

From clean data to useful predictions

Transformations, pipelines & leakage

Build on what we learned about
data cleaning and quality.

Mohammad Sabokrou

New Uzbekistan University

THE DATA WE HAVE

Student	Score	Minutes
A	70	60
B	80	180
New	79	100



THE QUESTION WE ASK

What happens next?

We checked the data. What can we answer now?

Start from Week 3: data cleaning and quality.

WHAT WE ALREADY HAVE

We checked errors, missing values and duplicates.

Student	Score	Minutes
Past A	70	60
Past B	80	180
New	79	100

Illustrative rows; any remaining limitations are documented, not hidden.

TODAY'S QUESTION

At the end of Week 4, will this student withdraw later?

To answer it, what do we need?

- 1 What we know about this student now**
For example, an early score and study time.
- 2 Past students with known later outcomes**
Use their records from the same point in the course.
- 3 A way to predict and check the answer**
Compare examples, then check on other students.

Cleaning makes the records more reliable. Next, we prepare them for prediction.

Turn the question into inputs and an answer

End of Week 4: what is already known, and what do we want to predict?

ONE OBSERVATION

One row = one student.

Early score: 79

Study minutes: 100

Programme: CS

FEATURES = INPUTS

Measurements available now.

They describe what we know before the decision.

TARGET = ANSWER

Withdraw later?

YES or NO

Unknown for this new student.

A model is a rule that uses input measurements to produce a prediction.

A first rule: copy the closest past student

One feature first: early score. Distance means the size of the difference.

Student	Early score	Known outcome
Past A	70	Stay
Past B	80	Withdraw
NEW	79	?

$$d = |x_{\text{new}} - x_{\text{past}}|$$

$$d(A) = |79 - 70| = 9$$

$$d(B) = |79 - 80| = 1$$

1 < 9, so B is closer. This toy rule predicts: WITHDRAW.

Add minutes: the nearest student changes

The same A, B and NEW students; now combine two feature differences.

Student	Score	Minutes	Outcome
A	70	60	Stay
B	80	180	Withdraw
NEW	79	100	?

$$d = \sqrt{(\text{score gap})^2 + (\text{minutes gap})^2}$$

$$\begin{aligned} d(A) &= \sqrt{(79 - 70)^2 + (100 - 60)^2} \\ &= \sqrt{81 + 1600} = 41 \end{aligned}$$

$$d(B) = \sqrt{1 + 6400} \approx 80.01$$

Raw distance chooses A (STAY). The large minutes differences dominate.

Scaling changes the representation, not the student

Distance-based rules are sensitive to the size of each feature's numbers.

SCORE: POINTS

A score gap of 9 contributes 81 to the squared distance.



TIME: MINUTES

A minutes gap of 40 contributes 1600. Changing minutes to seconds would magnify it again.

Score²



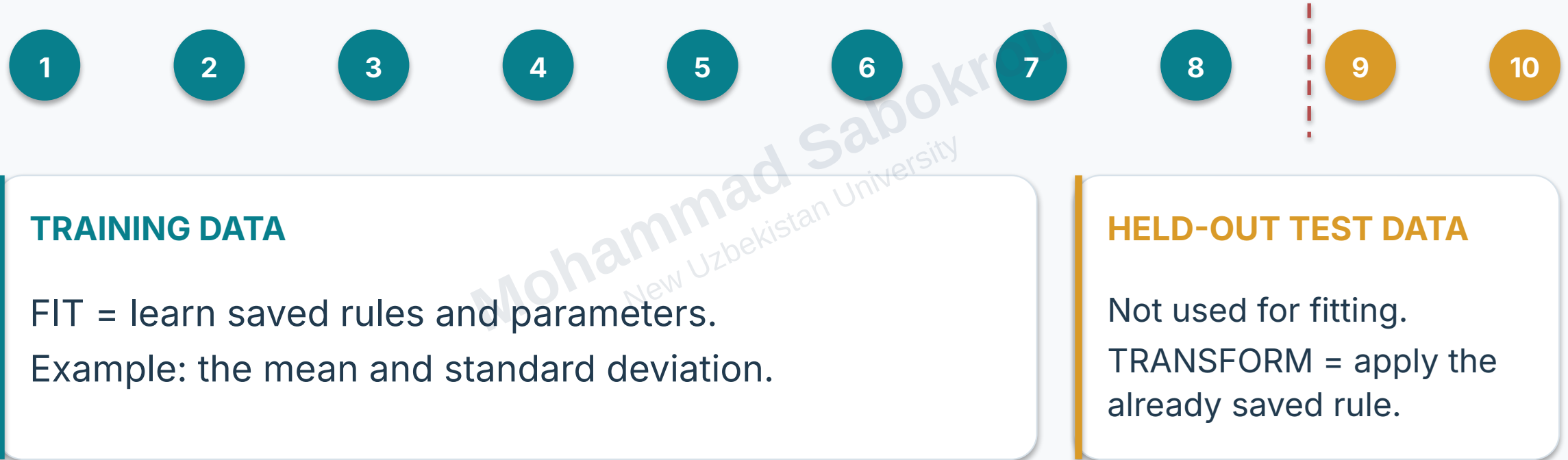
Minutes²



Scaling puts numerical features on a chosen common reference scale; it does not guarantee better predictions.

