TMA4268 Statistical Learning V2026

Mock Exam 10 (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 have an outcome Y and predictors X. A method that assumes a specific functional form $f(x;\theta)$ for the regression function and fits a small, finite set of parameters θ is called _____ (1) (*parametric / nonparametric / generative / ensemble*). When the goal is to forecast the response on new inputs we speak of _____ (2) (*clustering / calibration / prediction / inference*); when instead the goal is to understand which inputs drive the response and by how much, we speak of _____ (3) (*prediction / inference / boosting / cross-validation*).

A central decomposition of expected test error at a fixed test point x_0 splits it as $\text{Bias}^2 + \text{Var} + \sigma^2$, where the last term, the variance of the noise on the response, is called the _____ (4) (*reducible / Bayes-optimal / cross-validated / irreducible*) error. The classical wisdom relating the first two terms is the bias–variance *trade-off*; in the over-parameterized regime where the number of parameters far exceeds n and yet test error keeps falling past the interpolation threshold, the phenomenon is called _____ (5) (*shrinkage / double descent / stacking / cold start*).

Among classifiers, methods that model $\Pr(X \mid Y)$ together with a prior $\pi_k = \Pr(Y = k)$ and then invert with Bayes' rule are called _____ (6) (*discriminative / nonparametric / generative / shrinkage*) classifiers. The canonical example in this course assumes the class-conditional densities are Gaussian with a *shared* covariance matrix, giving a decision boundary that is linear in x; this method is called _____ (7) (*QDA / logistic regression / naive Bayes / LDA*).

In tree ensembles, growing many *independent* bootstrap-replicated trees in parallel and averaging them is called _____ (8) (*bagging / boosting / stacking / pruning*); growing many small trees *sequentially*, each one a small correction to the previous ensemble, is called _____ (9) (*bagging / boosting / cost-complexity pruning / stratified sampling*). When the goal is both to *select* a hyperparameter and to obtain an *honest* estimate of test error from the same training data, the appropriate resampling procedure is _____ (10) (*stratified cross-validation / validation-set approach / nested cross-validation / the bootstrap*).

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, noise, and benign overfitting *(3 %)*

Mark each statement true or false.

(i) In a problem with very large irreducible noise σ^2, a flexible (high-variance) estimator is generally a worse choice than a less-flexible (higher-bias) one, because the flexible model spends its capacity chasing noise that no estimator can recover.

(ii) Doubling the training-set size n, with everything else held fixed, lowers σ^2 in the bias–variance decomposition by a factor of $1/n$.

(iii) Two estimators $\hat{f}_A$ and $\hat{f}_B$ have, at a fixed test point x_0, equal squared bias but $\mathrm{Var}(\hat{f}_A) < \mathrm{Var}(\hat{f}_B)$. Then $\hat{f}_A$ has strictly lower expected squared test error than $\hat{f}_B$ at that point.

(iv) In the over-parameterized regime, mini-batch SGD applied to an over-parameterized neural network tends to converge to the *minimum-norm* solution among the many that achieve zero training loss — and this implicit bias is one of the ingredients of benign overfitting.

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

(i) *(1 %)* A 10-fold cross-validation gives per-fold test MSEs

$$\{2.20,\ 1.95,\ 2.40,\ 2.10,\ 2.05,\ 2.55,\ 2.25,\ 2.30,\ 2.00,\ 2.20\}.$$

Compute (a) the CV-MSE estimate of test error, and (b) the standard-error estimate $\widehat{\mathrm{SE}}(\mathrm{CV}_{10})$ used by the one-SE rule, taken as the sample standard deviation of the per-fold MSEs divided by $\sqrt{k}$. Two decimals each.

(ii) *(1 %)* A second statistician runs the same CV at six values of a hyperparameter λ and obtains the table below. Apply the *one-standard-error rule* to pick a λ; show the bound you used. Assume that "simpler" means *larger* λ here (more shrinkage = simpler model).

λ	0.01	0.05	0.10	0.20	0.50	1.00
$\mathrm{CV}_{10}(\lambda)$	2.40	2.20	2.15	2.18	2.30	2.60

Use the standard error you computed in (i) as a stand-in for $\widehat{\mathrm{SE}}(\mathrm{CV}_{10}(\hat{\lambda}))$.

(iii) *(1 %)* Mark each statement true or false. (A) Leave-one-out CV is generally an approximately *unbiased* estimator of test error, but its variance can be larger than that of 10-fold CV because its n training sets overlap almost perfectly. (B) When the observations are independently and identically distributed, a careful random partition into folds is required for the CV estimator to be approximately honest; when observations are *temporally autocorrelated* (e.g. daily time-series data), the same random-folds procedure gives an honest test-error estimate as long as $k \geq 10$.

