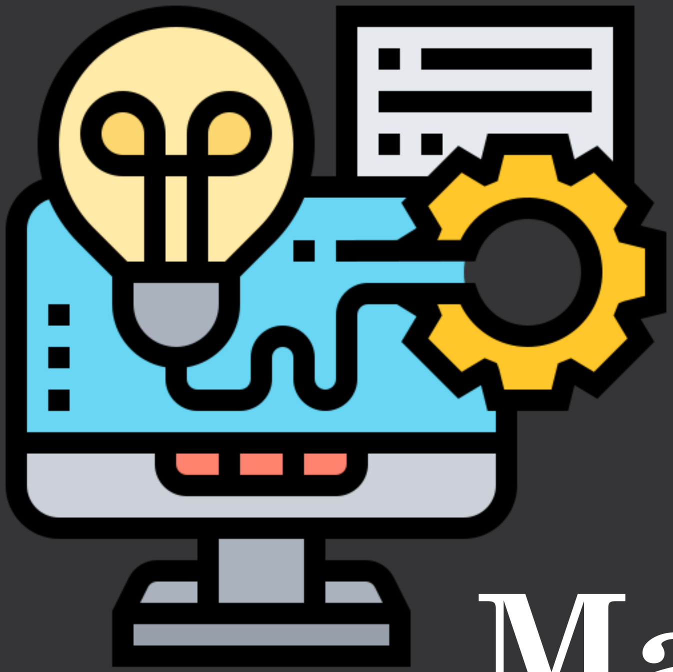


Machine Learning

Lecture 7 – HCCDA-AI

Imran Nawar

Overview

- Traditional Programming vs. Machine Learning
- **Mathematics for Machine Learning**
  - Linear Algebra
  - Calculus
  - Statistics and Probability
- **Machine Learning**
  - Introduction to Machine Learning
  - Machine Learning Examples
  - Evolution of Machine Learning
  - Types of Machine Learning
    - Supervised Learning
    - Unsupervised Learning
    - Reinforcement Learning
  - Workflow of Machine Learning
- **Key Concepts**
  - Data and Features
  - Algorithms and Models
  - Training and Testing
  - Evaluation Metrics
- Common Algorithms
- Tools and Frameworks
- Challenges and Limitations
- Future Trends

Traditional Programming vs Machine Learning

Traditional Programming

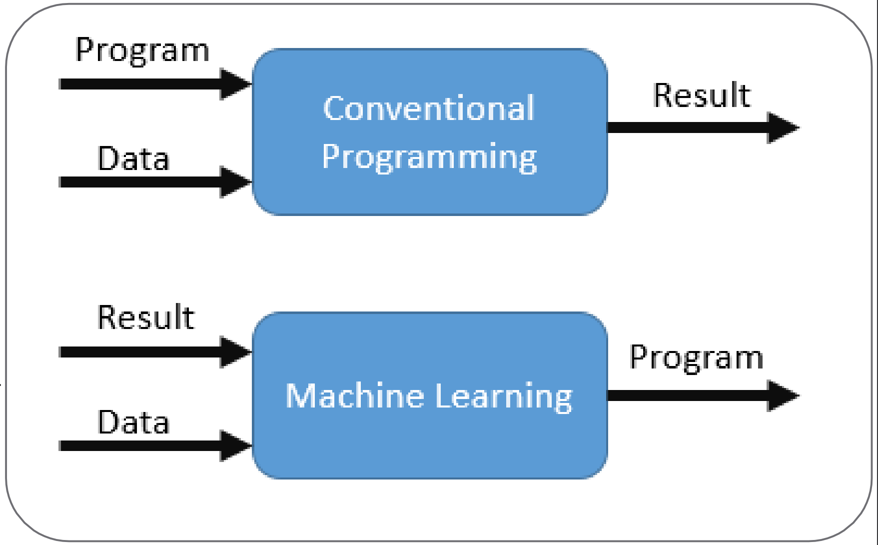
- **Approach**: Write explicit rules and logic for specific tasks.
- **Input**: **Rules** (logic) + **Data** → **Output**.
- **Key Focus**: Programmer defines all possible scenarios.
- **Example**:
  - Use if-else statements to classify emails as spam or not.

Machine Learning

- **Approach**: Learn patterns from data to make predictions or decisions.
- **Input**: **Data** + **Output** (labels) → Learns **Rules** (Model).
- **Key Focus**: Relies on high-quality training data.
- **Example**:
  - Train a spam classifier with labeled examples of spam and non-spam emails.

Key Difference

- **Traditional Programming**: Fixed rules; limited flexibility.
- **Machine Learning**: Adaptive rules; improves with more data.

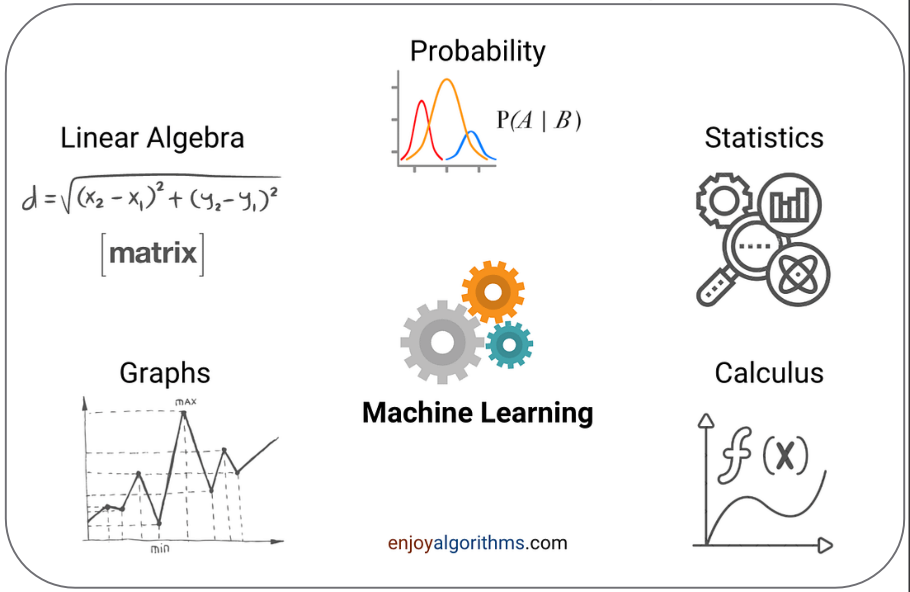


Mathematics for Machine Learning

Mathematics: The Foundation of Machine Learning

Why Mathematics Matters in ML

- Machine learning algorithms are built on mathematical foundations translated into code.
- Understanding key concepts helps you grasp how algorithms learns and why they behave the way they do.
- Essential for optimizing and debugging ML systems.
- Foundation for reading research papers and documentation.



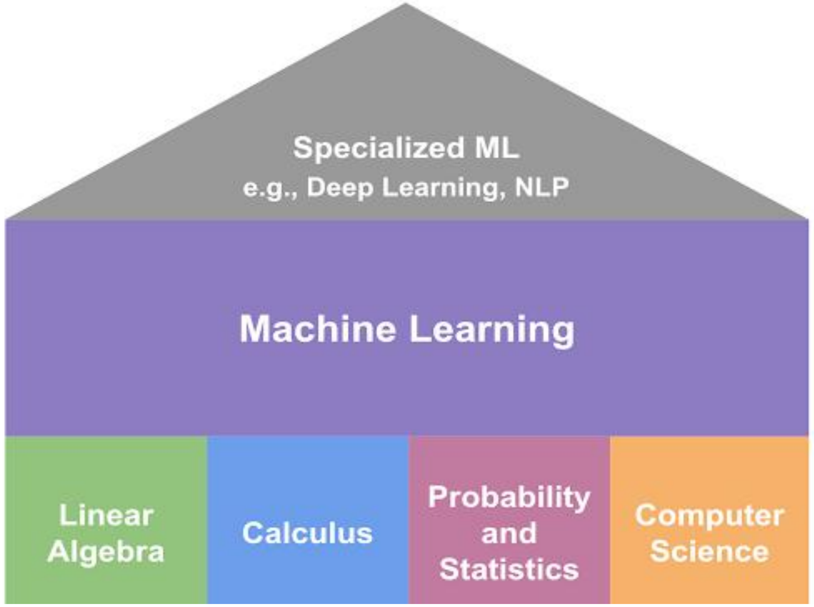
Key Areas for Mathematics in ML

- **Linear Algebra:** Vectors, matrices, transformations are foundational for data representation and manipulation.
- **Probability & Statistics:** Essential for model evaluation, hypothesis testing, and analyzing uncertainty in data.
- **Calculus:** Derivatives, gradients and optimization techniques are central to training models (e.g., gradient descent).

If you want an ML career:

- Machine Learning Engineer
- Data Scientist
- AI Engineer
- Data Analyst (for analytical and statistical modeling tasks)

You should focus on the mathematic concepts described here.



Linear Algebra: A Core Pillar of Machine Learning

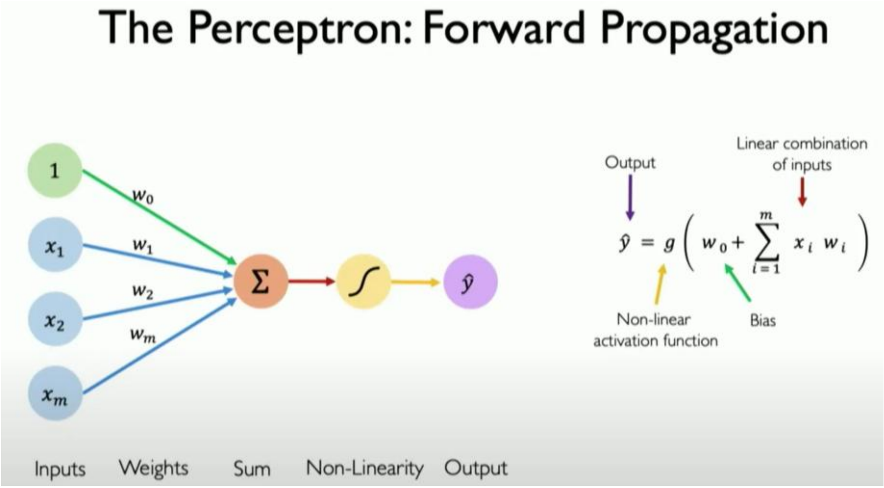
- Linear algebra is a branch of mathematics that deals with vectors, matrices, and how they transform or interact.

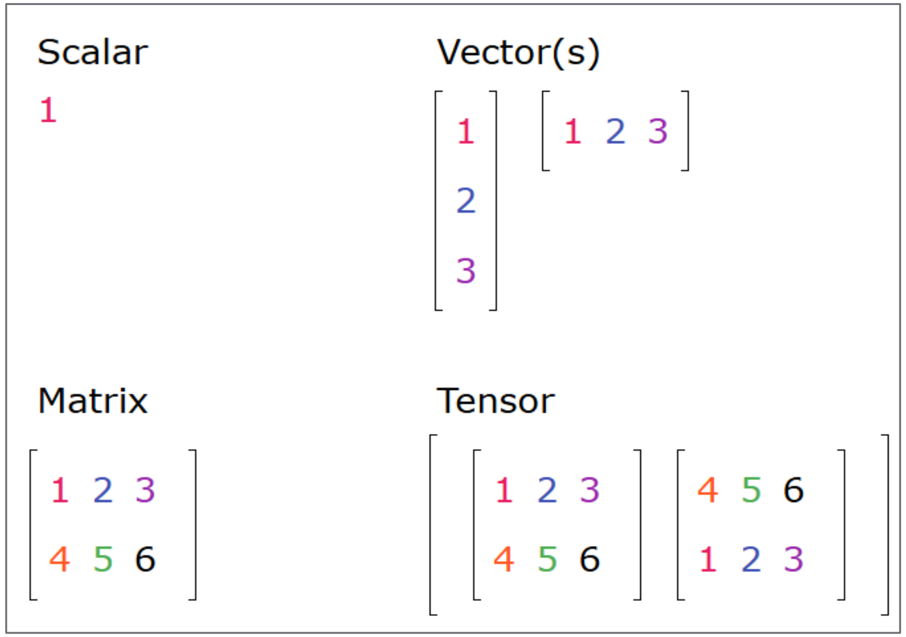
Key Concepts to Know:

- **Vectors & operations:** For example, adding vectors, multiplying by a number.
- **Systems of linear equations:** Solving multiple equations at once.
- **Matrix operations:** Like addition, multiplication, transpose.
- **Linear transformations:** Changing vectors while keeping their structure.

Why It Matters in Machine Learning:

- Linear algebra is used to:
  - Represent data (as vectors and matrices).
  - Power key algorithms (e.g., **PCA**, **SVD**).
  - Enable neural network computations (matrix operations in each layer).
  - Perform dimensionality reduction (simplifying complex data).





