TMA4268 Statistical Learning V2026

Mock Exam 8 (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 a calculator slip spoils 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 classification we can split methods into two paradigms. Methods that model the class-conditional density $f_k(x) = \Pr(X \mid Y = k)$ and the prior π_k, then flip with Bayes' rule to obtain the posterior, are called _____ (1) (*discriminative / generative / semi-supervised / kernel*) methods. Linear discriminant analysis is the canonical example. The decision boundary between two classes under LDA is obtained by setting the two discriminant scores $\delta_k(x)$ equal, and is _____ (2) (*quadratic / piecewise constant / linear / circular*) in x, because the term $x^\top \Sigma^{-1} x$ cancels (it does not depend on k when the covariance is _____ (3) (*class-specific / diagonal / block-diagonal / pooled*)).

When fitting a multiple linear regression with two predictors that are nearly proportional, the matrix $\mathbf{X}^\top \mathbf{X}$ becomes nearly singular and the variance of $\hat{\boldsymbol{\beta}}$ explodes. This phenomenon is called _____ (4) (*heteroscedasticity / collinearity / leverage / influence*). The remedy proposed in the lectures that *adds a penalty* $\lambda \sum_j \beta_j^2$ to the residual sum of squares without

setting any coefficient exactly to zero is called _____ (5) (*the lasso / forward selection / ridge regression / partial least squares*).

For tree ensembles, sampling, at each candidate split, a random subset of m predictors of size $m < p$ is the trick that distinguishes _____ (6) (*AdaBoost / gradient boosting / cost-complexity pruning / random forests*) from plain bagging. For each tree built on a bootstrap sample, roughly 37% of the original observations are not in the sample, and the predictions on those left-out observations are aggregated into the so-called _____ (7) (*validation error / out-of-bag error / training error / Bayes error*) estimate of test error.

In neural networks, the central optimization recipe is mini-batch stochastic gradient descent. The gradients with respect to all weights are computed efficiently by reusing intermediate values from the forward pass through repeated application of the chain rule; this algorithm is called _____ (8) (*boosting / the EM algorithm / backpropagation / Newton-Raphson*). To prevent the network from becoming over-confident in its class predictions, especially when labels may themselves contain errors, one can replace the hard one-hot target by a slightly softened one; this technique is called _____ (9) (*dropout / early stopping / batch normalization / label smoothing*).

Finally, the workhorse for estimating the standard error of a complicated statistic when no closed-form sampling distribution is available is to resample n observations with replacement from the original sample many times, recompute the statistic on each resample, and take the sample standard deviation of the resulting values. This procedure is called the _____ (10) (*F-test / bootstrap / Bonferroni correction / Newton step*).

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

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

a) Bias–variance, applied to a shrinkage estimator *(3 %)*

Let $X_1, \ldots, X_n$ be i.i.d. with mean μ and variance σ^2, and let $\bar{X}$ be the sample mean. Consider the shrunken estimator $\hat{\mu}_c = c\,\bar{X}$ for $c \in [0, 1]$.

(i) *(1 %)* Compute $\text{Bias}^2(\hat{\mu}_c)$ and $\text{Var}(\hat{\mu}_c)$, each as a function of c, μ, σ^2, and n. (Two short formulas.)

(ii) *(1 %)* Mark true or false: "Compared to the unbiased choice $c = 1$, the shrunken estimator with $c < 1$ has a smaller mean squared error $\text{Bias}^2 + \text{Var}$, regardless of μ and σ^2/n."

(iii) *(1 %)* Mark true or false: "In the over-parameterized regime where many fitted models interpolate the training data, the bias–variance decomposition $\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Bias}^2 + \text{Var} + \sigma^2$ no longer holds as an algebraic identity — which is why test error can exhibit a second descent past the interpolation peak."

b) Cross-validation: LOOCV vs k-fold and the OLS shortcut *(4 %)*

(i) *(1 %)* Mark true or false: "Compared to 5-fold or 10-fold CV, leave-one-out CV has *lower* bias as an estimator of test error, but typically *higher* variance, because the n leave-one-out training sets share nearly all of their observations and the per-fold errors are highly correlated."

(ii) *(1 %)* For ordinary least squares, the LOOCV mean squared error has a closed-form expression that needs only one fit. Write it explicitly in terms of the full-data fitted residuals $y_i - \hat{y}_i$ and the diagonal entries h_{ii} of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.

(iii) *(1 %)* Mark true or false: "For data with strong temporal autocorrelation (e.g. daily stock returns), randomly partitioning the observations into $k = 10$ folds gives an unbiased estimate of the test error of the procedure."

(iv) *(1 %)* The standard error of $\text{CV}_{(k)}$, computed as the sample standard deviation of the k per-fold MSEs, is described in the slides as "*strictly speaking, not quite valid.*" Briefly explain in one sentence why.

c) Bootstrap: percentile CI, basic CI, bias correction *(3 %)*

You have an i.i.d. sample $X_1, \ldots, X_n$ and a complicated statistic $\hat{\theta} = T(X_1, \ldots, X_n)$. You draw B bootstrap samples, recompute $\hat{\theta}^*_1, \ldots, \hat{\theta}^*_B$, and let q^*_α denote the α-quantile of those values.