(iv) *(1 %)* Briefly state *why*, even when applied correctly, the standard-error estimate used by the one-SE rule is "not quite valid." (One sentence.)

c) Bootstrap: pairs vs. residuals, percentile vs. basic CI *(3 %)*

(i) *(1 %)* You have $\{(x_i, y_i)\}_{i=1}^n$ from a regression problem and want a bootstrap SE for the prediction $\hat{y}(x_0)$ at a new fixed point x_0. Two standard bootstrap schemes exist: *paired* (resample (x_i^*, y_i^*) rows with replacement) and *residual* (refit the model, then resample residuals $\hat{\varepsilon}_i$ with replacement and reattach them to $\hat{f}(x_i)$). State which scheme is generally preferred when the design is observational with random X, and one short reason.

(ii) *(1 %)* You have collected $B = 1000$ bootstrap replicates $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$ of a statistic $\hat{\theta}$. Specify the *percentile* 95% CI for θ as two specific quantiles of the bootstrap distribution.

(iii) *(1 %)* Mark each statement true or false. (A) The bootstrap standard error of $\hat{\theta}$ is the sample standard deviation of $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$. (B) A typical value of B for a standard-error estimate is on the order of 1,000–10,000. (C) Bootstrap resampling *automatically* corrects for the bias of $\hat{\theta}$ as an estimator of θ, so a confidence interval centred on $\hat{\theta}$ is automatically unbiased.

d) Backpropagation, mini-batch SGD, and learning rate *(3 %)*

Mark each statement true or false.

(i) Backpropagation is an algorithm that *computes* the gradient of the per-example loss with respect to every parameter of a feed-forward network; it is not itself an optimizer.

(ii) In mini-batch SGD with batch size m, the gradient estimate $\widehat{\nabla L_m}$ has the same expectation as the full-data gradient but a variance that scales as $1/m$.

(iii) In standard practice, a learning rate $\eta = 2$ is recommended over $\eta = 0.1$ for feed-forward networks trained on tabular data, because larger steps speed up convergence.

(iv) For a feed-forward network with squared-error output loss $\frac{1}{2}(\hat{f} - y)^2$, the derivative $\partial L / \partial \hat{f}$ that seeds the backward pass at the output layer is $(\hat{f} - y)$.

e) Collinearity, eigenvalues, and dummy coding *(3 %)*

(i) *(1 %)* True or false: "If two predictors X_1 and X_2 in an OLS model are *exactly* collinear (say $X_2 \equiv 3X_1 - 2$), then the matrix $\mathbf{X}^\top \mathbf{X}$ is singular and the OLS coefficient vector $\hat{\boldsymbol{\beta}}$ is not unique; only the linear combination $\hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ is identified."

(ii) *(1 %)* True or false: "Encoding a categorical predictor with K levels using K dummy variables *in addition to* an intercept column produces an exactly collinear design ('dummy-variable trap'); the standard remedy is to add a small ridge penalty to the OLS objective, which restores identifiability of all K dummy coefficients individually."

(iii) *(1 %)* True or false: "Near-collinearity inflates the individual standard errors of the affected coefficients, but the standard error of the *joint* contribution $\hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ at any fixed (X_1, X_2) pair is much less affected — which is why predictions can still be stable even when the individual coefficients are not."

f) Boosting flavors: AdaBoost, gradient boosting, XGBoost *(3 %)*

(i) *(1 %)* True or false: "In gradient boosting, halving the shrinkage / learning rate ν approximately *doubles* the number of trees M needed to fit the training data to a comparable level; the pair (M, ν) is jointly tuned."

(ii) *(1 %)* True or false: "A reasonable default for the interaction depth of trees in gradient boosting is $d \in \{10, 20, \ldots, 50\}$ — deep trees that each capture rich, high-order interactions; using stumps ($d = 1$) would be too restrictive and defeat the bias–variance logic that motivates boosting."

(iii) *(1 %)* True or false: "XGBoost differs from vanilla gradient boosting in three concrete ways named in the lectures: (a) it adds an explicit L^2 (and optional L^1) penalty on the leaf weights to the tree-fitting objective, (b) it uses second-order (Hessian) information when scoring candidate splits, and (c) it supports row and column subsampling."

g) Logistic regression: odds with a multi-level categorical and an interaction *(3 %)*

A logistic regression of a binary outcome on a continuous predictor x, a categorical `group` $\in$ {A (ref.), B, C}, and the interaction $x \times$ `group` gives the coefficients

Term	$\hat{\beta}$
(Intercept)	-1.50
x	0.20
`group_B`	0.60
`group_C`	1.20
x : `group_B`	-0.05
x : `group_C`	-0.15

(i) *(1 %)* By what factor (two decimals) do the odds of the event change for a one-unit increase in x, holding all other predictors fixed, when the subject is in (a) `group` = A, (b) `group` = B, (c) `group` = C?

