

# **Data Science, AI, and Machine Learning in Public Health using R**

## **All Slides**

January 2026

Presented By: Wronski Associates

# Introduction: Directing AI Use in the Federal Government

## MOVING BARRIERS TO AMERICAN LEADERSHIP IN ARTIFICIAL INTELLIGENCE

The White House | January 23, 2025

### CDC's Vision for Using Artificial Intelligence in Public Health



Public Health  
AUG. 22, 2025

CDC is committed to using artificial intelligence/machine learning for innovation, operational efficiency, and fighting infectious disease. CDC's artificial intelligence innovation approach includes investment areas, partnerships, workforce readiness, and guidance.

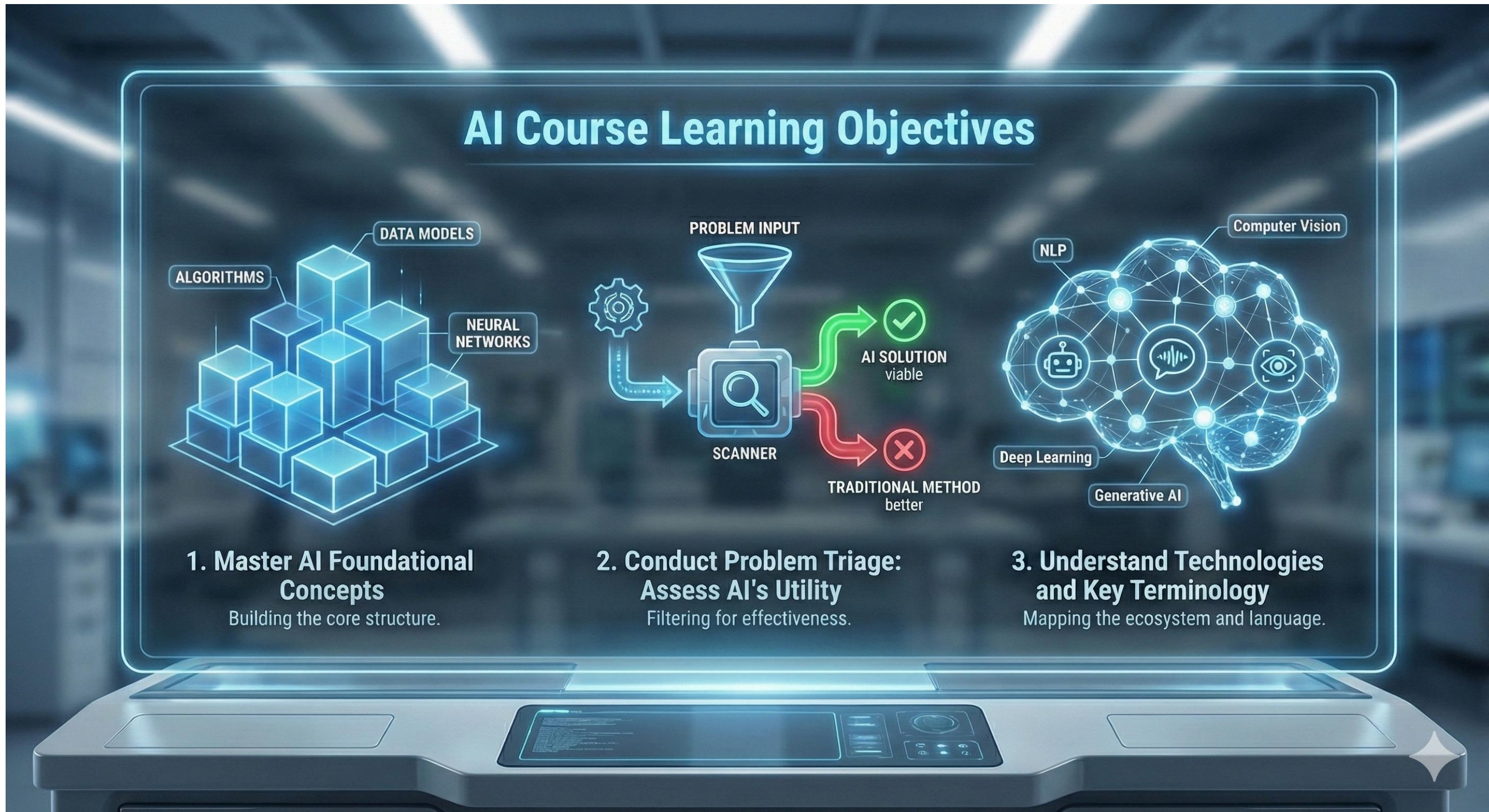
- **M-25-21:** "Accelerating Federal Use of AI through Innovation, Governance, and Public Trust" (April 3, 2025)
- **M-25-22:** "Driving Efficient Acquisition of Artificial Intelligence in Government" (April 3, 2025)
- **Executive Order 14319** - "Preventing Woke AI in the Federal Government" (July 23, 2025)
- **America's AI Action Plan** (July 2025)

Source: The White House

To do what, exactly?



# That is what we are here to figure out

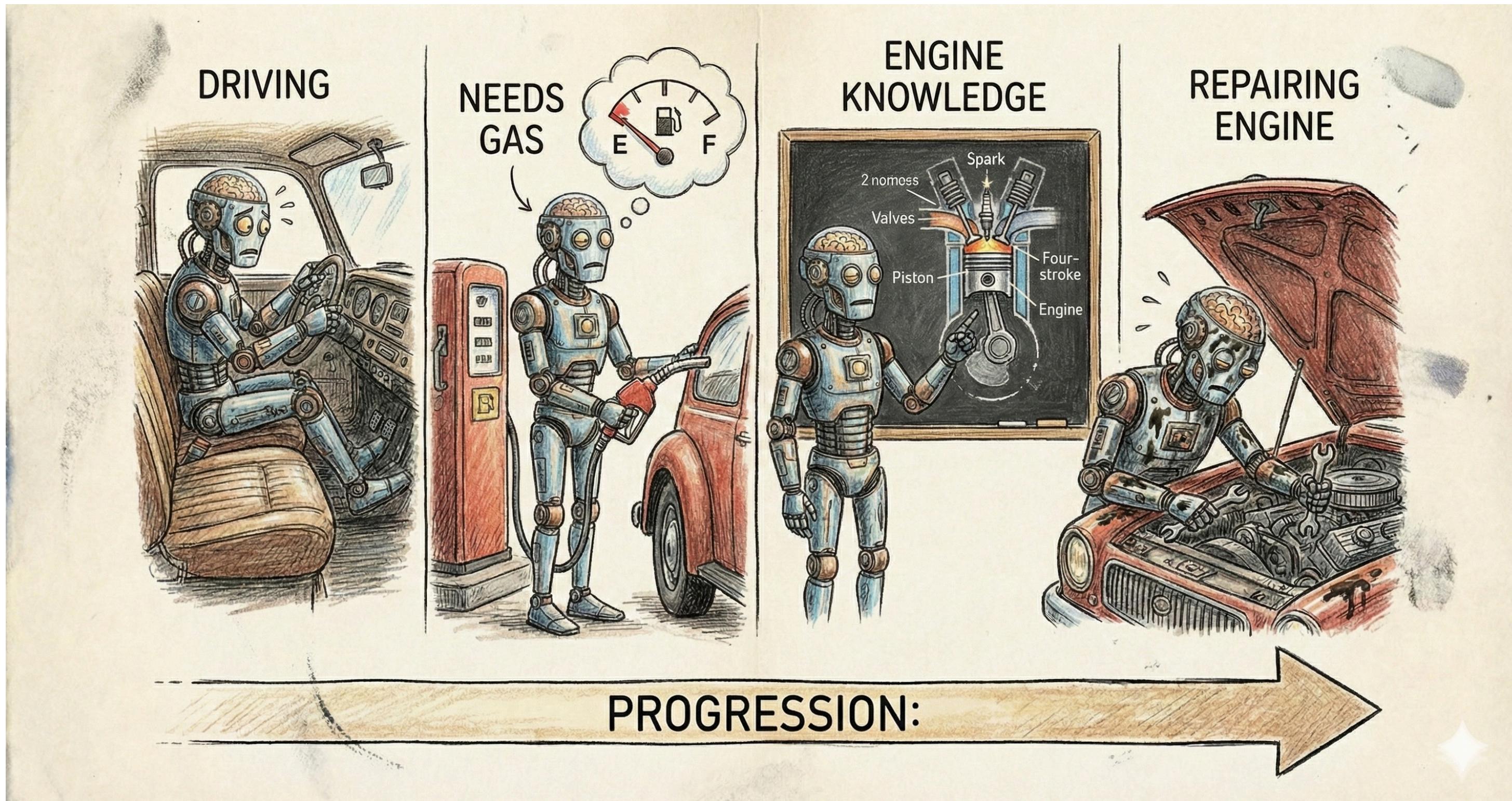


*Prompt: Create an image related to AI. The image presents the learning objectives for a course. The learning objectives are: Learning AI Foundational Concepts, Problem Triage - Can AI be useful?, and Technologies and Terminology.*

# 3 minute assignment

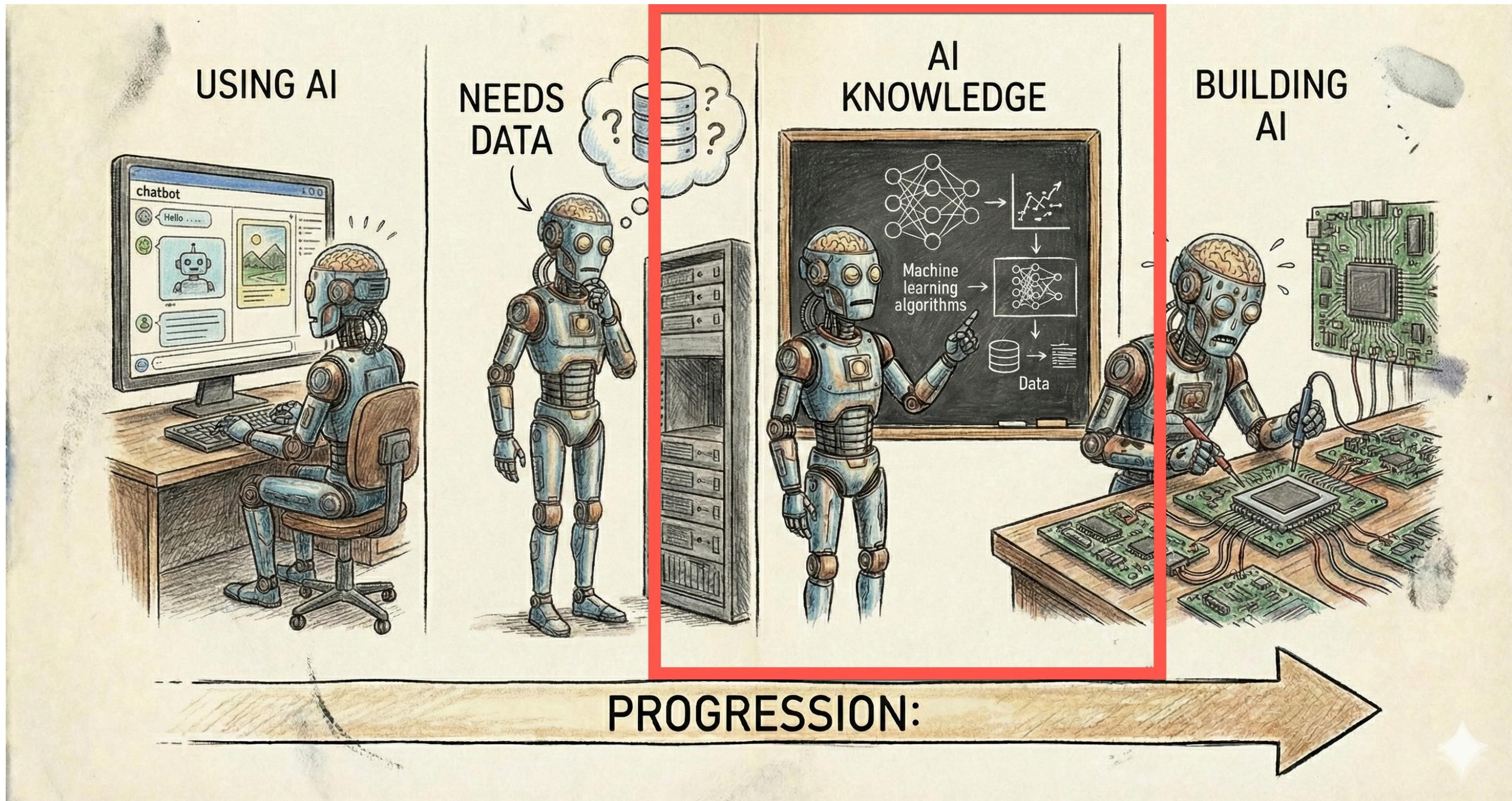
Think of a problem, opportunity, project that you have that could, potentially, be solved using AI.

# Level of knowledge



*Prompt: Use the robot, hand drawn style. Create an image of images that portray these ideas: driving, knowing the car needs gas, knowing, at a high level, how a combustion engine works, repairing a car engine. indicate there is a progression from the first to the last.*

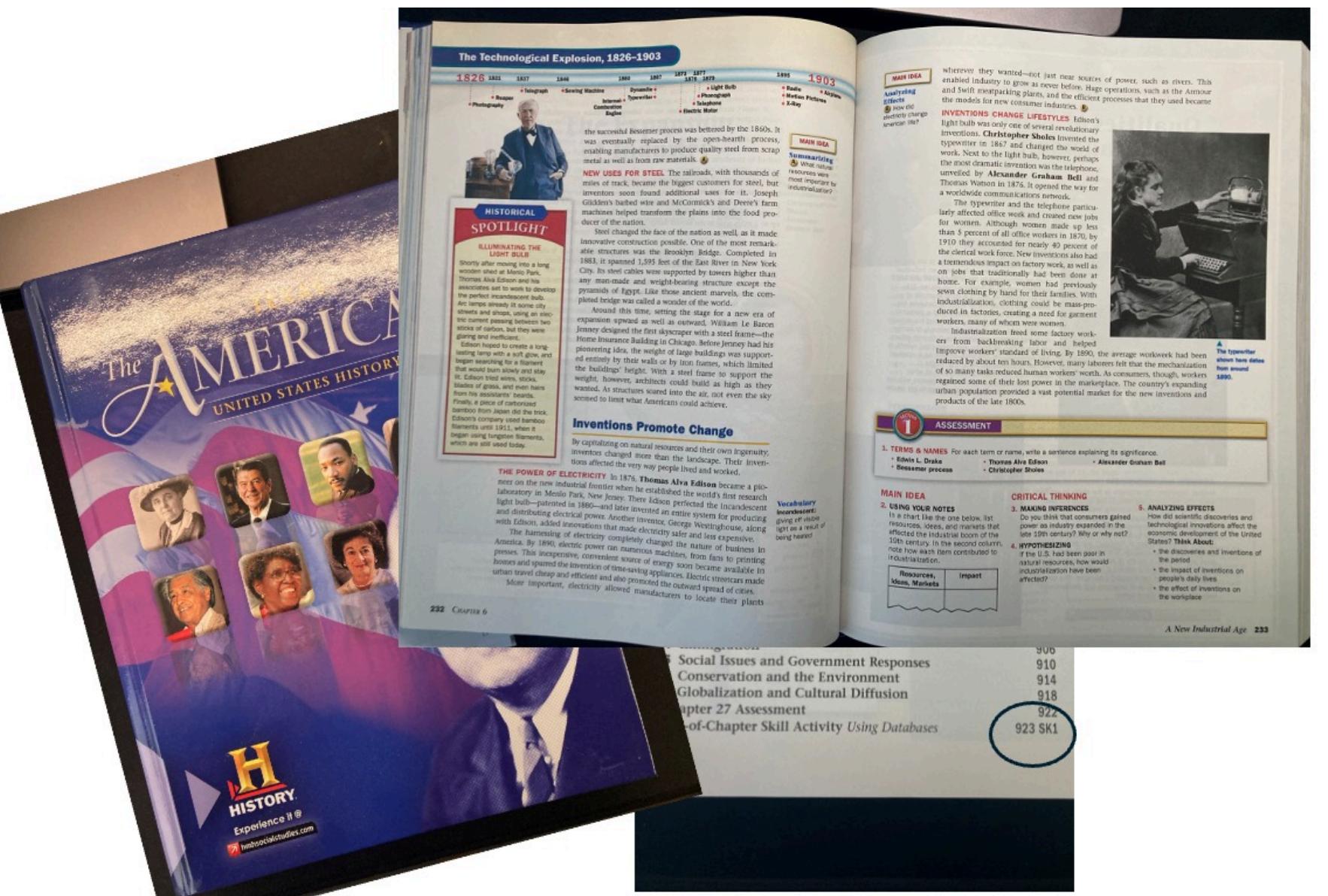
# Level of knowledge



*Prompt: Use the image you just created as a model to represent knowledge of AI*

# A quick working example

- I have 20 secondary school history textbooks.
- Each book is broken down into chapters. Each chapter has multiple sections.
- Each book has about 90 sections totaling around 1000 pages.
- I only want to read the sections that are pertinent to my study.



# Creating a Computational Filter - The Input

## PDF Topic Evaluator

Upload one or more PDFs and check if a specific topic is addressed.

🔍 Topic to evaluate (e.g. 'test-driven development')

national identity

🧠 Considerations to Analyze (one per line)

Values

National myths or symbols

Civic assumptions

Upload PDFs



Drag and drop files here

Limit 200MB per file • PDF

Browse files



Chapter 4 Section 4 Reading.pdf 4.3MB

X



Chapter 4 Section 3 Reading.pdf 7.0MB

X



Chapter 4 Section 2 Reading.pdf 2.3MB

X

Showing page 1 of 2

< >

Evaluate PDFs

# Creating a Computational Filter - The Output

## ▼ Chapter 4 Section 4 Reading.pdf

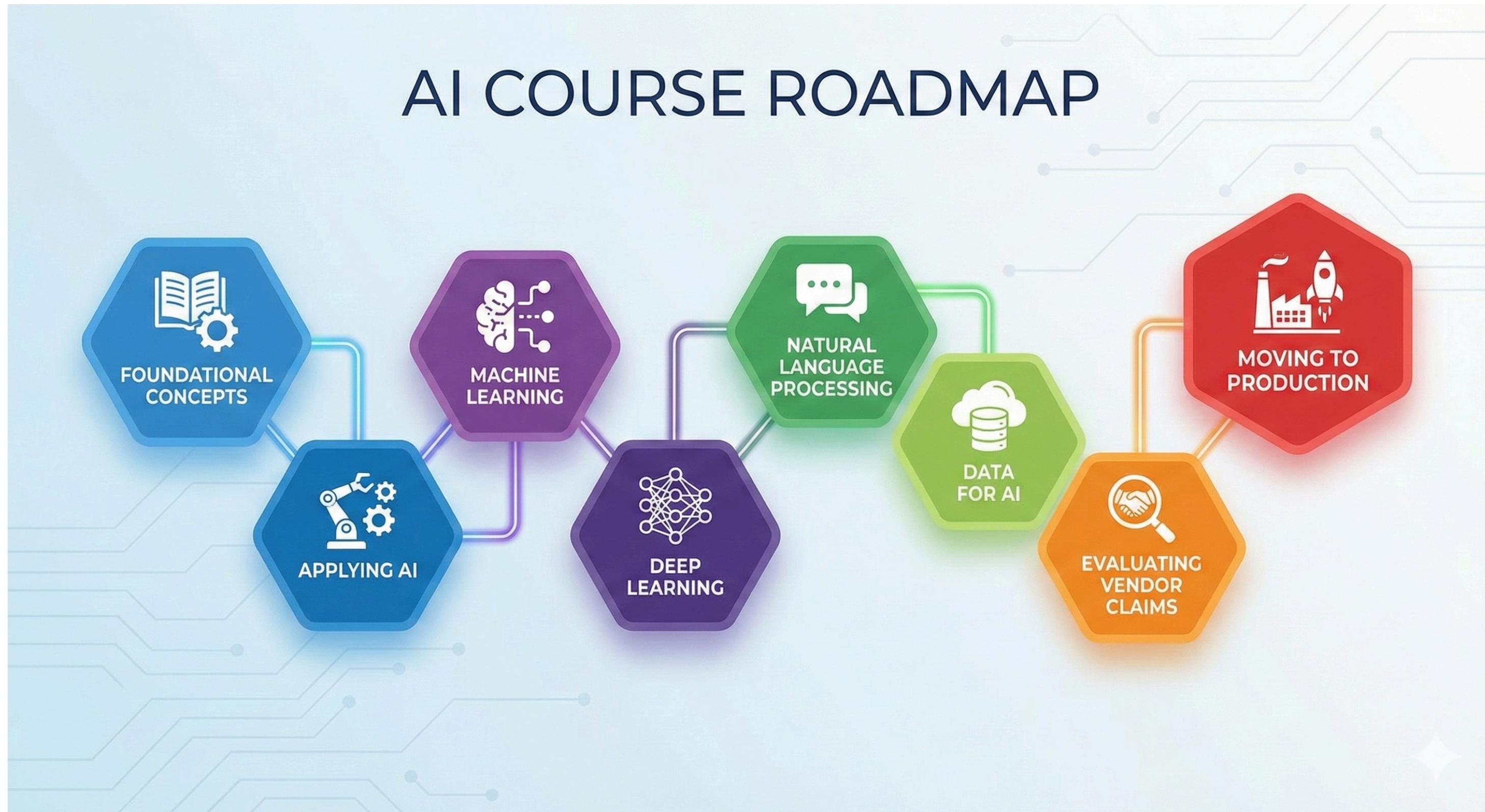
1. **Determination:** Yes, the document addresses the topic of "national identity."

2. **Analysis:**

- **Values:** The document reflects values such as liberty, courage, and patriotism. These are evident in the passages that discuss the contributions of women and soldiers during the war, emphasizing their dedication to the public good and the fight for freedom. The emphasis on liberty and resistance against tyranny are core values that contribute to the formation of a national identity.
- **National Myths or Symbols:** The document alludes to the mythic narrative of the American Revolution as a struggle for freedom and independence. The mention of the Treaty of Paris and the return of soldiers to their homes symbolizes the birth of a new nation and the establishment of its borders, which are key elements in the national identity of the United States.
- **Civic Assumptions:** The document assumes a shared commitment to the ideals of independence and self-governance. The references to the formation of a new government and the collective efforts of soldiers and civilians highlight the civic responsibility and unity necessary for nation-building.
- **Direct Quotes:**
  - "Born for liberty, disdaining to bear the irons of a tyrannic Government, we associate ourselves..."
  - "The courage of soldiers and civilians had made America's victory possible."

These elements collectively contribute to the construction and reinforcement of a national identity centered around the values and narratives of the American Revolution.

# Course roadmap

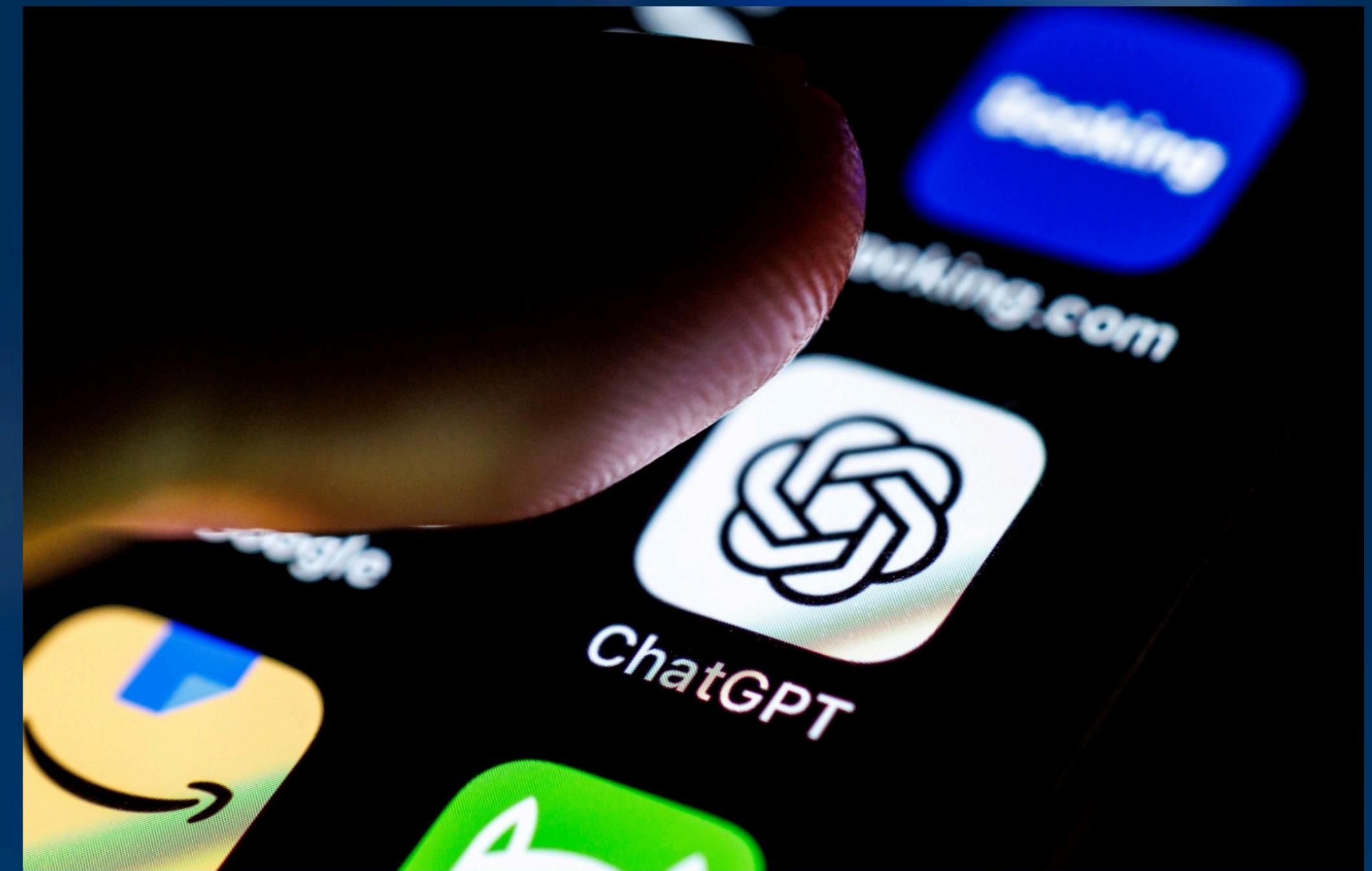


*Prompt: Create an image related to AI. The image presents the topics, or roadmap, for a course. The topics to be covered in order are: Foundational concepts, applying AI, Machine Learning, Deep Learning, Natural Language Processing, Data for AI Evaluating Vendor Claims, Moving to Production.*

RESEARCH

## Is the politicization of generative AI inevitable?

Valerie Wirtschafter and Nitya Nadgir  
October 16, 2025



<https://www.brookings.edu/articles/is-the-politicization-of-generative-ai-inevitable/>

# Foundational Concepts: What is AI?

*Artificial intelligence (AI) is a machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations, or decisions influencing real or virtual environments. AI systems use machine- and human-based inputs to perceive real and virtual environments; abstract such perceptions into models through analysis in an automated manner; and use model inference to formulate options for information or action. ([15 U.S.C. 9401\(3\)](#)).*

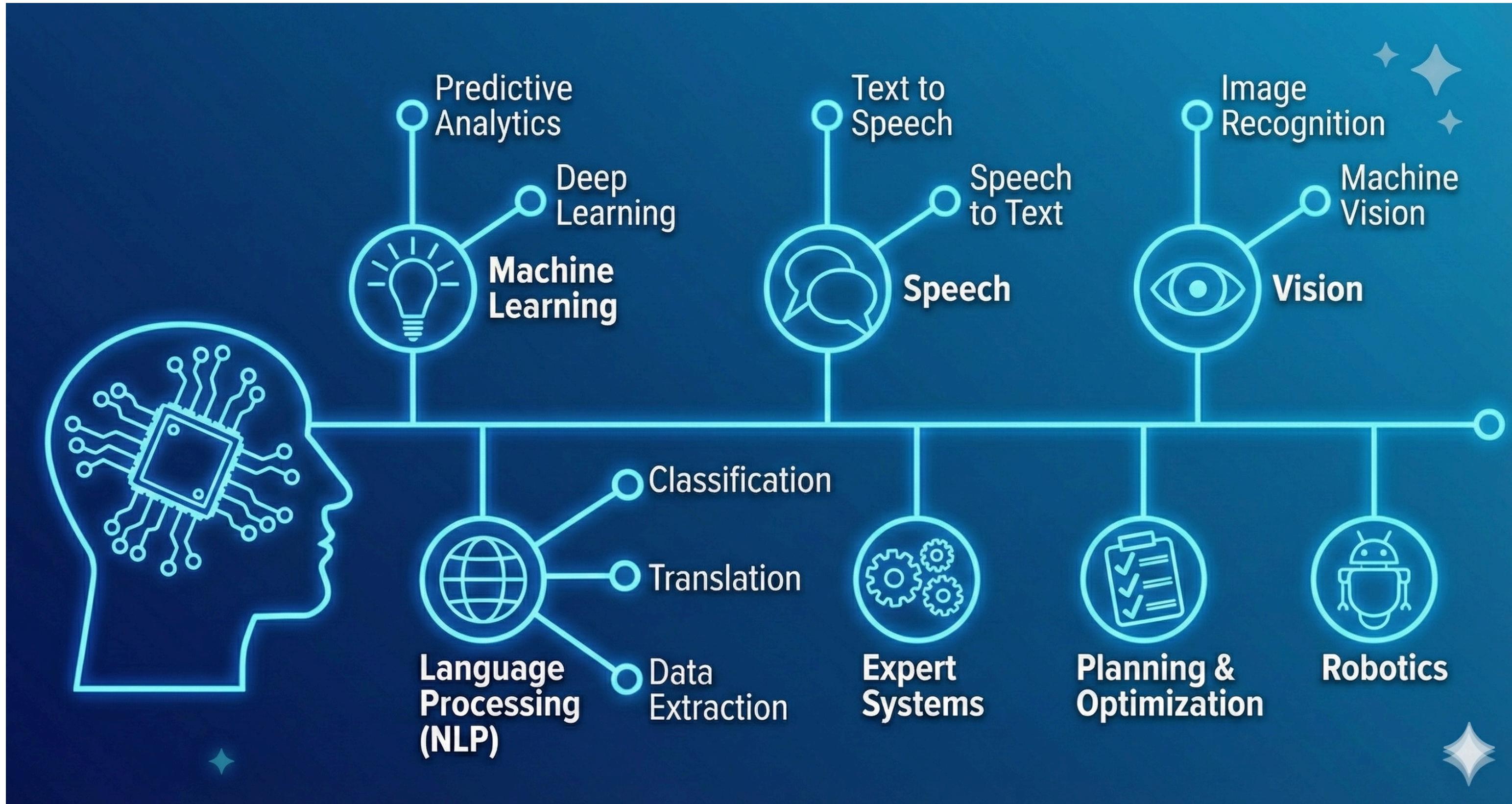
<https://www.cdc.gov/data-modernization/php/ai/cdcs-vision-for-use-of-artificial-intelligence-in-public-health.html>

Artificial Intelligence is **a set of computer techniques that let machines perform tasks that normally require human thinking**, such as recognizing patterns, making decisions, summarizing information, or predicting what might happen next.

