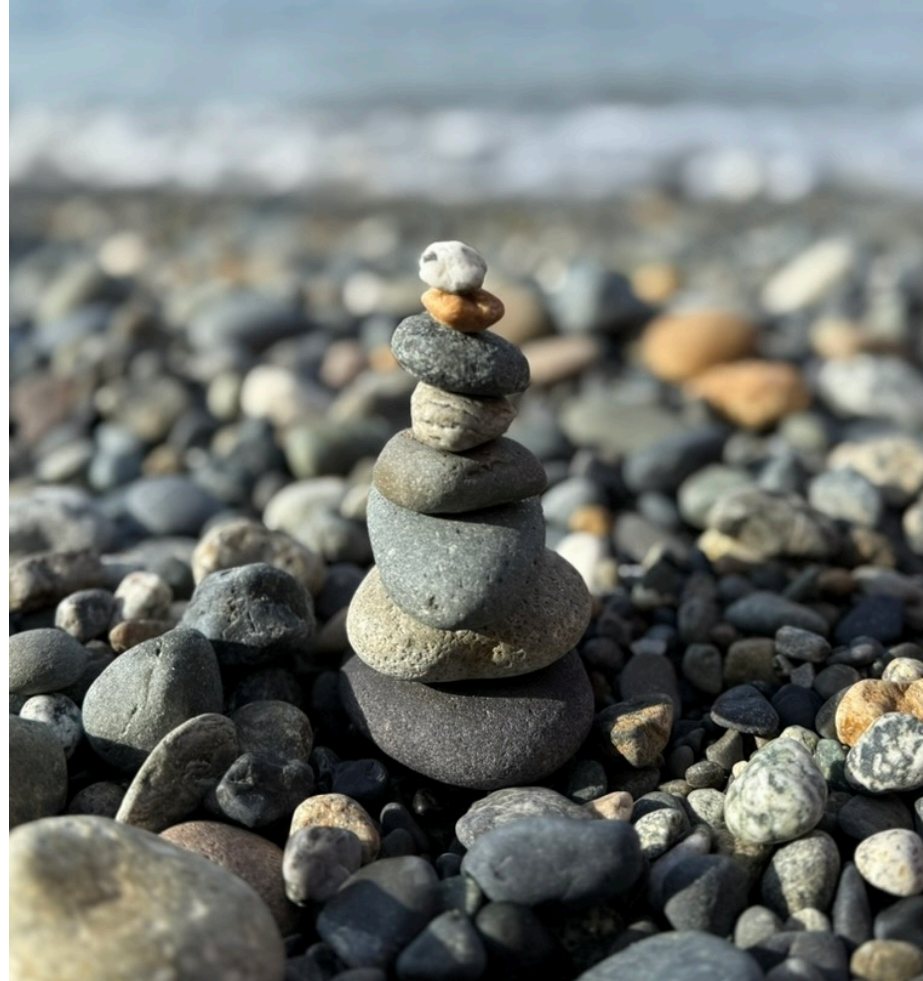


CPSC 330 Lecture 9: Classification Metrics

Varada Kolhatkar

Focus on the breath!



Announcements

- Important information about midterm 1
 - <https://edstem.org/us/courses/104703/discussion/8335467>
 - **Good news for you: You'll have access to our [course notes](#) and sklearn, pandas, and numpy documentation in the midterm!**
- HW3 was due on Monday, Oct 5th 11:59 pm.
- HW4 will be due next week.

Today's big questions

1. When can **accuracy** mislead us, and what does a **confusion matrix** reveal?
2. How do we choose between **precision, recall, and F1** based on the cost of errors?
3. How do **thresholds and ranking metrics** help us choose a decision rule, and how do we evaluate it independently?

A fraud classification example

- Each row is a transaction; **1 = fraud (positive)**, **0 = non-fraud**.
- A positive prediction flags a transaction for investigation.
- Reserve the test set; use stratified splits to preserve class proportions.

	Amount	V1	V2	Class
161019	33.35	1.995966	-0.247681	0
179531	10.00	2.049079	-0.189257	0
229408	18.63	2.030060	0.454393	0
44337	1.98	-1.724699	2.120453	0
143423	150.00	-1.361361	-0.955378	0

Problem: Class imbalance

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}}$$

	Count	Percentage
Class		
0	139313	99.827
1	241	0.173

- Fraud is rare: fewer than 2 transactions in every 1,000.
- Predicting non-fraud every time can achieve **over 99% accuracy**.

