

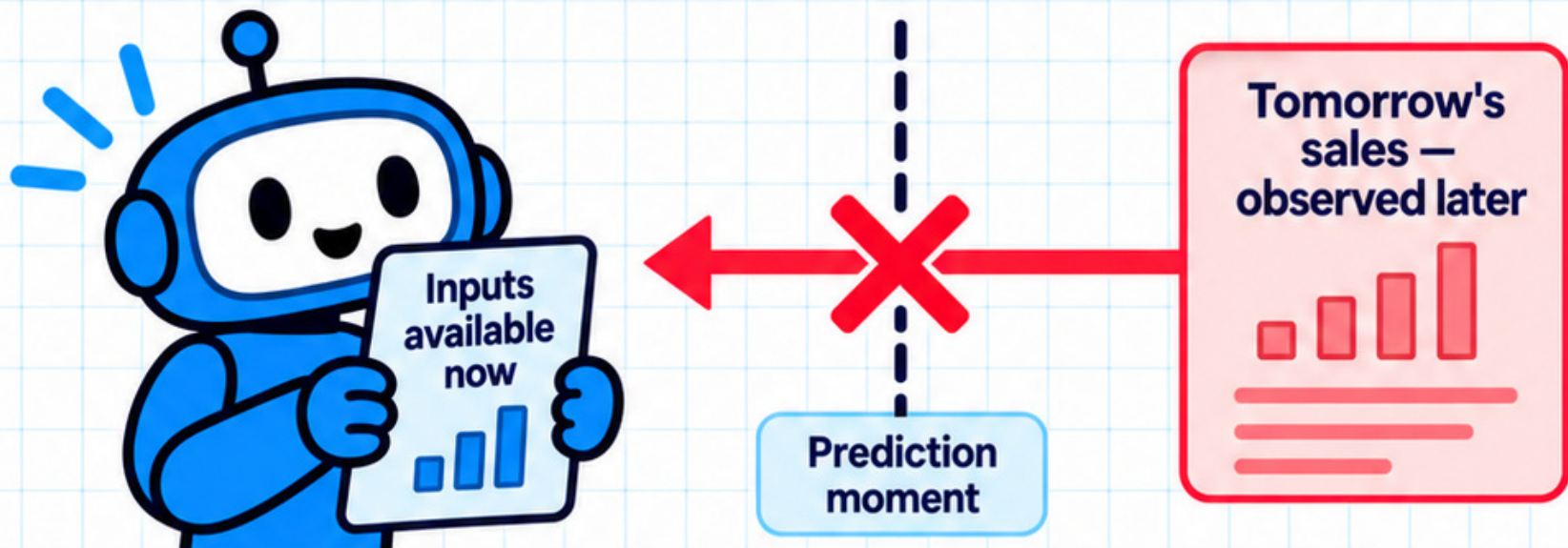
CAN YOUR MODEL SEE THE FUTURE?

Leakage-safe data splits

A model can look impressive when its evaluation contains information that would be unavailable in production.

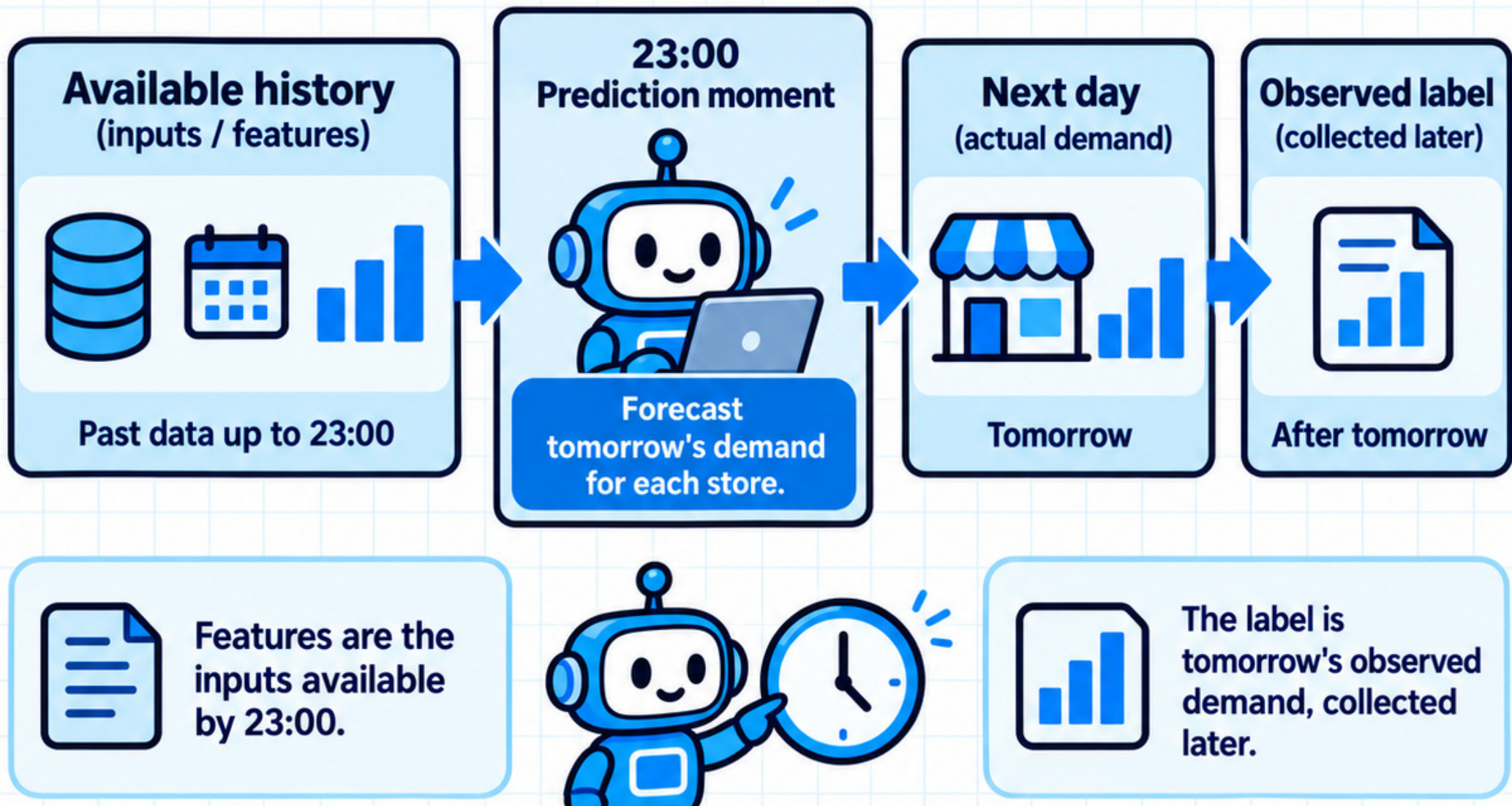
Today: design data splits that test the real prediction job.

