Practice Quiz 3: Validation and Classification

MSE 125 — Lectures 6–7

Use this practice quiz to prepare for Quiz 3 (Wednesday, April 22). The real quiz will have 2 questions in 10 minutes, closed-book. This practice set has 11 questions covering Lectures 6–7: the train/test split, overfitting and bias-variance, cross-validation, regularization, distribution shift, logistic regression interpretation, confusion matrices, precision and recall, accuracy under class imbalance, threshold choice via break-even precision, and ROC/AUC with subgroup analysis.

Every concept tested on the real quiz appears somewhere on this practice set, with a different scenario.

Question 1. A student fits a linear regression with **50 features on 100 rows** of data and proudly reports $R^2 = 0.98$ — computed on the same 100 rows used to fit the model.

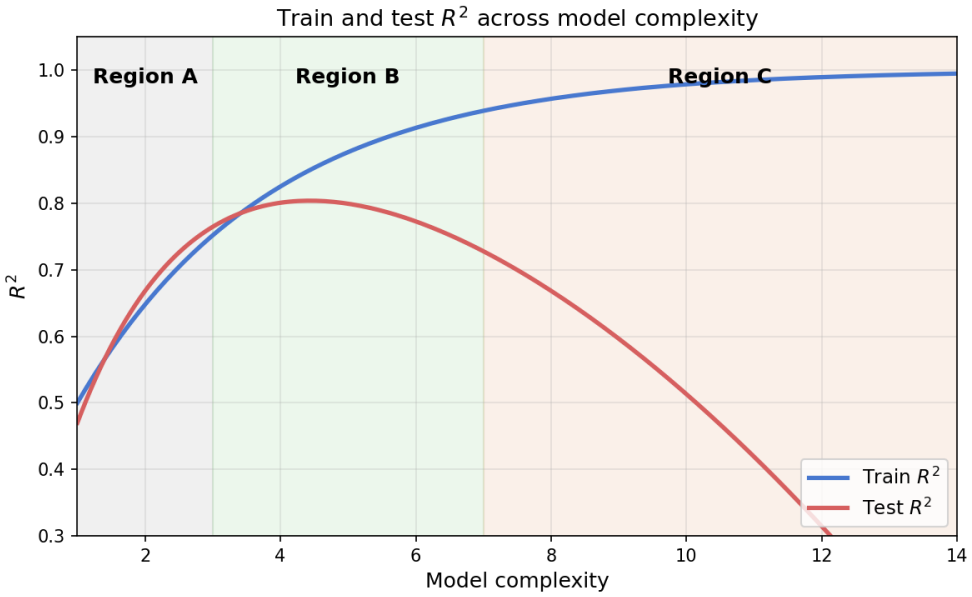
(a) In one or two words, what does this R^2 actually measure?

(b) What single number would you ask for to find out whether the model has learned the pattern or just memorized the data?

Solution

(a) Training R^2 (or equivalently, how well the model *fits* data it has already seen — its "memory" of the training set). Adding features can never make training R^2 go down, so a value near 1 on 50 features over 100 rows is entirely consistent with a model that has just memorized noise.
(b) Ask for the **test** R^2 on data that was held out before the model was fit. If test R^2 is close to the training R^2, the model has learned something generalizable; if it collapses, the 0.98 was memorization.

Question 2. The plot below shows training and test R^2 for models of increasing complexity. Three regions are labeled **A**, **B**, and **C**.



(a) Label each region with one of **underfitting**, **good fit**, **overfitting**.

Region A: _____ Region B: _____ Region C: _____

(b) In region C, is bias^2 (low / high) and is variance (low / high)? What about in region A?

Solution

(a) Region A = **underfitting** (both train and test R^2 are low; the model is too simple). Region B = **good fit** (test R^2 is near its maximum; adding more complexity stops helping). Region C = **overfitting** (train R^2 keeps climbing but test R^2 *falls*, so the extra capacity is fitting noise, not signal).
(b) Region C: **bias^2 low, variance high** — the flexible model can match almost any training shape, so its predictions depend strongly on which particular training sample you drew. Region A: **bias^2 high, variance low** — the model is too rigid to capture the pattern (high bias), but it gives similar predictions regardless of which training sample you hand it (low variance).

