

Machine Learning

(BE Computer 2019 PAT)

A.Y. 2025-26

Prepared by

Payal Hake

Unit-1 Introduction To Machine Learning

- Introduction to Machine Learning, Comparison of Machine learning with traditional programming, ML vs AI vs Data Science. **1 Hr.**
- Types of learning: Supervised, Unsupervised, and semi-supervised, reinforcement learning techniques, **1 Hr.**
- Models of Machine learning: Geometric model, Probabilistic Models, Logical Models, Grouping and grading models, **2Hr**
- Parametric and non-parametric models. **1 Hr.**
- Important Elements of Machine Learning- Data formats
- Learnability **1 Hr.**
- Statistical learning approaches **1 Hr.**

Introduction to Machine Learning

- **Term "Machine Learning" was coined by Arthur Samuel in 1959 at IBM.**
- **He defined it as:**
“The field of study that gives computers the ability to learn without being explicitly programmed.”
- **There is no single universal definition — different experts describe it in various ways.**
- **In simple terms:**
Machine learning means teaching computers to get better at a task by learning from examples or past experience, without writing every step manually.

What is Machine Learning?

- **Machine Learning** is a field that focuses on creating computer programs that **automatically improve through experience**.
- A **model** is created with some adjustable parts (**parameters**).
- **Learning** means adjusting those parameters using **training data** or past experience.

Types of Machine Learning Models:

- **Predictive Models**
 - Used to **make predictions** about future or unseen data.
 - *Example:* Predicting stock prices, weather forecasting.
- **Descriptive Models**
 - Used to **understand patterns** or gain insights from existing data.
 - *Example:* Analyzing customer behavior to find purchase patterns.

Examples:

- **Definition of learning:** A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**.
- if its performance at tasks **T**, as measured by **P** improves with experience **E**.
- **Examples Handwriting recognition learning problem**
 - **Task T** : Recognizing and classifying handwritten words within images.
 - **Performance P** : Percent(%) of words correctly classified.
 - **Training experience E** : A dataset of handwritten words with given classifications

Examples:

A robot driving learning problem

- **Task T** : Driving on highways using vision sensors.
- **Performance P** : Average distance traveled before an error.
- **Training experience E** : A sequence of images and steering commands recorded while observing a human driver.

Definition: A computer program which **learns from experience** is called a machine learning program or simply a learning program .

Machine Learning Examples

- Face recognition on your phone or voice understanding
- Diagnose diseases by symptoms (Watson)
- Advise product
- Books (Amazon)
- Movies (Netflix)
- Music (Spotify)



Comparison of Machine learning with traditional programming

TRADITIONAL PROGRAMMING



MACHINE LEARNING



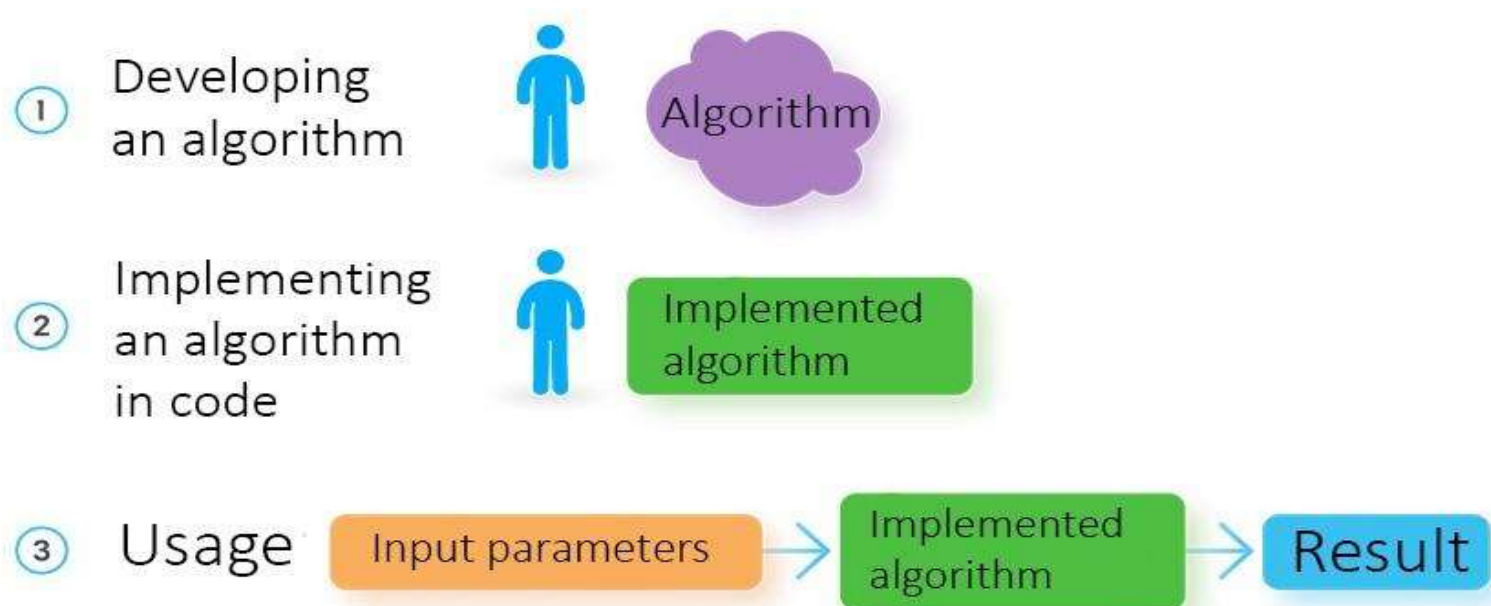
- Traditional programming you **hard code** the behavior of the **program**.
- In machine learning, you leave a lot of that to the **machine** to learn from data.

ML as a Supplement, Not a Replacement

- **Machine Learning (ML)** is not a replacement for traditional programming — it **works alongside** it.
- ML is used when **regular programming methods fall short**, especially for tasks like predictions or pattern recognition.
- Example:
 - In an **online trading platform**:
 - **ML** handles predictions (like stock trends).
 - **Traditional programming** (e.g., Java, Ruby) is used for UI, data display, etc.
- ML is **best suited** for complex tasks where **rules can't be clearly defined**.

Traditional programming approach

- First, **design the most suitable algorithm** for the problem.
- Then, **write the code** based on that algorithm.
- Next, **set the correct input parameters** for the program to work.
- If the algorithm is correct and properly implemented, it will give the **expected output**.

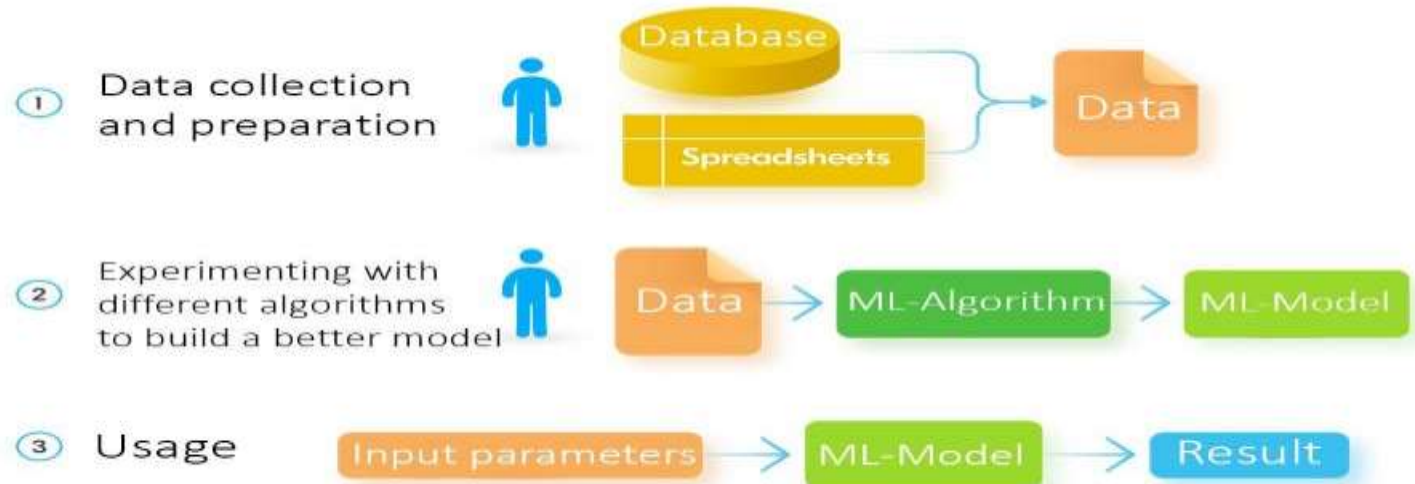


Using Algorithms for Prediction Tasks:

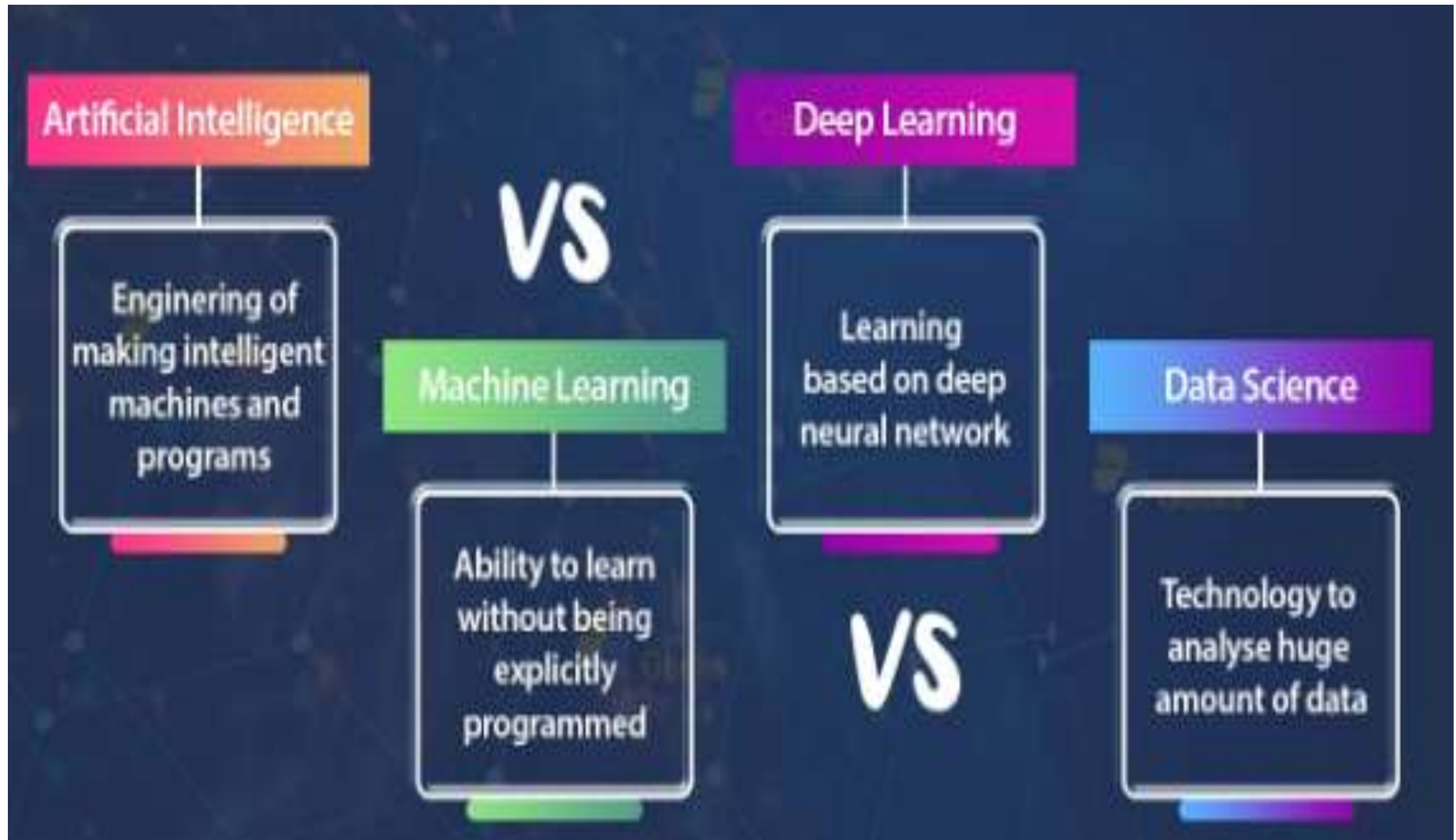
- **Prediction problems** (like exchange rate forecasting) need **many input parameters**.
- Example inputs:
 - Yesterday's exchange rate
 - Economic changes (both local and global)
- Using **only a few parameters** may create a **basic model** that is **not accurate or scalable**.
- For **better predictions**, we must include **hundreds or even thousands of relevant features**.

