TMA4268 Statistical Learning V2026

Mock Exam 6 (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 below and pick the best word or phrase for each blank from the choices in parentheses. Each correct fill is worth 1 %.

Statistical learning methods that assume a specific functional form $f(x;\theta)$ for the regression function and then estimate θ from data are called _____ (1) (*nonparametric / generative / parametric / kernel*) methods. Within the supervised-learning framework, the goal of estimating f so as to forecast the response on new inputs is called _____ (2) (*inference / prediction / classification / clustering*), while the goal of understanding which inputs drive the response and by how much is called _____ (3) (*prediction / inference / regression / bootstrap*).

The expected squared test error of a fitted regression model at a new point can always be written as the sum of a squared bias, a variance, and an _____ (4) (*reducible / cross-validated / biased / irreducible*) component arising from the noise in the response itself. Among the two main shrinkage methods we discussed, _____ (5) (*lasso / both / ridge / neither*) shrinks coefficients toward zero but, except in degenerate cases, does not set them exactly to zero.

To estimate the generalization error of a fitted model without using the test set, one applies a resampling procedure. When the goal is also to *tune* a hyperparameter (e.g. the penalty λ) and then assess the resulting pipeline honestly, one must wrap the hyperparameter-selection step inside an outer resampling loop — this is called _____(6) (*stratified cross-validation / nested cross-validation / the bootstrap / the validation-set approach*).

In the neural-network module we saw that updating the parameters using the gradient computed on a small random subset of training observations, rather than the full dataset, is called _____(7) (*batch gradient descent / Newton's method / backpropagation through time / mini-batch stochastic gradient descent*). Three regularization techniques specific to neural networks are: randomly zeroing a fraction of node outputs during each training pass, called _____(8) (*dropout / label smoothing / early stopping / batch normalization*); monitoring validation loss and halting training when it begins to rise, called _____(9) (*dropout / weight decay / early stopping / data augmentation*); and replacing the hard one-hot training targets by slightly softened versions to discourage overconfident predictions, called _____(10) (*dropout / label smoothing / transfer learning / batch normalization*).

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 benign overfitting *(3 %)*

Mark each statement as true or false.

(i) For a fixed estimator $\hat{f}$, 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, with the two cross-terms in the expansion vanishing because of independence assumptions, not because of any specific property of $\hat{f}$.

(ii) Adding ridge (L^2) regularization to ordinary least squares can lower test MSE *by simultaneously lowering both squared bias and variance.*

(iii) In the over-parameterized regime ($p \gg n$), a model that fits the training data with zero residual error generalizes badly to new data.

(iv) Doubling the noise variance σ^2, with everything else held fixed, generally makes the optimal flexibility of $\hat{f}$ *lower.*

b) Cross-validation, with and without nesting *(4 %)*

(i) *(0.5 %)* For an ordinary k-fold cross-validation estimate of test MSE, increasing k from 5 to 10 generally *decreases* the bias of the estimator. *True or false?*

(ii) *(0.5 %)* The cross-validation estimate of test error obtained by LOOCV has *higher variance* than the 5-fold cross-validation estimate. *True or false?*

(iii) *(1 %)* An analyst working on a high-dimensional ($p = 5000$) problem screens the predictors by correlation with the response, keeps the top 50, and then runs 10-fold CV on a logistic regression fit to these 50 predictors. He reports an unbiased estimate of test error. *True or false?* Justify in one sentence.

(iv) *(1 %)* Briefly state the *1-standard-error rule* for selecting a hyperparameter from a CV curve. (One or two sentences.)

(v) *(1 %)* Suppose you run 10-fold CV and obtain per-fold MSEs $\{2.1, 2.4, 2.0, 2.6, 2.2, 1.9, 2.3, 2.5, 2.0, 2.0\}$. Report (a) the CV-MSE estimate of test MSE; (b) the estimated standard error of this estimate. (Two numeric answers, two decimals each.)

c) Mini-batch SGD and learning rate *(3 %)*

Mark each statement as true or false.

