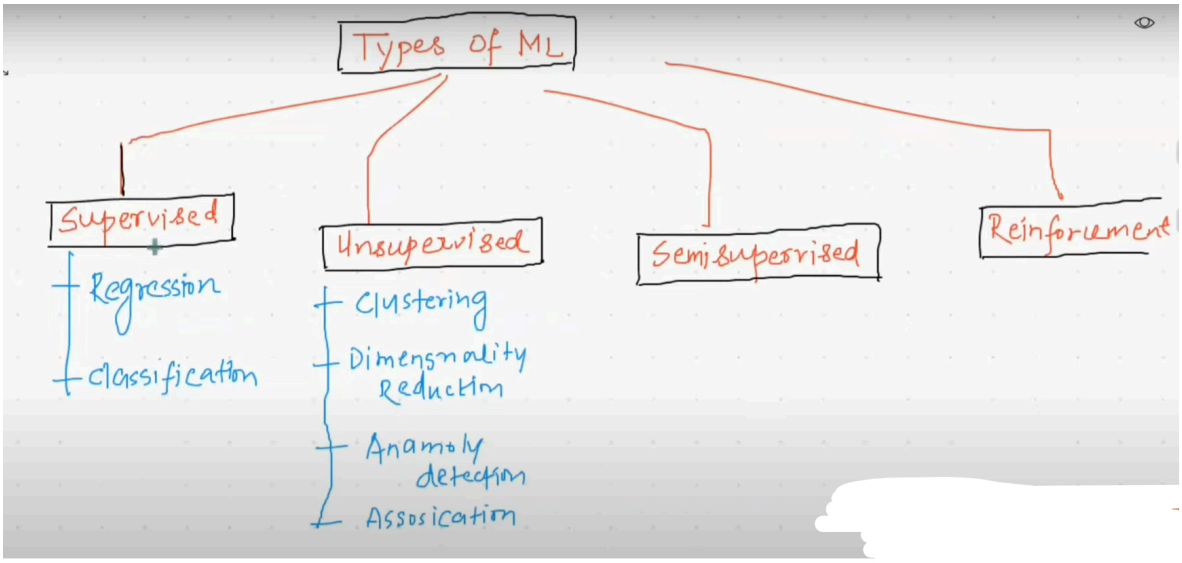
Machine Learning

Type of Machine Learning



1️⃣ **Supervised Learning** (Learning with labeled data)

2️⃣ **Unsupervised Learning** (Learning from patterns in unlabeled data)

3️⃣ **Reinforcement Learning** (Learning through rewards and punishments)

Type	Learns From?	Common Models	Example Use Case
Supervised	Given correct answers	🔷 Linear Regression 🔷 Decision Trees 🔷 Random Forest 🔷 SVM 🔷 Neural Networks 🔷 Naïve Bayes	Predict house prices, spam detection
Unsupervised	Hidden patterns	🔷 K-Means Clustering 🔷 DBSCAN 🔷 Hierarchical Clustering 🔷 PCA 🔷 Autoencoders	Group similar customers, anomaly detection

Type	Learns From?	Common Models	Example Use Case
Reinforcement	Rewards & penalties	🔷 Q-Learning 🔷 Deep Q-Network (DQN) 🔷 Proximal Policy Optimization (PPO) 🔷 Actor-Critic 🔷 Monte Carlo Methods	Teach robots to walk, self-driving cars

Supervised Learning Models (with Examples)

In supervised learning, the machine is given **labeled data** (data with correct answers) and learns to make predictions based on it.

- **Think of it like a student learning with an answer key.**
- The model is trained on input-output pairs and tries to generalize for new inputs.

🛠️ How does it work?

- The model sees **input data (X)** and its **correct output (Y)**.
- It learns a **pattern** between X and Y.
- Once trained, it can predict **Y for new X values**.

