


Indian Institute of Technology Madras
ZANZIBAR CAMPUS

Z3003 · DATA MINING: SCALING TO LARGE DATA

# Evaluating Classifiers: Beyond Accuracy

*Wednesday 4 November 2026 · 50 minutes*

**Dr. Innocent Nyalala** | BS Semester 5 (DSAI) | IIT Madras Zanzibar

# By the End of Today You Can

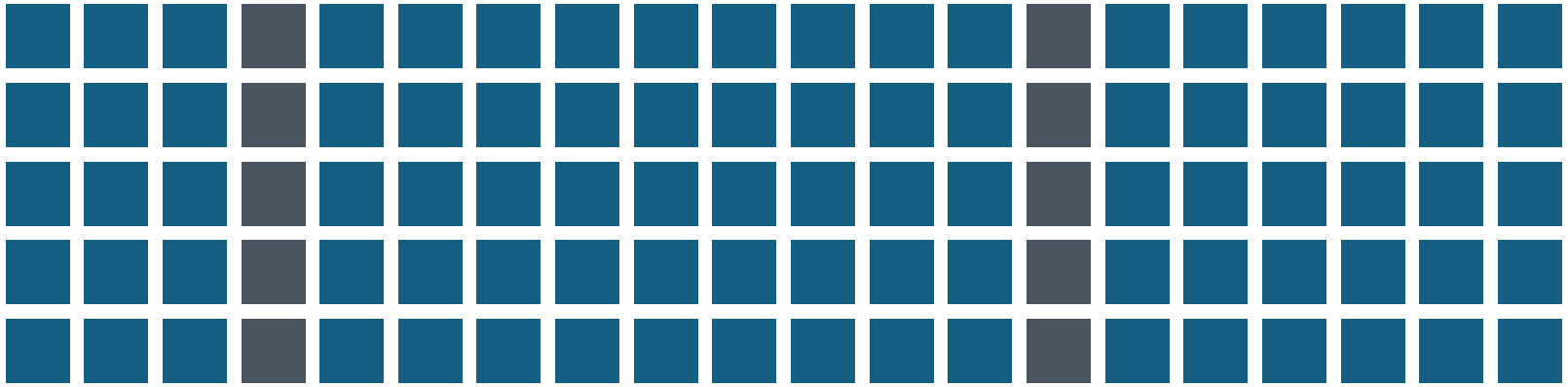
1. Explain why accuracy can lie when one class is rare
2. Build a confusion matrix and name TP, FP, FN, TN
3. Compute precision, recall and F1 by hand
4. Choose a threshold by the cost of each mistake
5. Read an ROC curve and explain AUC

# Today's Plan (40 minutes + questions)

IIT Madras Zanzibar

| minutes | part | what you get |
|---|---|---|
| 0 – 5 | Why accuracy can lie | a 90% model that helps nobody |
| 5 – 12 | The confusion matrix | four kinds of right and wrong |
| 12 – 20 | Precision, recall, F1 | computed step by step |
| 20 – 27 | Thresholds and ROC | trading false alarms for finds; AUC |
| 27 – 35 | The wider toolbox + Weka | MCC, kappa, macro/micro, costs, PR and lift curves |
| 35 – 40 | Code, exercise, summary | then questions and further reading |

# A Rare Event: Failed Seasons



**10.2%**
of seasons failed
5,816 of 56,988 in test

**100 farm-seasons: about 10 failed (dark grey), 90 were OK (teal)**

**Definition: failed season = yield in the bottom 10% for its crop: the farmers who need help**

**When the class we care about is rare, accuracy can fool us. Here is how.**

# A 90% Accurate Model That Helps Nobody

IIT Madras Zanzibar

**The lazy model**
says 'OK' for
every farm

**89.8%**
accuracy

**0**
struggling
farmers found

**accuracy = farms it gets right ÷ all farms = 51,172 OK farms ÷ 56,988 = 89.8%**

> **90% accurate and completely useless.**
> **We need measures that look at the rare class directly.**

# The Confusion Matrix: Four Outcomes

IIT Madras Zanzibar

| | predicted: FAILED | predicted: OK |
|---|---|---|
| actually FAILED | **TP true positive**<br>failed, flagged: found | **FN false negative**<br>failed, said OK: MISSED |
| actually OK | **FP false positive**<br>OK, flagged: false alarm | **TN true negative**<br>OK, said OK: left alone |