(i) *(1 %)* Write the *percentile* 95% bootstrap confidence interval for θ in terms of $q^*_\cdot$. (One short formula.)

(ii) *(1 %)* Mark each statement true or false. (A) Bootstrap samples are drawn from the original data *without* replacement, so that the same row of $\mathbf{X}$ never appears twice in a resample. (B) The bootstrap distribution of $\hat{\theta}^*$ is centred around $\hat{\theta}$ rather than the unknown θ, so the bootstrap can *not* by itself correct for the bias of $\hat{\theta}$. (C) Typical values of B for a confidence interval are in the range 1,000 to 10,000.

(iii) *(1 %)* You wish to compare the test MSE of *two* models, A and B, fit on the same training set and evaluated on the same test set of size m. You want a bootstrap estimate of the standard error of the difference $\widehat{\Delta} = \mathrm{MSE}_A^{\mathrm{test}} - \mathrm{MSE}_B^{\mathrm{test}}$. Briefly state, in one short sentence, why you should resample the test pairs (x_t, y_t) *jointly* (a so-called *paired* bootstrap) rather than resample the per-model squared errors of A and B independently.

d) Mini-batch SGD, activations, vanishing gradients *(3 %)*

Mark each statement as true or false.

(i) Mini-batch SGD with batch size m gives, on each step, an *unbiased* estimator of the full-batch gradient: the expectation over the batch equals the full-data gradient, but the variance grows when m shrinks.

(ii) A typical practical learning rate for mini-batch SGD on a neural network is $\eta \approx 2$; a learning rate of $\eta = 0.1$ would generally be too small and would cause the loss to make almost no progress per epoch.

(iii) One advantage of the *ReLU* activation $\mathrm{ReLU}(z) = \max(0, z)$ over the classical *sigmoid* $\sigma(z) = 1/(1+e^{-z})$ is that, for large positive z, ReLU's derivative is 1 rather than $\sigma(z)(1-\sigma(z)) \approx 0$ — which helps prevent gradients from shrinking towards zero as they back-propagate through deep networks.

e) Boosting: AdaBoost, gradient boosting, XGBoost *(4 %)*

(i) *(1 %)* In the AdaBoost classifier weight $\alpha_m = \log\big((1 - \mathrm{err}_m)/\mathrm{err}_m\big)$, suppose a particular weak learner G_{m^*} has weighted error $\mathrm{err}_{m^*} = 0.70$ (worse than random guessing). Compute α_{m^*} (two decimals) and briefly state, in one short sentence, what the negative sign *does* to that classifier's contribution in the final vote $G(x) = \mathrm{sign}\big(\sum_m \alpha_m G_m(x)\big)$.

(ii) *(1 %)* Mark 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."

(iii) *(1 %)* Mark true or false: "In gradient boosting, halving the shrinkage parameter ν from 0.10 to 0.05 roughly *halves* the number of trees M needed for the ensemble to reach a comparable fit, which is why M and ν are tuned jointly rather than independently."

(iv) *(1 %)* Mark true or false: "XGBoost differs from vanilla gradient boosting in that it adds explicit L_1 *and* L_2 regularization on the per-leaf prediction values, and uses a second-order (Newton-like) approximation of the loss when scoring candidate splits — making each split decision a function of both the gradient and the Hessian of the loss."

f) Neural network regularization *(3 %)*

(i) *(1 %)* Mark each as true or false. (A) Dropout is applied only during training; at test time the units are kept on, and the network's outputs are rescaled to account for the missing units. (B) Early stopping returns the network parameters from the epoch at which the *training* loss first stops decreasing. (C) Standard course recommendation for the dropout rate is around 20%, with 50% generally considered too aggressive.

(ii) *(1 %)* In a binary classification network with a sigmoid output, label smoothing with smoothing parameter $\varepsilon = 0.05$ replaces the hard target $(1, 0)$ for class 1 by what soft target (t_1, t_0)? (Two numerics.)

(iii) *(1 %)* Mark true or false: "Adding L_2 weight decay decreases the network's training error."

g) Logistic regression: odds, log-odds, interaction *(3 %)*

(i) *(1 %)* A logistic regression of a binary outcome on a single continuous predictor has fitted log-odds intercept $\hat{\beta}_0 = -2.0$ and slope $\hat{\beta}_1 = 0.40$. At $x_0 = 5$, what is the predicted probability of the event? (Three decimals.)

(ii) *(1 %)* A logistic model contains the predictors `age` (years) and `treatment` (binary 0/1) and the interaction `age:treatment`, with $\hat{\beta}_{\texttt{age}} = 0.05$ and $\hat{\beta}_{\texttt{age:treatment}} = -0.02$. By what factor (three decimals) do the odds of the event multiply for an *untreated* patient when `age` increases by 1 year? Repeat for a *treated* patient.

