WHAT EXACTLY ARE WE PREDICTING?

"Predict churn" is incomplete.

We need a customer population, a prediction moment, an outcome and a time window.

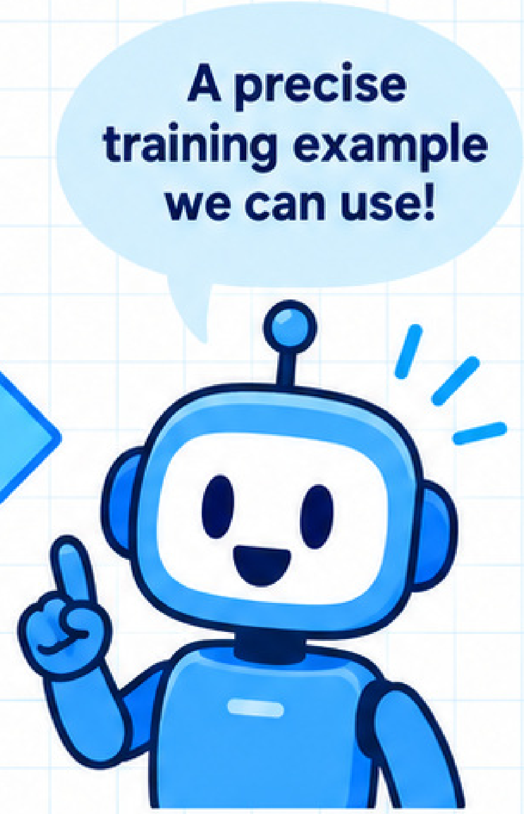
Today: turn that vague request into a precise training example.

1. CUSTOMER POPULATION
Which customers are we making predictions for?

2. PREDICTION MOMENT
When do we make the prediction?

3. OUTCOME
What does "churn" mean in this context?

4. TIME WINDOW
Over what future period do we measure the outcome?



EXAMPLE TIMELINE (ALL TIMES UTC)

OBSERVATION INTERVAL
[2 July 2026 00:00, 1 August 2026 00:00)

2 July 2026 00:00 — 1 August 2026 00:00

LABEL INTERVAL
[1 August 2026 00:00, 31 August 2026 00:00)

1 August 2026 00:00 — 31 August 2026 00:00

A LABEL IS AN OBSERVED OUTCOME

A label is the outcome a supervised model learns to predict.

Our toy label:

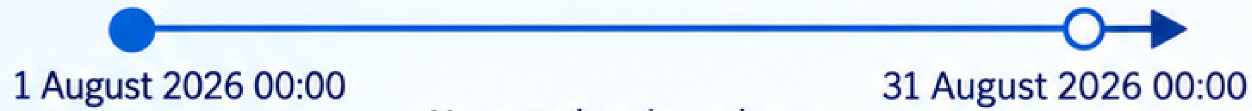
1 = no completed purchase during the next 30 days.

0 = at least one completed purchase during those 30 days.

This is purchase inactivity. It does not mean account cancellation or permanent customer loss.

Example 1

Label interval
[1 August 2026 00:00,
31 August 2026 00:00)



No completed purchases
in this 30 day window.

Label = 1

Example 2

Label interval
[1 August 2026 00:00,
31 August 2026 00:00)

At least one completed purchase
in this 30 day window.

