Unit-3: Understanding Supervised Learning

3.1 Overview of Supervised Learning

At its simplest, **Supervised Learning** is like learning with a teacher. Imagine a child learning to identify fruits: a teacher shows the child an apple and says, "This is an apple." The teacher shows a banana and says, "This is a banana." After seeing enough examples, the child can look at a new fruit they've never seen before and correctly identify it.

In technical terms, Supervised Learning is a type of Machine Learning where the algorithm is trained on a **labeled dataset**.

- **Input Data (X):** The features or variables we provide to the model (e.g., the color and shape of a fruit).
- **Output Labels (Y):** The correct answers or targets (e.g., the name of the fruit).

The goal of the algorithm is to learn a mapping function (f) that connects the input to the output:

$$Y = f(X)$$

Example

- Input: Email text
- Output: Spam or Not Spam
  The model learns from emails that are already labeled.

Common Uses

- Predicting house prices
- Identifying handwritten digits
- Detecting fraud in banking
- Disease diagnosis

3.1.1 Concepts of Supervised Learning

1. Labeled Dataset: A dataset where each input has a corresponding correct output.

Examples:

- An email classified as "spam" or "not spam".
- A customer review tagged with a sentiment label: "positive", "negative", or "neutral".
- A recording of a sound with the label "breaking glass" or "dog barking".
- A bank transaction record with a flag indicating "fraudulent" or "legitimate" activity.
- A house price in a spreadsheet, with columns for features like location, size, and number of rooms used as inputs to predict the price (the label).

2. Input Variables (Features)

- Independent variables used to make predictions
- Represent characteristics or attributes of data

Example: Age, salary, temperature, size

3. Output Variable (Target/Label)

- Dependent variable
- The value to be predicted

Example: Price, category, class label

4. Training Phase

- The model learns patterns from labeled data
- Errors are calculated and minimized

5. Testing Phase

- The trained model is tested on new data
- Used to evaluate performance

6. Learning Algorithm

Algorithms used in supervised learning include:

- Linear Regression
- Logistic Regression
- Decision Trees
- Support Vector Machines (SVM)
- k-Nearest Neighbors (KNN)

Advantages of Supervised Learning

✔ High accuracy
✔ Clear performance evaluation
✔ Easy to understand results

Limitations

✖ Requires large labeled datasets
✖ Labeling data can be time-consuming
✖ Less flexible to unseen patterns

3.1.2 Difference between Classification and Regression

Supervised Learning is generally divided into two main categories based on the type of output you want to predict: **Classification** and **Regression**.

1. Classification (Predicting a Category)

In Classification, the output variable is a **category** or a **discrete label**. You are asking the model to put data into specific "buckets."

- **Binary Classification:** Only two possible outcomes (e.g., Spam vs. Not Spam).
- **Multi-class Classification:** More than two categories (e.g., classifying an image as a "dog," "cat," or "bird").

2. Regression (Predicting a Number)

In Regression, the output variable is a **continuous, numerical value**. You are asking the model to predict "how much" or "how many."

- **Example:** Predicting the price of a stock, the temperature tomorrow, or the amount of rainfall.

Feature	**Classification**	**Regression**
Output Type	Discrete (Categories/Labels)	Continuous (Numbers)
Nature of O/P	Discrete classes	Numeric values
Used for	Grouping	Prediction
Goal	To find a decision boundary to separate classes.	To find a trend line that fits the data points.
Example	Is this email spam? (Yes/No)	What will the gold price be tomorrow?
Common Algorithms	Logistic Regression, Decision Trees, SVM.	Linear Regression, Polynomial Regression.

Simple Analogy

- **Classification:** Choosing the correct *bucket*
- **Regression:** Finding the *exact value*

3.2 Basic Terminologies in Machine Learning

Before building machine learning models, it is important to understand some basic terms. These terminologies help in understanding how data is prepared, how models learn, and how performance is evaluated.

3.2.1 Dataset, Features, Labels

1. Dataset

A **dataset** is a collection of data used to train and test a machine learning model.
It is usually organized in **rows and columns** (like a table).

- Each **row** represents one data item (example/record)
- Each **column** represents a variable or attribute

Example:
A dataset of students containing their marks, attendance, and final result.

Marks	Attendance	Result
80	90%	Pass
45	60%	Fail

2. Features

Features are the **input variables** used by the model to make predictions.

- Also called **independent variables**
- They describe the characteristics of the data

Examples of features:

- Age
- Salary
- Marks
- Height
- Attendance

In the student dataset above: **Marks** and **Attendance** are features

3. Labels

Labels are the **output values** that the model tries to predict.

- Also called **target variable** or **dependent variable**
- Present only in **supervised learning**

Examples of labels:

- Pass / Fail
- Price of a house
- Yes / No
- Disease / No disease

In the student dataset: **Result (Pass/Fail)** is the label.

Difference between Features and Labels

Features	Labels
Input to the model	Output of the model
Used for prediction	Value to be predicted
Independent variables	Dependent variable

3.2.2 Training Data, Test Data, Validation Data

1. Training Data

Training data is the portion of data used to **train the model**.

- The model learns patterns from this data
- Usually the **largest portion** of the dataset

Typical split: 60–70%

Example:
Using past student records to teach the model how marks and attendance affect results.

2. Test Data

Test data is used to **evaluate the final performance** of the trained model.

- The model has **never seen this data before**
- Used to check how well the model performs on new data

Typical split: 20–30%

3. Validation Data

Validation data is used to **tune and improve the model** during training.

