TMA4268 Statistical Learning V2026

Mock Exam 7 (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 %.

A central theme of this course is _____ (1) (*interpolation / regularization / model averaging / the Bayes rule*): any modification of a learning algorithm that aims to reduce _____ (2) (*training error / the intercept / generalization error / the residual variance*) without necessarily reducing training error.

In the linear regression module we discussed what happens when two predictors are highly correlated: the matrix $\mathbf{X}^\top\mathbf{X}$ becomes nearly singular, the variances of the estimated coefficients explode, and individual t-tests can become misleading. This phenomenon is called _____ (3) (*heteroscedasticity / leverage / collinearity / overfitting*).

For training neural networks the standard optimizer is _____ (4) (*coordinate descent / Newton–Raphson / the EM algorithm / mini-batch stochastic gradient descent*), and the gradients themselves are computed efficiently by an algorithm that reuses intermediates from the forward pass, namely _____ (5) (*boosting / backpropagation / bootstrapping / cross-validation*).

A neural network with many more weights than observations needs regularization to generalize well. One classification-specific trick is _____(6) (*batch normalization / Adam / label smoothing / the universal approximation theorem*), in which the hard one-hot targets are replaced by slightly softened versions.

For tree ensembles, growing many *independent* trees in parallel and averaging them describes _____(7) (*gradient boosting / AdaBoost / cost-complexity pruning / bagging and random forests*), whereas growing many small trees *sequentially*, each one a small correction to the previous ensemble's predictions, describes _____(8) (*bagging / boosting / stacking / the bootstrap*).

When estimating the standard error of a complicated statistic for which no clean closed-form sampling distribution is available, the standard nonparametric tool in this course is the _____(9) (*F-test / validation-set approach / Mallows' C_p / bootstrap*).

Finally, when we want both to *select* a hyperparameter and to *honestly assess* the resulting model on the same data set, the recommended procedure is _____(10) (*the validation-set approach / nested cross-validation / leave-one-out CV / 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 and double descent *(3 %)*

Mark each statement true or false.

(i) In the expected-squared-test-error decomposition $\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Bias}^2 + \text{Var} + \sigma^2$, the irreducible term σ^2 can be lowered by choosing a more flexible model.

(ii) In a regression with a heavily noisy response (large σ^2), making the model class more flexible is generally counter-productive because the extra flexibility "chases" noise.

(iii) In the over-parameterized regime where $p \gg n$ and many fits achieve zero training error, mini-batch SGD picks (approximately) the *minimum-norm* interpolator, which can generalize well — the so-called "benign overfitting" or double-descent phenomenon.

(iv) The lecturer's preferred name for what students usually call "the bias–variance trade-off" is "avoid bad overfitting", because in over-parameterized regimes one can drive *both* bias and variance down at once.

b) Cross-validation and nested CV *(4 %)*

(i) *(1 %)* True or false: "Validation-set approach with a 50/50 train/test split is the same as 2-fold cross-validation."

(ii) *(1 %)* True or false: "Compared to 5- or 10-fold CV, LOOCV has the lowest bias as an estimator of test error, but typically higher variance, because the n leave-one-out training sets overlap almost completely."

(iii) *(1 %)* A genomicist has $p = 5{,}000$ predictors and $n = 100$ random labels (truly no signal, Bayes error is 50%). She first ranks all 5,000 predictors by their marginal correlation with y, keeps the top 25, then runs 10-fold CV on a logistic regression of y on those 25. She reports CV-misclassification near 0% and concludes she has built an excellent classifier. True or false: "her conclusion is justified."

(iv) *(1 %)* True or false: "In nested cross-validation, the *inner* folds are used for model selection (hyperparameter tuning) and the *outer* folds for performance assessment."

c) Mini-batch SGD, dropout, label smoothing, early stopping *(4 %)*

(i) *(1 %)* True or false: "In mini-batch SGD, the mini-batch gradient is an *unbiased* estimator of the full-batch gradient: it has the same expectation, but larger variance."

