Introduction to Machine Learning

CMSC 173 - Machine Learning

Noel Jeffrey Pinton

October 17, 2025

Department of Computer Science
University of the Philippines - Cebu

Course Overview

What is Machine Learning?

What is Machine Learning?

Formal Definition

Machine Learning (ML) is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information.

Key Characteristics

- **Learning from data** without explicit programming
- **Improving performance** with experience
- **Discovering patterns** in complex datasets
- **Making predictions** or decisions

Traditional Programming vs ML

Traditional:
Rules + Data $\rightarrow$ Answers

Machine Learning:
Data + Answers $\rightarrow$ Rules

Core Insight

ML finds the rules automatically from examples!

Historical Context

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human." — Alan Turing

Major Milestones

- **1950s**: Alan Turing - "Can machines think?"
- **1957**: Perceptron (Frank Rosenblatt)
- **1986**: Backpropagation popularized
- **1990s**: Support Vector Machines
- **1997**: Deep Blue defeats Kasparov
- **2006**: Deep Learning renaissance
- **2012**: AlexNet wins ImageNet
- **2016**: AlphaGo defeats Lee Sedol
- **2020s**: Large Language Models

The Three AI Winters

Periods of reduced funding and interest:

- 1970s: Perceptron limitations
- 1987-1993: Expert systems fail
- Post-2000: AI hype deflation

Current Era

We're in the **Deep Learning Revolution**:

- Big data availability
- GPU acceleration
- Novel architectures (Transformers)

"Machine learning is the last invention that humanity will ever need to make." — Nick Bostrom

Computer Vision

- Medical image diagnosis
- Autonomous vehicles
- Facial recognition
- Object detection & tracking
- Image generation (DALL-E, Midjourney)

Natural Language Processing

- Machine translation

Other Domains

- **Finance**: Fraud detection, trading
- **Healthcare**: Drug discovery, medicine
- **E-commerce**: Recommendations
- **Gaming**: AI opponents
- **Manufacturing**: Quality control
- **Agriculture**: Crop monitoring

Impact

ML is transforming every industry!

Learning Objectives

By the end of this course, you will be able to:

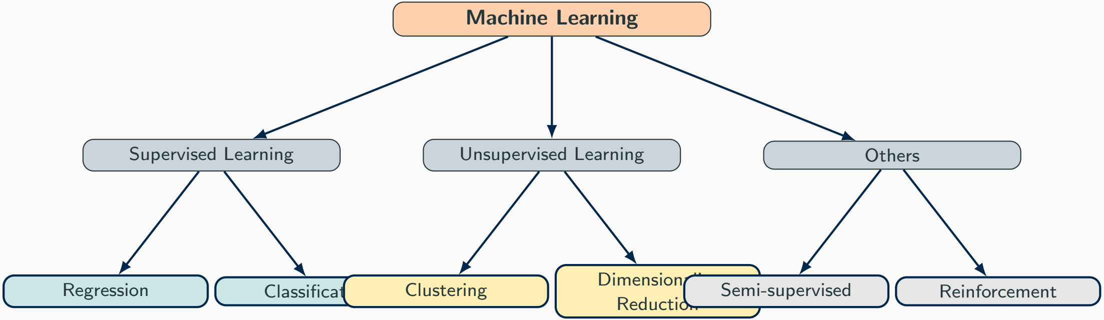
1. **Understand** the fundamental concepts and mathematical foundations of machine learning
2. **Distinguish** between different types of learning paradigms (supervised, unsupervised, etc.)
3. **Implement** core ML algorithms from scratch using Python
4. **Apply** appropriate ML techniques to real-world problems
5. **Evaluate** model performance using rigorous metrics
6. **Analyze** the theoretical properties of learning algorithms
7. **Compare** different approaches and select optimal methods
8. **Understand** state-of-the-art techniques in deep learning

Prerequisites

CMSC 170: Linear algebra, probability theory, calculus, Python programming

Types of Machine Learning

Machine Learning Taxonomy



Supervised Learning

"Learning is finding out what you already know. Doing is demonstrating that you know it." — Richard Bach

Definition

Learning from **labeled data** where each training example consists of:

- **Input**: Feature vector $\mathbf{x} \in \mathbb{R}^d$
- **Output**: Label/target y

Goal: Learn a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x}) \approx y$

Training Process

Given dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$:

1. Choose a hypothesis class $\mathcal{H}$
2. Define a loss function $\mathcal{L}(y, \hat{y})$

Key Properties

- **Labeled data required**
- **Teacher signal** guides learning
- **Generalization** to new examples
- **Performance measurable**

