CS 6243, EE 4463, EE 5573
Machine Learning

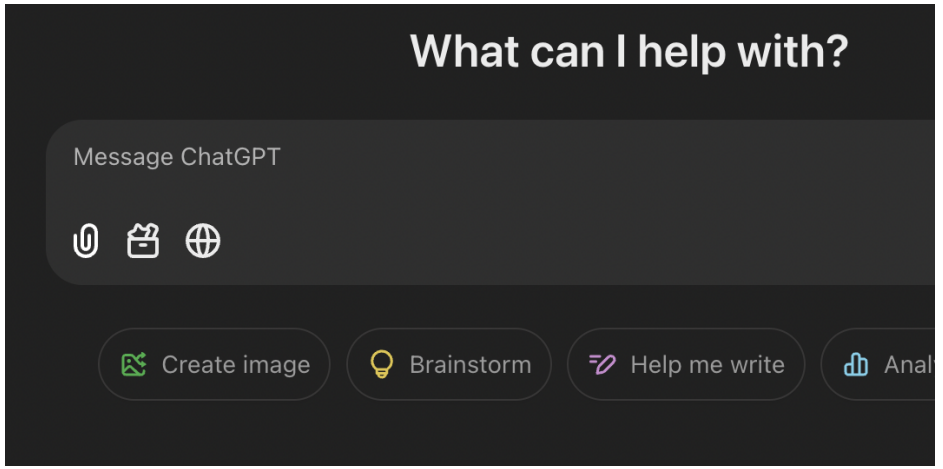
INTRODUCTION

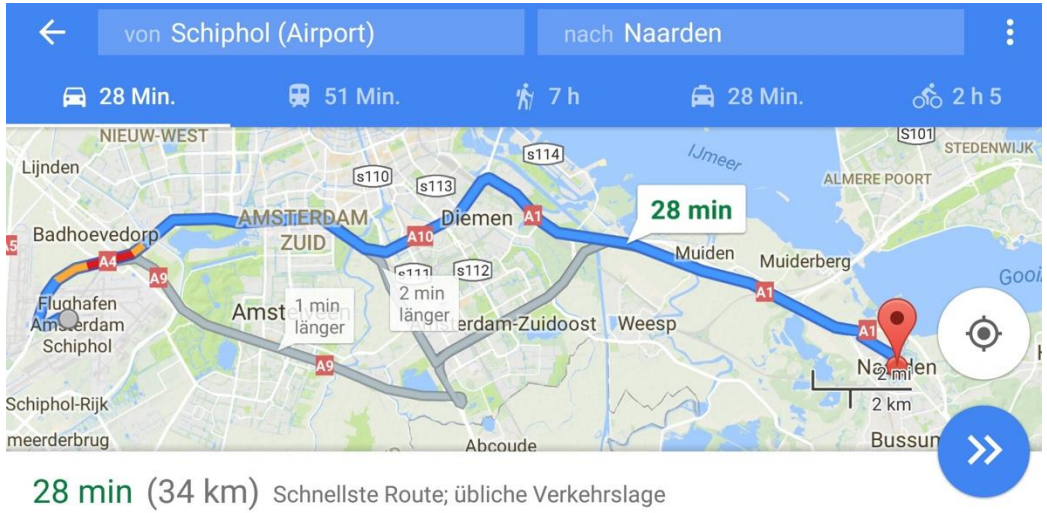
Dr. Panagiotis (Panos) Markopoulos
panos@utsa.edu