How a Data Engineer Builds an ML Solution:

- **Doesn't write the full algorithm from scratch.**
- First, **collects a large set of historical data** relevant to the problem.
- Loads the data into **pre-built ML algorithms** or tools (like Scikit-learn, TensorFlow, etc.).
- The ML algorithm uses this data to **train a predictive model**.
- Once trained, the model can **make predictions on new incoming data**.



ML vs AI vs Data Science.

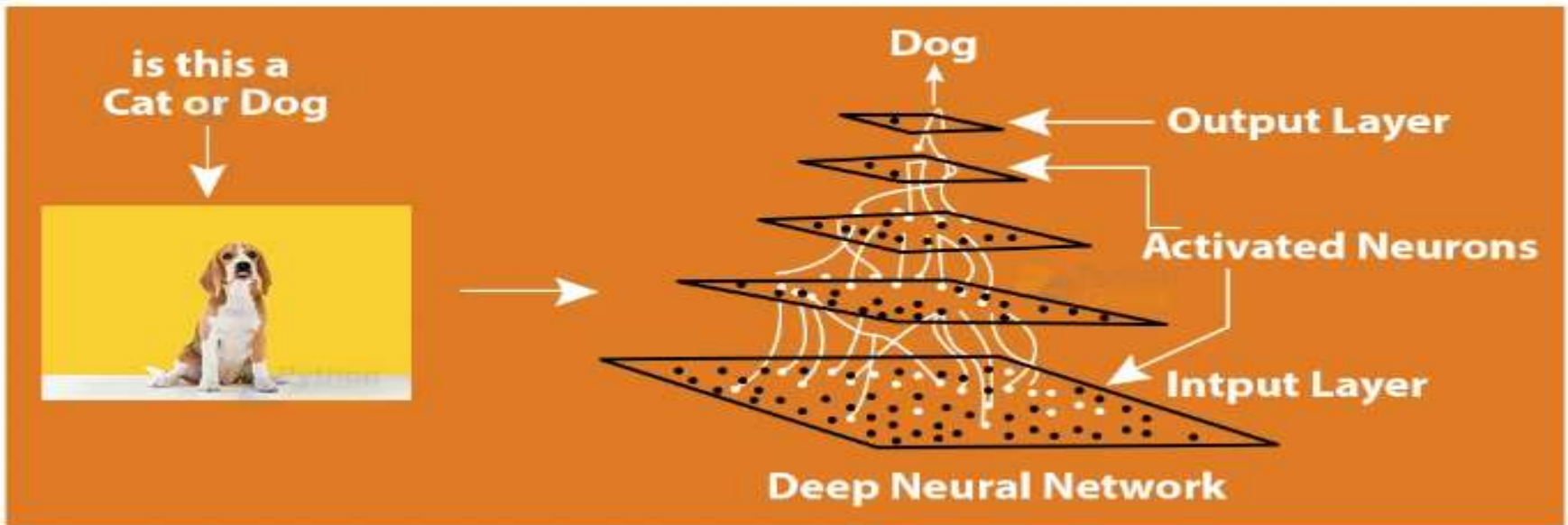


Ref- <https://pythongeeks.org/ai-vs-data-science-vs-deep-learning-vs-ml/>

Artificial Intelligence

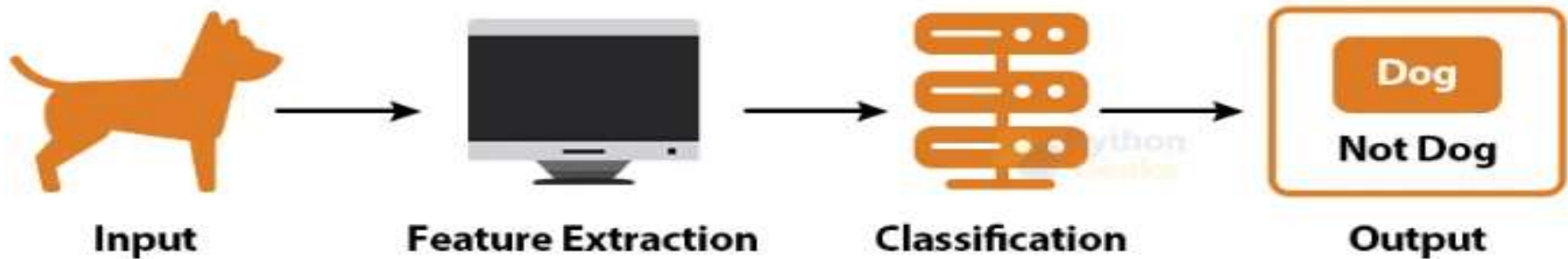
- It focuses mainly on building smart machines that are capable of performing tasks that **replicate human intelligence** without any **human interference**.
- These systems, built using **Artificial Intelligence** models, tend to **mimic human cognitive functions**, that allows decision making, and also helps to improve learning.
- **Example-** forecast financial and business outcomes and can efficiently provide solutions for businesses.

Artificial Intelligence



- We can demonstrate the working of AI in brief with the following steps:
 1. **Collect Data**
 2. **Clean and Prepare Data**
 3. **Train the Model**
 4. **Test the Data**
 5. **Improve**

Traditional Machine Learning



- Machine learning algorithms make use of **computational methods** and try to “**learn**” from the input data without the requirement of any predetermined equation.
- It comprises an application of AI that allows the systems to learn and improve significantly from the past data experience.

- The **working of the Machine Learning** models is simply put as:
 1. **Gather data** from source
 2. **Clean** and **filter** the data
 3. **Choose the effective algorithm** according to your problem
 4. **Train** the **test** model
 5. Tune in the **parameters for best performance**
 6. **Test** the models and try to **improve the efficiency**
 7. **Deploy** the **final model** having precise **outputs**

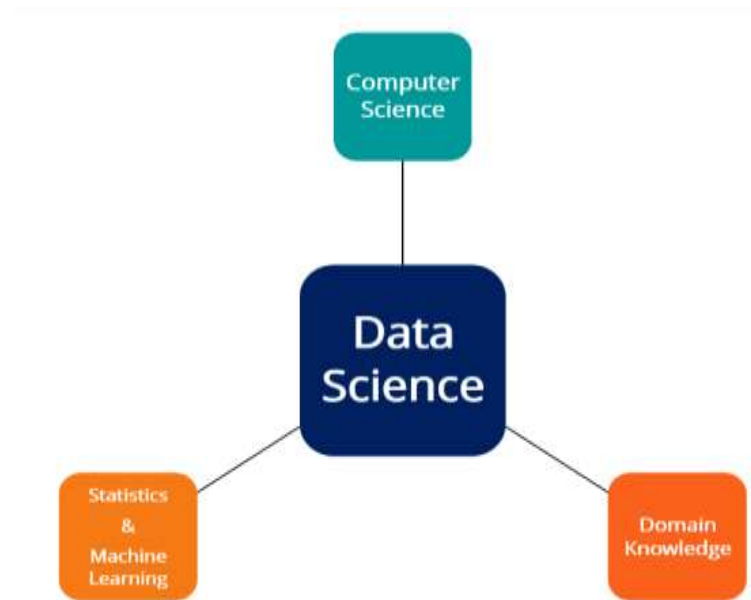


- Deep learning is again a **subfield** of the **artificial intelligence** domain.
- It makes use of a **multi-layered** structure of algorithms, more commonly known as a **neural network**.
- Deep Learning, like Machine Learning algorithms, also need **data for learning** and **solving problems** like **classification** and **prediction**.
- We can even consider **Deep Learning** as a **subdomain** of **machine learning**.

DEEP LEARNING

- In deep learning, we **don't** always need **labeled (classified)** data like in regular machine learning. Instead, deep learning models **can learn from raw or unlabeled** data by **finding useful patterns on their own**.
- They work by analyzing the data in layers, where each layer learns something new about the data. This means deep learning **can train without much human help**.
- The system keeps checking and adjusting itself, discovering new features or categories from the data automatically.

DATA SCIENCE



- The main focus of data science models is **to recognize patterns in the given input data sets.**
- It makes use of numerous **statistical techniques to analyze and extract information** like **features** and classifiers of the data from the given data sets.

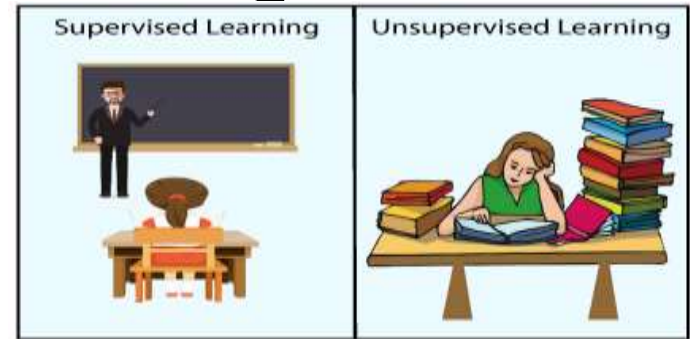
DATA SCIENCE

- Data scientists use these insights to **help companies make better decisions by learning from past sales and marketing data.**
- We can understand the **working of Data Science** model with the following points in brief:
 1. **Gather data** from sources
 2. **Filter and process** the data
 3. **Find trends** in data and get insight
 4. **Build useful data models**
 5. **Measure the performance**

Difference between AI vs Data Science vs Deep Learning vs ML

	Artificial Intelligence	Machine Learning	Deep Learning	Data Science
Technology	Uses decision trees, logic, and statistical data to mimic human intelligence	Uses statistical methods and algorithms to enable machines to learn from experience	Relies on algorithms and artificial neural networks, designed to imitate how humans think and learn	Uses math, programming, and business analysis to produce insights from huge volumes of data
Computational Requirements	Requires more computational power to make machines acquire human-level intelligence	Requires high-performance computers with good-quality GPU for the machine learning algorithms to function	Deep learning is computationally expensive given it is remarkable at modeling diverse phenomena	Requires higher RAM to find and extract patterns in data
Problem Solving Pattern	Chooses algorithms basis complexity of the problem, contributing towards cost and time saving	Divides a given problem into subsets, solves individually, and gives a combined output	Analyzes difficult problems in its hidden layers and performs automatic feature extraction	Collect, analyze, and define data. DS makes use of analytics-based approaches to extract knowledge and actionable insights
Data Dependency	Requires a lot of data to work on and obtain results	The more data a system receives, the more it learns to function better	Powered by massive amounts of data	Heavily dependent on data
Functionality	It accounts for a broader segment and focuses mainly on intelligent behavior than accuracy	Results are largely dependent on accuracy and patterns	Derives specific features from the raw input	Proposes insightful solutions from raw undefined data for effective decision making

Types of -Machine Learning



Supervised Machine Learning:

- Supervised learning is a machine learning method in which models are trained using labeled data. In supervised learning, models need to find the mapping function to map the **input variable (X)** with the **output variable (Y)**.

$$Y = f(X)$$

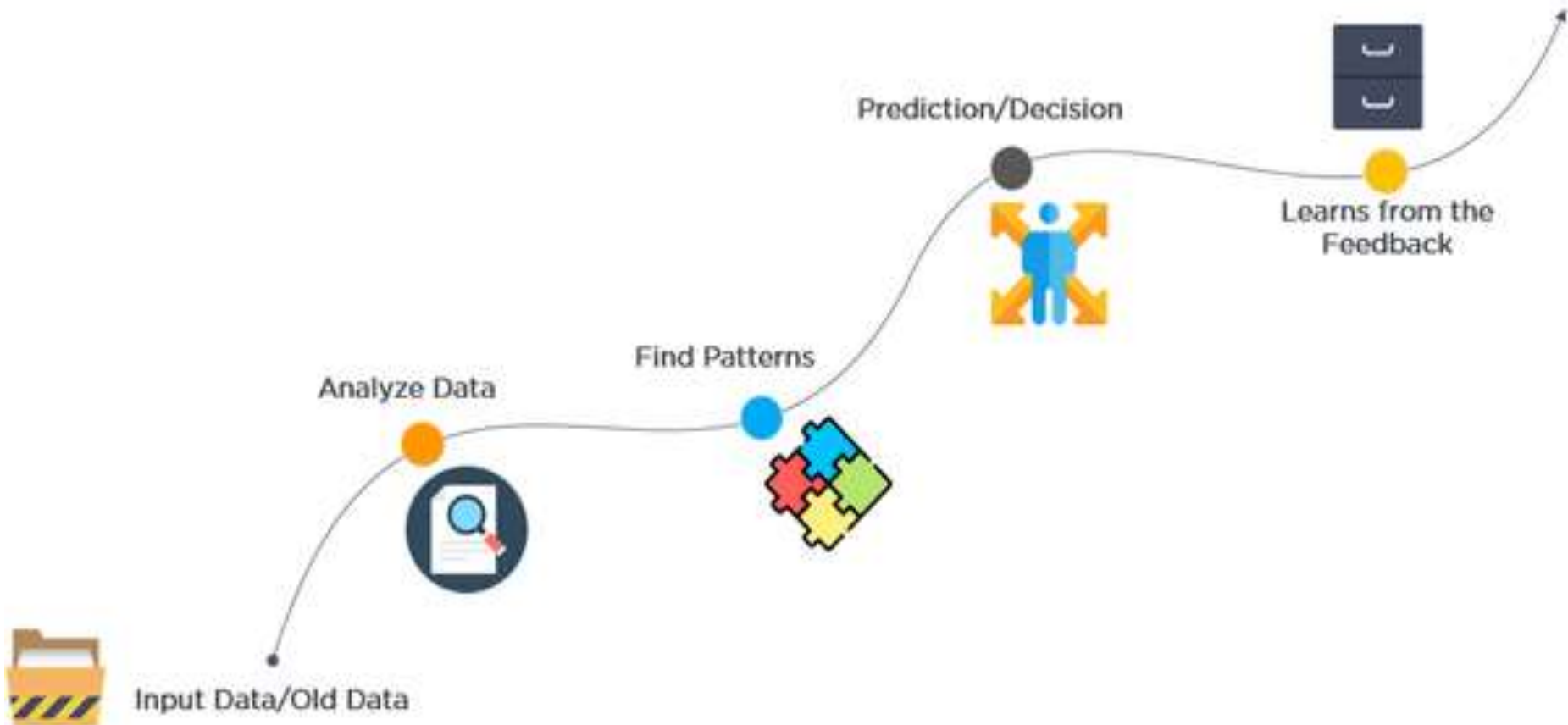
- Supervised learning needs **supervision** to train the model, which is similar to as a student learns things in the presence of a teacher.
- Supervised learning can be used for two types of problems: **Classification** and **Regression**.

