TMA4268 V2026 Mock Exam 3 — Solution Proposal

Compiled for Anders Bekkevard

Companion to `mock-exam-3.tex` (same directory).

Mock for: May 18, 2026

This document is a worked-solution proposal in the style of the official Stefanie/Sara solutions to the 2024 and 2025 finals. Point values are quoted in the heading of each solved sub-part. Partial-credit hints are inline. Refer back to `mock-exam-3.tex` for the problem statements.

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

Solution *(1 P per blank.)*

(1) **parametric**

(2) **nonparametric**

(3) **irreducible**

(4) **shrinkage**

(5) **lasso**

(6) **k-fold cross-validation**

(7) **distributional assumptions**

(8) **generalized additive models**

(9) **principal component analysis**

(10) **dendrogram**

Grading: 1 P per correct blank. No partial credit for "close" alternatives. For (5), "ridge" is a hard zero — ridge shrinks but does not zero out (the prof flagged this as a classic exam trap). For (7), "cross-validation" loops back on itself and is wrong; "shrinkage" / "regularization" are about lasso/ridge, not AIC.

Problem 2 (28 %) — Multiple choice, T/F, short numeric

a) Bias–variance and double descent *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. The classical bias–variance trade-off: more flexible classes capture more of f (lower squared bias) but track training noise harder (higher variance).

(ii) **True**. The decomposition $\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Bias}^2 + \text{Var} + \sigma^2$ is an algebraic identity (the cross-term vanishes by independence of ε and the training set). It holds for *any* estimator $\hat{f}$, including over-parameterized ones.

(iii) **False**. This is the *double-descent / benign-overfitting* regime: when $p \gg n$, an interpolating model can still generalize well, and indeed empirically the test error *decreases* again past the interpolation threshold rather than growing without bound. The optimization implicitly picks the minimum-norm interpolator from the infinite family of zero-training-error solutions. (Prof's hobbyhorse, hammered five times across the course.)

(iv) **False**. With large σ^2 a flexible model spends extra variance "chasing noise" that is irreducible by definition, so test error *rises*. Less-flexible models win when noise dominates.

b) Cross-validation variants *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. Each LOOCV training set has $n-1$ points (low bias as an estimator of test error trained on n), but the n training sets overlap almost completely, so the n fold errors are highly correlated and their average has higher variance than the 5-fold average.

(ii) **False**. The validation-set approach uses a *single* train/test split. Two-fold CV would average over both possible splits ($\mathcal{F}_1$ as training and as test); the validation-set approach does not.

(iii) **True**. The PRESS shortcut for OLS: LOOCV $= \frac{1}{n}\sum_i \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2$, where h_{ii} are the leverages from the hat matrix $H = X(X^\top X)^{-1}X^\top$. No re-fitting required.

(iv) **False**. This is the canonical CV pitfall. Filtering predictors using y *before* any CV leaks information from the held-out folds into the variable-selection step, which biases the CV estimate of test error *downward*. The selection step must be inside every CV iteration.

c) Neural network parameters and forward pass *(4 %)*

Solution *(2 %)* **(i)** Each layer contributes $(\#\text{inputs}+1)\cdot(\#\text{outputs})$ parameters; the $+1$ counts the bias.

$$\begin{aligned} \text{input} \to \text{hidden}_1: &\quad (6+1)\cdot 8 = 56 \\ \text{hidden}_1 \to \text{hidden}_2: &\quad (8+1)\cdot 4 = 36 \\ \text{hidden}_2 \to \text{output}: &\quad (4+1)\cdot 2 = 10 \\ \text{Total:} &\quad \boxed{56+36+10 = 102} \text{ parameters.} \end{aligned}$$

Grading: 1 P for the layer-by-layer breakdown, 1 P for the correct total. Deduct 0.5 if biases are forgotten (common slip $\Rightarrow 6\cdot 8+8\cdot 4+4\cdot 2 = 88$). The softmax on the output adds no parameters — it is a deterministic normalization.

Solution *(2 %)* **(ii)** Pre-activation:

$$\begin{aligned} z &= b + \sum_{j=1}^{6} w_j x_j \\ &= -0.1 + 0.5\cdot 1 + (-1.0)\cdot 0 + 0.2\cdot 2 + 1.0\cdot(-1) + (-0.5)\cdot 1 + 0.3\cdot 3 \\ &= -0.1 + 0.5 + 0 + 0.4 - 1.0 - 0.5 + 0.9 \\ &= 0.2. \end{aligned}$$

Sigmoid:

$$\sigma(0.2) = \frac{1}{1+e^{-0.2}} = \frac{1}{1+0.8187} = \frac{1}{1.8187} \approx \boxed{0.550}.$$

Grading: 1 P for the correct pre-activation $z = 0.2$, 1 P for the sigmoid step. Accept any answer in $[0.549, 0.551]$. If the question were asked with ReLU instead, the answer would be $\max(0, 0.2) = 0.2$.

d) Odds, log-odds, interaction *(3 %)*

Solution *(1 %)* **(i)** $p = 0.05 \Rightarrow \text{odds} = \dfrac{p}{1-p} = \dfrac{0.05}{0.95} = \dfrac{1}{19} \approx \boxed{0.0526}$.

Solution *(1 %)* **(ii)** $\text{odds} = 4 \Rightarrow p = \dfrac{\text{odds}}{1+\text{odds}} = \dfrac{4}{5} = \boxed{0.80}$.

