TMA4268 V2026 Mock Exam 1

Solution Proposal

Compiled for Anders Bekkevard

Companion to `mock-exam-1.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-1.tex` for the problem statements.

Problem 1 (10 %) — Fill-in-the-blank

Solution *(1 P per blank.)*

(1) **supervised**

(2) **classification**

(3) **inference**

(4) **prediction**

(5) **irreducible**

(6) **regularization methods**

(7) **logistic regression**

(8) **generative**

(9) **principal component analysis**

(10) **hierarchical clustering**

Grading: 1 P per correct blank. No partial credit for "close" alternatives such as "parametric" instead of "supervised"; the passage continues with "and unsupervised learning", which forces the partition that supervised/unsupervised is the relevant split.

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

a) Bias–variance reasoning *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. More flexible models can capture more of the true f, reducing squared bias.

(ii) **False**. Variance *increases* with flexibility, because the fit follows the noise.

(iii) **False**. Large σ^2 rewards *less* flexible models: they have lower variance, and the irreducible σ^2 floors the test error anyway, so paying extra variance buys little.

(iv) **False**. σ^2 is the *irreducible* part of the test error decomposition; no amount of training data can shrink it (only changing what features are observed could).

b) Ridge vs. Lasso *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. The ridge penalty is differentiable everywhere; the optimum sets coefficients small but generically nonzero.

(ii) **True**. The ℓ_1 ball has corners on the axes; the contour of the RSS first hits the ball at a corner, producing exact zeros.

(iii) **False**. Ridge is *less* flexible than OLS: it adds a penalty that shrinks coefficients, reducing the effective degrees of freedom.

(iv) **True**. The penalty is scale-dependent; without standardization a predictor measured in large units is barely penalized and one in small units is heavily penalized.

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

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

$$\begin{aligned} \text{input} \to \text{hidden:} &\quad (5+1)\cdot 6 = 36 \\ \text{hidden} \to \text{output:} &\quad (6+1)\cdot 1 = 7 \\ \text{Total:} &\quad \boxed{36+7=43} \text{ parameters.} \end{aligned}$$

Grading: 1 P for each layer (1 P each), full credit if 43 is given with the correct breakdown. Deduct 0.5 if biases are forgotten (common slip $\Rightarrow 5\cdot 6 + 6\cdot 1 = 36$*).*

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

$$z = b + \sum_{j=1}^{5} w_j x_j = 0.3 + 0.5\cdot 2 + (-1.0)\cdot 1 + 0\cdot(-3) + 2\cdot 0.5 + (-0.5)\cdot 4$$

$$= 0.3 + 1.0 - 1.0 + 0 + 1.0 - 2.0 = -0.7.$$

ReLU: $\max(0,-0.7) = \boxed{0}$.
Grading: 1 P for the correct pre-activation -0.7*, 1 P for applying ReLU correctly to get* 0*.*

d) Odds and probability *(3 %)*

Solution *(1 %)* **(i)** Odds $= p/(1-p) = 2.5 \;\Rightarrow\; p = \dfrac{2.5}{1+2.5} = \dfrac{2.5}{3.5} = \boxed{\frac{5}{7} \approx 0.714}$.

Solution *(1 %)* **(ii)** $p = 0.30 \;\Rightarrow\; \text{odds} = \dfrac{0.30}{0.70} = \boxed{\frac{3}{7} \approx 0.4286}$.

Solution *(1 %)* **(iii)** A unit change in a logistic predictor multiplies the odds by $\exp(\hat{\beta})$. Here:

$$\exp(0.8) \approx \boxed{2.226}.$$

So the smoker's odds of the event are about $2.23\times$ those of an otherwise-identical non-smoker.

e) Smoothing splines and degrees of freedom *(3 %)*

Solution *(0.75 P per statement.)*

(i) **True**. With $\lambda = 0$ the roughness penalty vanishes and the minimizer interpolates the data (the natural cubic spline through every point).

