Week 1: Innovation Foundations

Supervised vs Unsupervised Learning

BSc Course - ML for Design Thinking

Machine Learning & Generative AI for Innovation

Week 1 of 12

Week 1 Overview

Core Concept: Understanding the fundamental difference between supervised and unsupervised learning paradigms. Foundation for systematic innovation through machine learning approaches.

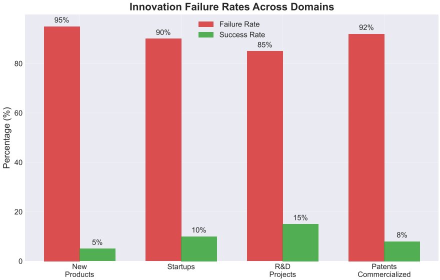
The Innovation Challenge

Opening Problem

The Challenge:

- 95% of new products fail
- Traditional methods insufficient
- Need systematic approach
- Data-driven insights required

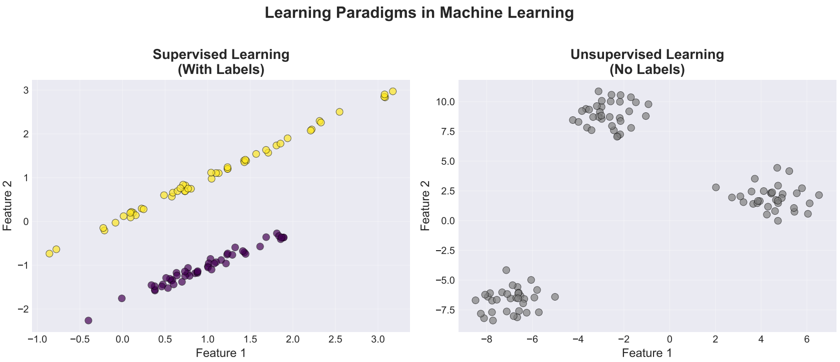
Core Question:
How can we systematically innovate when 95% of new products fail?



This Week: Foundation of ML thinking. Two fundamental learning paradigms. Integration with design thinking process.

Learning Paradigms in ML

Supervised vs Unsupervised Learning



Key

insight: Supervised learning uses labeled data, unsupervised discovers hidden patterns

Applications: Supervised for prediction and classification. Unsupervised for exploration and discovery. Both essential for comprehensive innovation strategy.

Core Concepts

Supervised Learning:

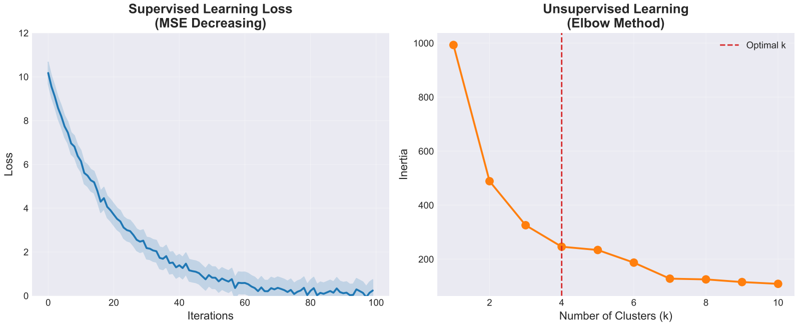
- Function: $f : X \to Y$
- Training with labels
- Examples: Email spam detection, success prediction
- Use: When outcomes are known

Unsupervised Learning:

- Function: $f : X \to Z$
- No labels required
- Examples: Customer segmentation, pattern discovery
- Use: Exploring unknown structures

$$\text{Supervised Loss: } L = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \quad (1)$$

Mathematical Foundations



$$\text{Supervised:} \quad \min_{\theta} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) + \lambda R(\theta) \tag{2}$$

$$\text{Unsupervised:} \quad \min \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2 \tag{3}$$

