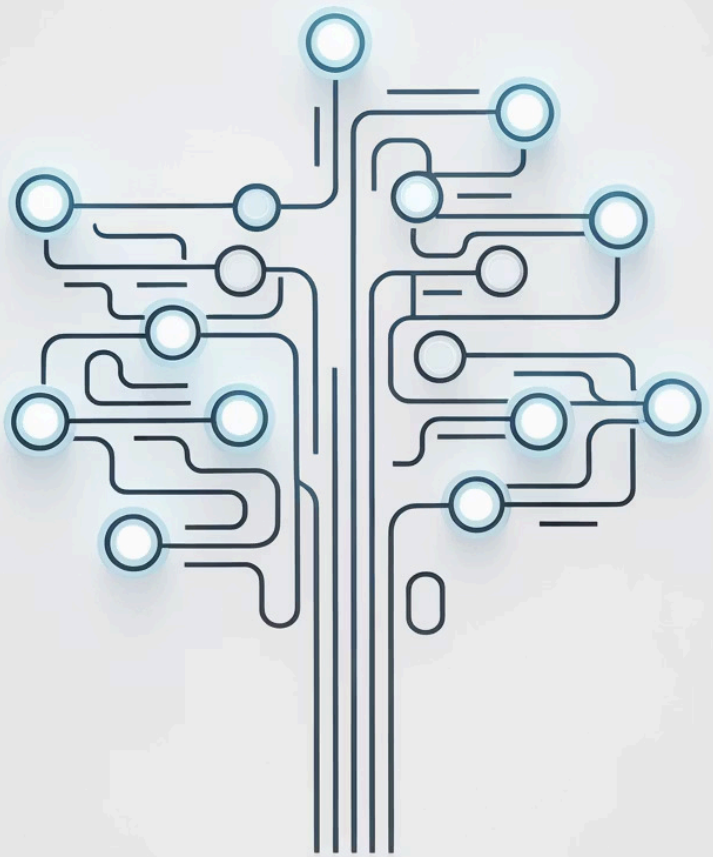
# Module 1: Introduction to Artificial Intelligence

To equip the team with a foundational understanding of AI, its project lifecycle, modern tools, and to empower them to identify and scope AI opportunities.

**Aron Kondoro**



# Why Should the WCF Tech Team Care About AI?

AI is the next evolution of software, moving from systems that follow rules to systems that learn from data. It's about empowering our experts, not replacing them.

### Speed

Reduce claim processing time from weeks to days, accelerating service delivery.

### Efficiency

Automate repetitive data entry tasks, freeing up valuable human resources.

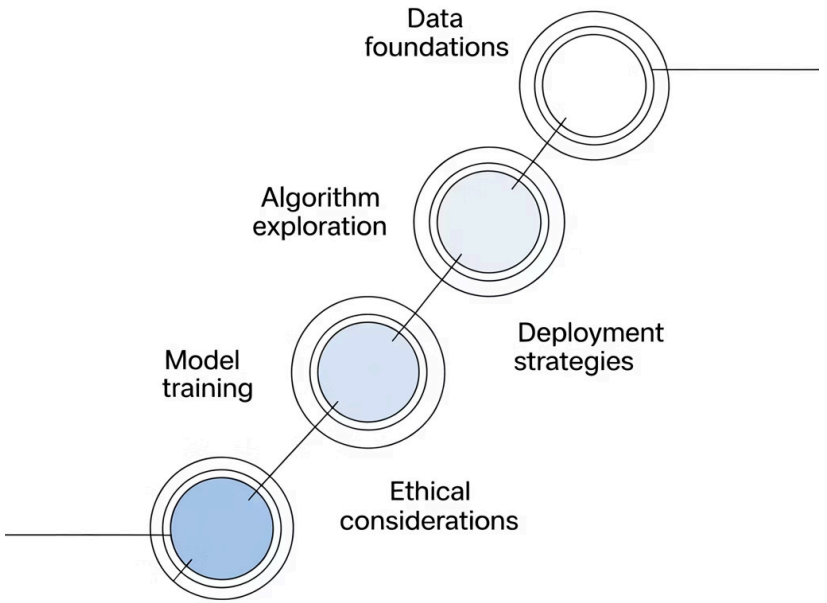
### Fairness

Ensure consistent and equitable decisions across all claims, minimizing bias.

### Sustainability

Proactively identify fraud and waste to protect the Fund's long-term viability.

# Today's Roadmap




01

## Foundations

What is AI, really?

02

## The Blueprint

The AI Project Lifecycle

03

## The WCF Opportunity

AI Use Cases for Us

04

## The Toolbox

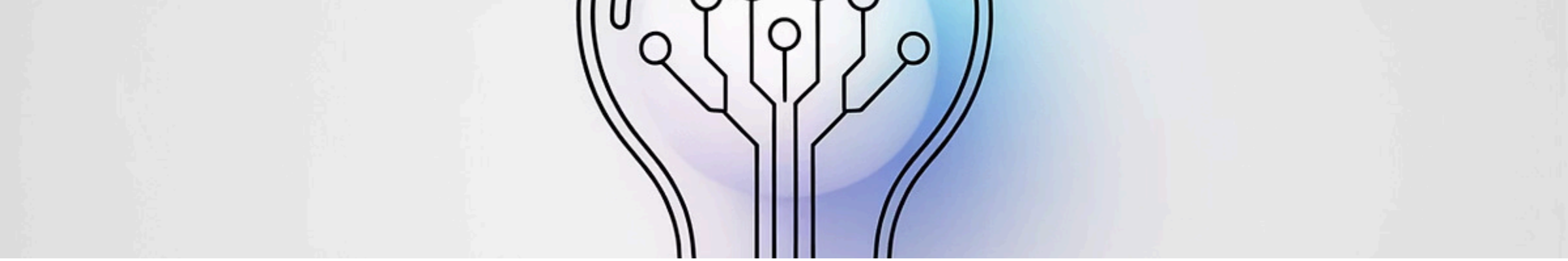
Modern Tools & Platforms

05

## Putting It All Together

Strategy & Ethics

We'll start with the 'what,' then move to the 'how,' the 'where,' and finally, the 'what's next.'



# Module 1: Foundations

### 1 Defining AI/Machine Learning

Understanding what ML is, how it differs from traditional programming, and why it matters for our work at WCF

### 2 Use Cases & Applications

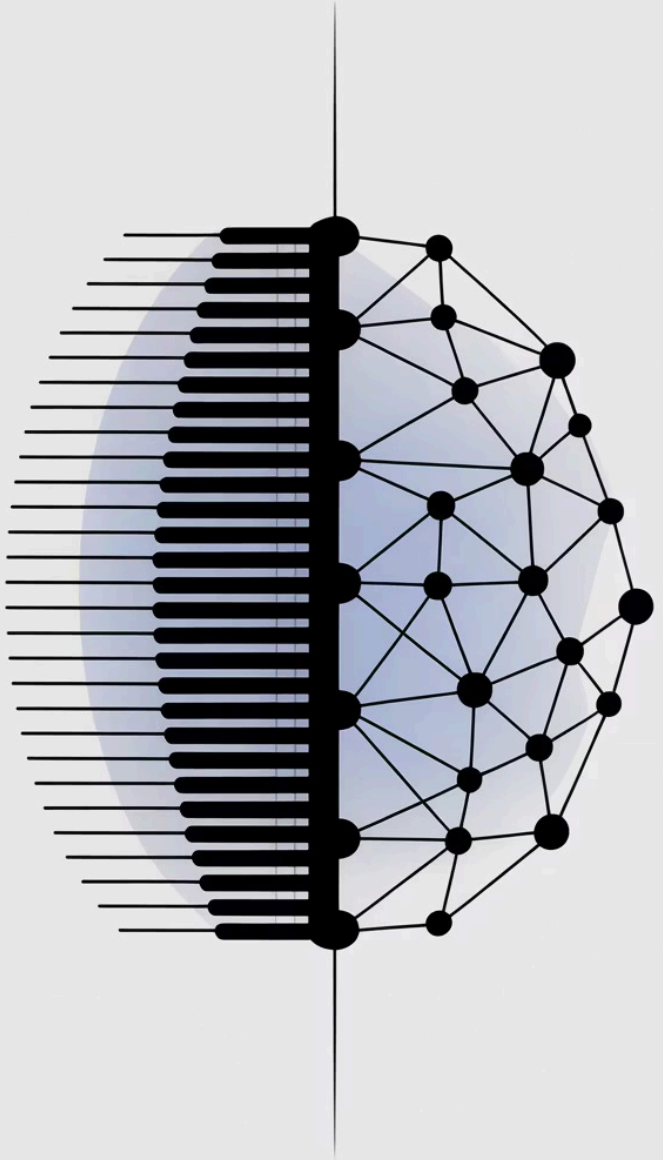
Exploring when to apply ML to solve complex problems, with specific WCF examples

### 3 ML System Categories

Mapping the landscape of different ML approaches and understanding their unique strengths

### 4 Challenges & Considerations

Identifying potential pitfalls and key success factors in ML implementation



# The Fundamental Shift: Rules vs. Patterns

## Traditional Software vs. AI-Powered Software

### Traditional Programming

We write **explicit rules** for the computer to follow:

```
IF Claim_Amount > 5,000,000 TZS
AND Industry = 'Mining'
THEN Flag_For_Review()
```

- System is **static**
- Good for **simple, defined** problems
- Limited by programmer's foresight

### Machine Learning

We provide **data & answers** for the computer to learn from:

*The machine learns the patterns* that lead to complex claims by analyzing thousands of past examples

- System **learns and adapts**
- Good for **complex, nuanced** problems
- Can discover patterns humans miss

# From Automation to Intelligence

### Manual Process

Human workers performing every step of a task

Example: Manually reviewing every claim document

### Scripted Automation

Software following fixed rules to handle routine tasks

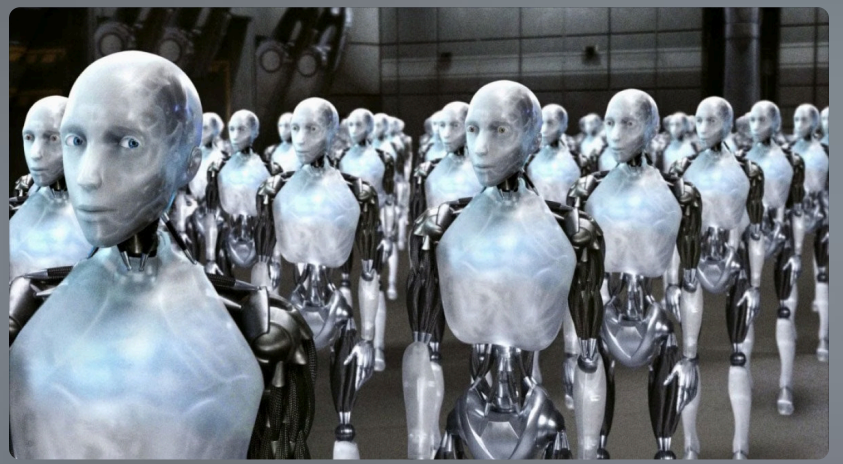
Example: Auto-routing claims based on injury codes

### Intelligent Automation (AI)

Systems that learn, adapt, and make judgment calls

Example: Predicting claim costs and identifying potential fraud patterns

We've already automated many things with scripts and software. AI is the next step, allowing us to automate tasks that require judgment and pattern recognition.



# What Hollywood Thinks AI Is...

When we hear "AI," our minds often go here. This is General AI - a machine with human-like consciousness.

> **General AI** (or Artificial General Intelligence) refers to a hypothetical machine that can understand, learn, and apply knowledge across a wide range of tasks at a human level or beyond.

It's fascinating science fiction, but it's not what we're building today at WCF.

# What Business & Tech AI *Really* Is...


## Narrow AI

Our focus is on **Narrow AI** - systems designed to do one specific task exceptionally well:

- Fraud detection
- Language translation
- Medical image analysis
- Product recommendations

This is where the real value is for the WCF.

