TMA4268 Statistical Learning V2026

Mock Exam 4 (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 ordinary least squares, the covariance matrix of the fitted coefficient vector is $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$. When two predictors are nearly linearly related, the matrix $\mathbf{X}^\top\mathbf{X}$ becomes nearly singular and the standard errors of the affected coefficients explode. This phenomenon is called _____(1) (*heteroscedasticity / collinearity / leverage / double descent*). One way to escape this problem without dropping a variable is to add a quadratic penalty on the coefficients to the OLS objective, a technique known as _____(2) (*the lasso / partial least squares / ridge regression / forward stepwise selection*).

When the goal is to honestly estimate the test error of a model, the simplest tool is the validation-set approach; a more reliable variant rotates the held-out part through k equal-sized folds and averages the k resulting errors, a procedure known as _____(3) (*k-fold cross-validation / the bootstrap / stratified sampling / permutation testing*). A common pitfall is to use the response y to pre-select predictors on the full data and only then run cross-validation on the reduced predictor set; this gives an unbiased estimate of test error only when the selection

step is moved *inside* each fold, an idea known as _____ (4) (*bootstrap aggregation / stratified k-fold / leave-one-out CV / nested cross-validation*).

In neural networks, the parameters are typically fit by an iterative optimizer that, at each step, takes a small step in the direction of the negative gradient computed on a random subset of the training data. This optimizer is called _____ (5) (*coordinate descent / mini-batch SGD / the Newton-Raphson method / conjugate gradient*). The gradient itself is computed efficiently by repeated application of the chain rule from output back toward input, an algorithm called _____ (6) (*boosting / forward propagation / backpropagation / the EM algorithm*).

A neural network with many more parameters than observations is, as a rule in this course, trained with some form of regularization. One option is to randomly zero out a fraction of the hidden-unit outputs at each training iteration; this technique is called _____ (7) (*label smoothing / batch normalization / cost-complexity pruning / dropout*). Another option is to monitor the validation error during training and return the parameter values from the epoch that minimized it, a technique called _____ (8) (*early stopping / annealing / transfer learning / boosting*). A third option, especially useful when the recorded class labels may themselves contain errors, replaces the one-hot target vector $(0, \dots, 1, \dots, 0)$ by a softened version such as $(\varepsilon/(C-1), \dots, 1-\varepsilon, \dots, \varepsilon/(C-1))$; this technique is called _____ (9) (*data augmentation / label smoothing / Laplace smoothing / weight decay*).

Finally, an ensemble method for trees that fits weak learners *sequentially*, each one to the negative gradient of the loss of the running ensemble, is called _____ (10) (*bagging / random forests / gradient boosting / stacked generalization*).

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 and double descent *(3 %)*

Mark each statement as true or false.

(i) The bias–variance decomposition $\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Bias}^2[\hat{f}(x_0)] + \text{Var}[\hat{f}(x_0)] + \sigma^2$ is an algebraic identity that holds for any fitted $\hat{f}$, regardless of whether $\hat{f}$ minimizes a squared-error loss.

(ii) In the over-parameterized regime where many fitted models achieve zero training error, the bias–variance decomposition no longer holds and must be replaced by a separate "double-descent identity".

(iii) Reducing the variance of $\hat{f}$ always requires a compensating increase in its bias.

b) Cross-validation and the wrong-way CV trap *(4 %)*

Mark each statement as true or false.

(i) In ordinary k-fold CV with $k = 10$, the procedure fits the model 10 times (not n times).

(ii) Suppose you have $n = 100$ observations and $p = 5000$ random predictors, and you (a) compute the marginal correlation $\hat{\rho}_j$ between each predictor x_j and the response y on the full data, (b) keep the 25 predictors with the largest $|\hat{\rho}_j|$, and (c) run 10-fold CV on a logistic regression using only those 25. The resulting CV misclassification rate is an unbiased estimator of the test error of this procedure.

(iii) Nested CV uses an *inner* cross-validation loop to select a hyperparameter and an *outer* cross-validation loop to estimate the test error of the *procedure that includes the selection step.*

(iv) For ordinary least squares, leave-one-out CV can be computed without refitting the model n times, by exploiting the diagonal h_{ii} of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.

c) Neural network parameters and forward pass *(3 %)*

Consider a fully-connected feed-forward neural network with:

- 4 input variables,
- one hidden layer of 5 neurons (with biases, ReLU activations),
- one hidden layer of 3 neurons (with biases, ReLU activations),
- one output neuron (with bias, linear activation).

