

99% accuracy. Zero frauds caught.

D8A.ACADEMY/RESOURCES/CLASS-
IMBALANCE

THE DEMO

accuracy 99%	← say "no" to everyone: 99%
positives caught	zero. the model learned the base rate

MEASURE IT RIGHT

recall	← of the real ones, how many caught
precision	of the flagged, how many were real
PR curve	over ROC when positives are rare

FIX IT

class_weight	'balanced': the rare class costs more
resample	oversample rare, undersample common
threshold	move it off 0.5, read the PR curve

WORTH MEMORIZING

01 Weight the rare class, move the threshold

Two lines change. The model pays more for a missed positive, and the cut-off stops being 0.5 by default.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

clf = LogisticRegression(class_weight="balanced",
                        max_iter=1000)
clf.fit(X_train, y_train)
proba = clf.predict_proba(X_test)[: , 1]
pred = (proba >= 0.3).astype(int)
print(classification_report(y_test, pred))
```

→ Report recall and precision for class 1, not accuracy. The 0.3 is a starting point: read it off the precision-recall curve with whoever pays for a missed case.

02 Trap: the metric that cannot fail

Base rate 1% fraud. The model that says no to everyone against a real one.

METRIC	SAY-NO MODEL	REAL MODEL
accuracy	99.0%	98.6%
recall (fraud)	0%	71%
precision (fraud)	nothing flagged	45%

→ The say-no model wins on accuracy and catches nobody. With a 1% base rate, any model has to beat 99% before it has done anything at all.