(ii) **True**. With $\lambda \to \infty$ the penalty forces the second derivative to zero everywhere, i.e. the fit is a straight line — the OLS line through the data.

(iii) **False**. Increasing λ *decreases* the effective degrees of freedom $\mathrm{tr}(S_\lambda)$, because the smoother becomes closer to a straight line (df $\to 2$).

(iv) **True**. $\mathrm{tr}(S_\lambda)$ is the trace of a smoother matrix and is generally a non-integer real number between 2 and n.

f) Principal component analysis *(5 %)*

Solution *(1 %)* **(i)** For *standardized* variables each has variance 1, so the total variance equals the number of variables: $\sum_j \lambda_j = p = \boxed{6}$. (Sanity: $2.4 + 1.5 + 1.2 + 0.5 + 0.3 + 0.1 = 6$. ✓)

Solution *(1 %)* **(ii)** $\mathrm{PVE}_1 + \mathrm{PVE}_2 = 0.40 + 0.25 = \boxed{0.65}$ (65%).

Solution *(1 %)* **(iii)** Cumulative PVE: PC1 = 0.40, PC1+PC2 = 0.65, through PC3 = 0.85. So $\boxed{3}$ components are needed to reach (exactly) 85%.

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

$$\begin{aligned} z_1^* &= 0.60(1) + 0.50(-1) + 0.40(0.5) + 0.30(2) + 0.30(0) + 0.20(-1) \\ &= 0.60 - 0.50 + 0.20 + 0.60 + 0 - 0.20 = \boxed{0.70}. \end{aligned}$$

Note for the grader: the loading vector as stated has $\|\phi_1\|^2 = 0.36{+}0.25{+}0.16{+}0.09{+}0.09{+}0.04 = 0.99$, *which is slightly off unit-norm (rounding). Accept any answer in* $[0.69, 0.71]$.

Solution *(1 %)* **(v)** The loading with largest absolute value is 0.60 on $\boxed{X_1}$, so X_1 contributes most to PC1.

g) Sensitivity, specificity, ROC, AUC *(3 %)*

Solution *(1 %)* **(i)** Sensitivity = TP rate among true positives: $\dfrac{140}{200} = \boxed{0.70}$.

Solution *(1 %)* **(ii)** Specificity = TN rate among true negatives: $\dfrac{720}{800} = \boxed{0.90}$.

Solution *(1 %)* **(iii) (A), (C), (D)** are true. (B) is *false*: AUC is a single scalar summary (the average TPR over thresholds, or equivalently the probability of correctly ranking a random positive above a random negative); two ROC curves with very different shapes can integrate to the same area, so equal AUC does not pin down the curve.

h) Why nonlinear activations are necessary *(2 %)*

Solution *(0.5 P per correct mark.)* **(i) True**, **(ii) True**, (iii) False (a nonlinear activation does not by itself shrink weights; regularization is a separate mechanism, e.g. an ℓ_2 penalty or dropout), (iv) False (introducing a nonlinearity generally makes the loss surface *non*-convex in the weights; convexity holds for linear/logistic regression, not for multi-layer networks).

i) LDA vs. QDA *(2 %)*

Solution *(0.5 P per statement.)* **(i) True** — this is exactly the structural distinction between the two methods. **(ii) True** — expanding $\delta_k(\mathbf{x}) = \log \pi_k - \frac{1}{2}\mu_k^\top \Sigma^{-1}\mu_k + \mathbf{x}^\top \Sigma^{-1}\mu_k - \frac{1}{2}\mathbf{x}^\top \Sigma^{-1}\mathbf{x}$, the last term is the same for every k and cancels in pairwise comparisons; what remains is linear in $\mathbf{x}$. **(iii) False** — the reverse: with few observations per class, QDA's many extra covariance parameters drive variance up faster than the bias drops, so LDA usually wins (*small-n regime favours the lower-variance model*). **(iv) True** — both are generative classifiers; they estimate π_k and the within-class density $f_k(\mathbf{x})$, then apply Bayes' rule to obtain $P(Y = k \mid \mathbf{X} = \mathbf{x})$.