Essential Linear Algebra Concepts for ML

- **Matrix:** Rectangular array of numbers ($m*n$ dimensions).
  - **Matrix Elements:** Entries of matrix.
  - **Operations:** Addition (same dimensions), scalar multiplication, matrix multiplication.
- **Vector:** Special case of matrix ($n*1$ or $1*n$).
  - Represents data points, features, or model parameters.
- **Identity Matrix (I):** Square matrix with 1s on diagonal, 0s elsewhere.
- **Matrix Transpose (A^T):** Rows become columns and vice versa
  - Essential for many ML computations

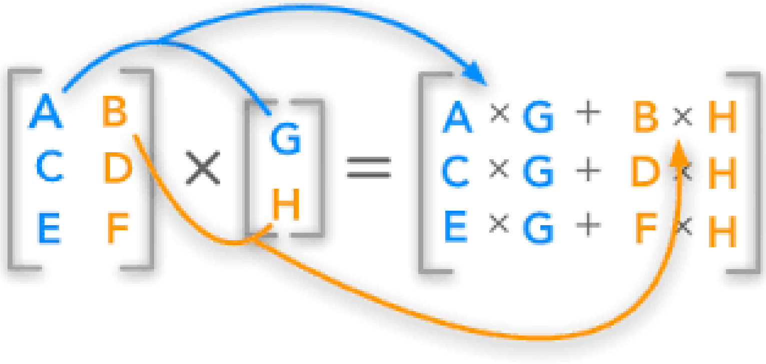
Formula:

For matrices A and B:

$$C_{i,j} = A_{i,j} + B_{i,j}$$

For matrices A and scalar k:

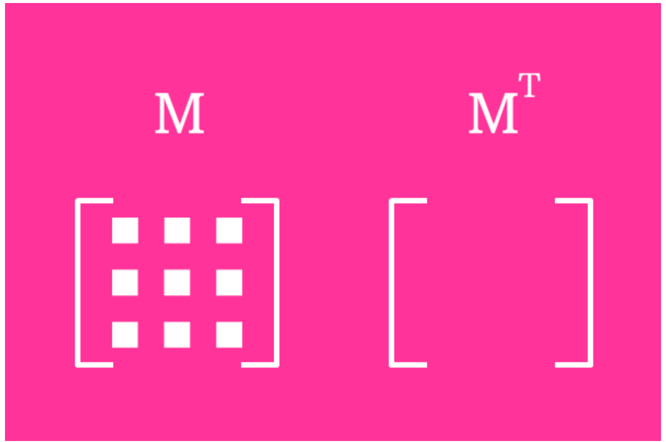
$$(kA)_{i,j} = k \,.\, A_{i,j}$$



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
2 x 2

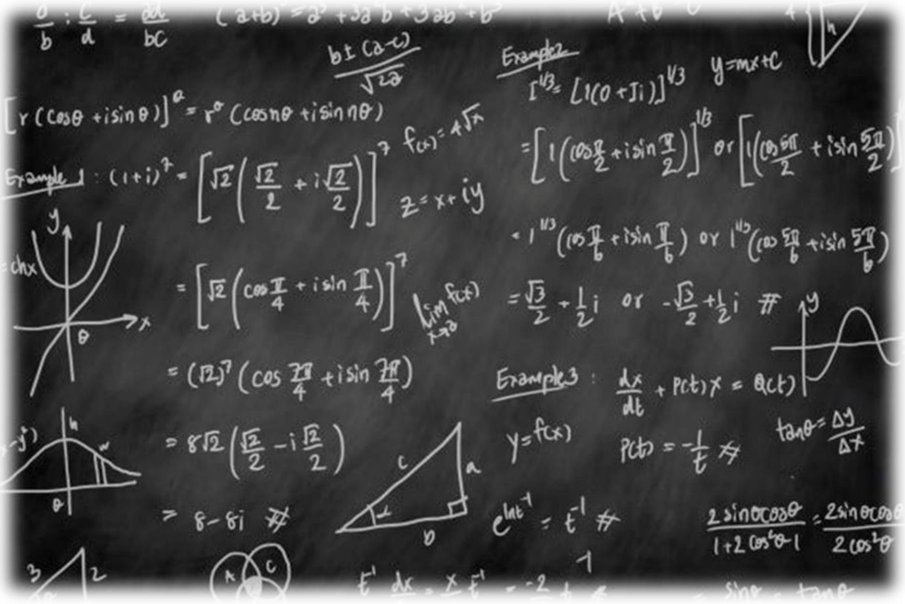
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
3 x 3

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
4 x 4



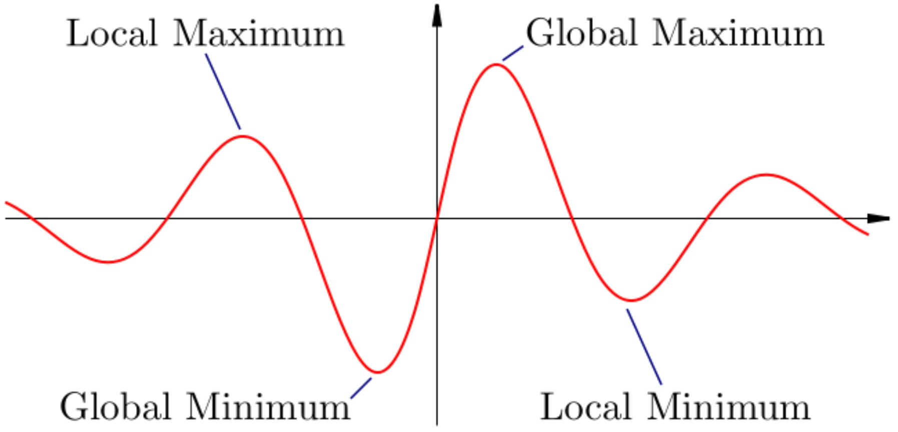
Calculus – The Engine of Machine Learning Optimization

- **Calculus** is the mathematics of change and rates. It is essential for understanding how ML models learn and improve through optimization.



Core Concepts for Machine Learning:

- **Derivatives**: Measure the rate of change of a function
- **Partial derivatives**: First-order derivatives w.r.t. one variable (others held constant).
- **Gradient:** A vector of all partial derivatives; direction of steepest ascent.
- **Second-Order Derivatives:** Describe how the slope itself changes (curvature).
- **Hessian Matrix:** Matrix of second-order partial derivatives; used to analyze curvature of loss surfaces.
- **Chain Rule:** Crucial for computing gradients in neural networks (backpropagation).



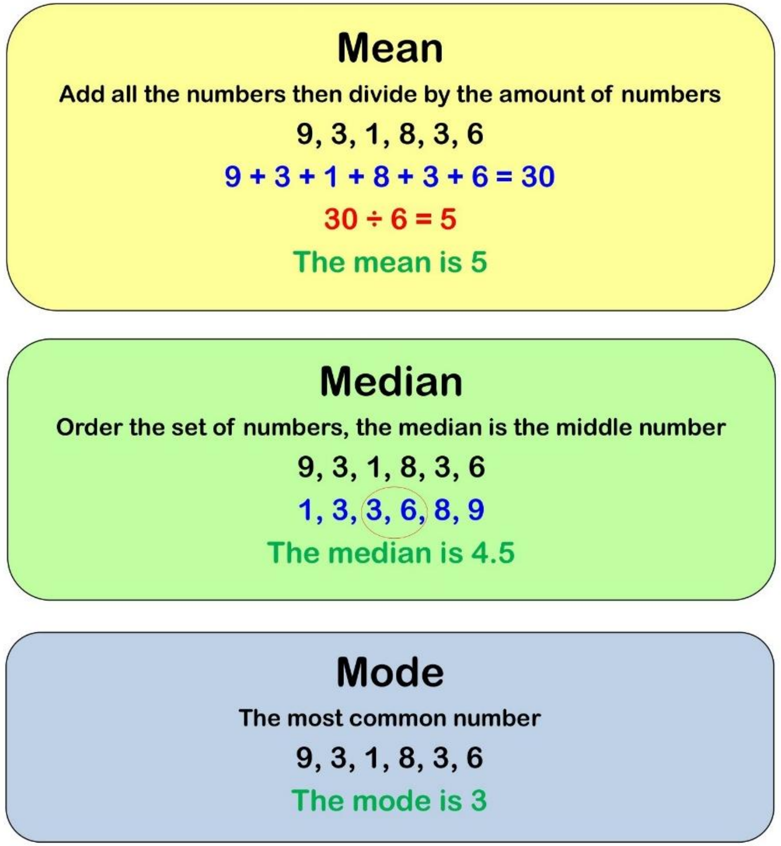
Statistics and Probability – Understanding Data and Uncertainty

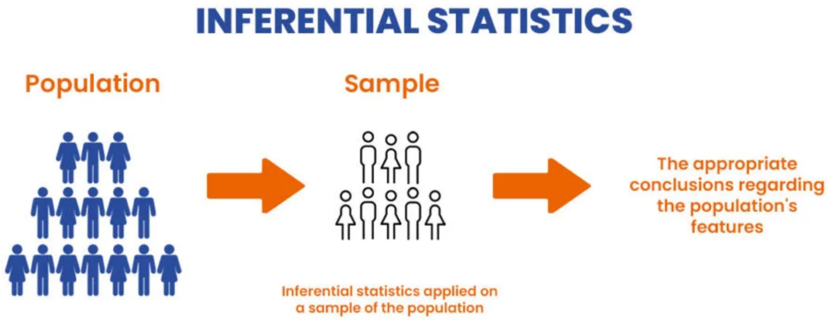
Statistics in Machine Learning

- Statistics is the science of collecting, analyzing, summarizing, and interpreting data.
- In machine learning, it helps us make sense of large datasets and draw meaningful conclusions.

- **Why is Statistics Important in ML?**
  - **Summarizing Data:** Measures like *mean*, *median*, and *mode* help us understand the central tendency of data.
  - **Understanding Variability:** *Variance* and *standard deviation* show how much data points spread out from the average.
  - **Making Inferences:** Inferential statistics (e.g., *hypothesis testing* and *confidence intervals*) help us generalize from samples to populations, which is vital when evaluating model performance.

- **Applications in ML:**
  - Statistical concepts are used in feature selection, model evaluation, and understanding data distributions.





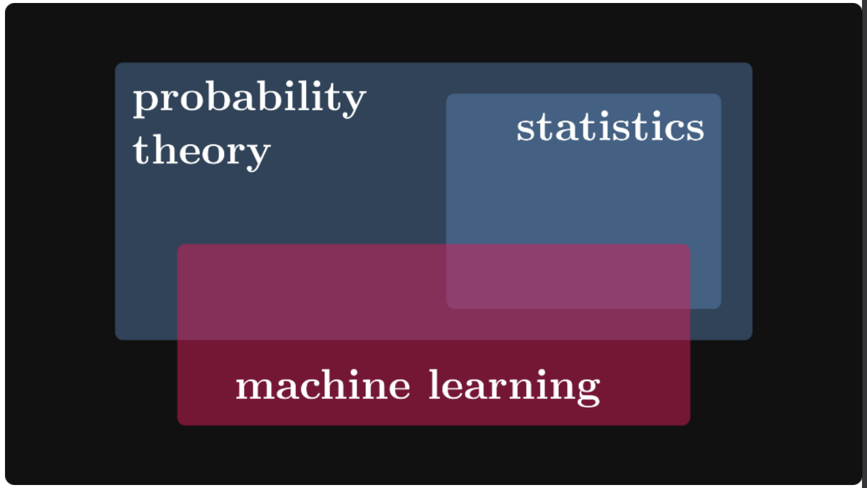
Statistics and Probability – Understanding Data and Uncertainty

Probability in Machine Learning

- Probability measures how likely an event is to happen, ranging from 0 (impossible) to 1 (certain).

- **Why It Matters in ML?**
  - **Handling Uncertainty:** Helps models deal with uncertain or noisy data (e.g., spam detection).
  - **Probability Distributions:** Many algorithms assume data follows certain distributions (like *normal distribution*), which helps in model design and evaluation.
  - **Bayesian Inference:** Updates predictions as new data arrives, foundational for algorithms like Naive Bayes.

- **Applications in ML:**
  - Algorithms such as Naive Bayes, logistic regression, and Markov models are built on probability theory.
  - Probability helps quantify how confident we are in model predictions and guides decision-making in uncertain situations.



Machine Learning

Introduction To Machine Learning

- Machine learning is the science (and art) of programming computers so they can learn from data.