Two Main Tasks

1. Regression ($y \in \mathbb{R}$)
Predict continuous values

2. Classification ($y \in \{1, \ldots, K\}$)
Predict discrete categories

Regression: Predicting Continuous Values

"All models are wrong, but some are useful." — George Box

Problem Formulation

Input: $\mathbf{x} \in \mathbb{R}^d$ (features)
Output: $y \in \mathbb{R}$ (continuous target)
Model: $\hat{y} = f(\mathbf{x}; \theta)$

Common Loss Functions

Mean Squared Error (MSE):

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Mean Absolute Error (MAE):

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Regression Algorithms

- **Linear Regression**
- Ridge Regression (L2 regularization)
- Lasso Regression (L1 regularization)
- **Polynomial Regression**
- **Cubic Splines**
- Support Vector Regression (SVR)
- Decision Tree Regression
- Neural Networks

Real-World Examples

- House price prediction

Classification: Predicting Categories

"The goal is to turn data into information, and information into insight." — Carly Fiorina

Problem Formulation

Input: $\mathbf{x} \in \mathbb{R}^d$ (features)
Output: $y \in \{1, 2, \ldots, K\}$ (class label)
Model: $\hat{y} = \arg\max_k P(y = k|\mathbf{x})$

Types of Classification

Binary Classification: $K = 2$

- Spam vs. not spam
- Disease vs. healthy

Multi-class: $K > 2$

- Digit recognition (0-9)
- Animal species classification

Classification Algorithms

- **Logistic Regression**
- **Naïve Bayes**
- **K-Nearest Neighbors (KNN)**
- **Decision Trees**
- Support Vector Machines (SVM)
- Random Forests
- Gradient Boosting
- Neural Networks

Common Loss Functions

Cross-Entropy (log loss):

Unsupervised Learning

Unsupervised Learning

Definition

Learning from **unlabeled data** without explicit target outputs:

- **Input**: Feature vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- **Output**: None (discover structure)

Goal: Discover hidden patterns, structures, or relationships in data

Main Tasks

1. Clustering

- Group similar data points
- Algorithms: K-Means, DBSCAN, Hierarchical

2. Dimensionality Reduction

- Compress high-dimensional data
- Algorithms: PCA, t-SNE, UMAP

3. Density Estimation

- Model the data distribution

Key Characteristics

- **No labels** required
- **Exploratory** in nature
- **Structure discovery**
- Performance harder to measure

Applications

- Customer segmentation
- Anomaly detection
- Data visualization
- Feature extraction
- Compression
- Recommender systems

Challenge

How do we evaluate without labels?

Clustering: Grouping Similar Data

"Without data, you're just another person with an opinion." — W. Edwards Deming

Problem Formulation

Input: Dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
Output: Cluster assignments $\{c_1, \ldots, c_n\}$
Goal: Maximize intra-cluster similarity, minimize inter-cluster similarity

K-Means Algorithm

Objective: Minimize within-cluster variance

$$\min_{\{\mu_k\},\{c_i\}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mu_{c_i}\|^2$$

Algorithm:

Other Clustering Methods

Hierarchical Clustering:

- Agglomerative (bottom-up)
- Divisive (top-down)
- Creates dendrogram

DBSCAN:

- Density-based
- Finds arbitrary shapes
- Handles noise/outliers

Gaussian Mixture Models:

Dimensionality Reduction: Compression & Visualization

"In God we trust, all others must bring data." — W. Edwards Deming

Motivation

High-dimensional data ($d \gg 1$) causes:

- **Curse of dimensionality**
- Computational complexity
- Overfitting
- Difficulty in visualization

Solution: Project to lower dimensions while preserving structure

Principal Component Analysis (PCA)

Goal: Find directions of maximum variance

Other Techniques

Linear Methods:

- PCA (maximum variance)
- LDA (maximum discrimination)
- ICA (independent components)

Non-linear Methods:

- Kernel PCA
- t-SNE (visualization)
- UMAP (topology preservation)
- Autoencoders (neural networks)

Semi-Supervised Learning

Semi-Supervised Learning

Definition

Learning from **both labeled and unlabeled data**:

- **Labeled**: $\mathcal{D}_L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$
- **Unlabeled**: $\mathcal{D}_U = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$
- Typically $l \ll u$ (few labels, many unlabeled)

Goal: Leverage unlabeled data to improve performance

Fundamental Assumptions

1. **Smoothness Assumption**
   - Nearby points share same label
2. **Cluster Assumption**
   - Data forms discrete clusters
   - Points in same cluster have same label
3. **Manifold Assumption**
   - High-dim data lies on low-dim manifold

Common Approaches

Self-Training:

- Train on labeled data
- Predict unlabeled data
- Add confident predictions to training set
- Iterate

Co-Training:

- Multiple views of data
- Train separate classifiers
- Exchange confident predictions

Graph-Based Methods:

- Construct similarity graph
- Propagate labels

Why Semi-Supervised?