Question 3. Fill in the blanks:

> The _____ set is used to choose hyperparameters (such as α for Lasso or polynomial degree). The _____ set is touched exactly once, at the end, to estimate how the final model will perform on new data.

If you tune α directly against the test set instead of a separate validation set, then the test R^2 estimate you report for the final model becomes (circle): **honest / optimistic / pessimistic**.

Solution

Validation set for tuning; **test** set for final evaluation. Tuning against the test set makes the reported test R^2 **optimistic**: you're implicitly picking the setting that happened to look best *on that specific test set*, so the number no longer represents genuinely unseen-data performance. This is why the three-way split exists.

Question 4. You run 5-fold cross-validation on a regression model and get these fold-wise R^2 values:

$$0.71, \quad 0.78, \quad 0.69, \quad 0.74, \quad 0.72$$

Your colleague instead does a single random 80/20 train/test split and reports $R^2 = 0.81$.

(a) Compute the mean 5-fold CV R^2.

(b) Which number is a better estimate of the model's generalization performance, and in one sentence, why?

Solution

(a) Mean CV $R^2 = (0.71 + 0.78 + 0.69 + 0.74 + 0.72)/5 = 3.64/5 = \mathbf{0.728}$.
(b) The **CV mean (0.73)** is the better estimate. A single 80/20 split gives one number that depends heavily on which 20% happened to land in the test set — CV averages over five different held-out folds, so it sees less of the sampling-luck variance and is much less likely to be flattered by an easy test slice. The colleague's 0.81 is most likely a lucky split, not a true gain.

Question 5. Match each goal to the right regularizer, and briefly justify.

Scenario	Lasso or Ridge?
A biologist wants the model to pick **10 genes** out of 20,000 candidates to explain a phenotype.	_____
An economist wants **stable, interpretable coefficients** on a handful of correlated macro indicators (GDP growth, unemployment, inflation).	_____

Why should you **standardize** features to mean 0, standard deviation 1 before fitting either regularizer? Answer in one sentence.

(c) As you *increase* the Ridge penalty α from 0 toward ∞, how does each term of the bias-variance decomposition move? Write $\uparrow$, $\downarrow$, or **unchanged**.

Bias^2: _____ Variance: _____ σ^2: _____

Solution

Biologist: Lasso (L1). Lasso's penalty $\sum_j |\beta_j|$ has corners that push most coefficients exactly to zero, performing automatic feature selection — exactly what "pick 10 out of 20,000" needs.
Economist: Ridge (L2). Ridge's penalty $\sum_j \beta_j^2$ shrinks all coefficients toward zero without forcing any to be exactly zero, which stabilizes coefficients when features are correlated (multicollinearity) instead of arbitrarily picking one and dropping the others.
Standardize first because both penalties act on the raw magnitudes of the coefficients: a feature measured in millimeters will have a much bigger coefficient than the same feature measured in kilometers, so without standardization the penalty hits small-unit features harder than large-unit features, for reasons that have nothing to do with their importance.
(c) As α grows: **Bias**^{2} $\uparrow$, **Variance** $\downarrow$, σ^2 **unchanged**. Ridge shrinks coefficients toward zero, pulling the fitted model away from the data (bias rises) but making predictions less sensitive to which training sample you drew (variance falls). The irreducible noise σ^2 is a property of the data-generating process — no estimator can change it.

Question 6. A bank trains a credit-risk model on 2015–2019 loan data and reports a test R^2 of **0.85** on a held-out 20% of that data. They deploy the model in 2021 (during COVID) and watch performance collapse to $R^2 = 0.30$ within months.

(a) Which type of distribution shift best describes this failure? Circle one: **covariate shift / temporal shift / label shift**.

(b) In one sentence, explain why the random 80/20 test split didn't catch the problem.