DummyClassifier

Predict the accuracy and number of fraud cases caught before revealing the result.

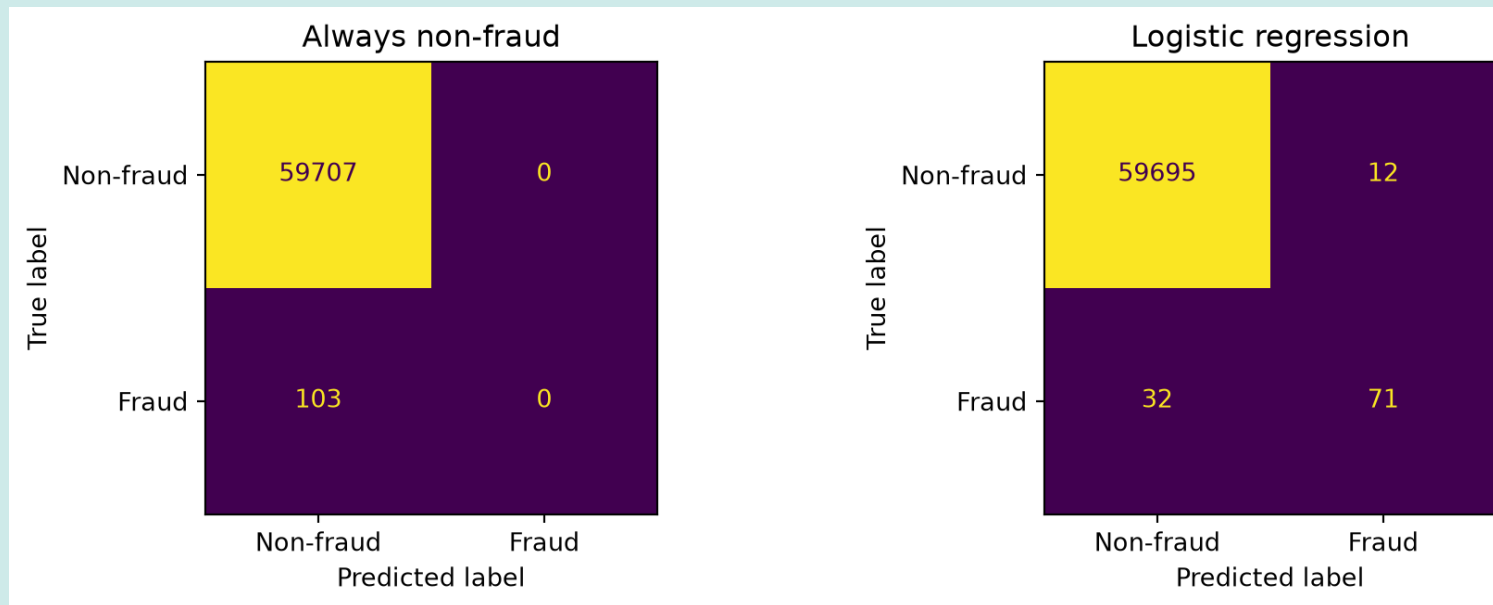
```
1 dummy = DummyClassifier(strategy="most_frequent")
2 dummy.fit(X_train, y_train)
3 print(f"Validation accuracy: {accuracy_score(y_valid, dummy.pre
```

Validation accuracy: 99.828%

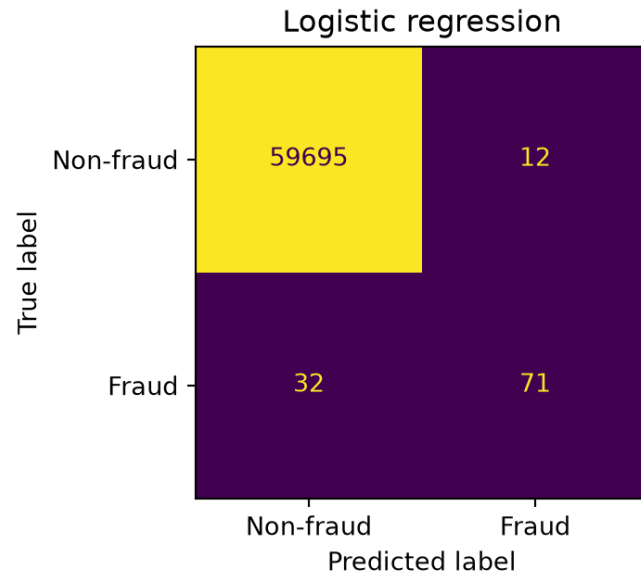
Should the bank deploy this model?