Problem 3 (16 %) — Theory and hand calculations

a) MLE = LS under Gaussian errors *(8 %)*

Solution *(2 %)* **(i)** The least-squares estimator solves

$$\hat{\theta}_{\text{LS}} \;=\; \arg\min_{\theta} \sum_{i=1}^{n} \big(Y_i - f_\theta(X_i)\big)^2.$$

Minimal assumptions: the errors ε_i are i.i.d. with mean zero and *finite variance* (homoscedasticity, $\text{Var}(\varepsilon_i) = \sigma^2$, is also typically assumed so that the estimator is well-defined and the Gauss–Markov theorem applies). No distributional shape is needed.

Solution *(2 %)* **(ii)** Under $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ we have $Y_i \mid X_i \sim \mathcal{N}\big(f_\theta(X_i),\, \sigma^2\big)$, so the density of one observation is

$$p(y_i \mid X_i; \theta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\!\left(-\frac{(y_i - f_\theta(X_i))^2}{2\sigma^2}\right).$$

By independence the joint log-likelihood is

$$\ell(\theta, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\big(y_i - f_\theta(X_i)\big)^2.$$

Solution *(3 %)* **(iii)** We treat σ^2 as fixed (or simply note it does not involve θ) and maximise ℓ over θ. Of the three terms above:

- $-\frac{n}{2}\log(2\pi)$: constant in θ.
- $-\frac{n}{2}\log(\sigma^2)$: also independent of θ.
- $-\frac{1}{2\sigma^2}\sum_i (y_i - f_\theta(X_i))^2$: the only θ-dependent term.

Thus

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta}\; -\frac{1}{2\sigma^2}\sum_{i}\big(y_i - f_\theta(X_i)\big)^2 = \arg\min_{\theta}\; \sum_{i}\big(y_i - f_\theta(X_i)\big)^2 = \hat{\theta}_{\text{LS}}.$$

The factor $1/(2\sigma^2)$ is a positive constant that does not change the argmin. □
Grading: 1 P for stripping the constants from ℓ; 1 P for the sign-flip "max of $-Q$ = min of Q"; 1 P for the final conclusion. Deduct 0.5 if the constant $1/(2\sigma^2)$ is silently kept inside the arg min.

Solution *(1 %)* **(iv) No.** The Laplace log-density is $\log p_{\text{Lap}}(y \mid \mu, b) = -\log(2b) - |y - \mu|/b$, so maximizing the likelihood now leads to *minimum absolute deviation*: $\hat{\theta}_{\text{MLE}} = \arg\min_\theta \sum_i |y_i - f_\theta(X_i)|$, an ℓ_1 rather than ℓ_2 problem. The equivalence MLE = LS hinges on the *quadratic* form of the Gaussian log-density.

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

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

$$d_{12} = 4,\ d_{13} = 9,\ d_{14} = 8,\ d_{23} = 5,\ d_{24} = 6,\ d_{34} = 3.$$

Merge 1 (height 3). The smallest dissimilarity is $d_{34} = 3$, so we merge $\{3\}$ and $\{4\}$ into $\{3, 4\}$ at height $\mathbf{h_1 = 3}$.