**Smoke alarm:**

**missed fire = FN**

**burnt toast alarm = FP**

**Definition: positive = the class we are searching for (failed season, fraud, disease), NOT 'good news'**

**Every measure today is built from these four counts.**

# Our Tree's Confusion Matrix

IIT Madras Zanzibar

| | predicted: FAILED | predicted: OK |
|---|---|---|
| actually FAILED | **TP 683** found | **FN 5,133** missed |
| actually OK | **FP 545** false alarms | **TN 50,627** left alone |

**Depth-8 tree**
**56,988 test seasons**
**flag if risk ≥ 0.5**

**The huge TN box (50,627) dominates accuracy and hides the 5,133 missed farmers.**

# Accuracy, Computed From the Matrix

IIT Madras Zanzibar

**accuracy = (TP + TN) ÷ all predictions**

| step | what | working |
|---|---|---|
| 1 | correct ones | 683 + 50,627 = 51,310 |
| 2 | divide by all | 51,310 ÷ 56,988 = 90.0% |
| **3** | **compare: lazy model** | **89.8%  (tree only 0.2 points better)** |

> **Accuracy says 'good'. It cannot see the 5,133 missed farmers.**

# Recall: Of the Real Failures, How Many Did We Find?

IIT Madras Zanzibar

**recall = TP ÷ (TP + FN)  'of the real failures, how many did we find?'**

| step | what | working |
|---|---|---|
| 1 | real failures (top row) | TP + FN = 683 + 5,133 = 5,816 |
| 2 | found | TP = 683 |
| **3** | **recall** | **683 ÷ 5,816 = 11.7%** |

**Recall 11.7%: the tree finds only about 1 in 9 struggling farms. Also called sensitivity.**

# Precision: Of Our Alarms, How Many Were Right?

IIT Madras Zanzibar

**precision = TP ÷ (TP + FP)** **'of our alarms, how many were right?'**

| step | what | working |
|---|---|---|
| 1 | all alarms (left column) | TP + FP = 683 + 545 = 1,228 |
| 2 | right alarms | TP = 683 |
| **3** | **precision** | **683 ÷ 1,228 = 55.6%** |

**Precision 55.6%: about half our alarms are real. Recall 11.7%: we miss most failures.**

# Recall and Precision in One Picture: the Fishing Net

IIT Madras Zanzibar

## Recall

Of all the fish in the lake,
what share did the net catch?

TP ÷ (TP + FN)

## Precision

Of everything in the net,
what share is fish (not boots)?

TP ÷ (TP + FP)