Example: forecasting tomorrow's demand.



DEFINE THE PREDICTION MOMENT

At 23:00 each day, forecast tomorrow's demand for each store.

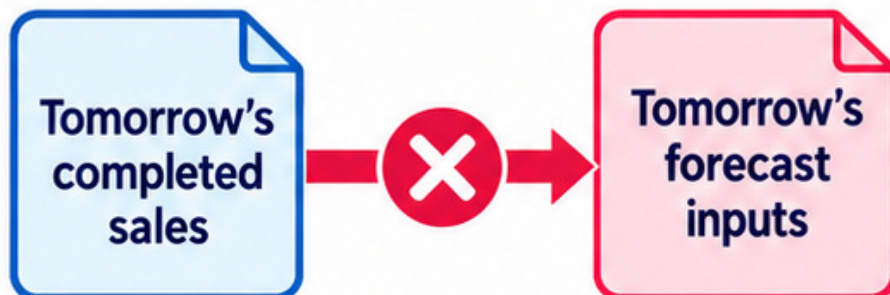


Write this timeline before building the dataset. "Historical data" can still contain facts learned after the prediction moment.

WHAT LEAKAGE MEANS

Leakage happens when information crosses a boundary the production system cannot cross.

Future outcome as a feature

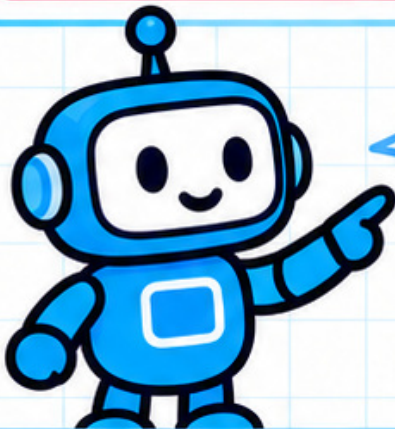


This arrow is **PROHIBITED** because the future outcome is unavailable at prediction.

Evaluation data teaches preprocessing



This final arrow is **PROHIBITED** because evaluation data must not influence preprocessing.



Example: using tomorrow's completed sales as an input to tomorrow's forecast.

A subtler example: filling missing values using statistics calculated from the final test data.



Both can make the measured result too optimistic.

GIVE EACH DATASET ONE JOB

Training set

learn model parameters and preprocessing rules.



Train model

Validation set

compare design choices, such as features and model settings.



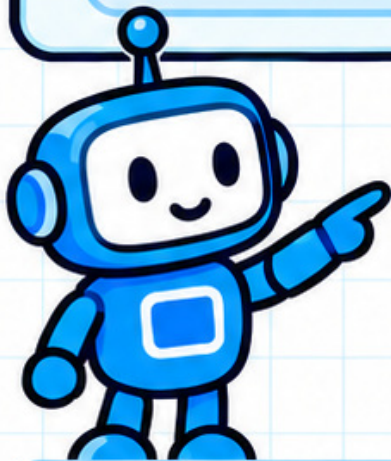
Compare design choices

Test set

estimate the chosen system's performance after those choices are fixed.



Estimate performance



The names alone do not protect these roles. The workflow must preserve the boundaries.

A RANDOM SPLIT MAKES AN ASSUMPTION

A random split mixes rows before assigning them to sets.

RANDOM SPLIT (SHUFFLED)

Shuffled data
Illustrative dates



Training set



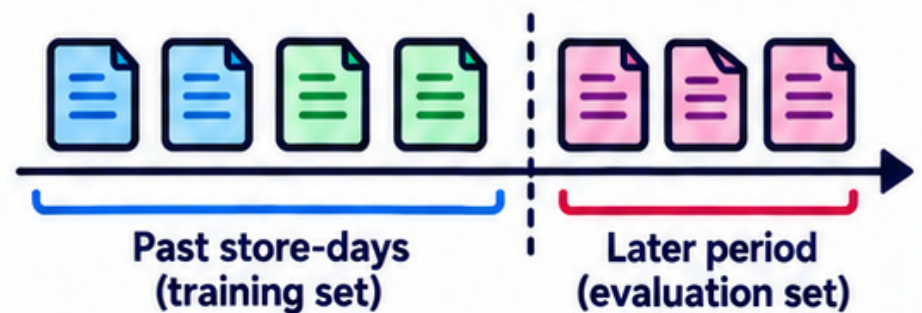
Evaluation set



It can be suitable when examples are sufficiently independent and represent the same prediction setting.

FORECAST TIMELINE (ORDERED)

Chronological data (store-days)