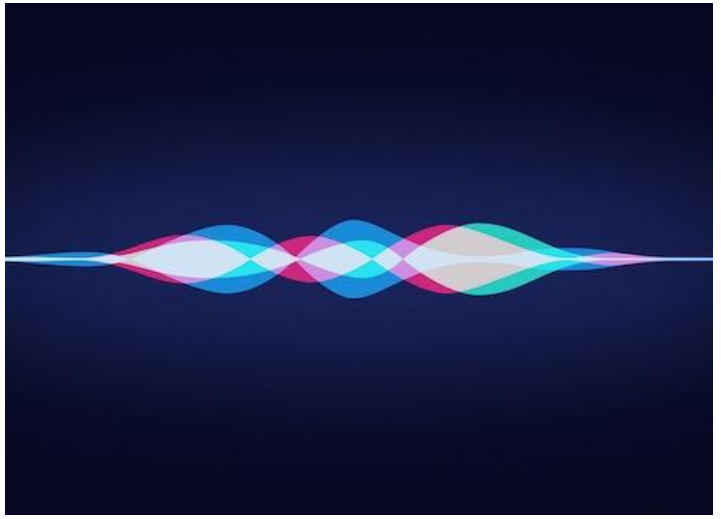
Artificial Intelligence

Today AI is Everywhere

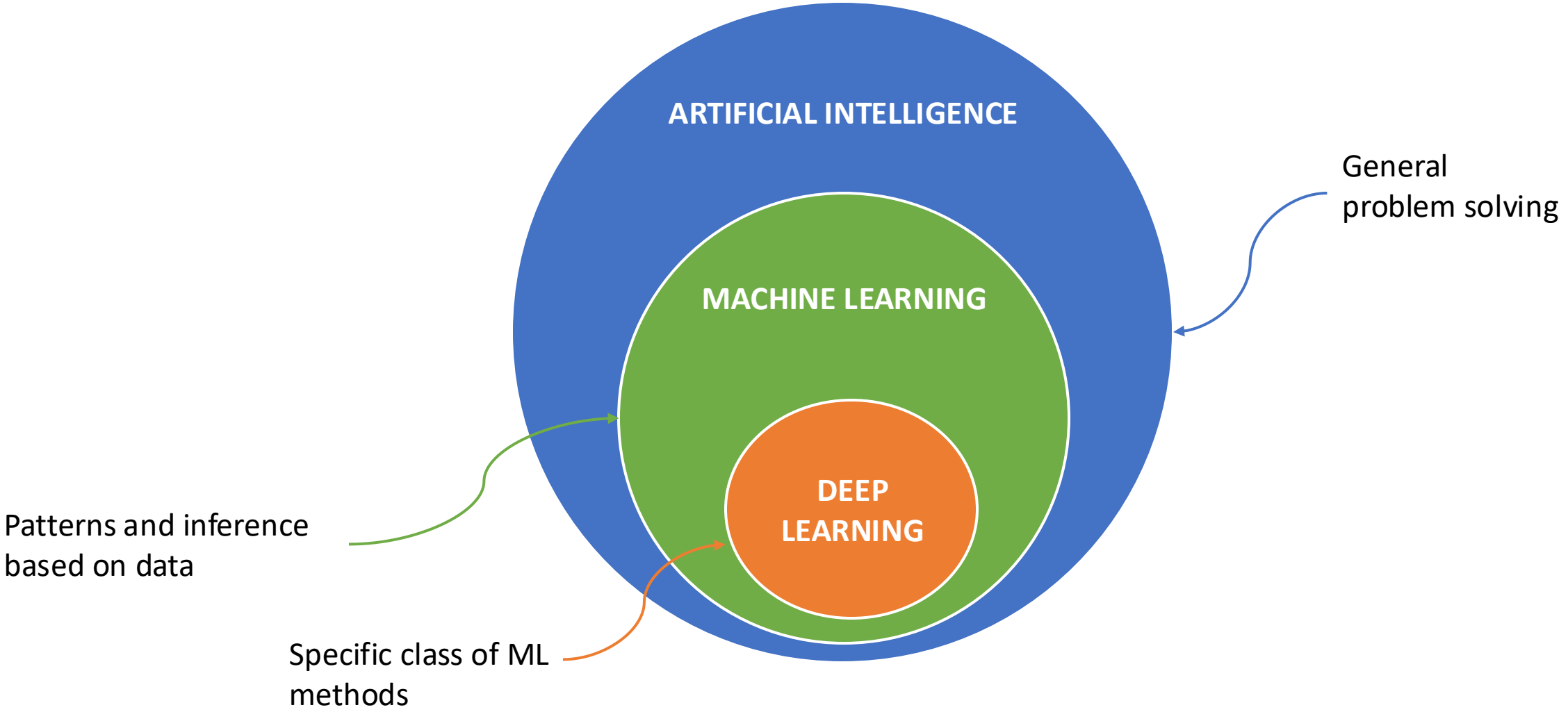
- ❑ Search
- ❑ Understand speech
- ❑ Create documents
- ❑ Autonomous Vehicles







AI > ML > DL


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
DEEP LEARNING
General problem solving
Patterns and inference based on data
Specific class of ML methods

Examples of AI that is not ML

Rule-Based Systems:
- Uses manually programmed rules to make decisions.