Training learns. Test checks.

Set aside test students before learning any data-dependent preprocessing.



Use the training rule on training, test and future rows — never learn a separate test scale.

Standardisation: fit the mean and spread

Training A and B only: $n = 2$ values per feature; x_i denotes one training value.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

SCORE: [70, 80]

$$\mu_s = \frac{70 + 80}{2} = 75$$

$$\sigma_s = \sqrt{\frac{25 + 25}{2}} = 5$$

MINUTES: [60, 180]

$$\mu_m = \frac{60 + 180}{2} = 120$$

$$\sigma_m = \sqrt{\frac{3600 + 3600}{2}} = 60$$

μ = training mean; σ = training standard deviation. We divide by n , matching StandardScaler.

Standardise, then reuse exactly the same distance rule

z tells us how many training standard deviations a value is above or below the mean.

$$Z = \frac{X - \mu_{\text{train}}}{\sigma_{\text{train}}}$$

$$Z_{\text{score}} = \frac{79 - 75}{5} = 0.8$$

$$Z_{\text{minutes}} = \frac{100 - 120}{60} = -\frac{1}{3}$$

A becomes (-1, -1)

B becomes (+1, +1)

$$d(A) = \sqrt{1.8^2 + (2/3)^2} \approx 1.92$$

$$d(B) = \sqrt{(-0.2)^2 + (-4/3)^2} \approx 1.35$$

B is now closer → WITHDRAW

Same students. Same prediction rule. Different, explicitly chosen representation.

Min-max: where is x within the training range?

Fit the minimum and maximum using training logs [4, 12, 16].

$$x' = \frac{x - \min_{\text{train}}}{\max_{\text{train}} - \min_{\text{train}}}$$

$$12 \mapsto \frac{12 - 4}{16 - 4} = \frac{8}{12} \approx 0.667$$

SAVE TWO VALUES

Minimum = 4

Maximum = 16

Range = 12



The training minimum maps to 0; the training maximum maps to 1.

Question 1: can a scaled value exceed 1?

Think for 45 seconds, then show A, B or C. Give one reason.

Training minimum = 4 | Training maximum = 16

A future student has 22 logins. What should the saved rule return?

A 1.0 — clip it

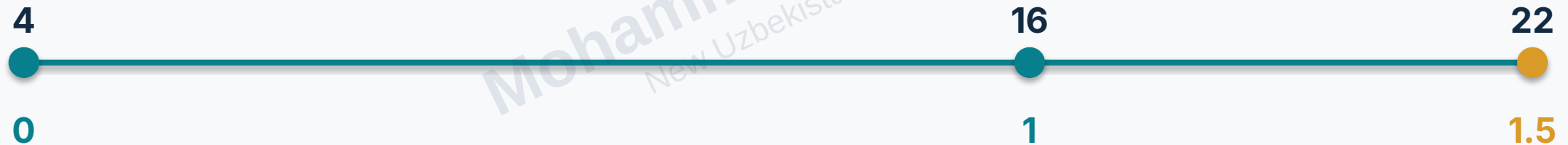
B 1.5 — keep the scale

C Refit with 22

Answer 1: B — the saved scale returns 1.5

A new value is allowed to lie outside the range observed during training.

$$22 \mapsto \frac{22 - 4}{16 - 4} = \frac{18}{12} = 1.5$$



Clipping is an extra policy.

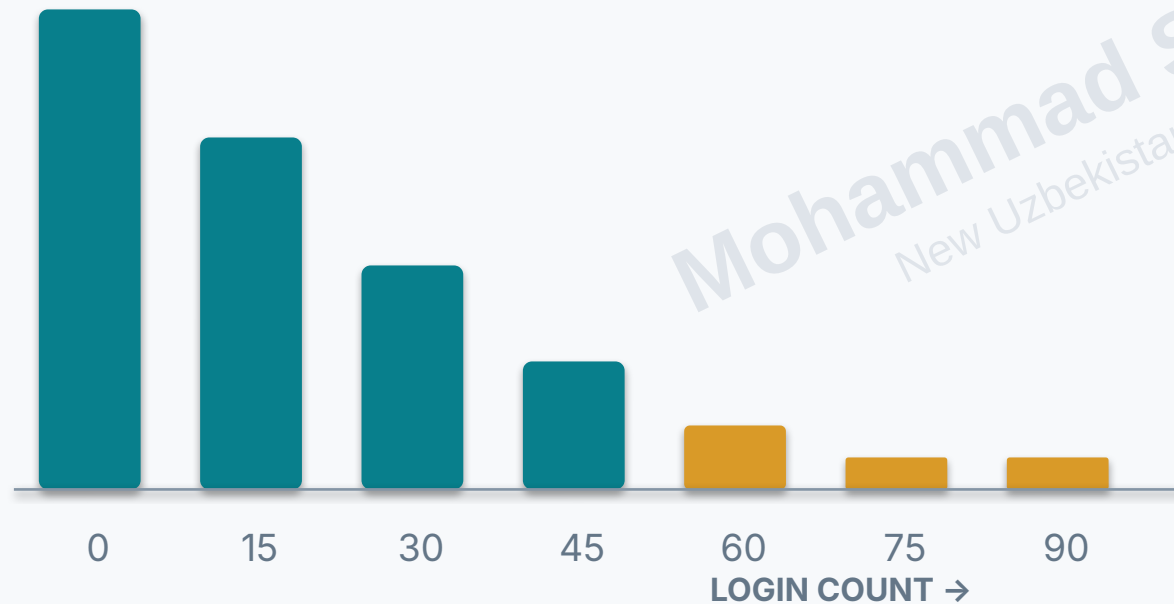
Refitting changes the reference scale.

