# TMA4268 Statistical Learning V2026

Mock Exam 9 (estimated final exam)

Compiled by Claude for Anders Bekkevard

Based on the Apr 28 exam-review lecture, the 2023–2025 finals, and the prof's stated scope rule

Mock for: May 18, 2026 (real exam date)

**Instructions.**

- **Duration:** 4 hours. **Open book.** Permitted aids: ISLP (2nd ed.), one handwritten A5 sheet of notes, calculator.
- **No code required.** Write answers as math, plain English, or pseudocode. Do *not* memorize R/Python package names.
- **Show your work** — partial credit is generously available, including when calculator slips spoil a numeric answer but the setup is correct.
- **No negative scoring.** Always answer, even when unsure.
- **If a question seems broken or ambiguous**, state the assumption you are making in one short sentence and proceed.
- Total: **100 points = 100 %**. Per-problem weights given in parentheses.

**Grade boundaries (NTNU *prosentvurderingsmetoden*, advisory):** A: 89–100 % B: 77–88 % C: 65–76 % D: 53–64 % E: 41–52 % F: 0–40 %.

---

## Problem 1 (10 %) — Fill-in-the-blank concepts

Read the passage and pick the best word or short phrase for each blank from the choices in parentheses. Each correct fill is worth 1 %.

In supervised learning we are interested in $Y$ given $X$. When the response is continuous we call the task _____ (1) (*regression / classification / clustering / dimensionality reduction*); when our main goal is to understand how $Y$ depends on the components of $X$ (rather than predict $Y$ on new inputs as accurately as possible), we say our goal is _____ (2) (*prediction / inference / compression / regularization*).

Linear discriminant analysis assumes that, within each class, the predictors follow a multivariate normal distribution with the *same* covariance matrix $\mathbf{\Sigma}$ across classes; under this assumption the resulting decision boundary between any two classes is _____ (3) (*quadratic / piecewise constant / linear / cubic*) in $x$. Relaxing the equal-covariance assumption so that each class has its own $\mathbf{\Sigma}_k$ gives _____ (4) (*LDA with shrinkage / logistic regression / naive Bayes / QDA*), whose boundary is then quadratic in $x$.

In the neural-network module the lecturer emphasized that *mini-batch* stochastic gradient descent, in the over-parameterized regime where many fits achieve zero training error, has an

effect that is, in his words, "super weird": it picks (approximately) the solution with minimum _____ (5) (*$L_2$ norm / $L_\infty$ norm / number of nonzero weights / training loss*). This is an example of *implicit* regularization.

For tree ensembles, the lecturer said he prefers _____ (6) (*Gini-based / coefficient-based / permutation-based / p-value-based*) variable importance over the impurity-based variant "because it makes more sense". In random forests, OOB error is, on each individual tree, computed using approximately what fraction of the original observations? _____ (7) (1/2 / 1/3 / 2/3 / $1/e^2$).

In model selection by cross-validation, when several values of a hyperparameter give CV errors within one estimated standard error of the minimum, the lecturer prefers to take the _____ (8) (*most complex / first-fitted / minimum-error / simplest*) such model; this is the *one-standard-error rule*.

In the classification module, the loss function used for AdaBoost (when one recasts it as forward stagewise additive modelling) is the _____ (9) (*squared-error loss / absolute-error loss / exponential loss / 0–1 loss*). XGBoost extends vanilla gradient boosting by, among other things, fitting each tree using _____ (10) (*first-order only / second-order / zeroth-order / stochastic-finite-difference*) gradient information.

## Problem 2 (28 %) — Multiple choice, true/false, and short numeric

For each subproblem, write *True/False* for each statement (or the requested numeric answer). For true/false subproblems you may add a one-sentence justification, but only if you think it helps; do not write essays.

### a) Bias–variance, numerically *(3 %)*

We are studying a regression estimator $\hat{f}$ at a fixed query point $x_0$ where the true function is $f(x_0) = 5$ and the response noise is $\varepsilon \sim (\text{mean } 0, \text{Var}\,\sigma^2 = 0.40)$, independent of the training set. From a large simulation we record:

$$\mathbb{E}\big[\hat{f}(x_0)\big] = 4.6, \qquad \text{Var}\big[\hat{f}(x_0)\big] = 0.30.$$

(i) *(1 %)* Compute $\text{Bias}^2\big[\hat{f}(x_0)\big]$.

(ii) *(1 %)* Compute the expected squared test error $\mathbb{E}[(y_0 - \hat{f}(x_0))^2]$ at $x_0$.

(iii) *(1 %)* A colleague proposes a more flexible estimator $\hat{f}'$ whose simulated values give $\mathbb{E}[\hat{f}'(x_0)] = 5.0$ exactly, but $\text{Var}[\hat{f}'(x_0)] = 0.85$. The training data and the response noise are unchanged. Is the expected squared test error at $x_0$ for $\hat{f}'$ larger, smaller, or equal to that of $\hat{f}$? Justify with one numeric comparison.