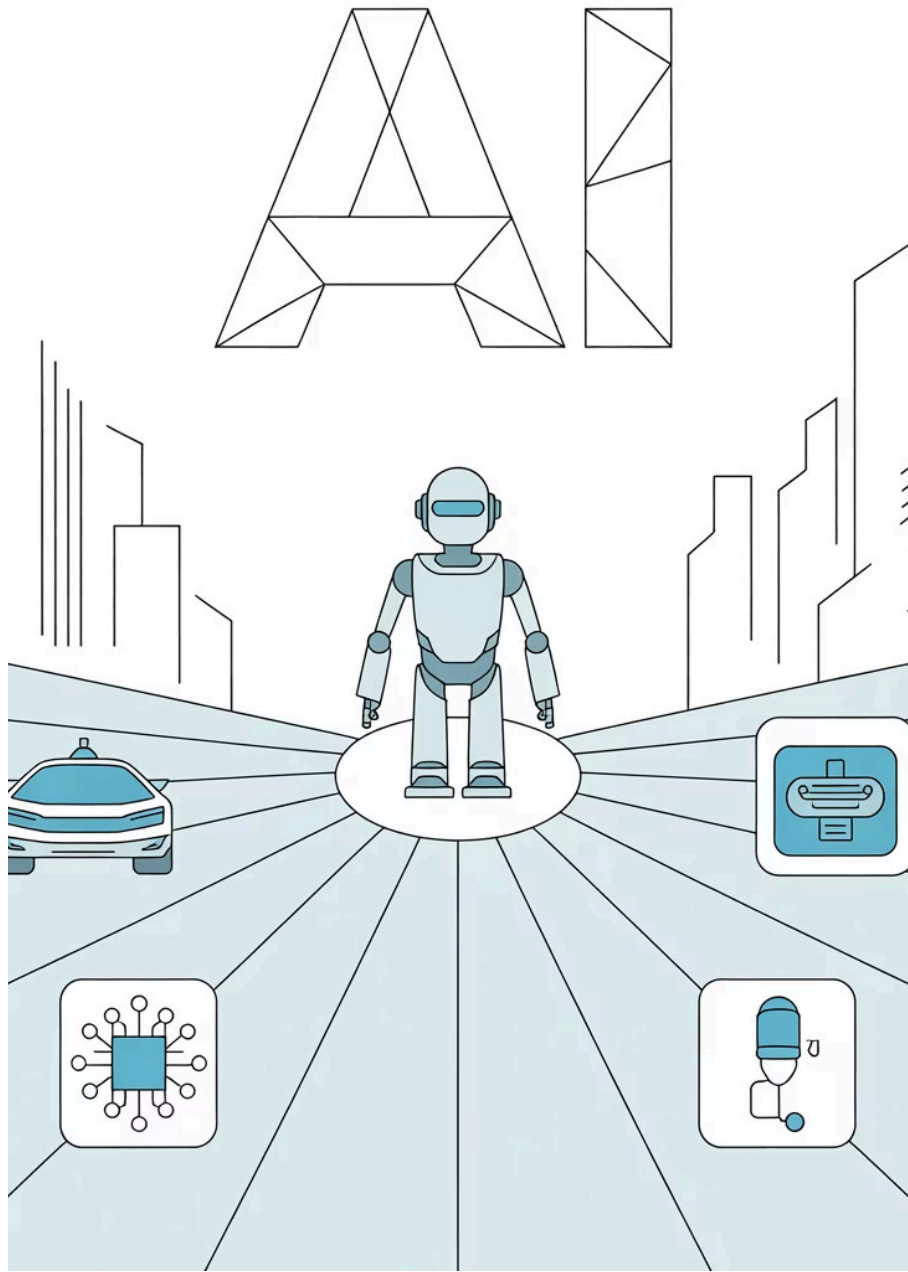
# Artificial Intelligence (AI)

**The broad science of making machines perform tasks that typically require human intelligence.**

## "The effort to automate intellectual tasks normally performed by humans."

AI encompasses all technologies that enable computers to mimic or perform functions that would normally require human intelligence, including:

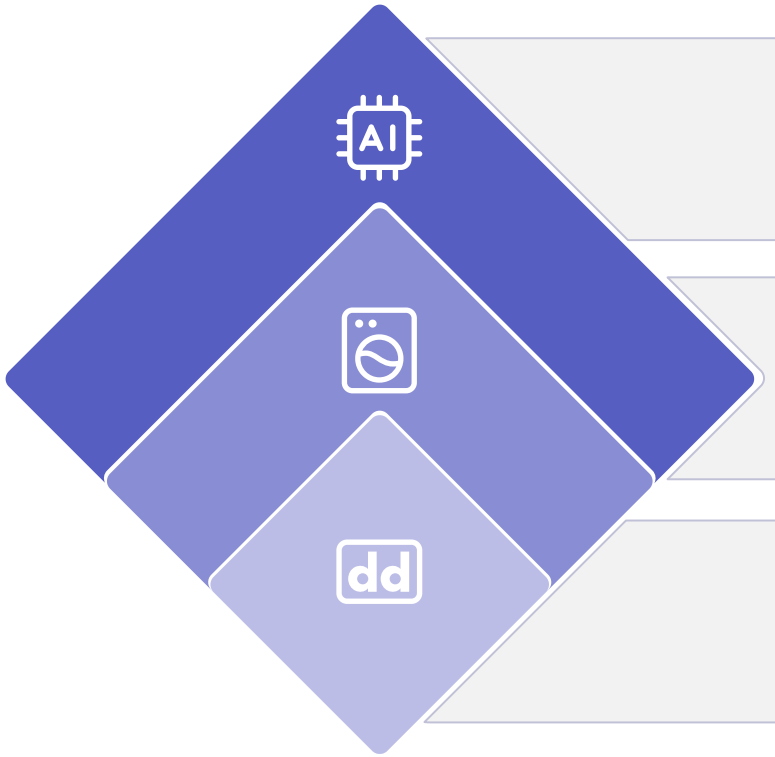
- Reasoning and problem solving
- Understanding language
- Learning from experience
- Perception of visual and audio input



# The AI Family Tree

## Unpacking the Buzzwords: AI, Machine Learning, and Deep Learning

These terms are often used interchangeably, but they represent distinct concepts with specific applications.




### AI

Broad field of intelligent systems

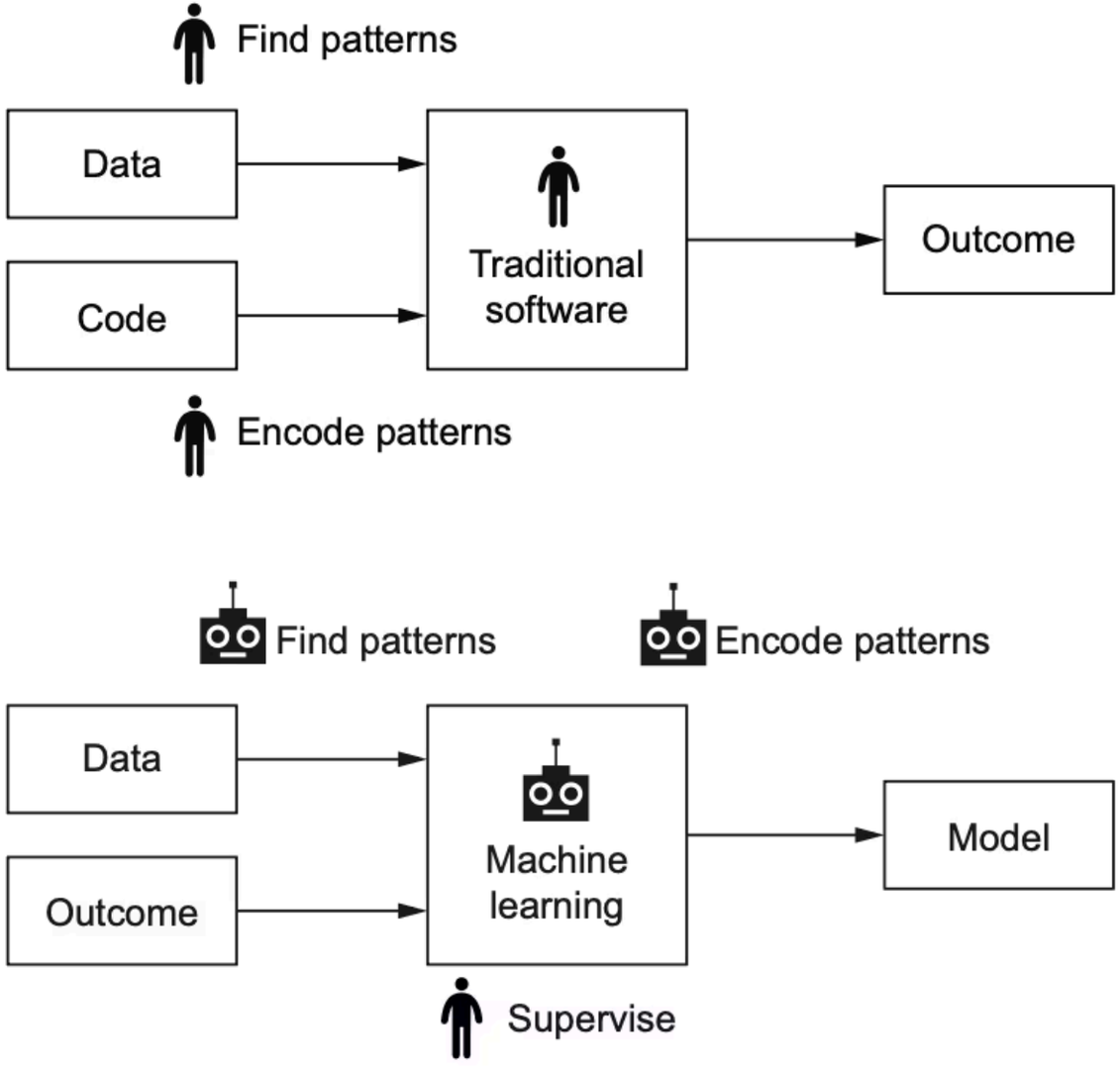
### Machine Learning

Algorithms that learn from data

### Deep Learning

Neural networks with many layers

# Machine Learning (ML)

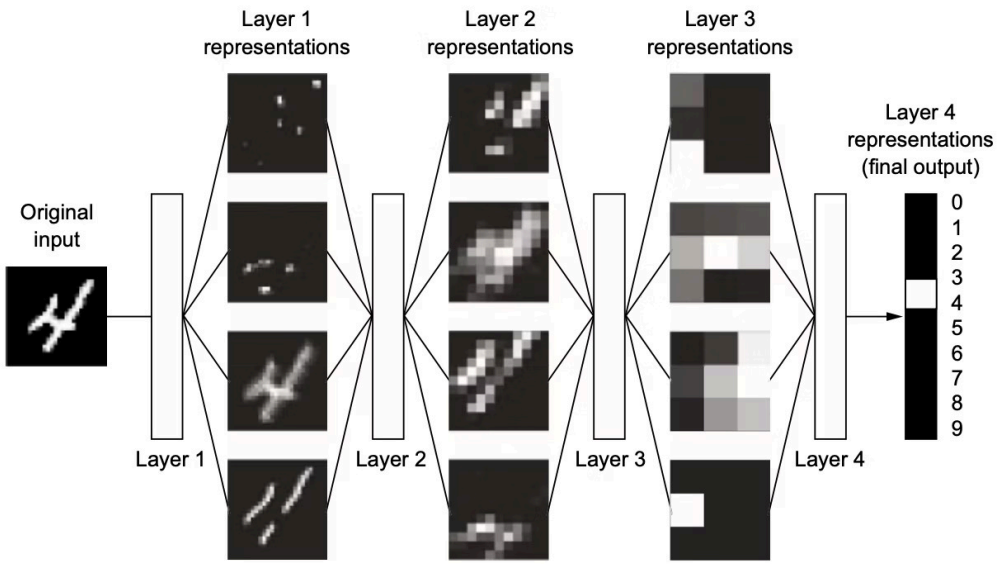



A subset of AI where systems **learn from data** to identify patterns and make decisions *without being explicitly programmed*.

## Key Characteristics:

- Improves with experience
- Finds patterns humans might miss
- Makes predictions based on historical data
- Adapts to new information

# Deep Learning (DL)




A powerful subfield of ML that uses "deep neural networks." It's ideal for very complex patterns in massive datasets.

### The "Deep" in Deep Learning

The depth refers to the number of processing layers in the neural network, not the depth of understanding. More layers enable more complex pattern recognition.

## Especially good at:

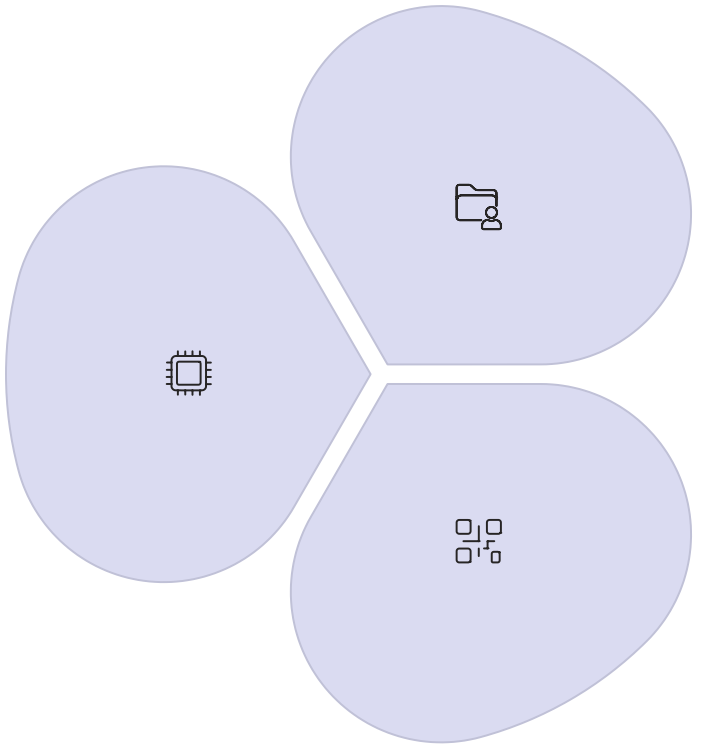
- Image recognition
- Natural language processing
- Speech recognition
- Complex pattern identification

Deep Learning powers the most advanced AI systems we use today.