Out-of-range does not mean the calculation is wrong. Apply the already fitted rule.

A long right tail means “a few very large values”

Most students have few logins; a small number have far more. This is not automatically bad data.

NUMBER OF STUDENTS



THE RIGHT TAIL

The few bars extending to the right represent unusually large counts.

A log transform compresses their numerical spacing.

Tail describes a distribution's shape — not missing data, an error, or proof of abnormal behaviour.

Log keeps the ordering, not the original distances

For non-negative values use $\log_{1p}(x) = \ln(1 + x)$. "Monotonic" means order-preserving.

$$X_1 < X_2 \Rightarrow \ln(1 + X_1) < \ln(1 + X_2)$$

RAW → LOG

0 → **0**

$\ln(1 + 0)$

RAW → LOG

9 → **2.303**

$\ln(10)$

RAW → LOG

99 → **4.605**

$\ln(100)$

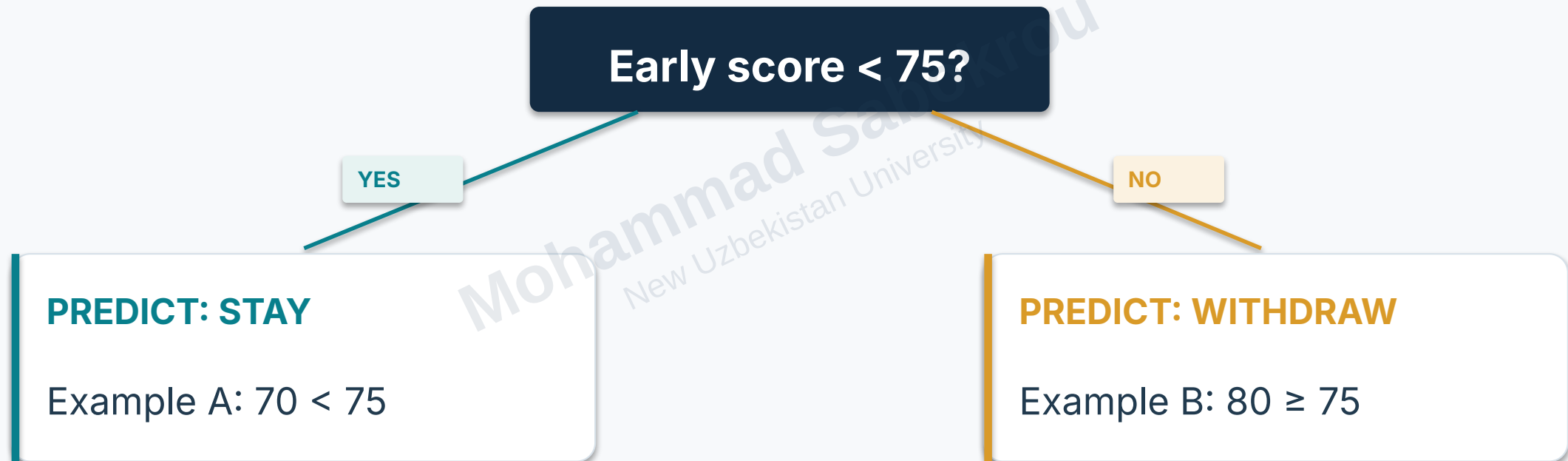
$$99 - 9 = 90$$

$$\ln(100) - \ln(10) = \ln(10) \approx 2.303$$

"More" stays "more". Gaps, ratios and units change; +1 makes zero valid.

A decision tree is a sequence of yes/no questions

A model is a prediction rule. A tree makes its prediction by following branches.



With $\mu = 75$ and $\sigma = 5$, "score < 75" becomes " $z < 0$ ": the same split.

Choose a transformation for a stated reason

Four situations, four visual examples — not a universal ranking.

DIFFERENT UNITS



Standardise

Compare features on a common spread scale.

REFERENCE RANGE



Min-max

Training endpoints become 0 and 1.

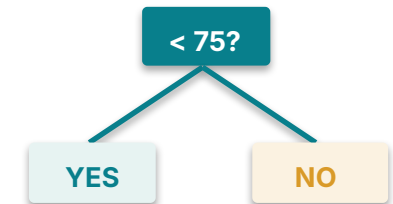
LONG RIGHT TAIL



$\ln(1 + x)$

Compress large numerical gaps.