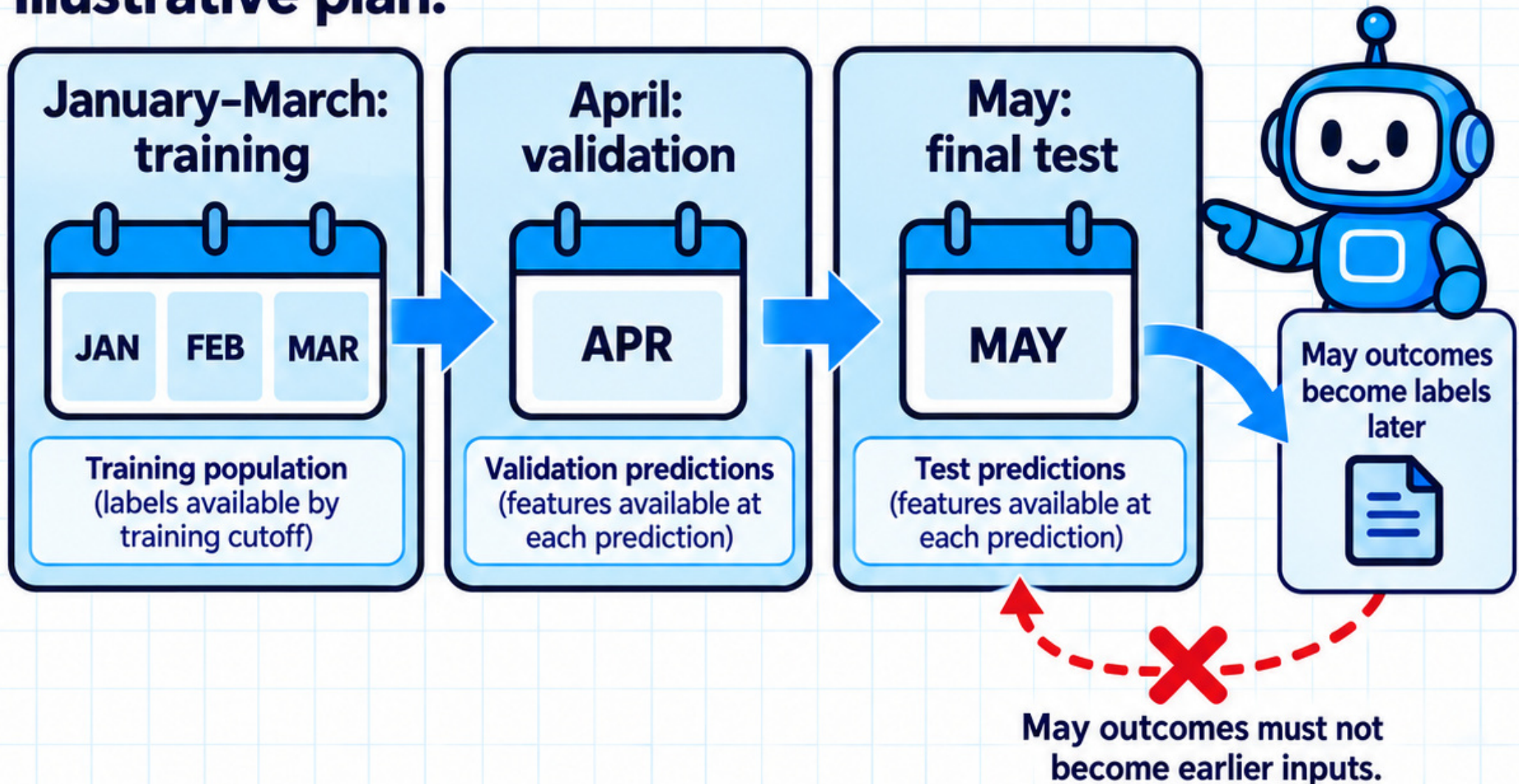
Our forecast predicts a later period. Mixing future store-days into training does not reproduce that job.



Choose the split from the deployment question.

USE TIME TO TEST FUTURE PREDICTIONS

Illustrative plan:

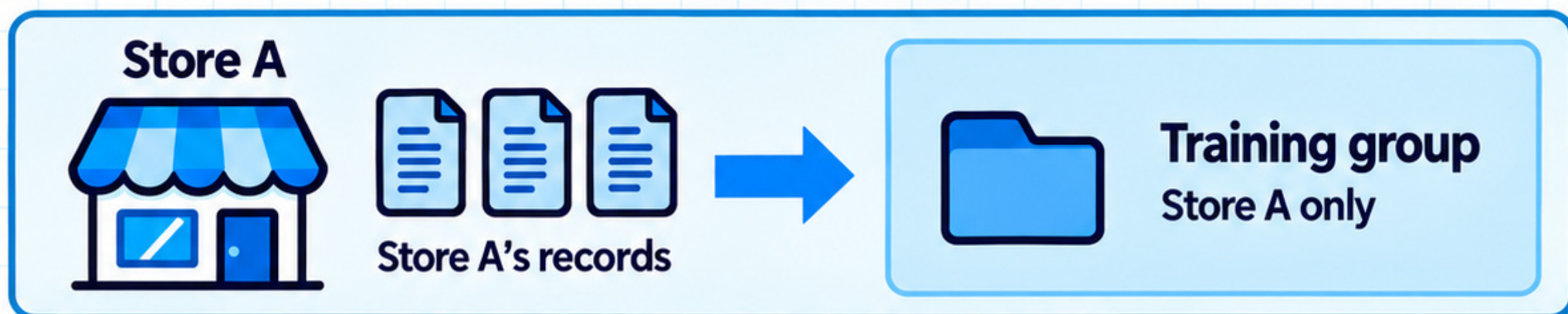


At each simulated training cutoff, include only examples whose labels were already available.

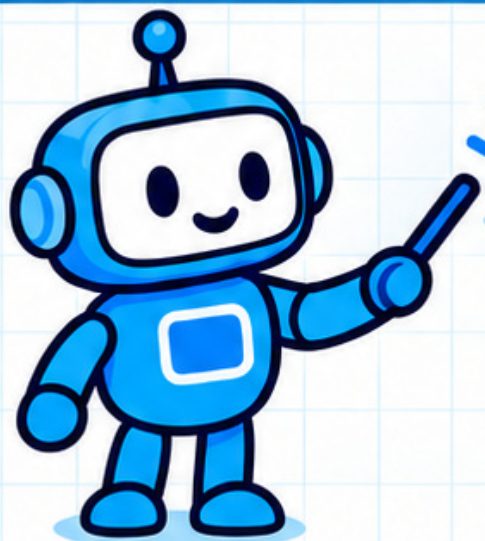
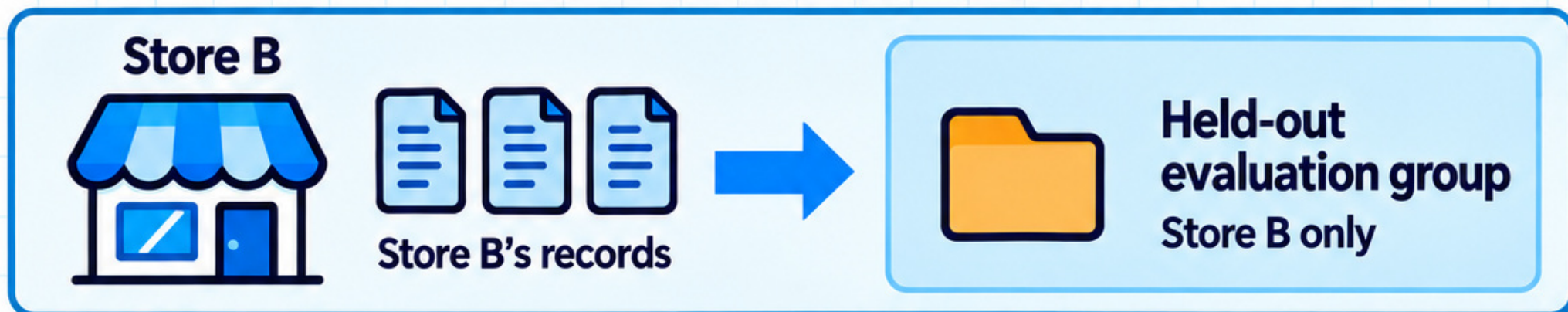
May features must use the history available at each May prediction. May outcomes become labels later; they must not become earlier inputs.

