Perfect 👍 Let's go step by step — I'll explain **each unit and each topic** in **very easy, simple language**, with **detailed points** under each heading.
This is based on your **SPPU Machine Learning (2019 Course, IT – TE)** syllabus.

🎯 UNIT I – INTRODUCTION TO MACHINE LEARNING (6 Hours)

🧠 1. What is Machine Learning?

Simple meaning:
Machine Learning (ML) means teaching computers to **learn from data** and **make decisions** without being directly programmed.

Example:
When you show a computer many pictures of cats and dogs, it learns the difference — then can tell whether a new image is a cat or dog.

📘 2. Definition

Machine Learning is a **subset of Artificial Intelligence (AI)** that allows computers to **learn patterns** from past data and **make predictions or decisions** without being explicitly programmed for each task.

🌍 3. Real-Life Applications of ML

1. 🎵 **Recommendation systems** – YouTube, Netflix suggest videos based on what you watch.
2. 📱 **Voice assistants** – Alexa, Siri understand and respond to speech.
3. 📧 **Spam filtering** – Gmail detects spam emails automatically.
4. 🚗 **Self-driving cars** – Learn to drive using camera and sensor data.
5. 💰 **Fraud detection** – Banks detect fake transactions using ML models.

📊 4. Learning Tasks

(A) Descriptive Tasks

- Aim: Describe or find patterns in existing data.
- Example: Grouping customers by buying habits (clustering).

(B) Predictive Tasks

- Aim: Predict future outcomes based on data.
- Example: Predicting house prices or exam marks.

🎓 5. Types of Learning

Type	Description	Example
Supervised Learning	Data has input (features) and output (labels). The model learns from both.	Predicting marks from study hours.
Unsupervised Learning	Only input data is given, no output labels. Model finds patterns itself.	Grouping similar customers (clustering).
Semi-Supervised Learning	Some data is labeled, some is not.	Face recognition when few faces are labeled.
Reinforcement Learning	System learns by trial and error to get rewards.	Teaching a robot to walk or a game AI to win.

💡 6. Features

(A) Types of Data

Type	Meaning	Example
Qualitative (Categorical)	Data that describes qualities or categories.	Gender, color, city
Quantitative (Numerical)	Data with numbers that can be measured.	Height, weight, marks

---

(B) Scales of Measurement

Scale	Meaning	Example
Nominal	Just names or categories, no order.	Male/Female
Ordinal	Ordered data, but no fixed difference.	Rank 1, 2, 3
Interval	Ordered, fixed difference, no true zero.	Temperature (°C)
Ratio	Ordered, fixed difference, has true zero.	Height, Weight

---

(C) Concept of Feature

A **feature** is a measurable property of data.
Example: In a student dataset – marks, attendance, and hours studied are features.

(D) Feature Construction

Creating new features using existing ones.
Example: “BMI” = weight/height^2 (made from weight & height).

(E) Feature Selection

Choosing the most important features and removing unnecessary ones to improve model performance.

(F) Feature Transformation

Changing data into another form (like scaling or normalizing) so the model understands it better.

⚠ (G) Curse of Dimensionality

When we have **too many features**, model performance **drops** because:

- Computation becomes harder.
- Data becomes sparse.
- Model may overfit (memorize instead of learning).

📚 7. Dataset Preparation

(A) Training vs. Testing Dataset

Dataset	Use
Training Dataset	Used to train the model (learn patterns).
Testing Dataset	Used to test accuracy and performance.

(B) Validation Techniques

1. **Hold-out Method**
    - Split dataset into training and testing parts (e.g., 70% train, 30% test).
2. **k-Fold Cross Validation**
    - Divide data into *k* equal parts.
    - Train on *k−1* parts, test on 1 part.
    - Repeat *k* times and take average accuracy.
3. **Leave-One-Out Cross Validation (LOOCV)**
    - Special case of k-fold where *k = number of data points*.
    - Train on all except one sample, test on that one.
    - Repeat for every data point.

