IN402

# Machine Learning

Chapter

# 4

## Evaluating Classification Models

**Author:** Abbas El-Hajj Youssef

**University:** Lebanese University

**Department:** Department of Computer Science

*These notes extend course materials taught by Prof. Ahmad Faour with additional content from textbooks and supplementary resources.*

***Disclaimer:*** *This is not an official course document.*

# Contents

**Learning Objectives**

By the end of this chapter, you should be able to:

1. Construct and interpret a confusion matrix for binary classification problems
2. Explain why accuracy can be misleading, particularly for imbalanced datasets
3. Define, calculate, and interpret the trade-offs between precision and recall
4. Compute the F1-score as a balanced performance measure
5. Plot and interpret Receiver Operating Characteristic (ROC) curves
6. Use Area Under the Curve (AUC) to compare classifier performance

## 1 Introduction: Beyond Simple Accuracy

Having learned to build classification models in previous chapters, we now turn to a fundamental question: How do we rigorously evaluate their performance? The most intuitive approach—measuring the percentage of correct predictions—provides a starting point through the metric called **accuracy**. However, as we shall demonstrate, this seemingly straightforward measure can be profoundly misleading.

Consider a diagnostic model for detecting a rare disease that affects only 1 in 1000 individuals. A trivial model that invariably predicts "no disease" achieves 99.9% accuracy while failing entirely at its critical task: identifying affected patients. This paradox reveals a fundamental principle in model evaluation: **not all errors carry equal weight**.

This chapter introduces a comprehensive framework for evaluating classifiers that extends far beyond simple accuracy. We develop sophisticated metrics that address essential questions about model performance:

- When the model predicts a positive outcome, what is the probability it is correct?
- Among all actual positive cases, what proportion does the model successfully identify?
- How does model performance vary as we adjust its decision threshold?

Understanding these evaluation metrics is as crucial as the model-building process itself. These tools enable practitioners to select appropriate models, optimize parameters, and assess real-world impact with confidence.

## 2 The Confusion Matrix: Foundation of Performance Analysis

To develop a nuanced understanding of model performance, we must first categorize predictions with greater precision than simple "correct" or "incorrect" labels. The **confusion matrix** provides this essential framework by organizing predictions into four distinct categories based on their relationship to ground truth.

**The Confusion Matrix**

A confusion matrix is a structured table that comprehensively summarizes classification performance by cross-tabulating predicted labels against true labels. For binary

classification, this yields a $2 \times 2$ matrix structure.
Consider a binary classification problem with:

- **Positive (1):** Presence of a condition (e.g., disease detected, email is spam)
- **Negative (0):** Absence of a condition (e.g., healthy patient, legitimate email)

| | | Predicted Class: Positive (1) | Predicted Class: Negative (0) |
|---|---|---|---|
| Actual Class | Positive (1) | **True Positive** (TP) | **False Negative** (FN) |
| | Negative (0) | **False Positive** (FP) | **True Negative** (TN) |

The four categories are:

- **True Positive (TP):** Correctly identified positive cases
- **True Negative (TN):** Correctly identified negative cases
- **False Positive (FP):** Incorrectly predicted as positive (Type I error)
- **False Negative (FN):** Incorrectly predicted as negative (Type II error)

The diagonal elements represent correct predictions, while off-diagonal elements indicate classification errors.

These four fundamental counts form the basis for all performance metrics discussed in this chapter, enabling us to quantify different aspects of model behavior with precision.

## 3 Accuracy and Its Limitations

Accuracy represents the most intuitive performance metric, answering the straightforward question: "What proportion of predictions are correct?" While conceptually simple, this metric harbors significant limitations that can mask poor model performance.

### Accuracy and Error Rate

Accuracy measures the proportion of correct predictions across all samples:

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

The complementary metric, error rate, quantifies the proportion of incorrect predictions:

$$\text{Error Rate} = \frac{\text{Incorrect Predictions}}{\text{Total Predictions}} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \text{Accuracy}$$

While accuracy appears to provide a comprehensive performance measure, it implicitly assumes equal importance across all classes and uniform costs for different error types. This assumption proves problematic when confronting **imbalanced datasets**, where class distributions are severely skewed.

**The Accuracy Paradox**

Imbalanced datasets, characterized by significant disparities in class frequencies, can render accuracy a misleading metric. High accuracy may mask complete failure in identifying minority class instances.