### b) Cross-validation: bias, variance, and the one-SE rule *(4 %)*

(i) *(1 %)* True or false: "Leave-one-out cross-validation (LOOCV) has *lower bias* than 5-fold CV as an estimator of the test error of a model trained on the full dataset, because each fold trains on $n - 1$ observations rather than $\frac{4n}{5}$."

(ii) *(1 %)* True or false: "LOOCV typically has *lower variance* than 5- or 10-fold CV because we average over $n$ rather than 5 or 10 per-fold errors."

(iii) *(2 %)* The table below shows mean CV-MSE and the per-fold standard error of the CV-MSE for a ridge-regression hyperparameter $\lambda$, computed with 10-fold CV on the same training set. "Simpler" here means *larger* $\lambda$ (more shrinkage).

| $\lambda$ | $\overline{\text{CV-MSE}}(\lambda)$ | $\widehat{\text{SE}}$ |
|---|---|---|
| 0.10 | 0.485 | 0.020 |
| 0.50 | 0.471 | 0.020 |
| 1.00 | 0.464 | 0.020 |
| 2.00 | **0.458** | **0.022** |
| 5.00 | 0.466 | 0.023 |
| 10.00 | 0.487 | 0.024 |
| 20.00 | 0.532 | 0.026 |

State which $\lambda$ the one-standard-error rule picks. Justify by writing down (a) the value $\text{CV}(\hat{\lambda}) + \widehat{\text{SE}}(\text{CV}(\hat{\lambda}))$, and (b) the set of candidate $\lambda$ values whose CV is at or below that bound.

### c) Backpropagation: what it is and what it is not *(3 %)*

(i) *(1 %)* True or false: "Backpropagation is the parameter-update rule used to train a neural network; without it the network would have no way to descend the loss."

(ii) *(1 %)* True or false: "Backpropagation is, mathematically, an application of the multivariate chain rule organised so that the intermediate quantities computed during the forward pass — pre-activations and activations — are stored and reused on the backward pass."

(iii) *(1 %)* True or false: "Standard backpropagation as taught in this course can be applied to an arbitrary directed graph of computations, including ones with feedback loops between layers (e.g. recurrent neural networks), without modification."

### d) Mini-batch SGD *(3 %)*

(i) *(1 %)* True or false: "Mini-batch SGD with mini-batch size $m \ll N$ produces an estimator of the full-batch gradient that is *unbiased* (its expectation equals the full-batch gradient) but with *higher* variance than the full-batch gradient itself."

(ii) *(1 %)* A neural network is trained on $N = 200{,}000$ examples with mini-batch size $m = 256$. How many parameter updates does the optimizer perform per *epoch* (one full pass through the data)? Give a numeric answer (rounded if needed).

(iii) *(1 %)* True or false: "Common mini-batch sizes $(32, 64, 128, 256, 512)$ are powers of two because of a statistical optimum, not for hardware reasons."

### e) Dropout, label smoothing, and early stopping *(3 %)*

Mark each statement true or false.

(i) Dropout with rate $p = 0.5$ is the standard course recommendation; smaller rates (e.g. $p = 0.2$) are mostly used only in toy examples.

(ii) Label smoothing softens the one-hot target vector toward a small nonzero value $\varepsilon/(C-1)$ on the off-classes; one motivation is the possibility that some training labels are themselves noisy / mislabelled.

(iii) Early stopping returns the model parameters from the epoch *after which* the validation error first stops decreasing (i.e. at the validation minimum), not from the final epoch of training.

### f) XGBoost vs. vanilla gradient boosting *(3 %)*

A junior data scientist on your team is comparing XGBoost to the plain `gbm` implementation. For each statement, mark true or false.

(i) XGBoost uses both the first *and* the second derivative of the loss with respect to the current ensemble's predictions when constructing each new tree, in contrast with vanilla gradient boosting which uses only the first derivative.

(ii) XGBoost adds, on top of the usual learning-rate / shrinkage $\nu$, both an $L_1$ and an $L_2$ penalty on the leaf-output values of each tree, plus a per-leaf complexity penalty $\gamma|T|$ that controls how aggressively trees are pruned.

(iii) The $\nu$ (learning-rate / shrinkage) hyperparameter does not exist in XGBoost — the second-order Newton step already provides automatic step-size selection.

## g) Collinearity in regression output *(3 %)*

A linear regression is fit on $n = 500$ training observations with $p = 8$ predictors. Two of the predictors, $x_4$ and $x_5$, are highly correlated ($\text{cor}(x_4, x_5) \approx 0.98$); the rest are roughly mutually uncorrelated.

(i) *(1 %)* True or false: "Standardizing $x_4$ and $x_5$ to mean 0 and variance 1 before fitting OLS fixes the collinearity problem by making $\mathbf{X}^\top\mathbf{X}$ exactly invertible."

