Machine Learning Evaluation Metrics: A Beginner's Guide

Aniakwa Nathan

Introduction

Evaluation metrics are essential tools for assessing the performance of machine learning models. This guide outlines key metrics for classification and regression tasks, designed for beginners to understand and apply in practice.

1 Classification Metrics

For classification tasks (e.g., predicting spam or not spam), metrics evaluate how well a model assigns data to categories. These are derived from a *confusion matrix*:

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

Table 1: Confusion Matrix

- **True Positive (TP)**: Correctly predicted positive (e.g., correctly identified a disease).
- **True Negative (TN)**: Correctly predicted negative (e.g., correctly identified no disease).
- **False Positive (FP)**: Incorrectly predicted positive (e.g., predicted disease, but none exists).
- **False Negative (FN)**: Incorrectly predicted negative (e.g., missed a disease).

1.1 Key Metrics

- **Accuracy**: Proportion of correct predictions.

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

  Use when: Classes are balanced. Avoid for imbalanced data (e.g., 1% disease prevalence).

- **Precision**: Proportion of positive predictions that are correct.

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

Use when: False positives are costly (e.g., avoiding mislabeling good emails as spam).

- **Recall (Sensitivity)**: Proportion of actual positives correctly identified.

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

  Use when: False negatives are costly (e.g., missing a disease diagnosis).

- **F1-Score**: Balances precision and recall.

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

  Use when: Both precision and recall are important, especially with imbalanced data.

2 Regression Metrics

For regression tasks (e.g., predicting house prices), metrics measure the error between predicted ($\hat{y}$) and actual (y) values.

2.1 Key Metrics

- **Mean Absolute Error (MAE)**: Average absolute difference between predictions and actuals.

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

  Interpretation: An MAE of $10,000 means predictions are off by $10,000 on average.

- **Mean Squared Error (MSE)**: Average of squared differences.

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

  Use when: Large errors are particularly undesirable. Note: Units are squared (e.g., dollarsš).

- **Root Mean Squared Error (RMSE)**: Square root of MSE, in original units.

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

  Interpretation: Like MAE but more sensitive to large errors.

- **Rš (R-squared)**: Proportion of variance explained by the model (0 to 1). *Interpretation*: Rš = 0.8 means 80% of data variability is explained. Negative Rš indicates a model worse than the mean.

3 Practice Tips for Beginners

- Use simple datasets (e.g., from Kaggle or scikit-learn) for classification (e.g., spam detection) or regression (e.g., house prices).
- Use Python with scikit-learn to compute metrics (e.g., `accuracy_score`, `mean_squared_error`).
- Visualize confusion matrices using Python (e.g., seaborns heatmap).
- Compare MAE, MSE, and RMSE on the same regression dataset.
- Experiment with imbalanced data to understand why F1-score matters.
- Choose metrics based on your problem: prioritize precision, recall, or error tolerance.