ANNAMACHARYA UNIVERSITY

• RAJAMPET •

Department of AI&ML

MACHINE LEARNING

III B.Tech | Sem V | AI&ML | R24 Regulations

LEARN • MODEL • PREDICT • IMPROVE

Author

Mr. M. Vayunandan Kumar

Assistant Professor,
Department of AI&ML,
Annamacharya University,
Rajampet

Title of the Course: Machine Learning
Category: PC
Year: III
Semester: I Semester
Course Code: 24ACSM51T
Branch/es: CSE, AI&ML, CSE(AIML) &CSE(IOTCSBCT)

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The course is introduced for students

1. To introduce the fundamental concepts and types of machine learning
2. To develop a deep understanding of supervised and unsupervised learning algorithms.
3. To understand mathematical foundations of learning models and algorithms.
4. To evaluate model performance using appropriate statistical and analytical tools.
5. To apply machine learning techniques to solve real-world problems using tools such as Scikit learn.

Course Outcomes:
After completion of the course, students will be able to:

1. Understand and distinguish among different types of learning methods.
2. Apply supervised and unsupervised learning algorithms to datasets
3. Analyze model performance using cross-validation and error metrics
4. Build, test, and improve machine learning models for classification and prediction.
5. Use Python-based libraries (e.g., Scikit-learn) to implement ML algorithms.

Unit 1 Introduction to Machine Learning and Linear Models 9

Definition and Scope of Machine Learning, Applications and Types of Learning: Supervised, Unsupervised, Reinforcement, Linear Regression: Least Squares, Cost Function, Gradient Descent, Polynomial Regression and Overfitting, Evaluation Metrics: RMSE, MAE, R^2 Score, Bias-Variance Trade off.

Unit 2 Classification Algorithms 9

Classification Overview and Decision Boundaries, Logistic Regression: Sigmoid Function and Cost, K Nearest Neighbors (KNN), Naïve Bayes Classifier, Decision Trees and Random Forests, Model Evaluation: Confusion Matrix, Precision, Recall, F1-Score.

Unit 3 Support Vector Machines and Ensemble Methods 9

Support Vector Machines: Concepts, Kernels, Hyperplane and Margin Concepts, Kernel Tricks: RBF and Polynomial, Ensemble Learning: Bagging, Boosting, and Voting, Gradient Boosting, AdaBoost, and XGBoost, Model Tuning and Hyperparameter Optimization.

Unit 4 Unsupervised Learning Techniques 9

Clustering Overview: Applications, K-Means Clustering Algorithm, Hierarchical Clustering, DBSCAN and Density-Based Methods, Principal Component Analysis (PCA) for Dimensionality Reduction, Silhouette Score, Davies-Bouldin Index for Cluster Validation.

Unit 5 Advanced Topics and Applications 9

Reinforcement Learning Basics and Markov Decision Processes, Introduction to Neural Networks and Deep Learning, Cross-Validation Techniques: k-Fold, Leave-One-Out, Feature Engineering and Feature Selection, Deployment of ML Models (Flask, Streamlit, etc.), Case Studies: Medical Diagnosis, Spam Detection, Credit Scoring.

Textbook:

1.Tom Mitchell, Machine Learning, McGraw-Hill Education.

Reference Books:

1.Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press.
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer.

Online Learning Resources:

1. Coursera – Machine Learning by Andrew Ng (Stanford University)
2. Scikit-learn Documentation
3. Kaggle Learn – Machine Learning
4. Google's Machine Learning Crash Course
5. YouTube – StatQuest with Josh Starmer

UNIT I:

Introduction to Machine Learning and Linear Models Definition and Scope of Machine Learning, Applications and Types of Learning: Supervised, Unsupervised, Reinforcement, Linear Regression: Least Squares, Cost Function, Gradient Descent, Polynomial Regression and Overfitting, Evaluation Metrics: RMSE, MAE, R^2 Score, Bias-Variance Trade off.
Concept Learning :FIND-S ALGORITHM ,Candidate Elimination Algorithm

Machine Learning : It is a Subset Of Arttificial Intelligence

Definition :

Father of the machine learning is "Arthur Samuel"
Father of the Deep learning is " Geoffrey Hinton"

- Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.
- Machine learning constructs or uses the algorithms that learn from historical data (Training Data). The more we will provide the information, the higher will be the performance.

Evolution of Machine Learning:

- Machine Learning technology has been in existence since **1952**.
- **Machine Learning** is said to learn from **experience (E)** with respect to a class of **tasks (T)** and **performance measure (P)**.
- The **Father of Machine Learning** is **Arthur Samuel**.
- In **1950**, **Alan Turing** proposed the **Turing Test** to test the capabilities of an intelligent agent (machine).
- In **1952**, **Arthur Samuel** created a program to play the game of **Checkers** and introduced the term **Machine Learning** through his research.
- In **1957**, **Frank Rosenblatt** proposed the **Perceptron**, an early neural network model inspired by the human brain and nervous system.
- In **1967**, the **Nearest Neighbor (K-Nearest Neighbor)** algorithm was introduced.
- In **1979**, students at **Stanford University, California**, developed the **Stanford Cart**, an autonomous vehicle that could navigate and avoid obstacles on its own.
- **In the year 1982**, **Recurrent Neural Network (RNN)** was introduced.
- **In the year 1989**, **Reinforcement Learning** was introduced.
- **In the year 1995**, **Support Vector Machine (SVM)** and the **Random Forest algorithm** were introduced.
- **In the year 1997**, **IBM's Deep Blue** defeated the **World Chess Champion Garry Kasparov**. This was a major milestone in Artificial Intelligence and Machine Learning.
- **In the year 2002**, **Torch**, a software library for Machine Learning, was first released.
- **In the year 2006**, **IBM** launched a **Machine Learning Competition** to encourage research and innovation in Machine Learning.
- **In the year 2016**, **Google DeepMind** developed the **AlphaGo** algorithm. AlphaGo defeated the world champion in the board game **Go**, marking a significant achievement in Artificial Intelligence.

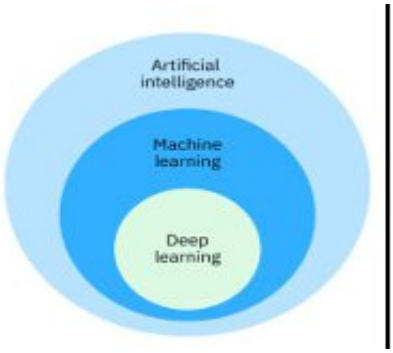


Fig : Machine learning

Popular Tools in Machine Learning:

Introduction:

Machine Learning (ML) requires different software tools, libraries, and platforms to build, train, test, and deploy intelligent models. These tools make it easier for developers and data scientists to analyze data and create AI applications.

1.)Tensor flow:

TensorFlow is an **open-source Machine Learning and Deep Learning library** developed by **Google**. It is one of the most widely used ML frameworks.

Features

- Free and open source.
- Supports Machine Learning and Deep Learning.
- Can run on CPU, GPU, and TPU.
- Used for image recognition, speech recognition, and NLP.
- Provides pre-built models and APIs.

Advantages

- High performance.
- Large community support.
- Easy deployment on mobile and cloud.
- Suitable for large-scale applications.

Applications

- Face recognition
- Language translation
- Self-driving cars
- Medical diagnosis
- Recommendation systems

Example: Google Translate and Google Photos use TensorFlow.

2. PyTorch:

PyTorch is an **open-source Machine Learning framework** developed by **Facebook AI Research (FAIR)**. It is based on the Torch library.

Features

- Easy to learn and use.
- Supports dynamic computation graphs.
- Excellent for research.
- Works well with Python.
- Supports GPU acceleration.

Advantages

- Simple syntax.
- Easy debugging.

- Faster model development.
- Widely used in AI research.

Applications

- Image classification
- Object detection
- Natural Language Processing (NLP)
- Chatbots
- Robotics

Example: Many AI research projects and universities use PyTorch.

3. Apache Mahout

Apache Mahout is an open-source framework used to build scalable Machine Learning applications on large datasets.

Features

- Supports clustering, classification, and recommendation.
- Works with Apache Hadoop and Spark.
- Suitable for Big Data.

Applications

- Recommendation systems
- Customer analysis
- Data mining

EXAMPLE:

Amazon recommends products like "Customers who bought this also bought..." using recommendation algorithms.

4. KNIME

KNIME (Konstanz Information Miner) is an open-source data analytics platform that allows users to create Machine Learning workflows using a graphical interface.

Features

- No coding required.
- Drag-and-drop interface.
- Easy data visualization.
- Supports Python and R integration.

Applications

- Data analysis
- Predictive analytics
- Business intelligence

EXAMPLE:

A **bank** uses KNIME to analyze customer data and identify customers who may apply for loans.

5. WEKA

WEKA (Waikato Environment for Knowledge Analysis) is an open-source software developed by the University of Waikato, New Zealand.

Features

- Easy graphical interface.
- Supports many Machine Learning algorithms.
- Good for beginners.
- Useful for educational purposes.

Applications

- Classification
- Clustering
- Data preprocessing
- Association rule mining

EXAMPLE: A **college student** uses WEKA to predict whether a student will pass or fail based on attendance and marks.

6. Google Cloud ML Engine

It is Google's cloud platform used to build, train, and deploy Machine Learning models online.

Features

- Cloud-based service.
- Supports TensorFlow.
- Automatic scaling.
- Fast model deployment.

Applications

- Fraud detection
- Customer analytics
- Image recognition

EXAMPLE:

An **online shopping website** uses Google Cloud ML to predict which products a customer is likely to buy.

7. Amazon Machine Learning (Amazon ML)

Amazon Machine Learning is a cloud service provided by Amazon Web Services (AWS) that helps users build predictive models without requiring extensive coding.

Features

- Easy model creation.
- Cloud storage integration.
- High scalability.
- Secure environment.

Applications

- Sales forecasting
- Customer prediction
- Fraud detection

EXAMPLE: An **e-commerce company** predicts future product sales to manage inventory.

8. Jupyter Notebook

Jupyter Notebook is an interactive web-based application used for writing and running Python code.

Features

- Runs code step by step.
- Displays graphs and charts.
- Supports Python, R, and Julia.
- Easy documentation.

Applications

- Machine Learning experiments
- Data analysis
- Education
- Research

EXAMPLE:

A **data scientist** writes Python code, creates graphs, and trains ML models in Jupyter Notebook.

9. Keras

Keras is a high-level Deep Learning library built on top of TensorFlow.

Features

- Easy to learn.
- Less coding.
- Fast model building.
- Supports Deep Learning.

Applications

- Image classification
- Text analysis
- Speech recognition
- EXAMPLE: A **hospital** builds a Deep Learning model to detect diseases from X-ray images using Keras.

10. Shogun

Shogun is an open-source Machine Learning library that supports many programming languages.

Features

- Fast performance.
- Supports SVM and classification algorithms.
- Suitable for research.

Applications

- Pattern recognition
- Classification
- Bioinformatics

EXAMPLE: A **research laboratory** uses Shogun to classify DNA or protein data for medical research.

11. Accord.NET

Accord.NET is a Machine Learning framework for the .NET platform developed using C#.

Features

- Supports Machine Learning.
- Image processing.
- Statistical analysis.
- Signal processing.

Applications

- Computer vision
- Face detection
- Industrial automation

EXAMPLE: A **security system** developed in C# uses Accord.NET for face recognition to unlock doors.

Scope of Machine Learning

1. Broad Industry Adoption:

> **Machine Learning is widely used in many industries to solve real-world problems and improve efficiency.**

- **Healthcare:** ML helps doctors diagnose diseases, analyze medical images, and discover new medicines.
- **Finance:** ML detects fraud, supports automated trading, and predicts financial risks.
- **Retail/E-commerce:** ML recommends products, groups customers, and adjusts prices based on demand.
- **Automotive:** ML enables self-driving cars, predicts traffic, and improves vehicle maintenance.
- **Manufacturing:** ML improves production, predicts machine failures, and increases factory efficiency.
- **IoT & Smart Devices:** ML allows smart devices to make intelligent decisions using real-time data.

2. Technical Advancements and Trends :

- **Deep Learning & Generative AI**: Powerful tools revolutionizing NLP and CV
- **Edge ML & IoT integration**: Enabling real-time intelligent processing
- **Quantum ML**: Early-stage field promising major leaps in computation

Technical Advancements and Trends

Definition:
Technical advancements are **new technologies and improvements** in Machine Learning that make computers **faster, smarter, and more accurate** in solving real-world problems.
Simple Example:
Earlier, mobile phones could only make calls. Today, smartphones can recognize faces, understand speech, and translate languages because of advancements in Machine Learning.

1. Deep Learning & Generative AI
Definition
Deep Learning is a branch of Machine Learning that uses **artificial neural networks** to solve complex problems like image recognition and speech recognition.
Generative AI is a type of AI that can **create new content**, such as text, images, music, videos, or computer code.
Student Example
Example 1: Student Notes

- A student gives a topic like **"Machine Learning"**.
- Generative AI creates complete notes or an assignment.

Example 2: Drawing Pictures

- A student types:

"Draw a tiger in a forest."

- The AI automatically creates the image.

Where It Is Used?

- Chatbots
- Image generation
- Language translation
- Medical image analysis
- Self-driving cars

NLP (Natural Language Processing) enables computers to **understand, interpret, and generate human language**.
Example

- Translating English into Telugu.
- Chatbots answering customer questions

CV (Computer Vision) enables computers to **see and understand images or videos**.
Example

- Face Unlock on a smartphone.
- Detecting vehicles in traffic cameras

2. Edge ML & IoT Integration
Definition
Edge Machine Learning means the ML model runs **directly on the device** instead of sending data to the cloud.
IoT (Internet of Things) means **smart devices connected to the internet** that can collect and exchange data.
Example 1: Smart CCTV Camera

- A CCTV camera detects a person immediately without sending the video to the internet.
- It sends an alert instantly.

Example 2: Smart Watch

- A smartwatch measures your heart rate.
- It immediately warns you if your heartbeat becomes abnormal.

Why is it Useful?

- Faster results
- Works even with slow internet
- Better privacy because data stays on the device

3. Quantum Machine Learning (Quantum ML)
Definition
Quantum Machine Learning combines **Quantum Computing** with **Machine Learning** to solve very complex problems much faster than normal computers.
Note: It is still an emerging technology and is under active research.
Student Example
Example 1: Weather Prediction

- A normal computer may take several hours to process large weather data.
- A quantum computer could perform the same task much faster.

Example 2: Drug Discovery

- Scientists use Quantum ML to test thousands of chemical combinations quickly and help discover new medicines.

Future Applications

- Medical research
- Space research
- Climate prediction
- Financial analysis
- Cybersecurity

Applications of Machine Learning

1. Image Recognition
Definition
Image Recognition is the ability of a computer to identify people, objects, animals, or places from an image. It works by comparing the image with previously learned data.
Example
Example 1: Face Unlock

- When you look at your mobile phone, it recognizes your face and unlocks automatically.

Example 2: Fruit Identification

- If you show a picture of an apple or banana to a mobile app, it identifies the fruit correctly.

2. Speech Recognition

Definition
Speech Recognition allows a computer to understand human speech and convert it into text or perform a command.
Example
Example 1: Voice Typing
- You speak into your phone, and it types your words automatically.

Example 2: Music Control
- You say "Play a song," and the speaker starts playing music.

3. Traffic Prediction
Definition
Traffic Prediction uses road and vehicle data to estimate traffic conditions and suggest the best route.
Example
Example 1: College Route
- While going to college, a navigation app shows a shorter route because there is traffic on the main road.

Example 2: Travel Time
- Before starting your journey, the app tells you it will take 30 minutes because of heavy traffic.

4. Product Recommendation
Definition
Machine Learning recommends products based on what a customer searches, views, or purchases.

Example
Example 1
- After buying a laptop, the shopping website recommends a laptop bag and mouse.

Example 2
- If you search for sports shoes, similar shoes are suggested on the next page.

5. Self-Driving Cars
Definition
Self-driving cars use Machine Learning to recognize roads, traffic signals, and obstacles so they can drive safely.
Example
Example 1
- The car stops automatically when a person crosses the road.

Example 2
- The car follows the road lane without driver assistance.

6. Email Spam Filtering
Definition
Machine Learning identifies unwanted or fake emails and moves them to the Spam folder.
Example
Example 1
- A fake lottery email is automatically moved to Spam.

Example 2
- Emails containing dangerous links are detected and blocked.

7. Virtual Personal Assistant
Definition
A Virtual Personal Assistant understands voice commands and performs tasks for the user.
Example
Example 1
- You ask it to set an alarm for 6:00 AM.

Example 2

- You ask, "What is today's weather?" and it gives the answer.

8. Online Fraud Detection
Definition
Machine Learning detects unusual banking or online payment activities to prevent fraud.
Example
Example 1

- If someone tries to use your ATM card in another city, the bank detects it as suspicious.

Example 2

- An online payment is stopped because it looks different from your normal transactions.

9. Stock Market Trading
Definition
Machine Learning studies previous stock prices to predict future market trends.
Example
Example 1

- An investor checks predicted stock prices before buying shares.

Example 2

- A trading system automatically buys or sells shares based on market conditions.

10. Medical Diagnosis
Definition
Machine Learning helps doctors detect diseases by analyzing medical reports and scan images.
Example
Example 1

- A computer identifies a brain tumor from an MRI scan.

Example 2

- A hospital system predicts whether a patient has diabetes using health data.

11. Automatic Language Translation
Definition
Machine Learning translates text or speech from one language to another automatically.
Example
Example 1

- A tourist translates an English sentence into Telugu.

Example 2

- A student translates a Hindi paragraph into English for study.

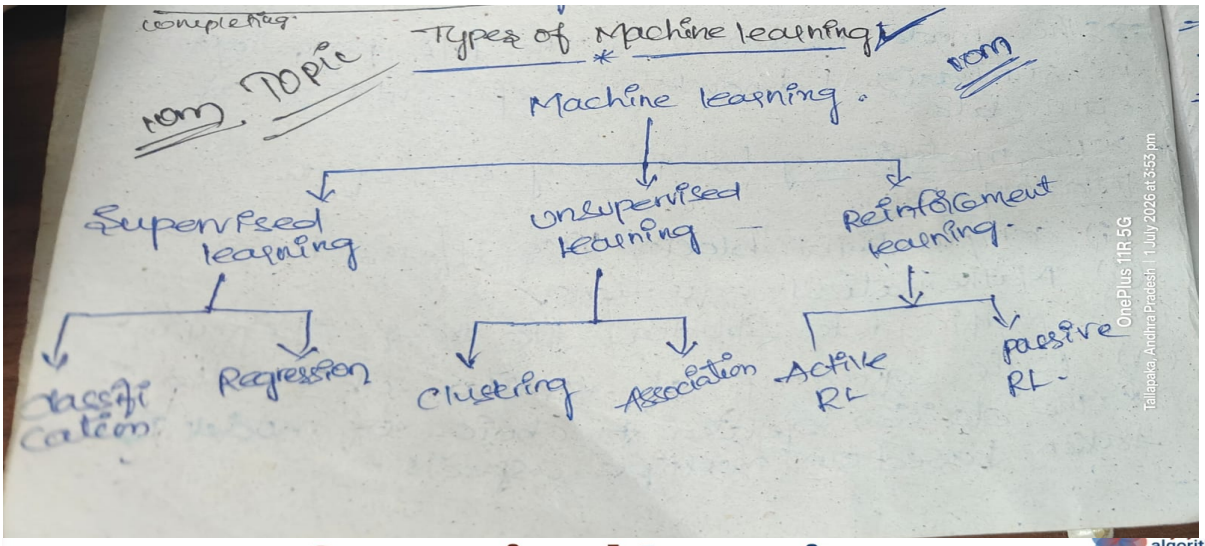
UNDERSTANDING WAY :

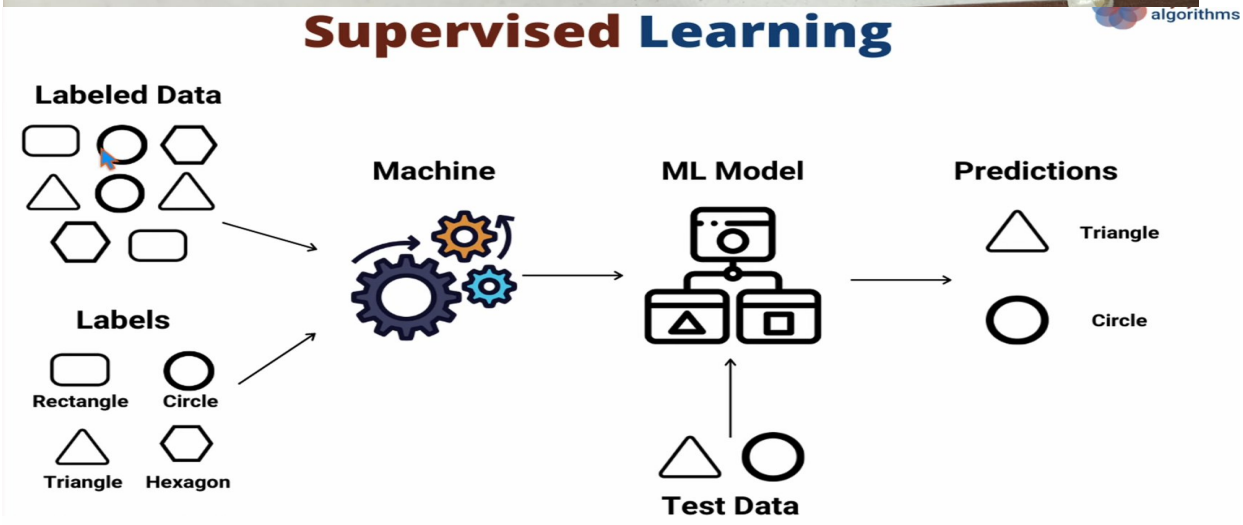
Application	**Simple Example 1**	**Simple Example 2**
Image Recognition	Face Unlock	Cat and Dog Detection
Speech Recognition	Voice Typing	Voice Commands
Traffic Prediction	Route Suggestion	Travel Time Prediction
Product Recommendation	Laptop Bag Suggestion	Movie Recommendation
Self-Driving Cars	Automatic Braking	Lane Detection
Email Spam Filtering	Spam Email Detection	Malware Detection
Virtual Personal Assistant	Setting Alarm	Weather Information
Online Fraud Detection	Bank Fraud Detection	Fake Payment Detection
Stock Market Trading	Stock Price Prediction	Automatic Trading

Application	Simple Example 1	Simple Example 2
Medical Diagnosis	Disease Detection	Diabetes Prediction
Automatic Language Translation	English to Telugu Translation	Speech Translation

TYPES OF MACHINE LEARNING:

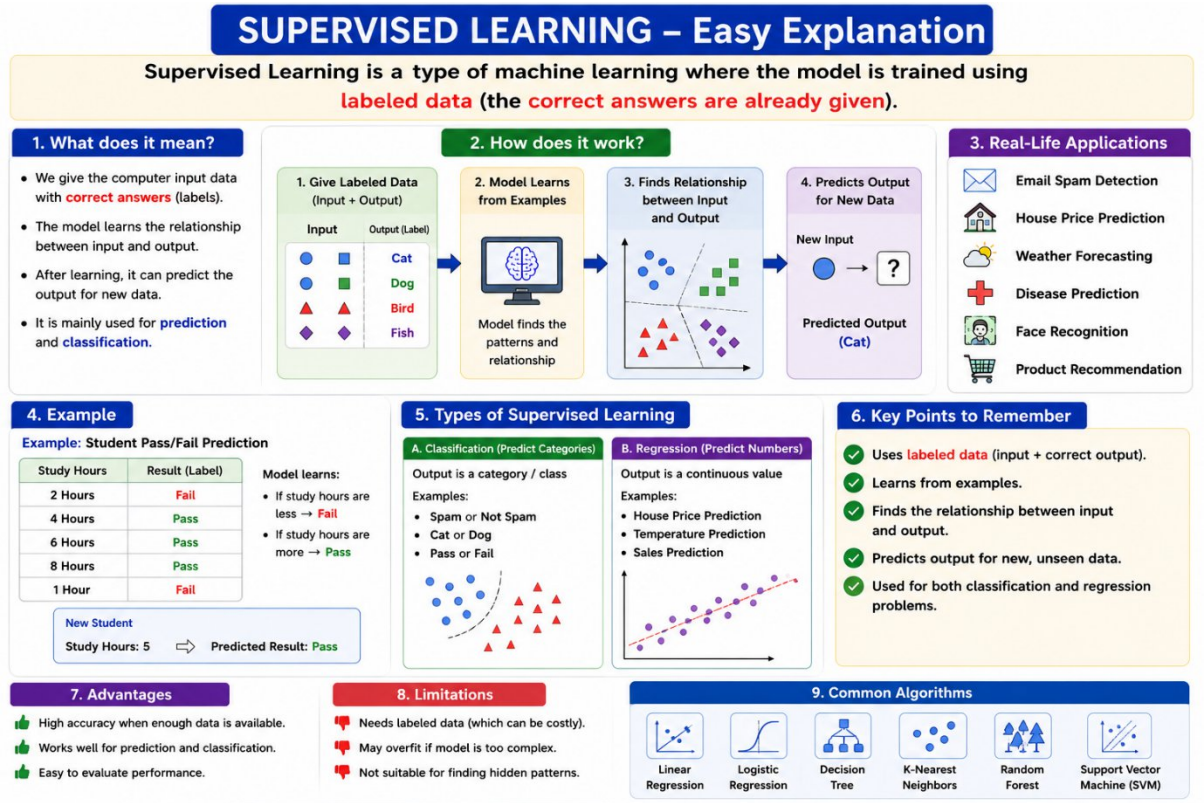
Machine Learning is mainly divided into **three types** based on how the model learns from data.





- Supervised learning is a type of machine learning where a model is trained using labeled data.
- In this process, the model learns to make predictions or decisions by being shown input data along with the correct output.

(or)



- It is like a **student learning from a teacher**. The teacher provides the correct answers, and the student learns from them. Similarly, the machine learning model learns from examples with known outputs.

- Super vised machine learning is based on the super vision .
- Super vised machine learning is a labelled data set and based on the training ,the machine predicts the output .
- Labelled data means : group of the objects (or)each and every data has some names and labels .
- Based on the input data set the machine will predicts classification of data by with the help of the understanding the feature of the objects the such as height ,shape etc…..

APPLICATIONNS OF ML :

- Fraud detection
- Risk assessment
- Spam filtering

Example:

Email Spam Detection:

- Input: Email text
- Output: "Spam" or "Not Spam"
- The algorithm is trained on many emails that are already labeled as spam or not spam.

Supervised Learning is mainly divided into **2 types**:

1. Classification

Definition

Classification is used when the **output is a category or label (discrete values).**. It predicts which class an input belongs to.

Output

- Categorical (Yes/No, Spam/Not Spam, Positive/Negative)

Examples

- Email classification: Predicting whether an email is "Spam" or "Not Spam".
- Disease detection: Classifying whether a patient has "Diabetes" or "No Diabetes"

SOME POPULAR CLASSIFICATION ALGM'S:

- RANDOM FOREST ALGORITHM .
  - DECISION TREE
- LOGISTIC REGRESSION
- SPPORT VECTOR MACHINE LEARNING ALGORITM .

2. Regression:

Definition

- Regression is used when the **output is a continuous numerical value**. It predicts a number based on the input data.
- To predict the numerical features.d

Output

- Numerical values (Price, Temperature, Salary)

Examples

- House Price Prediction
- Salary Prediction

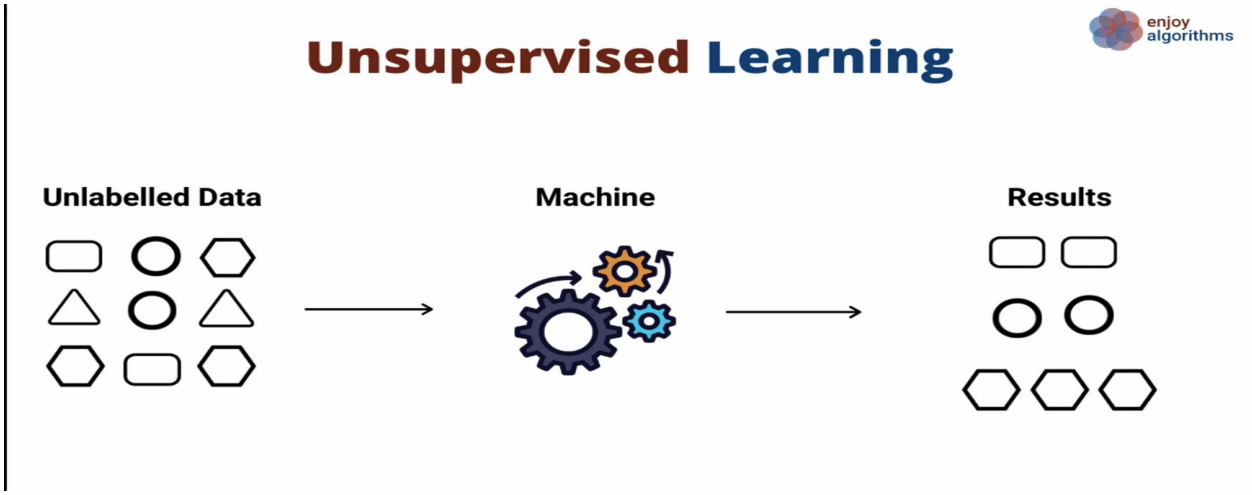
SOME POPULAR REGRESSION ALGM'S:

- SIMPLE LINEAR REGRESSION ALGMS
- MULTIVARIATE REGRESSION ALGMS.
- DECISION TREE ALGM
- LASSO REGRESSION.

