

Supervised Learning - I

By:
Dr. Shikha Deep

Contents:

- (1) Learning Models Generalities
- (2) Metrics for Evaluating the Model

Learning Models Generalities:

A learning model is like a **mathematical machine** that:

- Takes **inputs (X)** → e.g., study time, attendance, grades, etc.
- Produces **outputs (y)** → e.g., pass/fail, score prediction, etc.
- Tries to find a **relationship** between them.

Formally, the model tries to learn a function:

$$y = f(X)$$

But the model **approximates** it:

$$\hat{y} = f_{\text{model}}(X)$$

Here:

y = true label (actual result)

\hat{y} = predicted label (model's guess)

f_{model} = learned function (the “brain” of the model)

Points to Remember:

1. The model's goal is to minimize the difference between actual (y) and predicted (\hat{y}) outputs.
This difference is called error or loss.
2. The model learns through training.
 - The model starts with random guesses for its internal parameters (like slope and intercept in regression).
 - Then it looks at training examples and measures how wrong it is (the loss).
 - It then adjusts its parameters slightly to reduce that error.
 - It repeats this process many times — gradually improving.

This process is called **optimization or learning**.

Metrics for Evaluating the Model

Model evaluation refers to the process of assessing how well a trained model performs on unseen data using quantitative performance measures such as accuracy, precision, recall, and F1-score.

For predicting discrete categories we use:

Accuracy

Precision

F1 score

Confusion Matrix

Confusion Matrix

(Foundation for all Metrics)

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

Explanation with example :

- True Positives (TP): The model correctly predicts the positive class (e.g., correctly identifies spam emails).
- False Negatives (FN): The model incorrectly predicts the negative class when the true class is positive (e.g., misses a spam email, labeling it as not spam).
- False Positives (FP): The model incorrectly predicts the positive class when the true class is negative (e.g., labels a non-spam email as spam).
- True Negatives (TN): The model correctly predicts the negative class (e.g., correctly identifies non-spam emails).

Using TP, FN, FP, and TN, we calculate the following metrics to evaluate a classification model:

(1) Accuracy

The proportion of correct predictions (both positive and negative) out of all predictions.

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

- Measures how often the model is correct.
- Example:
If model predicted 90 out of 100 emails correctly →
Accuracy = 90%.

(2) Precision

The proportion of true positive predictions out of all positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP \text{ or } (Predicted \text{ Positive})}$$

- Out of all predicted positives, how many are actually positive?

(3) Recall (Sensitivity or True Positive Rate)

The proportion of **actual positives** correctly identified by the model (When it's actually yes, how often it is predicted yes).

$$\text{Recall} = \frac{TP}{TP + FN \text{ or } (Actual\ Positives)}$$

- Out of all actual positives, how many did the model detect?

(4) F1 Score

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

- It balances both — useful when you want to consider both false positives and false negatives.

(5) Specificity:

When it's actually no, how often does it predict no ?

$$\text{Specificity} = \frac{TN}{TN + FP \text{ or } (Actual\ No)}$$

- Out of all actual negatives, how many were correctly predicted as negative?

Example where this can be use:

Scenario	Metric
Spam detection	Precision
Disease diagnosis	Recall
Fraud detection	F1-Score
Balanced tasks (like sentiment analysis)	Accuracy