**Instead of following a strict list of rules, AI systems learn from examples**, adjust themselves as they go, and improve when given more data.

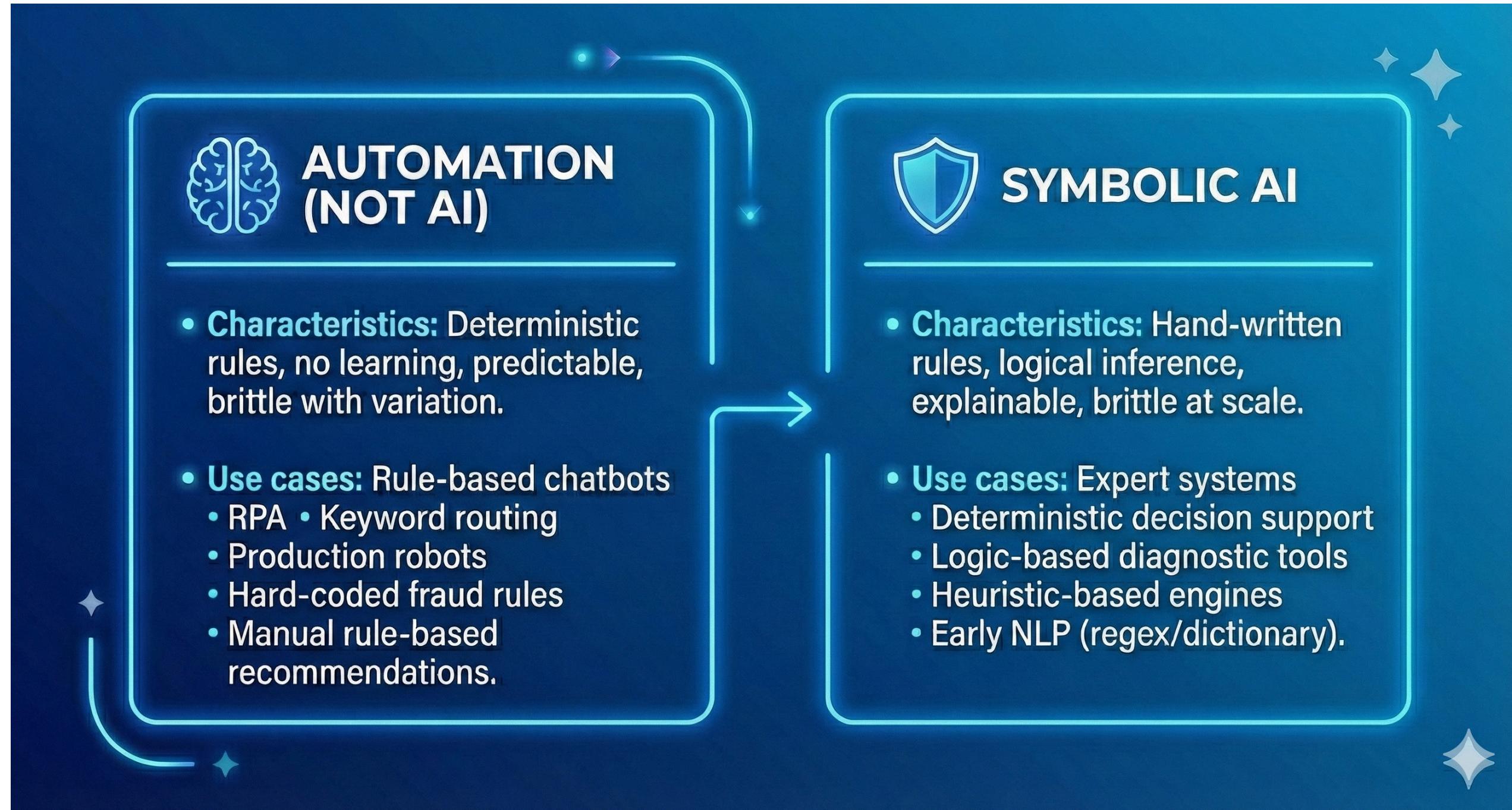
# A Very Broad View of AI

Note: This slide is a confusing mixture of AI techniques, possible use cases and use cases that might not be modern AI.



# It Probably Isn't AI

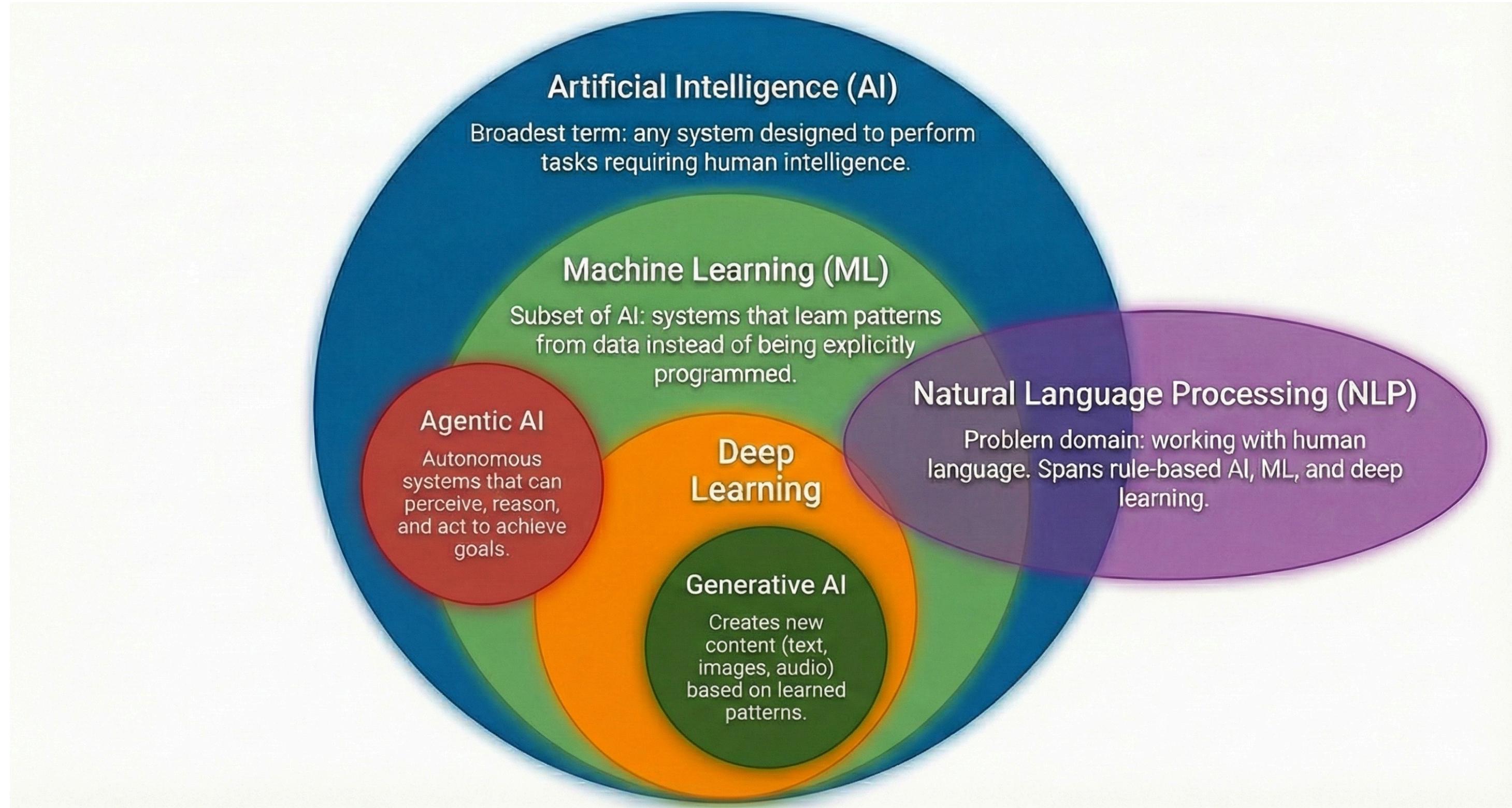
Some automation and expert systems use human-written if/then rules to imitate expert reasoning, while AI discovers its rules from data.



*Prompt: Make the first item after 'Use Cases:' a bullet point. Rules-based chatbots is a bullet point. Expert systems is a bullet point. Make sure each bullet point is on a separate line.*

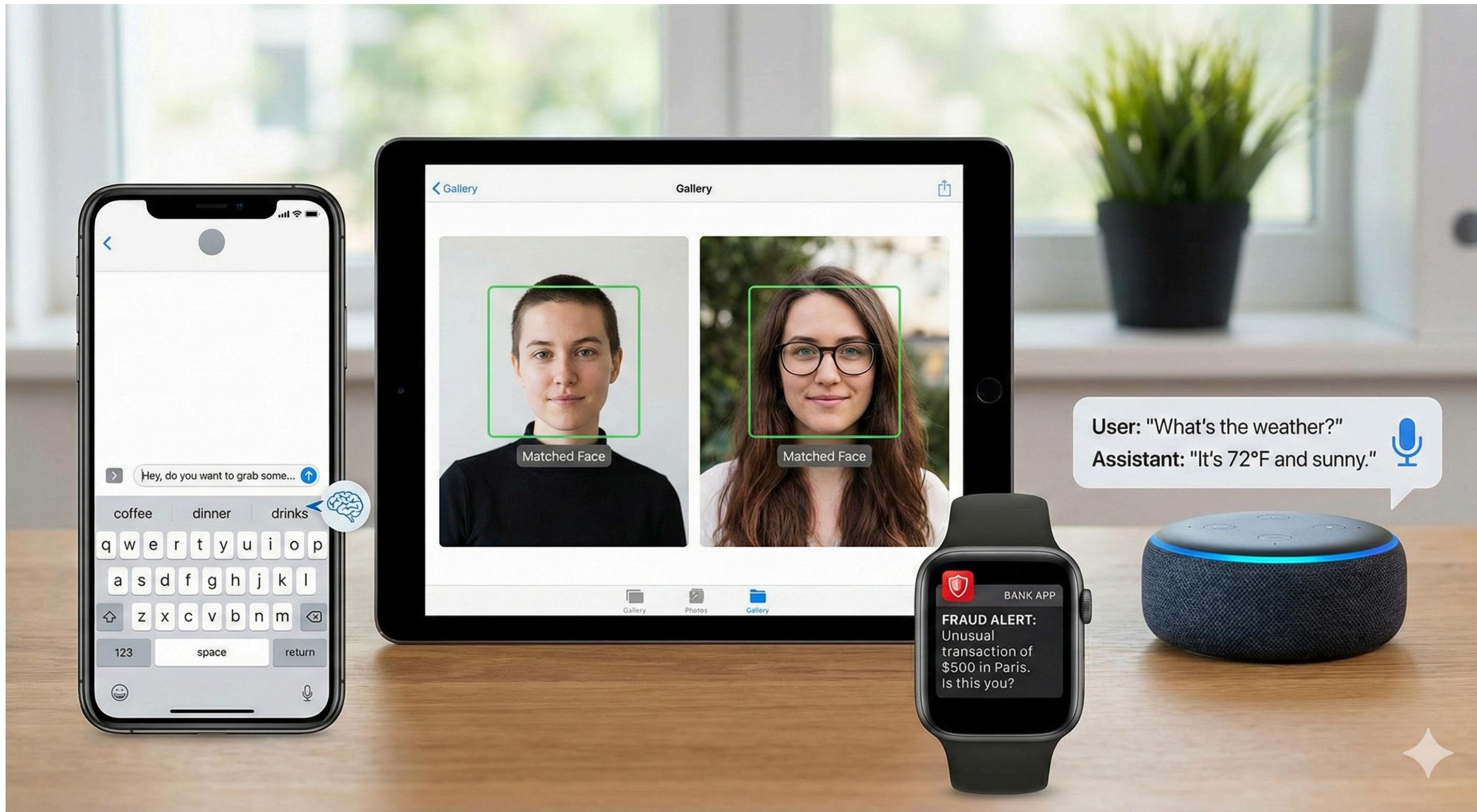
# AI Subfields

AI is a field that includes multiple techniques (ML, deep learning, etc.).



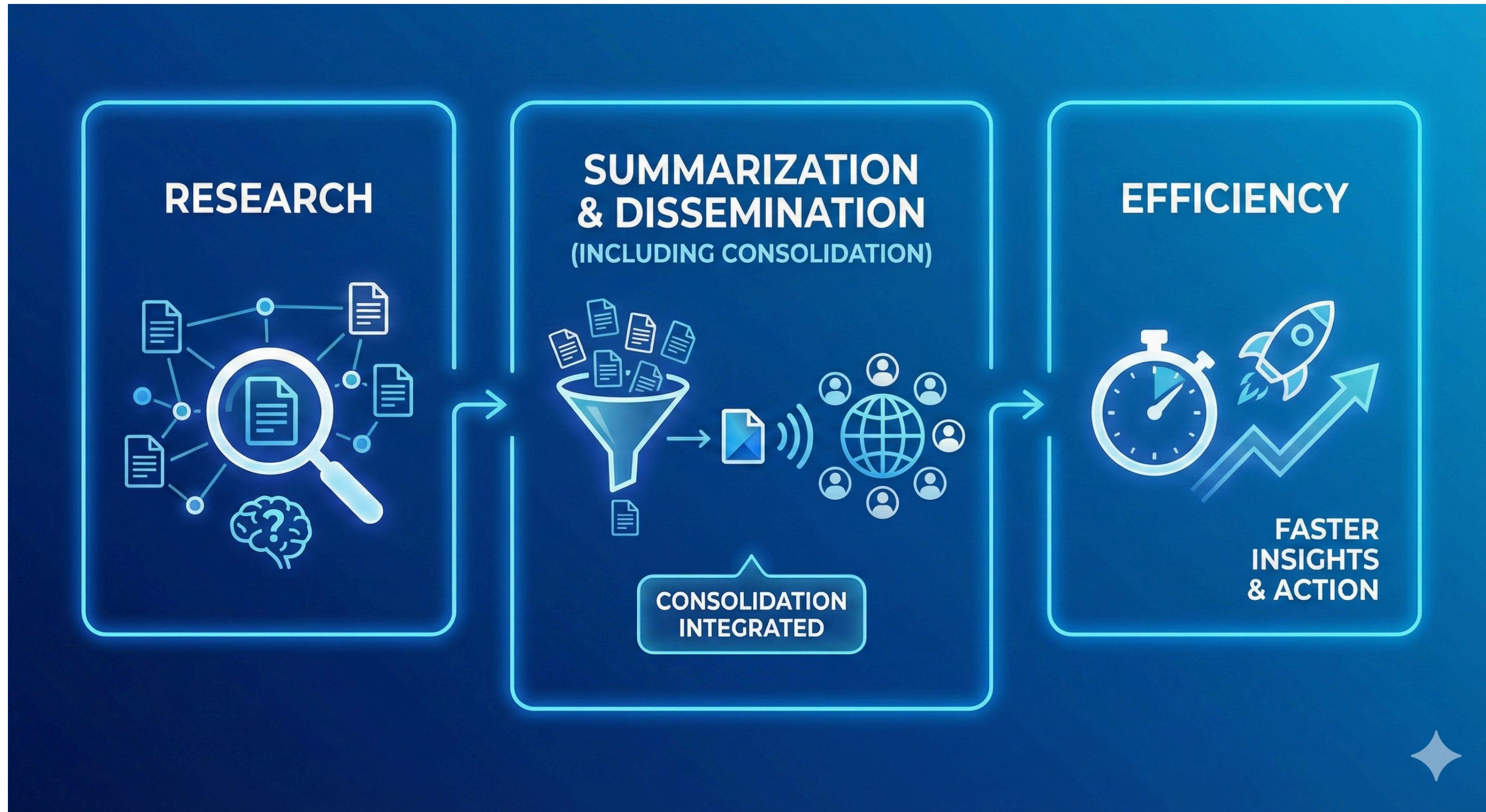
Note: NLP is a **problem domain** that leverages AI. Generative AI and Agentic AI are 'technologies'. They are systems built using the techniques.

# Everyday Examples of AI



*Prompt: Create an image that has images representing these ideas. Typing suggestions on your phone; the phone learns from previous words to predict the next one. Photo apps recognizing faces, even if the person changes their hair or angle. Fraud alerts from your bank, which notice unusual patterns in your spending. Chatbots or voice assistants, which understand your question and generate a helpful response.*

# Organizing AI by Use



*Prompt: Create an image that has images representing these aspects of AI - Research, Consolidation, Summarization and Dissemination, Efficiency.*

# Examples: AI for Research in Public Health

Projects focused on developing predictive models, identifying risk factors, and generating new insights

#	Title/Project	Institution/Organization	Year	Description
1	US Diabetes Risk Prediction Using BRFSS Data	Multiple institutions	2024-2025	Analysis of 253,680 adult respondents using machine learning models (Extra Trees Classifier, Random Forest, XGBoost) to identify influential predictors of diabetes likelihood
2	Social Media Analysis for Mental Health	Research study	2022	Research showing social media discussions can predict mental health consultations on US college campuses, enabling earlier intervention

# Examples: AI for Consolidation, Summarization & Dissemination in Public Health

Projects focused on extracting insights from large datasets, analyzing narratives, and synthesizing information for public health decision-making

#	Title/Project	Institution/Organization	Year	Description
3	BlueDot and HealthMap Surveillance Systems	BlueDot, HealthMap	Ongoing	AI-based epidemiological surveillance systems providing early warnings and real-time disease outbreak monitoring and visualization
4	Transforming Public Health Practice with Generative AI	US Health Departments	2024	Exploring how AI supports core public health functions including communications, organizational performance, and novel insights for decision-making

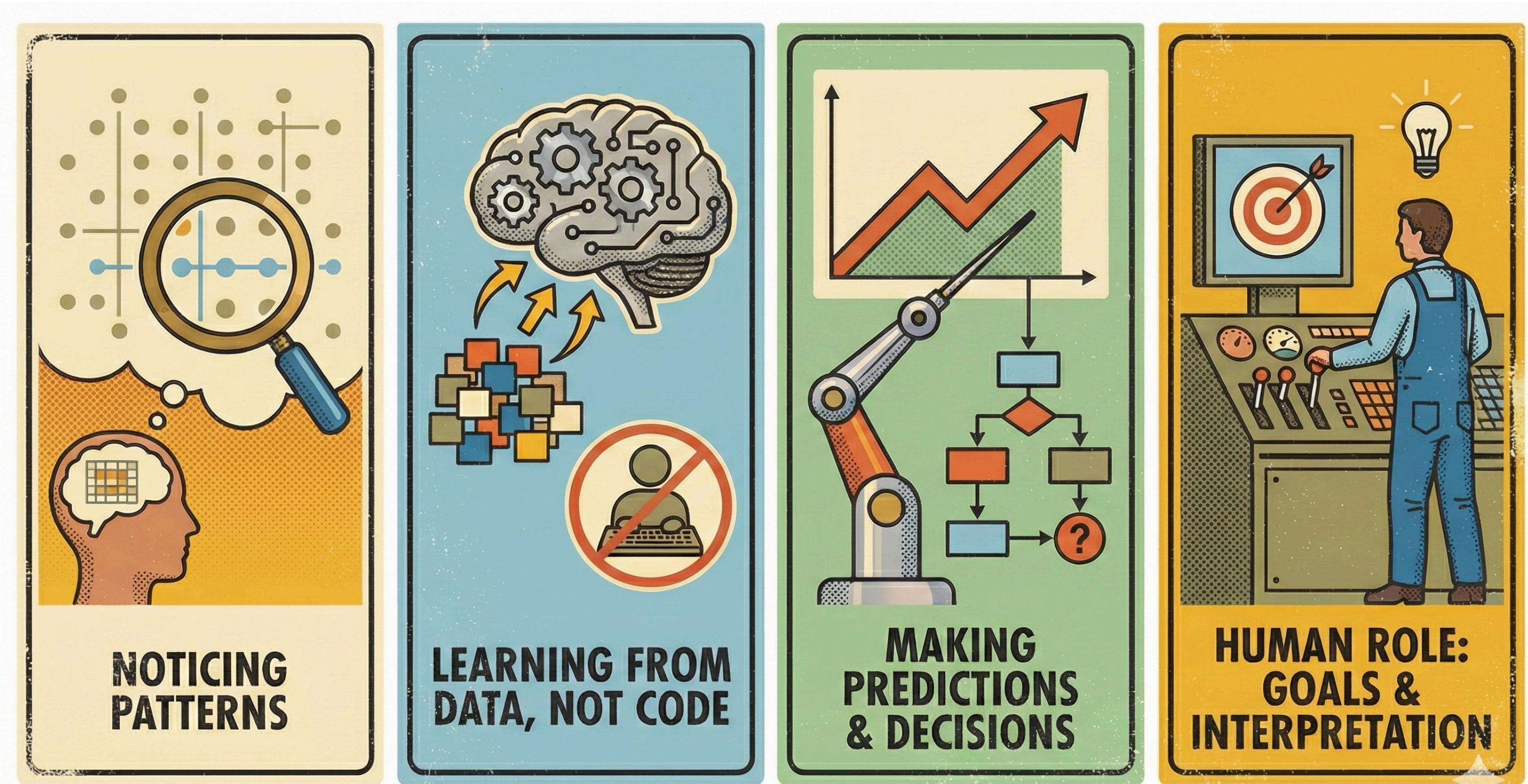
# Examples: AI for Efficiency & Automation in Public Health

Projects focused on automating routine tasks, speeding up processes, and improving operational efficiency

#	Title/Project	Institution/Organization	Year	Description
5	Conversational AI for Vaccine Communication	Research review	2023	Systematic review showing chatbot studies measuring influence on vaccine attitudes found evidence of positive effects with no "backfire effects"
6	Motivational Interviewing-Oriented AI Digital Assistant	Research study (Hong Kong & US)	2022-2024	RCT with 177 participants testing an AI-driven chatbot with motivational interviewing techniques

*Note* Some projects could fit into multiple categories. Classification is based on the primary application or objective of each initiative.

# The key idea: AI is not one thing; it is a collection of methods that let computers:



*Prompt: Create an image containing images that represent these statements. AI is not one thing; it is a collection of methods that let computers: Notice patterns in data (like humans noticing trends). Learn from those patterns (without a programmer giving step-by-step instructions). Use that learning to make predictions or decisions. Humans still define the goal, set limits, and interpret the results.*

# A helpful analogy

**Think of AI like teaching a new employee:**

- You don't give them a rulebook for every situation.
- Instead, they watch examples, practice, and get feedback.
- Over time, they get better and faster at handling similar tasks.
- AI learns the same way, but with data instead of experience.

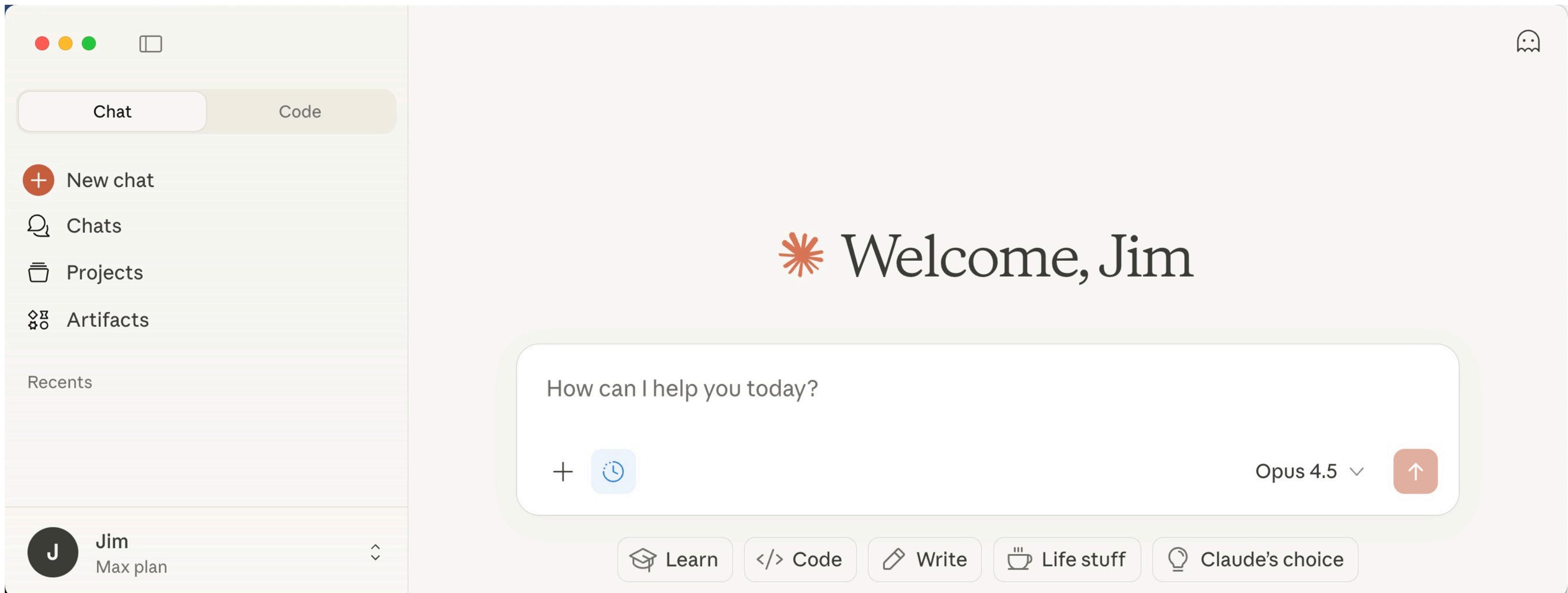
# What AI is not

- AI is not magic; it is math and pattern recognition.
- AI is not a full replacement for human judgment; it needs oversight.
- AI does not understand the world the way humans do, even when it generates fluent language.

# Major AI Subfields

Concept	Description
Artificial Intelligence (AI)	<b>Broadest term:</b> any system designed to perform tasks requiring human intelligence; includes both rule-based systems and learning-based systems; the umbrella category.
Machine Learning (ML)	<b>Subset of AI:</b> systems that learn patterns from data instead of being explicitly programmed. Requires structured data and engineered features; often interpretable. Examples include decision trees, random forests, logistic regression, and support vector machines.
Deep Learning	<b>Subset of ML:</b> neural networks with many layers that learn representations automatically. Works with unstructured data, needs large datasets and computation, and is often a “black box.”
Generative AI	<b>Subset of Deep Learning:</b> systems that create new content (text, images, code, audio, etc.) by learning patterns from data. Typically built using deep learning, especially transformer models.
Agentic AI	<b>System behavior:</b> AI that has autonomy to take actions toward a goal. Uses planning, decision-making, tool use, memory, and feedback loops.
Natural Language Processing (NLP)	<b>Problem domain:</b> working with human language. Spans rule-based AI, ML, and deep learning.

# Claude HHS walkthrough



# Models

How can I help you today?



Learn



Code



Write



Opus 4.5 ▾



Opus 4.5

Most capable for complex work



Sonnet 4.5

Best for everyday tasks

Haiku 4.5

Fastest for quick answers

More models >

# Options

How can I help you today?

+ ⏰

Opus 4.5 ▾ ↑

- Add files or photos
- Add to project >
- Research
- Web search ✓
- Use style >
- Connectors >

e Write Life stuff Claude's choice

# Chat responses

read the data from this url and show me the first ten rows in a table.

[https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic\\_data1K.csv](https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic_data1K.csv)

^ 1 step

🌐 Fetched https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic\_data1K.csv [githubusercontent.com](#)

● Parsed CSV data and structured tabular display.