When to Use Which Method

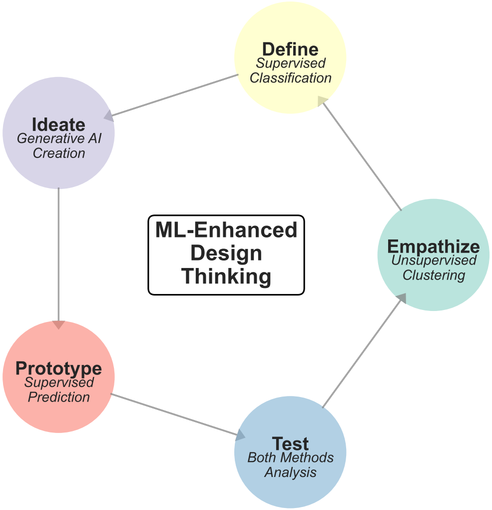
Problem Type	Learning Method	Example Application
Prediction	Supervised	Will this design succeed?
Classification	Supervised	Categorize user feedback
Discovery	Unsupervised	Find hidden patterns
Segmentation	Unsupervised	Group similar users
Generation	Generative AI	Create new ideas
Optimization	Reinforcement	Find best sequence

Decision Framework: Use supervised when labels exist. Use unsupervised for exploration. Combine both for comprehensive analysis.

Integration with Design Thinking

ML-Enha

Integration of ML Methods with Design Thinking Process



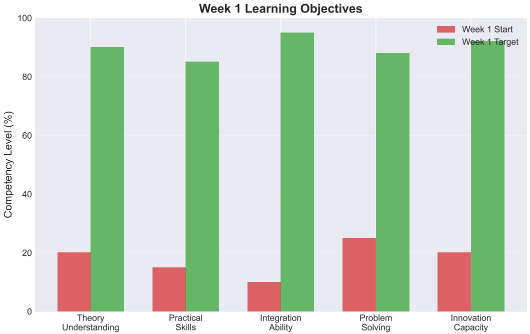
Design Thinking Stages & ML Methods

Five Stages:

1. Empathize
2. Define
3. Ideate
4. Prototype
5. Test

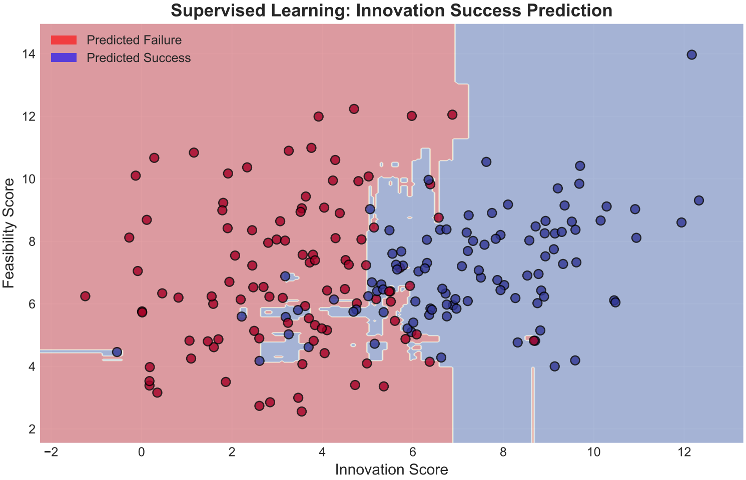
ML Enhancement:
Each stage augmented with appropriate ML technique

Stage	ML Method
Empathize	Clustering (unsupervised)
Define	Classification (supervised)
Ideate	Generative AI
Prototype	Prediction (supervised)
Test	Both paradigms