Example – Fruit Image Classification with Machine Learning:

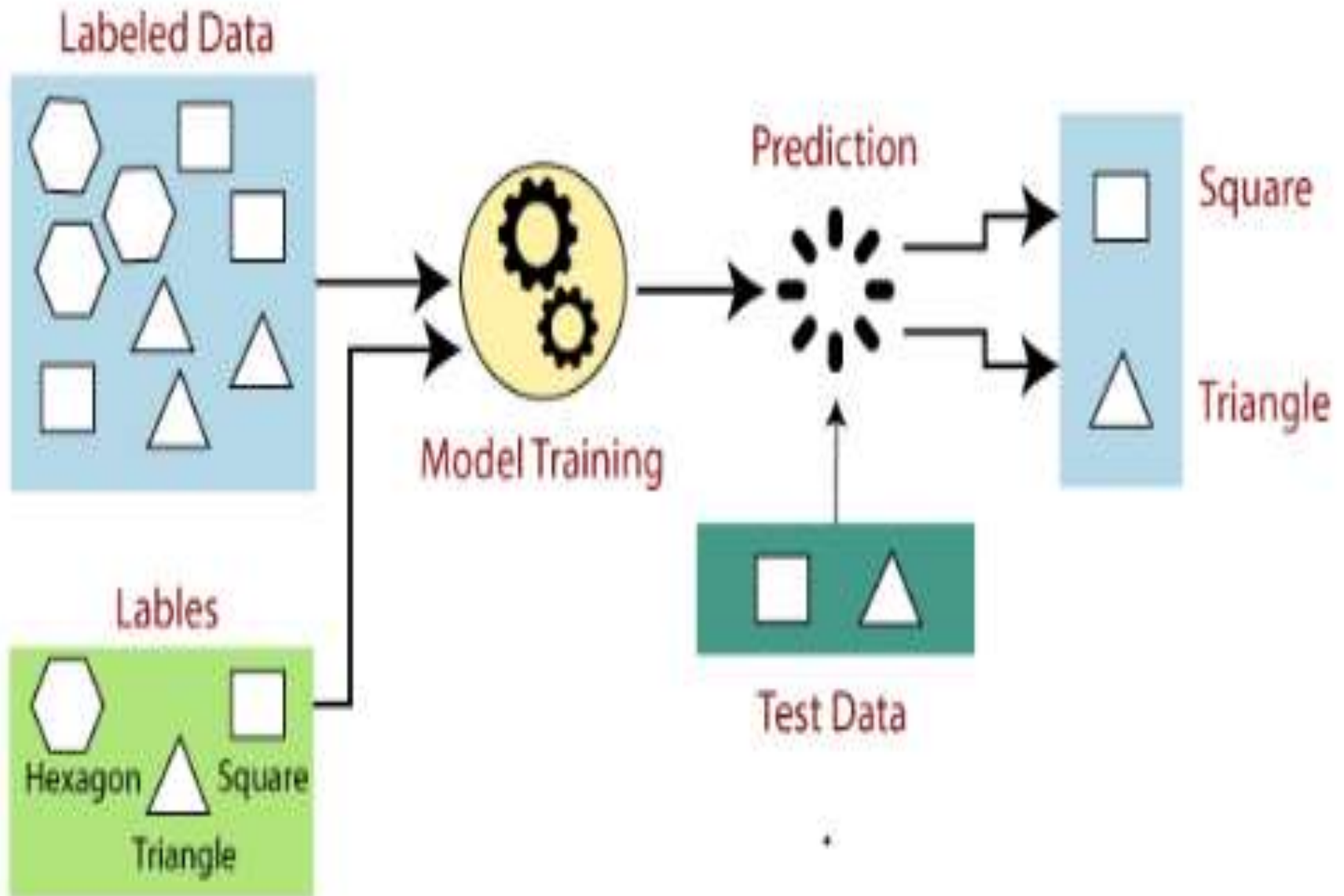
- We give the system a **dataset of fruit images** (input data).
- The system **analyzes patterns** like size, shape, and color.
- Based on these patterns, it **groups or classifies** the fruits.
- If it makes a mistake, it **learns from feedback** to improve next time.
- Over time, the system gets **better at correctly identifying** the fruit types.

Supervised Machine Learning

- In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.



How Supervised Learning Works?



How Supervised Learning Works?

- We have a dataset of different shapes (square, triangle, hexagon, etc.).
- The model is trained to recognize shapes based on their features (like number of sides).
 - Example: 4 equal sides → Square
 - 3 sides → Triangle
 - 6 equal sides → Hexagon
- After training, we test the model with new shapes.
- The model uses what it learned to predict and label the new shape correctly.

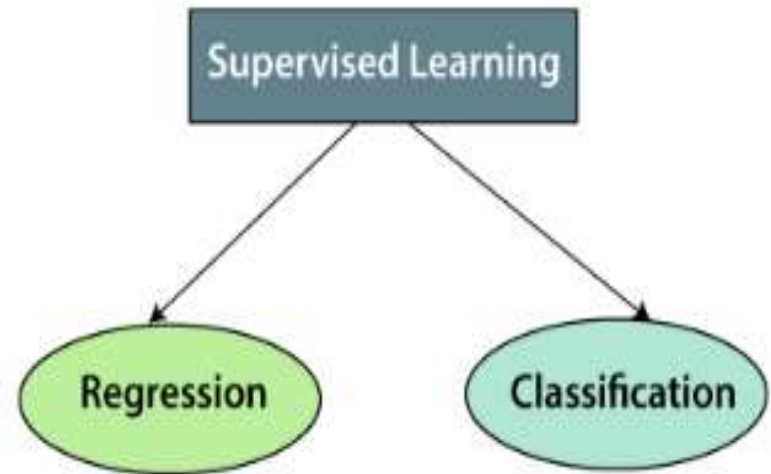
How Supervised Learning Works?

1. Choose the type of problem (**classification or regression**).
2. Collect labeled training data.
3. Split the data into training, validation, and test sets.
4. Select important input features from the data.
5. Pick a suitable algorithm (e.g., Decision Tree, SVM).
6. Train the model using the training set.
7. Use the validation set to fine-tune the model (if needed).
8. Test the model with the test set to check accuracy.

Types of supervised Machine learning Algorithms:

1. Regression

- Regression algorithms are used if there is a **relationship between the input variable and the output variable**.
- It is used for the prediction of **continuous variables**, such as **Weather forecasting, Market Trends, etc.**
- Below are some popular Regression algorithms which come under supervised learning:
 - Linear Regression
 - Regression Trees
 - Non-Linear Regression
 - Bayesian Linear Regression
 - Polynomial Regression



Supervised Machine Learning

2. Classification

- Classification algorithms are used when **the output variable is categorical**, which means there are **two classes** such as **Yes-No, Male-Female, True-false, etc.**
 - Spam Filtering,
 - Random Forest
 - Decision Trees
 - Logistic Regression
 - Support vector Machines

- **Advantages of Supervised Learning:**

- Learns from past data to make predictions.
- We know the exact labels or categories.
- Useful in real-world problems like fraud detection and spam filtering.

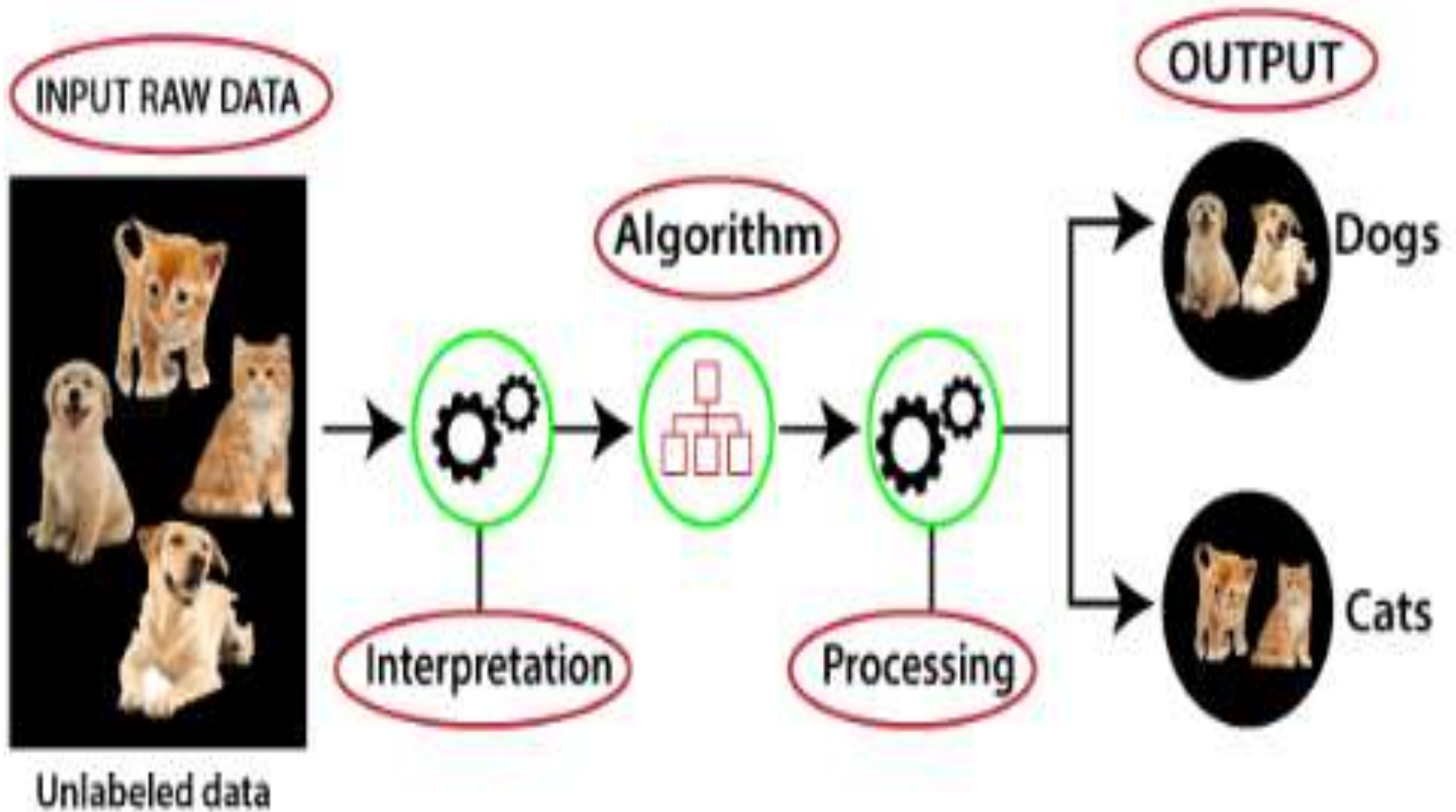
- **Disadvantages of Supervised Learning:**

- Not good for very complex tasks.
- May fail if test data is very different from training data.
- Training takes a lot of time and computing power.
- Needs detailed knowledge of data labels.