Application of supervised learning :

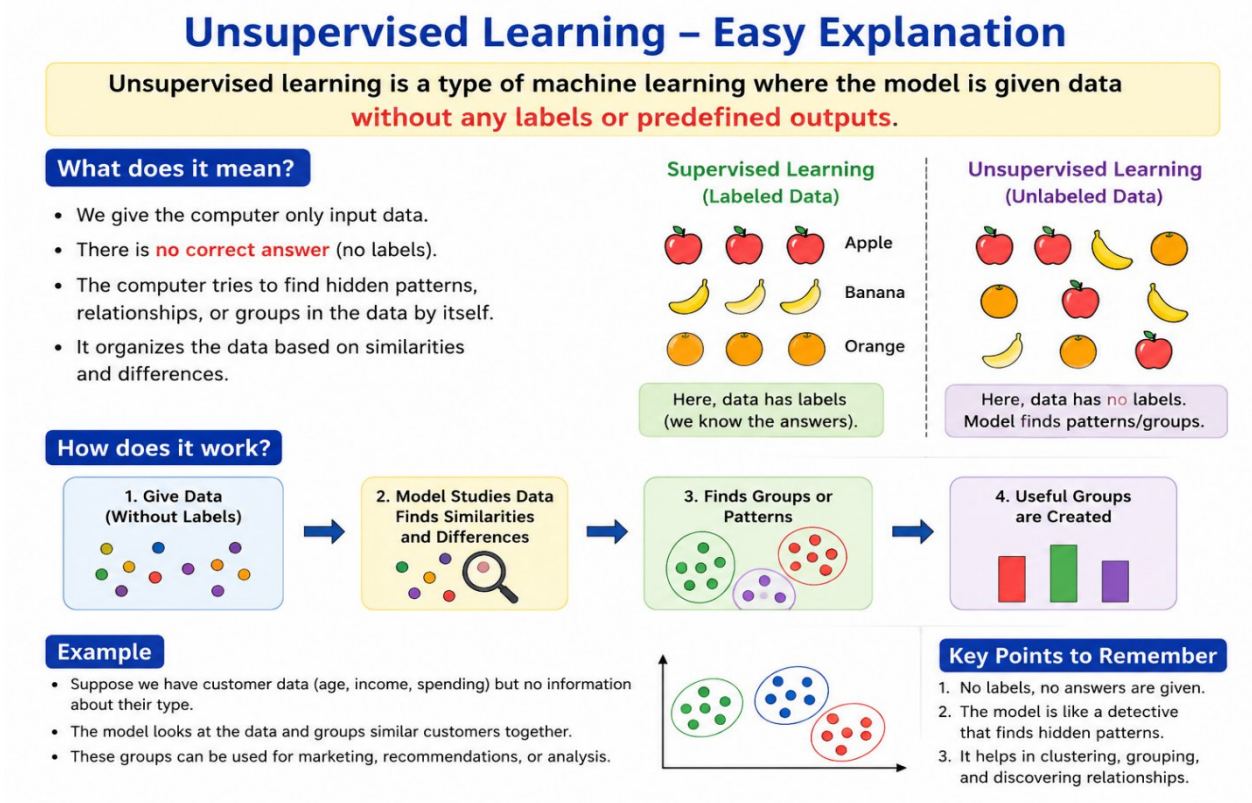
- Image recognition
- Medical diagnosis
- Spam detection
- Fraud detection
- Speech recognition

UNSUPERVISED LEARNING :



- Unsupervised learning is a type of machine learning where the model is given data without any labels or predefined outputs.
- The model tries to find patterns, relationships, or groupings in the data on its own, without being told what the correct answers are.

(or)



(or)

- It is like giving students a set of objects without any instructions.
- They identify similarities and group the objects by themselves. Similarly, the model discovers hidden patterns and relationships in the data.

- Unsupervised machine learning, the machine is trained using the un labelled dataset, And the machine predicts the output without any super vision .
- The main aim of the unsupervised learning algorithm is to group or categories the unsorted data set according to the similarities patterns and differences.
- Machines are instructed to find the hidden patterns from the input data set .
- The machine will discover it's patterns and differences, such as color differences, shape differences.
- Then it should predict the output when it is tested with the test dataset .

Example:
Customer Segmentation:
- Input: Customer purchasing behavior (no labels)
- Output:Groups of customers with similar buying habits.

Common Algorithms: K-Means Clustering, Hierarchical Clustering, PCA (Principal Component Analysis).

Unsupervised Learning is broadly divided into 2 types :
1.)Clustering :

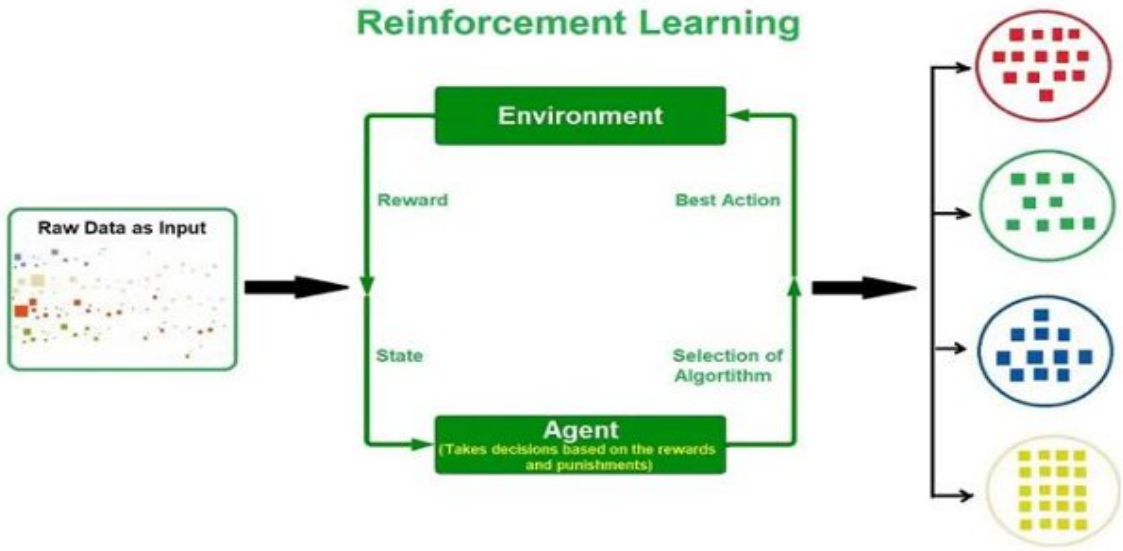
- Clustering is one type of unsupervised learning .
- Clustering is the process of grouping similar data points together based on similarities or distances.

Example:
- Grouping customers based on purchasing habits.
- Grouping news articles by topics without predefined labels.

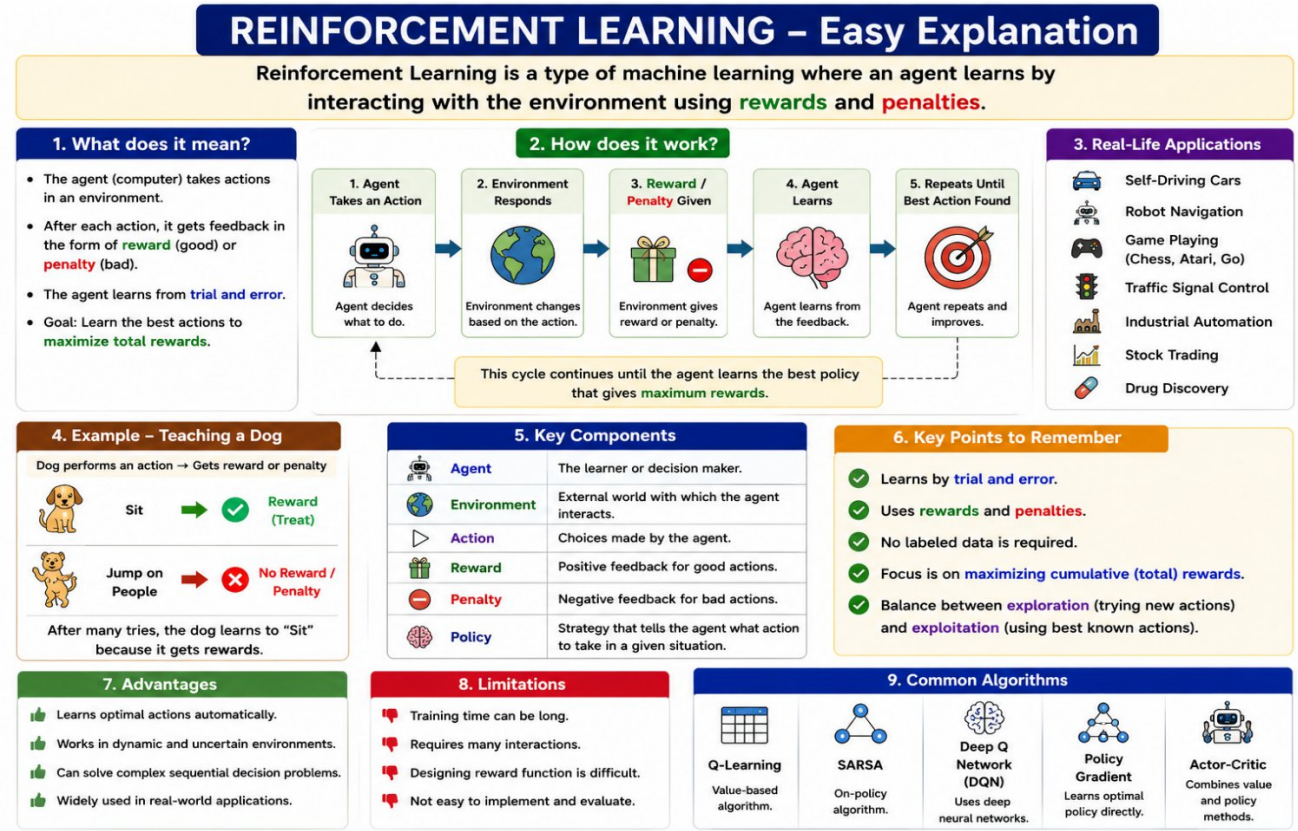
2. Association:
- Association is the process of finding rules and relationships between variables in large datasets.
- Example:
  - Market basket analysis: "If a customer buys bread, they are likely to buy butter."
  - Online recommendations (e.g., "People who bought this product also bought…").

Applications of unsupervised learning :
- Network analysis
- Recommendation systems
- Anamoly detection
- Singular value decomposition.

3. Reinforcement Learning:



(or)



A Problem can be formalized using Markov decision process

- **Reinforcement Learning** is a type of Machine Learning in which an **agent learns by interacting with the environment**.

- It improves its performance by receiving **rewards for correct actions** and **penalties for wrong actions** until it learns the best action.

- There is no specific input or output here. Just leave it in the environment. Let it train itself through its own infinite process.
- RL stands for reward-based learning—move and gain experience.

How It Works

- **Agent → Takes Action → Gets Reward or Penalty → Learns → Improves Performance**

Example 1: Self-Driving Car

- The car stays in its lane → **Reward** ✓
- The car hits an obstacle → **Penalty**

Example 2: Robot Learning

- The robot picks the correct object → **Reward** ✓
- The robot drops the object → **Penalty**

They are divided into two types:

Active Reinforcement Learning and **Passive Reinforcement Learning** are two ways an agent learns in Reinforcement Learning.

1. Passive Reinforcement Learning:
Definition:

- In **Passive Reinforcement Learning**, the agent **follows a fixed policy (rules)** and learns **how good that policy is**. It does **not choose new actions**.

Passive RL:

- **"Follow and Learn"**
- Fixed rules
- No new decisions

Applications:

- Video games
- Robotics
- Text mining
- Resource management etc….

2.)Active Reinforcement Learning:

- Active Reinforcement Learning is a type of RL in which the agent **chooses its own actions** and learns the best policy through rewards and penalties.

Active RL

"Choose and Learn"

- Makes decisions
- Finds the best action

Examples and applications

- Self-driving car
- Robot navigation

Characteristics	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	Supervised learning is when the algorithm learns on a labeled dataset and analyses the training data. These labeled data sets have inputs and expected outputs.	Unsupervised learning is when the algorithm learns on unlabeled data, inferring more about hidden structures to produce accurate and reliable outputs.	Reinforcement learning is the training of machine learning models to make a sequence of decisions. It concerns how intelligent agents are to take action in an environment to maximize cumulative reward.
Types Of Data	Labeled	Unlabeled	No Predefined Data
Goal	To Predict correct Labels for newly inputted data	To find hidden patterns through similarities and differences between data points	To find an effective model that maximize the benefits of the reward signal through the trail- error method
Output	Mapping	Classes	Action

LINEAR REGRESSION

What is Linear Regression?

Definition:

Linear Regression is a **supervised machine learning algorithm** used to **predict a continuous (numeric) value** based on one or more input variables.

(or)

Linear Regression is used to predict a numeric value by drawing the best-fit straight line between input and output data.

(or)

Linear Regression is a supervised machine learning algorithm used for predicting a continuous target variable based on one or more independent (input) variables.

How Does Linear Regression Work?

Training Data
↓
Learn Relationship
↓
Draw Best-Fit Line
↓
Predict New Values

Linear Regression Equation $Y = a + bX$

Where:

- **Y** = Predicted Output (Dependent Variable)
- **X** = Input (Independent Variable)
- **a** = Intercept (starting point)
- **b** = Slope (rate of change)

It models the relationship between the dependent variable Y and the independent variable(s) X by fitting a straight line (linear equation) to the data.

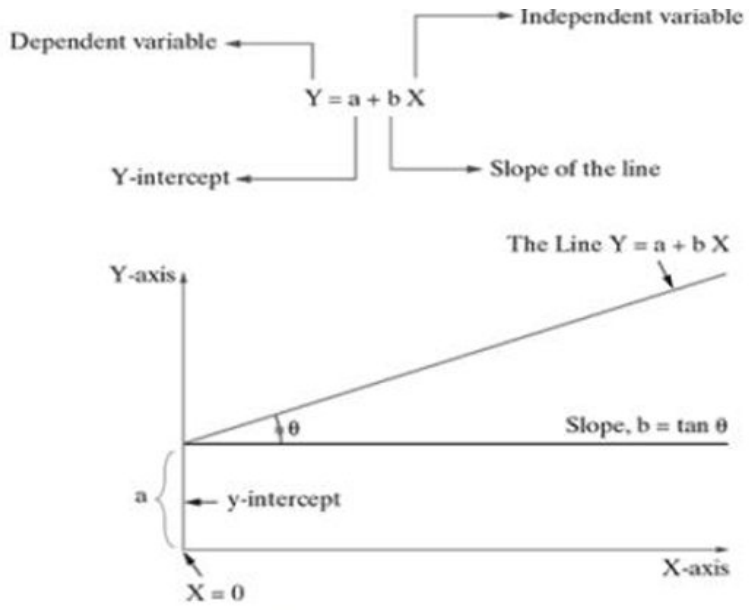


FIG. 8.1 Simple linear regression

- **The diagram represents the Simple Linear Regression model.**
- **The horizontal axis (X-axis)** represents the **Independent Variable (Input)**.
- **The vertical axis (Y-axis)** represents the **Dependent Variable (Output)**.
- **The relationship between X and Y** is represented by the equation **Y = a + bX**

- **'a' is the Y-intercept**, which is the value of **Y when X = 0**. It represents the starting value of the regression line.
- **'b' is the slope** of the regression line. It indicates **how much Y changes for every one-unit increase in X**.
- **The slanted line** in the graph is called the **Regression Line** or **Best-Fit Line**.
- **The regression line** is used to **predict the value of Y** based on the value of X.
- **The upward slope** of the line indicates a **Positive Linear Relationship** between X and Y.
- **A positive relationship** means that **as the value of X increases, the value of Y also increases**.
- **Simple Linear Regression** is widely used to **analyze relationships and make predictions** in Machine Learning.

- A **continuous numerical value** is a value that can take **any number**, such as 10, 25.5, 67.8, 100.25, etc.

- It is **not limited to fixed categories** like Yes/No or Pass/Fail.

Example : A data of 14 students is given below

Internal Exam	16	15	23	18	23	24	22	22	19	19	16	24	11	24
External Exam	49	49	63	58	60	58	61	60	63	60	52	62	30	59

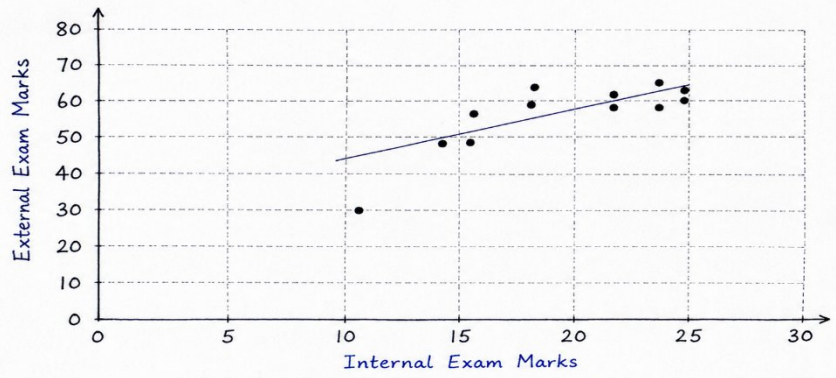
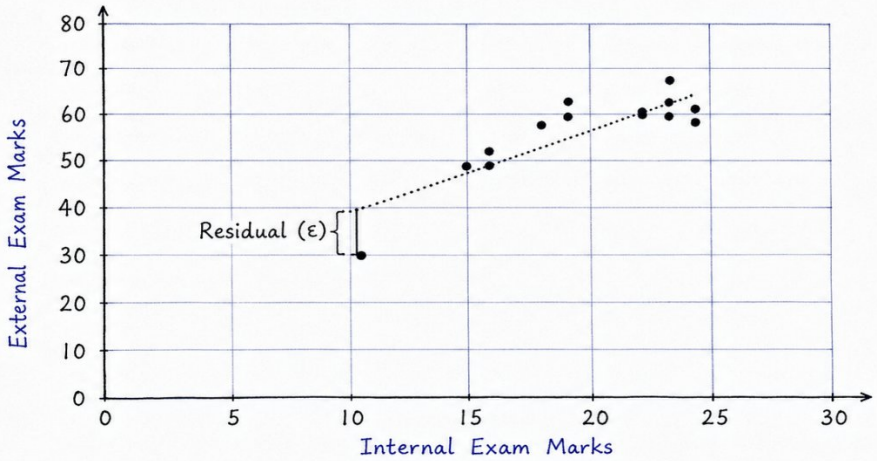


FIG. 8.8 Scatter plot and regression line

- Residual (ε) : It is the distance between the predicted point and the actual point.

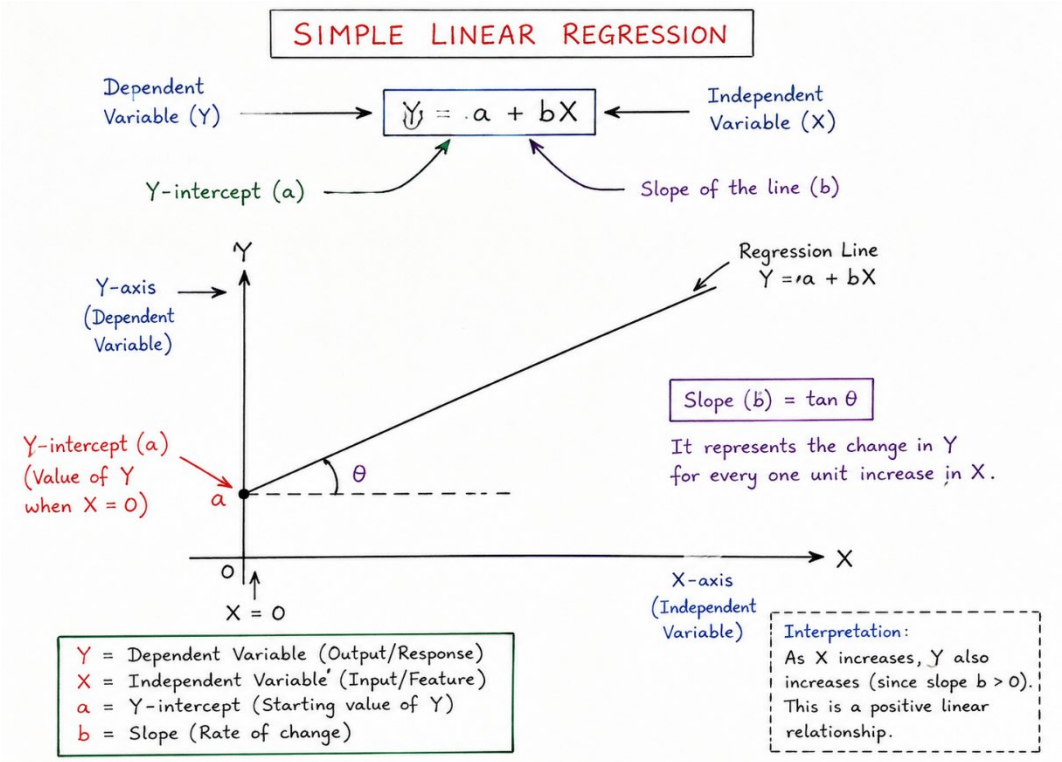


point as depicted in Figure 8.9.

As we know, in simple linear regression, the line is drawn using the regression formula.

$Y = (a + bX) + \varepsilon$

(OR)



Types of Linear Regression

➢ Simple Linear Regression:

If a single Independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

➢ Multiple Linear Regression:

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

1. Simple Linear Regression (SLR)

Definition

Simple Linear Regression is used to predict the value of **one dependent variable (Y)** using **one independent variable (X)**.

Equation

$$Y = a + bX$$

Where:

- **Y** = Dependent Variable (Output)
- **X** = Independent Variable (Input)
- **a** = Y-intercept
- **b** = Slope

Example

Predicting **External Exam Marks** based on **Internal Exam Marks**.

- X = Internal Marks
- Y = External Marks

Ex: Uses **one independent variable**.

Study Hours → Marks

Applications

- Student Marks Prediction
- House Price Prediction (using area only)
- Salary Prediction (using experience only)

2. Multiple Linear Regression (MLR)

Definition

Multiple Linear Regression is used to predict the value of **one dependent variable (Y)** using **two or more independent variables (X_1, X_2, X_3, …)**.

Equation

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_nX_n$$

Where:

- **Y** = Dependent Variable
- **X_1, X_2, X_3** = Independent Variables
- **a** = Intercept

- **b_1, b_2, b_3** = Coefficients (slopes)

Example

Predicting **Student Final Marks** using:

- Internal Exam Marks
- Attendance
- Assignment Marks
- Lab Marks

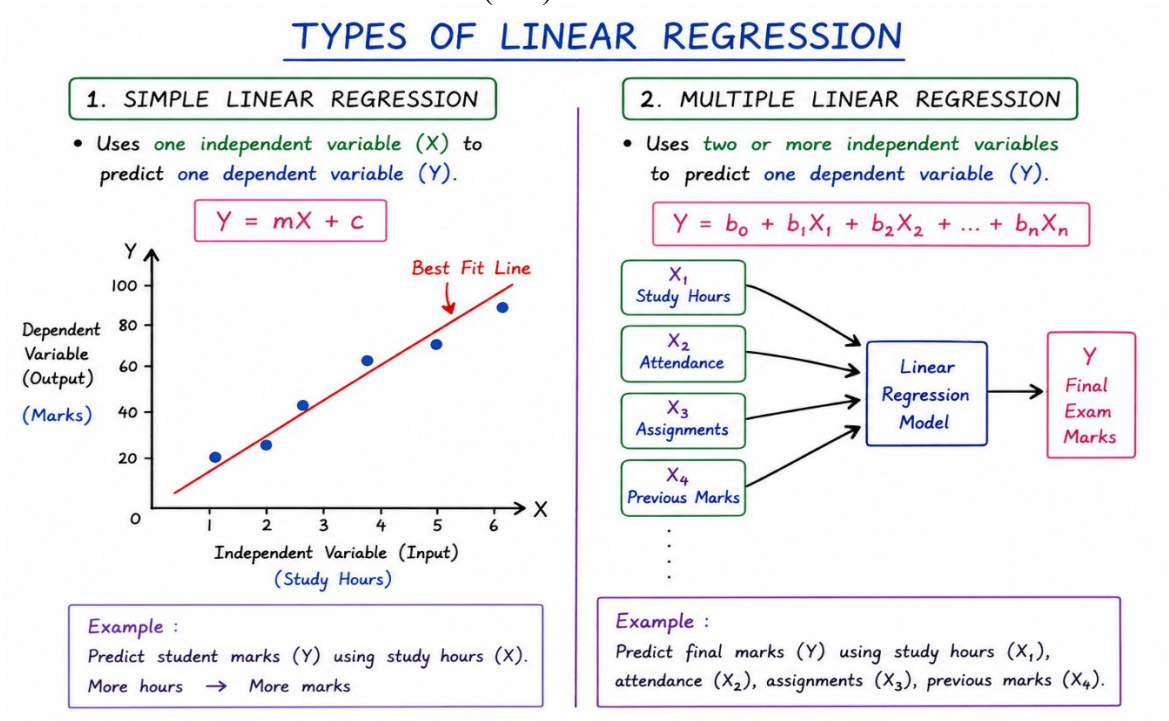
  - (or)

- Study Hours + Attendance + Assignment Marks → Final Exam Marks

Applications

- House Price Prediction (Area, Bedrooms, Location)
- Sales Prediction
- Weather Forecasting
- Medical Diagnosis

(OR)



Comparsion SLR VS MLR:

Feature	Simple Linear Regression	Multiple Linear Regression
Number of Inputs	One (X)	Two or More (X_1, X_2, X_3...)
Number of Outputs	One (Y)	One (Y)
Equation	(Y = a + bX)	(Y = a + b_1X_1 + b_2X_2 + \cdots + b_nX_n)
Complexity	Simple	More Complex
Example	Marks vs Study Hours	Marks vs Study Hours + Attendance + Assignments

LEAST SQUARES :

- Least squares linear regression is a method used to find the best-fitting straight line through a set of data points by minimizing the sum of the squares of the differences between the observed values and the values predicted by the line.
- The most common implementation of least squares linear regression is Ordinary Least Squares(OLS).

Algorithm :

1. Calculate the mean of x and y
2. Calculate the errors of x and y
3. Get the product of Errors
4. Get the summation of product of errors in step3
5. Square the difference of X
6. Get the sum of the squared difference
7. Divide the output of Step 4 by output step6 to calculate b

$$b = \frac{\Sigma_i(X_i - \overline{X})(Y_i - \overline{Y})}{\Sigma_i(X_i - \overline{X})^2} = \frac{\text{cov}(X, Y)}{\text{Var}(X)}$$

8. Calculate the a value by using b

a=Y-bX

		Step 2		Step 3	Step 5
X	Y	X – mean (X)	Y – Mean (Y)	$(X_i - \bar{X})(Y_i - \bar{Y})$	$(X_i - \bar{X})^2$
15	49	-4.9	-7.8	38.22	24.01
23	63	3.1	6.2	19.22	9.61
18	58	-1.9	1.2	-2.28	3.61
23	60	3.1	3.2	9.92	9.61
24	58	4.1	1.2	4.92	16.81
22	61	2.1	4.2	8.82	4.41
22	60	2.1	3.2	6.72	4.41
19	63	-0.9	6.2	-5.58	0.81
19	60	-0.9	3.2	-2.88	0.81
16	52	-3.9	-4.8	18.72	15.21
24	62	4.1	5.2	21.32	16.81
11	30	-8.9	-26.8	238.52	79.21
24	59	4.1	2.2	9.02	16.81
19.9	56.8		$\Sigma (X_i - \bar{X})(Y_i - \bar{Y})$	429.28	$\Sigma (X_i - \bar{X})^2$ 226.93
Step 1			Step 4		Step 6

Step 7: Divide (step4 / step6)

$b = 429.28 / 226.93 = 1.89$

$b = 1.89$

Step 8: Calculate a using the value of b

$a = \bar{Y} - b\bar{X}$

$a = 56.8 - 1.89 \times 19.9$

$a = 56.8 - 37.51$

$a = 19.05$

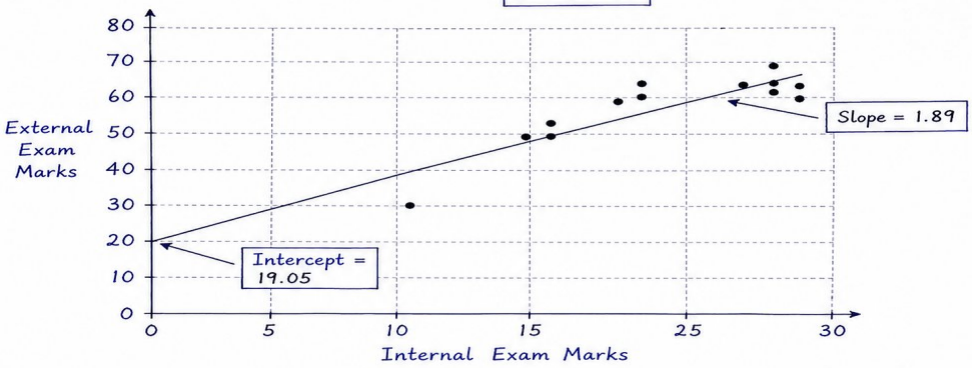
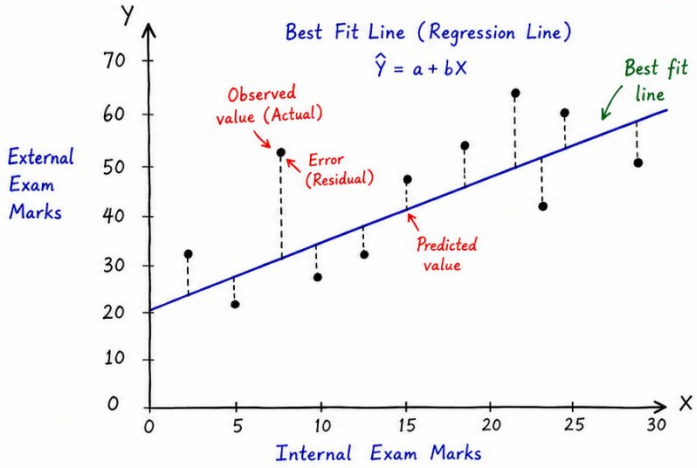


FIG. 8.11 Extended version of the regression graph

(OR)

Least Squares Linear Regression (OLS)



How Least Squares Works

1. Draw a trial straight line
2. Predict Y for each X using the line
3. Find error = Actual (Y) – Predicted ($\hat{Y}$)
4. Square each error
5. Add all squared errors
6. Adjust the line to minimize the Sum of Squared Errors (SSE)
7. The line with minimum SSE is the Best Fit Line.

Objective (Least Squares)

Minimize the Sum of Squared Errors

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

Error (Residual)

Error = Actual – Predicted

$e_i = Y_i - \hat{Y}_i$

Why Square the Error?

- Removes negative signs
- All errors become positive
- Larger errors are penalized more
- Easier for mathematical optimization

Ordinary Least Squares (OLS)

Most common method to calculate the best fit line.

Formulas:

$$b = \frac{\Sigma (X_i - \bar{X})(Y_i - \bar{Y})}{\Sigma (X_i - \bar{X})^2} \quad \text{(slope)}$$

$$a = \bar{Y} - b\bar{X} \quad \text{(intercept)}$$

Regression Equation: $\hat{Y} = a + bX$

Example (From the given data)

- Calculated values: $b = 1.89$, $a = 19.05$
- Regression Equation: $\hat{Y} = 19.05 + 1.89X$

1. Scatter Plot

- The **black dots (●)** represent the **actual student marks**.
- **X-axis** = Internal Exam Marks.
- **Y-axis** = External Exam Marks.

2. Best-Fit Line (Blue Line)

- The **blue line** is called the **Regression Line** or **Best-Fit Line**.
- It is the line that passes **closest to all the data points**.

3. Actual Value

- The **black dot** represents the **actual value** collected from the dataset.

Example:

- Internal Marks = **20**
- External Marks = **60**

4. Predicted Value

- The point on the **blue line** is the **predicted value**.
- It is calculated using the regression equation.

5. Error (Residual)

- The **dotted vertical line** between the actual point and the regression line is called the **Error (Residual)**.

Formula:

$$\boxed{\text{Error} = \text{Actual Value} - \text{Predicted Value}}$$

6. Why Square the Error?

- Removes negative signs.
- Makes all errors positive.
- Gives more importance to larger errors.
- Helps find the best-fit line accurately.

7. Objective of Least Squares

$$\boxed{SSE = \sum(Y_i - \widehat{Y}_i)^2}$$

- **SSE** = Sum of Squared Errors.
- The goal is to **minimize SSE**.
- Smaller SSE means a **better regression model**.

8. How Least Squares Works

1. Collect the dataset.
2. Draw a regression line.
3. Predict the output.
4. Calculate the error (Actual − Predicted).
5. Square each error.
6. Add all squared errors (SSE).
7. Choose the line with the **minimum SSE**.

