Measuring Classifier Performance

Introduction

In machine learning classification problems, developing a model is only half the task. The more important part is **evaluating how good the model is**. Simply looking at accuracy is usually not enough. Real-world data can be imbalanced (such as fraud detection, medical diagnosis), and accuracy can become misleading.

Therefore, we need **systematic evaluation methods**, including:

- Confusion matrix
- Performance metrics (accuracy, precision, recall, etc.)
- ROC curve and AUC
- Cross-validation
- Statistical comparison of classifiers

This document explains each concept clearly and shows how these tools help in assessing classifier quality.

Confusion Matrix

The confusion matrix is the foundation for almost all evaluation metrics.

For **binary classification**, it is represented as:

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

True Positive (TP)

Model predicts “positive” and the actual class is also positive.

False Positive (FP)

Model predicts “positive” but the actual class is negative.

This is also known as **Type-I Error**.

False Negative (FN)

Model predicts “negative” but actual class is positive.

This is **Type-II Error**, serious in medical cases.

True Negative (TN)

Model correctly predicts “negative”.

Basic Evaluation Metrics

All major evaluation metrics come from TP, FP, FN, TN values.

Accuracy

Accuracy=TP+TN+FP+FNTP+TN

Accuracy measures how many total predictions were correct.

When accuracy works well:

- Balanced datasets
- Equal cost of FP and FN

When accuracy is misleading:

- Imbalanced datasets
  Example: 95% negative, 5% positive
  Predicting all negatives gives 95% accuracy but 0% usefulness.

Precision

Precision=TP+FPTP

Precision answers:

"Out of all predicted positives, how many are correct?"

Useful when:

- False positives are costly
  Examples:
- Fraud detection
- Spam filtering
- Criminal identification

Recall (Sensitivity / True Positive Rate)

Recall=TP+FNTP

Recall answers:

"Out of all actual positives, how many did we identify?"

Useful when:

- False negatives are dangerous
  Examples:
- Cancer detection
- Safety alarms
- Loan default detection

Specificity (True Negative Rate)

Specificity=TN+FPTN

Measures how many actual negatives were correctly classified.

Important in:

- Medical screening (avoid unnecessary panic)
- Legal decisions (avoid false accusations)

F1-Score

F1=2×Precision+RecallPrecision×Recall

F1-score gives a balance between precision and recall.

Useful when:

- Dataset is imbalanced
- Need to consider both FP and FN
- Evaluating text or fraud classification models

ROC Curve and AUC

The **Receiver Operating Characteristic (ROC)** curve shows the performance of a classifier across different thresholds.

ROC axes:

- X-axis = **False Positive Rate (FPR)**

  FPR=FP+TNFP
- Y-axis = **True Positive Rate (Recall)**

  TPR=TP+FNTP

The curve shows the trade-off between TPR and FPR.

AUC (Area Under Curve)

AUC summarizes the ROC curve into one number between 0 and 1.

AUC Interpretation:

AUC	Meaning
1.0	Perfect classifier
0.90–1.0	Excellent
0.80–0.90	Good
0.70–0.80	Fair
0.60–0.70	Poor
0.50	Random guess
< 0.50	Worse than random

AUC shows how well the classifier differentiates classes.

Why AUC is powerful:

- Works well for imbalanced data
- Independent of classification threshold
- Measures ranking ability of the model

Example:
A model with **AUC = 0.95** separates positive and negative classes extremely well.

Cross Validation and Resampling Methods

Cross validation (CV) helps us **reliably estimate** a model's performance.

k-fold Cross Validation

Dataset is divided into *k* equal folds.

Steps:

1. Choose **k** (commonly 5 or 10).
2. Use **k-1 folds** for training.
3. Use **1 fold** for validation.
4. Repeat k times, changing the validation fold each time.
5. Take the **average performance**.

Why k-fold CV is used?

- More reliable than a single train-test split
- Uses entire dataset for training and testing
- Reduces variance of performance scores

Stratified k-fold Cross Validation

Ensures **equal class distribution** in all folds.
Best for **imbalanced datasets**.

Leave-One-Out Cross Validation (LOOCV)

Special case of k-fold where k = number of samples.
Gives extremely accurate but very slow estimation.

Statistical Evaluation of Classifier Performance

To compare two classifiers, we need statistical tests.

Hypothesis Testing for Classifiers

Null Hypothesis (H_0)

"There is no significant difference between the two classifiers."

Alternative Hypothesis (H_1)

"There is a significant difference."

Common Tests

t-test

Compares mean accuracy of two classifiers across cross-validation folds.

McNemar's Test

Used when two classifiers are compared on the **same dataset**.
Especially useful for binary classification.

Comparing Two Algorithms: Example

Suppose we compare:

- Classifier A (SVM)
- Classifier B (Random Forest)

Using 10-fold CV, we compute accuracy for both models.
Example results:

Fold	SVM	Random Forest
1	0.88	0.91
2	0.85	0.90
…	…	…
10	0.87	0.92

Random Forest has consistently higher accuracy.

But…

To claim it is **significantly better**, we apply a **paired t-test**.

If $p < 0.05$ → Random Forest is statistically superior.

Case Study: Classifying Spam Emails

Dataset: 5000 emails
Classes:

- 1 = Spam
- 0 = Not Spam

We train two models:

- Logistic Regression
- Random Forest

After testing:

Metric	Logistic Regression	Random Forest
Accuracy	93%	96%
Precision	0.89	0.95
Recall	0.84	0.92
F1-score	0.86	0.93
AUC	0.94	0.98

Interpretation:

- Random Forest outperforms LR in all metrics.
- High precision → fewer false spam alerts
- High recall → more spam detected correctly
- AUC = 0.98 → Excellent separability

Thus, **Random Forest > Logistic Regression** for this dataset.

Important Considerations in Evaluation

1. Use confusion matrix for detailed view

Accuracy alone may hide important weaknesses.

2. Use precision and recall for imbalanced data

For fraud or disease detection, recall is more important.

3. Always use cross-validation

Single train-test split is unreliable.

4. Avoid data leakage

Train set must never see test information.

5. Report confidence intervals

For example:
Accuracy = 93% ± 2%.

6. Use statistical tests when comparing models

To ensure differences are significant and not due to random chance.