Comparing models on the same validation set

Which model catches more fraud? What new mistakes does it make?



Understanding the confusion matrix



-  True negatives (TN):
Actual Non-fraud, predicted Non-fraud
-  True positives (TP):
Actual Fraud, predicted fraud
-  False positives (FP):
Actual Non-fraud, predicted Fraud
-  False negatives (FN):
Actual Fraud, predicted Non-fraud

Confusion matrix questions

Imagine a spam filter model where emails labeled **1 = spam, 0 = not spam**.

If a spam email is incorrectly classified as not spam, what kind of error is this?

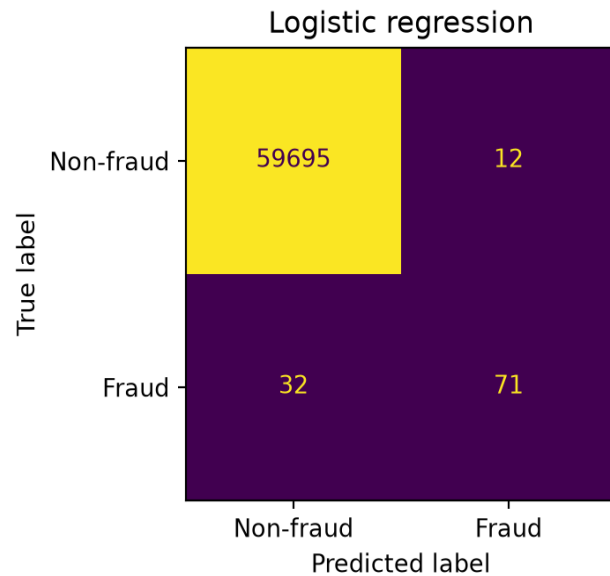
- a. A false positive
- b. A true positive
- c. A false negative
- d. A true negative

Precision and recall

- **Precision:** Of the transactions flagged, what fraction were fraud?
- **Recall:** Of the actual fraud cases, what fraction did we catch?

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

Precision and recall on our validation set



	Metric	Calculation	Value
0	Precision	$71 / (71 + 12)$	0.855
1	Recall	$71 / (71 + 32)$	0.689

High precision means a small **fraction of alerts** are false alarms. High recall means a small **fraction of actual positives** are missed.

Your turn: calculate the metrics

A toy fraud detector has **TP=6, FP=4, FN=2, TN=88**.

- What are its precision and recall?

Trade-off between precision and recall

- Increasing **recall** often decreases **precision**, and vice versa.
- Example:
 - Predict “*fraud*” for every transaction → perfect recall, terrible precision.
 - Predict “*fraud*” only when the predicted probability of fraud is > 0.95 → typically high precision, low recall.

The right balance depends on the application and cost of errors.

F1-score

- Sometimes, we want a **single metric** that balances precision and recall.
- The **F1-score** is the **harmonic mean** of the two:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

- High **F1** means both precision and recall are strong.
- F1 ignores TN and does **not** encode financial costs or review capacity.

For the toy fraud detector: $F1 = 12/(12 + 4 + 2) \approx 0.67$.

Summary

Metric	What it measures	High value means
Accuracy	Overall correctness	Model gets most predictions right
Precision	Quality of positive predictions	Small fraction of alerts are false alarms
Recall	Quantity of true positives caught	Small fraction of positives are missed
F1-score	Balance of precision & recall	Both precision and recall are high

iClicker Exercise 9.1

Select all of the following statements which are TRUE.

- a. In screening where a missed disease delays urgent treatment and a false alarm triggers a low-risk follow-up, prioritize precision over recall.
- b. If a spam filter deletes flagged mail, losing a crucial legitimate email can be more costly than letting a spam email reach the inbox (positive = spam).
- c. If method A gets a higher accuracy than method B, that means its precision is also higher.
- d. If method A gets a higher accuracy than method B, that means its recall is also higher.

Counterexamples

Both models see the same 95 negative and 5 positive examples.