📂 Common applications

✔️ Spam detection (Email: spam or not spam)

✔️ Medical diagnosis (X-ray: disease or no disease)

✔️ Credit scoring (Loan: approve or reject)

🔑 Trivia

✅ 80% of real-world ML problems are **supervised learning**!

✅ It needs **lots of labeled data**, which can be expensive to collect.

Regression vs Classification

- If output column in numerical→ **Regression**
- If output column in Categorical→ **Classification**

1. Linear Regression

- **What it does**: Predicts a continuous value (like house prices) based on input features (like size, location).
- **Example**: Predicting the price of a house based on its size.
- **How it works**: It draws a straight line through the data points to make predictions.

2. Decision Trees

- **What it does**: Makes decisions by splitting data into branches based on features.
- **Example**: Deciding if an email is spam based on words like "free" or "win."
- **How it works**: It asks a series of yes/no questions (e.g., "Does the email contain the word 'free'?") to classify data.

3. Random Forest

- **What it does**: Combines multiple decision trees to improve accuracy.
- **Example**: Predicting whether a loan applicant will default.
- **How it works**: It builds many decision trees and takes a "vote" from all of them to make a final decision.

4. Support Vector Machines (SVM)

- **What it does**: Finds the best boundary (line or plane) to separate data into classes.
- **Example**: Classifying images of cats and dogs.

- **How it works**: It draws a line (or **hyperplane**) that maximizes the gap between two classes.

5. Neural Networks

- **What it does**: Mimics the human brain to learn complex patterns.
- **Example**: Recognizing handwritten digits.
- **How it works**: It uses layers of "neurons" to process data and make predictions.

Unsupervised Learning Models (with Examples)

- **ONLY INPUT**
- In unsupervised learning, the machine is given **only input data (X) with no correct answers (Y)** and must find **hidden patterns or groupings**.
- **Think of it like a child sorting toys into groups without being told how.**
- The machine groups similar things together based on patterns.

🛠️ How does it work?

- The model finds **structure** in the data without labels.
- It **clusters** similar data points or reduces data complexity.

📌 Example: Customer segmentation

Customer	Age	Spending (₹ per month)
A	25	5,000
B	35	50,000
C	22	6,000
D	40	55,000

📌 The model **groups** customers:

🔷 Young & low spenders (A, C)

🔷 Older & high spenders (B, D)

✨**Common Algorithms**:

- **Clustering** (e.g., K-Means, DBSCAN)
- **Dimensionality Reduction** (e.g., PCA, t-SNE)

Examples:

- Grouping customers into distinct segments based on purchasing behavior (clustering).
- Reducing the number of features in a dataset for better visualization or to improve model efficiency (dimensionality reduction).

📂 Common applications

✔️ Customer segmentation (Marketing: group similar buyers)

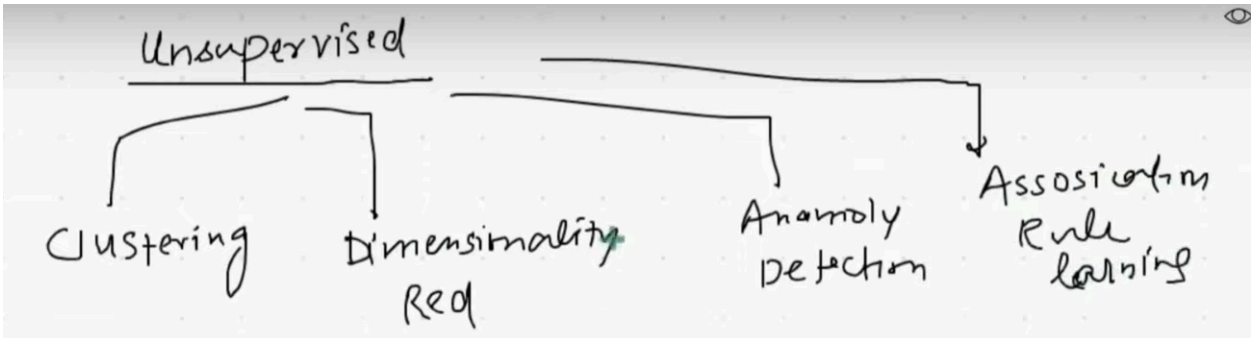
✔️ Anomaly detection (Fraud detection: find unusual transactions)

✔️ Topic modeling (News articles: group by topics)

🔑 Trivia

✅ Unsupervised learning is widely used in **big data analytics**.