✅ Mapping of Course Outcome for Unit I: CO1

- Understand basics of Machine Learning, its types, tasks, data features, and dataset preparation techniques.

□ UNIT II – CLASSIFICATION (6 Hours)

□ 1. Binary Classification

💡 What is Classification?

Classification means **dividing data into categories or classes**.
Example:
Predict whether an email is **spam** or **not spam** → two classes → *binary classification*.

□ Linear Classification Model

- In linear classification, we use a **straight line (or plane)** to separate the two classes.
- Example:
    - Class 1 → "Pass",
    - Class 2 → "Fail".
    - The model finds a line (in 2D) or plane (in 3D) that separates them.

Equation form:
[
y = w_1x_1 + w_2x_2 + b
]
where

- (x_1, x_2) = input features,
- (w_1, w_2) = weights (learned by model),
- (b) = bias (helps adjust the line).

📊 2. Performance Evaluation (Binary Classification)

We use various metrics to check how good the model is.

📘 (A) Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

👉 **Example:**
Model predicting if students pass or fail.

- TP = Predicted pass and actually passed.
- FP = Predicted pass but actually failed.

⚙️ (B) Accuracy

[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
]
Tells how many predictions are correct out of total.

⚙️ (C) Precision

[
Precision = \frac{TP}{TP + FP}
]
Tells how many predicted positives are actually correct.
□ *High precision → few false alarms.*

⚙️ (D) Recall (Sensitivity)

[
Recall = \frac{TP}{TP + FN}
]
Tells how many actual positives were correctly predicted.
□ *High recall → model catches most positives.*

⚙️ (E) F-Measure (F1-Score)

[
F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}
]
It balances both precision and recall.

⚙️ (F) ROC Curve (Receiver Operating Characteristic)

- It plots **True Positive Rate (TPR)** vs **False Positive Rate (FPR)**.
- **Area Under Curve (AUC)** shows performance:
    - AUC = 1 → perfect model
    - AUC = 0.5 → random guess

🎯 3. Multi-Class Classification

💡 What it is:

When data has **more than two classes**.
Example: Classifying fruits as 🍎 Apple, 🍌 Banana, 🍊 Orange.

□ Model

Model learns **boundaries** for multiple classes.

📊 Performance Evaluation Metrics

Metric	Meaning
Per-class Precision	Precision for each individual class
Per-class Recall	Recall for each individual class

Metric	Meaning
Weighted Average Precision / Recall	Weighted average based on number of samples per class

 Example:
If there are 3 classes (A, B, C), we calculate precision and recall for each, then average them.

⚙️ Handling More than Two Classes

1. **One vs One (OvO)**
   - Create classifier for every pair of classes.
   - For 3 classes (A, B, C): classifiers are (A vs B), (B vs C), (A vs C).
   - Total = n(n−1)/2 classifiers.
2. **One vs Rest (OvR)**
   - For each class, build one model that classifies that class vs all others.
   - For 3 classes → 3 models: (A vs Rest), (B vs Rest), (C vs Rest).

⚙️ 4. Linear Models

💡 Introduction

Linear models use **a linear equation** to make predictions or classification boundaries.

🔹 Linear Support Vector Machine (SVM)

- **Goal:** Find a line (or hyperplane) that best separates the classes.
- The **best hyperplane** maximizes the **margin** — the distance between the line and nearest data points (called **support vectors**).

🔹 Soft Margin SVM

- Used when data is **not perfectly separable**.
- Allows some **misclassifications** to get a better overall boundary.

🔹 SVM Kernels (for non-linear data)

If data is not linearly separable, we use kernels to transform data into higher dimensions.

Kernel	Description
RBF (Radial Basis Function)	Creates circular boundaries.
Gaussian	Similar to RBF; focuses on distance from a center point.
Polynomial	Creates curved decision boundaries.
Sigmoid	Works like a neural network activation function.

---

5. Logistic Regression

💡 Model

Used for **classification**, not regression (despite the name).
Predicts the **probability** of an instance belonging to a class.