Here are the first 10 rows of the diabetic data:

encounter_id	patient_nbr	race	gender	age	time_in_hospital	num_lab_procedures
2278392	8222157	Caucasian	Female	[0-10)	1	41
149190	55629189	Caucasian	Female	[10-20)	3	59
64410	86047875	AfricanAmerican	Female	[20-30)	2	11
500364	82442376	Caucasian	Male	[30-40)	2	44
16680	42519267	Caucasian	Male	[40-50)	1	51



# Projects

The screenshot shows a user interface for managing projects. On the left is a sidebar with the following items:

- Chat (selected)
- Code
- + New chat
- Chats
- Projects (selected)
- Artifacts
- Recents

In the bottom right corner of the sidebar, there is a dark circular badge with the letter "J" and the text "Max plan".

The main area is titled "Projects" and features a search bar at the top with the placeholder "Search projects...". Below the search bar, there is a "Sort by" dropdown set to "Activity".

The projects are listed in a grid:

Project Name	Last Updated
Motivation	Updated 2 days ago
Do math	Updated 2 days ago
Claude to R	Updated 4 days ago
stage1 topic model	Updated 6 days ago
data generation	Updated 6 days ago
C2R - create data	Updated 15 days ago

# Project Details

asciidoc

...



Reply...



Opus 4.5



CSS slide layout padding explanation

Last message 23 hours ago

CSS indentation for AsciiDoc markdown lists

Last message 23 hours ago

CSS text indent matching image alignment

Last message 1 day ago

CSS indentation for instructions and bullets

Last message 1 day ago

CSS padding strategy for markdown text

Last message 2 days ago

Memory

Only you



Purpose & context Jim is creating educational presentations using AsciiDoc converted to Reveal.js slides. The presentatio...

Last updated 12 hours ago

Instructions

I am converting asciidoc files to slide pdfs using reveal and Docker.

Files



1% of project capacity used

convert-to-slides.sh

18 lines

SH

command.txt

1 line

TXT

cdi-theme.yml

97 lines

YML

all-slides.adoc

23 lines

ADOC

# Project chats

asciidoc / CSS indentation for AsciiDoc markdown lists ▾

all-sliders.css

CSS

Attached is the most current version on my .css. When I use this markdown, I  
the lines below the image to be indented. This code does the indenting the  
way I want it.

```
[.indent.space-above-3]  
  
* *What is AI?* – The landscape: AI, Machine Learning, Deep  
Learning, and NLP  
* *Machine Learning Overview* – How machines learn from data:  
Show more
```

Examined CSS compatibility and formatting inconsistencies between working and broken examples. ▾

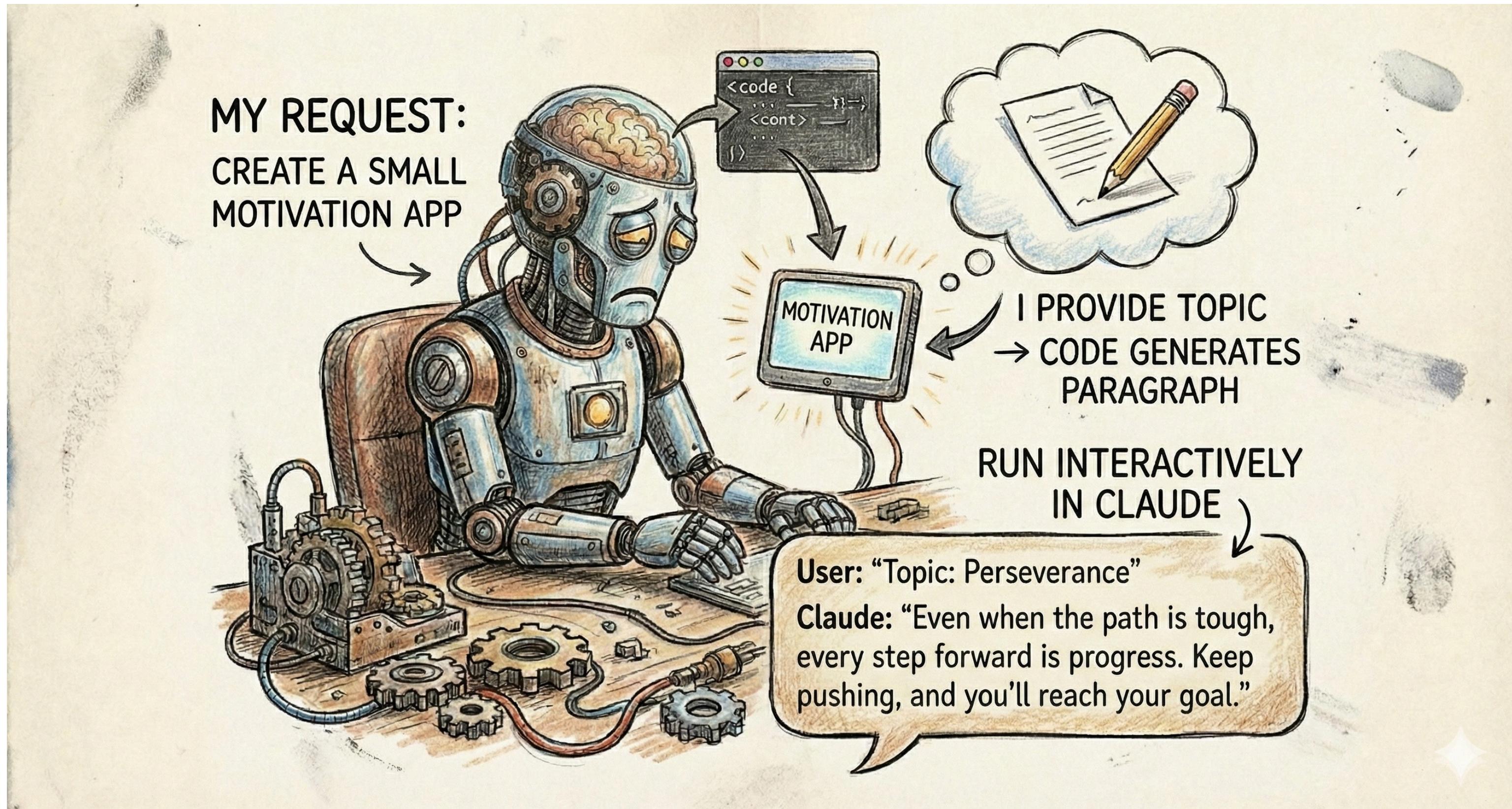
Let me look at your CSS file to see what styling options are available and understand what  
might be happening with the formatting.

css Reviewing the CSS file to understand available indent styles and formatting options Show working file

Diagnosed block structure issue; prescribed open block wrapper solution. 27s ▾

Looking at your CSS, I can see the `.indent` c' ↓ is defined and works correctly. The issue  
is with the **AsciiDoc structure**, not the CSS.

# Motivation app



# Prompt 1 - Getting started

## Prompt

*I need some motivation.*

Devised warm, conversational encouragement for motivation request. ▼

I've got you.

Here's the thing: the fact that you're asking for motivation means something in you is still fighting, still reaching. That spark matters more than you might realize right now.

Whatever you're facing—whether it's a project that feels too big, a goal that seems far off, or just getting through the day—remember that progress doesn't have to be dramatic to be real. Small steps still move you forward. Showing up, even when it's hard, is an act of courage.

You don't have to feel motivated to take action. Sometimes action is what creates the motivation. Start with one thing. Just one. The momentum will follow.

What's weighing on you? I'm happy to get more specific if you want to talk through it.

## Prompt 2

### Prompt

*How about asking me for a topic?*

Fair point—I jumped ahead. What do you need motivation for?

waking up.

That's one of the hardest ones—because it's not just physical, it's the question underneath: *what am I getting up for?*

A few thoughts:

**If it's about the morning battle itself:** Your brain lies to you at 6 a.m. It says the bed is the only good thing and the day has nothing for you. It's wrong. The day hasn't even had a chance yet. Sometimes the win is just feet on the floor. That's it. Everything else comes after.

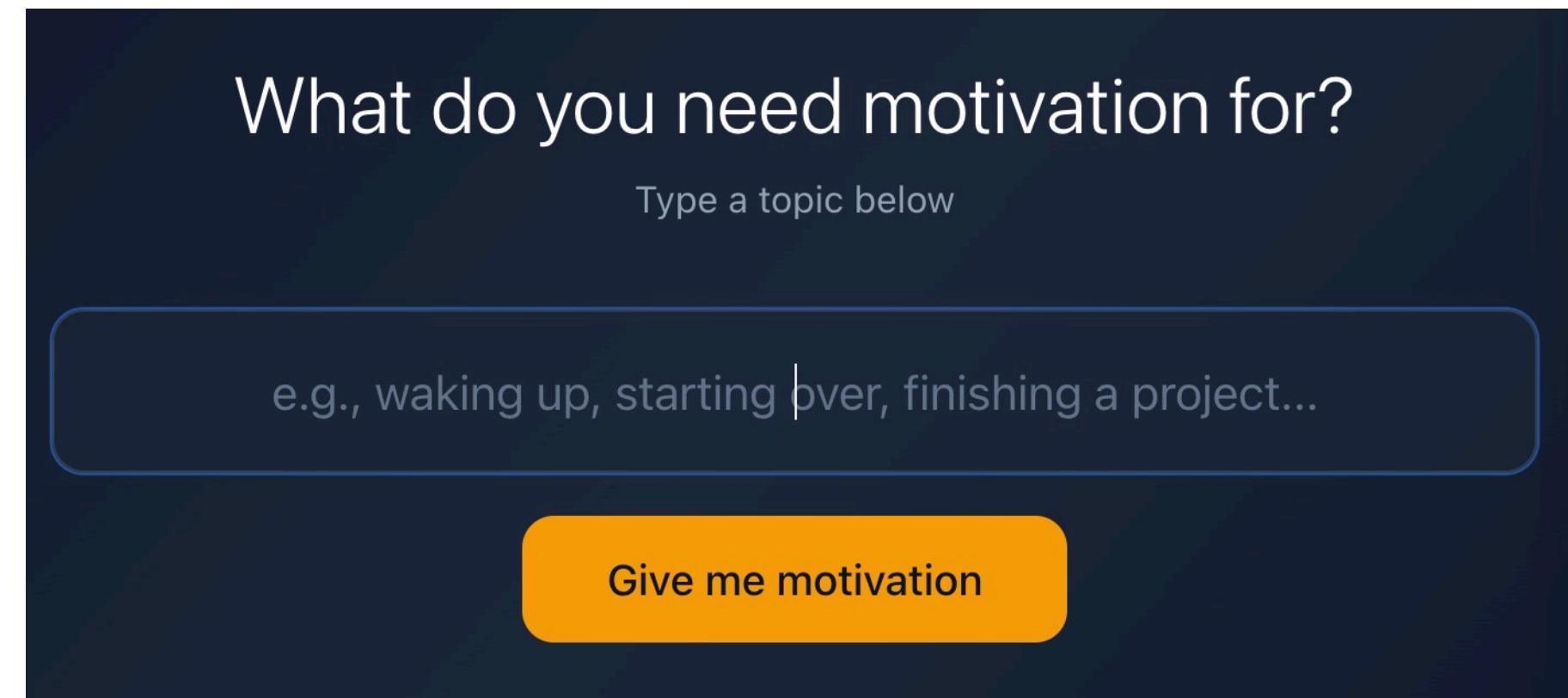
**If it's something deeper:** Not every day has to have a grand purpose waiting. Sometimes you wake up for small things—coffee, sunlight, curiosity about what might happen, a song you like, someone who'd notice if you weren't there. Those count. They're not small at all, actually.

Which side of it are you feeling?

## Prompt 3

### Prompt

*Create an interface that will run within this chat. It should prompt me for a topic and respond with a 5-6 line motivational statement*



## Prompt 4

### Prompt

*You decide.*

- Modify your motivation app. Give Claude instructions to shorten or lengthen the response, change the tone of the response, change colors, etc.

What do you need a little snarky motivation for?

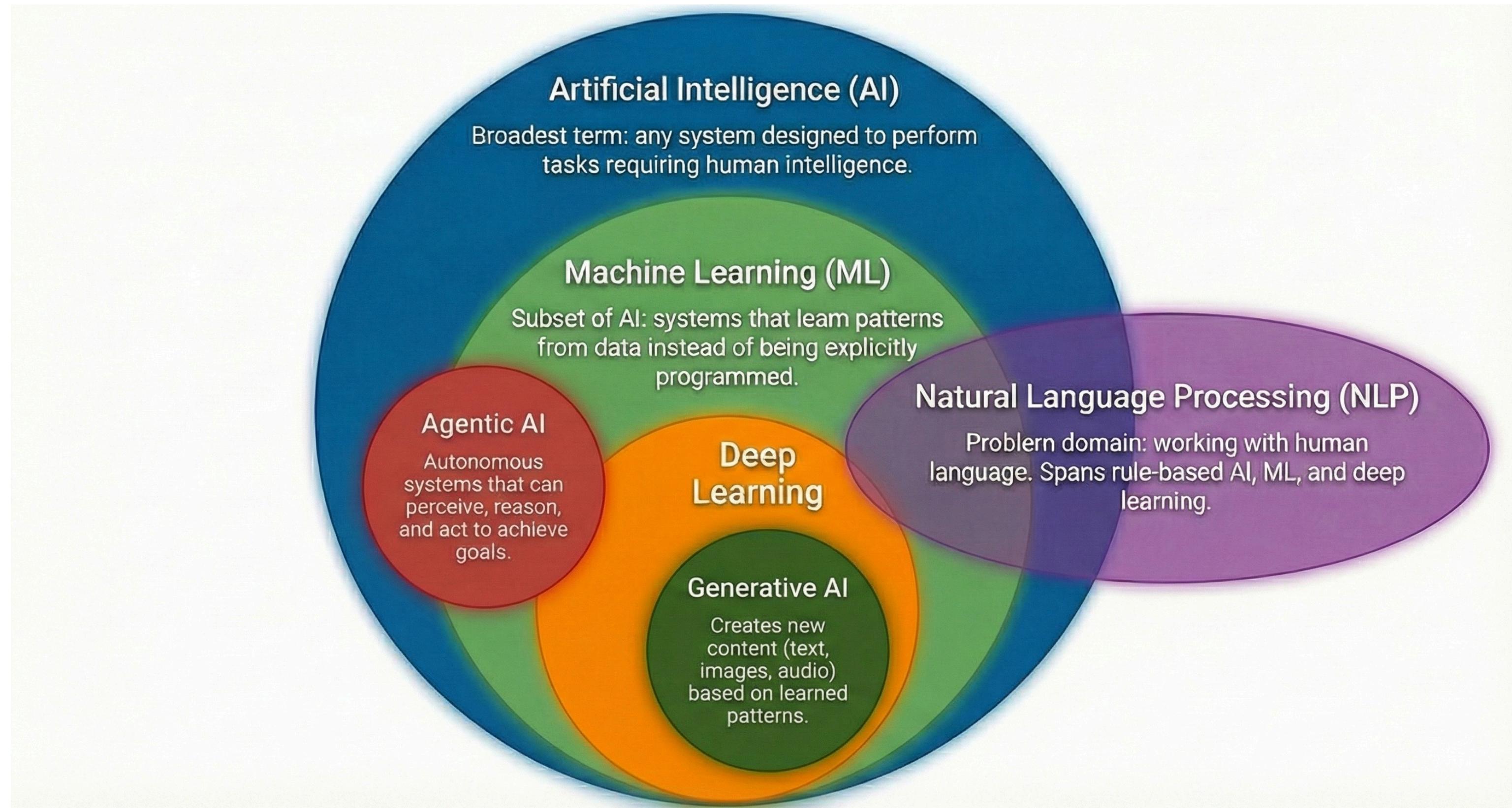
Type a topic below

e.g., waking up, starting over, finishing a project...

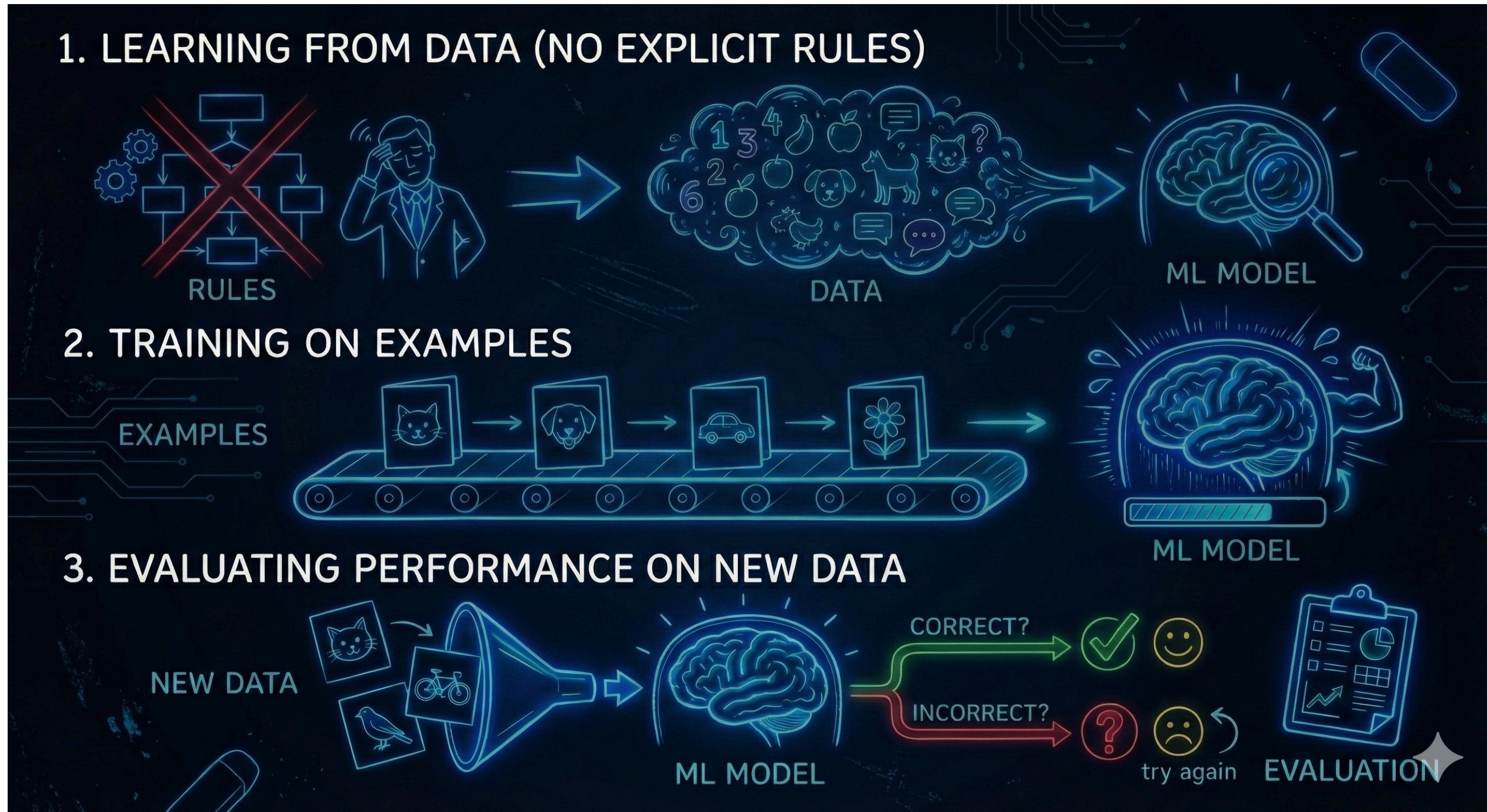
Give me motivation

# Foundational Concepts: Machine Learning Overview

**Machine learning (ML)** is the engine that powers modern AI. Deep learning, NLP, and other advanced techniques all build on ML's core ideas. Understanding ML is the foundation for understanding all of modern AI.



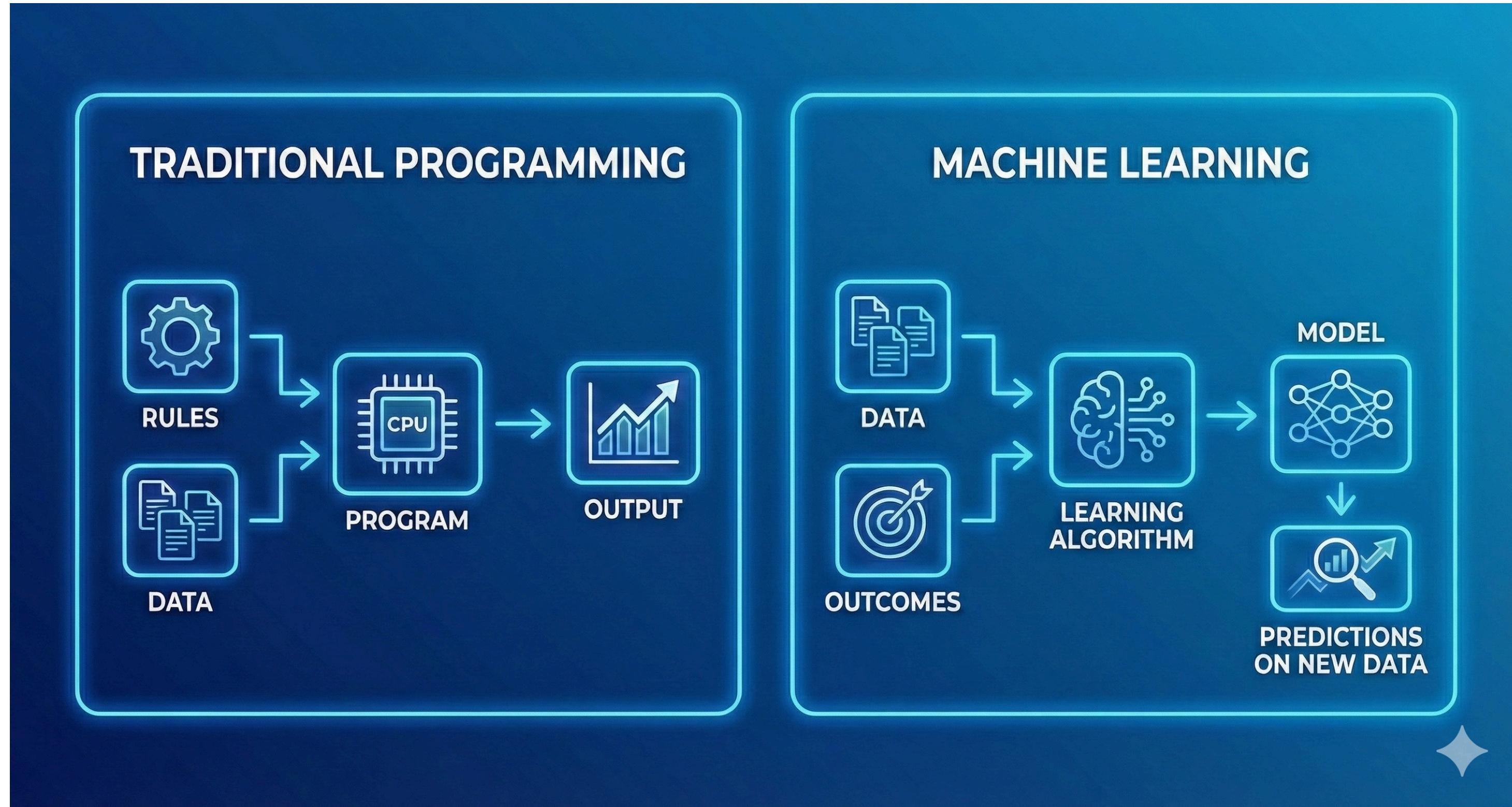
# Machine Learning Core Concepts



*Prompt: Create an image that portrays the three core concepts of Machine learning. Don't be word heavy. Use images. Create the image in a 'hand drawing' style. Use this as a basis for the image. ML provides the three core concepts that everything else builds on: Learning from data (not from explicit rules), Training on examples, Evaluating performance on new data. Follow-up: Can you make the images a little opaque so that the 1., 2. 3. text stands out more?*

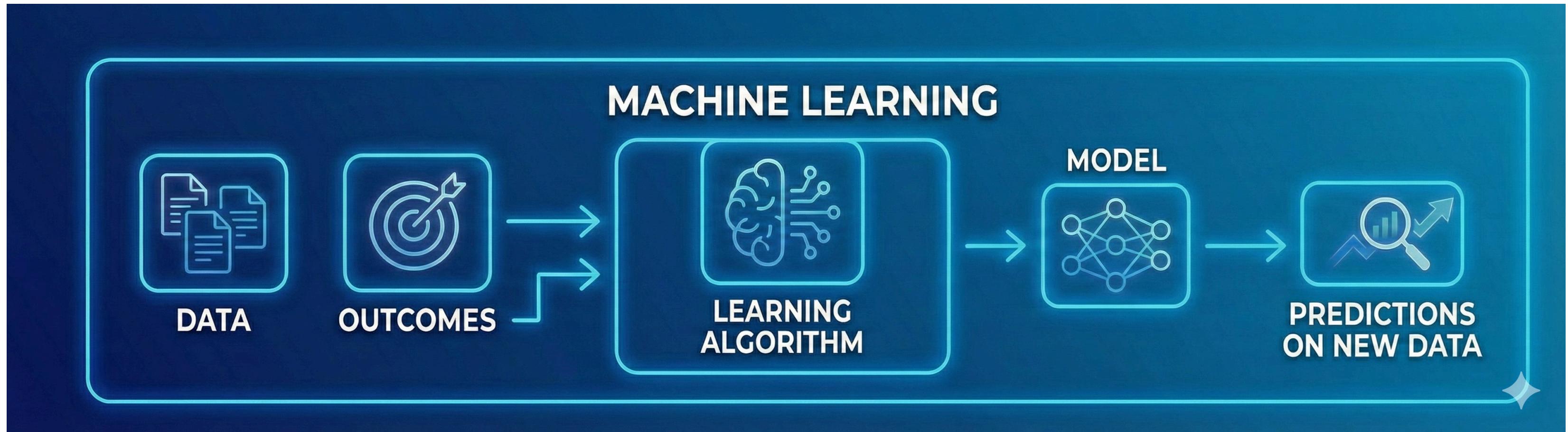
# The Core Idea: Learning from Data

Machine learning is a way for computers to **learn patterns from data**, rather than being explicitly programmed with rules.



*Prompt: I am discussing machine learning. create a Two-panel diagram: Left: Traditional programming (Rules + Data → Output) Right: Machine learning (Data + Outcomes → Learning Algorithm → Model → Predictions on new data). Use the attached file as a style guide. Follow-up: Leave the title of each box but remove the subtitle that describes the flow. Leave all else as is.*

# The Learning Process



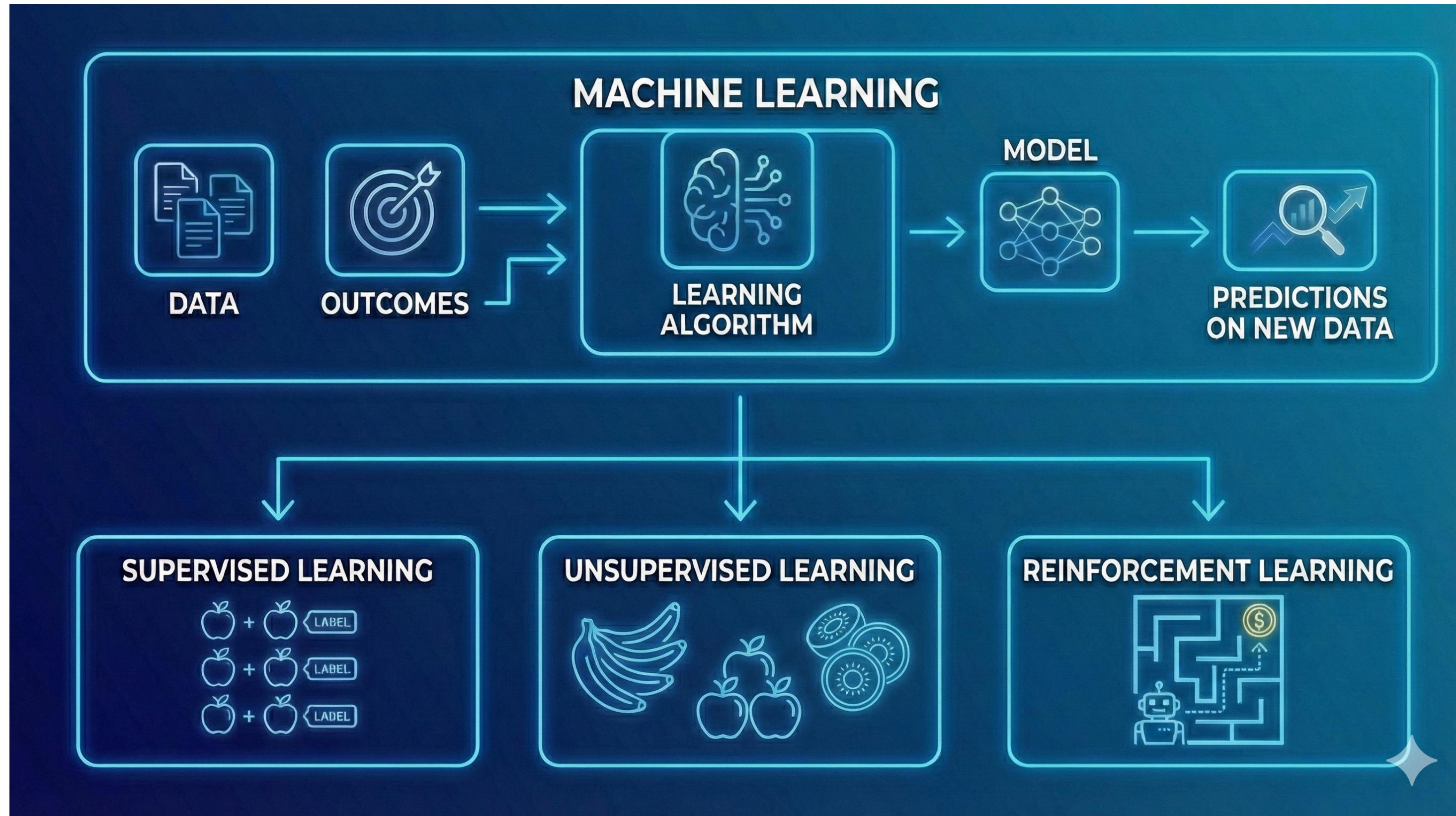
## How learning works (intuition, not math):

- The algorithm makes a guess, checks how wrong it is, and adjusts.
- It repeats this process—thousands or millions of times—until it minimizes errors.
- Think of it like learning to throw darts: we throw, see where it lands, adjust our aim, throw again.

