Evaluation metrics for ML models

domenica 12 ottobre 2025 09:28

Evaluating a machine learning model means measuring how well it performs the task it was trained for.
In other words, evaluation is about **assessing the model's ability to generalize** — to make accurate predictions on data it has never seen before.

When we train a model, we are not just interested in how well it fits the training data, but in how effectively it captures the *underlying patterns* that can be applied to new, unseen examples.
Without proper evaluation, a model might appear excellent during training but fail miserably in real-world scenarios — a common symptom of **overfitting**.

What does it mean to evaluate a model?

How Model Evaluation Works

The general process involves:

1. **Splitting the dataset** into at least two parts:
   - a **training set**, used to fit the model;
   - a **test set**, used exclusively for evaluation.
   (Sometimes a **validation set** is also used for tuning hyperparameters.)
2. **Computing quantitative metrics** such as accuracy, precision, recall, or RMSE to express model performance numerically.
3. **Comparing different models or configurations** using these metrics to choose the one that best balances bias and variance.

Ultimately, evaluating a model helps answer the question:

> "Has the model truly learned meaningful patterns, or is it just memorizing the training data?"

Training vs. Evaluation metrics ?

These two terms are closely related but serve different purposes in the model development process.

Training metrics are computed during or immediately after the training phase. They track how well the model is learning from the training data and are often used for diagnostics — for instance, to detect overfitting or underfitting, or to monitor whether the loss is decreasing as expected.

Evaluation metrics, on the other hand, are calculated after training is complete, using data that the model has never seen before (the test set). Their goal is to provide an *unbiased estimate* of how the model will perform in the real world.

Aspect	Training Metrics	Evaluation Metrics
When computed	During or after training	After training, on unseen data
Purpose	Monitor learning progress, tune hyperparameters	Assess generalization ability
Common examples	Training loss (e.g., MSE, cross-entropy)	Accuracy, F1-score, AUC, RMSE
Typical risk	Overfitting — metrics look great but don't generalize	Underperformance revealed on new data

WARNING! (overfitting or underfitting)

Example

Imagine a binary classification model that achieves **99% accuracy on the training set**, but only **75% on the test set**.
The training metric suggests excellent performance, yet the evaluation metric exposes poor generalization.
This discrepancy is exactly why both kinds of metrics are essential: training metrics guide the learning process, while evaluation metrics reveal the *true* performance.

Types of models & types of metrics

Machine learning models can be grouped into different categories depending on the kind of task they are designed to solve and the nature of the target variable they predict.
The three most common types are **regression**, **classification**, and **clustering**.
Although they share similar modeling principles, their goals, evaluation metrics, and interpretation differ significantly.

Regression

Goal: Predict a **continuous numerical value** based on input features.

Regression models learn a function that maps input variables (e.g., age, temperature, size) to a quantitative outcome (e.g., price, demand, energy consumption).

Examples:

- Predicting house prices from square footage and location
- Estimating a person's blood pressure from health data
- Forecasting sales or stock prices

Typical metrics:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R^2 (Coefficient of Determination)

These metrics measure the **magnitude of prediction errors** — the smaller the error, the better the model.

Classification

Goal: Assign each instance to one of a **finite set of categories or labels.**

Unlike regression, where outputs are continuous, classification deals with **discrete outcomes** — such as "spam or not spam," "disease or healthy," "image of a cat, dog, or bird," etc.

Examples:

- Email spam detection
- Sentiment analysis (positive / negative)
- Image recognition or medical diagnosis

Typical metrics:

- Accuracy
- Precision, Recall, F1-score
- ROC Curve, AUC (Area Under the Curve)
- Log-loss

Classification metrics often arise from the **confusion matrix**, which counts true positives, false positives, true negatives, and false negatives.
The goal is not only to be accurate overall but to **balance** correct predictions across all classes.

Clustering

Goal: Discover **hidden structure or natural groupings** in unlabeled data.

Clustering is an **unsupervised learning** task — meaning there are no predefined target labels. The algorithm tries to group similar data points together based on feature similarity or distance.

Examples:

- Customer segmentation in marketing
- Grouping similar news articles or images
- Detecting patterns in biological data

Typical metrics:

- Silhouette Score
- Davies–Bouldin Index
- Calinski–Harabasz Score
- (when ground truth is available) Adjusted Rand Index or Mutual Information

These metrics evaluate how **coherent** the clusters are internally and how **distinct** they are from one another.

