

# Model Evaluation Metrics

Understanding Performance in Machine Learning

A Day in the Life of a Data Scientist

# Confusion Matrix: The Foundation

All metrics start from the confusion matrix, which shows prediction outcomes:

**TP** **True Positives:** Correctly predicted positive

**TN** **True Negatives:** Correctly predicted negative

**FP** **False Positives:** Incorrectly predicted positive (Type I error)

**FN** **False Negatives:** Incorrectly predicted negative (Type II error)

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

# Accuracy

## What is it?

The proportion of correct predictions among all predictions made.

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

## Example:

80 correct, 20 incorrect → Accuracy = 80%

## When to use?

- Balanced datasets
- All errors equally important
- Quick performance assessment

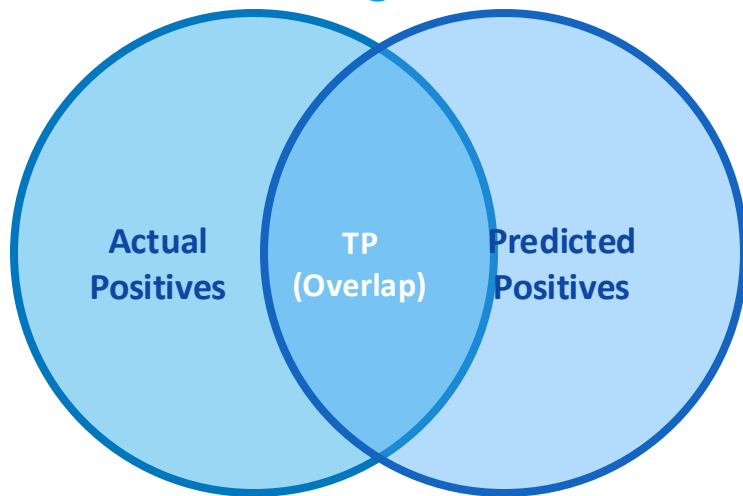
## Limitations:

- Misleading with imbalanced classes
- Doesn't distinguish error types

**Warning:** 99% accuracy on 1% fraud data means always predicting "no fraud"!

# Precision & Recall

## Visual Understanding



Recall =  $TP / \text{This Circle}$

Precision =  $TP / \text{This Circle}$

**Precision:** Of what I predicted positive, how much was actually positive?

**Recall:** Of what was actually positive, how much did I catch?

## Precision

$$\text{Precision} = TP / (TP + FP)$$

Use when false positives are costly

- Spam filtering: Don't mark good emails as spam
- Medical tests: Avoid unnecessary treatments

## Recall (Sensitivity)

$$\text{Recall} = TP / (TP + FN)$$

Use when false negatives are costly

- Disease screening: Don't miss sick patients
- Fraud detection: Catch all fraud cases

**The Precision-Recall Tradeoff: Improving one often decreases the other!**

# F1-Score: The Harmonic Mean

## What is it?

Harmonic mean of Precision and Recall, balancing both into a single score.

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

Punishes extreme imbalance and forces balance between both metrics.

## When to use?

- Single balanced metric needed
- Imbalanced datasets
- Both error types matter

## Example:

**Model A:** P=0.9, R=0.5 → F1=0.64

**Model B:** P=0.7, R=0.7 → F1=0.70

Model B wins!

# The Imbalanced Dataset Problem

## What is Label Imbalance?

When one class significantly outnumbers the other in training data.

### Common Examples:

- **Fraud:** 99% legitimate, 1% fraud
- **Disease:** 95% healthy, 5% sick
- **Churn:** 90% retain, 10% leave

### The Problem:

Models predict majority class and ignore the minority (often what we care about!).

**Result:** High accuracy, terrible recall on minority class!

## SMOTE to the Rescue

### Synthetic Minority Over-sampling Technique

#### How it works:

Creates realistic "fake" examples of the minority class by blending similar real examples together.

**Example:** If you have 2 fraud cases with similar patterns, SMOTE creates new synthetic fraud examples with features "in between" those two real cases.

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_train, y_train = smote.fit_resample(X, y)
```

# From Probabilities to Predictions

## The Decision Threshold

### Models Output Probabilities

ML models don't predict "yes" or "no" directly. They output a probability between 0.0 and 1.0.

#### Example predictions:

- Patient A:  $P(\text{disease}) = 0.95$
- Patient B:  $P(\text{disease}) = 0.62$
- Patient C:  $P(\text{disease}) = 0.31$
- Patient D:  $P(\text{disease}) = 0.08$

### We set a threshold!

Choose a cutoff value. If probability  $\geq$  threshold, predict positive.

#### If threshold = 0.5:

- Patient A (0.95) → Predict: **Positive**
- Patient B (0.62) → Predict: **Positive**
- Patient C (0.31) → Predict: **Negative**
- Patient D (0.08) → Predict: **Negative**

#### Different thresholds = Different results!

Lower threshold → More positives predicted (higher recall, lower precision)

Higher threshold → Fewer positives predicted (lower recall, higher precision)

# ROC-AUC: Evaluating All Thresholds

## ROC Curve

**Receiver Operating Characteristic:** Shows model performance at every possible threshold.

$TPR = Recall$

$FPR = FP / (FP + TN)$

Each point on the curve represents a different threshold choice.

## AUC Score

**Area Under Curve:** Performance across all thresholds.

1.0 = Perfect | 0.9+ = Excellent | 0.8+ = Good | 0.5 = Random

## ROC Curve Visualization

### ROC Curve Comparison

