Machine Learning Paradigms

Traditional vs AI-Based Approaches

Machine Learning Overview

2025

October 17, 2025

Traditional vs Machine Learning Approaches

Traditional Programming
Start with **explicit rules**:

- Experts encode domain knowledge as rules
- Programmer implements decision logic
- System follows predetermined instructions
- Behavior is fully specified in code

Characteristics:

- Transparent and interpretable
- Works well for well-defined problems
- Requires comprehensive domain expertise
- Struggles with complex, nuanced patterns
- Rules must be manually updated

Machine Learning
Start with **data and examples**:

- Algorithms discover patterns from data
- Rules emerge through learning process
- System improves with more experience
- Behavior learned, not programmed

Three Main Learning Paradigms:

- **Supervised** - Learn from labeled examples
- **Unsupervised** - Find structure in unlabeled data
- **Reinforcement** - Learn optimal actions through trial and error

Fundamental difference: Traditional = explicit programming, ML = learning from data

What is Machine Learning?

Core Definition

"A computer program learns from experience E with respect to task T if its performance P improves with experience." — Tom Mitchell

Key Concepts:

- **Learning** = improving performance through experience
- **Generalization** = performing well on new, unseen data
- **Not memorization** = patterns, not examples

What ML Needs:

- Data (the "experience")
- Features (representation of data)
- Learning algorithm
- Performance metric

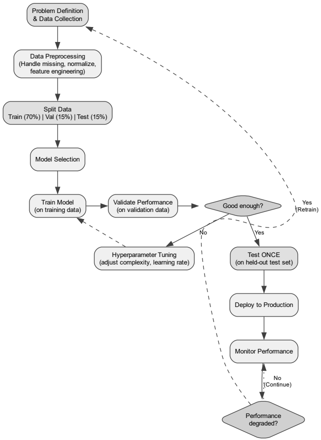
The Learning Process

1. **Collect data**: Gather examples and experiences
2. **Represent data**: Extract features (numbers, vectors)
3. **Split data**: Training, validation, testing
4. **Train model**: Learn patterns from training data
5. **Validate**: Tune parameters on validation data
6. **Test**: Evaluate on completely unseen test data

Critical Insight:
The goal is NOT to memorize training data, but to generalize to new situations.
Test performance measures true learning.

ML is fundamentally about pattern recognition and prediction from data, not rule following

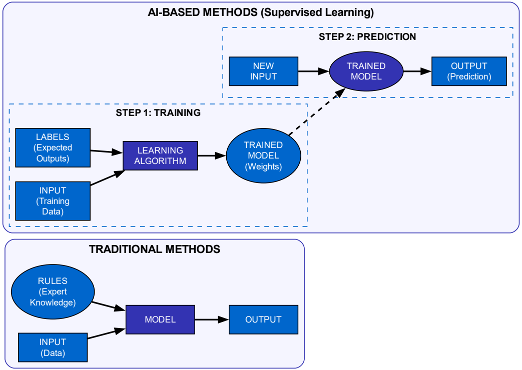
The Machine Learning Workflow



Key Stages: Data Collection → Preprocessing → Split (Train/Val/Test) → Train → Validate (iterate until good) → Test ONCE → Deploy → Monitor (retrain when needed)

Critical: Iterative process with feedback loops - validation guides tuning, monitoring triggers retraining

Supervised Learning: Learn from Labeled Examples



Core Idea:
Learn mapping $f : X \rightarrow Y$ from labeled examples (x_i, y_i)

Two Main Types:

- **Classification**: Predict discrete labels (cat/dog, spam/not spam)
- **Regression**: Predict continuous values (price, temperature)

Requirements:

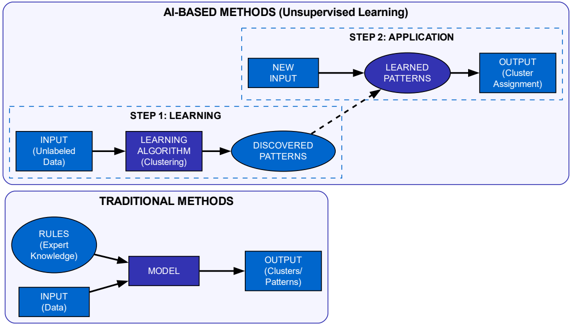
- Large labeled dataset
- Quality labels (expensive!)
- Representative examples

When to Use:

- Clear input-output relationship
- Labels available
- Prediction task

Evaluation:

Unsupervised Learning: Find Structure in Unlabeled Data



Core Idea:
Discover patterns in data without labels or guidance

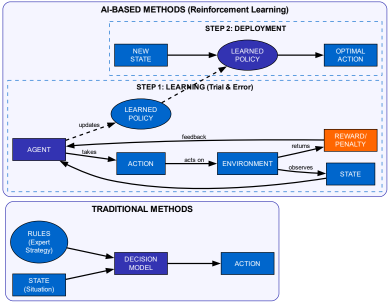
Main Techniques:

