DADS7305: MLOPs

Northeastern University

Instructor: Ramin Mohammadi

September 7, 2025

These materials have been prepared and sourced for the course **MLOPs** at Northeastern University. Every effort has been made to provide proper citations and credit for all referenced works.

If you believe any material has been inadequately cited or requires correction, please contact me at:

`r.mohammadi@northeastern.edu`

Thank you for your understanding and collaboration.

The Machine Learning Project Lifecycle

Scoping

Scoping Process

What is scoping?

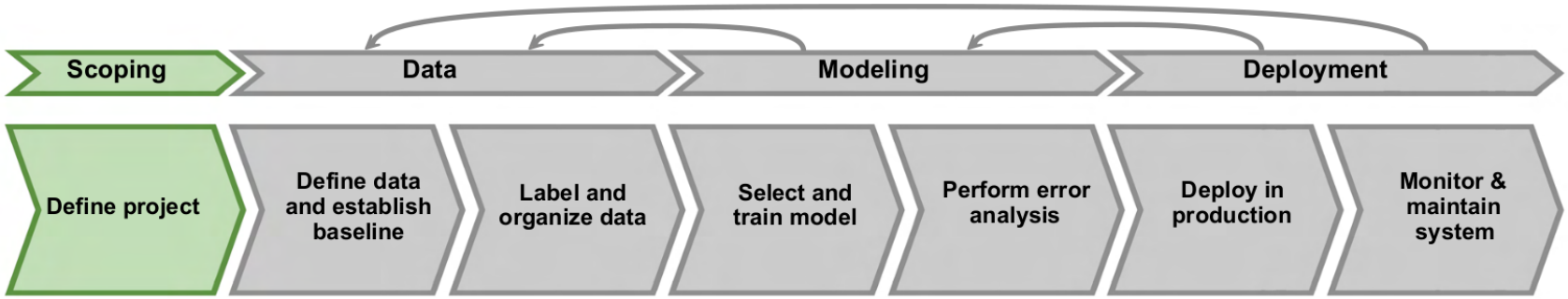


Figure: The ML project lifecycle

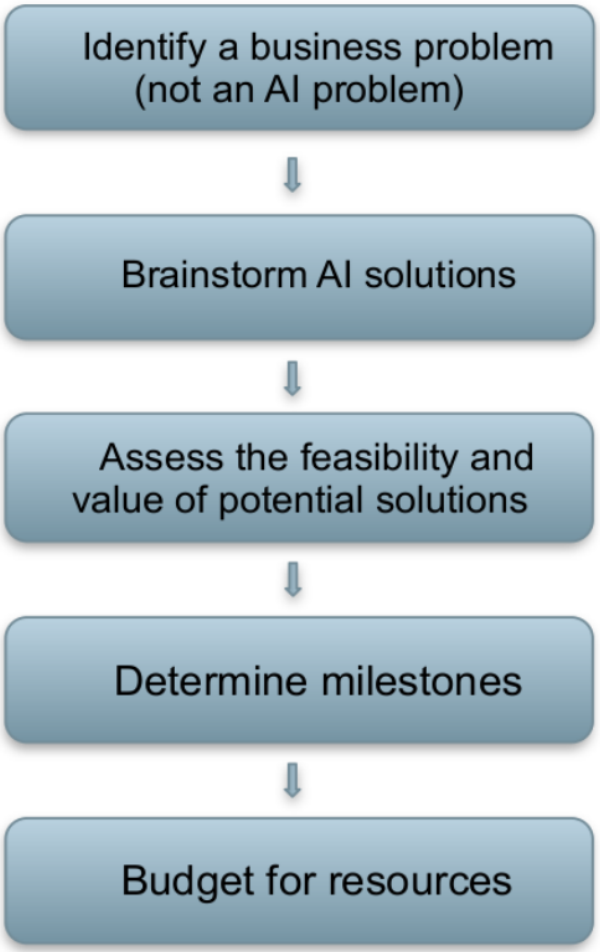
Scoping example: Ecommerce retailer looking to increase sales

- Better recommender system
- Better search
- Improve catalog data
- Inventory management item Price optimization

Questions:

- What projects should we work on?
- What are the metrics for success?
- What are the resources (data, time, people) needed?

Scoping process



What are the top 3 things you wish were working better?

- Increase conversion
- Reduce inventory
- Increase margin (profit per item)

Separating problem identification from solution

Problem	Solution
Increase conversion	Search, recommendations
Reduce inventory	Demand prediction, marketing
Increase margin (profit per item)	Optimizing what to sell (e.g., merchandising), recommend bundles
What to achieve	How to achieve

Scoping

Diligence on feasibility and value

Feasibility: Is this project technically feasible?

