Evaluation Metrics for Machine Learning: Mathematical Notes

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance. It is critical for tuning hyperparameters and ensuring the model generalizes well to unseen data.

I. Classification Metrics

Classification metrics are primarily derived from the Confusion Matrix, which compares predicted class labels against ground truth labels.

1. The Confusion Matrix (Binary)

For a binary problem (Positive/Negative), the matrix consists of:

- **True Positive (TP):** Predicted P, Actual P.
- **True Negative (TN):** Predicted N, Actual N.
- **False Positive (FP):** Predicted P, Actual N (Type I Error).
- **False Negative (FN):** Predicted N, Actual P (Type II Error).

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

2. Fundamental Equations

- **Accuracy:** Overall correctness.

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

- **Precision:** Of all predicted positives, how many were actually positive? (Focuses on minimizing FP).

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity):** Of all actual positives, how many did we catch? (Focuses on minimizing FN).

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:** The harmonic mean of Precision and Recall. Useful for imbalanced datasets.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3. Step-by-Step Numerical Calculation (Binary)

Scenario: A model predicts whether an email is "Spam" (Positive, 1) or "Ham" (Negative, 0).
Data: 10 samples.

- Targets (t): $[1, 0, 1, 1, 0, 0, 1, 0, 1, 1]$
- Predictions (y): $[1, 1, 0, 1, 0, 0, 1, 0, 1, 1]$

Step 1: Tally Results

- **TP:** 5 (Indices 0, 3, 6, 8, 9)

- **TN:** 3 (Indices 4, 5, 7)
- **FP:** 1 (Index 1 — Predicted 1, Actual 0)
- **FN:** 1 (Index 2 — Predicted 0, Actual 1)

Step 2: Calculate Metrics

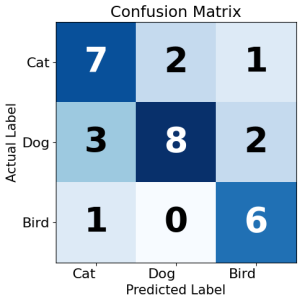
- **Accuracy:** $\frac{5+3}{10} = 0.8$ (80%)
- **Precision:** $\frac{5}{5+1} = \frac{5}{6} \approx 0.833$ (83.3%)
- **Recall:** $\frac{5}{5+1} = \frac{5}{6} \approx 0.833$ (83.3%)
- **F1-Score:** $2 \cdot \frac{0.833 \cdot 0.833}{0.833+0.833} \approx 0.833$

II. Multi-Class Confusion Matrix (3-Class Example)

In a 3-class problem (e.g., Cat, Dog, Bird), we evaluate the model by expanding the matrix to a 3×3 grid.

1. Numerical Matrix Setup

Assume 30 total samples distributed as follows:



2. Comprehensive Class-by-Class Calculations (One-vs-Rest)

To calculate metrics for a specific class, we isolate it by treating it as the "Positive" class and everything else as "Negative".

Class: Cat (C)

- **TP:** 7
- **FP:** $3 + 1 = 4$ (Sum of Cat column, excluding TP)
- **FN:** $2 + 1 = 3$ (Sum of Cat row, excluding TP)
- **Precision:** $\frac{7}{7+4} = \frac{7}{11} \approx 0.636$
- **Recall:** $\frac{7}{7+3} = \frac{7}{10} = 0.700$
- **F1-Score:** $2 \cdot \frac{0.636 \cdot 0.700}{0.636+0.700} \approx 0.666$

Class: Dog (D)

- **TP:** 8
- **FP:** $2 + 0 = 2$ (Sum of Dog column, excluding TP)
- **FN:** $3 + 2 = 5$ (Sum of Dog row, excluding TP)
- **Precision:** $\frac{8}{8+2} = \frac{8}{10} = 0.800$

- **Recall:** $\frac{8}{8+5} = \frac{8}{13} \approx 0.615$
- **F1-Score:** $2 \cdot \frac{0.800 \cdot 0.615}{0.800+0.615} \approx 0.695$

Class: Bird (B)

- **TP:** 6
- **FP:** $1 + 2 = 3$ (Sum of Bird column, excluding TP)
- **FN:** $1 + 0 = 1$ (Sum of Bird row, excluding TP)
- **Precision:** $\frac{6}{6+3} = \frac{6}{9} \approx 0.667$
- **Recall:** $\frac{6}{6+1} = \frac{6}{7} \approx 0.857$
- **F1-Score:** $2 \cdot \frac{0.667 \cdot 0.857}{0.667+0.857} \approx 0.750$