Symbolic AI:
- Based on symbolic reasoning and logic.
- Example: Chess-playing algorithms using exhaustive search and heuristics (e.g., Deep Blue).

Deterministic Pathfinding Algorithms:
- Finds optimal solutions using predefined rules.
- Example: Dijkstra's algorithm for shortest path in navigation systems.

Classical Planning Algorithms:
- Uses predefined rules to plan actions to achieve goals.
- Example: STRIPS (Stanford Research Institute Problem Solver).

Hardcoded Bots:
- Pre-programmed bots with no learning capability.
- Example: Early chatbot systems.

Constraint-Satisfaction Problems (CSPs):
- Solves problems based on predefined constraints and logic.
- Example: Sudoku solvers using backtracking algorithms.

When to Use Traditional (Non-Deep) ML?

Deep learning relies on copious parameters to express extremely complex data models.

However:

Not all data models are complex: Some applications involve simple or even linear relationships between inputs and outputs. Examples: Predictive maintenance in energy systems, basic economic forecasting, or linear dose-response relationships in healthcare.

Deep learning requires massive datasets: The large number of parameters necessitates exponentially larger amounts of data to prevent overfitting. Many critical applications lack such data volume. Examples: Medical research with rare diseases, niche defense systems, or early-stage material discovery.

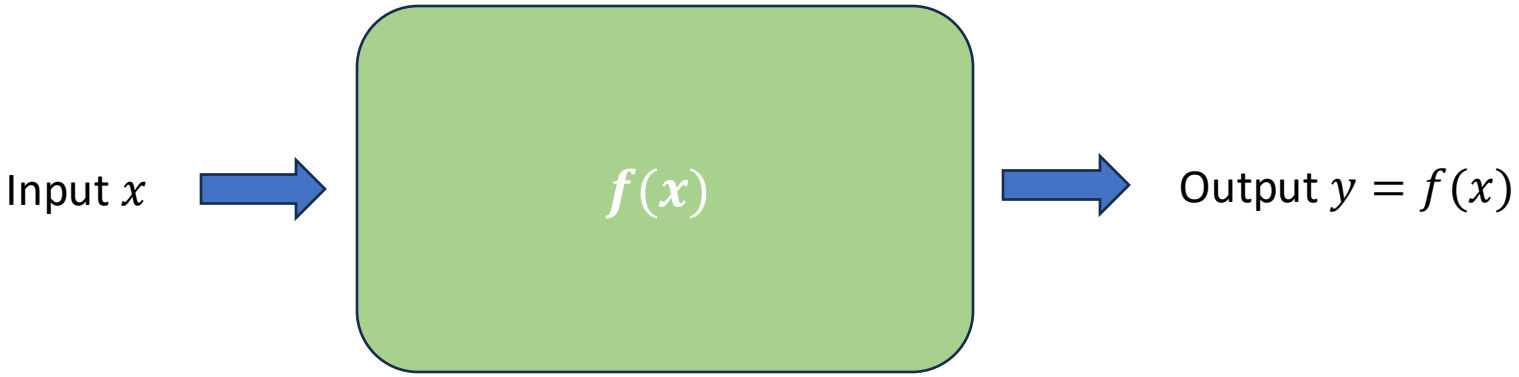
Overkill for simpler tasks: When simpler models can achieve comparable results with greater interpretability and efficiency. Examples: Fraud detection (finance), anomaly detection (energy), or small-scale clinical trials.

In these cases, traditional ML methods outperform deep learning in both practicality and effectiveness.

What is ML?

ML is a **subset of AI** that focuses on **algorithms and statistical models** that enable machines to learn patterns from data and make predictions or decisions without being explicitly programmed for specific tasks.