(ii) *(1 %)* True or false: "The estimated coefficients $\hat{\beta}_4$ and $\hat{\beta}_5$ will tend to be *individually insignificant* (large standard errors, $p$-values near 1), even though a joint test of 'at least one of $\beta_4, \beta_5$ is nonzero' may be highly significant."

(iii) *(1 %)* True or false: "Adding many new strongly-correlated predictors of $y$ to the model can only *increase* the multiple $R^2$, but it can decrease the *adjusted* $R^2$."

## h) Logistic regression with an interaction (short numeric) *(3 %)*

A logistic regression is fit on the `default` data with predictors `balance` (in thousands of USD, so 5 corresponds to a balance of \$5,000) and `student` (binary 0/1), plus the interaction `balance:student`. The fitted coefficients are:

$$\hat{\beta}_0 = -10.0, \quad \hat{\beta}_{\texttt{bal}} = 2.5, \quad \hat{\beta}_{\texttt{stu}} = -0.6, \quad \hat{\beta}_{\texttt{bal:stu}} = 0.3.$$

(So the linear predictor is $\hat{\eta} = -10.0 + 2.5 \cdot \texttt{balance} - 0.6 \cdot \texttt{student} + 0.3 \cdot \texttt{balance} \cdot \texttt{student}$.)

(i) *(1 %)* For a *non-student* ($\texttt{student} = 0$), by what factor do the odds of default change when `balance` increases by 1 (i.e. from \$$X$ to \$$X + 1{,}000$)? One numeric value, rounded to two decimals.

(ii) *(1 %)* For a *student* ($\texttt{student} = 1$), by what factor do the odds of default change for the same \$1,000 increase in `balance`? One numeric value, rounded to two decimals.

(iii) *(1 %)* True or false: "Because $\hat{\beta}_{\texttt{stu}} = -0.6 < 0$, we conclude that being a student *decreases* the odds of default by a factor of $e^{-0.6} \approx 0.55$ at every value of `balance`, holding all else equal."

## i) Random forests and OOB *(3 %)*

(i) *(1 %)* True or false: "In a random forest, the parameter `mtry` (number of predictors offered at each split) controls the *correlation* between trees: *smaller* `mtry` gives less correlated trees and therefore more variance reduction from averaging."

(ii) *(1 %)* True or false: "The number of trees $B$ in a random forest is a critical hyperparameter that, like the number of trees $M$ in gradient boosting, must be chosen by cross-validation because $B$ being too large can cause the random forest to overfit."

(iii) *(1 %)* You are running a random forest with $B = 500$ trees on $n = 1{,}000$ observations. Approximately how many of the 1,000 observations would you expect to be *out-of-bag* for any given tree? One numeric answer.

## Problem 3 (16 %) — Theory, hand calculations, pseudocode

### a) The mathy one — LDA decision boundary derivation *(8 %)*

Consider a two-class classification problem with classes $A$ and $B$, predictor vector $X \in \mathbb{R}^2$, equal priors $\pi_A = \pi_B = 1/2$, and class-conditional densities $f_k(x) = \mathcal{N}(x;\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ with a *shared* covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\mu}_A = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{\mu}_B = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

(i) *(2 %)* Starting from Bayes' rule and the multivariate-normal density, derive the form of the *LDA discriminant function*

$$\delta_k(x) \;=\; x^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \;-\; \tfrac{1}{2}\, \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \;+\; \log \pi_k$$

by taking the logarithm of $\pi_k f_k(x)$ and *discarding any terms that do not depend on the class label $k$*. State explicitly which terms you discard, and why doing so is legitimate when comparing $\delta_A(x)$ to $\delta_B(x)$.

(ii) *(3 %)* Set $\delta_A(x) = \delta_B(x)$ and use the explicit values of $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B, \boldsymbol{\Sigma}, \pi_A, \pi_B$ above to derive the *explicit equation of the LDA decision boundary* between classes $A$ and $B$ in the $(x_1, x_2)$ plane. Simplify to the form $a\, x_1 + b\, x_2 = c$ with concrete numerical values of $a, b, c$.

(iii) *(1 %)* A classmate asks: "Where exactly did the *quadratic* part of $\delta_k(x)$ go?" Identify the quadratic-in-$x$ term in $\log(\pi_k f_k(x))$, and explain in one short sentence why it cancels in $\delta_A - \delta_B$ for LDA but would *not* cancel if we relaxed the equal-covariance assumption ($\boldsymbol{\Sigma}_A \neq \boldsymbol{\Sigma}_B$), giving QDA.

(iv) *(2 %)* Now consider the same problem but with $\pi_A = 0.8$ and $\pi_B = 0.2$ (the prior probability of class $A$ is much higher), keeping $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B, \boldsymbol{\Sigma}$ as above. Write down the new decision boundary in the form $a\, x_1 + b\, x_2 = c'$, giving a numeric value of $c'$ to three decimals (you may use $\log 4 \approx 1.386$). In one short sentence, explain in plain English how the boundary has shifted relative to part (ii), and which class now occupies the larger region of $\mathbb{R}^2$.