(i) Mini-batch sizes for neural network training are conventionally chosen to be powers of two (e.g. 32, 128, 256) for hardware-efficiency reasons, not for any statistical reason.

(ii) Increasing the mini-batch size from 32 to 1024, with all other hyperparameters fixed, decreases the noise in each gradient estimate and therefore *strengthens* the implicit L^2 regularization that small-batch SGD provides.

(iii) The learning rate η in SGD trades off speed of convergence against stability: too large an η (e.g. $\eta = 2$) makes the loss bounce around and fail to converge, while $\eta \approx 0.1$ typically works.

(iv) In SGD, the gradient of the per-sample loss is computed using *backpropagation*, which is essentially the chain rule applied so that intermediate derivatives are reused rather than recomputed.

d) Odds, log-odds, interactions *(3 %)*

(i) *(1 %)* A patient has odds 0.25 of an adverse event. What is the corresponding probability? (Two decimals.)

(ii) *(1 %)* In a fitted logistic regression model the coefficient on a continuous predictor `x` is $\hat{\beta} = 0.18$. By what *factor* do the odds of the event multiply when `x` increases by 5 units? (Two decimals.)

(iii) *(1 %)* A logistic model contains the interaction `age:treatment`, with $\hat{\beta}_{\texttt{age}} = 0.04$ and $\hat{\beta}_{\texttt{age:treatment}} = -0.03$. By what factor do the odds multiply for a *treated* patient when `age` increases by one year, holding all other predictors fixed? (Three decimals.)

e) Collinearity and identifiability *(3 %)*

Mark each statement as true or false. (You may add a short justification if helpful.)

(i) If two predictors X_1 and X_2 in a linear regression are *exactly* collinear (say $X_2 = 2X_1$), then the ordinary least-squares estimator $\hat{\beta}$ is not uniquely defined.

(ii) Near-collinearity of two predictors typically *inflates* the standard errors of their individual coefficient estimates while leaving the standard error of the *sum* of those coefficients relatively well-controlled.

(iii) When two predictors are highly correlated, the individual t-tests on their coefficients may both fail to reject H_0 while an F-test for the joint hypothesis $\beta_1 = \beta_2 = 0$ *does* reject; in such a case it is wrong to conclude from the t-tests that neither variable matters.

(iv) A categorical predictor with K levels should be encoded with K dummy variables (one per level) together with an intercept; this ensures every level is represented in the design matrix.

f) Bootstrap standard error *(3 %)*

You have an i.i.d. sample $\{x_1, \ldots, x_n\}$ from some unknown distribution F and you want a standard-error estimate for a complicated estimator $\hat{\theta} = T(x_1, \ldots, x_n)$ (where no closed-form variance formula is available).

(i) *(1 %)* Write, in math or pseudocode, the formula for the bootstrap estimate of $\mathrm{SE}(\hat{\theta})$ in terms of the B bootstrap replicates $\hat{\theta}^*_1, \ldots, \hat{\theta}^*_B$.

(ii) *(1 %)* For $B = 4$ resamples of a particular estimator you obtain $\hat{\theta}^*_b \in \{2.1, 2.5, 1.8, 2.4\}$. Compute $\widehat{\mathrm{SE}}_{\mathrm{boot}}(\hat{\theta})$. (Two decimals; this B is far too small in practice but is fine for hand calculation.)

(iii) *(1 %)* Mark each as true or false. (A) Bootstrap samples are drawn from the original data *without* replacement. (B) The bootstrap distribution of $\hat{\theta}^*$ approximates the sampling distribution of $\hat{\theta}$, but is centered around $\hat{\theta}$ rather than around the unknown true θ, so the bootstrap cannot correct for bias of $\hat{\theta}$. (C) Typical values of B are 1,000 to 10,000.

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

Mark each statement as true or false.

(i) In AdaBoost, observations that are *misclassified* by the current weak learner receive larger weights at the next iteration, *relative to* correctly classified observations.

(ii) In gradient boosting with squared-error loss, the new tree at iteration b is fit to the *residuals* of the current ensemble; this is a special case of a more general procedure in which the new tree is fit to the *negative gradient* of the chosen loss.