## The train/test split:

- We can't test learning on the same examples we learned from (that's just memorization).
- So we split the data: learn from some (training data), test on the rest (test data).
- Performance on test data tells us if the model actually learned generalizable patterns.

# ML Learning Paradigms



# Supervised Learning

In supervised learning, the algorithm learns from **labeled** examples - data where we know the right answer.

encounter_id	patient_nbr	race	gender	age	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	A1Cresult	insulin	change	diabetesMed	readmitted
2278392	8222157	Caucasian	Female	[0-10)	1	41	0	1	0	0	0	None	No	No	No	NO
149190	55629189	Caucasian	Female	[10-20)	3	59	0	18	0	0	0	None	Up	Ch	Yes	>30
64410	86047875	AfricanAmerican	Female	[20-30)	2	11	5	13	2	0	1	None	No	No	Yes	NO
500364	82442376	Caucasian	Male	[30-40)	2	44	1	16	0	0	0	None	Up	Ch	Yes	NO
16680	42519267	Caucasian	Male	[40-50)	1	51	0	8	0	0	0	None	Steady	Ch	Yes	NO
35754	82637451	Caucasian	Male	[50-60)	3	31	6	16	0	0	0	None	Steady	No	Yes	>30
55842	84259809	Caucasian	Male	[60-70)	4	70	1	21	0	0	0	None	Steady	Ch	Yes	NO

- "Supervised" because the learning is being supervised with correct answers.
- The data includes both inputs (features) and the outcome (label) we want to predict.
- The algorithm learns the relationship between inputs and outcomes.
- Once trained, the model can predict outcomes for new cases when the answer is unknown.

## Classification examples:

- Will this patient be readmitted? (Yes / No)
- Is this email spam? (Spam / Not spam)
- What type of request is this? (Complaint / Question / Compliment)

## Regression examples:

- How many people will enroll next month? (a number)
- How long will this patient stay in the hospital? (days)
- What will the temperature be tomorrow? (degrees)

# Unsupervised Learning

In unsupervised learning, there are no labels. The algorithm finds structure or patterns in the data on its own.

- "Unsupervised" because there's no right answer to learn from.
- The algorithm explores the data to find natural groupings, patterns, or simplifications.
- Useful when we don't know what we're looking for, or when we want to understand the structure of your data.

Application	What It Does	Example
Clustering	Groups similar cases together	Which patients have similar profiles? What types of complaints do we receive?
Dimensionality Reduction	Simplifies data with many variables	We have 200 variables—which combinations capture most of the information?

# Clustering & Dimensionality Reduction

## Clustering:

- Finds natural groupings in data without being told what the groups are.
- You might discover that your population falls into 4–5 distinct segments you didn't know existed.
- Useful for: customer segmentation, identifying outbreak clusters, grouping similar facilities.

## Dimensionality reduction:

- When you have too many variables to work with, this technique reduces them to a smaller set.
- It finds combinations of variables that capture most of the important variation.
- Think of it as data simplification: 50 variables become 5 composite variables that retain most of the signal.
- Useful for: simplifying complex data before analysis, identifying which factors matter most.

# Reinforcement Learning

Reinforcement learning is a third paradigm where an agent learns by taking actions and receiving feedback (rewards or penalties).

- Different from supervised (no labeled examples) and unsupervised (not just finding structure).
- An agent interacts with an environment, takes actions, and learns from the results.
- Good outcomes = rewards; bad outcomes = penalties.
- Over time, the agent learns a strategy (policy) that maximizes rewards.

Classic examples:

- Game-playing AI (learns to win by playing millions of games)
- Robotics (learns to walk by trial and error)
- Recommendation systems (learns what to show you based on your clicks)

Why it's not our focus:

- Less common in public health and government contexts (so far).
- Requires an environment where the agent can take actions and observe results.
- Most agency AI applications are supervised or unsupervised.

# Why This Matters for Everything Else

The learning paradigms are core concepts. **Learning from data**, supervised vs. unsupervised, training and testing—appear in every AI application you'll encounter.

- Deep learning uses the same paradigms:
  - Supervised deep learning: image classification, speech recognition
  - Unsupervised deep learning: finding patterns in text or images
  - Reinforcement learning: game-playing AI, robotics
- NLP uses the same paradigms:
  - Supervised: classify emails as spam/not spam, sentiment analysis
  - Unsupervised: topic modeling, clustering documents by similarity
- This vocabulary transfers everywhere:
  - Training data, test data
  - Labels, features
  - Classification, regression, clustering
  - Overfitting (learning too much from training data)

# The Three Paradigms at a Glance

A summary reference for the three learning paradigms.

Paradigm	What It Needs	What It Does	Common Tasks	Public Health Example
Supervised	Labeled data (inputs + outcomes)	Learns to predict outcomes	Classification, Regression	Predict readmission risk
Unsupervised	Unlabeled data (inputs only)	Finds structure or patterns	Clustering, Dimensionality reduction	Identify patient segments
Reinforcement	Environment + feedback	Learns strategy through trial and error	Game playing, Robotics	(Less common in public health)

# Machine Learning Summary

Component	Key Takeaway
ML is the foundation	Everything else in AI builds on ML's core concepts
Learning from data	ML finds patterns through iteration, not explicit rules
Supervised learning	Learns from labeled examples; classification and regression
Unsupervised learning	Finds structure without labels; clustering and dimensionality reduction
Reinforcement learning	Learns from feedback; less common in public health
Why this matters	These concepts recur throughout deep learning and NLP
Three paradigms at a glance	Summary reference for quick recall

# Foundational Concepts Exercises

- Topic 1: Was AI used to build the model that performs the task?
- Topic 2: What ML learning paradigm was used?

## **Topic 1: Was AI used to build the model that performs the task?**

---

**A system that analyzes chest X-rays to flag potential tuberculosis cases for radiologist review.**

---

**An eligibility determination system that checks if an applicant's income is below 138% of the federal poverty level and they reside in the state.**

---

**A model that predicts which restaurant establishments are most likely to have critical food safety violations, so inspectors can prioritize visits.**

---

**A chatbot that answers Medicaid beneficiaries' questions about their coverage, eligibility status, and how to find a provider.**

---

## Topic 2: What ML learning paradigm was used?

**Predict which pregnant women enrolled in Medicaid are at high risk for preterm delivery so care coordinators can provide additional support.**

---

**Analyze emergency department visit data to discover natural groupings of patients with similar symptoms, demographics, and utilization patterns—without predefined categories.**

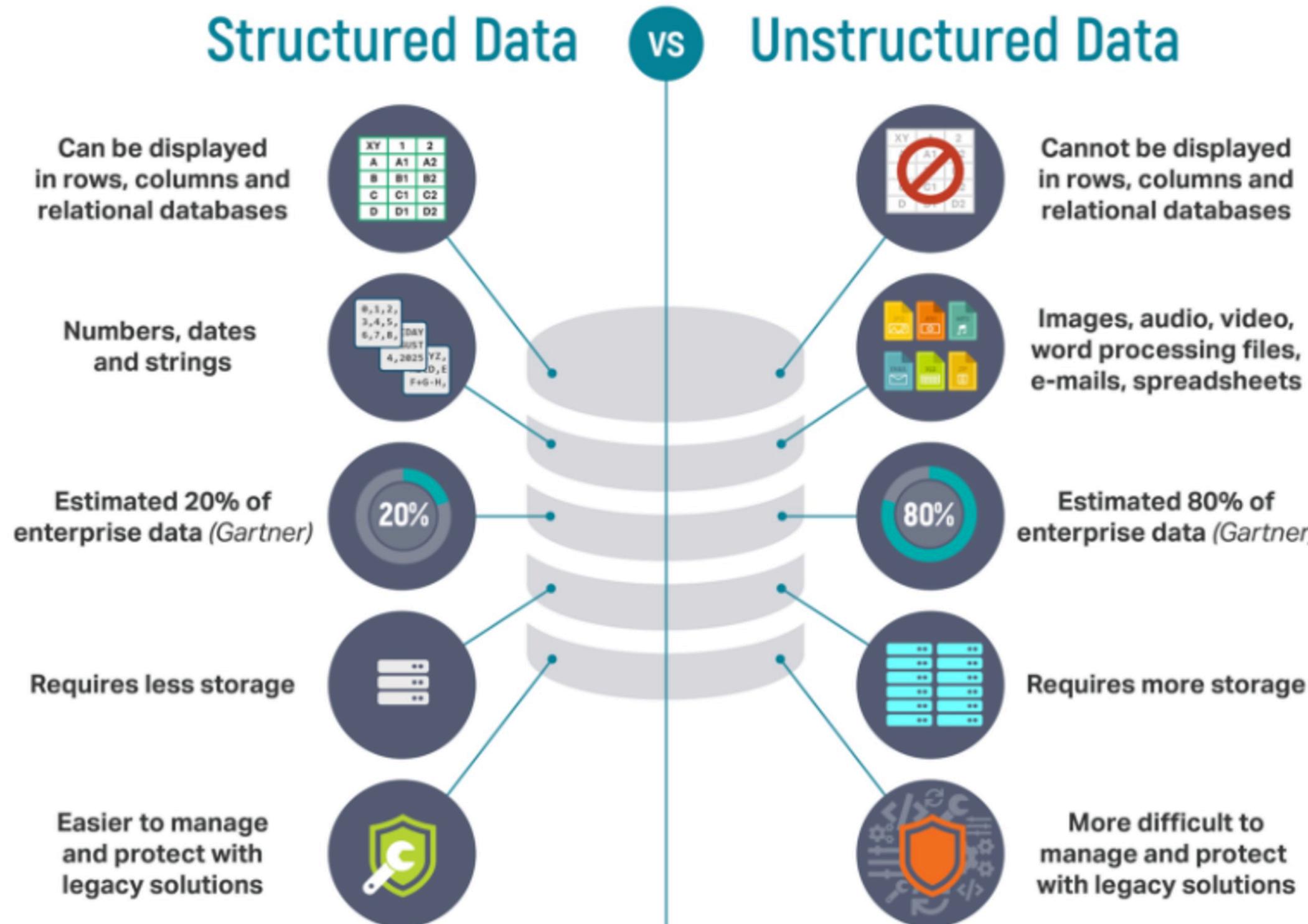
---

**Determine which WIC participants are likely to miss their recertification appointment based on past attendance, demographics, and enrollment history.**

---

**A simulation system that learns the optimal strategy for allocating limited vaccine doses across counties by trying different distribution approaches and observing outcomes over many simulated scenarios.**

# Foundational Concepts: Data Types



# Structured vs. unstructured data

Data comes in different forms, and the form affects how easily ML can work with it.

- **Structured data:** Organized in rows and columns, where each column is a defined field with a consistent format.
  - Examples: Spreadsheets, databases, claims data, enrollment records
  - Each row is a case/record; each column is a variable
  - ML algorithms can work with this directly
- **Unstructured data:** No predefined format; the information is embedded in content that requires processing to extract.
  - Examples: Free-text notes, emails, PDFs, images, audio recordings
  - The meaning isn't in neat columns—it's in the words, pixels, or sounds

# Implication for AI

## Data Type

## What It Means for ML

Structured

Most ML algorithms work directly; relatively straightforward

Unstructured

Requires extra processing before ML can use it

Extra processing for unstructured data:

- Text → Natural Language Processing (NLP) to extract meaning, categorize, or convert to structured features
- Images → Computer vision to identify objects, patterns, or classifications
- Audio → Speech recognition to transcribe, then NLP to analyze

Unstructured data isn't unusable—but it adds complexity, time, and specialized techniques. If someone proposes an AI project using free-text case notes or scanned documents, that's a signal the project will require more than basic ML.

# Labeled vs. unlabeled data

A **label** is the outcome being predicted, not the column headers. Availability of labels determines what types of machine learning can be used.

## What "label" actually means:

- The label is the outcome, target, or answer you want the model to learn to predict.
- In a dataset of patients, the label might be: "Was this patient readmitted within 30 days? Yes/No"
- In a dataset of emails, the label might be: "Is this spam? Yes/No"
- The label is what you're trying to predict for future cases where you don't know the answer yet.

## Labeled data:

- You have both the inputs (features) AND the outcome (label) recorded for historical cases.
- Example: Patient records where you know who was readmitted and who wasn't.
- This enables supervised learning—the algorithm can learn the relationship between inputs and outcomes.

## Unlabeled data:

- You have inputs but NOT the outcome.
- Example: Patient records, but readmission status was never tracked.
- This limits you to unsupervised learning—finding patterns or structure, but not predicting a specific outcome.

# Where labels come from

Source	Example	Consideration
Recorded in normal operations	Claims data includes whether patient was readmitted	Best case—labels already exist
Created from historical decisions	"Fraud" label based on cases that were investigated	Labels reflect past decisions, which may be biased
Manually created for the project	Staff review 1,000 cases and label them	Time-consuming and expensive; need enough labeled examples
Doesn't exist	"Success" was never defined or tracked	Can't do supervised learning without creating labels first

Labels don't appear magically—someone or something recorded them. And the way labels were created matters. If "fraud" labels come from who got investigated (not who actually committed fraud), the model learns to predict who gets investigated, not who commits fraud.

# Connection to learning paradigms

Data Situation	Learning Paradigm	What You Can Do
Have labels	Supervised	Predict outcomes (classification, regression)
No labels	Unsupervised	Find patterns, groups, structure (clustering, dimensionality reduction)

# Volume: How much data is available?

ML learns from examples. More examples generally means better learning but what is "enough" depends on the problem.

- ML algorithms find patterns by seeing many examples. Too few examples, and the algorithm can't learn reliably.
- There's no magic number, but rough guidelines help set expectations.

Situation	Rough Minimum	Notes
Simple problem, structured data	Hundreds to low thousands	Logistic regression, simple decision trees
Moderate complexity	Thousands to tens of thousands	Random forests, gradient boosting
Complex patterns, many variables	Tens of thousands or more	May need more sophisticated approaches
Deep learning (images, text)	Often tens of thousands to millions	Data-hungry techniques

- **Too little data:** Model may memorize the training examples rather than learning generalizable patterns (overfitting).
- **Rare outcomes:** If you're predicting something that happens 1% of the time, you need enough data to have sufficient examples of that rare event. For example, if readmission happens 10% of the time and you have 500 records, you only have ~50 readmission cases to learn from.

# Quality: Is the data any good?

Data quality issues can undermine or derail an AI project. Garbage in, garbage out.

- Real-world data is messy. It was usually collected for operational purposes, not for ML.
- Quality issues are the norm, not the exception. The question is how severe they are and whether they can be addressed.

## Common Quality Issues

Issue	Example	Why It Matters
Missing values	30% of records have no income data	Model may learn wrong patterns or exclude too many cases
Inconsistent coding	"Diabetes," "DM," "Type 2 DM," "diabetic" all mean the same thing	Model treats them as different; patterns get diluted
Data entry errors	Birthdate of 1/1/1900; weight of 5,000 lbs	Outliers distort learning
Duplicates	Different IDs for the same patient	Inflates apparent data volume
Outdated information	Address from 10 years ago	May not reflect current reality
Inconsistent definitions	"Enroll date" means different things in different systems	Apples-to-oranges comparisons

# Accessibility: Can you actually get to the data?

Just because data exists somewhere doesn't mean you can use it for an AI project. Accessibility barriers are common and often underestimated.

## Common Accessibility Barriers

Barrier	Example	What It Means
Different systems	Data is in three different databases that don't talk to each other	Need to extract and link—may require technical work and approvals
Permissions	No permission to access the data	Need to request access; may take weeks or months
Data sharing agreements	Data belongs to another agency or partner	Need legal agreements that can take months
Privacy and legal restrictions	Data contains PHI, PII, or is subject to consent limitations	May need legal review or de-identification
Format barriers	Data is in paper files, scanned PDFs, or legacy systems	May require extraction or digitization
Political barriers	Data owner is uncooperative or protective	May be insurmountable without executive intervention

# Five dimensions of data

Before proposing or evaluating an AI project, know what questions to ask about data.

Dimension	Key Question
Structure	Is the data structured (rows and columns) or unstructured (text, images, etc.)?
Labels	Do we have the outcome we want to predict recorded? Where did it come from?
Volume	How many records? How many examples of the outcome?
Quality	How complete and consistent is the data? What are the known issues?
Accessibility	Can we actually get to the data? What approvals are needed?

# Working in Posit Cloud

## Tasks:

- Create a posit.cloud account
- Install packages for other code (installs.R).
- Set an Anthropic API key (provided by instructor)
- Generate R code in Claude (math.R).
- Run the generated code (math.R) in posit.cloud

# Create an account



[Log In](#) [Sign Up](#)

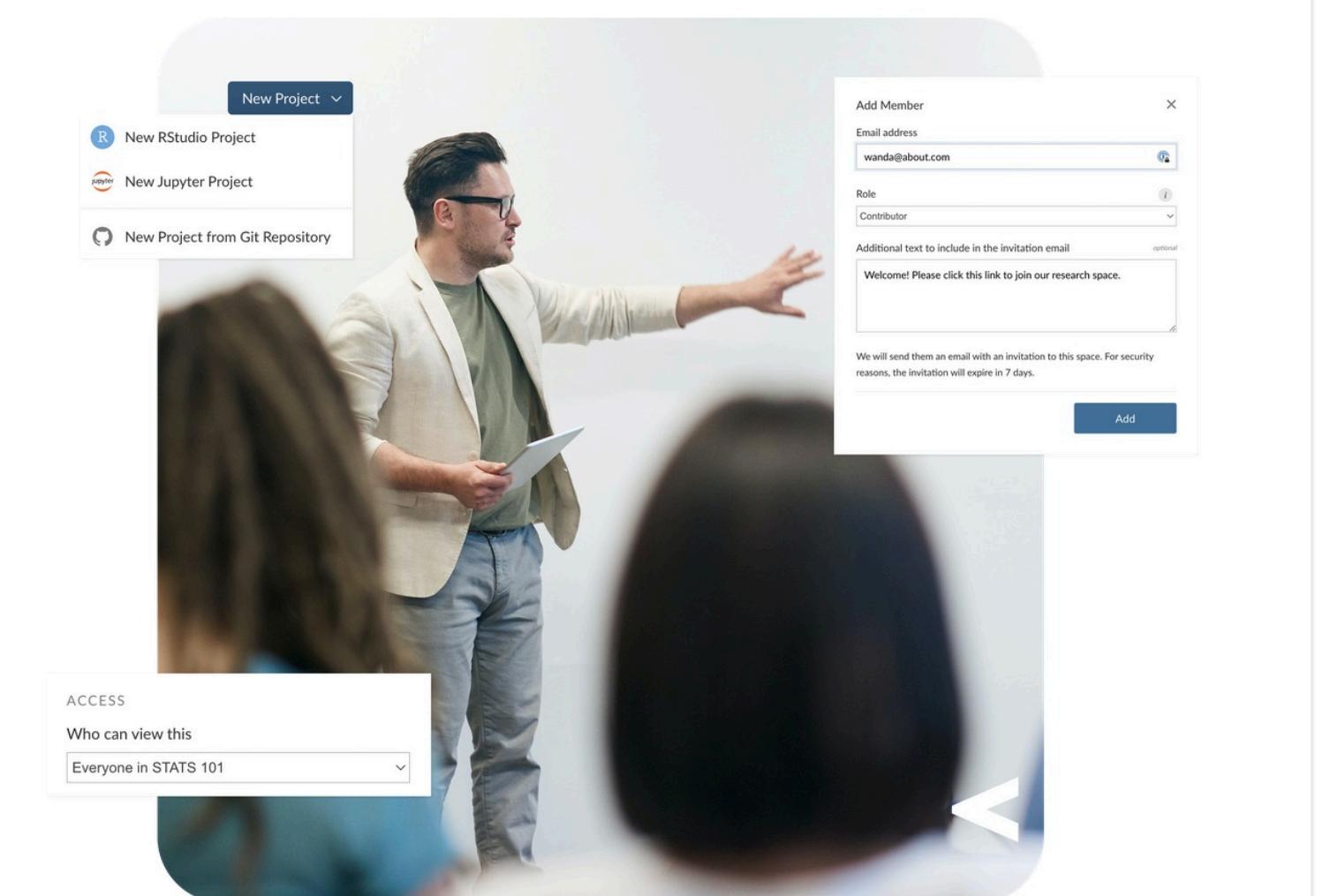
# Friction free data science

Posit Cloud lets you access Posit's powerful set of data science tools right in your browser – no installation or complex configuration required.

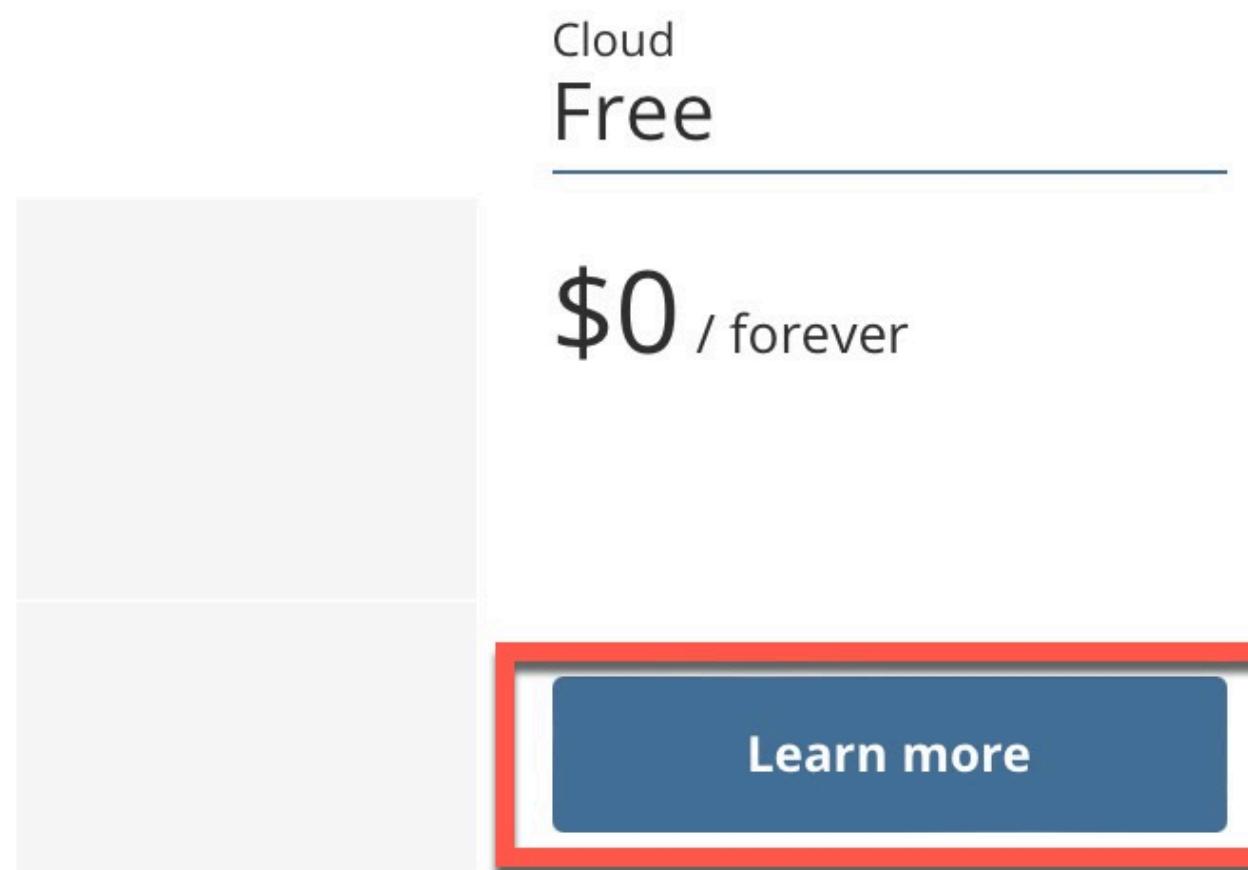
[GET STARTED](#)

[ALREADY A USER? LOG IN](#)

If you already have a shinyapps.io account, you can [log in](#) using your existing credentials.



## Select the Free option



# Enter account information

## Sign Up for Cloud Free

\$0 / forever

Questions? [Contact Us.](#)

If you make limited, occasional use of Posit Cloud, have access to an organization account, or just want to explore Cloud to see if it's right for you, our Cloud Free plan is all you need.

[Sign Up](#)

# Verify email

## Verify Your Email

We sent a link to verify your email address to  
[REDACTED] Please go to your email and click  
that link to verify that we have your correct email  
address - then return to this page and press  
Continue.

[Continue](#)

Wrong email address? [Update your email](#)

Didn't get the email? You can [resend the link](#), but  
you should also make sure our email didn't end up in  
your spam folder.

# Posit.cloud workspace

The screenshot shows the Posit.cloud workspace interface. The left sidebar contains navigation links for 'Spaces', 'Learn', 'Help', and 'Info'. The main content area is titled 'Your Workspace' and includes tabs for 'Content', 'Usage', and 'About'. The 'Content' tab is selected, showing a sub-section for 'Your Content' with options for 'New Project', 'Archive', and 'Trash'. A message at the top states: 'The ability to publish applications and documents to Posit Cloud is now removed for all users. Please visit [Posit Connect Cloud](#) to support your deployment needs.' The bottom of the page features the 'posit Cloud' logo and links for 'Terms' and 'Status', along with social media icons for Facebook, LinkedIn, Instagram, and GitHub.

Your Content - Your Workspace

posit.cloud/content/yours?sort=name\_asc

The ability to publish applications and documents to Posit Cloud is now removed for all users. Please visit [Posit Connect Cloud](#) to support your deployment needs.

posit Cloud JC

Your Workspace JC

Content Usage About

Your Content (0) New Project

ARCHIVE SORT A Z

no content

Spaces

Your Workspace

New Space

Learn

Guide

What's New

Recipes

Cheatsheets

Help

Current System Status

Posit Community

Info

Plans & Pricing

Terms and Conditions

posit Cloud

Terms Status

f in i g

© 2022 Posit Software, PBC

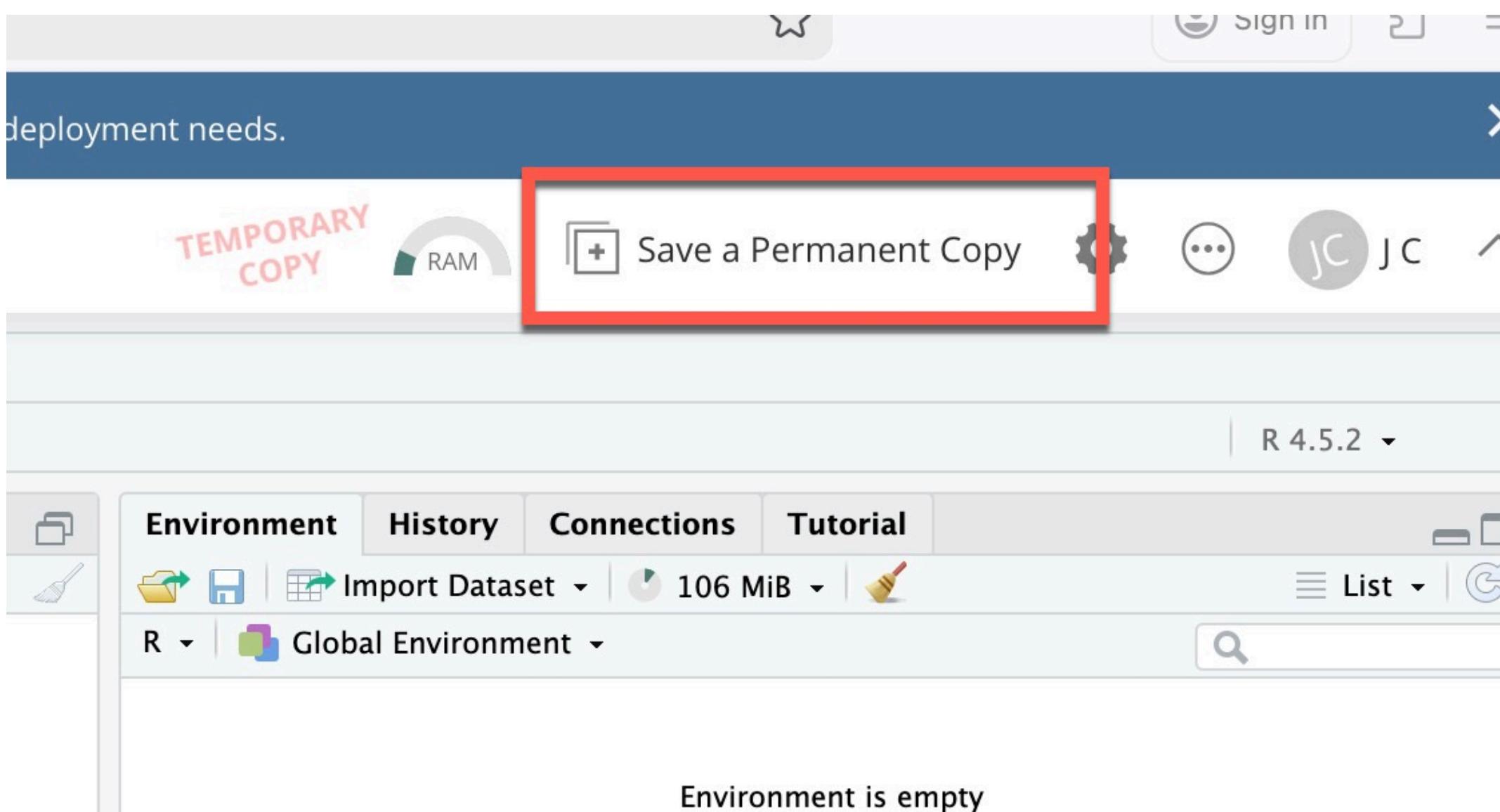
## Copy the instructor project

While logged in to your account, paste this link into a url: <https://posit.cloud/content/11680866>

# Copy in progress



# Save as a permanent copy



# Workspace layout

The screenshot displays the posit Cloud workspace interface with several panels:

- Scripting area:** The central workspace where code can be written and run.
- Console:** A panel at the bottom showing the R environment and current working directory.
- Environment & history:** A panel on the right showing the global environment and history of operations.
- Files available:** A file browser panel showing files in the project directory.