9. Ordinary Least Squares (OLS)

- **OLS** stands for **Ordinary Least Squares**.
- It is the **most common method** used to find the **best-fit line**.

Regression Equation:

$$\boxed{\hat{Y} = a + bX}$$

Where:

- **a** = Intercept
- **b** = Slope

10. Example

Using your dataset:

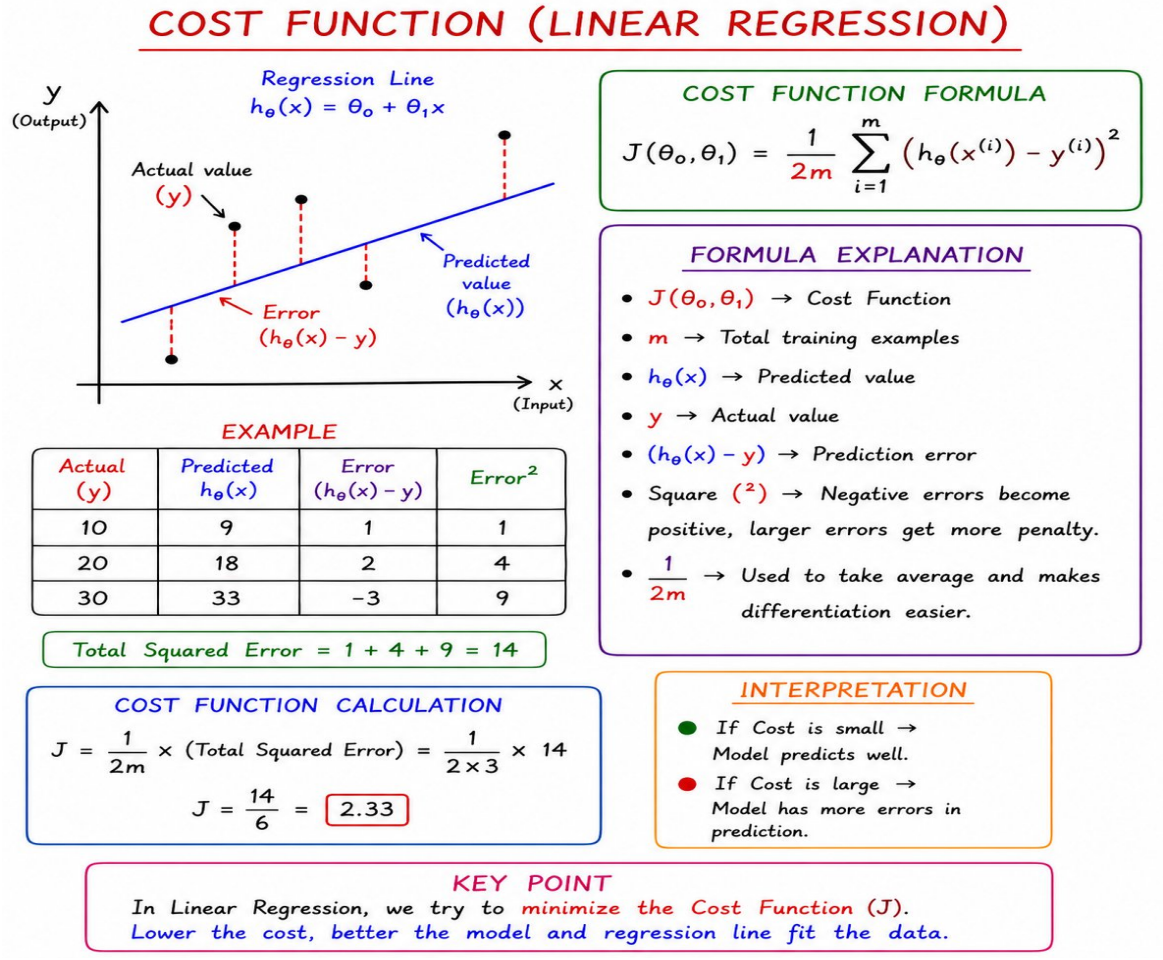
- **Slope (b) = 1.89**
- **Intercept (a) = 19.05**

Regression Equation:

$$\boxed{\hat{Y} = 19.05 + 1.89X}$$

Interpretation:

For every **1 mark increase** in the Internal Exam, the predicted External Exam marks increase by approximately **1.89 marks**



Definition
A **Cost Function** is a mathematical function that measures the difference between the **actual values** and the **predicted values**.

(or)

Definition
The **Cost Function** tells us **how much error** the Linear Regression model makes while predicting the output.

Why Do We Need a Cost Function?
Suppose we want to predict the **price of a house**.

House	Actual Price	Predicted Price
1	200	210
2	250	240
3	300	310

The predicted values are **not exactly equal** to the actual values.
Therefore, we need a method to calculate **how much error** the model is making.
That method is called the **Cost Function**.

Goal of the Cost Function
The main goal of the Cost Function is:

- **The smaller the cost, the better the regression model.**
- If the **cost is 0**, the regression model predicts all values perfectly.

Most Common Cost Function (Mean Squared Error - MSE)
The most commonly used Cost Function in Linear Regression is the **Mean Squared Error (MSE)**.
Formula

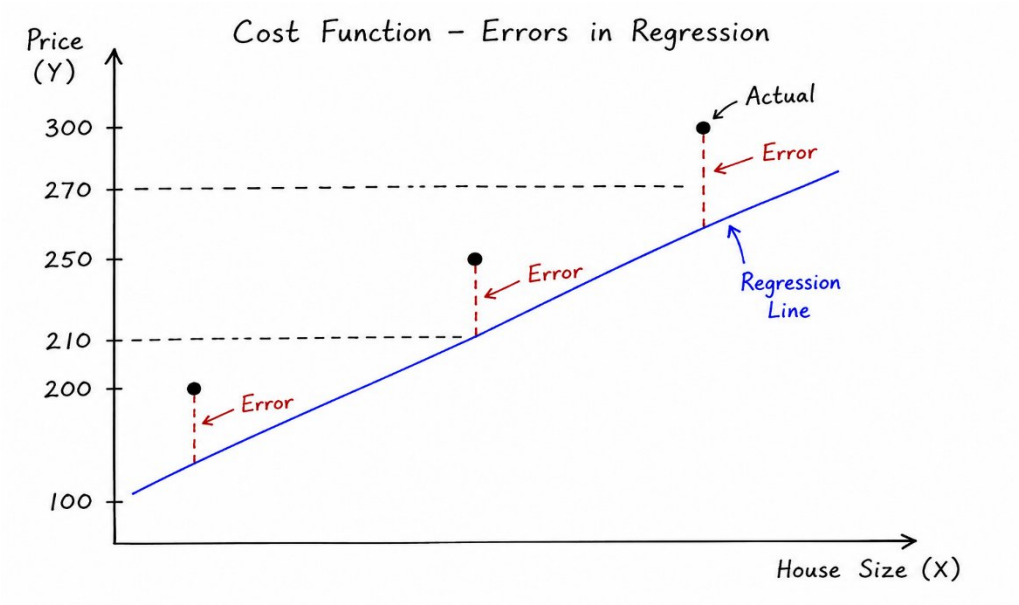
$$J(m,c) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Formula Explanation
Where:

- $J(m,c)$= Cost Function
- y_i= Actual value
- $\hat{y}_i$= Predicted value
- n= Total number of data points
- $(y_i - \hat{y}_i)$= Prediction error
- $(y_i - \hat{y}_i)^2$= Squared error

Explanation of the Graph
The graph contains:

- **Black dots** → Actual house prices.
- **Blue line** → Regression Line (Predicted values).
- **Red dotted lines** → Prediction errors (difference between actual and predicted values).



Why did we draw the Blue Line?
The **blue line** is called the **Regression Line** or **Best-Fit Line**.
We draw it because **the actual data points (black dots) do not lie on a single straight line**.
The blue line represents the **predicted values**.

Look at the figure

- **Black dots (●)** = Actual House Prices
- **Blue Line** = Predicted House Prices
- **Red Dotted Lines** = Error (Difference between Actual and Predicted)

Interpretation:
The **red dotted line** represents the **prediction error**.

- **Shorter red line** → Prediction is closer to the actual value.
- **Longer red line** → Prediction is farther from the actual value.

Objective: Make all the red error lines as **short as possible**.

Step-by-Step Calculation
Step 1: Actual Prices
Actual house prices are:

$$[200, 250, 300]$$

Step 2: Predicted Prices
The regression model predicts:

$$[210, 240, 310]$$

Step 3: Calculate the Error
Formula

$$\text{Error} = \text{Actual Value} - \text{Predicted Value}$$

Actual Price	Predicted Price	Error
200	210	-10
250	240	10
300	310	-10

Step 4: Square the Errors

We square each error to remove negative signs and give a larger penalty to bigger errors.

$$(-10)^2 = 100$$
$$(10)^2 = 100$$
$$(-10)^2 = 100$$

Error	Squared Error
-10	100
10	100
-10	100

Step 5: Calculate Total Squared Error
Add all the squared errors.

$$100 + 100 + 100 = 300$$

Therefore,

$$\boxed{\text{Total Squared Error} = 300}$$

Step 6: Calculate the Cost Function
The Cost Function is

$$J = \frac{1}{2n}\sum(y - \hat{y})^2$$

Here,

- Number of data points:

$$n = 3$$

Therefore,

$$2n = 6$$

Substituting the values,

$$J = \frac{300}{6}$$
$$\boxed{J = 50}$$

Thus, the **Cost Function value is 50**.

Why Do We Square the Errors?
Suppose,

- Error = **+10**
- Error = **−10**

If we simply add them,

$$10 + (-10) = 0$$

This incorrectly suggests that there is **no error**.
By squaring the errors,

$$(+10)^2 = 100$$
$$(-10)^2 = 100$$

Both become positive, so every error contributes correctly to the total cost.

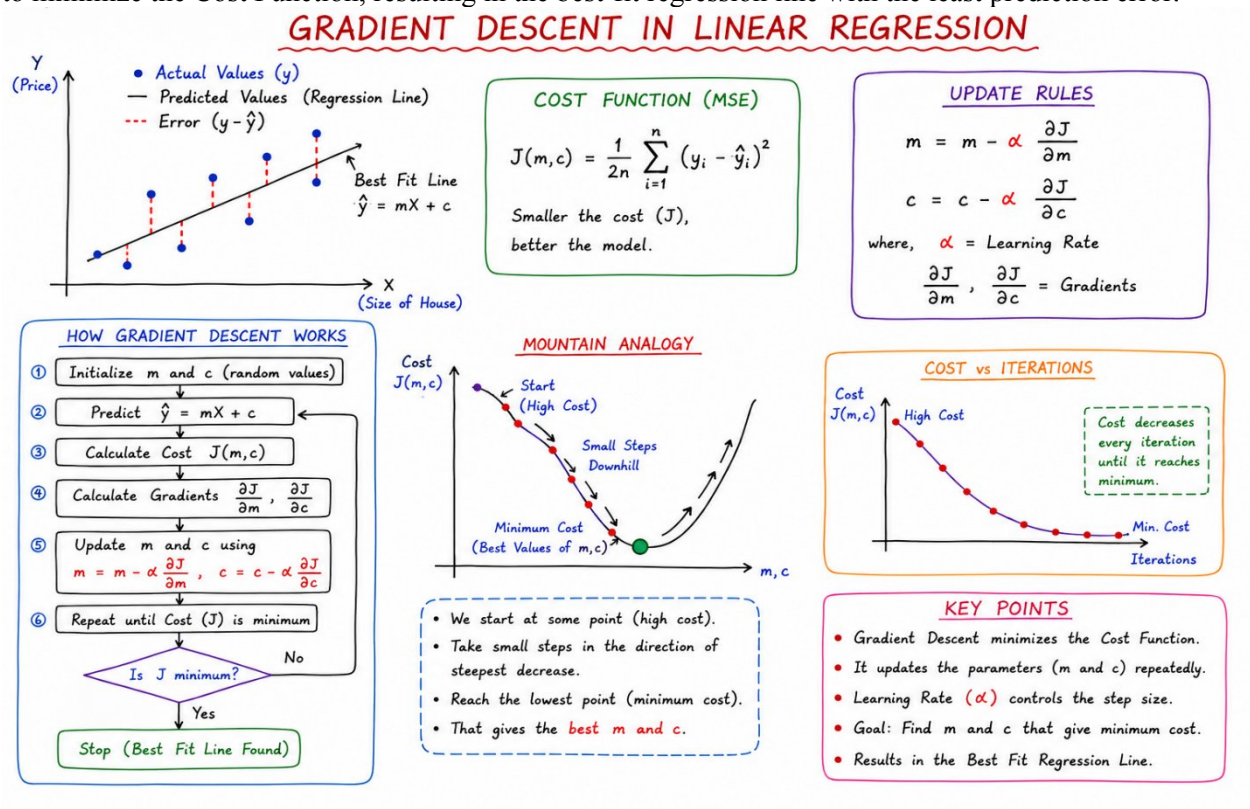
What Does Cost = 50 Mean?
A Cost Function value of **50** means there is still some prediction error in the model.

- **Smaller Cost** → Better regression model.
- **Larger Cost** → Poor regression model with higher prediction errors.

The objective of Linear Regression is to **adjust the regression line until the Cost Function becomes as small as possible**.

GRADIENT DESCENT

Gradient Descent is an optimization algorithm that repeatedly updates the slope (m) and intercept (c) to minimize the Cost Function, resulting in the best-fit regression line with the least prediction error.



1. Scatter Plot and Regression Line (Top Left)

The graph contains:

- **Blue dots** → Actual values (House Prices)
- **Black line** → Regression Line (Predicted values)
- **Red dotted lines** → Prediction Errors

Explanation

- Each **blue dot** is an actual data point.
- The **black line** predicts the output values.
- The **red dotted line** shows the **difference (error)** between the actual value and the predicted value.

Objective: Make the red error lines as **small as possible**.

2. Cost Function (Top Center)

The Cost Function is

$$J(m,c) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Explanation

- **J(m,c)** → Cost Function
- **y** → Actual value
- **ŷ** → Predicted value
- **n** → Number of data points

The Cost Function measures the **total prediction error**.
Smaller Cost = Better Model

3. Update Rules (Top Right)
Gradient Descent updates the model parameters using:

$$m = m - \alpha \frac{\partial J}{\partial m}$$
$$c = c - \alpha \frac{\partial J}{\partial c}$$

Explanation

- **m** → Slope
- **c** → Intercept
- **α (Alpha)** → Learning Rate
- **∂J/∂m** and **∂J/∂c** → Gradients

These formulas slightly change the slope and intercept after every iteration to reduce the Cost Function.

4. Mountain Analogy (Center)
The mountain represents the **Cost Function**.
Explanation

- Start at the **top of the mountain** (High Cost).
- Take **small steps downhill**.
- Continue moving down.
- Reach the **lowest point** (Minimum Cost).

The lowest point gives the **best values of m and c**.
This is exactly how Gradient Descent works.

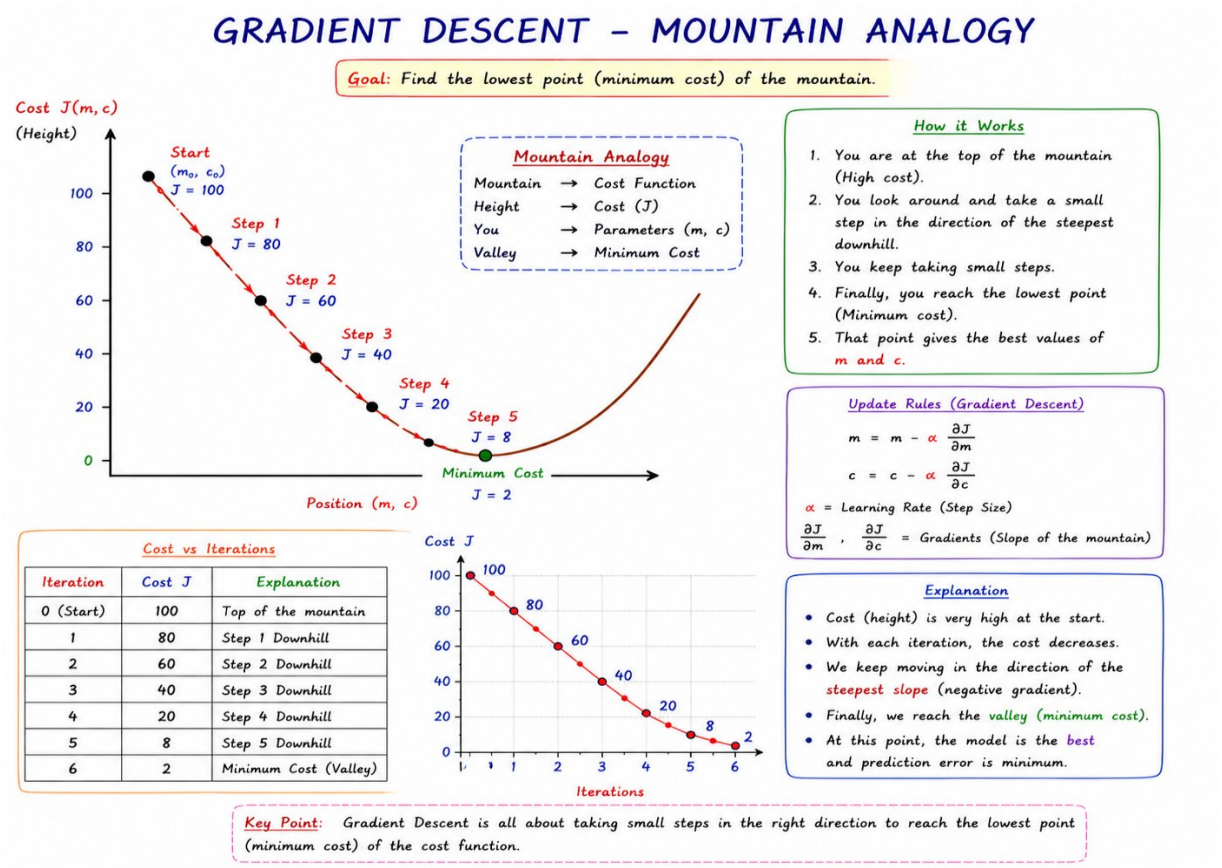
5. Cost vs Iterations (Right Bottom)
The graph shows:

- **X-axis** → Iterations
- **Y-axis** → Cost Function

Explanation

- Initially, the Cost is **high**.
- After every iteration, the Cost decreases.
- Finally, the Cost reaches its **minimum value**.

This means the model becomes more accurate after each iteration.



This figure explains **Gradient Descent** using a **mountain analogy**. Imagine you are standing on the **top of a mountain** and your goal is to reach the **lowest point (valley)**.
In Machine Learning:

- **Mountain** → Cost Function
- **Height** → Cost (Error)
- **Valley** → Minimum Cost
- **Small Steps** → Gradient Descent
- **Lowest Point** → Best-Fit Model

1. Start Point (High Cost)
At the beginning, the model has a **high error**.
Cost = 100
This means the regression model is **not predicting accurately**.

2. Step 1
The model moves a small step downhill.
Cost decreases

$$100 \rightarrow 80$$

Prediction becomes slightly better.

3. Step 2
Again, Gradient Descent updates the model parameters.

$$80 \rightarrow 60$$

The prediction error becomes smaller.

4. Step 3
The model continues moving downhill.

$$60 \rightarrow 40$$

The regression line improves further.

5. Step 4
Another update reduces the error.

$$40 \rightarrow 20$$

The model is now much closer to the best solution.

6. Step 5
The model takes another small step.

$$20 \rightarrow 8$$

Only a small amount of error remains.

7. Minimum Cost (Valley)
Finally, Gradient Descent reaches the **lowest point**.

$$8 \rightarrow 2$$

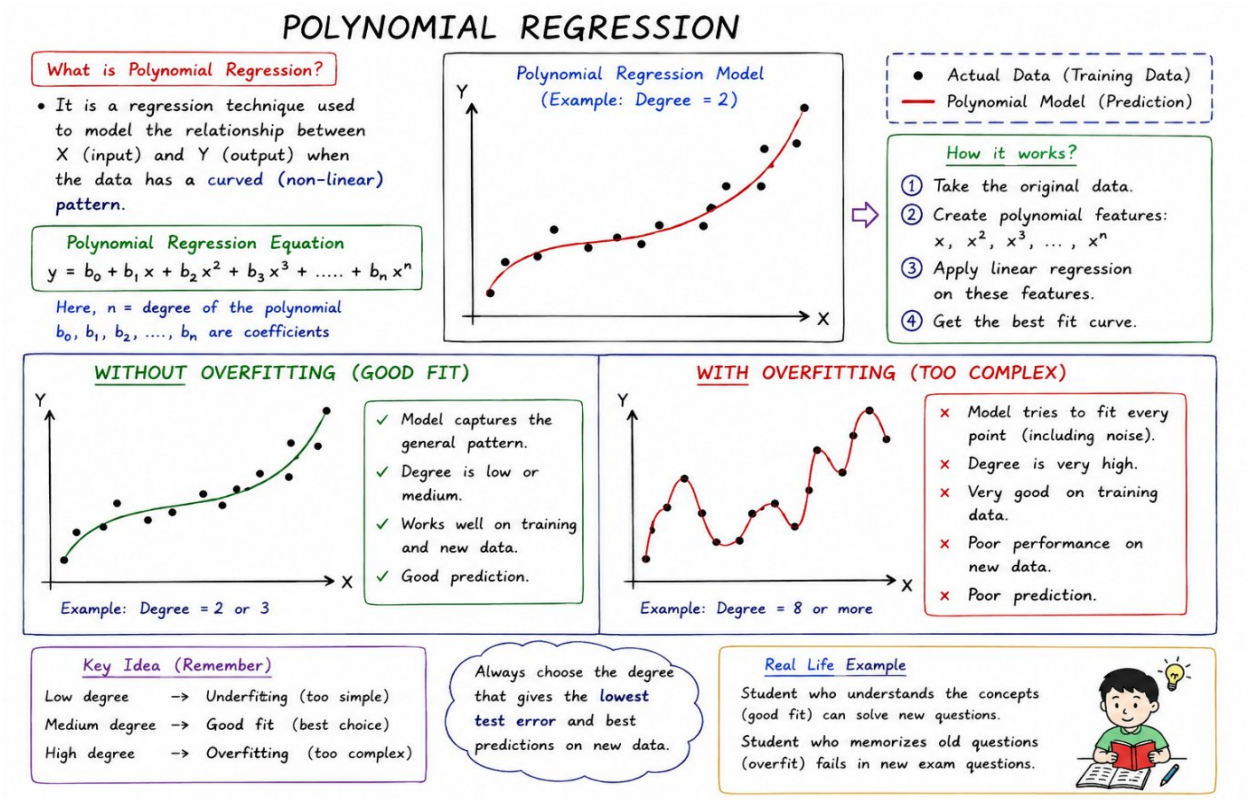
Cost = 2 (Minimum Cost)
At this point:

- Prediction error is minimum.
- The regression line is the **Best-Fit Line**.
- The values of **slope (m)** and **intercept (c)** are optimal.

Iteration	Cost (J)	Meaning
Start	100	High error
Step 1	80	Error decreases
Step 2	60	Better prediction
Step 3	40	Model improves
Step 4	20	Small error
Step 5	8	Very small error
Final	2	Minimum cost (Best Model)

How Gradient Descent Works

1. Start with a random regression line.
2. Calculate the Cost Function.
3. Move in the direction that **reduces the cost**.
4. Update the model parameters (**m** and **c**).
5. Repeat until the **minimum cost** is reached.



1. What is Polynomial Regression?

Polynomial Regression is a regression technique used when the relationship between **X (Input)** and **Y (Output)** is **curved (non-linear)**.

Input and Output Variables

Variable	Meaning	Example
X	**Input Variable (Independent Variable / Feature)**	Study Hours, House Size, Age
Y	**Output Variable (Dependent Variable / Target)**	Marks, House Price, Salary

Unlike Linear Regression, which fits a **straight line**, Polynomial Regression fits a **curved line**.

Example

Suppose we want to predict **student marks** based on **study hours**.

- If the marks increase in a curved pattern, a straight line cannot fit the data properly.
- Therefore, Polynomial Regression is used to fit a **curved best-fit line**.

2. Polynomial Regression Equation

The equation is

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_n x^n$$

Explanation

- **y** → Predicted Output
- **x** → Input Variable
- **b_0** → Intercept
- **b_1, b_2, b_3...** → Coefficients
- **n** → Degree of the Polynomial

Simple Idea

The terms **x^2, x^3, x^4...** help the model draw a **curve** instead of a straight line.

3. Polynomial Regression Model (Middle Graph)

What does the graph show?

- **Black Dots** → Actual Training Data
- **Red Curve** → Polynomial Regression Model

Explanation

The red curve passes close to most of the data points.

It learns the relationship between **X** and **Y** and gives the **best-fit curve**.

Example

If study hours increase, marks also increase in a curved pattern.

The red curve represents this relationship.

4. How Polynomial Regression Works

The diagram explains the working process in **4 simple steps**.

Step 1: Take the Original Data

Collect the dataset.

Example:

Study Hours	Marks
1	20
2	35
3	50
4	70
5	90

Step 2: Create Polynomial Features

Convert the original feature into polynomial features.

Example:

If $X = 3$ Then

- $X = 3$
- $X^2 = 9$
- $X^3 = 27$

These new features help the model learn curved relationships.

Step 3: Apply Linear Regression

Apply Linear Regression to the polynomial features.

The model calculates the best coefficients.

Step 4: Get the Best-Fit Curve

Finally, the model draws a smooth curved line that best represents the data.

5. Without Overfitting (Good Fit)

Look at the Green Box

What do you observe?

- The green curve passes close to most data points.
- It does not try to pass through every point.

Explanation

- Learns the actual pattern.
- Ignores random noise.

- Works well on both training data and testing data.
- Gives accurate predictions.

Example

A student understands the concepts instead of memorizing answers.

Even if new questions are asked in the exam, the student can answer correctly.

This is called **Good Fit**.

Characteristics

- Medium polynomial degree (Degree = 2 or 3)
- Good prediction
- Low testing error
- Good generalization

6. With Overfitting (Too Complex)

Look at the Red Box

What do you observe?

- The red curve bends many times.
- It passes through almost every data point.

Explanation

The model becomes too complex.

It learns not only the actual pattern but also the **noise (random variations)**.

As a result,

- Training accuracy becomes very high.
- Testing accuracy becomes low.
- The model performs poorly on new data.

Example

A student memorizes only previous year's questions.

If different questions appear in the exam, the student cannot answer properly.

This is called **Overfitting**.

Characteristics

- Very high polynomial degree (Degree = 8 or more)
- Fits every training point
- Learns noise
- Poor prediction on new data

7. Key Idea (Remember)

The degree of the polynomial is very important.

Low Degree

➡ Underfitting

- Model is too simple.
- Cannot learn the data properly.

Medium Degree

➡ Good Fit ✓

- Learns the actual pattern.
- Gives accurate predictions.

High Degree

➡ Overfitting ✕

- Model becomes too complex.
- Learns noise.
- Poor prediction on new data.

8. Real-Life Example

Imagine two students preparing for an exam.

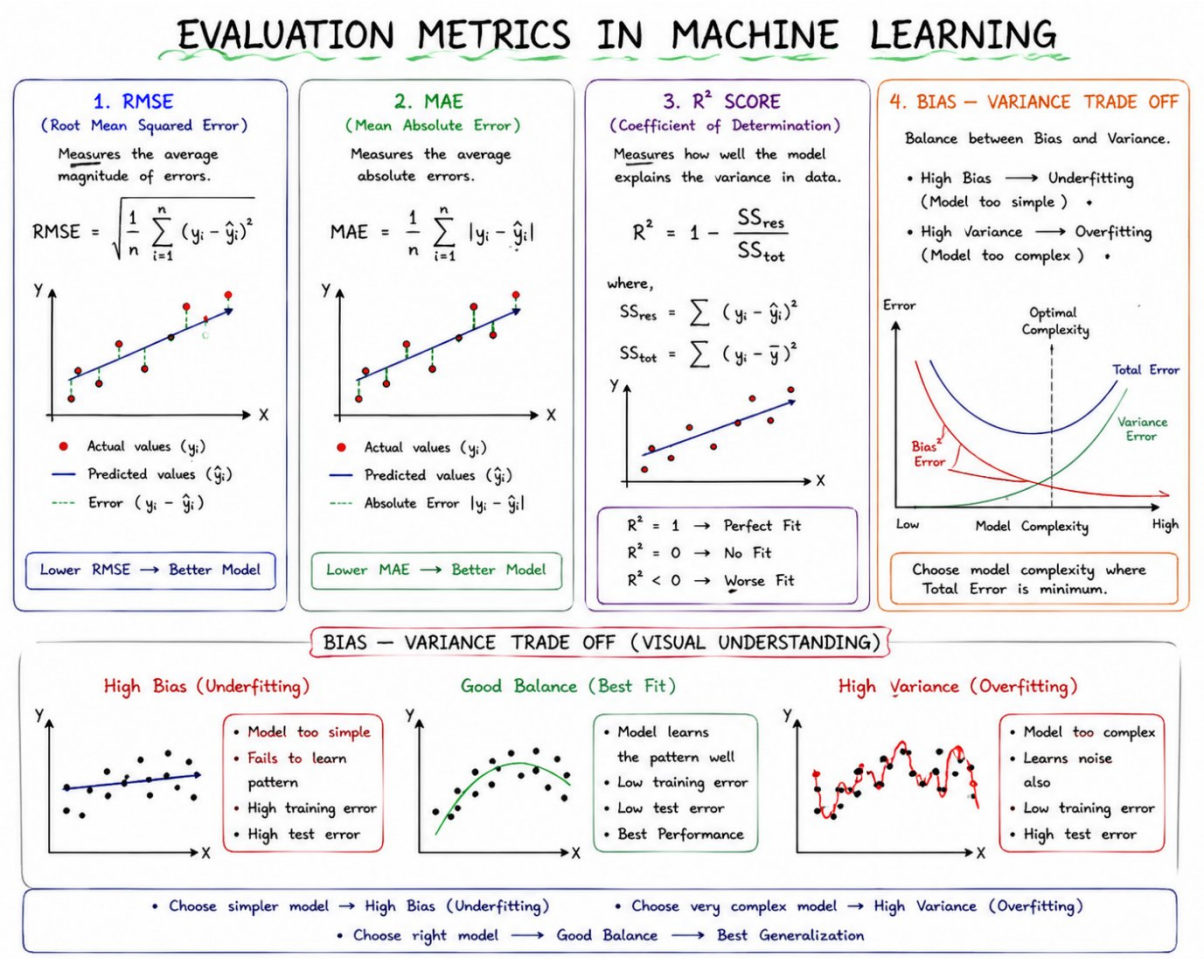
Student 1 – Good Fit

- Understands the concepts.
- Can answer both old and new questions.
- Performs well in the exam.

Student 2 – Overfitting

- Memorizes only previous year's questions.
- Cannot answer new questions.
- Performs poorly in the exam.

This is exactly how **Good Fit** and **Overfitting** work in Machine Learning.



1. RMSE (Root Mean Squared Error):

RMSE measures the **average prediction error** between the **actual values** and the **predicted values** made by a machine learning model.

It squares the errors before calculating the average, so **large errors receive more penalty**.

