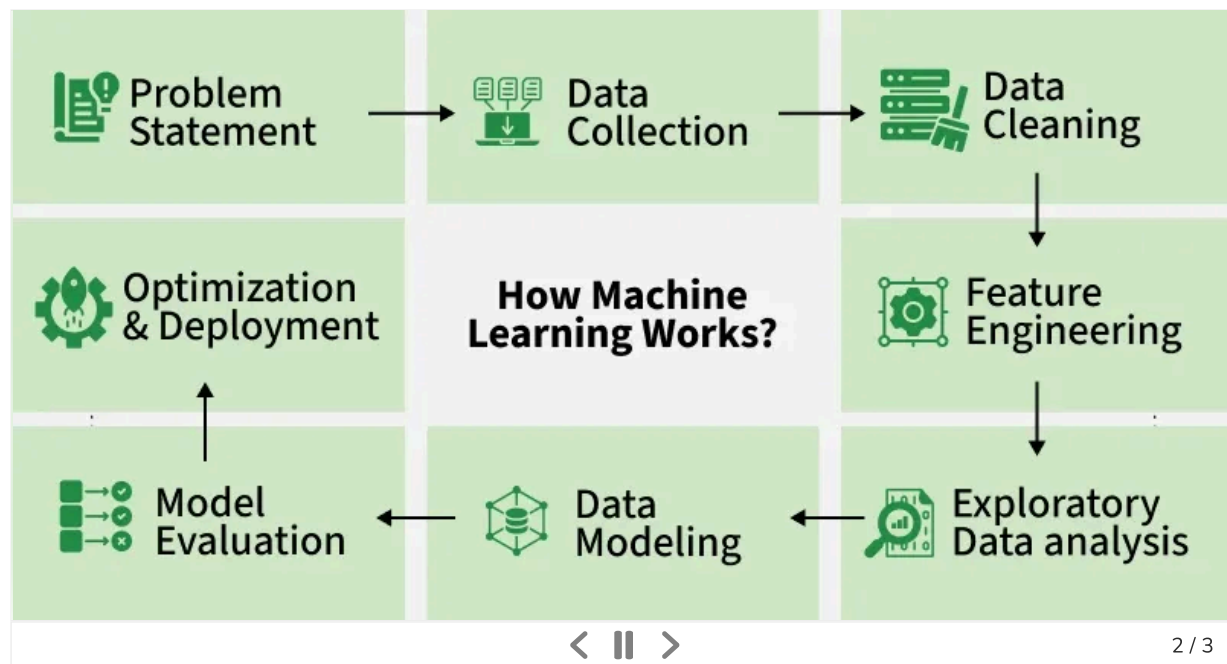




Machine Learning Tutorial

Last Updated : 26 Dec, 2025

Machine learning is a branch of Artificial Intelligence that focuses on developing models and algorithms that let computers learn from data without being explicitly programmed for every task. In simple words, ML teaches systems to think and understand like humans by learning from the data.



Machine Learning is mainly divided into three core types: Supervised, Unsupervised and Reinforcement Learning along with two additional types, Semi-Supervised and Self-Supervised Learning.

- **Supervised Learning:** Trains models on labeled data to predict or classify new, unseen data.
- **Unsupervised Learning:** Finds patterns or groups in unlabeled data, like clustering or dimensionality reduction.
- **Reinforcement Learning:** Learns through trial and error to maximize rewards, ideal for decision-making tasks.

Note: The following are not part of the original three core types of ML, but they have become increasingly important in real-world applications, especially in deep learning.

Additional Types:

- **Self-Supervised Learning:** Self-supervised learning is often considered as a subset of unsupervised learning, but it has grown into its own field due to its success in training large-scale models. It generates its own labels from the data, without any manual labeling.

- ***Semi-Supervised Learning:*** *This approach combines a small amount of labeled data with a large amount of unlabeled data. It's useful when labeling data is expensive or time-consuming.*

Module 1: Machine Learning Pipeline

This section covers preprocessing, exploratory data analysis and model evaluation to prepare data, uncover insights and build reliable models.

