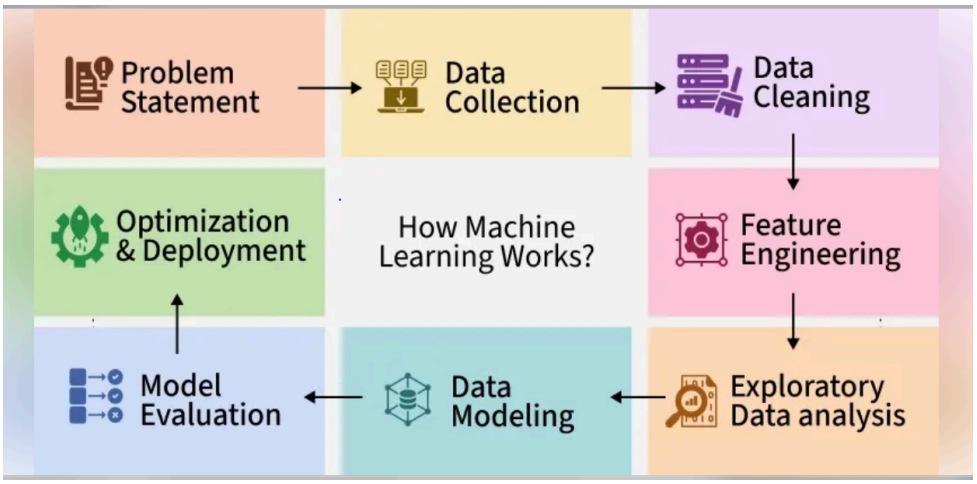
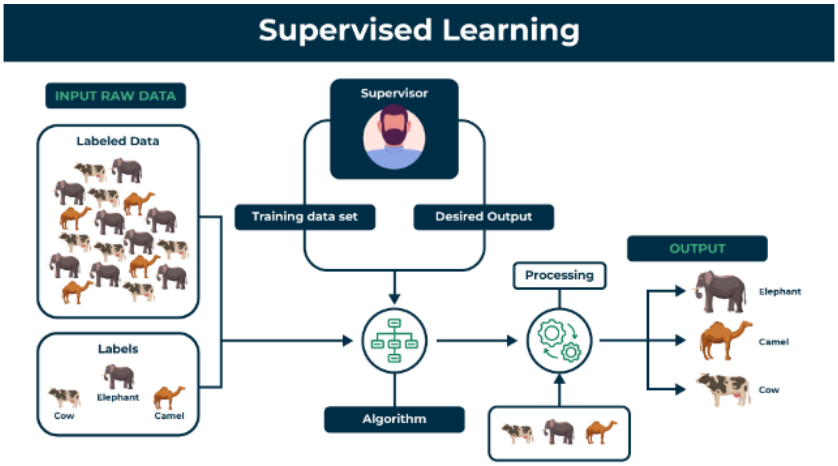
Machine Learning (ML)

Machine Learning (ML) is a branch of artificial intelligence (AI) that focuses on building algorithms and models that allow systems to learn from data and improve over time.



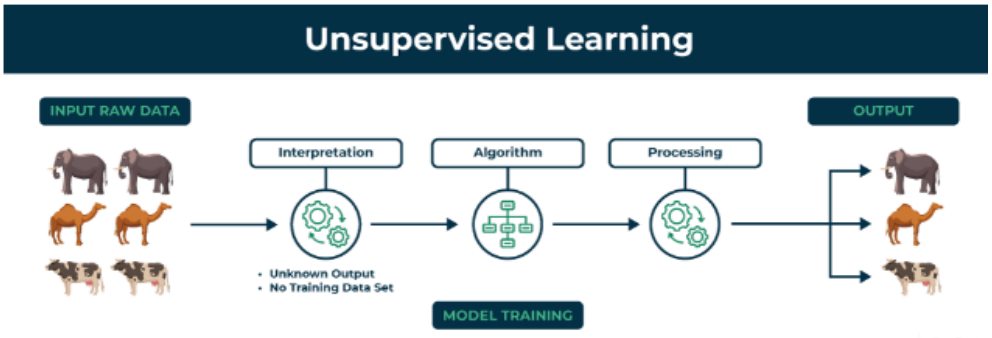
1. Supervised Learning



- The model is trained using labeled data (input-output pairs), and the goal is to learn the relationship between them
- Examples:
    - **Classification**: Spam email detection (inputs = emails, outputs = spam or not spam).

    - **Regression**: Predicting house prices (inputs = features like area, location, etc., outputs = price).
- **Common Algorithms**:
    - **Linear Regression** (for regression tasks)
    - **Logistic Regression** (for binary classification)
    - **Support Vector Machines (SVM)**

2. Unsupervised Learning



What it is: The model tries to find hidden patterns or groupings in data without labeled outputs.