(ii) *(1 %)* True or false: "A common course-recommendation for the dropout rate is in the range 0.2–0.5, with 0.2 a frequent default; dropout is applied during *training* only, and is turned off (or absorbed by rescaling) at test time."

(iii) *(1 %)* True or false: "Label smoothing replaces the hard one-hot target vector $(0, \dots, 0, 1, 0, \dots, 0)$ by a softened version such as $(\varepsilon/(C-1), \dots, 1-\varepsilon, \dots)$ and is motivated, in part, by the possibility that the training labels themselves are imperfect."

(iv) *(1 %)* True or false: "Early stopping is a regularization technique that monitors the *training* error during gradient descent and returns the network parameters from the epoch at which it first stops decreasing, motivated by the observation that overfitting begins precisely when the training loss flattens out."

d) Boosting — AdaBoost and gradient boosting *(4 %)*

(i) *(1 %)* True or false: "In gradient boosting with squared-error loss, fitting the next tree to the current residuals is equivalent to fitting a tree to the *negative gradient* of the loss with respect to the current ensemble's predictions."

(ii) *(1 %)* True or false: "Boosting reduces *variance* relative to a single deep tree because each of its constituent trees is fit on an independent bootstrap replicate of the training data and the resulting predictions are averaged, in the same fashion as in bagging and random forests."

(iii) *(1 %)* True or false: "In gradient boosting, halving the shrinkage / learning rate ν approximately doubles the number of trees M required for the ensemble to fit well, so M and ν are tuned jointly."

(iv) *(1 %)* True or false: "In AdaBoost, the classifier weight is $\alpha_m = \log\big((1-\mathrm{err}_m)/\mathrm{err}_m\big)$, so a base classifier with weighted error *above* 0.5 would receive a *negative* α_m in the final vote."

e) Collinearity *(4 %)*

A linear regression of y on $p = 7$ continuous predictors is fit on $n = 200$ observations. Two of the predictors, x_2 and x_3, are almost perfectly correlated ($\mathrm{cor}(x_2, x_3) \approx 0.99$); the other predictors are roughly uncorrelated.

Mark each statement true or false.

(i) The coefficient estimates $\hat{\beta}_2$ and $\hat{\beta}_3$ are likely to have large standard errors and to be each individually *insignificant* by their t-tests, while their joint contribution to the model may still be highly significant.

(ii) The *predicted values* $\hat{y}$ are likely to be much more unstable across resamples than the individual coefficients $\hat{\beta}_2$ and $\hat{\beta}_3$ are.

(iii) Standardizing x_2 and x_3 to mean 0 and variance 1 before fitting OLS makes the matrix $\mathbf{X}^\top\mathbf{X}$ *exactly* invertible and fixes the collinearity problem.

(iv) Adding an L2 (ridge) penalty $\lambda\sum_j \beta_j^2$ to the residual sum of squares makes $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ invertible for any $\lambda > 0$ and tends, in the limit of strong collinearity, to make the two collinear coefficients converge toward each other (sharing the effect roughly equally) rather than letting one of them blow up.

f) Bootstrap for the standard error of a derived quantity *(3 %)*

You have fit a logistic regression of `default` on `balance`, `income`, and `student_status`. From it you compute the predicted probability of default $\hat{p}(x_0)$ for a specific new customer with covariates x_0 = (`balance` = 2,000, `income` = 40,000, `student` = yes). The closed-form $\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$ formula does *not* give you a direct standard error for $\hat{p}(x_0)$ because the sigmoid step in $\hat{p} = \sigma(x_0^\top\hat{\beta})$ makes the propagation of variance nonlinear.

(i) *(2 %)* Write pseudocode (or math) for a bootstrap procedure that returns an estimate $\widehat{\mathrm{SE}}_{\mathrm{boot}}(\hat{p}(x_0))$ and a 95% percentile confidence interval for $\hat{p}(x_0)$. State your choice of the number of bootstrap resamples B and justify it in one short sentence.