Under *complete linkage* the new row/column entries are the *maxima* of the merged rows:

$$d(\{3,4\},\, 1) = \max(d_{31}, d_{41}) = \max(9, 8) = 9,$$
$$d(\{3,4\},\, 2) = \max(d_{32}, d_{42}) = \max(5, 6) = 6.$$

The recomputed (3×3) dissimilarity matrix is therefore

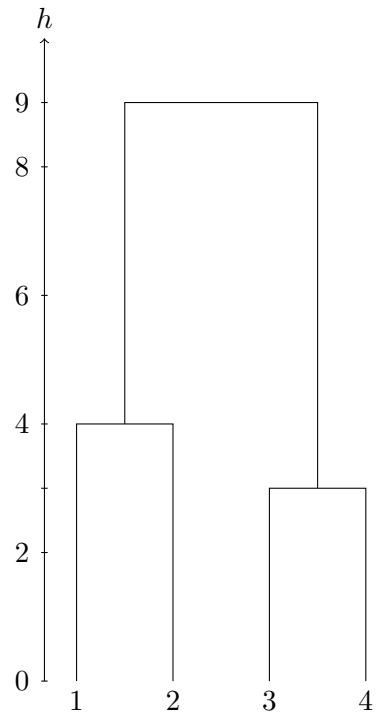
$$D^{(1)} = \begin{pmatrix} 0 & 4 & 9 \\ 4 & 0 & 6 \\ 9 & 6 & 0 \end{pmatrix} \quad \text{(rows/cols: } \{1\}, \{2\}, \{3,4\}\text{)}.$$

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

$$d(\{1,2\}, \{3,4\}) = \max(d_{13}, d_{14}, d_{23}, d_{24}) = \max(9, 8, 5, 6) = 9.$$

Merge 3 (height 9). Only two clusters remain; merge them at height $\mathbf{h_3 = 9}$.

Dendrogram (text + tikz). Leaves 3 and 4 fuse first at height 3; then leaves 1 and 2 fuse at height 4; finally the two clusters fuse at height 9.



Grading: 1 P first merge and updated matrix; 1 P second merge with correct complete-linkage update; 1 P final merge and a dendrogram showing all three heights labelled on the vertical axis. Deduct 0.5 if the heights are right but the dendrogram is drawn upside-down or unlabelled.

Solution *(1 %)* **(ii)** *Nothing* would change about the first merge: the smallest entry in D is $d_{34} = 3$ irrespective of linkage, since the first merge only requires choosing the closest *pair* of singletons. Linkage only matters from the *second* merge onward, when at least one cluster has size ≥ 2 and the linkage rule decides how to summarise it.

Solution *(1 %)* **(iii)** Cutting the complete-linkage dendrogram at height 7 crosses the bottom two fusions (heights 3 and 4) but not the top fusion (height 9). We obtain $\boxed{2}$ clusters: $\boxed{\{1, 2\}}$ and $\boxed{\{3, 4\}}$.

c) The bias–variance decomposition *(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 re-draws 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)** "Decomposition" is preferred because the identity above holds *exactly* for every estimator — it is an algebraic equality, not a constraint forcing one quantity to grow when the other shrinks. The label "trade-off" is misleading because (a) the bias term is *squared*, so a small absolute increase in bias contributes very little to MSE while variance can shrink a lot, and (b) changing the model class (e.g. adding regularization, or moving to the over-parameterized / *double-descent* regime of large neural networks) can lower variance *without* a matching increase in bias. *Any one of these two angles, stated clearly, earns the full 1 P.*

Problem 4 (22 %) — Used-car regression

a) Reading the table *(6 %)*

Solution *(1 %)* **(i)** Count the rows of the coefficient table: intercept, `mileage`, `age`, $I(\texttt{age}^2)$, `horsepower`, `engine`, `mpg`, two `fuel` dummies = $\boxed{9}$ parameters. Consistency check: residual d.f. $= n_{\text{train}} - p = 350 - 9 = 341$, which matches the printout.