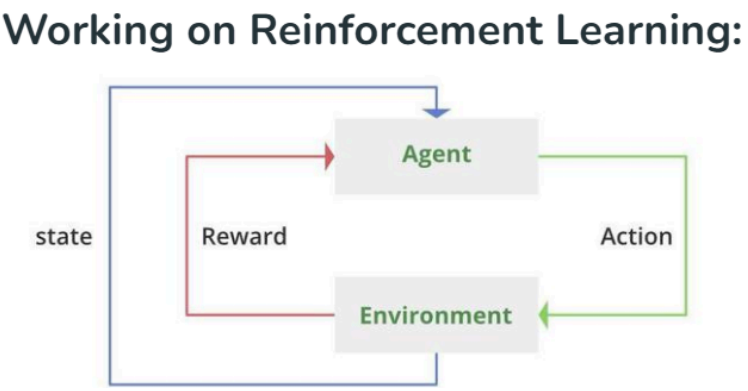
Examples:

- **Clustering**: Grouping customers by purchasing behavior (e.g., customer segmentation).
- **Dimensionality Reduction**: Reducing the number of features in a dataset while preserving its structure (e.g., PCA for feature reduction).

Common Algorithms:

- **K-Means Clustering**
- **Principal Component Analysis (PCA)**

3. Reinforcement Learning



The model (called an **agent**) learns by **interacting with an environment** and receiving **rewards or penalties** based on its actions. The goal is to learn a **strategy (policy)** that maximizes cumulative reward over time.

Examples:

- **Game AI**:
  In AlphaGo, the agent learned to play the board game Go by playing millions of games and receiving feedback (win/loss).
  (Inputs = game state, Outputs = move to make)

- **Self-Driving Cars**:
  The car learns how to drive by receiving rewards for safe driving actions (e.g., staying in lane, stopping at red lights) and penalties for risky behaviors (e.g., collisions).
  (Inputs = sensor data, Outputs = steering, braking, accelerating)

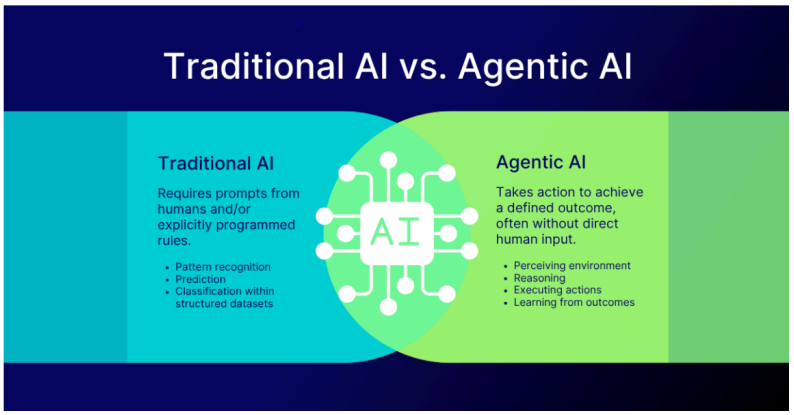
Key Components:

- **Agent**: The learner/decision-maker.
- **Environment**: The world the agent interacts with.
- **State**: The current situation the agent is in.
- **Action**: What the agent can do.
- **Reward**: Feedback signal (positive or negative) received after an action.

Common Algorithms:

- **Q-Learning**: Learns the value of taking a certain action in a given state.
- **Deep Q-Networks (DQN)**: Combines Q-Learning with deep neural networks for complex problems.
- **Policy Gradient Methods**: Directly learn the best policy instead of value functions.

Unlike traditional AI, which typically operates within predefined parameters and requires constant human intervention, agentic AI is characterized by its ability to pursue goals autonomously, make independent decisions, and adapt to dynamic environments. This distinction is crucial, as it shifts the focus from merely processing information to actively solving problems and achieving objectives in the real world.



🤖 Agentic AI vs Traditional AI

Aspect	Traditional AI	Agentic AI
Behavior	Reactive and task-specific	Proactive, goal-driven, and self-directed
Autonomy	Low — waits for commands	High — initiates actions independently
Adaptability	Limited to training scope	Adapts to dynamic goals and changing environments
Examples	Classification, prediction, translation, etc.	AI assistants that plan, decide, and act across tasks
Tools vs Agents	Tool-like: serves one function	Agent-like: acts with intent to achieve goals
Learning	Often static (trained once, used repeatedly)	Often continual, learns from interactions