TREE SPLITS



Often no scaling

Ordered threshold splits do not use distance.

Keep the original measurements, the fitted parameters, and the reason for the transformation.

Category names and category order are different

Encoding means representing a category with numbers a model can use.

NOMINAL: NAMES, NO ORDER

Programme:

CS, Economics, Psychology

There is no meaningful "bigger" programme.

ORDINAL: A MEANINGFUL ORDER

Self-reported workload:

Low < Medium < High

The order is meaningful.
Equal gaps are not guaranteed.

A number is a valid representation only when its implied relationships make sense.

One-hot encoding: one yes/no column per category

Do not make CS, Economics and Psychology look like equally spaced numbers.

ARBITRARY CODES

CS = 0

Economics = 1

Psychology = 2

An invented order.

Invented distances.

$$I_c(X) = \mathbf{1}\{X = c\}$$

Indicator = 1 when the statement is true; otherwise 0.

Programme	CS	Econ.	Psych.
CS	1	0	0
Economics	0	1	0
Psychology	0	0	1

CS becomes [1, 0, 0]. This records membership without inventing an ordinal ranking.

An unseen category needs a rule decided in advance

Suppose the fitted category order is [CS, Economics, Psychology].

NEW RECORD

Physics

Not seen during fit

POLICY A: UNKNOWN

Use a predeclared
Unknown/Other
column.

Physics → Unknown

POLICY B: IGNORE

Configured encoder:
Physics → [0, 0, 0]

Meaning: no known
match.

Choose one policy, document it, and monitor unknown labels. Do not refit on each new student.

Feature engineering: turn a history into one useful number

Question: how much recorded login activity occurred in the first four weeks?



$$\text{early_logins} = \sum_{w=1}^4 \text{logins}_w = 3 + 4 + 2 + 5 = 14$$

Define the source (login records), unit (count), window (W1-W4), and availability (after W4 is complete).

Same ratio does not mean the same activity

A new activity example: the ratio measures average minutes per login.

Student	Total minutes	Logins	Minutes / login
R	200	10	$200 \div 10 = 20$
S	100	5	$100 \div 5 = 20$

RATIO ONLY

R and S both look like [20].

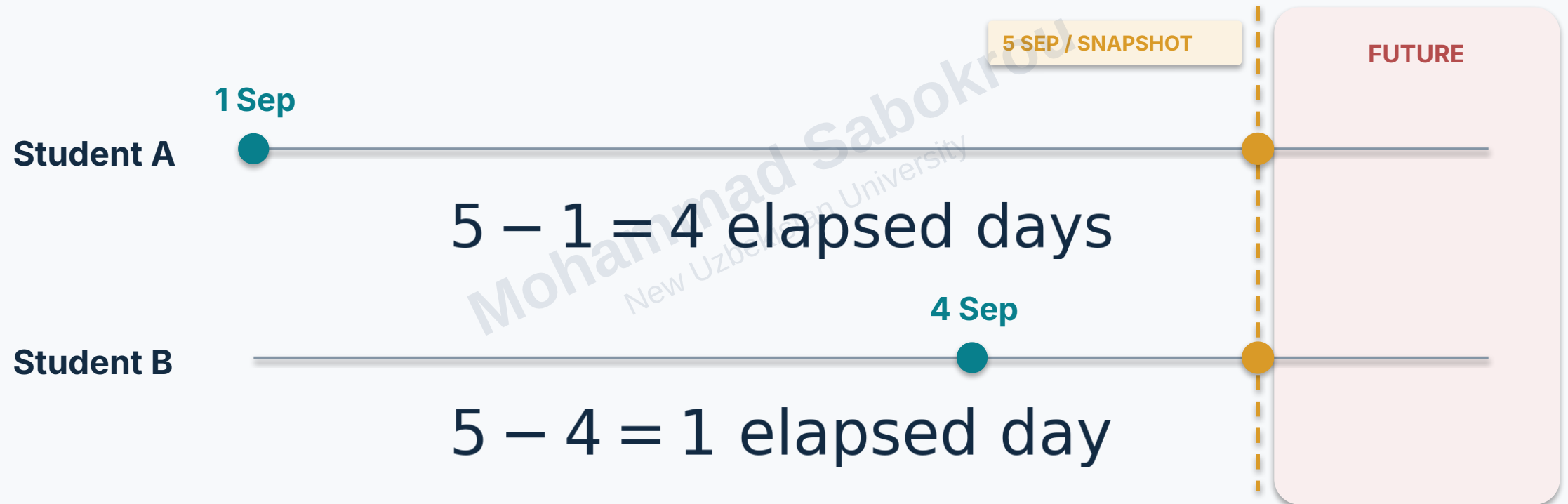
KEEP TOTALS WHEN THEY MATTER

$R = [200, 10, 20]$; $S = [100, 5, 20]$.