Unsupervised Learning

- We only have **input data, no output labels**.
- The model learns **without supervision or labeled data**.
- It finds hidden patterns or structures in the data on its own.
- Similar to **how humans learn by observing**.
- Main goal: group similar data and simplify complex data (e.g., clustering, compression).

Working of Unsupervised Learning



Working of Unsupervised Learning

- We use **unlabeled data** (no categories or output given).
- This data is given to the machine learning model for training.
- The model finds **hidden patterns** in the data.
- It uses algorithms like **K-Means clustering** or **Decision Trees**.
- The model groups similar data objects based on their features.

Types of Unsupervised Learning Algorithm:

Clustering:

- Groups similar objects into the same cluster.
- Objects in one group are very similar to each other.
- Objects in different groups are quite different.
- Used to find hidden patterns or natural groupings in data.

Association:

- Finds relationships between items in large datasets.
- Shows which items often appear together (e.g., Bread → Butter).
- Helps in marketing strategies like **Market Basket Analysis**.
- Commonly used in retail and recommendation systems.

Types of Unsupervised Learning Algorithm

- Below is the list of some popular unsupervised learning algorithms:
 - **K-means clustering**
 - **KNN (k-nearest neighbors)**
 - **Hierarchal clustering**
 - **Anomaly detection**
 - **Neural Networks**
 - **Principle Component Analysis**
 - **Independent Component Analysis**
 - **Apriori algorithm**
 - **Singular value decomposition**

Advantages of Unsupervised Learning:

- Handles more complex tasks than supervised learning.
- Easier to get **unlabeled data** (no need for manual labeling).

Disadvantages of Unsupervised Learning:

- Harder to train because there's **no known output** to guide learning.
- Results may be **less accurate** due to lack of labeled data.

Difference between Supervised vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.

Difference between Supervised vs Unsupervised Learning

Supervised learning can be categorized in **Classification** and **Regression** problems.

Unsupervised Learning can be classified in **Clustering** and **Associations** problems.

Supervised learning can be used for those cases where we know the input as well as corresponding outputs.

Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.

Supervised learning model produces an accurate result.

Unsupervised learning model may give less accurate result as compared to supervised learning.

Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.

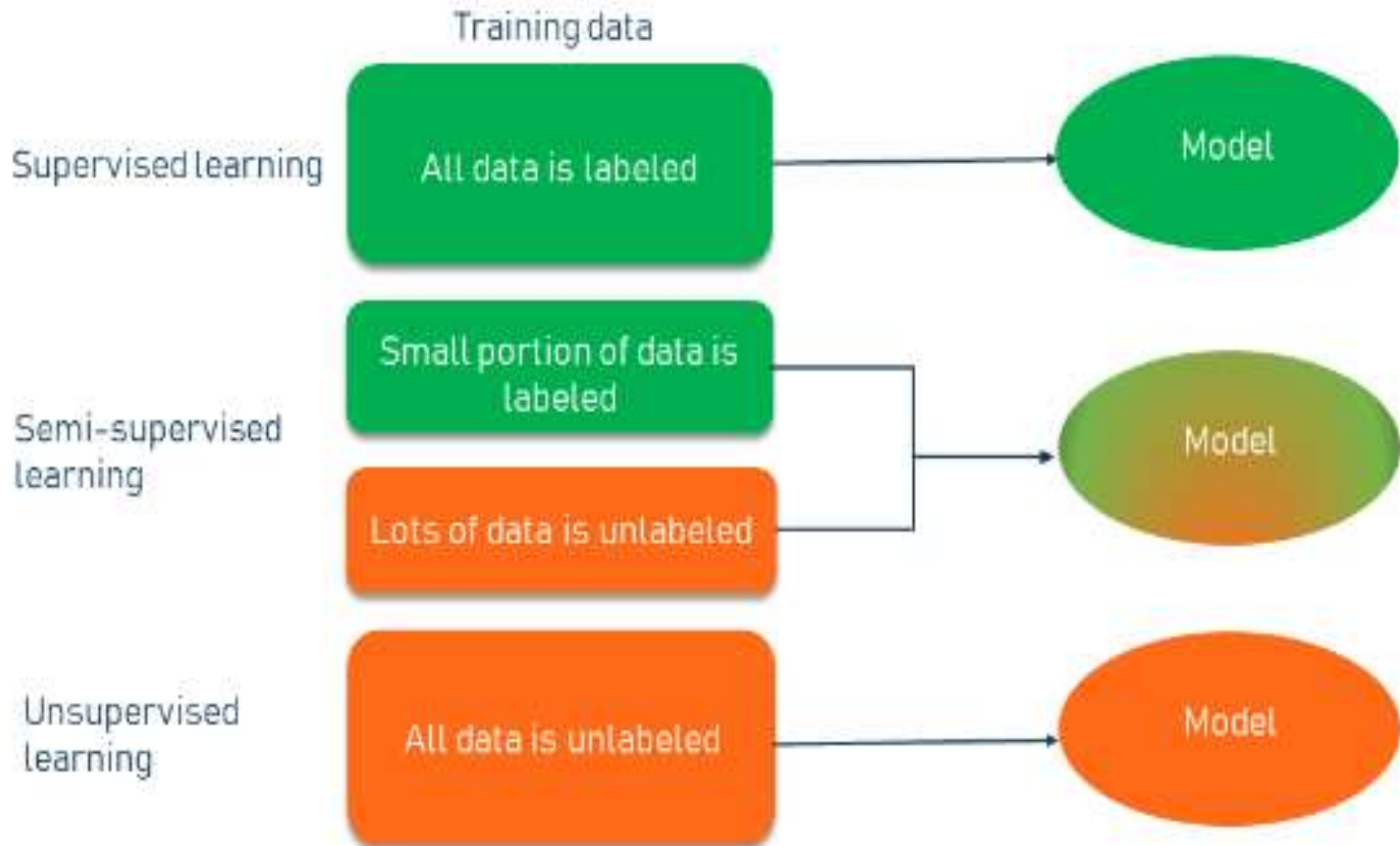
Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.

It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.

It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

Semi-Supervised Learning

SUPERVISED LEARNING vs SEMI-SUPERVISED LEARNING vs UNSUPERVISED LEARNING



Semi-supervised learning

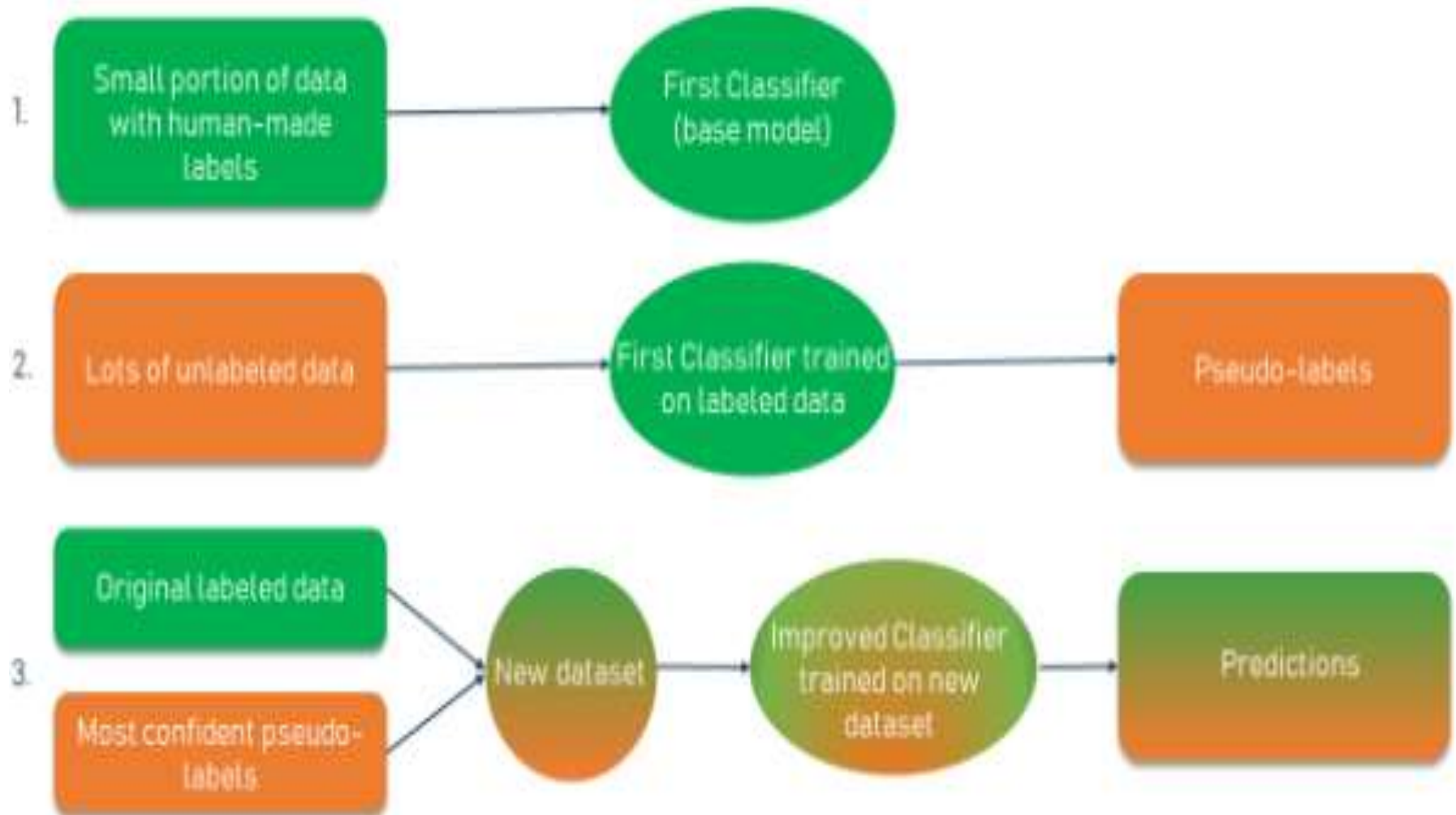
- Combines **supervised** and **unsupervised** learning methods.
- Starts with a **small amount of labeled data** and a **large amount of unlabeled data**.
- Trains the model on labeled data, then improves it using unlabeled data.
- Works for problems like **classification, regression, and clustering**.
- Saves time and cost by **reducing the need for manual labeling**.

Pseudo-Labeling in Semi-Supervised Learning:

- Start with a **small labeled dataset** (e.g., images of cats and dogs).
- Train a basic model using **supervised learning**.
- Use the model to **predict labels** for the unlabeled data — these are called **pseudo-labels**.
- Select predictions with **high confidence** (e.g., 80% sure it's a cat).
- Add these confident pseudo-labels to the labeled dataset.
- Retrain the model with this new, larger dataset.
- **Repeat the process** for several rounds (e.g., 10 times).
- The model **keeps improving** with each iteration — if the data is good.

How Semi-supervised learning work

SEMI-SUPERVISED SELF-TRAINING METHOD



Semi-supervised learning examples

- **Speech Recognition-** Facebook (now Meta) has [successfully applied](#) semi-supervised learning (namely the self-training method) to its speech recognition models and improved them.
- **Web content classification-** Many search engines, including [Google](#), apply SSL to their ranking component to better understand human language and the relevance of candidate search results to queries.
- **Text document classification-** building of a text [document classifier](#).

Reinforcement Learning Techniques

- A type of machine learning where an **agent learns by trial and error**.
- The goal is to take **actions that maximize rewards** in a given environment.
- There is **no labeled training data** — the agent learns from **experience**.
- The agent receives **rewards or penalties** based on its actions.

Example:

- An agent must find the best path to reach a reward while avoiding hurdles.
- It learns which actions lead to success through **feedback** from the environment.

Reinforcement Learning Example – Robot and Diamond:



- A **robot (agent)** wants to reach a **diamond (reward)** while avoiding **fire (penalty/hurdle)**.
- The robot **tries different paths** and learns from the outcomes.
- A **correct move** gives a **positive reward**; a **wrong move** (like stepping in fire) gives a **negative reward**.
- The robot **learns the best path** by maximizing total rewards.
- Final reward is calculated when the robot reaches the diamond with **minimum mistakes**.

Reinforcement Learning Techniques

- **Input:** An initial state where the agent starts.
- **Output:** Multiple possible outputs (actions/paths).
- **Training:**
 - The agent takes actions based on the current state.
 - It receives a **reward** or **punishment** based on the result.
- The agent **keeps learning** from its experiences.
- The **best solution** is the one that gives the **maximum total reward**.

Difference

Reinforcement learning

Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input

In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions

Example: Chess game

Supervised learning

In Supervised learning, the decision is made on the initial input or the input given at the start

In supervised learning the decisions are independent of each other so labels are given to each decision.