Supervised Machine Learning



Big Picture:

- $f(x)$ is unknown and sought-after.
- ML model typically estimates or mimics $f(x)$, given experience.

Broad Classification of ML Problems

- Based on the type of examples/experience:
  - Supervised, Unsupervised, Semi-supervised
  - Reinforcement Learning
- Based on the type of output/task:
  - Regression, Classification
- Based on the type of model:
  - Generative, Discriminative
- Based on structure of assumed f:
  - Linear; Non-linear, including neural nets etc.

Some Daily Applications

- **Personalized Recommendations**: Netflix, Spotify, and Amazon use ML to recommend movies, music, and products based on user behavior.
- **Smart Assistants**: Alexa, Siri, and Google Assistant rely on natural language processing (NLP) for voice recognition and response generation.
- **Spam Detection**: Email services like Gmail employ ML models to filter spam and phishing emails.
- **Facial Recognition**: Used in smartphones (Face ID) and social media platforms for tagging or security purposes.
- **Navigation**: Google Maps and Waze leverage ML to optimize routes, predict traffic, and suggest alternative paths.
- **Generative AI**: Tools like ChatGPT, Jasper, and Canva AI assist in daily content generation, writing, and communication tasks.

Some Applications (cont'd)

- **Fraud Detection**: Banks utilize ML to identify unusual patterns in transactions for credit card fraud or money laundering.
- **Healthcare Diagnostics**: ML algorithms support early diagnosis of diseases such as diabetes, cancer, and retinal disorders through image analysis.
- **Predictive Maintenance**: Industries use ML to predict equipment failure in manufacturing, reducing downtime and maintenance costs.
- **Customer Sentiment Analysis**: NLP models analyze reviews, social media posts, and feedback to gauge customer sentiment.
- **Dynamic Pricing**: E-commerce and airlines adjust prices in real-time based on demand and market trends using ML.
- **Drug Discovery**: ML models like AlphaFold predict protein folding, accelerating drug discovery and personalized medicine.
- **Cybersecurity**: Advanced anomaly detection models secure systems against evolving cyber threats, including zero-day exploits.
- **Digital Twins**: ML powers virtual replicas of physical systems for optimization in fields like aerospace and smart cities.

Some ML Challenges

- **Data Quality and Availability**: High-quality, unbiased, and labeled data is critical for reliable ML models.
- **Bias and Fairness**: Ensuring models and data do not perpetuate or amplify societal biases.
- **Model Interpretability**: Creating explainable models for trust in high-stakes fields like healthcare and finance.
- **Privacy Concerns**: Safeguarding sensitive data while complying with privacy regulations (e.g., GDPR, HIPAA).
- **Scalability**: Designing algorithms to handle massive datasets and real-time processing demands.
- **Energy Efficiency**: Reducing the computational and environmental costs of large-scale model training and deployment.
- **Robustness to Adversarial Attacks**: Building models resilient to deliberate inputs designed to mislead.
- **Domain Generalization**: Ensuring models perform well on new or shifted data distributions beyond training datasets.
- **Ethical Use of Generative AI**: Preventing misuse of tools like deepfakes and synthetic media for harm.

Supervised Learning

Definition: Learns a mapping function from labeled input-output pairs to predict outputs for unseen inputs.

Structure: Data is organized with corresponding inputs and outputs.

Process: The model identifies patterns and relationships between inputs and outputs.

Nature: Predictive.

Types of Problems: Classification, Regression

Examples: Spam email detection, Face recognition, Medical diagnosis

Unsupervised Learning

Definition: Learns patterns and structures from unlabeled data without predefined outputs

Structure: Data is provided without corresponding outputs

Process: The model identifies inherent relationships, clusters, or structures within the data

Nature: Descriptive or exploratory

Types of Problems: E.g., Clustering, dimensionality reduction, denoising, feature extraction

Examples: Customer clustering, anomaly detection, stock data compression

Difference between supervised and unsupervised: Supervised learning uses labeled data to map inputs to outputs for prediction; unsupervised learning uses unlabeled data to discover patterns or structures without predefined outputs.

Semi-Supervised Learning

Definition: Learns through trial and error by interacting with an environment to maximize cumulative rewards.