Model	Actual class	Predicted negative	Predicted positive
A	Negative	90 (TN)	5 (FP)
A	Positive	5 (FN)	0 (TP)
B	Negative	80 (TN)	15 (FP)
B	Positive	0 (FN)	5 (TP)

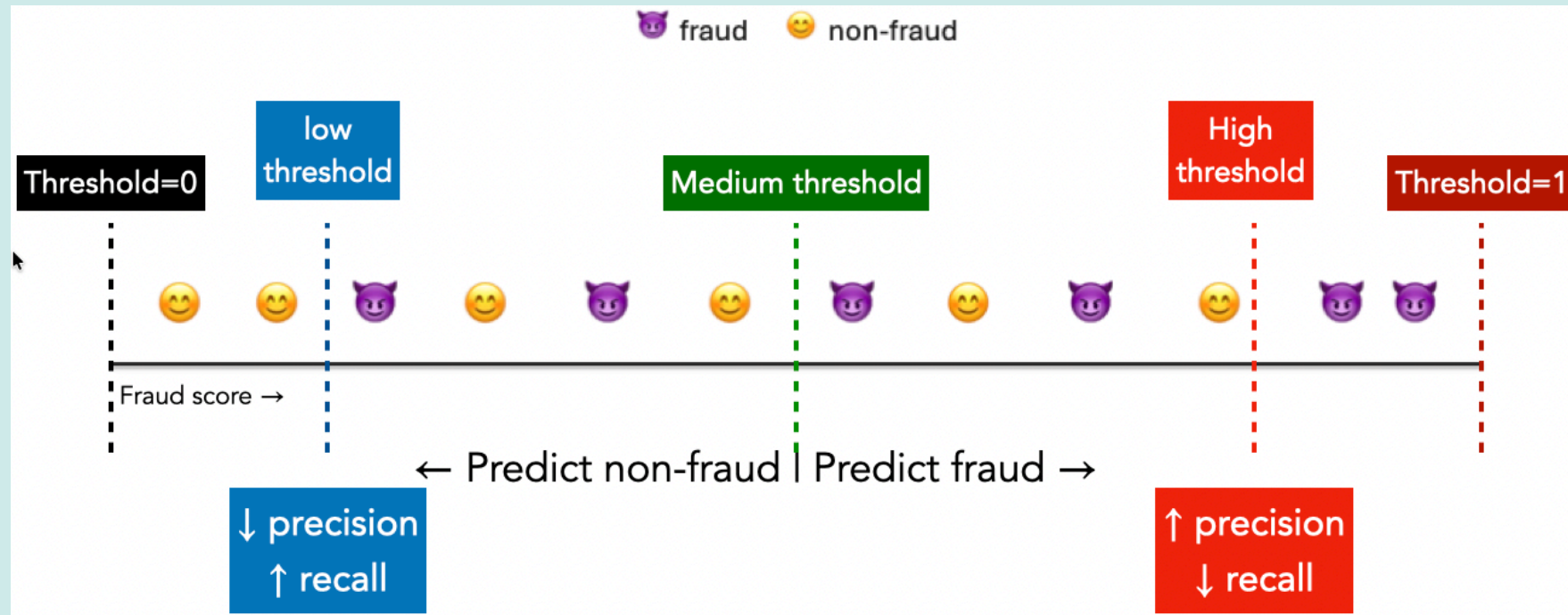
Model	Accuracy	Precision	Recall
A	0.90	0	0
B	0.85	0.25	1

Higher accuracy does not guarantee higher precision or recall.

From probabilities to predictions

- `predict(X)` returns a class label: fraud or non-fraud.
- `predict_proba(X)` returns estimated probabilities for each class.
- We can choose a threshold: predict fraud when its estimated probability exceeds that value.
- Changing the threshold changes the predictions without retraining the model.

Predicting with logistic regression



Move from the medium to the low threshold. Which transactions become TP or FP?

Precision arrows show a tendency, not a guarantee. Scores stay fixed.

Same model, different thresholds

	Threshold	Caught (TP)	Missed (FN)	False alarms (FP)	Precision	Recall
0	0.50	71	32	12	0.855	0.689
1	0.10	82	21	26	0.759	0.796
2	0.01	88	15	113	0.438	0.854