(iii) *(1 %)* Mark true or false: "If we relabel the binary predictor `treatment` so that the new reference level is what was previously called 'treated' (instead of 'untreated'), then the sign of the coefficient on `treatment` in the new fitted model flips, but the predicted probability of the event for any given observation is unchanged."

h) Discriminant analysis and naive Bayes *(2 %)*

(i) *(1 %)* Mark each as true or false. (A) Under the LDA assumption that the class-conditional densities are Gaussian and share a common covariance $\mathbf{\Sigma}$, the Bayes-optimal decision boundary between any two classes is linear in $\mathbf{x}$. (B) QDA differs from LDA only in that it allows the class-conditional covariances $\mathbf{\Sigma}_k$ to differ between classes; in particular, the term $\mathbf{x}^\top \mathbf{\Sigma}_k^{-1} \mathbf{x}$ in the discriminant survives in QDA but cancels in LDA. (C) Increasing the prior $\hat{\pi}_k$ of class k, while holding all $\hat{\mu}_k$ and $\hat{\Sigma}$ fixed, moves the LDA decision boundary *toward* class k (less area is classified as k).

(ii) *(1 %)* Mark true or false: "Naive Bayes is the special case of LDA in which the within-class covariance matrix $\mathbf{\Sigma}$ is taken to be diagonal — equivalently, the predictors are assumed conditionally independent given the class."

i) Principal component analysis *(3 %)*

You perform PCA on a dataset with $p = 5$ **standardized** variables and obtain eigenvalues

$$\lambda_1 = 2.1, \quad \lambda_2 = 1.4, \quad \lambda_3 = 0.8, \quad \lambda_4 = 0.5, \quad \lambda_5 = 0.2.$$

The loading vector for PC1 (entries rounded to two decimals) is

$$\phi_1 = (\,0.55,\ 0.45,\ -0.35,\ 0.40,\ -0.45\,)^\top.$$

(i) *(1 %)* What is the total variance in the standardized data, and how many principal components are needed to capture at least 80% of that total variance? Show the cumulative sum.

(ii) *(1 %)* A new standardized observation has values $x^* = (\,2,\ 1,\ 0,\ -1,\ 1\,)^\top$. Compute its score z_1^* on PC1. (Two decimals.)

(iii) *(1 %)* Mark true or false: "If we had *not* standardized the variables before running PCA, the leading principal component would be invariant under the change of units, because PCA only operates on the variables' linear combinations."

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

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

Consider a binary classification setting with two classes $\{0, 1\}$, priors π_0, π_1, class-conditional densities $f_k(x) = \Pr(X \mid Y = k)$, $k \in \{0, 1\}$, and Bayes-optimal classifier $\hat{y} = \arg\max_k \pi_k f_k(x)$. Assume that within each class, $X \mid Y = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, where the covariance matrix $\boldsymbol{\Sigma}$ is the same in both classes.

(i) *(3 %)* Starting from $\log\big(\pi_k f_k(\mathbf{x})\big)$, derive the LDA discriminant score

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

In your derivation, be explicit about *which* terms in $\log\big(\pi_k f_k(\mathbf{x})\big)$ you discard and *why* you may discard them, and *which* term is the one that cancels between δ_0 and δ_1 specifically because $\boldsymbol{\Sigma}$ does not depend on k. In one sentence, state what would change if the covariance were instead allowed to depend on k (yielding QDA).

(ii) *(3 %)* Suppose now that $p = 2$ (so $\mathbf{x} = (x_1, x_2)^\top$), with parameters

$$\boldsymbol{\mu}_0 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \qquad \boldsymbol{\mu}_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \qquad \pi_0 = \pi_1 = 0.5.$$

Derive the equation of the LDA decision boundary in the form $a\,x_1 + b\,x_2 + c = 0$ by setting $\delta_0(\mathbf{x}) = \delta_1(\mathbf{x})$ and simplifying. Then solve for x_2 as a function of x_1. Show your algebra step-by-step (the prof himself flubbed this live and has explicitly warned against "doing algebra in public"; we want every line).

(iii) *(1 %)* A new observation has $\mathbf{x}^* = (1, 1)^\top$. Use your boundary equation to decide which class it is assigned to under LDA, and verify your answer by computing $\delta_0(\mathbf{x}^*)$ and $\delta_1(\mathbf{x}^*)$ directly and comparing them.

(iv) *(1 %)* The data scientist now considers using priors $\pi_0 = 0.8$, $\pi_1 = 0.2$ instead of the equal priors above. In one sentence, state in which direction (toward class 0 or toward class 1) the decision boundary moves, and briefly justify by reference to the $\log \pi_k$ term in δ_k.

b) Mini-batch SGD pseudocode and implicit regularization *(4 %)*

You are training a feed-forward neural network with parameter vector $\boldsymbol{\theta}$ on a training set of size N, using cross-entropy loss

$$L(\boldsymbol{\theta}) \;=\; \frac{1}{N}\sum_{i=1}^{N} \ell_i(\boldsymbol{\theta}),$$

mini-batch SGD with batch size m, learning rate η, and E epochs.