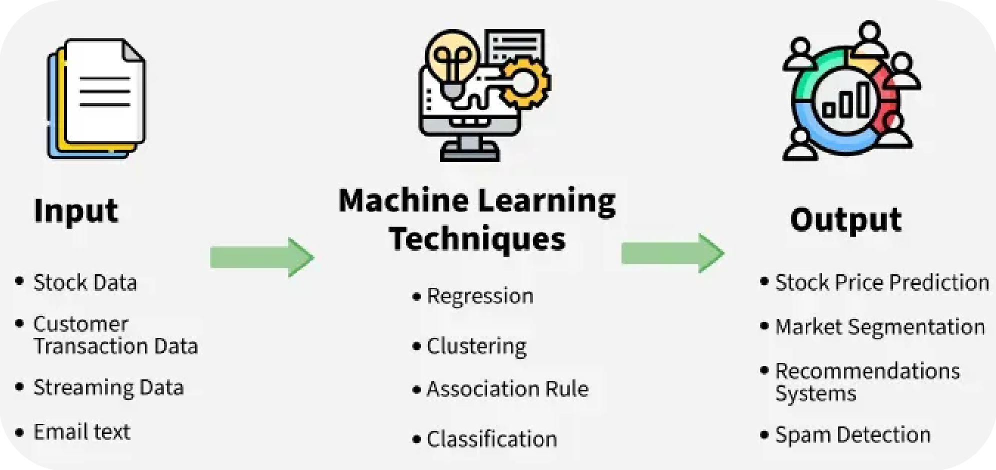
- [Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

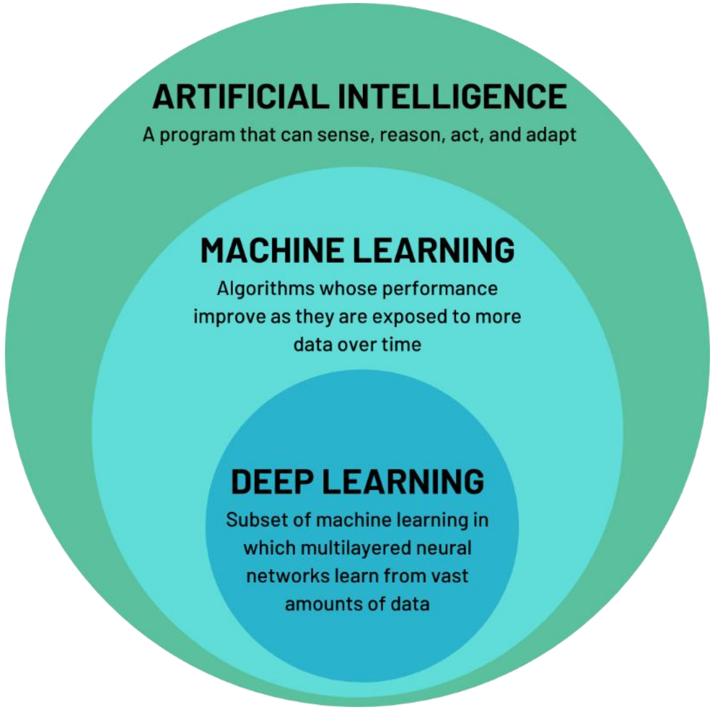
  — Arthur Samuel, 1959

- "A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

  —Tom Mitchell, 1997

  - [Machine Learning means a computer improves at a task (T) over time as it gains experience (E), measured by a performance metric (P)].





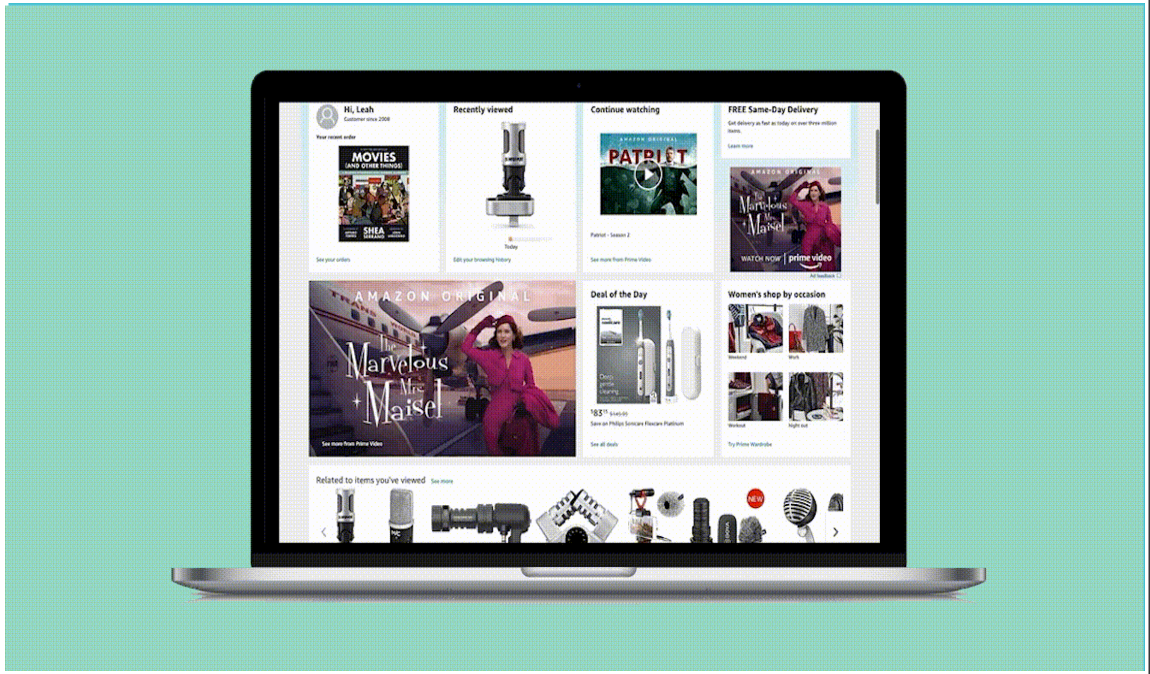
ML Examples

- **Complex Tasks Too Difficult to Program**
  - **Autonomous Vehicles**: Navigate roads, detect obstacles, make real-time decisions.
  - **Computer Vision**: Medical imaging (tumor detection), face recognition.
  - **Natural Language Processing**: Language translation, chatbots, sentiment analysis.

- **Intelligent Personalization Systems**
  - **Recommendation Engines**: Amazon product suggestions.

- **Pattern Discovery in Big Data**
  - **Fraud Detection**: Real-time suspicious transaction identification.
  - **Predictive Analytics**: Equipment failure prediction, market trend analysis.

- **Why ML Excels:**
  - Traditional programming can not handle the complexity, variability, and scale
  - ML learns patterns from data and improves with experience.



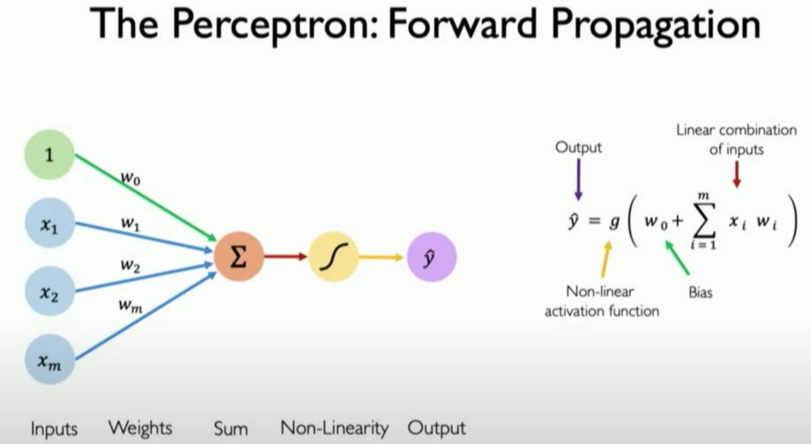
Evolution of Machine Learning

Early Foundations (1950s – 1980s)

- **Dartmouth Conference (1956):**
  - Dartmouth Summer Research Project on Artificial Intelligence – officially founded AI as a field.
  - Researchers discussed creating intelligent machines.
- **Initial Algorithms and Theories:**
  - **Perceptron** (1957): *Frank Rosenblatt* – first trainable neural network.
  - **Decision Trees** (1960s): ID3 algorithm.
  - **Bayesian Networks** (1980s): Probabilistic reasoning models.

Key Algorithmic Breakthroughs (1980s – 1990s)

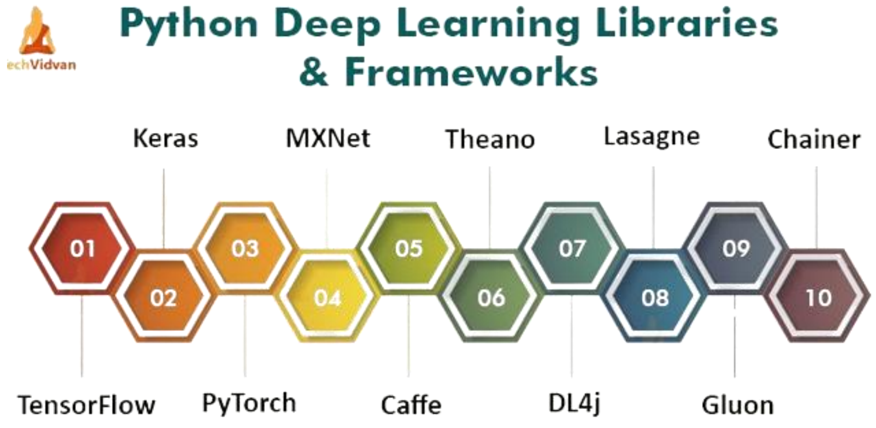
- **Neural Network Revival:**
  - Backpropagation (1986) – *Rumelhart, Hinton*, and *Williams* – enabled training deep networks
  - Support Vector Machines (1990s) – *Vapnik* and *Cortes* – powerful classification method



Evolution of Machine Learning

Big Data Era (2000s-2010s):

- **Rise of Computational Power:**
  - Internet and sensors generated massive datasets for training.
  - GPU acceleration revolutionized neural network training.
  - Open-source frameworks: TensorFlow, PyTorch, Scikit-learn.



Deep Learning Revolution (2010s-Present):

- **Breakthrough Architectures:**
  - Convolutional Neural Networks (CNNs) – image recognition
  - Recurrent Neural Networks (RNNs) – sequential data processing
- **Major Milestones:**
  - AlphaGo (2016) – Google DeepMind defeats world Go champion.
  - GPT Models (2018-present) – OpenAI's language models.
  - Transformer Architecture (2017) – "Attention is All You Need"

Current Era: Foundation Models

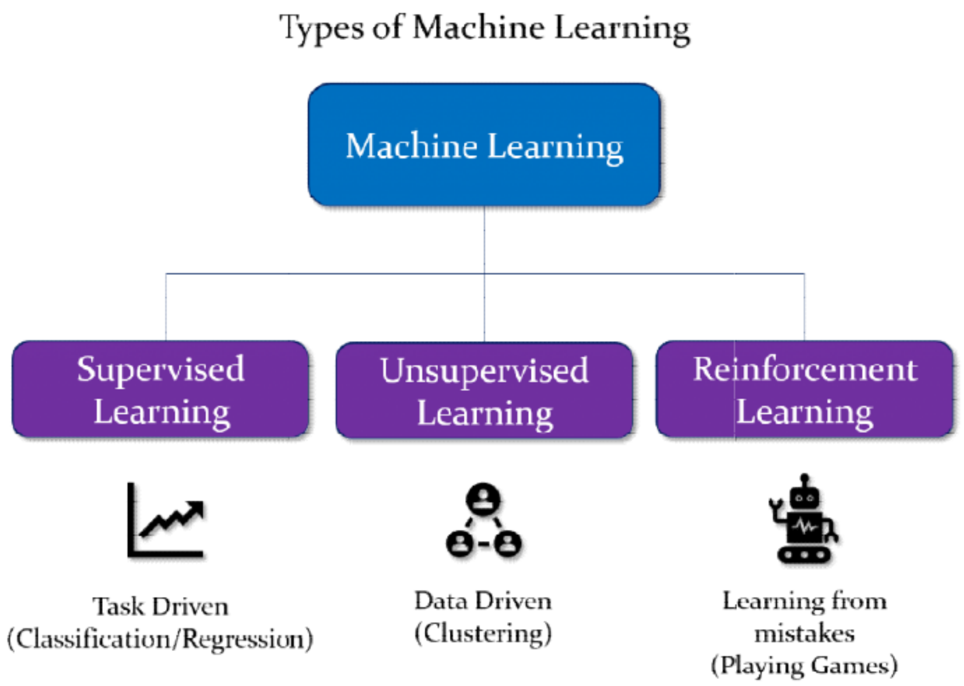
  - Large-scale pre-trained models (GPT, BERT, DALL-E) transforming AI applications.