Structure: No labeled input-output pairs; the agent learns by receiving feedback (rewards or penalties) based on its actions.

Process: The model explores actions, evaluates outcomes, and adjusts strategies to optimize long-term rewards.

Nature: Goal-oriented and adaptive.

Types of Problems: Sequential decision-making, Control optimization, Game strategy

Examples: Game-playing AI (e.g., AlphaGo), Robotics for task automation, Dynamic pricing and bidding systems

Reinforcement Learning

Definition: Combines a small amount of labeled data with a large amount of unlabeled data to improve learning accuracy.

Structure: Data includes both labeled and unlabeled examples.

Process: The model leverages labeled data to infer patterns and extend learning to unlabeled data.

Nature: Predictive and exploratory.

Types of Problems: Classification, Regression

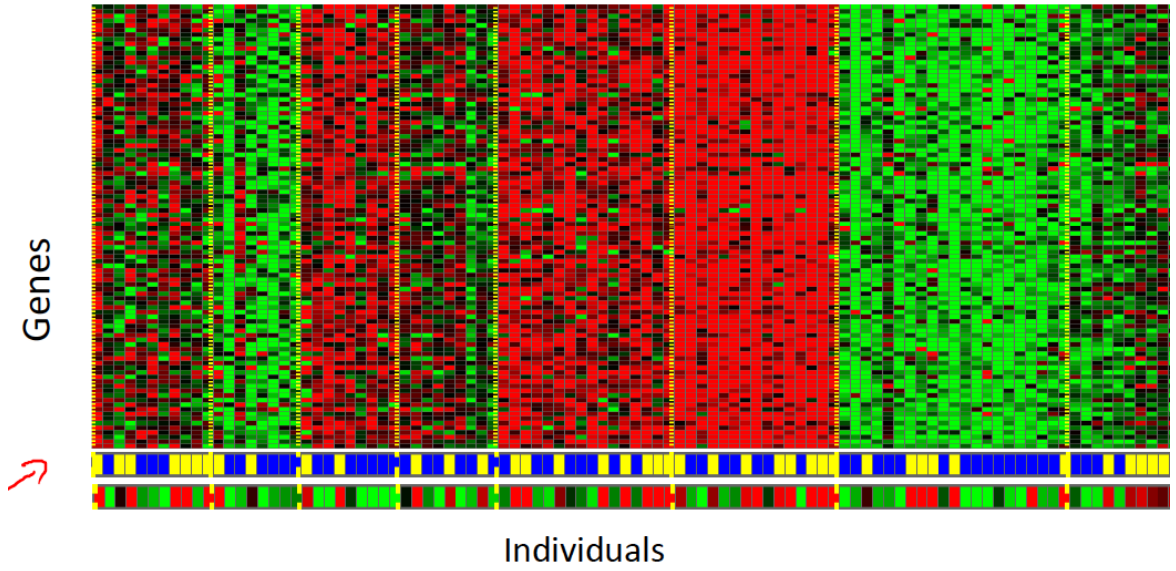
Examples: Text classification with limited labeled documents, Image recognition with partially labeled datasets, Medical diagnosis with a small number of annotated cases

Difference with supervised learning: RL optimizes rewards through interaction with an environment, using delayed feedback; SL minimizes a loss function using immediate feedback from labeled input-output pairs (outputs are used to form a loss that serves as immediate feedback)

SL: Classification Problems

- **Input**: Any type of data.
- **Output**: A discrete class label.
- **Engineer's Role:**
  - Design model structure and define learnable parameters.
  - Determine the algorithm for training the model's parameters.
  - Provide labeled input-output examples (training data).
- **Machine's Role:**
  - Use the provided data and algorithm to train model parameters.
- **Goal**: For unseen data, predict outputs from inputs as accurately as possible.
- **Modes of Operation**:
  - Training: Learn parameters from labeled data.
  - Testing: Evaluate performance on unseen data.

Examples



Classify a tissue sample into one of several cancer classes, based on a gene expression profile.

Handwritten digit classification.