USE GROUPS TO TEST NEW ENTITIES

An entity is the thing being predicted: a customer, store, driver, or device.



Illustrative split: hold out Store B



If the goal is performance on unseen stores, keep every row from a held-out store out of training. Otherwise, repeated records can let the model recognize stores it was supposed to encounter for the first time.

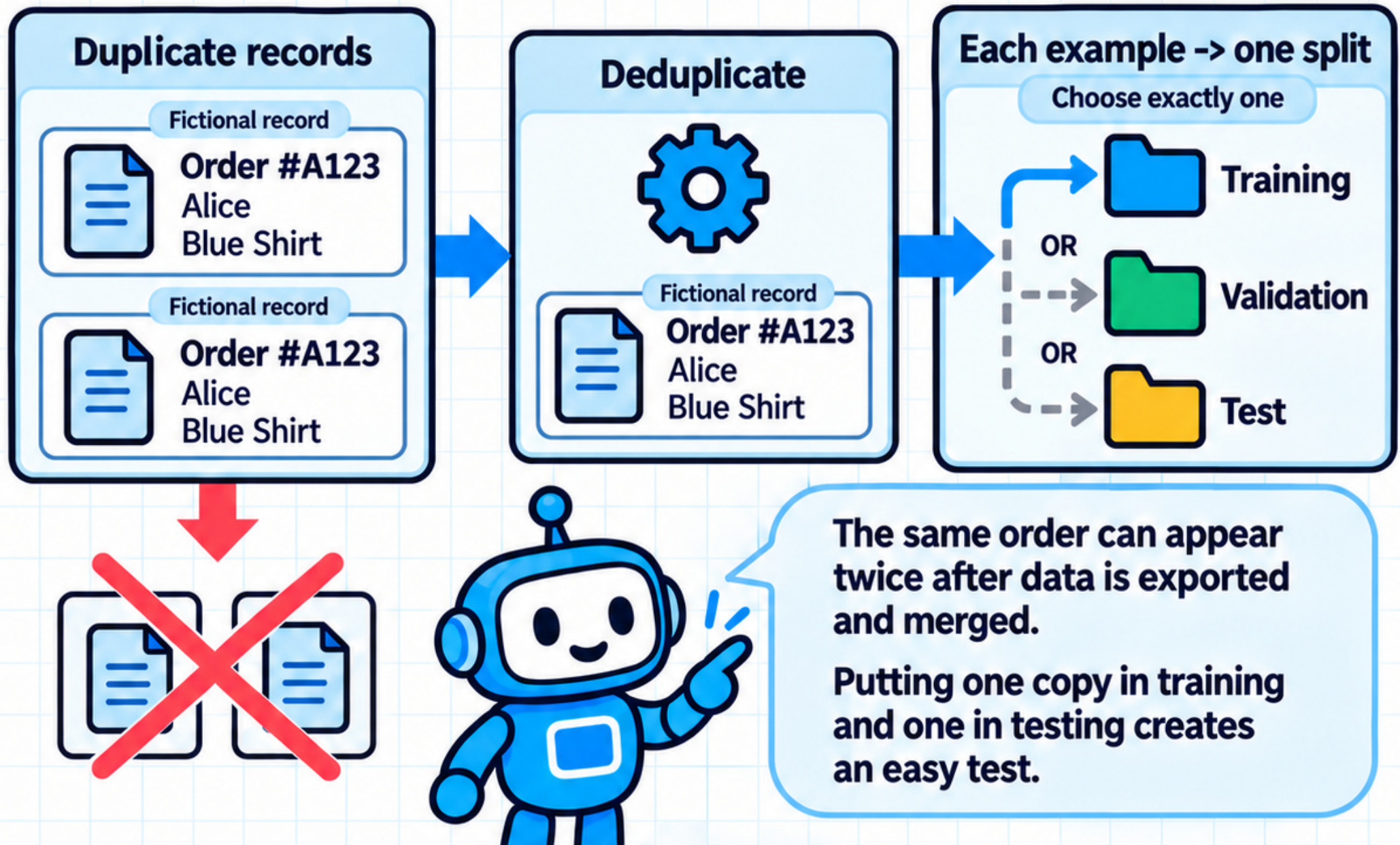
TIME AND GROUPS ANSWER DIFFERENT QUESTIONS

	 Existing stores (known store identities)	 Unseen stores (new store identities)
Current period (earlier time period) 	 Reference population	 Demand at unseen stores? Hold out store identities.
Later period (later time period) 	 Later demand at existing stores? Test a later time period.	 Later demand at unseen stores? Enforce both conditions.



One split does not answer every question.
Report each evaluation population so readers know what the score actually measures.

