Subject: U24CST362 – Machine Learning Fundamentals

Course: B. Sc Computer Science with specialization in Artificial Intelligence and Machine Language

Prepared By: SAKSHI PANDEY

UNIT I : Introduction

Learning Objectives

- Understand the foundation of machine learning
- Explore how ML evolved from traditional computing
- Identify types of machine learning
- Learn real-world applications
- Know basic tools for ML development

Topics

- Definition and Evolution of Machine Learning
- Types of Machine Learning
- Components of a Learning System
- Key Steps in Designing a Learning System
- Challenges in Machine Learning
- Difference Between AI, ML, Deep learning
- Applications of Machine Learning in Real world
- Data Preprocessing Basics
- Overview of ML Tools
- Role of Data in Machine Learning

What is Machine Learning ?

Definition:

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Arthur Samuel (1959) defined ML as:

"The field of study that gives computers the ability to learn without being explicitly programmed."

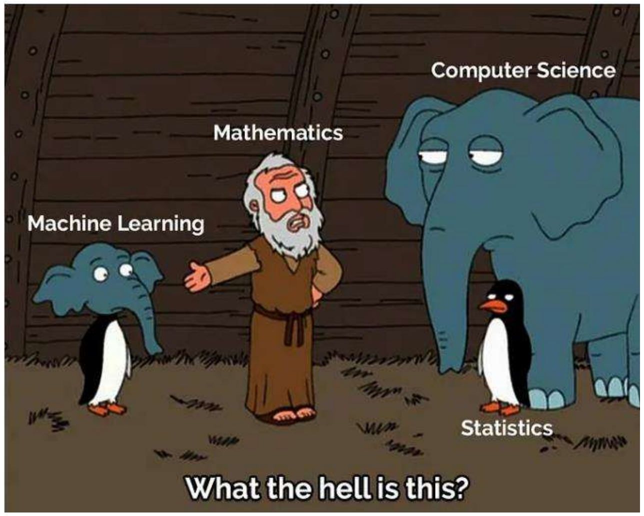
Example:

Email spam filters

Movie recommendations on Netflix

Fraud detection in banking

A Brief History/ Evolution of Machine Learning

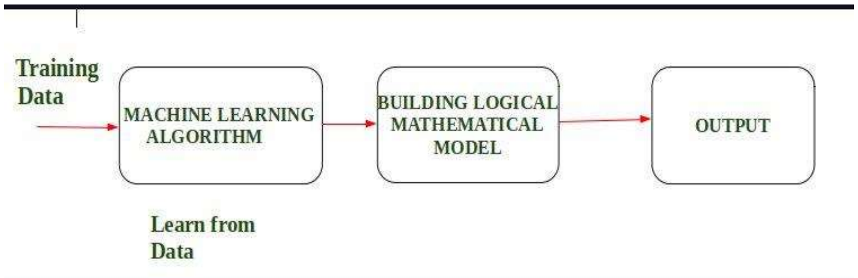


ERA	MILESTONE
1950s	Turing Test, Alan Turing proposes "learning machines"
1957	Perceptron introduced by Frank Rosenblatt
1967	Nearest Neighbor algorithm developed
1980s	Emergence of neural networks (backpropagation)
1997	IBM's Deep Blue defeats chess champion Garry Kasparov
2006	"Deep Learning" term popularized by Geoffrey Hinton
2012–Present	ImageNet, AlphaGo, GPT, Self-driving cars

Traditional Programming vs ML

Aspect	Traditional Programming	Machine Learning
Approach	Rules and logic are hardcoded	Learns from data
Input	Data + Program	Data + Output
Output	Output	Program (Model)
Flexibility	Fixed for all inputs	Adapts to new data
Example	If age > 18 then eligible	Learns eligibility from examples

Key Components of ML System



Core Components:

Data (examples or observations)

Model (the pattern-finding logic)

Learning Algorithm (optimizes the model to fit the data)

Design of Learning System

Step 1: Choosing the Training Experience

Select meaningful and high-impact training data.

Example: In chess, data helps evaluate better moves.

Step 2: Choosing the Target Function

Define the goal or function the model should learn.

Example: In chess, `NextMove()` function predicts the best legal move.