### b) Pseudocode — mini-batch SGD training of a feedforward neural network *(5 %)*

You are training a one-hidden-layer feedforward neural network with $p$ inputs, $M$ hidden units (ReLU activation), and a single scalar regression output. The loss is squared error per example, $\ell_i = \frac{1}{2}\big(y_i - \hat{y}_i\big)^2$, and you want to use *mini-batch* stochastic gradient descent with batch size $m$, learning rate $\eta$, for $E$ epochs over a training set $\{(x_i, y_i)\}_{i=1}^N$.

(i) *(4 %)* Write pseudocode (math, plain English, or imperative pseudocode — whatever is clearest for you) for one full training run. Your pseudocode should make explicit:

- the initialization of the parameter vector $\boldsymbol{\theta}$ (the network's weights and biases);
- the outer loop over epochs $e = 1, \ldots, E$;
- the random partitioning of the training set into mini-batches of size $m$ within each epoch;
- one *forward pass* per example in the mini-batch, storing the intermediate quantities you will need for the backward pass;

- the *backward pass* (backpropagation), which uses the chain rule to compute $\nabla_{\boldsymbol{\theta}} \ell_i$ for each example in the mini-batch (you may write this step as a single line "compute $\nabla_{\boldsymbol{\theta}} \ell_i$ via backprop using the stored intermediates"; you do not need to expand the chain rule explicitly here);
- the SGD update rule $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \widehat{\nabla_{\boldsymbol{\theta}} L}$, where $\widehat{\nabla_{\boldsymbol{\theta}} L}$ is the mini-batch-average gradient.

Add one or two lines of text *outside* the pseudocode where you indicate that $\widehat{\nabla_{\boldsymbol{\theta}} L}$ is an *unbiased* estimator of the full-batch gradient.

(ii) *(1 %)* In one short sentence, state the lecturer's "headline NN fact" connecting mini-batch SGD to *implicit* $L_2$ regularization in the over-parameterized regime where many parameter vectors interpolate the training data exactly.

### c) Bootstrap by hand — the OOB probability *(3 %)*

You have a training set of $n$ observations and you draw a single bootstrap sample of size $n$ *with replacement* from it.

(i) *(1 %)* Write $\Pr(\text{observation } i \notin \text{this bootstrap sample})$ as a function of $n$, and then evaluate it numerically for $n = 5$ (give a decimal value to three places).

(ii) *(1 %)* Show that as $n \to \infty$ this probability tends to $1/e \approx 0.368$, and conclude that the probability that observation $i$ *is* in the bootstrap sample tends to $1 - 1/e \approx 0.632$.

(iii) *(1 %)* In two short sentences, explain how this result justifies the use of *out-of-bag* (OOB) error in random forests and bagging as a no-extra-cost estimate of test error: roughly what fraction of trees can each observation be evaluated on as a held-out test point, and why does that make OOB error a CV-equivalent quantity?

## Problem 4 (22 %) — Data analysis: fuel-economy regression

A team of automotive researchers has a sample of $n = 392$ car models from the 1970s and 1980s. The response variable is `mpg` (miles per gallon, continuous). The available predictors are

- `weight` (continuous, lbs, range $\sim$ 1,600–5,100);
- `displacement` (continuous, cubic inches, range $\sim$ 68–455);
- `horsepower` (continuous, hp, range $\sim$ 46–230);
- `year` (continuous, 70–82);
- `origin` — categorical, 3 levels: `American` (reference), `European`, `Japanese`.

**Important pairwise correlations among the continuous predictors (training set, after standardization):**

$$\text{cor}(\texttt{weight}, \texttt{displacement}) \approx 0.93, \qquad \text{cor}(\texttt{weight}, \texttt{horsepower}) \approx 0.86.$$

The data are split 292/100 into training and test sets. All continuous predictors are standardized to mean 0, variance 1 before fitting any of the models below.

### a) OLS with a B-spline term and a categorical predictor *(8 %)*

The course staff first fits, on the training set, the OLS model

$$\texttt{mpg} \sim \text{bs}(\texttt{weight}, \text{df} = 4) + \texttt{displacement} + \texttt{horsepower} + \texttt{year} + \texttt{origin}.$$