✅ Unlike supervised learning, it **doesn't need labeled data**.



1. K-Means Clustering

- **What it does**: Groups data into a fixed number of clusters (groups).
- **Example**: Grouping customers based on shopping habits.
- **How it works**: It finds the center of each cluster and assigns data points to the nearest center.

2. Hierarchical Clustering

- **What it does**: Creates a tree-like structure of clusters.
- **Example**: Grouping animals into species based on traits.
- **How it works**: It starts with each data point as its own cluster and merges similar clusters step by step.

3. Principal Component Analysis (PCA)

- **What it does**: Reduces the number of features in data while keeping the most important information.
- **Example**: Simplifying a dataset with 100 features to just 2 or 3 for visualization.
- **How it works**: It finds the directions (components) in the data that explain the most variance.

4. DBSCAN

- **What it does**: Groups data based on density (how close points are to each other).
- **Example**: Detecting outliers in fraud detection.
- **How it works**: It identifies dense regions of data points and separates them from sparse regions.

Reinforcement Learning Models (with Examples)

In reinforcement learning (RL), the machine learns by **trial and error**, getting **rewards for good actions** and **penalties for bad actions**.

- **Think of it like training a dog.**
- If the dog sits when told, you give a treat (reward).
- If it misbehaves, you scold it (punishment).
- Over time, the dog learns what to do to get more treats.

🛠️ How does it work?

- The agent (AI model) interacts with the **environment**.
- It takes **actions** based on a strategy (policy).
- It gets **rewards or penalties** and updates its policy.

📌 Example: Training a robot to walk

- If the robot moves correctly, it gets **+10 points** (reward).
- If it falls, it gets **5 points** (penalty).
- Over time, it learns the best way to walk without falling.

📂 Common applications

✔️ Robotics (Self-learning robots like Boston Dynamics)

✔️ Gaming (AI in chess, AlphaGo, and OpenAI's Dota 2 bot)

✔️ Self-driving cars (Learn to navigate roads safely)

🔑 Trivia

✅ Reinforcement learning is behind **game-playing AI like AlphaGo**.

✅ It is used in **stock trading and supply chain optimization**.

1. Q-Learning

- **What it does**: Learns the best action to take in a given state to maximize rewards.
- **Example**: Teaching a robot to navigate a maze.

- **How it works**: It builds a table (Q-table) that stores the best action for every possible state.

2. Deep Q-Networks (DQN)

- **What it does**: Combines **Q-learning with neural networks** to handle complex environments.
- **Example**: Playing video games like Atari Breakout.
- **How it works**: It uses a neural network to approximate the Q-table, making it scalable to large state spaces.

3. Policy Gradients

- **What it does**: Learns a policy (strategy) directly instead of a Q-table.
- **Example**: Training a robot to walk.
- **How it works**: It adjusts the policy to increase the likelihood of actions that lead to higher rewards.

4. Monte Carlo Tree Search (MCTS)

- **What it does**: Simulates possible future actions to decide the best move.
- **Example**: Playing board games like Go or Chess.
- **How it works**: It builds a tree of possible moves and evaluates the best path to take.

Fun Trivia

- **Linear Regression** is one of the oldest ML models, dating back to the 1800s!
- **K-Means Clustering** is used by Netflix to group users with similar tastes in movies.
- **Q-Learning** was used to train the first AI to beat a world champion in Backgammon in the 1990s.

Which Model Should You Use?