Step 3: Representing the Target Function

How the target function is structured internally

Example: Out of 4 legal chess moves, choose the one with highest win probability.

Step 4: Choosing the Function Approximation Algorithm
Use learning algorithms to approximate the target function.
Example: ML model fails initially but improves with repeated play.
Step 5: Final Design of the Learning System
Model is finalized after:
Experiencing many examples
earning from past mistakes
Example: **IBM's Deep Blue** defeated Garry Kasparov by learning from past

Types of Machine Learning

1. Supervised Learning

Trains on labeled data (input + output).

Goal: Predict outputs from known inputs.

Common Tasks: Classification (spam detection), Regression (house price prediction).

Examples: Decision Trees, SVM, Linear Regression.

2. Unsupervised Learning

Trains on unlabeled data (only inputs).

Goal: Discover hidden patterns or groupings.

Common Tasks: Clustering (customer segmentation), Association (market basket analysis).

Examples: K-Means, DBSCAN, Apriori.

3. Reinforcement Learning

Goal: Learn through **trial-and-error** with rewards/penalties.

Agent interacts with environment, improves performance over time.

Examples: Q-Learning, Deep Q-Networks.

Use Cases: Robotics, Game AI, Self-driving cars.

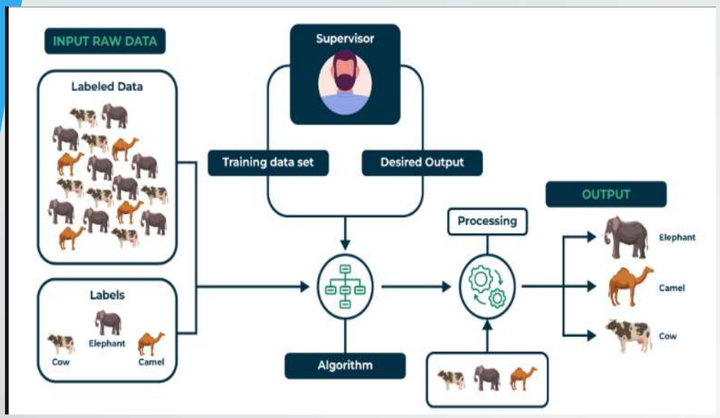
4. Semi-Supervised Learning

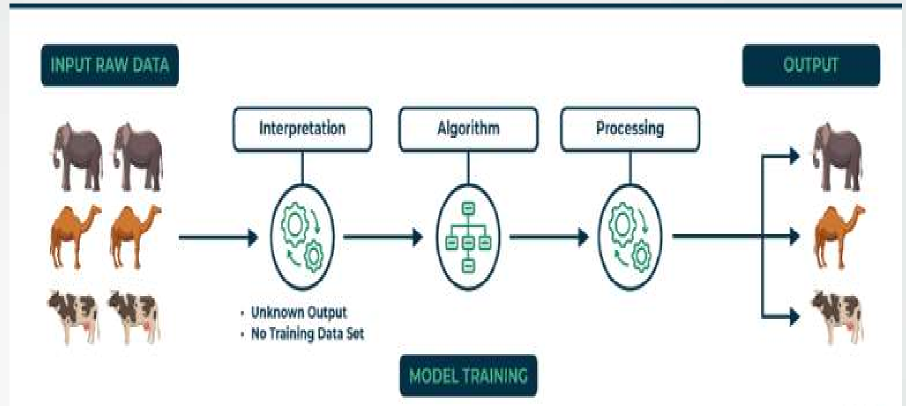
Combines **small labeled** + **large unlabeled** data.

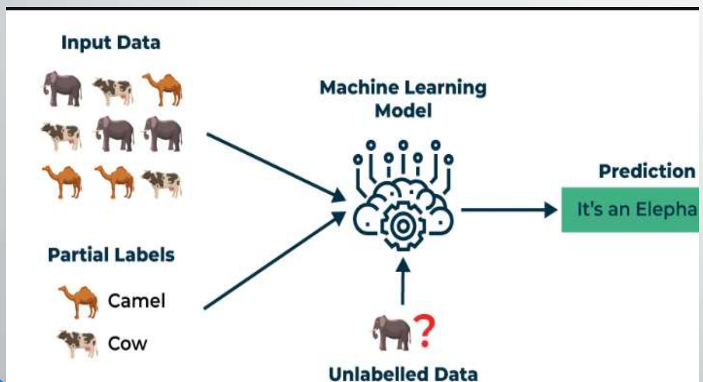
Useful when labeling data is expensive.