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

(ii) *(2 %)* A neuron in the *first* hidden layer has weights $w = (1.0, -0.5, 2.0, 0.0)$ and bias $b = -1.5$. Its inputs on a given observation are $x = (1, 2, -1, 4)$. Using ReLU activation, compute the output of this neuron.

d) Backprop and mini-batch SGD *(3 %)*

Mark each statement as true or false.

(i) Backpropagation is an *optimizer* that updates a network's weights by stepping in the direction of the negative gradient.

(ii) In mini-batch SGD with batch size $m \ll N$, the gradient computed on a single batch is an unbiased estimator of the full-data gradient $\nabla_{\boldsymbol{\theta}} L$.

(iii) Doubling the mini-batch size from m to $2m$ tends to *reduce* the variance of the per-step gradient estimate, and accordingly *reduces* the implicit regularization that mini-batch SGD provides.

(iv) For a feed-forward network with squared-error output loss, the backward-pass "seed" at the output layer is the residual $\delta_i^{\text{out}} = y_i - \hat{f}(x_i)$ itself, not its negative.

e) Neural-network regularization *(3 %)*

Mark each statement as true or false.

(i) Dropout in a neural network is active at *both* training and test time.

(ii) In this course, the prof's typical default dropout rate for hidden layers in fully-connected tabular networks is around 20%, with rates near 50% flagged as too aggressive on signal-poor tabular data.

(iii) "Early stopping" means halting training as soon as the *training* loss stops decreasing.

(iv) Label smoothing replaces hard one-hot targets by softened ones, and is motivated in part by the possibility that the recorded class labels themselves may contain errors.

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

Mark each statement as true or false.

(i) AdaBoost re-weights the *training observations* between rounds so that points misclassified by the current ensemble receive larger weight on the next round.

(ii) In gradient boosting with squared-error loss, fitting the next tree to the current residuals is mathematically equivalent to fitting a tree to the *negative gradient* of the loss with respect to the current ensemble's predictions.

(iii) In gradient boosting, decreasing the shrinkage/learning rate ν generally requires *fewer* trees M to achieve a given training fit, because each tree contributes more strongly.

(iv) XGBoost differs from vanilla gradient boosting in that it adds explicit L_1 / L_2 regularization on the leaf-weight values, and also uses second-order (curvature) information about the loss when choosing splits.

g) Collinearity *(3 %)*

Mark each statement as true or false.

(i) Strong collinearity among predictors typically *inflates* the standard errors of the affected coefficients, making each one individually look statistically insignificant even when the predictors jointly explain a large fraction of the variance in y.

(ii) Strong collinearity always degrades the predictive accuracy of the model on a held-out test set.

(iii) Suppose a linear regression model includes both age (in years) and age_decades = age/10 as predictors. The OLS estimates of the two coefficients are well-defined but their standard errors are infinite.

h) Principal component analysis *(3 %)*

You perform PCA on a dataset with seven **standardized** variables $X_1, \ldots, X_7$ and obtain the following eigenvalues of the (centered) sample covariance matrix:

PC	1	2	3	4	5	6	7
Eigenvalue	2.5	1.5	1.1	0.8	0.5	0.4	0.2

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

$$\phi_1 = (\,0.50,\ -0.40,\ 0.45,\ 0.30,\ -0.35,\ 0.40,\ 0.15\,)^\top.$$

A new observation has standardized values $x^* = (\,2,\ 0,\ 1,\ -1,\ 0,\ 1,\ 2\,)^\top$.

(i) *(1 %)* How many principal components must be retained to capture at least 80% of the total variance? Show the cumulative-variance calculation.

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

(iii) *(1 %)* True or false: "Running PCA on the same dataset *without* first standardizing the variables would generally yield the same loading vectors, because PCA is invariant under coordinate rescaling."

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

Mark each statement as true or false.

(i) In agglomerative hierarchical clustering, replacing *single* linkage with *complete* linkage on the same dissimilarity matrix can never change the identity of the *first* merge, because the first merge is always between the two observations with the smallest pairwise dissimilarity.

