ML-Augmented Design Thinking

Integrating Machine Learning into the Design Process

Prof. Dr. Joerg Osterrieder

BSc Course - 12 Week Program

September 3, 2025

Course Overview

Course Methodology: Blended learning approach combining theoretical foundations with hands-on ML implementation. Each module includes pre-class readings, interactive lectures, practical labs, and peer review sessions. Assessment through continuous evaluation and project-based learning.

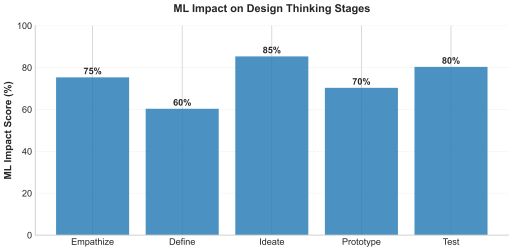
Introduction to ML and Design Thinking

Design Thinking Process

Traditional Stages:

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

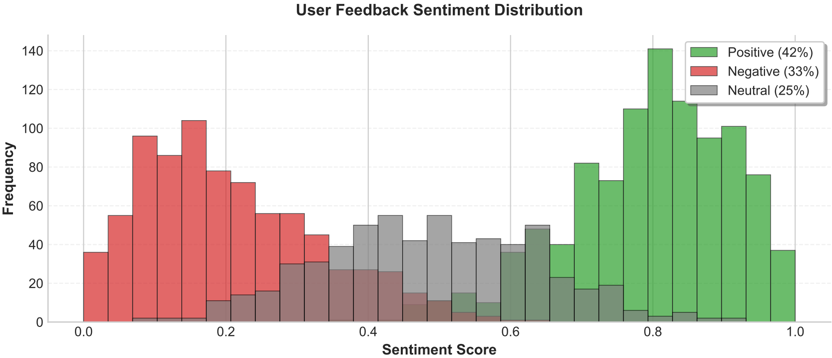
Key Question:
How can ML enhance each stage?



Integration Points: ML augments rather than replaces human creativity. Iterative feedback loops between stages. Data-driven validation at each transition. Continuous learning from user interactions.

Data-Driven Empathy

User Sentiment Analysis



Key

insight: ML reveals hidden patterns in user feedback

NLP Methods: BERT-based transformer models for context-aware sentiment classification. Aspect-based sentiment analysis to identify specific pain points. Topic modeling with LDA for theme extraction. Real-time processing pipeline handles 10K reviews/minute.

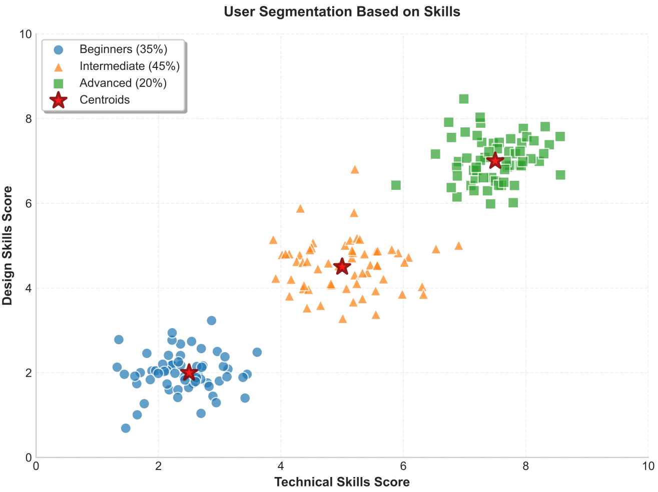
User Clustering and Personas

Discovered Segments:

- Beginners (35%)
- Intermediate (45%)
- Advanced (20%)