# Why Deep Learning? Why Now?

### Hardware

GPUs provide massive parallel processing power



### Data

Internet created unprecedented access to training data

### Algorithms

Better techniques for training deep networks

> "The real bottlenecks throughout the 1990s and 2000s were data and hardware... It's an engineering science."

These three forces converged to create the conditions for today's AI revolution, transforming theoretical concepts into practical tools.

# What Deep Learning Has Achieved So Far

Although deep learning is a fairly old subfield of machine learning, it only rose to prominence in the early 2010s. In the few years since, it has produced remarkable results, revolutionizing areas that were once thought to be exclusively human domains.

### Perceptual Tasks

- Near-human image classification
- Near-human speech transcription
- Near-human handwriting recognition

### Natural Language Processing

- Dramatically improved machine translation
- Enhanced text-to-speech conversion
- Powering digital assistants (Google Assistant, Alexa)
- Ability to answer natural language questions

### Advanced Automation

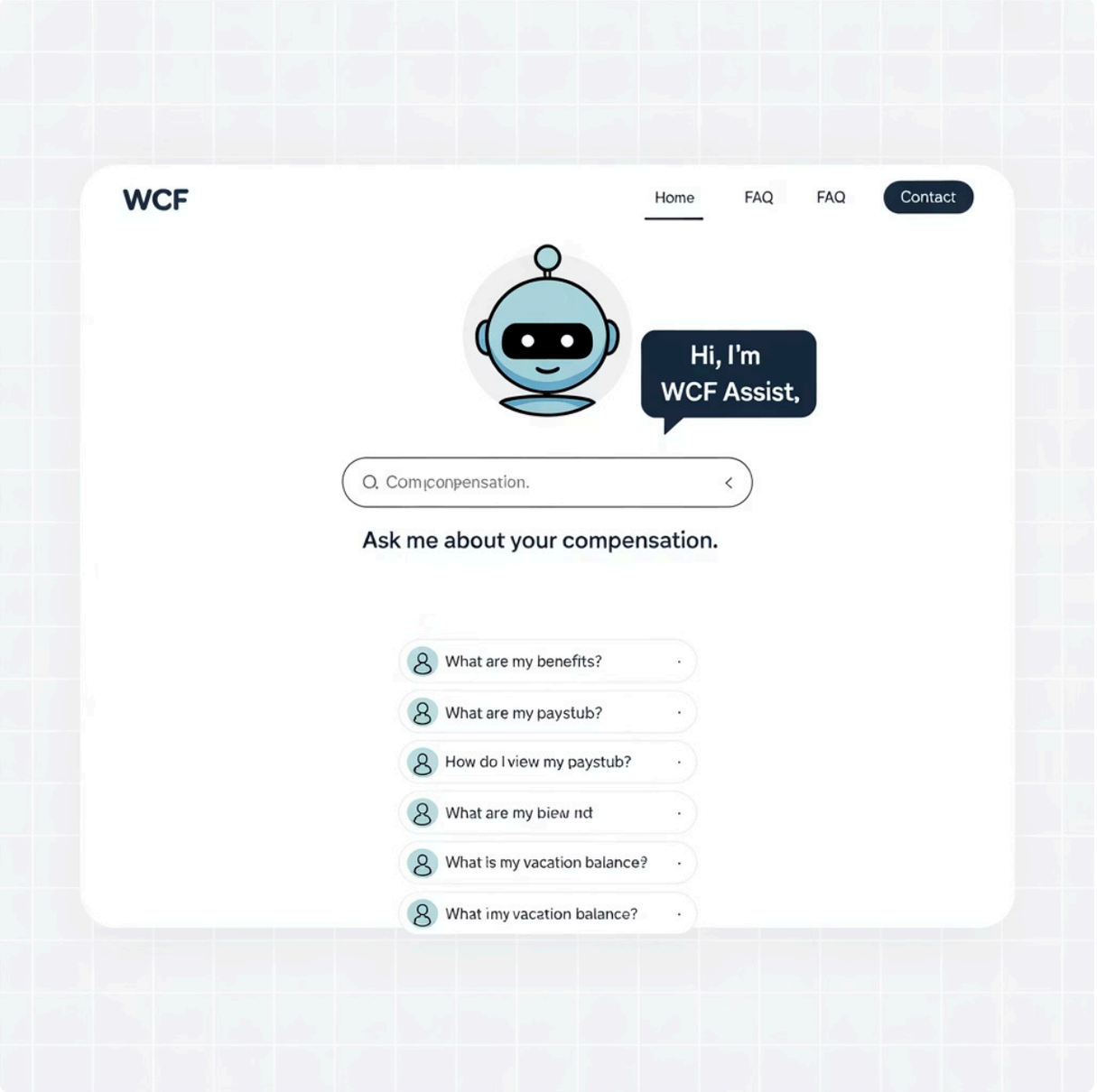
- Near-human autonomous driving
- Improved ad targeting for major platforms
- More relevant web search results

### Complex Reasoning

- Superhuman Go playing (e.g., AlphaGo)
- Identifying subtle patterns in massive datasets

These breakthroughs showcase deep learning's ability to tackle problems involving skills that seem natural and intuitive to humans, but have long been elusive for machines.

# Conversational AI




## 24/7 Claim Support

AI-powered chatbots can:

- Answer common questions instantly
- Guide claimants through the filing process
- Provide status updates on existing claims
- Escalate complex cases to human agents

**Value:** Improved service, reduced call volume, consistent information

# Predictive Analytics

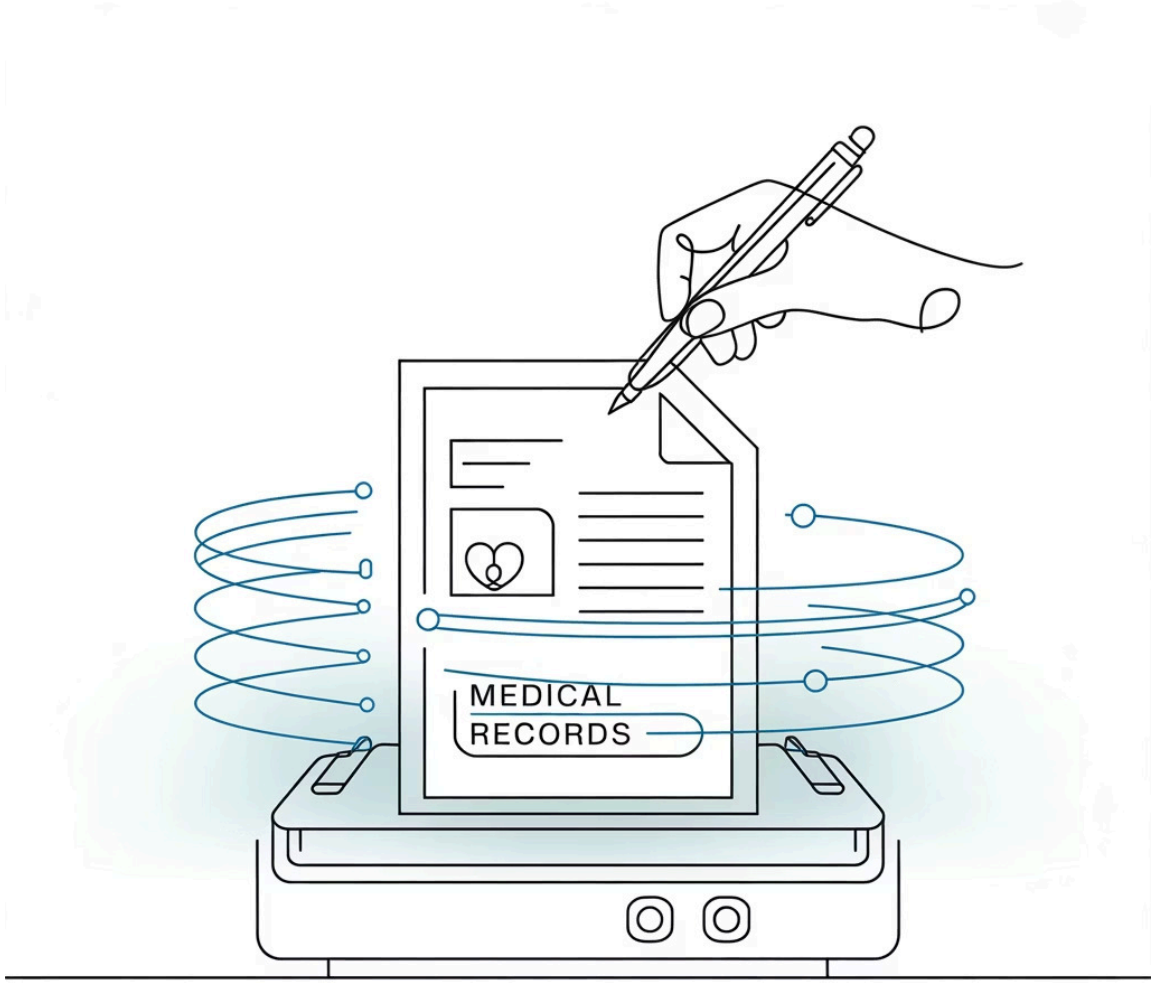
## Claim Cost Prediction

Machine Learning can analyze a new claim and predict:

- Likely final cost
- Expected duration
- Probability of complications
- Potential for litigation

**Value:** Better resource allocation, proactive case management, improved financial forecasting

# Document Intelligence




## Automated Form Processing

Deep Learning can:

- Read handwritten forms
- Extract key information
- Categorize documents
- Flag missing information

**Value:** Faster processing, reduced data entry errors, searchable digital records

# Quick Poll

Let's check our understanding:

## Is a system that reads a license plate from a security camera an example of...?

### A) General AI

A system with human-like intelligence across multiple domains

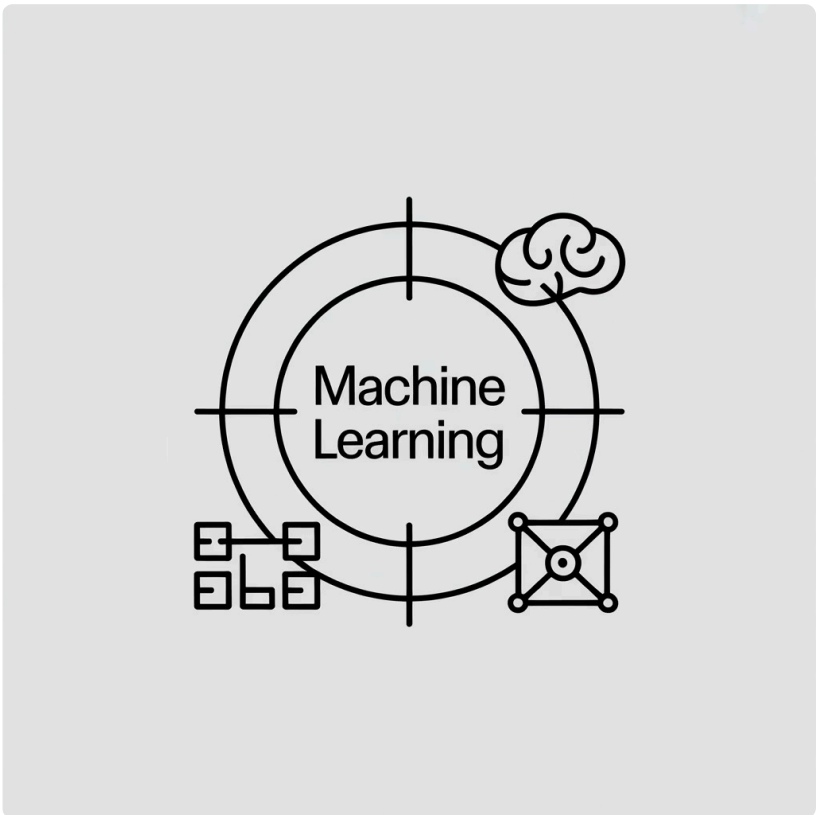
### B) Machine Learning / Deep Learning

A system trained on image data to recognize specific patterns

### C) Traditional Programming

A system using explicit rules to identify characters

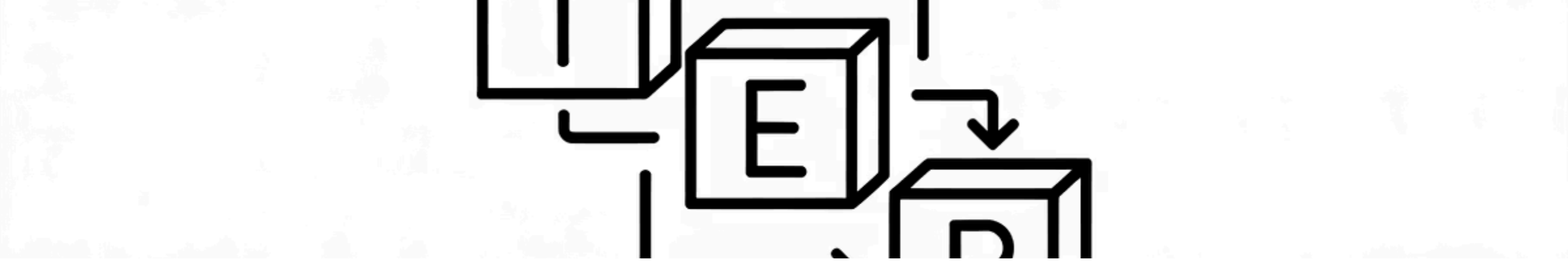
# Our Focus Today




## Machine Learning

Our primary focus is on ML because it's the most versatile and accessible tool for solving WCF's biggest challenges:

- It works with the structured data we already have
- It offers the best balance of complexity vs. value
- It can be implemented with existing IT resources
- It addresses our most pressing operational needs




# A More Formal Definition

> "A computer program is said to learn from **experience E** with respect to some **task T** and some **performance measure P**, if its performance on T, as measured by P, improves with experience E."