Solution *(2 %)* **(ii)** The point estimate of the electric-vs-petrol effect is the coefficient on `fuel_electric`: $\hat{\beta} = 3.20$ (thousand EUR), with $\widehat{\mathrm{SE}} = 1.10$. A 95% (Wald) CI is

$$3.20 \pm 1.96 \cdot 1.10 \;=\; 3.20 \pm 2.156 \;=\; \boxed{[\,1.044,\; 5.356\,]} \text{ (in 1000 EUR).}$$

Interpretation: an electric car is, on average, about $3.20 \cdot 1000 = 3{,}200$ EUR more expensive than an otherwise-identical petrol car, with 95% CI roughly [1,000; 5,400] EUR. The interval excludes 0, consistent with $p = 0.004$.
Grading: 1 P point estimate, 1 P CI. Accept any minor rounding (e.g. $[1.05, 5.36]$*).*

Solution *(2 %)* **(iii)** The joint test of both `fuel` dummies against the reference (`petrol`) is an F**-test** (equivalently a partial-F / ANOVA-against-the-reduced-model). The output reports $F = 4.85$ on $(2, 341)$ d.f., $p = 0.008 < 0.05$, so we reject $H_0 : \beta_{\text{diesel}} = \beta_{\text{electric}} = 0$ and conclude that fuel type is jointly associated with price after controlling for the continuous predictors. The marginal z / t-tests on each dummy alone are insufficient here because they ignore the joint pattern (and diesel is individually non-significant).

Solution *(1 %)* **(iv)** She is right that the *marginal* t-test on `mpg` in this model fails to reject ($p = 0.135$): there is no clear linear effect of `mpg` on price *after controlling for the other predictors.* She is wrong if she means `mpg` is irrelevant absolutely — `mpg` is highly correlated with `engine` and `horsepower`, so its effect can be soaked up by those collinear predictors; on its own (or after variable selection) it might still matter.

b) Training vs. test error *(3 %)*

Solution *(1 %)* **(i) Lower (or equal).** Adding parameters to a least-squares fit can only decrease the training RSS, because the smaller model is nested inside the larger one and OLS minimises RSS. In the special case where the new terms are exactly collinear with existing ones (as $I(\texttt{age}/12)$ is with `age`) the training RSS is *unchanged.*

Solution *(1 %)* **(ii) Not certain.** The test RSS can move in either direction: adding a true cubic term might reduce test RSS if the relationship is genuinely cubic, but adding collinear or noise terms increases variance and typically inflates the test RSS. Without seeing the true generating process we cannot predict the sign.

Solution *(1 %)* **(iii)** The redundant term $I(\texttt{age}/12)$ is a perfect linear function of `age`, so the design matrix is *rank-deficient* and $X^\top X$ is singular: OLS has *no unique solution* (the software either drops one term or refuses to solve). Adding an ℓ_2 (ridge) penalty makes $X^\top X + \lambda I$ strictly positive-definite, hence invertible: the ridge fit is unique even with perfect collinearity, and the redundant predictor's coefficient is split between the two collinear columns rather than being undefined.

c) Lasso with 10-fold CV *(6 %)*

Solution *(1 %)* **(i)** For a fixed λ:

1. Partition the training set into $K = 10$ folds $\mathcal{F}_1, \ldots, \mathcal{F}_{10}$ of roughly equal size.
2. For each $k = 1, \ldots, 10$: fit the lasso at penalty λ on the data in $\bigcup_{j \neq k} \mathcal{F}_j$; obtain $\hat{f}_\lambda^{(-k)}$.
3. Compute the fold MSE $\text{MSE}_k(\lambda) = |\mathcal{F}_k|^{-1} \sum_{i \in \mathcal{F}_k} (y_i - \hat{f}_\lambda^{(-k)}(x_i))^2$.
4. Average over folds: $\widehat{\text{CV}}(\lambda) = K^{-1} \sum_{k=1}^{K} \text{MSE}_k(\lambda)$.