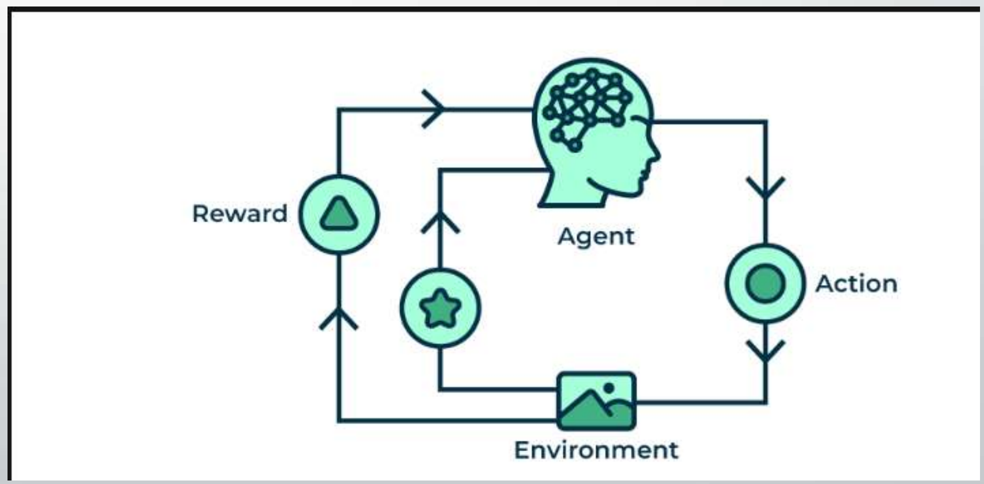
Methods: Label Propagation, Self-training, GANs.

Applications: NLP, Image Recognition, Healthcare.


INPUT RAW DATA
Labeled Data
Labels
Cow
Elephant
Camel
Supervisor
Training data set
Desired Output
Algorithm
Processing
OUTPUT
Elephant
Camel
Cow


INPUT RAW DATA
Interpretation
Algorithm
Processing
OUTPUT
• Unknown Output
• No Training Data Set
MODEL TRAINING


Input Data
Partial Labels
Camel
Cow
Machine Learning Model
Unlabelled Data
Prediction
It's an Elepha


Reward
Agent
Action
Environment

Applications of Machine Learning

Domain	Use Case
Healthcare	Disease prediction, medical imaging
Finance	Credit scoring, fraud detection
Retail	Recommendation systems
Manufacturing	Predictive maintenance
Agriculture	Crop yield prediction
Security	Intrusion detection, facial recognition
Education	Student performance prediction

Real-World Examples

Netflix / YouTube	Recommender systems based on user history
Google Translate	NLP models trained on multilingual data
Amazon Alexa / Siri	Voice recognition using deep learning
Self-Driving Cars	Computer vision and reinforcement learning
Facebook / Instagram	Face recognition and content filtering
Stock Market	Algorithmic trading and price prediction

Tools & Libraries for Machine Learning

Language	Why it's Used in ML
Python	Most popular ML language; simple syntax, huge library support (NumPy, scikit-learn, TensorFlow, PyTorch).
R	Best for statistical modeling and data visualization. Preferred by statisticians.
Java	Used in large-scale, high-performance systems (e.g., Apache Spark MLlib).
Julia	Fast mathematical computation, good for numerical ML tasks and large datasets.

Platform	Description
Jupyter Note book	Interactive coding environment; ideal for step-by-step ML model development.
Google Colab	Free cloud-based Jupyter Notebook with free GPU/TPU support. Ideal for students & researchers.
Azure ML	Microsoft's cloud-based ML platform with automation, deployment, and scalability support.
AWS SageMaker	Amazon's fully-managed ML service that allows building, training, and deploying ML models at scale.