Explanation

- RMSE tells us **how far the model's predictions are from the actual values**.
- It squares each error, calculates the average, and then takes the square root.
- Because the errors are squared, **large errors are penalized more** than small errors

Formula

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Formula Explanation

- **n** → Total number of data points
- **y** → Actual value
- **ŷ** → Predicted value
- **(y − ŷ)** → Prediction error
- **Square (²)** → Makes all errors positive and penalizes larger errors
- √ → Converts the squared error back to the original unit

- ## Step 1: Calculate Error

Actual	Predicted	Error (Y-Ŷ)	Error²
10	9	1	1
12	14	-2	4
15	14	1	1
20	19	1	1

- Total Squared Error

$$1 + 4 + 1 + 1 = 7$$

RMSE

$$RMSE = \sqrt{\frac{7}{4}} = \sqrt{1.75} = 1.32$$

- ## Interpretation
- **RMSE = 1.32**
- ➡ Lower RMSE means a better model.

Example

Suppose the actual house price is **₹50 lakh**, but the model predicts **₹48 lakh**.

Error = 50 − 48 = **2 lakh**

RMSE calculates the average of all such prediction errors.

Key Point

Lower RMSE = Better Model Performance

2. MAE (Mean Absolute Error)

Definition

MAE measures the **average absolute difference** between the actual values and predicted values.

It treats **all errors equally**.

Explanation

- MAE calculates the **average prediction error**.
- It ignores whether the error is positive or negative by taking the absolute value.
- Every error contributes equally to the final result.

Formula

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |$$

Formula Explanation

- **n** → Number of data points
- **y** → Actual value
- **ŷ** → Predicted value
- **|y − ŷ|** → Absolute error (always positive)

Step 1: Absolute Error

Actual	Predicted	Error	Absolute Error
10	9	1	1
12	14	-2	2
15	14	1	1
20	19	1	1

Total Absolute Error

$$1 + 2 + 1 + 1 = 5$$

MAE

$$MAE = \frac{5}{4} = 1.25$$

Interpretation

MAE = 1.25

➡ Lower MAE means a better model.

Example

Actual Marks = **90**

Predicted Marks = **85**

Error = |90 − 85| = **5**

MAE is the average of these absolute errors.

Key Point

Lower MAE = Better Model Performance

Difference Between RMSE and MAE

RMSE	**MAE**
Squares the errors	Uses absolute errors
Large errors get more penalty	All errors are treated equally
More sensitive to outliers	Less sensitive to outliers

3. R² Score (Coefficient of Determination):

Definition

The **R² Score** measures **how well the regression model explains the variation in the data**.

It indicates **how well the regression line fits the data points**.

Explanation

- It shows **how well the regression line fits the data points**.
- The value of R² lies between **0 and 1** (it can be negative in some cases).

Formula : $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$

Formula Explanation

- **SSres** → Sum of Squared Residual Errors

- **SStot** → Total Sum of Squares

R² Score (Coefficient of Determination)

Step 1: Mean of Actual Values

$$\bar{Y} = \frac{10 + 12 + 15 + 20}{4} = 14.25$$

Step 2: Calculate SSres

$$SS_{res} = (10 - 9)^2 + (12 - 14)^2 + (15 - 14)^2 + (20 - 19)^2$$
$$= 1 + 4 + 1 + 1 = 7$$

Step 3: Calculate SStot

$$SS_{tot} = (10 - 14.25)^2 + (12 - 14.25)^2 + (15 - 14.25)^2 + (20 - 14.25)^2$$
$$= 18.06 + 5.06 + 0.56 + 33.06 = 56.74$$

Step 4: Calculate R²

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$
$$R^2 = 1 - \frac{7}{56.74}$$
$$R^2 = 0.88$$

Interpretation

R² = 0.88

➡ The model explains **88%** of the variation in the data.

➡ Since R² is close to **1**, the model fits the data well.

Example

If **R² = 0.92**, it means the model explains **92% of the variation** in the data.

Key Point

Higher R² Score = Better Model Performance

Bias–Variance Trade-Off

Definition

Bias–Variance Trade-Off is the process of finding the right balance between **Bias** and **Variance** to build a model that performs well on both **training data** and **new (test) data**.

A good Machine Learning model should have:

- **Low Bias**
- **Low Variance**

so that it gives accurate predictions

What is Bias?

Definition

Bias is the error caused because the model is **too simple** to learn the actual pattern in the data. This leads to **Underfitting**

Example

🧒🏫 Teacher teaches only **important questions**.

🧒🎓 Student learns only those questions.

In the exam, if a different question appears, the student cannot answer it.

- The student did not understand the concept.

This is called **High Bias (Underfitting).**

What is Variance?

Definition

Variance is the error caused because the model is **too complex** and learns the **training data and noise**. This leads to **Overfitting**.

Example

🧒🏫 Teacher asks students to **memorize every answer** from previous question papers.

🧒🎓 Student memorizes everything.

Training questions are answered correctly.

But in the semester exam, new questions appear.

The student cannot answer them.

➡ This is **High Variance (Overfitting).**

4. Bias–Variance Trade-Off

Definition

The **Bias–Variance Trade-Off** is the process of finding the **right balance between a simple model and a complex model** to achieve the best prediction accuracy.

Explanation

A good machine learning model should:

- Learn the important patterns in the data.
- Avoid making too many mistakes.
- Generalize well to new, unseen data.

There are three situations:

(a) High Bias (Underfitting)

Definition
High Bias occurs when the model is **too simple** to learn the important patterns in the training data.

Characteristics

- Model is too simple.
- Fails to learn the data properly.
- High training error.
- High testing error.

Example
Predicting house prices using only the number of bedrooms while ignoring location, area, and age of the house.

(b) Good Balance (Best Fit)

Definition
A balanced model learns the correct patterns without memorizing the training data.

Characteristics

- Learns important patterns.
- Low training error.
- Low testing error.
- Best prediction performance.

Example
A house price prediction model that uses relevant features such as location, area, number of bedrooms, and age of the house.

(c) High Variance (Overfitting)

Definition
High Variance occurs when the model is **too complex** and memorizes the training data instead of learning general patterns.

Characteristics

- Model is too complex.
- Learns noise as well as patterns.
- Very low training error.
- High testing error.

Example
A model that memorizes the prices of all training houses but performs poorly when predicting the prices of new houses.

EASY understand the trade off :

🎯 **High Bias**

"Always makes the same mistake."

➡ Model is too simple.

➡ Underfitting.

🎯 **High Variance**

"Changes prediction every time."

➡ Model is too complex.

➡ Overfitting.

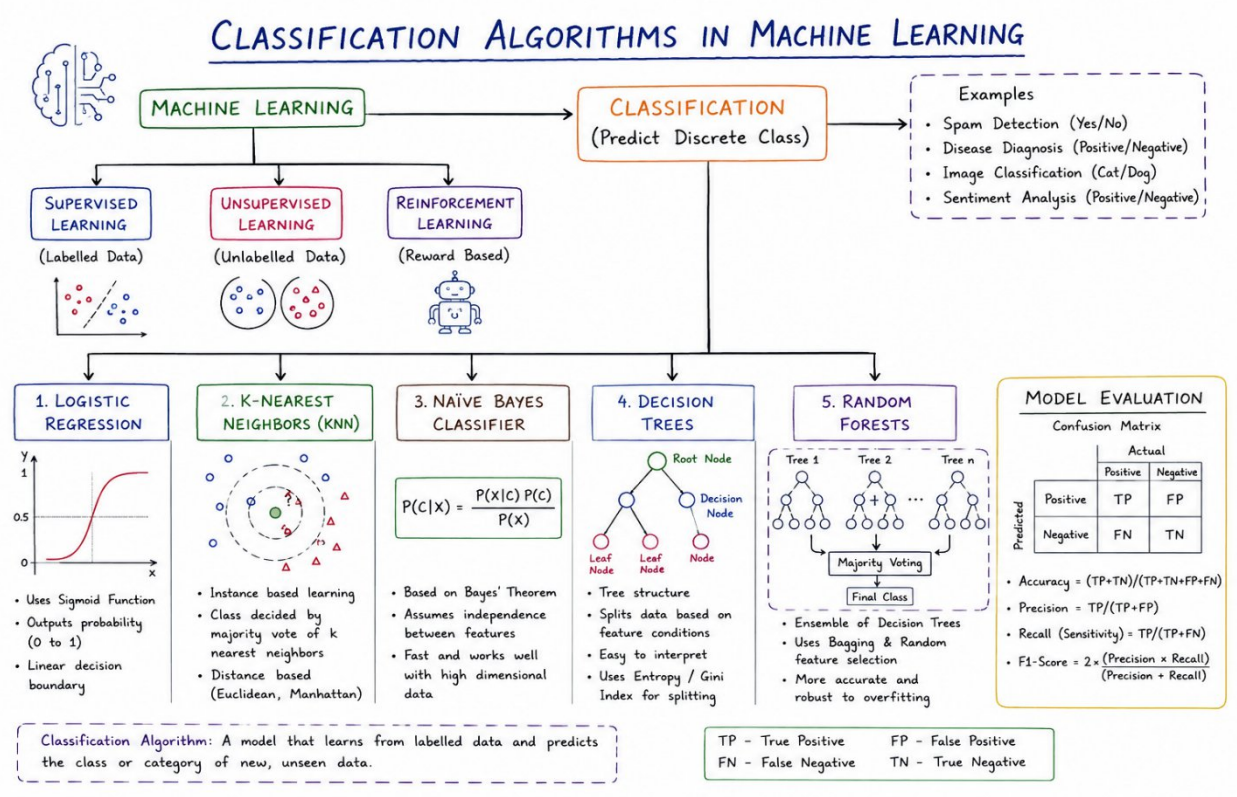
🎯 **Balanced Model**

"Understands the concept and predicts correctly."

➡ Best Model.

Unit 2 Classification Algorithms

Classification Overview and Decision Boundaries, Logistic Regression: Sigmoid Function and Cost, K Nearest Neighbors (KNN), Naïve Bayes Classifier, Decision Trees and Random Forests, Model Evaluation: Confusion Matrix, Precision, Recall, F1-Score.



1. Classification Overview

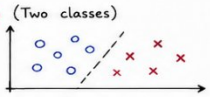
Classification is a Supervised Machine Learning technique used to predict a discrete (category/label) class for given input data.

Example Applications:
- Email → Spam / Not Spam
- Disease Diagnosis → Positive / Negative
- Image Classification → Cat / Dog
- Customer → Buy / Not Buy

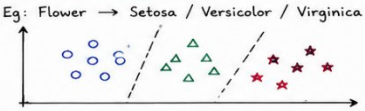
Output is a Category (not a number)

Types of Classification:

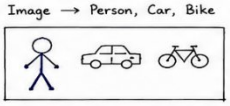
① Binary Classification (Two classes)
Eg: Pass / Fail



② Multi-class Classification (More than two classes)
Eg: Flower → Setosa / Versicolor / Virginica



③ Multi-label Classification (Multiple labels possible)
Eg: Image → Person, Car, Bike (more than one object)



Output:
- Person ✓
- Car ✓
- Bike ✓

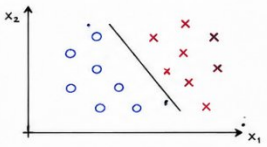
Key Points:
- Uses labeled training data.
- Learns patterns between input features and classes.
- Predicts the class label for unseen data.

2. Decision Boundaries

A Decision Boundary is a line (or curve or surface) that separates different classes in the feature space.

In 2-D Feature Space:

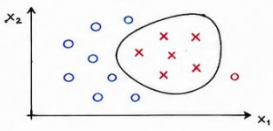
① Linear Decision Boundary



Equation of line: $w_1x_1 + w_2x_2 + b = 0$

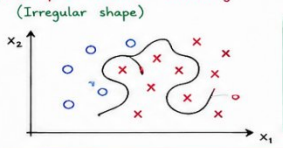
Used in:
- Logistic Regression
- Linear SVM

② Non-linear Decision Boundary (Curved boundary)



Used in:
- KNN
- Decision Trees
- SVM (with kernel)
- Neural Networks

③ Complex Decision Boundary (Irregular shape)



Why Important?
- Helps the model to separate classes correctly.
- Better boundary → Better accuracy.

Note:
If classes are well separated → model performs well.
If classes overlap → model may make errors.

Classification Overview

Definition:
Classification is a **Supervised Machine Learning** technique that predicts the **category (class or label)** of new data using previously labeled data.

(OR)

Classification means putting things into the correct group or category.
Just like humans classify objects every day, computers also learn to classify data.

Real-Life Example 1: Teacher Checking Exam Results
Imagine a teacher has students' marks.

Marks	Result
95	Pass
85	Pass
70	Pass
32	Fail
25	Fail

The teacher learns the rule:

- Marks ≥ 35 → Pass
- Marks < 35 → Fail

Now a new student scores **40 marks**.
The teacher predicts:
40 → Pass
This is **Classification** because the output is a **category**, not a number.

Real-Life Example 2: Email Inbox
Every day you receive emails.
The Gmail system automatically predicts

- Spam
- Not Spam

It has already learned from millions of emails.
This is Machine Learning Classification.

Real-Life Example 3: Hospital
Doctor enters

- Age
- Blood Pressure
- Sugar Level
- Temperature

Machine predicts

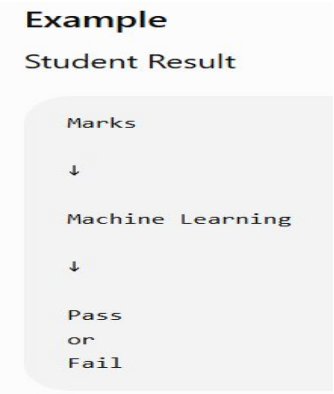
- Disease Present
- Disease Not Present

Again, this is Classification.

Characteristics of Classification:

- Uses **Labeled Data**
- Learns patterns from previous examples
- Predicts categories
- Output is not a number

Types of Classification:
1. Binary Classification
Binary means **only two classes**.



Only one of two possible outputs.

Examples:

- Yes / No
- Male / Female
- Spam / Not Spam
- Disease / No Disease

Example :Bank Loan

Customer applies for a loan.

Machine predicts

Loan Approved

or

Loan Rejected

Only two classes.

2. Multi-Class Classification

Here there are **more than two classes**.

Example1

Flower Identification

Machine predicts

- Rose
- Lotus
- Sunflower

Only one class is selected.

Example2

Handwritten Digit Recognition

Image

↓

Machine predicts

0

or

1

or

2

...

or

9

Ten possible classes.

3. Multi-Label Classification:

One object can belong to **multiple classes at the same time**.

Example

Suppose an image contains

🧒 Person

🚗 Car

🚲 Bicycle

The machine predicts

✓ Person

✓ Car

✓ Bicycle

All labels are correct simultaneously.

Example

Facebook Photo Tagging

When you upload a group photo,

Facebook detects

- Person A
- Person B
- Person C

One image contains many labels.

2. Decision Boundaries

A **Decision Boundary** is a **line, curve, or surface** that separates different classes.

(OR)

Think of it as a **boundary or divider** that tells the machine:

"Everything on this side belongs to Class A, and everything on the other side belongs to Class B."

Example 1: School LIFE

IF U CAN TAKE boys stand on one side of the playground and girls stand on the other.

Girls | Boys

○ ○ ○ ○ | × × × ×

---------- Boundary ----------

The line in the middle separates both groups.

This line is called the **Decision Boundary**.

Example 2: Pass or Fail

Suppose WE CAN TAKE A STUDENTS passing marks are **35**.

0----------35----------100

Fail Boundary Pass

The mark **35** acts as the Decision Boundary.

- Marks below 35 → Fail
- Marks 35 or above → Pass

Types of Decision Boundaries:

A **Decision Boundary** is a **line, curve, or surface** that separates different classes (categories) in a classification problem. It helps the machine decide which class a new data point belongs to.

1. Linear Decision Boundary

Definition: A **Linear Decision Boundary** is a **straight line** (in 2D) or a **flat plane** (in 3D) that separates two classes.

It is used when the data can be separated using a straight line.

Class A (○)

○ ○ ○

---------- ← **Decision Boundary**

× × ×

Class B (×)

The straight line separates the two classes.

(OR)

Linear Boundary 🚶

Think of a straight road dividing two villages.

Village A | Village B

Example 1: Pass or Fail

A college decides that students scoring **35 or more marks** will pass.

Marks	Result
20	Fail
25	Fail
40	Pass

Marks	Result
55	Pass
80	Pass

0----------35----------100

Fail Boundary Pass

Example 2: Loan Approval

A bank approves loans based only on salary.

- Salary ≥ ₹50,000 → Loan Approved
- Salary < ₹50,000 → Loan Rejected

Salary

Rejected | Approved

---------|---------

₹50,000

A straight line separates approved and rejected customers.

Algorithms Using Linear Decision Boundary

- Logistic Regression
- Linear Support Vector Machine (Linear SVM)
- Perceptron

Advantages

- Simple
- Fast
- Easy to understand
- Works well for simple datasets

Disadvantages

- Cannot separate complex data
- Low accuracy for non-linear problems

2. Non-Linear Decision Boundary

Definition

A **Non-Linear Decision Boundary** is a **curved line** that separates data.

It is used when a straight line cannot separate the classes correctly.

A curved boundary surrounds one class and separates it from the other.

(OR)

Non-Linear Boundary 🌊

Think of a **river** separating two villages.

~~~~ River ~~~~

Example 1: Disease Prediction

A doctor predicts diabetes based on:

- Age
- Weight
- Blood Sugar
- Blood Pressure

A patient cannot be classified using only one straight rule because all these features work together.

The machine creates a **curved decision boundary**.

Example 2: Fruit Classification

Suppose we classify fruits using:

- Weight
- Color

A fruit may be:

- Apple
- Orange
~~~~

The data points overlap, so a straight line cannot separate them.

A curved boundary provides better classification.

Algorithms Using Non-Linear Decision Boundary

- K-Nearest Neighbors (KNN)
- Decision Trees
- Kernel SVM
- Neural Networks

Advantages

- Higher accuracy
- Handles complex datasets
- Better for real-world problems

Disadvantages

- More computation
- Harder to understand

3. Complex Decision Boundary

Definition

A **Complex Decision Boundary** has an **irregular shape**. It is used when the data has highly complex patterns that cannot be separated by a simple line or curve.**Diagram**

○ ○

× × ×

○ × ○ × ○

× ○ ×

The boundary becomes irregular to separate all classes correctly.

(OR)

Complex Boundary

Think of a **state border on a map** with many twists and turns. /\/\/\/\/\

Example 1: Face Recognition

Your smartphone unlocks only for your face.

It checks:

- Eye distance
- Nose shape
- Face shape
- Smile
- Chin
- Forehead

These features are very complex, so the model creates an irregular boundary. **Example 2: Self-Driving Cars**

A self-driving car identifies:

- Pedestrians
- Cars
- Bikes
- Traffic Signs
- Animals

Since the objects have different shapes, colors, and sizes, the machine uses a **complex decision boundary**.

Algorithms Using Complex Decision Boundary

- Random Forest
- Deep Neural Networks
- Convolutional Neural Networks (CNN)
- Gradient Boosting Models

Advantages

- Very high accuracy
- Learns highly complex patterns
- Suitable for large datasets

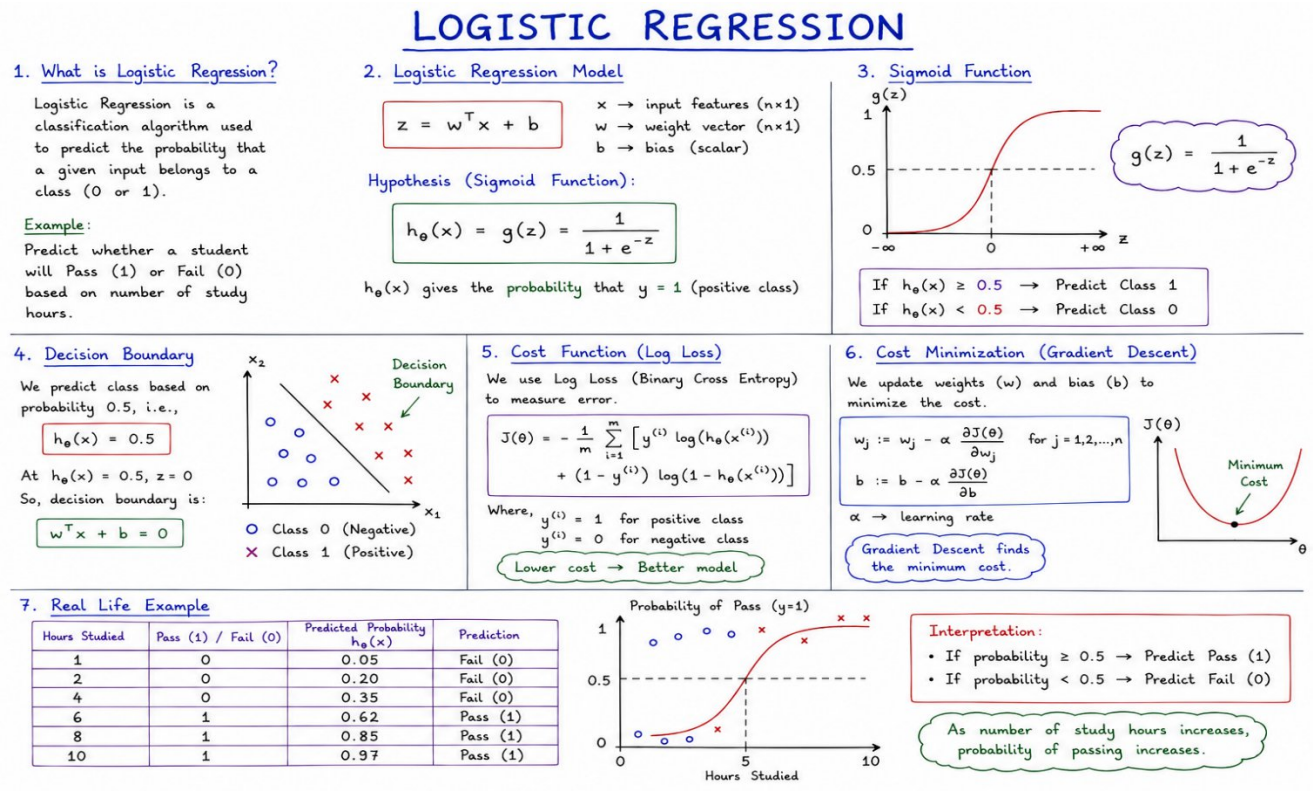
Disadvantages

- Requires more training time
- Higher computational cost
- More difficult to interpret

Type	Shape	Best For	Algorithms	Real-Life Example
Linear Decision Boundary	Straight Line	Simple data	Logistic Regression, Linear SVM	Pass/Fail, Loan Approval

Type	Shape	Best For	Algorithms	Real-Life Example
Non-Linear Decision Boundary	Curved Line	Moderately complex data	KNN, Decision Tree, Kernel SVM	Disease Prediction, Fruit Classification
Complex Decision Boundary	Irregular Shape	Highly complex data	Random Forest, Neural Networks, CNN	Face Recognition, Self-Driving Cars

Logistic Regression:



1. What is Logistic Regression?

Definition

Logistic Regression is a **Supervised Machine Learning Classification Algorithm** used to predict **categorical outputs**, mainly **Yes/No** or **0/1**.

Important Points

- Used for **Binary Classification**
- Predicts **Probability (0 to 1)**
- Uses the **Sigmoid Function**
- Output is a **class**, not a continuous value

Real-Life Example

Student Exam Prediction

Study Hours	Prediction
2 Hours	Fail (0)
8 Hours	Pass (1)

Easy Memory Tip

Logistic Regression = **Probability + Classification**

2. Logistic Regression Model

Formula

$$z = w^T x + b$$

Where

- **x** → Input Features
- **w** → Weights
- **b** → Bias
- **z** → Linear Combination

The model first calculates **z**, then sends it to the **Sigmoid Function**.

Important Point

Model learns the best weights to make correct predictions.

Example

Predicting whether a customer will buy a product using:

- Age
- Salary
- Shopping History

3. Sigmoid Function:

Formula

$$g(z) = \frac{1}{1 + e^{-z}}$$

The **Sigmoid Function** converts any real number into a probability between **0 and 1**.

Output Range

0 ---------- 1

Prediction Rule

- Probability ≥ 0.5 → Class 1
- Probability < 0.5 → Class 0

Example

Hospital Disease Prediction

Probability = **0.90**

↓

Disease = **Yes**

Probability = **0.20**

↓

Disease = **No**

Important Point

Sigmoid converts linear output into probability.

4. Decision Boundary

Definition

A **Decision Boundary** separates two different classes.

It is the point where

$$h_\theta(x) = 0.5$$

Simple Explanation

Everything on one side belongs to **Class 0**.

Everything on the other side belongs to **Class 1**.

Real-Life Example

Pass Marks = **35**

0------35------100

Fail Pass

35 is the **Decision Boundary**.

Important Point

Decision Boundary divides the data into two groups.

5. Cost Function (Log Loss)

Definition

The **Cost Function** measures **how wrong** the predictions are.

Higher Cost = More Errors

Lower Cost = Better Model

Purpose

Find the best weights by reducing prediction errors.

Real-Life Example

Suppose

Actual Result = Pass

Model Prediction = Fail

↓

Large Error

↓

High Cost

If prediction is correct,

↓

Small Error

↓

Low Cost

Important Point

Goal = Minimize Cost

6. Cost Minimization (Gradient Descent)

Definition

Gradient Descent is an optimization algorithm that updates the weights and bias to reduce the Cost Function.

Steps

1. Calculate Cost
2. Update Weights
3. Reduce Error
4. Repeat Until Minimum Cost

Example

Imagine throwing a ball from the top of a hill.

The ball naturally rolls downward until it reaches the **lowest point**.

Similarly,

Gradient Descent reduces the cost until it reaches the **minimum value**.

Important Point

Gradient Descent = Finds the Best Model by Minimizing Cost

7. Example

Suppose we predict whether a student will **Pass or Fail**.

Study Hours	Probability	Prediction
1 Hour	0.05	Fail
2 Hours	0.20	Fail
4 Hours	0.35	Fail
6 Hours	0.62	Pass
8 Hours	0.85	Pass
10 Hours	0.97	Pass

Observation

- More Study Hours ↑
- Probability of Passing ↑

Important Point

As input increases, the probability of belonging to the positive class also increases.

Topic	Points
Logistic Regression	Predicts categories (Yes/No, Pass/Fail)
Logistic Model	Calculates $z = w^T x + b$
Sigmoid Function	Converts output into probability (0–1)
Decision Boundary	Separates Class 0 and Class 1
Cost Function	Measures prediction error
Gradient Descent	Reduces cost and improves the model
Real-Life Example	More study hours → Higher probability of passing

Exam point view :

Logistic Regression is a supervised machine learning algorithm used for **binary classification**. It uses the **Sigmoid Function** to convert the linear output into a probability between **0 and 1**. The **Cost Function** measures prediction error, and **Gradient Descent** minimizes the cost by updating the model parameters, resulting in accurate classification.

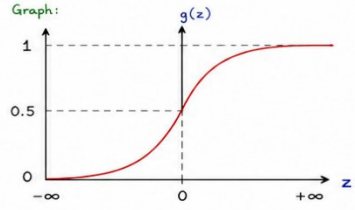
Sigmoid Function and Cost

1. Sigmoid Function

The Sigmoid function converts any real value into a probability between 0 and 1.

$$\sigma(z) = g(z) = \frac{1}{1 + e^{-z}}$$

Graph:



Properties:

- Range: (0, 1)
- $g(0) = 0.5$
- As $z \to +\infty$, $g(z) \to 1$
- As $z \to -\infty$, $g(z) \to 0$
- S-shaped curve

Interpretation:

$g(z)$ gives the probability that $y = 1$ (positive class)

$h(z) = g(z) = P(y = 1 \mid ; x; \theta)$

Example:

z	$g(z) = \frac{1}{1+e^{-z}}$	Interpretation
−4	0.018	Very small probability (Class 0)
−2	0.119	More towards Class 0
0	0.500	Boundary (Equal chance)
2	0.881	More towards Class 1
4	0.982	Very high probability (Class 1)

Decision Rule:

If $g(z) \geq 0.5$

Predict Class 1

else

Predict Class 0

2. Cost (Loss) Function

Cost function measures how wrong our prediction is.
For Logistic Regression, we use Log Loss (Binary Cross Entropy).

For one training example $(x^{(i)}, y^{(i)})$:

$$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = -\left[y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

Where,

$y^{(i)} = 1$ for positive class

$y^{(i)} = 0$ for negative class

$h_\theta(x^{(i)}) = g(z^{(i)})$ = predicted probability

Intuition:

- If $y = 1$ and $h(x) \to 1$, cost $\to 0$
- If $y = 1$ and $h(x) \to 0$, cost $\to \infty$
- If $y = 0$ and $h(x) \to 0$, cost $\to 0$
- If $y = 0$ and $h(x) \to 1$, cost $\to \infty$

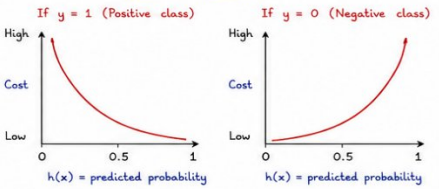
For m training examples:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

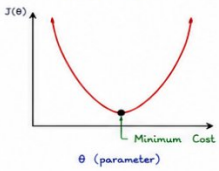
Goal:

Find θ (weights and bias) that minimize $J(\theta)$ (minimum cost).

Cost vs Prediction



J(θ) Curve



Summary: Sigmoid function maps any value into probability (0 to 1). Cost function (Log Loss) penalizes confident wrong predictions heavily. We use Gradient Descent to minimize the cost and get the best model.

Sigmoid function

→ The Sigmoid function (also called the Logistic function) converts any real numbers into a value between 0 and 1.

→ This value is interpreted as the probability that a sample belongs to the positive class

Formula:

$$y = \frac{1}{1+e^{-x}}$$

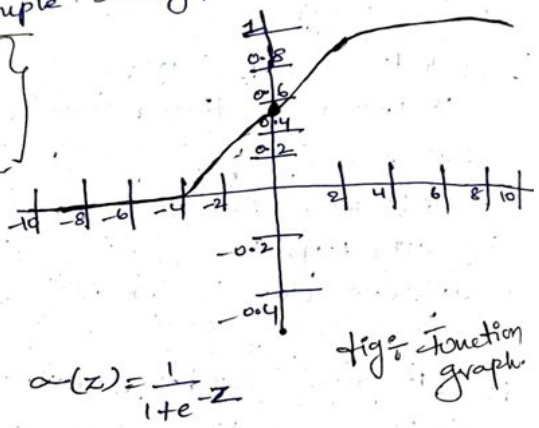


fig: function graph

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

where

→ $\sigma(z)$ = predicted probability

→ $z = wx + b$

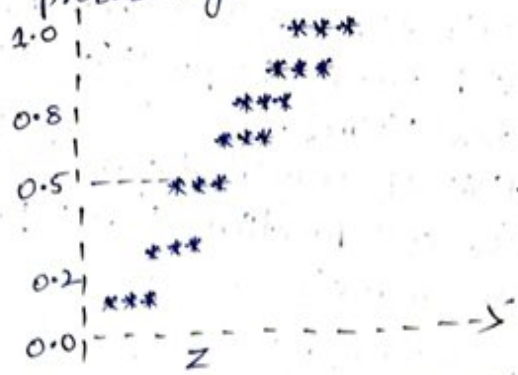
⇒ w = weight

→ X = input feature

→ b = Bias.