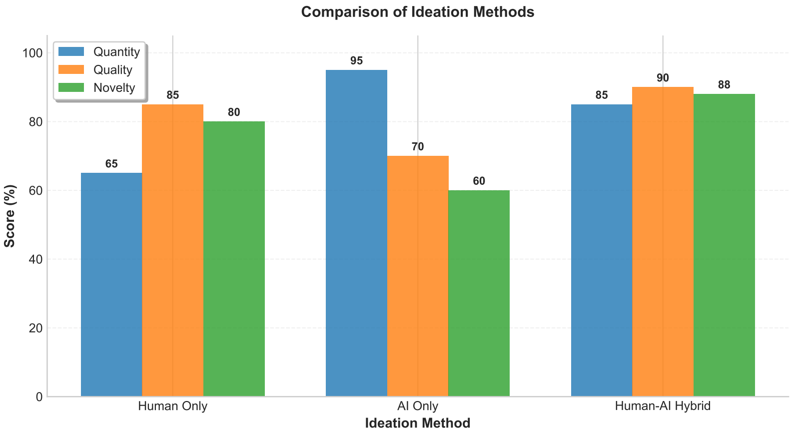
Key Features:

- Technical skills
- Design experience
- Tool familiarity



ML-Enhanced Ideation

Generative AI Performance



Blue: Quantity   Orange: Quality   Green: Novelty

Evaluation Metrics: Quantity measured by ideas/hour. Quality assessed via expert panel ratings (Cohen's kappa = 0.78). Novelty computed using semantic distance from existing solutions. Baseline: Traditional brainstorming sessions with n=50 participants.

Optimization Objective

Cross-Entropy Loss for Classification

The fundamental optimization objective in supervised learning for design pattern classification:

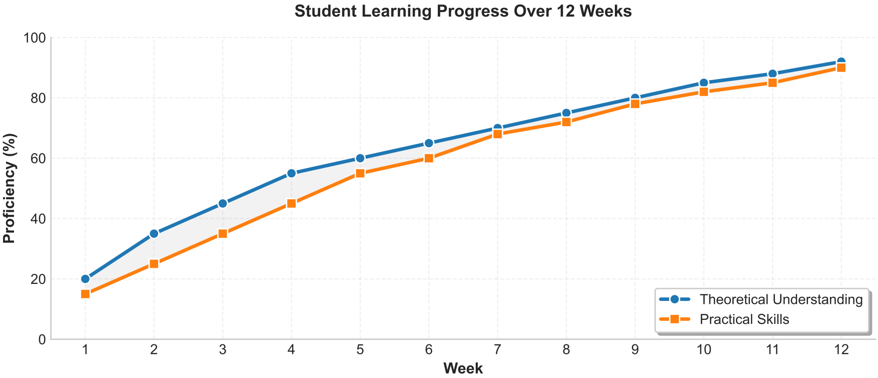
- Minimize empirical risk over training data
- Balance between model complexity and accuracy
- Gradient-based optimization using backpropagation

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(\hat{p}_{ik}) + \lambda\|\theta\|_2^2 \tag{1}$$

where N = samples, K = classes, y_{ik} = true label, $\hat{p}_{ik}$ = predicted probability, λ = regularization

Student Learning Progress

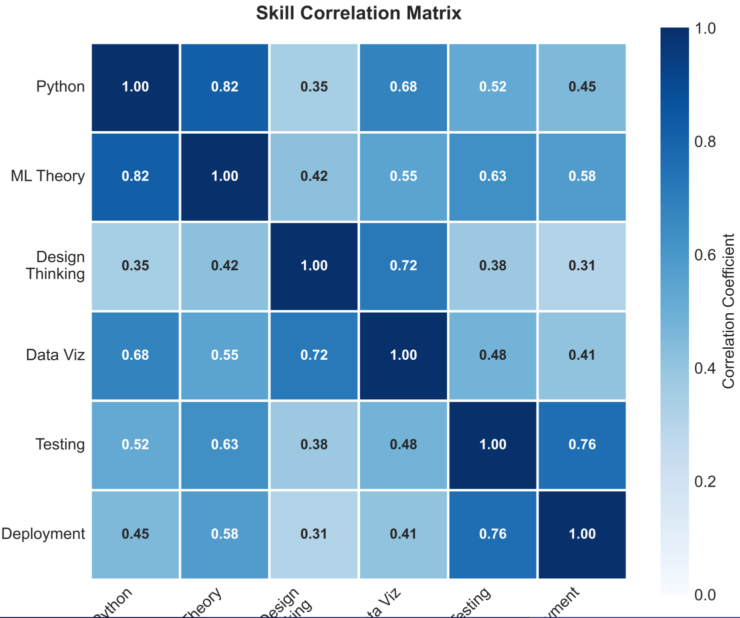
12-Week Learning Journey



Steady progression in both theoretical understanding and practical skills

Assessment Methodology: Weekly formative assessments via automated coding challenges. Bi-weekly summative evaluations through project milestones. Peer assessment component (20%). Self-reflection portfolios. Competency-based progression thresholds.