Use external benchmark (literature, other company, competitor)

	Unstructured (e.g., speech, images)	Structured (e.g., transactions, records)
New	HLP	Predictive feature available?
Existing	HLP History of project	New Predictive features? History of project

HLP: Can a human, given the same data, perform the task?

Why use HLP to benchmark?

People are very good on unstructured data tasks

Criteria: Can a human, given the same data, perform the task?

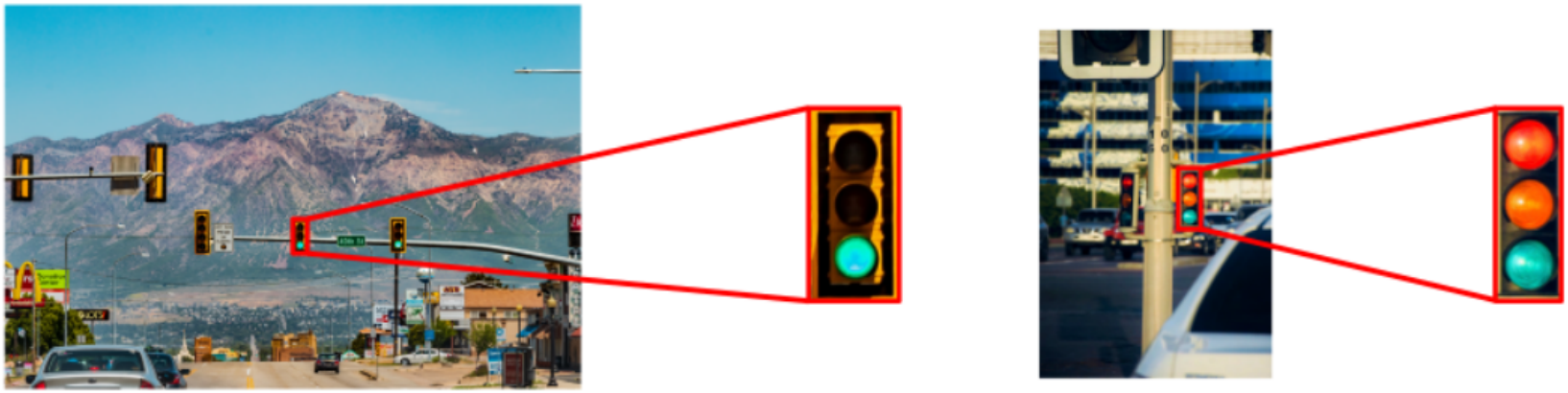


Figure: Data Example

Do we have features that are predictive?

Given pas purchase, predict future purchases

Given past purchases, predict future purchases

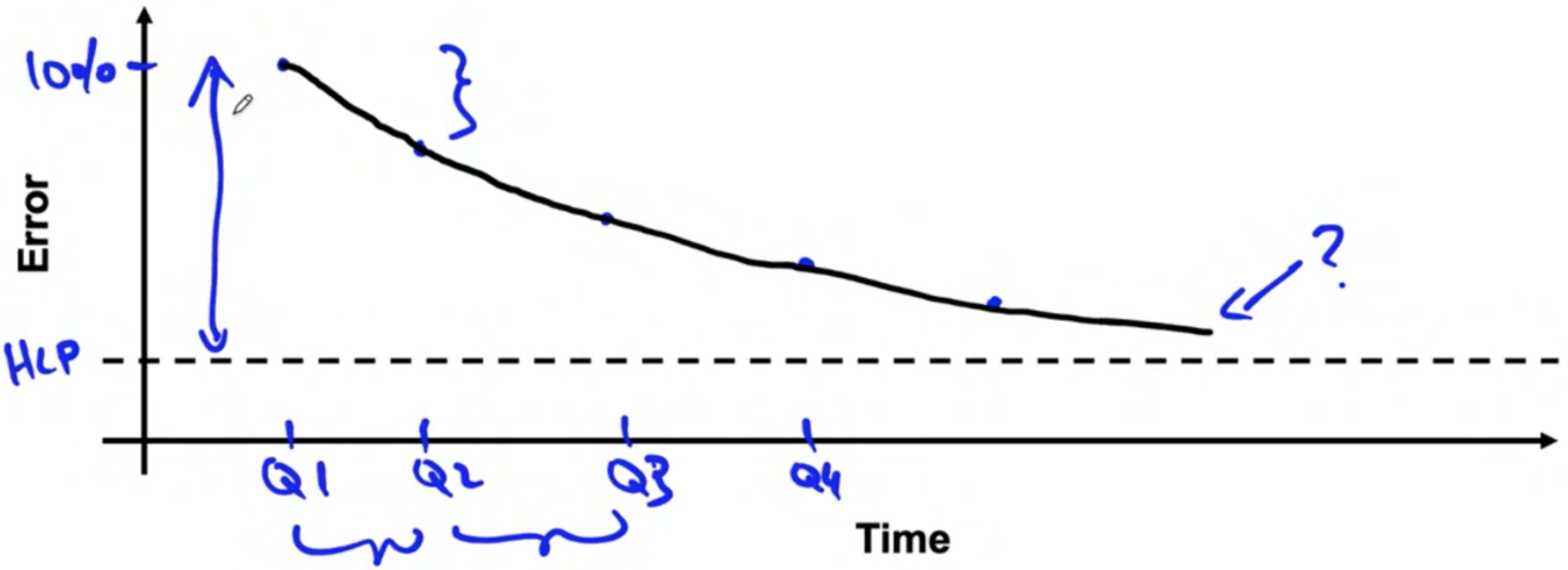
Given weather, predict shopping mall foot traffic