Labels are expensive! (Human annotation, expert knowledge, time)

Reinforcement Learning

"You can use a spoon to eat soup, but it's better to use a ladle. Learning is choosing the right tool." — Yann LeCun

Definition

Learning through **interaction with an environment**:

- **Agent** takes actions
- **Environment** provides states & rewards
- **Goal**: Maximize cumulative reward

Markov Decision Process (MDP)

Formal framework: $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$

- $\mathcal{S}$: State space
- $\mathcal{A}$: Action space
- $P(s'|s, a)$: Transition probabilities
- $R(s, a, s')$: Reward function

RL vs Other Paradigms

Key Differences:

- No direct supervision
- Delayed rewards
- Exploration vs exploitation
- Sequential decision making
- Trial and error learning

Classic Algorithms

- **Q-Learning**
- SARSA
- Policy Gradient

RL Example: Q-Learning

Q-Learning Algorithm

Goal: Learn optimal action-value function

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid s_0 = s, a_0 = a, \pi\right]$$

Update Rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

where:

- α: Learning rate
- r: Immediate reward
- s': Next state
- γ: Discount factor

Policy: $\pi(s) = \arg\max_a Q(s,a)$

Algorithm Pseudocode

```
Initialize Q(s, a) arbitrarily
for each episode do
    Initialize state s
    repeat
        Choose action a using ε-greedy policy
        Take action a, observe r, s'
        Q(s, a) ← Q(s, a) + α[r + γ max_a' Q(s', a') − Q(s, a)]
        s ← s'
    until s is terminal
end for
```

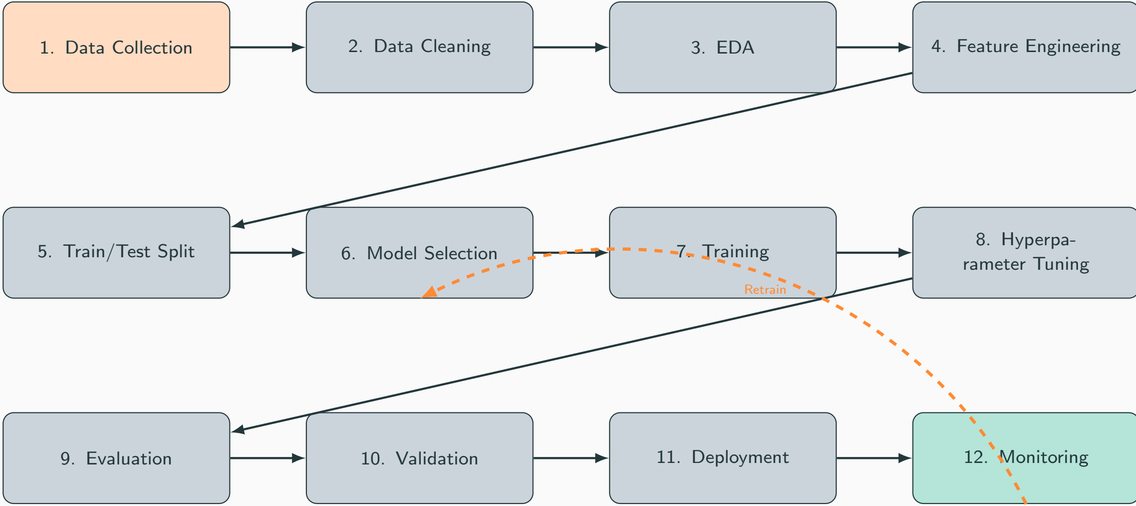
Key Concepts

Exploration vs Exploitation:

- ϵ-greedy: explore with probability ϵ
- Balances trying new actions vs using known good ones

The Machine Learning Pipeline

The ML Pipeline: From Data to Deployment



Key Insight

ML is iterative! Model performance informs feature engineering, data collection, etc.