Solution *(1 %)* **(iii)** For a *smoker*, a unit increase in `bmi` changes the log-odds by $\hat{\beta}_{\texttt{bmi}} + \hat{\beta}_{\texttt{bmi:smokerYes}} = 0.02 + 0.10 = 0.12$, so the odds multiply by

$$e^{0.12} \approx \boxed{1.13}.$$

Grading: full credit for the smoker-specific factor $e^{0.12} \approx 1.13$. Half credit for $e^{0.02} \approx 1.02$ (ignoring the interaction — the standard trap).

e) Splines and degrees of freedom *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. A cubic spline with K interior knots has $K + 4$ basis functions (the cubic part contributes 4, and each knot adds one truncated-power basis); equivalently $K + 3$ basis functions plus the global intercept.

(ii) **True**. The natural cubic spline imposes *four* extra linearity constraints (the second and third derivatives must vanish at each boundary), reducing the parameter count from $K+4$ to K (or $K + 2$ if you also count the two boundary slopes/intercepts that remain free, depending on the textbook convention). In ISLP's count it is $K+2$ vs $K+4$, so two fewer.

(iii) **False**. A cubic regression spline is constructed to be continuous in the function and its first *and second* derivatives at every knot; the *third* derivative is generally discontinuous (it is the lowest derivative the construction is allowed to "break"). A function whose third derivative were continuous everywhere would be a single global cubic.

(iv) **False**. In a regression spline the knot locations are *fixed in advance* by the user (e.g. at quantiles of the predictor); only the spline coefficients are estimated by least squares. Joint estimation of knot positions is what would turn the problem non-linear, which is precisely why classical regression splines do *not* do it.

f) Principal component analysis *(5 %)*

Solution *(1 %)* **(i)** For *standardized* variables each has variance 1, so the total variance equals the number of variables: $\sum_j \lambda_j = p = \boxed{5}$. (Sanity: $2.1 + 1.4 + 0.8 + 0.5 + 0.2 = 5.0$. ✓)

Solution *(1 %)* **(ii)** $\text{PVE}_1 + \text{PVE}_2 = 0.42 + 0.28 = \boxed{0.70}$ (70%).

Solution *(1 %)* **(iii)** Cumulative PVE: PC1 $= 0.42$, PC1+PC2 $= 0.70$, through PC3 $= 0.86$, through PC4 $= 0.96$. We first cross 0.95 at PC4, so $\boxed{4}$ components are needed.

Solution *(1 %)* **(iv)** The score of x^* on PC1 is the inner product $z_1^* = \phi_1^\top x^*$:

$$\begin{aligned} z_1^* &= 0.55 \cdot 1.5 + 0.45 \cdot (-1) + (-0.40) \cdot 0.5 + 0.30 \cdot 2 + (-0.50) \cdot (-0.5) \\ &= 0.825 - 0.45 - 0.20 + 0.60 + 0.25 \\ &= \boxed{1.025}. \end{aligned}$$

Grading: 1 P for the right linear combination. Accept any answer in $[1.02, 1.03]$. *The loadings as stated have* $\|\phi_1\|^2 = 0.3025 + 0.2025 + 0.16 + 0.09 + 0.25 = 1.0050$, *slightly off unit-norm (rounding); do not penalise students for noticing.*

Solution *(1 %)* **(v)** The loading with largest absolute value is $|0.55|$ on $\boxed{X_1}$, so X_1 contributes most to PC1 in absolute terms.

g) Bootstrap *(3 %)*

Solution *(1 %)* **(i)** For $n = 8$, the probability that a particular original observation is *not* drawn in any of the n resamples is $(1 - 1/n)^n = (7/8)^8$. So

$$P(\text{included}) = 1 - \left(\tfrac{7}{8}\right)^8 = 1 - 0.3436\ldots \approx \boxed{0.656}.$$

Solution *(1 %)* **(ii)** As $n \to \infty$, $\left(1 - \frac{1}{n}\right)^n \to e^{-1}$, so

$$P(\text{included}) \to 1 - e^{-1} \approx \boxed{0.632}.$$

This is exactly the $\sim 1/3$ "out-of-bag" fraction underlying random-forest OOB error estimation.

Solution *(1 %)* **(iii) (A) True** — bootstrap is *with replacement*, so the same row can appear multiple times in one resample. Without replacement at the same size n you would just permute the data and learn nothing. **(B) False** — the bootstrap quantifies the *variability* of $\hat{\theta}$ (its sampling distribution / standard error / confidence interval); it does *not* correct for bias. The bootstrap distribution is centered around the original $\hat{\theta}$, not around the unknown true θ, so a biased estimator stays biased. **(C) False** — a *single* bootstrap sample yields one realisation $\hat{\theta}^*$ and says essentially nothing about the sampling distribution. The whole point of the bootstrap is to draw B resamples (B typically in the hundreds or thousands) and look at the spread.
Grading: 1 P all-or-nothing for the correct $(\mathrm{T}, \mathrm{F}, \mathrm{F})$ *pattern; deduct* 0.5 *if (B) is marked True (the canonical confusion: bootstrap quantifies variance, not bias).*

h) Random forests and out-of-bag error *(2 %)*

Solution *(0.5 P per statement.)*

(i) **True**. Standard ISLR/sklearn defaults: $m = \sqrt{p}$ for classification, $m = p/3$ for regression.

(ii) **False**. B is *not* tuned by CV. Random forest test error is monotonically non-increasing in B (averaging only reduces variance), so one just picks "enough" trees — typically 500–1000. (Contrast: in *boosting*, B is a real tuning parameter because boosting can overfit.)

(iii) **True**. Each tree is fit on a bootstrap sample, leaving roughly $1 - 1/e \approx 1/3$ of observations untouched. Predicting each i from only the trees that did not see it yields the OOB prediction; averaging the resulting losses gives the OOB error, an essentially-free test-error estimate.