Annotations highlight specific features:

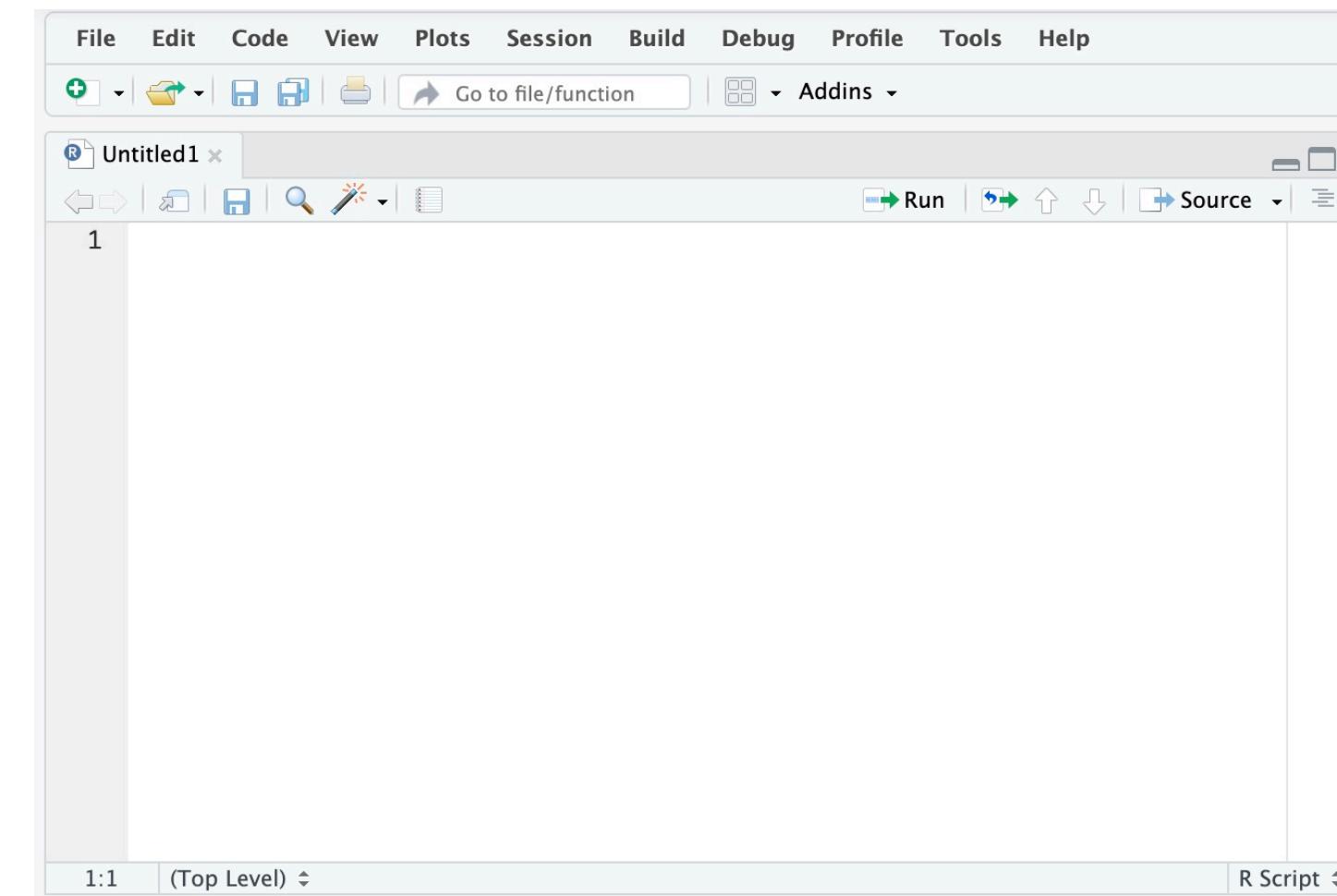
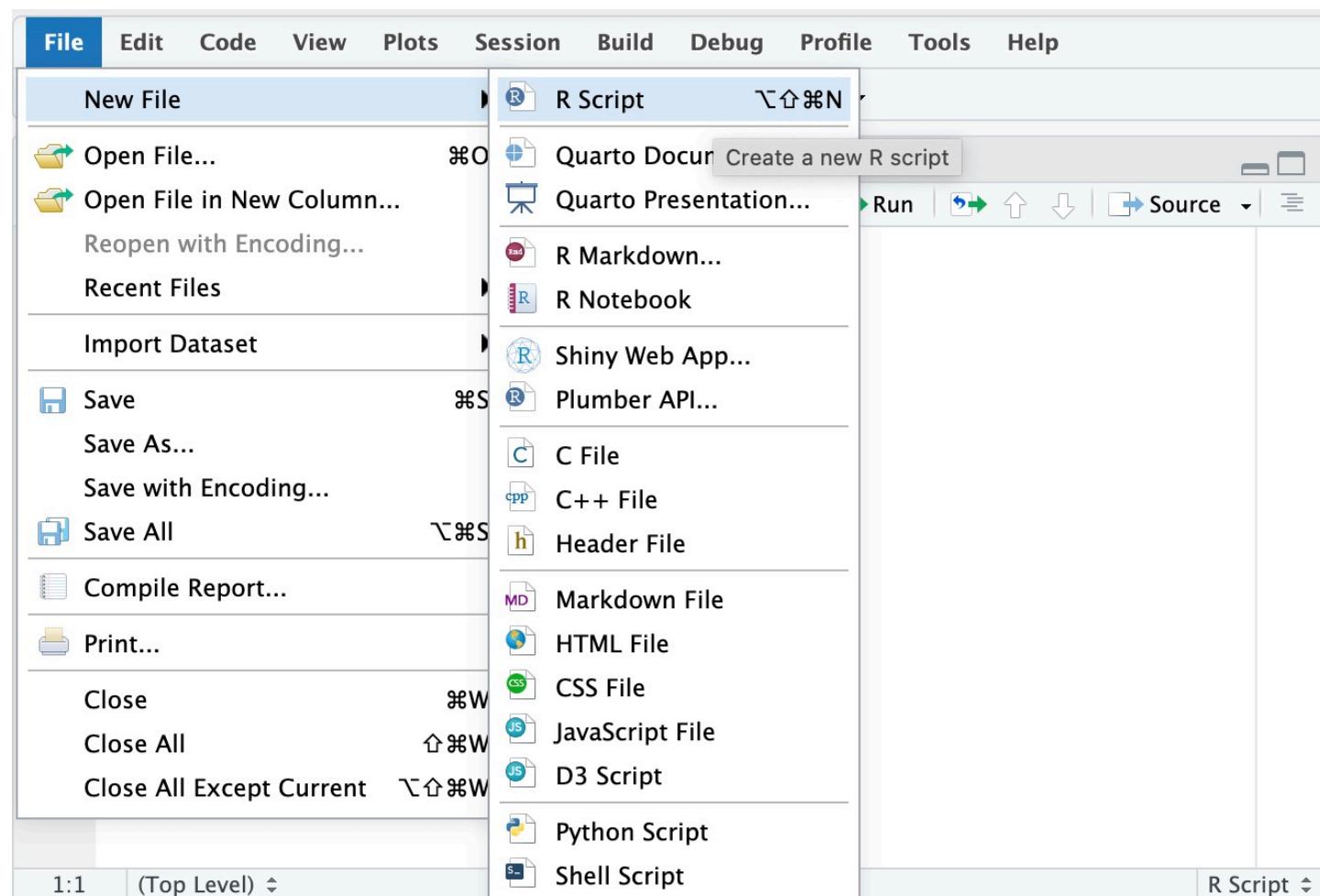
- Scripting area:** Labeled "Scripting area" in the center of the workspace.
- Console for results & messages:** Labeled "Console for results & messages" at the bottom left.
- Environment & history:** Labeled "Environment & history" in the top right.
- Files available:** Labeled "Files available" in the bottom right.

The workspace sidebar includes links for "Your Workspace", "Clone\_Test", "New Space", "Learn", "Guide", "What's New", "Recipes", "Cheatsheets", "Help", "Current System Status", "Posit Community", "Info", "Plans & Pricing", and "Terms and Conditions".

The file browser panel shows the following files:

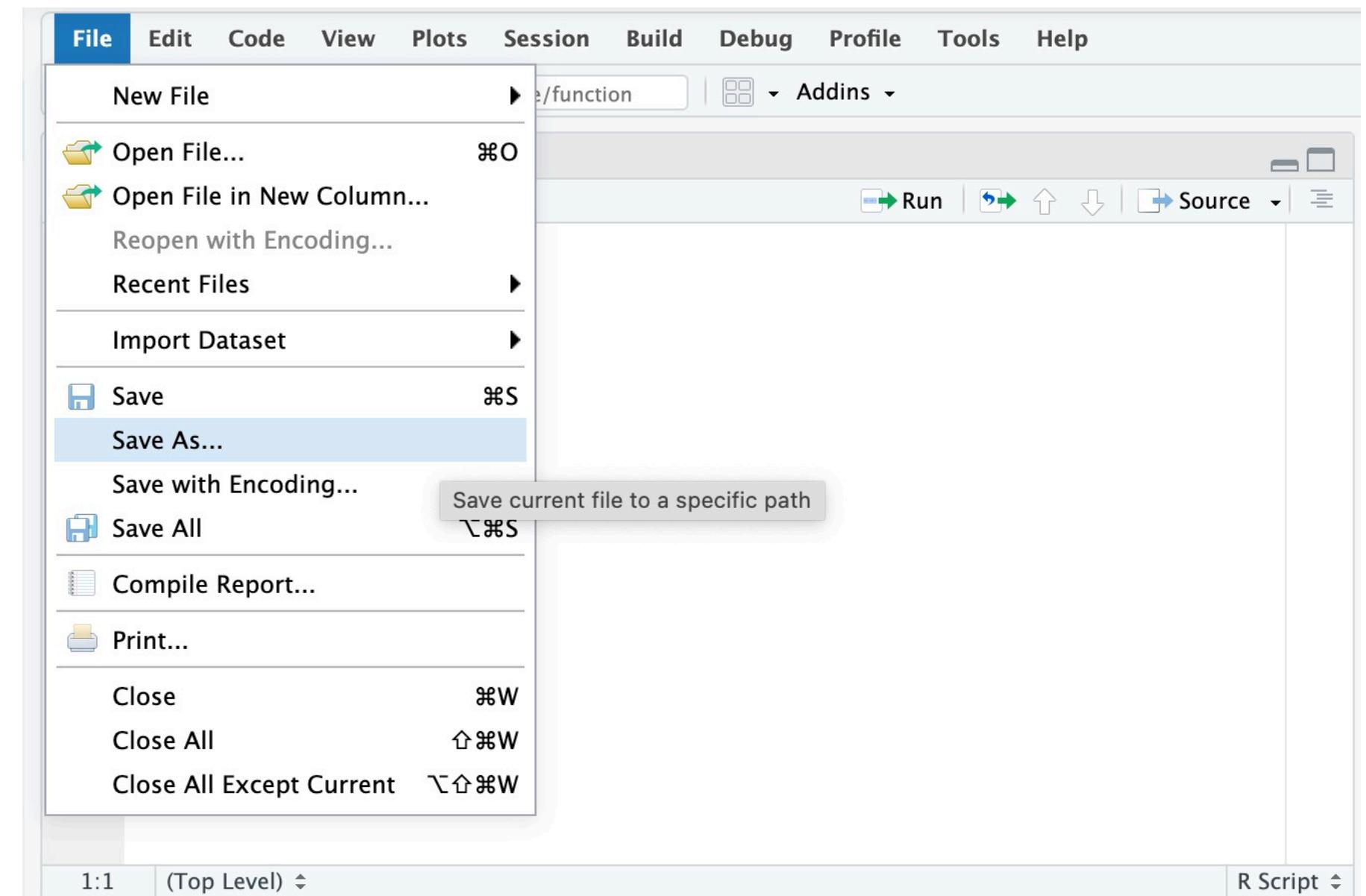
Name	Size	Modified
..		
.Rhistory	0 B	Dec 29, 2025, 3:28 PM
project.Rproj	205 B	Dec 29, 2025, 3:28 PM
math-backup.R	287 B	Dec 29, 2025, 3:30 PM

# New R Script



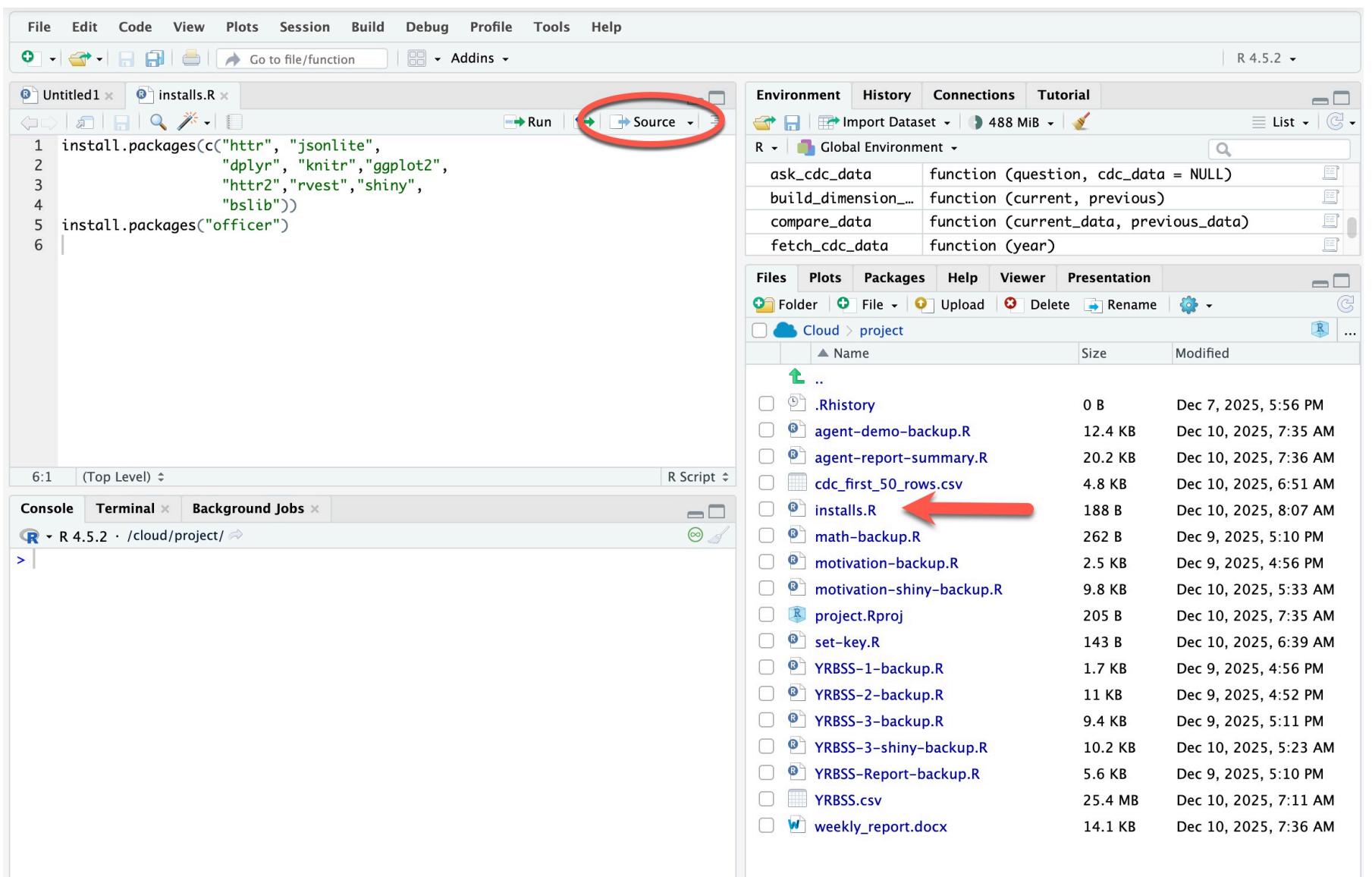
# File → Save As

- From the Menu bar, select File → Save As
- Give the script as name (e.g., test1)
- Press the **Save** button
- The script will appear in the files list.



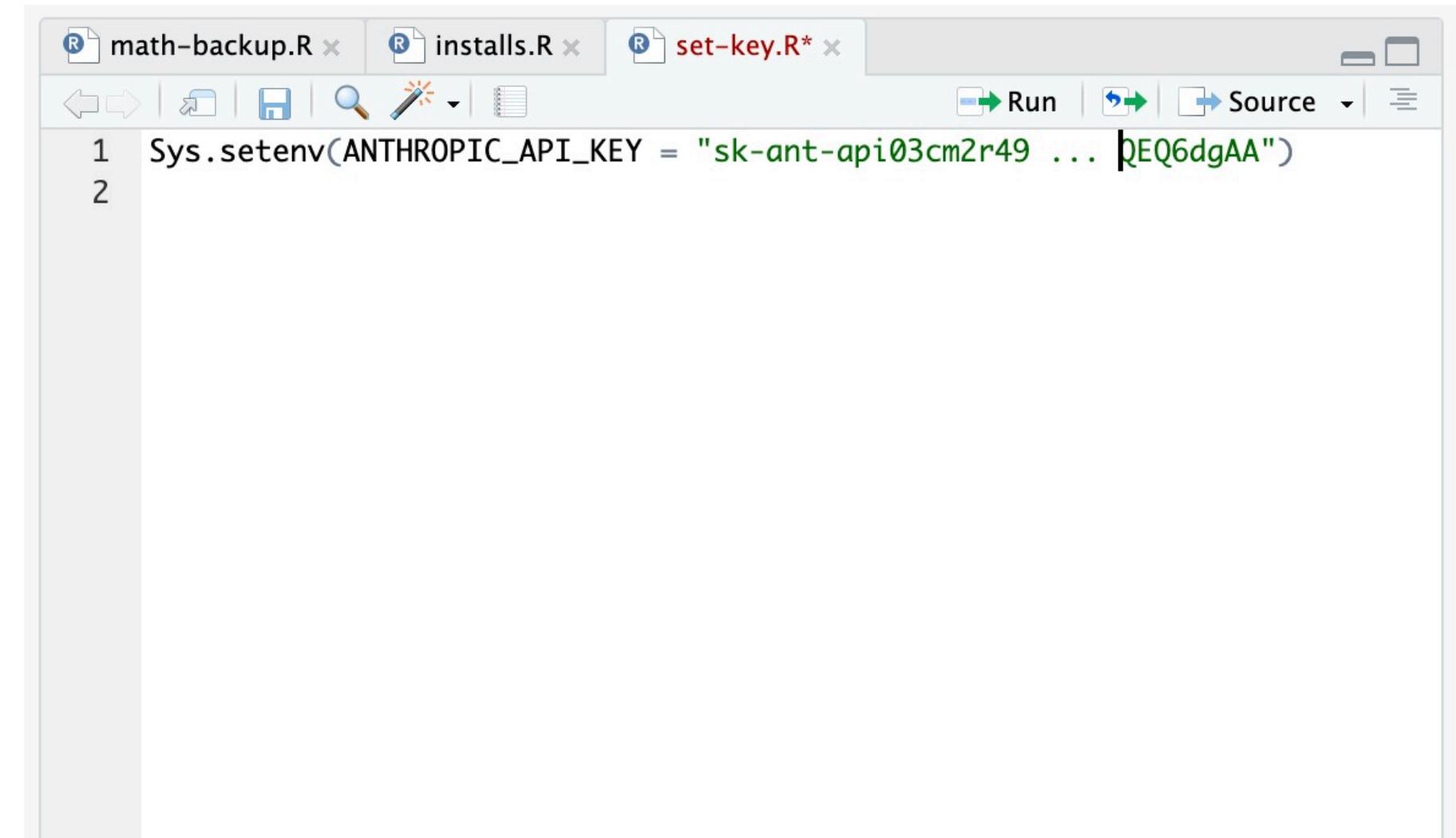
# Install R packages

- In the Files section, click on **installs.R**
- This opens the install code.
- Press **Source** to run the installs. Packages need to be installed in your environment before they can be used.



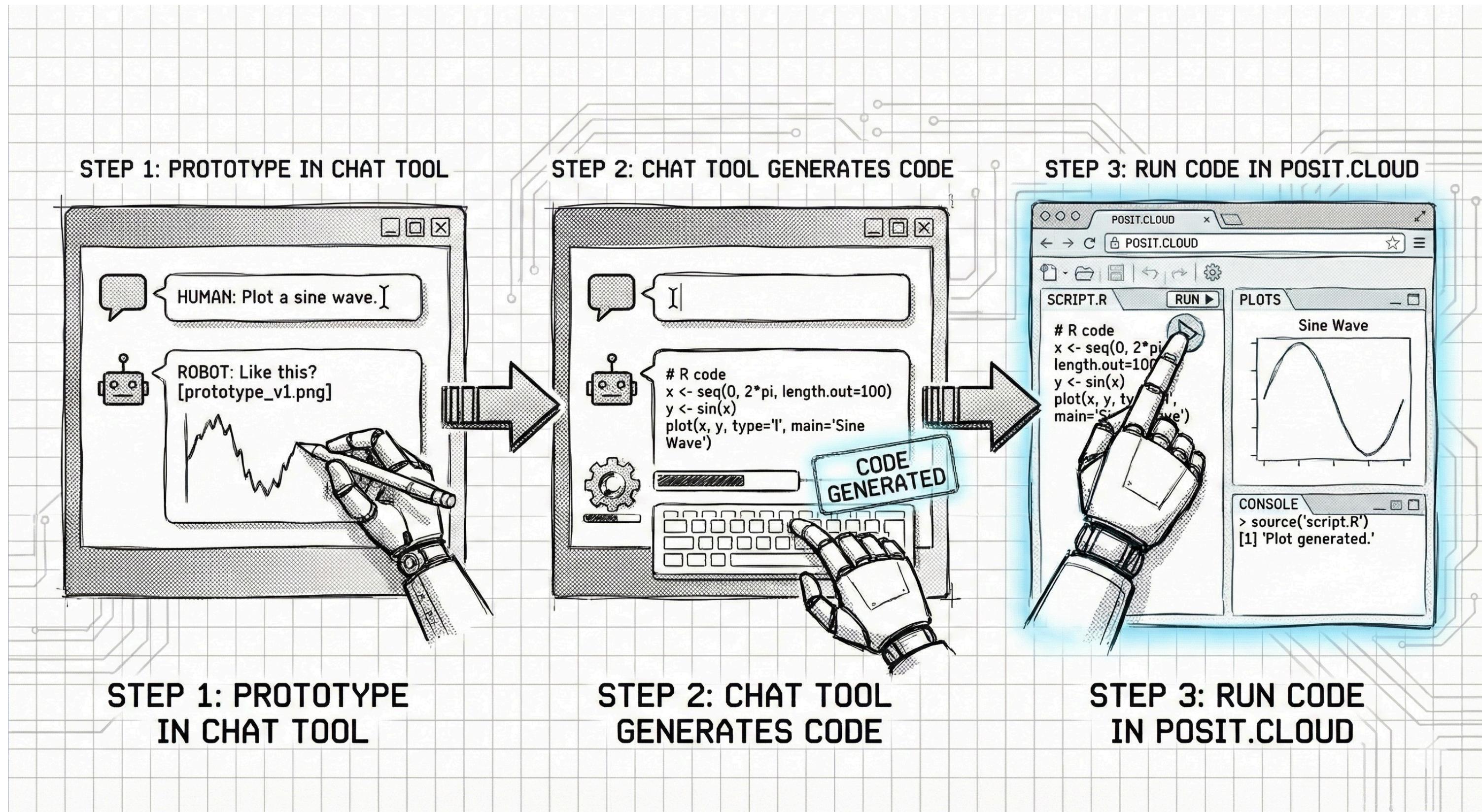
# Set the Anthropic API key

- In the Files section, click on **set-key.R**
- This opens the set-key code.
- Put your Anthropic API key in between the double-quotes
  - This key needs to be in place in order to use Anthropic's (Claude) LLMs.
- Press **Source** to set the key.



```
1 Sys.setenv(ANTHROPIC_API_KEY = "sk-ant-api03cm2r49...QEQ6dgAA")  
2
```

# From Chat to Code



*Prompt: Use the robot hand drawn style. Create an image of images to portray these three steps: prototype in chat tool, chat tool generates code, run code in posit.cloud*

# Code from Claude

## Prompt Instructions

1. I want to provide 2 numbers that you will then multiply together.
2. Generate R code that allows me to interactively enter 2 numbers that will be multiplied together. The results should be displayed in the console.
3. If needed -- Modify the code to use readline

Copy this code if Claude refuses to use readline.