Given DNA info, predict heart disease

Given social media chatter, predict demand for a clothing style

Given history of a stock price, predict future price of that stock

History of project



Diligence on value

Have technical and business teams try to agree on metrics that both are comfortable with.

Ethical Consideration

Ethical considerations

- Is this project creating net positive societal value?
- Is this project reasonably fair and free from bias?
- Have any ethical concerns been openly aired and debated?

Scoping

Milestones and resourcing

Milestones

Key specifications:

- ML metrics (accuracy, precision/recall, etc.)
- Software metrics (latency, throughput, etc. given compute resources)
- Business metrics (revenue, etc.)
- Resources needed (data, personnel, help from other teams)

Timeline

If unsure, consider benchmarking to other projects, or building a POC (Proof of Concept) first.

Project Example

Scoping

Template

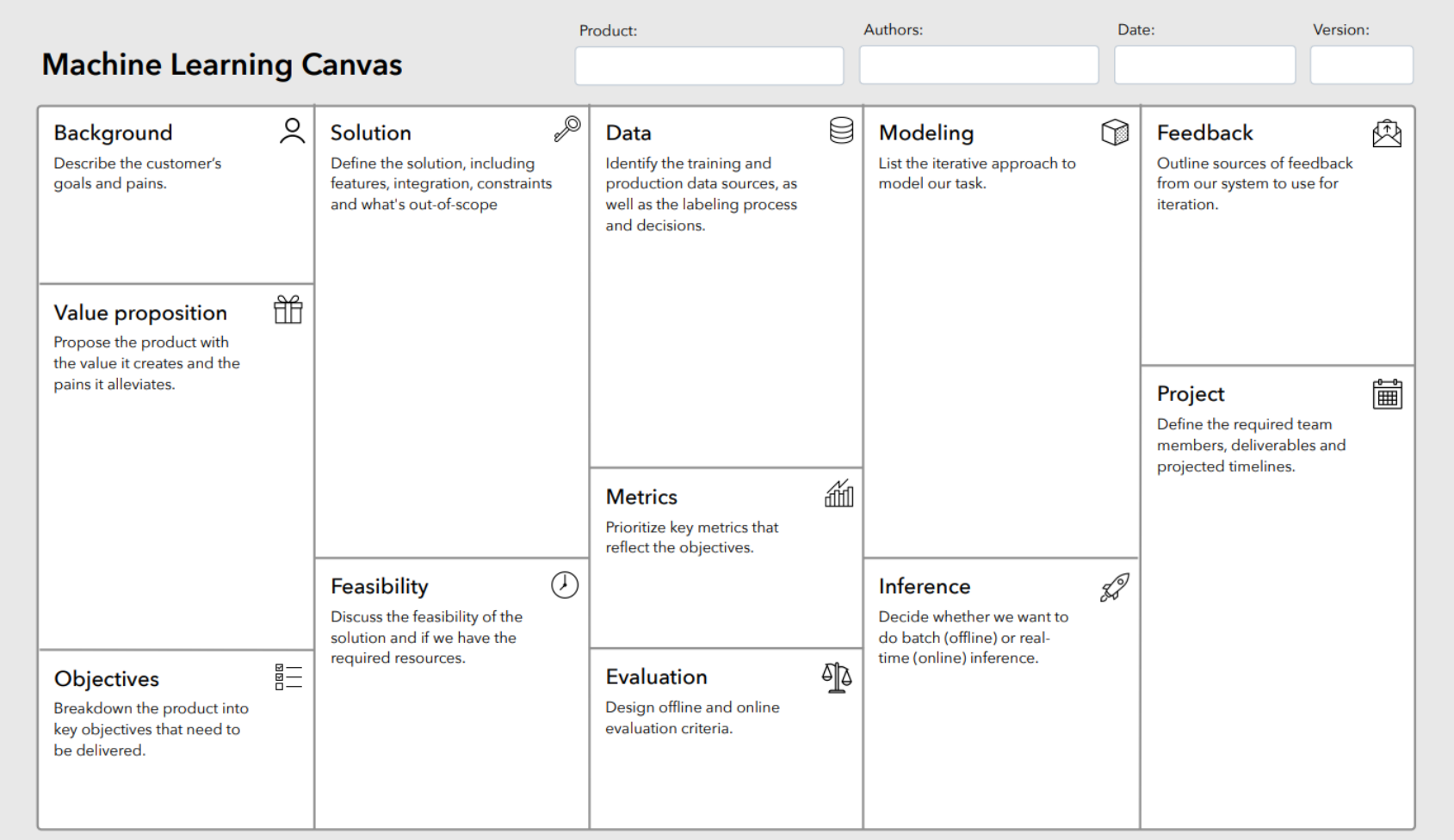


Figure: Machine learning canvas from Made With ML

LLMOps

LLM Scoping

5 Steps to LLM Project Success

Why Scope an LLM Project Carefully?

- ChatGPT has brought LLMs into the spotlight, now companies are eager to explore potential.
- Common questions:
  - Will it take 3 months or 1 year?
  - Will it cost \$50k or 10× more?
- Every company is different, but successful projects share similar scoping patterns.

Let's walk through 5 proven steps to LLM project success.

Step 1: Define a Use Case Around a Real Pain Point

Start small, but solve something real.

- Many LLM projects fail by aiming too broadly.
- Focus on one or two specific use cases where:
  - Processes are repetitive or inefficient
  - Domain-specific jargon is common
  - Outcomes directly impact business metrics
- Simplify your goals, use success as a stepping stone.

Step 1 (continued): Two Approaches to Use Case Discovery

Approach 1: Build New Capabilities