(iv) **True**. In a random forest the per-tree pruning is *not* done; trees are grown deep (high variance, low bias individually). Averaging across many decorrelated trees is what reduces variance — pruning would just inflate bias.

i) K-means clustering *(2 %)*

Solution *(0.5 P per statement.)*

(i) **True**. The K-means objective is the within-cluster sum of squares (WCSS): $\sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$.

(ii) **False**. K is a *user-chosen* hyperparameter; the algorithm itself does not select it (one usually picks K via the elbow on the WCSS-vs-K curve, the silhouette, or domain knowledge).

(iii) **True**. The Lloyd algorithm is only guaranteed to reach a *local* minimum of WCSS; multiple random restarts and picking the lowest WCSS is the standard remedy.

(iv) **False**. A *hierarchy* cuttable at any level is what *hierarchical* clustering produces (the dendrogram). K-means returns one flat partition for each chosen K.

Problem 3 (16 %) — Theory and hand calculations

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

Solution *(2 %)* **(i)** The two key LDA modelling assumptions:

1. **Gaussian class-conditional densities.** Within each class, the predictor vector is multivariate normal: $\mathbf{X} \mid Y = k \sim \mathcal{N}_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$.
2. **Common (pooled) covariance matrix.** All classes share the *same* $\boldsymbol{\Sigma}$; only the means differ.

Why prefer LDA to QDA in the small-n_k regime: QDA estimates a separate $\boldsymbol{\Sigma}_k$ per class ($K \cdot p(p+1)/2$ extra covariance parameters); when each class has few observations these estimates have large variance and dominate the test error. LDA pools all training data into a single $\boldsymbol{\Sigma}$, drastically reducing variance at the price of some bias. When n_k is small, the variance saving dominates.

Solution *(3 %)* **(ii)** Starting from the log of the Bayes numerator:

$$\log\big(\pi_k f_k(\mathbf{x})\big) = \log \pi_k + \log f_k(\mathbf{x}).$$

For a multivariate normal with shared $\boldsymbol{\Sigma}$,

$$\log f_k(\mathbf{x}) = -\tfrac{p}{2}\log(2\pi) - \tfrac{1}{2}\log|\boldsymbol{\Sigma}| - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k).$$

Expand the quadratic form:

$$(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) = \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - 2\,\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k.$$

Drop terms that do not depend on k. The constants $-\frac{p}{2}\log(2\pi)$, $-\frac{1}{2}\log|\boldsymbol{\Sigma}|$, and the term $-\frac{1}{2}\mathbf{x}^\top\boldsymbol{\Sigma}^{-1}\mathbf{x}$ are the same for every k (this is the crucial step — the quadratic-in-$\mathbf{x}$ part survives only because $\boldsymbol{\Sigma}$ is shared, and so it cancels in pairwise comparisons of classes). What remains is

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

Why linear in x: only the first term involves $\mathbf{x}$, and it is a linear function $\mathbf{x} \mapsto (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k)^\top \mathbf{x}$. The other two terms depend only on k. Hence $\delta_k(\mathbf{x}) = \mathbf{a}_k^\top \mathbf{x} + c_k$ with $\mathbf{a}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$, and the boundary $\delta_k(\mathbf{x}) = \delta_\ell(\mathbf{x})$ is therefore a hyperplane.

Grading: 1 P for setting up $\log \pi_k + \log f_k(\mathbf{x})$ *and expanding the Mahalanobis quadratic form; 1 P for explicitly identifying which terms drop out (the* $\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}$ *cancellation is the punchline); 1 P for the linearity argument. Half credit is reasonable if the student writes the formula but does not explain why the* $\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}$ *piece cancels — this is the exact step where LDA differs from QDA.*

Solution *(2 %)* **(iii)** Set $\delta_0(\mathbf{x}) = \delta_1(\mathbf{x})$. With $\pi_0 = \pi_1$ the $\log \pi_k$ terms cancel, leaving

$$\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = \tfrac{1}{2}\big(\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0\big).$$

Compute the LHS direction. $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 = \begin{pmatrix}3\\5\end{pmatrix} - \begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}2\\4\end{pmatrix}$, so

$$\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = \frac{1}{15}\begin{pmatrix}4 & -1\\ -1 & 4\end{pmatrix}\begin{pmatrix}2\\4\end{pmatrix} = \frac{1}{15}\begin{pmatrix}4\cdot 2 + (-1)\cdot 4\\ (-1)\cdot 2 + 4\cdot 4\end{pmatrix} = \frac{1}{15}\begin{pmatrix}4\\14\end{pmatrix}.$$

Therefore

$$\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = \tfrac{1}{15}(4x_1 + 14x_2).$$

Compute the RHS scalar. $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 = \frac{1}{15}\begin{pmatrix}4-1\\-1+4\end{pmatrix} = \frac{1}{15}\begin{pmatrix}3\\3\end{pmatrix}$, so $\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 = \frac{1}{15}(1\cdot 3 + 1\cdot 3) = 6/15$. Similarly $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 = \frac{1}{15}\begin{pmatrix}4\cdot 3 - 5\\ -3 + 4\cdot 5\end{pmatrix} = \frac{1}{15}\begin{pmatrix}7\\17\end{pmatrix}$, so $\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 = \frac{1}{15}(3\cdot 7 + 5\cdot 17) = (21+85)/15 = 106/15$. Hence

$$\text{RHS} = \tfrac{1}{2}\big(106/15 - 6/15\big) = \tfrac{1}{2}\cdot\tfrac{100}{15} = \tfrac{50}{15}.$$