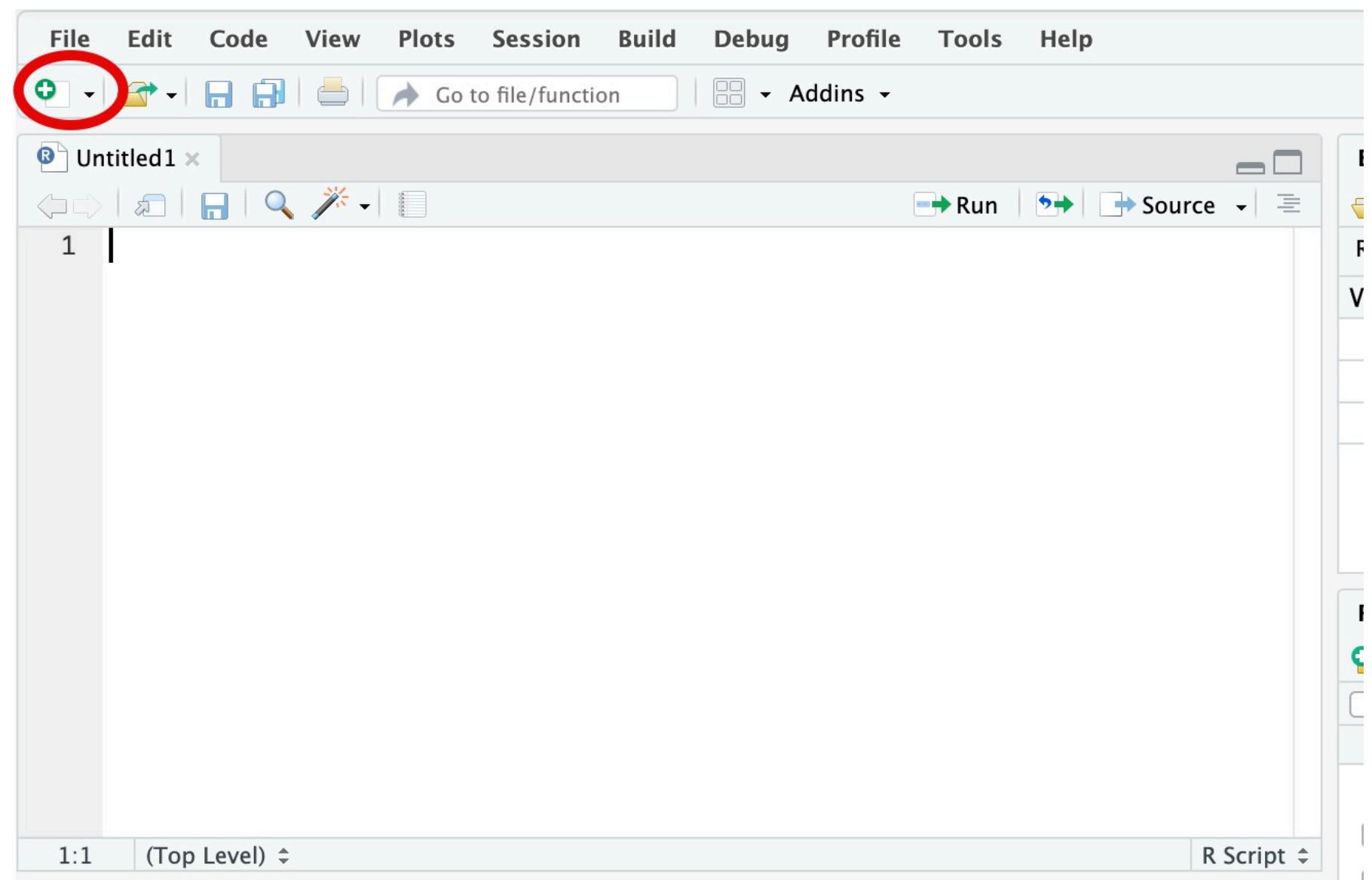
```
# Prompt user for two numbers
num1 <- as.numeric(readline(prompt = "Enter the first number: "))
num2 <- as.numeric(readline(prompt = "Enter the second number: "))
# Multiply the numbers
result <- num1 * num2
# Display the result
cat("The result of", num1, "*", num2, "=", result, "\n")
```

```
r
# Get two numbers from user input
num1 <- as.numeric(readline(prompt = "Enter first number: "))
num2 <- as.numeric(readline(prompt = "Enter second number: "))

# Multiply and display result
result <- num1 * num2
cat("Result:", num1, "*", num2, "=", result, "\n")
```

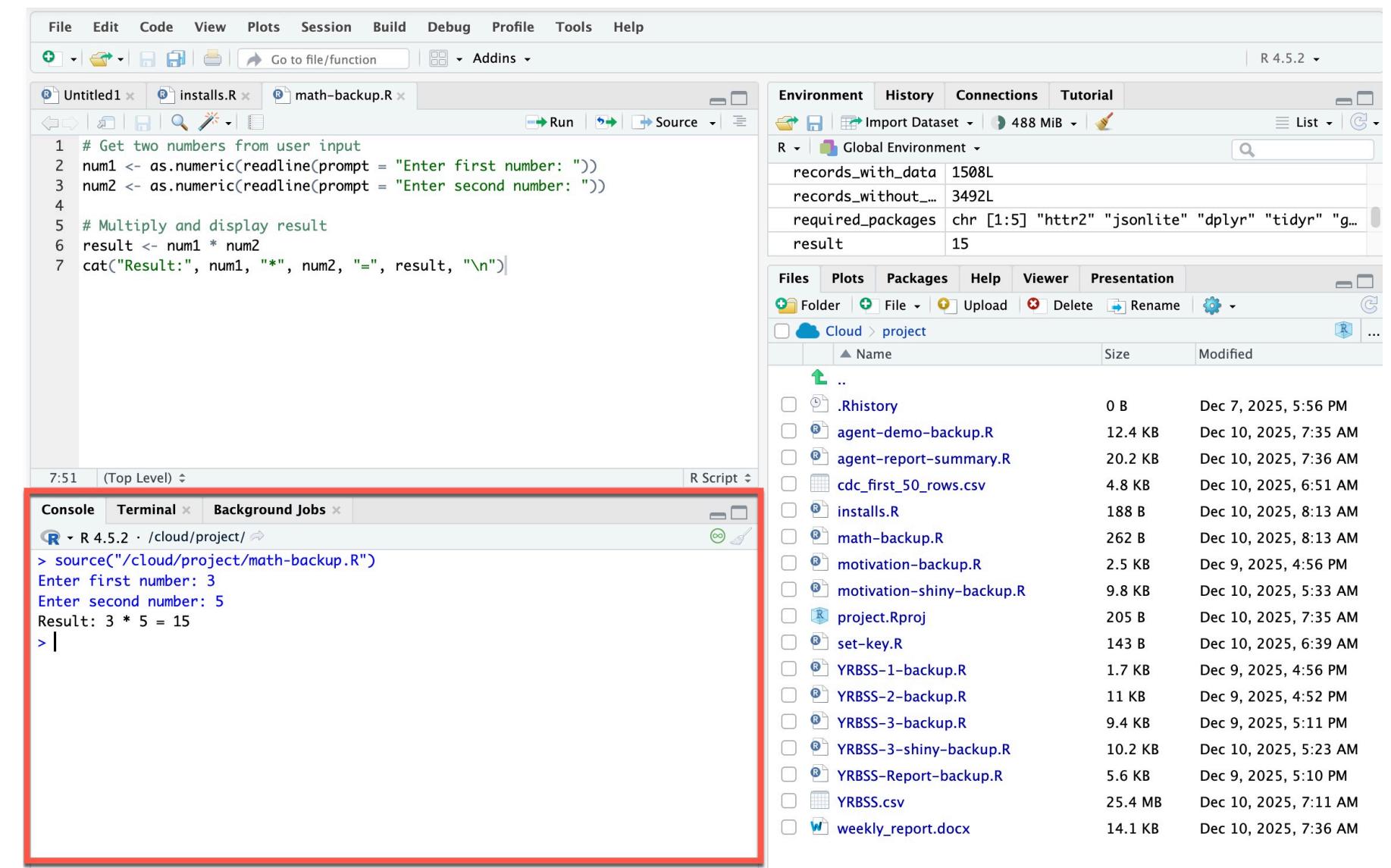
# Create a new R script

- Copy the R code generated by Claude.
- In posit.cloud, select **R Script**. It is the first icon with a plus sign.
- Paste Claude code into posit.cloud
- File → Save As **Math**.
- Press **Source**



# Enter values in the console

- Enter the values in the **Console**
- Review the result.



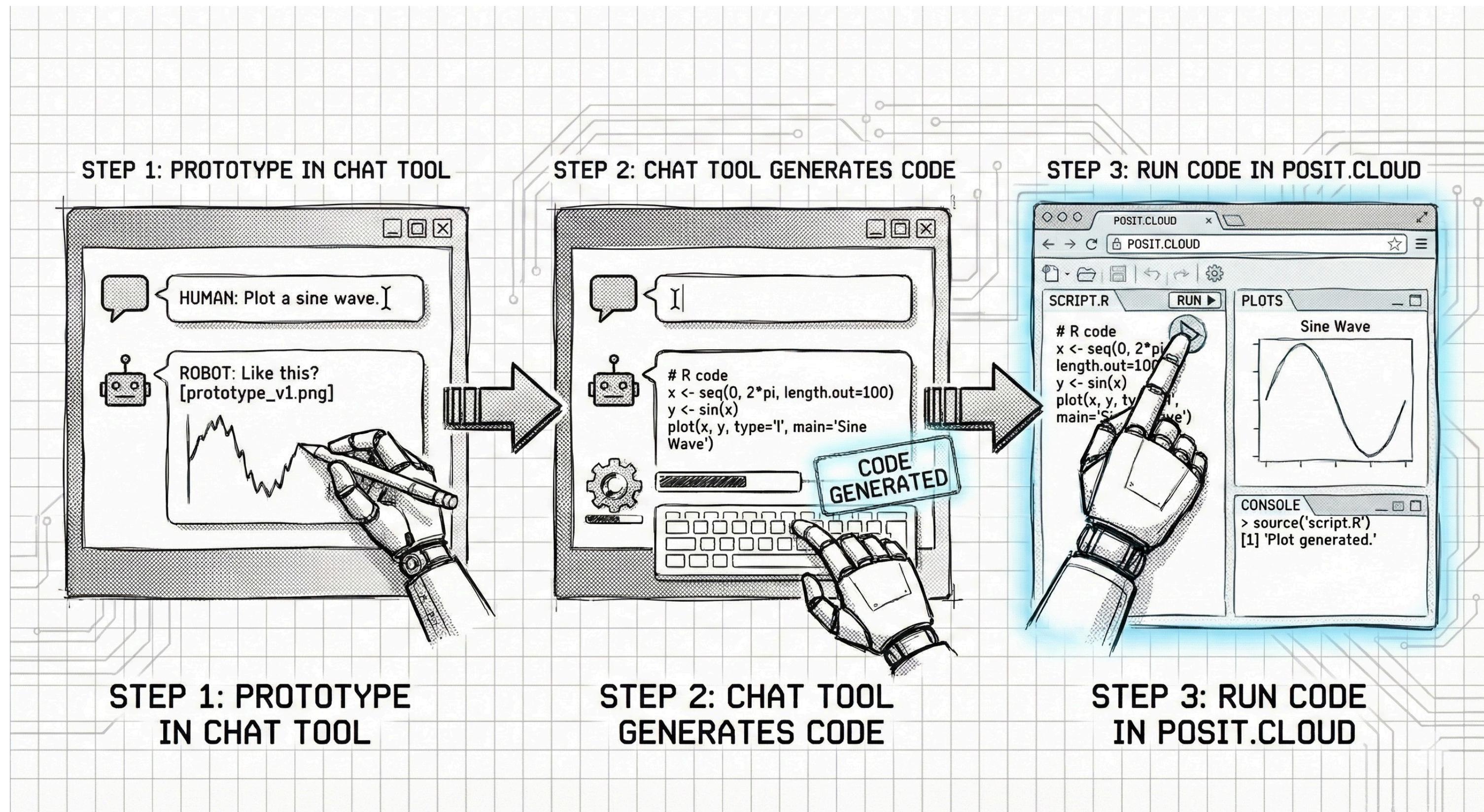
The screenshot shows the RStudio interface with the following components:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Script Editor:** Untitled1, installs.R, math-backup.R. The math-backup.R code is:

```
1 # Get two numbers from user input
2 num1 <- as.numeric(readline(prompt = "Enter first number: "))
3 num2 <- as.numeric(readline(prompt = "Enter second number: "))
4
5 # Multiply and display result
6 result <- num1 * num2
7 cat("Result:", num1, "*", num2, "=", result, "\n")
```

- Environment Pane:** Shows objects like records\_with\_data (1508L), records\_without\_... (3492L), required\_packages (chr [1:5] "httr2" "jsonlite" "dplyr" "tidyR" "g...), and result (15).
- File Browser:** Cloud > project, listing files such as .Rhistory, installs.R, motivation-backup.R, etc.
- Console Tab:** The tab is highlighted with a red box. It shows the command `> source("/cloud/project/math-backup.R")`, followed by user input `Enter first number: 3`, `Enter second number: 5`, and the output `Result: 3 * 5 = 15`.

# From Chat to Code



*Prompt: Use the robot hand drawn style. Create an image of images to portray these three steps: prototype in chat tool, chat tool generates code, run code in posit.cloud*

# Bring in data from an external source

## Prompt Instructions

1. Read the data from this url and show the first 10 lines. Do not do anything else.

[https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic\\_data1K.csv](https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic_data1K.csv)

Hide steps

Thinking about retrieving and displaying initial dataset rows from a CSV file.

Fetched [https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic\\_data1K.csv](https://raw.githubusercontent.com/jmcody2014/BA/refs/heads/main/diabetic_data1K.csv) githubusercontent.com ↗

Thought process

The user asked me to show the first 10 lines of the data. Let me extract just the first 10 lines from this CSV data.

Here are the first 10 lines of the data:

```
encounter_id,patient_nbr,race,gender,age,time_in_hospital,num_lab_procedures,nur
2278392,8222157,Caucasian,Female,[0-10),1,41,0,1,250.83,?,?,None,No,No,NO
149190,55629189,Caucasian,Female,[10-20),3,59,0,18,276,250.01,255,None,Up,Yes,>30
64410,86047875,AfricanAmerican,Female,[20-30),2,11,5,13,648,250,V27,None,No,Yes,NO
500364,82442376,Caucasian,Male,[30-40),2,44,1,16,8,250.43,403,None,Up,Yes,NO
16680,42519267,Caucasian,Male,[40-50),1,51,0,8,197,157,250,None,Steady,Yes,NO
35754,82637451,Caucasian,Male,[50-60),3,31,6,16,414,411,250,None,Steady,Yes,>30
55842,84259809,Caucasian,Male,[60-70),4,70,1,21,414,411,V45,None,Steady,Yes,NO
63768,114882984,Caucasian,Male,[70-80),5,73,0,12,428,492,250,None,No,Yes,>30
12522,48330783,Caucasian,Female,[80-90),13,68,2,28,398,427,38,None,Steady,Yes,NO
```

# What will go into a summary table

## Prompt Instructions

2. *What would you put into a summary table for this data?*

What would you put into a basic summary table?

Catalogued dataset variables and identified key summary statistics.



For this diabetic dataset, a basic summary table would typically include:

### Dataset Overview:

- Total number of records (rows)
- Total number of variables (columns)
- Missing value counts

**Numeric Variables** (time\_in\_hospital, num\_lab\_procedures, num\_procedures, num\_medications):

- Count
- Mean
- Standard deviation
- Min / Max



# Create a summary table

## Prompt Instructions

3. Create a dataset overview and the mean, min and max for the numeric values.
4. Create the summary table and save it as a word document.

=====

DATASET OVERVIEW

=====

Total Records: 105  
Total Variables: 16  
Numeric Variables: 4  
Categorical Variables: 12

=====

=====

NUMERIC VARIABLES SUMMARY

=====

Variable	Mean	Min	Max
-----			
time_in_hospital	4.8	1	14
num_lab_procedures	46.5	2	87
num_procedures	1.6	0	6
num_medications	15.3	1	32

=====

# Summary table in a Word document

## Diabetic Data Summary Table

### Dataset Overview

Metric	Value
Total Records	105
Total Variables	16
Numeric Variables	4
Categorical Variables	12

### Numeric Variables Summary

Variable	Mean	Min	Max
time_in_hospital	4.8	1	14
num_lab_procedures	46.5	2	87
num_procedures	1.6	0	6
num_medications	15.3	1	32

# Code from Claude

## Prompt Instructions

5. Convert this process to R code that writes out a word doc.

The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with icons for file operations. Below it, a header bar says "url data / Reading CSV data from GitHub URL". The main area contains the following content:

- To run locally:**

```
r  
source("create_summary.R")
```
- A note: "The script reads directly from the GitHub URL, but you can change the path to a local file if preferred."
- A sidebar titled "Artifacts" lists two items:
  - "Create summary" (Code - R)
  - "Diabetic summary" (Document - DOCX)
- A section titled "Project content" shows a single item: "url data" (Created by you).

R isn't available on this system, but here's the R code that will create the Word document. I'll save it as a file you can download and run locally:

The screenshot shows a Jupyter Notebook cell containing the following R code:

```
2 steps  
Presented file  
Devised R script for generating Word documents.
```

Here's the R script that creates the Word document. It uses the `officer` and `flextable` packages which are the standard approach for creating Word docs in R.

### Key components:

- 1. Packages:** `officer` (Word doc creation) + `flextable` (formatted tables)
- 2. Process:**
  - Read CSV data
  - Calculate summary statistics
  - Create formatted tables with `flextable()`
  - Build Word doc with `read_docx()` and `body_add_*` functions
  - Save with `print(doc, target = "filename.docx")`

# Working in posit.cloud

## Instructions

6. Create a new R Script.
7. Copy the code generated by Claude into the new, empty script
8. File → Save As 'diabetes-summary'
9. Press **Source** to run the code.
10. Download the Word document (*diabetic\_summary.doc*) and examine it.

The screenshot shows the posit.cloud RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The main workspace contains two R script files: 'diabetes-summary-backup.R' and 'installs.R'. The 'diabetes-summary-backup.R' file contains R code for generating a summary report. The 'installs.R' file is empty. The bottom console window shows the command 'source("./cloud/project/diabetes-summary-backup.R")' being run, followed by the message 'Document saved to: diabetic\_summary.docx'. To the right of the workspace is a file browser titled 'Cloud > project' which lists several files: '.Rhistory' (0 B, Dec 7, 2025, 5:56 PM), 'agent-demo-backup.R' (12.4 KB, Dec 17, 2025, 8:48 AM), 'agent-report-summary.R' (20.2 KB, Dec 12, 2025, 12:23 PM), 'bethany1.R' (599 B, Dec 11, 2025, 10:54 AM), 'cdc\_first\_50\_rows.csv' (4.9 KB, Dec 10, 2025, 9:06 AM), 'create-data.R' (2.2 KB, Dec 19, 2025, 7:36 AM), 'diabetes-summary-backup.R' (3.8 KB, Dec 31, 2025, 2:33 PM), 'diabetic\_summary.docx' (13.4 KB, Dec 31, 2025, 2:33 PM), and 'installs.R' (185 B, Dec 31, 2025, 2:26 PM).

# Exercise

## Instructions:

1. In the same project, begin a new chat.
2. Remind Claude not to create or run any code until you say it is ok to do so.

### 3. Ask Claude to:

- a. Use this url to read the data -

[https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic\\_data101k.csv](https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic_data101k.csv)

- b. This is the same data but has 101K rows.
- c. Display the first 10 rows of data as a table.
- d. Explain how it would go about identifying anything important or interesting in the data.
- e. Ask you 3 important questions before proceeding. Remind Claude to ask them one at a time.
- f. Once you are satisfied, ask Claude to proceed with its analysis.

g. **Note:** Within the tool, code will be written using python. Don't worry, we will have Claude convert to R.

4. Tell Claude you want the code converted to R and you will run it locally.
5. Copy and paste the code into a new posit.cloud R Script. Save as 'diabetes-analysis'.
6. Test to see if it runs. Ignore any warning messages at the end

# Create code that uses machine learning

## Instructions:

1. Create a new project named **Using ML**. All chats for this walkthrough will be in this project.
2. Add an instruction to the project. (Adjust as you see fit)

a. I am a public health professional with little technical and programming experience.

## 3. Ask Claude to:

a. Read the data from this url and tell me how many rows are available. Do not generate any other code until I tell you it is ok to do so.

[https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic\\_data101k.csv](https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic_data101k.csv)

b. Using the downloaded data, give me 3-4 suggestions about how I can go about predicting whether the next patient we encounter will be re-admitted.

c. Are any of these approaches non-ml algorithms?

d. For the ML suggestions, give me a few bullet points for each describing pros and cons.

e. What do you need to know from me in order to use XGBoost?

i. Mine returned 4 questions

ii. If you do not understand what Claude is asking for, ask it for an explanation.

f. Proceed with building the model.

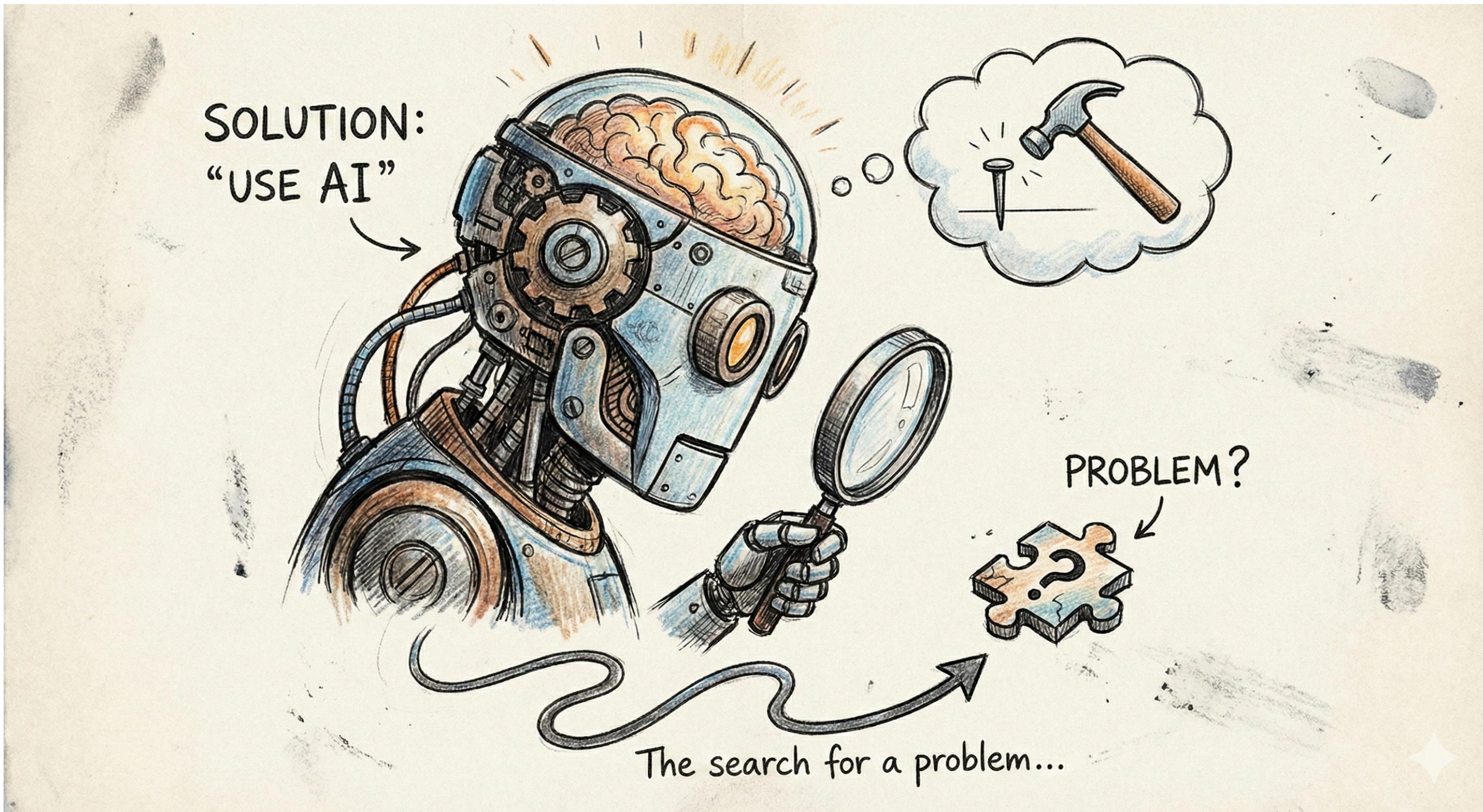
i. **This will take a few minutes.**

g. Convert this code to R. **Do not try to run the R code.** Prepare a file for me and I will run it locally.

h. Look at the code to make sure it is usig the url to get the data. If it is not, tell Claude to use the url.

4. Copy and paste the code into posit.cloud. Name the file predict-readmit. Run the code

# Applying AI: From Directive to Problem Statement



*“Use AI” is not a problem statement—it’s a solution looking for a problem. That is a clarifying statement, not a criticism. Before we can evaluate whether AI is appropriate, we need to articulate what we’re actually trying to accomplish.*

# Common Ways Directives Arrive (and How to Unpack Them)

Directives come in predictable flavors; each has a different underlying question to surface.

Directive Type	Example	Underlying Question to Surface
Vague area	"We should use AI to improve disease surveillance."	What specific outcome? Measured how? What's not working now?
Task-based	"Can AI help us with our AI project backlog?"	What's causing the backlog? What decisions are being delayed? What would "helped" look like?
Peer pressure	"Other states are using AI for fraud detection —we should look into that."	Is our context similar? Do we have the same problem? The same data?
Buzzword-driven	"Let's use AI to be more data-driven."	What decisions aren't currently informed by data? What would change if they were?
Solution-embedded	"We need a chatbot for routine public inquiries."	What problem is the chatbot solving? High call volume? Long wait times? Repetitive questions?

# The Anatomy of a Good Problem Statement

A good problem statement is specific, outcome-focused, and solution-agnostic.

A complete problem statement answers:

## 1. What decision or action will this inform?

- Not "improve outcomes" but "identify which patients to prioritize for outreach"
- Not "be more efficient" but "reduce time spent manually categorizing incoming requests"

## 2. Who needs the output and when?

- Is this for frontline staff making daily decisions? For leadership reviewing quarterly trends? For an annual report?
- Real-time? Daily batch? One-time analysis?

## 3. What does success look like?

- How will you know it worked?
- Is there a measurable target? (e.g., "reduce no-show rate from 28% to 20%)
- What's the counterfactual—what happens if this isn't done?

## 4. What's the current state?

- How is this handled today?
- What's the baseline performance?
- Why isn't the current approach working?

# Example Transformation

## Before

"Use AI to reduce no-shows at WIC clinics."

## After

"Identify WIC participants likely to miss their upcoming appointment so clinic staff can send targeted reminders 48 hours in advance. Success = reduce no-show rate from 28% to 20% within 6 months. Currently, reminders are mailed to everyone 3 days before."

# Red Flags in Problem Framing

Some problem framings are warning signs that more work is needed before proceeding.

Red Flag	Example	Why It's a Problem
No clear outcome	"Make things better" / "Improve quality"	You can't measure success; you can't define what AI would produce.
Solution embedded	"We need a chatbot" / "Build a predictive model"	You've skipped problem definition; you may be solving the wrong thing.
Too broad	"Improve public health" / "Reduce costs"	There's no specific decision or action; this is a goal, not a problem.
Too narrow (premature)	"Predict which of our 50 clients will default"	The volume may not justify AI; you've scoped before understanding.
Unmeasurable	"Make staff feel more supported"	You can't define or detect success.
No baseline	"We want to be faster"	Faster than what? How fast is it now?

# The Reframing Conversation

Reframing isn't just an internal exercise. It often requires a conversation with the person who issued the directive.

- Pushing back on framing can feel risky, but it's essential.
- The goal isn't to say "no". It's to say "let me make sure I understand what we're trying to accomplish."
- Reframing is part of the process.

"I want to make sure we're solving the right problem. Can you tell me more about what's not working today?"

"When you say 'improve X,' what would that look like if it worked? How would we know?"

"Other states are doing this—do we know if their situation is similar to ours?"

"If we had a perfect solution, what would you do differently tomorrow?"

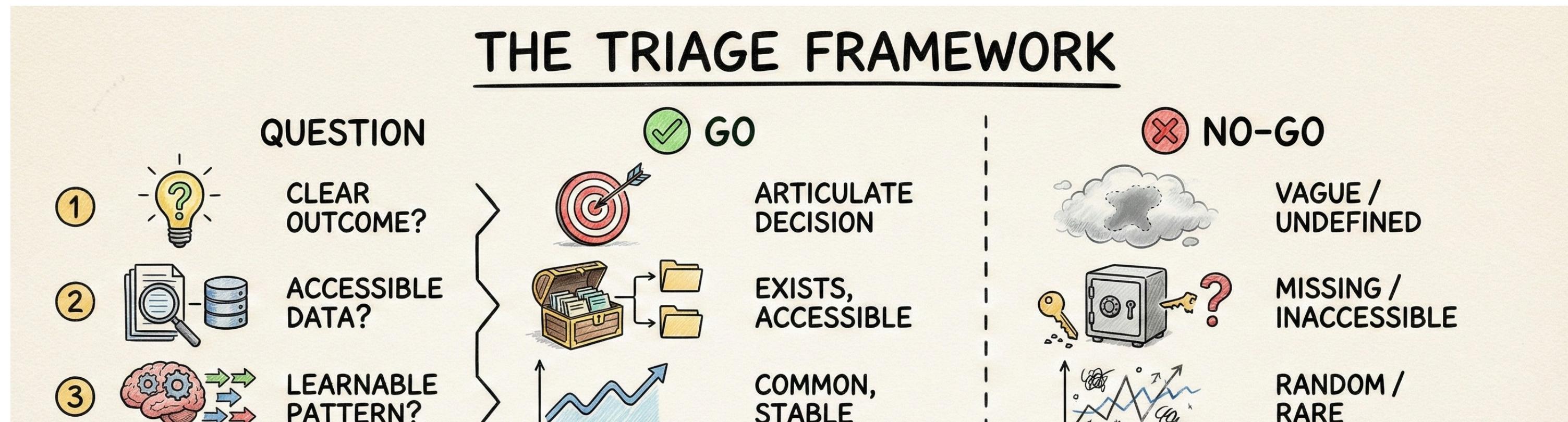
# From Directive to Problem Statement Summary

Component	Key Takeaway
Why "use AI" isn't a starting point	AI is a tool; surface the underlying pain point
How directives arrive	Recognize the type; have questions ready
Anatomy of a problem statement	Four elements: decision, audience/timing, success, current state
Red flags	Spot vague, embedded, or unmeasurable framings
The reframing conversation	Have language to push back diplomatically

# Draft AI Directives (20 minutes in breakout rooms)

The following statements represent high-level, intentionally vague directives intended for staff to translate into well-scoped projects. Select 2 - 3 statements and add context to make them more specific. Bullet points are ok for this exercise.

1. Use AI to make our reports faster.
2. Figure out how AI can help detect problem signals early.
3. Create something that uses AI to help states submit better data.
4. Develop an AI tool that can answer questions about problem patterns.
5. Look into AI for automating routine tasks in the branch.
6. Build an AI system that reads every email we get and automatically tells us what to do with it. == Applying AI: The Problem Triage Framework (Triage)



# Is There a Clear Outcome or Decision?

If you can't define what success looks like or what decision the AI would inform, stop here.

- This is the first gate. Without a clear outcome, everything else is speculation.
- The outcome should be specific: what exactly will the model produce?
- The outcome should be tied to a decision or action: who will use this output, and for what?
- Vague outcomes ("improve things," "be more efficient") are red flags.

## Diagnostic Questions

Question	What You're Looking For
What would the model output?	A specific prediction, classification, score, or generated content
What decision does this inform?	A concrete action someone would take based on the output
Who uses the output?	A named role or team with capacity to act
How would you know it worked?	A measurable change in outcomes, efficiency, or accuracy

# Clear vs. Unclear Outcomes

## Clear Outcome

Flag patients with >50% probability of 30-day readmission

Classify emails as complaint, question, or compliment

Generate a first draft of the weekly surveillance summary

## Unclear Outcome

Improve patient outcomes

Make the inbox more manageable

Help with reporting

## Go/No-Go:

- **Go:** You can write a sentence: "The model will produce [X] so that [person/team] can [decision/action]."
- **No-Go:** You cannot complete that sentence, or it's vague.

# Is There Data—and Is It Accessible?

No data, no ML. But "data exists" isn't enough—you need data you can actually get your hands on.

- AI learns from examples. You need historical data that includes both the inputs (features) and the outcomes (labels).
- Existence isn't enough: the data must be accessible, usable, and of sufficient quality.
- Common blockers: data is in another system, requires legal agreements, is on paper, or has quality issues.