Library	Purpose / Use Case	Key Features / Highlights
NumPy	Numerical computations	Fast array operations, linear algebra, used in TensorFlow, PyTorch
Pandas	Data manipulation & preparation	DataFrames, handling missing data, filtering, grouping
Matplotlib	Data visualization	2D plotting, histograms, bar charts, line graphs
SciPy	Scientific computing	Optimization, signal processing, linear algebra
Scikit-Learn	Classical ML models	Regression, classification, clustering, model selection
Theano	Symbolic math computation	Deep learning backend, supports GPU/CPU
TensorFlow	Deep learning, numerical computation	Tensor-based computation, used for AI/ML production systems
Keras	High-level neural networks API	Easy to use, runs on TensorFlow, quick prototyping
PyTorch	Deep learning with dynamic graphs	Tensors, GPU acceleration, NLP/CV applications

UNIT II

Learning Objectives

Explain the principles of supervised learning and the role of regression in predictive modeling.

Differentiate between simple and multiple linear regression, including their assumptions and applications.

Apply polynomial regression to capture nonlinear relationships and **analyze** the risk of overfitting.

Understand the concept of regularization and **implement** Ridge (L2) and Lasso (L1) regression techniques.

Compare the strengths, limitations, and use cases of linear, Ridge, and Lasso regression models.

Evaluate regression models using appropriate performance metrics and interpret their results effectively.

Topics of Unit II

Simple Linear Regression
Multiple Linear Regression
Polynomial Regression
Ridge Regression
Lasso Regression
Comparison of Regression Techniques
Overfitting and Underfitting
Bias, Variance and Bias-Variance Tradeoff
Training data, Testing data, Datasets
Model Evaluation Metrics for Regression
Cross – Validation and Model Selection Techniques
Bagging Concept and Example with Decision Trees
Ensemble Methods: Boosting vs Bagging Overview

SIMPLE LINEAR EGRESSION

Simple Linear Regression aims to describe how one variable i.e the dependent variable changes in relation with reference to the independent variable.

For **example** consider a scenario where a company wants to predict sales based on advertising expenditure. By using simple linear regression the company can determine if an increase in advertising leads to higher sales or not.

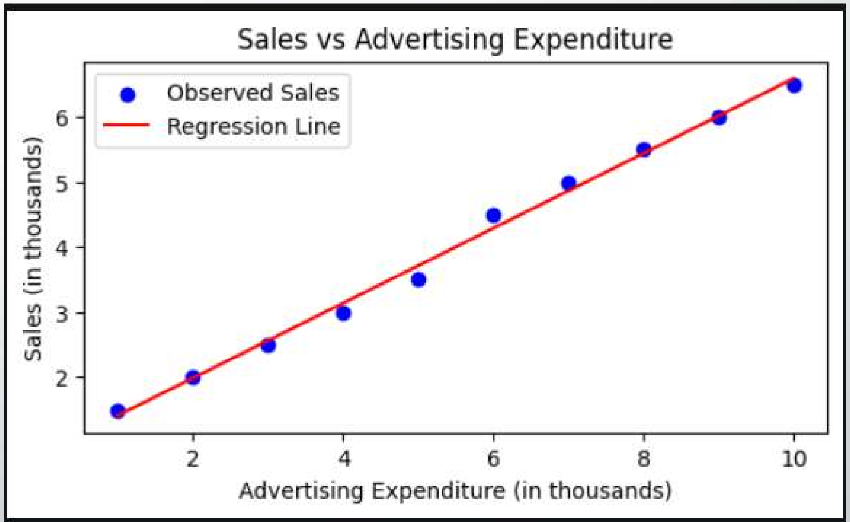
Syntax:

$$y = \theta x + b$$

where,

θ *- It is the model weights or parameters*

b *- It is known as the bias.*



Simple Linear Regression

MULTIPLE LINEAR REGRESSION

Multiple Linear Regression extends simple linear regression by using **two or more independent variables** to predict the dependent variable.
It helps when the outcome depends on several factors.

Example:
A company wants to predict house prices based on size, number of bedrooms, location, and age of the house.

Syntax:

$y=\theta_1x_1+\theta_2x_2+\theta_3x_3+\ldots+\theta_nx_n+b$

Where:

$x_1,x_2,\ldots,x_n$ = independent variables

$\theta_1,\theta_2,\ldots,\theta_n$ = model weights/parameters for each predictor

b = bias term

POLYNOMIAL LINEAR REGRESSION

Polynomial Regression models the relationship between the dependent and independent variable(s) as an **nth-degree polynomial**. It is useful when data shows a **non-linear trend** that a straight line cannot capture.