Combine. Multiply both sides by 15:

$$4x_1 + 14x_2 = 50, \qquad \text{or, dividing by 2,} \qquad \boxed{2x_1 + 7x_2 = 25}.$$

Grading: 1 P for arriving at the correct $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ *direction; 1 P for the correct constant on the RHS and the final equation in the form* $ax_1 + bx_2 = c$. *Accept any equivalent scaling, e.g.* $4x_1 + 14x_2 = 50$ *or* $x_2 = (25 - 2x_1)/7$. *Sanity check: the midpoint* $(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)/2 = (2, 3)$ *should sit on the line:* $2\cdot 2 + 7\cdot 3 = 4 + 21 = 25$. ✓

Solution *(1 %)* **(iv)** With class-specific covariance matrices the term $-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\mathbf{x}$ no longer cancels (it depends on k), so the discriminant is *quadratic* in $\mathbf{x}$ and the boundary is a quadratic curve (in 2D: a conic — ellipse, parabola, hyperbola, or pair of lines, depending on the covariance contrast). The method is **Quadratic Discriminant Analysis (QDA)**.

b) Hierarchical clustering by hand *(5 %)*

Solution *(3 %)* **(i)** The starting dissimilarity matrix has off-diagonal entries

$$d_{12} = 3,\ d_{13} = 7,\ d_{14} = 8,\ d_{23} = 5,\ d_{24} = 6,\ d_{34} = 4.$$

Merge 1 (height 3). The smallest dissimilarity is $d_{12} = 3$, so we merge $\{1\}$ and $\{2\}$ into $\{1, 2\}$ at height $\mathbf{h_1 = 3}$.

Under *single linkage* the new row/column entries are the *minima* of the merged rows:

$$d(\{1,2\},\, 3) = \min(d_{13}, d_{23}) = \min(7, 5) = 5,$$
$$d(\{1,2\},\, 4) = \min(d_{14}, d_{24}) = \min(8, 6) = 6.$$

The recomputed (3×3) dissimilarity matrix is therefore

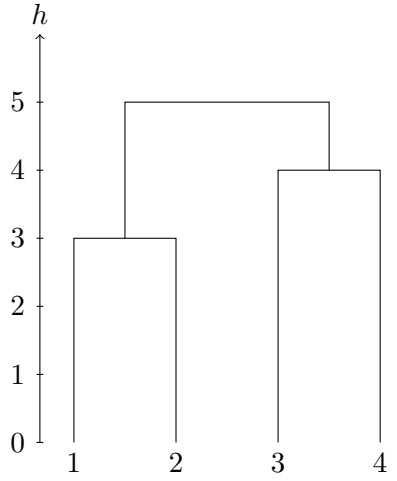
$$D^{(1)} = \begin{pmatrix}0 & 5 & 6\\ 5 & 0 & 4\\ 6 & 4 & 0\end{pmatrix} \quad \text{(rows/cols: } \{1,2\}, \{3\}, \{4\}\text{)}.$$

Merge 2 (height 4). The smallest off-diagonal in $D^{(1)}$ is $d(\{3\},\{4\}) = 4$; merge to get $\{3,4\}$ at height $\mathbf{h_2 = 4}$. Update with single linkage:

$$d(\{1,2\},\{3,4\}) = \min(d_{13}, d_{14}, d_{23}, d_{24}) = \min(7,8,5,6) = 5.$$

Merge 3 (height 5). Only two clusters remain; merge them at height $\mathbf{h_3 = 5}$.

Dendrogram. Leaves 1 and 2 fuse first at height 3; then leaves 3 and 4 fuse at height 4; finally the two clusters fuse at height 5.



Grading: 1 P first merge ($\{1,2\}$ at height 3) plus the recomputed 3×3 matrix; 1 P second merge ($\{3,4\}$ at 4) with correct single-linkage update; 1 P final merge at height 5 and a dendrogram with all three heights labelled on the vertical axis. Deduct 0.5 for an upside-down or unlabelled dendrogram.

Solution *(1 %)* **(ii)** Under *complete linkage* the first two merges are identical (the smallest pair is still $\{1,2\}$ at 3, then $\{3,4\}$ at 4 — both merges involve only singletons, where linkage has no effect). The final fusion height changes:

$$d(\{1,2\},\{3,4\}) = \max(d_{13}, d_{14}, d_{23}, d_{24}) = \max(7,8,5,6) = \boxed{8}.$$

It differs because complete linkage uses the *maximum* pairwise distance between the two clusters, while single linkage uses the *minimum*; whenever a cluster has more than one member the two rules disagree.

Solution *(1 %)* **(iii)** Cutting the single-linkage dendrogram at height 4.5 slices *above* the two bottom fusions (heights 3 and 4) but *below* the top fusion (height 5). We therefore obtain $\boxed{2}$ clusters: $\boxed{\{1,2\}}$ and $\boxed{\{3,4\}}$.

c) The bias–variance decomposition and double descent *(3 %)*

Solution *(2 %)* **(i)** The expected squared test error at the fixed test point x_0 decomposes *exactly* as

$$\mathbb{E}\big[(y_0 - \hat{f}(x_0))^2\big] = \underbrace{\big(f(x_0) - \mathbb{E}[\hat{f}(x_0)]\big)^2}_{\mathrm{Bias}^2[\hat{f}(x_0)]} + \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\text{variance}} + \underbrace{\sigma^2}_{\text{irreducible}}.$$

What each term is. The squared bias measures the systematic discrepancy between the truth $f(x_0)$ and the *average* prediction at x_0 across training sets; it captures error from using the

wrong model class. The variance measures how much the prediction at x_0 jitters across redraws of the training set; it captures error from chasing noise. The irreducible σ^2 is the variance of the observation noise ε and is independent of the estimator.