— Tom Mitchell, 1997

This engineering-focused definition frames machine learning as a measurable discipline with clear components: the task we want to accomplish, the experience (data) we learn from, and how we measure improvement.

# An Example: A Fraud Detection Model

### Task (T)

To flag new incoming claims as "Potentially Fraudulent" or "Normal" - a binary classification problem that helps WCF prioritize investigations

### Experience (E)

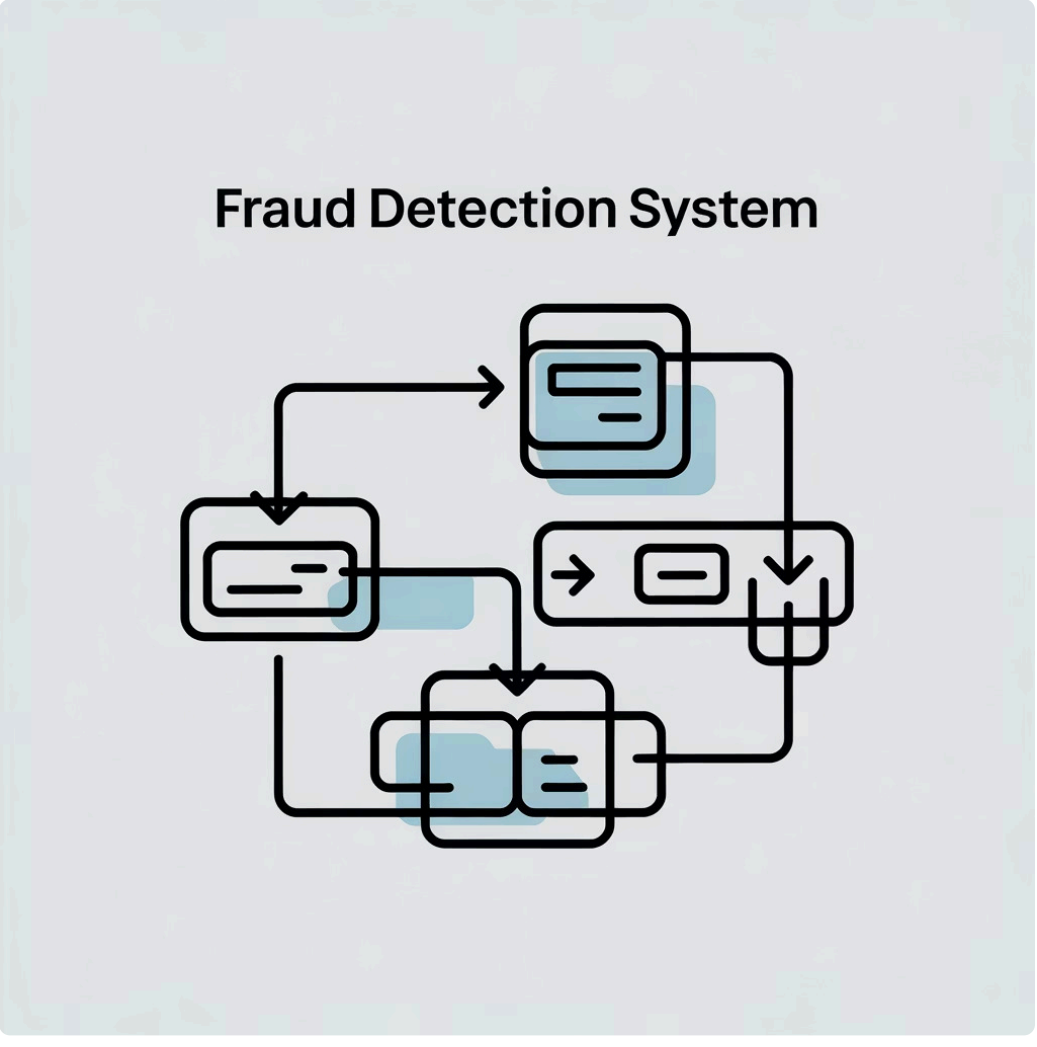
The **training data** – thousands of past claims that have been manually audited and labeled as either fraudulent or normal, including their features and outcomes

### Performance (P)

The **accuracy** of the model - what percentage of new claims does it classify correctly? Also precision, recall, and other fraud-specific metrics

As we feed the model more examples (more experience), its accuracy (performance) at flagging fraud (the task) should continuously improve over time.

# The Traditional Approach: A World of Rules

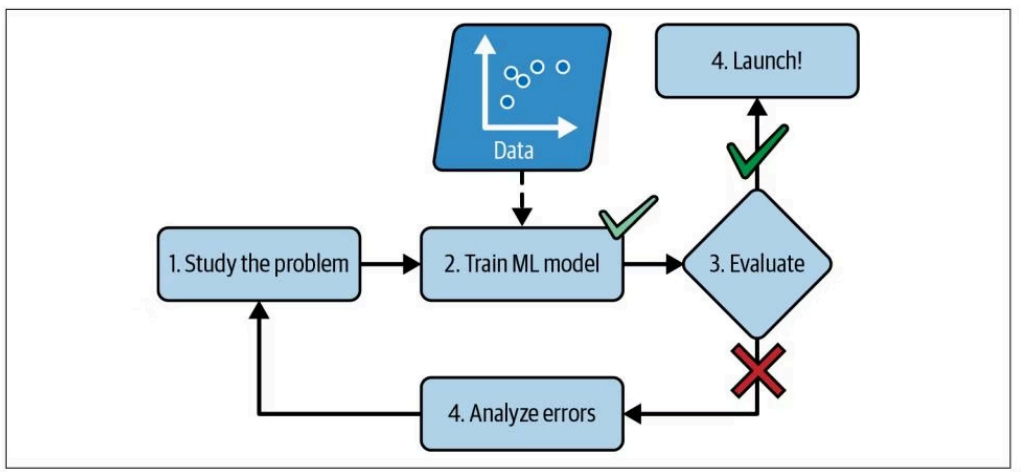



### WCF Traditional Process:

1. **Study the problem:** Interview fraud investigators about patterns
2. **Write explicit rules:** IF clinic_is_new AND lawyer_is_involved AND cost > X THEN flag_as_fraud()
3. **Evaluate & repeat:** Test against known fraud cases, find gaps, add more rules

**The Downside:** Your program quickly becomes a long list of complex, hard-to-maintain rules that require constant updates as fraud tactics evolve

# The ML Approach: Let the Data Speak




### WCF ML Process:

1. **Study the problem:** Still understand the domain fundamentals
2. **Train ML model:** Feed labeled historical claims (E) to an algorithm which **automatically learns** which patterns best predict fraud
3. **Evaluate:** Test against new data to ensure it generalizes well

**The Advantage:** The resulting program is shorter, easier to maintain, more accurate, and can adapt to new fraud patterns

# So, Machine Learning is great for...

### Problems with long lists of complex rules

When traditional programming would require hundreds of intricate, conditional statements that are difficult to maintain

### Complex problems with no known traditional solution

Tasks that humans can do intuitively but struggle to articulate as explicit algorithms (image recognition, natural language understanding)

### Fluctuating, dynamic environments

Situations where patterns change over time, requiring constant adaptation of the underlying rules

### Getting insights from large amounts of data

Discovering hidden patterns, correlations, and trends within massive datasets (Data Mining)

Let's examine specific WCF examples for each of these scenarios...

# Great for... Problems with Complex Rules


## Example: Claim Triage

Instead of programming hundreds of explicit rules to determine which claims need senior case manager review, an ML model can:

- Learn subtle patterns from thousands of historical complex cases
- Identify combinations of factors that aren't obvious to human reviewers
- Adapt to changing patterns in claim complexity over time
- Provide consistent, objective assessments across all claims

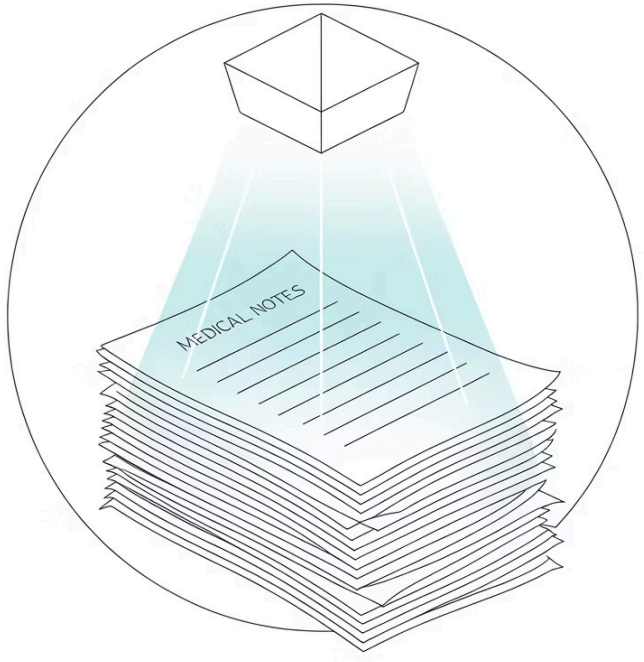
The code is simpler, more maintainable, and likely more effective at identifying truly complex cases.

# Great for... Problems with No Known Algorithm

## Example: Reading Doctor's Notes

How would you write a traditional program to read a doctor's messy, handwritten prescription or medical notes?

- Nearly impossible to create explicit rules for every handwriting style
- ML models can be trained on thousands of examples to recognize patterns
- Modern OCR systems use deep learning to achieve human-level accuracy
- Can extract critical information about diagnosis, treatment plans, and restrictions




**DEMO:** Azure Form Recognizer / Google Vision API showing real-time transcription of handwritten medical notes

# Great for... Fluctuating Environments

## Example: Evolving Fraud Tactics

Fraudsters continuously create new schemes to exploit the system:

- A traditional rules-based system would need constant manual updates
- An ML system can be **automatically retrained** on new data
- Nightly retraining allows detection of emerging fraud patterns
- System performance improves over time as it encounters more examples


The automated retraining loop creates a system that adapts to changing fraud tactics without requiring constant programmer intervention.