## Diagnostic Questions

Question	What You're Looking For
Does the data exist?	Confirmed, not assumed
Where does it live?	Specific system, database, or files
Who controls access?	Named owner or governance process
What approvals are needed?	Data use agreements, IRB, legal review
Does it include the outcome?	The thing you're trying to predict must be recorded
How much data is there?	Thousands of examples is usually a minimum
What's the quality?	Completeness, consistency, accuracy

# Common Data Blockers

Blocker	Example
Data doesn't exist	"Success" was never tracked; outcome is undefined
Data exists but is inaccessible	Records are in another agency's system with no data-sharing agreement
Data is on paper or in PDFs	Would require manual extraction or OCR
Data is too sparse	Only 50 examples of the outcome in 5 years
Data quality is poor	Inconsistent coding, missing fields, duplicates
Data doesn't include the outcome	You have inputs but not what happened next

Go/No-Go:

- **Go:** Data exists, is accessible, includes the outcome, and has reasonable quality and volume.
- **No-Go:** Any of the above blockers are present and can't be resolved.

# Is There a Pattern to Learn?

AI finds patterns. If the phenomenon is random, rare, or driven by factors not in your data, AI won't help.

- AI needs signal—a relationship between inputs and outcomes that holds across examples.
- If the outcome is essentially random, or driven by factors you can't measure, there's no pattern to learn.
- If the outcome is extremely rare, there may not be enough examples to learn from.
- If the underlying process has changed, historical patterns may not apply.

## Diagnostic Questions

Question	What You're Looking For
Is there reason to believe inputs relate to the outcome?	Prior research, domain knowledge, or exploratory analysis
Is the outcome common enough?	Enough positive and negative examples to learn from
Is the process stable?	Patterns from the past are likely to hold in the future
Are the drivers in your data?	The factors that influence the outcome are captured

# Warning Signs

Warning Sign	Example
Outcome is essentially random	Predicting lottery numbers
Outcome is extremely rare	3 events in 20 years
Key drivers aren't in the data	Compliance depends on family support, which isn't recorded
Process has shifted	Policy change makes historical patterns obsolete
Adversarial context	Fraudsters adapt once they know you're predicting

Go/No-Go:

- **Go:** There's reason to believe a learnable pattern exists, the outcome is common enough, and the process is stable.
- **No-Go:** The outcome is random, too rare, or the drivers aren't in your data.

# The Problem is Repeatable. Wait.... What?

AI shines when you need to make similar decisions many times. One-off analyses rarely justify the investment.

**One-off analyses rarely justify the investment → Is this really true?**

- Building, validating, and deploying AI takes effort. That effort pays off when making the same type of decision repeatedly.
- One-time questions ("Why did the rate of rural hospital closings increase so dramatically last year?") are often better answered with traditional analysis.
- Repeatability also means the problem will keep occurring. There is ongoing value, not just a one-time fix.

## **High Repeatability (Good for AI)**

Triage 500 incoming applications per week

Flag high-risk patients daily

Classify 10,000 public comments on a rule

Predict monthly caseload

## **Low Repeatability (Probably Not AI)**

Decide whether to open a new regional office

Investigate why one clinic has high turnover

Write the agency's strategic plan

Explain a one-time budget shortfall

# Is the Effort Justified?

Could a simpler rule, checklist, or existing process accomplish the same thing? Is the problem high-volume or high-stakes enough to warrant the investment?

- AI projects aren't free. They require data preparation, model development, validation, deployment, and ongoing maintenance.
- The investment should be proportional to the value.
- Sometimes a simple rule or manual process is good enough—or even better.

## Diagnostic Questions

Question	What You're Looking For
What's the volume?	How many decisions per week/month/year?
What's the cost of the current process?	Staff time, errors, delays
What's the cost of errors?	Consequences of getting it wrong
Could a simple rule work?	Can an expert write down the logic?
Is the improvement worth the investment?	Realistic estimate of benefit vs. cost

# When AI May Not Be Justified

Situation	Better Alternative
Low volume (dozens per year)	Manual review
Logic is simple and documented	Rule-based system or checklist
Current process works well	Don't fix what isn't broken
Marginal improvement expected	Invest elsewhere
One-time analysis	Traditional statistics or manual review

Go/No-Go:

- **Go:** The problem is high-volume, high-stakes, or high-cost, and simpler alternatives won't suffice.
- **No-Go:** A simpler approach would achieve most of the benefit at a fraction of the cost.

# Can You Act on the Output?

A prediction is useless if there's no intervention pathway. Knowing the answer must change what you do.

- This is often the forgotten question. People get excited about prediction without asking: "Then what?"
- Actionability requires: someone who will act, capacity to act, an intervention that works, and a process to deliver the output.
- If you can't act—due to capacity, authority, or policy—the AI output is wasted.

## Diagnostic Questions

Question	What You're Looking For
What would you do with the output?	Specific intervention or decision
Who would act on it?	Named role or team
Do they have capacity?	Staff time, resources, bandwidth
Is there an intervention that works?	Evidence that the action improves outcomes
Is there a process to deliver the output?	Integration into workflow

# Actionability Failures

Failure	Example
No capacity	Model flags 500 patients, but only 2 care coordinators available
No authority	Model identifies fraud, but only another agency can investigate
No intervention	Model predicts turnover, but there's no retention program
No integration	Output sits in a report no one reads
Staff override	Recommendations are routinely ignored

Go/No-Go:

- **Go:** There's a clear action, a capable actor, and a process to deliver the output.
- **No-Go:** Any link in the action chain is broken.

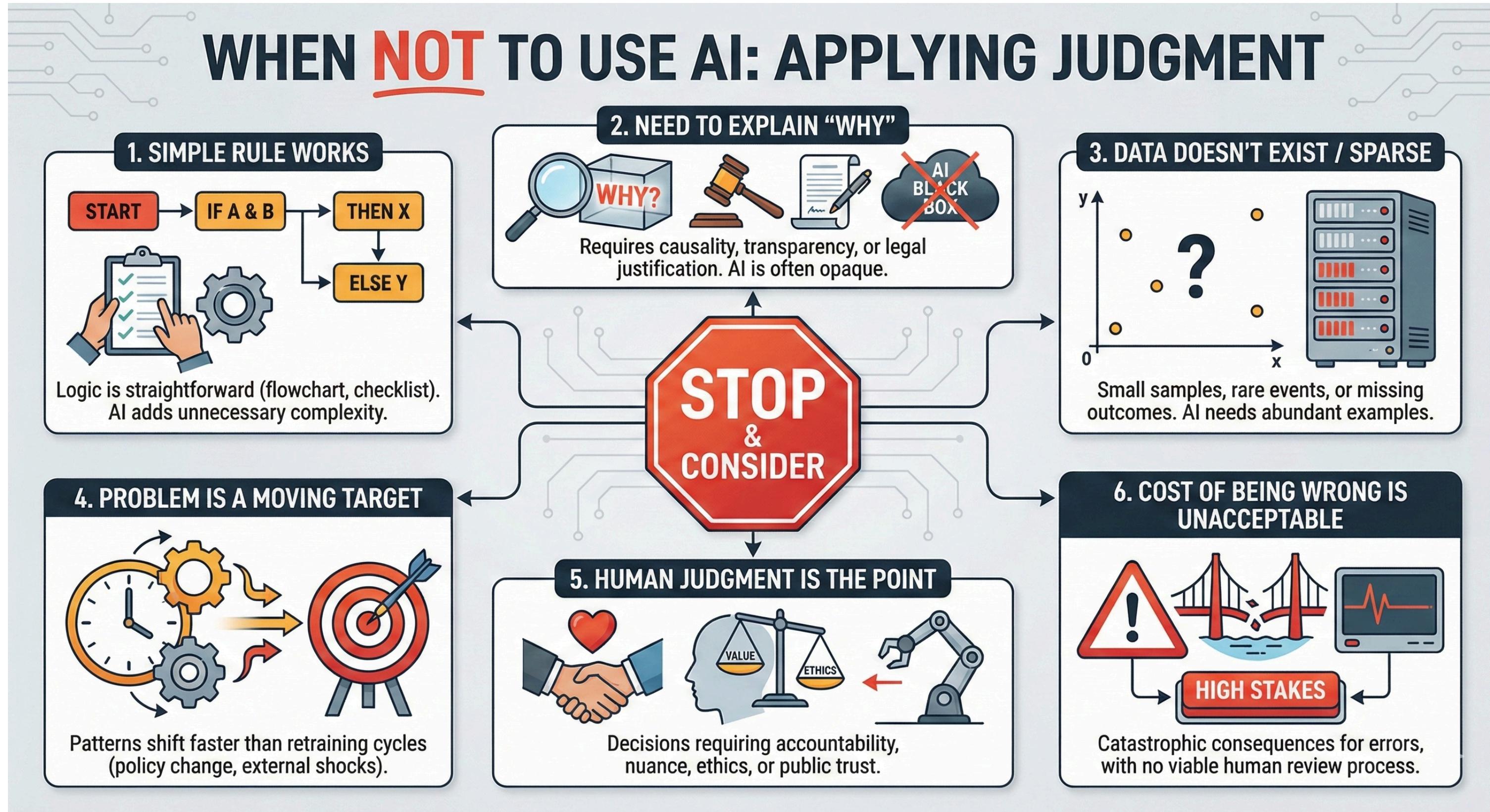
# Putting It Together: The Triage Decision

The five questions form a sequential filter. Failing any one is a reason to pause—not necessarily to stop forever, but to address the gap.

#	Question	Go	No-Go
1	Is there a clear outcome or decision?	Can articulate output and decision	Vague or undefined
2	Is there data—and is it accessible?	Exists, accessible, includes outcome	Missing, inaccessible, or poor quality
3	Is there a pattern to learn?	Learnable, common enough, stable	Random, rare, or shifted
4	Is the effort justified?	High-volume, high-stakes, no simpler way	Low volume or simple rule works
5	Can you act on the output?	Clear action, capable actor, integrated	No capacity, authority, or pathway

# Applying AI: When AI is Overkill or Wrong

Knowing when **not** to use AI is as valuable as knowing when to use it.



# When a Simple Rule Works

If the logic can be written as a flowchart or a few IF-THEN statements, AI is not needed. Explainability and maintainability favor simple rules.

- AI is powerful when patterns are complex and hard to articulate. But many decisions follow straightforward logic.
- Rules are transparent, easy to audit, and easy to change.
- AI adds complexity, maintenance burden, and opacity—only justified when rules can't capture the pattern.

## Signs a Rule Might Be Enough

Situation	Example
Logic is already documented	Eligibility criteria for a benefit program
Decisions follow a checklist	Inspection pass/fail based on defined criteria
Experts can articulate the logic	"If A and B, then X; otherwise Y"
Exceptions are rare and known	A few edge cases that can be handled manually

## Questions to ask:

- Can a subject matter expert write down the decision rules?
- How often do exceptions occur that the rules can't handle?
- Would a decision tree or flowchart capture 90% of cases?

# When You Need to Explain the "Why"

If the primary goal is understanding causal mechanisms or defending a decision in a legal or policy context, classical statistics may be more appropriate than AI.

- AI optimizes for prediction, not explanation. Many models (especially deep learning) are "black boxes."
- Some contexts require explainability: legal proceedings, regulatory compliance, public accountability, clinical decision-making.
- Classical statistics—regression, hypothesis testing—is designed to quantify relationships and attribute effects.

## When Explanation Matters More Than Prediction

Context	Why Explanation Matters
Policy evaluation	Need to know <b>if</b> an intervention caused an outcome, not just predict outcomes
Legal or regulatory decisions	May need to justify why a specific decision was made
Clinical care	Physicians need to explain recommendations to patients
Public accountability	Citizens may demand to know why they were flagged or denied
Root cause analysis	Need to understand <b>why</b> something happened, not just predict it will

## Questions to ask:

- Will someone need to explain **why** this decision was made?
- Is the goal to understand a relationship or to predict an outcome?
- Are there legal, regulatory, or ethical requirements for transparency?

# When Data Doesn't Exist or Is Too Sparse

AI can't conjure patterns from nothing. Small samples, rare events, or missing outcome data are showstoppers.

- AI learns from examples. If there aren't enough examples, especially of the outcome being predicted, the model won't learn well.
- "Enough" depends on the complexity of the problem, but dozens or low hundreds is usually too few for most ML approaches.
- Missing outcome data is particularly problematic: A model can't predict something it's never observed.

## Data Insufficiency Scenarios

Scenario	Problem
Rare outcome	Predicting which of 10,000 facilities will have a catastrophic failure when it's happened 3 times in 20 years
New program	No historical data exists because the program just launched
Outcome not recorded	"Success" was never systematically tracked
Biased sample	Data does not exist for all participants
Data exists but is inaccessible	Records are in paper files, another agency's system, or legally restricted

## Questions to ask:

- How many examples of the outcome do we have?
- Is the outcome recorded consistently and reliably?
- Are there systematic gaps in who or what is represented?

# When the Problem Is a Moving Target

If the underlying pattern changes faster than the model can be retrained, AI may produce stale or misleading outputs.

- AI models learn patterns from historical data. They assume the future will resemble the past.
- When the underlying process shifts, due to policy changes, external shocks, or evolving behavior, models degrade.
- "Model drift" is manageable with monitoring and retraining, but some contexts change too fast or too unpredictably.

## When Patterns Are Unstable

Situation	Problem
Policy or regulation change	Eligibility rules changed; historical patterns no longer apply
External shock	Pandemic, economic crisis, natural disaster disrupts normal patterns
Adversarial behavior	Fraudsters adapt to detection methods; the pattern shifts as you act on it
Rapid evolution	Technology, markets, or behaviors changing faster than retraining cycles
One-time event	Patterns from a unique historical period won't generalize

## Questions to ask:

- Has the underlying process changed recently?
- How quickly does the environment change?
- Can we retrain frequently enough to keep up?
- Are we trying to predict something that actors will game once they know we're predicting it?

# When Human Judgment Is the Point

Some decisions should remain human because of accountability, nuance, or public trust—even if AI could technically make them.

- Not every decision **should** be automated, even if it **could** be.
- Some decisions require human accountability: someone needs to be responsible for the outcome.
- Some decisions require nuance that AI can't capture: context, relationships, values.
- Some decisions affect public trust: automating them may be legal but unwise.

## When to Keep Humans in the Loop

Situation	Why Human Judgment Matters
High-stakes individual decisions	Denying benefits, removing children, criminal sentencing
Value-laden tradeoffs	Balancing efficiency vs. equity, cost vs. access
Relationship-dependent contexts	Case management, counseling, community engagement
Novel or ambiguous situations	Edge cases that don't fit historical patterns
Public-facing decisions	Where automation could erode trust or appear dehumanizing

## Questions to ask:

- Who is accountable if this decision is wrong?
- Does this decision require context that isn't in the data?
- Is there value in the human interaction itself?

# When the Cost of Being Wrong Is Unacceptable

If a wrong prediction has catastrophic consequences and a human review cannot be built in, the risk may outweigh the benefit.

- All models make errors. The question is whether the cost of those errors is acceptable.
- False positives and false negatives have different consequences; both need to be understood.
- High-stakes contexts may require human review of all AI recommendations, which may negate efficiency gains.

## Error Cost Considerations

Error Type	Low-Cost Example	High-Cost Example
False positive	Sending an unnecessary appointment reminder	Falsely flagging a family for child abuse investigation
False negative	Missing a low-priority inspection	Failing to detect a disease outbreak

## Risk mitigation options:

- Human review of all AI-flagged cases (but this adds cost and delay)
- Human review of high-confidence flags only (but low-confidence cases may be wrong)
- AI as one input among many (but staff may over-rely on it anyway)
- Threshold tuning to favor one error type over another (but tradeoffs remain)

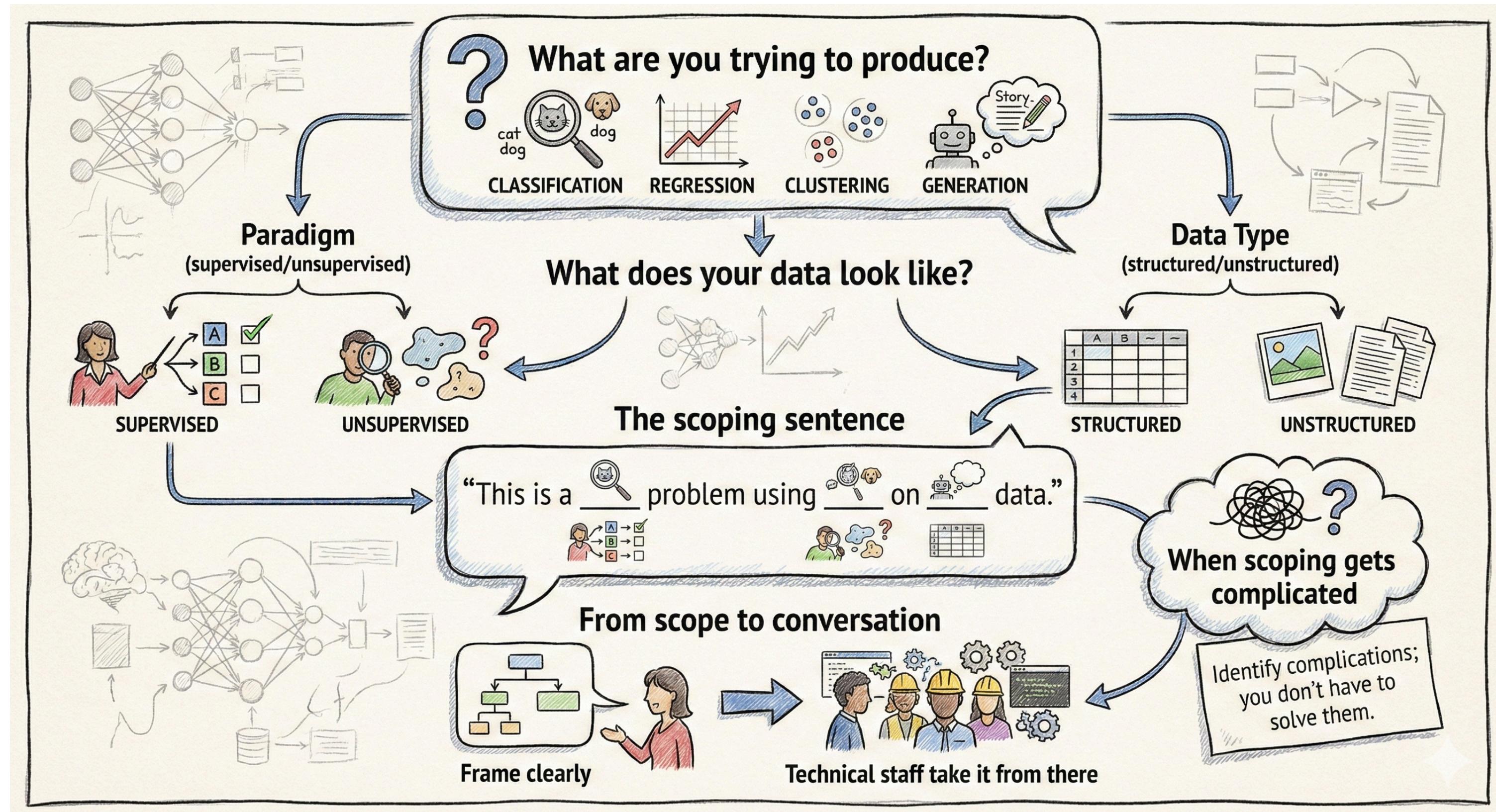
### Questions to ask:

- What happens if the model is wrong? Who is harmed?
- Can we build in human review? At what cost?
- Is the error rate acceptable given the stakes?
- Are we comfortable with the tradeoff between false positives and false negatives?

# When AI is Overkill or Wrong Summary

Component	Key Takeaway
Simple rule works	AI should earn its complexity; rules are transparent and maintainable
Need to explain the "why"	Prediction and explanation are different goals; choose accordingly
Data doesn't exist or is too sparse	No data, no AI; verify before proposing
Problem is a moving target	AI assumes the future looks like the past; unstable contexts mislead
Human judgment is the point	"Can we?" is different from "Should we?"
Cost of being wrong is unacceptable	Every model errs; high stakes demand oversight

# Applying AI: Matching Problems to AI Task Types (Scope)



# What Are You Trying to Produce?

The first scoping question is: what kind of output does the model produce? This determines the task type.

- AI tasks are categorized by what they produce.
- Getting the task type right is essential—it determines what algorithms are appropriate, what data you need, and how you evaluate success.
- The four main task types cover most agency use cases.

## The Four Task Types

Task Type	What It Produces	Example
Classification	A category or label	Is this patient high-risk or low-risk? Is this email a complaint, question, or compliment?
Regression	A number	How many WIC participants will enroll next month? What's the predicted length of stay?
Clustering	Groups or segments	What natural groupings exist among overdose cases? Which facilities have similar inspection profiles?
Generation	New content	A summary of this report. A first draft of this communication.

# How to Identify the Task Type

## If the output is...

Yes/no, or one of several categories

A continuous number (count, amount, duration)

Groups discovered from the data (no predefined categories)

Text, images, or other content created by the model

## Then the task type is...

Classification

Regression

Clustering

Generation

## Common confusion:

- **Classification vs. Regression:** If predicting "will they or won't they," it's classification. If predicting "how much" or "how many," it's regression.
- **Classification vs. Clustering:** Classification assigns to **predefined** categories (supervised). Clustering **discovers** categories from the data (unsupervised).

# What Does Your Data Look Like?

The structure and labeling of data determines the learning paradigm—supervised or unsupervised.

- **Supervised learning:** You have labeled data—examples where you know the outcome. The model learns to predict the label from the inputs.
- **Unsupervised learning:** You don't have labels. The model finds structure or patterns in the data without being told what to look for.
- The distinction matters because it determines what's possible and what data you need.

## Learning Paradigms

Paradigm	What You Have	What the Model Does	Example
Supervised	Labeled data (inputs + known outcomes)	Learns to predict the outcome from inputs	Predict readmission (yes/no) from patient data
Unsupervised	Unlabeled data (inputs only)	Finds patterns, groups, or structure	Group outbreak reports by similarity

# How Task Type and Paradigm Connect

Task Type	Typical Paradigm
Classification	Supervised (you have labels for each category)
Regression	Supervised (you have the number you're predicting)
Clustering	Unsupervised (you're discovering groups, not predicting them)
Generation	Supervised (deep learning trained on examples)

Data Type	Description	Examples
Structured	Organized in rows and columns; each column is a defined field	Spreadsheets, databases, claims data
Unstructured	No predefined format; requires processing to extract information	Free text, images, audio, PDFs

Why it matters:

- Structured data works with most ML algorithms.
- Unstructured data typically requires NLP (for text) or deep learning (for images, audio).
- Many real problems involve a mix—structured fields plus free-text notes.

# Putting It Together: The Scoping Sentence

"This is a [task type] problem using [learning paradigm] on [data type] data."

# Scoping Sentence Examples

Problem	Task Type	Paradigm	Data Type	Scoping Sentence
Predict which patients will be readmitted within 30 days	Classification	Supervised	Structured	"This is a classification problem using supervised learning on structured data."
Estimate the number of WIC enrollees next quarter	Regression	Supervised	Structured	"This is a regression problem using supervised learning on structured data."
Group disease outbreak reports by similarity	Clustering	Unsupervised	Unstructured	"This is a clustering problem using unsupervised learning on unstructured data."
Generate plain-language summaries of inspection findings	Generation	Supervised (deep learning)	Unstructured	"This is a generation problem using supervised learning (deep learning) on unstructured data."

# Scoping Sentence Examples

Problem	Task Type	Paradigm	Data Type	Scoping Sentence
Classify incoming emails by topic	Classification	Supervised	Unstructured	"This is a classification problem using supervised learning on unstructured data."
Predict length of hospital stay	Regression	Supervised	Structured	"This is a regression problem using supervised learning on structured data."
Identify natural groupings among overdose cases	Clustering	Unsupervised	Structured	"This is a clustering problem using unsupervised learning on structured data."
Flag public comments that contain misinformation	Classification	Supervised	Unstructured	"This is a classification problem using supervised learning on unstructured data."

Why this matters:

- The scoping sentence gives technical staff a starting point.
- It ensures you and the technical team are talking about the same thing.
- It helps identify what data preparation is needed (e.g., labeling, text processing).

# When Scoping Gets Complicated

Not every problem fits neatly into one box. Recognize common complications and know when to consult.

- Some problems involve multiple task types (e.g., classify, then summarize).
- Some problems are borderline (e.g., classification with many categories vs. regression with discretized output).
- Some problems require hybrid approaches (e.g., structured data plus free text).
- It's okay not to have all the answers—the goal is to get close enough to have a productive conversation with technical staff.

## Common Complications

Complication	Example	What to Do
Multiple steps	Flag high-risk patients (classification), then generate outreach letter (generation)	Scope each step separately
Borderline task type	Predict "low / medium / high" risk—is this classification or regression?	Either can work; discuss with technical team
Mixed data	Structured claims data plus free-text case notes	May need separate processing pipelines; flag for technical team
Unclear labels	You want to classify, but outcomes aren't consistently recorded	May need to create labels first (labeling project)
Rare outcome	Classification where only 2% of cases are positive	May need special techniques (class imbalance); flag for technical team

# From Scope to Conversation

The output of scoping is not a technical specification. Usually, it's a starting point for conversation with technical staff or vendors.

- Participants are not expected to design the solution.
- The goal is to communicate clearly: what's the problem, what's the task type, what data is available, and what complications exist.
- Technical staff will take it from there—selecting algorithms, designing features, setting up evaluation.

## What to Bring to the Conversation

Element	What to Communicate
Problem statement	From Topic 1—what decision or action does this inform?
Triage results	From Topic 5—the five questions and your answers
Scoping sentence	Task type, paradigm, data type
Data description	What data exists, where it lives, volume, quality, access
Complications	Anything that doesn't fit neatly—mixed data, unclear labels, rare outcomes
Concerns	Ethical flags, political sensitivities, timeline constraints

# Matching Problems to AI Task Types (Scope) Summary

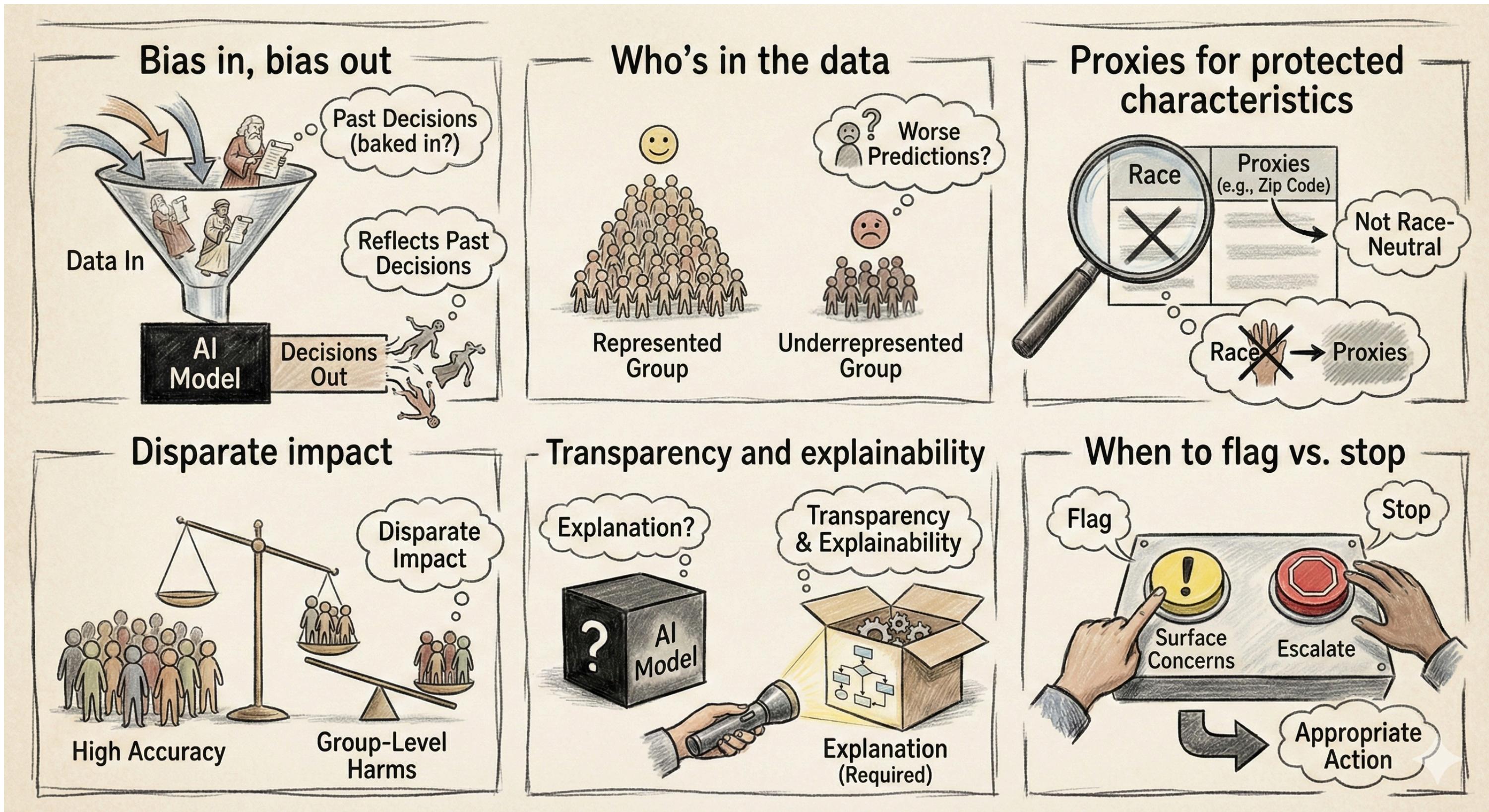
Component	Key Takeaway
What are you trying to produce?	Task type: classification, regression, clustering, or generation
What does your data look like?	Paradigm (supervised/unsupervised) and data type (structured/unstructured)
The scoping sentence	"This is a [task type] problem using [paradigm] on [data type] data."
When scoping gets complicated	Identify complications; you don't have to solve them
From scope to conversation	Frame clearly; technical staff take it from there

# Should I use AI for this problem?

For this exercise, you will continue to work in a group. Have each member of the breakout room share the problem/opportunity/project they wrote down this morning. Select one as the subject for triage and work through the following steps:

- 1. Is the problem suitable for AI/ML?** Consider what parts involve prediction, pattern-finding, or relationship exploration.
- 2. Apply an AI triage framework.** Evaluate data sufficiency, feasibility, impact, risk, operational use, and stakeholder needs.
- 3. Is AI/ML overkill?** Determine whether simpler statistical approaches could answer the question.