Example: Object recognition

Negative Reinforcement

Before	Behaviour	After
Bad smell	Have a shower	Bad smell gone

Positive Reinforcement

Before	Behaviour	After
No hot water	Turn on hot water tap (faucet)	Hot water added

Types of Reinforcement:

Positive Reinforcement:

- When a This means **rewarding good actions** so that they happen more often.
- **Example:** If a robot does something right and gets a reward (like a point), it will try to do the same thing again next time.
- This helps the robot learn which actions are good.

Advantages of Reinforcement Learning:

- Helps to Helps the robot perform better.
- It can learn good behavior over a long time.
 - **But if you give too many rewards for too many things**, the robot might get confused and not learn properly.

Types of Reinforcement:

Negative Reinforcement:

- This means **removing something bad** when the robot does the right thing.
- **Example:** If a robot goes near fire, it loses points. But if it avoids fire, it stops losing points. So, it learns to avoid fire.
- It's like teaching by avoiding punishment.

Advantages of Negative Reinforcement:

- Helps the robot avoid bad behavior.
- Teaches it to at least reach a minimum level of performance.
 - **But** it doesn't really encourage the robot to do extra good—just enough to avoid problems.

Types of Reinforcement:

- **SARSA (State-Action-Reward-State-Action):**
- An **On-policy** reinforcement learning algorithm.
- Based on the **Markov Decision Process**.
- Learns using the **action taken by the current policy**.
- Works with the tuple: (s, a, r, s', a')
 - s = current state
 - a = current action
 - r = reward
 - s' = next state
 - a' = next action



Practical applications Reinforcement Learning

- RL can be used in **robotics for industrial automation**.
- RL can be used in **machine learning and data processing**
- RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.
- RL can be used in large environments in the following situations:
A model of the environment is known, but an analytic solution is not available;
- Only a simulation model of the environment is given (the subject of simulation-based optimization)
- The only way to collect information about the environment is to interact with it.

Models of Machine learning: Geometric model

- Some broad categories of models:

1. Geometric models

- E.g. K-nearest neighbors, linear regression, support vector machine, logistic regression, ...

2. Probabilistic models

- Naïve Bayes, Gaussian process regression, conditional random field, ...

3. Logical models

- Decision tree, random forest, ... Compositional models Neural networks, logistic regression, ..

4. Ensemble models- Boosting, bagging, random forest.

5. Grading vs grouping models

Models of Machine learning: Geometric model

Goal of ML: To choose the right features (important data points) and build the right models that solve a specific task (like predicting price, classifying photos, etc.).

3 Main Types of Learning Models:

1. Logical Models

- These models use **rules or logic** to decide outcomes.
- Think of it like: *"IF condition is true, THEN this is the result."*
- **Example:** IF age > 18 AND income > ₹20k → approve loan.
- These are simple **if-then** decision rules.

2. Geometric Models

- These models use **geometry or shapes** to separate data points in space.
- The **instance space** is just the set of all possible examples (like all people).
 - A single example (instance) could be:
 $x = \{30 \text{ years old, female, medium height}\}$
- Models like **Support Vector Machine (SVM)** or **Linear Regression** use this idea.
- They try to draw lines or curves to divide the space into categories.

Models of Machine learning: Geometric model

3. Probabilistic Models

- These models use **probability** to make predictions.
- Instead of saying "yes or no", they say **how likely** something is.
- Example: There is an 85% chance this email is spam.
- Models like **Naive Bayes** or **Bayesian Networks** use this approach.

Grouping and Grading

- This refers to **Clustering** and **Classification**:
 - **Grouping**: Putting similar instances together (like people with similar age and interest).
 - **Grading**: Assigning a label or category (like Pass/Fail, Spam/Not Spam).

What Are Logical Models?

Definition:

- Logical models use **logical expressions** (True/False conditions) to divide the **instance space** (the total data).
- These expressions group data into meaningful categories for solving tasks like **classification**.

Key Point:

- After grouping, each group becomes **homogeneous** (same type/class).

Example:

If Age > 18 AND Income > ₹30,000 → Eligible = True

Logical Expressions

- A **logical expression** gives a **Boolean output**: either **True** or **False**.
- Helps segment the data:
 - If condition is True → belongs to the group
 - If condition is False → goes elsewhere

Usage:

- Especially useful in **classification problems** where you assign instances to specific categories.

Types of Logical Models

- **Tree Models**

- Use a **tree structure** of IF-THEN rules.
- Each path from root to leaf is a rule.
- **Example:** Decision Tree

- **Rule Models**

- A **collection of IF-THEN rules** (not in tree form).
- No fixed order; each rule stands on its own.
- **Example:** Rule-Based Classifiers

- Tree models = Structured rules

Rule models = Independent rule blocks

How Do Logical Models Work?

- **IF-part:** Defines the **condition** or **segment**
- **THEN-part:** Defines the **output** or **decision**

Common in supervised learning:

- Use labeled examples to **learn rules**
- Models try to explain why a data point is in a certain class

Concept Learning & Logical Models

Concept Learning = learning rules from examples

Formal Definition:

“Inferring a Boolean-valued function from training examples of its input and output.”

Steps:

1. Get **positive** and **negative** training examples
 2. Learn the rule (hypothesis) that describes the **positive class**
 3. Anything not matching the rule is **negative**
 - **Only positive cases are learned explicitly.**
- You can improve prediction using Decision Trees (e.g., ID3, C4.5) or Rule Learners if you want generalization from both positive and negative examples.
 - This is a classic Concept Learning problem used in AI & ML courses.

Define the Logical Models Problem

- You are given a set of **instances X** where each instance is described by 6 attributes:

Attribute	Possible Values
Sky	Sunny, Cloudy, Rainy
AirTemp	Warm, Cold
Humidity	Normal, High
Wind	Strong, Weak
Water	Warm, Cold
Forecast	Same, Change

- Goal: Learn a rule (Boolean function) from examples to predict $\text{EnjoySport} = 1$ or 0 .

Example – Concept Learning in Action

Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
Sunny	Warm	Normal	Strong	Warm	Same	Yes (1)
Sunny	Warm	High	Strong	Warm	Same	Yes (1)
Rainy	Cold	High	Strong	Warm	Change	No (0)
Sunny	Warm	High	Weak	Warm	Change	Yes (1)
Cloudy	Cold	Normal	Weak	Cold	Same	No (0)

Example – Concept Learning in Action

Use Find-S Algorithm

We start with the most **specific** hypothesis:

$h = [\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset]$ (nothing is known yet)

Now update h only when EnjoySport = Yes:

1. Instance 1 (Yes): $h = [\text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same}]$
2. Instance 2 (Yes): compare with $h \rightarrow$ update where different
 $h = [\text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same}] \rightarrow$ (Humidity changed)
3. Instance 4 (Yes):
 $h = [\text{Sunny}, \text{Warm}, ?, ?, \text{Warm}, ?] \rightarrow$ (Wind and Forecast changed)

Final Hypothesis (h):

If

- Sky = Sunny
- AirTemp = Warm
- Water = Warm

Then

- EnjoySport = Yes

Final Boolean Function (Predictor)

EnjoySport(x) = 1 if (Sky == Sunny AND AirTemp == Warm AND Water == Warm)
 else 0

This function classifies a day as “**EnjoySport = Yes**” only when these three conditions are met. All other cases are predicted as **No**.

Visualization as IF-THEN Rule

```
def enjoy_sport(day):  
    if day['Sky'] == 'Sunny' and day['AirTemp'] == 'Warm' and  
day['Water'] == 'Warm':  
        return 1 # Enjoy Sport = Yes  
    else:  
        return 0 # Enjoy Sport = No
```

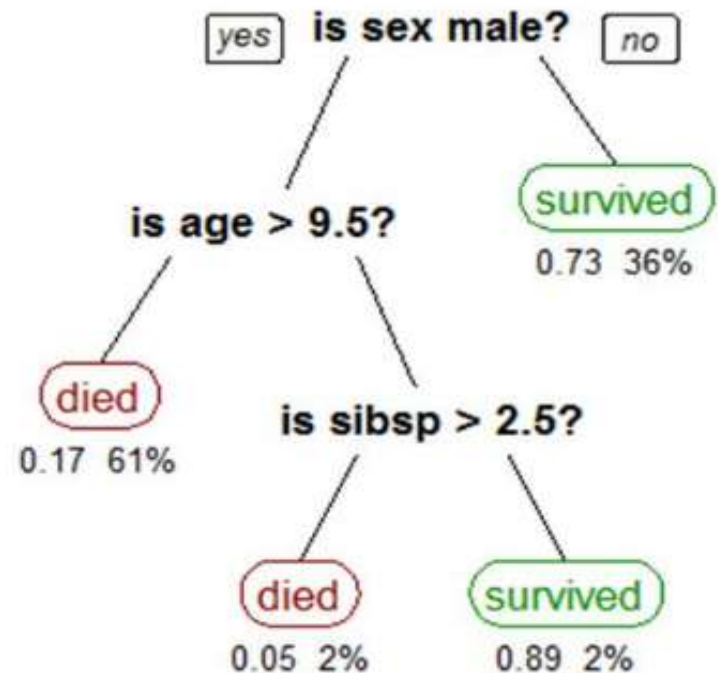
Introduction to Geometric Models

- Geometric models use the geometry of the instance space to define similarity or boundaries between instances.
- Features are considered as points in multi-dimensional space (2D, 3D, or more).
- Two types:
 - **Linear Models:** Use lines or planes to separate data.
 - **Distance-Based Models:** Use distance measures to group or classify data.

-

Real-World Example – Titanic Dataset

- Geometric interpretation in decision boundaries:
- "sibsp" (number of siblings/spouses aboard)
- Survival probabilities represented in leaf nodes
- Summary: High survival if (i) Female OR (ii) Male < 9.5 yrs and < 2.5 siblings.

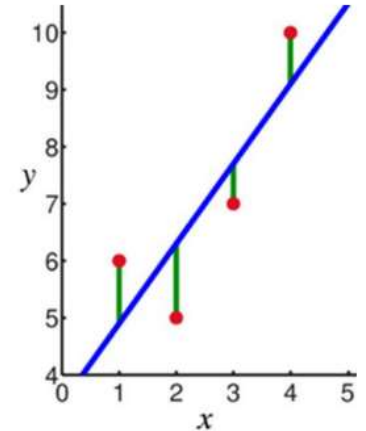


Geometric Models vs Logical Models

- **Logical Models:**
 - Use IF-THEN logical expressions
 - Partition space into segments using rules
- **Geometric Models:**
 - Use geometric concepts (line, distance)
 - Define similarity with spatial closeness

Linear Models – Concept

- Function is represented as a linear combination of inputs
- Example: $f(x) = mx + c$
 - m = slope, c = intercept
- Linear models are **parametric**: Few parameters to learn
- Example Algorithms:
 - Linear Regression, Logistic Regression, Linear Discriminant Analysis

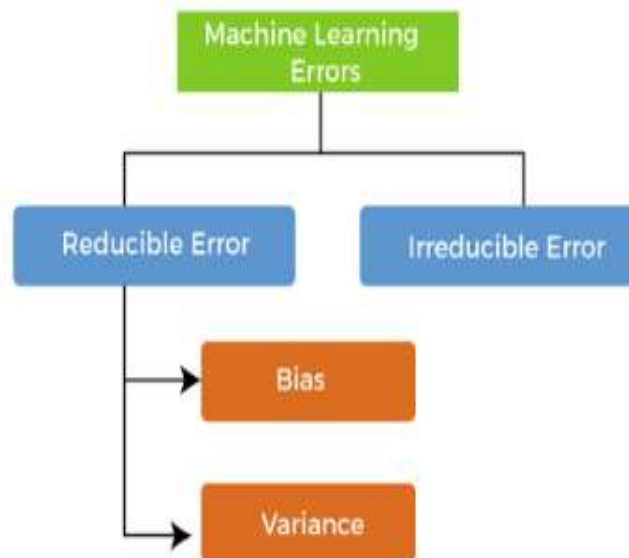


Properties

- **Stable**: Small change in data = small change in model
- **Low Variance, High Bias**
 - Less likely to overfit
 - More likely to underfit complex patterns
- Good for simple problems with linear boundaries