- Use your domain expertise to launch innovative LLM-powered tools.
- Aim to grow revenue through novel user experiences.

Approach 2: Streamline Existing Inefficiencies

- Automate or accelerate tasks that lead to churn or burnout.
- Example: manual reviews, policy checks, customer interactions.
- Not glamorous, but highly impactful.

Use Case Example: Enhancing Customer Service

Problem:

- Current chatbots and self-service tools often frustrate users.
- Customer Service Representative (CSRs) lack context and repeat information collection.

LLM Opportunity:

- Provide CSRs with dynamic, real-time customer history summaries.
- Enable personalized, efficient, and empathetic support.

Impact:

- Happier customers $\rightarrow$ brand advocates.
- Lower churn $+$ scalable CSR workload $\rightarrow$ business growth.

Step 2: Assemble a Cross-Functional Data Team

Scoping requires collaboration across disciplines:

- Involve data scientists, data engineers, software engineers, product managers, and stakeholders.
- Ensure product builders have direct access to end users for continuous feedback.
- If talent is lacking in-house, start with consultants while developing internal capacity.
- Plan ahead, technical and business teams define problems differently and need time to align.

Step 2 (continued): Why Collaborative Scoping Matters

Engineering $\neq$ Product Thinking

- Engineers think in terms of constraints and algorithms.
- Product owners focus on outcomes and user experience.
- A strong cross-functional team bridges this gap, ensuring:
  - Clear problem definitions
  - Realistic scoping
  - Faster iteration cycles

Step 3: Create a Data Product Requirements Document (DPRD)

The DPRD aligns business and technical teams.

- A machine learning–specific variant of the traditional PRD.
- Lightweight, readable, and serves as a shared roadmap.
- Avoid premature tech stack decisions; focus on clarity and flexibility.