(ii) Because the K-means algorithm depends on a random initialization of the cluster centroids and converges only to a local minimum of the within-cluster sum of squares, it is recommended in practice to run it from several different random initializations and pick the solution with the smallest within-cluster sum of squares.

Problem 3 (16 %) — Theory and hand calculations

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

Consider a regression problem in which the response is generated as

$$y_0 = f(x_0) + \varepsilon, \qquad \mathbb{E}[\varepsilon] = 0, \qquad \mathrm{Var}(\varepsilon) = \sigma^2,$$

where x_0 is a fixed test point and the noise ε is independent of any random training set. Let $\hat{f}$ be a model fit on a random training sample $\mathcal{D}$ drawn independently of ε, and write $\hat{f}(x_0)$ for its prediction at the test point. Expectations below are taken jointly over $\mathcal{D}$ and over ε.

(i) *(2 %)* Show that the expected squared prediction error at x_0 splits cleanly into a "reducible" and an "irreducible" part:

$$\mathbb{E}\Big[(y_0 - \hat{f}(x_0))^2\Big] \;=\; \mathbb{E}\Big[(f(x_0) - \hat{f}(x_0))^2\Big] \;+\; \sigma^2.$$

Be explicit about *which* of the cross terms vanish and *why*.

(ii) *(3 %)* Starting from your answer to (i), show that the reducible part further decomposes as

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

Be explicit about the "add and subtract $\mathbb{E}[\hat{f}(x_0)]$" step and the vanishing cross term.

(iii) *(1 %)* Combining (i) and (ii), state the full bias–variance decomposition of $\mathbb{E}[(y_0 - \hat{f}(x_0))^2]$, identify the irreducible component by name, and state what kind of *randomness* the expectations $\mathbb{E}[\hat{f}(x_0)]$ and $\mathrm{Var}[\hat{f}(x_0)]$ are taken over.

(iv) *(1 %)* Explain in one or two sentences why the prof in this course has urged you to call the result a "*decomposition*" rather than a "*trade-off*." (You may, for example, refer to the existence of regularizers that reduce variance without increasing bias, or to the over-parameterized / benign-overfitting regime.)

(v) *(1 %)* Lasso regression at a small positive penalty λ has, on a particular dataset, a test MSE that is *lower* than ordinary least squares. Using the decomposition you derived, write down in one or two short sentences which of the three terms changed and in which direction. (No numbers required; conceptual answer.)

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

You have a training set $(\mathbf{X}, \mathbf{y})$ of size n, a model class indexed by a hyperparameter λ taking values in a finite grid $\Lambda = \{\lambda_1, \ldots, \lambda_L\}$, and you wish to (a) *select* the best λ and (b) report an *honest* estimate of the test error of the resulting procedure. You have decided to use $K_{\text{outer}} = 5$ outer folds for assessment and $K_{\text{inner}} = 5$ inner folds for selection.

(i) *(3 %)* Write *pseudocode* (math, plain English, or programming-language-style is fine) for the nested k-fold CV procedure. Your pseudocode should make explicit, by the structure of nested loops:

- which data are used to *fit* the model;
- which data are used to *select* the hyperparameter;
- which data are used to *assess* the test error;

- how the outer-fold scores are aggregated into a single test-error estimate.

(ii) *(1 %)* A classmate proposes an alternative "shortcut": run a single (not nested) $K_{\text{inner}} = 5$ CV to pick the λ with the smallest CV error, and then *report that same CV error as the test-error estimate of the procedure.* Briefly explain in one sentence why this estimate is biased *downward*.

c) Bootstrap standard error *(4 %)*

You have an i.i.d. sample $X_1, \ldots, X_n$ from an unknown distribution and you wish to estimate the standard error of the sample *median* $\hat{\theta} = \text{median}(X_1, \ldots, X_n)$. No clean closed-form expression for the sampling distribution of $\hat{\theta}$ is available.

(i) *(2 %)* Write *pseudocode* for the bootstrap algorithm that returns an estimate $\widehat{\text{SE}}_{\text{boot}}(\hat{\theta})$. Be explicit about (i) how each bootstrap sample is drawn, (ii) what statistic is recomputed on each bootstrap sample, and (iii) how the final standard-error estimate is computed from the B recomputed values.