(ii) *(1 %)* True or false: "If $\hat{p}(x_0)$ is biased as an estimator of the true conditional probability $\Pr(Y = 1 \mid X = x_0)$, the bootstrap standard error and the percentile CI computed in (i) automatically *correct* for that bias."

g) Logistic regression coefficients *(3 %)*

(i) *(1 %)* A logistic regression of the binary outcome `spam` on a continuous predictor `n_links` (the number of links in an e-mail) has fitted coefficient $\hat{\beta}_{\texttt{n\_links}} = 0.18$. By what factor do the odds of `spam` change between two otherwise identical messages whose `n_links` values differ by 5? (One numeric value, rounded to two decimals.)

(ii) *(1 %)* For another e-mail the fitted linear predictor equals $\hat{\eta} = 1.2$. What is the predicted probability of `spam`? (One numeric value, rounded to three decimals.)

(iii) *(1 %)* True or false: "Adding $\lambda \sum_j |\beta_j|$ as an L1 penalty to the logistic-regression log-likelihood, with λ chosen by cross-validation, can drive some $\hat{\beta}_j$ exactly to zero, performing automatic variable selection on top of logistic regression."

h) Random forests *(1 %)*

Mark each statement true or false. *(0.5 % per statement.)*

(i) In a random forest, the parameter `mtry` (number of predictors sampled at each split) controls the *correlation* between trees: smaller `mtry` tends to give less correlated trees and therefore more variance reduction from averaging.

(ii) The number of trees B in a random forest is a tuning parameter selected by cross-validation, because the test error of a random forest follows a U-shaped curve in B and rises again once B becomes too large.

i) Principal component analysis *(2 %)*

You perform PCA on a dataset with four **standardized** variables X_1, X_2, X_3, X_4 and obtain eigenvalues $\lambda_1 = 1.80$, $\lambda_2 = 1.10$, $\lambda_3 = 0.70$, $\lambda_4 = 0.40$. The first principal-component loading vector (entries rounded to two decimals) is

$$\phi_1 = (\,0.60,\ 0.50,\ 0.40,\ 0.50\,)^\top.$$

A new observation has standardized values $x^* = (\,1,\ -1,\ 0.5,\ 0\,)^\top$.

(i) *(1 %)* How many principal components must be retained to capture at least 85% of the total variance? Show the cumulative-PVE calculation.

(ii) *(1 %)* Compute the score z_1^* of the new observation on PC1.

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

a) The mathy one — the bias–variance decomposition *(8 %)*

Let $y_0 = f(x_0) + \varepsilon$ where ε is a zero-mean noise term with $\mathrm{Var}(\varepsilon) = \sigma^2$, independent of the training set. Let $\hat{f}$ be a fixed estimator trained on a random training set $\mathcal{D}$, and let $\hat{f}(x_0)$ denote its prediction at the fixed query point x_0.

(i) *(2 %)* State the two key independence / zero-mean assumptions you will need in the derivation below: one about ε relative to the training set $\mathcal{D}$, and one about $\mathbb{E}[\varepsilon]$. Define explicitly what randomness the outer expectation $\mathbb{E}[\,\cdot\,]$ in $\mathbb{E}[(y_0 - \hat{f}(x_0))^2]$ is taken over.

(ii) *(4 %)* Starting from

$$\mathbb{E}\big[(y_0 - \hat{f}(x_0))^2\big] \;=\; \mathbb{E}\big[(f(x_0) + \varepsilon - \hat{f}(x_0))^2\big],$$

derive the decomposition

$$\boxed{\mathbb{E}\big[(y_0 - \hat{f}(x_0))^2\big] \;=\; \mathrm{Bias}^2\big[\hat{f}(x_0)\big] \;+\; \mathrm{Var}\big[\hat{f}(x_0)\big] \;+\; \sigma^2.}$$

Show the algebraic steps clearly. In particular: (a) explain why a cross-term involving ε vanishes, (b) use the add-and-subtract trick $\hat{f}(x_0) - f(x_0) = (\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]) + (\mathbb{E}[\hat{f}(x_0)] - f(x_0))$ to separate the bias and the variance, and (c) identify each of the three resulting terms.