→ e = Euler's Constant (≈ 2.718)

Sigmoid Curve :-



properties of the Sigmoid function :-

→ output always lies between 0 and 1.
→ produces an S-shaped (sigmoid) curve.
→ converts linear output into probability.
→ smooth and differentiable.
→ used mainly in binary classification.

Decision Rule :-

After calculating the probability;
→ If probability ≥ 0.5 → class = 1 (+ve)
→ If probability < 0.5 → class = 0 (-ve)

Ex :- probability — predicted class

0.92 ⟶ +ve
0.75 ⟶ +ve
0.48 ⟶ -ve
0.10 ⟶ -ve

→ Helps Gradient Descent find the best model parameters.

Steps in logistic Regression:

→ collect labeled training data.
→ Compute the linear equation $Z = WX + b$.
→ Apply the Sigmoid-function to obtain probabilities
→ Calculate the cost-function (Log Loss)
→ Update weights using Gradient Descent.
→ Repeat until the cost is minimized
→ use the trained model to classify new da

Advantages of Logistic Regression:

→ Simple and easy to implement.
→ Fast training and prediction.
→ produces probability outputs.
→ Easy to intrepret.
→ Works label for binary classifications.

Limitations of Logistic Regression:

→ Assumes a linear decision boundary.
→ performs poorly on highly complex & non-linear datasets.
→ Sensitive to outliers
→ Requires sufficient labeled training data

Cost function in Logistic Regression:

→ The cost function measures how much error exists between the predicted values and the actual values.

→ The objective of Logistic Regression is to minimize the cost function so that predicting between more accurate.

→ Logistic Regression uses Log Loss (Cross-Entropy Loss).

Logistic Regression cost function (Log Loss):

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y \log(h_\theta(x)) + (1-y) \log(1-h_\theta(x))].$$

Where

$J(\theta)$ = cost function

m = Number of training examples

y = Actual class label (0 or 1)

$h\theta(x)$ = predicted probability

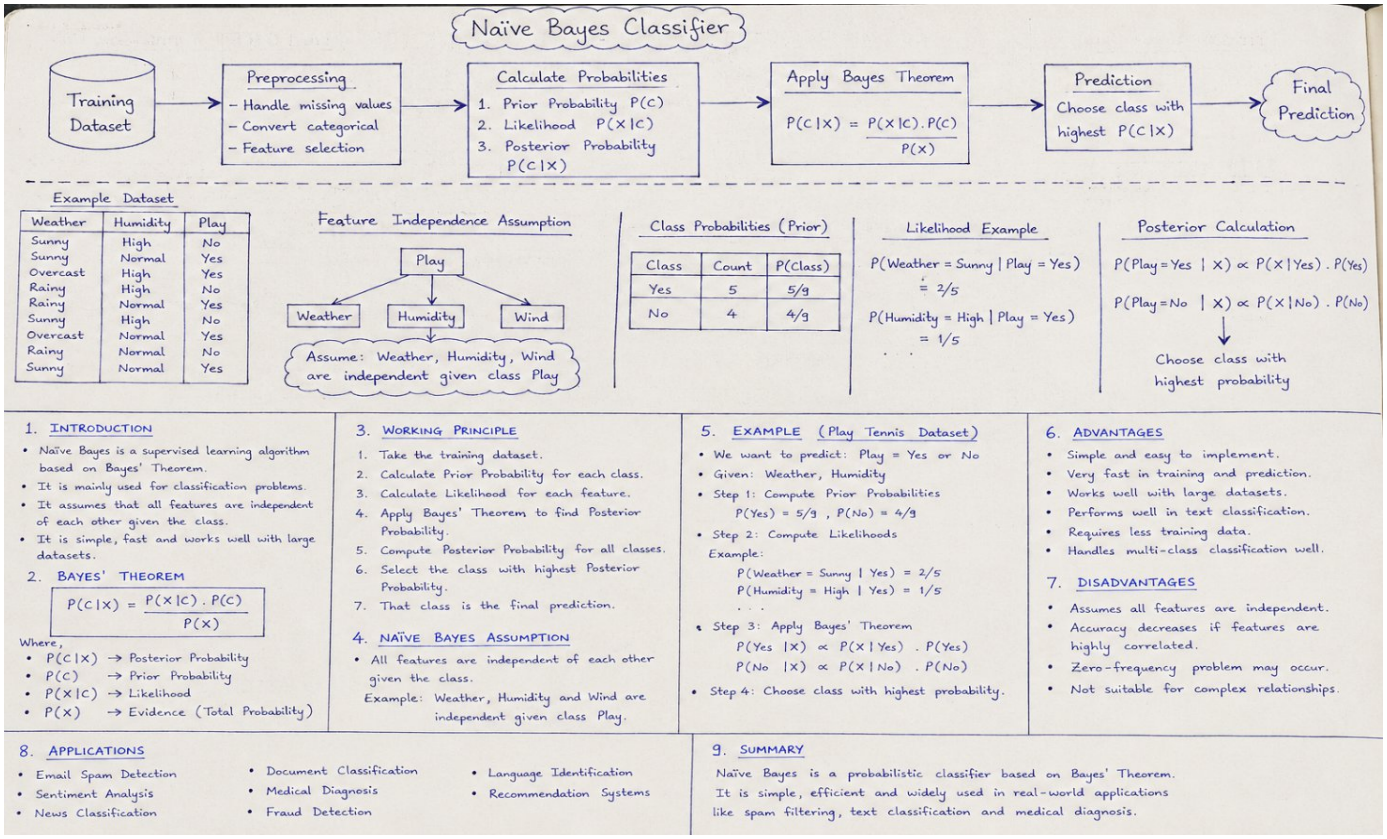
log = Natural algorithm.

Why use Log uses?

→ Give low cost when predictions are correct.

→ Give high cost " " " are incorrect.

→ provides a smooth function for efficient optimization.



1. INTRODUCTION

What is Naïve Bayes?

Naïve Bayes is a **supervised machine learning algorithm** used for **classification**. It is based on **Bayes' Theorem**, which calculates probability.It predicts the class that has the **highest probability**.

(OR)

Naïve Bayes is a probability-based classification algorithm that uses Bayes' Theorem and assumes all features are independent of each other.

Why is it called "Naïve"?

It is called **Naïve** because it assumes that every feature is independent.

Example: Suppose we want to predict whether a student will pass.

Features:

- Study Hours
- Attendance
- Sleep Hours

Naïve Bayes assumes

Study Hours

↑

Attendance

↑

Sleep Hours

All are independent.

But in real life they may be related.

Where is it used?

- Email Spam Detection
- Text Classification
- Medical Diagnosis
- Sentiment Analysis

Key Points

✔ Supervised Learning

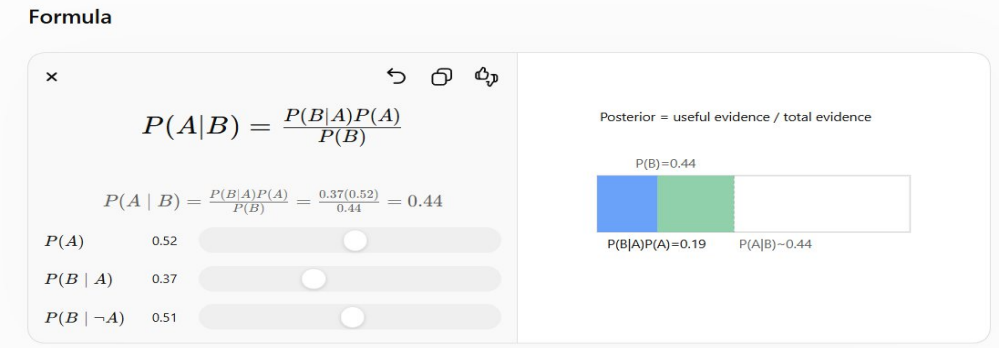
✔ Classification Algorithm

✔ Probability Based

✔ Fast Algorithm

2. Bayes' Theorem

Naïve Bayes works using **Bayes' Theorem**.



Symbol	Meaning
(P(C	X))
(P(X	C))
$P(C)$	Prior Probability
$P(X)$	Evidence

Example (Play Tennis Dataset)

Suppose we want to predict whether a person will **Play Tennis = Yes**.

Given:

- **Weather = Sunny**
- **Humidity = High**

From the training data:

```
P(Play = Yes) = 5/9
```

```
P(Weather = Sunny | Play = Yes) = 2/5
```

```
P(Humidity = High | Play = Yes) = 1/5
```

Assume:

```
P(X) = 0.10
```

Substitute the values : $P(\text{Yes} \mid X) = \frac{\left(\frac{2}{5} \times \frac{1}{5}\right) \times \frac{5}{9}}{0.10}$

Step-by-Step Calculation : $\frac{\left(\frac{2}{25}\right) \times \frac{5}{9}}{0.10} = \frac{\frac{10}{225}}{0.10} = \frac{0.0444}{0.10} = 0.444$

So,

```
P(Play = Yes | Sunny, High)
= 0.444
```

Similarly, calculate **P(Play = No | Sunny, High)**.

Suppose,

```
P(Play = No | Sunny, High)
= 0.256
```

Since

```
0.444 > 0.256
```

Final Prediction

```
Play Tennis = YES
```

3. WORKING PRINCIPLE

Naïve Bayes predicts the class in **6 simple steps**.

Step 1

Take the training dataset.

Example

Weather	Humidity	Play
Sunny	High	No
Sunny	Normal	Yes
Rainy	High	No

Step 2

Calculate Prior Probability.

Example

Play=Yes = 5

Play=No =4

Total=9

P(Yes)=5/9

P(No)=4/9

Step 3: Calculate Likelihood.

Example

P(Sunny|Yes)=2/5

P(High|Yes)=1/5

Step 4: Apply Bayes Formula.

$P(Yes|X) = P(X|Yes) \times P(Yes) \; P(X)$

Step 5

Calculate Posterior Probability.

Suppose

P(Yes|X)=0.75

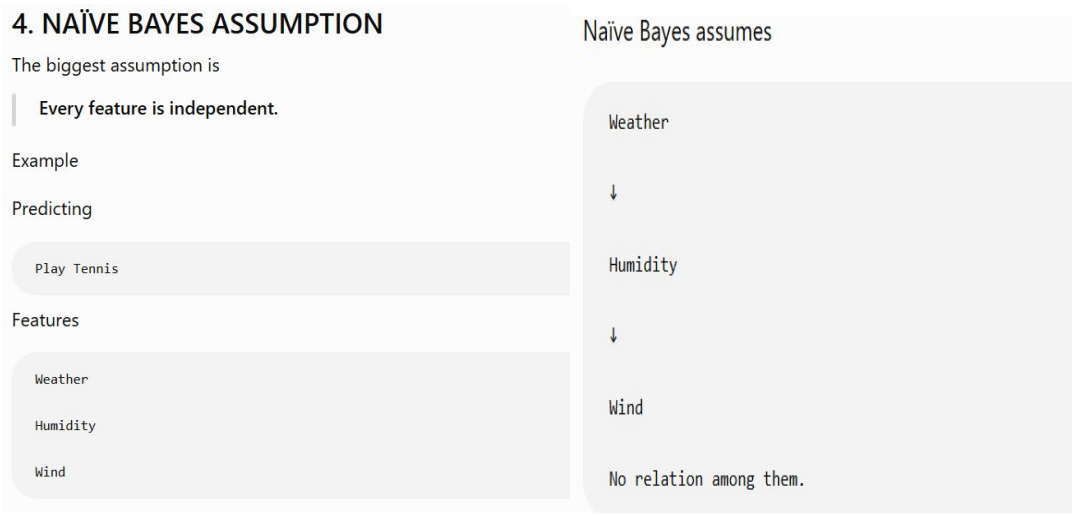
P(No|X)=0.25

Step 6

Choose highest probability.

Highest=0.75

Prediction : Play=Yes



Naïve Bayes Independence Assumption

Feature 1	**Feature 2**	**Naïve Bayes Assumption**
CGPA	**Communication**	CGPA does **not** affect Communication.
Programming	**CGPA**	Programming does **not** affect CGPA.
Communication	**Programming**	Communication does **not** affect Programming.

Visual Table

Feature	**Independent Of**	**Assumption**
CGPA	Communication	No relationship
CGPA	Programming	No relationship
Communication	Programming	No relationship

Example Dataset

Student	CGPA	Communication	Programming	Placement
A	High	Good	Good	Yes
B	Low	Poor	Good	No
C	Medium	Good	Average	Yes

5. EXAMPLE (Play Tennis Dataset)

Suppose we want to predict

```
Play Tennis

Yes or No
```

Input

```
Weather=Sunny

Humidity=High
```

Step 1

Prior

```
P(Yes)=5/9

P(No)=4/9
```

Step 2

Likelihood

```
P(Sunny|Yes)=2/5

P(High|Yes)=1/5
```

Step 3

Apply Bayes Formula

```
P(Yes|X)

=

2/5 ×1/5 ×5/9
```

Similarly

Calculate No.

Step 4

Suppose

```
P(Yes)=0.72

P(No)=0.28
```

Prediction

```
Play Tennis

YES
```

Conclusion:

- Naïve Bayes is a probability-based supervised classification algorithm that uses Bayes' Theorem to predict the output class.
- It assumes that all features are independent of each other and selects the class with the highest posterior probability.
- It is simple, fast, memory-efficient, and suitable for large datasets.
- Naïve Bayes is widely used in spam filtering, text classification, sentiment analysis, medical diagnosis, and fraud detection.

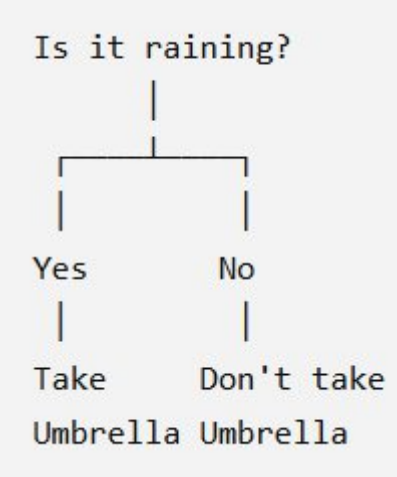
Decision Tree

A **Decision Tree** is a **Supervised Machine Learning algorithm** used for **Classification** and **Regression**.

It makes decisions by asking a sequence of **questions**, just like a flowchart.

Example

Suppose you want to decide whether to carry an umbrella.



This is exactly how a Decision Tree works.

Common Classification Algorithms

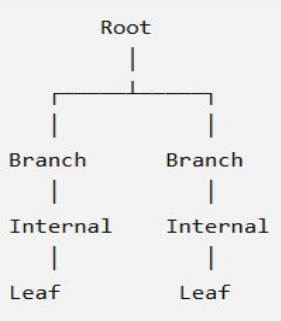
Your notes list these algorithms:

1. K-Nearest Neighbors (KNN)
2. Decision Tree
3. Random Forest
4. Support Vector Machine (SVM)
5. Naïve Bayes

Decision Tree is one of the most popular algorithms because it is simple and easy to understand.

Why is it called a Decision Tree?

It looks like an upside-down tree.



Each question creates a branch until the final answer is reached.

Goal of Decision Tree:

The goal is:

To predict the output class using input features.

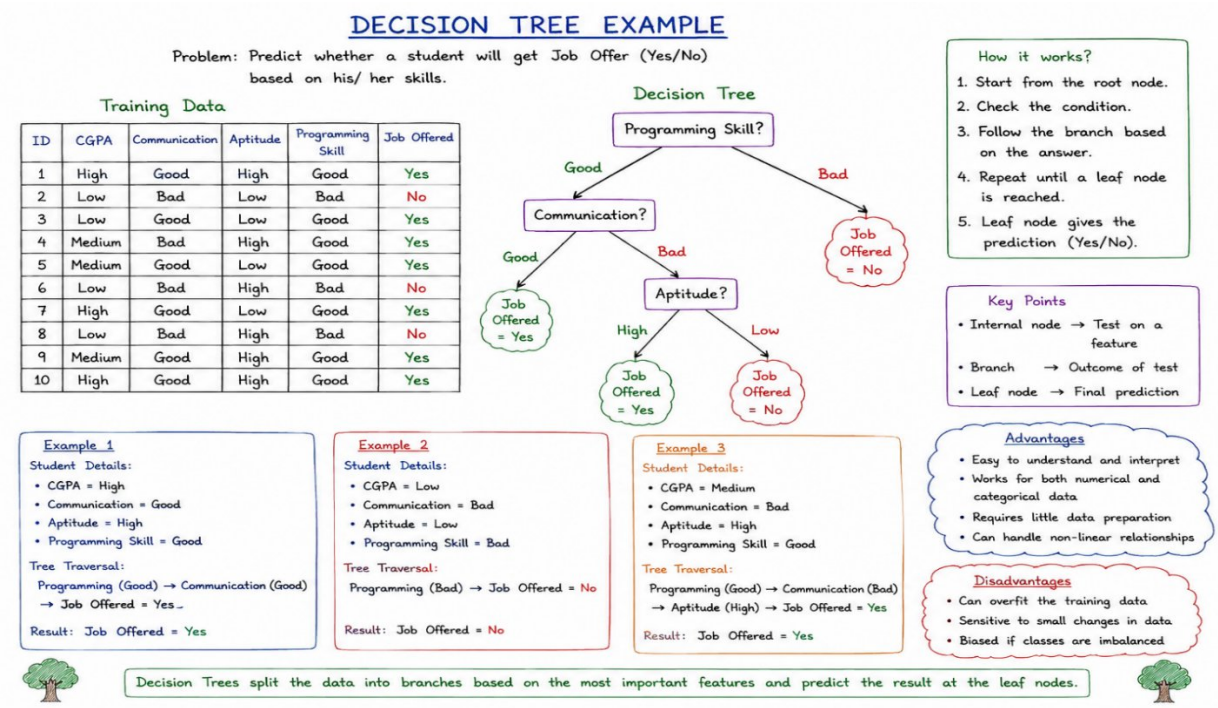
Example

Input

- CGPA
- Communication
- Aptitude
- Programming Skill

Output

- Job Offered = Yes/No



1. Training Data

The left table is called the **Training Dataset**.

It contains previously known student information.

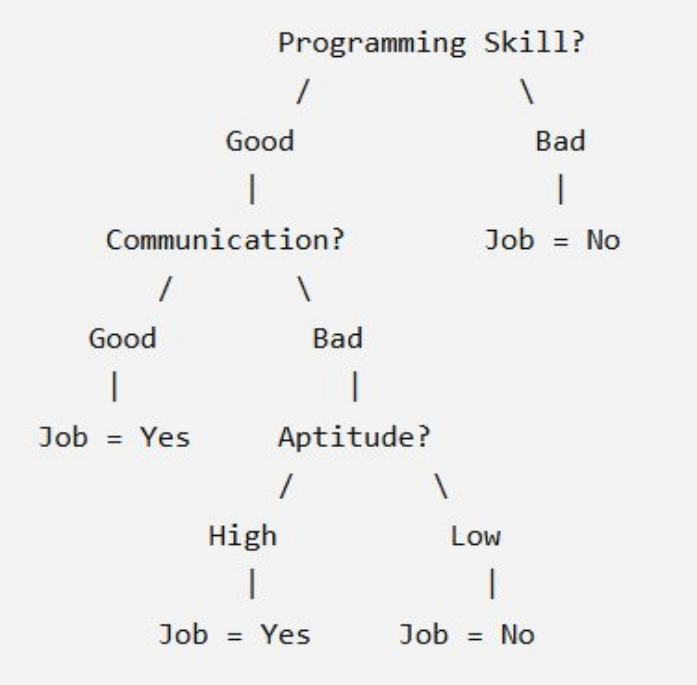
Feature	Meaning
CGPA	Academic Performance
Communication	Speaking Skills
Aptitude	Problem Solving Ability
Programming Skill	Coding Skill
Job Offered	Target Class (Yes/No)

Example

CGPA	Communication	Aptitude	Programming	Job
High	Good	High	Good	Yes
Low	Bad	Low	Bad	No

2. Decision Tree Structure

The middle figure shows the Decision Tree.



Every box asks a question.

Every branch represents an answer.

The circles at the bottom give the final prediction.

Types of Nodes

1. Root Node : The first and main node.

Example

Programming Skill?

Every prediction starts from here.

2. Internal Node :A node that asks another question.

Example

Communication?

3. Branch : The answer to a question.

Example

Good

Bad

High

Low

4. Leaf Node : The final prediction.

Example

Job = Yes

Job = No

Job = No

How a Decision Tree is Built

The tree is built using the training data.

Step 1 : Collect the training dataset.

Example

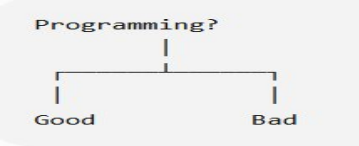
Student	Programming	Communication	Job
A	Good	Good	Yes
B	Bad	Bad	No
C	Good	Bad	Yes

Step 2 : Choose the best feature.

Example

Programming Skill

This becomes the Root Node.**Step 3 :**Split the dataset.



Step 4 : Repeat for every branch.

Programming?

|

Communication?

|

Aptitude?

Step 5 : Stop when all records belong to one class. Then create a Leaf Node.

Searching in a Decision Tree:

Searching in a Decision Tree means **finding the correct class (prediction) for a new data record** by starting from the **root node** and following the appropriate branches until a **leaf node** is reached. (OR)
Searching in a Decision Tree is the process of moving from the **root node** to a **leaf node** by checking feature values one by one to make a prediction.

Advantages of Searching in a Decision Tree

- Simple and easy to understand.
- Fast prediction.
- Follows logical if–else rules.
- Suitable for both numerical and categorical data.

Entropy in Decision Tree

Entropy is a measure of impurity, uncertainty, or disorder in a dataset.

It tells us how mixed the classes are in the data.

- Low Entropy → Data is pure (mostly one class).
- High Entropy → Data is mixed (different classes).

(OR)

Entropy is a measure of the impurity or randomness of a dataset. It is used by Decision Trees to choose the best feature for splitting the data

Why do we use Entropy?

A Decision Tree tries to split the data into **pure groups**.

Entropy helps the algorithm answer:

"Which feature separates the data best?"

The feature that reduces entropy the most is selected.

Entropy Formula

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Where:

- S = Dataset
- c = Number of classes
- p_i = Probability of class i

Meaning of the Formula

Entropy is calculated by:

1. Find the probability of each class.
2. Multiply it by $\log_2$of the probability.
3. Add all values.
4. Change the sign to positive.

Example 1 (Pure Dataset)

Suppose we have 10 students.

Job Offered	Count
Yes	10
No	0

Probability:

- Yes = 10/10 = 1
- No = 0/10 = 0

Entropy:

$$Entropy = 0$$

Interpretation

All students belong to one class.

There is **no confusion**.

This is called a **Pure Dataset**.

Example 2 (Mixed Dataset)

Suppose

Job Offered	Count
Yes	5
No	5

Probability

Yes = 0.5

No = 0.5

Entropy

$$Entropy = -\left(0.5\log_2 0.5 + 0.5\log_2 0.5\right)$$

Entropy = **1**

Interpretation

Both classes are equally mixed.

Maximum uncertainty.

Example 3 (Your Notes)

Original Dataset

Class	Count
Yes	8
No	10
Total	18

Probability

Yes $P(Yes) = \frac{8}{18} = 0.44$

No $P(No) = \frac{10}{18} = 0.56$

Entropy

$$Entropy(S) = -\left(0.44\log_2 0.44 + 0.56\log_2 0.56\right)$$

Entropy ≈ **0.99**

Interpretation

The dataset is mixed because both Yes and No classes are present.

Information Gain :

What is Information Gain?

Information Gain tells us **how much uncertainty is reduced after splitting the data**.

The Decision Tree selects the feature with the **highest Information Gain**.

Formula : $\boxed{IG(S,A) = Entropy(S) - Entropy(S \mid A)}$

Where

- **Entropy(S)** = Entropy before splitting.
- **Entropy(S|A)** = Entropy after splitting using attribute A.

Easy Meaning

Suppose

Original Entropy = 0.99

After splitting on CGPA

Entropy = 0.30

Information Gain

$$IG = 0.99 - 0.30 = 0.69$$

Interpretation

CGPA reduces uncertainty by **0.69**, so it is a good feature for splitting.

Examples

Original Dataset

Yes	No
8	10

Entropy = **0.99**Split by **CGPA**

High

Yes	No
4	2

Entropy ≈ **0.9Medium**

Yes	No
4	3

Entropy ≈ **0.99Low**

Yes	No
0	5

Entropy = **0**Weighted Entropy after split ≈ **0.69**

Information Gain

$$0.99 - 0.69 = 0.30$$

This means **CGPA** helps reduce uncertainty, so it is a useful feature.

Pruning

Sometimes a Decision Tree becomes too large and memorizes the training data.

This is called **Overfitting**.

To avoid this, we use **Pruning**.

Types of Pruning

1. Pre-Pruning

Stop the tree from growing too much.

Example: Limit the maximum tree depth to 3 levels.**2. Post-Pruning**

Allow the complete tree to grow first.

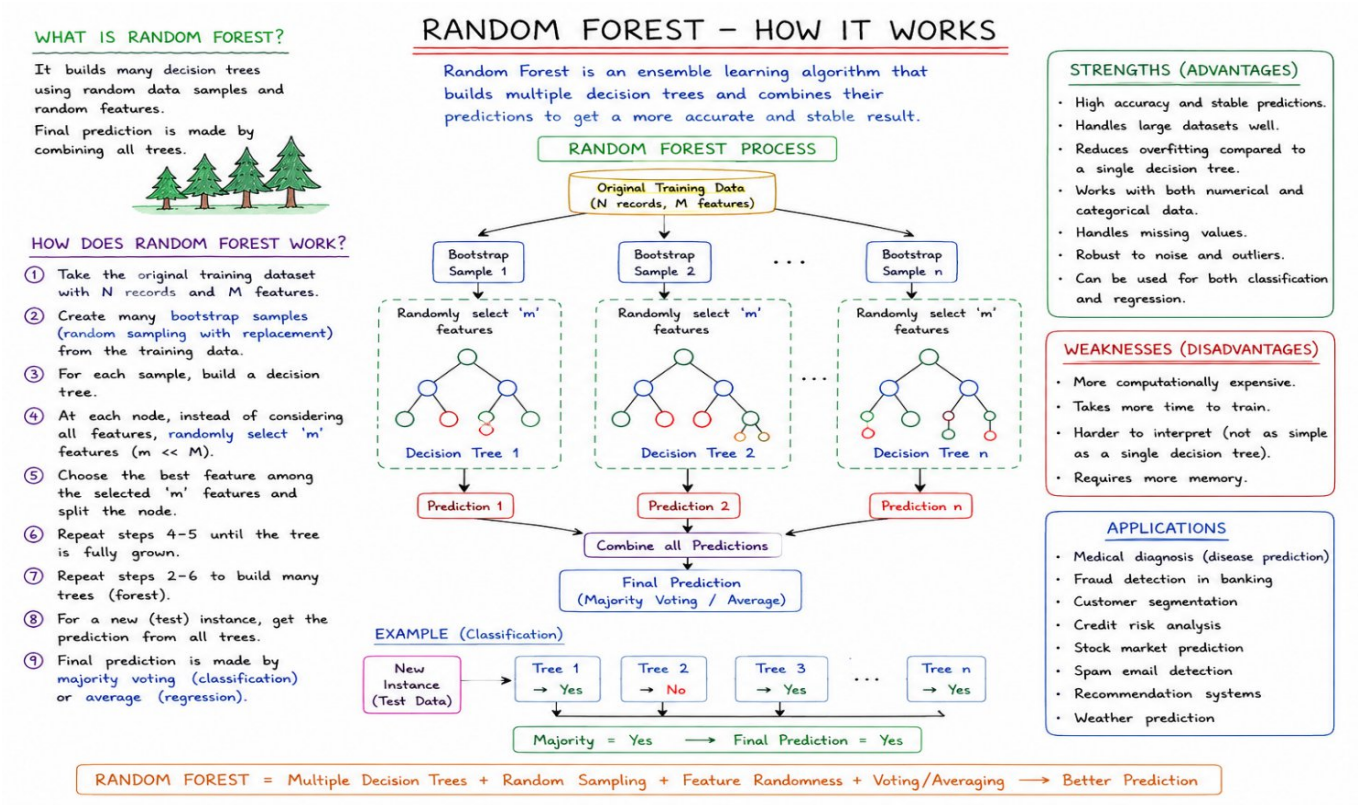
Then remove unnecessary branches that do not improve accuracy.

Advantages of Entropy

- Helps find the best feature for splitting.
- Reduces uncertainty in the dataset.
- Improves Decision Tree accuracy.
- Produces better classification results.

Disadvantages

- Requires repeated calculations.
- Can be computationally expensive for very large datasets.
- Sensitive to noisy or incorrect data.



RANDOM FOREST

Definition

Random Forest is a **supervised machine learning algorithm** used for **classification and regression**. It is an **ensemble learning method** that combines the predictions of multiple Decision Trees to produce a more accurate and stable result.

Idea:
Random Forest = Many Decision Trees + Majority Voting (Classification) / Average (Regression)

How Random Forest Works

Step 1: Prepare the Training Dataset

The algorithm starts with the original training dataset containing all records and features.

Example:

Student	CGPA	Programming	Job
A	High	Good	Yes
B	Low	Bad	No
C	Medium	Good	Yes

Step 2: Create Bootstrap Samples

Random Forest creates several **bootstrap samples** by randomly selecting records from the original dataset **with replacement**.

Example

Original Data:

- Every student appears **once**.
- No duplicate records.
- No missing records.

A B C D E

Bootstrap Sample:

- **C** appears **twice** (duplicate).
- **B** is **missing**.
- This is called **Sampling with Replacement**.

A C C D E

Step 3: Build Multiple Decision Trees

Each bootstrap sample is used to build one Decision Tree.

Sample 1 → Tree 1

Sample 2 → Tree 2

Sample 3 → Tree 3

Many trees together form the **Random Forest**.**Step 4: Random Feature Selection**

While building each tree, the algorithm selects only a **random subset of features** at every split.

Decision Tree	**Randomly Selected Features**
Tree-1	CGPA, Programming
Tree-2	Communication, Aptitude
Tree-3	CGPA, Communication
Tree-4	Programming, Aptitude
Tree-5	Communication, Programming

This increases diversity among trees.**Step 5: Prediction by Each Tree**

Suppose a new student is given.

CGPA = High

Programming = Good

Each tree predicts independently.

Tree 1 → Yes

Tree 2 → No

Tree 3 → Yes

Tree 4 → Yes**Step 6: Final Prediction**

For **Classification**

Use **Majority Voting**

Yes = 3

No = 1

Final Prediction = Yes

For **Regression**

Take the average of all predictions.

Example

Tree1 = 75

Tree2 = 80

Tree3 = 78

Average = 77.67

Random Forest Flow chart

Step No.	Process	Description
1	**Training Dataset**	Collect the original training data containing all records and features.
2	**Bootstrap Sampling**	Create multiple random datasets by sampling the training data

Step No.	Process	Description
		with replacement.
3	**Build Multiple Decision Trees**	Train one Decision Tree using each bootstrap sample.
4	**Random Feature Selection**	At each node, randomly select a subset of features to split the data.
5	**Each Tree Makes Prediction**	Every Decision Tree independently predicts the class (or value).
6	**Majority Voting / Average**	Combine predictions from all trees. Use **Majority Voting** for Classification and **Average** for Regression.
7	**Final Prediction**	The combined result becomes the final prediction of the Random Forest model.