Expectations. The outer $\mathbb{E}[\cdot]$ in $\mathbb{E}[(y_0 - \hat{f}(x_0))^2]$ is taken *jointly* over (a) the random training sample used to fit $\hat{f}$, and (b) the noise ε in the new test point $y_0 = f(x_0) + \varepsilon$. The inner $\mathbb{E}[\hat{f}(x_0)]$ and $\mathrm{Var}(\hat{f}(x_0))$ in the formula above are over the training-set distribution only. *Grading: 1 P for the correct three-term formula with each term named; 1 P for stating what the expectations are over. Deduct 0.5 if σ^2 is omitted, and 0.5 if the bias is written without the square.*

Solution *(1 %)* **(ii)** Profile B is **double descent** (also called *benign overfitting* — the prof's hobbyhorse, returned to in modules 2, 5, 6, and 11). Both profiles are consistent with the same algebraic decomposition because that identity is *exact* for every estimator; it merely says that test-MSE $-\sigma^2$ splits into bias^2 and variance, not that variance must be U-shaped in flexibility. Past the interpolation point ($p \approx n$) the optimization changes character: there are infinitely many zero-training-error solutions and the algorithm (pseudoinverse, SGD, etc.) implicitly picks the *minimum-norm interpolator*. That implicit regularization is what causes variance to *decrease* again as p keeps growing, producing the second descent. Bias and variance still sum exactly to test MSE $-\sigma^2$ at every flexibility level. *Either of (a) "the identity is exact, only the variance shape changes" or (b) "implicit norm-minimization in the over-parameterized regime" earns the 1 P.*

Problem 4 (22 %) — Insurance premiums

a) Linear regression with interaction *(7 %)*

Solution *(1 %)* **(i)** Count the rows of the coefficient table: intercept, `age`, $I(\texttt{age}^2)$, `bmi`, `smoker_yes`, `bmi:smoker_yes`, `sex_male`, three `region` dummies (`NW, SE, SW`; `NE` is the reference), `children` = $\boxed{11}$ parameters. Sanity: residual d.f. = $n_{\text{train}} - p = 600 - 11 = 589$, which matches the printout. ✓

Solution *(2 %)* **(ii)** `charges` is in 1000 EUR. For a one-unit increase in `bmi`, holding everything else fixed:

$$\text{Non-smoker (}\texttt{smoker}{=}0\text{):}\quad \widehat{\Delta\texttt{charges}} = \hat{\beta}_{\texttt{bmi}} = 0.10\ (1000\,\text{EUR}) = \boxed{+100\ \text{EUR}},$$

$$\text{Smoker (}\texttt{smoker}{=}1\text{):}\quad \widehat{\Delta\texttt{charges}} = \hat{\beta}_{\texttt{bmi}} + \hat{\beta}_{\texttt{bmi:smoker\_yes}} = 0.10 + 1.45 = 1.55\ (1000\,\text{EUR}) = \boxed{+1{,}550\ \text{EU}}$$

Grading: 1 P each. Half credit if the EUR/1000 EUR conversion is forgotten. Deduct 1 P if the smoker effect is reported as 1.45 *alone (forgetting the main* `bmi` *effect) — the standard interaction trap.*

Solution *(2 %)* **(iii)** The main-effect coefficient $\hat{\beta}_{\texttt{smoker\_yes}} = -22.50$ is the smoker–non-smoker contrast *at* `bmi` $= 0$, holding everything else fixed. `bmi` $= 0$ is biologically impossible (BMI is in kg/m^2 with empirical range 16–50), so the coefficient is an *out-of-support extrapolation* with no real-world interpretation. The actual effect of smoking on charges depends strongly on `bmi` through the interaction $\hat{\beta}_{\texttt{bmi:smoker\_yes}} = 1.45$: at e.g. `bmi` $= 30$ the smoker–non-smoker contrast is $-22.50 + 1.45 \cdot 30 = +21.0$ (i.e. smokers pay $\sim$ 21,000 EUR more), the opposite sign of the main effect alone. Whenever an interaction term is in the model, the main effects are conditional contrasts at zero of the interacting variable, not "average" effects.

Solution *(1 %)* **(iv)** An ***F*-test** (a partial-*F* / ANOVA comparison of the full model to the reduced model with `region` dropped) on the *three* `region` dummies jointly. The null is

$$H_0:\ \beta_{\texttt{region\_NW}} = \beta_{\texttt{region\_SE}} = \beta_{\texttt{region\_SW}} = 0,$$

i.e. region has no effect on `charges` after controlling for the other predictors. (*Marginal t-tests on the individual dummies are not enough — they ignore the joint pattern, and individual dummies can be insignificant while the group is.*)

Solution *(1 %)* **(v)** The closeness of $R^2 = 0.78$ and adjusted $R^2 = 0.776$ (gap ≈ 0.004) indicates that essentially no predictor in the model is "free" — the few parameters added beyond the intercept all earn their keep, and there is no large penalty for over-parameterization at $p = 11, n = 600$. Equivalently, dropping any one predictor would change R^2 by only a small amount.

b) Ridge regression with 10-fold cross-validation *(5 %)*

Solution *(1 %)* **(i)** The ridge penalty $\lambda \sum_j \beta_j^2$ is *scale-dependent*: a predictor measured in large units has small coefficients and is barely penalized, while a predictor in small units has large coefficients and is heavily penalized. Standardising every predictor to mean 0 and variance 1 ensures the penalty acts on every predictor on the same footing, so shrinkage is determined by signal strength rather than units.