(ii) *(1 %)* For $n = 6$, what is the probability that observation X_1 is included in a given bootstrap sample? (Numeric, three decimals.) What does this probability tend to as $n \to \infty$? (Symbolic plus a numeric approximation.)

(iii) *(1 %)* A second statistician wants a 95% bootstrap *confidence interval* for $\hat{\theta}$ (not just a standard error). Briefly describe the *percentile* method in one or two sentences.

Problem 4 (22 %) — Data analysis: regression on apartment rents

A property-management company records $n = 750$ city apartments. The response is the monthly `rent` (in 1000 NOK). The available predictors are:

- `area` — total floor area in m^2 (continuous);
- `rooms` — number of rooms (count, 1–6);
- `area_per_room` — `area`/`rooms`, in m^2/room (continuous);
- `age` — age of the building in years (continuous, 0–120);
- `floor` — which floor the apartment is on (continuous, 0–15);
- `distance` — distance to the city centre, in km (continuous, 0.2–15);
- `has_balcony` — binary 0/1 indicator;
- `type` — categorical, 3 levels: *flat* (reference), *studio*, *loft*.

The data are split 500/250 into a training and a test set. The continuous predictors `area`, `rooms`, `area_per_room`, `floor`, and `distance` are standardized to mean 0 and standard deviation 1 *before* fitting any of the models below. The variable `age` (and consequently $\texttt{age}^2$) is kept on its original scale, in years.

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

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

$$\begin{aligned}\texttt{rent} \sim\ & \texttt{area} + \texttt{rooms} + \texttt{area\_per\_room} \\ & + \texttt{age} + I(\texttt{age}^2) + \texttt{floor} + \texttt{distance} \\ & + \texttt{has\_balcony} + \texttt{type} \\ & + \texttt{distance:has\_balcony}.\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**t-value**	**Pr($>\lvert t\rvert$)**
(Intercept)	14.50	0.95	15.26	< 0.001
`area`	5.80	2.10	2.76	0.006
`rooms`	2.40	1.95	1.23	0.219
`area_per_room`	0.50	1.40	0.36	0.720
`age`	−0.080	0.018	−4.44	< 0.001
$I(\texttt{age}^2)$	0.00045	0.00012	3.75	< 0.001
`floor`	0.45	0.18	2.50	0.013
`distance`	−1.95	0.28	−6.96	< 0.001
`has_balcony`	0.70	0.30	2.33	0.020
`type_studio`	−2.10	0.45	−4.67	< 0.001
`type_loft`	1.40	0.60	2.33	0.020
`distance:has_balcony`	0.55	0.22	2.50	0.013

Multiple $R^2 = 0.812$, Adjusted $R^2 = 0.808$. Residual standard error: 1.85 on 488 d.f.

(i) *(1 %)* How many parameters does this model estimate, including the intercept? Verify your count against the printed residual degrees of freedom ($n_{\text{train}} = 500$).

(ii) *(2 %)* Notice that the three predictors `area`, `rooms`, and `area_per_room` are not algebraically independent: `area` = `rooms` · `area_per_room` (and after standardization, they remain nearly so). Look at the estimate / SE / p-value pattern in the rows for `rooms` and `area_per_room`. Identify (in one short sentence) the statistical phenomenon at work, and state what would happen to the standard errors of these three predictors if you dropped `area_per_room` from the model.

(iii) *(1 %)* A classmate concludes: "Since the p-value of `rooms` is 0.219 and that of `area_per_room` is 0.720, neither variable is associated with rent, so the model captures no information about apartment size beyond `area`." Briefly say in one sentence why this reading is *wrong* given the diagnostic you identified in (ii).

(iv) *(2 %)* The model includes a quadratic term $I(\texttt{age}^2)$. Treating $\hat{\beta}_{\texttt{age}}$ and $\hat{\beta}_{\texttt{age}^2}$ jointly, find the value of `age` (in years) at which the model's predicted partial effect of *age* on `rent` is *minimized*, holding all other predictors fixed. Show your work; one decimal is fine.