For zero logins, mark the ratio undefined/missing and keep a zero-login indicator. Do not divide by zero.

Snapshot = freeze the data at a specified time

Date-arithmetic mini-example: freeze both student records at 5 September.



Days since registration = snapshot date – verified first-registration date. A revision timestamp is a different event.

10-minute break

10

MINUTES

When we return

What was really available?

→ Safe pipeline → honest error

Would this value really be available at the end of W4?

The event date alone is not enough. Check when the value reached the prediction system.

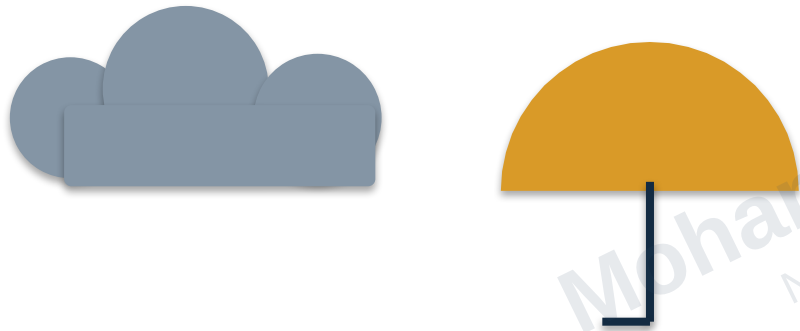
Candidate input	Decision	Reason
W1–W4 logins, already visible	USE	Observed and accessible by W4
Programme recorded at enrolment	USE	The historical version is known
Exam date announced in W1	USE	A future schedule can be known now
W3 quiz score released in W5	DO NOT USE	Quiz happened; score was not yet available
"Current GPA" updated in W6	DO NOT USE	Today's file is not the W4 snapshot
Whole-semester attendance	DO NOT USE	Includes classes after W4

A future event's scheduled date can be safe; a past event's late-released result can be unsafe.

Leakage in everyday life: should I take an umbrella?

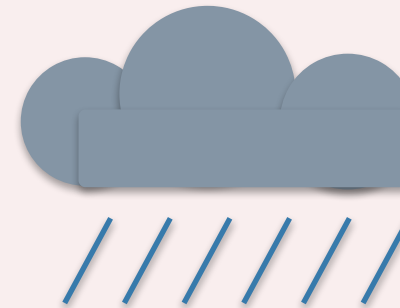
At 08:00, choose using what you can know at 08:00 — not what you will learn at 15:00.

08:00 / MAKE THE DECISION



**Current clouds + forecast
are available clues.**

15:00 / OUTCOME KNOWN



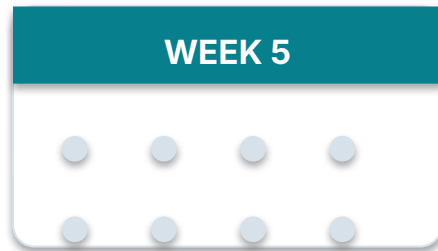
**"It actually rained at 15:00"
is an unavailable future clue.**

Now return to students: W4 inputs may not include realised W5 activity or the later withdrawal outcome.

Three leakage routes — and three different checks

A feature can be available now and still be used in a statistically invalid way.

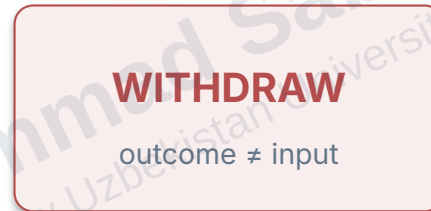
FUTURE LEAKAGE



Use W5 logins in a W4 prediction.

Check availability at prediction time.

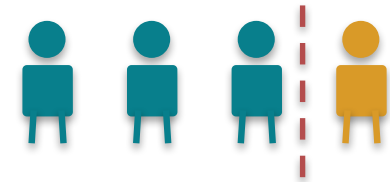
TARGET LEAKAGE



Use “will withdraw later” as an input.

Check whether the input reveals the answer.

TEST LEAKAGE



Fit the mean using held-out test rows.

Check who was allowed to influence fitting.

Past-cohort outcomes are valid training labels. The problem is sneaking unavailable answers into the inputs.

Question 2: the quiz happened before the snapshot. Safe?

Vote, then explain which time matters. Answer on the next slide.

WEEK 3

Quiz taken

END OF WEEK 4

Predict withdrawal

WEEK 5

Score first released

**Can the W3 quiz score be an input
for the end-of-W4 prediction?**

A Yes — the quiz already happened. B No — the score was not yet accessible.

Answer 2: B — a past event can still have a late result

The score is unavailable at the moment of the real prediction.

EVENT TIME

W3: the student took the quiz.

This tells us when the event happened.

AVAILABILITY TIME

W5: the score first became accessible.

This is too late for W4.

Use a genuinely available substitute, omit the score, or change the decision time — do not backfill it.

Why missing values again? A new question this week

Impute = fill a missing model input with a defined replacement.

WEEK 3: DATA QUALITY

Why is it missing?

What does that absence mean?

WEEK 4: PREDICTION

Which replacement can we fit without using test or future data?

