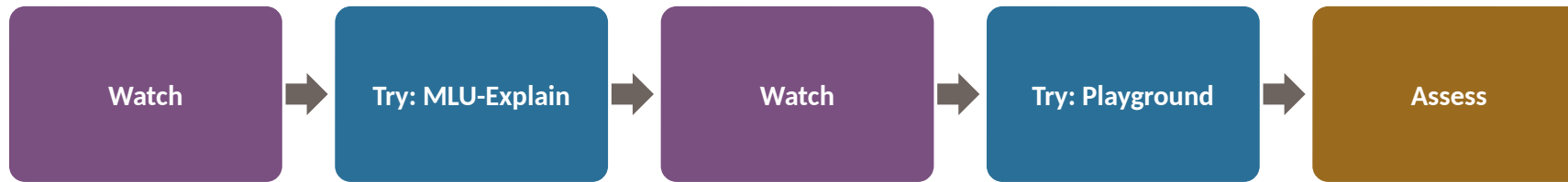


## MODULE 4

# Supervised learning: regression and classification

Choose the problem type first, fit a model you can interpret, and prove it with a leakage-safe validation.

## Watch a little, then do a little



*Each recorded segment is short. You pause, practice the concept hands-on, then return for the next piece.*

## Where this module sits in the lifecycle



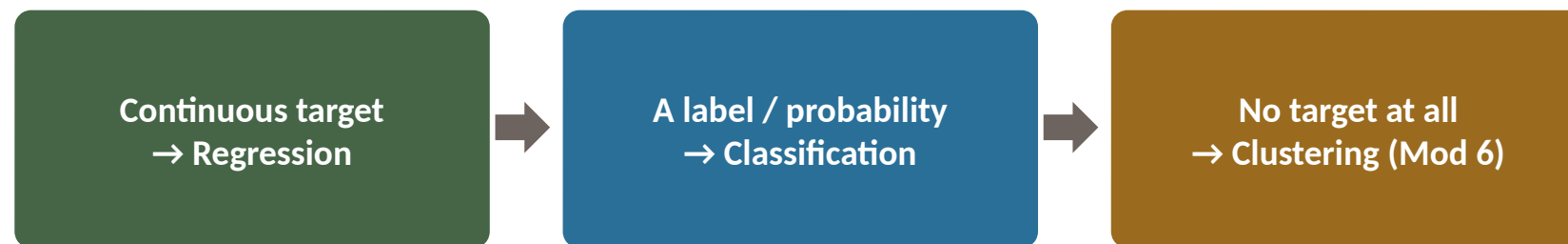
This is the model layer. A continuous target points to regression, a label to classification, and the choice constrains everything downstream, but the real deliverable is an honest estimate of out-of-sample performance.

PART 1 OF 2

# Choosing the model and reading it

*About 12 minutes on camera*

# Choose the problem type first



*The framing decision constrains method, metric, and validation. Diagnose which one a stated business or research question actually requires before touching a model.*

# Every family carries a bias

## Linear models

Assume an additive, roughly linear signal; pay for violations with underfitting. Win when data are scarce or you need transparent effects.

## Tree ensembles

Low bias, average away variance. Win when the signal has interactions and nonlinearities on wildly different scales.

## A coefficient

Change in expected outcome per unit of a predictor, holding others fixed. If you cannot say the sentence, you are not done.

## An odds ratio

The exponent of a logistic coefficient, a change in the odds. Interpretation comes bundled with the fit.

NOW TRY IT

## MLU-Explain: regression, logistic, random forest

- Drag points in Linear Regression to feel how least squares responds to leverage and outliers.
- Connect the sigmoid in Logistic Regression to the log-odds interpretation.
- Watch Random Forest average many decorrelated trees into a smooth boundary.