**Illustrative Example: Cancer Screening**

Consider a screening program examining 1000 patients where:

- 990 patients are healthy (negative class)
- 10 patients have cancer (positive class)

A degenerate model that uniformly predicts "healthy" yields:

- TP = 0 (no cancer cases identified)
- FN = 10 (all cancer cases missed)
- TN = 990 (all healthy patients correctly identified)
- FP = 0 (no false alarms)

This model achieves:

$$\text{Accuracy} = \frac{0 + 990}{1000} = 99\%$$

Despite 99% accuracy, this model completely fails its primary objective: detecting cancer. This phenomenon, termed the **accuracy paradox**, demonstrates that high accuracy can coincide with zero predictive utility for critical minority classes.

This limitation necessitates more sophisticated metrics that explicitly consider class-specific performance and the relative importance of different error types.

## 4 Precision and Recall: Complementary Performance Perspectives

To address accuracy's limitations, we introduce two fundamental metrics that examine model performance from complementary perspectives, particularly focusing on the positive class. These metrics—precision and recall—illuminate different aspects of classification quality and help practitioners understand the nature of their model's errors.

### 4.1 Recall: Measuring Completeness of Detection

Recall addresses a critical question in many applications: Among all actual positive cases, what proportion does our model successfully identify?

**Recall (Sensitivity, True Positive Rate)**

Recall quantifies the model's ability to identify positive instances comprehensively:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{True Positives}}{\text{All Actual Positives}}$$

This metric is also known as **sensitivity** or the **true positive rate (TPR)**.

**Applications Requiring High Recall**

High recall becomes paramount when false negatives carry severe consequences. Missing positive cases can be costly or dangerous in various domains:

- **Medical Screening:** Failing to detect a disease may result in delayed treatment and adverse outcomes. Comprehensive detection takes precedence over occasional false alarms.
- **Fraud Detection:** Undetected fraudulent transactions directly impact financial losses. Organizations prefer investigating legitimate transactions over missing actual fraud.
- **Security Screening:** Missing a security threat could have catastrophic consequences, justifying heightened sensitivity even at the cost of increased false alarms.

These applications prioritize minimizing false negatives, making recall the primary optimization target.

## 4.2 Precision: Measuring Prediction Reliability

While recall focuses on completeness, precision addresses reliability: When our model predicts a positive outcome, how confident can we be in that prediction?

**Precision**

Precision measures the reliability of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{True Positives}}{\text{All Predicted Positives}}$$

**Applications Requiring High Precision**

High precision becomes essential when false positives incur significant costs or consequences:

- **Email Filtering:** Incorrectly classifying important emails as spam may cause users to miss critical communications. Users expect high confidence in spam classifications.
- **Search Engine Ranking:** Presenting irrelevant results prominently degrades user experience and erodes trust. Top-ranked results must demonstrate high relevance.
- **Legal Evidence Classification:** Incorrectly flagging evidence as incriminating could have serious legal implications, necessitating high confidence in positive classifications.

These scenarios demand minimizing false positives, making precision the key performance indicator.

## 4.3 The Fundamental Trade-off

An inherent tension exists between precision and recall, reflecting a fundamental trade-off in classification decisions. Most probabilistic classifiers generate continuous scores that require

thresholding for discrete predictions. Adjusting this threshold directly impacts the precision-recall balance:

- **Lowering the threshold** (e.g., from 0.5 to 0.3) increases positive predictions, thereby:
  - Improving recall by capturing more true positives
  - Reducing precision by introducing more false positives
- **Raising the threshold** (e.g., from 0.5 to 0.8) restricts positive predictions, thereby:
  - Improving precision by reducing false positives
  - Reducing recall by missing some true positives

This trade-off reflects an unavoidable reality: increasing our net to catch more positive cases inevitably captures more negative cases as well. The optimal balance depends entirely on the specific application context and the relative costs of different error types.

## 5 F-Score: Harmonizing Precision and Recall

While precision and recall provide valuable insights individually, comparing models often requires a unified metric that considers both perspectives. The F-score family of metrics addresses this need by combining precision and recall into a single, interpretable measure.

### 5.1 The F1-Score: Balanced Performance

The F1-score represents the most widely used member of the F-score family, providing equal weight to precision and recall through their harmonic mean.

**F1-Score**

The F1-score balances precision and recall through their harmonic mean:

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This metric achieves high values only when both precision and recall are simultaneously high, making it particularly valuable for imbalanced datasets.