$[1, 2, \text{missing}, 5] \Rightarrow \text{median}([1, 2, 5]) = 2$

Fit median = 2 on observed training values; use 2 for missing training, test and future inputs. This is not recovered truth.

Pipeline, step 1: learn all rules from training rows

A pipeline is an ordered set of reusable transformations, followed by the prediction rule.

Training row	Score	Logins	Programme
1	60	2	CS
2	60	2	Economics
3	80	6	Psychology
4	80	6	CS

SAVE THESE RULES

Login median = 4

Score: $\mu = 70$, $\sigma = 10$

Logins: $\mu = 4$, $\sigma = 2$

Category order:

[CS, Econ., Psych.]

$$\text{median logins} = \frac{2 + 6}{2} = 4$$

Fit order: impute training values first, then learn the scaler from the completed training columns.

Pipeline, step 2: transform one future student

Raw input: score 78, logins missing, programme CS.

1 IMPUTE

Fill the missing login count.

missing $\rightarrow$ 4

$$z_s = \frac{78 - 70}{10} = 0.8$$

2 STANDARDISE

Rescale completed numbers.

score $\rightarrow$ 0.8; logins $\rightarrow$ 0

$$z_l = \frac{4 - 4}{2} = 0$$

3 ENCODE

Turn a category into indicators.

CS $\rightarrow$ [1, 0, 0]

[0.8, 0, 1, 0, 0]

The final vector goes to the prediction rule. No new median, scale, or category vocabulary is learned here.

Different column types take different routes

“Impute + standardise” simply means: fill it, then rescale it.

Input type	Route	Concrete example
Missing number	Impute, then standardise	Logins: missing $\rightarrow 4 \rightarrow (4-4)/2 = 0$
Observed number	Standardise	Score: 78 $\rightarrow (78-70)/10 = 0.8$
Category	One-hot encode	CS $\rightarrow [1, 0, 0]$
Verified date	Derive elapsed time	1 Sep $\rightarrow$ 5 Sep snapshot $\rightarrow$ 4 days
Identifier	Join; exclude as input	student_id $\rightarrow$ retained for tracking only

ColumnTransformer is a software tool for applying different routes to different column groups.

Audit a broken workflow: three information mistakes

End-of-W4 prediction. Each row has a specific failure — not a syntax error.



Combine training and test before fitting

TRAIN = [2, 4]; TEST = [100] → “training mean” = 35.33

Correct mean: $(2 + 4)/2 = 3$.



Use a later activity report

Use the student's W5 logins to predict at end W4.

Use only the accessible W1–W4 history.

WITHDRAW

outcome ≠ input

Put the answer in the input

“Will withdraw later = YES” becomes a feature.

Keep the outcome as a label, not a predictor.

Do not let held-out rows, future activity, or the answer itself enter the fitted input representation.

Audit the reported score: practice is not the exam

The fifth workflow mistake is evaluating only on the records used to fit.

MISLEADING REPORT

Fit on 10 students.
Predict those same 10 students.
Report 10/10 correct as evidence
that the model works on new
people.

AN HONEST CHECK

Keep other students out of fitting.
Use the saved pipeline on them.
Compare their predictions with
their true outcomes.

A 100% training score tells us how well the rule fitted familiar cases — not how it performs on unseen ones.

The corrected workflow keeps learning inside training

A clean order makes the process reproducible.

1 Define the question

Outcome, decision time, valid inputs

2 Separate the records

Training students / held-out test students

3 Fit preprocessing on training

Imputation, scaling and category rules

4 Transform both sets

Reuse the same saved preprocessing

5 Fit the prediction rule

Use transformed training + training labels

6 Evaluate once on test

Use test outcomes only to score predictions

The later test labels may be used to assess predictions — not to fit, pick, or repair the model.

Classification predicts a category

A class label is the name of the category the prediction chooses.

EMAIL



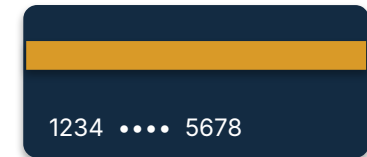
Spam / Not spam

STUDENT



Withdraw / Stay

TRANSACTION



Fraud / Normal

Training example: score + logins → known class.

New example: score + logins → predicted class.

Predicting a category is classification; predicting a numerical quantity is a different task.

Classification error = the fraction of wrong labels

Define every symbol, then substitute the example values.

$$\text{Err}_{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbf{1}\{\hat{y}_i \neq y_i\}$$

N_{test} : number of test examples
 y_i : true label; $\hat{y}_i$: predicted label
 $\mathbf{1}\{\dots\}$: 1 when wrong, 0 when correct

Test student	True label y	Predicted label $\hat{y}$	Wrong?
T1	Withdraw	Withdraw	0
T2	Stay	Withdraw	1
T3	Withdraw	Withdraw	0

$$\text{Err}_{\text{test}} = \frac{0 + 1 + 0}{3} = \frac{1}{3} \approx 33.3\%$$

$$\text{Accuracy} = 1 - \text{Err} \approx 66.7\%$$

Training error and test error answer different questions

Same error formula. Different records. Different interpretation.