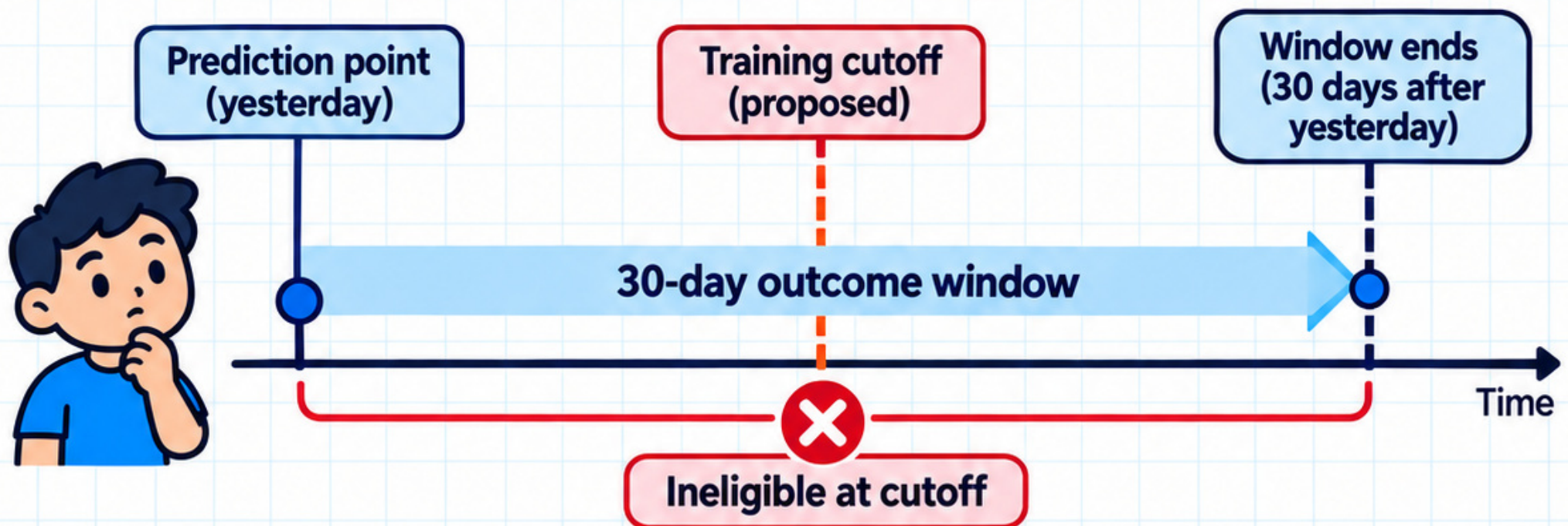
KEEP RELATED COPIES TOGETHER



Deduplicate by a defined identity rule. Keep derived copies together when they represent the same example. Do this before assigning split membership.

LABELS CAN SPAN A TIME WINDOW

Our inactivity example: predict no completed purchase in the next 30 days.



A label for a prediction made yesterday is not complete today.



At a training cutoff, exclude examples whose outcome window has not finished or whose labels have not arrived.



Also inspect overlapping outcome windows across evaluation boundaries.

A GAP NEEDS A REASON



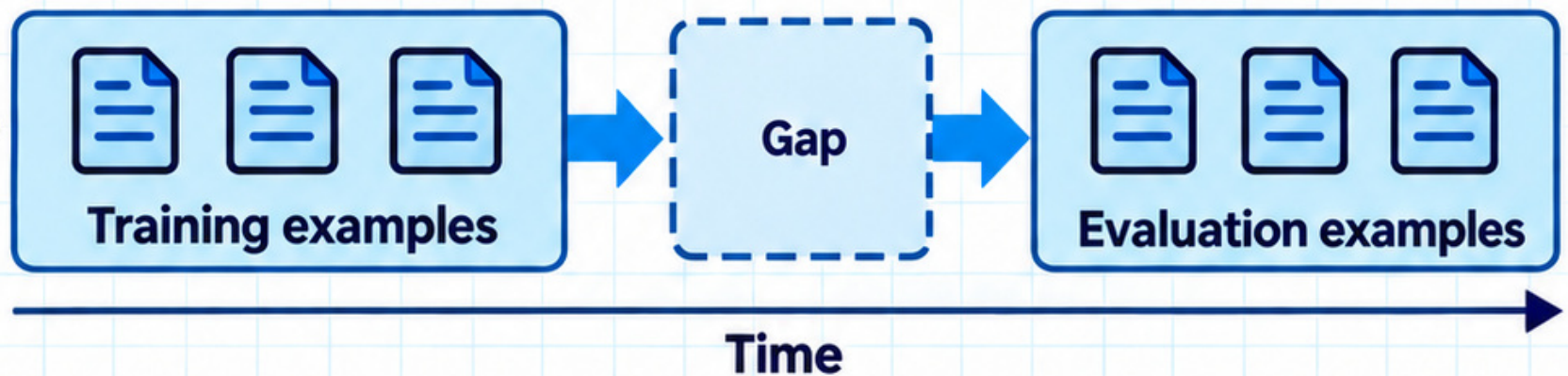
A gap separates the end of training examples from the start of evaluation examples.



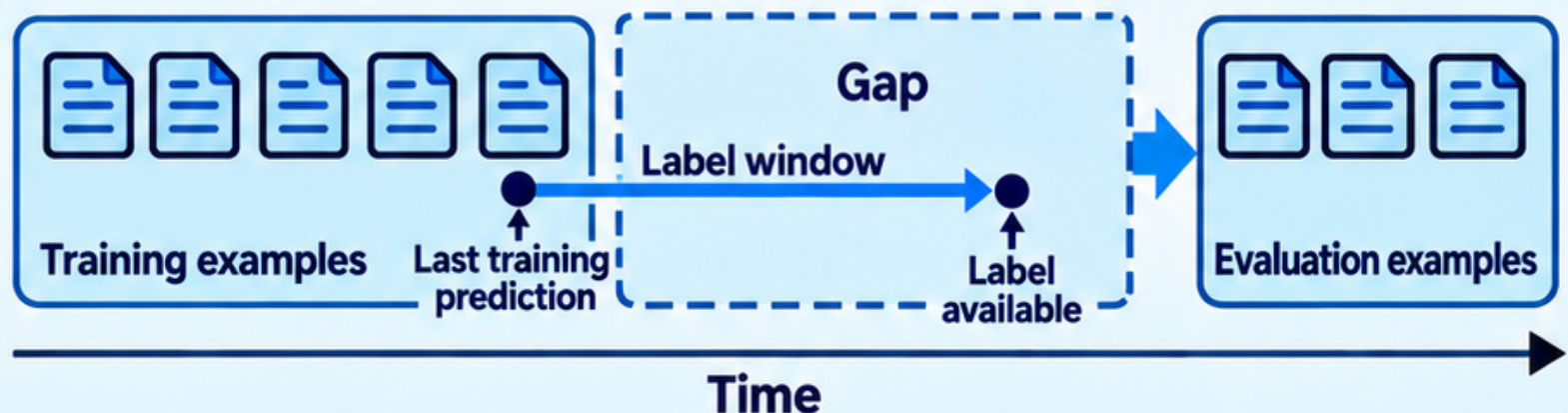
It can help keep unfinished label windows or tightly related observations apart.