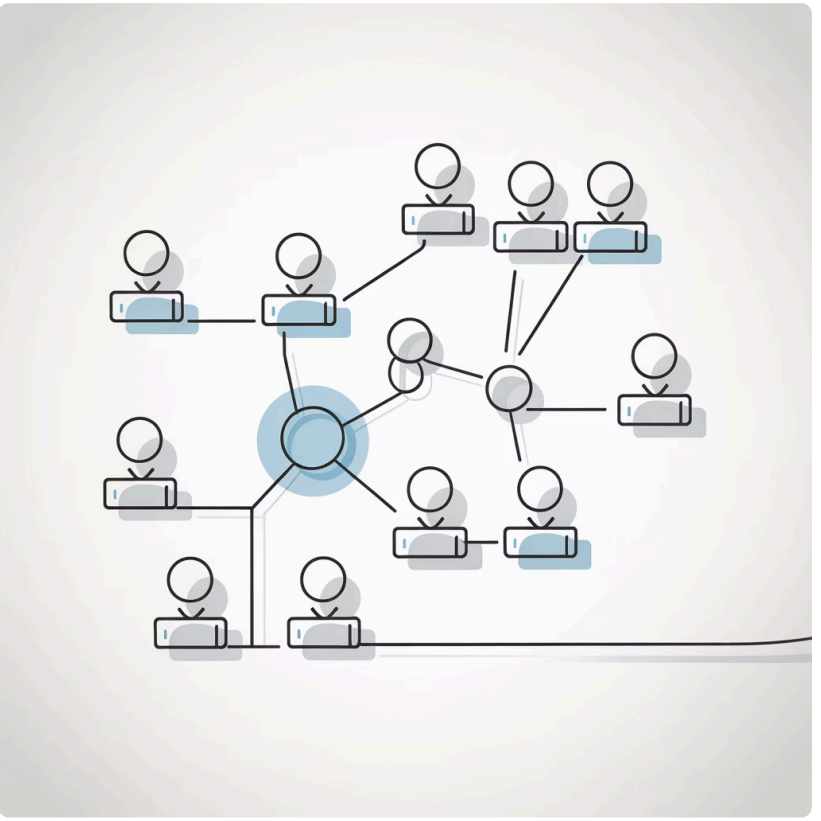
# Great for... Data Mining

## Example: Uncovering Provider Abuse

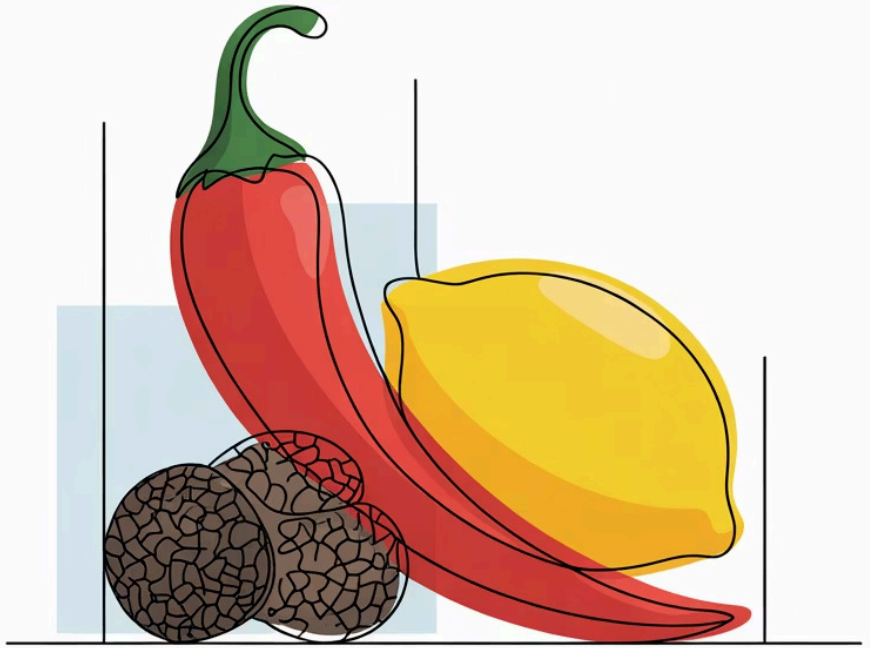
By analyzing massive volumes of billing data across providers, an ML system might reveal unexpected correlations:

- A specific medical clinic consistently keeps patients off work 30% longer than the national average for identical injuries
- A physical therapy provider bills for an unusual combination of treatments that rarely appear together elsewhere
- A particular lawyer is associated with a statistically improbable spike in certain claim types

These insights weren't explicitly programmed—they emerged from the data patterns.



Machine learning can identify unusual patterns that human analysts might never discover through manual investigation.



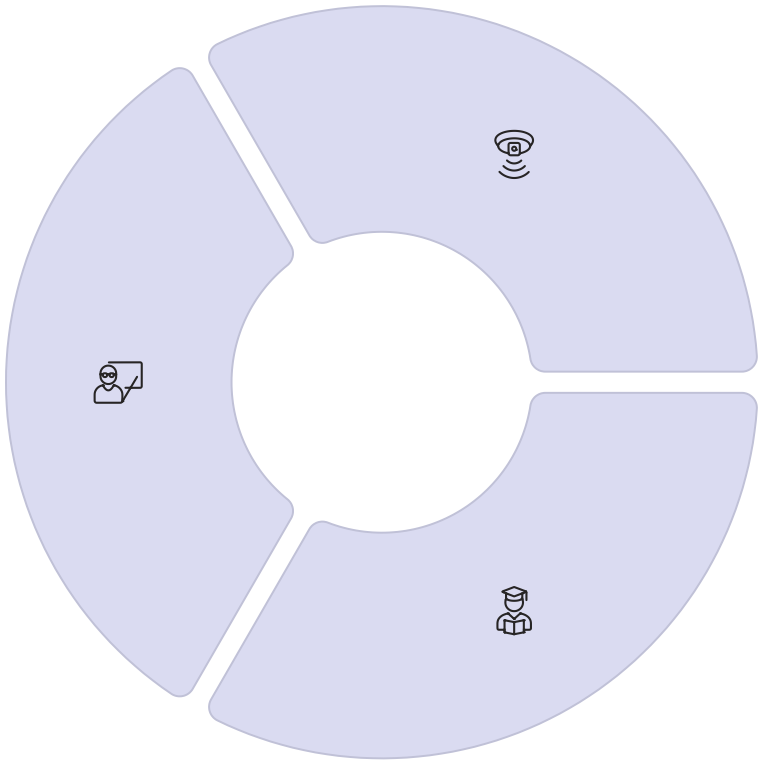
# The Three Flavors of Machine Learning

Now that we understand what Machine Learning is, let's explore the three main approaches to training these systems. Each approach solves different types of problems and requires different kinds of data.

# The Three Flavors of Machine Learning

### Supervised Learning

Learning from labeled examples



### Unsupervised Learning

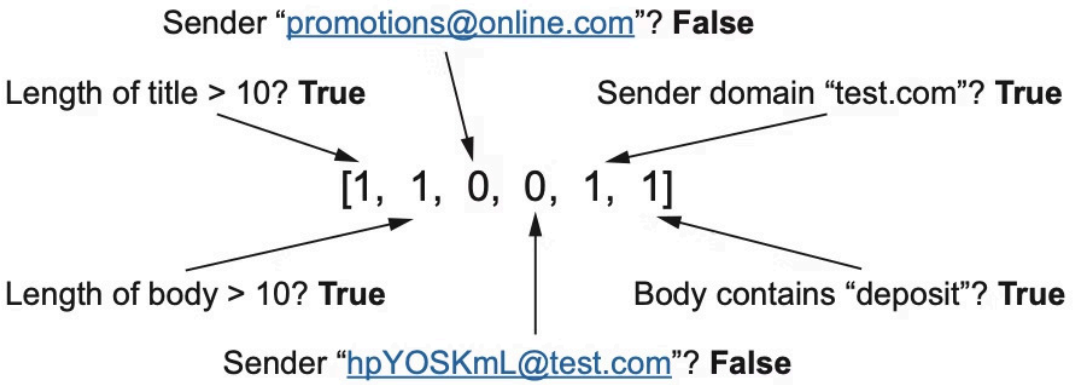
Finding hidden patterns

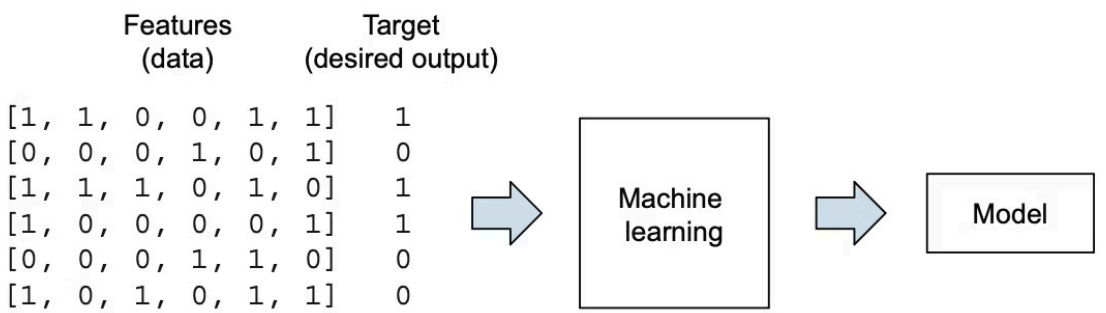
### Reinforcement Learning

Learning through trial and error

Each approach has distinct strengths and applications for WCF operations.

# Supervised Learning




The model learns from data that is already **labeled** with the correct answer. You act as the teacher, providing examples and the right answers.

## How it works:

1. Collect historical data
2. Label each example with the correct answer
3. Train the model to recognize patterns
4. Use the model to predict outcomes for new data

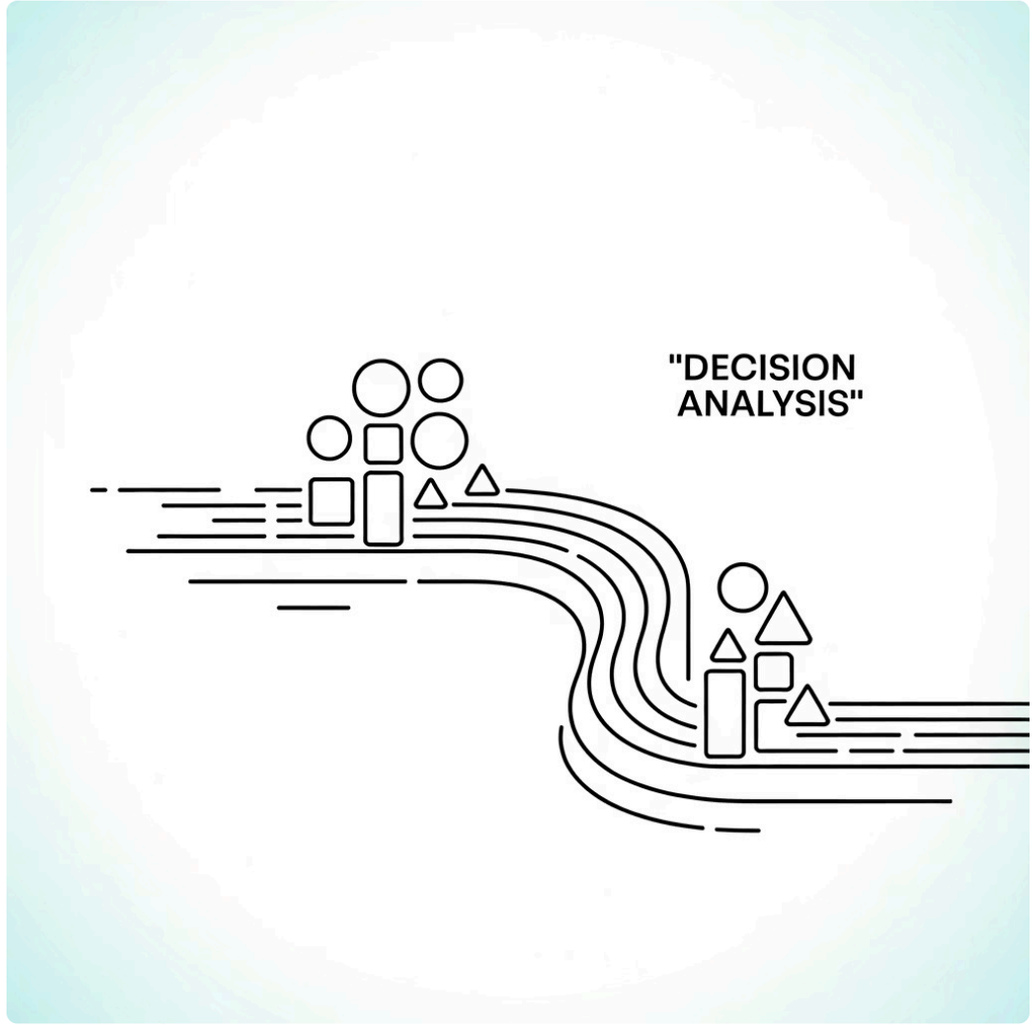
# Supervised Task: Classification

## Concept: Predicting a Category

Classification models predict which **discrete class or category** a given example belongs to. The output is a class label or probability distribution across classes.

### Classification Examples:

- **Fraud Detection:** Is this claim fraudulent? (Yes/No)
- **Injury Categorization:** What is the primary body part injured? (Hand/Back/Foot/Eye)
- **Industry Classification:** What industry does this employer belong to? (Mining/Construction/Retail)
- **Claim Complexity:** Will this claim require specialist intervention? (Simple/Complex)




Classification algorithms learn to create decision boundaries that separate different classes in the feature space.