(i) *(2 %)* Write *pseudocode* (math, plain English, or programming-language-agnostic notation) for **one epoch** of mini-batch SGD on $\boldsymbol{\theta}$. Be explicit about: (a) how mini-batches of size m are formed at the start of the epoch (you may assume m divides N); (b) how the gradient $\widehat{\nabla L}$ on a single mini-batch is computed; (c) the parameter update rule; (d) what happens when the epoch finishes. You do not need to spell out the backpropagation step internally; assume a gradient oracle is available.

(ii) *(1 %)* Mark each statement true or false. **(A)** The mini-batch gradient is an *unbiased* estimator of the full-batch gradient: $\mathbb{E}_{\text{batch}}[\widehat{\nabla L}] = \nabla L(\boldsymbol{\theta})$. **(B)** Doubling the batch size m from 32 to 64 approximately *doubles* the variance of $\widehat{\nabla L}$, all else equal. **(C)** Smaller mini-batches provide *stronger* implicit regularization, because the noise in $\widehat{\nabla L}$ steers the optimizer away from sharp minima.

(iii) *(1 %)* In one or two short sentences, tie the "implicit regularization" of mini-batch SGD to the bias–variance lens of Problem 2(a): in what sense does this implicit regularization mirror, in the optimization stage, the effect of explicit regularizers such as ridge, lasso, dropout, or early stopping?

c) k-fold CV pseudocode and the one-standard-error rule *(4 %)*

You have a training set $(\mathbf{X}, \mathbf{y})$ of size n, a regression model class $\mathcal{M}$ indexed by a scalar hyperparameter θ (e.g. a ridge λ, a polynomial degree, a k in KNN) on a finite grid $\Theta = \{\theta_1, \ldots, \theta_T\}$, and you want to (a) report a CV-MSE for each $\theta \in \Theta$, (b) report a standard-error estimate $\widehat{\mathrm{SE}}(\mathrm{CV}_k(\theta))$ for each, and (c) pick a value of θ using the one-standard-error rule.

(i) *(2 %)* Write *pseudocode* (math, plain English, or programming-language-agnostic notation) for the procedure that produces, for each $\theta \in \Theta$, the pair $(\mathrm{CV}_k(\theta), \widehat{\mathrm{SE}}(\mathrm{CV}_k(\theta)))$, where $\widehat{\mathrm{SE}}$ is the sample standard deviation of the per-fold mean squared errors. Be explicit about: (a) how the data is partitioned, (b) what is held out and what is fit at each iteration, (c) how the per-fold MSEs are aggregated into $\mathrm{CV}_k(\theta)$, and (d) how $\widehat{\mathrm{SE}}(\mathrm{CV}_k(\theta))$ is defined.

(ii) *(1 %)* State the one-standard-error rule for selecting a value $\hat{\theta}_{1\mathrm{SE}}$ from Θ in one or two short sentences. Make explicit *which direction* of θ is the "simpler" direction for (a) ridge λ, (b) polynomial degree, and (c) k in KNN. (Three short answers.)

(iii) *(1 %)* A statistician fits ridge regression for $\theta = \lambda \in \{0.001, 0.01, 0.1, 1, 10\}$, getting the following 10-fold CV-MSEs and $\widehat{\mathrm{SE}}$'s:

λ	0.001	0.01	0.1	1	10
$\mathrm{CV}_{10}(\lambda)$	0.452	0.430	0.418	0.435	0.510
$\widehat{\mathrm{SE}}(\mathrm{CV}_{10})$	0.025	0.024	0.022	0.023	0.030

Identify $\hat{\lambda}_{\min}$, state the corresponding "one-SE band" explicitly, and give the value of $\hat{\lambda}_{1\mathrm{SE}}$ that the one-standard-error rule recommends. Briefly justify in one sentence.

Problem 4 (22 %) — Data analysis: regression on bike-sharing demand

An urban-mobility researcher records, for each of $n = 730$ days in a city's two-year bike-share programme, the daily `count` of bike rentals (in thousands; range 0.4–8.7). The available predictors are:

- `temp` — daily mean air temperature, in °C (continuous);
- `feel_temp` — daily mean apparent temperature in °C, i.e. "what it feels like"; nearly equal to `temp`, with $\text{cor}(\texttt{temp}, \texttt{feel\_temp}) \approx 0.99$ on the training data;
- `humidity` — daily mean relative humidity, in % (0–100);
- `windspeed` — daily mean wind speed in km/h (continuous);
- `workingday` — binary indicator: 1 if a workday, 0 for weekend or holiday (reference);
- `season` — categorical with 4 levels: *spring* (reference), *summer*, *autumn*, *winter*;
- `weather` — categorical with 3 levels: *clear* (reference), *misty*, *rain_or_snow*.