Step 3 (continued): What to Include in a DPRD

Core Elements:

- **Purpose:** Define the shared end goal and success criteria.
- **Data Sources:** Internal, external, or hybrid, clearly specify.
- **Architecture Diagram:** Show data flow and system interactions.
- **Dependencies:** List people, teams, tools, and external blockers.
- **Evaluation Metrics:** Define how model performance will be measured.
- **Feedback/Monitoring Framework:** Include mechanisms for:
  - Tracking real-world usage
  - Detecting drift
  - Capturing end-user feedback

Step 4: Choose the Right LLM for the Job

Not all LLMs are created equal, select based on your needs:

- **Closed-source (e.g., ChatGPT):**
  - Powerful for general tasks, but:
  - Expensive, limited control, privacy concerns, unpredictable updates
  - Underperforms in narrow, domain-specific use cases
- **Open-source (e.g., Mistral, Huggingface):**
  - Can be fine-tuned with your data
  - Greater control over features and IP
  - Ideal for specialized, high-accuracy applications

Why Open-Source LLMs + Transfer Learning Win

Transfer learning makes domain adaptation accessible:

- Open-source LLMs are pretrained on large corpora
- Fine-tune on a small, domain-specific dataset
- Benefits:
  - Lower compute and data requirements
  - Higher accuracy for your specific use case
  - Full control over deployment and integration

Step 5: Cost Out the Whole Project

LLM projects require realistic costing beyond model training:

- Include both financial costs and developer time
- Off-the-shelf tools rarely offer full automation
- Avoid research-style, waterfall ML projects, think agile
- Value must be delivered incrementally and early

Agile Approach to LLM Project Execution

Break down the project into sprints:

- Tasks:
  - Data ingestion
  - Feature selection and engineering
  - Model selection and evaluation
  - Model deployment
  - Model monitoring
- Estimate 1–2 sprints per task (2–4 weeks)
- Build in time for continuous feedback and iteration

Outcome: Faster learning cycles, higher model quality, and a better chance at success.

Project Cost Considerations

Strategic trade-offs when planning your LLM solution:

- **Build vs. Buy**
  - Buying saves internal effort but limits flexibility and control.
  - Building enables customization and privacy, but requires internal or consultant expertise.
- **Open Source vs. Proprietary LLMs**
  - Open source: free or low-cost, with flexible licensing.
  - Proprietary: expensive, with potential privacy and vendor risks.
- **First-party vs. Third-party Data**
  - First-party data is often sufficient and cheaper.
  - Use third-party data only to fill clear gaps.
- **Compute Requirements**
  - Many domain-specific LLMs can be built on a single high-end laptop.
  - Large GPU clusters are not always necessary.

Overview: Defining Success Through User Needs

Why AI Products Fail

Even the most advanced AI will fail if it doesn't deliver **unique value** to users.

Human-Centered AI

Ask First:

- Who are your users?
- What are their values?
- Which problem should you solve for them?
- How will you solve it?
- How will you know when you're done?

What This Section Covers (Part 1)

- **Which user problems are AI systems uniquely positioned to solve?**
- **What problem should you solve for people with AI?**
- **How and when can we augment human capabilities vs. automate tasks?**

Why These Questions Matter

These decisions shape your product's core value and guide every downstream decision.

What This Section Covers (Part 2)

- **How can we ensure a helpful user experience when people interact with AI systems?**
- **How can we anticipate potential pitfalls and prepare to course-correct?**
- **How do we define success, total and over time?**

Defining Success

User experience and iteration planning are where long-term value is created.

Identify user needs and AI strengths

⚖ Balance People-First and Tech-First Thinking

Key Principle

Balance AI capabilities with user needs.

People-First Approach:

- Identify real user problems that need solving.
- Validate your assumptions: *what*, *for whom*, and *in what context*.

Technology-First Approach:

- Start with capabilities and limitations of the tech.
- Identify product areas where AI can enhance the experience.
- Example: Using a safety classifier to improve recommendations.

Both approaches are valid, use research, data, and user observation to find the right balance.

Decide if AI Adds Unique Value

Once the problem is identified, ask:

- Does solving it require AI?
- Can AI meaningfully enhance the solution?
- What type of AI is most appropriate?
- Will AI degrade the experience in any way?

Example Consideration

High-value tasks may be disrupted by automation, causing frustration for users who prefer manual control.

Critically assess whether AI improves or regresses the user experience. Don't default to using AI, validate its value.

⚖ Consideration: Stakes

What Are the Stakes?

Situations can have high or low stakes. Imagine the types of situations where people may use your system's outputs, and assess how much they should trust it to be:

- Accurate and precise (e.g., safety-critical systems), or
- Creative and comprehensive (e.g., generative assistants)

Consideration: User Context

Understanding User Expectations

There are often social expectations around how people interact with technology that shape whether an AI solution is seen as appropriate.

Imagine how users will respond to AI output in a likely real-world context.