4 – 1

Types of Machine Learning

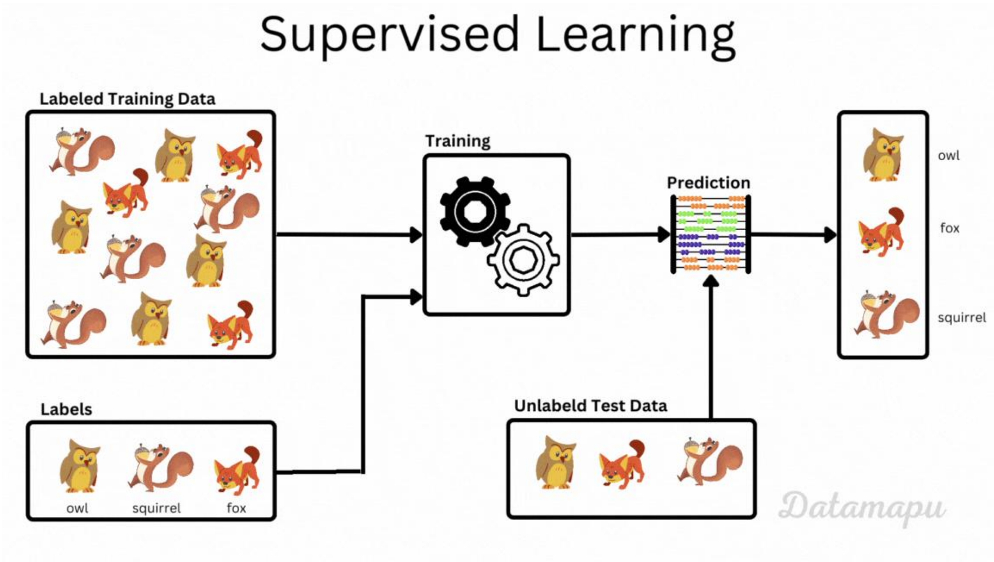
1. ## Supervised Learning

- Supervised Learning involves training a model on a labeled dataset, where the input data is paired with the correct output.
- “right answers” given.
- Supervised learning learn form data labeled with the “right answers”.

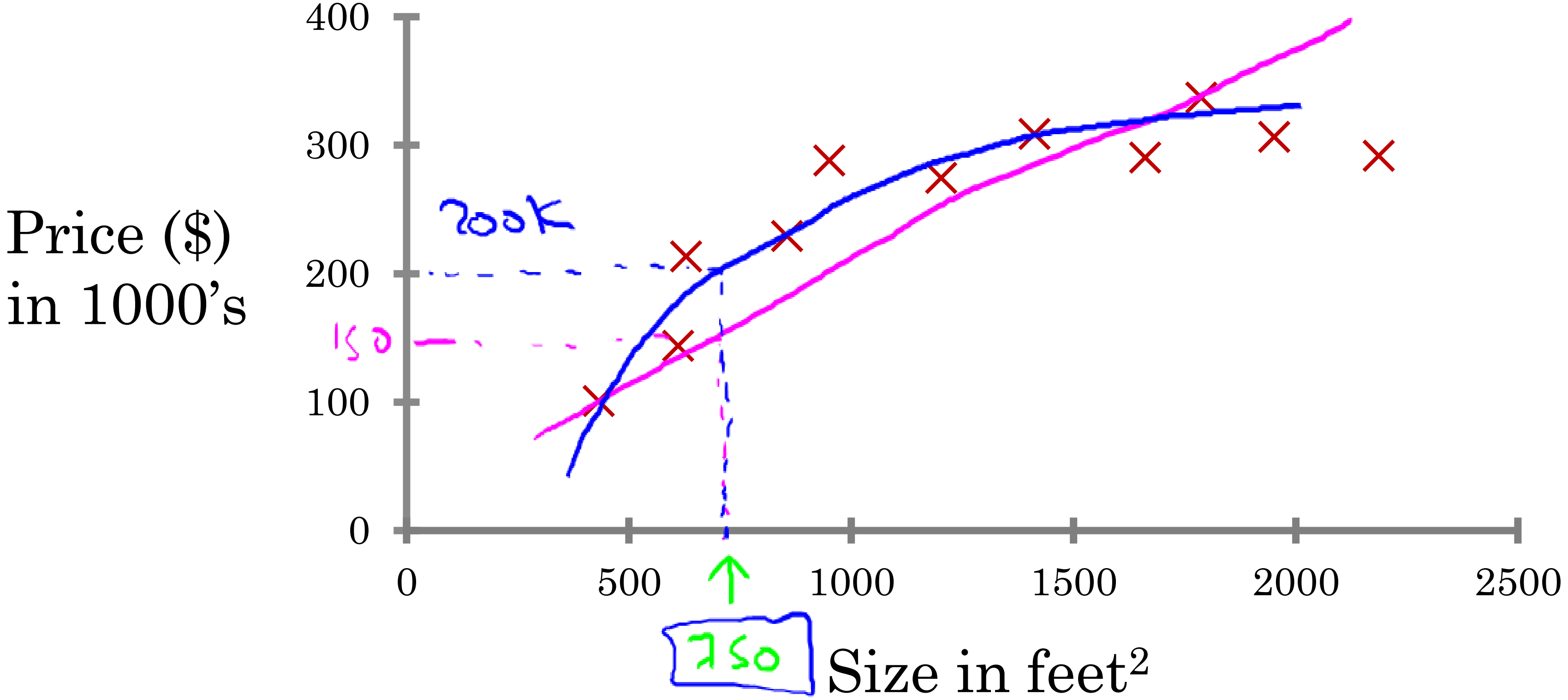


Common Tasks in Supervised Learning:

1. **Regression:**
   - Predict continuous valued output.
   - For example, predicting house prices based on features like size, location, and age.
2. **Classification:**
   - Predict discrete valued output.
   - Categorizing input data into predefined classes.
   - For example, identifying whether an email is spam or not spam based on its content.



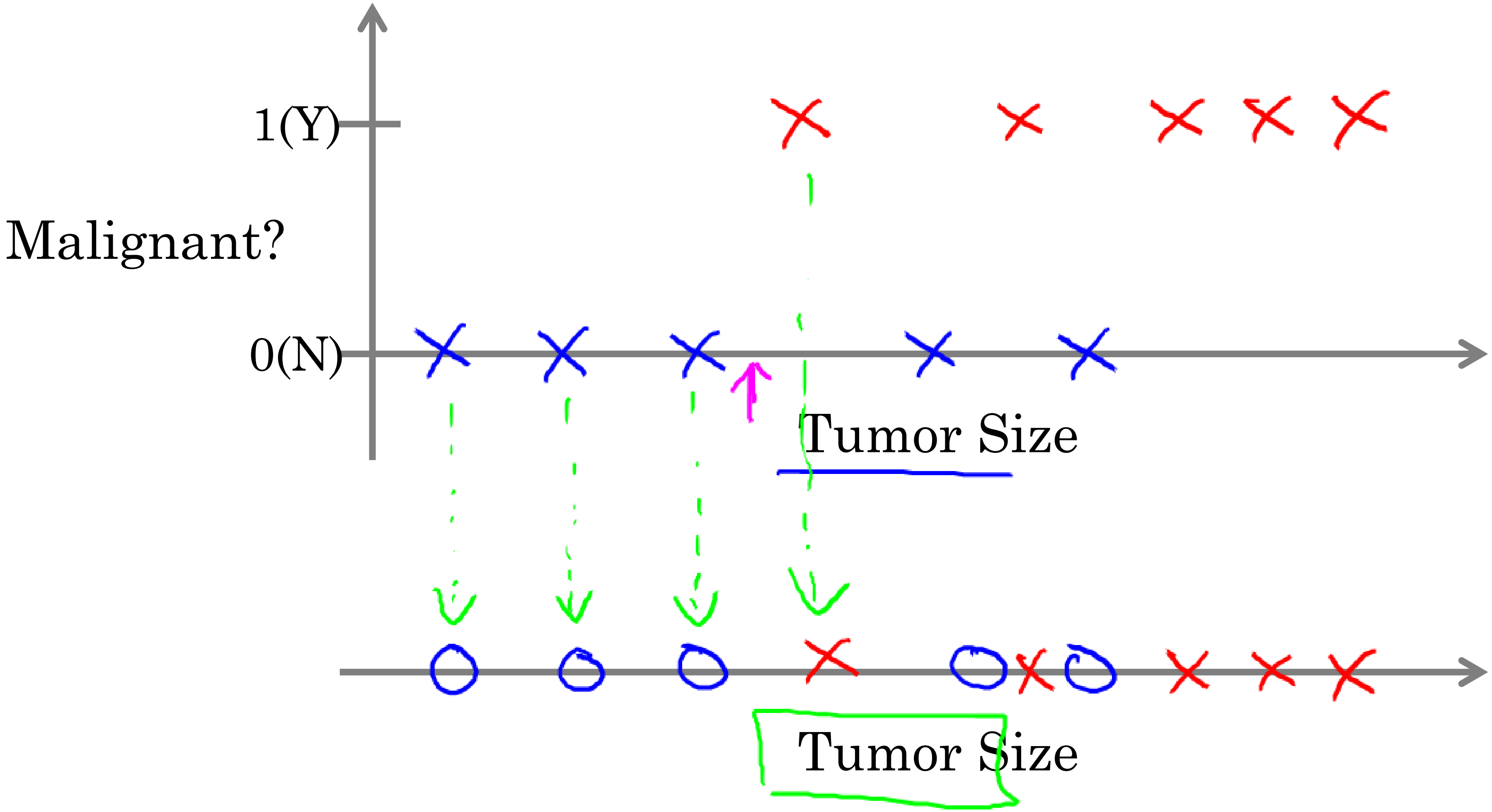
Housing price prediction



Supervised Learning
"right answers" given

Regression: Predict continuous valued output (price)

Breast cancer (malignant, benign)



Classification

Discrete valued output (0 or 1)

Classification vs. Regression: Identifying the Right Approach

- You're running a company, and you want to develop learning algorithms to address each of two problems.

- **Problem 1:** You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
- **Problem 2:** You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

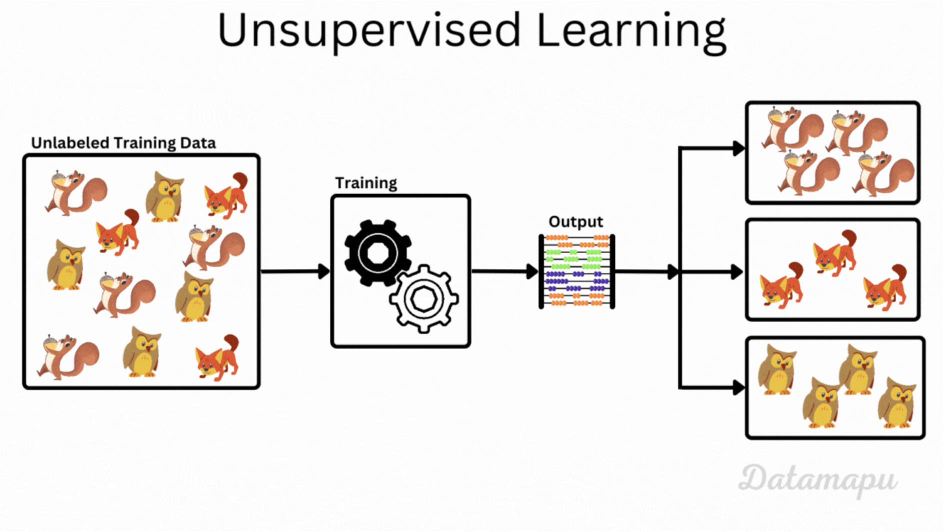
- Should you treat these as classification or as regression problems?

1. Treat both as classification problems.
2. Treat problem 1 as a classification problem, problem 2 as a regression problem.
3. Treat problem 1 as a regression problem, problem 2 as a classification problem.
4. Treat both as regression problems.