Equation:
[
P(y=1|x) = \frac{1}{1 + e^{-(w_1x_1 + w_2x_2 + b)}}
]

⚙️ Cost Function

We use **Log Loss (Binary Cross-Entropy)** to measure how far predicted probabilities are from actual labels.

[
Cost = -\frac{1}{n} \sum [y\log(p) + (1-y)\log(1-p)]
]

Model learns by adjusting weights (w) and bias (b) to **minimize cost**.

✅ Mapping of Course Outcome for Unit II: CO2

- Understand classification techniques (binary and multi-class), evaluation metrics, SVMs, and logistic regression.

📈 UNIT III – REGRESSION (6 Hours)

□ 1. What is Regression?

Simple meaning:
Regression means **predicting a continuous (numerical) value** based on data.

□ Example:
Predicting:

- House price based on area
- Student marks based on study hours
- Temperature based on time of year

👉 Classification → categories (Yes/No)
👉 Regression → numbers (e.g., 78.5 marks)

📊 2. Univariate Regression

Univariate = "one variable"
It means the model predicts the output using **only one input feature**.

💡 Example:

Predicting student marks based on hours studied.

- Input (X): hours studied
- Output (Y): marks scored

□ Least Square Method (Simple Linear Regression)

We try to fit a **straight line** through the data points that minimizes errors.

Equation of line:
[
Y = mX + c
]
where

- (Y) = predicted value

- (X) = input feature
- (m) = slope (how steep the line is)
- (c) = intercept (where line cuts Y-axis)

⚙️ Goal of Least Squares:

Find **m** and **c** such that the **sum of squared errors** between actual and predicted Y is minimum.

Error for each point = (Actual Y – Predicted Y)

📉 Cost Function

The **cost function** tells how wrong the model's predictions are.

Common cost functions:

Cost Function	Formula	Meaning
MSE (Mean Squared Error)	(\frac{1}{n}\sum(y_{actual} - y_{pred})^2)	Average of squared errors.
MAE (Mean Absolute Error)	(\frac{1}{n}\sum	y_{actual} - y_{pred}
R² (R-Square)	(1 - \frac{SS_{res}}{SS_{tot}})	Measures how well line fits data (1 = perfect fit).

▯ Example (Simple):

Hours (X)	Marks (Y)
1	35
2	40
3	50
4	55
5	60

Regression line (approx):
[
Y = 6X + 30
]

So, if a student studies for 6 hours → predicted marks = 6×6 + 30 = 66.

⚙️ 3. Optimization of Linear Regression with Gradient Descent

💡 What is Gradient Descent?

Gradient Descent is a method used to **find the best values of m and c** that minimize the cost function.

It’s like walking downhill to reach the **lowest point** (minimum error).

💠 Steps of Gradient Descent:

1. Start with random values of (m) and (c).
2. Calculate the cost (MSE).
3. Adjust (m) and (c) slightly in the direction that **reduces cost**.
4. Repeat until cost is minimal.

📘 Formula Updates:

[
m = m - \alpha \frac{\partial J}{\partial m}
]
[
c = c - \alpha \frac{\partial J}{\partial c}
]

where:

- (J) = cost function
- (\alpha) = learning rate (step size)

👉 The process continues until the line fits data perfectly (or almost perfectly).

💠 4. Estimating Regression Coefficients

These coefficients ((m, c)) can be found:

1. Using **Mathematical formula** (Normal Equation).
2. Using **Gradient Descent** (iterative learning).

Normal Equation:
[
$\theta = (X^TX)^{-1}X^TY$
]
where

- (X) = input data matrix
- (Y) = output vector
- (θ) = coefficients (m, c)

---

5. Multivariate Regression

When we have **more than one input feature**, we use **Multivariate (Multiple Linear) Regression**.

Example:

Predicting house price based on:

- Size (sq. ft)
- Number of rooms
- Location rating

Equation:
[
$Y = w_1X_1 + w_2X_2 + w_3X_3 + b$
]
Each (w_i) shows how much each feature affects the output.