The data are split 530/200 into training and test sets. The continuous predictors (`temp`, `feel_temp`, `humidity`, `windspeed`) are standardized to mean 0 and standard deviation 1 *before* fitting any of the models below. (So a unit change in one of these predictors corresponds to one standard deviation on the raw scale.)

a) OLS with a polynomial, an interaction, and collinearity *(10 %)*

The course staff fits, on the training set, the linear model

$$\begin{aligned}\texttt{count} \sim\ & \texttt{temp} + I(\texttt{temp}^2) + \texttt{feel\_temp} + \texttt{humidity} + \texttt{windspeed} \\ & + \texttt{workingday} + \texttt{season} + \texttt{weather} \\ & + \texttt{temp:workingday}.\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**t-value**	**Pr($>\lvert t\rvert$)**
(Intercept)	3.20	0.18	17.78	< 0.001
`temp`	2.10	0.85	2.47	0.014
$I($`temp`$^2)$	−0.55	0.10	−5.50	< 0.001
`feel_temp`	−0.80	0.87	−0.92	0.358
`humidity`	−0.35	0.09	−3.89	< 0.001
`windspeed`	−0.20	0.08	−2.50	0.013
`workingday`	0.25	0.14	1.79	0.074
`season_summer`	0.95	0.20	4.75	< 0.001
`season_autumn`	1.30	0.22	5.91	< 0.001
`season_winter`	−0.40	0.21	−1.90	0.058
`weather_misty`	−0.45	0.13	−3.46	< 0.001
`weather_rain_or_snow`	−1.80	0.30	−6.00	< 0.001
`temp:workingday`	0.40	0.12	3.33	< 0.001

Multiple $R^2 = 0.768$, Adjusted $R^2 = 0.762$. Residual standard error: 0.90 on 517 d.f.

Diagnostic note from the analyst: dropping `feel_temp` from this model and refitting leaves the test MSE essentially unchanged (0.94 → 0.93), and the coefficient on `temp` swings from +2.10 (with SE 0.85) to +1.32 (with SE 0.09).

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

(ii) *(2 %)* Identify two numerical symptoms in the regression output that, taken together, point to a problem with the pair (`temp`, `feel_temp`). (a) Name in one word the statistical phenomenon at work. (b) Explain in one short sentence, by reference to $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$, *why* the individual standard errors of `temp` and `feel_temp` blow up to roughly 5–10× those of the other continuous predictors. (c) Using the analyst's diagnostic note above (test MSE essentially unchanged, $\hat{\beta}_{\texttt{temp}}$ swings from +2.10 to +1.32 when `feel_temp` is dropped), comment in one short sentence on whether the problem in (a) is hurting *prediction* (held-out MSE) or hurting *interpretation* (which coefficient does what).

(iii) *(2 %)* The model includes a quadratic $I(\texttt{temp}^2)$ in addition to the linear `temp` term. On a weekend (`workingday`= 0), with all other predictors held fixed, the predicted `count` is a quadratic function of standardized `temp` of the form $\beta_{\texttt{temp}} \cdot \texttt{temp} + \beta_{\texttt{temp}^2} \cdot \texttt{temp}^2 + \text{const}$. Find the value of standardized `temp` at which this predicted `count` is *maximized*, by solving $\partial/\partial\,\texttt{temp} = 0$. (Show your work; one decimal.) Briefly comment in one sentence on why a polynomial in temperature is a reasonable specification for the bike-share count.

(iv) *(2 %)* For a workday (`workingday` = 1) with all other predictors at their reference levels, compute the change in predicted `count` (in thousands of rentals) when standardized `temp` increases by one unit *from* 0 *to* +1, taking the quadratic, the linear, and the interaction terms *all* into account. Repeat the calculation for a weekend (`workingday` = 0). Compare the two numbers and comment in one short sentence on what the sign of the interaction tells you about how temperature drives demand differently on workdays versus weekends.

(v) *(1 %)* A colleague writes: "Since the coefficient on `workingday` is +0.25 with a p-value of 0.074, the workday effect on bike usage is not statistically significant, and we should drop `workingday` from the model." Briefly explain in one or two sentences why this conclusion is wrong, by reference to the principle of hierarchy and to the fact that the model also contains the interaction term `temp:workingday`.

(vi) *(1 %)* A residual-versus-fitted plot from this fitted model shows clear "fanning out": the spread of residuals grows with the fitted value. Name the regression assumption that this violates, and state, in one short sentence, one common remedy.

(vii) *(1 %)* A new day has predictor values that lie inside the convex hull of the training data, and you would like to attach an *interval* to your prediction. State briefly which is wider, a 95% confidence interval for $\mathbb{E}[\,\texttt{count} \mid x_0\,]$ or a 95% prediction interval for an individual new `count` at x_0, and *why*.

b) Ridge vs. lasso under collinearity *(5 %)*

The same standardized predictors as in part (a), including the polynomial and the interaction (so $p = 12$ predictors after dummy coding), are now fed into both ridge regression and the lasso. The penalty parameter λ is chosen by 10-fold CV in each case (the CV curve in each case is roughly U-shaped with a clear minimum). The resulting test-MSE numbers on the held-out 200 days, and the number of non-zero estimated coefficients, are:

	Test MSE	**Nonzero coefs**
OLS (full model from (a))	0.842	13
Ridge at $\hat{\lambda}^{\text{ridge}}_{\text{min}}$	0.795	13
Ridge at $\hat{\lambda}^{\text{ridge}}_{\text{1SE}}$	0.810	13
Lasso at $\hat{\lambda}^{\text{lasso}}_{\text{min}}$	0.802	9
Lasso at $\hat{\lambda}^{\text{lasso}}_{\text{1SE}}$	0.825	6

(i) *(1 %)* Write down the ridge regression objective explicitly. Be careful about whether the intercept is penalized. Then state, in one short sentence, why all continuous predictors were standardized *before* this fit.