# Applying AI: Surfacing Bias and Ethical Flags



# Bias In, Bias Out

If historical data reflects past inequities, the model will learn and perpetuate those patterns.

- This isn't a bug—it's how the technology works. The model finds patterns; it doesn't know if those patterns are fair.
- The question isn't "is there bias?" (there almost always is) but "what kind, how much, and does it matter for this use case?"

## How Bias Gets Into Data

Source of Bias	Example
Biased decisions	Child welfare investigations historically initiated more often in low-income communities
Biased access	Healthcare utilization data reflects who had insurance and access, not who needed care
Biased measurement	Performance reviews that systematically rate certain groups lower
Biased labels	"Fraud" labels based on who got investigated, not who actually committed fraud
Selection bias	Data only includes people who stayed in a program, not those who dropped out

Questions to ask:

- What human decisions produced this data?
- Who was more likely to be in this dataset? Who was less likely?
- If we train on this data, whose patterns are we learning?

# Who's in the Data and Who's Missing?

If certain populations are underrepresented in training data, the model may perform poorly for them. Participants should ask: does this data reflect the population we'll apply this to?

## Common Representation Problems

Problem	Example
Geographic gaps	Model trained on urban clinic data applied to rural populations
Demographic gaps	Model trained mostly on white patients applied to diverse population
Temporal gaps	Model trained on pre-pandemic data applied post-pandemic
Institutional gaps	Model trained on one hospital's data applied across a health system
Outcome gaps	Rare outcomes for certain groups mean less data to learn from

Questions to ask:

- Who is in this dataset? Who is missing or underrepresented?
- Does the training data reflect the population where we'll deploy this?
- Are there groups for whom we have very few examples?
- Have we checked whether the model performs differently across groups?

# Transparency and Explainability Expectations

Some decisions require explanation to the affected person or the public.

- Some models (especially deep learning) are "black boxes"—they produce outputs, but it's hard to say why.
- Some contexts require explainability: legal proceedings, clinical decisions, public accountability.
- Even if not legally required, explainability builds trust and enables oversight.
- There's often a tradeoff: more complex models may be more accurate but less explainable.

## When Explainability Matters

Context	Why Explainability Matters
Benefit denials	Applicants may have a right to know why they were denied
Clinical recommendations	Physicians need to explain decisions to patients
Child welfare	Families and courts may demand to know why a flag was raised
Public-facing decisions	Citizens expect transparency in government decisions
Audits and oversight	Regulators or legislators may require explanation

Questions to ask:

- Will anyone need to explain why this decision was made?
- Can we articulate what factors drove the model's recommendation?
- Is there a legal or regulatory requirement for explainability?
- Would a "black box" decision erode public trust?

# When to Flag vs. When to Stop

Not every concern is a dealbreaker.

- Bias and ethical concerns exist on a spectrum. Some are serious enough to stop a project; others can be noted and managed.

## Escalation Framework

Level	Description	Example	Action
Note and monitor	Minor concern; manageable with design choices	Model uses zip code; could proxy for race	Document; test for disparate impact; monitor
Pause and consult	Significant concern; needs expert input	Training data reflects known historical bias	Consult with equity office or ethics board before proceeding
Stop and escalate	Serious concern; potential for significant harm	Model would be used for high-stakes decisions on vulnerable population with no human review	Escalate to leadership; do not proceed without explicit approval

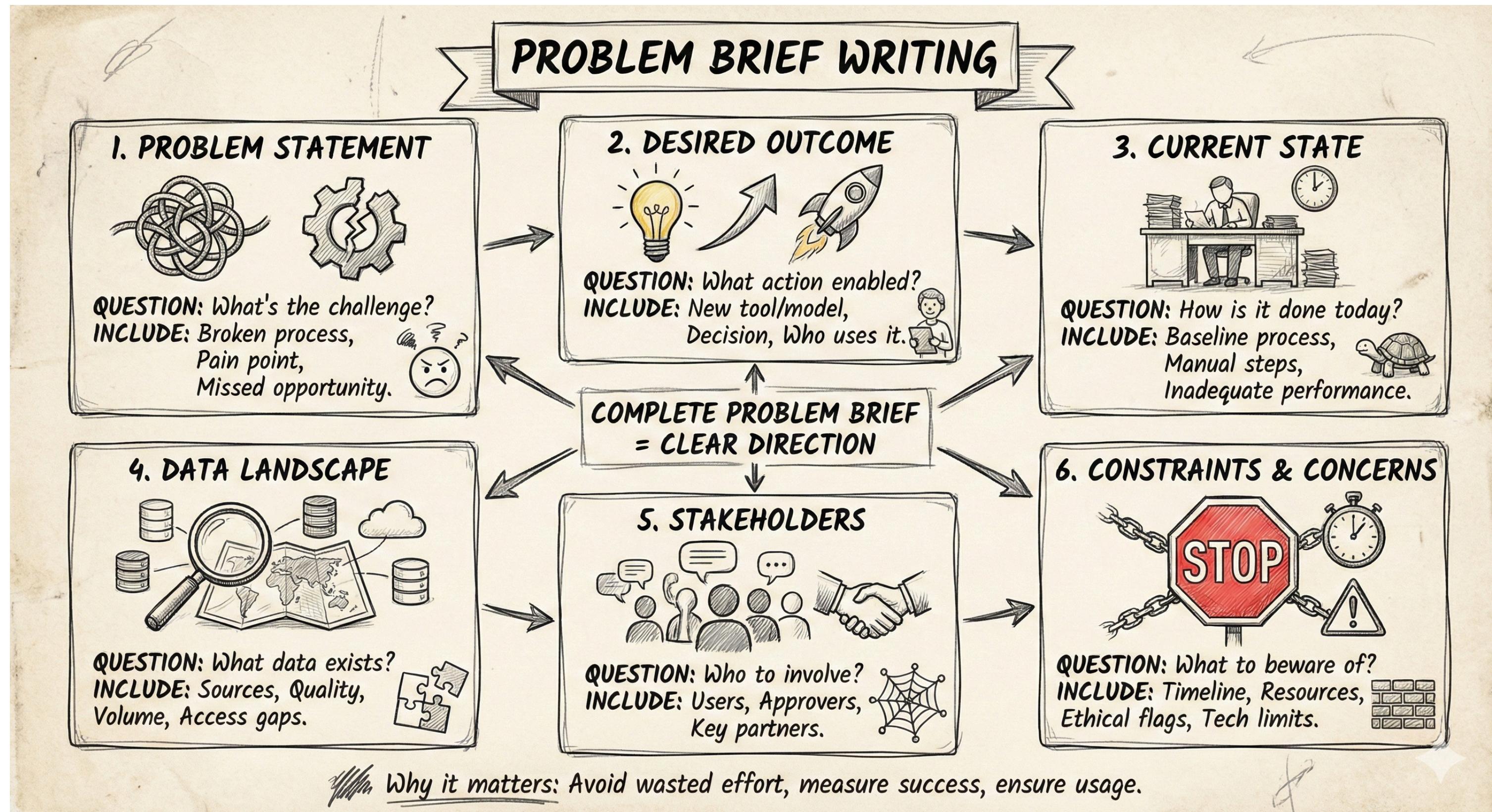
Questions to ask:

- Is this a concern I can address with design choices, or does it need outside input?
- Who in my organization should know about this?
- What's the worst-case scenario if this concern isn't addressed?
- Is there a formal review process this should go through?

# Surfacing Bias and Ethical Flags Summary

Component	Key Takeaway
Bias in, bias out	Historical data reflects past decisions—ask what biases are baked in
Who's in the data	Underrepresented groups may get worse predictions
Proxies for protected characteristics	Removing race doesn't make a model race-neutral
Disparate impact	Overall accuracy can hide group-level harms
Transparency and explainability	Some decisions require explanation
When to flag vs. stop	Surface concerns; escalate appropriately

# Applying AI: Writing a Problem Brief



# What a Problem Brief Is (and Isn't)

A problem brief is a clear, concise articulation of the problem, the desired outcome, and the context—written for someone who doesn't know the program area. Articulating a problem clearly is the bridge between "we should use AI" and actually getting something built.

## What a Problem Brief Does and Doesn't Do

### It Does

- Articulate the problem clearly
- Define what success looks like
- Describe the current state
- Identify stakeholders and constraints
- Flag concerns and unknowns

### It Doesn't

- Prescribe a technical solution
- Specify algorithms or tools
- Include project timelines or budgets
- Assign tasks or responsibilities
- Resolve all questions

# Core Elements of a Problem Brief

A complete problem brief has six elements. Each one answers a question that technical staff or decision-makers will need answered.

## The Six Elements

Element	Question It Answers	What to Include
Problem statement	What's the challenge?	What's not working, what's the pain point, or what opportunity exists. Be specific.
Desired outcome	What decision or action would this enable?	What the model would produce, who would use it, and what they would do with it.
Current state	How is this handled today?	The baseline: current process, performance, volume, and why it's inadequate.
Data landscape	What data exists?	Where the data lives, what it contains, volume, quality, access, and gaps.
Stakeholders	Who needs to be involved?	Who would use the output, who needs to approve, who needs to be consulted.
Constraints and concerns	What should we be aware of?	Timeline, resources, ethical flags, political sensitivities, technical limitations.

# Writing a Good Problem Statement

The problem statement is the foundation. It should be specific, outcome-focused, and solution-agnostic.

- A good problem statement describes the pain point without embedding a solution.
- It's specific enough that someone outside the program area can understand it.
- It focuses on outcomes (what's not working) rather than activities (what you want to do).

## **Problem statement formula:**

[Who] is experiencing [what problem], which results in [what consequence]. Currently, [how it's handled], but [why that's inadequate].

# Weak vs. Strong Problem Statements

## Weak Problem Statement

"We need a chatbot for our call center."

"We want to use AI to improve maternal health."

"We should predict fraud."

## Strong Problem Statement

"Constituents calling our benefits hotline wait an average of 18 minutes to reach a representative. 40% of calls are simple status checks that don't require human assistance. This leads to frustration and ties up staff who could handle complex cases."

"Women in rural counties are 2.3x more likely to experience severe maternal morbidity than those in urban areas. We lack early warning indicators to identify high-risk pregnancies before complications occur. Currently, risk assessment is done at the first prenatal visit using a paper checklist."

"Our program integrity unit reviews 200 provider claims per month for potential fraud, selected based on manual referrals. We estimate we're missing 80% of fraudulent claims. Each undetected fraud case costs an average of \$45,000."

# Describing the Data Landscape

Data landscape includes: what data exists, where it lives, what format it's in, how much there is, what's known about quality, and what approvals are needed to access it. It's okay to have gaps but they need to be identified.

## What to Include in Data Landscape

Question	What to Document
What data exists?	Tables, fields, systems, sources
Where does it live?	System names, databases, file locations
What format is it in?	Structured (database, spreadsheet) or unstructured (text, PDFs)
How much is there?	Number of records, time range covered
What's the quality?	Completeness, consistency, known issues
Does it include the outcome?	Is the thing you're trying to predict recorded?
What approvals are needed?	Data use agreements, IRB, legal review
What's missing or unknown?	Gaps, uncertainties, things you need to find out

## Example data landscape description:

"Data sources include Medicaid claims (5 years, 2M+ records, stored in our data warehouse), vital records (birth certificates linked by patient ID), and WIC enrollment data (separate system, requires data sharing agreement). Claims data is generally complete but has known issues with diagnosis coding consistency. We do not currently have social determinants of health data. Outcome variable (hospital readmission within 30 days) is recorded in claims. Access requires DUA with Medicaid agency (in progress)."

# Common Mistakes in Problem Briefs

Certain mistakes are predictable and avoidable. Recognizing them helps participants write better briefs and critique others' briefs.

Mistake	Example	Why It's a Problem
Embedding a solution	"We need a machine learning model that..."	Skips problem definition; may be solving the wrong thing
Being too vague	"Improve outcomes for clients"	Can't evaluate success; technical staff don't know what to build
Being too broad	"Use AI to transform our department"	No specific problem; no way to scope or prioritize
Overcomplicating	"A system that predicts X, flags Y, generates Z, and integrates with A, B, C"	Too many goals; likely to fail or take forever
Ignoring the "so what"	"Predict which clients will miss appointments" (but no intervention)	Prediction without action is useless
Skipping constraints	No mention of timeline, resources, or sensitivities	Sets up for surprise blockers later
Assuming data exists	"We'll use our data to..." without verification	May discover data doesn't exist or isn't accessible

## How to avoid these mistakes:

- Write the problem statement before mentioning any technology.
- Ask: "If someone outside my agency read this, would they understand the problem?"
- Ask: "What would we do differently if we had the answer?"
- Ask: "What could go wrong, and have I mentioned it?"

# The Review Process: Giving and Receiving Feedback

Problem briefs improve with feedback.

- Writing is iterative. A first draft is rarely final.
- Peer review catches blind spots—things that are obvious to you but not to others.
- A good critique focuses on clarity, completeness, and feasibility—not style preferences.

## Review Rubric

Criterion	Questions to Ask
Problem is specific and outcome-focused	Can I explain the problem to someone outside this program? Is there a clear pain point?
Success is defined	Do I know what "working" looks like? Is it measurable?
Current state is described	Do I understand how this is handled today and why that's inadequate?
Data is addressed	Do I know what data exists, where it is, and what's uncertain?
Stakeholders are identified	Do I know who will use the output and who needs to be consulted?
Concerns are flagged	Are constraints, risks, and sensitivities mentioned?
No solution is embedded	Does the brief describe a problem, or does it prescribe a solution?

# Giving and Receiving Feedback

Giving feedback:

- Start with what's clear and strong.
- Ask questions rather than making assertions: "I wasn't sure what you meant by..." or "What would happen if...?"
- Focus on gaps, not style: "I didn't see anything about data access—is that a concern?"
- Be specific: "The problem statement is clear, but the success criteria are vague" is more useful than "needs work."

Receiving feedback:

- Assume good intent—the goal is a better brief.
- Listen for confusion—if the reviewer is confused, others will be too.
- Ask clarifying questions: "What would make this clearer?"
- Revise based on patterns—if multiple reviewers flag the same issue, it's real.

# Writing a Problem Brief Summary

Component	Key Takeaway
What a brief is	Frames the problem; doesn't prescribe the solution
Core elements	Six elements: problem, outcome, current state, data, stakeholders, constraints
Problem statement	Specific, outcome-focused, solution-agnostic
Data landscape	Name what exists, where it is, and what's unknown
Common mistakes	Embedding solutions, being vague, ignoring "so what"
Review process	Use the rubric; give and receive feedback constructively

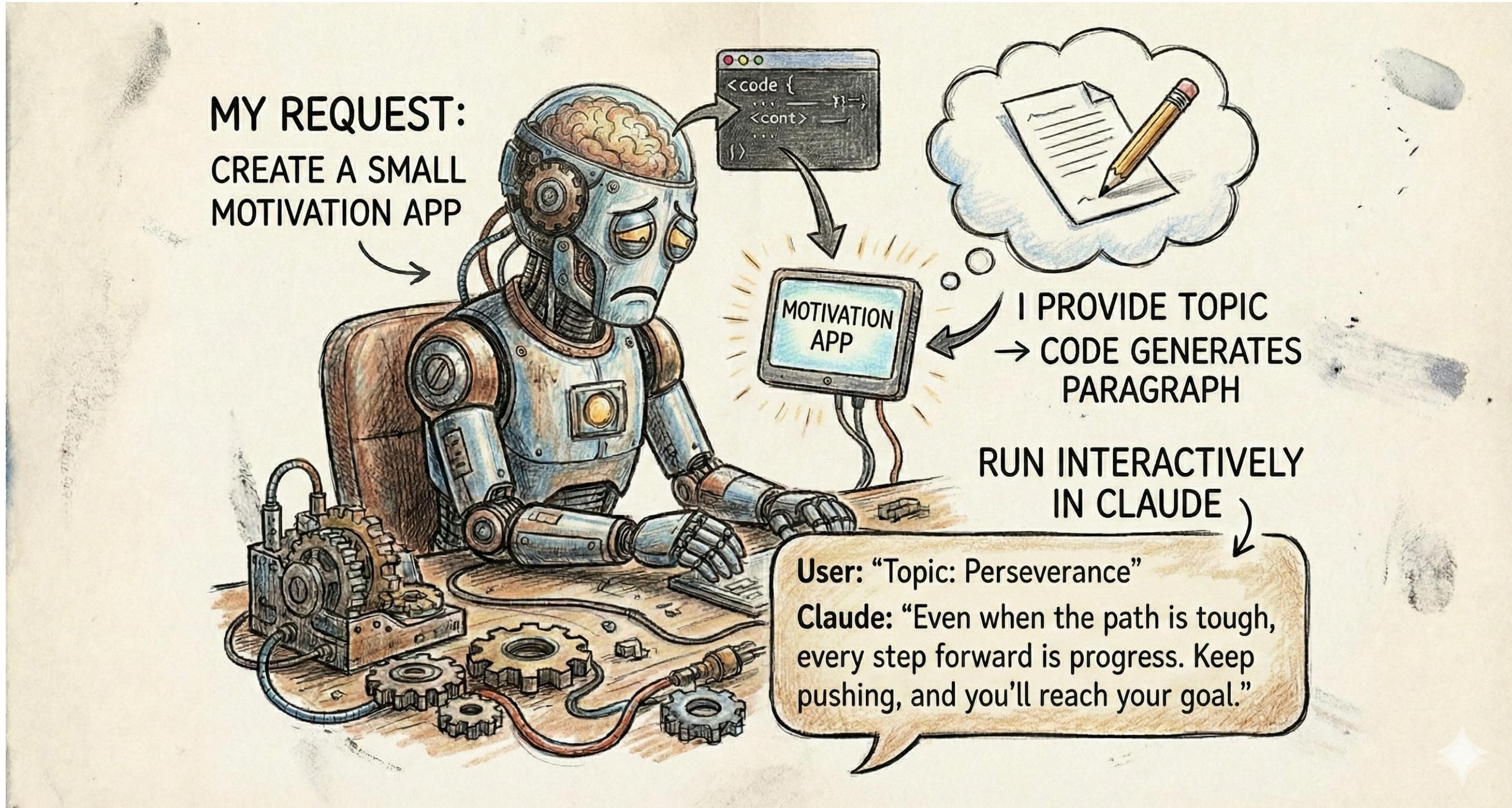
# Using an LLM with R

## Purpose

- More practice getting Claude to do what we want.
- **Using an LLM in our code**
- Working with API keys
- We will create 2 reports. Motivation (in console) and Motivation-shiny (nicer interface).
  - Both reports use AI for creating the R code.
  - Both use AI as part of the code.

# Motivation app

I want to create a small motivation app. I will provide a topic and the code will generate a motivational paragraph. Let's run it interactively in Claude.



# Motivation Generator

✨ Motivation Generator

Enter any topic and get inspired!

Enter a topic (e.g., fitness, career, creativity)

Inspire Me!

Your motivation awaits...



# A Motivation Message

Winter darkness

Inspire Me!

 Winter darkness is not your enemy—it's your forge, the sacred crucible where resilience is hammered into unbreakable steel. In these shortened days and lengthened nights, you discover that your inner light burns brightest when external illumination fades, proving that true strength doesn't depend on perfect conditions but thrives in spite of them. The bare trees and frost-covered ground aren't symbols of death but of powerful patience, teaching you that sometimes the greatest growth happens underground, invisible and unstoppable. Embrace this season of introspection as your competitive advantage, using these quiet hours to plan, prepare, and kindle the fire that will explode into brilliance when spring arrives. Let winter's darkness become the canvas upon which you paint your most audacious dreams, knowing that those who harness the power of the dormant season emerge as unstoppable forces when the world awakens.

# Generating R code

**Prompt: If we want to convert this to so I can run it in posit.cloud using R, what has to change. I don't need an interface, just the console. Don't generate code yet. Just tell me what is different.**

1. API Key Requirement: The React artifact handles authentication automatically behind the scenes

In R, you'll need your own Anthropic API key, typically stored as an environment variable or passed directly

2. HTTP Requests: JavaScript uses `fetch()`

R would use the `httr` or `httr2` package to make POST requests

3. JSON Handling: JavaScript has native JSON support

R would use the `jsonlite` package to build the request body and parse the response

4. User Input: Instead of a web form, you'd use `readline()` to prompt for topics in the console

5. No UI Layer: All the React components, state management, styling, and error display go away

Just a simple script that prints output to the console

6. Structure: Instead of a component with hooks and event handlers, it becomes a straightforward linear script or a simple function

Could add a while loop if you want to keep prompting for multiple topics

# The API Key

API keys 1

API keys are owned by workspaces and remain active even after the creator is removed

KEY	WORKSPACE	CREATED BY	CREATED AT ^	LAST USED AT	COST ⓘ
motivation-key sk-ant-api03-lxS...dgAA	Default ⓘ	Jim jcodygroup@gmail.com	Dec 6, 2025	Dec 10, 2025	\$0.20 ...

**Credit balance**

Your credit balance will be consumed with API, Claude Code and Workbench usage. You can buy credits directly or set up auto-reload thresholds.

**\$4.79**  
Remaining Balance

Charged to  
American Express •••• 8019 ⓘ [Buy credits](#)

(X) Auto reload is disabled. Enable auto reload to avoid API interruptions when credits are fully spent. [Edit](#)

## Code: Get the API key

```
# Set your API key (or use Sys.setenv(ANTHROPIC_API_KEY = "your-key-here"))  
# You can also set this in your .Renviron file  
  
api_key <- Sys.getenv("ANTHROPIC_API_KEY")
```

# Code: Build the prompt

```
# Build the request body
body <- list(
  model = "claude-sonnet-4-20250514",
  max_tokens = 1000,
  messages = list(
    list(
      role = "user",
      content = paste0(
        'Write a single inspiring and motivational paragraph about: "', topic, '". ',
        'Make it uplifting, energizing, and actionable. Use vivid language that sparks determination. ',
        'Keep it to one well-crafted paragraph (4-6 sentences). ',
        'Do not include any preamble or explanation--just the motivational paragraph itself.'))
```

more code ...

## Code: Make the API request

```
# Make the API request

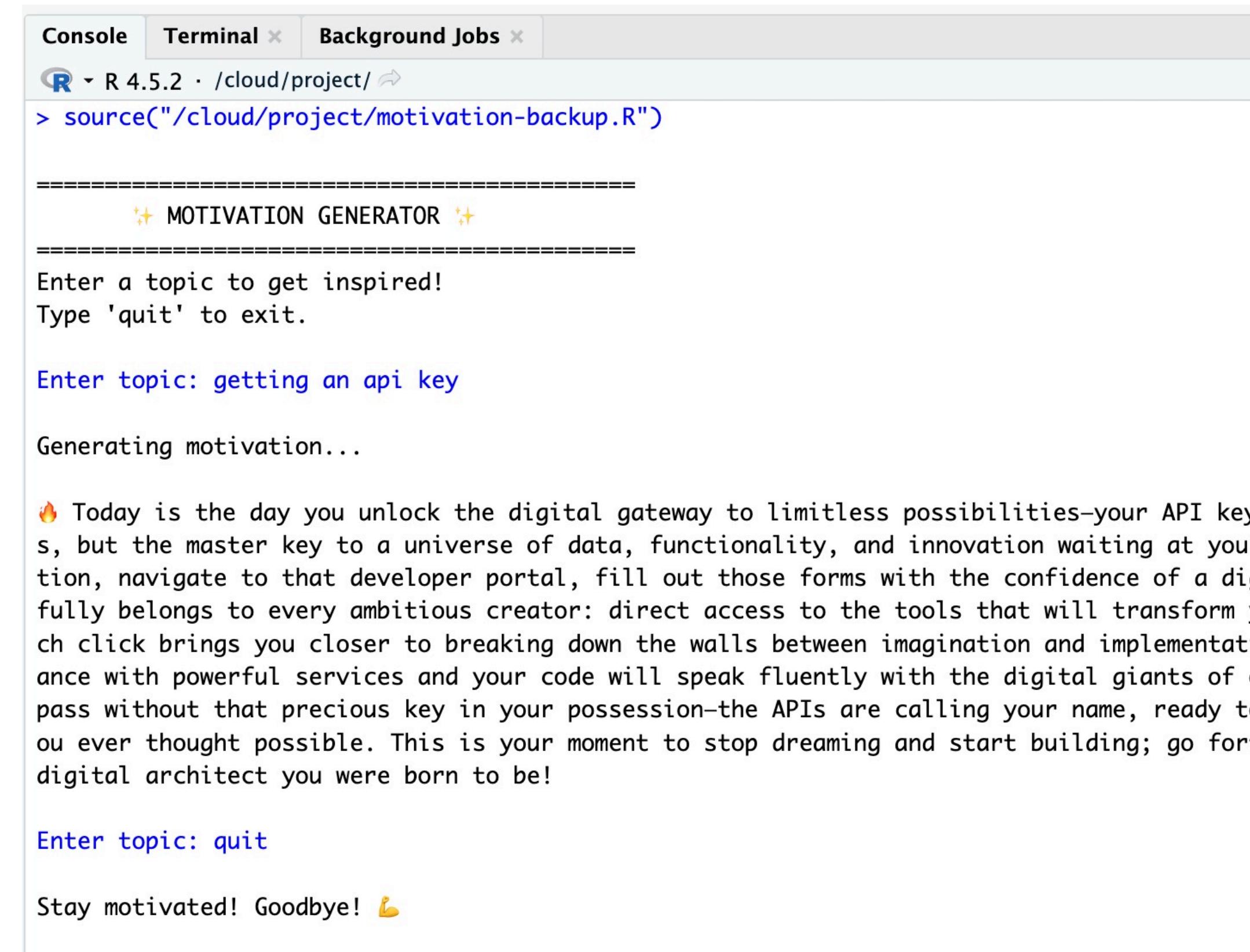
response <- POST(
  url = "https://api.anthropic.com/v1/messages",
  add_headers(
    `x-api-key` = api_key,
    `anthropic-version` = "2023-06-01",
    `content-type` = "application/json"
  ),
  body = toJSON(body, auto_unbox = TRUE),
  encode = "raw"
)
```

# Code: The interface

```
cat("\n")
cat("=====\\n")
cat("      █ MOTIVATION GENERATOR █ \\n")
cat("=====\\n")
cat("Enter a topic to get inspired!\\n")
cat("Type 'quit' to exit.\\n\\n")
```

more code . . .

# Running the code



The screenshot shows an RStudio interface with three tabs at the top: 'Console' (selected), 'Terminal x', and 'Background Jobs x'. The 'Terminal' tab shows the command: > source("/cloud/project/motivation-backup.R"). The output window displays the following:

```
R 4.5.2 · /cloud/project/ 
> source("/cloud/project/motivation-backup.R")

=====
    ✨ MOTIVATION GENERATOR ✨
=====

Enter a topic to get inspired!
Type 'quit' to exit.

Enter topic: getting an api key

Generating motivation...

🔥 Today is the day you unlock the digital gateway to limitless possibilities—your API keys, but the master key to a universe of data, functionality, and innovation waiting at your fingertips. With this key in hand, anything is possible. As you explore the developer portal, navigate to that developer portal, fill out those forms with the confidence of a digital architect. This key belongs to every ambitious creator: direct access to the tools that will transform your dreams into reality. Each click brings you closer to breaking down the walls between imagination and implementation. With powerful services at your disposal and your code will speak fluently with the digital giants of tomorrow. APIs are the language of the future, and with this key in your possession—the APIs are calling your name, ready to help you achieve what you ever thought possible. This is your moment to stop dreaming and start building; go forth and be the digital architect you were born to be!

Enter topic: quit

Stay motivated! Goodbye! 💪
```

# Using an LLM in our diabetes analysis

## Instructions:

1. Create a new project named **Using an LLM**. All chats for this walkthrough will be in this project.
2. Add an instruction to the project. (Adjust as you see fit)
  - a. I am a public health professional with little technical and programming experience.
3. **Ask Claude to:**
  - a. Read the data from this url and tell me how many rows are available. Do not generate any other code until I tell you it is ok to do so.  
[https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic\\_data101k.csv](https://raw.githubusercontent.com/jimcody2014/BA/refs/heads/main/diabetic_data101k.csv)
  - b. I know I can run EDA and some other basic analytics on this data. Give me 3-4 suggestions on how I can use an LLM to generate insights.
  - c. If Automated narrative reporting (or something similar) does not show up, ask "Let's do some automated narrative reporting. Tell me what you can do to enhance reporting."
  - d. Ask Claude to describe some aspect of this work in more detail. Make sure you know what Claude is going to do.
  - e. Proceed with the project.
    - i. **This will take a few minutes.**
  - f. Convert this code to R. **Do not try to run the R code.** Prepare a file for me and I will run it locally. Make sure it uses the url. Make sure it includes an LLM.
  - g. Look at the code to make sure it is usig the url to get the data.
4. Copy and paste the code into posit.cloud. Name the file predict-readmit. Run the code.