Solution *(1 %)* **(ii)** Ridge minimises

$$\hat{\beta}_\lambda^R = \arg\min_\beta \sum_{i=1}^{n} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

As $\lambda \to 0$ the penalty vanishes and $\hat{\beta}_\lambda^R \to \hat{\beta}^{\text{OLS}}$ (the OLS estimate, when it exists). As $\lambda \to \infty$ the penalty dominates and $\hat{\beta}_\lambda^R \to 0$ — every coefficient is shrunk to zero (the intercept is conventionally left unpenalised, so the fit collapses to $\bar{y}$).

Solution *(2 %)* **(iii) One-standard-error rule.** Among all values of λ whose CV-MSE is within one standard error (across the K folds) of the minimum CV-MSE, choose the *largest* (= most-regularized, simplest) one. Formally, with $\hat{\lambda}_{\min} = \arg\min_\lambda \text{CV}(\lambda)$ achieving CV-MSE $m^\star$ and standard error $\text{SE}^\star$,

$$\hat{\lambda}_{\text{1SE}} = \max\{\lambda : \text{CV}(\lambda) \le m^\star + \text{SE}^\star\}.$$

Why prefer it: the CV minimum is itself a noisy estimate of test error, so picking the exact argmin tends to slightly *overfit the validation folds*. The 1-SE choice yields a sparser, more interpretable, more stable model whose test performance is, within noise, indistinguishable from the best.
Grading: 1 P for stating the rule precisely (mention the "within one standard error" and *"simplest/largest λ" parts); 1 P for at least one defensible reason to prefer it (overfitting-of-CV / parsimony / stability).*

Solution *(1 %)* **(iv)** Ridge introduces bias to reduce variance; if the variance saving exceeds the bias introduced, ridge wins, otherwise OLS wins. Here ridge does *worse* (38.4 > 38.0), which suggests the bias from shrinkage exceeds whatever variance OLS suffers — i.e. at $n_{\text{train}} = 600$ and only 11 parameters the predictors carry real signal, $X^\top X$ is well-conditioned, and OLS is not over-fitting in the first place. There is little variance for ridge to remove.

c) GAM with splines *(5 %)*

Solution *(2 %)* **(i)** Term-by-term breakdown of consumed degrees of freedom:

- global intercept: 1,
- $s($`age`, df $= 4)$: 4 (smoothing spline, intercept removed),
- bs(`bmi`) with $K = 3$ knots $\{25, 30, 35\}$: cubic regression spline contributes $K + 3 = 6$ basis functions (intercept removed),

- $\beta_{\texttt{smoker}} \cdot \texttt{smoker}$: 1 (one binary dummy),
- $g(\texttt{region})$: 3 (region has 4 levels coded by 3 dummies against the `NE` reference),
- $\beta_{\texttt{kids}} \cdot \texttt{children}$: 1.

Total: $1 + 4 + 6 + 1 + 3 + 1 = \boxed{16}$ degrees of freedom.
Grading: 1 P for the per-term breakdown, 1 P for the correct total. Accept $K + 4 = 7$ for the BMI spline if the student counts the (suppressed) intercept; the global total shifts to 17 but the principle is the same. Deduct 0.5 for forgetting the categorical encoding on `region`*.*

Solution *(1 %)* **(ii)** The GAM is *additive*: each predictor enters through its own univariate function $f_j(x_j)$ and the contributions are summed. This rules out *interactions* unless they are added by hand — the model cannot, for example, learn that the effect of `age` on `charges` is different at low and high `bmi`. A regression tree, by contrast, can capture such interactions for free, because every split below the root is conditional on the splits above it.

Solution *(2 %)* **(iii)** Test MSE drops from 38.0 (OLS) to 32.5 (GAM), a substantial reduction of about 14%, which suggests genuine *nonlinearity in the continuous predictors*. The GAM is still additive, so the gain comes from non-linear shapes per predictor, not from interactions. Looking back at part (a): `age` entered only as $\texttt{age} + I(\texttt{age}^2)$ (the prof allowed only a polynomial of degree 2), but the GAM gives `age` four df — if the true age–charges relationship has more curvature than a parabola (e.g. a flat plateau in middle age followed by a steep climb after 50), the GAM captures it. `bmi` also gains substantially from the cubic spline. So I would suspect the `age` (and possibly `bmi`) smooths to be the main contributors to the improvement.

d) Tree boosting *(5 %)*

Solution *(2 %)* **(i)**

- ***B*** *(number of trees)*: chosen by CV (or a validation set). *Too small* $\Rightarrow$ high bias (underfit). *Too large* $\Rightarrow$ boosting can *overfit*, unlike random forests. So B trades bias for variance, but the variance penalty is mild if ν is small.
- ***d*** *(interaction depth, "tree size")*: each tree captures interactions up to order d. Small $d = 1$ (stump) means the ensemble is additive in inputs (lowest variance); larger d enables higher-order interactions but with more variance per tree. Chosen by CV; typical defaults $d \in \{1, 2, 4\}$.
- $\boldsymbol{\nu}$ *(shrinkage / learning rate)*: each tree's contribution is multiplied by ν. Small ν slows learning and forces a larger B to converge, but generally improves generalisation. Trades training-time for variance. Typical values $\nu \in [0.001, 0.1]$. The pair (B, ν) is tuned jointly — halving ν approximately doubles the required B.