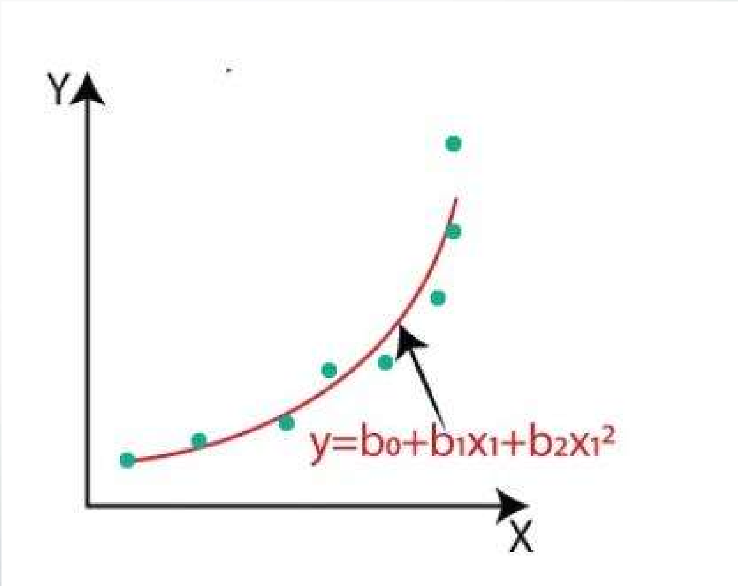
Example:
A retail company predicts sales revenue from advertising expenditure. Sales first increase rapidly but then saturate (non-linear curve).

Syntax:

$y=\theta o+\theta 1x+\theta 2x2+\theta 3x3+\ldots+\theta nxny = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n y=\theta o +\theta 1\ x+\theta 2\ x2+\theta 3\ x3+\ldots+\theta n\ xn$

Where:

Higher powers of xxx help capture **curvature** in the data.



RIDGE REGRESSION

Ridge Regression is a **regularized version of linear regression**.
It adds a **penalty term (L2 regularization)** to reduce overfitting by shrinking large coefficients.

Example:
When predicting house prices with many correlated features (like size in sq.ft and number of rooms), ridge regression helps avoid unstable coefficients.

Syntax:

$y=\theta_1x_1+\theta_2x_2+\ldots+\theta_nx_n+b$

with penalty term added to cost function

$$Loss = \text{MSE} + \lambda \sum_{i=1}^{n} w_i^2$$

Where:

λ = regularization strength (controls shrinkage of coefficients)

LASSO REGRESSION

Lasso (Least Absolute Shrinkage and Selection Operator) Regression is also a **regularized linear regression** technique.
It adds a **penalty term (L1 regularization)** that can shrink some coefficients to **zero**, performing **feature selection**.

Example:
If predicting sales from 50 different factors, but only 5 are important, Lasso automatically reduces the weights of unimportant predictors to zero.

Syntax:

$y=\theta_1x_1+\theta_2x_2+\ldots+\theta_nx_n+by = \theta_1x_1 + \theta_2x_2 + \ldots + \theta_nx_n + by=\theta_1 x_1 +\theta_2 x_2 +\ldots+\theta_n x_n +b$

with penalty term:

$$Loss = \text{MSE} + \lambda \sum_{i=1}^{n} |w_i|$$

Where:

- λ= controls how strongly coefficients are pushed toward zero.