SL: Regression Problems

- **Input**: Any type of data.
- **Output**: A continuous value.
- **Engineer's Role:**
  - Design model structure and define learnable parameters.
  - Determine the algorithm for training the model's parameters.
  - Provide labeled input-output examples (training data).
- **Machine's Role:**
  - Train model parameters using examples and the prescribed algorithm.
- **Goal**: Accurately predict continuous values for unseen inputs.
- **Modes of Operation**:
  - **Training**: Learn parameters from labeled data.
  - **Testing**: Evaluate performance on unseen data.

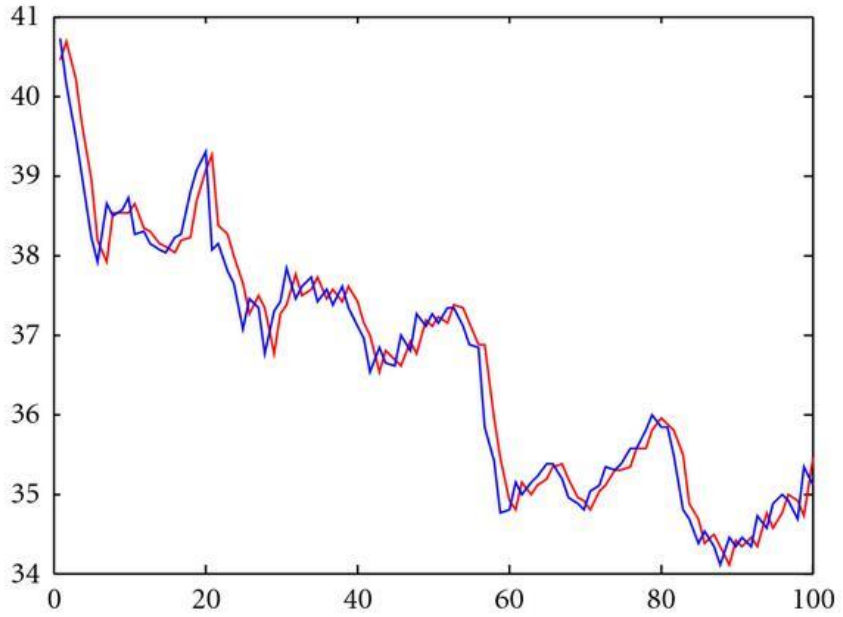
SL: Regression Problems

- **Input**: Any type of data.
- **Output**: A continuous value.
- **Engineer's Role:**
  - Design model structure and define learnable parameters.
  - Determine the algorithm for training the model's parameters.
  - Provide labeled input-output examples (training data).
- **Machine's Role:**
  - Train model parameters using examples and the prescribed algorithm.
- **Goal**: Accurately predict continuous values for unseen inputs.
- **Modes of Operation**:
  - **Training**: Learn parameters from labeled data.
  - **Testing**: Evaluate performance on unseen data.

SL: Classification vs Regression

- **Output:**
  - **Classification**: Predicts discrete (categorical) values with a limited number of possible outputs.
  - **Regression**: Predicts continuous values, though practically discrete due to finite numerical precision, with a larger range and higher resolution.
- **Key Insight**: Continuous values do not exist in computation; all outputs are discrete due to the limits of digital precision. Regression treats outputs as continuous within numerical precision. Classification: Operates on far fewer output options, often categorical rather than numerical.
- **Loss Function:** Different loss functions are required due to differences in output types (e.g., categorical for classification, numerical for regression).

Example

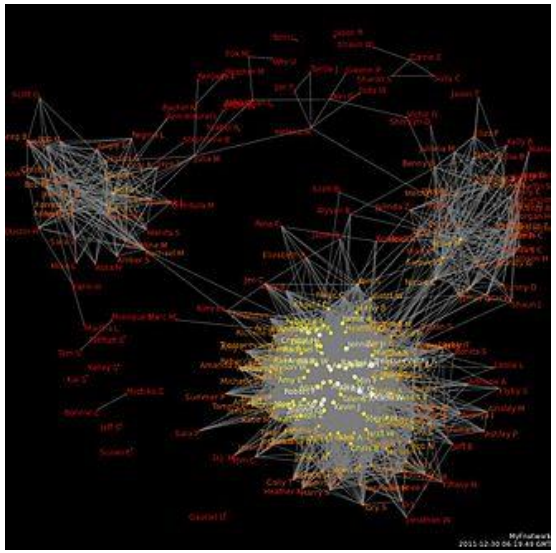


Stock value prediction, house price prediction, etc.