1. Data Preprocessing

- [ML workflow](#)
- [Data Cleaning](#)
- [Data Preprocessing in Python](#)
- [Feature Scaling](#)
- [Feature Extraction](#)
- [Feature Engineering](#)
- [Feature Selection Techniques](#)

2. Exploratory Data Analysis

- [Exploratory Data Analysis](#)
- [Exploratory Data Analysis in Python](#)
- [Advance EDA](#)
- [Time Series Data Visualization](#)

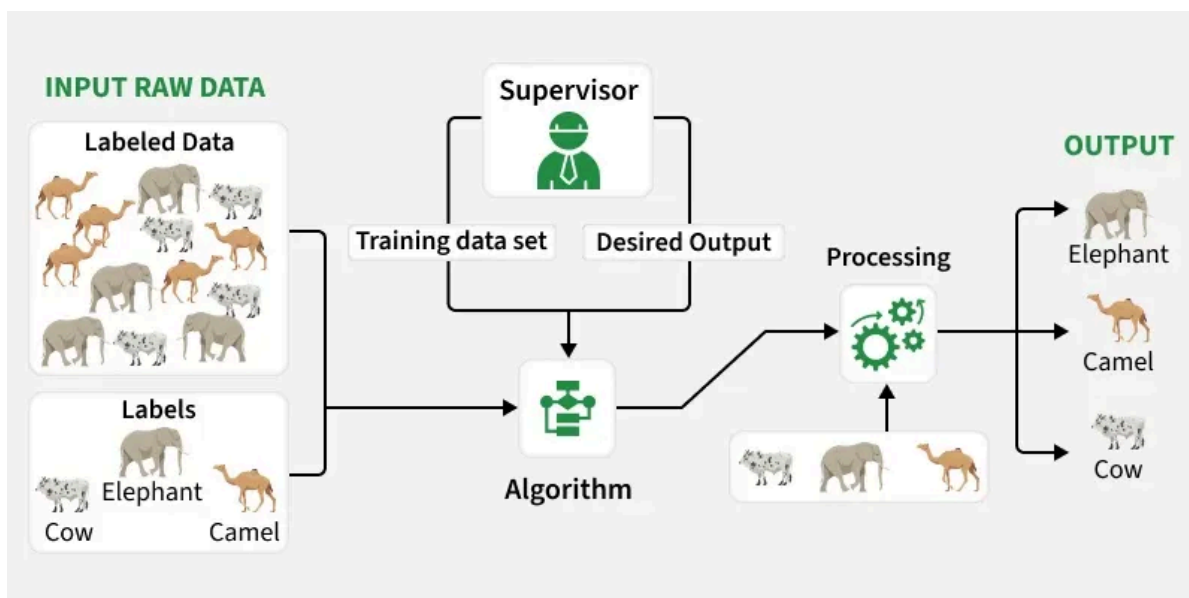
3. Model Evaluation

- [Regularization in Machine Learning](#)
- [Confusion Matrix](#)
- [Precision, Recall and F1-Score](#)
- [AUC-ROC Curve](#)
- [Cross-validation](#)
- [Hyperparameter Tuning](#)

Module 2: Supervised Learning

Supervised learning algorithms are generally categorized into **two main types**:

- **[Classification](#)** - where the goal is to predict discrete labels or categories
- **[Regression](#)** - where the aim is to predict continuous numerical values.



Supervised Learning

There are many algorithms used in supervised learning each suited to different types of problems. Some of the most commonly used supervised learning algorithms are:

1. Linear Regression

This is one of the simplest ways to predict numbers using a straight line. It helps find the relationship between input and output.

- [Introduction to Linear Regression](#)
- [Gradient Descent in Linear Regression](#)
- [Multiple Linear Regression](#)

2. Logistic Regression

Used when the output is a "yes or no" type answer. It helps in predicting categories like pass/fail or spam/not spam.

- [Understanding Logistic Regression](#)
- [Cost function in Logistic Regression](#)

3. Decision Trees

A model that makes decisions by asking a series of simple questions, like a flowchart. Easy to understand and use.

- [Decision Tree in Machine Learning](#)
- [Types of Decision tree algorithms](#)
- [Decision Tree - Regression \(Implementation\)](#)
- [Decision tree - Classification \(Implementation\)](#)

4. Support Vector Machines (SVM)

A bit more advanced—it tries to draw the best line (or boundary) to separate different categories of data.

- [Understanding SVMs](#)
- [SVM Hyperparameter Tuning - GridSearchCV](#)
- [Non-Linear SVM](#)

5. k-Nearest Neighbors (k-NN)

This model looks at the closest data points (neighbors) to make predictions. Super simple and based on similarity.