(iii) In a random forest the number of trees B is a *tuning* parameter that must be chosen by cross-validation, whereas in tree boosting the number of trees M is simply set "as many as you can afford" — because boosting cannot overfit by adding more trees.

(iv) Compared to plain gradient boosting, XGBoost (i) uses second-order (Hessian) information when constructing splits, (ii) supports row and column subsampling, and (iii) adds an explicit L^2 penalty on the leaf weights to the tree-fitting objective.

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

Mark each statement as true or false.

(i) Dropout is applied during *both* training and inference: at test time, a fraction of units is still randomly zeroed.

(ii) A typical dropout rate is around 20%; rates of 50% or higher are generally considered too aggressive and not used in practice.

(iii) Early stopping consists in monitoring a held-out validation loss during training and returning the weights from the iteration at which validation loss reached its minimum, rather than the weights at the end of training.

(iv) Training a neural network without any form of regularization is generally fine as long as the architecture is small enough relative to n.

i) Principal component analysis *(3 %)*

You perform PCA on a dataset with *four* **standardized** variables $X_1, \ldots, X_4$ and obtain:

PC	Eigenvalue	PVE
PC1	2.4	0.60
PC2	1.0	0.25
PC3	0.4	0.10
PC4	0.2	0.05

The loading vector for PC1 (entries rounded to two decimals) is $\phi_1 = (\,0.60,\ -0.20,\ 0.70,\ 0.30\,)^\top$.

(i) *(1 %)* What is the smallest number of components needed to explain at least 90% of the total variance?

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

(iii) *(1 %)* Mark each as true or false. (A) When the input variables are on very different scales, PCA performed on the unstandardized data will tend to be dominated by the variable with the largest variance. (B) The PC loadings are constrained so that $\|\phi_j\|_2 = 1$ for every j. (C) Among all unit-norm linear combinations of the (centered) input variables, the first principal component is the one with the *smallest* sample variance.

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

a) The mathy one — bias–variance decomposition and a shrinkage application *(8 %)*

(i) *(4 %)* Let $y_0 = f(x_0) + \varepsilon_0$ with $\mathbb{E}[\varepsilon_0] = 0$, $\mathrm{Var}(\varepsilon_0) = \sigma^2$, and let $\hat{f}$ be an estimator of f fit on a random training set $\mathcal{T}$ that is *independent* of ε_0. Derive, step by step, that

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

In your derivation make explicit:

- the two cross-terms that appear when you expand the square, and *why* each one vanishes;
- over which sources of randomness each expectation/variance is taken.

Name the three terms on the right-hand side and state which is *reducible* by choosing a different estimator.

(ii) *(3 %)* Consider the simplest possible setting: $y_i = \mu + \varepsilon_i$ for $i = 1, \dots, n$ with $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, so $f(x_0) \equiv \mu$. Define the family of shrinkage estimators

$$\hat{\mu}_c \;=\; c\,\bar{y}, \qquad c \in [0, 1].$$

Compute, as functions of c (and μ, σ^2, n):

(a) $\mathrm{Bias}^2(\hat{\mu}_c)$,

(b) $\mathrm{Var}(\hat{\mu}_c)$,

(c) the value of c that minimizes $\mathrm{Bias}^2(\hat{\mu}_c) + \mathrm{Var}(\hat{\mu}_c)$ at the test point.

Show your algebra and comment in one line on what happens to the optimal c when σ^2/n is very large compared to μ^2.

(iii) *(1 %)* In a sentence each, explain why the prof is *skeptical of the word "trade-off"* when discussing this decomposition — naming the regime in which one can reduce both the squared bias and the variance simultaneously.

b) Pseudocode for nested cross-validation *(5 %)*

A statistician wants to honestly estimate the test error of a procedure of the form "fit a ridge regression with λ chosen by inner cross-validation." She has n observations and a grid of candidate values $\Lambda = \{\lambda_1, \dots, \lambda_T\}$.