Choose its length from the label horizon, data delays, and task dependence. A fixed seven-day gap is not a universal leakage fix.



A longer label window can need a larger gap:

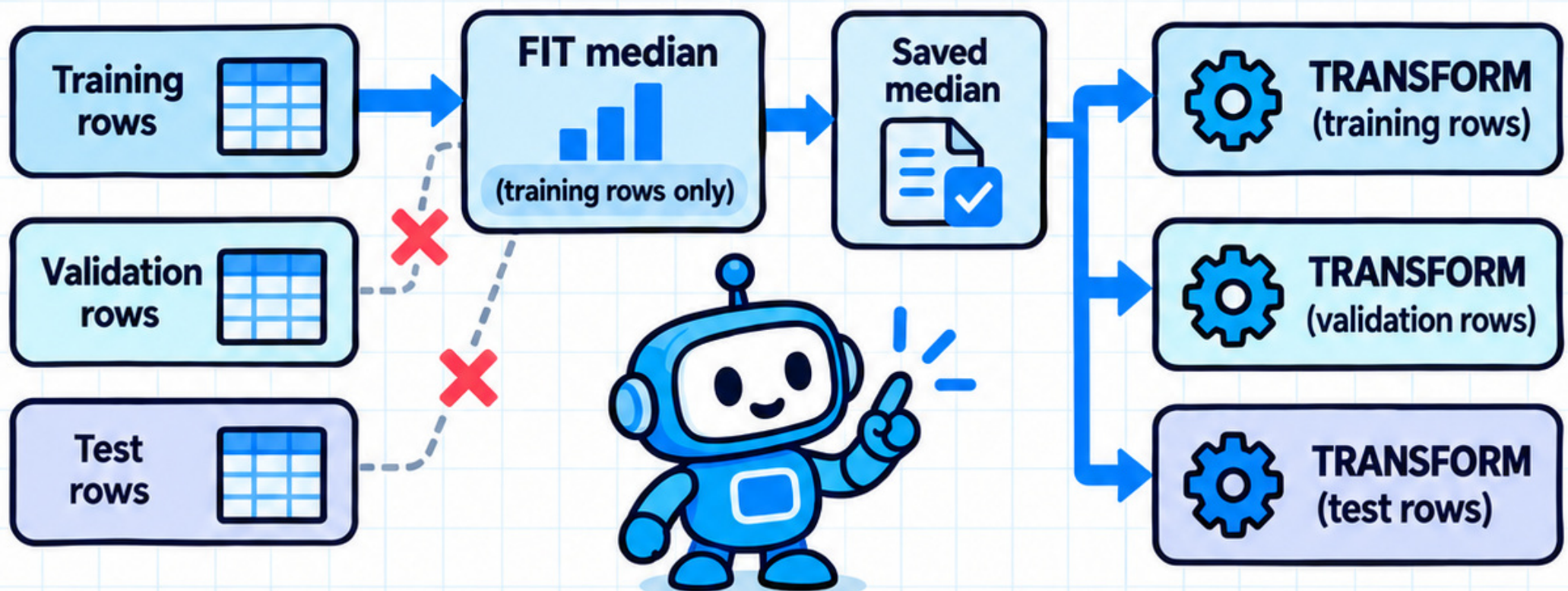


FIT PREPROCESSING ON TRAINING ONLY

Suppose missing values in a past-sales feature are replaced with the training-set median.

Learn that median from training rows only.

Apply the same saved value to validation and test rows.



The same rule applies to learned scaling, category vocabularies, and feature selection. Do not let evaluation data teach these steps.

A PIPELINE PRESERVES THE ORDER

A scikit-learn Pipeline can connect preprocessing and the model. For each validation fold, fit the whole pipeline using that fold's training rows. Then transform and predict on its held-out rows.

Toy feature data

Row	Feature A	Feature B	Feature C
1	10	cat	0
2	?	dog	1
3	7	cat	0
4	12	dog	1
5	?	cat	0
6	3	dog	1

Training rows
(this fold)

Held-out rows
(this fold)

Fit the whole pipeline on training rows
(for this validation fold)

Imputer



fit on
training rows

Encoder



fit on
training rows

Model



fit on
training rows



One pipeline.
Same order.

Use the fitted pipeline on held-out rows

Imputer



transform
held-out rows

Encoder



transform
held-out rows

Model



predict
on held-out rows

Predictions
(for this fold)



For each validation fold,
fit the whole pipeline using
that fold's training rows.



The pipeline helps prevent preprocessing leakage. It cannot repair a future-derived input column or an unsuitable split.

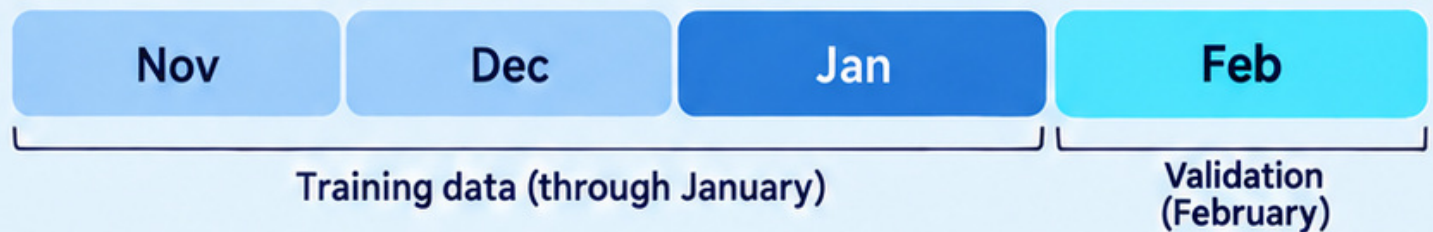
REHEARSE MORE THAN ONE FUTURE PERIOD

Rolling evaluation repeats the past-to-future test.