Solution *(1 %)* **(ii)** Both are defensible. $\hat{\lambda}_{\min}$ is the penalty that minimises CV-MSE on the curve and is the right choice if pure prediction accuracy is the goal. $\hat{\lambda}_{\mathbf{1SE}}$ is the largest λ whose CV-MSE is within one standard error of the minimum; it produces a *sparser, more interpretable* model at the cost of a small bump in CV error. Given that the prof has emphasised parsimony and that $\hat{\lambda}_{\text{1SE}}$ trades only $5.55 - 5.20 = 0.35$ MSE for substantially more shrinkage and likely some zeroed coefficients, I would pick $\hat{\lambda}_{\text{1SE}}$. (Picking $\hat{\lambda}_{\min}$ with the prediction-accuracy justification also gets full credit.)

Solution *(2 %)* **(iii)** OLS test MSE $= 5.15 <$ lasso test MSE at $\hat{\lambda}_{\min} = 5.20$. The lasso fit, which shrinks (and possibly zeros) coefficients, performs *worse* than OLS, which suggests there is no strong bias–variance gain to be had: most of the predictors carry real signal, so shrinkage trades

useful bias for too little variance reduction. Equivalently, the predictor set is already roughly the right size and not severely over-parameterised relative to $n_{\text{train}} = 350$. In the bias–variance language: the additional bias introduced by the ℓ_1 penalty exceeds the variance it saves.

Solution *(2 %)* **(iv) One-standard-error (1-SE) rule.** Among all values of λ whose CV-MSE is within one standard error of the minimum CV-MSE, choose the *largest* one (equivalently: the *simplest* / most-regularized model whose performance is statistically indistinguishable from the best). Formally, if $\hat{\lambda}_{\min}$ achieves CV-MSE $m^\star$ with standard error $\text{SE}^\star$, set

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

Why prefer $\hat{\lambda}_{\mathbf{1SE}}$ over $\hat{\lambda}_{\min}$. (1) The CV minimum is itself a noisy estimate of test error, so picking the exact argmin tends to slightly *overfit* the validation folds. (2) The 1-SE choice yields a sparser, more interpretable, more stable model whose test performance is, within noise, no worse. The cost is a small bias the practitioner accepts in exchange for parsimony — a direct bias–variance lever applied to model selection itself.
Grading: 1 P for stating the rule precisely (mention the "within one standard error" and "simplest" parts); 1 P for at least one defensible reason to prefer it (overfitting-of-CV, parsimony, or stability).

d) Tree boosting *(5 %)*

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

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

Solution *(1 %)* **(ii)** Boosting can *overfit* for very large B: each new tree fits residuals of the current ensemble, and with enough trees those residuals are just noise. So B is a genuine tuning parameter and "infinite B" is not safe (in contrast to random forests, where averaging $B \to \infty$ only reduces variance).

Solution *(2 %)* **(iii)** A boosting test MSE of 3.85 versus OLS 5.15 suggests *nonlinearities and/or interactions* that the linear model cannot capture (e.g. nonlinear depreciation in `age`, interactions between `horsepower` and `engine`, or between fuel type and `mileage`). For interpretation you would compute *variable importance* (mean decrease in node impurity or permutation importance), inspect *partial dependence plots* for the top variables, and possibly fit a GAM with smooth terms (which trades some accuracy for transparency).

e) GAM degrees of freedom *(2 %)*

Solution *(1 %)* **(i)** A cubic regression spline with K internal knots uses $K + 3$ basis functions when the global intercept is removed (or, equivalently, $K + 4$ d.f. including the intercept). Here $K = 4$, so $f_1(\texttt{mileage})$ alone consumes $\boxed{K + 3 = 4 + 3 = 7}$ d.f.

Solution *(1 %)* **(ii)** Total d.f. (including global intercept):

$$\begin{aligned}\text{intercept} \; &+ f_1 + f_2 + f_3 + \beta_4(\texttt{engine}) + \beta_5(\texttt{mpg}) + g(\texttt{fuel}) \\ &= 1 + 7 + 7 + 7 + 1 + 1 + 2 \;=\; \boxed{26} \text{ d.f.}\end{aligned}$$

($g(\texttt{fuel})$ contributes 2 because the categorical variable has 3 levels coded by 2 dummy contrasts against the reference `petrol`.)