(i) *(3 %)* Write *pseudocode* for K-fold **nested** cross-validation that returns an estimate of the generalization error of this whole pipeline. Your pseudocode must:

- make the outer and inner loops explicit (using indices k and j, say);
- make clear that the *outer* fold is held out for assessment and is *not* used at all during λ-selection on its own training portion;
- indicate at which step the chosen λ is allowed to differ across outer folds, and what is returned at the end.

You may write the loss as MSE. Roughly 10–15 lines is appropriate; longer than that suggests over-engineering.

(ii) *(1 %)* Briefly explain what goes wrong if, instead of *nested* CV, the analyst simply picks λ by running a single K-fold CV over Λ and then *reports the minimum CV-MSE* on that same run as her estimate of test error.

(iii) *(1 %)* The textbook ISLP Exercise 5.3 walks through a deliberately incorrect pipeline in which an analyst first filters thousands of predictors by their correlation with y, keeping the top 50, and *then* wraps 10-fold CV around the downstream model. Explain in one sentence why this CV estimate is biased *downward*, and state the structural fix.

c) Backpropagation in a tiny network *(3 %)*

Consider a feed-forward network with one input $x \in \mathbb{R}$, one hidden unit using a sigmoid activation $\sigma(z) = 1/(1 + e^{-z})$, and one linear output:

$$z = w_1 x + b_1, \qquad h = \sigma(z), \qquad \hat{y} = w_2 h + b_2.$$

For a single training pair (x, y), the loss is squared error $L = \frac{1}{2}(\hat{y} - y)^2$.

(i) *(2 %)* Using the chain rule, derive expressions for $\partial L/\partial w_2$ and $\partial L/\partial w_1$ in terms of x, y, z, h, $\hat{y}$, w_2. You may use that $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big) = h(1 - h)$.

(ii) *(1 %)* Now plug in numbers: $x = 2$, $y = 1$, $w_1 = 0.5$, $b_1 = 0$, $w_2 = 1$, $b_2 = 0$. Compute $\hat{y}$ and $\partial L/\partial w_1$ to three decimals. (Show both the value of h and of $\hat{y}$ along the way.)

Problem 4 (18 %) — Data analysis: concrete compressive strength

A civil engineering lab records the 28-day compressive strength `strength` (in MPa) of $n = 1030$ concrete mixes. Each mix has the following predictors:

- `cement` — cement content (kg/m^3, continuous);
- `slag` — blast-furnace slag content (kg/m^3, continuous);
- `water` — water content (kg/m^3, continuous);
- `superplast` — superplasticizer content (kg/m^3, continuous);
- `coarse_agg` — coarse aggregate (kg/m^3, continuous);
- `fine_agg` — fine aggregate (kg/m^3, continuous);
- `age` — age of the specimen (days, continuous, 1–365);
- `mix_type` — categorical, three levels: *standard* (reference), *high-strength*, *lightweight*.

The data are split 700/330 into a training set and a test set.

It is well known in this domain that `superplast` and `water` are nearly redundant: superplasticizers are added *precisely* to allow the same workability at a lower water content, so the two predictors are strongly (negatively) correlated.

a) Linear regression with collinearity, interactions, and a categorical *(8 %)*

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

`strength` $\sim$ `cement`+`slag`+`water`+`superplast`+`coarse_agg`+`fine_agg`+log(`age`)+`mix_type`+`cement:mix_t`

where `cement:mix_type` denotes the (continuous $\times$ categorical) interaction. The fitted output is:

	Estimate	**Std. Error**	**t-value**	**Pr($>\lvert t\rvert$)**
(Intercept)	11.20	14.50	0.77	0.440
`cement`	0.110	0.012	9.17	< 0.001
`slag`	0.084	0.010	8.40	< 0.001
`water`	-0.150	0.090	-1.67	0.096
`superplast`	0.300	0.260	1.15	0.249
`coarse_agg`	0.014	0.009	1.56	0.119
`fine_agg`	0.018	0.010	1.80	0.072
log(`age`)	7.40	0.30	24.67	< 0.001
`mix_high`	-3.20	1.10	-2.91	0.004
`mix_light`	-5.80	1.20	-4.83	< 0.001
`cement:mix_high`	0.030	0.015	2.00	0.046
`cement:mix_light`	-0.020	0.014	-1.43	0.154

Multiple $R^2 = 0.71$, Adjusted $R^2 = 0.707$. Residual standard error: 6.40 on 688 d.f. F-statistic on the joint hypothesis $H_0 : \beta_{\texttt{water}} = \beta_{\texttt{superplast}} = 0$: $F_{2,688} = 21.3$, $p < 10^{-9}$.