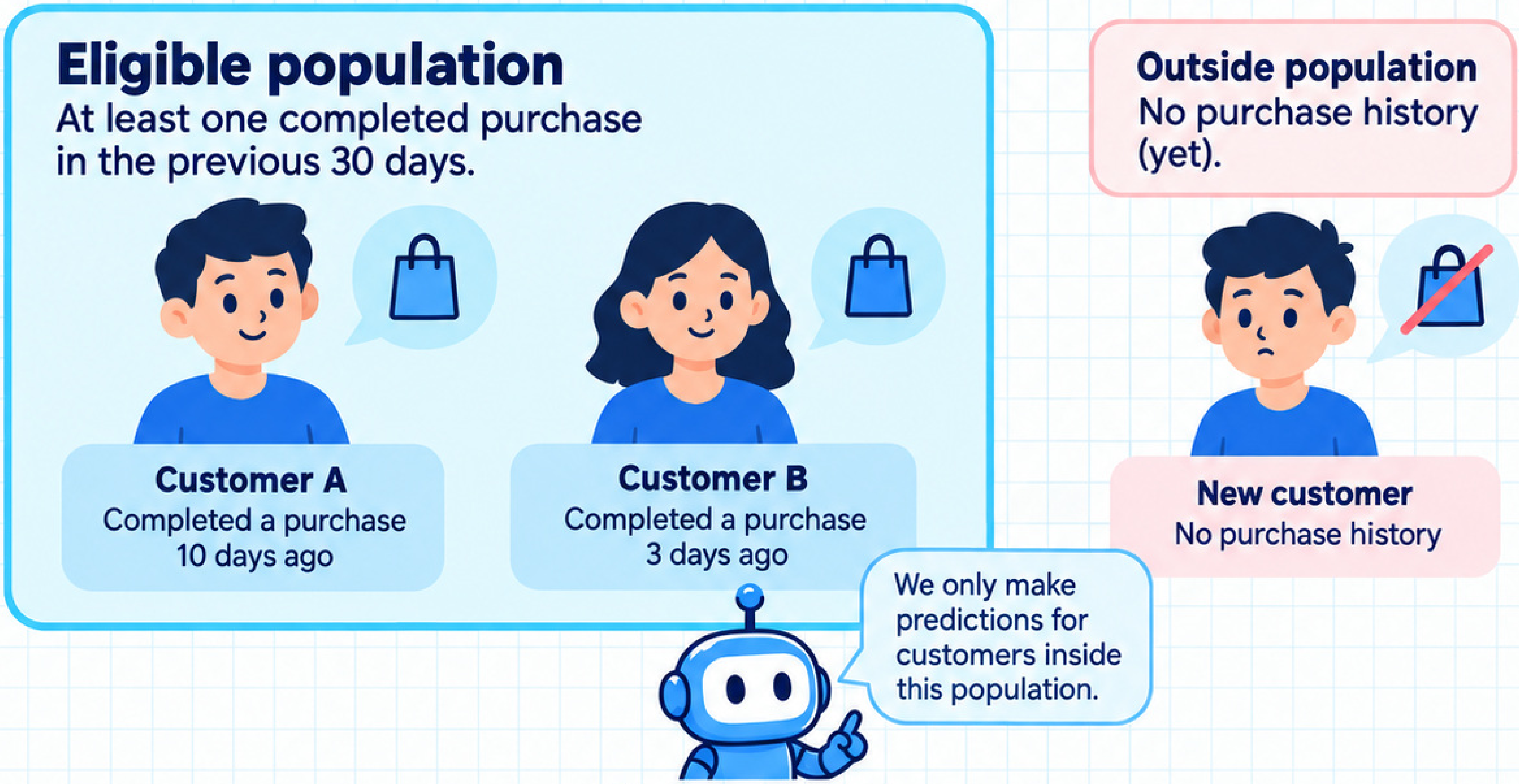
Label = 0

DECIDE WHO RECEIVES A PREDICTION

Our eligible population: customers with at least one completed purchase in the previous 30 days.

One training row represents one customer at one prediction moment.

A new customer with no purchase history is outside this toy population. Supporting that customer would require an explicit design choice.



FIX THE PREDICTION MOMENT

Prediction moment: 1 August 2026, 00:00 UTC.

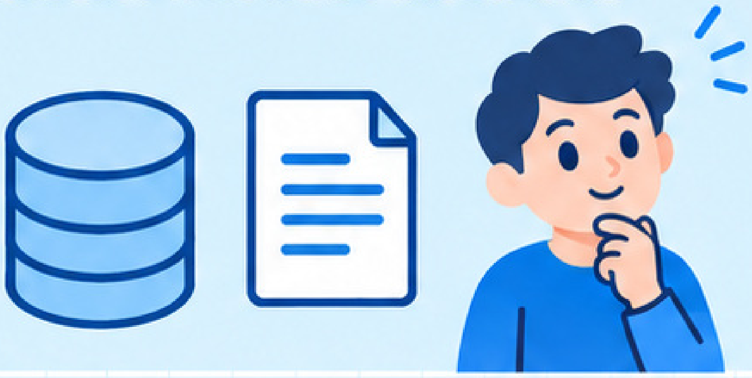
Imagine the model is making its decision at that instant.

The training features must reflect information the serving system could use then. A field discovered later cannot travel backward simply because it now exists in the database.

Prediction moment
1 August 2026, 00:00 UTC

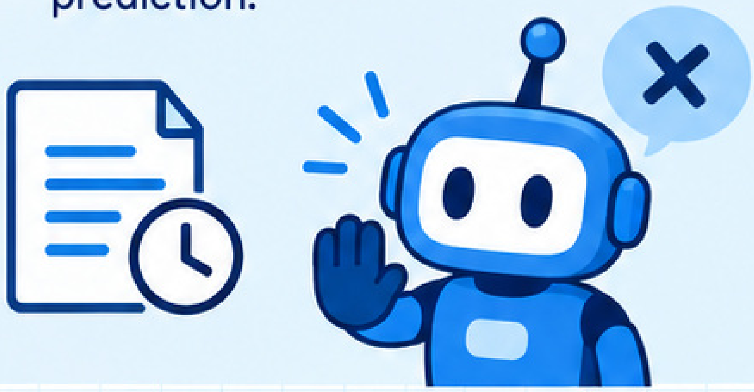
Information available at prediction moment

Can be used as training features because it would be known then.