"Garbage in, garbage out." — George Fuechsel

Data Cleaning

Common Issues:

- **Missing values**: Imputation, deletion
- **Outliers**: Detect and handle
- **Duplicates**: Remove
- **Inconsistencies**: Standardize formats
- **Noise**: Filter or smooth

Feature Scaling

Why? Many algorithms sensitive to feature scales

Standardization (Z-score):

Feature Engineering

Creating new features from existing ones:

- Polynomial features: x_1x_2, x_1^2
- Domain-specific transformations
- Binning continuous variables
- One-hot encoding categorical
- Date/time feature extraction
- Text vectorization (TF-IDF)

Train/Test Split

Why? Evaluate generalization

Model Selection & Training

Choosing a Model

Consider:

- **Problem type**: Regression, classification, etc.
- **Data size**: Deep learning needs more data
- **Interpretability**: Linear models vs black boxes
- **Training time**: Real-time vs offline
- **Prediction speed**: Production requirements

No Free Lunch Theorem

Theorem: No single algorithm works best for all problems

Implication: Must try multiple approaches and validate empirically

Start Simple!

1. Simple baseline (mean, majority class)
2. Linear model
3. More complex models
4. Ensemble methods

Training Process

Optimization: Minimize loss function

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D})$$

Common Optimizers:

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Adam (adaptive learning rate)
- RMSprop

Hyperparameter Tuning

Hyperparameters: Set before training

- Learning rate, regularization strength
- Number of layers, hidden units
- Tree depth, number of trees

Search Methods:

- Grid search
- Random search

Model Evaluation Metrics

Regression Metrics

Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root MSE (RMSE):

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

R-squared (coefficient of determination):

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Range: $(-\infty, 1]$, closer to 1 is better

Classification Metrics

Accuracy:

$$\text{Acc} = \frac{\text{correct predictions}}{\text{total predictions}}$$

Precision (positive predictive value):

$$\text{Prec} = \frac{TP}{TP + FP}$$

Recall (sensitivity, true positive rate):

$$\text{Rec} = \frac{TP}{TP + FN}$$

F1-Score (harmonic mean):

$$F_1 = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

ROC-AUC: Area under ROC curve

Important

Key Challenges in ML

Bias-Variance Tradeoff

Decomposition of Expected Error

For regression, expected test error:

$$\mathbb{E}[(y - \hat{f}(x))^2] = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

Bias: Error from wrong assumptions

$$\text{Bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$

Variance: Error from sensitivity to training set

$$\text{Var}[\hat{f}(x)] = \mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$$

Noise: Irreducible error σ^2

The Tradeoff

- **Simple models**: High bias, low variance
- **Complex models**: Low bias, high variance
- **Goal**: Find sweet spot!

Underfitting

Symptoms:

- High training error
- High test error
- Model too simple

Solutions:

- More features
- More complex model
- Less regularization

Overfitting

Symptoms:

- Low training error
- High test error
- Model too complex

Solutions:

- More training data

Regularization Techniques

L2 Regularization (Ridge)

Modified objective:

$$\min_{\theta} \mathcal{L}(\theta) + \lambda ||\theta||_2^2$$

where $\lambda > 0$ is regularization strength

Effect:

- Penalizes large weights
- Shrinks coefficients toward zero
- Improves generalization
- Handles multicollinearity

Closed-form solution (linear regression):

$$\hat{\theta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

L1 Regularization (Lasso)

Modified objective:

$$\min_{\theta} \mathcal{L}(\theta) + \lambda ||\theta||_1$$

Effect:

- Sparse solutions (some $\theta_i = 0$)
- Automatic feature selection
- More aggressive than L2

No closed-form: Use iterative methods

Elastic Net

Combines L1 and L2:

$$\min_{\theta} \mathcal{L}(\theta) + \lambda_1 ||\theta||_1 + \lambda_2 ||\theta||_2^2$$

Benefits:

- Sparsity from L1
- Stability from L2
- Best of both worlds

The Curse of Dimensionality

Problem Statement

As dimensionality d increases:

- **Volume grows exponentially**: $V \propto r^d$
- **Data becomes sparse**: Points far apart
- **Distance metrics break down**: All points equidistant
- **Overfitting risk increases**: More parameters to fit

Mathematical Insight

In high dimensions, volume concentrated in corners:

$$\frac{V_{\text{corners}}}{V_{\text{total}}} = 1 - \left(1 - \frac{1}{2^d}\right)^{2^d} \approx 1 - e^{-1}$$

For unit hypercube, most volume is near edges!

