metrics

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!

When Circles Don't Work

Real User Groups Aren't Always Round

K-Means Assumes Spherical Clusters

But what about:

- Users connected through social networks (chains)
- Geographic clusters (irregular shapes)
- Behavioral patterns (crescents, spirals)
- Outliers and noise points

K-Means Forces Round Pegs into Round Holes

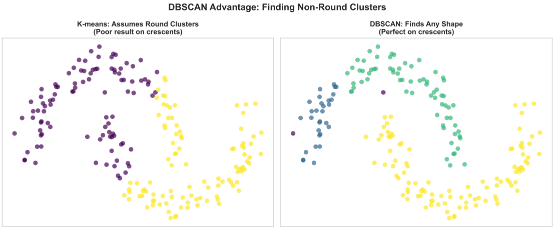
Solution: Density-Based Clustering

DBSCAN Advantages:

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

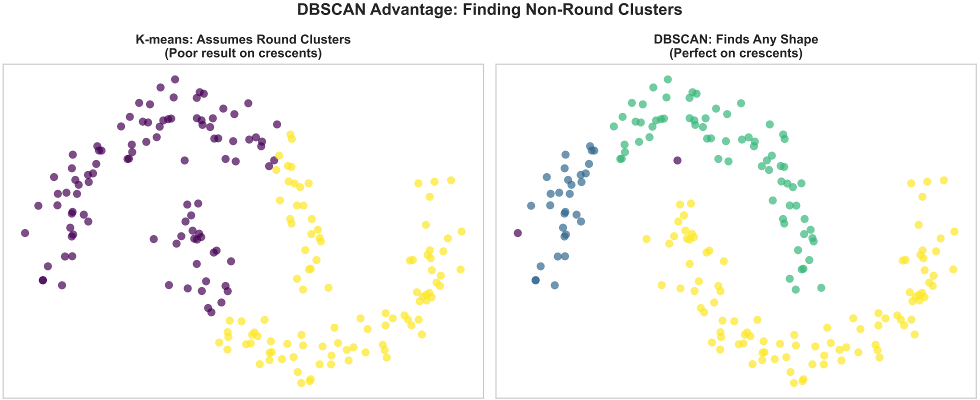
Perfect for:

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

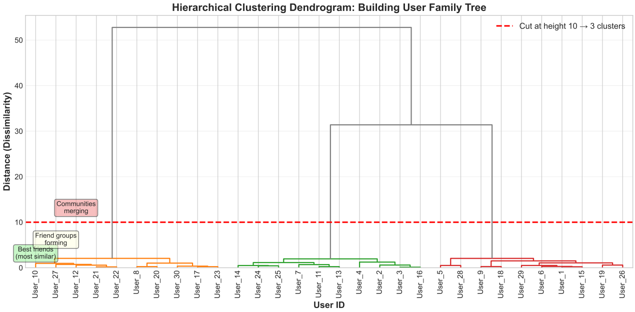


DBSCAN: Complex Patterns

When K-Means Isn't Enough



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



Dendrogram Benefits:

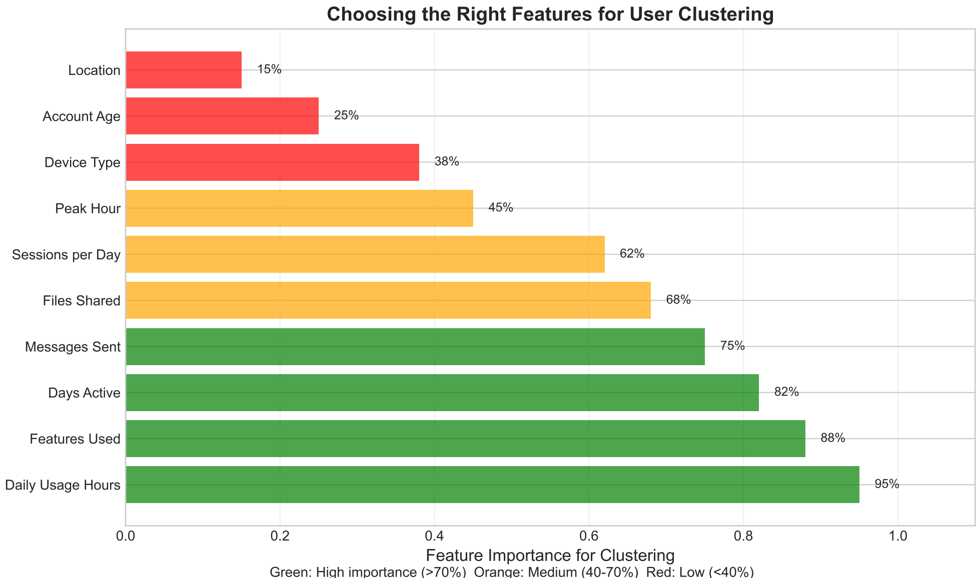
- Shows cluster hierarchy
- Multiple granularities
- Natural relationships
- No preset K needed