(ii) *(1 %)* A patient in `group` = C has $x = 4$. Compute the linear predictor $\hat{\eta}$ and the predicted probability $\hat{p}$ (three decimals).

(iii) *(1 %)* True or false: "In a binary logistic regression, the regression coefficient $\hat{\beta}_j$ for a continuous predictor admits the interpretation 'a one-unit increase in x_j raises the *probability* of the event by approximately $\hat{\beta}_j$'."

h) Neural-network regularization, with a numeric label-smoothing step *(3 %)*

(i) *(1 %)* A multiclass classifier with $C = 5$ classes uses label smoothing with parameter $\varepsilon = 0.10$. Write out the modified target vector that replaces the one-hot vector $(0, 0, 1, 0, 0)$. (Five numeric entries.)

(ii) *(1 %)* Mark each statement true or false. (A) Dropout is applied during training only; at test time the network is run without random masking, with the weights typically rescaled (or, equivalently, the activations rescaled during training) so that the expectation of each unit's input matches between train and test. (B) Early stopping returns the parameters from the epoch at which the *training* loss first stops decreasing.

(iii) *(1 %)* True or false: "A typical, course-recommended dropout rate is around 20%. Pushing the rate as high as 50% is, per the prof, generally too aggressive in fully-connected feed-forward networks."

i) PCA: total variance, scree-plot decision, score *(3 %)*

You perform PCA on **five standardized** variables and obtain eigenvalues

$$\lambda_1 = 2.10,\ \lambda_2 = 1.30,\ \lambda_3 = 0.80,\ \lambda_4 = 0.50,\ \lambda_5 = 0.30.$$

The first principal-component loading vector (entries rounded to two decimals) is

$$\phi_1 = (\,0.55,\ -0.40,\ 0.45,\ 0.50,\ -0.30\,)^\top.$$

(i) *(1 %)* What is the total variance in the standardized data, and what proportion is explained by the first three principal components combined? (Two decimals.)

(ii) *(1 %)* A new observation has standardized values $x^* = (\,1.0,\ 2.0,\ -1.0,\ 0.5,\ -0.5\,)^\top$. Compute its score z_1^* on PC1, two decimals.

(iii) *(1 %)* Mark each statement true or false. (A) The loadings ϕ_1 satisfy the unit-norm constraint $\sum_j \phi_{1j}^2 = 1$. (B) Performing PCA on the same dataset *without* first standardizing the variables generally yields the same loading vectors, because PCA is invariant under coordinate rescaling. (C) The eigenvalue λ_j equals the empirical variance of the score z_j along PC j in the standardized data.

Problem 3 (16 %) — Theory, math, and pseudocode

a) The mathy one — LDA decision boundary and the LDA → QDA bridge *(8 %)*

Consider a binary classification problem with two classes $\{A, B\}$. Assume the class-conditional densities are multivariate normal

$$X \mid Y = k \;\sim\; \mathcal{N}\big(\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}\big), \qquad k \in \{A, B\},$$

with a *shared* covariance matrix $\boldsymbol{\Sigma}$, and class priors π_A, π_B with $\pi_A + \pi_B = 1$.

(i) *(3 %)* Starting from Bayes' rule

$$\Pr(Y = k \mid X = x) \;=\; \frac{\pi_k\, f_k(x)}{\sum_{\ell \in \{A,B\}} \pi_\ell\, f_\ell(x)}, \qquad f_k(x) \;=\; \frac{1}{(2\pi)^{p/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\Big(-\tfrac{1}{2}(x-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu}_k)\Big),$$

derive 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 dropping every additive term that does not depend on k. Be explicit about *which* terms you drop and *why* they can be dropped without changing $\arg\max_k \delta_k(x)$.

(ii) *(2 %)* Setting $\delta_A(x) = \delta_B(x)$, show that the resulting Bayes-optimal decision boundary between the two classes is the affine hyperplane

$$\big(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\big)^\top \boldsymbol{\Sigma}^{-1}\, x \;=\; \tfrac{1}{2}\big(\boldsymbol{\mu}_A^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_A - \boldsymbol{\mu}_B^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_B\big) \;-\; \log\frac{\pi_A}{\pi_B},$$

i.e. a function that is *linear* in x. State, in one sentence, *which* feature of the LDA assumption is responsible for the boundary being linear (rather than quadratic).

(iii) *(2 %)* Take the worked example $\boldsymbol{\mu}_A = (1, 0)^\top$, $\boldsymbol{\mu}_B = (0, 3)^\top$, $\boldsymbol{\Sigma} = \mathbf{I}_2$, $\pi_A = \pi_B = 1/2$. Plug these values into the boundary equation derived in (ii) and reduce it to a single explicit equation of the form $a_1 x_1 + a_2 x_2 = c$. (One numeric equation; show the algebra.)