Types of Machine Learning

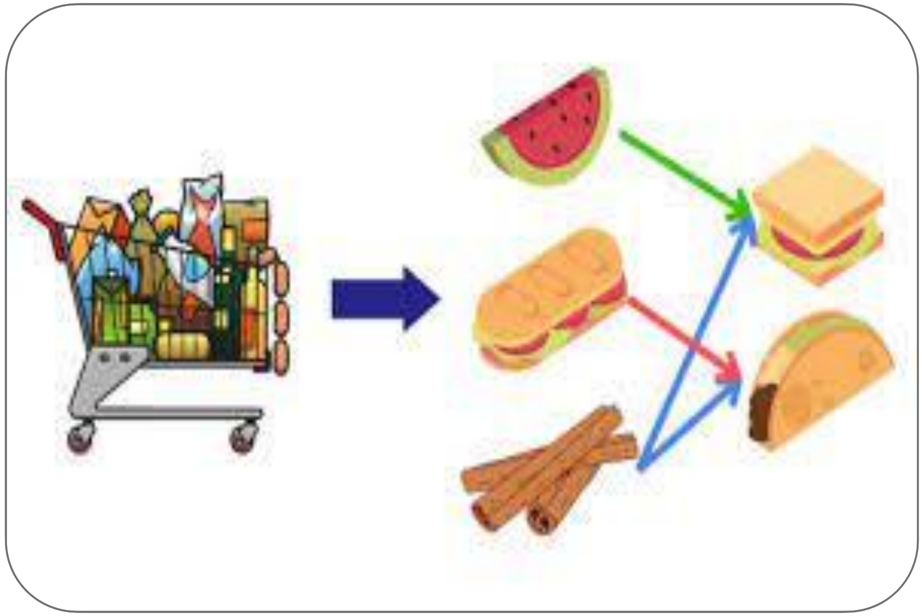
2. Unsupervised Learning

- Unsupervised Learning involves training a model on data that does not have labeled responses.
- The model tries to find hidden patterns or intrinsic structures in the input data.



Common Tasks in Unsupervised Learning:

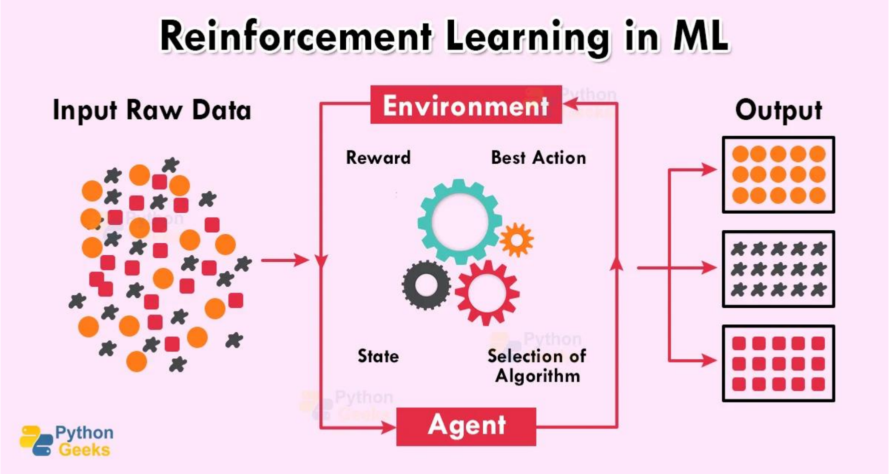
1. **Clustering:**
   - Grouping similar data points together.
   - For example, customer segmentation in marketing, where customers are grouped based on purchasing behavior.
2. **Anomaly Detection:**
   - Identifying outliers or unusual data points that do not conform to the expected pattern.
   - For example, detecting fraudulent transactions in financial systems.
3. **Association:**
   - Discovering rules that describe large portions of the data.
   - For example, market basket analysis in retail, where associations between products bought together are identified.



Types of Machine Learning

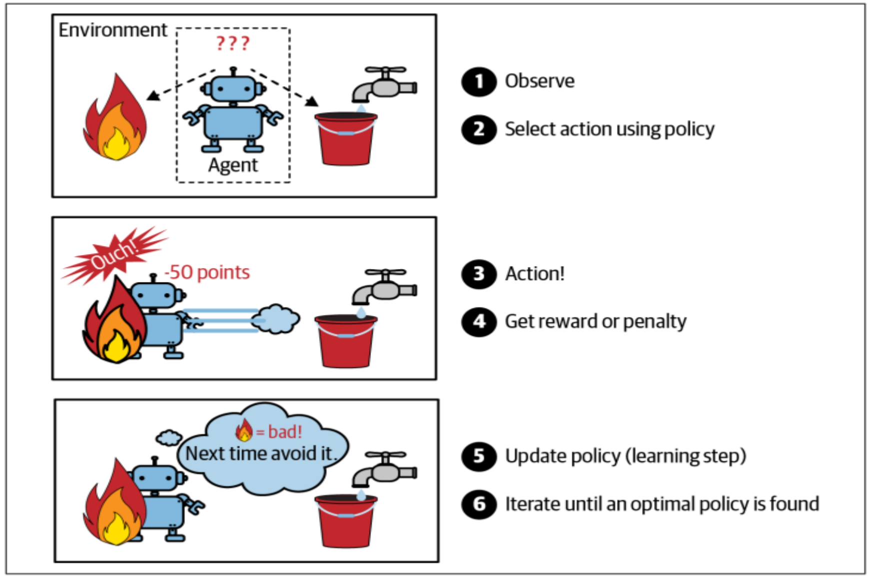
3. Reinforcement Learning

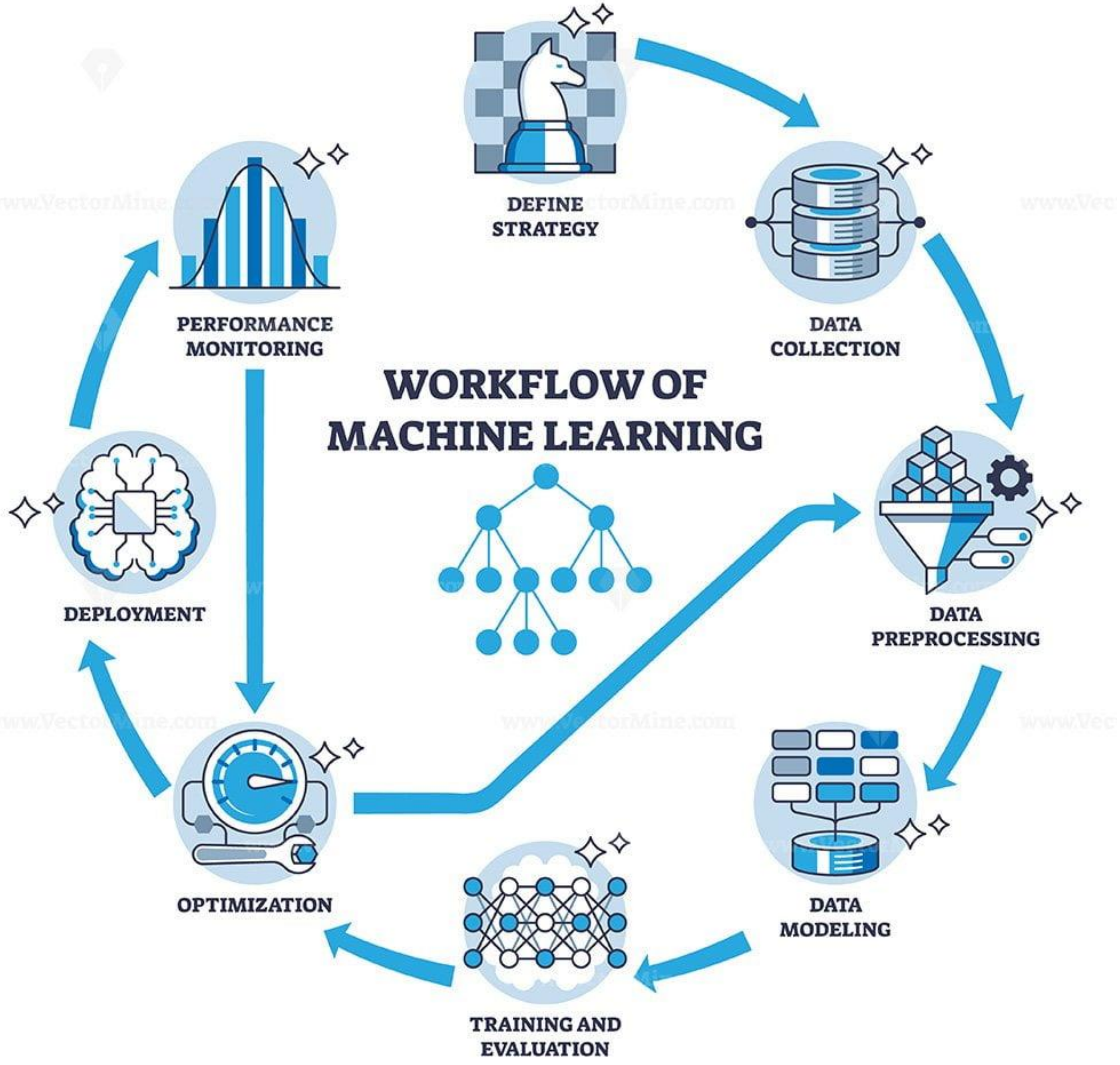
- Reinforcement Learning is a type of machine learning where an agent learns by interacting with an environment, not from fixed labeled data, but from trial and error.
- It involves training a model to make sequences of decisions by interacting with an environment.
- The model receives feedback in the form of rewards or penalties and learns to optimize its actions to maximize cumulative rewards.



Common Reinforcement Learning Tasks:

1. **Game Playing:**
   - Training an AI to play games like Chess or Go, where the AI learns strategies through trial and error and feedback from wins or losses.
2. **Robotics:**
   - Teaching robot to perform tasks such as walking, picking up objects, or navigating spaces, based on feedback from the environment.




WORKFLOW OF
MACHINE LEARNING
DEFINE
STRATEGY
DATA
COLLECTION
DATA
PREPROCESSING
DATA
MODELING
TRAINING AND
EVALUATION
OPTIMIZATION
DEPLOYMENT
PERFORMANCE
MONITORING

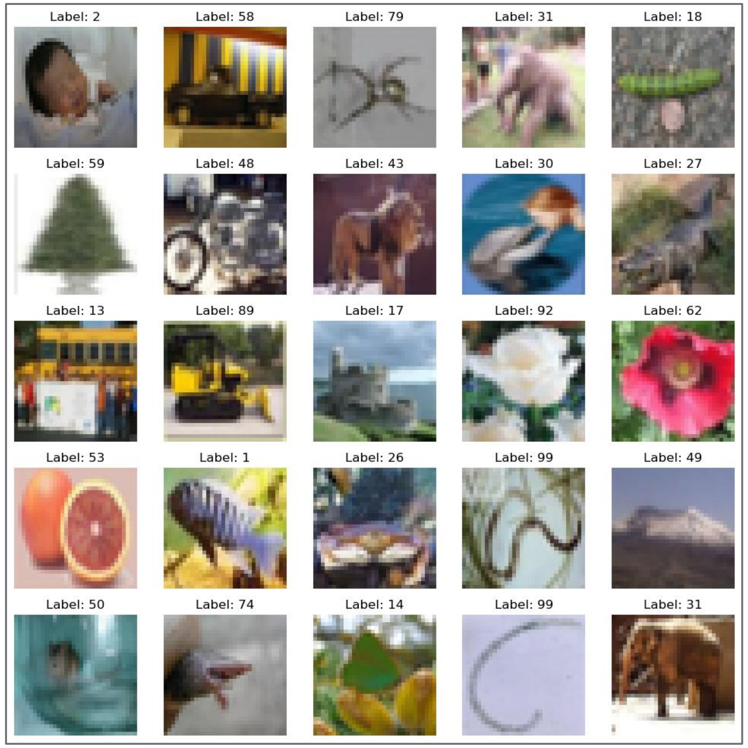
Key Concepts

Data and Features

- **Data**
  - The raw information collected from various sources, such as text, images, or numerical values, that is used to train machine learning models.
  - **Example:** A dataset of housing prices with columns like size, number of bedrooms, location, and price.

- **Features**
  - The individual measurable properties or characteristics of the data that are used as input for the model.
  - **Example:** In the housing dataset, features would include size, number of bedrooms, and location.