(ii) *(2 %)* On this dataset, the OLS-vs-ridge gap is about 0.047 and the ridge-vs-lasso gap (at the minimum-CV λ) is small (0.795 vs. 0.802). On the collinear pair (`temp`, `feel_temp`), however, the *lasso* typically retains *only one* of the two, while *ridge* retains *both*, with each coefficient roughly halved compared to the OLS magnitudes. Explain in two short sentences *why* the two methods behave differently on this pair, by reference to the geometry of the constraint regions $\sum_j \beta_j^2 \le t$ versus $\sum_j |\beta_j| \le t$.

(iii) *(1 %)* The lasso at $\hat{\lambda}^{\text{lasso}}_{\text{1SE}}$ keeps only 6 of 13 predictors. A practitioner argues: "This is the model I want for a public-facing dashboard, even though its test MSE is slightly worse." State the one-standard-error rule precisely and give the one short reason this practitioner's preference is reasonable.

(iv) *(1 %)* A junior analyst proposes "solving the collinearity problem" by running PCA on the predictors first, retaining the first few components, and then regressing `count` on those component scores. Name the resulting procedure in one short term and state, in one sentence, the one structural cost (compared to lasso) that it pays.

c) Gradient boosting and learning-rate / depth tradeoffs *(4 %)*

A gradient-boosted regression-tree ensemble is fit on the training set. Its three principal hyper-parameters are the number of trees M, the interaction depth d, and the shrinkage / learning rate ν. To pick them, a colleague reports the following 10-fold CV-MSE values, evaluated on the same grid for two different choices of (d, ν):

	$M = 200$	$M = 500$	$M = 1500$	$M = 5000$
Shallow, fast: $d = 2$, $\nu = 0.10$	0.66	0.58	0.56	0.61
Deeper, slow: $d = 5$, $\nu = 0.01$	0.92	0.74	0.55	0.54

(i) *(2 %)* For *squared-error* loss, write *pseudocode* for one round of gradient boosting (the move from the iteration-$(m-1)$ ensemble $\hat{f}^{(m-1)}$ to $\hat{f}^{(m)}$). Be explicit about (a) what quantity the new tree T_m is fit to, (b) where the shrinkage parameter ν enters, and (c) what the final ensemble after M rounds looks like. Two or three short lines is fine.

(ii) *(1 %)* Read off the table: which (d, ν, M) triple has the smallest CV-MSE? Briefly comment, in one short sentence, on the trend along the first row ($d = 2$, $\nu = 0.10$) from $M = 1500$ to $M = 5000$: what does the slight *increase* from 0.56 to 0.61 suggest, and why does this trend not appear for a random forest?

(iii) *(1 %)* A junior data-scientist proposes: "Let's just pick the deeper, slower regime ($d = 5$, $\nu = 0.01$) but use $M = 200$ trees because it's faster." Explain in one sentence, by reference to the qualitative coupling between M and ν, why this is a poor configuration on this dataset.

d) GAM with B-splines *(3 %)*

Finally, a generalized additive model is fit on the same training set:

$$\texttt{count} = \beta_0 + s(\texttt{temp},\, \text{df} = 5) + s(\texttt{humidity},\, \text{df} = 4) + \text{bs}(\texttt{windspeed},\, \text{knots} = \{q_{0.25}, q_{0.5}, q_{0.75}\}) + \beta_{\text{wd}} \cdot \texttt{workingda}$$

where $s(\cdot, \text{df} = k)$ is a smoothing spline with k effective degrees of freedom (intercept removed), $\text{bs}(\cdot, \text{knots} = \ldots)$ is a cubic regression spline with the three quartile knots (intercept removed), and $g(\texttt{season})$ and $h(\texttt{weather})$ use dummy coding as in part (a).

(i) *(1 %)* How many degrees of freedom does bs(`windspeed`, knots = $\{q_{0.25}, q_{0.5}, q_{0.75}\}$) alone consume? (Count basis functions; exclude the global intercept.)

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

(iii) *(1 %)* State, in one short sentence, one *structural limitation* of this additive specification compared to the gradient-boosted ensemble of part (c). (Hint: think about which patterns of joint dependence on two predictors the GAM can and cannot represent.)

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

A regional bank records, for $n = 4{,}000$ retail-loan applicants who were granted a loan, a binary response `default` (1 if the loan defaulted within 24 months, 0 otherwise). The available predictors are:

- `annual_income` — annual income, in thousand EUR (continuous);
- `debt_to_income` — debt-to-income ratio at loan origination, in % (continuous, 0–60);
- `credit_score` — credit-bureau score (continuous, 300–850);
- `n_inquiries` — number of credit inquiries in the past 12 months (count, 0–10);
- `employment` — employment status, categorical with 3 levels: *permanent* (reference), *self-employed*, *temporary*;
- `home_ownership` — binary 0/1: 1 if the applicant owns a home, 0 otherwise.