Solution *(1 %)* **(ii)** A random forest averages B *independently grown* bootstrap trees; averaging only *reduces variance* and never inflates bias, so test error is monotonically non-increasing in B (it plateaus). One simply picks "enough" (say 500). Boosting, by contrast, is *sequential*: tree b is fit to the residuals of trees $1 \dots b-1$. With enough trees the residuals become pure noise, and additional trees fit it — so boosting *can overfit* as B grows, and B must be tuned by CV.

Solution *(2 %)* **(iii)** The ranking boosting = 26.8 < GAM = 32.5 < OLS = 38.0 suggests both *nonlinear shapes* (already captured by the GAM, which beats OLS) *and interactions among predictors* (which the GAM cannot capture but boosting can) — e.g. between `age` and `bmi`, or between `smoker` and `bmi`, the latter strongly flagged by part (a)'s significant interaction term. For interpretation from a black-box boosted model you would compute (i) a *variable-importance plot* (mean decrease in node impurity across all trees, or permutation importance) to rank

predictors by total contribution, and (ii) *partial dependence plots* for the top variables to recover the direction and rough shape of each effect on `charges`.

Problem 5 (24 %) — Stroke risk classification

a) Logistic regression *(8 %)*

Solution *(2 %)* **(i)** A unit increase in any continuous predictor multiplies the odds of the response by $\exp(\hat{\beta})$. For `age`:

$$\exp(0.073) \approx \boxed{1.076}.$$

Interpretation: each additional year of age multiplies the odds of stroke by about 1.076, i.e. roughly a 7.6% increase in the odds, holding all other predictors fixed. The effect compounds multiplicatively: a 10-year age difference raises the odds by a factor of $\exp(0.073 \cdot 10) \approx 2.08$.

Solution *(3 %)* **(ii)** The linear predictor is the dot product of the coefficient vector with the patient's covariates, including the intercept and the appropriate smoking dummy (`smoking_current` = 1, the other smoking dummies = 0):

$$\begin{aligned}\hat{\eta} &= -8.50 + 0.073 \cdot 68 + 0.005 \cdot 180 + (-0.005) \cdot 28 + 0.45 \cdot 1 + 0.50 \cdot 0 + 0.32 \cdot 1 \\ &= -8.50 + 4.964 + 0.900 - 0.140 + 0.450 + 0 + 0.320 \\ &= \boxed{-2.006}.\end{aligned}$$

Sigmoid step:

$$\hat{p} = \sigma(\hat{\eta}) = \frac{1}{1+e^{-\hat{\eta}}} = \frac{1}{1+e^{2.006}} = \frac{1}{1+7.434} = \frac{1}{8.434} \approx \boxed{0.119}.$$

Grading: 1 P each for the per-term contributions / correctly assembled $\hat{\eta}$, the value of $\hat{\eta} \approx -2.0$, and the sigmoid step / final $\hat{p} \approx 0.12$. Accept any answer in $[0.115, 0.125]$. Deduct 0.5 if the patient's `heart_disease` $= 0$ is incorrectly treated as 1 (a common slip when reading the problem). Deduct 0.5 if the wrong smoking dummy is used.

Solution *(1 %)* **(iii)** $\hat{p} \approx 0.119 < 0.5$, so at the default threshold the patient is classified as $\widehat{\texttt{stroke}} = 0$ (**no stroke**) — despite the patient being elderly, hypertensive, hyperglycaemic, and a current smoker. (The base rate of stroke is so low that even a fairly bad covariate profile does not cross 50%; see part (e) for why 0.5 is a poor threshold here.)

Solution *(2 %)* **(iv)** In this classification problem:

- **Sensitivity** = $P(\hat{Y} = 1 \mid Y = 1)$ = the proportion of *actual stroke cases* the classifier correctly flags. ("Of all the patients who actually had a stroke, what fraction did we catch?")
- **Specificity** = $P(\hat{Y} = 0 \mid Y = 0)$ = the proportion of *actual non-stroke cases* correctly cleared. ("Of all the patients who did not have a stroke, what fraction did we correctly leave alone?")

Clinically, high sensitivity + low specificity means the screening tool catches almost every patient who would have a stroke (very few missed strokes — few false negatives), at the cost of also flagging many patients who would not (many false alarms — many false positives). For a screening tool this is often the *right* trade-off: a missed stroke can be lethal, while a false alarm leads only to a follow-up test or unnecessary worry.

b) LDA vs. QDA *(6 %)*

Solution *(2 %)* **(i)** LDA confusion matrix totals: $50 + 70 = 120$ true strokes, $100 + 980 = 1080$ true non-strokes, $n = 1200$.

$$\text{Sensitivity}_{\text{LDA}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{50}{120} \approx \boxed{0.417},$$
$$\text{Specificity}_{\text{LDA}} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{980}{1080} \approx \boxed{0.907},$$
$$\text{Error rate}_{\text{LDA}} = \frac{\text{FP} + \text{FN}}{n} = \frac{100 + 70}{1200} = \frac{170}{1200} \approx \boxed{0.142}.$$

Solution *(2 %)* **(ii)** QDA confusion matrix totals: still 120 strokes, 1080 non-strokes, $n = 1200$.

$$\text{Sensitivity}_{\text{QDA}} = \frac{75}{120} = \boxed{0.625},$$
$$\text{Specificity}_{\text{QDA}} = \frac{830}{1080} \approx \boxed{0.769},$$
$$\text{Error rate}_{\text{QDA}} = \frac{250 + 45}{1200} = \frac{295}{1200} \approx \boxed{0.246}.$$

Grading: 1 P each for sensitivity & specificity, 1 P for the overall error rate, in each of (i) and (ii). Accept any reasonable rounding.