---

6. Polynomial Regression

When data is **not in a straight line**, we use **polynomial regression** — curve fitting.

Equation:
[
$Y = a_0 + a_1X + a_2X^2 + a_3X^3 + ...$
]

Example: Predicting temperature over time — curve shape instead of line.

⚖️ 7. Generalization Concepts

▯ Overfitting

- Model learns the **training data too perfectly** (including noise).
- Performs badly on new data.
- Example: Memorizing answers instead of understanding the topic.

▯ Underfitting

- Model is **too simple** and can't learn patterns properly.
- Performs badly on both training and test data.
- Example: Not studying enough.

⚖ Bias vs Variance

Term	Meaning	Effect
Bias	Error due to oversimplification (underfitting).	Model too rigid.
Variance	Error due to too much complexity (overfitting).	Model too sensitive.

👉 **Goal:** Find a **balance** between bias and variance for best performance.

✅ Mapping of Course Outcome for Unit III: CO3

- Understand regression concepts, types, cost functions, gradient descent optimization, and performance evaluation.

🌳 UNIT IV – TREE BASED AND PROBABILISTIC MODELS (6 Hours)

🌲 1. TREE BASED MODEL

💡 What is a Decision Tree?

A **Decision Tree** is a model that uses a **tree-like structure** to make decisions.

- Each **internal node** → checks a condition on a feature.
- Each **branch** → result of that condition (Yes/No).
- Each **leaf node** → final decision or output.

□ Example:

Predict whether a person will buy a car:

```
             Age > 25?
             /        \
          Yes          No
         /              \
   Income > 50k?         Buy = No
       /      \
     Yes       No
    /            \
Buy=Yes       Buy=No
```

So, the model splits data based on questions (conditions) — just like humans think logically.

□ Concepts and Terminologies

Term	Meaning
Root Node	The starting point (first question).
Decision Node	A node where the data is split further.
Leaf Node	Final output (no further split).
Splitting	Dividing data into groups based on a condition.
Branch	Path representing an outcome of a test.
Pruning	Removing unnecessary branches to avoid overfitting.

□ Impurity Measures (to decide the best split)

We use formulas to measure how “mixed” the data is.
The purer (less mixed) → the better the split.

(A) Gini Index

- Measures impurity of data.

- Formula:
  [
  Gini = 1 - \sum p_i^2
  ]
  where (p_i) = probability of each class.

👉 **If Gini = 0** → perfect purity (only one class in node).

Example:
If 50% yes, 50% no → Gini = 1 - (0.5^2 + 0.5^2) = 0.5 (impure)

(B) Entropy

- Another measure of impurity.
  [
  Entropy = -\sum p_i \log_2(p_i)
  ]
  👉 Lower entropy = purer data.

Example:
If node has 100% "Yes" → Entropy = 0
If node has 50% "Yes", 50% "No" → Entropy = 1 (most impure)

(C) Information Gain

Tells how much **entropy is reduced** after a split.

[
Information,Gain = Entropy(before) - Entropy(after)
]
👉 Higher gain = better split!

✂ Tree Pruning

Why prune?
Sometimes tree becomes too large and fits the training data perfectly (overfitting).

Pruning = cutting off less useful branches.

Types:

1. **Pre-Pruning** → Stop growing early based on conditions (like minimum samples).

2. **Post-Pruning** → Grow full tree, then remove weak branches.

▯ Algorithms:

Algorithm	Description
ID3 (Iterative Dichotomiser 3)	Uses **Entropy** and **Information Gain** to choose splits.
C4.5	Extension of ID3. Handles **continuous values** and **missing data**, uses **Gain Ratio**.

✅ Advantages of Decision Trees:

- Easy to understand and interpret.
- No need for data scaling or normalization.
- Works for both classification and regression.

⚠ Limitations:

- Can easily **overfit** data.
- Sensitive to small changes in data.
- For large datasets, trees can become complex.