Here $\text{bs}(\texttt{weight}, \text{df} = 4)$ is a cubic B-spline basis on `weight` with 4 basis functions (so 3 interior knots and no separate intercept — the basis spans the same space as a cubic polynomial in `weight` plus three additional "hinge"-style basis functions, and the overall constant is absorbed into the intercept). The fitted output is:

| | **Estimate** | **Std. Error** | **t-value** | **Pr($>\lvert t\rvert$)** |
|---|---|---|---|---|
| (Intercept) | 29.40 | 0.230 | 127.8 | $< 0.001$ |
| bs(`weight`)$_1$ | $-2.20$ | 0.480 | $-4.58$ | $< 0.001$ |
| bs(`weight`)$_2$ | $-5.10$ | 0.620 | $-8.23$ | $< 0.001$ |
| bs(`weight`)$_3$ | $-3.60$ | 0.700 | $-5.14$ | $< 0.001$ |
| bs(`weight`)$_4$ | $-1.40$ | 0.640 | $-2.19$ | 0.029 |
| `displacement` | $-0.130$ | 0.380 | $-0.34$ | 0.732 |
| `horsepower` | $-0.620$ | 0.310 | $-2.00$ | 0.046 |
| `year` | 2.85 | 0.190 | 15.0 | $< 0.001$ |
| `origin_European` | 1.95 | 0.420 | 4.64 | $< 0.001$ |
| `origin_Japanese` | 2.30 | 0.380 | 6.05 | $< 0.001$ |

Residual standard error: 2.85 on 282 degrees of freedom. Multiple $R^2 = 0.842$, Adjusted $R^2 = 0.837$. Training MSE = 8.13, Test MSE = 8.95.

(i) *(1 %)* How many parameters (including the intercept) does this model estimate? Verify the count against the printed residual degrees of freedom ($292 - \text{df} = 282$).

(ii) *(2 %)* The categorical `origin` variable has 3 levels (`American`, `European`, `Japanese`). The fit uses 2 dummy variables, not 3. State (a) which level is the reference, (b) what the intercept estimates given that all standardized continuous predictors are at 0 and all B-spline basis values are at 0, and (c) why using 3 dummies plus an intercept would make the model unidentifiable.

(iii) *(2 %)* For a Japanese car (`origin_Japanese` = 1) in the year 1976 whose *standardized* continuous predictors are all 0 (so it is an "average" car in terms of `weight, displacement, horsepower`) and whose standardized `year` is exactly 0 as well, compute the predicted `mpg`. Show the linear-combination step.

(iv) *(2 %)* The estimates for $\text{bs}(\texttt{weight})_1, \ldots, \text{bs}(\texttt{weight})_4$ are all negative and individually significant, but the B-spline basis values themselves are *not* the predicted `mpg` of any particular car. Briefly explain in one short sentence what these four coefficients *do* mean in terms of the fitted curve, and state which testable claim about the response of `mpg` to `weight` would be more naturally expressed as a *joint* test of all four B-spline coefficients than as four separate $t$-tests.

(v) *(1 %)* A classmate writes: "`displacement` has $p = 0.732$, so it has no effect on `mpg`, and we should drop it from the model." Rebut this reading in one short sentence by appealing to a specific feature of *this* table.

### b) Diagnosing collinearity, and a fix *(4 %)*

(i) *(2 %)* Identify the two symptoms in the fitted output above that, taken *together*, are consistent with `displacement` being collinear with `weight` (in the sense $\text{cor}(\texttt{weight}, \texttt{displacement}) = 0.93$). Be specific: name the rows and the columns of the output, and explain in one or two sentences why each is what we would expect when collinearity inflates the variance of one coefficient via $\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$.

(ii) *(1 %)* Refitting the OLS model after dropping `displacement` alone (keeping the B-spline on `weight`, plus `horsepower, year, origin`) gives training MSE = 8.14 and test MSE = 8.92, essentially unchanged from part (a). In one short sentence, what does this near-identity tell us about whether the collinearity in part (a) was hurting *prediction* (as opposed to *interpretation*)?

(iii) *(1 %)* The course staff lists two additional principled fixes for collinearity that do *not* require dropping a predictor: (a) adding an $L_2$ (ridge) penalty $\lambda \sum_j \beta_j^2$, and (b) replacing the correlated predictors with their first few principal components, i.e. *principal-component regression* (see part (c)). In one short sentence, state *when* you would prefer (b) over (a) (the slides phrase this as "a question of *compression vs. shrinkage*").

### c) Principal-component regression with cross-validation *(6 %)*

The course staff now applies principal-component regression (PCR) to the four continuous predictors {`weight`, `displacement`, `horsepower`, `year`} (still standardized), and then regresses `mpg` on the first $M$ principal components plus the two `origin` dummies (which are *not* entered into the PCA). The CV-MSE as a function of $M$ is shown below, with the per-$M$ standard errors. Smaller $M$ corresponds to a simpler model.

| $M$ | $\overline{\text{CV-MSE}}(M)$ | $\widehat{\text{SE}}$ |
|---|---|---|
| 1 | 11.40 | 0.90 |
| 2 | 9.40 | 0.85 |
| 3 | 8.55 | 0.80 |
| 4 | 8.51 | 0.80 |

The estimated loading vector of the first principal component, $\phi_1 = (\phi_{1,\texttt{weight}}, \phi_{1,\texttt{disp}}, \phi_{1,\texttt{hp}}, \phi_{1,\texttt{year}})^\top$, has entries (rounded)

$$\phi_1 \approx (0.56,\ 0.58,\ 0.55,\ -0.20)^\top, \qquad \|\phi_1\|_2 = 1.$$

(i) *(2 %)* Which $M$ does the *one-standard-error rule* pick? Justify by computing $\mathrm{CV}(\hat{M}) + \widehat{\mathrm{SE}}(\mathrm{CV}(\hat{M}))$ for the CV minimum, and listing all $M$ values whose CV-MSE is at or below that bound. (Recall "simpler" means *smaller* $M$ here.)