(iv) *(1 %)* Now consider QDA: drop the shared-covariance assumption and let $\boldsymbol{\Sigma}_k$ depend on the class. State, in one or two sentences, *which* term in the analogue of $\delta_k(x)$ now fails to cancel between classes, and *therefore* why the QDA decision boundary is quadratic rather than linear.

b) Pseudocode — k-fold cross-validation with the one-SE rule *(4 %)*

You have a training set $(\mathbf{X}, \mathbf{y})$ of size n, a model class indexed by a hyperparameter λ taking values on a finite grid $\Lambda = \{\lambda_1, \ldots, \lambda_T\}$ (ordered so that *larger* λ means a *simpler* model), and a fixed number of folds $K = 10$. You want to pick λ *by the one-standard-error rule*, then return a single final fitted model that uses this λ and is trained on the full data.

(i) *(3 %)* Write *pseudocode* (math, plain English, or programming-language-style is fine) for the procedure. Make explicit, by the structure of the loops:

- the random partition of the n observations into K disjoint folds;
- the per-fold loop that, at each $\lambda \in \Lambda$, fits on $K-1$ folds and evaluates on the held-out fold;

- how you aggregate per-fold MSEs into $\mathrm{CV}_K(\lambda)$ and $\widehat{\mathrm{SE}}(\mathrm{CV}_K(\lambda))$;
- how the one-SE rule picks $\lambda^* \in \Lambda$ given the curve and its SE band;
- the final refit on the full training set at λ^*.

Roughly 12–18 lines is appropriate.

(ii) *(1 %)* Briefly explain in one short sentence *why* the one-SE rule is, in the prof's view, the preferred selection criterion compared to picking $\arg\min_\lambda \mathrm{CV}_K(\lambda)$.

c) Backpropagation: binary cross-entropy with a sigmoid output *(4 %)*

Consider the tiny network with one input $x \in \mathbb{R}$, one hidden unit using a ReLU activation, and one output unit using a sigmoid activation:

$$z_1 = w_1 x + b_1, \qquad h = \mathrm{ReLU}(z_1) = \max(z_1, 0), \qquad z_2 = w_2 h + b_2, \qquad \hat{p} = \sigma(z_2) = \frac{1}{1 + e^{-z_2}}.$$

For a single training pair (x, y) with $y \in \{0, 1\}$, the binary cross-entropy loss is

$$L(y, \hat{p}) \;=\; -\big[\, y \log \hat{p} \;+\; (1 - y) \log(1 - \hat{p}) \,\big].$$

(i) *(2 %)* Using $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ and the chain rule, derive that

$$\frac{\partial L}{\partial z_2} \;=\; \hat{p} - y,$$

i.e. the gradient at the pre-activation of the output collapses to the classifier residual. Show the two intermediate steps ($\partial L / \partial \hat{p}$ and $\partial \hat{p} / \partial z_2$) and the cancellation that produces this result. *This is one of the reasons the sigmoid–cross-entropy pairing is preferred over sigmoid–squared-error.*

(ii) *(1 %)* Building on (i), derive the gradient with respect to w_2:

$$\frac{\partial L}{\partial w_2}.$$

Express your answer in terms of $\hat{p}$, y, and h.

(iii) *(1 %)* Finally, derive the gradient with respect to w_1:

$$\frac{\partial L}{\partial w_1}.$$

Express your answer in terms of $\hat{p}$, y, w_2, x, and an indicator $\mathbb{1}[z_1 > 0]$ (the derivative of ReLU at z_1).

Problem 4 (20 %) — Data analysis: diabetes progression

A research clinic has $n = 442$ diabetes patients (Efron et al. 2004) and measures a continuous quantitative response `prog` = disease progression one year after baseline. The available baseline predictors are:

- `age` (years, continuous);
- `sex` (binary: *female* = 0 reference, *male* = 1);
- `bmi` (continuous);
- `bp` — mean arterial pressure (continuous);
- `s1` — total cholesterol (continuous);
- `s2` — LDL cholesterol (continuous);
- `s5` — log of serum triglycerides (continuous).

The data are split 350/92 into training and test sets. *All continuous predictors are standardized to mean* 0 *and standard deviation* 1 *before fitting any of the models below.* It is known that on the training set $\mathrm{cor}(\texttt{s1}, \texttt{s2}) \approx 0.97$; all other pairwise predictor correlations are below 0.6 in absolute value.