🎲 2. PROBABILISTIC MODELS

Now let’s move to models based on **probability and statistics**.

▯ (A) Conditional Probability

Conditional Probability = Probability of event A happening **given** event B already happened.

[
P(A|B) = \frac{P(A \text{ and } B)}{P(B)}
]

Example:
If 70% students study, and 60% of those pass →
P(Pass | Study) = 0.6

□ (B) Bayes Theorem

This is the core of probabilistic learning.

[
P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}
]

It helps update probability based on **new evidence**.

Example (Spam Detection):

- P(Spam | "win money")
  → How likely an email is spam if it contains the words "win money".

□ (C) Naïve Bayes Classifier

A simple but powerful **probabilistic classifier** based on Bayes Theorem.

Naïve = assumes all features are **independent** of each other.

Example:

Predict if a person buys a phone based on:

- Age
- Income
- Ad response

It calculates probability for each class (Buy = Yes/No) and picks the higher one.

[
P(Buy=Yes|Data) = P(Data|Buy=Yes) * P(Buy=Yes)
]

✅ Advantages:

- Simple, fast, and efficient.
- Works well for text classification (like spam filtering).
- Needs small training data.

⚠️ Disadvantages:

- Assumes features are independent (not always true).
- Poor performance if this assumption is violated.

□ (D) Bayesian Network

A **graphical model** that represents relationships between variables using **nodes and directed edges**.

- **Nodes** → random variables.
- **Edges** → dependencies (cause–effect relationships).

Example:

Weather prediction

```
  Rain → Wet Grass
    ↓
Traffic
```

- “Rain” affects “Wet Grass” and “Traffic”.
- Bayesian Network shows such conditional dependencies.

⚙️ Uses of Bayesian Networks:

- Medical diagnosis
- Weather forecasting
- Fault detection systems

✅ Mapping of Course Outcome for Unit IV: CO4

- Understand Decision Tree working, splitting measures, pruning, and probabilistic models like Naïve Bayes and Bayesian Networks.

📓 UNIT V – DISTANCE AND RULE-BASED MODELS (6 Hours)

▯ 1. DISTANCE-BASED MODELS

These models make predictions by **measuring distance** between data points.
They are based on the idea:

“Similar things stay close together.”

💥 Example:

If you have a new student and you want to predict whether they will pass or fail based on attendance and marks —
→ check which old students are closest (similar) and see what happened with them.

▯ (A) K-Nearest Neighbour (KNN)

💡 Definition:

KNN is a **lazy learning algorithm** that classifies a new data point based on the **majority of its K nearest neighbours**.

🔢 Step-by-step Working:

1. **Choose value of K** (e.g., K = 3).
2. **Calculate distance** between the new data point and all training data points.
3. **Select K nearest neighbours** (those with smallest distance).
4. **Count class labels** among the K neighbours.
5. **Assign the most common label** to the new point.

▯ Example:

Student	Attendance	Marks	Result
A	80	85	Pass
B	90	88	Pass
C	40	35	Fail
D	50	45	Fail

Now we want to predict for a student with (70, 75).

Steps:

1. Calculate distance from all 4 students.
2. Pick **K=3** nearest ones.
3. Majority label = **Pass** → So prediction = **Pass** ✅

📏 Common Distance Metrics:

Distance Type	Formula	Use Case
Euclidean	$\sqrt{\Sigma(x_1 - y_1)^2}$	Continuous data
Manhattan	Σ	$x_1 - y_1$
Minkowski	$(\Sigma$	$x_1 - y_1$
Cosine Similarity	$\cos(\theta) = A \cdot B / A$	

⚙️ Important Concepts:

- **Lazy Learner:** No model is built during training; all computation happens at prediction time.
- **Non-parametric:** Doesn't assume any fixed shape or distribution of data.

✅ Advantages:

- Simple and easy to implement.
- No training phase (fast to set up).
- Works for both classification and regression.