Comparison of Regression Techniques

Regression Type	Purpose	When to Use	Advantages	Drawbacks
Simple Linear Regression	Predicts a dependent variable using one independent variable	When relationship is approximately linear and only one predictor is involved	Easy to interpret, fast	Cannot handle multiple predictors, limited for complex data
Multiple Linear Regression	Predicts using two or more independent variables	When multiple features affect the outcome	Captures effect of multiple predictors, widely used	Assumes linearity, sensitive to multicollinearity
Polynomial Regression	Extends linear regression by adding polynomial terms of predictors	When relationship is non-linear (e.g., curved trends)	Can model non-linear relationships	Risk of overfitting with high-degree polynomials, less interpretable
Ridge Regression (L2 Regularization)	Linear regression with penalty on squared coefficients	When multicollinearity exists or to prevent overfitting	Shrinks coefficients, handles multicollinearity well	Coefficients never become zero (all features kept)
Lasso Regression (L1 Regularization)	Linear regression with penalty on absolute values of coefficients	When multicollinearity exists or to prevent underfitting	can shrink some coefficients to **zero**, performing **feature selection**.	When feature selection is important

Aspect	Overfitting	Underfitting
Definition	Model learns the training data too well, including noise and random fluctuations	Model fails to learn the underlying pattern in the data
Cause	Model is too complex (too many parameters, features, or high-degree polynomial)	Model is too simple (few parameters, weak assumptions, not enough features)
Training Accuracy	Very high	Low
Testing/Validation Accuracy	Low (poor generalization)	Low (fails on both training & testing data)
Model Behavior	Memorizes training data instead of generalizing	Oversimplifies relationships, misses important patterns
Example	Using a 10th-degree polynomial for data that follows a straight line	Using a straight line for data that actually follows a curve
Solution	Reduce model complexity, use regularization (L1/L2), more training data, cross-validation	Increase model complexity, add more features, reduce bias, train longer

Bias, Variance and Bias-Variance Tradeoff

Bias

Error due to overly simple assumptions in the model

Leads to **underfitting**

Example: Using a straight line for curved data

Variance

Error due to model's sensitivity to small fluctuations in training data

Leads to **overfitting**

Example: Using a high-degree polynomial that fits noise

Bias-Variance Tradeoff

Goal: Find a balance between bias and variance

Low Bias + Low Variance = Optimal Model

Controlled using: Regularization, Cross-validation, Proper model selection

Training data, Testing data, Datasets

Dataset

A collection of examples used to build and evaluate machine learning models.

Training Data

Used to **teach** the model

Model learns patterns, parameters, and relationships

Testing Data

Used to **evaluate** the trained model

Measures accuracy and generalization on unseen data

Validation Data

Helps in **tuning hyperparameters** and avoiding overfitting

Model Evaluation Metrics for Regression

1. Mean Absolute Error (MAE)

Average of absolute differences between predicted and actual values

Easy to interpret, less sensitive to outliers

2. Mean Squared Error (MSE)

Average of squared differences

Penalizes large errors more than small ones

3. Root Mean Squared Error (RMSE)

Square root of MSE

Same unit as target variable, interpretable

4. R^2 Score (Coefficient of Determination)

Measures how well regression line fits the data

Ranges from 0 to 1 (closer to 1 = better fit)

Cross – Validation and Model Selection Techniques

- **Cross-Validation (CV)**

Technique to **assess model performance** on unseen data

Prevents **overfitting & underfitting**

Common methods:

- **Hold-Out** → Split into train/test sets
- **k-Fold CV** → Data split into *k* parts, rotate train/test
- **Leave-One-Out CV (LOOCV)** → Each sample tested once
- **Stratified k-Fold** → Maintains class balance in folds

- **Model Selection Techniques**

Grid Search → Exhaustively searches best hyperparameters

Random Search → Randomly samples hyperparameters

Bayesian Optimization → Probabilistic approach to tune parameters efficiently

Regularization (Ridge, Lasso, ElasticNet) → Avoids overfitting

Early Stopping → Stops training when performance stops improving

Bagging Concept and Example with Decision Trees

- Concept

Ensemble learning method that combines predictions from multiple models.

Creates multiple subsets of training data using bootstrap sampling (sampling with replacement).

Each subset trains a separate model (e.g., Decision Tree).

Final prediction:

Regression → Average of predictions

Classification → Majority voting

- Why Bagging?

Reduces variance

Helps avoid overfitting in high-variance models (like decision trees).

Improves stability & accuracy.

- Example with Decision Trees

Given dataset → Perform bootstrap sampling to create multiple datasets.

Train Decision Tree on each bootstrap dataset.

Combine predictions:

If classifying → take majority vote

If predicting numbers → take average

Famous Algorithm: Random Forest = Bagging + Random Feature Selection

Key Idea: Bagging = “Train many models on random subsets → combine results for more accurate & stable predictions.”