## Small net

high precision, low recall: our tree at 0.5

## Huge net

high recall, low precision: many false alarms

# F1: One Number That Needs Both

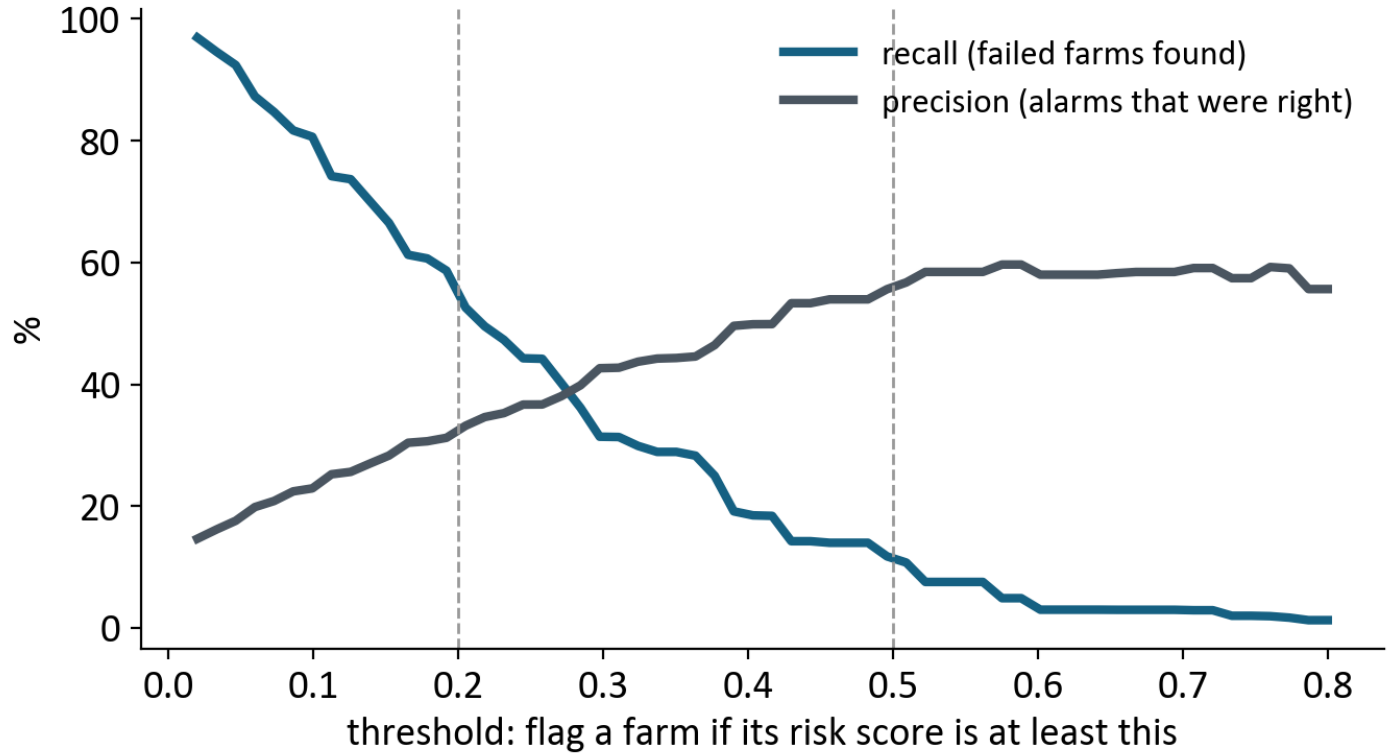
IIT Madras Zanzibar

**F1 = 2 × precision × recall ÷ (precision + recall)**

| step | what | working |
|---|---|---|
| 1 | top | 2 × 0.556 × 0.117 = 0.130 |
| 2 | bottom | 0.556 + 0.117 = 0.673 |
| 3 | F1 | 0.130 ÷ 0.673 = 0.19 |