⚠️ Disadvantages:

- Slow for large datasets (must compare with every point).
- Sensitive to noisy data and irrelevant features.
- Choosing the right **K** is important.

---

2. RULE-BASED MODELS

These models make predictions using **a set of IF-THEN rules** derived from data.

Definition:

A **rule-based classifier** uses logical rules in the form:

IF (condition) THEN (class label)

Example:

IF (age < 25) AND (income > 50k) THEN (buy = yes)

Components of a Rule:

Part	Description
Antecedent (IF part)	Condition(s) on input attributes.
Consequent (THEN part)	Output or predicted class.

Example:

Age	Income	Buy
23	70k	Yes
30	30k	No
22	80k	Yes

Rule generated →

IF age < 25 AND income > 60k THEN buy = yes

How Rules Are Generated

Rules can be created by:

1. **Manually (expert system)** – Rules written by experts.
2. **Automatically (machine learning)** – Using algorithms like:
   - **ID3/C4.5** → Decision Tree converted into rules.

- **RIPPER (Repeated Incremental Pruning to Produce Error Reduction) →** Generates optimized rule sets.

▯ Conflict Resolution:

Sometimes multiple rules apply to one case.
To decide which rule to choose:

Strategy	Description
Rule ordering	Rules arranged by priority.
Specificity ordering	More specific rule preferred.
Rule weighting	Rules with higher accuracy are preferred.

✅ Advantages:

- Easy to interpret and explain.
- Can combine human knowledge and data.
- Works well with symbolic or categorical data.

⚠ Disadvantages:

- Difficult to maintain large sets of rules.
- Not good with continuous data unless discretized.
- May conflict if not carefully designed.

🔁 Comparison: KNN vs Rule-Based

Feature	KNN	Rule-Based
Type	Distance-based	Logic-based
Training	No explicit training	Requires rule generation
Explainability	Hard to interpret	Very interpretable
Speed	Slow at prediction	Fast once rules are made
Use case	Continuous data	Categorical data

✅ Mapping of Course Outcome for Unit V: CO5

- Understand and apply **Distance-based (KNN)** and **Rule-based** models for classification and prediction.

🤖 UNIT VI – INTRODUCTION TO ARTIFICIAL NEURAL NETWORK (6 Hours)

▯ 1. Introduction to Artificial Neural Networks (ANN)

💥 What is ANN?

An **Artificial Neural Network (ANN)** is a **machine learning model** inspired by how the **human brain** works.
It is made up of small units called **neurons**, which are connected in layers and help the system **learn patterns** from data.

💡 Simple Idea:

Just like our brain learns from experience,
ANN learns from **data** by adjusting the **connections (weights)** between neurons.

▯ ANN Structure:

1. **Input Layer:** Takes the input features (like marks, hours studied, etc.).
2. **Hidden Layer(s):** Processes data and learns complex relationships.
3. **Output Layer:** Gives the final result or prediction.

▯ Example:

Predicting whether a student will **pass or fail**:

- Input → hours studied, attendance
- Output → pass/fail
  The network learns the relationship between them!

□ 2. Biological Neuron vs Artificial Neuron

Biological Neuron	Artificial Neuron
Brain cell that processes signals	Mathematical model that processes inputs
Dendrites receive signals	Inputs ($x_1, x_2, x_3, \dots$)
Axon sends signals	Output (y)
Synapses control signal strength	Weights ($w_1, w_2, w_3, \dots$) control input importance

---

⚙️ 3. McCulloch-Pitts Neuron Model

□ Concept:

This is the **simplest mathematical model** of a neuron.

Formula:

[
$$y = f\left(\sum w_i x_i + b\right)$$
]

Where:

- (x_i): Input values
- (w_i): Weights for each input
- (b): Bias (constant)
- ($f()$): Activation function
- (y): Output

🔍 Working:

1. Multiply each input by its weight.
2. Add them all + bias.
3. Apply activation function.
4. Output the result.

□ 4. Perceptron and Its Learning Algorithm

💡 Definition:

A **Perceptron** is the **simplest type of ANN** used for **binary classification** (yes/no type problems).