a) OLS with a quadratic, an interaction, and a collinear pair *(7 %)*

On the training set, the course staff fit the OLS model

$$\begin{aligned}\texttt{prog} \sim\ & \texttt{age} + \texttt{sex} + \texttt{bmi} + I(\texttt{bmi}^2) + \texttt{bp} \\ & + \texttt{s1} + \texttt{s2} + \texttt{s5} + \texttt{bmi:sex}.\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**t-value**	**Pr(>|t|)**
(Intercept)	152.5	4.1	37.20	< 0.001
`age`	1.5	2.6	0.58	0.563
`sex`	−10.2	5.4	−1.89	0.060
`bmi`	24.0	3.0	8.00	< 0.001
$I(\texttt{bmi}^2)$	3.8	1.5	2.53	0.012
`bp`	16.0	2.6	6.15	< 0.001
`s1`	22.0	18.0	1.22	0.222
`s2`	−6.5	17.5	−0.37	0.711
`s5`	31.0	4.0	7.75	< 0.001
`bmi:sex`	−9.0	4.5	−2.00	0.046

Multiple $R^2 = 0.51$, Adjusted $R^2 = 0.49$. Residual standard error: 53.0 on 340 d.f.

(i) *(1 %)* How many parameters does this model estimate, including the intercept? Verify the count against the printed residual degrees of freedom ($n_{\text{train}} = 350$).

(ii) *(2 %)* The estimates of $\hat\beta_{\texttt{s1}}$ and $\hat\beta_{\texttt{s2}}$ are individually insignificant ($p = 0.222$ and 0.711) and their SEs (18.0 and 17.5) are roughly 4–6× larger than the SEs of the other predictors. Identify, in one or two sentences, the statistical phenomenon at work, and state *which two pieces of information from the table* (and from the problem statement) together let you diagnose it without needing to compute a VIF.

(iii) *(1 %)* A classmate writes: "Since the p-values of `s1` and `s2` are both > 0.05, neither variable is associated with disease progression, and we should drop them both." In one short sentence, explain why this reading is *wrong* given the diagnosis in (ii).

(iv) *(2 %)* The model contains `bmi`, $I(\texttt{bmi}^2)$, and `bmi:sex`. Consider two patients with all other (standardized) predictors at zero. Patient A: *female* (`sex` $= 0$), standardized `bmi` $= +1$. Patient B: *male* (`sex` $= 1$), standardized `bmi` $= +1$. Compute the predicted `prog` for each. Show the contribution of each relevant term.

(v) *(1 %)* State your standardization convention for `bmi` explicitly, then compute the predicted change in `prog` when standardized `bmi` is increased from $+1$ to $+2$ for a female patient (`sex` $= 0$), all other predictors fixed. *Don't forget the* $\texttt{bmi}^2$ *term.*

b) Lasso with 10-fold cross-validation *(4 %)*

The course staff next fit a lasso on the same 9-predictor design (the eight features above plus the `bmi:sex` interaction; $p = 9$ predictors), with λ chosen by 10-fold CV. The CV curve has its minimum at $\hat{\lambda}_{\min}$ and its one-SE choice at $\hat{\lambda}_{1\text{SE}}$. Test MSEs on the held-out 92 observations:

Method	**Test MSE**	**No. nonzero (excl. intercept)**
OLS (full model from part (a))	2,850	9
Lasso at $\hat{\lambda}_{\min}$	2,780	7
Lasso at $\hat{\lambda}_{1\text{SE}}$	2,820	5

(i) *(1 %)* Write down the lasso objective function explicitly (math or pseudocode), being careful about whether the intercept is penalised.

(ii) *(1 %)* Briefly explain in one sentence *why* the lasso path is sensitive to the choice of reference level for a categorical predictor (such as `sex`). What practical consequence does this have for interpretation when a dummy coefficient is shrunk to zero?

(iii) *(1 %)* Interpret the gap OLS 2,850 vs. Lasso-$\hat{\lambda}_{\min}$ 2,780 in bias–variance terms; refer back to the collinearity diagnosis from part (a)(ii) when explaining *which* reduction (bias or variance) the lasso is exploiting here.

(iv) *(1 %)* A practitioner argues: "I want the simpler $\hat{\lambda}_{1\text{SE}}$ model because the test-MSE penalty ($2{,}820 - 2{,}780 = 40$) is small compared to the gain in interpretability and stability." Briefly state the one-standard-error rule and give one short reason why this practitioner's preference is reasonable.

c) Gradient boosting — learning-rate / depth tradeoff *(5 %)*

A gradient-boosted regression-tree model is fit on the same training data. The course staff sweep two of the three boosting hyperparameters: shrinkage $\nu \in \{0.01, 0.05, 0.10\}$ and tree depth $d \in \{1, 3, 5\}$. The number of trees M is, in each case, chosen by an inner 10-fold CV (so M^* depends on ν, d). The resulting test MSEs and M^*'s are:

	$\nu = 0.01$	$\nu = 0.05$	$\nu = 0.10$
$d = 1$ (stumps)	2,700 ($M^* = 6{,}500$)	2,680 ($M^* = 1{,}400$)	2,690 ($M^* = 700$)
$d = 3$	2,650 ($M^* = 5{,}800$)	2,640 ($M^* = 1{,}200$)	2,660 ($M^* = 600$)
$d = 5$	2,680 ($M^* = 5{,}500$)	2,710 ($M^* = 1{,}100$)	2,780 ($M^* = 500$)

(i) *(2 %)* Write *pseudocode* for the squared-error gradient-boosting algorithm with M trees, learning rate ν, and a fixed tree-fitting subroutine $\mathcal{T}(\cdot)$ (which you may treat as a black box). Be explicit about (a) the initialization $\hat{f}^{(0)}$, (b) what the new tree $\widetilde{h}_m$ is fit to at iteration m, (c) where ν enters the update, (d) the final returned function. Three to six lines is appropriate.