TRAINING ERROR

Familiar fitting examples.

"How well did it fit
what it already saw?"

TEST ERROR

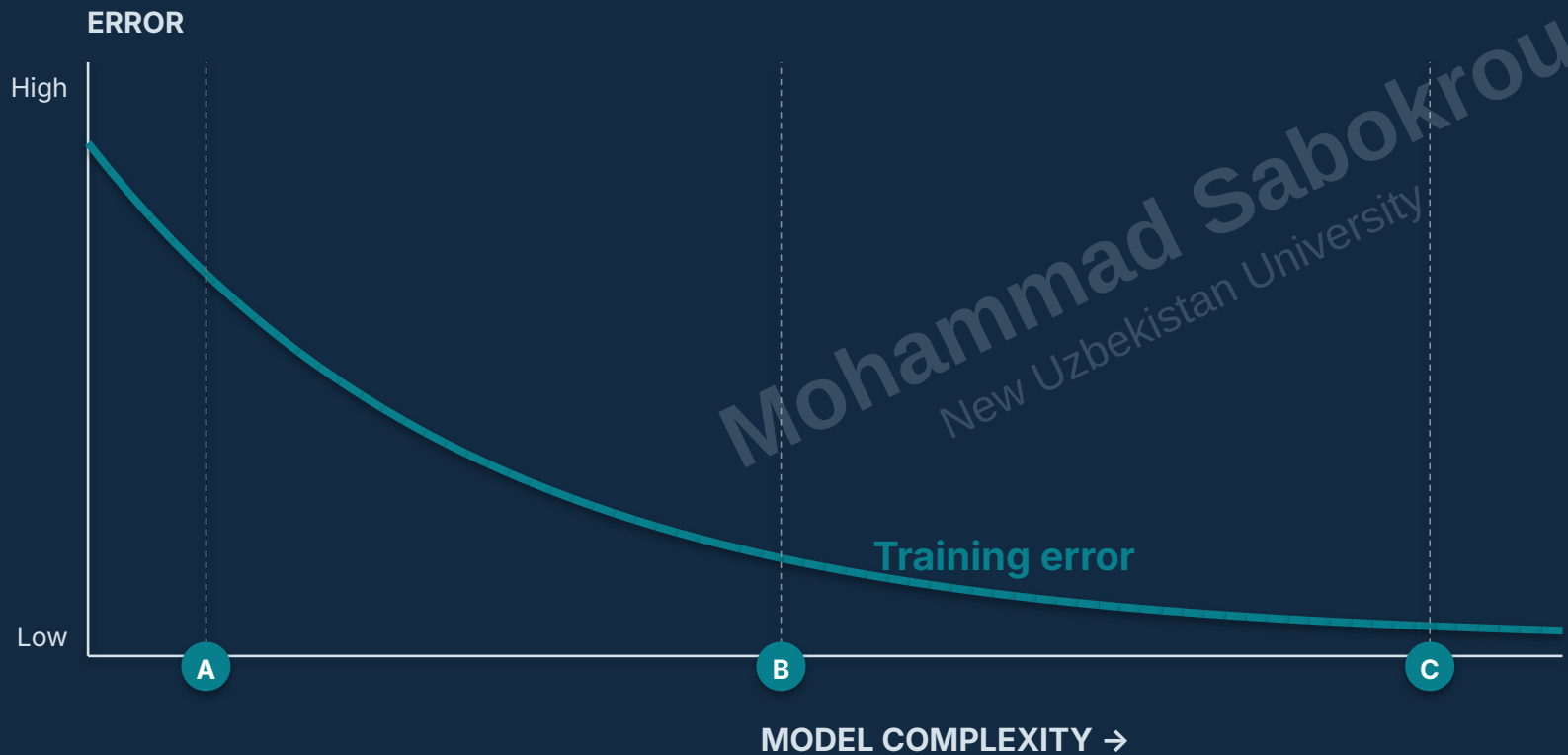
Unseen held-out examples.

"How well might it work
on new cases?"

Low training error alone is not evidence of low future error. The ordering is not a guarantee for every sample.

Question 3: does more complexity always help?

Complexity = how detailed or flexible the prediction rule can be.