- Use **Linear Regression** or **Decision Trees** for simple supervised learning tasks.
- Use **K-Means** or **PCA** for exploring patterns in unlabeled data.
- Use **Q-Learning** or **DQN** for training agents in games or robotics.

Brief Explanation of Each Model

🔷 Supervised Learning Models

✔️ **Linear Regression** – Predicts a continuous value (e.g., house price).

✔️ **Decision Trees** – Splits data into decision paths (e.g., loan approval).

✔️ **Random Forest** – Multiple decision trees combined for better accuracy.

✔️ **Support Vector Machines (SVM)** – Draws a boundary to separate classes.

✔️ **Neural Networks** – Mimics the human brain to learn complex patterns.

✔️ **Naïve Bayes** – Uses probability (common in spam filters).

🔷 Unsupervised Learning Models

✔️ **K-Means Clustering** – Groups data points into clusters (e.g., customer segmentation).

✔️ **DBSCAN** – Detects clusters with varying shapes (good for anomaly detection).

✔️ **Hierarchical Clustering** – Builds a hierarchy of clusters.

✔️ **PCA (Principal Component Analysis)** – Reduces data complexity while keeping important features.

✔️ **Autoencoders** – Neural networks that compress and reconstruct data.

🔷 Reinforcement Learning Models

✔️ **Q-Learning** – A basic RL algorithm that learns an action-reward table.

✔️ **Deep Q-Network (DQN)** – Uses deep learning to improve Q-learning.

✔️ **Proximal Policy Optimization (PPO)** – Balances exploration and exploitation for stable learning.

✔️ **Actor-Critic** – Uses two models: one for decision-making (actor) and one for evaluating (critic).

✔️ **Monte Carlo Methods** – Learns by averaging outcomes over multiple simulations.

How ML Models are Trained?

Batch Vs Online ML

1️⃣ Batch Learning (Offline Learning)

🔷 What is it?

- The model is trained on the **entire dataset at once**.
- It doesn't update with new data unless retrained from scratch.
- Used when data **doesn't change frequently** or is available in large chunks.

🛠️ How does it work?

1. Collect **all data** first.
2. Train the model **once** on this data.
3. Use the trained model for predictions.
4. If new data comes, retrain from scratch (costly process).

📂 Common applications

✔️ Fraud detection (Train once on past transactions)

✔️ Image recognition (Train once on millions of labeled images)

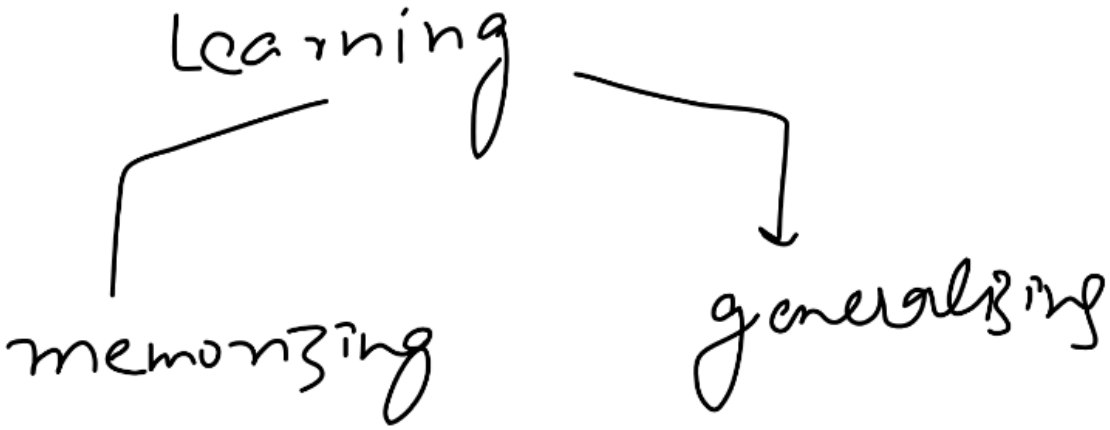
✔️ Medical diagnosis (Train once on historical patient data)