$ Consideration: Cost

Weighing the Cost of AI Solutions

AI solutions can be resource-intensive:

- Require more energy and compute power
- May be more expensive than rule-based or human alternatives

Always assess cost-benefit tradeoffs in production environments.

Understand the Nature of the Problem

Start With a Holistic View

Before designing a solution, ensure you fully understand the problem space.

- Clearly define the problem you aim to solve.
- Validate your framing with diverse users.
- Interview "extreme users" to explore range and pervasiveness.
- Probe for unknowns: frequency, context, severity, and triggers.
- Consider how the problem may evolve with AI in the loop.

People's Experiences Vary Widely

Why Diversity Matters

People experience the same problem differently based on their identity.

- Consider socio-economic status, language, location, gender, and ability.
- Gather feedback at key inflection points in development.
- Identify sub-problems AI must solve across dimensions of identity.

Inclusive input improves generalizability, trust, and usefulness.

AI in Context: A Systems Perspective

AI Models Interact With Broader Systems

Effective AI design considers how models function within full ecosystems.

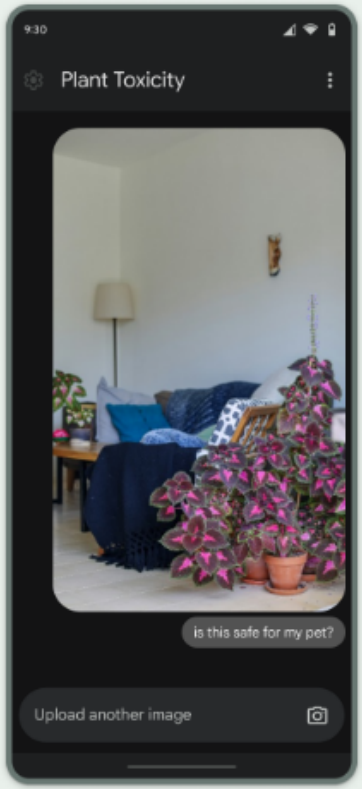
- AI often works alongside safety classifiers, RL, and rule-based logic.
- Model cards and data cards help evaluate performance and usability.
- Use AI to support, not disrupt, user goals in specific contexts.

Example

EXAMPLE

AI for different tasks

A user takes an image of a plant to see if it's safe for their pet. But there can be challenges if there are multiple plants in the image, or if the AI model can't identify the plant. Click through the images to learn more.



Example

EXAMPLE

AI for different tasks

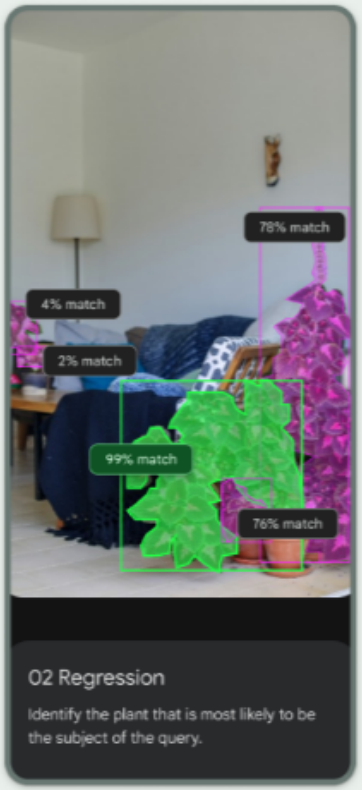
There could be many steps involved in arriving at this response. First, a computer vision 'segmentation' model needs to separate all plants from the background and all other objects in the image.

Example

EXAMPLE

AI for different tasks

Next, a regression system needs to identify the plant that is most likely the subject of the user query.



Example

EXAMPLE

AI for different tasks

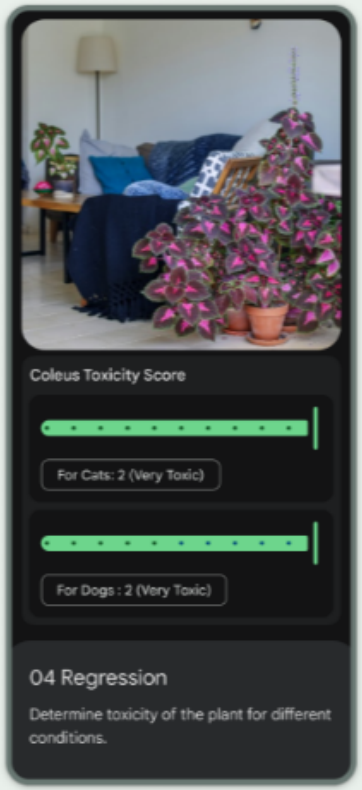
Then, a classification system designed specifically to identify plants needs to determine what kind of plant it is.