(iii) *(1 %)* The decomposition above is an exact *identity* that holds for every estimator $\hat{f}$. Briefly explain why it is therefore compatible *both* with the classical U-shaped test-MSE curve *and* with the double-descent / benign-overfitting curve seen in highly over-parameterized models.

(iv) *(1 %)* Briefly state one practical implication: explain in one sentence why, in this course, a method that introduces a small extra bias to obtain a large reduction in variance (e.g. ridge, lasso, dropout, or boosting at well-chosen M) can have *lower* test MSE than ordinary least squares.

b) Pseudocode — k-fold and nested cross-validation *(4 %)*

(i) *(2 %)* Write pseudocode for ordinary k-fold cross-validation that returns the CV-error of a specified model class $\mathcal{M}(\theta)$ at a specified hyperparameter value θ, evaluated on a training set $\{(x_i, y_i)\}_{i=1}^n$. Include: the random partition into folds, the inner fit-and-evaluate loop, the per-fold error definition, and the aggregation. Then state, in one or two sentences, how you would use this procedure to *select* a hyperparameter from a grid $\{\theta_1, \ldots, \theta_T\}$ and what you would do with the chosen hyperparameter afterwards.

(ii) *(2 %)* Now suppose you want *both* to select θ *and* to obtain an honest estimate of the test error of the selected model, using only the training set. Sketch pseudocode for **nested cross-validation** that does this in a single run, clearly distinguishing the *inner* and *outer* loops. State, in one short sentence, why a naive single-layer CV approach in which one reports $\min_\theta \mathrm{CV}_k(\theta)$ as the final test-error estimate is *biased downward*.

c) AdaBoost by hand — one round *(4 %)*

A small training set of $N = 5$ observations $(i, y_i, w_i^{(1)})$ is given, with labels coded ± 1 and initial weights $w_i^{(1)} = 1/N$:

i	y_i	$w_i^{(1)}$	Prediction $G_1(x_i)$
1	$+1$	0.20	$+1$ (correct)
2	$+1$	0.20	-1 (wrong)
3	-1	0.20	-1 (correct)
4	-1	0.20	-1 (correct)
5	$+1$	0.20	-1 (wrong)

A weak classifier $G_1(x)$ has been fit to these weighted data and its per-observation predictions are shown in the rightmost column.

(i) *(1 %)* Compute the weighted misclassification error of G_1,

$$\mathrm{err}_1 = \frac{\sum_i w_i^{(1)} \cdot \mathbb{1}[y_i \neq G_1(x_i)]}{\sum_i w_i^{(1)}}.$$

(ii) *(1 %)* Compute the classifier weight $\alpha_1 = \log\big((1 - \mathrm{err}_1)/\mathrm{err}_1\big)$ (give the answer to three decimals; you may use $\log = \ln$).

(iii) *(2 %)* Apply the AdaBoost weight update

$$w_i^{(2)} \;\propto\; w_i^{(1)} \cdot \exp\big(\alpha_1 \cdot \mathbb{1}[y_i \neq G_1(x_i)]\big),$$

and report the renormalized weights $w_i^{(2)}$ (with $\sum_i w_i^{(2)} = 1$), rounded to three decimals. In one sentence, comment on which observations have gained weight and why this is the entire point of AdaBoost.

Problem 4 (20 %) — Data analysis: wine quality (regression)

A wine merchant has $n = 1{,}200$ red wines, each scored by sommeliers on a continuous `quality` index (response, range 3–8). The available predictors are the eight chemical measurements

- `alcohol` — alcohol content (% vol.);
- `volatile_acidity` (g/L);
- `sulphates` (g/L);
- `pH`;
- `density` (g/cm^3);
- `residual_sugar` (g/L);
- `free_so2` — free sulfur dioxide (mg/L);
- `total_so2` — total sulfur dioxide (mg/L).