UL: Clustering Problems

- **Input**: Any data collection.
- **Output**: Groups (clusters) of similar data points.
- **Engineer's Role:**
  - Define a measure of similarity between data points (e.g., Euclidean distance).
  - Select the algorithm for clustering (e.g., k-means, hierarchical).
- **Machine's Role:** Apply the algorithm to the data and form clusters based on the similarity measure.
- **Goal**: Group data points into clusters that maximize similarity within groups and minimize similarity across groups.

Examples



Social network analysis.

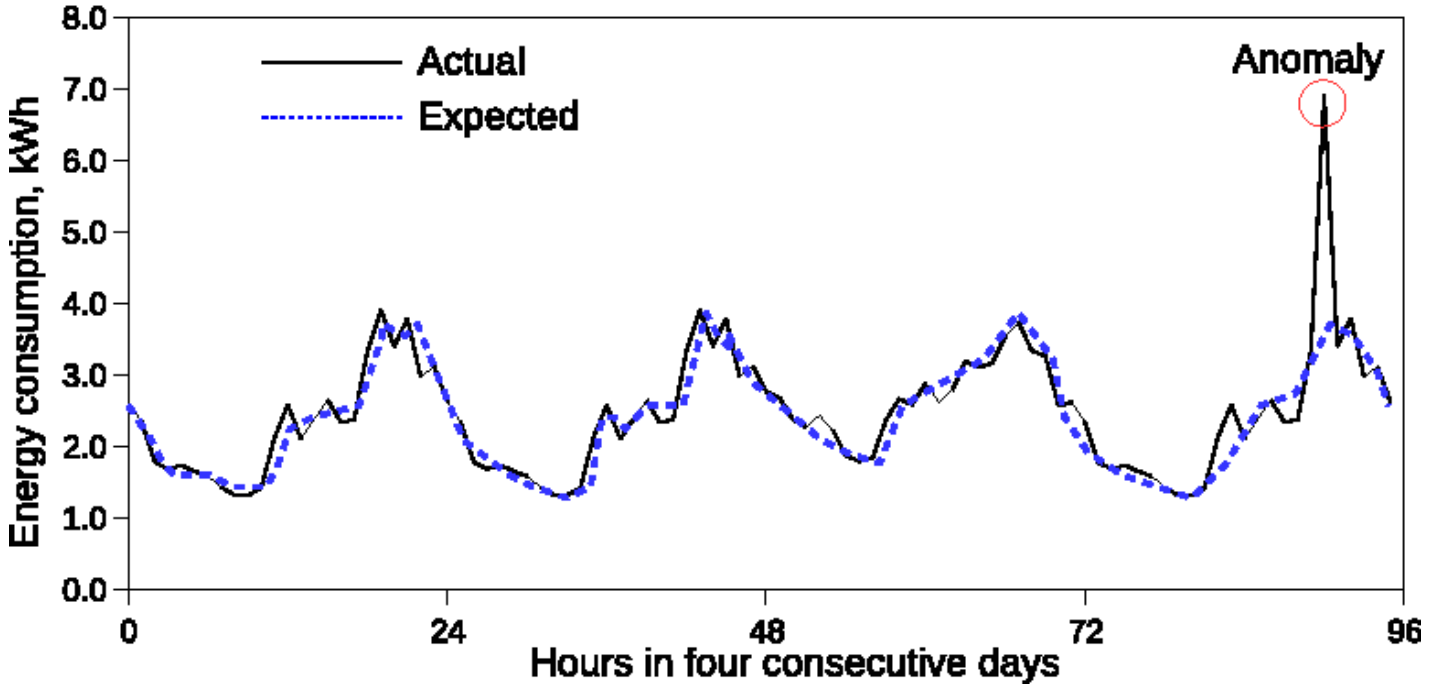
Image segmentation and for vegetation identification.

Organize computer clusters.