Solution

(a) Temporal shift (or equivalently **covariate shift** — both are accepted here). The world itself changed between training and deployment: employment rates, interest rates, default patterns, and the meaning of features like "recent missed payments" all shifted when COVID arrived. "Temporal" emphasizes that time is the driver; "covariate" emphasizes that the input distribution is what changed. Either framing is correct.
(b) A random 80/20 split treats 2015–2019 data as interchangeable, so both train and test are drawn from the *same* pre-COVID world — the split tests whether the model generalizes *within* that era, not whether it generalizes *across* eras. To catch temporal drift you need a temporal split (train on 2015–2018, test on 2019) or explicit backtesting.

Question 7. A logistic regression model predicts the probability of a heart attack within 5 years from patient features. One reported coefficient is

$$\hat{\beta}_{\text{smoker}} = 0.69$$

(a) Interpret this coefficient as an **odds ratio**: smoking multiplies the odds of a heart attack by approximately _____. (Use $e^{0.69} \approx 2$.)

(b) Apply the **divide-by-4 rule** to estimate the maximum change in the *probability* of a heart attack associated with being a smoker: up to _____ percentage points.

Solution

(a) Odds ratio $= e^{0.69} \approx \mathbf{2}$. Smoking roughly **doubles** the odds of a heart attack, holding other features constant.
(b) Divide-by-4 rule: $0.69/4 \approx \mathbf{0.17}$, or about **17 percentage points**. This is the maximum change in probability associated with being a smoker, and it's attained when the baseline probability is near 0.5; at more extreme baseline probabilities the change is smaller.

Question 8. A company deploys an automated resume-screening model. On a balanced evaluation set of **200 candidates**, the confusion matrix is:

	Predicted: hire	**Predicted: reject**
Actually qualified (80)	60	20
Actually unqualified (120)	30	90

Compute each metric. Show your arithmetic.

(a) Accuracy = _____

(b) Precision = _____

(c) Recall = _____

Solution

- **Accuracy** $= (60 + 90)/200 = 150/200 = \mathbf{0.75}$
- **Precision** $= 60/(60 + 30) = 60/90 \approx \mathbf{0.67}$ (of those flagged as hire, how many were actually qualified)
- **Recall** $= 60/(60 + 20) = 60/80 = \mathbf{0.75}$ (of actually-qualified candidates, how many got flagged)

Because the classes here are close to balanced (80 vs 120), accuracy is a *reasonable* headline number — unlike the rare-disease case where the base rate distorts it.

Question 9. A vendor claims their fraud-detection model has **99.2% accuracy** on credit card transactions. The actual base rate of fraud in the data is **0.3%** (about 3 fraudulent transactions per 1,000).

(a) What accuracy would the trivial "always predict not-fraud" baseline achieve? _____

(b) Compared to this baseline, the vendor's model is (circle one): **much better / slightly better / essentially equal / worse**.

(c) The issuer estimates that **missing a fraudulent transaction costs 20× more than flagging a legitimate one**. Which metric should the issuer ask for instead of accuracy, and in one sentence, how does the 20:1 cost ratio justify your choice?

Solution

(a) The baseline hits **99.7%** accuracy — it correctly calls every one of the 997 non-fraud transactions in 1,000 as "not fraud", and simply misses all 3 fraud cases.
(b) Worse. A model that is *less accurate* than the constant-prediction baseline on a hugely imbalanced dataset is giving you no actionable signal.
(c) Ask for **recall** — the fraction of actual fraud caught. With a 20:1 cost asymmetry, each missed fraudulent transaction is worth 20 false alarms, so the issuer should accept many false positives (lower precision) in exchange for catching more real fraud (higher recall). Overall accuracy is dominated by the 99.7% majority class and gives no information about either error type on the minority class.

Question 10. A bank's fraud-detection team has trained a model and tuned it to four candidate operating points on the precision-recall curve. They estimate that a missed fraudulent transaction costs **10× more** than flagging a legitimate one for review.

Point	Recall	Precision
A	0.30	0.90
B	0.60	0.70
C	0.85	0.45
D	0.97	0.10

(a) Compute the **break-even precision** p^* for this cost ratio. _____

(b) Which operating point should the bank choose? Circle one: **A / B / C / D**.

(c) In one sentence, explain why your answer is right — using p^* and the rule that precision is $P(\text{fraud} \mid \text{flagged})$.