Problem 5 (24 %) — Loan-default classification

a) Logistic regression with interaction *(10 %)*

Solution *(3 %)* **(i)** With the interaction `balance:sex`, the log-odds of default for a unit increase in `balance` (1000 EUR) depend on sex:

$$\begin{aligned}\text{Male (reference)}: &\quad \Delta \log \text{odds} = \beta_{\texttt{balance}} = 1.00 \quad \Rightarrow \quad \text{odds factor} = e^{1.00} \approx \boxed{2.72}, \\ \text{Female}: &\quad \Delta \log \text{odds} = \beta_{\texttt{balance}} + \beta_{\texttt{balance:female}} = 1.25 \quad \Rightarrow \quad \text{odds factor} = e^{1.25} \approx \boxed{3.49}.\end{aligned}$$

So each extra 1,000 EUR of monthly balance multiplies a man's odds of default by ~ 2.72 and a woman's by ~ 3.49.
Grading: 1 P male, 1 P female, 1 P for using the interaction correctly (deduct 1 if both factors are $e^{1.00}$, which ignores the interaction).

Solution *(2 %)* **(ii)** The encoding assumption: `sex` is dummy-coded with *male* as the reference, so `sex_female` = 1 for women, 0 for men. In a model with an interaction `balance:sex`, the main-effect coefficient $\hat{\beta}_{\text{sex\_female}} = -0.18$ is the difference in log-odds between a woman and a man *at* `balance` = 0. Since `balance` = 0 is far outside the bulk of the data, this number is a near-meaningless intercept-level contrast; the actual gender effect varies with `balance` (because of the interaction) and is dominated by the much-larger interaction coefficient 0.25 for moderate-to-high balances. A clean "average gender effect" would require integrating over the empirical distribution of `balance` or fitting a model without the interaction.

Solution *(2 %)* **(iii)** In this problem:

- **Sensitivity** = $P(\hat{Y} = 1 \mid Y = 1)$ = the proportion of *actual defaulters* the classifier correctly flags as defaulters. ("How many true defaulters do we catch?")
- **Specificity** = $P(\hat{Y} = 0 \mid Y = 0)$ = the proportion of *actual non-defaulters* correctly cleared. ("How many good customers do we leave alone?")

Solution *(3 %)* **(iv)** Test set has 3000 rows; 600 actual defaulters ($270 + 330$) and 2400 actual non-defaulters ($80 + 2320$).

$$\begin{aligned}\text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{270}{270 + 330} = \frac{270}{600} = \boxed{0.45}, \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{2320}{2320 + 80} = \frac{2320}{2400} \approx \boxed{0.967}, \\ \text{Error rate} &= \frac{\text{FP} + \text{FN}}{n} = \frac{80 + 330}{3000} = \frac{410}{3000} \approx \boxed{0.137}.\end{aligned}$$

Grading: 1 P per number. Round to two or three decimals.

b) ROC and AUC interpretation *(3 %)*

Solution *(1 %)* **(i)** The ROC curve plots the **true positive rate (TPR = sensitivity)** on the y-axis against the **false positive rate (FPR = 1−specificity)** on the x-axis, as the classification threshold is varied from 0 to 1.

Solution *(1 %)* **(ii)** The AUC is the area under the ROC curve; equivalently the probability that the classifier ranks a randomly chosen positive higher than a randomly chosen negative. AUC = 0.5 corresponds to a classifier no better than random guessing (the diagonal), while AUC = 1 is perfect ranking.