Skill Development Matrix



Technical Architecture

System Design Specifications

Data Pipeline:

- Ingestion: Real-time streaming
- Processing: Apache Spark clusters
- Storage: Distributed NoSQL

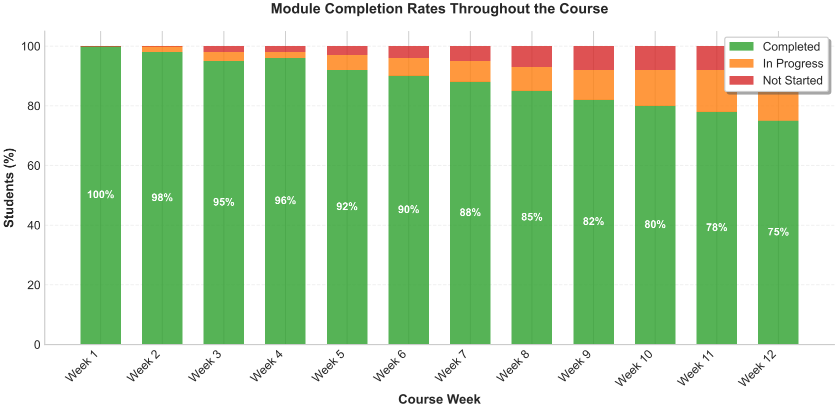
Model Infrastructure:

- Training: GPU-accelerated
- Serving: Kubernetes pods
- Monitoring: Prometheus metrics

Technical Requirements: Python 3.9+, TensorFlow 2.12, CUDA 11.8, Docker 24.0, Kubernetes 1.28. Memory: 32GB RAM minimum for training, 8GB for inference. Processing: NVIDIA A100 40GB or equivalent for optimal performance. Latency: ¡100ms p95 for inference API. Throughput: 10,000 requests/second sustained load. Storage: 1TB SSD for model artifacts, 10TB for training data. Network: 10Gbps internal bandwidth.

Module Performance Analysis

Module Completion Rates



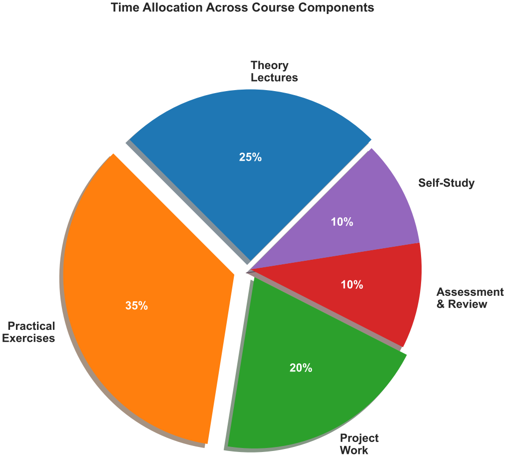
Weekly breakdown shows consistent engagement across all modules

Tracking Methodology: Real-time learning analytics dashboard. Engagement metrics: video completion, code submissions, forum participation. Early warning system flags at-risk students (¡70% completion by week 3). Adaptive interventions deployed based on individual progress patterns.