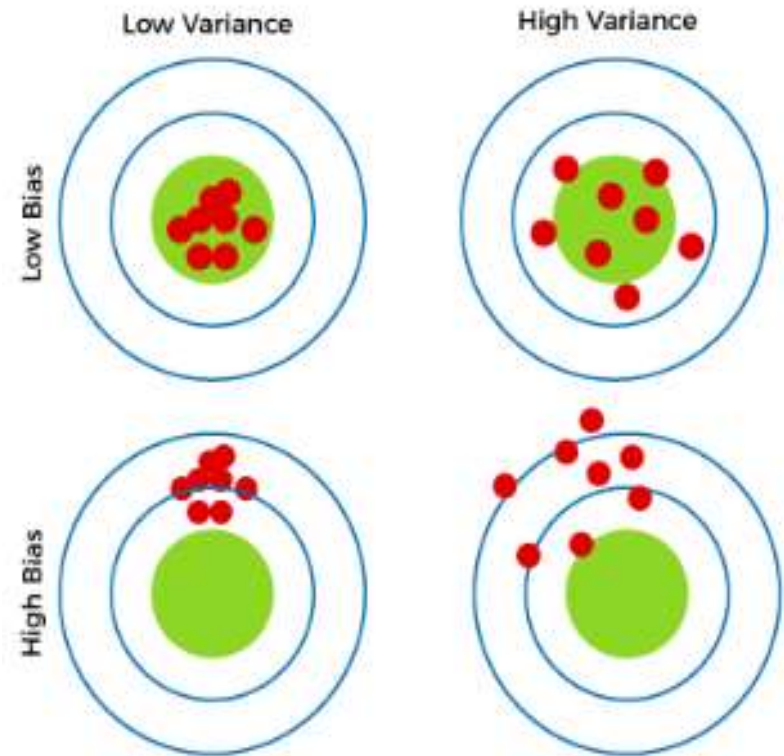
Bias in ML Models

- **Bias:** Error due to overly simplistic assumptions
- **Low Bias:** Few assumptions, flexible models (e.g., Decision Trees, SVM)
- **High Bias:** Many assumptions, rigid models (e.g., Linear Regression)
- To reduce high bias:
 - Add more features
 - Use more complex models (e.g., polynomial regression)



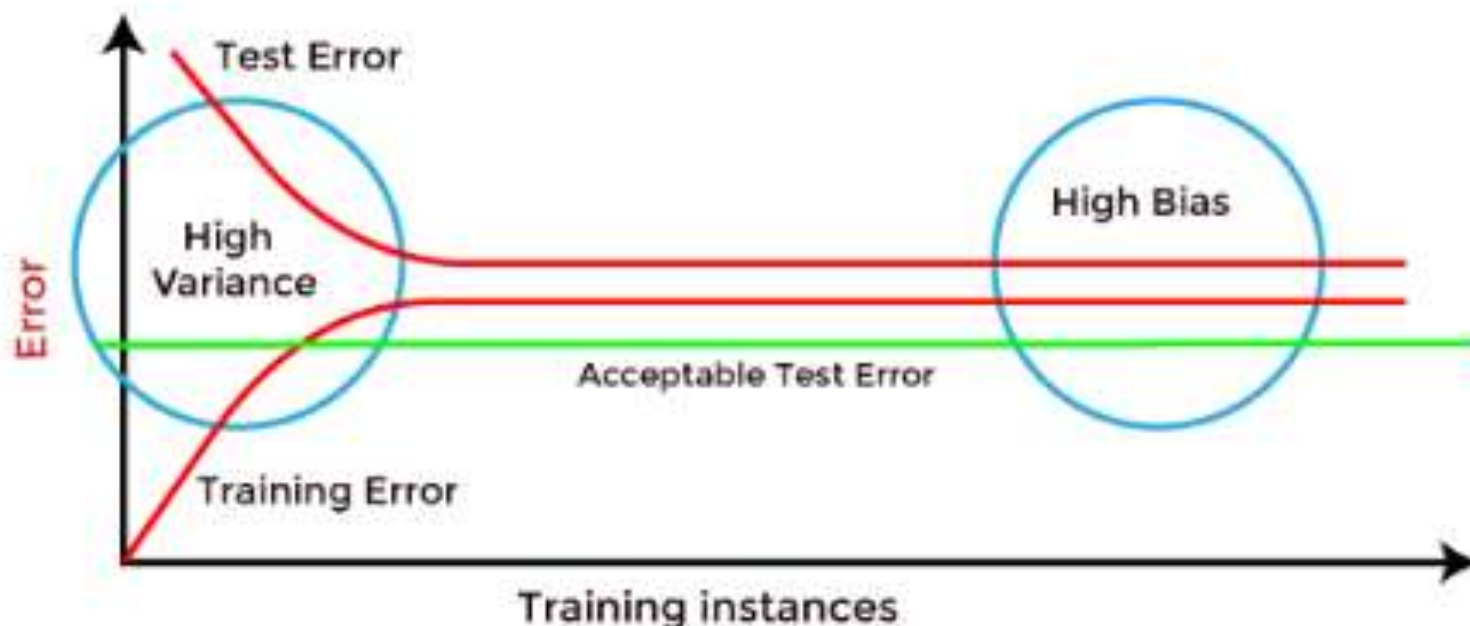
Variance in ML Models

- **Variance:** Error due to sensitivity to training data
- **Low Variance:** Predictions don't change much with new data (e.g., Linear Regression)
- **High Variance:** Overfits training data (e.g., Decision Trees, KNN)
- To reduce high variance:
 - Reduce input features
 - Use simpler models
 - Increase training data
 - Use regularization



How to identify High variance or High Bias?

High variance can be identified if the model has:



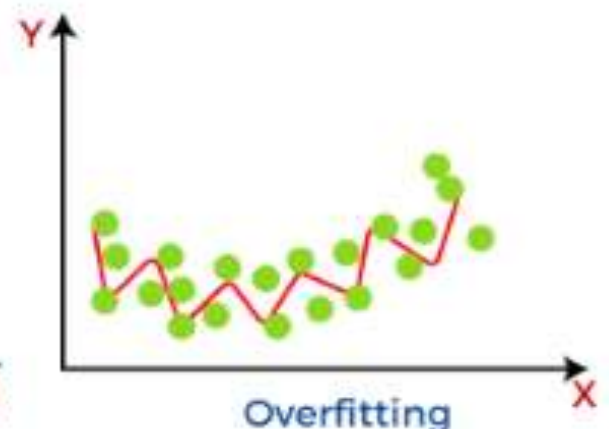
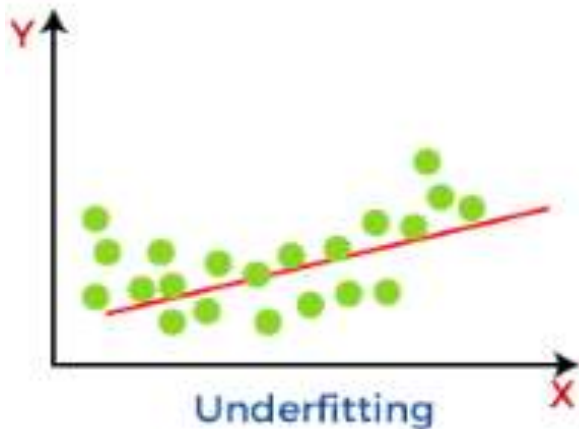
- Low training error and high test error.

High Bias can be identified if the model has:

- High training error and the test error is almost similar to training error.

Variance in ML Models

Bias	Variance	Description
Low Bias	Low Variance	Ideal model (rare)
Low Bias	High Variance	Overfitting
High Bias	Low Variance	Underfitting
High Bias	High Variance	Worst case (poor learning)



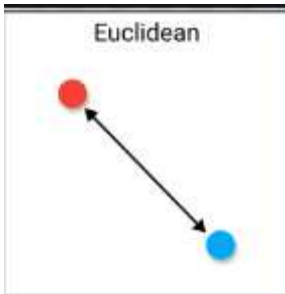
Distance-Based Models

- Use **distance** to determine similarity
- Closer points = More similar
- Not limited to physical distance (can consider context)
- Examples:
 - K-Nearest Neighbors (KNN)
 - K-Means Clustering
 - Medoids-based Clustering

Distance Metrics

- **Minkowski Distance:** Generalized distance measure
 - $p = 1$: Manhattan Distance
 - $p = 2$: Euclidean Distance
 - $p = \infty$: Chebyshev Distance
- Requirements:
 - Zero Vector (distance to self = 0)
 - Scalar Factor (distance scales linearly)
 - Triangle Inequality (shortest path is direct)

Distance Metrics



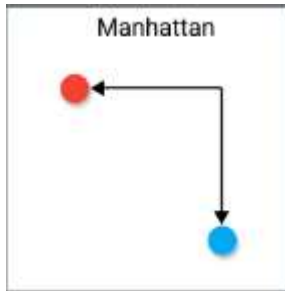
length of a segment connecting two points.

Useful for Less dimensionality

increases of your data might be skewed

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Euclidean distance



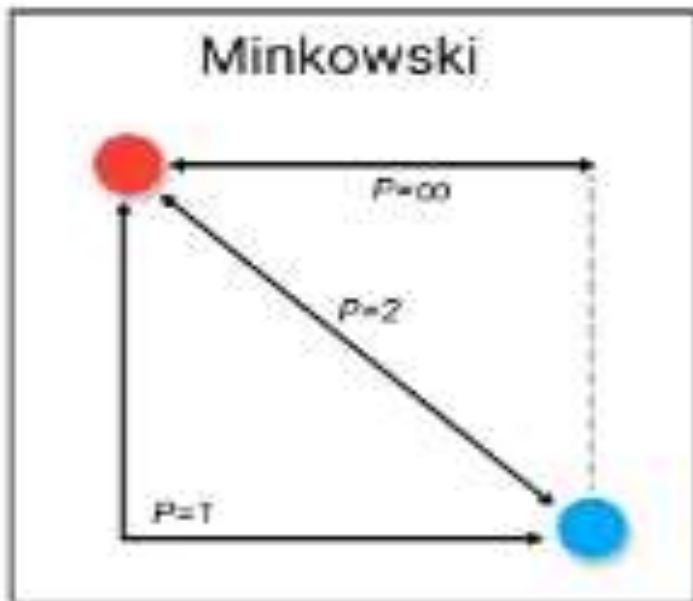
Distance measure on chess board

distance between two vectors

if they could only move right angles

$$D(x, y) = \sum_{i=1}^k |x_i - y_i|$$

Manhattan distance

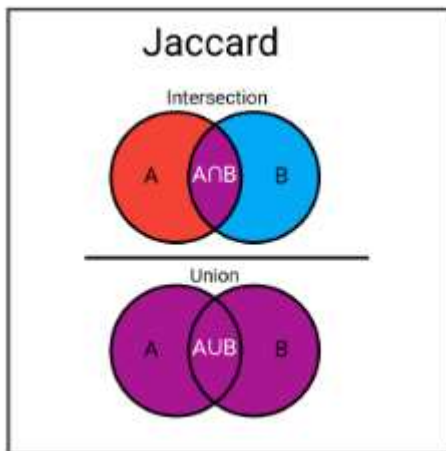


$$D(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Minkowski distance

Special Distances

- **Chebyshev Distance:**
 - Max difference along any axis
 - Used in Chessboard moves (King's distance)
- **Jaccard Index:**
 - Measures similarity between sets
 - Jaccard = (Intersection) / (Union)
 - Useful for binary & categorical data



$$D(x, y) = 1 - \frac{|x \cap y|}{|y \cup x|}$$

Examples of Distance-Based Models

- **K-Nearest Neighbors (KNN):**
 - Classify based on closest neighbors
- **K-Means Clustering:**
 - Groups points into clusters by minimizing within-cluster distances
- **Medoids:**
 - Similar to K-means but uses actual data points as cluster centers
 - Better for non-numeric data or irregular distributions

Summary of Geometric Models

- Use spatial understanding to classify or group data
- Two Types:
 - Linear (line/plane separation)
 - Distance-based (closeness-based grouping)
- Consider bias, variance, and distance metrics when choosing a model

Introduction to Probabilistic Models

Definition:

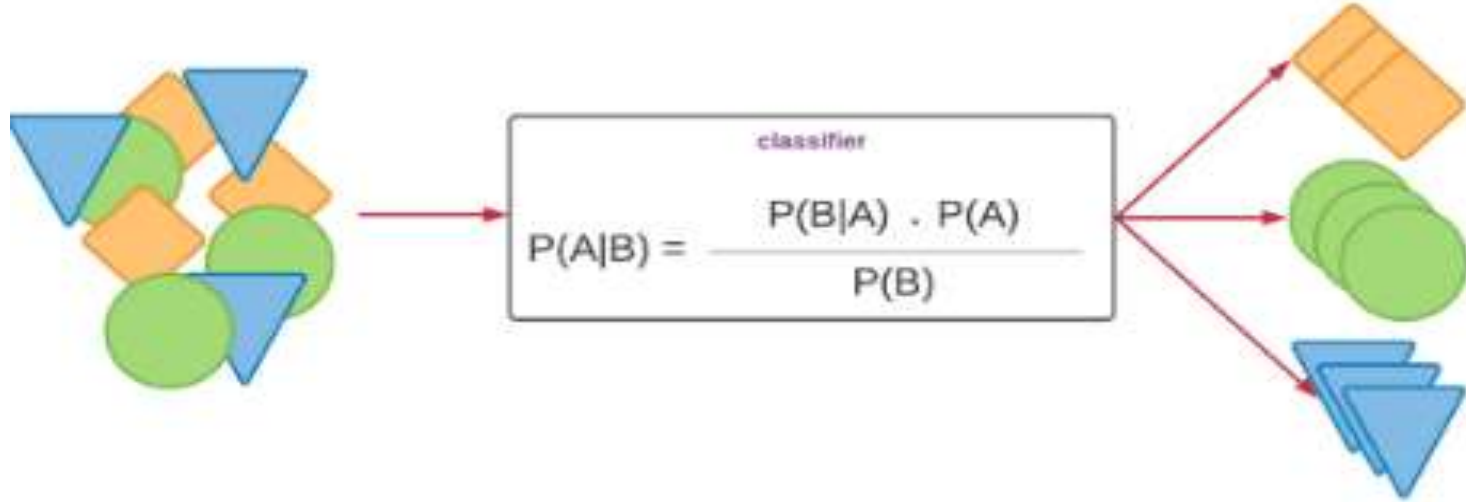
- Use probability to classify new data.
- Treat features (X) and targets (Y) as **random variables**.
- Manage **uncertainty** in prediction.