Practical Applications

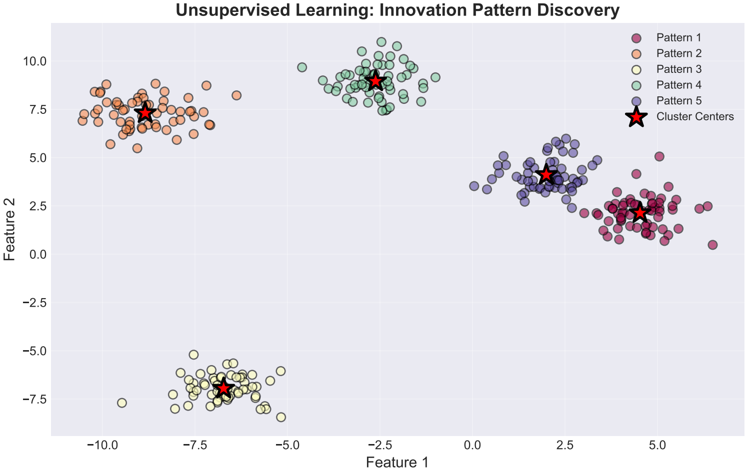
Supervised Learning: Decision Boundaries



Innovation success prediction using supervised classification

Application: Predict which innovations will succeed based on features like innovation score and feasibility. Random Forest classifier creates non-linear decision boundaries.

Unsupervised Learning: Pattern Discovery



Discovering hidden innovation patterns without labels

Discovery: K-Means clustering reveals 5 distinct innovation types. No prior labels needed. Patterns emerge from data structure alone.

Algorithm Fundamentals

Random Forest Classifier:

1. Bootstrap sample from training data
2. Build decision tree with random feature subset
3. Repeat for n_estimators trees
4. Aggregate predictions by voting

Key Parameters:

- n_estimators: Number of trees (typically 100-500)
- max_depth: Tree depth (controls overfitting)
- min_samples_split: Minimum samples to split node

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), ..., h_n(x)\} \tag{4}$$

Unsupervised Learning Algorithm

K-Means Clustering:

1. Initialize k centroids randomly
2. Assign each point to nearest centroid
3. Update centroids as mean of assigned points
4. Repeat until convergence

Convergence Criteria:

- Centroids stop moving
- Maximum iterations reached
- Inertia improvement below threshold

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \tag{5}$$

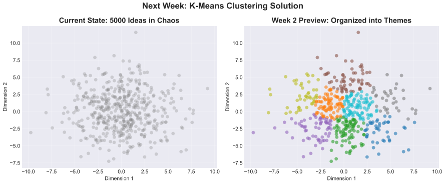
Temperature Control in Generation:

Temperature Effects:

- $T \to 0$: Deterministic
- $T = 1$: Balanced
- $T \to \infty$: Random

Applications:

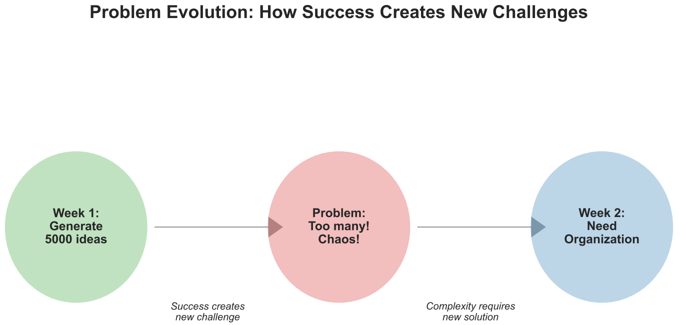
- Low T: Focused ideation
- Medium T: Diverse ideas
- High T: Creative exploration



$$P'(w_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{6}$$

Emerging Challenge

Success Creates New Problems



Problem Evolution: Week 1 success generates 5000+ ideas. Too many to handle manually. Need systematic organization. We will introduce K-Means clustering solution.

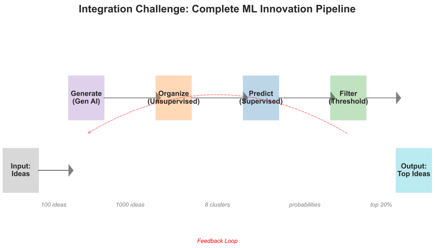
The Chaos Problem

Current State:

- 5000+ generated ideas
- No organization
- Hidden relationships
- Overwhelming complexity

Some Solution:

- K-Means clustering
- Automatic categorization
- Theme identification
- Systematic organization



Summary

Key Takeaways

Concepts Learned:

- Supervised vs Unsupervised
- When to use each method
- Mathematical foundations
- Design thinking integration
- Practical applications

Skills Developed:

- Identify learning paradigms
- Choose appropriate methods
- Understand algorithms
- Apply to innovation

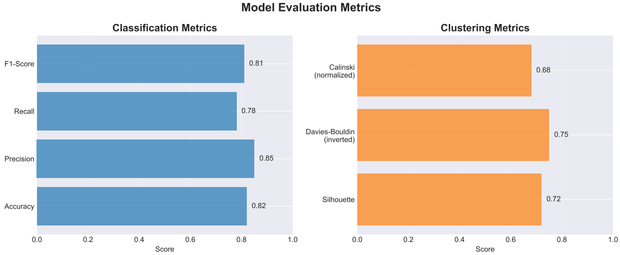
Key Insights:

1. ML transforms innovation
2. Supervised predicts success
3. Unsupervised finds patterns
4. Generative AI amplifies ideas
5. Integration multiplies impact

Problem Chain:
Success → Too many ideas → Need organization

Learning Objectives

Objective	Status
Understand supervised learning	✓
Understand unsupervised learning	✓
Differentiate use cases	✓
Apply to design thinking	✓
Recognize problem evolution	✓



Next Week: K-Means Clustering - Organizing the chaos of 5000 ideas

Assignment

Individual Tasks:

1. Compare supervised and unsupervised learning approaches
2. Identify three real-world applications for each paradigm
3. Analyze when to use which method
4. Prepare for K-Means clustering next week

Group Discussion:

- How does ML enhance traditional design thinking?
- What are the limitations of each learning paradigm?
- How can we combine both for maximum impact?

Reading: Bishop, Pattern Recognition and Machine Learning, Chapter 1. Hastie et al., Elements of Statistical Learning, Sections 2.1-2.3

Mathematical Details

Supervised Learning - Gradient Descent:

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta L(\theta_t) \tag{7}$$

$$\nabla_\theta L = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \ell(f_\theta(x_i), y_i) \tag{8}$$

Unsupervised Learning - K-Means Objective:

$$J = \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \mu_j||^2 \tag{9}$$

$$\frac{\partial J}{\partial \mu_j} = -2 \sum_{i \in C_j} (x_i - \mu_j) = 0 \tag{10}$$

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i \tag{11}$$

Complexity Analysis

Algorithm	Time Complexity	Space Complexity
Random Forest	$O(n \cdot m \cdot \log n \cdot t)$	$O(n \cdot t)$
K-Means	$O(n \cdot k \cdot i \cdot d)$	$O(n \cdot d)$
Neural Network	$O(n \cdot l \cdot m^2)$	$O(l \cdot m^2)$

Where: n = samples, m = features, t = trees, k = clusters, i = iterations, d = dimensions, l = layers

Practical Considerations:

- Random Forest scales well with data size
- K-Means sensitive to initialization
- Deep learning requires large datasets
- Choose based on data characteristics

Common Pitfalls and Solutions

Problem	Symptom	Solution
Overfitting	High train, low test accuracy	Regularization, cross-validation
Wrong k	Poor clustering	Elbow method, silhouette
Imbalanced data	Biased predictions	SMOTE, class weights
Feature scaling	Distorted clusters	StandardScaler
High dimensions	Curse of dimensionality	PCA, feature selection

Best Practices:

- Always visualize data first
- Use cross-validation for model selection
- Scale features for distance-based methods
- Check assumptions before applying algorithms

Resources and References

Essential Reading:

- Bishop, C. (2006). Pattern Recognition and Machine Learning
- Hastie, T. et al. (2009). Elements of Statistical Learning
- Murphy, K. (2012). Machine Learning: A Probabilistic Perspective

Online Resources:

- scikit-learn documentation
- Andrew Ng's Machine Learning Course
- Fast.ai Practical Deep Learning

Tools Required:

- Python 3.8+
- scikit-learn, pandas, numpy, matplotlib
- Jupyter notebooks