Important pairwise correlation among the predictors (computed on the training set, after standardization):

$$\text{cor}(\texttt{free\_so2}, \texttt{total\_so2}) \approx 0.99.$$

All other absolute correlations among predictors are below 0.55.

The data are split 800/400 into training and test sets. *All continuous predictors are standardized to mean* 0*, variance* 1 *before fitting any of the models below.*

a) OLS with a polynomial term and an interaction *(8 %)*

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

$$\texttt{quality} \sim \texttt{alcohol} + I(\texttt{alcohol}^2) + \texttt{volatile\_acidity} + \texttt{sulphates} + \texttt{pH} + \texttt{density} + \texttt{residual\_sugar} + \texttt{free\_s}$$

The fitted output is:

	Estimate	**Std. Error**	**t-value**	**Pr($>\lvert t\rvert$)**
(Intercept)	5.62	0.030	187.3	< 0.001
`alcohol`	0.420	0.045	9.33	< 0.001
$I($`alcohol`$^2)$	-0.110	0.030	-3.67	< 0.001
`volatile_acidity`	-0.305	0.034	-8.97	< 0.001
`sulphates`	0.180	0.031	5.81	< 0.001
`pH`	-0.090	0.040	-2.25	0.025
`density`	-0.060	0.062	-0.97	0.334
`residual_sugar`	0.022	0.038	0.58	0.563
`free_so2`	0.260	0.180	1.44	0.149
`total_so2`	-0.330	0.190	-1.74	0.083
`alcohol:volatile_acidity`	-0.140	0.034	-4.12	< 0.001

Residual standard error: 0.655 on 789 degrees of freedom. Multiple $R^2 = 0.371$, Adjusted $R^2 = 0.363$. Training MSE $= 0.425$, Test MSE $= 0.453$.

(i) *(1 %)* How many parameters (including the intercept) does this model estimate? Verify the count against the printed residual degrees of freedom.

(ii) *(2 %)* Two predictors, `free_so2` and `total_so2`, have large estimates (+0.26 and −0.33) but each appears individually *insignificant* ($p = 0.149$ and 0.083). Their standard errors (0.180 and 0.190) are about 5× larger than those of the other predictors. Identify, in one or two sentences, what is going on, and state *precisely* which two pieces of information together let you diagnose this (no need to compute a VIF).

(iii) *(2 %)* For a wine in which both `alcohol` and `volatile_acidity` are at their training-set means (so their *standardized* values are both 0), and all other predictors are also at their means, compute the predicted `quality`. Then compute the predicted `quality` for the same wine if its *standardized* `alcohol` is increased from 0 to +1 while *standardized* `volatile_acidity` stays at 0. (Two numeric answers; remember to include both the `alcohol`^{2} term and the interaction term.)

(iv) *(2 %)* Now repeat the second calculation in (iii) but with *standardized* `volatile_acidity` fixed at +1 (a wine with notably more volatile acidity than average) instead of at 0, while *standardized* `alcohol` again moves from 0 to +1. Report (a) the implied change in predicted `quality`, and (b) one short sentence interpreting the sign and magnitude of the interaction term in plain English.

(v) *(1 %)* A classmate writes: "`density` has $p = 0.334$, so density does not affect wine quality, and we should drop it." Give one short sentence rebutting this reading.

b) Diagnosing collinearity and choosing a fix *(4 %)*

Refit the OLS model after *dropping* `total_so2` (keeping `free_so2` and all other predictors). The new fitted coefficients and SEs for the two SO_2-related rows are:

	Estimate	**Std. Error**	**t-value**	**Pr(>|t|)**
`free_so2` (in reduced model)	−0.072	0.040	−1.80	0.072

Other coefficients are essentially unchanged from part (a). Training MSE = 0.428, Test MSE = 0.451.