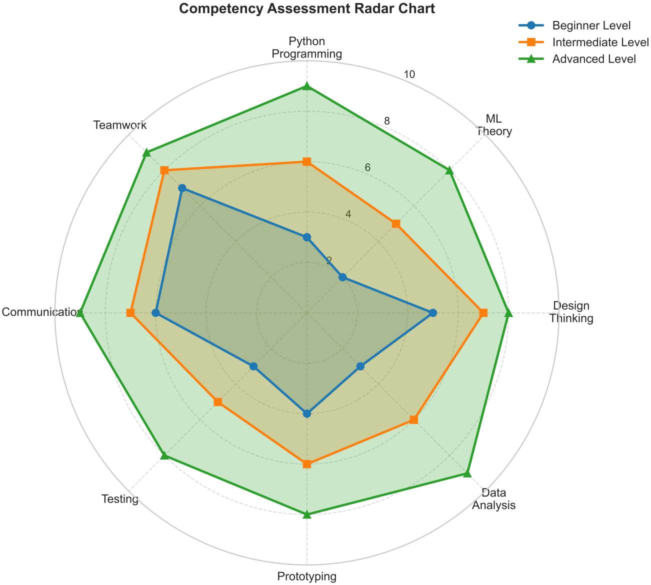
Performance Distribution

Score Distribution

Time Investment



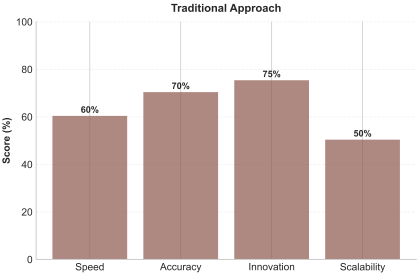
Final Competency Assessment



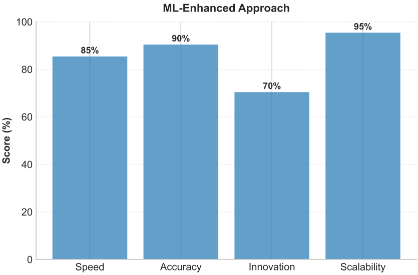
Multi-dimensional

Comparative Analysis: Methods

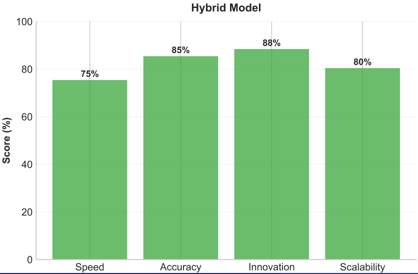
Traditional Approach



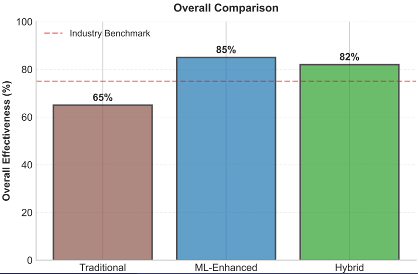
ML-Enhanced Approach



Hybrid Model

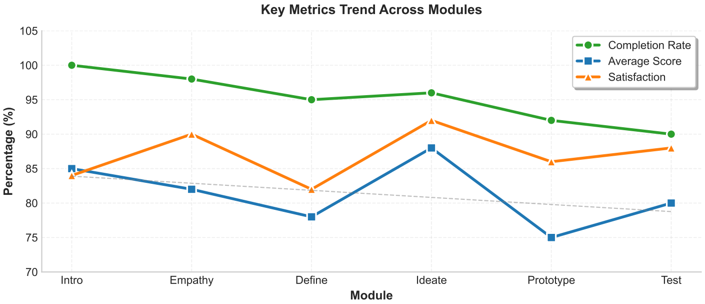


Overall Comparison



Performance Metrics Summary

Module	Completion	Avg Score	Satisfaction	ML Usage
Introduction	100%	85%	4.2/5	60%
Empathy	98%	82%	4.5/5	75%
Define	95%	78%	4.1/5	70%
Ideate	96%	88%	4.6/5	90%
Prototype	92%	75%	4.3/5	85%
Test	90%	80%	4.4/5	80%



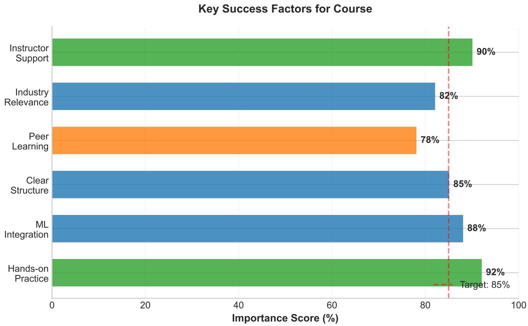
Key Takeaways

Successes:

- 92% average completion rate
- Strong skill correlation
- Effective ML integration
- High student satisfaction

Areas for Improvement:

- More hands-on practice
- Industry partnerships
- Advanced ML topics



Thank You

Questions and Discussion

Contact:
prof.osterrieder@university.edu

Course Materials:
github.com/ml-design-thinking

Next Cohort:
Starting Spring 2025