Regression metrics

Absolute & Squared Metrics

- Mean Absolute Error (MAE)

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

- Measures the **average absolute difference** between predicted and true values.
- **Advantages:** simple to interpret; less sensitive to outliers.
- **Disadvantages:** penalizes all errors equally (doesn't highlight large deviations).

- Mean Squared Error (MSE)

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

- Squares each error, emphasizing **larger mistakes**.
- **Advantages:** differentiable (good for optimization); common loss function.
- **Disadvantages:** can be dominated by outliers.

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE}$$

- Same interpretation as MSE, but in the **same units** as the target variable.
- **Lower RMSE** indicates better model performance.

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- Expresses the average **percentage error**.
- **Advantages:** scale-independent, easy to communicate.
- **Disadvantages:** undefined for $y_i = 0$; biased when values are small.

- Symmetric Mean Absolute Percentage Error (SMAPE)

$$SMAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}$$

- Solves MAPE's division-by-zero issue; more robust when actuals are near zero.

- Mean Squared Logarithmic Error (MSLE)

$$MSLE = \frac{1}{n} \sum_{i=1}^{n} (\log(1+y_i) - \log(1+\hat{y}_i))^2 \rightsquigarrow RMSLE = \sqrt{MSLE}$$

- Reduces the impact of large absolute errors; suitable when **relative differences** matter (e.g., growth rates).
- Interpretable in the same units as the log-transformed target; used often in Kaggle competitions.

Relative & Scaled Metrics

- Relative Absolute Error (RAE)

$$RAE = \frac{\sum |y_i - \hat{y}_i|}{\sum |y_i - \bar{y}|}$$

- Compares model performance to a **naïve baseline** (predicting the mean).
- **RAE** < 1 → model better than baseline; **RAE** > 1 → worse.

- Relative Squared Error (RSE)

$$RSE = \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

- Squared-error counterpart of RAE; measures relative performance.

Goodness-of-fit & Correlation Metrics

- Coefficient of Determination (R^2)

$$R^2 = 1 - RSE$$

- Measures the **proportion of variance explained** by the model.
- $R^2 = 1$ → perfect fit; $R^2 = 0$ → no better than predicting the mean; can be **negative** if the model performs worse than the baseline.

- Adjusted R^2

$$R^2_{adj} = 1 - (1 - R^2) \frac{n-1}{n-p-1}$$

- Penalizes for the number of predictors p; prevents misleading improvements when adding irrelevant variables.

- Correlation coefficient (Pearson's)

$$r = \frac{Cov(y, \hat{y})}{\sigma_y \cdot \sigma_{\hat{y}}}$$

- Measures **linear correlation** between predictions and actuals ($-1 \le r \le 1$).
- High positive correlation implies good alignment, but not necessarily good calibration.

Classification Metrics

Basic Metrics

- $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
  - Proportion of correct predictions overall.
  - **Advantages:** simple and intuitive.
  - **Disadvantages:** misleading with *imbalanced datasets* (when one class dominates).

- $\text{Missclassification Rate} = 1 - \text{Accuracy} = \frac{FP+FN}{TP+TN+FP+FN}$

- $\text{Precision} = \frac{TP}{TP+FP}$
  - Of all instances predicted as positive, how many are truly positive.
  - High precision = few false alarms.

- $\text{Recall (Sensitivity)} = \frac{TP}{TP+FN}$
  - Of all true positives, how many the model correctly identified.
  - High recall = few missed positives.

- $\text{Specificity} = \frac{TN}{TN+FP}$
  - Ability to correctly identify negatives.

- $F_1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
  - Harmonic mean of precision and recall.
  - Balances both metrics — useful when classes are imbalanced.

↳ generalization:
  - Generalization of F1; $\beta > 1$ favors recall, $\beta < 1$ favors precision.

$$F_\beta\text{-score} = (1+\beta^2)\frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}$$

For multiclass problems, F1 and similar metrics can be averaged in different ways:

- **Macro-average:** unweighted mean across classes (treats all classes equally).
- **Micro-average:** aggregates TP, FP, FN globally (weighted by class frequency).
- **Weighted-average:** mean weighted by support (number of instances per class).

Threshold-Dependent Metrics

Many classifiers output **probabilities**, which must be thresholded (e.g., 0.5) to make class predictions.
Metrics below depend on that threshold.

ROC Curve (Receiver Operating Characteristic)
Plots **True Positive Rate (Recall)** vs **False Positive Rate (1 – Specificity)** for different thresholds.

AUC (Area Under the ROC Curve)

- Measures the model's ability to rank positive examples higher than negative ones.
- **AUC = 1** → perfect classifier; **AUC = 0.5** → random guessing.