(ii) *(2 %)* The first three components of $\phi_1$ are all roughly $+0.55$ and positive; the fourth (loading on `year`) is $-0.20$. In one or two sentences, interpret this loading vector in plain English: what underlying property of the car does PC1 appear to be capturing, and why does it make sense, given the correlation structure stated above, that the loadings on `weight, displacement, horsepower` are similar and large?

(iii) *(2 %)* A classmate proposes choosing $M$ by minimising the *training* MSE of the PCR fit instead of the CV-MSE. Briefly explain (a) why this choice is essentially *guaranteed* to pick $M = 4$ (the maximum) rather than a smaller value, and (b) in one sentence, what is wrong with picking $M = 4$ in this dataset, given parts (a)–(b) above.

### d) GAM with smoothing splines for comparison *(4 %)*

The course staff also fits a generalized additive model

$$\texttt{mpg} \;=\; \beta_0 + f_1(\texttt{weight}) + f_2(\texttt{horsepower}) + f_3(\texttt{year}) + \boldsymbol{\beta}_{\texttt{origin}}^{\top}\texttt{origin} + \varepsilon,$$

where each $f_j$ is a smoothing spline whose smoothness is chosen by leave-one-out CV. The fitted effective degrees of freedom for the three smooth terms, reported by the GAM summary, are

$$\widehat{\mathrm{edf}}(f_1) = 6.4, \quad \widehat{\mathrm{edf}}(f_2) = 3.2, \quad \widehat{\mathrm{edf}}(f_3) = 1.0.$$

Test MSE for this GAM is 7.95.

(i) *(1 %)* The smoothing parameter $\lambda_j$ of the smoothing spline $f_j$ is connected to the *effective degrees of freedom* via $\widehat{\mathrm{edf}}(f_j) = \mathrm{tr}(\mathbf{S}_{\lambda_j})$, where $\mathbf{S}_{\lambda_j}$ is the smoother matrix mapping $\mathbf{y}$ to the fitted vector. State (a) the qualitative direction $\widehat{\mathrm{edf}} \to \lambda$ (smaller edf $\Leftrightarrow$ larger or smaller $\lambda$?), and (b) what $\widehat{\mathrm{edf}}(f_3) = 1.0$ implies about the fitted shape of $f_3$.

(ii) *(1 %)* The OLS fit from part (a) achieved test MSE 8.95; ridge (not shown) achieves 8.60; this GAM achieves 7.95; a gradient-boosted tree ensemble (depth 3, $\nu = 0.05$, $M$ chosen by 10-fold CV at $M^\star \approx 2000$) achieves test MSE 6.10. In one short sentence, what is the most natural conclusion about the structure of the `mpg` vs. predictors relationship?

(iii) *(1 %)* A junior colleague writes: "Boosting got the lowest test MSE, so we should always use boosting." Briefly contradict in one short sentence, mentioning what we are losing by switching from the GAM to the boosted tree ensemble.

(iv) *(1 %)* A separate junior colleague writes: "The gradient-boosted tree achieved test MSE 6.10 with $M^\star = 2000$; let's run it again with $M = 20{,}000$ to push the test MSE even lower." Briefly explain in one short sentence why this is risky for boosting (in contrast with the analogous proposal for a random forest).

## Problem 5 (24 %) — Data analysis: bank loan default classification

A retail bank wants a model for predicting whether a personal-loan applicant will default on the loan within the loan's first year. The training data are $n = 4{,}000$ historical applicants with binary response `default` (1 if defaulted, 0 otherwise), and a held-out test set of $n_{\text{test}} = 1{,}500$ applicants of whom 300 (20%) defaulted. The predictors are:

- `fico` — continuous, FICO credit score, range $\sim$ 540–820, mean $\sim$ 690;
- `debt_to_income` — continuous, ratio of monthly debt payments to monthly income, range $\sim$ 0.05–0.55;
- `loan_amount` — continuous, in thousands of USD;
- `employment_years` — continuous, years at current employer;
- `home_ownership` — categorical 3 levels: `rent` (reference), `mortgage`, `own`.

All continuous predictors are standardized to mean 0, variance 1 before fitting.