Information from later events

Cannot be used, because it would not be available at the time of prediction.



Observation interval
[2 July 2026 00:00, 1 August 2026 00:00)

Label interval
[1 August 2026 00:00, 31 August 2026 00:00)

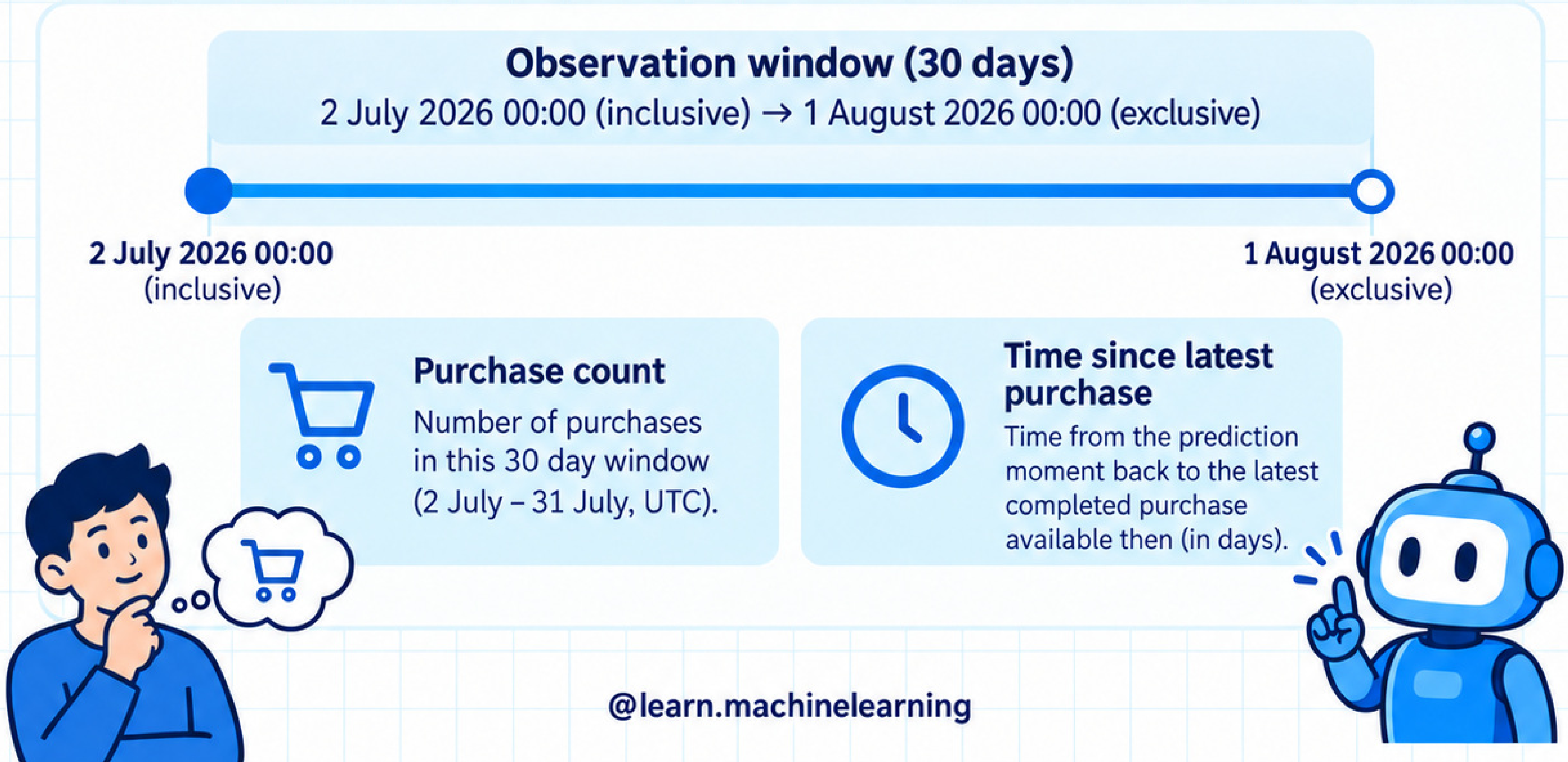
LOOK BACKWARD TO BUILD FEATURES

Observation window: the 30 days before prediction.

For this snapshot: 2 July through 31 July, UTC.

Possible features include purchase count and time since the latest purchase.