Precision–Recall (PR) Curve
Plots **Precision vs Recall** across thresholds.

- More informative than ROC when dealing with **highly imbalanced datasets**.

Average Precision (AP)

- Area under the PR curve; used widely in information retrieval and object detection.

- Log-loss (Cross-entropy function)

$$\text{Log-loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\hat{p}_{i,c})$$

  - Penalizes confident but wrong predictions.
  - **Lower LogLoss** = better calibrated probabilities.
  - Used as the default scoring function for many probabilistic classifiers.

- Brier Score

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

  - Measures the mean squared difference between predicted probabilities and actual outcomes (binary only).
  - Lower values indicate better calibration.

Additional Metrics

- Balanced Accuracy = $\frac{\text{Recall}_{positive} + \text{Recall}_{negative}}{2}$

- Matthews Correlation Coeff. (MCC)

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

  - Correlation coefficient between true and predicted labels (–1 to +1).
  - **Robust summary metric** even with class imbalance.

- Cohen's kappa

$$k = \frac{p_o - p_e}{1 - p_e}$$

  - Compares observed accuracy (p_o) to expected accuracy by chance (p_e).
  - Often used in medical or human-annotated tasks.

- Hamming loss = $\frac{1}{n \cdot L}\sum_{i=1}^{n}\sum_{l=1}^{L}[y_{i,l} \neq \hat{y}_{i,l}]$

  - Fraction of incorrect labels; particularly relevant for **multi-label** classification.

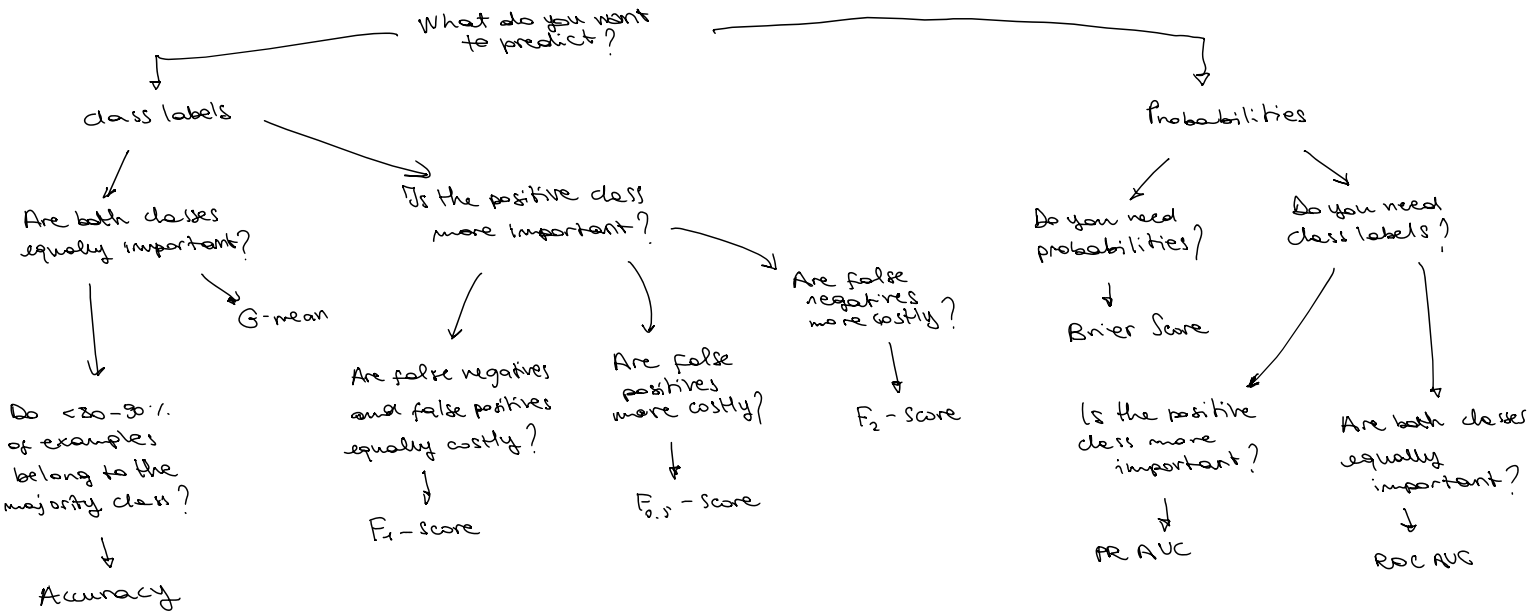
- Jaccard Index = $\frac{TP}{TP+FP+FN}$

  - Measures the overlap between predicted and actual sets.
  - Common in segmentation and multi-label problems.