🔑 Pros & Cons

Pros (✅ Advantages)	Cons (❌ Disadvantages)

Simpler and more stable model	Needs a lot of memory (big dataset)
Can use all data at once (high accuracy)	Cannot adapt to new data unless retrained
Works well when data is static	Computationally expensive to retrain

Instance-Based vs. Model-Based Learning



1️⃣**Instance-Based Learning**(Memorization)

2️⃣**Model-Based Learning**(Generalization)

1️⃣ Instance-Based Learning (Lazy Learning)

🔷 What is it?

- The algorithm **memorizes** the training data.
- No real "learning" happens during training—data is simply stored.
- When a new query comes in, it **compares it to stored instances** and makes a prediction.
- **Fast training but slow predictions** (because it must search through data).

🛠️ How does it work?

1. Store all training examples in memory.
2. When making a prediction, find the most **similar** stored example(s).
3. Use these to determine the output (e.g., majority vote, average value).

Example: Nearest Neighbor Voting (KNN)

Let's say we have a dataset of animals based on height and weight.

Height	Weight	Animal
20 cm	5 kg	Cat
50 cm	20 kg	Dog
15 cm	3 kg	Cat

📌 If a **new animal** has **Height = 22 cm, Weight = 6 kg**, it will be compared to stored instances and classified as **Cat** (because it's closest to other cats).

📂 Common Models

✔️ **K-Nearest Neighbors (KNN)** – Finds the closest neighbors in memory.

✔️ **Support Vector Machines (SVM, some versions)** – Uses stored data to compare new instances.

🔑 Pros & Cons

Pros (✅ Advantages)	Cons (❌ Disadvantages)
No training time – Just store the data	**Slow predictions** – Must compare new data to all stored data
Works well with small datasets	**Memory-intensive** – Stores all data in memory
Can handle complex patterns	**Sensitive to noise** – Outliers can mislead results

Model-Based Learning (Eager Learning)