### a) Logistic regression — main effects only, odds and a categorical contrast *(7 %)*

A logistic regression with only *main effects* (no interaction) is fit on the training set. The fitted coefficients are

| | **Estimate** | **Std. Error** | **z-value** | **Pr($>\|z\|$)** |
|---|---|---|---|---|
| (Intercept) | $-1.80$ | 0.10 | $-18.0$ | $< 0.001$ |
| `fico` | $-1.20$ | 0.08 | $-15.0$ | $< 0.001$ |
| `debt_to_income` | 0.85 | 0.07 | 12.1 | $< 0.001$ |
| `loan_amount` | 0.40 | 0.08 | 5.0 | $< 0.001$ |
| `employment_years` | $-0.20$ | 0.08 | $-2.5$ | 0.012 |
| `home_ownership_mortgage` | $-0.50$ | 0.11 | $-4.55$ | $< 0.001$ |
| `home_ownership_own` | $-0.90$ | 0.18 | $-5.00$ | $< 0.001$ |

(i) *(2 %)* For an otherwise-identical pair of applicants, by what factor do the odds of default change for each +1 increase in *standardized* `fico`? Comment in one sentence on the sign: does this match what you would have predicted economically?

(ii) *(2 %)* Consider a specific applicant with *standardized* predictor values

`fico` $= -1$, `debt_to_income` $= +1$, `loan_amount` $= 0$, `employment_years` $= 0$, `home_ownership`

Compute (a) the linear predictor $\hat{\eta}$ and (b) the predicted probability $\hat{p}$ of default. Round $\hat{p}$ to three decimals.

(iii) *(2 %)* Consider two applicants who are *identical* on every standardized continuous predictor (`fico, debt_to_income, loan_amount, employment_years` all at 0), but one of them `rent`s their home and the other `own`s their home outright. By what factor are the odds of default of the home-owner smaller than those of the renter? One numeric value, rounded to two decimals. Then state in one sentence what this factor would equal on the *probability* scale (i.e. if the renter has predicted probability $\hat{p}$, is the owner's predicted probability also $0.41\hat{p}$, or something else?).

(iv) *(1 %)* A classmate writes: "`home_ownership_own` has $p < 0.001$, so we can conclude that *owning* a home *causes* the probability of default to drop." Rebut this reading in one short sentence by appealing to the kind of model the prof flagged repeatedly — "fancy correlations, not causal".

### b) LDA vs. QDA — explicit discriminant computation *(6 %)*

A second analyst fits LDA and QDA on the same training data, using only the two continuous predictors `fico` and `debt_to_income` (both standardized). The estimated class-conditional means and covariance(s), and the estimated class priors, are:

$$\hat{\boldsymbol{\mu}}_0 = \begin{pmatrix} +0.20 \\ -0.30 \end{pmatrix}, \quad \hat{\boldsymbol{\mu}}_1 = \begin{pmatrix} -0.80 \\ +1.20 \end{pmatrix},$$

$$\hat{\boldsymbol{\Sigma}}^{\text{LDA pooled}} = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}, \qquad \hat{\pi}_0 = 0.80,\ \hat{\pi}_1 = 0.20.$$

For QDA only, the class-specific covariances are

$$\hat{\boldsymbol{\Sigma}}_0 = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}, \qquad \hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 1.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}.$$

(All other parameters are the same as LDA.)

(i) *(2 %)* Using the pooled $\hat{\boldsymbol{\Sigma}}$ above and the LDA discriminant function $\delta_k(x) = x^\top \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_k - \frac{1}{2}\hat{\boldsymbol{\mu}}_k^\top \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_k + \log \hat{\pi}_k$, classify the test applicant with standardized predictors

$$x_0 = (\,\texttt{fico} = -0.5,\ \texttt{debt\_to\_income} = +0.8\,)^\top$$

by computing $\delta_0(x_0)$ and $\delta_1(x_0)$ to three decimals and choosing the larger.

(ii) *(2 %)* Now classify the same $x_0$ under *QDA*, using the class-specific $\hat{\boldsymbol{\Sigma}}_k$ above. Recall that for QDA the discriminant is

$$\delta_k^{\text{QDA}}(x) \;=\; -\tfrac{1}{2}\log|\hat{\boldsymbol{\Sigma}}_k| \;-\; \tfrac{1}{2}(x-\hat{\boldsymbol{\mu}}_k)^\top \hat{\boldsymbol{\Sigma}}_k^{-1}(x-\hat{\boldsymbol{\mu}}_k) \;+\; \log\hat{\pi}_k.$$

Compute $\delta_0^{\text{QDA}}(x_0)$ and $\delta_1^{\text{QDA}}(x_0)$ to three decimals and report the predicted class. (You may use $\log 1 = 0$, $\log 1.5 \approx 0.405$, $\log 0.5 \approx -0.693$.)