Strengths (Advantages)

1. High Accuracy

Combines predictions from many trees, giving better accuracy than a single Decision Tree.

2. Reduces Overfitting

Random Forest reduces overfitting because the final prediction is based on many trees instead of one.

3. Handles Large Datasets

Works efficiently with datasets containing thousands of records.

4. Handles Numerical and Categorical Data

Can process both numbers (Age, Salary) and categories (Gender, City).

5. Robust to Noise

Even if some data contains errors, the model still performs well.

6. Supports Classification and Regression

Can solve both classification and regression problems.

Weaknesses (Disadvantages)

1. Computationally Expensive

Requires more CPU and memory than a single Decision Tree.

2. Slow Training

Building many trees takes more time.

3. Difficult to Interpret

The model is harder to understand because it contains many trees.

4. Large Memory Requirement

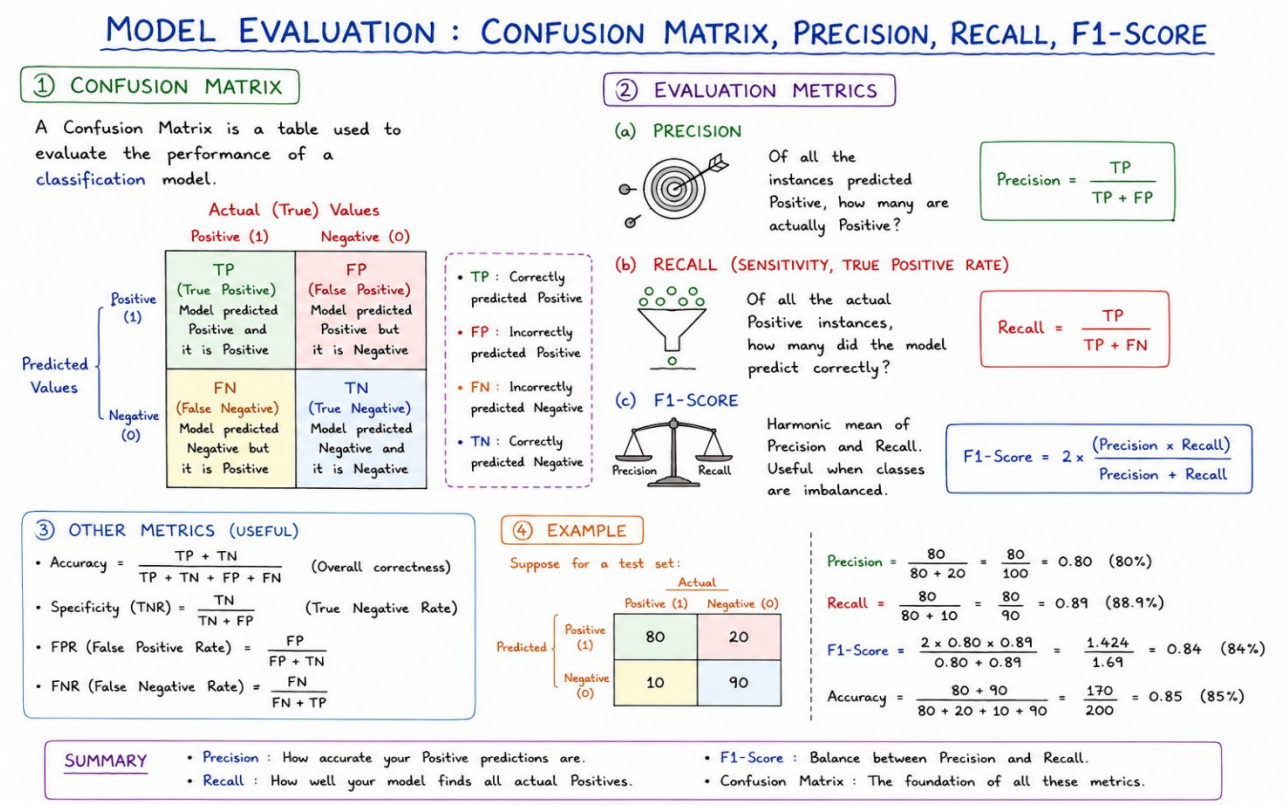
Stores many trees, requiring more memory.

Applications

- Medical diagnosis
- Fraud detection
- Credit risk analysis
- Customer segmentation
- Stock market prediction
- Weather forecasting
- Recommendation systems
- Spam email detection
- Image classification

Why Random Forest is Better than Decision Tree

Decision Tree	Random Forest
Uses one tree	Uses many trees
More overfitting	Less overfitting
Lower accuracy	Higher accuracy
Faster	Slightly slower
Easy to understand	Harder to interpret



1. What is Model Evaluation?

Definition:
Model Evaluation is the process of checking **how well a Machine Learning model performs** after training.

Example:

Suppose a hospital uses an ML model to predict whether a patient has **Diabetes**.

After prediction we compare

- Actual Result (True)
- Predicted Result (Model Output)

This comparison is called **Model Evaluation**.

2. Confusion Matrix

Definition

A **Confusion Matrix** is a table used to compare the **Actual values** with the **Predicted values** of a classification model.

It tells us

- Correct Predictions
- Wrong Predictions

Example Dataset

Student	Actual Value	Predicted Value	Result
1	Positive	Positive	TP (True Positive)
2	Positive	Positive	TP
3	Positive	Positive	TP
4	Positive	Negative	FN (False Negative)
5	Positive	Positive	TP
6	Negative	Negative	TN (True Negative)
7	Negative	Positive	FP (False Positive)
8	Negative	Negative	TN
9	Negative	Negative	TN
10	Negative	Negative	TN

Structure of Confusion Matrix

Actual / Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Remember:

Rows → Actual Values

Columns → Predicted Values

3. Four Important Terms

A) TP – True Positive

Meaning

Model predicted **Positive**

Actual is also **Positive**

Prediction is Correct.

Example

Actual Disease = Yes

Model Prediction = Yes

✓ Correct Prediction

TP = True Positive

B) TN – True Negative

Meaning

Model predicted **Negative**

Actual is also **Negative**

Prediction is Correct.

Example

Actual Disease = No

Model Prediction = No

✓ Correct Prediction

TN = True Negative

C) FP – False Positive

Meaning

Model predicted Positive

Actual is Negative

Prediction is Wrong.

Example

Actual Disease = No

Prediction = Yes

✕ Wrong Prediction

This is called

False Alarm

D) FN – False Negative

Meaning

Model predicted Negative

Actual is Positive

Prediction is Wrong.

Example

Actual Disease = Yes

Prediction = No

✕ Dangerous mistake

Patient actually has disease

Model says No Disease.

4. Precision

Definition

Precision tells

Out of all predicted Positive cases, how many are actually Positive?

Formula : Precision $= \frac{TP}{TP+FP}$

Example

Suppose

TP = 80

FP = 20

Precision

= 80/(80+20)

= 80/100

= **0.80 = 80%**

Meaning

Out of 100 predicted positive patients

Only 80 are really positive.

Easy Memory

Precision = Correct Positive Predictions

5. Recall (Sensitivity)

Definition

Recall tells

Out of all Actual Positive cases, how many did the model correctly identify?

Formula : Recall $= \frac{TP}{TP+FN}$

Example

TP = 80

FN = 10

Recall

=80/(80+10)

=80/90

=0.89

=89%

Meaning

Out of 90 actual patients

The model detected 80.

Easy Memory

Recall = Finds all Positive Cases

6. F1-Score

Definition

F1-Score is the **balance between Precision and Recall**.

Used when classes are **imbalanced**.

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

Example

Precision = 0.80

Recall = 0.89

F1 Score

= 0.84

=84%

Meaning

Higher F1 Score means

The model is good at both

- Finding positives
- Avoiding false alarms

7. Accuracy

Definition

Accuracy tells

How many predictions are correct overall.

Formula : $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

Example:

TP=80

TN=90

FP=20

FN=10

Accuracy

=(80+90)/(80+90+20+10)

=170/200

=85%

Meaning

The model correctly predicted **85%** of the total cases.

8. Other Evaluation Metrics

Specificity (True Negative Rate)

Formula : $Specificity = \frac{TN}{TN+FP}$

Meaning

Measures how well the model identifies **Negative** cases.

Example

Cancer Detection

Specificity tells

How many healthy people are correctly identified.

False Positive Rate (FPR)

Formula :$FPR = \frac{FP}{FP+TN}$

Meaning

Percentage of healthy people incorrectly predicted as sick.

Lower FPR is better.

False Negative Rate (FNR)

Formula :$FNR = \frac{FN}{FN+TP}$

Meaning

Percentage of sick people incorrectly predicted as healthy.

Lower FNR is better.

9. Example from the Figure

Given

Actual / Predicted	**Positive**	**Negative**
Positive	80	10
Negative	20	90

Step	**Metric**	**Formula / Calculation**	**Result**
Step 1	**Confusion Matrix Values**	TP = 80, TN = 90, FP = 20, FN = 10	TP=80, TN=90, FP=20, FN=10
Step 2	**Accuracy**	(TP + TN) / (TP + TN + FP + FN) = (80 + 90) / (80 + 90 + 20 + 10) = 170/200	**85%**
Step 3	**Precision**	TP / (TP + FP) = 80 / (80 + 20) = 80/100	**80%**
Step 4	**Recall**	TP / (TP + FN) = 80 / (80 + 10) = 80/90	**89%**

Step 5	**F1-Score**	2 × (Precision × Recall) / (Precision + Recall) = 2 × (0.80 × 0.89) / (0.80 + 0.89)	**84%**

10. Real-Life Example

Spam Email Detection

Suppose there are **200 emails**.

- 90 are Spam
- 110 are Normal

The ML model predicts

- TP = 80 → Spam correctly detected
- FP = 20 → Normal email marked as Spam
- FN = 10 → Spam missed
- TN = 90 → Normal email correctly detected

Using these values

- Accuracy = **85%**
- Precision = **80%**
- Recall = **89%**
- F1 Score = **84%**

Advantages

- Evaluates classification models.
- Shows correct and incorrect predictions.
- Helps compare different ML models.
- Improves model performance.
- Useful in medical diagnosis, fraud detection, and spam filtering.

Applications

- Medical Diagnosis
- Email Spam Detection
- Face Recognition
- Fraud Detection
- Credit Card Approval
- Disease Prediction

UNIT -III

Unit 3 Support Vector Machines and Ensemble Methods 9

Support Vector Machines: Concepts, Kernels, Hyperplane and Margin Concepts, Kernel Tricks: RBF and Polynomial, Ensemble Learning: Bagging, Boosting, and Voting, Gradient Boosting, AdaBoost, and XGBoost, Model Tuning and Hyperparameter Optimization.

SUPPORT VECTOR MACHINES (SVM)

Definition

Support Vector Machine (SVM) is a **Supervised Machine Learning algorithm** used for **Classification** and **Regression**. It separates different classes by finding the **best possible boundary (hyperplane)** with the **maximum margin**.

Example:

Suppose a bank wants to classify loan applicants.

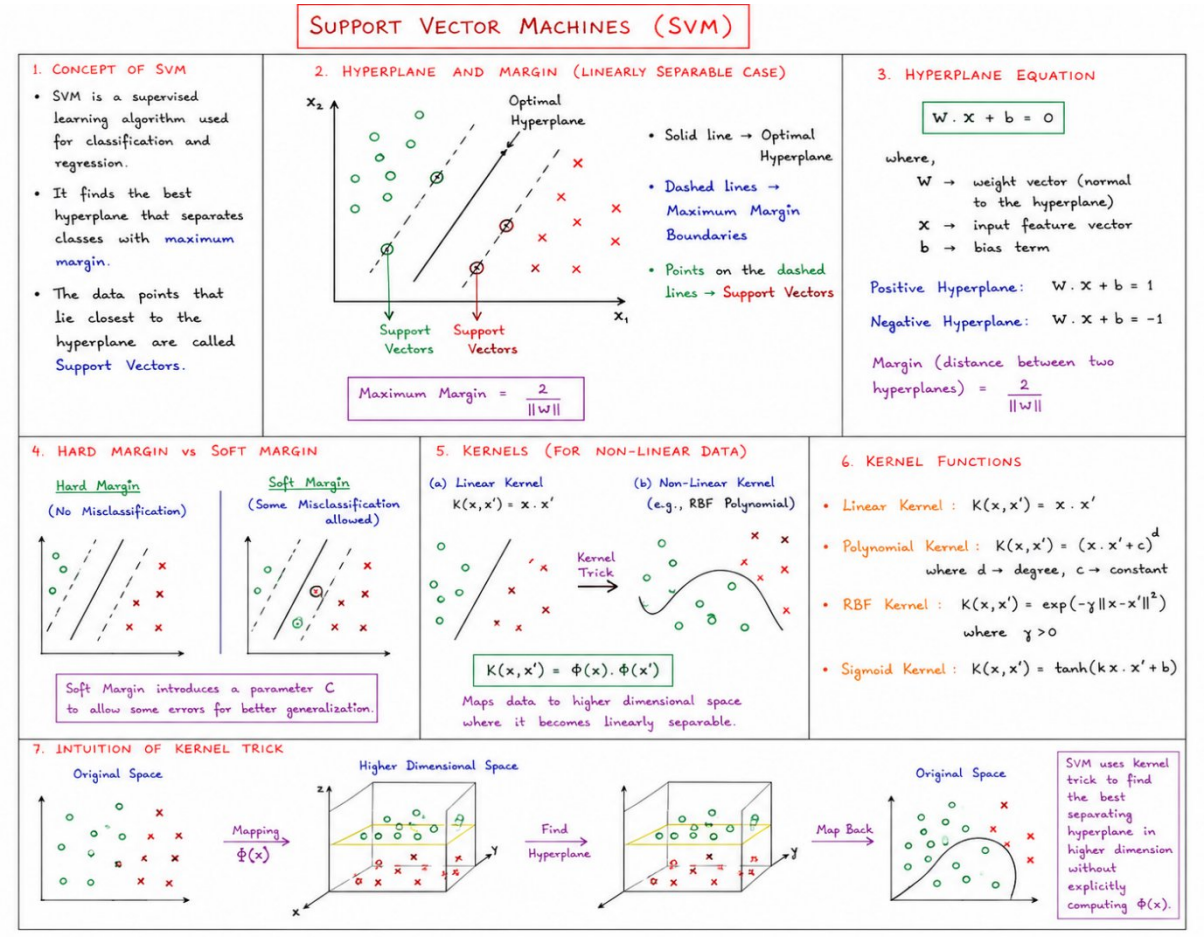
- Good Customer → ✔ Loan Approved
- Bad Customer → ✖ Loan Rejected

SVM finds the best line that separates the two groups.

✔ ✔ ✔ ✔

---------- Best Boundary

✖ ✖ ✖ ✖



1. CONCEPT OF SVM.

Explanation

SVM tries to find the **best decision boundary** that separates two classes.
Instead of simply separating the classes, SVM chooses the boundary that is **farthest from both classes**.
The nearest points to this boundary are called **Support Vectors**.

Example

Suppose we classify fruits.
Green circles = Apples
Red crosses = Oranges
○ ○ ○ ○

× × × ×
Many lines can separate them.
SVM chooses the **best line**.

Why Maximum Margin?

A larger margin gives

- Better prediction
- Less overfitting
- Higher accuracy

2. HYPERPLANE AND MARGIN (Linearly Separable Case)

Green Class
○ ○ ○ ○
..... Positive Margin
---------- Optimal Hyperplane
..... Negative Margin
× × × ×
Red Class

Hyperplane

The **middle line** is called the **Hyperplane**.
It separates the two classes.

Hyperplane

The **middle line** is called the **Hyperplane**.
It separates the two classes.

Margin

The space between the two dotted lines is called the **Margin**.
SVM always tries to maximize this margin.

Support Vectors

The points touching the dotted lines are called **Support Vectors**.
They decide where the hyperplane should be placed.

Example

Class A = Cats
Class B = Dogs
The animals closest to the boundary determine the final decision line.
Even if all other animals move slightly, the support vectors determine the classifier.

Why is Maximum Margin Important?

IMAGINE two roads.

Small road
Car | Bike

Large road
Car Bike

Which is safer?

☞ The second one.

Similarly,

Large Margin = Better Classification

Hyperplane

A **Hyperplane** is the decision boundary that separates two different classes.

- In **2D**, it is a straight line.
- In **3D**, it is a plane.
- In higher dimensions, it is called a hyperplane.

Hyperplane Equation

$$W \cdot X + b = 0$$

Where

- **W** = Weight Vector
- **X** = Feature Vector
- **b** = Bias

Example

Suppose students are classified based on Marks.

Marks	Result
90	Pass
85	Pass
40	Fail
35	Fail

The hyperplane separates **Pass** and **Fail** students.

4. Hard Margin SVM

- Hard Margin SVM is used when the data is perfectly separable. It does not allow any misclassification.
- The hyperplane separates the two classes with the maximum margin, and every data point is correctly classified.

Example

Suppose we classify students based on marks.

Marks	Result
90	Pass
85	Pass
35	Fail
30	Fail

Since all pass students have high marks and all fail students have low marks, a straight line can separate them perfectly. This is called Hard Margin SVM.

Advantage

- Gives high accuracy for clean and perfectly separated data.

Disadvantage

- Cannot handle noisy or overlapping data.

2. Soft Margin SVM

- Soft Margin SVM is used when the data is not perfectly separable. It allows a few misclassifications to find the best decision boundary.
- It uses a parameter called C to balance the margin and classification errors.

Example

Suppose we classify emails into Spam and Not Spam.
Most spam emails are easy to identify, but some genuine promotional emails may look like spam. Soft Margin SVM allows a few such errors to improve the overall classification performance.

Advantage

- Handles noisy and real-world data effectively.
- Better generalization for new data.

Disadvantage

- Selecting the correct C value is important.

Difference Between Hard Margin and Soft Margin

Hard Margin	Soft Margin
No misclassification is allowed.	A few misclassifications are allowed.
Used for perfectly separable data.	Used for noisy or overlapping data.
Sensitive to outliers.	Handles outliers better.

- In Support Vector Machines (SVM), **misclassification** means the model assigns a training data point to the wrong class or allows it to cross the separating margin boundary.
- The C parameter is a **regularization penalty** that controls how strictly the model avoids these errors.

5. Kernels (For Non-Linear Data)

This figure explains **Kernel Functions** in **Support Vector Machine (SVM)**. A kernel helps SVM classify **non-linear data** by converting it into a form that can be separated easily.

(a) Linear Kernel

A **Linear Kernel** is used when the data can be separated using a **straight line (hyperplane)**.
Example
Suppose we classify students based on marks.

- **Green circles (○)** = Pass
- **Red crosses (×)** = Fail

A straight line can easily separate the two classes.

○ ○ ○

---------- Straight Line

× × ×

Here, the **Linear Kernel** works well.

(b) Non-Linear Kernel

Sometimes, the data is **mixed together**, and a straight line cannot separate it.
In this case, SVM uses a **Non-Linear Kernel** (such as **RBF** or **Polynomial Kernel**) to create a **curved decision boundary**.

Example:

Suppose we classify **healthy** and **diseased** patients.
The patient data overlaps, so a straight line cannot separate them.
A **curved boundary** is needed to classify them correctly.

○ ○ ×

○ × ○

× × ○

Here, a **Non-Linear Kernel** separates the classes.

Kernel Trick

The **Kernel Trick** maps the original data into a **higher-dimensional space**, where it becomes easy to separate using a straight hyperplane.

The mapping is represented by:

$$K(x, x') = \phi(x) \cdot \phi(x')$$

You do **not** need to remember the formula for understanding. Just remember:

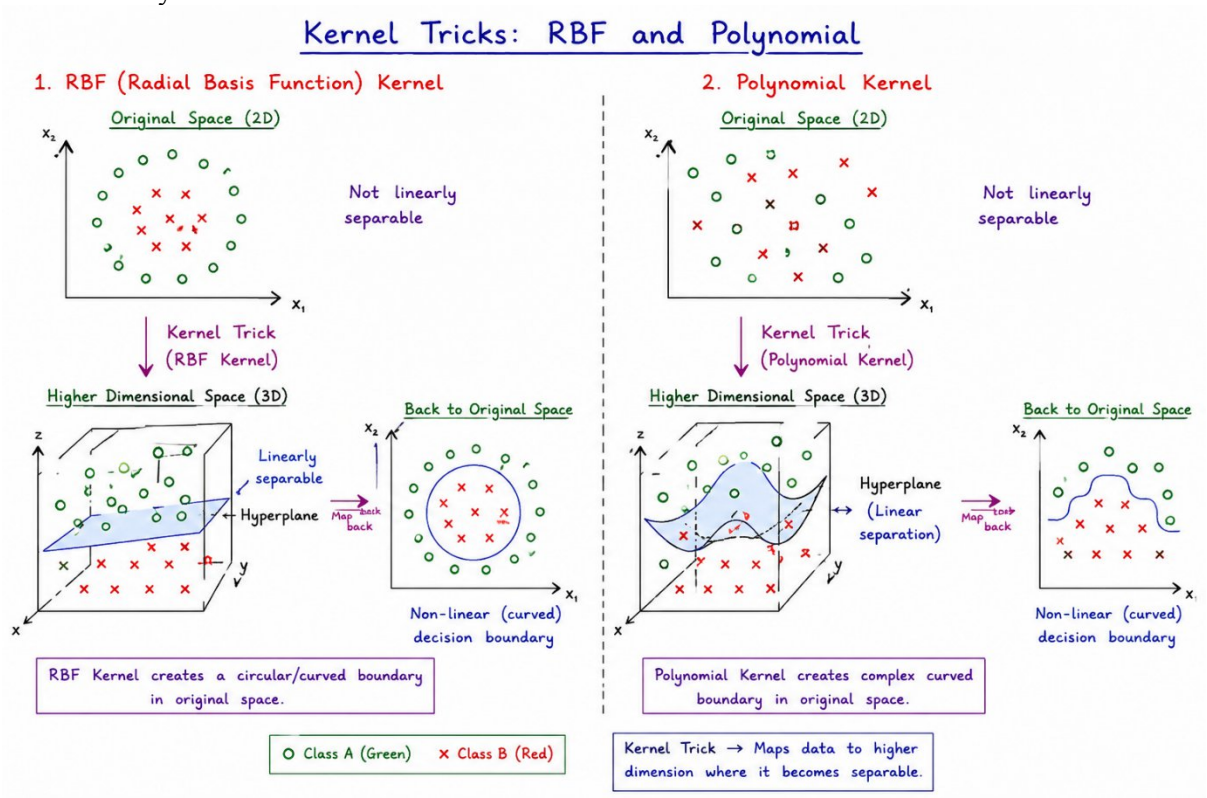
Kernel Trick converts difficult (non-linear) data into an easier form so that SVM can classify it accurately.

Real-Life Example

Imagine **red and green balls** mixed on a table.

- Using a ruler (straight line), you **cannot separate** them.
- If you **lift some balls to another level (higher dimension)**, they can now be separated by a flat sheet.

This is exactly what the **Kernel Trick** does.



1. RBF (Radial Basis Function) Kernel

The **RBF Kernel** is used when the data is highly complex and cannot be separated by a straight line. It maps the data into a higher-dimensional space and creates a **circular or smooth curved decision boundary**.

Figure Explanation

1. In the **original 2D space**, the green circles (○) surround the red crosses (×), so a straight line cannot separate them.
2. The **RBF Kernel** maps the data to a **3D (higher-dimensional) space**.

3. In 3D, SVM finds a **hyperplane** that separates the two classes.
4. After mapping back to 2D, the hyperplane becomes a **circular decision boundary**.

Example

In **cancer detection**, cancer cells are often surrounded by normal cells. The **RBF Kernel** creates a circular boundary to classify cancer cells accurately.

2. Polynomial Kernel

The **Polynomial Kernel** is used when the relationship between data points is **curved or polynomial**. It maps the data into a higher-dimensional space and creates a **complex curved decision boundary**.

Figure Explanation

1. The original data is mixed and cannot be separated by a straight line.
2. The **Polynomial Kernel** maps the data into a higher-dimensional space.
3. SVM finds a **hyperplane** in this space.
4. After mapping back, a **complex curved boundary** separates the classes.

Example

In **loan approval**, customer details such as **salary, age, and credit score** have a complex relationship. The **Polynomial Kernel** creates a curved boundary to classify approved and rejected customers.

Kernel Tricks and Types

Definition

A **Kernel Function** is a mathematical function used in **Support Vector Machine (SVM)** to classify data. It helps SVM separate **non-linear data** by mapping it into a **higher-dimensional space**, making classification easier.

Types of Kernel Functions

1. Linear Kernel

- Used for **linearly separable data**.
- It separates data using a **straight line (hyperplane)**.

Formula:

$$K(x, x') = x \cdot x'$$

Example:

Classifying students as **Pass** or **Fail** based on marks.

2. Polynomial Kernel

- Used when the data has a **curved relationship**.
- Creates a **curved decision boundary**.

Formula:

$$K(x, x') = (x \cdot x' + c)^d$$

Where:

- **d** = Degree
- **c** = Constant

Example:

Student performance prediction based on marks and attendance.

3. RBF (Radial Basis Function) Kernel

- Used for **complex and non-linear datasets**.
- It is the **most commonly used kernel**.

Formula:

$$K(x, x') = \exp\left(-\gamma \,||\, x - x' \,||^2\right)$$

Where:

- **γ (Gamma)** controls the influence of data points.

Example:
Face recognition and cancer detection.

4. Sigmoid Kernel

- Works similarly to a **Neural Network activation function**.
- Used for pattern recognition.

Formula:

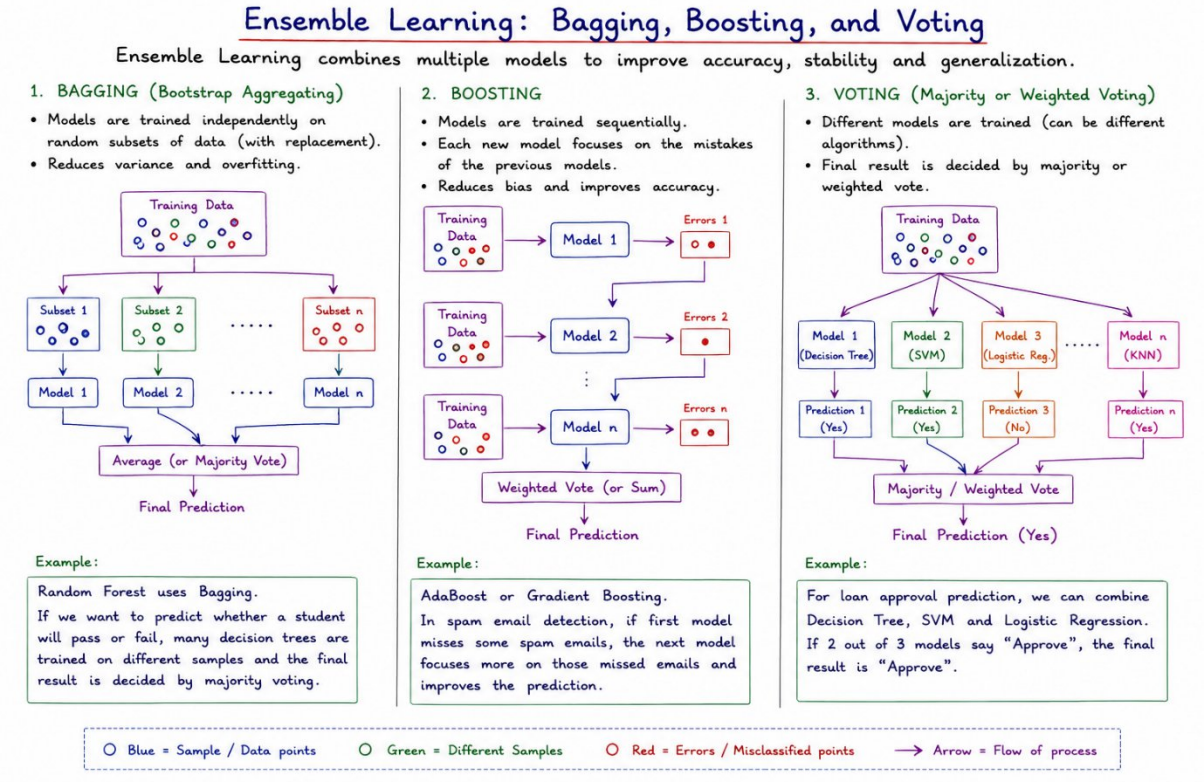
$$K(x, x') = \tanh\left(kx \cdot x' + b\right)$$

Example:
Image classification and speech recognition.

The **Kernel Trick** is a technique used in **Support Vector Machine (SVM)** to classify **non-linear data**. It maps the original data into a **higher-dimensional space**, where the data becomes easy to separate using a **hyperplane**, without explicitly calculating the new coordinates.

Steps of Kernel Trick

1. **Original Space:** The data is mixed and cannot be separated by a straight line.
2. **Mapping:** The kernel function maps the data to a **higher-dimensional space** using **Φ(x)**.
3. **Find Hyperplane:** In the higher-dimensional space, SVM finds the **best separating hyperplane**.
4. **Map Back:** The result appears as a **curved decision boundary** in the original space.



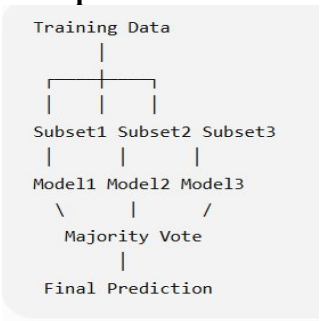
1. Bagging (Bootstrap Aggregating)

Figure Explanation

In the figure:

- The **training data** is divided into **different random subsets**.
- Each subset is used to train a **different model independently**.
- Finally, the outputs of all models are combined using **Majority Voting** (Classification) or **Average** (Regression).

Example



Suppose we want to predict whether a student will **Pass or Fail**.

- Decision Tree 1 → Pass
- Decision Tree 2 → Pass
- Decision Tree 3 → Fail

Majority Vote = Pass
Result: Student is predicted as **Pass**.
Advantages

- Reduces overfitting.
- Improves prediction accuracy.
- Models work independently.

Real-Time Example
Random Forest uses the **Bagging** technique.

2. Boosting
Figure Explanation
In the figure:

- Models are trained **one after another (sequentially)**.
- Each new model learns from the **mistakes (errors)** of the previous model.
- The final prediction is obtained by combining all models.

Training Data
↓
Model 1
↓
Finds Errors
↓
Model 2
↓
Finds Remaining Errors
↓
Model 3
↓
Final Prediction
Example
Suppose we want to detect **Spam Emails**.

- Model 1 correctly identifies **80 emails** but misses **20 emails**.
- Model 2 focuses only on the **20 missed emails**.
- Model 3 improves the remaining errors.

Finally, all models work together to give a more accurate prediction.
Advantages

- Improves accuracy.
- Reduces prediction errors.
- Works well for complex datasets.

Real-Time Example

- AdaBoost

- Gradient Boosting
- XGBoost

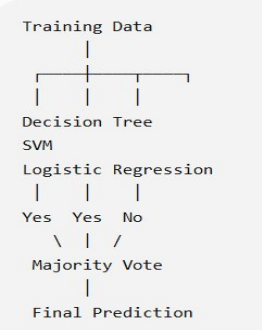
3. Voting

Figure Explanation

In the figure:

- Different Machine Learning algorithms are trained on the **same dataset**.
- Each model gives its own prediction.
- The final result is selected using **Majority Voting** or **Weighted Voting**.