Example

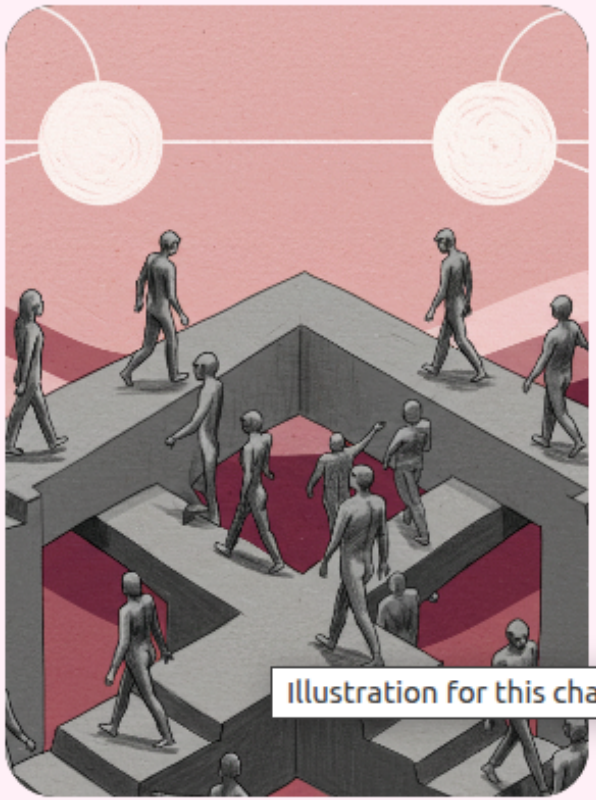
EXAMPLE

AI for different tasks

Finally, the product may use another regression system to determine the likelihood of the plant's toxicity.



Guiding Questions



GUIDING QUESTIONS

Understanding The Nature Of The Problem

- How pervasive is the problem?
- Who does it affect?
- How does this problem affect various individuals?
- How does this problem vary across different groups and dimensions of identity?
- Does the problem's impact change over time?

Understand How People Frame Problems

Deterministic vs. Open-ended Problems

People often solve problems creatively, improvising with their context and lived experience.

- Understand user expectations for what it means for a problem to be "solved."
- Use this to define success criteria for your AI system.
- Align the AI's behavior with real-world user goals.

Defining the system's purpose clearly is critical for building trustworthy AI.

Specification Alignment: Matching AI to Intent

- Users express their goals through a process called **specification**.
- **Specification alignment** is guiding AI to deliver on user intent.
- Misalignment causes:
  - Poor performance,
  - Compromised safety,
  - Reward hacking (AI learns unintended goals).

UX teams and engineering must collaborate to ensure the AI behaves in ways that feel natural and useful.

Four Key Dimensions for Framing

Primary Goal

- What does the user want the AI to accomplish?

Sub-goals

- What dependencies or side-tasks must be solved along the way?

Underspecification

- What assumed knowledge is left out? How will you uncover it?

Optimization

- Can optimizing one sub-goal compromise another?
- Could the AI learn harmful behaviors from these trade-offs?

Designing for Dimensions of Identity

Bias Risks in AI

AI may have uneven performance across cultures, contexts, and identity groups.

- Bias can result from unrepresentative data or biased labels.
- Lack of evaluation on accepted bias benchmarks (e.g., *BBQ, WinoBias*) can mask issues.
- Understand how different communities perceive utility, intent, and consequences.

Bias can be subtle, systemic, and deeply contextual, plan for it.

Inclusive Design: Diverse Perspectives Matter

- Different people experience the same problem in very different ways.
- Marginalized groups may be underserved, or harmed, by generalized models.
- Over-corrections can also marginalize majority groups.

Best Practices

- Incorporate diverse viewpoints early and throughout the design cycle.
- Run adversarial and fairness testing before launch.
- Avoid unintentionally excluding or displacing communities, cultures, or professions.

Key Dimensions of Identity for Generative AI

Why It Matters

Generative AI outputs are unpredictable. Identity-aware testing helps reduce stereotype risks and hallucinated bias.

Common Identity Dimensions:

- Age, Disability, Education and Literacy
- Geography, Socioeconomic Status, Technical Proficiency
- Culture, Ethnicity, Religion, Sexual Orientation
- Physical attributes (e.g., size, clothing)

Consider intersectionality and the goals of your product when evaluating fairness.