The data are split 70/30 into training (2,800 applicants) and test (1,200 applicants). In the test set, 216 loans truly defaulted and 984 did not. In the full sample, the empirical default rate is approximately 18%.

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

A logistic regression model is fit on the training set:

$$\begin{aligned}\operatorname{logit}\big(\Pr(\texttt{default} = 1 \mid X)\big) &= \beta_0 + \beta_1 \texttt{annual\_income} + \beta_2 \texttt{debt\_to\_income} + \beta_3 \texttt{credit\_score} \\ &\quad + \beta_4 \texttt{n\_inquiries} + \boldsymbol{\beta}_5^{\top} \texttt{employment} + \beta_6 \texttt{home\_ownership} \\ &\quad + \beta_7 \big(\texttt{debt\_to\_income:credit\_score}\big).\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**z-value**	**Pr(>|z|)**
(Intercept)	4.50	1.20	3.75	< 0.001
`annual_income`	−0.012	0.003	−4.00	< 0.001
`debt_to_income`	0.250	0.080	3.13	0.002
`credit_score`	−0.012	0.002	−6.00	< 0.001
`n_inquiries`	0.150	0.040	3.75	< 0.001
`employment_self`	0.450	0.150	3.00	0.003
`employment_temporary`	0.700	0.140	5.00	< 0.001
`home_ownership`	−0.400	0.110	−3.64	< 0.001
`debt_to_income:credit_score`	−0.00030	0.00010	−3.00	0.003

(`annual_income` is in thousand EUR; `debt_to_income` in %; `credit_score` on the 300–850 scale; `n_inquiries` a count. Reference levels: `employment` = *permanent*, `home_ownership` = 0.)

(i) *(2 %)* For an applicant whose `credit_score` is 600, by what factor (three decimals) do the odds of default change for each one-unit increase in `debt_to_income` (i.e. one extra percentage point of debt-to-income), holding all other predictors fixed? Repeat for an applicant whose `credit_score` is 800. State your encoding assumption explicitly.

(ii) *(1 %)* The main-effect coefficient $\hat{\beta}_2 = 0.250$ on `debt_to_income` is sometimes (incorrectly) quoted as "the average effect of a one-percentage-point increase in debt-to-income on the log-odds of default." Briefly explain, in one short sentence, why this reading is misleading *given that the model also contains the interaction* `debt_to_income:credit_score`, and state at what specific value of `credit_score` the coefficient $\hat{\beta}_2$ would actually equal that "average effect."

(iii) *(3 %)* Consider a specific applicant: `annual_income` = 50, `debt_to_income` = 25, `credit_score` = 650, `n_inquiries` = 3, `employment` = *self-employed*, `home_ownership` = 0. Compute the linear predictor $\hat{\eta}$ (showing one short line per term) and the predicted probability $\hat{p}$ of default for this applicant. Round $\hat{p}$ to three decimals.

(iv) *(1 %)* At the default classification threshold $\hat{p} = 0.5$, is this applicant predicted to default? Briefly comment in one short sentence on whether 0.5 is a sensible threshold for this loan-screening task, given that the base default rate in the population is $\approx 18\%$.

(v) *(1 %)* A classmate writes: "$\hat{\beta}_{\texttt{credit\_score}} = -0.012$ with $p < 0.001$, so we can confidently say that *raising* a customer's credit score *causes* the probability of default to fall." Briefly explain, in one short sentence, why that causal reading is unwarranted from this fitted model, citing the prof's mantra that regression coefficients are "fancy correlations, not causal."

(vi) *(2 %)* On the test set, the logistic model at threshold $\hat{p} = 0.5$ produces the confusion matrix

	Predicted: default	**Predicted: no default**
Actual: default	108	108
Actual: no default	79	905

Compute, to two decimals, the test *sensitivity*, *specificity*, and overall *error rate* of this classifier. Then briefly state, in one short sentence, the direction (up / down) of *sensitivity* and *specificity* if the bank lowered the threshold from 0.5 to 0.3 in order to flag more potential defaulters.

b) AdaBoost from exponential loss *(6 %)*

AdaBoost can be viewed as the forward-stagewise greedy minimizer of the exponential loss

$$L(y, f) = \exp(-y\, f(x)), \qquad y \in \{-1, +1\},$$

where the running prediction is an additive expansion $f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$ with $G_m(x) \in \{-1, +1\}$. At stage m, with the previous fit $f^{(m-1)}$ held fixed, one minimizes the loss *jointly* over the new classifier G_m and its weight α_m:

$$(\alpha_m,\, G_m) \;=\; \arg\min_{\alpha,\, G} \sum_{i=1}^{N} \underbrace{\exp\!\big(-y_i\, f^{(m-1)}(x_i)\big)}_{=:\, w_i^{(m)}} \cdot \exp\!\big(-y_i\, \alpha\, G(x_i)\big).$$

The factor $w_i^{(m)} = \exp(-y_i f^{(m-1)}(x_i))$ depends only on the previous fit, so it acts as a per-observation *weight* at stage m.