- **Clustering**: Group similar data points (K-means, hierarchical)
- **Dimensionality Reduction**: PCA, t-SNE, UMAP
- **Anomaly Detection**: Find outliers
- **Generative Models**: Learn data distribution

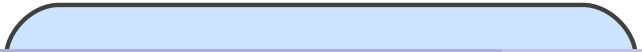
When to Use:

- No labels available
- Exploratory analysis
- Data preprocessing
- Understanding structure

Challenge:
No objective ground truth for evaluation -



The RL Loop:



Core Idea:
Agent learns policy $\pi(a|s)$ to maximize cumulative reward through trial-and-error

Key Components:

- **State** s: Current situation
- **Action** a: Possible choices
- **Reward** r: Feedback signal
- **Policy** π: Strategy to learn

When to Use:

- Sequential decisions
- Delayed rewards
- Control problems
- Game AI, robotics

Challenges:

- Sample inefficient (needs many trials)
- Credit assignment problem
- Exploration vs exploitation

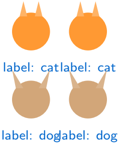
Supervised Learning Example: Cat vs Dog Classification

Traditional Approach
Expert writes rules:

- IF pointy ears AND whiskers AND meows → CAT
- IF floppy ears AND barks → DOG

CAT

ML Approach
Step 1: Training



→ LEARNS

Step 2: Prediction

→ **CAT** 99%

?

Reality Check:

- Model doesn't "understand" cats - it learns statistical patterns in pixel values that correlate with labels
- Needs thousands of labeled examples to generalize
- Performance depends heavily on training data quality and diversity
- Can fail on edge cases not represented in training data

Key: Supervised learning finds statistical correlations between features and labels, not semantic understanding

Unsupervised Learning Example: Animal Clustering

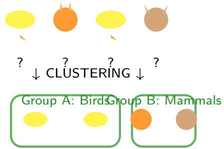
Traditional Approach

Expert defines categories:

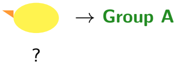
- Birds: Have feathers, wings, beaks
- Mammals: Have fur, give birth to live young

ML Approach

Step 1: Learning (NO LABELS)



Step 2: Application

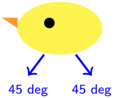


Example: Algorithm discovers natural groupings based on visual features without being told what defines each category

Traditional Approach

Expert programs instructions:

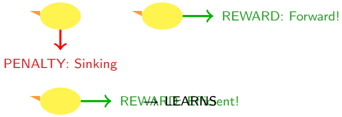
- Paddle left foot at angle 45 degrees
- Then paddle right foot
- Repeat every 0.5 seconds
- Adjust for current



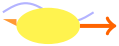
Programmed

ML Approach

Step 1: Learning (Trial & Error)



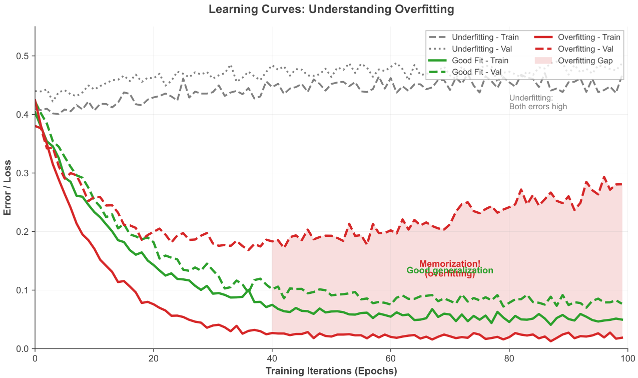
Step 2: Deployment



Efficient!

Example: Duck learns optimal swimming through thousands of attempts, discovering techniques experts might never explicitly program

Model Evaluation: Understanding Overfitting



The Problem:
Overfitting: Model memorizes training data, fails on new data
Good generalization: Small gap between train and validation error

Critical Rule:
NEVER evaluate on training data!
Training accuracy is meaningless - can achieve 100% by memorization.

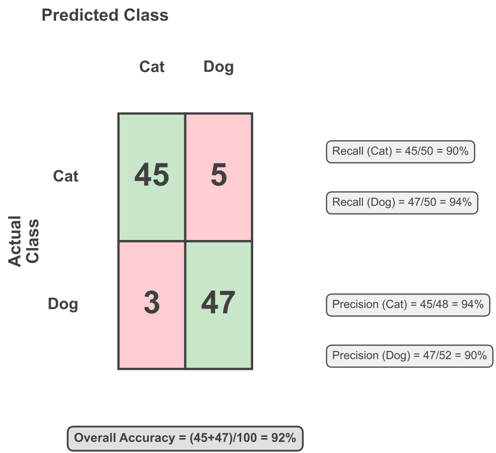
Data Splits:

- **Train** (70%): Fit
- **Val** (15%): Tune
- **Test** (15%): Evaluate ONCE

Solutions:

- More data
- Regularization
- Simpler model
- Early stopping

Evaluation Metrics & Confusion Matrix



Classification Metrics:

- **Accuracy**: Correct / Total
  (Can be misleading with imbalanced data!)
- **Precision**: TP / (TP + FP)
  How many predicted positives are actually positive?
- **Recall**: TP / (TP + FN)
  How many actual positives did we find?
- **F1 Score**: Harmonic mean of precision/recall
- **ROC-AUC**: Trade-off curve

Regression Metrics:

- MSE, RMSE, MAE
- R^2: Explained variance (0-1)

Confusion matrix reveals which types of errors your model makes - essential for understanding performance

Common Pitfalls in Machine Learning

1. Data Leakage
Information from test set “leaks” into training
Example: Normalizing before splitting data
Result: Artificially inflated performance

2. Overfitting
Model too complex, memorizes training data
Signs: High train accuracy, low test accuracy
Solutions: Regularization, more data, simpler model, dropout

3. Selection Bias
Training data not representative of real distribution
Example: Face recognition trained only on certain demographics
Result: Poor performance on underrepresented groups

4. Distribution Shift
Test distribution differs from training
Result: Model degrades in production
Solution: Continuous monitoring and retraining

5. Ignoring Class Imbalance
99% accuracy meaningless if 99% of data is one class
Solutions: Stratified sampling, weighted loss, SMOTE, use F1/AUC not accuracy

6. Not Using Baseline
Always compare to simple baseline:

- Random guessing
- Most frequent class
- Simple rules

If ML doesn’t beat baseline significantly, don’t use ML!

Most ML failures come from data problems, not algorithm choice - focus on data quality and proper evaluation

Real-World Applications

Supervised

Email Spam Filter

- Input: Email text
- Label: Spam or Not Spam
- Learn: Text patterns

Medical Diagnosis

- Input: Patient scans
- Label: Disease present
- Learn: Visual markers

Fraud Detection

- Input: Transaction data
- Label: Fraudulent or legitimate
- Learn: Fraud patterns (often semi-supervised)

Unsupervised

Customer Segmentation

- Input: Purchase behavior
- No labels needed
- Find: Customer groups

Anomaly Detection

- Input: System logs, sensor data
- No labels for anomalies
- Find: Unusual patterns

Topic Modeling

- Input: Document text
- No topic labels
- Find: Hidden themes

Reinforcement

Game AI (AlphaGo)

- State: Board position
- Action: Move choice
- Reward: Win or lose

Autonomous Vehicles

- State: Road conditions
- Action: Steering, speed
- Reward: Safe driving

Robot Control

- State: Robot position
- Action: Motor commands
- Reward: Task success

Each paradigm excels at different types of problems

Paradigm Comparison

Aspect	Supervised	Unsupervised	Reinforcement
Data	Labeled examples	Unlabeled data	Environment interaction
Feedback	Correct answers provided	No explicit feedback	Rewards/penalties
Goal	Predict outputs for new inputs	Discover structure in data	Learn optimal policy
Common Use	Classification, regression	Clustering, dimensionality reduction	Control, decision making
Challenges	Requires labeled data	Evaluation difficulty	Training complexity
Examples	Image recognition, spam detection	Customer segmentation, anomaly detection	Game AI, robotics

Choosing the Right Paradigm:

- **Supervised**: When you have labeled data and clear target outputs
- **Unsupervised**: When you want to explore data and find hidden patterns
- **Reinforcement**: When learning through sequential decisions and feedback

Modern AI often combines multiple paradigms for optimal results

Key Insights

Traditional vs ML
The fundamental shift:

- Traditional: **Rules first**, then apply to data
- ML: **Data first**, then learn rules

When Traditional Methods Work Best:

- Well-defined, stable rules
- High interpretability required
- Limited data available
- Regulatory compliance critical

The ML Advantage
Why ML excels:

- Handles complex, high-dimensional patterns
- Adapts to changing environments
- Scales with data availability
- Discovers non-obvious relationships

Critical Considerations:

- **Data quality ¿ quantity**: Garbage in, garbage out
- **Bias in data = bias in model**: Training data must be representative
- **Interpretability crisis**: Complex models are black boxes
- **Maintenance burden**: Models degrade, need monitoring and retraining
- **Cold start problem**: ML needs data to begin

Ethical and Practical Concerns:

- **Fairness**: Biased training data leads to discriminatory models (e.g., hiring, lending, criminal justice)
- **Privacy**: ML often requires large datasets - data governance critical
- **When NOT to use ML**: Simple rules work, need explainability, insufficient data, high-stakes decisions requiring

Summary

Machine Learning offers three powerful paradigms:

Supervised - Learn from labeled examples
Unsupervised - Discover hidden patterns
Reinforcement - Learn through interaction

Each excels at different types of problems
Choose based on your data, goals, and constraints