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 real people with real problems

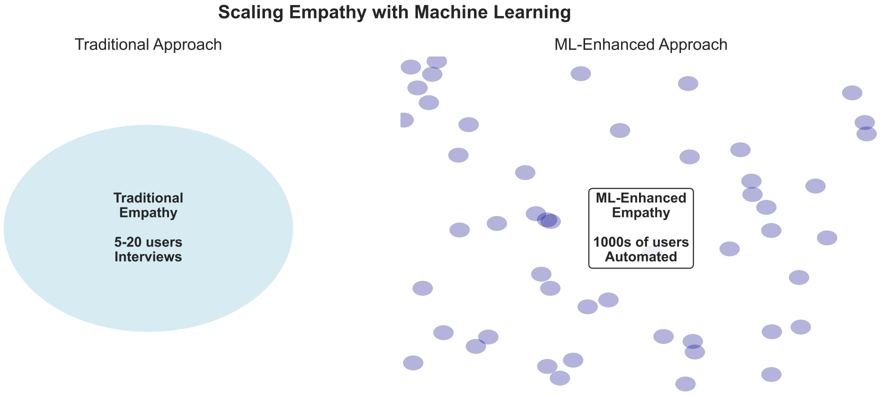
PART 3

Design Integration

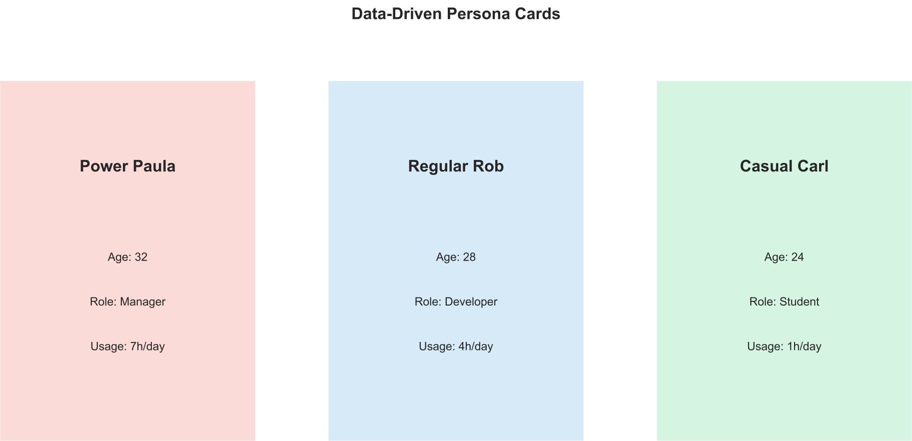
What we'll create:

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

Where ML meets human-centered design

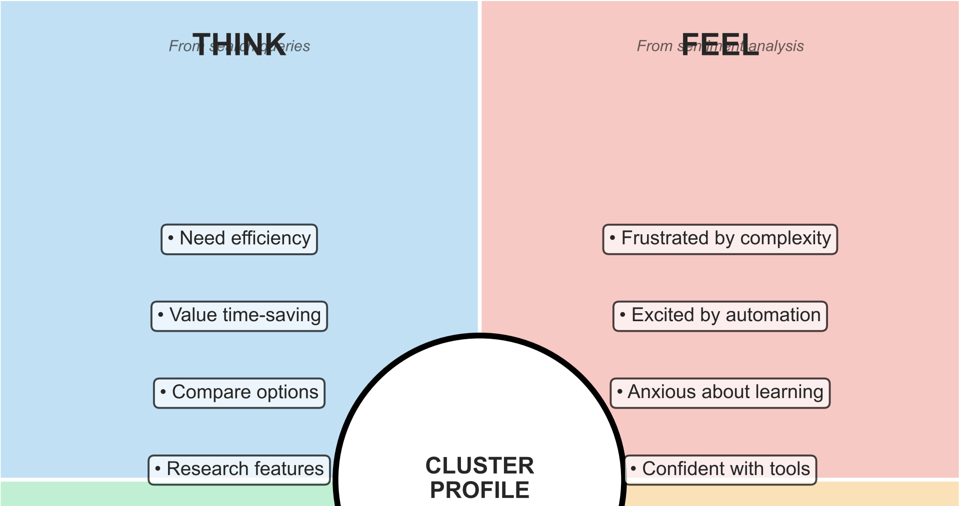


Each cluster represents real human needs



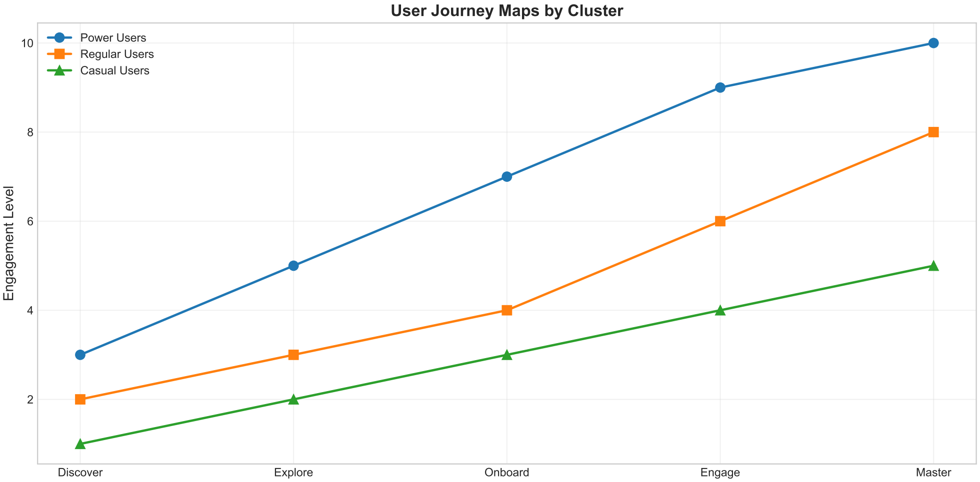
Power Users — Casual Browsers — Price-Conscious — Feature Seekers — New Users

Empathy Map: Data-Driven User Understanding



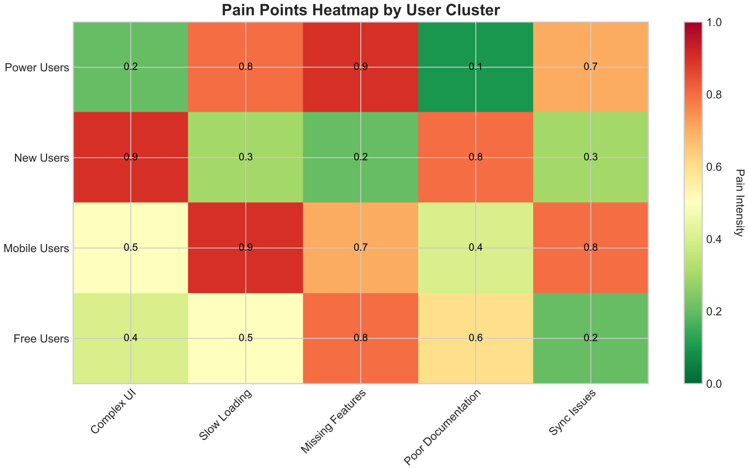
Different Journeys for Different Clusters