Solution *(2 %)* **(iii) Comparison.** LDA has a much lower overall error rate (0.142 vs. 0.246) and higher specificity (0.907 vs. 0.769), but QDA has substantially higher sensitivity (0.625 vs. 0.417).

- **Modelling difference:** LDA assumes a single shared covariance matrix while QDA estimates a separate covariance per class. With $K = 2$ classes and continuous + binary predictors, the binary predictors (`hypertension`, `heart_disease`) are Bernoulli, so the multivariate-normal assumption is questionable for both. With only ~ 140 stroke cases in the training set ($5\% \cdot 2800$), QDA's stroke covariance is poorly estimated (high variance), which inflates QDA's false-positive rate.
- **Medical context:** a missed stroke (false negative) is a much more serious clinical error than a false alarm (false positive — which leads only to a follow-up test). QDA's higher sensitivity — catching 63% vs. 42% of strokes — is therefore clinically valuable, even at the cost of more false alarms.
- **Preferred: QDA on sensitivity** (the medically dominant criterion); **LDA on overall error rate** (the conventional but here misleading criterion). I would pick QDA for clinical deployment, but would also re-examine the threshold of *either* classifier (lowering it on LDA can recover sensitivity at the same modelling cost).

c) A tree-based competitor: random forest *(5 %)*

Solution *(1 %)* **(i)** For a classification random forest the standard default is $m = \sqrt{p}$. With $p = 6$ predictors we have $\sqrt{6} \approx 2.45$, and rounding up gives $\boxed{m = 3}$, which is exactly `mtry`. (Some implementations use $\lfloor \sqrt{p} \rfloor = 2$; 3 is the typical sklearn / R `randomForest` default.) The randomness decorrelates the trees and lowers ensemble variance compared to bagging.

Solution *(1 %)* **(ii)** RF confusion matrix: 120 strokes, 1080 non-strokes, $n = 1200$.

$$\text{Sensitivity}_{\text{RF}} = \frac{65}{120} \approx \boxed{0.542},$$
$$\text{Specificity}_{\text{RF}} = \frac{950}{1080} \approx \boxed{0.880},$$
$$\text{Error rate}_{\text{RF}} = \frac{130 + 55}{1200} = \frac{185}{1200} \approx \boxed{0.154}.$$

The forest sits between LDA and QDA on every metric — moderately better sensitivity than LDA without LDA's specificity collapse.

Solution *(1 %)* **(iii)** The **out-of-bag (OOB) error** exploits the fact that each bootstrap sample omits $\approx 1 - 1/e \approx 37\%$ of the training observations (part 2g). For each training observation i, the OOB prediction is the majority vote (classification) or the mean (regression) of *only* those trees in whose bootstrap sample i did *not* appear. Averaging the resulting losses over all n training points gives an estimate of test error that is essentially free — no separate validation set, no CV refits. For random forests OOB error is known to be a near-unbiased proxy for test error.

Solution *(2 %)* **(iv)**

- **What a variable-importance plot shows:** for each predictor it reports a scalar summary of how much that predictor contributes to the forest's predictions. The two common variants are (a) the *mean decrease in node impurity* (Gini for classification, RSS for regression) summed over all splits on that variable across all trees, and (b) the *permutation importance* — the increase in OOB error when the values of that predictor are randomly permuted in the OOB sample.
- **What it cannot answer:** it does not tell you the *direction* or *shape* of each predictor's effect on the predicted probability — only the total magnitude of contribution. To recover direction you need partial-dependence plots, ICE plots, or a parametric model (e.g. logistic regression) for comparison. It also says nothing about *causality* — a predictor that is downstream of the outcome via a confounder can have very high importance while having no causal effect.

d) ROC and AUC *(3 %)*

Solution *(1 %)* **(i)** AUC = 0.84 means: if you draw one true stroke patient and one true non-stroke patient at random and look at their predicted probabilities $\hat{p}$ from the model, the stroke patient gets a higher $\hat{p}$ than the non-stroke patient about 84% of the time. AUC = 0.5 would be coin-flip ranking; AUC = 1 would be perfect ranking.

Solution *(1 %)* **(ii)** Lowering the threshold from 0.5 to 0.3 classifies *more* patients as positive. Therefore **sensitivity increases** (more true strokes crossed the lower threshold and were caught) and **specificity decreases** (more true non-strokes are now incorrectly flagged).

Solution *(1 %)* **(iii)** On the ROC curve (TPR vs. FPR), lowering the threshold moves the operating point **up and to the right** — TPR (= sensitivity) increases, FPR (= 1−specificity) increases.

e) Class imbalance *(2 %)*

Solution *(1 %)* **(i)** If the classifier always predicts "no stroke," it misclassifies exactly the 5% true strokes and correctly classifies the 95% non-strokes. The error rate is therefore $\boxed{0.05}$ (5%). This is already *lower* than every test error in parts (b)–(c) above (0.142 for LDA, 0.154 for RF, 0.246 for QDA), which illustrates the failure of the metric.

Solution *(1 %)* **(ii)** Test error rate is misleading on this dataset: a trivial "always no" classifier scores 0.05 and beats every model in part (b), simply because the class is so imbalanced that getting non-strokes right is cheap. Better metrics:

- the *pair* (**sensitivity, specificity**), which separately measures per-class performance;
- a threshold-independent summary such as **AUC** (or *precision-recall AUC*, which is more sensitive to imbalance);
- in a cost-sensitive clinical setting, **sensitivity at a fixed acceptable specificity** (or vice versa) directly encodes the medical trade-off that a missed stroke costs much more than a false alarm.

End of solution proposal. Total awarded: $10 + 28 + 16 + 22 + 24 = 100$ points.