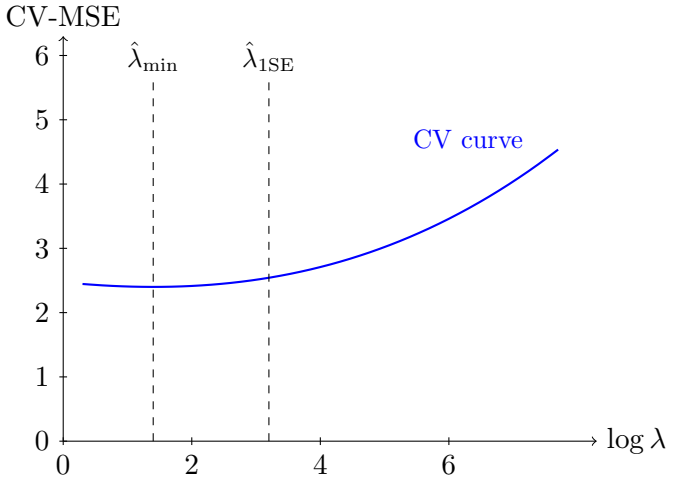
(v) *(2 %)* The interaction `distance:has_balcony` is included. *State your standardization convention for* `distance` *explicitly* (recall that all continuous predictors have been centred to mean 0 and scaled to standard deviation 1 before fitting, so a one-unit change in the predictor as it enters the model corresponds to a one-standard-deviation change on the original scale), then compute, in this model, the predicted change in `rent` (in 1000 NOK) when `distance` is increased by one standard deviation, separately (a) for an apartment with no balcony and (b) for an apartment with a balcony. Briefly comment in one sentence on why the two answers differ.

(vi) *(1 %)* The model reports adjusted $R^2 = 0.808$. The course staff considers adding a second polynomial term $I(\texttt{age}^3)$ to the model. State, with one short justification, whether you would expect the *training* R^2 to increase, decrease, or stay the same, and what about *adjusted* R^2.

(vii) *(1 %)* Suppose you would like to test whether `type` (a 3-level categorical) is *jointly* relevant after controlling for the other predictors. State (briefly) which test you would use and write the null hypothesis. You do *not* need to compute the test statistic.

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

To attack the collinearity flagged in (a), the staff re-fits the same predictors (now without the redundant `area_per_room`; $p = 10$ predictors after dummy coding) with a lasso. The penalty parameter λ is chosen by 10-fold cross-validation:



The test-MSE values, evaluated on the held-out 250 apartments, are:

	Test MSE	**Nonzero coefs**
OLS (full model from part (a))	3.85	11
Lasso at $\hat{\lambda}_{\min}$	3.40	8
Lasso at $\hat{\lambda}_{1\mathrm{SE}}$	3.50	5

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

(ii) *(1 %)* Briefly explain why the lasso reduces the number of nonzero coefficients as λ increases, whereas ridge regression does not.

(iii) *(2 %)* Compare lasso-at-$\hat{\lambda}_{\min}$ (test MSE 3.40, 8 nonzero) to OLS (test MSE 3.85, 11 nonzero). Interpret the comparison in bias–variance terms, and connect to your diagnosis in part (a)(ii).

(iv) *(1 %)* A practitioner argues: "I'm just going to use $\hat{\lambda}_{1\mathrm{SE}}$ because it gives a simpler model." Briefly state the *one-standard-error rule* and give one short reason why this practitioner's preference is reasonable.

c) Gradient boosting *(4 %)*

A gradient-boosted regression-tree model is fit on the same training set with parameters $M = 800$ trees, interaction depth $d = 4$, and shrinkage $\nu = 0.05$. The test MSE is 2.20, lower than every method tried above.

(i) *(2 %)* Write *pseudocode* for a single round of gradient boosting in the regression case (squared-error loss). Be explicit about (i) what quantity the new tree is fit to, (ii) what role ν plays in the update, (iii) how the running ensemble is updated. You do not need to spell out the tree-fitting routine internally.

(ii) *(1 %)* The prof has said: "*by updating according to the residuals, what we were really doing is fitting the gradient of the function we're optimizing.*" Briefly verify this by computing the (negative) gradient of $L(y, f) = \frac{1}{2}(y - f)^2$ with respect to f, and identifying it.