Personalized Path Understanding



Pain Points by Cluster

Where Each Segment Struggles



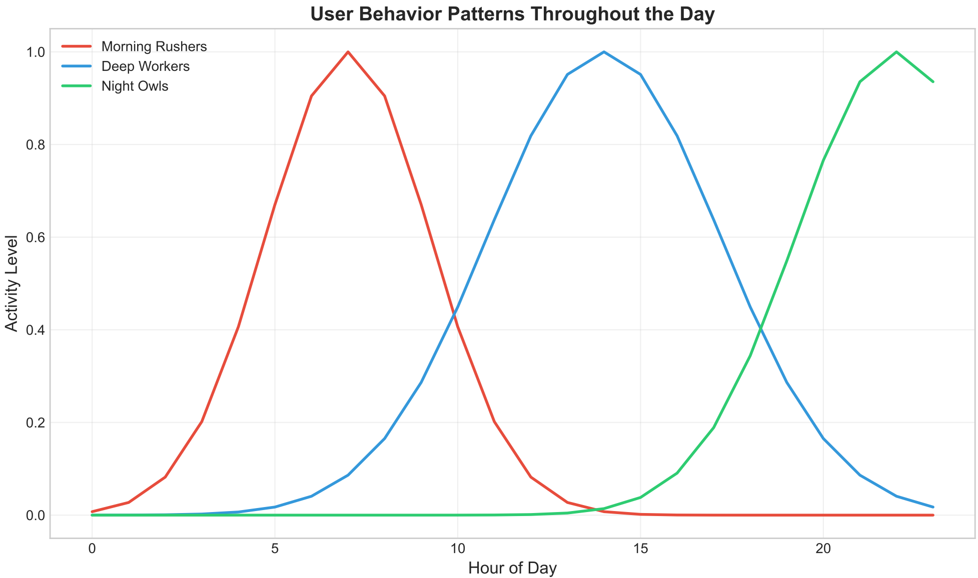
Key Findings:

- New users: Onboarding
- Power users: Speed
- Casual: Complexity
- Price-conscious: Value

Design implication:
One solution won't fit all!