3. Averaged Metrics & Overall Accuracy

- **Overall Accuracy:** Sum of True Positives (diagonal elements) divided by Total Samples.

$$\text{Accuracy} = \frac{7 + 8 + 6}{30} = \frac{21}{30} = 0.70 \ (70\%)$$

- **Macro Precision:** $\frac{0.636+0.800+0.667}{3} \approx 0.701$
- **Macro Recall:** $\frac{0.700+0.615+0.857}{3} \approx 0.724$
- **Macro F1-Score:** $\frac{0.666+0.695+0.750}{3} \approx 0.704$

III. ROC Curve and Data Calculation

The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

$$TPR = \frac{TP}{TP + FN} \quad \text{(Recall)}$$

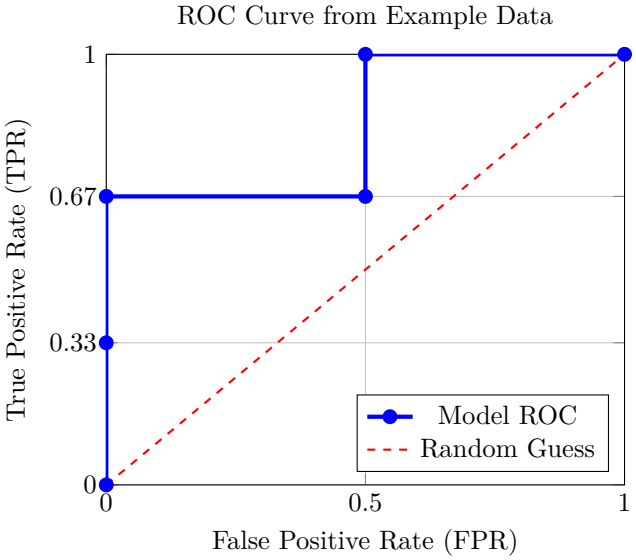
$$FPR = \frac{FP}{FP + TN}$$

1. Step-by-Step Numerical Example

Given a dataset of 5 samples with predicted probabilities ($\hat{p}$) for the positive class. Total Actual Positives (P) = 3. Total Actual Negatives (N) = 2. We sort by probability descending and shift the threshold.

Instance	**Prob ($\hat{p}$)**	**Actual (y)**	**Threshold**	**TP**	**FP**	**TPR ($TP/3$)**	**FPR ($FP/2$)**
1	0.90	1	≥ 0.90	1	0	$1/3 \approx 0.33$	0.0
2	0.80	1	≥ 0.80	2	0	$2/3 \approx 0.67$	0.0
3	0.60	0	≥ 0.60	2	1	$2/3 \approx 0.67$	0.5
4	0.40	1	≥ 0.40	3	1	$3/3 = 1.00$	0.5
5	0.20	0	≥ 0.20	3	2	$3/3 = 1.00$	1.0

2. Visualizing the Calculated ROC Curve



IV. Regression Metrics

Regression models predict continuous values. The error is the residual $e_i = y_i - \hat{y}_i$.

1. Error Formulas

- **Mean Absolute Error (MAE):** $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **Mean Squared Error (MSE):** $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- **Root Mean Squared Error (RMSE):** $\sqrt{MSE}$
- **Coefficient of Determination (R^2):** $1 - \frac{SS_{res}}{SS_{tot}}$

2. Step-by-Step Numerical Example (R^2)

Given a dataset with 4 samples: Actual $y = [3, 5, 2, 7]$, Predicted $\hat{y} = [2.5, 5.0, 2.2, 8.0]$.

Actual (y_i)	**Pred** ($\hat{y}_i$)	**Absolute** ($\lvert y_i - \hat{y}_i \rvert$)	**Squared** $(y_i - \hat{y}_i)^2$
3	2.5	0.5	0.25
5	5.0	0.0	0.00
2	2.2	0.2	0.04
7	8.0	1.0	1.00
	Sum ($\sum$)	**1.7**	**1.29** (SS_{res})

- **MAE:** $1.7/4 = 0.425$
- **MSE:** $1.29/4 = 0.3225$
- **RMSE:** $\sqrt{0.3225} \approx 0.5679$

Calculating R^2:
Mean of actual values $\bar{y} = (3+5+2+7)/4 = 4.25$.
$SS_{tot} = (3-4.25)^2 + (5-4.25)^2 + (2-4.25)^2 + (7-4.25)^2 = 1.5625 + 0.5625 + 5.0625 + 7.5625 = 14.75$.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{1.29}{14.75} \approx 0.9125 \text{ (91.25\% variance explained)}$$

V. Dataset Splitting Techniques

To objectively evaluate a model, we must test it on data it has never seen during training. Dataset splitting techniques establish how we divide our finite data into **Training** and **Testing** sets.