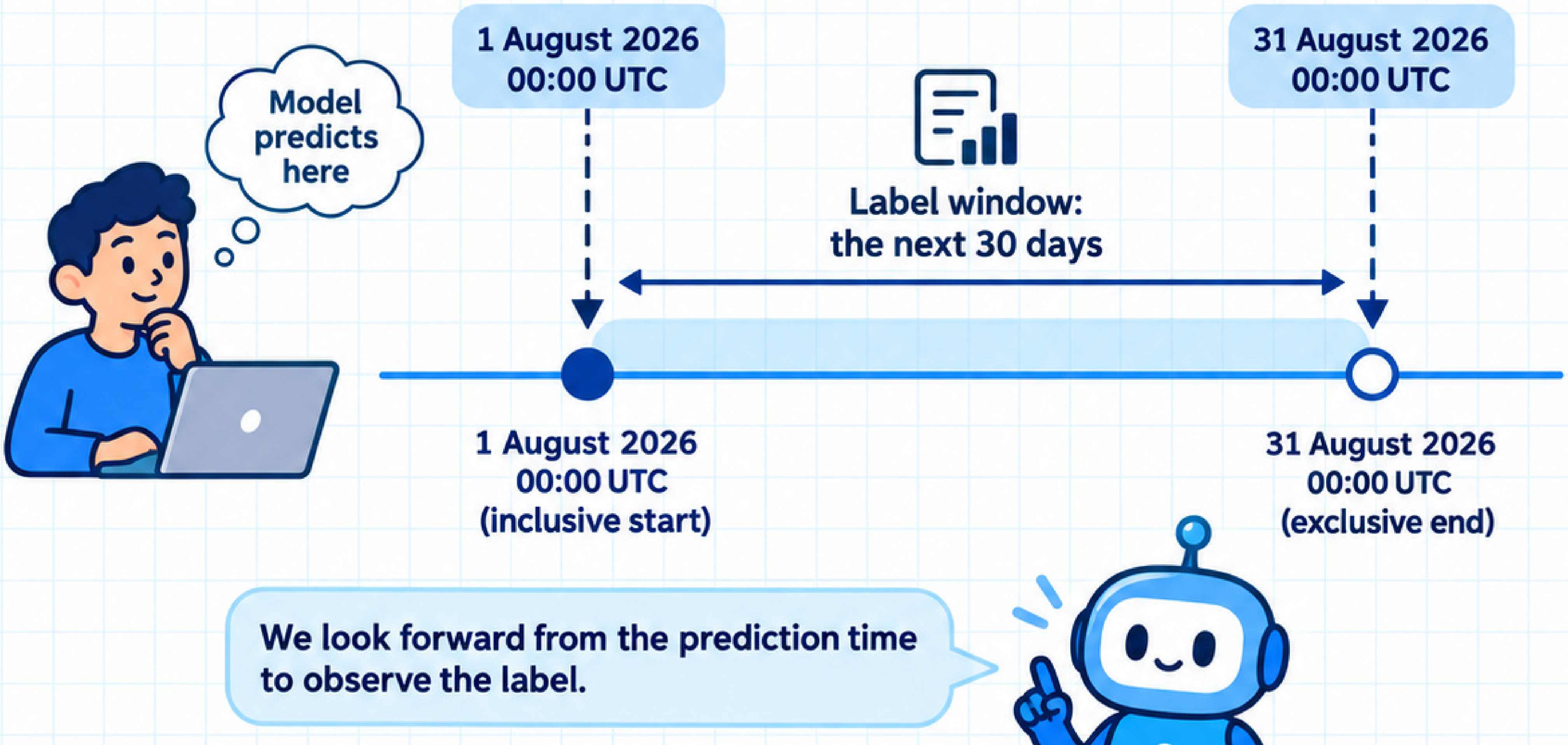
Define the window and aggregation for every feature. Different features can use different windows when the design calls for them.



LOOK FORWARD TO OBSERVE THE LABEL

Label window: the next 30 days.
Include events from 1 August 2026, 00:00 UTC.
Exclude events at or after 31 August 2026, 00:00 UTC.

The model predicts before this outcome is known.
The training dataset receives the label later, after enough evidence is available.



PUT THE TWO WINDOWS ON ONE TIMELINE

Past observations → prediction → future outcome

Observation window
(build features)

2 July 2026
00:00
(included)

1 August 2026
00:00
(excluded)

Label window
(observe purchases)

1 August 2026
00:00
(included)

31 August 2026
00:00
(excluded)

2 July to
1 August:
build features.

1 August:
produce the
inactivity
score.

1 August to
31 August:
observe
purchases
for the label.

The feature window and label window answer different questions. Keep their time boundaries visible in the dataset design.