(ii) *(1 %)* Across the row of stumps ($d = 1$), the inner-CV-chosen M^* goes from 700 to 1,400 to 6,500 as ν decreases from 0.10 to 0.01. In one sentence, state the quantitative "rule of thumb" for how M^* scales as ν is halved, and connect it to the bias–variance role of ν.

(iii) *(1 %)* The best test MSE in the table is at $(d, \nu) = (3, 0.05)$, with 2,640. The worst at small $\nu = 0.01$ is at $d = 5$ (depth-5 trees), 2,680, and the worst at large $\nu = 0.10$ is also at $d = 5$, 2,780. In two short sentences, explain why deeper trees combined with *either* extreme of ν tend to hurt — and frame your answer in bias–variance terms.

(iv) *(1 %)* A junior colleague proposes simply "setting $M = 10{,}000$ and $\nu = 0.01$ to be safe" — without inner CV. Comment in one short sentence on whether this is a sound approach for gradient boosting (and contrast with the analogous claim for a *random forest*, where the number of trees is not a serious tuning parameter).

d) GAM and random forest *(4 %)*

The course staff also fit (a) a generalized additive model (GAM) and (b) a random forest on the same training data.

GAM. The GAM is

$$\texttt{prog} \;=\; \beta_0 + s(\texttt{age}, \mathrm{df} = 4) + \beta_{\texttt{sex}}\texttt{sex} + s(\texttt{bmi}, \mathrm{df} = 5) + \mathrm{bs}(\texttt{bp}, \mathrm{knots} = \{q_{1/3}, q_{2/3}\}) + \beta_{\texttt{s1}}\texttt{s1} + \beta_{\texttt{s2}}\texttt{s2} + \beta_{\texttt{s5}}$$

where $s(\cdot, \mathrm{df} = k)$ is a smoothing spline with effective d.f. k (intercept removed), and $\mathrm{bs}(\cdot, \mathrm{knots})$ is a *cubic* regression spline with two interior knots (intercept removed). Test MSE = 2,760.

Random forest. A random forest is fit with $B = 500$ trees, each grown deep. Test MSE = 2,690.

(i) *(1 %)* How many degrees of freedom does the term $\mathrm{bs}(\texttt{bp}, \mathrm{knots} = \{q_{1/3}, q_{2/3}\})$ alone consume? (Count basis functions, excluding the global intercept.)

(ii) *(1 %)* How many total degrees of freedom does this GAM consume, including the global intercept?

(iii) *(1 %)* For a regression problem with $p = 9$ predictors, state the standard default value of `mtry` for a random forest, and in one short sentence state *why* one chooses `mtry` $< p$ (relate to bagging-with-correlated-trees).

(iv) *(1 %)* The random-forest variable-importance plot reports very high importance for `bmi`, `s5`, and `bp`. State *one* thing this plot *cannot* tell you about these predictors that a coefficient in part (a)'s OLS output *can*.

Problem 5 (26 %) — Data analysis: South African heart disease

A cardiology clinic has $n = 462$ adult males from the Western Cape and records the binary outcome `chd` (1 = coronary heart disease present, 0 = absent). The available predictors are:

- `age` — age in years (continuous);
- `tobacco` — cumulative kg-years of tobacco (continuous);
- `ldl` — LDL cholesterol (mmol/L, continuous);
- `adiposity` — body adiposity score (continuous);
- `typea` — Type-A behavior score (continuous);
- `obesity` — BMI-based obesity score (continuous);
- `famhist` — binary 0/1 (1 if family history of CHD).