Illustrative dates



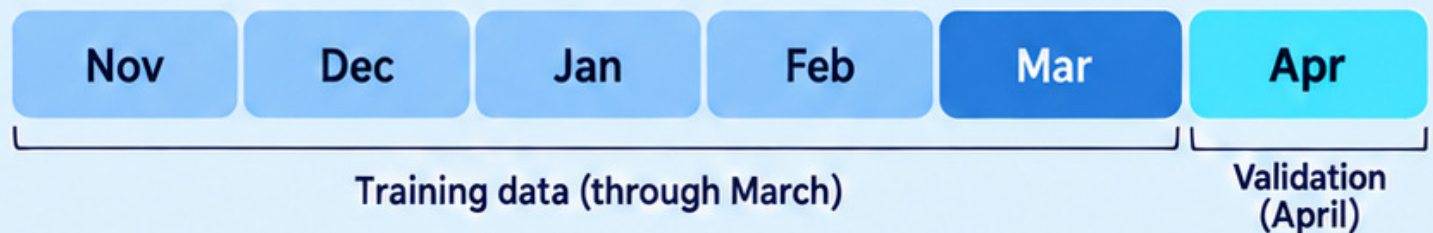
Train through January → validate February.



Train through February → validate March.



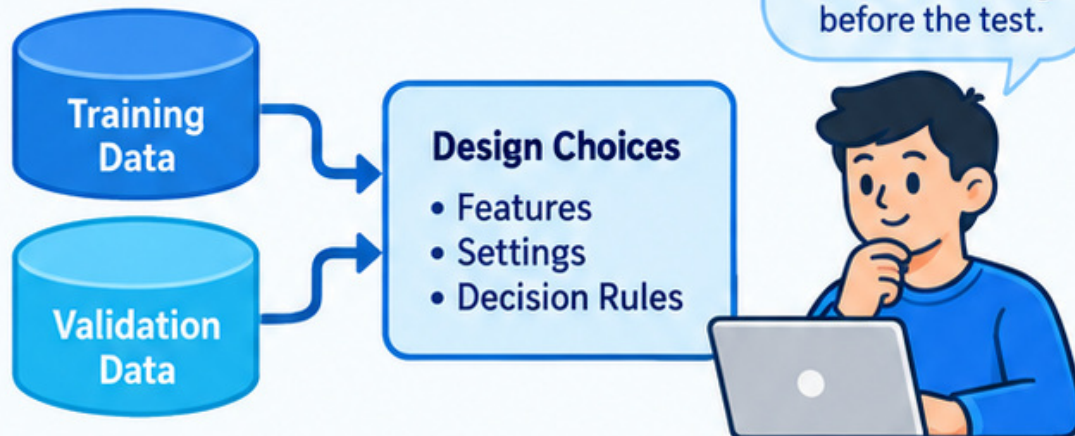
Train through March → validate April.



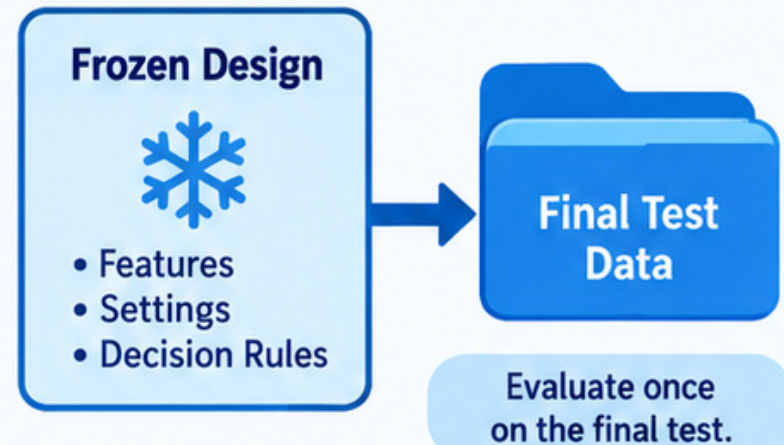
Rebuild each training run using only information available at its cutoff.
Compare results across periods; an average can hide a difficult month.

PROTECT THE FINAL TEST

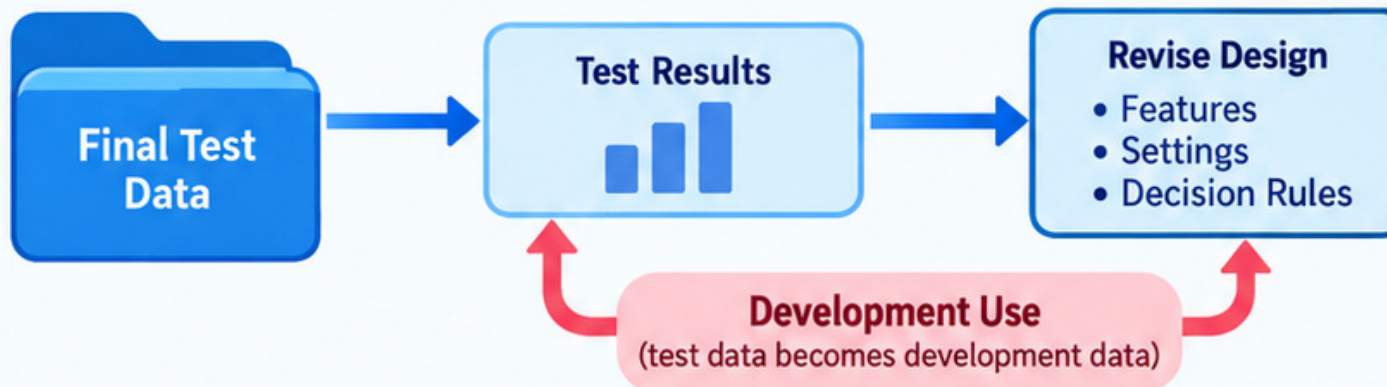
- 1** Choose features, settings, and decision rules with training and validation data.