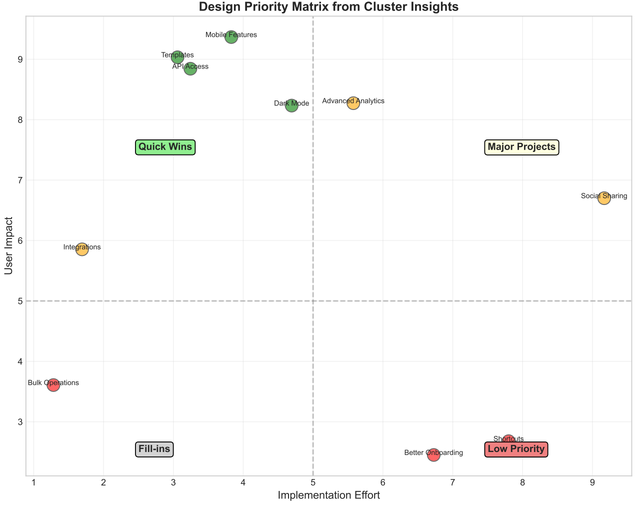
Behavioral Patterns Revealed

What Clusters Tell Us About Usage



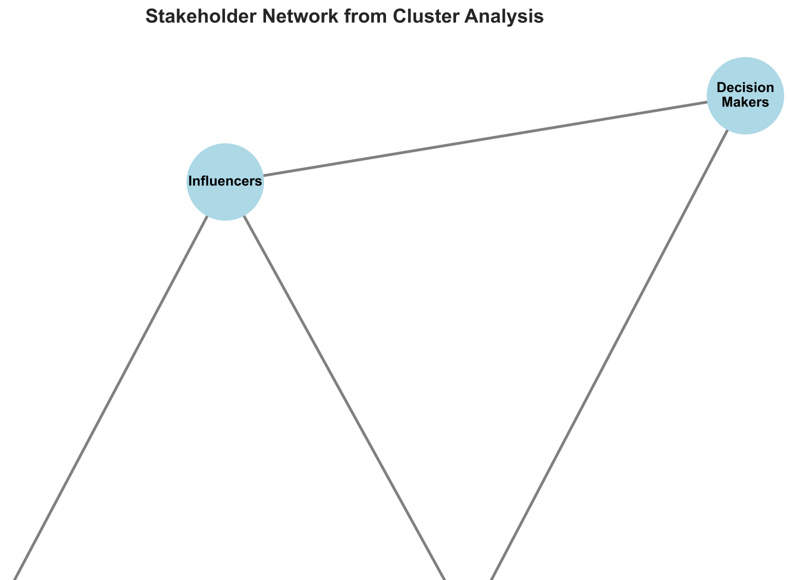
Design Priority Matrix

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


Stakeholder Network from Cluster Analysis
Decision Makers
Influencers

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

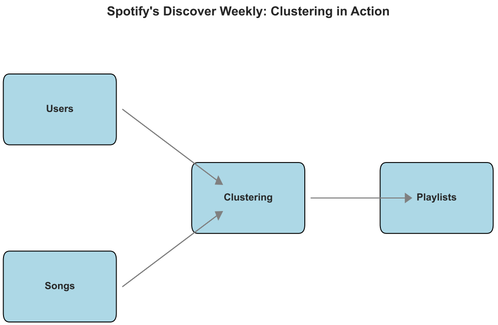
What we'll do:

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

From learning to doing

Case Study: Spotify's Clustering Success

Real-World Application



Spotify Uses Clustering For:

- Music taste profiles
- Discover Weekly playlists
- User segmentation
- Recommendation engine

Results:

- Personalized experience
- Increased engagement
- Better retention
- Discovery of new artists

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

Your Turn: Practice Exercise

Apply What You've Learned

Exercise: Segment Your Users

Scenario: You have data from 1000 app users including:

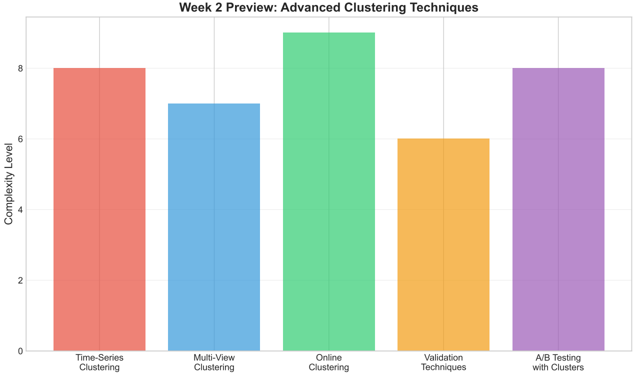
- Usage frequency
- Feature preferences
- Time spent
- Purchase history

Tasks:

1. Choose appropriate features for clustering
2. Determine optimal number of clusters
3. Interpret what each cluster represents
4. Create one persona per cluster
5. Identify key pain points for each segment

Next Week: Advanced Clustering

Going Deeper into User Understanding



Week 2 Topics:

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

Design Focus:

- Dynamic personas
- Evolving segments
- Predictive empathy
- Micro-segmentation

Resources & Further Reading

Deepen Your Understanding

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: $\mu_j = \frac{1}{|S_j|} \sum_{i \in S_j} x^{(i)}$

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:

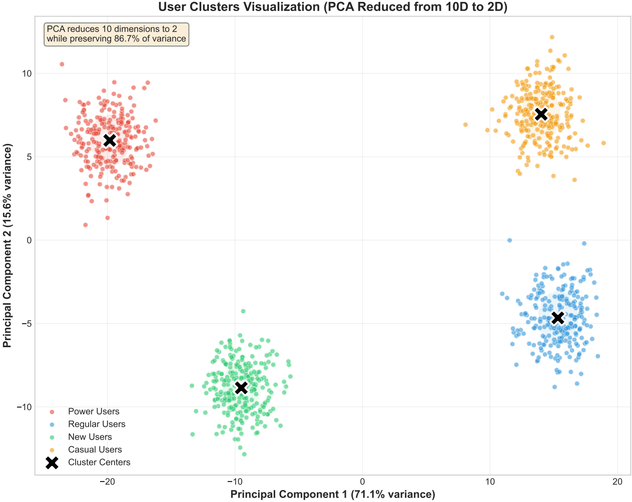
- $s(i) \approx 1$: Well clustered
- $s(i) \approx 0$: On border between clusters
- $s(i) \approx -1$: Misclassified

Overall Score:

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

Appendix: PCA for Cluster Visualization

Dimensionality Reduction



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