**The Significance of Harmonic Mean**

The harmonic mean penalizes extreme imbalances between precision and recall more severely than the arithmetic mean. Consider two illustrative models:

- **Model A:** Precision = 0.9, Recall = 0.9
  - F1-Score = 0.9
  - Arithmetic mean = 0.9
- **Model B:** Precision = 1.0, Recall = 0.1
  - F1-Score = $2 \cdot \frac{1.0 \times 0.1}{1.0 + 0.1} \approx 0.18$
  - Arithmetic mean = 0.55

Despite perfect precision, Model B's poor recall results in a low F1-score, correctly identifying it as inadequate. The arithmetic mean would misleadingly suggest moderate performance.

### 5.2 The General $F_\beta$ Score: Weighted Performance

Real-world applications often require prioritizing either precision or recall based on domain-specific considerations. The $F_\beta$ score generalizes the F1-score to accommodate such preferences.

**$F_\beta$ Score**

The $F_\beta$ score provides a weighted harmonic mean of precision and recall:

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

The parameter $\beta$ controls the relative importance of recall versus precision:

- $\beta = 1$: Equal weighting (standard F1-score)
- $\beta > 1$: Emphasizes recall (e.g., $\beta = 2$ for medical diagnosis)
- $0 < \beta < 1$: Emphasizes precision (e.g., $\beta = 0.5$ for spam filtering)

This flexibility allows practitioners to encode domain knowledge and business requirements directly into their evaluation metric.

## 6 ROC Curves and AUC: Comprehensive Performance Visualization

While precision, recall, and F-scores provide valuable insights at specific operating points, they depend on the chosen classification threshold. The Receiver Operating Characteristic (ROC) curve transcends this limitation by visualizing performance across all possible thresholds simultaneously.

### 6.1 Understanding the ROC Space

The ROC curve maps classifier performance in a two-dimensional space that captures the fundamental trade-off between correctly identifying positives and avoiding false alarms.

**ROC Curve Components**

The ROC curve plots two key rates as the classification threshold varies:

1. **True Positive Rate (TPR)**: Plotted on the y-axis, equivalent to recall

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

2. **False Positive Rate (FPR)**: Plotted on the x-axis, measuring false alarm rate

$$\text{FPR} = \frac{FP}{FP + TN} = \frac{\text{False Positives}}{\text{All Actual Negatives}}$$

The curve is constructed by varying the classification threshold from its maximum to minimum value, calculating the corresponding (FPR, TPR) pair at each point, and connecting these points to form a continuous curve.

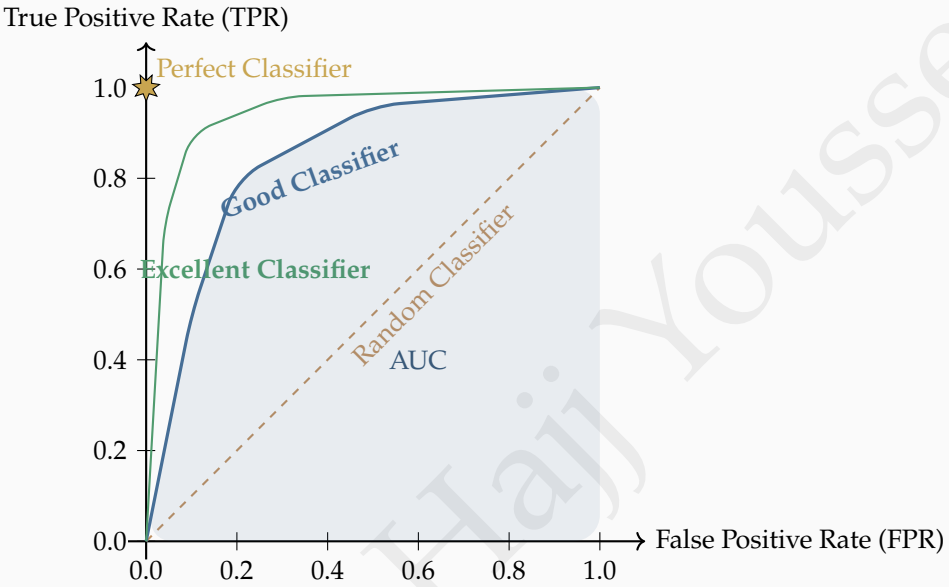


**Figure 1:** *ROC curves comparing classifier performance. Superior models produce curves that bow toward the upper-left corner, maximizing true positive rate while minimizing false positive rate.*

## 6.2 Complementary Metrics: Specificity and Its Relationship to FPR

Understanding the ROC curve benefits from examining the complementary relationship between false positive rate and specificity.