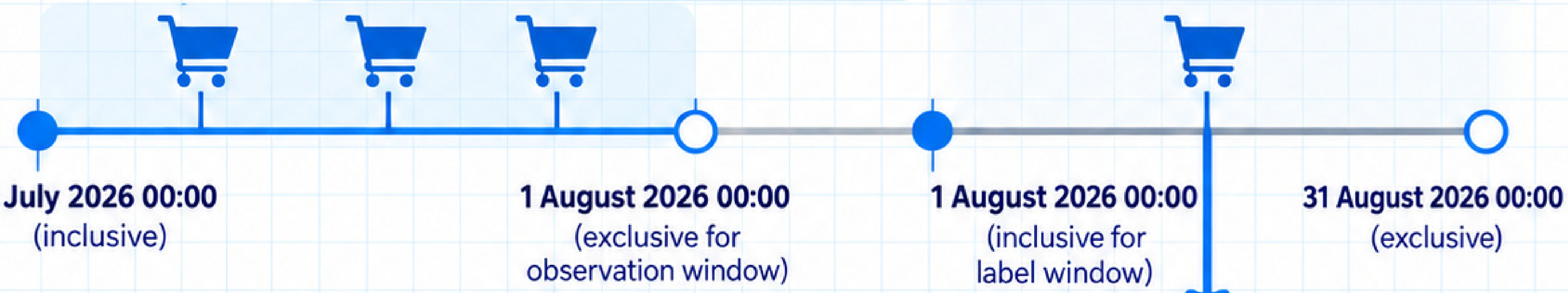
BUILD ONE COMPLETE EXAMPLE

Customer A had three completed purchases in the observation window.

Customer A

Observation window (for features)
[2 July 2026 00:00,
1 August 2026 00:00)

Label window (for outcome)
[1 August 2026 00:00,
31 August 2026 00:00)



At prediction time, that history becomes a feature:

purchase_count_30d	3

Customer A then purchases on 10 August.

Their label is 0.

The future purchase creates the label; it is not part of the earlier feature count.

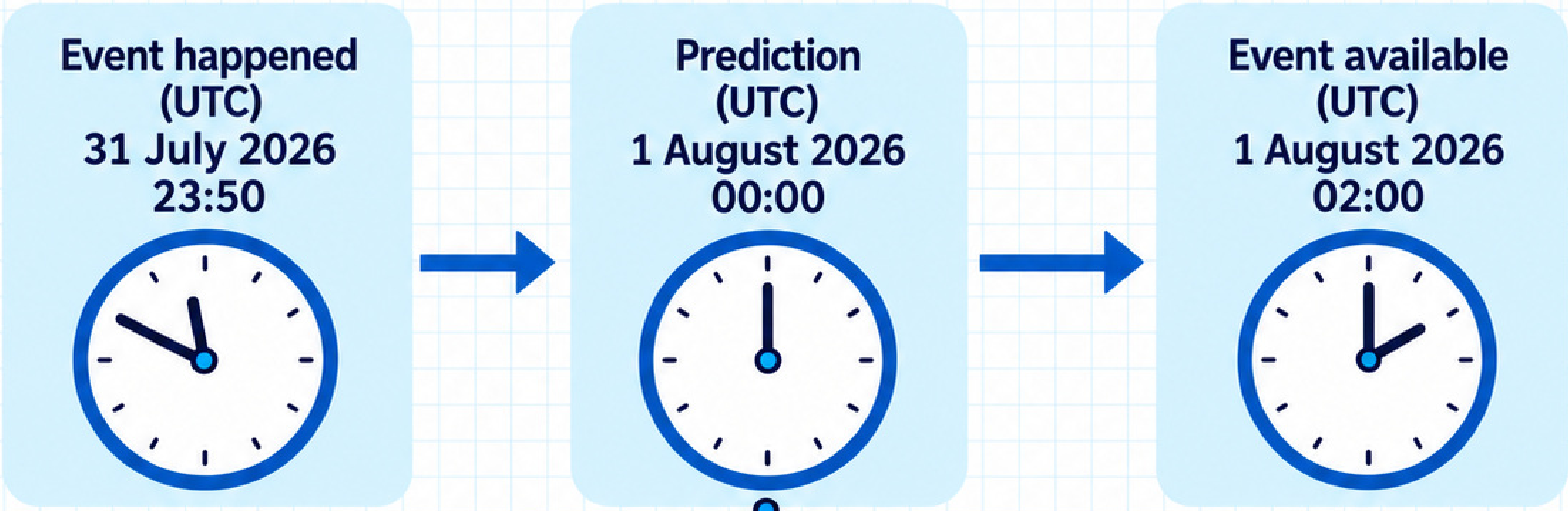
EVENT TIME AND AVAILABILITY ARE DIFFERENT

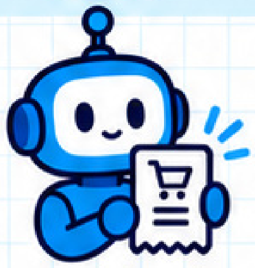
A purchase happened on 31 July at 23:50.

Its event record became available on 1 August at 02:00.

It happened before our prediction, but arrived after it.

Our historical feature reconstruction must respect both facts. Store when an event happened and when the system could actually use it.





LABELS ARRIVE WITH A DELAY

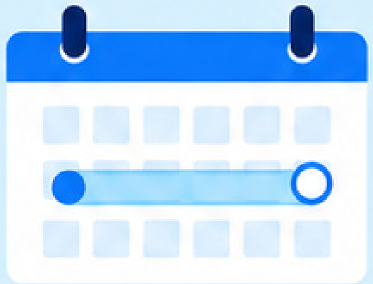
No purchase by 15 August does not prove 30-day inactivity.