- 2** Then evaluate the frozen design on the final test.



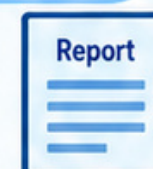
- 3** Repeatedly changing the design after reading test results turns that set into development data.



If you change the design after seeing test results...



- 4** If that happens, report it and obtain fresh evaluation evidence for the revised design.

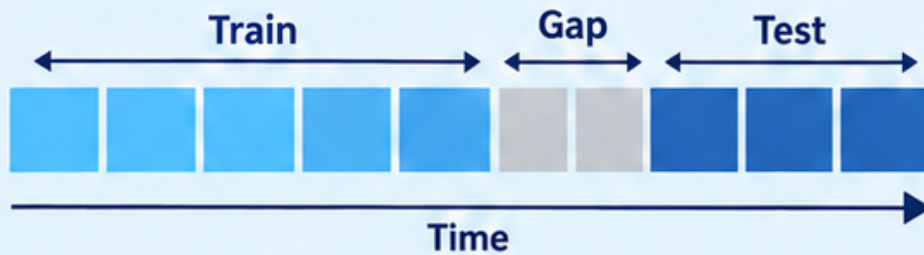


Obtain fresh evaluation evidence for the revised design.

TOOLS SUPPORT THE SPLIT POLICY

TimeSeriesSplit

TimeSeriesSplit: generate ordered folds; configure any gap deliberately.



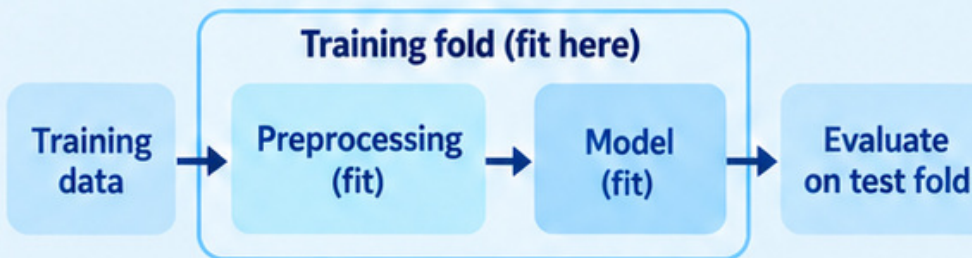
GroupKFold

GroupKFold: keep an entity group out of its training fold.



Pipeline

Pipeline: fit preprocessing inside each training fold.



MLflow

MLflow: record dataset references and run results.



Save row IDs, cutoffs, and split rules too.
Tool names do not prove the data is safe.



Split rule metadata (save with the run)

	Row IDs	_____
	Cutoffs	_____
	Split rules	_____

FIND THE HIDDEN FUTURE FACT

At 09:00, an agent requests today's demand forecast.

Input A



yesterday's demand,
available at 02:00.

Input B



today's final sales,
available at 00:00
tomorrow.

Input C



today's promotion
schedule,
published yesterday.



09:00
Forecast Service



Which inputs could the forecast service use at 09:00?

Explain the time boundary, not just the column names.

A:

B:

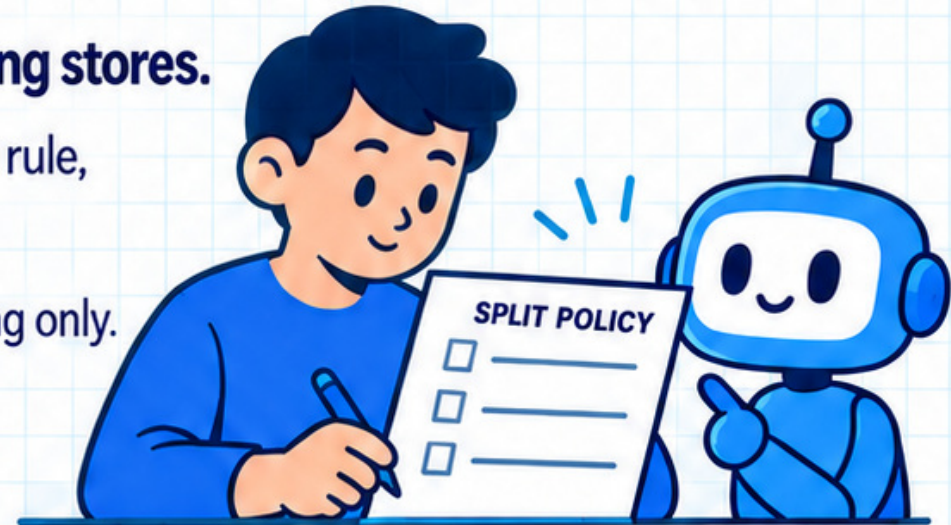
C:

DESIGN YOUR EVALUATION BOUNDARY

Write a split plan for next-day demand at existing stores.

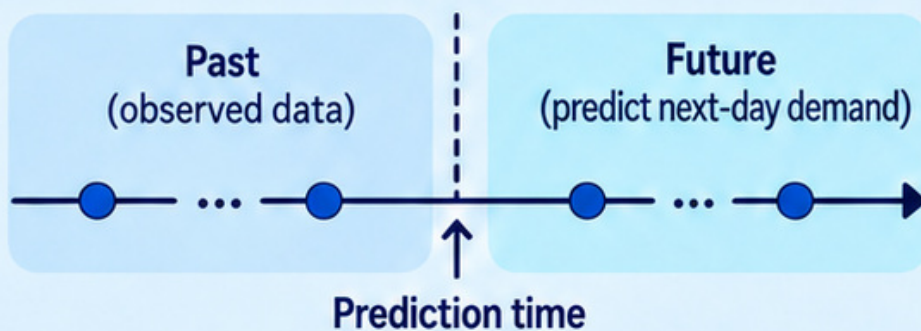
State the prediction time, label window, data-availability rule, validation periods, and final test period.

Name one preprocessing step that must learn from training only.



PAST TO FUTURE SETUP

Use only information available at prediction time.



DAY 10: Reproducible features



Save this before trusting a model score.

SPLIT POLICY (FILL IN)

Prediction time:

Label window:

Data-availability rule:

Validation periods:

Final test period:

Preprocessing step (training only):