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

(ii) *(2 %)* The individual t-tests on `water` and `superplast` are both insignificant at $\alpha = 0.05$, yet the joint F-test on the pair rejects with $p < 10^{-9}$. (a) Explain in one or two sentences why this is the *expected* symptom when two predictors are highly correlated — relate your answer to the variance-covariance matrix $\mathrm{Var}(\hat{\beta})$. (b) On the basis of this output alone, would it be correct to conclude that `water` and `superplast` are both irrelevant and can be dropped from the model? Briefly justify.

(iii) *(2 %)* For a concrete mix of type *lightweight*, by how much (in MPa) does an increase of one kg/m^3 in `cement` change the predicted `strength`, holding all other predictors fixed? Repeat the calculation for a *high-strength* mix. (Two numeric answers.)

(iv) *(1 %)* A colleague refits the same model after re-labelling `mix_type` so that *lightweight* becomes the new reference level (instead of *standard*). State whether each of the following quantities changes or stays the same: (a) the residual standard error; (b) the coefficient on `cement`; (c) the coefficient on `mix_high`; (d) the predicted strength of any given concrete mix.

(v) *(1 %)* A new mix has predictor values that lie inside the convex hull of the training data, and one would like to predict its 28-day strength. The analyst is asked to attach an *interval* to the prediction. State, in one sentence each: (a) the difference between a 95% confidence interval for $\mathbb{E}[\texttt{strength} \mid x_0]$ and a 95% prediction interval for an individual new `strength` at x_0; (b) which of the two is wider, and *why*.

(vi) *(1 %)* The residuals-vs-fitted plot for this model shows a clear "fanning out" pattern (residual spread grows with the fitted value). Name the assumption violated and state, in one short sentence, one common remedy.

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

The same predictors are now fed (after standardization) into a ridge regression. A 10-fold cross-validation is run on a grid of λ values; the resulting CV-MSE curve has its minimum at $\hat{\lambda}_{\min}$ and the 1-SE choice gives a noticeably larger $\hat{\lambda}_{1\mathrm{SE}}$. On the held-out test set:

Method	**Test MSE (MPa^2)**
OLS	41.5
Ridge at $\hat{\lambda}_{\min}$	39.8
Ridge at $\hat{\lambda}_{1\mathrm{SE}}$	40.6

(i) *(1 %)* Why is it important to standardize the predictors *before* fitting ridge regression? (One sentence.)

(ii) *(1 %)* Write the ridge regression objective and state what happens to $\hat{\beta}^R_\lambda$ as $\lambda \to 0$ and as $\lambda \to \infty$.

(iii) *(1 %)* Briefly state the *1-standard-error rule* for choosing $\hat{\lambda}_{1\mathrm{SE}}$ from a CV curve.

(iv) *(2 %)* Interpret the test-MSE ranking $\mathrm{ridge}(\hat{\lambda}_{\min}) < \mathrm{ridge}(\hat{\lambda}_{1\mathrm{SE}}) <$ OLS in *bias–variance* terms. In particular: (a) what does the OLS–vs–ridge gap tell you about whether OLS is over- or under-fitting on this dataset; (b) why does the more heavily-regularized $\hat{\lambda}_{1\mathrm{SE}}$ fit sit *in between*?

c) Gradient boosting *(5 %)*

A gradient-boosted regression-tree ensemble is fit on the same training data. It uses M trees, each of (small) interaction depth d, with a shrinkage parameter ν. It achieves test MSE = 26.3 on the same test set.