- Helps in selecting the best model
- Used to adjust parameters (hyper parameters)

Typical split: 10–20%

Data Type	Purpose
Training Data	Learn patterns
Validation Data	Improve and tune model
Test Data	Final performance evaluation

3.2.3 Overfitting and Underfitting

These are common problems in machine learning models.

---

1. Underfitting

Underfitting occurs when a model is **too simple** to learn patterns from the data.

- Performs poorly on **training data**
- Performs poorly on **test data**

Causes:

- Very simple model
- Too few features
- Insufficient training

Example: Trying to predict house prices using only the number of rooms and ignoring location, area, and amenities.

2. Overfitting

Overfitting occurs when a model learns the **training data too well**, including noise.

- Performs very well on training data
- Performs poorly on test data

Causes:

- Too complex model

- Too many features
- Very small training dataset

Example: A model that memorizes answers instead of understanding concepts.

Underfitting	**Overfitting**
Model is too simple	Model is too complex
Low accuracy everywhere	High training accuracy, low test accuracy
Fails to learn patterns	Learns noise and details

Balanced (Good Fit) Model

A good model:

- Learns important patterns
- Performs well on both training and test data

3.3 Loss Functions

In machine learning, a **loss function** measures **how wrong a model's predictions are**. It compares **actual values** with **predicted values** and returns a numerical value.

Lower loss = Better model performance

Loss functions are mainly used in **regression problems**.

3.3.1 Mean Squared Error (MSE)

Definition

Mean Squared Error (MSE) calculates the **average of the squared differences** between actual and predicted values.

Formula

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

Where:

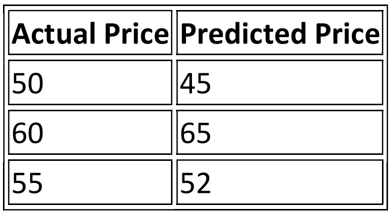
- yi = actual value
- y^i = predicted value
- n = number of observations

3.3.2 Definition of MSE

MSE measures the **average of the squares of the errors**. An "error" is simply the distance between the actual value (y) and the predicted value (y^).

- Find the difference between actual and predicted values
- Square the difference
- Take the average of all squared values

→Real-World Example: House Price Prediction



Actual Price	Predicted Price
50	45
60	65
55	52

Errors:
$(50-45)^2 = 25$
$(60-65)^2 = 25$
$(55-52)^2 = 9$

MSE = (25+25+9)/3 = 19.67

3.3.2.1 Computing MSE and its Properties

To calculate MSE, you follow these steps:

1. Find the difference between the actual and predicted value for every data point.
2. Square each of those differences.
3. Sum them all up.
4. Divide by the total number of data points (n).

The Formula:

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

Key Properties:

- **Always Positive:** Because we square the errors, MSE can never be negative.
- **Penalizes Outliers:** Because the error is squared, a big mistake is penalized much more heavily than a small one. (e.g., an error of 10 becomes a loss of 100, while an error of 2 is only 4).
- **Differentiable:** This makes it very "math-friendly" for optimization algorithms.

3.3.3 Mean Absolute Error (MAE)

MAE is a simpler, more intuitive way to look at error.

3.3.3.1 Definition of MAE

MAE is the **average of the absolute differences** between predicted and actual values. It treats all errors equally regardless of their size.

3.3.3.2 Computing MAE and its Properties

1. Calculate the difference between actual and predicted values.
2. Take the absolute value (ignore the negative sign).
3. Average these values.

The Formula:

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

Example:

Actual	Predicted	Absolute Error
50	45	5
60	65	5
55	52	3

MAE=(5+5+3)/3=4.33

Key Properties:

- **Robust to Outliers ():** Since we don't square the error, a single massive mistake doesn't "blow up" the total loss as much as it does in MSE.
- **Easy Interpretation:** The result is in the same units as your data. If you are predicting house prices and your MAE is 5,000, it means your predictions are off by $5,000 on average.
- **Linear Penalty:** Treats all errors equally
- **Non-negative:** MAE ≥ 0

MSE Vs MAE

Metric	MSE	MAE
Error type	Squared	Absolute
Sensitivity	High to outliers	Low
Units	Squared units	Same as target

Metric	MSE	MAE
Use case	Large error penalty	Robust models

3.3.3.3 Understanding Regression and R^2

While MSE and MAE tell us the "error," **R^2 (R-Squared)**, or the **Coefficient of Determination**, tells us about the "goodness of fit."

R^2 score measures **how well the regression model explains the variance** in the data.

- **What it represents:** It measures the proportion of the variance in the dependent variable that is predictable from the independent variables.
- **The Scale: R^2** usually ranges from **0 to 1** (or 0% to 100%).
  - **R^2 = 0:** The model explains none of the variability (it's as good as just guessing the average every time).
  - **R^2 = 1:** The model perfectly predicts every single data point.

The Formula:

$$R^2 = 1 - \frac{\text{Residual Sum of Squares}}{\text{Total Sum of Squares}}$$

R^2 Value Interpretation:

R^2 Value	Meaning
1.0	Perfect prediction
0.8	Very good model
0.0	No improvement over mean
Negative	Very poor model

Analogy: If you are trying to predict a student's grade based on study hours, an **R^2** of 0.85 means that 85% of the "reason" grades vary between students can be explained by how much they studied. The remaining 15% is due to other factors (luck, sleep, etc.).