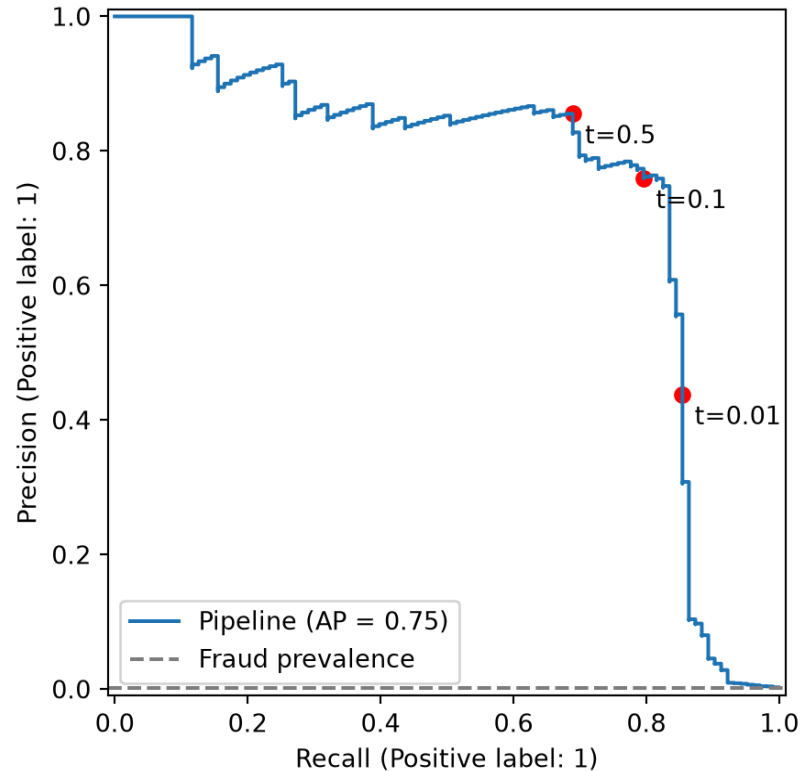
Lowering the threshold cannot decrease recall on fixed data.
Precision can increase or decrease.

Your turn: can both metrics improve?

Start with **TP=6, FP=4, FN=2, TN=88**. Lowering the threshold flags three more transactions: **two fraud, one legitimate**.

1. Calculate the new precision and recall.
2. Does this contradict the usual trade-off?

PR curve



- Horizontal axis: **recall**; vertical axis: **precision**.
- Lower thresholds move toward greater recall; precision can move either way.
- Desirable region: **top right**.

Your turn: choose an operating point

The bank needs **recall of at least 0.8** and wants the most reliable alerts.

	Threshold	Caught (TP)	Missed (FN)	False alarms (FP)	Precision	Recall	F1
0	0.50	71	32	12	0.855	0.689	0.763
1	0.10	82	21	26	0.759	0.796	0.777
2	0.01	88	15	113	0.438	0.854	0.579

Which of these thresholds qualifies? Which qualifying threshold has highest precision?

Average Precision (AP) score example

Rank	Predicted P(fraud)	Actual label	Precision so far	Recall so far
1	0.95	Positive	$1/1 = 1.00$	$1/3$
2	0.85	Negative	$1/2 = 0.50$	$1/3$
3	0.80	Positive	$2/3 = 0.67$	$2/3$
4	0.65	Positive	$3/4 = 0.75$	$3/3$
5	0.40	Negative	$3/5 = 0.60$	$3/3$
6	0.20	Negative	$3/6 = 0.50$	$3/3$

Key observation: Recall increases only when we encounter another positive example. AP summarizes precision at the points where recall increases.

iClicker Exercise 9.2

Choose the appropriate evaluation metric:

Scenario 1: Balance precision and recall at one threshold.

Scenario 2: Summarize retrieval of positives across thresholds.

- a. F1 for 1, AP for 2
- b. AP for 1, F1 for 2
- c. AP for both
- d. F1 for both