The outcome window is still open.

After it closes, late event delivery can create another delay. Our example waits 48 hours and checks source completeness before finalizing the cohort.

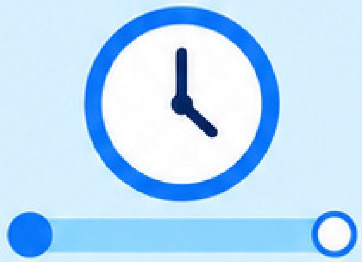
Prediction:
1 August 00:00 UTC

Outcome window:
30 days



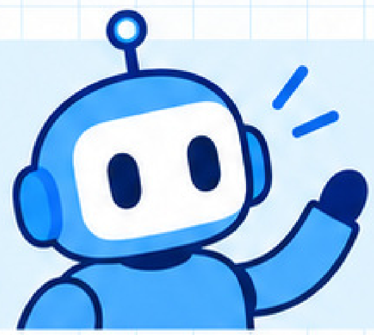
1 August 2026 00:00 UTC
to
31 August 2026 00:00 UTC
(30 days)

After 31 August 00:00 UTC:
illustrative 48-hour buffer



31 August 2026 00:00 UTC
to
2 September 2026 00:00 UTC
(48 hours)

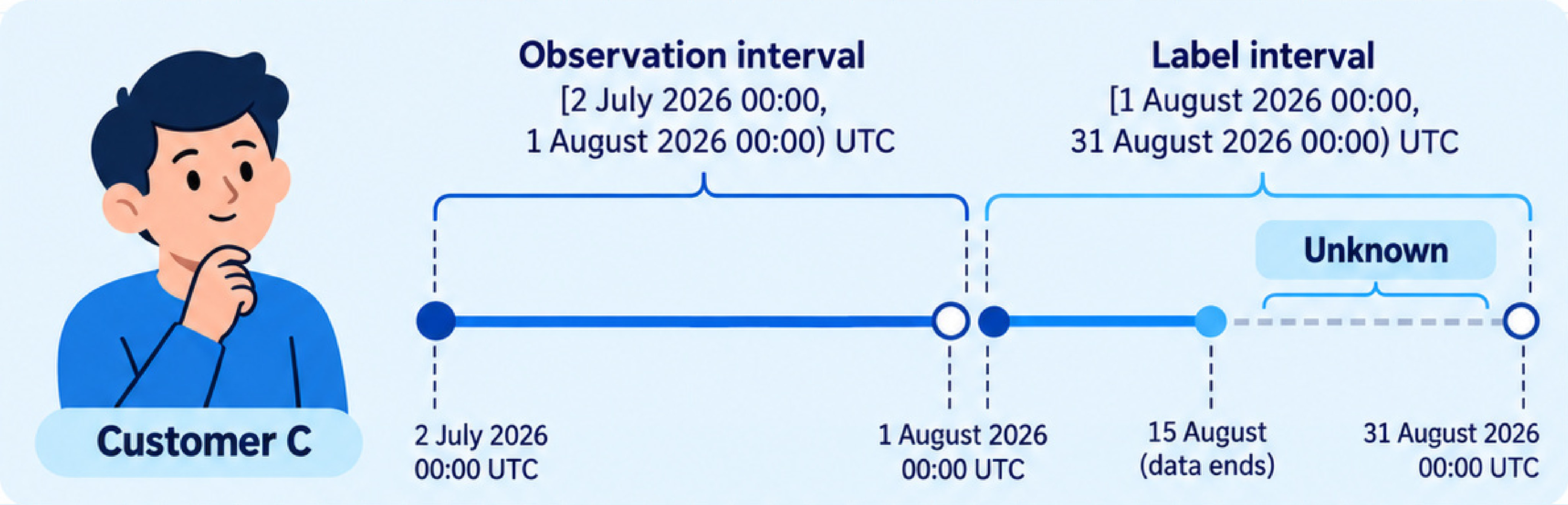
Source completeness check



That buffer is a teaching assumption, not a universal guarantee.

INCOMPLETE OBSERVATION IS NOT A NEGATIVE EXAMPLE

Customer C has no recorded purchase, but their data ends on 15 August.