Example



Suppose a bank wants to approve a loan.

Model	Prediction
Decision Tree	Approve
SVM	Approve
Logistic Regression	Reject

Since **2 out of 3 models** predict **Approve**, the final decision is:

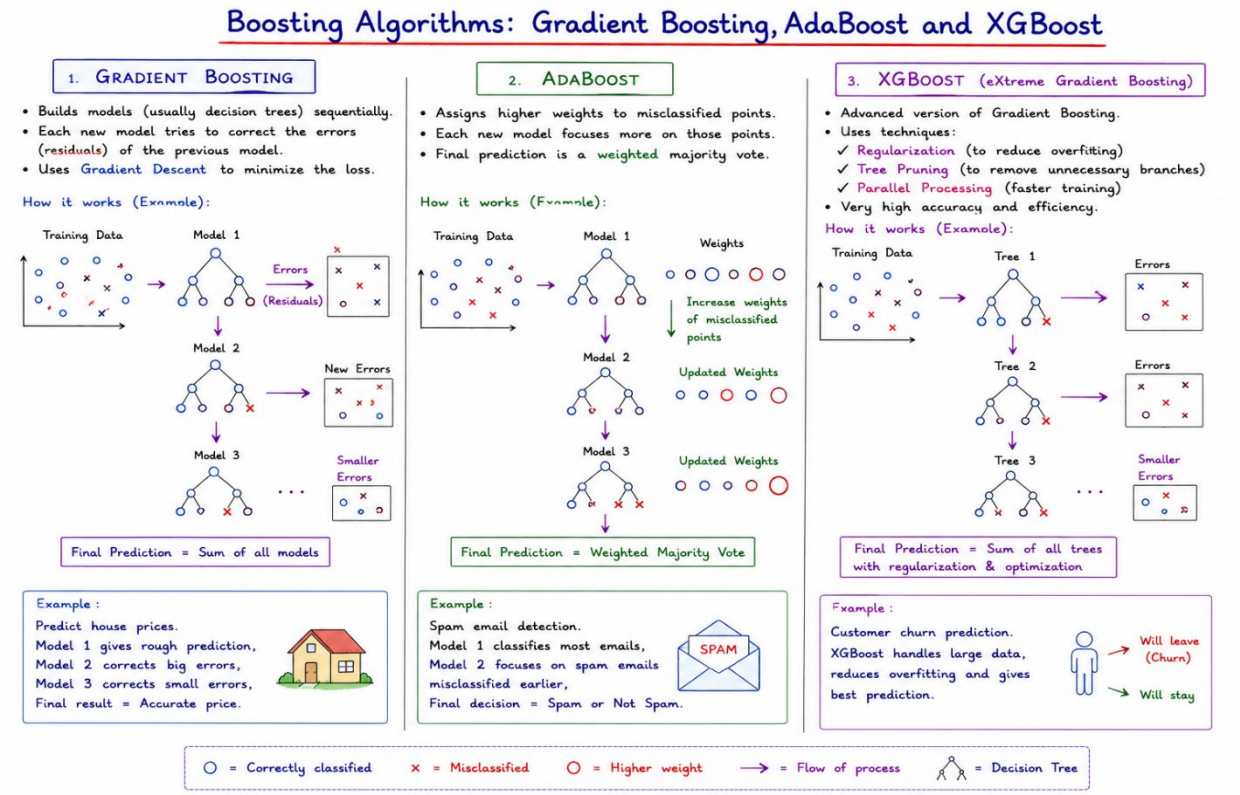
Loan Approved

Advantages

- Simple and easy to implement.
- Increases prediction accuracy.
- Uses different algorithms together.

Difference Between Bagging, Boosting and Voting

Bagging	Boosting	Voting
Models train independently.	Models train one after another.	Different models vote for the final result.
Reduces variance.	Reduces bias and errors.	Improves overall accuracy.
Example: Random Forest	Example: AdaBoost, XGBoost	Example: Decision Tree + SVM + Logistic Regression



1. Gradient Boosting

Definition

Gradient Boosting is an ensemble learning algorithm that builds **multiple decision trees one by one**. Every new tree learns from the **errors (residuals)** of the previous tree to improve prediction accuracy.

Figure Explanation

Step 1: Training Data

The first decision tree (**Model 1**) is trained using the training dataset.

↓

Step 2: Find Errors (Residuals)

After prediction, some values are predicted incorrectly.
These prediction mistakes are called **Residual Errors**.
In the figure, the **red crosses (×)** represent the errors.

↓

Step 3: Build Model 2

The second tree is trained mainly to correct the mistakes made by Model 1.

↓

Step 4: Build Model 3

The third tree corrects the remaining small errors.

↓

Final Prediction

All the predictions from the trees are combined to produce an accurate final result.

Real-Life Example – House Price Prediction

Suppose we want to predict the price of a house.

Model	Prediction
Model 1	₹48 Lakhs
Actual Price	₹50 Lakhs
Error	₹2 Lakhs

Model 2 learns this error and predicts ₹49.5 Lakhs.
Model 3 further corrects it to ₹50 Lakhs.
Finally, Gradient Boosting predicts the correct house price.

Advantages

- High prediction accuracy.
- Reduces prediction errors.
- Suitable for regression and classification problems.

2. AdaBoost (Adaptive Boosting)

Definition

AdaBoost is a boosting algorithm that gives **more importance (higher weight)** to the data points that were classified incorrectly.
The next model focuses more on those difficult data points.

Figure Explanation

Step 1: Train Model 1
The first model classifies the data.
Some points are misclassified.
↓
Step 2: Increase Weights
The wrongly classified points are given **higher weights**.
The correctly classified points receive lower weights.
↓
Step 3: Train Model 2
The second model pays more attention to the higher-weight data points.
↓
Step 4: Repeat
This process continues until prediction accuracy improves.
↓
Final Prediction
The final prediction is obtained using a **Weighted Majority Vote**.

Real-Life Example – Spam Email Detection

Suppose there are **100 emails**.
Model 1 correctly identifies **90 emails**.
It misses **10 spam emails**.
AdaBoost gives **higher importance** to these 10 emails.
The next model learns these difficult emails and improves the prediction.
Finally, spam detection becomes more accurate.

Advantages

- Focuses on difficult cases.
- Improves classification accuracy.
- Reduces classification errors.

3. XGBoost (Extreme Gradient Boosting)

Definition

XGBoost is an **improved and faster version of Gradient Boosting**.
It provides **higher accuracy**, faster training, and reduces overfitting using advanced optimization techniques.

Figure Explanation

Step 1: Training Data
The first decision tree is trained.
↓

Step 2: Find Errors
The model identifies prediction errors.
↓
Step 3: Build More Trees
The next trees correct these errors one by one.
↓
Step 4: Optimization
Unlike Gradient Boosting, XGBoost also performs:

- **Regularization** → Prevents overfitting.
- **Tree Pruning** → Removes unnecessary branches.
- **Parallel Processing** → Makes training much faster.

↓
Final Prediction
The predictions from all trees are combined to produce the most accurate result.

Real-Life Example – Customer Churn Prediction
A telecom company wants to predict which customers may leave the company.
Customer details include:

- Monthly bill
- Internet usage
- Recharge history
- Complaints

XGBoost analyzes all these features and predicts:

- **Will Stay**
- **Will Leave (Churn)**

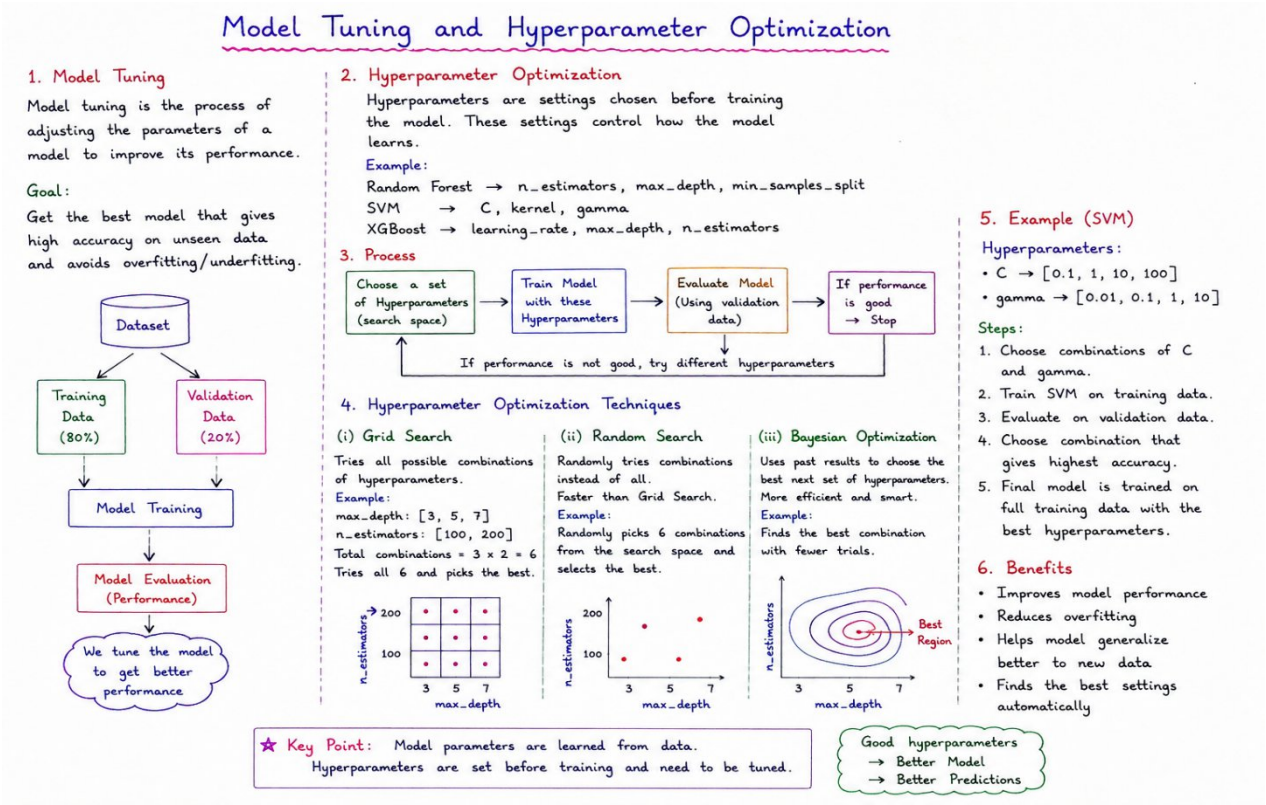
This helps the company offer discounts to customers who are likely to leave.

Advantages

- Very high accuracy.
- Faster than Gradient Boosting.
- Handles large datasets efficiently.
- Reduces overfitting.

Difference Between Gradient Boosting, AdaBoost and XGBoost

Gradient Boosting	**AdaBoost**	**XGBoost**
Corrects previous model errors (residuals).	Gives higher weight to misclassified data.	Improved version of Gradient Boosting with optimization.
Learns from residual errors.	Learns from weighted errors.	Uses residuals + regularization + pruning.
Good accuracy.	Better for simple classification tasks.	Highest accuracy and fastest performance.
Example: House Price Prediction	**Example:** Spam Email Detection	**Example:** Customer Churn Prediction



Definition

Model Tuning is the process of adjusting a machine learning model to improve its performance and prediction accuracy.

Hyperparameter Optimization is the process of selecting the **best hyperparameter values** before training the model so that it gives the highest accuracy.

1. Model Tuning

Explanation (Figure)

The left side of the figure shows the **Model Tuning process**.

Step 1: Split the Dataset

The dataset is divided into:

- **Training Data (80%)**
- **Validation Data (20%)**

↓

Step 2: Train the Model

The model is trained using the **Training Data**.

↓

Step 3: Evaluate the Model

The model is tested using the **Validation Data**.

↓

Step 4: Tune the Model

If the accuracy is low, change the model settings (hyperparameters) and train again until better performance is achieved.

Real-Life Example

Suppose a teacher prepares students for an exam.

- First, the teacher teaches the lessons.
- Then gives a **practice test**.
- If students score low, the teacher changes the teaching method and gives another test.
- This process continues until students perform well.

This is called **Model Tuning**.

2. Hyperparameter Optimization

Definition

Hyperparameters are **settings chosen before training** the model. They control how the model learns.
Examples:

Algorithm	Hyperparameters
SVM	C, Gamma, Kernel
Random Forest	Number of Trees, Max Depth
XGBoost	Learning Rate, Max Depth

Process (Figure)

The middle part of the figure explains the process.

1. Choose a set of hyperparameters.
2. Train the model.
3. Evaluate the model using validation data.
4. If accuracy is good → Stop.
5. Otherwise, choose another set of hyperparameters and repeat.

Real-Life Example

Suppose you are making tea.
Hyperparameters are like:

- Amount of sugar
- Amount of milk
- Amount of tea powder

You prepare the tea.
If the taste is not good, you change the ingredients and prepare it again.
Finally, you get the **best tea**.
Similarly, Machine Learning selects the **best hyperparameters** to get the **best model**.

3. Hyperparameter Optimization Techniques

(i) Grid Search

- Tests **all possible combinations** of hyperparameters.
- Selects the combination with the highest accuracy.

Example:
Choose:

- Max Depth = 3, 5, 7
- Number of Trees = 100, 200

Grid Search checks **all 6 combinations** and selects the best one.

(ii) Random Search

- Tests **random combinations** instead of all combinations.
- Faster than Grid Search.

Example:
Instead of testing all 6 combinations, Random Search randomly tests only 3 combinations and selects the best one.

(iii) Bayesian Optimization

- Uses previous results to intelligently choose the next hyperparameter values.
- Requires fewer trials than Grid Search.

Example:
Like a GPS finding the shortest route based on previous traffic information instead of checking every possible road.

4. Example (SVM)

Suppose we train an SVM model.

Hyperparameters:

- **C = 0.1, 1, 10, 100**
- **Gamma = 0.01, 0.1, 1**

Steps:

1. Choose values of C and Gamma.
2. Train the SVM model.
3. Test it using validation data.
4. Select the combination with the highest accuracy.

5. Benefits

- Improves model accuracy.
- Reduces overfitting.
- Gives better predictions on new data.
- Finds the best model settings automatically.

Real-Life Example

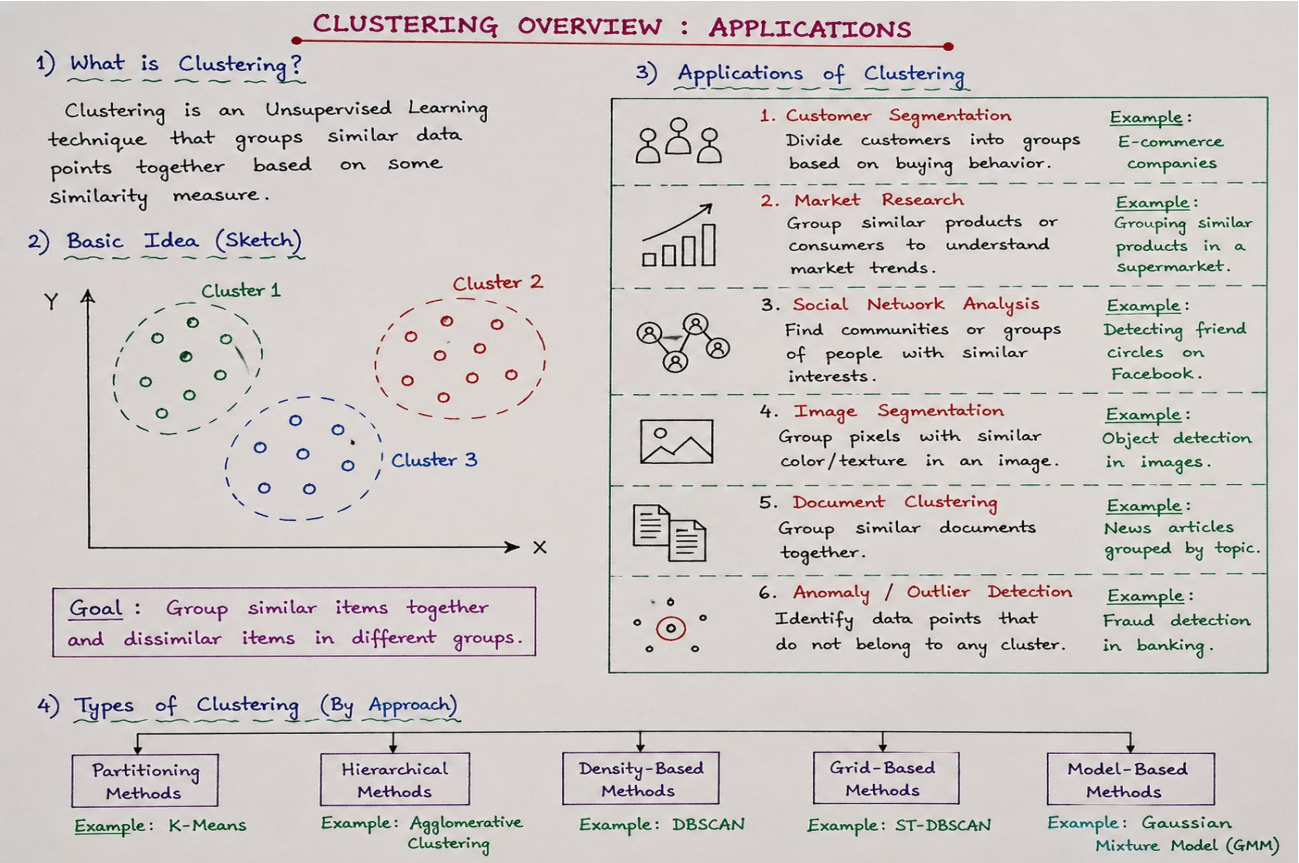
Suppose you are buying a mobile phone.
You compare different combinations:

- RAM (6GB, 8GB)
- Storage (128GB, 256GB)
- Battery (5000mAh, 6000mAh)

After comparing different combinations, you choose the best phone.
Similarly, **Hyperparameter Optimization** tries different parameter combinations and selects the **best-performing model**.

Unit 4 Unsupervised Learning Techniques 9

Clustering Overview: Applications, K-Means Clustering Algorithm, Hierarchical Clustering, DBSCAN and Density-Based Methods, Principal Component Analysis (PCA) for Dimensionality Reduction, Silhouette Score, Davies-Bouldin Index for Cluster Validation.



Clustering Overview and Applications

Definition

Clustering is an **Unsupervised Machine Learning** technique used to group similar data points into the same cluster and separate dissimilar data points into different clusters. Since there are **no predefined labels**, the algorithm automatically discovers hidden patterns in the data.

Example: A teacher groups students based on their marks without knowing their grades beforehand. Students with similar marks are placed in the same group.

Explanation of the Diagram

1. What is Clustering?

The left side of the figure explains the meaning of clustering.

- Similar data points are grouped together.
- Different data points are placed in different groups.
- The machine automatically finds these groups based on similarity.

Real-Life Example

Suppose a library has thousands of books.

Instead of mixing all books together, the librarian arranges them into groups.

- 📚 Science Books
- 📖 History Books
- 💻 Computer Books
- 📘 Mathematics Books

Here, books with similar subjects are grouped into the same section. This is called **clustering**.

2. Basic Idea of Clustering

The graph shows three different clusters.

- **Cluster 1 (Green)** contains similar data points.
- **Cluster 2 (Red)** contains another group of similar points.
- **Cluster 3 (Blue)** contains a different group.

Each cluster represents objects that are similar to one another but different from the objects in other clusters.

Real-Life Example

A college groups students according to their marks.

Cluster	Students
Cluster 1	Marks above 90
Cluster 2	Marks between 70–89
Cluster 3	Marks below 70

Students with similar marks are grouped together automatically.

Goal of Clustering

The main goal of clustering is:

- To group similar objects together.
- To separate different objects into different groups.
- To discover hidden patterns in data.

Real-Life Example

In a supermarket,

- Fruits are kept together.
- Vegetables are kept together.
- Dairy products are kept together.
- Bakery items are kept together.

This grouping makes shopping easier.

Applications of Clustering

1. Customer Segmentation

Companies divide customers into different groups based on their purchasing behavior.

Real-Life Example

Amazon and Flipkart analyze customer shopping habits.

Suppose,

Customer A buys:

- Laptop
- Mouse
- Keyboard

Customer B buys:

- Shoes
- T-shirts
- Jeans

The company creates two clusters:

- Electronics Customers
- Fashion Customers

Later, similar products are recommended to each customer.

2. Market Research

Businesses group customers or products to understand market trends.

Real-Life Example

A supermarket observes that customers buying bread also buy butter and milk.

These products are placed close together to increase sales.

This grouping helps improve marketing strategies.

3. Social Network Analysis
Social media platforms group users with similar interests.
Real-Life Example
Facebook and Instagram analyze user interests.
If you like:

- Machine Learning
- Python
- Artificial Intelligence

The platform recommends:

- AI groups
- Python communities
- Programming friends

because users with similar interests belong to the same cluster.

4. Image Segmentation
Clustering separates an image into different regions.
Real-Life Example
A self-driving car camera identifies:

- Cars
- Pedestrians
- Trees
- Roads

Each object forms a separate cluster, helping the vehicle understand its surroundings.

5. Document Clustering
Documents with similar topics are grouped together.
Real-Life Example
Google News automatically groups articles into categories such as:

- Sports
- Business
- Entertainment
- Technology

This allows users to easily find news related to their interests.

6. Anomaly (Outlier) Detection
Clustering identifies unusual data points that do not belong to any cluster.
Real-Life Example
A bank monitors a customer's daily spending.
Normally, the customer spends around:

- ₹500
- ₹700
- ₹1000

Suddenly, a transaction of **₹2,50,000** occurs in another country.
This transaction is very different from the customer's usual spending pattern, so it is detected as a **fraudulent transaction (outlier)**.

Types of Clustering (Shown at the Bottom of the Figure)
1. Partitioning Method

- Example: **K-Means**
- Divides data into a fixed number of clusters.

Real-Life Example: Dividing students into **three sections (A, B, C)** based on performance.

2. Hierarchical Method

- Example: **Agglomerative Clustering**

- Creates clusters in the form of a tree (hierarchy).

Real-Life Example: Organizing computer files into folders and subfolders.

3. Density-Based Method

- Example: **DBSCAN**
- Groups densely packed data points and identifies isolated points as noise.

Real-Life Example: At a railway station, people standing together form clusters, while a person standing alone is treated as an outlier.

4. Grid-Based Method

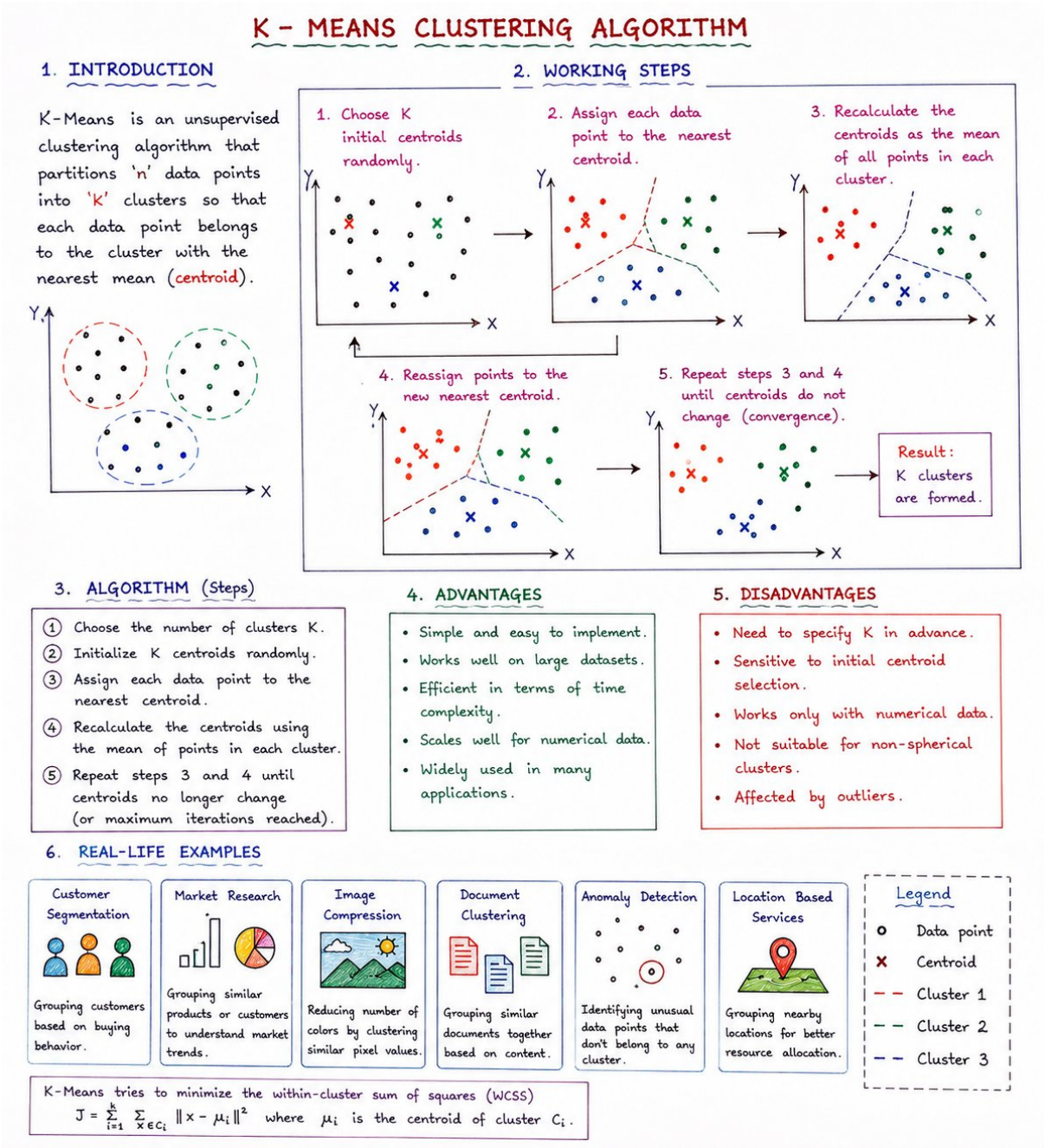
- Divides the data space into grids and forms clusters based on dense grid cells.

Real-Life Example: Google Maps divides a city into small regions to analyze traffic density.

5. Model-Based Method

- Example: **Gaussian Mixture Model (GMM)**
- Assumes data follows a probability distribution and assigns each data point to the most likely cluster.

Real-Life Example: Hospitals group patients into different health-risk categories based on medical reports.



Introduction

- Clustering is one of the most popular techniques in **Unsupervised Learning** because it helps discover hidden patterns in data without using predefined labels.
- Among all clustering techniques, **K-Means** is the simplest and most widely used algorithm due to its speed, simplicity, and efficiency.
- It is commonly used in **customer segmentation, market analysis, image processing, document clustering, recommendation systems, and medical data analysis**.
- The main objective of K-Means is to minimize the distance between data points and their cluster centroid.

Working Principle of K-Means

K-Means works by repeatedly assigning data points to the nearest centroid and updating the centroid until no further changes occur.

Step 1: Choose the Number of Clusters (K)

First, decide how many clusters are required.

For example,

Suppose a college wants to classify students into:

- High Performers
- Average Performers
- Low Performers

Then,
K = 3

Step 2: Select Initial Centroids
Randomly select **K centroids** from the dataset.
Initially, these centroids may not represent the actual center of the clusters.

Step 3: Assign Data Points to the Nearest Centroid
Each data point calculates its distance from every centroid.
It joins the cluster whose centroid is nearest.
Thus,
similar data points are grouped together.

Step 4: Calculate New Centroids
For every cluster,
calculate the average position (mean) of all data points.
This new average becomes the new centroid.
The centroid shifts toward the center of the cluster.

Step 5: Repeat Until Convergence
Repeat

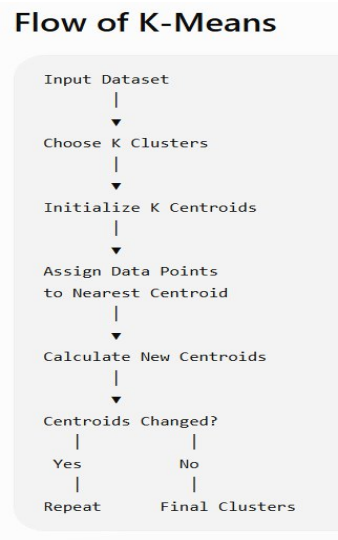
- Assign points
- Calculate new centroid

until

- No data point changes its cluster.
- Centroids stop moving.

The algorithm then terminates.

Algorithm
Input: Dataset D, Number of Clusters (K)
Output: K Clusters
Algorithm Steps
Step 1: Choose the value of K.
Step 2: Randomly initialize K centroids.
Step 3: Compute the distance between every data point and each centroid.
Step 4: Assign each data point to its nearest centroid.
Step 5: Recalculate the centroid by taking the average of all points in that cluster.
Step 6: Repeat Steps 3–5 until centroids no longer change.
Step 7: Display the final clusters.



Advantages

1. Simple to Understand

The algorithm is very easy to learn and implement.

Example: Students can easily understand it using the marks grouping example.

2. Fast Execution

K-Means processes large datasets quickly.

Example: Amazon groups millions of customers within seconds.

3. Efficient

It requires less computational time compared to many clustering algorithms.

4. Scalable

It works well even for very large datasets.

Example: Google clusters billions of search queries every day.

5. Widely Used

Used in many real-world applications like

- Healthcare
- Banking
- E-commerce
- Image Processing
- Social Media

Disadvantages

1. K Must Be Specified in Advance

The user must know the number of clusters before running the algorithm.

2. Sensitive to Initial Centroids

Different random centroids may produce different clustering results.

3. Works Only for Numerical Data

Categorical or text data must be converted into numerical form before applying K-Means.

4. Sensitive to Outliers

Extreme values affect the centroid position.

5. Cannot Handle Irregular Cluster Shapes

K-Means performs well only when clusters are approximately circular or spherical.

Real-Life Applications

1. Customer Segmentation
Companies classify customers based on purchasing behavior.
Example
Amazon groups customers into

- Electronics Buyers
- Fashion Buyers
- Grocery Buyers

This helps recommend suitable products.

2. Market Research
Businesses analyze customer purchasing patterns.
Example
A supermarket discovers that customers buying bread also buy butter and milk.
These products are placed together to increase sales.

3. Image Compression
K-Means groups similar-colored pixels to reduce image size.
Example
WhatsApp compresses images before sending them.

4. Document Clustering
Groups similar documents based on their content.
Example
Google News automatically groups articles into:

- Sports
- Technology
- Business
- Entertainment

5. Fraud Detection
Detects abnormal transactions.
Example
If a customer usually spends ₹500–₹1000 per day but suddenly spends ₹2,00,000 in another country, the bank flags it as suspicious.

6. Healthcare
Hospitals classify patients based on symptoms and medical records.
Example
Patients with similar symptoms are grouped for easier diagnosis and treatment planning.