**Plain average (0.34) flatters the model. F1 = 0.19 is high only when BOTH are high. Lazy model: F1 = 0.**

# The Threshold: When Do We Raise the Alarm?




**Risk score** — the tree gives each farm a number from 0 to 1

**Threshold** — flag the farm if score ≥ threshold

**Lower it** — bigger net: more finds, more false alarms

**The threshold is OUR choice, not the model's. Choose it by the cost of each mistake.**

# Moving the Threshold From 0.5 to 0.2

IIT Madras Zanzibar

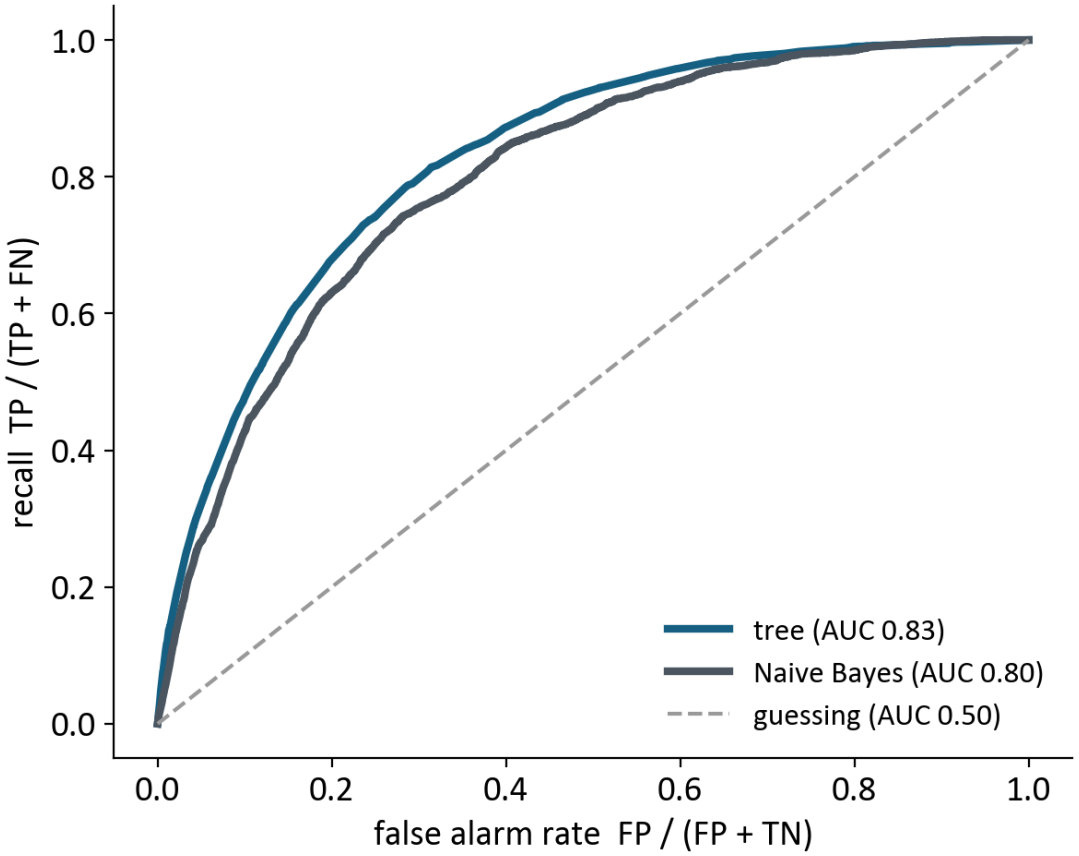
| | threshold 0.5 | threshold 0.2 |
|---|---|---|
| **farmers found (TP)** | 683 | 3,070 |
| **missed (FN = 5,816 − TP)** | 5,133 | 2,746 |
| **false alarms (FP)** | 545 | 6,191 |
| **left alone (TN = 51,172 − FP)** | 50,627 | 44,981 |
| **precision TP ÷ (TP + FP)** | 56% | 3,070 ÷ 9,261 = 33% |
| **recall TP ÷ 5,816** | 12% | 3,070 ÷ 5,816 = 53% |
| **F1** | 0.19 | 0.41 |
| **accuracy (TP + TN) ÷ 56,988** | 90.0% | 84.3% |

**3,070 ÷ 683 = 4.5 × more farmers found.**

**Accuracy fell, usefulness rose.**

**Seed sent unnecessarily is cheap; a missed failed harvest can mean hunger. Lower the threshold.**

# The ROC Curve: Every Threshold at Once




| | |
|---|---|
| **Side** | recall: share of failures found |
| **Bottom** | false alarm rate: FP ÷ (FP + TN) |
| **Each point** | one threshold |
| **Best** | bends towards the top-left corner |

**The diagonal is guessing. The further the curve bends to the top-left, the better.**

# AUC: The Area Under the Curve

IIT Madras Zanzibar

**AUC = the chance that a random FAILED farm gets a higher risk score than a random OK farm.**

**0.5**

guessing: a coin toss

**1.0**

perfect ranking

**Ours**

tree 0.83 · Naive Bayes 0.80

**AUC compares models over ALL thresholds, before you choose one.**

# Which Measure When?

IIT Madras Zanzibar

| situation | use | example |
|---|---|---|
| balanced classes, equal costs | accuracy | High vs Low yield |
| missing a positive is costly | recall | malaria screening; failed harvests |
| false alarms are costly | precision | blocking a mobile-money account |
| one number, rare class | F1 | comparing models on failures |
| compare over all thresholds | AUC | tree vs Naive Bayes |

> **Always report the confusion matrix: every other measure comes from its four numbers.**

# Evaluation in Python

IIT Madras Zanzibar

```
 1  from sklearn.metrics import confusion_matrix, roc_auc_score
 2  from sklearn.metrics import precision_score, recall_score, f1_score
 3  tree = DecisionTreeClassifier(max_depth=8).fit(Xtr, ytr)        # y: 1 = failed
 4  risk = tree.predict_proba(Xte)[:, 1]                             # risk score 0..1
 5  pred = (risk >= 0.2).astype(int)                                 # our threshold
 6  print(confusion_matrix(yte, pred))      # [[TN FP] [FN TP]]
 7  print("precision", precision_score(yte, pred))                   # 0.33
 8  print("recall   ", recall_score(yte, pred))                      # 0.53
 9  print("F1       ", f1_score(yte, pred))                          # 0.41
10  print("AUC      ", roc_auc_score(yte, risk))                     # 0.83
```

| line | what it does |
|---|---|
| 4 | risk scores |
| 5 | threshold 0.2 |
| 6 | confusion matrix |
| 7 – 9 | P, R, F1 |
| 10 | AUC (uses scores) |