(i) *(2 %)* Show that, for fixed $\alpha > 0$, the inner minimization over G reduces to choosing the classifier that minimizes the *weighted* misclassification error

$$G_m \;=\; \arg\min_{G} \sum_{i=1}^{N} w_i^{(m)} \cdot \mathbb{1}[y_i \neq G(x_i)].$$

(Hint: split the sum into correctly classified and misclassified observations using $y_i G(x_i) \in \{-1, +1\}$, and observe that $\sum_i w_i^{(m)}$ does not depend on G.)

(ii) *(2 %)* Given the optimal G_m, derive the closed-form expression for the optimal α_m in terms of the weighted error rate

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

(Differentiate the inner objective in (i) with respect to α, set the derivative to zero, and solve.) Show that the result is

$$\boxed{\alpha_m \;=\; \tfrac{1}{2} \log \frac{1 - \text{err}_m}{\text{err}_m}.}$$

(Note: this is the textbook AdaBoost.M1 formula up to a factor of 2 *in the convention; the prof's slide deck states the unscaled form* $\alpha_m = \log\big((1 - \text{err}_m)/\text{err}_m\big)$*, which differs by absorbing the* $1/2$ *into the final* $\text{sign}(\cdot)$*. Either factor is accepted.)*

(iii) *(1 %)* From the update $f^{(m)}(x) = f^{(m-1)}(x) + \alpha_m G_m(x)$, show that the new weights satisfy

$$w_i^{(m+1)} \;=\; w_i^{(m)} \cdot \exp\big(-y_i\, \alpha_m\, G_m(x_i)\big),$$

and explain in one short sentence *why* this gives bigger weight to currently-misclassified observations.

(iv) *(1 %)* On a separate run on the loan-default training set, the AdaBoost ensemble with $M = 300$ stumps reaches test sensitivity 0.56, test specificity 0.92, and test error rate 0.16 on the same 1,200 test loans. Compare these to the logistic regression numbers from part (a)(vi). Briefly state, in one or two sentences, which classifier you would recommend to the bank *if* the cost of approving a loan that will default is substantially larger than the cost of denying a loan that would not have defaulted, and which classifier you would recommend if it is the other way round.

c) A neural-network classifier and a backprop derivation *(8 %)*

The bank also fits a feed-forward neural network classifier with the following architecture (using the six predictors as input, after standardizing the four continuous ones and one-hot encoding the two-level `employment` dummies for a total of $p = 7$ inputs):

- input layer of 7 predictors;
- one hidden layer of $M = 12$ neurons, biases included, ReLU activations;
- output layer of 1 neuron, with a bias and a sigmoid activation, returning $\hat{p}$.

The network is trained by mini-batch SGD on binary cross-entropy loss $L_i = -\big[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)\big]$ with mini-batches of size 64, dropout at rate 0.2 on the hidden layer, early stopping on a held-out 20% of the training set, and label smoothing with $\varepsilon = 0.05$.

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

(ii) *(1 %)* A particular hidden neuron has weights $w = (0.30,\ -0.20,\ 0.10,\ 0.50,\ -0.40,\ 0.20,\ 0.60)^\top$ and bias $b = -0.10$. For a standardized input $x = (1.0,\ -0.5,\ 0.0,\ -1.0,\ 1.0,\ 0.0,\ 0.5)^\top$, compute the pre-activation z and the post-activation $h = \text{ReLU}(z)$, both to two decimals.

(iii) *(3 %)* Consider, for one training observation (x, y), the simplified 1-hidden-layer network with output $\hat{p} \in (0, 1)$:

$$z_k = \alpha_{k0} + \sum_{j=1}^{p} \alpha_{kj}\, x_j, \qquad h_k = g(z_k), \qquad \eta = \beta_0 + \sum_{k=1}^{M} \beta_k\, h_k, \qquad \hat{p} = \sigma(\eta) = \frac{1}{1 + e^{-\eta}},$$

trained with the binary cross-entropy loss $L = -\big[y \log \hat{p} + (1 - y) \log(1 - \hat{p})\big]$ for $y \in \{0, 1\}$ and a generic differentiable activation $g(\cdot)$.

(a) *(1 %)* Show that $\dfrac{\partial L}{\partial \eta} = \hat{p} - y$. (Use $\sigma'(\eta) = \sigma(\eta)(1 - \sigma(\eta)) = \hat{p}(1 - \hat{p})$.)

(b) *(1 %)* Using (a) and the chain rule, compute $\dfrac{\partial L}{\partial \beta_k}$ (for $k \geq 1$) in terms of $\hat{p}$, y, and h_k.

(c) *(1 %)* Using (a) and (b), compute $\dfrac{\partial L}{\partial \alpha_{kj}}$ in terms of $\hat{p}$, y, β_k, $g'(z_k)$, and x_j.

(iv) *(2 %)* For the dropout, early stopping, and label smoothing regularizers used above, briefly state in one short sentence each: *when* the regularizer is active (training, inference, both), and *what* concrete change it makes to the loss or to the network's forward pass at training time.

(v) *(1 %)* A colleague suggests increasing the dropout rate from 0.2 to 0.5 "to regularize even more aggressively." Briefly comment, in one short sentence, on whether this is standard practice in this course, and on the bias–variance reason a dropout rate this high is rarely used.

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