(i) *(2 %)* Give a brief but accurate description (math *or* pseudocode, 5–8 lines) of one iteration of squared-error gradient boosting: what is fit, what is updated, and where the shrinkage parameter ν enters. State explicitly which quantity the new tree at iteration $b+1$ is fit to.

(ii) *(2 %)* Of the three hyperparameters (M, d, ν): (a) state *how* you would tune each one in practice (one short sentence each); and (b) describe the qualitative *coupling* between M and ν — specifically, what happens to the required M when ν is halved.

(iii) *(1 %)* The test MSEs across the regression problem are: OLS = 41.5, ridge = 39.8, boosting = 26.3. What does this large gap between ridge and boosting suggest about the underlying relationship between predictors and `strength`?

Problem 5 (28 %) — Data analysis: telecom customer churn

A telecom operator records, for $n = 5000$ customers, a binary response `churn` (whether the customer cancelled service within the next 6 months). The predictors are:

- `tenure` — months as customer (continuous, 1–72);
- `monthly` — monthly charges, EUR (continuous);
- `age` — customer age in years (continuous);
- `contract` — categorical, three levels: *month-to-month* (reference), *one-year*, *two-year*;
- `senior` — binary, 1 if customer is age ≥ 65, else 0.

The data are split 3500/1500 into training and test. In the full sample, the empirical churn rate is approximately 26%.

a) Logistic regression with a categorical × continuous interaction *(8 %)*

A logistic regression is fit on the training set with all five predictors and an interaction `monthly:contract`. The output is:

	Estimate	**Std. Error**	**z-value**	**Pr($>\|z\|$)**
(Intercept)	−1.90	0.35	−5.43	< 0.001
`tenure`	−0.040	0.004	−10.0	< 0.001
`monthly`	0.030	0.005	6.00	< 0.001
`age`	−0.010	0.004	−2.50	0.012
`contract_1yr`	1.50	0.40	3.75	< 0.001
`contract_2yr`	2.20	0.50	4.40	< 0.001
`senior`	0.20	0.10	2.00	0.046
`monthly:contract_1yr`	−0.020	0.006	−3.33	< 0.001
`monthly:contract_2yr`	−0.040	0.008	−5.00	< 0.001

(i) *(2 %)* For a customer on a *month-to-month* contract, by what factor do the odds of churn multiply when `monthly` increases by 10 EUR, holding all other predictors fixed? Repeat the calculation for a customer on a *two-year* contract. (Two numeric answers, three decimals.)

(ii) *(1 %)* Briefly explain why the main-effect coefficient on `monthly` (here 0.030) is not, by itself, a meaningful summary of "how monthly charges affect churn on average," and state which other quantities in the table must be combined with it depending on the customer's contract.

(iii) *(3 %)* Consider a customer with the following profile: `tenure` = 12, `monthly` = 70, `age` = 45, `contract` = *one-year*, `senior` = 0. Compute the predicted probability $\hat{p}$ of churn for this customer. Show the linear predictor $\hat{\eta}$ step by step (one line per term), then apply the sigmoid.

(iv) *(1 %)* At a default classification threshold $\hat{p} \geq 0.5$, is this customer predicted to churn? Comment in one sentence on whether 0.5 is an appropriate threshold here given the base rate of churn (26%).

(v) *(1 %)* A colleague refits the same model, but he has used a different encoding convention in which the reference level of `contract` is *two-year* instead of *month-to-month*. State briefly which of the following change: (a) the predicted probability of churn for any given customer; (b) the signs and magnitudes of the coefficients `contract_1yr`, `contract_2yr`; (c) the residual deviance / overall fit of the model.

b) AdaBoost with pseudocode and a small hand calculation *(6 %)*

The team also fits an AdaBoost classifier with $M = 200$ depth-1 trees (… "decision stumps").

(i) *(3 %)* Write *pseudocode* for the AdaBoost algorithm with M rounds. Your pseudocode should make explicit:

- how observation weights w_i are initialized;
- how, at round m, the weighted training error err_m, the classifier weight α_m, and the updated weights $w_i^{(m+1)}$ are defined;
- the form of the final classifier $G(x)$.

Assume class labels are coded as $y_i \in \{-1, +1\}$.