**Careful: scikit-learn prints the matrix as [[TN, FP], [FN, TP]], negatives first.**

# Evaluation in the Real World

IIT Madras Zanzibar

### Medical tests

sensitivity (recall) and specificity

### Mobile money

precision: wrong blocks anger customers

### Credit scoring

AUC is the industry standard

### Spam filters

precision: a lost real email hurts

### Food security

recall: missing a district is costly

### Our farms

recall: find struggling farmers

**Decide which mistake is worse BEFORE choosing the measure.**

# More Measures From the Same Four Counts

IIT Madras Zanzibar

| measure | how | threshold 0.5 | threshold 0.2 |
|---|---|---|---|
| **specificity** | TN ÷ (TN + FP) = 50,627 ÷ 51,172 | 98.9% | 87.9% |
| **false alarm rate** | FP ÷ (FP + TN) = 1 − specificity | 1.1% | 12.1% |
| **balanced accuracy** | (recall + specificity) ÷ 2 | 55.3% | 70.3% |
| **MCC** | uses all four counts (worked out on the next slide) | 0.22 | 0.33 |
| **Cohen's kappa** | accuracy beyond chance (next slide) | 0.16 | 0.32 |

> **For a rare class, report balanced accuracy or MCC next to the confusion matrix, never accuracy alone.**

# MCC and Kappa, Worked Out Step by Step

IIT Madras Zanzibar

**Threshold 0.5: TP = 683, FN = 5,133, FP = 545, TN = 50,627, all = 56,988**

| step | working | result |
|---|---|---|
| **MCC: top** | TP × TN − FP × FN = 34,578,241 − 2,797,485 | 31,780,756 |
| **MCC: bottom** | √( 1,228 × 5,816 × 51,172 × 55,760 ) (the four row and column totals) | 142,754,222 |
| **MCC** | 31,780,756 ÷ 142,754,222 | 0.22 |
| **kappa: observed** | accuracy = (683 + 50,627) ÷ 56,988 | 0.900 |
| **kappa: chance** | (1,228 × 5,816 + 55,760 × 51,172) ÷ $56{,}988^2$ | 0.881 |
| **kappa** | (0.900 − 0.881) ÷ (1 − 0.881) = 0.019 ÷ 0.119 | 0.16 |

**MCC: 1 = perfect, 0 = guessing, −1 = always wrong. Kappa: share of the room above chance that the model uses.**
**At threshold 0.2: MCC 0.33, kappa 0.32.**

# More Than Two Classes: Macro and Micro Averages

IIT Madras Zanzibar

| class | photos | found | recall |
|---|---|---|---|
| **maize** | 80 | 72 | 72 ÷ 80 = 0.90 |
| **cassava** | 15 | 9 | 9 ÷ 15 = 0.60 |
| **banana** | 5 | 1 | 1 ÷ 5 = 0.20 |

**macro average = 0.57**
(0.90 + 0.60 + 0.20) ÷ 3: every class counts equally

**micro average = 0.82**
(72 + 9 + 1) ÷ 100: big classes dominate

**When small classes matter, report the macro average. scikit-learn: average='macro', 'micro' or 'weighted'.**

# Putting a Price on Mistakes: the Cost Matrix

IIT Madras Zanzibar

| cost matrix | predicted FAILED | predicted OK |
|---|---|---|
| actually FAILED | 0 | 10 (missed) |
| actually OK | 1 (wasted visit) | 0 |

| threshold | FN | FP | total cost |
|---|---|---|---|
| **0.5** | 5,133 | 545 | 5,133 × 10 + 545 = 51,875 |
| **0.2** | **2,746** | **6,191** | **2,746 × 10 + 6,191 = 33,651** |

**FN at 0.2 = 5,816 failed – 3,070 found = 2,746.  Saving: (51,875 – 33,651) ÷ 51,875 = 35%.**

**Cheapest threshold wins: 0.2 costs about 35% less, although its accuracy is lower.**
**Weka: CostSensitiveClassifier.  scikit-learn: class_weight, or move the threshold.**

# Three More Curves You Will Meet

IIT Madras Zanzibar

### Precision–recall curve

precision vs recall at every threshold; area = average precision

**best for rare classes**

### Lift / gain chart

visit the top 10% riskiest farms: what share of failures do we reach?