*About 20 minutes, then come back for Part 2.*

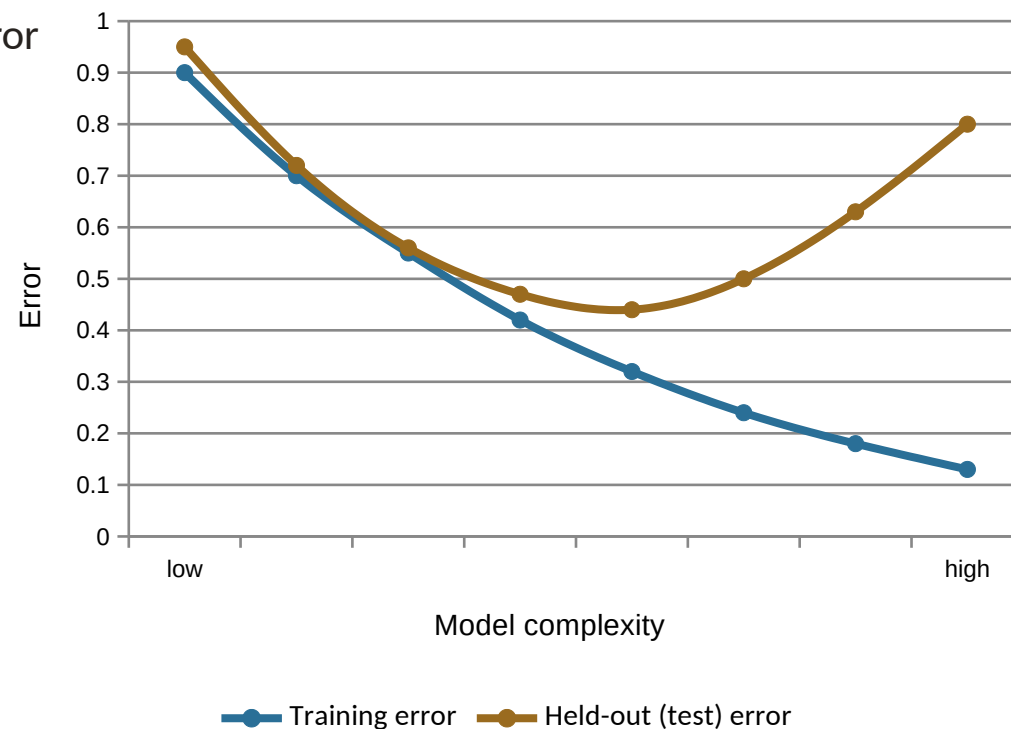
PART 2 OF 2

# Bias, variance, and honest validation

*About 13 minutes on camera*

# The bias–variance trade-off is not abstract

- Training error keeps dropping as complexity rises, while held-out error turns back up. That gap is the whole story.
- Model selection must happen on data the model never touched.
- A number without a leakage-safe split behind it is decoration, not evidence.



# The private leaderboard

## Public leaderboard

a minority of hidden test rows, updated live, easy to chase

## Private leaderboard

the remaining majority, revealed only at close, the real score

*Overfit the public board and the private board punishes you. Honest validation is the only defense.*

NOW TRY IT

## TensorFlow Playground and R2D3

- In TensorFlow Playground, use only the spiral and circle datasets and watch a purely linear model fail where added features and capacity succeed.
- Read R2D3's visual introduction to see how a tree partitions feature space and how depth trades bias for variance.

*About 15 minutes, then build the formative comparison.*

## Formative deliverable (completion points)

- On the shared dataset in the module folder, fit a baseline model (intercept-only or single-feature) and one stronger model (full regression or a tuned random forest).
- Report a single held-out metric for each (RMSE or MAE for regression; ROC-AUC or F1 for classification), chosen and justified for the target.
- Write two sentences on why the stronger model does or does not help. A stronger model that fails to beat the baseline is a legitimate result, say so.

**How it counts.** Submit the work with a short reflection or evidence you ran it. Credited on completion, not scored for correctness. This is the practice that makes the graded simulation go well.

## GRADED ASSESSMENT

# The Private-Leaderboard Competition

- You compete in a hosted prediction competition on a client task, forecasting an outcome from tabular data with a hidden test truth.
- Each submission is scored on a live public leaderboard (a minority of rows) and a private leaderboard (the majority), revealed only at close.
- Chasing the public board overfits; the private board rewards the honest, leakage-safe validation this module builds.