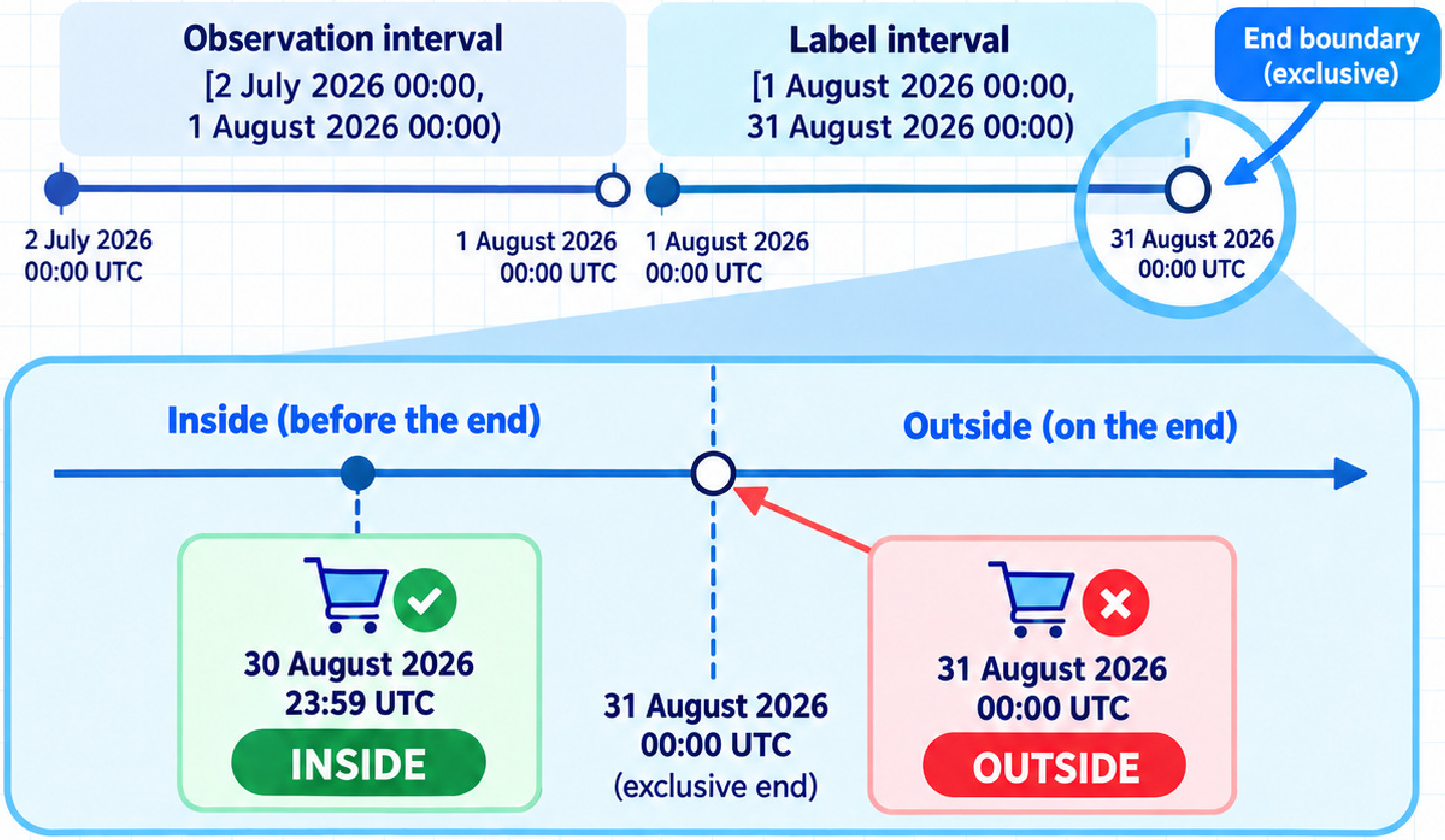


We cannot conclude what happened during the remaining window.

Keep the label unknown until sufficient observation exists. This is incomplete follow-up, often called censoring.

For our simple classifier, keep unresolved examples out of the finalized training cohort.

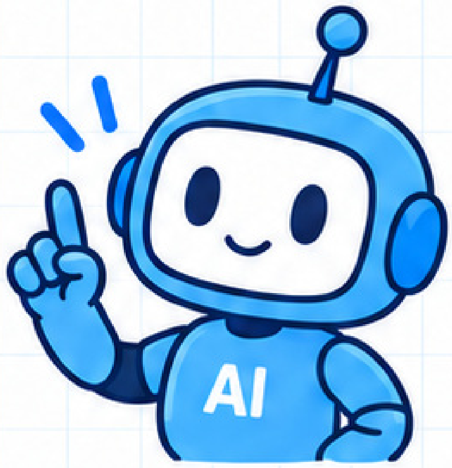
WRITE DOWN THE BOUNDARY RULE



A completed purchase at 30 August, 23:59 UTC is inside the window.

A completed purchase at 31 August, 00:00 UTC is outside it.

Use one explicit timezone and an inclusive start with an exclusive end. Boundary tests catch mistakes that an ordinary mid-month example will miss.



DEFINE WHAT COUNTS AS A PURCHASE

Our event is purchase_completed.

Completed purchase

Counts as purchase_completed.