(iii) *(1 %)* In one short sentence, identify the structural reason the LDA boundary between class 0 and class 1 is a *line* in the $(x_1, x_2)$-plane but the QDA boundary is a *conic* (here, with $\hat{\boldsymbol{\Sigma}}_0$ and $\hat{\boldsymbol{\Sigma}}_1$ both diagonal but unequal, a tilted ellipse or a hyperbola).

(iv) *(1 %)* The decision boundary for LDA between the two classes can be derived from $\delta_0 = \delta_1$. Without re-deriving the full boundary, state in one sentence how the boundary would *shift* (which direction, toward which class's mean) if the prior $\hat{\pi}_0$ were changed from 0.80 to 0.50, all other parameters held fixed. (You may simply state, e.g., "shifts $\Delta$ units toward class $X$".)

### c) XGBoost: the regularization knobs *(8 %)*

The risk team next tries an XGBoost classifier on the same training data, including all 5 predictors. Cross-validating with 5 folds, they sweep a coarse grid of hyperparameters and find a CV-best configuration with

$$M^\star \approx 700 \text{ trees}, \quad \eta = 0.05, \quad \texttt{max\_depth} = 3, \quad \texttt{min\_child\_weight} = 5,$$

plus, importantly, the explicit XGBoost regularization parameters

$\lambda = 1.0$ ($L_2$ on leaf weights), $\alpha = 0.0$ ($L_1$ on leaf weights), $\gamma = 0.5$ (per-leaf complexity penalty /

Plain `gbm` with the same depth and learning rate (and *no* $\lambda, \alpha, \gamma$ controls) gives a CV-AUC slightly below XGBoost.

(i) *(2 %)* In one or two short sentences, state *which two derivatives of the loss with respect to the current ensemble's predictions* XGBoost uses (vanilla GBM uses one of these, XGBoost uses both), and informally why the second one allows XGBoost to take more accurate per-tree steps than vanilla GBM.

(ii) *(3 %)* XGBoost adds three regularizers beyond what plain GBM has: an $L_2$ penalty $\lambda$ on leaf weights, an $L_1$ penalty $\alpha$ on leaf weights, and a per-leaf complexity penalty $\gamma|T|$ (where $|T|$ is the number of leaves in the tree). State, in *one short sentence each*, what behavioural effect each of these three parameters has on the resulting tree ensemble. (You may explicitly draw the parallel between the $L_2$ on leaf weights and ridge regression on linear-regression coefficients, and between $\gamma|T|$ and cost-complexity pruning.)

(iii) *(2 %)* A junior colleague proposes *both* (a) doubling $M$ from 700 to 1,400, *and* (b) doubling the learning rate from $\eta = 0.05$ to $\eta = 0.10$. They argue that these changes will simply "go twice as fast in the right direction" and roughly preserve the bias–variance balance of the current ensemble. State briefly (one short sentence each) (a) what is wrong with the "preserve the bias–variance balance" claim, given the coupling between $M$ and $\eta$ in boosting, and (b) what diagnostic plot you would actually use to pick $M$ once $\eta$ is fixed.

(iv) *(1 %)* True or false: "The L1 penalty $\alpha$ on leaf weights in XGBoost, by analogy with the lasso in linear regression, can drive some leaf-output values *exactly* to zero, effectively pruning small leaves at training time — and this is on top of the explicit $\gamma|T|$ leaf penalty."

### d) Sensitivity, specificity, ROC, and class imbalance *(3 %)*

The three classifiers (logistic, LDA, XGBoost) are applied to the 1,500 test applicants. Their test-set confusion matrices at the default threshold of $\hat{p} = 0.5$ are summarized below (recall: 300 true defaulters, 1,200 non-defaulters in the test set):

| | **Sensitivity (recall)** | **Specificity** | **Accuracy** |
|---|---|---|---|
| Logistic (main effects) | 0.40 | 0.95 | 0.84 |
| LDA | 0.36 | 0.97 | 0.85 |
| XGBoost (CV-best) | 0.55 | 0.93 | 0.85 |

The XGBoost ROC curve over the full threshold sweep achieves AUC = 0.88, vs. 0.83 for logistic and 0.81 for LDA.

(i) *(1 %)* A trivial classifier that always predicts "no default" has what accuracy on this test set? One numeric value. Use this to comment in one short sentence on why *accuracy* alone is a poor metric for choosing among the three rows above.

(ii) *(2 %)* The risk team's stated objective is to maximize sensitivity (catching as many true future defaulters as possible) subject to keeping specificity at or above 0.90 on the test set. Among the three classifiers above, which would you recommend for deployment at *the default threshold*? Then state, in one short sentence, how you would adjust the chosen classifier's *threshold* below $\hat{p} = 0.5$ if the team also wanted to raise sensitivity further, and which other quantity in the table you would expect to move (and in which direction) as a consequence.

---

**End of exam.** Total: $10 + 28 + 16 + 22 + 24 = 100$ points.