7. Education
Schools group students according to academic performance.
Example
Students are divided into:

- High Performers
- Average Performers
- Slow Learners

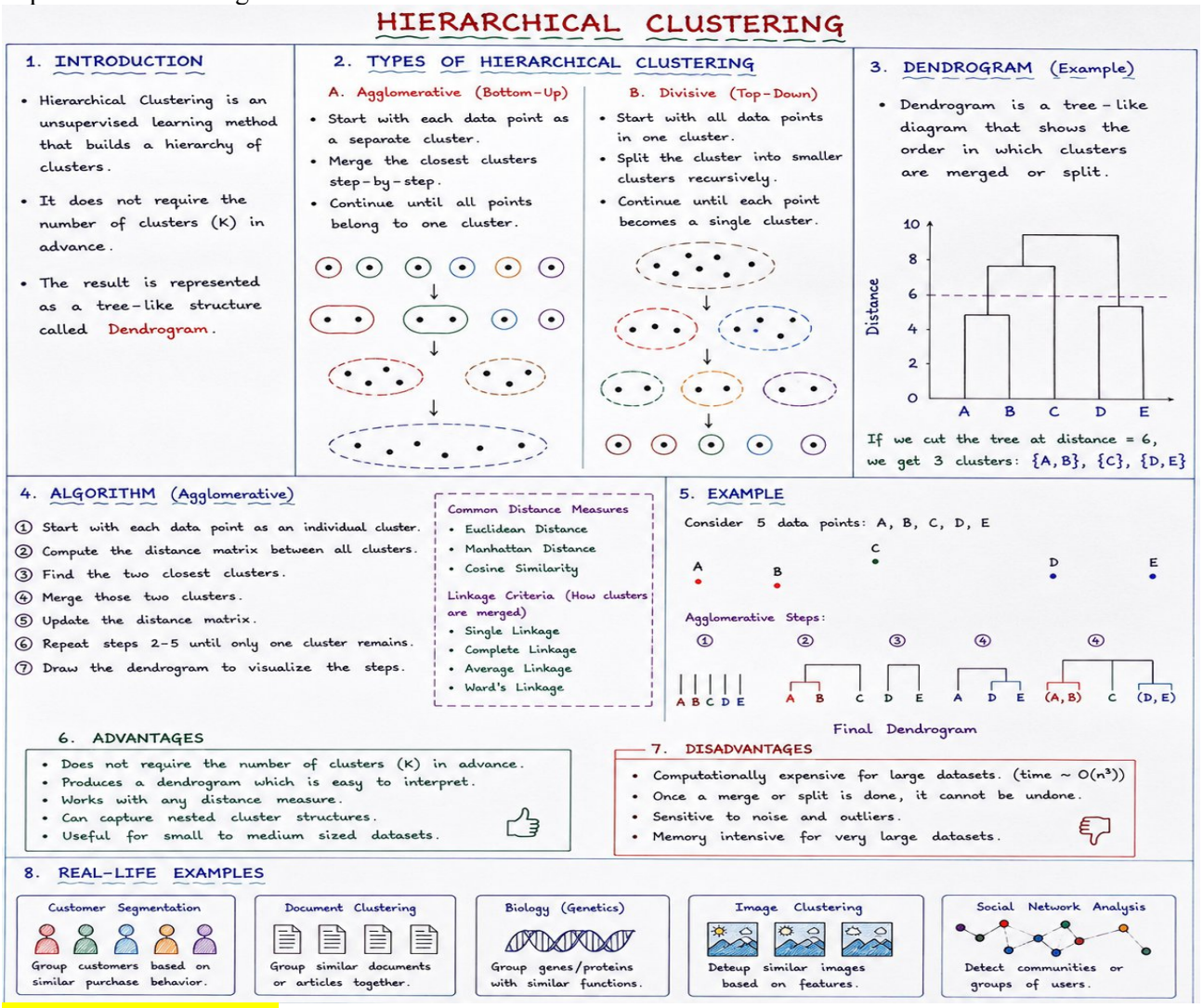
This helps teachers provide personalized support.

8. Location-Based Services
Nearby places are grouped together.
Example
Google Maps groups nearby:

- Restaurants
- Hospitals
- Petrol Pumps

- ATMs

to provide better navigation.



Hierarchical Clustering

Definition

Hierarchical Clustering is an **Unsupervised Machine Learning** algorithm that groups similar data points into clusters and represents them in a **tree-like structure called a Dendrogram**. Unlike K-Means, it **does not require the number of clusters (K) to be specified in advance**.

Introduction

Hierarchical Clustering creates a **hierarchy (tree)** of clusters. It is mainly used when we want to understand the relationship between data points.

The algorithm works in two ways:

1. **Agglomerative Clustering (Bottom-Up Approach)**
2. **Divisive Clustering (Top-Down Approach)**

The final clustering result is represented using a **Dendrogram**, which helps visualize how clusters are merged or divided.

Types of Hierarchical Clustering

The figure shows **two types** of Hierarchical Clustering.

1. Agglomerative Clustering (Bottom-Up Approach)

Explanation

Agglomerative means **"to combine or merge."**

Initially,

- Every data point is treated as a separate cluster.
- The algorithm repeatedly merges the two closest clusters.
- This process continues until all points become one large cluster.

Steps

Initially
A B C D E
Each point is an individual cluster.
↓
Merge closest points
(A,B) C D E
↓
Merge next closest
(A,B) (D,E) C
↓
Merge again
(A,B,C) (D,E)
↓
Final
(A,B,C,D,E)
All points become one cluster.

Real-Life Example

Suppose there are **five students**.
Initially
Every student stands alone.
The teacher asks students with similar interests to stand together.
Eventually,
the whole class becomes one large group.
This is exactly how Agglomerative Clustering works.

2. Divisive Clustering (Top-Down Approach)

Explanation

Divisive means **"to divide or split."**
Initially,
All data points belong to one large cluster.
The algorithm repeatedly divides the cluster into smaller clusters.
The process continues until every data point becomes an individual cluster.

Steps

Initially
(A,B,C,D,E)
↓
Split
(A,B,C) (D,E)
↓
Split again
(A,B) C D E
↓
Final
A B C D E
Every point becomes an individual cluster.

Real-Life Example

Imagine a school.
Initially,

All students attend one assembly.
Later,
they are divided into

- CSE
- ECE
- Mechanical

Then,
CSE students are divided into

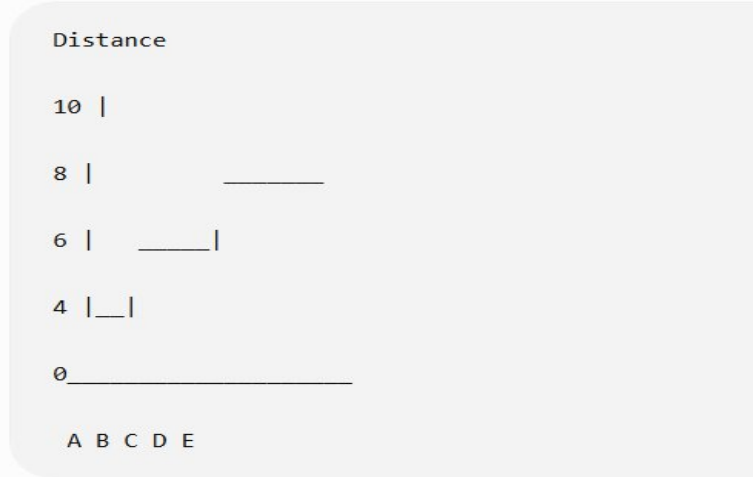
- AIML
- Data Science
- Cyber Security

This is Divisive Clustering.

Difference Between Agglomerative and Divisive

Agglomerative	**Divisive**
Bottom-Up Approach	Top-Down Approach
Starts with many clusters	Starts with one cluster
Merges clusters	Splits clusters
Ends with one cluster	Ends with individual clusters

3.)Dendrogram:
A **Dendrogram** is a **tree-like diagram** that shows how clusters are merged or split.
The vertical axis represents the **distance (similarity)** between clusters.
The horizontal axis represents the **data points**.
In the given figure,



If we draw a horizontal line at **Distance = 6**,

The dendrogram produces

```
Cluster 1 = {A,B}

Cluster 2 = {C}

Cluster 3 = {D,E}
```

Real-Life Example
Imagine a family tree.
Grandfather

↓
Father
↓
Children
The tree shows relationships among family members.
Similarly,
A dendrogram shows relationships among clusters.

Algorithm (Agglomerative Clustering)

Definition

Agglomerative Clustering is a **Bottom-Up Hierarchical Clustering** algorithm. It starts by treating each data point as an individual cluster and repeatedly merges the two closest clusters until all data points become one cluster. The final result is represented using a **Dendrogram**.

Example Dataset

Suppose five students have the following marks:

Student	Marks
A	20
B	22
C	50
D	80
E	82

Initially, every student is treated as a separate cluster.
{A} {B} {C} {D} {E}

Algorithm Steps

Step 1: Treat Every Data Point as a Separate Cluster

Initially, each student forms one cluster.
{A} {B} {C} {D} {E}

Step 2: Find the Closest Clusters

Calculate the distance between all clusters.
The smallest distances are:

- A and B = 2
- D and E = 2

Merge them.
(A,B) C (D,E)

Step 3: Update the Clusters

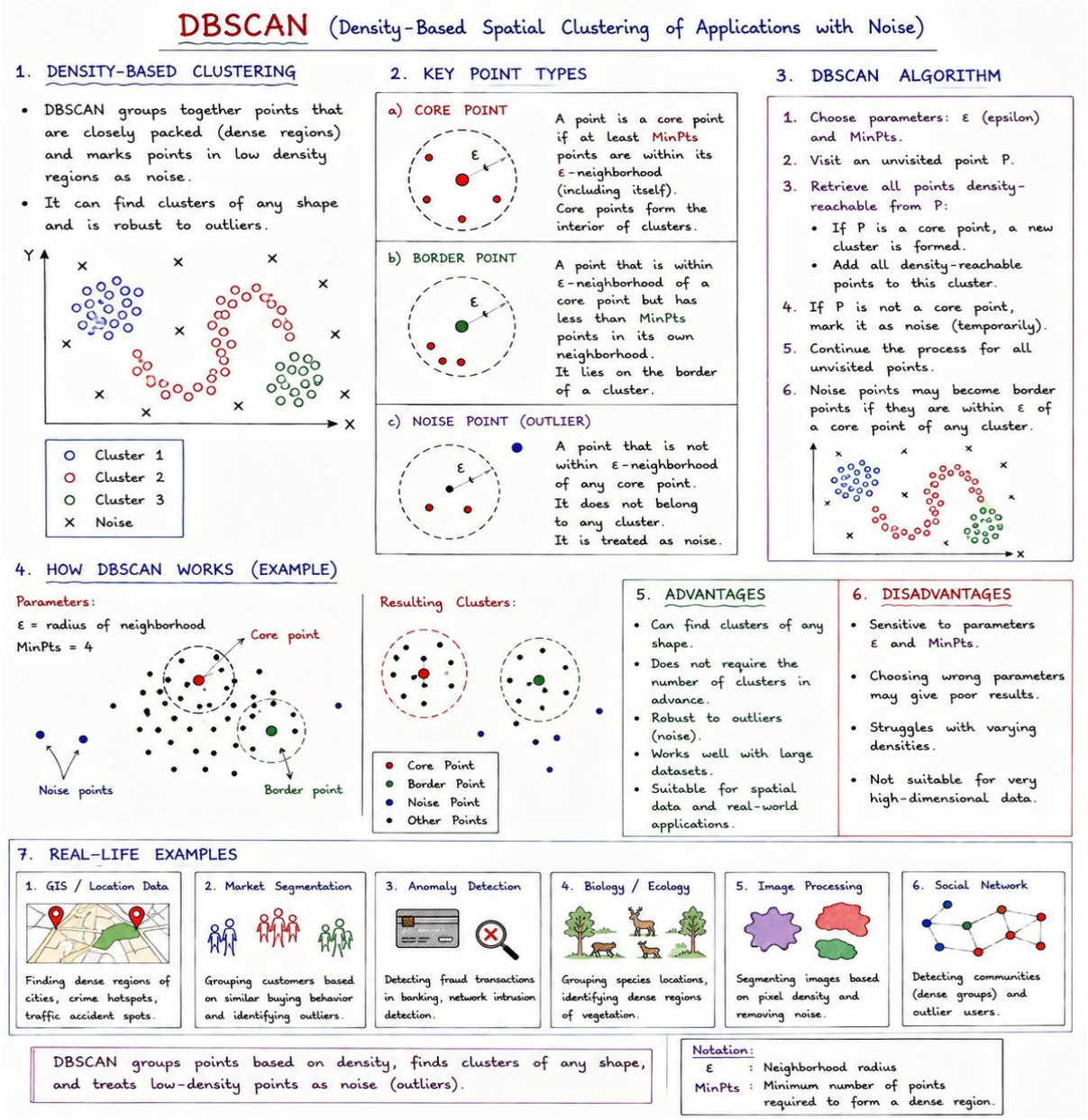
Now calculate the distance again.
The nearest cluster to **(A,B)** is **C**.
Merge them.
(A,B,C) (D,E)

Step 4: Repeat the Process

Merge the remaining two clusters.
(A,B,C,D,E)
Now only one cluster remains, so the algorithm stops.

Applications

- Customer Segmentation
- Document Clustering
- Medical Data Analysis
- Image Classification
- Social Network Analysis



Definition

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** is an **Unsupervised Machine Learning** algorithm that groups data points based on **density**.
- It forms clusters in dense regions and identifies isolated data points as **Noise (Outliers)**. Unlike K-Means, DBSCAN **does not require the number of clusters (K) to be specified in advance**.

Example Dataset

Suppose **10 students** are standing in a playground.

```
Group 1                Group 2

A  B  C  D  E          F  G  H  I


                  J
          (Standing Alone)
```

Assume:

- ε (Epsilon) = 2
- MinPts = 4

Types of Points

1. Core Point

A **Core Point** has at least **MinPts** neighboring points within the ε radius.

Example:

Student **D** has nearby students **A, B, C, and E**.

Therefore, **D is a Core Point**.

2. Border Point

A **Border Point** has fewer neighbors than MinPts but lies within the neighborhood of a Core Point.

Example:

Student **E** is close to **D**, so it belongs to the cluster but is on its boundary.

Therefore, **E is a Border Point**.

3. Noise Point (Outlier)

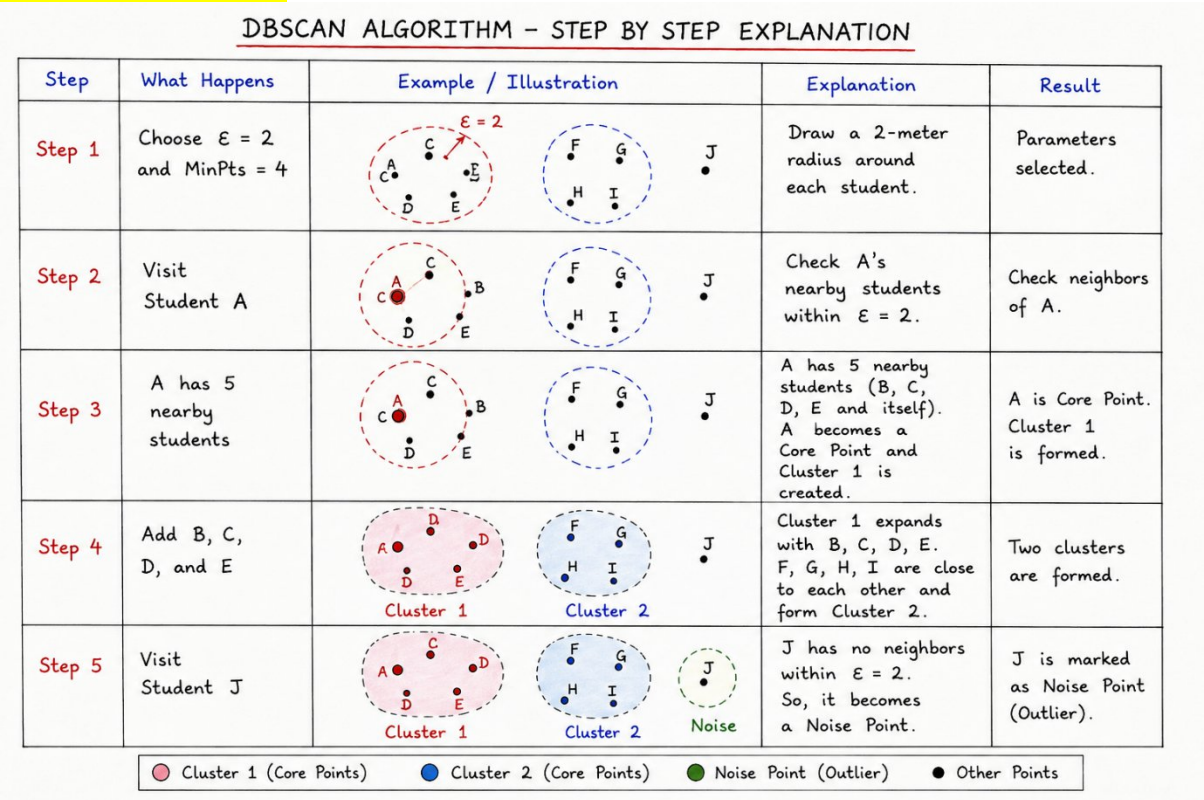
A **Noise Point** is far away from every cluster and has no nearby neighbors.

Example:

Student **J** is standing alone.

Therefore, **J is a Noise Point (Outlier).**

ALGORITHM STEPS :



Advantages

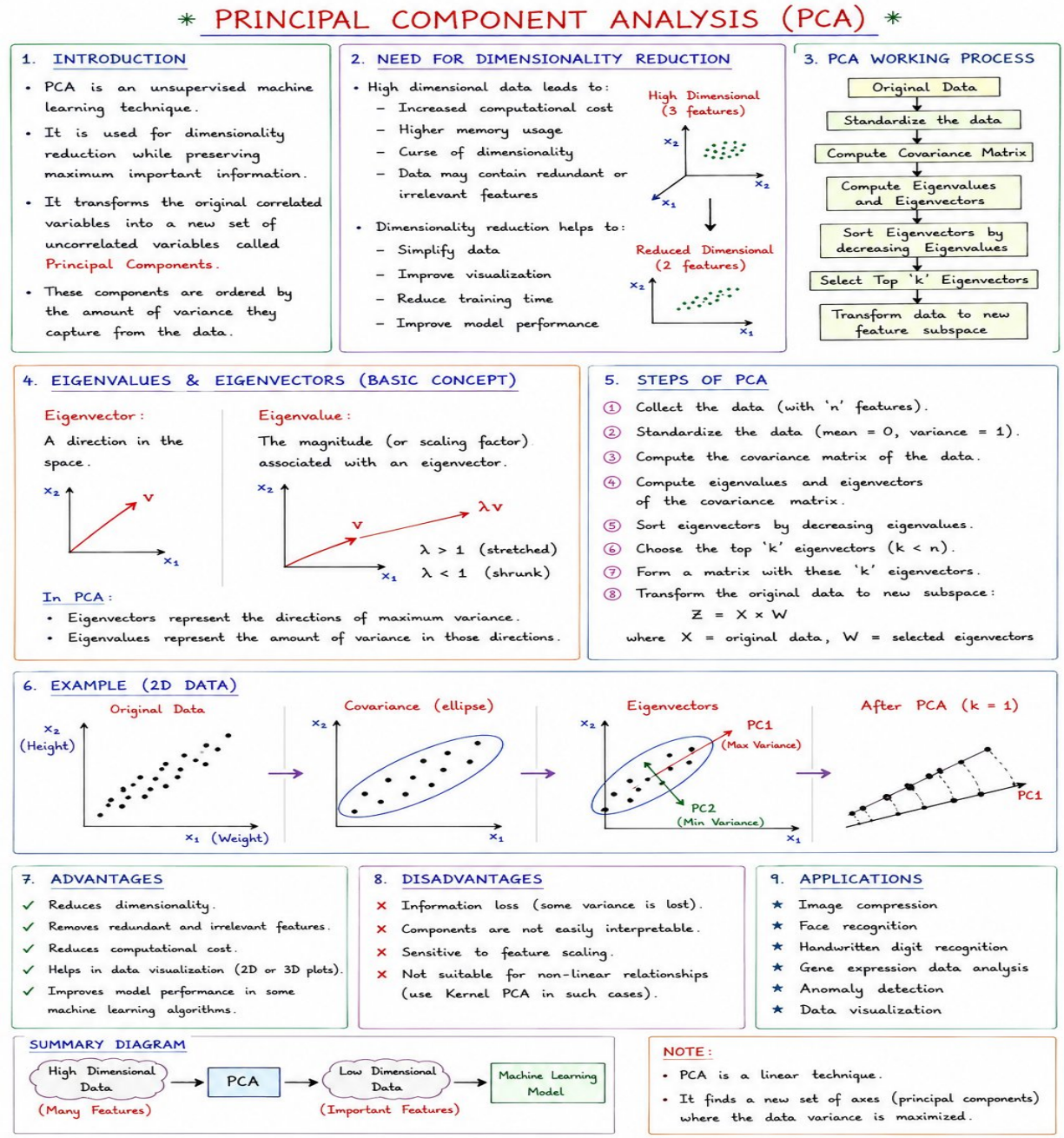
- Finds clusters of **any shape**.
- No need to specify the number of clusters (**K**).
- Detects **Noise (Outliers)** automatically.
- Works well for spatial and geographical data.

Disadvantages

- Sensitive to **ε** and **MinPts** values.
- Difficult to cluster data with varying densities.
- Not suitable for very high-dimensional datasets.

Real-Life Applications:

- **Fraud Detection:** Detects unusual bank transactions.
- **Google Maps:** Identifies traffic hotspots.
- **Customer Segmentation:** Groups customers based on buying behavior.
- **Image Processing:** Segments images into meaningful regions.
- **Social Networks:** Finds communities with similar interests



Steps of PCA (Principal Component Analysis)

Example Dataset
Suppose a teacher collects the details of **5 students**.

Student	Height (cm)	Weight (kg)	Marks
A	150	45	70
B	155	48	75
C	160	52	80
D	165	55	85
E	170	60	90

Here,

- **Height**
- **Weight**
- **Marks**

are the **3 features (dimensions)**.
PCA reduces these **3 features** into **2 important features (Principal Components)** while preserving most of the information.

Step 1: Collect the Data
First, collect the dataset containing multiple features.
Example

Student	Height	Weight	Marks
A	150	45	70
B	155	48	75
C	160	52	80
D	165	55	85
E	170	60	90

This is called the **Original Dataset**.

Step 2: Standardize the Data
Since Height, Weight, and Marks have different units, convert them to the **same scale**.
This ensures that one feature does not dominate the others.
Example
Before Standardization
Height = 150–170

Weight = 45–60

Marks = 70–90
After Standardization
All features have

Mean = 0

Variance = 1
Now all features are equally important.

Step 3: Compute the Covariance Matrix
Find the relationship between the features.
Example

- Height ↑ → Weight ↑
- Height ↑ → Marks ↑

This means these features are **positively correlated**.

The covariance matrix identifies which features move together.

Step 4: Compute Eigenvalues and Eigenvectors
PCA calculates:
Eigenvector
It shows the **direction** of maximum variation.
Eigenvalue
It shows **how much information (variance)** is present in that direction.
Example
Suppose we get

Component	Eigenvalue
PC1	8.5
PC2	2.0
PC3	0.5

Here,
PC1 contains the **maximum information**.

Step 5: Sort Eigenvalues
Arrange the eigenvalues from **highest to lowest**.

Principal Component	Eigenvalue
PC1	8.5
PC2	2.0
PC3	0.5

PC1 is the most important because it preserves the maximum variance.

Step 6: Select Top 'k' Components
Choose the components with the **largest eigenvalues**.
Suppose
Original Features = **3**
Choose
k = 2
Now only

- PC1
- PC2

are selected.

Step 7: Form the Feature Matrix
Create a new feature matrix using the selected eigenvectors.
This matrix contains only the important components.

Step 8: Transform the Data
Project the original data onto the selected principal components.
Before PCA
Height

Weight

Marks
3 Features
↓
After PCA
PC1

PC2

2 Features

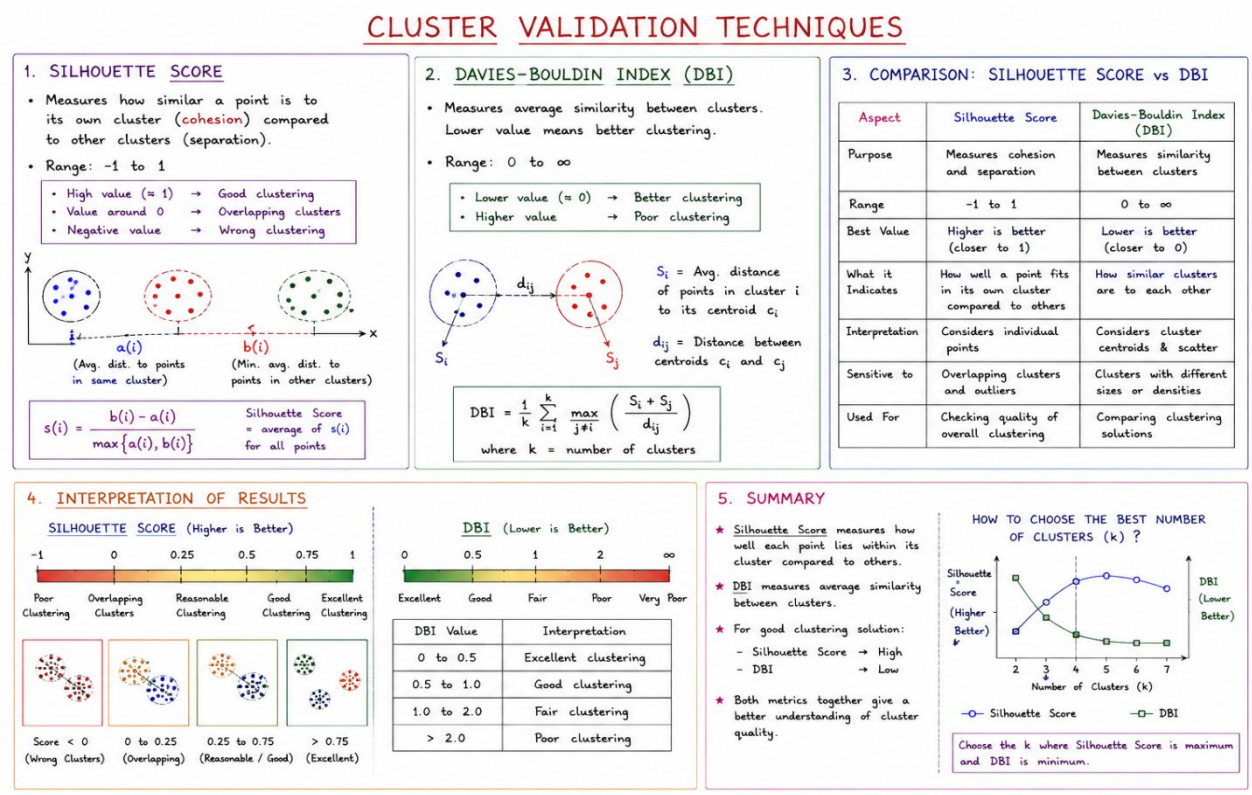
The dataset size is reduced while retaining most of the important information.

Easy Real-Life Example

Suppose a college stores student information:

- Name
- Height
- Weight
- Marks
- Attendance
- Sports Score

Instead of using all **6 features**, PCA selects only the **2 or 3 most important features**.
This reduces storage space and speeds up machine learning algorithms while preserving the most useful information.



Cluster Validation Techniques are used to **measure the quality of clusters** formed by clustering algorithms. They help determine whether the clusters are **well separated and compact**. Two commonly used techniques are:

1. **Silhouette Score**
2. **Davies-Bouldin Index (DBI)**

1. Silhouette Score

Definition

The **Silhouette Score** measures how similar a data point is to its **own cluster (cohesion)** compared to **other clusters (separation)**.

- **Range: -1 to +1**

The formula is:

$$S(i) = \frac{b(i) - a(i)}{\max\left(a(i)\, b(i)\right)}$$

Where:

- **a(i)** = Average distance of a point from other points in the **same cluster**.
- **b(i)** = Average distance of a point from the **nearest neighboring cluster**.

Explanation of the Figure

In the figure,

There are **three clusters**.

- **Blue Cluster**
- **Red Cluster**
- **Green Cluster**

For one blue data point,

- **a(i)** is the average distance to other **blue points**.
- **b(i)** is the distance to the nearest **red cluster**.

If the point is much closer to its own cluster than to other clusters, the Silhouette Score will be high.

Interpretation

Silhouette Score	Meaning
+1 (≈1)	Excellent clustering
0	Clusters overlap
Negative (<0)	Wrong clustering

Real-Life Example

Suppose a college groups students into:

- CSE
- ECE
- Mechanical

If every CSE student is much closer to other CSE students than to ECE students, the **Silhouette Score will be high**, indicating good clustering.

2. Davies-Bouldin Index (DBI)

Definition

The **Davies-Bouldin Index (DBI)** measures the average similarity between clusters.

It compares:

- Distance within the cluster
- Distance between clusters

Lower DBI means better clustering.

Explanation of the Figure

The figure shows

Two clusters

- Blue Cluster
- Red Cluster

Where

- **Si** = Average distance of points from the centroid of Cluster 1.
- **Sj** = Average distance of points from the centroid of Cluster 2.
- **dij** = Distance between the two centroids.

If clusters are

- Compact (small Si and Sj)
- Far apart (large dij)

Then DBI becomes **small**, which indicates good clustering.

Interpretation

DBI Value	Meaning
0 – 0.5	Excellent clustering
0.5 – 1.0	Good clustering

DBI Value	Meaning
1 – 2	Fair clustering
> 2	Poor clustering

Real-Life Example
Suppose a supermarket groups customers into:

- Electronics Buyers
- Grocery Buyers

If both customer groups are clearly separated, the **DBI value will be low**, indicating good clustering.

3. Comparison: Silhouette Score vs Davies-Bouldin Index

Feature	Silhouette Score	Davies-Bouldin Index
Purpose	Measures cluster quality	Measures cluster similarity
Range	-1 to +1	0 to ∞
Best Value	Higher is better (close to +1)	Lower is better (close to 0)
Measures	Cohesion and Separation	Cluster compactness and separation
Interpretation	High value = Good clusters	Low value = Good clusters

4. Interpretation of Results
The figure shows how to interpret the scores.
Silhouette Score

Score	Interpretation
< 0	Wrong Clustering
0 – 0.25	Overlapping Clusters
0.25 – 0.75	Good Clustering
> 0.75	Excellent Clustering

Davies-Bouldin Index

DBI Value	Interpretation
0 – 0.5	Excellent
0.5 – 1	Good
1 – 2	Fair
> 2	Poor

5. Summary (From the Figure)

- **Silhouette Score** measures how well each data point fits within its own cluster.
- **Higher Silhouette Score** indicates better clustering.
- **Davies-Bouldin Index** measures the similarity between clusters.
- **Lower DBI** indicates better clustering.
- The **best clustering solution** is achieved when:
    - **Silhouette Score is maximum (close to +1).**
    - **DBI is minimum (close to 0).**

Real-Life Example
Suppose a shopping mall divides customers into three groups:

- Electronics Customers
- Fashion Customers
- Grocery Customers

Good Clustering

- Customers in each group have similar buying habits.
- The groups are clearly separated.

Result:

- **Silhouette Score = 0.82 (High)** ✓
- **DBI = 0.35 (Low)** ✓

This means the clustering is of **high quality**