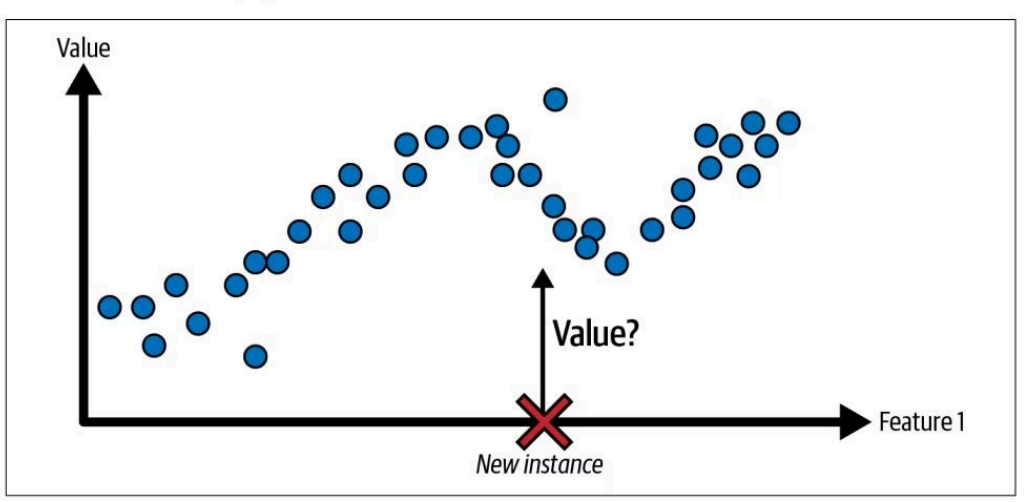
# Supervised Task: Regression

## Concept: Predicting a Numeric Value

Regression models predict **continuous quantities** rather than discrete categories. The output is a number within a possible range.

### Regression Examples:

- **Cost Prediction:** What will be the final total cost of this claim? (e.g., 15,450,000 TZS)
- **Time-Loss Estimation:** How many days will the employee be unable to work? (e.g., 90 days)
- **Recovery Timeline:** How long until maximum medical improvement? (e.g., 120 days)
- **Resource Allocation:** How many case manager hours will this claim require? (e.g., 25 hours)




Regression algorithms learn to fit a function (often a line or curve) that best predicts the target value based on input features.

# Unsupervised Learning

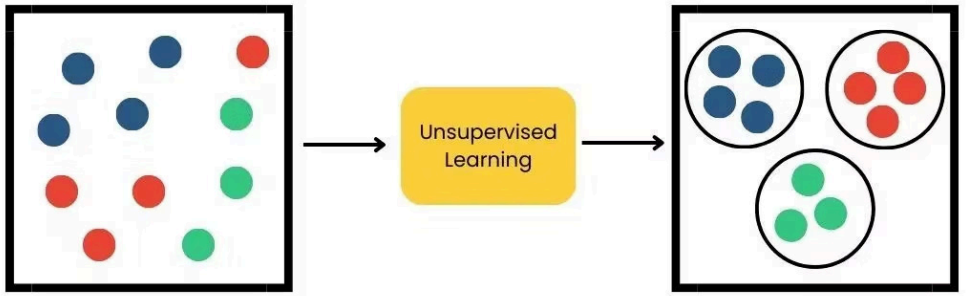
## Finding Hidden Patterns

The model is given **unlabeled** data and its job is to find hidden patterns, structures, or "odd ones out" on its own.

It's a detective looking for clues without being told what to look for.

### Key Benefit:

Can discover patterns we didn't know existed or even think to look for.



# Unsupervised Task: Clustering

## Concept: Finding Natural Groupings

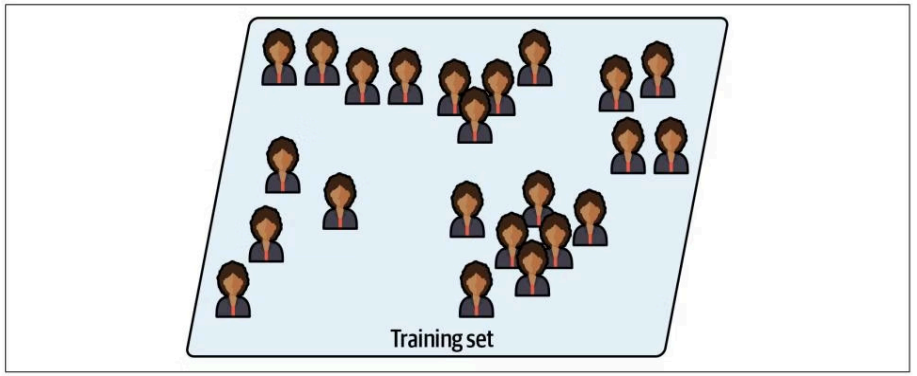
Clustering algorithms detect groups of similar instances within data, identifying underlying patterns without being told what to look for.

### Clustering Example:

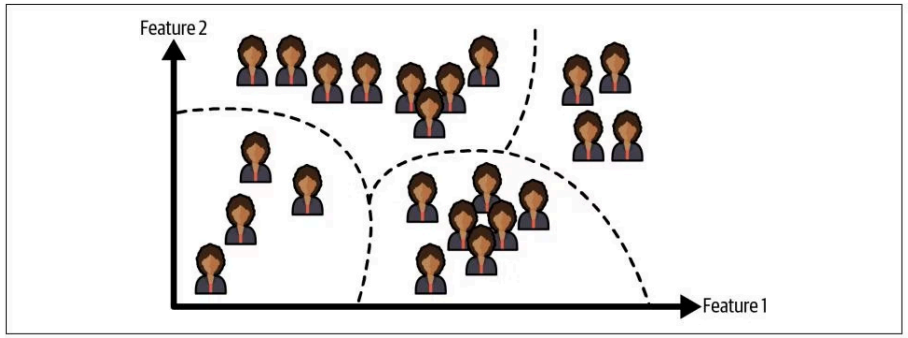
By analyzing claimant and claim data without predefined categories, the system might automatically discover distinct profiles:

- "Young, short-term claims in urban areas"
- "Older, long-term claims in the mining sector"
- "Mid-career, moderate severity claims with multiple providers"
- "High-cost claims with specific co-morbidity factors"

These discovered patterns can inform targeted intervention strategies and resource allocation.




*An unlabeled training set*




Clustering similar instances

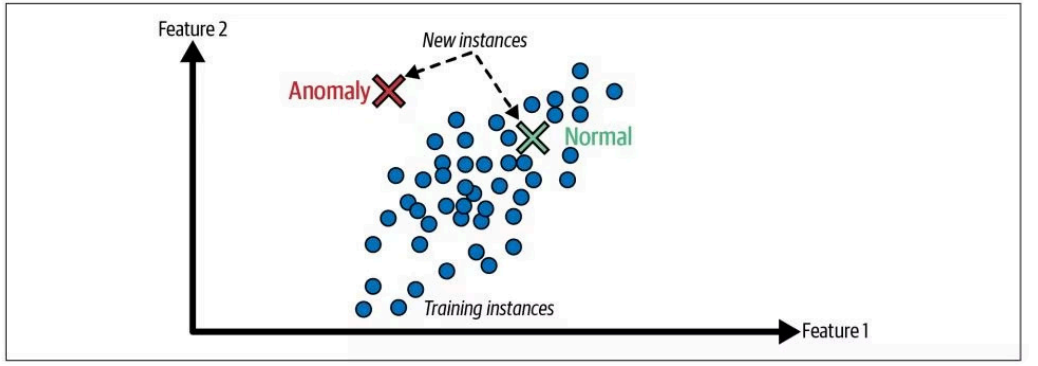
# Unsupervised Task: Anomaly Detection

## Concept: Finding Outliers

Anomaly detection algorithms learn what "normal" patterns look like, then identify instances that significantly deviate from those patterns.

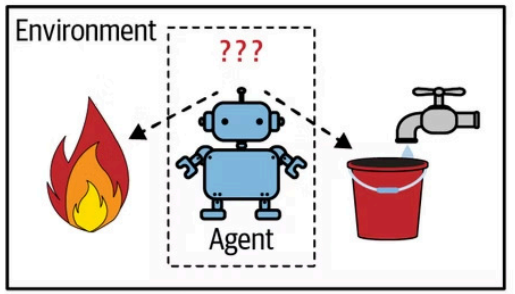
### Anomaly Examples:

- A medical provider suddenly billing 10x more procedures than their historical average
- Unusual patterns in treatment sequences that differ from standard care protocols
- Abnormal timing patterns in claim documentation submission
- Statistically improbable clusters of similar claims from unrelated claimants

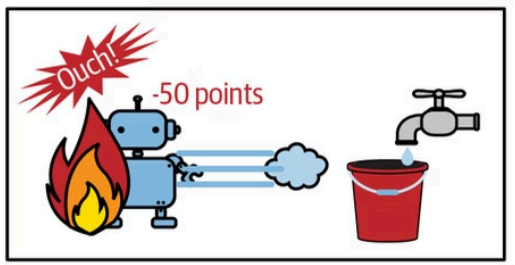



Anomaly detection is critical for fraud prevention, quality control, and identifying exceptional cases that may require special handling or investigation.

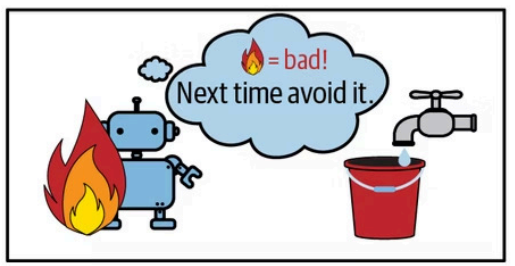
# Reinforcement Learning




1. Observe
2. Select action using policy




3. Action!
4. Get reward or penalty




5. Update policy (learning step)
6. Iterate until an optimal policy is found

The model learns by "trial and error." It takes actions in an environment and learns to maximize a cumulative reward.

## How it works:

1. Define the environment and possible actions
2. Set up rewards for desired outcomes
3. Allow the agent to explore and learn
4. The agent improves by maximizing rewards

This is the technology behind game-playing AI like AlphaGo and is used in robotics.

**Example:** Optimizing case management interventions by learning which actions lead to better outcomes for different claim types.

# The Main Categories

| Category | Data Has Labels? | WCF Goal |
|---|---|---|
| **Supervised Learning** | Yes | Predict a value or category (fraud, cost, duration, complexity) |
| **Unsupervised Learning** | No | Find hidden structure (claim groups, anomalies, patterns) |
| **Semi-Supervised** | Partially | Leverage limited labeled data with abundant unlabeled data |
| **Reinforcement Learning** | No (has rewards) | Learn optimal strategies (intervention timing, resource allocation) |

The right approach depends on your data situation, the problem you're solving, and your specific goals. Most real-world ML applications at WCF will likely begin with supervised learning techniques.

# Batch vs. Online Learning

### Batch Learning

System is trained using all available data at once

Training happens offline, then model is deployed

Cannot learn incrementally from new data

Must be retrained from scratch to incorporate new data

### Online Learning

System learns incrementally from data streams

Each learning step is fast and cheap

Can adapt to change rapidly

Works well with limited computing resources

The choice between batch and online learning depends on your data volume, how quickly it changes, and available computing resources.

# Batch Learning: Offline Approach

In batch learning (also called offline learning), the system:

- Is trained on the complete dataset
- Cannot learn incrementally once deployed
- Applies what it has learned without further training

A key challenge is **model rot** or **data drift** - the model's performance decays over time as the world changes while the model remains static. The solution is to regularly retrain the model on up-to-date data.

Batch learning works well when you have sufficient computing resources and don't need to adapt to rapidly changing data.

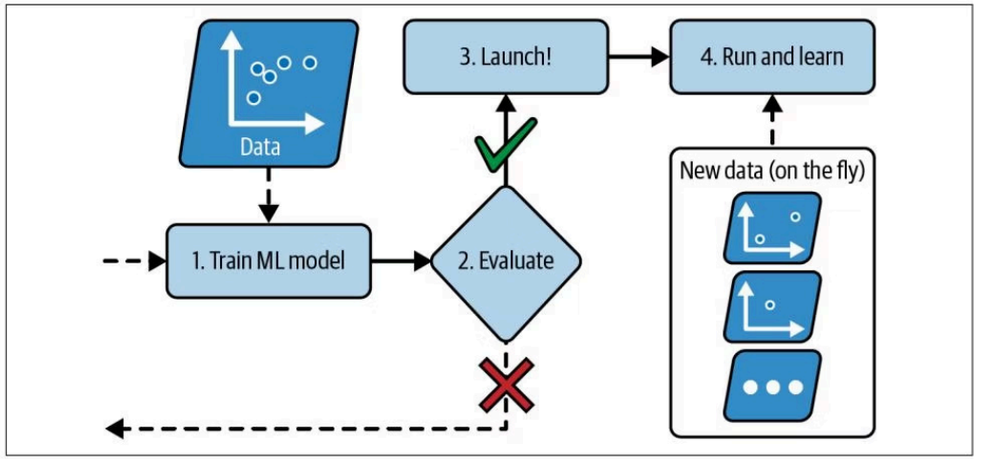
# Online Learning: Incremental Approach

Online learning systems train incrementally by feeding data instances sequentially, either individually or in mini-batches.