(ii) *(1 %)* At round m, the weak learner G_m achieves a weighted misclassification error of $\text{err}_m = 0.30$. Compute the corresponding classifier weight α_m. (Two decimals.)

(iii) *(1 %)* Using α_m from (ii), by what factor is the weight w_i of a *misclassified* observation multiplied at the next iteration? By what factor is the weight of a *correctly classified* observation multiplied? (Two numeric answers, two decimals each.)

(iv) *(1 %)* The prof emphasized in lecture that AdaBoost (and tree boosting more generally) uses "weak learners" — shallow trees rather than deep trees. Give, in one or two sentences, a bias–variance argument for *why* deep individual trees would make a poor boosting base learner.

c) A neural-network classifier with regularization *(8 %)*

The team builds a feed-forward neural network classifier with:

- 7 input variables (the five raw predictors, plus the two `contract` dummies);
- one hidden layer with 16 ReLU neurons and biases;
- one output layer with 1 neuron, sigmoid activation, and a bias.

The network is trained by mini-batch SGD on binary cross-entropy loss, with dropout (rate 20%) on the hidden layer, early stopping on a held-out 20% of the training set, and label smoothing ($\varepsilon = 0.05$).

(i) *(2 %)* How many parameters does this network have in total, *including all biases*? Give the layer-by-layer breakdown.

(ii) *(2 %)* A particular hidden neuron has weights $w = (0.5,\ -0.3,\ 0.1,\ 1.0,\ -0.5,\ 0.2,\ 0.4)^\top$ and bias $b = -0.2$. For an input $x = (1,\ 0,\ 2,\ 1,\ 0,\ -1,\ 1)^\top$, compute the output of this neuron under the stated ReLU activation. Show the pre-activation z and the post-activation h.

(iii) *(1 %)* Explain in one or two sentences why a feed-forward network with *no* non-linear activations (e.g. identity activations on every layer) cannot benefit from being made deep — and what the resulting model is mathematically equivalent to.

(iv) *(1 %)* For each of the three regularizers used (**dropout**, **early stopping**, **label smoothing**), state in one short sentence *what* it does mechanically during training. Whether at training time, inference time, or both — be specific about *when* each one is active.

(v) *(1 %)* A junior analyst suggests increasing the dropout rate to 50% to "get even more regularization." Comment in one short sentence on whether this is a good idea, citing the standard convention.

(vi) *(1 %)* Suppose the team instead chose to remove all regularization (no dropout, no early stopping, no label smoothing, no weight decay). State, in one sentence, what symptom you would expect to see in the train-loss vs. validation-loss curves over the training epochs.

d) LDA vs. QDA, plus random-forest variable importance *(4 %)*

LDA, QDA, and a random forest are also fit on the same training data, using all five predictors. On the test set their confusion matrices give the test error rates: LDA 0.196, QDA 0.221, random forest 0.179.

(i) *(2 %)* State the two main modelling assumptions of LDA. Then give one bias–variance reason why one might prefer LDA to QDA when one of the classes is small (here: "churners" are roughly 26% of the data, so about 900 training points). Use one short paragraph.

(ii) *(1 %)* For the random forest with $p = 7$ predictors at the tree-fitting stage and a binary classification task, state the standard default value of `mtry` and briefly explain why one would want `mtry` $< p$.

(iii) *(1 %)* The random forest's variable-importance plot reports very high importance for `tenure` and `contract`, and low importance for `age`. State *one* thing that this plot *cannot* tell you about the underlying relationship between these predictors and `churn`.

e) Class imbalance *(2 %)*

The base churn rate is $\approx 26\%$.

(i) *(1 %)* A trivial classifier that always predicts "no churn" achieves what test error rate on this dataset? Justify in one sentence.

(ii) *(1 %)* Comparing your answer in (i) to the test error rates of LDA (0.196), QDA (0.221), and the random forest (0.179) above, briefly comment on what this tells you and which metric or metrics you would privilege for picking the best classifier in this setting.

End of exam. Total: $10 + 28 + 16 + 18 + 28 = 100$ points.