**targeting budgets**

### Calibration curve

farms given 30% risk: did about 30% really fail?  Brier score

**honest probabilities**

**scikit-learn: precision_recall_curve, average_precision_score, CalibrationDisplay, brier_score_loss.**

# Evaluation in Weka: No Code (5-Minute Demo)

IIT Madras Zanzibar

| | | |
|---|---|---|
| 1 | **Run** | Open farm_seasons_5000.csv → Classify → J48 → 10-fold CV → Start |
| 2 | **Read** | Detailed accuracy by class: TP rate, FP rate, precision, recall, F, MCC, ROC area, PRC area |
| 3 | **Curves** | right-click result → Visualize threshold curve → ROC or precision–recall |
| 4 | **Costs** | Choose → meta → CostSensitiveClassifier → J48 + cost matrix → Start |

**Weka prints every measure from today for each class, plus the confusion matrix.**

# Your Turn (3 minutes, pairs)

IIT Madras Zanzibar

**Malaria test on 1,000 people. 50 have malaria.**
**The test flags 80 people; 40 of them really have malaria.**

1. Fill in the confusion matrix: TP, FP, FN, TN
2. Compute accuracy, precision and recall
3. Compute F1
4. Which matters more here, precision or recall? Why?

# Solution 1: Fill the Matrix Step by Step

IIT Madras Zanzibar

| | predicted: POSITIVE | predicted: NEGATIVE |
|---|---|---|
| actually MALARIA | **TP = 40**<br>flagged and sick | **FN = 50 − 40 = 10**<br>sick, not flagged |
| actually HEALTHY | **FP = 80 − 40 = 40**<br>flagged, healthy | **TN = 910**<br>1000 − 40 − 40 − 10 |

**Check: 40 + 40 + 10 + 910 = 1,000 people.**

# Solution 2: The Measures Step by Step

IIT Madras Zanzibar

| measure | working | answer |
|---|---|---|
| **accuracy** | (40 + 910) ÷ 1,000 | 95% |
| **precision** | 40 ÷ 80 | 50% |
| **recall** | 40 ÷ 50 | 80% |
| **F1** | 2 × 0.5 × 0.8 ÷ (0.5 + 0.8) = 0.8 ÷ 1.3 | 0.62 |
| **which matters?** | **a missed case can be fatal; false alarm = one more test** | **recall** |

# Summary

IIT Madras Zanzibar

| | | |
|---|---|---|
| 1 | **Accuracy can lie** | lazy model 89.8% accurate, 0 farmers found |
| 2 | **Confusion matrix** | TP, FP, FN, TN: report it every time |
| 3 | **Precision, recall, F1** | alarms right / failures found / both: 56%, 12%, 0.19 |
| 4 | **Threshold and AUC** | choose by cost (0.2: recall 53%); AUC tree 0.83, NB 0.80 |

**Next: cross-validation, bias and variance: making these scores trustworthy.**

# Questions

IIT Madras Zanzibar

**Q: Why 'confusion' matrix?**

A: It shows exactly where the model confuses one class with the other.

**Q: Is high recall always good?**

A: No: flag everybody → 100% recall, useless precision.

**Q: Why does AUC start at 0.5?**

A: A coin toss already ranks a random pair right half the time.

**Q: What is specificity?**

A: TN ÷ (TN + FP): leaving OK farms alone. Ours at 0.5: 98.9%.

# Further Reading: Evaluating Classifiers

IIT Madras Zanzibar

| resource | why read it | access |
|---|---|---|
| Han, Pei & Tong (2022); Tan, Steinbach, Karpatne & Kumar (2019) | evaluation chapters of the course texts | library |
| Japkowicz & Shah, Evaluating Learning Algorithms, Cambridge University Press (2011) | a whole book on evaluation | library |
| Fawcett (2006), Pattern Recognition Letters 27(8): 861–874 | introduction to ROC analysis | journal |
| Davis & Goadrich (2006), ICML | ROC vs precision–recall curves | proceedings |
| Saito & Rehmsmeier (2015), PLoS ONE 10(3): e0118432 | precision–recall for rare classes | free, open access |
| Chicco & Jurman (2020), BMC Genomics 21: 6 | why MCC beats F1 and accuracy | free, open access |
| scikit-learn User Guide: Metrics and scoring | every measure, with code | free: scikit-learn.org |