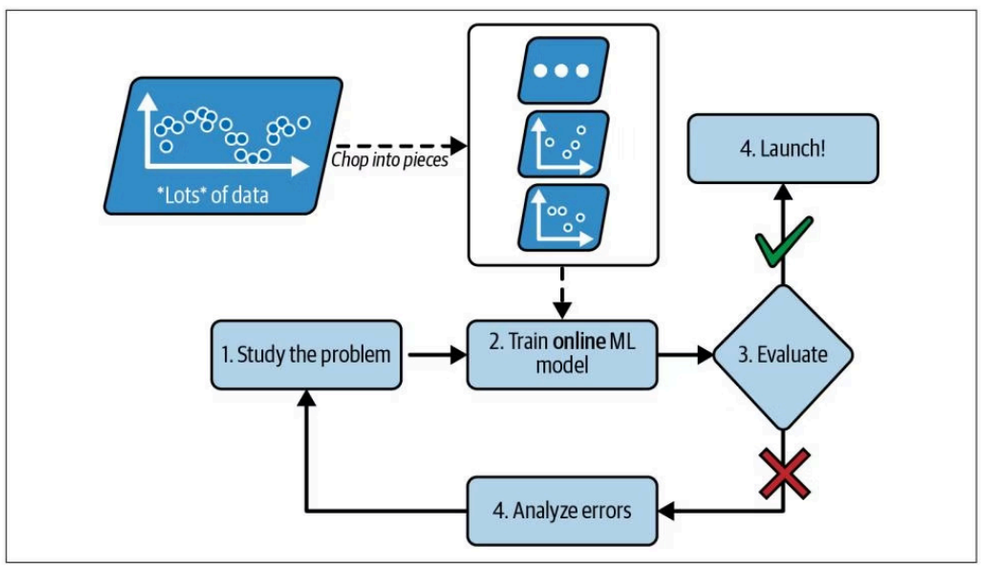
Advantages:

- Adapts quickly to changing data
- Works with limited computing resources
- Handles huge datasets through out-of-core learning

The learning rate controls how quickly the system adapts to new data - higher rates adapt faster but may forget old patterns, while lower rates learn more slowly but are less sensitive to noise.




*Online learning model continues learning as new data arrives*




Out-of-core learning for huge datasets

# Instance-Based vs. Model-Based Learning

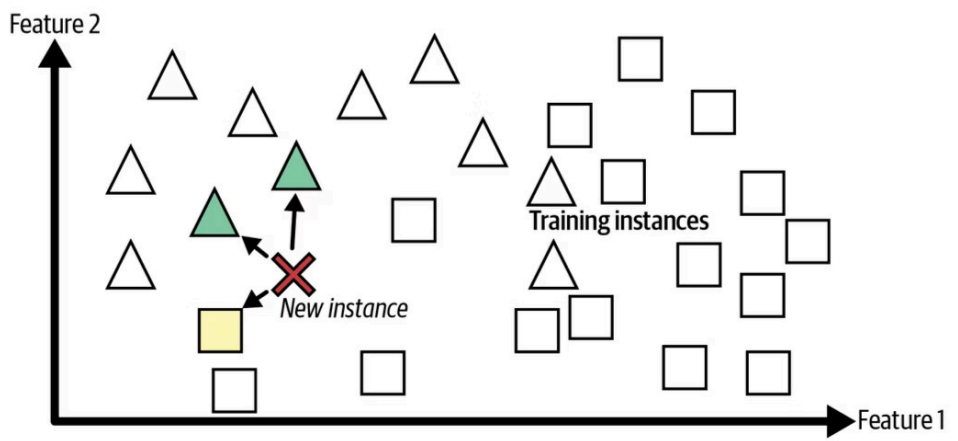
## Instance-Based Learning

Learns examples by heart

Generalizes to new cases using a similarity measure

Compares new instances to learned examples

Example: k-nearest neighbors algorithm



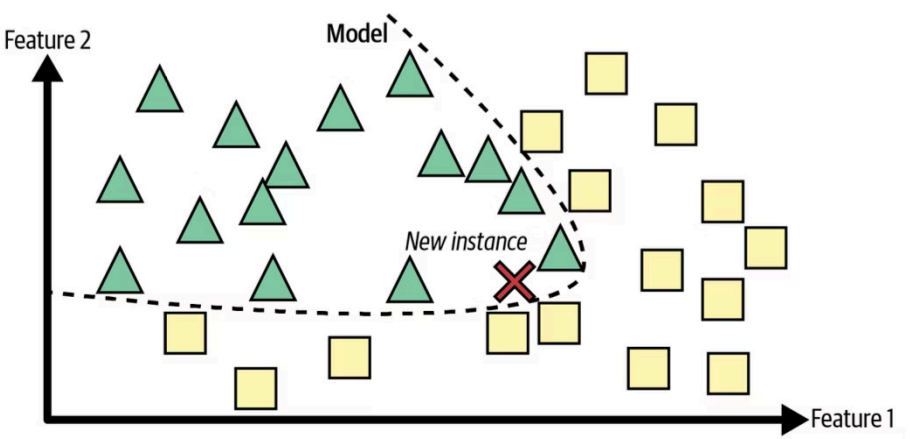

## Model-Based Learning

Builds a model from examples

Uses the model to make predictions

Involves model selection and training

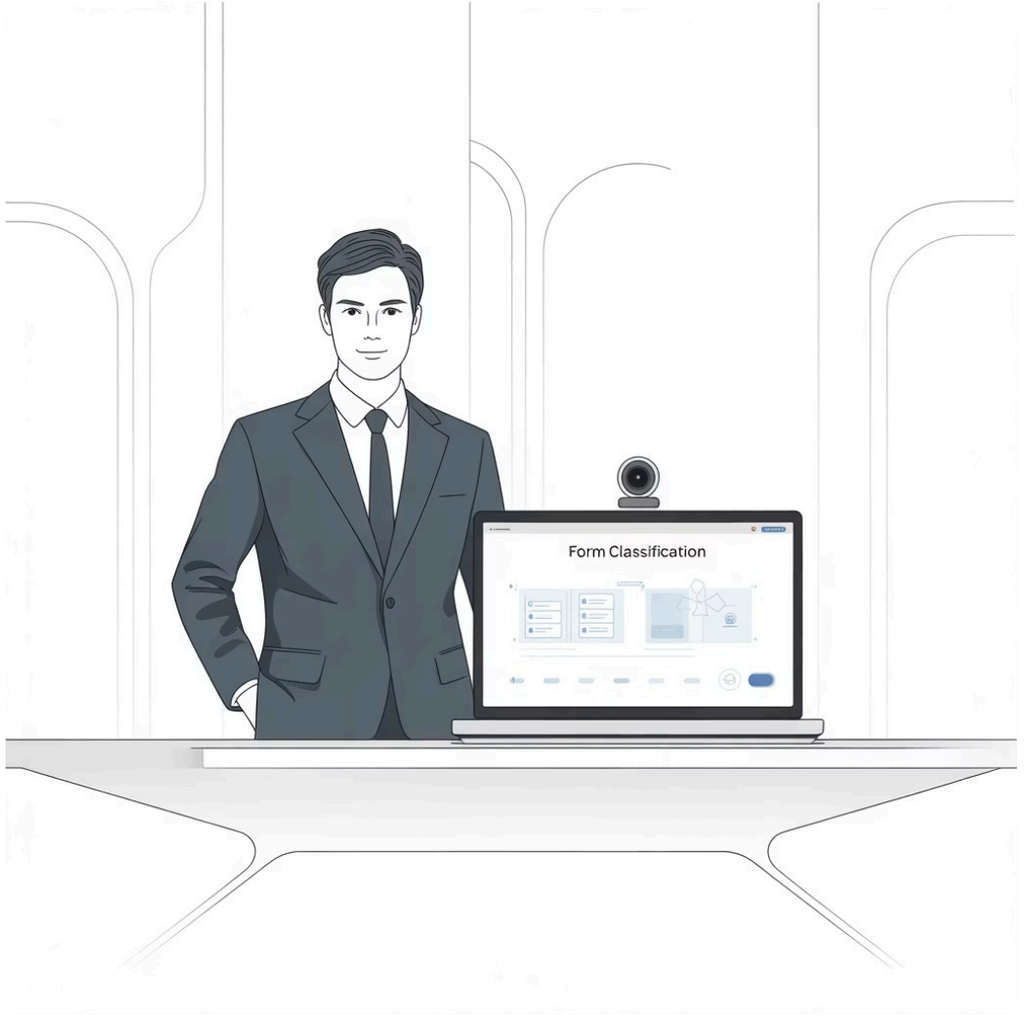
Example: linear regression



# LET'S TRY IT! - Building a Classifier in 5 Minutes

## Hands-On Demo: Teachable Machine

1. Open Google's Teachable Machine "Image Project"
2. Create two classes:
   - "Valid WCF Form"
   - "Incomplete WCF Form"
3. Use the webcam to capture 5-10 examples of each class
   - Show complete forms for the first class
   - Show forms with missing signatures, incomplete sections, or other issues for the second class
4. Click "Train Model" and watch it learn
5. Test with new forms to see classification in real-time

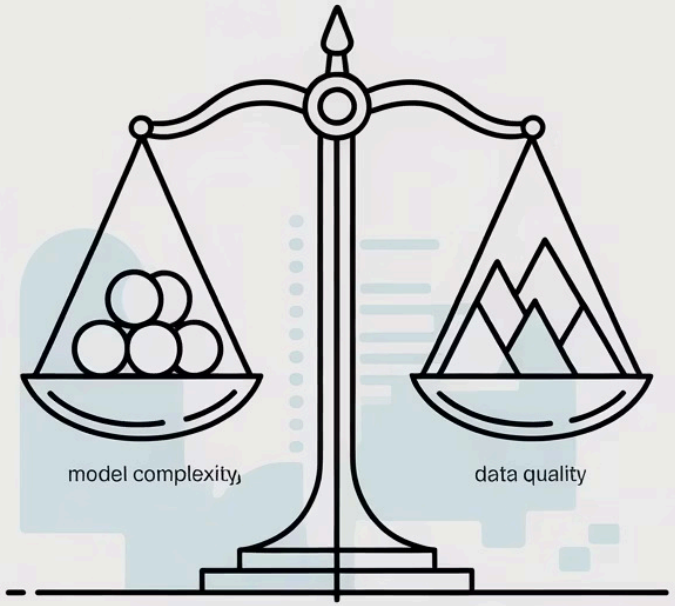



This simple demo powerfully illustrates supervised learning in action—a machine learning model learning to distinguish between valid and invalid forms with just a few examples.

# It's Not Magic: The Main Challenges

> "In short, since your main task is to select a model and train it on some data, the two things that can go wrong are **'bad model'** and **'bad data.'**"

Understanding these core challenges is essential for successful ML implementation. Both dimensions require careful consideration and often involve tradeoffs that must be managed thoughtfully.



# Challenge 1: "Bad Data"

### Insufficient Quantity

ML needs lots of data, especially for complex problems. Rule of thumb: you need at least 10 times as many training examples as there are dimensions in your data.

**Impact:** Rare fraud schemes or unusual claim types may have too few examples for reliable modeling.

### Non-Representative Data (Sampling Bias)

"Your training data must be representative of the cases you want to generalize to." Historical biases in claim handling can perpetuate through models.

**Impact:** If your model is trained only on claims from certain industries, it may perform poorly on others.

### Poor Quality Data

Errors, outliers, missing values, and noise can significantly degrade model performance.

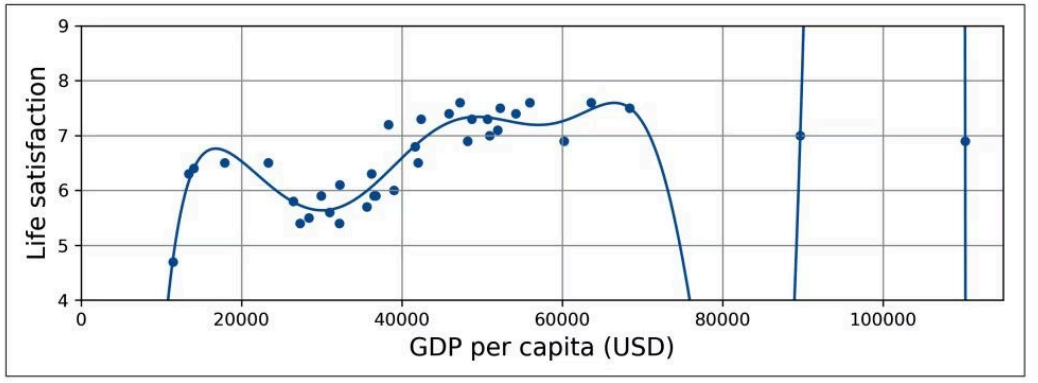
**Impact:** Inconsistent data entry practices, legacy system migrations, and manual processes can introduce data quality issues.