**Specificity (True Negative Rate)**

Specificity measures the model's ability to correctly identify negative instances:

$$\text{Specificity} = \text{TNR} = \frac{TN}{TN + FP} = \frac{\text{True Negatives}}{\text{All Actual Negatives}}$$

**The FPR-Specificity Duality**

False positive rate and specificity are complementary metrics that sum to unity:

$$\text{FPR} + \text{Specificity} = \frac{FP}{FP + TN} + \frac{TN}{FP + TN} = 1$$

This relationship means the ROC curve implicitly represents specificity along its x-axis as (1 - Specificity). Medical literature often emphasizes this by labeling axes as "Sensitivity" versus "1 - Specificity," highlighting the trade-off between correctly identifying positives and negatives.

### 6.3 Quantifying Overall Performance: Area Under the Curve

While ROC curves provide rich visual information, comparing multiple models requires a scalar summary metric. The Area Under the Curve (AUC) serves this purpose elegantly.

**Area Under the Curve (AUC)**

AUC represents the total area beneath the ROC curve, providing a threshold-independent measure of classifier quality:

- **AUC = 1.0**: Perfect classification
- **AUC = 0.5**: Performance equivalent to random guessing
- **AUC < 0.5**: Systematic misclassification (predictions inversely correlated with truth)

**The Probabilistic Interpretation**

AUC possesses an elegant statistical interpretation:

*AUC equals the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance.*

This interpretation provides intuitive understanding: a perfect classifier (AUC=1.0) always assigns higher scores to positives than negatives, while a random classifier (AUC=0.5) provides no ranking information.

## 7 Constructing an ROC Curve: A Detailed Example

To solidify understanding, we present a complete worked example demonstrating ROC curve construction from raw classifier outputs.

### 7.1 Initial Setup and Data Preparation

Consider a binary classifier evaluated on 10 instances, with equal class distribution (5 positive, 5 negative). The model outputs probability scores that we must threshold for final predictions.

**Table 1:** *Classifier outputs: true labels and predicted probabilities*

| Instance | True Label ($y$) | Predicted Probability ($h_\theta(x)$) |
|---|---|---|
| 1 | 1 | 0.90 |
| 2 | 0 | 0.80 |
| 3 | 1 | 0.70 |
| 4 | 0 | 0.60 |
| 5 | 1 | 0.55 |
| 6 | 0 | 0.40 |
| 7 | 1 | 0.30 |
| 8 | 0 | 0.20 |
| 9 | 1 | 0.15 |
| 10 | 0 | 0.10 |

## 7.2 The Threshold Sweep Algorithm

We systematically vary the classification threshold to generate ROC curve points:

**ROC Construction Algorithm**

1. Initialize with threshold $\tau > 1.0$, yielding no positive predictions: (FPR, TPR) = (0, 0)
2. Sort instances by predicted probability in descending order
3. For each unique probability value:
   - Set it as the new threshold
   - Classify all instances with probability $\geq \tau$ as positive
   - Count TP, FP, TN, FN based on true labels
   - Calculate TPR = TP/P and FPR = FP/N
   - Record the (FPR, TPR) point
4. Connect points to form the complete ROC curve

## 7.3 Computing ROC Points

[Understanding the Calculations] For each threshold $\tau$:

- **Predicted Positive**: All instances with probability $\geq \tau$
- **Predicted Negative**: All instances with probability $< \tau$
- **TP**: True label = 1 AND probability $\geq \tau$ (cumulative count)
- **FP**: True label = 0 AND probability $\geq \tau$ (cumulative count)
- **FN**: True label = 1 AND probability $< \tau$ = Total Positives - TP = 5 - TP
- **TN**: True label = 0 AND probability $< \tau$ = Total Negatives - FP = 5 - FP

Following the algorithm, we calculate performance at each threshold:

**Table 2:** *ROC point calculation through threshold variation*

| Threshold $\tau$ | True Label | Pred. Prob. | Confusion Matrix TP | FP | FN | TN | Rates TPR | FPR |
|---|---|---|---|---|---|---|---|---|
| $> 0.90$ | - | - | 0 | 0 | 5 | 5 | 0.0 | 0.0 |
| 0.90 | 1 | 0.90 | 1 | 0 | 4 | 5 | 0.2 | 0.0 |
| 0.80 | 0 | 0.80 | 1 | 1 | 4 | 4 | 0.2 | 0.2 |
| 0.70 | 1 | 0.70 | 2 | 1 | 3 | 4 | 0.4 | 0.2 |
| 0.60 | 0 | 0.60 | 2 | 2 | 3 | 3 | 0.4 | 0.4 |
| 0.55 | 1 | 0.55 | 3 | 2 | 2 | 3 | 0.6 | 0.4 |
| 0.40 | 0 | 0.40 | 3 | 3 | 2 | 2 | 0.6 | 0.6 |
| 0.30 | 1 | 0.30 | 4 | 3 | 1 | 2 | 0.8 | 0.6 |
| 0.20 | 0 | 0.20 | 4 | 4 | 1 | 1 | 0.8 | 0.8 |
| 0.15 | 1 | 0.15 | 5 | 4 | 0 | 1 | 1.0 | 0.8 |
| 0.10 | 0 | 0.10 | 5 | 5 | 0 | 0 | 1.0 | 1.0 |