- [Introduction to KNN](#)
- [Decision Boundaries in K-Nearest Neighbors \(KNN\)](#)

6. Naïve Bayes

A quick and smart way to classify things based on probability. It works well for text and spam detection.

- [Introduction to Naive Bayes](#)
- [Gaussian Naive Bayes](#)
- [Multinomial Naive Bayes](#)
- [Bernoulli Naive Bayes](#)
- [Complement Naive Bayes](#)

7. Random Forest (Bagging Algorithm)

A powerful model that builds lots of decision trees and combines them for better accuracy and stability.

- [Introduction to Random forest](#)
- [Random Forest Classifier](#)
- [Random Forest Regression](#)
- [Hyperparameter Tuning in Random Forest](#)

Introduction to Ensemble Learning

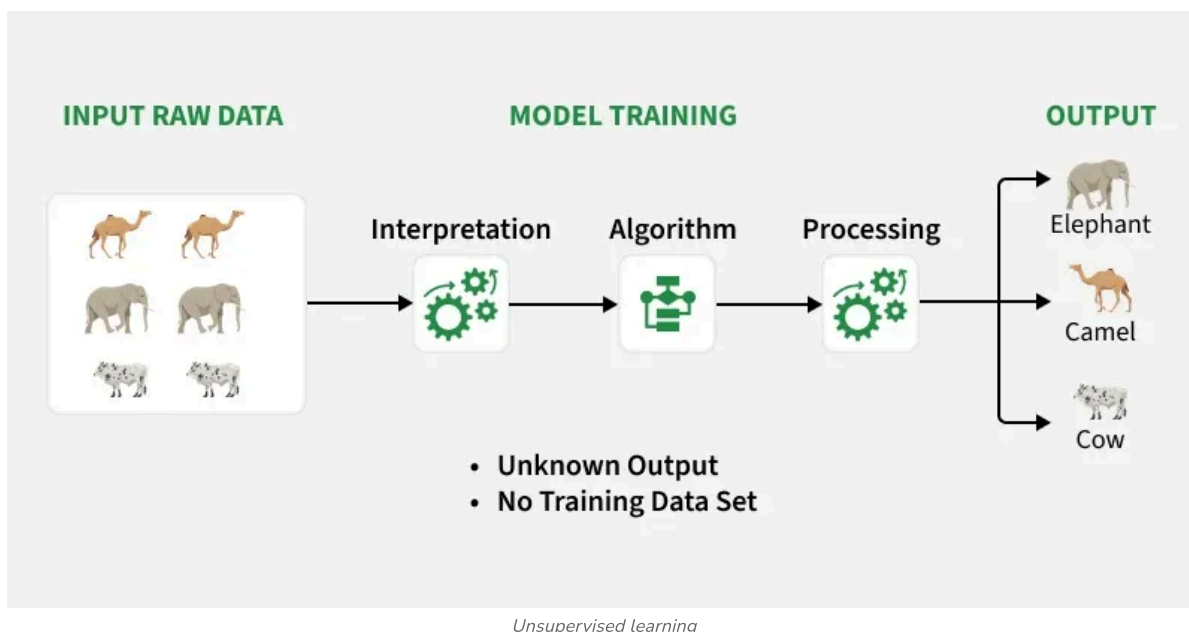
[Ensemble learning](#) combines multiple simple models to create a stronger, smarter model. There are mainly two types of ensemble learning:

- [Bagging](#) that combines multiple models trained independently.
- [Boosting](#) that builds models sequentially each correcting the errors of the previous one.

Module 3: Unsupervised learning

Unsupervised learning are again divided into **three main categories** based on their purpose:

- [Clustering](#)
- [Association Rule Mining](#)
- [Dimensionality Reduction](#)



1. Clustering

Clustering algorithms group data points into clusters based on their similarities or differences. Types of clustering algorithms are:

Centroid-based Methods:

- [K-Means clustering](#)
- [Elbow Method for optimal value of k in KMeans](#)
- [K-Means++ clustering](#)
- [K-Mode clustering](#)
- [Fuzzy C-Means \(FCM\) Clustering](#)

Distribution-based Methods:

- [Gaussian mixture models](#)
- [Expectation-Maximization Algorithm](#)
- [Dirichlet process mixture models \(DPMMs\)](#)

Connectivity based methods:

- [Hierarchical clustering](#)
- [Agglomerative Clustering](#)
- [Divisive clustering](#)
- [Affinity propagation](#)

Density Based methods:

- [DBSCAN \(Density-Based Spatial Clustering of Applications with Noise\)](#)
- [OPTICS \(Ordering Points To Identify the Clustering Structure\)](#)

2. Dimensionality Reduction

Dimensionality reduction is used to simplify datasets by reducing the number of features while retaining the most important information.