1. The Holdout Method

The simplest approach. The dataset D of size N is partitioned into two mutually exclusive sets: a training set (D_{train}) and a testing set (D_{test}).

- **Mathematical Definition:**

$$D_{train} \cup D_{test} = D \quad \text{and} \quad D_{train} \cap D_{test} = \emptyset$$

- **Split Ratio:** Given a split proportion $p \in (0, 1)$ (commonly $p = 0.7$ or 0.8):

$$N_{train} = p \cdot N$$

$$N_{test} = (1 - p) \cdot N$$

- **Pros & Cons:** Computationally fast, but sensitive to how the data is split (high variance), especially with small datasets.

2. k-Fold Cross-Validation

To reduce the variance of the holdout method, the data D is randomly partitioned into k equal-sized, mutually exclusive subsets (folds) $D_1, D_2, \ldots, D_k$.

- **Mathematical Size of each fold:**

$$N_{fold} = \frac{N}{k}$$

- **Process:** The model is trained and evaluated k times. In the i-th iteration, the test set is D_i, and the training set is the union of all other folds: $D \setminus D_i$.
- **Evaluation Error (E_{CV}):** The overall performance is the average of the error E_i calculated in each of the k iterations:

$$E_{CV} = \frac{1}{k} \sum_{i=1}^{k} E_i$$

- **Note:** When $k = N$, this is called **Leave-One-Out Cross-Validation (LOOCV)**.

3. The Bootstrap

The Bootstrap method relies on **random sampling with replacement**. Given a dataset of size N, we create a training set by randomly drawing N samples, allowing the same sample to be picked multiple times.

- **Mathematical Probability:** If we select a sample uniformly at random, the probability of any specific sample *not* being chosen in one draw is $1 - \frac{1}{N}$.
- The probability that a specific sample is *never* chosen after N draws is:

$$P(\text{never chosen}) = \left(1 - \frac{1}{N}\right)^N$$

- **Convergence (Euler's limit):** As N becomes large ($N \to \infty$):

$$\lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = e^{-1} \approx 0.368$$

- **Result:** This means approximately 36.8% of the original data points will *not* end up in the training set. These unused instances form the **Out-of-Bag (OOB)** test set. The training set is formed by the remaining $\approx 63.2\%$ unique instances (some duplicated).

V. Advanced Discussion & Pitfalls

- **The Accuracy Paradox (Imbalanced Data):** In highly skewed datasets (e.g., 99% legitimate transactions, 1% fraud), predicting the majority class universally yields 99% Accuracy. However, this model is practically useless. In such scenarios, **Precision, Recall, and the F1-Score** (or Area Under the Precision-Recall Curve - AUPRC) are strictly required to evaluate the minority class accurately.
- **The Bias-Variance Tradeoff in Splitting:**
  - *Holdout Method:* Can suffer from **high variance** in evaluation because the performance heavily depends on which data points randomly ended up in the test set.
  - *Leave-One-Out CV (LOOCV):* Where $K = N$. This has very **low bias** (since it trains on almost all data) but **high variance** (the models are highly correlated) and is computationally exorbitant.
  - *5-Fold or 10-Fold CV:* The standard heuristic that balances the bias-variance tradeoff optimally while keeping computational load manageable.
- **Data Leakage:** A critical error where information from outside the training dataset is used to create the model. If a dataset is scaled (e.g., min-max normalization) *before* the train/test split, the statistical properties of the test set have "leaked" into the training set, resulting in overly optimistic and mathematically invalid evaluation metrics. **Rule:** Always split data first, then compute transformations exclusively on D_{train}, and apply those same transformations to D_{test}.

Prepared By:

Md. Atikuzzaman

Lecturer

Department of Computer Science and Engineering

Green University of Bangladesh

Email: atik@cse.green.edu.bd