Types of Probabilistic Models:

- **Predictive Models:** Estimate $P(Y|X)$
 - Purpose: Direct prediction
 - Example: Logistic Regression
- **Generative Models:** Estimate $P(Y, X)$
 - Purpose: Learn data generation
 - Example: Naïve Bayes

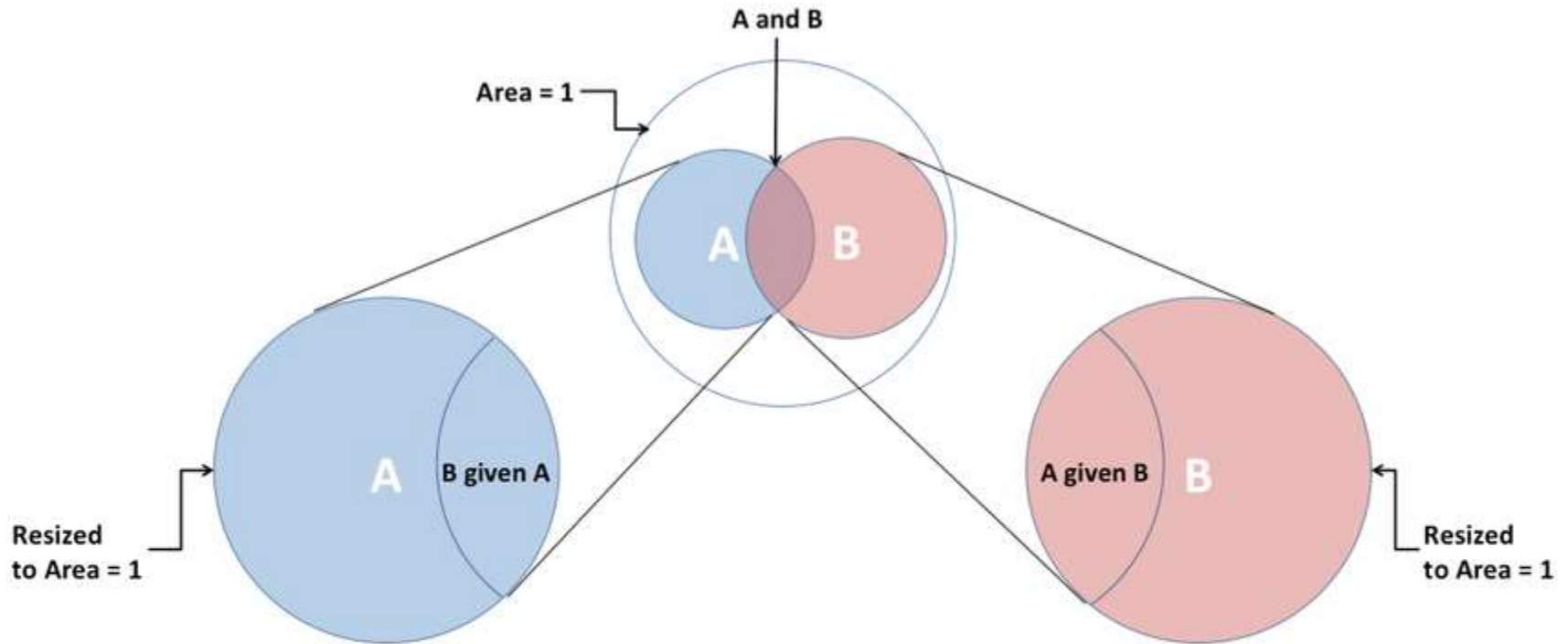
Example: Predict if a student will pass (Y) based on hours studied, attendance (X).

Bayes' Theorem Refresher



- $P(A|B)$ - **Posterior probability** of event A given event B has occurred (what we want to find)
- $P(B|A)$ - **Likelihood** of observing event B given event A is **true**
- $P(A)$ - **Prior** probability of event A occurring (initial belief)
- $P(B)$ - **Total** probability of observing event B (regardless of A)

Bayes' Theorem Refresher



$$\begin{aligned} P(A \text{ and } B) &= P(A \text{ given } B)P(B) \\ &= P(B \text{ and } A) = P(B \text{ given } A)P(A) \end{aligned}$$

$$P(A \text{ given } B)P(B) = P(B \text{ given } A)P(A)$$

$$P(A \text{ given } B) = \frac{P(B \text{ given } A)P(A)}{P(B)}$$

Naïve Bayes Classifier Algorithm

- **Supervised** learning classifier based on Bayes' theorem
- Assumes **independent features** (Naïve assumption)
- Popular for **text classification**

Examples:

- Spam detection
- Sentiment analysis
- News classification

Why “Naïve” and “Bayes”?

- **Naïve**: Assumes all features are independent
- **Bayes**: Based on Bayes' Theorem

Fruit Example:

- Color: Red
- Shape: Round
- Taste: Sweet => Recognized as Apple independently of other features

Naïve Bayes Classifier Algorithm

Now, use Bayes theorem to calculate the posterior probability.

- **Problem:** If the weather is sunny, then the Player should play or not?
- **Solution:** To solve this, first consider the below dataset:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Naïve Bayes Classifier Algorithm

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	$5/14 = 0.35$
Rainy	2	2	$4/14 = 0.29$
Sunny	2	3	$5/14 = 0.35$
All	$4/14 = 0.29$	$10/14 = 0.71$	

Naïve Bayes - Algorithm Steps

1. Convert dataset into frequency tables
2. Calculate probabilities (likelihood)
3. Apply Bayes' theorem

Calculations:

- $P(\text{Sunny}|\text{Yes}) = 3/10 = 0.3$
- $P(\text{Sunny}) = 0.35$
- $P(\text{Yes}) = 0.71$
- $P(\text{Sunny}|\text{No}) = 2/4 = 0.5$
- $P(\text{No}) = 0.29$
- **Conclusion:** Since $0.60 > 0.41 \rightarrow \text{Play}$

Calculations:

Applying Bayes'theorem:

$$P(\text{Yes}|\text{Sunny})= P(\text{Sunny}|\text{Yes})*P(\text{Yes})/P(\text{Sunny})$$

- $P(\text{Sunny}|\text{Yes})= 3/10= 0.3$
- $P(\text{Sunny})= 0.35$
- $P(\text{Yes})=0.71$
- So $P(\text{Yes}|\text{Sunny}) = 0.3*0.71/0.35= \mathbf{0.60}$

$$P(\text{No}|\text{Sunny})= P(\text{Sunny}|\text{No})*P(\text{No})/P(\text{Sunny})$$

- $P(\text{Sunny}|\text{NO})= 2/4=0.5$
- $P(\text{No})= 0.29$
- $P(\text{Sunny})= 0.35$
- So $P(\text{No}|\text{Sunny})= 0.5*0.29/0.35 = \mathbf{0.41}$

So as we can see from the above calculation that $P(\text{Yes}|\text{Sunny})>P(\text{No}|\text{Sunny})$ i.e., $\mathbf{0.60>0.41}$

Hence on a Sunny day, Player can play the game.

Naïve Bayes - Advantages

Advantages

- Fast and easy to implement
- Works for binary & multi-class classification
- Handles high-dimensional data well
- Great for text classification

Disadvantages:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications:

- Credit scoring
- Medical diagnosis
- Real-time predictions
- Spam filtering
- Sentiment analysis

Grouping and Grading Model

Model Type	Grouping Model	Grading Model
Idea	Divides input space into segments/groups	Uses one global model across the entire space
Method	Simple decision rule in each group	Learns a function over the whole data
Output Type	Local prediction per group	Global decision surface
Examples	K-Nearest Neighbors (KNN), Decision Trees	Linear Classifiers, Neural Networks

Grouping Model (Local Decision Maker):

Like a group of teachers handling different subjects - each teacher decides within their own subject group.

Grading Model (One Big Judge):

Like a single principal evaluating students using a common grading policy across all subjects.

Grouping and Grading Model

Real-World Examples

Problem	Grouping Model	Grading Model
Classify plants based on leaf shape & color	Decision Tree: If leaf is long and green → class A	Logistic Regression: Computes global probability
Image classification	KNN: Compare new image to similar ones	CNN (Neural Net): Learns from all images together

When to Use?

- **Grouping Models:**
 - When data naturally divides into groups
 - Easy to interpret (e.g., decision trees)
- **Grading Models:**
 - When you expect complex or overlapping classes
 - Better generalization (e.g., neural networks)

Parametric Machine Learning Algorithms

Definition:

- Fixed number of parameters
- Form of function + learn coefficients

Example (Linear Function): $y = b_0 + b_1x_1 + b_2x_2$

Algorithms:

- Logistic Regression
- LDA
- Perceptron
- Naïve Bayes
- Simple Neural Networks

Distribution Parameters:

- Mean, Standard Deviation

Parametric - Advantages & Limitations

Advantages:

- Simpler and faster
- Works with less data
- Easy to interpret

Limitations:

- Fixed form can be inaccurate
- May not model complex patterns well

Nonparametric Machine Learning Algorithms

Definition:

- No fixed parameters
- Fit function based on training data

Example:

- k-NN: Uses k similar data points to classify

Algorithms:

- k-Nearest Neighbors
- Decision Trees (CART, C4.5)
- Support Vector Machines

Non-Parametric - Advantages & Limitations

Advantages:

- Flexible
- Powerful for complex patterns
- Few assumptions

Limitations:

- Need more data
- Slower to train
- Can overfit easily

Parametric model	Non-parametric model
Constant number of parameters, independent of training data	Number of parameters grows with the number of training samples
Strong assumption about the training data	No assumption about the training data
Fewer training samples required	Many training samples required
Fast training, fast inference	Slow training and slow inference
Examples: Linear regression and logistic regression	Examples: Decision trees and k-nearest neighbors

Parametric Methods

Parametric Methods uses a fixed number of parameters to build the model.

Parametric analysis is to test group means.

It is applicable only for variables.

It always considers strong assumptions about data.

Parametric Methods require lesser data than Non-Parametric Methods.

Parametric methods assumed to be a normal distribution.

Non-Parametric Methods

Non-Parametric Methods use the flexible number of parameters to build the model.

A non-parametric analysis is to test medians.

It is applicable for both – Variable and Attribute.

It generally fewer assumptions about data.

Non-Parametric Methods requires much more data than Parametric Methods.

There is no assumed distribution in non-parametric methods.

Parametric data handles – Intervals data or ratio data.

But non-parametric methods handle original data.

Here when we use parametric methods then the result or outputs generated can be easily affected by outliers.

When we use non-parametric methods then the result or outputs generated cannot be seriously affected by outliers.

Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is different.

Similarly, Non-Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is the same.

Parametric methods have more statistical power than Non-Parametric methods.

Non-parametric methods have less statistical power than Parametric methods.

As far as the computation is considered these methods are computationally faster than the Non-Parametric methods.

As far as the computation is considered these methods are computationally slower than the Parametric methods.

Examples: Logistic Regression, Naïve Bayes Model, etc.

Examples: KNN, Decision Tree Model, etc.