▯ Structure:

- **Inputs:** $x_1, x_2, \ldots$
- **Weights:** $w_1, w_2, \ldots$
- **Summation Unit:** Calculates weighted sum
- **Activation Function:** Decides output (1 or 0)

⚙ Working Example:

If output = 1 → class A
If output = 0 → class B

▯ Perceptron Learning Algorithm:

Step 1: Initialize weights randomly.
Step 2: For each training sample:

- Calculate predicted output
- Compare with actual output
- Update weights as:
  [
  $w_i = w_i + \eta (y_{actual} - y_{predicted})x_i$
  ]
  where **η** = learning rate
  Step 3: Repeat until error becomes very small.

✅ Advantages:

- Easy to understand and implement.
- Works for linearly separable data.

⚠ Disadvantages:

- Cannot solve non-linear problems (like XOR).

⚡ 5. Sigmoid Neuron

When data is **not linearly separable**, we use **sigmoid neurons** instead of simple perceptrons.

📈 Sigmoid Activation Function:

$$
f(x) = \frac{1}{1 + e^{-x}}
$$

✨ Properties:

- Output always between 0 and 1
- Smooth and continuous
- Good for probabilistic outputs

□ 6. Other Activation Functions

Activation Function	Formula	Output Range	Shape / Use
Tanh	$\tanh(x)$	-1 to +1	Centered around 0
ReLU (Rectified Linear Unit)	$f(x) = \max(0, x)$	0 to ∞	Fast and simple
Sigmoid	$\frac{1}{1+e^{-x}}$	0 to 1	Smooth S-shape

□ 7. Multi-Layer Perceptron (MLP)

💡 Definition:

A **Multi-Layer Perceptron (MLP)** is an ANN that has:

- One **input layer**
- One or more **hidden layers**
- One **output layer**

This helps solve **non-linear** problems.

□ Example Structure:

```
Input Layer → Hidden Layer → Output Layer
```

Each neuron in one layer is connected to **all neurons** in the next layer.

⚙️ Learning Parameters:

Parameter	Meaning
Weights (w)	Strength of connections between neurons
Bias (b)	Shifts activation function to fit data better
Learning Rate (η)	Controls how much weights change per step

📉 8. Loss Function: Mean Square Error (MSE)

📘 Formula:

[
MSE = \frac{1}{n}\sum (y_{actual} - y_{predicted})^2
]

Goal:
To minimize the error between actual and predicted outputs.

▯ Example:

If actual = [1, 0, 1]
and predicted = [0.9, 0.2, 0.8]

[
MSE = \frac{(1-0.9)^2 + (0-0.2)^2 + (1-0.8)^2}{3} = 0.0266
]

Lower MSE = better model 🎯

🤖 9. Introduction to Deep Learning

💡 Definition:

Deep Learning = Machine Learning using **many-layered neural networks** (more than 2 hidden layers).

🌊 Why called "Deep"?

Because it has **deep (many)** layers that extract complex patterns from data automatically.

🧠 Examples of Deep Learning:

1. 🙋‍♂️ Face Recognition
2. 🎙 Speech Recognition
3. 🚗 Self-Driving Cars
4. 📸 Object Detection (in photos)

✅ Mapping of Course Outcome for Unit VI: CO6

- Understand the structure and working of **Artificial Neural Networks**, **Perceptrons**, and **Activation Functions**, and get an **introduction to Deep Learning**.

🎉 Summary of All Units (Quick Recall)

Unit	Topic	Focus
I	Introduction to ML	Types, Features, Datasets
II	Classification	Binary & Multi-class, SVM, Logistic Regression
III	Regression	Simple, Multiple, Polynomial
IV	Tree & Probabilistic Models	Decision Trees, Naïve Bayes
V	Distance & Rule-based Models	KNN, Association Rules
VI	Neural Networks	ANN, Perceptron, Deep Learning