The data are split 350/112 into training and test sets. Among the 112 test patients, 38 truly have CHD and 74 do not.

a) Logistic regression with a continuous-by-binary interaction *(8 %)*

A logistic regression model is fit on the training set:

$$\begin{aligned}\operatorname{logit}\big(\Pr(\texttt{chd} = 1 \mid X)\big) \;&= \beta_0 + \beta_1 \texttt{age} + \beta_2 \texttt{ldl} + \beta_3 \texttt{famhist} + \beta_4 \texttt{tobacco} \\ &\quad + \beta_5 \texttt{typea} + \beta_6 \texttt{adiposity} + \beta_7 \texttt{obesity} + \beta_8 (\texttt{ldl:famhist}).\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**z-value**	**Pr(>|z|)**
(Intercept)	−5.50	1.10	−5.00	< 0.001
`age`	0.045	0.010	4.50	< 0.001
`ldl`	0.18	0.06	3.00	0.003
`famhist`	0.85	0.30	2.83	0.005
`tobacco`	0.080	0.025	3.20	0.001
`typea`	0.035	0.012	2.92	0.004
`adiposity`	0.020	0.025	0.80	0.424
`obesity`	−0.050	0.040	−1.25	0.211
`ldl:famhist`	0.15	0.08	1.88	0.061

(Reference levels: `famhist` = 0. Predictors are on their raw scales — `age` in years, `ldl` in mmol/L, `tobacco` in kg-years, etc.)

(i) *(2 %)* For each additional mmol/L of `ldl`, by what factor do the odds of `chd` change for (a) a patient with *no* family history (`famhist` = 0) and (b) a patient *with* family history (`famhist` = 1), holding all other predictors fixed? (Two numeric factors, three decimals.) State your encoding assumption explicitly.

(ii) *(1 %)* Briefly explain in one or two sentences *why* the answer to (i) is not the same in the two groups, and *why* the main-effect coefficient $\hat{\beta}_{\texttt{ldl}} = 0.18$ alone is therefore not a useful summary of "the effect of LDL on CHD" in this fitted model.

(iii) *(3 %)* Consider a specific patient: `age` = 55, `tobacco` = 4.0, `ldl` = 5.0, `adiposity` = 25, `typea` = 55, `obesity` = 27, `famhist` = 1. Compute the linear predictor $\hat{\eta}$ and the predicted probability $\hat{p}$ (three decimals). Show $\hat{\eta}$ term by term, then apply the sigmoid.

(iv) *(1 %)* At the default classification threshold $\hat{p} = 0.5$, would this patient be flagged as "CHD"? Comment in one sentence on whether 0.5 is an appropriate threshold for screening given the base rate of `chd` in this dataset (roughly 35% positive).

(v) *(1 %)* A classmate writes: "$\hat{\beta}_{\texttt{obesity}} = -0.050$ is *negative*, so once we control for the other predictors, more obesity is *protective* against CHD." Briefly state, in one short sentence, why one should not take this estimate as a *causal* conclusion — referencing the kind of model the prof flagged repeatedly ("fancy correlations, not causal").

b) AdaBoost: deriving the classifier weight α_m *(6 %)*

To compare with logistic regression, an AdaBoost classifier with $M = 200$ stumps is fit on the same training data. Recall the algorithm: labels coded $y_i \in \{-1, +1\}$; sample weights initialized $w_i = 1/N$; at each round m, the weak learner G_m is chosen to minimize the weighted error

$$\text{err}_m = \frac{\sum_i w_i \mathbb{1}[y_i \neq G_m(x_i)]}{\sum_i w_i},$$

the classifier weight is set to $\alpha_m = \log\big((1 - \text{err}_m)/\text{err}_m\big)$, and the weights are updated as $w_i \leftarrow w_i \cdot \exp\big(\alpha_m \mathbb{1}[y_i \neq G_m(x_i)]\big)$.

A unifying view (see [[boosting-loss-functions]] and ESL ğ10.4) is that AdaBoost is *forward stagewise additive modeling* under the exponential loss

$$L(y, f) \;=\; \exp(-y\, f(x)).$$

At round m the ensemble is $f_{m-1}(x) = \sum_{\ell<m} \alpha_\ell G_\ell(x)$, and one chooses (α_m, G_m) jointly to minimize

$$Q(\alpha, G) \;=\; \sum_{i=1}^{N} w_i^{(m)} \exp\big(-\alpha\, y_i\, G(x_i)\big),$$

where $w_i^{(m)} = \exp(-y_i\, f_{m-1}(x_i))$ are the working weights.

(i) *(3 %)* For a fixed weak classifier $G \in \{-1, +1\}$, split the sum $\sum_i w_i^{(m)} \exp(-\alpha y_i G(x_i))$ into a part over correctly classified observations ($y_i G(x_i) = +1$) and a part over misclassified observations ($y_i G(x_i) = -1$), and show that minimizing $Q(\alpha, G)$ over α yields

$$\alpha^* \;=\; \tfrac{1}{2} \log \frac{1 - \text{err}_m}{\text{err}_m},$$

where err_m is the weighted error of G as defined above. The AdaBoost.M1 convention (used in the prof's notes) absorbs the $\frac{1}{2}$ into the weight update, which is why the algorithm above uses $\alpha_m = \log\big((1 - \text{err}_m)/\text{err}_m\big)$ rather than $\frac{1}{2}$ of it. Show the algebra clearly.