(iii) *(1 %)* Comparing the test MSEs (boosting 2.20, lasso 3.40, OLS 3.85), what does the ranking suggest about the structure of the relationship between the predictors and `rent`? Name one diagnostic plot you would produce from the boosted ensemble to recover some interpretability about *which* predictors matter.

d) GAM degrees of freedom *(3 %)*

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

$$\texttt{rent} = \beta_0 + s(\texttt{area}, \mathrm{df} = 4) + s(\texttt{age}, \mathrm{df} = 5) + \mathrm{bs}(\texttt{distance}, \mathrm{knots} = \{2, 4, 7\}) + \beta_{\texttt{floor}} \cdot \texttt{floor} + \beta_{\texttt{bal}} \cdot \texttt{has\_balcony} +$$

where $s(\cdot, \mathrm{df} = k)$ is a smoothing spline with k effective degrees of freedom (intercept removed), $\mathrm{bs}(\cdot, \mathrm{knots} = \dots)$ is a cubic regression spline with the specified interior knots (intercept removed), and $g(\texttt{type})$ uses dummy coding as in part (a).

(i) *(1 %)* How many degrees of freedom does the cubic regression spline $\mathrm{bs}(\texttt{distance}, \mathrm{knots} = \{2, 4, 7\})$ 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 %)* In one sentence, state one *structural limitation* of this GAM compared to the gradient-boosted ensemble of part (c). (Hint: think about interactions.)

Problem 5 (24 %) — Data analysis: image classification of skin lesions

A teaching dermatology clinic has $n = 6{,}000$ images of skin lesions, each labelled by a panel of physicians as `lesion_type` $\in \{\text{benign}, \text{malignant}\}$. The data are split 4,500/1,500 into a training set and a test set. The available features are tabular "hand-crafted" summaries of each image:

- `size_mm` — maximum diameter of the lesion, in mm (continuous);
- `asymmetry` — a continuous asymmetry score in $[0, 1]$;
- `darkness` — mean pixel intensity of the lesion (continuous);
- `age` — patient age in years;
- `sex` — categorical: *female* (reference), *male*;
- `family_history` — binary 0/1 indicator.

Among the 1,500 test images, 300 are truly malignant and 1,200 are truly benign.

a) Logistic regression with an interaction *(9 %)*

The clinic fits the model

$$\begin{aligned}\operatorname{logit}\big(\Pr(\texttt{lesion\_type} = \text{malignant} \mid X)\big) &= \beta_0 + \beta_1 \texttt{size\_mm} + \beta_2 \texttt{asymmetry} + \beta_3 \texttt{darkness} \\ &\quad + \beta_4 \texttt{age} + \beta_5 \texttt{sex} + \beta_6 \texttt{family\_history} \\ &\quad + \beta_7 (\texttt{size\_mm}{:}\texttt{age}).\end{aligned}$$

The fitted output is:

	Estimate	**Std. Error**	**z-value**	**Pr(>|z|)**
(Intercept)	−6.20	0.55	−11.27	< 0.001
`size_mm`	0.20	0.04	5.00	< 0.001
`asymmetry`	2.50	0.70	3.57	< 0.001
`darkness`	−0.012	0.005	−2.40	0.016
`age`	0.030	0.006	5.00	< 0.001
`sex_male`	0.18	0.15	1.20	0.230
`family_history`	0.55	0.18	3.06	0.002
`size_mm:age`	0.0020	0.0008	2.50	0.012

(`size_mm` is in millimetres; `age` is in years.)

(i) *(2 %)* For a patient of age `age` = 30, by what factor do the odds of a malignant lesion change for each additional millimetre of `size_mm`, holding all other predictors fixed? Repeat the calculation for a patient of age `age` = 70. Two numeric factors, to three decimals.

(ii) *(1 %)* Briefly explain (one or two sentences) why the answer to (i) is *not* the same at `age` = 30 and `age` = 70, and why the main-effect coefficient $\hat{\beta}_{\texttt{size\_mm}} = 0.20$ alone is therefore not a useful summary of the size effect.

(iii) *(3 %)* Consider a patient with `size_mm` = 8, `asymmetry` = 0.6, `darkness` = 120, `age` = 60, `sex` =*male*, `family_history` = 1. Compute the linear predictor $\hat{\eta}$ and the fitted probability $\hat{p}$ of a malignant lesion. Show the linear-predictor sum and the sigmoid step explicitly.

(iv) *(1 %)* At the default threshold $\hat{p} = 0.5$, would this patient be flagged as "malignant"? Justify in one sentence.