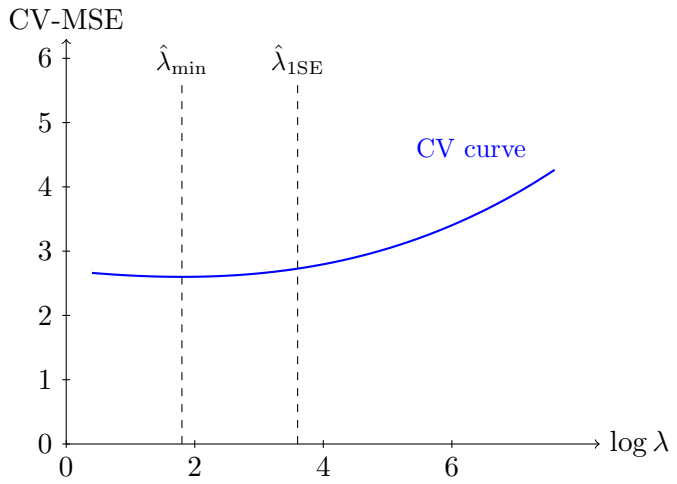
(i) *(2 %)* The estimate of $\hat{\beta}_{\texttt{free\_so2}}$ has swung from +0.26 (in the original model) to −0.072 (in the reduced model), and its SE has shrunk from 0.180 to 0.040. Briefly explain, in one or two sentences, why this is the canonical fingerprint of strong collinearity between `free_so2` and `total_so2` in the original model, by reference to what happens to $(\mathbf{X}^\top\mathbf{X})^{-1}$ when two columns of $\mathbf{X}$ are nearly proportional.

(ii) *(1 %)* The training and test MSE are essentially the same between the full model in (a) and the reduced model in (b). What does this tell you about whether the collinearity was hurting *prediction* (as opposed to *interpretation*)?

(iii) *(1 %)* The course staff lists three legitimate alternative fixes for collinearity: (1) *drop a variable*, (2) *combine the collinear predictors* (e.g. replace them by a single "`total_so2`" channel), and (3) *regularize with ridge / use principal components regression.* In one sentence, briefly state when option (2) is preferable to option (1).

c) Ridge regression and the one-SE rule *(5 %)*

The full 10-predictor design (now including both `free_so2` and `total_so2`, plus the polynomial and the interaction) is fed into ridge regression. The 10-fold CV-MSE profile is:



Test MSEs on the same 400-observation test set:

	Test MSE	**No. nonzero coef's**
OLS (full 11-coefficient model from part (a))	0.453	11
Ridge at $\hat{\lambda}_{\min}$	0.430	11
Ridge at $\hat{\lambda}_{1\mathrm{SE}}$	0.435	11
Lasso at $\hat{\lambda}^{\mathrm{lasso}}_{1\mathrm{SE}}$	0.439	6

(i) *(1 %)* Write down the ridge objective function explicitly, being careful about whether the intercept is penalised. Then state, in one sentence, why the predictors were standardised *before* this fit.

(ii) *(2 %)* State the one-standard-error rule precisely. Then give one bias–variance reason why ridge at $\hat{\lambda}_{\min}$ improves on OLS by about 0.023 in test MSE on *this* dataset, even though OLS is unbiased.

(iii) *(1 %)* Among the four rows in the table above, which would you recommend to a colleague who values *interpretability* and is comfortable trading a small amount of predictive accuracy for it? Justify in one short sentence.

(iv) *(1 %)* Briefly explain why the *lasso* CV curve would, in this dataset, plausibly produce a model that *zeroes out one of* `free_so2` or `total_so2` while ridge does not.

d) Gradient boosting for comparison *(3 %)*

The course staff also fits a gradient-boosted tree model (interaction depth $d = 3$, shrinkage $\nu = 0.05$, M chosen by 10-fold CV; $M^\star \approx 1{,}400$). On the same 400-observation test set it achieves test MSE $= 0.382$.

(i) *(1 %)* Compare the test MSEs — boosting 0.382, ridge 0.430, OLS 0.453 — in one short sentence: what does the gap between boosting and the linear methods suggest about the structure of the relationship between predictors and `quality`?