	A	B	C	D	E	F
1	House	Price ($M)	House (K sf)	Lot (K sf)	Bedrooms	Bathrooms
2	House 1	6	6.9	42.7	6	9
3	House 2	5.8	8	36.6	6	8
4	House 3	5.6	8	44	7	7
5	House 4	3.5	3.8	18	4	4
6	House 5	3.4	6.1	27.4	5	5
7	House 6	3.4	4.3	22.2	5	4
8	House 7	2.7	3.8	22	4	5
9	House 8	2.6	5	29.3	4	5
10	House 9	2.6	3.6	31.4	4	4
11	House 10	2.3	3.1	22.2	4	5
12	House 11	2.3	3.9	21.7	5	4
13	House 12	2.3	3.2	24.4	4	4
14	House 13	1.9	3.5	25.3	4	3
15	House 14	1.9	3.4	24	5	4
16	House 15	1.9	3.2	21.8	4	4
17	House 16	1.6	3.3	6.6	4	4
18	House 17	1.6	2.3	15.9	4	4
19	House 18	1.5	2.3	21.2	3	3

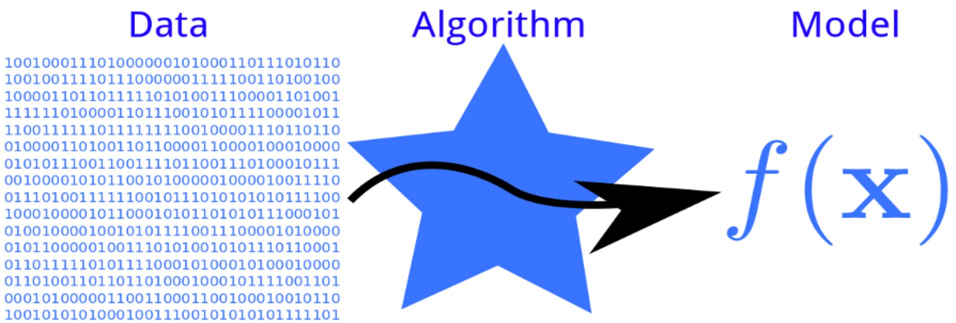


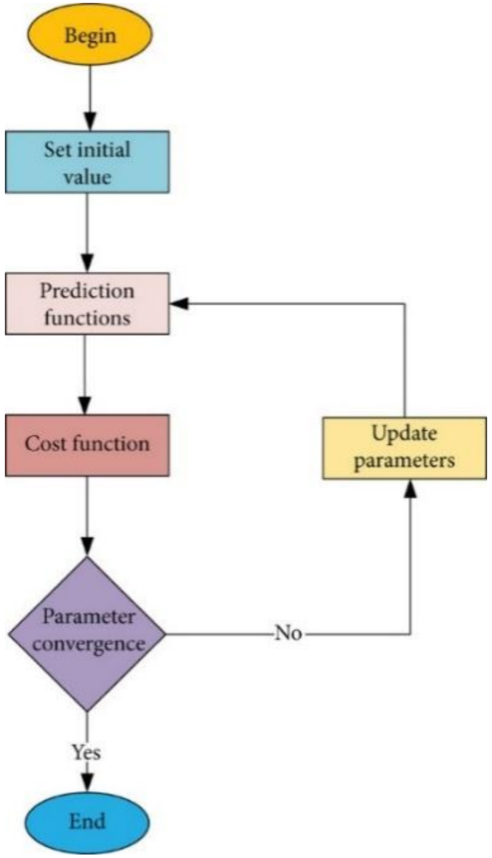
Key Concepts

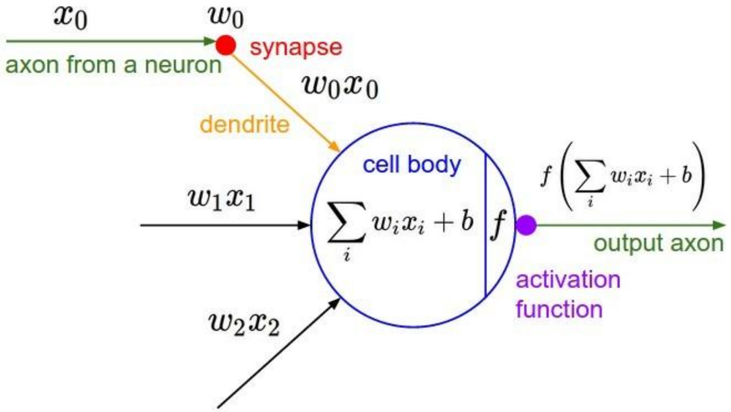
Algorithms and Models

- **Algorithms**
  - Step-by-step procedures or mathematical formulas used to process data and learn patterns.
  - In machine learning, algorithms are the methods used to train models on data.
  - **Example**: Linear Regression, Logistic Regression, Neural Networks.

- **Model**
  - The final outcome of a machine learning algorithm after being trained on data.
  - A models can make predictions or decisions when given new data.
  - **Example:** A trained model that predict house prices based on features like size, number of bedrooms, and location.



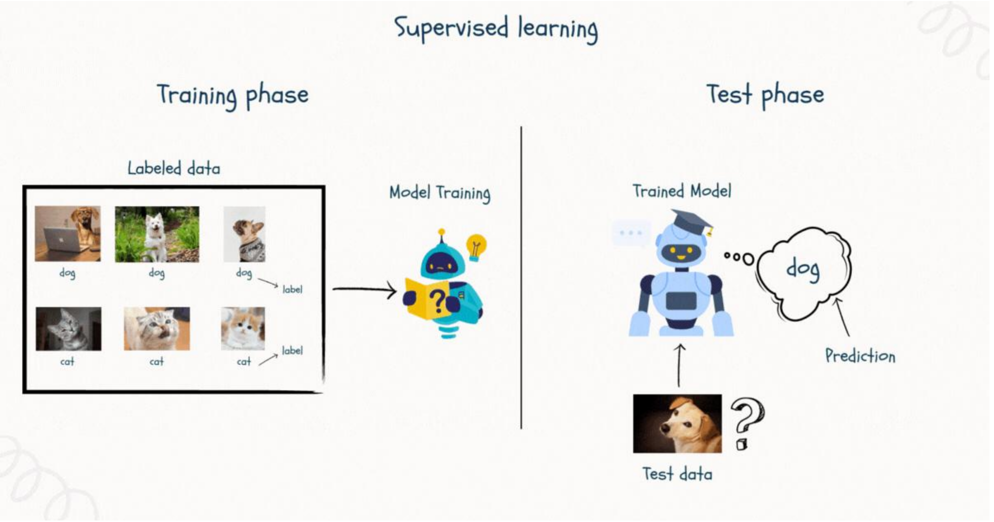


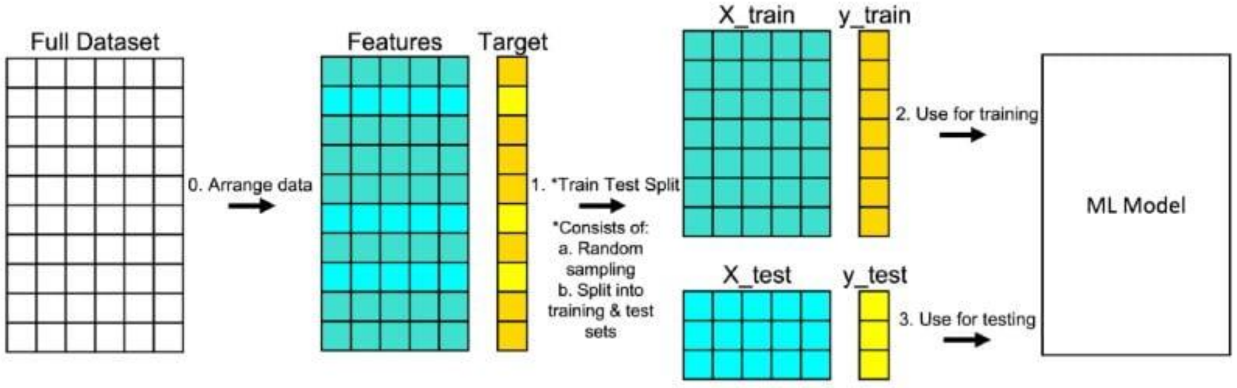


Key Concepts

Training and Testing

- **Training**
  - The process of teaching a machine learning model using a dataset.
  - The model learns patterns and relationships from the training data.
- **Testing**
  - The process of evaluating the trained model on a separate dataset (not seen during training).
  - Helps assess how well the model performs on new, unseen data and whether it generalizes effectively.





Key Concepts

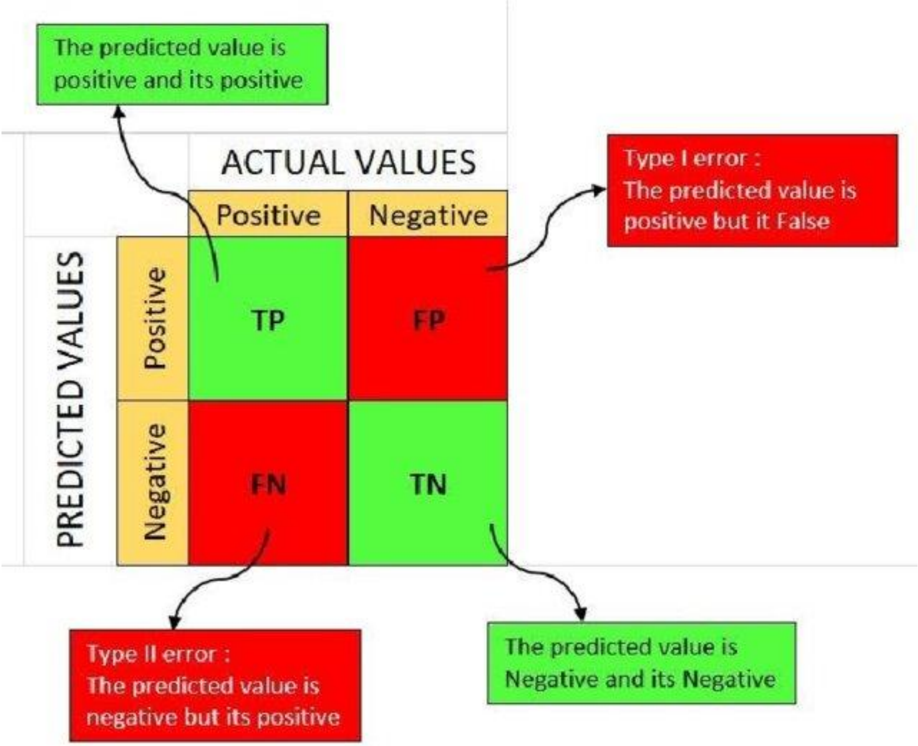
Evaluation Metrics

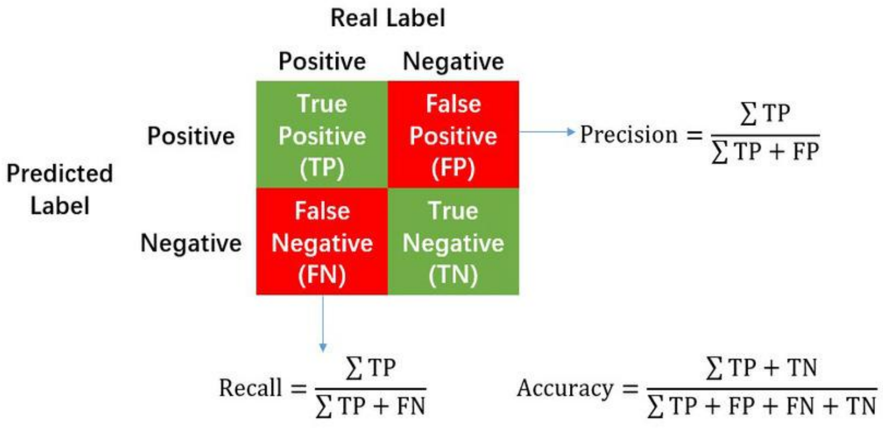
- Evaluation metrics are used to assess the performance of machine learning models.
- Different tasks require different metrics, depending on the type of prediction problem.