Solution *(1 %)* **(iii)** Lowering the threshold from 0.5 to 0.3 flags *more* observations as defaulters: sensitivity *increases* (we catch more true defaulters) while specificity *decreases* (more false alarms among non-defaulters). On the ROC curve the operating point moves *up and to the right* (higher TPR, higher FPR).

c) K-nearest neighbours *(4 %)*

Solution *(1 %)* **(i)** For a new test point x^*, the K-NN classifier computes the distance (Euclidean by default) from x^* to every training observation, picks the K closest ones, and predicts the *majority class* among those K neighbours (for a probability output, the fraction of neighbours in the positive class).

Solution *(2 %)* **(ii)** Same 600/2400 split.

$$\text{Sensitivity} = \frac{200}{200+400} = \frac{200}{600} \approx \boxed{0.333},$$
$$\text{Specificity} = \frac{2280}{2280+120} = \frac{2280}{2400} = \boxed{0.95},$$
$$\text{Error rate} = \frac{120+400}{3000} = \frac{520}{3000} \approx \boxed{0.173}.$$

Solution *(1 %)* **(iii)** KNN uses raw Euclidean distance, which is dominated by features on large scales. Here `balance` is in thousands while `pay_0` is on a small integer scale, so distances are effectively governed by `balance` alone and `pay_0`'s information is washed out. Standardising each predictor to unit variance gives every feature comparable influence on the neighbour ranking.

d) A tree-based competitor *(5 %)*

Solution *(2 %)* **(i)** I would fit a **random forest** for binary classification. Tuning parameters:

- **Number of trees** B: ~ 500. Variance keeps decreasing with B but plateaus; 500 is the usual "safe default". Random forests do not overfit in B.
- **Number of predictors sampled per split** m: for classification the standard default is $m = \lfloor \sqrt{p} \rfloor$. With $p = 9$ predictors that is $m = 3$. The randomness decorrelates the trees and lowers variance versus bagging.
- **Terminal node size / depth**: leave trees grown deep (small minimum node size, e.g. 1 for classification). Variance is controlled by averaging across B trees, not by per-tree pruning.

Solution *(2 %)* **(ii)** The forest gives test sensitivity 0.42, specificity 0.96, error rate 0.14. Logistic regression gave 0.45/0.967/0.137. The two classifiers are *very close*, with logistic slightly ahead on every metric (higher sensitivity, marginally higher specificity, lower error rate). Given

the comparable performance and the fact that logistic regression also provides *interpretable coefficients* (an odds-ratio per predictor, with p-values and CIs), I would prefer logistic regression here. The criterion driving the choice is *interpretability at no cost in accuracy*; if one had to choose on accuracy alone the two are within sampling noise and the random forest's slightly worse sensitivity is the deciding factor in a default-flagging application where catching defaulters matters more than avoiding false alarms.

Solution *(1 %)* **(iii)** A variable-importance plot ranks predictors by the average decrease in node impurity (Gini or entropy) attributable to each predictor across all trees, or by the increase in out-of-bag error when that predictor's values are permuted. It *can* tell you which predictors the model uses most strongly. It *cannot* tell you the *direction* or *magnitude* of each predictor's effect on the probability of default — for that you need partial-dependence plots or a parametric model.

e) Class imbalance *(2 %)*

Solution *(1 %)* **(i)** If the classifier always predicts "no default," it misclassifies exactly the 20% true defaulters and correctly classifies the 80% non-defaulters. The error rate is therefore $\boxed{0.20}$ (20%).

Solution *(1 %)* **(ii)** Test error rate is a misleading metric here, because the naive 0.20 baseline already beats many "working" classifiers and a 0.137 error rate (part (a)) mostly reflects the bank's ability to predict the easy non-default cases. Better metrics for an imbalanced classification problem are the pair (**sensitivity, specificity**), since they separately measure performance on the two classes, and/or a threshold-independent summary such as the **AUC** or the F_1 **score**; in a cost-sensitive setting where missed defaulters are expensive, sensitivity at a fixed acceptable specificity (or precision-recall AUC) is the right target.

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