(ii) *(1 %)* A junior colleague proposes *quadrupling* M to 5,600 "just to be safe." Briefly say why this can hurt *boosting*, even though it would not hurt a random forest.

(iii) *(1 %)* You also want some interpretability from the boosted model. Name one diagnostic plot and one summary statistic you can extract from it; for each, state in one short sentence what question it answers and what question it does *not* answer.

Problem 5 (26 %) — Data analysis: customer churn (classification)

A telecom company has $n = 5{,}000$ customers; the binary response `churn` equals 1 if the customer cancelled their subscription within the next six months. The available predictors are:

- `tenure` — months as a customer (continuous);
- `monthly_charges` — current monthly bill, USD (continuous);
- `senior` — binary 0/1 (senior citizen);
- `contract` — categorical, 3 levels: *month-to-month* (reference), *1-year*, *2-year*;
- `tech_support` — binary 0/1 (subscribed to tech support add-on);
- `online_security` — binary 0/1.

Data are split 70/30 into training (3,500) and test (1,500) sets. Among the 1,500 test customers, 420 truly churned and 1,080 did not.

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

A logistic regression model is fit on the training set with all six predictors above, plus an interaction between `tenure` and `contract`:

$$\operatorname{logit}\big(\Pr(\texttt{churn} = 1 \mid X)\big) \;=\; \beta_0 + \beta_1 \texttt{tenure} + \beta_2 \texttt{monthly\_charges} + \beta_3 \texttt{senior} + \boldsymbol{\beta}_{\texttt{contract}}^{\top} \texttt{contract} + \boldsymbol{\beta}_{\texttt{ten:contract}}^{\top}$$

The fitted output is:

	Estimate	**Std. Error**	**z-value**	**Pr($>\lvert z\rvert$)**
(Intercept)	−0.40	0.18	−2.22	0.026
`tenure`	−0.060	0.005	−12.0	< 0.001
`monthly_charges`	0.018	0.003	6.00	< 0.001
`senior`	0.40	0.12	3.33	< 0.001
`contract_1year`	−1.10	0.20	−5.50	< 0.001
`contract_2year`	−2.00	0.25	−8.00	< 0.001
`tenure:contract_1year`	0.025	0.008	3.13	0.002
`tenure:contract_2year`	0.040	0.010	4.00	< 0.001
`tech_support`	−0.55	0.13	−4.23	< 0.001
`online_security`	−0.45	0.13	−3.46	< 0.001

(Reference levels: `contract = month-to-month`; `senior = 0`; `tech_support = 0`; `online_security = 0`.)

(i) *(2 %)* For each additional month of `tenure`, by what factor do the odds of churn change for a customer on (a) a *month-to-month* contract, (b) a *1-year* contract, and (c) a *2-year* contract? Give all three odds-multiplication factors, rounded to three decimals, and state the encoding assumption you have made explicit.

(ii) *(1 %)* Briefly explain in one short sentence why $\hat{\beta}_{\texttt{tenure:contract\_2year}} = +0.040$ has a *positive* sign even though $\hat{\beta}_{\texttt{tenure}} = -0.060$ is negative. (Don't just restate that "it's an interaction.")

(iii) *(3 %)* Consider a specific customer: `tenure` = 12 months, `monthly_charges` = 85 USD, `senior` = 0, `contract` = *month-to-month*, `tech_support` = 0, `online_security` = 0. Compute the predicted probability $\hat{p}$ of churn for this customer. Show the linear predictor $\hat{\eta}$ and the sigmoid step.

(iv) *(2 %)* Repeat the calculation in (iii) for an *otherwise identical* customer who is on a *2-year* contract. *Don't forget the interaction.* Then comment in one sentence on why the marketing team might use this comparison to argue for incentivising long contracts.