**Step-by-Step Calculation Example**

Consider threshold $\tau = 0.55$:

1. **Identify predictions**: All instances with probability $\geq 0.55$ are predicted positive
2. **Count predictions**: Instances 1, 2, 3, 4, 5 have probability $\geq 0.55$
3. **Determine TP**: Of these, instances 1, 3, 5 have true label = 1, so TP = 3
4. **Determine FP**: Instances 2, 4 have true label = 0, so FP = 2
5. **Calculate FN**: Total positives - TP = 5 - 3 = 2
6. **Calculate TN**: Total negatives - FP = 5 - 2 = 3
7. **Compute rates**: TPR = 3/5 = 0.6, FPR = 2/5 = 0.4

## 7.4 Visualizing and Computing AUC

The resulting ROC curve connects the calculated points:

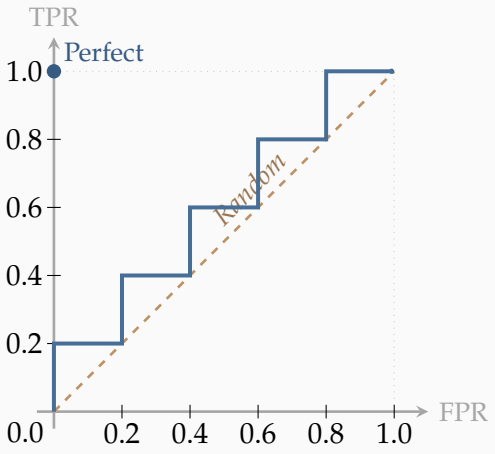


**Figure 2:** *ROC curve constructed from the example data, showing the characteristic step pattern from discrete test sets*

The AUC is calculated by summing the areas of trapezoids formed beneath the curve. For

our example, this yields AUC = 0.60, indicating modest discriminative ability above random chance.

## 8 Chapter Summary and Practical Guidelines

This chapter has developed a comprehensive framework for evaluating binary classifiers, progressing from simple accuracy to sophisticated threshold-independent metrics. Each metric serves specific purposes and addresses particular limitations of simpler approaches.

**Metric Selection Guidelines**

Choose evaluation metrics based on your specific application requirements:

**Accuracy:** Appropriate only for balanced datasets with symmetric error costs

**Precision:** Prioritize when false positives are costly (e.g., spam filtering, legal applications)

**Recall:** Prioritize when false negatives are dangerous (e.g., disease screening, fraud detection)

**F1-Score:** Use for balanced performance assessment, particularly with imbalanced datasets

**AUC:** Ideal for comparing models' overall discriminative ability independent of threshold selection

The ROC curve and AUC provide particularly powerful tools for understanding classifier behavior across all operating points. By visualizing the trade-off between true positive and false positive rates, practitioners can make informed decisions about threshold selection based on application-specific requirements.

**Advanced Considerations**

While ROC curves and AUC serve well for most applications, extreme class imbalance may warrant alternative approaches. When positive instances are exceedingly rare, small changes in false positive rate can translate to large absolute numbers of false positives. In such cases, precision-recall curves may provide more informative visualizations. Additionally, cost-sensitive evaluation incorporating explicit misclassification costs offers another avenue for domain-specific assessment.

Mastery of these evaluation techniques empowers practitioners to move beyond superficial performance measures, enabling nuanced assessment that aligns with real-world objectives and constraints.

# A Notation Reference

| Symbol | Description |
|---|---|
| TP, TN, FP, FN | True/False Positives, True/False Negatives |
| $P$ | Total actual positive instances ($P$ = TP + FN) |
| $N$ | Total actual negative instances ($N$ = FP + TN) |
| $h_\theta(\boldsymbol{x})$ | Model's predicted probability for input $\boldsymbol{x}$ |
| $\tau$ | Classification threshold |
| TPR | True Positive Rate (Sensitivity, Recall) |
| FPR | False Positive Rate |
| TNR | True Negative Rate (Specificity) |
| ROC | Receiver Operating Characteristic |
| AUC | Area Under the Curve |

**Table 3:** *Key notation and abbreviations used throughout this chapter*