Solution

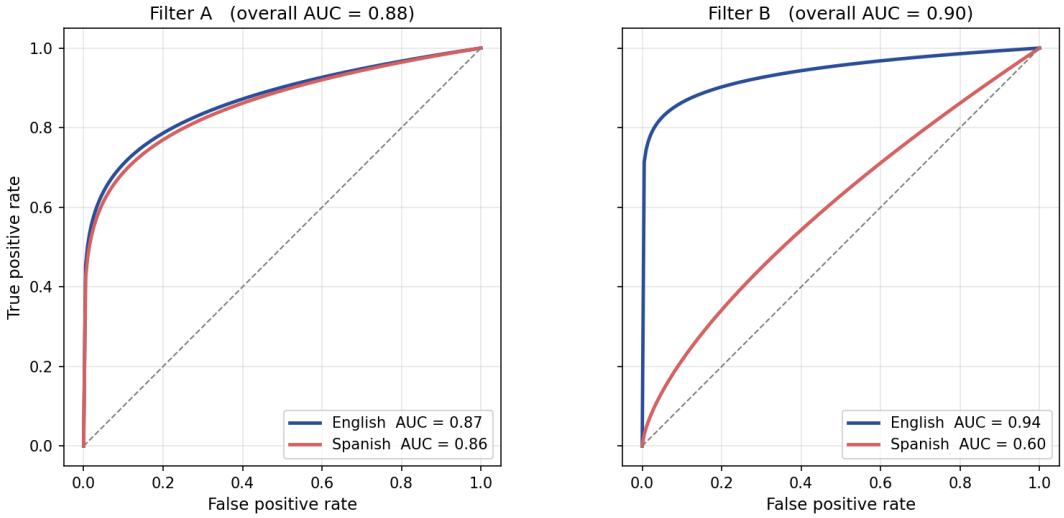
(a) Break-even precision $p^* = 1/(k+1) = 1/(10+1) = \mathbf{1/11} \approx \mathbf{9}\%$. A flag is worth it only when $P(\text{fraud} \mid \text{flagged}) > 9\%$.
(b) Point C. Reasoning: A, B, and C all sit comfortably above $p^* = 9\%$ (precisions 90%, 70%, 45%) — each one is profitable, so we should keep moving down the curve to gain recall. D's precision (10%) sits *right at* break-even, meaning the *marginal* flags added between C and D have precision below 9% — those flags are net-negative. Stop at C, the last point with comfortable margin above p^*.
(c) *Sample answer:* Precision IS $P(\text{fraud} \mid \text{flagged})$, and a flag pays off only when this probability exceeds $p^* \approx 9\%$. C's 45% sits well above p^* so flagging is profitable; D's 10% is right at break-even, meaning the *new* flags between C and D have precision below 9% and lose money on average.
Sanity check by counting (per 1000 transactions, 50 fraudulent):

- A: FN=35, FP$ \$2 \Rightarrow$ cost $\approx 35 \cdot 10 + 2 = 352$
- B: FN=20, FP$ \$13 \Rightarrow$ cost ≈ 213
- **C: FN=7.5, FP$ \$52 $\Rightarrow$ cost ≈ 127 (minimum)**

- D: FN=1.5, FP$ \$437 \Rightarrow$ cost ≈ 452

The break-even rule reaches the same answer without any of this arithmetic.

Question 11. A multilingual email provider is choosing between two spam filters. The ROC curves below show overall and per-language performance.



(a) Which filter would you ship for this provider? Circle: **Filter A / Filter B**

(b) In one sentence, explain why the "overall AUC" comparison would have given you the wrong answer.

Solution

(a) Filter A. Filter B has a slightly higher *overall* AUC (0.90 vs 0.88), but its Spanish AUC is 0.60 — barely better than random guessing — while Filter A treats English and Spanish subscribers roughly equally well (AUCs of 0.87 and 0.86).

(b) Overall AUC is an average weighted by subgroup size, so a model that excels on the majority subgroup can win overall while being **useless for the minority subgroup**. Whenever the product serves multiple populations, subgroup performance is the thing you actually care about; a single aggregate number can hide exactly the failure mode that matters most.