(v) *(2 %)* A classmate writes: "`senior` has $p < 0.001$, so we can conclude that being a senior *causes* customers to churn more often." Briefly state one reason that conclusion is unwarranted from this fitted model (the answer should appeal to the kind of model the prof flagged repeatedly — "fancy correlations, not causal").

b) LDA vs. logistic regression on the same data *(5 %)*

LDA is also fit on the same training set, using the same six predictors. On the test set:

LDA confusion matrix

	Predicted: churn	**Predicted: no churn**
Actual: churn	180	240
Actual: no churn	130	950

For the logistic regression of part (a), at threshold $\hat{p} = 0.5$, the corresponding test-set confusion matrix is:

	Predicted: churn	**Predicted: no churn**
Actual: churn	210	210
Actual: no churn	135	945

(i) *(2 %)* Compute the sensitivity, specificity, and test error rate of the LDA classifier and of the logistic-regression classifier (both at their default thresholds). Six numeric answers, two decimals each.

(ii) *(1 %)* LDA and logistic regression are both linear classifiers in this problem. State the one substantive *modelling* difference between them; if a student were to add a categorical variable with many levels like `contract` into the input vector, which of the two methods' assumptions is the more obviously violated, and why?

(iii) *(2 %)* A clinician colleague hands you the same problem and proposes *QDA* instead of LDA. Mention one situation in which QDA would be expected to outperform LDA on a problem like this, and one situation in which it would be expected to do *worse*.

c) A boosting classifier *(8 %)*

The course staff fits an AdaBoost ensemble of $M = 500$ shallow trees (depth $d = 2$) on the training data, and separately a gradient-boosted classifier with $M = 800$ trees, depth $d = 3$, and shrinkage $\nu = 0.05$, M chosen by 10-fold CV. The gradient-boosting test-set confusion matrix at the default classification threshold is:

Gradient boosting confusion matrix (default threshold)

	Predicted: churn	**Predicted: no churn**
Actual: churn	260	160
Actual: no churn	145	935

(i) *(2 %)* Briefly justify the choice of each of the gradient-boosting hyperparameters ($M = 800$, $d = 3$, $\nu = 0.05$) in one short sentence per parameter. "Sufficiently many" is not a sufficient justification for M; tie each choice to the role the parameter plays in the bias–variance behaviour of the ensemble.

(ii) *(2 %)* Sketch the conceptual *difference* between AdaBoost and gradient boosting in two short bullet points, each at most one sentence: one bullet for the AdaBoost mechanism (sample re-weighting + ± 1 vote), one for the gradient-boosting mechanism (fit next tree to the negative gradient of the loss). Then state, in one extra sentence, in what sense AdaBoost can be viewed as a special case of gradient boosting.

(iii) *(2 %)* Compute the sensitivity, specificity, and test error rate of the gradient-boosting classifier (three numeric answers, two decimals). Briefly compare it to the logistic regression from part (b), one short sentence, telling the marketing team which model you would recommend deploying *if* the cost of failing to identify a churner (a false negative) is substantially larger than the cost of unnecessarily targeting a loyal customer.

(iv) *(2 %)* The same boosted model is used to produce a *variable-importance plot*; the top three predictors by permutation importance are `tenure`, `contract`, and `monthly_charges`. State (a) what a permutation variable-importance plot *is* mechanically (one short sentence), and (b) one specific question that this plot *cannot* answer about `tenure` that a logistic-regression coefficient *can*.

d) Class imbalance and the metric debate *(3 %)*

In the full population, only 28% of customers churn within the six-month window. A junior colleague suggests using *accuracy* (i.e. $1-$ test error rate) as the sole criterion for picking between the three classifiers above.

(i) *(1 %)* A trivial classifier that always predicts "no churn" would have test error rate of approximately what value on this data set? Justify in one sentence.

(ii) *(2 %)* Comment in two or three sentences on whether accuracy is the right criterion here. In particular: (a) name one alternative metric or pair of metrics that you would privilege, and (b) explain in one sentence *which decision* that metric / pair of metrics would make easier than raw accuracy would (e.g. choosing between two classifiers that have nearly identical accuracy but very different false-negative rates).

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