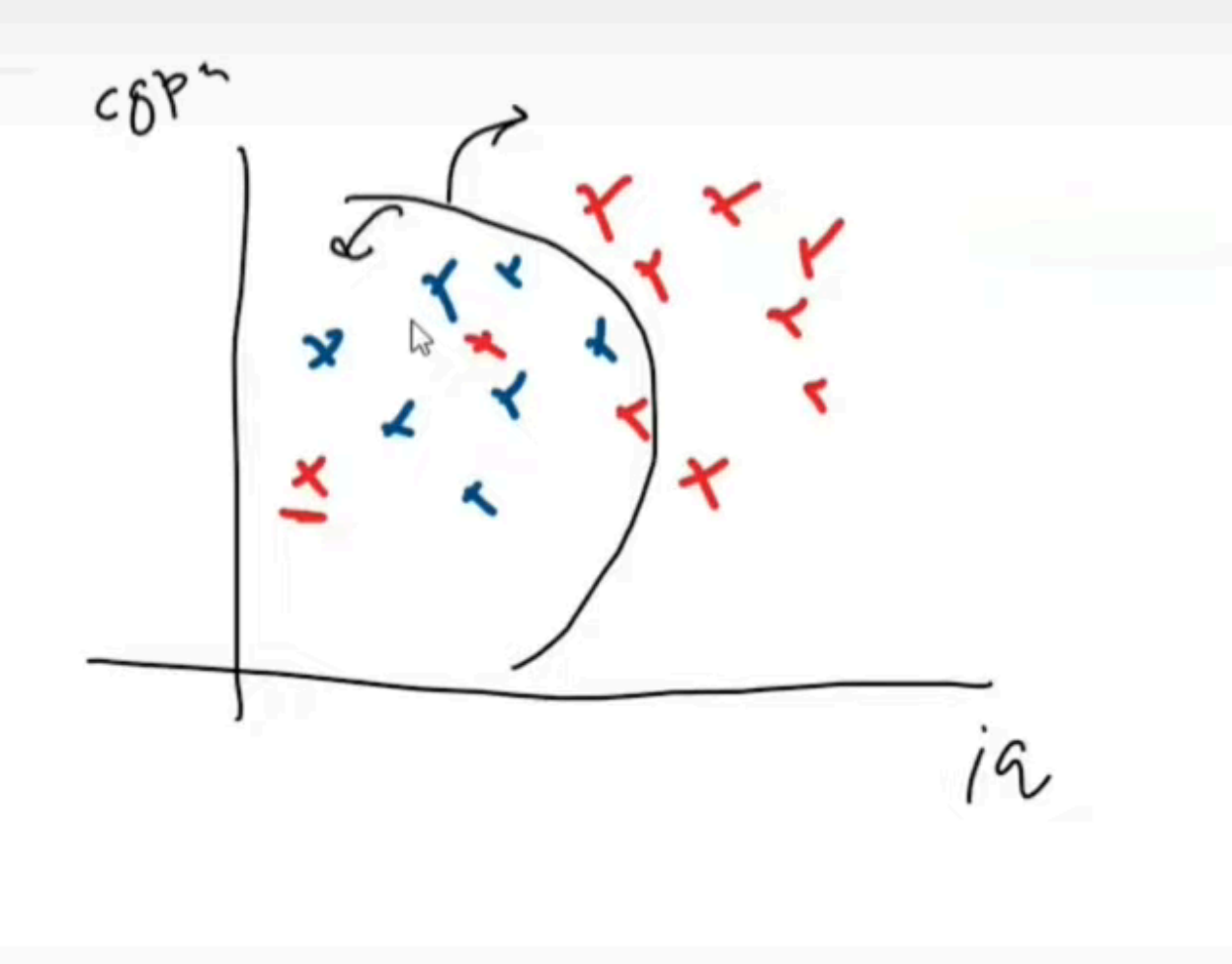
💡 **Most of the algorithms are model based learning.**

🔷 What is it?

- The algorithm **creates a model** from training data.
- It doesn't store all instances but instead **learns patterns** and **generalizes** from them.
- When a new query comes in, it **uses the trained model** to predict instead of searching stored examples.
- **Slow training but fast predictions** (because the model is pre-built).

🛠️ How does it work?

1. Analyze all training data and find **patterns**.
2. Build a **mathematical model** (e.g., equation, decision rules).
3. Use this model to make predictions without storing the raw data.



📌 *Example: Linear Regression*

Suppose we want to **predict house prices** based on square footage.

Size (sq ft)	Price ($1000s)
1000	150
2000	250
3000	350

📌 The model finds the **best-fit line**:

Price = 100 + 0.1 × (Size in sq ft)

Now, for a **new house** of **2500 sq ft**, the model predicts:

Price = 100 + 0.1 × (2500) = 350 ($1000s)

✅ No need to store all training examples.

📂 Common Models

✔️ **Linear Regression** – Finds a mathematical relationship between inputs and outputs.

✔️ **Decision Trees** – Creates rules to make predictions.

✔️ **Neural Networks** – Mimics the human brain to learn patterns.

🔑 Pros & Cons

Pros (✅ Advantages)	Cons (❌ Disadvantages)
Fast predictions – Uses a trained model	**Training takes time** – Needs computation to build the model
Works well for large datasets	**May lose details** – If the model oversimplifies, it may not capture all patterns
Memory-efficient – Doesn't store all instances	**Requires good model selection** – Choosing the wrong model can lead to poor results

Comparison Table: Instance-Based vs. Model-Based Learning

Feature	Instance-Based (Lazy)	Model-Based (Eager)
Learning Style	Memorizes data	Generalizes from patterns
Training Time	**Fast** (no model building)	**Slow** (must build a model)
Prediction Time	**Slow** (must compare all stored examples)	**Fast** (just uses model)
Memory Usage	**High** (stores all data)	**Low** (only stores model)
Best For	Small datasets, complex patterns	Large datasets, fast real-time predictions
Example Models	KNN, Some SVM	Linear Regression, Decision Trees, Neural Networks

Which One to Use?