Ensemble Methods: Boosting vs Bagging Overview

Aspect	Bagging	Boosting
Full Form	Bootstrap Aggregating	Adaptive Boosting
Approach	Trains models independently on random subsets of data	Trains models sequentially, each new model focuses on errors of the previous one
Sampling	Uses bootstrap sampling (sampling with replacement)	Uses the entire dataset, but re-weights misclassified/poorly predicted samples
Focus	Reduces variance	Reduces bias
Model Combination	Combines predictions by averaging (regression) or majority vote (classification)	Combines predictions by weighted voting or weighted average
Strength	Prevents overfitting in high-variance models (e.g., decision trees)	Builds a strong learner from many weak learners
Training	Models are trained in parallel	Models are trained sequentially
Stability	More stable, less prone to overfitting	More prone to overfitting if too many weak learners are added
Famous Algorithms	Random Forest	AdaBoost, Gradient Boosting, XGBoost, LightGBM

CAT –1 ASSESSEMENT TIME

TAKE TWO PAGE AND PEN

PUT YOUR PHONE AND LAPTOP IN YOUR BAG

3 MINUTES FOR THIS

CASE STUDY UNIT I

Case Study 1: Predicting Student Performance

A university wants to predict the performance of students in final exams based on their attendance, assignment submission, and internal test marks.

1. Identify the type of machine learning suitable for this problem.

2. Justify why this type of learning is appropriate in the given scenario.

3. List the steps involved in designing such a learning system.

4. Mention at least two possible challenges in preprocessing the data.

5. Suggest one evaluation metric that can be used to measure the model's performance and explain briefly.

Case Study 2: Fraud Detection in Banking

A bank wants to detect fraudulent credit card transactions in real-time.

1. Identify the type of Machine Learning technique suitable for this problem.

2. Justify why this type of learning is appropriate in fraud detection.

3. How would you handle the issue of imbalanced data during training? Give at least one technique.

4. Suggest at least two ML algorithms or tools that can be used for fraud detection in real-world banking systems.

5. Mention one possible challenge in deploying a fraud detection system in real-time and suggest a possible solution.

CASE STUDY UNIT II

Case Study 1: Predicting Housing Prices

A real estate company wants to predict house prices. Initially, they use only the size of the house (in sq. ft.) as a predictor (Simple Linear Regression). Later, they expand the model by including the number of bedrooms, location index, and age of the house (Multiple Linear Regression).

1. Identify which type of regression is used when only one predictor variable (house size) is considered.
2. Explain why Multiple Linear Regression might give better predictions compared to Simple Linear Regression in this scenario.
3. Discuss one case where Simple Linear Regression may be more suitable than Multiple Linear Regression.
4. Mention one challenge that arises when using Multiple Linear Regression with many predictors.
5. Suggest an evaluation metric (e.g., RMSE, R^2) to compare the performance of both models and explain briefly why it is useful.

Case Study 2: Sales Forecasting with Polynomial Regression

A retail company tracks the relationship between advertising expenditure and sales revenue. A linear model fails to capture the curve in the data (increasing at first, then saturating).

1. What limitation of Linear Regression makes it unsuitable for this problem?
2. Explain how Polynomial Regression can model the non-linear relationship between advertising expenditure and sales.
3. Give one real-world example (other than sales) where Polynomial Regression would be more effective than Linear Regression.
4. Mention one drawback of using very high-degree Polynomial Regression.
5. Suggest an appropriate evaluation metric to compare the performance of linear and polynomial regression models and explain why it is suitable.

SAMPLE QUESTIONS

1. Explain Machine Learning and difference between AI,ML and Deep Learning.
2. Explain types of Machine Learning and comparison between them
3. Explain Tools, Platforms and Libraries used for Machine Learning.
4. Explain Application of Machine Learning with real world examples.
5. Explain key components of machine learning and design steps of learning system with example.

1. Explain Simple Linear Regression with equation and example.
2. Compare Simple, Multiple and Polynomial Regression.
3. Compare Ridge and Lasso Regression and what is purpose of using them.
4. Explain Underfitting and Overfitting with proper reason of their cause and how to overcome them.
5. Explain Bagging, Boosting and Cross-Validation.