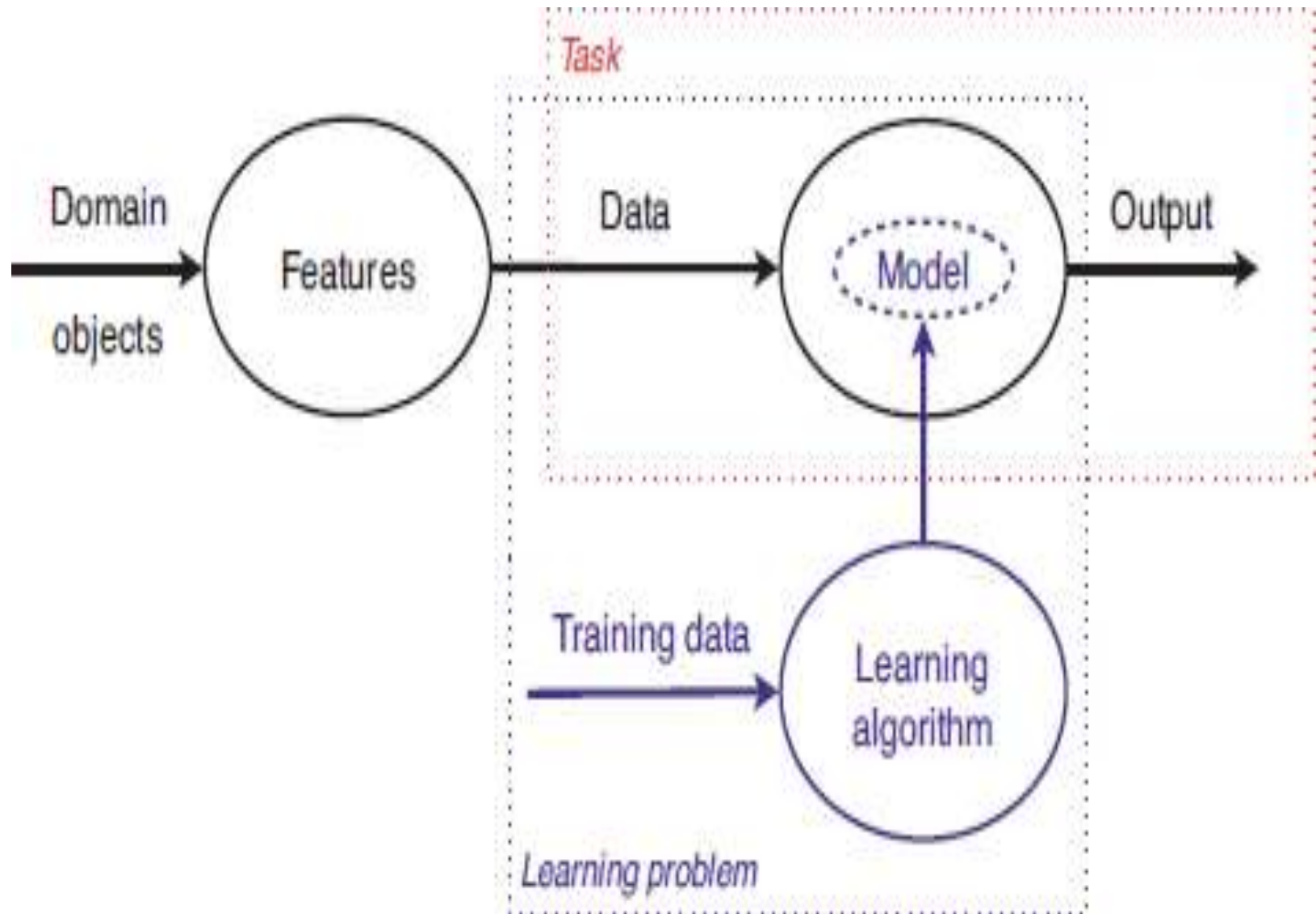
Important Elements in Machine Learning

Data formats

- In a supervised learning problem, there will always be a dataset, defined as a **finite set of real vectors with m features each**:

$$X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\} \text{ where } \bar{x}_i \in \mathbb{R}^m$$

The elements of Machine Learning



ML- Data Formats

- **Definition:** How data is structured for ML models.
- **Depends on:** Model type and problem domain
- **Used for:** Training and inference

Common Types:

- Tabular Data
- Image Data
- Text Data
- Time Series
- Audio
- Video
- Graphs

Tabular Data Formats

- Structured rows and columns
- Stored as: CSV, TSV, Excel
- Each row = 1 data point
- Each column = feature
- **Example:**

Rooms	Location	Size (sqft)	Price
3	Pune	1200	65L

Image Data Formats

- Represented as pixel matrices (2D for grayscale, 3D for RGB)
- Stored as: JPEG, PNG

Example:

- Classifying images of cats vs dogs
- Each image = 1 data point with pixel features

Text Data Formats

Represented as strings, tokens, or vectors

Techniques: Bag of Words, TF-IDF, Word2Vec

Example:

Sentiment analysis on product reviews:

Text: "Amazing product!"

Converted to numerical format for processing

Time Series Data Formats

- Data indexed by time
- Each entry includes timestamp + feature(s)

Date	Stock Price
01-Jan-24	₹1250
02-Jan-24	₹1275

Audio, Video, and Graph Formats

- **Audio:** Waveform or spectrograms
Example: Speech-to-text systems
- **Video:** Sequence of image frames over time
Example: Activity recognition (walking, jumping)
- **Graph:** Nodes and edges
Example: Social networks (users = nodes, friendships = edges)

Feature Vector & Feature Space

- **Feature Vector:** List of feature values for a data point
- **Feature Space:** m-dimensional space of all vectors

Example:

- Data point: [Red, Round, Sweet] → Apple
- Feature Vector: [color, shape, taste]

Learnability in Machine Learning

Definition:

- Ability of a model to learn patterns and generalize to new data

Goal: Build models that perform well on **unseen** (test) data

Key Concept: Generalization

Example A: 2, 4, 6, 8

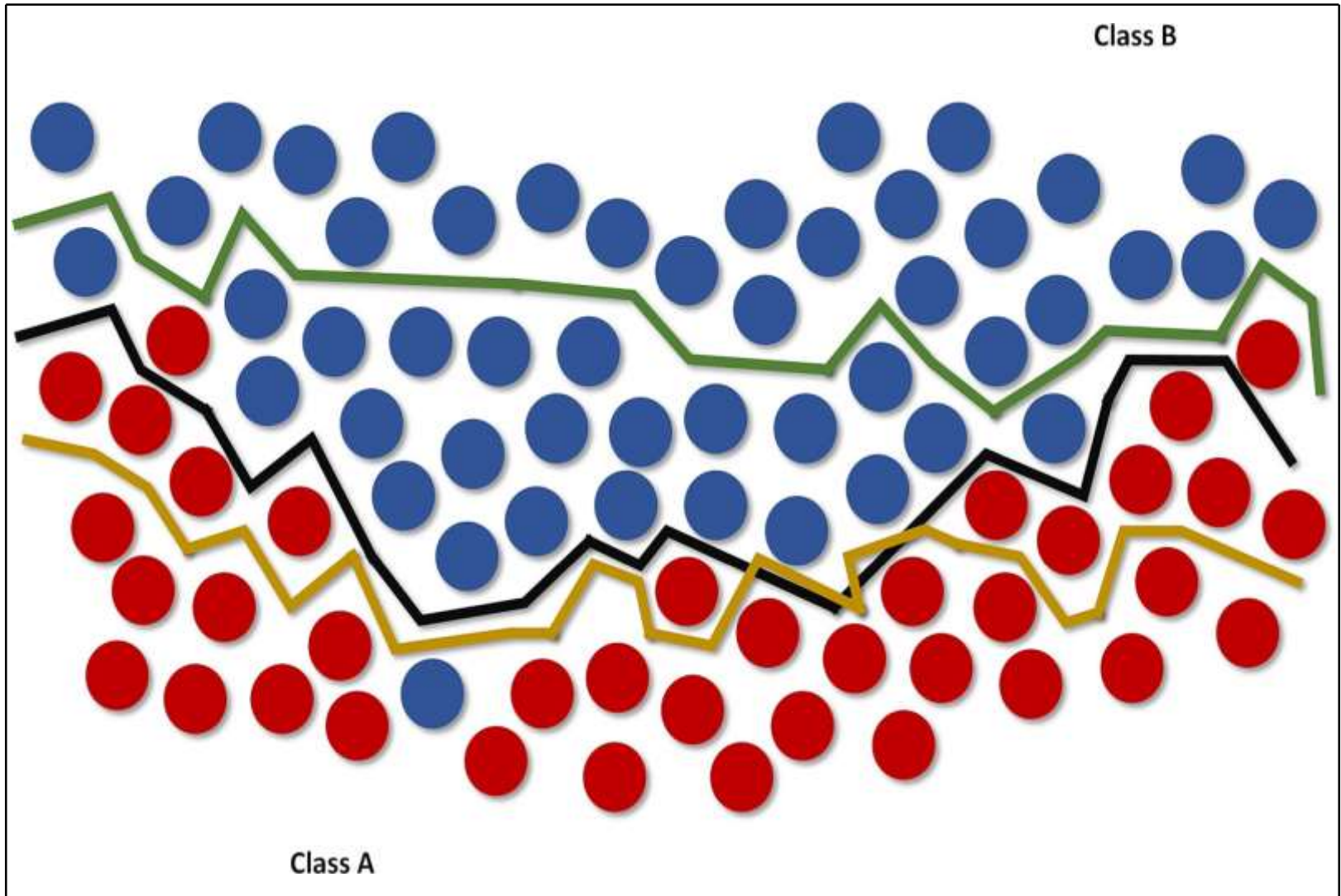
- Easy to learn (pattern is +2)

Example B: 6, 5, 1, 3

- No obvious rule → hard to learn

Conclusion: Learnable patterns make model training effective

Learnability



- Classifying Red (Class A) and Blue (Class B)
- Three hypotheses are shown: the first one (the middle line starting from left) misclassifies one sample,
- while the lower and upper ones misclassify **13** and **23** samples respectively:
- The first hypothesis is optimal and should be selected; however, it's important to understand an essential concept which can determine a potential overfitting
- Three hypotheses shown:
 - One misclassifies 1 point (best)
 - One misclassifies 13 points
 - One misclassifies 23 points
- **Point:** Even the best model should avoid overfitting

Statistical learning approaches

- The statistics for machine learning come along in the **first phase of machine learning algorithm.**
- It helps us **to deal with the data**, as it is the foundation of implementing statistical concepts and ultimately interpreting conclusions from it.
- **It is a mathematics field used for collecting, organizing, and analyzing data.**
- its subcategories are:
 - Descriptive statistics
 - Inferential statistics

Statistical learning approaches

- In the context of machine learning, descriptive statistics help to **gain insights into the characteristics of the data before building and training a model.**
- Used to **summarize and understand** data

Descriptive Statistics

Common Measures:

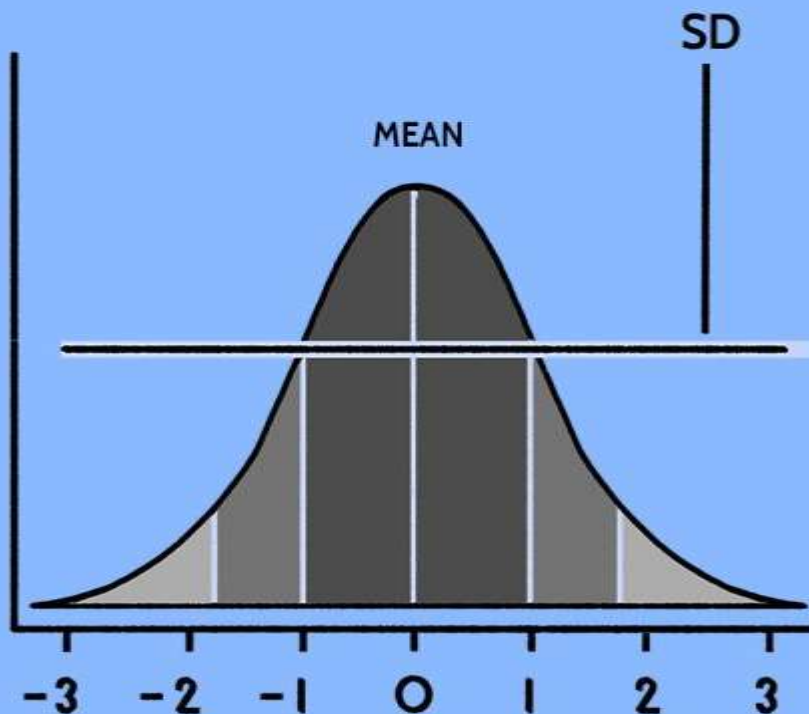
- Mean, Median, Mode
- Min, Max
- Standard Deviation (σ)
 - Low $\sigma \rightarrow$ Data clustered near mean
 - High $\sigma \rightarrow$ Data spread out
- Percentages

Application: Feature analysis before modeling

Standard Deviation

[ˈstan-dərd dē-vē-'ā-shən]

A statistic that measures the dispersion of a dataset relative to its mean and is calculated as the square root of the variance.



Inferential statistics

- In machine learning, inferential statistics are often employed to **make predictions on new, unseen data using a model trained on a representative sample.**
- Used to draw conclusions about a **population from a sample**
- The process of inferential statistics involves the following steps
 - Random sampling
 - Model Building
 - Model Evaluation

Purpose in ML: Generalize model to unseen data

How Statistics is Used in Machine Learning