✔️ **Use Instance-Based Learning** if:

- You have **small datasets** where storing examples is okay.

- You want **flexibility** (works well on complex, irregular patterns).
- You need **no training time** (e.g., recommendation systems, anomaly detection).

✔️ **Use Model-Based Learning** if:

- You need **fast real-time predictions** (e.g., self-driving cars, fraud detection).
- You have **large datasets** where memorization isn't practical.
- You want a **generalized model** that works on new data.

Machine Learning Development Life Cycle(MLDLC/MLDC)

Problem Definition → Data Collection → Data Preparation → Model Selection → Model Training → Model Evaluation → Model Deployment → Monitoring and Maintenance

📌 Key Stages:

1️⃣ **Problem Definition** (What are we solving?)

2️⃣ **Data Collection** (Getting the right data)

3️⃣ **Data Preprocessing** (Cleaning and preparing data)

4️⃣ **Feature Engineering** (Extracting useful information)

5️⃣ **Model Selection** (Choosing the right algorithm)

6️⃣ **Model Training** (Teaching the model)

7️⃣ **Model Evaluation** (Testing performance)

8️⃣ **Hyperparameter Tuning** (Optimizing the model)

9️⃣ **Deployment** (Making the model available)

🔟 **Monitoring & Maintenance** (Ensuring long-term performance)

1️⃣ Problem Definition

🔷 **Why is this important?**

- Defines the **business goal** (e.g., fraud detection, demand forecasting).
- Identifies **what type of ML problem it is** (Classification? Regression?).

📌 **Example:**

- A bank wants to **predict loan default** risks.
- Type: **Classification (Yes/No: Will default or not?)**

2️⃣ Data Collection

🔷 **Why is this important?**

- ML models learn from data, so **quality & quantity** matter.
- **Sources:** Databases, APIs, Web Scraping, User Input, Sensors.

📌 **Example:**

- A healthcare ML model collects **patient records** (age, symptoms, test results).
- Collect emails labeled as "spam" or "not spam."

3️⃣ Data Preprocessing (Cleaning)

🔷 **Why is this important?**

- Raw data is **messy**: It contains **missing values, duplicates, outliers**.
- Preprocessing makes data **usable** for training.

✅ **Common Steps:**

✔️ Remove **duplicates**

✔️ Handle **missing values** (drop or fill)

✔️ Normalize / Scale data

✔️ Convert **categorical data** to numerical

📌 **Example:**

- If a dataset has **missing age values**, we can **fill with the average age**.

4️⃣ Feature Engineering

🔷 **Why is this important?**

- Features are **input variables** used for prediction.
- Creating better features = **better model accuracy**.

✅ **Common Techniques:**

✔️ **Feature Scaling** (Standardization/Normalization)

✔️ **Feature Extraction** (e.g., Extracting "Day of Week" from date)

✔️ **Feature Selection** (Removing irrelevant features)

📌 **Example:**

- Instead of using **"date of birth"**, use **"age"** (more useful for models).

5️⃣ Model Selection

🔷 **Why is this important?**

- Choosing the right **ML algorithm** affects performance.
- Depends on **data size, problem type, and complexity**.

✅ **Common ML Algorithms:**

Problem Type	Algorithm
Regression (Predict continuous values)	Linear Regression, Random Forest, Neural Networks
Classification (Yes/No, Categories)	Logistic Regression, SVM, Decision Trees, CNNs
Clustering (Grouping similar items)	K-Means, DBSCAN, Hierarchical Clustering

📌 **Example:**

- For **predicting house prices**, use **Linear Regression**.