Guiding Questions

GUIDING QUESTIONS

Understanding How People Currently Frame the Problem

- How do people think about the problem?
- How do they experience the problem?
- How does their socio-technical context impact how they frame or experience this problem?
- How are these similar or different across the dimensions of identity?

Traditional AI , Classification, Regression, & Recommenders

Overview

Leverages existing, cataloged information to select content and/or make predictions about new examples.

Good uses for these systems:

- **Automations such as optimizing** for efficiency
  E.g., smart thermostats improving comfort and energy use.
- **Recognize entire classes of entities**
  E.g., face detection, language identification.
- **Show dynamic content** for improved relevance
  E.g., auto-generating online ads from small business websites.
- **Recommend different content** to individuals
  E.g., personalized movie or song suggestions.
- **Predict events** using historical data
  E.g., crowd prediction for locations.
- **Detect low occurrence events** with evolving patterns
  E.g., fraud detection over time.
- **Provide predictable bot- or agent-like experiences**
  E.g., booking systems, customer call routing.

Generative AI , LLMs, Diffusion, & Frontier Models

Overview

A type of AI that generates new content, such as text, images, or other media, based on input(s).

Good uses for these systems:

- **Reduce user fatigue** in expert tools by simulating complex scenarios.
- **Produce dynamic responses** by summarizing from multiple sources.
- **Answer context-sensitive questions** using multimodal inputs.
- **Augment user creativity** across text, image, or design workflows.
- **Perform collaborative tasks** like remixing or ideation with users.
- **Provide a contained agent experience** for low-risk personalization tasks.

Rule-Based Systems , Heuristics

Overview

Uses predefined rules or logic to make decisions without AI-based inference or learning.

Good uses for these systems:

- **Maintain predictability** so users can safely undo or navigate actions.
- **Provide stored/limited info** with consistent formats (e.g., forms).
- **Minimize costly errors** where risks of wrong predictions are high.
- **Offer simple explanations** for transparency and trust.

Understand How People Solve the Problem

Designing for Generative AI

Generative AI requires special care for open-ended, creative use cases.

- Generative models can create text, images, code, or media.
- Users interact via natural language , not linear menus.
- This makes the interface less predictable, but more flexible.

Design must accommodate improvisation, not just decision trees.

Map Workflows to Discover Opportunities for AI

Step 1: Understand the current process

- Walk through how users complete the task today.
- Identify steps that could be automated or enhanced by AI.

Step 2: Investigate adjacent products

- Study how users interact with similar AI or non-AI tools.
- Use service design templates to document pain points.

Step 3: Prototype and validate early

- Test assumptions via *Wizard of Oz* or with Google AI Studio.
- Let user feedback reveal root causes and real opportunities.

User research fuels more relevant, trustworthy AI experiences.

Map Existing Workflows

Formal and Ad-hoc Patterns

Assess how people interact with both AI- and non-AI tools that address the problem, even partially.

- Observe formal workflows *and* informal workarounds.
- Note how users react when features are unclear or misaligned with expectations.
- Include user pain points to discover unmet needs.

Even failed attempts or misuse may reveal valuable signals.

Find Similarities and Variations in Tasks

- Look for clusters of related tasks to understand common patterns.
- Identify variations within a single task that could affect AI performance.
- Use these findings to inform the **range and flexibility** required in your system.

Generalization starts with understanding variation.

Map Points of Decision and Feedback

- Identify where users make key decisions in the journey.
- Highlight moments where feedback is expected, explicitly or implicitly.
- These points often represent **opportunities for intervention** or AI support.

User decisions shape how your AI should respond or guide.

Critical Moments and Failures in the Journey

- Identify where users interact directly with AI capabilities.
- Open-ended interactions need **extra design care**.
- Document errors users will tolerate, and those they won't.
- These inform which failures are acceptable, recoverable, or unacceptable.

Understanding user tolerance helps shape safe and trusted AI behavior.

Labs for This Week

Objective

Learn how to setup an API for your application and add logging to your pipeline.

Lab Activities:

- Lab 1: [FastAPI] — [FastAPI Tutorial]
- Lab 2: [Logging] — [Logging Tutorial]

Submission Deadline: [Before the next class]

- Assignment 2: [FastAPI] — [Create a API of your choice]

Reading Materials

This Week's Theme

Topic focus: [Challenges in Deploying Machine Learning: a Survey of Case Studies]

Required Readings:

- [Challenges in Deploying Machine Learning: a Survey of Case Studies]

Be prepared to discuss highlights and open questions in class.


DeepLearning.AI


The People + AI Guidebook