- [Principal Component Analysis \(PCA\)](#).
- [t-distributed Stochastic Neighbor Embedding \(t-SNE\)](#).
- [Non-negative Matrix Factorization \(NMF\)](#).
- [Independent Component Analysis \(ICA\)](#).
- [Isomap](#)
- [Locally Linear Embedding \(LLE\)](#).

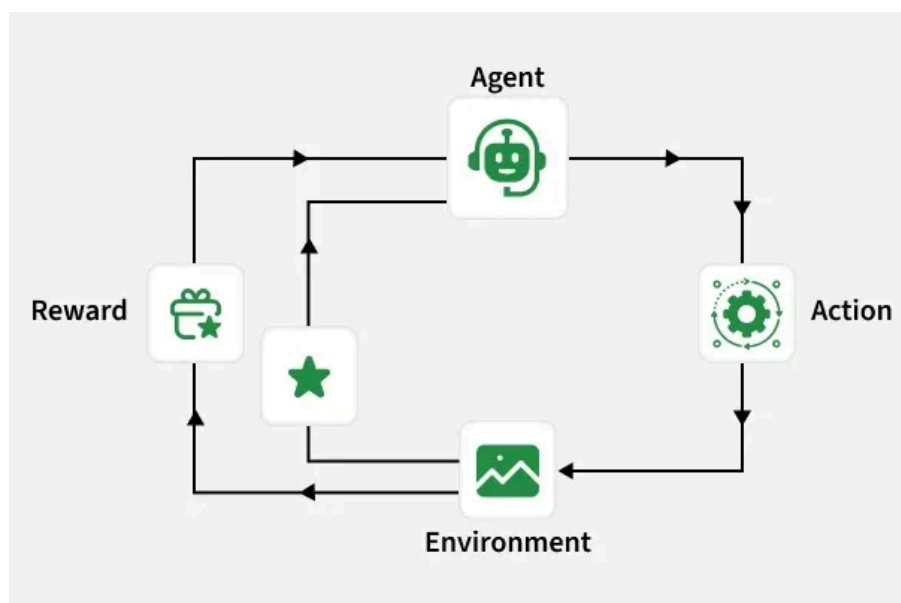
3. Association Rule

Find patterns between items in large datasets typically in [market basket analysis](#).

- [Apriori algorithm](#)
- [Implementing apriori algorithm](#)
- [FP-Growth \(Frequent Pattern-Growth\)](#).
- [ECLAT \(Equivalence Class Clustering and bottom-up Lattice Traversal\)](#).

Module 4: Reinforcement Learning

Reinforcement learning interacts with environment and learn from them based on rewards.



Reinforcement Learning

1. Model-Based Methods

These methods use a model of the environment to predict outcomes and help the agent plan actions by simulating potential results.

- [Markov decision processes \(MDPs\)](#).
- [Bellman equation](#)
- [Value iteration algorithm](#)
- [Monte Carlo Tree Search](#)