UL: Anomaly Detection

- **Input**: Any data collection.
- **Output**: Data points that deviate significantly from the norm (anomalies).
- **Engineer's Role:**
  - Define measures of conformity to identify normal data points.
  - Select the algorithm to test conformity and detect anomalies.
- **Machine's Role:** Apply the algorithm to the data and flag anomalies based on the conformity measures.
- **Goal**: Identify and isolate data points that differ significantly from expected patterns.

Examples



Anomaly detection in power grid data.

https://www.semanticscholar.org/paper/Regression-based-Online-Anomaly-Detection-for-Smart-Liu-Nielsen/ea729aa7fec8c6808b8eef0e862a29b9ebe8a2da

RL

- **Input**: An environment with states and possible actions.
- **Output**: A policy (strategy) that maximizes cumulative rewards over time.
- **Engineer's Role:**
  - Define the environment, state space, actions, and reward structure.
  - Select the algorithm to optimize the agent's decision-making (e.g., Q-learning, policy gradient).
- **Machine's Role:** Interact with the environment, learn from rewards/penalties, and refine the policy to maximize long-term rewards.
- **Goal**: Enable the agent to make optimal decisions in an environment through exploration and learning.

This Course

Scope of course:

- **Foundations and Core Approaches and Methods of ML**

Course Objectives:

- Solidify mathematical foundations of ML.
- Understand the principles, problems, and inner workings of ML methods.
- Implement ML methods and conduct numerical experiments.
- Test and compare between ML configurations.
- Identify challenges and potential solutions.

Our Scope

Our Scope

STATE-OF-THE-ART DEEP LEARNING

- Advanced frameworks: TensorFlow, PyTorch, Scikit-learn, specialized libraries.
- Latest advancements in deep learning: Transformer models, GANs, diffusion models.
- Scalable and data-intensive methodologies for large-scale applications.
- Custom solutions tailored for domain-specific and industrial applications.

Covered by courses: CS 6283 Deep Learning, CS 6313 Deep Reinforcement Learning, EE 5553 Deep Learning

CORE MACHINE LEARNING APPROACHES AND METHODS

- Supervised Learning: Regression, classification.
- Unsupervised Learning: Clustering, dimensionality reduction, anomaly detection.
- Reinforcement Learning: Policy optimization, reward-driven learning, game theory.
- Model Evaluation and Generalization: Metrics, cross-validation, regularization techniques.

FOUNDATIONS

- Mathematical Foundations: Linear Algebra, Statistics and Probability, Multivariate Calculus, Numerical and Stochastic Optimization Methods
- Programming Foundations: Algorithmic thinking and problem-solving, Python, libraries (NumPy, Pandas), and basic programming principles

Necessary Background

The following background is necessary for this course:

- **Linear algebra (intermediate)**
  - In particular: matrix and vector operations (e.g., addition, multiplication, inner/outer product product), inversion, decompositions, trace, norms
- **Calculus (intermediate)**
  - In particular: derivatives
- **Probability and Random Variables (intermediate-strong)**
  - In particular: probability theory, Bayes, random variable distributions (Gaussian, uniform) and sample statistics (mean, variance, covariance, correlation, independence)
- **Programming (basic)**
  - Ideally python

<u>If you lack this background, the course will be hard to follow.</u>

If you have questions/doubts, please contact me.

Refresh your Memory

Examples of material you can study to refresh your memory. You can use any other source of your choice.

Linear Algebra:

- Introduction to Applied Linear Algebra, S. Boyd, L. Vandenberghe, https://web.stanford.edu/~boyd/vmls/
- Chapters 1, 3, 5, 6, 8, 10, 11.1-2

Probability:

- Introduction to Probability for Data Science, S. H. Chan, https://probability4datascience.com/
- Chapters 1-6.

Calculus:

- Convex Optimization, S. P. Boyd, L. Vandenberghe, https://stanford.edu/~boyd/cvxbook/
- Chapter A.4

Programming:

- Python (https://wiki.python.org/moin/BeginnersGuide); numpy (https://numpy.org/); pandas (https://pandas.pydata.org/)

Related Announcements

Research Experiences for Undergraduates (REU) Course: EE-4913 Signal Processing and ML

NSF-Sponsored AI Spring School: https://ai.utsa.edu/ai-spring-school-2025/