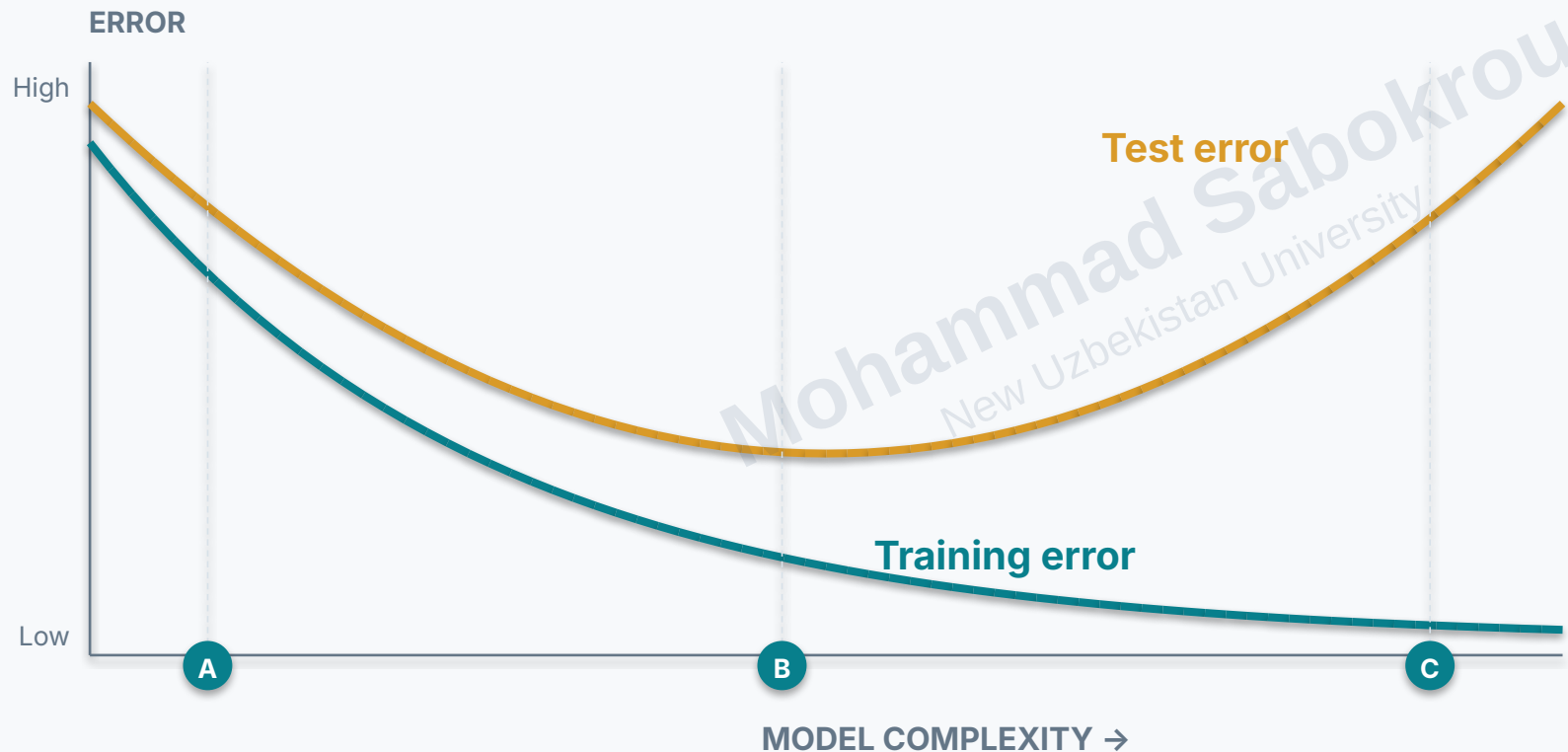
Only the training error is shown.

Must C work best on new students?

Think → vote → justify

Answer 3: no — fitting familiar cases can go too far

A common illustrative pattern: training error falls while test error eventually rises.



A UNDERFITTING

Too simple: both errors high.

B USEFUL BALANCE

Captures useful patterns; lower test error.

C OVERFITTING

Fits training-specific details; worse on new cases.

Validation = separate practice-check data for choosing settings. Keep the final test for one last check.

In the lab: explain the decisions, not only the code

Use this checklist to defend one transformed student record.

MEANING

What do the source, unit and time window mean?

FORMULA

Which calculation or encoding did you apply?

FIT

Which values were learned — and from which rows?

TIMING

Was every input accessible at prediction time?

REUSE

Can the same saved rule handle a future row?

CHECK

Was evaluation kept separate from fitting?

Lab evidence: observed activity, participation, implementation and explanation. No routine lab-email submission.

Trust the representation. Protect the boundary.

Week 4 in three sentences.

- 01 **Choose transformations that match the data and the rule.**
- 02 **Fit on training; use only inputs genuinely available at prediction time.**
- 03 **Check predictions on records that did not influence learning.**

READING

Week 4 notes + ISLP §2.2. Supporting revision: OpenIntro Statistics §§2.1–2.2.

Sources & teaching scope

Instructor reference — not an additional lecture topic.

1 Main text

James, Witten, Hastie, Tibshirani & Taylor (2023). ISLP, §2.2.
statlearning.com

2 Numerical & categorical preprocessing

Official scikit-learn preprocessing and StandardScaler documentation.
scikit-learn.org/stable/modules/preprocessing.html

3 Leakage, fitted workflows & trees

Official scikit-learn Common pitfalls, Pipelines and Decision trees guides.
scikit-learn.org/stable/common_pitfalls.html

4 Supporting review

OpenIntro Statistics, 4th ed., §§2.1–2.2.
openintro.org/book/os/

All classroom records and curves are illustrative; no student outcomes or fitted-model experiment are reported.