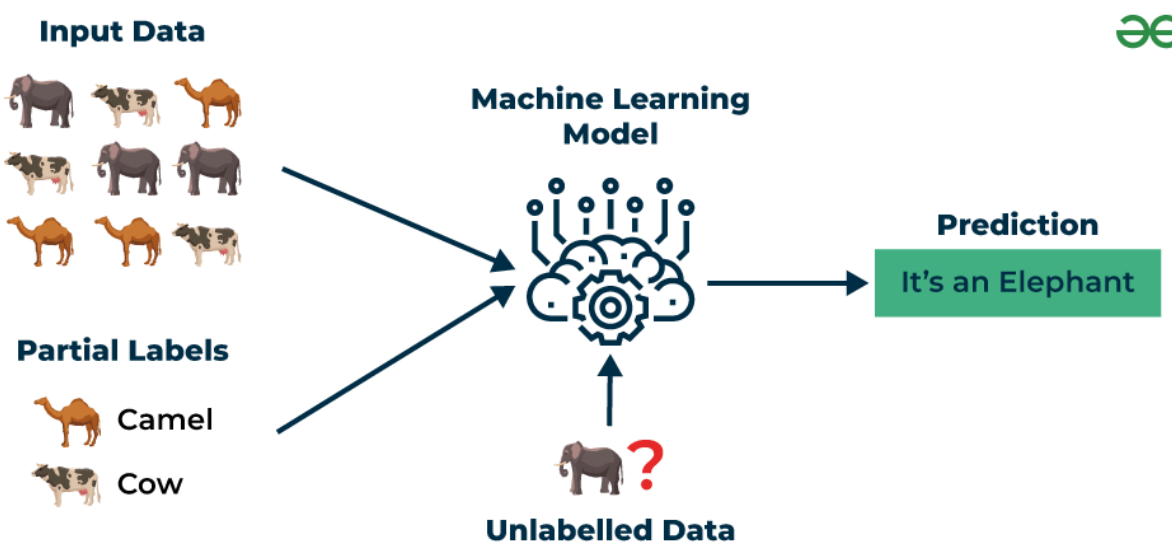
2. Model-Free Methods

The agent learns directly from experience by interacting with the environment and adjusting its actions based on feedback.

- [Q-Learning](#)
- [SARSA](#)
- [Monte Carlo Methods](#)
- [Reinforce Algorithm](#)
- [Actor-Critic Algorithm](#)
- [Asynchronous Advantage Actor-Critic \(A3C\)](#)

Module 5: Semi Supervised Learning

It uses a mix of labeled and unlabeled data making it helpful when labeling data is costly or it is very limited.



Semi Supervised Learning

- [Semi Supervised Classification](#)
- [Self-Training in Semi-Supervised Learning](#)
- [Few-shot Learning in Machine Learning](#)

Module 6: Forecasting Models

Forecasting models analyze past data to predict future trends, commonly used for time series problems like sales, demand or stock prices.

- [ARIMA \(Auto-Regressive Integrated Moving Average\)](#)
- [SARIMA \(Seasonal ARIMA\)](#)
- [Exponential Smoothing \(Holt-Winters\)](#)

Module 7: Deployment of ML Models

The trained ML model must be integrated into an application or service to make its predictions accessible.

- [Machine learning deployment](#)
- [Deploy ML Model using Streamlit Library](#)

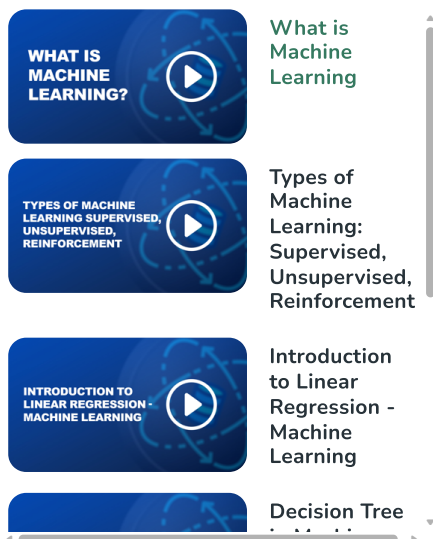
- [Deploy ML web app on Heroku](#)
- [Create UIs for prototyping Machine Learning model with Gradio](#)

APIs allow other applications or systems to access the ML model's functionality and integrate them into larger workflows.

- [Deploy Machine Learning Model using Flask](#)
- [Deploying ML Models as API using FastAPI](#)

MLOps ensure they are deployed, monitored and maintained efficiently in real-world production systems.

- [MLOps](#)
 - [Continuous Integration and Continuous Deployment \(CI/CD\) in MLOps](#)
 - [End-to-End MLOps](#)
- **For project ideas refer to:** [100+ Machine Learning Projects with Source Code \[2025\]](#) for hands-on implementation on projects
 - **After machine learning and have hands on experience in it we can start with deep learning from here:** [Deep Learning Tutorial](#)



What is Machine Learning

[Visit Course](#)

Comment

K kartik + Follow

302

Article Tags: [Machine Learning](#) [AI-ML-DS](#) [Tutorials](#) [Machine Learning](#)

Explore

Machine Learning Basics

Python for Machine Learning

Feature Engineering