Counts ✓

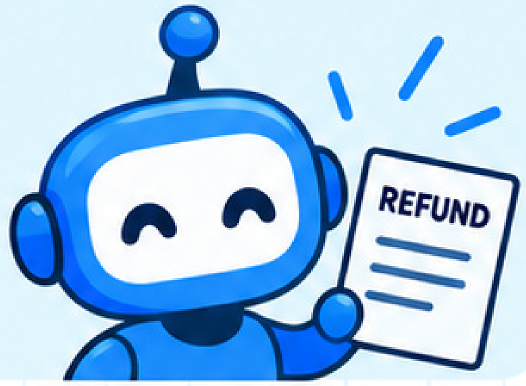
Cancelled checkout

A cancelled checkout does not count.

Does not count ✗

Refund (separate event)

For this toy definition, a later refund does not erase the completed-purchase event. Another business might choose a different outcome.



Document event meaning, corrections and exclusions. Otherwise, two teams can generate different labels from the same customer history.

A SCORE IS NOT THE OBSERVED LABEL

The model might output an inactivity probability of 0.7.

That is a prediction, not a label and not a guarantee.

The label comes from the defined outcome window. Whether to contact the customer is another decision, involving cost, permissions and evidence that the intervention helps.

MODEL SCORE

The model might output an inactivity probability of 0.7.

OBSERVED OUTCOME

The label comes from observed purchases in the defined outcome window.

APPLICATION DECISION

The label comes from the defined outcome window. Whether to contact the customer is another decision, involving cost, permissions and evidence that the intervention helps.

- A score is a model prediction.
- A label is the observed outcome.
- A score is not a label.
- A score is not a guarantee.
- Contacting a customer is a separate decision.
- That decision involves cost, permissions and evidence that the intervention helps.

MAKE A LABEL CONTRACT

Label Contract v1

- Eligible population and row identity
- Prediction timestamp and timezone
- Feature windows and availability rules
- Outcome event and label window
- Observation completeness and label delay
- Corrections and definition version

Changing any of these can change the learning problem, even when the label column keeps the same name.

TOOLS CAN HELP RECONSTRUCT THE PAST

SQL

SQL can build customer snapshots and time-window aggregates.

Customer snapshots and aggregates

Feast

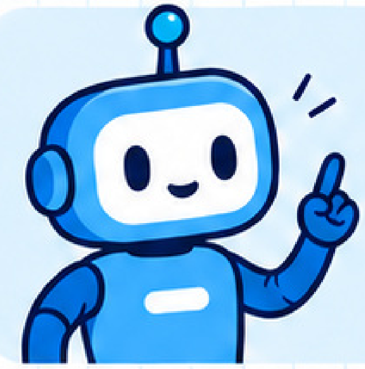
Feast supports historical feature retrieval with point-in-time joins.

Retrieve historical feature values

Tests

Tests can check boundaries and known labels.

Check known cases and boundaries



Inspect the timestamp semantics of any tool.
Event-time filtering alone may not reproduce when late or backfilled data became available to your application.

Check event time AND availability.

YOUR TURN: LABEL THREE CUSTOMERS

Use our 1 August prediction and 30-day outcome window.

A

A: completed purchase on 10 August.

Observation interval
[2 July 2026 00:00,
1 August 2026 00:00)

Label interval
[1 August 2026 00:00,
31 August 2026 00:00)

2 July 2026 00:00 UTC — 1 August 2026 00:00 UTC — 10 August 2026 00:00 UTC — 31 August 2026 00:00 UTC

Label (0, 1 or unknown):

B

B: no purchase; observation is complete through the window.

Observation interval
[2 July 2026 00:00,
1 August 2026 00:00)

Label interval
[1 August 2026 00:00,
31 August 2026 00:00)

2 July 2026 00:00 UTC — 1 August 2026 00:00 UTC — 31 August 2026 00:00 UTC

Label (0, 1 or unknown):

C

C: no purchase; observation stops on 15 August.

Observation interval
[2 July 2026 00:00,
1 August 2026 00:00)

Label interval
[1 August 2026 00:00,
31 August 2026 00:00)

2 July 2026 00:00 UTC — 1 August 2026 00:00 UTC — 15 August 2026 00:00 UTC — 31 August 2026 00:00 UTC

Label (0, 1 or unknown):

Assign 0, 1 or unknown. Then explain whether a 2 August feature update belongs in the prediction snapshot.

A CLEAR LABEL MAKES A CLEARER SYSTEM

- **Define the outcome.**
- **Freeze the prediction moment.**
- **Reconstruct available history.**
- **Wait for enough outcome evidence.**

These decisions shape collection, training and evaluation before we choose a model.

Observation interval
[2 July 2026 00:00, 1 August 2026 00:00) UTC

Prediction moment
1 August 2026 00:00 UTC

Label interval
[1 August 2026 00:00, 31 August 2026 00:00) UTC

For your own product, write the population, observation window, prediction moment and label window.

DAY 8: Checking data quality