(v) *(2 %)* Define, *in words appropriate to this specific dermatology problem*, what *sensitivity* and *specificity* mean. Then state, with one short clinical justification, which of the two you would weigh more heavily when choosing a threshold for this screening task.

b) A neural-network classifier with regularization *(7 %)*

A small fully-connected feed-forward neural network is fit on the same training set with the same 6 features as input (after standardizing the continuous ones). The architecture is: 6 inputs $\to$ 32 hidden ReLU units $\to$ 16 hidden ReLU units $\to$ 1 sigmoid output. The network is trained with mini-batch SGD using cross-entropy loss.

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

(ii) *(2 %)* The student who fits the network first tries it *without any explicit regularization*, with batch size 32. The training accuracy quickly reaches 100%, while the test accuracy is 0.78 — worse than the logistic regression of part (a). Briefly explain in two short sentences (a) why this outcome is consistent with the prof's iron rule that "you should never train a neural network without regularization" and (b) why, somewhat surprisingly, mini-batch SGD itself already provides *some* regularization on top of the explicit one she ultimately adds.

(iii) *(2 %)* The student then adds (a) dropout at rate 0.2 on each hidden layer, (b) early stopping on a held-out validation slice, and (c) label smoothing with $\varepsilon = 0.05$. Briefly describe what each of these three regularizers does at training time (one sentence each). For dropout, also state what changes between training time and test time.

(iv) *(1 %)* A junior data scientist proposes increasing the dropout rate from 0.2 to 0.5 to "regularize even more aggressively." Briefly state, in one sentence, why this is not the standard recommendation in this course.

(v) *(1 %)* After all three regularizers are added, the network achieves a test accuracy of 0.86, beating the logistic regression. Frame this improvement in one short sentence using the bias–variance lens you derived in Problem 3(a).

c) AdaBoost *(5 %)*

The clinic also fits an AdaBoost classifier on the same training set, using $M = 200$ rounds of *stumps* (depth-1 trees) as the weak learner. Labels are encoded as $y \in \{-1, +1\}$ with $+1$ = malignant. Recall the AdaBoost recipe: initialize weights $w_i = 1/N$; for $m = 1, \ldots, M$, fit G_m to the weighted data, compute err_m, set $\alpha_m = \log\big((1 - \mathrm{err}_m)/\mathrm{err}_m\big)$, and update $w_i \leftarrow w_i \exp\big(\alpha_m \mathbb{1}[y_i \neq G_m(x_i)]\big)$. The final classifier is $G(x) = \mathrm{sign}\big(\sum_m \alpha_m G_m(x)\big)$.

(i) *(2 %)* Briefly explain (one or two sentences) what the weight-update rule does to observations that are repeatedly misclassified across rounds, and why this re-weighting mechanism is fundamentally different from the row-resampling mechanism used by bagging / random forests.

(ii) *(1 %)* Suppose at some round m^* the chosen weak learner G_{m^*} happens to have weighted error $\mathrm{err}_{m^*} = 0.5$. What value does α_{m^*} take, and what is its effect on the final ensemble's predictions? (One short sentence each.)

(iii) *(1 %)* A second researcher reports the same AdaBoost run achieves test sensitivity 0.84, specificity 0.91, and accuracy 0.90, vs. the logistic regression at threshold 0.5 which reports test sensitivity 0.55, specificity 0.95, and accuracy 0.87. State, with one short clinical justification, which classifier you would deploy as a screening tool.

(iv) *(1 %)* AdaBoost can be shown to be a special case of gradient boosting under a particular choice of loss function. State the name of that loss function. (One word / one formula.)

d) Class imbalance and threshold tuning *(3 %)*

In the full skin-lesion dataset, roughly 20% of images are malignant.

(i) *(1 %)* A naive classifier that always predicts "benign" would achieve a test accuracy of approximately what value? Justify in one sentence.

(ii) *(1 %)* The logistic regression of part (a) at threshold $\hat{p} = 0.5$ has test accuracy 0.87, only slightly above (i). Briefly explain (one or two sentences) in what sense its sensitivity / specificity behaviour is nevertheless much better than the naive classifier's.

(iii) *(1 %)* A clinician proposes lowering the threshold from 0.5 to 0.3. State the direction of the effect on *sensitivity* and on *specificity*, and in which direction the operating point of the classifier moves along the ROC curve.

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