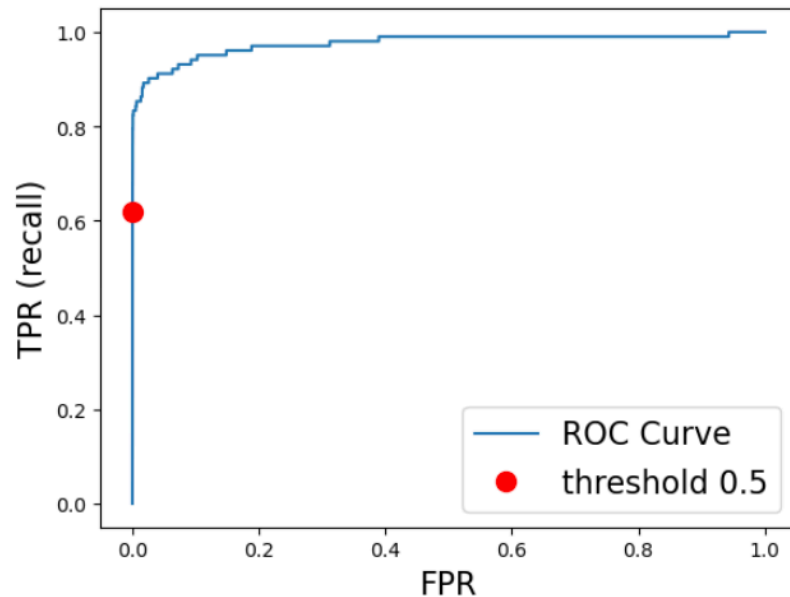
PR vs ROC

- Both curves use **recall (TPR)**: the fraction of actual fraud caught.
- **PR** pairs recall with **precision**: the fraction of flagged transactions that are fraud.
- **ROC** pairs recall with **FPR**: the fraction of legitimate transactions incorrectly flagged.

$$\text{TPR} = \text{Recall} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

ROC plots **TPR versus FPR** as the threshold varies; the desirable region is **top left**.

ROC Curve example



- **Bottom-left** → very high threshold (almost everything predicted negative: low recall, low FPR).
- **Top-right** → very low threshold (almost everything predicted positive: high recall, high FPR).

ROC AUC

The area under the ROC curve (AUC) represents the probability that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative (ties count as half).

For example, $\text{AUC-ROC} = 0.90$ means: If we randomly pick one fraudulent transaction and one legitimate transaction, there's a 90% chance the model assigns a higher fraud score to the fraudulent transaction, with ties counted as half.

Why a small FPR can still mean many false alarms

Suppose 100 of 100,000 transactions are fraud.

At 80% recall and 1% FPR:

- Fraud caught: 80.
- Legitimate transactions flagged: 999.
- Precision: $80 / (80 + 999) \approx 7.4\%$.

PR shows alert reliability; **ROC** shows the fraction of negatives affected. Both are useful with imbalanced data. Inspect the operating region and counts.

Exercise 9.4: which model?

The review team wants to catch at least 80% of positive cases ($\text{recall} \geq 0.8$) while keeping false alarms as low as possible (maximizing precision).

Model	AP	Best validation precision meeting the requirement
A	0.90	0.50
B	0.86	0.70

Questions:

- Which model would you choose for this application?

Compare several metrics with cross-validation

```
1 cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
2 scoring = ["accuracy", "precision", "recall", "f1", "average_precision", "roc_auc"]
3 cv_results = cross_validate(pipe, X_train, y_train, cv=cv, scoring=scoring)
4 pd.Series({metric: cv_results[f"test_{metric}"].mean()
5           for metric in scoring}, name="Mean CV score").round(3)
```

```
accuracy          0.999
precision          0.865
recall            0.610
f1                0.714
average_precision  0.743
roc_auc           0.962
Name: Mean CV score, dtype: float64
```

- Accuracy, precision, recall, F1 use **class predictions**.
- AP and ROC AUC use **scores**.
- Here, **test_** means held-out **CV folds**, not the reserved test set.

Select a pipeline, then a threshold

1. **Training/CV:** compare candidate pipelines with an explicit objective, e.g. `GridSearchCV(..., scoring="average_precision")`.
2. **Validation:** choose the threshold for the operational goal, e.g. maximize precision subject to **recall ≥ 0.8** .
3. **Test:** evaluate the frozen fitted pipeline and threshold once.

The highest CV AP need not yield the best precision at the required recall.

iClicker Exercise 9.5: find the leak

A student selects a pipeline using CV, tries **100 thresholds on the test set**, and reports the highest test F1.

“The test set was never passed to **fit**, so the evaluation is independent.”

Is this correct? How would you repair the workflow?

Final evaluation: freeze both choices

```
1 chosen_threshold = chosen["Threshold"]
2 test_scores = pipe.predict_proba(X_test)[: , 1]
3 pd.DataFrame([threshold_metrics(y_test, test_scores, chosen_threshold)]).round(3)
```

	Threshold	Caught (TP)	Missed (FN)	False alarms (FP)	Precision	Recall	F1
0	0.092	119	29	36	0.768	0.804	0.785

- Report **counts** as well as scores to show the scale of the mistakes.
- Do not adjust the model or threshold after seeing these results.
- Refitting can change scores; the chosen threshold may no longer transfer.

Summary questions

1. Why can **accuracy** be misleading on imbalanced data? What do **FP** and **FN** reveal that accuracy hides?
2. When would you prioritize **precision** or **recall**, and when is **F1** useful?
3. How do **AP** and **ROC AUC** differ from metrics at one threshold? Where should we select the pipeline and threshold before final evaluation?