For Classification Tasks:

- **Accuracy**
  - The proportion of correct predictions out of all predictions made.
- **Precision**
  - The proportion of correctly predicted positive cases out of all predicted positives.
- **Recall**
  - The proportion of correctly predicted positive cases out of all actual positives.
- **F1 Score**
  - The harmonic mean of precision and recall — useful when class imbalance exists.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

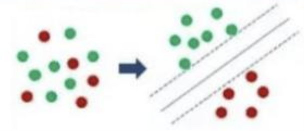
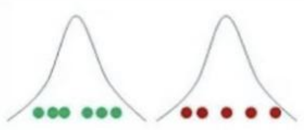
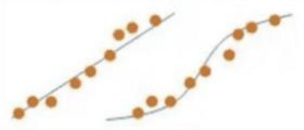
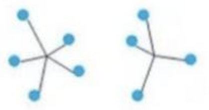
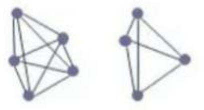
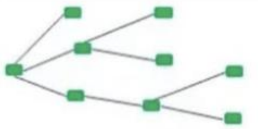
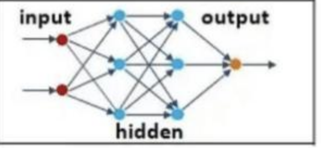




Common Algorithms

Supervised Learning

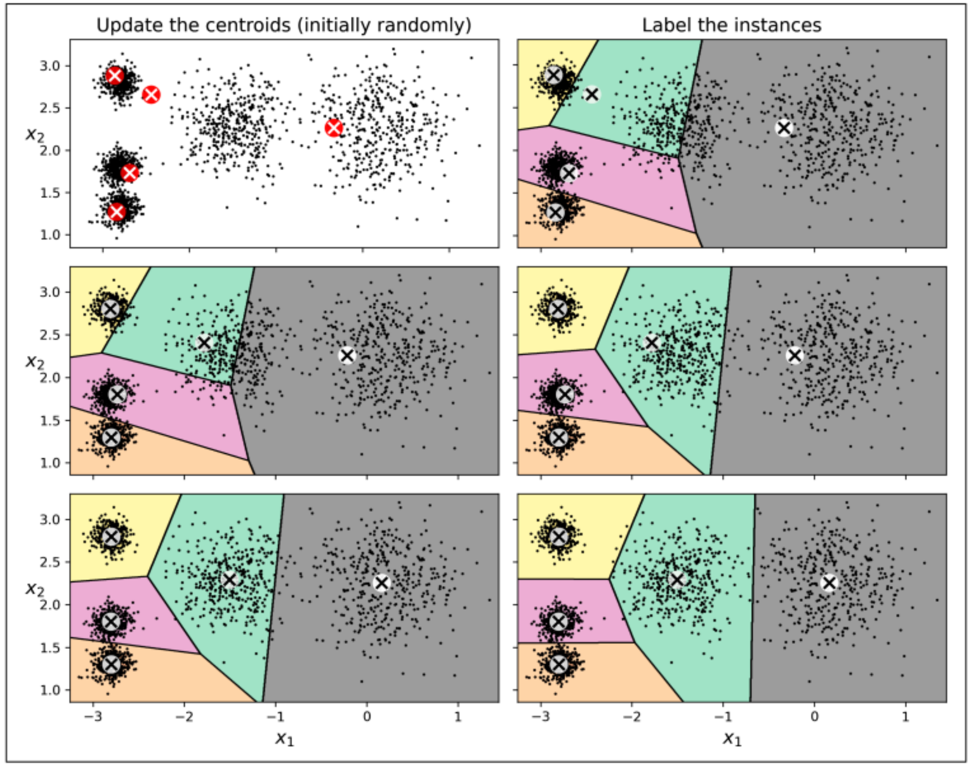
1. **Linear Regression:** Predicting continuous values.
2. **Logistic Regression:** Predicting discrete values.
3. **Decision Trees:** Making decisions based on data splits.
4. **Support Vector Machines (SVM):** Classifying data points.
5. **Neural Networks:** Mimicking brain functions for complex tasks.

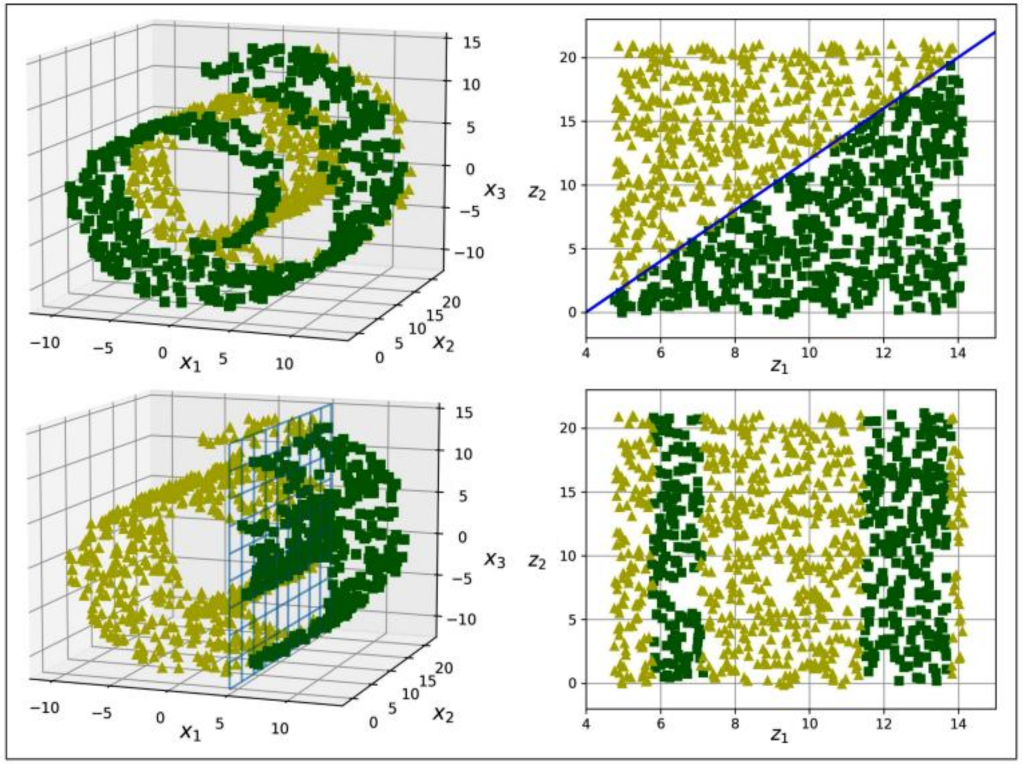
Algorithm	Keyword	Diagram
Support Vector Machines (SVM)	Vector on Points	
Naïve Bayes	Probability Distribution	
Linear Regression Logistic Regression	Straight Line Logarithmic Line	
K-Means	Kernel (*central*) Mean	
K-Nearest Neighbour	Neighbouring Points	
Decision Trees	Tree Branches	
Neural Networks	Network with Layers of elements	

Common Algorithms

Unsupervised Learning

1. **K-Means Clustering:** Grouping similar data points.
2. **Principal Component Analysis (PCA):** Reducing dimensionality.

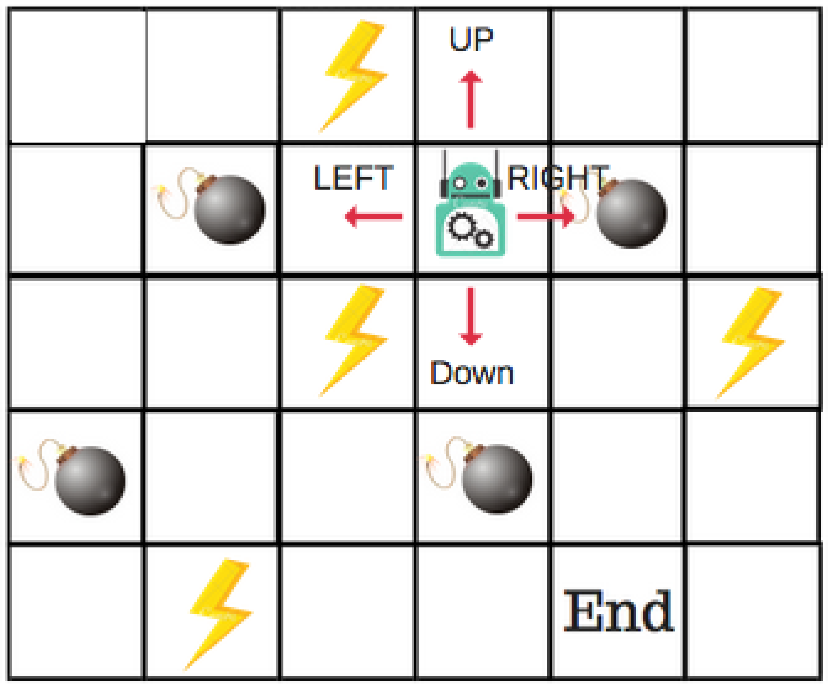


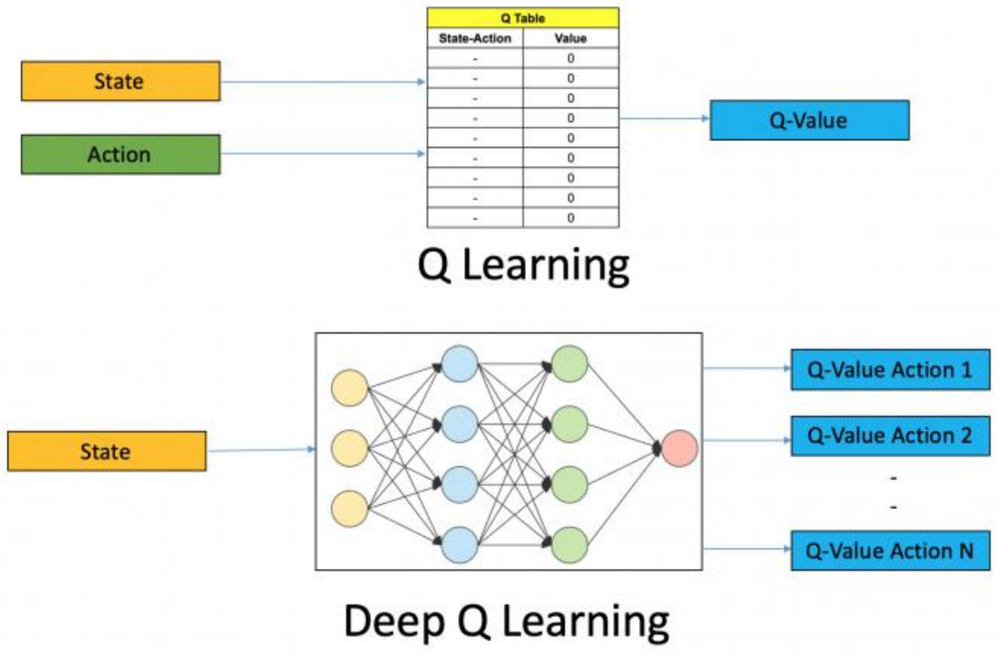


Common Algorithms

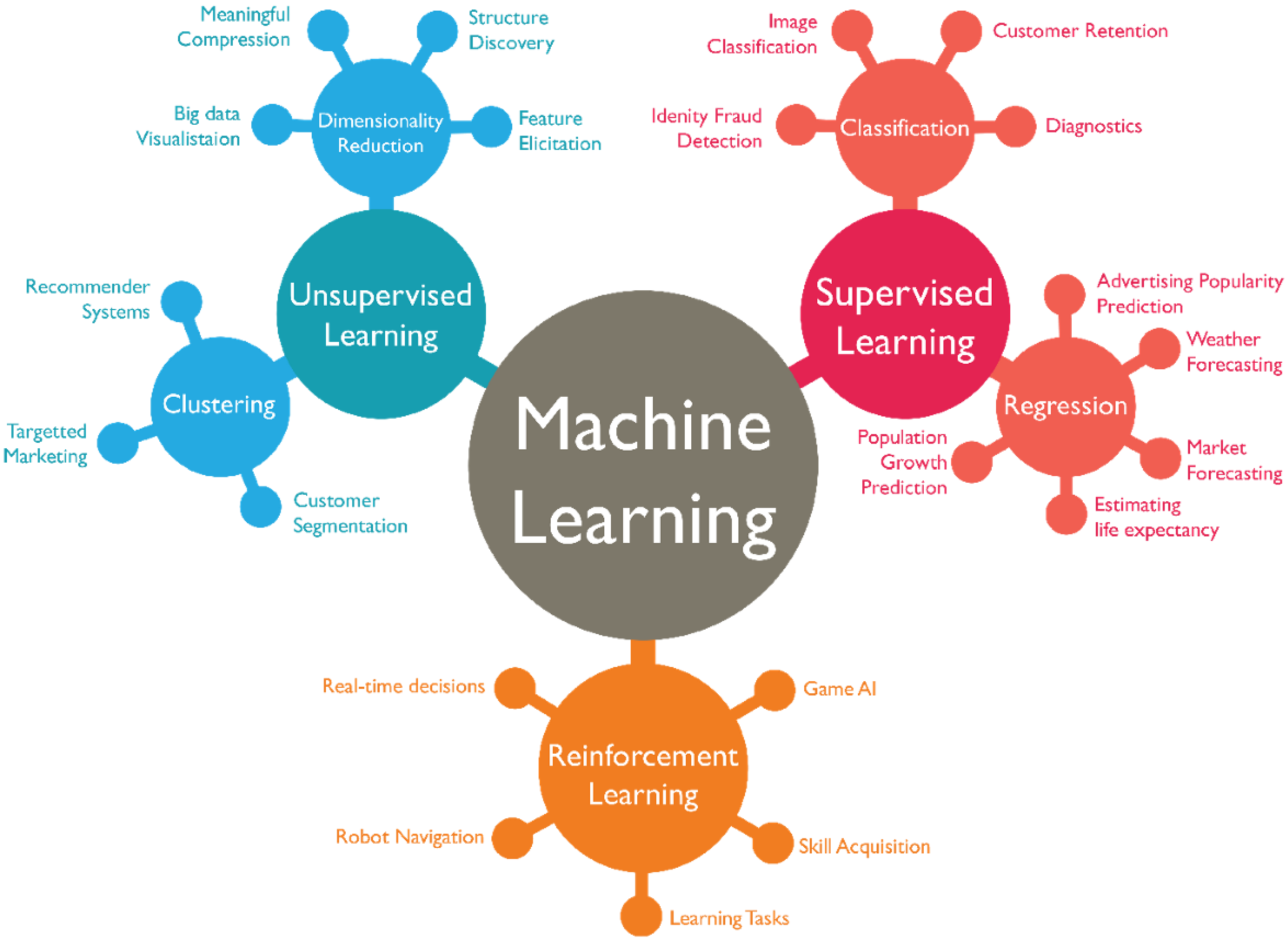
Reinforcement Learning

1. **Q-Learning:** Learning optimal actions.
2. **Deep Q-Networks (DQN):** Advanced game-playing models.