6️⃣ Model Training

🔷 **Why is this important?**

- The model **learns patterns** from training data.
- The goal is to **minimize error** using mathematical techniques.

✅ **Steps:**

✔️ Split data into **Training Set (80%)** and **Test Set (20%)**.

✔️ Feed training data into the model.

✔️ Adjust model parameters to **reduce prediction errors**.

📌 **Example:**

- A self-driving car model **learns from millions of driving images**.

7️⃣ Model Evaluation

🔷 **Why is this important?**

- Checks if the model is **accurate and reliable**.
- Uses **performance metrics** to measure success.

✅ **Common Evaluation Metrics:**

Problem Type	Metric
Regression	Mean Squared Error (MSE), R^2 Score
Classification	Accuracy, Precision, Recall, F1 Score

📌 **Example:**

- A spam detection model achieves **92% accuracy** → Good model!

8️⃣ Hyperparameter Tuning

🔷 **Why is this important?**

- Fine-tuning **hyperparameters** improves model performance.
- Hyperparameters are **settings** we adjust manually (e.g., Learning Rate, Number of Neurons).

✅ **Techniques:**

✔️ **Grid Search** – Tries all combinations of hyperparameters.

✔️ **Random Search** – Tests random sets of values.

✔️ **Bayesian Optimization** – Predicts the best combination.

📌 **Example:**

- Tuning the **depth of a decision tree** to **reduce overfitting**.

9️⃣ Deployment (Making the Model Available)

🔷 **Why is this important?**

- The model is **packaged and integrated** into a real-world system.
- Users can now **input new data** and get predictions.

✅ **Deployment Methods:**

✔️ Web API (Flask, FastAPI)

✔️ Mobile Apps

✔️ Embedded Devices (IoT)

📌 **Example:**

- A chatbot AI is deployed on **a website to answer customer queries**.

🔟 Monitoring & Maintenance

🔷 **Why is this important?**

- Models **degrade over time** if real-world data changes (data drift).
- Continuous monitoring **prevents poor predictions**.

✅ **Common Issues:**

✔️ **Data Drift** – New data trends make the model outdated.

✔️ **Concept Drift** – Relationships between inputs and outputs change.

✔️ **Bias Detection** – The model may become unfair over time.

📌 **Example:**

- A **fraud detection model** is retrained every month with new fraud patterns.

📌 Summary Table: ML Development Life Cycle

Step	Description	Example
1. Problem Definition	Define the ML goal	Predict loan default
2. Data Collection	Gather data from sources	Bank transaction data
3. Data Preprocessing	Clean and structure data	Remove missing values
4. Feature Engineering	Extract meaningful features	Convert "DOB" → "Age"
5. Model Selection	Choose the right algorithm	Logistic Regression for classification
6. Model Training	Train the model using data	Train fraud detection on past cases
7. Model Evaluation	Test performance using metrics	Check accuracy, recall, etc.
8. Hyperparameter Tuning	Optimize the model settings	Adjust learning rate in Neural Network
9. Deployment	Integrate model into real-world system	Deploy spam filter for emails
10. Monitoring & Maintenance	Continuously update model	Retrain model every month

🚀 **Key Takeaways:**

✅ MLDLC ensures a structured way to build **reliable ML systems**.

✅ **Good data → Better model** (Garbage in, garbage out).

✅ **Model monitoring is crucial** (to keep predictions accurate).

	ANALYTICAL SKILLS	BUSINESS ACUMEN	DATA STORYTELLING	SOFT SKILLS	SOFTWARE SKILLS
DATA ANALYST	HIGH	MEDIUM TO HIGH	HIGH	MEDIUM TO HIGH	MEDIUM
DATA ENGINEER	MEDIUM	LOW	LOW	MEDIUM	HIGH
DATA SCIENTIST	HIGH	HIGH	HIGH	HIGH	MEDIUM
ML ENGINEER	MEDIUM TO HIGH	MEDIUM	LOW	HIGH	HIGH