(ii) *(1 %)* For $\text{err}_m = 0.30$, compute (a) the prof's $\alpha_m = \log\big((1 - \text{err}_m)/\text{err}_m\big)$ and (b) the multiplicative weight update factor $\exp(\alpha_m)$ applied to a *misclassified* observation at the next round. Two decimals each.

(iii) *(1 %)* Explain, in one short sentence, what happens to α_m when a base learner has weighted error *above* 0.5, and what consequence this has for the role of G_m in the final ensemble vote.

(iv) *(1 %)* The prof said in lecture that AdaBoost typically uses very *shallow* trees (often stumps) as base learners, rather than deep trees. Give a one-sentence bias–variance argument explaining *why* deep individual trees would make a poor AdaBoost base learner.

c) A neural-network classifier with regularization *(6 %)*

The clinic also fits a small fully-connected feed-forward neural network on the same training set with $p = 7$ inputs (the seven raw predictors above, with the continuous ones standardized). Architecture: 7 inputs $\to$ 20 hidden ReLU units (with biases) $\to$ 10 hidden ReLU units (with biases) $\to$ 1 sigmoid output (with bias). Trained with mini-batch SGD on binary cross-entropy, batch size 64.

(i) *(1 %)* How many parameters does this network have in total, *including all bias terms*? Show the layer-by-layer breakdown.

(ii) *(1 %)* A neuron in the *first* hidden layer has weights $w = (0.10, 0.30, -0.20, 0.50, 0.05, -0.40, 0.20)^\top$ and bias $b = -0.10$. For an input $x = (0.5,\ 1.0,\ -1.0,\ 2.0,\ 0.5,\ 0.5,\ -0.5)^\top$, compute the output of this neuron under ReLU activation. Show both the pre-activation z and the post-activation h, two decimals.

(iii) *(1 %)* Trained without any explicit regularization, the network reaches training accuracy 98% but test accuracy 0.70 — noticeably below the logistic regression of part (a). Briefly explain in one sentence why this outcome is consistent with the prof's iron rule that "you should never train a neural network without regularization," and frame it in bias–variance terms.

(iv) *(1 %)* The student then adds three regularizers: (a) L^2 weight decay with coefficient $\lambda = 10^{-4}$, (b) dropout at rate 0.2 on each hidden layer, and (c) early stopping on a held-out 20% of the training set. For each one, state in *one short sentence* the mechanism by which it regularizes *during training*. For dropout, also state what changes between training time and test time.

(v) *(1 %)* A second student instead uses *label smoothing* (with $\varepsilon = 0.10$) on the binary output target. State the modified target vector for a positive training example (originally $y = 1$, treated as the two-class one-hot $(0, 1)$) and one short sentence on why this helps when the recorded labels may themselves contain errors.

(vi) *(1 %)* The fully-regularized network achieves test accuracy 0.78, beating the logistic regression. A friend says: "The network had 7 inputs and only 350 training points — by the bias–variance trade-off it should not work." In one short sentence, refer back to the over-parameterized / benign-overfitting regime to explain why the friend's claim is too strong.

d) Comparison of classifiers and class imbalance *(4 %)*

On the test set, the following confusion matrices are obtained at default thresholds:

Logistic regression (threshold $\hat{p} = 0.5$):

	Pred: CHD	**Pred: no CHD**
Actual: CHD	18	20
Actual: no CHD	10	64

AdaBoost ($M = 200$ stumps):

	Pred: CHD	**Pred: no CHD**
Actual: CHD	25	13
Actual: no CHD	14	60

(i) *(2 %)* Compute the sensitivity, specificity, and overall test error rate for each of the two classifiers (six numeric answers, two decimals each).

(ii) *(1 %)* A naive classifier that always predicts "no CHD" achieves what test error rate on this dataset? Justify in one sentence, and briefly comment on whether *accuracy alone* would let you discriminate between the logistic regression, the naive classifier, and AdaBoost.

(iii) *(1 %)* Suppose the clinical decision is to flag patients for early-intervention follow-up: a *false negative* (sending a true CHD patient home without follow-up) is much more costly than a *false positive* (an unnecessary follow-up appointment). Which of the two classifiers above would you deploy, and on which single metric do you base your recommendation?

e) KNN with mixed-unit predictors *(2 %)*

The team also fits a K-nearest-neighbours classifier using Euclidean distance on the 7 raw predictors.

(i) *(1 %)* Describe in one or two short sentences how KNN classifies a new test point x_0 (no math required).

(ii) *(1 %)* The raw predictors are on very different scales (`age` in years, `tobacco` in kg-years, `ldl` in mmol/L, `adiposity` as a unitless score). Briefly state, in one sentence, *why* fitting KNN on these raw predictors is a mistake, and what standard remedy resolves it.

End of exam. Total: $10 + 28 + 16 + 20 + 26 = 100$ points.