

Facilitator guide: Module 9, Machine learning on auction data

A teacher-facing companion to Module 9. The module is the full lesson; this guide adds the facilitation layer: a preparation checklist, timing cues, guidance for the discussion questions, and notes on what to emphasize and where students get stuck. Slides and a printable version are below.

At a glance

- **Length:** 90 minutes. A 60 minute and a 120 minute variant are noted under Teaching notes.
- **Level:** advanced undergraduates or graduate students in an introductory machine learning, applied data science, or quantitative methods course, who have taken or are concurrently taking a first course in statistics or machine learning.
- **Access:** the no-code sandbox at sandbox.altfndata.com is the primary path, and is all that is needed for the core session. No API key is required. An optional Python extension, fitting a baseline regression model and a baseline classification model on an exported training extract, requires a class API key from info@altfndata.com and a standard Python data science environment.
- **Goal:** students leave able to separate pre-sale features from outcome fields for a price-prediction and a sold-versus-unsold task, explain target leakage in their own words, and justify a point-in-time train/test split by `sale_date` rather than a random split.

Before class

- Register your own sandbox account and run both guided queries end to end, so you have a known-good training extract and cutoff-count result ready to show.
- Prepare the before/after diagram or handout distinguishing fields known before a sale from outcomes known only after, for the feature-framing segment.
- Open the schema tab on the watches data table beforehand and confirm the exact documented field list you will read aloud to the class.
- Load the slides and confirm the projector or screen share shows the SQL editor and export tab clearly, including small text in the query results.
- Optional: if you plan to run the Python extension, request a class API key from info@altfndata.com ahead of time, and confirm the exported training extract loads and both baseline models fit in your environment.

Timed agenda with cues

Time	Segment	What to do	Watch for
0 to 15	Introduce the two tasks and the leakage trap	Introduce price prediction and sold/unsold classification, and preview that both tasks contain a trap: using information not available at prediction time.	Students assuming the model-fitting is the hard part. The framing and leakage discussion is the load-bearing content, not a model.

Time	Segment	What to do	Watch for
15 to 30	Feature framing	As a class, sort the documented fields into known-before-the-sale and known-only-after-the-sale, building the diagram on the board.	Students placing <code>sale_estimates_high_usd_price</code> in the "after" column. Stress that a house sets its estimate ahead of the sale.
30 to 45	Leakage and point-in-time discussion	Anchor on why a random train/test split would overstate performance and why a split by <code>sale_date</code> is required instead.	Students nodding along without being able to say specifically what a random split hides. Ask them to state it in their own words.
45 to 60	Guided demo: export a training extract	Use the sandbox to explore candidate features for one category and export a training extract limited to pre-sale fields.	Including <code>usd_price_decimal</code> or <code>status</code> among the features by accident. Check the exported field list together before moving on.
60 to 75	Small-group exercise	Groups sketch a feature list and a train/test split plan for a category or brand of their choosing, and identify one leakage risk specific to their plan.	Groups picking a cutoff date after eyeballing results. Remind them the cutoff must be fixed in advance, not chosen afterward.
75 to 85	Class discussion	Groups present their feature lists and leakage risks for critique.	Vague leakage answers like "overfitting." Push groups for the specific field or mechanism behind the risk.
85 to 90	Wrap-up and homework	Restate the one-sentence takeaway and describe the optional Python extension for students who want to fit an actual model.	Running short on time for the optional-extension explanation. Keep it to two sentences if the clock is tight.

Guidance for the discussion questions

Use these as the "what to listen for" behind each question in the module. They are talking points, not a graded key.

- 1. Why using `usd_price_decimal` to predict `usd_price_decimal` is leakage.** Listen for: it feels circular because it is, but the subtler point is that any field derived from or correlated with the outcome after the fact, not only the outcome itself, has the same problem. The test is always whether this value would have existed at the moment the prediction needed to be made.
- 2. Whether the high estimate is a legitimate feature, and where it gets weaker.** Listen for: yes, since a house sets it before the sale, so it is a genuine pre-sale feature for both tasks. It becomes less reliable for a lot type or category the house rarely handles, where the house has less basis for an accurate estimate and the field itself becomes noisier.
- 3. What a random split hides that a date-based split reveals.** Listen for: a random split can put future rows in training and past rows in test, letting the model implicitly learn from information that would not have existed yet. This inflates reported performance in a way that will not survive contact with real, forward-only use.
- 4. What to ask before trusting 90 percent classification accuracy.** Listen for: what fraction of lots sell overall. If the base rate is already close to 90 percent, a model that always predicts "sold" reaches the same accuracy without having learned anything.
- 5. How designer or vendor identity could leak information subtly.** Listen for: if designer or vendor encodes information correlated with the outcome only because of how the data was collected or curated, for example a vendor that only lists lots it expects to sell, that identity becomes a proxy for the outcome

rather than a genuine predictive signal. Checking requires looking at how the field's categories relate to the target's distribution, not just treating it as an innocuous input.

6. What changes when the target becomes a category's future median price. Listen for: the feature list would shift toward category-level and time-aggregated features rather than single-lot attributes, and a new leakage risk appears, using data from the same future quarter being predicted, which the single-lot framing does not raise in the same way.

7. What real-world decision a price model would support, and the cost of leakage. Listen for: a dealer or house sizing a reserve or estimate, or a buyer judging a fair price. A model trained on leaked data would report strong performance in testing but would make poor real decisions once deployed, since it never actually learned to predict from information available in advance.

8. Why the cutoff date must be fixed before looking at test performance. Listen for: choosing the cutoff after seeing results lets you implicitly select the split that makes the number look best, which is the same overstatement problem a date-based split is meant to solve, just moved to a later step in the process.

Teaching notes

- **Most common misconception:** students think leakage only happens when you literally include the outcome field by name. Work through the designer/vendor question live, showing how a field can proxy for the outcome without ever naming it.
- **Second misconception:** assuming any train/test split is fine as long as a large enough percentage is held out. Show the row-count query results for two different cutoff dates side by side and ask what a random 80/20 split would have hidden that the date-based counts reveal.
- **60 minute variant:** shorten the feature-framing board exercise to a pre-built diagram you present rather than build live, and drop the small-group exercise in favor of a single class-wide feature list; keep the leakage discussion and guided demo intact.
- **120 minute variant:** after the guided demo, have groups additionally run the class-balance query and discuss why 90 percent accuracy on an imbalanced sold/unsold split can be deceptive, then present that alongside their feature lists.
- **If the sandbox is slow or blocked on the room network:** fall back to the pre-run query results and training extract you captured in preparation, walk the before/after diagram and leakage discussion entirely on the board, and assign the sandbox export as the first step of homework.

For the full lesson content, queries, and homework, see Module 9. Questions or a class API key: info@altfndata.com.