Data Requirements

To maintain density, need $n \propto c^d$ samples where $c > 1$

Solutions

1. Dimensionality Reduction

- PCA, t-SNE, UMAP
- Feature selection

2. Feature Selection

- Filter methods (correlation)
- Wrapper methods (RFE)
- Embedded (Lasso, trees)

3. Regularization

- L1/L2 penalties
- Early stopping

4. Collect More Data

- Exponentially more needed
- Often impractical

Rule of Thumb

$n > 10 \cdot d$ for reliable models

Course Structure

CMSC 173 Course Topics

Core Foundations

I. Overview (Today!)

- Learning paradigms
- Applications

II. Parameter Estimation

- Method of Moments
- Maximum Likelihood Estimation

III. Regression

- Linear Regression
- Lasso & Ridge
- Cubic Splines

IV. Model Selection

- Bias-Variance Decomposition
- Cross-Validation
- Regularization

Advanced Methods

V. Classification

- Logistic Regression, Naïve Bayes
- KNN, Decision Trees

VI. Kernel Methods

- Support Vector Machines
- Kernel trick

VII. Dimensionality Reduction

- Principal Component Analysis

VIII. Neural Networks

- Feedforward Networks
- CNNs, Transformers
- Generative Models

IX. Clustering

- K-Means, Hierarchical

Learning Resources

Recommended Textbooks

Primary:

- Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Supplementary:

- Hastie et al. (2009). *The Elements of Statistical Learning*. Springer.
- Goodfellow et al. (2016). *Deep Learning*. MIT Press.

Online Resources

- Scikit-learn documentation
- PyTorch/TensorFlow tutorials
- Coursera ML courses (Andrew Ng)
- Stanford CS229 lecture notes
- ArXiv.org for research papers

Tools We'll Use

- Python 3.8+
- NumPy, Pandas, Matplotlib
- Scikit-learn
- Jupyter Notebooks
- PyTorch (for deep learning)

Installation

Ensure you have Python and required packages installed before next session!

Best Practices in Machine Learning

"The best way to predict the future is to invent it." — Alan Kay

Development Workflow

1. **Start with baseline**
   - Simple model first
   - Establish minimum performance
2. **Iterate systematically**
   - Change one thing at a time
   - Track experiments
   - Version control (Git)
3. **Validate rigorously**
   - Cross-validation

Common Pitfalls to Avoid

- **Data leakage**: Test data in training
- **Ignoring class imbalance**
- **Not checking for overfitting**
- **Using wrong metrics**
- **Not scaling features**
- **Forgetting randomness**: Set seeds!
- **Over-engineering**: Keep it simple

Reproducibility

Essential for science:

Ethics & Responsible AI

"With great power comes great responsibility." — Stan Lee (adapted from Voltaire)

Ethical Considerations

Bias & Fairness:

- Training data may contain biases
- Models can amplify discrimination
- Ensure fairness across groups

Privacy:

- Protect sensitive information
- Anonymization techniques
- Comply with regulations (GDPR)

Transparency:

Societal Impact

Positive:

- Healthcare improvements
- Scientific discoveries
- Accessibility tools
- Environmental monitoring

Concerns:

- Job displacement
- Deepfakes & misinformation
- Surveillance
- Autonomous weapons

Summary

Key Takeaways

What We Covered Today

1. **Definition of Machine Learning**: Learning from data to improve performance
2. **Supervised Learning**: Regression & classification with labeled data
3. **Unsupervised Learning**: Clustering & dimensionality reduction
4. **Semi-Supervised Learning**: Leveraging both labeled & unlabeled data
5. **Reinforcement Learning**: Learning through interaction & rewards
6. **ML Pipeline**: From data collection to deployment
7. **Key Challenges**: Bias-variance tradeoff, overfitting, curse of dimensionality
8. **Best Practices**: Systematic development, validation, ethics

Next Lecture

Parameter Estimation: Method of Moments & Maximum Likelihood Estimation

Prepare for Next Session

Required Reading

Murphy (2022):

- Chapter 4: Statistics (4.1-4.3)
- Chapter 5: Decision Theory (5.1-5.2)

Bishop (2006):

- Chapter 1: Introduction (1.1-1.5)
- Chapter 2: Probability (2.1-2.3)

Practice Problems

1. Review probability theory
2. Linear algebra refresher
3. Set up Python environment
4. Install required packages

Questions to Ponder

1. When would you choose supervised vs unsupervised learning?
2. How do you decide on train/test split ratio?
3. What metrics are appropriate for imbalanced datasets?
4. How can we detect overfitting early?
5. What are ethical concerns in your domain of interest?

Office Hours

Available for questions and discussion after class or by appointment

Questions?

Thank you for your attention!

Next Lecture: Parameter Estimation
See you next time!