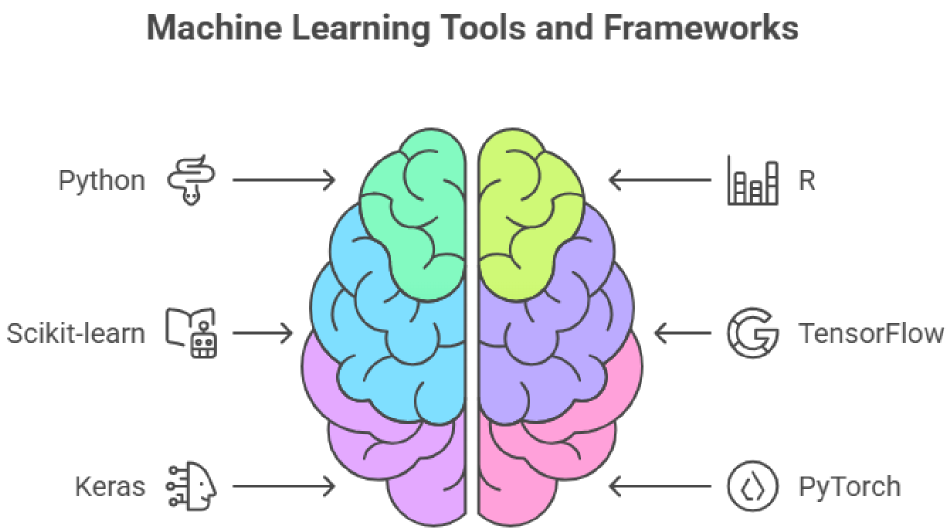


Types and Applications of Machine Learning



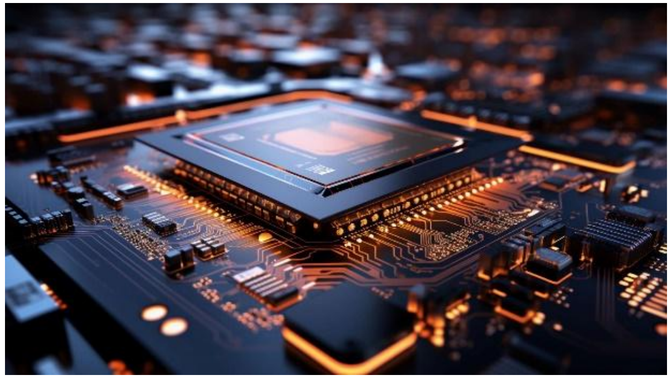
Tools and Frameworks

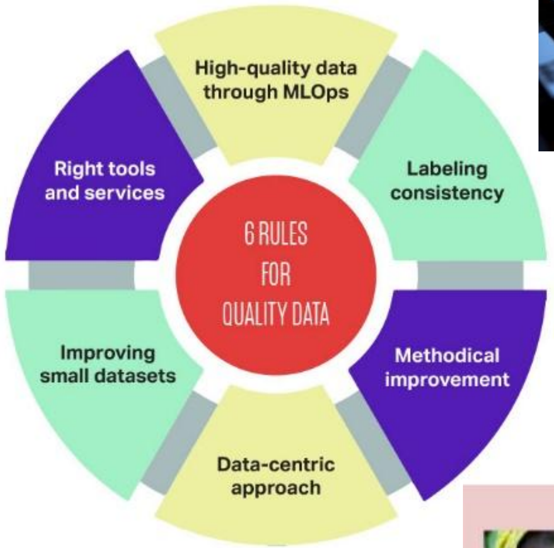
- **Programming Languages**
  - **Python:**
    - Most widely used language in ML.
    - Knows for its simplicity and rich ecosystem.
  - **R:**
    - Preferred for statistical analysis and data visualization, especially in academic and research settings.
- **Key Libraries and Frameworks**
  - **Scikit-learn:** A comprehensive library for classical machine learning algorithms (e.g., regression, classification, clustering), ideal for beginners and rapid development.
  - **TensorFlow:** A powerful open-source library developed by Google for building and deploying machine learning and deep learning models.
  - **Keras:** A high-level API running on top of TensorFlow; great for quick prototyping and building neural networks.
  - **PyTorch:** A flexible and widely adopted deep learning framework developed by Facebook; popular in both research and production.

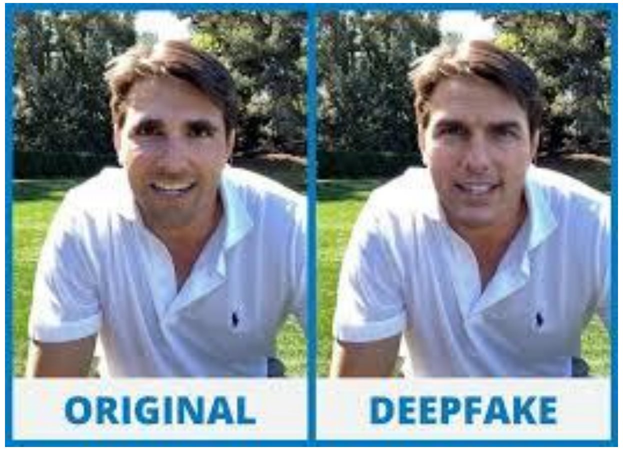


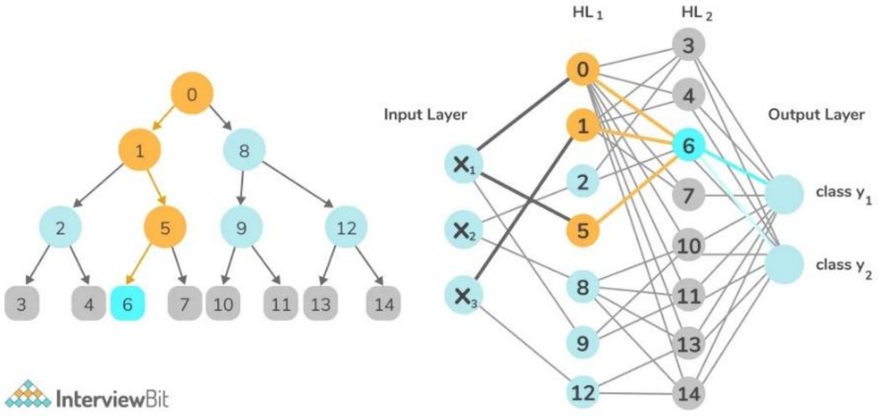
Challenges and Limitations

- **Data Quality and Quantity**
  - Need for large, high-quality datasets.
- **Computational Power**
  - High resource requirements.
- **Model Interpretability**
  - Difficulty in understanding complex models.
- **Ethical and Privacy Concerns**
  - Issues around data usage and bias.



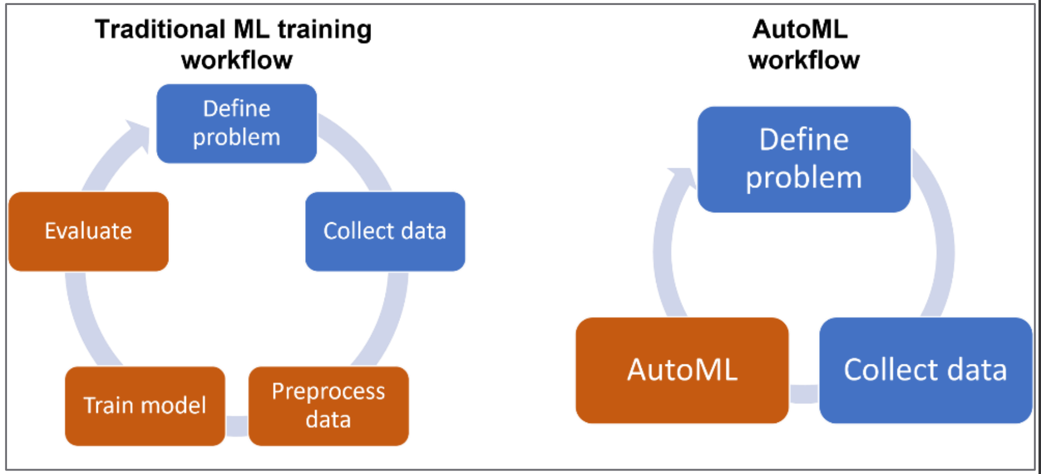


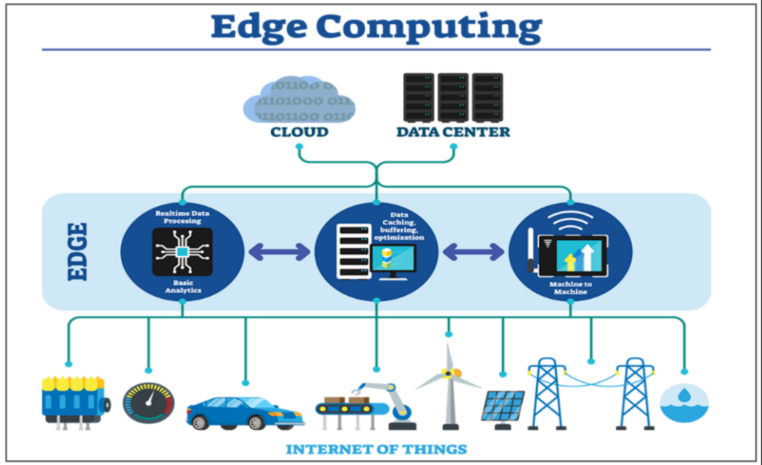


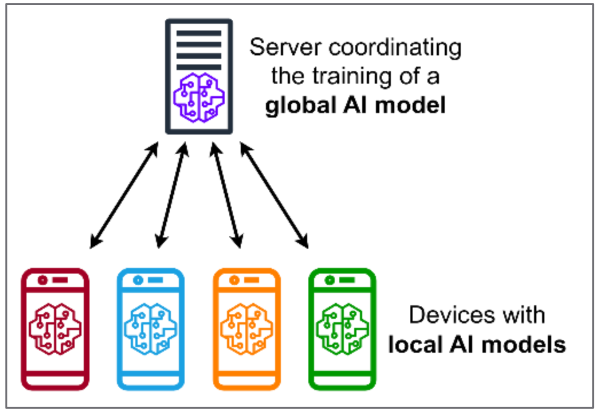


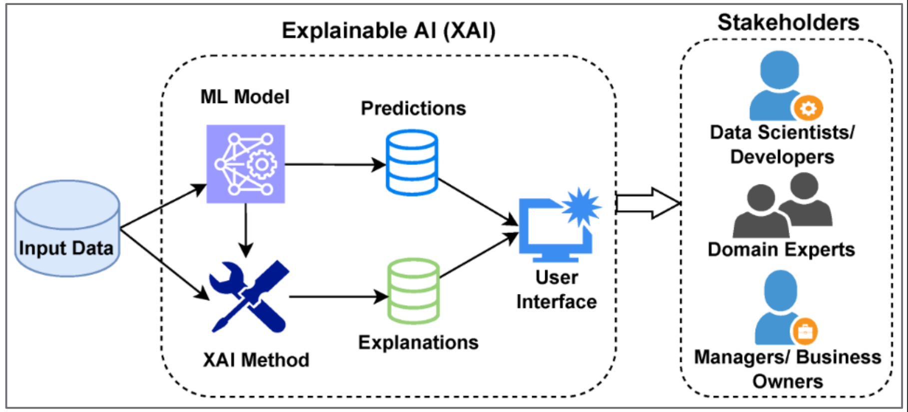
Emerging and Future Trends

- **AutoML**
  - Automates the process of building, training, and tuning ML models, making ML more accessible to non-experts.
- **Edge Computing**
  - Runs ML models directly on devices like smartphones, sensors, and IoT devices, reducing latency and dependence on the cloud.
- **Explainable AI (XAI)**
  - Focuses on making ML models more transparent and interpretable, helping users understand how decisions are made.
- **Federated Learning**
  - Enables training models across multiple devices or organizations without sharing raw data, enhancing privacy and data security.









Thank You