Summary Table

Category	Metric	Description	Ideal Value
Basic	Accuracy, Error Rate	Overall correctness	High
Confusion-based	Precision, Recall, F1, Specificity	Trade-off between FP and FN	High
Probabilistic	LogLoss, Brier Score	Confidence calibration	Low
Threshold-free	AUC, PR-AUC, AP	Ranking performance	High
Balanced / Correlation	MCC, Kappa, Balanced Accuracy	Robustness to imbalance	High
Multi-label / Ranking	Jaccard, Hamming, MAP, NDCG	Complex scenarios	High

~ if you don't know, follow this ~



Clustering Metrics

Internal Metrics

- Silhouette coeff.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

The **Silhouette Score** for the dataset is the average of all $s(i)$.

- Range: **–1 to 1**
  - Close to **1** → well-separated and cohesive clusters
  - Around **0** → overlapping clusters
  - Negative → points assigned to the wrong cluster
- **Advantages:** interpretable and widely used.
- **Limitations:** computationally expensive for large datasets.

where:

- $a(i)$ = avg distance between i and all the other points in the same cluster

- $b(i)$ = smallest avg distance between i and points in the nearest different cluster

- Davies-Bouldin Index (DBI)

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{d_{ij}} \right)$$

- Lower values indicate **better clustering** (dense and well-separated clusters).
- **Advantages:** efficient to compute.
- **Limitations:** sensitive to noise and non-spherical clusters.

where:

- S_i = avg intra-cluster distance (i is the class)

- d_{ij} = distance between centroids i and j

- Calinski-Harabasz Index (CHI)

$$CHI = \frac{Tr(B_k)}{Tr(W_k)} \cdot \frac{n-k}{k-1}$$

- Higher values indicate **dense and well-separated clusters**.
- **Advantages:** scales well, easy to interpret.
- **Limitations:** assumes spherical clusters.

where:

- $Tr(B_k)$: between-cluster dispersion
- $Tr(W_k)$: within-cluster dispersion

- Dunn Index (DI)

$$DI = \frac{\min_{i \neq j} d(C_i, C_j)}{\max_k diam(C_k)}$$

- Ratio of the **minimum inter-cluster distance** to the **maximum intra-cluster distance**.
- Higher Dunn Index → better separation and compactness.
- **Advantages:** intuitive, simple.
- **Limitations:** sensitive to noise; difficult to compute for large datasets.

- Within-Cluster Sum of Squares (WCSS)

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

- Measures total compactness of clusters.
- Lower WCSS = tighter clusters.
- Used in the **Elbow Method** to determine the optimal number of clusters.

where μ_i is the centroid of cluster C_i

External Metrics

- Adjusted Rand Index (ARI)

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]}$$

where RI is the **Rand Index**, which measures agreement between clustering and ground truth.

- Range: **-1 to 1**
  - 1 → perfect match
  - 0 → random labeling
  - Negative → worse than random
- **Advantages:** corrected for chance; handles different numbers of clusters.

- Mutual Information (MI)

$$MI(U,V) = \sum_i \sum_j P(i,j) \log \frac{P(i,j)}{P(i)P(j)}$$

- Measures information shared between the predicted clusters (U) and true labels (V).
- **Higher MI** means better agreement.

- Normalized Mutual Information (NMI)

$$NMI = \frac{2\,MI(U,V)}{H(U) + H(V)}$$

- Normalized between 0 and 1 for easier interpretation.
- **NMI = 1** → perfect clustering; **0** → independent labeling.
- **Advantages:** robust to different cluster labelings.

- Fowlkes - Mallows Index (FMI)

$$FMI = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}$$

- Geometric mean of **precision** and **recall** for pairwise cluster membership.
- Range: 0–1
- **Advantages:** simple and symmetric.

Model evaluation is the bridge between experimentation and deployment.
Choosing the right metric is as important as choosing the right algorithm.

In real-world workflows, the best practice is to **track multiple metrics simultaneously** — accuracy and F1-score for classification, RMSE and R^2 for regression, Silhouette and DBI for clustering — and interpret them in context.

Metrics are tools, not goals: they guide decisions, highlight biases, and ensure that models serve their intended purpose effectively and responsibly.

"All models are wrong, some are useful"

~ George Box ~