### Irrelevant Features

The data must contain useful signals related to what you're trying to predict.

**Impact:** Administrative or process-related fields may not correlate with outcome variables of interest.

# Challenge 2: "Bad Model"




## Finding the Sweet Spot

### Overfitting

"The model performs well on the training data, but does not generalize well to new data." It's too complex and learns the noise in the data.

**Analogy:** Like a taxi driver who memorizes specific routes but gets lost on streets they haven't seen before.

### Underfitting

"The model is too simple to learn the underlying structure of the data." It fails to capture important patterns and performs poorly even on training data.

**Analogy:** Like a taxi driver who only knows how to drive in straight lines, missing important turns.

# Testing and Validating

### Split Data

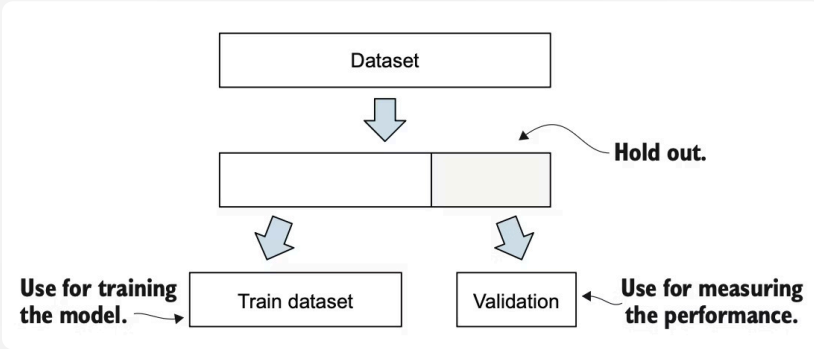
Divide your dataset into training and test sets (typically 80% training, 20% testing)

### Train Model

Use only the training set to train your model

### Evaluate Performance

Test the model on the test set to estimate generalization error




If training error is low but generalization error is high, your model is overfitting the training data.

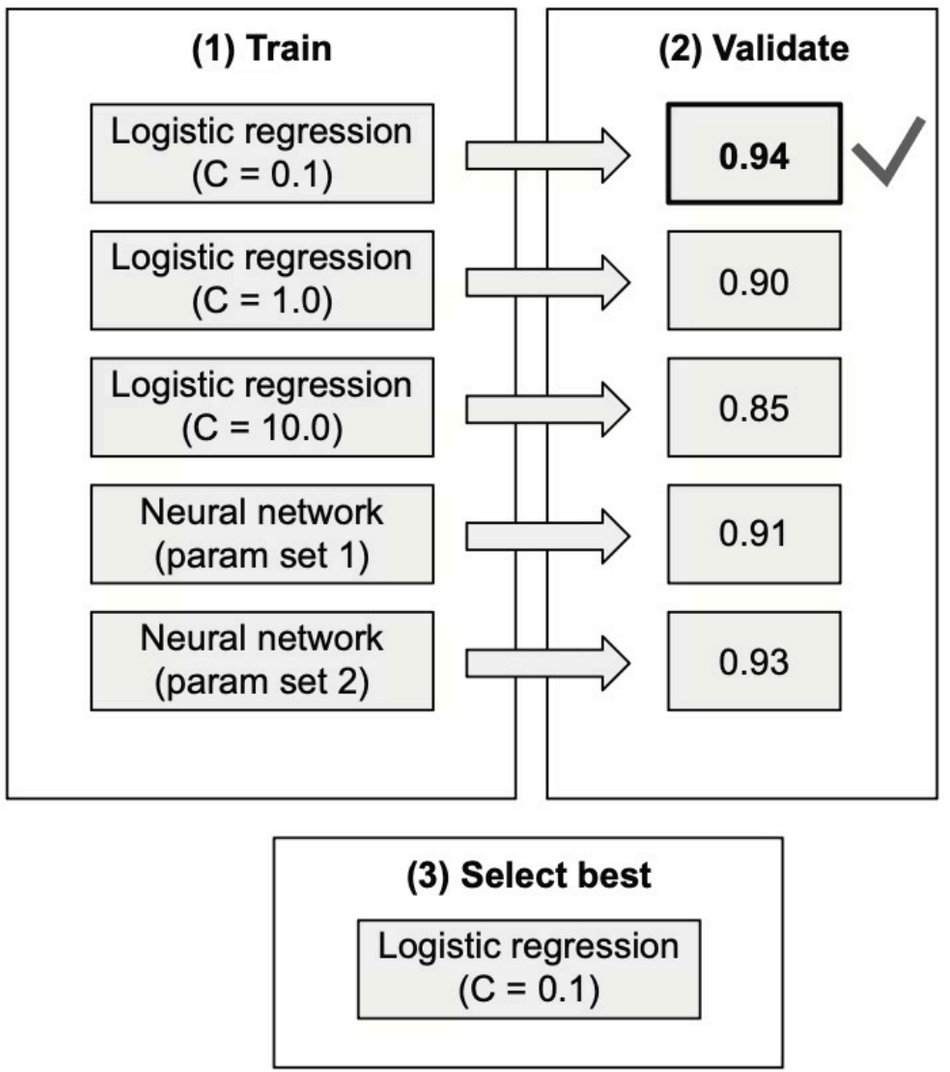
The test set should be large enough to provide a good estimate of performance. With very large datasets, even 1% might be sufficient (e.g., 100,000 instances from a 10-million instance dataset).

# Hyperparameter Tuning and Model Selection

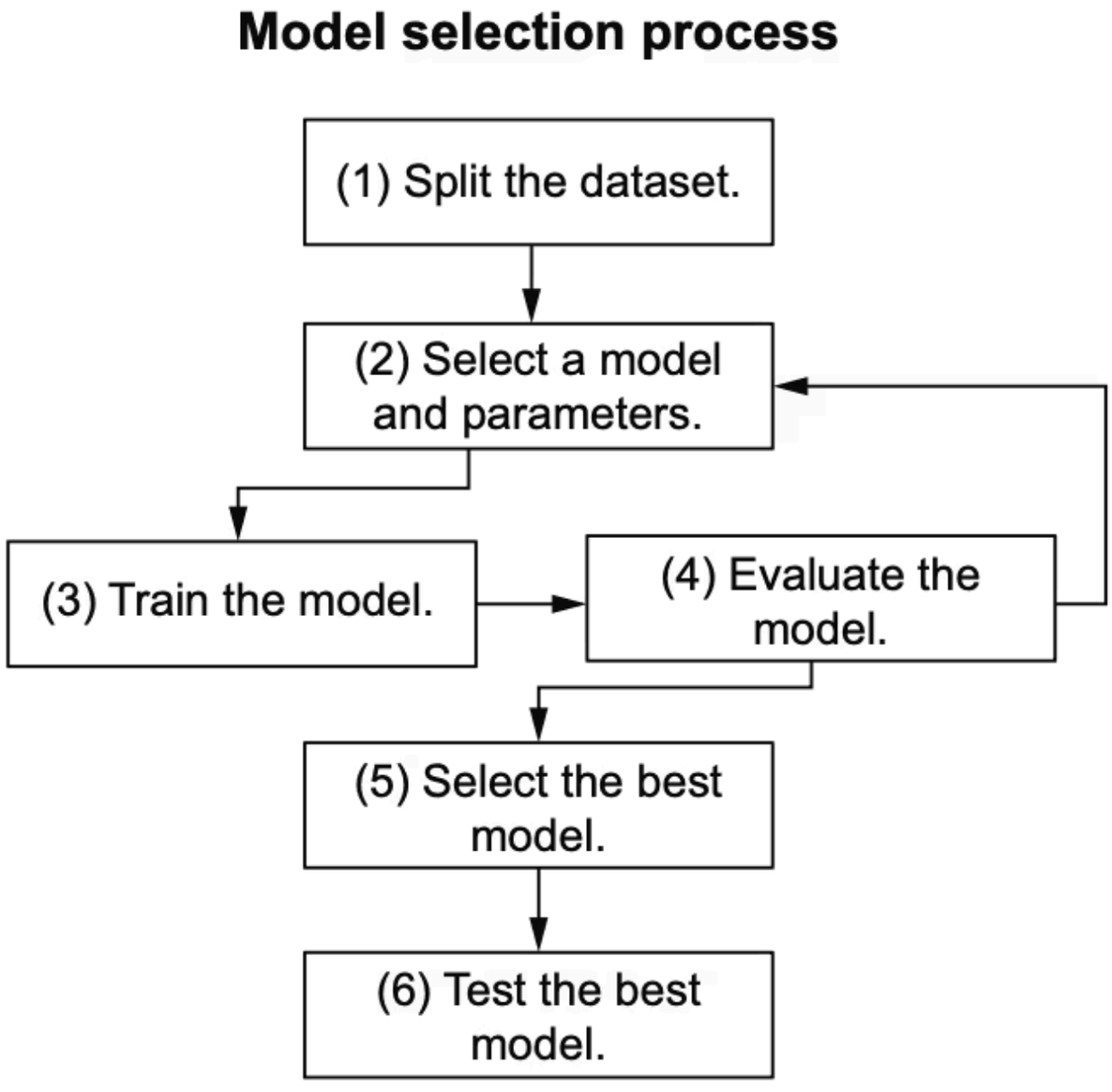
When comparing models or tuning hyperparameters, you need a validation strategy:

1. Hold out part of the training data as a validation set
2. Train multiple models with different hyperparameters on the reduced training set
3. Select the model that performs best on the validation set
4. Train the final model on the full training set (including validation data)
5. Evaluate on the test set to estimate generalization error

For data mismatch situations (when training data differs from real-world data), use a train-dev set to diagnose whether poor performance comes from overfitting or data mismatch.



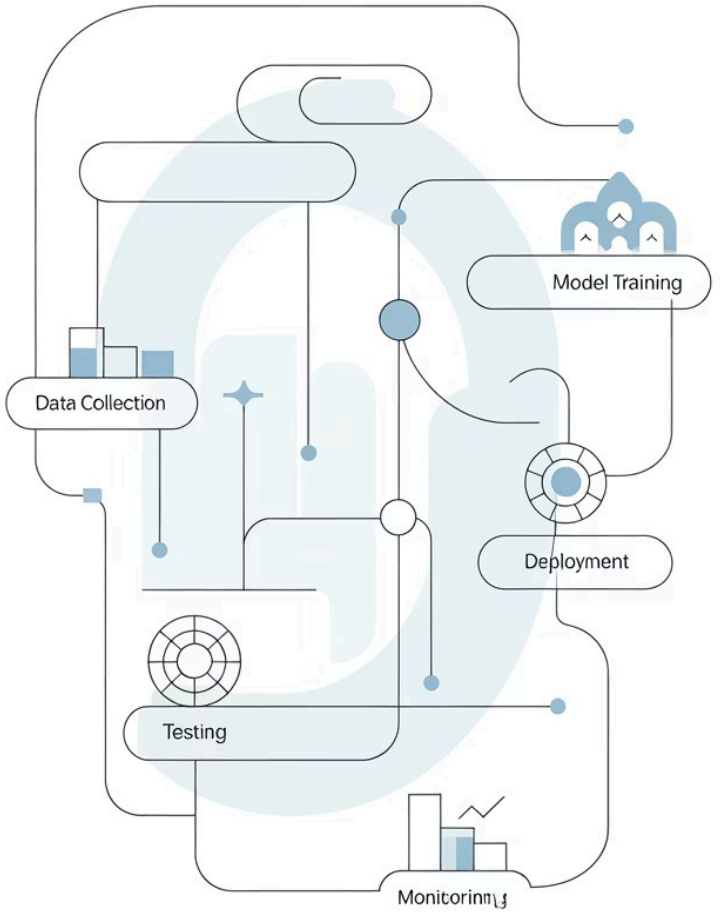
# Model Selection



# What We've Learned

1. ML enables computers to learn rules from data, rather than requiring explicit programming
2. ML excels at handling complex, evolving problems and finding hidden insights in large datasets
3. We can classify ML systems by their learning approach (Supervised, Unsupervised, etc.)
4. Supervised learning predicts categories (classification) or values (regression)
5. Unsupervised learning discovers patterns without labeled examples
6. The biggest challenges involve ensuring good data quality and building models that generalize well

# AI Project Lifecycle




# Next: End-to-end project

Next, we'll explore the end-to-end ML project workflow:

1. Problem framing and data acquisition
2. Data preparation and feature engineering
3. Model selection and training
4. Evaluation and refinement
5. Deployment and monitoring

We'll see how these stages connect to form a complete ML solution.